Master Thesis

for the study programme MSc. Interaction Technology

Designing a System for Evaluating the Performance of Computer Vision Applications, and Managing the Events Generated by these Applications

October 2020

UNIVERSITY OF TWENTE
DEEPOMATIC

AUTHOR:
Priscilla Onivie Ikhena, MSc Candidate
Study Programme: MSc Interaction Technology
Email: niniikhena@gmail.com

GRADUATION COMMITTEE:
Dr. Randy Klaassen
Faculty: Electrical Engineering, Mathematics and Computer Science
Department: Human Media Interaction (HMI)
Email: r.klaassen@utwente.nl

Dr. Mariët Theune
Faculty: Electrical Engineering, Mathematics and Computer Science
Department: Human Media Interaction (HMI)
Email: m.theune@utwente.nl

Thibaut Duguet
Company: Deepomatic
Position: Senior Product Manager
Email: thibaut@deepomatic.com

CONTENTS

1. INTRODUCTION
    - 1.1 Deepomatic
    - 1.2 Deepomatic's Approach - Lean AI
    - 1.3 Problem Statement
    - 1.4 Research Question
    - 1.5 Outline

2. BACKGROUND
    - 2.1 What is AI
    - 2.2 Industrializing AI
    - 2.3 Computer Vision Systems
    - 2.4 The Life-cycle of a Computer Vision System
    - 2.5 Challenges with Implementing, Evaluating and Managing Computer Vision Systems in Industrial Settings
    - 2.6 No-code / Little-code AI Platforms
    - 2.7 Related Work
    - 2.8 Conclusion

3. APPLICATION EVALUATION - UX PROTOTYPE ITERATION
    - 3.1 User Research and Ideation
        - 3.1.1 Interviews
        - 3.1.2 Personas
        - 3.1.3 User Journeys
        - 3.1.4 Ideation
    - 3.2 First Iteration - Low Level Prototype and Review
    - 3.3 Second Iteration - Mid Level Prototype and Study
        - 3.3.1 First Task - Creating and Evaluating a New App - Day One Experience
        - 3.3.2 Second Task - Creating an App Version and Marking it Ready for Deployment (None Day One Experience)
        - 3.3.3 Third Task - Deploying the App Version in Production
        - 3.3.4 Study
    - 3.4 Third Iteration - High Level Prototype
    - 3.5 Conclusion

4. MONITORING EVENTS - UX PROTOTYPE ITERATION
    - 4.1 User Research and Ideation
        - 4.1.1 Interviews

4.1.2 Personas
4.1.3 User Journey
4.1.4 Ideation
4.2 First Iteration - Low Level Prototype and Review
4.3 Second Iteration - Mid Level Prototype and Study
4.3.1 Study
4.4 Conclusion

5. DISCUSSION
5.1 Limitations
5.2 Future Work

6. CONCLUSION

LIST OF FIGURES

Figure 1.0	The Lean AI Loop, Deepomatic White paper - Lean AI Methodologies
Figure 1.1	Life-cycle of a computer vision system
Figure 1.2	Woman checking out at her company's cafeteria, the checkout system is powered by the Deepomatic's Smartcheck out app. Source: Deepomatic.com
Figure 1.3	woman checking out at her company's cafeteria, the checkout system is powered by Deepomatic's Smart Check out app. Source: Deepomatic.com
Figure 3.1	The Annotator Persona
Figure 3.2	The Annotator Manager Persona
Figure 3.3	The AI Manager Persona
Figure 3.4	The Solution Architect Persona
Figure 3.5	Core Personas Relationship
Figure 3.6	The Customer Persona
Figure 3.7	The Solution Architect/AI Manager's User Journey
Figure 3.8	Paper sketch of Landing Page
Figure 3.9	Application Evaluation Landing Page
Figure 4.0	Set up evaluation Paper Sketch
Figure 4.1	Setting up Evaluation
Figure 4.2	Selecting Application Paper Sketch
Figure 4.3	Selecting Application
Figure 4.4	Importing Groundtruth
Figure 4.5	Defining KPIs, KPI Details
Figure 4.6	Defining KPIs, KPI Formula
Figure 4.7	Defining KPIs, KPI Formula
Figure 4.8	Choosing a Subset in creating a KPI
Figure 4.9	Choosing a Subset in creating a KPI
Figure 5.0	Running the evaluation
Figure 5.1	Viewing Evaluation Results
Figure 5.2	Viewing Evaluation Results
Figure 5.3	Viewing Evaluation Results
Figure 5.4	An illustration of applications and application Versions
Figure 5.5	Landing page
Figure 5.6	Get started page
Figure 5.7 a	App Info
Figure 5.7 b	Defining App Workflow
Figure 5.7 c	Configure application - Defining KPIs

Figure 5.8	Clicking on an application from the deployments page
Figure 5.8 a	App Detail Page
Figure 5.8 b	App Detail Page
Figure 5.8 c	App Detail Page
Figure 5.9	Creating new app version
Figure 6.0	Send Deploy Notification
Figure 6.1	Viewing Deployment Notification
Figure 6.2	Updating the site with the latest app version
Figure 6.3 a	Mid Level version of Create App Template
Figure 6.3 b	High Level/Updated version of Create App Template
Figure 6.4 a	Viewing an Imported Workflow
Figure 6.4 b	Viewing an Imported Workflow
Figure 6.5 a	Defining Evaluation Metrics
Figure 6.5 b	Defining Evaluation Metrics
Figure 6.5 c	Defining Evaluation Metrics
Figure 6.6 a	Adding an Event Set
Figure 6.6 b	Adding an Event Set
Figure 6.6 c	Adding an Event Set
Figure 6.7 a	Viewing the Detail Page of an App
Figure 6.7 b	Viewing the Detail Page of an App
Figure 6.7 c	Viewing the Detail Page of an App
Figure 6.7 d	Viewing a metric chart that has been re-scaled
Figure 6.8	The Technician Persona
Figure 6.9	The Operator Persona
Figure 7.0	The IT Manager Persona
Figure 7.1	The Persona Relationship for Augmented Workers
Figure 7.2	User Journey of the Technician
Figure 7.3	Landing page of events monitoring
Figure 7.4 a	Activating attributes
Figure 7.4 b	Viewing the activated attributes
Figure 7.5	Deactivating a Single Attribute
Figure 7.6 a	Reordering the position of attributes
Figure 7.6 b	Reordering the position of attributes
Figure 7.7 a	Deleting a single event
Figure 7.7 b	Carrying out a bulk action on all events.
Figure 8.0	Searching through events.
Figure 8.1 a	Sorting events
Figure 8.1 b	Sorting events
Figure 8.2 a	Landing page

Figure 8.2 b	Searching with a specific ID
Figure 8.3 a	Activating Attributes
Figure 8.3 b	Filtering through events
Figure 8.3 c	Specifying the date
Figure 8.3 d	Viewing search and filtered results
Figure 8.4 a	Viewing events that occurred on the week of the 9th.
Figure 8.4 b	Viewing events that occurred on the week of the 9th.
Figure 8.4 c	Viewing the Event Details of an Event
Figure 8.4 d	Changing status to KO from OK.
Figure 8.5	Viewing the Technician's Comment.
Figure 8.6 a	Reassigning the Event
Figure 8.6 b	Reassigning the Event
Figure 8.7	Results breakdown
Figure 8.6 c	Shopping site

1. INTRODUCTION

This thesis project is about designing systems that allow non-expert users with little to no programming knowledge, to easily carry out performance evaluations on their computer vision systems before they are deployed in production, and to monitor and manage the events generated by these systems post deployment.

1.1 DEEPOMATIC

For my thesis and internship project, I carried out research and designedUX experiences at a French startup called Deepomatic. Deepomatic [*Deepomatic's Website*] is an artificial intelligence (AI) startup whose ambition is to deploy image recognition applications and solutions on an industrial scale, and empower their client enterprises to properly manage and benefit from these systems. In doing this, they enable enterprises to better reach their business goals, by automating certain processes that these client enterprises already have in place, using computer vision, Computer Vision and artificial intelligence. With Deepomatic's end to end solutions, enterprises are able to create these custom AI applications and solutions, and operate them at scale in as little as three months. They aim to enable them to do this with little or no prior software programming knowledge, thereby rendering AI and AI systems more accessible to non-expert users that work at these enterprises.

Deepomatic provides a platform that strives to allow their enterprise clients to manage the entire life-cycle of computer vision applications and solutions. They work with their enterprise clients to implement and deploy computer vision solutions that meet their needs, in different types of industries, with use-cases such as Augment Workers, Smart Checkout Systems, Object Sorting for Waste Management, Quality Control for Field Service Management, and Alerting for Security CCTVs. However, the applications of these types of solutions are quite vast.

Within most industries these days, executives are looking to create faster and more cost-effective ways to deliver products and build innovative services that differentiate them from their competitors, and Deepomatic essentially strives to enable them to do this with solutions powered by AI and Computer Vision.

1.2 DEEPOMATIC'S APPROACH - LEAN AI

Throughout this thesis, we will be delving into computer vision systems, and how we can design systems that allow them to be evaluated and managed by non-expert users with little or no programming experience.

However, in this section, we explore Deepomatic's approach in making Image Recognition, Computer Vision and AI successful in the Industrial field. Deepomatic approach leverages Lean AI methodology and tools, which borrows from both lean manufacturing and lean startup methodologies.

Lean AI stems from Lean Management, a method of setting up and managing a business, whereby product development cycles are shortened, and smaller changes are made incrementally. In doing this, the overall process is made more efficient. Lean Management is said to have driven changes in corporate culture and in general when introducing artificial intelligence into manufacturing, part of the process now includes the adoption of Lean Management. [*Gaspari, Four Principles, Lean AI*].

Lean AI is thus the concept created from merging Lean Management with Artificial Intelligence. By merging the two, we create an even more efficient way of managing an organization by leveraging the capabilities that Artificial Intelligence and Machine Learning have in providing smart and predictive insights to users. In doing this, human resources are freed up allowing them to spend more time focusing on solving issues, rather than investigating them [*Gaspari, Four Principles, Lean AI*].
Thus in the world of businesses and manufacturing, Lean AI is known to be a great asset across all types of industries and in the world of Industrializing AI [*Gaspari, Four Principles, Lean AI*].

By leveraging Lean AI, Deepomatic is able to not only implement these Computer Vision and AI solutions, but also create a life cycle that enables these solutions to get improved over time. The goal of Lean AI, is to improve the process of building a product, by removing certain steps that lead to a waste in time and resources, while navigating the uncertainty of production conditions. It's model is an iterative and agile one, carried out by repeating the following steps over and over, the performance of the system is then improved. Deepomatic implements the Lean AI approach using these steps [*Deepomatic's Website*]:

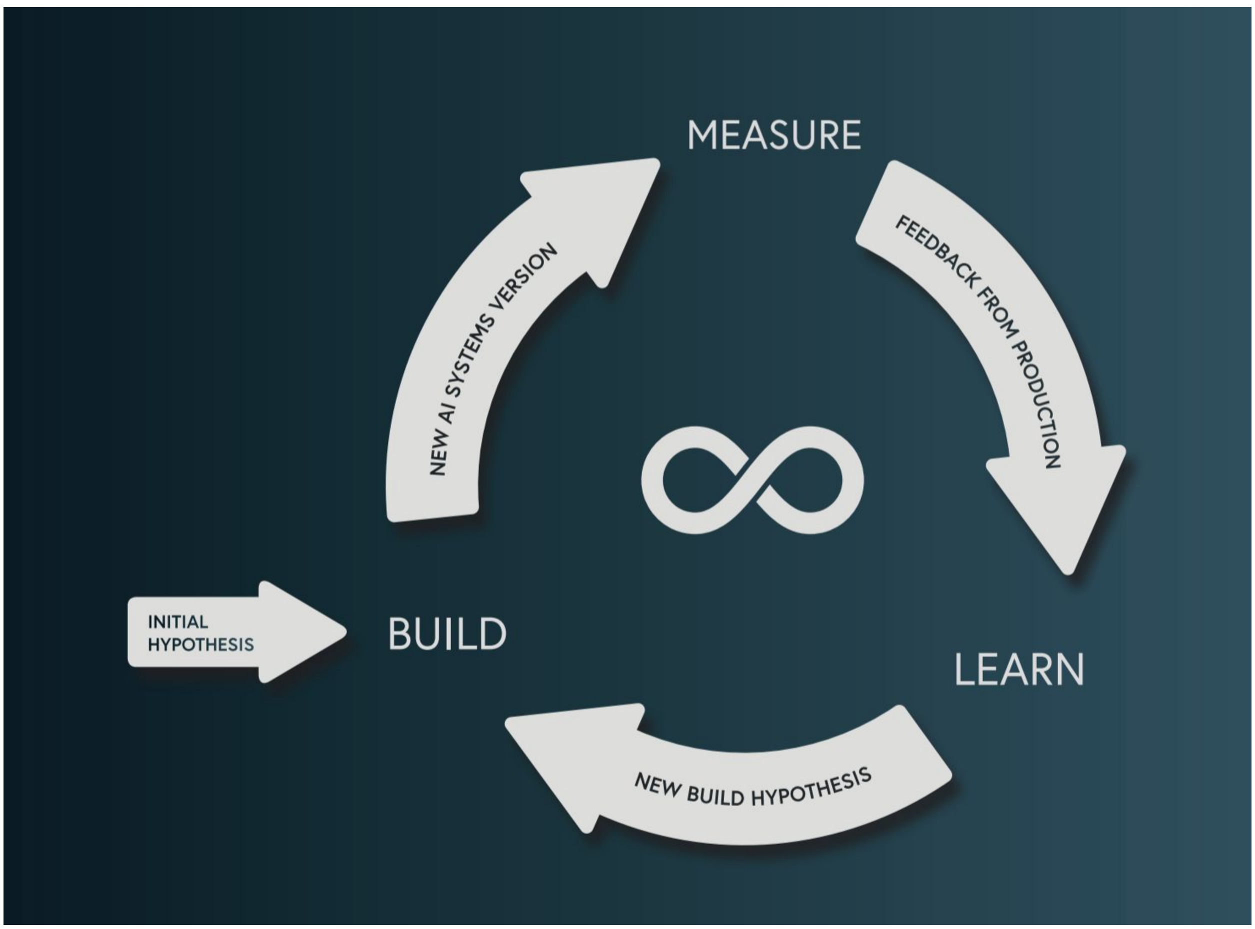


Figure 1.0 - The Lean AI Loop, Deepomatic Whitepaper - Lean AI Methodologies

Build:

As seen in *Figure 1.0*, the very first step of the Lean AI process, after defining the initial hypothesis, is the Build step. Image annotation is a key part of the build process. Image annotation is an important task in computer vision and is what enables a computer to *see.* With image annotation, images are labeled by humans, usually an AI engineer, who typically works for Deepomatic. This engineer then provides information on what objects are in the image. The process of doing this can be quite time consuming, depending on the amount of labels present in an image. Some projects only need one label to be tagged in an image, while others contain images with multiple components that all need to be tagged.
[*Deepomatic Whitepaper - Lean AI Methodologies*].

In the Build process, a dataset of images is taken, all images are annotated by the data annotators, and then AI models are trained with the annotated images to assemble them into an AI system for a given project [*Deepomatic Whitepaper - Lean AI Methodologies*]. This task of training models, is typically done in conjunction with annotation, as an iterative loop [*Deepomatic Whitepaper - Lean AI Methodologies*]. After the models have been trained and the application/AI system has been implemented, it is then evaluated to ensure it performs properly, before it is deployed in production to be used officially by the enterprise client.

Measure:
In this step, the new AI system that has been deployed, as seen in *Figure 1.0*, is measured to see how well it performs in production mode. The first step in measuring the quality of such an AI system, is to deploy it in production, and depending on the nature of the AI system, it may be possible to get corrective feedback on how the system is performing. Corrective feedback are typically human validations that confirm or contradict the outcome proposed by the AI system. Often they are gathered as part of the use of the system through a human-machine interface. When it comes to choosing what performance metrics are used to evaluate the system's performance, these metrics typically have the following characteristics [*Deepomatic Whitepaper - Lean AI Methodologies*]:

- ***Comparative*** - Meaning, it is possible to compare the performance with previous versions of the system over a period of time.
- ***Responsive*** - Meaning the chosen metric is easy enough to compute so that it doesn't slow the cycle iteration.
- ***Linked to Business Value*** - Meaning the metric directly contributes to a specific business KPI, which helps the enterprise client better understand how well the system improves their business, and ensure that the Lean AI loop is optimizing for the right thing.

Learn: Lastly, gathering feedback from the production, that is used to then understand where the AI system could be improved [*Deepomatic Whitepaper - Lean AI Methodologies*]. This feedback is then integrated into the next version of the system. The main use of the learn phase, is to generate a new set of assumptions for the next cycle of the Lean AI loop. These new assumptions are usually composed of a new set of images to annotate.
Once an AI system is in production and is being used by the enterprise client, it has the possibility to send back data to a central location in order to create the new set of images to annotate. This means, raw data (images or videos), predictions made by the system and the corrections made by the user, are all sent back in order to improve the system and feedback to the loop [*Deepomatic Whitepaper - Lean AI Methodologies*].

With these three main steps, the loop is continued, and thus the AI systems are made more efficient when deployed in an industrial setting. Although these three steps simplify the process of implementing AI in an industrial setting, each step in the process is quite layered and is a composition of sub-steps, that all together need to be implemented with efficiency and functionality. The process of evaluating these applications before they are deployed, is a sub-step of the Build step, and is one that is crucial to ensuring that there are no problems with the application functioning in production after it has been deployed. Similarly, the process of managing and monitoring the events generated by these applications post deployment, is a sub-step of the *Measure* step and is one that also needs to be implemented as efficiently as possible, in order to keep the Lean AI loop running smoothly. However, the process of rendering these sub-step easy to maneuver by users that have little to no programming experience, remains a great challenge, and is in line with on-going discussions around making Artificial Intelligence more accessible as a whole.

1.3 PROBLEM STATEMENT

From the common news and research around Artificial Intelligence and AI applications/solutions, there is a common theme that AI remains a blackbox. Most people, including experts, do not know how it works, what it can be used to do, and the opportunities it offers [*Hayes et al. 2017*]. Consequently, it is also challenging for non-expert users to understand various stages that are part of the life-cycle of their AI solutions, such as developing the application, evaluating it, deploying it and then monitoring/managing it.

As AI systems become ubiquitous in our lives, the human side of the equation needs more careful attention and investigation [*Zhu et al. 2018*]. More specifically, the more companies pick up using AI to automate a lot of their processes, the more crucial it is becoming to make AI easy to understand by non-expert users who work for these enterprises. Right now, not enough systems and platforms are available that permit ease of use to these users [*Zhu et al. 2018*].

In this thesis, we will work towards unraveling how the entire life-cycle of implementing a computer vision system works and how users interact with them to get their work done, and then, we will work towards demystifying the challenges that lie in certain stages of this life-cycle and interaction. We will also explore what current solutions exist to resolve some of these challenges, and then design two UX experiences on the Deepomatic Studio platform, in an attempt to resolve these challenges as well, and render these systems easier to implement, use and interact with by non-expert users.

It is also important to note that it seems to be the case that in order to explain AI systems that are complex, the human interaction with the AI system will need to be simplified and rendered more natural, and it is this process that is often referred to as demystifying AI *[Brock et al. 2019]*.

1.4 RESEARCH QUESTIONS

Following our quest to resolve some of the challenges with rendering the implementation of the life-cycle of an AI application more accessible to non-expert users, the main research question that will be explored in this paper is:

How do we design a more accessible UX experience for carrying out the evaluation of computer vision systems and the monitoring and management of the events they generate after they have been deployed in production, and are being used by enterprise clients?

From that, we deduce these six sub-questions that will need to be explored in order to answer our main question:

- SRQ1: *What is a computer vision system?*
- SRQ2: *Who are the stakeholders involved in implementing a computer vision system?*
- SRQ3: *What are the challenges involved in implementing a computer vision system?*
- SRQ4: *What is the life-cycle of a computer vision system?*
- SRQ5: *What challenges are typically experienced at the **system performance evaluation** and the **event monitoring** stages of the life-cycle?*
- SRQ6: *What are existing ways to go about resolving some of these identified challenges?*

1.5 OUTLINE

In the following chapters, we will explore related work, the state of the art and research around the implementation of Computer Vision and computer vision systems, the life-cycle, the challenges encountered during the implementation and evolution of computer vision systems, as well as the relevant stakeholders involved at each stage.

In order to answer the sub-research questions, we will review literature and related work. We will then try to answer the main research question by using all of the research and insights gathered to propose a solution and then create prototypes inline with this solution, that will be tested with real users and iterated upon, until we arrive at a system that resolves some of the challenges explored.

In Chapter Two - Background and Related work, we will give some background context to what AI is, and the importance and state of the art of the industrialization of AI. We will then answer the first six sub-research questions stated in section 1.4.

In Chapters Three and Four, we will explore the UX research to explore potential solutions we could use to go about resolving some of the identified challenges, based on how Deepomatic currently leads with resolving some of them. We will also explore different UX (user experience) iterations of the solutions we came up with, and test them with a set of real users. In Chapter Five, we will gather the insights and results gotten from carrying out this study, and then in Chapters Six we will discuss these results, what could be implemented in the future, limitations we had, and finally in Chapter 6, we will conclude on the study and answer the main research question we started with.

2. BACKGROUND

Computer vision is one of the most important fields to have stemmed from deep learning and AI. In this chapter, we'll delve into a brief history of artificial intelligence, to understand how it all began and the different types/categories of artificial intelligent systems. Then we will delve into what computer vision is, how it functions as an AI system, and how computer vision technologies are being leveraged today by different industries and sectors to improve a variety of processes.

2.1 WHAT IS AI

The beginning of AI dates back to the 1950s, when two computer scientists, Minsky and McCarthy, coined artificial intelligence as any task performed by a computer that would be considered intelligent if a human had performed the same task.
It is a field in computer science that focuses on the ability of machines and computers to act and react to things the way humans do *[Mijwil. 2015]*.
By categorizing AI technologies based on their intelligence, we get the following main types of AI as of date *[Mijwil. 2015]:*

Artificial Narrow Intelligence (ANI):

This type of AI is often referred to as "weak AI" *[Miaihe & Hodes. 2017]*, and is focused on completing or performing a single task. This task could be driving a car, or recognizing a face or someone's speech, and so forth. Thus, ANI is quite intelligent when it comes to completing a particular task based on the way it has been programmed. Examples of such a program include Google's Search engine, Siri by Apple, Alexa by Amazon and other virtual assistants.

Artificial General Intelligence (AGI): If ANI is considered weak AI, AGI is considered the stronger version of AI or deep AI as it is the category of machines that have intelligence similar to that of a human. It is also able to learn and use this intelligence to solve future tasks and problems. As of today, AI researchers haven't been able to achieve AGI because by doing so, they would need to create consciousness in machines by implementing a full group of cognitive abilities which is a massive task *[Miaihe & Hodes. 2017]*.

Artificial Super Intelligence (ASI): This is a type of AI that is considered hypothetical and is said could have existential consequences for the human kind *[Miaihe & Hodes. 2017]*. With ASI, not only is human behavior mimicked, but it is explained as when machines themselves achieve self-awareness that supersedes that of human intelligence. ASI is a concept that has been largely used in science fiction and although it may seem exciting it may also come with threatening consequences *[Miaihe & Hodes. 2017]*.
For the purposes of this thesis, we will be focusing on the only type of AI we have currently been able to implement, and is being used across several spaces and industries - ***Artificial Narrow Intelligence*** *[Hodes et al. 2017].*
ANI is known to be mainly used in these ways today:

Expert Systems: Expert systems represent one of the most important research areas of artificial intelligence *[Hadzic et al. 2015].* An expert system is a computer program that solves problems using inference procedures. These solutions often take a significant effort of intellect and intelligence. However, it becomes limited when data is lacking.

Machine Learning: Machine learning is a way of applying artificial intelligence through computer algorithms thereby making it possible to improve automatically through learning from experience *[Bishop, 2006].* It is a sub-field in Artificial Intelligence that covers a range of statistical techniques giving computers the ability to learn. That is, they can progressively improve their capacity to execute a task over time and the ability to learn. There are more than a dozen of these statistical techniques, of which deep learning is one of them. Machine Learning algorithms are used in different applications such as filtering emails, computer vision and other areas where it is difficult to use conventional algorithms to carry out certain tasks *[Bishop, 2006].* Oftentimes, depending on the way feedback is given back to the system, Machine Learning is divided into these three categories:

- **Supervised Learning:** Supervised learning is a way to implement Machine Learning where the ML system is given the input data already labeled, and what the expected output should be as well. The AI system is in that way guided to know what to look for and is trained until it is able to identify underlying patterns and connections between the input and output data. By doing this, when the system sees new data it hasn't been trained with, it is to predict good results [*Bishop, 2006].* It is often used in Risk Evaluation and Forecast Sales.

- **Unsupervised Learning:** Unlike Supervised Learning, no labels or expected output is given to the learning algorithm, thus, the system has to find structure in its given input. With the goal often discovering hidden patterns and learning about features based on the patterns in the input data *[Bishop, 2006].* Unsupervised Learning is often used in recommendation systems and anomaly detections.

- **Reinforced Learning:** In this case, the training of the machine learning models is done to enable the computer program to interact with a changing environment by performing given goals through making a sequence of decisions *[Bishop, 2006].* The program learns to achieve a goal in an uncertain and potentially complex environment. The program employs trial and error to come up with a solution to the problem. To get the machine to do this, the AI system gets either rewards or penalties for the actions it performs, and the goal is to overall maximize the total reward. An example of this includes gaming and self-driving cars.

Natural Language Processing(NLP): NLP is a sub-field of Artificial Intelligence that focuses on interactions between human languages referred to as *natural languages* as well, and computers. More specifically, it is the way by which we program computers to process large sets of language data. It is utilized in chatbots and virtual assistants such as Apple's Siri and Amazon's Alexa, for the most part.

Computer Vision: If NLP is for words, then Computer Vision is for images and videos. It is a field focused on how computers see the world and understand images and video in the way humans do. It aims to understand tasks that our visual system can do, and mimics complex parts of the human vision system as well there by enabling computers to view the world the way we do *[Ballard et al. 1982]*. An example of a computer vision system that recognizes faces in a human way is called DeepFace, and it's able to recognize faces with an accuracy of 97.25% *[Taigman et al. 2014].* As mentioned earlier in chapter one, Deepomatic is also focused on implementing computer vision systems for their enterprise clients, which they often refer to as *Visual Automation systems.*

Automated Speech Recognition (ASR): As NLP is concerned with the meaning of words, and computer vision is concerned with recognizing images and videos, ASR is concerned with the meaning of sounds. It is also considered closely linked with recognising images and videos, and NLP. Speech Recognition applications include voice user interfaces, speech to text processing, determining a speaker's characteristics and so forth *[Nguyen. 2010]*.

AI Planning: Also known as, Automated Planning and Scheduling, is a branch of AI concerning strategies and action sequences. Self-driving cars and other autonomous robots need AI planning to operate *[Malik et al. 2004]*.

With this overview of what AI is, its different categories, and some of the application domains of existing ANI systems, we are going to explore in the next section how these systems exist in the context of the industrialization of AI, and how they are leveraged by enterprises today.

2.2 INDUSTRIALIZING AI

The industrialization of AI is the application of Artificial Intelligence systems such as the ones discussed in the section above, to the challenges that come with complex industrial operations.

Machines and efficiency have always been a part of the industrial revolution. When we travel back in history to the 17th century to see what industrialization looked like then, we see that the industries then ran quite slowly. Workers had to create objects by hand, because mass production didn't exist. Workers in that age would see today's world as simply magical. The biggest change between then and now is in the introduction of machines into a lot of business processes to make them more efficient *[Leurent et al. 2019].* The industrial revolution that started in 1760, allowed us to build products at faster rates, and to scale up creations quickly to levels we once deemed impossible. There were also a number of industries created over time as a result such as the shipping industry, furniture, automobile industries and so forth. What the industrial revolution did was replace physical labour that humans performed with machines that could lift weights much heavier than us, and speed up processes that once took us months and years to complete *[Leurent et al. 2019].*

Artificial Intelligence has been spoken about as the ***next industrial revolution*** of this era *[Lee et al. 2019],* with the belief being that the world as we know it will change when organizations are able to use smarter solutions to make current processes more efficient and automated. There are a vast number of areas where AI can be applied in industries, in business and in society. This all matters because from the personal assistants that also exist in our mobile phones, to customer service and commercial interactions, AI influences almost every area of our lives and we are still at its infancy. Based on an analysis done by PwC, the world's GDP will have a 14% increase by 2030, due to the acceleration, development and adoption of AI. It is also said that by 2030, AI would have contributed up to 15.7 trillion dollars *[PwC's Global Artificial Intelligence Study]*. This economic impact of AI will be caused by a few factors. 1 - the gains business will have from automating a lot of their processes, for example the way they'll leverage robots and autonomous vehicles to carry out certain tasks). 2 - the productivity benefits businesses will have from augmenting the work employees do with AI technologies. This is generally referred to as assisted and augmented intelligence. And lastly, 3 - the impact will also come from an increased number of consumer demand due to the presence of higher-quality products and services that have been enhanced with AI *[PwC's Global Artificial Intelligence Study]*.

In general thought seems to be around how AI can solve existing problems, including those we did not realize existed *[Begam et al. 2013].* What we are witnessing with AI and Machine Learning industrialization, is that it is becoming core to a lot of enterprises around the world, as it saves them not only money and resources, but also creates new business and new product opportunities as it did in the past *[Begam et al. 2013].* If leveraged responsibly, AI has also been shown to be able to significantly break barriers we currently are unable to, and this is how a number of Industrial AI solution providers, such as Deepomatic, are approaching transforming industries. They provide value to their client enterprises by understanding what some of these barriers are in different sectors and types of industries, and then seeing how these barriers can be removed, and what processes can be sped-up, improved, automated, augmented, transformed and even made to reproduce new industries and product opportunities, through machine learning and AI.

In the next section, we look at what computer vision systems are, what role they play in the industrialization of AI and what value they offer to enterprise clients.

2.3 COMPUTER VISION SYSTEMS

The concept of computer vision was presented for the first time in the 1970s *[Huang. 1996]*. The initial ideas were exciting but lacking the technology to bring them to life. It is widely accepted that Larry Roberts is the father of Computer Vision. Many researchers have followed his work since then. However, nowadays, the world has witnessed a bigger leap in technology that now leverages computer vision more significantly and has put it on the priority list of a variety of industries.

Computer vision is a field in AI that targets the challenge of making computers see and interpret the visual world in the way humans do. Computers are able to do this based on training with photos from cameras and videos, leveraging deep learning models. They are then able to accurately identify objects and then react to them in a way similar to how we react to these same objects. Computer vision is now being used in a variety of industries such as driverless car testing, telecommunication, agriculture to monitor livestock and the health of crops, health care for daily diagnostics, and so forth *[Sathiyamoorthy. 2014]*.

Based on research, we see that computers are getting quite good at recognizing images and identifying labels and objects in these images. For this reason, a good number of today's top technology companies such as Google, Amazon, Microsoft and Facebook are investing billions of dollars into computer vision research and developing products that leverage computer vision. Here are a few ways it is being leveraged in industry today*:*

Monitoring
Computer vision aids with monitoring how well AI systems are performing and makes it easier to identify or predict errors or situations that may lead to unwanted results *[Charrington. 2017]*. By using machine learning, applications can be training with data sets to learn how the complex systems work. These applications can then be utilized later on to predict future states based on the input data.

Quality control

Because process compliance is often tiresome and expensive, enterprise clients are now starting to leverage computer vision to visually inspect work done on site, or products that are made *[Charrington. 2017]*. Often different factors are inspected to ensure quality control. This is one of the use-cases that is particularly home to Deepomatic. One of the use cases that Deepomatic's Computer Vision systems address is one that enables network and infrastructure operators and network installers to carry out real-time quality installations and diagnosis. The goal here is that their telecommunication enterprise clients leverage their computer vision application to reduce the errors made in carrying out installations, and improve the quality of the work done. This process occurs with five simple steps.

1. First, the network installer carries out an installation, which could be a cable installation, a TV installation, underground conduit, access, etc.
2. Then they take a picture of the installation.
3. The photo is then analyzed by the computer vision application that has learned to recognize the elements of interest.
4. The application then detects and locates the presence of an error, an omission or an inconsistency in the photo.
5. Depending on the case, the installer or technician may be guided by the computer vision application, until the operation is validated or a statement of conformity may be issued.

The computer vision application in this case, consists of models which all interact with each other following a provided logic. The models together with the logic behind, make up the computer vision application. These models are trained with large data-sets of images of installations, and are thus able to detect and identify installations that are wrongly done. This in turn has helped client enterprises improve the quality of their customers' experiences, customers that are in need of these installations. Thereby freeing up their operational staff to focus on tasks with higher added value *[Deepomatic's Website]*. It also enables them to receive real-time feedback of these installations and serve as a second *eye* for the technicians carrying out these installations, thereby augmenting their productivity *[Deepomatic's Website].* In addition to this use case, quality control is also leveraged by retail and retail security companies.

Retail and Retail Security

Amazon uses computer vision in retail security. One of its sub-enterprises launched in 2018, called Amazon Go, removes the need for customers to wait in long-lines while checking out. When customers lift off items from shelves, the computer vision system called *Just Walk Out* backs up the cameras, and is able to identify the customer's action. The overall system uses sensor fusion, and deep learning algorithms as well. It is able to detect who has taken an item off the shelf and what item was taken off and add it to the customer's basket. With a network of cameras around the store, the system is able to detect people in the store and keep track of their bills at all times, so that the wrong shopper doesn't leave the store without having paid. Stoplift is another company that leverages this type of computer vision system.

Optimization

In addition to monitoring, AI systems can also be used that helps optimize a business' metrics. It does so within a variety of ways through process planning, job shop scheduling, yield management, product design and so forth *[Charrington. 2017]*.

Control

Control systems are a core part of industrial operations and are needed by organizations that want to benefit from automation. This is done in a variety of ways including - robotics, autonomous vehicles, factory automation, smart grids and so forth *[Charrington. 2017]*.

Autonomous Vehicles:
1.25 million people die each year due to traffic incidents, and it is said that this will be the seventh leading cause of death by the year 2030 if nothing is done about it *[World Health Organization, 2018]*. According to this same research, most of the incidents are caused by human error and lack of attention on the road. For this reason, there are a number of companies that use computer vision to help reduce this number of incidents by creating self-driving cars.
Tesla is one of these companies that makes self-driving cars and their auto-pilot car models are said to be fully equipped for self-driving capability [*Tesla's Website]*. The camera system is called *Tesla Vision* and it is a computer vision system that has been built on a deep neural network, making it possible to move through complex roads and warn drivers to pay attention while driving. The car eventually stops running after three warnings, until the car is able to detect that the driver is paying attention again. Another company that works on self-driving cars using computer vision systems is called *Waymo [Waymo's Website]*.

In this section, we have only discussed a few methods but computer vision is also used in other sections such as Healthcare, Agriculture, and Banking and so forth.

In the next section, we will look into just what it takes to build a typical computer vision system, and what the life-cycle of such a system in an industrial setting typically is.

2.4 THE LIFE-CYCLE OF A COMPUTER VISION SYSTEM

Though it is an emerging technology, computer vision application development cycles are quite similar to that of typical applications.

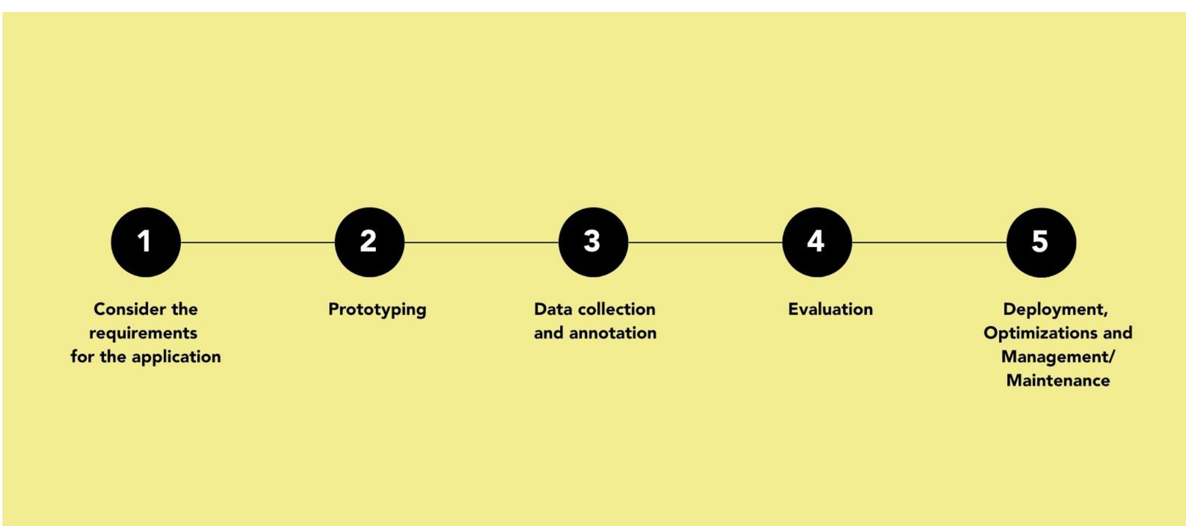


Figure 1.1 - Life-cycle of a Computer Vision System

For the purpose of rendering this life-cycle explanation easier to follow, we'll leverage one of Deepomatic's computer vision solutions - ***Automated Checkout.*** This use case explains each step of the life-cycle process in depth.

Automated Checkout

The automated checkout solution is a vision application that allows accurate identification of products and all the characteristics that influence their price in order to fully automate the invoicing or checkout process. In doing this, long wait lines are eradicated, and automatic quotes on product items are given based on image analysis. One way this solution is used is in a self-checkout system installed at corporate cafeterias in France through a company called *Compass Group*.

Compass Group

Compass Group *[Essays, UK. 2018]* is one of Deepomatic's enterprise clients. They specialize in contract catering, and provide catering services to the core sectors of Business and Industry, Health and Care for the Elderly, Education, Sport and Leisure and Defense. In Compass' use-case, they use the automated checkout solution implemented by Deepomatic, to enable smart checkout systems across all of their restaurants. Their main objectives are to reduce the checkout wait time in company restaurants, and to overall improve the customer experience people have in their company restaurants. Compass Group thus uses a Deepomatic computer vision AI application called ***Smart Checkout***, which is one of Deepomatic's many automated checkout solutions, to enable them to achieve these objectives. With this use-case in mind, we will delve into each stage of a computer vision's life-cycle, as shown in *Figure 1.1.*

Consider Requirements for Application

As shown in *Figure 1.1,* the life-cycle process of a computer vision application often begins with considering the requirements for the application/system. Who will be using it? What would they want to do with it? What type of budget one is working with? And what can be done and can't be done with machine learning. It is also important at this stage to define who and what will generate the input data *[Belani et al. 2019][Jin et al. 2020].* Organizations need to be strategic when deciding to develop an AI system/application, and choose between a "low-hanging-fruit initiative and a bold challenge" *[Desouza et al. 2019]*. Then, they would need to ensure that the right infrastructure needed is in place to complete the project successfully.

Organizations lacking in developed infrastructure, can often benefit from dealing with low-hanging fruit initiatives first *[Desouza et al. 2019].* This is said to be a smart strategy for software systems in general, but more importantly, for AI systems. Once this is done, they can then decide to build on the infrastructure they already have, while learning more about implementing AI systems. On the other hand, the organizations with in-house IT resources can right away delve into addressing bold challenges using computer vision systems.

Once this initial pre-work is completed, the challenge that needs to be solved has been identified, and all the questions have been answered, the next step is ***prototyping***.

Prototyping

Although there may be multiple use-cases for the application, it is often the case that to create a prototype, a single use-case is used. The value of prototyping is that it allows developers to determine an array of needs, and expose blind spots they may be overlooking. Once a particular use-case has been selected, the input and output requirements of the application are then properly defined. Once the initial problem and use-case has been established, collecting data comes next *[Jin et al. 2020]*. In the case of the Smartcheckout solution, we would need to use data-sets of food images that have been properly annotated to train our models later. It is usually not expected to have large amounts of datasets during the prototyping phase. Sometimes, pre-trained models also already exist and can be used. The main goal at this stage is to see if a dataset needs to be created. A lot of time and resources can be saved if the dataset-creation phase of the project can be skipped.

Data Collection and Annotation

If needed, the next step would be to collect and annotate the data. This stage is quite cumbersome and it can be done in-house or it could be contracted out to a data-annotation company to do *[Treccani, 2018]*. For instance, in the case of the Smartcheckout app, the images will be annotated with labels such as “rice pudding” “apple juice” and other food items that show up in the images of trays, and that could typically be found in cafeterias in France. Once the annotation is finalized, we’re ready for the next step.

Evaluation

This stage of the life-cycle is where the application is evaluated to ensure that it works accordingly. First, the evaluation metrics are defined indicating what metrics will be used to determine how well the application is performing *[Hernandez-Orallo. 2014][Hernandez-Orallo. 2016]*. An example of an evaluation metric for the Smartcheckout app would be the *Detection Rate*, meaning, the rate at which the application correctly identifies and detects the objects on a user’s tray while they’re in the process of checking out. Additionally, the metric targets are identified for each evaluation metric. After the application is evaluated, the results are then examined to understand if the application is ready to be deployed in production (meaning made publicly available for users in cafeterias to start using) or if there are some improvements that could be made to the project to improve the application such as training with more data, or utilizing a different model.

Deployment, Optimization, and Management/Maintenance

After the application has been evaluated and produces optimal results, the application is then deployed in production, and users are able to automatically checkout at their various cafeterias, by simply placing their trays below the checkout stand. In doing this, the deployed Smartcheckout application will detect all the items on the user’s tray, and bill them accordingly as shown in *Figure 1.2 and 1.3* below.

Figure 1.2 - woman checking out at her company's cafeteria, the checkout system is powered by the Deepomatic's Smartcheck out app. Source: Deepomatic.com

Figure 1.3 - woman checking out at her company's cafeteria, the checkout system is powered by the Deepomatic's Smartcheck out app. Source: Deepomatic.com

During the initial deployment, any failure and their associated costs are analyzed, for example, cases where the application gets certain detections wrong will be kept track of and analyzed. Another factor that could be kept track of is the detection

latency, and how long it takes to process, amongst other factors that would define how well the application is performing. These observations can be useful in optimizing the application and improving the next version of the application.
The maintenance phase usually consists of keeping track of/monitoring the events generated by the application to ensure that it works well, updating the datasets by adding additional objects to detect or incorporating the data from failure cases, updating the models with improved logic if need be and updating the app, re-evaluating it and then deploying this updated app version in production to replace the previous one, if it works better. In this way, this process of creating an app's version - evaluating it - monitoring and maintaining it and then improving it - creating a new app version again, creates a mini cycle within the life-cycle of a computer vision application, to ensure that the app is continually being improved to meet the enterprise client's (in this case, *Compass Groups) needs.*

2.5 CHALLENGES WITH IMPLEMENTING, EVALUATING AND MANAGING COMPUTER VISION SYSTEMS IN INDUSTRIAL SETTINGS

With new artificial intelligence solutions entering the markets at a fast pace, they create value, AI is now offering new approaches to do things in a variety of ways that enables businesses to make more money. However, there are challenges and obstacles that come today with implementing AI systems, in this case computer vision systems, in industrial settings *[Shaw et al. 2019]*. A big obstacle for starting AI projects is the high total cost of solution ownership, and additionally, there is a huge lack of AI specialists with all the necessary skills. There are more obstacles and in this section we delve into a few of them *[Polachowska, 12 Challenges of AI Adoption. 2019]*.

People

Lack of support from executive team
Oftentimes, not all executive leadership share the mindset change to adopt AI, and this can be due to a few reasons such as not having enough budget to update older systems they already have in place, or due to fears around not really understanding how the AI systems work. Additionally, understanding how value propositions vary for the different people involved influences the adoption of ML *[Polachowska, 12 Challenges of AI Adoption. 2019][Schlögl et al. 2019]*.

Non-technical employees
The implementation of AI requires management to also have some understanding of how AI works in general, existing AI systems, what's doable, and what is not. However, there are false beliefs that still exist around needing to build an in-house data science team in order to get started with implementing AI systems. Due to the fact that there aren't enough AI experts as it is today, many industries fall behind in adopting AI *[Polachowska, 12 Challenges of AI Adoption. 2019]*.
.
Lack of field specialists
Developing an AI solution that works successfully requires a team that has both technical and business knowledge. However, at most enterprises, it is often one or the other. The executive teams and senior leaders often do not know enough technically to enable AI adoption and on the other hand, most data-scientists are not as interested in how the models they build work and are used in real life *[Polachowska, 12 Challenges of AI Adoption]*. Outside of Facebook, Apple, Amazon, Google and Microsoft, most companies struggle to build teams that have the talent needed for AI adoption, and thus they end up falling behind. Due to this, more and more companies are starting to hire vendor companies to help them with setting up the right AI systems. For instance, Deepomatic has solution architects that work closely with enterprise

clients to understand the solutions they are trying to solve with AI. They then implement computer vision applications that solve these problems, and work with the enterprise clients to implement, and deploy these systems in production. However, this process of solving business problems with AI systems, would be more efficient and scalable, if certain employees at these enterprise clients are themselves able to evaluate, deploy and manage these applications, especially as new application versions get created along the way.

Data

One of the principal problems companies encounter with AI adoption is data related issues. Since an AI system is typically only as good as the data it is trained with, it makes having the right data a key part of building successful AI solutions. In gathering utilizing the right data, there are a number of problems that come up along the way *[Polachowska, 12 Challenges of AI Adoption. 2019]*.

Quality and Quantity of Data

As discussed above, the quality of an AI system depends on the quality of data it is trained with. AI systems often need large datasets to train them, as AI learns from trained information in the same way humans do. However, in order to detect patterns, they need to have been exposed and trained with tons of data.

Unlike humans, AI is able to process and analyze the data it is fed with at a much faster and accurate rate. The first step with solving this data issue is first knowing what type of data one needs to get in comparison to data already owned. In order to build the right data-sets, the AI manager would need to know exactly what the AI solution is intending to do, what data-sets are owned and if the data is structured or unstructured. Sometimes, it can be challenging to obtain certain types of data such as health data and sometimes due to privacy and security, certain data-sets are made difficult to access.

Data labeling/annotating

As discussed earlier, computer vision systems rely heavily on images and videos. Many of these systems are trained in a supervised way and this needs the data they are fed (images and videos) to be properly labeled. In generating and utilizing large amounts of data each day, it becomes increasingly difficult to label them but there are few existing solutions to these. There are some databases available to use that already contain labeled data such as ImageNet which contains over 14 million images that have been labeled manually. In addition to this, it is possible to also outsource companies around the world that specialize in labeling/annotating data.

Bias

There has been a lot of discussion arounds how AI systems can be biased towards women or people of color. As AI systems make predictions and operate based on the data made available to it, it is evident that if it is fed biased data, it would also output biased results. For instance, if a set of people using an AI system prefer certain features to other ones, the system would not learn about other features that are going unused. And because people themselves have stereotypes, the training data can end up leaning towards also making stereotyped predictions *[Yavuz, 2019]*.

Business

No Business Alignment

In order to know where to apply AI solutions to their business processes, managers need to have an understanding of AI technologies. Because of this lack of understanding, many organizations lag behind, or others pick up AI but have no strategy on how it will be best utilized. AI implementation needs a clear strategy so that when implemented it actually improves the processes that aids the companies in achieving their business goals and key performance indicators (KPIs). Otherwise, companies then struggle with understanding whether or not the AI system is actually helping to improve their business successfully, or not.

Challenges with Accessibility

There are a number of factors that setting up and integrating AI into a company's existing systems requires. Factors such as data storage, training with data, how well the data is labeled/data quality, and so forth, all need to be taken into account. Additionally, there's model training, evaluating how well the AI system works, creating a feedback loop to continually improve these models and applications, and data sampling to reduce the amount of data that ends up being stored *[Skrop et al, 2018]*. It is also often challenging for some businesses to know that an integrated system is working efficiently, and that it is worth their money. In order to overcome some of these challenges, certain companies work with other enterprises that specialize in AI, or vendors, and part of this work would entail making sure that everyone understands how it all works. Post integrating the computer vision application into a system, the employees would still need to be able to use these models over time, and to understand how they work, how to measure success, how to evaluate them if needed, how to evaluate and deploy updated versions of the models and how to monitor them as they are being used by customers post deployment in production. On the other hand, the vendors or AI companies that these enterprise clients work with, would also need to work with them and advise them on everyday use of their AI system, and how to improve it over time.

For the purposes of this thesis, we are particularly delving into the challenge of integration and accessibility discussed above. The question on how to make AI systems more accessible to employees in different industries is one that has persisted as a trendy topic in AI more recently. AI is generating a demand for new types of talent at work, but unfortunately there is a shortage of people with the needed technical skills and this continues to be an obstacle for most companies because it ends up being costly to hire the right people. Furthermore, when companies try to carry out these processes themselves, they have to deal with issues that come with the complexity of data management, and also with managing the applications. One notion that is being talked about as a solution to this problem is: **no-code or little-code AI** *[*Balakrishnan, *No Code Products within AI and ML. 2020][Sanchis et al. 2019].*

As the name suggests, this method of non-technical machine learning is designed for people with little or no exposure to programming. These sort of platforms make it possible to create applications, without advanced coding, resulting in a development process that is quicker and more effective. It would particularly make it easier for most businesses that lack employees with AI knowledge, to more easily set up AI systems that help them reach their KPIs more efficiently and at a faster rate.

2.6 NO-CODE / LITTLE-CODE AI PLATFORMS

What is a no-code platform

Now that the artificial intelligence and machine learning sectors are growing and evolving quite quickly, digital tools are becoming more leveraged both by large and small businesses. Thus, as the industry starts to use more no and little-code AI, the future of AI seems like it will largely demand for platforms that are safe, user friendly and require little to no code or prior programming knowledge. This notion has been on the rise in recent times, and overall with the way software development has evolved over time. In the past, we've also seen examples of such platforms with applications such as Microsoft Excel. Then, it started getting more leveraged in the mobile application and web-development spheres, and more recently it has advanced to machine learning and artificial intelligence *[Sanchis et al. 2019].*

As there is in general still a high ask for developers, because companies do not have the technical skill-sets needed to build these AI applications, companies who cannot hire developers, or those who want their developers to have more free-time to focus on other tasks, are increasing leaning towards no-code platforms *[Sanchis et al. 2019].* Such platforms help business professionals with no prior programming knowledge to easily create AI applications that their organizations need, thereby filling up the skill-set gap.

Can AI and Machine Learning be Implemented without code?

No code platforms are increasing the ease of developing an AI system by instead allowing users to build these systems using graphical user interfaces (GUI) as opposed to programming them traditionally. Usually, all the components - front-end code, back-end code, and so forth, are generated automatically. New machine learning tools are now rendering it possible to create AI applications with prior programming experience, by providing UI experiences that non-technical employees can use to build their computer vision systems. Doing this helps save time, costs and increases productivity, efficiency, and learning opportunities for employees. All of these enable businesses to create AI applications that meet their criteria, needs and expectations. If later on in the process of developing the system, some updates and adjustments need to be made, they can be done through an IT team in house or outsourced, without necessarily needing to change the application's fundamental functionality.

In the next chapter, we will explore some no-code / little-code platforms that currently exist in machine learning and artificial intelligence, and how they are being leveraged by different industries and businesses.

2.7 RELATED WORK

Artificial intelligence and machine learning are two domains that are increasingly being relied on by companies to build and deploy a variety of models in order to have a smooth operation of their businesses. As discussed previously however, doing this often requires programmers or data scientists with a good background in coding, which enterprises often lack. In trying to fill this gap, tech giants provide these tools to ensure that businesses can create the applications they need without hiring the right technical talent. Here are a few solutions that are being used to develop models without prior programming knowledge.

Google's AI Platform

Google's AI platform makes it possible for data scientists and engineers to make ideas a reality, by leveraging tools that help them to run machine learning application.s With this platform, enterprise clients are able to either improve their already existing machine learning workflows by utilizing AutomALL. The process starts by the user first storing the data they have on a cloud storage on BigQuery, next they classify the data by labelling it with labels such as images, videos, text, audio. By doing this, they create different categories of the data. After this is done, the user is then able to use the data to train their models by importing the data into their console. Once these models are trained, the user can then create the machine learning/AI applications they need on the Google Cloud Platform (GCP). They are also able to manage these models post training by using the AI platform *[Google's AI Platform]*.

Google's northstar goal is to make machine learning more accessible to everyone across the board, from engineers to data scientists. In recent times, they have made available a set of cloud based tools that have been designed to help ease the onboarding to using AI and implementing AI applications.

BigQuery

With BigQuery ML, data analysts are able to develop machine learning models using sql. Additionally, it allows them to quickly prototype by letting them run these models directly where their data is. At the moment, the types of models they are able to run are limited to regression models - linear, binary, multiclass and logistic *[Google's AI Platform]*. The only current limitation is the fact that there are only regression based models, and this limits the users from developing models that are more accurate and that are based on real world data.

ML Engine

ML Engine is a tool that is targeted to be used by both developers and data scientists. It lets them train and deploy ML models. The solution is said to work pretty well for developers who only need to quickly train models and deploy them to be used in production, without needing the help of engineers. Developers are then able to get these apps on cloud storage if needed.

Cloud ML

This is a tool that is once again targeted at developers that have a limited knowledge of machine learning. It helps them deploy a variety of pre-built models by Google, for specific use cases. Some of these models options include the AutoML Vision model for image classification, AutoML Translation for translating, and Natural Language for text classification.

Using these products, a user is able to quite straightforwardly import their data from Google Cloud Storage using the AutoML product in use, and then train the model. The performance of the model is afterwards evaluated using the command line and its results and predictions are viewed with the UI. Managing these models afterwards is also able to be done using Google's AI platform. Some of the companies currently using GCP and Cloud ML to automatically create and deploy these ML models on data successfully, include Urban Outfitters and Disney *[Mavenwave.com].*

Today, Google's Cloud Auto ML tool lets users create models through a GUI without need to write code. They're able to upload data, create a model, view its metrics and then call the model with an endpoint. Thus, developers enjoy using it. Another advantage with using Google's Cloud Auto ML is that it uses Google's Auto ML and Transfer Learning technologies to produce models that are of better quality, which then removes the need to do a lot of parameter tuning, and other common ML related challenges. All the user then has to do is label the data and upload it to AutoML. Afterwards, they are guided by a friendly UI. One of the challenges in solving business challenges with Machine Learning, is that often different people need to do different parts of the process. Meaning that usually, a data scientist would focus on building the model, and when that is done, it gets handed off to a developer who places the model in a secure end-point. Because AutoML handles this process, the speed to production is significantly reduced.

Although Cloud Auto ML is currently used to address a number of challenges for many enterprises globally, there remains a few things to consider. First, there is still the challenge of data quality as discussed above, thus it is important to account for biases while leveraging these models in order to avoid future costs *[Mavenwave.com].* Secondly, because of its easy of use, companies can get too reliant on the models quickly. And this could be dangerous because data processes and proper model evaluation would still need to be carried out on a model created by AutoML. Another challenge is that Auto ML is not really flexible when it comes to the range of problems it is able to solve. For instance when examining the distance between objects in a given image, there may need to be custom work done on a pre built model and since they're not often customizable, this poses a problem *[Mavenwave.com]*.

Create ML by Apple

Create ML is an application created by Apple that makes it possible for a user to deploy machine learning models, without having prior knowledge about machine learning. With Create ML, users are able to create models for specific targets such as image detection. The user is able to train a number of models using different datasets at the same time, and then test them before they get deployed into production. Some of the advantages of using Create ML is that it's quite easy to use, relatively speaking and a developer is able to start using it right away with data they have locally stored. However, the disadvantage is that Mac laptops and desktop machines do not have the computing horsepower needed to process large amounts of data that is typically needed to run a solid training cycle. It is also difficult to customize and tweak the model if the developer wants to optimize it, because all the algorithms are hidden from the developer *[AndPlus.com].*

What-If Tool

The What-if tool is a visualization tool created by Google's People + AI Research (PAIR) initiative. It has been designed to function in an easy way which can be used by anyone, ranging from designers to product managers. It lets users compare two models that are simultaneously running on the same datasets, by creating visual features to compare their differences. Upon doing this, a user can then edit any of the data points by adding or removing features and ultimately running a test before deploying it into production. It is known to provide transparency in the similarity of data points to ensure the right comparison is made between the two models. The tool also uses confusion matrices and ROC curves to determine the precision of the models. The main advantage of this tool is that it makes it easy and playful to train ML models with a UI that is easy to navigate. Part of what machine learning is, is understanding why and how a model was created, this by giving people a way to probe into models, they gain a better understanding on how the model helps them solve what problems they are facing *[Wexler et al. 2020]*.

These solutions are part of the number of solutions that exist today, and they have their various advantages. However, a common similarity most of them share, is that they are all focused on training and deploying models as efficiently and as quickly as possible. However, training a model efficiently is only part of the longer process of developing a computer vision AI application for enterprises. Oftentimes, applications are made up of a chain of models that each carry out a different task. These models are also often linked through implemented logic that tells them how to relate with each other to enable the computer vision application to function properly. In addition to this, these applications are often updated with new versions, and so there needs to be a way to monitor them while they have been deployed in production as well either on public cloud or on premises or at the edge. As we've seen earlier in section 2.4, it is only in approaching the project in its entirety can we focus on closing the loop and keeping the life-cycle going.

2.8 CONCLUSION

In this chapter, we took a journey into the world of AI, starting from the very beginning with a brief history on AI, to where we are at today with the role AI plays in today's industrial revolution, as more and more industries start to adopt AI to enable them reach and supersede their goals faster. We also explored what an AI system/application is, particularly a computer vision AI application, and the different stages that are entailed in creating a computer vision application of an enterprise client. Additionally, we explored the challenges that currently exist in the process of going through these stages/life-cycle, a major one of them being the lack of accessibility with AI systems, in that they're often too complex for non-expert users to use and leverage in their businesses. This led us to the concept of no-code/little code platforms, that are being implemented in different ways by a variety of companies, in order to enable enterprise clients to get started with creating their own machine learning and AI solutions with little to no background in programming. In exploring these existing solutions, we noticed that there are certain loopholes with the way they approach solving the current challenges. One being the main focus on data gathering, modeling, training and evaluating these models before deploying them in production. While this is useful, efficient and is a good start, we see that enterprise clients still struggle to understand exactly how all of the moving parts come together to enable them achieve their business goals, especially for the non-technical employees, and this is due to a couple of reasons:

One being, as we discussed earlier, the fact that the complexity of building an end-to-end computer vision or AI application spans beyond implementing, training and evaluating models. The chain of models need to be evaluated together to ensure that they all together work towards helping the enterprise client achieve their business goals.

Another point is that what we need is an intelligent system and end-to-end process that supports collaboration between human and machine, and doesn't solely focus on making the technology of the models work. There seems to be a need to look beyond the AI cycle of developing and deploying a model, in order to identify touch points that can support further end-user interactions, and render the experience easier to use and to follow by enterprise clients.

To tie this all back to our initial main research question we seek to answer with this thesis, based on the background research we have gathered, it seems that the right approach to designing such a system that is more accessible for carrying out the evaluation of computer vision systems, and the monitoring of the events they generate after they have been deployed, would need to account for a few factors:

- There is a need to step back and make the process more natural and more connected to how enterprise clients think about these solutions. Enterprises tend to think of the solution as one moving part and not necessarily a breakdown of individual models. The latter is more of a data scientist approach. Thus the system we design would need to be a system that allows for the evaluation of *whole applications and solutions*, and one that produces evaluation results that enterprise clients can easily understand, are able to easily connect back to their return of investments (ROIs) and key performance indicators (KPIs).
- Along the same vein, as enterprises go through the life-cycle of creating a computer vision solution, it is crucial that they understand how these applications perform after they have been deployed in production and are being utilized by real customers. Some employees would need to receive live events being generated by the solution in order to troubleshoot as quickly as possible to avoid business interruptions. Thus, it is important to have a way to view these incoming events in a way that is relevant to the company employee that will be reviewing them.
- A lot of the AI solution providers mentioned above in 2.5, are mainly focused on providing pre-trained models to users to leverage for their applications, and for the applications that aren't, they provide tools that enable the user build their own custom application but often requiring a very wide set of technical skill-sets such as data science and software development, in order to integrate everything together. It may be worthwhile exploring how these applications can themselves be created and put together, including custom models if clients would rather make their own, all with little to no code.
- Ultimately, the process that has to render more technical ideas easier to follow and implement with friendly UX, with little to no code, in order to appeal to non-expert users who work for these companies and are often overwhelmed. A lot of the solutions that exist today are still catered towards use by data-scientists, and are thus limiting for enterprises who don't have an allowance to high data-scientists or vendor companies. The solution and design we come up with, would have to be easy to use by different groups of people and job descriptions, so a majority of companies are able to leverage it.

These are just a few factors to keep in mind as we move towards more user research and ideation. It is relevant to keep in mind as well, that the solution to all the challenges we explored form a bigger picture to which these factors above are a part of. These factors have been scoped down to address the main research question posed. In the next section, we start off by exploring how to go about designing an evaluation system for computer vision applications, taking into account the learning and insights from this chapter.

3. APPLICATION EVALUATION - UX PROTOTYPE ITERATION

Designing an evaluation system for computer vision applications, is a process that takes into account a variety of factors. These factors are mainly centered around ensuring the evaluation system has human-centric functionality and that there's transparency and clarity for the user, with the user having the sense of being in control of the process. In this chapter, we work through the process it took to create a UX prototype of an evaluation system for computer vision applications, using the Smartcheckout application as the use-case being explored. We also look into the insights garnered from the prototype iterations and reviews of these different versions of the prototype. Here's a quick overview on the layout of this chapter and the different phases that were enacted in designing this feature:

Phase	**Task**	**Details**
Phase One	User Research and Ideation	Interviews (with Solution Architects) Personas User Journeys Ideation - Defining stakeholders, identifying the northstar vision, defining the day one experience for

		the user, and the non-day one experience.
Phase Two	First Design Iteration - Low-level Prototype and Review	Paper sketches that were created and then reviewed with the product team. Updating paper sketches with a Figma prototype.
Phase Three	Second Design Iteration - Mid-level Prototype and Study	Iteration updated incorporating feedback from the first iteration's review. Mid-level then created around three main tasks instead of how it was structured in Phase Two.. Tasks that the user would typically carry out. Official study carried out with a group of participants to test the mid-level prototype
Phase Four	Third Iteration	Updating the mid-level iteration to incorporate feedback from the study, and to leverage Deepomatic's design systems - fonts, colors and so forth.

Table 3.1 - Chapter Three Overview

3.1 USER RESEARCH AND IDEATION

The process of creating an application evaluation system started with first understanding who would be potentially using it, and what their user profiles are. This birthed the user research process by first and foremost exploring these user groups, represented by personas.

3.1.1 - Interviews

Personas are fictional characters that are designed based on preliminary research in order to represent or capture groups of users that may be using a product or service in a similar manner *[Chang et al. 2008]*. In creating these personas, we get a better understanding of who the users are, and it becomes easier to empathise with and design for them. We also get a chance to identify the general needs, pain-points, objectives, likes and dislikes of a group of user profiles, by using fictional personas to represent each group.

Once we have personas, it becomes a lot easier to ideate and make design decisions. In the case of designing an evaluation system for AI systems, it was particularly crucial to spend enough time working on developing these personas, as the field of industrialization AI is still quite new, as explored in Chapter 2, and the question of who carries out certain tasks along the way from the implementation to the deployment of the AI system, is one that is still in its infancy as it varies from one company to the other.

The first step to creating these personas was speaking with the employees at Deepomatic that currently go about running these evaluations on the AI systems before they are deployed. They are known as Solution Architects. The goal of these conversations was to get a better understanding on what the nature of their jobs entailed, how they currently go about evaluating these systems, what they struggle with, what works well, what they would have changed, and how they relate with the enterprise clients that need these systems built. In addition to the solution architects, other stakeholders such as the product manager who works at Deepomatic as well as an AI manager who works at one of the enterprise clients were interviewed.

Solution Architects

Before delving into how the interviews were carried out with the solution architects, it is important to understand what solution architecture is, and who solution architects are. Solution architecture is the process of defining, designing, and managing the engineering and development of a solution to specific business problems. Solution architects play a key role at Deepomatic, as they work with the enterprise clients to first understand their use case, and the business problem they are trying to solve. After understanding the business problem(s) in question, they are typically in charge of then leading the process of introducing, implementing and deploying the AI system created by Deepomatic, in order to solve these identified business problems. In order to gather user research insights from them on how they currently go about the work they do, particularly around implementing, evaluating and deploying AI systems, a round of qualitative user research was conducted, leveraging a style of interviewing known as *Narrative Interviews.* A narrative interview is one that strives to get insights on a user's specific experiences, by having them answer questions where they have to respond by narrating the events they have experienced in relation to the question. The questions they were asked included:

- *Could you tell me about the companies you currently manage, and what their business goals are?*
- *Tell me about the last time you made a change to an existing AI application, what was the change?*
- *Could you tell me about the last time you went about evaluating the app, I'd love for you to walk me through the process.*
    - *What were you looking to test/evaluate?*
    - Could you walk me through how the Python Script works? What evaluation rules are set in place?
    - *How are the events of the Groud-truth matched with events of the app, is this done in a standard way?*
- *Could you tell me about the very first time you went about running an app evaluation? How did you get started? What has changed over time?*
- *What metrics were most important to you and why? What does your client want to know?*
- *What aspects of this process do you handle, and what aspects does the employee from the enterprise client take charge of?*
- *What are the pain points you currently have with the overall process?*

These were the questions that were asked during the qualitative research, and overall, all three solution architects were interviewed, with the interview sessions recorded. The questions were asked with an intention to properly understand the work they do, what their main objectives and goals are, the stakeholders involved, how they interact with these stakeholders, how the current process works, what challenges they currently have, if there are current workarounds for these problems, how the process has evolved over time, what they enjoy most about their jobs and what they least enjoy as well, who they are as people and what their likes and dislikes are in general and what the northstar overall vision for the process is.

The narrative interview style was chosen because it is known to be more effective in gathering insights from users, than asking them direct questions, which users often answer with shorter responses *[Domnisoru, 2013]*. The idea behind narrative interviewing, is the belief that it is through stories that people interpret and understand the world they live in. Thus, people often express their experiences of the world through their interpretation and understanding of their realities. By opting for the narrative interview, we get to understand the user's subjective experience of the work, objectives and challenges that do and have, and before coming to better informed conclusions. The narrative interview in this case was also balanced with more specific questions after the user had had an opportunity to share their perspectives. This balance with more direct questions allowed us to get specific answers and data points from the interviews.

Product Managers and AI Managers

In addition to the solution architects, the product manager was also interviewed in a similar manner to better understand how he views the overall product and what the northstar/grand vision for the product and eco-system of their services is. This interview was crucial in order to ensure that the design approach was overall in line with the company's overall vision. The AI manager interview at one of their enterprise clients was also interviewed in a similar manner. The AI manager is the employee that often works directly with the solution architect, to enable the process of implementing, evaluating and deploying the AI application. It is important to note that this AI Manager's job title varies from company to company, and oftentimes, their technical skills also vary depending on the company and company structure. The product manager, having had more exposure to the different user groups over the past few years, also provided valuable information on the different types of user profiles that are part of the stakeholders, where each of them fit in the bigger picture, and what their pain-points, objectives and challenges are.

After carrying out these sessions of interviews with the solution architects, product manager and AI manager, here are some of the key insights that were garnered:

Deepomatic has two main use-cases they focus on in implementing computer vision applications. Namely - Augmented Workers and Augmented Customers.

> **Augmented Workers:** This use case is centered around workers that work on the field such as technicians that carry out fibre installations amongst other things, that have to carry out their tasks quickly while keeping to quality standards. The technician is able to ensure the quality of the work done, by using the computer vision application to take a picture of the work done, and then receives feedback from the application letting the technician know if the installation was properly done. It also provides a means for people behind the scenes, who are keeping track of the work that technicians do, to ensure quality control and to keep track of the installations being carried out.
>
> **Augmented Customer:** Deepomatic also makes computer vision solutions that provide customers with guidance and self services in a way that increases the quality of a customer's experience.

The *Automated Checkout Solution* is an example of the augmented customer solution, that enables people to automatically checkout there by freeing up cash registers and the need for long lines to check out.

Deepomatic focuses on creating computer vision applications and experiences around these use-cases and use-cases they may have in the future. These computer vision applications can be created in two ways, through writing code, or by creating them using the Deepomatic studio platform. Writing code is usually opted for when custom logic that dictates how models should interact with each other is needed. Oftentimes, the applications that Deepomatic builds are written in code by the solution architects, so that custom logic is defined that allows the models to interact with each other in a way that enables the AI computer vision application to resolve the enterprise client's problem.

At the moment, as explained, the solution architects are in charge of implementing these applications, however in order to scale up, part of the northstar vision of the company is to allow more and more enterprise clients to have the skills and ease of experience needed to implement these applications themselves. After an initial application is created, it is referred to as the first version of the application. New app versions can be easily created in the studio, by updating the model versions associated with an application. However, once custom logic of an application has been defined, it cannot be changed in subsequent application versions. If the custom logic needs to be changed, a new application will need to be created all together. Although, this is a rare case. After an application has been implemented, the next step is to evaluate it to ensure it works accordingly, before it is deployed. The evaluation is at the moment done by running python scripts on the application, and viewing the performance metric results afterwards. The goal of the evaluation is to see how well the application performs against a given ground-truth. This ground-truth is a JSON file usually. The performance metrics are also defined in the python script, and are calculated based on how well the application does. The photo and video files are also passed into this evaluation as part of the groundtruth in some cases. Creating a correct ground truth file, remains a challenge in a lot of use cases that need an alert raised at a particular point during a video that is part of a groundtruth. In the scenario where the AI application handles videos and not images, this process becomes complex and solution architects need to opt for manually running the application on individual videos, to see if they work as expected. This process ends up being long and tedious.

The performance metrics that are observed during these evaluations vary based on the use case, as well as the structure of the ground-truth file used. The evaluation process can take hours to complete, and oftentimes, the process breaks in between and it is difficult to know what point it was at in the evaluation process, and debug why it broke. Once the evaluation is complete, it is often challenging for them to quickly compare application versions to see right away what version works better and should thus be deployed. Overall, AI Managers often feel intimidated by the process of implementing, evaluating and deploying applications, because it is still quite programming based. They also struggle with understanding how well an application is performing in relation to their key performance indicators (KPIs) over time as their users interact with it.

In addition to this information above, a better understanding of their individual goals based on the use-cases and projects they're handling was gotten, in order to properly create the right user personas. In addition to carry out qualitative research, time was also spent reading up on the enterprise clients, and on each of these stakeholders' profiles.

3.1.2 - Personas

After carrying out all of this user research, it was understood that different use-cases and solutions have different sets of personas and in some cases, some use-cases share similar personas. There are also personas that are seen in all of Deepomatic's use-cases. Here are the personas that were created to guide the design ideation and implementation process of the application evaluation feature.

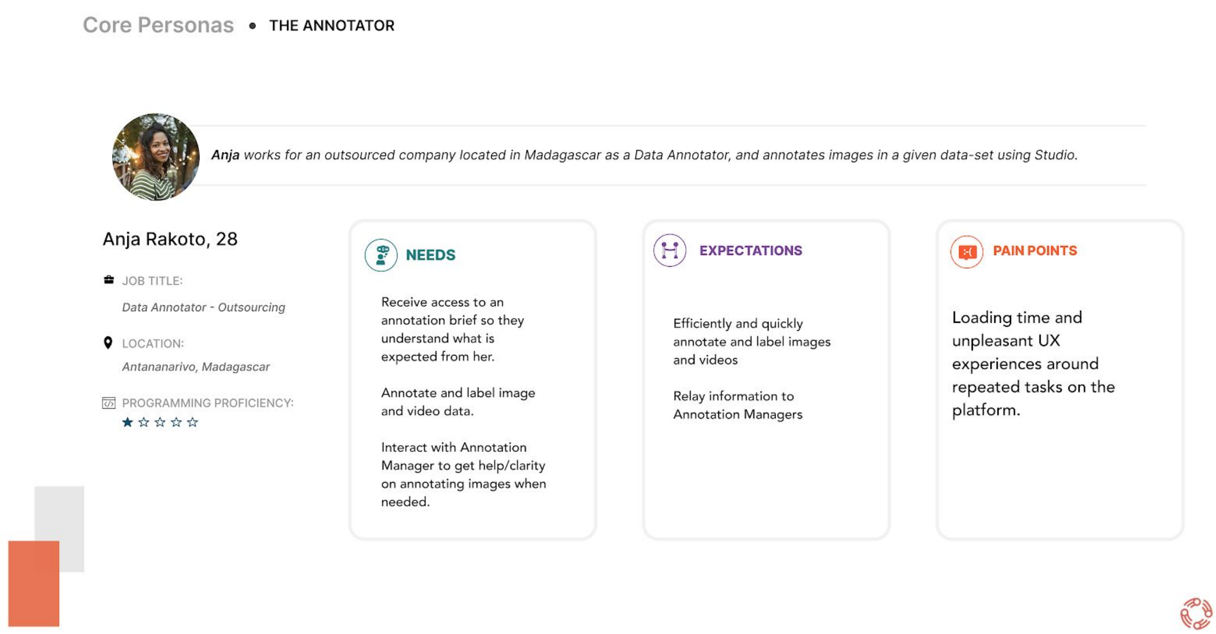


Figure 3.1 - The Annotator Persona

The annotator, as seen in Figure 3.1, is one of core personas that are applicable to all of Deepomatic's use-cases. We see that in a typical persona profile, such as Figure 3.1, the needs, the expectations and the pain points of the persona have been clearly defined, as well as other factors that are important to know for the context of the features being built, such as the *programming proficiency*. Other core personas are illustrated in the images below:

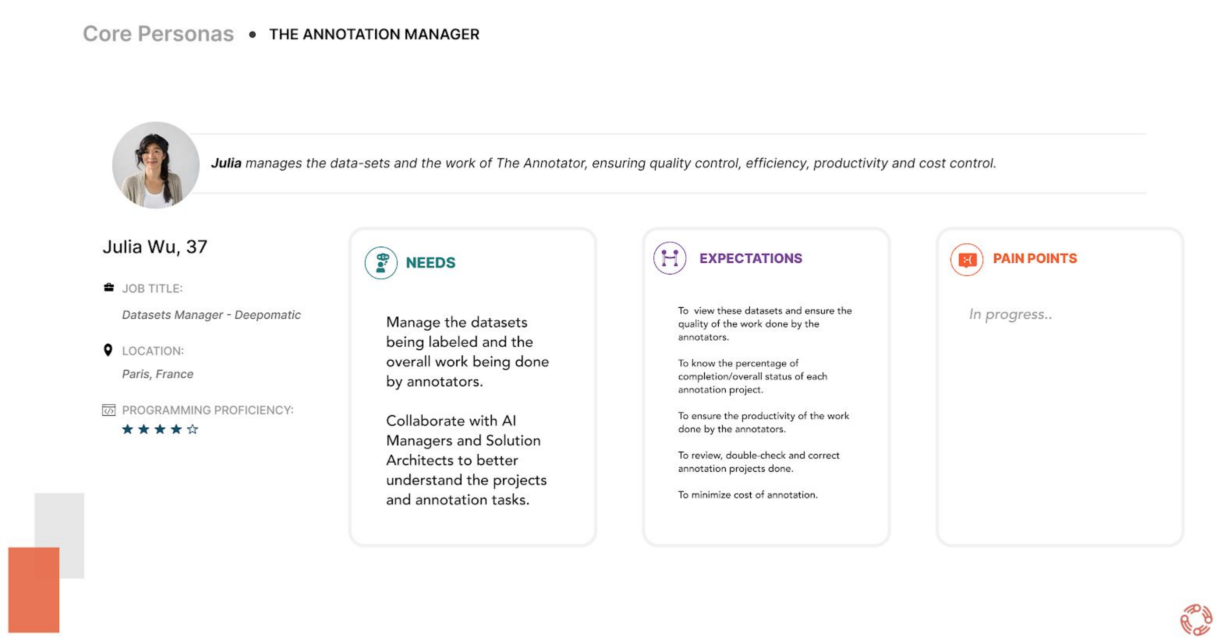


Figure 3.2 - The Annotator Manager Persona

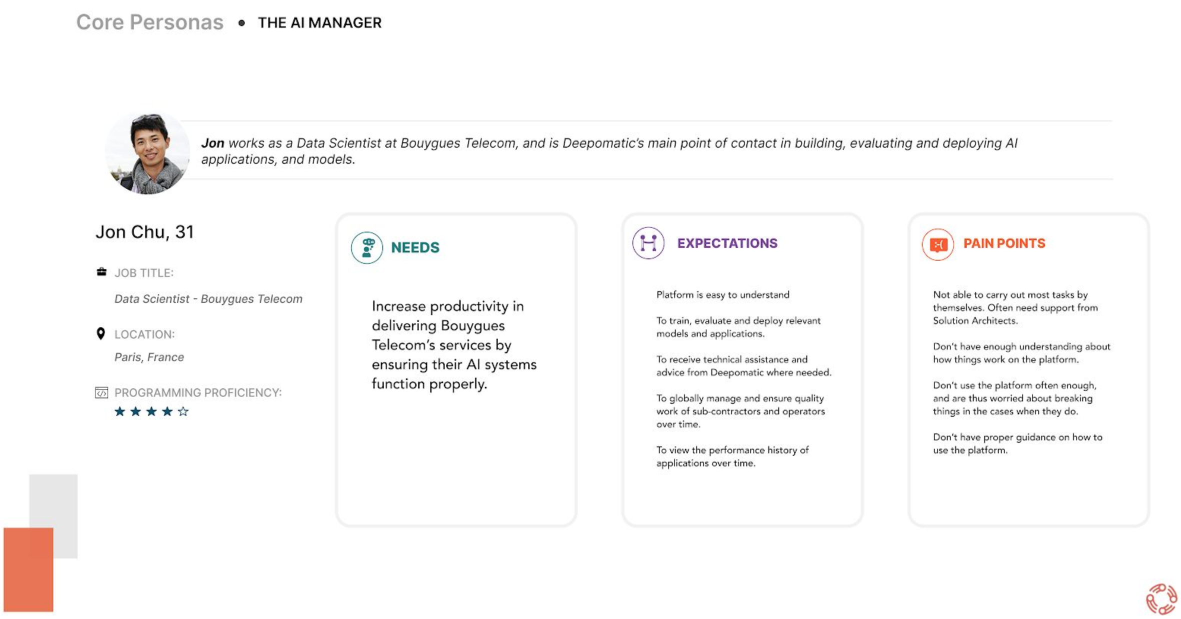


Figure 3.3 - The AI Manager Persona

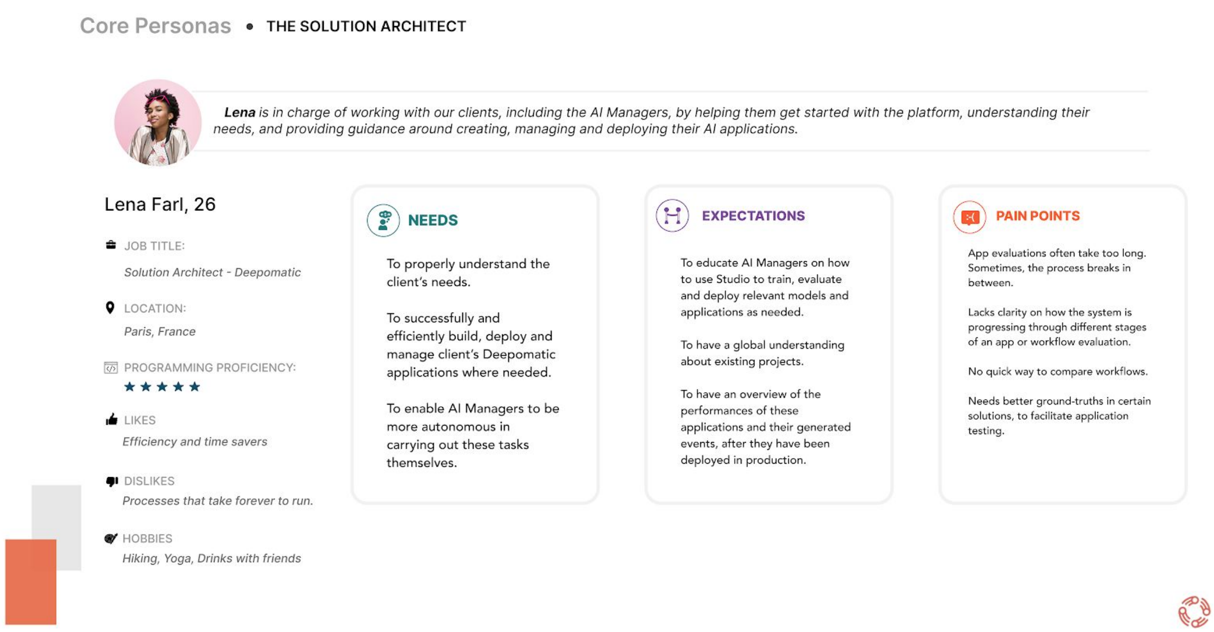


Figure 3.4 - The Solution Architect Persona

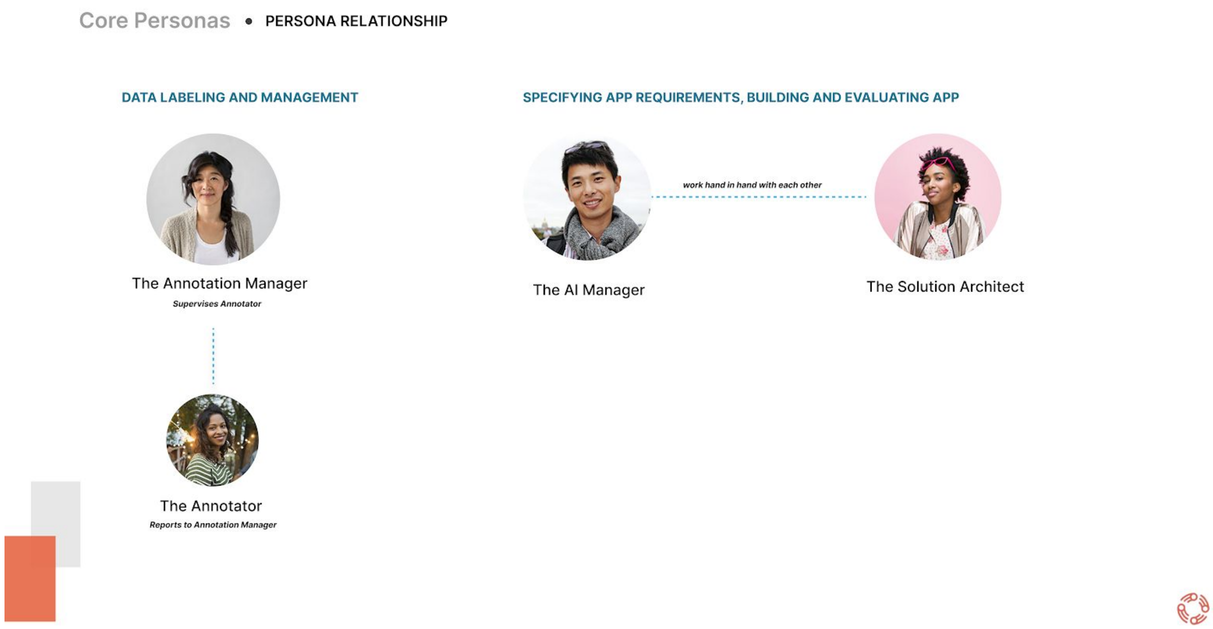


Figure 3.5 - Core Personas Relationship

In Figure 3.5, we see how these personas/stakeholders work together to achieve different goals. Each of these goals leverages a different feature in Deepomatic's studio.

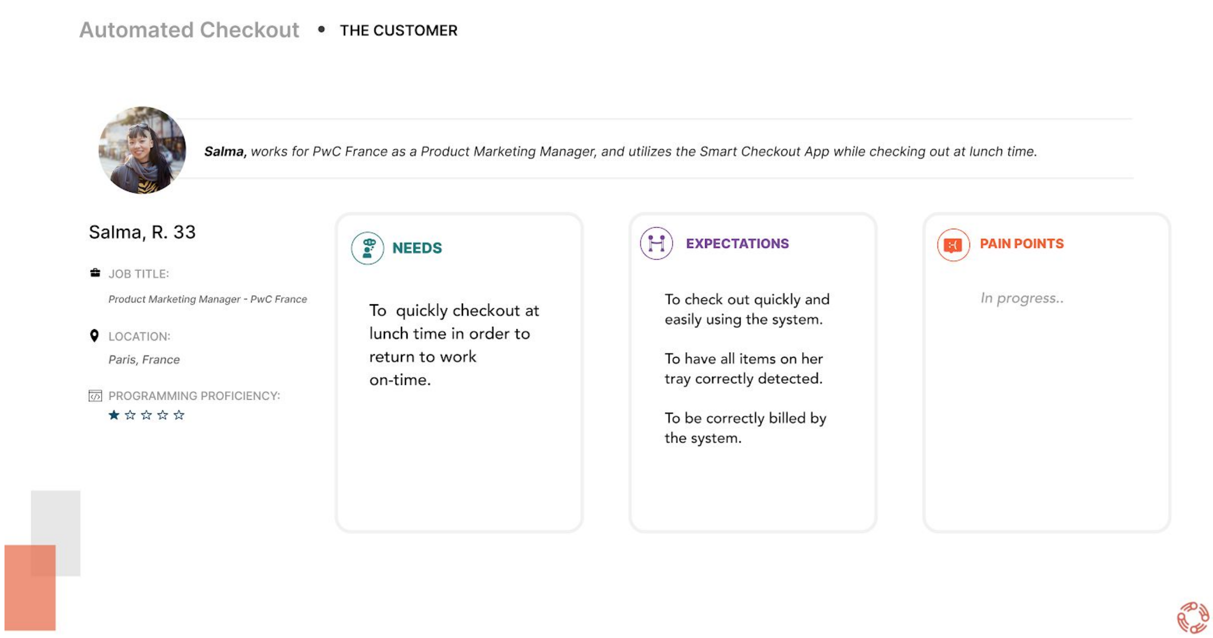


Figure 3.6 - The Customer Persona

Lastly here, we see another persona in Figure 3.6 that is unique to the Automated Checkout scenario, Salma, who would be a typical user of an Automated Checkout solution. The Automated Checkout solution was the primary use-case used in designing the application evaluation feature. Consequently, it was designed around the AI Manager persona and the Solution Architect persona.

It is important to note that these personas are ever evolving, as these use-cases are further defined, and more research is done into each use-case and the stakeholders involved.

3.1.3 - User Journeys

Part of the User Research process included creating user journeys in order to once again guide ideation and design decisions. User journeys are created to illustrate the path a user/a persona may take to reach a particular goal while using a product or service. In this user journey, each step the user would make is highlighted, as well as what their mindset may be at each point, what emotions they may feel, and what challenges they may acquire. There are a number of advantages that come with creating user journeys, such as making it easier to build empathy for the users, to identify loopholes and current gaps, to predict the user behavior, and so forth. For both the AI Manager and Solution Architect personas, a user journey was mapped out as a guide as shown in Figure below

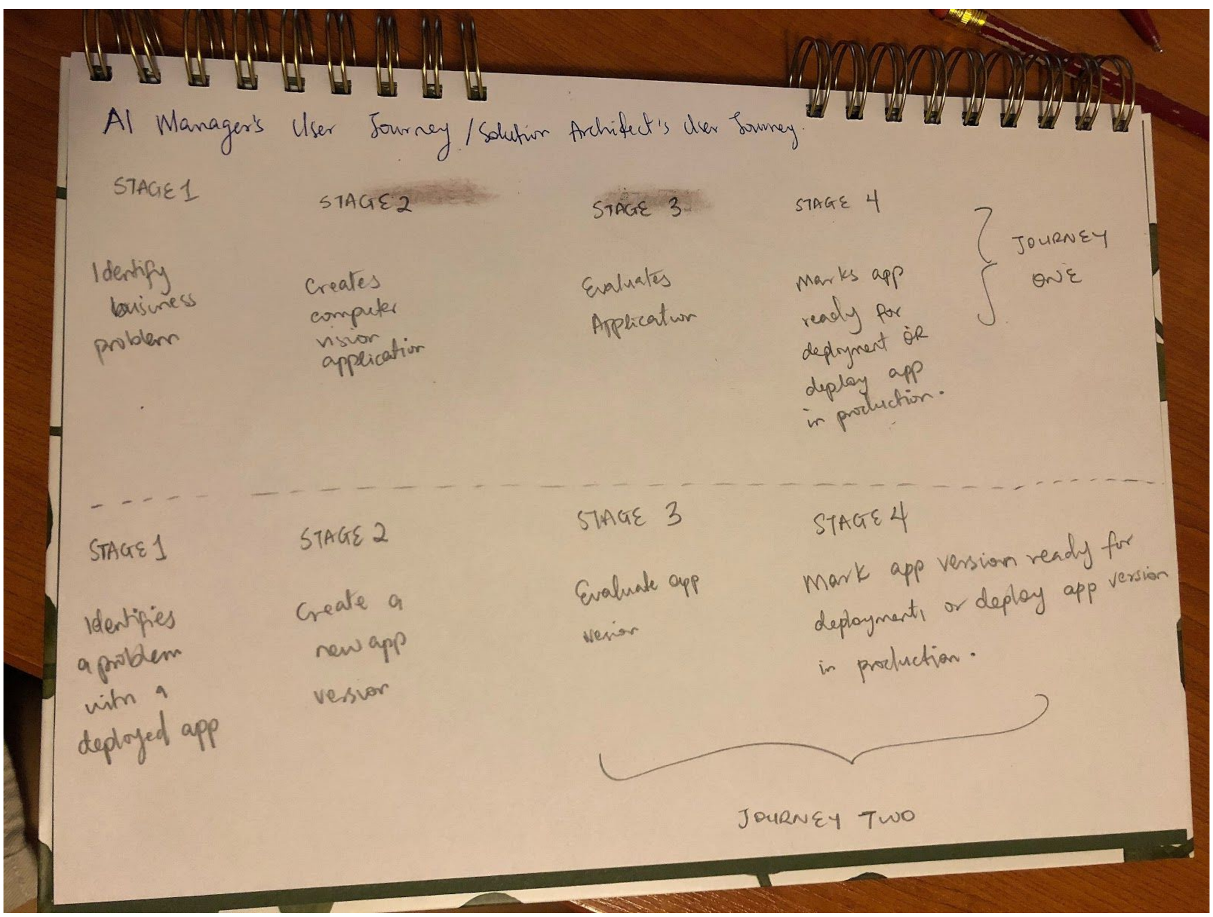


Figure 3.7 - The Solution Architect/AI Manager's User Journey

3.1.4 - Ideation

The first part of the ideation process was to keep in mind who the end-user of the application evaluation system would be, out of all of the personas/stakeholders we identified. The main user personas that will be interacting with the application evaluation experience are namely the *solution architect*, and the *AI Manager.* The northstar vision of the feature was also taken into account, to create a short term experience and a long term one. Short term, we decided that the experience would be focused more primarily by the solution architect in conjunction with the AI manager who may or may not be proficient and programming. Long term however, the northstar goal is to make the experience completely usable by only the AI manager with minimal help from the solution architect, thereby making it easier to scale up.

The second part of this process was breaking down the different entry points into the experience. A day one experience where the user is creating an application for the very first time, and a non-day one experience, where the user already has an application created. Depending on the entry point, the flow of the experience for the user is highlighted, guided by the user journey map for the stakeholders in question for the application evaluation feature, being the solution architect and the AI manager.

After the flow and different routes each persona could take had been identified, the experience's structure was broken down as follows.
Day One Experience

For the day one experience, the solution architect would be the primary persona as they would go about setting up/implementing the application, as well as carrying out the evaluation of the application, which are both major aspects of the overall flow. The initial flow for this process included:

- Creating the evaluation by defining the name of the evaluation and what the intent is for creating it.
- Then selecting the application(s) that need to be evaluated by the evaluation. This application can be selected from an already existing application that was created in the Deepomatic platform, or by importing one that was implemented with code, usually in the form of python and yaml files.
- After importing or selecting the application, the solution architect then goes ahead to select the ground truth that would be used for the evaluation.
- Then they identify how the event objects predicted by the application should be paired with the event objects in the groundtruth.
- The Solution Architect then sets up the key performance indicators that will be used to evaluate how well the application is performing.
- Once all of this is set up, the user is then able to run the evaluation, view the results and deploy the application should they choose to.

This was the very first user flow that was structured out and used to kick off the designing and sketching process. The next step was brainstorming on how all of this would be represented visually. Over time, this user flow shifted, as we gained a better understanding of how the personas would interact with the Deepomatic studio through studies we carried out that will be discussed later, and in what situations they would need to create new applications and evaluate them as a day one experience, and also what situations they would need to carry out a non-day one experience. It is also important to note that these prototype versions were created with a use-case primarily in mind - the Automated Checkout solution. We decided to go about it this way so we could finetune the experience for one use-case that is largely used, and ensure it works for it before expanding and adopting the user experience and design to other use-cases. With the Automated Checkout solution, the user also only works with photo data and not video data, and thus this helps simplify certain aspects of the prototype. However, other use-cases were slightly designed for simultaneously, as some of their requirements were taken into account, but more so as secondary solutions.

In the later versions of this prototype, we ultimately arrived at a design and experience that was better fitting for the user's key scenarios with creating computer vision applications and evaluating them, and that flowed better with all parts of the scenario. The process of ideating involved journaling and illustrating the user flow, as well as drawing out initial paper sketches and highlighting key experiences that need to be incorporated in the design along the way, based on the user research insights that were gathered. Part of the ideation process was also exploring where this experience would live in the current Deepomatic platform, that would be natural for the user and inline with how they currently use the platform.

In the next sections, we break this ideation process down and explore the design iterations, and design decisions that were made at each step of the way.

3.2 FIRST ITERATION - LOW LEVEL PROTOTYPE AND REVIEW

The first iteration of the application evaluation was created as a mockup version of a standalone feature in Deepomatic studio, and not necessarily as a continuation of the current processes of creating an application on the Deepomatic studio. First, sketches were made on paper and reviewed with the product manager as well as other members of the product team. As explained in the section above, this version of the prototype was created based on the initial draft of the day one experience detailed in the section above, in order to get the ball rolling.

After making paper sketches, and the low-level version was established, the design was migrated to Figma, which is a UX/UI Design and prototyping tool. Here are some parts of the low-level design, that illustrated the initial way we thought about the user flow (some of these include the paper sketch versions):

Landing Page

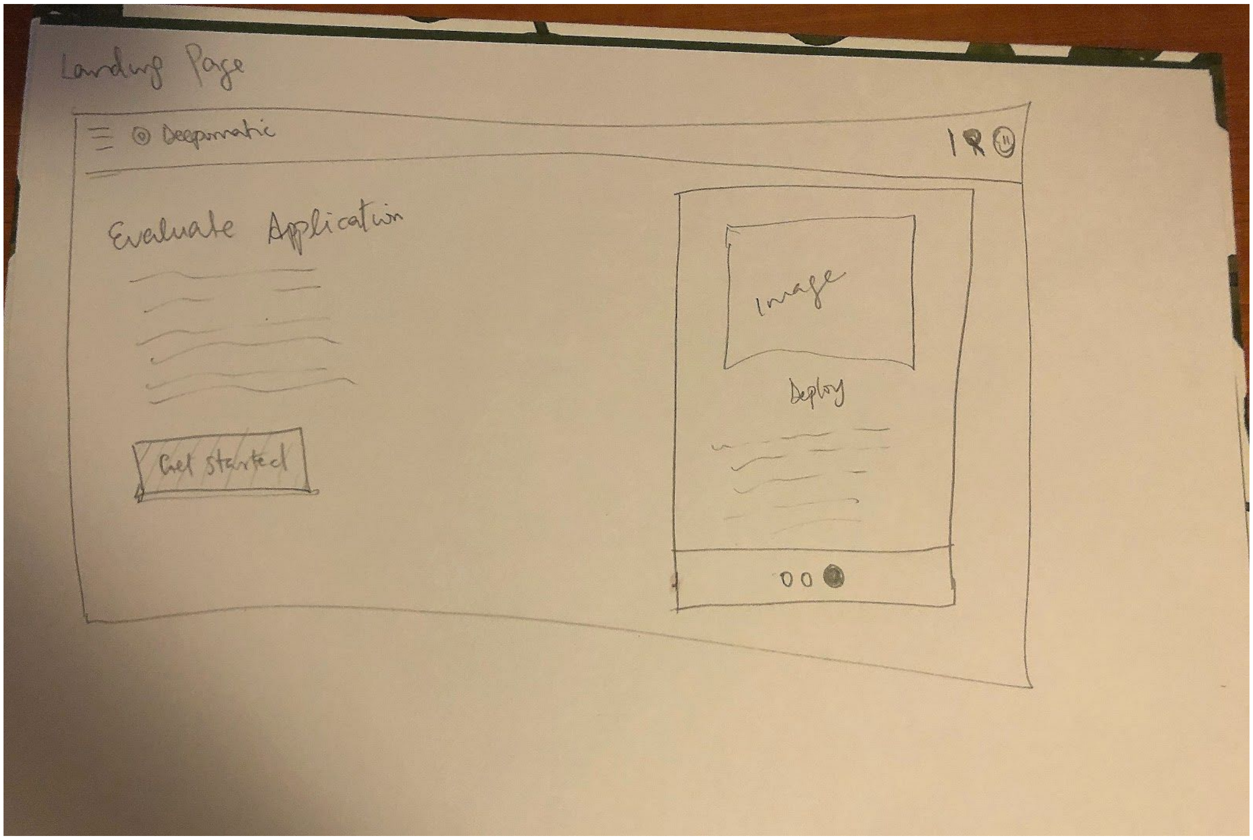


Figure 3.8 - Paper sketch of Landing Page

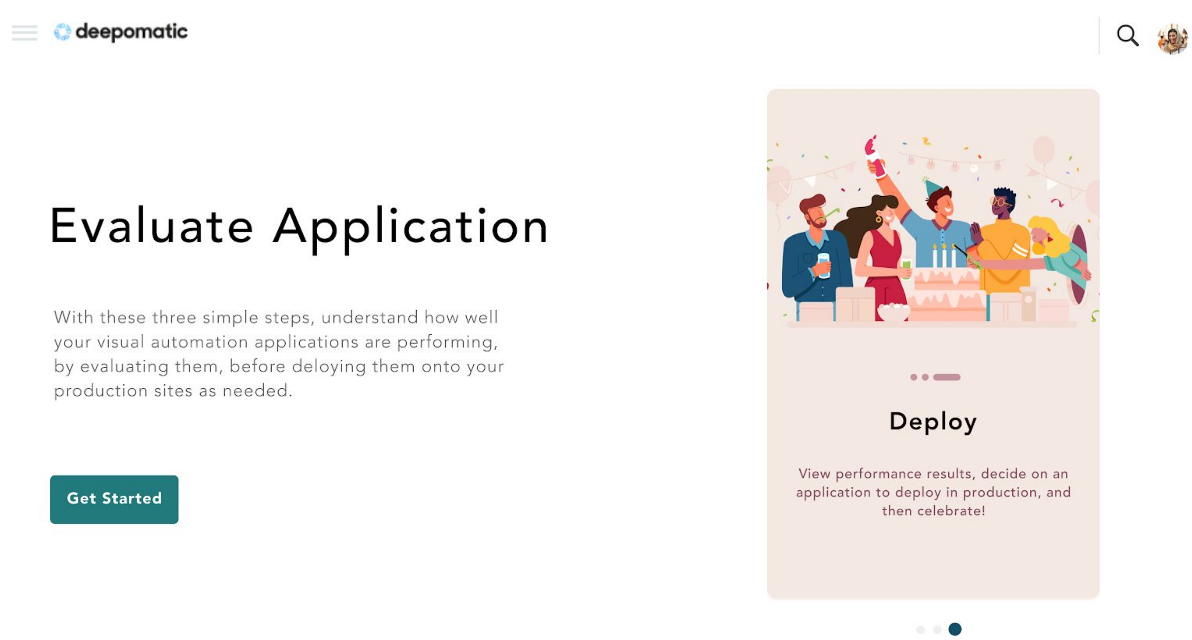


Figure 3.9 - Application Evaluation Landing Page

This landing page was created to be informative and playful. As a result of the UX research carried out, we aimed to go for a design that was easy to understand by AI Managers that aren't tech savvy and aren't as familiar with the processing of evaluating an application and deploying it. Playful and inviting illustrations were opted for to create this ambience. The process of the evaluation was also compressed into three major steps - Set Up, Run and Deploy, to make a process that is quite complex, easier to follow and implement.

Setup Evaluation

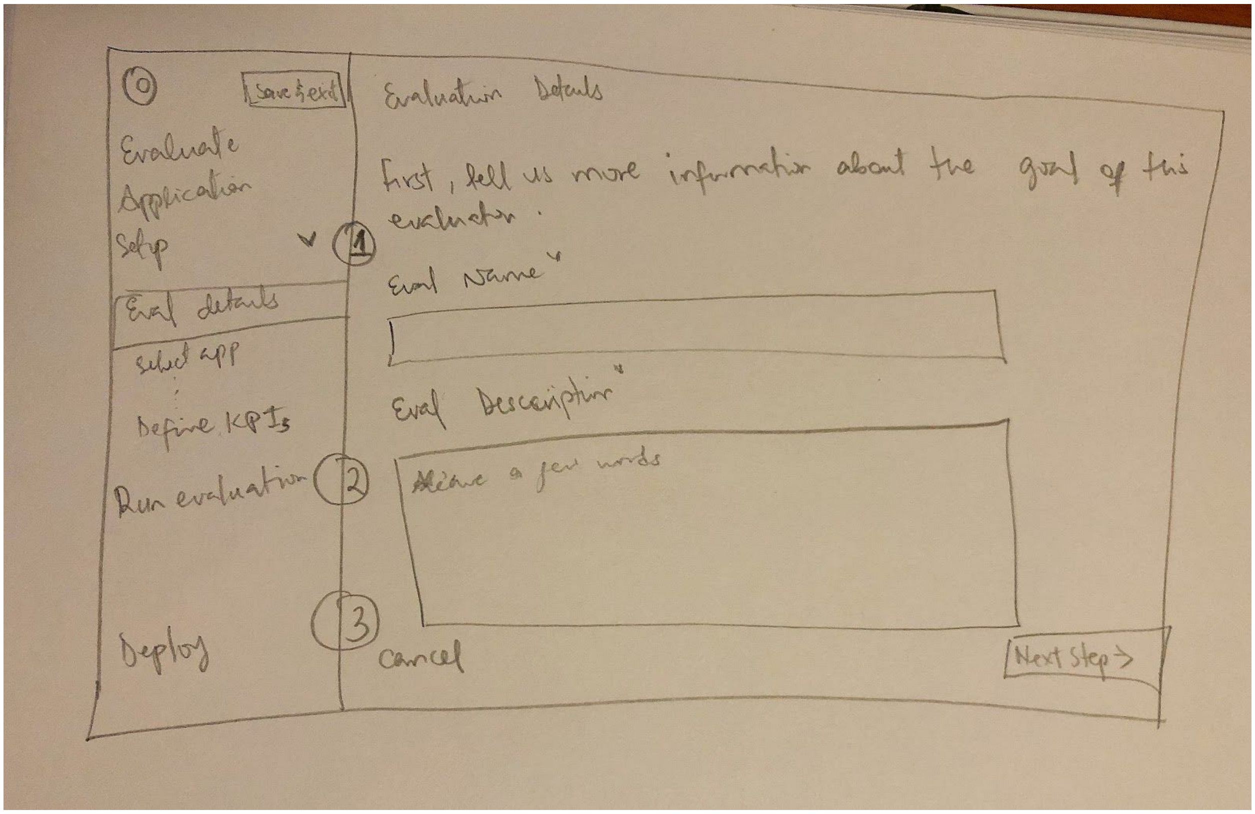


Figure 4.0 - Set up evaluation Paper Sketch

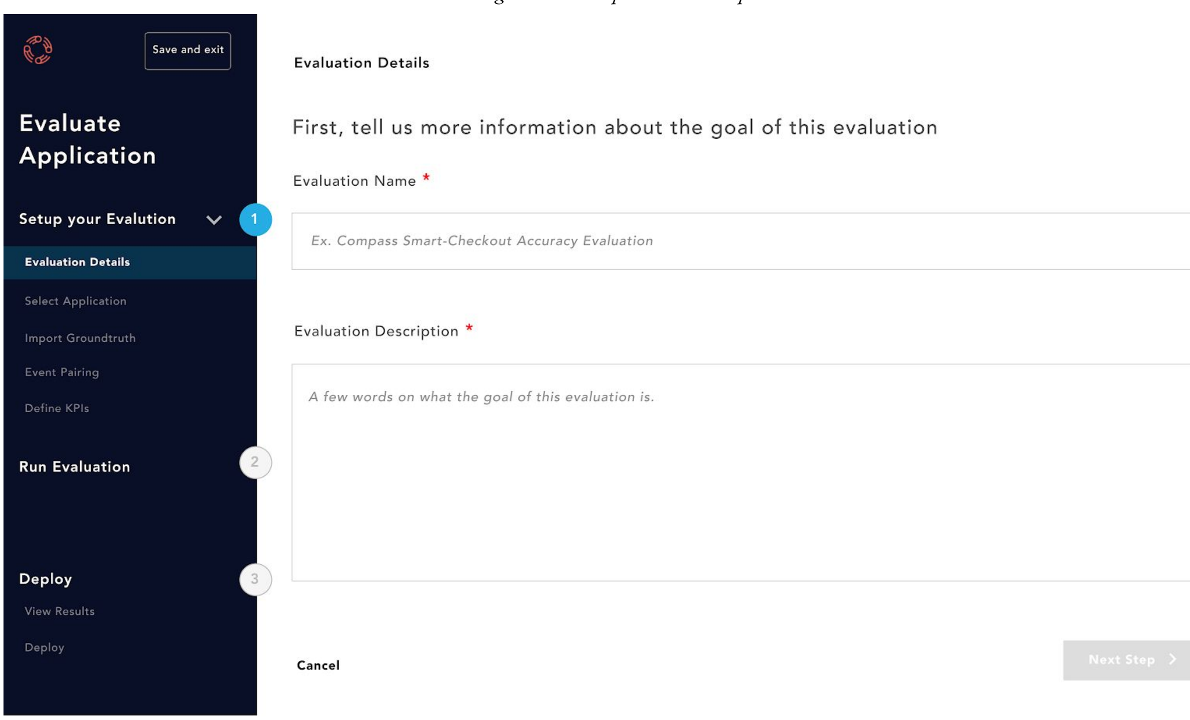


Figure 4.1 - Setting up Evaluation

The first step of the process here is setting up the evaluation that will be used to evaluate a given app. Here the user is able to input the evaluation name and description, to refer to as well at a later time. To the left of the design in *Figure 4.1,* we see that the steps of this process are clearly stated - Setup your Evaluation, Run Evaluation and Deploy. Each major step that were highlighted in the Landing Page, are now highlighted on this page with the sub-tasks as well.

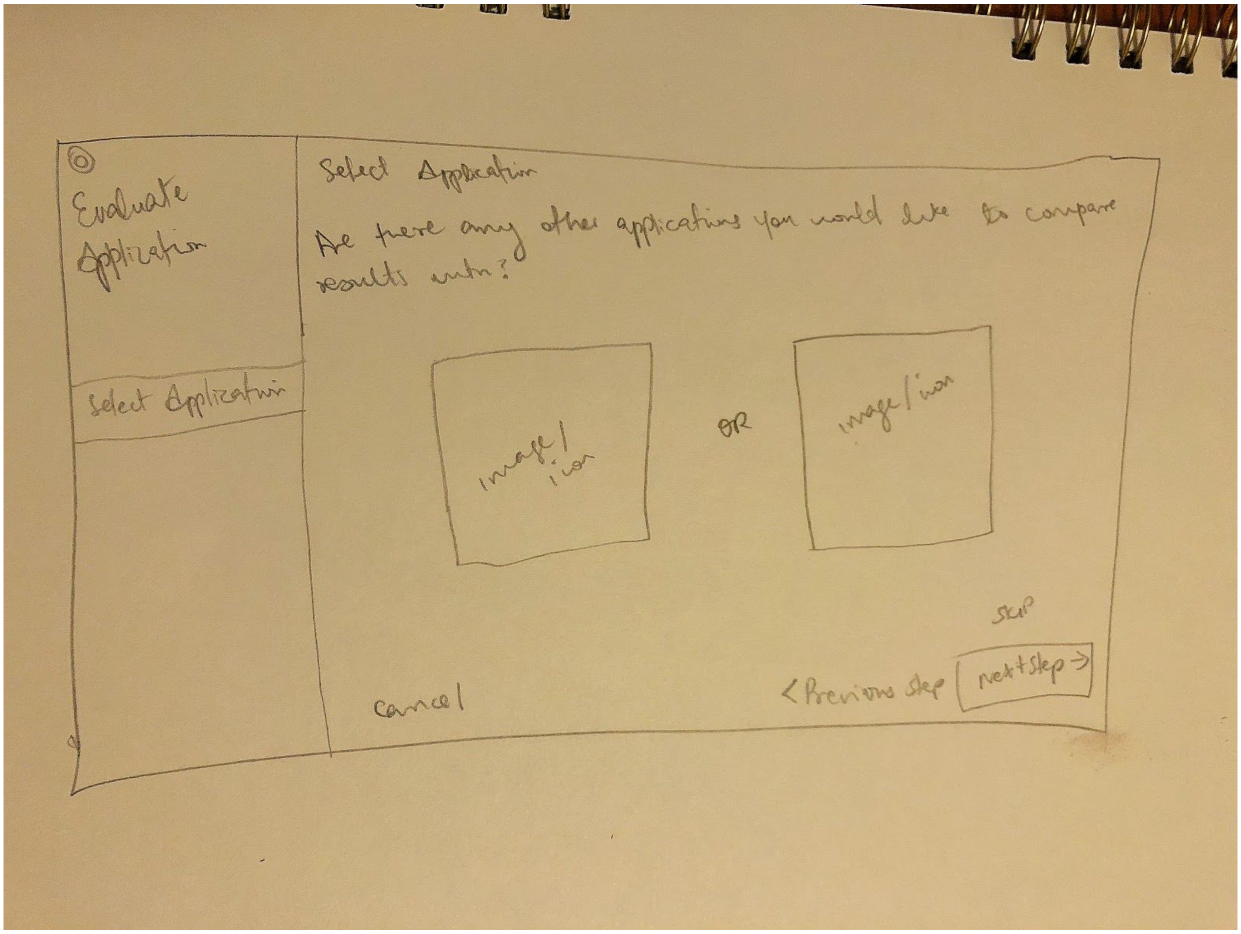


Figure 4.2 - Selecting Application Paper Sketch

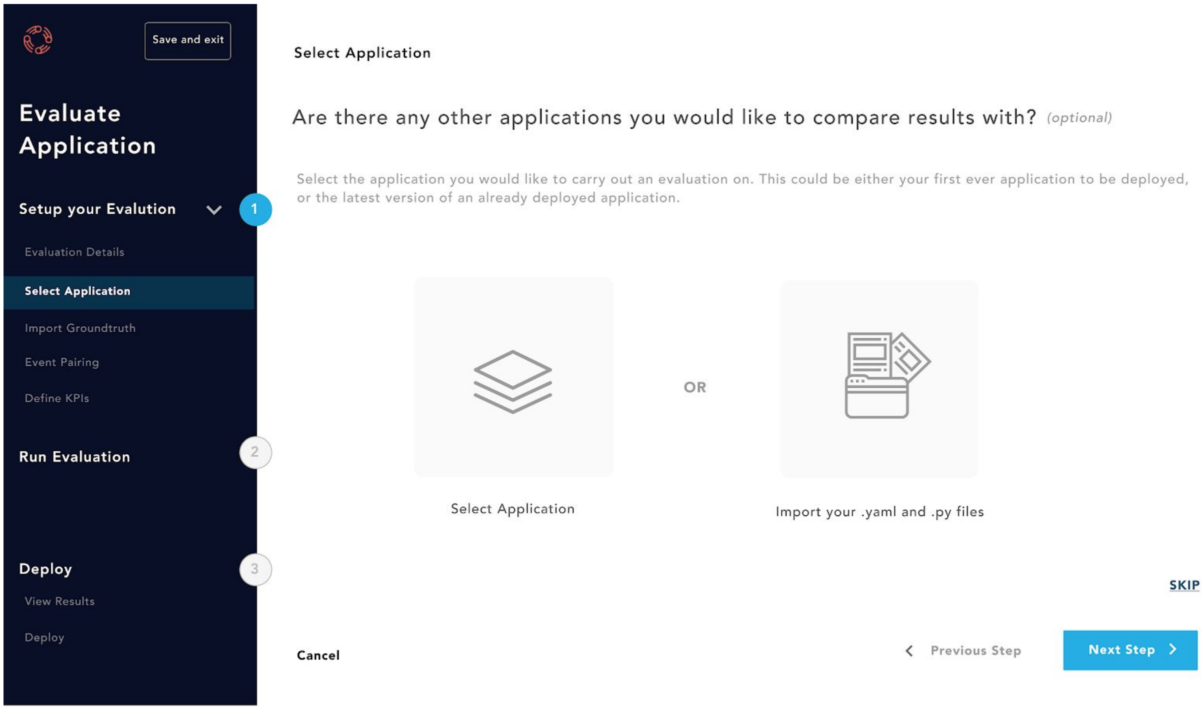


Figure 4.3 - Selecting Application

After providing the evaluation details, the user then goes ahead to select the application that they would like evaluated. This can be done by selecting an application on the Deepomatic studio, or by importing .yaml or .py files if the application was implemented with code. This is also designed here to be an optional step, in the case where the user would rather just focus on setting up an evaluation, and running it on the applications at a later time.

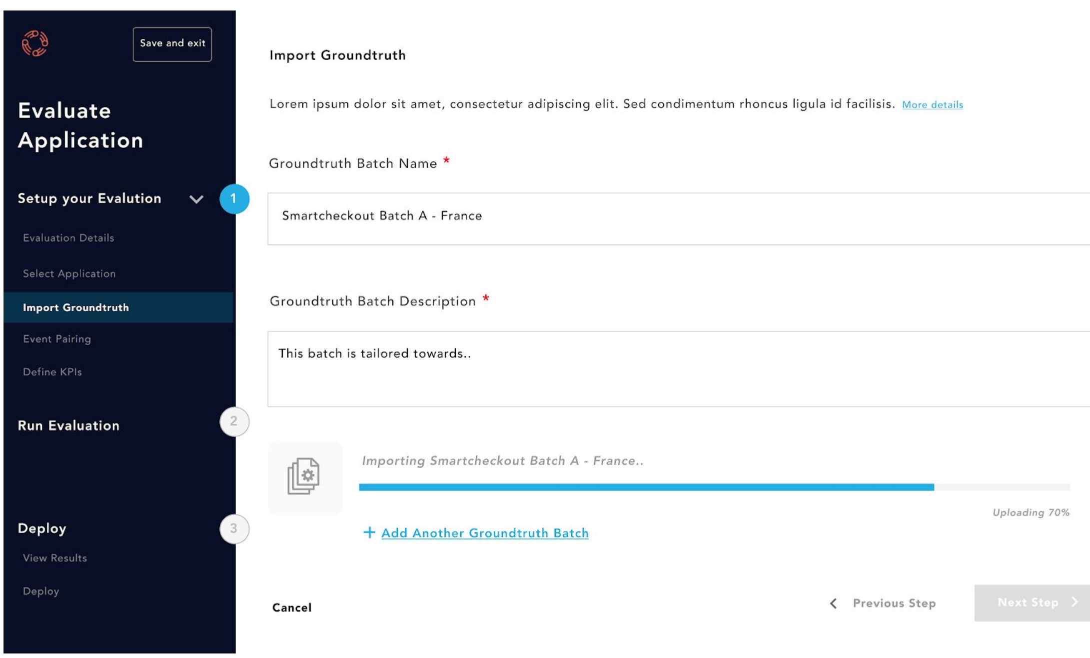


Figure 4.4 - Importing Groundtruth

The user then goes ahead to import the groundtruth, which is a file that will be used to cross check the events that the application generates and predicts, when it is being evaluated. Afterwards, the next step in the process is *Event Pairing*, which allows the user to define how they'd like the system to pair events from the ground truth to the events predicted by the application.

Define KPIs

After the user has defined the system of event pairing, the user then defines what key performance indicators (KPIs) would need to be tracked and reported in order to understand how well the application is working. Before working through this experience, the solution architect / AI manager knows what KPIs they would like to implement. Once they've defined these and how they can be calculated, they then define the KPIs on the studio platform as shown below:

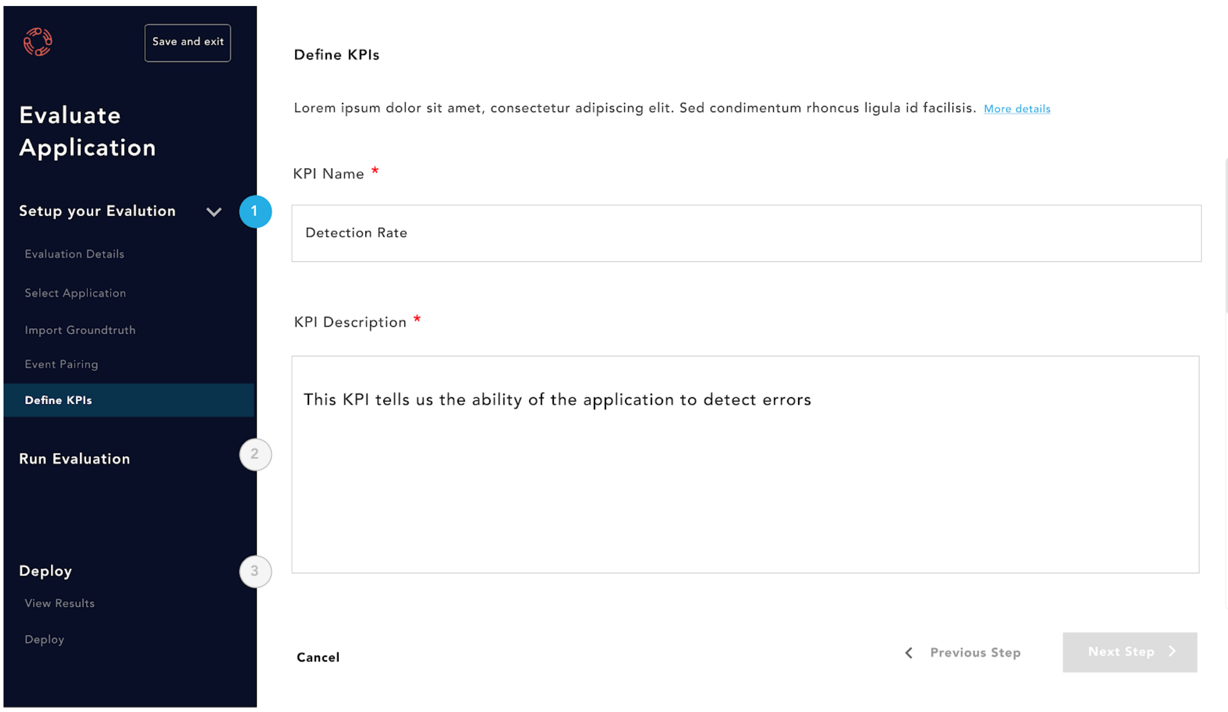


Figure 4.5 - Defining KPIs, KPI Details

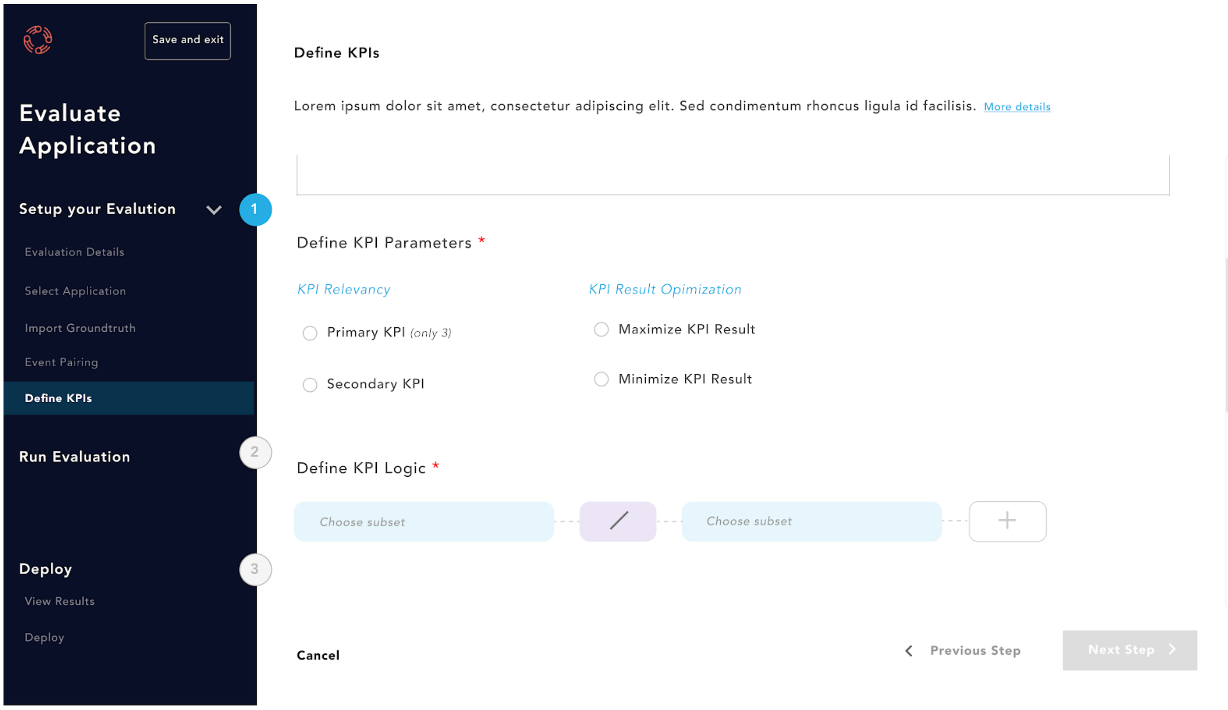


Figure 4.6 - Defining KPIs, KPI Formula

The process of defining a KPI is one that is quite technical as it entails writing lines of code typically that implement mathematical formulas. Thus the goal was to simplify the process as much as possible with friendly UX and steps that were readable and easy to understand. As shown in *Figure 4.5 and 4.6,* the user is able to define the name and description of a KPI, and then determine if it's a primary or secondary KPI in order to determine how the results page would display the KPI. The user is also able to specify here how they would like for the KPI to be optimized. Maximizing means that the higher the KPI is, the better the application is performing, while inversely, minimizing the KPI result means the lower the result, the better the application is.

Another crucial part of defining a KPI is stating the KPI logic, which is a mathematical formula formed by sets of events, that tells the system how to calculate a given KPI, based on the set of events in the groundtruth, the set of events generated and predicted by the application, and successful matches and pairs made between these event sets. For example, we see that the *Detection Rate* KPI's formula, has been constructed as shown in *Figure 4.7.*

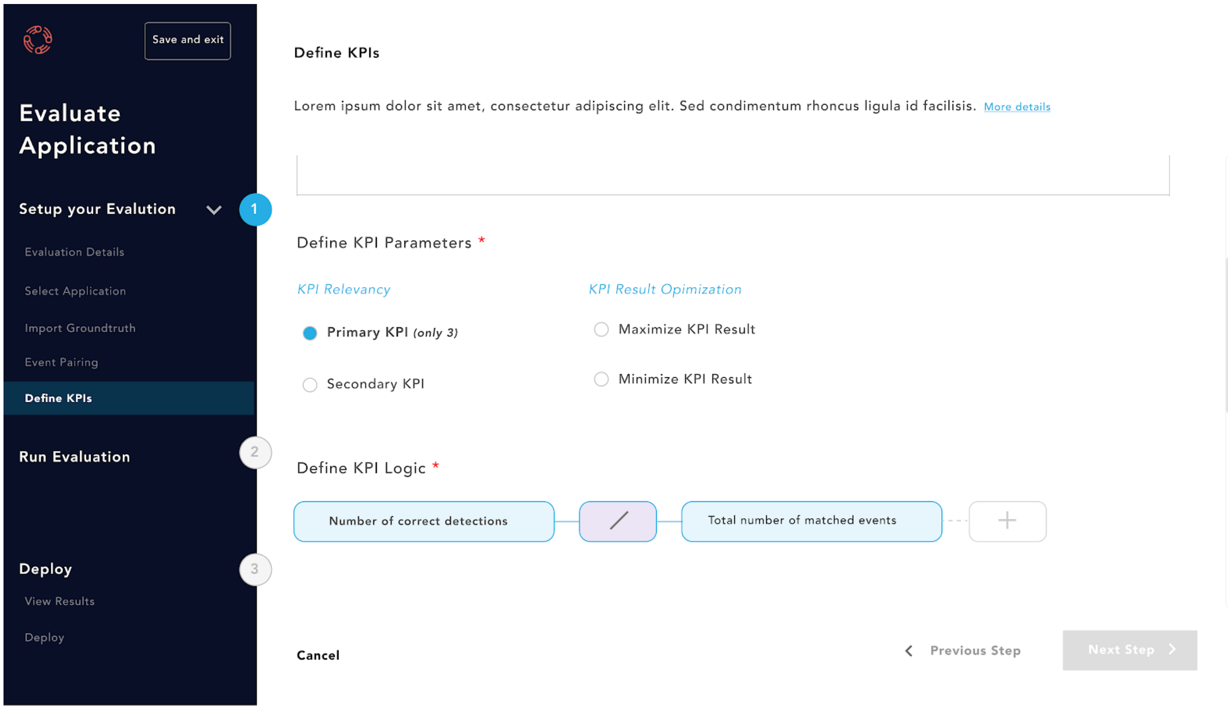


Figure 4.7 - Defining KPIs, KPI Formula

We see that the formula is surrounded by a blue line, to show that the formula is one that is mathematically sound. This is checked technically by the platform that will be programmed to know what makes a valid mathematical formula. In this case, the Detection Rate KPI, will be calculated during an evaluation as the *Number of correct detections / the Total number of matched events.* The idea behind creating the numerator and denominator for this KPI logic, allows the user to create logic that also informs the system on how to create these sets. This means that in order to create this formula, the user will first have to click on the blue bar, and then curate a subset logic *(as shown in Figure 4.8 and 4.9),* that filters through event sets to allow the system successfully gather the number of correct detections. In Figure 4.8, we see that a placeholder text has been placed under the "Choose a Subset" title, however, this was updated in later versions of the prototype, with text that informs the user on how to interact with the page, and guides them through the process.

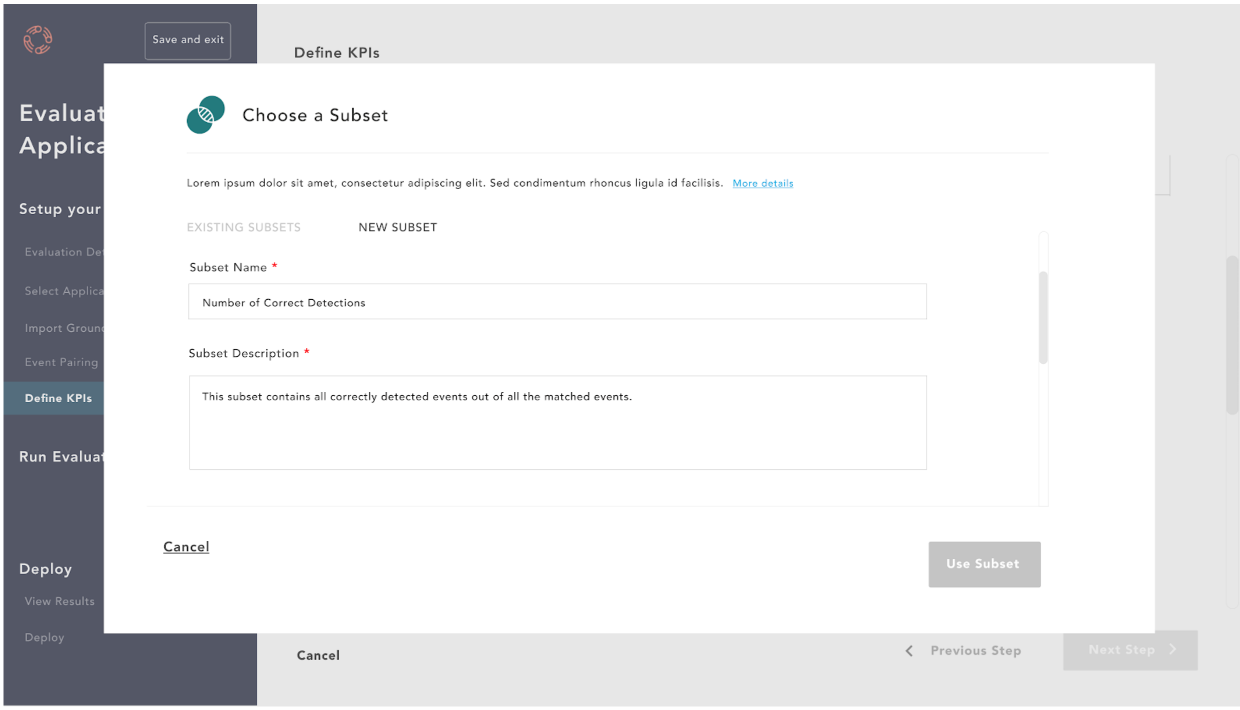


Figure 4.8 - Choosing a Subset in creating a KPI

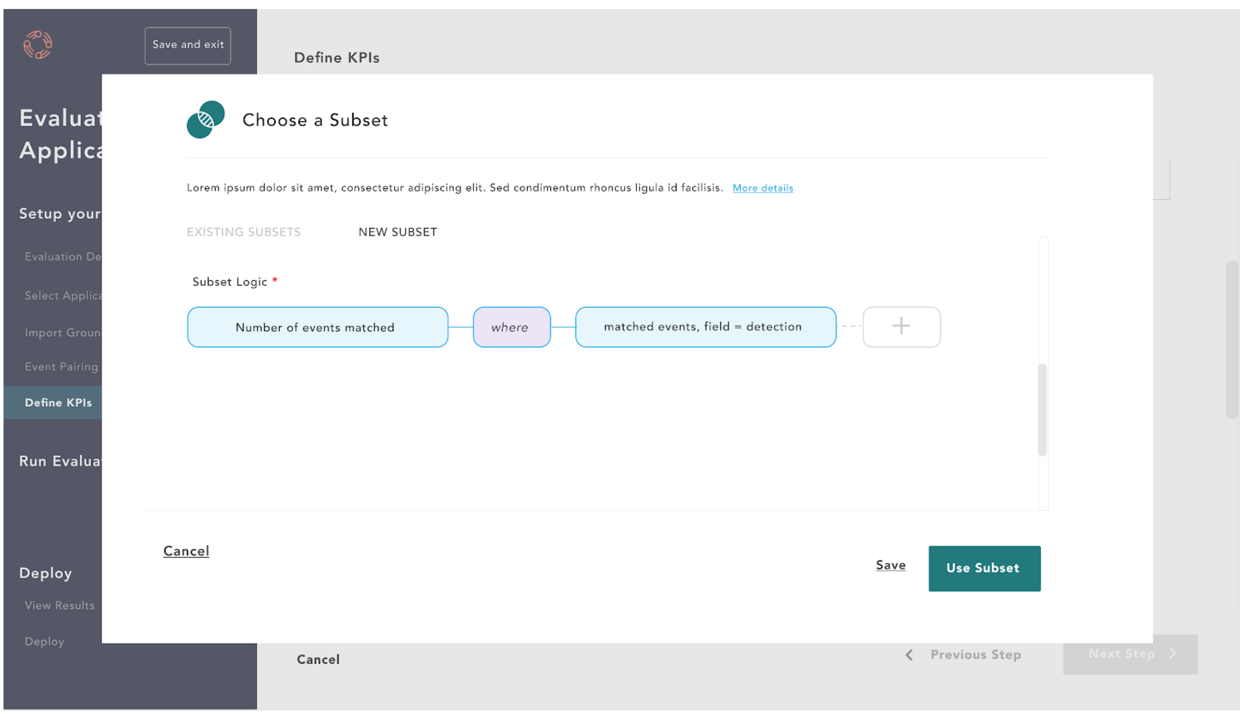


Figure 4.9 - Choosing a Subset in creating a KPI

We see that the numerator here - *Number of Correct Detections*, is created by defining a set logic that the system uses to deduce the right number of detections that have been correctly made. The logic here, as shown in *Figure 4.9*, defines the number of correct detections as the total number of matched events (matched events between the ground-truth and the events predicted by the application), where the paired event is labeled as a detection. The idea is that once a subset has been created, the subset in this case being - the number of correct detections, it can be used in other KPI formulas, and would then appear under “Existing Subsets”. Once the KPIs have been defined, the evaluation is considered set up.

Run Evaluation

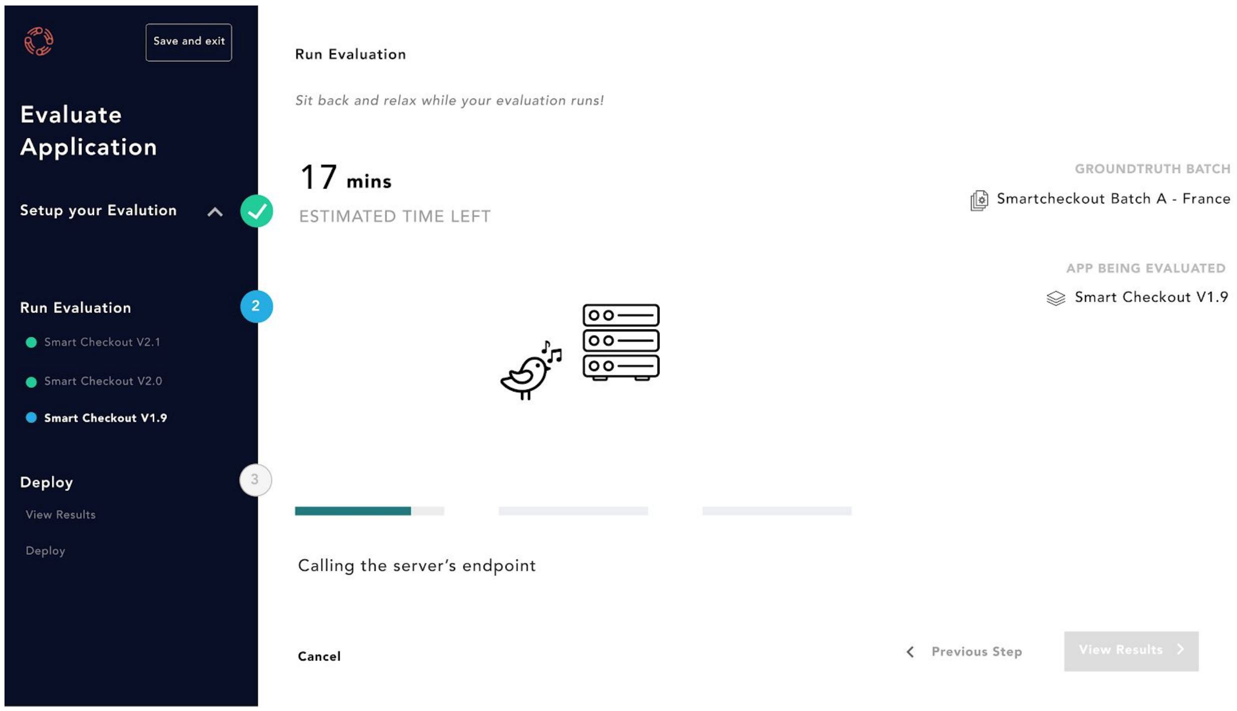


Figure 5.0 - Running the evaluation

After the setup is completed, the next step in the evaluation process is running the evaluation that has been set up on the application. In *Figure 5.0* above, we see the evaluation process in this instance is done on three different versions of the same application at once. The purpose of having it this way, is to allow for comparisons of final results before choosing what version gets deployed. This page was designed to make the evaluation process easy to understand and to follow. A big issue that was raised during user research was that the process takes long to run, can be vague and sometimes breaks in the middle. Thus, by showing what step the user is at, the time left, the ground truth batch being used and what application version is being evaluated, the user is able to have a clear grasp on the evaluation process.

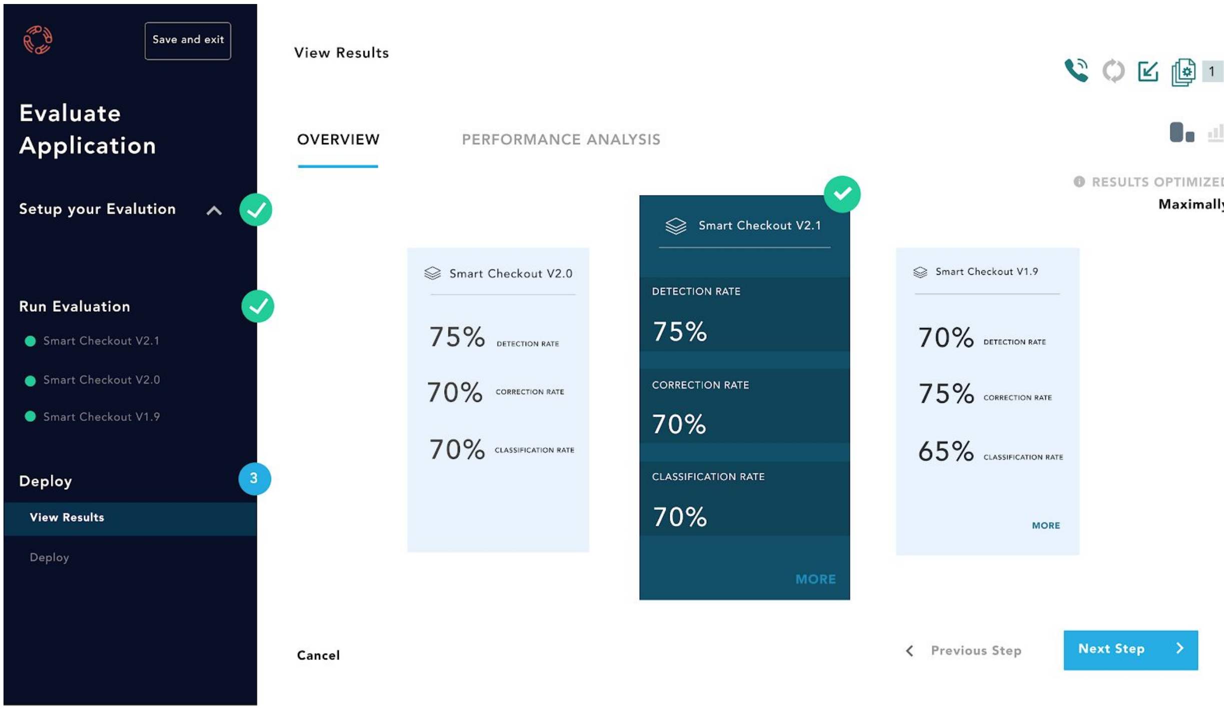


Figure 5.1 - Viewing Evaluation Results

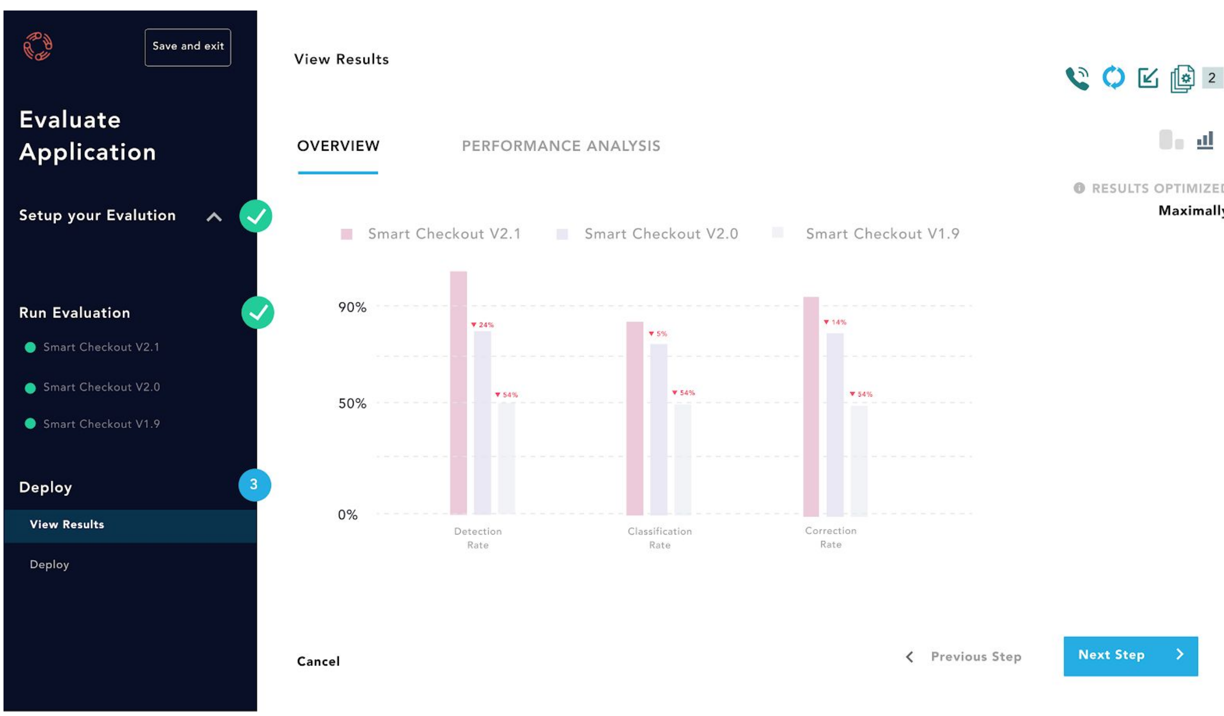


Figure 5.2 - Viewing Evaluation Results

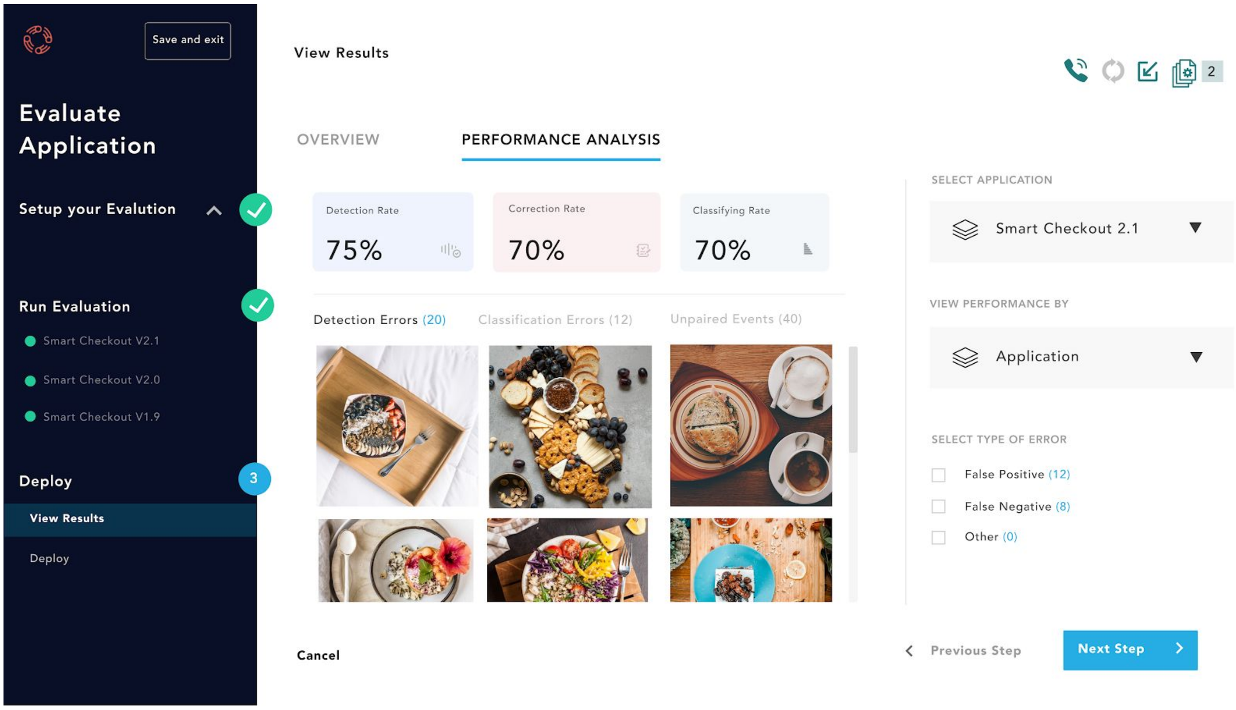


Figure 5.3 - Viewing Evaluation Results

After the evaluation is run successfully, as seen in Figure 5.1 and 5.2 above, the user is then able to view the results of this evaluation on the different app versions, and the one with the highest version is displayed in the middle (Figure 5.1). They're also able to view the same results in two different ways visually. Note that the results in Figure 5.1 and 5.2, do not have the same exact figures, as the designs were done to solely illustrate the idea of showing the results page in two different ways. In *Figure 5.3, t*he user is able to view the performance analysis page of an application version's evaluation. Here, they can see all of the KPIs they defined during the set up phase, namely the Detection Rate, Correction Rate and Clarifying Rate. The user is also able to drill into what images these analyses were done on, and the rates of false positives, negatives and so forth.

First Iteration - Review

Upon reviewing this iteration with the product and design team. This iteration was then reviewed with the product and design team. This was done by showing them the paper prototypes that were created, and taking notes of their feedback. During this review, one thing that became more apparent, is that we needed to fine-tune the user scenarios, in order to make sure that this experience of evaluating an application, is in line with how the user would arrive at needing to do this evaluation. Upon doing this review, here are some keys scenarios/requirements that were deduced:

As an AI Manager or Solution Architect, I want to be able to:

- Launch an evaluation for a new app I just created. An evaluation corresponds to the KPIs and ground truth data.
- Compare my current application with past applications (or application versions), especially seeing comparison with the application that is currently deployed.
- Visualize the history of the evaluations and performances that I have launched over time, and what ground truth batches I used for these evaluations.

- When I edit my ground truth batch, I want to be able to relaunch the evaluation on present and past apps, so that I can get an idea on how previous and current app versions perform while using this new ground truth batch.

As a Solution Architect uniquely, I want to be able to:

- Specify some ground truth set to compute my evaluation.
- Build custom KPIs that matter to specific projects.
- Duplicate and edit an evaluation to change the ground-truth or KPIs that have been defined.

With these scenario and user needs made more crystalized, we decided to make the following key changes for the next version of the app evaluation feature:

- The starting point of the experience needs to start with selecting an existing application, or creating a new application, as opposed to starting with creating the evaluation first, before adding the application. This is because the apps are the main focus of the deepomatic studio experience, and the direction of the company is to have these new features and experiences to be application centered. In addition to this, based on the research we conducted, the user (being the solution architect or the AI manager) would typically begin with an examination of an application version that is currently deployed and isn't working as efficiently and thus needs to be updated, or with examining the need to create a new application. Thus, the structure and flow of the experience would need to be updated.
- The process of setting a KPI is crucial to the experience and would need to be further simplified for users that aren't familiar with a lot of these terms and technology.
- The process of delving into understanding the results shown, can get quite complex quickly. As it also varies from use-case to use-case. Thus, there is a need to start off with simplifying the process of viewing the results in a way that shows the most important information right away.
- Oftentimes, the AI Manager or the Solution Architect, aren't the ones who also carry out the deployment. Thus the experience would need to account for this. Oftentimes, the IT Manager, *Figure 7.0* is the one responsible for this. However, once again, this also depends on the enterprise client in question, but the experience needs to cater to both scenarios.
- The user may quickly get lost with certain terms being used such as - application, application versions and so forth. The experience would need to cater to this need to simplify these notions and make them easier to follow.

After coming to a conclusion on these takeaways from the first iteration, and feedback received, the second iteration of the feature was then worked on, and tested upon by a group of solution architects.

3.3 SECOND ITERATION - MID LEVEL PROTOTYPE AND STUDY

In this part of the iteration process, the feedback received in section 3.2 was incorporated into the prototype, to produce a number of screens. This time, the prototype was created and tested based on two main scenarios that the participants later tested:

3.3.1 First Task - Creating and Evaluating a New App - Day One Experience

In the first task, the participant, a solution architect working for Deepomatic, needed to perform two tasks:

a. Create a new application on the platform - Smartcheckout App.
b. Create the first version of the application.

c. Save the app version created so that the AI Manager can evaluate and deploy it later.

In the second part of the task, the AI Manager steps in to evaluate the newly created application, and marks it ready for deployment.

This task was created based on the key scenarios identified in 3.2. The idea is that the solution architect at Deepomatic and the AI manager at the enterprise client would collectively create and evaluate these applications and their app versions. Thus, we see that the solution architect primarily handles the creation of the very first app version, when the app itself is being built for the first time, and the AI manager would then handle the evaluation and kick off the deployment process if necessary. Recall again that the northstar goal is to streamline the experience down to a point where the AI Manager at the enterprise client is able to implement all parts of this process, but as a short term goal, it is a collaborative process.

In order to properly understand the screens that were created for the mid level iteration, it is important to understand the structure of applications and application versions at Deepomatic.

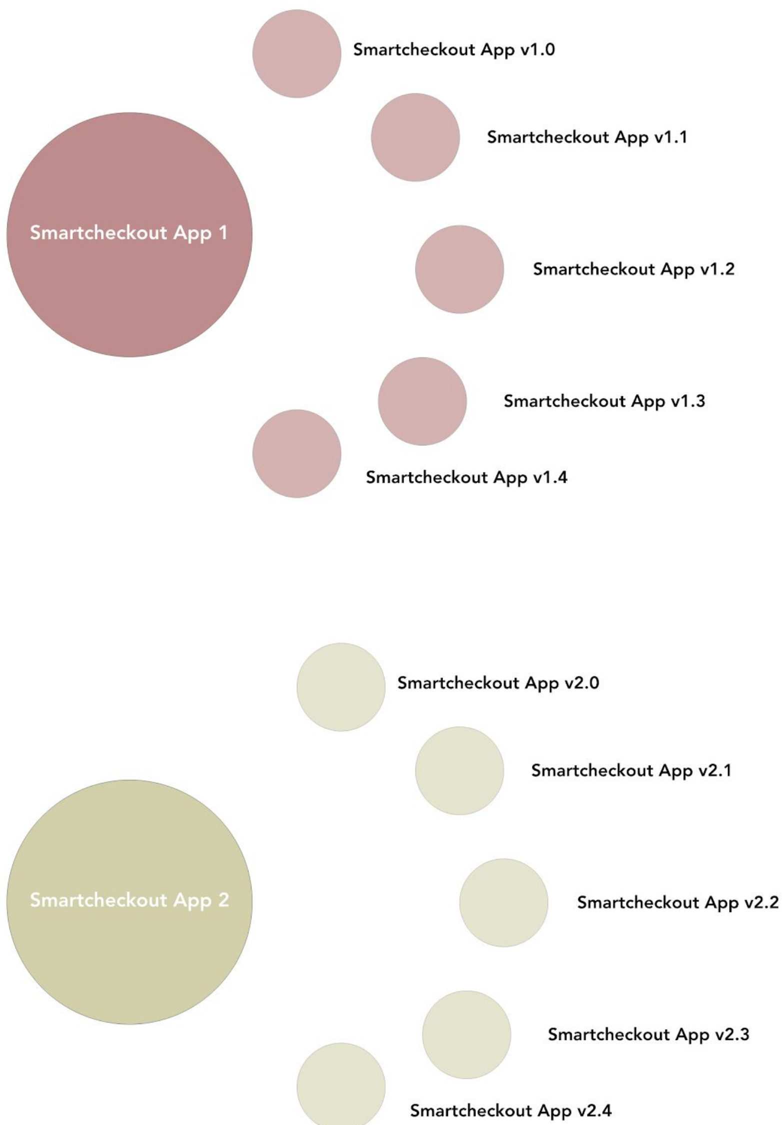


Figure 5.4 - An illustration of applications and application Versions

In Figure 5.4, we see an illustration of how these applications are structured. Take for instance the scenario being used to create these prototypes - creating a Smartcheck out App for the Automated Checkout solution, for an enterprise client called Compass Group, which is actually a real customer for Deepomatic. At the beginning of creating the application, *Smartcheck out App 1* is created, version 1.0. If there are updates made to the models inside of the application, the app would need to be updated, and thus the enterprise client would deploy version 1.1 to replace v1.0, and so forth. The enterprise client is also able to refer to a previous version if after an update deployment, the application doesn't work as expected.

Recall that an application is made up of models and a workflow/custom logic that explains how the models need to interact with each other. In the case that the workflow needs to be updated, a new application would need to be created, and this is where the Smart Checkout App 2 comes in, as shown in Figure 5.4. Similar to Smartcheckout 1, if there's a version update made, then the user is able to upgrade versions.

We see from this that the notion of an application is really based on the app versions. The applications themselves serve as blueprints upon which these versions are made.

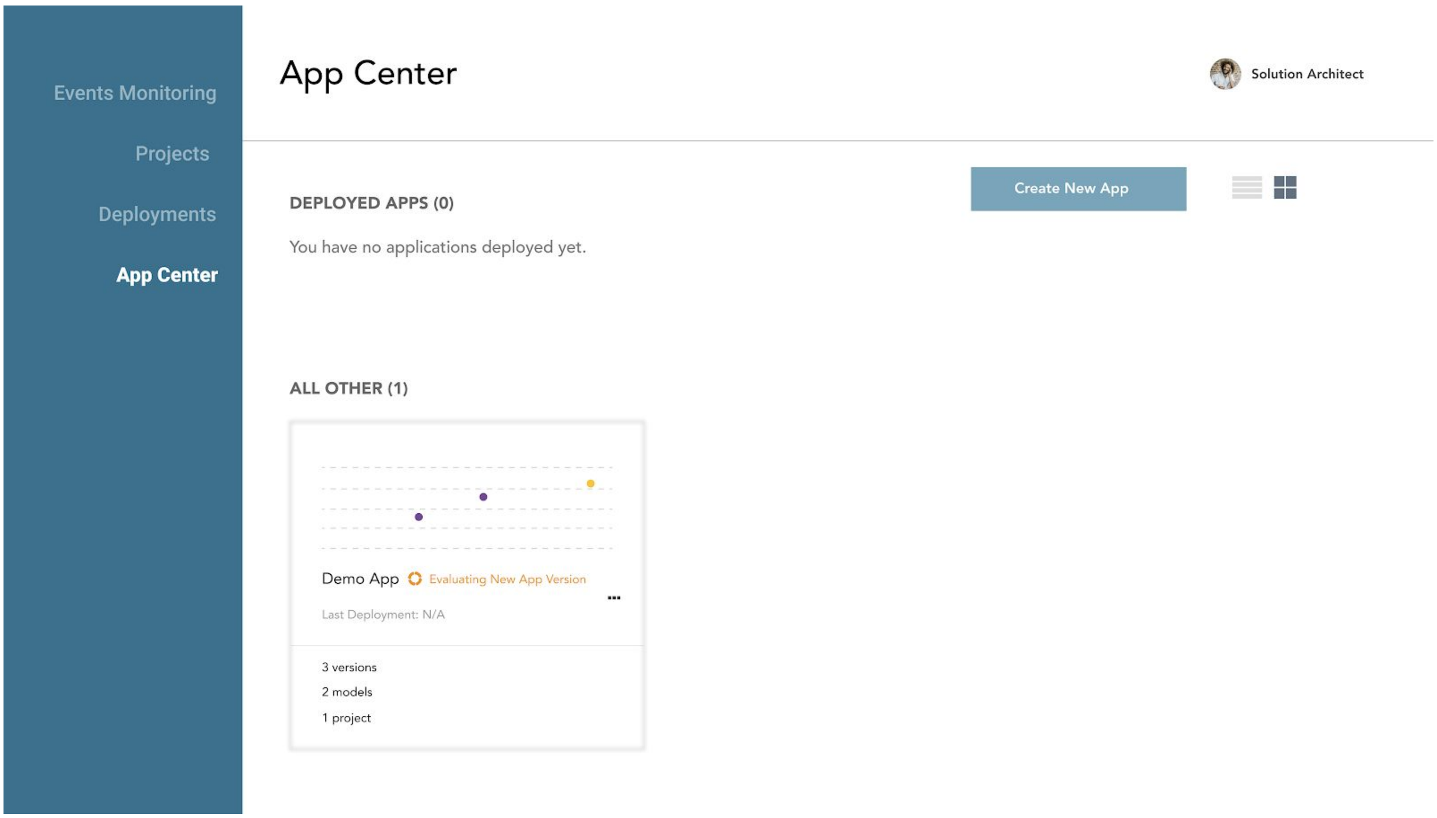


Figure 5.5 - Landing page

We see in Figure 5.5, that this version of the prototype begins with the user landing on a newly introduced section of the Deepomatic studio - the App Center. Upon arriving, they're able to view all of the application(s) they own, and their status(es). They're also able to create a new app from this page, and change the way of viewing the displayed information. Following the first task introduced above, in this scenario the solution architect needs to create a new app - a Smart checkout app, for the enterprise client - Compass Group.

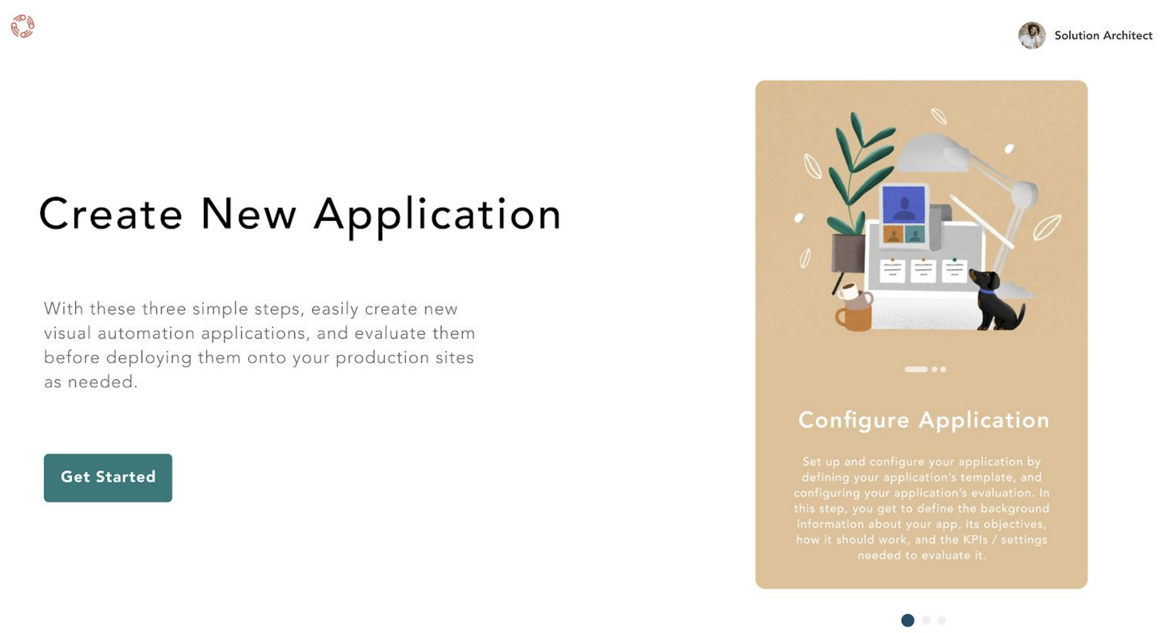


Figure 5.6 - Get started page

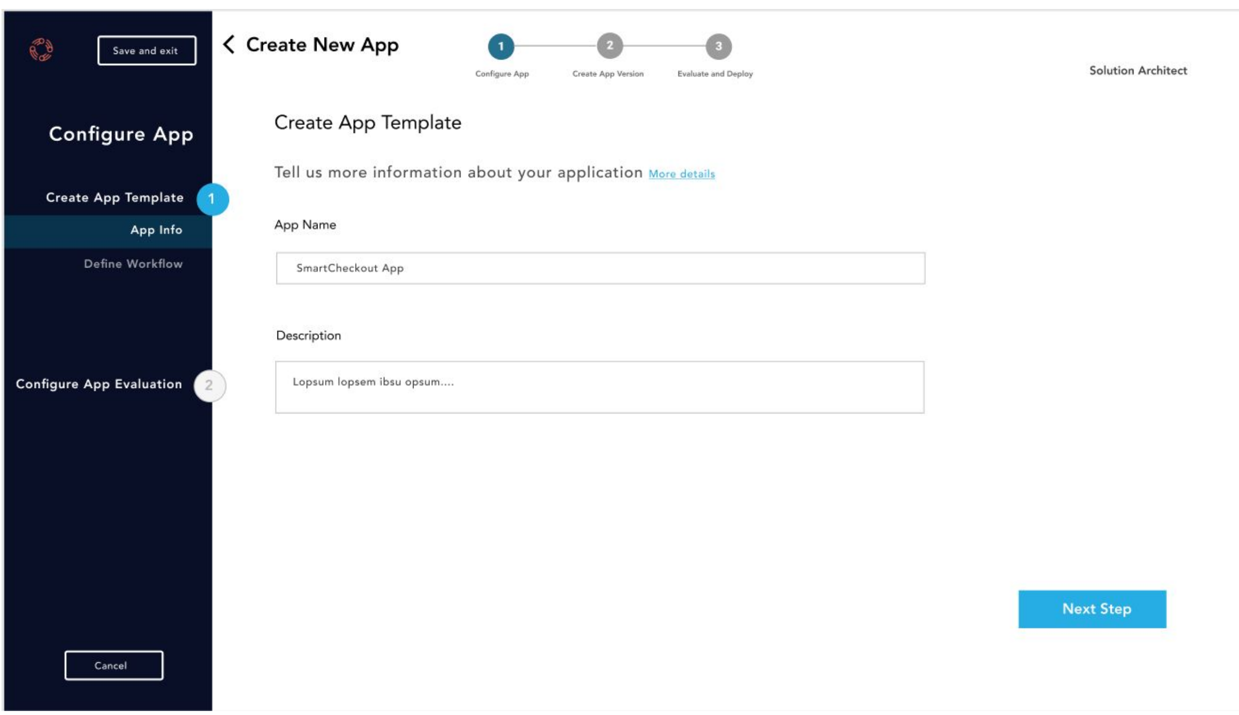


Figure 5.7 a - App Info

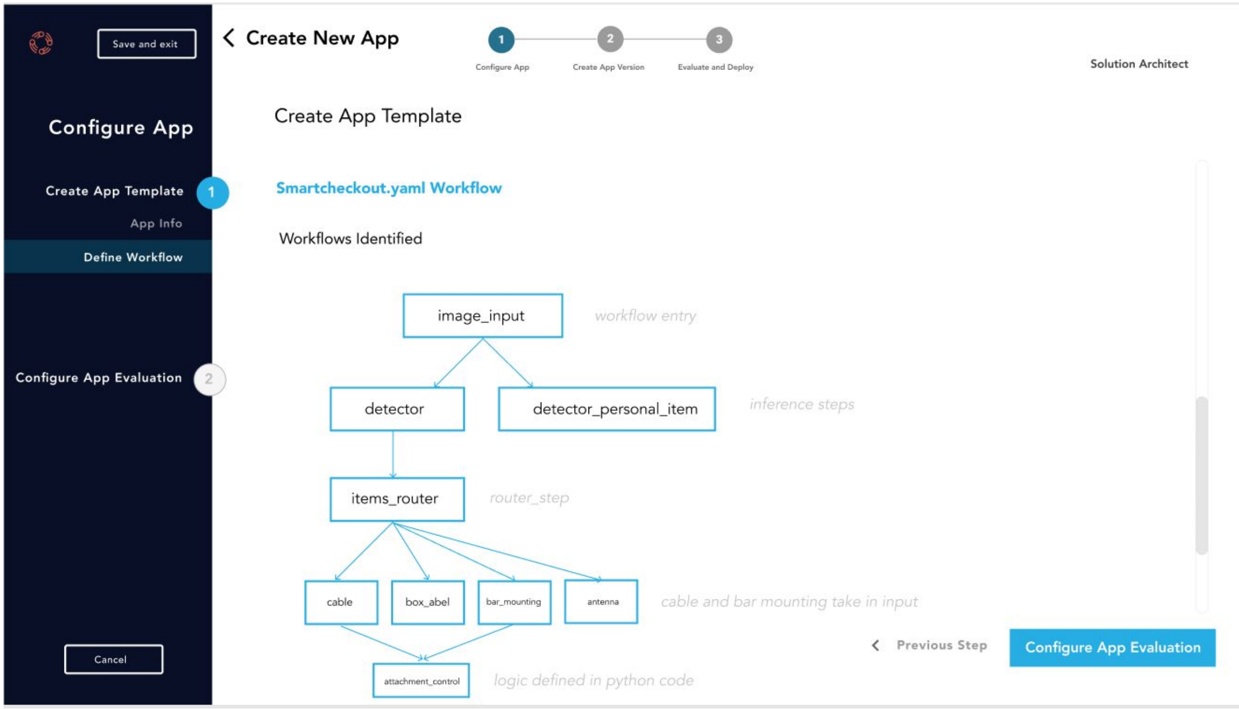


Figure 5.7 b - Defining App Workflow

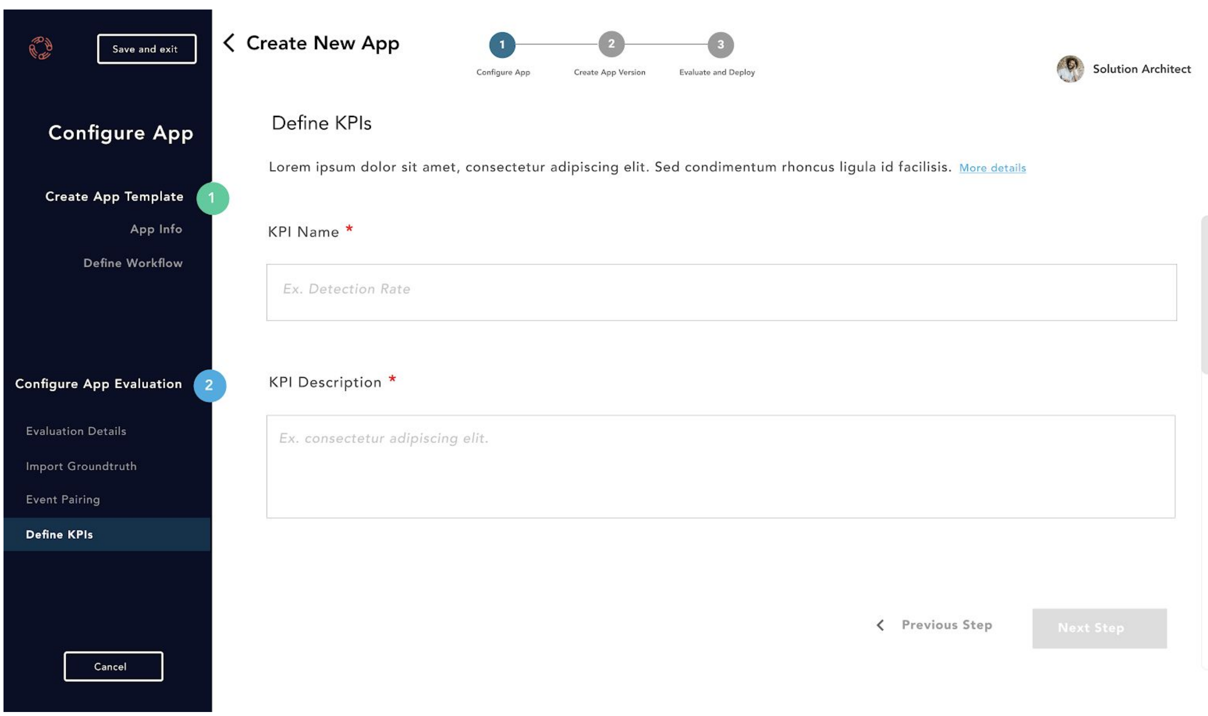


Figure 5.7 c - Configure application - Defining KPIs

As seen in *Figure 5.6,* the experience once again starts off with a Get Started experience created in the former iteration and then upon starting, the user views a configure application page, to get started with the process. There are now three main steps to creating an application. These three steps are shown at the top of the page in Figures 5.9 a, b and c - Configuring the application, creating the very first application version and the thirdly, deploying the app/marking it ready for deployment. For this first task, the user only needs to do the very first step - configuring the application.

Configuring the app: In this step, the user is building the blueprint for their application (app template) by providing the app info (name and description) of the application (Figure 5.7 a), as well as defining the workflow that defines what models and custom logic they would like to utilize (Figure 5.7 b). Afterwards, as part of this step, they need to configure the App evaluation, which tells the system how they would like to evaluate the very first version of the app that is yet to be created, and all future versions to be evaluated. In this part of the experience, they would once again need to define the KPIs they'd like to use, as explained in the previous section (*Figure 5.7 c).* For this iteration, the process of creating KPIs was simplified. The user is able to track these sub steps on the left pane, for each of the main 3 steps in the top middle.

Once the process of configuring the application is completed, the solution architect then saves the app template, and the AI manager is then notified that there is a new app that has been created. From there the AI manager completes the process by creating an app version for the app, evaluating it and marking it ready for deployment. In the next task, we show how the AI Manager does this, in a non-day one experience setting. We skipped showing the day one experiences for creating and evaluating an app version, in order to save time with prototyping and reviewing.

3.3.2 Second Task - Creating an App Version and Marking it Ready for Deployment (None Day One Experience)

The premise of this task is that a number of application versions have already been created, and now the AI Manager notices that there is an anomaly with the version of the app currently deployed, and needs to create a new app version to update the version deployed, evaluate this new app version and then mark it ready for deployment.

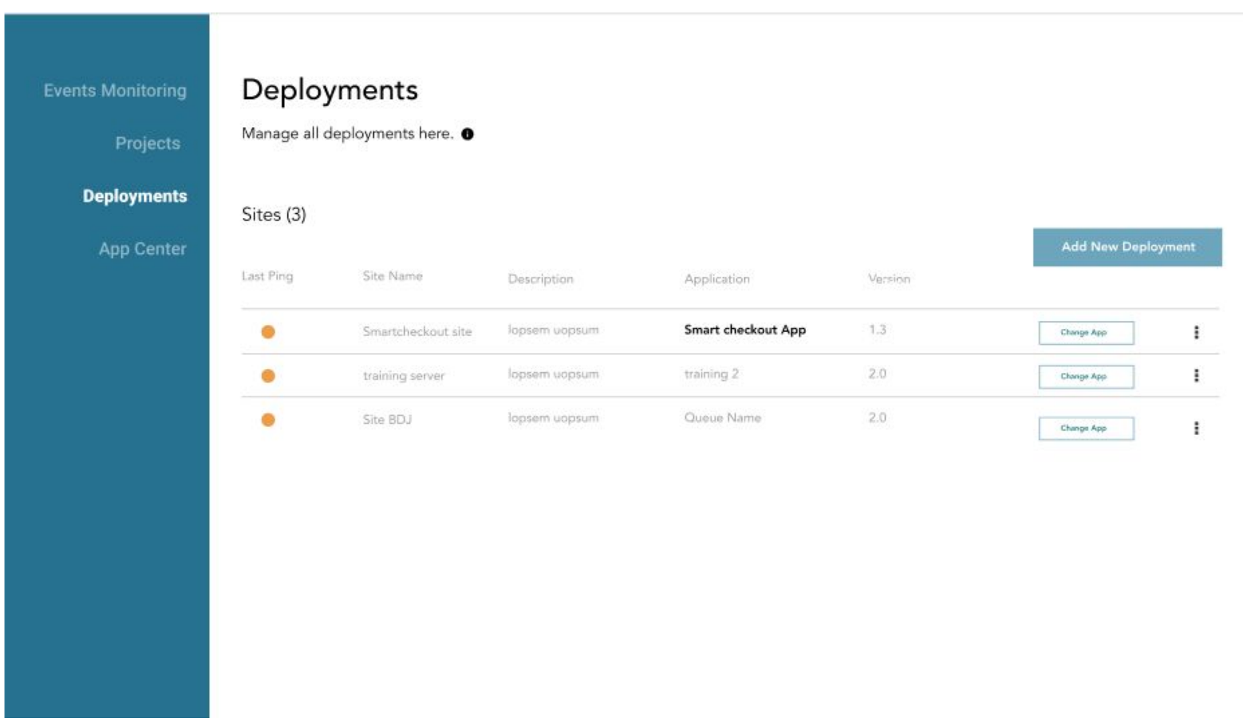


Figure 5.8 - Clicking on an application from the deployments page

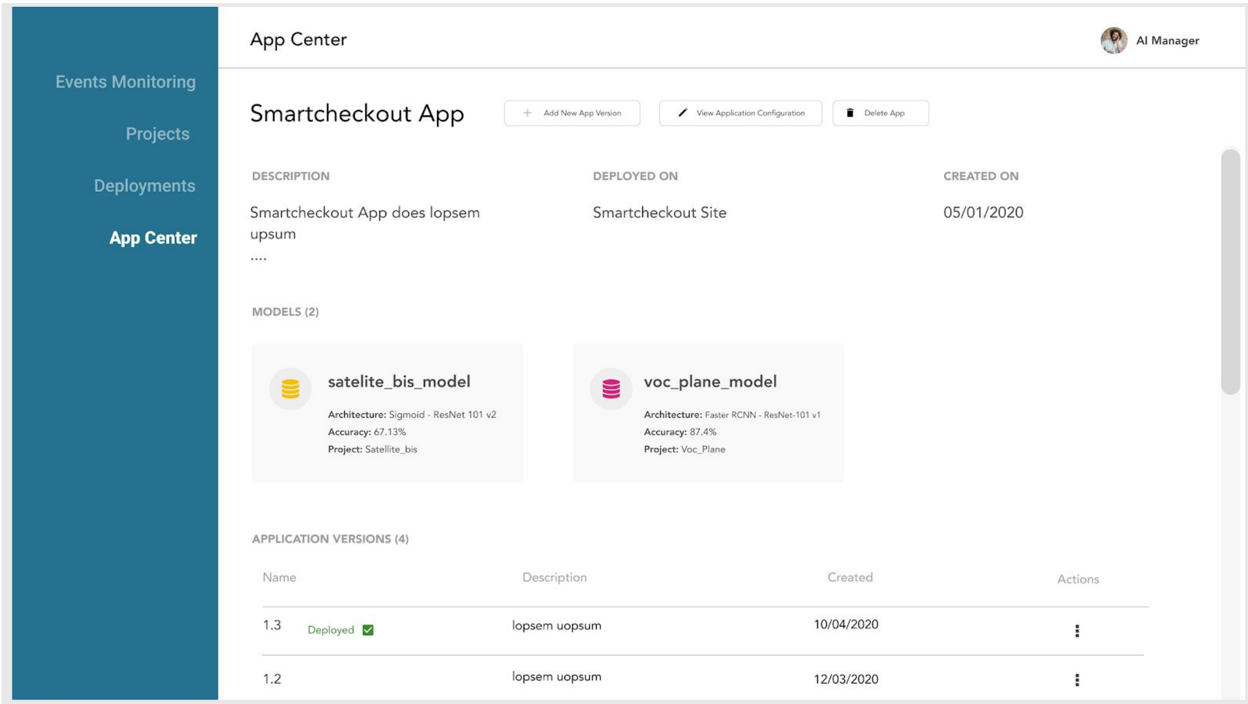


Figure 5.8a - App Detail Page

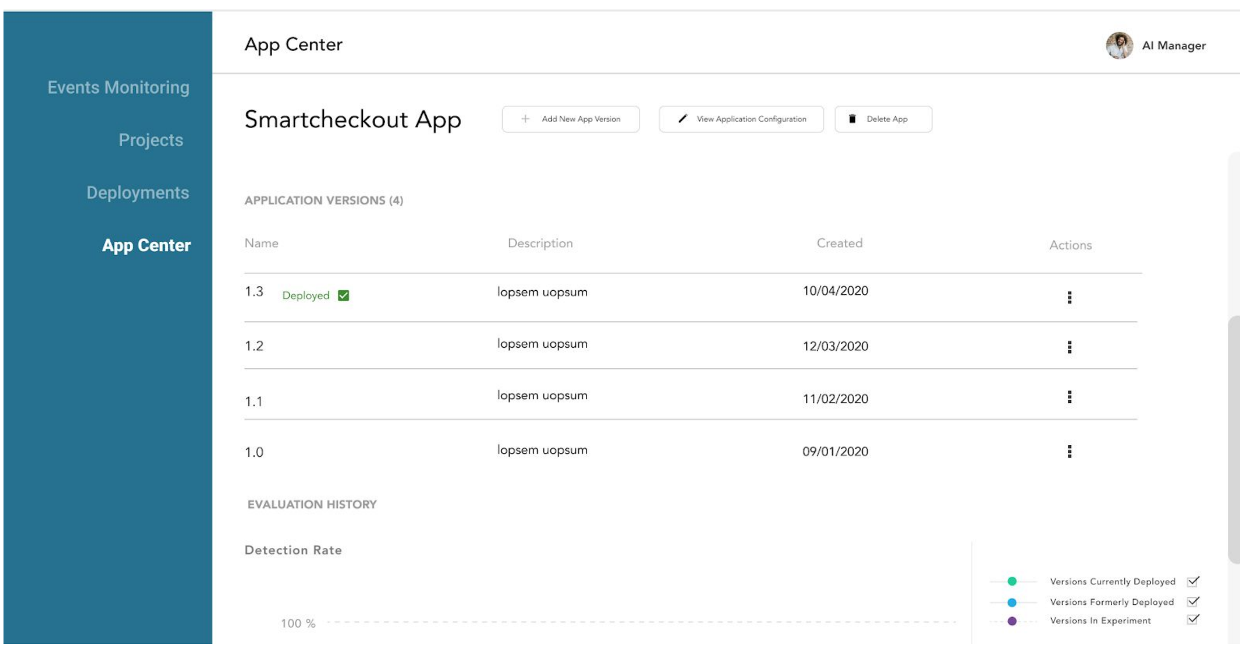


Figure 5.8b - App Detail Page

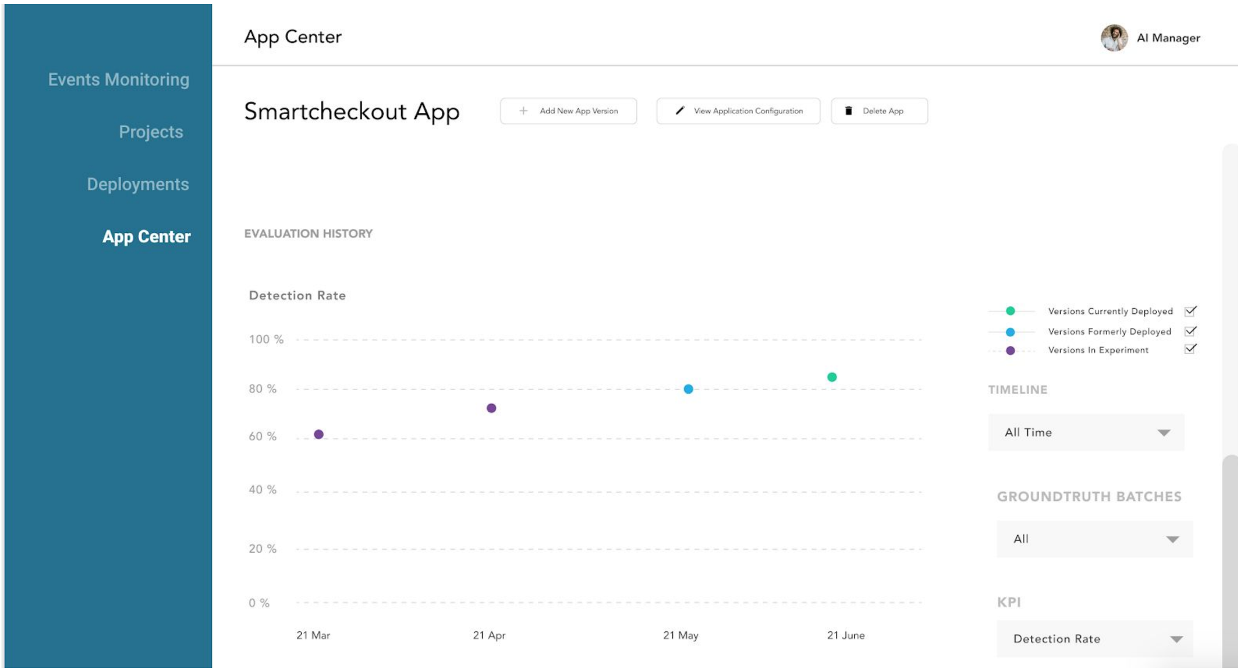


Figure 5.8c -App Detail Page

From Figure 5.8, 5.8 a, b and c, we see the AI Manager is able to first find the application that has been deployed and has an anomaly (Figure 5.8). They then click on the application to arrive at the App Detail page of the Smartcheckout Page (Figure 5.8a), and then from here, they are able to create a new app version. On this detail page, information such as the models in the application and number of versions of the application that exists and their statuses has been displayed. We also see the Evaluation History over time, and the results of the KPIs for each app version.

From this page, the AI manager creates a new application

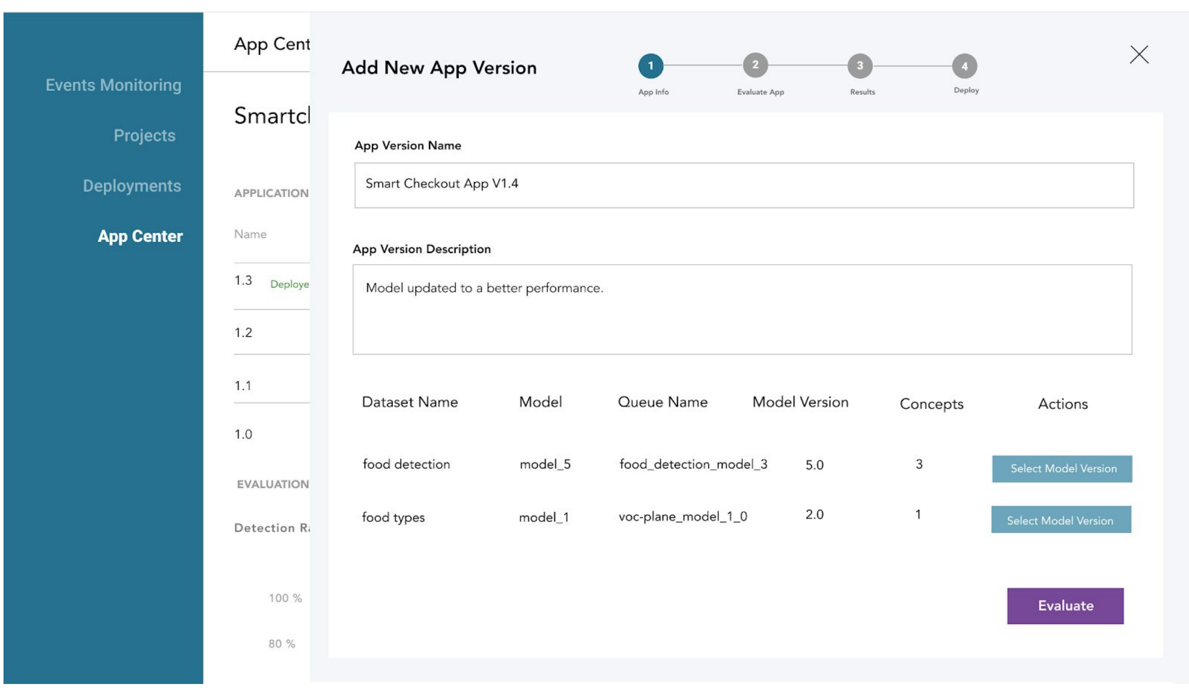


Figure 5.9 - Creating new app version

As seen in Figure 5.9, the AI manager then walks through the steps on creating this app version namely - filling out the app info, evaluating the app version using the evaluation that was configured at the creation of the application itself as explained in the first task, viewing the results from the evaluation, and then marking the application ready for deployment. We see in Figure 6.0, that the process of marking it ready for deployment requires the AI manager to send a notification to the IT Manager, stating what site to deploy the application on, and more notification about the application and deployment.

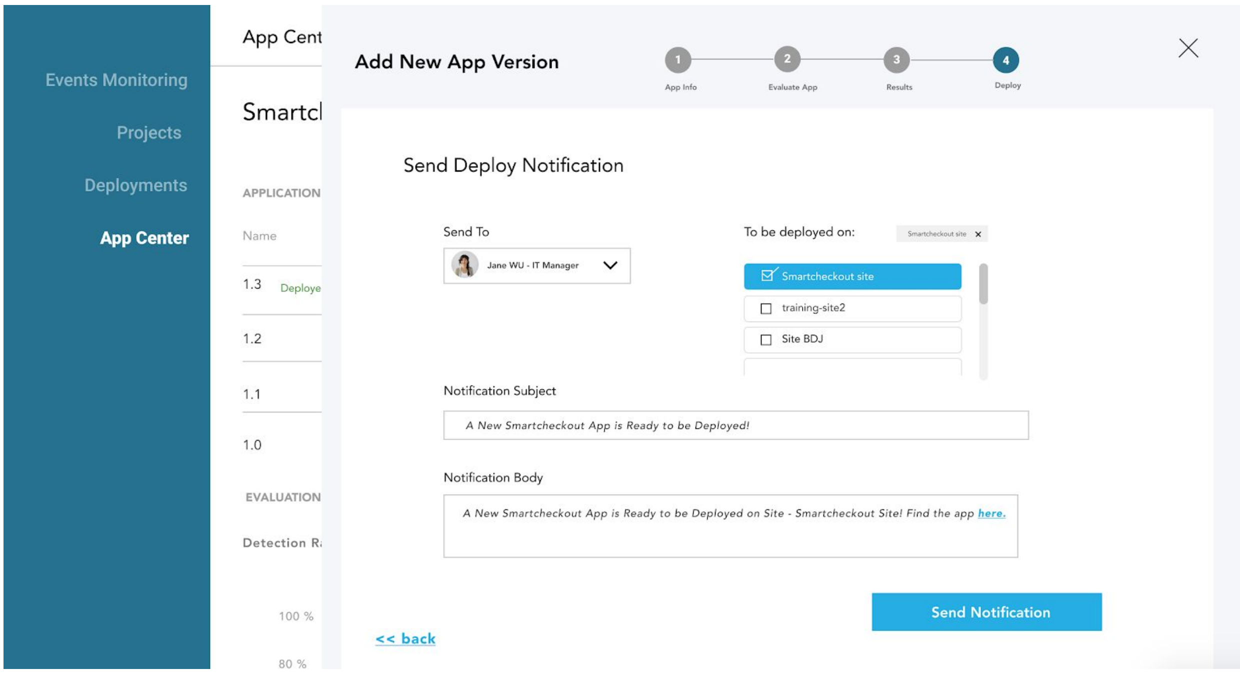


Figure 6.0 -Send Deploy Notification

3.3.3 Third Task - Deploying the App Version in Production

In the third section of the prototype, we're covering another major scenario which is deploying the app version in production. This task involves the IT Manager who also works with the AI Manager at Compass Group, the enterprise client. The IT Manager receives a notification (Figure 6.1) sent by the AI Manager to deploy a new app version and then carries out the deployment by updating the site as shown in Figure 6.2.

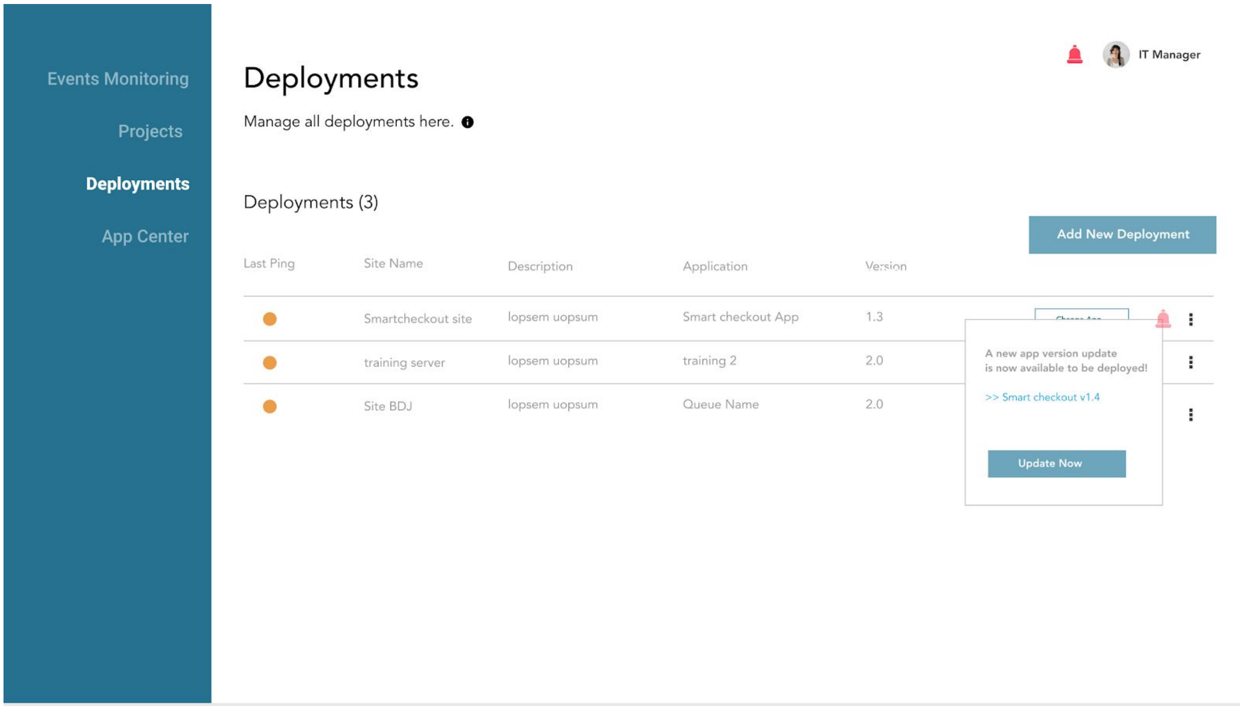


Figure 6.1 -Viewing Deployment Notification

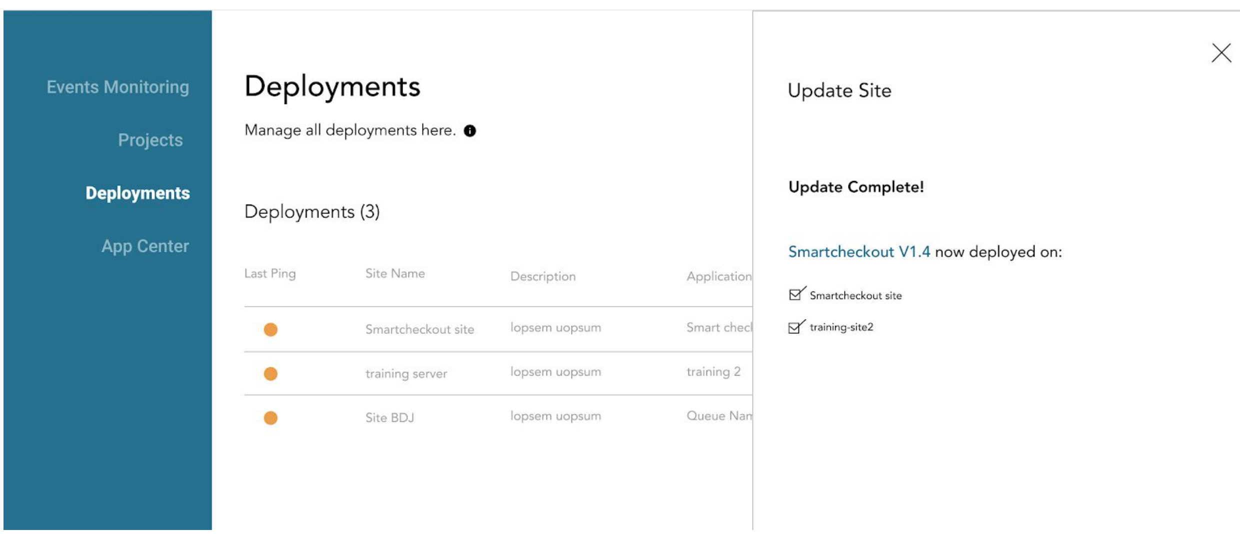


Figure 6.2 - Updating the site with the latest app version

3.3.4 Study

Once the prototype was built around the tasks and scenarios highlighted above, a study was carried out to see how efficient and practical it was for actual users to carry out these key tasks. The goals of the study were:

- **To test the feature usability and practicability:** This study was carried out to test the ease of use and how practical all participants found the experiences, ranging from participants familiar with the concept of the feature, and others with less familiarity.
- **Reveal points of friction:** Another goal of the study was to understand what elements participants found confusing, to see what areas or parts of the experience needed to be smoothed out.
- **Understanding user needs:** Lastly, the study was carried out to understand what type of information users need to see at different points during the experience. What they found relevant, what element they didn't care as much for.

Methodology

As the application evaluation feature is to be used by both internal and external personas - namely solution architects (internal persona that works for Deepomatic), AI and IT managers (external employees that work for the enterprise clients), this iteration was tested by two solution architects, who were already familiar with the process of creating applications, evaluating and deploying them, all currently being done via code, with an exception of applications that are created on the studio platform without custom logic. This prototype was also demoed and presented to the CTO (chief technology officer) and VP (vice president) of Engineering, in order to also get feedback from the engineering perspective. This gave us a chance to receive immediate feedback on the usability and practicality of the design, and understand if the experience would be practically used in the way it was imagined to.

There participants were all selected as internal employees with different exposure with the product as previously explained. Using a Figma prototype that was created, participants were told to complete the three tasks illustrated above, around configuring an application, an application's evaluation, creating new application versions and deploying applications. After they completed each task, they were able to then advance to the next to keep a sense of continuity.

The study was conducted online via Zoom, and each session with a user was recorded with approval from the user.

As a recap, the tasks they needed to carry out were:

1. Creating a new app, by configuring the app template and app's evaluation - Day One Experience, as a solution architect.
2. Creating an app version, evaluating it and marking it ready for deployment, as an AI manager.
3. Identifying a site ready for an app version update, and deploying the new app version in production on the right sites, as the IT manager.

In the first task, they played the role of a Solution Architect (themselves) and in the second and third tasks, they played the roles of an AI Manager and an IT Manager. We decided to test these roles internally, because at the beginning of the feature being used, the tasks carried out by the AI Manager and IT Manager will be done initially by the solution architects, as they currently do it today with code, before a total transition is made to AI and IT Managers covering these tasks. In the future studies, these prototypes would also be tested with external users (AI and IT Managers).

The questions asked during and post the sessions included:

- How would you rate the prototype's ease of use on a scale of one to ten?
- What was your favorite part of the experience?
- What parts did you struggle with?
- What parts did you find frustrating?
- Do you have additional feedback for us?

Along the way, more questions were asked as we got into conversation with the participants to better understand their thought processes while working through it. Once the study itself was completed by each persona, the questions above were asked.

Feedback

Common feedback was received from responses to the questions we asked the users stated above, observations, and thoughts that the users expressed along the way during and post the sessions. The feedback was then consolidated around tasks one, two and three, by merging some of the repeated feedback together. The main feedback received were:

Task One:

- Users needed to be able to view an application's workflow on the application's detail page, and not only the models and other information displayed. As it is a crucial part of the application, it is important to visualize it to see the individual boxes that make up the workflow and what each box does.
- In the day one experience in task one, users expressed wanting to be able to skip the *Configure an Evaluation* part, and do it at a later time since it often takes some time to complete.
- Participants struggled with understanding certain concepts such as how Groundtruths are set up, and the process of matching events from the ground truth with events predicted by the application. In order to do this, the system needs to be able to know how to pair events, and this event pairing process designed in the experience caused confusion and increased the amount of time participants spent in completing a task.
- Creating a KPI was still quite difficult for all participants to understand. Participants talked about different aspects to it that were abstract, such as defining the KPI logic, and working with event sets to do so.

Tasks Two and Three :

- On the app detail page, the participants appreciated the ability to view all the information about an application all in one place, such as the evaluation history of the application as shown in Figure 5.8c, however, this was a part of the experience that also received a lot of feedback. Some participants expressed wanting to be able to view all KPIs at once on the graph.
- Around Deployments, users expressed the need to sometimes make a deployment at a later time, and there were discussions around making a clear distinction between the app creation and evaluation process, and the deployment process, since in some cases they may be most likely handled by two different departments.

Overall, all participants and employees that were demoed to, found it easy to use and were quite impressed by the ease of use and the translation of a programming experience into a UI/UX version that was easier to follow and understand. They were excited about the new application evaluation feature and the prospect of clients positively reacting to it. However, there was also overall feedback on creating an experience that could be more customizable for other use-cases and applications, outside of the Smartcheckout app. In the next section, we will discuss how the final iteration was created, based on the feedback from the mid-level iteration.

3.4 THIRD ITERATION - HIGH LEVEL PROTOTYPE

This last iteration was focused on getting the mid-level version of the prototype shown and discussed in section 3.3, to be completely in line with Deepomatic's current UI design systems and attributes, and also incorporating the feedback received. Adapting to Deepomatic's UI entailed changing colors, fonts, icons and other attributes that were used in designing previous iterations, to ensure that the UX/UI for the high-level prototype is seamless with the current Deepomatic studio platform, unlike the design used in the mid-level.
Once again, this version of the prototype was created and updated around major themes that are part of the user experience.

Create New App

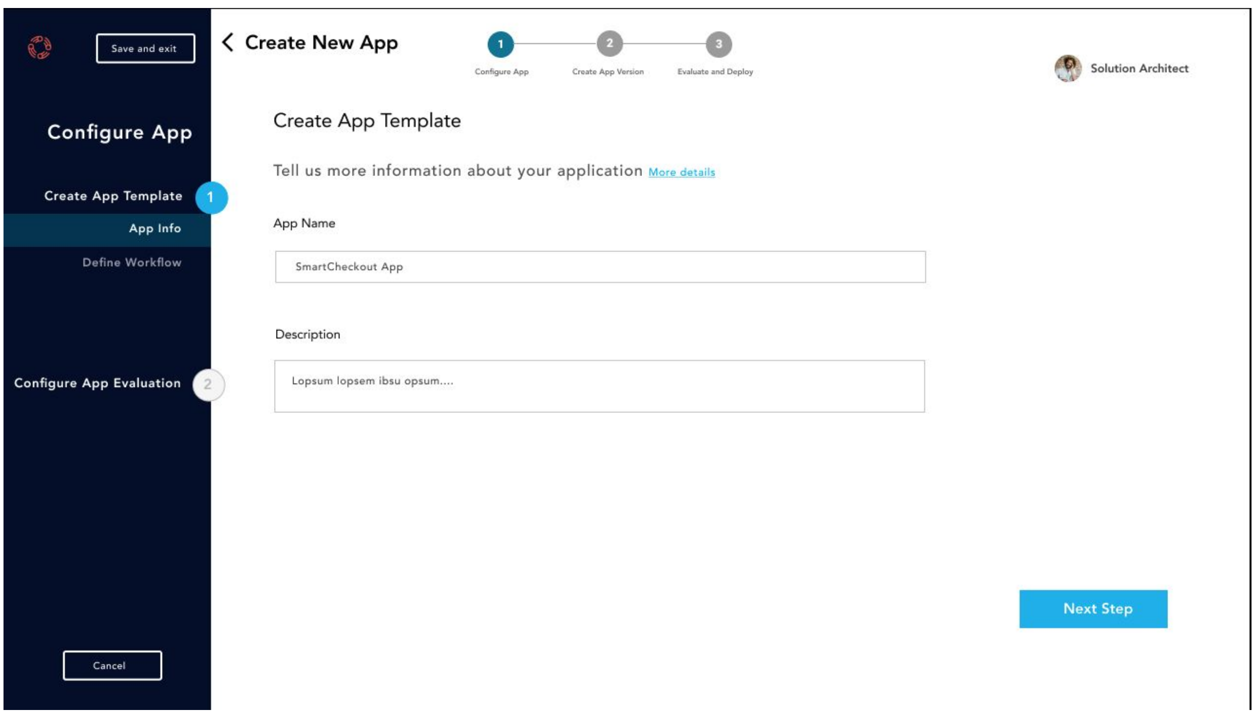


Figure 6.3 a - Mid Level version of Create App Template

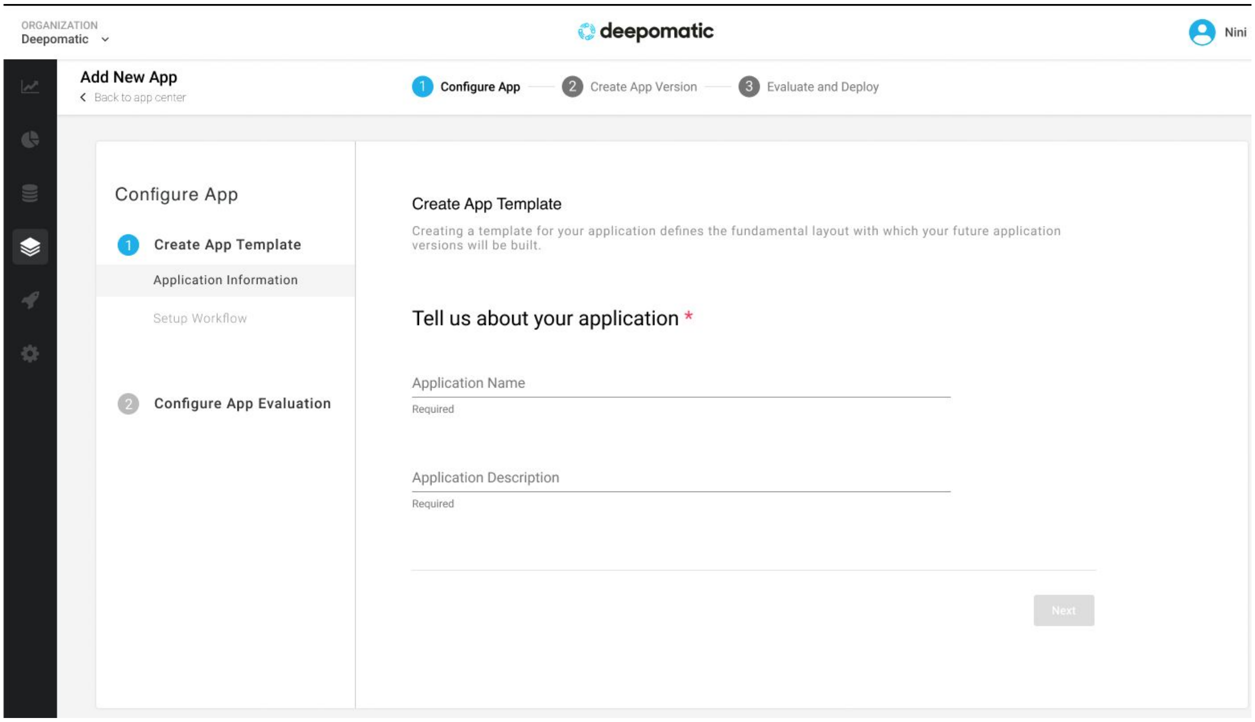


Figure 6.3 b - High Level/Updated version of Create App Template

This screen was updated *(Figure 6.3 a and b)* to be more in line with a screen that would match Deepomatic's studio. This was all done by displaying the main steps and substeps in a format that followed Deepomatic's UI standards of displaying similar types of information on the platform.

The wording was also updated to use more natural language in telling the user what we would like for them to do on this page. Also, more description and guidance has been added on this page, with the text just beneath the "Create App Template" header, that tells the user what this step is all about, and also the text beneath the question the user is asked here - "How would you like to set up your workflow?", that informs the user how workflows are usually set up, so they're more guided on what button to select based on what they need (Figure 6.4 a).

Viewing a Workflow After it Has Been Imported

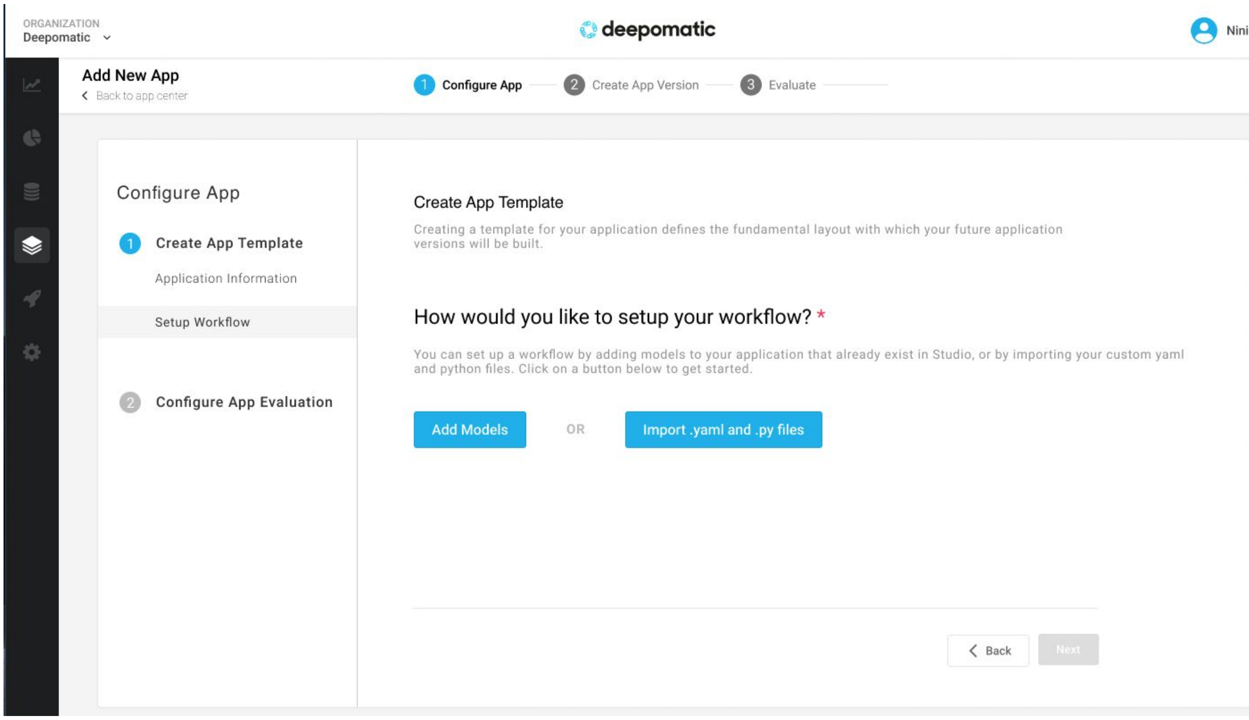


Figure 6.4 a - Viewing an Imported Workflow

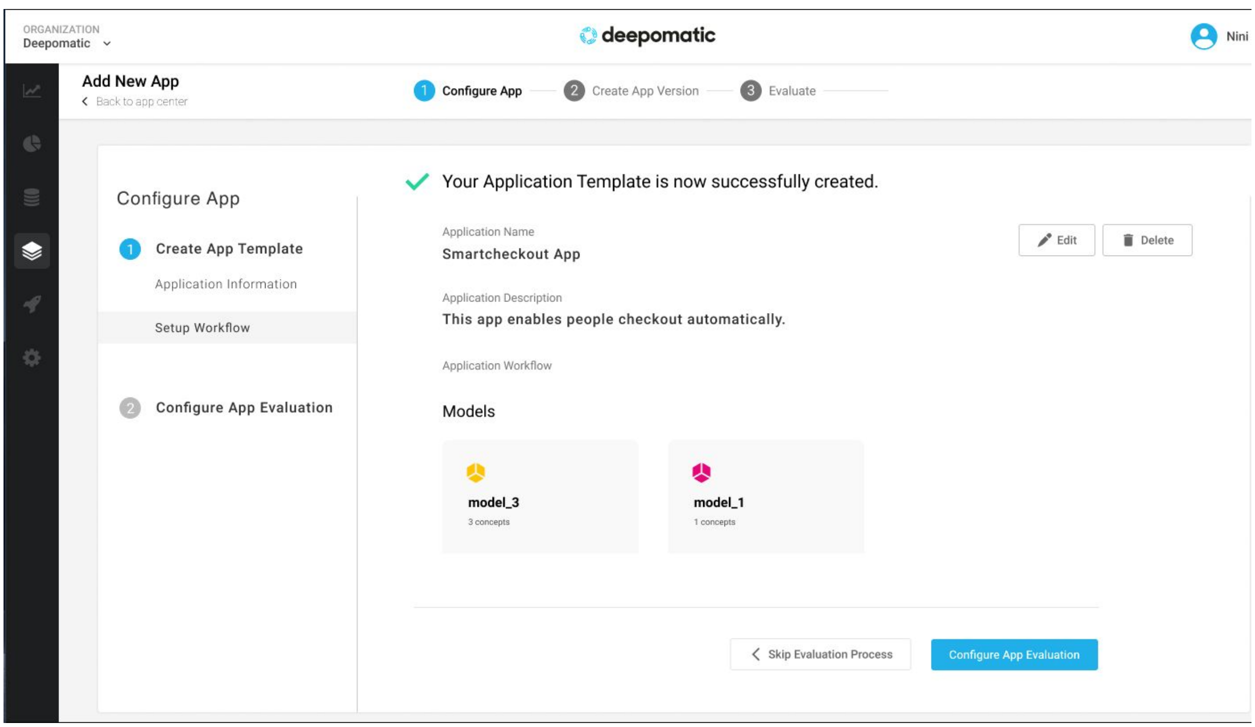


Figure 6.4 b - Viewing an Imported Workflow

These (Figures 6.4 a and b) show the new screens that were created for the high-level experience of importing and viewing an imported workflow. We got a lot of feedback on the mid-level iteration discussed in section 3.3, about how important it was for solution architects and AI managers to have access to viewing and understanding the workflow. Thus, we see in *Figure 6.4 a and b*, that in the process of creating an application, the user is asked to import the workflow and then is able to view it right after it is imported.

Defining a new Evaluation Metric

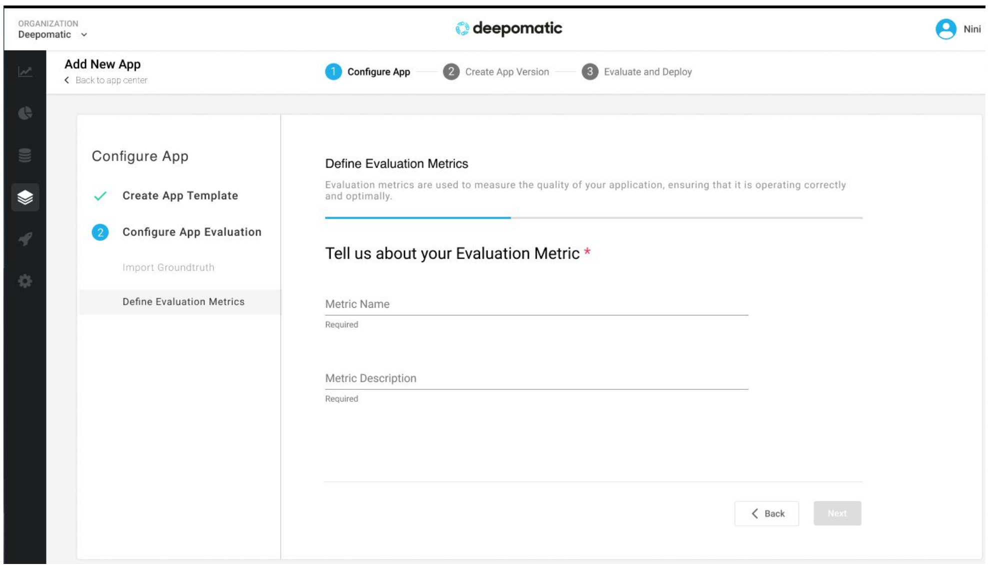


Figure 6.5 a - Defining Evaluation Metrics

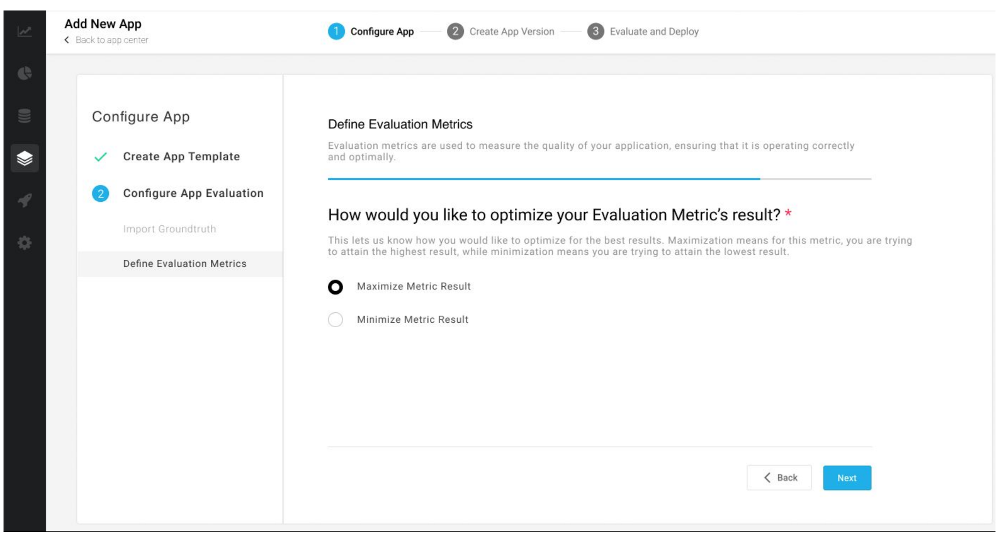


Figure 6.5 b - Defining Evaluation Metrics

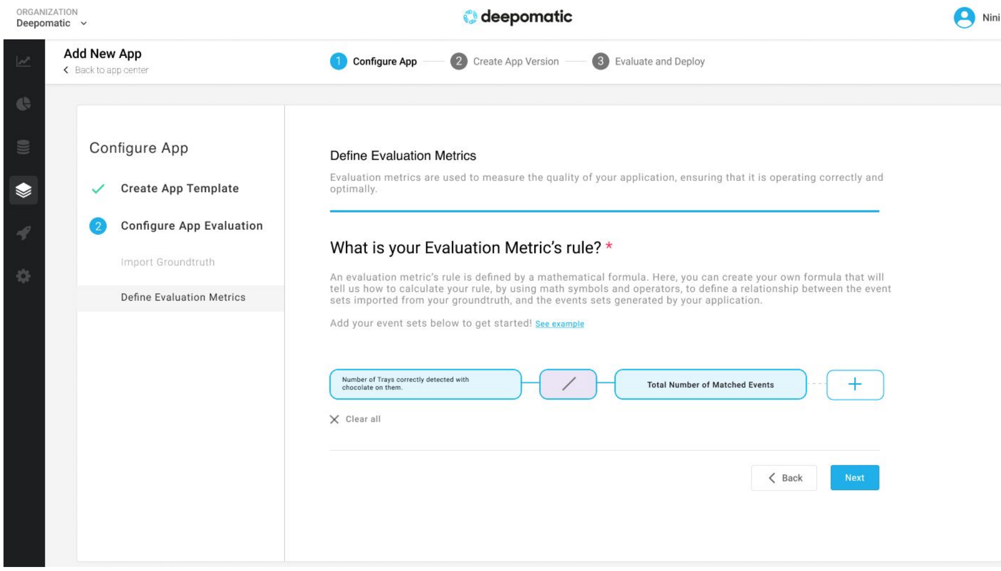


Figure 6.5 c - Defining Evaluation Metrics

Defining a KPI from previous iterations got changed to Defining an Evaluation Metric. There was a conscious effort to make this step of the evaluation configuration process as easy to understand as possible, and changing the name from KPI to Evaluation Metric was part of it. The steps to creating an evaluation metric were listed all in one page in the previous iterations, however in this high-level version, they have been broken down to separate pages, with a progression bar to help the user know where they are in the process (Figure 6.5 a, b and c). Along the way, more guidance was provided to the user with text, and examples at each step. This was done based on one of the major feedback received on the mid-level iteration. The text and guiding examples aided in breaking down the definitions of some of these concepts, by explaining that an evaluation metric is at the base a mathematical formula, that tells the system how to calculate the rule, by using math symbols and operators, to define the relationship between event sets imported from the user's groundtruth, and the events sets generated by the application.

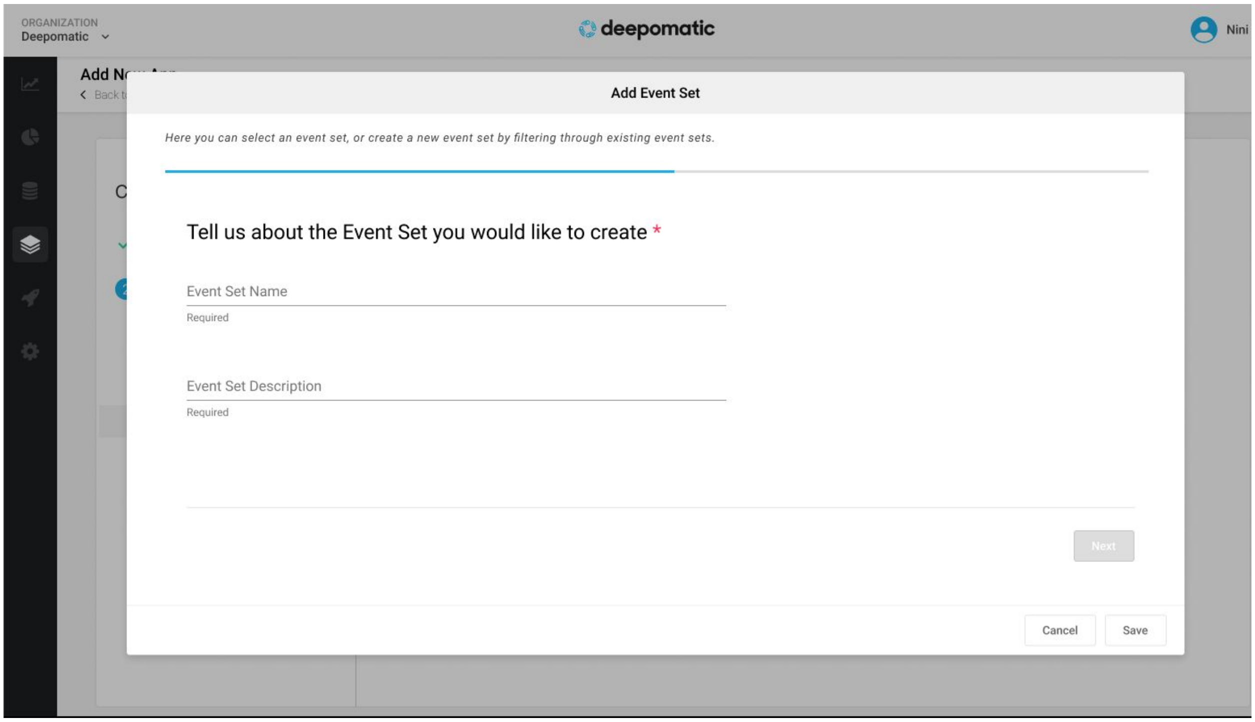


Figure 6.6 a - Adding an Event Set

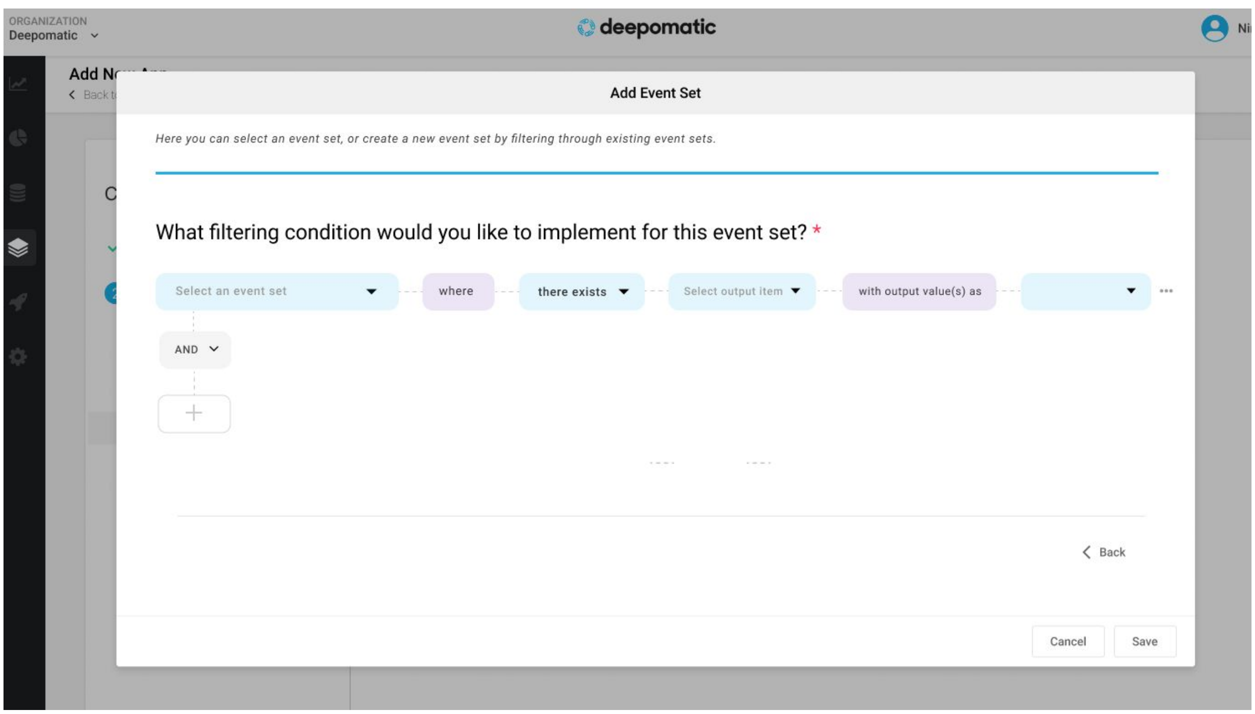


Figure 6.6 b - Adding an Event Set

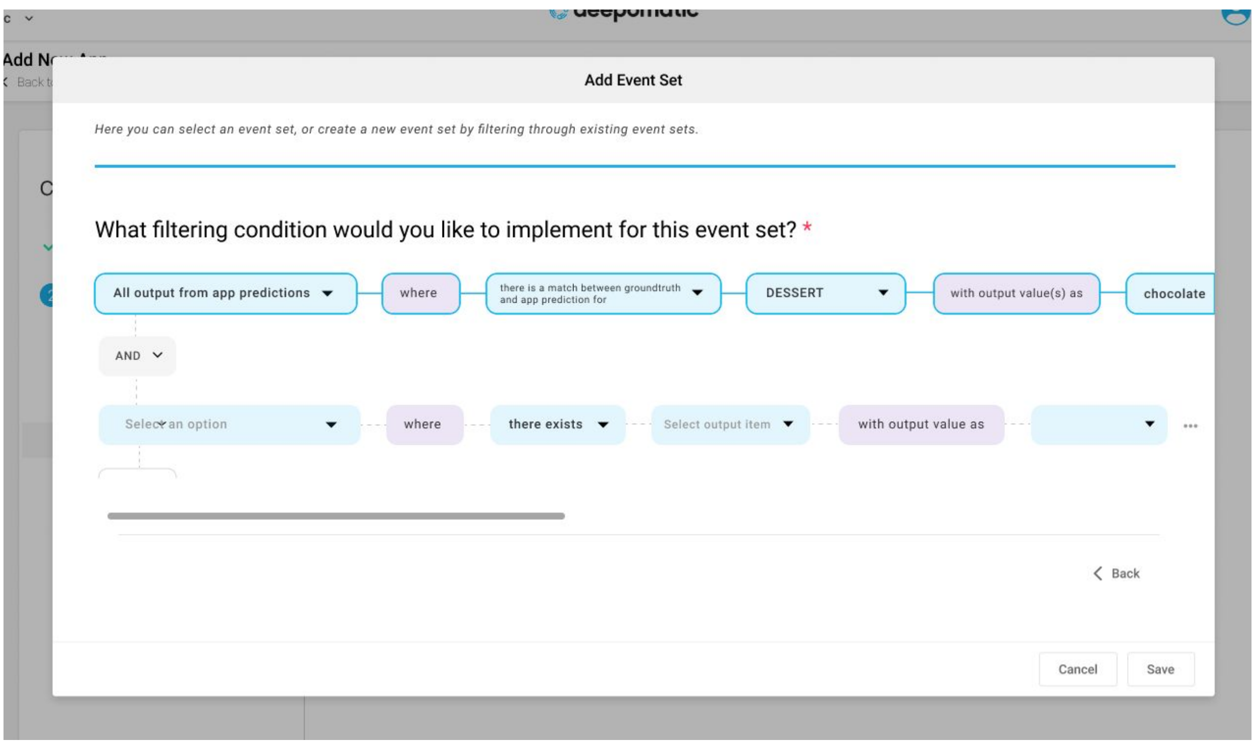


Figure 6.6 c - Adding an Event Set

More effort was also put into ensuring that the process of adding a numerator and denominator to the evaluation metric was seamless and made clearer. This process is called Adding an Event Set. These event sets are what form the numerator and denominator of the evaluation metric rule, and they are created by creating a filtering condition on the event sets gotten from the ground-truth and the events being predicted/generated from the application, as shown in Figure 6.6 a, b, c above.

Viewing the Detail Page of an App

The App Detail page (Figure 5.8 a) was updated below in the high-level version to display the workflow from the application, and changes were also made to the Evaluation History chart, in Figure 6.7 c. The main changes being, the ability to view and compare several evaluation points, based on multiple evaluation metrics, at the same time. Another being the ability to easily view and choose the metrics, as they were moved from being displayed on the right of the chart, to being displayed just above it. Also notice in Figure 6.7 c on the evaluation chart, you see two lines shown on the chart, each corresponding with an evaluation metric. Because two evaluation metrics have been selected in this instance, two lines are shown, and they're also shown on different Y scales. When a line is double clicked on, the user is able to view a readjusted graph to the Y scale of that sole line/metric (it went from being displayed from 0 - 100%, to 0 - 50%), as shown in Figure 6.7 d. This made the experience more interactive and comparative - thereby making it easy to compare several evaluation points, based on multiple evaluation metrics at the same time on the same page.

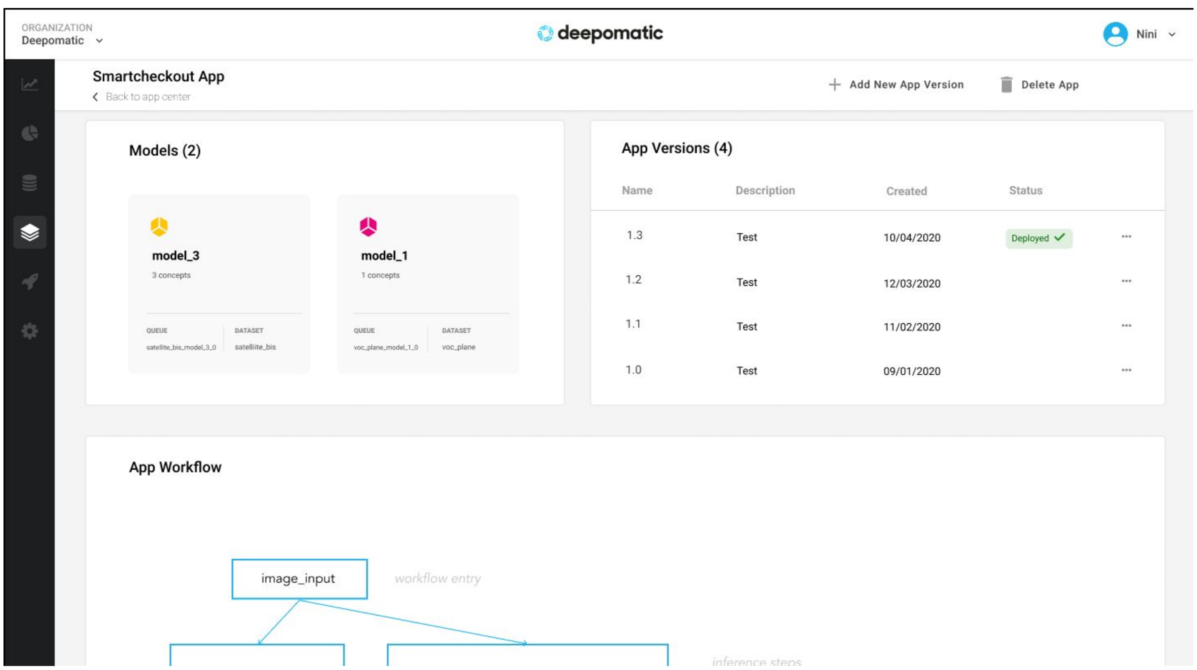


Figure 6.7 a - Viewing the Detail Page of an App

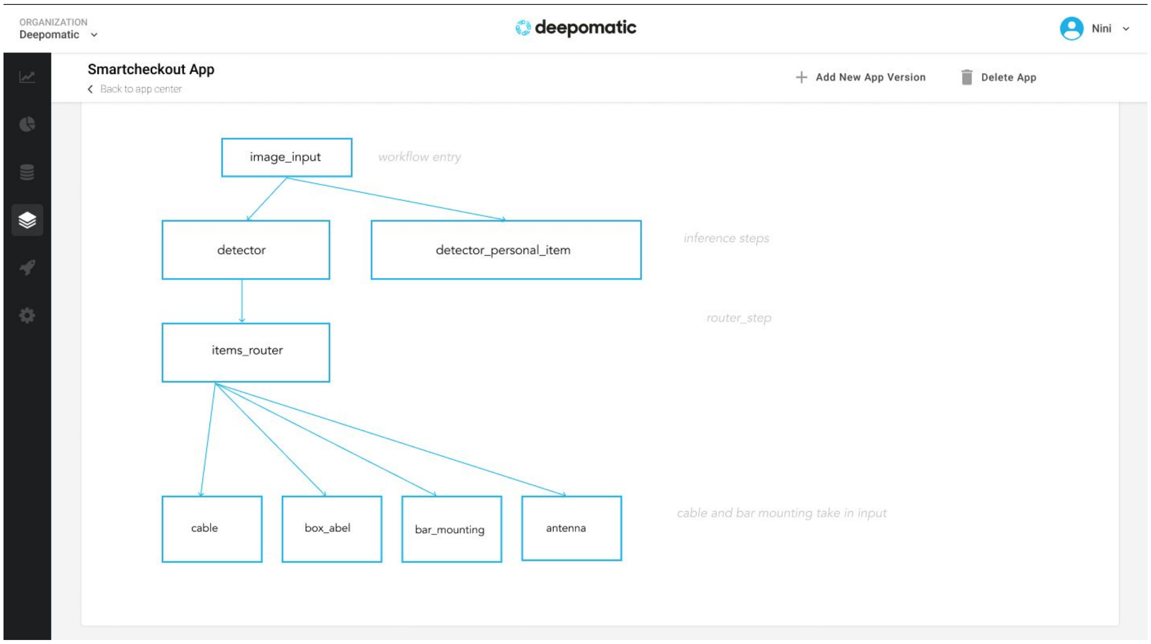


Figure 6.7 b - Viewing the Detail Page of an App

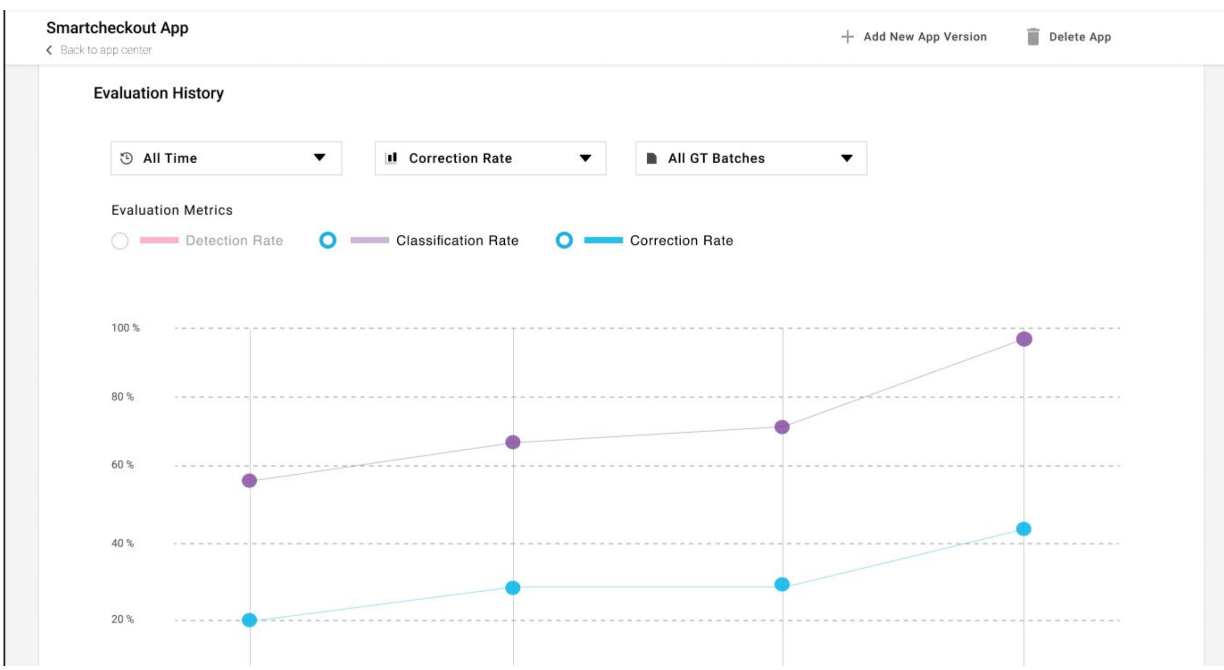


Figure 6.7 c - Viewing the Detail Page of an App

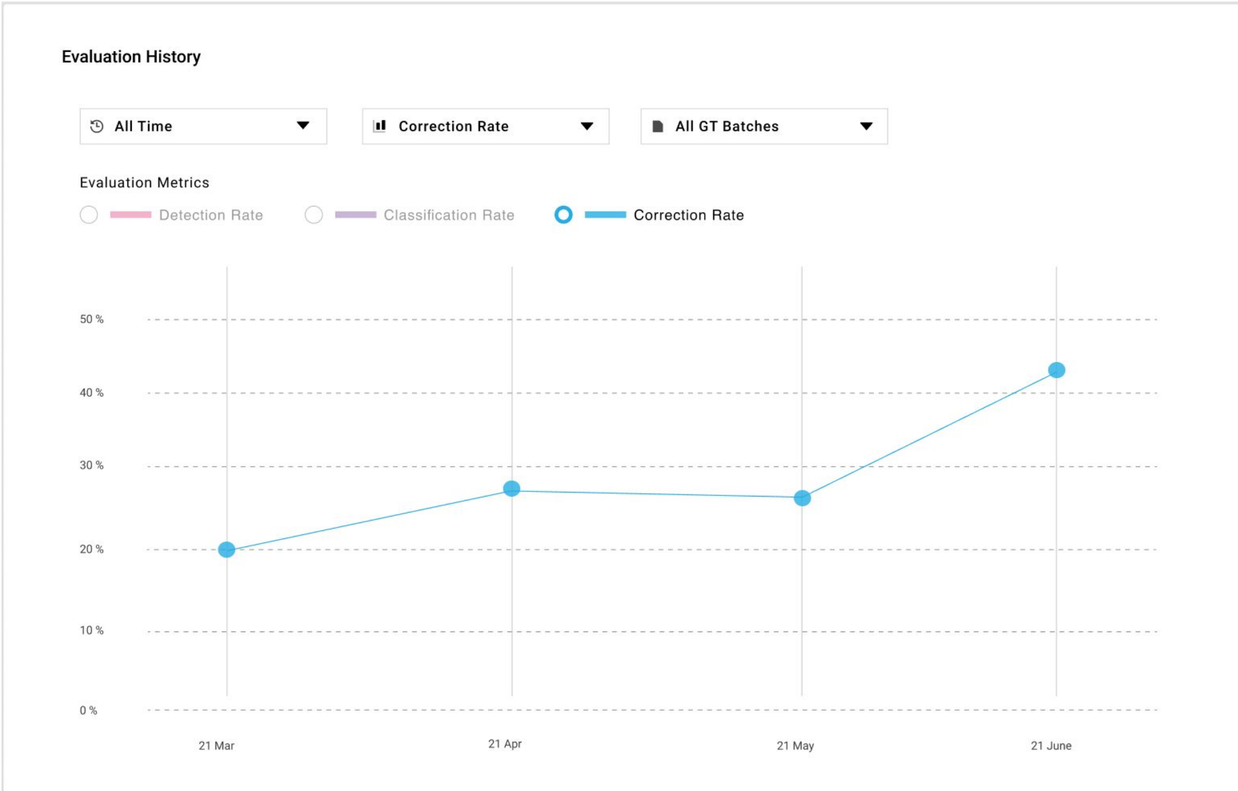


Figure 6.7 d - Viewing a metric chart that has been re-scaled.

3.5 CONCLUSION

In this chapter, we worked through the user research and ideation process of the Application Evaluation feature, as well as three versions of the prototype that were iterated on and reviewed/evaluated at each point, until we arrived at a version that fit well visually and interactively with the Deepomatic studio. Due to the end of the internship, this version didn't get to be reviewed with a study, but it received positive feedback from the product team, and was considered feasible enough to be implemented and well-scoped by the engineering team.

Design Principles

Some of the UX design principles that were leveraged in the design process, included ensuring that **the user's needs were met**. For this reason, the user research process was critical, and also evolved over time. Another design principle that was used is **keeping users informed of system status**, which was done by providing feedback at several points of the system by displaying steps, and what step the user was on along the way. The was also done by displaying the evaluation status while the system ran evaluations. Another important principle that was upheld, was **ensuring that user's didn't get stuck** and could go back to previous pages if needed. In Figure 5.0 for example, we see that the user is easily able to click on "Previous Step" at any point, and a similar design was done for all pages. Additionally, this experience was designed to be **flexible**, making it usable by both novices and experts by making it easy and friendly enough to be used by beginners, but also technically sound enough to allow for more complex tasks that may be carried out by expert users - such as defining new KPIs. It was also important while designing to **maintain an easy-to-scan visual hierarchy** that reflected the user's needs. We see this once again in the way the main steps are sub-steps were displayed in the high-level version of the prototype discussed in 3.4. There was also an intentional effort to ensure the design was **minimal** and the user was not overwhelmed with a lot of information. This was at times tricky to implement, because there's a lot of context behind some of the highly technical tasks that were represented with a friendly user experience. Thus ensuring that only critical information was displayed to the user was essential. The design was made to take into account the **emotion of the user/engagement of the user** along the way, with illustrations on the landing page, the evaluation process and incorporated in other pages as well. This was done to enable the user stay engaged and not get lost in the process. Using natural language to instruct the user was also done in adherence to this design principle. Lastly, ensuring that the experience **provided context/cohesiveness** was important, by showing how the user could select an application from the Deployments page, and then from there create a new app version and deploy it as shown in section 3.3. This was also done to ensure that the user is able to get the bigger picture of how elements are interconnected on the platform.

In the next chapter, we will discuss the second half of this design study, which entailed designing an experience to manage and monitor the events generated by these computer vision applications. Later in chapter 5, we will discuss limitations and insights from carrying out user research and designing both systems.

4. EVENTS MONITORING - UX PROTOTYPE ITERATION

In this chapter, we explore the process of designing a feature and system for the monitoring and management of events generated by computer vision applications, as they are being used by enterprise clients. In chapter 3, we use the smartcheckout app use-case to drive the design for the prototype. However, in this chapter, our prototype is driven by a different use-case, namely the Augmented Workers use-case as briefly discussed in chapter three. We once again review each iteration, and carry out a study, in order to ensure that the user's needs and objectives are met along the way.

Phase	**Task**	**Details**
Phase One	User Research and Ideation	Interviews (with Operators) Personas User Journey - Technician Ideation - Defining stakeholders, identifying the northstar vision, defining the day one experience for the user, and the non-day one experience.
Phase Two	First Design Iteration - Low-level Prototype and Review	Created with Balsamiq Reviewed with product team
Phase Three	Second Design Iteration - Mid-level Prototype and Study	Iteration updated incorporating feedback from the first iteration's review. Mid-level was then created around fine-tuned scenarios that were done post the product team review. Official study carried out with a group of participants to test the mid-level prototype.

Table 4.1 - Chapter Four Overview

4.1 USER RESEARCH AND IDEATION

The process of designing a feature and system for the monitoring and management of events generated by the computer vision applications being used by enterprise clients started once again by understanding who would potentially be using it and what their profiles are. The design process was done around a solution scenario of an enterprise client that provides TV and fibre installations services to people and other enterprises. Following the persona research and curation carried out in Chapter 3, the two key personas that were identified for this feature were - the Operator and the Technician *(Figures 6.8 and 6.9).*

4.1.1 - Interviews

As a recap the Technician *(Figure 6.8)* is a persona whose main needs are to successfully carry out interventions and installations, and to be able to effectively communicate with their operator counterpart while doing so, especially in the event of an issue arising. Thus, they expect to be able to take an upload photos of an installation after they are done with it, and then receive a verification from the Deepomatic system as well as the operator they're working with. Some of the pain-points identified that technicians often have to go through, is sometimes they don't have the right tools, sometimes the installations take longer than planned and thus cause a ripple effect of delayed appointments. In addition to this, they are not always able to understand how the application works or when it crashes while verifying the quality of an installation. Once in a while, they also have situations where an installation has been installed properly, but the application marks it as not well done, and they then have to call the operator working behind the scenes to manually verify the work.
The other main persona that is a part of this feature, is the Operator (*Figure 6.9).* The operator's main goal is to get through their workload of live interventions coming in, as technicians they work with are carrying out their installations, as quickly and as efficiently as possible. They also need to be able to understand the status of these operations and installations in real time. And to also be able to understand issues that arise and resolve them as quickly as possible. They also need to be able to understand the Deepomatic applications that they have in place, facilitating their business operations. Oftentimes, these operators are former technicians, and have an understanding of the work the technician has to do. The Events Monitoring feature is tailored to the Operator persona, who needs to be able to see the events that arise from the technicians using the Deepomatic app to verify the work they're doing on the field around TV and fibre installations. Thus the platform needs to be easy to use and functional for the operator, to ensure business runs smoothly.

At the moment, the solution architects who work internally for Deepomatic, are also primarily the ones that manage and monitor these events at the moment for the technician, but the goal is to have it usable by more operators that work at the enterprise clients. Thus, after identifying the user persona, it was further fine tuned by running interview calls with an operator in one of the enterprise clients, to better understand the work they do and the expectations they have. It was a qualitative interview that was conducted over Zoom. The questions asked during these interviews were:

- What work do you do at your company? What does it entail? And what are the objectives of your job?
- What is the work you do around monitoring events raised by your deployed applications?
- Who are the stakeholders involved in carrying out these objectives? Who are the stakeholders you communicate with the most?
- What is the process of working with a technician? How do you communicate/work with them? In what scenarios do you have to? And how often do you have to do so?
- What are the most common problems you experience and have to resolve in your daily work with managing events?

- What tools/software do you currently use to carry out these event management/monitoring tasks? What problems do you currently encounter with these tools?
- What would make your work easier?

From these calls and interviews, we were able to gather the following insights from the user's responses to the questions we asked, and feedback we got during conversations with them:

- They need a way to view all the events being generated and to be able to understand these events and their attributes. An event is generated when a technician is carrying out an installation. Each event corresponds to an installation by a technician, which Deepomatic also refers to as an intervention. When an installation is carried out, it often comes with relevant data and attributes to all stakeholders involved, such as the type of the event (categorized by the client enterprise. For instance, was the event an TV installation, or a Fibre installation and so forth), photos taken by the technician after an installation is complete, the date it was carried out, if the installation/event is live or not, meaning if the technician is currently on site carrying out the intervention or not, if it is not live, the date the installation was carried, the location, if an anomaly was detected, and so forth. All of these information that tells us more about what is going on with an event, are referred to as an event's attributes.
- They need to be able to keep track of what events are open, being processed and have been resolved and thus closed.
- We got to understand what type of events and installations they usually track and care to have more information and insights on.
- It is urgent they have a clear way to view these events being generated as they occur in real time, while the technician is still on the field and is able to take action.
- They would need to have a way to easily verify that the Deepomatic system works efficiently, at any moment.
- They need a way to keep track of the technician's busy periods to better understand down moments when they aren't receiving as many interventions.

4.1.2 - Personas

Below, we explore the specific personas that are involved in implementing the event monitoring features.

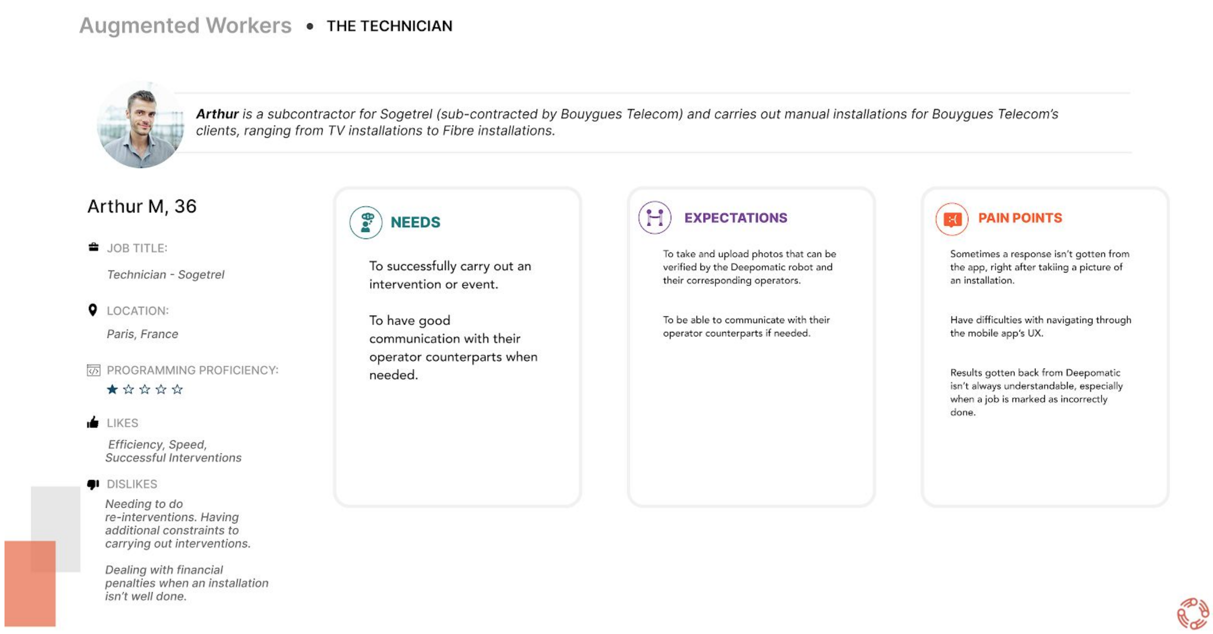


Figure 6.8 - The Technician Persona

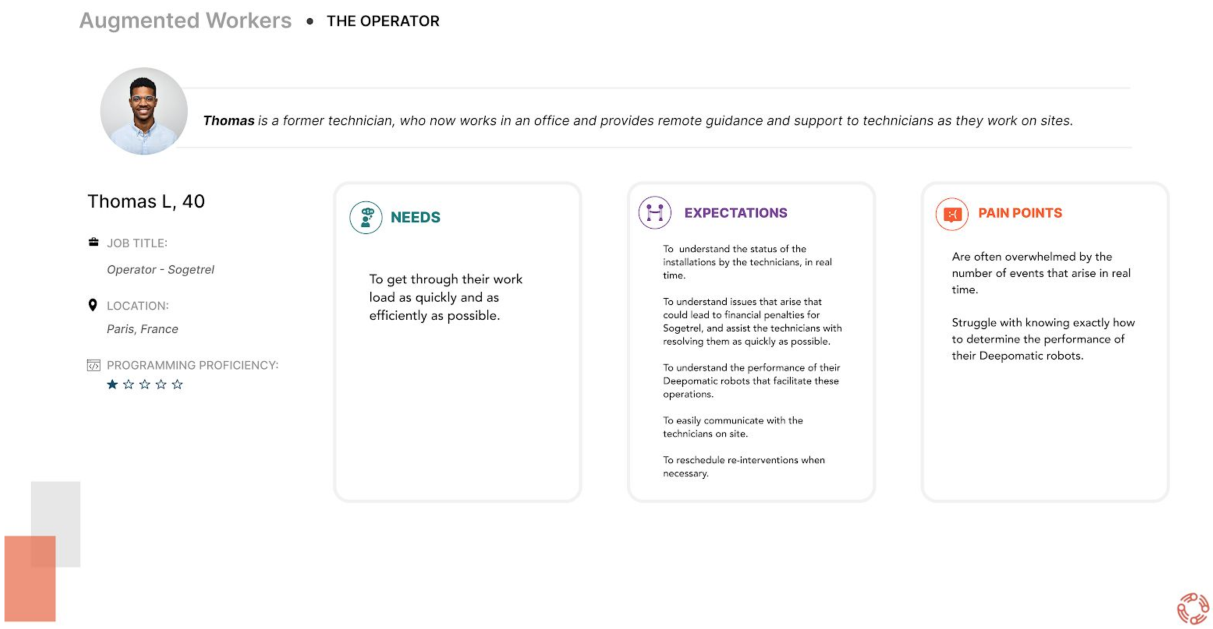


Figure 6.9 - The Operator Persona

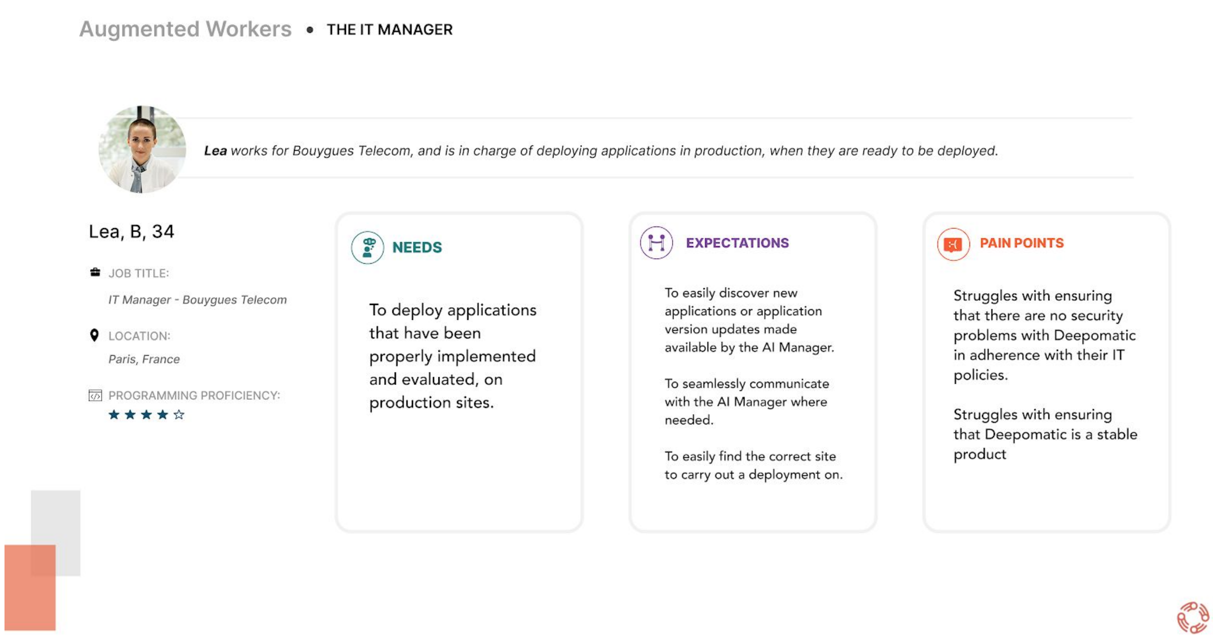


Figure 7.0 - The IT Manager Persona

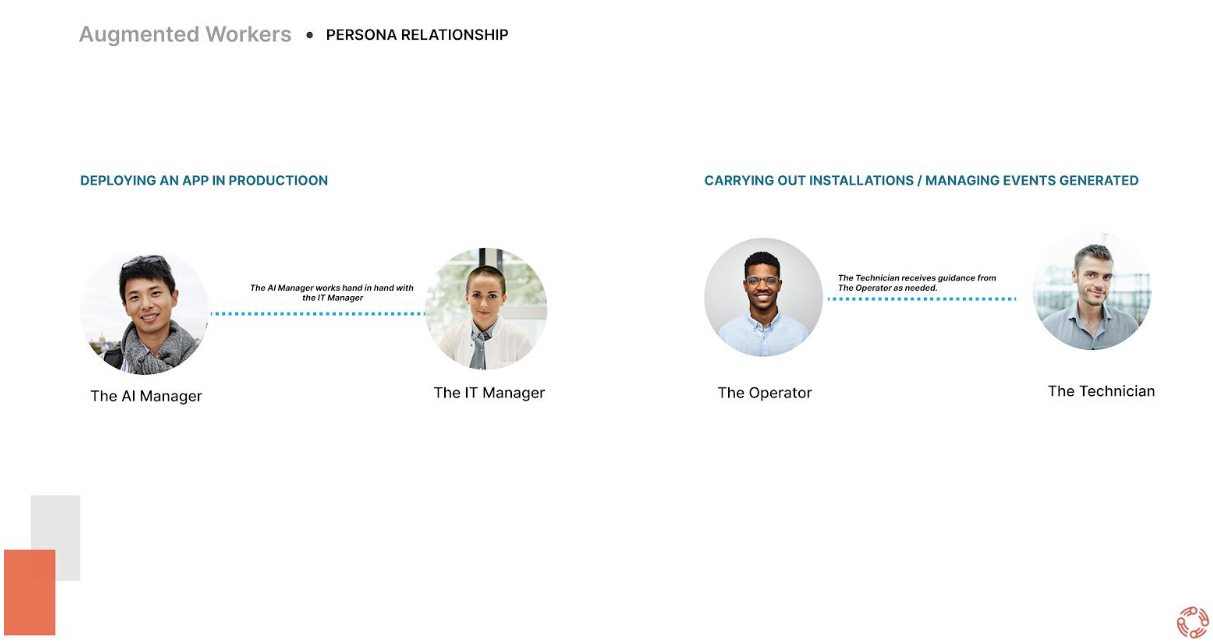


Figure 7.1 - The Persona Relationship for Augmented Workers

In *Figure 7.1,* we see how the personas related to the Augmented Workers use-case, all relate with each other, to carry out a set of tasks they need to. These augmented workers personas, the AI Manager persona and the Solution Architect persona were primarily used in designing the Events Monitoring use-case.

4.1.3 User Journey

Here is how the user journey for the technician was illustrated:

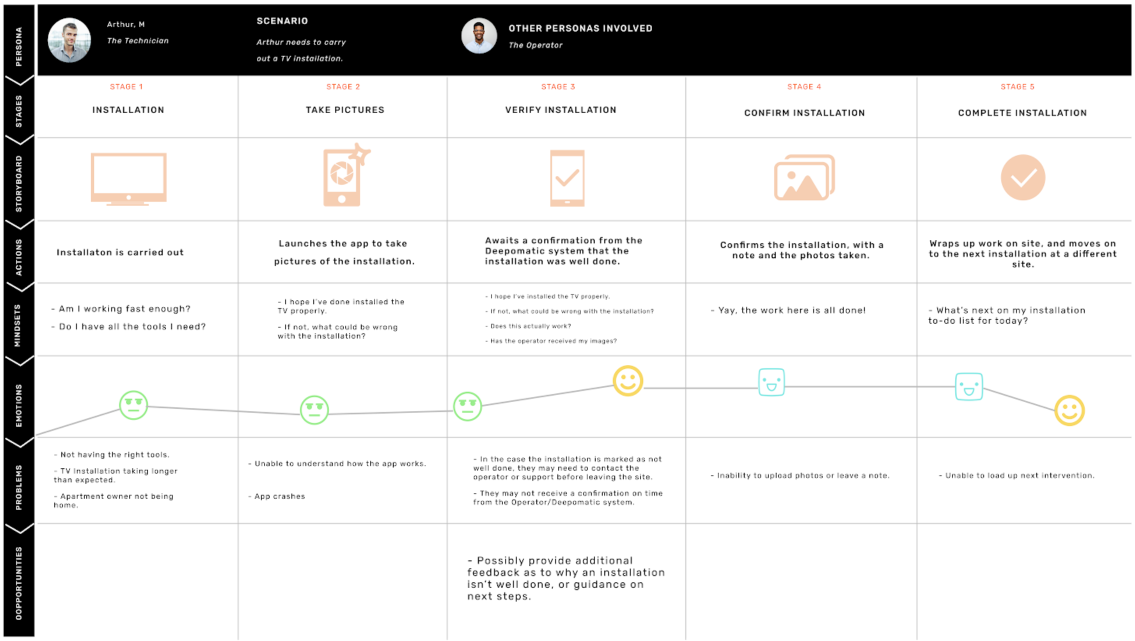


Figure 7.2: User Journey of the Technician

In Figure 7.2, Arthur, the Technician persona starts off the overall process for the Event Monitoring feature, by going through a number of tasks in order to get his work done. He first carries out the installation that could be a fibre installation, then he takes pictures of it to ensure quality control, which is verified by the Deepomatic computer vision application. Then, he confirms that the installation has been carried out, and then the installation is officially complete. We see that certain factors are highlighted in the user's journey along the way, such as the emotions he feels along the way, as well as the problems he encounters. All of these informed the design process, and gave us personas to their user journeys to keep in mind, while brainstorming and iterating on the design. After the technician's user journey was illustrated, the operator's, operations manager and other secondary personas involved in the overall process were illustrated as well.

4.1.4 - Ideation

The ideation process started by breaking down the design process into smaller bits, starting with designing a way to display the events being generated and their attributes clearly. This was starred as the first priority to building the overall feature. Once we had an idea in the form of a low-level prototype made in Balsamiq, of how the events and their attributes would be displayed, this was all verified with the internal product team and iterated upon. There were also brainstorming sessions held, keeping in mind the user journey of the operator, and how collaborative the operator's experience could sometimes be, in the sense that they sometimes work simultaneously with other operators, to resolve all of the live events coming up as quickly as possible. The low level prototype was created to cover these core scenarios, namely:

- Managing the attributes of the events and how they are displayed on the dashboard.
- Carrying out bulk and single actions on events. Such as deleting an event(s), or reassigning it to another colleague.
- Searching and sorting through events.

We will explore how these scenarios were implemented across different iterations, and the lessons learned through reviews and studies along the way.

4.2 FIRST ITERATION - LOW LEVEL PROTOTYPE AND REVIEW

The very first iteration was created on paper and also using a software called Balsamiq, which is tailored towards low-level designing. The screens were created with the scenarios discussed in 4.1b, starting with viewing events and their attributes, which tell us more about an event.

Viewing and Managing Attributes

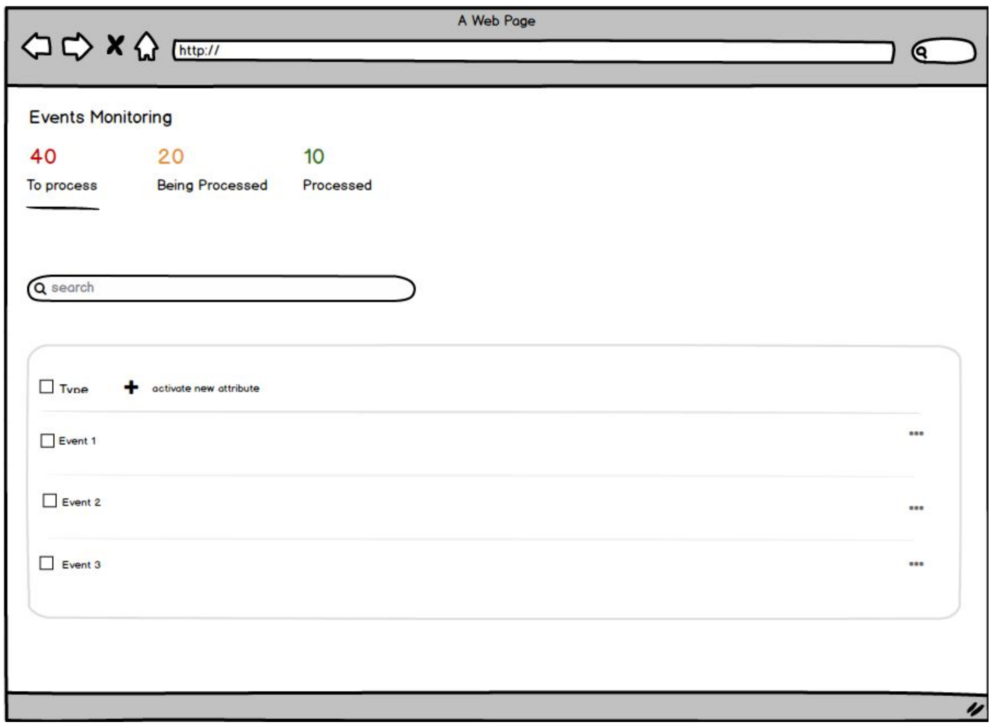


Figure 7.3 - Landing page of events monitoring

In Figure 7.3 we see that the dashboard displays all of the events based on the Type of the event, thus only the "Type" column is displayed here. There's also a way to search through all of these events, and these events have been separated based on what events are yet to be processed, what events are being processed and which ones have already been processed and closed. Notice that there's also an "activate new attribute button" which the user clicks on to view more attributes and information about the event, and then they can choose whether or not to activate this attribute, meaning post activation the attribute will be displayed as a new column on the dashboard (Figure 7.4 a and 7.4 b).

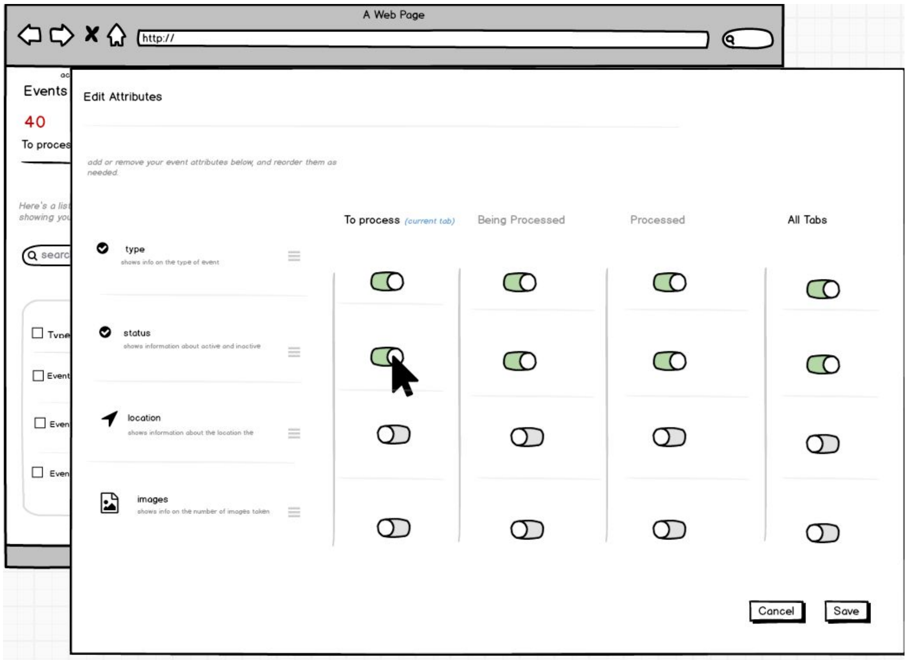


Figure 7.4 a - Activating attributes

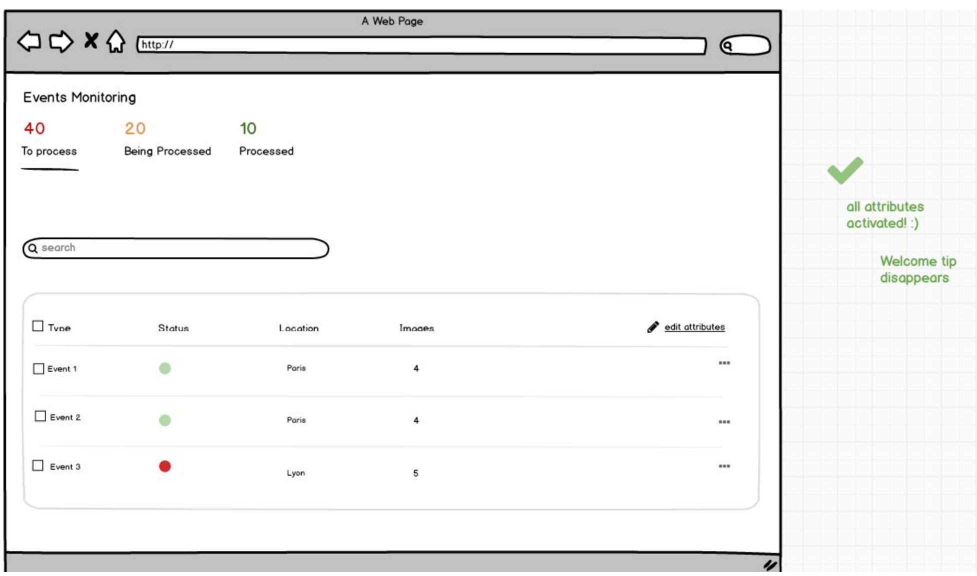


Figure 7.4 b - Viewing the activated attributes

Once the user clicks on "activate new attribute" they're able to view a list of attributes they can activate and then view on the dashboard, such as the event's status, the location where the event is taking place, and images that are attached to the event (that is, images of the installation being done). Here the user can simply activate these and also specify if they would like the events to show on all tabs (To process, Being Processed and Processed), just one of them in particular as seen in Figure 7.4a. In Figure 7.4b we see that the activated attributes now show on the dashboard. The user is able to deactivate an attribute if needed from this page, as seen in Figure 7.5, and they're also able to reorder the positions of these attributes as needed, as shown in Figure 7.6a and b. Notice that from Figure 7.6 a to 7.6 b, a column is selected in 7.6 a and then dragged over to the location the user wants it to be in, in 7.6 b.

Figure 7.5 - Deactivating a single attribute

Figure 7.6 a - reordering the position of attributes

Figure 7.6 b - reordering the position of attributes

Carrying out Event Actions

Here, users are able to delete a single event or carry out bulk actions if needed. An event may for instance, need to be deleted if it contains incorrect information wrongly detected by the computer vision system. These actions buttons were incorporated based on the user research carried out based on the need for flexibility with management of the events.

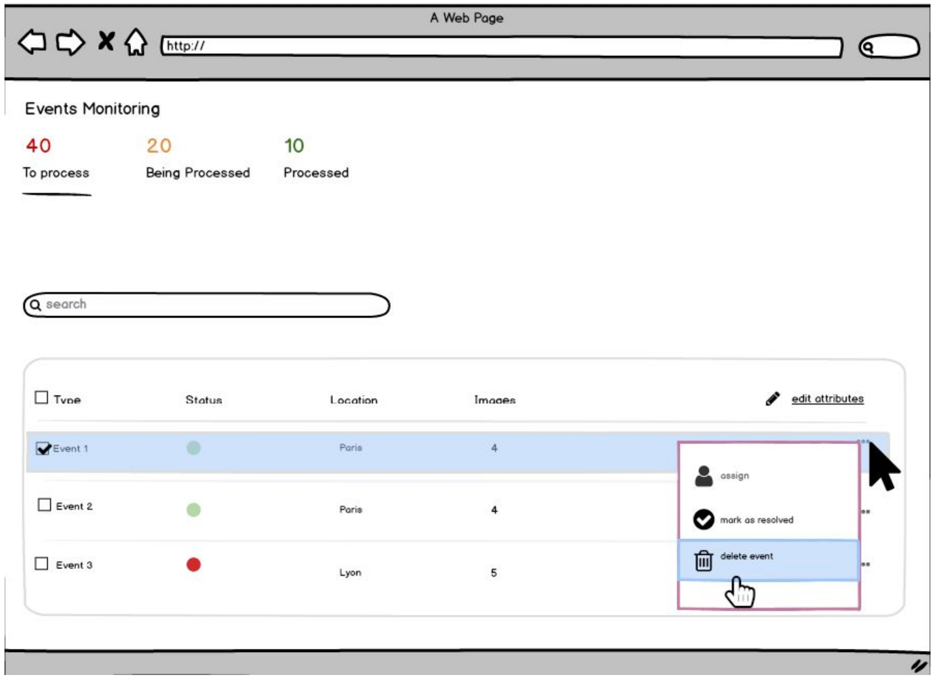


Figure 7.7 a - deleting a single event

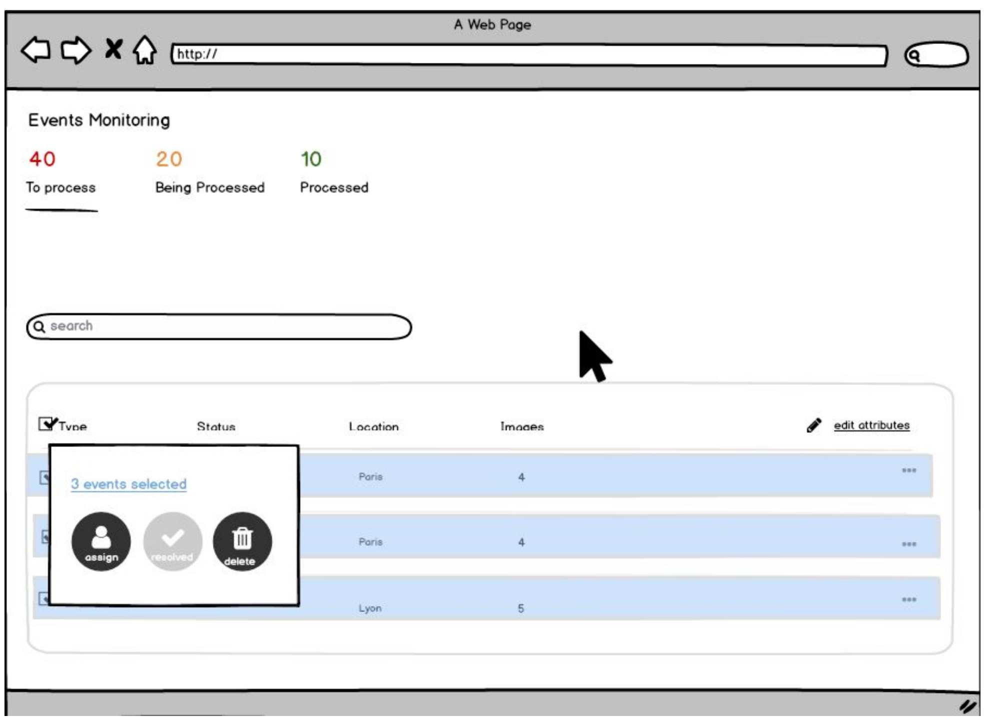


Figure 7.7 b - carrying out a bulk action on all events.

Searching and Sorting Through Events

In this part, we explored what the experience of search could be. The user is able to simply use the search bar to type in for instance, the title of an event as shown in Figure 8.0, where the event title "RACC" is being searched for, and post the search, the search text would remain in the search bar so the user is aware at all times of where they are in the searching process. Similarly, while sorting the list of events by the ascending order to the location of events (location attribute) for example, the platform also highlights this by storing what attribute the events have been sorted by, as shown in Figure 8.1 a and b.

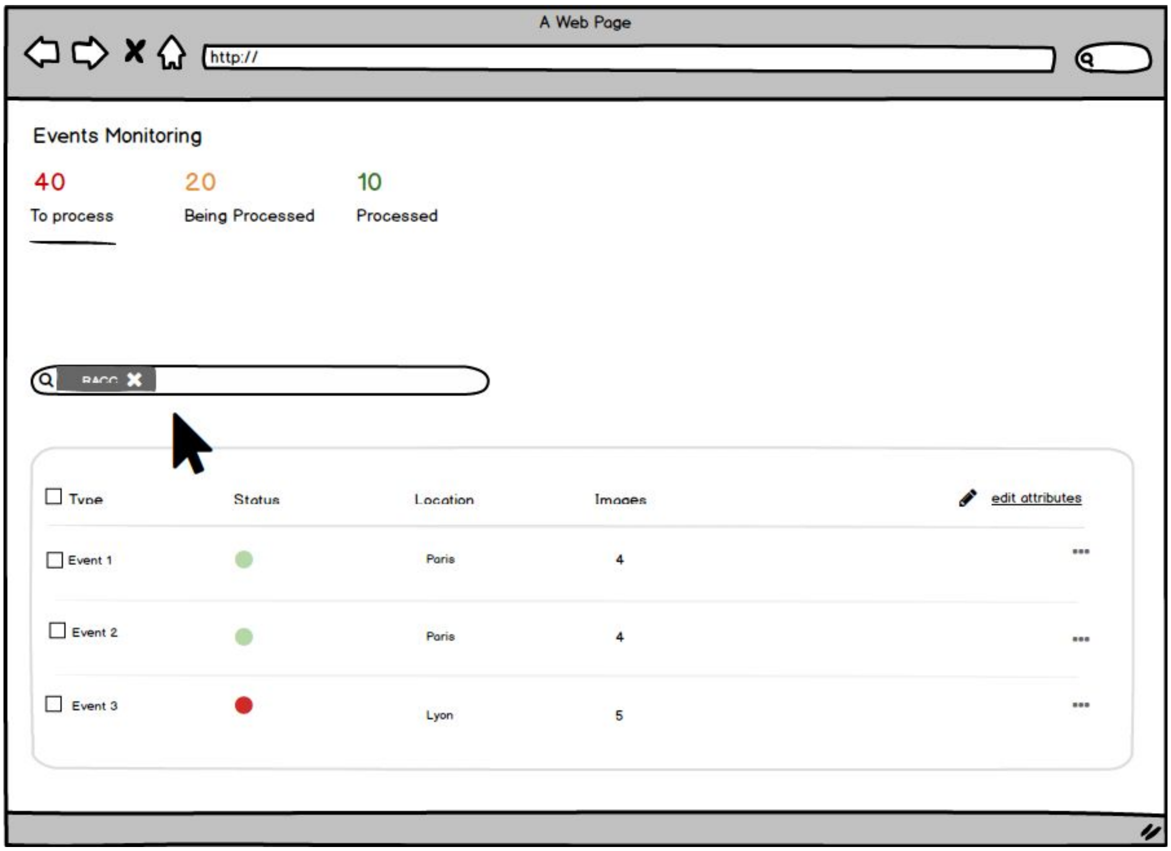


Figure 8.0 - Searching through events.

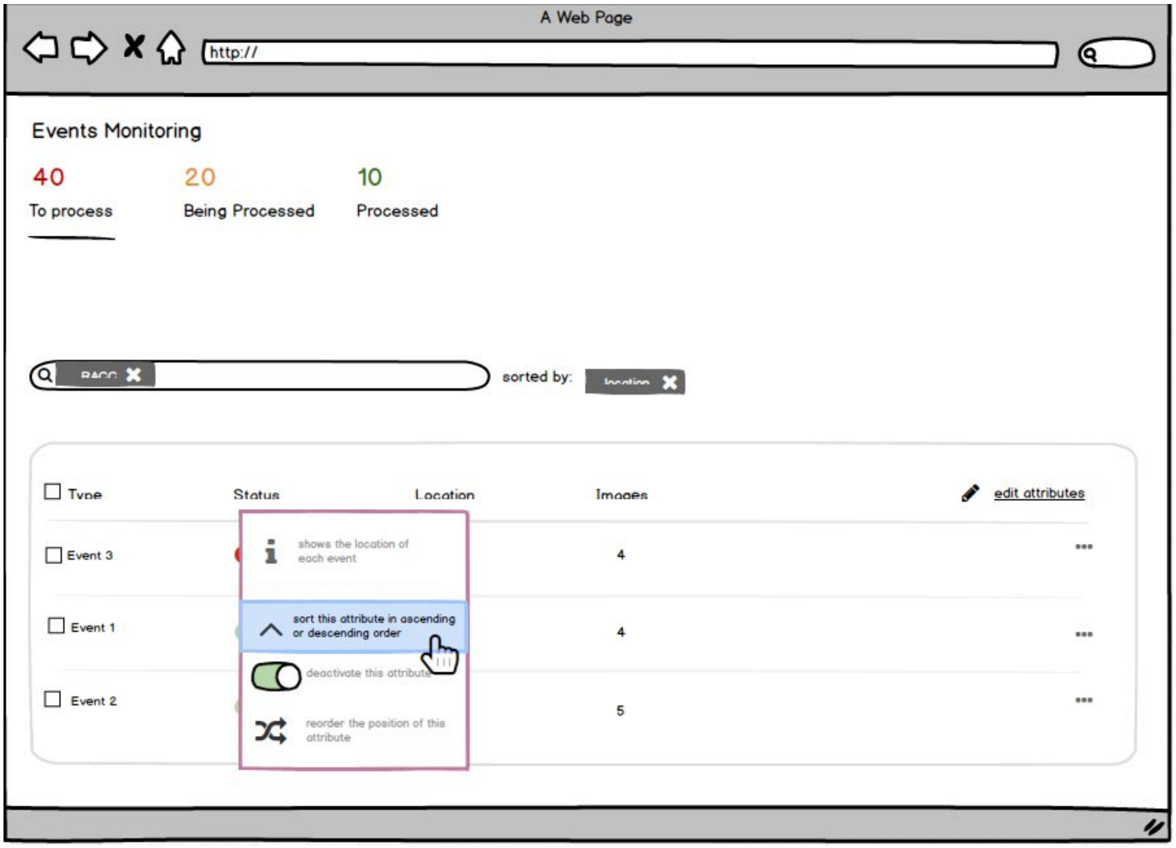


Figure 8.1 a - Sorting events

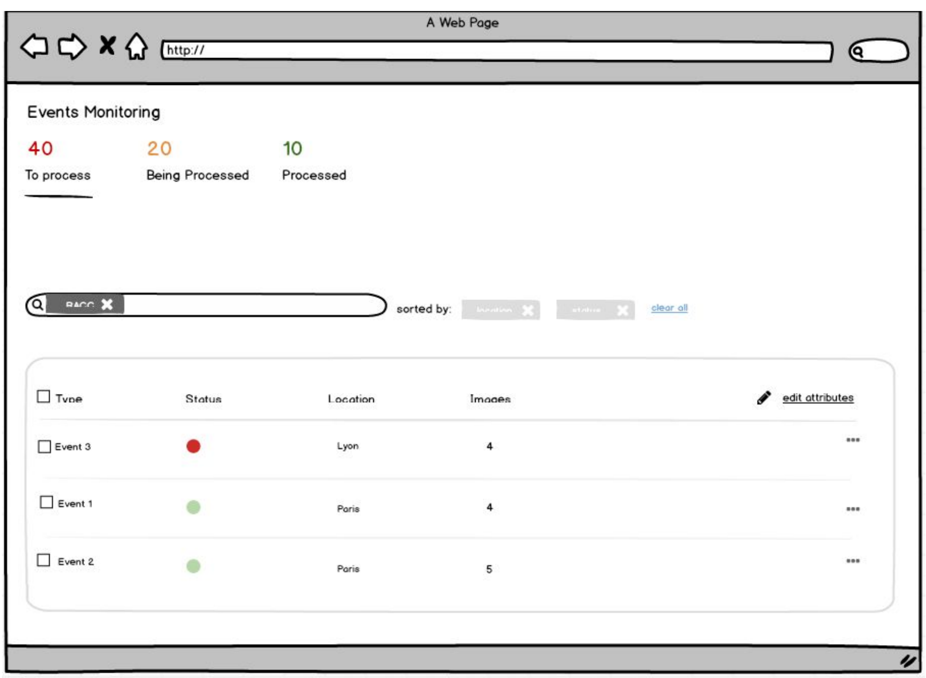


Figure 8.1 b - Sorting events

These starting slides were created in conjunction with paper sketches, to present some of the ideas that were later to be translated into higher definition design for the mid-level process. These were then reviewed with the product team, comprising a product manager and a front-end software engineer who also works as a product designer at Deepomatic. During our meetings, an approval was received to move this from Balsamiq sketches to the second iteration of the designs. The product team also gave more information about what some of the main tasks the personas involved in this feature would typically have to do, which will be discussed and explored in the following section 4.3. The suggestion was also to curate the Figma prototype around these tasks. Thus, the second iteration and mid-level prototyping process began.

4.3 SECOND ITERATION - MID LEVEL PROTOTYPE AND STUDY

Once again, this mid-level was built around tasks that participants would ultimately need to carry out during the study, and that are in line with the typical tasks that are often carried out while using the Events Monitoring feature.

Task One

In the first task, the user simply has to search through all the events, with a specific event ID, similar to the experience shown in Figure 8.0, but in Figure 8.2b, we see an upgraded version of this experience.

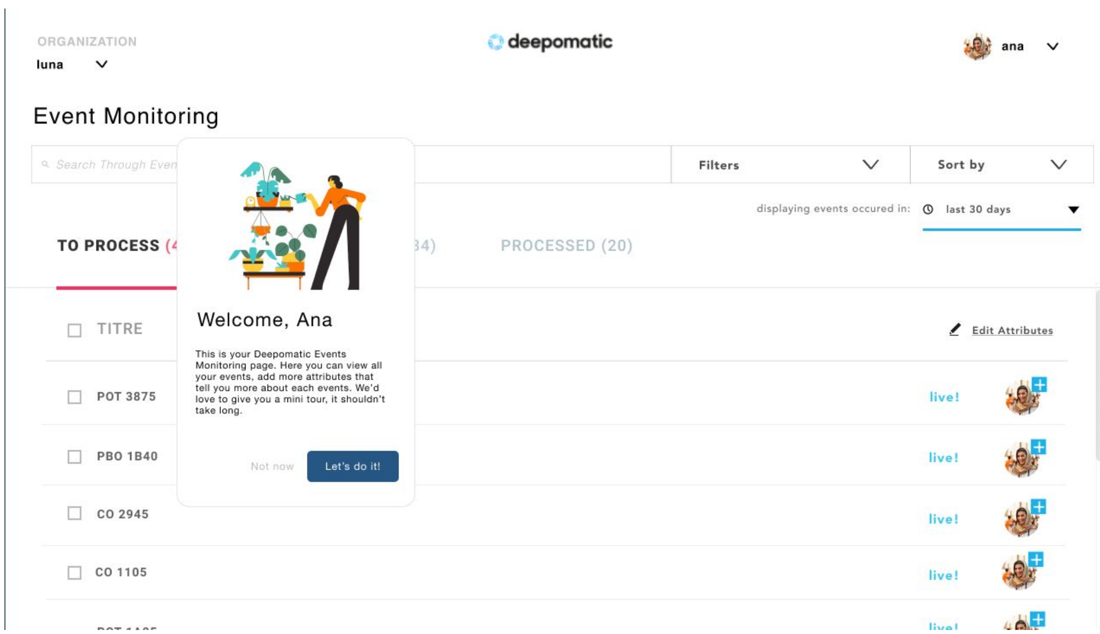


Figure 8.2 a - Landing page

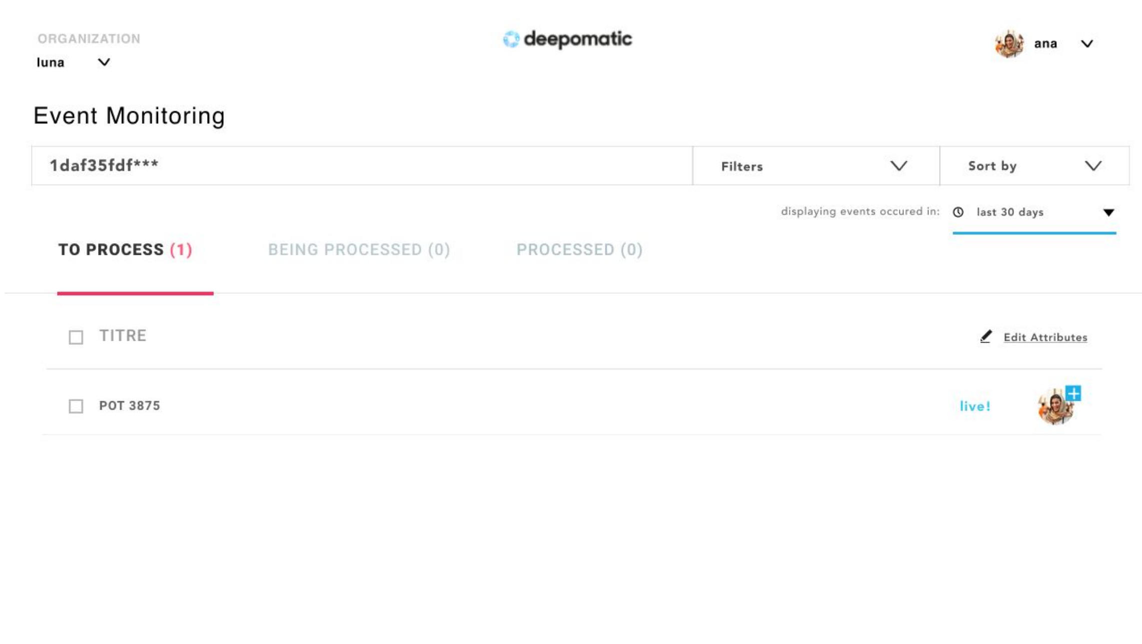


Figure 8.2 b - Searching with a specific ID

As seen in Figure 8.2 a, the landing page was made to be more user friendly and more instructional with a welcome pop-up guide. This was done to provide more guidance to the user while interacting with the dashboard. The user, in this case Ana, who is an operator, is also able to right away view all events happening live, which are displayed first on the event list with a sign that says “live”. On the top right, in addition to the user being able to see the events, they are also able to filter through the events based on a timeline. The search bar was also placed above the tabs - to process, being processed and processed, so that the search process is easier for the user and once they search through an event, all tabs are filtered accordingly. We see that this is what happens when the user carries out a specific search with an event ID, as the user views only one event present under the *to process* tab, while the others are left empty as shown in Figure 8.2b.

Task Two

In the second task, the user has to carry out a different type of search through the events that would typically be carried out while an operator needs to investigate the quality of an installation that was done in the past. In this case, the user needs to find events that took place in Bordeaux, on the 28th of March by a technician called Guillaume J. In order to do this, the operator first activates all of the attributes of the events being displayed that are needed to carry out the investigation, on all of the tabs. Namely the technician attribute, the location and date as needed for the investigation they are looking to carry out (Figure 8.3 a). Once the attributes/columns have been activated, the user is then able to filter through this list based on the location being Bordeaux and the Technician, Guilllaume, as shown in Figure 8.3b. In Figure 8.3c, they choose the exact date they’re searching for - March 28th, and in 8.3d, they’re able to view the results of their search and filtering, thereby completing this task.

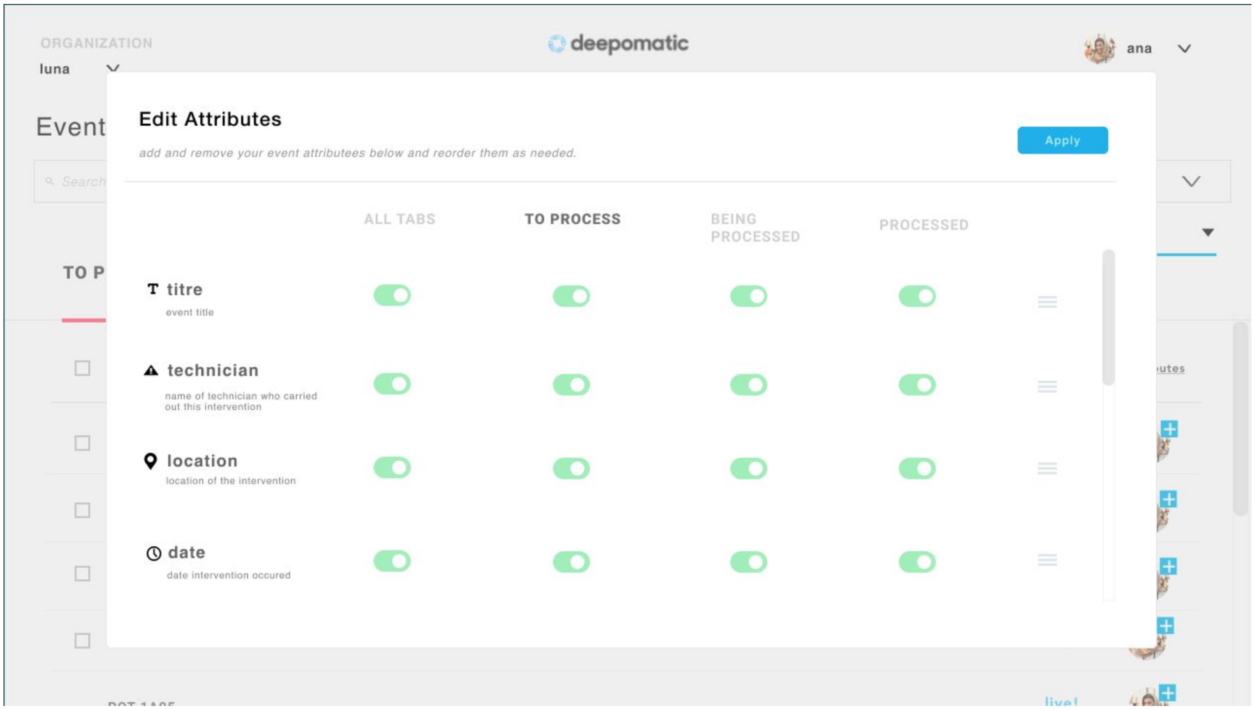


Figure 8.3 a - Activating Attributes

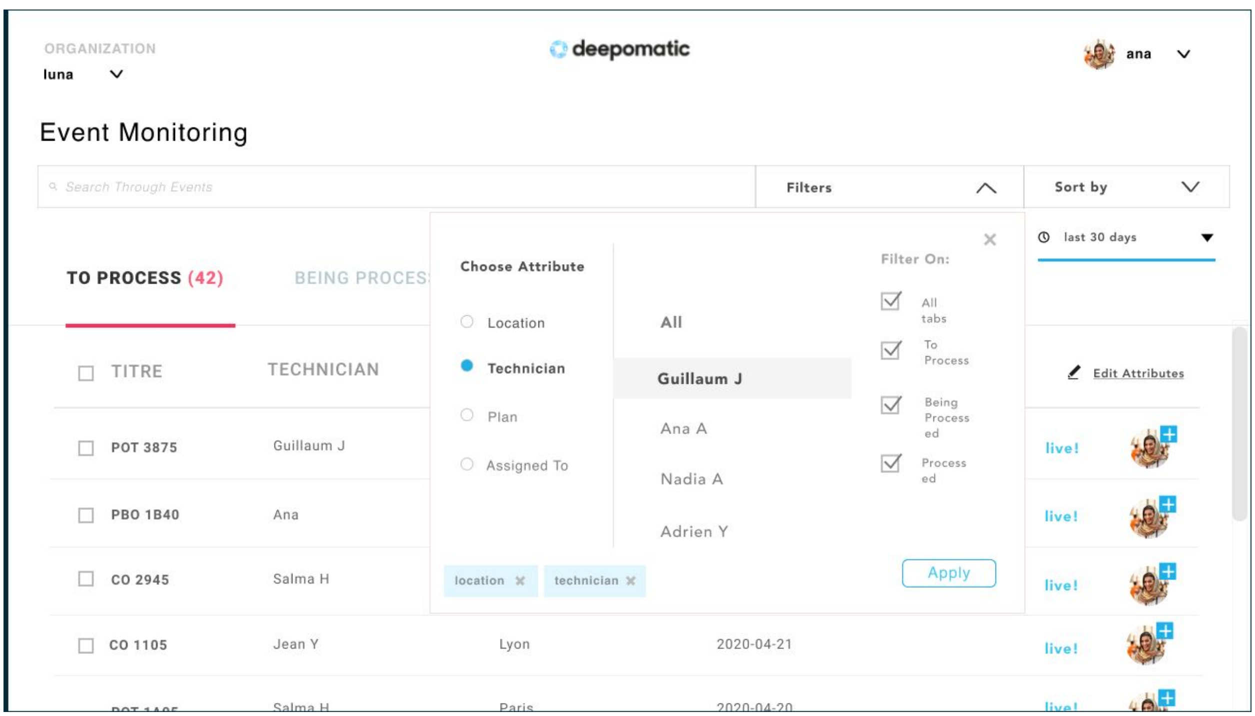


Figure 8.3 b - Filtering through events

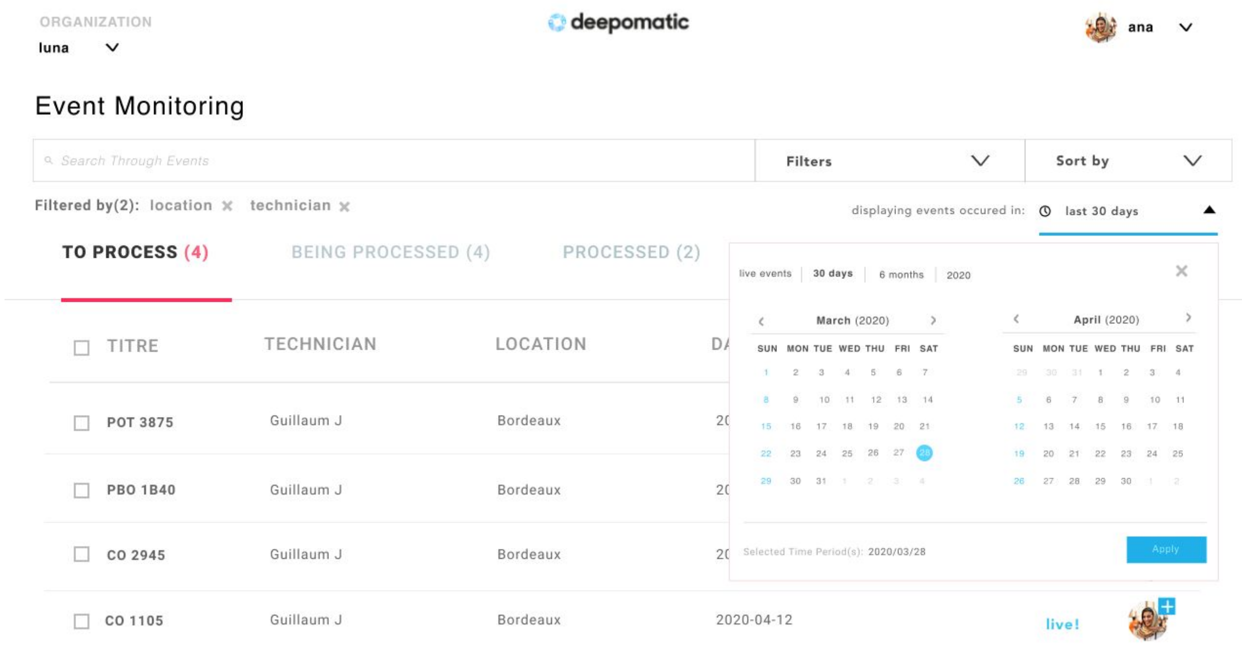


Figure 8.3 c - Specifying the date

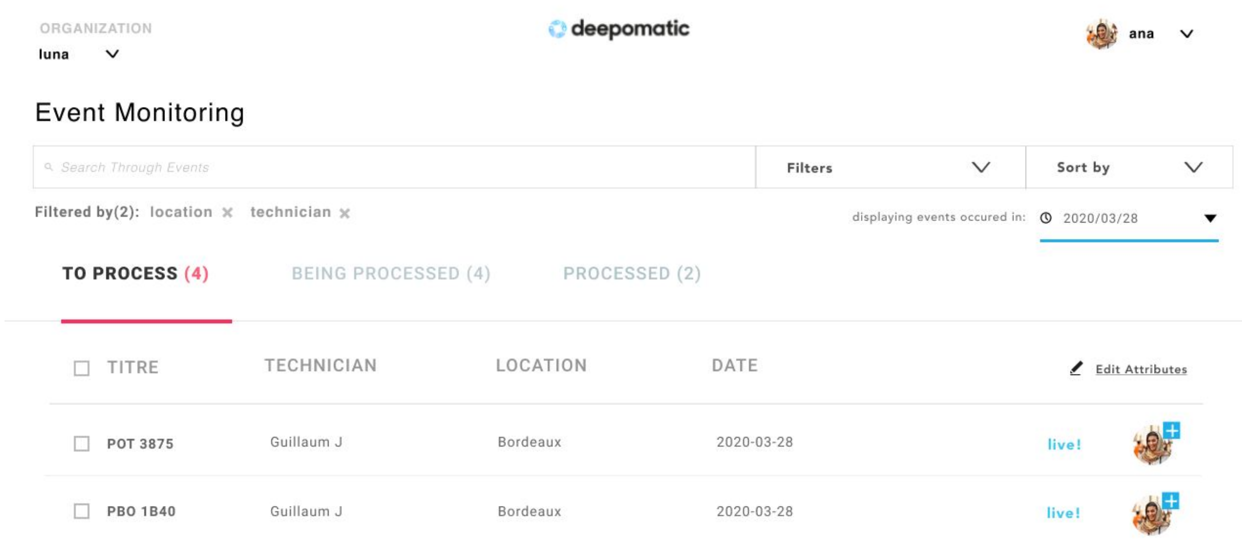


Figure 8.3 d - Viewing search and filtered results

Task Three

In the third task, the operator realizes that there was a job done the week of the 9th of March that wasn't well done. However, it was marked as well done by the Deepomatic system. Now, the operator has to once again sort through all events done the week of the 9th of March, and look through the events that are currently being processed by the operator, then find the TV installation event, look through to see what TV installation wasn't well done, and then change the status of the event from well done (OK) to not well done (KO). This task was also chosen as it's a core scenario that an operator sometimes needs to resolve. The user carries out this task by first narrowing the list of events being shown to all events that occurred on the week of the 9th, as shown in Figure 8.4.

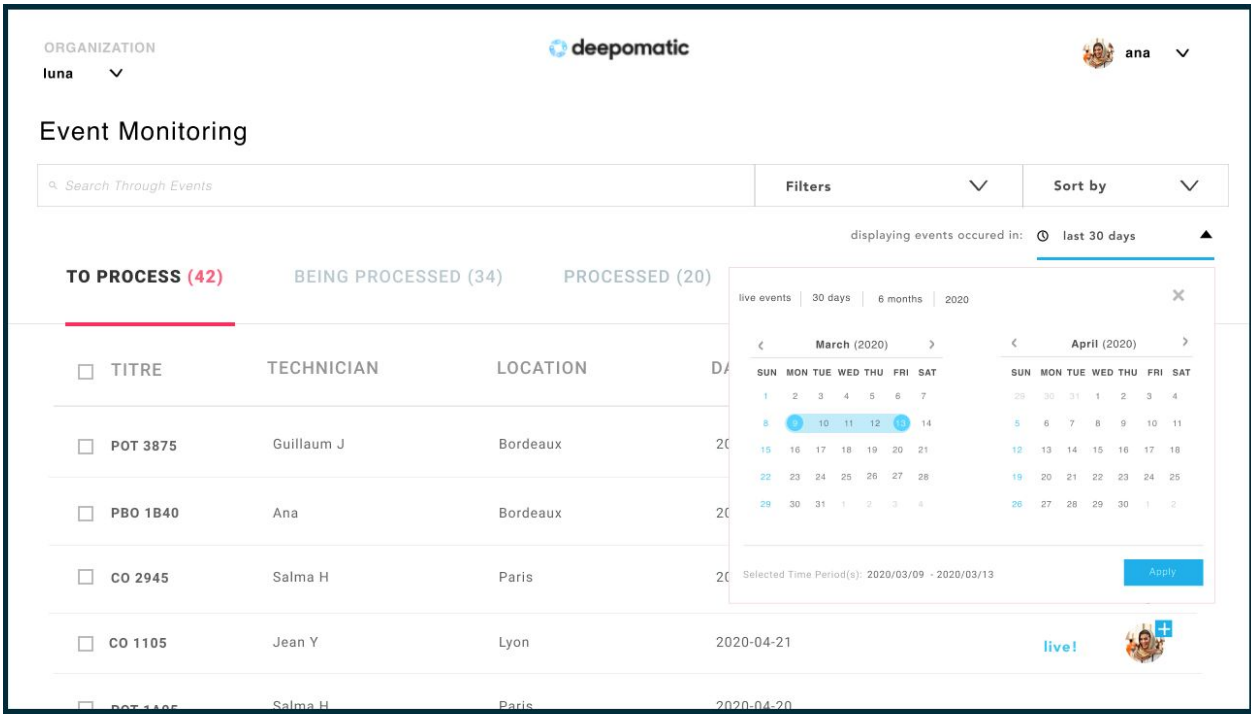


Figure 8.4 a - Viewing events that occurred on the week of the 9th.

Afterwards, the user can view all the events being processed by them, as shown in Figure 8.4 b. They see the title of the event, what technician carried out the installation, where it occurred and on what date.

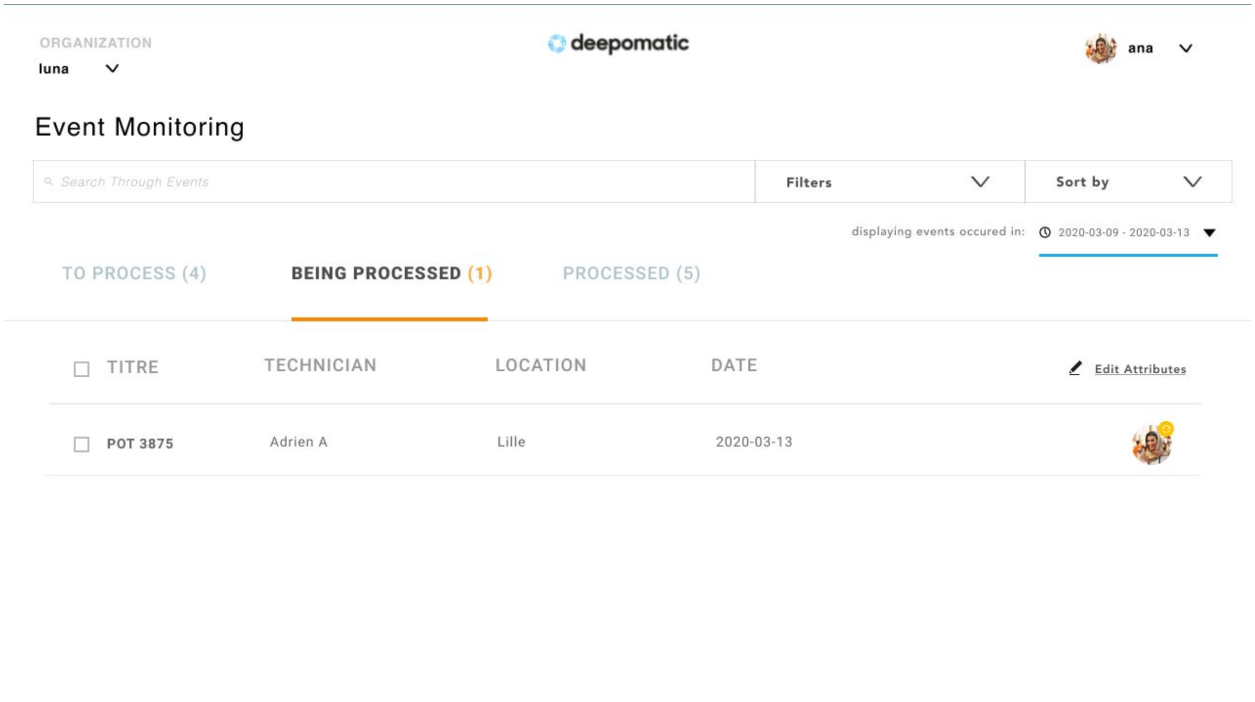


Figure 8.4 b - Viewing events that occurred on the week of the 9th.

The operator is then able to launch more details about this event by clicking on it. On the Event Details pop up, shown in Figure 8.4c, they're able to see the photos that were taken during the installation, and then change the status of the event from OK to KO, as shown in Figure 8.4d. Thereby closing off the task.

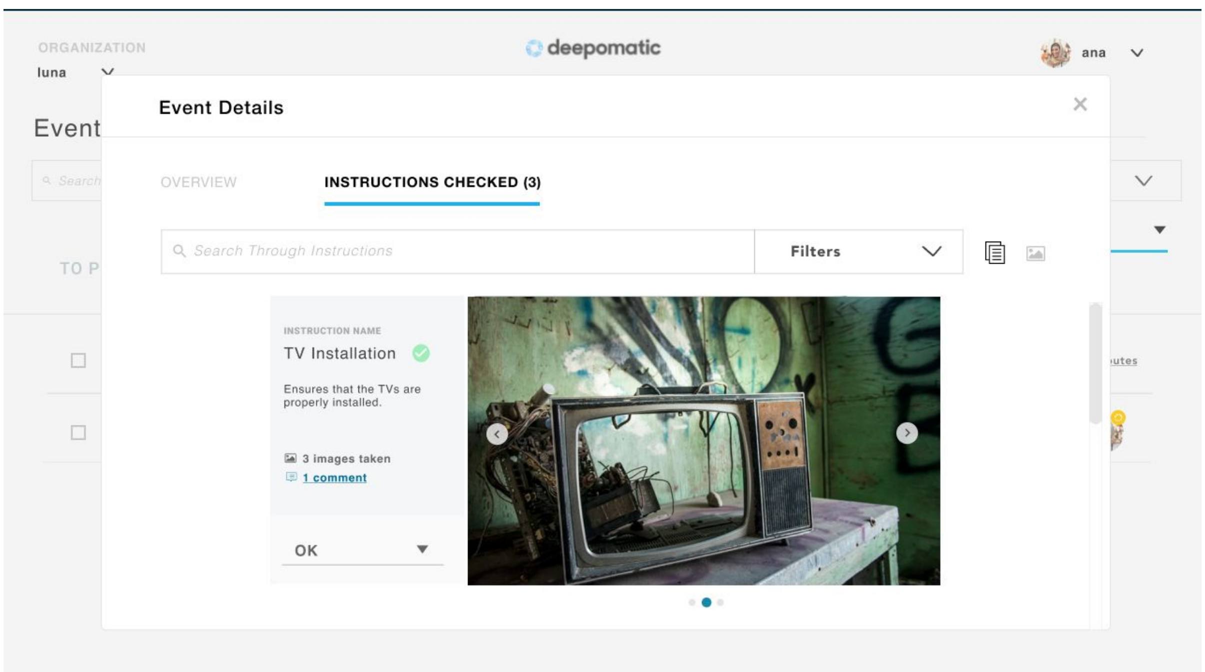


Figure 8.4 c - Viewing the Event Details of an Event

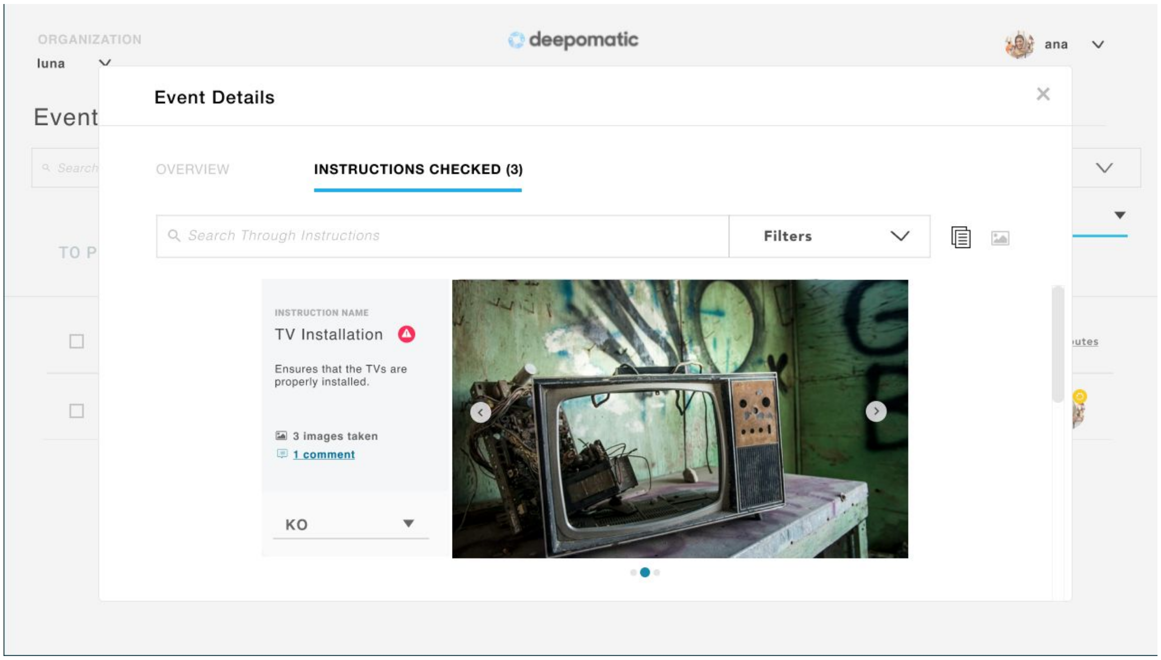


Figure 8.4 d - Changing status to KO from OK.

Task Four

In the fourth and last task, we wanted to test out the usability of the feature during a more complex scenario, where the operator also has to engage in collaborative work with other operators. Here the operator has to find an installation/event that is currently happening on the field and is being handled by one of their colleagues - Jean P. This event has an anomaly that has been detected by the technician, and thus the operator has to find all events currently happening, thus live, that are being handled by their colleague - Jean P. Then they have to find all interventions/events with an anomaly detected. Next, they view the comment made by the technician and hand it over to another colleague Lisa, who is in charge of rescheduling installations so the technician can come back on another day to resolve the issue. Once again, after filtering in a similar manner in the previous task, to get to all events being handled by Jean P, that are live, and have an event status of - anomaly detected, the operator is then able to view the event detail, and the comment made by the technician (Figure 8.5). Then they go ahead and reassign the event to be handled instead by Lisa, J, as shown in Figure 8.6 a and b.

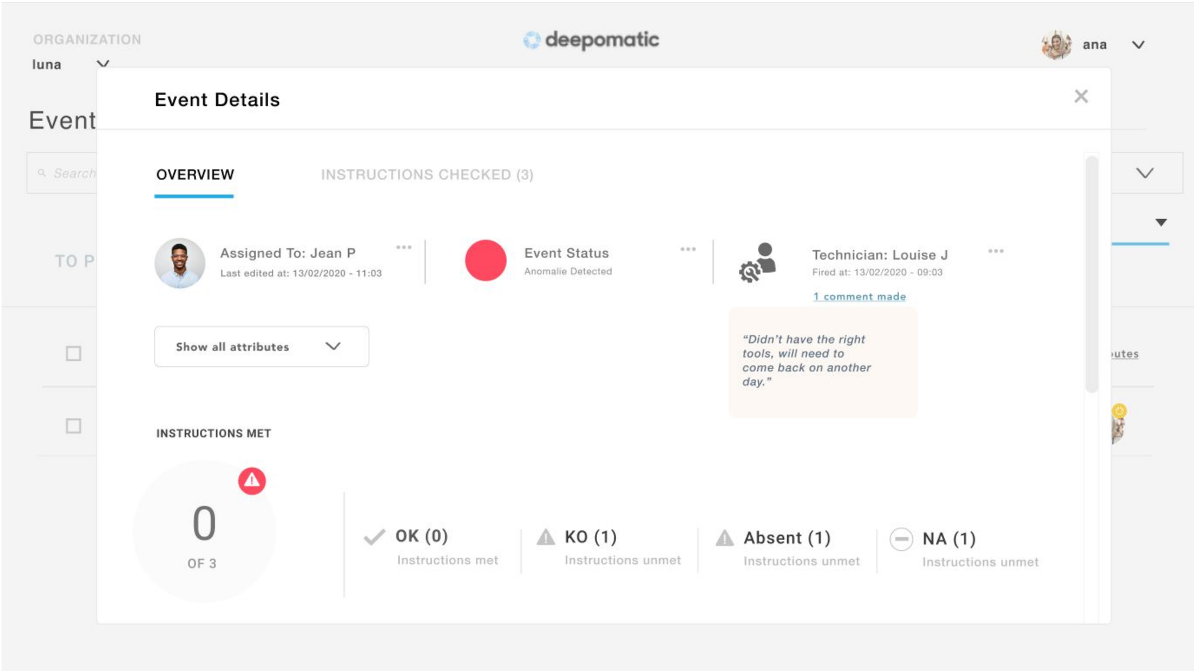


Figure 8.5 - Viewing the Technician's Comment.

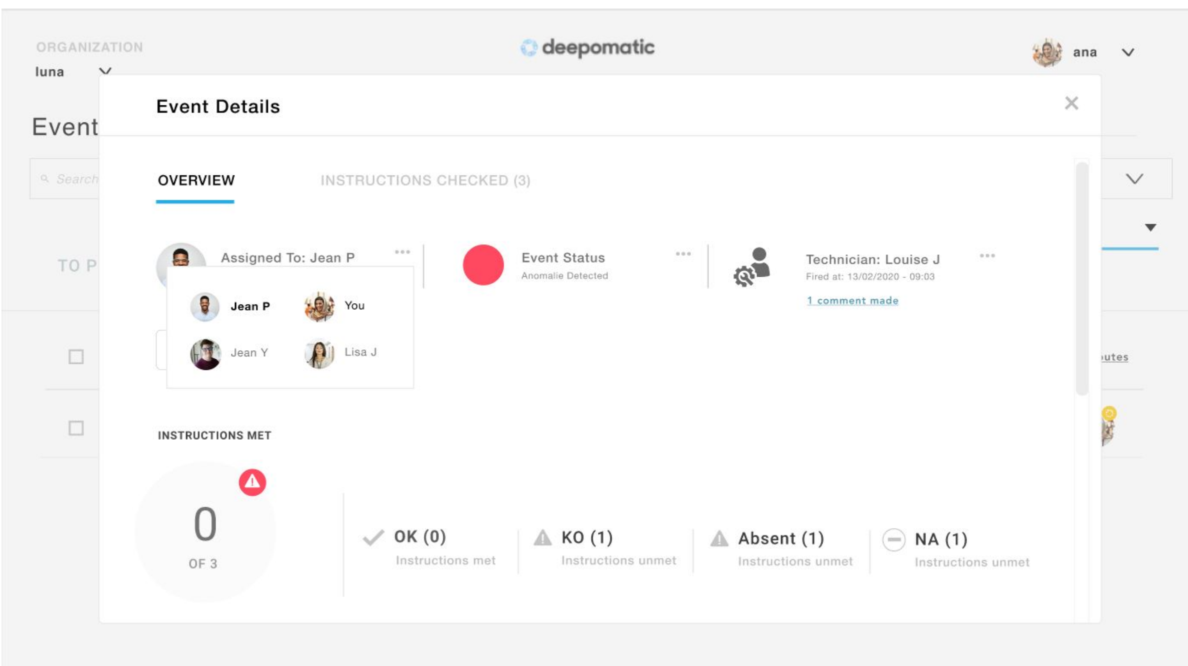


Figure 8.6 a - Reassigning the Event

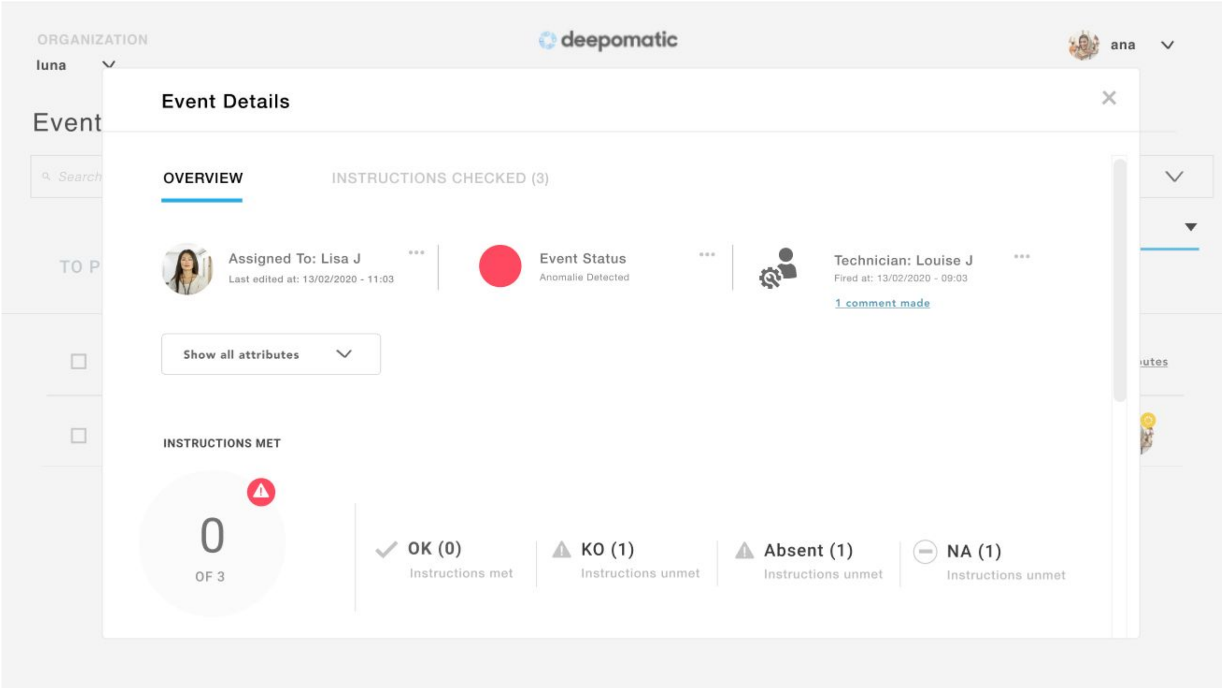


Figure 8.6 b - Reassigning the Event

4.3.1 - Study

Goals

After finalizing the second iteration of the prototype, a study was carried out following these goals:

- **Test Feature Usability:** Events Monitoring as explained above, is primarily used by internal solution architects that work for Deepomatic and for this iteration we focussed on having it tested by 2 solution architects as well as a research engineer who works for Deepomatic and wasn't as familiar with the feature. This gave us a chance to receive feedback on the usability of the design from a range of user profiles, and to better understand their needs.

- **Reveal Points of Friction:** Pointing out the confusing experiences was a crucial element for the overall success of the EVents Monitoring feature, the study was carried out to reveal what part of the experience caused confusion and difficulty with completing the tasks.

- **Understand Key Scenarios:** While users worked through tasks, we had the opportunity to learn more about what scenarios were important and relevant to them, and which ones weren't, and the nature with which they approached carrying out these scenarios.

Methodology

The participants consisted of two women, both solution architects at Deepomatic, and a man who was a research engineer at Deepomatic. Figma was used to create and present the prototype that participants carried out the tasks with. Some of the metrics we collected were:

Time - the time it took to complete a task. This was recorded by keeping track of how long it took the participant to complete a task, starting from when they started reading what the task entailed, to when they reached the end of the task.

Number of non-blue clicks - Figma provides guidance while working through a prototype, so users know where to click on the prototype in order to advance to the next frame. They do this by finding blue borders around the clickable areas, and clicking on them. By measuring the non-blue clicks, we counted the number of clicks made by the user on the page, before they noticed and followed the blue border guidance indicated by Figma. A high number of non blue clicks, indicated that the user found the task confusing.

Satisfaction Rate: This metric kept track of how users felt about the task, and how easy it was for them to complete. The questions that I asked and were discussed at the end of the study were:

- How would you describe the experience you had overall with the prototype?
- On a scale of 1 to 10, how would you evaluate the ease with which you understood the tasks?
- What was your favorite aspect of the prototype?
- What surprised you during? What frustrated you?
- Do you have any additional feedback for us?

Procedure

Both the study and the qualitative interview afterwards were both conducted online by myself and my supervisor using Zoom, which is a video call conferencing software. Participants joined the Zoom call, and were also told to complete the four tasks centered around finding and resolving interventions/events. The task increased progressively in complexity and built on top of each other. After they completed each task, they were able to theen advance to the next. While participants went through the tasks, myself and my supervisor guided their experiences by answering questions they had along the way, and probing them with questions in order to better understand why they were making certain decisions along the way. For example, when a user clicked on a button we didn't expect them to click on, while completing a task, we asked more questions to find out why they did, or what they expected to see on the page. In addition to asking the participants questions, we also kept track of the metrics mentioned above.

Tasks

As a recap from above, the tasks the participants had to do included:

- Finding a specific intervention using a given Event ID
- Finding an intervention based on its attributes - where it took place, the date, and what technician handled it.
- Resolving a TV installation that was done wrongly, and occured in the past, by changing the status of the TV installation intervention.
- Resolving a live intervention, by finding it based on its attributes, and then re-assigning it to a different colleague.

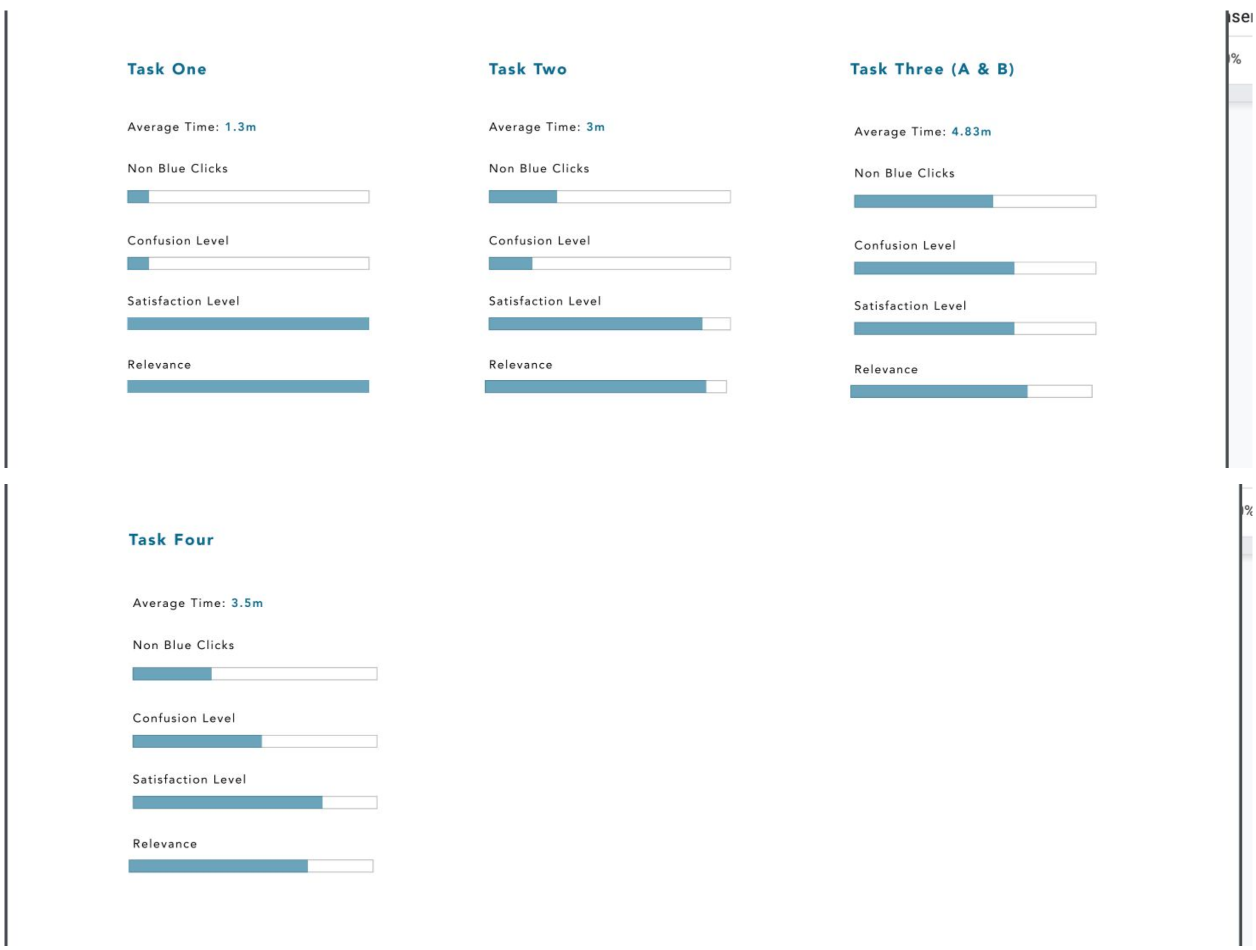
Once again, these tasks were chosen in line with some of the key scenarios operators often have to carry out, and issues they often need to resolve, in order to keep processes running smoothly.

Results
Data and feedback were collected and analyzed after the tests were completed by the participants. The test was carried out with participants that fell into two categories, and the categorization was made depending on the participants' familiarity with the Events Monitoring feature. *Category 1* included Solution Architects who had a strong background whereas *Category 2* included the Research Engineer with less experience.

Overall
The Good - All three participants were overall highly satisfied with the experience and the UI design, giving the experience and the UI design ratings out of 10 - 10, 8.5 and 8. They found the flow of the test and the tasks easy to follow and understand.

The Not so Good - 66% of participants (⅔) struggled with understanding how to carry out the third and fourth tasks, which included viewing an intervention in detail and carrying out specific actions to resolve them.

Breakdown
Please note that the scale of the metrics' bar graphs shown in Figure 8.7 below are measured in percentages.



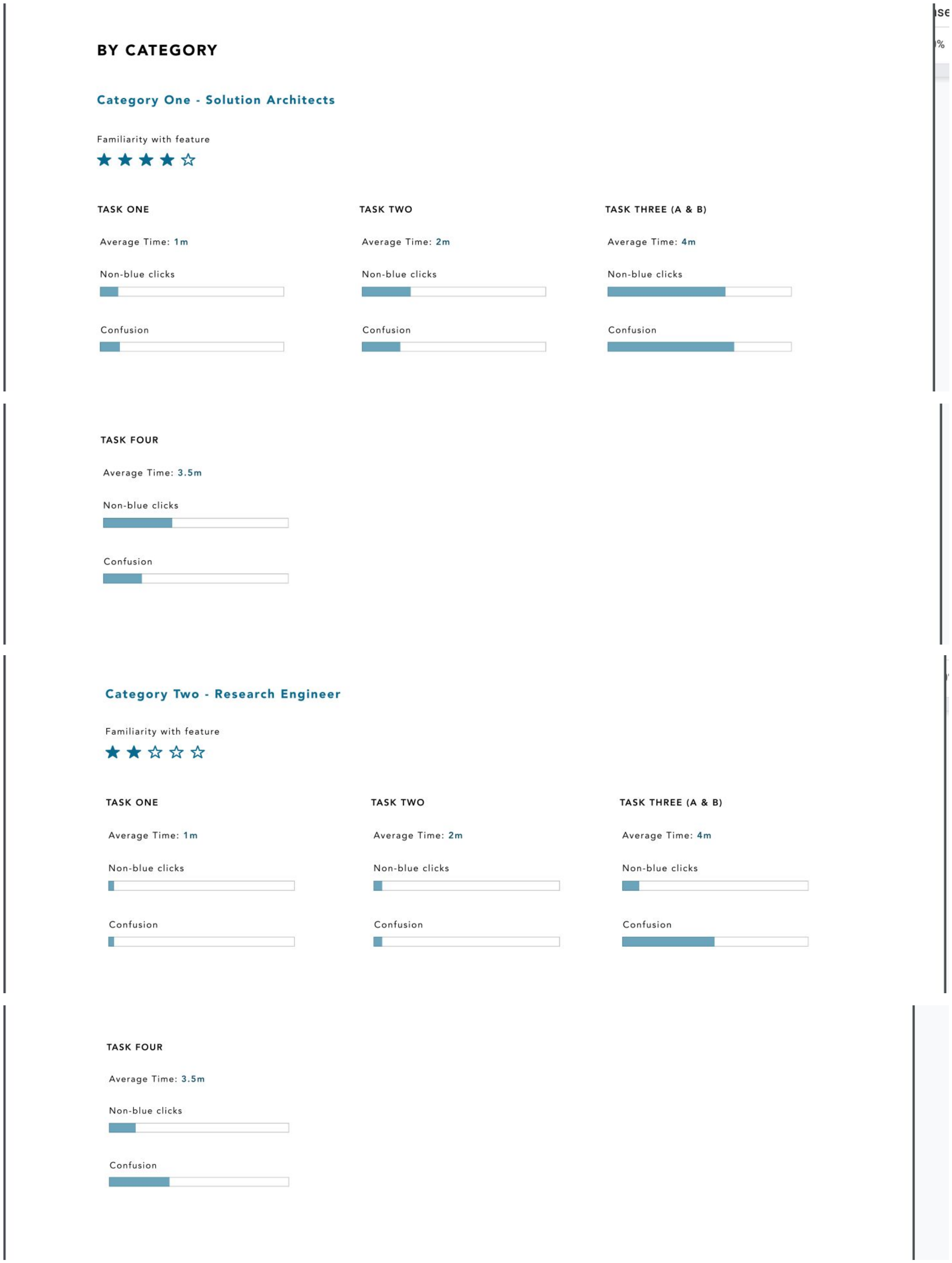


Figure 8.7 - Results breakdown

Common Feedback

Issues with the Edit Attributes Experience: Some of the participants found it to be quite repetitive. The process of individually having to activate an attribute for some tabs and not others, felt redundant for them as in most cases, they would prefer to have the same attributes automatically activated for all tabs, should they activate it for one tab. Some participants said that the phrasing "edit attributes" wasn't intuitive. Other phrasing such as - "See More Columns" or "Display Columns" were suggested.

Issues with Tabs and their relevance: One participant particularly struggled with understanding this necessity of the different tabs - Being Processed and Processed, as they are not applicable to the enterprise client she works with, and their use case.

Resolving an intervention: Although participants admired the design of this experience, it was a challenge to navigate through it and understand how to resolve the issue. This was also partly due to some of the participants not really understanding the use-case and context of the enterprise client in the prototype, that we explained above.

Filtering: Suggestions were also given on how to make the filtering process clearer to use, as well as the reassigning process. Similarly, there were a few difficulties with some participants navigating through finding how to find *live events* versus using a calendar to input the time period they wanted to filter the events based on. There was a need to simplify these processes.

Due to time limitations, we could not advance to a more high fidelity version of this prototype, with a good number of the feedback incorporated. However, it was a good start to establishing this feature and understanding the scenarios that are core to it. In the next chapter, we will discuss some of the limitations experienced in building both features - Application Evaluation introduced in Chapter 3, and Events Monitoring, introduced in this Chapter. We will also answer the research and sub research questions posed at the beginning of this thesis, as well as explore how both of these features and user experiences can be improved in the future.

4.4 CONCLUSION

In this chapter, we explored the second feature in this thesis and designed a system that enables operators, and operations managers, view and monitor events generated by their deployed apps. In the next chapter, we will discuss limitations encountered in designing both the events monitor feature, as well as the application evaluation feature in chapter 3. We will also answer the main and sub research questions posed in chapter one, and finally wrap up the following chapter by exploring future work and what lies ahead.

Design Principles and Decisions

In order to arrive at these designs above, once again some of the UX design principles in section 3.5 were applied, to ensure that users were able to carry out their main objectives and tasks with the experience, and at all moments they didn't find themselves stuck during the experience. The design once again was minimal, and aimed to provide relevant information needed to the user, as to avoid cases where they are overwhelmed with unnecessary data while trying to resolve an intervention/event, by investigating the event through viewing its event details. Some inspiration from these designs (particularly Figure 8.4 c and d) were also drawn from related work around displaying more information on an item while on a shopping site and an item is selected. Such as Figure 8.6 c shown below.

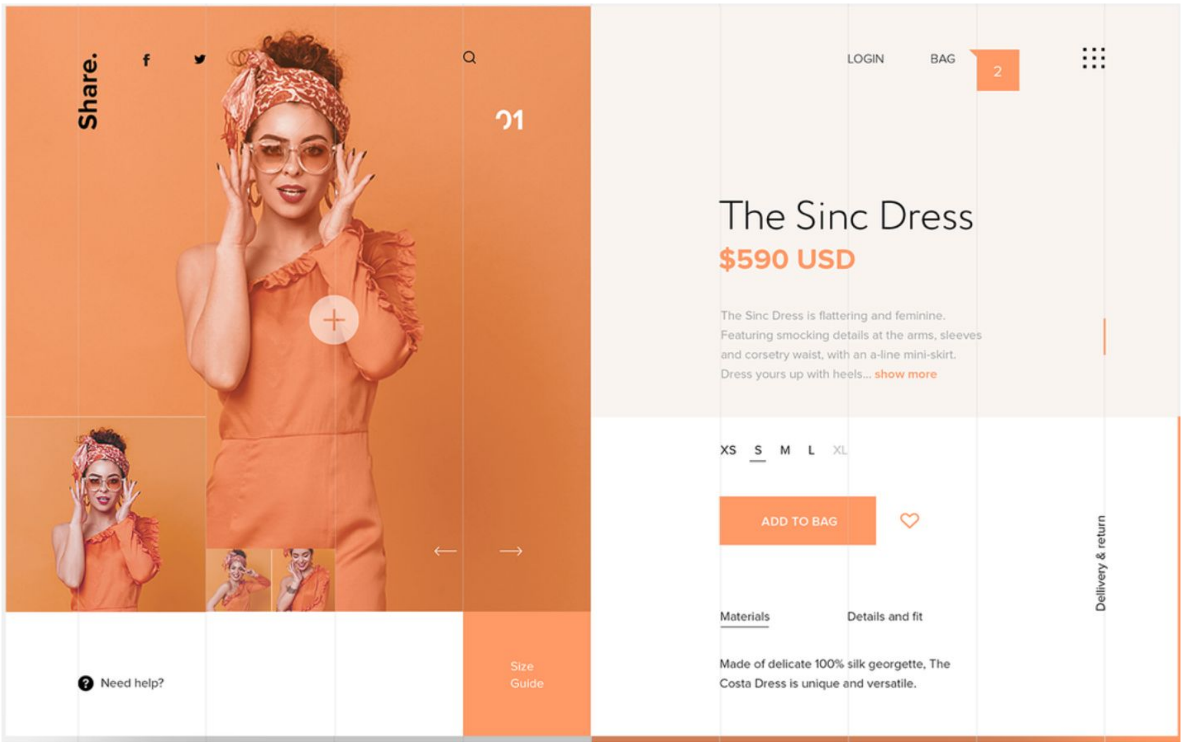


Figure 8.6 c - Shopping site

5. DISCUSSION

In this chapter, we explore certain limitations that were experienced during the design process, as well as the work that would have been if we had more time, and how these projects can be improved overall in the future.

5.1 LIMITATIONS

There are a number of factors that posed as limitations during the design processes for both the Application Evaluation and Events Monitoring feature. During the user research process, we were unable to speak to as many external clients and users due to the COVID-10 pandemic situation, but also due to the fact that most of the work is still being handled internally by Deepomatic, while external clients are getting up to speed with understanding how the process works, and what employees can take care of different parts of it. For this reason, Deepomatic is currently working on an academy that helps bring these enterprise clients up to speed. Thus a lot of the user research was done with internal employees, such as the product manager, the research engineers, the solution architects and the executive team, as shared above. A huge part of this research was also done based on readings and doing research on related work and challenges, as shared in Chapter 2. In the future, we would love to interview more employees that work at enterprise clients, to improve the personas created during the user research process, and create better experiences that are better scoped to their needs. Another reason why we focused more on internal employees, is because these features for now, will be primarily used by internal employees, although the longer term plan is to have external employees onboard with using them themselves, so they needed to be designed for both types of users ultimately. Another limitation was the time constraint. There were a lot more ideas that there wasn't enough time to implement, due to the 6 months time frame of the internship that will be listed in section 5.3 - Future Work. Another constraint was remotely working and having to initially adjust to this new way of collaborative work, that posed some difficulties in the beginning before getting into the swing of things.

5.2 FUTURE WORK

There are a few factors that can be incorporated into the design process for future work. One being that in the nearest future, Deepomatic would still need to rely on code for some part of the application evaluation system. What was presented in Chapter 3, was a code-free user experience, however, what may be more practical would be to update the design to make it a *little-code* experience, where certain steps leave room to input code in the UX, in order to facilitate the process, with the northstar vision still being a no-code experience. This is because there are parts of the process that will be technically challenging to implement an UX experience for, and would thus require more time and investment. For instance, the process of defining evaluation metrics, is one that would more realistically be done with code to start in the first version, before building a full UX experience that is also technically sound with detecting mathematical formulas and creating event sets and subsets. The updated high-level experience would also need to be tested with non-internal employees, to get tangible feedback from them on the usability of the feature.

In the Events Monitoring feature, some of the improvements that could be made in the near future include addressing some of the feedback received after the mid-level user test. This includes, making the experience of adding new columns/event attributes more intuitive and less redundant, and improving the experience of viewing more details about an intervention/event by providing more relevant information to the operator based on the use-case being addressed. For instance, prioritizing the display of the *Instructions Checked* tab (Figure 8.4 c), over the information shown on the *Overview* tab (Figure 8.6 b), since operators most of the time care more about viewing the photos associated with the intervention, or perhaps integrating both views into one, and organizing the information shown so these photos, and the technician who handled them are prioritized over other information on the page such as the who the event has been assigned to and its anomaly status.
Overall, there is a need to update the experience keeping the main types of users that use the Events Monitoring feature in mind - the Operations Managers, the Solution Architects and other secondary uses that may need to interact with this feature, and ensuring it addresses most of their more common scenarios. Due to the fact that these users need to carry out different objectives from the platform, there will need to be user experiences that somewhat vary based on what user was interacting with the platform.

Also, because these both experiences have been designed based on one of two use-cases that Deepomatic solves for, in order to have experiences that work for most to all use-cases, advanced customization may need to be incorporated into the designs. Meaning that these designs would need to be customizable by the enterprise clients, flexible and adaptable to their different use-cases. Users would need to be able to choose how they want to display or aggregate important data they need. Based on a conversation had with an Operations Manager at Bouygues Telecom who particularly uses the Events Monitoring feature, the platform should also be better tailored towards operators who often feel overwhelmed by viewing all of these events in a list, and may prefer to see them one at a time. This feedback is also in line with creating different types of views and experiences for the different users using the feature. The operators who are handling and supervising live installations may prefer to view one at a time, while operations managers or solution architects, may prefer a historical view, in the way the current design presents it.
Overall, all of these features, including others on the Deepomatic platform, would need to work together in a more cohesive way that tells the user a simpler story about how all these moving parts work together, and how they are connected to ultimately enable the enterprise client/user achieve their business goals. This means that there would need to be a clearer tie in between viewing the applications, their performances post deployment, and overall success business wise for the client.

6. CONCLUSION

As a recap, here is the main research question that was posed at the start of this thesis in Chapter One:

How do we design a more accessible UX experience for carrying out the evaluation of computer vision systems and the monitoring and management of the events they generate after they have been deployed in production, and are being used by enterprise clients?

Subsequently, we deduced these six sub-questions that we decided to explored in order to answer our main question:

- SRQ1: *What is a computer vision system?*
- SRQ2: *Who are the stakeholders involved in implementing a computer vision system?*
- SRQ3: *What are the challenges involved in implementing a computer vision system?*
- SRQ4: *What is the life-cycle of a computer vision system?*
- SRQ5: *What challenges are typically experienced at the* ***system performance evaluation*** *and the* ***event monitoring*** *stages of the life-cycle?*
- SRQ6: *What are existing ways to go about resolving some of these identified challenges?*

In chapter two, section 2.3, we answered our first sub-research question - *what is a computer vision system,* as a system that is created to enable computers see and interpret the visual world, in the way humans do. Throughout this thesis, we also had a chance to explore who the stakeholders are in this universe, thereby answering the second sub-research question - *who are the stakeholders involved in implementing a computer vision system*. We identified them as the enterprise clients, the AI solution providers and occasionally, a contracted company hired by the enterprise clients, to be the middleman between them and the AI solution, helping them with implementing these solutions in their existing systems. In chapter 3, we further investigated this by representing the principal stakeholders in the form of user personas. In chapter 2, we also explored answering the 3rd, 4th and 5th sub-research questions, by clearly illustrating the life cycle of a computer vision system, what challenges are involved with implementing one, particularly in the performance evaluation and event monitoring stages of the life-cycle. In chapter 2, section 2.7, we presented work done by other companies and literature around the state of the art, in order to explore ways that some of these challenges are being addressed in other spaces. This thesis, was overall carried out to answer our main research question posed in chapter one -

How do we design a more accessible UX experience for carrying out the evaluation of computer vision systems and the monitoring and management of the events they generate after they have been deployed in production, and are being used by enterprise clients?

We see from chapters 2, 3 and 4, that this question has been addressed by carrying out research to first have a better understanding of what stakeholders are involved in this process, what enterprise clients need, what solutions Deepomatic and other AI companies offer, and how they all work together to implement and develop these AI applications. More specifically, in chapters 3 and 4, we carried out user research to represent the user groups involved in the process of evaluating computer vision systems and monitoring the events they generate, with users. Next, we created user journey maps as part of the user research process for each of these personas, to inform the designs we created as a way of answering our main research question. The main driving force in both designs was creating experiences that didn't require code, to carry out main tasks and scenarios that these personas often need to do, in order to properly evaluate their computer vision systems, and monitor events post their deployments. Experiences that were easy to onboard onto, and could be used by a range of users with different programming proficiencies. Based on the studies and reviews carried out and discussed in both chapters 3 and 4, we were able to successfully do this. However, there remains work that can be done in the future, to improve these systems.

In this thesis, we presented a way to design systems that are more accessible and user friendly to people with little to no programming knowledge. We designed two systems that work hand in hand with each other, one that enables users evaluate the performances of their computer vision systems, and another that enables them monitor and manage the events these systems generate post deployment in production, in order to improve their business processes. In doing this, we answered our main research question posed in chapter 1, and discussed in chapter 5, we also shared our future vision in section 5.3, on how these systems could evolve. Following the work done in this thesis, these mockups/UX experiences will be developed by the software engineering team at Deepomatic and made available for use by internal users and enterprise clients.

Overall, when it comes to designing experiences that make AI more accessible, it is important to know how AI and Machine Learning work, and who is involved in the universe of developing and rendering these AI systems available to customers. Another point to emphasise with AI is that because of the increasingly wide number of user profiles that will be interacting with these systems, it is important to keep this mind while designing, that different user profiles may need and prefer different things. Not everyone interacting with these systems are technically proficient when it comes to programming, and working with AI systems more specifically and additionally, not everyone trusts AI systems in the same way. There needs to be an increased amount of transparency and clarity between the user and the AI system while it is running different operations. We see from the research and studies carried out that explainability is also important in that the user should be able understand why the computer vision system came to a conclusion, and the lack of this causes confusion. It is thus clear that implementing good UX would be a big contribution to how the use and adoption of AI in industries evolves in the future. The reason being that good UX practices and experiences, take into account a balance of an analytical way of thinking and an understanding of how AI works with a sense of empathy and understanding about how humans work, and how accessible, clear and trustworthy these experiences and systems need to be.

REFERENCES

1. Deepomatic's Website. https://deepomatic.com/en/ (accessed Sep. 10, 2020)
2. B. Hayes, J.A. Shah, *Improving robot controller transparency through autonomous policy explanation, in: Proceedings of the 12th ACM/IEEE International Conference on Human–Robot Interaction (HRI 2017).*
3. Tesla's Website *https://www.tesla.com/about*
4. Juegen Brock, Florian von Wangenheim. (2019). *Demystifying AI: What Digital Transformation Leaders Can Teach You about Realistic Artificial Intelligence.* California Management Review. 153650421986522. 10.1177/1536504219865226.
5. *Zhu, J., Liapis, A., Risi, S., Bidarra, R., & Youngblood, G. M. (2018, August). Explainable AI for designers: A human-centered perspective on mixed-initiative co-creation. In 2018 IEEE Conference on Computational Intelligence and Games (CIG) (pp. 1-8). IEEE.*
6. Stefano Gaspari, Principal, Four Principles, *Lean AI: 'Marrying' Artificial Intelligence and Lean Management in Manufacturing.* Available: here *(accessed: Sep. 10, 2020).*
7. Begam, M., Swamynathan, R., & Sikkizhar, J. (2013). *Current trends on lean management – A review. International Journal of Lean Thinking, 4(2), 1–7.*
8. Deepomatic Whitepaper, *Lean AI Methodologies.* Available: here *(accessed: Sep. 10, 2020).*
9. Nicolas Miaihe, Cyrus Hodes. *The Third Age of Artificial Intelligence (2017).* Available here *(accessed: Sep. 10, 2020).*
10. Maad M. Mijwil.(2015). *History of Artificial Intelligence. 3. 1-8.* Available: here.
11. Muhedin Hadzic, Salahudin Fetic, Emrus Azizovic. (2015). *Application of the Expert Systems in Artificial Intelligence.* University Journal of Information Technology and Economics. ISSN: 2335-0628.
12. Bishop, C. M. (2006), *Pattern Recognition and Machine Learning, Springer, ISBN 978-0-387-31073-2*
13. Hutchins, J. (2005). *"The history of machine translation in a nutshell".* Available: here. *(accessed: Sep. 10, 2020).*
14. Dana H. Ballard; Christopher M. Brown(1982). *Computer Vision. Prentice Hall. ISBN 978-0-13-165316-0.*
15. Yaniv Taigman, Ming Yanf, Marc'Aurelio Ranzato, Lior Wolf. *DeepFace: Closing the Gap to Human-Level Performance in Face Verification (2014).* Conference on Computer Vision and Pattern Recognition (CVPR).
16. P. Nguyen (2010). *"Automatic classification of speaker characteristics". International Conference on Communications and Electronics 2010. Pp. 147-152. ISBN 978-1-4244-7055-6*
17. Ghallab, Malik; Nau, Dana S.,; Traverso, Paolo (2004), *Automated Planning: Theory and Practice, Morgan Kaufmann, ISBN 1-55860-856-7*
18. Leurent, H., Boer, E. D., (White Paper, Jan, 2019) *"Fourth Industrial REvolution Beacons of Technology and Innovation in Manufacturing".* World Economic Forum. Available: here. *(accessed: Sep. 10, 2020).*
19. Jay Lee, Jaskaran Singh, Moslem Azamfar(2019). *Industrial Artificial Intelligence.* ResearchGate.

20. Begam, M., Swamynathan, R., & Sikkizhar, J. (2013). *Current trends on lean management – A review. International Journal of Lean Thinking, 4(2), 1–7.*

21. PwC's Global Artificial Intelligence Study: *Exploiting the AI Revolution.* *https://www.pwc.com/gx/en/issues/data-and-analytics/publications/artificial-intelligence-study.html*. *(accessed: Sep. 10, 2020).*

22. T. S. Huang. (1996). *Computer Vision: Evolution and Promise.* Available: here. *(accessed: Sep. 10, 2020).*

23. S. Sathiyamoorthy. (2014). *Industrial Application of Machine Vision. International Journal of Research in Engineering and Technology. Volume: 03. 10.15623/ijret.2014.0319120.*

24. Alex Polacco, Kayla Backes. (2018). *The Amazon Go Concept: Implications, Applications and Sustainability.* Journal of Business and Management.

25. Sreela Sasi, Ph. D. (2012). *Security Applications using Computer Vision.* ResearchGate. 60-77. 10.4018/978-1-4666-2672-0.ch004.

26. Stoplift's Website. https://www.stoplift.com/ (accessed Sep. 10, 2020).

27. World Health Organization, 2018. *Global Status Report on Road Safety.* Available: here. *(accessed: Sep. 10, 2020).*

28. Waymo's Website. https://waymo.com/tech/ (accessed Sep. 10, 2020).

29. Joel Janai, Fatma Guney, Aseem Behl, Andreas Geiger. (2017). *Computer Vision for Autonomous Vehicles: Problems, Datasets and State-of-the-Art.* Available: here (accessed Sep. 10, 2020).

30. Jiayin Qi, Feng Wu, Ling Li, Huaying Shu. (2007). *Artificial Intelligence Applications in the Telecommunications Industry.* Available: here. (accessed Sep. 10, 2020).

31. MDDI Online's Website. www.mddionline.com *(accessed Sep. 10, 2020).*

32. Amazon Web Services (AWS). (2020). *AWS DeepLens Developer Guide.* Available: here. *(accessed: Sep. 10, 2020).*

33. Hongkun Tian, Tianhai Wang, Yadong Liu, Xi Qia. (2019). *Computer vision technology in agricultural automation.*

34. Slant Range. (2020). *Aerial Phenotyping for Research & Breeding.* Available: here. *(accessed: Sep. 10, 2020).*

35. Ting Ting Liu, Ding Feng Wu, and Lii Yun Wang. (2020). *Development process of animal image recognition technology and its application in modern cow and pig industry.* IOP Conference Series: Earth and Environmental Science. 512. 012090. 10.1088/1755-1315/512/1/012090.

36. Essays, UK. (November 2018). Compass Group: An Analysis. Available: here. *(accessed Sep. 10, 2020).*

37. Belani, Vukovic, Car (2019). *Requirements Engineering Challenges in Building AI-Based Complex Systems.* ResearchGate. Available: here. *(accessed Sep. 10, 2020).*

38. Guang Jin, Long Bai, Hua Lin Hong, Xing Cai Guang, Zai Nong. (2020). *Developing an Artificial Intelligence (AI) Management System to Improve Product Quality and Production Efficiency in Furniture Manufacture.* Procedia Computer Science. Volume 166, 2020, Pages 486-490. Available: here. *(accessed Sep. 10, 2020).*

39. Carloalberto Treccani. (2018). *How Machines see the World: Understanding Image Annotation.* Available: here. *(accessed Sep. 10, 2020).*

40. Jose Hernandez-Orallo. (2014). *AI Evaluation: past, present and future.* Available: here. *(accessed Sep. 10, 2020).*
41. Jose Hernandez-Orallo. (2016). *Evaluation in artificial intelligence: from task-oriented to ability-oriented measurement.* Artificial Intelligence Review. Volume 48, Number 3, Pages 397 - 447.
42. Kevin C. Desouza, Gregory S. Dawson, Daniel Chenok. (2019). *Designing, developing and deploying artificial intelligence systems: Lessons from and for the public sector.*
43. James Shaw, Frank Rudzicz, Trevor Jamieson, Avi Goldfarb. (2019). *Artificial Intelligence and the Implementation Challenge.* J Med Internet Res. 2019 Jul 10;21(7):e13659. doi: 10.2196/13659. PMID: 31293245; PMCID: PMC6652121.
44. Stephan Schlögl, Claudia Postulka, Reinhard Bernsteiner & Christian Ploder. (2019. *Artificial Intelligence Tool Penetration in Business Adoption, Challenges and Fears.* ResearchGate. 259-270. 10.1007/978-3-030-21451-7_22.
45. Can Yavuz. (2019). *Machine Bias Aritificial Intelligence and Discrimination*
46. Adrienn Skrop, Tibor Holczinger, Krisztian Bakon, Balint Mihalics, Szilard Jasko. (2018). *Industry 4.0 - Challenges in Industrial Artificial Intelligence.* International Scientific Conference on Tourism and Security.
47. Raquel Sanchis, Oscar Garcia-Perales, Francisco Fraile, Raul Poler. (2019). *Low-Code as Enabler of Digital Transformation in Manufacturing Industry.* Applied Sciences. 10. 12. 10.3390/app10010012.
48. Shakkeel Ahmed, Ravi S. Mula, Soma Dhavala. (2020). *A Framework for Democratizing AI.* Available: here. *(accessed: Sep. 10, 2020).*
49. *Sam Charrington. (2017). Artificial Intelligence for Industrial Applications.* Available: here. *(accessed: Sep. 10, 2020).*
50. Neoteric.eu. *12 Challenges of AI Adoption.* Available: here. *(accessed: Sep. 10, 2020).*
51. Manish Balakrishnan. *No Code Products within AI and ML.* Nocodejournal.com. Available: here. *(accessed: Sep. 10, 2020).*
52. Google's AI Platform. https://cloud.google.com/ai-platform. *(accessed: Sep. 10, 2020).*
53. Mavenwave.com. *Why Google's Cloud ML is its Most Important ML Launch Yet.* Available: here *(accessed: Sep. 10, 2020).*
54. AndPlus.com. *Create ML - Machine Learning in Swift*. Available: here *(accessed: Sep. 10, 2020)*
55. James Wexler, Mahima Pushkarna, Tolga Bolukbasi, Martin Wattenberg, Fernanda Viegas, and Jimbo Wilson. (2019). *The What-If Tool: Interactive Probing of Machine Learning Models.* CoRR, abs/1907.04135.
56. Yen-ning Chang, Youn-kyung Lim, Erik Stolterman. (2008). *Personas: From Theory to Practices.* DOI: 10.1145/1463160.1463214. Conference: NordiCHI '08 Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges.
57. Florentina Scarneci-Domnisoru. (2013). *Narrative Technique of Interviewing.* Bulletin of the Transilvania University of Braşov Series VII: Social Sciences • Law • Vol. 6 (55) No. 1 - 2013