BSc Thesis Creative Technology

Empowering Gardeners: Developing an Adaptive Task Management System for Hobby Vegetable Gardeners

Daisy Baars

Supervisor: dr. ir. Dennis Reidsma
Critical Observer: dr. K. Zalewska
January 31, 2025

UNIVERSITY OF TWENTE.

Abstract

In the last decade, there has been a significant increase in hobby gardeners. Gardening offers benefits to physical, psychological and social health. However, gardeners can experience challenges, such as task management. This thesis explores the development of *TuinMoes*, an adaptive task management system designed to support hobby vegetable gardeners.

Through a combination of literature review, state-of-the-art analysis and user interviews, key motivators and challenges were identified. The *TuinMoes* prototype was developed through an iterative ideation process, integrating task management with a visual garden aiming to enhance task visibility at a glance. A user evaluation with 20 participants assessed the glancibility of the overview page.

Box plots indicated an improvement in task glancibility with the overview, however, statistical analysis revealed no significant difference between the overview page and a traditional task list. Future research could focus on exploring the long-term impact of the platform in a real-world gardening setting.

Acknowledgment

I would like to express my gratitude to everyone who supported me during this graduation project. First of all, I would like to thank my supervisor Dennis Reidsma and critical observer Kasia Zalewska-Kurek for all their support. Their belief in my vision and strength carried me through this journey. In addition, they continually challenged me and encouraged me to strive for the best possible results. Furthermore, I would like to thank the gardeners I interviewed for providing meaningful insights and showing me their beautiful gardens, as well as the participants of the user tests, who helped me create a clear understanding of the platform's limitations and possibilities. Lastly, I want to thank my family, partner and friends for their support during both the challenging and rewarding moments.

Contents

List of Figures

Chapter 1

Introduction

In the last decade, there has been a significant increase in hobby gardeners. During the COVID-19 pandemic gardening became more popular as it served as a way to release stress and increase a connection with nature (Egerer et al., 2022). Working in a garden and watching plants grow provides multiple physical and psychological benefits, such as reducing stress and anxiety, creating a sense of accomplishment and improving health through exercise in fresh air. Besides the benefits on an individual scale, gardening places such as allotments and community gardens also serve as social places where gardeners can meet each other and share knowledge, which improves the sense of community (Hoffman, 2018). Gardening is an accessible hobby for all people, including children, the elderly and disabled individuals (Soga et al., 2017). These benefits make gardening an interesting topic for research.

However, next to these benefits, gardeners may also face several challenges. These challenges range from dealing with unpredictable weather and physical strain to managing conflict in community gardens. One persistent and impactful challenge faced by gardeners is task management. Task management in gardening involves keeping track of activities such as harvesting, watering and planning what to sow and when. This challenge can occur across different types of gardens, ranging from balcony gardens of $4m^2$ to larger allotments of $100m^2$ and community gardens shared by multiple gardeners.

Commonly used task management methods, such as relying on memory or scribbling tasks on sticky notes could be unreliable and confusing. These methods can lead to forgetting important tasks and frustration. This challenge might decrease the enjoyment of gardening and may discourage people from working in their gardens.

The growing interest in gardening as a hobby and the challenge faced with task management, make it an interesting area to explore. Currently, it is unclear what features and functionalities could help gardeners to effectively manage their gardening tasks and what specific difficulties they face in their daily gardening practices. This report aims to address this gap by investigating how an adaptive task management system can be designed to support hobby vegetable gardeners.

1.1 Research questions

The main Research Question of this thesis is defined as:

How can an adaptive task management system be designed to support hobby vegetable gardeners?

To answer this main research question, there are nine sub-questions divided by their compartments

Literature review:
SQ1: How can task management be described?
SQ2: What are the advantages and disadvantages of task management methods?
SQ3: What is important to consider when implementing task management?

Related work:
SQ4: What technologies exist to support gardeners?
SQ5: What other adaptive task management systems exist?

User interviews:
SQ6: What are the daily practices of gardeners?
SQ7: What are the challenges and motivators experienced by gardeners?
SQ8: How do they manage their tasks?

Ideation:
SQ9: How can technology help gardeners to manage their daily gardening tasks and which elements are best addressed with technology?

Evaluation:
SQ10: To what extent does the system help gardeners with viewing their tasks at a glance?
SQ11: How do gardeners experience the interface?

1.2 Outline

This report begins with background research as described in Chapter 2, covering literature on Personal Task Management (PTM), an examination of existing applications and insights from user interviews, which address SQ1 to SQ9. Next, Chapter 3 provides the methods for each phase. Chapter 4 explores the brainstorming process consisting of diverging and converging phases to create ideas. Moreover, this chapter presents the selection process and the final chosen concept. The chosen concept is further specified in Chapter 5 through specific requirements and user scenarios. In addition, Chapter 6 explains how the prototype is created, while Chapter 7 focuses on the evaluation phase. Lastly, Chapter 8 discusses the findings and Chapter 9 describes the conclusion.

Chapter 2

Background Research

This chapter will focus on the background research of this project. First of all, reviewed literature on Personal Task Management will be discussed. Secondly, related work, which consists of general and gardening task management applications, will be examined in the State of the Art Section. Lastly, the last section will provide insights from interviews with seven gardeners.

2.1 Literature Review

The main objective of this section is to gain insight into how personal task management methods can be used to support gardeners. This literature review consists of three parts. First, the definition of personal task management will be explored and evaluated. Then, an overview of the advantages and disadvantages of personal task management methods will be provided. Finally, existing literature about implementing personal task management will be explored.

2.1.1 Literature Research Methods

For this literature research, various search terms were used. The key search term was Personal Task Management / PTM. In addition, other terms were used such as Task Systems, Managing Tasks, Planning, Collaborative Task Management, Personal Task Management Tools and Personal Information Management. The starting point was articles found using Google Scholar. Furthermore, forward and backwards references and citations have been scanned to broaden the scope. This led to sixteen relevant papers, of which fourteen were peer-reviewed and the other two consist of a PHD thesis and a Masters Thesis. Different literature matrices were made to answer each sub-question, where the key points of authors were categorized for each sub-question. In this way, the arguments of the authors could be seen side-by-side which made it possible to find a well-reasoned answer to each sub-question.

2.1.2 Findings

Definition of Personal Task Management

There is not one clear definition of Personal Task Management (PTM), different papers discuss the various aspects of PTM. Personal Task Management can be described with three main components: list-making, prioritizing and scheduling tasks. First of all, in their master's thesis El Khateeb (2019) views PTM as a process of organizing responsibilities and daily tasks, which relate to the components of prioritizing and scheduling. Furthermore, in their PhD thesis Kamsin (2014) identifies the prioritizing and list-making of tasks as the two main concepts of PTM. In addition to Kamsin's view, Haraty and McGrenere (2016) mention that PTM is creating, adding and editing tasks in a list. Mizrachi and Bates (2013) link task management to projects where the focus is on scheduling and calendaring tasks. Although the papers do not explicitly mention gardening tasks, PTM can be applied to gardening. Thus, for this literature review, Personal Task Management for gardeners is described as the process of prioritizing and scheduling gardening tasks and the act of list-making by individual users.

Advantages of PTM tools

Four advantages come forward after reading the literature on PTM tools. The first advantage is the documentation of tasks to remember them later on. Williams et al. (2023) note that PTM tools allow users to capture their thoughts the moment they arise. In addition, Lu and Corriveau (2011) notes that PTM tools help users to document their thoughts quickly, which helps them to remember their thoughts in a future moment. The memorization of tasks can be useful for gardening tasks as well, while this may help them to recall what they have to do in the garden.

Secondly, PTM tools can help users to recall the context of previous tasks. According to Williams et al. (2023) PTM tools help users to remember the relevance and context of their tasks after an interruption. Moreover, PTM tools allow users to alternate between tasks and projects according toAhmetoglu et al. (2021). Thirdly, Bernstein et al. (2008) note that PTM tools enable users to document tasks and information for other projects while engaging with their main project. This suggests that PTM tools could help gardeners to switch between different environments, such as switching between their gardening tasks and work or household tasks. The last advantage of PTM tools as described by Ahmetoglu et al. (2021), is that these tools can provide an overview and help to prioritize tasks. These advantages, such as alternating between tasks, prioritizing and memorizing tasks, appear to apply to gardening tasks as well. Therefore it is important to consider these advantages when designing PTM tools for gardeners.

Disadvantages of PTM tools

PTM tools do not only have advantages but also disadvantages, which can be grouped into two categories. The first category is linked to the mental workload required to use and manage PTM tools. Firstly, Bellotti et al. (2004) point out that users often find it time-consuming and a lot of effort to add and update tasks in a tool. In addition, Ahmetoglu et al. (2021) note that another challenge users face is that it can be hard to estimate the amount of time a specific task takes to complete. These disadvantages can increase the mental workload, which may make PTM tools less appealing to use.

The second category that can be derived is the overall user experience of PTM tools. According to Williams et al. (2023) users do not experience enough integration between PTM tools and other applications, which can limit the effectiveness of PTM tools. Additionally, Bellotti et al. (2004) mention that users find the output of the tools not as expected. Williams et al. (2023) further note that in collaborative PTM tools, users feel that their managers have too much control over how they manage their time. These disadvantages can decrease the overall user experience, which may make PTM tools less appealing to use. These two categories of disadvantages can be experienced by gardeners when using PTM Tools. Therefore designers should consider these issues when designing PTM tools.

Implementation of PTM tools

As previously stated designers should consider the advantages and disadvantages of PTM tools during the design process. In addition, four aspects come forward in the literature that designers should take into account as well. The first aspect relates to the differences in strategies. According to Boardman and Sasse, designers should remember that task management differs between individual users. Furthermore, designers should be aware of the different strategies that may be used depending on the user or task. In addition, Bellotti et al. (2004) note that the context and different types of tasks demand different strategies. Both Bellotti et al. and Boardman and Sasse argue that PTM tools should allow flexibility and offer diverse ways to view and manipulate to-dos. Similarly, research of Hu et al. (2024) shows that users find flexibility and personalization important features of PTM tools. Users should be able to filter tasks and turn on or off specific functions, for example when they want to focus on specific types of tasks. The need for flexibility could be applied to gardening tasks as well because gardeners appear to have a diverse set of tasks they should attend to, such as sowing, weeding and harvesting. In addition, Boardman and Sasse note that the organizing tendencies of users and thus PTM tool requirements change over time, which means designers should be innovative and redesign often. Likewise, Blandford and Green (2001) note that PTM tools should be flexible to adapt to the changing needs of users. Thus, when designing PTM tools for gardeners, PTM tools should allow them flexibility and the ability to adapt their strategies.

The next aspect that designers of PTM tools should consider is that users feel that these tools can be time-consuming. Williams et al. (2023) state that participants of their research reported that they wish to capture information quickly the moment task-related thoughts arise. Therefore, users have to be able to log information without knowing how to organize it or needing to make an elaborate plan. In addition, users dislike authentication systems, since they feel these are time-consuming as well. In addition, Bellotti et al. (2004) emphasize that PTM tools should allow users to make abstract plans, without knowing the exact details yet. Moreover, Bernstein et al. (2008) note that users should be able to easily document information without using a lot of effort or time. However, Ahmetoglu et al. (2021) state that detailed plans help users to stay more focused in comparison to vague and abstract plans. For this literature review, abstract plans and capturing information quickly are aspects that designers should consider.

The third aspect of implementing PTM tools is the user interface. Bellotti et al. (2004) argue that PTM tools should be easy to use and have clear visualizations. Visualizations of task histories and time constraints will help users to easily make decisions about workload and tasks. Furthermore, they argue that the tools should have a feature to handle old to-dos since these tasks probably will be forgotten after two weeks. They also note social relationships play an important role in task management and therefore there must be a function to have multiple participants.

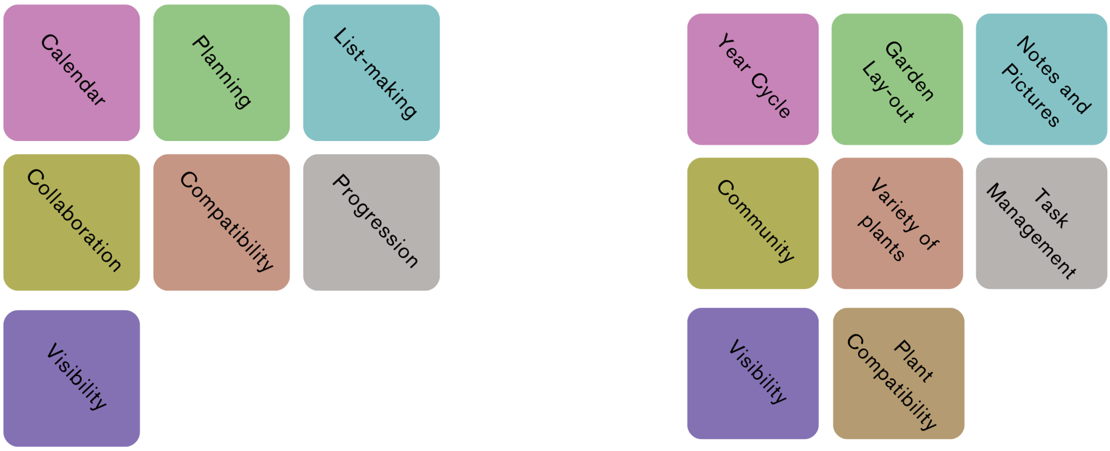
Lastly, PTM could be implemented into smart systems. According to Williams et al. (2023) PTM should be incorporated into intelligent systems. In addition, Ahmetoglu et al. (2024), mention that Artificial Intelligence (AI) could be suitable for PTM tools, while AI software could support users and make reasonable recommendations for them. Moreover, White (2022) notes that implementing AI in PTM tools can enhance the productivity of users. Therefore, AI can support gardeners in making task decisions and thus increase productivity. This appears to be beneficial to gardeners because it can increase the time they spend in the garden.

2.2 State of the Art

This section explores the current applications designed for task management. This research will look into general task management applications as well as specific applications for gardening.

2.2.1 State of the Art Methods

There are many applications that support users in managing their tasks. However, there is considerable overlap between these applications. For this study, a selection of twelve applications has been made, which have been divided into two groups: General and Gardening Specific task management Applications. The analysis involved the creation of two separate tables, consisting of multiple categories as shown in Figure 2.1. Detailed versions of the categories can be found in Appendix B. For each application, the characteristics were examined and categorized. Finally, the overlapping characteristics of these applications have been analyzed to determine which aspects are important for designing a gardening task management system.



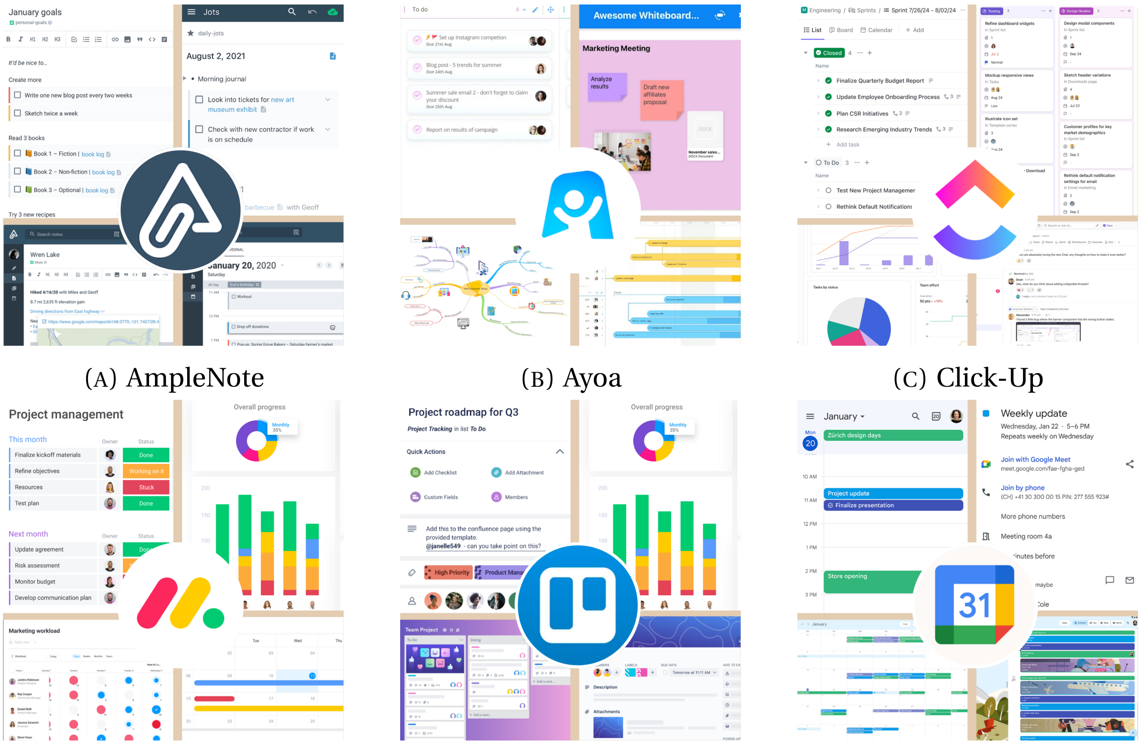
(A) General Task Management Tools

(B) Garden Task Management Tools

FIGURE 2.1: Categories for the analysis of Task Management Tools

2.2.2 General Task Management Applications

General task management applications can be described as tools to help users with task planning, list- making and scheduling. These applications are important to examine, to gain insight into the current applications that are used and to understand which aspects are important to take into account when designing a task management system for gardeners. For this section, six apps as shown in Figure 2.2 will be investigated. The key aspects of these applications can be grouped into seven categories.



(A) AmpleNote

(B) Ayoa

(C) Click-Up

(D) Monday

(E) Trello

(F) Google Calendar

FIGURE 2.2: General Task Management Applications

The first category investigated is the ability to create **lists of tasks**. List-making can be described as organizing a series of tasks in a specific order. Of the six applications examined, five allow users to create multiple lists for tasks. One key aspect shown in Monday, Click-up and Ayoa is that they provide users the option to sort tasks according to their priority. In addition, Monday enables users to sort their tasks according to their progress status. Another interesting characteristic is the ability to link tasks to other tasks, which users can do in the Click-up application.

The next category consists of the ability to make a **planning for tasks**. All the researched applications allow users to make a planning, however they implement different strategies that can be used to plan tasks. One example of a specific approach is Gantt Charts, which provide a general overview of a project with the due dates and the interdependence between tasks. A similar approach is the Timeline, which shows the road map of a project in chronological order. Another strategy implemented by Trello and Ayoa is the Kanban Board. This method helps users order tasks in three categories: To-do, Doing and Done, which can enhance the visuality of the progress. Lastly, users can plan their tasks using a calendar, where they will be able to see an overview of their tasks for the day, week or month. As previously stated in 2.1.2, strategies differ between users and the context of the tasks. Therefore, designers should take different strategies into account when designing applications for task management.

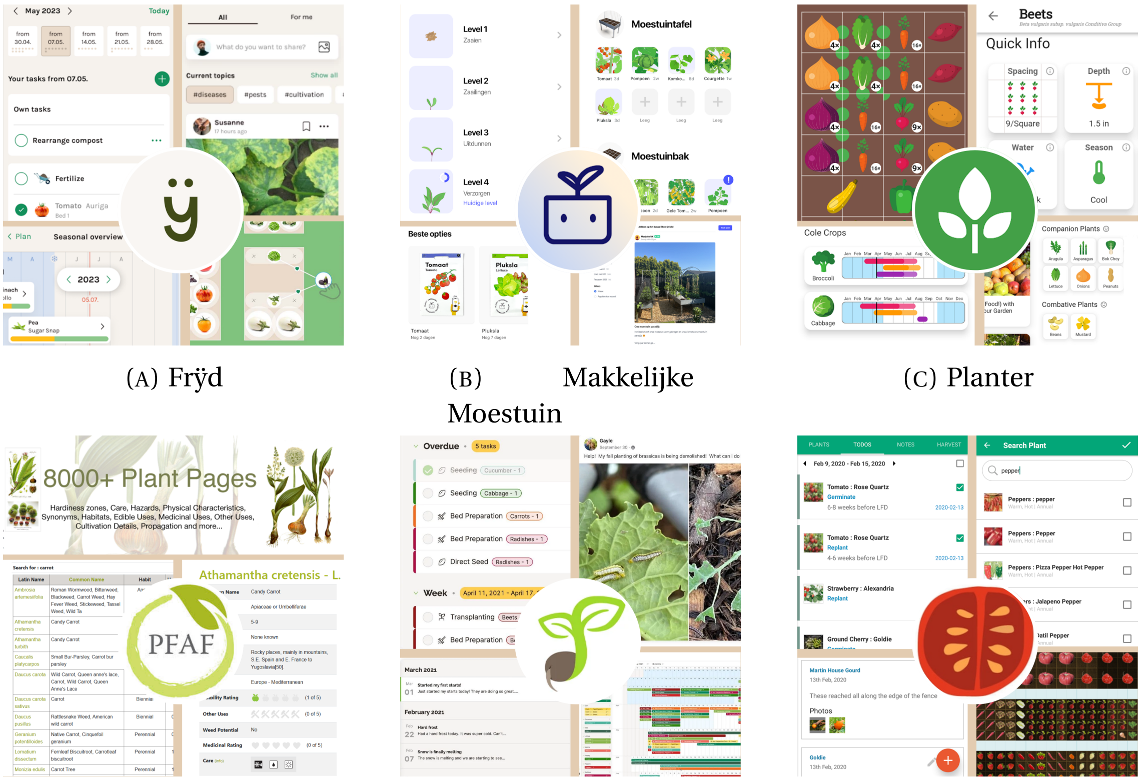
Furthermore, another feature that six of the applications in this research possess is the ability to **track progression**. The first feature all six applications have is the ability to label specific tasks as done when finished. As mentioned in the above paragraph a Kanban Board allows users to see what tasks are finished and what still needs to be done. Moreover, AmpleNote allows users to see the amount of tasks finished in a specific time frame. The last feature, implemented by Click-up, is that users can create a goal with sub-tasks. This allows users to see their progress of the whole goal when they have finished a task.

Some of the applications have been designed specifically for **working in groups**. Tools such as Ayoa, Click-up, Monday and Trello enable groups to collaborate both simultaneously and separately and enable groups to divide tasks when needed. Google Calendar allows users to share calendars, which could assist groups with planning. Another feature that enhances group work is if applications are compatible with other applications. All researched applications offer this support.

Lastly, the **complexity of user interfaces** will be examined. There are some notable differences between the interfaces of the applications. The first thing users notice is the layout of applications which can be considered as an important aspect. Most applications researched, AmpleNote, Monday, Trello and Google Calendar, were easy to use and had an intuitive interface. Other applications, such as Click-Up focus on collaboration and an all-in-one -solution, which makes it more difficult to navigate. The same applies to Ayoa which focuses on task management combined with mind-mapping. Even though these two applications have their advantages, this also means the layout is less intuitive than simple applications.

2.2.3 Gardening Task Management Applications

Gardening Task Management applications can be defined as tools to help users with planning, making lists and scheduling concerning their gardening tasks. In addition, gardeners can plan the layout of their garden and plan accordingly to the year circle. For this section, six apps as shown in Figure 2.3 will be investigated according to eight categories.



(A) Frÿd

(B) Makkelijke Moestuin

(C) Planter

(D) Plants For A Future

(E) SeedTime

(F) Smart Gardener

FIGURE 2.3: Garden Specific Task Management Applications

The first category to investigate is the ability to make a **layout of a garden bed**. Gardeners can create an online layout of their garden bed before they sow. This can help them to decide where each plant should go and consider the compatibility between different plants. Of the six researched applications, four allow users to make a virtual garden layout, namely Makkelijke Moestuin, Frÿd, Planter and Smart Gardener. They all use the square foot method where in each square a plant should be placed and some plants such as tomatoes take up multiple squares. The Frÿd application allows users to use layouts created by other gardeners. With the Smart Gardener application, gardeners can choose what kinds of plants they want in their garden and the application will make the layout for them. A notable limitation of these applications is that they are all square foot based and thus mostly applicable for small gardens and not for large gardens. The two other applications, SeedTime and Plants For A Future, do not allow users to make a garden layout.

The next category to discuss is whether applications show **plant compatibility**. Plant Compatibility focuses on whether plants can be sewn alongside each other or if it is better if they have different plants beside them. All investigated applications have this feature, except for SeedTime. The Makkelijke Moestuin Application shows if plants are not compatible with each other, however, users can not find more information on which plants are compatible. Three applications also show in the garden bed layout if the picked neighbouring plants are compatible, namely Frÿd, Planter and Smart Gardners. Furthermore, these applications allow users to click on the plants and see their specifics and the companion plants of this specific plant. Lastly, on the Plants For A Future website, gardeners can find the companion plants for each specific plant, however, the website can be hard to navigate and can be experienced as not user-friendly.

Another important aspect is the **variety of plants** users can find on these websites and applications. All applications allow users to find information about a set of plants. This is an important aspect, because the more plants are on the website or application, the more useful is it for the most people. The first application, Makkelijke Moestuin, allows the users to select 57 different plants, specific for small gardens. This is a limitation of this application, because it is not much. Planter for example has already a bit more, 80+ plants with more than 1000 different varieties. Frÿd, SeedTime and Smart Gardener all have around 3000 varieties, which gives the user a notable amount of freedom to choose the plants of their preference. Lastly, Plants For A Future is a big database with around 8000+ plant varieties.

Furthermore, the **sowing and harvest cycle** is another important aspect to consider for gardening planning applications. This cycle supports users to decide when to plant what crop and show around what time a crop is ready to harvest. Two applications, SeedTime and Frÿd are calender based where users can see when to sew and harvest each specific crop. In addition, SeedTime considers the hardiness location of the garden. These apps allow the users to see the cycle of each crop and adjust their schedule a little bit, but these schedules are rigid. Makkelijke Moestuin also uses this cycle to give the gardeners notifications when they have to tend to a crop, for example, sewing, transplanting or harvesting. Planter, Smart Gardener and Plants For A Future, do not give the users a fixed overview, but users can click on a specific plant and see their sowing and harvest cycle.

Moreover, the ability to **manage gardening tasks** in these applications will be examined. Four applications have the ability to see pre-programmed tasks based on the year cycle. SeedTime allows users to sort these tasks by date, crop or task type. Frÿd allows users to add their own tasks alongside these preprogrammed tasks. Furthermore, Smart Gardener can link the pre-programmed schedule to the personal calendar of gardeners. Plants For A Future does allow users to schedule tasks, however they give much information about weed potential and growing conditions, which gardeners can consider when scheduling their tasks.

Another category is the ability to **take notes and pictures**. Smart Gardener, SeedTime and Planter allow users to add notes and pictures to specific plants. Furthermore, with SeedTime users can link a specific tag to a note or picture and Planter allows users to add them to a specific garden bed as well. As mentioned in the user interviews in section 2.3, the social aspect of gardening is considered important as well, which can be stimulated by an online community. Half of the researched applications have this capability, namely SeedTime, Makkelijke Moestuin and Frÿd. Online communities allow users to create a connection with other gardeners. On these platforms, users can show their progress, ask questions and give tips to each other.

Lastly, the **complexity of these user interfaces** will be examined. Planter, Makkelijke Moestuin and Smart Gardeners have an interface with multiple different tabs where users can find information, such as the garden bed layouts, calendar, community and plant information. In addition, Makkelijke Moestuin gives step-by-step guides for tasks and is therefore beginner-friendly. Frÿd also has an intuitive user interface. However, SeedTime and Plants For A Future can both be experienced as overwhelming. SeedTime has the cycle of multiple different crops simultaneously, which can overlap and therefore could confuse users. Plants For A Future is a large database with a lot of useful information, but the information is harder to find and the website is not easy to navigate.

2.3 User Interviews

Next to the literature review and the state of the art, user interviews have been conducted. The goal of the interviews was to gain a better understanding of the practices of the gardeners. In addition, understanding the motivators and challenges of gardeners will help to understand what is important to consider when designing a task management system. This section will show the characteristics of the gardeners, the motivators and the challenges they face. Furthermore, the task management strategies will be examined. Lastly, participant input about designing a task management tool will be given.

2.3.1 Methods

A semi-structured approach has been chosen because this allows in-depth insights into the motivators and challenges gardeners face. The interview questions have been divided into three categories: The characteristics of the participants' gardening, their experience of gardening and the task management aspect. The interviews were held in the gardens of the participants because this was considered a place where they felt comfortable. First participants had to read the information letter (Appendix D) and sign the consent form (Appendix E). The interviews started with walking around the garden and asking the participants to show their garden. This made the conversation more natural, which helped the participants to give information freely without the constraints of the interview questions. When there were unanswered questions, I asked the participants the questions at the end and also asked them if they had any additional insights into what they would like in a task management tool.

In total, seven interviews have been conducted. To get an overview of different types of gardeners and gardens, participants with different projects were asked to get a broad range of participants. Most participants live around Enschede, but one participant lives in Hoofddorp.

The interviews have been transcribed, anonymized and color-coded according to multiple categories. Furthermore, the information has been organized for each person in a table to get an overview of all information by the corresponding categories, as shown in Figure2.4. A detailed version of the categories can be found in Appendix C.

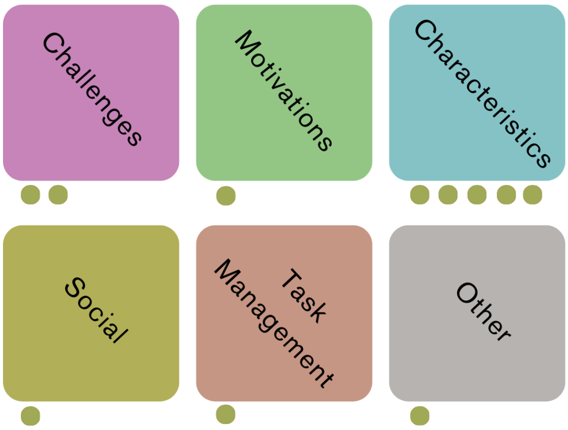


FIGURE 2.4: Categories of color-coding user interviews

2.3.2 Findings

Characteristics

There is a broad range of different types of gardens, such as balcony gardens, allotment gardens and a food forest. In addition, some gardeners focus more on flowers and trees, while others focus on vegetables. The variety of plans has to do with the total area of the gardens. The gardens range from $4m^2$ up to $7500m^2$.

Furthermore, the years of experience differ as well between the participants, ranging from 2 years up to 30 years with an average of 12 years of experience.

The amount of time gardeners spend in their gardens differs between gardeners and between seasons. Some gardeners spend around 1 to 2 hours in high season and let the garden rest in the winter. Other gardeners spent around 10 to 15 hours a week in high season and a couple of hours in low season.

Gardening can benefit their social health and the gardeners have a wide range of the amount of social contact in and around their gardeners. One participant commented that they work together with volunteers every Saturday. Others mentioned they work together at the same time on different neighbouring gardens or work on collaborative projects in the gardens.

When designing a task management tool it is important to consider these characteristics, because these factors impact the strategies and needs of the gardeners.

Motivators

Six different motivators were found during the user interviews, as shown in Table 2.1. The first motivator was social connection, which two participants experienced as a main motivator. The second motivator was the quality of food, which both participant 6 and participant 1 experienced as important. Participant 6 mentioned: 'Because I find the products that come from there, like strawberries and cucumbers, tastier than the ones from the store.' [Omdat ik de producten die er af komen, zoals aardbeien en komkommer, toch lekkerder vind als uit de winkel vandaan.] Furthermore, a motivator that six participants noted was the connection with nature. In addition, five of the seven participants stated that being outside and enjoying physical activity was another major reason for gardening. Participant 2 said: 'I have a job where I sit behind a computer all day as a secretary. And now I am forced to be outside. Even if I don't have a harvest, at least I'm outside.' [Ik heb een baan dat ik de hele dag achter de computer zit als secretaresse. En nu ben ik gedwongen buiten. Ook al heb ik geen oogst, ik ben wel buiten]

The last two motivators were experienced by Participant 1, who noted that they like the ability to experiment in the garden and that the food is cheaper than buying it in the store.

TABLE 2.1: Motivators of seven participants

	1	2	3	4	5	6	7
Social connection	x					x	
Quality of Food	x					x	
Experimenting	x						
Physical Activity	x	x		x	x		x
Nature connection	x	x	x	x	x		x
Cheaper food	x						

Challenges

The participants mentioned eight different challenges faced, as shown in Table 2.2. The first two challenges are influenced by the weather conditions. First, three of the seven gardeners experience pests that can harm their crops. In addition, three gardeners mentioned the challenge of the growth of weeds due to the weather conditions. For example, Participant 1 commented: 'The balcony is mainly, considering the number of plants there, very intense. All the weeds come up at the same time. So, at certain times of the year, you're just busy with it for a good Saturday. ... If we've had damp weather followed by one and a half to two weeks of a lot of sun, the weeds really shoot up like crazy." [Het balkon is voornamelijk, gezien de hoeveelheid planten die daar staan, is de intensiteit heel heftig. Al het onkruid komt tegelijkertijd. Dus daar ben je gewoon op bepaalde momenten door het jaar ben je er vaak een goede zaterdag mee druk mee. ...... Als we vochtig weer hebben gehad en vervolgens anderhalve tot twee weken veel zon, dan schiet onkruid echt als een malle omhoog]. Furthermore, weed growth due to weeds of neighbors is a challenge faced by two gardeners as well. Participant 2 mentioned: 'So I'll get a lot of weeds from them next year, even though I really did my best this year.' [Dus ik krijg wel heel veel zaden volgend jaar van hen. Terwijl ik dit jaar heel erg mijn best heb gedaan.]

Moreover, challenges related to social connections have been found. First of all, Participant 1 commented that they experience friction when collaborating. Two other gardeners noticed that all the information was in their heads. This could be challenging when they can not think clearly but still want to communicate tasks with others. Participant 4 mentioned: 'How clear is my mind at that moment to even be able to come up with something?' [Hoe helder is op dat moment mijn hoofd dat ik überhaupt wat kan verzinnen?]

The last category of challenges relates to personal time and task management styles. Two younger gardeners noted that they lack time due to social activities and their studies and often do not remember to go to their vegetable garden. Participant 7 noted: 'But yeah, as a student who is busy, I also often have other things to do.' [Maar ja, als student die druk heeft, heb ik ook wel vaak andere dingen te doen.] Furthermore, Participant 5 mentioned: 'At the previous vegetable garden, I wasn't there, and then I just forget it; if I don't see it, I just forget it.' [Bij de vorige moestuin dan was ik er niet en dan vergeet ik het gewoon, als ik het niet zie dan vergeet ik het gewoon.] Lastly, participant 6 mentioned that when experimenting with new plants they do not have enough knowledge about the plants and therefore do not know what to expect.

TABLE 2.2: Challenges of seven participants

	1	2	3	4	5	6	7
Pests	x	x	x				
Weeds due to weather	x				x		x
Weeds due to neighbours		x					x
Information only in their head			x	x			
Lack of time						x	x
Remembering to go						x	x
Friction between collaborating strategies	x						
Unkown plants					x		

Task Management Strategies

As previously mentioned in Section 2.1.2 task management strategies differ from person to person and depend on the type of task and the context. All seven gardeners told me that for most tasks they just go to the garden and look at what needs to be done or what they want to do. Participant 5 mentioned: 'I just think sometimes... I'll go there again sometime. I'll see what I'll do then.' [Ik denk gewoon af en toe van... Ik ga er weer een keer heen. Zie ik wel wat ik ga doen.] Furthermore, two participants looked at the seed sachet to understand how to sow and harvest the vegetables. In addition, two participants noted that they look at what others are doing to know what and when they have to do their tasks. Lastly, Participant 1 and Participant 4 commented that if they have multiple tasks they will add 'Garden' or specific garden tasks to their general task list.

TABLE 2.3: Task Management Strategies

	1	2	3	4	5	6	7
Looking in the garden	x	x	x	x	x	x	x
Studying the seed sachets	x				x		
Watching others				x	x		
Adding 'garden' to general task list	x					x	

Participant Insights

Three of the participants provided additional recommendations and insights for the design of a task management tool. Participants 1 and 5 noted that a garden and task visualization to remind them of their garden and their garden tasks could help them. In addition, Participant 1 commented that they think a system for accountability to themselves and others could enhance social interaction proactively. Lastly, Participant 4 had ideas relating to the history and reflection of gardening. They would like a system that uses the past to predict future events and a system where they can log their activity to later reflect on.

2.4 Conclusion

The reviewed literature on personal task management gave several insights into possible implementations of these tools. Firstly, different strategies between users and context were considered during designing a task management tool for gardeners. Moreover, PTM tools can be perceived as complex and time-consuming and therefore such a tool should allow users to input tasks easily, quickly and abstractly.

In addition, the State of the Art provides insight into the current applications that can be used to manage tasks and garden activities. There are certain features that I consider valuable for this graduation project. Both the literature review and user interviews point out that social relations are a motivator for executing tasks and thus features that enhance collaboration can be seen as valuable. Furthermore, the interface should not be too complex, as the literature shows that this can discourage users. Moreover, there are two garden-specific features that I consider as important for this graduation project. The first feature is the ability to create a realistic garden layout, where users can add their vegetables. Secondly, a larger variety of vegetables makes it possible to alter and use an app to represent the garden of the user.

Lastly, the user interviews highlight that there are multiple types of gardeners who all have different vegetables, flowers, and plants. Even though a tool could likely not support all gardeners, it is important to realize these differences and therefore I have considered them when designing a gardening task management tool. Furthermore, a tool should not hinder important motivators, such as the connection with nature or physical activity. The motivator of social connection is considered during the ideation phase. The participants mentioned eight different challenges, where the focus during the ideation phase is on the challenge of remembering their tasks and going to their garden. In addition, I focused on the collaborative challenge of the information that is only in one person's head. Furthermore, participants mentioned that they learn from each other and look at what others are doing to understand what they have to do and when they have to do specific tasks. These insights have been taken into consideration during the ideation and specification phase to design a garden task management tool that supports gardeners to their best potential.

Chapter 3

Methods

This chapter will report on the methods used during the design process of this project. The phases of the design process are based on the Creative Technology design method as described by Mader and Eggink (2014). A description of how the phases were used is given in the sections. Furthermore, additional methods, such as a Harris Profile, the Moscow Method and statistical methods will be described.

3.1 Design Process

During this report, the principles of the Creative Technology Design process were used as described by Mader and Eggink (2014). This method consists of four phases: Ideation, Specification, Realisation and Evaluation. Every phase has specific characteristics that are part of that phase. However, during this report, specific parts were ordered differently and will be discussed for each phase separately.

3.2 Ideation Phase

The ideation phase consists of multiple different brainstorming tactics and sketching concepts. In contrast to Mader and Eggink (2014), the background research has been done before the ideation phase. The insights from the background research are used as a source of inspiration.

The first brainstorming tactic that has been used is the Brainstorming Matrix. This matrix has a column on the left with ten different constraints or design questions, such as: "What would a superhero do?" or "Make it a game". For each cell of this column, a row of 10 cells was created in which I added ideas that meet the constraint. If each cell is filled, this creates 100 ideas. The goal of this tactic is to create as many ideas as possible, which is the divergent part of the brainstorming phases.

Furthermore, categories were made based on combined ideas from the Brainstorming Matrix and the background research. The categories were further diverged by creating mind maps. Besides, a brainstorming session with a previous Creative Technology Student was held to dive deeper into the important aspects of the mind maps. From these mind maps and the brainstorming session concepts were created.

These concepts are part of the convergent stage where I developed a clear view of how these different ideas could look.

Moreover, preliminary requirements were created. These requirements are not specific yet, but they give more insight into the important parts to conclude in a final design. After establishing these requirements a Harris Profile for the concepts was created. With a Harris Profile, I could compare each concept by giving points for each requirement. The requirements were scored ranging from -2 up to 2. If a concept strongly failed to meet the requirement it scored -2, if it strongly met the requirement it scored 2. The concept that had the most positive outcome and the least negative was thus the concept that met the preliminary requirements the most.

Finally, a concept to further develop was chosen. This decision was based on the outcomes of the Harris Profile along side with my personal opinion and the opinions of my supervisors. This final concept was the starting point of the specification phase.

3.3 Specification Phase

Mader and Eggink (2014) argue that the specification phase consists of multiple prototypes, an evaluation and a feedback loop. In this report, the multiple stages of the specification phase are considered. However they were not done explicitly, but by considering the user context and through written scenarios functional requirements were established.

Stakeholders have been established during the user interviews in Section 2.3. In the specification phase, the gardening context and target audience were specified. This created a specified context for the design, which helped to decide what to include in the design and what not.

Secondly, written scenarios were created to gain a better understanding of the probable user experience with the design. This helped to understand what aspects of the design are important and how the platform could be used in multiple ways.

Lastly, functional and non-functional requirements were created and prioritized with the MoSCoW method. Functional requirements consist of the measurable features of the system and specific functions the system should have. Non-functional requirements focus on how users interact with the platform and the experience of the users. The MoSCoW method helped to prioritize the requirements into four different categories: Must have, Should have, Could have and Won't have. The goal of this method is to decide what parts of the design were crucial and should therefore be the main attention during the realisation phase.

3.4 Realisation Phase

The structure of the realisation phase as described by Mader and Eggink (2014) is characterized as follows: "decomposition of the start specification, realisation of the components, integration of the components and evaluation". The first three steps are the same in this report as described by Mader and Eggink (2014). However, both functional and non-functional requirements will be evaluated during the evaluation phase.

First, the specifications were decomposed and divided by the different tabs that were created. This helped to decide what to do first and how the different parts interact with each other. Furthermore, three web pages were created with HTML, CSS and JavaScript. In the realisation phase, ChatGPT has been used during the development stage. This allowed me to make a working prototype in the given time frame. The creation of the pages was done in a specific order. First, the Task tab and the Layout tab were created. Next, the two separate pages were combined on the Overview page to get the garden layout combined with the tasks. Besides the creation of the tabs, plants were drawn in Photoshop with a drawing tablet. This was done to create a cohesive and visually pleasing design.

3.5 Evaluation Phase

The last phase as described by Mader and Eggink (2014) is the evaluation phase. During this phase, I checked if functional and non-functional requirements were met. The functional requirements I evaluated before evaluating the non-functional requirements to ensure that the functional requirements were met before user testing. The non-functional requirements were evaluated by user testing. The user-testing group consisted of gardeners and people with an affinity for gardening, whom I asked to use the platform for a prebuilt garden. A test was created in which Group A could see all pages, while Group B could only use the layout and the task page. Participants were asked to enter tasks into the task page and after that, they had to find the corresponding tasks for specific dates. After completing these tasks, participants were asked to fill out a questionnaire which assessed their confidence in finding the right gardening tasks and asked for general feedback. Afterwards, the results were examined using box plots and statistical tests, such as the Shapiro-Wilk test, the Mann-Whitney U test and the one-tailed independent t-test. Finally, general feedback was categorized and examined to gain meaningful insights.

Chapter 4

Ideation

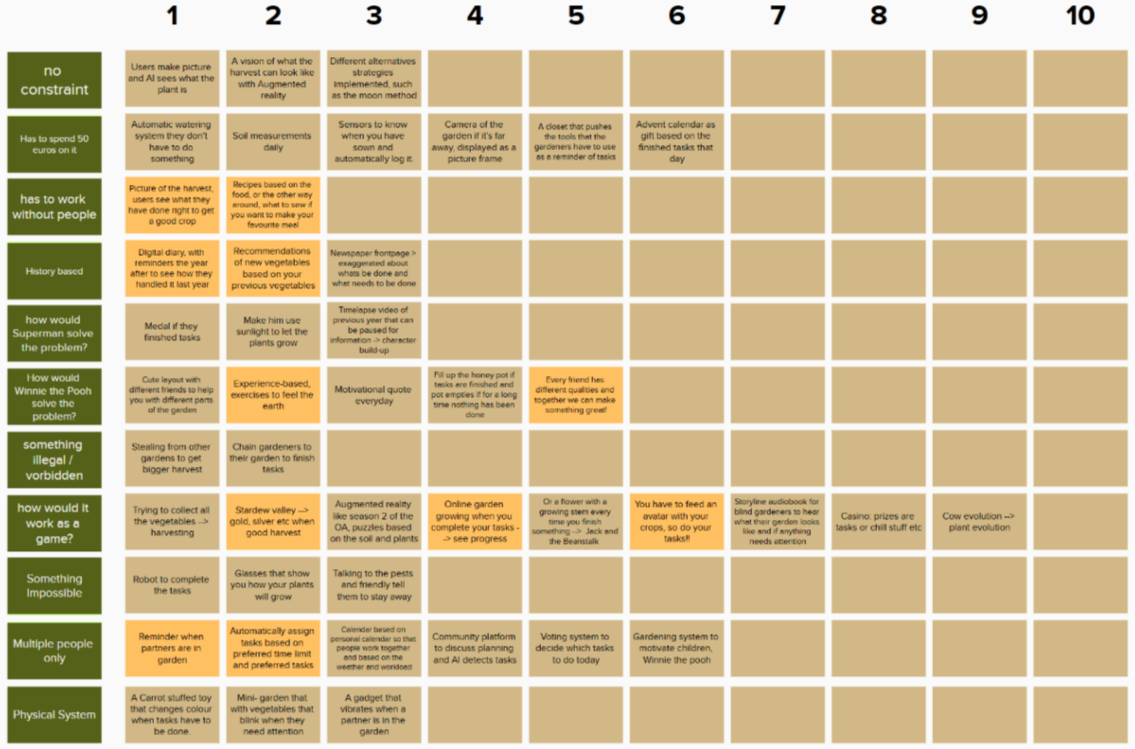
This chapter reports on the ideation stage of this project. First, the process will start with a diverging phase by creating a brainstorming matrix and several mind maps. Secondly, these ideas will be converged into three concepts, illustrated by sketches. Lastly, a final concept will be selected based on preliminary requirements.

4.1 Concept Generation

After establishing the background research the concept generation could start. Various stages and types of brainstorming were done to gain broad insights. This contains broadening techniques like a brainstorming Matrix and deepening methods like mind mapping.

4.1.1 Brainstorming Matrix

A brainstorming matrix consisting of 11 rows and 11 columns has been made. This strategy was chosen because it allows multiple types of ideas which helped with thinking in various directions. The first column contains the constraint an idea has to conform to, such as gamifying or "What would Superman do?". When the constraints were established, it was tried to develop 10 ideas for each constraint. After various rounds of brainstorming, these sessions produced 45 ideas, as shown in Figure 4.1.



	1	2	3	4	5	6	7	8	9	10
no constraint	Users make picture and AI sees what the plant is	A vision of what the harvest can look like with Augmented reality	Different alternatives strategies implemented, such as the moon method							
Has to spend 50 euros on it	Automatic watering system they don't have to do something	Soil measurements daily	Sensors to know when you have sown and automatically log it.	Camera of the garden if it's far away, displayed as a picture frame	A closet that pushes the tools that the gardeners have to use as a reminder of tasks	Advent calendar as gift based on the finished tasks that day				
has to work without people	Picture of the harvest, users see what they have done right to get a good crop	Recipes based on the food, or the other way around, what to sew if you want to make your favourite meal								
History based	Digital diary, with reminders the year after to see how they handled it last year	Recommendations of new vegetables based on your previous vegetables	Newspaper frontpage > exaggerated about whats be done and what needs to be done							
how would Superman solve the problem?	Medal if they finished tasks	Make him use sunlight to let the plants grow	Timelapse video of previous year that can be paused for information -> character build-up							
How would Winnie the Pooh solve the problem?	Cute layout with different friends to help you with different parts of the garden	Experience-based, exercises to feel the earth	Motivational quote everyday	Fill up the honey pot if tasks are finished and pot empties if for a long time nothing has been done	Every friend has different qualities and together we can make something great!					
something illegal / vorbidden	Stealing from other gardens to get bigger harvest	Chain gardeners to their garden to finish tasks								
how would it work as a game?	Trying to collect all the vegetables --> harvesting	Stardew valley --> gold, silver etc when good harvest	Augmented reality like season 2 of the OA, puzzles based on the soil and plants	Online garden growing when you complete your tasks --> see progress	Or a flower with a growing stem every time you finish something --> Jack and the Beanstalk	You have to feed an avatar with your crops, so do your tasks!!	Storyline audiobook for blind gardeners to hear what their garden looks like and if anything needs attention	Casino: prizes are tasks or chill stuff etc	Cow evolution --> plant evolution	
Something Impossible	Robot to complete the tasks	Glasses that show you how your plants will grow	Talking to the pests and friendly tell them to stay away							
Multiple people only	Reminder when partners are in garden	Automatically assign tasks based on preferred time limit and preferred tasks	Calendar based on personal calendar so that people work together and based on the weather and workload	Community platform to discuss planning and AI detects tasks	Voting system to decide which tasks to do today	Gardening system to motivate children, Winnie the pooh				
Physical System	A Carrot stuffed toy that changes colour when tasks have to be done.	Mini- garden that with vegetables that blink when they need attention	A gadget that vibrates when a partner is in the garden							

FIGURE 4.1: Image of the Brainstorming Matrix

Furthermore, the ideas that aligned the best with the outcomes of the user interviews and background research have been highlighted. This resulted in four interesting categories:

1. History-based progress
2. Nature connection
3. Social connection
4. Simulation Online Garden

4.1.2 Mind mapping

After these categories were established mind maps correlating with these ideas were made.

History-based progress

First, one of the participants of the user interviews noted that seeing their progress and what has changed in a year would be interesting to them. According to the literature review, AI could be implemented in Task Management tools. This led to the idea of having recommendations of tasks and vegetables based on previous tasks and vegetables. Note- and picture taking has been deemed desirable as well, which can be implemented in a history-based idea where users can see their pictures of last year and see what they had done around this time last year. However, no implementation of history-based progress landed on one concept. Some aspects, such as the recommendations and note-taking have been implemented into three lo-fi concepts.

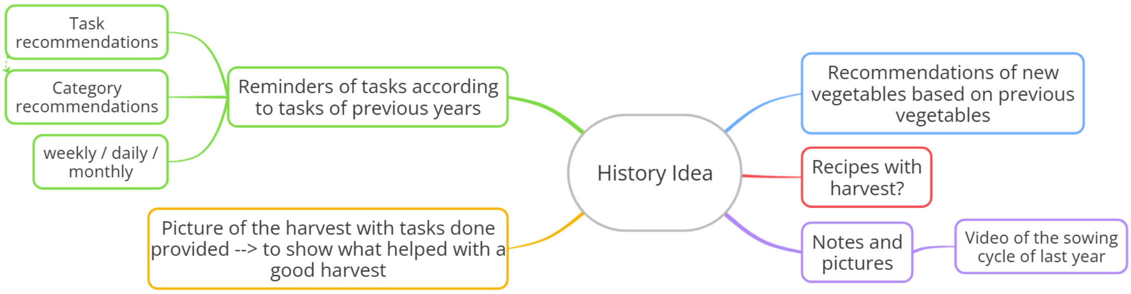

Task recommendations
Category recommendations
weekly / daily / monthly
Reminders of tasks according to tasks of previous years
Picture of the harvest with tasks done provided --> to show what helped with a good harvest
History Idea
Recommendations of new vegetables based on previous vegetables
Recipes with harvest?
Notes and pictures
Video of the sowing cycle of last year

FIGURE 4.2: Mind map of history-based category

Nature Connection

As mentioned in Section 2.3.2, a connection with nature is a great motivator for multiple gardeners. They mentioned that they like to be outside and want to contribute to the birds and insects. Therefore multiple ideas relating to the connection to nature have been combined to come up with Figure 4.3. Even though this mind map helped to get a clear overview of ways to implement this connection, it lacked other motivators and challenges established in Section 2.3 and thus did not lead to a whole concept as well. However, the connection with nature is an important motivator for gardeners and therefore should be considered when designing concepts.

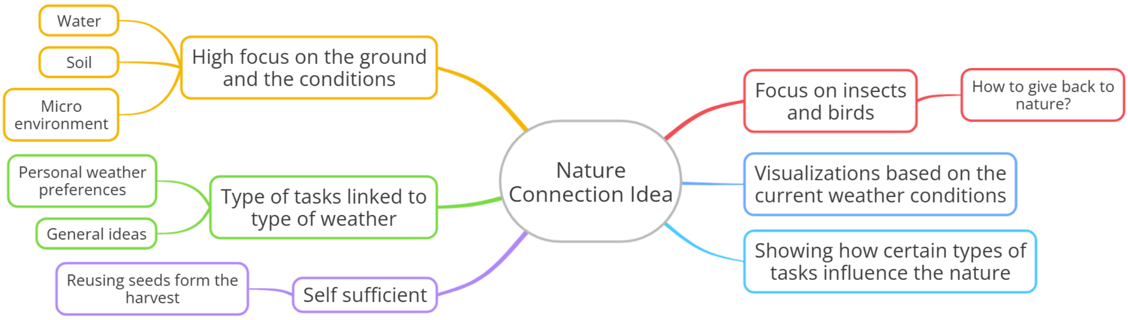

Water
Soil
Micro environment
High focus on the ground and the conditions
Personal weather preferences
General ideas
Type of tasks linked to type of weather
Reusing seeds form the harvest
Self sufficient
Nature Connection Idea
Focus on insects and birds
How to give back to nature?
Visualizations based on the current weather conditions
Showing how certain types of tasks influence the nature

FIGURE 4.3: Mind map of Nature Connection

Social Connection

This category focuses on the social connection between gardeners. As mentioned in Section 2.1 and Section 2.3.2, gardening can enhance social connection. Therefore, this idea focuses on various aspects that enhance collaboration and a social community. This can be done for example by notifications if partners are in the garden or an online platform where gardeners can support each other. This mind map contributed to the formation of the Online Community concept, which is discussed in Section 4.2.

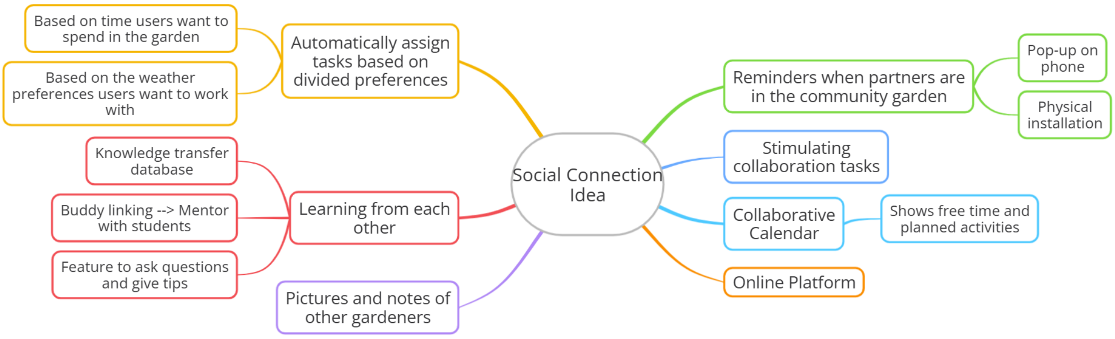

Based on time users want to spend in the garden
Based on the weather preferences users want to work with
Automatically assign tasks based on divided preferences
Knowledge transfer database
Buddy linking --> Mentor with students
Feature to ask questions and give tips
Learning from each other
Pictures and notes of other gardeners
Social Connection Idea
Reminders when partners are in the community garden
Pop-up on phone
Physical installation
Stimulating collaboration tasks
Collaborative Calendar
Shows free time and planned activities
Online Platform

FIGURE 4.4: Mind map of Social Connection

Simulation Online Garden

The last section is mostly based on one specific idea created during the brainstorming matrix. The idea is to have an online garden representing the actual garden of the users. Users can tag tasks to specific parts of the garden. The different aspects of the online garden, such as specific vegetables or the watering can, will notify users if they need attention. Another feature that could be implemented is the ability to plan your garden and see what the ideal layout would be. This could motivate users to finish tasks and remind them what has to be done in a playful way. According to the literature review, users desire to write down their tasks in an abstract way without knowing the exact details. This is why this idea also contains "Untagged Tasks" which allows users to write their abstract tasks. This mind map and idea have been translated into a sketch which is explained in Section 4.2.1.

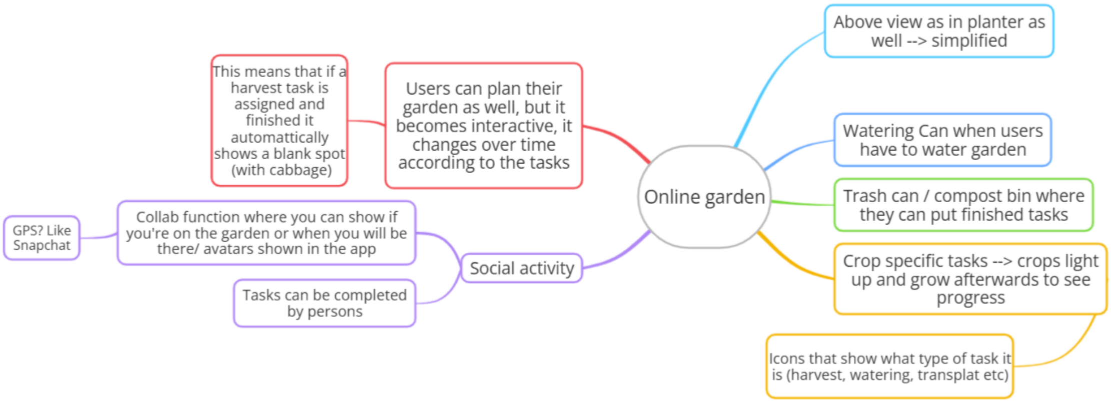

This means that if a harvest task is assigned and finished it automattically shows a blank spot (with cabbage)
Users can plan their garden as well, but it becomes interactive, it changes over time according to the tasks
GPS? Like Snapchat
Collab function where you can show if you're on the garden or when you will be there/ avatars shown in the app
Tasks can be completed by persons
Social activity
Online garden
Above view as in planter as well --> simplified
Watering Can when users have to water garden
Trash can / compost bin where they can put finished tasks
Crop specific tasks --> crops light up and grow afterwards to see progress
Icons that show what type of task it is (harvest, watering, transplat etc)

FIGURE 4.5: Mind map of the online garden category

4.1.3 Brainstorming session with a previous Creative Technology Student

Furthermore, a brainstorming session with a previous Creative Technology student was held. In this session, the mind maps and the brainstorming matrix were examined. In addition, the background research was explained to provide a clear overview of the previous knowledge. During this session, the previous brainstorming methods served as a source of inspiration. They examined certain ideas and explained their vision and ideas that came to mind. This led to a storm of ideas where parts were added or subtracted from each other's ideas to create new ideas, resulting in three concepts explained in Section 4.2.

4.2 Concepts

Based on the 45 ideas from the brainstorming matrix, the four mind maps and the brainstorming session with a previous Creative Technology student, three concepts were created.

4.2.1 The online Garden

The first concept generated was inspired by the mind map illustrated in Figure 4.5. This design [Figure 4.6] shows a garden layout with vegetables and other plants based on the real gardens of the users. Users can input tasks and link them to specific plants or a watering can. Furthermore, they can put the task in a category such as weeding, harvesting or sowing. If users do not want to tag their tasks or if they have tasks that are not linked to specific plants, they can leave the tasks in an "Untagged Tasks" list. In addition, users can plan the layout of the garden from a bird-view perspective, which gives them insight into what plants are in their garden. All tasks that were created can be found under another tab as well, which shows the tasks in different lists. Some of the participants from the user interviews mentioned their challenges with remembering to go to the garden and with all the information only in their heads. This concept could reduce these challenges and motivate gardeners to keep track of their tasks. The goal of this design is to help gardeners gain insights into their gardens by allowing them to view tasks at a glance and add tasks easily.

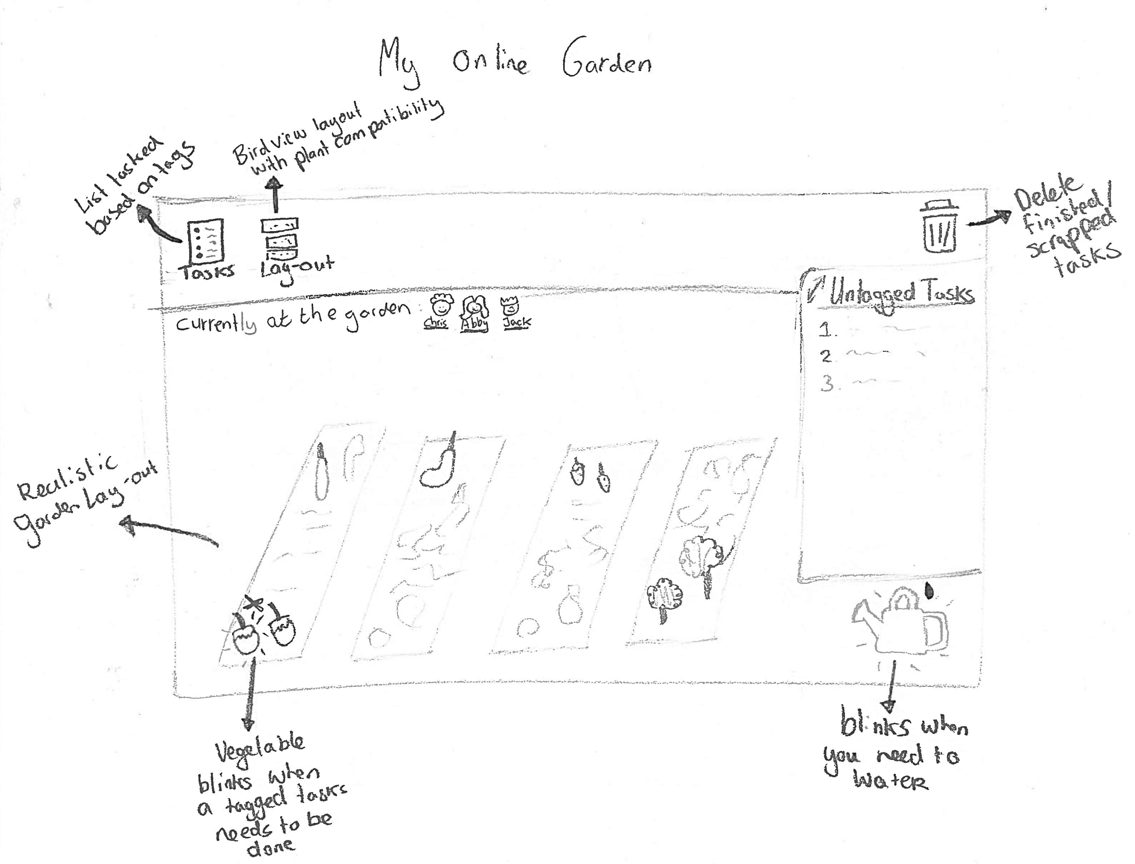

My Online Garden
List tasked based on tags
Bird view layout with plant compatibility
Delete finished/ scrapped tasks
Tasks
Log-out
Currently at the garden:
Chris
Abby
Jack
Untagged Tasks
1.
2.
3.
Realistic Garden Lay-out
Vegetable blinks when a tagged tasks needs to be done
blinks when you need to Water

FIGURE 4.6: Sketch of the Online Garden Concept

4.2.2 The online community

The second concept was partially based on the mind map illustrated in Figure 4.4. This concept [Figure 4.7] uses social connections as a motivator and reminder for task management. According to Section 2.3.2 social connections are a great motivator for gardeners. The design is an online platform where users can post pictures and texts about their progress and give tips or ask questions to other gardeners. Furthermore, gardeners can see what others in their community have posted and connect to gardeners outside their community. In addition, users can chat and plan meet-ups with fellow gardeners. Lastly, users can see their profile which shows their progress and work of previous years, which shows elements from the mind map illustrated in Figure 4.2. The goal of this design is to motivate users through online communities.

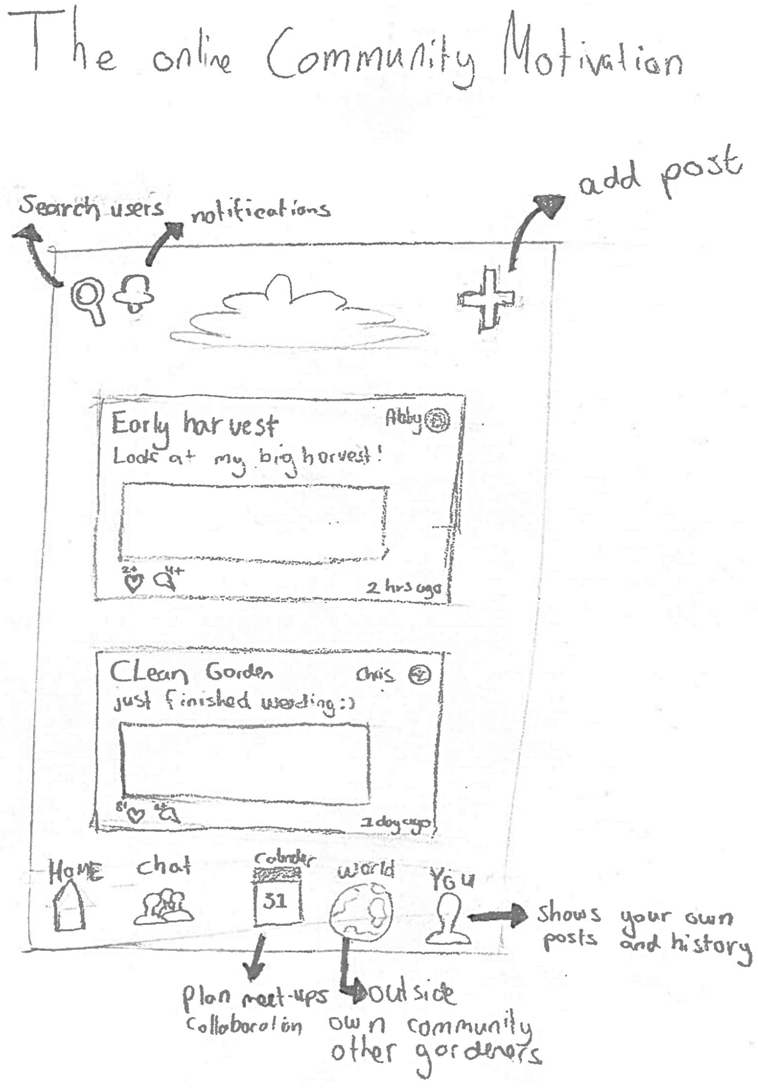


FIGURE 4.7: Sketch of the Online Community Concept

4.2.3 The Garden Game

The last concept is based on the gamify category of the brainstorming matrix [Figure 4.1]. This concept [Figure 4.8] shows an online virtual garden that grows when tasks are completed. Users can add tasks and estimate the amount of time the tasks will take. The amount of time the tasks take is the amount of "Gold" users can earn in the game. For example, if a task takes 45 minutes to complete, users earn 45 Gold. Furthermore, with this Gold they can "buy" online vegetables that grow in their virtual garden. There is a health bar that drops if users have overdue tasks. In addition, users can visit the virtual gardens of friends and the application gives recommended/expected tasks that users can add to their upcoming tasks. Lastly, users can log their progress with the camera function, which encourages gardeners to go to their garden and show the progress they have so far. The goal of this design is to stimulate task management among users in a fun and challenging way, which reminds them about their gardens.

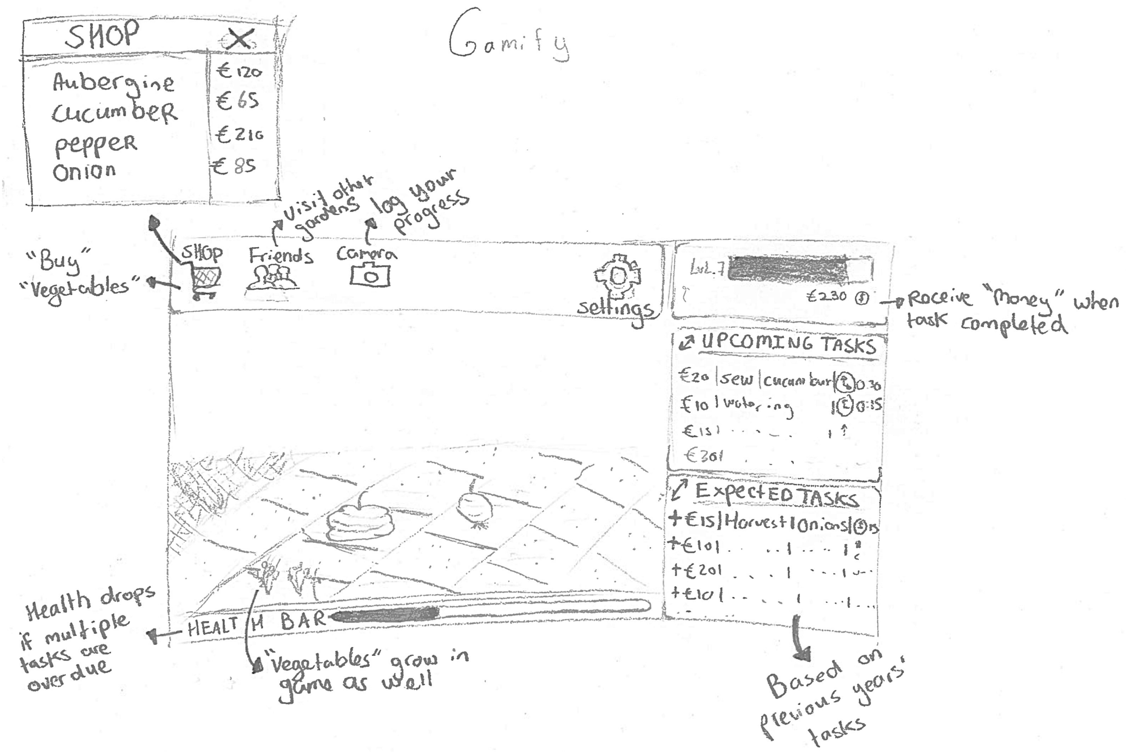


FIGURE 4.8: Sketch of the Garden Game

4.3 Selection of Final Concept

To determine which concept best suited the problem statement, preliminary requirements were created and a Harris profile for all three ideas was made to decide which concept to develop further. These requirements are based on several factors derived from the background research.

4.3.1 Preliminary Requirements

The first three requirements are based on the literature review of implementations for PTM Tools [Section 2.1.2]. Furthermore, preliminary requirement number four is derived from the user interviews. The last three requirements stem from the State of the Art.

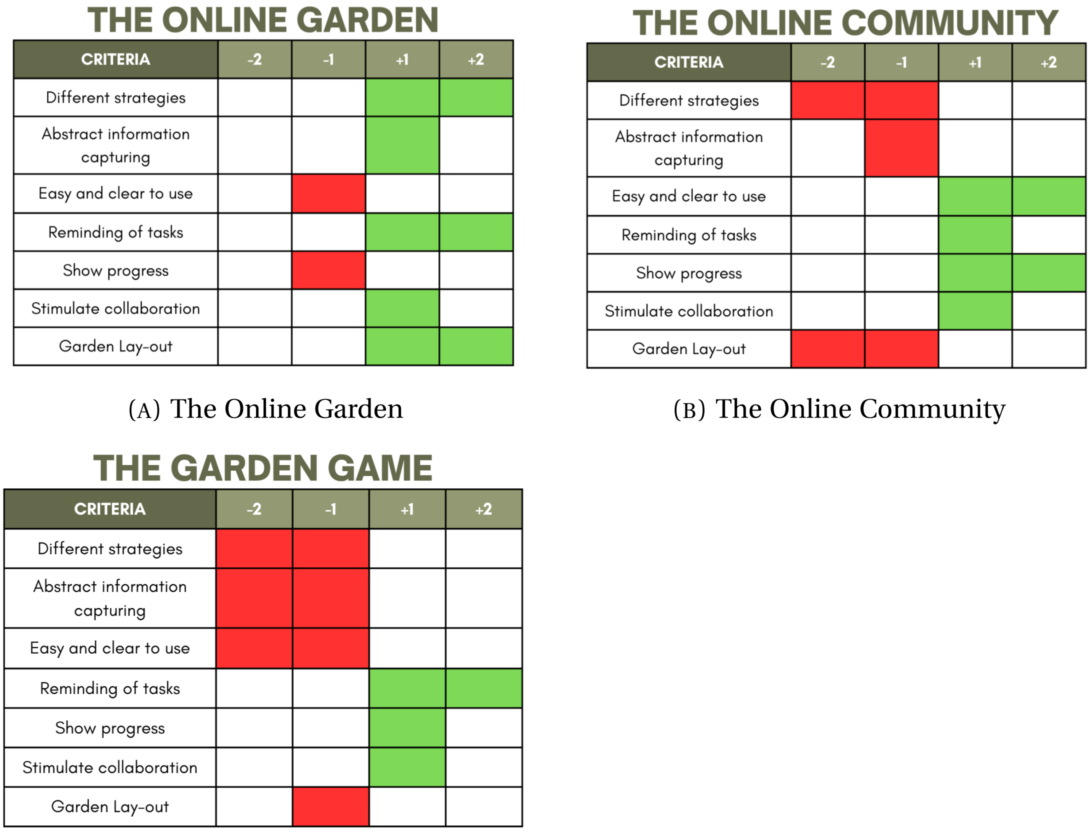
1. The design must allow gardeners to use different strategies depending on the tasks and context.
2. The design must allow users to capture information quickly without needing to know how to organize it.
3. The interface must be easy to use and have clear visualizations.
4. The design must help remind users of their garden and their garden tasks.
5. The design must show the progress of the gardeners.

6. The design must stimulate collaboration.

7. The design must show a garden layout.

4.3.2 Harris Profile

The preliminary requirements have been translated into a Harris Profile of the three concepts. The Harris Profile showed for each concept per requirement a score ranging from -2 to 2 points. If a concept strongly failed to meet the requirement it scored -2 points, if it strongly met the requirement it scored 2 points. This showed that The Online Garden concept [Figure 4.9a] meets the requirements the best.



THE ONLINE GARDEN

CRITERIA	-2	-1	+1	+2
Different strategies			🟩	🟩
Abstract information capturing			🟩	
Easy and clear to use		🟥		
Reminding of tasks			🟩	🟩
Show progress		🟥		
Stimulate collaboration			🟩	
Garden Lay-out			🟩	🟩

(A) The Online Garden

THE ONLINE COMMUNITY

CRITERIA	-2	-1	+1	+2
Different strategies	🟥	🟥		
Abstract information capturing		🟥		
Easy and clear to use			🟩	🟩
Reminding of tasks			🟩	
Show progress			🟩	🟩
Stimulate collaboration			🟩	
Garden Lay-out	🟥	🟥		

(B) The Online Community

THE GARDEN GAME

CRITERIA	-2	-1	+1	+2
Different strategies	🟥	🟥		
Abstract information capturing	🟥	🟥		
Easy and clear to use	🟥	🟥		
Reminding of tasks			🟩	🟩
Show progress			🟩	
Stimulate collaboration			🟩	
Garden Lay-out		🟥		

(C) The Garden Game

FIGURE 4.9: Harris Profile of the three concepts

4.4 Final Concept

The Harris Profile showed that The Online Garden met the most requirements. However, some requirements are considered more important, based on the findings of the background research. These requirements are ranked and further described in Chapter 5. The concepts were discussed with my supervisors and together we concluded that The Online Garden concept showed the most promise. This concept allows users to capture information quickly, contrary to the other designs where users have to think about how to represent themselves or know exactly how long a task takes. Furthermore, the concept allows users to plan the layout of their garden and see this and their tasks at a glance. Lastly, The Online Garden design has different types of views and ways users can interact with the interface. First of all, the overview page shows the garden in a visually pleasing way where users can see their tasks and garden quickly. The second page shows the garden layout from above, which allows users to plan their garden layout and see clearly where there is some free space. The third page shows the tasks ordered in different lists, which allows users to see all their tasks at once and filter or add tasks. Furthermore, an editorial page might be added that shows news and updates of the community garden. Lastly, the design might show if there are other users in the garden. This concept allows users to add and view tasks quickly and view them in combination with their garden layout. As shown in Section 2.2 related work focused on the garden layout and tasks. However, none of the examined applications combined these to see the relation between the tasks and layout at a glance.

Chapter 5

Specification

This chapter will describe the specification phase of the design process of this project. First, the gardening context and target audience will be specified. After establishing the user context, written scenarios are created to give insight into how users would interact with the prototype. Furthermore, requirements will be established using the MoScoW method. The requirements are divided into functional- and non-functional requirements.

5.1 Specification of Gardening Context

During this report, all types of gardens were included. However to ensure the best user experience and to make sure the design is feasible in the given time frame, a selection of gardens has been made. The gardens chosen for The Online Garden platform consist of small gardens up to mid sized gardens. Users must be able to see their whole garden layout at a glance. Therefore, gardens larger than $240m^2$ have been excluded. Gardens could also be too small for the platform to work nicely, thus gardens have to be at least $4m^2$.

5.2 Specification of Target Audience

Experienced gardeners can know a lot about their garden, however remembering all the tasks at the right moment can be challenging for experienced and inexperienced gardeners. Furthermore, having information about tasks in one specific place could be useful for community gardens. Therefore, community gardens were included as well. The target audience consists of all types of gardeners and community gardens.

5.3 Written scenarios

Written scenarios have been created to show how different types of gardeners would interact with the platform. This gave insight into what kind of functions the platform should have and how these should be incorporated.

5.3.1 Chris

Chris is 74 years old and likes to get his exercise outdoors while working in his backyard. He has 30 years of experience with vegetable gardening and created a plot of $250m^2$. Chris has always been a pretty forgetful and disorganized person. He has trouble remembering all his different tasks and tries to get a grip on them by writing the tasks on sticky notes. However, these notes get lost in his house, which frustrates him.

On Tuesday he decided he wanted to change his task management behaviour and looked for new ways to organize his tasks. He was reading the paper and saw an advertisement for the TuinMoes platform and decided to give it a try. He translated his garden layout on the overview page. Chris searched his house for lost sticky notes and entered these tasks in different categories in the Task tab. He does not want to put a single date for his tasks, but prefers flexibility and therefore decides to give each task a range of a couple days or weeks.

The next Sunday Chris finds the weather perfect for a day working in the garden. To prepare himself he opens TuinMoes to see what kind of tasks he can do today. He goes to the platform and sees the carrots blinking on the overview page. He clicks on the carrot and reads that they can be harvested. He goes to his garden and harvest the carrots. At the garden, he notices that his zucchini has greenfly on it. He opens TuinMoes and adds that he has to eliminate these pests in the upcoming week.

5.3.2 Jack

Jack is a freelance designer and works mostly from home. He felt lonely and decided to become a volunteer on a $150m^2$ community garden to have more social connections. The owner of the community garden told they use TuinMoes to assign tasks to other volunteers. On this platform Jack could also see when other volunteers are on the garden.

Friday afternoon he got a notification that Sarah was working in the garden. Before going, he looked at what tasks he could do and saw that he could harvest the lettuce. Arrived at the garden, Jack got confused, because there where multiple patches with lettuce. Sarah told him to look at the overview page of the TuinMoes platform to see what patch of lettuce he had to harvest. Jack knew now what to do and harvested the lettuce of the front row. At the end of his working moment, he noticed that different patches have different watering needs. He added two new tasks: watering the back patch of lettuce twice a week and watering the front row of lettuce once a week. When it got dark he said good bye to Sarah and went home satisfied.

5.3.3 Abby and Oliver

Abby just graduated and moved in together with her partner Oliver. They wanted to have a hobby together and decided to rent a vegetable garden of $50m^2$ on an allotment complex. Their neighbor has an allotment as well and gave them the tip to use TuinMoes to plan their garden layout. Abby and Oliver decided to use TuinMoes and made a layout on the grid. They discovered the Task tab, which their neighbor had not talked about. They added their tasks to different lists and divided each task and tagged who is going to do what task. Abby was happy with this function, because this helped her with knowing what Oliver had already done and what she could do.

5.4 Requirements

After establishing the specific use cases and written possible scenarios the requirements could be established. This has been done using the MoSCoW method. This method orders the requirements in four separate categories: Must have, Should have, Could have and Won't have. Lastly, the requirements were split in functional and non-functional requirements. Functional requirements could be evaluated based on their completeness and could be evaluated without user testing. Non-functional requirements consider interaction between the platform and users and therefore should be evaluated with user testing.

5.4.1 Functional Requirements

Must have

The must have requirements are critical for the prototype and therefore should all be included in the prototype. The first requirement the platform must have is that it must have a page where users can add, delete and change tasks. Users must be able to select a plant, type of task, date and add comments. This is one of the three core functions of the platform and therefore users should be able to modify their tasks and have an input feature that adds all necessary information to perform a task. The platform must have four different categories in which the tasks will be shown. Moreover, the design must have a bird's eye view of the garden layout where users can add and delete vegetables to their layout. Furthermore, the tasks must be visually shown on the main page (Overview tab) with the use of icons on the corresponding plants. Users must be able to choose for what date or period they want to visually see the icons. Lastly, the platform must save the garden layout and the tasks defined by the users. This is important because users should be able to go back to the platform and retrieve the information quickly. These functions are the base of the platform and without these functions, the platform could not be used for its purpose which is to see the tasks for a selected period combined with their garden at a glance.

Should have

The should have requirements are important but not critical for the working of the prototype. The design should have a specific color palette consisting of these four base colors: #432D12, #ECDFCC and #A5B68D. These colors can be seen as neutral and not disrupting the main purpose of the platform. Furthermore, they make the platform cohesive with the same colors on all pages. Moreover, the prototype should have at least 12 plants. It has been chosen to incorporate 12 plants because I feel that with 12 plants users can make a decent layout of their garden for the Hi-Fi prototype. Next, users should be able to specify task details later on. For example, if they want to add a date or comment. This will allow users to enter their tasks quickly, which is an important feature of task management as stated in Section 2.1.2. In addition, the platform should allow gardens from $4m^2$ up to $250m^2$. This range of gardens is chosen because I think most gardens will be in this range. Lastly, the task icons should have an explanation below the overview to make it clear to the users what the icons mean.

Could have

The requirements in the Could have category could improve the user experience and are desirable, but not important or necessary. The design could have hand-drawn vegetables. This could make the whole platform feel more cohesive while it is in the same style as the garden ground. In addition, the platform could have an editorial page, where community garden owners could upload newsletters. This was desirable to have because having updates could improve the social cohesion of community gardens. Lastly, the platform could have a function to give tasks to different community members. This could make planning in communities easier and could make users feel more involved in the community.

Won't have

These requirements are the least critical but are still nice to have. However, due to the scope of this project, these requirements were not fulfilled. First of all, it would be nice if users could change the shape of their garden layout and the garden's area. This is a nice feature and could improve the user experience because this could help gardeners visualize their gardens on the platform. However, this feature is not necessary for the prototype and is not feasible in the scope of this project. In addition, the platform would not have the ability to give notifications when other gardeners are in the garden. This could be nice to have for community gardens, however, the prototype does not focus on this and this would be hard to implement in the given time frame. Lastly, users would not be able to create an account. Having an account would allow gardeners to use the platform on different devices, however, this is not needed for the prototype and therefore I have chosen to focus on other aspects to implement.

5.4.2 Non-Functional Requirements

Must have

First of all, the prototype must help users to see their garden tasks at a glance. As described in Section 2.1.2 users wish to capture and retrieve information quickly. Furthermore, section 2.1.2 described that users have different task management strategies and PTM tools should allow users to apply these different strategies. To enhance this, different pages are created where users can see their tasks and garden in different ways. A non-functional requirement derived from this is that the prototype must allow users to apply different strategies. Lastly, the platform must enhance the scheduling and prioritizing of tasks. This is important because scheduling and prioritizing are essential to Task Management.

Should have

A Non-functional requirement labeled as Should have is that the task icons should be self-explanatory. These icons are designed to help users see their tasks in relation to their garden quickly. Therefore these icons should be self-explanatory to stimulate the visualization, which can help gardeners to see their tasks quickly.

Could have

A Could have requirement is that the platform could stimulate collaboration. As shown in Section 2.3.2 social connections are a great motivator for gardeners. The prototype would not focus on the motivators, however, stimulating collaboration could improve the user experience and therefore is taken into consideration during the realization.

Won't have

The platform would not help gardeners to broaden their gardening knowledge. It would be nice if the platform had personalized information that could help gardeners learn new gardening tricks. However, due to the scope of the project, this feature was not focused on.

Chapter 6

Realisation

This chapter describes the realization of the three pages of the platform. For each page, the design choices and the implementation methods will be discussed.

6.1 General

The platform has been built with a combination of HTML, CSS and JavaScript. Figma has been considered, because of the tight time span in which the prototype had to be created. However, due to the complexity of the connection between the pages and the broad range of possible actions users had to be able to do, Figma did not seem as the right fit. This, in combination with the previous knowledge I already have with HTML, CSS and JavaScript is why this has been chosen.

Due to the tight timeframe, ChatGPT has been used to search for bugs in the code and to understand the implementation-specific parts. This has been used for all HTML, CSS and JavaScript code. This allowed me to write the complete platform in a tight time frame. The code used to create the platform can be found in Appendix F.

The pages were created in a specific order, which allowed collaboration between the pages. First, the Tasks page has been created. Next the Layout page and lastly the Overview page which uses input from the Layout and Tasks page.

6.2 Style

As mentioned in Section 5.4 the platform must have a specific color palette. To make sure this is consistent on each page three variables containing these colors have been created in CSS. This made it easy to obtain each color when needed. Next, two fonts have been chosen, one for the titles and one for the text of the body, which are Poppins and Lustria.

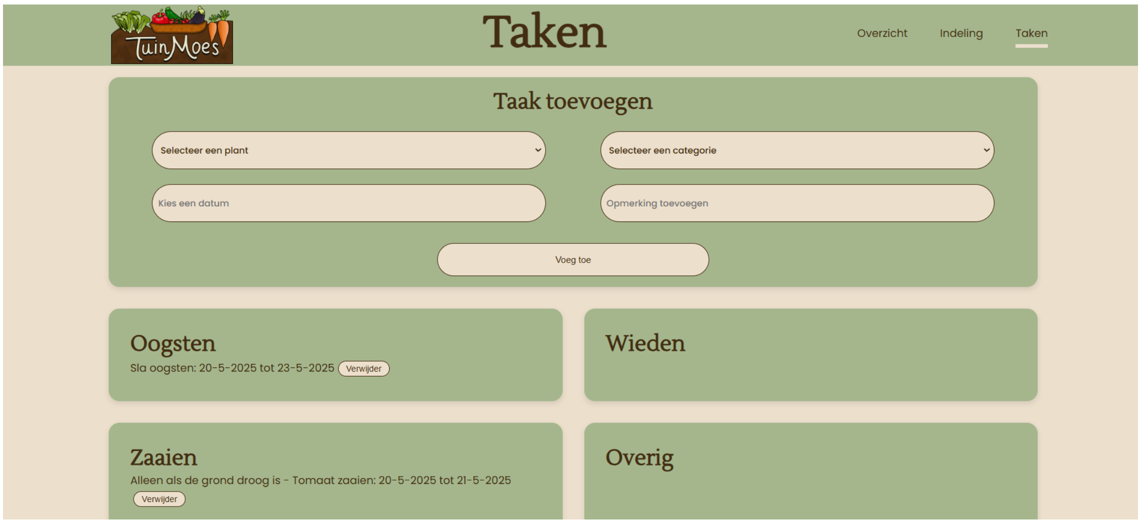
This color palette is accompanied by hand-drawn images to create a cohesive design. The images of the vegetables and trashcan were created with Photoshop. They all were made with the same line thickness and masks were created to add highlights and shadows to the picture. Furthermore, a logo with a name has been designed.

The logo has been made with Photoshop and previously drawn vegetables are incorporated into the logo.

Lastly, four emojis have been chosen, each representing a specific task category, such as harvesting, weeding, sowing or other. A fifth emoji has been added, which shows if there are multiple different task categories for a specific vegetable in a specific time range.

6.3 Tasks Page

The first page that has been created is the tasks page, as shown in Figure 6.1. On this page, there is one input box where users can add tasks. They can specify the type of tasks: harvesting, weeding, sowing and other. Furthermore, they can select the vegetable or fruit the task has to do with. Next, a date or a date range can be selected and lastly, users can add a comment if they want to. After adding a task it will be displayed in the corresponding task category underneath the input box.



Taken

Taak toevoegen

Oogsten

Sla oogsten: 20-5-2025 tot 23-5-2025

Wieden

Zaaien

Alleen als de grond droog is - Tomaat zaaien: 20-5-2025 tot 21-5-2025

Overig

FIGURE 6.1: Tasks Page

6.3.1 Design Choices

Four task categories were created, because three types of tasks are the most typical for gardening and in addition there is one category where users can put additional tasks. The first design choice was to have an input box for each category separately, however, this felt counter-intuitive and was not necessary. Therefore the layout was changed to one input box as described above. Furthermore, a calendar where users can select one date or a range of dates has been chosen. This allows users flexibility in their planning, while still knowing which task to do when. Lastly, I decided to only add delete buttons and no edit buttons. There would be no big difference in the workload for users to edit existing tasks in contrast to deleting and adding new tasks. The platform should be easy and clear to use and therefore I feel that having fewer buttons would make the interface more intuitive for the gardeners.

6.3.2 Implementation

The layout of the tasks page is made with a combination of HTML and CSS. The boxes have the same style because each box has the class "task-box". Furthermore, the id "task-input" has been given to the input box, which makes sure that the layout of the input box is in a grid style, with the title on top and the add button below the options. This is done with display: flex and the class task-select has a display: grid and a grid-template-columns: 1fr 1fr, which creates two columns and two rows. Lastly, the display of the four categories is handled with the class "task-grid", which ensures the categories are displayed in a grid of two by two.

Besides the HTML and CSS files, a JavaScript file has been created. First, Flatpickr is initialized, which handles the data selection in the input box. In the function addTask the input from the HTML file is retrieved and stores the data in local storage. This function also handles the display of the task in the corresponding task category box and calls the deleteTask function which ensures that tasks are removed when users click on the delete button. The function loadTasks makes sure to load tasks from local storage after revisiting the page and displays them in the right category. Besides these main functions, there is a helper function normalizeDate, which reformats the date from the US display to the NL display style. Lastly, window.onload calls the loadTasks function for each category.

6.4 Layout Page

The second page created is the Layout page, as shown in Figure 6.2. On this page, users can plan the layout of their garden. They can select vegetables/fruits from a drop-down selection tool and drag them onto the garden grid. If they harvest a plant from their garden or make a mistake during planning, they can drag the plant to a trashcan and the plant would be removed from the grid.

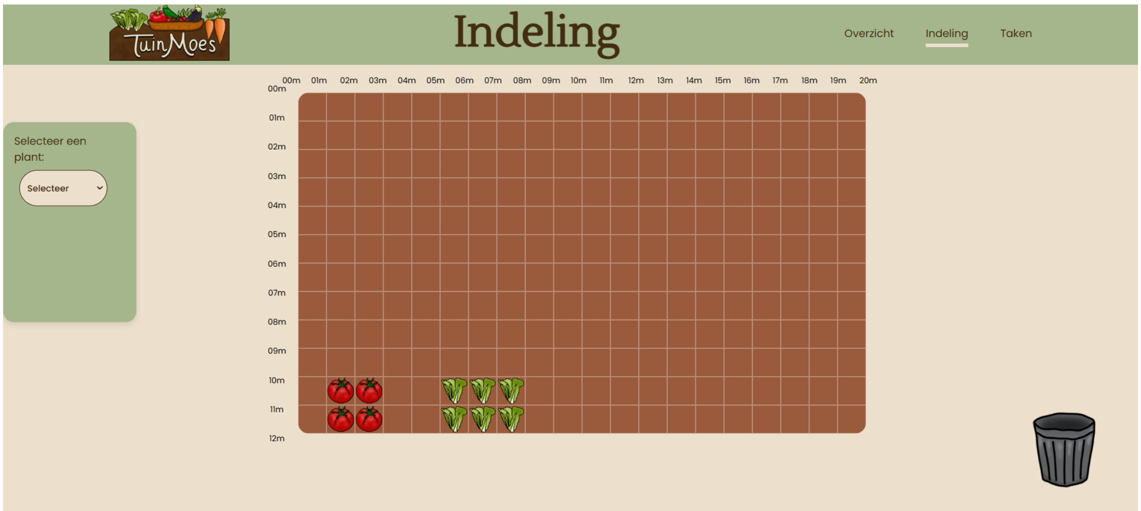

TuinMoes
Indeling
Overzicht
Indeling
Taken
Selecteer een plant:
Selecteer
00m 01m 02m 03m 04m 05m 06m 07m 08m 09m 10m 11m 12m 13m 14m 15m 16m 17m 18m 19m 20m
00m 01m 02m 03m 04m 05m 06m 07m 08m 09m 10m 11m 12m

FIGURE 6.2: Layout Page

6.4.1 Design Choices

To ensure most gardeners can use the platform the prototype has a grid with an area of $240m^2$. This grid is divided into cells of $1m^2$in which a plant can be placed. Alongside the grid a ruler has been created which helps users identify the placement of specific vegetables and shows users the scale of the grid. Without a ruler, it can be unclear to users what the size of the garden is. Furthermore, all plants and the trashcan have been hand-drawn, to create a cohesive and appealing layout.

6.4.2 Implementation

The ground square and the selection tool have been created using HTML and CSS. Furthermore, the HTML file gives structure to the layout of this page and ensures the right placement of all the parts.

Besides the HTML and CSS files, a JavaScript file has been created. This file creates the grid structure on top of the ground square and the ruler alongside the grid. It makes sure that users can use the drop-down menu and that the selected plant is displayed underneath the drop-down menu. Furthermore, the file handles the dragging and dropping of plants into the grid and the dropping and dragging from the grid to the trashcan. Moreover, it stores the information of the vegetable with a unique ID, the image source and in what cell the plant is placed. Lastly, it loads all the vegetables into the grid.

6.5 Overview page

Lastly, the Overview page has been created, as shown in Figure 6.3. This is the first page users see when opening the platform. This page shows the ground grid with the vegetables in a first-person perspective instead of a bird's eye perspective. On this page, users can enter a period and see their tasks linked to the plants displayed on the grid for the selected period. For this, five icons have been chosen, each representing a category or multiple categories. If a gardener has tasks for the selected period the corresponding icons are shown on the corresponding plants. Users can click on the icons to see more information, such as the plant, the task, the period and any comments. This overview helps to see all tasks visualized in the garden at a glance.

6.5.1 Design Choices

For this page, a first-person perspective has been created, which can make the garden feel more realistic. The virtual garden aimed to be displayed in the same way the user sees their real garden when they are standing in front of it. Instead of only showing today's date, a selection tool has been created. A selection tool has been chosen because this gives users more freedom to see a period or to see their tasks for a future date.

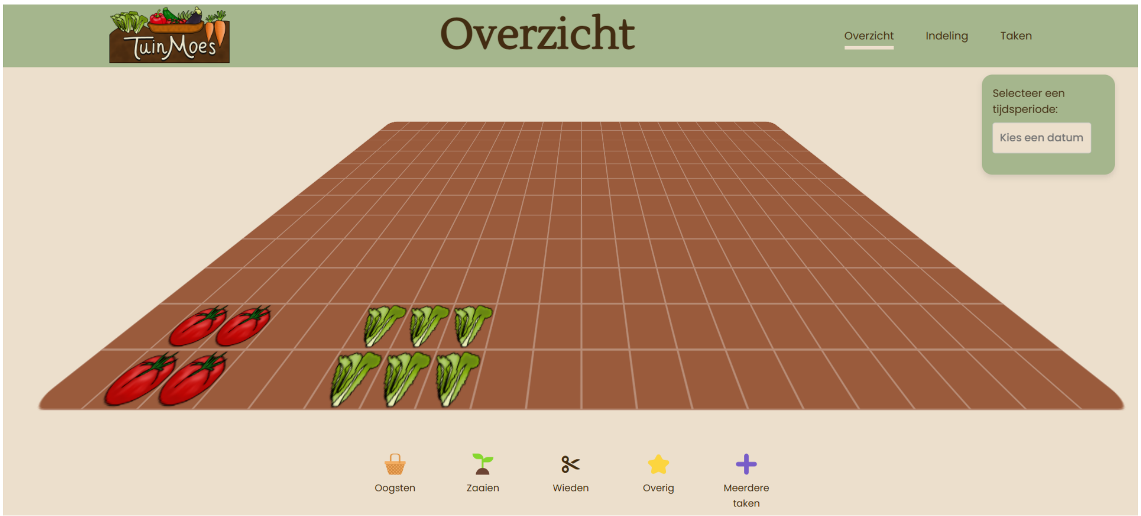


FIGURE 6.3: Overview Page

6.5.2 Implementation

The overview HTML file handles the placement of all the elements and the structure of the layout. Secondly, the CSS file tilts the ground square to create the first-person perspective and contra tilts the vegetables to make them pop a bit. Lastly, the JavaScript file handles the interaction on the page. The grid and the vegetables as entered on the Layout page are imported and shown on the Overview page. The selection of time is handled with Flatpickr in the same way as on the Tasks page. Furthermore, the JavaScript file handles the placement of the icons and the description of the icons underneath the garden. Lastly, the file ensures that when users click on an icon a Popup with task details is displayed.

Chapter 7

Evaluation

This chapter will focus on the evaluation of the requirements specified in Chapter 5. First, the functional requirements will be evaluated. Next, the non-functional requirements will be assessed through user testing. The design of the user testing will be described. Then, the results and their significance will be analyzed using statistical tests, including box plots, the Mann-Whitney U test and a one-tailed independent t-test.

7.1 Evaluating functional requirements

After creating the prototype, the functional requirements were evaluated. This was done before evaluating the non-functional requirements to ensure that the platform works properly before user testing. The functional requirements created in Section 5.4.1 have been evaluated based on their completeness.

Must have

The first Must have requirement stated that users must be able to add tasks and select a plant, type of task, date and add comments. Users must be able to edit and delete tasks as well. This requirement was mostly met, however, editing the tasks was not possible with this prototype. Due to the scope of the project and complications while trying to implement this feature, I have decided not to implement it in the prototype. The feature was not essential for using the prototype and if users made a mistake or wanted to edit the task, they could delete the existing task and add a new task with changed information. Furthermore, four categories were created and users could add or delete vegetables to the layout on the Layout page. The Must have requirements of the Overview page were met as well, such as the selection tool for the dates and the icons. Lastly, the platform saves the garden layout and the tasks.

Should have

The platform's color palette specified as a Should have requirement is incorporated. Another Should have requirement was the implementation of twelve vegetables. However, for the prototype eight vegetables have been created. This allowed users to test the prototype and gave them an idea of how they could use the platform themselves. If users wanted to replicate their garden this would not be possible with the current prototype, but I have decided that this would not be needed for user testing. Furthermore, social features such as an editorial page and dividing tasks were not implemented. I decided that the focus of the prototype would not lie on the social parts.

Could have and Won't have

The Could have requirement for hand-drawn vegetables was incorporated into the current prototype. All other requirements from the Could have and Won't have categories focus on social relations and personalization. Even though these are desirable and nice to have, due to the scope of the project I have decided not to focus on these aspects and therefore they are not met.

7.2 Evaluating non-functional requirements

The goal of the design was to help gardeners gain insight into their garden and their garden tasks, where users can see and capture their information quickly. The prototype claims that users can view their tasks in correlation with the vegetables at a glance using the Overview page. This evaluation tested if users experience more grip on their tasks due to the overview page.

7.3 Design of user evaluation

7.3.1 Objective

The aim of the user evaluation is to determine whether the Overview page improves the perceived and actual overview users have in managing their gardening tasks. The evaluation focused on one key non-functional requirement: The prototype must help users see their garden tasks at a glance. This will be tested with two methods:

1. **Objective method:** Measuring how accurately participants could retrieve specific tasks.
2. **Subjective method:** Measuring how confident participants felt about knowing their tasks.

7.3.2 Participants

Participants were recruited from allotment gardens around Enschede and through reaching out to peers and friends. For this user test, only people with a vegetable garden and people with an affinity for gardening were chosen as participants. This focus helped to understand the impact the platform has on the intended target group. Including people with an affinity for gardening made it possible to find enough participants to user test in the scope of the project. The participants were divided into two separate groups:

1. **Group A:** Participants who had access to all three pages (Overview, Layout and Tasks pages).

2. **Group B:** Participants who had access to the Layout and Tasks pages.

This division allowed to test if the overview page that Group A could see provided additional value in comparison to the task list page that Group B could see. The participants did not know in which group they were placed and did not know the objective of the evaluation until after the user test.

7.3.3 Set-up

The user tests took place at various locations, such as the homes of users, the library or the university. To ensure the participants felt comfortable, they could choose where they wanted to test. The testing took around 30 minutes, including the input of tasks and the questionnaire. Before participants entered the testing, they were asked to read the information letter (Appendix G) and sign the consent form (Appendix H). If needed they could ask questions or ask me to explain certain aspects further.

The user test consisted of three tasks and six questionnaire questions. First of all participants of Group A and Group B were asked to enter a task list consisting of 24 tasks into the task list page. When they were finished Group A was asked to go to the Overview page, while Group B had to stay on the task list page. The second task they received was to try to remember as many tasks as possible for a specific date, the 21st of March. Group A could use the date filter and had to use the overview page with the icons to remember the tasks, Group B had to find the tasks in the columns of the Tasks page. After a minute they could not look at the pages anymore and were guided to the questionnaire. After the task, they had to answer two questionnaire questions, which asked about their confidence in knowing the tasks and how much time the tasks would take them for that specific date. The third task had the same structure as the second task, but now instead of one date, they were asked to find the tasks corresponding to a date range: all tasks for April. Again they got one minute and the same questions but now for April. Furthermore, they were asked how easy they experienced to find the tasks for the specific dates. Lastly, I allowed both groups to explore the whole platform and asked them to give general feedback. This feedback is useful for understanding how users interact with the platform and which aspects they like and which aspects they do not like.

7.3.4 Hypothesis

The hypotheses for this evaluation are as follows:

Perceived overview

Null Hypothesis H0
There is no difference in perceived overview between Group A and B

Alternative Hypothesis H1

Group A will have a higher perceived overview than Group B

Actual accuracy

Null Hypothesis H0
There is no difference in actual accuracy between Group A and B

Alternative Hypothesis H1
Group A will have a higher accuracy score than Group B

7.4 Results

In this section, the results of the user evaluation are presented. This is done in three main parts: the perceived overview based on Likert scale questions, actual overview accuracy and general feedback provided by the participants.

7.4.1 Perceived Overview

Participants were asked to answer five questions concerning their perceived overview. These questions were ordered on a Likert scale ranging from 1 to 5, where 1 indicated "strongly disagree" and 5 indicated "strongly agree". For the results, questions that had similar aspects were grouped together, leading to three different groups:

1. Feeling confident about knowing the tasks for the specific dates
2. Feeling confident about estimating the time the tasks will take to complete
3. Finding it easy to find the tasks for the right date.

Perceived confidence in knowing the tasks

The first category describes the participants' perceived confidence in knowing their tasks. In the table below the scores are combined for the two similar questions.

TABLE 7.1: Confidence in knowing the tasks

Question 1:		Question 2:		Combined Questions		Ranks:	
Group A	Group B	Group A	Group B	Group A	Group B	Group A	Group B
2	2	1	1	3	3	2	2
2	2	1	1	3	3	2	2
2	2	2	2	4	4	5,5	5,5
3	2	2	2	5	4	8,5	5,5
2	2	4	2	6	4	11	5,5
4	2	2	2	6	4	11	5,5
3	2	3	3	6	5	11	8,5
4	3	3	2	7	5	13,5	8,5
4	3	4	4	8	7	15,5	13,5
5	4	4	4	9	8	17	15,5

Furthermore, a box plot was created, showing the minimum, maximum, median and mean of each group. The horizontal lines inside the boxes show the medians and the X's show the means. As illustrated in Figure 7.1, the average confidence about knowing tasks is higher in Group A in comparison to Group B. The median is two points higher in Group A in comparison to Group B.

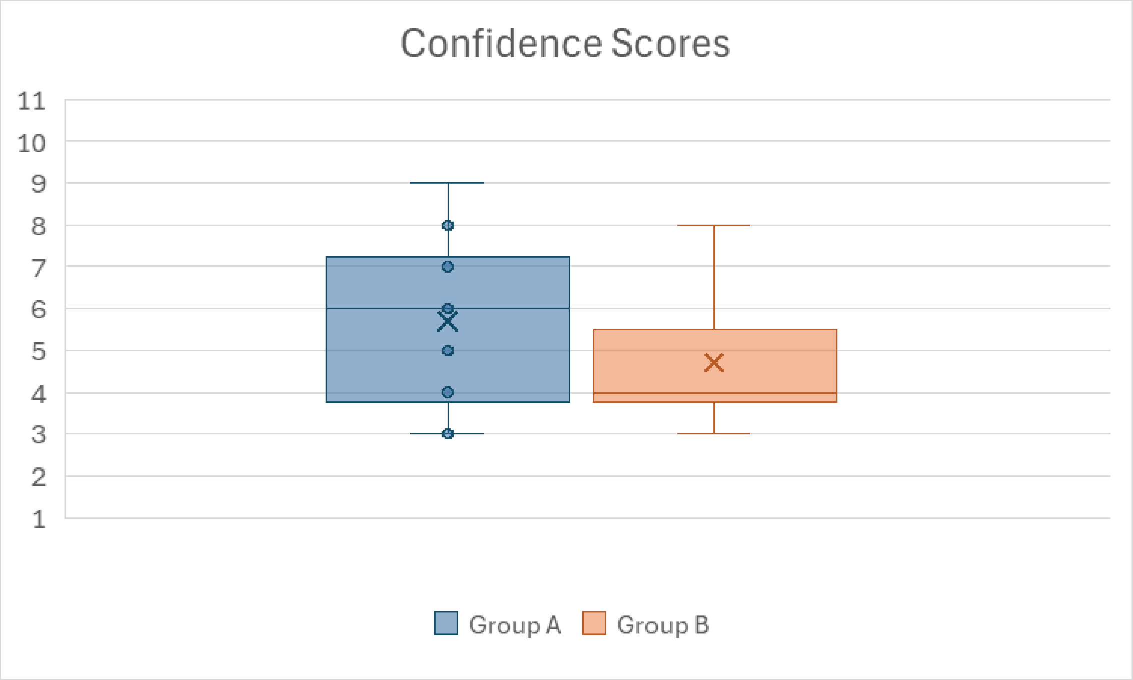


FIGURE 7.1: Box plot of confidence scores

The box plots shows that there is a difference in confidence scores between Group A and Group B. To see if there is a significant difference between the two groups a statistical test was performed. The Likert Scale is a form of ordinal data, as ordinal data is ranked on categories. Ordinal data is considered to be not normally distributed, because the data is discrete and has separate categories, unlike continuous data where the numbers themselves are the outcomes. This means this data had to be tested with a non-parametric test, which is designed for ranked data.

The Mann-Whitney U test

The Mann-Whitney U test is chosen to test the significance of the Likert scale data. This test works well with small sample sizes and compares the ranks between two groups. For this test, the data of both groups is combined and each score gets a rank as shown in Figure 7.1. If scores are tied, they share the same rank and therefore get the two ranks divided by the multiple same scores. For example, scores 7 and 7 are tied and share ranks 13 and 14, thus they both will be ranked 13,5. The Mann-Whitney U test exists out of six different steps:

1. Compute the rank for each value
2. Compute the sum of ranks for Group A (R_1) and Group B (R_1)
3. Compute the U value for both groups, using the formulas:

   $$U_1 = n_1 n_2 + \frac{n_1(n_1+1)}{2} - R_1, \quad U_2 = n_1 n_2 + \frac{n_2(n_2+1)}{2} - R_2$$

   In which n_1 is the number of participants in Group A and n_2 is the number of participants in Group B, which in this case is both ten.
4. Determine the smaller U
5. Compare the U to the critical value from the U-table at a significance level of $\alpha = 0.05$
6. Reject or retain the H0 hypothesis

The ranks as mentioned in step 1 are shown in Table 7.1. This led to $R_1 = 97$ and $R_2 = 172$. Filled out the formulas would look like this:

$$U_1 = (10)(10) + \frac{10(10+1)}{2} - 97 = 58$$

$$U_2 = (10)(10) + \frac{10(10+1)}{2} - 72 = 83$$

The smallest U-value is 58, since 58 < 83. Given for $n_1 = 10$ and $n_2 = 10$ and $\alpha = 0.05$ the critical value is 23, which was found in the U-table. If the smallest U value is smaller than the critical value the H0 hypothesis is rejected, otherwise it is retained. Since 58 is not smaller than 23 the H0 hypothesis is retained and therefore there is no significant difference in confidence scores between Group A and Group B.

Perceived confidence in estimating task completion time

The second category describes the participants' perceived confidence in estimating the time it takes to complete the tasks. In the table below the scores are combined for the two similar questions.

TABLE 7.2: Confidence in estimating the completion time of the tasks

Question 3:		Question 4:		Combined Questions		Ranks:	
Group A	Group B	Group A	Group B	Group A	Group B	Group A	Group B
1	1	1	1	2	2	2,5	2,5
2	1	2	1	4	2	7	2,5
2	1	2	1	4	2	7	2,5
3	2	2	2	5	4	11	7
2	2	3	2	5	4	11	7
3	2	2	2	5	4	11	7
4	3	3	3	7	6	15,5	13,5
4	3	3	3	7	6	15,5	13,5
5	4	4	4	9	8	19	17,5
5	4	5	4	10	8	20	17,5

Furthermore, a box plot as shown in Figure 7.2 was created, showing that the average perceived confidence about knowing tasks is slightly higher in Group A in comparison to Group B. The median in Group A is higher by one point.

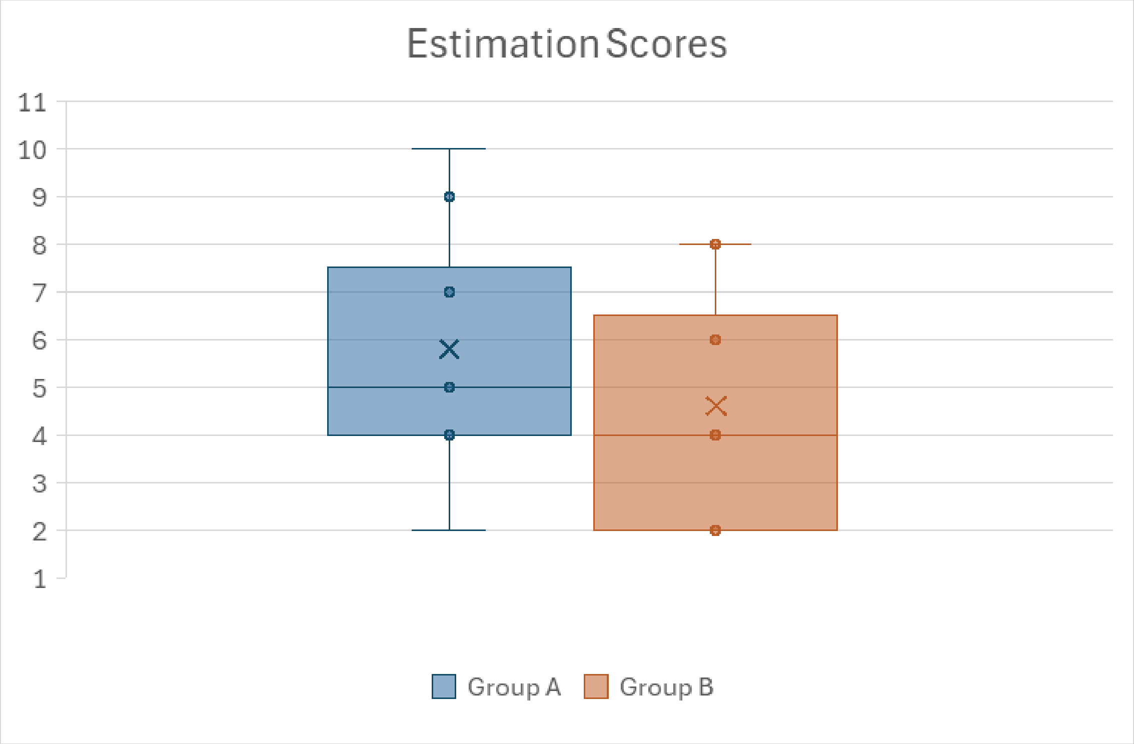


FIGURE 7.2: Box plot of estimation scores

To test the statistical significance the Mann-Whitney U test was conducted. The ranks for each combined value are shown in Table 7.2. Computing the sum of ranks for Group A and Group B gives $R_1 = 119$ and $R_2 = 90,5$. Next, we compute $U_1$and U_2 using the formulas:

$$U_1 = (10)(10) + \frac{10(10+1)}{2} - 119 = 35,5$$

$$U_2 = (10)(10) + \frac{10(10+1)}{2} - 90,5 = 64,5$$

The smallest U-value is 35,5. Since 35,5 is greater than the critical value of 23, the null hypothesis is retained and thus there is no significant difference in estimation scores between Group A and Group B.

Finding it easy to find the tasks for the right date

The last category tested with the Likert scale describes how easy it was for the participants to find the tasks for the right date. This was one question where participants could answer the questions for both completed tasks. In the table below the scores are given for Group A and Group B with the corresponding ranks.

TABLE 7.3: Ease of finding tasks for the right date

Question 5:		Ranks:	
Group A	Group B	Group A	Group B
2	1	2	1
4	3	5,5	3
4	4	5,5	5,5
4	4	5,5	5,5
4	4	5,5	5,5
5	4	14,5	5,5
5	4	14,5	5,5
5	4	14,5	5,5
5	4	14,5	5,5
5	5	14,5	14,5

Furthermore, a box plot as shown in Figure 7.3 was created, showing that the average confidence about knowing tasks is slightly higher in Group A in comparison to Group B. However as shown in the box plot, there are multiple outliers for both categories.

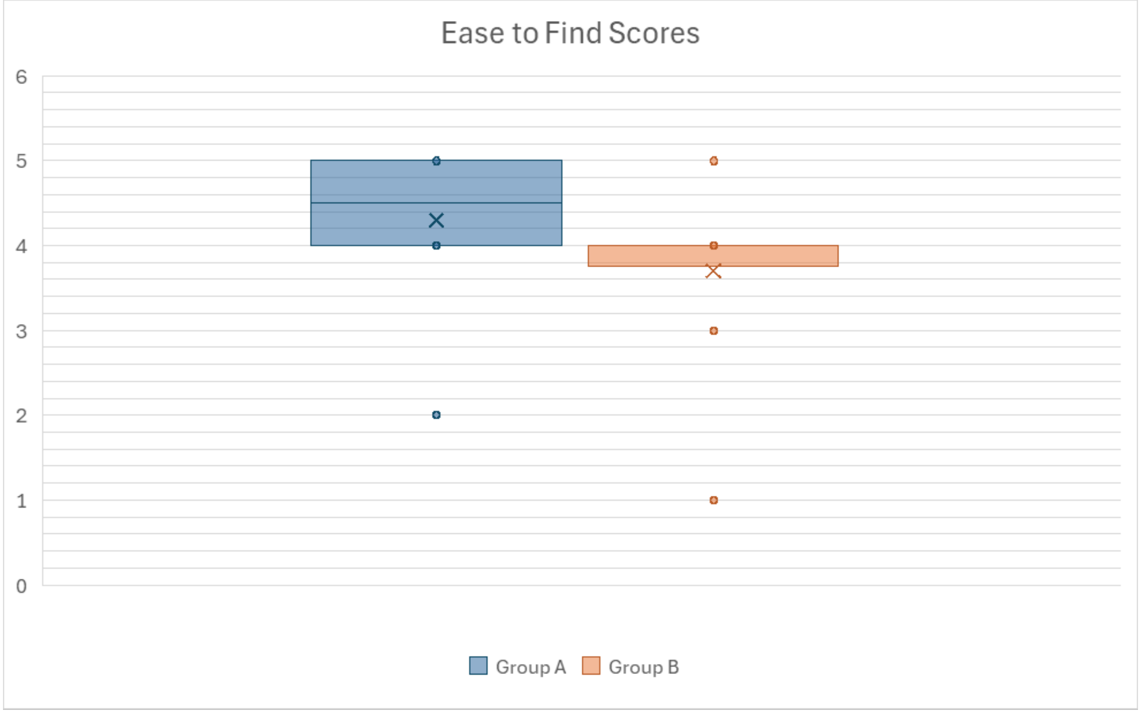


FIGURE 7.3: Box plot of ease of finding tasks for the right date

To test the statistical significance the Mann-Whitney U test was conducted. The ranks for each combined value are shown in Table 7.3. Computing the sum of ranks for Group A and Group B gives $R_1 = 96,5$ and $R_2 = 57$. Next, we compute $U_1$and U_2 using the formulas:

$$U_1 = (10)(10) + \frac{10(10+1)}{2} - 96,5 = 58,5$$

$$U_2 = (10)(10) + \frac{10(10+1)}{2} - 57 = 98$$

The smallest U-value is 58,5. Since 58,5 is greater than the critical value of 23, the null hypothesis is retained and thus there is no significant difference in estimation scores between Group A and Group B.

7.4.2 Actual accuracy of overview

Besides the perceived confidence of having a grip on tasks, the grip on tasks was also tested with an objective method. After one minute participants answered two questions and after that, they were asked to recall the tasks they had assigned for a specific date or date range. If they answered it correctly they got a +1 points, if they only remembered half of it or if they skipped a question they got 0 points, if they recalled the wrong task(s) they got -1 points. This resulted in two scores, one for the specific date of 21st of March and one score for the date range of the month of April. These scores were combined and ranked as shown in Table 7.4.

TABLE 7.4: Accuracy scores for recalling the right tasks

Combined Accuracy Score	
Group A	Group B
-3	-7
0	-4
0	-2
2	-2
3	0
4	3
5	3
8	5
9	5
15	5

Furthermore, a box plot as shown in Figure 7.4 was created. As illustrated the mean and median of the accuracy lay higher in Group A with two points. This means that on average Group A can remember two extra vegetables correctly.

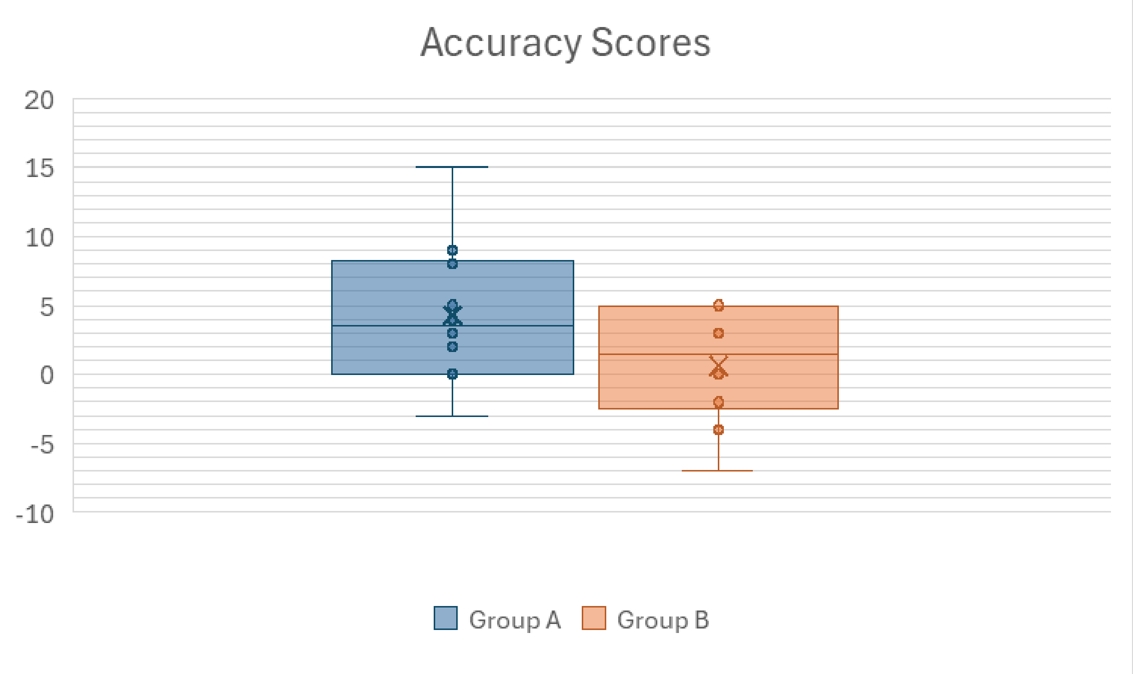


FIGURE 7.4: Box plot of Accuracy scores

To determine what type of statistical test fits the best, the accuracy scores had to be tested on normality first. This was done by conducting the Shapiro-Wilk test which showed if the data was normally distributed or not.

Shapiro-Wilk Test

The Shapiro-Wilk test evaluates whether a given sample comes from a normally distributed population. Below are the detailed steps, calculations, and results for conducting the Shapiro-Wilk test. This test was done for Group A and Group B separately.

TABLE 7.5: Calculations for differences between data points for Group A

Calculation	**Result**	a **Coefficients**	$a * result$
$x_{10} - x_1$	18	$a_1 = 0.5739$	10.3302
$x_9 - x_2$	9	$a_2 = 0.3291$	2.9619
$x_8 - x_3$	8	$a_3 = 0.2141$	1.7128
$x_7 - x_4$	3	$a_4 = 0.1224$	0.3672
$x_6 - x_5$	1	$a_5 = 0.0399$	0.0399
			$+ = 15{,}412$

The Shapiro-Wilk test consisted of seven steps:

1. First, the ordered scores as shown in Table 7.4 were assigned to a x value ranging from 1 to 10. For Group A this was the following:

$$x_1 = -3,\ x_2 = 0,\ x_3 = 0,\ x_4 = 2,\ x_5 = 3,\ x_6 = 4,\ x_7 = 5,\ x_8 = 8,\ x_9 = 9,\ x_{10} = 15$$

2. Next, the differences between each ordered score and its counterpart was calculated as shown in Table 7.5.

3. Furthermore, the five corresponding a values were looked up in the Shapiro-Wilk Coefficients table for $n = 10$.

4. Moreover, b^2 was calculated which is the sum of $a * result$ squared. For Group A, the value of b^2 is $15{,}412^2$ which is $237{,}529744$.

5. Next, the Sum of Squares (SS) was computed, in which the mean x was deducted of each x_i and squared. The sum of this is the SS value, which was for Group A $248{,}1$.

6. The W value could be computed with this formula:

$$W = b^2/SS$$

For Group A the W-value was

$$W = 237{,}529744/248{,}1 = 0.957$$

7. Lastly the estimated p - value for $n = 10$ and $W = 0.957$ could be found in the Shapiro-Wilk p - values table. The p - value for Group A is between 0.5 and 0.9. If the p - value is greater than a of 0.05 the null hypothesis is retained and thus the data is normal distributed.

The same calculations will be done to test if the data of Group B is normal distributed. First, the ordered scores as shown in Table 7.4 will be assigned to a x value for Group B:

$$x_1 = -7,\ x_2 = -4,\ x_3 = -2,\ x_4 = -2,\ x_5 = 0,\ x_6 = 3,\ x_7 = 3,\ x_8 = 5,\ x_9 = 5,\ x_{10} = 5$$

Next, the differences between each counterpart will be calculated, which is shown in the Result column in Table 7.6. The Sum of the $a * result$ is the corresponding b -value. The Sum of Square (SS) for Group B is 162.4 . This will led to the following W-value:

$$W = 12.0791^2/162.4 = 0.898$$

Lastly, the estimated p-value for $n = 10$ and $W = 0.898$ for Group B is between 0.1 and 0.5. The range of the p-value is greater than the value a of 0.05, which means the null hypothesis is retained and the data is normal distributed.

TABLE 7.6: Calculations for differences between data points for Group B

Calculation	**Result**	a **Coefficients**	$a * result$
$x_{10} - x_1$	12	$a_1 = 0.5739$	6.8868
$x_9 - x_2$	9	$a_2 = 0.3291$	2.9619
$x_8 - x_3$	7	$a_3 = 0.2141$	1.4987
$x_7 - x_4$	5	$a_4 = 0.1224$	0.612
$x_6 - x_5$	3	$a_5 = 0.0399$	0.1197
			$+ = 12.0791$

One-tailed Independent t-test

The accuracy scores for recalling tasks are normally distributed for Group A and Group B. An independent t-test was chosen to compare the means of two randomized groups on the same variable, which is the score of recalling tasks for this data . This test was a one-tailed t-test, because the alternative hypothesis stated that the accuracy score of Group A is greater than the accuracy score of Group B. The dependent variable is the accuracy score and the independent value is whether or not participants could see the overview page. The independent t-test used the means and variances of both groups. Variances showed how much the data is spread and deviates from the group's mean. The mean could be calculated as the sum of all values divided by n. The variances were computed using this formula:

$$s^2 = \frac{\sum(X_i - \bar{X})^2}{n-1}$$

The computed means and variances for Group A and Group B can be found below in Table 7.7 .

TABLE 7.7: Means and Sample Variances

	Group A	**Group B**
Means	4.3	0.6
Sample Variances	27.567	18.044

The variances are reasonably close to each other and therefore the pooled variance could be calculated. The pooled variance was calculated using this formula:

$$s_p^2 = \frac{(n_A - 1)s_A^2 + (n_B - 1)s_B^2}{n_A + n_B - 2}$$

For this test the formula becomes:

$$s_p^2 = \frac{(10-1)27.567+(10-1)18.044}{10+10-2} = 22.806$$

The last step was to calculate the t-value using this formula:

$$t = \frac{\bar{X}_A - \bar{X}_B}{\sqrt{s_p^2\left(\frac{1}{n_A}+\frac{1}{n_B}\right)}}$$

For this test, the formula became:

$$t = \frac{4.3-0.6}{\sqrt{22.806\left(\frac{1}{10}+\frac{1}{10}\right)}} = 1.732$$

Next, the degrees of freedom was calculated as: $df = nA + nB - 2$, which for this test was 18 . The significance level for this test is $\alpha = 0.05$. The critical t-value could found in a t-test table for $df = 18$ and $\alpha = 0.05$, which was 1.734. Since the critical t-value was greater than the calculated t-value (1.734 > 1.732) the null hypothesis was retained, indicating there was no significant difference between Group A and Group B.

7.4.3 General Feedback

At the end of user testing, participants were asked to provide general feedback. The feedback is divided into four parts, focusing on the three sections of the platform: the Overview page, the Tasks page and the Layout page. Additionally, feedback regarding other aspects of the platform was given.

Overview Page

Half of the participants had difficulties with seeing the back part of the garden. This made it challenging to identify the plants in that area, and especially the task icons were hard to distinguish. Moreover, some of the participants had trouble with the task icons. They first had to look up what the icons meant and noted that the icon for weeding (now a scissor) could be better described with another type of icon. The icon for multiple tasks was self-explanatory. However, participants mentioned that the icon does not help them see tasks at a glance because they still had to click on it if they wanted to know their tasks. Furthermore, participants would have liked to have other filtering options besides the date, for example on the type of task or plant. Lastly, one of the participants desired the pop-up to show up by hovering over the plants instead of needing to click on a specific plant.

Tasks Page

Participants mentioned that the filter option as displayed on the Overview page would be nice to have implemented on the Tasks page as well. In addition, they would have liked to tick off tasks instead of deleting them because this made them feel more accomplished. Seeing the history of finished tasks was mentioned by one of the participants as well, this could help them to see what they have already done and what type of tasks they had last year. Multiple participants noted that the list of vegetables and tasks was not in alphabetical order, which bothered them when inputting tasks. Lastly, adding their own categories besides the already existing categories was something desired by multiple participants.

Layout Page

The layout of the garden area did not resonate with most participants. They noted that they wanted to decide how big and what kind of shape the garden area would look like. In addition, they wanted to change the size of the grid cells; some would like multiple plants in $1m^2$.

Other Feedback

Another feedback point was that participants would like to have an extra view, the calendar view. In this view, they could see their tasks for a day, week, month or year and have specific tasks color-coded.

7.5 Summary of Results

The box plots for ease of finding tasks, estimation and task completion confidence showed a slight higher score for Group A compared to Group B. However, the statistical analysis, using the Mann-Whitney U test, indicated no significant differences between Group A and Group B. In all three categories, the null hypothesis was retained. Furthermore, the box plot for the recall accuracy scores, revealed that the mean score of Group A was two points higher than Group B. The accuracy scores were tested for normality using the Shapiro-Wilk test, which showed that both groups were normally distributed. Thus a one-tailed independent t-test was performed, which also indicated no significant difference and the null hypothesis was retained. The feedback given by the participants showed issues with distinguishing tasks and plants in the back part of the garden. There were also suggestions on implementing more filtering options and adding filtering options to the Tasks page. Furthermore, the participants wanted more customization, such as adding categories and changing the garden area layout. Lastly, a calendar view was suggested to add on an extra page.

Chapter 8

Discussion

This chapter presents the major findings, strengths and limitations of this project. In addition, recommendations for future prototype development and evaluation methods are proposed.

8.1 Major Findings

This research aimed to design an adaptive task management system to support hobby vegetable gardeners. Understanding the challenges and motivators of gardeners was a crucial step in identifying how to support gardeners the best. In addition, reviewing existing literature and related applications gave insight into key aspects and possible improvements. The platform, TuinMoes, was designed following the Creative Technology design method Mader and Eggink (2014), consisting of four phases: ideation, specification, realisation and evaluation. The TuinMoes platform distinguishes itself by providing multiple viewing options, such as a garden layout, tasks in lists and an overview page combining tasks with a garden layout. These features aimed to allow users to implement different strategies and a quick and easy method to see and add tasks. As stated in the literature review, these features are important for task management.

The platform TuinMoes was evaluated through a user test with 20 participants, divided into Group A, who could see all three pages (Overview, Layout and Tasks) and Group B who could only see the Layout and Tasks pages. The participants entered tasks in lists on the Tasks page and were asked to try to remember their tasks for a specific date in one minute. Afterwards, the participants had to answer Likert scale questions about their perceived confidence of knowing their tasks. Furthermore, they were tested and scored on recalling their tasks. An analysis of the box plots indicated a slight higher score for Group A. However, the results of the statistical tests showed no significant difference in perceived and actual grip between participants who had access to the overview page and those who did not. The thematic feedback analysis suggests that participants tended to be either visually or textually oriented, which makes the platform work for both groups.

Overall, the findings reinforce the need for adaptive task management and the importance of offering diverse strategies and viewing options. These results suggest that the aspects of TuinMoes could address the varying needs of gardeners.

8.2 Strengths and Limitations

It is essential to critically reflect on the research process and the platform itself. In this section, the strengths and limitations of the research methodology and the TuinMoes platform are discussed.

8.2.1 Strengths

First, one of the strengths of this research is the commitment to understanding the needs and challenges experienced by users. Qualitative user interviews were held and coded afterwards. This made it possible to identify both the challenges and motivators of gardeners. In addition, the interviews provided insights into the daily practices of the gardeners and helped define the target group. This detailed understanding was crucial for designing a platform that would help gardeners face their task management challenges.

Secondly, the extensive ideation phase was another strength of the design process. Both divergent and convergent methods were used to explore a broad range of idea directions, which led to new possibilities. After converging to three lo-fi concepts, a Harris Profile with preliminary requirements was created to decide which concept to continue with. This approach strengthened the decision behind the chosen concept because it underwent multiple ideation iterations and was objectively criticized with a Harris profile.

Furthermore, the evaluation was robust, combining a statistical analysis with a thematic analysis of the feedback. This was done by analyzing box plots, testing on normality with the Shapiro-Wilk test and testing on significant differences using the Mann-Whitney U test and the one-tailed independent t-test. This elaborate statistical approach in combination with a thematic analysis of feedback added strength to the evaluation.

Another strength is the viewing options on the TuinMoes platform. This allowed users to decide in which way they wanted to see their tasks and thus allowed users to use diverse strategies.

8.2.2 Limitations

While this research provided insights and showed several strengths, there were multiple limitations in the research design, evaluation and the platform itself that should be considered. First of all, no literature on task management in combination with gardening could be found. The reviewed literature was mostly on Personal Task Management and not specifically for gardening. Task Management methods could be different for gardening compared to general task management.

Furthermore, the participants of the user interviews all live in the same country and motivations and challenges can be different in other locations. The interviews were held with seven participants and therefore may not have been representative of all gardeners.

Moreover, another limitation is the small sample size of the evaluation and the similar backgrounds of the participants. A bigger and more diverse group of participants would make the results more applicable to a wider range of gardeners.

Additionally, the evaluation of the platform lacked real-world context. The TuinMoes platform was not usable for participants' gardens and tasks during the evaluation. This made the testing less realistic.

Lastly, the duration of the evaluation was very limited and lasted only for a short period. Due to the short time frame, there was no clear picture of how the platform would work for gardeners in a real-world setting over a longer period of time.

8.3 Future Recommendations

This section focuses on recommendations for future work and research. The recommendations are divided into two categories: recommendations for further developing the prototype and recommendations for future evaluation methods.

8.3.1 Prototype recommendations

First of all, feedback given during the evaluation showed that the distant part of the garden on the Overview page was difficult to see. To improve the TuinMoes platform, it would be desirable for users to rotate the garden or zoom in on specific parts. Furthermore, it is recommended to make the platform adjustable to user preferences. This could be done by allowing users to add categories and choose which icon they want for what category. In addition, providing adjustable shapes and sizes for the garden layout and implementing more vegetables to choose from would allow users to create layouts that accurately represent their gardens. Having multiple filtering options, such as by date, vegetable type and task type, would further enhance the personalization of the TuinMoes platform. Another aspect of future recommendations is to incorporate and stimulate collaboration. As noted during the user interviews, social connection is a great motivator for gardeners. Stimulating collaboration might motivate gardeners and make the platform feasible for community gardens. This could be done in several ways, for example, by adding an editorial page or getting notifications if other gardeners are in the garden. Moreover, having the option to divide tasks or to claim tasks could be helpful for community gardens. Finally, adding a calendar view would allow gardeners to manage tasks using their preferred strategy, improving the overall user experience.

8.3.2 Evaluation recommendations

In addition to the prototype recommendations, several recommendations for improving the evaluation methods are proposed. First of all, increasing the sample size and including a more diverse group of participants would improve the reliability of the evaluation results. Furthermore, alternative testing methods may be better suited for evaluating specific aspects of the TuinMoes platform. For instance, the recall test used in the evaluation, may not have been the best way to test whether users can view their tasks at a glance on the overview page compared to a task list. While the ability to recall tasks might be related, a more suitable method would be to measure the time it takes participants to verbally list all the tasks to see how well "at a glance" works. Another recommended method to test would be to let participants use the platform in a real-world setting, allowing users to interact with the TuinMoes platform for a longer period. Tracking user behaviour, such as how frequently they interact with the platform and which features they use, could lead to deeper insights into user needs and interaction.

Chapter 9

Conclusion

This thesis aimed to develop an adaptive task management system specifically designed for gardeners, to answer the main research question:

How can an adaptive task management system be designed to support hobby vegetable gardeners?

Through the background research, consisting of a literature review, related work and user interviews, key aspects were identified that should be considered when designing an adaptive task management system for gardeners. One major challenge for gardeners in task management is remembering their tasks, often information is in their heads and not shared with others. In addition, social connections and connection to nature were identified as great motivators. Furthermore, personal task management should be quick and easy to add and view. Strategies differ for each user, task and context, and therefore, task management platforms should be flexible and adaptive to user preferences.

Based on these insights and after several iterations of ideation, the TuinMoes platform was developed. This platform is designed to allow users to see their tasks in different views and to view their tasks at a glance. The platform was evaluated by user-testing with 20 participants, divided into two groups. Group A had access to all three pages of the platform, including the Overview page, while Group B was limited to the Layout page and the Task page. The evaluation assessed if the overview page enhanced the feeling of having a grip on their tasks and seeing the tasks at a glance. The evaluation measured the participants' perceived sense of overview using a Likert scale and the accuracy of knowing tasks was tested by assessing their ability to recall their tasks. Although participants reacted positively to the prototype and box plots showed that Group A scored higher compared to Group B, no significant difference was found in the results between Group A and Group B. This suggests that the overview page did not enhance task management as hypothesised.

Future work could focus on implementing collaborative features into the TuinMoes platform. In addition, it is recommended to improve the evaluation method by testing with a larger and more diverse sample size. Instead of relying on task recall, future evaluations could measure the time it takes to verbally list the tasks for Group A compared to Group B. If the TuinMoes platform is further developed, real-world user testing over a longer period could provide deeper insights into user interaction and the platform's long-term feasibility.

9.1 Contribution to the research field

This research serves as a bridge between the domains of task management and hobby gardening. Although there is existing literature and research about task management and gardening as a hobby, this research explores their intersection by providing insights into how adaptive task management systems can be designed specifically for hobby vegetable gardeners. Furthermore, this report contributes to the field by incorporating qualitative user interviews, which helped to identify motivators and challenges experienced by hobby vegetable gardeners.

9.2 Final Remarks

The platform TuinMoes and the research conducted to develop the prototype was designed with the goal of supporting hobby vegetable gardeners through developing an adaptive task management system. Hoping to minimize the challenge of task organization, the platform aims to support gardens with their tasks while maintaining the joy and relaxation of gardening. Ultimately, gardening as a hobby should be fun and relaxing. The decision of whether and how to manage gardening tasks should remain in the hands of the gardeners themselves, with TuinMoes designed to provide a flexible and supportive tool to empower gardeners.

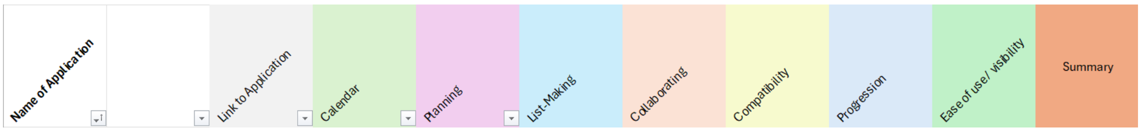
Appendix A

Use of Generative AI

During the preparation of this work the author used Grammaraly in order to help with the grammar and improve the sentence structure. Furthermore, the author used ChatGPT during the preparation of this work in order to help with writing code in HTML, CSS and JavaScript for the final platform. After using these tools, the author reviewed and edited the content as needed and takes full responsibility for the content of the work.

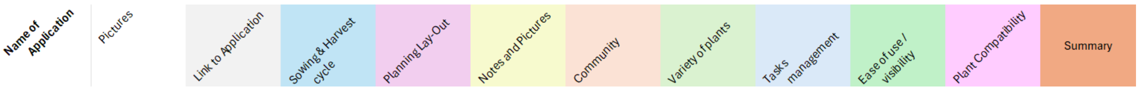
Appendix B

Categories for the Analysis of Task Management Tools



Name of Application		Link to Application	Calendar	Planning	List-Making	Collaborating	Compatibility	Progression	Ease of use/ visibility	Summary

FIGURE B.1: Categories for the Analysis of General Task Management Tools

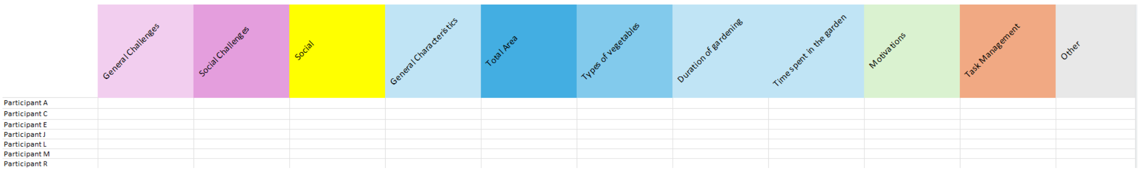


Name of Application	Pictures	Link to Application	Sowing & Harvest cycle	Planning Lay-Out	Notes and Pictures	Community	Variety of plants	Tasks management	Ease of use / visibility	Plant Compatibility	Summary

FIGURE B.2: Categories for the Analysis of Garden Task Management Tools

Appendix C

Table of color-coding categories for the User Interviews

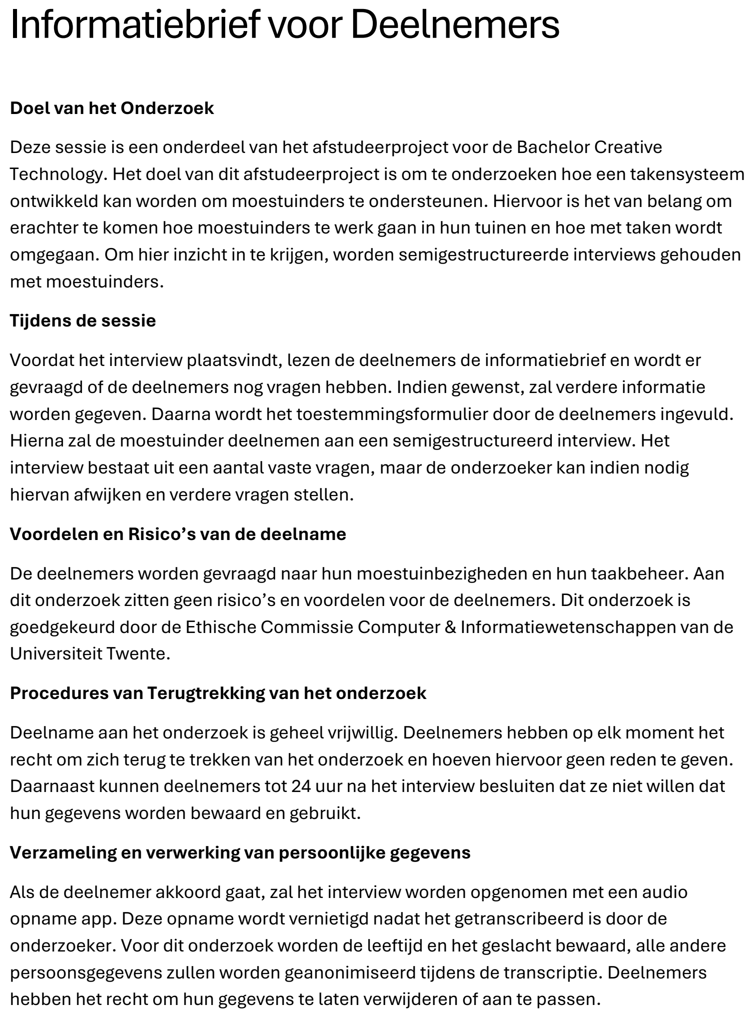


	General Challenges	Social Challenges	Social	General Characteristics	Total Area	Types of vegetables	Duration of gardening	Time spent in the garden	Motivations	Task Management	Other
Participant A											
Participant C											
Participant E											
Participant J											
Participant L											
Participant M											
Participant R											

FIGURE C.1: Color-coded categories for user interviews

Appendix D

Information Letter User Interviews



Informatiebrief voor Deelnemers

Doel van het Onderzoek

Deze sessie is een onderdeel van het afstudeerproject voor de Bachelor Creative Technology. Het doel van dit afstudeerproject is om te onderzoeken hoe een takensysteem ontwikkeld kan worden om moestuinders te ondersteunen. Hiervoor is het van belang om erachter te komen hoe moestuinders te werk gaan in hun tuinen en hoe met taken wordt omgegaan. Om hier inzicht in te krijgen, worden semigestructureerde interviews gehouden met moestuinders.

Tijdens de sessie

Voordat het interview plaatsvindt, lezen de deelnemers de informatiebrief en wordt er gevraagd of de deelnemers nog vragen hebben. Indien gewenst, zal verdere informatie worden gegeven. Daarna wordt het toestemmingsformulier door de deelnemers ingevuld. Hierna zal de moestuinder deelnemen aan een semigestructureerd interview. Het interview bestaat uit een aantal vaste vragen, maar de onderzoeker kan indien nodig hiervan afwijken en verdere vragen stellen.

Voordelen en Risico's van de deelname

De deelnemers worden gevraagd naar hun moestuinbezigheden en hun taakbeheer. Aan dit onderzoek zitten geen risico's en voordelen voor de deelnemers. Dit onderzoek is goedgekeurd door de Ethische Commissie Computer & Informatiewetenschappen van de Universiteit Twente.

Procedures van Terugtrekking van het onderzoek

Deelname aan het onderzoek is geheel vrijwillig. Deelnemers hebben op elk moment het recht om zich terug te trekken van het onderzoek en hoeven hiervoor geen reden te geven. Daarnaast kunnen deelnemers tot 24 uur na het interview besluiten dat ze niet willen dat hun gegevens worden bewaard en gebruikt.

Verzameling en verwerking van persoonlijke gegevens

Als de deelnemer akkoord gaat, zal het interview worden opgenomen met een audio opname app. Deze opname wordt vernietigd nadat het getranscribeerd is door de onderzoeker. Voor dit onderzoek worden de leeftijd en het geslacht bewaard, alle andere persoonsgegevens zullen worden geanonimiseerd tijdens de transcriptie. Deelnemers hebben het recht om hun gegevens te laten verwijderen of aan te passen.

Contactinformatie voor vragen over het onderzoek

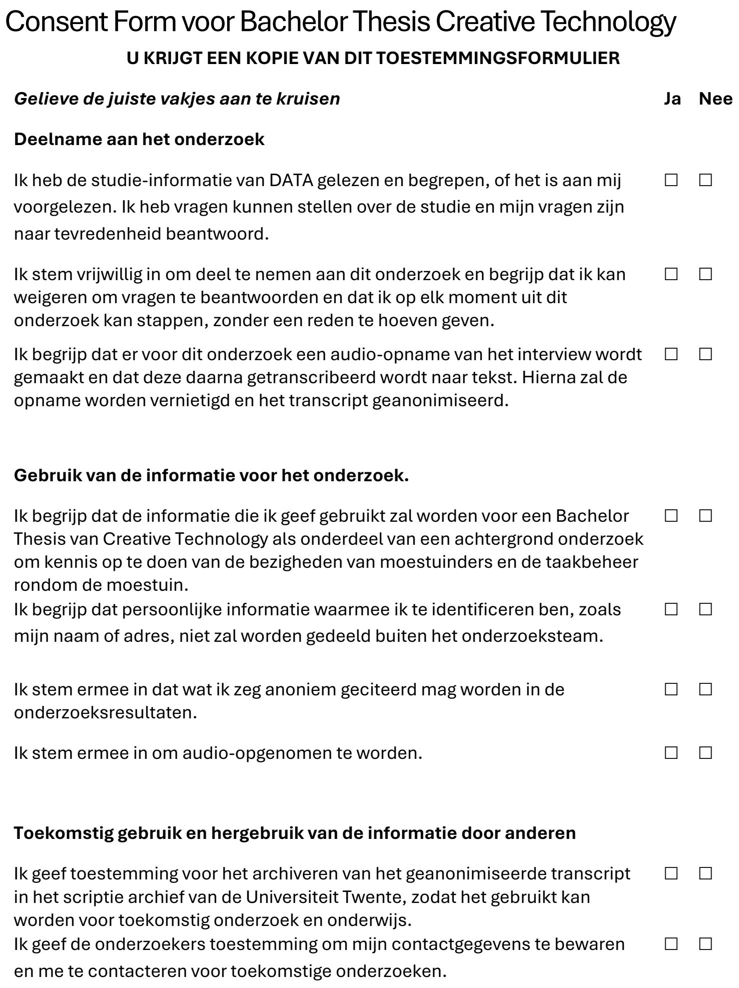
Indien de deelnemers vragen hebben of verdere informatie willen over het onderzoek, kunnen ze Daisy Baars contacteren via e-mail: d.baars@student.utwente.nl. Indien er klachten zijn over dit onderzoek, kunnen deze gestuurd worden naar Dennis Reidsma via e-mail: d.reidsma@utwente.nl

Contactinformatie voor vragen over uw rechten als onderzoekdeelnemer

Indien de deelnemers vragen hebben of informatie willen over het onderzoek, losstaand van de onderzoekers, kunnen ze contact opnemen met de secretaris van de Ethische commissie Computer & Informatiewetenschappen via e-mail: ethicscommittee-CIS@utwente.nl.

Appendix E

Consent Form User Interviews



Consent Form voor Bachelor Thesis Creative Technology

U KRIJGT EEN KOPIE VAN DIT TOESTEMMINGSFORMULIER

Gelieve de juiste vakjes aan te kruisen	**Ja**	**Nee**
Deelname aan het onderzoek		
Ik heb de studie-informatie van DATA gelezen en begrepen, of het is aan mij voorgelezen. Ik heb vragen kunnen stellen over de studie en mijn vragen zijn naar tevredenheid beantwoord.	☐	☐
Ik stem vrijwillig in om deel te nemen aan dit onderzoek en begrijp dat ik kan weigeren om vragen te beantwoorden en dat ik op elk moment uit dit onderzoek kan stappen, zonder een reden te hoeven geven.	☐	☐
Ik begrijp dat er voor dit onderzoek een audio-opname van het interview wordt gemaakt en dat deze daarna getranscribeerd wordt naar tekst. Hierna zal de opname worden vernietigd en het transcript geanonimiseerd.	☐	☐
Gebruik van de informatie voor het onderzoek.		
Ik begrijp dat de informatie die ik geef gebruikt zal worden voor een Bachelor Thesis van Creative Technology als onderdeel van een achtergrond onderzoek om kennis op te doen van de bezigheden van moestuinders en de taakbeheer rondom de moestuin.	☐	☐
Ik begrijp dat persoonlijke informatie waarmee ik te identificeren ben, zoals mijn naam of adres, niet zal worden gedeeld buiten het onderzoeksteam.	☐	☐
Ik stem ermee in dat wat ik zeg anoniem geciteerd mag worden in de onderzoeksresultaten.	☐	☐
Ik stem ermee in om audio-opgenomen te worden.	☐	☐
Toekomstig gebruik en hergebruik van de informatie door anderen		
Ik geef toestemming voor het archiveren van het geanonimiseerde transcript in het scriptie archief van de Universiteit Twente, zodat het gebruikt kan worden voor toekomstig onderzoek en onderwijs.	☐	☐
Ik geef de onderzoekers toestemming om mijn contactgegevens te bewaren en me te contacteren voor toekomstige onderzoeken.	☐	☐

Handtekeningen

Naam deelnemer Handtekening Datum

Ik heb deelnemers nauwkeuring de informatiebrief zien lezen of ik heb de brief nauwkeurig voorgelezen en de deelnemer de kans gegeven vragen te stellen. Ik heb ervoor gezorgd dat de deelnemer begrijpt waarvoor hij/zij vrijwillig toestemming heeft gegeven.

Naam onderzoeker Handtekening Datum

Contact gegevens onderzoeker voor verdere informatie

Voor vragen of verdere informatie kunt u de onderzoeker Daisy Baars contacteren via e-mail: d.baars@student.utwente.nl.

Contact gegevens voor vragen over uw rechten als deelnemer

Als uw vragen heeft over uw rechten als deelnemer, meer informatie wilt of vragen wilt stellen losstaand van de onderzoeker, kunt u de secretaris van de Ethiek Commissie van Computer & Informatiewetenschappen contacteren via e-mail: ethicscommittee-CIS@utwente.nl .

Appendix F

HTML,CSS and JavaScript Code

F.1 HTML code of the overview page

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="./style.css">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link rel="stylesheet"
        href="https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.css">
    <link
        href="https://fonts.googleapis.com/css2?family=Lustria&family=Nunito:ital,wght@0,200
        rel="stylesheet">
    <title>Overzicht</title>
</head>
<body id="index-page" class="index-page">
    <nav>
        <div class="container">
            <a href="index.html" class="logo"><img src="./Images/logo.png"
                id="imgagelogo" alt="Logo"></a>
            <h1>Overzicht</h1>
            <ul>
                <li><a href="index.html" class="active">Overzicht</a></li>
                <li><a href="/birdview/birdview.html">Indeling</a></li>
                <li><a href="/tasklist/taskslists.html">Taken</a></li>
            </ul>
        </div>
    </nav>

    <div class="birdview-layout">
        <!-- Ground Grid for Garden Layout on Overview Page -->
        <div class="ground">
            <div class="grid-container"></div>
        </div>


        <!-- Date Selection Tool -->
        <div class="timeform">
            <div class="form-container">
                <label for="date-range">Selecteer een tijdsperiode:</label>
                <input type="text" id="date-range" placeholder="Kies een
                    datum">
            </div>

        </div>

        <div id="taskPopup" class="popup">
            <div class="popup-content">
                <span id="closePopup" class="popup-close">&times;</span>
                <h2>Taak Details</h2>
                <p><strong>Plant:</strong> <span
                    id="popupVegetable"></span></p>
                <p><strong>Taak:</strong> <span
                    id="popupCategory"></span></p>
                <p><strong>Datum:</strong> <span id="popupDate"></span></p>
                <p><strong>Opmerking:</strong> <span
                    id="popupText"></span></p>
            </div>
        </div>




    <!-- Script to Import Grid and Task List Functionality -->
    <script src="https://cdn.jsdelivr.net/npm/flatpickr"></script>
    <script type="module" src="./index.js"></script>
</body>
</html>
```

F.2 JavaScript code of the overview page

```
    import { createGrid, loadVegetables, saveGridState } from
        ’/birdview/birdview.js’;

document.addEventListener(’DOMContentLoaded’, function () {
    createGrid(’.grid-container’);
    loadVegetables(’.grid-container’);

    // Initialize date picker
    const today = new Date();
    const formattedDate = today.toLocaleDateString(’nl-NL’); // UK format
        (dd/mm/yyyy)

    flatpickr("#date-range", {
        mode: "range",
        dateFormat: "d-m-Y",
        onClose: function (selectedDates) {
            let startDate, endDate;
            if (selectedDates.length === 1) {
                startDate = endDate = selectedDates[0];
            } else if (selectedDates.length === 2) {
                [startDate, endDate] = selectedDates;
            }
            updateTaskIcons({ startDate, endDate });
        }
    });

    // Add task icons and descriptions below the ground
    addTaskIconsDescription();

    // Close the popup when clicked outside of it
    window.addEventListener("click", function (event) {
        const popup = document.getElementById("taskPopup");
        if (event.target === popup) {
            popup.style.display = "none";
        }
    });

    // Close the popup when the close button is clicked
    const closeButton = document.getElementById("closePopup");
    if (closeButton) {
        closeButton.addEventListener(’click’, function () {
            document.getElementById("taskPopup").style.display = "none";
        });
    }
});

// Function to update task icons
```

```
function updateTaskIcons(dateRange) {
    console.log("Updating task icons with range:", dateRange);
    document.querySelectorAll('.oogsten-icon, .zaaien-icon, .wieden-icon,
        .overig-icon, .overlap-icon').forEach(icon => icon.remove());
    let overlapTracker = {}; // To track overlaps
    addTaskIcons('.grid-container', dateRange, 'oogsten', overlapTracker);
    addTaskIcons('.grid-container', dateRange, 'zaaien', overlapTracker);
    addTaskIcons('.grid-container', dateRange, 'wieden', overlapTracker);
    addTaskIcons('.grid-container', dateRange, 'overig', overlapTracker);
    handleOverlaps(overlapTracker);
}

// Add task icons and descriptions below the ground
function addTaskIconsDescription() {
       \begin{verbatim}
    const iconData = [
        { icon: '  ', description: 'Oogsten' },
        { icon: '     ', description: 'Zaaien' },
        { icon: '  ', description: 'Wieden' },
        { icon: '  ', description: 'Overig' },
        { icon: '  ', description: 'Meerdere taken' }
    ];
    \end{verbatim}

    const iconsGuide = document.createElement('div');
    iconsGuide.classList.add('icons-guide');

    iconData.forEach(item => {
        const iconItem = document.createElement('div');
        iconItem.classList.add('icon-item');

        // Create and append the icon
        const iconSpan = document.createElement('span');
        iconSpan.classList.add('icon');
        iconSpan.textContent = item.icon;

        // Create and append the description text
        const descriptionText = document.createElement('p');
        descriptionText.textContent = item.description;

        iconItem.appendChild(iconSpan);
        iconItem.appendChild(descriptionText);

        iconsGuide.appendChild(iconItem);
    });

    // Append the icons guide to the element directly below the ground
    const ground = document.querySelector('.ground');
    ground.parentNode.appendChild(iconsGuide); // Place the icons below
        the ground container
}

// Function to add task icons to the grid
function addTaskIcons(gridContainerSelector, dateRange, taskCategory,
    overlapTracker) {
    const gridContainer = document.querySelector(gridContainerSelector);
    const tasks = JSON.parse(localStorage.getItem(`${taskCategory}Tasks`))
        || [];
    const gridState = JSON.parse(localStorage.getItem('gridState')) || [];
    const { startDate, endDate } = dateRange;

    tasks.forEach(({ vegetable, date, task }) => {
        let taskDates = date;

        if (taskDates.length > 1) {
            const [taskStartDate, taskEndDate] =
                taskDates.map(normalizeDate);
            const normalizedStartDate = normalizeDate(startDate);
            const normalizedEndDate = normalizeDate(endDate);

            if (taskStartDate <= normalizedEndDate && taskEndDate >=
                normalizedStartDate) {
                addIconToGrid(taskCategory, vegetable, gridContainer,
                    gridState, dateRange, task);
                overlapTracker[vegetable] = overlapTracker[vegetable] || [];
                overlapTracker[vegetable].push({ category: taskCategory,
                    task, dates: taskDates });
            }
        } else {
            const taskDate = normalizeDate(taskDates[0]);
            const normalizedStartDate = normalizeDate(startDate);
            const normalizedEndDate = normalizeDate(endDate);

            if (taskDate >= normalizedStartDate && taskDate <=
                normalizedEndDate) {
                addIconToGrid(taskCategory, vegetable, gridContainer,
                    gridState, dateRange, task);
                overlapTracker[vegetable] = overlapTracker[vegetable] || [];
                overlapTracker[vegetable].push({ category: taskCategory,
                    task, dates: taskDates });
            }
        }
    });
}

function addIconToGrid(taskCategory, vegetable, gridContainer, gridState,
    dateRange, task) {
```

```
    gridState.forEach(({ index, src }) => {
        const normalizedVegetable =
            src.split(’/’).pop().split(’.’)[0].toLowerCase().trim();
        const taskVegetable = vegetable.toLowerCase().trim();

        if (normalizedVegetable === taskVegetable) {
            const gridItem = gridContainer.children[index];
            if (gridItem &&
                !gridItem.querySelector(‘.${taskCategory}-icon‘)) {
                const icon = document.createElement(’span’);
                icon.className = ‘${taskCategory}-icon‘;
                icon.textContent = taskCategory === ’oogsten’ ? ’ ’ :
                                 taskCategory === ’zaaien’ ?
                                     String.fromCodePoint(0x1F331) :
                                 taskCategory === ’wieden’ ? ’’ :
                                 ’   ’;

                icon.addEventListener(’click’, function (event) {
                    event.stopPropagation();
                    showPopup(vegetable, taskCategory, dateRange.startDate,
                        dateRange.endDate, task);
                });

                gridItem.appendChild(icon);
            }
        }
    });
}

// Normalize date
function normalizeDate(date) {
    const localDate = new Date(date);
    if (isNaN(localDate.getTime())) {
        console.error("Invalid date:", date);
        return "Invalid Date";
    }
    return ‘${localDate.getFullYear()}-${String(localDate.getMonth() +
        1).padStart(2, ’0’)}-${String(localDate.getDate()).padStart(2,
        ’0’)}‘;
}

// Show popup
function showPopup(vegetable, category, startDate, endDate, task) {
    document.getElementById("popupVegetable").textContent = vegetable;
    document.getElementById("popupCategory").textContent =
        capitalizeFirstLetter(category);
    document.getElementById("popupDate").textContent =
        ‘${startDate.toLocaleDateString("nl-NL")} tot
        ${endDate.toLocaleDateString("nl-NL")}‘;
    document.getElementById("popupText").textContent = task || "Geen
        opmerking";
    document.getElementById("taskPopup").style.display = "block";
}

// Capitalize first letter
function capitalizeFirstLetter(string) {
    return string.charAt(0).toUpperCase() + string.slice(1).toLowerCase();
}

// Handle overlaps
function handleOverlaps(overlapTracker) {
    for (const vegetable in overlapTracker) {
        if (overlapTracker[vegetable].length > 1) {
            const gridContainer = document.querySelector(’.grid-container’);
            const gridState = JSON.parse(localStorage.getItem(’gridState’))
                || [];

            gridState.forEach(({ index, src }) => {
                const normalizedVegetable =
                    src.split(’/’).pop().split(’.’)[0].toLowerCase().trim();
                const taskVegetable = vegetable.toLowerCase().trim();

                if (normalizedVegetable === taskVegetable) {
                    const gridItem = gridContainer.children[index];
                    if (gridItem &&
                        !gridItem.querySelector(’.overlap-icon’)) {
                        const icon = document.createElement(’span’);
                        icon.className = ’overlap-icon’;
                        icon.textContent = ’ ’;
                        gridItem.appendChild(icon);

                        icon.addEventListener(’click’, function () {
                            const overlapDetails =
                                overlapTracker[vegetable].map(({ category,
                                dates, task }) => ({
                                category: capitalizeFirstLetter(category),
                                dates: dates.length > 1
                                    ? ‘${new
                                        Date(dates[0]).toLocaleDateString("nl-NL")}
                                        tot ${new
                                        Date(dates[1]).toLocaleDateString("nl-NL")}‘
                                    : new
                                        Date(dates[0]).toLocaleDateString("nl-NL"),
                                task: task || "Geen opmerking"
                            }));

                            const popup =
                                document.getElementById("taskPopup");
                            document.getElementById("popupVegetable").textContent
                                = vegetable;
                            document.getElementById("popupCategory").textContent
                                = overlapDetails.map(d => d.category).join(",
                                ");
                            document.getElementById("popupDate").textContent
                                = overlapDetails.map(d => d.dates).join("; ");
                            document.getElementById("popupText").textContent
                                = overlapDetails.map(d => d.task).join("; ");
                            popup.style.display = "block";
                        });
                    }
                }
            });
        }
    }
}
```

F.3 HTML code of Lay-out page

```
    <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Indeling</title>
    <link rel="stylesheet" href="../style.css">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link
        href="https://fonts.googleapis.com/css2?family=Lustria&family=Nunito:ital,wght@0,200
        rel="stylesheet">
    <link rel="stylesheet"
        href="https://unicons.iconscout.com/release/v4.0.8/css/line.css">
</head>
<body>
    <nav>
        <div class="container">
            <a href="./../index.html" class="logo"><img
                src="../Images/logo.png"></a>
            <h1>Indeling</h1>
            <ul>
                <li><a href="./../index.html">Overzicht</a></li>
                <li><a
                    href="/birdview/birdview.html"class="active">Indeling</a></li>
                <li><a href="/tasklist/taskslists.html">Taken</a></li>
            </ul>
        </div>
    </nav>

    <div class="birdview-layout">


        <!-- Rulers -->
        <div class="ruler-wrapper">
            <div class="ruler horizontal top"></div>
            <div class="ruler vertical left"></div>
        </div>

        <!-- Ground with Grid -->
        <div class="ground">
            <div class="grid-container"></div>
        </div>

        <!-- Vegetable Selection Tool -->
        <div class="vegetableform">
            <div class="form-container">
            <label for="vegetable">Selecteer een plant:</label>
            <select id="vegetable">
                <option value="" disabled selected>Selecteer</option>
                <option value="tomaat">Tomaat</option>
                <option value="wortel">Wortel</option>
                <option value="ui">Ui</option>
                <option value="aardbei">Aardbei</option>
                <option value="sla">Sla</option>
                <option value="courgette">Courgette</option>
                <option value="aubergine">Aubergine</option>
                <option value="bonen">Bonen</option>

            </select>
        </div>
        <img id="vegetableImage" alt="Drag a vegetable to the grid"
            style="display: none;">
    </div>

    <div id="vuilnisBak" class="vuilnisbak">
        <img src="../Images/vuilnisbak.png" alt="Vuilnisbak"
            id="vuilnisBak">
    </div>
</div>

<script type="module">
    import { createGrid, loadVegetables, generateRulers,
        setupVegetableSelector } from ’./birdview.js’;

    // Initialize the grid
    createGrid(’.grid-container’);

    // Generate rulers
    generateRulers(’.ruler.horizontal.top’, ’.ruler.vertical.left’);

    // Load saved vegetables into the grid
    loadVegetables(’.grid-container’);

    // Set up vegetable selector
    setupVegetableSelector(
        ’vegetable’, // ID of the vegetable selector dropdown
        ’vegetableImage’, // ID of the image preview element
        {
            tomaat: ’../Images/tomaat.png’,
            wortel: ’../Images/wortel.png’,
            ui: ’../Images/ui.png’,
            aardbei: ’../Images/aardbei.png’,
            sla: ’../Images/sla.png’,
            courgette: ’../Images/courgette.png’,
            aubergine: ’../Images/aubergine.png’,
                bonen: ’../Images/bonen.png’
            }
        );
    </script>
</body>
</html>
```

F.4 JavaScript code of Lay-out page

```
    // Create the grid dynamically
export function createGrid(gridContainerSelector, gridSize = { rows: 12,
    cols: 20 }) {
    const gridContainer = document.querySelector(gridContainerSelector);
    gridContainer.innerHTML = ''; // Clear previous grid if any

    for (let i = 0; i < gridSize.rows * gridSize.cols; i++) {
        const gridItem = document.createElement('div');
        gridItem.className = "grid-item";
        gridItem.dataset.index = i;
        gridItem.addEventListener("dragover", allowDrop);
        gridItem.addEventListener("drop", drop);
        gridContainer.appendChild(gridItem);
    }
}

// Load vegetables from localStorage and add them to the grid
export function loadVegetables(gridContainerSelector) {
    const gridState = JSON.parse(localStorage.getItem('gridState')) || [];
    const gridContainer = document.querySelector(gridContainerSelector);

    gridState.forEach(({ index, src, id }) => {
        const gridItem = gridContainer.children[index];
        const img = createVegetableImage(src);
        img.id = id; // Assign the saved unique ID
        gridItem.appendChild(img);
    });

    // Only set up trash bin if it's present on the page
    setupTrashBin();
}

// Create a draggable vegetable image with a unique ID
function createVegetableImage(src) {
    const img = document.createElement('img');
    img.src = src;

    if (img.id == null){  // only create a new vegetable if there isn't
        already one
        const uniqueId = `vegetable-${Date.now()}`; // Use timestamp as a
            unique ID
        img.id = uniqueId; // Assign a unique ID
        img.className = "vegetable-image";
        img.draggable = true;
    }

    img.addEventListener("dragstart", dragStart);
    return img;
}

// Drag-and-drop logic for deleting vegetables from the grid
function drop(event) {
    event.preventDefault();

    const draggedId = event.dataTransfer.getData("text/plain");
    const draggedElement = document.getElementById(draggedId);

    if (event.target.classList.contains("grid-item") &&
        event.target.children.length === 0) {
        const clonedElement = draggedElement.cloneNode(true);
        clonedElement.id = ‘vegetable-${Date.now()}‘; // Ensure unique ID
            for the clone
        clonedElement.addEventListener("dragstart", dragStart); //
            Re-attach dragstart event
        event.target.appendChild(clonedElement);
        saveGridState(’.grid-container’);
    }

    // Handle drop on trash bin
    if (event.target.id === "vuilnisBak") {
        // Remove the vegetable from the DOM
        if (draggedElement) {
            draggedElement.parentElement.removeChild(draggedElement);

            // Remove the vegetable from localStorage
            const gridState = JSON.parse(localStorage.getItem(’gridState’))
                || [];
            const updatedGridState = gridState.filter(item => item.id !==
                draggedElement.id); // Remove by unique ID

            // Update localStorage with the new grid state
            localStorage.setItem(’gridState’,
                JSON.stringify(updatedGridState));

            saveGridState(’.grid-container’); // Update the grid state in
                localStorage
        }
    }
}

// Save the grid state to localStorage with the unique vegetable IDs
export function saveGridState(gridContainerSelector) {
    const gridState =
        [...document.querySelectorAll(‘${gridContainerSelector}
        .grid-item‘)]
        .map((item, index) => {
            const img = item.querySelector("img");
            return img ? { index, src: img.src, id: img.id } : null;
        })
        .filter(Boolean); // Remove nulls from the array
    localStorage.setItem(’gridState’, JSON.stringify(gridState)); // Store the grid state in localStorage
}

// Setup trash bin to delete vegetable images (only if the trash bin exists)
function setupTrashBin() {
    const vuilnisBak = document.getElementById(’vuilnisBak’);
    if (vuilnisBak) {
        vuilnisBak.addEventListener(’dragover’, allowDrop);
        vuilnisBak.addEventListener(’drop’, drop);
    } else {
        console.log("Trash bin element not found.");
    }
}

// Drag-and-drop helpers
function dragStart(event) {
    event.dataTransfer.setData("text/plain", event.target.id);
}

function allowDrop(event) {
    event.preventDefault();
}

// Helper function to generate rulers for the grid (optional)
export function generateRulers() {
    // Target the horizontal and vertical rulers using the correct classes
    const horizontalRuler = document.querySelector(’.ruler.horizontal’);
    const verticalRuler = document.querySelector(’.ruler.vertical’);

    // Check if the rulers exist before attempting to generate ticks
    if (horizontalRuler) {
        generateTicks(horizontalRuler, 20, 21, true); // Assuming widthMeters is 20
    } else {
        console.log(’Horizontal ruler not found’);
    }

    if (verticalRuler) {
        generateTicks(verticalRuler, 12, 13, false); // Assuming heightMeters is 12
    } else {
        console.log(’Vertical ruler not found’);
    }
}

// Helper function to generate ruler ticks (optional)
function generateTicks(container, length, numTicks, isHorizontal = true) {
    if (!container) return; // Prevent errors if container is null

    container.innerHTML = ''; // Clear existing ticks

    const interval = length / (numTicks - 1);

    for (let i = 0; i < numTicks; i++) {
        const tick = document.createElement('span');
        tick.textContent = `${(i * interval).toFixed(0).padStart(2, '0')}m`;
        tick.style.position = "absolute";
        if (isHorizontal) {
            tick.style.left = `${(i / (numTicks - 1)) * 100}%`;
        } else {
            tick.style.top = `${(i / (numTicks - 1)) * 100}%`;
        }
        container.appendChild(tick);
    }
}

// Vegetable Selection and Image Display (optional)
export function setupVegetableSelector(selectorId, imageId,
    vegetableImages) {
    const vegetableSelect = document.getElementById(selectorId);
    const vegetableImage = document.getElementById(imageId);

    if (!vegetableSelect || !vegetableImage) {
        return; // If the vegetable selector or image preview isn't found,
            skip the initialization
    }

    vegetableSelect.addEventListener("change", () => {
        const selectedVegetable = vegetableSelect.value;
        const imageUrl = vegetableImages[selectedVegetable];

        vegetableImage.src = imageUrl || "";
        vegetableImage.alt = selectedVegetable || "Select a vegetable to
            see the image";
        vegetableImage.style.display = imageUrl ? "block" : "none";
        vegetableImage.draggable = !!imageUrl;

        if (imageUrl) {
            vegetableImage.id = `vegetable-${selectedVegetable}`;
            vegetableImage.addEventListener("dragstart", dragStart);
        }
    });
}

// Example vegetable images for the selector
const vegetableImages = {
    tomaat: '../Images/tomaat.png',
    wortel: '../Images/wortel.png',
    ui: '../Images/ui.png',
    aardbei: '../Images/aardbei.png',
    sla: '../Images/sla.png',
    courgette: '../Images/courgette.png',
    aubergine: '../Images/aubergine.png',
    bonen: '../Images/bonen.png'
};

// Initialize vegetable selection functionality
export function initializeVegetableSelection() {
    setupVegetableSelector(
        'vegetable', // ID of the vegetable selector dropdown
        'vegetableImage', // ID of the image preview element
        vegetableImages
    );
}

// Call the initialization function once the DOM is ready
document.addEventListener('DOMContentLoaded', function () {
    // Check if we're on the birdview page
    if (window.location.pathname.includes("birdview")) {
        createGrid('.grid-container');
        loadVegetables('.grid-container');
        setupTrashBin();
        generateRulers();
        initializeVegetableSelection();
    }
});
```

F.5 HTML code of Tasks page

```
    <html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Taken</title>
    <link rel="stylesheet" href="./../style.css">
    <link rel="stylesheet"
        href="https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.css">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link
        href="https://fonts.googleapis.com/css2?family=Lustria&family=Nunito:ital,wght@0,200
        rel="stylesheet">
    <link rel="stylesheet"
        href="https://unicons.iconscout.com/release/v4.0.8/css/line.css">

</head>
<body>
    <nav>
        <div class="container">
            <a href="./../index.html" class="logo"><img
                src="./../Images/logo.png"></a>
            <h1>Taken</h1>
            <ul>
                <li><a href="./../index.html" >Overzicht</a></li>
                <li><a href="/birdview/birdview.html">Indeling</a></li>
                <li><a
                    href="/tasklist/taskslists.html"class="active">Taken</a></li>
            </ul>




            <button id="menubtn"><i class="uis uis-bars"></i></button>
            <button id="closebtn"><i class="uis uis-multiply"></i></button>
        </div>
    </nav>


    <div class="body">
        <div class="task-box" id="task-input">
            <h3>Taak toevoegen</h3>

            <div class="task-select">
                <select id="task-vegetable-selector">
                    <option value="" disabled selected>Selecteer een
                        plant</option>
                    <option value="tomaat">Tomaat</option>
                    <option value="wortel">Wortel</option>
                    <option value="ui">Ui</option>
                    <option value="aardbei">Aardbei</option>
                    <option value="sla">Sla</option>
                    <option value="courgette">Courgette</option>
                    <option value="aubergine">Aubergine</option>
                    <option value="bonen">Bonen</option>
                </select>
                <select id="task-category-selector">
                    <option value="" disabled selected>Selecteer een
                        categorie</option>
                    <option value="oogsten">Oogsten</option>
                    <option value="wieden">Wieden</option>
                    <option value="zaaien">Zaaien</option>
                    <option value="overig">Overig</option>
                </select>

                <input type="date" id="task-date" placeholder="Kies een
                    datum">
                <input type="text" id="task-input-text"
                    placeholder="Opmerking toevoegen">
            </div>
            <!-- <div class="task-action-container"> -->
                <button id="add-btn" class="btn btn-primary"
                    onclick="addTask()">Voeg toe</button>

                <!-- <div class="timefilter">
                    <label for="daterange">Filter een tijdsperiode:</label>
                    <input type="text" id="daterange" placeholder="Kies een
                        datum">
                </div> -->
            <!-- </div>
             -->
        </div>
        <div class="task-grid">


            <div class="task-box" id="oogsten">
                <h3>Oogsten</h3>
                <ul id="oogsten"></ul>
                <ul id="oogsten-tasks"></ul>
            </div>
            <div class="task-box" id="wieden">
                <h3>Wieden</h3>
                <ul id="wieden"></ul>
                <ul id="wieden-tasks"></ul>
            </div>
            <div class="task-box" id="zaaien">
```

```
            <h3>Zaaien</h3>
            <ul id="zaaien"></ul>
            <ul id="zaaien-tasks"></ul>
        </div>
        <div class="task-box" id="overig">
            <h3>Overig</h3>
            <ul id="overig"></ul>
            <ul id="overig-tasks"></ul>
        </div>




    </div>




</div>
<script src="https://cdn.jsdelivr.net/npm/flatpickr"></script>
<script src="/tasklist/tasklist.js"></script>
</body>
</html>
```

F.6 JavaScript code of Tasks page

```
    document.addEventListener('DOMContentLoaded', function () {
    // Initialize Flatpickr on the date inputs
    flatpickr("#task-date", {
        dateFormat: "d-m-Y", // UK format
        mode: "range", // Set mode to 'range' to support date ranges
        enableTime: false, // Disable time picker
    });

    flatpickr("#daterange", {
        mode: "range",
        dateFormat: "d-m-Y",
        onClose: function (selectedDates) {
            let startDate, endDate;
            if (selectedDates.length === 1) {
                startDate = endDate = selectedDates[0]; // Match single day
            } else if (selectedDates.length === 2) {
                [startDate, endDate] = selectedDates;
            }
            updateTaskIcons({ startDate, endDate });
        }
    });
});


// Function to add or edit a task in a specific category
function addTask() {
    const taskInput = document.getElementById("task-input-text");
    const categorySelector =
        document.getElementById("task-category-selector");
    const vegetableSelector =
        document.getElementById("task-vegetable-selector");
    const dateInput = document.getElementById("task-date");

    const selectedCategory = categorySelector.value;
    const selectedVegetable = vegetableSelector.value;
    const taskDate = dateInput.value ? dateInput.value.split(' to ') : [];

    // Capitalize the first letter of the vegetable name
    const capitalizedVegetable = selectedVegetable ?
        selectedVegetable.charAt(0).toUpperCase() +
        selectedVegetable.slice(1).toLowerCase() : "";

    if (taskInput.value.trim() !== "" || selectedCategory ||
        capitalizedVegetable || dateInput.value) {
        const taskItem = document.createElement('li');

        let taskDetails = taskInput.value ? `${taskInput.value}` : '';
        if (capitalizedVegetable) taskDetails += ‘ ${capitalizedVegetable}‘;
        if (selectedCategory) taskDetails += ‘ ${selectedCategory}‘;
        if (taskDate.length) taskDetails += ‘: ${taskDate.join(’ tot ’)}‘;

        taskItem.textContent = taskDetails;

        taskItem.originalTask = {
            task: taskInput.value,
            vegetable: capitalizedVegetable,
            category: selectedCategory,
            date: taskDate.map(d => normalizeDate(d))
        };

        const tasks =
            JSON.parse(localStorage.getItem(‘${selectedCategory}Tasks‘)) ||
            [];
        tasks.push(taskItem.originalTask);

        // Sort tasks before saving them to localStorage
        tasks.sort((a, b) => {
            const dateA = new Date(a.date[0]); // First date in the range
            const dateB = new Date(b.date[0]); // First date in the range
            return dateA - dateB; // Sort ascending
        });

        localStorage.setItem(‘${selectedCategory}Tasks‘,
            JSON.stringify(tasks));

        // Clear and reload the tasks to reflect the updated order
        const taskList =
            document.getElementById(‘${selectedCategory}-tasks‘);
        taskList.innerHTML = ""; // Clear current tasks
        loadTasks(selectedCategory);

        taskInput.value = "";
        categorySelector.value = "";
        vegetableSelector.value = "";
        dateInput.value = "";
    } else {
        alert("Please enter at least one detail (task, category, vegetable,
            or date).");
    }
}


// Helper function to normalize date to local date (dd-mm-yyyy format)
function normalizeDate(date) {
    const parts = date.split(’-’);
    if (parts.length !== 3) {
        console.error("Invalid date format:", date);
        return "bla";
    }

    const day = parseInt(parts[0], 10);
    const month = parseInt(parts[1], 10) - 1; // Month is 0-indexed
    const year = parseInt(parts[2], 10);

    const normalizedDate = new Date(Date.UTC(year, month, day));
    if (isNaN(normalizedDate.getTime())) {
        console.error("Invalid date:", date);
        return "bloob";
    }

    const normalizedDay = String(normalizedDate.getUTCDate()).padStart(2,
        ’0’);
    const normalizedMonth = String(normalizedDate.getUTCMonth() +
        1).padStart(2, ’0’);
    const normalizedYear = normalizedDate.getUTCFullYear();

    return ‘${normalizedYear}-${normalizedMonth}-${normalizedDay}‘;
}

// Function to load tasks from localStorage and display them
function loadTasks(category) {
    const tasks = JSON.parse(localStorage.getItem(‘${category}Tasks‘)) ||
        [];

    tasks.sort((a, b) => {
        const dateA = new Date(a.date[0]); // First date in the range
        const dateB = new Date(b.date[0]); // First date in the range
        return dateA - dateB; // Sort ascending
    });

    tasks.forEach(({ task, vegetable, category: taskCategory, date }) => {
        const taskItem = document.createElement(’li’);
        date = date.map(date => new Date(date).toLocaleDateString("nl-NL"))
        const taskDate = Array.isArray(date) ? date.join(’ tot ’) : date;

        // Capitalize the first letter of the vegetable name
        const capitalizedVegetable = vegetable ?
            vegetable.charAt(0).toUpperCase() +
            vegetable.slice(1).toLowerCase() : "";

        let taskDetails = task ? ‘${task} -‘ : ’’;
        if (capitalizedVegetable) taskDetails += ‘ ${capitalizedVegetable}‘;
        if (taskCategory) taskDetails += ‘ ${taskCategory}‘;
        if (taskDate) taskDetails += ‘: ${taskDate}‘;
        taskItem.textContent = taskDetails;

        // Store the original task data for later reference (for deletion)
        taskItem.originalTask = { task, vegetable, category: taskCategory,
            date };

        // Create Delete button
        const deleteButton = document.createElement(’button’);
        deleteButton.textContent = ’Verwijder’;
        deleteButton.classList.add(’btn-small’);
        deleteButton.onclick = () => deleteTask(taskItem, category);

        taskItem.appendChild(deleteButton);

        // Append the task item to the task list
        document.getElementById(‘${category}-tasks‘).appendChild(taskItem);
    });
}

// Function to delete a task
function deleteTask(taskItem, category) {
    // Remove the task from the list
    taskItem.remove();

    // Remove task from localStorage
    const tasks = JSON.parse(localStorage.getItem(‘${category}Tasks‘)) ||
        [];
    const updatedTasks = tasks.filter(t => t.task !==
        taskItem.originalTask.task);
    localStorage.setItem(‘${category}Tasks‘, JSON.stringify(updatedTasks));
}

function filterTasksByDateRange(startDate, endDate) {
    // Normalize start and end dates to ensure proper date format
        comparison
    const startTimestamp = startDate ? new Date(startDate +
        "T00:00:00").getTime() : null;
    const endTimestamp = endDate ? new Date(endDate +
        "T23:59:59").getTime() : null;

    // Iterate through categories to filter tasks
    [’oogsten’, ’wieden’, ’zaaien’, ’overig’].forEach(category => {
        const allTasks =
            JSON.parse(localStorage.getItem(‘${category}Tasks‘)) || [];
        const filteredTasks = allTasks.filter(task => {
            const taskStartTimestamp = new Date(task.date[0] +
                "T00:00:00").getTime();
            const taskEndTimestamp = task.date[1] ? new Date(task.date[1] +
                "T23:59:59").getTime() : taskStartTimestamp;

            // Check if task date range overlaps with selected date range
            if (!endTimestamp && startTimestamp) {
                return taskStartTimestamp === startTimestamp;
            }

            return (
                (!startTimestamp || taskEndTimestamp >= startTimestamp) &&
                (!endTimestamp || taskStartTimestamp <= endTimestamp)
            );
        });

        // Update UI with filtered tasks
        const taskList = document.getElementById(‘${category}-tasks‘);
        taskList.innerHTML = ""; // Clear current tasks
        filteredTasks.forEach(task => {
            const taskItem = document.createElement(’li’);
            const taskDate = task.date.map(d => new
                Date(d).toLocaleDateString("nl-NL")).join(’ tot ’);
            let taskDetails = task.task ? ‘${task.task} -‘ : ’’;
            if (task.vegetable) taskDetails += ‘ ${task.vegetable}‘;
            if (task.category) taskDetails += ‘ ${task.category}‘;
            if (taskDate) taskDetails += ‘: ${taskDate}‘;
            taskItem.textContent = taskDetails;

            // Append the task to the list
            taskList.appendChild(taskItem);
        });
    });
}

function updateTaskIcons({ startDate, endDate }) {
    // Normalize the dates for filtering to local timezone (to avoid UTC
        issue)
    const normalizedStartDate = startDate ? formatLocalDate(startDate) :
        null;
    const normalizedEndDate = endDate ? formatLocalDate(endDate) : null;

    console.log("Selected Dates:", normalizedStartDate,
        normalizedEndDate); // Debugging log

    // Filter and update the tasks
    filterTasksByDateRange(normalizedStartDate, normalizedEndDate);
}

// Helper function to format the date to local timezone (yyyy-mm-dd)
function formatLocalDate(date) {
    const localDate = new Date(date);

    const year = localDate.getFullYear();
    const month = String(localDate.getMonth() + 1).padStart(2, ’0’); //
       Months are 0-indexed
    const day = String(localDate.getDate()).padStart(2, ’0’); // Ensure
       day is 2 digits

    return ‘${year}-${month}-${day}‘;
}

// Load tasks for all categories on page load
window.onload = function () {
    loadTasks(’oogsten’);
    loadTasks(’wieden’);
    loadTasks(’zaaien’);
    loadTasks(’overig’);
};
```

F.7 CSS code of all pages

```
    :root {
    --color-primary-variant: #ECDFCC;
    --color-black: #432D12;
    --color-light:#A5B68D;

    --width: 84%;
    --mobile-width: 92%;

}

* {
    margin: 0;
    padding: 0;
    outline: 0;
    border: 0;
    text-decoration:none ;
    list-style: none;
    box-sizing: border-box;
}

html {
    scroll-behavior: smooth;
}

body {
    font-family: "Poppins", sans-serif;
    color: var(--color-black);
    background: var(--color-primary-variant);
    overflow: auto;
}

.container{

    width: var(--width);
    margin: 0 auto;

}

img{
    max-width: 100%;
    height: auto;
}

a {
    color: var(--color-black);
    transition: all 300ms ease;
}
```

```css
h1, h2, h3, h4, h5 {
    color: var(--color-black);
    font-family: "Lustria", serif;
    font-weight: 700;
}

h1 {
    font-size: 4rem;
    letter-spacing: -0.4 rem;
    line-height: 0.8;
}

h2 {
    font-size: 3.2rem;
    letter-spacing: -0.2rem;
}

h3{
    font-size: 2rem;
}

h4, h5 {
    font-size: 1.4rem;
    font-family: "Poppins", sans-serif;
    font-weight: 500;
}

h5{
    font-size: 0.9rem;
}

p{
    line-height: 1.6;
}

.logo{
    width: 13%;
    height: auto;


}



.btn{
    padding: 1rem 2rem;
    border-radius: 3rem;
    font-weight: 500;
    background: var(--color-black);
    color: var(--color-primary-variant);
    border: 1px solid transparent;
    cursor: pointer;
    transition: all 300ms ease;
}

.btn:hover {
    background: var(--color-black);
    color: var(--color-primary-variant);
    border: 1px solid var(--color-black);
}

.btn-primary{
    background: var(--color-primary-variant);
    color: var(--color-black);
    border: 1px solid var(--color-black)
}

.btn-small {
    margin: 0.3rem;
    padding: 0.2rem 0.7rem;
    font-size: 0.8rem;
    border-radius: 1cqmax;
    background: var(--color-primary-variant);
    color: var(--color-black);
    border: 1px solid var(--color-black);
    cursor: pointer;
    transition: all 300ms ease;
}

.btn-small:hover {
    background: var(--color-black);
    color: var(--color-primary-variant);
    border: 1px solid var(--color-black);
}

section{
    max-width: 100vw;
    padding: 12rem 0 0;
}

/*navBar*/
nav{
    width: 100vw;
    height: 6rem;
    display: flex;
    justify-content: center;
    align-items: center;
    position: fixed;
    z-index: 2;
    background: var(--color-light);
    box-sizing: border-box;

}

nav button {
    display: none;
}

nav .container{
    display: flex;
    justify-content: space-between;
    align-items: center;
    width: 100%;
    padding: 0 10rem;

}

nav .container ul{
    display: flex;
    gap: 3rem;
    align-items: center;
}

nav .container ul a {
    padding: 0 0 0.3rem;
    text-align: center;

}

nav .container ul a.active{
    border-bottom: 6px solid var(--color-primary-variant);
}

nav a:hover {
    color: var(--color-primary-variant);
}

.body {
    padding: 5rem;
}



/* ------------TASKS---------*/
```

```
#task-input {
    display: flex;
    flex-direction: column;
    justify-content: center;
    padding: 1rem;
    background: var(--color-light);
    border-radius: 1rem;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    width: 90%;
    margin: 3rem auto 2rem auto;
    align-items: center;
}



#add-btn{
    width: 30%;
}

.task-action-container {
    display: flex;
    flex-direction: row;
    align-items: center;
    gap: 20rem;
    margin-top: 1rem;
}

.timefilter {
    width: 500px;
    height: 100px;
    background: var(--color-light);
    padding: 1rem;
    border-radius: 1rem;
    border: 1px solid var(--color-black);
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    margin-right: 2rem;
    position: relative;
}

#daterange{
    height: 40px;
}


.task-select {
    display: grid;
    grid-template-columns: 1fr 1fr;
    gap: 1rem;
    justify-items: center;
    align-items: center;
    padding: 1rem;
    width: 100%;
    margin-bottom: 1rem;
}


input, select {
    width: 90%;
    height: 90%;
    margin: 0.5rem 0.5rem 0 0.5rem;
    box-sizing: border-box;
    font-family: "Poppins", sans-serif;
    color: var(--color-black);
    background: var(--color-primary-variant);
    border: 1px solid var(--color-black);

    padding: 1rem 0.5rem;
    align-self: center;
    border-radius: 3rem;
    font-weight: 500;

    cursor: pointer;
    transition: all 300ms ease;
}



.task-grid {
    display: grid;
    grid-template-columns: repeat(2, 1fr);
    grid-template-rows: repeat(2, auto);
    gap: 2rem;
    margin: 0 auto;
    width: 90%;

}

.task-box {
    background: var(--color-light);
    padding: 2rem;
    border-radius: 1rem;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

}

/* ----------------BIRDS-----------*/
.birdview-layout {
    display: flex;
    justify-content: space-between;
    align-items: flex-start;
    padding: 2rem;
}

.ground {
    position: fixed;
    top: 51%;
    left: 51%;
    transform: translate(-50%, -50%);
    width: 50vw;
    aspect-ratio: 5 / 3;
    background-color: #9a5b3c;
    border-radius: 1rem;
    overflow: hidden;


}

.icons-guide {
    position: absolute;
    bottom: 10px;
    left: 50%;
    transform: translateX(-50%);
    display: flex;
    gap: 2rem;
    flex-wrap: wrap;
    justify-content: center;
    text-align: center;
    width: 100%;
    font-family: "Poppins", sans-serif;
    font-size: 1rem;
    color: var(--color-black);
}

.icon-item {
    text-align: center;
    width: 100px;
}

.icon {
    font-size: 2rem;
    margin-bottom: 0.5rem;
}

.icon-item p {
    font-size: 0.9rem;
    color: var(--color-black);
}
```

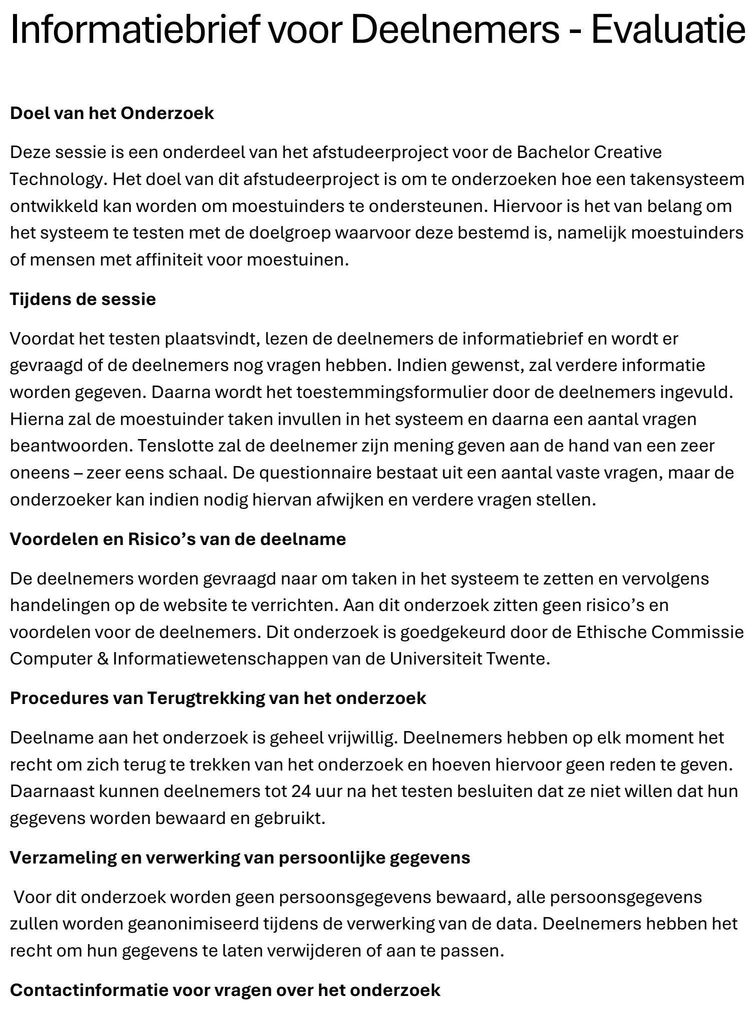
```
.vegetableform {
   width: 200px;
   height: 300px;
   background: var(--color-light);
   padding: 1rem;
   right: 8%;
   transform: translate(50%, 50%);
   border-radius: 1rem;
   box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
   margin-right: 2rem;
   position: relative;
}

.form-container {
   display: flex;
   flex-direction: column;
   margin-right: 20px;
}

#vegetableImage {
   max-width: 40px !important;
   max-height: 40px !important;
   width: 50%;
   height: auto;
   cursor: grab;
   display: block;
   overflow: hidden;
}

.vegetable-image {
   width: 100%;
   height: auto;
   object-fit: contain;
   pointer-events: none;
   overflow: hidden;
}


.ruler-wrapper {
   position: fixed;
   top: 50%;
   left: 50%;
   transform: translate(-50%, -50%);
   width: 50vw;
   aspect-ratio: 5 / 3;
}
```

```
.ruler {
    position: absolute;
    display: flex;
    font-size: 12px;
    color: black;
    align-items: center;

}

.ruler.horizontal {

    width: calc(100% + 14px);
    flex-direction: row;
    margin-left:-7px ;
    justify-content: space-between;
    height: 20px;

}

.ruler.vertical {
    height: calc(100% + 14px);
    flex-direction: column;
    margin-top: -7px;
    justify-content: space-between;
    width: 10px;
}

.ruler.horizontal.top {
    top: -20px;
}


.ruler.vertical.left {
    left: -20px;
}

.grid-container {
    display: grid;
    width: 100%;
    height: 100%;
    grid-template-columns: repeat(20, 1fr);
    grid-template-rows: repeat(12, 1fr);

}


.grid-item {
    max-width: 100%;
    max-height: 100%;
    border: 1px solid rgba(255, 255, 255, 0.3);
    display: flex;
    justify-content: center;
    align-items: center;
    position: relative;
    width: 100%;
    height: 100%;
}

.vuilnisbak {
    position: fixed;
    bottom: 5%;
    right: 2%;
    width: 150px;
    height: 150px;
    background-color: transparent;
    display: flex;
    justify-content: center;
    align-items: center;
    cursor: pointer;
}

.vuilnisbak img {

    max-width: 100%;
    max-height: 100%;
}

/*--------------------------HOMEPAGE----------*/
.garden-overview {
    padding: 2rem;
    text-align: center;
}

.grid-and-tasks {
    display: flex;
    justify-content: space-around;
    align-items: flex-start;
    gap: 2rem;
    margin-top: 2rem;
}

.timeform {
    width: 200px;
    height: 150px;
    background: var(--color-light);
    padding: 1rem;
    right: 8%;
    transform: translate(80%, 50%);
    border-radius: 1rem;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    margin-right: 2rem;
    position: relative;
}


#date-range {
    width: 100%;
    padding: 10px;
    font-size: 16px;
    border: 1px solid #ccc;
    border-radius: 4px;
}


.tasks-container {
    width: 30%;
    max-height: 80vh;
    overflow-y: auto;
    border: 1px solid #ccc;
    padding: 1rem;
    background-color: #f5f5f5;
    border-radius: 10px;
    box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);
}

#tasks-list li {
    margin: 0.5rem 0;
    font-size: 1rem;
    list-style: none;
}



.index-page .ground {
    transform: translate(-50%) perspective(400px) rotateX(50deg);
    transform-style: preserve-3d;
    position: relative;
}


.index-page .grid-item {
    transform: none !important;
    max-width: 100%;
    max-height: 100%;
    border: 1px solid rgba(255, 255, 255, 0.3);
    display: flex;
    justify-content: center;
    align-items: center;
    position: relative;
    width: 100%;
    height: 100%;
}


.index-page .vegetable-image {
    width: 100%;
    height: auto;
    object-fit: contain;
    position: absolute;
    top: 0;
    left: 0;
    transform: perspective(20px) rotateX(340deg) !important;
    transform-style: preserve-3d;
    transition: transform 0.2s ease-in-out;
}

.oogsten-icon, .wieden-icon, .zaaien-icon, .overig-icon, .overlap-icon {
    position: absolute;
    background: rgba(255, 255, 255, 0.8);
    border-radius: 50%;
    font-size: 12px;
    z-index: 10;
}

/* Popup styles */
.popup {
    display: none;
    position: fixed;
    z-index: 1;
    left: 0;
    top: 0;
    width: 100%;
    height: 100%;
    overflow: auto;
    background-color: rgba(0, 0, 0, 0.4);
}

.popup-content {
    background-color: white;
    margin: 15% auto;
    padding: 20px;
    border: 1px solid #888;
    width: 80%;
    max-width: 400px;
}
```

```
.popup-close {
   color: #aaa;
   float: right;
   font-size: 28px;
   font-weight: bold;
}

.popup-close:hover,
.popup-close:focus {
   color: black;
   text-decoration: none;
   cursor: pointer;
}
```

Appendix G

Information Letter User Tests



Informatiebrief voor Deelnemers - Evaluatie

Doel van het Onderzoek

Deze sessie is een onderdeel van het afstudeerproject voor de Bachelor Creative Technology. Het doel van dit afstudeerproject is om te onderzoeken hoe een takensysteem ontwikkeld kan worden om moestuinders te ondersteunen. Hiervoor is het van belang om het systeem te testen met de doelgroep waarvoor deze bestemd is, namelijk moestuinders of mensen met affiniteit voor moestuinen.

Tijdens de sessie

Voordat het testen plaatsvindt, lezen de deelnemers de informatiebrief en wordt er gevraagd of de deelnemers nog vragen hebben. Indien gewenst, zal verdere informatie worden gegeven. Daarna wordt het toestemmingsformulier door de deelnemers ingevuld. Hierna zal de moestuinder taken invullen in het systeem en daarna een aantal vragen beantwoorden. Tenslotte zal de deelnemer zijn mening geven aan de hand van een zeer oneens – zeer eens schaal. De questionnaire bestaat uit een aantal vaste vragen, maar de onderzoeker kan indien nodig hiervan afwijken en verdere vragen stellen.

Voordelen en Risico's van de deelname

De deelnemers worden gevraagd naar om taken in het systeem te zetten en vervolgens handelingen op de website te verrichten. Aan dit onderzoek zitten geen risico's en voordelen voor de deelnemers. Dit onderzoek is goedgekeurd door de Ethische Commissie Computer & Informatiewetenschappen van de Universiteit Twente.

Procedures van Terugtrekking van het onderzoek

Deelname aan het onderzoek is geheel vrijwillig. Deelnemers hebben op elk moment het recht om zich terug te trekken van het onderzoek en hoeven hiervoor geen reden te geven. Daarnaast kunnen deelnemers tot 24 uur na het testen besluiten dat ze niet willen dat hun gegevens worden bewaard en gebruikt.

Verzameling en verwerking van persoonlijke gegevens

Voor dit onderzoek worden geen persoonsgegevens bewaard, alle persoonsgegevens zullen worden geanonimiseerd tijdens de verwerking van de data. Deelnemers hebben het recht om hun gegevens te laten verwijderen of aan te passen.

Contactinformatie voor vragen over het onderzoek

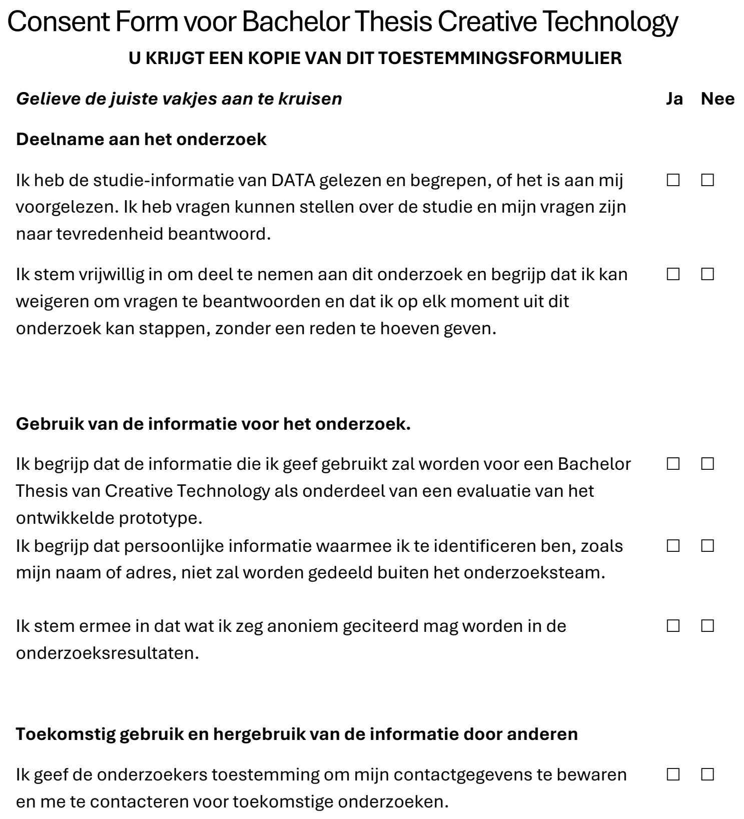
Indien de deelnemers vragen hebben of verdere informatie willen over het onderzoek, kunnen ze Daisy Baars contacteren via e-mail: d.baars@student.utwente.nl. Indien er klachten zijn over dit onderzoek, kunnen deze gestuurd worden naar Dennis Reidsma via e-mail: d.reidsma@utwente.nl

Contactinformatie voor vragen over uw rechten als onderzoekdeelnemer

Indien de deelnemers vragen hebben of informatie willen over het onderzoek, losstaand van de onderzoekers, kunnen ze contact opnemen met de secretaris van de Ethische commissie Computer & Informatiewetenschappen via e-mail: ethicscommittee-CIS@utwente.nl.

Appendix H

Consent Form User Tests



Consent Form voor Bachelor Thesis Creative Technology

U KRIJGT EEN KOPIE VAN DIT TOESTEMMINGSFORMULIER

Gelieve de juiste vakjes aan te kruisen	**Ja**	**Nee**
Deelname aan het onderzoek		
Ik heb de studie-informatie van DATA gelezen en begrepen, of het is aan mij voorgelezen. Ik heb vragen kunnen stellen over de studie en mijn vragen zijn naar tevredenheid beantwoord.	□	□
Ik stem vrijwillig in om deel te nemen aan dit onderzoek en begrijp dat ik kan weigeren om vragen te beantwoorden en dat ik op elk moment uit dit onderzoek kan stappen, zonder een reden te hoeven geven.	□	□
Gebruik van de informatie voor het onderzoek.		
Ik begrijp dat de informatie die ik geef gebruikt zal worden voor een Bachelor Thesis van Creative Technology als onderdeel van een evaluatie van het ontwikkelde prototype.	□	□
Ik begrijp dat persoonlijke informatie waarmee ik te identificeren ben, zoals mijn naam of adres, niet zal worden gedeeld buiten het onderzoeksteam.	□	□
Ik stem ermee in dat wat ik zeg anoniem geciteerd mag worden in de onderzoeksresultaten.	□	□
Toekomstig gebruik en hergebruik van de informatie door anderen		
Ik geef de onderzoekers toestemming om mijn contactgegevens te bewaren en me te contacteren voor toekomstige onderzoeken.	□	□

Handtekeningen

____________________ ____________________ ________

Naam deelnemer Handtekening Datum

Ik heb deelnemers nauwkeuring de informatiebrief zien lezen of ik heb de brief nauwkeurig voorgelezen en de deelnemer de kans gegeven vragen te stellen. Ik heb ervoor gezorgd dat de deelnemer begrijpt waarvoor hij/zij vrijwillig toestemming heeft gegeven.

_______________________ _________________ ________

Naam onderzoeker Handtekening Datum

Contact gegevens onderzoeker voor verdere informatie

Voor vragen of verdere informatie kunt u de onderzoeker Daisy Baars contacteren via e-mail: d.baars@student.utwente.nl.

Contact gegevens voor vragen over uw rechten als deelnemer

Als uw vragen heeft over uw rechten als deelnemer, meer informatie wilt of vragen wilt stellen losstaand van de onderzoeker, kunt u de secretaris van de Ethiek Commissie van Computer & Informatiewetenschappen contacteren via e-mail: ethicscommittee-CIS@utwente.nl .

References

Y. Ahmetoglu, D. P. Brumby, and A. L. Cox. Disengaged From Planning During the Lockdown? An Interview Study in an Academic Setting. *IEEE Pervasive Computing*, 20(4):18–25, Oct. 2021. ISSN 1558-2590. doi: 10.1109/MPRV.2021.3118900. Conference Name: IEEE Pervasive Computing.

Y. Ahmetoglu, D. Brumby, and A. Cox. Bridging the Gap Between Time Management Research and Task Management App Design: A Study on the Integration of Planning Fallacy Mitigation Strategies. In *Proceedings of the 3rd Annual Meeting of the Symposium on Human-Computer Interaction for Work*, CHIWORK '24, pages 1–14, New York, NY, USA, June 2024. Association for Computing Machinery. ISBN 9798400710179. doi: 10.1145/3663384.3663404.

V. Bellotti, B. Dalal, N. Good, P. Flynn, D. G. Bobrow, and N. Ducheneaut. What a to-do: studies of task management towards the design of a personal task list manager. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 735–742, Vienna Austria, Apr. 2004. ACM. ISBN 978-1-58113-702-6. doi: 10.1145/985692.985785.

M. Bernstein, M. Van Kleek, D. Karger, and m. schraefel. Information scraps: How and why information eludes our personal information management tools. *ACM Trans. Inf. Syst.*, 26(4):24:1–24:46, 2008. ISSN 1046-8188. doi: 10.1145/1402256.1402263.

A. E. Blandford and T. R. G. Green. Group and Individual Time Management Tools: What You Get is Not What You Need. *Personal and Ubiquitous Computing*, 5(4): 213–230, Dec. 2001. ISSN 1617-4909. doi: 10.1007/PL00000020.

R. Boardman and M. A. Sasse. "Stuff goes into the computer and doesn't come out". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, volume 6, pages 583–590, Vienna Austria, Apr. 2004. ISBN 1-58113-702-8. doi: 10.1145/985692.985766.

M. Egerer, B. Lin, J. Kingsley, P. Marsh, L. Diekmann, and A. Ossola. Gardening can relieve human stress and boost nature connection during the COVID-19 pandemic. *Urban Forestry & Urban Greening*, 68:127483, Feb. 2022. ISSN 1618-8667. doi: 10.1016/j.ufug.2022.127483.

A. El Khateeb. In an Endeavour for a New Vision for Personal Task Management: Exploring User Preferences in Personal Task Management Tools. Master's thesis, Dalhousie University, Halifax, Nova Scotia, Apr. 2019.

M. Haraty and J. McGrenere. Designing for Advanced Personalization in Personal Task Management. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, pages 239–250, Brisbane, QLD, Australia, June 2016. Association for Computing Machinery. doi: 10.1145/2901790.2901805.

A. J. Hoffman. Community Gardening, Volunteerism and Personal Happiness: "Digging In" to Green Space Environments for Improved Health. *Metropolitan State University*, 4:1–7, July 2018. ISSN 25730150. doi: 10.24966/PDA-0150/100015. URL `http://www.heraldopenaccess.us/fulltext/Psychiatry-Depression-&-Anxiety/Community-Gardening-Volunteerism-and-Personal-Happiness-Digging-In-to-Green-Space` `php`.

D. Hu, M. M. Bhuiyan, S. Lim, J. Wiese, and S. W. Lee. Unpacking Task Management Tools, Values, and Worker Dynamics. In *Proceedings of the 3rd Annual Meeting of the Symposium on Human-Computer Interaction for Work*, CHIWORK '24, pages 1–16, New York, NY, USA, June 2024. Association for Computing Machinery. ISBN 9798400710179. doi: 10.1145/3663384.3663402. URL `https://dl.acm.org/doi/10.1145/3663384.3663402`.

A. Kamsin. *Improving Tool Support for Personal Task Management (PTM)*. PhD, University College London, London, 2014.

M. Lu and P. Corriveau. From Personal to Collaborative Information Management: A Design Science's Approach. In G. Salvendy and M. J. Smith, editors, *Human Interface and the Management of Information. Interacting with Information*, pages 36–45, Berlin, Heidelberg, 2011. Springer. ISBN 978-3-642-21669-5. doi: 10.1007/978-3-642-21669-5_5.

A. H. Mader and W. Eggink. A Design Process for Creative Technology. In *Proceedings of the 16th International conference on Engineering and Product Design, E&PDE 2014*, pages 568–573. The Design Society, Sept. 2014. URL `https://research.utwente.nl/en/publications/a-design-process-for-creative-technology`.

D. Mizrachi and M. J. Bates. Undergraduates' personal academic information management and the consideration of time and task-urgency. *Journal of the American Society for Information Science and Technology*, 64(8): 1590–1607, 2013. ISSN 1532-2890. doi: 10.1002/asi.22849. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/asi.22849.

M. Soga, K. J. Gaston, and Y. Yamaura. Gardening is beneficial for health: A meta-analysis. *Preventive Medicine Reports*, 5:92–99, Mar. 2017. ISSN 2211-3355. doi: 10.1016/j.pmedr.2016.11.007.

R. W. White. Intelligent futures in task assistance. *Commun. ACM*, 65(11):35–39, 2022. ISSN 0001-0782. doi: 10.1145/3528083.

A. C. Williams, S. Iqbal, J. Kiseleva, and R. W. White. Managing Tasks across the Work–Life Boundary: Opportunities, Challenges, and Directions. *ACM Trans. Comput.-Hum. Interact.*, 30(3):48:1–48:31, June 2023. ISSN 1073-0516. doi: 10.1145/3582429.