Designing a Camera-based Automated Food Tracking App

Sytze de Witte
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
s.w.dewitte@student.utwente.nl

ABSTRACT

Nowadays there are many smartphone apps that let users journal their food intake. However, unlike most quantified self apps, food trackers still require a lot of manual work. All this manual journaling could deter users from regularly tracking their food intake and potentially living a healthier life. In this research, an optimal app design of a food tracker where as many functionalities as possible are automated using the smartphone camera is explored.

Keywords

quantified self, food tracking, food detection, smartphone application design

1. INTRODUCTION

When trying to lose, maintain or gain weight efficiently, tracking energy intake and macronutrients of food is a must [1]. Traditionally, this used to be done with pen and paper, looking up the calories of each product and adding them up. Nowadays, smartphone apps provide a simpler way to do this, improving dietary tracking consistency, and working just as effective as the traditional methods in helping with weight loss [2].

Popular examples of such apps are *MyFitnessPal*, *YAZIO* and *Virtuagym Food*, which are the most downloaded apps internationally in the iOS App Store. These apps are very similar and generally work as follows. When the user has consumed food or a drink, the user should enter its name, the portion size, the time when it was consumed, and the app will tell you how much the user has eaten that day and how many calories there are still left for his daily goal.

Although there are plenty of smartphone apps that help users with tracking their food intake, most of these apps, like the aforementioned examples, still involve a lot of manual work. Meals have to be selected manually, portion sizes have to be entered manually, and other details that potentially could be useful to track, such as when or where a meal was eaten, have to be classified manually.

All this manual work could deter users from using these apps and tracking their food. Furthermore, even when people are willing to put in the work, matching food items and estimating portion sizes still is challenging for many users. According to Chen et al. (2019), a common problem is that energy-dense and nutrient-poor foods are omitted in logbooks by users, partly due to "confusing portion sizes" and "time-consuming data entry", causing significant underestimation of energy intake (on average by 445 kcal) and intake of macronutrients. [3] In the same study, only 20% of the participants answered that they would continue to use *MyFitnessPal*, due to the aforementioned challenges affecting their motivation for long-term use.

That is where the idea of a simpler app design comes up, namely a food tracking app where the camera is used for logging food automatically. In such an app, information that is filled in manually in regular food tracking apps (food items, portion sizes, nutritional facts, etc.), will be detected using machine learning, in order to calculate the intake of energy and possibly other nutritional values.

As will be mentioned in the next section, there has already been done a lot of research into food recognition and portion size detection, which are crucial for the camera-based design to work in practice. Also, there already exists an app that uses the camera for logging food, which is proof that the technology can work. Therefore, this research will not focus on the technical aspects of food and portion size detection. Rather, this research is trying to investigate how the design of food tracking apps can be improved to minimize the efforts of logging food intake, making use of the upcoming and ever improving food recognition technology.

Since the camera plays such a fundamental role in a camera-based food tracking app, the app design will be developed from the ground up, rather than adding the camera to an existing regular app as an afterthought.

For this research, the optimal design for a camera-based food tracking app will be explored, a prototype app will be designed and the design will be user tested. From here, the first research question can be answered, namely: What is the optimal design for a camera-based food tracking app? (**RQ1**) After that has been done, the second research question can be answered: How does a camera-based automated food tracking app compare to a regular food tracking app? (**RQ2**)

2. RELATED WORK

This section will go over related work in designing food tracking apps. Besides existing food tracking apps (such as the aforementioned examples), which provided a great starting point for gathering common features, a literature review was performed. In order to perform this literature review, Google Scholar and Scopus are used, using search terms such as "food tracking", "food detection" and "app design".

36th Twente Student Conference on IT February 4th, 2022, Enschede, The Netherlands.

In 2018 and 2020 very similar camera-based solutions have been created by Ming et al. and Lu et al. respectively [4, 5]. For both researches an app was created that detects food and subsequently retrieves nutritional information. Both researches show very promising results in terms of food logging speed and detection accuracy. This indicates that there is great potential for this kind of food tracking apps. Whereas those researches focus on the implementation of the functionalities of such an app, the research that is done in this work will focus specifically on the design of such an app.

In 2019 research has been done by Luo et al. on the design of food tracking apps, working together with dietitians [6]. This research contains useful specific design choices and also a list of tracking items that are important according to dietitians. Examples of tracking items are meal type, meal time, meal location, portion size and so forth. These are all tracking items that could be automatically detected using a smartphone camera.

Creating pictures of food also has an advantage that contextual information of the meal is also stored. According to Zhang and Parker (2020) this is also essential to help people reflect upon and change attitudes about eating behaviors [7].

At last, much research has been done on detecting food [8, 9, 10, 11, 12] and portion sizes [4, 5, 13]. This indicates that creating a camera-based automated food tracking app is technically possible. However, further developing these functionalities is out of scope of the research that is proposed.

3. METHODOLOGY

The research was divided into three phases. In the first phase, interviews were held and a low fidelity prototype was tested. In the second phase, a high fidelity prototype was created and tested. Finally, in the third phase, the results of these tests were evaluated, from which the research questions could be answered.

3.1 Interviews

Interviews were held with both people that already use food tracking apps regularly and people who do not. Interviewing participants that already used food tracking apps helped with identifying processes that could be simplified or automated using the smartphone camera. Interviews with non-users helped to find ways to create an intuitive user experience for first time users food tracking apps. In total ten people were interviewed. The interview questions can be found in Appendix A, and the results can be found in Section 4.1

The interview was set up to be semi-structured, meaning that there was room for follow-up questions and clarifications. This allowed for the possibility for interviewees to bring up ideas that were not thought of beforehand. The structure of the interview is inspired by Research Methods in Human Computer Interaction, Chapter 8 [14].

The interview questions were divided into three sections: initial exploration, requirements gathering, and evaluation. The questions of the first two sections were asked before the lo-fi prototype, and the questions of the third section were asked afterwards.

3.2 Low fidelity prototype

Before the interviews were held, a low fidelity (lo-fi) prototype was created. The lo-fi prototype is a mock-up of an app, that was created using *XCode* and pen and paper, as can be seen in Figure 1. This lo-fi prototype was also tested by ten people. The interaction with the main components of the app was tested. The main goal is to find out if the general design is intuitive and all essential features are included.

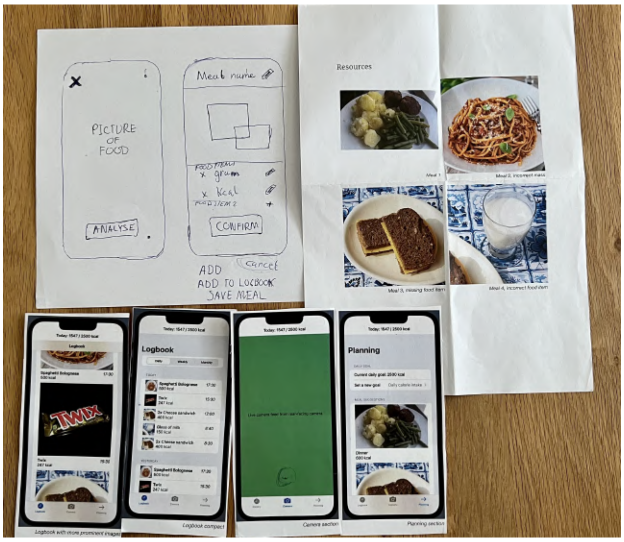


Figure 1: App screenshots, app sketches and pictures of meals were used for the lo-fi prototype test.

The interviews and lo-fi prototype testing were held during the same meeting. Answers to the interview and results of the prototype test were noted in a spreadsheet.

The goals of the low fidelity prototype test are to evaluate the different design concepts and the interaction with the main components. To find out whether the general design is intuitive and all essential features are included, but also to find out what ideas might not work or are not needed. Feedback from this test is used to create a high fidelity prototype.

This subsection will explain the general design of the prototype, which later was adjusted and extended in the hi-fi prototype. Furthermore, the testing procedure for the lo-fi prototype will be explained.

3.2.1 General prototype description

In this section, the design of the lo-fi prototype, as of before the first test will be described. This will also be the foundation for the hi-fi prototype.

Based on observations from existing food tracking apps and prototypes from other researches, a camera-based food tracking app should have three main tasks: logging meals, reflection on past meals, and planning future meals. That is why the idea for the prototype is to have an app with three top level sections. In the middle, there will be the camera section, which also will be the main screen of the app. This is where food can be logged by taking pictures. On the left, there will be the logbook section, where all consumed meals are listed. On the right, there will be the meal planning section, where personal goals can be set and future meals can be planned based on the personal goals. At the bottom there is a navigation bar, to switch between the sections.

In the **camera section** (Figure 3), the user takes a picture of the meal that they are about to eat. This section will be completely filled with a live camera preview, and a button to take an image. Optionally, an upload button could be added to the screen, so that a picture that has been taken earlier can still be analysed, if considered necessary by participants from the interview.

Ideally, the camera can recognize pictures of food on a plate, drinks in a glass, but also packaged food and drinks. Furthermore, the camera should also be able to recognize and discern multiple food items in one picture, so that the user does not have to take pictures of every individual food item of their meal. When a picture is taken, the user can view the picture instantly, to check if the picture is not blurry and the entire meal is in the frame of the picture. When the user wants to take the picture again, the user can discard the picture by pressing on a cross button. When the picture is taken correctly, the user can press on a button to analyze the picture. The meal is automatically analysed by the app using a machine learning model. This is wizarded during the lo-fi prototype. The user has to confirm the analysis for the meal to be added to the logbook. If any part of it is not accurate, the user can change the meal name, add, alter or remove food items, change the quantity of a food item, etc. The quantity of a food item can either be displayed by slices, or more exactly by mass or volume (e.g. grams, milliliters), or energy (e.g. kcal, joules), depending on feedback from participants. Whenever a meal is modified, this should be remembered by the app for the next time when that meal is detected.

In the **logbook section** (Figure 4a), all recorded meals are visible and can be accessed to add, edit or remove meals or food items. The optimal layout is something that will be explored during the lo-fi prototype test. One option is to make each meal as compact as possible, to have as many meals as possible visible at a glance. Another option is to make the images somewhat bigger (since all pictures of all meals are available), if that would make the user be more aware of their eating behavior. The added value of having photos of all the user's meals will be investigated. The nutritional facts that are displayed are also up to users.

The user is able to view their eating behavior over a larger period of time. When viewing the weekly or monthly overview, averaged values are visible, such as the average calorie intake of a time period. It also might be interesting to show the user the most eaten meals per time period. The exact data that is shown in these overviews is something to be explored during the lo-fi prototype testing.

Meal planning is very important for creating healthy eating routines. In this section goals for future meals can be set. (Figure 4b) Also, the user can get suggestions at a glance what to eat to meet those goals (for example to stay within a calorie limit).

Additionally, at all times, at the top of the screen the current situation of the day is visible, called the **Today section**.

The data that is visible in the Today section can be changed in the settings, but possible options are calorie intake or protein intake at that moment. This will be explored during the lo-fi prototype testing. The intent of this section is to be able to view crucial information of the current situation at a glance at any moment while using the app.

The Today section can be extended at any moment when it is pressed, which will show more details of the current situation such as showing the current intake of macronutrients or other specific nutrients a user is interested in.

This additional section could be considered unnecessary or redundant by some users, since the data is also visible under Today in the Logbook section. The necessity and the implementation of this section is something to be explored during the lo-fi prototype testing.

3.2.2 Testing procedure

For the lo-fi prototype test, screenshots of main screens of the UI that were created in XCode were printed on paper. Furthermore, additional screens were created using pen and paper, so that these sketches could be discarded easily without having invested too much time in creating them. Another advantage of using paper, is that additions could be drawn by either the participant or the interviewer. The exact tasks that the participant had to do are described in Appendix B. The participant can comment on any aspect of the prototype during the test, and will be asked questions about the experience after the test (see Appendix A.3).

3.3 High fidelity prototype

After the interviews and the lo-fi prototype tests, a high fidelity (hi-fi) prototype was created. The feedback from the interviews and the lo-fi prototype testing was taken into account for creating the hi-fi prototype. The hi-fi prototype is a working iOS app, which was developed in *XCode*. The goal of testing the hi-fi prototype is to find out how well the app design for the proposed camera-based food tracking app is being received by users, and to compare the usage of the prototype with the usage of a regular food tracking app. In the following subsections, some critical aspects of the hi-fi prototype will be explained.

3.3.1 Food recognition

To simulate the food recognition in the hi-fi prototype, *CreateML* was used to create a *CoreML*[1] model. For each of the nine meals that are used in the prototype test, 10 training images from the internet are used, resulting in a training set with 90 images of meals in total. To improve the model, all training images are also cropped, exposed and rotated, and noise and blur are added.

3.3.2 Data structure

The prototype consists of two very similar databases, a local logbook database, and the central meal database, where all meals are stored. When the food recognition model has recognized a meal (and in reality also the portion size, in the prototype this is wizarded), the corresponding *Meal* object can be added to the logbook database by the user. Both databases consist of two classes, namely a *Meal* class and *FoodItem* class. An object is an instance of a class. Each *Meal* object contains at least one *FoodItem* object. A *FoodItem* object can have a many to many relation with a *Meal* object. All attributes can be found in the class diagram in Figure 2.

The reason why each food item has both a *count* and a *quantity*, is because in the initial interviews it appeared that users prefer to log their meals using pieces and slices over exact masses or volumes. Here, the count is the amount of slices or pieces, and the quantity is the mass or volume of a single slice or piece. The value of a nutritional fact (energy, fats, carbs, protein, etc.) of a food item can be calculated by multiplying the count with the quantity and the density of that nutritional value. The total value of that nutritional fact for the entire meal can be retrieved by calculating the sum of that nutritional value for each food item in the meal.

If there are two or more of the same food items in a meal with different masses or volumes, they will each be counted

[1] `https://developer.apple.com/documentation/coreml`

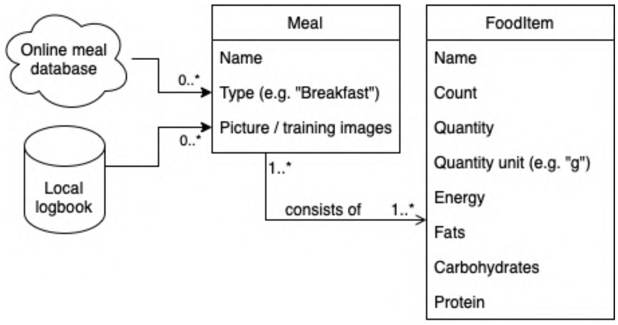
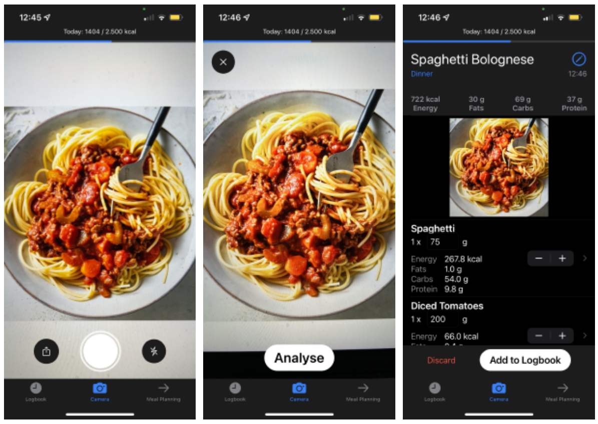
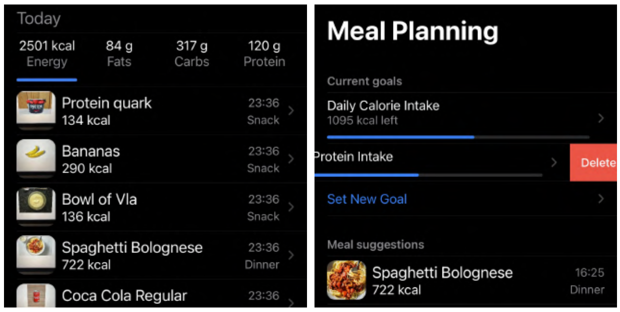


Figure 2: Class diagram of the classes used in the prototype.



(a) Live camera view. (b) Picture taken view. (c) Meal view.

Figure 3: Steps for logging a meal in the hi-fi prototype.



(a) Logbook section. (b) Meal planning section.

Figure 4: Screenshots from the hi-fi prototype. After adding a meal, it appears in the logbook. New meals can be planned and goals can be set or removed in the meal planning section.

as a separate food items. For example, a 150 ml glass and a 250 ml glass of orange juice will be stored as 1 x 150 ml orange juice and 1 x 250 ml orange juice, while if they were the same portion size, it would be stored as a single food item with a *count* of 2 and a *quantity* of 250 ml.

In general, meals can be divided in simple and complex meals. In simple meals, all food items are visible and can be detected distinctly. In complex meals, the food items are intertwined, and food items have to retrieved from a digital cookery book. Therefore, the chance that a meal is inaccurately logged, thus food items have to be adjusted, is greater with complex meals than with simple meals.

3.3.3 *User Interface Details*

The app is programmed in *Swift* using *SwiftUI*. Since the hi-fi prototype is built upon the lo-fi prototype, this section is an extension of Section 3.2.1.

The camera section now consists of three views. The first view is the live camera view. Here the user can take a picture of their meal or upload a meal from their photo gallery (Figure 3a). After a photo has been taken, the user can verify that the entire meal is clearly visible in the taken picture, and press a button to analyse the picture (Figure 3b). When the model has classified the meal, the corresponding *Meal* object from the meals database is loaded in the meal view (Figure 3c). If the meal is classified incorrectly, it can be replaced entirely by another meal manually, by pressing a button in the top right. In that case the user can search the correct meal in the meals database or create a new meal from scratch. In the prototype of Lim et al., that process is reversed. In their prototype, meals had to be selected from a candidate list after the picture was taken, and then the nutritional values of that meal were shown.

If the meal is classified mostly correctly, each individual food item can be added, edited, replaced or removed. The same view is used when a meal is opened from the logbook.

The meal screen is used in two ways, when a meal is added, and when a meal from the logbook is selected.

3.3.4 *Testing procedure*

The hi-fi prototype test was performed by running an iOS application containing the prototype on a physical iPhone and letting the participants control it. Afterwards, the participants were asked to rate certain metrics on a scale from 1 to 5 and name strong and weak points about certain parts of the app. See the literal testing procedure and assessment questions in Appendix C.

3.4 Evaluation

After all testing results are in, comparisons with traditional food tracking apps can be made, the research question can be answered. The metrics that will be evaluated will be completion time of in-app tasks, usability, appearance and loyalty (how likely they would use the camera-based food tracking app in the future).

4. RESULTS

This section contains an analysis of all the interviews, the lo-fi prototype feedback, and an evaluation of the hi-fi prototype tests.

4.1 Interview analysis

For this research, ten participants were interviewed. The median age was 20 years old and the mean age was 27 years old. Almost all participants had at least some experience with tracking food, with all of them using a smartphone app to do so. Furthermore, all participants responded that they only would track their food intake, when they have a goal. This might imply that a food tracking app must stimulate users to set goals, so that they are more likely to keep track of their food intake.

Half of the participants responded that they kept track of all their meals, the rest of them would skip too complicated and insignificant meals. Three participants log meals before eating, the rest afterwards. The preferred portion size unit for all participants was slices or pieces, rather than measuring portion sizes in grams or milliliters, but only if it is accurate enough. The participants were interested in only very few nutritional facts. All participants were interested in energy intake and some were also interested in macronutrients (fats, carbohydrates and protein). Therefore, in the hi-fi prototype, these were the only nutritional facts that were available to keep track of.

All participants would prefer using a camera-based food tracking app, rather than a regular manual app, with for some participants under the condition that it is sufficiently accurate.

All participants agreed that meals should be able to be photographed in as much different ways as possible. For example, not only when it lies on a plate, but also the package and the bar code on the package should be able to be analysed. This way, if a user deems taking a picture of a plate is not reliable enough for a specific meal, they can always have to possibility to log the food by taking a picture of the package or the bar code. As a last resort, when the camera does not work at all, or the meal already has been eaten, the meal can be entered manually.

Some users go as far as stating they would not mind photographing each ingredient separately, if that were to result in more accurate results.

Also when explicitly asked about the trade off between accuracy or completion time, the participants generally supported the idea of faster food tracking, but slightly preferred accurate food logging over swiftness. These findings are in line with the conclusions of Lim et al..

When asked about what section should be visible first when opening the app, 2 participants preferred the camera, 7 participants preferred the logbook, and 1 participant preferred the meal planning section. The participants that wanted the camera to be visible on opening, also preferred a shorter completion time over higher accuracy. The initial idea was to directly have the camera open, so that meals could immediately be scanned and analysed. However, as it turns out, most participants would rather first check their current food intake before scanning another meal. However, all participants that preferred the logbook also noted that the camera should be easily accessible from the logbook.

Integration with other apps was an aspect that was deemed very important by some participants. Those participants wanted the option to connect data from other health apps with a food tracker and vice versa. A possibility for example would then be to subtract the calories burnt from a connected activity app. All participants agreed with the option of an upload button, in case they had taken a picture of a meal using another (social media) app.

4.2 Lo-fi prototype

From all lo-fi prototype tests feedback was received on a wide variety of aspects.

The camera section was the most complete section and did not receive a lot of new ideas. For example, on idea was to have the upload button to be a thumbnail of the gallery, so that it is more clear that the user can upload their own pictures after taking them.

For the logbook section, two approaches for the layout were preferred most. One approach for the layout is to list all meals chronologically as done in the prototype. The other approach is to group each meal under the meal type (breakfast, lunch, dinner or snack), and have overview for each meal type. Most participants all stated that having an image increases eating behaviour awareness, but the image should not be larger than it is in the lo-fi prototype. So the logbook should not become a gallery of meals, but rather a list. The only nutrition fact in the list should be energy according to the participants, the rest can be visible when the meal is selected in the meal view. An interesting idea from one participant was to also show a nutrition fact that is overabundant in a meal. For example, if a user has a goal to eat less sugar, meals with too large amounts of sugar, that amount should be shown and highlighted.

	MyFitnessPal			Prototype		
	Min	Max	Mean	Min	Max	Mean
Usability	2	4	3.1	4	5	4.3
Compl. time	1	4	3.0	4	5	4.6
Appearance	2	5	3.6	3	5	4.1
Loyalty	1	4	3.0	3	5	4.1
Logging meals	3	5	4.4	3	5	4.1
Setting goals	2	4	3.4	3	5	3.7

Table 1: Average scores for user experience on a scale from 1 to 5 (n=7).

Participants were not necessarily interested in a weekly or monthly overview, but rather a general overview. Such an overview would be automatically generated, using images of meals, to recap the user's eating behaviour in a more story-like manner. In this recap, it is highlighted what the user is doing well and not so well to reach their goal, using colorful text highlighting main takeaways, the images from certain 'good' and 'bad' meals, and graphs displaying the user's progress over time.

For the meal planning section, participants suggested warnings for unrealistic goals, and the possibility to set broad abstract goals, such as lose a specific amount of weight, or eat more protein. However, most important was to set calorie goals and weight goals.

4.3 Hi-fi prototype

There were seven participants for the hi-fi prototype test. The average scores given to *MyFitnessPal* and the camera-based hi-fi prototype can be found in Table 1.

The camera-based prototype scored higher in all metrics, except the ability to log meals. The general reason given by the participants why the ability to log meals was lower in the prototype, was because the meal database was much more limited than the database of *MyFitnessPal*, and they expected that that would also be the case when the prototype would be worked out into a real app.

Even though the possible goals that can be set are very limited in the prototype (mainly due to time), the prototype still scored 0.3 points higher than the existing app. This might be caused by the fact that most participants had simple goals, that were easier to set in the prototype than in the existing app, where it was much more complicated.

Many participants also noted that the scores they have given, particularly for completion time and usability, are dependant on how accurate and reliable the food recognition is. In the prototype, the food recognition was working mostly well. However, if the food recognition was less accurate and they had to correct/adjust meals, they would have given the prototype definitely lower scores.

5. CONCLUSION

In this research, interviews were held and a lo-fi and hi-fi prototype were created and tested in order to answer the research questions, which are described below.

5.1 RQ1: Optimal design recommendations

The first research question: "What is the optimal design for a camera-based food tracking app?" can be answered by listing the most important points that were discovered during this research.

By far the most important aspect would be both the meal detection and portion size detection. Both need to be accurate enough, so that the amount of in-app actions to correct misjudged meals and portion sizes is significantly lower than the amount of actions to log a meal in a regular food tracking app. Many participants of both prototype tests have mentioned that this is the deciding factor for long-term use of a camera-based food tracking app. However, due to the fact that participants were only able to test the hi-fi prototype once, the results cannot confirm that users would actually go on to use a camera-based app more consistently than a regular app.

Secondly, the database containing all detectable meals should be large enough so that it contains all products available in supermarkets, and recipes from cookbooks. The database should be tailor-made for specific regions where specific foods are eaten. Otherwise, users would still have to add meals manually, undoing the time that is saved by using a camera-based food tracking app.

Thirdly, it is essential for an app to be able to set goals, after virtually all participants said that they would only use a food tracking app when they have a goal in mind to achieve. Additionally, the user should be given help when specifying the goal. For example, when the user inputs an abstract goal such as losing a certain amount of weight, the app should assist the user by calculating a daily calorie goal.

Lastly, the appearance should be simple, and comprehensive options and settings should be hidden in menus, so that the main view stay uncluttered and clear. This way, the app is both usable by novice users and more experienced users.

5.2 RQ2: Comparison to regular food tracking apps

The second research question was: "How does such a camera-based food tracking app compare to a regular food tracking app?". In order to answer this question, participants of this research gave scores to certain metrics on both the hi-fi prototype and *MyFitnessPal*, which is the most downloaded food tracking app internationally in the iOS App Store. The metrics that were evaluated are usability, completion time, appearance, loyalty, the ability to log meals and the ability to set personal goals.

Compared to a regular food tracking app, the camera-based hi-fi prototype scored higher on all metrics except one, namely the ability to log meals. This was partially caused by limitations of the meal database of the prototype. This result could have been expected, since the incompleteness of food databases is also a major complaint amongst *MyFitnessPal* users [3]. Thus, a limitation of this research is that this problem has not been resolved.

Although higher scores on usability, completion time and loyalty were expected, it should be mentioned that the prototype was working under optimal conditions. Namely, largely all meals and portion sizes were detected correctly, because the model had been trained on nine meals only.

Nevertheless, the results show that under the aforementioned optimal conditions, a camera-based food tracking app scores higher than a regular food tracking app on almost all important metrics.

6. FUTURE WORK

Due to a limited time frame, the scope of the research was relatively small. The general design of the camera-based food tracking app that was created in this research seems to be a step in the right direction. However, for future tests some parts could be implemented more extensively.

Namely, in the hi-fi prototype test, the participants had to log nine meals. These nine representative meals were selected beforehand, to avoid overcomplicating the test. Also, this was done so that the food recognition model only had to be trained on nine meals, thus reducing the chance of the attention being diverted to misclassified meals, instead of the design and workflow of the prototype. However, in future work the machine learning model that is responsible for the meal and portion size detection could be greatly improved, so that the online meal database can be fully implemented. This way, the completeness of the database can be tested by participants by choosing their own meals, instead of predefined meals.

Lastly, participants of this research were generally young, Dutch and male. So for future user tests, a larger and more diverse group of participants could be selected.

7. REFERENCES

[1] Jean Harvey, Rebecca Krukowski, Jeff Priest, and Delia West. Log often, lose more: Electronic dietary self-monitoring for weight loss. *Obesity*, 27:380–384, 03 2019. doi: 10.1002/oby.22382.

[2] Christopher M. Wharton, Carol S. Johnston, Barbara K. Cunningham, and Danielle Sterner. Dietary self-monitoring, but not dietary quality, improves with use of smartphone app technology in an 8-week weight loss trial. *Journal of Nutrition Education and Behavior*, 46(5):440–444, 2014. ISSN 1499-4046. doi: https://doi.org/10.1016/j.jneb.2014.04.291. URL https://www.sciencedirect.com/science/article/pii/S1499404614004692.

[3] Juliana Chen, William Berkman, Manal Bardouh, Ching Yan Kammy Ng, and Margaret Allman-Farinelli. The use of a food logging app in the naturalistic setting fails to provide accurate measurements of nutrients and poses usability challenges. *Nutrition*, 57:208–216, 2019. ISSN 0899-9007. doi: https://doi.org/10.1016/j.nut.2018.05.003. URL https://www.sciencedirect.com/science/article/pii/S0899900718303678.

[4] Zhao-Yan Ming, Jingjing Chen, Yu Cao, Ciaran Forde, Chong-Wah Ngo, and Tat Chua. *Food Photo Recognition for Dietary Tracking: System and Experiment*, pages 129–141. 01 2018. ISBN 978-3-319-73599-3. doi: 10.1007/978-3-319-73600-6_12.

[5] Ya Lu, Thomai Stathopoulou, Maria F. Vasiloglou, Lillian F. Pinault, Colleen Kiley, Elias K. Spanakis, and Stavroula Mougiakakou. gofoodtm: An artificial intelligence system for dietary assessment. *Sensors*, 20(15), 2020. ISSN 1424-8220. doi: 10.3390/s20154283. URL https://www.mdpi.com/1424-8220/20/15/4283.

[6] Yuhan Luo, Peiyi Liu, and Eun Kyoung Choe. Co-designing food trackers with dietitians: Identifying design opportunities for food tracker customization.

In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–13, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450359702. doi: 10.1145/3290605.3300822. URL `https://doi.org/10.1145/3290605.3300822`.

[7] Yixuan Zhang and Andrea G. Parker. Eat4thought: A design of food journaling. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, page 1–8, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368193. doi: 10.1145/3334480.3383044. URL `https://doi-org.ezproxy2.utwente.nl/10.1145/3334480.3383044`.

[8] Evgeniy Miasnikov and Andrey Savchenko. *Detection and Recognition of Food in Photo Galleries for Analysis of User Preferences*, pages 83–94. 06 2020. ISBN 978-3-030-50346-8. doi: 10.1007/978-3-030-50347-5_9.

[9] Hokuto Kagaya, Kiyoharu Aizawa, and Makoto Ogawa. Food detection and recognition using convolutional neural network. In *Proceedings of the 22nd ACM International Conference on Multimedia*, MM '14, page 1085–1088, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450330633. doi: 10.1145/2647868.2654970. URL `https://doi.org/10.1145/2647868.2654970`.

[10] Yoshiyuki Kawano and Keiji Yanai. Food image recognition with deep convolutional features. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, UbiComp '14 Adjunct, page 589–593, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450330473. doi: 10.1145/2638728.2641339. URL `https://doi.org/10.1145/2638728.2641339`.

[11] Yoshiyuki Kawano and Keiji Yanai. Foodcam: A real-time food recognition system on a smartphone. *Multimedia Tools and Applications*, 74, 04 2014. doi: 10.1007/s11042-014-2000-8.

[12] Brian Y. Lim, Xinni Chng, and Shengdong Zhao. Trade-off between automation and accuracy in mobile photo recognition food logging. In *Proceedings of the Fifth International Symposium of Chinese CHI*, Chinese CHI 2017, page 53–59, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450353083. doi: 10.1145/3080631.3080640. URL `https://doi-org.ezproxy2.utwente.nl/10.1145/3080631.3080640`.

[13] Gianluigi Ciocca, Paolo Napoletano, and Raimondo Schettini. Food recognition and leftover estimation for daily diet monitoring. In Vittorio Murino, Enrico Puppo, Diego Sona, Marco Cristani, and Carlo Sansone, editors, *New Trends in Image Analysis and Processing – ICIAP 2015 Workshops*, pages 334–341, Cham, 2015. Springer International Publishing. ISBN 978-3-319-23222-5.

[14] Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. Chapter 8 - interviews and focus groups. In Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser, editors, *Research Methods in Human Computer Interaction (Second Edition)*, pages 187–228. Morgan Kaufmann, Boston, second edition edition, 2017. ISBN 978-0-12-805390-4. doi: https://doi.org/10.1016/B978-0-12-805390-4.00008-X. URL `https://www.sciencedirect.com/science/article/pii/B978012805390400008X`.

APPENDIX

A. INTERVIEW

A.1 Food tracking apps (initial exploration)

- Do you have any experience with tracking food intake?
- Have you ever used a smartphone app for food tracking? (for example: MyFitnessPal, Virtuagym Food, etc.)
- When do you (or would you) track your food intake? (When I am on a diet / Always / Never / Other, namely when . . .)
- How often do you (or would you) use a food tracking app? (For every meal / For most meals / For some meals / Never)
- When are you most likely to log a meal using a food tracking app? (Before preparing a meal / After preparing a meal / After eating a meal / At the end of the day / Other, namely . . .)
- What portion size unit do you prefer to use when keeping track of food intake? (Slices / Mass/volume, e.g. grams, milliliters / Energy, e.g. kcal, joules / Depends on the food)
- What facts of a meal would you like to keep track of? (Just the name of meal / Quantity / Energy / Specific nutrients / Other, namely . . .)

A.2 Camera-based food tracking app (requirements gathering)

- Would you prefer using a camera-based food tracking app over logging food intake manually?
- What should the camera-based app be able to recognize? (Food on a plate / Drinks in a glass / Packaged food / Barcodes of food packages / Other, namely . . .)
- Would you use a camera-based food tracking app, if ingredients had to be photographed separately for a more accurate reading?
- Accuracy versus completion time. What do you prefer? Less accurate food logging, but faster and easier process of logging meals. / More accurate food logging, but slower and more difficult process of logging meals.
- What should be directly visible when opening the app? (The camera, to log food / The current food intake / Meal suggestions / Other, namely . . .)
- Would you consider it necessary to be able to upload pictures to be able to analyse those meals as well?
- Is there any metric besides completion time, usability and appearance, that you would consider as important for judging a camera-based food tracking app?

A.3 Evaluation Lo-fi prototype

A.3.1 Camera section

What is your general opinion about the Camera section? Is there anything missing?

A.3.2 Logbook section

What is your general opinion about the Logbook section? Is there anything missing?

Does viewing the photos of all the meals eaten on a day increase awareness of your eating behavior, rather than a list of meals?

What nutrition facts would you like to be visible directly in the logbook? (without pressing on the meal for more information)

Would you be interested in a weekly, monthly or yearly overview of eating behavior besides the daily overview? If so, what data would you like to see in such overviews?

A.3.3 Planning section

What is your general opinion about the Logbook section? Is there anything missing?

Would you consider it useful to receive suggestions about what to eat to reach a certain goal?

A.3.4 General questions

Do you think having the Today section visible at all times is useful?

Are there any problems that you think might arise when using an app like the prototype?

Are there any things that are missing from the prototype in your opinion? (which should be included in the next prototype)

B. LO-FI PROTOTYPE TASKS

B.1 Camera section

- Log Meal 1. Meal 1 consists of three different food items.
- Discard the taken image. This can be done when the photo is not clear.
- Log Meal 1 again.
- Analyse Meal 1. All three food items are recognized correctly.
- Add Meal 1 to the Logbook.
- Log Meal 2 and analyse it. Meal 2 is a typical spaghetti bolognese. It is recognized correctly, but the quantity is estimated twice as high as the correct value.
- Correct Meal 2 by editing the mass, and add it to the Logbook.
- Log Meal 3 and analyse it. Meal 3 are two cheese sandwiches. The bread is recognized, but the sandwich filling (the cheese) is not.
- Correct Meal 3 by adding the missing food item, and add it to the Logbook.
- Log Meal 4 and analyse it. Meal 4 is a glass of buttermilk. However, it gets recognized as a glass of whole milk, which has twice as many calories.
- Correct Meal 4 by editing the food item, and add it to the Logbook.
- Go to the logbook section.
- Go to the planning section.
- See how many calories you have consumed today (using the Current situation view).

B.2 Logbook section

- Remove a meal.
- Edit a meal.
- Go to the weekly and monthly overview.

B.3 Planning section

- Set a daily calorie intake goal.
- See food suggestions
- Select a suggestion, add it to the logbook.

C. HI-FI PROTOTYPE TESTING

Perform the tasks first using the existing app *MyFitnessPal*, then using the prototype.

- Determine your personal goal, and try to set it in the app.
- Log the following food using the app (in your own preferred way).
  - Bread with cheese (breakfast)
  - Glass of milk (breakfast)
  - Bread with fried egg (lunch)
  - Bar of Twix (snack)
  - Can of Cola (snack)
  - Spaghetti Bolognese (dinner)
  - Dessert: small bowl of vla (dinner)
  - Two bananas (snack)
  - Protein quark (snack)