Prototype #4 Hi-Fi Prototype

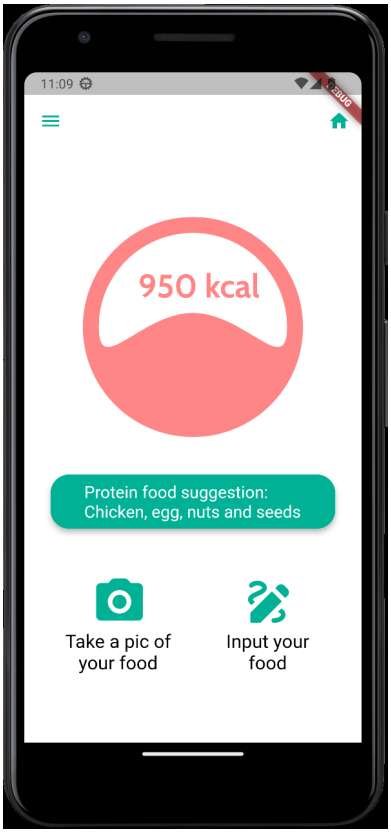
4. **Prototype #4 Hi-Fi Prototype**

    a. Design

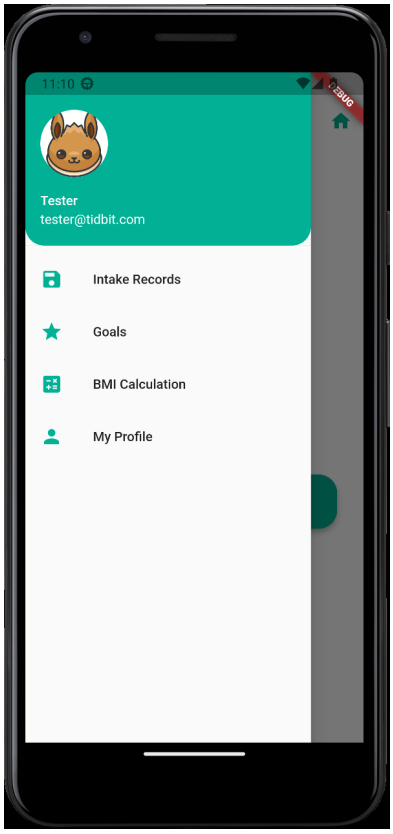
    In this prototype, we maintained the overall user interface design from the low-fidelity computer prototype, incorporating the modifications suggested by the previous user study. Along with the implementation of our application’s user interface, we will also introduce the use of a Deep Learning model in detecting food and calories.

        i. Frontend Implementation

        We developed an Android application using the Flutter framework. The final prototype encompassed the majority of the important features, and we utilized the **Figma to Code** plugin [7] to assist us in the implementation process. To begin with, let's discuss the home page, which closely resembled the design from Figma, with minor modifications made to the icons on the buttons. Additionally, we successfully incorporated a user profile header into the application's menu, utilizing the **UserAccountsDrawerHeader** widget, and included icons for each option within the menu.

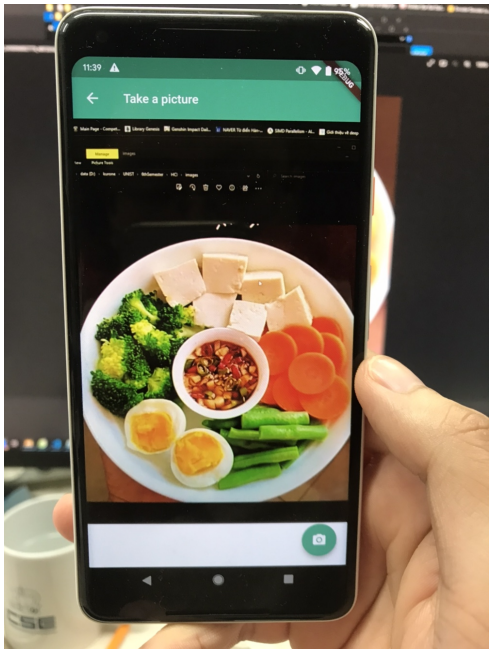


Home page

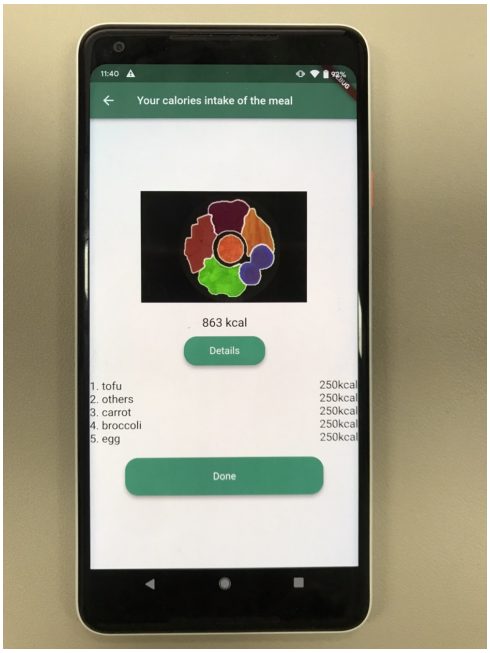


Application's Menu

The next UI component we implemented was taking a picture of the food by utilizing Flutter's camera plugin [8]. We also slightly modified the layout by introducing the app bar on top of the screen and moving the camera button to the lower right corner of the phone. It can easily detect the food but note that the displayed calories were probably incorrect due to not having connected to a food database.



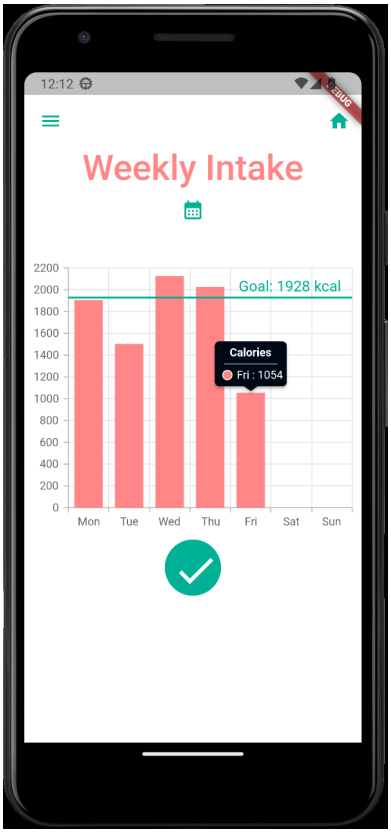
Taking a picture of the food



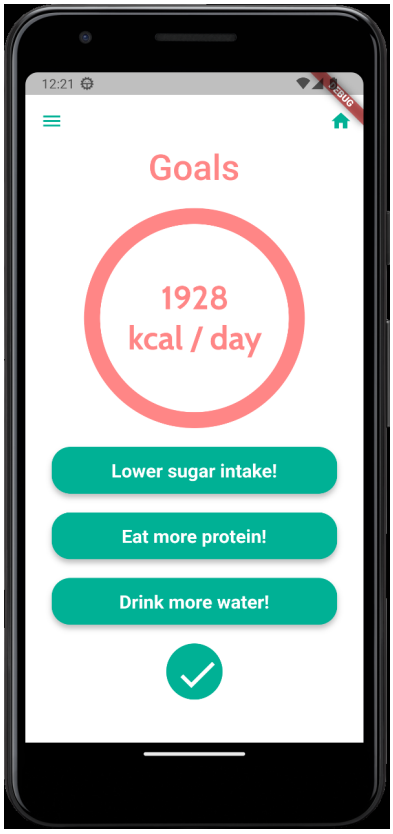
Food detection results

The next implemented feature was the Intake Records of the user. The bar chart was created by the use of a powerful visualization plugin, namely **syncfusion_flutter_charts** [9]. The green horizontal line is the goal set by the user when registering to our system. The user can also interact with it by tapping each bar to view a tooltip that provides detailed information about how many calories they have consumed during the chosen day.

One of the last UI components left to be implemented was the display of the user's daily goal. This was done simply by utilizing the **ListView** widget to display the suggestions from our system. The number of calories would be generated based on the user's registration profile.

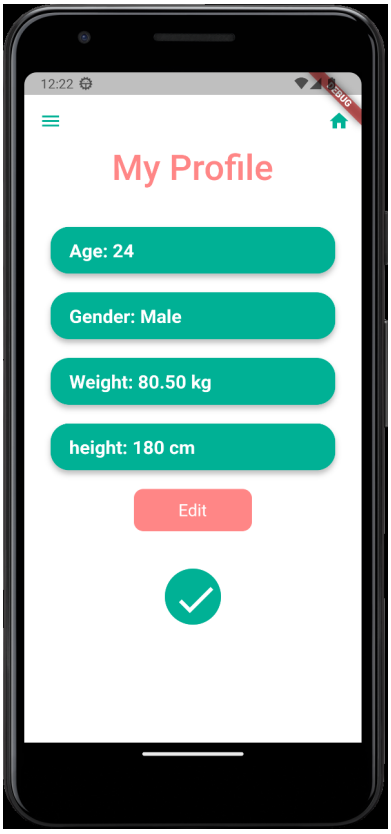


Intake Records



User's daily goal

Last but not certainly not least, the user's profile page was also implemented quite quickly as the page just consists of simple information from the user.

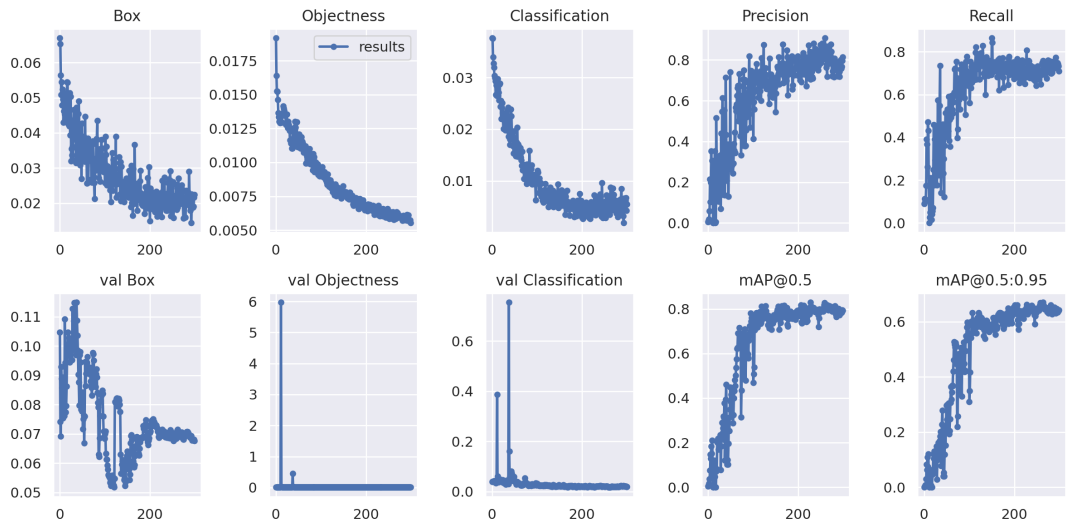


My Profile Page

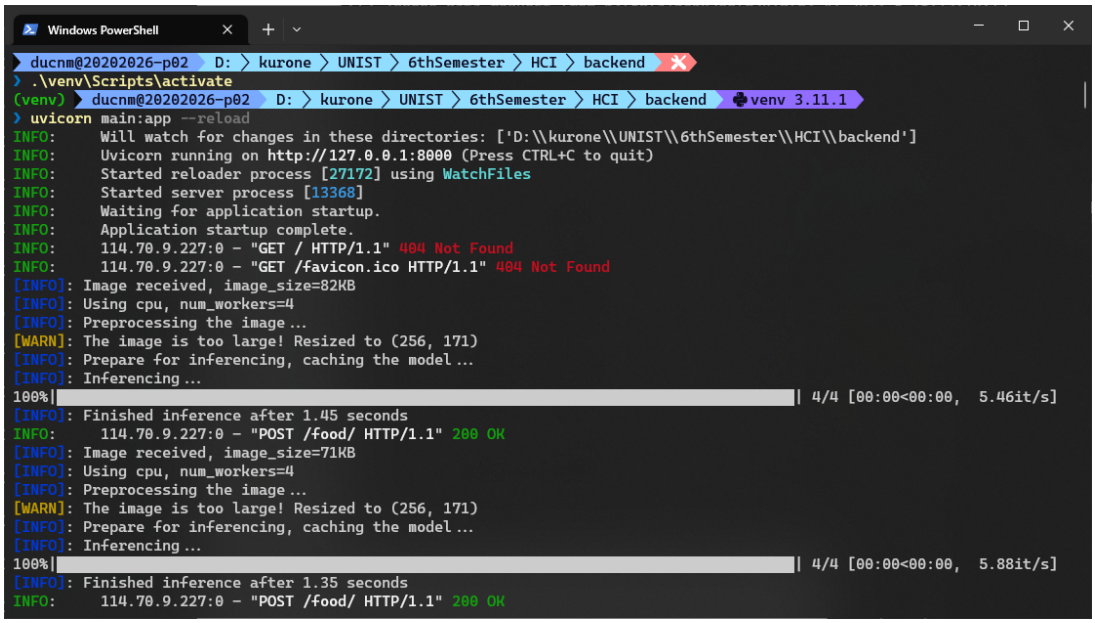
ii. Backend Implementation

The main innovation in our product is the integration of Artificial Intelligence, hence, a backend server was needed to handle complex computations to automatically recognize and calculate the calories from the user's taken image. To achieve this, we used Python and FastAPI package [10] to quickly set up the API. The overall flow of the system is the same as in the Proposed Solution section.

To incorporate the proposed Deep Learning model into our application, we decided to fine-tune the YOLOv7 architecture from Wang, et. al. [11] on the FoodSeg103 data set [12]. Please refer to the following figure for detailed training results. Our backend system takes 1.4 seconds on average to process the user's input. To deploy our system, we used ngrok [13].



Training results



Backend server using FastAPI

b. Design Justification

The feedback we received from participants in the previous user study highlighted the effectiveness of the layout and the intuitiveness of the interaction elements. Therefore, we decided to preserve the core design principles while making minor adjustments to enhance the user experience when implementing the high-fidelity prototype.

We chose Flutter to implement our application because it provided us with a rich set of pre-built UI components and widgets, allowing us to create a visually appealing and consistent user interface across different devices and

screen sizes. The framework's extensive widget library offered us flexibility in designing interactive elements, such as buttons, text fields, and navigation menus, ensuring a smooth and intuitive user experience. Also, Flutter is a cross-platform framework developed by Google, hence, we can easily deploy our application to the iOS platform (if needed) with minimal modification of our codebase.

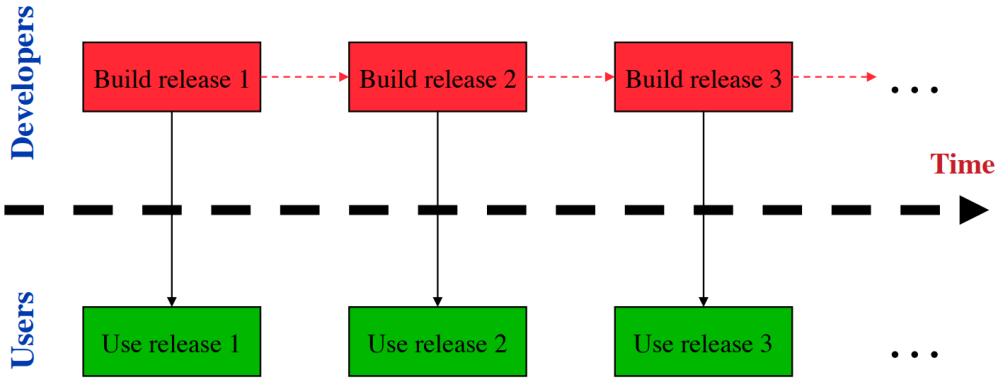
In the overall high-fi design of our application, we encountered a challenge with the buttons' icons due to limitations in Flutter's assets library. Specifically, some of the icons utilized in the previous prototyping stage on Figma were not available within Flutter's default icon set. We tried our best to find the best-fitted icons to replace those in Figma to ensure a cohesive and visually appealing user interface.

In the application's menu, we maintained a design that closely resembled the one in Figma but incorporated a user profile header in the application's menu and icons for each option to enhance the menu's visual appeal and usability. We did this because human visual perception is much better at recognizing visual cues such as shapes and colors. Therefore, we added those icons to further improve the overall user experience and learnability.

To visualize the user's intake data in an informative and visually appealing manner, we incorporated a powerful visualization plugin called **syncfusion_flutter_charts** [9]. This plugin had more than 2.5 thousand likes on Flutter’s package manager page, and provided us with a wide range of customizable chart types and options, allowing us to quickly create a dynamic and interactive bar chart.

It is important to note that the resulting image, after taking a picture of food, undergoes segmentation into different categories of foods with different colors. By assigning different colors for each type (qualitative color scheme) of food rather than boxes, we utilize human great visual perception to classify different items as colors have a high ranking in the visual perception hierarchy for categorical attributes [15]. This segmentation process significantly enhances the interpretability of our product, ultimately earning the trust of our customers. By categorizing the different foods within the image, users can easily understand and analyze their dietary choices. This transparency and clarity in presenting the segmented categories instill confidence in our customers, as they can trust the accuracy and reliability of our application's food recognition capabilities.

During the development process, it is crucial to gather feedback from targeted users at an early stage. By doing so, we can identify potential issues, gather valuable insights, and make necessary improvements before releasing the full version of the application. Therefore, to develop our product, we decided to follow the Phased Software Development Process as it promotes modularity, allowing early feedback and frequent releases and updates [14]. In this development process, we don't release the full version right away but ensure the application is stable, secure, and performs optimally, which explains some features that exist in Figma but do not in our current state of the product.



Phased Software Development Process [14]

For the backend server, we utilized FastAPI, a high-performance web framework for building APIs with Python. FastAPI offers a range of features that enable fast, easy, efficient, and scalable development [10], making it an ideal choice for our application. With FastAPI, we were able to create a robust and reliable backend that could handle incoming requests, process data, and provide responses in a fast and efficient manner. Furthermore, the choice of ngrok [13] to deploy our server is simple: a free domain that can be accessed outside of our LAN network.

In terms of the Deep Learning model, we employed the YOLOv7 architecture [11] (recently accepted as a conference paper at CVPR), a state-of-the-art object detection model known for its superior accuracy and speed in real-time image analysis tasks. It can do the inference at real-time speed on GPU, and also relatively fast on just a single CPU. By training the model on the FoodSeg103 data set [12], we enabled it to detect and identify different food items within images captured by the application. The utilization of

YOLOv7's advanced algorithms and neural network architecture [11] ensured accurate and real-time food recognition, enhancing the overall functionality and user experience of our application.

c. User Study

i. Participants

5 UNIST students are part of our target user group but had no prior exposure to our product through previous user studies. 2 of them are third-year design students with expertise in color schemes and product presentation, providing valuable insights into the design aspects.

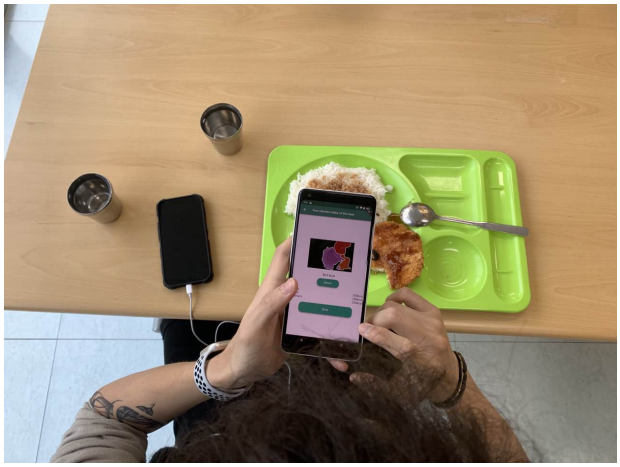
ii. Tasks

- Start to learn how to use: Navigate to different pages of the app
- Check your profile
- Check your health goals
- Take an image of food and see the results
- Check your intake record

After finishing all of the interaction tasks, users were asked to complete a long survey about their satisfaction and giv

iii. Research Method

We

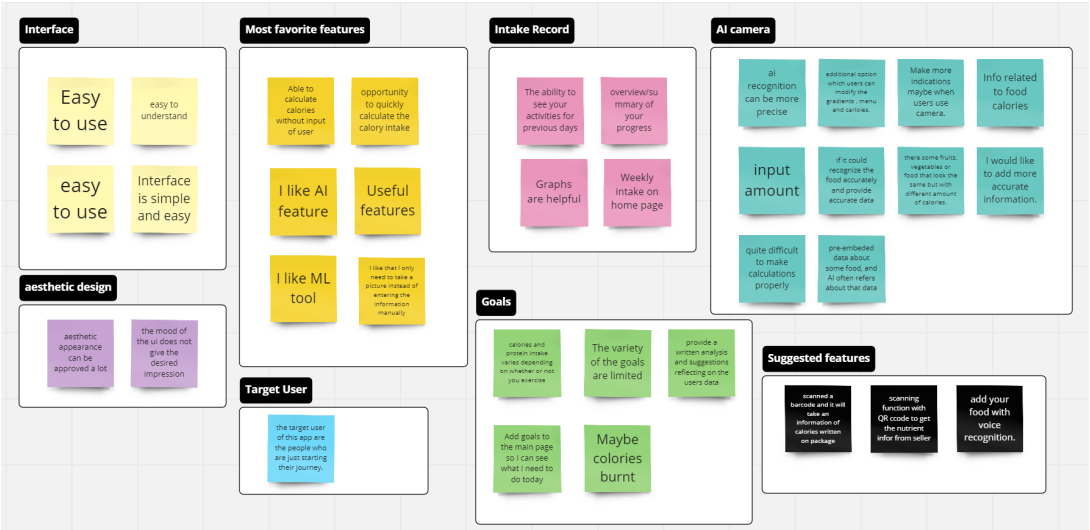


Users are interacting with Tidbit Hi-Fi prototype.

iv. Results

- Time consumption for an inexperience user to do these task as below:

- This is the affinity diagram analysis of users' feedback for 3 questions: what to improve, what …..



Affinity Diagram Analysis of users' feedback.

d. Insights and Reflections

(Future note for the writer: The reason that the AI system performed poorly on the user study is in the training data set, most foods are not Korean food (mostly Western foods), but in the field study, which was in the cafeteria, most foods were Korean foods → the accuracy of the system was not as expected. In the future, we would like to extend the training data set to include Korean foods → help targeted users at UNIST.

You can write some more about this…)

Reference

[7] Ferrari, https://www.figma.com/community/plugin/842128343887142055/, last accessed: June 2023.

[8] Flutter, https://pub.dev/packages/camera, last accessed: June 2023.

[9] Syncfusion, https://pub.dev/packages/syncfusion_flutter_charts, last accessed: June 2023.

[10] FastAPI, https://fastapi.tiangolo.com/, last accessed: June 2023.

[11] Wang, C., Bochkovskiy, A., & Liao, H.M. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *ArXiv, abs/2207.02696*.

[12] Wu, X., Fu, X., Liu, Y., Lim, E., Hoi, S.C., & Sun, Q. (2021). A Large-Scale Benchmark for Food Image Segmentation. *Proceedings of the 29th ACM International Conference on Multimedia*.

[13] ngrok, https://ngrok.com/, last accessed: June 2023.

[14] Kim (Spring 2023), Lecture 5: Development Process, CSE364 - Software Engineering, UNIST.

[15] Munzner, T. (2015). *Visualization Analysis and Design*. CRC Press. ISBN: 9781498759717