

CS3105: Documentation App Development

Written by: **Team Orion (Team 5)**

CS20B026 Ganesh Priyatham

CS20B034 Sachin Sahu

CS20B040 Siddhartha G

CS20B043 Sirish Sekhar

CS20B054 Anush Mangal

Problem Statements:

To make an efficient system to keep track of a student's mess meal consumption, thereby giving a digital solution that keeps track of a student's calorie intake on a daily basis and also tracks mess meals consumed over the entire campus.

Proposed Solutions:

The two SDG goals targeted are as follows:

1. **Good Health and Well-being:** Ensure healthy lives and promote well-being for all at all ages (Goal 03)
2. **Responsible Consumption and Production:** Ensure sustainable consumption and production patterns (Goal 12)

Our team made an app for students which helps them to keep track of their eating habits. Our app reminds them of the daily calorie intake requirements and how much they have fulfilled by generating push notifications.

As we all know, students generally skip meals without considering the health issues they might face. This also results in food waste, as the mess committee needs to get the proper count.

So, our app targets **SDG Goal 03** to ensure the healthy life of students and **SDG Goal 12** to ensure sustainable consumption of food produced.

The students can mark their meals, and our app analyzes and reports the calorie requirements fulfilled and remaining. This will help students to analyze their eating habits and let them plan their calorie requirements.

Software engineering principles used:

Since it was only a short-term project, we used the **Classic Waterfall Model** to build our application.

The first step was to decide the problem statement, that would fulfill the Sustainable development goals. Then we came up with the idea of a mess app named **MessTime**.

Requirements

The next step was to decide the requirements specifications and spend a considerable amount of time finalizing the requirements. It was done to ensure that the least amount of errors propagates to the next layers of the **Classical Waterfall Model**. Our requirements include calculating the total amount of calories consumed by the student, notifying the student to take food, visualizing the calories consumed for different days of the week, and also visualizing the daily consumption status as a progress bar which indicates the amount of calories yet to be consumed. This requirement specifications stage took **6-7 days**.

Design

The next step was to **finalize a design** and the basic user interface of our application. We all team members sat together and discussed different possibilities for the user interface. We sat with pen and paper and brainstormed over different versions of the design and finally the design evolved to the current version of UI. Our design is **simplistic and user friendly**. It displays the calories using progress bars as well as point charts made using **fl_chart** library for flutter, which enhances the UI and grabs the attention of users. This design part took **4-5 days** to complete. This design was the **Prescriptive Architecture** of our app.

Our design decision was such that our app **compatible with iOS and Android**, i.e. it will run on both Android and iOS devices to follow the **portability quality**. Thus, we **finally** decided to use **Flutter and Dart** for our app development project.

Implementation and coding

The next step was implementation and coding. We implemented our ideas and the prescriptive architecture gradually changed to the current **Descriptive Architecture**. Both of them are slightly different as suggested by the software engineering principles.

Implementation of our app also took a considerable amount of effort as we all had to learn **Flutter and Dart** for **Android App Development**.

We also faced a lot of **Accidental Difficulties** during implementation such as **Flutter installation problems, hard syntax for Dart language, and understanding the Child based component architecture of Dart**. These accidental difficulties took a considerable amount of time, effort and energy, and resulted in mental frustration. We could not even install Flutter & Android Studio on one of the member's laptops due to some technical configurational issues. All this implementation took more than **7 days**.

Testing and Evaluation

Our next step was to evaluate our app. Most of the evaluation was done during implementation, but we still had to evaluate our app after implementation. We installed the APK file in our mobile phones and tested the working application for one full day. It was during this time, that we figured out some bugs like icon colors not changing, long API call time delay, etc. We fixed these issues during this phase.

Anticipation For the Change:

The mess menu is kept in a JSON file and in the entire code, it has been used by referring to it. Hence if the mess menu changes in the future, the app will continue to support just by changing the menu data in that file.

This calorie data will also be updated since they are API calls!

Portability:

The app is supported for both Android and ios users.

Note:

The calorie data of each food item is taken through an API call from [nutritionix](https://nutritionix.com/) and we do not guarantee its accuracy.

Possible Extensions for the App:

- 1) Capturing the attendance through QR code/Id Card
- 2) User Login System
- 3) Using the data that is being collected to predict the line in the mess!
- 4) More proper database

User Manual

1) Home page:

It consists of:

MessCard:

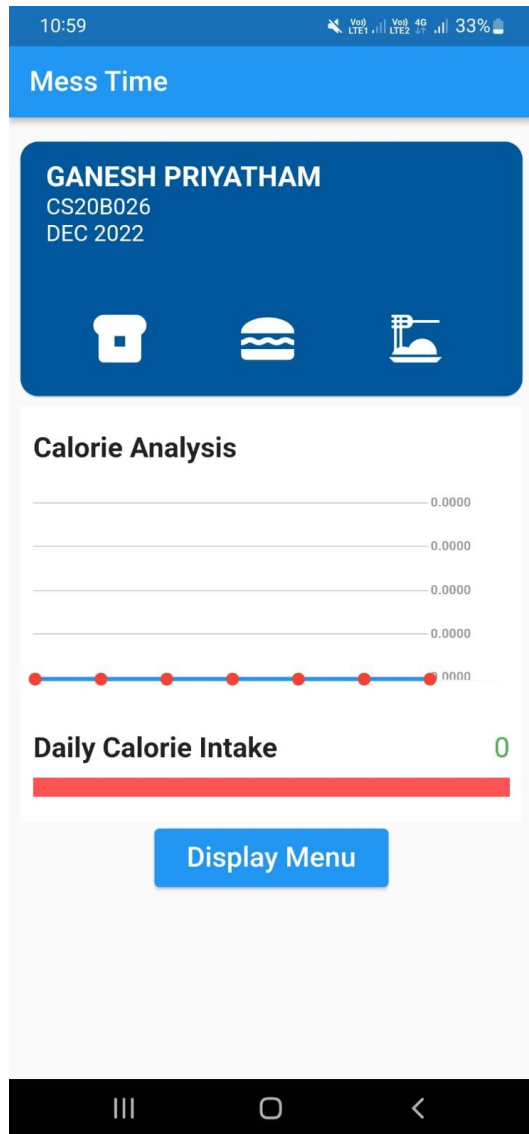
- a) It three icons each signify breakfast, lunch, and dinner.
- b) All these icons are validated based on time i.e user can't open the breakfast page before 8:00 am similarly, for the remaining
- c) The user can fill in the calorie intake data only once per every meal

Calorie Graph:

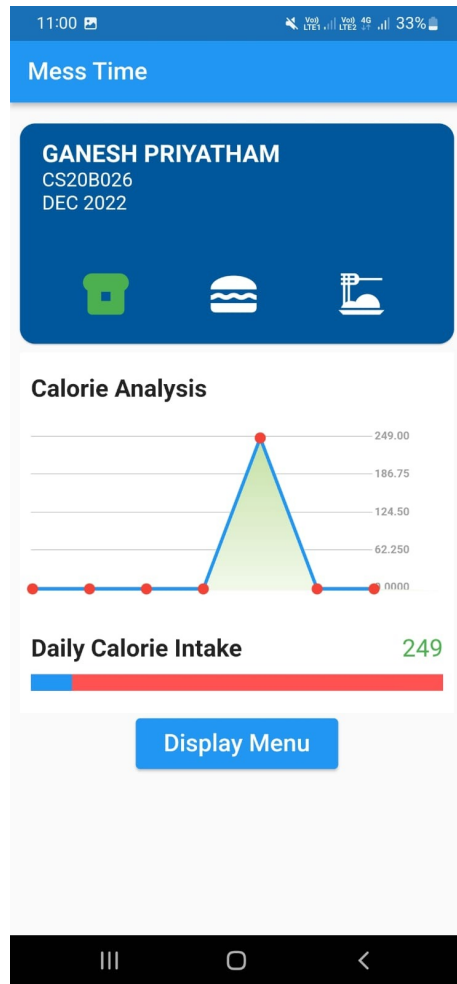
- a) Plots the graph of calorie intake for a week
- b) Daily progress will be shown in a progress bar

Display Menu:

- a) A button which on click will display the menu



(Note the calorie graph is initially set to zero by default for every user and gets updated as the user fills in the information)

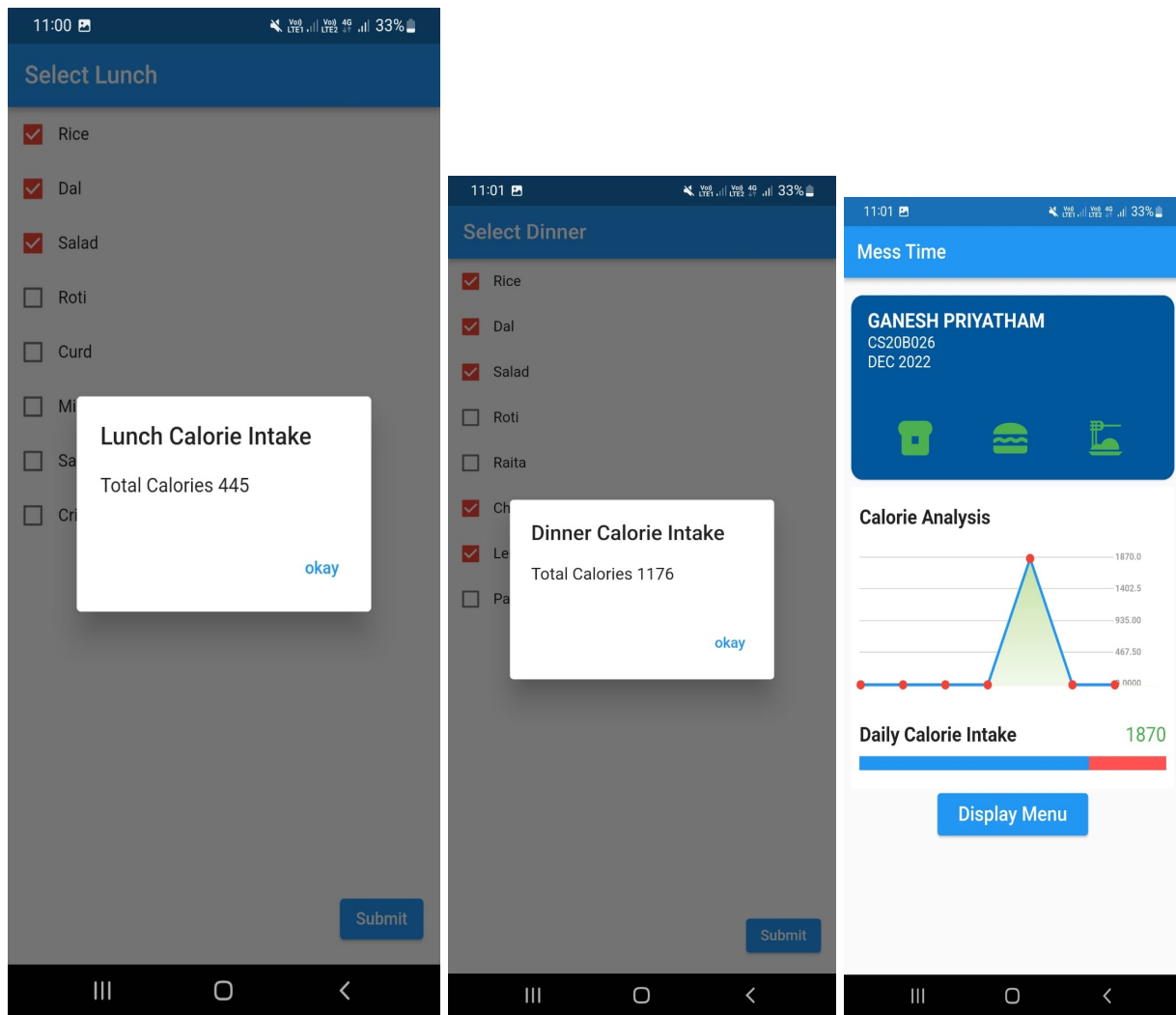


(On entering the breakfast details, the icon changes color, and the graph and progress bar gets updated. Please observe the time in the screenshot as well)

2)Calorie Calculation:

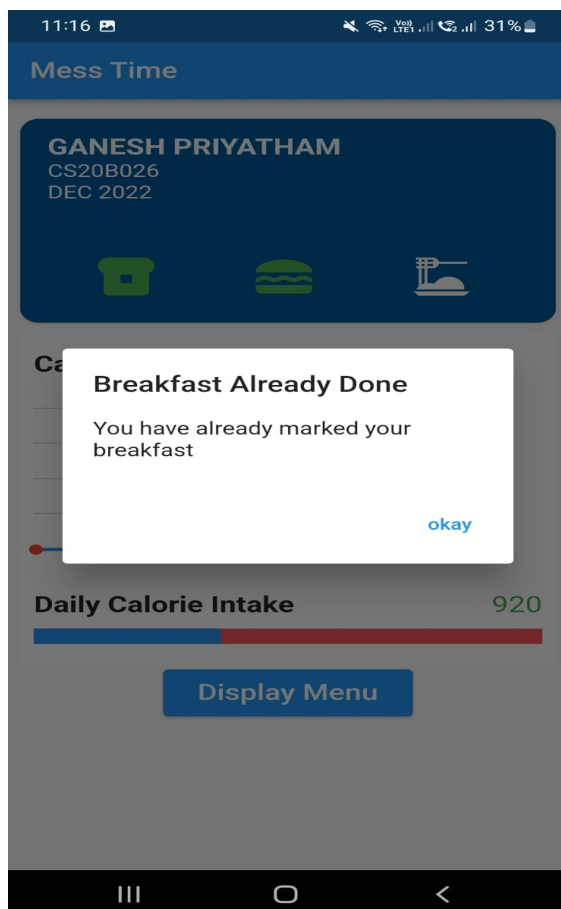
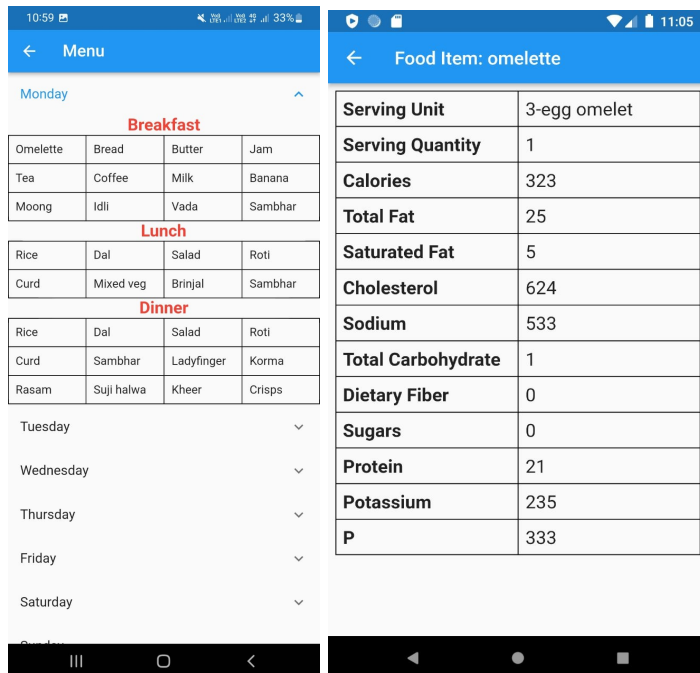
- On clicking the icons in the mess card based on the day, the list of food items will be displayed along with checkboxes.
- Once entering this page user can fill in the checkboxes of the food items he consumed
- If he hasn't consumed anything, he can click the submit button
- The user can't go back until he fills/submits
- The calories will be calculated and displayed

- f) He will now be directed to the homepage where the graph and daily calorie intake will be auto-updated.



3)Display Menu:

- On clicking the button in the homepage, the user will be redirected to a page where the menu of the mess is displayed in a user friendly manner
- The user can click on any item to know the complete calorie breakage of that item.



(Once marked cant be marked again and will reopen the next day)