Google Developer Group
On Campus

TechSprint

Leveraging the power of AI

Team Details

Team name: SmartMess: Crowd-Aware and Data-Driven Mess Management System
Team leader name: Aryan Sisodiya
Team Members: Aryan Sisodiya (B25126), Daksh Rathi (B25132)
Problem Statement: Open Innovation

Google Developer Group
On Campus

Team Details

Team name: SmartMess: Crowd-Aware and Data-Driven Mess Management System

Team leader name: Aryan Sisodiya

Problem Statement: Open Innovation

Team Member Details:

Aryan Sisodiya (Leader):

- B25126
- aryansisodiya091107@gmail.com
- https://github.com/InfinityxR9/

Daksh Rathi:

- B25132
- iamdakshrathi@gmail.com
- https://github.com/dakshrathi-india/

Google Developer Group
On Campus

Brief about your solution and problem statement addressing

SmartMess is an intelligent mess management system designed to improve food quality, reduce wastage, and enhance the student dining experience using real-time data. The primary aim of the project is to analyze crowd levels in the mess by comparing live attendance with available seating capacity and predict the best time for students to have their meals. This helps avoid overcrowding, ensures smoother meal distribution, and prevents sudden high demand that often degrades food quality.

In addition, SmartMess enables students to provide food feedback, helping mess authorities continuously improve meal quality. The provided constructive and meal-specific feedback allows the mess to prepare food more accurately and significantly reduce wastage. Daily menus can be created and displayed digitally for easy access. Overall, SmartMess creates a more efficient, transparent, and sustainable mess ecosystem for both students and managers.

Opportunities

a. How different is it from any of the other existing ideas?

- Our idea of SmartMess is different because it is a Data-Driven Solution, It predicts the crowd for next 15 minutes, using a simple Regression model, thus providing students an optimum time to go to the mess and prevent overcrowding problems.
- SmartMess, is not just a plain review system meant for increasing mess quality, but a comprehensive solution that integrates major aspects to address the problems of students, regarding overcrowding and food quality with a sophisticated feedback system, and a beginner friendly UI.

Opportunities

b. How will it be able to solve the problem?

- SmartMess solves the problem of overcrowding in a very unique way. It does so by collecting the data for the number of students going inside the mess in a slot of 15 minutes, and using the Machine Learning algorithm for that specific mess and slot, it trains the model using that data.
- It makes use of this trained model to predict the crowd levels for next 15 minutes slot, thus effectively showing the “optimum” timing for mess-visit, and reducing the chances of overcrowding at mess.

List of features offered by the solution

1. Attendance Marker
    1. Students can mark attendance via a QR Code generated by Manager. This QR code is valid for a period of 15 minutes Slot only and can be generated.
    2. However, in case there is some problem in marking the attendance by QR, a manual Attendance marker is also setup.
    3. This feature ensures **Mess Isolation** (eg, attendance marked by student of alder mess doesn't show up in oak's manager Analytics)
    4. This is achieved by creating an Authentication service, and having different login pages and UI for both students and Managers
    5. This attendance can be used as a counter for students inside the mess (for a period of 15 minutes), and perform Analytics and ML crowd predictions

2. Crowd Prediction

1. The ML model incorporated performs crowd prediction in the interval of every 15 minutes, using the data collected by marking attendance in that period
2. The predicted crowd level is shown in student's UI indicating the "best" time for Mess visit
3. The underlying logic is optimized so that the predictions are available for "mess-open" timings only.
4. Since, currently the model is training on the very small dataset, no historic data and seasonality, the predictions' accuracy can't be checked.
5. However, As attendance data accumulates over days and weeks, the model can be transitioned from short-window retraining to persistent, context-aware learning, enabling more accurate crowd forecasting by capturing temporal patterns and seasonal behavior.

3. Review System
    1. A separate, independent and anonymous review system is integrated
    2. Students can give reviews for that specific slot of mess (Breakfast, Lunch, Dinner), which will be displayed on the Analytics section of Manager's UI
    3. The review will be anonymous and the underlying logic Ensures there is **Mess Isolation**.
4. Menu Creation
    1. The Managers can create the menu for specific slot of the day (Breakfast, Lunch, Dinner).
    2. The Standard Menu of Mess is always displayed, and can be modified in times of Special Dinner.
    3. This menu will be displayed on the student's UI for that mess only, thus ensuring **Mess Isolation**

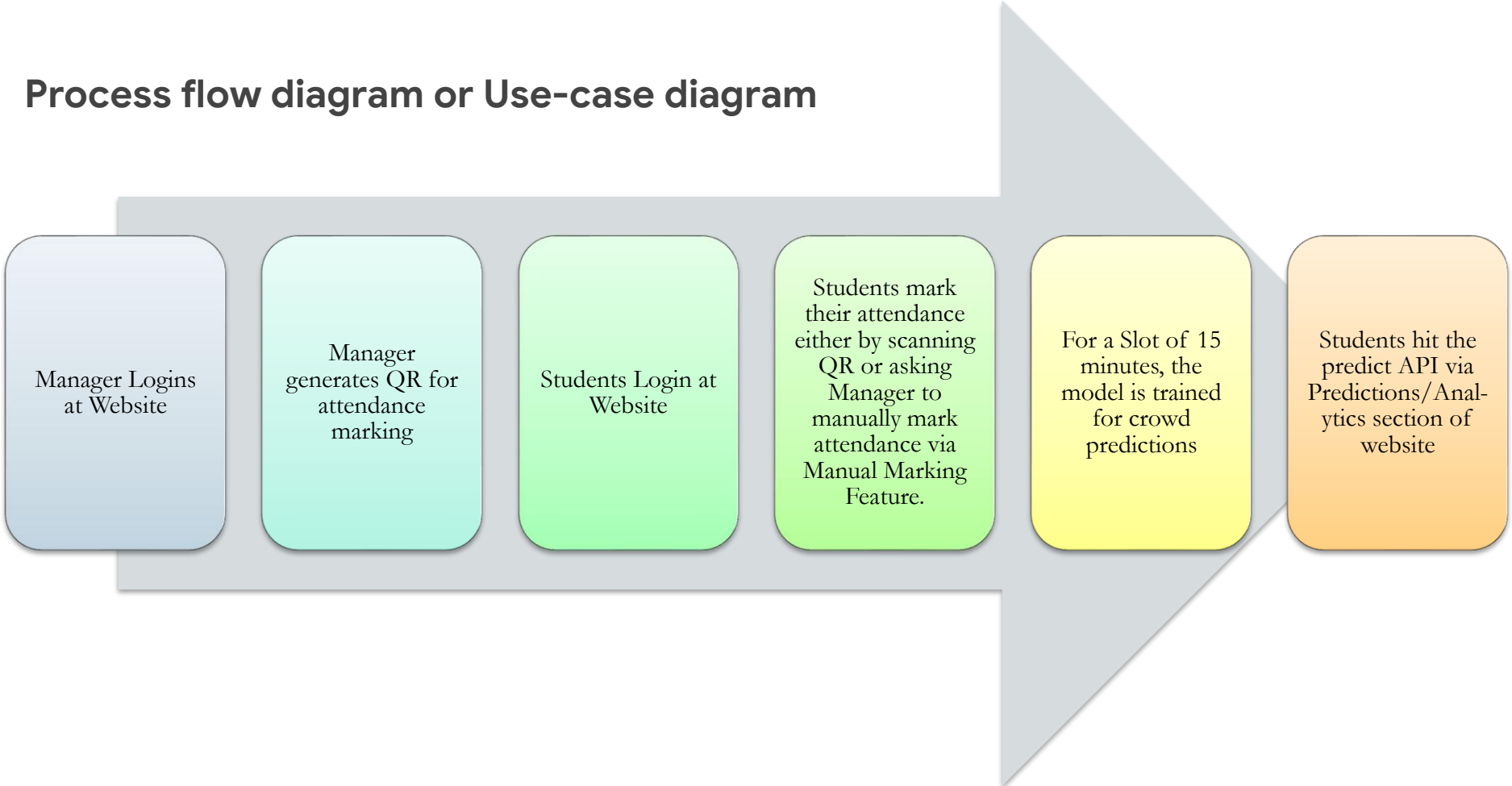
Google Developer Group
On Campus

Google Technologies used in the solution

- Flutter and Dart (For Frontend UI)
- TensorFlow (For ML Crowd Prediction Model)
- Firebase Firestore (For Database)
- Firebase Hosting (For Frontend Hosting)

Google Developer Group
On Campus

Process flow diagram or Use-case diagram



Google Developer Group
On Campus

Architecture diagram of the proposed solution

ML Model

- /train API to train the model with parameter passed for the mess information using the Linear Regression algorithm
- /predict API to predict the crowd level for next 15 minutes slot

FrontEnd Student

- Student Login System by fetching data from Firebase Firestore Database in loginCredentials collection.
- Student Features include marking attendance via QR code, submitting meal review, viewing menu and analytics including Predictions.

FrontEnd Manager

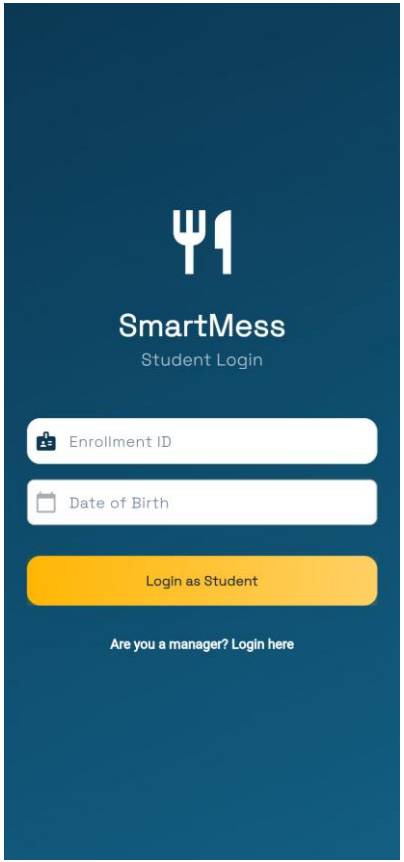
- Manager Login System by fetching email and raw password (currently without hashing for simplicity) from Firebase Firestore Database in the same collection.
- Manager features include QR code generation, manual attendance marking, creation of menus and viewing analytics including predictions.

BackEnd

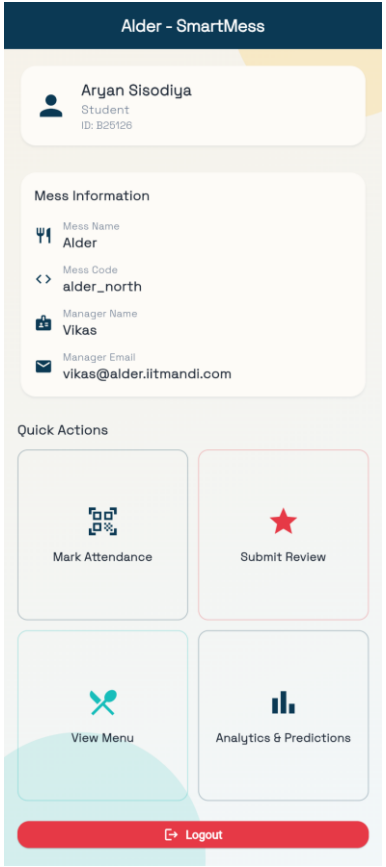
- Runs at the flask Backend server.
- Handles the ML Model APIs including /predict and /train APIs.
- Additionally, Handles the Analytics API for showing Different Analytics.

Google Developer Group
On Campus

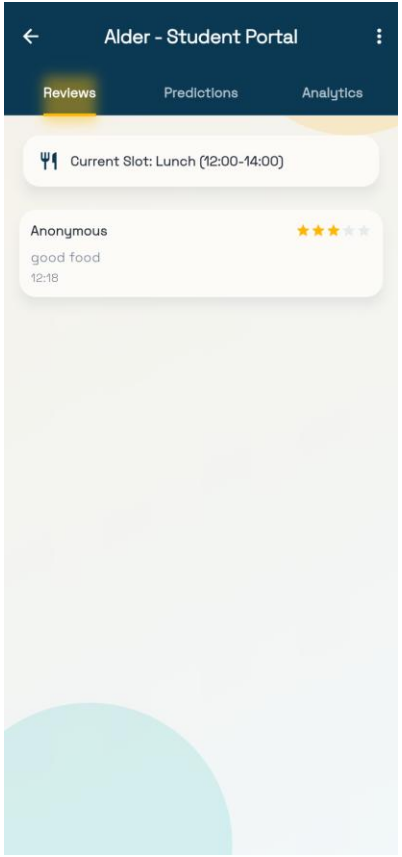
Snapshots of the MVP (Web App)



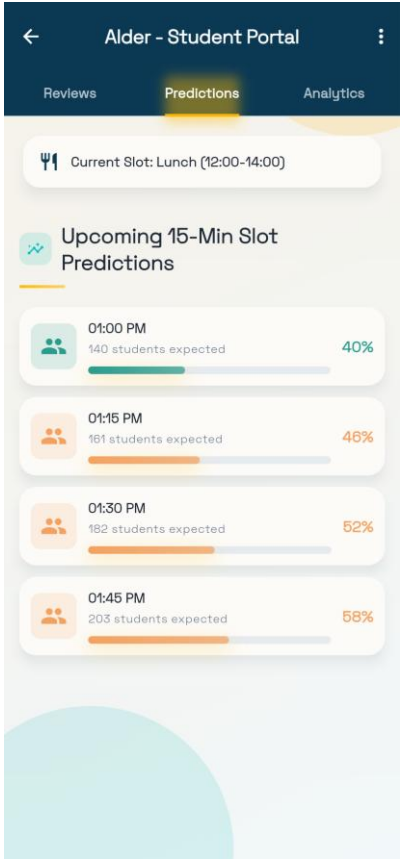
Student Web Login Screen



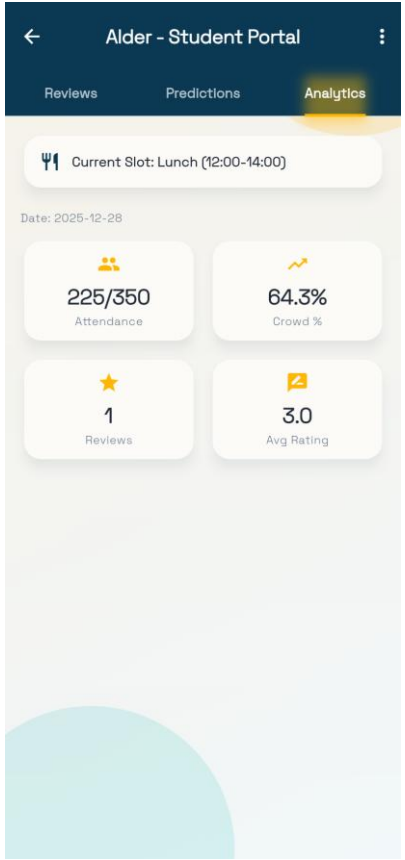
Student Home Screen



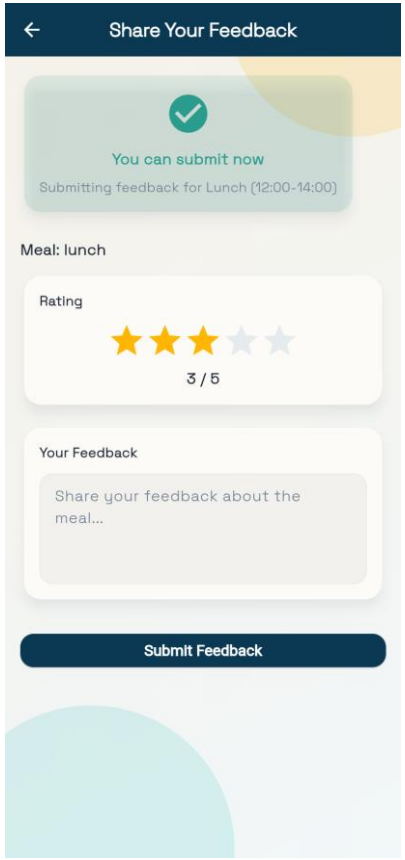
Student Review View Screen



Student Crowd Prediction Screen

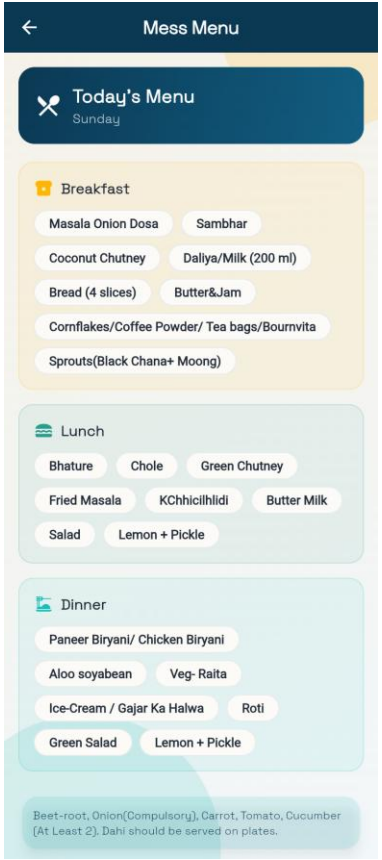


Student Analytics Screen

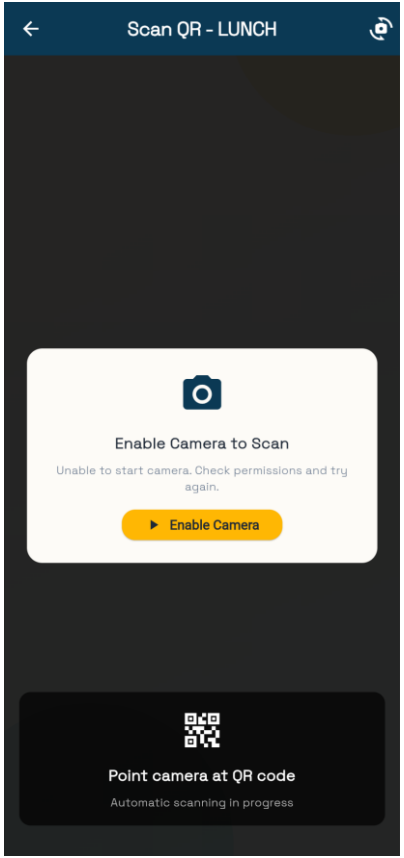


Student Meal Review (Anonymous)

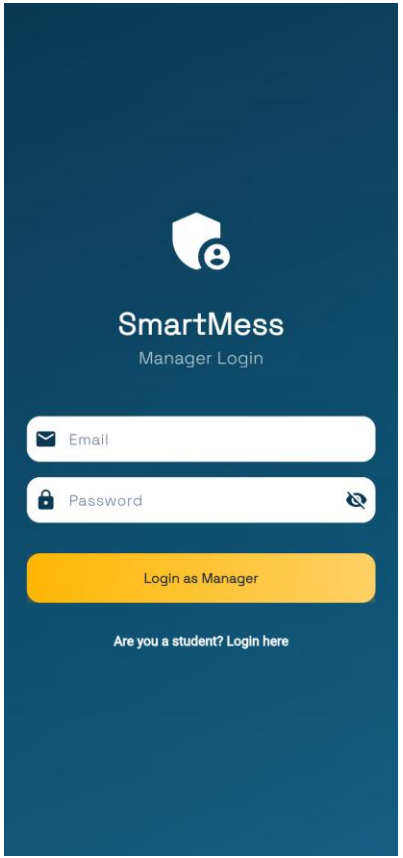
Google Developer Group
On Campus



Mess Menu View Screen

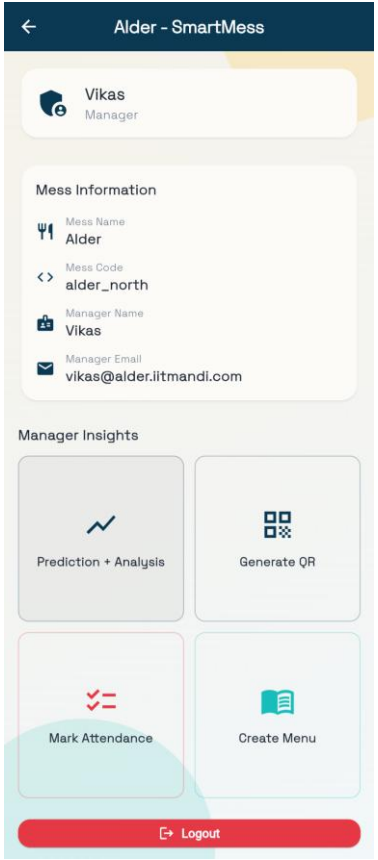


*Attendance Marking via QR Scan
(See #2 of Future Developments)



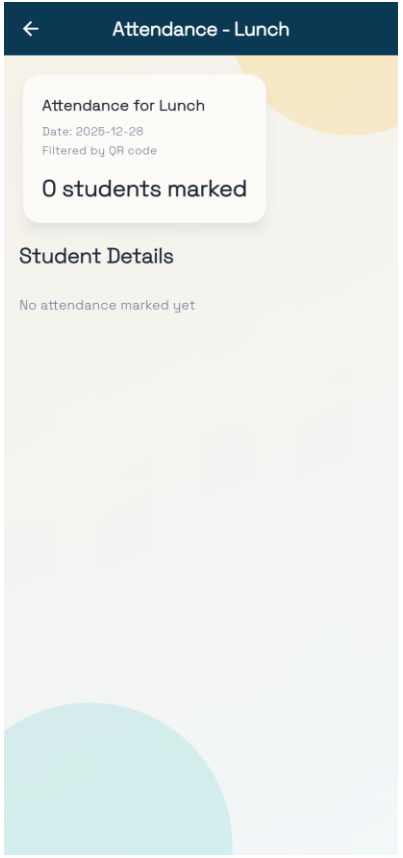
Manager Web Login Screen

Google Developer Group
On Campus

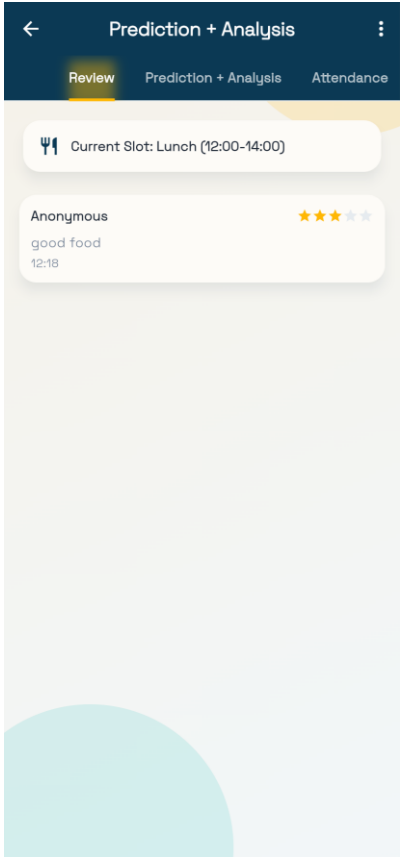


Manager Home Screen

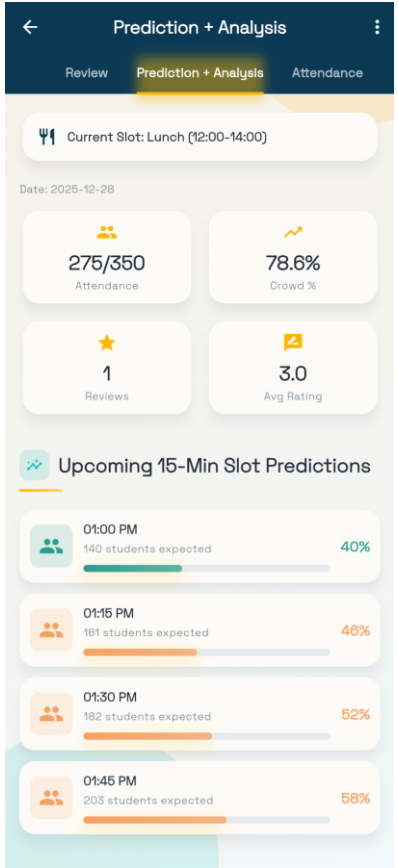
QR Code Generation Screen



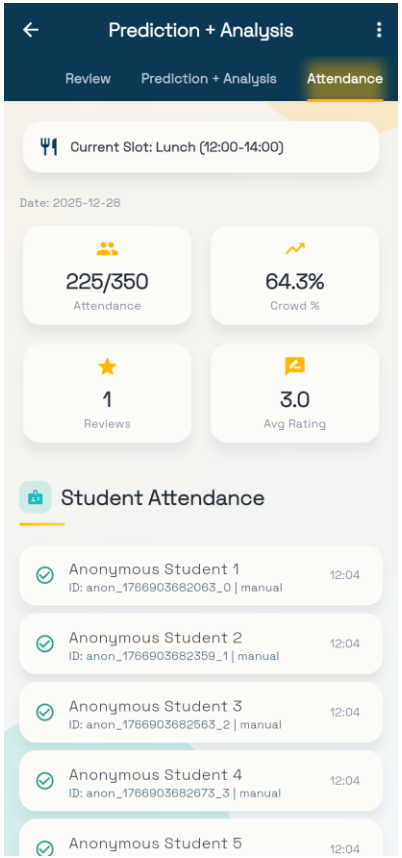
Marked Attendance by QR



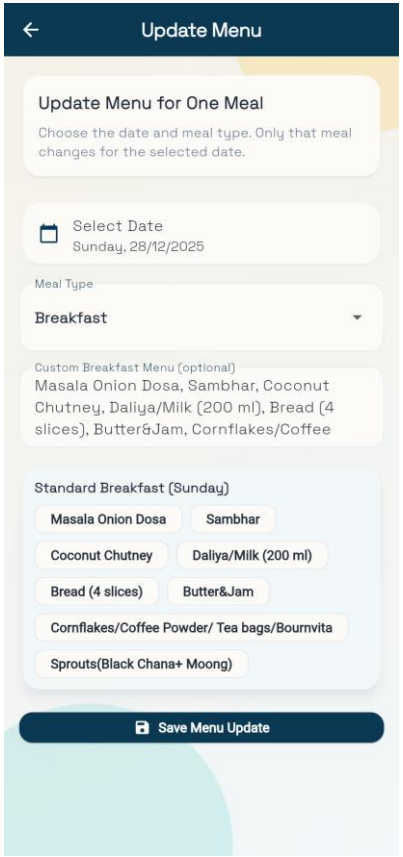
Student Reviews View Screen



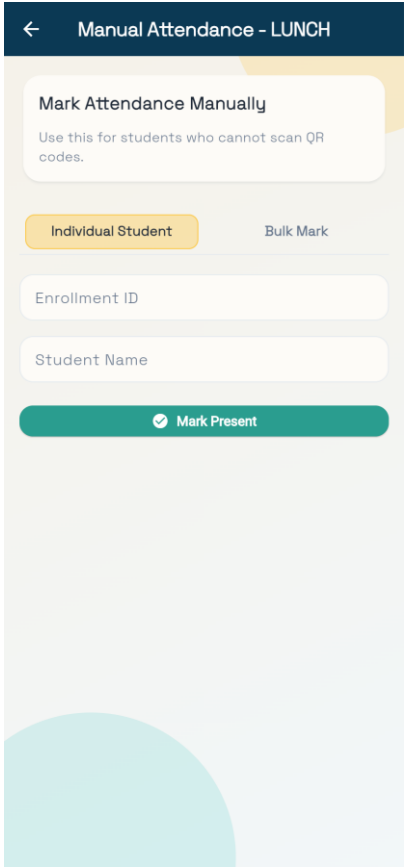
Crowd Predictions Screen



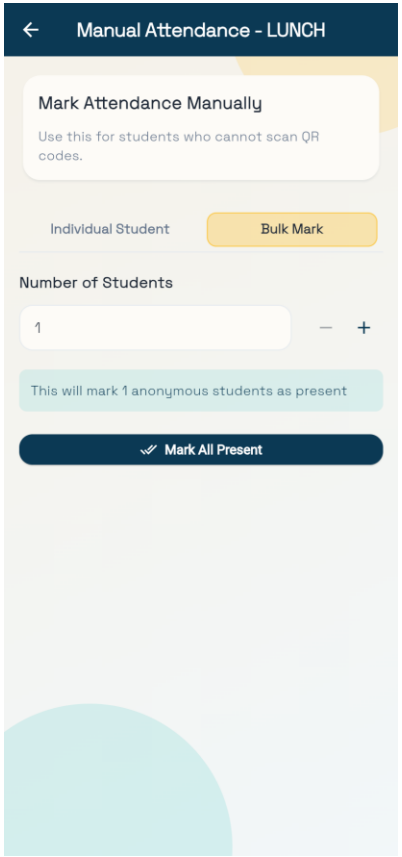
All Marked Attendance



Menu Update Screen



Manual Attendance Marking



Bulk Attendance Marking

Additional Details/Future Developments

1. Authentication Methodology MUST be improved
    1. Currently, For student's authentication, DOB is used as password and Enrollment ID as login IDs, both the data is currently stored as hardcoded directly in the database without hashing for simplicity and testing purposes.
    2. Same goes for Manager's login too, currently, for testing purposes, a dummy email and dummy password is hardcoded directly in the database
    3. In the real, functioning model of the Project, there will be a dedicated Admin Page, where services like adding Manager's data will be there, and through that mechanism, the passwords will be stored perfectly via hashing.

2. Inclusion of Camera-based QR Scanning:
    1. While initially planned, Camera-based QR scanning is intentionally excluded from the MVP due to current limitations of Flutter Web's camera APIs and unstable cross-browser behavior.
    2. The MVP focuses on validating core attendance logic, while camera scanning is planned as a future enhancement for mobile platforms or dedicated web implementations.
3. Google Maps API integration
    1. Google Maps API can be integration to show the allotted mess's location in a web-integrated Map, along with Building details, like A19 3rd Floor for Alder Mess.
    2. This will increase the ease of locating mess for new students
4. History section for manager's UI to see past attendance, crowd levels and reviews.
5. History section for student's UI to see their past attendance and submitted reviews.
6. Other Minor fixes:
    1. Adding service for "Forgot Password" in the Manager's Login Page
    2. Showing Mess Menu for whole week using a date-picker, instead of showing it just for the current day.

Google Developer Group
On Campus

IMPORTANT!

1. The web services will be active during mess-timings only, so plan accordingly!
2. To view actual predictions, first make sure there are some attendance in the last 15 minutes slot and have patient while they are loading!
3. For the purpose of testing and simplicity, there're a few dummy login credentials for student and managers, this data can be used to test different features offered by the SmartMess Web Application.

```
enrollmentId: "B25132"
messCode: "oak_north"
messId: "oak"
name: "Daksh Rathi"
password: "2006-09-29"
type: "student"
```

```
enrollmentId: "B25126"
messCode: "alder_north"
messId: "alder"
name: "Aryan Sisodiya"
password: "2007-11-09"
type: "student"
```

```
email: "vikas@alder.iitmandi.com"
messCode: "alder_north"
messId: "alder"
name: "Vikas"
password: "vikas123"
type: "manager"
```

```
email: "vansh@oak.iitmandi.com"
messCode: "oak_north"
messId: "oak"
name: "Vansh Bhai"
password: "vansh123"
type: "manager"
```

Provide links to your:

1. **GitHub Public Repository:** https://github.com/InfinityxR9/SmartMess
2. **Demo Video Link (7 Minutes):** https://drive.google.com/file/d/1MVgDEax4aREU27eQ6RrTETFEcv5-q-hh/view?usp=sharing
3. **MVP Link:** https://smartmess-project.web.app/

Google Developer Group
On Campus

TechSprint

Leveraging the power of AI

Thank you!