

# **Sprint 2 Documentation**

## **Sprint 2 Design Document**

**Project Team: Institute Food Review System (IFRS)**

Members:

- Mayank Mittal
- Newton Anandagopal Mukhopadhyay
- E J V S Sathvik Goud
- Mohammad Sameer
- Ameen N
- Deepankar Vijay Dongare
- Divyanshu Mohan Nagrale

### **Deployment Environment:**

The project is not currently deployed on the internet. Our plan is to deploy this project on Heroku around the end of November.

### **Functional Requirements:**

- User Login and verification
  - Verify admin
  - Allow login only for institute mail id
- Form for users to submit reviews
  - Ability to comment and like/dislike
  - Data will not stored
  - An alert to show like/dislike
- Admin Functionalities
  - View user and mess menu info
- Dynamic Nav bar for all the webpages
  - Home, table and logout components for admin
  - Home, mess menu, feedback and logout for logged in users
  - For guest user, only home and login are accessible
- Complete landing page

- A homepage with attractive css with images of IIT Dharwad mess.

## Database Design:

user		menu	
Objectld	int	Objectld	int
Name	varchar	fooditem	varchar
email	varchar	image	varchar
password	varchar	day	varchar
image	varchar	timing	varchar
gender	varchar		
isadmin	boolean		
iscustomer	boolean		
isrestricted	boolean		

## Project Direction:

The work was divided among the team members according to the roles they were comfortable with. The sprint velocity of our team was slow in the beginning of the sprint period but gradually gained pace and we were able to accomplish most of our goals for this sprint within the deadline. One possible reason as to why the sprint velocity was lower in the beginning could be that we dedicated most of our time during the initial days for learning the technologies required. We hope that the sprint velocity would align more with the ideal velocity for the subsequent sprints.

**Completed:**

- Google authentication based login was implemented for the purpose of user authentication.
- The different pages of our web application were designed like review page, admin page, 404 page, home page and menu pages, and implemented with emphasis on being simple but effective and attractive.
- Review form was implemented on the frontend to collect reviews and the corresponding user information.
- Backend and database design was done for users and menu items.

**To-Do:**

- No backend has been implemented yet for storing the reviews collected using the review page.

**Instructions to Start the website:**

- The website was written using Node.js, react.js, Express.js, Bootstrap and MongoDB. Note that internet connection is a must to run the website.
- The zip file has two folders, backend and frontend.
- To start the backend, navigate to the backend folder and run command 'npm install nodemon --save-dev' to install nodemon globally, so that the server automatically refreshes when any changes occur in the .js file. If this doesn't work use 'node app.js' command to start the backend at localhost:8000.
- To start the frontend, navigate to frontend folder (at this point use command 'npm i' to install all the packages available in package.json)
- Then navigate to the src folder, at this point run command 'npm react-google-login', to install google authentication. Then run 'npm start' command to start the frontend.
- The website will automatically start in your default browser at localhost:3000.

## **Important Files:**

- Home Page: The home page uses Home.js, Hero.js, MainHome.js, Video.js, Footer.js and Navbar.js. These are all various components of the homepage.
- Review Page: This page uses Navbar.js, Form.js and Footer.js.
- Mess Menu Page: This page uses Navbar.js, MessMenu.js, MenuCard.js and Footer.js.
- The login and logout buttons are included in the navbar.
- Admin: To view the tables, admin uses MenuTable.js, Usercard.js and UserTable.js files.

## **Note:**

- By default when a user signs up to the website, they are considered as a user. To get admin privileges, changes have to be done in the database manually. To do so, please contact the team members.
- To get access to the cloud MongoDB database, please contact the team members.