



**MANIPAL**  
ACADEMY of HIGHER EDUCATION

*(Deemed to be University under Section 3 of the UGC Act, 1956)*

AT Lab Report  
B.Tech IT 6<sup>th</sup> Semester

## **Mess Management App - Don't Starve**

*Submitted by*

Siddhant Dorman  
Nikesh Kumar  
Anjali Singh

180911142  
180911202  
180911274

# Contents

<b>Abstract</b>	<b>i</b>
<b>List of Abbreviations</b>	<b>i</b>
<b>Introduction</b>	<b>1</b>
<b>Language Description</b>	<b>2</b>
<b>Database Description</b>	<b>2</b>
Firebase . . . . .	2
Console . . . . .	3
<b>Project Description</b>	<b>4</b>
<b>Project Functions</b>	<b>4</b>
Verify FC Users . . . . .	4
Geofencing . . . . .	4
Daily Menu . . . . .	4
NFC Food Token . . . . .	4
<b>Designs</b>	<b>5</b>
<b>UI Screenshots</b>	<b>7</b>
<b>Results</b>	<b>9</b>
<b>Conclusion</b>	<b>9</b>
<b>Future Work</b>	<b>9</b>
<b>References</b>	<b>9</b>

# Abstract

This project aims to create an efficient mess management application that addresses various problems as discussed ahead.

Food tokens in FC2 work on the principle of Radio-Frequency Identification (RFID), which has several downfalls, including loss and theft of mess cards. To curb this problem, the app will use an inbuilt Near Field Communication (NFC) to emit the identification for the user. Thus, it will render physical mess cards obsolete, replacing them with portable ones.

Another common issue faced frequently is excessive crowding, especially during the time of the pandemic. With the help of geofencing, users will observe the real-time crowd and choose a time best suitable for them.

Furthermore, the app will have other functionalities like accessing the daily mess menu.

## List of Abbreviations

### Acronyms

**NFC** Near Field Communication.

**OCR** Optical Character Recognition.

**RFID** Radio-Frequency Identification.

### Description

**geofence** a virtual boundary triggered when an entity enters/leaves.

**NFC** is a communication protocol for communication between two electronic devices over a distance of *4cm*.

**RFID** is wireless communication via tags and readers like *NFC tags*.

# Introduction

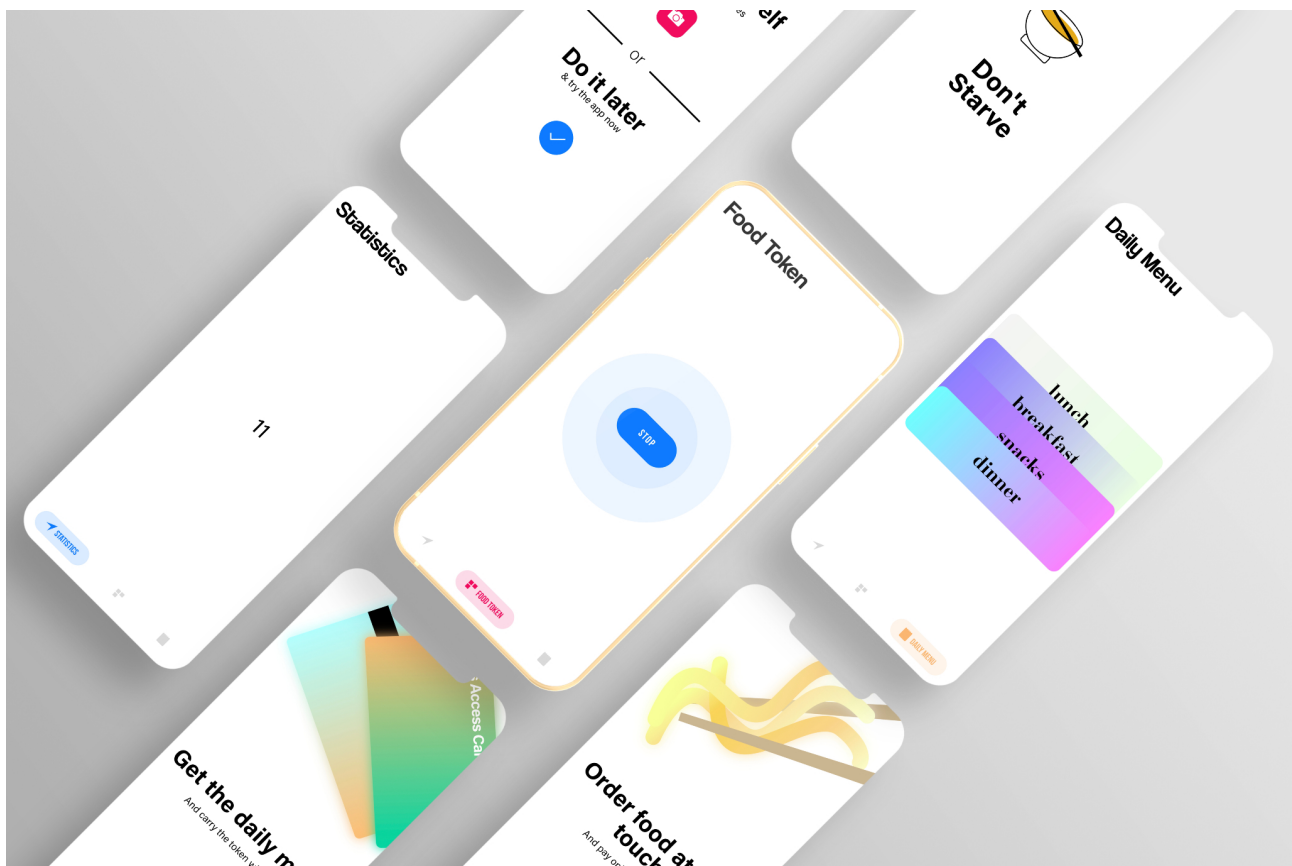
COVID-19, overcrowding, and obsolete technology are the reasons for the existence of this project. The idea behind the project is to deal with the shortcomings as listed but on a smaller level, i.e., in food courts/messes. The Android application, 'Don't Starve' aims to provide a smooth and effortless experience coupled with some guise of safety from COVID-19 in food courts to the user.

This application will allow users to register and get verified (more information under Product Functions) and access mess resources with ease. There is a provision of an 'editable' (by the admin) daily menu. This is to provide the students with a list of delicacies that will be served on that day. Furthermore, this will reduce the usage of paper as printing menus daily would be unnecessary.

Mess cards are subject to theft or loss. Furthermore, carrying them with oneself is also a task sometimes. One might forget to take their cards with them during meals. Since everyone brings their phone everywhere with them, the ideal solution would be to amalgamate the cards, i.e., the identifier, with phones. With technology like NFC, mess cards will become obsolete, and the process of procuring food tokens will be trouble-free.

Food courts are incredibly crowded at certain times. Given the current scenario and how critical social distancing is in today's world, the technology of Geofencing comes to the rescue. This feature will provide real-time data about the number of people in a particular geographical area, i.e., the food courts. Students can refer to the statistics and decide at what time they'd like to have their meals.

The project will be able to provide a seamless experience to the users during the ongoing pandemic. It will also have a social and environmental impact simultaneously. The app helps eliminate the use of plastic for mess cards and provides a safe environment for students to have their meals without worrying about the crowds, thanks to geofencing.



# Language Description

Most of the project has primarily been coded in *Java* in the IDE provided by Android Studio. Our project comprises two parts: the application & the Console. The back-end is hosted on Firebase, which has been connected to both entities of our project. Development related to the mobile application has been done entirely on *Java* with XML Designs, fonts and static images to serve the application. The Console have been made using *React JS*, a JavaScript framework that has been hosted on GitHub pages and connects users to perform back-end changes without needing to login to Firebase.



Figure 1: Language distribution of Application

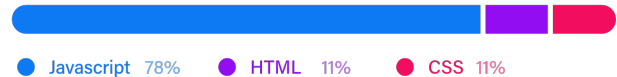


Figure 2: Language distribution of Web App

## Database Description

### Firebase

The backend has been hosted on Firebase and makes use of Firebase Realtime Database & Firebase Storage. Firebase Realtime Database is a real-time NoSQL Cloud database that stores data as a JSON object, and ours comprises three keys to handle crowd data, menu data, and registered users. For simplicity of usage, the login system based on username and password has been scrapped to support one-time identification based on the registration number, name, and email id. When a user enters their data, they will be asked to verify themselves. This data will be pushed to the database. Geofencing connects to the crowd key to fetch real-time crowd numbers and provide them to the user. The console retrieves the menu data and allows the admin to make changes and push it back to the database, fetched by the user whenever they open the app. Firebase Storage is cloud storage like AWS S3 Bucket, where objects/blob objects can be stored/retrieved. Here we primarily store the photos submitted by the user for verification which the console can retrieve to verify the user.

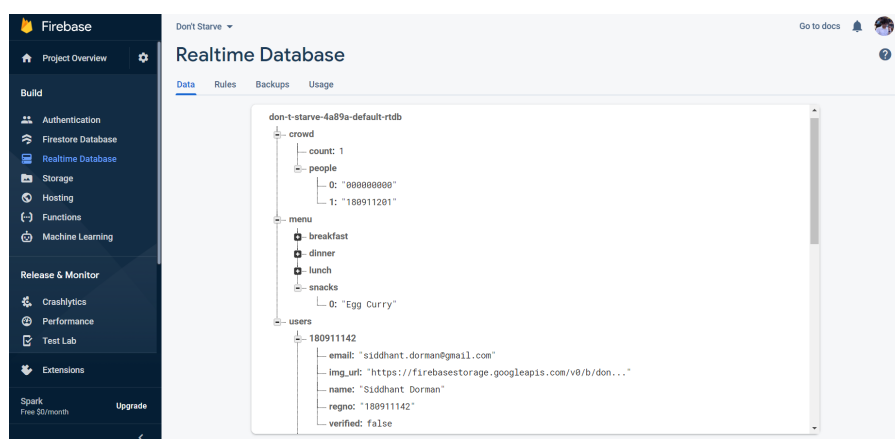


Figure 3: Firebase Realtime Database

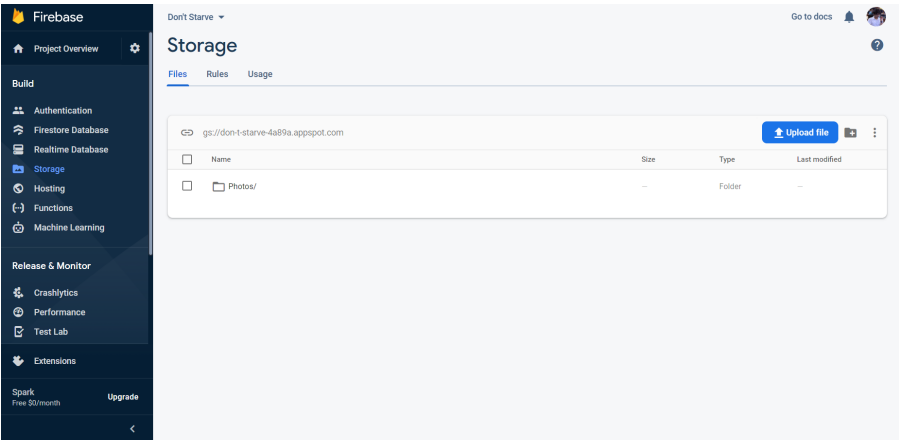


Figure 4: Firebase Storage

## Console

The console handles the admin side and while it is not the backend, it can be termed as an extension of Firebase functionality. It enables the admin to approve users and to edit the daily menu which reflects changes on the Firebase side. It has been designed on React JS, a framework of Javascript which has been hosted on GitHub Pages of the same repository.

Verification ☐ Menu

Approve	Name	Email	Reg no	Verification image
<input type="checkbox"/>	sdsdsadsad	sadsad@sdfhasd.com	123456789	<a href="#">One moment</a>
<input type="checkbox"/>	noiki	noil@esdia.com	129022493	<a href="#">One moment</a>
<input type="checkbox"/>	Siddhant Dorman	siddhant.dorman@gmail.com	180911142	<a href="#">Click here</a>
<input type="checkbox"/>	nike	nike@gmail.com	180911167	<a href="#">Click here</a>
<input type="checkbox"/>	sup boi	supboi@hehe.com	180911201	<a href="#">Click here</a>
<input type="checkbox"/>	Siddhant Dorman	siddhant.dorman@gmail.com	333333333	<a href="#">One moment</a>
<input type="checkbox"/>	Sid	sidd@gmail.com	339199919	<a href="#">One moment</a>
<input type="checkbox"/>	Siddhant Dorman	siddhant.dorman@gmail.com	377377888	<a href="#">One moment</a>
<input type="checkbox"/>	Siddhant Dorman	siddhant.dorman@gmail.com	383838388	<a href="#">One moment</a>
<input type="checkbox"/>	Siddhant Dorman	siddhant.dorman@gmail.com	383883838	<a href="#">One moment</a>

Figure 5: Console - Verification Tab

Verification ☒ Menu

Timing	Menu
Breakfast	Vada,Sambar,Eggs/Fruits,Bread and Butter/Jam,Tea/Coffee/Milk,Corn Flakes
Lunch	Sandwich,Ketchup,Maggi,Watermelon Juice,Tea/Coffee
Snacks	Butter Chicken,Paneer Jalfrezi,Roti,Dal,Rice
Dinner	Egg Curry
<div>SUBMIT</div>	

Figure 6: Console - Menu Tab

# **Project Description**

The current pandemic has been unprecedented in its reach and depth of impact on the functioning of modern society. Social distancing is more important now than ever. We aim to provide an optimized solution to an otherwise unavoidable aspect of daily college life and some semblance of safety for the users. We aim to create an application providing real-time geofencing data of the food court, with contactless meal token and menu features for usage by the students

## **Product Functions**

### **Verify FC Users**

This app makes use of inbuilt Optical Character Recognition (OCR) to identify the student from their identity card issued by MAHE. It will then compare it with the registration number and send it to the backend for storage. The user can also be verified via console in case of failure to identify, which serves as a fallback to the OCR.

### **Geofencing**

Mess crowdings are a big issue which is why to tackle this we make use of android's Geofencing. This feature makes use of background location access to detect any entry/exit in the area of geofence to generate the crowd data. When triggered, it makes changes to the database which is then reflected in the application real-time.

### **Daily Menu**

One of the primary issues faced by the students is that they have no idea of the Daily Menu of the mess. This feature will allow the user to see the daily menu that is set by the admin from the Console.

### **NFC Food Token**

Mess card are prone to physical damage, theft which can be tackled with the usage of virtual tokens. Currently this is a prototype to display how NFC could be implemented as a method of Food Token in mess to validate user data with the machine.

# Designs

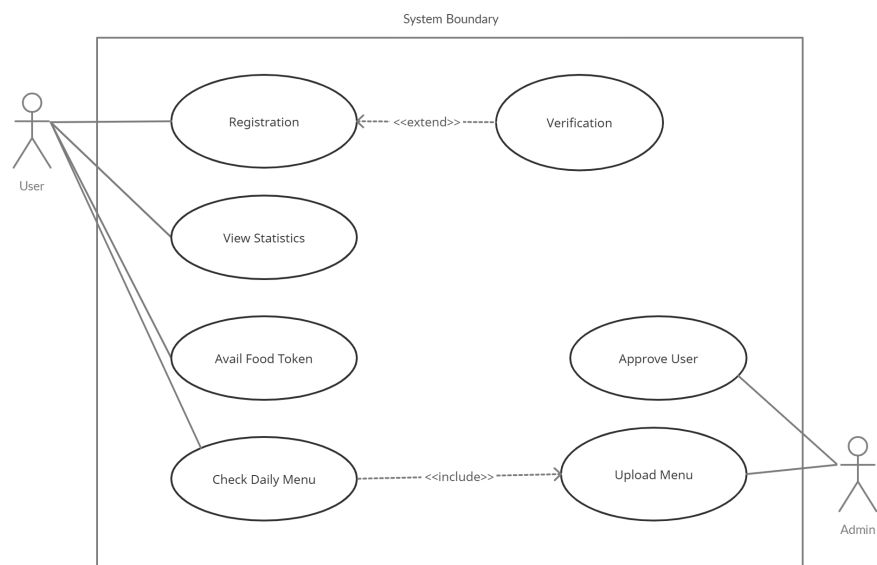


Figure 7: Use Case Diagram



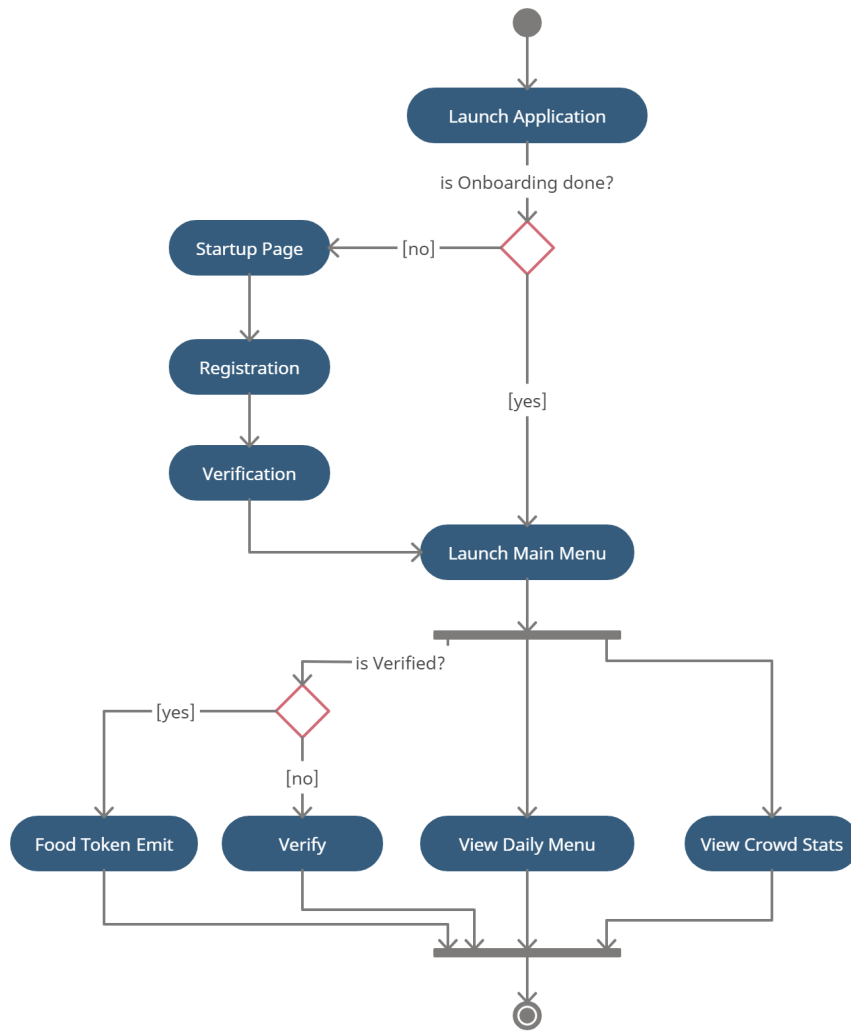


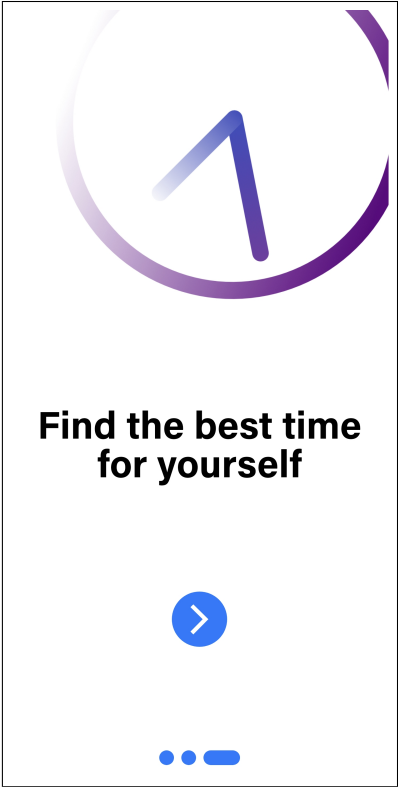
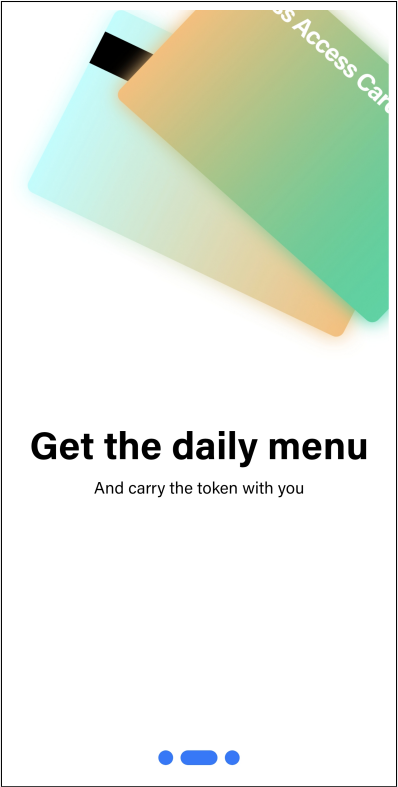
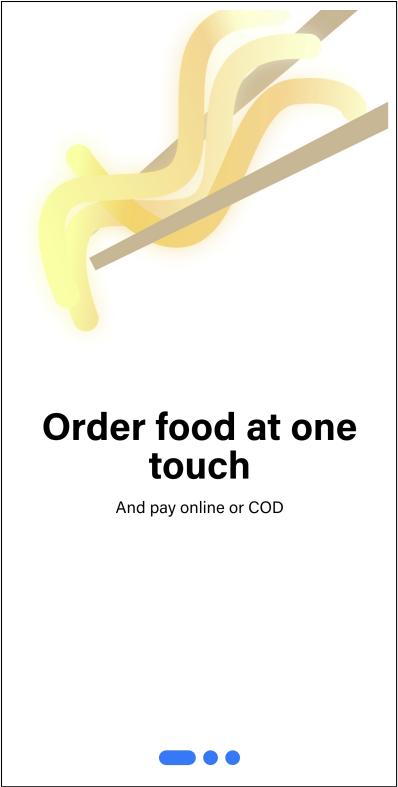
Figure 8: Activity Diagram

# UI Screenshots

Main Splash Screen




Onboarding Screens



## Registration & Verification Screens


### Get Started

by registering below

### Verify yourself


to access all features

\_\_\_\_\_ or \_\_\_\_\_

### Do it later


& try the app now




## Main Screen


### Statistics

11



### Food Token






### Daily Menu

lunch

breakfast

snacks

dinner



# Results

Thus we have implemented the following **App Demo** & **Repo** <sup>1</sup>

We have managed to achieve the following objectives:

1. A prototype of NFC Food Token to show the use case of Virtual Token over an application. Due to limitation of hardware, complete food token emulation could not be achieved;
2. Geofencing to gather crowd statistics
3. A pretty and minimal UI for faster usage
4. Verification using AI/ML

## Conclusion

The application proves itself useful especially at the time of the pandemic, due to crowd control and virtual token system which help one to adhere to pandemic rules and also save your time and as the saying goes, time is money but this time you own it. This app is an environment-friendly initiative as the virtual token allows us to cut down on plastic RFID cards and any loss of it as well. Moreover, the daily menu saves paper spent on printing the menu. These benefits are not only eco-friendly but also very economical as well. All these aside, the app features a minimal and neat UI that is easily navigable.

## Future Work

Possible extensions to the app are feasible

- Predict food menu preference & crowding correlation to improve efficiency of the mess
- Using crowd data and dwell timings to provide weekly crowding statistics
- Implementation of Host-Card Emulation for better facilitation of food tokens
- An ordering service feature
- UPI Payment Integration for buying food tokens temporarily or for ordering
- Using ML for recognising user from user's picture in Identity Card

## References

- [1] <https://square.github.io/picasso/>
- [2] <https://github.com/tommybuonomo/dotsindicator>
- [3] <https://github.com/akshay2211/BubbleTabBar>
- [4] <https://github.com/loopeer/CardStackView>
- [5] <https://firebase.google.com/>

---

<sup>1</sup>other functional components will be explained in PPT