



islington college  
(इस्लिङ्टन कलेज)

## **CS6P05 - Final Year Interim Report**

### **Mobile Application (Food ordering App)**

**2019-20 Autumn**

**Student Name: Aman Maharjan**

**London Met ID: 17031166**

**College ID: NP01CP4A170231**

**Word count: 3896**

**External Supervisor: Ishwor Shrestha**

**Internal Supervisor: Subeksha Shrestha**

**Submission Date: 07<sup>th</sup> January, 2020**

*I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded*

## **Abstract**

This report gives an insight view of the Mobile based Food Ordering Application System. The report is split into various sections (Chapters) i.e. Introduction, Background, developments, Progress, Future work, References and Appendix. It provides an overview to all the readers about what the research is all about

In the first chapter i.e. Introduction includes all the topics with problem statement, objectives and features. In the second chapter (Background) contains the research done for the project. The third chapter (development) includes the work completed till date for the project. In the Progress, the status of the each task is shown and if the task is not completed in time reason and how to recover for the delay is given. In Future work the remaining work of the project is included along with the date it will be completed. Then in reference section all the reference for the project i.e. books, documents, web-sites, etc. are included. And in Appendix section all the additional documents are provided for the proof that all the progress is successfully done.

## Contents

1.	Chapter 1: Introduction.....	1
1.1.	Introduction to Subject Matter.....	1
1.2.	Aims & Objectives .....	2
1.3.	Project Features.....	3
1.4.	Problem scenario .....	4
1.5.	Project as a Solution .....	4
1.6.	Structure of report .....	5
2.	Chapter 2: Background/ literature review-.....	6
2.1.	Similar System Comparison and Analysis.....	7
2.1.1.	Bhoj Deals.....	7
2.1.2.	Foodmandu .....	9
2.2.	Findings of the Research.....	11
2.3.	Resource required.....	11
3.	Chapter 3: Development to date.....	12
3.1.	Methodology .....	12
3.1.1.	Comparison of Methodologies.....	12
3.1.2.	Selected Methodologies.....	14

3.2.	Wireframe .....	15
3.3.	Use-case Diagram.....	19
3.4.	Initial ER-Diagram .....	20
3.5.	Normalization .....	21
3.6.	Final ER-Diagram.....	25
3.7.	Frontend Development.....	26
4.	Chapter 4: Analysis of progress .....	29
4.1.	Assignment of student table .....	29
4.2.	Progress detail .....	31
5.	Chapter 5: Future work.....	32
5.1.	Future work detailed.....	33
6.	Chapter 6: References .....	34
7.	Appendix-A (Gantt chart).....	36
8.	Appendix-B (Survey Result) .....	37
9.	Appendix-C (Meeting Logs) .....	41
10.	Appendix-D .....	53
10.1.	Frontend Code .....	53

## Table of Tables

Table 1: Analysis of progress.....	30
Table 2: Future Work.....	32

## Table of Figures

Figure 1: Bhoj Location Selection.....	7
Figure 2: Bhoj home page.....	7
Figure 3: Bhoj drawer board.....	8
Figure 4: Bhoj food list .....	8
Figure 5: Foodmandu location Selection .....	9
Figure 6: Foodmandu home page .....	9
Figure 7: Foodmandu drawer board.....	10
Figure 8: Foodmandu food items .....	10
Figure 9: Waterfall model.....	12
Figure 10: Agile model.....	13
Figure 11: Prototyping model .....	14
Figure 12: front page wireframe.....	15
Figure 13: Register page wireframe .....	15
Figure 14: Home page wireframe.....	16

Figure 15: Nearby wireframe.....	16
Figure 16: user dashboard wireframe.....	17
Figure 17: Change password wireframe.....	17
Figure 18: De-activate wireframe .....	18
Figure 19: Logout wireframe .....	18
Figure 20: Use-case diagram.....	19
Figure 21: Initial ER-Diagram.....	20
Figure 22: Final ER-Diagram .....	25
Figure 23: Signup page .....	26
Figure 24: Signup page .....	26
Figure 25: Signup validation.....	27
Figure 26: Home page .....	27
Figure 27: Dashboard.....	28
Figure 28: Gantt chart.....	36
Figure 29: Survey 1 .....	37
Figure 30: Survey 2 .....	37
Figure 31: Survey 3 .....	38
Figure 32: Survey 4 .....	38
Figure 33: Survey 5 .....	39

Figure 34: Survey 6 .....	39
Figure 35: Survey 7 .....	40
Figure 36: Meeting log 1 (External) .....	41
Figure 37: Meeting log 1 (Internal) .....	42
Figure 38: Meeting log 2 (External) .....	43
Figure 39: Meeting log 2 (Internal) .....	44
Figure 40: Meeting log 3 (External) .....	45
Figure 41: Meeting log 3 (Internal) .....	46
Figure 42: Meeting log 4 (External) .....	47
Figure 43: Meeting log 4 (Internal) .....	48
Figure 44: Meeting log 5 (External) .....	49
Figure 45: Meeting log 5 (Internal) .....	50
Figure 46: Meeting log 6 (External) .....	51
Figure 47: Meeting log 6 (Internal) .....	52

## 1. Chapter 1: Introduction

The following report gives an insight view of the Food Ordering Application System project. It assists on providing an overview to all the readers about what the research is all about. A brief overview is mentioned in the report along with the problem statement, objectives and features.

### 1.1. Introduction to Subject Matter

The food supply market is € 83 billion worldwide, or 1% of the total food industry and 4% of food sold by restaurants and fast food chains. In most countries, it has already matured, with an expected average annual growth rate of just 3.5 percent over the next five years. (McKinsey & Company, 2019) In today's age of fast food and take-out, many restaurants have the customers who order food from home rather than visiting the restaurant, but the problem was the order was placed over the phone which has many disadvantages like manual listing the order over phone may result in slow response in a customer service. Secondly, due to the oral communication over phone there may be misunderstanding which may lead to confusion and incorrect orders, and there is a lack of visual confirmation that the order was placed correctly.

Food ordering app is a mobile based application that stimulates the customers to place order through the mobile app by finding their favorite or nearest restaurant. For both customer and the restaurant, food ordering app greatly simplifies the ordering process. The app offers a user-friendly, interactive, and up-to-date menu with all the options available for the customers. People can find all the restaurants within one mile radius so that even if they want to visit nearby restaurants they can get all the information about the restaurant and the reviews given by other users for that restaurant. Customers will also be provided with the list of most popular food in their area. So, in this project I am going to develop a Mobile-based Food Ordering System from where people can Order food online from any restaurants.

## 1.2. Aims & Objectives

The main aim of this project is simplify and to increase the efficiency of ordering process for both customer and restaurant, reduce the human error, and provide high quality service to the customers. Customers can also view the product they are ordering which helps customers in visually conforming the order they have placed. The aim of this project are as following:

- Reduce the time-consuming phone orders.
- No more busy phones while other customer is over phone ordering.
- Reduce incorrect order placement.
- Greater customer satisfaction.
- No more long queues while ordering.
- Management of remaining food of the day by offering it in minimum price.

The objective of this project is to develop a mobile application and accomplish the following objectives:

- Make food ordering easy and fast.
- Provide information about nearby restaurants.
- Online payment.
- Food waste management.
- Track the ordered item.
- Recommend customer about the trending food in their area.

### 1.3. Project Features

- **Food Types** – There will be Food Types like fast food, bakeries, and beverage. For example fast food will contain foods like sausages, burgers, french-fries and so on, whereas bakeries contain foods like cakes, pastries, doughnut and many more, and beverages will contain all kind of non-alcoholic drinks.
- **Restaurants Details** – The details of the restaurants will be shown on the app like what type of restaurant is it (i.e. Nepalese, Indian, Korean, etc.) along with the number of the restaurant and their location.
- **Cart System** – In cart system user can put the foods they want to buy in the cart and order them all at once.
- **History** – History will hold all the orders done by the customer through which admin can create report like which restaurant does the specific user order more from which will help in recommending the restaurants for that user.
- **User Roles** – User roles are the types of roles that are to be made in the system, there are three user roles i.e. Admin, Customers and Restaurants.
- **Password Encryption** – The password of the user will be converted into hash value while being saved in the database.
- **Food Waste Management** – Food waste management basically means that during closing time (2-3 hours before closing) the food will be provided in heavy discount.
- **Recommend Foods** – Customers will get recommendation like trending foods in their area, todays special, discount offers and more.

## 1.4. Problem scenario

Nearly every human need has been filled by technology. In today's generation, most people have a busy lifestyle. Time is becoming one of the priceless things. Nowadays people prefer ordering food from home/work rather than visiting the restaurant, but the problem is we have to place our order through the medium of phone call which has many disadvantages like manual listing the order over phone may result in slow response in a customer service. Secondly, due to the oral communication over phone there may be misunderstanding which may lead to confusion and incorrect orders, and there is a lack of visual confirmation that the order was placed correctly. In context of Nepal, We have food delivering app like Bhoj deals, Foodmandu which are great but they do not have system like tracking the food like in which state is our order in. We all know how hard it is to wait for the food we ordered and we even don't know if it is cooked or it is on the way. By developing my proposed food ordering application system, all the complexity of this issue can be easily removed.

## 1.5. Project as a Solution

A mobile application will be developed to tackle the problem mentioned above. By developing food ordering application system, people can order food online which would save them a great amount of time as well as make their task a lot easier. The restaurants will also have benefits in managing their customer, the restaurants will not have to put a separate staff for the phone calls. ([Reference: Appendix-B](#))

## **1.6. Structure of report**

### **Chapter 2 Background/ literature review**

This chapter includes system elaboration, system architecture, system comparison, outcome of the research, and technical aspects of the project

### **Chapter 3 Development to date**

.this chapter includes the work that have been completed till date like wireframes, use case, normalization, ER-D etc.

### **Chapter 4 Analysis of progress**

This chapter includes the status of the task carried out till date, and if the work is not completed according to the Gantt chart then reason for delay is provided along with the plan to complete the delayed work

### **Chapter 5 Future work**

This chapter includes the work to be done in the near future.

### **Chapter 6 References**

It contains the references to web sites, books, articles, etc.

### **Chapter 7 Appendix**

This contains the extra documents for the proof of completed tasks. This section also contains the meeting log of the project as well as survey done for the system.

## **2. Chapter 2: Background/ literature review-**

The mobile app developed in this project is an application for e-commerce. E-commerce (electronic commerce) is the purchasing and sale of goods and services or the transfer of funds or data via an electronic network i.e. internet. In the 1960s, companies began to exchange business documents with other companies using Electronic Data Interchange (EDI). ASC X12 was designed as a universal standard for companies to share documents through electronic networks by the American National Standards Institute in 1979. In the 1980s, the advent of eBay and Amazon revolutionized the e-commerce markets, with the number of individual users exchanging electronic documents with each other. Consumers can now buy infinite quantities of items online from e-trailers, typical brick and mortar stores with e-commerce capabilities and one another. E-commerce history is a history of a new virtual world that develops according to the benefit of the customers. It's a world where everyone builds brick by brick together and lay a strong foundation for future generations (Rouse, 2019).

Online shopping or e-commerce is one the most popular business in Nepal. Bhoj deals, Foodmandu, etc. are one of the finest e-commerce sites of Nepal in fooding sector. They have saved time of so many people and have made their life so convenient as well.

## 2.1. Similar System Comparison and Analysis

### 2.1.1. Bhoj Deals

BHOJ is a one stop mobile app for all the foodies in Nepal, where they can find their favorite restaurants, get food delivered to their home or office, get great deals when they dine in restaurants, read and post restaurant reviews, explore restaurant menus, earn credits to their Bhoj Wallet and much more.

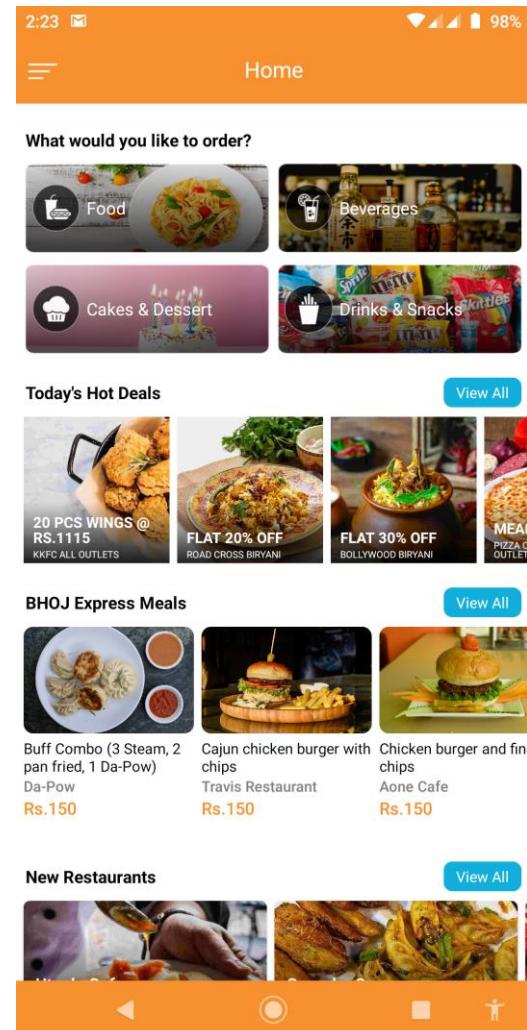
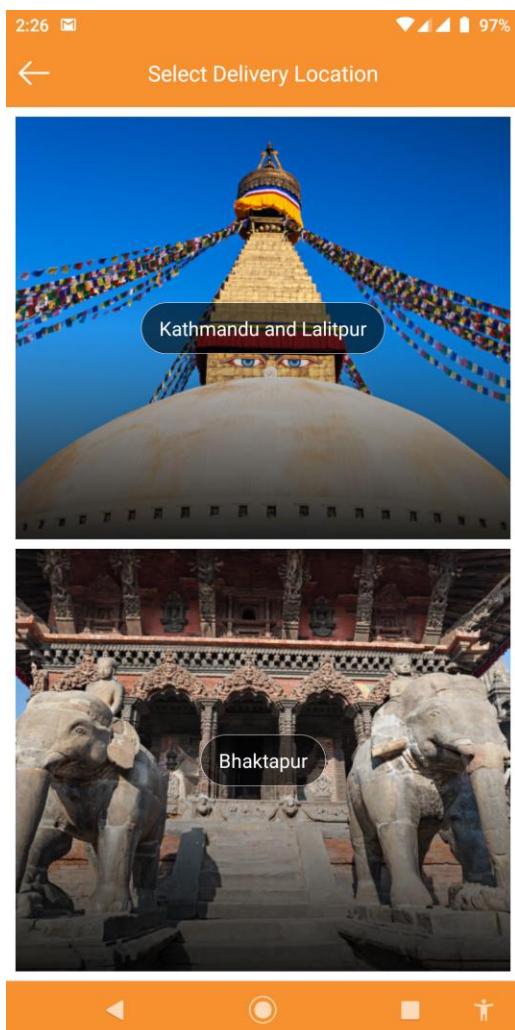


Figure 1: Bhoj Location Selection

Figure 2: Bhoj home page

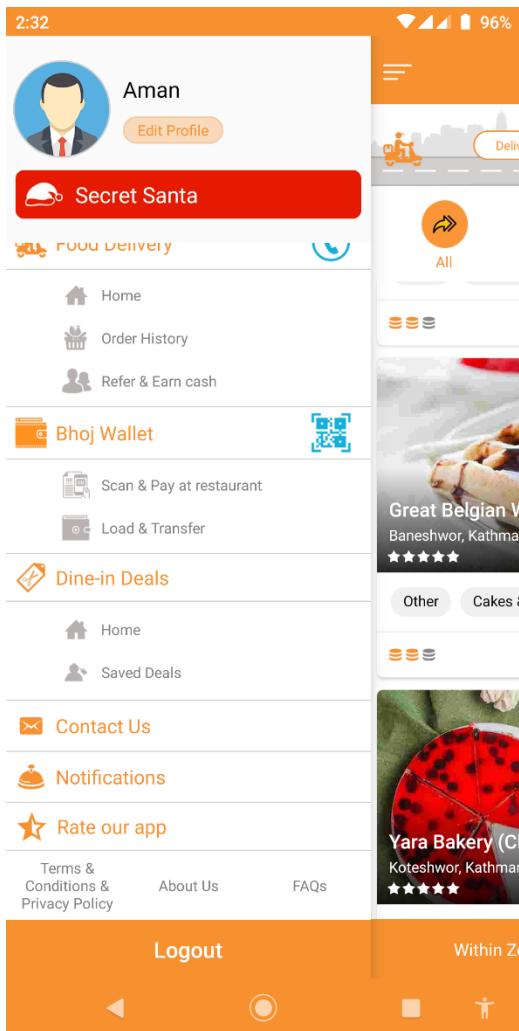


Figure 3: Bhoj drawer board

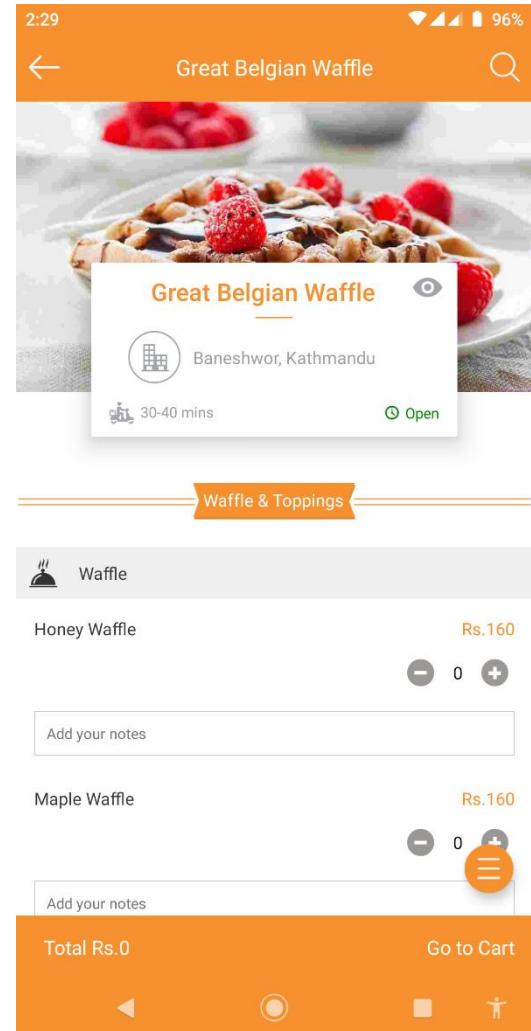


Figure 4: Bhoj food list

## 2.1.2. Foodmandu

Foodmandu is the first company in Nepal that delivers food from hundreds of popular restaurants.

As a pioneer food delivery service provider, we are making life easier through online ordering.

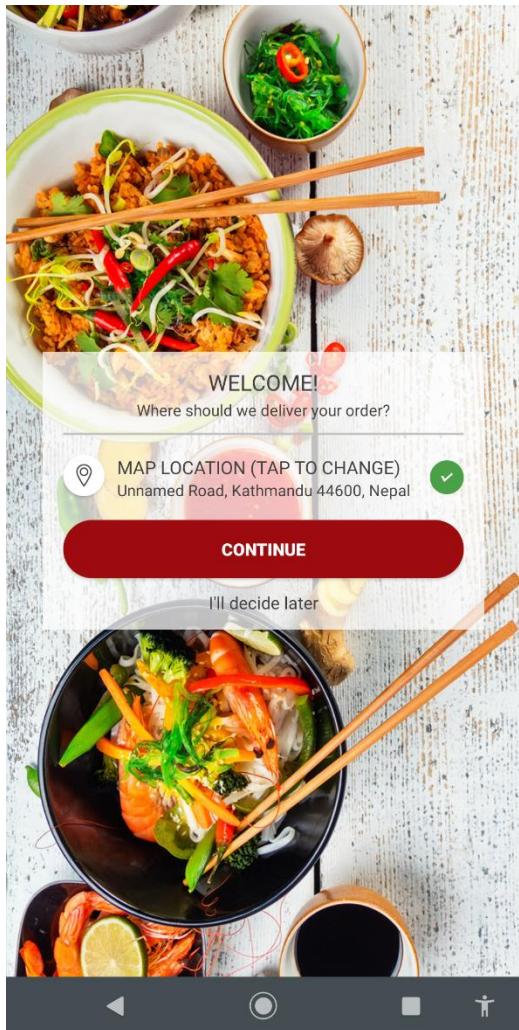


Figure 5: Foodmandu location Selection

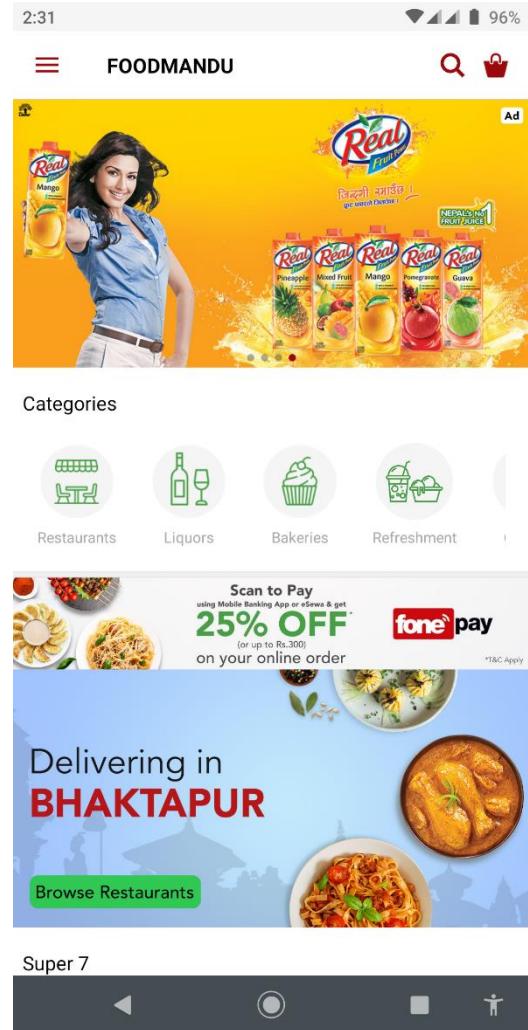


Figure 6: Foodmandu home page

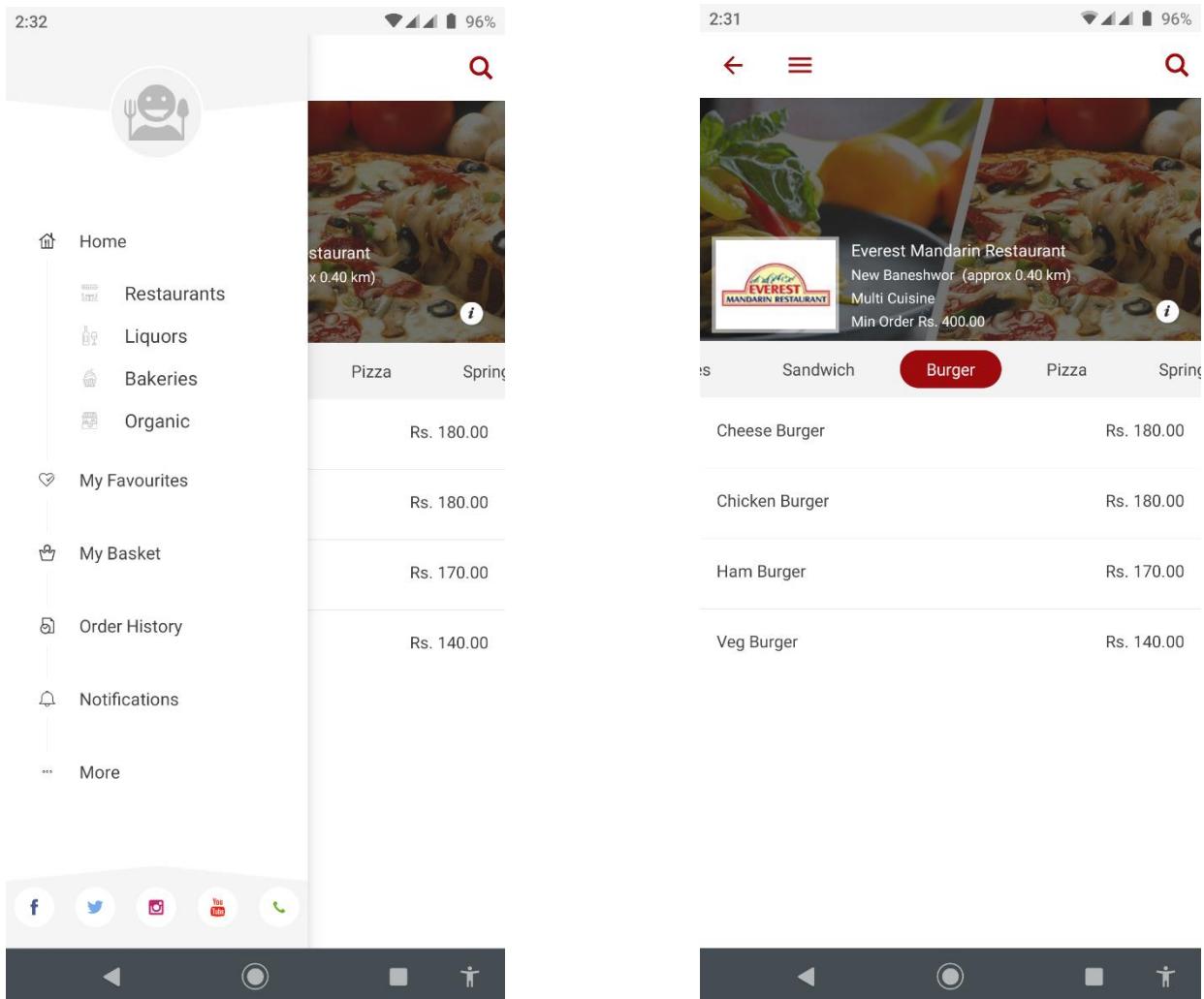


Figure 7: Foodmandu drawer board

Figure 8: Foodmandu food items

## 2.2. Findings of the Research

From the above research section, I have found that all of the Nepalese Food ordering applications have a great UI designs several features from these applications can be implemented in my application too. In Bhoj deals I like the way they have presented the foods. Bhoj have many unique features like Bhoj wallet, Scan and pay system. Whereas in Foodmandu I love their UI design and color combination. It looks simple but very attractive. I have also found that none of these applications have the food tracking system nor do they have food waste management system which I am going to implement on my application.

## 2.3. Resource required

- **Hardware requirements:** Laptop, Android phone, Internet Connection.
- **Software requirements:** Flutter, Dart, Python, and Database
- **For frontend:** Flutter, Dart will be used, Flutter is an open source framework to create high quality, high performance mobile applications across mobile operating systems - Android and iOS. It provides a simple, powerful, efficient and easy to understand SDK to write mobile application in Google's own language, *Dart* (tutorialspoint, 2019).
- **For Database:** MySQL is used. MySQL is a Structured Query Language (SQL) based open source relational database management system (RDBMS) supported by Oracle (Rouse, 2003).
- **For backend:** Python with Django framework has been used. Django is a free and open source framework written in Python. A framework is nothing more than a series of modules that facilitates development. These are bundled together and allow you, instead of scratch, to build applications or websites from an existing source. (PythonForBeginners, 2019) It follows a model-view-templet design pattern. It works perfectly with MySQL and SQLite. Django offers authentication which can be used to build a fully functional authentication system by running a simple command. (Django Software Foundation and individual contributors, 2019)

### 3. Chapter 3: Development to date

#### 3.1. Methodology

SDLC or the Software Development Life Cycle is a cycle is a process that produces software with the highest quality and lowest cost in the shortest time. SDLC includes a detailed plan for the development, alteration, maintenance and replacement of a software system. SDLC includes several different stages, including planning, design, installation, testing, and implementation. Popular SDLC models include model waterfall, spiraling, Agile, Prototype and many more. (Stackify, 2019)

##### 3.1.1. Comparison of Methodologies

- **Waterfall methodology**

The first Process Model to be implemented was the Waterfall System. Learning and using it is very easy. That phase must be completed in a Waterfall model before the next step can begin and the phases do not overlap. The concept of the waterfall is the earliest SDLC technique used to develop software. (TOOLSSQA.COM, 2013- 2020)

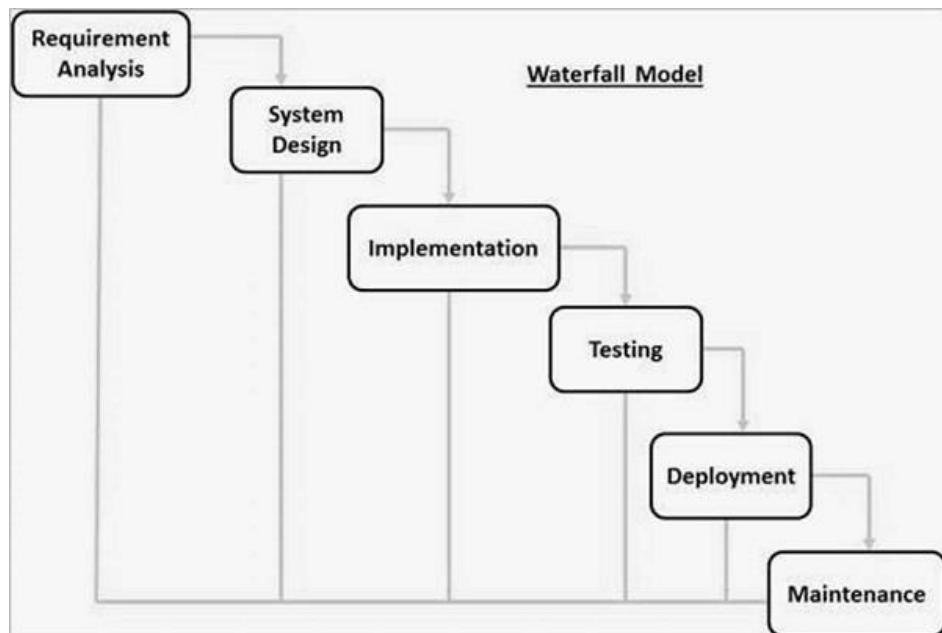


Figure 9: Waterfall model

- **Agile Model**

Agile is a process through which a team can manage a project by splitting it into several stages and involving constant stakeholder collaboration and continuous improvement and iteration at each stage. The Agile approach begins with consumers explaining how to use the end product and how it will solve the problem. It clarifies the project team's perceptions of the client. Once the work begins, teams go through a preparation, implementation, and evaluation process—which could only adjust the final deliverable to best fit the needs of the customer. Continuous cooperation is key to making full informed decisions, both among team members and with project stakeholders.

(Wrike, 2006-2019)

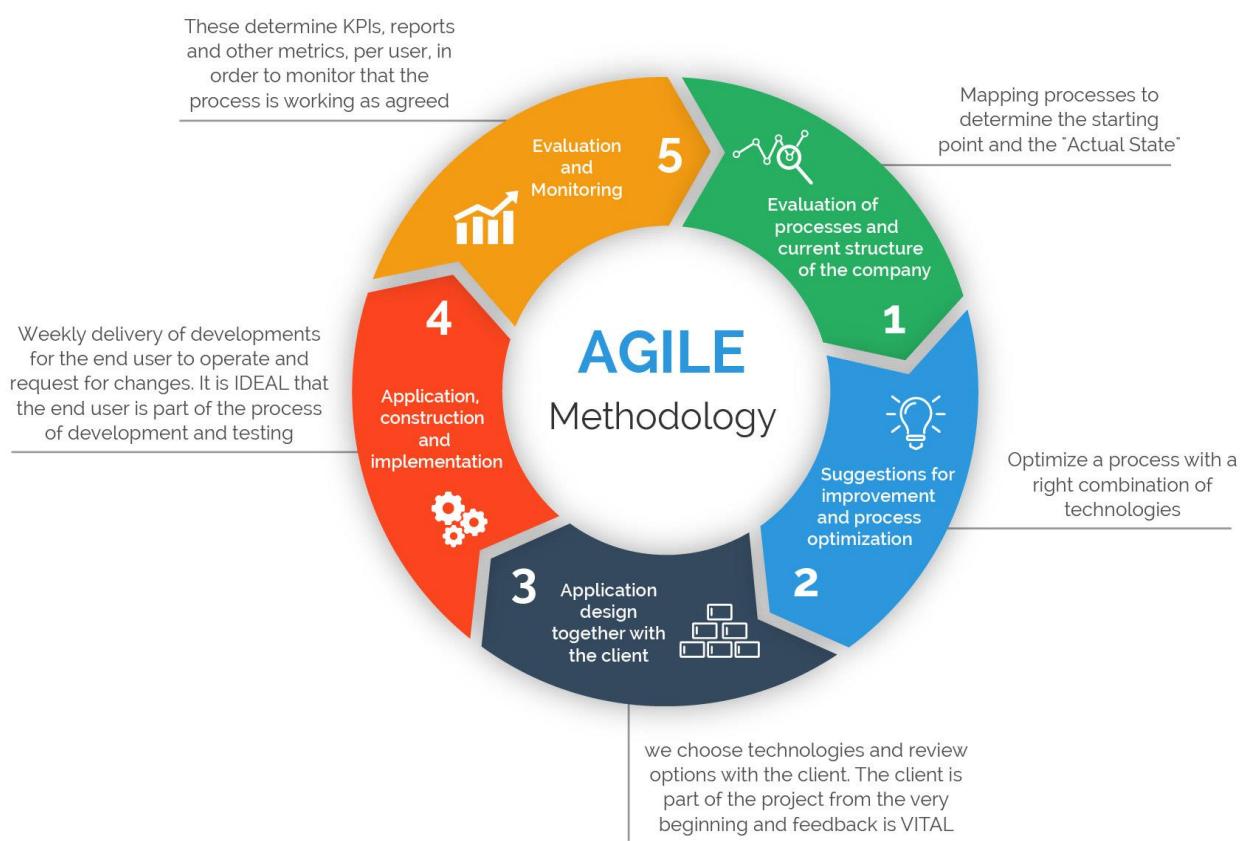
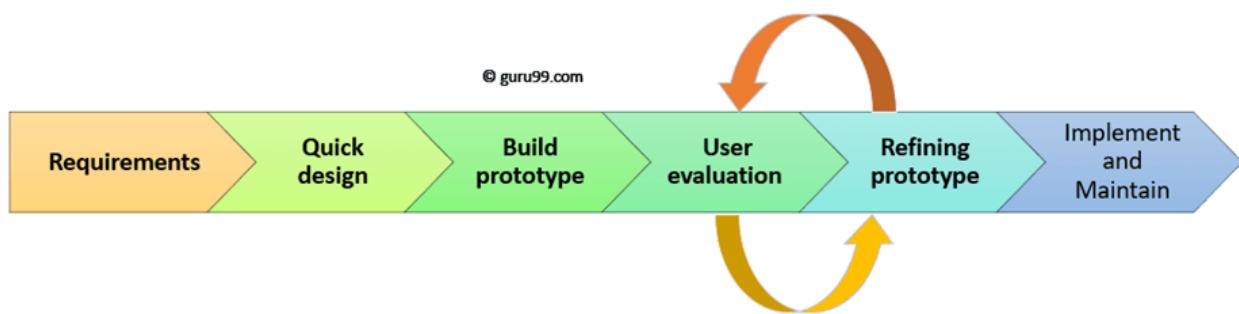


Figure 10: Agile model

- **Prototyping model**

Prototype methodology is defined as a model for software development in which a prototype is constructed, tested, and then reworked when necessary until an acceptable prototype is achieved. It also creates a basis for the final system to be produced. (Guru99, 2019)

- Requirements gathering and analysis
- Quick design
- Build a Prototype
- Refining prototype
- Implement Product and Maintain (Guru99, 2019)



*Figure 11: Prototyping model*

### 3.1.2. Selected Methodologies

I have selected prototyping model as a methodology to complete this project. In prototyping model a prototype is built, tested and then reworked as necessary until an acceptable outcome is achieved from which the complete product can be developed. (TechTarget, 2019)

This model works best in situations where not all requirements of the project are known in detail ahead of time. It is an iterative process of trial-and-error between developers and users. So, prototype module will help in check if the features would be functional or not. And in case it is not functional I can change and modify it or re-build the function.

### 3.2. Wireframe



Figure 12: front page wireframe



Figure 13: Register page wireframe

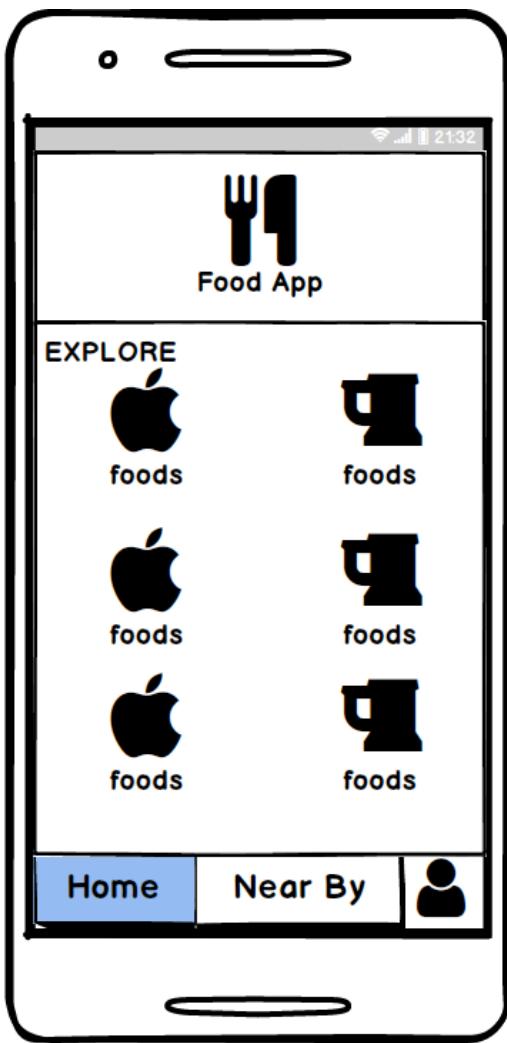


Figure 14: Home page wireframe



Figure 15: Nearby wireframe

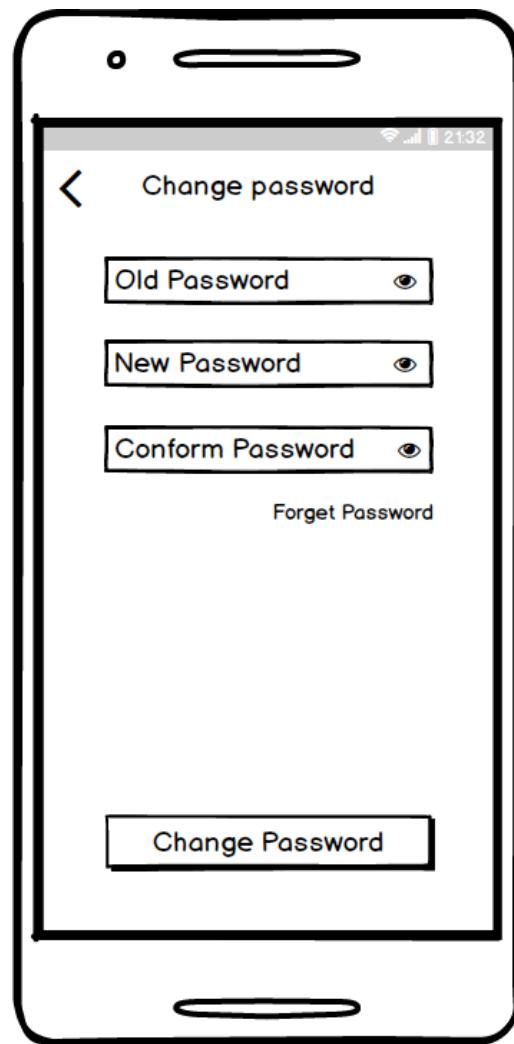
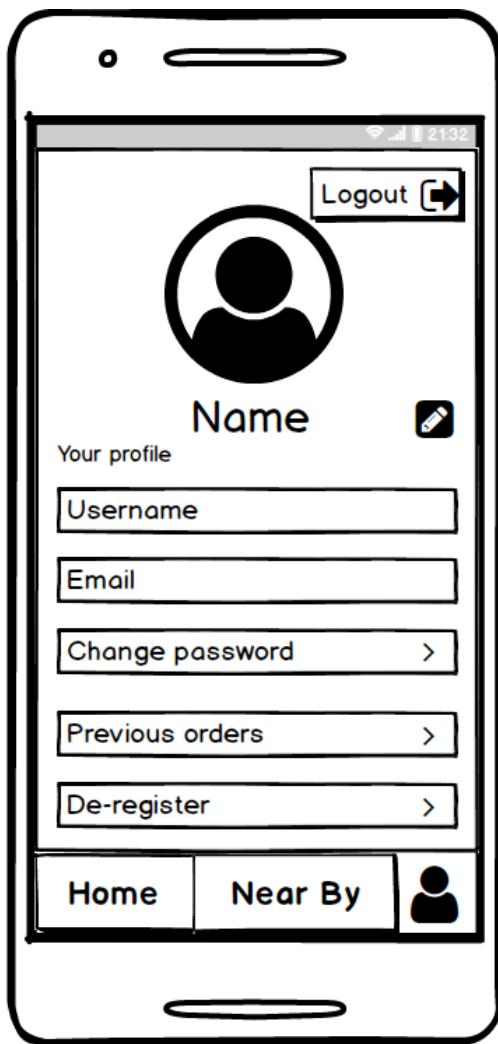


Figure 16: user dashboard wireframe

Figure 17: Change password wireframe



Figure 18: De-activate wireframe

Figure 19: Logout wireframe

### 3.3. Use-case Diagram

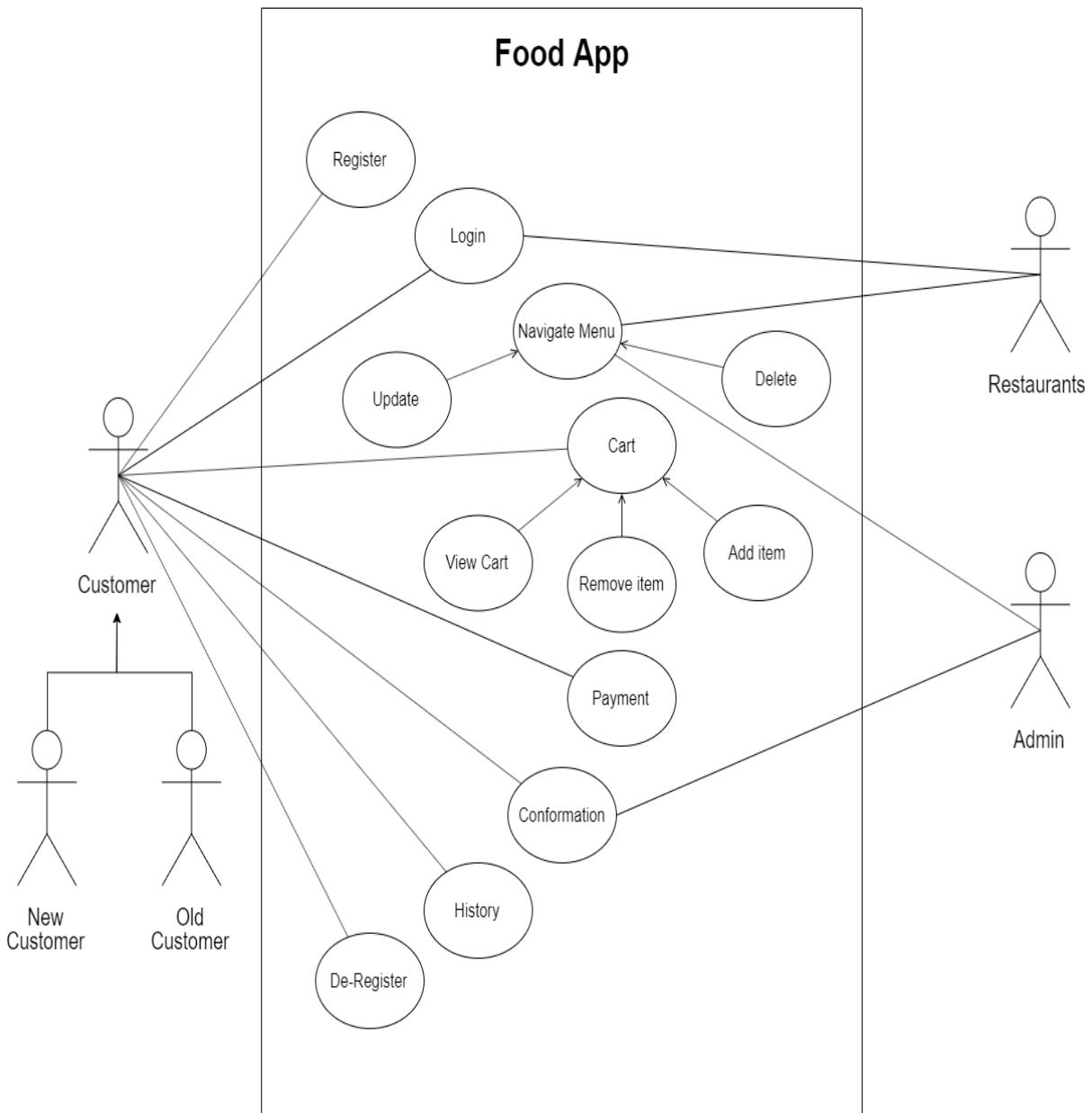


Figure 20: Use-case diagram

### 3.4. Initial ER-Diagram

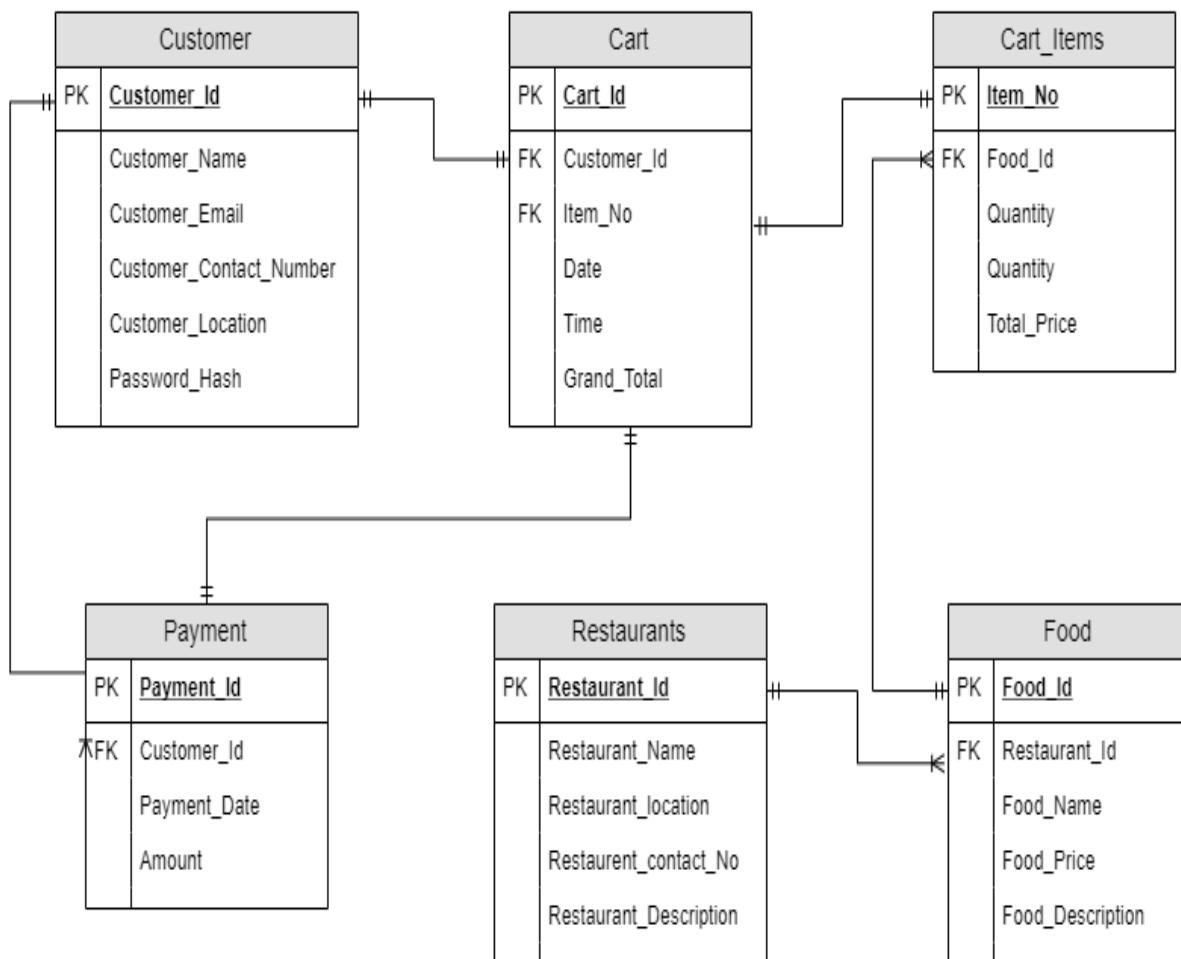


Figure 21: Initial ER-Diagram

### 3.5. Normalization

#### UNF

Restaurant (rest\_id, rest\_name, rest\_location, rest\_contact, rest\_type {cust\_id, cust\_name, cust\_email, cust\_contact, cust\_address\_id, country, state, city, zip\_code, street\_name {cart\_id, created\_date, time {food\_id, food\_name, food\_image, category\_id, food\_category, food\_quantity, unit\_price, line\_total}, grand\_total, payment\_id, payment\_date}})

#### 1NF

Removing repeating groups and creating separate table for those groups.

#### Final 1NF

Restaurant-1 (rest\_id, rest\_name, rest\_location, rest\_contact, rest\_type)

Rest-Cust-1 (rest\_id\*, cust\_id, cust\_name, cust\_email, cust\_contact, cust\_address\_id, country, state, city, zip\_code, street\_name)

Rest-Cust-Cart-1 (rest\_id\*, cust\_id\*, cart\_id, created\_date, time, grand\_total, payment\_id, payment\_date)

Rest-Cust-Cart-Food-1 (rest\_id\*, cust\_id\*, cart\_id\*, food\_id, food\_name, food\_image, category\_id, food\_category, food\_quantity, unit\_price, line\_total)

#### 2NF

Since Restaurant has only one attribute key it is already in 2NF,

Checking Partial dependency in Rest-Cust

- $\text{Cust\_id} \rightarrow \text{cust\_name}, \text{cust\_email}, \text{cust\_contact}, \text{cust\_address\_id}, \text{country}, \text{state}, \text{city}, \text{zip\_code}, \text{street\_name}$

Checking Partial dependency in Rest-Cust-Cart

- $\text{Cart\_id} \rightarrow \text{created\_date}, \text{time}, \text{grand\_total}, \text{payment\_id}, \text{payment\_date}$

Checking Partial dependency in Rest-Cust-Cart-Food

- $\text{food\_id} \rightarrow \text{food\_name}, \text{food\_image}, \text{category\_id}, \text{food\_category}, \text{unit\_price}$
- $\text{cart\_id}, \text{food\_id} \rightarrow \text{food\_quantity}, \text{line\_total}$

## Final 2NF

Restaurant-2 (rest\_id, rest\_name, rest\_location, rest\_contact, rest\_type)

Rest-Cust-2 (rest\_id\*, cust\_id\*)

Customer-2 (cust\_id, cust\_name, cust\_email, cust\_contact, cust\_address\_id, country, state, city, zip\_code, street\_name)

Rest-Cust-Cart-2 (rest\_id\*, cust\_id\*, cart\_id\*)

Cart-2 (cart\_id, created\_date, time, grand\_total, payment\_id, payment\_date)

Rest-Cust-Cart-Food-2 (rest\_id\*, cust\_id\*, cart\_id\*, food\_id\*)

Cart-food-2 (cart\_id\*, food\_id\*, food\_quantity, line\_total)

Food -2 (food\_id, food name, food\_image, category\_id, food\_category, unit\_price)

## 3NF

No transitive dependency in Restaurant,

Since Rest-Cust has only attribute keys it is already in 3NF,

Checking transitive dependency in Customer,

- $\text{cust\_address\_id} \rightarrow \text{country, state, city, zip\_code, street\_name}$

Since Rest-Cust-Cart has only attribute keys it is already in 3NF,

Checking transitive dependency in Cart,

- $\text{payment\_id} \rightarrow \text{payment\_date}$

Since Rest-Cust-Cart-Food has only attribute keys it is already in 3NF,

No transitive dependency in Cart-Food,

Checking transitive dependency in Food

- $\text{Category\_id} \rightarrow \text{Food\_category}$

## Final 3NF

Restaurant-3 (rest\_id, rest\_name, rest\_location, rest\_contact, rest\_type)

Rest-Cust-3 (rest\_id\*, cust\_id\*)

Customer-3 (cust\_id, cust\_name, cust\_email, cust\_contact, cust\_address\_id\*)

Location-3 (cust\_address\_id, country, state, city, zip\_code, street\_name)

Rest-Cust-Cart-3 (rest\_id\*, cust\_id\*, cart\_id\*)

Cart-3 (cart\_id, created\_date, time, grand\_total, payment\_id\*)

Payment-3 (payment\_id, payment\_date)

Rest-Cust-Cart-Food-3 (rest\_id\*, cust\_id\*, cart\_id\*, food\_id\*)

Cart-food-3 (cart\_id\*, food\_id\*, food\_quantity, line\_total)

Food -3 (food\_id, food name, food\_image, unit\_price, category\_id\*)

Category-3 (category\_id, food\_category)

### 3.6. Final ER-Diagram

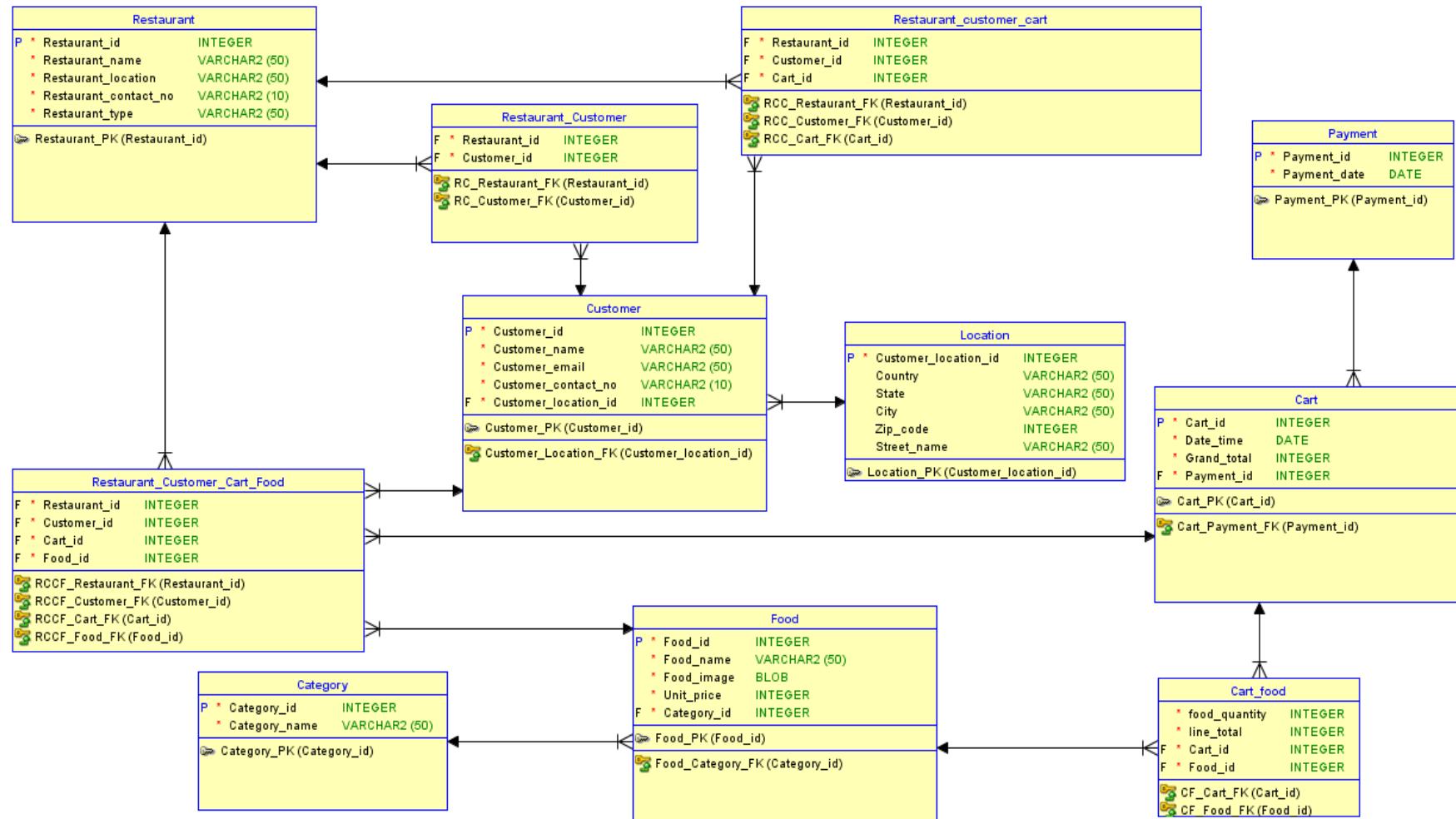


Figure 22: Final ER-Diagram

### 3.7. Frontend Development

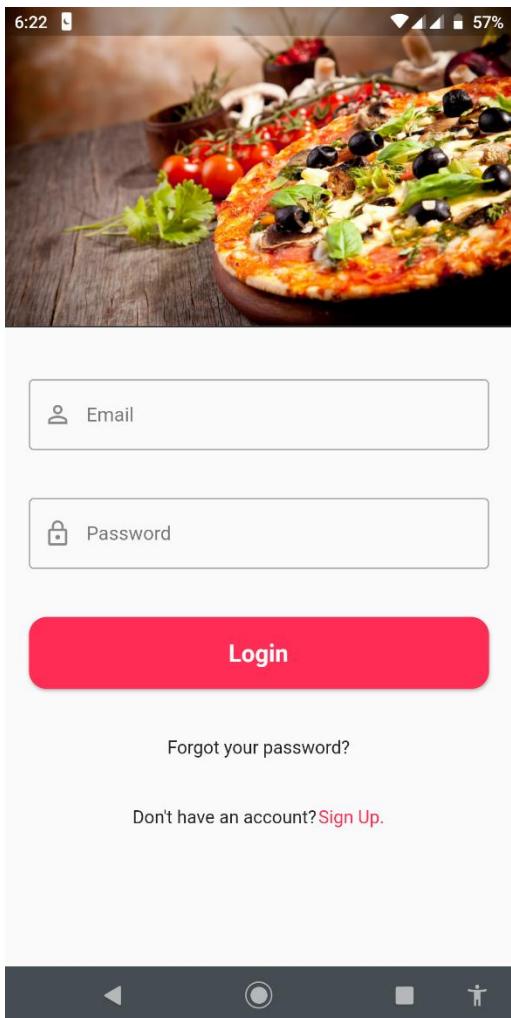


Figure 23: Signup page

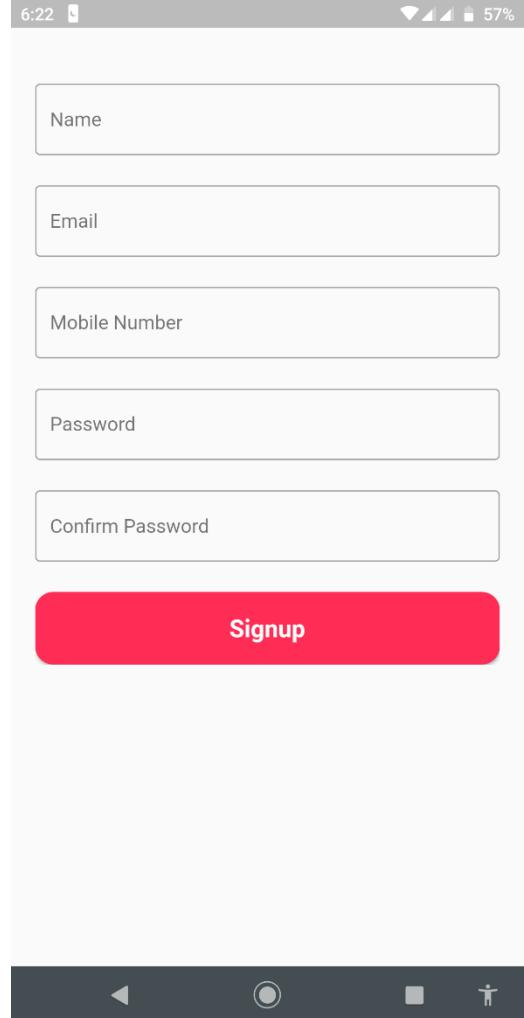


Figure 24: Signup page

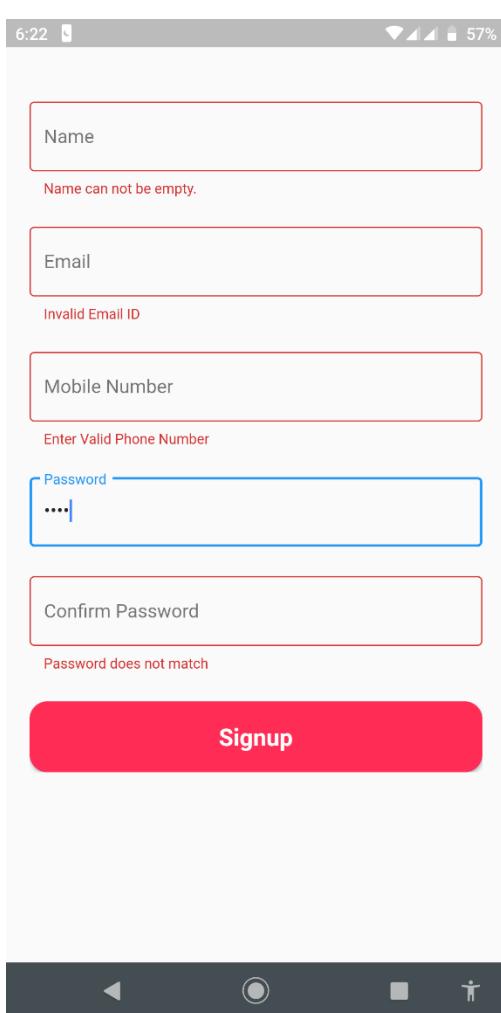


Figure 25: Signup validation

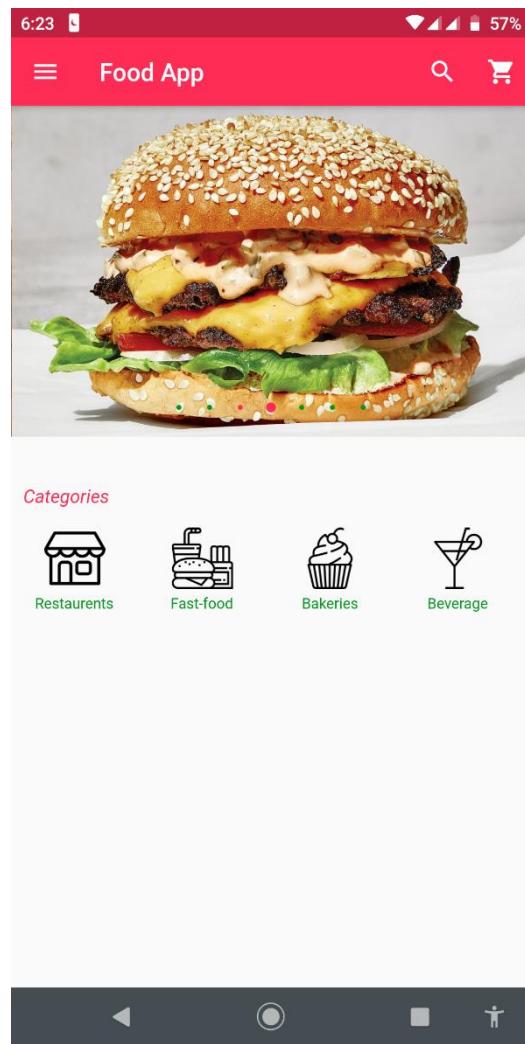


Figure 26: Home page

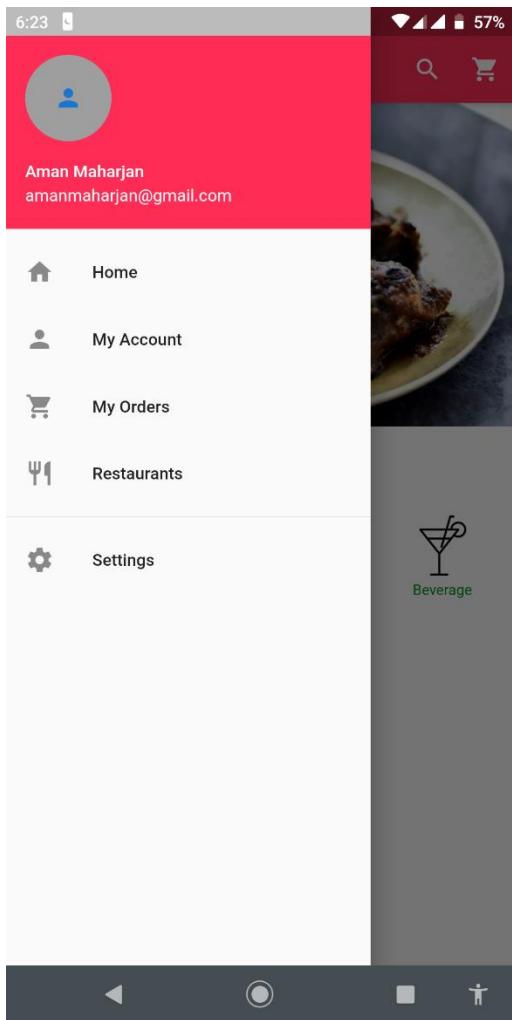


Figure 27: Dashboard

In the Frontend development part I have successfully developed the Signup page, login page, home page along with the validation of the fields. ([Reference: Appendix-D](#))

## 4. Chapter 4: Analysis of progress

### 4.1. Assignment of student table

S.N.	Task	Sub-Task	Start-Date	End-Date	Status
1.	Requirement gathering and analyzing	Brainstorming, Research, analysis	Nov 1, 2019	Nov 28, 2019	Completed
2.	Quick Design	ER-Diagram, Use-case, Normalization, Enhancement of diagrams	Nov 29, 2019	Dec 19, 2019	Completed
3.	Build a Prototype	Development login and signup page	Dec 20, 2019	Dec 24, 2019	Completed
		Development of home page	Dec 28, 2019	Jan 7, 2020	On going
		Food waste management	Jan 8, 2020	Jan 21, 2020	Not Started
		Development of other functions and pages	Jan 22, 2020	Feb 9, 2020	Not Started

4.	Evaluation	Evolution of the prototype	Feb 10, 2020	Feb 14, 2020	Not Started
5.	Refining Prototype	Code Management	Feb 15,2020	Feb 27, 2020	Not Started
		Unit Testing	Feb 21, 2020	Mar 19, 2020	Not Started
6.	Testing	Black box and white box testing	Mar 27,2020	Apr 9, 2020	Not Started
7.	Implement Product and maintain	Project final checking by implementing with few users	Apr 10,2020	Apr 16, 2020	Not Started

*Table 1: Analysis of progress*

## 4.2. Progress detail

**Requirement Gathering and Analysis:** According to the Gantt chart ([Reference appendix-A](#)) the first stage of the project was requirement gathering, and analysis which has been successfully completed. During the requirement gathering period I have finalized my project topic and have done some research about and went through some similar apps like Bhoj, Foodmandu.

**Quick design:** The second stage of the project was quick design in which I have created wireframes, UML designs i.e. Use-case diagram, Initial ER-Diagram, Normalization and final ER-Diagram. So the second stage was also successfully completed.

**Building a prototype:** The third stage of the project is building the prototype. This is the current state of the project. In the stage I have created Login page, Sign-up page, homepage within the time shown in Gantt chart ([Reference appendix-A](#)). The payment gateway will be postponed because according to the discussion with the supervisor the payment gateway for khalti will be upgrading soon So, I will be working on other features before.

Discussion with supervisors ([Reference appendix-C](#))

## 5. Chapter 5: Future work

S.N.	Task	Sub-Task	Start-Date	End-Date	Status
1.	Build a Prototype	Food waste management	Jan 8, 2020	Jan 21, 2020	Not Started
		Development of other functions and pages	Jan 22, 2020	Feb 9, 2002	Not Started
2.	Evaluation	Evolution of the prototype	Feb 10, 2020	Feb 14, 2020	Not Started
3.	Refining Prototype	Code Management	Feb 15,2020	Feb 27, 2020	Not Started
		Unit Testing	Feb 21, 2020	Mar 19, 2020	Not Started
4.	Testing	Black box and white box testing	Mar 27,2020	Apr 9, 2020	Not Started
5.	Implement Product and maintain	Project final checking by implementing with few users	Apr 10,2020	Apr 16, 2020	Not Started

Table 2: Future Work

## 5.1. Future work detailed

**Build a prototype:** The current state of the project is building the prototype in which food waste management, payment gateway, others functions are to be build.

**Evaluation:** The next stage of the project is evaluation in which we check the bugs and error in the prototype and if any function doesn't work we the function will be rebuild in the next stage.

**Refining prototype:** in this stage we refine the system make it bug free and make sure that all the functions are working properly.

**Testing:** The testing stage of the project is where we test the system in every way possible and finalize the product.

## 6. Chapter 6: References

Django Software Foundation and individual contributors. (2019) *Django* [Online]. Available from: <https://docs.djangoproject.com/en/3.0/ref/databases/> [Accessed 28 Dec 2019].

Guru99. (2019) *Prototyping Model in Software Engineering: Methodology, Process, Approach* [Online]. Available from: <https://www.guru99.com/software-engineering-prototyping-model.html> [Accessed 28 Oct 2019].

Iarashout. (2018) *What is Laravel and Why You Should Learn it?* [Online]. Available from: <https://www.larashout.com/what-is-laravel-and-why-you-should-learn-it> [Accessed 23 December 2019].

McKinsey & Company. (2019) *The changing market for food delivery* [Online]. Available from: <https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/the-changing-market-for-food-delivery> [Accessed 3 Nov 2019].

PythonForBeginners. (2019) *PythonForBeginners* [Online]. Available from: <https://www.pythonforbeginners.com/learn-python/what-is-django/> [Accessed 28 Dec 2019].

Rouse, M. (2003) *What is MySQL? - Definition from WhatIs.com* [Online]. Available from: <https://searchoracle.techtarget.com/definition/MySQL> [Accessed 2019].

Rouse, M. (2019) *e-commerce (electronic commerce)* [Online]. Available from: <https://searchcio.techtarget.com/definition/e-commerce> [Accessed 2019].

SoftwareTestingHelp.com. (2019) *Spiral Model – What Is SDLC Spiral Model?* [Online]. Available from: <https://www.softwaretestinghelp.com/spiral-model-what-is-sdlc-spiral-model/> [Accessed 02 Jan 2020].

Stackify. (2019) *What is SDLC? Understand the Software Development Life Cycle* [Online]. Available from: <https://stackify.com/what-is-sdlc/> [Accessed 02 Jan 2020].

TechTarget. (2019) *Prototyping Model* [Online]. Available from: <https://searchcio.techtarget.com/definition/Prototyping-Model> [Accessed 28 Oct 2019].

TOOLSQA.COM. (2013- 2020) *WaterFall Model* [Online]. Available from:  
<https://www.toolsqa.com/software-testing/waterfall-model/> [Accessed 02 Jan 2020].

tutorialspoint. (2019) *Flutter Tutorial* [Online]. Available from:  
<https://www.tutorialspoint.com/flutter/index.htm> [Accessed 21 December 2019].

tutorialspoint. (2019) *Laravel - Overview* [Online]. Available from:  
[https://www.tutorialspoint.com/laravel/laravel\\_overview.htm](https://www.tutorialspoint.com/laravel/laravel_overview.htm) [Accessed 2019].

Wrike. (2006-2019) *What is Agile Methodology in Project Management?* [Online]. Available from:  
<https://www.wrike.com/project-management-guide/faq/what-is-agile-methodology-in-project-management/> [Accessed 02 Jan 2020].

## 7. Appendix-A (Gantt chart)

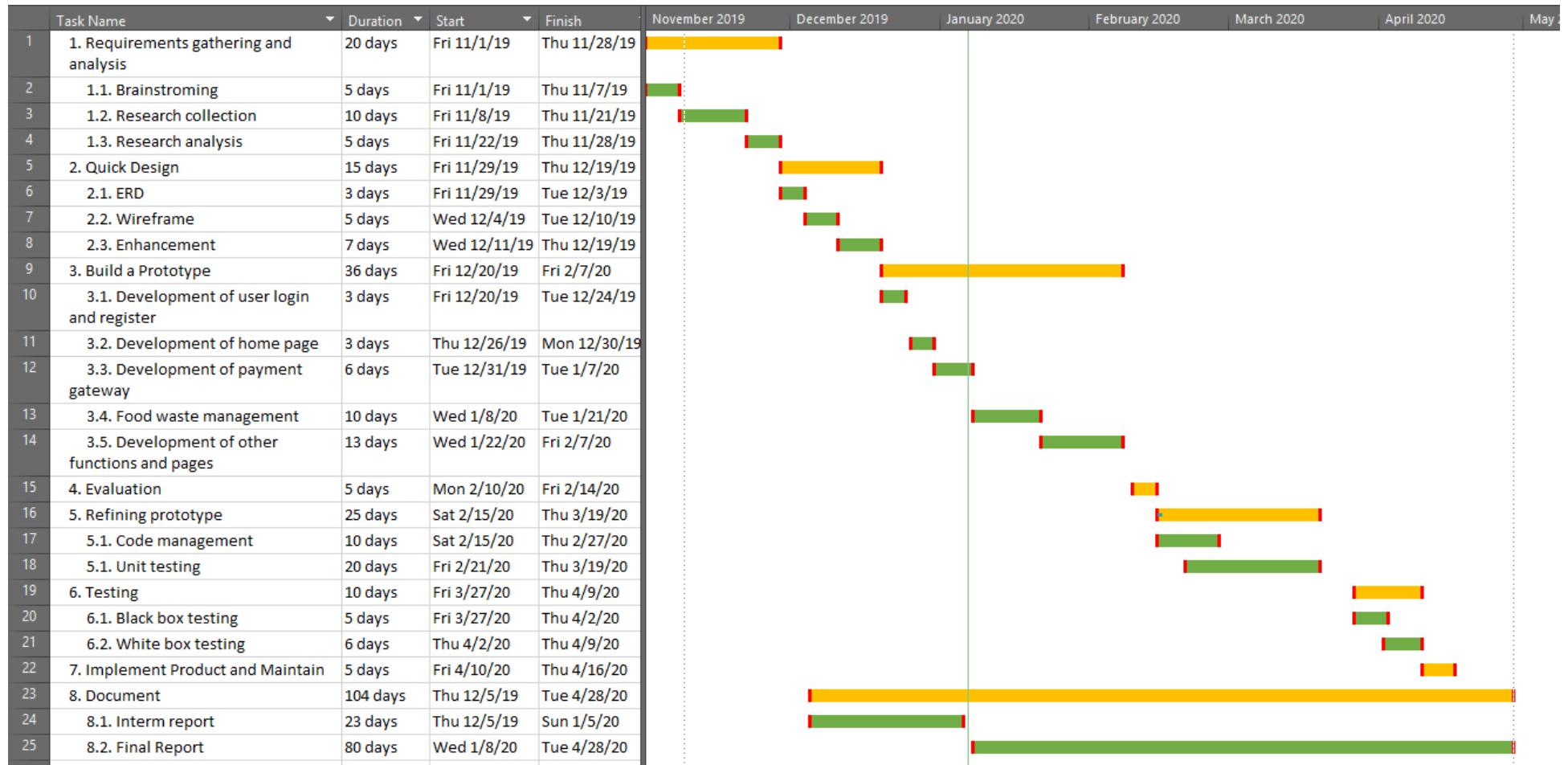


Figure 28: Gantt chart

## 8. Appendix-B (Survey Result)

Which smartphone do you use?

40 responses

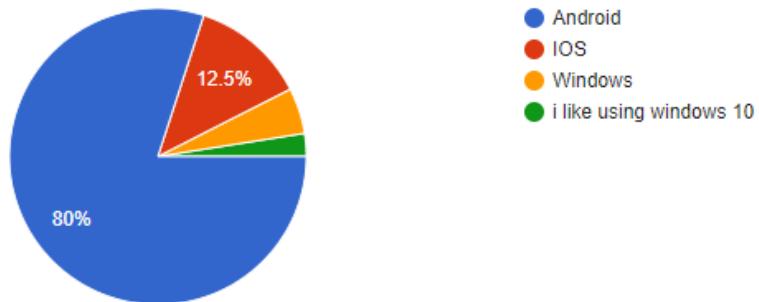


Figure 29: Survey 1

Have you ever ordered food online?

40 responses

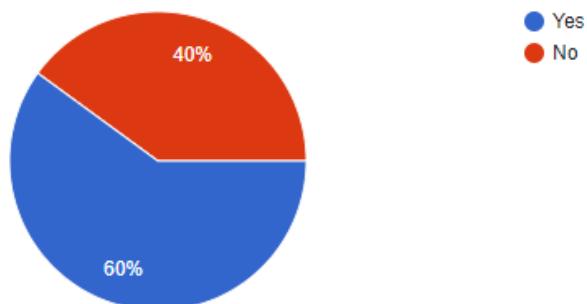


Figure 30: Survey 2

Do you face problem finding good quality food?

40 responses

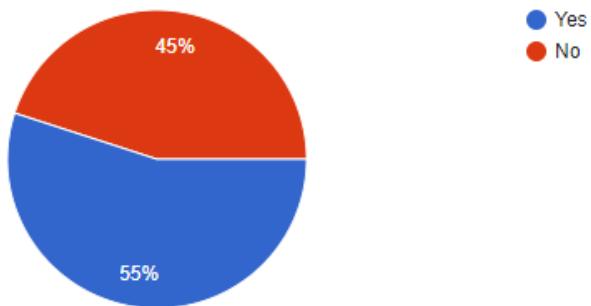


Figure 31: Survey 3

How often do you go out to eat?

40 responses

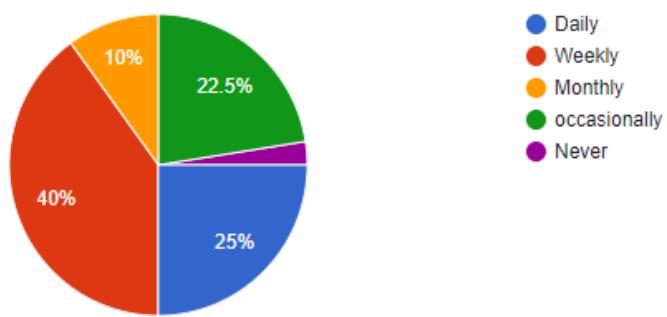


Figure 32: Survey 4

Do you think ordering food online will make things easier for both customers and restaurants?

40 responses

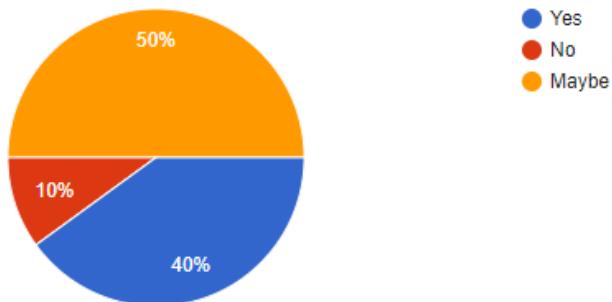


Figure 33: Survey 5

On a scale of 1 - 5, do you think ordering food online will be effective?

40 responses

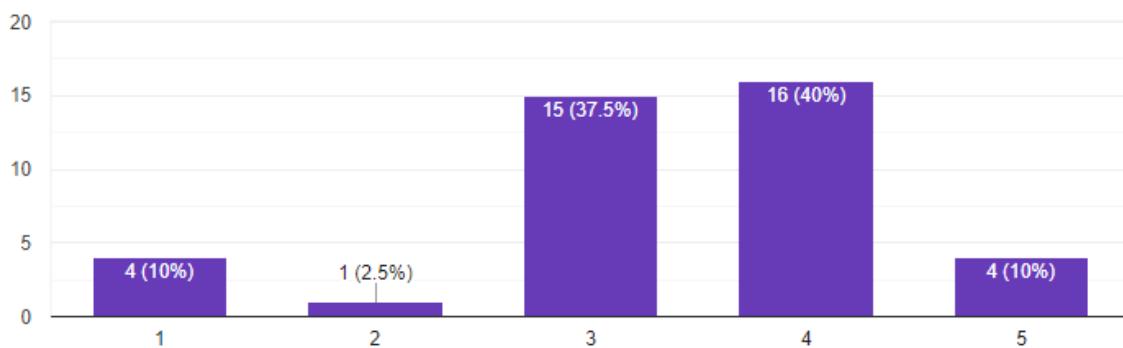


Figure 34: Survey 6

Do you think food waste management system (Offer food in less price during closing time) will be effective?

40 responses

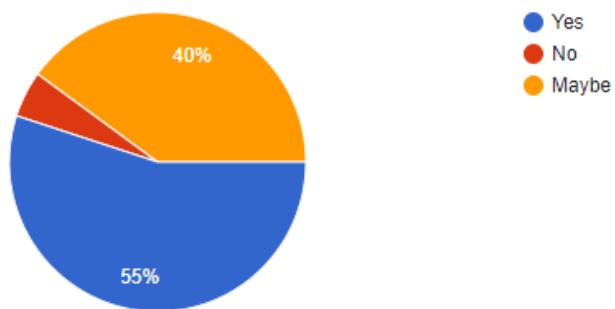


Figure 35: Survey 7

## 9. Appendix-C (Meeting Logs)

### Meeting log 1

**External Supervisor - Logbook Entry Sheet**

Use this form to record meetings with the supervisor. The completed form needs to be signed off by the student and the supervisor.

<b>Logbook Entry Sheet</b>	
Meeting No: 1	Date: Nov 10, 2019
Start Time: 12:00 PM	End Time: 02:00 PM
<b>Items Discussed:</b> <ul style="list-style-type: none"> <li>↳ Features</li> <li>↳ Payment</li> </ul>	
<b>Achievements:</b> <ul style="list-style-type: none"> <li>↳ Payment (Khalti)</li> </ul>	
<b>Problems (if any):</b> <ul style="list-style-type: none"> <li>↳ Lack of research in payment</li> <li>↳ Tracking research</li> <li>↳</li> </ul>	
<b>Tasks for Next Meeting:</b> <ul style="list-style-type: none"> <li>↳ Research about payment</li> <li>↳ food waste management features</li> <li>↳ tracking system research</li> </ul>	
<hr/> Student Signature	
<hr/> External Supervisor Signature	

Figure 36: Meeting log 1 (External)

**Internal Supervisor - Logbook Entry Sheet**

Use this form to record meetings with the supervisor. The completed form needs to be signed off by the student and the supervisor.

<b>Logbook Entry Sheet</b>	
Meeting No: 1	Date: Nov 10, 2019
Start Time: 12:00 PM	End Time: 02:00 PM
<b>Items Discussed:</b>	
<ul style="list-style-type: none"> <li>↳ features</li> <li>↳ Payment</li> </ul>	
<b>Achievements:</b>	
<ul style="list-style-type: none"> <li>↳ Payment (khalti)</li> </ul>	
<b>Problems (if any):</b>	
<ul style="list-style-type: none"> <li>↳ Lack of research in payment</li> <li>↳ Tracking research</li> </ul>	
<b>Tasks for Next Meeting:</b>	
<ul style="list-style-type: none"> <li>↳ Research about payment</li> <li>↳ food waste management features</li> <li>↳ tracking system</li> </ul>	
<hr/> Student Signature	
<hr/> Internal Supervisor Signature	

Figure 37: Meeting log 1 (Internal)

**Meeting log 2**

**External Supervisor - Logbook Entry Sheet**

Use this form to record meetings with the supervisor. The completed form needs to be signed off by the student and the supervisor.

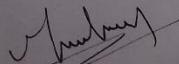
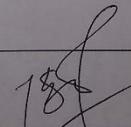
<b>Logbook Entry Sheet</b>	
Meeting No: 2	Date: 17 Nov, 2019
Start Time: 12:00pm	End Time: 2:00pm
<b>Items Discussed:</b> 1) Methodology 2) Payment 3) Google API (No of users)	
<b>Achievements:</b> 1) Payment 2) Methodology (Prototype)	
<b>Problems (if any):</b> 1) Google API (No of users)	
<b>Tasks for Next Meeting:</b> 1) Wireframe 2) MAP Research 3) Complete Proposal	
 Student Signature	
 External Supervisor Signature	

Figure 38: Meeting log 2 (External)

**Internal Supervisor - Logbook Entry Sheet**

Use this form to record meetings with the supervisor. The completed form needs to be signed off by the student and the supervisor.

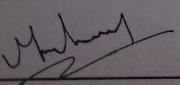
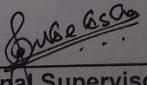
<b>Logbook Entry Sheet</b>	
Meeting No: 2	Date: 17 <sup>th</sup> Nov, 2019
Start Time: 12:00 PM	End Time: 2:00 PM
<b>Items Discussed:</b> 1) Methodology 2) Payment 3) Google API (No of users)	
<b>Achievements:</b> 1) Payment 2) Methodology (Prototype)	
<b>Problems (if any):</b> 1) Google API (No of users)	
<b>Tasks for Next Meeting:</b> 1) Wireframe 2) MAP Research 3) Complete proposal	
 Student Signature	
 Internal Supervisor Signature	

Figure 39: Meeting log 2 (Internal)

**Meeting log 3**

**External Supervisor - Logbook Entry Sheet**

Use this form to record meetings with the supervisor. The completed form needs to be signed off by the student and the supervisor.

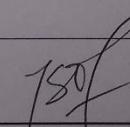
<b>Logbook Entry Sheet</b>	
Meeting No: 03	Date: 22 NOV, 2019
Start Time: 12:00 pm	End Time: 02:00 pm
<b>Items Discussed:</b> 1) Wireframe	
<b>Achievements:</b> 1) Wireframe finalized	
<b>Problems (if any):</b> 1) C# 2) Wireframe has few changes	
<b>Tasks for Next Meeting:</b> 1) C# System 2) ERD / Use case 3) Rest API 4) Extended Use case	
<hr/> Student Signature	
 <hr/> External Supervisor Signature	

Figure 40: Meeting log 3 (External)

**Internal Supervisor - Logbook Entry Sheet**

Use this form to record meetings with the supervisor. The completed form needs to be signed off by the student and the supervisor.

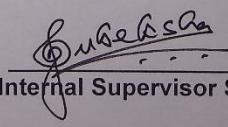
<b>Logbook Entry Sheet</b>	
Meeting No: 03	Date: 22 NOV, 2019
Start Time: 12:00 pm	End Time: 02:00 pm
<b>Items Discussed:</b> 1) Wireframe	
<b>Achievements:</b> 1) wireframe finalized	
<b>Problems (if any):</b> 1) Cart 2) wireframe unclear few changes	
<b>Tasks for Next Meeting:</b> 1) Cart system 2) ERD / Use case 3) Rest API 4) Extended Use case	
<hr/> Student Signature	
 <hr/> Internal Supervisor Signature	

Figure 41: Meeting log 3 (Internal)

**Meeting log 4**

**External Supervisor - Logbook Entry Sheet**

Use this form to record meetings with the supervisor. The completed form needs to be signed off by the student and the supervisor.

<b>Logbook Entry Sheet</b>	
Meeting No: 04	Date: 18 Dec, 2019
Start Time: 12:00 PM	End Time: 2:00 PM
<b>Items Discussed:</b> 1) ERD 2) Use case 3) Git 4) UI	
<b>Achievements:</b> 1) ERD 2) Use case	
<b>Problems (if any):</b> 1) ERD (Payment)	
<b>Tasks for Next Meeting:</b> 1) Development 2) work on Git 3) Make change in ERD (Payment)	
<hr/> Student Signature	
<hr/> External Supervisor Signature	

Figure 42: Meeting log 4 (External)

**Internal Supervisor - Logbook Entry Sheet**

Use this form to record meetings with the supervisor. The completed form needs to be signed off by the student and the supervisor.

<b>Logbook Entry Sheet</b>	
Meeting No: 04	Date: 20 Dec, 2019
Start Time: 12:00 PM	End Time: 2:00 PM
<b>Items Discussed:</b>	
<p>1) ERD 2) Use Case 3) Git 4) UI</p>	
<b>Achievements:</b>	
<p>1) ERD 2) Use case</p>	
<b>Problems (if any):</b>	
<p>1) ERD (Payment)</p>	
<b>Tasks for Next Meeting:</b>	

**Student Signature**

**Internal Supervisor Signature**

Figure 43: Meeting log 4 (Internal)

**Meeting log 5**

**External Supervisor - Logbook Entry Sheet**

Use this form to record meetings with the supervisor. The completed form needs to be signed off by the student and the supervisor.

<b>Logbook Entry Sheet</b>	
Meeting No: 05	Date: 22 <sup>nd</sup> December 2019
Start Time: 12:00 pm	End Time: 2:00 pm
<b>Items Discussed:</b> <ul style="list-style-type: none"> <li>• Final ERD</li> <li>• UI</li> </ul>	
<b>Achievements:</b> <ul style="list-style-type: none"> <li>• UI (homepage)</li> </ul>	
<b>Problems (if any):</b> <ul style="list-style-type: none"> <li>• Final ERD</li> </ul>	
<b>Tasks for Next Meeting:</b> <ul style="list-style-type: none"> <li>• Final ERD</li> <li>• Normalization</li> <li>• High Level use case</li> </ul>	
<hr/> Student Signature	
<hr/> External Supervisor Signature	

Figure 44: Meeting log 5 (External)

**Internal Supervisor - Logbook Entry Sheet**

Use this form to record meetings with the supervisor. The completed form needs to be signed off by the student and the supervisor.

<b>Logbook Entry Sheet</b>	
Meeting No: 05	Date: 22 <sup>nd</sup> dec, 2019
Start Time: 12:00 PM	End Time: 2:00 PM
<b>Items Discussed:</b> ↳ final ERD ↳ UI	
<b>Achievements:</b> ↳ UI (Homepage)	
<b>Problems (if any):</b> ↳ final ERD	
<b>Tasks for Next Meeting:</b> ↳ final ERD ↳ Normalization ↳ High level usecase	
<hr/> Student Signature	
<hr/> Internal Supervisor Signature	

Figure 45: Meeting log 5 (Internal)

**Meeting log 6**

**External Supervisor - Logbook Entry Sheet**

Use this form to record meetings with the supervisor. The completed form needs to be signed off by the student and the supervisor.

<b>Logbook Entry Sheet</b>	
Meeting No: 06	Date: 5 <sup>th</sup> Jan, 2020
Start Time: 12:00	End Time: 2:00
Items Discussed: • Normalization • Interim Report • Payment Gateway	
Achievements: • Normalization	
Problems (if any): • Add Category • Stop Payment Gateway	
Tasks for Next Meeting: • Completed Interim Report • Final ERD	
<hr/> Student Signature	
<hr/> External Supervisor Signature	

Figure 46: Meeting log 6 (External)

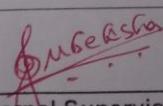
Logbook Entry Sheet	
Meeting No: 06	Date: 5 <sup>th</sup> Jan, 2020
Start Time: 12:00	End Time: 2:00
Items Discussed:	
<ul style="list-style-type: none"><li>• Normalization</li><li>• Interim Report</li><li>• Payment Gateway</li></ul>	
Achievements:	
<ul style="list-style-type: none"><li>• Normalization</li></ul>	
Problems (if any):	
<ul style="list-style-type: none"><li>• Add Category</li><li>• Postpone Payment Gateway</li></ul>	
Tasks for Next Meeting:	
<ul style="list-style-type: none"><li>• Completed Interim Report</li><li>• Final ERD</li></ul>	
Student Signature	 Internal Supervisor Signature

Figure 47: Meeting log 6 (Internal)

## 10. Appendix-D

### 10.1. Frontend Code

#### Main.dart

```
import 'package:flutter/material.dart';
import 'src/app.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
    Widget build(BuildContext context) {
        return MaterialApp(
            debugShowCheckedModeBanner: false,
            home: App(),
        );
    }
}
```

#### App.dart

```
import 'package:flutter/material.dart';
import 'screens/login.dart';

class App extends StatelessWidget{
    Widget build (context){
        return Scaffold(
            body: Login(),
        );
    }
}
```

#### Login.dart

```
import 'package:flutter/material.dart';
import '../mixins/validation_mixin.dart';
import 'homepage.dart';
import 'forgotpassword.dart';
import 'signup.dart';
```

```

class Login extends StatefulWidget {
  createState(){
    return LoginState();
  }
}

class LoginState extends State<Login> with ValidationMixin{

  final GlobalKey<FormState> formKey = GlobalKey<FormState>(); //gain access on form widget
  String email = '';
  String password = '';

  Widget build(context) {
    return Container(
      child: Form(
        key: formKey,
        child: SingleChildScrollView(
          child: Container( //to put image
            decoration: BoxDecoration(
              image: DecorationImage(
                image: AssetImage('Assets/img/image1.png'),
                fit: BoxFit.fitWidth,
                alignment: Alignment.topCenter,
              )
            ),
            child: Container( //for margin in form
              margin: EdgeInsets.all(20.0),
              child: Column(
                children: [
                  Container(margin: EdgeInsets.only(top: 290.0)),
                  emailField(),
                  Container(margin: EdgeInsets.all(20.0)),
                  passwordField(),
                  Container(margin: EdgeInsets.all(20.0)),
                  submitButton(),
                  Container(margin: EdgeInsets.all(20.0)),
                  forgotPasswordLink(),
                  Container(margin: EdgeInsets.all(20.0)),
                  signupLink(),
                ],
              ),
            ),
          ),
        ),
      ),
    );
  }
}

```

```
    );
}

Widget emailField(){
    return TextFormField(
        decoration: InputDecoration(
            border: OutlineInputBorder(),
            prefixIcon: Icon(Icons.person_outline),
            labelText: 'Email',
            hintText: 'you@example.com',
        ),
        keyboardType: TextInputType.emailAddress,
        validator: validateEmail,
        onSaved: (String value){
            email = value;
        },
    );
}

Widget passwordField(){
    return TextFormField(
        obscureText: true,
        decoration: InputDecoration(
            border: OutlineInputBorder(),
            prefixIcon: Icon(Icons.lock_outline),
            labelText: 'Password',
            hintText: '*****',
        ),
        validator: validatePassword,
        onSaved: (String value){
            password = value;
        },
    );
}

Widget submittButton(){
    return ButtonTheme(
        minWidth: MediaQuery.of(context).size.width,
        height: 60.0,
        shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(15.0)
        ),
        child: RaisedButton(
            //color: Colors.red,
            color: Color.fromRGBO(255, 45, 85, 1),

```

```
        textColor: Colors.white,
        child: Text('Login',
            style: TextStyle(
                fontSize: 20,
                fontWeight: FontWeight.bold,
            ),
        ),
        onPressed: () {
            if (formKey.currentState.validate()){ //only render if valid
                Navigator.push(context, MaterialPageRoute(builder: (context){
                    return Homepage();
                }));
            }
        },
    );
}
}

Widget forgotPasswordLink(){
    return InkWell(
        child: Text("Forgot your password?", style: TextStyle(fontSize: 15,,)),
        onTap: () {
            Navigator.push(context, MaterialPageRoute(builder: (context){
                return ForgetPassword();
            }));
        },
    );
}

Widget signupLink(){
    return Row(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
            Text("Don't have an account?", style: TextStyle(fontSize: 15,,)),
            InkWell(
                child: Text("Sign Up.", style: TextStyle(fontSize: 15, color: Color(0x
ffff2d55),,,),
                onTap: () {
                    Navigator.push(context, MaterialPageRoute(builder: (context){
                        return Signup();
                    }));
                },
            ),
        ],
    );
}
```

```

    }
}
```

**Signup.dart**

```

import 'package:flutter/material.dart';
import '../mixins/validation_mixin.dart';

class Signup extends StatefulWidget {
  createState() {
    return SignupState();
  }
}

class SignupState extends State<Signup> with ValidationMixin {
  //get all the methods from validateMixin

  final GlobalKey<FormState> formKey = GlobalKey<FormState>(); //gain access on form widget
  final TextEditingController passwordController = TextEditingController(); //to
get access of password field value
  String email = '';
  String password = '';
  String name = '';
  String mobileNumber = '';

  Widget build(context) {
    return Scaffold(
      body: Container(
        margin: EdgeInsets.all(20.0),
        child: SingleChildScrollView(
          child: Form(
            key: formKey,
            child: Column(
              children: [
                Container(margin: EdgeInsets.only(top: 50.0)),
                nameField(),
                Container(margin: EdgeInsets.only(top: 25.0)),
                emailField(),
                Container(margin: EdgeInsets.only(top: 25.0)),
                numberField(),
                Container(margin: EdgeInsets.only(top: 25.0)),
                passwordField(),
                Container(margin: EdgeInsets.only(top: 25.0)),
                confirmPasswordField(),
                Container(margin: EdgeInsets.only(top: 25.0)),

```

```
        submittButton(),
    ],
),
),
),
),
);
}

Widget nameField() {
return TextFormField(
decoration: InputDecoration(
border: OutlineInputBorder(),
labelText: 'Name',
hintText: 'Adam Levine',
),
validator: (String value){
if (value.isEmpty)
return 'Name can not be empty.';
else
return null;
},
onSaved: (String value) {
name = value;
},
);
}

Widget emailField() {
return TextFormField(
decoration: InputDecoration(
border: OutlineInputBorder(),
labelText: 'Email',
hintText: 'you@example.com',
),
keyboardType: TextInputType.emailAddress,
validator: validateEmail,
onSaved: (String value) {
email = value;
},
);
}

Widget numberField() {
return TextFormField(
```

```
decoration: InputDecoration(
    border: OutlineInputBorder(),
    labelText: 'Mobile Number',
    hintText: '9841000000',
),
validator: validateNumber,
onSaved: (String value) {
    mobileNumber = value;
},
);
}

Widget passwordField() {
return TextFormField(
    obscureText: true,
    decoration: InputDecoration(
        border: OutlineInputBorder(),
        labelText: 'Password',
        hintText: '*****',
),
controller: passwordController,
validator: validatePassword,
onSaved: (String value) {
    password = value;
},
);
}

Widget confirmPasswordField() {
return TextFormField(
    obscureText: true,
    decoration: InputDecoration(
        border: OutlineInputBorder(),
        labelText: 'Confirm Password',
        hintText: '*****',
),
validator: (String value) {
    if (passwordController.text != value)
        return 'Password does not match';
    else
        return null;
},
onSaved: (String value) {
    password = value;
},
}
```

```
    );
}

// Widget termsandconditionsButton(){
//   return SwitchListTile(
//     value: _formData['Accept Terms and Conitions'],
//     onChanged: (bool value) {
//       setState(() {
//         _formData['Accept Terms and Conitions'] = value;
//       });
//     },
//   )
// }

Widget submittButton() {
  return ButtonTheme(
    minWidth: MediaQuery.of(context).size.width,
    height: 60.0,
    shape:
      RoundedRectangleBorder(borderRadius: BorderRadius.circular(15.0)),
    child: RaisedButton(
      //color: Colors.red,
      color: Color.fromRGBO(255, 45, 85, 1),
      textColor: Colors.white,
      child: Text(
        'Signup',
        style: TextStyle(
          fontSize: 20,
          fontWeight: FontWeight.bold,
        ),
      ),
      onPressed: () {
        if (formKey.currentState.validate()) {
          //only saves if valid
          formKey.currentState.save();
          print('email is $email and password is $password');
        }
      },
    )));
}
```

**Homepage.dart**

```
import 'package:flutter/material.dart';
import 'components/image_carousel.dart';
import 'components/drawer.dart';
import 'components/vertical_listview.dart';

class Homepage extends StatefulWidget {
    @override
    _HomepageState createState() => _HomepageState();
}

class _HomepageState extends State<Homepage> {
    Widget build(context) {
        return MaterialApp(
            title: 'Food App',
            debugShowCheckedModeBanner: false,
            theme: ThemeData(
                primaryColor: Color.fromRGBO(255, 45, 85, 1),
                accentColor: Color.fromRGBO(3, 150, 30, 1),
            ),
            home: Scaffold(
                appBar: AppBar(
                    title: Text("Food App"),
                    actions: <Widget>[
                        IconButton(
                            icon: Icon(
                                Icons.search,
                                color: Colors.white,
                            ),
                            onPressed: () {},
                        ),
                        IconButton(
                            icon: Icon(
                                Icons.shopping_cart,
                                color: Colors.white,
                            ),
                            onPressed: () {},
                        )
                    ],
                ),
                body: ListView(
                    children: <Widget>[
                        ImageCarousel(),

```

```

        Padding(
            padding: EdgeInsets.all(10.0),
            child: Text('Categories', style: TextStyle(fontSize: 15.0, fontStyle: FontStyle.italic, color: Color.fromRGBO(255, 45, 85, 1),)),
        ),
        HorizontalListView(),
    ],
),
//side navigation
drawer: DrawerExtend(),
),
);
}
}

```

**Validation\_mixin.dart**

```

class ValidationMixin{

    String validateEmail (String value){
        Pattern pattern =
            r'^(?:[^<>()[]\\.,;:\\s@\\"]+\\.(?:[^<>()[]\\.,;:\\s@\\"]+)*|\\\"\\\")@((\\[[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\])|(([a-zA-Z\\-0-9]+\\.)+[a-zA-Z]{2,}))$';
        RegExp regex = new RegExp(pattern);
        if (!regex.hasMatch(value))
            return 'Invalid Email ID';
        else
            return null;
    }

    String validatePassword (String value){
        if (value.isEmpty || value.length < 4)
            return 'Password must have atleast 4 characters';
        else
            return null;
    }

    String validateNumber (String value){
        Pattern pattern =
            r'^(?:[+0]9)?[0-9]{10}$';
        RegExp regex = new RegExp(pattern);
        if (!regex.hasMatch(value))
            return 'Enter Valid Phone Number';
    }
}

```

```
        else
            return null;
    }
}
```

### Drawer.dart

```
import 'package:flutter/material.dart';

class DrawerExtend extends StatelessWidget{
    @override
    Widget build(BuildContext context){
        return Drawer(
            child: ListView(
                children: <Widget>[
                    user(),
                    home(),
                    myaccount(),
                    myorders(),
                    restaurants(),
                    Divider(),
                    settings(),
                ],
            ),
        );
    }

    Widget user() {
        return UserAccountsDrawerHeader(
            accountName: Text('Aman Maharjan'),
            accountEmail: Text('amanmaharjan@gmail.com'),
            currentAccountPicture: GestureDetector(
                child: CircleAvatar(
                    backgroundColor: Colors.grey,
                    child: Icon(Icons.person),
                ),
            ),
        );
    }

    Widget home() {
        return InkWell(
            child: ListTile(
                title: Text('Home'),
                leading: Icon(Icons.home),

```

```
        },
        onTap: (){

    },
    );
}

Widget myaccount() {
    return InkWell(
        child: ListTile(
            title: Text('My Account'),
            leading: Icon(Icons.person),
        ),
        onTap: (){

    },
    );
}

Widget myorders() {
    return InkWell(
        child: ListTile(
            title: Text('My Orders'),
            leading: Icon(Icons.shopping_cart),
        ),
        onTap: (){

    },
    );
}

Widget restaurants() {
    return InkWell(
        child: ListTile(
            title: Text('Restaurants'),
            leading: Icon(Icons.restaurant),
        ),
        onTap: (){

    },
    );
}

Widget settings() {
    return InkWell(
```

```

        child: ListTile(
            title: Text('Settings'),
            leading: Icon(Icons.settings),
        ),
        onTap: (){
            },
        );
    }
}

```

**Horizontal\_listview.dart**

```

import 'package:flutter/material.dart';

class HorizontalListview extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
        return Container(
            height: 100.0,
            child: ListView(
                scrollDirection: Axis.horizontal,
                children: <Widget>[
                    Category(
                        imageLocation: 'Assets/img/cats/restaurants.png',
                        imageCaption: 'Restaurents',
                    ),
                    Category(
                        imageLocation: 'Assets/img/cats/fast-food.png',
                        imageCaption: 'Fast-food',
                    ),
                    Category(
                        imageLocation: 'Assets/img/cats/bakeries.png',
                        imageCaption: 'Bakeries',
                    ),
                    Category(
                        imageLocation: 'Assets/img/cats/beverage.png',
                        imageCaption: 'Beverage',
                    ),
                ],
            ),
        );
    }
}

```

```

class Category extends StatelessWidget {
    final String imageLocation;
    final String imageCaption;

    Category({
        this.imageLocation,
        this.imageCaption,
    });

    @override
    Widget build(BuildContext context) {
        return Padding(
            padding: EdgeInsets.all(2.0),
            child: InkWell(
                child: Container(
                    width: 100.0,
                    child: ListTile(
                        title: Image.asset(
                            imageLocation,
                            width: 50.0,
                            height: 50.0,
                        ),
                        subtitle: Container(
                            padding: EdgeInsets.only(top: 5.0),
                            alignment: Alignment.topCenter,
                            child: Text(
                                imageCaption,
                                style: TextStyle(
                                    fontSize: 12.0,
                                    color: Color.fromRGBO(3, 150, 30, 1),
                                ),
                            ),
                        )),
                ),
                onTap: () {},
            );
    }
}

```

**Image\_carousel.dart**

```

import 'package:flutter/material.dart';
import 'package:carousel_pro/carousel_pro.dart';

```

```
class ImageCarousel extends StatelessWidget{
  @override
  Widget build(BuildContext context){
    return Container(
      padding: EdgeInsets.only(top: 30),
      height: 300.0,
      transform: Matrix4.translationValues(0.0, -30.0, 0.0),
      child: Carousel(
        boxFit: BoxFit.cover,
        images: [
          AssetImage('Assets/img/choila.jpg'),
          AssetImage('Assets/img/kattiroll.jpg'),
          AssetImage('Assets/img/kimbap.jpg'),
          AssetImage('Assets/img/burger.jpg'),
          AssetImage('Assets/img/biryani.jpg'),
          AssetImage('Assets/img/momo.jpg'),
          AssetImage('Assets/img/pizza.png'),
        ],
        autoplay: true,
        animationCurve: Curves.fastOutSlowIn,
        animationDuration: Duration(milliseconds: 1000),
        dotSize: 4.0,
        dotColor: Color.fromRGBO(3, 150, 30, 1),
        dotIncreasedColor: Color.fromRGBO(255, 45, 85, 1),
        dotBgColor: Color.fromRGBO(3, 150, 30, 0),
      ),
    );
  }
}
```