

DailyDish

Final Year Project – Mid Report

Session 2017-2020

A project submitted in partial fulfilment of the
COMSATS University Degree
of
BS in Computer Science (CUI)



Department of Computer Science
COMSATS Institute of Information Technology, Lahore

29 July 2020

Evaluation

Project Group Members (To be filled by students)				(To be filled by supervisor)
Sr. #	Reg. #	Student Name	*Signature	Obtained Marks (Total Marks: 10)
(i)	SDP-SP17-BCS-142	Saad Ahmed		
(ii)	SDP-SP17-BCS-159	Umar Anwaar		

Supervisor Name: Professor Abdul Qayyum Supervisor Signature: _____

Remarks (if any): _____

Abstract:

Smartphones have become the need of everybody in this modern era. There are tons of Android and IOS apps facilitating humans in enormous ways. We are developing Android and IOS App that would help the non-residential citizens have affordable access to fresh, home-cooked food at their door steps. Dozens of people migrate to cities from villages and small cities for higher studies or doing jobs who do not have access to affordable and hygienic food. Our app would enable them to have convenient access to nutritious homemade food. Moreover, this application will create self-employment among people, especially women who can create a sustainable business from their very own kitchen.

Table of Contents

DailyDish.....	i
Final Year Project – Mid Report.....	i
Session 2017-2020.....	i
Evaluation.....	ii
Abstract:	iii
1 Introduction	1
1.1 Introduction.....	1
1.2 Objectives.....	2
1.3 Problem Statement.....	2
1.4 Assumptions and Constraints.....	3
1.5 Project Scope	4
2 Requirements Analysis.....	4
2.1 Literature review / Existing system study	4
2.2 Stakeholders list (Actors)	8
2.3 Requirements elicitation.....	9
2.3.1 Functional requirements	9
2.3.2 Non-functional requirements	12
2.4 Use case descriptions	14
2.4.1 Customer Meal Ordering	14
2.4.2 Chef Menu Update	16
2.4.3 Kitchen Registration.....	16
2.4.4 Delivery-Man Job.....	17
2.4.5 User Registration	18
2.4.6 Delivery Man Registration	19
2.4.7 Administrator.....	20
2.5 Use case design.....	22
2.5.1 Diagram 1	22
2.5.2 Diagram 2	23
2.5.3 Diagram 3	24
2.5.4 Diagram 4	25
2.5.5 Diagram 5	26
2.5.6 Diagram 6	27
2.5.7 Diagram 7	28
2.6 Software development life cycle model (justification on why this particular model is considered)	29
3 System Design	30
3.1 Work breakdown structure (WBS).....	30
3.2 Activity diagram.....	31

3.2.1	Diagram 1	31
3.2.2	Diagram 2	32
3.2.3	Diagram 3	33
3.2.4	Diagram 4	34
3.2.5	Diagram 5	35
3.3	Sequence diagram	36
3.4	Software architecture	37
3.5	Class diagram.....	37
3.6	Database diagram.....	38
3.7	Network diagram (Gantt chart)	39
3.8	Prototyping.....	40
3.8.1	Diagram 1	40
3.8.2	Diagram 2	42
3.8.3	Diagram 3	43
3.8.4	Diagram 4	44
3.8.5	Diagram 5	45
3.8.6	Diagram 6	46
3.8.7	Diagram 7	47
3.8.8	Diagram 8	48
3.8.9	Diagram 9	49
3.8.10	Diagram 10	50
3.8.11	Diagram 11	51
3.8.12	Diagram 12	52
4	Conclusion	53
4.1	Problems Faced and Lesson Learned	53
4.1.1	Problems Faced	53
4.1.2	Lessons Learned.....	53
4.2	Project Summary	53
5	References.....	54

List of Figures

Figure 2. 1 Home Page of Food Panda Application	5
Figure 2. 2 Homepage Cheetay Application	6
Figure 2.3 Homepage of Eat Mubarak Application	6
Figure 2.4 Home Chef Page Eat Mubarak	7
Figure 2.5 Homepage of McDelivery	7
Figure 2. 6 Use Case Diagram for Customer Ordering Meal	22
Figure 2. 7 Use case diagram for Deliveryman	23
Figure 2. 8 User case diagram for Chef	24
Figure 2. 9 Use case diagram for Kitchen Registration	25
Figure 2. 10 Use case diagram for Client Registration	26
Figure 2. 11 Use case diagram for Delivery Man Registration	27
Figure 2. 12 Use case diagram for Client Registration	28
Figure 2. 13 Framework of Agile model with each phase of SDLC shown	29
Figure 3. 1Work Breakdown Structure Representing various functions and components of product.....	30
Figure 3. 2 Activity Diagram showing login process	31
Figure 3. 3 Activity diagram showing functions of Customer.....	32
Figure 3. 4 Activity diagram showing functions of the Chef.....	33
Figure 3. 5 Activity diagram showing functions of the Deliveryman	34
Figure 3. 6 Activity diagram new use	35
Figure 3.7 Sequence diagram representing relationship between Customer, Chef, and Deliveryman.....	36
Figure 3. 8 Brief representation of Software architecture	37
Figure 3. 9 Class diagram for the project.....	37
Figure 3. 10 ERD illustrating the relationship between the entities in the project	38
Figure 3. 11 Network Diagram presenting the project schedule.....	39
Figure 3. 12 Homepage	40
Figure 3. 13 Homepage2	41
Figure 3. 14 Homepage3	41
Figure 3. 15 Profile Selection	42
Figure 3. 16 Registration page	43
Figure 3. 17 Profile Page.....	44

Figure 3. 18 Menu Page from view of the Customers	45
Figure 3. 19 Chef Signup	46
Figure 3. 20 Chef Log in.....	47
Figure 3. 21 Profile page for the Cook.....	48
Figure 3. 22 Menu Update Form.....	49
Figure 3. 23 Chef Profile Options	50
Figure 3. 24 Chef's Menu	51
Figure 3. 25 Contact Us	52

List of Tables

Table 1. 1 Abbreviation Table	viii
Table 2. 1 FR01 Functional Requirements of Registering.....	9
Table 2. 2 FR02 Functional Requirements of Login	10
Table 2. 3 FR03 Functional Requirements for Menu	10
Table 2. 4 FR04 Functional Requirements of Order	11
Table 2. 5 FR05 Functional Requirements of Notifications.....	11
Table 2. 6 FR06 Functional Requirements of Profile Management	12
Table 2. 7 FR07 Functional Requirements of Feedback System.....	12
Table 2. 8 NFR02 Non-Functional Requirements of Interface Design	13
Table 2. 9 NFR02 Non-Functional Requirements of Performance	13
Table 2. 10NFR03 Non-Functional Requirements of Reliability	13
Table 2. 11 NFR04 Non-Functional Requirements of Maintainability	14
Table 2. 12 NFR05 Non-Functional Requirements of Security	14
Table 2. 13 Use Case Description 1- Customer	14
Table 2. 14 Use Case Description 2- Chef.....	16
Table 2. 15 Use Case Description 3- Chef Registration	16
Table 2. 16 Use Case Description 4- Delivery-Man.....	17
Table 2. 17 Use Case Description 5- Guest	18
Table 2. 18 Use Case Description 6 Delivery Man Reg	19
Table 2. 19 Use Case Description 7- Admin	20

Abbreviation table

Term	Full form
SDLC	Software Development Life Cycle
HTML	Hyper Text Mark-up Language
IDE	Integrated Development Environment
Expo	Expo CLI Development Environment
Admin	Administrator
FR	Functional Requirement
NFR	Non Functional Requirement

Table 1. 1 Abbreviation Table

1 Introduction

This chapter includes a review about the project “**DailyDish**”

1.1 Introduction

Healthy and nutritious food is essential for human beings to perform optimal in their respective fields. Our Application is focused upon delivering healthy, full of vital nutrients and yet delicious food to the people who don't have access to home cooked food.

For many centuries people have migrated from their home towns and villages to cities for better opportunities. One of the major problems settling as a bachelor or as a Student in urban life is access to clean and healthy food. The Application we are designing will let our users order food directly from home based kitchens.

Anyone who wants to work from their very own kitchen can sign up to the application with a few simple steps by providing a few credentials. This will enhance self-employment among people such as women, who prefer to work from their own kitchen and can earn reputable income. This application will also provide a favourable comfort zone to handicapped or physically challenged vendors to run business from their home.

From a consumer point of view this application will provide psychological satisfaction of consuming clean and healthy food every day without any hassle. This will lead towards a healthy lifestyle as well as the capability of the individual will also rise ultimately.

This application will open opportunities for individuals who own a bike or any vehicle. They can deliver food from the kitchen to the customer and earn a reasonable amount of money. Main functionalities of application are listed below:

- 1) User can sign-up or Log-in into the application through their email id and password.
- 2) Customer can order their favourite dish by searching or selecting from variety of kitchens made home-chef on the application.
- 3) Application will allow users options like self-pick up or home delivery.
- 4) If any chef introduce special offer or dish, customer within 5 to 10 km of range will be notify through notification on their phone.
- 5) Integrity / Popularity of any chef and their kitchen will be based upon user ratings and reviews.
- 6) Deliverymen around the area will receive notification of job and will have option to accept the job.
- 7) Users can see status of their order at each stage and estimated time to receive the order.
- 8) Chefs can accept or decline order (along with proper reason) and update their menu or pricing.

Furthermore we would integrate Admin Panel through which admin can view sales reports, activity feed, complaints and even can remove or ban deliveryman due to regular customer complaints and negative reviews.

1.2 Objectives

DailyDish will not be just an ordinary application which connects restaurants and its customers but will also address everyday problems of struggling individuals for clean, healthy and sustainable nutritious food. Moreover, the application will focus upon to provide business opportunities for household women which can turn their cooking skills into a profitable business venture.

- 1) Daily Dish will allow home based kitchens to sign up and set up a kitchen-restaurant on our application.
- 2) Users within 5 kilometer of radius would be notified about the new kitchens added around their area.
- 3) Users can order meals one time or can get weekly or monthly subscriptions according to the variety of menu provided by the kitchen.
- 4) Application focuses upon to provide benefit to both parties e.g. Application users as well as the vendors.
- 5) Our application aims to provide clean homemade food to the person who wants to enjoy a delicious meal during or after his/her hectic routine.
- 6) Moreover, applications provide the ability to turn your kitchen into a business hub which can lead to endless opportunities in future.
- 7) Electronic mode of payments to ensure transparency and equity.
- 8) Designing user-friendly interface to allow customers to place order and cook to manage menus with convenience.

1.3 Problem Statement

Due to increase in urbanization, in recent years people from less developed areas are settling in cities. Majorly students and people working in these cities have very limited access to clean, healthy and hygienic meal. Food offered by restaurants are not fit to be consumed on regular basis due to which our targeted audience find it immensely difficult to gain access to home cooked food for their everyday consumption.

- **Inconvenient access:**

People living in hostels or independently without their families, find it difficult to attain nutritious food which lead to health problems such as obesity and even in some cases anxiety, depression and low productivity in their respective fields.

- **Unemployment:**

In recent time as we go through pandemic, many business have hit economically. Our application will provide opportunity for home based chef to start a scalable business at ease of their kitchen. Especially for women who have basic skills of cooking can make decent return from their skills in a secure environment.

Students or any person who owns a vehicle (Motorcycle) can register to be DailyDish Deliveryman and can earn little side money by performing delivery jobs near their area.

- **Limited Service for Targeted Market or Audience:**

In recent times many restaurant affiliated platforms (FoodPanda [1] , Cheetay [2]) have introduced option of home cooked food but their main focus is upon promoting commercial restaurants along with it. This can cause confusion and even distraction for customer seeking for home cooked food. Our application solely focus upon the motive of clean, healthy and nutritious food sourced only from home based kitchens.

1.4 Assumptions and Constraints

It is assumed that the deliverymen have their own conveyance for transporting and carrying the food from cook to the customers. It's also supposed that the prospective users of our app own and possess mobile phones. Customers are well-acquainted with a smart phone. The users should be familiar with the user-interface of our app. We suppose that the interface of our app is intuitive, simple, and self-explanatory. A cook must take care of hygiene and consistency in quality.

Constraints that we possibly could face are listed below:

- The food from the cook could be late because of some certain personal matters of home cook. For example, sudden death might occur and unavailability of generator during load shedding etc.
- Deliverymen might get late due to strange directions.
- Some pre orders might get cancelled due to unavailability of certain products or ingredients in market at that time.
- Deliverymen may not be able to reach their designated destinations in time due to serious nature acts like heavy rain, and thus unable to deliver the food in time due to factors beyond the scope of the app.
- Prices of ingredients might compromise the buying power of suppliers, thereby preventing them to purchase the ingredient, and hence, update the food menu,
- Deliverymen or cooks might not be able to get their salaries from the customers due to the certain factors unmanageable by the app, for example, a customer could refuse the deliverymen the payment due to a reason unspecified.
- There could be glitches and bugs in app that could prevent the execution of an essential task, for example, app might be incompatible with certain smart phone.

1.5 Project Scope

This product will open opportunities for household women or men to earn money by using their passion and talent in their own kitchen. With the moto “**Mera Kitchen Mera Restaurant**”

This product will be useful for both the customer and the chef. The customer will get hygienic food of his choice at lower prices than the market and the chef can work from their home and earn money and allows them to work whenever they want without any restrictions.

The scope of the project is vast but simple. The customer will be notified about the nearest chef i.e., within a 5 km or 10 km radius. Application will also provide the user to view all the restaurants if they are not satisfied with the recommended kitchen options. The chef will be rated on their cooking and representation skills by the customers.

The Food Delivery system will provide employment opportunities to the people who are delivering food from the kitchen to the customer. This delivery job can result in a part time job or can lead to a full time source of income for the delivery mediator.

The following services are examples of manifestation of our system:

- **Popular food chain delivery systems** such as FoodPanda [1], Cheetay [2] and EatOye! [3]

On the contrary to these services, our platform will be accessible to people with who are physically challenged, away from families or crave from home based cooked food.

- **Goods delivery services**

Uber Eats [4] and Careem Delivery [5] provide delivery of equipment and essential tools for day-to-day business. People rely on these services to manage their business and offices. Using the same principle as these services, our platform delivers food to our customers.

2 Requirements Analysis

Requirements analysis, also referred as requirements engineering, is the process of defining, documenting and maintaining the requirements. It is a gradual process in which gathering and service provided by the system is collected. It is a way through which we can determine user expectations for new product. These features are basically called as requirements. In software engineering, such requirements are called functional specification. Analysis of requirement is important aspect of project management.

2.1 Literature review / Existing system study

As we were proceeding in research, major food delivery platforms start introducing home cooked food as an option on their application. As major players focus upon the commercial supply of the food from restaurants, home based cook food category seems to be overshadowed by it . However, most of the platforms provide ease of access to their customer by providing them their food at door step. Customer feedback is valued and most of time customer preference is kept in mind. Seamless transaction and smooth navigation is offered by most platform which is also a key factor in our application. Our system also implements schemes for moderating consumer response and progress of the customers.

FoodPanda (Android/iOS) Application

FoodPanda[1] has recently turned into a very versatile application which offer numerous services such as restaurant meals, home cooked food and groceries. It shows restaurant with in a certain range from the user and often keep track of your favourite restaurant. Although a tab of “Ghar ka Khana” is available but it's overshadowed by commercial options and user might often miss it while seeking for healthier options.

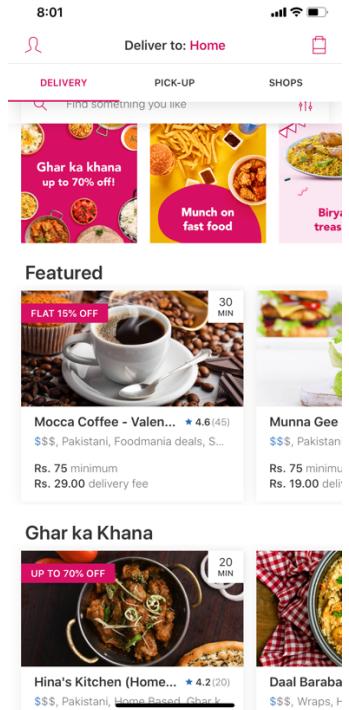


Figure 2. 1 Home Page of Food Panda Application

Cheetay (Android/iOS) Application

This application[2] provide smooth experience to the customer by listing down restaurants and latest offers or discounts to the customers. Card style UI of application make navigation of the app much more easier. Recently Cheetay application had a complete overhaul, in which new services were offered such as delivery of products, groceries' , Home cooked food, restaurant meals and medical supplies. Although the overall experienced is smooth but application seemed to crowded. Home based cook food option is just shown by small “Tiffin” icon which can be easily missed by the user in a hurry. As it said; Jack of all trades, master of none.

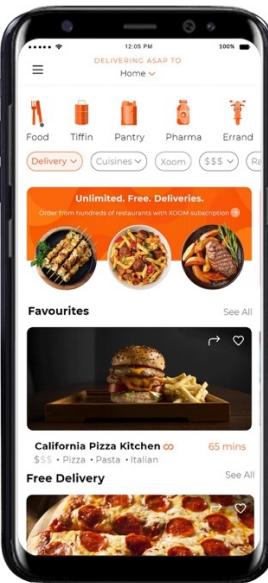


Figure 2. 2 Homepage Cheetay Application

Eat Mubarak (Android/ iOS) Application

Eat Mubarak[8] is locally based application which also provide users option to deliver the food at their door steps. Exclusive offers are given to customers who are using the applications. The app offers fast food, bakery items and home cooked food. Although option of home chefs exist but currently there are no kitchens on the app.

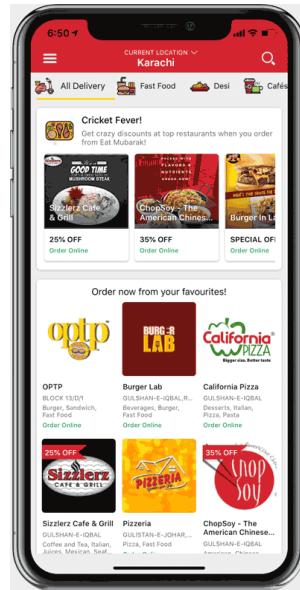


Figure 2.3 Homepage of Eat Mubarak Application

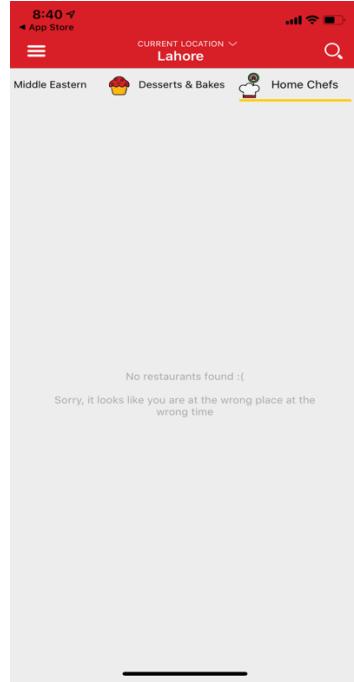


Figure 2.4 Home Chef Page Eat Mubarak

McDelivery (Website/ Android/ iOS)

McDelivery [6] is one of the most used food ordering applications backed up by multinational food chain McDonald's. Recently McDonalds have broken the record of delivering most orders in the current year. This indicates the rise in use of food delivery applications in Pakistan.

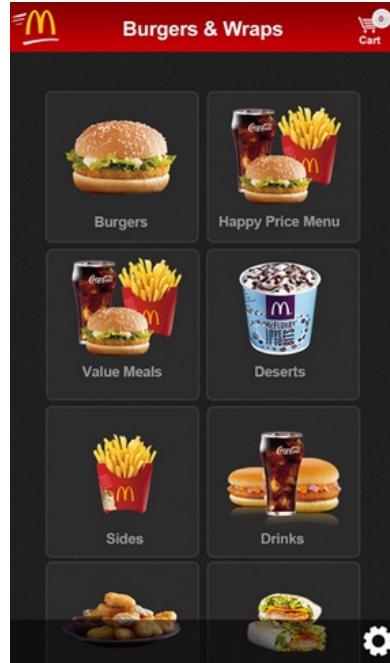


Figure 2.5 Homepage of McDelivery

2.2 Stakeholders list (Actors)

2.2.1 Stakeholders

1. Customer
2. Chef
3. Deliverymen
4. Developers

2.2.2 Product Functions

Admin:

- Login to the site
- Approve or disapprove the request form of the user. There could be many reasons to disapprove user's request i.e. Detection of suspicious activity or incorrect credentials
- Manage orders
- Update vouchers

Customer:

- Login as a user
- Register a new account
- Make orders
- Make payments to either deliveryman or cook
- Provide feedback on the service of deliverymen or cook
- View profiles of other users
- Manage own profile
- Search different cooks within specific radius of an area

Deliveryman:

- Login as a deliveryman

- Register himself with Unique ID and password
- Manage profile and orders
- Receive orders from cook
- Receive extra orders by the customer for better tip

Chef:

- Login as a chef
- Register themselves using unique ID and passwords
- Manage own profile
- Transfer order to deliverymen
- Different menu for regular customers like hostilities
- Updating menu on demand

2.3 Requirements elicitation

2.3.1 Functional requirements

Functional Requirement (FR) is description of the service that system must deliver. They often describe the software system or its module. Function can be summarise as input to the system, its behaviour and the output. In our case Customer use the system to order food from the provided menu, Chef accepts the order and prepares it. Deliveryman deliver the food to the customer from the kitchen. Customer can make payment to the deliveryman or online to the chef.

Table 2. 1 FR01 Functional Requirements of Registering

Req. ID	Functional Requirements
FR01_1	The system allow the user to register themselves to the system

FR01_2	The system should allow user to enter First Name, Last Name, Email, Password and Phone number.
FR01_3	The system should allow to edit the registration fields

Table 2. 2 FR02 Functional Requirements of Login

Req. ID	Functional Requirements
FR02_1	User enters Email and Password to access the system .
FR02_2	If password or email doesn't match the data base credentials then system will prompt an error to retry.
FR02_3	If the user credentials are matched from the data base then user is greeted with home screen of the application
FR02_4	If password is incorrect, email is used to verify the user and reset the password

Table 2. 3 FR03 Functional Requirements for Menu

Req. ID	Functional Requirements
FR03_1	Customers and Chefs can access the menu from Home Screen
FR03_2	Chefs can update, add or delete their menu, on the other hand Customer can only view, select and add to the cart.
FR03_3	Customer can search specific dish from the search bar and add to the cart.

Table 2. 4 FR04 Functional Requirements of Order

Req. ID	Functional Requirements
FR04_1	Customer can select to view kitchen and add desired dish to the cart
FR04_2	Customer can check out from the cart by adding address, payment method and additional note
FR04_3	Customer can select for Online payment or Cash on delivery
FR04_4	After checkout Chef can either accept the order or reject the order.
FR04_5	If order is rejected than chef have to enter a valid reason for cancellation
FR04_6	If order is accepted by chef notification of job is received by deliveryman

Table 2. 5 FR05 Functional Requirements of Notifications

Req. ID	Functional Requirements
FR05_1	Customer would receive notification of latest menu updates or offers around there area
FR05_2	Customer shall receive notification of order status such as received, accepted, On way or Delivered.
FR05_3	Deliverymen should be notify about the available delivery job around there area
FR-05_4	Chef should be notify about the latest rating on their delivered meal

FR-05_5	Owners/ Administrators should receive notification of any new complaint launched against their order
----------------	--

Table 2. 6 FR06 Functional Requirements of Profile Management

Req. ID	Functional Requirements
FR06_1	Customers can update their address, first name , last name
FR06_2	Chef can update their profile photo and Menu images
FR06_3	Profiles can be linked to social media or accounts such as Gmail, Apple Account etc

Table 2. 7 FR07 Functional Requirements of Feedback System

Req. ID	Functional Requirements
FR07_1	Customers can provide rating to the chef or their dishes.
FR07_2	Ratings are visible to every user logged in to the system
FR07_3	Customers are allowed to select chef or dishes on basis of their ratings.

2.3.2 Non-functional requirements

Nonfunctional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as restrictions on the design of the system across the different uncompleted works. Here are described the non-functional requirements that must be met for efficient and effective operations of the app.

Table 2. 8 NFR02 Non-Functional Requirements of Interface Design

Req. ID	Non-Functional Requirements
NFR02_1	The interface design to be receptive to the show screen.
NFR02_2	The interface configuration should take after ease of use assessment and testing procedures.
NFR02_3	The interface ought to be outlined considering the heuristic standards for UI configuration, (Know The client, consistency, negligible plan, blunder aversion, perceivability of framework status, client control and opportunity and so on.)

Table 2. 9 NFR02 Non-Functional Requirements of Performance

Req. ID	Non- Functional Requirements
NFR02_1	The system should be react convenient according to user search.
NFR02_2	Different database for employees
NFR02_3	real time information can be transferred through wireless connection

Table 2. 10NFR03 Non-Functional Requirements of Reliability

Req. ID	Non-Functional Requirements
NFR03_1	User's data would be backed up and keep safe.

NFR03_2	User's online transactions would be safe and reliable.
NFR03_3	Application should be reliable and free of sudden crashes.

Table 2. 11 NFR04 Non-Functional Requirements of Maintainability

Req. ID	Non-Functional Requirements
NFR04_1	The system should be stable during peak traffic hours.
NFR04_2	The system should support a peak transaction rate
NFR04_3	System should be effective or able to maintain integrity.

Table 2. 12 NFR05 Non-Functional Requirements of Security

Req. ID	Non-Functional Requirements
NFR05_1	Private information of users should be protected.
NFR05_2	The system must be able to detect and deal with user who abuse the system
NFR05_3	The online transactions would be secure and keep private

2.4 Use case descriptions

2.4.1 Customer Meal Ordering

Table 2. 13 Use Case Description 1- Customer

ID	UseCase_1
Title	Order Food
Description	User has access to the menu through which meal can be selected and added to the cart. After user checks out, food Is delivered to specific location.
Primary Actor	Client
Pre-conditions	<ol style="list-style-type: none"> 1. Client is logged into the Application System. 2. Client must allow location services to the system.
Post-conditions	<ol style="list-style-type: none"> 1. Meal is added to the cart 2. Menu is updated once client checks out.
Basic Flow of Events	<p>1.0 Order a Single Meal</p> <ol style="list-style-type: none"> 1. Client request the system to show the latest updated menu. 2. The Application system fetch the last updated menu from the data base. 3. Client selects the desired meal 4. Meal is added to the client's cart. 5. Client checks out from the cart confirming the order request. 6. Payment options are selected by the client. 7. Client is notified with confirmation of his/her order.
Exceptional Flow	<ol style="list-style-type: none"> 1. Adding Discount code 2. Selecting Multiple Meals 3. Special Deals and Offers.

2.4.2 Chef Menu Update

Table 2. 14 Use Case Description 2- Chef

ID	UseCase_2
Title	Menu Update
Description	System provide control to the Chef that they can update their menu on regular basis. Options such as editing prices, quantity, discount offers , special deals , coming soon and out of stock labels can be applied on the menu. Changes are viewed by customers once database is updated.
Primary Actor	Chef
Pre-conditions	<ol style="list-style-type: none"> 1. Approved account already exist 2. Kitchen verified by the administrators.
Post-conditions	Update/Changes to menu are saved.
Basic Flow of Events	<ol style="list-style-type: none"> 1. Chef request the system to see the menu 2. System display last edited menu to the chef. 3. Chef updates his menu for the day. 4. Chef save the changes that are made 5. Changes are saved to the database while showing success message to the chef.
Exceptional Flow	<ol style="list-style-type: none"> 1. No changes made to the menu for specific date. 2. System display reminder to chef to update menu every 3 days.

2.4.3 Kitchen Registration.

Table 2. 15 Use Case Description 3- Chef Registration

ID	UseCase_3
-----------	------------------

Title	Register Kitchen.
Description	Chef can apply to open its kitchen and request the system for approving the request.
Primary Actor	Chef
Pre-conditions	<ol style="list-style-type: none"> 1. Chef should have an email verified account 2. Chef should be logged in to the system
Post-conditions	Kitchen registration is successful.
Basic Flow of Events	<ol style="list-style-type: none"> 1. Chef request the system to create new kitchen 2. System displays registration form for the request 3. Chef supplies the necessary information and submit 4. Chef request is submitted to the database and becomes pending for administrator approval. 5. Chef is notified when kitchen is approved by the admin.
Exceptional Flow	<ol style="list-style-type: none"> 1. Information provided by the chef are not correct 2. Request for kitchen registration is denied by the administrator

2.4.4 Delivery-Man Job

Table 2. 16 Use Case Description 4- Delivery-Man

ID	UseCase_4
Title	Order Pick Up

Description	Delivery Man is notified when a job is available.
Primary Actor	Delivery-Man
Pre-conditions	<ol style="list-style-type: none"> 1. Delivery Man should be logged in to the system 2. Status should be online to show the availability
Post-conditions	<ol style="list-style-type: none"> 1. Notification of successful delivery 2. Rating given by the client on delivery service 3. Amount earned from the job earned is notified.
Basic Flow of Events	<ol style="list-style-type: none"> 1. Delivery Man receives the notification of available job 2. System assigns the delivery man as he/she accepts the job 3. Delivery Man pick up the order from kitchen 4. Food is delivered to the client 5. Delivery Man update the job as complete.
Exceptional Flow	<ol style="list-style-type: none"> 1. Delivery Man doesn't take job in spite of being online on the system 2. Job is not completed due to unforeseen circumstances

2.4.5 User Registration

Table 2. 17 Use Case Description 5- Guest

ID	UseCase_5
-----------	------------------

Title	Registration
Description	A new visitor can request for registration
Primary Actor	Guest
Pre-conditions	Account already Exist
Post-conditions	Registered successfully to the system
Basic Flow of Events	<ol style="list-style-type: none"> 1. Request for registration 2. System accept the request for registration 3. Request form is shown to be filled by the guest 4. Email is verified by sending a code to Guest email 5. Guest Enters the code send through his email. 6. Account is successfully registered in the system.
Exceptional Flow	<ol style="list-style-type: none"> 1. Email already exist in database. 2. Email is previously banned on system by administrator.

2.4.6 Delivery Man Registration

Table 2. 18 Use Case Description 6 Delivery Man Reg

ID	UseCase_6
Title	Delivery Man Registration
Description	Deliveryman can request the system for delivery account
Primary Actor	User
Pre-conditions	<ol style="list-style-type: none"> 1. Account already Exist 2. User Should be logged in

Post-conditions	Registered as delivery man is successful
Basic Flow of Events	<ol style="list-style-type: none"> 1. User Request system for delivery man registration. 2. System display required form 3. User enter verification details and information 4. Request is submitted to the data base and become pending for admin approval 5. User is notified when request is approved by admin
Exceptional Flow	<ol style="list-style-type: none"> 1. Request user CNIC or vehicle registration number is already linked to other account 2. User with similar detail is banned from the system 3. Registration request denied by the administrator

2.4.7 Administrator

Table 2. 19 Use Case Description 7- Admin

ID	UseCase_7
Title	Administrator
Description	Administrator can approve, deny request of registration from chef, Customer and Deliverymen. Admin can also suspend , ban or kick out any user from the system due to breach of terms and conditions. Admin can also push notifications and suspend system due to maintenance. Admin can view activity feed, sale, order and number of user reports on specific timeline.
Primary Actor	Administrator
Pre-conditions	Verified System account

Post-conditions	Update to the database and system is successful
Basic Flow of Events	<ol style="list-style-type: none"> 1. Admin request to login into admin panel 2. Admin can approve registration request from chefs 3. Admin can respond to complaint by the users 4. Push new notifications to the users 5. Submit changes to database
Exceptional Flow	<ol style="list-style-type: none"> 1. Database not updated due to technical issue 2. Admin is removed from the panel by the system owner

2.5 Use case design

2.5.1 Diagram 1

This use case diagram is for customer who can make an order or pay for that order.

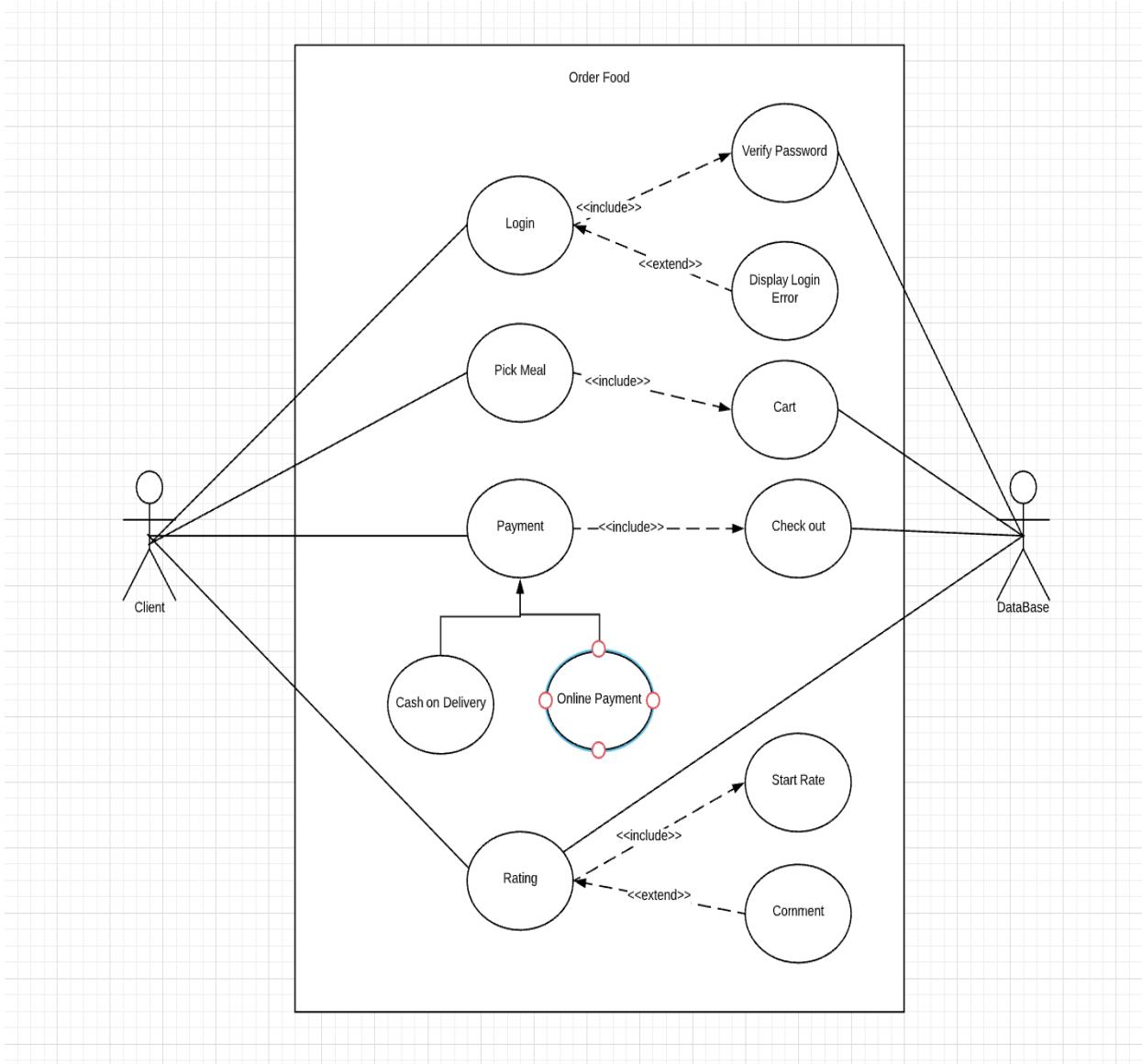


Figure 2. 6 Use Case Diagram for Customer Ordering Meal

2.5.2 Diagram 2

This use case diagram is for Delivery-Man who can perform the following functionalities.

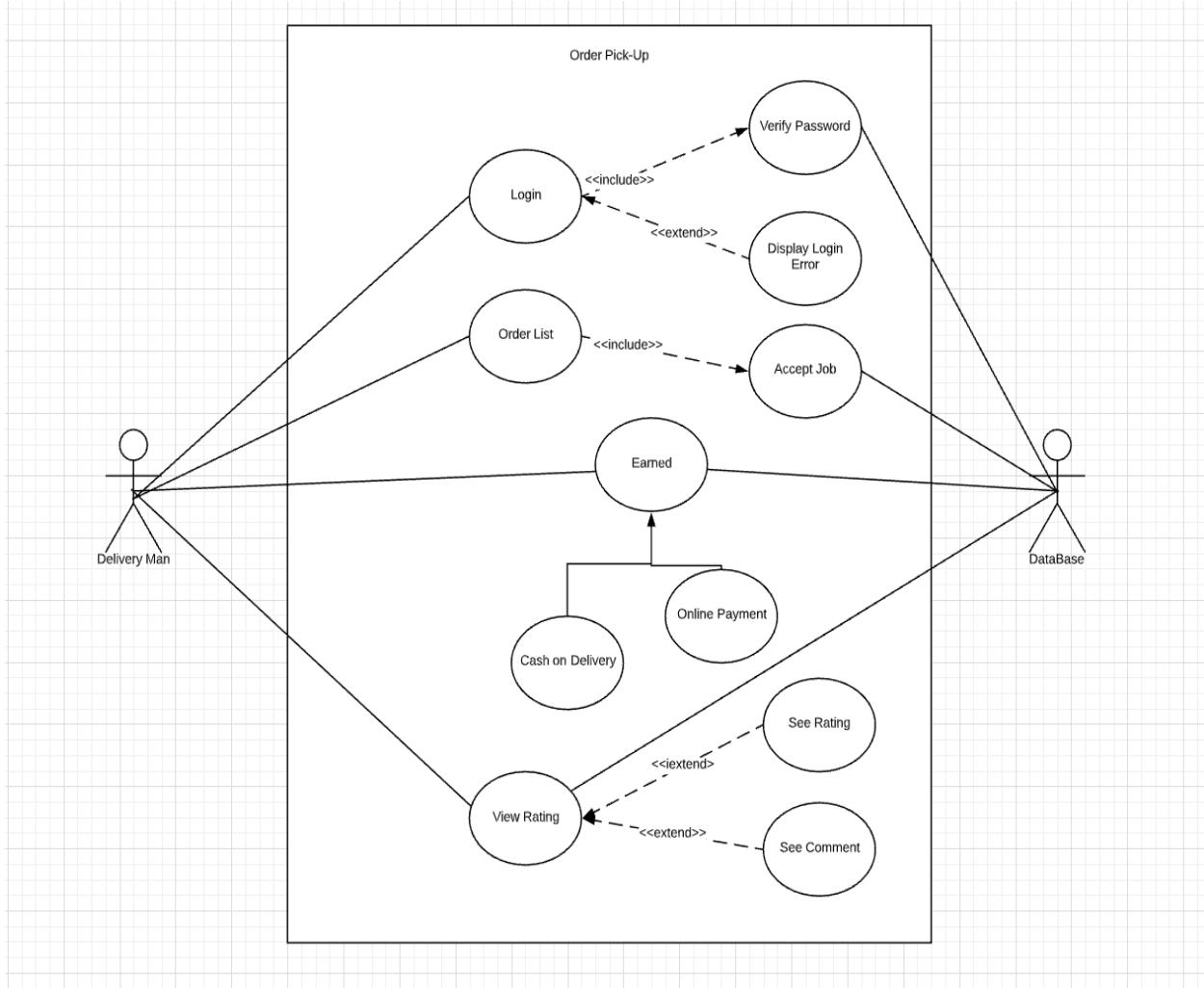


Figure 2. 7 Use case diagram for Deliveryman

2.5.3 Diagram 3

This use case diagram is for Chef who can update the menu and notify the orders to Delivery-Man.

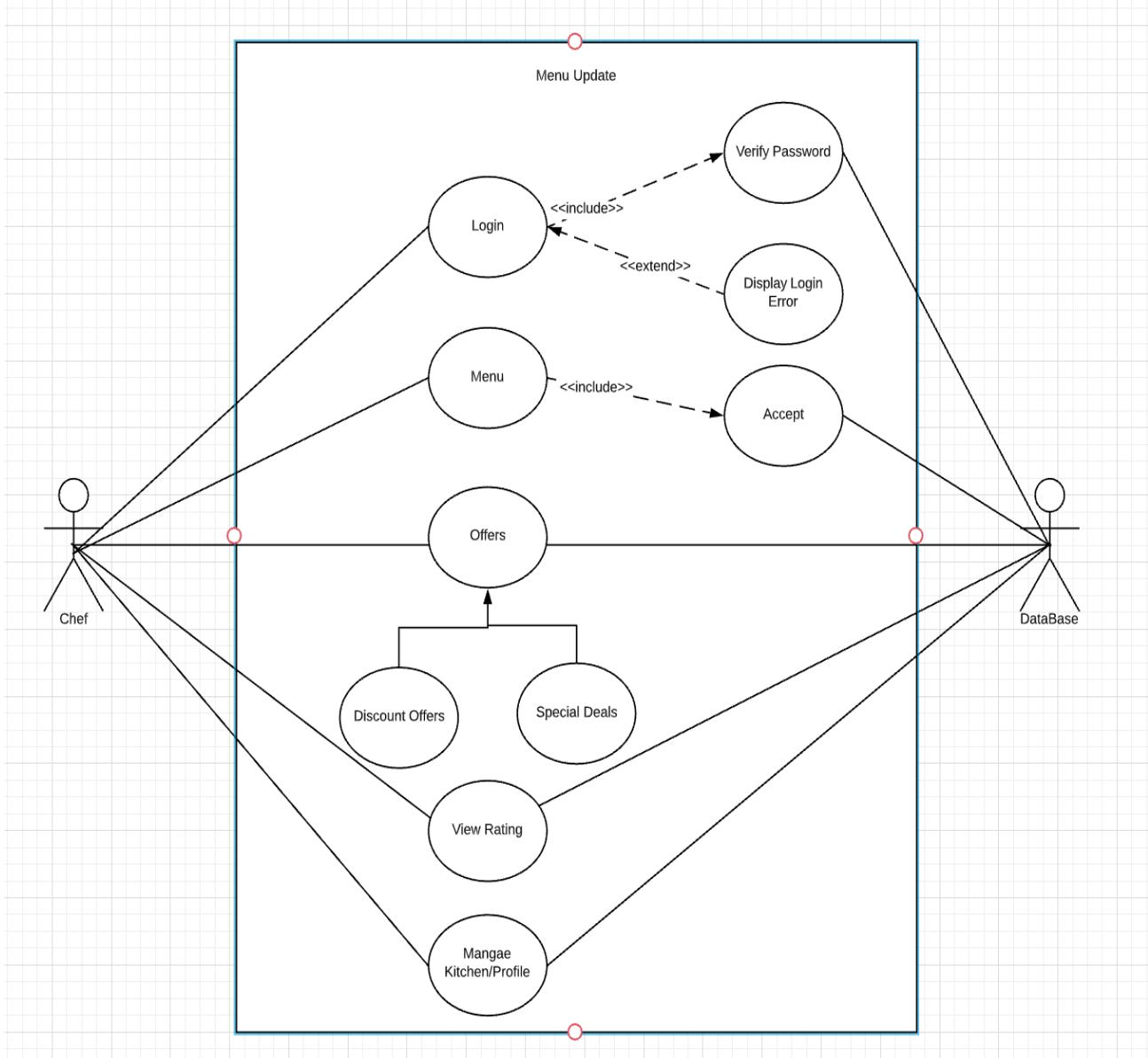


Figure 2. 8 User case diagram for Chef

2.5.4 Diagram 4

This use case diagram is for how chef can register a kitchen.

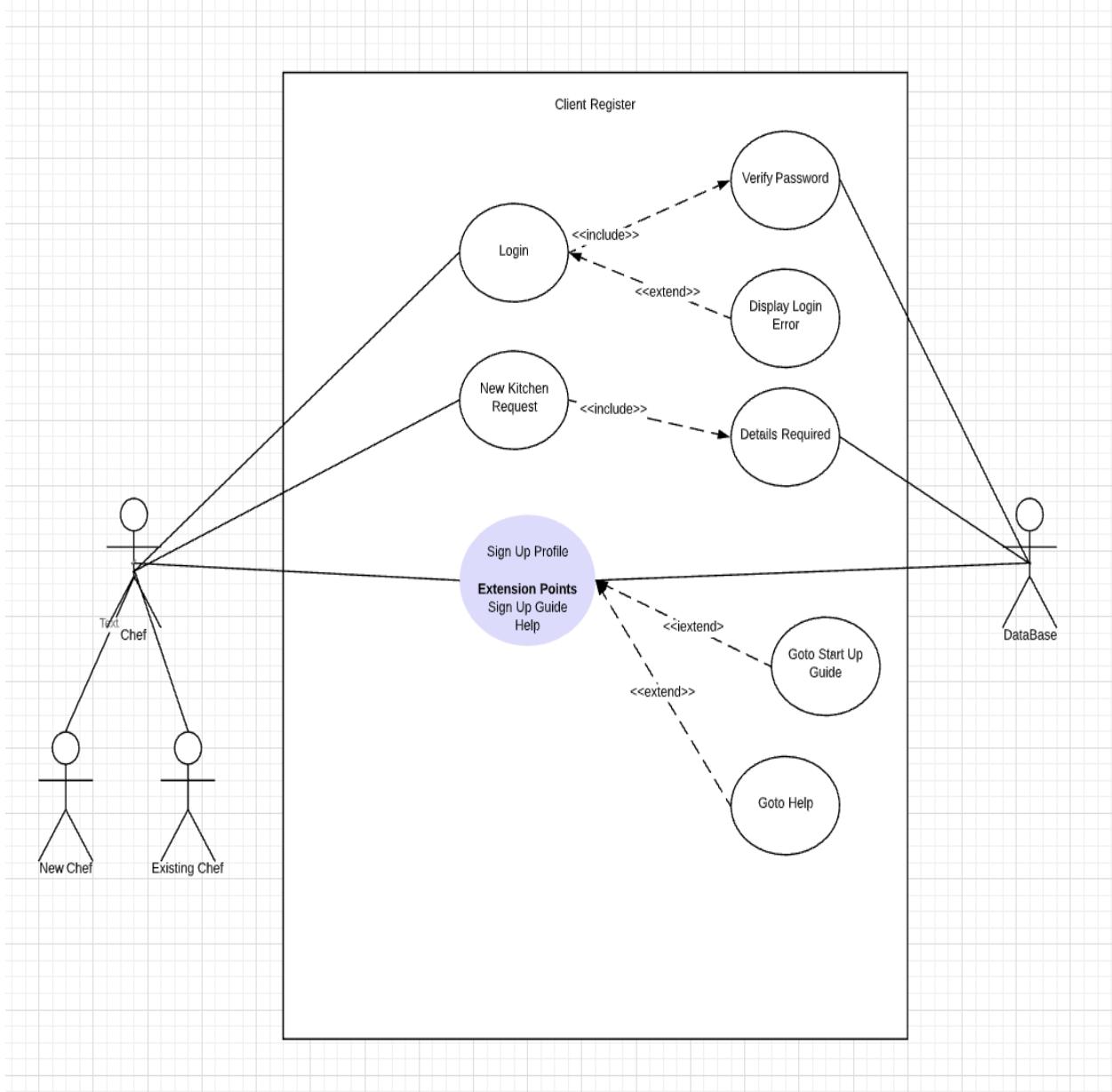


Figure 2. 9 Use case diagram for Kitchen Registration

2.5.5 Diagram 5

This use case diagram is for Guest who can register the application

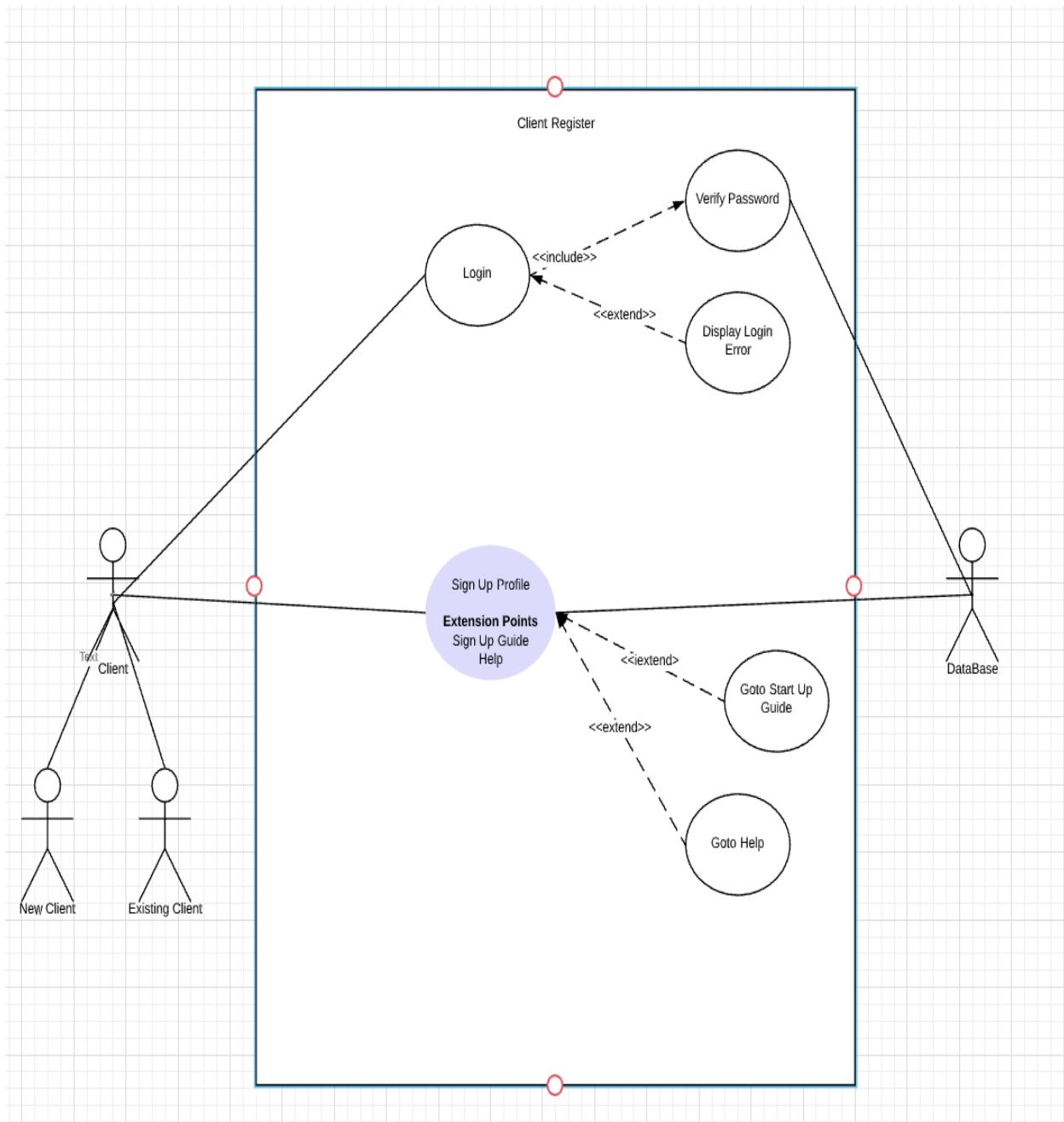


Figure 2. 10 Use case diagram for Client Registration

2.5.6 Diagram 6

This use case diagram is for Delivery Man who can register the application

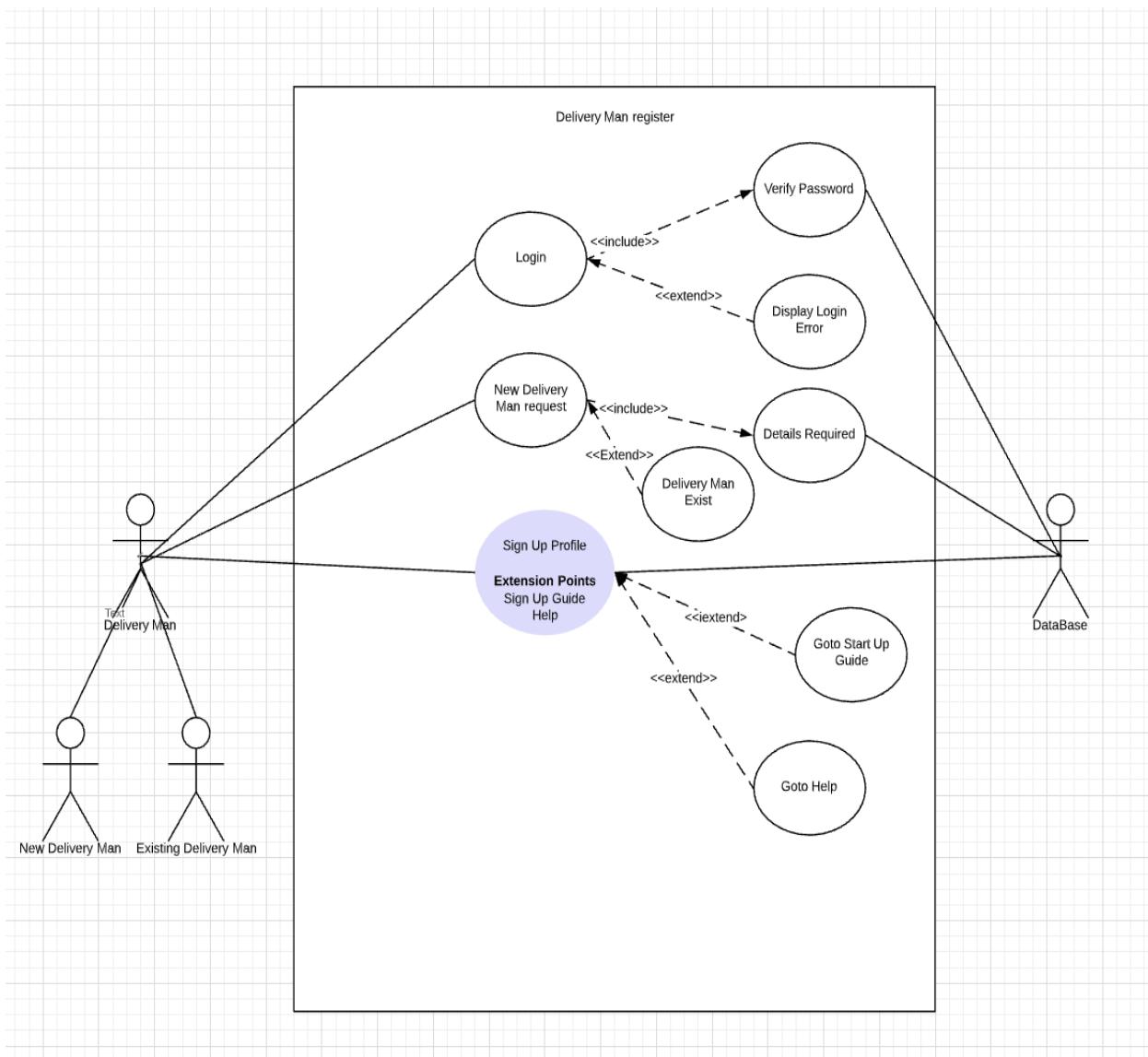


Figure 2. 11 Use case diagram for Delivery Man Registration

2.5.7 Diagram 7

This use case diagram is for Admin interaction with Admin Panel

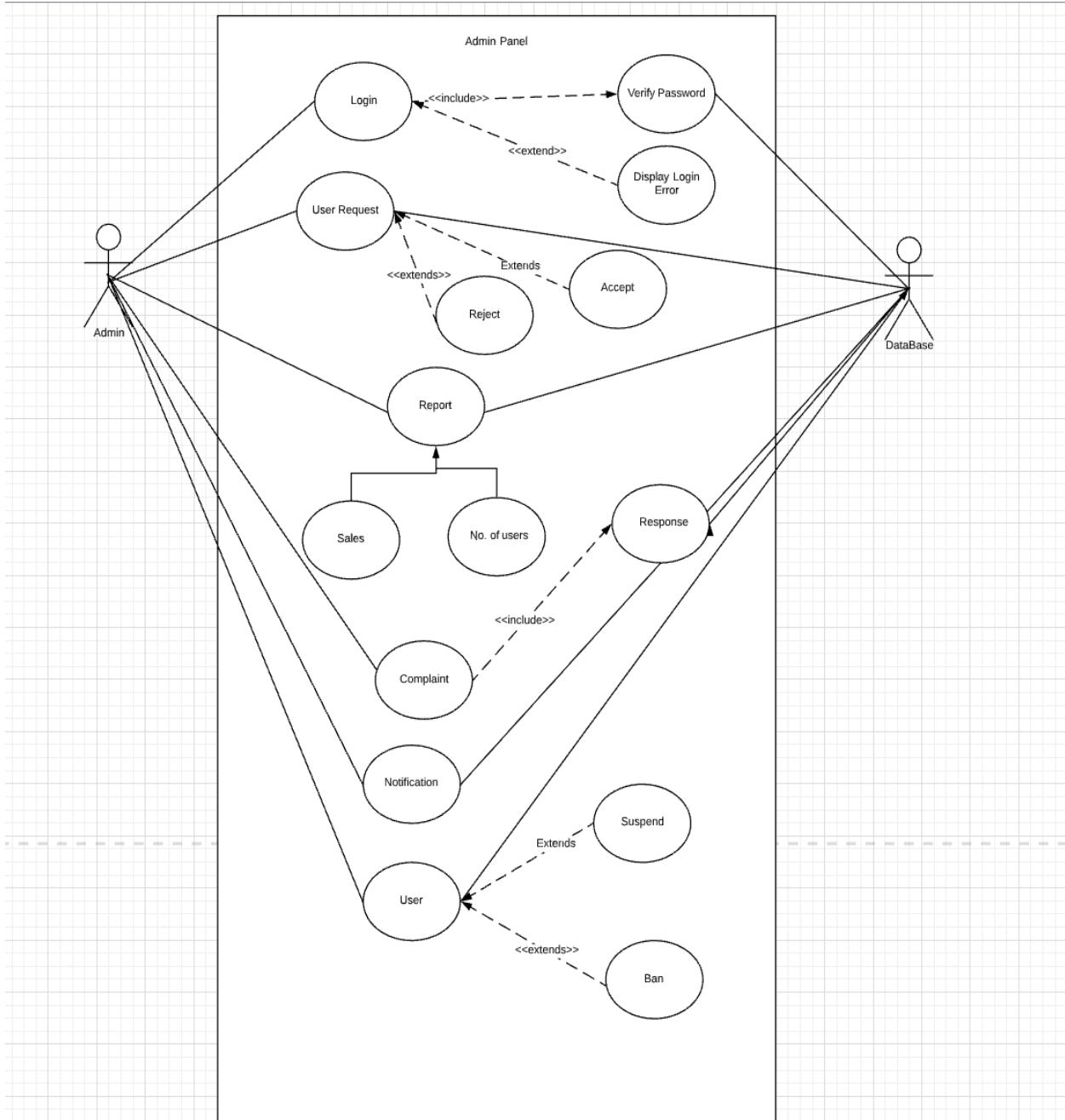


Figure 2. 12 Use case diagram for Client Registration

2.6 Software development life cycle model (justification on why this particular model is considered)

The system development life cycle also referred to as application development life cycle, is used for planning, designing, developing and testing an information system.

Selected model:

We have used **Agile[8] process model**, a software development approach based on iterative development. Agile methods break tasks into smaller iterations. The project scope and requirements are finalized at the beginning of the development process. Number of iterations, the duration and the scope of each iteration are clearly defined in advance.

Reasons:

Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The entire project being divided into smaller parts helps to eliminate the project risk and to reduce the overall project delivery time. Each iteration involves a team working through a full software development life cycle (SDLC) including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client. After releasing the product feedback is the last phase out of six phases of this efficient process model.

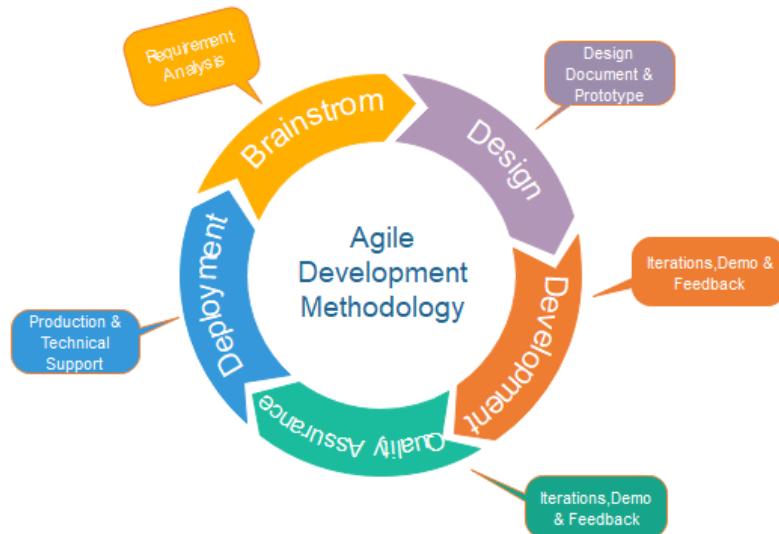


Fig. Agile Model

Figure 2. 13 Framework of Agile model with each phase of SDLC shown

Rejected models:

We didn't deploy Waterfall or Prototype methods mainly because our product would be developed in small task that makes it easier and faster to solve the issues related to the development. Waterfall has main disadvantage like once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the initial concept stage. On the other hand, prototype methodology may increase the **complexity** of the system as scope of the system may expand beyond original plans. These methodologies would require us to work on a long term and one time development of the project, which would be inefficient and difficult, therefore, we resorted to **Agile process models** to preserve the requirements and other preliminary parameters of the product.

3 System Design

System design describes the major tasks that develop the architecture, components, modules, interfaces, and data for a system that satisfies specified requirements. Systems design could be seen as the application of systems theory to product development

3.1 Work breakdown structure (WBS)

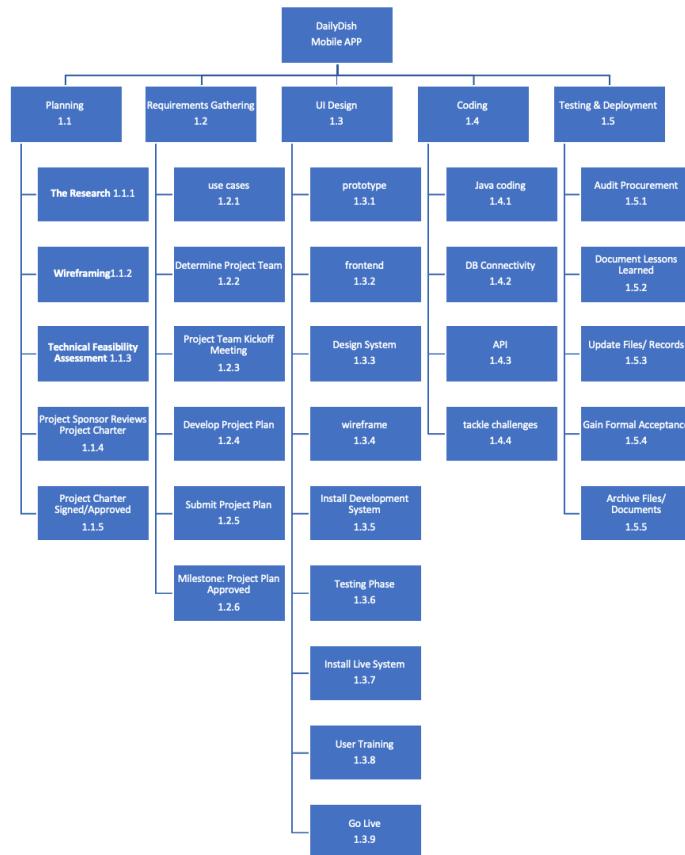


Figure 3. 1 Work Breakdown Structure Representing various functions and components of product

3.2 Activity diagram

3.2.1 Diagram 1

The flow of the activity diagram is showing that every user has to login for the access of the application. User will enter the username and password for login but if they are wrong user have to try again to get access to the applications.

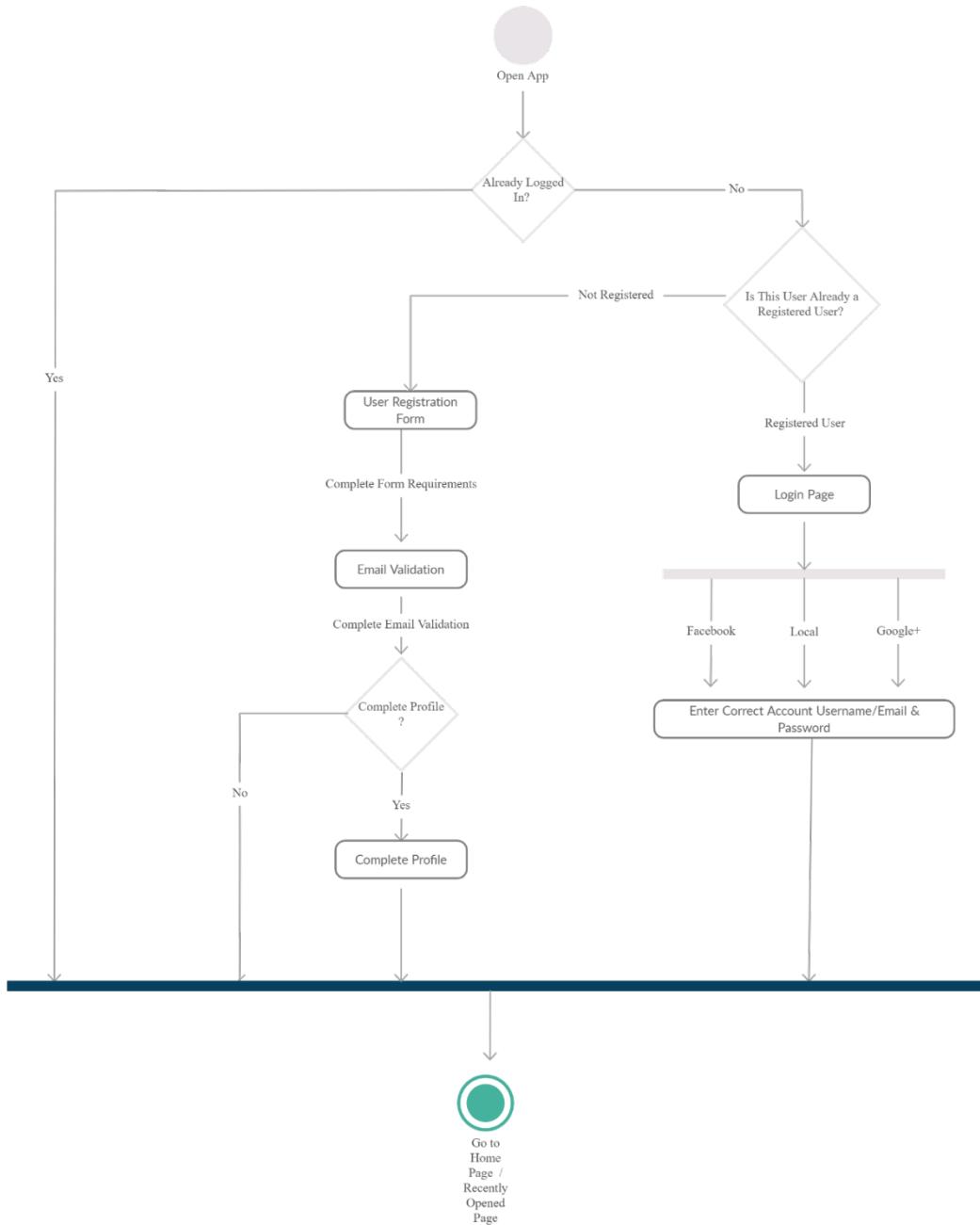


Figure 3. 2 Activity Diagram showing login process

3.2.2 Diagram 2

The flow of the activity diagram is showing that customer can enter into a system to manage his profile, make an order and pay the dues of order. Customer can also feedback to the following order.

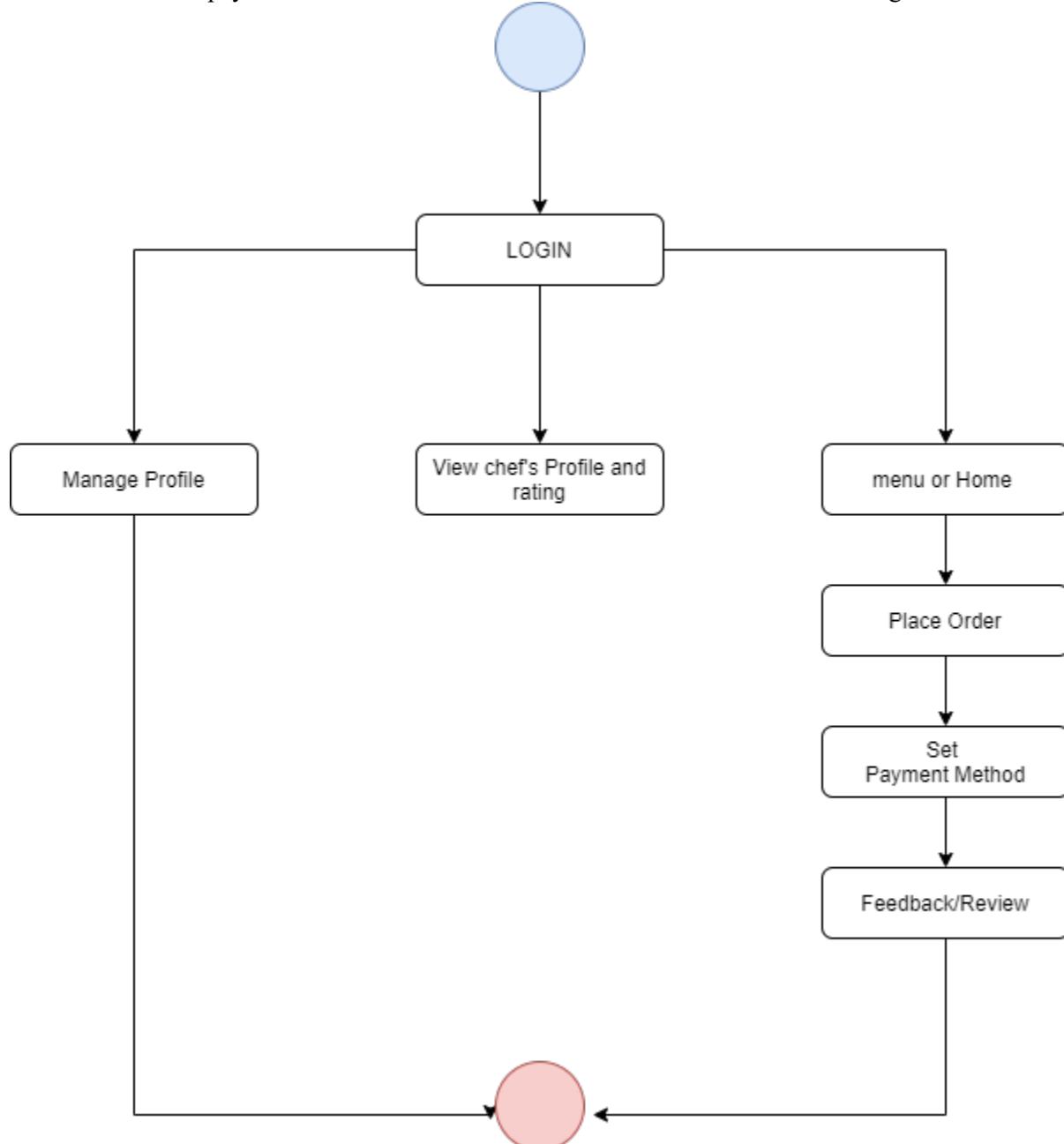


Figure 3. 3 Activity diagram showing functions of Customer

3.2.3 Diagram 3

The flow of the activity diagram is showing that Cook can update the menu list of food items and also can change the price of the foods. Cook can receive payments or manage the profile by logging into the account.

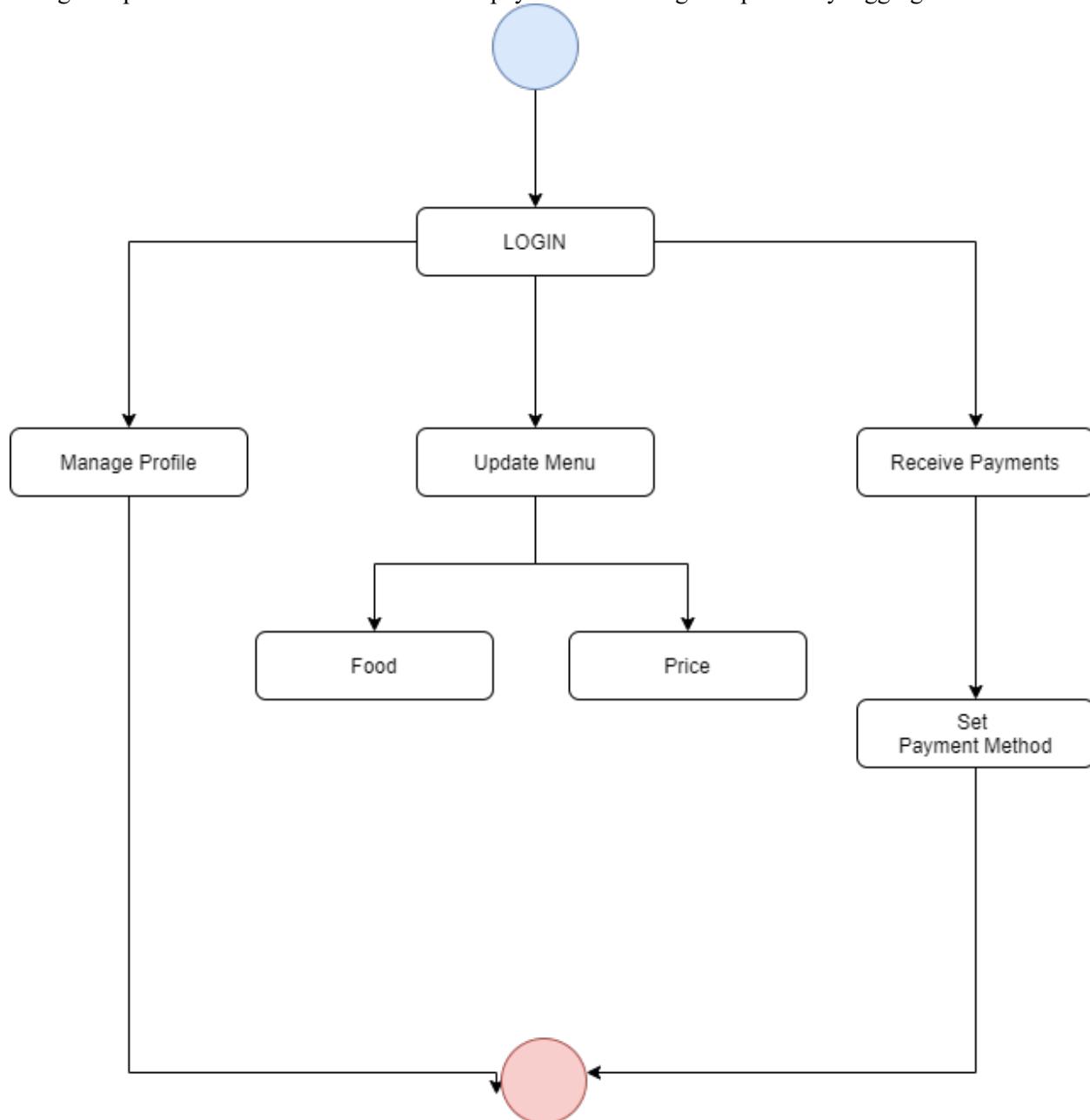


Figure 3. 4 Activity diagram showing functions of the Chef

3.2.4 Diagram 4

The flow of the activity diagram is showing that Deliveryman can login, manage profile and see the pickups list.

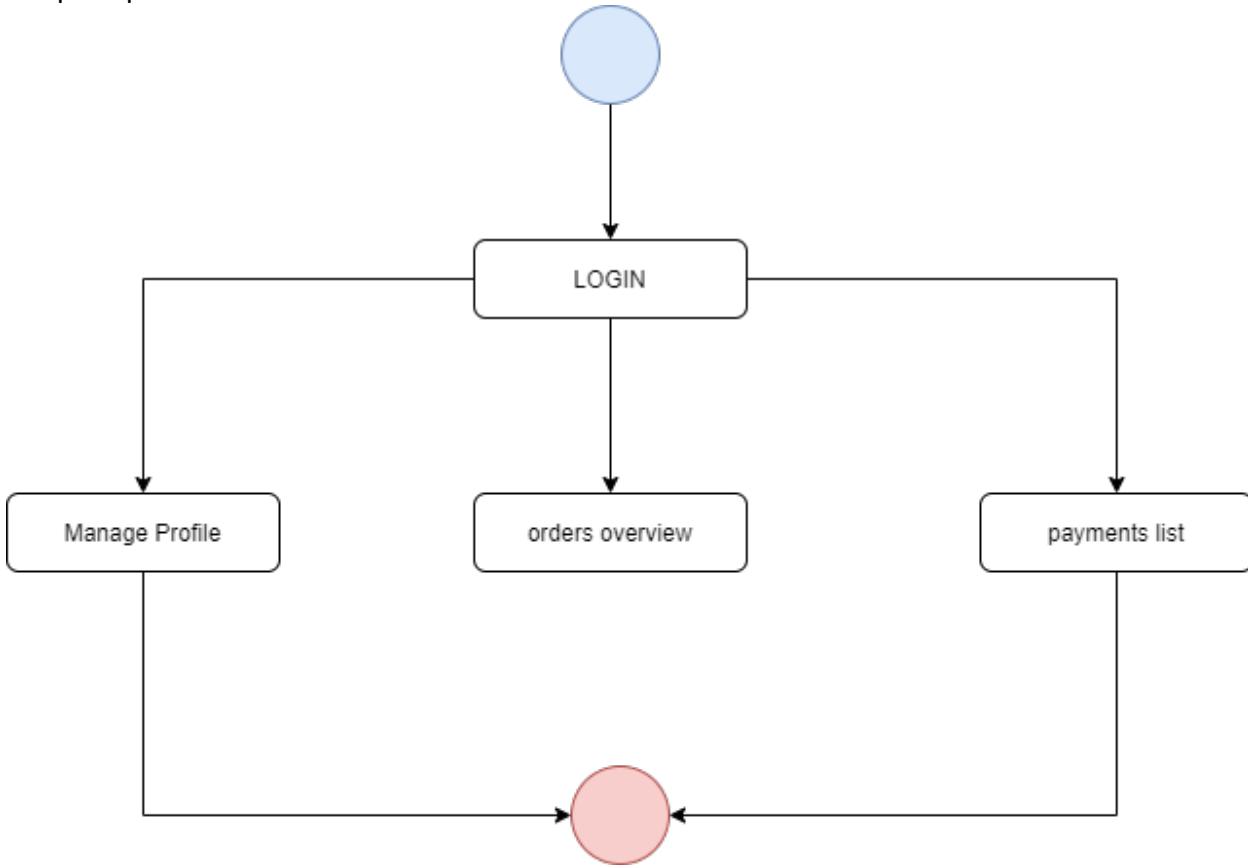


Figure 3. 5 Activity diagram showing functions of the Deliveryman

3.2.5 Diagram 5

The flow of the activity diagram is showing that a Guest don't have to Login. Guest can review all the fields and register an account. (For System Use)

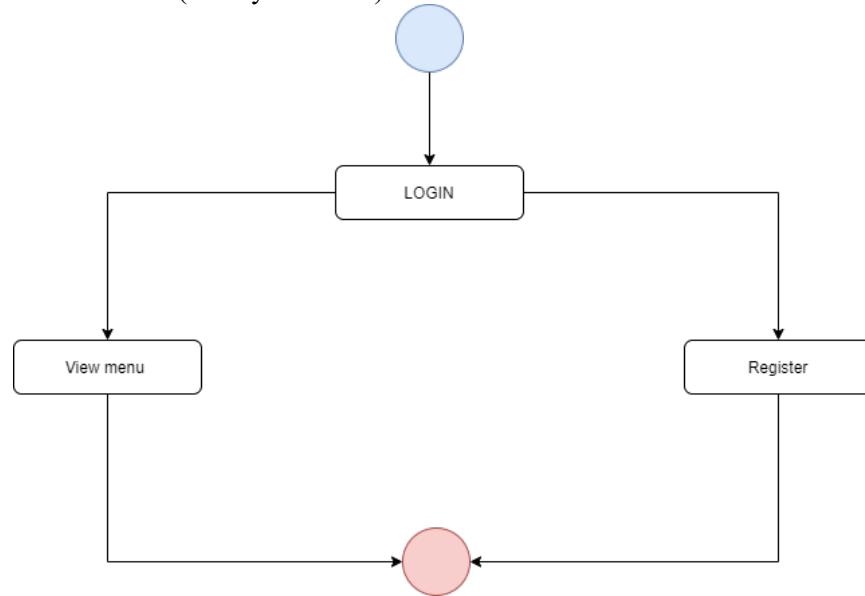


Figure 3. 6 Activity diagram new use

3.3 Sequence diagram

The diagram below shows the interactions, transactions, and relationship between the users, chefs, and deliverymen, and how customer could place the order in the application and make payments.

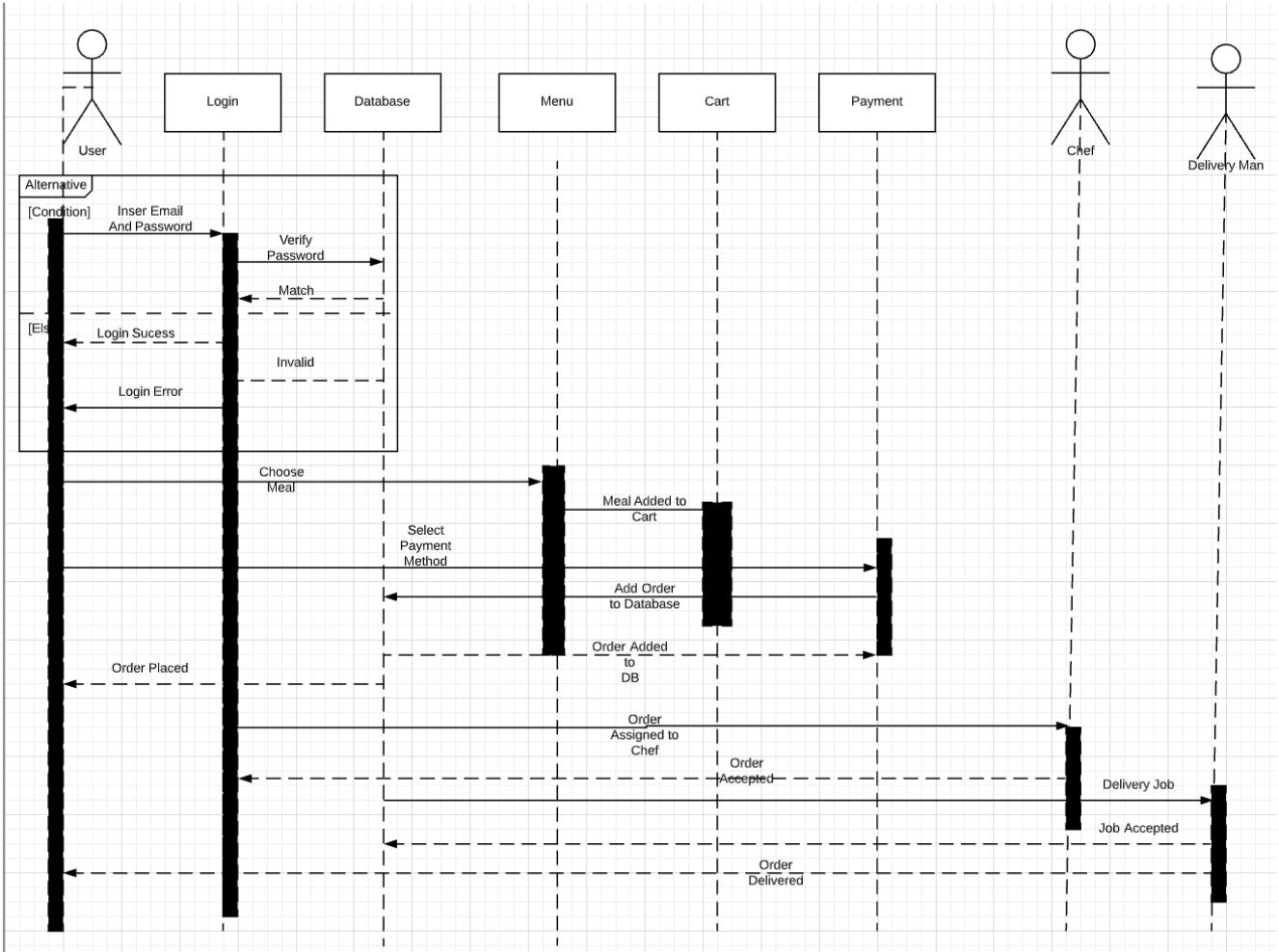


Figure 3.7 Sequence diagram representing relationship between Customer, Chef, and Deliveryman.

3.4 Software architecture

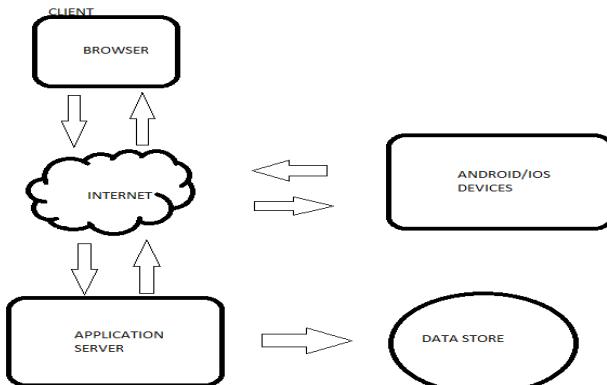


Figure 3. 8 Brief representation of Software architecture

3.5 Class diagram

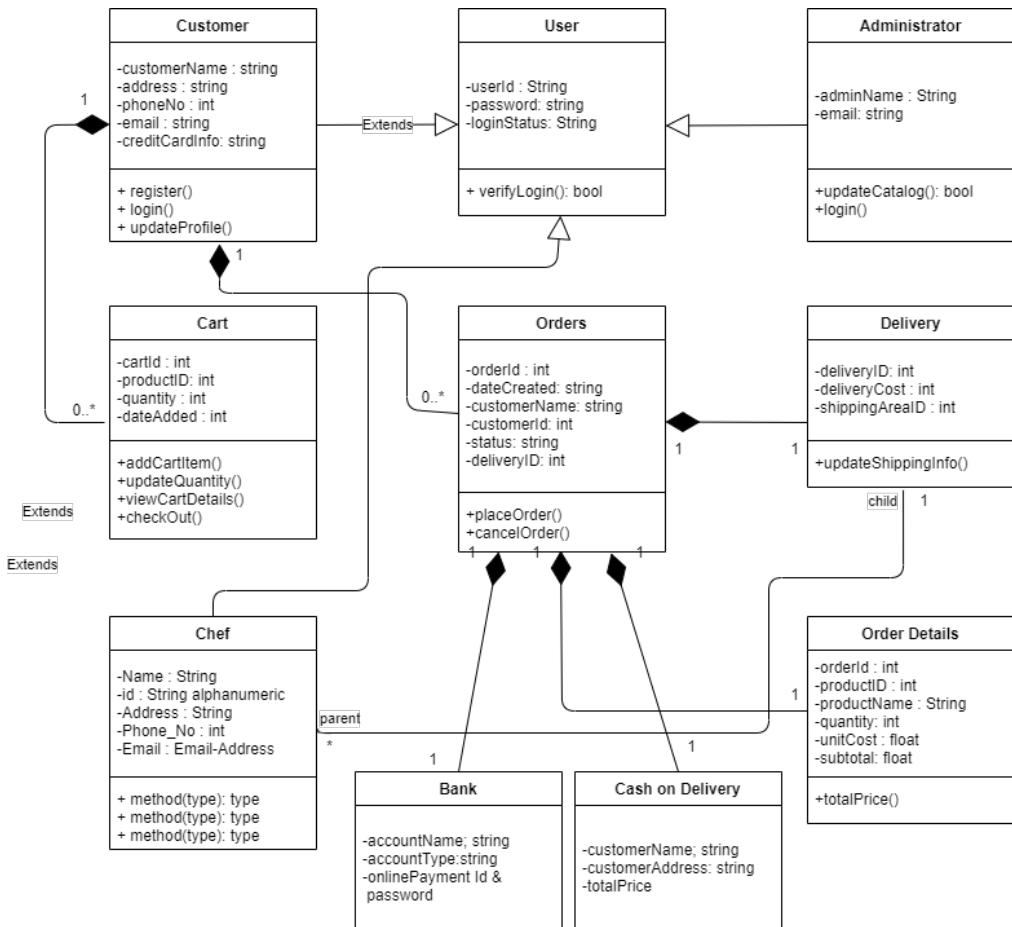


Figure 3. 9 Class diagram for the project

3.6 Database diagram

Entity Relationship Diagram (ERD) designed below shows the associations among the entities/modules of the system along with their attributes.

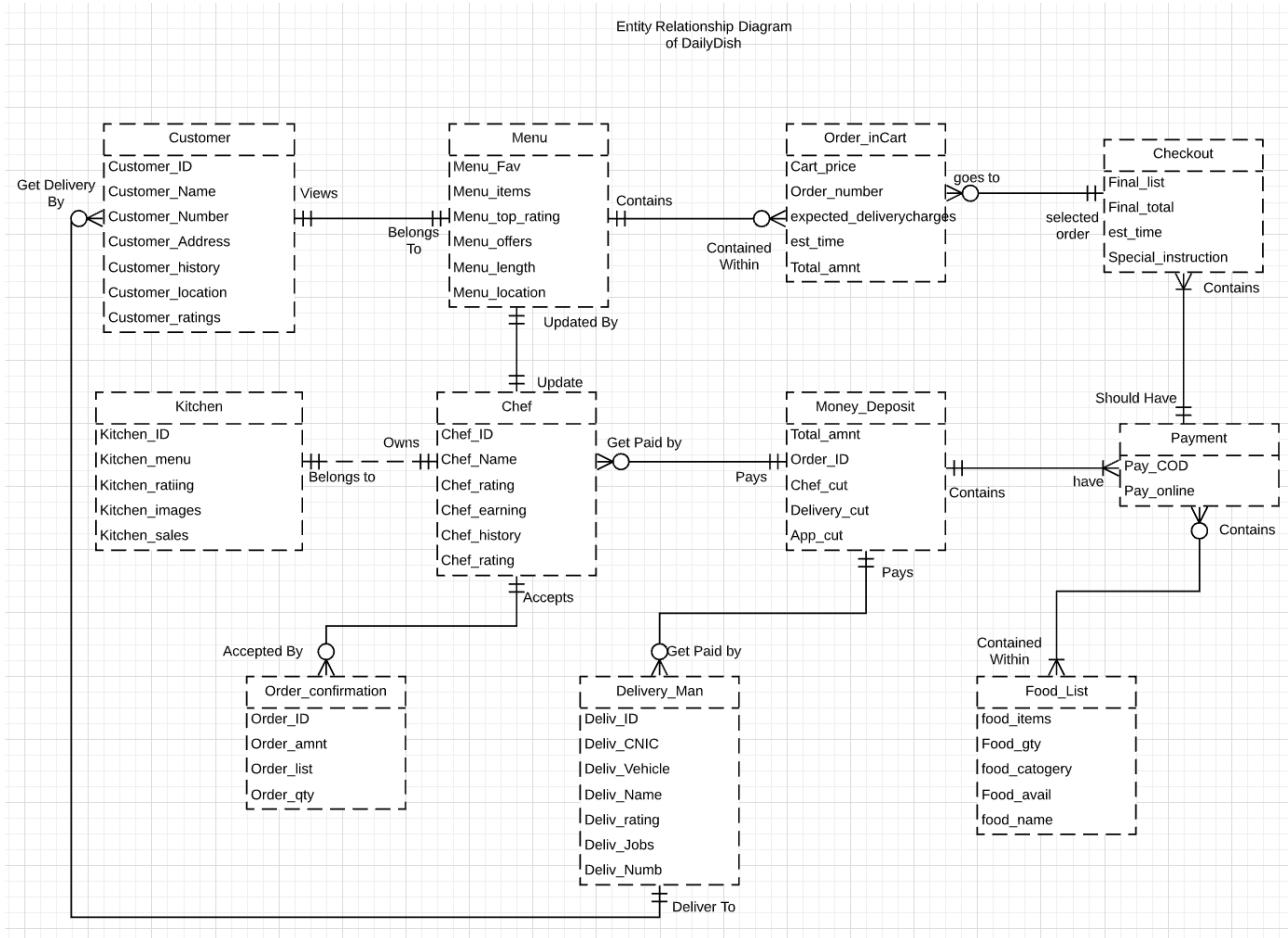


Figure 3. 10 ERD illustrating the relationship between the entities in the project

3.7 Network diagram (Gantt chart)

It illustrates the timeline of the project alongside the timetable of the major activities involved in the project.

ACTIVITIES	FEB-APR	MAY-JUL	AUG-OCT	NOV-DEC
RESEARCH WORK	Black	White	White	White
UI/UX DESIGN	White	Black	White	White
API/ Database	White	White	Black	White
Backend Development	White	White	Black	White
Testing	White	White	White	Black
Final Touches / Optimization	White	White	White	Black

Figure 3. 11 Network Diagram presenting the project schedule

3.8 Prototyping

3.8.1 Diagram 1

The homepage is shown here.

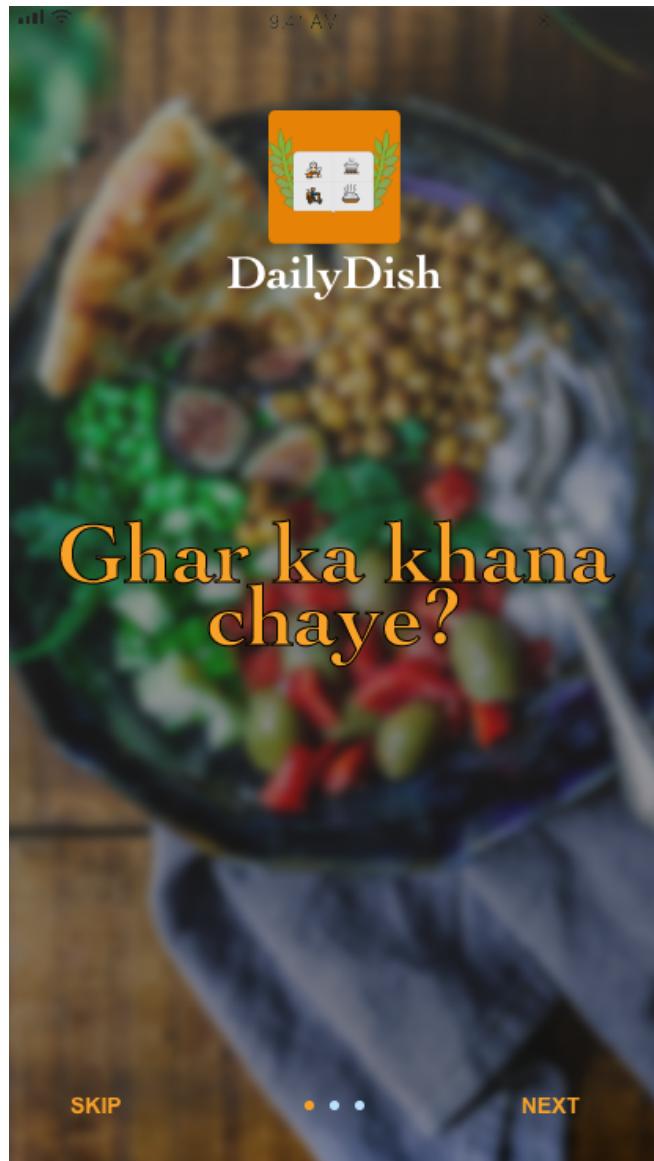


Figure 3. 12 Homepage

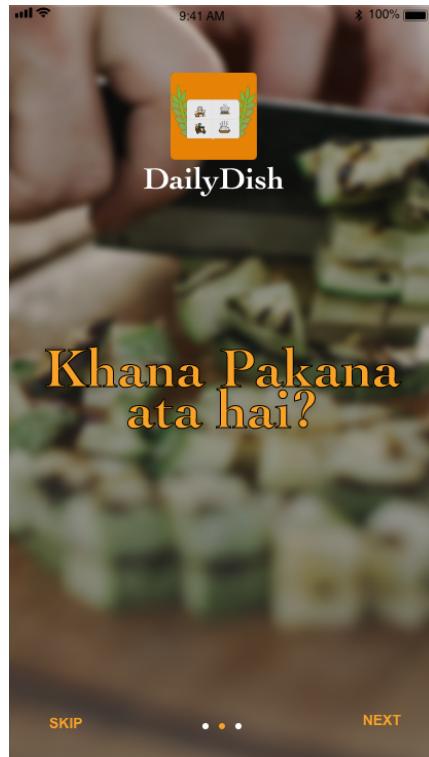


Figure 3. 13 Homepage2

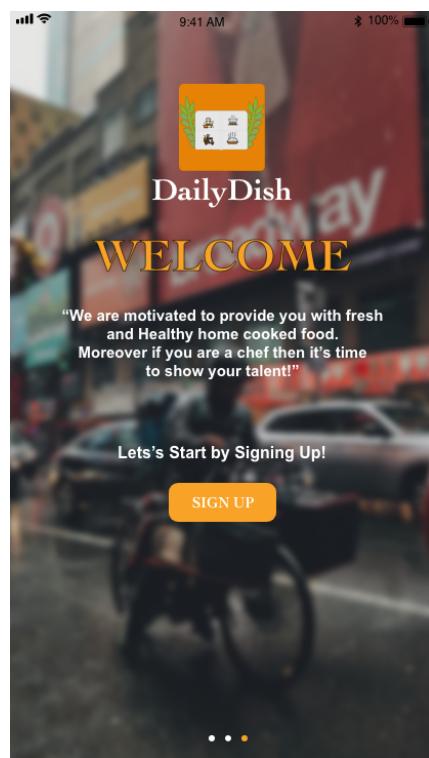


Figure 3. 14 Homepage3

3.8.2 Diagram 2

Profile Selection.

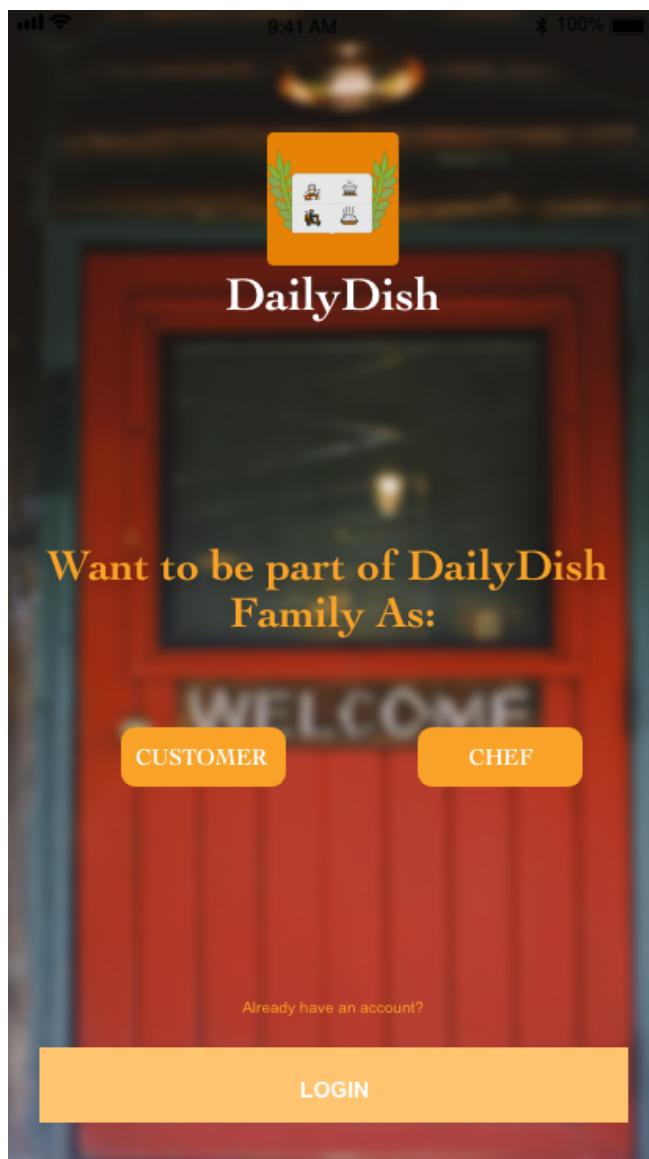


Figure 3. 15 Profile Selection

3.8.3 Diagram 3

The diagram below shows the user registration form.

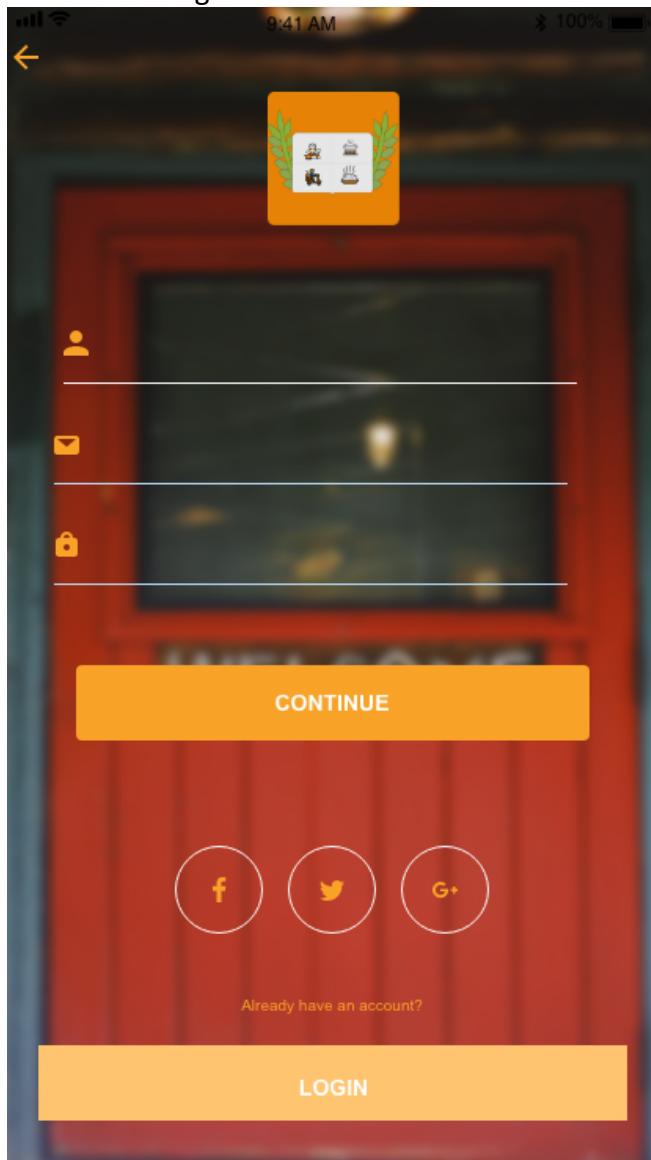


Figure 3. 16 Registration page

3.8.4 Diagram 4

Users can access their profiles after logging in to their accounts, and edit them as per their preferences.

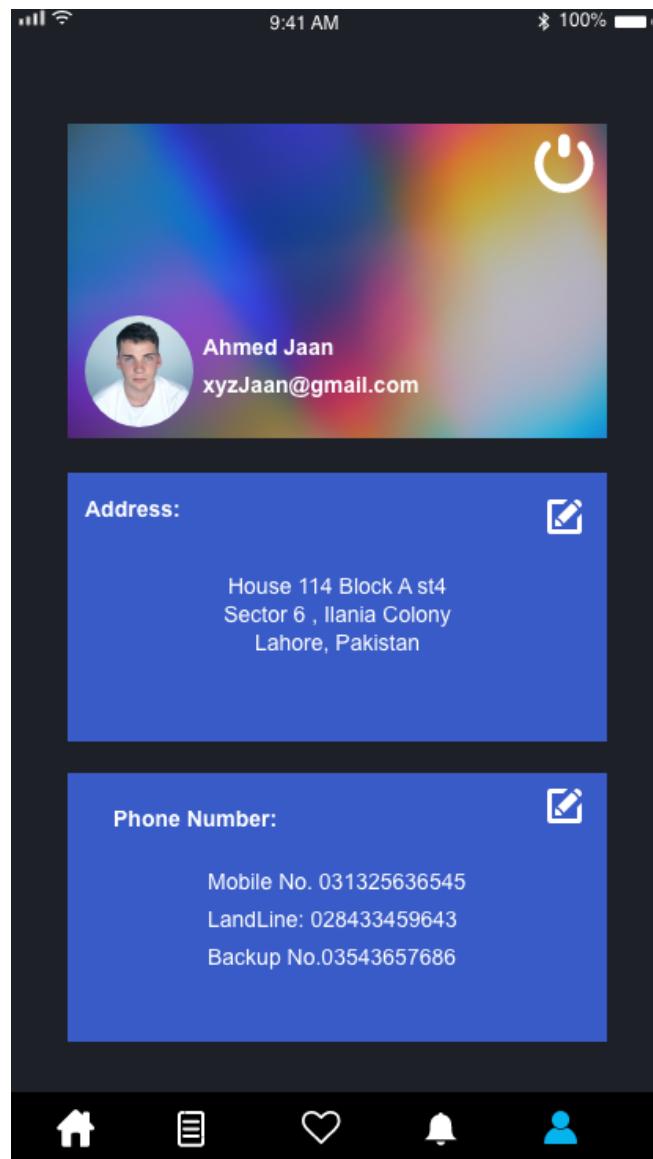


Figure 3. 17 Profile Page

3.8.5 Diagram 5

Customers could conveniently access the menu after logging in

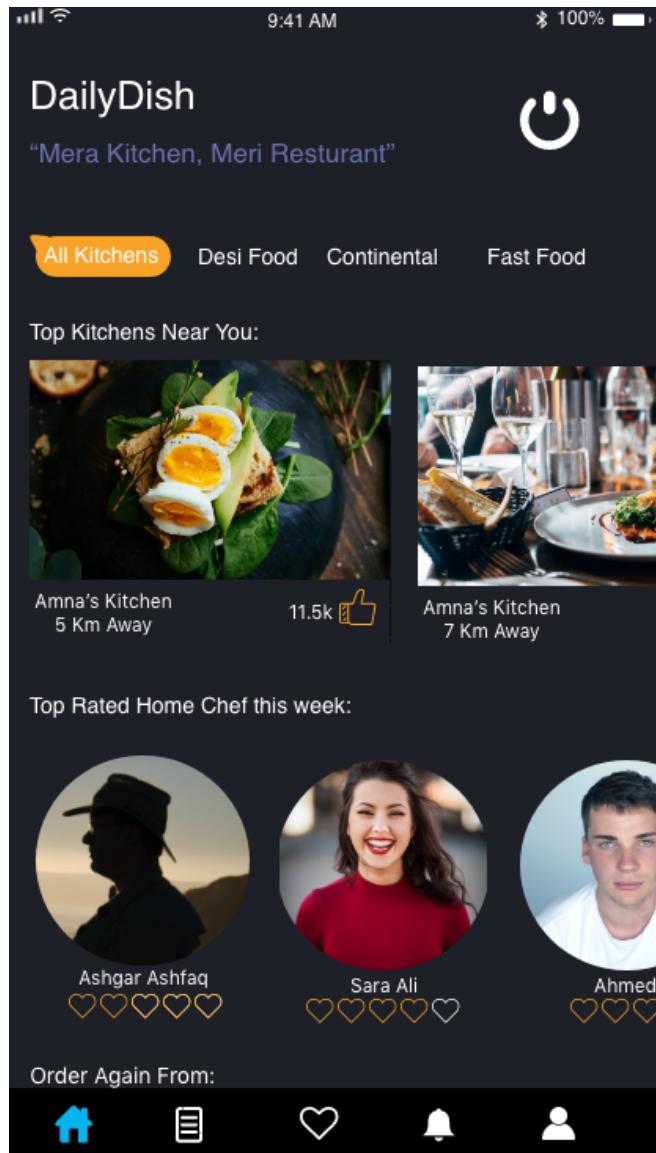


Figure 3. 18 Menu Page from view of the Customers

3.8.6 Diagram 6

Chef Sign Up

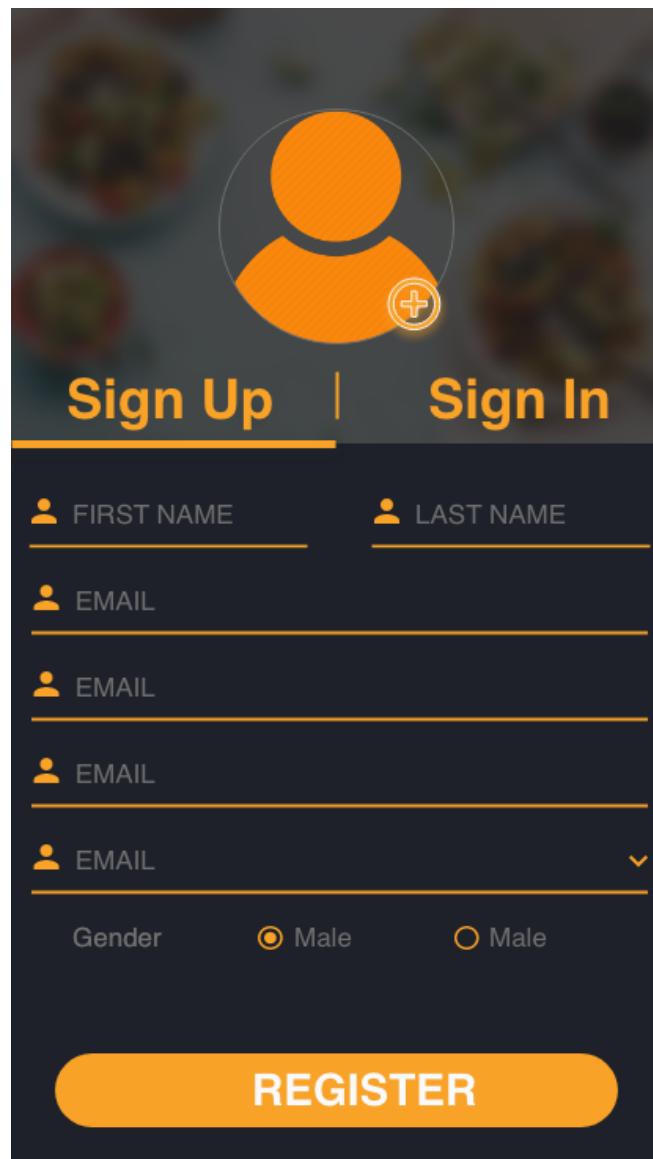


Figure 3. 19 Chef Signup

3.8.7 Diagram 7

Chef Login

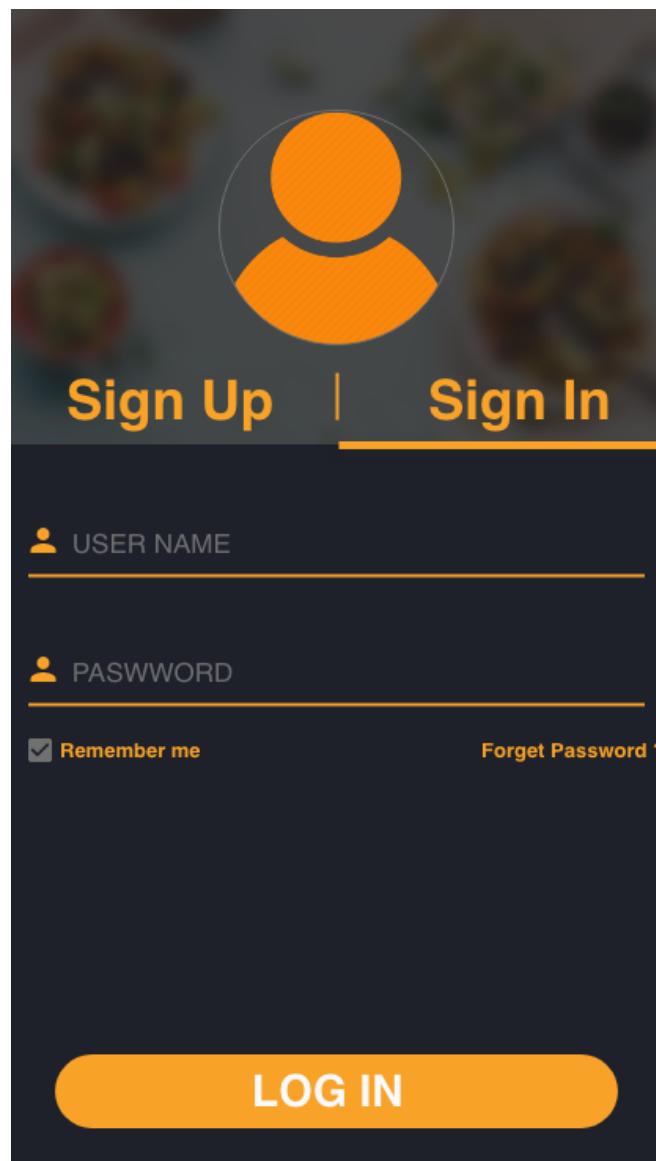


Figure 3. 20 Chef Log in

3.8.8 Diagram 8

There would be a separate view for the profile of the cooks, which could be modified by the cooks as they wish.

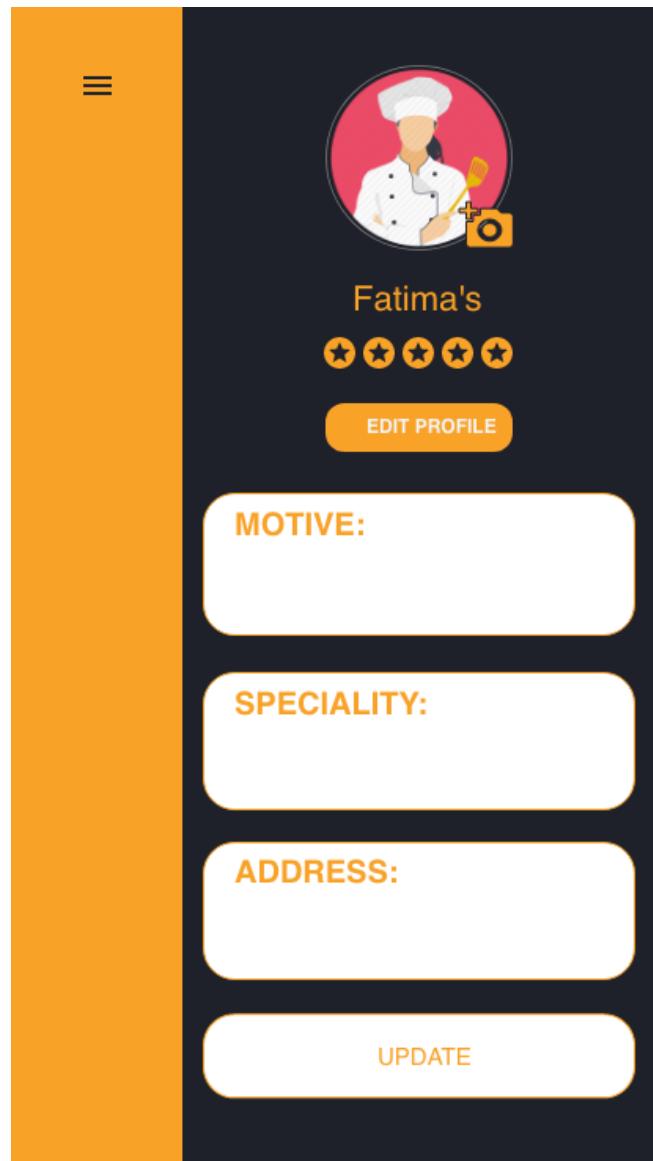


Figure 3. 21 Profile page for the Cook

3.8.9 Diagram 9

The cook could edit and update the menu

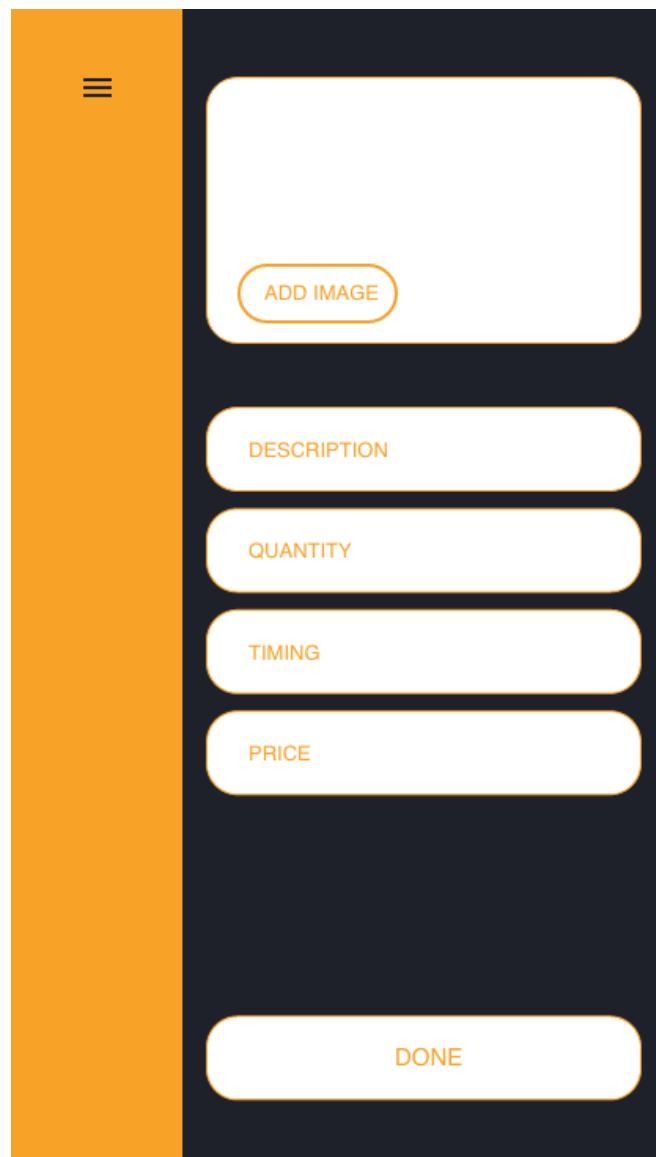


Figure 3. 22 Menu Update Form

3.8.10 Diagram 10

The profiles of chef options

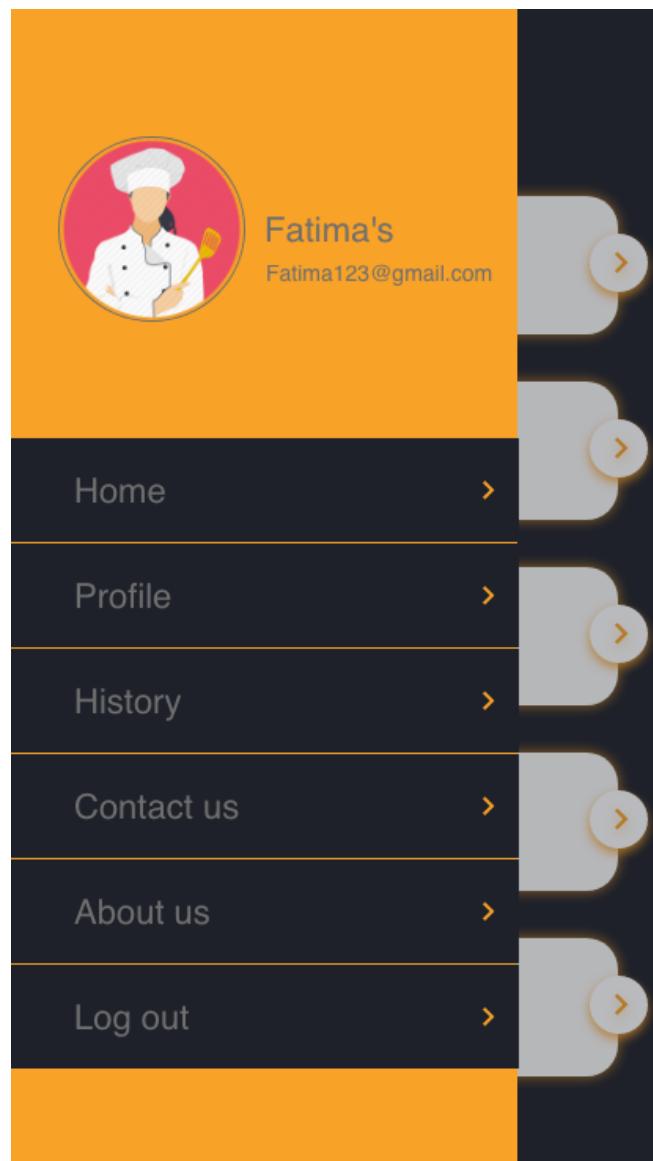


Figure 3. 23 Chef Profile Options

3.8.11 Diagram 11

Chef Menu View

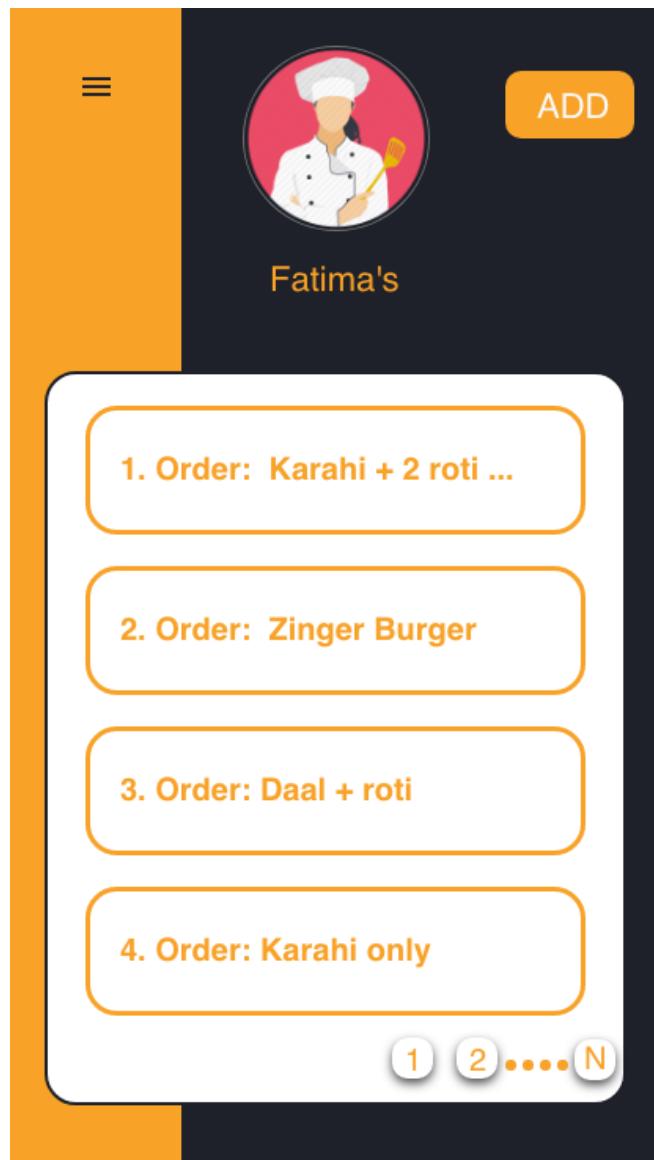


Figure 3. 24 Chef's Menu

3.8.12 Diagram 12

Chef or Users Contact Options

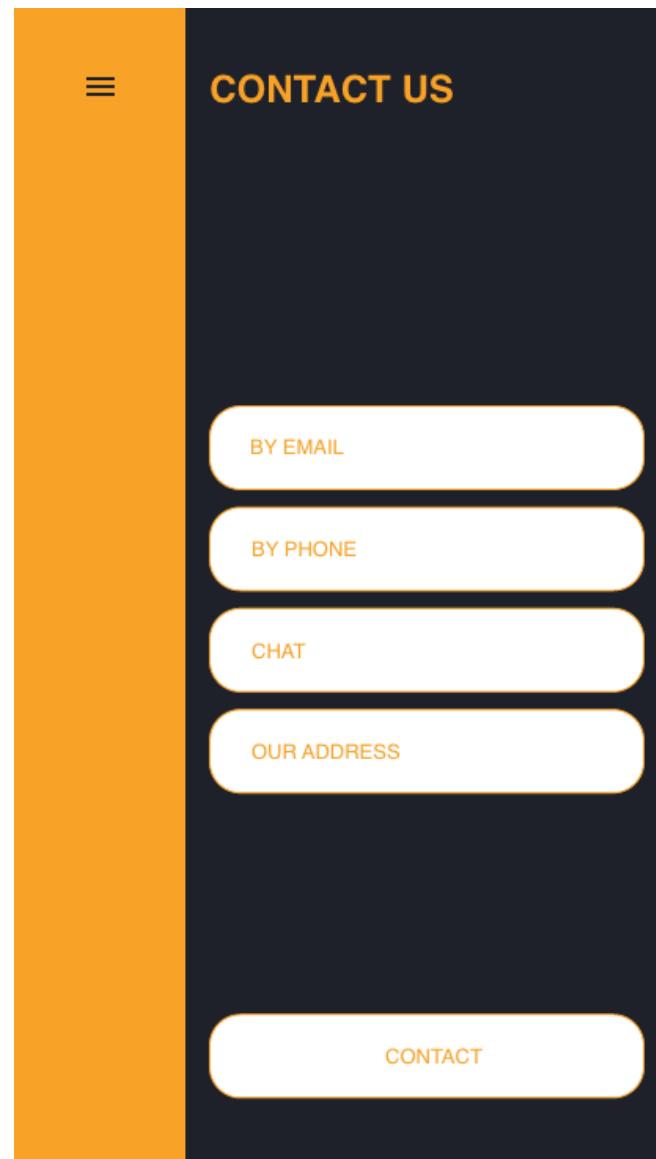


Figure 3. 25 Contact Us

4 Conclusion

4.1 Problems Faced and Lesson Learned

4.1.1 Problems Faced

- Frequently changing versions and documentation of tools used.
- Undefined is not an object error in React Native
- Library Error (Packages not installed properly) in React Native

4.1.2 Lessons Learned

- Use of tools and technologies that are mature and have a large online community.
- Use of JSON based database only when handling data in bulk.
- Rather than testing on a local server, better to test on an online server from early on.

4.2 Project Summary

DailyDish as a cross-Platform application which will provide clean, healthy and hygienic food to people living in hostels or independently in urban cities. The application focuses upon the healthier side of eating habit, as it is believed that due to better eating routine, productivity of an individual increases. User can pick meal from their desired kitchen or chef. Home based chef who is preparing food from ease of their home will accept the order on basis of availability and Delivery man will deliver food to the customer's doorstep in no time. This will also generate employment for two sectors e.g. individuals who have cooking skills and people who have bike or vehicle they can use to earn extra money by performing delivery jobs. Customer can subscribe to weekly or monthly meal plans. On whole project focuses upon providing healthy meals while generating employment for physically challenged individuals, women and students.

5 References

1. "FoodPanda." <https://www.foodpanda.pk/>. Accessed 7 Feb. 2020.
2. "Cheetay." <https://www.cheetay.pk/>. Accessed 7 Feb. 2020.
3. "FoodPanda acquires EatOye - Recent - Aurora." <http://aurora.dawn.com/news/1140757>. Accessed 7 Feb. 2020.
4. "Uber Eats". <https://www.ubereats.com/>. Accessed July 24, 2020
5. "Careem Delivery". <https://www.careem.com/en-pk/careem-now/>. Accessed July 24, 2020
6. "McDelivery". <https://www.mcdelivery.com.pk/pk/>. Accessed July 29, 2020
7. "Agile" <https://www.javatpoint.com/software-engineering-agile-model> Accessed July 26, 2020