**INDEX**

|  |  |
| --- | --- |
| **Index No.** | **Contents of Project Folder** |
| 1. | README File |
| 2. | SRA |
| 3. | Test Plan |
| 4. | User Manual |
| 5. | Source Code |

**README**

**Team Members:**

1. Cameron Chilcoat
2. [clc0553@mavs.uta.edu](mailto:clc0553@mavs.uta.edu)
3. (469) 525-2007
4. Ja’Lun Morris
5. [jalun.morris@mavs.uta.edu](mailto:jalun.morris@mavs.uta.edu)
6. (225) 916-5512
7. Sumaita Sabaha
8. [sumaita.sabaha@mavs.uta.edu](mailto:sumaita.sabaha@mavs.uta.edu)
9. (682) 306-2358
10. Gia Dao
11. [gia.daoduyduc@mavs.uta.edu](mailto:gia.daoduyduc@mavs.uta.edu)
12. (682) 360-8728

**Note for graders:**

Username: [giacoi2907@gmail.com](mailto:giacoi2907@gmail.com)

Password: 12345678

In the SRA, section 4.1 “Registration and Login”, User will not be able to choose Forgot Password option.

In the SRA, section 4.2 “Search”, Search function is not implemented therefore User cannot search for a specific restaurant.

In the SRA, section 4.4 “Orders”, User will not be able to choose “Remove from cart” button.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
| Project: | University Food System (UFS) | | |
| Team No.: | Team 3 | | |
| Class: | CSE 3310; Fall 2021 | | |
| Module: | System Requirements Analysis (SRA) | | |
| Deliverable: | SRA Document | | |
| Version: | [1.0] | Date: | [10/28/2021] |

Contributors:

Cameron Chilcoat

Ja’Lun Morris

Sumaita Sabaha

Gia Dao

Revision History

| Version number | Date | Originator | Reason for change | High level description of changes |
| --- | --- | --- | --- | --- |
| 1.0 | 10/28/2021 | Team# | Initial draft |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

TABLE OF CONTENTS

[1. Introduction and Project Overview 3](#_Toc86273630)

[2. Objectives 4](#_Toc86273631)

[2.1 BUSINESS Objectives 4](#_Toc86273632)

[2.2 SYSTEM Objectives 5](#_Toc86273633)

[3. Project Context Diagram 7](#_Toc86273634)

[4. Systems Requirements 8](#_Toc86273635)

[4.1 “Registration And Login” Requirements 8](#_Toc86273636)

[4.2 “Search” Requirements 11](#_Toc86273637)

[4.3 “Payments” Requirements 12](#_Toc86273638)

[4.4 “Orders” Requirements 13](#_Toc86273639)

[4.5 “Restaurants” Requirements 14](#_Toc86273640)

[4.6 “Advertisement” Requirements 17](#_Toc86273641)

[4.7 “Delivery” Requirements 18](#_Toc86273642)

[4.8 “Communication” Requirements 19](#_Toc86273643)

[4.9 “Rating” Requirements 20](#_Toc86273644)

[5. Software Processes and Infrastructure 22](#_Toc86273645)

[5.1 Hardware and Infrastructure 22](#_Toc86273646)

[5.2 UML Diagrams 22](#_Toc86273647)

[5.3 Screen Shots 32](#_Toc86273648)

[5.4 Test Plan 32](#_Toc86273649)

[6. Assumptions and Constraints 33](#_Toc86273650)

[6.1 ASSUMPTIONS 33](#_Toc86273651)

[6.2 CONSTRAINTS 33](#_Toc86273652)

[6.3 Out of Scope material 33](#_Toc86273653)

[7. Delivery and Schedule 34](#_Toc86273654)

[8. Stakeholder Approval Form 36](#_Toc86273655)

[Appendix: 37](#_Toc86273656)

# Introduction and Project Overview

Develop an android application to help university students order food and provide information about on-campus and near-campus restaurants and food providers (e.g., restaurant review and comments). The system should have an appealing home screen and minimally include the following functionality (you may modify or add additional functionality after receiving approval from your client):

* Registration: Students must register before they can use the system. University ID can be used as member Id.
* Login: Members must login to use the system after initial registration.
* Payments: System should be able to handle payments (Meal plan and Credit cards only; You do not need to validate credit cards for this project, just make them 16 digits long, have an expiration date, and a 3-digit security code)
* Restaurants: System should allow vendors to add their restaurants and their menu (e.g., subway, Panda express, connection café and other on campus restaurants).
* Orders: Place food order (e.g., Sandwich and coke with chips etc.). System should also allow for modifying your order after the initial placement within 5 minutes of your initial order time (i.e., cancel an order to modify existing order by adding new items).
* Communication: System should allow for sending text messages/e-mails to a single, subgroup or all members (e.g., Your order has been placed, your order is on its way etc.)
* Delivery: System should allow students to track their order and get their food delivered or select an option for pick up.
* Favorites list: Allow creating a favorite list of restaurants (you must have ordered at least once from a restaurant before you can add it to your favorite’s list)
* Search: You should be able to search for different food items or restaurants
* Review and Rating: Use Stars (one to five stars) to rate the service received, and allow to enter for a short comment to help future users
* Advertisements: System should provide advertising space (e.g. Local subway store offers specials or coupons to members). This could be a source of revenue from the app.
* {**for Honors Credit only**}
  + Create a map displaying your location and the resultants.
  + Include an accurate time estimation of the project. Prepare a 6-page report (single-space, font 11 or 12) using “Softstar Systems – COCOMO estimation” at: <http://www.softstarsystems.com/demo.htm>

# 2. Objectives

## 2.1 BUSINESS Objectives

The following is a list of business objectives:

**Objective 1**: Login/Registration: All members must provide the following information prior to using the system:

If the user is not registered, then he/she needs to provide the following information

* Name {First, Last name}
* Email address
* Phone number
* Select an ID {at least 8 character long, alphanumeric, wild characters not allowed}
* Select a password {Must include, letters, at least one number, at least Capital letter, wild characters allowed)
* Select a security question/answer (to be Used for password forgot/reset)

If the user already registered, he/she can type in the user ID and password and press the button to log in.

If the user forgets the password, ask user for the Answer to the “Security Question” entered during the “Registration” phase. Also ask for e-mail address. If both e-mail and security answer is correct, reset the password and e-mail the temporary password via e-mail

**Objective 2**: Search: System should be able to allow user to use the search bar, which will display the suggested restaurants according to the result of the search

To search for a specific location within the university area, user must provide either:

* Name of the restaurant
* Location of the restaurant

After search and suggestions are shown in the drop-down menu, user can press on the restaurant to navigate to the restaurant’s main page. If the restaurant is not available, the system should notify the user with error message, and ask if the user want to select a different location.

**Objective 3**: Payments: System should be able to handle payments (Meal plan and Credit cards only)

Display two options of payment on the screen for the user to choose from: Credit cards or Meal Plan

If the user selected ‘credit card’:

* If the user is not registered this method, they must provide the following information:
* Name {First, Last name}
* Address
* Email
* Phone number
* 16-digit card number an ID {at least 8 character long, alphanumeric, wild characters not allowed}
* CVV number
* Expiration date

**Objective 4**: Order: System would allow user to make an order (or select items) from a specific restaurant:

Display menu that contains the pictures of the items and their prices according to the chosen restaurant. The system allows the user to add the items to their cart by clicking on the icon or remove items from their cart. The user can also view the sub total of their order every time they access their cart.

The user can press the buttons:

* Add Button to add the items
* Remove Button to remove the items
* Cancel Button to discard the order

**Objective 5**: Restaurant: Each restaurant will have the following information

* Name
* Location
* Cuisine
* Hours
* A Menu
* Current backup time

In the app there will be a screen to view the restaurant’s menu when placing an order. An administrator side to the system will update restaurant and menu qualities. The server side will maintain a list of restaurants and its qualities.

**Objective 6**: Advertisement: Each restaurant will be able to create an advertisement that can be clicked to view the order placing menu screen. An advertisement includes the following.

* Image file
* Text
* Font
* Background color
* Restaurant

**Objective 7:** Delivery: The system should let the customer choose between pick and delivery:

The system should display two buttons for user to choose: Delivery and Pick Up

Customer chooses pickup, customer will get alert when order is ready

If the customer picks delivery:

* The system will notify the shipper’s information, including vehicle types, color, …
* Calculated the estimated time before reaching the user’s location
* Display successful delivery message if the user received their items
* At the same time, the user will be able to track their order using map displayed on the screen

**Objective 8:** Communication: System should allow the user to send messages/e-mails to the administrators

If the user has any complaint, they can directly contact the phone or email address that’s attached below the selected restaurant. The system will then notify the user it has received the feedback and recommend the user to wait for response.

**Objective 9:** Ratings: Each order completed will be able to be reviewed by the placing customer. A rating will include the following:

* Number of stars
* Reason
* Month of order
* Year of order
* Reviewed

Ratings are maintained and stored via a ratings list for each restaurant. Reviewed ratings are displayed in app.

## 2.2 SYSTEM Objectives

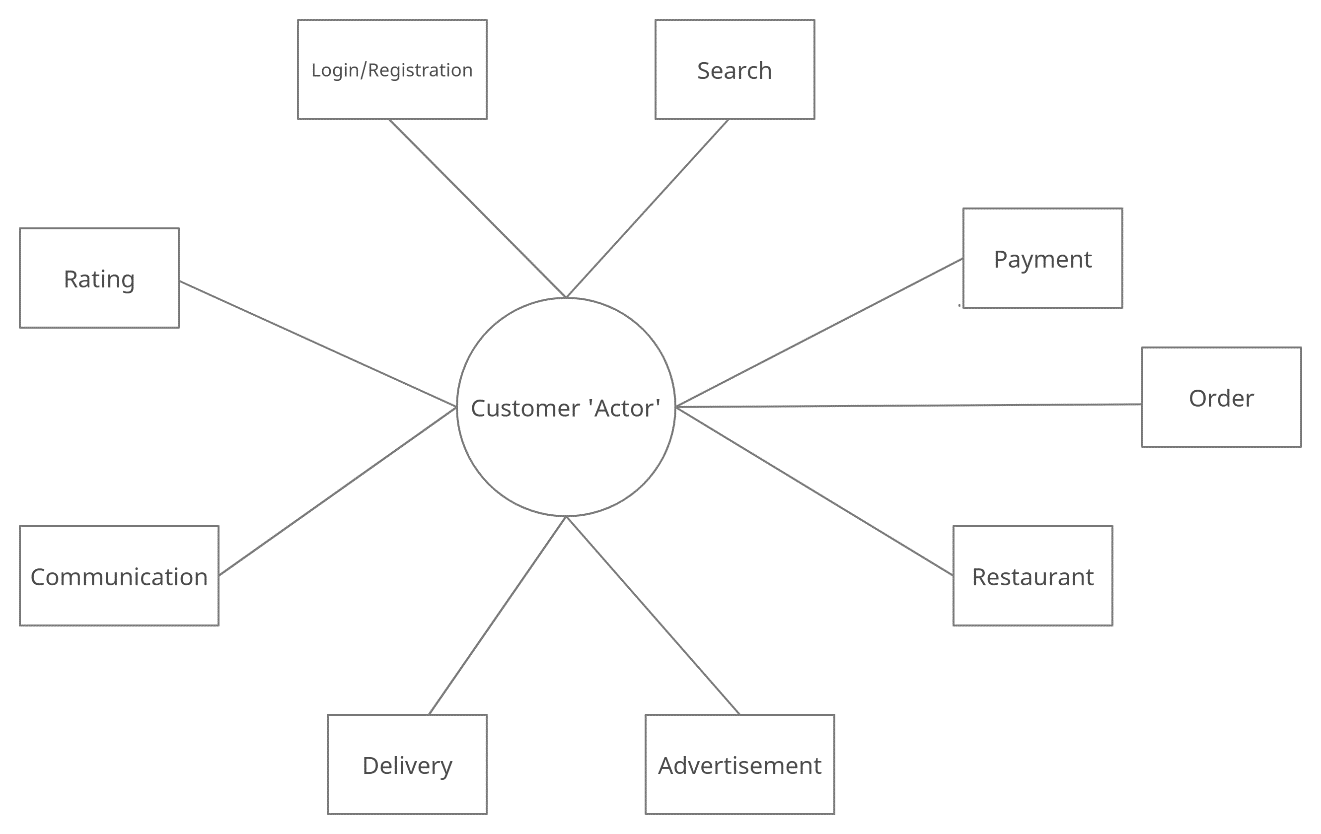
The following is a list of system objectives:

**Objective 1**: System will be an Android application version 2020.3.1 Bumblebee

**Objective 2**: Search functionality will utilize Google’s search

**Objective 3:** Firebase will be used to save limited amount of data (stored the user’s information, such as name, phone number, email…)

# 3. Project Context Diagram

****

# 4. Systems Requirements

## 4.1 “Registration And Login” Requirements

|  |  |
| --- | --- |
| **Requirement Title:** | Login and Registration |
| **Sequence No:** | 001 |
| **Short description:** | Register new user |
| **Description:** | New users must register before accessing the application. The following information will be collected:   * Name {First, Last name} * Email address * Phone number * Select an ID {at least 8 character long, alphanumeric, wild characters not allowed} * Select a password {Must include, letters, at least one number, at least Capital letter, wild characters allowed) * Select a security question/answer (to be Used for password forgot/reset)   User can press:   * Submit * Cancel (i.e., discard changes) * Exit screen (or close) |
| **Pre-Conditions**: | * Application must be loaded already * Duplicate registration is not allowed |
| **Post Conditions:** | * All “Saved” changes will be kept permanently |
| **Other attributes:** | * None |

|  |  |
| --- | --- |
| **Requirement Title:** | Login and Registration |
| **Sequence No:** | 002 |
| **Short description:** | Login Existing User |
| **Description:** | Enter User Id and Password (already established during Registration) and Press Submit  User can press:   * Submit * Forgot ID or Password * Cancel (i.e., discard changes) * Exit screen (or close) |
| **Pre-Conditions**: | * User must have already registered |
| **Post Conditions:** | * Keep a log of the date and time of login |
| **Other attributes:** | * Allow a maximum of three tries, then recommend using “Forgot Password” |

|  |  |
| --- | --- |
| **Requirement Title:** | Login and Registration |
| **Sequence No:** | 003 |
| **Short description:** | Forgot Password |
| **Description:** | Ask user for the Answer to the “Security Question” entered during the “Registration” phase, i.e. see 001 above. Also ask for e-mail address. If both e-mail and security answer is correct, reset the password and e-mail the temporary password via e-mail  User can press:   * Submit * Cancel (i.e., discard changes) * Exit screen (or close) |
| **Pre-Conditions**: | * User must have already registered * Validate e-mail and security question |
| **Post Conditions:** | * Recommend use to change their temporary password after accessing the system |
| **Other attributes:** | * Lock system after 3 tries * Consider adding more than one security question |

## 4.2 “Search” Requirements

|  |  |
| --- | --- |
| **Requirement Title:** | Search |
| **Sequence No:** | 001 |
| **Short description:** | Search any data available to users |
| **Description:** | Application will have search functionality. Users will be able to search for any restaurants or food items. Search functionality will be able to utilize Google search.   * User will type the name of the restaurant or food item in the search box * Search suggestions will be based on recent queries * If input matches suggestions, search results will be displayed * If input doesn’t match the data, then results error message is shown * User can search unlimited times   User can press:   * Search button * Cancel (i.e., discard changes) |
| **Pre-Conditions**: | Application must be loaded already  Search button  A search dialog that appears at the top of the screen when search button is clicked |
| **Post Conditions:** | Providing valid input in search leads to search results  Providing invalid input in search displays error message |
| **Other attributes:** | None |

## 4.3 “Payments” Requirements

|  |  |
| --- | --- |
| **Requirement Title:** | Payment |
| **Sequence No:** | 001 |
| **Short description:** | User decides on how to pay between meal plan or card. |
| **Description:** | Display meal plan or credit card for payment method. If user press payment plan it will prompt the customer to enter student ID. If the customer press ‘credit card’ he or she must enter 16-digit card number, expiration date, and 3-digit security code.  In addition, if the customer changes their mind, after seeing price they can add to cart or take off cart.  User can press:   * Add more to cart Button * Take off Button * Type 3-digit security code Button * Enter student ID * Enter 16-digit card number * Enter expiration date * Place order button * Track order |
| **Pre-Conditions**: | User have already shopped for their items and wants to make a payment to place order |
| **Post Conditions:** | Ask the user do you want to track your  If no:  Show customer receipt and order number.  If yes:  Direct customer to Delivery page |
| **Other attributes:** | After the Customer is finished Paying, notification “Thank you for shopping is” is popped up |

## 4.4 “Orders” Requirements

|  |  |
| --- | --- |
| **Requirement Title:** | Orders |
| **Sequence No:** | 001 |
| **Short description:** | User selects their choice.  Add/Remove the items to/from cart |
| **Description:** | Display a list of categories at the selected restaurant, such as Beverages, Breakfast, Combo Meals, Desserts, …  User can see the images of the food that is currently served at the restaurant, and add them to their cart  User can view their cart and remove an item if the item is not their choice, or cancel their order  In addition, if the user views their cart, display the total cost of the selected items  User can press:   * Menu Button * Add to Cart Button * Remove from Cart Button * Checkout Button * Cancel Button |
| **Pre-Conditions**: | User have already logged in  User must select a specific restaurant |
| **Post Conditions:** | Ask the user do you want to continue shopping  If no:  navigate user to proceed to checkout  If yes:  remain in the same food displaying screen |
| **Other attributes:** | After selection, notification “Item added to your cart” is popped up |

## 

## 4.5 “Restaurants” Requirements

|  |  |
| --- | --- |
| **Requirement Title:** | Restaurant |
| **Sequence No:** | 001 |
| **Short description:** | Restaurant Account  Each restaurant will have a restaurant account |
| **Description:** | A restaurant class contains the following information:   * Name * Location * Cuisine * Hours * A Menu * Current backup time   A restaurant will be able to conduct or respond to the following actions:   * View menu: Show a customer a menu in app * Place an order: A customer can submit an order to a restaurant at which point the restaurant will need to accept the order and mark the order completed when it is done * Check hours: The app will be able to send an hour check to the restaurant * Temporarily close: The restaurant can temporarily close * Change hours: A restaurant can change hours freely * Edit restaurant: A restaurant can submit a change to the restaurant’s name, address, location, or cuisine to the administrators * Edit menu: the restaurant account has access to edit all aspects of its menu |
| **Pre-Conditions**: | A restaurant is created by admins via the restaurants list |
| **Post Conditions:** | None |
| **Other attributes:** | None |

|  |  |
| --- | --- |
| **Requirement Title:** | Restaurant |
| **Sequence No:** | 002 |
| **Short description:** | Restaurant List-  All restaurant accounts are managed by the restaurant list |
| **Description:** | The restaurant list contains the following information:   * List of restaurants   A restaurant will be able to conduct or respond to the following actions:   * Add restaurant: The restaurant list can add a restaurant providing all restaurant information. Two restaurants should not have the same location and name. * Delete restaurant: The restaurant list can remove restaurants * Edit restaurant- the restaurant list can change any restaurant information |
| **Pre-Conditions**: | * The restaurant list can be empty but always exists in the system |
| **Post Conditions:** | None |
| **Other attributes:** | None |

|  |  |
| --- | --- |
| **Requirement Title:** | Restaurant |
| **Sequence No:** | 003 |
| **Short description:** | Menu  Each restaurant account contains a menu |
| **Description:** | The restaurant list contains the following information:   * List of Items   A Menu will be able to conduct or respond to the following actions:   * Add item: The restaurant can call to add an item * Delete item: The restaurant can call to delete an item * Edit item: The menu can be called to edit any menu items info * View: view menu |
| **Pre-Conditions**: | None |
| **Post Conditions:** | None |
| **Other attributes:** | None |

## 4.6 “Advertisement” Requirements

|  |  |
| --- | --- |
| **Requirement Title:** | Advertisement |
| **Sequence No:** | 001 |
| **Short description:** | Advertisement  Advertisements linked to restaurants present in the app which will be displayed on user’s home page and link to the restaurant page |
| **Description:** | An advertisement contains the following information:   * Image file * Text * Font * Background color * Restaurant   An Advertisement responds to a display request |
| **Pre-Conditions**: | None |
| **Post Conditions:** | None |
| **Other attributes:** | None |

## 4.7 “Delivery” Requirements

|  |  |
| --- | --- |
| **Requirement Title:** | Delivery |
| **Sequence No:** | 001 |
| **Short description:** | Show delivery process |
| **Description:** | Display options to see if the user wants their items to be delivered  If yes, notify the restaurant to find a driver, display message: “Finding your driver”  If found, send the user the driver’s information  Calculated the estimated time before the driver reaches the destination  If the user decided to pick up, calculated preparing time, then notify user to pick up if finished  User can press:   * Delivery Button * Pick Up Button |
| **Pre-Conditions**: | The user must log in to the application  The user must already make the order and pay for the items |
| **Post Conditions:** | None |
| **Other attributes:** | Send the user message: “Thank you for using delivered service” if the user request for delivery |

## 4.8 “Communication” Requirements

|  |  |
| --- | --- |
| **Requirement Title:** | Communication |
| **Sequence No:** | 001 |
| **Short description:** | Get order updates |
| **Description:** | System should allow for sending text message/e-mails to a single, subgroup or all users. Users will get updates on their orders once they confirm their order. Updates will show:   * If order has successfully placed * Delivery Status * Estimated time of arrival * If order had been picked up * If order has been delivered   User can press:   * Yes * Cancel |
| **Pre-Conditions**: | Application must be loaded already  Order must be confirmed |
| **Post Conditions:** | All updates’ records will be saved |
| **Other attributes:** | None |

## 4.9 “Rating” Requirements

|  |  |
| --- | --- |
| **Requirement Title:** | Ratings |
| **Sequence No:** | 001 |
| **Short description:** | Ratings  Customers are given the option to rate previously placed orders |
| **Description:** | The customer will give the following info in a rating:   * Rating out of 5 * If a rating over 4 is given what was positive is asked answer is optional * If rating was under 4 what was negative is asked answer is non-optional   Allow user to type into the feedback blank space  User can press:   * Submit Feedback Button |
| **Pre-Conditions**: | Order has been previously placed and completed but has not yet been reviewed |
| **Post Conditions:** | Order updated to rated, ratings are sent to review |
| **Other attributes:** | None |

|  |  |
| --- | --- |
| **Requirement Title:** | Ratings |
| **Sequence No:** | 002 |
| **Short description:** | Rating list  Each restaurant contains two ratings lists a reviewed and non-reviewed list |
| **Description:** | The rating list contains the following attributes:   * average rating * ratings   The ratings respond to the following:   * add\_rating() function * delete\_rating() function |
| **Pre-Conditions**: | None |
| **Post Conditions:** | None |
| **Other attributes:** | None |

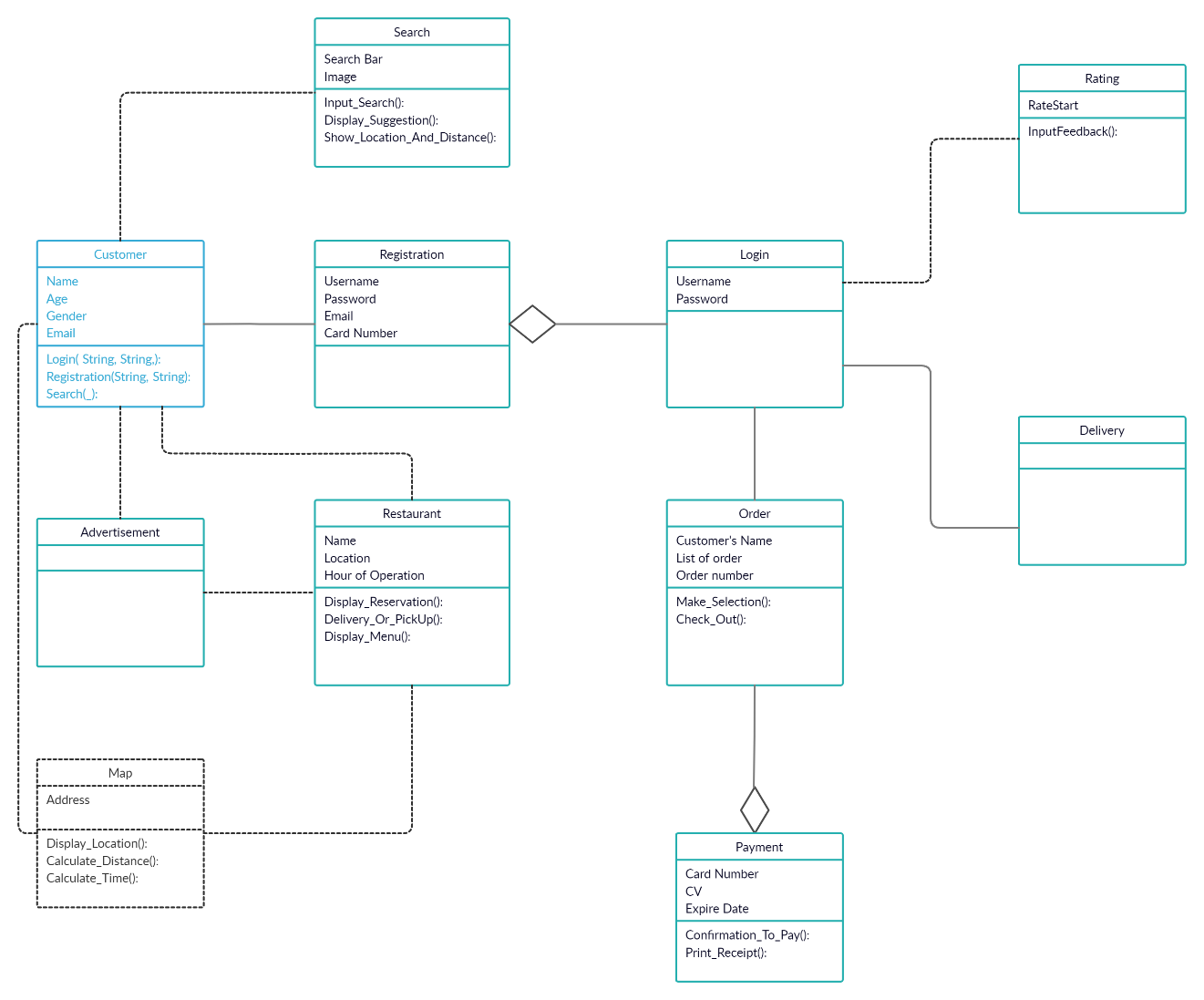
# 5. Software Processes and Infrastructure

## 5.1 Hardware and Infrastructure

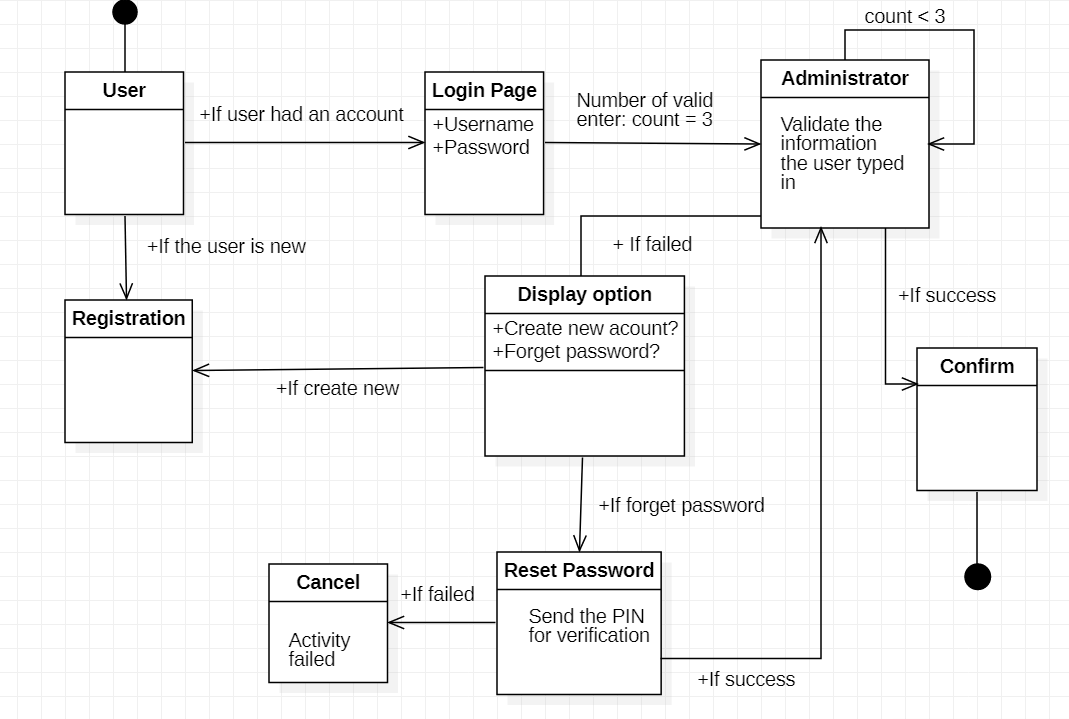
The University Food System Project will be implemented using Android version KitKat

## 5.2 UML Diagrams

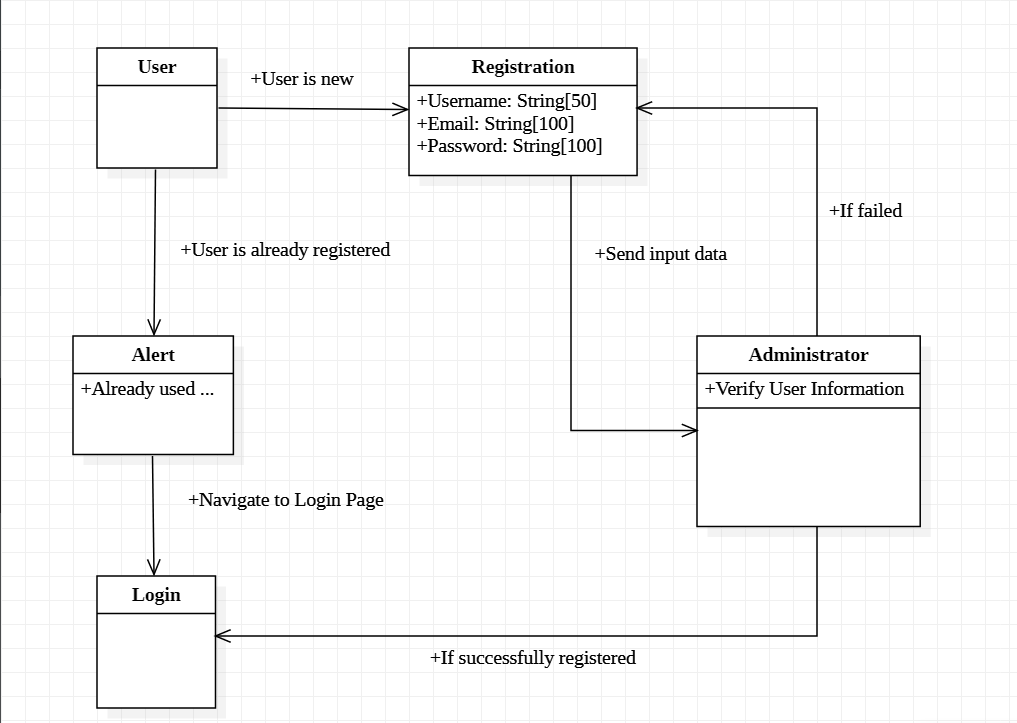
**UML – Class Diagram – University Food System**

****

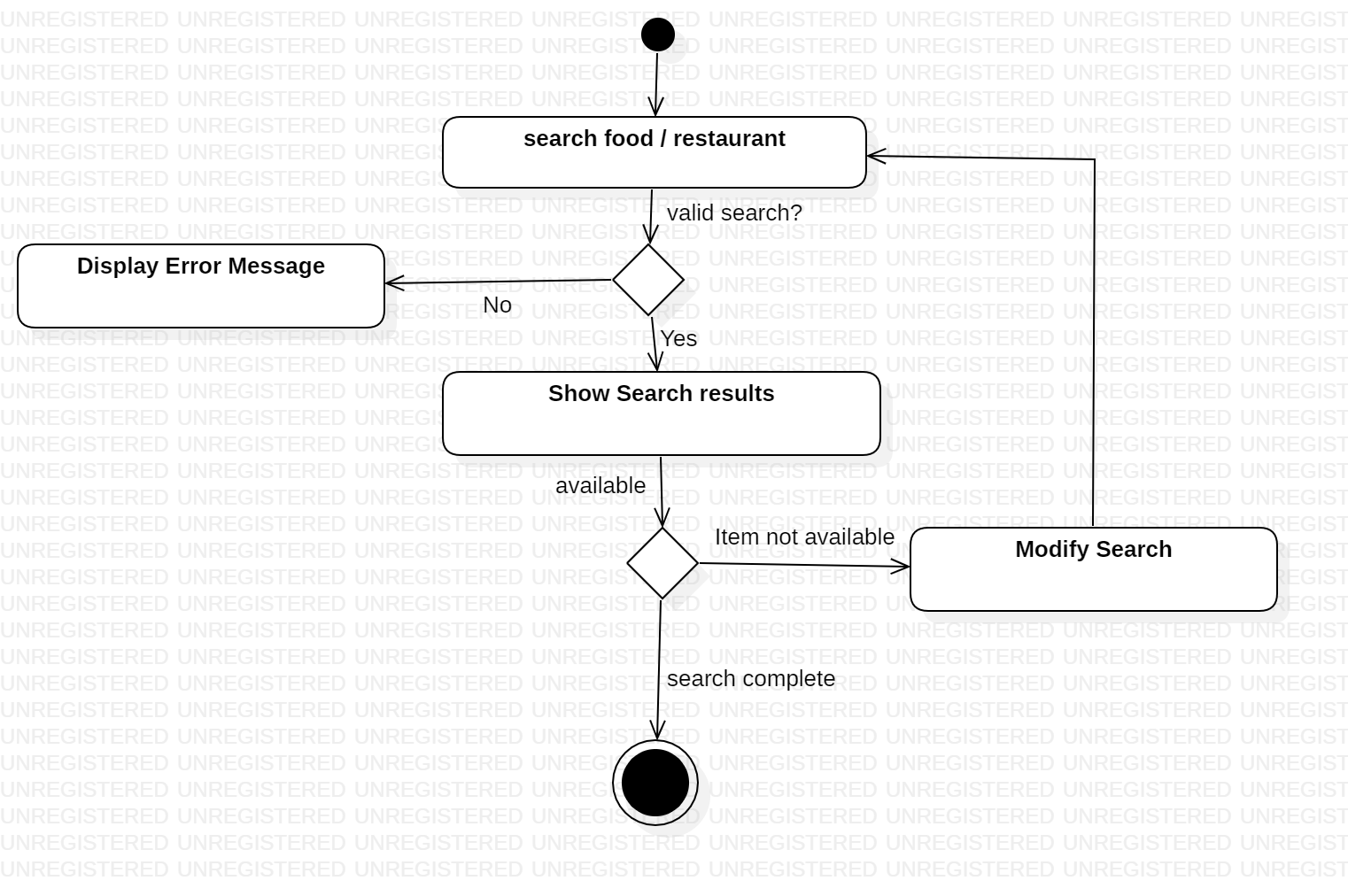
**UML – State Transition Diagram – Login and Registration (Part 1)**

****

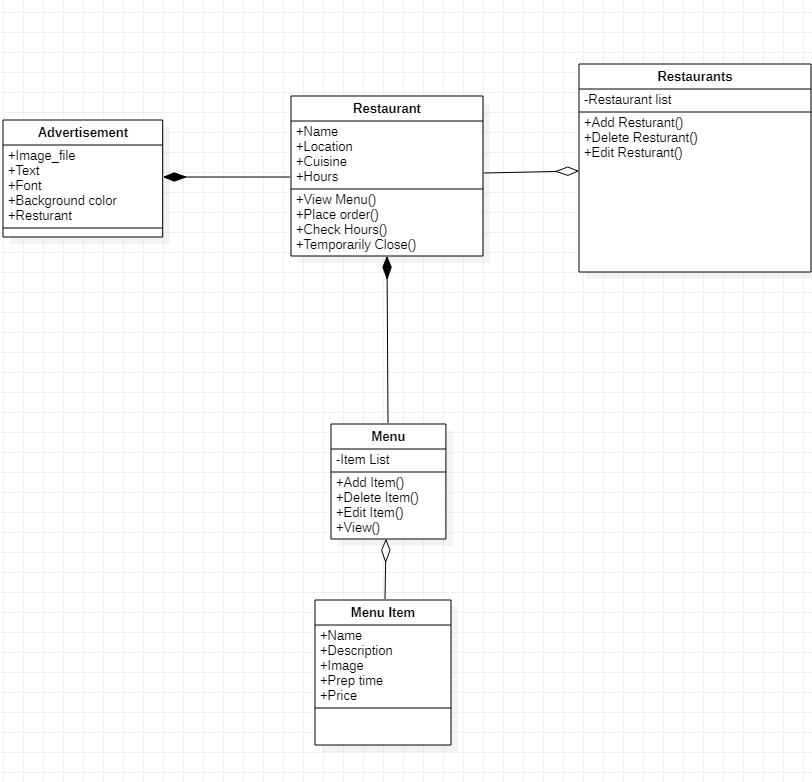
**UML – State Transition Diagram – Login and Registration (Part 2)**

****

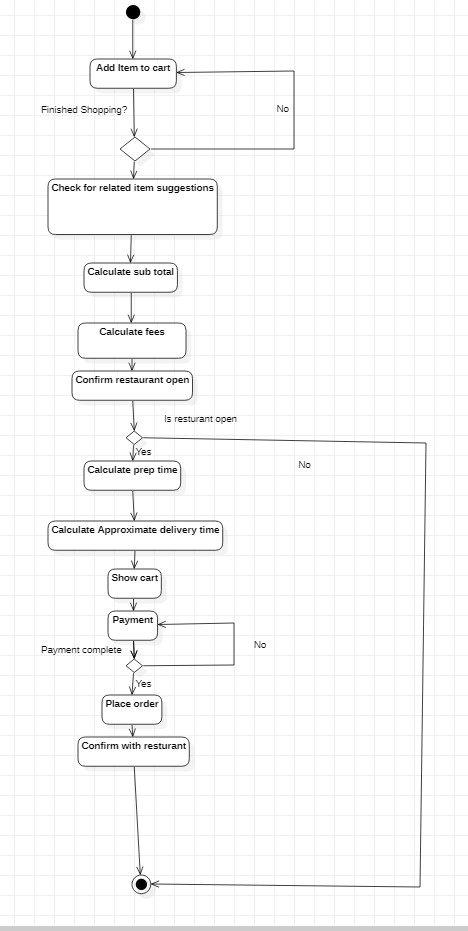
**UML – Activity Diagram – Search**

****

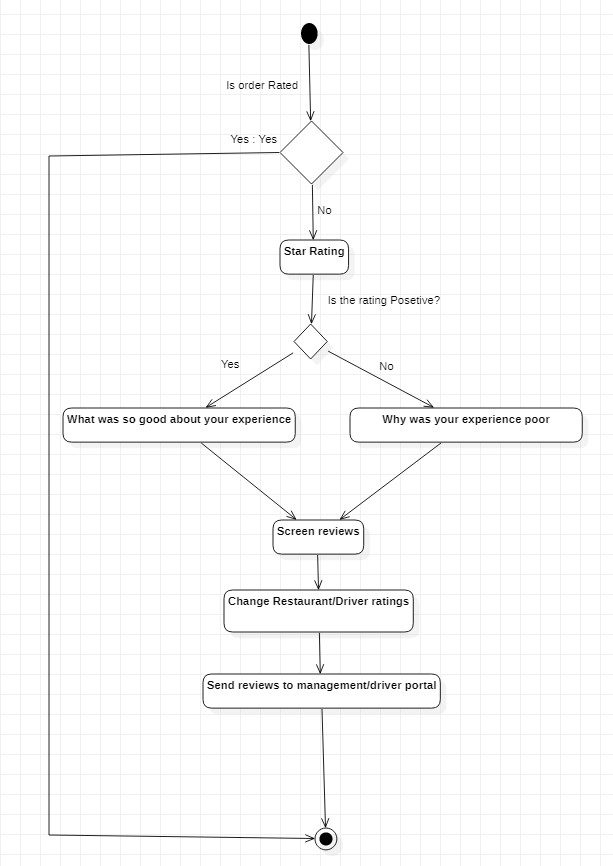
**UML –Class Diagram – Restaurant and Advertisement**

****

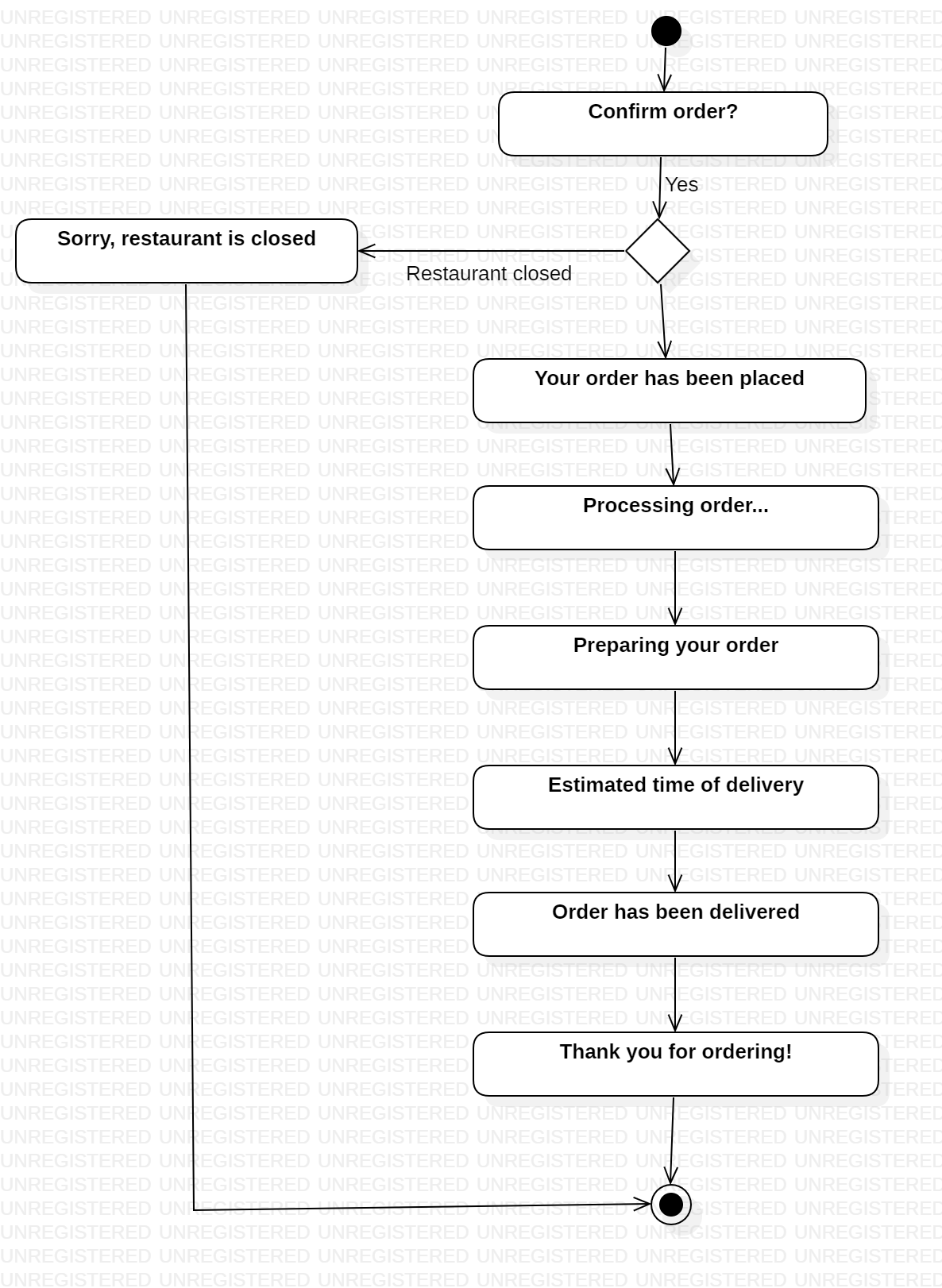
**UML – State Transition Diagram – Order**

****

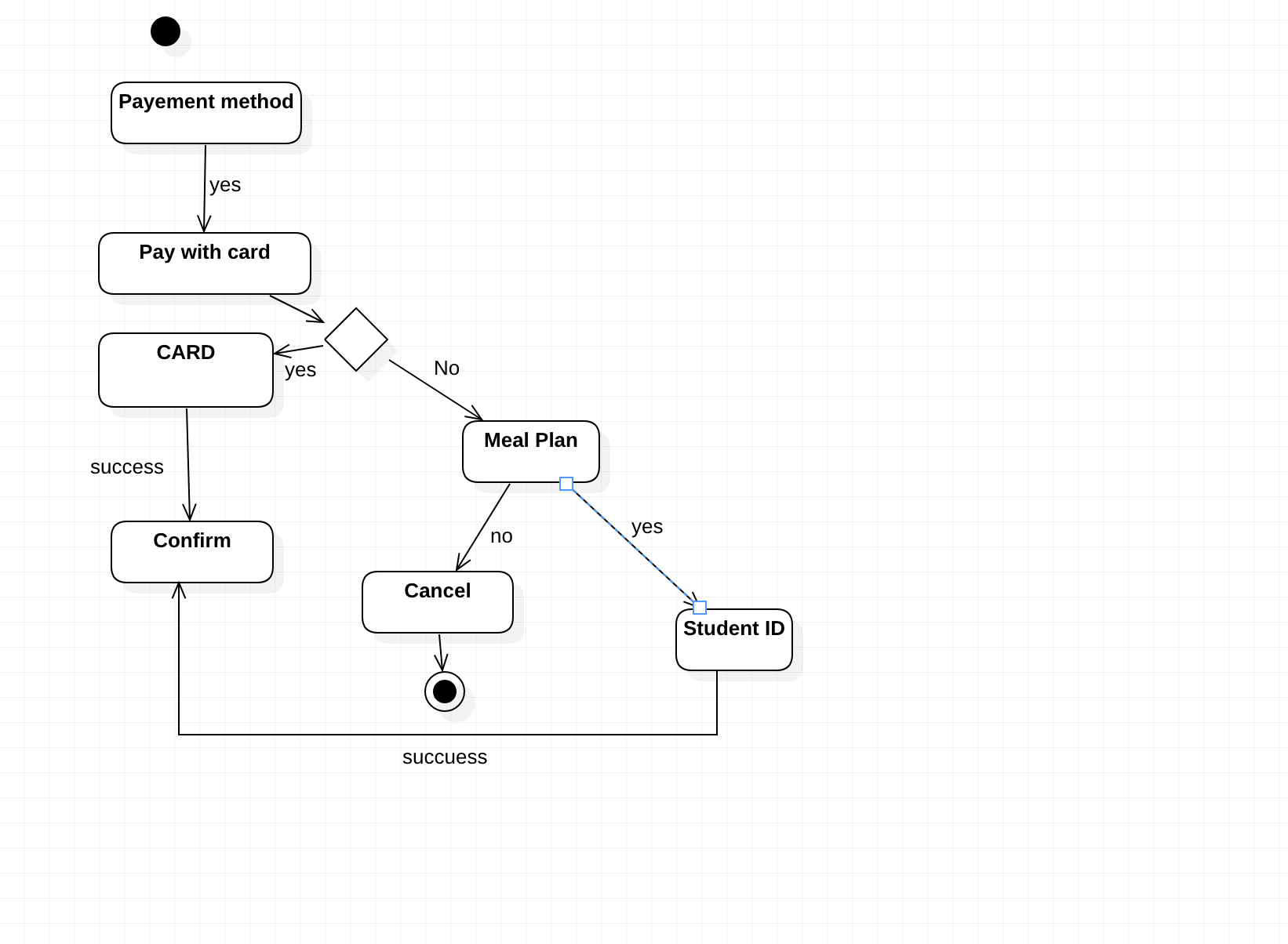
**UML – State Transition Diagram – Ratings**

****

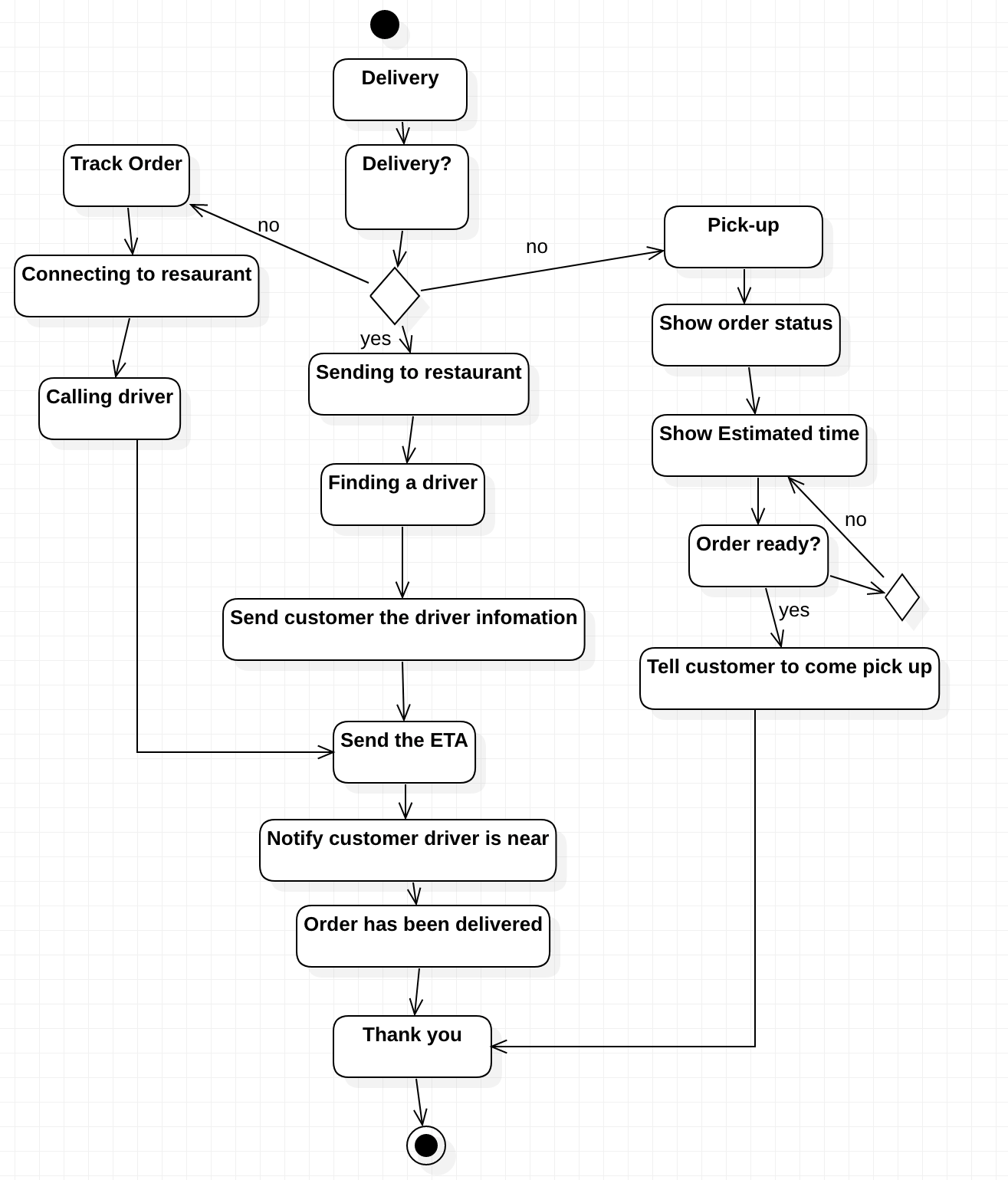
**UML – Activity Diagram – Communication**

****

**UML – State Transition Diagram – Payment**

****

**UML – State Transition Diagram – Delivery**

****

## 5.3 Screen Shots

There are no screen shots available currently

## 

## 5.4 Test Plan

The test plan will be provided at the later stage of the project

# 6. Assumptions and Constraints

## 6.1 ASSUMPTIONS

The following is a list of assumptions:

* Ignore validating credit card numbers initially
* Ignore compliance issues
* Assume the user is already a student on campus to access the system
* The restaurants displayed in the drop-down menu are all located within university area
* Meal exchanges using in payment can be applied to all restaurants within any time

## 6.2 CONSTRAINTS

The following is a list of constraints:

* Team lacks android experience
* Schedule very aggressive
* Team does not have experiencing in implementing database and use search tool before
* Team lack experience in designing interface

## 6.3 Out of Scope material

The following is a list of “out of scope” material:

* Post Project maintenance is not covered
* Creating a map displaying the location of the customer and the restaurants is not part of the first delivery phase

# 7. Delivery and Schedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task/Milestone Description | Anticipated Start Date | Anticipated End Date | Status | Comments |
| Prepare Requirements and UML diagram | 9/14/2021 | 9/30/2021 | Complete | Deliverable UML document. Increment 1 Deliverable |
| SRA document (Includes project objectives, Requirements and UML diagrams) | 10/5/2021 | 10/28/2021 | In Progress | Deliverable will be the SRA document. All stakeholders agree on the content of the SRA by signing in section 8.  Increment 2 Deliverable |
| Home screen design and implementation | 10/18/2021 | 10/20/2021 | In Progress | Gia Dao |
| Login and registration design and implementation | 10/28/2021 | 11/9/2021 | In Progress | Sumaita Sabaha |
| Search implementation | 10/28/2021 | 11/9/2021 | To Be Completed (TBC) | Sumaita Sabaha |
| Payment Implementation | 11/9/2021 | 11/16/2021 | To Be Completed (TBC) | Ja’Lun Morris |
| Order Implementation | 11/9/2021 | 11/16/2021 | To Be Completed (TBC) | Gia Dao |
| Restaurant Implementation | 11/16/2021 | 11/23/2021 | To Be Completed (TBC) | Cameron Chilcoat |
| Advertisement Implementation | 11/16/2021 | 11/23/2021 | To Be Completed (TBC) | Cameron Chilcoat |
| Delivery Implementation | 11/16/2021 | 11/23/2021 | To Be Completed (TBC) | Ja’Lun Morris |
| Communication Implementation | 11/16/2021 | 11/23/2021 | To Be Completed (TBC) | Gia Dao and Sumaita Sabaha |
| Rating Implementation | 11/16/2021 | 11/23/2021 | To Be Completed (TBC) | Cameron Chilcoat |
| Test case design | 10/29/2021 | 11/18/2021 | To Be Completed (TBC) | Increment 3 Deliverable |
| External Documentation (i.e., User Manual) | 12/5/2021 | 12/7/2021 | To Be Completed (TBC) | Team |
| Project presentation | 11/30/2021 | 12/7/2021 | To Be Completed (TBC) | Team |
| Final Milestone: project delivery |  | 12/7/2021 | To Be Completed (TBC) | Increment 4 Deliverable |

# 8. Stakeholder Approval Form

|  |  |  |  |
| --- | --- | --- | --- |
| Stakeholder Name | Stakeholder Role | Stakeholder Comments | Stakeholder Approval Signature and Date |
| Bahram Khalili | Development Mgr |  |  |
| Akshit Singhal | Project Assistant |  |  |
| Cameron Chilcoat | Developer |  | C.C.  10/27/2021 |
| Ja’Lun Morris | Developer |  | J.M  10/27/2021 |
| Sumaita Sabaha | Developer |  | S.S  10/27/2021 |
| Gia Dao | Developer |  | G.D  10/25/2021 |

# Appendix:

None

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Project: | University Food System (UFS) | | | |
| Team No.: | Team 3 | | | |
| Class: | CSE 3310; Fall 2021 | | | |
| Module: | Test Plan | | | |
| Deliverable: | Test Plan Document | | | |
| Version: | | [1.0] | Date: | [11/18/2021] |

Team members are:

Cameron Chilcoat

Sumaita Sabaha

Ja’Lun Morris

Gia Dao

Revision History

| Version number | Date | Originator | Reason for change | High level description of changes |
| --- | --- | --- | --- | --- |
| 1.0 | 11/13/2021 | Team 3 | Initial draft |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

TABLE OF CONTENTS

[1. Introduction and Plan of Approach 3](#_Toc88087100)

[2. Test Cases: “Login and Registration” 4](#_Toc88087101)

[3. Test Cases: “Order” 6](#_Toc88087102)

[4. Test Cases: “Search” 8](#_Toc88087103)

[5. Test Cases: “Payments” 9](#_Toc88087104)

[6. Test Cases: “Delivery” 11](#_Toc88087105)

[7. Test Cases: “Communication” 12](#_Toc88087106)

[8. Test Cases: “Restaurant and Advertisement” 13](#_Toc88087107)

[9. Test Cases: “Ratings” 14](#_Toc88087108)

# 1. Introduction and Plan of Approach

Provide a brief description for the following areas:

Develop an android application to help university students order food and provide information about on-campus and near-campus restaurants and food providers (e.g., restaurant review and comments). The system should have an appealing home screen and minimally include the following functionality (you may modify or add additional functionality after receiving approval from your client):

* Login and Registration: Students must register before they can use the system. University ID can be used as member Id. Members must login to use the system after initial registration.
* Search: You should be able to search for different food items or restaurants
* Payments: System should be able to handle payments (Meal plan and Credit cards only; You do not need to validate credit cards for this project, just make them 16 digits long, have an expiration date, and a 3-digit security code)
* Orders: Place food order (e.g., Sandwich and coke with chips etc.). System should also allow for modifying your order after the initial placement within 5 minutes of your initial order time (i.e., cancel an order to modify existing order by adding new items).
* Restaurants: System should allow vendors to add their restaurants and their menu (e.g., subway, Panda express, connection café and other on campus restaurants).
* Advertisements: System should provide advertising space (e.g. Local subway store offers specials or coupons to members). This could be a source of revenue from the app.
* Delivery: System should allow students to track their order and get their food delivered or select an option for pick up.
* Communication: System should allow for sending text messages/e-mails to a single, subgroup or all members (e.g., Your order has been placed, your order is on its way etc.)
* Review and Rating: Use Stars (one to five stars) to rate the service received, and allow to enter for a short comment to help future users
* {**for Honors Credit only**}
  + Create a map displaying your location and the resultants.
  + Include an accurate time estimation of the project. Prepare a 6-page report (single-space, font 11 or 12) using “Softstar Systems – COCOMO estimation” at: <http://www.softstarsystems.com/demo.htm>

# 2. Test Cases: “Login and Registration”

**Project Name:** University Food System

**Test Case Name:** Login and Registration

**Test Case Id**: CSE3310/Fall 2021/Team3/ Login-and-Registration

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case No.** | **Test Case Description** | **Expected results** | **Outcome**  **Pass, Fail, Other (comments)** |
| TC1 | {In Login Screen}  Tab into the User and password fields and enter a valid user id/password (a test id should be created and provided to testers) | System should let you in |  |
| TC2 | {In Login Screen}  Tab into the User and password fields and enter an invalid user id/password | System should not accept and prevent you from entry.  Display error messages:  “Invalid user id/password” |  |
| TC3 | {In Login Screen}  Enter valid user id (see TC1 above) and press “Forgot password” | System should:   * Ask answer to security question and if correct * An e-mail with a temporary password should be sent |  |
| TC4 | {In Login Screen}  Tab into ‘Create New Account’ button | System should navigate user to registration screen for user to create a new account |  |
| TC5 | {In Registration Screen}  Tab into the email, password fields and enter a valid user id/password. Then click on the Sign In button | System should accept the newly created account and navigate the user to the homepage of the application |  |
|  |  |  |  |
| TC6 | {In Registration Screen}  Tab into the email, passwords fields and enter invalid email or password | System should reject the user and display notification to retype the information again.  Error message for email:  “Email not exists or already taken”  Error message for password:  “Password must be at least 8 characters and combine with numeric and alphabetic characters” |  |
| TC7 | Tab into the Phone Number fields and enter alphabetic characters | System should not accept the input and display message:  “Phone Number only consists of number” |  |
| TC8 | The user fails to fill in all fields, or leave a field blank, including Name, Email, Password, Phone Number | System should not accept and display messages: “Please fill in all required fields to create new account” |  |

# 3. Test Cases: “Order”

**Project Name:** University Food System

**Test Case Name:** Order

**Test Case Id**: CSE3310/Fall 2021/Team3/ Order

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case No.** | **Test Case Description** | **Expected results** | **Outcome**  **Pass, Fail, Other (comments)** |
| TC1 | Enter screen and try to place an order without logging in first | System should prevent this action by displaying an error message “Please login first” |  |
| TC2 | After Login, attempt to place an order | System should not allow you to order, instead display message:  “Please choose a restaurant before order” |  |
| TC3 | After Restaurant Selection, click on the drop-down menu, select some food and drink items to order, press submit | System should not allow you to proceed, instead it should take you to the shopping cart to review the order |  |
| TC4 | For each image of the food or drink on the menu, click on the ‘+’ or ‘-‘ buttons | System should update the shopping cart after each change:  If click on ‘+’: increment the item from the user’s order  If click on ‘- ‘: remove the item from the user’s order |  |
| TC5 | Click on ‘View Cart’ to view the purchased items and modify if necessary | The cart should display all items and their price, at the same time display the total amount that the user has to pay |  |
| TC6 | Click on ‘Proceed to Checkout’ button | System should navigate user to payment screen |  |
| TC7 | Click on ‘Continue Shopping’ button | System should let user go back to the main screen of the menu to continue their order |  |
| TC8 | Click on ‘Cancel Order’ | System prompts the question for user to select if they want to cancel their order  If ‘Yes’: Delete the current shopping cart, navigate the user back to the menu  If ‘No’: Remain in current state |  |

# 4. Test Cases: “Search”

**Project Name:** University Food System

**Test Case Name:** Search

**Test Case Id**: CSE3310/Fall 2021/Team3/Search

|  |  |  |  |
| --- | --- | --- | --- |
| **Test**  **Case**  **No.** | **Test Case Description** | **Expected Results** | **Outcome**  **Pass, Fail, Other**  **(comments)** |
| TC1 | If nothing is typed in the search box, search button cannot be clicked. | Search box will require an input to show relevant results. |  |
| TC2 | Tab into the search box and enter a valid food name/ restaurant name | System should display search results. Search should be relevant to searched keyword. |  |
| TC3 | Tab into the search box and enter an invalid item name | System should show the error message - “No results found. Please try again” |  |
| TC4 | If search input is invalid, tab into the search box again | System should allow the user to search for items unlimited times. Even from the results page |  |
| TC5 | If search box is used multiple time, keep track of history | Search box should maintain history of previously entered keywords |  |
| TC6 | The user typed in some characters that matched the name of location of a restaurant | Search box should display suggestions in a list |  |

# 5. Test Cases: “Payments”

**Project Name:** University Food System

**Test Case Name:**  Payments

**Test Case Id**: CSE3310/Fall 2021/Team3/Payments

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case No.** | **Test Case Description** | **Expected results** | **Outcome**  **Pass, Fail, Other (comments)** |
| TC1 | Enter Payment screen user tries to press Next without choosing payment method | System will tell user they must choose between Meal plan or Credit Card |  |
| TC2 | Ask User Payment method  (User press Credit Card) | System should allow User to  Enter 16-digit card number along with expiration date |  |
| TC3 | Ask User Payment method  (User press Meal Plan) | System should allow User to  Enter 10-digit student ID number |  |
| TC4 | If User would like to add more items to cart | User will be prompted back to the Search page |  |
| TC5 | If User would like to delete an item from the cart | User will be prompted back to the Order Page. |  |
| TC6 | If user enters a card number greater than 16 or less than 16, or the card number does not exist | System will tell user “Invalid card number. Please re-enter card’s information” |  |
| TC7 | If user enters a student ID number greater than 10 or less than 110 | System will tell user “Invalid student number. Please enter valid student ID” |  |
| TC8 | If user enters an expiration date number greater than 10 or less than 10 | System will tell user “Invalid expiration date” |  |
| TC9 | {Meal Plan screen}  If the user enters a valid student ID number, such as  Student ID: 1001703030  Click on the “Submit payment” | System will tell user “Purchase by Meal Plan successfully”. Display the number of meals left on the card and dining dollars |  |
| **Test Case No.** | **Test Case Description** | **Expected results** | **Outcome**  **Pass, Fail, Other (comments)** |
| TC10 | {Credit Card Payment screen}  If the user enters a valid card number, security code, and expiration date such as:  Card Number: ‘5431 5894 8091 6212’  CV: 789  Expiration date: 10/24  Click on “Pay for order” button | System will ask user if he/she wants to save the card’s information. If the user agrees, store the user’s card and display the message “Thank you for your purchase” |  |
| TC11 | For both {Meal Plan} and {Credit Card} methods, if the user selects delivery options | System request user to enter phone number and home address. Also, allow user to write additional instructions for delivery |  |
| TC12 | After payment method was entered and completed Delivery method will appear | User will receive an order confirmation code once payment is completed |  |

# 6. Test Cases: “Delivery”

**Project Name:** University Food System

**Test Case Name:** Service Request

**Test Case Id**: CSE3310/Fall 2021/Team3/Delivery

## 

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case No.** | **Test Case Description** | **Expected results** | **Outcome**  **Pass, Fail, Other (comments)** |
| TC1 | User will be directed to Delivery page | System will ask user “Delivery or Pickup” |  |
| TC2 | If delivery is picked | System will allow user to contact driver, restaurant, and check on the ETA |  |
| TC3 | If pickup is picked | System will give user and ETA is pick up order |  |
| TC4 | If delivery was picked and User wants to track order, User enters order number | System will tell user how far the driver is away |  |
| TC5 | If delivery was picked and User wants to track order, User enters order number, and order doesn’t exist | System will tell user “Sorry no order found” |  |
| TC6 | If user wants to contact driver or restaurant | Driver info and Restaurant info will be shown to user |  |

# 7. Test Cases: “Communication”

**Project Name:** University Food System

**Test Case Name:** Communication

**Test Case Id**: CSE3310/Fall 2021/Team3/Communication

|  |  |  |  |
| --- | --- | --- | --- |
| **Test**  **Case**  **No.** | **Test Case Description** | **Expected Results** | **Outcome**  **Pass, Fail, Other**  **(comments)** |
| TC1 | Select some  food and drink items to  order, press submit | System should display message. “Your order has been confirmed” |  |
| TC2 | User placed an order | System will automatically display notifications to user. Example - “Your order is being processed” |  |
| TC3 | Tab on check updates | System will display ETA |  |
| TC4 | User can keep on checking for updates till the order has been delivered | System will display any new updates if available, e.g.- “Driver has picked up your order” |  |
| TC5 | Get delivery notification | System will notify user that order has been delivered |  |
| TC6 | User wants to contact the restaurant or send an email to make special request or complaints | System will attach the restaurant’s phone number and email address below its main page |  |

# 8. Test Cases: “Restaurant and Advertisement”

**Project Name:** University Food System

**Test Case Name:** Restaurant

**Test Case Id**: CSE3310/Fall 2021/Team3/Restaurant and Advertisement

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case No.** | **Test Case Description** | **Expected results** | **Outcome**  **Pass, Fail, Other (comments)** |
| TC1 | {on Homepage}  Scroll through all restaurants | Each test restaurant (6 total) should be viewable and display a name, image, and type of cuisine |  |
| TC2 | {On Homepage}  Click a restaurant box | You should be taken to the restaurant page where the menu can be seen (each restaurant will display 10 test menu items) |  |
| TC3 | {On homepage}  Check a closed restaurant | One restaurant will be closed (to be specified) if clicked a message will appear notifying user |  |
| TC4 | {Restaurant Page}  Scroll through menu display | Each restaurant will display 10 menu items with a name, description, image and price |  |
| TC5 | {On homepage}  Advertisement at bottom of page | One advertisement should appear at the bottom of the page for a restaurant being promoted. If clicked user will be directed to that restaurants page |  |

# 9. Test Cases: “Ratings”

**Project Name:** University Food System

**Test Case Name:** Ratings

**Test Case Id**: CSE3310/Fall 2021/Team3/Ratings

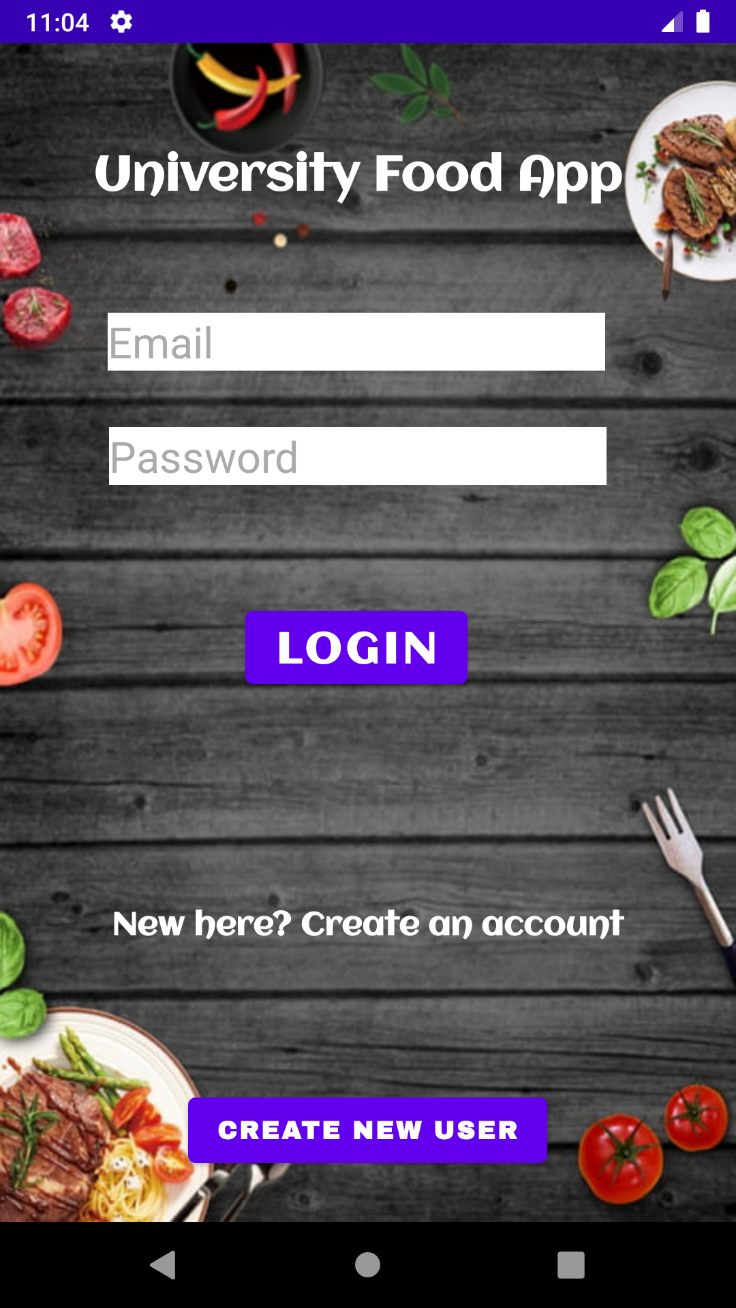
## 

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case No.** | **Test Case Description** | **Expected results** | **Outcome**  **Pass, Fail, Other (comments)** |
| TC1 | {On ratings page}  Scroll through ratings boxes | Each previous order will be seen and given the option to rate if an order is already rated number of stars will be shown and order won’t be clickable |  |
| TC2 | {On ratings page}  Click a rating | Clicking a rating will direct you to the rating page |  |
| TC3 | {On rate page}  Complete rating giving number of stars and description of why | On the ratings page completed rating should show number of stars and no longer be clickable |  |

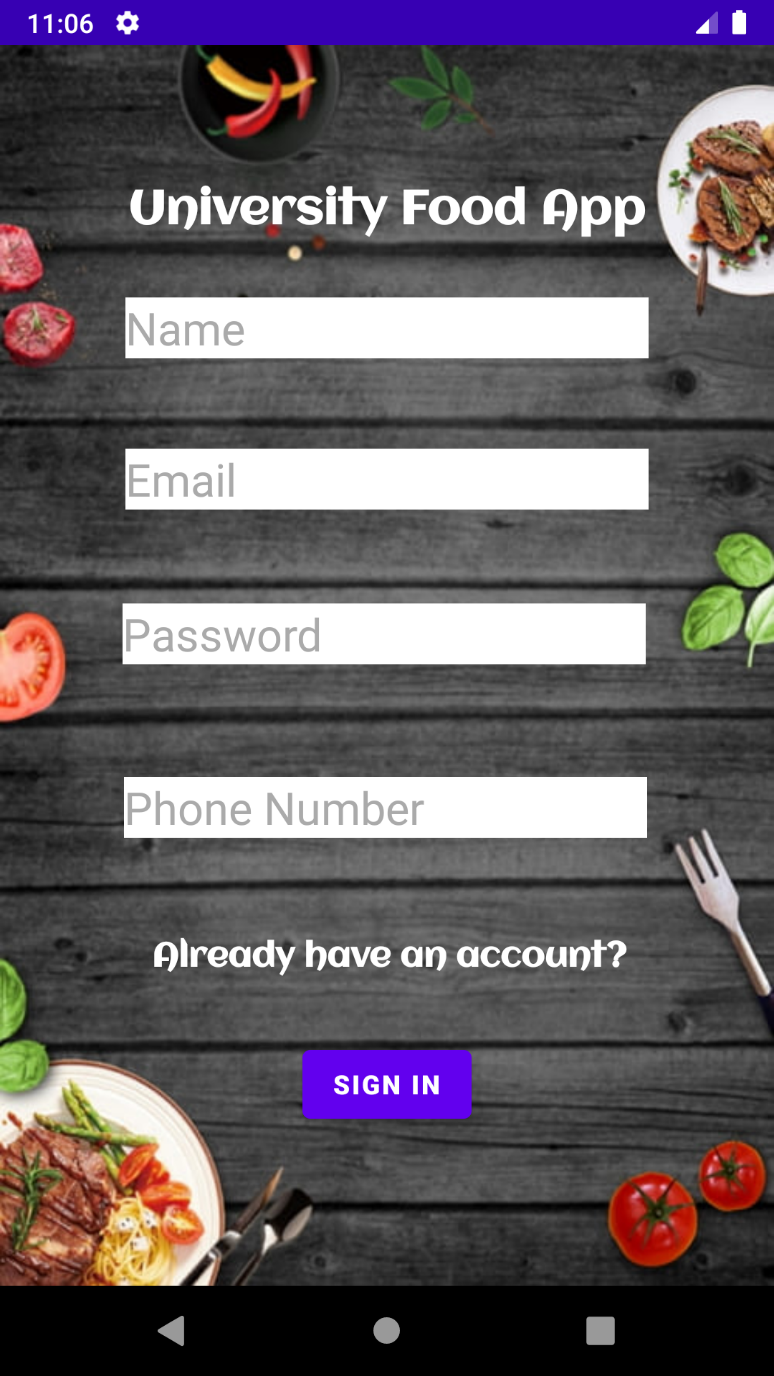
**User Manual**

|  |  |
| --- | --- |
| **Index No.** | **Content** |
| **1.** | **Login Page** |
| **2.** | **Registration Page** |
| **3.** | **Home Page** |
| **4.** | **Menu Page** |
| **5.** | **Cart Page** |
| **6.** | **Payment Page** |
| **7.** | **Meal Plan Page** |
| **8.** | **Credit Card Page** |
| **9.** | **Delivery Page** |
| **10.** | **Ratings Page** |

**Login Page**



Upon opening the app the user is taken to the login page. The user enters their registered email and password and presses login to enter the homepage. Alternatively, the user can hit create new user if they do not have an account.

**Registration Page**  


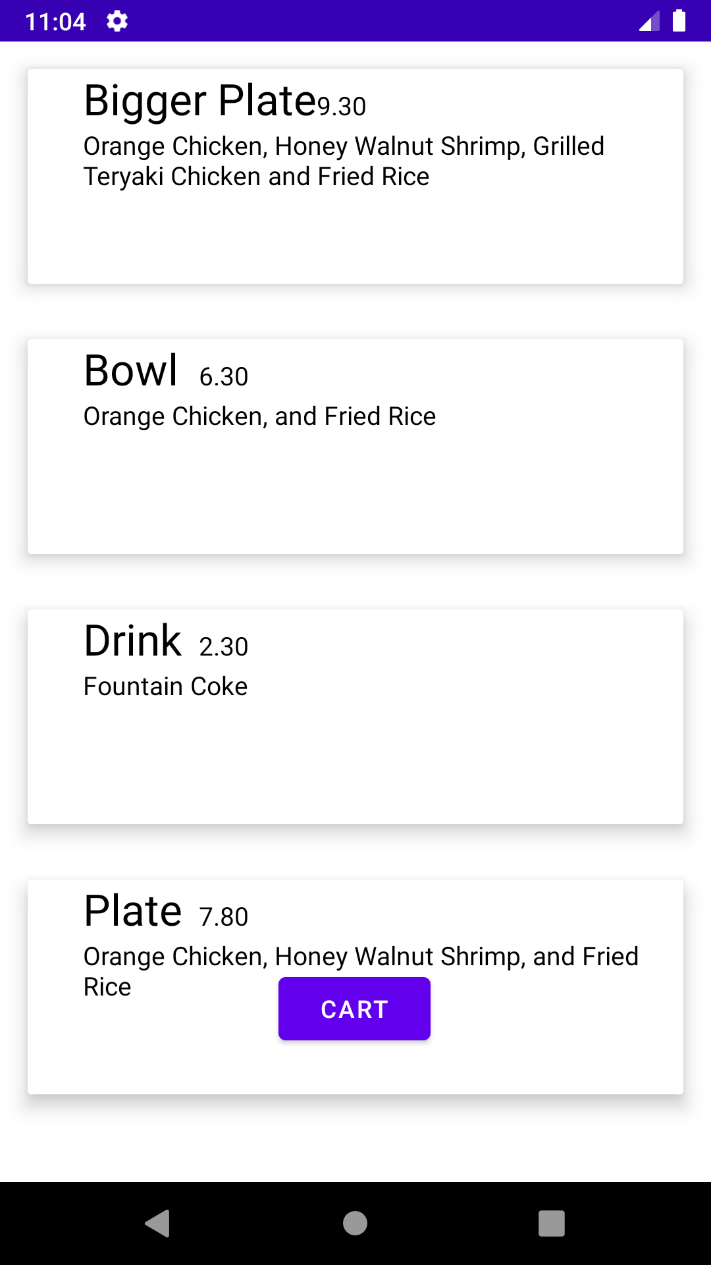
On the new user page, the user enters the specified information. An account is created in our firebase server and is directed to the homepage.

**Home Page**



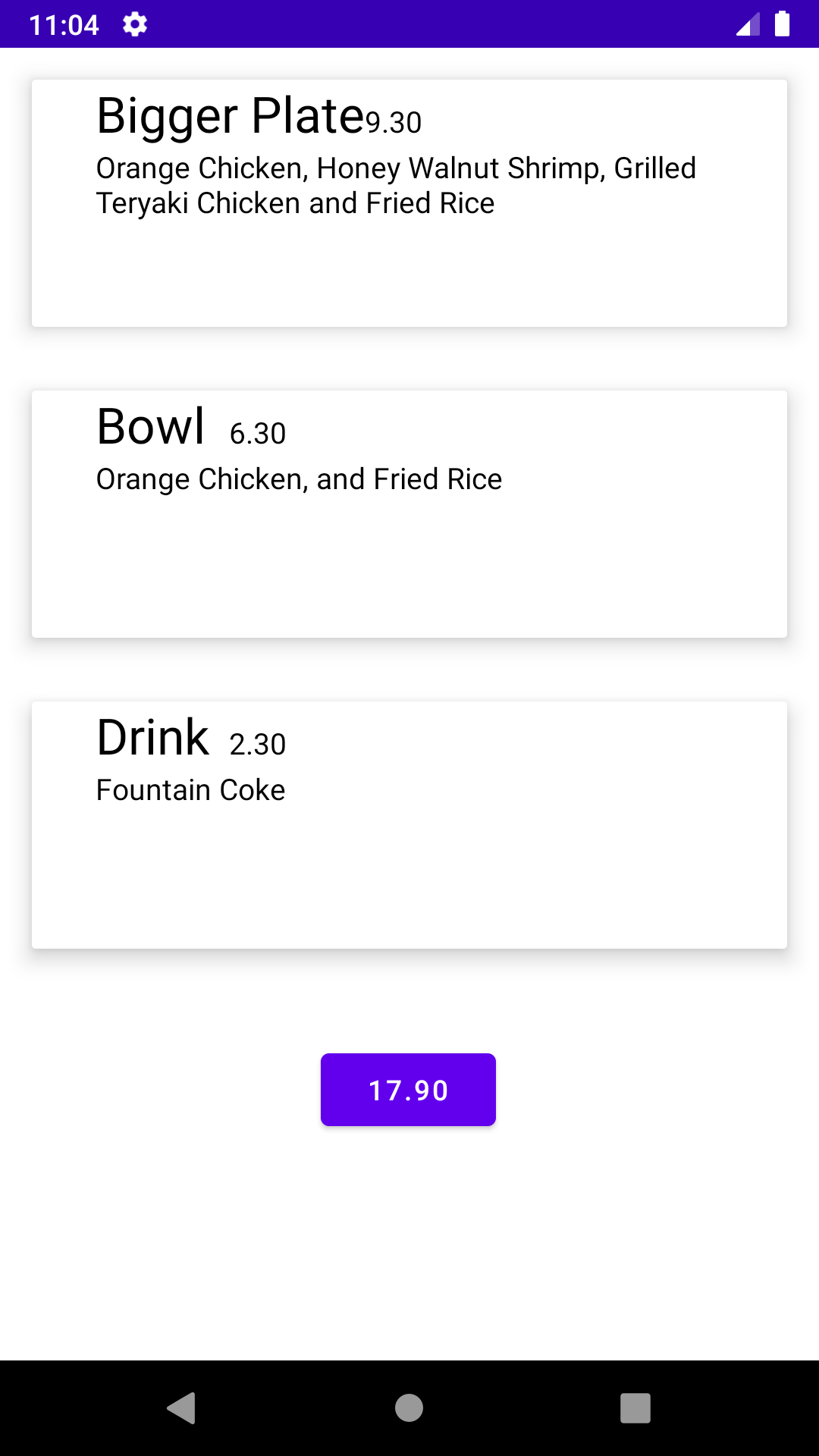
On the homepage all restaurants are displayed and can be clicked to go to the restaurant’s menu and start an order. Estimated times for deliveries are showed for each restaurant. An ad can be seen at the bottom for one of the restaurants.

**Menu Page**



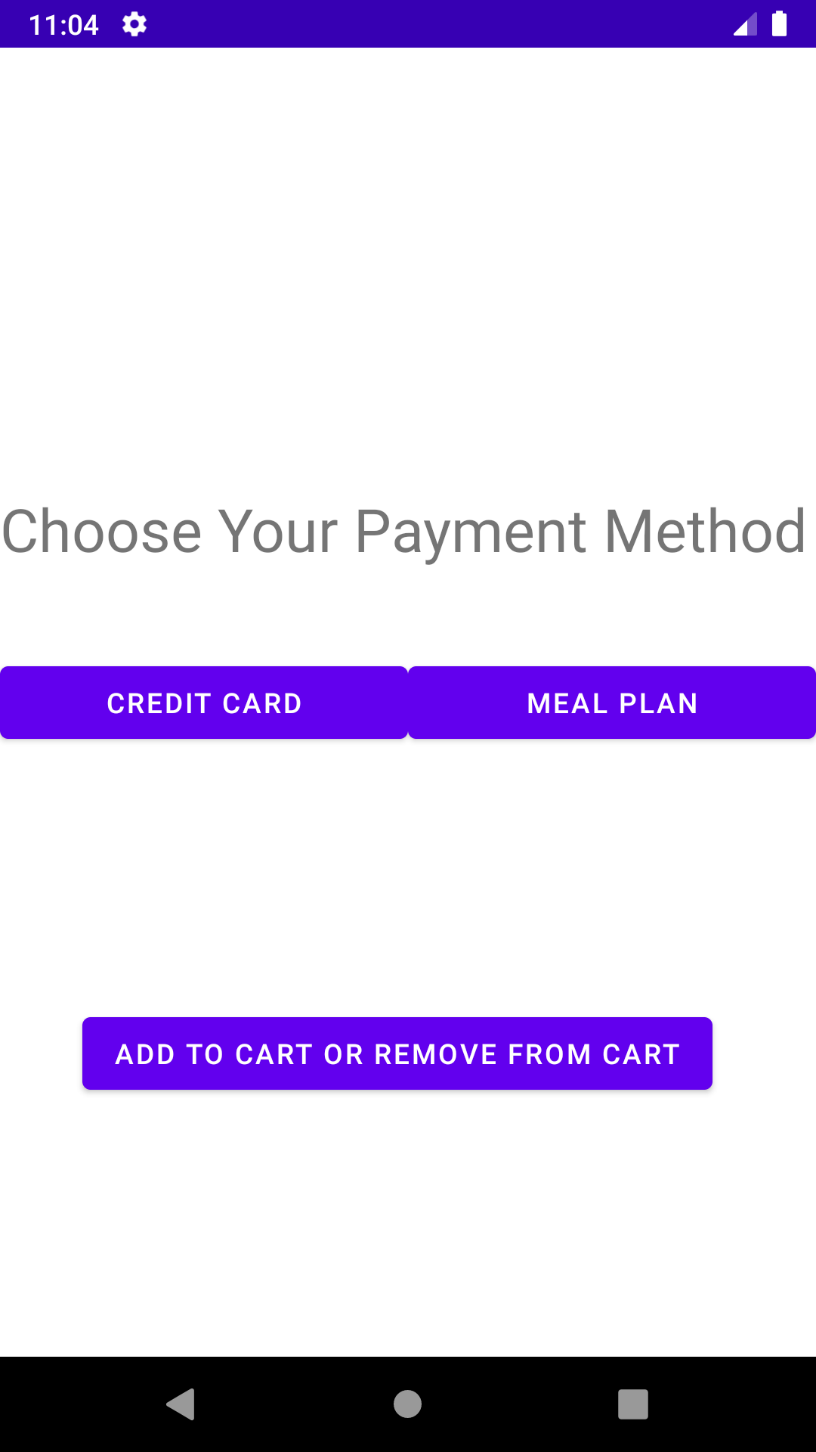
The menu page shows all items on the menu from the database. To add an item to the user’s cart, click that item and a toast is displayed. Once all items are selected cart can be clicked to continue the order.

**Cart Page**



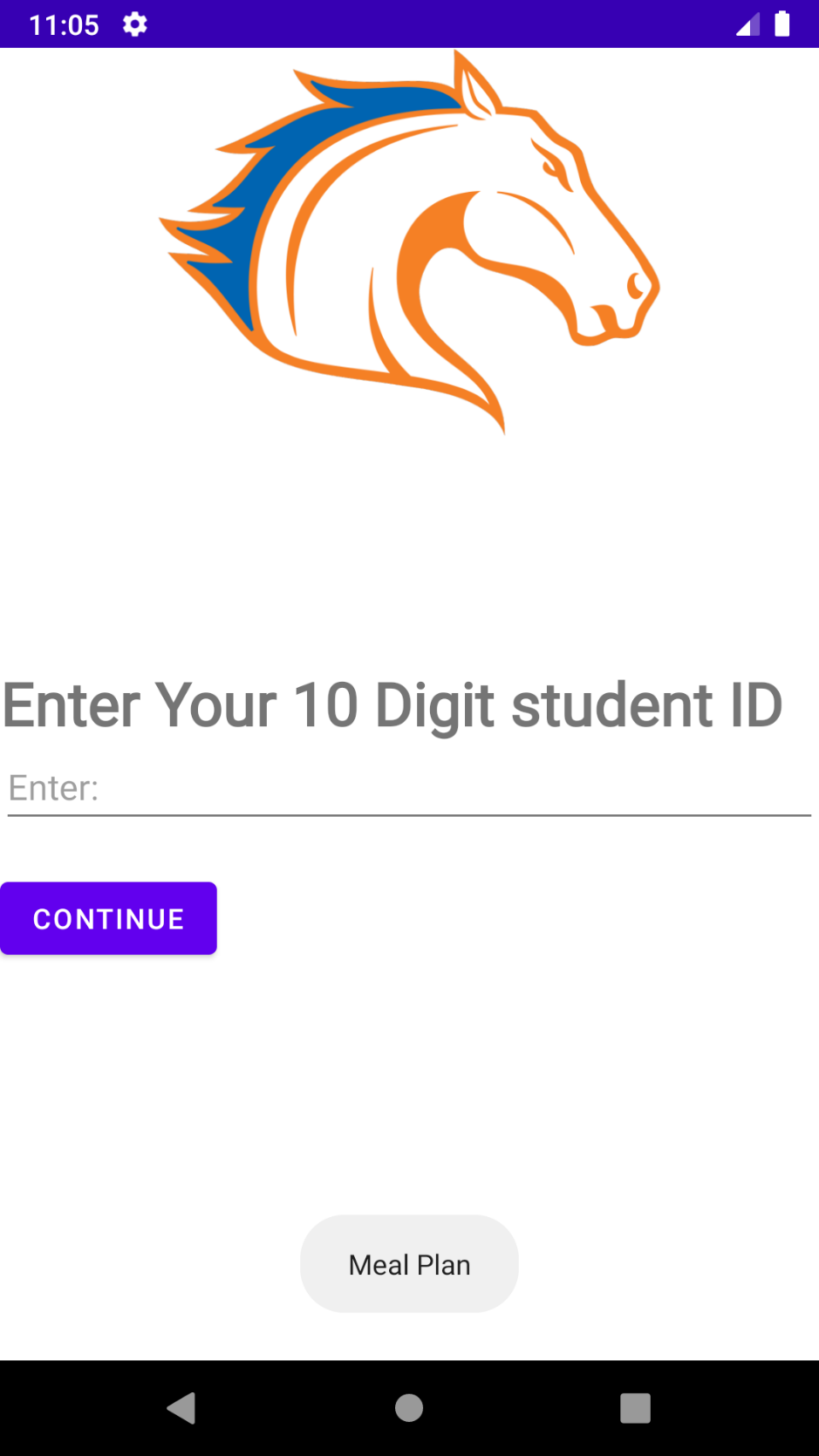
The cart page displays all selected items and the total at the bottom. The user clicks the total to move to payment.

**Payment page**



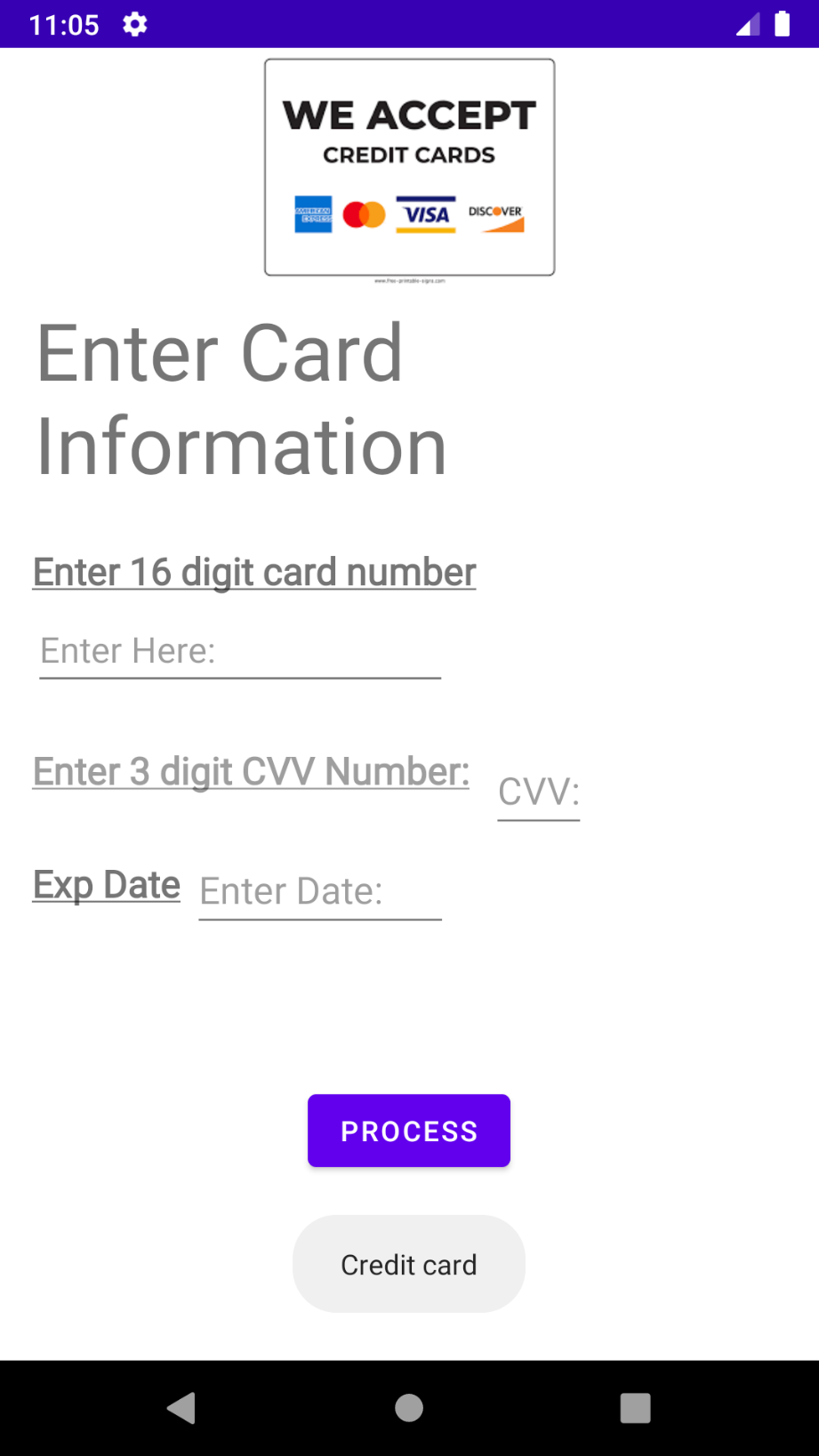
From the payment page the user chooses credit card or meal plan account and is directed to the respected pages.

**Meal Plan Page**



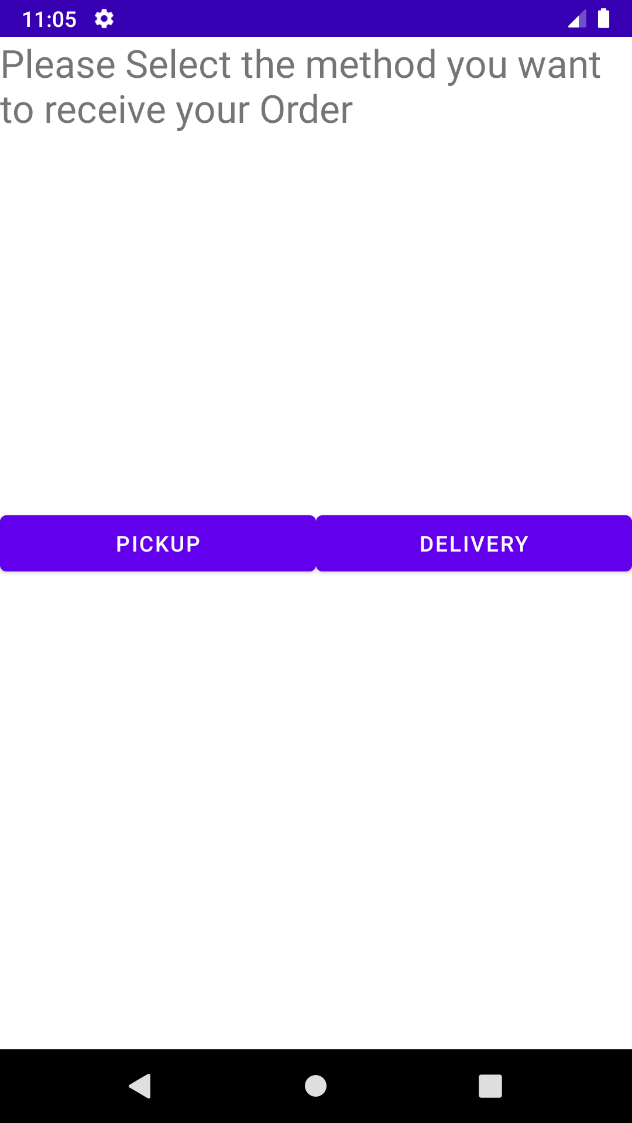
On the meal plan page the user enters their 10 digit numeric ID and continue to move to deliver.

**Credit Card Page**



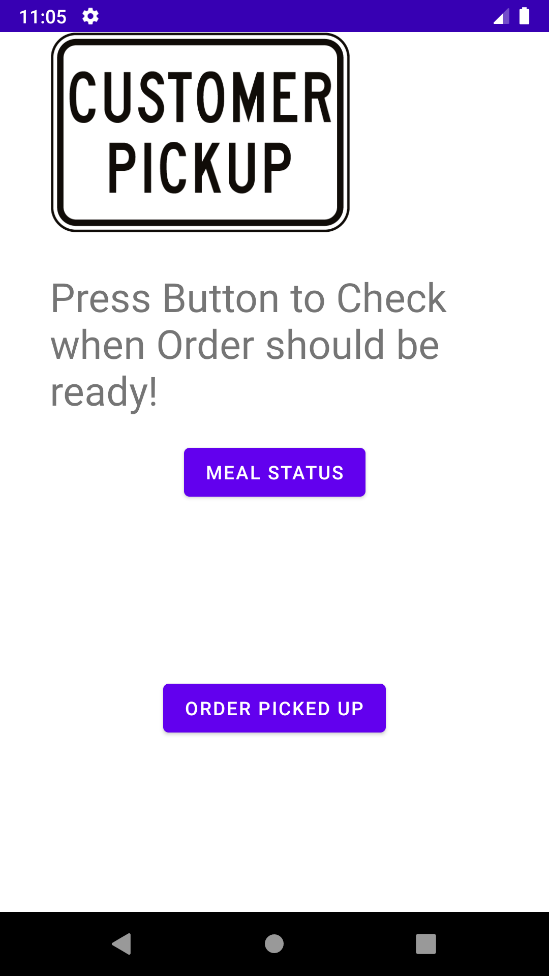
On the credit card page the user enters their needed card info and clicks process to move to the delivery and pick up pages.

**Delivery and Pick up Page**



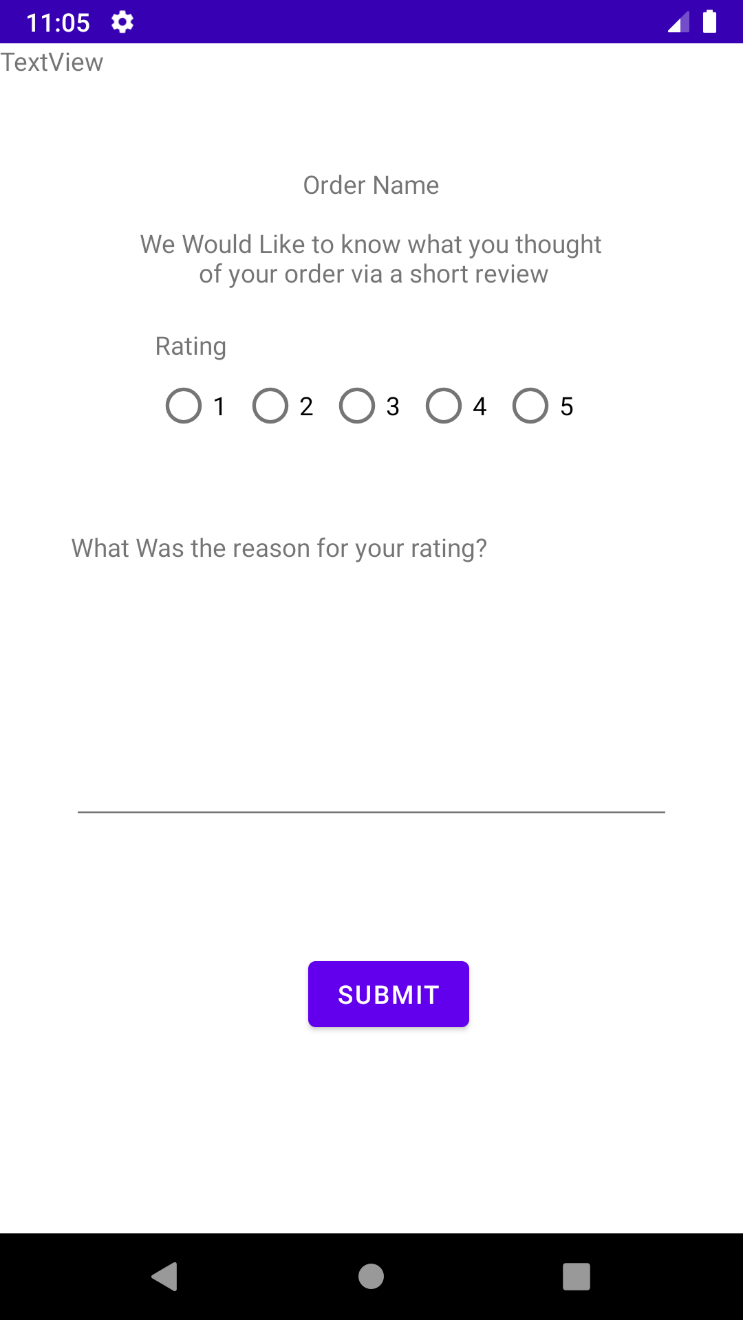
From the delivery and pickup page the user can choose between delivery and pick up and is moved to the respective pages.

**Delivery and Pickup pages**

From the delivery and pickup pages orders can be checked, drivers numbers can be obtained from the contact button and when the order is received the user can click the order reviewed or picked up buttons to move to reviews.

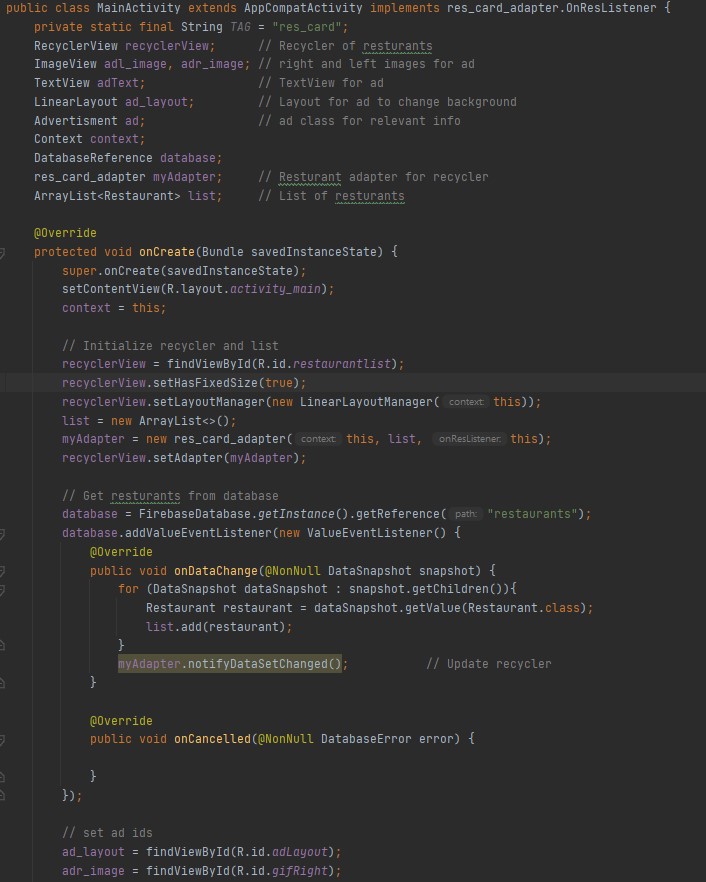
**Ratings page**



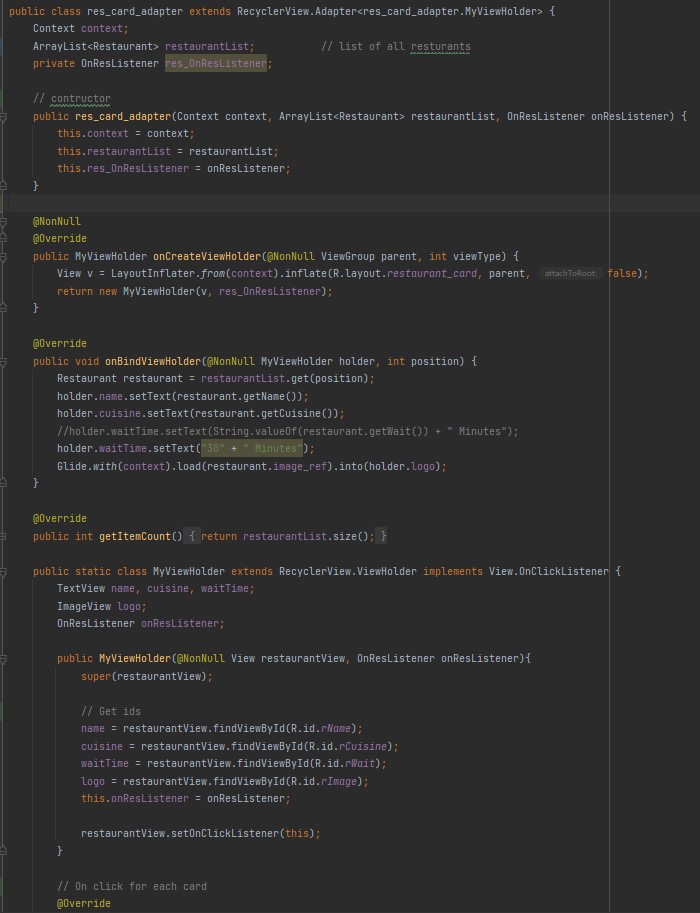
From the ratings page the user can give a number rating and type a short paragraph giving their reason for their review. When the user hits submit, they are sent back to the homepage

**SELECTED SOURCE CODE**

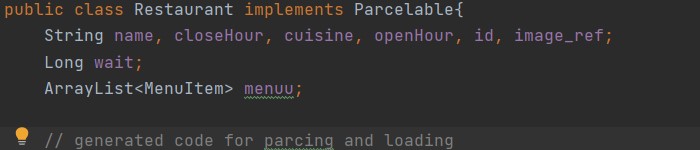
**Main Activity (Homepage) Source Code**

****

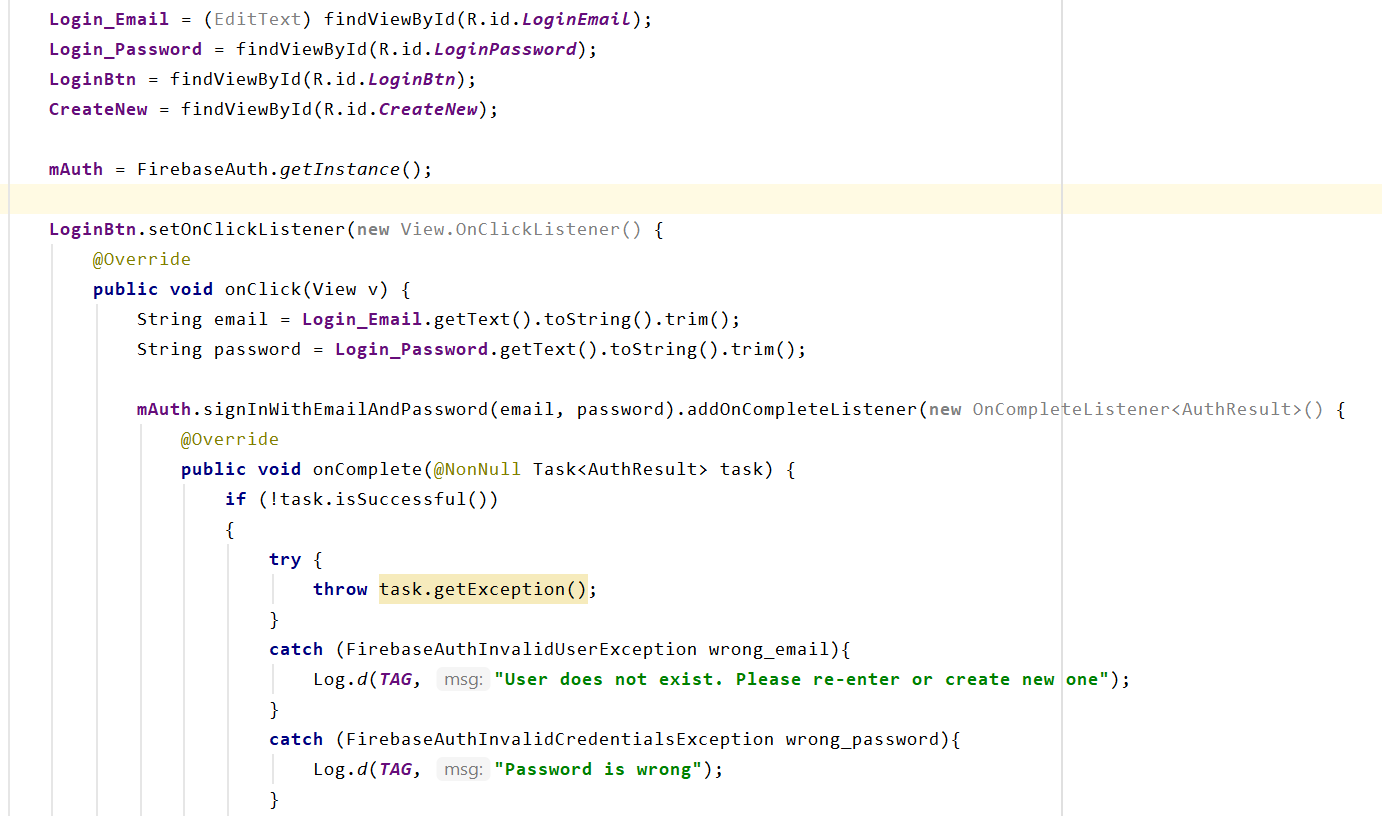
**Restaurant Adapter Source Code**

****

**Restaurant class Source Code**

****

**Login Source Code**

****

**Registration Source Code**

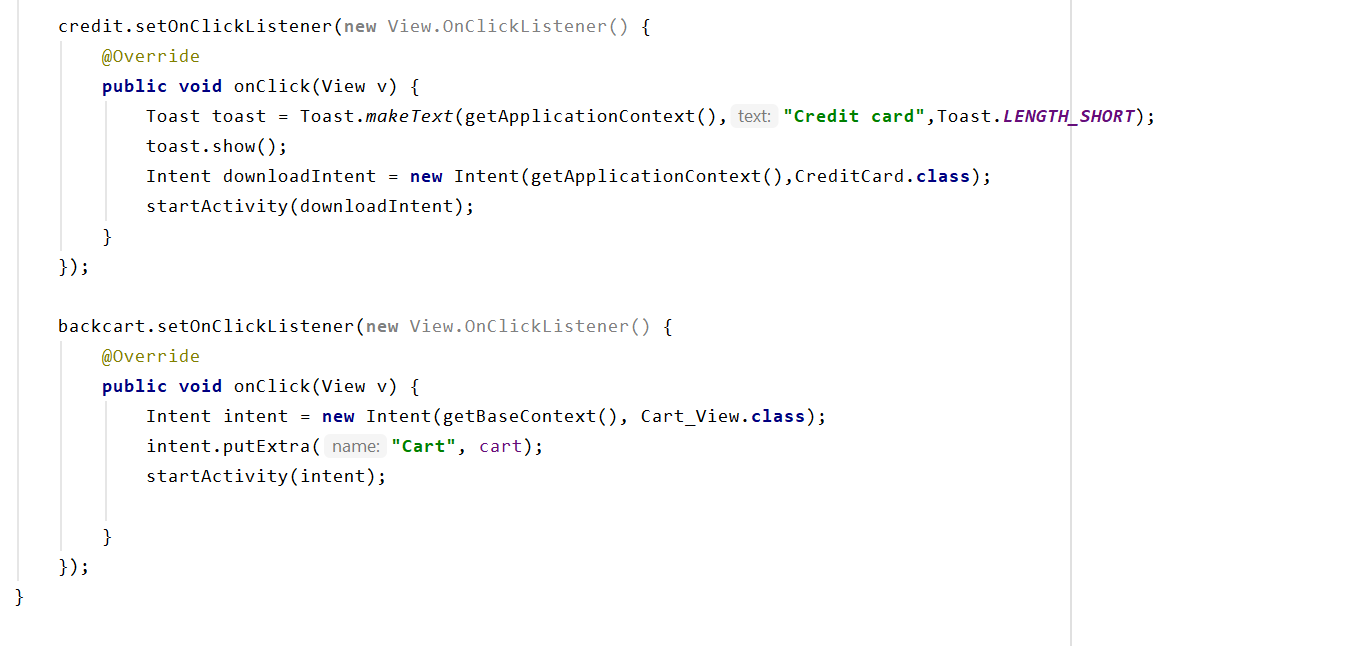
****

**Delivery Source Code**

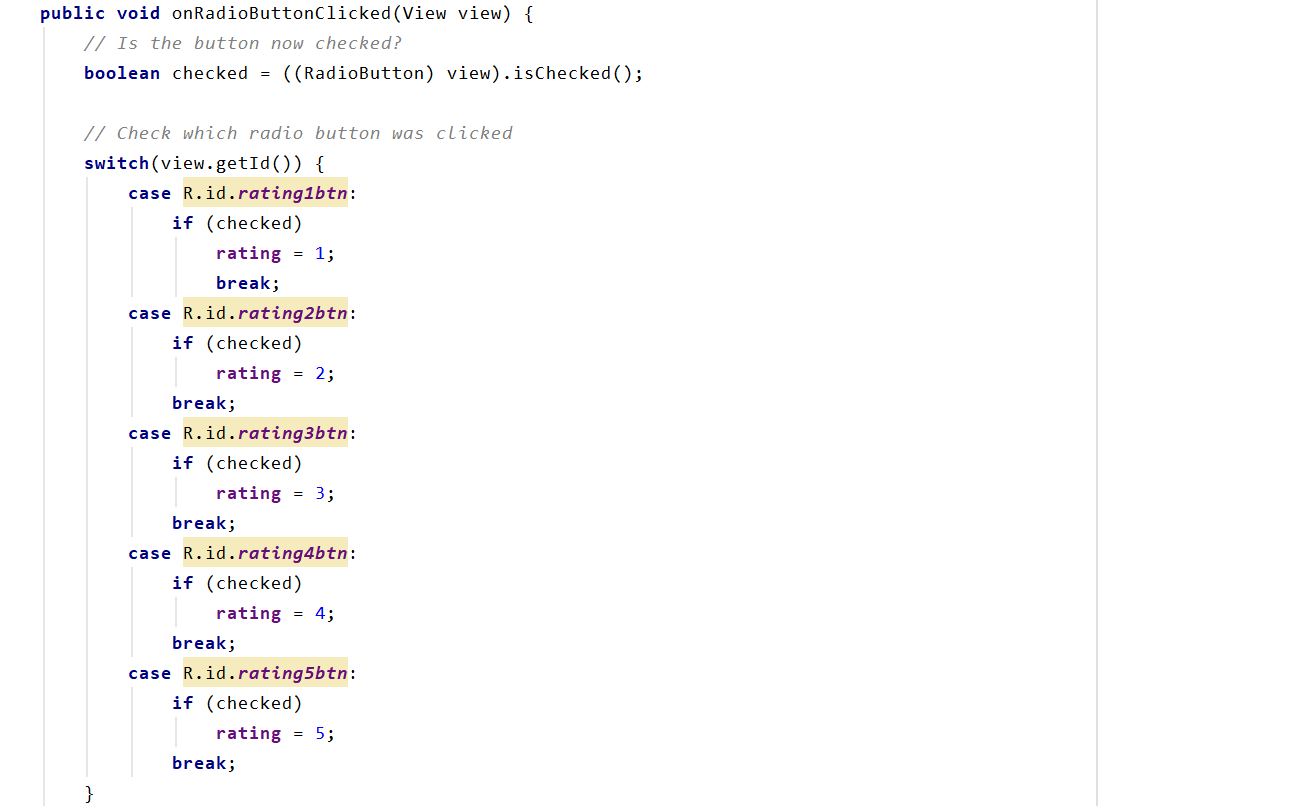
****

**Payment Source Code**

****

****

**Ratings Source Code**

****