**HomeCook : Food Delivering System**

**Version 1.0**

Submitted by

**Shrirang Pinjarkar**

Third Year Information Technology

**ID No. : 151080008**

Submitted in partial fulfillment of

Advanced Software Engineering

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| 1 | Executive Summary | 3 |
| 1.1 | Project Overview | 3 |
| 1.2 | Purpose and Scope of this Specification | 3 |
| 2 | Product/Service Description | 3 |
| 2.1 | Product Context | 3 |
| 2.2 | User Characteristics | 4 |
| 2.3 | Assumptions | 4 |
| 2.4 | Constraints | 4 |
| 2.5 | Dependencies | 4 |
| 3 | Specific Requirements | 4 |
| 3.1 | External Interface Requirements | 4 |
| 3.1.1 | User Interfaces | 4 |
| 3.1.2 | Hardware Interfaces | 5 |
| 3.1.3 | Software Interfaces | 5 |
| 3.1.4 | Communications Interfaces | 5 |
| 3.2 | Functional Requirements | 5 |
| 3.3 | Use Case Diagram | 6 |
| 3.4 | Use Case Specification | 7 |
| 3.5 | Non-functional Requirements | 14 |
| 3.5.1 | Performance | 14 |
| 3.5.2 | Reliability | 15 |
| 3.5.3 | Availability | 15 |
| 3.5.4 | Security | 15 |

**1.Executive Summary**

**1.1 Project Overview**

HomeCook is an android application which is a food delivering system. It is actually delivers dishes which are prepared by the homecook. System will interact with homecook, customers, manager and delivery boy. It will help in delivering food from homecook address to customers address.

**1.2 Purpose and Scope**

The major purpose of the proposed system is to provide new food dishes to the customers and to help homecook to work in their home. Most of the homecook don’t want to work in restaurant. They want to work in their home and get paid for their creative dishes. People get bored of restaurant food. So they love to try new dishes.

**Scope**

* This system is useful for food delivery.
* People will get attracted towards different food dishes.
* Homecook will love to apply for this job because they don’t need to go out for job.

**2 Project Service Description**

**2.1 Project Context**

Most of the food delivering system provides food from the restaurant. People get bored this day from hostels food. Also, most of the cooks don’t want work in restaurant.

**2.2 User Characteristics**

User who is going to use doesn’t require special knowledge of system. Anyone who uses android mobile will be able to use this application.

Below users will operate the system for their particular task.

* **Homecook :** To add food menu and accept orders from customers.
* **Customers :** To place food orders.
* **Admin :** To manage delivery of orders and contact delivery boy for delivery of the food.
* **Delivery Boy :** To get delivery order from the admin.
  1. **Assumptions**

The proposed android system is an online application so the organization should have network and user can access this remotely.

**2.4 Constraints**

Regarding development constraints for this system, we have studied some other food delivering android application. So we will not require much time as we can refer these models.

**Software Constraints :**

We have reviewed the currently working system and we have chosen below platforms.

* Android : It is an open source operating system which is very easy to use.
* Firebase : It is open source database provided by Google.
  1. **Dependancies**

Android application will require internet to connect with system.

**3. Specific Requirements**

**3.1 External Interface Requirements**

**3.1.1 User Interfaces**

* GUI along with meaningful Frames and buttons
* Order status, details and menu are generated as per the requirement.
* Different payments method are available.

**3.1.2 Hardware Interfaces**

|  |  |
| --- | --- |
| Hardware Environment | Android Mobile |
| System Configuration | RAM-1GB HDD-8GB |
| Operating System | Android OS |

**3.1.3 Software Interfaces**

|  |  |
| --- | --- |
| Front End | XML |
| Back End | Android Java & Firebase |

When invalid inputs are given to the modules then the error messages will be popped up in order to inform the user that the input provided is not taken by the database. When incomplete information is provided by the user and the user tries to submit the form in order to store the details in the database the system will pop up a message box asking the user to enter all the details required.

**3.1.4 Communications Interfaces**

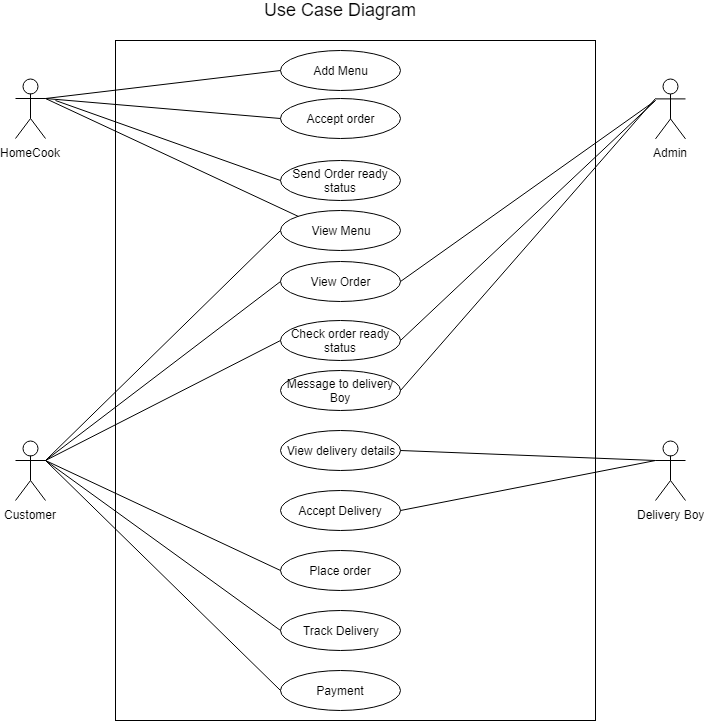
Android mobile should have to be connected to the internet.

**3.2 Functional Requirements**

Homecook android application will have the following functions.

* Provide food delivering from homecook address to customer address.
* Try new dishes from menu.
* Homecook can easily add new dishes in the menu.

**3.3 Use Case Diagram**

****

* 1. **Use Case Specification**

1. **Use case :**  Add menu

**Description :** Homecook will add new food dishes in the menu.

**Actors involved :** Homecook

**Precondition :** Homecook must have valid UserID and password.

**Basic Flow of Events :**

|  |  |
| --- | --- |
| **Actor** | **System** |
| **1.**Homecook will click on add menu button. |  |
| **2.**He will enter all details of the dish. |  |
|  | **3.**App will check if all minimum required details of a dish are provided or not.  (Dish name, price, availability, time) |
| **4.**If all details are valid, he will submit. |  |
|  | **5.**App will displays menu added status. |

**Alternate Flow :**

|  |  |
| --- | --- |
| **Actor** | **System** |
|  | **3.** App will check if all minimum required details of a dish are provided or not. |
|  | **3.1** If all details are not provided, information need to be filled is shown. |
|  | **3.2** Flow is returned to input interface |

**Post Condition :** App will display the added menu details.

**Special Condition :** None.

1. **Use case :**  View menu

**Description :** User will see the menu available.

**Actors involved :** Homecook, Customers

**Precondition :** User must have valid UserID and password.

**Basic Flow of Events :**

|  |  |
| --- | --- |
| **Actor** | **System** |
| **1.**User will click on request for menu. |  |
|  | **2.** Check if menu is available or not**.** |
|  | **3.**App will displays available menu. |

**Alternate Flow :**

|  |  |
| --- | --- |
| **Actor** | **System** |
|  | **2.**Check if menu is available. |
|  | **2.1** If menu is not available, app will give the not available status. |

**Post Condition :** None.

**Special Condition :** None.

1. **Use case :**  Accept order

**Description :** When customer give order, homecook will accept order to confirm that order is available.

**Actors involved :** Homecook

**Precondition :** Customer must have ordered food.

**Basic Flow of Events :**

|  |  |
| --- | --- |
| **Actor** | **System** |
| **1.**HomeCook will click on request to check if there are any orders for him/her. |  |
|  | **2.**System will check if he has some orders or not. |
|  | **3.** If yes, then it will displays order details. |

**Alternate Flow :**

|  |  |
| --- | --- |
| **Actor** | **System** |
|  | **2.**Check for orders |
|  | **2.1** If no, then displays no order status. |

**Post Condition :** None.

**Special Condition :** None.

1. **Use case :**  Send order ready status.

**Description :** When the order is ready, homecook will send ready status to the admin.

**Actors involved :** Homecook

**Precondition :** Homecook must have any order.

**Basic Flow of Events :**

|  |  |
| --- | --- |
| **Actor** | **System** |
| **1.**Homecook will click on view orders. |  |
|  | **2.**System will displays order lists. |
| **3.**Homecook will select order from the list. |  |
| **4.**Homecook will click on update to change status. |  |
|  | **5.**System will update status. |
|  | **6.**System shows updated status of order |

**Alternate Flow :** None.

**Post Condition :** None.

**Special Condition :** None.

1. **Use case :**  View Orders.

**Description :** User will see his/her orders.

**Actors involved :** Homecook, Customer, admin

**Precondition :** Customer must have order something. Homecook must have some requested order. Admin can see all orders placed.

**Basic Flow of Events :**

|  |  |
| --- | --- |
| **Actor** | **System** |
| **1.**User will request for view orders. |  |
|  | **2.**Check if any orders available. |
|  | **3.**If yes, then app will show all or particular orders. |

**Alternate Flow :**

|  |  |
| --- | --- |
| **Actor** | **System** |
|  | **2.**Check if any orders available. |
|  | **2.1**Check if any orders available. |

**Post Condition :** None.

**Special Condition :** None.

1. **Use case :**  Check order ready status.

**Description :** Admin will see if the order is ready or not.

**Actors Involved :** Admin

**Precondition :** Customer must place some order.

**Basic Flow of Events :**

|  |  |
| --- | --- |
| **Actor** | **System** |
| **1.**Admin will click on check if there are any orders placed. |  |
|  | **2.**System will check for orders. |
|  | **3.**If yes, then app will displays all orders which are ready to delivered. |

**Alternate Flow :**

1. **1** If not, system will show no orders status.
2. **1** If not, it will displays all orders with statement no order is ready yet.

**Post Condition :** None.

**Special Condition :** None.

1. **Use case :**  Message to delivery boy.

**Description :** Admin will send order details with message when order is ready.

**Actors Involved :** Admin

**Precondition :** Order must be ready.

**Basic Flow of Events :**

|  |  |
| --- | --- |
| **Actor** | **System** |
| **1.**Admin will click on check if any order is ready. |  |
|  | **2.**System will check for orders. |
|  | **3.** If yes, then it will display ready orders. |
| **4.**Admin will select order and send it to the delivery boy. |  |
|  | **5.** System will create message and send it to the delivery boy. |
| **6.**Admin can chat with delivery boy. |  |

**Alternate Flow :**

1. **1** If no order is placed, system will displays no order status.

**Post Condition :** None

**Special Condition :** None.

1. **Use case :**  View Delivery Order Details

**Description :** Delivery boy will request for order details.

**Actors Involved :** Delivery boy

**Precondition :** Admin must send order details to delivery boy.

**Basic Flow of Events :**

|  |  |
| --- | --- |
| **Actor** | **System** |
| **1.**Admin will send order details to delivery boy. |  |
|  | **2.**System will send notification to delivery boy. |
| **3.** Delivery boy will open chat tab. |  |
| **4.** Delivery boy will see order details. |  |

**Alternate Flow :**

**1.1** If admin didn’t send any order, no order status will get displayed.

**Post Condition :** System will displays order details.

**Special Condition :** None.

1. **Use case :**  Accept Delivery

**Description :** After admin sends order details, delivery boy must accept order so that admin will get confirmation.

**Actors Involved :** Delivery Boy

**Precondition :** Admin must send order details.

**Basic Flow of Events :**

|  |  |
| --- | --- |
| **Actor** | **System** |
|  | **1.** App will notify delivery boy that order has to be delivered. |
| **2.**Delivery boy will see order details. |  |
| **3.** Delivery boy will send confirmation to admin. |  |

**Alternate Flow :** None.

**Post Condition :** None.

**Special Condition :** None.

1. **Use case :**  Tracking Order.

**Description :** Customer can see their order tracking status.

**Actors Involved :** Customer

**Precondition :** Customer must have some order placed.

**Basic Flow of Events :**

|  |  |
| --- | --- |
| **Actor** | **System** |
| **1.**Customer will select view orders. |  |
|  | **2.**System will displays all order list. |
| **3.**He will select particular order from the list. |  |
|  | **4.** App will displays order details. |
| **5.** Customer will click on track delivery button. |  |
|  | **6.** App will displays tracking status. |

**Alternate Flow :** None.

**Post Condition :** System displays tracking status.

**Special Condition :** None.

1. **Use case :**  Payment

**Description :** Customer will pay for their orders

**Actors Involved :** Customers

**Precondition :** Customer must place some order.

**Basic Flow of Events :**

|  |  |
| --- | --- |
| **Actor** | **System** |
| **1.**Customer after selecting order will select payment option. |  |
| **2.**Customer will request to place order. |  |
|  | **3.**System will displays payment gateway for payment. |
| **4.**Customer will pay from gateway. |  |
|  | **5.**System will displays payment  Confirmation status. |

**Alternate Flow :**

1. **1** If payment is failed, app will displays payment failed status.

**5.2** App will return to payment interface.

**Post Condition :** None.

**Special Condition :** None.

1. **Use case :**  Place order.

**Description :** Customer can place order from the menu.

**Actors Involved :** Customer.

**Precondition :** Menu must be available.

**Basic Flow of Events :**

|  |  |
| --- | --- |
| **Actor** | **System** |
| **1.**Customer will click on view menu. |  |
| **2.**He will select order from the list. |  |
| **3.** He then place order. |  |
|  | **4.**System will accept the order  and displays order accepted status. |

**Alternate Flow :** None.

**Post Condition :** App will displays orders placed status.

**Special Condition :** None.

1. **Non-Functional Requirements** 
   1. **Performance**

Easy food delivery can be done.

**Static Requirements**

These requirements do not impose any constraints on the execution characteristics of the system. They are:

**1) Number of Users :**

The Homecook application can be used by numbers of users. It depends on how better service is.

**2) Number of homecook :**

The number of homecooks may vary from city to city.

**3.5.2 Reliability**

The application will not be able to connect to the centralized database if user has no internet connection on their mobile and if server is being down due to a hardware or software failure.

So the system needs additional servers so that if one of the server has too much traffic, first server will put half of the load on the second one. In case of failure of any single server won’t affect the total system

**3.5.3 Availability**

System should be from 9AM to 12PM available.

**3.5.4 Security**

The security requirements deal with the primary security. Password based login.

Firewall in the network can control the incoming attacks. Access to the database is restricted to specific user only.