Software Requirements Specification

for

<Solution Finder>

Version 1.0

Prepared by

<Waleed Umar (FA18-BSE-102) >

<Ahmad bin Ashraf (FA18-BSE-107) >

<M. Saqib khan (FA18-BSE-113) >

Supervised by

<Sir Atique Ahmad>

Department of Computer Science, CU, Islamabad.<27/12/2020>

**Table of Contents**

Revision History iii

1. Introduction 1

1.1 Purpose 1

1.2 Scope 1

2. Overall description 1

2.1 Product perspective 1

2.2 Operating environment 1

2.3 Design and implementation constraints 1

3. Requirement identifying technique 2

3.1 Use Case Diagram 2

3.2 Use Case Description 2

4. Specific Requirements 4

4.1 System feature X 4

5. Quality attributes 5

5.1 Usability 6

5.2 Performance 6

6. External interface requirements 6

6.1 User interfaces 6

6.2 Software interfaces 7

6.3 Hardware interfaces 7

6.4 Communications interfaces 7

7. Project Gantt chart 8

8. References 8

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

**Application Evaluation History**

|  |  |
| --- | --- |
| **Comments (by committee)**  **\*include the ones given at scope time both in doc and presentation** | **Action Taken** |
|  |  |
|  |  |

Supervised by

<Supervisor’s Name>

Signature\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Introduction

The introduction presents an overview to understand how the SRS is organized and how to use it.

## Purpose

This SRS document contains functional and non-functional requirements of Solution Finder, an android based application designed especially for students. The purpose of the application is to provide all in one platform where students can solve their problems like buying selling books, use transport services, find accommodation and get grocery.

## Scope

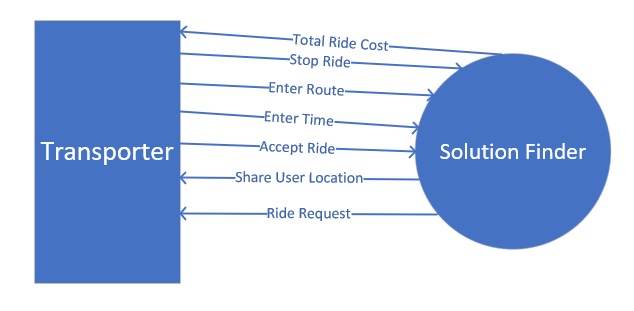
For students who find it difficult to find a place to live, buy new books, find suitable transport according to their financial situation, the Solution Finder is an Android based social application where students/ users can share their problems and find suitable resources to solve their issues. With help of their desired location, they can find the resources available corresponding to their problems.

The software solution will provide users a chance to rent out their places to others for a specific period of time, share a cab or find nearest transport facility helping them save both time and money. Students can chose a hostel room from their home instead of coming to the area and searching them. This will help in cutting their travelling cost. Unlike other apps that only provide either the option of finding a property or booking a ride but not both, our product will provide you all the options in one platform. Moreover, while booking transport you will have a fixed price for your route unlike others where total cost depend on traffic and peak factor*.*

# Overall description

## Product perspective

Solution Finder is a new system that aims at solving problems for students in particular and public in general. In past people have to use different apps for food ordering, ride ordering and for booking accommodation. With Solution Finder you can use all these services in one platform. The system is expected to evolve over further updates and to expand in more cities over time.



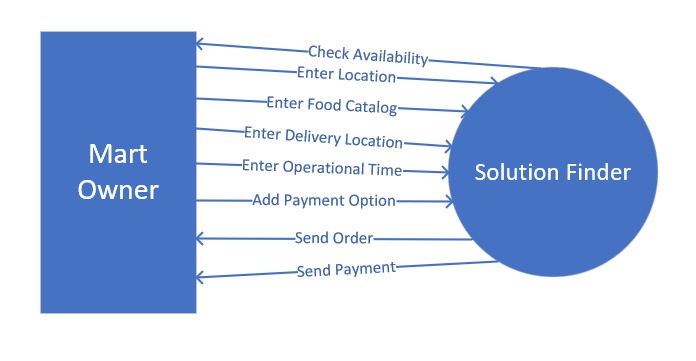


Figure 1.1: Mart Owner Level 0 DFD

Figure 1: Transporter Level 0 DFD

## 

Figure 2 Student Level DFD

## 2.2 Operating environment

|  |  |
| --- | --- |
| OE-1 | The system shall work on android devices with OS version greater 5.0 |
| OE-2 | System shall be available for use in all metropolitan cities of Pakistan in first stage |
| CO-3 | Android X shall be used as an android library |

## Design and implementation constraints

|  |  |
| --- | --- |
| CO-1 | The system shall use firebase as database to store data |
| CO-2 | Coding shall be done using Java and Flutter as languages |
| CO-3 | Android X shall be used as an android library |

# Requirement identifying technique

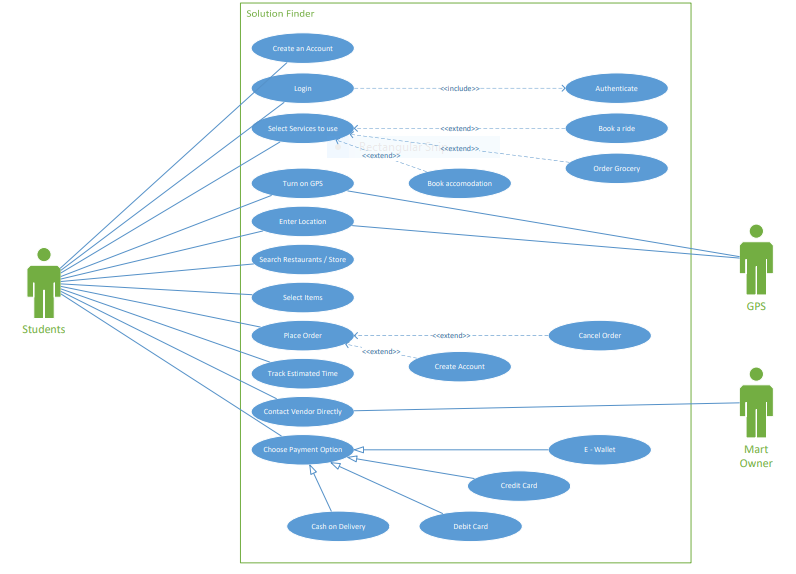
This section describes the requirements identifying technique(s) which further help to derive functional requirements specification. The selection of the technique(s) will depend on the type of project. For instance,

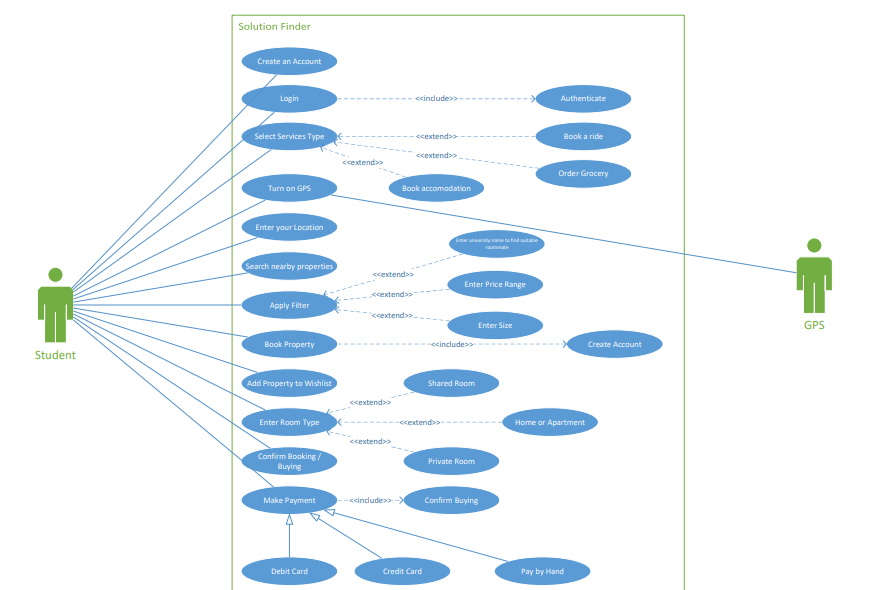
* **Use case** is an effective technique for interactive end-user applications
* **Event- response tables** is for real time system and
* **Story boarding** for graphically intensive applications.

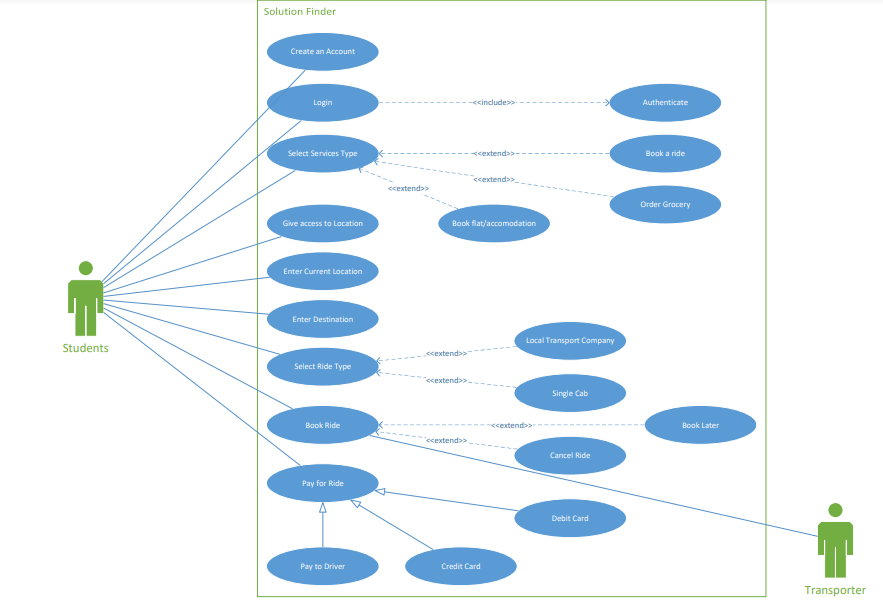
In addition to above, the projects involving data warehouses, batch processes, hardware devices with embedded control software, and computationally intensive applications required to follow other suitable techniques. Such techniques are described further in Chapter 12, “A picture is worth 1024 words.” For documenting this section let consider identifying requirements through use case as an example.

## Use case diagram

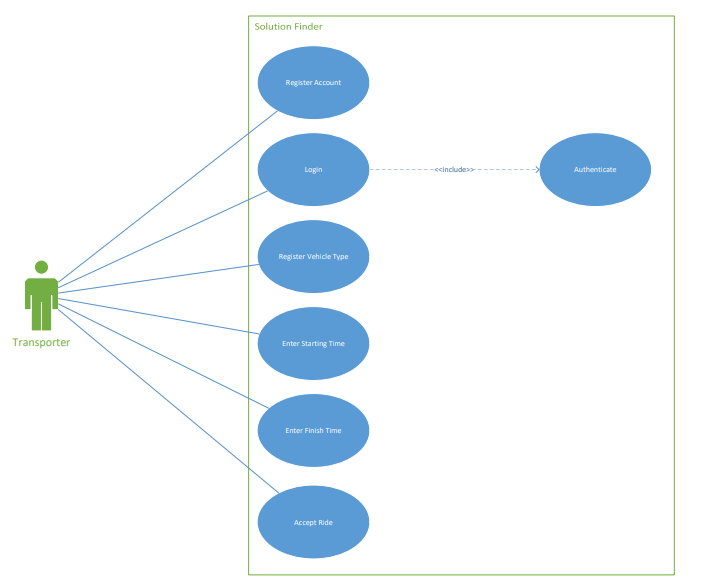
**Actor Name: Student**

****

****

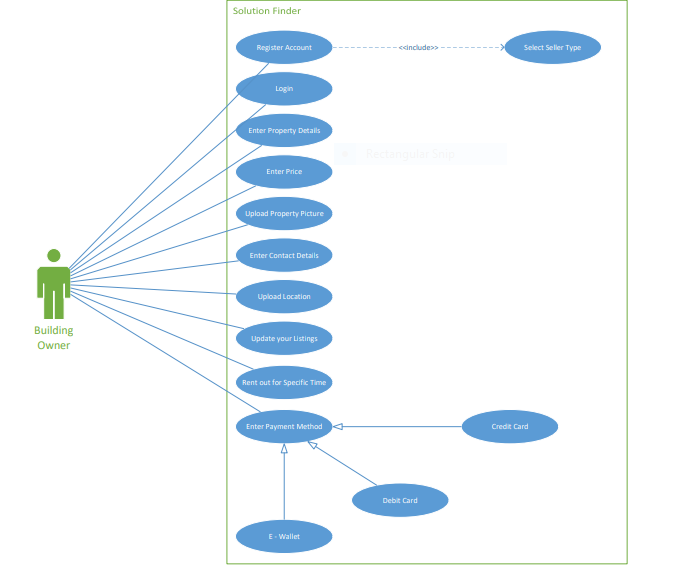
****

## Actor Name: Transporter



## Actor Name: Mart owner

## Actor Name: Building owner



## Use case description

The table below indicate a comprehensive use case template filled in with an example drawn from the Cafeteria ordering system (COS). (Appendix C) shows more sample use cases written according to this template. As with all templates, you don’t complete this from top to bottom, and you don’t necessarily need all the template information for every use case. The template is simply a structure in which to store the information you encounter during a use case discussion in an organized and consistent fashion. The template reminds you of all the information you should contemplate regarding each use case. For more detail see Chapter 8, “Understanding user requirements”

### Transporter

### Register Account

|  |  |
| --- | --- |
| **Use Case ID** | **TUC-1** |
| Use Case Name | Register Account |
| Created By: | Waleed Umar |
| Actors | Primary: Transporters |
| Description | User will enter his email, password and select seller type to register on system. If the entered data is valid then the account will be created or else he would be asked to enter the credentials again |
| Trigger | Transporter wants to make an account and use the system |
| Pre - Conditions | 1. Transporter has an active internet connection. 2. Transporter has an active email address. 3. Transporter has all the documents like CNIC and vehicle documents available. |
| Post - Conditions | 1. Information is verified. 2. Receives confirmation for account registration. 3. Access the account. |
| Normal Flow | 1. Transporter enters the email address 2. System checks the email address 3. Transporter enters the password 4. Transporter enters name 5. Transporter enters username 6. Checks availability of username 7. Transporter uploads pictures of documents 8. Transporter authenticates it 9. Account Created |
| Alternate Flow | In step 2a of normal flow if email is invalid:   1. The database checks the email 2. System will prompt user to correct the error 3. Transporter corrects the email 4. Use case resumes to step 3   In step 2b of normal flow if the email is invalid:   1. The database checks the email 2. System will prompt user to correct the error 3. Transporter does not correct it and leave 4. The system stops and exits. |
| Exceptions | In step 1 of normal flow:   1. The email entered is invalid or already exists 2. System prompts the user to change the email 3. Transporter changes the email 4. Authenticated by the system 5. Moves to selecting password   In step 3 of normal flow:   1. The password entered is invalid and doesn’t match the pattern. 2. The system prompts the user to change password. 3. User changes password. 4. System authenticates it. |
| Includes | Authentication  Select Seller Type |
| Business Rule | The data of the user should be stored and not given to anyone else |
| Special Requirement | Fast database authentication |

### Login

|  |  |
| --- | --- |
| **Use Case ID** | **TUC-2** |
| Use Case Name | Login |
| Created By: | Waleed Umar |
| Actors | Primary: Transporter  Secondary: Database |
| Description | Transporter will enter his account by entering login details. The system check for the details and if they are correct the user is allowed to proceed else he is asked to either enter the details again or create an account |
| Trigger | Transporter wants to access his account and use it again |
| Pre - Conditions | 1. Transporter has an active internet connection. 2. Already registered an account |
| Post - Conditions | 1. Logs in successfully 2. Starts using the services |
| Normal Flow | 1. Transporter enters the email and password 2. System checks for details 3. Logged in successfully |
| Alternate Flow | In step 2a of normal flow if login details are incorrect:   1. Database prints message “Incorrect Username or Password” 2. Transporter corrects the details 3. System prompts message “Logged in Successfully” |
| Include | Authenticate |
| Exceptions | In step 2 of normal flow:   1. System checks for email 2. System checks for password 3. If email is correct but not password, system asks user to re-enter password 4. If password is correct and not email, system asks user to enter email 5. If both email and password are not present in database, system asks user re-enter both. |
| Includes | Authenticate |
| Business Rule | Only person having an account should be able to access the services |
| Special Requirement | Authentication should not take long time |

### Register Vehicle Type

|  |  |
| --- | --- |
| **Use Case ID** | **TUC-3** |
| Use Case Name | Register Vehicle Type |
| Created By: | Waleed Umar |
| Actors | Primary: Transporter  Secondary Database |
| Description | Transporter adds detail about the vehicle |
| Trigger | Transporter wants to register his vehicle in the system |
| Pre - Conditions | 1. Transporter has an active internet connection. 2. Transporter has a registered account 3. Transporter is logged in the account |
| Post - Conditions | Vehicle registers successfully |
| Normal Flow | 1. Transporter logs into the account 2. Transports selects the register vehicle option 3. Transporter uploads the vehicle details 4. Transporter submits the documents and details |
| Alternate Flow | In step 1 of normal flow if the user isn’t logged in:   1. System asks the user to login 2. Transporter logs in 3. System moves to step 2 |
| Exceptions | In step 3 of normal flow if transporter enters invalid details:   1. System checks for vehicle details 2. The entered details are not correct for specific category 3. System asks the transporter to enter details again 4. Transporter enter the details again 5. System verifies the information. |
| Includes | NULL |
| Business Rule | The vehicle registered must not be involved in any theft or criminal activity and it must have clearance certificated from the police. |

### Enter Route Covered

|  |  |
| --- | --- |
| **Use Case ID** | **TUC-4** |
| Use Case Name | Enter route covered |
| Created By | Waleed Umar |
| Actors | Primary: Transporter |
| Created By: | Waleed Umar |
| Description | User will enter his specified routes and does not get the orders for areas outside the specified areas |
| Trigger | Transporter wants to add the routes he wants to use his services on |
| Pre - Conditions | 1. Transporter has an active internet connection. 2. Transporter has a registered account 3. Transporter is logged in the account |
| Post - Conditions | Routes entered successfully |
| Normal Flow | 1. Transporter logs into the account 2. Transporter selects “Enter Routes” option 3. Transporter enters the starting point 4. Transporter enters the finish point 5. Transporter enters vehicle for the route 6. Transporter submits details |
| Alternate Flow | In step 5 of normal flow:   1. The vehicle entered is already in use at other point at the same time. 2. System will prompt the error 3. User enters the route of that vehicle 4. Transporter adds a new vehicle for route |
| Exceptions | In step 1 of normal flow:   1. System checks for email 2. System checks for password 3. If email is correct but not password, system asks user to re-enter password 4. If password is correct and not email, system asks user to enter email 5. If both email and password are not present in database, system asks user re-enter both. |
| Includes | Enter Starting Time  Enter Vehicle Stop Areas |
| Business Rule | The vehicle registered must not be involved in any theft or criminal activity and it must have clearance certificated from the police. |
| Special Requirement | System should add details quickly and check the routes to add 2 vehicles on different routes at the same time |

### Enter Starting Time

|  |  |
| --- | --- |
| **Use Case ID** | **TUC-5** |
| Use Case Name | Enter Starting Time |
| Created By | Waleed Umar |
| Actors | Primary: Transporters |
| Description | Transporter enters the starting time for their services |
| Trigger | Transporter wants to add the time at which their services can be accessed |
| Pre - Conditions | 1. Transporter has an active internet connection. 2. Transporter has registered with the software. 3. Transporter is logged into his account. |
| Post - Conditions | Time updated successfully |
| Normal Flow | 1. Transporter logs into his account 2. Transporter selects starting time 3. Transporter selects the day 4. Transporter selects the time 5. Transporter enters vehicle details 6. Transporter enters route 7. Transporter submits the details |
| Alternate Flow | In step 2 of normal flow if the user is not logged in:  System asks transporter to log in  In step 1 of normal flow while login:   1. The user doesn’t have an account 2. The user registers for the account 3. Account created successfully 4. System Proceeds |
| Exceptions | In step 1 of normal flow:   1. User enters login credentials 2. The password entered is incorrect 3. System prompts the message “Invalid Password” 4. Transporter enters correct password 5. Transporter enters the username 6. System checks the username 7. It is incorrect 8. System asks the user to correct it |
| Includes | Enter route covered |
| Business Rule | The vehicle registered must not be involved in any theft or criminal activity and it must have clearance certificated from the police. |
| Special Requirement | The data entered is safe and has no access to outsiders. |

### Accept Ride

|  |  |
| --- | --- |
| **Use Case ID** | **TUC-6** |
| Use Case Name | Accept Ride |
| Created By | Waleed Umar |
| Actors | Primary: Transporters |
| Description | The driver receives a ride and accepts it else decline the ride |
| Trigger | Transporter receives a ride and wants to accept it |
| Pre - Conditions | 1. Transporter has logged into account 2. Transporter has GPS turned on 3. Transporter has internet connection 4. Transporter has received a request for ride |
| Post - Conditions | 1. Transporter accepts the ride 2. Receives the location of the client |
| Normal Flow | 1. Transporter logs in his account 2. Transporter turns on the GPS 3. Transporter receives a request for the ride 4. Transporter accepts the offer 5. Transporter receives the location of the client   In case of the transport service:   1. Transporter logs in his account 2. Transporter turns on the GPS 3. Transporter receives a request for the ride 4. Transporter accepts the offer 5. User gets the location of the nearby stop of the vehicle. |
| Alternate Flow | In step 3 of transport service normal flow if there is no space in vehicle:   1. System asks the user to search again 2. User searches again 3. System asks the user to book the ride for another time 4. User books the ride for another time 5. User gets the location of the nearby vehicle 6. User quits the application |
| Exceptions | In step 1 of normal flow:   1. User enters login credentials 2. The password entered is incorrect 3. System prompts the message “Invalid Password” 4. Transporter enters correct password 5. Transporter enters the username 6. System checks the username 7. It is incorrect 8. System asks the user to correct it |
| Includes | Turn on GPS |
| Business Rule | The vehicle registered must not be involved in any theft or criminal activity and it must have clearance certificated from the police. |
| Special Requirement | The system should provide the location of user accurately. |

## Mart Owner

### Create an Account

|  |  |
| --- | --- |
| **Use Case ID** | **MUC-1** |
| Use Case Name | Create an Account |
| Created By | M. Saqib Khan |
| Actors | Primary: Mart Owner |
| Description | Mart Owner will enter his username, password, and other details to make account |
| Trigger | Mart Owner wants to make an account to access these services |
| Pre – Conditions | 1. Mart Owner has an active internet connection. 2. Mart Owner has an active email address. 3. Mart Owner has all the documents like CNIC and vehicle documents available. |
| Post - Conditions | 1. Logs in successfully 2. Starts using the services |
| Normal Flow | 1. Mart Owner enters the email address 2. System checks the email address 3. Mart Owner enters the password 4. Mart Owner enters name 5. Mart Owner enters username 6. Checks availability of username 7. Mart Owner uploads pictures of documents 8. Mart Owner authenticates it 9. Account Created |
| Alternate Flow | In step 2a of normal flow if email is invalid:   1. The System checks the email 2. System will prompt user to correct the error 3. Mart Owner corrects the email 4. Use case resumes to step 3   In step 2b of normal flow if the email is invalid:   1. The System checks the email 2. System will prompt user to correct the error 3. Mart Owner does not correct it and leave 4. The system stops and exits. |
| Exceptions | In step 1 of normal flow:   1. The email entered is invalid or already exists 2. System prompts the user to change the email 3. Mart Owner changes the email 4. Authenticated by the system 5. Moves to selecting password   In step 3 of normal flow:   1. The password entered is invalid and doesn’t match the pattern. 2. The system prompts the user to change password. 3. User changes password. 4. System authenticates it. |
| Includes | * Authentication * Select Seller Type |
| Business Rule | The data of the user should be stored and not given to anyone else |
| Special Requirement | Fast System authentication |

### Login

|  |  |
| --- | --- |
| **Use Case ID** | **MUC-2** |
| Use Case Name | Login |
| Created By | M. Saqib Khan |
| Actors | Primary: Mart Owner |
| Description | Mart Owner will enter his details to access account. System check if the credentials are correct. If credentials are correct the system proceeds further else user is asked to enter details again |
| Trigger | Mart Owner wants to enter his account and use it again |
| Pre – Conditions | 1. Mart Owner has an active internet connection. 2. Already registered an account |
| Post - Conditions | 1. Logs in successfully 2. Starts using the services |
| Normal Flow | 1. Mart Owner enters the email and password 2. System checks for details 3. Logged in successfully |
| Alternate Flow | In step 2a of normal flow if login details are incorrect:   1. System prints message “Incorrect Username or Password” 2. Mart Owner corrects the details 3. System prompts message “Logged in Successfully” |
| Exceptions | In step 2 of normal flow:   1. System checks for email 2. System checks for password 3. If email is correct but not password, system asks user to re-enter password 4. If password is correct and not email, system asks user to enter email 5. If both email and password are not present in database, system asks user re-enter both. |
| Includes | Authenticate |
| Business Rule | Only person having an account should be able to access the services |
| Special Requirement | Authentication should not take long time |

### Enter your Address

|  |  |
| --- | --- |
| **Use Case ID** | **MUC-3** |
| Use Case Name | Enter your Address |
| Created By | M. Saqib Khan |
| Actors | Primary: Mart Owner |
| Description | Mart Owner will enter the address of where the store is. |
| Trigger | Mart Owner wants to enter the location of store |
| Pre – Conditions | 1. Mart Owner has an active internet connection. 2. Mart Owner has a registered account. |
| Post - Conditions | Address entered successfully |
| Normal Flow | 1. Mart Owner logs in his account 2. Mart Owner selects the “enter address” option 3. Mart Owner enters the address and phone number 4. Mart Owner submits information |
| Alternate Flow | In step 3 of normal flow:   1. The entered address is invalid 2. Mart Owner corrects the address 3. Entered number is invalid 4. Mart Owner corrects number 5. System proceeds to do step 4 |
| Exceptions | In step 3 of normal flow:   1. Entered phone number is invalid 2. System checks prints error of invalid number 3. Mart Owner corrects it 4. System proceeds forward |
| Includes | NULL |
| Special Requirement | Database authentication should be fast |

### Upload Item list and Price

|  |  |
| --- | --- |
| **Use Case ID** | **MUC-4** |
| Use Case Name | Upload item list and price |
| Created By | M. Saqib Khan |
| Actors | Primary: Mart Owner |
| Description | Mart Owner will upload the items available with price |
| Trigger | Mart Owner wants to upload the items that are available at the mart |
| Pre – Conditions | 1. Mart Owner has an account 2. Mart Owner is logged in the system. |
| Post - Conditions | 1. Item List added successfully 2. Buyer can see the list |
| Normal Flow | 1. Mart Owner logs in his account 2. Mart Owner selects ‘upload items’ option 3. Mart Owner enters the item name 4. Mart Owner clicks the add item option 5. Item added successfully |
| Alternate Flow | In step 1 of normal flow if the credentials added are incorrect:   1. The system checks the credentials 2. Database prints a message “Incorrect Credentials” 3. Mart Owner corrects them 4. Authenticated successfully 5. System moves forward to step 2. |
| Exceptions | In step 4 of normal flow if the item is already added:   1. The item isn’t added twice 2. System prompts the message “Item already exists” 3. Mart Owner enters another item |
| Includes | NULL |
| Business Rule | The items added shouldn’t be expired |
| Special Requirement | The system should not add the same item twice |

### Enter Areas Covered

|  |  |
| --- | --- |
| **Use Case ID** | **MUC-5** |
| Use Case Name | Enter areas covered |
| Created By | M. Saqib Khan |
| Actors | Primary: Mart Owner |
| Description | Mart Owner will add the areas where he can provide services |
| Trigger | Mart Owner wants to add the areas he covers for delivery |
| Pre – Conditions | 1. Mart Owner has an active internet connection. 2. Mart Owner’s GPS is turned on. 3. Mart Owner’s account is registered. 4. Mart / Shop is already added to the system |
| Post - Conditions | Locations added successfully |
| Normal Flow | 1. Mart Owner logs in the account 2. Mart Owner turns on the GPS 3. Mart Owner selects the mart registered 4. Mart Owner selects the ‘areas covered’ option 5. Mart Owner adjusts the location to provide services with respect to the current location |
| Alternate Flow | In step 5 of normal flow while adjusting the location parameter if GPS isn’t turned on:   1. System asks the user to turn on GPS 2. Mart Owner turns on the GPS 3. System shows your current location 4. Mart Owner adjusts the parameter to offer services |
| Includes | Turn on GPS |
| Business Rule | System shouldn’t accept offers from buyers that outside the delivery range |
| Special Requirement | The areas shown by the GPS should be accurate to the mart owner |

### Enter Payment Options

|  |  |
| --- | --- |
| **Use Case ID** | **MUC-6** |
| Use Case Name | Enter Payment Options |
| Created By | M. Saqib Khan |
| Actors | Primary: Mart Owner |
| Description | Mart Owner will provide the payment options through which he wants to receive payment. |
| Trigger | Mart Owner wants to add payment method to receive payment from users |
| Pre – Conditions | 1. Mart Owner has a registered account. 2. Mart Owner has a valid account on the services to receive payment from users. |
| Post – Conditions | 1. Mart Owner receives payment 2. Mart Owner can see the record of transactions |
| Normal Flow | 1. Mart Owner logs into his account 2. Mart Owner selects the “Enter Payment” option 3. Mart Owner selects the payment option 4. Mart Owner enters the details 5. System authenticates it 6. Method added successfully |
| Alternate Flow | In step 5 of normal flow during authentication:   1. The entered information is incorrect 2. System will prompt user to correct the error 3. Mart Owner corrects the details 4. System authenticates 5. System moves to step 6 |
| Exceptions | In step 4 of normal flow if mart owner enters invalid credentials:   1. System prints message “Invalid Details” 2. Mart Owner enters the details again 3. System authenticates 4. System verifies the data and ask the mart owner to confirm. 5. Mart owners confirms adding the data 6. System moves to step 6 |
| Includes | Payment Authentication |
| Special Requirement | The system should be able to keep the data of the user safe |

## Building Owner

### Register Account

|  |  |
| --- | --- |
| **Use Case ID** | **BUC-1** |
| Use Case Name | Register Account |
| Created By | Ahmad bin Ashraf |
| Actors | Primary: Building Owner |
| Description | Building Owner will enter his email, password to make an account |
| Trigger | Building Owner wants to make an account and use the system |
| Pre - Conditions | 1. Building Owner has an active internet connection. 2. Building Owner has an active email address. 3. Building Owner has all the documents like CNIC and vehicle documents available. |
| Post - Conditions | 1. Information is verified. 2. Receives confirmation for account registration. 3. Access the account. |
| Normal Flow | 1. Building Owner enters the email address 2. System checks the email address 3. Building Owner enters the password 4. Building Owner enters name 5. Building Owner enters username 6. Checks availability of username 7. Building Owner uploads pictures of documents 8. Building Owner authenticates it 9. Account Created |
| Alternate Flow | In step 2a of normal flow if email is invalid:   1. The system checks the email 2. System will prompt user to correct the error 3. Building Owner corrects the email 4. Use case resumes to step 3   In step 2b of normal flow if the email is invalid:   1. The system checks the email 2. System will prompt user to correct the error 3. Building Owner does not correct it and leave 4. The system stops and exits. |
| Exceptions | In step 1 of normal flow:   1. The email entered is invalid or already exists 2. System prompts the user to change the email 3. Building Owner changes the email 4. Authenticated by the system 5. Moves to selecting password   In step 3 of normal flow:   1. The password entered is invalid and doesn’t match the pattern. 2. The system prompts the user to change password. 3. User changes password. 4. System authenticates it. |
| Includes | Authentication  Select Seller Type |
| Business Rule | The data of the user should be stored and not given to anyone else |
| Special Requirement | Fast database authentication |

### Login

|  |  |
| --- | --- |
| **Use Case ID** | **BUC-2** |
| Use Case Name | Login |
| Created by | Ahmad bin Ashraf |
| Actors | Primary: Building Owner |
| Description | Building Owner will enter his details to access account |
| Trigger | Building Owner wants to enter his account and use it again |
| Pre – Conditions | 1. Building Owner has an active internet connection. 2. Already registered an account |
| Post - Conditions | 1. Logs in successfully 2. Starts using the services |
| Normal Flow | 1. Building Owner enters the email and password 2. Database checks for details 3. Logged in successfully |
| Alternate Flow | In step 2a of normal flow if login details are incorrect:   1. System prints message “Incorrect Username or Password” 2. Building Owner corrects the details 3. System prompts message “Logged in Successfully” |
| Exceptions | In step 2 of normal flow:   1. System checks for email 2. System checks for password 3. If email is correct but not password, system asks user to re-enter password 4. If password is correct and not email, system asks user to enter email 5. If both email and password are not present in database, system asks user re-enter both. |
| Includes | Authenticate |
| Business Rule | Only person having an account should be able to access the services |
| Special Requirement | Authentication should not take long time |

### Enter Property Details

|  |  |
| --- | --- |
| **Use Case ID** | **BUC-3** |
| Use Case Name | Enter Property Details |
| Created By | Ahmad bin Ashraf |
| Actors | Primary: Building Owner |
| Description | Building Owner will enter the details of the building like number of rooms, location, etc. |
| Trigger | Building Owner wants to add property details on the system |
| Pre - Conditions | 1. Building Owner has an active internet connection. 2. Building Owner has an account |
| Post - Conditions | 1. Details entered successfully 2. Buyer can see the details |
| Normal Flow | 1. Building Owner logs into the account 2. Building Owner clicks the “Add Property” option 3. Building Owner enters details 4. Building Owner submits the details |
| Alternate Flow | In step 4 of normal flow if the submitted details are already in the system:   1. System prompts the message “Property already exists” 2. Building Owner checks the details 3. Building Owner corrects the details 4. System authenticates the details 5. Authenticated Successfully |
| Exceptions | In step 2 of normal flow if the system crashes:   1. Building owner is redirected to the home page 2. Building Owner clicks on the button again 3. Building Owner enter the property details 4. System move to step 3 |
| Includes | NULL |
| Business Rule | The building added must be cleared from law agencies and must not be involved in any illegal activities |

### Enter Price

|  |  |
| --- | --- |
| **Use Case ID** | **BUC-4** |
| Use Case Name | Enter Price |
| Created by | Ahmad bin Ashraf |
| Actors | Primary: Building Owner |
| Description | Building Owner logs into his account and enter the price of desired property. System updates the listing after the price is added |
| Trigger | Building Owner wants to set the price of property |
| Pre - Conditions | 1. Building Owner has an account. 2. The property details are already added |
| Post - Conditions | Price updated successfully |
| Normal Flow | 1. Building Owner logs into the account 2. Building Owner selects the property to add price to 3. Building Owner enters price 4. Building Owner submits the price |
| Alternate Flow | In step 1 of normal flow if the user does not have an account:   1. System checks the credentials and prompts the message “User not Found” 2. System asks user to create an account 3. Building Owner registers for account 4. System proceeds to step 2 |
| Exceptions | In step 1 of normal flow if the login credentials are incorrect:   1. System prompts the user to enter details again 2. Building Owner enters the details again 3. System checks the details 4. Building Owner logs in successfully |
| Includes | NULL |
| Special Requirement | The system should update details quickly |

### Enter Contact Details

|  |  |
| --- | --- |
| **Use Case ID** | **BUC-5** |
| Use Case Name | Enter Contact Details |
| Created By | Ahmad bin Ashraf |
| Actors | Primary: Building Owner |
| Description | Building Owner will enter contact details |
| Trigger | Building Owner wants to enter contact details, so that buyer can contact easily |
| Pre - Conditions | 1. Building Owner has an account. 2. Building Owner is already logged in |
| Post - Conditions | 1. Phone number added successfully 2. Contact details added successfully 3. Buyer can contact using given details |
| Normal Flow | 1. Building Owner logs into the account 2. Building Owner selects “Enter Contact Details 3. Building Owner enters the email address 4. Building Owner enters mobile number |
| Alternate Flow | In step 3 of normal flow if the email entered is invalid:   1. It doesn’t match the pattern of email 2. System prints error message 3. Building Owner corrects it |
| Includes | NULL |
| Business Rule | The phone number added should be registered and traceable |

### Update your Listings

|  |  |
| --- | --- |
| **Use Case ID** | **BUC-6** |
| Use Case Name | Update your Listings |
| Created By | Ahmad bin Ashraf |
| Actors | Primary: Building Owner |
| Description | Building Owner will change the status of listing after logging to his account and selecting the intended property. To edit the listing, the property must first be added in the system |
| Trigger | Building Owner wants to enter some properties for rent and want to update their details |
| Pre - Conditions | 1. Building Owner has an account. 2. Building Owner is already logged in 3. Building Owner has added some properties in the past |
| Post - Conditions | Details updated successfully |
| Normal Flow | 1. Building Owner logs into the account 2. Building Owner selects the property 3. Building Owner updates the details 4. Building Owner submits the new details |
| Alternate Flow | In step 2 of normal flow |
| Exceptions | In step 3 of normal flow:   1. The system does not updates the listing 2. Building owner updates the details and submit again |
| Includes | NULL |
| Business Rule | The property shouldn’t be involved in legal cases |
| Special Requirement | The system should change the list quickly |

### Enter Payment Options

|  |  |
| --- | --- |
| **Use Case ID** | **BUC-7** |
| Use Case Name | Enter Payment Options |
| Created by | Ahmad bin Ashraf |
| Actors | Primary: Building Owner |
| Description | Building Owner will add payment methods to make and receive payment |
| Trigger | Building Owner wants to add payment method |
| Pre – Conditions | 1. Building Owner has a registered account. 2. Building Owner has a valid account on the services to receive payment from users. |
| Post – Conditions | 1. Building Owner receives payment 2. Building Owner can see the record of transactions |
| Normal Flow | 1. Building Owner logs into his account 2. Building Owner selects the “Enter Payment” option 3. Building Owner selects the payment option 4. Building Owner enters the details 5. System authenticates it 6. Method added successfully |
| Alternate Flow | In step 5 of normal flow during authentication:   1. The entered information is incorrect 2. System will prompt user to correct the error 3. Building Owner corrects the details 4. System authenticates 5. System moves to step 6 |
| Exceptions | In step 4 of normal flow if Building owner enters invalid credentials:   1. System prints message “Invalid Details” 2. Building Owner checks the credentials and correct them 3. System authenticates 4. Payment option added successfully |
| Includes | Payment Authentication |
| Special Requirement | The system should be able to keep the data of the user safe |

## Student

### Create an Account

|  |  |
| --- | --- |
| **Use Case ID** | **SUC-1** |
| Use Case Name | Create an Account |
| Created by | Waleed Umar |
| Actors | **Primary**: Student  **Secondary**: Database |
| Description | Student will enter details and create account |
| Trigger | Student wants to use system to have access to services and wants to create an account |
| Pre - Conditions | 1. Student has an active internet connection. 2. Student has an active email address. 3. Student has all the documents like CNIC and vehicle documents available. |
| Post - Conditions | 1. Information is verified. 2. Receives confirmation for account registration. 3. Access the account. |
| Normal Flow | 1. Student enters the email address 2. Database checks the email address 3. Student enters the password 4. Student enters name 5. Student enters username 6. Checks availability of username 7. Student uploads pictures of documents 8. Student authenticates it 9. Account Created |
| Alternate Flow | In step 2a of normal flow if email is invalid:   1. The database checks the email 2. System will prompt user to correct the error 3. Student corrects the email 4. Use case resumes to step 3   In step 2b of normal flow if the email is invalid:   1. The database checks the email 2. System will prompt user to correct the error 3. Student does not correct it and leave 4. The system stops and exits. |
| Exceptions | In step 1 of normal flow:   1. The email entered is invalid or already exists 2. System prompts the user to change the email 3. Student changes the email 4. Authenticated by the system 5. Moves to selecting password   In step 3 of normal flow:   1. The password entered is invalid and doesn’t match the pattern. 2. The system prompts the user to change password. 3. User changes password. 4. System authenticates it. |
| Includes | Authentication  Select Seller Type |
| Business Rule | The data of the user should be stored and not given to anyone else |
| Special Requirement | Fast database authentication |

### Login

|  |  |
| --- | --- |
| **Use Case ID** | **SUC-2** |
| Use Case Name | Login |
| Created By | Waleed Umar |
| Actors | **Primary**: Student  **Secondary**: Database |
| Description | Student is logged out and wants to access the system again |
| Trigger | Student wants to sign in again to use their account |
| Pre – Conditions | 1. Student has an active internet connection. 2. Already registered an account |
| Post - Conditions | 1. Logs in successfully 2. Starts using the services |
| Normal Flow | 1. Student enters the email and password 2. Database checks for details 3. Logged in successfully |
| Alternate Flow | In step 2a of normal flow if login details are incorrect:   1. Database prints message “Incorrect Username or Password” 2. Student corrects the details 3. System prompts message “Logged in Successfully” |
| Exceptions | In step 2 of normal flow:   1. Database checks for email 2. Database checks for password 3. If email is correct but not password, system asks user to re-enter password 4. If password is correct and not email, system asks user to enter email 5. If both email and password are not present in database, system asks user re-enter both. |
| Includes | Authenticate |
| Business Rule | Only person having an account should be able to access the services |
| Special Requirement | Authentication should not take long time |

### Select Service Type

|  |  |
| --- | --- |
| **Use Case ID** | **SUC-3** |
| Use Case Name | Select Service Type |
| Created By | Waleed Umar |
| Actors | **Primary**: Student |
| Description | Student will select the service type he wants to use |
| Trigger | Student wants to select a particular service type to order services |
| Pre – Conditions | 1. Student has an active internet connection. 2. Already registered an account 3. Student is already logged in |
| Post – Conditions | 1. Service type selected successfully 2. Student can use the selected services type |
| Normal Flow | 1. Student logs into the account 2. Student selects the service type |
| Alternate Flow | In step 1 of normal flow:   1. Student attempts to log into the account 2. Student enters the wrong credentials 3. System prints message “Incorrect Password or Username” 4. Student rechecks the details and corrects them 5. System authenticates the information and prints message “Login Successful” |
| Exceptions | In step 1 of normal flow:   1. Username entered is incorrect 2. System prints message “Incorrect Username” 3. Student corrects it 4. Student forgot password 5. Students asks to change password 6. Student enters new password 7. System authenticates it 8. Logged in successfully |
| Includes | NULL |
| Business Rule | The service provided should comply with local laws |
| Special Requirement | System should be fast and reliable  Students should be allowed to select only one service type at a time |

### Search nearby Property

|  |  |
| --- | --- |
| **Use Case ID** | **SUC-4** |
| Use Case Name | Search Nearby Property |
| Created By | Waleed Umar |
| Actors | **Primary**: Student  **Secondary**: GPS |
| Description | Student will have access to nearby property according to their choice |
| Trigger | Student wants to see the available property in his nearby area |
| Pre – Conditions | 1. Student should be signed into his account 2. GPS must be turned on |
| Post – Conditions | Student will select the desired property available |
| Normal Flow | 1. Student turns on GPS 2. Student logs into his account 3. Student enters the type of property 4. Student starts searching nearby property |
| Alternate Flow | In step 1 of normal flow, if the GPS is turned off:  System prints message “Turn on GPS” |
| Exceptions | In step 3 of normal flow if there is no available property:   1. System prompts the student to try again or try later 2. Student select for try again 3. System move to home page |
| Includes | GPS turned on |
| Special Requirement | The system shows the location as accurately as possible |

### Book Property

|  |  |
| --- | --- |
| **Use Case ID** | **SUC-5** |
| Use Case Name | Book Property |
| Created By | Waleed Umar |
| Actors | **Primary:** Student |
| Description | Student will select a property and book it |
| Trigger | Student wants to book property to use it. If the selected property is already booked for particular dates, student can save for later or look for another building |
| Pre – Conditions | 1. Student should be signed into his account 2. Student has an active internet connection 3. GPS must be turned on |
| Post – Conditions | Property booked successfully |
| Normal Flow | 1. Student turns on GPS 2. Student logs into his account 3. Student searches the property 4. Student selects a particular property 5. Student books the property |
| Alternate Flow | In step 5 of normal flow, if the property is already booked or currently not available for a specific period:  System prints message “Not Available. Check another property” |
| Exceptions | In step 1 of normal flow if GPS is not turn   1. System asks the user to turn it on 2. Student turns it on 3. System move to step2 |
| Includes | NULL |
| Business Rule | Students must read and accept terms and conditions in order to get control of the property |
| Special Requirement | The system updates the listings quickly |

### Search Restaurant / Mart

|  |  |
| --- | --- |
| **Use Case ID** | **SUC-6** |
| Use Case Name | Search Restaurant / Mart |
| Created By | Waleed Umar |
| Actors | **Primary:** Student  **Secondary**: GPS |
| Description | Student will search the nearby restaurant / mart |
| Trigger | Student wants to buy grocery or order food and search the nearby available marts or restaurants |
| Pre – Conditions | 1. Student should be signed into his account 2. Student has an active internet connection 3. GPS must be turned on |
| Post – Conditions | Student will select the desired outlet to order grocery / food |
| Normal Flow | 1. Student logs into his account 2. Student selects the ‘Order Food / Grocery’ service 3. Student turns on GPS 4. Student selects the available outlets 5. Student orders food / services |
| Alternate Flow | In step 3 of normal flow, if the GPS is turned off:  System prints message “Turn on GPS” |
| Exceptions | In step 4 of normal flow if available outlet is out of service:   1. System prints the message out of service 2. Student search for another outlet 3. System move to step4 |
| Includes | GPS turned on |
| Special Requirement | The system shows the location as accurately as possible |

### Place Order

|  |  |
| --- | --- |
| **Use Case ID** | **SUC-7** |
| Use Case Name | Place Order |
| Created By | Waleed Umar |
| Actors | **Primary**: Student |
| Description | Student will place order to get grocery. If the store does not deliver in the area system prints the message “Not available for delivery” & he shops from another one |
| Trigger | Student has selected the items and wants to buy them |
| Pre – Conditions | 1. Student has an account 2. Student is logged in the account 3. Student has selected the items to buy |
| Post – Conditions | Order created successfully and receives the receipt of order |
| Normal Flow | 1. Student logs into his account 2. Student turns on GPS 3. Student searches the marts 4. Student selects the mart 5. Student selects the items to buy 6. Student places order |
| Alternate Flow | In step 2 of normal flow, if the GPS is turned off:  System prints message “Turn on GPS” |
| Exceptions | In step 5 of normal flow if the rider isn’t available:   1. Mart Owner informs student that the rider isn’t available 2. Student selects delivery time for later |
| Includes | NULL |
| Special Requirement | The system should not take long time to process the order |

### Make Payment

|  |  |
| --- | --- |
| **Use Case ID** | **SUC-8** |
| Use Case Name | Make Payment |
| Created By | Waleed Umar |
| Actors | **Primary:** Student  **Secondary**: Payment Service Provider |
| Description | Student will choose the given payment methods to make payment. If the |
| Trigger | Student wants to make payment to complete the order |
| Pre – Conditions | 1. Student has a registered account. 2. Student has a valid account on the services to make payment |
| Post – Conditions | 1. Student makes payment 2. Student receives a receipt of transaction |
| Normal Flow | 1. Student logs into his account 2. Student makes his order 3. Student selects the “Make Payment” option 4. Student selects the payment option 5. Student enters the details 6. System authenticates it 7. Payment completed successfully |
| Alternate Flow | In step 6 of normal flow during authentication:   1. The entered information is incorrect 2. System will prompt user to correct the error 3. Student corrects the details 4. System authenticates 5. System moves to step 7 |
| Exceptions | In step 1 of normal flow if Student enters invalid credentials:   1. System prints message “Invalid Details” 2. Student checks the credentials and correct them 3. System authenticates 4. Logged in successfully |
| Includes | Payment Authentication |
| Special Requirement | The system should be able to keep the data of the user safe |

# Specific requirements

This section describes the functional requirements of the system expressed in natural language style. This section is typically organized by feature as system feature name and specific functional requirements associated with this feature. It is just one possible way to arrange them. Other organizational options include arranging functional requirements by use case, process flow, mode of operation, user class, stimulus, and response depend what kind of technique which has been used to understand functional requirements. Hierarchical combinations of these elements are also possible, such as use cases within user classes. For further detail see Chapter 10 “Documenting the requirements”. Let consider feature scheme as an example.

## System feature X

* **Login**

1. 2 Login.Verification
2. 2.1 Login.Unregistered

* **Register Account**

1. 1 RegisterAccount.UniqueUsername
2. 1.1 RegisterAccount.choosenew

* **Accept ride**

1. 3 AcceptRide.oneattime

* **Enter areas covered**

1. 4 AreaCovered.WithinDistance

* **Update your listing**

1. 5 Updatelisting.2stepVerification
2. 5.1 Updatelisting.2stepVerification.ReceiveOTP
3. 5.2 Updatelisting.2stepVerification.Generate

* **Place Order**

1. 6 PlaceOrder.Login
2. 6.1 PlaceOrder.Registered
3. 6.2 PlaceOrder.Register.Notify

* **Make payment**

1. 7 MakePayment.Notification
2. 7.1 MakePayment.UnsufficientBalance

* **Update Item list**

1. 8 UpdateItemList.Notify

### Functional requirements

# Use case Name: Register Account

### <RegisterAccount.UniqueUsername >

|  |  |
| --- | --- |
| **Identifier** | Req\_1 |
| **Title** | RegisterAccount.UniqueUsername |
| **Requirement** | The system shall not assign same username to two different accounts |
| **Source** | The source of this requirement comes from studying similar systems |
| **Rationale** | The reason is to provide each user a unique id so that it is easy to fetch data for a particular user and not to cause a confusion for system while working on the database |
| **Business Rule (if required)** | No two users can be given same username |
| **Dependencies** | Req\_1.1, Req\_6.1 |
| **Priority** | High |

### <RegisterAccount.choosenew>

|  |  |
| --- | --- |
| **Identifier** | **Req\_1.1** |
| **Title** | RegisterAccount.choosenew |
| **Requirement** | If the username is already taken, the system shall prompt user to select new username |
| **Source** | Where this requirement is come from (who originate it) |
| **Rationale** | The reason is to provide each user a unique id so that it is easy to fetch data for a particular user and not to cause a confusion for system while working on the database |
| **Business Rule (if required)** | No two users can be given same username |
| **Dependencies** | NULL |
| **Priority** | High |

# Use case Name: Login

### <Login. Verification >

|  |  |
| --- | --- |
| **Identifier** | Req\_2 |
| **Title** | Login.Verification |
| **Requirement** | The system shall first check if the entered details have an account attached to them first.  The system shall prompt a verification for account user if there are 3 consecutive failed login attempts due to invalid password entered by sending **OTP** to the number or email at which the account is registered |
| **Source** | Client |
| **Rationale** | To protect the account of the real user by preventing it from unauthorized access and increase its security |
| **Business Rule (if required)** | Prevent anyone else from using the services apart from the real user |
| **Dependencies** | NULL |
| **Priority** | High |

### <Login.Unregistered >

|  |  |
| --- | --- |
| **Identifier** | Req\_2.1 |
| **Title** | Login. Unregistered |
| **Requirement** | The system shall first check if the entered details have an account attached to them first.  If the details do not have an account, system shall prompt a message asking the user to create an account |
| **Source** | Client |
| **Rationale** | To make it easy for user to create account. They would not have to get back to go to account creation instead make with single click |
| **Business Rule (if required)** | Prevent anyone else from using the services apart from the real user |
| **Dependencies** | NULL |
| **Priority** | Medium |

# Use case Name: Accept Ride

### <AcceptRide.oneattime >

|  |  |
| --- | --- |
| **Identifier** | Req\_3 |
| **Title** | AcceptRide.oneatatime |
| **Requirement** | The system shall ensure that a particular transporter registered for an individual car can accept only one ride at a particular time.  The system shall not provide an option of ride sharing with another order.  In order to accept another ride, the system shall ask the transporter to cancel the current ride. |
| **Source** | Client |
| **Rationale** | The reason is to provide the user (students) to have a comfortable ride experience especially for female |
| **Business Rule (if required)** | The system should not provide an option for ride cancellation to driver after 5 minutes of accepting it. |
| **Dependencies** | NULL |
| **Priority** | High |

# Use case Name: Enter Areas Covered

### <AreaCovered.WithinDistance >

|  |  |
| --- | --- |
| **Identifier** | Req\_4 |
| **Title** | AreaCovered.WithinDistance |
| **Requirement** | The system checks if the area to deliver order falls in the jurisdiction.  If it does not fall in the jurisdiction then the system shall not accept order and prompt the message to buyer that area does not fall in the covered zone. |
| **Source** | Client (Mart Owner) |
| **Rationale** | To provide a timely prompt message to the user so he can manage and avail other services and protect the stakeholder in question from getting bad rating |
| **Business Rule (if required)** | Do not accept order from the user that do not fall in the covered areas for a particular seller |
| **Dependencies** | NULL |
| **Priority** | Medium |

# Use case Name: Update Your Listing

### <Updatelisting.2stepVerification >

|  |  |
| --- | --- |
| **Identifier** | Req\_5 |
| **Title** | UpdateListing.2StepVerification |
| **Requirement** | The system shall let user change the property listing if he is logged into the Solution Finder system  When the user cause change in the property details, the system shall send and OTP registered email or mobile number. |
| **Source** | Client |
| **Rationale** | This functionality is added to make sure that the user really wants to change the listings and that no one else was trying to cause a change to his data. It provides high sense of security |
| **Business Rule (if required)** | The OTP is send to the registered email or mobile number |
| **Dependencies** | Req\_5.2 |
| **Priority** | High |

### <Updatelisting.2stepVerification.ReceiveOTP >

|  |  |
| --- | --- |
| **Identifier** | **Req\_5.1** |
| **Title** | UpdateListing.2StepVerification.ReceiveOTP |
| **Requirement** | The system shall let user change the property listing if he is logged into the Solution Finder system  The system shall generate and send OTP within 60 seconds from the time the user request for it |
| **Source** | The source of this requirement comes from studying similar systems |
| **Rationale** | This functionality is added to make sure that the user really wants to change the listings and that no one else was trying to cause a change to his data. It provides high sense of security |
| **Business Rule (if required)** | The OTP is send to the registered email or mobile number |
| **Dependencies** | Req\_5 |
| **Priority** | High |

### <Updatelisting.2stepVerification.Generate >

|  |  |
| --- | --- |
| **Identifier** | **Req\_5.2** |
| **Title** | UpdateListing.2StepVerification.Generate |
| **Requirement** | If the user does not receive an OTP within 60 seconds, the system shall give option to user to request for new OTP or try again later |
| **Source** | The source of this requirement comes from studying similar systems |
| **Rationale** | This functionality is added to make sure that the user really wants to change the listings and that no one else was trying to cause a change to his data. It provides high sense of security |
| **Business Rule (if required)** | The OTP is send to the registered email or mobile number |
| **Dependencies** | NULL |
| **Priority** | High |

# Use case Name: Place Order

### <PlaceOrder.Login >

|  |  |
| --- | --- |
| **Identifier** | **Req\_6** |
| **Title** | PlaceOrder.Login |
| **Requirement** | The system shall let user who is logged into the Solution Finder to place order. |
| **Source** | User requirement |
| **Rationale** | The reason is to make sure that more and more people utilize the software to place order and order services. |
| **Business Rule (if required)** | The user who is not logged in cannot place order |
| **Dependencies** | NULL |
| **Priority** | Medium |

### <PlaceOrder.Registered >

|  |  |
| --- | --- |
| **Identifier** | **Req\_6.1** |
| **Title** | PlaceOrder.Registered |
| **Requirement** | The system shall confirm that user is registered in order to place order |
| **Source** | User requirement |
| **Rationale** | The reason is to make sure that more and more people utilize the software to place order and order services. |
| **Business Rule (if required)** | The user who is not register cannot place order |
| **Dependencies** | Req\_6 |
| **Priority** | High |

### <PlaceOrder.Register.Notify >

|  |  |
| --- | --- |
| **Identifier** | **Req\_6.2** |
| **Title** | PlaceOrder.Register.Notify |
| **Requirement** | If the user is not registered in Solution Finder the system shall give option to user to make an account. |
| **Source** | Where this requirement is come from (who originate it) |
| **Rationale** | The reason is to provide each user a unique id so that it is easy to fetch data for a particular user and not to cause a confusion for system while working on the database |
| **Business Rule (if required)** | No two users can be given same username |
| **Dependencies** | Req\_6.2 |
| **Priority** | Medium |

# Use case Name: Make Payment

### < MakePayment.Notification>

|  |  |
| --- | --- |
| **Identifier** | **Req\_7** |
| **Title** | MakePayment.Notification |
| **Requirement** | The system shall notify user after a payment Is made through registered email or mobile number |
| **Source** | This requirement comes from the **user** who wants to be notified whenever a transaction is made from their account |
| **Rationale** | The reason is ensure the safety of the account for the user to increase transparency of the transactions made |
| **Business Rule (if required)** | User is notified through registered email or mobile number within 5 minutes of transaction |
| **Dependencies** | NULL |
| **Priority** | High |

### < MakePayment.UnsufficientBalance>

|  |  |
| --- | --- |
| **Identifier** | **Req\_7.1** |
| **Title** | MakePayment.UnsufficientBalance |
| **Requirement** | System shall notify buyer if his account balance is not sufficient. |
| **Source** | Buyer |
| **Rationale** | To make sure that buyer have enough balance in his account to make payment. |
| **Business Rule (if required)** | The bank account of user must be verified in order to make payment |
| **Dependencies** | Req\_7 |
| **Priority** | Medium |

# Use case Name: Update item list

**<UpdateItemList.Notify>**

|  |  |
| --- | --- |
| **Identifier** | **Req\_8** |
| **Title** | UpdateItemList.Notify |
| **Requirement** | System shall notify the mart owner if the expiry date of the product is after 5 days or less |
| **Source** | Mart owner |
| **Rationale** | To make sure that the things people are buying are not expired and are according to health standards |
| **Business Rule (if required)** | No product on the system shall be expired |
| **Dependencies** | NULL |
| **Priority** | Medium |

# Quality attributes

This section specifies nonfunctional requirements other than constraints, which are recorded in section 2.3, and external interface requirements, which will appear in section 7. These quality requirements should be specific, quantitative, and verifiable. Chapter 14 “beyond functionality” presents more information about these quality attribute requirements and many examples. Following are some example for documenting guideline.

## Usability

|  |  |
| --- | --- |
| USE-1 | The user using system for the first time shall be able to learn how to use it by using it for the third 95% of time |
| USE-2 | User using the system for first shall be given option to see tutorial on how to use |
| USE-3 | A trained user shall be able to order their services within 3 minutes and max of 5 minutes 90% of the time |

## Performance

|  |  |
| --- | --- |
| PE-1 | The system shall authenticate credentials and login within 5 seconds 95% of the time |
| PE-2 | The system shall process payment within 5 seconds 90% of the time |
| PE-3 | The system shall be able to process 5000 credit cards within 5 seconds 90% of the time |

## Security

|  |  |
| --- | --- |
| SEC-1 | After three consecutive failed login attempts, the system shall send an OTP on registered email within 2 minutes |
| SEC-2 | User shall not be allowed to order any service without logging in or making an account |
| SEC-3 | A particular user shall not be given access to other user’s data. For instance mart owner shall not be given access to student’s data |

## Availability

|  |  |
| --- | --- |
| AVL-1 | The system shall be available for ordering ride between 8-11 A.M and 2-5 P.M 95% of time according to Pakistan Standard Time |

## Installability

|  |  |
| --- | --- |
| INS-1 | An untrained user shall be able to install and setup the system within 10 minutes 95% of the time |

## Robustness

|  |  |
| --- | --- |
| ROB-1 | If the internet connection error occurs while the user is using it, the system shall recover from the last step before it went out |

# External interface requirements

This section provides information to ensure that the system will communicate properly with users and with external hardware or software elements. A complex system with multiple subcomponents should create a separate interface specification or system architecture specification. The interface documentation could incorporate material from other documents by reference. For instance, it could point to a hardware device manual that lists the error codes that the device could send to the software.

## User interfaces

|  |  |
| --- | --- |
| UI-1 | The system shall provide help link on every page to guide how to use the functions on that page |
| UI-2 | The password selection part on signup page shall have guidance regarding the pattern of password |

## Software interfaces

|  |  |
| --- | --- |
| SI-1 | The mart inventory shall inform the system if a product is not available so that it is displayed to the user |

## Hardware interfaces

**No** hardware interfaces for **Solution Finder**

## Communications interfaces

|  |  |
| --- | --- |
| CI-1 | The system shall send an email to user after order has been placed |
| CI-2 | If there is some issue with the order, the system shall inform the user through email or mobile number as specified the user |