

**CSE-5324**  
**\*\* Software Engineering \*\***

**GROUP PROJECT: “Mav\$hop”**  
**GROUP NAME:- “Smartinis” (Team-2)**



**ITERATION-3**

**Project By:**  
**Neha Anil Shimpi (Captain)**  
**Sucharitha Cheenepalli**  
**Rupal Ravindra Godbole**  
**Akshat Ashwin Shah**  
**Sai Sravanti Podilli**  
**Sushmitha Srinath Kenkare**

Project Change Log			
Release #	Change Description	Date	Affected Page(s)
Iteration 1	Original release	3/10/2021	All
Iteration 2	Added line numbers to Project Description	3/26/2021	4
	Added required 'registration' details in R2 and 'payment' details in R8 of the requirements list.	3/27/2021	5
	Defined user roles well in Use Case diagrams to avoid ambiguity.	3/27/2021	9, 10
	Corrected dates and total effort on the increment matrix	3/27/2021	12
	Changed Domain Model Diagram accordingly as per Use Case Diagrams	3/30/2021	27
	Added Task List	04/3/2021	29
	Final Iteration 2 release	04/5/2021	4, 5, 9, 10, 12, 27, 29
Iteration 3	Placed relevant UIP next to EUC	04/20/2021	Pages 13 to 18
	Analysis Sequence diagram changed to Design Sequence Diagram	04/23/2021	Pages 19 to 26
	Domain Class Diagram changed with respect to Sequence Diagrams	04/25/2021	28
	Iteration 3 release	05/03/2021	Pages 13 to 26, 28

## TABLE OF CONTENTS

<b>PROJECT DESCRIPTION .....</b>	<b>4</b>
APP TITLE .....	4
APP DESCRIPTION.....	4
<i>Function 1.....</i>	<i>4</i>
<i>User-Registration .....</i>	<i>4</i>
<i>User-Login.....</i>	<i>4</i>
<i>Function 2 .....</i>	<i>4</i>
<i>Categorize items .....</i>	<i>4</i>
<i>Function 3.....</i>	<i>4</i>
<i>Item details.....</i>	<i>4</i>
<i>Function 4.....</i>	<i>4</i>
<i>Buy an item.....</i>	<i>4</i>
<i>Payments.....</i>	<i>4</i>
<b>REQUIREMENTS.....</b>	<b>5</b>
<b>USE CASE LIST.....</b>	<b>6</b>
<b>HIGH LEVEL USE CASE.....</b>	<b>7</b>
UC 1: REGISTER.....	7
UC 2: LOGIN .....	7
UC 3: SELECT ITEM CATEGORY .....	7
UC 3.1: VIEW AND SELECT ITEM TO PURCHASE FROM THE LISTING .....	7
UC 3.1.1: VIEW SELLER INFORMATION .....	7
UC 3.1.2:ENTER PAYMENT INFORMATION AND PAY .....	7
UC 4: ENTER PRODUCT DETAILS .....	7
UC 5: VIEW USER PROFILE .....	8
UC 5.1: EDIT USER DETAILS .....	8
UC 5.2: VIEW PURCHASE HISTORY .....	8
UC 5.3: VIEW AND MANAGE SELLING .....	8
UC 5.3.1: DELETE'ITEM FROM LISTING .....	8
UC 5.4: LOGOUT .....	8
UC 6: RESET FORGOTTEN PASSWORD.....	8
<b>USE CASE DIAGRAM .....</b>	<b>9-10</b>
<b>REQUIREMENTS TO USE CASE TRACEABILITY MATRIX.....</b>	<b>11</b>
<b>INCREMENT MATRIX .....</b>	<b>12</b>
<b>EXPANDED USE CASE AND UIP .....</b>	<b>13-18</b>
<b>SEQUENCE DIAGRAM.....</b>	<b>19-26</b>
<b>DOMAIN MODEL DIAGRAM .....</b>	<b>27</b>
<b>DOMAIN CLASS DIAGRAM.....</b>	<b>28</b>
<b>TASK LIST.....</b>	<b>29</b>
<b>CODE SNIPPETS .....</b>	<b>30-32</b>
<b>SUMMARY .....</b>	<b>33</b>

# PROJECT DESCRIPTION

- 1     • **App Title:- MAV\$HOP**
  - 2     • **App Description:-** We plan to create an app which will facilitate students to sell or buy items when moving in/moving  
3       out of the campus.  
4       Problems:- When students come to stay on-campus, they need necessary items like a table, chairs, microwave etc.  
5       Similarly, when students move out of the campus they sell the items which are unnecessary to them.  
6       Solution:- It would be easy for the students to perform buying and selling items through an app. Also, students will get  
7       the items at cheap rates and this app will be within UTA students only.  
8       a. Function 1:-
    - 9           • User-Registration -User will register with student email address and fill out all the necessary details required.  
10           The necessary details will include First-Name, Last-Name, Address, Phone number, valid UTA email address,  
11           password, confirm password.
    - 12           • User-Login -User will be redirected to the login page once he registers. He will login with email address and  
13           password. Reset password option is provided through the email link. We do not need admin login.
  - 14       b. Function 2:-
    - 15           • Categorize Items -User will view the items divided into categories. All the items will be organized into  
16           divisions like furniture, electronic devices, kitchen utensils, books, organizers, miscellaneous which will help  
17           for easy access.
  - 18       c. Function 3:-
    - 19           • Item Details -User who wishes to sell an item will upload photos of the item, write some description about  
20           product (which will be restricted to 180 characters) (required), enter the price(required), list the product  
21           in proper category(required) and provide phone number, name (required) on which he can be contacted  
22           on. User will also have negotiable radio button.
  - 23       d. Function 4:-
    - 24           • Buy an item -Once the user chooses the item, he will be provided with two options. 1) Click on buy now  
25           button where he will be directed to payment option or 2) Click on "contact seller" button where he will be  
26           provided with email address of the seller. The seller should mark the item as sold.
    - 27           • Payment -User enters the details of credit/debit card.
- 29       Resources:- SQLite
- 30
- 31     • **Team Member Experience:-**
    - 32       1. Neha Anil Shimpi:- I have worked on android projects in my undergraduate program. I developed an app which  
33           reminded the user of assignment completion dates in Java
    - 34       2. Rupal Ravindra Godbole: I have previous work experience in web development . I also have experience with  
35           different cloud platforms like AWS,Azure and IBM bluemix.I also have very basic knowledge of Node.js and  
36           MongoDB.
    - 37       3. Sushmitha Srinath Kenkare: I have worked on static and dynamic databases during my undergraduate and  
38           graduate studies. I also have previous experience in web design.
    - 39       4. Sucharitha Cheenepalli : I am good at database building and SQL queries. I have no prior experience in  
40           Android studios but can help with Java. I have a good knowledge of wireless communication networks and also  
41           worked in designing websites using elementor.
    - 42       5. Sai Sravanti Podili : I have a basic knowledge on Angular and STS applications. In my experience as a team we  
43           developed a Health Care System in which I have been the UI Developer.
    - 44       6. Akshat Ashwin Shah: I have worked on iOS app development during my internship at Darkhorse. I have basic  
45           knowledge of react.js and mongodb. I have worked on Game development on Unity platform and secured first  
46           rank.

## REQUIREMENTS

Req ID	Req Statement	Line Reference
R1	The app shall allow the user to choose between login or register.	Derived
R2	If the user wants to register, the app shall allow the user to enter details like first-name, last-name, email-id, password and address for the registration.	9,10,11
R3	If the user selects to login, the app shall allow the users to login through Authentication of user email-id and password.	12
R4	The app shall provide a reset password function for registered users.	13
R5	The app shall provide the user with list of categories.	15-17
R6	The app shall allow the user to select an item from the list.	Derived
R7	The app shall allow the user to buy the item which he has selected	24,25
R8	The app shall allow the user to enter payment details like card number, CVV, expiry date and name if willing to buy the item.	27
R9	The app should check the card details provided by user and process the payment.	Derived
R10	The app shall allow the user to sell the item after entering contact details, required information and uploading appropriate images of the item.	19-22
R11	The app shall allow user to edit his personal details such as Phone Number and Address in the profile section.	Derived
R12	The app shall provide user with his purchase and selling history.	Derived
R13	The app shall allow user to logout from application.	Derived

## USE CASE LIST

Use Cases	Use Case Name
UC1	Register
UC2	Login
UC3	Select Item Category
UC3.1	View and Select item to purchase from the listing.
UC3.1.1	View Seller Information
UC3.1.2	Enter Payment Information and Pay
UC4	Enter Product Details
UC5	View User Profile
UC5.1	Edit User Details
UC5.2	View Purchase History
UC5.3	View and Manage Selling
UC5.3.1	Delete Item from Listing
UC5.4	Logout
UC6	Reset Forgotten Password

## HIGH LEVEL USE CASE

- **UC 1: Register**

- TUCBW the user selects the ‘Register’ button on the App.
- TUCEW user is redirected to the ‘Login’ page and is prompted to login.

- **UC 2: Login**

- TUCBW user enters their credentials and clicks on ‘Sign in’ button.
- TUCEW user is redirected to the ‘Categories’ page.

- **UC 3: Select Item Category**

- TUCBW buyer selects a product category in the ‘Categories’ page.
- TUCEW buyer is redirected to the page that contains product listings from the selected category.

- **UC 3.1: View and Select the item to purchase from the listing**

- TUCBW buyer clicks on the product of their interest.
- TUCEW buyer is redirected to ‘Product Information’ page of the selected product.

- **UC 3.1.1: View Seller Information**

- TUCBW buyer clicks on product of their choice.
- TUCEW buyer completes viewing the seller information and may proceed to ‘Payment’ page.

- **UC 3.1.2: Enter Payment Information and Pay**

- TUCBW buyer clicks on the ‘Buy Now’ button on the ‘Product Information’ page.
- TUCEW buyer clicks on ‘Submit Payment’ and sees ‘Categories’ page.

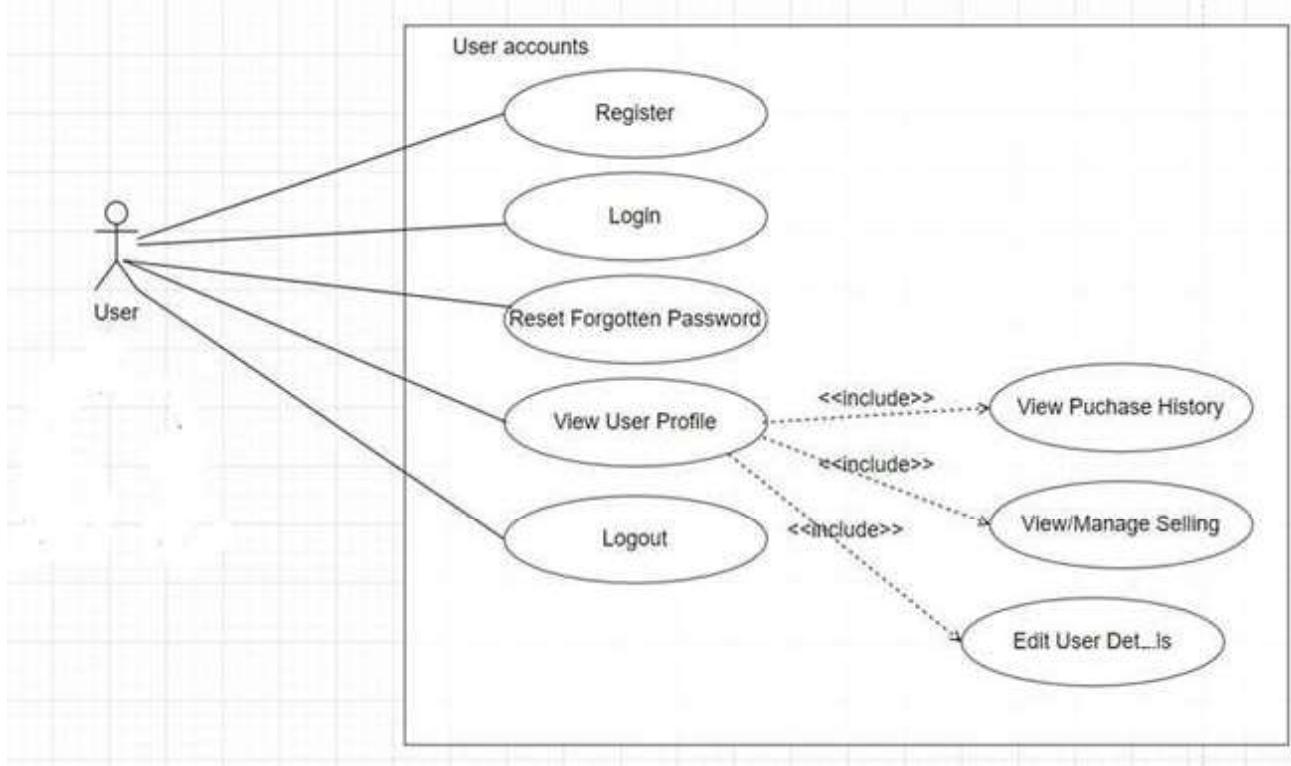
- **UC 4: Enter Product Details**

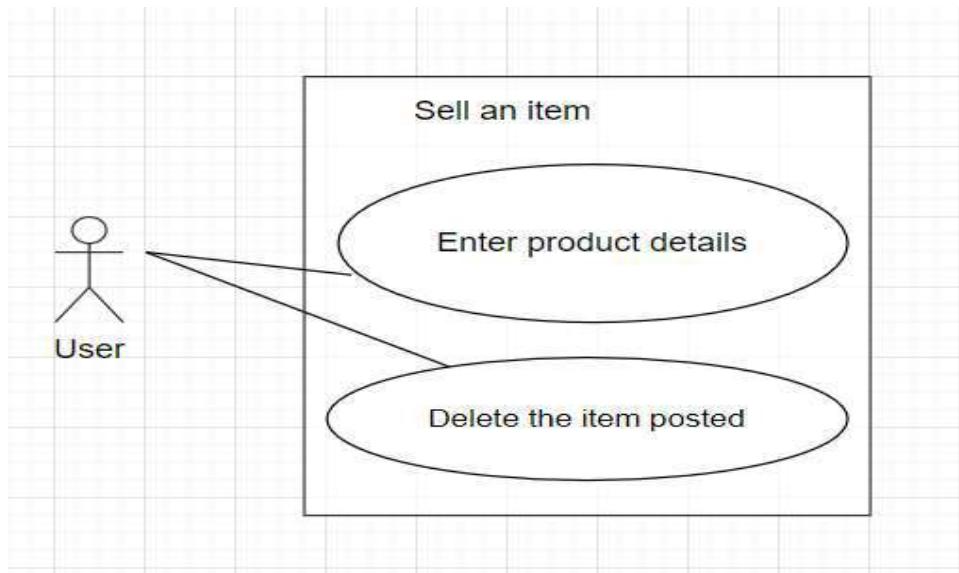
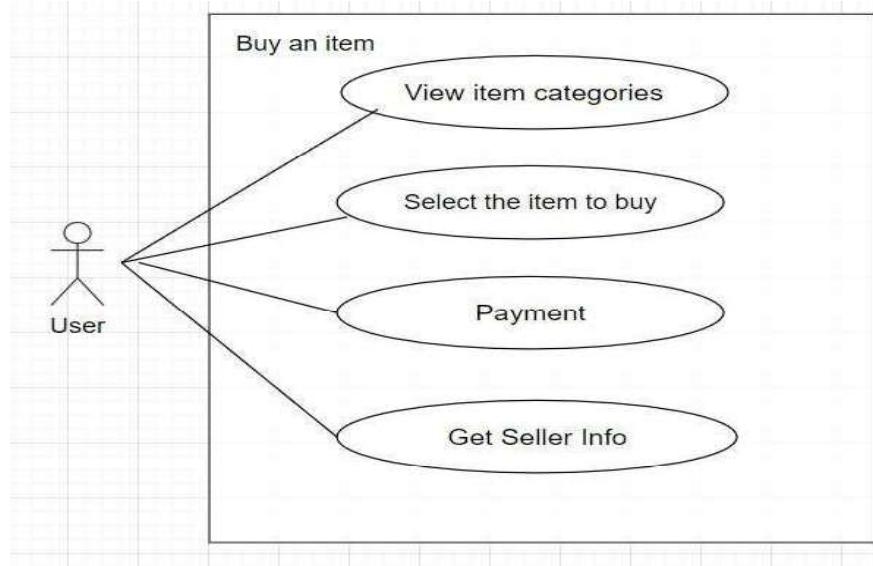
- TUCBW seller clicks on the ‘Post’ button on the bottom navigation bar.
- TUCEW seller clicks on the ‘Finish’ button and sees the ‘Categories’ page.

- **UC 5: View User Profile**
  - TUCBW user clicks on the ‘Profile’ button on the bottom navigation bar.
  - TUCEW user completes viewing personal information and may visit any other page such as ‘Categories’.
- **UC 5.1: Edit User Details**
  - TUCBW user clicks on the ‘Edit Information’ button in the ‘Profile’ page.
  - TUCEW user clicks on ‘Done’ and sees the ‘Profile’ page.
- **UC 5.2: View Purchase History**
  - TUCBW user clicks on ‘Your Orders’ button in the ‘Profile’ page.
  - TUCEW user completes viewing and sees ‘Profile’ page.
- **UC 5.3: View and Manage Selling**
  - TUCBW user clicks on ‘View and Manage Selling’ button in the ‘Profile’ page.
  - TUCEW user completes viewing and sees the ‘Profile’ page.
- **UC 5.3.1: Delete Item from Listing**
  - TUCBW user clicks on the ‘Delete’ icon provided next to a product listed in the ‘View and Manage Selling’ page.
  - TUCEW user confirms the deletion of the listing and sees the ‘View and Manage Selling’ page.
- **UC 5.4: Logout**
  - TUCBW user clicks on ‘Logout’ in the ‘Profile’ page.
  - TUCEW user is redirected to the ‘Login’ page.
- **UC 6: Reset Forgotten Password**
  - TUCBW the user clicks on ‘Forgot Password’ provided on ‘Login’ page.
  - TUCEW user clicks on ‘Reset Password’ and is redirected to ‘Login’ screen.

## USE CASE DIAGRAM

In this use case diagram, the user acts as a buyer when he/she prefers to buy items or, the user is also the seller if they wish to sell items. The user has access to register, login, reset the password, view their profile, and in the profile, they can buy, sell items, and can edit their details. Finally, the user can logout if they no longer prefer to be logged in.





## REQUIREMENTS TO USE CASE TRACEABILITY MATRIX

PRIORITY WEIGHT	UC1	UC2	UC3	UC3.1	UC3.1.1	UC3.1.2	UC4	UC5	UC5.1	UC5.2	UC5.3	UC5.3.1	UC5.4	UC6
R1      1	X	X												
R2      1	X													
R3      1		X												
R4      5														X
R5      2			X											
R6      3				X										
R7      3					X									
R8      3						X								
R9      3							X							
R10     2							X							
R11     4								X	X					
R12     4									X	X	X			
R13     5												X		
SCORE	2	2	2	3	3	6	2	4	4	4	4	4	5	5

**NOTE: Priority weight 1 is the highest priority**

## INCREMENT MATRIX

Use Case	Priority	Effort (Person - weeks)	Depends On	Iteration 1 (03/10/2021)	Iteration 2 (04/05/2021)	Iteration 3 (05/03/2021)
UC1	2	3	NONE	3		
UC2	2	3	UC1	3		
UC3	2	2	UC2		2	
UC3.1	3	5	UC3			5
UC3.1.1	3	1	UC3.1			1
UC3.1.2	6	2	UC3.1.1			2
UC4	2	4	UC2			4
UC5	4	6	UC2			6
UC5.1	4	1	UC5			1
UC5.2	4	2	UC5			2
UC5.3	4	2	UC5			2
UC5.3.1	4	1	UC5			1
UC5.4	5	2	UC5.4			2
UC6	5	2	NONE	2		
<b>Total Effort</b>		<b>36</b>		<b>8</b>	<b>2</b>	<b>26</b>
<b>1 PW = 5 Hrs</b>				<b>40</b>	<b>10</b>	<b>130</b>

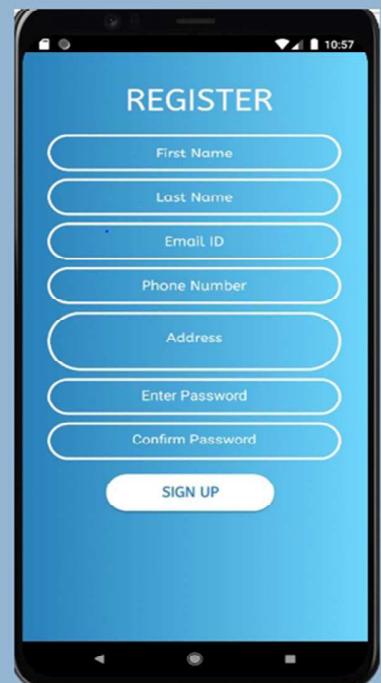
**TOTAL PW = 180**

## EXPANDED USE CASES AND UIP

### EUC1: Register

**Precondition:** This use case assumes user is on the main page and wants to register for the application

Actor: User	System: MavShop
<p><b>1. TUCBW</b> the user selects the 'Register' button on the App.</p> <p><b>3. User enter all his details and click on register button.</b></p> <p><b>5. TUCEW</b> user is redirected to the 'Login' page and is prompted to login.</p>	<p><b>0. System displays the main page.</b></p> <p><b>2. User is redirected to Register page.</b></p> <p><b>*4. System validates all the data entered by user and create a new entry into the database if no error is found and redirected to "login" page.</b></p>
	<p><b>Postcondition:</b> User is registered for application and can login.</p>



### EUC2: Login

**Precondition:** User must be registered into MavShop

Actor: User	System: MavShop
<p><b>1. TUCBW</b> user enters their credentials and clicks on 'Sign in' button.</p> <p><b>5. TUCEW</b> user is redirected to the 'Categories' page.</p>	<p><b>0. System displays the login page.</b></p> <p><b>*2. System will validate user input and display the following:</b></p> <ul style="list-style-type: none"> <li>1. If the login credentials don't match, user will be showed "User not found! Please Register" message.</li> <li>2. If the user puts any email id other than UTA email address he will be displayed with message saying "Invalid email id. Please use UTA email address".</li> <li>3. If the login credentials match system, then user will be redirected to app homepage.</li> <li>4. If user forgets the password, he will be asked to reset password.</li> </ul>
	<p><b>Postcondition:</b> User must be provided with the Categories page which displays various product categories.</p>

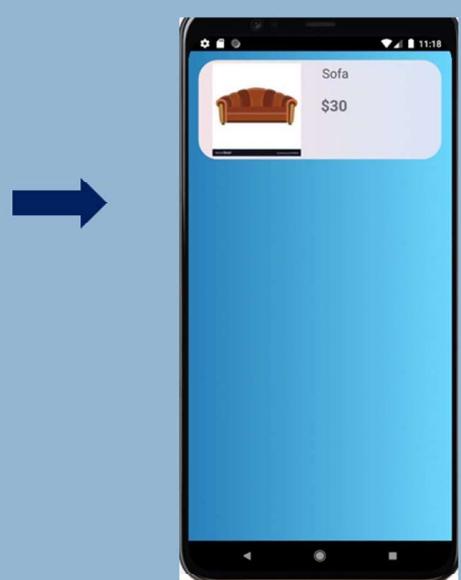


### EUC3.1: View and Select Item to Purchase from the Listing

**Precondition:** Use case assumes that the user is on Categories Page

Actor: User	System: MavShop
1. TUCBW buyer clicks on the product of their interest.	0. System displays the product categories.  *2. System should fetch the products of the selected category from the database.
3. TUCEW buyer is redirected to 'Product Information' page of the selected product.	

**Postcondition:** User should be able to select any product which he wants to buy.

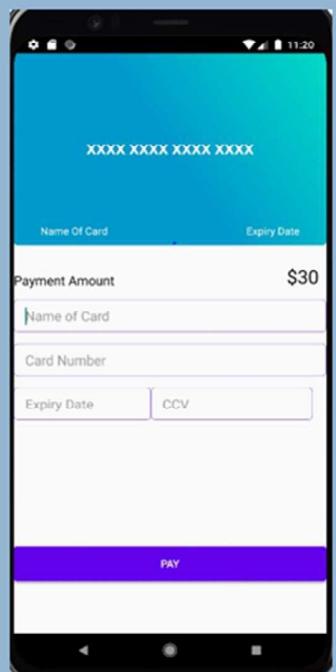


### EUC3.1.2: Enter Payment Information and Pay

**Precondition:** Use case assumes that the user has selected the product he wants to buy.

Actor: User	System: MavShop
1. TUCBW buyer clicks on the 'Buy Now' button on the 'Product Information' page.	* 0. System displays the product information to the product he wants to buy.
3. User enters the payment details and clicks on submit button.	2. System displays the form to enter payment details of the user.  *4. System will validate the user card credentials and display the following:  1.If the card details entered are correct, user will be shown "Payment Successful" message. 2.If the card number is not 16-digit, user will be shown "Invalid Card Number" message. 3.If the card CVV number is not three-digit, user will be shown "Invalid CVV" message.
5. TUCEW buyer clicks on 'Submit Payment' and sees 'Categories' page.	

**Postcondition:** Product should be disabled from the product list.

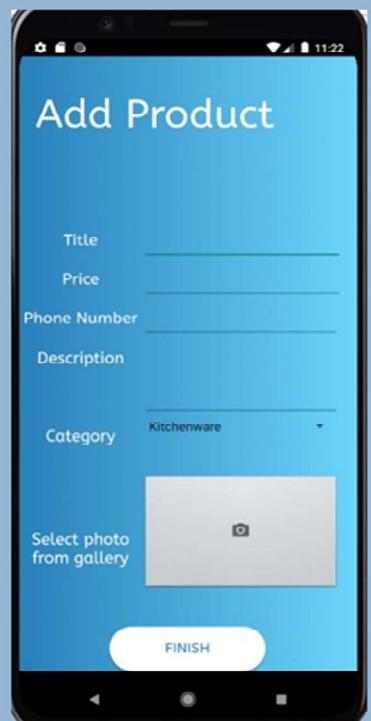


## EUC4: Enter Product Details

**Precondition:** This use case assumes user is on MavShop and wants to sell a product.

Actor: User	System: MavShop
1. <b>TUCBW</b> seller clicks on the 'Post' button on the bottom navigation bar.	0. System displays the app pages with bottom navigation bar.  2. System displays a page where user should fill up product details.
3. The user enters all the necessary details of the product and clicks the "submit" button.	*4. System validates all the mandatory field information and details are inserted into database.
5. <b>TUCEW</b> seller clicks on the 'Finish' button and sees the 'Categories' page.	

**Postcondition:** User must see the product he added in the product listing.

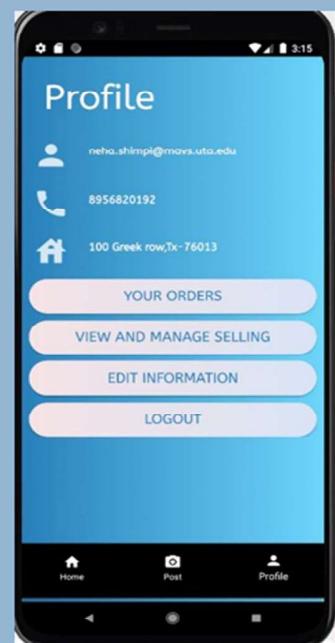


## EUC5: View User Profile

**Precondition:** This use case assumes user wants to view Profile Page

Actor: User	System: MavShop
1. <b>TUCBW</b> user clicks on the 'Profile' button on the bottom navigation bar.	0. System displays the app pages with bottom navigation bar.  *2. System displays the profile page where user's personal details, order history and selling information are displayed.
3. <b>TUCEW</b> user completes viewing personal information and may visit any other page such as 'Categories'.	

**Postcondition:** User must see the edited fields in the personal details.

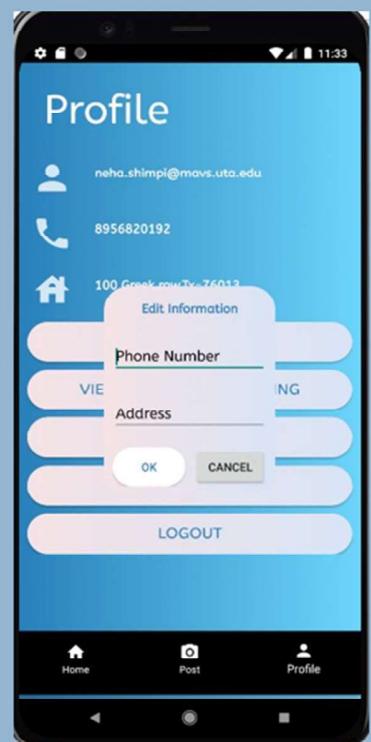


## EUC5.1: Edit User Details

**Precondition:** This use case assumes user is on Profile Page

Actor: User	System: MavShop
<p><b>1. TUCBW</b> user clicks on the 'edit information' button provided on the 'Profile' page.</p>	<p><b>0.</b> System displays the user personal details.</p> <p><b>*2.</b> System validates the values and inserts the new values to the database.</p>
<p><b>3. TUCEW</b> user clicks on 'Done' and sees the 'Profile' page.</p>	

**Postcondition:** User must see the updated information on the Profile Page.



## EUC5.2: View Purchase History

**Precondition:** This use case assumes user is on Profile Page

Actor: User	System: MavShop
<p><b>1. TUCBW</b> user clicks on 'Your Orders' button in the 'Profile' page.</p>	<p><b>0.</b> System displays the user personal details.</p> <p><b>*2.</b> System checks all the purchase history of that particular user from the database and return it to "purchase history" page.</p>
<p><b>3. TUCEW</b> user completes viewing and sees 'Profile' page.</p>	

**Postcondition:** User has seen his purchase history up to date.



## EUC5.3: View and Manage Selling

**Precondition:** This use case assumes user is on Profile Page and wants to view Manage selling tab.

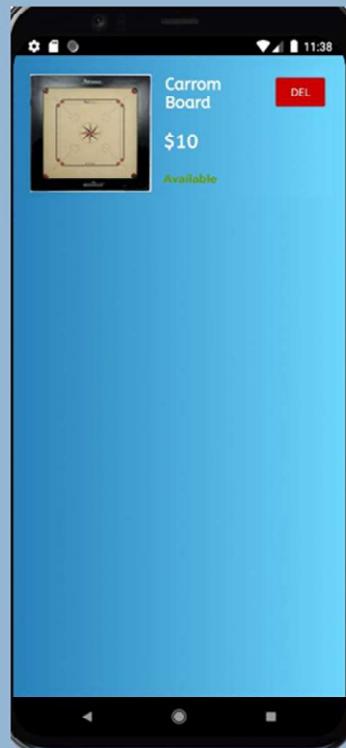
Actor: User	System: MavShop
1. TUCBW user clicks on 'View and Manage Selling' button in the 'Profile' page.	0. System displays the Profile Page.  *2. System displays a page where all the products previously sold by the user and his active listings are displayed.
3. TUCEW user completes viewing and sees the 'Profile' page.	
<b>Postcondition:</b> User has seen his Selling history and active listings up to date	



### EUC5.3.1: Delete Item from Listing

**Precondition:** This use case assumes user is on View and Manage Selling Page

Actor: User	System: MavShop
1. TUCBW user clicks on the 'Delete' icon provided next to a product listed in the 'View and Manage Selling' page.	0. System displays the active products listing.  *2. System deletes the entry of that product from the database.
3. TUCEW user confirms the deletion of the listing and sees the 'View and Manage Selling' page.	
<b>Postcondition:</b> User must not see the deleted product in the products listing.	

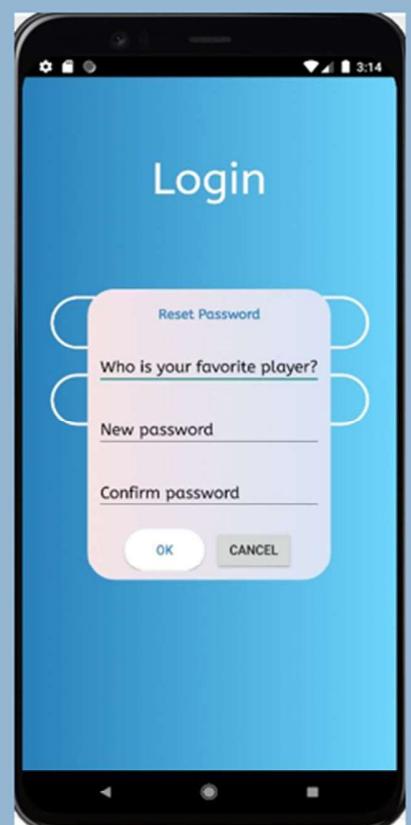


## EUC6: Reset Forgotten Password

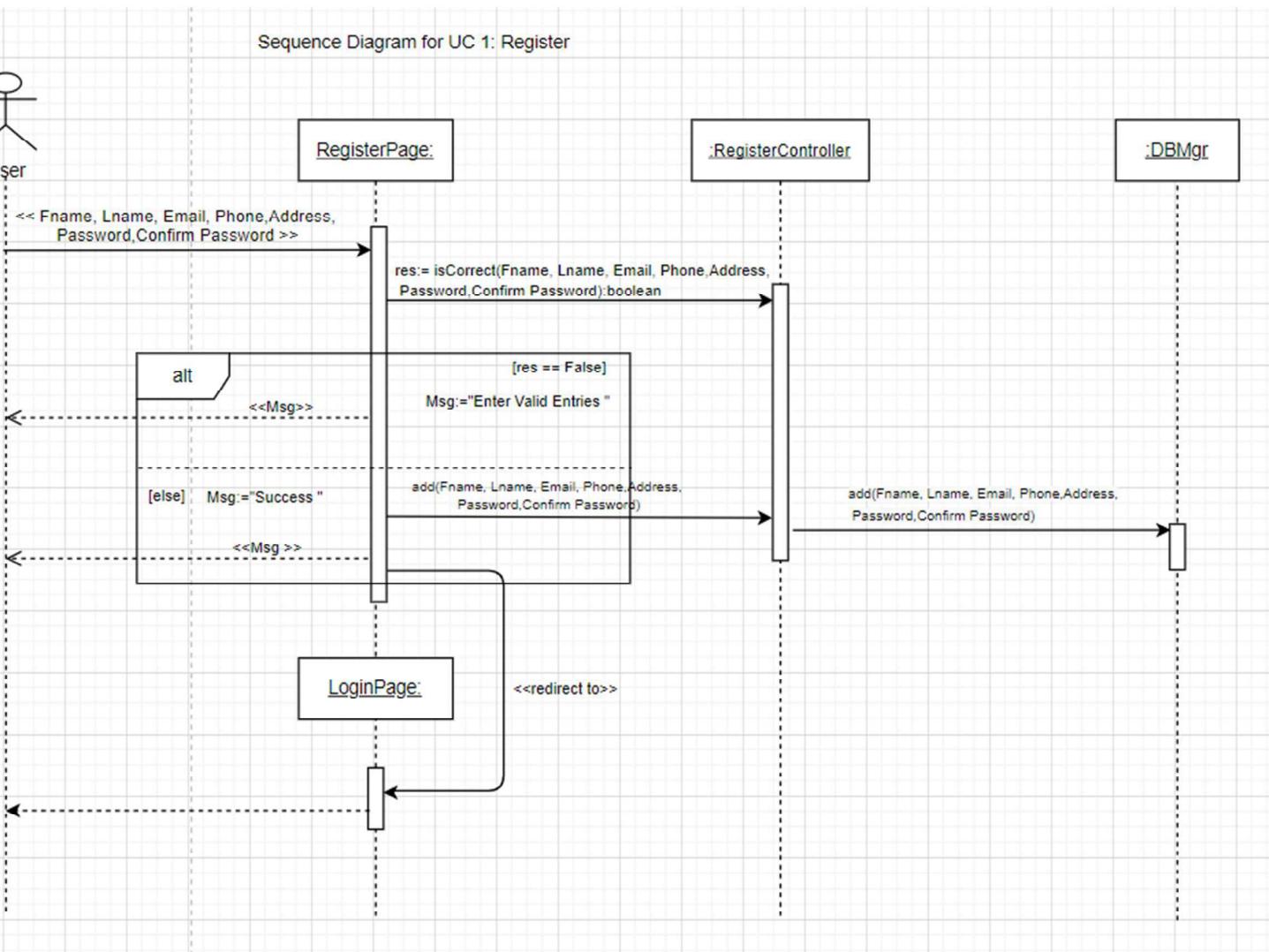
**Precondition:** This use case assumes user has forgotten the password and clicked on Forgot password link.

Actor: User	System: MavShop
	<b>0.</b> System displays the Login UI page.
<b>1.</b> <b>TUCBW</b> the user clicks on 'Forgot Password' provided on 'Login' page.	<b>2.</b> System displays the Reset - Password Page.
<b>3.</b> User enters the answer to security question and new password he wants to set and clicks on Submit	<b>*4.</b> New password is updated in the database.
<b>5.</b> <b>TUCEW</b> user clicks on 'Reset Password' and is redirected to 'Login' screen.	

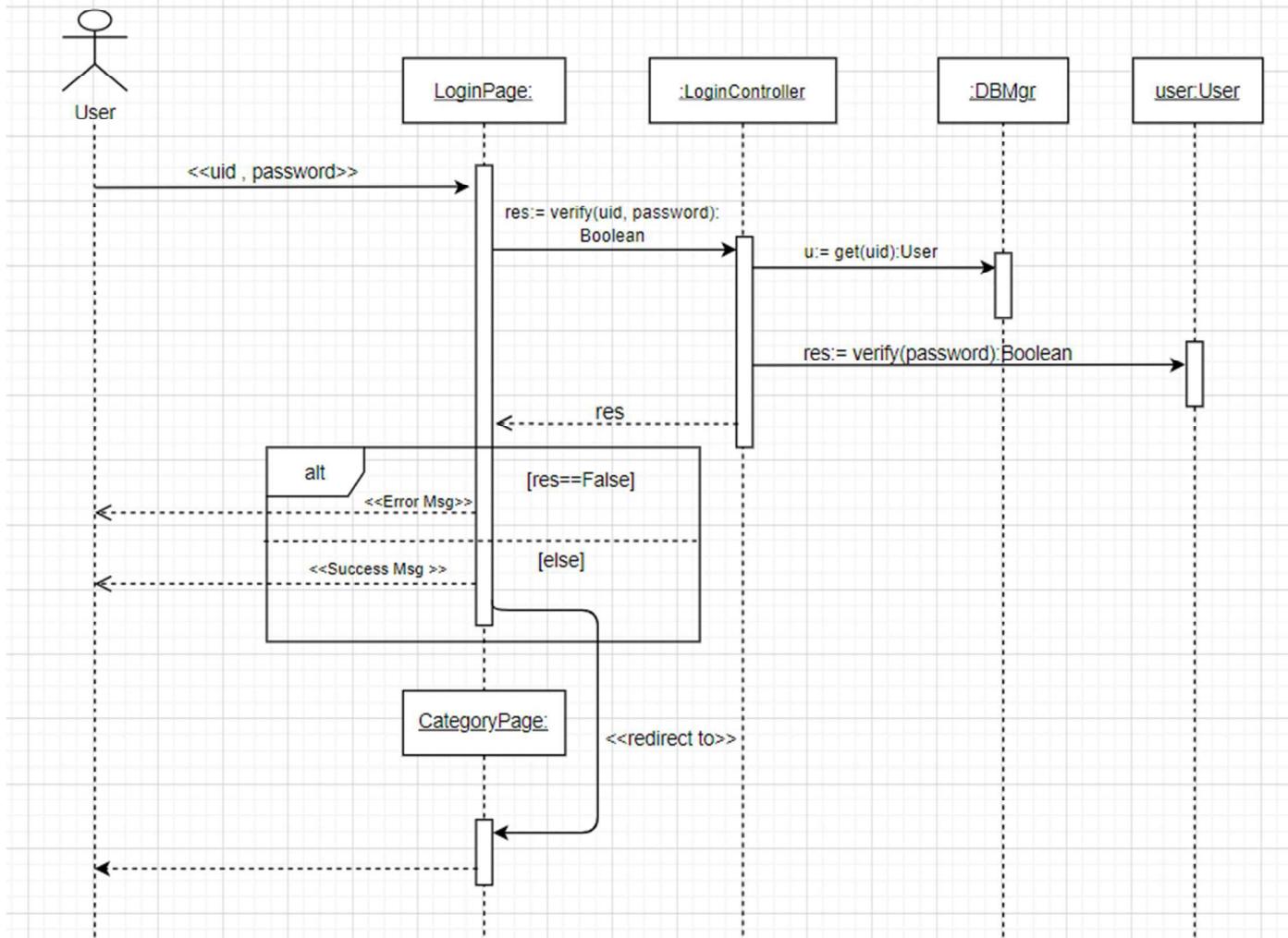
**Postcondition:** User must sign in to application with the new password which he just changed.



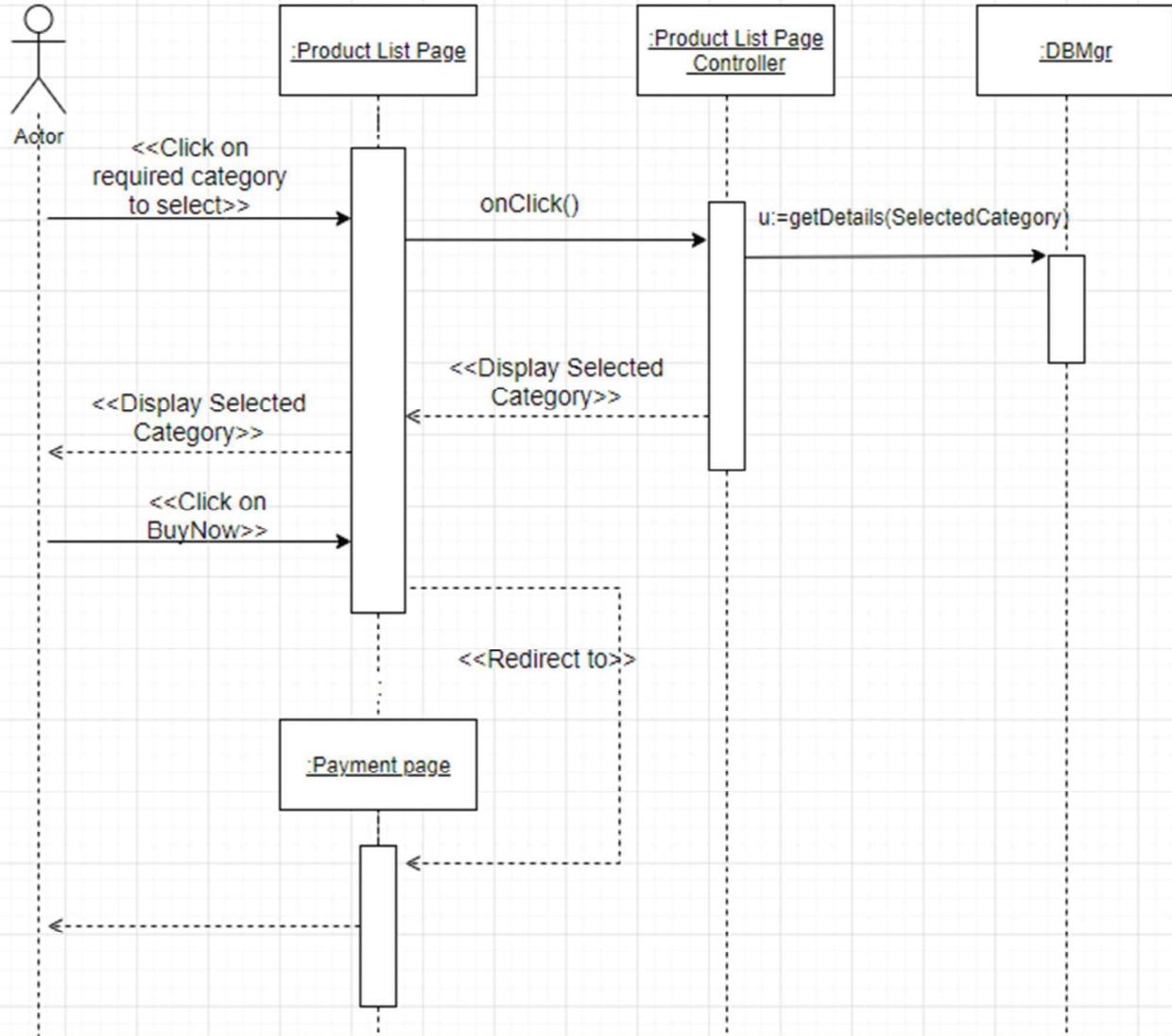
# SEQUENCE DIAGRAMS



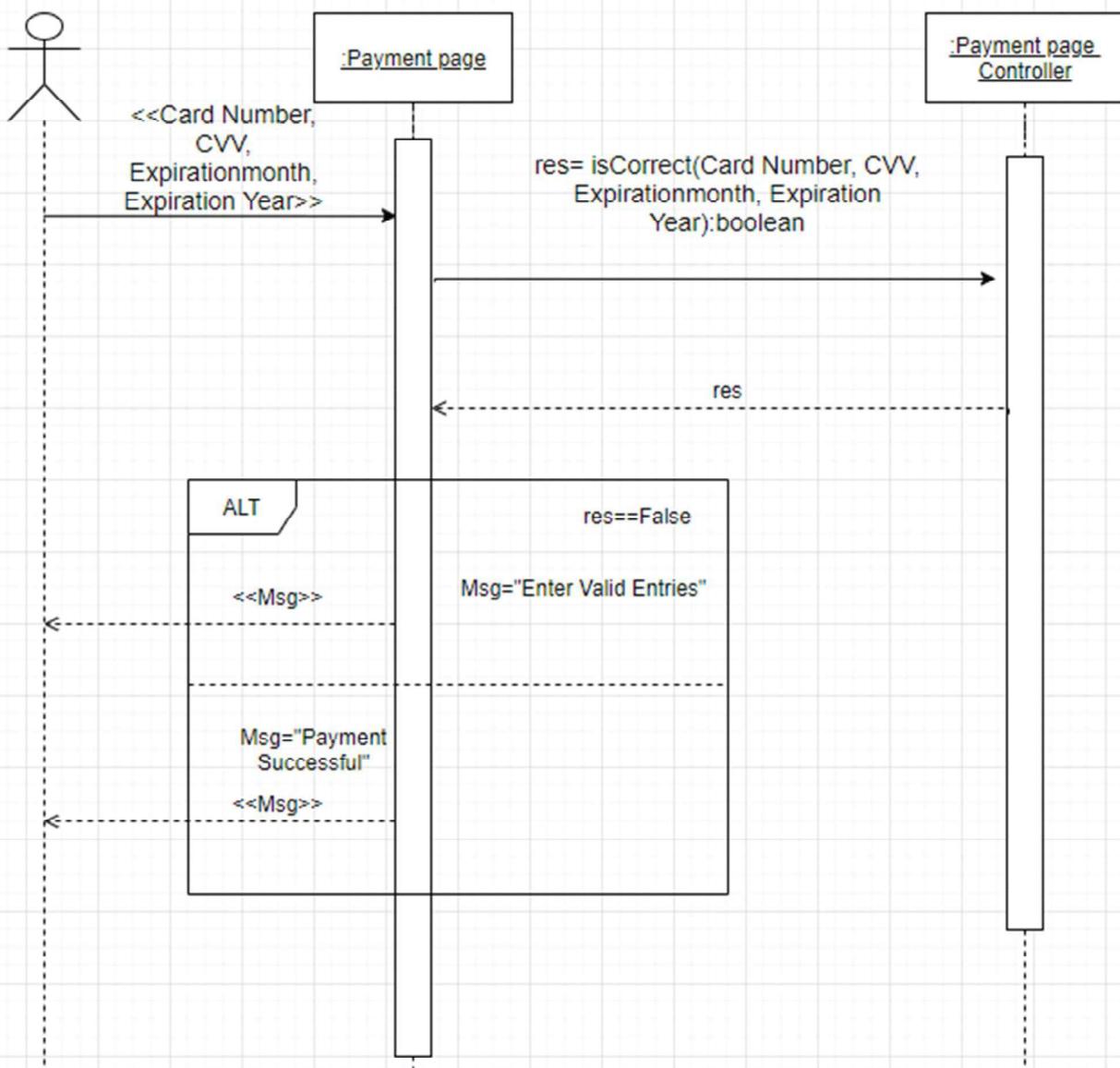
Sequence Diagram for UC 2: Login



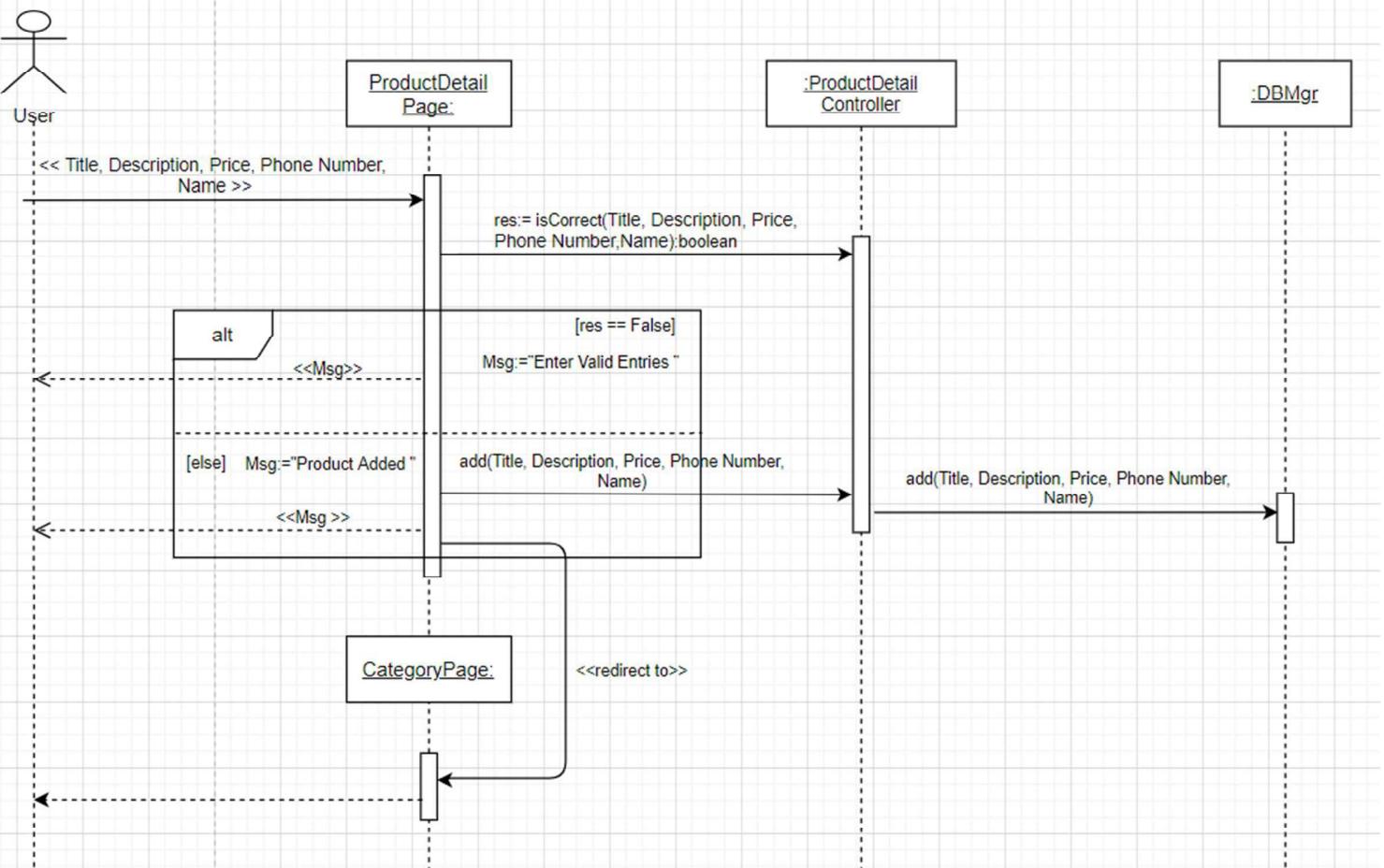
UC 3.1 View and select item to purchase from the listing



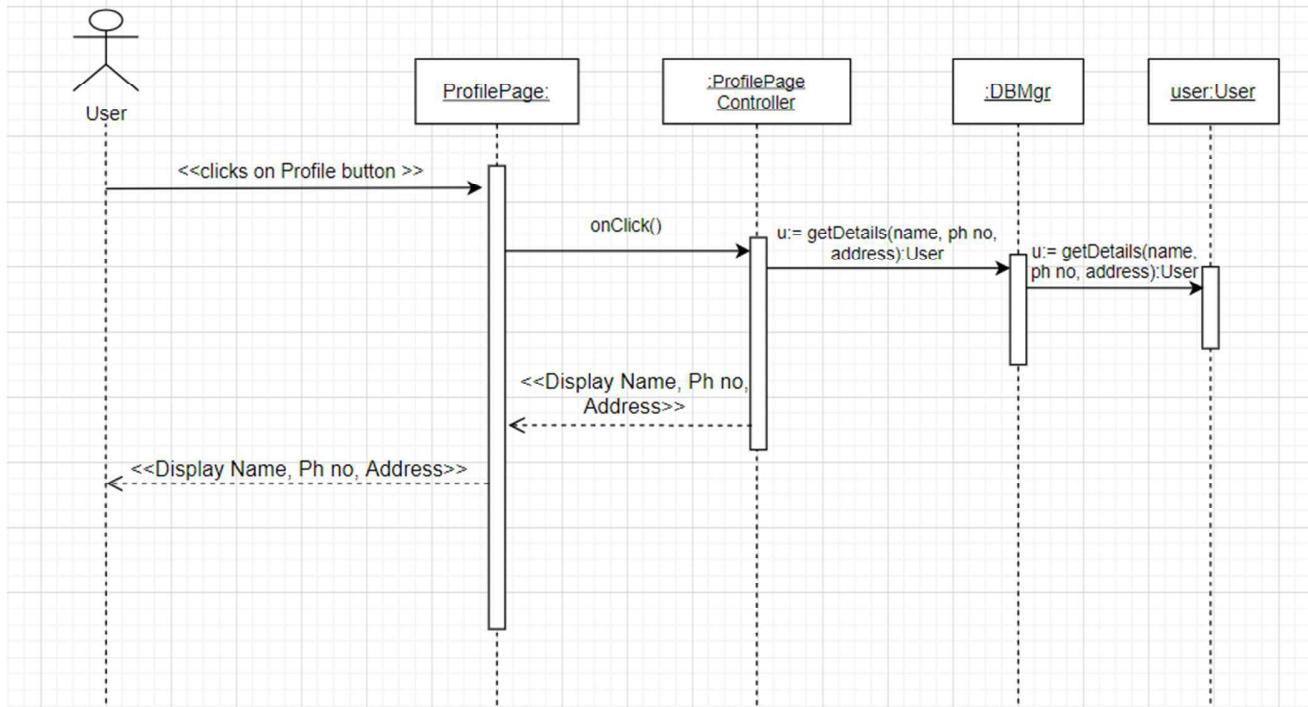
UC 3.1.2 Enter Payment information and Pay



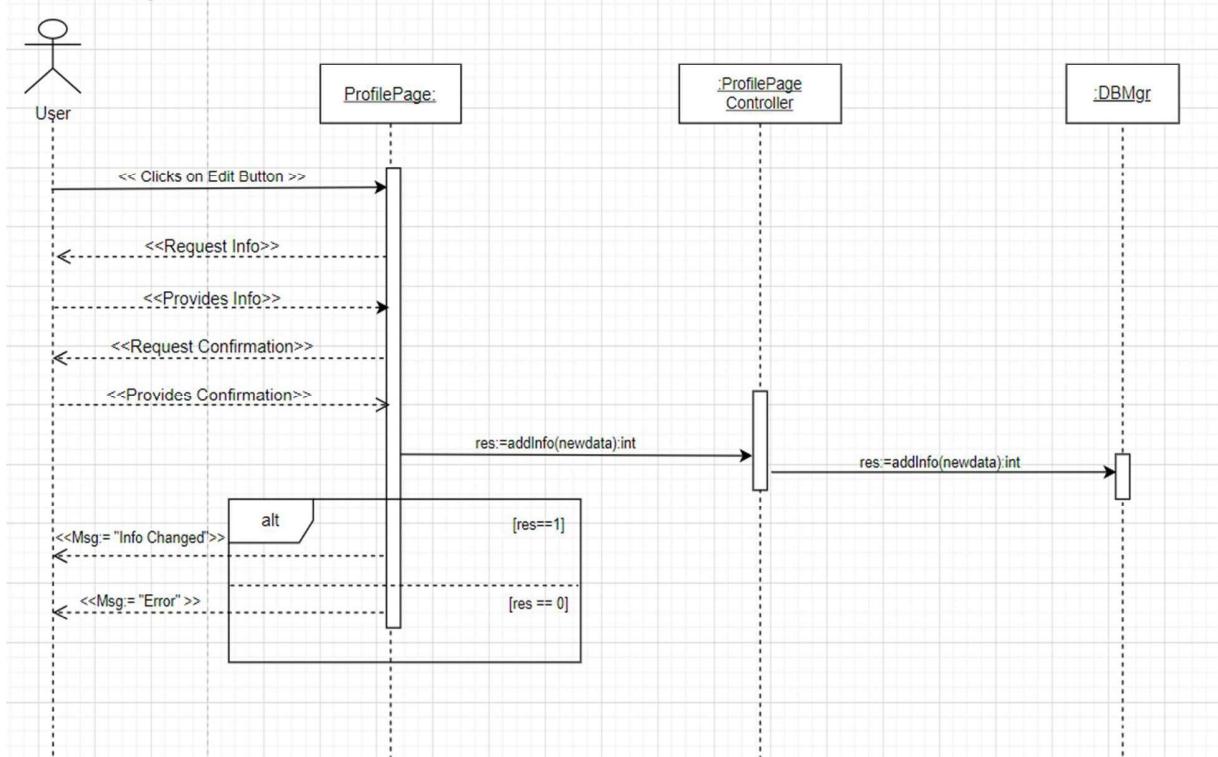
Sequence Diagram for UC 4: Enter Product Details



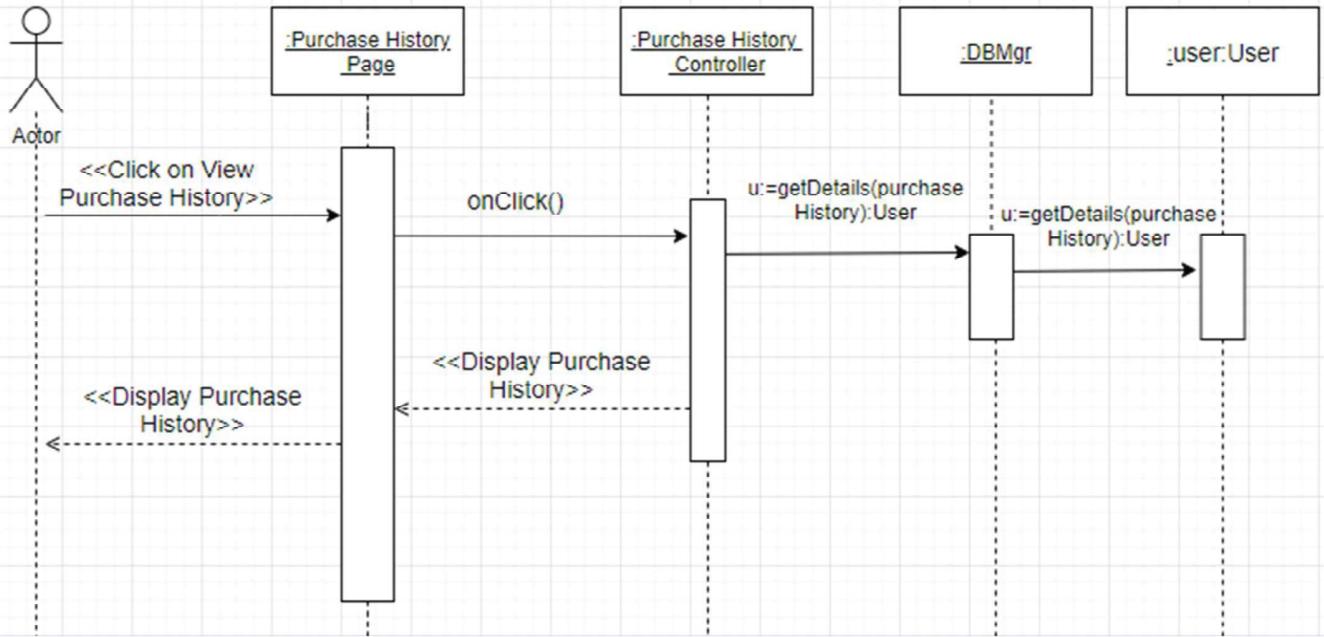
Sequence Diagram for UC 5: View User Profile



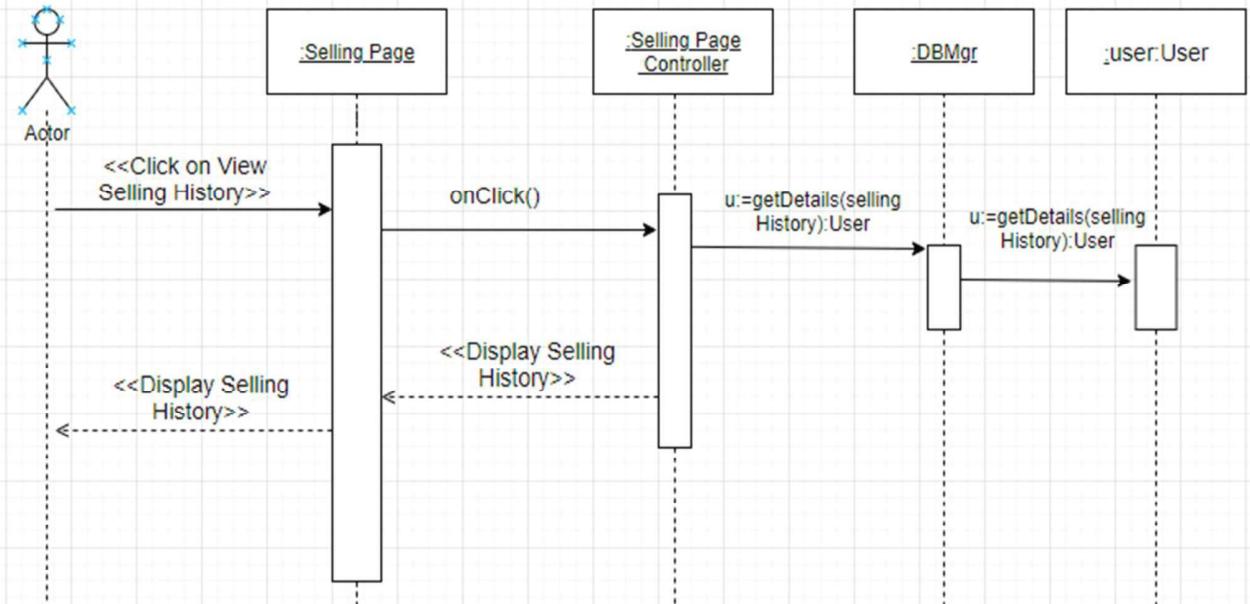
Sequence Diagram for UC 5.1: Edit User Details



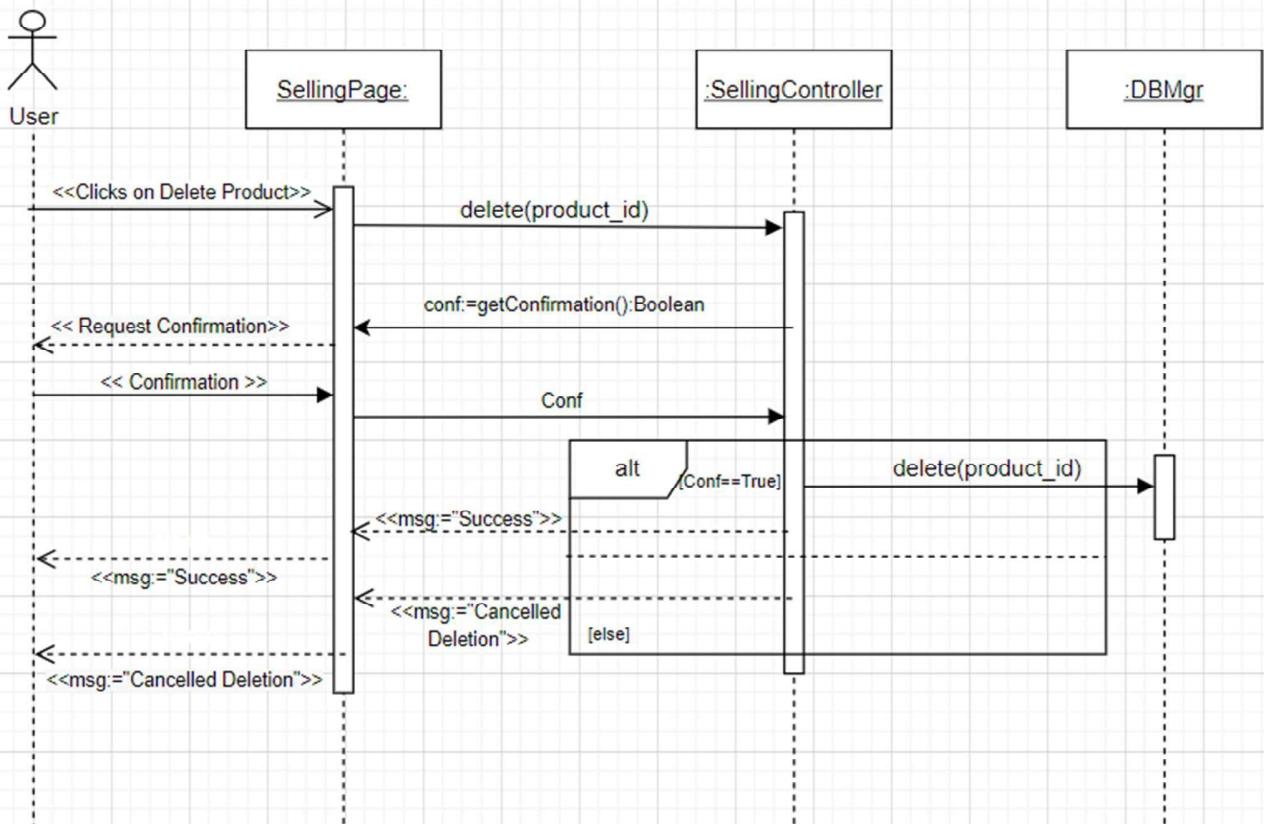
### UC 5.2 View Purchase History



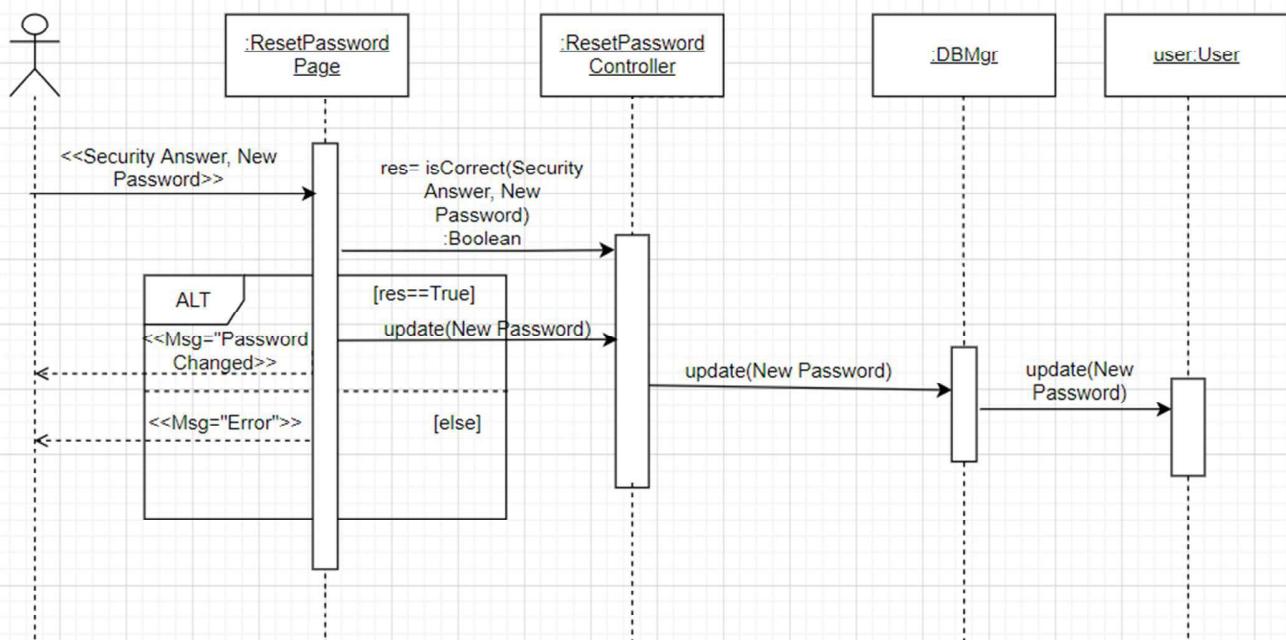
### UC 5.3 View and Manage Selling



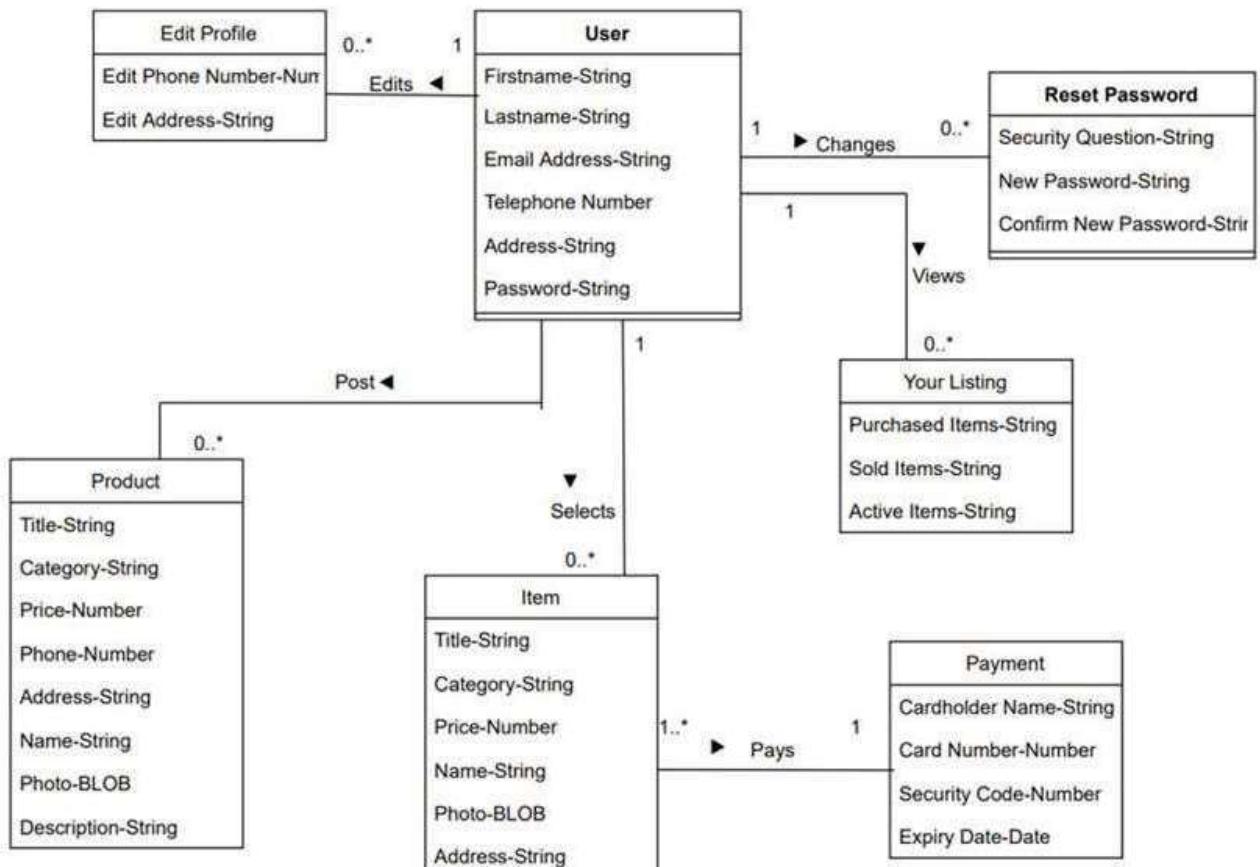
Sequence Diagram for UC 5.3.1 - Delete Item from Listing



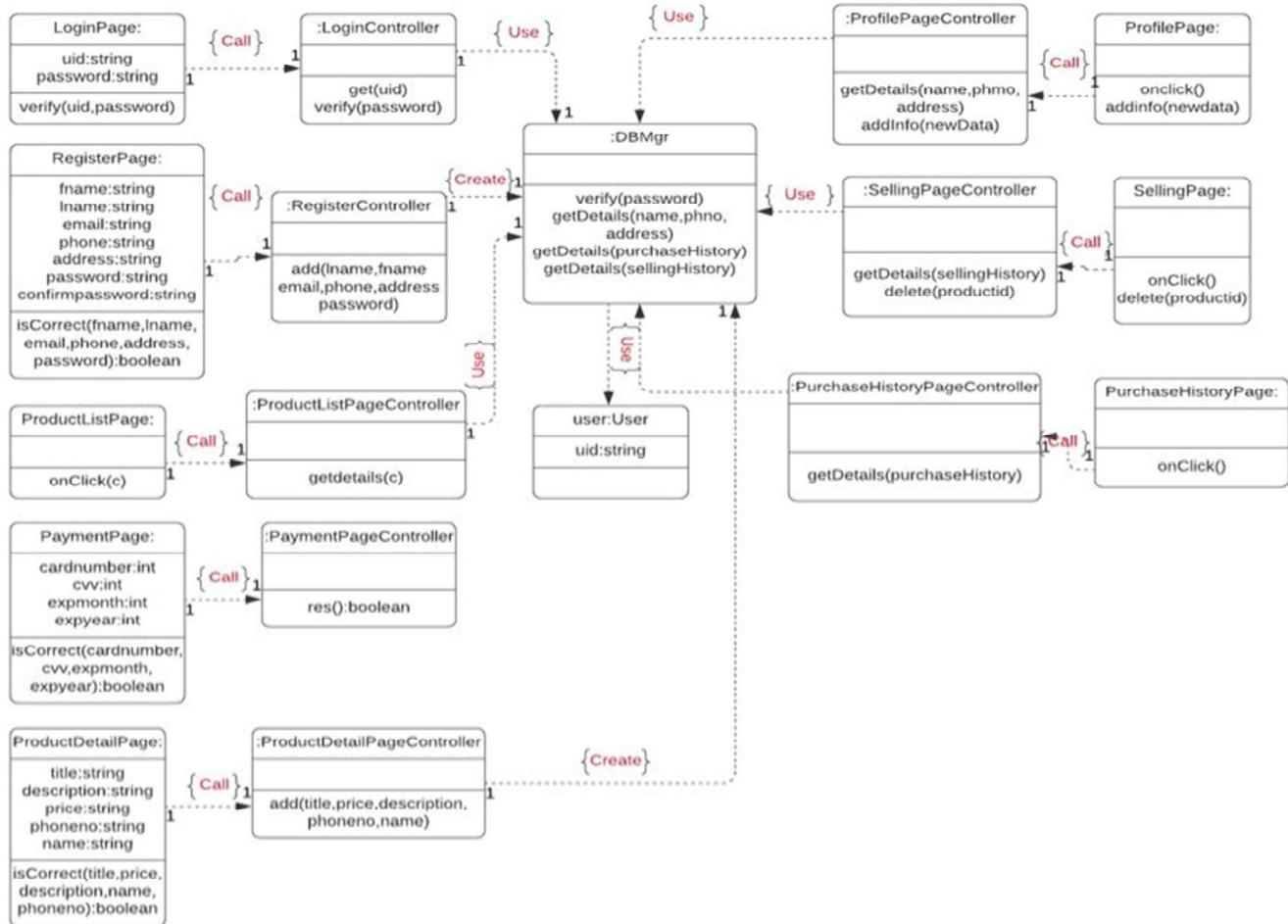
UC 6: Reset Forgotten Password



## DOMAIN MODEL DIAGRAM



# DOMAIN CLASS DIAGRAM



## TASK LIST

Use Case #	Use Case Name	Team Member Assigned	Iteration(Due by)
UC1	Register	Akshat	03/10/2021
UC2	Login	Akshat	03/10/2021
UC3	Select Item Category	Neha	04/05/2021
UC3.1	View and Select the item to purchase from the listing	Sucharitha	05/03/2021
UC3.1.1	View Seller Information	Sushmitha	05/03/2021
UC3.1.2	Enter Payment Information and Pay	Rupal	05/03/2021
UC4	Enter Product Details	Sucharitha	05/03/2021
UC5	View User Profile	Sravanthi	05/03/2021
UC5.1	Edit User Details	Sravanthi	05/03/2021
UC5.2	View Purchase History	Rupal	05/03/2021
UC5.3	View and Manage Selling	Sushmitha	05/03/2021
UC5.3.1	Delete Item from Listings	Neha	05/03/2021
UC5.4	Logout	Akshat	05/03/2021
UC6	Reset Forgotten Password	Akshat	05/03/2021

# CODE SNIPPETS

## Register Page

### XML

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fillViewport="true">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".Register">

        <LinearLayout
            android:id="@+id/linearLayout3"
            android:layout_width="150dp"
            android:layout_height="498dp"
            android:clickable="false"
            android:orientation="vertical"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintVertical_bias="0.595">

            <TextView
                android:id="@+id/fn"
```

### JAVA

```
private void registerUser() {

    firstName_str = firstName.getText().toString();
    lastName_str = lastName.getText().toString();
    emailid_str = emailid.getText().toString();
    phone_str = phone.getText().toString();
    address_str = address.getText().toString();
    password_str = password.getText().toString();
    confirmPassword_str = confirmPassword.getText().toString();

    if(firstName_str.equals("")||lastName_str.equals("")||emailid_str.equals("")||phone_str.equals("")||address_str.equals("")||password_str.equals("")||confirmPassword_str.equals(""))
    {
        Toast.makeText(context: Register.this, text: "Some field is missing!",Toast.LENGTH_LONG).show();
    }
    else if(!emailid_str.contains("@mavs.uta.edu"))
    {
        Toast.makeText(context: Register.this, text: "Please enter UTA Email-Id only!",Toast.LENGTH_LONG).show();
    }
    else if(!password_str.equals(confirmPassword_str))
    {
        //System.out.println(password_str+ " "+confirmPassword_str);
        Toast.makeText(context: Register.this, text: "Passwords do not match!",Toast.LENGTH_LONG).show();
    }
    else if(phone_str.length()!=10)
    {
        Toast.makeText(context: Register.this, text: "Please enter valid Phone Number!",Toast.LENGTH_LONG).show();
    }
}
```

## Login Page

### XML

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:and
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Login">

    <TextView
        android:id="@+id/textView2"
        android:layout_width="99dp"
        android:layout_height="58dp"
        android:layout_marginTop="88dp"
        android:text="Login"
        android:textSize="36sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.506"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textView4"
        android:layout_width="121dp"
        android:layout_height="35dp"
        android:layout_marginTop="88dp"
        android:text="User ID" />
```

### Java

```
private void login_auth() {
    DataBaseAdapter mDbHelper = new DataBaseAdapter(this.getApplicationContext());
    mDbHelper.createDatabase();
    mDbHelper.open();
    email_str = email_text.getText().toString();
    password_str = password_text.getText().toString();

    if(email_str.equals("") && password_str.equals(""))
    {
        Toast.makeText( context: Login.this, text: "Please enter the details!",Toast.LENGTH_LONG).show();
    }
    else if(email_str.equals("") || password_str.equals(""))
    {
        Toast.makeText( context: Login.this, text: "UserName and Password missing!",Toast.LENGTH_LONG).show();
    }
    else if(!email_str.contains("mavs.uta.edu"))
    {
        Toast.makeText( context: Login.this, text: "Invalid Email-Id.Please enter valid id!",Toast.LENGTH_LONG).show();
    }
    else
    {
        if(mDbHelper.checkifUserExists(email_str,password_str))
        {
            Toast.makeText( context: Login.this, text: "Login successful!",Toast.LENGTH_LONG).show();
            Intent intent1 = new Intent( packageContext: this, categories.class);
            startActivity(intent1);
        }
    }
}
```

## Categories Page

**XML:**

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".categories">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_scrollFlags="scroll|enterAlways"
        >

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:orientation="vertical">

            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:padding="20dp"
                android:text="Categories"
                android:textSize="30sp" />
        
```

**Java:**

```
public class Categories extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_categories);
    }

    public void OpenFurniture (View view) {
        Intent intent1 = new Intent( mContext, Furniture.class);
        startActivity(intent1);
    }

    public void OpenElectronics (View view) {
        Intent intent1 = new Intent( mContext, Electronics.class);
        startActivity(intent1);
    }

    public void OpenKitchenware (View view) {
        Intent intent1 = new Intent( mContext, Kitchenware.class);
        startActivity(intent1);
    }

    public void OpenOrganizers (View view) {
        Intent intent1 = new Intent( mContext, Organizers.class);
        startActivity(intent1);
    }
}
```

# SUMMARY

- Mav\$hop is an android application which aims at buying and selling of product within UTA.
- Iteration 3 helped to learn:
  - Design implementation of the rest of the project.
  - Detailed flow of events between components.
  - Hands-on coding of the functionality.
- Completed the coding and testing phases of the project.

Project Demo Link: <https://youtu.be/GgLKF9vBVLU>