# Table of Contents

1. Abstract
2. Introduction
3. Software Requirements Specification
4. Design

1.ABSTRACT

2. INTRODUCTION

2.1 Brief Introduction:

News feeds are an example of automated syndication. News feed technologies allow information to be automatically provided and updated on Web sites, emailed to users, etc. As the name implies news feeds are normally used to provide news; however, the technology can be used to syndicate a wide range of information. The BBC ticker is an example of a news feed application. A major limitation with this approach is that the ticker can only be used with information provided by the BBC. We tried to create that type of news service using WCF service which provides real time updates.

2.2 Tools/Technologies Used:

Technologies:

* WCF – Windows Communication Foundation
* ASP.net Web Application

Tools:

* Microsoft Visual Studio 2019
* SQL server

3. SOFTWARE REQUIREMENTS

SPECIFICATION

Types of Users :

* Service subscriber
* Service provider

System function requirements:

R.1 Service subscriber

R.2.1 Add news update :

Input : News Information.

Output : Confirmation message.

R.2.2 View all news updates :

Input : User selection.

Output : List of all news updates.

R.2.3 Search news updates :

Input : Keyword.

Output : Information about that news.

R.2.4 Update news :

Input : New news updates

Output : Success message.

R.2.5 Delete news update :

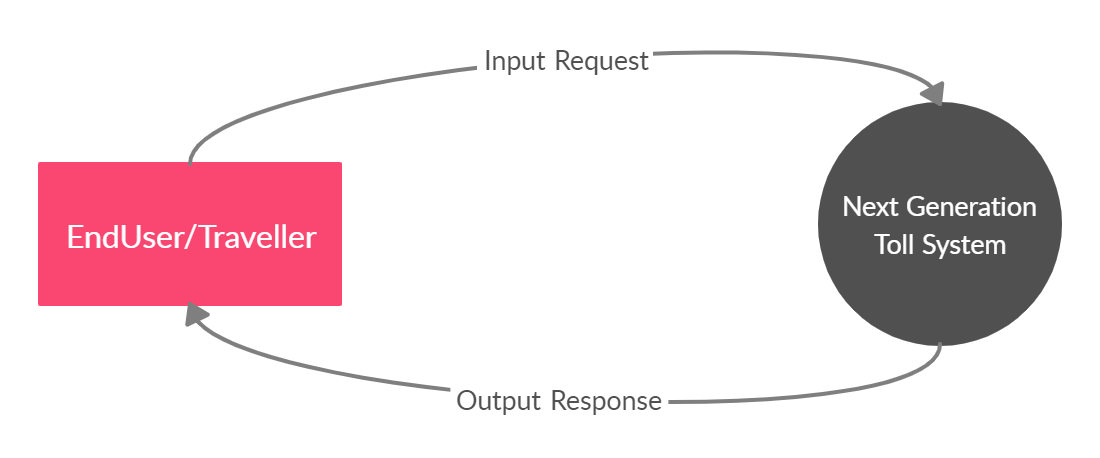
Input : Select news updates .

Output : Success message.

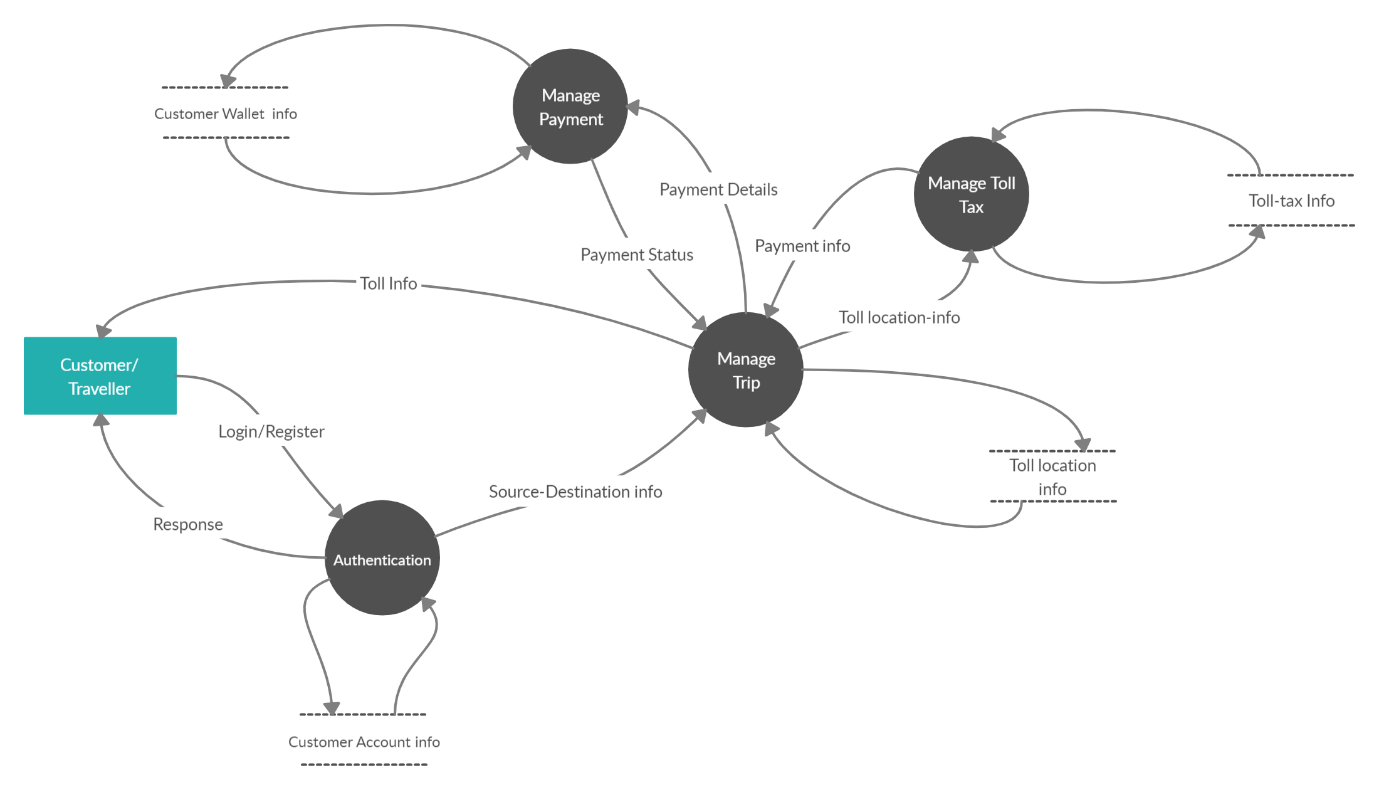
4. DESIGN

**DFD diagram :**

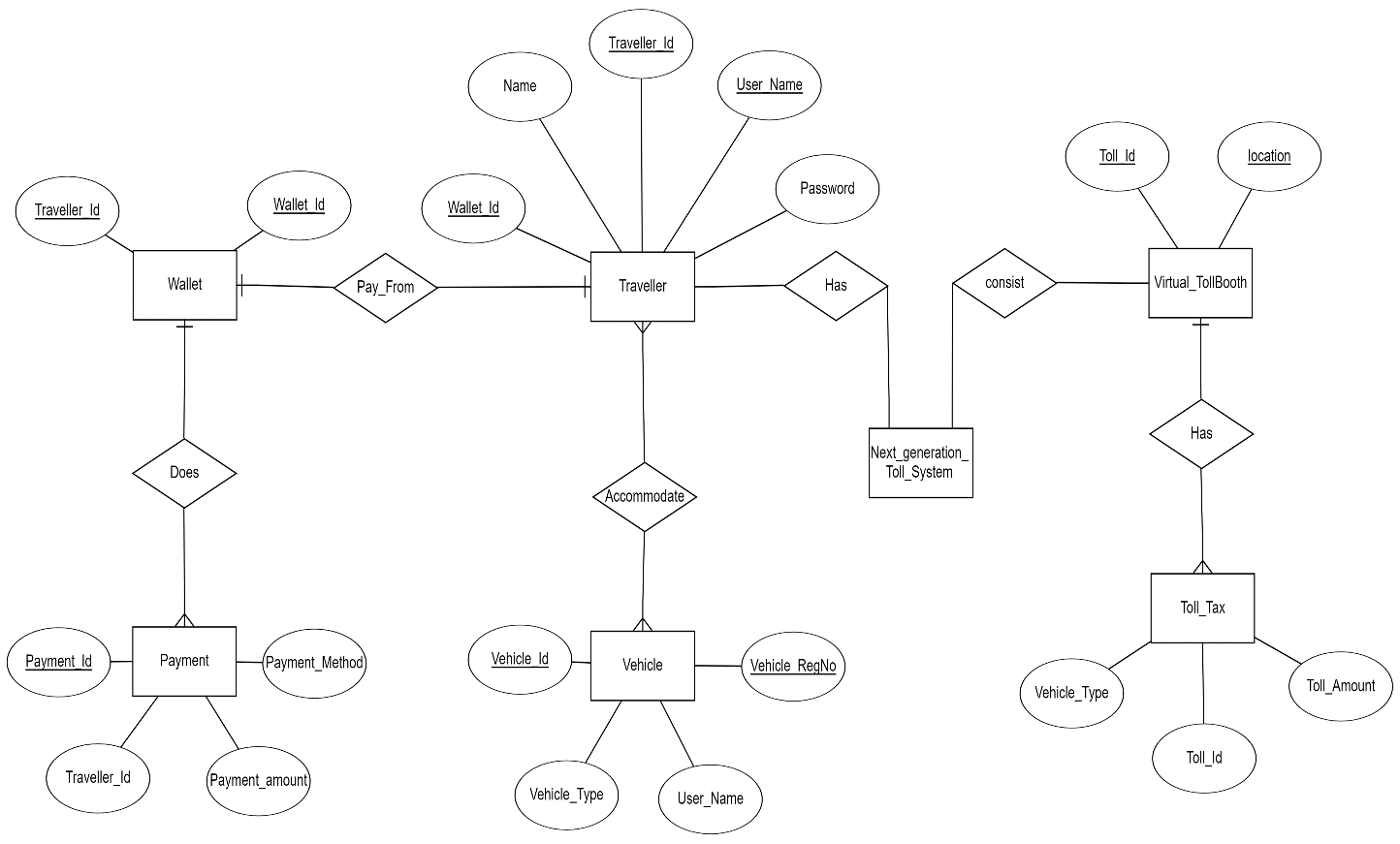
**Level 0 :**



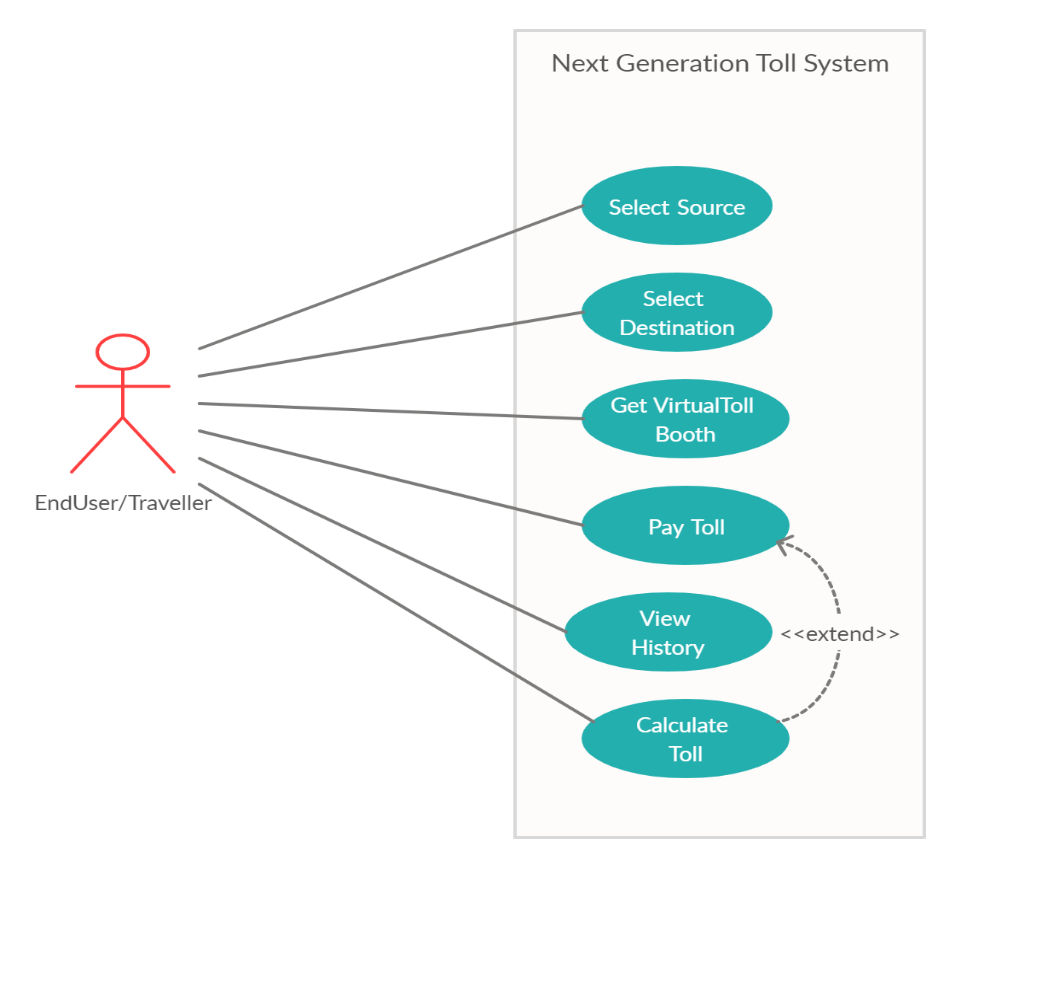
**Level 1 :**



**ER Diagram :**

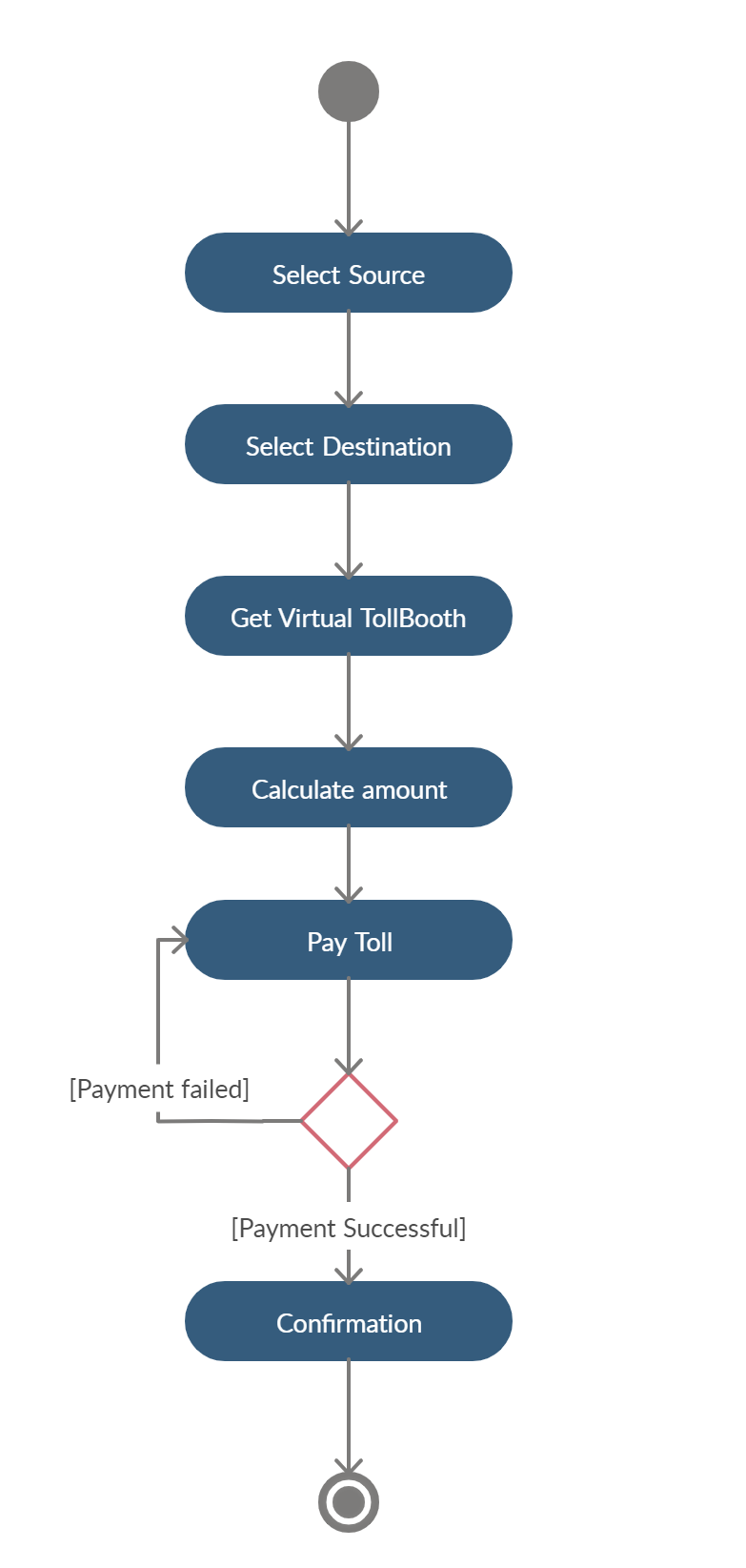


**Use case Diagram :**

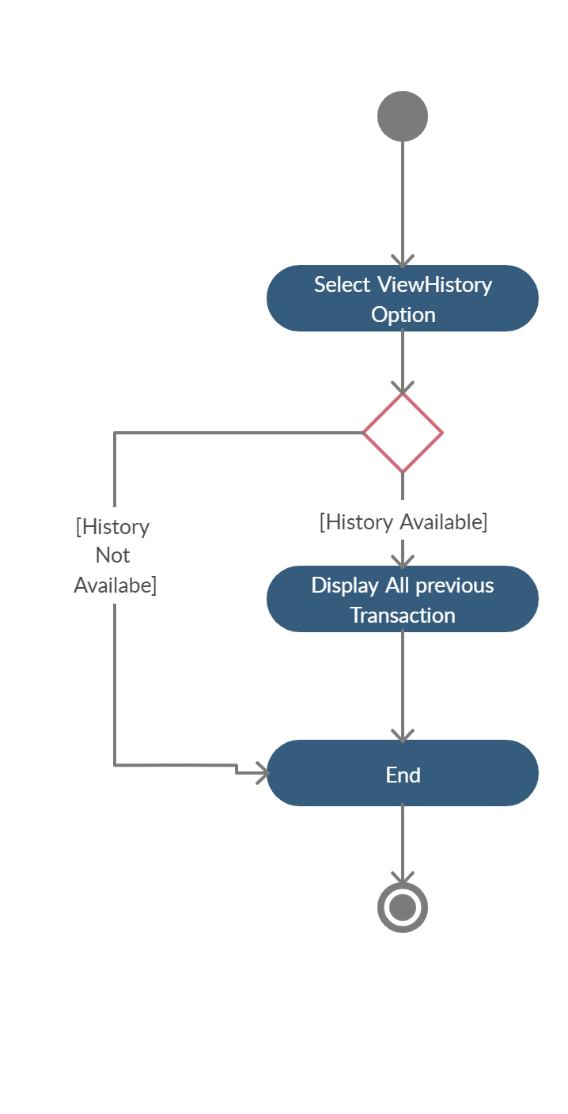


**Activity Diagram :**

For Toll Pay :

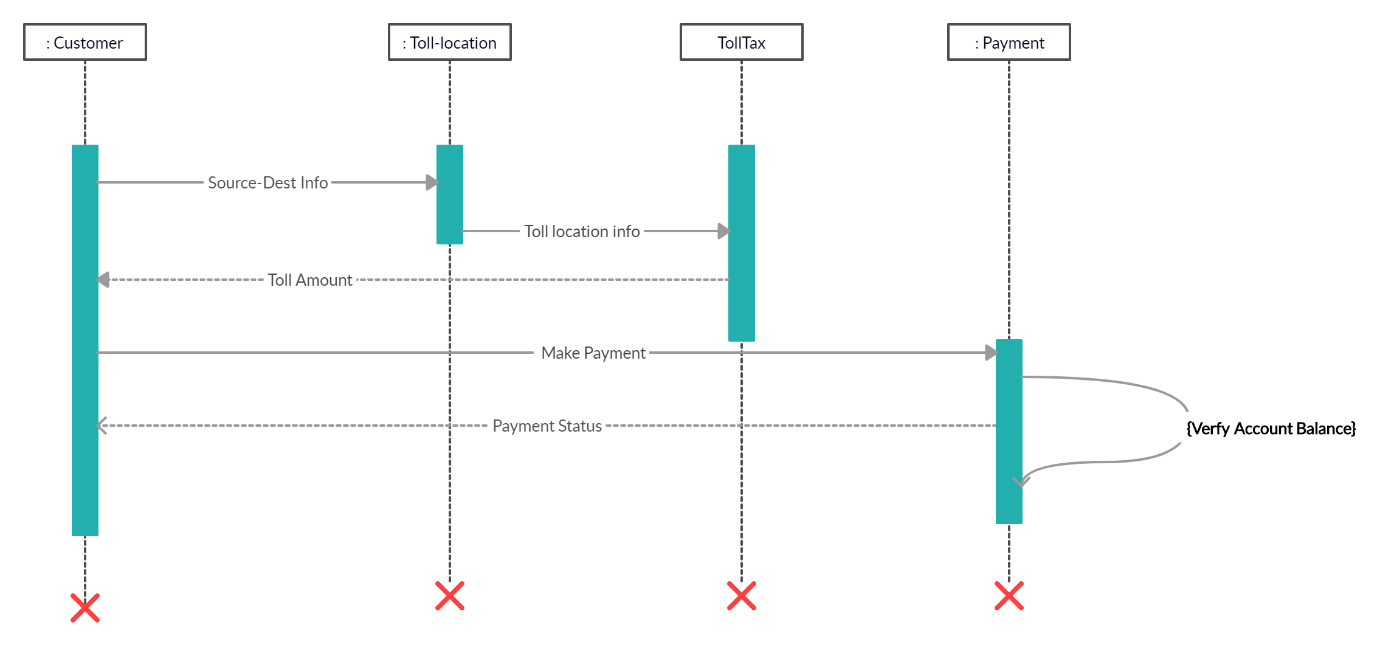


For View History :



**Sequence Diagram :**

For Make Payment :



8. IMPLEMENTATION DETAIL

I). Modules

* Main User Module :

This module contains the main homepage of the system which provides basic information about the website. It contains different functionalities like view history, pay toll etc. This module does require authentication.

* Login Module :

This module takes user credentials and then verifies it with registered users, if user have entered incorrect credentials then alert with “Incorrect details” , otherwise user will be redirected to the home page.

* Registration Module :

This module is used to create the account on system. All fields are containing some types of validations. Then user can login to our system using his/her credentials.

* Map route Module :

This module allows driver to set source and destination , it helps to display tollbooth between two points and based on car type and finds total toll amount of journey. It has been implemented by use of Maps SDK for Android and geo-location API.

* Payment Module :

This module is build for pay the toll by using Razorpay API. Razorpay is a payment gateway service. It accept and validate internet payments via credit card, debit card, Net banking modes, also other wallets like Paytm , PhonePe, fastcharge.

* Previous transaction Module :

This module is displaying all previous transaction for user of the system. It display all information regarding previous journey.

II). Function prototypes

* Login :

Future signInwithemailandpassword(String email,String password)

* Register :

Future registerwithemailandpassword(String email,String password)

* Load google map :

void \_onMapCreated(GoogleMapController controller)

* Set source & destination :

searchandnavigatesource()

searchandnavigatesource()

* Mark tollbooth :

tollmark(toll,tolllength)

* Payment handle methods :

void \_handlePaymentSuccess(PaymentSuccessResponse response)

void \_handlePaymentError(PaymentFailureResponse response)

void \_handleExternalwallet(ExternalWalletResponse response)

9. TESTING

In this system we have used Black Box testing.

The main focus of black box testing is on the validation of your functional requirements.

Here are the generic steps followed to carry out any type of Black Box Testing.

* Initially, the requirements and specifications of the system are examined.
* Tester chooses valid inputs (positive test scenario) to check whether SUT processes them correctly. Also, some invalid inputs (negative test scenario) are chosen to verify that the SUT is able to detect them.
* Tester determines expected outputs for all those inputs.
* Software tester constructs test cases with the selected inputs.
* The test cases are executed.
* Software tester compares the actual outputs with the expected outputs.
* Defects if any are fixed and re-tested.

For Register & Login :

If user doesn’t entered valid information then system not allowed for register in the app. It displays error message “INVALID USERNAME OR PASSWORD”.

If user doesn’t provide valid details at login time system displays error message like “Enter Valid Email” , “Enter a Password 6+ chars long.”