**Project Form 3: Development Test Cases**

Create test cases for your project using TDD.

You should think about unit testing, component testing and system testing.

Include the following:

* Test Type – unit testing/component testing/system testing
* Test scenario
* Test steps
* Test data
* Expected Results
* Comments

|  |  |
| --- | --- |
| **Test Type** | Unit Testing |
| **Test Objective** | Invalid Input In search field |
| **Test scenario** | Test the input from the user in search bar. |
| **Test Steps** | 1. User opens the app. 2. user taps on search option. 3. types the desired search input |
| **Test Data** | This will be testing the input search data. User inputs the search option and this test will validate the input. Search option can be based on suburbs, bus stations and places which won’t accept any special character and null value. |
| **Expected Results** | Search result based on user option should be displayed. |
| **Remarks (pass/fail)** | If pass, search result should be displayed and if user inputs incorrect characters, they will be prompt for incorrect error alert. |
| **Comments** |  |

|  |  |
| --- | --- |
| **Test Type** | Unit Testing |
| **Test Objective** | Mandatory field testing |
| **Test scenario** | This test is intended to test the mandatory field for any inputs in the app. Mandatory fields can be input option in search field, starting and destination place, time field in journey planner. |
| **Test Steps** | 1. Look for any input user has given in different field. 2. Check the field has null value |
| **Test Data** | Null value will be checked in any input from the user. |
| **Expected Results** | User filling out all the mandatory field. |
| **Remarks (pass/fail)** | If pass, this test will prompt user for furthur steps. If failed, user will be asked to input again saying this field should not be left empty. |
| **Comments** |  |

|  |  |
| --- | --- |
| **Test Type** | Unit Testing |
| **Test Objective** | Check if layout is consistent with design criteria. |
| **Test scenario** | This test will check for the usability and accessibility of the app layout in context to design criteria. Correct color combination and color grading will be checked during this test to make sure its accessible by all type of user. |
| **Test Steps** | 1. Look for color combination of all app layout from text to background color grading. 2. Look for brand design and app color palette. |
| **Test Data** | Test the color grading of the app. |
| **Expected Results** | Usable app, accessible to every users. |
| **Remarks (pass/fail)** | pass |
| **Comments** |  |

|  |  |
| --- | --- |
| **Test Type** | Unit Testing |
| **Test Objective** | Meaning Error Message |
| **Test scenario** | Any error in app should not reveal the system information instead it should prompt an meaningful error message to user in layman term. |
| **Test Steps** | 1. Display any error. 2. Read the error message. |
| **Test Data** | This test will look into error message that has been provided to user when user encounter any errors. |
| **Expected Results** | Meaningful message for any kind of error without revealing any system information. |
| **Remarks (pass/fail)** | If pass, users will not see any information regarding system error, instead they will be seeing error alert with simple language. |
| **Comments** |  |

|  |  |
| --- | --- |
| **Test Type** | Unit Testing |
| **Test Objective** | All the credentials should be transferred from an encrypted channel. |
| **Test scenario** | This tests the credentials being encrypted and been securely accessed in the database. |
| **Test Steps** | 1. Look for hashing of password. 2. check if database has password in plain text stored. 3. Check any secured socket layer is been used to access the database. |
| **Test Data** | Password and other credentials are tested. |
| **Expected Results** | Use of secure route and password hashing for storage. |
| **Remarks (pass/fail)** | If pass this should provide an extra layer of security to the app. |
| **Comments** |  |

|  |  |
| --- | --- |
| **Test Type** | Component Testing |
| **Test Objective** | Create account |
| **Test scenario** | test Registration form to use the app |
| **Test Steps** | 1. if do not have an account, click create account 2. fill up the registration form 3. submit |
| **Test Data** | name, mail address, mobile number, current address |
| **Expected Results** | creation of an account |
| **Remarks (pass/fail)** | pass |
| **Comments** |  |

|  |  |
| --- | --- |
| **Test Type** | Component Testing |
| **Test Objective** | Successful login |
| **Test scenario** | Test Login to use main features of the app |
| **Test Steps** | 1. register 2. log in 3. enter the main user interface |
| **Test Data** | user name and user password |
| **Expected Results** | successful log in |
| **Remarks (pass/fail)** | pass |
| **Comments** |  |

|  |  |
| --- | --- |
| **Test Type** | Component Testing |
| **Test Objective** | To find out functional route search bar |
| **Test scenario** | Route Search Bar |
| **Test Steps** | 1. Login 2. Search Destination |
| **Test Data** | Destination name |
| **Expected Results** | all available bus routes |
| **Remarks (pass/fail)** | pass |
| **Comments** |  |

|  |  |
| --- | --- |
| **Test Type** | Component Testing |
| **Test Objective** | Make sure calendar is working |
| **Test scenario** | Select date to book ticket |
| **Test Steps** | 1. select departure date |
| **Test Data** | date |
| **Expected Results** | successfully select date |
| **Remarks (pass/fail)** | pass |
| **Comments** |  |

|  |  |
| --- | --- |
| **Test Type** | System Testing |
| **Test Objective** | Test Ticket System |
| **Test scenario** | Purchasing and scanning tickets using the app |
| **Test Steps** | 1. Purchase ticket using app  2. Scan app to verify ticket |
| **Test Data** | Test ticket to verify scanning capability |
| **Expected Results** | The user will be able to purchase a bus ticket using the app, and be able to scan their phone to verify it |
| **Remarks (pass/fail)** | If test fails, revisit components to ensure they are working together properly. Check databases and component code. |
| **Comments** |  |

|  |  |
| --- | --- |
| **Test Type** | System Testing |
| **Test Objective** | Test Bus Scheduling System |
| **Test scenario** | Testing that bus times and routes are accurate |
| **Test Steps** | 1. Use the app to check current location  2. Provide estimated arrival time of next bus  3. Ensure arrival times are accurate based on timetable data |
| **Test Data** | Location of user (close to a bus stop), map, timetables |
| **Expected Results** | The app will provide an estimated arrival time of the next bus based on the location of the user |
| **Remarks (pass/fail)** | If test fails, check location data and timetable data and ensure they correspond accordingly. |
| **Comments** |  |

|  |  |
| --- | --- |
| **Test Type** | System Testing |
| **Test Objective** | Test Account System |
| **Test scenario** | Registering and logging into an account in the app |
| **Test Steps** | 1. Create an account  2. Login to the account  3. Log out of the account |
| **Test Data** | Placeholder user data to create an account with |
| **Expected Results** | The user will be able to create an account, login, and logout |
| **Remarks (pass/fail)** | If test fails, revisit account code and check for errors. Ensure databases are being queried correctly. |
| **Comments** |  |

|  |  |
| --- | --- |
| **Test Type** | System Testing |
| **Test Objective** | Test saving trips in the app |
| **Test scenario** | Save a history of recent trips, allow the user to save trips as favourites |
| **Test Steps** | 1. Access trip history  2. Save a trip to “favourites” |
| **Test Data** | Placeholder trip history for testing |
| **Expected Results** | Recent trip history can be saved to an account, and the user will be able to save trips to a favourites list |
| **Remarks (pass/fail)** | If test fails, check for errors in database code. Ensure that account data is saved correctly. |
| **Comments** |  |

**Paper and Digital Prototype: Joel**