# Role - TESTER

The Testers are the most important link between the technical vision of the TL and the market-requirement of the BA. They work closely with BAs to develop User Acceptance criteria and user-stories based test plan which ensure that quality is build into the process and thus the product. They enforce software quality and ensure the product is as bug-free and market-friendly as possible and document the test results

# Revision History

| Week Number | Author | Description of changes |
| --- | --- | --- |
| 1 - ? | Eden & Matt | Added possible testing criteria and layout to this file |
|  |  |  |
|  |  |  |
|  |  |  |

*Comments:*

* *This page should be updated each week and the TAs will use it to determine the week’s work done by the team-member in the particular role.*
* *For each of the items mentioned in the following pages. Be as brief as possible in your responses. A good rule of thumb is to keep each response within two paragraphs.*

*- This document represents your log of decisions. You are not bound to follow a decision blindly. You may change the decision if new aspects come to light which make your decision inappropriate. However, this may include repercussions like code rewrite etc. So choose wisely.*

*- You may delete the commented regions for your weekly turn-in submissions.*

1. Testing Strategy
   1. Test Objectives

All tests will be well conducted in each module before performing a test with a combination of modules to ensure that errors are taken care of as early as possible.

* 1. Scope and Level of Testing

Unit testing will be used in each individual module, integration test will be used for combined modules, and regression test will be used each time a new change is made.Lastly, a validation test will be needed to ensure that the overall functionality of the app is good for publish.

* 1. Functional Testing

Since we are building an Android Application, JUnit test is a preferable testing environment.

* 1. Future Enhancements

Adding more details in all tables as we consult with our Business Analyst and Project Manager and as each module of the project is developed.

* 1. Known Issues

|  |  |  |
| --- | --- | --- |
| Known Issues | Causes | Solutions |
| Incorrect Login | Page does not exist | Create the page. |
| Incorrect Transactions | There is no transactions | Verify transactions with database |
| Missing backend | No server | Create Server |
|  |  |  |
|  |  |  |
|  |  |  |

1. User Acceptance Scenarios

|  |  |  |
| --- | --- | --- |
| BA Crteria | PM Criteria | Overall Criteria |
| User should be able to securely log in as a bank employee or account holder | Connection should be secure from login to logout | Create login page |
| User should be able to log out and return to the log in screen | User must be able to login and logout | Create logout button |
| New customers should be able to open their first bank/user account with name, address, date of birth, email and account type. | Must be able to create accounts | Create server to handle accounts |
| Teller should be able to look up a customer’s account to perform transactions for them. | Transaction lookup shuld exist | Server handles lookups |
| Account holder/teller should be able to add a new bank account. | New account should appear in database | Create adding feature for new account |
| User should be able to retrieve password by submitting their email and account number | Account details should exist in database for retrieval | Create backend with password retrieval |
| Account holder should be able to see all of their accounts | Account should appear in users list | Create listing feature |
| Account holder/teller should be able to disable and delete one of the customer’s bank accounts. | Delete account but still exist on server for history purposes | Handle deletion feature |
| Notify user that account must have zero balance to be closed. | Create criteria to be checked before deletion | Handle zero balance to close case |
| Account holder/teller should be able to withdraw money from a customer’s bank account IF withdrawal amount does not exceed the account balance. | Create criteria to be checked before withdrawals | Grant administration power to teller account |
| Check is emailed to account holder once they debit their account. | Backend to handle check creation | Email check to user with a generator |
| Notify user that they can’t perform this action due to insufficient funds in source account | Create criteria to check funds | Criteria to check funds |
| Account holder/teller should be able to deposit money into one of customer’s bank account. Account holder can do this through credit and check. Teller can do this through cash as well. | Teller account mode has administrative power to do more actions on user account. | Create account with more administrative functions |
| Account holder should be able to take a picture of a check to credit their account. | Camera is used to take the picture and Teller account mode handles the deposit. | Camera handles check |
| Account holder should be able to transfer money from one of their bank accounts to another. | Transfer funds by account number | Verify transations |
| User must confirm this transaction | Second request to verify transaction | Verify transations |
| Account holder should be able to view interests of each of their bank accounts. | Display transaction history | Calculate interest and display it in account balance |
| Balances of accounts should be listed to Account holder. | Create a page that displays accounts | Create account balance in account page |
| Account holder should be able to view their transaction history | A button is created that shows a log with the transaction history | Feature is listed to view transaction history |
| Account holder should be able to retrieve and print out their transaction history | Print function will handle export of transaction history. | Print button function in transaction history |

*Comment: there should be an acceptance criteria for each user story developed by the BA or PM. Remember that you are looking for a balance here, enough to ensure a glitch-free experience for the end-user and relaxed enough to ensure productive and quick code-iteration cycles.*

1. Testing Risks

| S. No | Risk | Probability | Impact | Mitigation Plan |
| --- | --- | --- | --- | --- |
| 1. | Security |  |  | Create a secure connection. |
| 2 |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. Test Results:
   1. Document the results of the tests you have done, along with severity (high, medium or low) — No Test has been done at this time.

|  |  |  |
| --- | --- | --- |
| User-story/ Acceptance Criteria | Results | Severity |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |