**TEST PLAN**

**VERSION HISTORY**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version # | Implemented  By | Revision  Date | Approved  By | Approval  Date | Reason |
| 1.0 | Promolac team | 06/06/20 | Mr. Farrukh Hasan | 07/06/20 | Test Plan draft |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Contents

[1. INTRODUCTION 3](#_Toc42385345)

[1.1. Purpose 3](#_Toc42385346)

[2. ENVIRONMENTAL NEEDS 3](#_Toc42385347)

[3. TEST STRATEGY 4](#_Toc42385348)

[3.1. Test Objectives 4](#_Toc42385349)

[3.2. Test Assumptions 4](#_Toc42385350)

[3.3. Test Principles 5](#_Toc42385351)

[3.4. Data Approach 5](#_Toc42385352)

[3.5. Scope and Levels of Testing 5](#_Toc42385353)

[3.5.1. Exploratory 5](#_Toc42385354)

[3.5.2. Functional Test 6](#_Toc42385355)

[3.5.3. User Acceptance Test (UAT) 7](#_Toc42385356)

[4. EXECUTION STRATEGY 7](#_Toc42385357)

[4.1. Entry and Exit Criteria 7](#_Toc42385358)

[4.2. Test Cycles 8](#_Toc42385359)

[4.3. Validation and Defect Management 9](#_Toc42385360)

[4.4. Test Metrics 10](#_Toc42385361)

[4.5. Defect tracking & Reporting 10](#_Toc42385362)

[5. TEST MANAGEMENT PROCESS 11](#_Toc42385363)

[5.1. Test Design Process 11](#_Toc42385364)

[5.2. Test Execution Process 12](#_Toc42385365)

[5.3. Test Risks and Mitigation Factors 13](#_Toc42385366)

[5.1. Communications Plan and Team Roster 14](#_Toc42385367)

[5.2. Role Expectations 14](#_Toc42385368)

[5.2.1. Project Supervisor 14](#_Toc42385369)

[5.2.2. Test Planning (Test Lead) 14](#_Toc42385370)

[5.2.3. Test Team 15](#_Toc42385371)

[5.2.4. Test Lead 15](#_Toc42385372)

[5.2.5. Development Team 15](#_Toc42385373)

[6. TEST ENVIRONMENT 16](#_Toc42385374)

[7. TEST CASES 16](#_Toc42385375)

# INTRODUCTION

## Purpose

The purpose of the testing is to analyse the system errors and problems generated in user experience. Management of web application and android application is optimized by testing. UI defects in Human Computer Interactions and unusual exceptions are mainly the concern. The bugs fixing in privacy and security is crucial for every user to maintain their personal information.

# ENVIRONMENTAL NEEDS

* Active internet connection to connect Android Application with Firebase and Web Application with SQL server
* Some External Android Devices with location enabled with permission access granted.
* Data will be usually generated for testing by manually user input on Android app as well as Web
* Each module will be tested on every sort of data, and exceptions are properly handled at each point.
* Testing with every aspect will be done to make sure app don’t crash
* No special Power requirements are required as Promolac is a simple system consisting of Android and Web Application.

# TEST STRATEGY

## Test Objectives

The objective of the test is to verify that the functionality of Promolac works according to the specifications.

The test will execute and verify the test scripts, identify, fix and retest all high and medium severity defects per the entrance criteria, prioritize lower severity defects for future fixing via CR.

The final product of the test is twofold:

* A production-ready software
* A set of stable test scripts that can be reused for Functional and validation test execution.

## Test Assumptions

**Key Assumptions**

* The data required will be available in the system prior to start of Functional Testing
* In each testing phase, Cycle 3 will be initiated if the defect rate is high in Cycle 2.

**General**

* Exploratory Testing would be carried out once the build is ready for testing
* Performance testing is not considered for this estimation.
* All the defects would come along with a snapshot JPEG format.
* Test case design activities will be performed by Team Members.
* Test environment and preparation activities will be owned by Team.
* Team will provide Defect fix plans based on the Defect meetings during each cycle to plan. The same will be informed to Test team prior to start of Defect fix cycles.
* Supervisors will review and sign-off all Test cases prepared by Test Team prior to start of Test execution
* The defects will be tracked. Any defect fixes planned will be shared with Test Team prior to applying the fixes on the Test environment.
* Project Supervisor will review and sign-off all test deliverables.
* The project will provide test planning, test design and test execution support.
* Test team will manage the testing effort with close coordination with Project Supervisor.
* Project team has the knowledge and experience necessary, or has received adequate training in the system, the project and the testing processes.
* There is no environment downtime during test due to outages or defect fixes.
* The system will be treated as a black box; if the information shows correctly online and, in the reports, it will be assumed that the database is working properly.

**Functional Testing**

* During Functional testing, testing team will use preloaded data which is available on the system at the time of execution
* The Test Team will be performing Functional testing on Android and Web Application

**UAT**

* UAT test execution will be performed by end users and Group will provide their support on creating UAT script.

## Test Principles

* Testing will be focused on meeting the business objectives, cost efficiency, and quality.
* There will be common, consistent procedures for all teams supporting testing activities.
* Testing processes will be well defined, yet flexible, with the ability to change as needed.
* Testing activities will build upon previous stages to avoid redundancy or duplication of effort.
* Testing will be a repeatable, quantifiable, and measurable activity.
* Testing will be divided into distinct phases, each with clearly defined objectives and goals.
* There will be entrance and exit criteria.

## Data Approach

* In functional testing, Promolac will contain pre-loaded test data and which is used for testing activities.

## Scope and Levels of Testing

### Exploratory

**PURPOSE**: the purpose of this test is to make sure critical defects are removed before the next levels of testing can start.

**SCOPE**: First level navigation, simple communication and every module working according to required condition.

**TESTERS**: Group Members

**METHOD**: this exploratory testing is carried out in the application without any test scripts and documentation

**TIMING**: at the end of each cycle.

### Functional Test

**PURPOSE:**  Functional testing will be performed to check the functions of application. The functional testing is carried out by feeding the input and validates the output from the application.

**Scope:** The working of the ever module should be as per required in the documentation. Note: The scope is high level due to changes in the requirement.

**TESTERS**: Team Members

**METHOD**: The test will be performed according to Functional scripts and analyzing multiple aspect of the module.

**TIMING**: after Exploratory test is completed.

#### TEST ACCEPTANCE CRITERIA

1. Approved Functional Specification document, Use case documents must be available prior to start of Test design phase.
2. Test cases approved and signed-off prior to start of Test execution
3. Development completed, unit tested with pass status and results shared to Testing team to avoid duplicate defects
4. Test environment with application installed, configured and ready to use state

#### TEST DELIVERABLES

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Deliverable Name** | **Author** | **Reviewer** |
| 1. | Test Plan | Group Members | Supervisor/Jury |
| 2. | Functional Test | Group Members | Supervisor and team leader |
| 3. | Unit Testing | Group Members | Supervisor |
| 4. | System testing | Group Members | Supervisor |
| 5. | Test Closure report | Group Members | Supervisor |

#### MILESTONE LIST

The milestone list is tentative and may change due to below reasons

1. Any issues in the System environment readiness
2. Any change in scope/addition in scope
3. Any other dependency that impacts efforts and timelines

### User Acceptance Test (UAT)

**PURPOSE**: this test focuses on validating the business logic. It allows the end users to complete one final review of the system prior to deployment.

**TESTERS**: Normal people along with the Special People.

**METHOD**: Since the Normal and special users are the most indicated to provide input around business needs and how the system adapts to them, it may happen that the users do some validation not contained in the scripts. Test team write the UAT test cases based on the inputs from End user.

**TIMING**: After all other levels of testing (Exploratory and Functional) are done. Only after this test is completed the product can be released to production.

# EXECUTION STRATEGY

## Entry and Exit Criteria

* The entry criteria refer to the desirable conditions in order to start test execution; only the migration of the code and fixes need to be assessed at the end of each cycle.
* The exit criteria are the desirable conditions that need to be met in order proceed with the implementation.
* Entry and exit criteria are flexible benchmarks. If they are not met, the test team will assess the risk, identify mitigation actions and provide a recommendation. All this is input to the project supervisor for a final “go-no go” decision.
* Entry criteria to start the execution phase of the test: the activities listed in the Test Planning section of the schedule are 100% completed.
* Entry criteria to start each cycle: the activities listed in the Test Execution section of the schedule are 100% completed at each cycle.

|  |  |  |  |
| --- | --- | --- | --- |
| **Exit Criteria** | **Test Team** | **Technical Team** | **Notes** |
| 100% Test Scripts executed |  |  |  |
| 95% pass rate of Test Scripts |  |  |  |
| No open Critical and High severity defects |  |  |  |
| 95% of Medium severity defects have been closed |  |  |  |
| All remaining defects are either cancelled or documented as Change Requests for a future release |  |  |  |
| All expected and actual results are captured and documented with the test script |  |  |  |
| All test metrics collected based on reports. |  |  |  |
| All defects logged in files. |  |  |  |
| Test Closure Memo completed and signed off |  |  |  |
| Test environment cleanup completed and a new back up of the environment |  |  |  |



## Test Cycles

* + There will be two cycles for functional testing. The first cycle includes the functionality of the messaging and the second cycle includes the Leap motion conversion of Sign language.
  + The objective of the first cycle is to identify any blocking, critical defects, and most of the high defects in the message passing from one side to another. It is expected to use some work-around in order to get to all the functionality working.
  + The objective of the second cycle is to identify leap motion sensor working accordingly and analyze its remaining high and medium defects also remove the work-around from the first cycle, correct gaps in the functionality and obtain performance results.
  + UAT test will also consist of one cycle.

## Validation and Defect Management

* It is expected that the testers execute all the scripts in each of the cycles described above. However, it is recognized that the testers could also do additional testing if they identify a possible gap in the scripts. This is especially relevant in the second cycle, when the leap motion module will be testing for their accuracy with having a deeper knowledge of the Sign Language. If a gap is identified, the functionality and the data set will be updated and then a defect logged against the scripts.
* It is the responsibility of the tester to open the defects, link them to the corresponding script, assign an initial severity and status, retest and close the defect; it is the responsibility of the Tester to review the severity of the defects and facilitate with the development team the fix and its implementation, communicate with testers when the test can continue or should be halt, request the tester to retest, and modify status as the defect progresses through the cycle; it is the responsibility of the technical team to review, fix the defect, communicate to the Defect Manager the fix is done, implement the solution per the Defect Manager request.

Defects found during the Testing will be categorized according to the bug-reporting tools and the categories are:

|  |  |
| --- | --- |
| **Severity** | **Impact** |
| 1 (Critical) | * This bug is critical enough to crash the system, cause file corruption, or cause potential data loss. * It causes an abnormal return to the operating system (crash or a system failure message appears). * It causes the application to hang and requires re-booting the system. |
| 2 (High) | * It causes a lack of vital program functionality with workaround. |
| 3 (Medium) | * This Bug will degrade the quality of the System Reducing the accuracy of the system. However there is an intelligent workaround for achieving the desired functionality. * This bug prevents other areas of the product from being tested. However other areas can be independently tested. |
| 4 (Low) | * There is an insufficient or unclear error message, which has minimum impact on product use. |
| 5(Cosmetic) | * There is an insufficient or unclear error message that has no impact on product use. |

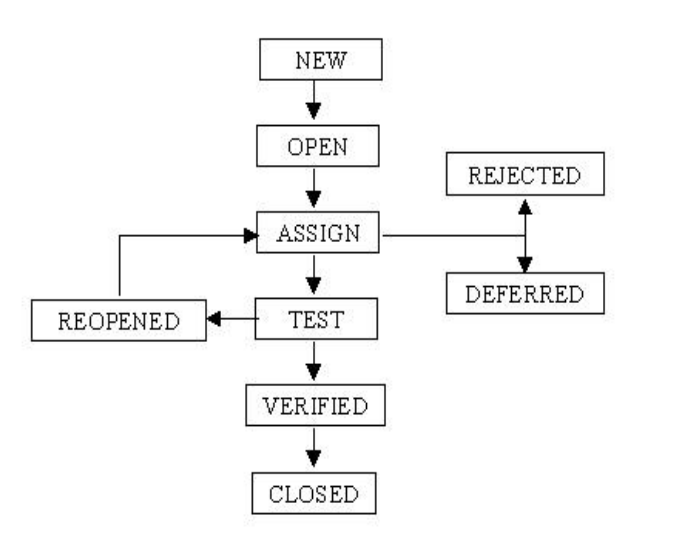
## Test Metrics

Test metrics to measure the progress and level of success of the test will be developed and shared with the project manager for approval. The below are some of the metrics

|  |  |  |
| --- | --- | --- |
| **Report** | **Description** | **Frequency** |
| Test preparation & Execution Status | To report on % complete, %WIP, % Pass, % Fail  Defects severity wise Status – Open, closed, any other Status | Weekly / Daily (optional) |
| Daily execution  status | To report on Pass, Fail, Total defects, highlight Showstopper/ Critical defects | Daily |
| Project Weekly Status report | Project driven reporting (As requested by PM) | Weekly – If project team needs weekly update apart from daily and there is template available with project team to use. |

## Defect tracking & Reporting

Following flowchart depicts Defect Tracking Process:



# TEST MANAGEMENT PROCESS

## Test Design Process

* The tester will understand each requirement and prepare corresponding test case to ensure all requirements are covered.
* Each Test case will be mapped to Use cases to Requirements as part of Traceability matrix.
* Each of the Test cases will undergo review and the review defects are captured and shared to the Test team. The testers will rework on the review defects and finally obtain approval and sign-off.
* During the preparation phase, tester will use the prototype, use case and functional specification to write step by step test cases.
* Testers will maintain a clarification Tracker sheet and same will be shared periodically with the Requirements team and accordingly the test case will be updated. The clarifications may sometimes lead to Change Requests or not in scope or detailing implicit requirements.
* Sign-off for the test cases would be communicates through mail by Business Analyst’s.
* Any subsequent changes to the test case if any will be directly updated.

## Test Execution Process

* Once all Test cases are approved and the test environment is ready for testing, tester will start an exploratory test of the application to ensure the application is stable for testing.
* Each Tester is assigned Test cases directly.
* Testers to ensure necessary access to the testing environment and updating test status and raise defects. If any issues, will be escalated to the Test Lead and in turn to the Project Manager as escalation.
* If any showstopper during exploratory testing will be escalated to the respective development SPOCs for fixes.
* Each tester performs step by step execution and updates the executions status. The tester enters Pass or Fail Status for each of the step directly.
* Tester will prepare a Run chart with day-wise execution details
* If any failures, defect will be raised as per severity guidelines detailing steps to simulate along with screenshots if appropriate.
* Testing team will participate in defect triage meetings in order to ensure all test cases are executed with either pass/fail category.
* This process is repeated until all test cases are executed fully with Pass/Fail status.

As per Process, final sign-off or project completion process will be followed

## Test Risks and Mitigation Factors

| Risk | Prob. | Impact | Mitigation Plan |
| --- | --- | --- | --- |
| **SCHEDULE**  Testing schedule is tight. If the start of the testing is delayed due to design tasks, the test cannot be extended beyond the UAT scheduled start date. | High | High | * The testing team can control the preparation tasks (in advance) and the early communication with involved parties. * Some buffer has been added to the schedule for contingencies, although not as much as best practices advise. |
| **RESOURCES**  Not enough resources, resources on boarding too late (process takes around 6 months | High | High | Holidays and vacation have been estimated and built into the schedule; deviations from the estimation could derive in delays in the testing. |
| **DEFECTS**  Defects are found at a late stage of the cycle or at a late cycle; defects discovered late are most likely be due to unclear specifications and are time consuming to resolve. | Medium | High | Defect management plan is in place to ensure prompt communication and fixing of issues. |
| **SCOPE**  Scope completely defined | Low | Medium | Scope is well defined but the changes are in the functionality are not yet finalized or keep on changing. |
| Natural disasters | Low | Medium | Teams and responsibilities have been spread to two different geographic areas. In a catastrophic event in one of the areas, there will resources in the other areas needed to continue (although at a slower pace) the testing activities. |
| Non-availability of Independent Test environment and accessibility | Medium | High | Due to non-availability of the environment, the schedule gets impacted and will lead to delayed start of Test execution. |
| Delayed Testing Due to New Issues | High | High | During testing, there is a good chance that some “new” defects may be identified and may become an issue that will take time to resolve.  There are defects that can be raised during testing because of unclear document specification. These defects can yield to an issue that will need time to be resolved.  If these issues become showstoppers, it will greatly impact on the overall project schedule.  If new defects are discovered, the defect management and issue management procedures are in place to immediately provide a resolution. |

## Communications Plan and Team Roster

## Role Expectations

### Project Supervisor

* Project Supervisor: reviews the content of the Test Plan, Test Strategy and Test Estimates signs off on it.

### Test Planning (Test Lead)

* Ensure entrance criteria are used as input before start the execution.
* Develop test plan and the guidelines to create test conditions, test cases, expected results and execution scripts.
* Provide guidelines on how to manage defects.
* Attend status meetings in person or via the conference call line.
* Communicate to the test team any changes that need to be made to the test deliverables or application and when they will be completed.
* Provide on premise or telecommute support.
* Provide functional and technical team to test team personnel (if needed).

### Test Team

* Develop test conditions, test cases, expected results, and execution scripts.
* Perform execution and validation.
* Identify, document and prioritize defects according to the guidance provided by the Test lead.
* Re-test after software modifications have been made according to the schedule.
* Prepare testing metrics and provide regular status.

### Test Lead

* Acknowledge the completion of a section within a cycle.
* Give the OK to start next level of testing.
* Facilitate defect communications between testing team and technical / development team.

### Development Team

* Review testing deliverables (test plan, cases, scripts, expected results, etc.) and provide timely feedback.
* Assist in the validation of results (if requested).
* Support the development and testing processes being used to support the project.
* Certify correct components have been delivered to the test environment at the points specified in the testing schedule.
* Keep project team and leadership informed of potential software delivery date slips based on the current schedule.
* Define processes/tools to facilitate the initial and ongoing migration of components.
* Conduct first line investigation into execution discrepancies and assist test executors in creation of accurate defects.
* Implement fixes to defects according to schedule.

# TEST ENVIRONMENT

Promolac App will be hosted at Firebase.

Promolac Web Portal will be hosted on online server.

Promolac will be hosted on two servers: One to host the actual website and (language) code, and the other will be connected to it.

A windows environment with and with Firefox 27.0 or later, Internet Explorer 8, 9 and 10, as well as Google Chrome 32.0 and later should be available to each tester

The JDK and SDK 3.0 or later should be installed. The tester hand should have internet connectivity and location service enabled. The tester should have android device having android 5.0 and above. Tester should have Visual Studio download on his PC or Laptop. Tester should have SQL Server Management Studio and Firebase.

# TEST CASES

**User Test Case:**

**Test 1:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TEST CASE ID: PRL001**  **DESCRIPTION: User Login Check (Android Application)** | | | | | |
| **No.** | **STEPS** | **EXPECTED RESULTS** | **ACTUAL RESULTS** | **PASS/ FAIL** |
|  | Correct id and password | Successful Login | Successfully login | Pass |
|  | Correct id, Wrong password | Show Error (Wrong Id Exception) | Login denied with error message | Pass |
|  | Wrong id and password | Show Error (Wrong id and password Error) | Login denied with error message | Pass |
|  | Wrong id, Correct password | Show Error (Wrong password Error) | Login denied with error message | Pass |
|  | Empty Password or Username | Empty Field Prompt | App Crashed | Fail |
|  | Empty Password or Username (updated) | Empty Field Prompt | Empty Field Prompt | Pass |

**Test 2:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TEST CASE ID: PRL002**  **DESCRIPTION: User Signup Check (Android Application)** | | | | | |
| **No.** | **STEPS** | **EXPECTED RESULTS** | **ACTUAL RESULTS** | **PASS/ FAIL** |
|  | Register new account | Register Successful | Register Successfully | Pass |
|  | Register by using used Email | Show Error (Already Register Prompt) | Successfully Registered | Fail |
|  | Register by using used Email | Show Error (Already Register Prompt) | Login denied with error message | Pass |
|  | Register by using wrong email without domain address | Email Address is in wrong pattern | Nothing happened User Register Successfully | Fail |
|  | Register by using wrong email without domain address | Email Address is in wrong pattern | Nothing happened | Fail |
|  | Register by using wrong email without domain address | Email Address is in wrong pattern prompt | Email Address is in wrong pattern prompt | Pass |
|  | Empty Password or Name or Email | Empty Field Prompt | App Crashed | Fail |
|  | Empty Password or Name or Email | Empty Field Prompt | Empty Field Prompt | Pass |

**Test 3:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TEST CASE ID: PRL003**  **DESCRIPTION: Upload Profile (Android Application)** | | | | | |
| **No.** | **STEPS** | **EXPECTED RESULTS** | **ACTUAL RESULTS** | **PASS/ FAIL** |
|  | Upload Profile | Upload Successful | Upload Successfully | Pass |
|  | Checking Upload on Firebase | Upload Successfully | No Record Found | Fail |
|  | Checking Upload on Firebase (again) | Upload Successfully | Uploading Error in App | Fail |
|  | Checking Upload on Firebase (again) | Upload Successfully | App Crashed | Fail |
|  | Checking Upload on Firebase (again) | Upload Successfully | Upload Successfully | Pass |

**Test 4:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TEST CASE ID: PRL004**  **DESCRIPTION: Deal List View Test (Android Application)** | | | | | |
| **No.** | **STEPS** | **EXPECTED RESULTS** | **ACTUAL RESULTS** | **PASS/ FAIL** |
|  | Deals Added | Deals will be Show Correctly | Last Deal is not shown completely | Fail |
|  | Deals Added | Deals will be Show Correctly | First Deal is not shown completely | Fail |
|  | Deals Added | Deals will be Show Correctly | Deal Show Correctly | Pass |
|  | Deals Image Downloading | Download Successfully | App Crashed | Fail |
|  | Deals Image Downloading | Download Successfully | Downloaded Successfully | Pass |
|  | Checking Upload on Firebase | Upload Successfully | App Crashed | Fail |
|  | Checking Upload on Firebase | Upload Successfully | Upload Successfully | Pass |

**Test 5:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TEST CASE ID: PRL005**  **DESCRIPTION: Message List View Test (Android Application)** | | | | | |
| **No.** | **STEPS** | **EXPECTED RESULTS** | **ACTUAL RESULTS** | **PASS/ FAIL** |
|  | Messages Added | Messages will be Show Correctly | Last Message is not shown completely | Fail |
|  | Messages Added | Messages will be Show Correctly | First Message is not shown completely | Fail |
|  | Message Added | Message will be Show Correctly | Message Show Correctly | Pass |
|  | Message Downloading | Download Successfully | App Crashed | Fail |
|  | Message Downloading | Download Successfully | Downloaded Successfully | Pass |
|  | Checking Upload on Firebase | Upload Successfully | App Crashed | Fail |
|  | Checking Upload on Firebase | Upload Successfully | Upload Successfully | Pass |
|  | Checking save in file | Saved in File | Not saved in file | Fail |
|  | Checking save in file | Saved in File | Saved in File | Pass |
|  | Delete Message | Delete Successfully | Not Deleted | Fail |
|  | Delete Message | Delete Successfully | Not Deleted | Pass |

**Vendor/Admin Test Cases**

**Test Case 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST CASE ID:  PRLV001**  **DESCRIPTION: Vendor/Admin login on Web Portal** | | | | |
| **No.** | **STEPS** | **EXPECTED RESULTS** | **ACTUAL RESULTS** | **PASS/ FAIL** |
|  | Authentic Id and Password | Successful Login | Successfully Login | pass |
|  | Correct Id but wrong Password | Access Denied with Error message | Access Denied | pass |
|  | Correct password with wrong ID | Access Denied with Error message | Access Denied | pass |
|  | Both Id and password wrong | Access Denied with Error message | Access Denied | pass |
|  | Empty Id/Password | Field Required Prompt | Fields Required Prompt | pass |

**Test Case 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST CASE ID:  PRLV002**  **DESCRIPTION: Package and Trying to Access** | | | | |
| **No.** | **STEPS** | **EXPECTED RESULTS** | **ACTUAL RESULTS** | **PASS/ FAIL** |
|  | Package available | Successful Login | Successfully Login | pass |
|  | Package expired | Access Denied with Error message | Access Denied | pass |

Test Case 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST CASE ID:  PRLV003**  **DESCRIPTION: Deals Create** | | | | |
| **No.** | **STEPS** | **EXPECTED RESULTS** | **ACTUAL RESULTS** | **PASS/ FAIL** |
|  | Deal Create with Package available | Successful Create | Successfully Created | pass |
|  | Deal Create with Package bucket expired | Create Deny | Create Denied | pass |

Test Case 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST CASE ID:  PRLV004**  **DESCRIPTION: Geofence Create** | | | | |
| **No.** | **STEPS** | **EXPECTED RESULTS** | **ACTUAL RESULTS** | **PASS/ FAIL** |
|  | Fence Create with Package available | Successful Create | Successfully Created | pass |
|  | Fence Create with Package bucket expired | Create Deny | Create Denied | pass |

Test Case 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST CASE ID:  PRLV005**  **DESCRIPTION: Send Notification** | | | | |
| **No.** | **STEPS** | **EXPECTED RESULTS** | **ACTUAL RESULTS** | **PASS/ FAIL** |
|  | Send Notification with Package available | Successful Notification send | Successfully Sent | pass |
|  | Send Notification with Package bucket expired | Send Notification Deny | Send Notification Denied | pass |

Test Case 6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST CASE ID:  PRLV006**  **DESCRIPTION: Send Message** | | | | |
| **No.** | **STEPS** | **EXPECTED RESULTS** | **ACTUAL RESULTS** | **PASS/ FAIL** |
|  | Send Message with Package available | Successful Message send | Successfully Sent | pass |
|  | Send Message with Package bucket expired | Send Message Deny | Send Message Denied | pass |

Test Case 7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST CASE ID:  PRLV007**  **DESCRIPTION: Create Sub Vendor** | | | | |
| **No.** | **STEPS** | **EXPECTED RESULTS** | **ACTUAL RESULTS** | **PASS/ FAIL** |
|  | Create Sub Vendor limited by Package | Successful Create | Successfully Created | pass |
|  | Create Sub Vendor in Empty bucket | Create Deny | Denied | pass |

1. **LIMITATION OF SYSTEM**

For now, our system is limited to Karachi users. Internet connectivity for once is required along with location services kept turned on.

1. **FUTURE WORK**

In future we will implement location-based group chat in mobile application and two of our work Geo-Conquesting and Beacons-Technology. User experience will get major upgrade by applying Machine Learning to learn user pattern.

1. **CONCLUSION**

A software which can help companies to get marketing services of their product’s in real life. Our goal is to make something for them that can make their life easy. Our aim is to target that kind of people and contribute in society with our product. Promolac system for e-marketing using location details was successfully executed with high accuracy. The method gives output in push notifications and message updates that helps to reduce the marketing gap between companies and end users. Thus, the implementation of system

output is done in Android and Python.