KHAMAR BONDHU

KHAMAR BONDHU V Test Plan

# *Test Plan Summary*

| Product Name | KHAMAR BONDHU |
| --- | --- |
| Product Description | A connected single-source mobile app for Khamari in Bangladesh to assist them all the time in managing their cattle farm more efficiently. |
| Test Codename | **KHAMAR BONDHU v** |
| Test Objective | Evaluate the stability of the product and its functions and features. Stability acceptance is the user evaluation of product stability, which includes quality, performance, reliability, and functionality of product topics. Measure the API performance under the stated load. And also Automate the UI operations of users. |

**Test Phases**

| Stage/Phase | Start Date | Duration | Test Cases | Hardware Testing | Total |
| --- | --- | --- | --- | --- | --- |
| Test Period | TBD |  | TBD | - | TBD |

**Schedule**

| Week / Number of Days | Topics/Activity | Performer |
| --- | --- | --- |
| Week 1 / 5 Days | End-to-end integration testing of released features |  |
| Week 1 / 5 Days | API testing |  |
| Week 2 / 5 Days | Exploratory testing / UI testing |  |
| Week 2 / 5 Days | Jmeter scripting preps for Load test |  |
| Week 3 / 5 Days | Bug reports and test closure preparation |  |
| Week 3 / 2 Days | Load test on live server or server with similar config \* |  |
| Week 4 / 5 Days | Appium Scripting for Automation |  |

**Results Measurement**

| Feedback Type | Objectives |
| --- | --- |
| Bug Reports | Test quality, interoperability, and real-world performance |

**Test Notes**

| Notes |
| --- |
| Load testing of the APIs will happen in Week 3 should all database connections and API links be completed |
| App distribution will be completed using Google Play. |
| Server should allow HTTP protocol for preparing scripts for the Load test; A debug APK is needed as well |

**Test Dependencies**

| Dependencies |
| --- |
| QA Server configuration resembling the LIVE Server FOR LOAD TEST |
| App distribution should be completed using Google Play. |
| A debug APK |
| Server with both HTTPS and HTTP protocol |
| Postman API collection |
| Mockups |
| SRS/PRD |

**Test Change Log**

| Date | Change Description | Author | Version |
| --- | --- | --- | --- |
| Oct 2, 23 | Draft test plan for SMT review | Mojammel Hossain | 0.1 |
| Oct 8, 23 | Test Case and Test dependency *added* | Mojammel Hossain | 0.2 |

Note: Any deviations from the current plan will be logged here.

# *Test Objectives*

**Released Features**

The current release of the KHAMAR BONDHU android app features the following functionality

| Features (all of these) |
| --- |
| Beasts detail profile record and relevant features (view, update) |
| Beast’s weight and the approximate amount of meat calculated from chest girth and body length |
| Farm management e-learning platform with the course and content features (22 courses, 123 contents) for Khamari. The update will be synced as users come online. |
| Income expenditure records in every sector |
| Income and cost analysis based on their submitted record |
| Debt and due records and analysis based on submitted record |
| Milk amount record and analysis |
| Cattle diet analysis |
| Service record from Vet |
| Service schedule, and relevant features (view, update) for Vet, LSP |
| Online Telemedicine service for all Khamari |
| Prescription records |
| Push Notification for Service Schedule, Telemedicine |
| Miscellaneous items (Help Line, sync) |

**Test Requirements**

All testers must meet the following base requirements to be considered a candidate for this test.

| Requirement (all of these) |
| --- |
| Measure the reliability of each of the different product features for information accuracy and presentation according to PRD/SRS |
| UI testing according to PRD/SRS |
| Test the API connectivity and ensure that data is available and reliable |
| Assess database integrity by adding and removing records throughout the application |
| Establish performance benchmarks through Performance Testing |
| Automate the UI (released features for future iteration) |
| Test against different versions of Android OS (7-13) and different hardware specifications (architecture, resolution) |

**Test Scope**

All the in-scope and out-of-scope items are listed below for this test cycle.

| In Scope | Out of Scope |
| --- | --- |
| Functionality of the released features according to PRD/SRS | Database testing |
| UI according to PRD/SRS | Center code Unit Testing |
| Conduct Performance testing \* |  |
| Preparing automation |  |
| Preparing scripts for Performance testing |  |

# *Testing Approach*

**Types of Testing**

These testing techniques will be used for the current version

| **API testing** - testing the functionality, reliability, performance, and security of the application programming interfaces (APIs) provided. |
| --- |
| **Integration testing** - testing front-end, a middleware layer that processes data, and a back-end database that stores data. Integration tests will verify if the data submitted in the front end is processed by the middleware and then stored by the backend database. |
| **Load testing** - testing that assesses the performance and response time of a software application under a specific workload. |
| **Exploratory testing** - testing unscripted, without a pre-defined test plan or test case. The tester relies on their own expertise, intuition, and creativity to explore the software and find defects. |
| **UI testing** - testing that focuses on testing the graphical user interface (GUI) of an application. The purpose of user interface testing is to ensure that the application’s GUI is functioning correctly and meets the requirements and expectations of end-user |
| **Appium automation** – preparing Appium scripts for future iteration |

**Testing Tools**

These testing tools will be used for the current version

| **Jmeter** – for performing the load test of the APIs |
| --- |
| **Appium** – for scripting the Android test automation |
| **Postman** – for testing the API as documented by the developers' |
| **Burp Suite** – for testing the API security, and functionality |
| **Android emulators** – for testing on Android 7 to Android 13 devices with varying resolutions and specifications |

# *Testing Environment and Data*

**Test Environment Configuration**

* All the servers should be in the scope of the QA environment
* Google Play store app link
* OTP should be configured for the deployment
* A default OTP generation process for load testing purposes
* The server should accommodate HTTP protocol for load-testing purposes
* A debug APK for traffic interception

**Test Data Requirements**

* Deployment should contain existing data from the live server.

# *Test Design and Execution*

**Test Cases and Scenarios**

Test Cases and Activities utilize an approach to gather insights on prioritized features of the product. This approach lays topics out as a map of the product. Cases are created on key features of the product. Issues encountered during testing are filed. Through analysis, actionable and insightful recommendations are provided in reports and weekly meetings

| Topic | Description | Activity | Size | Weight | OS |
| --- | --- | --- | --- | --- | --- |
| Week 1 | | | | | |
| Cattle Profile | Record relevant cattle details (name, age, color, number of teeth, prescription diet, weight, vaccination record, etc.) | • Record new beast information  • Measure beast’s weight  • Calculate beast’s diet  • Record beast’s medical history  • Record beast’s vaccination history  • Record beast’s breeding history  • Verify removing the beast from the list | L | 2.0 | ALL |
| Cattle Weight Measurement | Beast’s weight and the approximate amount of meat calculated from chest girth and body length | • Verify cattle’s weight calculation from chest girth and body length  • Verify the approximate amount of cattle meat calculation | L | 2.0 | ALL |
| Farm Management | A learning section, having some course and content to gain knowledge about cattle management | • Verify course content is appearing properly | L | 2.0 | ALL |
| Accounting | View accumulated income, expense, debit and credits in daily or monthly timespan | • Verify total and categorical income, expense, due, and credit in monthly and daily basis  • Record new income, expense, due or credit in details | L | 2.0 | ALL |
| Diet Calculation | Cattle diet chart calculation based on type, weight and amount of milk | • Verify cattle diet based on type, weight and amount of milk | S | 1.0 | ALL |
| Vaccination | Record new vaccination data and view vaccination history | • Record new vaccination data  • View vaccination history | M | 2.0 | ALL |
| Telemedicine | Connect with an online Vet for a telemedicine service | • Search from available Vet and book a Vet  • Make payments and get telemedicine service from a Vet | L | 2.0 | ALL |
| Notification | user will get notification when vet accept to provide sevice | • Verify getting notification for vet’s acceptance and prescription  • Verify view notification details |  |  |  |
| Side menu | Access sync and log out button | • Verify syncing persistent data properly from database  • Verify user can log out | L | 2.0 | ALL |
| Milk amount barchart | A graphical view of the amount of sold milk | • Verify amount of sold milk bar chart in daily, weekly and monthly basis | S | 1.0 | ALL |
| Hardware | Test against varying phone and tablet interfaces to verify compatibility | • Measure hardware compatibility and performance using required devices | L | 2.0 | ALL |
| Software | Verify across all compatible Android OS (7-13) | • Review system and build history and verify viability for beta testing | L | 2.0 | ALL |

**Product Topic**

These are not specifically covered in Test Activities but are used for the categorization of reports and for establishing common terms between stakeholder groups.

| Topics | Details | Weight |
| --- | --- | --- |
| Database | Issues related to database performance or accuracy | 1.75 |
| User Interface | The design elements of the application and their related usability | 2.0 |
| Performance | Speed and responsiveness related to application functionality | 1.5 |
| Links | Identify connections in the application and reach associated links | 1.5 |
| Text | Grammar, spelling, and other text elements in the application | 1.0 |
| Commerce | Any and all materials related to the commerce aspects of the application | 2.0 |
| Privacy | Systems regarding front-end access to security, personal identifying information, etc. | 2.0 |
| Other | Other areas of the product not covered by the above topics | 1.0 |

Weight: A value of 1.0 will not impact the Feedback Score. Higher values signify more importance while lower values signify less. This results in identifying higher-priority issues and ensures we gather sufficient feedback on key topics.

# *Testing Resources and Stakeholders*

**Test Team**

The following team will be assigned to this project and is responsible for the successful execution of this test.

| Name | Role | Responsibilities |
| --- | --- | --- |
| Mojammel Hossain | SQA Lead | Test plan and case design, test management, schedule maintenance, team participation, and test operations |
| Ashif Reza | SQA Engineer | Test case operation, test performance, bug tracking, bug submission, and suggestion submission |

**Product Team**

Team members will be invited to the platform and receive reports and status updates.

| Name | Project Role | Email Address | Phone Number | Time Zone |
| --- | --- | --- | --- | --- |
| Himel Baidya | Product Manager | himel@mpower-social.com | n/a | GMT +6 |
| Devashish Chy | Technical Lead | [devashish@mpower-social.com](mailto:vgonzales@explorecalifornia.org) | n/a | GMT +6 |

# *Bug Reports*

Testers will be expected to report detailed accounts of any negative issues they experience as Bug Reports. Once filtered by the Customer Validation team, existing Bug Reports are reported in JIRA under the *affected version* of this release; these bugs are available to other testers in order to collaborate and collect additional details.

**Bug Report Fields**

| Name | Description |
| --- | --- |
| Summary | A single line summary of the issue |
| Steps to Reproduce | Detailed steps to reproduce the issue |
| Test Platform | A profile of the device the problem was experienced on |
| File Attachments | Any associated files (such as screenshots or logs) |
| Topic | The impacted area of the product (see Topic list) |
| Severity | See Bug Report Severities |
| Status | Choices: Open, Re-opened, |
| Dev team Resolution | Choices: Resolved (Done, Won’t do, Later, Cannot Reproduce, Duplicate) |
| Comments | A collection of comments including additional file attachments |

**Bug Report Field: Severity**

| Label | Description | Weight |
| --- | --- | --- |
| Blocker | Defect causes a block in executing test cases or test works | 5 |
| Critical | Defect causes failure of essential operation of the system. A problem with a major Topic that exhibits itself frequently. A workaround may exist, but its use is not feasible for normal use. Item is affecting other items or systems not directly related to the test. Crashes, loss of data, severe memory leak, for example. | 4 |
| Major | Defects cause failure of noncritical aspects of the system, within a primary Topic. A problem with non-primary Topic that exhibits itself frequently | 3 |
| Minor | Nonessential operations are disrupted. A satisfactory or easy workaround exists to an otherwise Major severity. The problem is noncritical and not easily duplicated or does not reoccur often. | 2 |
| Trivial | No significant impact on operations. The defect does not impact performance. | 1 |

# *Testing Timelines*

Testers will be expected to report detailed accounts of any negative issues they experience as Bug Reports. Once filtered by the Customer Validation team, existing Bug Reports are reported in JIRA under the *affected version* of this release; these bugs are available to other testers in order to collaborate and collect additional details.

**Test Cycle Summary**

| Total Test Duration | 15 Days |
| --- | --- |
| Test Deliverables | Bug reports, Test Closure reports |

**Effort Estimations and Milestones**

| Week / Number of Days | Topics/Activity |  |
| --- | --- | --- |
| Week 1 / 5 Days | End-to-end integration testing of released features | Ashif Reza |
| Week 1 / 5 Days | API testing | Mojammel H |
| Week 2 / 5 Days | Exploratory testing / UI testing / Appium Scripting for Automation | Ashif Reza |
| Week 2 / 5 Days | Jmeter scripting preps for Load test | Mojammel H |
| Week 3 / 5 Days | Bug reports and test closure preparation | Ashif Reza |
| Week 3 / 2 Days | Load test on live server or server with similar config \* | Mojammel H |

**Risk Assessments**

* Results from load testing on a QA server won't accurately represent actual conditions. Any performance test should be run on the production server or a temporary server with a similar configuration in order to successfully understand the findings.
* Dependency for the performance test might cause delays in the milestone set in the Effort Estimation.
* Lack of proper and structured *class*, *id,* or *type* inside app component elements might lead to unnecessary obstacles in preparing automation scripts. This will be reported when encountered.

# *Communication and Collaboration*

This test plan will be communicated to all stakeholders, including the development team, product owner, and product managers via the JIRA task that’s been assigned to the QA team. Any changes in this test plan will be logged in the Summary section and the test plan will be replaced in the JIRA task.