GamingBets Application

Test Plan

Version 1.0

Table of Contents

1. Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Intended Audience 3

2. Evaluation Mission and Test Motivation 3

2.1 Background 3

2.2 Evaluation Mission 3

2.3 Test Motivators 3

3. Target Test Items 3

4. Outline of Planned Tests 3

5. Test Approach 4

5.1 Initial Test-Idea Catalogs and Other Reference Sources 4

5.2 Testing Techniques and Types 4

5.2.1 Function Testing 4

5.2.2 User Interface Testing 4

5.2.3 Installation Testing 4

6. Deliverables 5

6.1 Test Evaluation Summaries 5

6.2 Reporting on Test Coverage 5

6.3 Perceived Quality Reports 5

6.4 Incident Logs and Change Requests 5

6.5 Smoke Test Suite and Supporting Test Scripts 5

6.6 Additional Work Products 5

6.6.1 Detailed Test Results 5

7. Testing Workflow 5

8. Environmental Needs 5

8.1 Base System Hardware 5

8.2 Base Software Elements in the Test Environment 5

8.3 Productivity and Support Tools 6

9. Responsibilities, Staffing, and Training Needs 6

9.1 People and Roles 6

10. Iteration Milestones 8

Test Plan

# Introduction

## Purpose

The purpose of the Iteration Test Plan is to gather all of the information necessary to plan and control the test effort for a given iteration. It describes the approach to testing the software, and is the top-level plan generated and used by managers to direct the test effort.

This *Test Plan* for the GamingBets Application supports the following objectives:

* The most critical areas like the evaluation and creation of Bets, as well as the rewarding system
* The test to ensure we don’t damage existing code by adding new features
* The test coverage should be at least 20%
* The use of Sonar Cube for identifying test issues

## Scope

Our Test plan contains Unit Testing, Functional Testing and Stress Testing. GamingBets is divided into two Projects, the Server and the App. Because of that Functional Testing will be provided for the App to ensure a consistent behavior, unit test for the app aren´t very reasonable, because it does not contain significant logic.

The Server functionality will be covered by Unit Tests and also will be stressed by some Stress Tests. Functional Test will not be provided for the Server Side, because there is simply no behavior to test. The most important part to test is the general betting logic, because this is the main component of our application.

## Intended Audience

This document is not meant for any end-user, it is supposed to explain the test behavior while the application is still under development. It helps to understand the focus of this project.

# Evaluation Mission and Test Motivation

## Background

Testing is a key element of software development for us to be able to develop software that matches the requirements.

## Evaluation Mission

The overall goal is to ensure features and components work as there are intend to. Further we ensure to not accidentally brake a working feature, or at least get notified about it.

## Test Motivators

By good and consistent testing we will spend a lot less time/money to ensure software quality and the right behavior.

# Target Test Items

The listing below identifies those software-elements that have been identified as targets for testing. This list represents what items will be tested:

* Android Application: Functional Testing with at least one Scenario for each Use Case
* Server: Unit Testing for key services
* Server: Stress Testing

# Outline of Planned Tests

n/a

# Test Approach

## Initial Test-Idea Catalogs and Other Reference Sources

n/a

## Testing Techniques and Types

### Function Testing

|  |  |
| --- | --- |
| Technique Objective: | Testing the functionality of server sided code. |
| Technique: | There for execute different micro services which handle behaviors like data extraction and parsing from third party API´s, creating available bets, evaluating bets and closing finished bets. |
| Oracles: | - |
| Required Tools: | JUnit |
| Success Criteria: | All Tests pass. |
| Special Considerations: | For testing behavior regarding Data from third party APIs, generate some own data, don´t use live data and/or do API calls. |

### User Interface Testing

|  |  |
| --- | --- |
| Technique Objective: | Test UC-behaviour. |
| Technique: | Test the behavior of activity diagrams for each Use-Case, which are defined in the UC-Documents. |
| Oracles: | - |
| Required Tools: | * Gherkin * Cucumber * Calabash * Android device or emulator |
| Success Criteria: | All test run successful. |

### Installation Testing

|  |  |
| --- | --- |
| Technique Objective: | Test installation of the *GamingBets*-Android App on different devices |
| Technique: | * Install via AndroidStudio directly to multiple devices * Generate .apk file, copy it to multiple android devices and install then on devices via Android installer |
| Oracles: | - |
| Required Tools: | * Android Studio * Multiple Android devices or multiple android emulators with different versions of android and different models |
| Success Criteria: | App should install successfully and afterwards work correctly. |

# Deliverables

## Test Evaluation Summaries

Test evaluation is done after each test run, and can be found in the build log files on [Jenkins](http://134.255.218.20:5000/job/GamingBetsServer/).

## Reporting on Test Coverage

Test coverage reports can be found as [interactive html form on GitHub](https://github.com/GamingBets/documents/tree/master/TestReports/Function%20Testing). (We recommend that you download the whole folder and open it in your browser)

## Perceived Quality Reports

Quality reports can be found on [SonarQube](http://193.196.7.25/overview?id=6417).

## Incident Logs and Change Requests

n/a

## Smoke Test Suite and Supporting Test Scripts

n/a

## Additional Work Products

### Detailed Test Results

The latest test results can be found here:

* [Function Testing](http://134.255.218.20:5000/job/GamingBetsServer/)
* [User Interface Testing](https://github.com/GamingBets/documents/tree/master/TestReports/User%20Interface%20Testing)

# Testing Workflow

Functional Tests can be / are triggered in different ways:

* On each automated build by Jenkins, function tests will executed while building. The build won´t be deployed if not all tests are passing.
* On each manual maven build, function test will be executed while building. The build is never auto deployed.
* Test can be triggered manually while developing in our IDEs.

User Interface Tests can be triggered manually.

# Environmental Needs

## Base System Hardware

The following table sets forth the system resources for the test effort presented in this *Test Plan*.

| **System Resources** | | |
| --- | --- | --- |
| **Resource** | **Quantity** | **Name and Type** |
| Database Server |  | MySQL Database |
| —Network or Subnet |  | 134.255.218.20/localhost |
| —Server Name |  | - |
| —Database Name |  | gamingbets |

## Base Software Elements in the Test Environment

The following base software elements are required in the test environment for this *Test Plan*.

| **Software Element Name** | **Version** | **Type and Other Notes** |
| --- | --- | --- |
| debian / Windows | 7 or higher (each) | Operating System |
| Java JDK | 7 or higher |  |
| JUnit | 4 |  |
| Glassfish | 3 | Full Application Server |
| Android |  |  |
| Calabash-Android |  |  |
| Ruby |  | For Calabash-Android |
| Appropriate IDE |  | Android Studio, Eclipse, NetBeans |
| Maven |  |  |
| GIT |  |  |
| MySQL Server |  | e.g. through XAMPP |

## Productivity and Support Tools

The following tools will be employed to support the test process for this *Test Plan*.

| **Tool Category or Type** | **Tool Brand Name** | **Vendor or In-house** | **Version** |
| --- | --- | --- | --- |
| DBMS tools | MySQL Workbench |  |  |

# Responsibilities, Staffing, and Training Needs

## People and Roles

This table shows the staffing assumptions for the test effort.

| **Human Resources** | | |
| --- | --- | --- |
| **Role** | **Minimum Resources Recommended**  **(number of full-time roles allocated)** | **Specific Responsibilities or Comments** |
| Test Manager | Felix Morsbach (1) | Provides management oversight.  Responsibilities include:   * planning and logistics * agree mission * identify motivators * acquire appropriate resources * present management reporting * advocate the interests of test * evaluate effectiveness of test effort |
| Test Analyst | Felix Morsbach (2) | Identifies and defines the specific tests to be conducted.  Responsibilities include:   * identify test ideas * define test details * determine test results * document change requests * evaluate product quality |
| Test Designer | Felix Morsbach (1) | Defines the technical approach to the implementation of the test effort.  Responsibilities include:   * define test approach * define test automation architecture * verify test techniques * define testability elements * structure test implementation |
| Tester | All team member (2) | Implements and executes the tests.  Responsibilities include:   * implement tests and test suites * execute test suites * log results * analyze and recover from test failures * document incidents |
| Test System Administrator | Felix Morsbach (1) | Ensures test environment and assets are managed and maintained.  Responsibilities include:   * administer test management system * install and support access to, and recovery of, test environment configurations and test labs |
| Database Administrator, Database Manager | Felix Morsbach, André Helbig (2) | Ensures test data (database) environment and assets are managed and maintained.  Responsibilities include:   * support the administration of test data and test beds (database). |
| Designer | Felix Morsbach (1) | Identifies and defines the operations, attributes, and associations of the test classes.  Responsibilities include:   * defines the test classes required to support testability requirements as defined by the test team |
| Implementer | All team member (2) | Implements and unit tests the test classes and test packages.  Responsibilities include:   * creates the test components required to support testability requirements as defined by the designer |

# Iteration Milestones

Planned date of milestone achievement: 05.06.2016

* > 20 % Test Coverage for Server Side Logic
* 10 Scenarios for UCs.
* Successfully completed installation on at least 5 different android devices.