SheeshApp

Version <1.0>

[Note: The following template is provided for use with the Rational Unified Process. Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]

[To customize automatic fields in Microsoft Word (which display a gray background when selected), select File>Properties and replace the Title, Subject and Company fields with the appropriate information for this document. After closing the dialog, automatic fields may be updated throughout the document by selecting Edit>Select All (or Ctrl-A) and pressing F9, or simply click on the field and press F9. This must be done separately for Headers and Footers. Alt-F9 will toggle between displaying the field names and the field contents. See Word help for more information on working with fields.]

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 3.5.2018 | 1.0 | Initial Version | Alexander Grundmann |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 5

1.1 Purpose 5

1.2 Scope 5

1.3 Intended Audience 5

1.4 Document Terminology and Acronyms 5

1.5 References 5

1.6 Document Structure 5

2. Evaluation Mission and Test Motivation 5

2.1 Background 6

2.2 Evaluation Mission 6

2.3 Test Motivators 6

3. Target Test Items 6

4. Outline of Planned Tests 6

4.1 Outline of Test Inclusions 6

4.2 Outline of other candidates for potential inclusion 6

4.3 Outline of Test Exclusions 7

5. Test Approach 7

5.1 Initial Test-Idea Catalogs and other reference sources 7

5.2 Testing Techniques and Types 7

5.2.1 Data and Database Integrity Testing 7

5.2.2 Function Testing 8

5.2.3 Business Cycle Testing 10

5.2.4 User Interface Testing 11

5.2.5 Performance Profiling 11

5.2.6 Load Testing 13

5.2.7 Stress Testing 14

5.2.8 Volume Testing 16

5.2.9 Security and Access Control Testing 17

5.2.10 Failover and Recovery Testing 18

5.2.11 Configuration Testing 20

5.2.12 Installation Testing 21

6. Entry and Exit Criteria 22

6.1 Test Plan 22

6.1.1 Test Plan Entry Criteria 22

6.1.2 Test Plan Exit Criteria 22

6.1.3 Suspension and resumption criteria 22

6.2 Test Cycles 22

6.2.1 Test Cycle Entry Criteria 22

6.2.2 Test Cycle Exit Criteria 22

6.2.3 Test Cycle abnormal termination 22

7. Deliverables 22

7.1 Test Evaluation Summaries 22

7.2 Reporting on Test Coverage 22

7.3 Perceived Quality Reports 22

7.4 Incident Logs and Change Requests 23

7.5 Smoke Test Suite and supporting Test Scripts 23

7.6 Additional work products 23

7.6.1 Detailed Test Results 23

7.6.2 Additional automated functional Test Scripts 23

7.6.3 Test Guidelines 23

7.6.4 Traceability Matrices 23

8. Testing Workflow 23

9. Environmental Needs 24

9.1 Base System Hardware 24

9.2 Base Software Elements in the Test Environment 24

9.3 Productivity and Support Tools 25

9.4 Test Environment Configurations 25

10. Responsibilities, Staffing and Training Needs 25

10.1 People and Roles 25

10.2 Staffing and Training Needs 27

11. Iteration Milestones 28

12. Risks, Dependencies, Assumptions and Constraints 28

13. Management Process and Procedures 29

13.1 Measuring and Assessing the Extent of Testing 29

13.2 Assessing the deliverables of this Test Plan 29

13.3 Problem Reporting, Escalation and Issue Resolution 30

13.4 Managing Test Cycles 30

13.5 Traceability Strategies 30

13.6 Approval and Signoff 30

# 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 SheeshApp supports the following objectives:

* Basic testing of core functionality
* Test driven Development

## Scope

* Unit Testing
* Runtime tesing

## Intended Audience

We as developers

## Document Terminology and Acronyms

N/A

## References

N/A

## Document Structure

N/A

# Evaluation Mission and Test Motivation

The intention is to find old bugs and implement test-driven-development to prevent future ones

## Background

Bugs are not always obvious, Unit testing will show us bugs and prevents us from changing crucial parts of the code into disfunctionality

## Evaluation Mission

* Preventing future bugs
* Finding old hidden bugs
* Ensuring not to break code by changes

## Test Motivators

N/A

# Target Test Items

The listing below identifies those test items⎯software, hardware, and supporting product elements ⎯that have been identified as targets for testing. This list represents what items will be tested.

* Internal Code
* Expected Workflow

# Outline of Planned Tests

## Outline of Test Inclusions

* Any Usecase should offer a couple of tests to ensure correct behaviour

## Outline of Other Candidates for Potential Inclusion

N/A

## Outline of Test Exclusions

External Code is expected to be working so it is not necessary to test those cases.

Many cases are bound to individual Android test-devices and are therefore not easy to test.

# Test Approach

## Initial Test-Idea Catalogs and Other Reference Sources

UseCase based testing

## Testing Techniques and Types

### Data and Database Integrity Testing

|  |  |
| --- | --- |
| Technique Objective: | Ensuring data consistency and correct calculation bahaviour. Testing Server Connections |
| Technique: | Connect to Server, get Data of faked User. Ensure data is correct. |
| Oracles: | N/A |
| Required Tools: | Server  SQL  JUnit |
| Success Criteria: | Data contains valid information |
| Special Considerations: | Automated on build. Care with offline dependency. |

### Function Testing

|  |  |
| --- | --- |
| Technique Objective: | Ensuring correct behavior of the GUI |
| Technique: | Automated Execution by test software |
| Oracles: | N/A |
| Required Tools: | Espresso  A valid setup of tests |
| Success Criteria: | Expected behavior  No errors thrown  No missing buttons |
| Special Considerations: | Maybe Additionally Unit Tests |

### 

### 

### Volume Testing

|  |  |
| --- | --- |
| Technique Objective: | Ensuring the database does not break on long histories etc. |
| Technique: | Manual Unit tests |
| Oracles: | N/A |
| Required Tools: | JUnit  Server  SQL |
| Success Criteria: | Loading a fake User with a large History size suceeds |
| Special Considerations: | N/A |

### Security and Access Control Testing

|  |  |
| --- | --- |
| Technique Objective: | Ensuring the database can not be accessed in illegal ways.  Encryption |
| Technique: | Manual Break Tests (Trying to break in) |
| Oracles: | N/A |
| Required Tools: | Various |
| Success Criteria: | Everything is secured and not illegally accessible |
| Special Considerations: | Encryption for precaution |

### Failover and Recovery Testing

|  |  |
| --- | --- |
| Technique Objective: | Ensuring rare exceptions do not lead to unwanted behavior:   * Connection loss * Weird Characters * … |
| Technique: | Simulation |
| Oracles: | N/A |
| Required Tools: | Any |
| Success Criteria: | Every Exception is handled properly |
| Special Considerations: | Testing different machines might be useful |

# Entry and Exit Criteria

## Test Plan

### Test Plan Entry Criteria

Tests are written

### Test Plan Exit Criteria

When we can’t think of any more possible bugs

### Suspension and Resumption Criteria

Time pressure might cancel testing. We try to avoid this.

## Test Cycles

### Test Cycle Entry Criteria

Upon build automatically

### Test Cycle Exit Criteria

N/A

### Test Cycle Abnormal Termination

We terminate the build when tests trigger

# Deliverables

## Test Evaluation Summaries

Generated each build

## Reporting on Test Coverage

Issued after each cycle

## Perceived Quality Reports

No

## Incident Logs and Change Requests

No

## Smoke Test Suite and Supporting Test Scripts

JUniz+Espresso, automated

## Additional Work Products

### Detailed Test Results

Generated per build

### Additional Automated Functional Test Scripts

Per build

### Test Guidelines

Test-driven

### Traceability Matrices

No

# Testing Workflow

Testing happens automatically, after adding a feature tests have to be written and added to the autotest

# 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 |  |  |
| —Network or Subnet |  | TBD |
| —Server Name |  | TBD |
| —Database Name |  | TBD |
| Client Test PCs |  |  |
| —Include special configuration requirements |  | TBD |
| Test Repository |  |  |
| —Network or Subnet |  | TBD |
| —Server Name |  | TBD |
| Test Development PCs |  | TBD |

## Base Software Elements in the Test Environment

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

JUnit

## Productivity and Support Tools

|  |  |  |  |
| --- | --- | --- | --- |
| Test Coverage Monitor or Profiler |  |  |  |
| Project Management |  |  |  |

## Test Environment Configurations

The following Test Environment Configurations needs to be provided and supported for this project.

| **Configuration Name** | **Description** | **Implemented in Physical Configuration** |
| --- | --- | --- |
| Average user configuration | The default app installation | Several Hardware |

# 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 | both | 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 Designer | both | 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 | both | 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 |
| Database Administrator, Database Manager | 1 | Ensures test data (database) environment and assets are managed and maintained.  Responsibilities include:   * support the administration of test data and test beds (database). |

## Staffing and Training Needs

Not Needed

# Iteration Milestones

| **Milestone** | **Planned Start Date** | **Actual Start Date** | **Planned End Date** | **Actual End Date** |
| --- | --- | --- | --- | --- |
| Iteration Plan agreed |  |  |  |  |
| Iteration starts |  |  |  |  |
| Requirements baselined |  |  |  |  |
| Architecture baselined |  |  |  |  |
| User Interface baselined |  |  |  |  |
| First Build delivered to test |  |  |  |  |
| First Build accepted into test |  |  |  |  |
| First Build test cycle finishes |  |  |  |  |
| [Build Two will not be tested] |  |  |  |  |
| Third Build delivered to test |  |  |  |  |
| Third Build accepted into test |  |  |  |  |
| Third Build test cycle finishes |  |  |  |  |
| Fourth Build delivered to test |  |  |  |  |
| Fourth Build accepted into test |  |  |  |  |
| Iteration Assessment review |  |  |  |  |
| Iteration ends |  |  |  |  |

# Risks, Dependencies, Assumptions, and Constraints

See Risk Sheet

# Management Process and Procedures

N/A

## Measuring and Assessing the Extent of Testing

N/A

## Assessing the Deliverables of this Test Plan

N/A

## Problem Reporting, Escalation, and Issue Resolution

N/A

## Managing Test Cycles

N/A

## Traceability Strategies

N/A

## Approval and Signoff

N/A