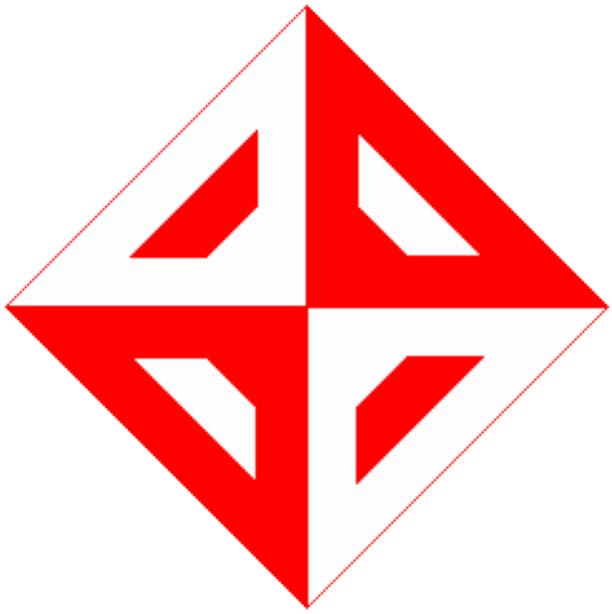
SOFTWARE TEST DOCUMENT

03.05.2015

MASTER KING PROJECT



BEDRETTİN ÇETİNKAYA - 1819788
UĞUR ÇOĞALAN -1949825
OĞUZHAN YÜZER - 1819713
OĞUZ ÇAKMAK - 1819739

Table of Contents

1. Introduction

These sections below identify this software test document and its scope. These sections also give information about the context in which this document is prepared and about detailed test conditions and results .

1.1 Document Identifier

This document is the Software Test Document of the project which is Master King by team Rebellion. It is based on the IEEE Std 829 - 2008, IEEE Standard for Software and System Test Documentation.

Purpose of this document is, verifying and validating the system's feature, determining undesirable system's behavior and situation. Also, this document provides the team information about

- performance
- bug detection
- logically correct system
- usable and reliable product

1.2 Scope

The software product is Android application developed with server-client architecture. It has three components which are Decision AI, Client and Network. Therefore, each component of the system should be tested individually (component testing). Also, system testing should be conducted on a complete, integrated system. Since the product is developed in incremental development scrum methodology, unit testing has been applied during the development phase. For this reason, unit testing results are not included in this document.

1.3 References

- IEEE Std 829-2008, IEEE Standard for Software and System Test Documentation
- Software Design Description of Master King
- Software Requirement Specification of Master King

1.4 Level in the overall sequence

As mentioned in section 1.2, unit testing is conducted before. Hence, there are two level testing which are component and system testing.

1.5 Test Classes and Overall Test Conditions

Component testing level covers functionalities of components described in Software Design Document of Master King. It is conducted to verify that each component functions as expected.

System testing level covers whole system. It is conducted to verify that communication between components operates as expected.

Hence, test conditions are composed by considering system's and users' requirements described in Software Requirement Specification Document of Master King and users' expectation/feedbacks.

2. Details for System Test Plan

These sections below describe test items, the features to be tested and not tested, traceability matrix, evaluation criteria for pass/fail the test and which approaches are used for testing.

2.1 Test Items and Their Identifiers

The test items are the functionalities specified in Software Requirement Specification document and expectation of developer/end users. Some of these functionalities are described by means of use cases. The test items are listed below.

1. Choose Language
2. Multi-user Support
3. Different Android Densities
4. Animations
5. User Interaction
6. The Real Life Game Rules
7. Network Data Communication
8. Card Fetch Time
9. Popup
10. Agent Playing Strategy
11. Ending Session
12. Folder System

2.2 Test Traceability Matrix

Use Cases / Test Cases	Choose Language UC	Choose Game UC	Play Card UC	Display Score Table UC	Register UC	Login UC	Logout UC
MasterKing.TC.1	✔						
MasterKing.TC.2							
MasterKing.TC.3		✔	✔		✔	✔	✔
MasterKing.TC.4			✔				
MasterKing.TC.5		✔	✔				
MasterKing.TC.6		✔	✔				
MasterKing.TC.7		✔	✔				
MasterKing.TC.8			✔				
MasterKing.TC.9		✔					
MasterKing.TC.10		✔	✔				
MasterKing.TC.11							✔
MasterKing.TC.12		✔	✔				

Table 1: Test Traceability Matrix

2.3 Features to be Tested

In section 2.1, the cases to be tested are given. These cases are the functionalities that the user is provided with, system's behavior and correct component and data communication.

Functionalities are tested manually. On the other hand, system's behavior and correct component and data communication are tested with system's codes and manually.

2.4 Features not to be Tested

Unit testing is not included in this testing document. Therefore, functionalities of each unit has not been tested for this document.
Also, unimplemented modules of the project, (database connection , user login and register operation, score table of the game and no last two tricks AI) will be tested later.

2.5 Approach

The approach that is followed in this testing phase is Black Box. In this Black Box method, the inputs are the user's interaction, game rules, data specific to the component and the outputs are the system's responses to these interactions.

- The interactions are touching the buttons on the screen.
- Game rules are real life rules of the king game.
- Data can vary in accordance with component.

2.6 Item pass/Fail criteria

The tested items are considered to pass the test if the output is exactly the same as the specified output in SRS document, end users' expectation and correct data processing. On the other hand, the items that are used for testing are evaluated as fail, if the output is not the expected one, anomaly, bug or error is detected in the system's response or behaviour.

2.7 Test Deliverables

Test deliverables are already specified in this text. It includes the combination of the context of the following documents:

- Level Test Plan(s)
- Level Test Case
- Level Test Detailed Report

Section 1, 2 and 3 cover the Level Test Plan(s), section 4 covers the context in Level Test Case and section 5 covers the Level Test Detailed Report.

3. Test management

3.1 Planned Activities and Tasks; Test Progression

In this section, the way in which the system should be tested to verify stability, reliability and responsiveness is explained.

Firstly, system boundaries should be determined before starting the test. Also, minimum hardware and software properties should be detected. According to these boundaries, test conditions should be composed. Secondly, while testing, different scenarios should be considered for accuracy.
Finally, test cases should be efficient, each test should give as much information as possible with minimum effort and they should force the boundaries. In the consideration of these planned activities and tasks, test progression will proceed accordingly

3.2 Environment / Infrastructure

All environments that are used are explained at section 5.1.1. of this STD document.

4. Test case details

This section will explain the detailed information about the test case for each functional requirement. Each test case includes objective, input, outcome, environmental needs, special procedural requirements and intercase dependencies.

4.1. Test Case: Choose Language

Test Case Id	**MasterKing.TC.1**
Objective	This test case aims to verify that the language of the MasterKing application should be changed with respect to the language of the Android device.
Input	User chooses the language from the settings menu on the Android device.
Outcome	The language of the MasterKing application should be changed with respect to the language of the Android device.
Environmental needs	-
Special procedural	-

requirements	
Intercase Dependencies	-

Table 2: Test Case 1

4.2 Test Case: Multi-User Support

Test Case Id	**MasterKing.TC.2**
Objective	This test case aims to verify that whether MasterKing application is able to support multiple users simultaneously or not.
Input	User enters the game and Android device connects to the server.
Outcome	MasterKing application server opens a thread for each user and users should be able to play the game without any problem.
Environmental needs	User should have consistent Internet connection to be able to communicate with the server.
Special procedural requirements	Before starting the MasterKing application, user should open the Wi-Fi or 3G connection.
Intercase Dependencies	-

Table 3: Test Case 2

4.3 Test Case: Different Android Densities

Test Case Id	**MasterKing.TC.3**
Objective	This test case aims to verify that whether MasterKing application can work different Android screen sizes.
Input	-
Outcome	MasterKing application's all activities should look same and consistent regardless of the screen size.
Environmental needs	User should have consistent Internet connection to be able to open all activites.
Special procedural requirements	Before starting the MasterKing application, user should open the Wi-Fi or 3G connection.
Intercase Dependencies	-

Table 4: Test Case 3

4.4 Test Case: Animations

Test Case Id	MasterKing.TC.4
Objective	This test case aims to verify that whether MasterKing application's sound, click and translate animations are working properly or not.
Input	The cards of the user are animated when the cards are clicked. The cards of the agents are animated when the card number and type returns from the server. Sounds should be played in accordance with animations.
Outcome	The cards should be animated smoothly and should be located in their corresponding places. Animation start triggers the playing of sound.
Environmental needs	User should have proper working touch screen Android device.
Special procedural requirements	User should click his/her cards and server should return the card number and type.
Intercase Dependencies	-

Table 5: Test Case 4

4.5 Test Case: User Interaction

Test Case Id	MasterKing.TC.5
Objective	This test case aims to verify that whether MasterKing application is able to interact with user's preferences.
Input	User should interact with the cards.
Outcome	When it's user's turn, the Android device should wait until the user interacts with the cards. When it's not the user's turn, application disables the user's interaction.
Environmental needs	User should have proper working touch screen Android device.
Special procedural requirements	This test case lasts until the game finishes.
Intercase Dependencies	-

Table 6: Test Case 5

4.6 Test Case: The Real Life Game Rules

Test Case Id	MasterKing.TC.6
Objective	This test case aims to verify that whether MasterKing application applies the real life game rules of the King card game.
Input	-
Outcome	For both negative and positive hands, there shouldn't be any cases that violates the real life game rules.
Environmental needs	-
Special procedural requirements	The game should be run for each sub game from beginning to end to observe the results.
Intercase Dependencies	-

Table 7: Test Case 6

4.7 Test Case: Network Data Communication

Test Case Id	MasterKing.TC.7
Objective	This test case aims to verify that whether MasterKing application is able to transmit and receive data correctly without any data loss.
Input	The Android device should open the TCP socket and server should accept the connection over this socket.
Outcome	There shouldn't be any errors on bidirectional data communication between client and server.
Environmental needs	User should have consistent Internet connection to be able to communicate with the server.
Special procedural requirements	Before starting the MasterKing application, user should open the Wi-Fi or 3G connection.
Intercase Dependencies	-

Table 8: Test Case 7

4.8 Test Case: Card Fetch Time

Test Case Id	MasterKing.TC.8
Objective	This test case aims to verify that whether MasterKing server is able to fetch card data inside a huge amount of card data as fast as possible.
Input	The MasterKing server should have relative path to find the data in our system.
Outcome	The demanded data should return within the expected time.
Environmental needs	User should meet the hardware requirements.
Special procedural requirements	Before fetching the card data, corresponding path should be calculated.
Intercase Dependencies	-

Table 9: Test Case 8

4.9 Test Case: Popup

Test Case Id	MasterKing.TC.9
Objective	This test case aims to verify that whether MasterKing application's popup satisfies its functional requirements.
Input	Popup should open when it is user's turn.
Outcome	Popup should direct user to chosen game and popup should be updated according to the available games.
Environmental needs	Popup opens automatically when it is necessary.
Special procedural requirements	User should run the game and should start the play activity.
Intercase Dependencies	-

Table 10: Test Case 9

4.10 Test Case: Agent Playing Strategy

Test Case Id	MasterKing.TC.10
Objective	This test case aims to verify that whether MasterKing application's agents satisfy their taught strategies.
Input	-
Outcome	Agent playing strategies should meet end-users' expectations.
Environmental needs	Since it is an Android base application, to test application with end-users any Android device is required.
Special procedural requirements	End-users should play the game in detail.
Intercase Dependencies	-

Table 11: Test Case 10

4.11 Test Case: Ending Session

Test Case Id	MasterKing.TC.11
Objective	This test case aims to verify that whether MasterKing server should finish the connection with server.
Input	User should leave the game using back press button.
Outcome	When user leaves the game, the connection with the server should be ended.
Environmental needs	User should have consistent Internet connection to be able to communicate with the server.
Special procedural requirements	User can exit the game any time he/she wants.
Intercase Dependencies	-

Table 12: Test Case 11

4.12 Test Case: Folder System

Test Case Id	MasterKing.TC.12
Objective	This test case aims to verify that whether the folder system and its inner file structures are created successfully or not.
Input	Corresponding java code that creates the folder system.
Outcome	The folder system should have correct number of directories and file

	structures should be created correctly.
Environmental needs	JDK should be installed on the Operating System.
Special procedural requirements	Java code should be compiled and afterwards it should be executed.
Intercase Dependencies	-

Table 13: Test Case 12

5. System Test Report Details

5.1 Overview of Test Results

After all test cases have been conducted, it has been seen that system works as expected, meaning that system passed all tests. However, it cannot be said that the system is completely without bugs. In some cases, system may behave inconsistenly. When such cases are detected the problems will be fixed and new test cases will be added.

5.1.1 Test Environments

- Ubuntu 14.10, 2.40Ghz Processor , 2 Mb cache ,143 Gb Harddisk,2 cores for the server
- Android 4.4.2 , 2.66 Gb Ram , 32 Gb Harddisk for Client
- Android 4.4.2 , 2 Gb Ram , 16 Gb Harddisk for Client
- Android 4.1.2 , 463 Mb Ram, 4 Gb Harddisk for Client
- Android 4.1.2 , 804 Mb Ram , 4.64 Gb Harddisk for Client
- Android 4.4.2 , 1.5Gb Ram ,16 Gb Harddisk for Client

5.2 Detailed Test Results

MasterKing.TC.1	Passed
MasterKing.TC.2	Passed
MasterKing.TC.3	Passed
MasterKing.TC.4	Passed
MasterKing.TC.5	Passed

MasterKing.TC.6	Passed
MasterKing.TC.7	Passed
MasterKing.TC.8	Passed
MasterKing.TC.9	Passed
MasterKing.TC.10	Passed
MasterKing.TC.11	Passed
MasterKing.TC.12	Passed

Table 14: Test Case Results

5.3 Rationale for decisions

Test process is done to demonstrate to the end users that the software meets is requirements and to discover program defects, bugs before the application is put into Android market. The result of the test run is checked to find errors, anomalies. However, testing process does not guarantee that the application does not face with any more problems.

5.4 Conclusions and Recommendation

As a result, project Master King application passed all the tests case that is in this document. However, it does not mean that system will work perfectly. Also, unimplemented method/modules will be tested.