MAKERERE****UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATION SCIENCES

DEPARTMENT OF NETWORKS

BACHELOR OF SCIENCE IN SOFTWARE ENGINEERING (YEAR 2)

RECESS TERM 2 (BSE 2301)

SOFTWARE REQUIREMENTS SPECIFICATIONS FOR:

GROUP 11

GROUP MEMBERS

|  |  |  |
| --- | --- | --- |
| NAME | REGISTRATION NUMBER | STUDENT NUMBER |
| KEKIRUNGA JEAN | 16/U/5850/PS | 216004629 |
| MUGISA BEST | 16/U/7279/PS | 216018395 |
| KISIGA TIMOTHY | 16/U/6173/PS | 216003537 |

PROJECT LEADER: KISIGA TIMOTHY

ADVISOR: NOAH KANGE

SOFTWARE REQUIREMENTS SPECIFICATION

For

IOS MOBILE APP ANALYSIS DOCUMENT

Version 1.0 approved

Prepared by Kekirunga Jean, Kisiga Timothy, Mugisa Best

GROUP 11

26th June, 2018

Table of Contents

[1. Introduction 5](#_Toc519528079)

[1.1 Purpose 5](#_Toc519528080)

[1.2 Document Conventions 5](#_Toc519528081)

[1.3 Intended audience and Reading suggestions 5](#_Toc519528082)

[1.4 References 5](#_Toc519528083)

[1.5 Product scope 6](#_Toc519528084)

[2. Overall description 6](#_Toc519528085)

[2.1 Product perspective 6](#_Toc519528086)

[2.2 Product functions 6](#_Toc519528087)

[2.3 User classes and characteristics 9](#_Toc519528088)

[2.4 Operating environment 9](#_Toc519528089)

[2.5 Design and implementation constraints 9](#_Toc519528090)

[2.6 User documentation 9](#_Toc519528091)

[3. External interface requirements 9](#_Toc519528092)

[3.1 User interfaces 9](#_Toc519528093)

[3.2 Hardware interfaces 10](#_Toc519528095)

[3.3 Software interfaces 10](#_Toc519528096)

[3.4 Communication interfaces 10](#_Toc519528097)

[4. System features 11](#_Toc519528098)

[4.1 App statistics analysis for different groups 11](#_Toc519528099)

[4.1.1 Description and policy 11](#_Toc519528100)

[4.1.2 Stimulus and Response sequences 11](#_Toc519528101)

[4.1.3 Functional requirement 11](#_Toc519528102)

[4.2 User ratings and app details analysis 11](#_Toc519528103)

[4.2.1 Description and priority 11](#_Toc519528104)

[4.2.2 Stimulus/ response sequence 11](#_Toc519528105)

[4.2.2 Functional requirement 11](#_Toc519528106)

[4.3 App comparison in the same group 11](#_Toc519528107)

[4.3.1 Description and policy 11](#_Toc519528108)

[4.3.2 Stimulus / Response 12](#_Toc519528109)

[4.3.3 Functional requirement 12](#_Toc519528110)

[4.4 App description analysis 12](#_Toc519528111)

[4.4.1 Description and priority 12](#_Toc519528112)

[4.4.2 Stimulus / Response 12](#_Toc519528113)

[4.4.3 Functional Requirement 12](#_Toc519528114)

[4.5 Information Update 12](#_Toc519528115)

[4.5.1 Description and policy 12](#_Toc519528116)

[4.5.2 Stimuli/ Response 12](#_Toc519528117)

[4.5.3 Functional Requirement 12](#_Toc519528118)

[5. Other Non-functional Requirements 13](#_Toc519528119)

[5.1 Performance requirements 13](#_Toc519528120)

[5.2 Safety requirements 13](#_Toc519528121)

[5.3 Security requirements 13](#_Toc519528122)

[5.4 Software quality attributes 13](#_Toc519528123)

[5.5 Business rules 14](#_Toc519528124)

[6. Appendix: Glossary 14](#_Toc519528125)

**List of figures**

[figure 1: Context diagram 1 7](#_Toc518382414)

[figure 2: Use case diagram 1 8](#_Toc518382556)

[figure 3: Home page 1 10](#_Toc518382397)

# Introduction

# 1.1 Purpose

The purpose of this SRS document is to present a detailed description of the different functionalities of the system which we are going to develop and implement; a software that studies the relationship between app details and user ratings in order to help Apple developers study the features of the most downloaded apps on the Apple app store and incorporate them in the apps they develop to increase the user ratings for their apps.

The document will explain the purpose, features, interfaces, functional requirements, and non-functional requirements of the system.

The document is intended for the stakeholders of this project that is developers, Apple store administration, developers of the system, more importantly the latter.

# 1.2 Document Conventions

This document follows the IEEE format; bold faced font has been used for emphasis, headings and sub headings. Highlighted words are used in the glossary and italicized text is used in the diagram labelling.

# 1.3 Intended audience and Reading suggestions

This document is intended for all individuals, both the IOS users and IOS mobile applications developers. The intended users mentioned above will be in position to get insights and patterns on the most trending IOS apps based on different features (details) such as size, price, number of supporting devices, number of screen shots for display, etc.

This software is an IOS mobile application analysis system that analyses the IOS mobile application tools to have an insight and determine the patterns from the Apple app store be able to determine which apps are trending according to user rating.

# References

* SRS template 2016
* R package from github, with devtools::install\_github(“ramamet/applestoreR”)

# 1.5 Product scope

The IOS mobile app analysis system is unified for the use of all IOS mobile app users and developers all over the world.

The system is web based. It requires the user to search on any browser of their choice. The user then gets access to the home page of the system which has the systems functionalities: display app and app details,, compare user ratings and app details, compare app statistics for different groups and compare apps in the same group. The system also allows the user to import their data set however; this data set must be from the Apple store. The system basically manipulates the data set to come up with an analysis report in a visualised form.

The system accepts data from the files of previous inventory system and the new data added by the user.

# 2. Overall description

In this part, background information about specific requirements of the system will be provided briefly. General issues that affect the product and outline of the functional requirements will be mentioned, too. In short, this section will mainly give information about product perspective, product functions, constraints, assumptions and dependencies.

# 2.1 Product perspective

The system being developed is a new self-contained software product. It is to be developed for all individuals interested in graphical visualisation. It handles various graph formats and supports them.

It is an open source project and has a very active development team to support it and provide feedback to the intended users.

# 2.2 Product functions

The system is going to perform the different functionalities.

* Compare user ratings and app details.
* Compare app statistics for different groups.
* Compare different apps in the same group.
* Analyses the descriptions for the different apps and determine whether they are positive or negative comments and how it affects user rating.

**THE CONTEXT DIAGRAM OF IOS MOBILE APP ANALYSIS SYSTEM**



figure 1: Context diagram 1

**THE USE CASE DIGRAM FOR THE IOS MOBILE APP ANALYSIS SYSTEM**



figure 2: Use case diagram 1

# 2.3 User classes and characteristics

This software will be used by mostly IOS mobile applications developers in order for them to be able analyse how the different application features affect the user rating. This will eventually enable them to wisely decide on which features to give their applications when developing.

# 2.4 Operating environment

The software will be able to run on any personnel computer and smart phone that is connected to the internet.

# 2.5 Design and implementation constraints

The software will only be developed on Windows 10 laptops. Millions of data will be needed to test the software.

# 2.6 User documentation

User manuals and video tutorials shall be delivered along with the software to enable the user understand how to use the system.

# 3. External interface requirements

# 3.1 User interfaces

The user will be able to access a home page screen that has all the functionalities of the system as shown in the diagram.

**THE HOME PAGE OF THE IOS MOBILE APP ANALYSIS SYSTEM**

# 

figure 4: Home page 1

# 3.2 Hardware interfaces

The system has no hardware interfaces.

# 3.3 Software interfaces

The systems user interface will be integrated with a web browser. The client side gathers information from users, investigates some actions of the users, and provides the connection with the server.

The server side system will hold the entire data in a graph database and must include all functionality to perform operations on this database, receives requests from the clients, evaluate, create and send actions to their commands.

# 3.4 Communication interfaces

The system requires http to communicate with the server. The system can be configured to be accessed via any available port.

The web based UI is the only means of communication between the user and the system. The system is accessible through all popular well browsers that interact with JSP and HTML pages.

# 4. System features

# 4.1 App statistics analysis for different groups

# 4.1.1 Description and policy

This feature allows the user to compare app statistics for different groups.

# 4.1.2 Stimulus and Response sequences

The user opens his or her browser and searches for the IOS mobile app analysis system. The user then clicks the visualisation button. This button has a drop down list with Histogram, Bar plot and Scatter plot. For this system feature the user will select “Bar plot” then a select input widget pops up asking you to choose the horizontal and vertical axis. After that selection, the user clicks “plot” a bar graph will be produced.

# 4.1.3 Functional requirement

**REQ 1**

**NAME:** Comparing app statistics for different app groups

# 4.2 User ratings and app details analysis

# 4.2.1 Description and priority

This feature compares user ratings and the app details.

# 4.2.2 Stimulus/ response sequence

The user opens his or her browser and searches for the IOS mobile app analysis system. The user then clicks the visualisation button. This button has a drop down list with Histogram, Bar plot and Scatter plot. For this system feature the user will select “Histogram” then a select input widget pops up asking you to choose the horizontal and vertical axis. After that selection, the user clicks “plot” a histogram graph will be produced.

# 4.2.2 Functional requirement

**REQ 2**

**NAME**: comparing user ratings and app details.

# 4.3 App comparison in the same group

# 4.3.1 Description and policy

This feature helps the user to compare different apps in the same group

# 4.3.2 Stimulus / Response

The user opens his or her browser and searches for the IOS mobile app analysis system. The user then clicks the visualisation button. This button has a drop down list with Histogram, Bar plot and Scatter plot. For this system feature the user will select “Scatter plot” then a select input widget pops up asking you to choose the horizontal and vertical axis. After that selection, the user clicks “plot” a scatter plot will be produced.

# 4.3.3 Functional requirement

**REQ 3**

**NAME:** Comparing different apps in the same group.

# 4.4 App description analysis

# 4.4.1 Description and priority

The feature analyses the descriptions for the different apps and determines whether they are positive or negative comments.

# 4.4.2 Stimulus / Response

# 4.4.3 Functional Requirement

**REQ 4**

**NAME:** Analysing app descriptions

# 4.5 Information Update

# 4.5.1 Description and policy

This feature enables the administrator to add or delete information from the system.

# 4.5.2 Stimuli/ Response

The administrator clicks “Update” on the menu page and adds any information, or deletes information accordingly.

# 4.5.3 Functional Requirement

**REQ 5**

**NAME**: Update information

**DESC**: The administrator adds or deletes information from the system.

# 5. Other Non-functional Requirements

# 5.1 Performance requirements

The response time is expected to be fast to increase efficiency, with a maximum waiting time of thirty seconds. This prevents the user from assuming that the system is down.

The system will allow many users at a time. The incoming requests will be queued, until there are slots for them to use the system.

The system should have a very low consumption of power.

The system should be able to perform failure handling that is to say if the system components fail independently of others, the system components must be built so they can handle the failure of other components they depend on.

# 5.2 Safety requirements

The system will be integrated with Data Integrity Gateway tools to perform data cleaning to detect, eliminate and correct all errors and inconsistencies.

# 5.3 Security requirements

The system will have different access levels of the different users. It will require a special login for the administrator while he is making updates to prevent access of users who are not authorised to make updates.

# 5.4 Software quality attributes

**Portability**

The system should support new versions of the related browsers. The administrative and server technologies should be standard and supported by most platforms.

**Maintainability**

The system will be well documented to enable proper maintenance and in cases of further development and change of team members, they can still follow up.

**Reliability**

The system should work reliably, with automatic backup and recovery features. In case of unexpected termination of a session, the unsaved data should be recovered without loss and displayed to the respective users.

**Availability**

The entire system should be available round the year, except for a periodic maintenance. The maintenance period should be pre-scheduled and short. The users should be reminded of the unavailability period, well in advance.

**Testability**

The system will be tested with various data sets, and also tested for syntax and semantic errors, which will be debugged to increase efficiency.

# 5.5 Business rules

Only the administrator is responsible for all updates made to the system.

# 6. Appendix: Glossary

|  |  |
| --- | --- |
| Term/ Acronym | Description/ Meaning |
| App | Application |
| IOS | I phone Operating System |
| IEEE | Institution of Electrical and Electronics Engineers |
| REQ | Requirements |
| DESC | Description |