Software Requirements Specification

for

Java Stats

Version 1.0 approved

Prepared by Melroy Dmello

Arizona State University

6th November 2016

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
None	None	None	None

1. Introduction

1.1 Purpose

The purpose of this document is to specify the software requirements of the 'Java Stats' project. This document serves as the sole document for requirements specification and covers the project details in entirety.

1.2 Intended Audience and Reading Suggestions

This document is intended to be read and understood well by the developers developing the project. It also serves as a reference guide for the clients to ascertain if the final product does conform to the requirements mentioned in this document.

1.3 Product Scope

The software being developed should help to assess the volume of any java project. It should display valuable metrics such as the number of classes, interfaces, and lines of code. Highlighting the inheritance relationships between classes should also be supported. This project can then be used by professors/ graders to get a quick overview of a students work as well as by a developer to evaluate his own work

1.4 References

None

2. Overall Description

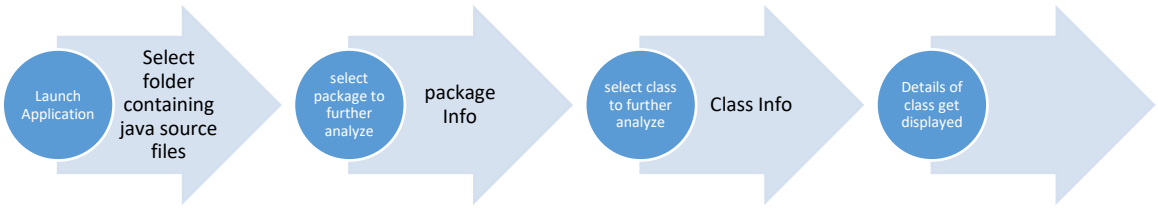
2.1 Product Perspective

This system does not rely on any prior system, nor is it a part of a larger system. It is a standalone application meant to be run independently to analyze a java project

2.2 Product Functions

The sequence of steps which a user can perform to successfully use the application are:

- Select folder containing java source code
- Click a package to view its details
- Click a class to view its details



2.3 User Classes and Characteristics

The different types of users who can use this project are Professors, graders, and developers themselves who need to evaluate if they conform to the project guidelines specified about the volume of a java project

2.4 Operating Environment

The software can work in any system which has java 1.8 installed

2.5 Design and Implementation Constraints

None

2.6 User Documentation

This document combined with the design document serve as all the documents required for developing the project conforming to all the requirements.

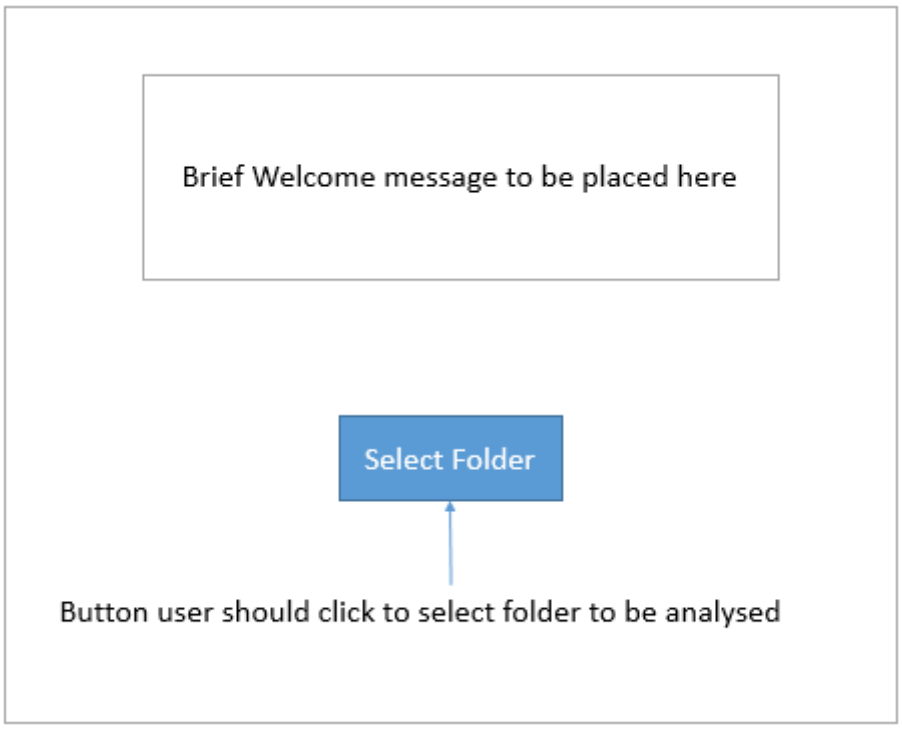
2.7 Assumptions and Dependencies

The software requires java version 1.8 to be installed to run successfully. Also, the person using the application must have access to the Java source files and not just the JAR file. If the above dependencies are met, the application should be able to run successfully.

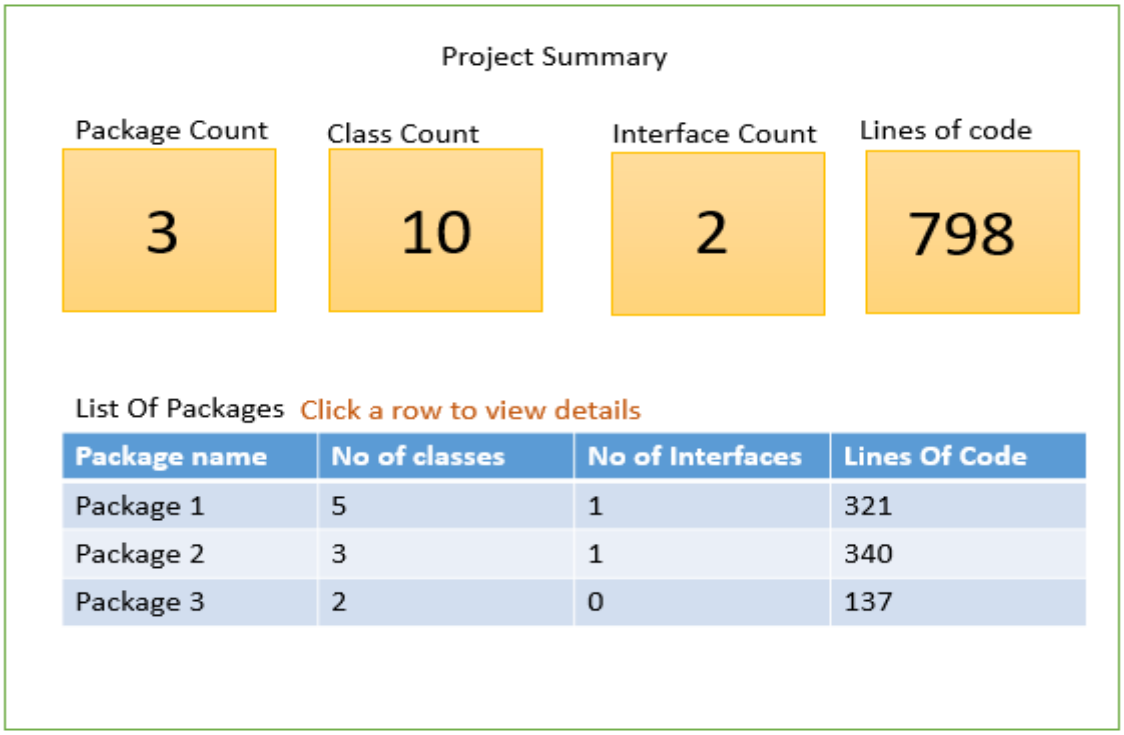
3. External Interface Requirements

3.1 User Interfaces

The application should consist of 4 screens whose layout should appear as below

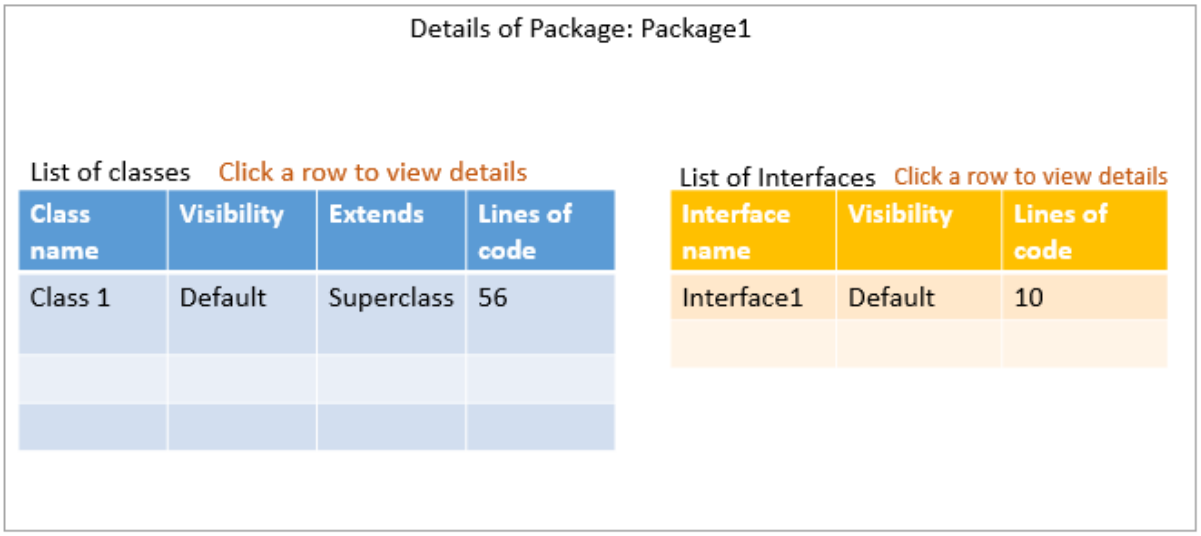


(Screen 1: Welcome Screen)

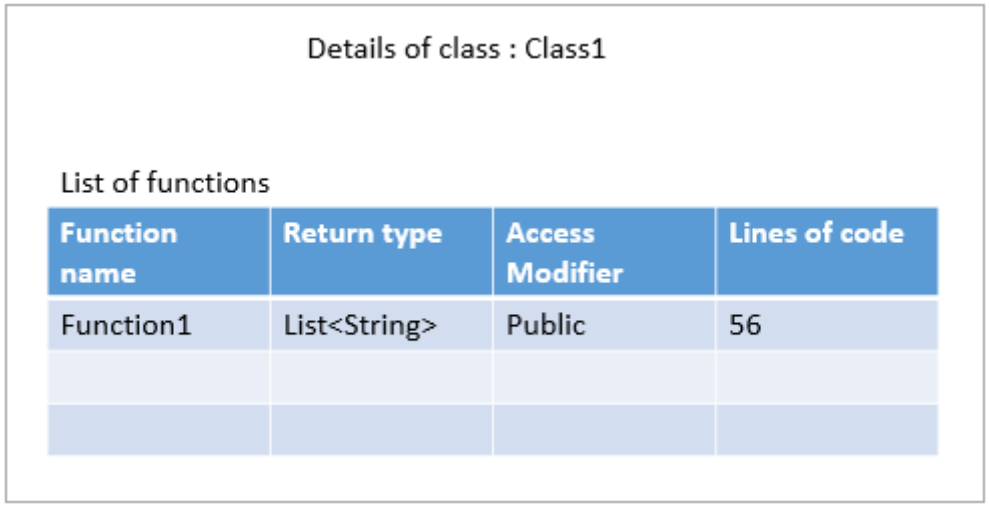


Package name	No of classes	No of Interfaces	Lines Of Code
Package 1	5	1	321
Package 2	3	1	340
Package 3	2	0	137

(Screen 2: Project Summary details)



(Screen 3: Package Details screen)



(Screen 4: Class Details screen)

3.2 Hardware Interfaces

This application requires a standard PC with any operating system running.

3.3 Software Interfaces

This project uses a 3rd party library 'java parser' which is bundled as a JAR file and is included as part of the project to traverse java files to extract information about its functions and inheritance relationships.

3.4 Communications Interfaces

None

4. System Features

The system should support the below mentioned features

4.1 Welcome screen to select the java source code folder

4.1.1 Description and Priority

This screen should be the first screen the user encounters when they open the application. It should contain a brief welcome message along with a button to choose the java source code folder from the file system. Priority: High

4.1.2 Stimulus/Response Sequences

- Launch the application
- Click choose folder button (This should open a file explorer)
- Select the folder containing java source code to be analyzed

4.1.3 Functional Requirements

This requirement serves as the entry point of the application. The welcome screen should open by either double clicking the application or building the project.

4.2 Package Details viewer

4.2.1 Description and Priority

After a user, has selected a package they want to analyze, this screen should show up which would list down the classes and interfaces present in that package

4.2.2 Stimulus/Response Sequences

- Choose a package to analyze further by clicking on it
- View the list of classes and interfaces present in that package and click on any class if it needs to be further analyzed.

4.2.3 Functional Requirements

This requirement serves the purpose of letting a user know the list of classes and interfaces that are contained within a package. It provides information such as the class name, its access modifier, and how it relates to other classes in the inheritance tree. A user should be able to click on any class if they wish to further analyze it by considering its functions.

4.3 Class Details viewer

4.3.1 Description and Priority

After a user, has selected a class they want to analyze further by clicking on it in the package details screen, they should be navigated to the class details viewer screen. Priority: High

4.3.2 Stimulus/Response Sequences

- Choose a class to analyze further by clicking on it
- View the list of functions present in that class. Details of the functions such as its return type, signature, arguments, and lines of code should be mentioned.

4.3.3 Functional Requirements

This requirement serves the purpose of letting a user know the list of functions that are contained within a class. It provides information such as the functions signature, its return type and the number of lines it contains.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The performance of the system should be optimal even while analyzing java projects of extremely large volume spanning over 10000 lines of code. The system should not crash and should provide instant feedback as the user interacts and navigates through the application.

5.2 Safety Requirements

The application in no way should try and alter the structure or the code of the application it is trying to analyze. Apart from that, since this is a standalone application with no database connectivity, there is no risk for potential loss of data.

5.3 Security Requirements

It is not intended for this application should be extremely secure, rather it is meant to be distributed as an open source application. It contains no critical information and hence has low security requirements.

5.4 Software Quality Attributes

The application should run successfully on any operating system and should require minimal user learning. The user interface should be self-explanatory. The code should be scalable to include more metrics on which a java project can be further analyzed

5.5 Business Rules

The system should be accessible by any user and should be navigated only in the intended order, the user should not be able to navigate to a screen without going through the actual entry point for the screen.