



Software Requirements Specification

for

A5-<ProjectTitle>

Version <X.X>

Prepared by

Team Number: <nn>

<name of Team Lead first>

<Roll #>

<name>

<Roll #>

<name>

<Roll #>

<name>

<Roll #>

<name>

<Roll #>

Project Owner: <place your allotted TA's name here>

Course: CS4096D Software Engineering Laboratory

Date: <place the date of submission here>

This template is based on the one available from GMU site by Dr. Rob Pettit.

Modifications specific to NITC are made and will be used for academic purpose only.

1 Introduction	3
1.1 Document Purpose	3
1.2 Product Scope	3
1.3 Intended Audience and Document Overview	3
1.4 Definitions, Acronyms and Abbreviations	3
1.5 Document Conventions	3
1.6 References and Acknowledgments	4
2 Overall Description	4
2.1 Product Overview	4
2.2 Product Functionality	4
2.3 Design and Implementation Constraints	4
2.4 Assumptions and Dependencies	5
3 Specific Requirements	5
3.1 External Interface Requirements	5
3.1.1 User Interfaces	5
3.1.2 Hardware Interfaces	5
3.1.3 Software Interfaces	5
3.2 Functional Requirements	6
3.3 Use Case Model	6
3.3.1 Use Case #1 (use case name and unique identifier – e.g. U1)	6
3.3.2 Use Case #2	7
4 Other Non-functional Requirements	7
4.1 Performance Requirements	7
4.2 Safety and Security Requirements	7
4.3 Software Quality Attributes	8
5 Other Requirements	8
Appendix A - Activity Log	8

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	Names of all team members	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	00/00/00

1 Introduction

<TO DO: Please provide a brief introduction to your project and a brief overview of what the reader will find in this section.>

1.1 Document Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

TO DO: Write 1-2 paragraphs describing the purpose of this document as explained above.>

1.2 Product Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals.>

TO DO: 1-2 paragraphs describing the scope of the product. Make sure to describe the benefits associated with the product.>

1.3 Intended Audience and Document Overview

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized.>

1.4 Definitions, Acronyms and Abbreviations

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations.>

TO DO: Please provide a list of all abbreviations and acronyms used in this document sorted in alphabetical order.>

1.5 Document Conventions

<In general this document follows the IEEE formatting requirements. Use Arial font size 11, or 12 throughout the document for text. Use italics for comments. Document text should be single spaced and maintain the 1" margins found in this template. For Section and Subsection titles please follow the template.>

TO DO: Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. Sometimes, it is useful to divide this section into several sections, e.g., Formatting Conventions, Naming Conventions, etc.>

1.6 References and Acknowledgments

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. >

2 Overall Description

2.1 Product Overview

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. In this part, make sure to include a simple diagram that shows the major components of the overall system, subsystem interconnections, and external interface. In this section it is crucial that you will be creative and provide as much information as possible.

TO DO: Provide at least one paragraph describing product perspective. Provide a general diagram that will illustrate how your product interacts with the rest of the environment and in what context it is being used. This is not a formal diagram, but rather something that is used to illustrate the product at a high level. You may draw this diagram using any online tool>

2.2 Product Functionality

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary is needed here. These can be at the level given in the project description.>

TO DO:

Provide a bulleted list of all the major functions of the system. No need to explain them.

2.3 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software). You can be creative here to some degree. As you know, we will be using Android Studio for development and will follow SDLC. You may mention them here.>

2.4 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project. If you are using a location based project, for example, you may decide to depend on google maps. This should come here.

TO DO: Provide a short list of some major assumptions that might significantly affect your design.>

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users.

.

TO DO: Provide the GUI of the app and state how users are expected to interact with it>

3.1.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware. You are not required to specify what protocols you will be using to communicate with the hardware, but it will be usually included in this part as well.

TO DO: Since we do not intend to use sensors or other typical hardware, we may not need this. But if the mess management project people would like to have an interface with the finger print scanners that needs to be mentioned here.>

3.1.3 Software Interfaces

<Describe the connections between this product and other specific software components. If yours is a standalone app there is nothing to include here >

3.2 Functional Requirements

< Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform. This section is the direct continuation of section 2.2 where you have specified the general functional requirements. Here, you should list in detail the different product functions. It is normal to have 10-15 requirements for each of your projects.

F1: The system shall ... **<Functional Requirement or Feature #1>**

F2: The system shall ... **<Functional Requirement or Feature #2>**

...

3.3 Use Case Model

TO DO: Provide a use case diagram that will encapsulate the entire system and all actors. Please follow UML conventions strictly. You need to study about Use Case Modeling before jumping to draw one. Depending on your team number, you will be using either StartUML or PlantUML

3.3.1 Use Case #1 (use case name and unique identifier – e.g. U1)

TO DO: Provide a specification for each use case diagram. Please refer to similar documents before proceeding to fill this up.

Author – Identify team member who wrote this use case. We expect each team member to have at least 1 use case.

Purpose - What is the basic objective of the use-case. What is it trying to achieve?

Requirements Traceability – Identify all requirements traced to this use case - the Fn numbers from Section 3.2 above

Priority - What is the priority. Low, Medium, High. Importance of this use case being completed and functioning properly when system is deployed

Preconditions - Any condition that must be satisfied before the use case begins

Post conditions - The conditions that will be satisfied after the use case successfully completes

<Please be careful while filling up Pre and Post conditions>

Actors – Actors (human, system, devices, etc.) that trigger the use case to execute or provide input to the use case

Extends – If this is an extension use case, identify which use case(s) it extends <Study what “extends” actually means before proceeding>

Flow of Events

1. Basic Flow - flow of events normally executed in the use-case
2. Alternative Flow - a secondary flow of events due to infrequent conditions
3. Exceptions - Exceptions that may happen during the execution of the use case

Includes (other use case IDs)

Notes/Issues - Any relevant notes or issues that need to be resolved

3.3.2 Use Case #2

...

4 Other Non-functional Requirements

4.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.

TODO: Provide performance requirements based on the information your idea. You may also take opinion of your project owner for this>

4.2 Safety and Security Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be

satisfied. Specify any requirements regarding security or privacy issues surrounding the use of the product or protection of the data used or created by the product. Define any user identity authentication requirements.

TODO:

- Provide safety/security requirements based on your idea of the App. The owner may help.

4.3 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.

TODO: Use subsections (e.g., 4.3.1 Reliability, 4.3.2 Adaptability, etc...) provide requirements related to the different software quality attributes. Make sure that you do not just write "This software shall be maintainable..." Indicate how you plan to achieve it, & etc.... Please note that you need to include **at least 2** quality attributes. You can Google for examples that may pertain to your system. Please do not define what reliability, portability etc. in your document. How these attributes are specified for your App is what is required.>

5 Other Requirements

<This section is **Optional**. Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project. I am OK if you leave this with just NIL>

Appendix A - Activity Log

<Provide details of group meetings - when you met and for how long - including the meeting details with the owner. You must also state what was the contribution (the sections mainly, then diagrams) of each of the team members. Team Lead will have complete responsibility and freedom to complete the Activity Log.>