Software Requirements Specification

for

One Remote Program

Version 1.0 approved

Prepared by LinhTT

One Remote Group

10/5/2015

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Scope 1

1.3 Intended Audience and Reading Suggestions 1

1.4 References 1

1.5 Overview 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

This section gives a scope description and overview of everything included in this SRS document. The aim of this document is to gather, analyze, and give an in-depth insight of the complete **One Remote software** by defining the problem statement in detail. Nevertheless, it also concentrates on the capabilities required by stakeholders and their needs while defining high-level product features. The detailed requirements of the **One Remote software** are provided in this document.

## Purpose

The purpose of this document is to give a detailed description of the requirements for the “One Remote” (OR) software. It will illustrate the purpose and complete declaration for the  
development of system. It will also explain system constraints, interface and interactions with other  
external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

The intended audience of this document is all of the stakeholders for a project involving the development of elevator controller software. This includes, but is not limited to, software developers, project managers, quality assurance personnel, and customers.

## Scope

The “One Remote ” is a client – server mobile application which helps people to remotely control computer. This application should be free to download from either a mobile phone application store or similar services

Application owners can do many things to their computer through an mobile phone or smart phone.

They can do:

* Shutdown their computer
* Hibernate their computer
* Restart their computer
* Log off their computer
* Take photos

Furthermore, the software needs Internet connection to fetch and display results. The software also interacts with the firebase.com (which we do not need to be installed in my program). By using firebase.com, client only have to send instructions to firebase and server only have to get information from firebase.

## Intended Audience and Reading Suggestions

<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. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

## References

[1] Software Requirements Specification – Amazing Lunch Indicator

[2] Srs\_template-1

## Overviews

The remainder of this document includes three chapters and appendixes. The second one provides an overview of the system functionality and system interaction with other systems. This chapter also introduces different types of stakeholders and their interaction with the system. Further, the chapter also mentions the system constraints and assumptions about the product.

The third chapter provides the requirements specification in detailed terms and a description of the  
different system interfaces. Different specification techniques are used in order to specify the  
requirements more precisely for different audiences.

The fourth chapter deals with the prioritization of the requirements. It includes a motivation for the  
chosen prioritization methods and discusses why other alternatives were not chosen.

The Appendixes in the end of the document include the all results of the requirement prioritization and a release plan based on them.

# Overall Description

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented

## Product Perspective

This system will consist of two parts: one mobile (client) application and one computer(server) application. The mobile application will be used to send commands to server through firebase.com and the computer application will be used to receive and complete commands from client through firebase.com

Since this is a data-centric product it will need somewhere to store the data. For that, a database will be used. Both the mobile application and computer application will communicate with the database, however in a slightly different ways. The mobile application will only use the database to get data while the web portal will also add and modify data. All of the database communication will go over the Internet (google picasa) in Figure 1

The mobile application has some restrictions about the resource allocation. To avoid problems with overloading the operating system the application is only allowed to use 2MB of memory while running the application.

Another is that we use firebase.com to play an intermediary role in our program. The mobile application will send commands to firebase.com and the computer application will receive commands from firebase.com

Get Data

Add data

Server

Client

Database

Figure 1

## Product Functions

With the mobile application, the users will be able to control the computer from remote area (not too far). The result will be based on criteria the user inputs. There are several criteria and it will be possible for the administrator of the system to manage the options for those criteria that have that

The result of the remote control will be viewed in the computer you are remotely controlling. After you have sent a command, you will receive a message saying that “Send Successfully”

## User Classes and Characteristics

There is one type of user that interact with the system: users of the mobile application

The mobile application users can use the application to remotely control the computer. This means that the users can be able to shutdown, restart or take photos from a very far place

## Operating Environment

We use junit to code our computer (server) application. As a result, you have to install Android and IOS Operating System before installing our computer application.

In mobile application, we only have version for android smart phone. Consequently, if you want to use our application, you have to install our mobile application in an android smart phone.

## Constraints

The Internet Connection is a big constraint for the application. Since the application fetches data from the database over the Internet, it is crucial that there is an Internet connection for the application to function

## Apportioning of requirements

In the case that the project is delayed, there are some requirements that could be transferred to the next version of the application. Those requirements are to be developed in the third release, see Appendix IV.

## 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, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

# External Interface Requirements

This section contains all of the functional and quality requirements of the system. It gives a detailed description of the system and all its features

## User Interfaces

A first-time user of the mobile application should see log-in page when he/she opens the application, see Figure 2. If the user has not registered, he/she should be able to do that on the log-in page

If the user is not a first-time user and he/she click ‘Remember Me’ button the last time before, he/she can be able to see his/her email and password when the application is opened, see Figure 3. Here the user log into the main menu

If he/she wants to create new accounts, they click ‘’ in Figure 2 and see Figure 4.

In figure 5, you see the result after log in successful is main-menu page. You see ToggleLeft button and our functions (now we have three main functions)

If you click Toggleleft, you see devices and log out. You can choose devices (server – computer you want to remotely control). In this figure 6, you see no device because at the moment we do not log this account into any computer.

If you click any of our functions, for example shutdown options, you can see figure 7, or camera options, see figure 8.

After you choose shutdown options, you can choose one of our four tasks. For example, shutdown and choose timer: 60 and click “send request”. If the request has been sent successfully, you can see figure 9.

This is everything in our mobile application, next is our computer application.

If this is the first time you use our computer application, you should see log-in page when he/shes opens the application, see figure 10.

After you have logged in successfully, you can see figure 11.

## Hardware Interfaces

Since neither the mobile application nor computer application have any designated hardware, it does not have any direct hardware interfaces.

## Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

The mobile application communicates with the firebase.com in order to send user commands to computer application, see Figure 1.

The communication between the firebase.com and the computer application consists of operation concerning both reading and modifying data, while the communication between the firebase.com and the mobile application consists of only sending user commands

## Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems for both mobile(client) and computer(server) application

# System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## System Feature 1

<Don’t really say “System Feature 1.” State the feature name in just a few words.>

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## System Feature 2 (and so on)

# Other Nonfunctional Requirements

## 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.>

## Safety 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.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## 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.>

## Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

# Other Requirements

<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.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>

Source: http://www.frontiernet.net/~kwiegers/process\_assets/srs\_template.doc