

Software Requirements Specification

for

GroupScheduleWebApp

Version 0.001

Prepared by

Group Name: <*place your group name here*>

|  |  |  |
| --- | --- | --- |
| Skylar Cruz | #11619299 | Skylar.cruz@wsu.edu |
| Alberto Clara | #11603391 | alberto.clara@wsu.edu |
| Fabiane Yeung | #11625240 | Fabiane.yeung@wsu.edu |
| <name> | <student #> | <e-mail> |
| <name> | <student #> | <e-mail> |

|  |  |
| --- | --- |
|  |  |
| Date: | October 17th, 2018 |
|  |  |
|  |  |
|  |  |

Content

Revisions iii

1 Introduction 1

1.1 Document Purpose 1

1.2 Product Scope 1

1.3 Intended Audience and Document Overview 1

1.4 Definitions, Acronyms and Abbreviations 1

1.5 Document Conventions 1

1.6 References and Acknowledgments 2

2 Overall Description 3

2.1 Product Perspective 3

2.2 Product Functionality 3

2.3 Users and Characteristics 3

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 4

2.6 User Documentation 4

2.7 Assumptions and Dependencies 4

3 Specific Requirements 5

3.1 External Interface Requirements 5

3.2 Functional Requirements 6

3.3 Behaviour Requirements 6

4 Other Non-functional Requirements 7

4.1 Performance Requirements 7

4.2 Safety and Security Requirements 7

4.3 Software Quality Attributes 7

5 Other Requirements 8

Appendix A – Data Dictionary 9

Appendix B - Group Log 10

Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| Draft Type and Number | Full Name | 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 |

# 

# *<In this template you will find text bounded by the “<>” symbols. This text appears in italics and is intended to provide explanations and guide you through the document. There are two types of comments in this document. The comments that are in black are intended specifically for the course. The comments that are in blue are more general and apply to any SRS. Please make sure to delete all of the comments before submitting the document**.>*

# Introduction

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

## Document Purpose

The purpose of this document is to describe the requirements of the GroupSchedule web application. This includes system requirements such as hardware and software as well as the functionality of the application. It explains the use by its intended audience in detail. This document is intended to be used as a proposal for developing initial versions.

## 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.>

The GroupSchedule has the purpose of gathering the information of the group members and display their common availability. The objective of this software is to help the users with time management and our goal is to present a friendly interface where any individual with basic knowledge in computer can use.

## Intended Audience and Document Overview

Professor:

An evaluation for feedback to improve the document.

User:

This document was written to easier the use of the program and to provide a better understanding of it. A basic computer knowledge is required to be able to use and/or understand the content in this file. The definitions of acronyms used can be found under section 1.4, a full description of the software and the functionality of it can be located furthermore.

## Definitions, Acronyms and Abbreviations

**GS** = GroupSchedule

## 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 to several sections, e.g., Formatting Conventions, Naming Conventions, etc.>

This SRS document used the following styles:

Headers: Arial Font, Size 14, Black

Text: Arial Font, Size 11, Black.

Any bold letters used in this document signal that it is important for the given section, such as headers.

All the acronyms defined on section 1.4 will be marked in in bold and use blue color.

## 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.

TO DO: Use the standard IEEE citation guide (attached) for this section.>

# Overall Description

## Product Perspective

<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 environment and in what context it is being used, i.e., context diagram.>

## 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. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, will be effective.

TO DO:

1. Provide a bulleted list of all the major functions of the system

2. **(Optional)** Provide a Data Flow Diagram of the system to show how these functions relate to each other. This is useful when there is a clear sequence for the functions being performed.>

* Create user accounts
* Allow users to create/join groups
* Allow users to create availability schedules
* Allow users to submit availability schedules within a group
* Compare availability of all combinations of users in a group
* Admin privileges given within groups.

## Users and Characteristics

<Identify the various users that you anticipate will use this product. Users may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience.

TO DO:

1. Describe the pertinent characteristics of each user. Certain requirements may pertain only to certain users.

3. Distinguish the most important users for this product from those who are less important to satisfy.>

**System Administrators:**

Developers of the application

* Has access to all users and groups
* Can delete groups/users from the application

**Group Administrators:**

Comprised of important users within groups

* Users that create a group automatically become an administrator of that group
* Users may be promoted to group administrator by the current administrators of a group
* Invite/Remove/Ban privileges within groups
* Inherits all standard user privileges

**Users:**

Any individual may create a user account through the web application

* Create/join multiple groups
* Create availability schedules including master schedules and unique group schedules for each group they belong to

## Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist. In this part, make sure to include a simple diagram that shows the major components of the overall system, subsystem interconnections, and external interface

TO DO: As stated above, in at least one paragraph, describe the environment your system will have to operate in. Make sure to include the minimum platform requirements for your system. >

The program is written in JavaScript and it’s designed to work under browsers such as Google Chrome. The recommended hardware to be used is either a desktop or a laptop but it can also be accessed through mobile devices. Be aware that it could present problems when using those devices as it wasn’t designed to fit them.

## 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).

TO DO: In this section you need to consider all of the information you gathered so far, analyze it and correctly identify relevant constraints.>

* It should work on mobile devices since it’s a web app, but it might have bugs
* The source file of the program will be available for the public so people can modify it
* If we want to publish online we would have to buy a domain
* No password recovery, so if the user loses their account they will have to make a new one

## User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.

TO DO: You will not actually develop any user-manuals, but you need to describe what kind of manuals and what kind of help is needed for the software you will be developing. One paragraph should be sufficient for this section.>

The Web application will include a help page which will provide users information on how to use the application. It will be accessible by all users and show them how to create schedules, join groups, create groups and administrate their groups.

## 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.

TO DO: Provide a short list of some major assumptions that might significantly affect your design. For example, you can assume that your client will have 1, 2 or at most 50 Automated Banking Machines. Every number has a significant effect on the design of your system. >

# Specific Requirements

## External Interface Requirements

### User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., Cancel) that will appear on every screen, error message display standards, and so on. Define the software components for which a user interface is needed.

TO DO: The least you can do for this section is to describe in words the different User Interfaces and the different screens that will be available to the user. Optional: You may also provide an initial Graphical User Interface design (does not have to be final).>

The web application will be an accessible website with several pages for users to access

* A splash page explaining the application and instructions on joining/creating an account
* A sign up/log in page
* An account page listing all currently joined groups and buttons for creating/joining groups
* Group pages which display various schedules for each user and combined schedules

### 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: Please provide a short description of the different hardware interfaces. If you will be using some special libraries to communicate with your software mention them here. In case you have more than one hardware interface divide this section into subsections.>

### Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems (Windows? Linux? Etc…), 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. 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.

TO DO: The previous part illustrates some of the information you would usually include in this part of the SRS document. To make things simpler, you are only required to describe the specific interface with the operating system.>

### 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.

TO DO: Do not go into too much detail, but provide 1-2 paragraphs were you will outline the major communication standards. For example, if you decide to use encryption there is no need to specify the exact encryption standards, but rather, specify the fact that the data will be encrypted and name what standards you consider using. >

All user account information will be encrypted to ensure security for users. Messages between users will only be possible outside of the **GS**.

## 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 with specific explanations regarding every function.*

*TO DO: Break the functional requirements to several functional areas and divide this section into subsections accordingly. Provide a detailed list of all product operations related to these functional areas.*

**Create user accounts**

* Sign up page will provide users the ability to create user accounts
* Users provide an e-mail address and request a user name and password
* Passwords are to be encrypted and e-mail addresses hidden from other users/group admins

**Allow users to create/join groups**

* Users can create groups.
* Groups can be given any name but are also supplied with an ID number
* Users can join groups by searching for the group name and ID number. An administrator must approve any user attempting to join a group.

**Allow users to create availability schedules**

* Users set individual schedules of availability.
* Scheduling is broken down into 30-minute chunks
* Users can set availability for any given day, to repeat weekly, monthly, or annually.
* Multiple user schedules can be created

**Allow users to submit one availability schedule to a group**

* All groups contain a master schedule which combines the schedules of all the users in the group
* Users can submit a single availability schedule to a group
* Any changes applied to the individual schedule will be reflected in the group schedule

**Compare availability of all combinations of users in a group**

* Group pages will display all schedules of individuals in a group
* Master schedule shows the availabilities of the combined schedules
* Availabilities will be shown for any combination of combined users
* Users can toggle user schedules on and off on the master schedule

**Admin privileges given within groups.**

* *Invite users to group*
* *Accept users requesting access to group*
* *Remove users (not administrators) from group*
* *Promote users to administrators of a group*
* *Demote self to user (if another administrator exists in the group)*

## Behavior Requirements

### Use Case View

<A use case defines a goal-oriented set of interactions between external actors and the system under consideration.

TO DO: Provide a use case diagram which shows the entire system and all possible actors. Do not include detailed use case descriptions (these will be needed when you will be working on the Test Plan), but make sure to include a short description of what every use-case is, who are the actors in your diagram.>

# Other Non-functional 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.

TODO: Provide relevant performance requirements based on the information you collected from the client. For example you can say “1. Any transaction will not take more than 10 seconds, etc…>

Comparing availability schedules should be a quick process. Combining schedules within a group and properly displaying combined availabilities should not take longer than a couple of seconds and in most cases will be instantaneous.

## 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 use of the product or protection of the data used or created by the product. Define any user identity authentication requirements.

TODO:

* Provide relevant safety requirements based on your interview with the client or, on your expectation for the product.
* Describe briefly what level of security is expected from this product by your client and provide a bulleted (or numbered) list of the major security requirements.>
* Passwords will be encrypted to protect our users
* An account can only be made under one e-mail

## 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 Portability, etc…) provide requirements related to the different software quality attributes. Base the information you include in these subsections on the material you have learned in the class. Make sure, that you do not just write “This software shall be maintainable…” Indicate how you plan to achieve it, etc.>

# 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.>

Appendix A – Data Dictionary

*<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>*

Appendix B - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist the Teaching Assistant to determine the effort put forth to produce this document>

Sept. 27 2:40pm to 6pm

* Come up with ideas of what we want to do for the project
* Choose and implement the idea
* Sketch the interface of the web application
* Discuss what language we want to use
* Started the SRS document

Oct. 3 11:30am to 12:30pm

* Divide sections between team members
* Review what was done on the document