

Software Requirements Specification

for

Schedule Helper

Version <1.0>

Prepared by

Group Name: Test Team

|  |  |  |
| --- | --- | --- |
| Rawad Bader | 011678766 | rawad.bader@wsu.edu |
| Steven Arbuckle | 011689992 | Steven.arbuckle@wsu.edu |
| Caleb Hooper | 11684910 | caleb.hooper@wsu.edu |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
|  |  |
| Date: | <10/25/2019> |
|  |  |
|  |  |
|  |  |

Contents

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 |

# 

Revisions will be added in the future during development of application.

# Introduction

This document is the Software Requirements Specification (SRS) for the Schedule Helper application to help people find time that work with fridns for projects or get togetjhers.

This document was developed by our team for the Fundamentals of Software Engineering

course offered by the Department of Computer Science at the Washington State University Vancouver.

This document is based on IEEE Std 830-1998 [1].

This project is a tool to help in managing people schedule and published to other team members. It is more useful in a school project team and it organize people free time for meeting to work on specific martial or get together with friends.

## Document Purpose

The purpose of this document is to present a detailed description of the Web scheduler system that helps the member of team to manage their time and availability for work or any other subject they are working on. It will explain the purpose and features of the application, the interfaces of the application, what the application will do to manage the schedule from different people. This application will make tasks easy for people trying to fit manage their schedule for study groups or meetings for school projects or get togethers.

## Product Scope

This software system will be a Web scheduler system will be accesed on a browser that requires a login. By adding the error checking the amount of work and effort will be minimized which make setting the schedule efficient then setting it manually.

More specifically, this system is designed to allow a student to manage and communicate with a group to manage their time accordingly and make their schedule available to view by every member of the team. The application as a mentioned before it let the person what are the spots free to add their schedule. The platform of this web application will be designed mainly in JavaScript and PHP.

## Intended Audience and Document Overview

Our audience for this document will be our Dr. Xinghui Zhao.

## Definitions, Acronyms and Abbreviations

Admin Administrator

HTML Hyper Text Markup Language

JS JavaScript

PM Project Manager

RSC Research, Development, Test and Evaluation Directorate.

SLT Software Lifecycle Tools

SRS Software Requirements Specification

TDD ﻿Test-Driven Development

UML Unified Modeling Language

## References and Acknowledgments

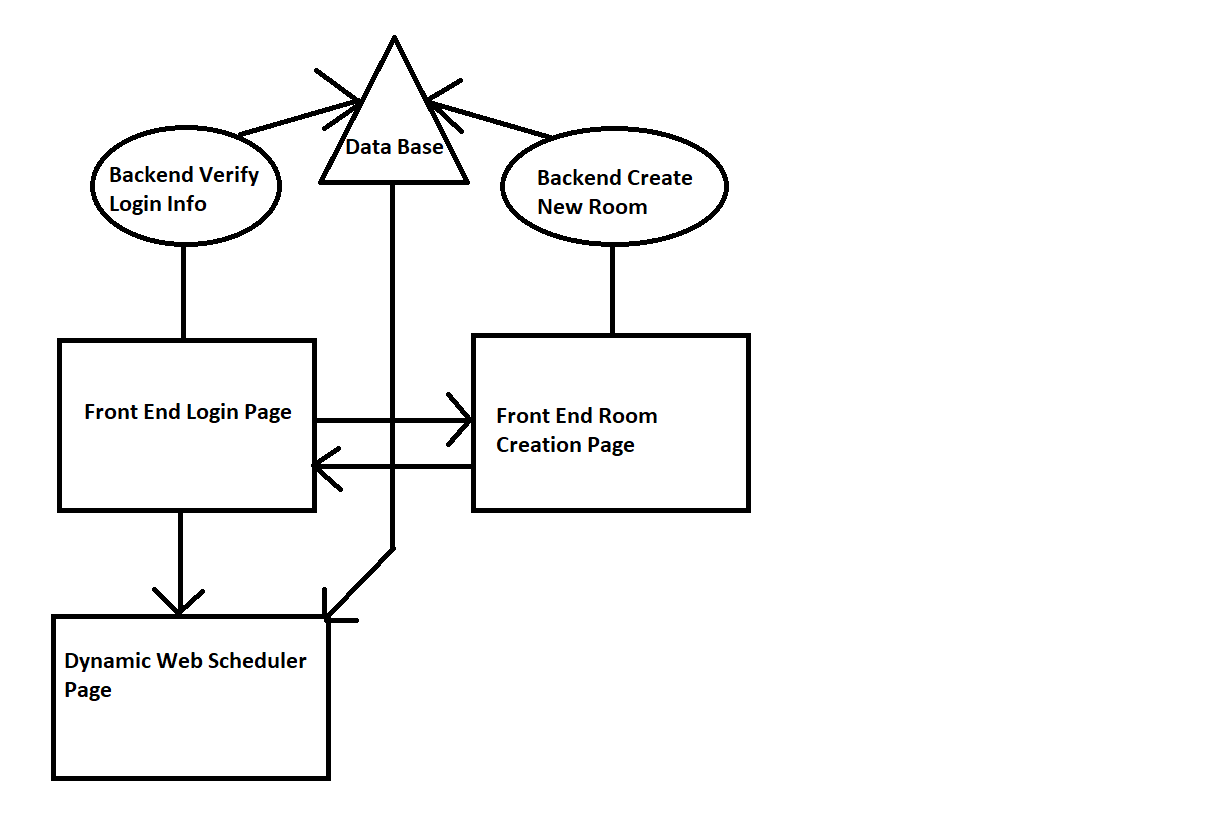
[1] IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications. This document is published by and available from The Institute of Electrical and Electronics Engineers (IEEE).

# Overall Description

## Product Perspective

The scheduler helper will be a self-contained product made from scratch. It will require the following pieces before its operational.

* Front End Login UI
* Front End Room Creation UI
* Dynamic Web Page for Actual Scheduler
* Backend or Logging In
* Backend for Creating A Room
* Database for User Info



## Product Functionality

* Submit your schedule With Excel **style** grid UI
* Display time overlaps with other members of the room and times where no one has overlaps
* Group chat log for everyone in a room.
* Login page for logging into a room
* Room creation page where you can create a new room

## Users and Characteristics

The product will be geared to anyone with a group of friends that may want to get together and do something as a group, but an additional unintended audience could be corporate office workers organizing events or gatherings.

**People with Friend Groups (MOST IMPORTANT TO STATISFY):**

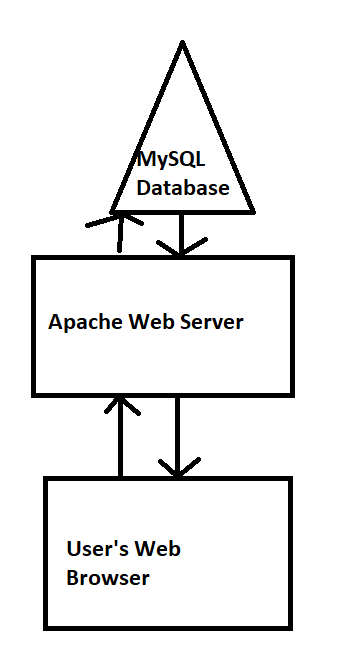
* Low to moderate technical skills
  + Product will need to be user fridley (Simple UI)
* Low on Free Time
  + If user is using a scheduler to find free time with friends, they likely have very little to begin with, so we need to reduce the amount of time spend on the scheduler as much as possible.
* Carefree Environment
  + Users will be more carefree so program could take on a looser theme or tone like other chat programs like (Discord)

**Corporate Office Worker:**

* Moderate to Advance Technical Skills
  + Product Should still be as simple as possible but user base would have fundamentals on how programs are usually laid out and work.
* Professional Environment
  + Cooperate environments are often very professional so we should refrain from covering our program with unprofessional themes or jokes and keep program a having a simple professional tone. For example (Slack).

## Operating Environment

Program will be targeted for web browser on both PC and Smart Phone Devices. The product will involve the following systems.

* MySQL database (User Info)
* Apache Web Server (Web Site Hosting)
* Web Bowser (How User Interact with Program)

## Design and Implementation Constraints

The project will be created using HTML/CSS for front end web development and we will create dynamic webpages using PHP and JavaScript. We will use MySQL as a database for storing user data and we will access the database using PHP. Hardware requirements will most likely not be a limitation on this product as the stuff we are doing is relatively low resource intensive. Security will be a feature we implement but as there is little to no important information we will be storing on our servers security is not a priority. We will create the software and provide source for project and after completion its up to users and community to maintain and add new features to the project.

## User Documentation

The program should be designed user friendly enough as to there is no need or very little need for guidance and direction. Provided documentation and help could be simple an minimum for example tool tips explaining what sections of the site do or a quick brief overview video displayed when creating a new room.

## Assumptions and Dependencies

We will be programing all parts of the website from scratch and not using old code but we will be using open source web servers and database servers. Since we are only using opensource products there will be no legal problems. We will assume that users have either a phone or a PC to access the internet which will mean we need to optimize our front-end UI design to work on all kinds of these devices. For example, the layout of the page needs to work and make intuitive sense no matter what size screen you are using on the product.

# Specific Requirements

## External Interface Requirements

### User Interfaces

The first page the user will see will be for initial connection. This page will have a space for starting a new room and a space for connecting to an existing room. Both spaces will have a space for entering a user-name and a password. The space for connecting to an existing room will also have a space for entering a room-code. The first person to enter the room will be responsible for sharring the room code with the rest of the team.

The second page will be where the information is shared between users. There will be a section for entering availability information into a weekly or monthly calander. There will be a separate section for viewing the availability information of the other users projected onto a similar calander.

### Hardware Interfaces

We will be using a PHP dynamic web server for communicating with the user. The user will submit their login info as well as their request and the server will send back a html file with their desired information on. We will access and modify user data in PHP through MySQL.

### Software Interfaces

Whenever the user changes their schedule their changes will have to be uploaded to a server in order to have that information shared with the rest of the team. Each user can run this program in their web browser, so this product will have to be designed to run on both Mozilla Firefox, Google Chrome, and Internet Explorer. The server side of the software will have to be disigned to run on a Linux machine. We will be using a MySQL 7.3 database for storing user data and we will use PHP for creating a dynamic web page.

### Communications Interfaces

This program will use HTTP protocol as a way to transfer the HTML and javascript interfaces to the users. We will communicate with backend PHP with GET and POST methods. Server will only accept information from clients who submit correct login info.

## Functional Requirements

**3.2.1 Connecting a group**

**3.2.1a Initializing a new group**

* Select an unused group code
* Share that group code with the host user
* Connect the host user to the newly created group

**3.2.1b Connecting to an existing group**

* Get the group code from the connecting user
* Find the group associated with that group code
* Give the user the appropriate information

**3.2.2 Schedule Sharring**

* Add a users schedule to the database
* Congregate all other user’s schedules into group schedule
* Display group schedule to all users

**3.2.3 Text Chat**

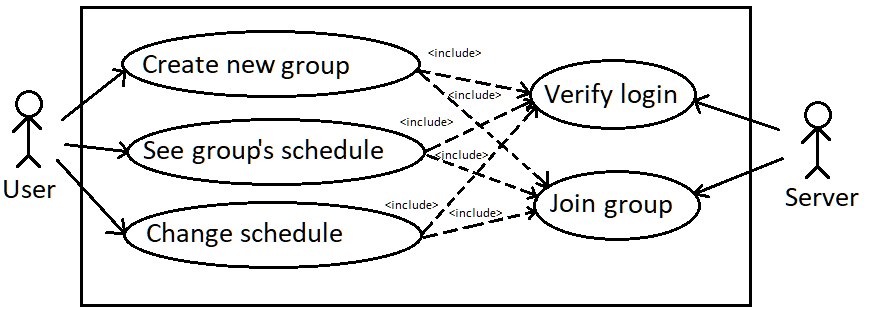
* Send Message
* Receive Message
* Display past messages

**3.2.4 Storing Schedule Data**

* Save message to database when user chooses to apply changes

## Behaviour Requirements

### Use Case View



# Other Non-functional Requirements

## Performance Requirements

All database queries and features from the web page in general should respond quicker than the average human reaction time of .17 seconds. The largest delay will likely be from the user’s internet connection, but we will reduce the response time as much as possible by using efficient database queries and minimal, data efficient web design.

## Safety and Security Requirements

We will implement secure data transfer of user data to our database server but the most important thing we will do to protect the user’s data is not ask anything personal of them. In case of a data breach the user will be protected because we will not store any important data on the user. Just their username password they use for our site and basic schedule information.

## Software Quality Attributes

The most important attribute will be ease-of-use. This app needs to help people connect with their friends, so we need to not waste the users time learning a new piece of software or drive users away with unneeded complexity. The software should be kept reasonably minimal so there is little need for maintain.

It will be important for the user to have access to this software on as many devices as possible so it must run within a web-based environment since almost all devices have a web browser of some kind

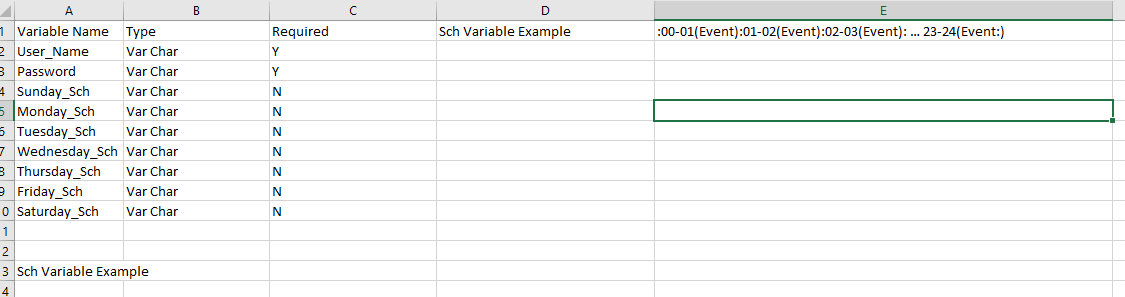
# Other Requirements

## Legal

This software is open source, and anyone has rights to modify and distribute it.

Appendix A – Data Dictionary

This table shows all variables/information we will store on each user in our database. If a variable is not required, the user does not need to provide it to use our system



Appendix B - Group Log

**Group Meeting One:**

**People:** All Group Members

**Time:** 1 hour

**Date:** 10-9-19

**Description:** We decided on group project and flushed out details of what we wanted project to be as a group. Then assigned chunks of the milestone 1 project to each group member.

**Group Meeting Two:**

**People:** All Group Members

**Time:** 1 hour

**Date:** 10-9-19

**Description:** We got development environments setup on individual group member PC’s

**Group Meeting Three:**

**People:** Caleb, Rawad

**Time:** 45 mins

**Date:** 10-9-19

**Description:** Finalized SRS document.