

CS 320 Course Project Final Report

Schedule Helper

Version <2.0>

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: | <place the date of submission here> |
|  |  |
|  |  |
|  |  |

Contents

Contents ii

1 Introduction 1

1.1 Project Overview 1

1.2 Definitions, Acronyms and Abbreviations 1

1.3 References and Acknowledgments 1

2 Design 2

2.1 System Modeling 2

2.2 Interface Design 2

3 Implementation 3

3.1 Development Environment 3

3.2 Task Distribution 3

3.3 Challenges 3

4 Testing 4

4.1 Testing Plan 4

4.2 Tests for Functional Requirements 4

4.3 Tests for Non-functional Requirements 4

4.4 Hardware and Software Requirements 4

5 Analysis 5

6 Conclusion 6

Appendix A - Group Log 7

# 

# Introduction

This document is the Software Requirements Specification (SRS) for the Web Application to help people with their schedule.

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 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 in a specific martial. In this case sharing document & data related to project the team working on it.

## Project Overview

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 system, the interfaces of the system, what the system will do to manage the register entries from a different people, and make sure there is empty space for each request. This development tools will make tasks easy for people trying to fit manage their schedule for study groups or meetings for school related project.

This software system will be a Web scheduler system will be running on a local editor system that requires login administers and proved the code for the people only will be included in the project or work in the same job. This system will have error check system to make sure the people do not pick the similar days in the schedule, which would otherwise have to be performed manually. 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.

## Definitions, Acronyms and Abbreviations

Admin Administrator

EDS ﻿Event-Driven System

HTML Hyper Text Markup Language

JS JavaScript

PM Project Manager

RSC Research, Development, Test and Evaluation Directorate.

SLT Software Lifecycle Tools

SD Sequence diagrams

SRS Software Requirements Specification

TDD ﻿Test-Driven Development

UML Unified Modeling Language

﻿

## References and Acknowledgments

This document is the Software Requirements Specification (SRS) for the Web Application to help people with their schedule.

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

# Design

## System Modeling

< Update your UML diagrams in milestone 2, to reflect the real implementation of this software.

TO DO: Provide an updated version of the UML diagrams, including use case diagrams, sequence (or state) diagrams, activities diagrams, and class diagrams. If you don’t have an updated version, just mention: “our implementation strictly follows the design document (milestone 2)”. >

## Interface Design

<Provide several screenshots to illustrate your interface design.

TO DO:

For each subsystem, pick one or two representative screenshots and paste here.>

# Implementation

## Development Environment

<Describe the devleopment environment you were using for the project.

TO DO: List the programming lanagues, IDEs, tools, etc.>

## Task Distribution

*<Describ how the implementation tasks are distributed among team members.*

*TO DO: For each team member, describe his/her main implementation tasks in this project.*

*If this is a one-person project, mention: “all the work presented here is done by \*\*\* (your name).” >*

## Challenges

*<This section is optional. Describ the challenges in the implementation, if there are any, and how you dealt with them.*

*TO DO: If you don’t have anything to fill in, just leave this section blank.>*

# Testing

## <*This section is a summary of your testing report>*

## Testing Plan

<Describe your testing plan for the project.

TODO: Give a list of items or functions you want to test, and also a schedule for performing the testing. >

## Tests for Functional Requirements

<Describe your test results for the functional requirements.

TODO: Provide a list of use cases or functions you have tested, as well as the testing results (whether or not the system passed the tests).>

## Tests for Non-functional Requirements

<Similar to the Section 4.2, but this section is for the non-functional requirements. >

## Hardware and Software Requirements

<Describe the hardware and software requirements for performing the tests. >

# Analysis

<In this Section you need to analyze the effort that has been put on this project.

TODO: Describe how many hours (approximately) each team member spent on the project, for each milestone, which milestone takes the most effort and why. >

# Conclusion

<Conclude the document with what you have learned through working on the project.>

Appendix A - Group Log

< Describe how frequently the group meembers meet during the semester, and how effective the communication is. This is optional for one-person projects.>