Software Requirement Specification

for

Student One-Click Scheduling

Version 1.0

Prepared By

Derek Hernandez

Brian Pugnali,

Barry Rominger

Ehsan Samani

Valeria Jara Rodriguez

4398\_RR\_1

Table of Contents

[1 Preface 3](#_Toc772762)

[1.1 Introduction 3](#_Toc772763)

[1.2 Purpose 3](#_Toc772764)

[1.3 Document Conventions 3](#_Toc772765)

[1.4 Intended audience 3](#_Toc772766)

[1.5 Proposed Document Scope 3](#_Toc772767)

[1.6 Definitions 3](#_Toc772768)

[2 Overall Description 4](#_Toc772769)

[2.1 Document Perspective 4](#_Toc772770)

[2.2 Document Functions 4](#_Toc772771)

[2.3 User Characteristics 4](#_Toc772772)

[2.4 End-User Operating Environment 4](#_Toc772773)

[2.5 Design and Implementation Constraints 4](#_Toc772774)

[3 Functional Requirements 4](#_Toc772775)

[3.1 Summery 4](#_Toc772776)

[3.2 Use Cases 4](#_Toc772777)

[4 Non Functional Requirements 4](#_Toc772778)

[4.1 Reliability 4](#_Toc772779)

[4.2 Robustness: 4](#_Toc772780)

[4.3 Maintainability 4](#_Toc772781)

[4.4 Security 4](#_Toc772782)

# Preface

## Introduction

This Software Requirements Document (SRS) should be used to define expectations of the auto-scheduler software.

## Purpose

The following is a software requirements specifications document for a university auto-scheduler. The below sections are intended to define the necessary requirements and interactions with the auto-scheduler and the end-user.

## Document Conventions

This document organizes the information into four different sections: Preface, Overall Description, Functional Requirements, and Non-Functional Requirements. The preface discusses this document. The Overall Description describes the software. The Functional requirements outline use cases. And the Non-Functional Requirements section will outline the reliability, robustness, maintainability, and security.

## Intended audience

This document is intended for both the end-users, which are university students, and the software developers of the application.

## Proposed Document Scope

This SRS will cover the operations of the auto-scheduler software and its interactions with the end-user. Topics covered by this document are: handling degree paths, handling degree path requirements, and the application to automatically choose the best schedule for the student.

## Definitions

1. Student Class Level – This defines the level year of the student whether it be freshman, sophomore, junior, or senior.
2. Major – This is the coursework that composes the largest portion of the degree plan chosen by the student.
3. End-user – This is the college student who will use this application to automatically select the best schedule for them.
4. Peak – This is a feature of the software to allow the student to view the automatically chosen schedule without committing to it.
5. Application – For the remainder of this document refers to the auto-scheduler software.
6. Course – This refers to the class assigned to the student.
7. Degree – This will refer to the level and school that the student will be receiving their diploma from. For example, Bachelor of Science.

# Overall Description

## Document Perspective

This document describes the auto-scheduler application and both its use cases and non-functional requirements. The functional requirements describe the process of interaction between a student and the application in generating the best schedule for them automatically.

## Document Functions

This software requirements specification document will serve as the guideline for the software developer and end-user in driving functionality. It will also list the expected interactions and error states as agreed upon for future reference.

## User Characteristics

The user is a typical student of higher learning from an approved institution. They will have a predefined set of courses and prerequisites that apply to their specific college and degree plan. A typical B.S. or B.A. degree is compromised of 120 hours as applied to each student required for graduation. The user must be able to confirm this information is accurate as presented by the application.

## End-User Operating Environment

The user will be presented with a console or graphical user interface that they will use to generate a choice or choices of optimized schedules for their current situation. This console or gui will need to be downloaded to their laptop for use.

## Design and Implementation Constraints

The application will be a java application so the end user will need to have the correct java JRE environment installed on the computer that they are using to run the application.

# Functional Requirements

## Summery

## Use Cases

# Non Functional Requirements

## Reliability

## Robustness:

## Maintainability

## Security