Software Requirements Specification

for

Optimized Course Schedule Generator

**Version 1.0 approved**

**Prepared by**

**Brett M. Stricker**

**Doyle D. Bigelow**

**Diego R. Draguicevich**

**OCSG**

**March 25, 2021**

**Table of Contents**

**Table of Contents ii**

**Revision History ii**

**1. Introduction 1**

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 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** |
| --- | --- | --- | --- |
| First Draft | 03/03/21 | Initial Draft | 1.0 |
|  |  |  |  |

# Introduction

## Purpose

*<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>*

The Optimized Course Schedule Generator is a software tool to assist universities in creating the master schedule for the semester. Schools have to manage thousands of students, and hundreds of teachers and classrooms. The software will generate a master schedule requiring minimal edits decreasing the labor required by a university.

## Document Conventions

*<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>*

The Optimized Course Schedule Generator is hereafter referred to as the OSCG

“Permissions” hereafter refers to the contents of table 5.3.1

“Priority” is listed in descending importance, so first priority is highest, then second, third, and so on

“Minimum Hardware Requirements” hereafter refers to the specifications outlined in section 2.4

## 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.>*

This document is intended to be used in the development, deployment, and maintenance of the OCSG System. Readers might include system developers, system testers, salesmen and other marketing personnel, and university staff deploying this system.

## Product Scope

*<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>*

The aim of this project is to develop a software project that can be utilized by schools and universities to generate a master schedule for class times and locations, easing the jobs of school administration. This software should be able to resolve schedule conflicts including but not exclusive to, classroom timing, classroom location, teacher workload/subject expertise, and classroom student capacity. The software should allow students to easily follow their major flowcharts. This project will be considered a success if less than 10% of the generated schedule must be changed manually.

## References

*<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. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>*

[1] “Advanced Scheduling,” Advanced Scheduling - SchoolInsight - Common Goal Systems, Inc. [Online]. Available: https://www.teacherease.com/advancedscheduling.aspx. [Accessed: 24-Feb-2021].

[2] U. S. A. Scheduler, “Home,” Best Scheduling Software - School Master Schedule Software. [Online]. Available: https://usascheduler.com/index.php/usa-scheduler/class-schedule-maker. [Accessed: 24-Feb-2021].

# 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. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>*

The OSCG is a master schedule generating system built for educational institutions. It is designed as a replacement for existing systems or to be implemented as a new system.

## Product Functions

*<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 (such as a bullet list) 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, is often effective.>*

* Generate master schedule for a school
* add/remove/modify/view buildings and classrooms
* add/remove/modify/view courses
* add/remove/modify/view course sections
* add/remove/modify/view timeslots
* add/remove/modify/view students
* add/remove/modify/view teachers
* add/remove/modify/view flowcharts

diagram?

## User Classes and Characteristics

*<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>*

* Students
* Teachers/Counselor
* Department Chairs

## 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.>*

* The user frontend shall be compatible with the WindowsTM Operating System
* MySQL RDBMS
* Workplace devices running WindowsTM
* Backend shall be compatible with any system running MySQL and Python 2.7.5
* <Minimum Hardware Requirements here>

## Design and Implementation Constraints

*<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; 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).>*

* Generation of the schedule shall take no more than two minutes from the time the user presses the “Generate Master Schedule” button.
* must be secure so that unauthorized modification can not happen
* During operation the system shall hold no more than 4GB of RAM.
* During any given second of system operation the system shall hold no more than 90% of all processing time across all logical cores
* During any given second of system operation on the user’s personal computer the system shall hold no more than 50% of all processing time across all logical cores.
* System shall be developed using Python 2.7.5
* Database Systems shall be designed in MySQL Community Server 8.0.23
* System shall communicate to the database using MySQL Connector/Python 8.0

## 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.>*

* Web based user manual that has documented step based tutorials on how to perform each function.
* A support email to ask technical questions.

## 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).>*

* The server that the system is installed on meets minimum hardware requirements.
* The user PC that the system is installed on meets minimum hardware requirements.
* The system has been installed correctly and has not been modified.
* Server and user PC are connected.

# 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., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>*

## 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, and communication protocols to be used.>*

* Windows Operating System API

## 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.>*

* Python MySQL Connector

## 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.>*

* Server and user PC able to communicate over a TCP/IP interface

# 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.>*

## View MySQL Table

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).>*

A user can view a list of courses, sorting by subject, location, timeslot, teachers, course sections, and flowchart compatibility. Since this is necessary for all other functionality except for Add to MySQL Table, it is second priority.

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.>*

*TBD*

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: The system can display a list of courses based on search results from any specified table column and filter provided the user has permissions

REQ-2: The system can display a list of students based on search results from any specified table column and filter provided the user has permissions

REQ-3: The system can display a list of teachers based on search results from any specified table column and filter provided the user has permissions

REQ-4: The system can display a list of buildings and classrooms based on search results from any specified table column and filter provided the user has permissions

REQ-5: The system can display a list of major flowcharts based on search results from any specified table column and filter provided the user has permissions

REQ-6: The system can display a list of course sections based on search results from any specified table column and filter provided the user has permissions

REQ-7: The system can display a list of timeslots based on search results from any specified table column and filter provided the user has permissions

## Add to MySQL Table

4.2.1 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).>*

A user can insert data into the following tables: courses, course sections, buildings and classrooms, timeslots, students, teachers, and flowcharts. Since initializing the school requires this functionality it is essential for all other functionality it is first priority.

4.2.2 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.>*

* *TBD*

4.2.3 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: The system can add courses into the MySQL Courses Table.

REQ-2: The system can add courses sections into the MySQL Course Sections Table.

REQ-3: The system can add buildings and classrooms into the MySQL Buildings and Classrooms Table.

REQ-4: The system can add time slots into the MySQL Time Slots Table.

REQ-5: The system can add students into the MySQL Students Table.

REQ-6: The system can add teachers into the MySQL Teachers Table.

REQ-7: The system can add flowcharts into the MySQL Flowcharts Table.

## Remove from MySQL Tables

4.3.1 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).>*

A user can remove data from the following tables: courses, course sections, buildings and classrooms, timeslots, students, teachers, and flowcharts. Since removing data is essential for correcting incorrect inputs it is essential for generating the schedule so it is third priority.

4.3.2 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.>*

* *TBD*

4.3.3 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: The system can remove courses from the MySQL Courses Table.

REQ-2: The system can remove courses sections from the MySQL Courses Sections Table.

REQ-3: The system can remove buildings and classrooms from the MySQL Buildings and Classrooms Table.

REQ-4: The system can remove time slots from the MySQL Time Slots Table.

REQ-5: The system can remove students from the MySQL Students Table.

REQ-6: The system can remove teachers from the MySQL Teachers Table.

REQ-7: The system can remove flowcharts from the MySQL Flowcharts Table.

## Generate Master Schedule

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).>*

* Will take all input factors classrooms, timeslots, course sections, projected student enrollment, teacher workload, courses a teacher is able to teach and create a master schedule(s) that fit all criteria. Since errors are permitted and thus must be corrected, add, remove, and view are all prerequisites so it is fourth priority.

4.4.2 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.>*

* once all data is entered into the system for the school, navigate to the top toolbar and select GMS (hovering over will show tooltip Generate master Schedule.
* This will open up a tab in the environment that will have a list of possible schedules labeled by their order (1,2,3,4,5).
  + Each will have the shown student satisfaction (percentage of students in their required classes)
* Clicking on one will bring up a window ***DESCRIBE HOW WINDOW MAY LOOK***

4.4.3 Functional Requirements

*<Itemize the detailed functional requirements associated with this feature. These are 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: The system will have the ability to generate a Master Schedule for a school, based on the data found in the database.

REQ-2: The generated master schedule shall enable 98% of students to take all their courses that semester.

REQ-3: The generation shall create course sections based on course enrollment and classroom timeslots capacities.

## Modify MySQL Tables

4.5.1 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).>*

* A user can modify data in the following tables: courses, course sections, buildings and classrooms, timeslots, students, teachers, and flowcharts. Since modifying tables eases use but is not critical to system function it is fifth priority.

4.5.2 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.>*

* once all data is entered into the system for the school, navigate to the top toolbar and select GMS (hovering over will show tooltip Generate master Schedule.
* This will open up a tab in the environment that will have a list of possible schedules labeled by their order (1,2,3,4,5).
  + Each will have the shown student satisfaction (percentage of students in their required classes)
* Clicking on one will bring up a window ***DESCRIBE HOW WINDOW MAY LOOK***

4.5.3 Functional Requirements

REQ-1: The system can modify data in the MySQL Courses Table.

REQ-2: The system can modify data in the MySQL Course Sections Table.

REQ-3: The system can modify data in the MySQL Buildings and Classrooms Table.

REQ-4: The system can modify data in the MySQL Time Slots Table.

REQ-5: The system can modify data in the MySQL Students Table.

REQ-6: The system can modify data in the MySQL Teachers Table.

REQ-7: The system can modify data in the MySQL Flowcharts Table.

# 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.>*

* The system shall generate the Master Schedule within no more than three minutes of the user requesting a Master Schedule.
* During operation the system shall hold no more than 4GB of RAM.
* During any given second of system operation the system shall hold no more than 90% of all processing time across all logical cores
* During any given second of system operation on the user’s personal computer the system shall hold no more than 50% of all processing time across all logical cores.
* The master schedule generation shall take no more than 3 minutes to generate a schedule from the time the user requests a schedule

## 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.>*

* All tables must maintain atomicity during transactions.
* Users shall not receive permissions without providing login credentials and shall not receive more or less than the specified permissions for their associated credentials
* User login credentials shall not be exposed during login, and shall be made available to view/add/remove/modify only to administrators

## 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.>*

* prevent unauthorized modification of data
* viewership of data is noncritical since most is publicly viewable anyways
  + maybe there is critical data if so what
* Authorization levels: Administration, Teacher, Student

| Table 5.3.1 | Student | Teacher | Administrator |
| --- | --- | --- | --- |
| View Courses | X | X | X |
| View Students |  | X | X |
| View Teachers |  | X | X |
| View Building/Classrooms |  |  | X |
| View Major Flowcharts | X | X | X |
| View Course Sections | X | X | X |
| View Timeslots |  |  | X |
| Add/Remove/Modify MySQL Table |  |  | X |
| Generate Schedule |  |  | X |
| View/Add/Remove/Modify Users |  |  | X |

## 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.>*

* Users are not permitted more than the minimum necessary access to system functions and data

# 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.>*

* Customers shall not redistribute any system component for any reason.
* Support shall not be offered to any customer who has modified their system.
* Any modified system is not guaranteed by the OCSG team.

**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.>*

*4.1.2*

*4.2.2*

*4.3.2*

*4.4.2*

*4.4.5*