

UNCG Schedule Creator: Software Requirements Documentation

Ilyass Sfar, Jamison Valentine, Nick Econopouly, and Nathan Tolodziecki

May 5, 2020

Contents

1	Introduction	2
1.1	Purpose	2
1.2	Document Conventions	2
1.3	Intended Audience	2
1.4	Definitions/Jargon	2
1.5	Project Scope	2
1.6	Technical Challenges	3
1.7	References	3
2	Overall Description	3
2.1	Product Features	3
2.2	User Characteristics	3
2.3	Operating Environment	3
2.4	Design and Implementation Constraints	4
2.5	Assumptions and Dependencies	4
3	Functional Requirements	4
3.1	Primary	4
3.2	Secondary	4
4	Technical Requirements	4
4.1	Operating Systems Compatibility	4
4.2	Interface Requirements	4
4.2.1	User Interface	4
4.2.2	Hardware Interface	4
4.2.3	Software Interface	5
4.2.4	Communication Interface	5
4.3	Non-functional Requirements	5
4.3.1	Performance Requirements	5
4.3.2	Safety/Recovery Requirements	5
4.3.3	Security Requirements	5
4.3.4	Policy Requirements	5
4.3.5	Software Quality Attributes	6
4.3.6	Process Requirements	6

We have abided by the UNCG Academic Integrity Policy on this project.

1 Introduction

1.1 Purpose

The purpose of the Schedule Creator project is to create a tool to help UNCG students visualize class schedules in preparation for class registration. UNCG's current student portal at <https://banweb.uncg.edu> provides a schedule view, but it's only available for classes that have already been registered. Schedule Creator allows students to experiment and view possible schedules without committing to anything.

1.2 Document Conventions

Sections are numbered and summarized for quick navigation in the Table of Contents. Definitions and Jargon are described below for convenience. All external links are written out in full to allow for printing.

1.3 Intended Audience

This document is a reference for current and future stakeholders in the project, including Mr. Ike Quigley (UNCG lecturer and software engineer).

1.4 Definitions/Jargon

Term	Meaning	Notes/Additional Explanation
DB	Database	
JRE	Java Runtime Environment	
CRN	Course Registration Number	5-digit section UUIDs assigned by UNCG
MVC	Model-View-Controller	
API	Application programming interface	
HTTPS	Hypertext Transfer Protocol Secure	
section	Individual course unit, with unique schedule	each section has its own CRN
GUI	Graphical User Interface	
UNCG	University of North Carolina at Greensboro	

1.5 Project Scope

The Schedule Creator is a single-user, local desktop application, and we (the company) will not provide any cloud services. Schedule Creator will:

- allow students to create and visualize real UNCG schedules as described in the overall description (section 2).
- consist of a GUI
- be intuitive enough for most casual computer users to understand
- integrate an external API
- save the user's work automatically via persistent storage

Schedule Creator will *not*:

- provide applications for platforms other than Windows, macOS, and Linux
- integrate with UNCG's registration systems to allow students to register from the application
- provide guidance or recommendations
- maintain a personalized profile of the user other than the saved workspace for each semester (e.g. no record of the user's major or minor)

1.6 Technical Challenges

Schedule Creator posed a number of technical challenges, and each feature described in section 2.1 could be considered a technical challenge. However, the three overarching technical challenges are:

- Cross-platform GUI application development (overcome with the cross-platform JRE and the JavaFX GUI library)
- parsing and translation of UNCG’s course data (overcome with regex-based parsing and Java classes)
- calendar-like schedule display (overcome with a custom calendar coded in a JavaFX grid)

1.7 References

- JavaFX Documentation: <https://docs.oracle.com/javase/8/javafx/api/toc.htm>
- UNCG Registration Guide: <https://reg.uncg.edu/registration/>
- Java Regex Documentation: <https://docs.oracle.com/javase/8/docs/api/index.html?java/util/regex/package-summary.html>

2 Overall Description

2.1 Product Features

- access and switch between multiple UNCG semesters
- search for courses by name, number, or course abbreviation
- view all available sections for each course
- view all possible schedules for the chosen sections in a graphical weekly calendar
- view all CRNs (and associated sections) for the currently displayed schedule
- email CRNs to yourself or your adviser from within the application
- automatically resume work on any semester, even after closing and reopening the program
- toggle a “dark mode” theme for night viewing
- view UNCG’s registration and catalog system from an in-app browser
- [Experimental] manually add new semesters to the program’s database when UNCG releases course information for that semester

2.2 User Characteristics

All users of the program are current students at UNCG who need to register for classes for an upcoming semester. As users of UNCG’s current registration system, they already know how to browse the web, copy/paste, and generally use a computer. Schedule Creator is designed to be straightforward to use for this computer-literate demographic.

2.3 Operating Environment

Schedule Creator is a desktop application that must run on all popular desktop operating systems, because the operating environment is personal computers of UNCG students.

2.4 Design and Implementation Constraints

- the program must integrate an external API
- the program must include a persistent data store
- the program must follow the MVC architecture style when appropriate

Additional considerations:

- the program must be fast enough to run smoothly on a low-end consumer PC

2.5 Assumptions and Dependencies

The user must have the JRE installed on their machine. Additionally, the program includes library dependencies (which will be bundled with the application).

3 Functional Requirements

3.1 Primary

Schedule Creator must allow users to search for UNCG courses, select sections, and view their selections in a weekly calendar. Users must be able to switch to other available semesters. Upon finding a satisfactory schedule, Schedule Creator must include a way to save the CRNs for later use. The program should also include a way to add new semesters when UNCG releases course information.

3.2 Secondary

To search and select courses, the program must include a search bar, selectable lists of courses and sections, and buttons to add and remove courses or sections. To view the multiple possible schedules, there must be buttons for switching weeks. To allow the user to look up classes, and to facilitate the experimental “Add New Semesters” feature, there should be an in-app browser that displays UNCG’s registration and catalog system.

4 Technical Requirements

4.1 Operating Systems Compatibility

The program should support every major operating system run by UNCG students, including Windows, macOS, and Linux.

4.2 Interface Requirements

4.2.1 User Interface

The GUI should support the common UI paradigm of pointing and clicking on buttons and dropdowns with a mouse pointer. This is the most accessible UI style for most users. In addition to the mouse-based navigation, the program should support keyboard navigation for most operations, based on keyboard shortcuts common to web applications (Tab to move to the next field, enter to “click” the current button, Shift+Tab to move to the previous field).

4.2.2 Hardware Interface

There is no hardware interface for this project.

4.2.3 Software Interface

Since Schedule Creator is written entirely in Java, no additional OS-level software interaction is necessary to get full functionality (including reading/writing files, making network calls, and drawing graphics).

4.2.4 Communication Interface

Schedule Creator must communicate with the Mailjet API in order to send CRNs to the user. This functionality is described in the `APIInterface`, which is implemented in the `EmailAPI`.

4.3 Non-functional Requirements

4.3.1 Performance Requirements

Schedule Creator should be able to run smoothly on a low-end consumer PC to accommodate all UNCG students.

4.3.2 Safety/Recovery Requirements

There are multiple failsafes where the user has open ended prompts, to prevent the user from entering or doing something they are not supposed to be doing, and potentially causing problems with the functionality of the program. Emailing CRNs:

- The user is not allowed to access the email portion of the program unless they have a semester and course(s) selected, otherwise there would be nothing to email.
- Once the user has selected a semester and course(s), they are prompted for their email, which is validated, to prevent faulty API calls.

Drag and Drop Database addition:

- The name of the text file that the user drags and drops is validated before its added to the database, to ensure its a valid semester and will be compatible with the database naming conventions.
- The content of the file is checked before it's added to the database, to ensure that there is valid course content in the file.
- The textfiles location is validated, to ensure it was moved from its original location, into the programs resources file, which is necessary to add it to the database.

4.3.3 Security Requirements

There are no user accounts and the information being used and given is not sensitive, the only time a user would be inputting sensitive information is if they choose to register within the application. Which is done through a safe HTTPS website that has its own security and procedures that extend to the application. Therefore there is no implementation of strict security outside of using adequate variable and method access modifiers.

4.3.4 Policy Requirements

Users are expected to not spam the "Email CRNs" feature in order to keep the API-based feature available for all users.

4.3.5 Software Quality Attributes

Availability The application, other than the API and registration website which aren't central to the applications function of making schedules, can work offline. Therefore the availability is extremely high for the user as there are no internet connection requirements to use the base function of the application.

Correctness The use of Unit Testing will ensure that parts of the program such as the EmailAPI don't have unexpected error cases in production.

Reusability The core functionality of the application is designed to be infinitely reusable. The user has the ability to add to the semester database used to make schedules. Based on this the user could use the program indefinitely for future semesters to make schedules, up until UNCG changes its format for course information if ever. It would simply require a few new regular expressions to adapt if that change ever happened.

Portability Since the application was written in Java, any platform that is supported by the JRE is capable of running the application, which makes the application very portable, since the JRE supports most (if not all).

4.3.6 Process Requirements

Methodology/Development Process In order to coordinate the work of four software developers, we set up a central version control repository and gave every developer write access. In order to control the flow of new contributions, we only gave implicit write access to the main "branch" to one developer, who coordinated the various code contributions. For communication, we used email and the comment/topic systems provided by Github (such as Github Issues).

To prevent editor incompatibilities, all developers used the Netbeans IDE to prepare code for submission to the main branch.

Time Constraints This project was constrained to an approximately three month time period, with time available outside of class to coordinate and work on the software.

Cost and Delivery Date The final presentation of the project to shareholders is scheduled for May 7, 2020 at 7:00 PM.