Duck Planner: An Improved Schedule Builder for University of Oregon Students

Annika Huston, Ethan Dinh, Nathan Gong
Web Feet Team
J. Flores (JJF) – 10-10-23 – v1.0

Table of Contents

1 SRS Revision History

This lists every modification to the document and the respective authors. Entries are chronologically ordered.

Date	Author	Description
10-08-2023	ED	Added initial document to repository
10-09-2023	ED	Filled in Sections 2.1 – 2.2
10-10-2023	AH, NG	Filled in Sections 2.3 – 2.6
10-11-2023	AH, ED	Filled in Sections 3 - 4

2 The Concept of Operations

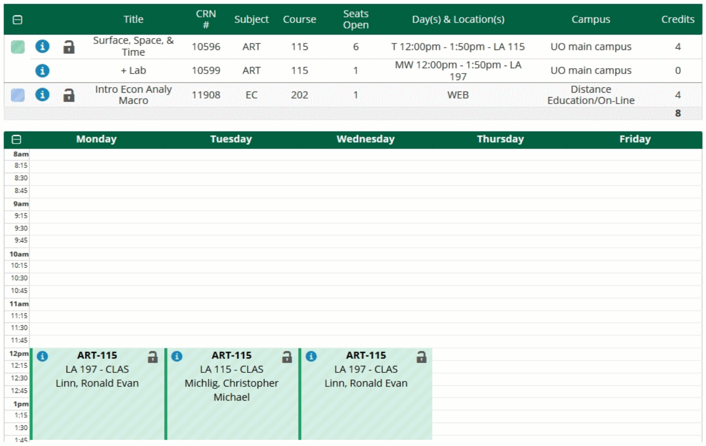
The following section describes our system's characteristics. It provides rationale for our proposed system and outlines the specific capabilities of our implementation when compared to the existing software available for current users.

2.1 Current System

The University of Oregon (UO) has a rich tradition of academic excellence, with thousands of students registering for classes every term. As each term concludes, there's a rising chorus of anticipation, planning, and preparation. Advisors from across departments urge and guide students to chart their academic trajectories for subsequent terms. Currently, when students look to plan their courses, they are met with two choices:

1. *Schedule Builder*

Schedule Builder is the University's proprietary scheduler. It has been designed keeping in mind the specific needs and nuances of UO. While it serves as an intrinsic part of the University's academic program, it possess several limitations. As such, these constraints can hinder students from obtaining a holistic view of available courses or from optimizing their schedules to best fit their individual needs.



⊟			Title	CRN #	Subject	Course	Seats Open	Day(s) & Location(s)	Campus	Credits
🟩	ⓘ	🔒	Surface, Space, & Time	10596	ART	115	6	T 12:00pm - 1:50pm - LA 115	UO main campus	4
	ⓘ		+ Lab	10599	ART	115	1	MW 12:00pm - 1:50pm - LA 197	UO main campus	0
🟦	ⓘ	🔒	Intro Econ Analy Macro	11908	EC	202	1	WEB	Distance Education/On-Line	4
										8

⊟	Monday	Tuesday	Wednesday	Thursday	Friday
8am					
8:15					
8:30					
8:45					
9am					
9:15					
9:30					
9:45					
10am					
10:15					
10:30					
10:45					
11am					
11:15					
11:30					
11:45					
12pm	ⓘ **ART-115** 🔒 LA 197 - CLAS Linn, Ronald Evan	ⓘ **ART-115** 🔒 LA 115 - CLAS Michlig, Christopher Michael	ⓘ **ART-115** 🔒 LA 197 - CLAS Linn, Ronald Evan		
12:15					
12:30					
12:45					
1pm					
1:15					
1:30					
1:45					

Figure 1: Schedule Builder's Frontend Design

2. *DuckWeb's Open Class Search*

Acting as an alternative, DuckWeb's open class search grants students the freedom to peruse through available classes. It's main benefits lie in the platform's pure section format. While this method offers a certain degree of flexibility, it too has its shortcomings. The interface and user experience may not be as streamlined as one would expect for a modern academic platform.

These two primary methods, each bearing their set of pros and cons, represent the pillars upon which UO's current class scheduling rests; but therein lies the issue. Neither solution provides a truly unified, modern, and intuitive platform that addresses all student needs.

2.2 Justification for a New System

The aforementioned shortcomings present in the current UO scheduling infrastructure necessitate an overhaul. Students today need a system that's adaptive, responsive, and multifunctional. Based on class surveys, students often express the desire for features such as advanced filters to narrow down choices based on various criteria. The addition of a map or location display would assist students in understanding the geographical layout of their classes, aiding in efficient route planning bet3ween back-to-back classes. Moreover, a modern interface, reminiscent of the digital platforms students frequently interact with, would enhance the ease of use and overall user experience. Below, we detail the specific advantages and disadvantages present in the UO's current scheduling infrastructure:

Schedule builder enables users to conveniently line up classes for enrollment, and the system then suggests potential schedules accommodating all selected courses (Figure 1). However, issues arise when a student opts for a course with many sections, as they are then faced with hundreds of varying schedules to sort through individually. Despite schedule builder's comparison tool, users are restricted to a view presenting only two possible schedules, which could affect their ability to choose their preferred schedule effectively and efficiently. Moreover, for both incoming first-year students and those pursuing multiple majors, locating new buildings can be tricky. Given that schedule builder only provides abbreviated building codes, students might find it difficult to locate their classes on the first day of school.

Like schedule builder, DuckWeb's open-class search also has certain limitations. This system is rather rudimentary, simply displaying all of the courses in a tabular view using various filters including requirements, dates, majors, or location. While this system does not create schedules, it includes more information than schedule builder. As a result, students frequently employ both systems concurrently for a thorough assurance that their final schedule fulfills their needs.

2.3 Operational Features of the System

Our proposed system plans to enhance the class scheduling process for students by using university class data and presenting schedules in a more user-friendly manner. This section outlines the primary operational features of this system.

1. User Inputted Classes

Students can easily input the classes they plan to take during the upcoming term. This will allow the scheduler to create a schedule based on these inputted classes.

2. Dynamic Filtering Capabilities

To aid in simplifying the scheduling process, there will be a variety of filters to allow students to personalize their potential schedules based on their needs. These filtering options would include most compact schedule, earliest and latest start times, least walking time, etc.

3. Visual Schedule Comparison

Instead of manually selecting the schedules and comparing them individually like in the current scheduler, students will be able to select and view the schedule on the same page. This will aid in simplifying the process to selecting a schedule that best matches their preferences.

2.4 User Classes

Duck Planner primarily serves a singular, specific user class - students attending the University of Oregon. This includes graduate and undergraduate students. The main function of the site is to aid in efficiently planning a class schedule for a single term at UO. Therefore, the exclusive users who require Duck Planner's services are students currently enrolled at the University of Oregon.

2.5 Modes of Operations

Our proposed system has one mode of operation. All students will have access to a login page where they are able to create an account or sign in. They will then have access to our software to create and filter their schedules.

2.6 Operational Scenarios

Use Case 1: Create an Account

Brief Description

This use case describes how a student would be able to start using Duck Planner and create a profile using their 95 number.

Actors: University of Oregon students

Preconditions

- The student has access to the internet.
- The student has a 95 ID number and is currently enrolled in the University of Oregon.

Steps to Complete the Task

1. The user follows the prompt on the initial screen to create a new account.
2. Student will enter their full name, 95 number, and a password.

Postconditions

The user has successfully created an account using their student information. They will now be able to use our Duck Planner and begin creating their future schedules.

Use Case 2: Generate Class Schedules

Brief Description

This use case describes how a student would use an existing Duck Planner account to select their desired classes and create a schedule.

Actors: University of Oregon students

Preconditions

- A pre-existing Duck Planner account.
- The user has access to the internet.
- The user is logged in.

Steps to Complete the Task

1. The student logs in to their Duck Planner account.
2. The student navigates to the left column labeled with Class Selection.
3. The student uses the filter and sort bar to find the classes they want to enroll in. The student may hover over any interesting classes to find further information about them such as possible prerequisite classes, number of credits, and possible labs attached to the class.
4. Once the student has enrolled in all of their desired classes, they click the Generate Schedule

button in the middle column under Schedule Previews.

5. Using the filter bar under the Schedule Previews column, the student searches for the schedule with their desired conditions (Earliest start time, latest end time, least total distance, etc.) and applies the filter to the schedule options.
6. The student may scroll through the Schedule Previews column and click on any schedule image that interests them, displaying the selected schedule more thoroughly on the largest column to the right.
7. If no schedule pleases the student, they can deselect any class(es) and repeat steps 1-6 to find their desired combination of classes that creates a schedule fit for them.
8. Once a schedule looks satisfactory, the student may save the schedule and the combination of classes and times will be saved to their account, viewable for later use or change even after they log out of their account.

Postconditions

The student has selected a schedule and may refer to the connected map and/or schedule at any time to see the details of their class.

3 Specific Requirements

3.1 External Interfaces (Inputs & Outputs)

This section describes inputs into and outputs from the software system. (ISO/IEC/IEEE 29148:2011)

3.1.1 Gathering & Compiling Course Data

Purpose

Employ Selenium to extract data from the University of Oregon's open-class search, then compile it into a CSV file containing all course sections.

Source of input/output

The input into the software is a CSV produced by a Selenium web-scraper. Each row in the CSV contains all course sections for all undergraduate classes at the University of Oregon. The output is a backend success log that notifies admin that the course dataset has been updated within the database.

Valid ranges of input/output

The web-scraper generates data that undergoes a validity assessment prior to its integration into the database. The outcome is a straightforward log message indicating the success or failure of the data insertion.

Units of Measure

There are no units of measure for this external interface.

Data Formats

The data produced from the web-scraper is a comma-separated values (CSV) file.

3.1.2 Creating an Account

Purpose

To enable users to save their proposed schedules, they must create an account on the Duck Planner system.

Source of input/output

The input will be the user's 95 ID number and their chosen password. The output will be a successfully created account on the Duck Planner system.

Valid ranges of input/output

The user's input (ID number) must not already be present in the system. Output should consist of a unique account in the database.

Units of Measure

There are no units of measure for this external interface.

Data Formats

There are no data formats for this external interface.

3.1.3 Selecting Courses

Purpose

To allow users to select their desired courses.

Source of input/output

The user will input the class code for the desired classes. In response, a message will be generated, indicating whether the selected course is valid or not.

Valid ranges of input/output

The class code must be within the database. The output will be a message indicating the validity of the input.

Units of Measure

There are no units of measure for this external interface.

Data Formats

There are no data formats for this external interface.

3.1.4 Selecting Filters

Purpose

To allow users to select from several filters for their proposed schedules.

Source of input/output

The user will select a filter from a dropdown menu. The output will be a list of matrix formatted schedules based on the filtering criteria.

Valid ranges of input/output

The selected filter must be present within the dropdown menu. The output will be displaying the arranged schedules based on the filter.

Units of Measure

There are no units of measure for this external interface.

Data Formats

There are no data formats for this external interface.

3.1.5 CRN Output

Purpose

To generate a list of CRNs based on the user's selected schedule.

Source of input/output

The input will be a user's selection from the list of proposed schedules. The output will be a string containing all of the CRNs present in the user's selection.

Valid ranges of input/output

A decision on the desired schedule must be selected in order to produce a list of appropriate CRNs.

Units of Measure

There are no units of measure for this external interface.

Data Formats

There are no data formats for this external interface.

3.2 Functions

This section defines the actions that must take place in the software to accept and process inputs and generate outputs (ISO/IEC/IEEE 29148:2011).

1. Validity Check on Inputs

There are two primary measures for validating user inputs: login and course selection. In the login process, the program must confirm that no new user is able to create an account using a 95 ID number that's already taken by a registered user. This is a necessary procedure to avoid one user's proposed schedule being accidentally replaced by another user. Regarding course selections, the program needs to verify that the entered user information aligns with a valid course name. This will enable the scheduling algorithm to accurately identify the appropriate course and subsequently arrange the correct sections.

2. Sequence of Operations in Processing Inputs

Backend: Once the web scraper completes scanning through DuckWeb's open-class search, the python backend compiles the data into a CSV and sends the data to the SQL database.

Frontend: Once a user creates an account with the ID number, they will have access to the scheduler. After the user has created an account, they are allowed to select their desired courses to generate schedules. Once the proposed schedules have been generated, the user can select from the various filters. If the user is satisfied with a desired schedule, they can proceed to output the CRNs for registration.

3. Response to abnormal situations

In the event of failed account creation, the user will be prompted to retry typing their 95 ID number and password. If the user selects an invalid course, they will be prompted to retype the class code. Lastly, if no schedules are possible, the user will be prompted to change their desired courses.

4. Relationship of outputs to inputs

Need to hash this section out with the team / Professor Flores. Unsure how we can describe the relationship for our system.

3.3 Usability Requirements

The proposed usability requirements and objectives for our software system, as per the ISO/IEC/IEEE 29148:2011 standard, are as follows:

Effectiveness

The system must be capable of generating all the proposed schedules for a given list of desired courses. Furthermore, the system must be able to compile all of the University of Oregon undergraduate courses offered each term and update them on a per-term basis.

Efficiency

The system is designed to process 100% of schedule generating computations and display matrix after filter application in under one second, thereby ensuring prompt responses.

User Satisfaction

The system aims to be comprehensive and user-friendly, providing detailed course information akin to DuckWeb's open-class search but effectively integrating it with convenient schedule building. This eliminates the need for users to navigate multiple platforms simultaneously. These objectives should be measurable through specific usability tests or data analytics that gauge system effectiveness, efficiency, and user satisfaction levels.

3.4 Performance Requirements

When a user creates their proposed schedules, the system is designed to process 95% of the computations in under one second. Likewise, when filters are used, the matrix display is expected to be visible in under one second.

3.5 Software System Attributes

The Duck Planner system prioritizes a select set of necessary software attributes to create an effective and user-friendly academic scheduling solution.

Reliability

It is imperative that the system consistently functions as intended without inconveniencing the user. To ensure this, we plan to conduct extensive system testing before launch, and regularly scheduled maintenance checks post-launch.

Maintainability

Our system also places an emphasis on maintainability. As university class data and student needs evolve over time, the system should adapt accordingly. To ensure this, we will design the software with a flexible architecture allowing for easy updates and alterations.

Security

Students' personal 95 ID number and password will be used and thus its protection becomes a significant priority. We plan to integrate encryption measures via hashing to ensure the security of user data. Furthermore, none of the users private information will be presented to any other user and stored securely in our database.

Portability

Lastly, the Duck Planner system aims to offer high portability, providing users with a seamless experience across multiple devices and platforms. This demands that the design and development phases consider various operating environments. In doing so, we will adopt responsive web design techniques and perform rigorous cross-platform testing to achieve optimal portability.

4 References

Student Schedule Builder. (2015, October 15). [Text]. Office of Registrar. registrar.uoregon.edu/current-students/schedule-builder