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

1 System Overview

For students at the University of Oregon, the process of selecting and scheduling classes each term can be complicated and time consuming. With the large amount of class options, times offered, and personal scheduling preferences, students need a tool that will simplify the process and cater to their needs. Our intention behind the system is to create a personalized scheduling aid, allowing students to input their desired classes, apply filters, and view schedule variations in a simplified manner.

Duck Planner consists of two main components: a login screen and a scheduling page. The login screen allows users to create a new account or facilitates users access to their personalized dashboard containing previously entered classes and saved schedules. The scheduling and filtering features fetch real-time class data where students can select or modify the classes they plan to take. Once a schedule is chosen, it is saved to their account for future reference. Our system integrates class data with personal preferences while providing an interface between the schedule selection and schedule visualization that saves students time and simplifies their academic planning process.

2 Software Architecture

The following section effectively details and conveys the key design decisions, which include how the system is broken down into components and the relationships between these components (Faulk, 2017).

2.1 Components and Their Functionality

Client Side

1. Create Account/ Login Page

Allows students to create a new account or log into an existing account. First screen users will see when they open the Duck Planner website. After creating an account, they will immediately have access to their scheduling dashboard.

2. Scheduling Dashboard

Class Selection

Interface lets users select classes they are interested in taking. Real-time data is fetched from the University of Oregon's DuckWeb to provide students with the latest information.

Dynamic Filtering

Users can apply various filters to view tailored schedules. This is help aid in simplifying the scheduling process.

Schedule Visualization

Once classes are selected and filters are applied, students are able to view potential schedules on the same dashboard. They also have the option to save schedules to their account.

Server Side

Database

- Account Information
  - 95 number – Proprietary UO ID
  - Password – User Defined
- Class Information
  - Time – 24hr Format
  - Professor – First and last name
  - Location – Building Codes
  - Requirements – Per University Codes
- Saved Schedules
  - Stores schedules that users saved for future use.

Web Scraper

Periodically scrapes class data from the university's website and updates the class information within the database. This ensures that the most current class data is available to users on the client side.

2.2 Module Interaction

In our system, every module has a distinct role, yet they all interact to provide an effective scheduling experience. The modules include:

Authentication Module

This module handles the creation of new user account and verification of existing users. It directly interacts with the Database Module. When a user creates a new account, their information is stored in the database, and when a user logs in, their information fetches and verifies their 95 number and password. After they are verified, students are directed to the Dashboard Module.

Dashboard Module
This is the central hub where users interact with class data and visualize schedules. It communicates with the Database Module to retrieve real-time data and have options to view schedules. It also must communicate with the Database Module when the student wasn't to save a chosen schedule.

Database Module
The module stores and manages all the system data. It serves as the primary data access layer, responding to queries from the Authentication Module and the Dashboard Module. The WebScraping Module updates the Database Module to keep the data up to date.

Scheduling Module

2.3 Rationale for Architecture

The primary objective is to provide students with a platform where they can seamlessly create their academic schedules. The architecture was designed with this objective in mind, with each module serving a specific purpose. The decision to join the log on and create profile on one module was due to the shared common goal of user verification. The unified approach allows us to have a smooth transition from creating an account and logging in.

The Dashboard Module is separate as it is a central hub for all user functionalities. It holds all the class selection, dynamic filtering, and schedule visualization functionalities. Keeping it distinct allows us to easily adjust for enhancements in the future. The Database Module and WebScraping Module are critical back-end components. The Database Module acts as a storage and retrieval system and the WebScraping Module ensure the Database is up to date. Keeping them separate allows us to easily integrate other data sources in the future without any major changes, for example, if we extended our services to other universities.

3 Software Modules

3.1 Authentication Module

3.1.1 Primary Role and Function
This module allows users to create a new account or log in to their existing account. New users are prompted to enter their ID (95 number), full name, and a password. Returning users are prompted to enter their ID and their password. Correct credentials will allow students access to their account, while incorrect credentials will display a message explaining the system is unable to verify their credentials.

3.1.2 Interface
Upon opening the website, users are prompted to enter in their log in information. There is also a button under this screen where the user is able to create a new account. After logging in or creating a new account, users are directed to the Dashboard Module.

3.1.3 Static Model
3.1.4 Dynamic Model
3.1.5 Design Rationale
3.1.6 Alternative Designs
One alternative design was to have separate modules for the login and account creation features. However with the shared goal and overlap in access to the Database Module, it seemed more efficient to combine them into one.

3.2 Dashboard Module

3.2.1 Primary Role and Function
3.2.2 Interface
3.2.3 Static Model
3.2.4 Dynamic Model
3.2.5 Design Rationale
3.2.6 Alternative Designs

3.3 Database Module

3.3.1 Primary Role and Function
3.3.2 Interface
3.3.3 Static Model
3.3.4 Dynamic Model
3.3.5 Design Rationale
3.3.6 Alternative Designs

3.4 Scheduling Module

3.4.1 Primary Role and Function
3.4.2 Interface
3.4.3 Static Model
3.4.4 Dynamic Model
3.4.5 Design Rationale
3.4.6 Alternative Design

4 Architectural Design