

EECS 582 Final Design Document

Team No: 10

Team Members: Rafael Alaras, Drake Prebyl, Tiger Ruan, Reece Mathews, James Krajicek

Project Name: Course Correct

Project Synopsis: An improved version of GradPlan to help EECS students decide which courses to take each semester.

Project Description

We have found that the current GradPlan application for the EECS Department is insufficient on its own in guiding new and existing students on how to lay out their courses into each semester block. In response to this, we created GradPlan Graphic in EECS 448. GradPlan Graphic focused on visualizing course prerequisites and corequisites by adding arrows between the courses to help students understand how courses can and cannot be moved.

Course Correct will take this project further by developing a backend database and hosting for our application. We will provide stateful hosting, allowing for student plans to be saved under their KU login. We will develop an administrative interface, allowing department administrators to create and modify the degrees offered and alter the properties of courses such as prerequisites and seasons offered as the department grows and evolves. Staff will also have the ability to easily browse through student's plans, and students will be able to share plans with their advisors.

We will also make improvements to the user experience. We will incorporate more of the information hidden in the handbook, such as the requirements for senior electives. Error messages will be clarified and the prerequisite/corequisite visualization will be made easier to understand. We will also provide the ability for students to upload an academic summary from Enroll & Pay to automatically fill out completed courses, ensuring only those with a passing grade are counted. These changes will make the enrollment experience better for students and advisors.

Project Milestones

Key Milestones

- Fall 2021
 - Hosting the current project on an Apache web server
 - Implementation of a stateful web app with a MySQL server
 - Creation of an admin frontend UI mockup
- Spring 2022
 - Use the backend database for saving and loading plans
 - Improve existing student UI (including arrow coloring)
 - Develop the admin backend based on KU EECS staff feedback from the FA21 Admin UI mockup
 - Creating and modifying degrees and courses
 - Looking up student plans by IDs

Additional Feature Milestones (time permitting)

- Academic Summary Import
- Analytics (Timestamps, EECS101 List)
- Shareable Plans

Project Budget

- Development/test server hosting for 8 months
 - Needs support for PHP and SQL
 - DigitalOcean Droplet - \$18 per month

Preliminary Project Design

The frontend of the project is a Javascript web application. It will have a backend written with PHP and store data in an SQL database. For development and testing, it will be hosted using an Apache web server on a DigitalOcean Droplet. Students will login to the frontend which will send a request to the backend. The student's stored plans will be pulled from the database and sent to the frontend for viewing and editing by the user. While editing a plan, the software will check course selections against requirements from the selected handbook. Errors such as Upper Level Eligibility or a prereq course not being met will be displayed to the user. These checks will be handled by the Javascript frontend, using information pulled from the backend.

The admin interface will likewise be implemented as a Javascript frontend. This webpage will only be accessible to logged-in admin accounts. The most notable feature of this interface will be the ability for admins to create and modify degrees and handbook versions. These objects will be stored in the database and will be used in determining degree requirements for users. Admins will also be able to view plans created by students.

Page Flow

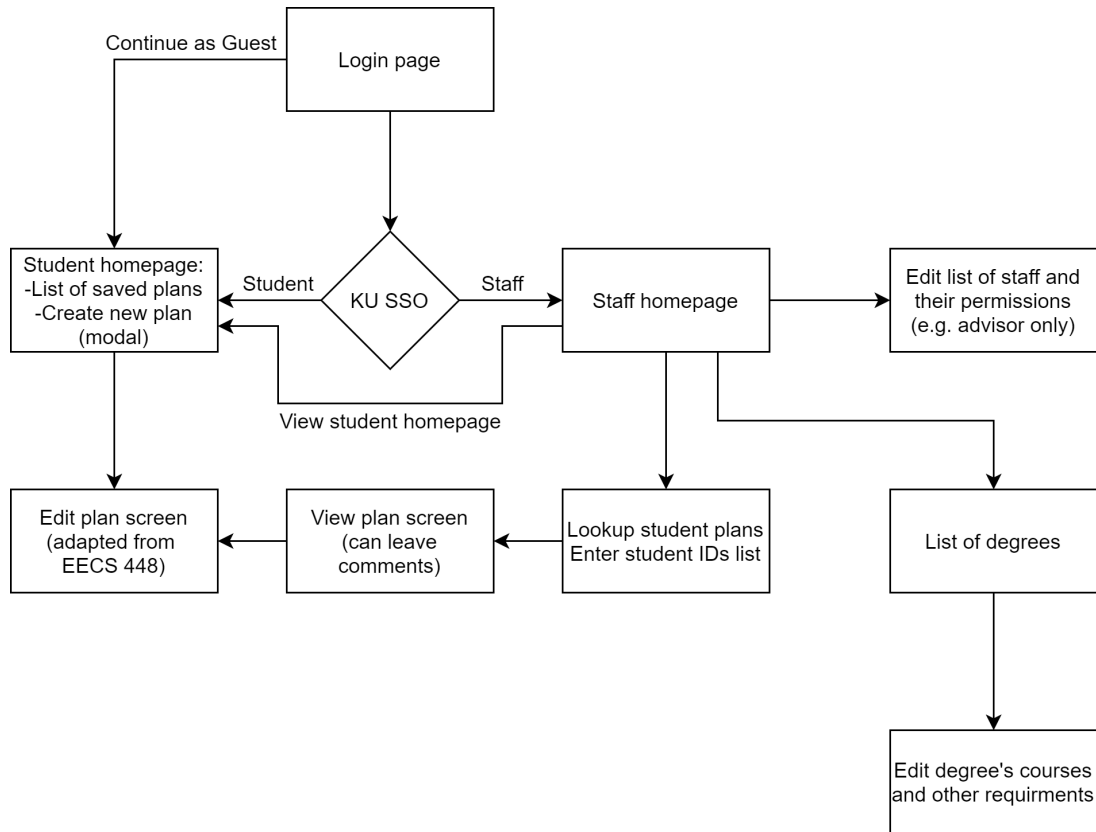


Figure 1: Page flow through the application

Our application starts on the login page. Like the current gradplan, users have the option to continue as a guest but will be unable to save their plans. If the users do login, staff will be taken to a separate page than students. They will also be able to access the student page, for example if they want to show students how to use the application in class. The student page will display a list of saved plans, with the option to view and edit them, and a modal dialog for selecting a major to create a new plan. The plan editing interface will be based on our EECS 448 project, such as including arrows between courses to visualize prerequisites.

On the staff interface, staff will be able to create and edit the degrees offered with a similar interface. New degrees can be created by cloning existing degrees, which is useful for new handbook versions and new flavors of Interdisciplinary Computing. They will also be able to edit the list of staff and their permissions. Lastly, they will be able to look up student's plans by their IDs. This is particularly useful for advisors, who will be able to leave comments on student's plans and approve them.

Database Table Structure

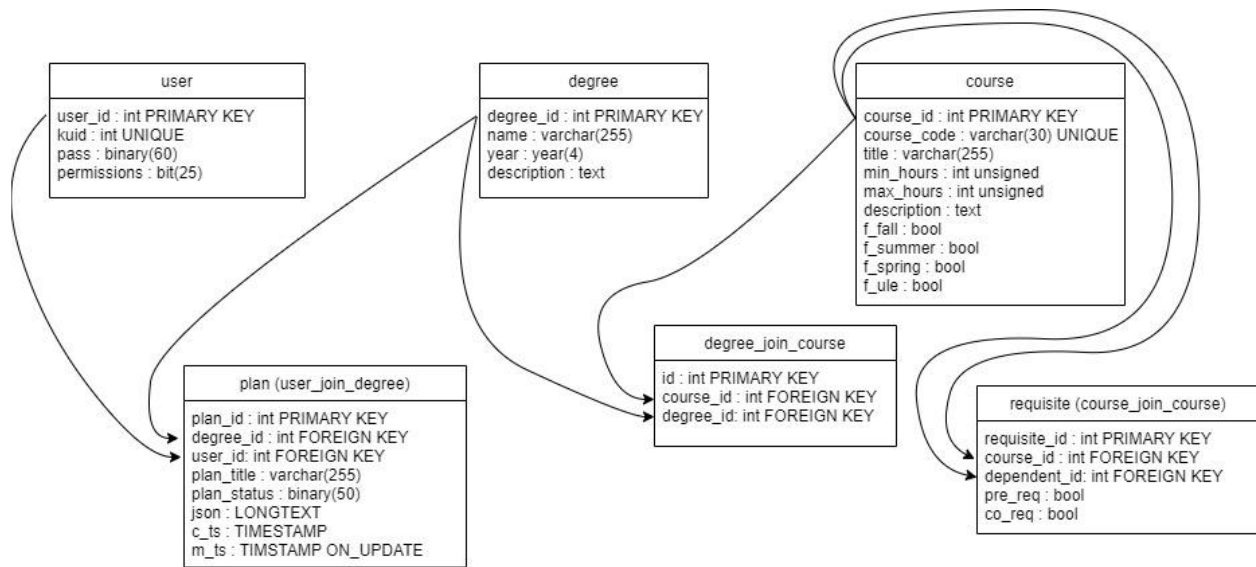


Figure 2 : Database Table Structure

Our database primarily stores information about degrees and the plans students have made with them. Each student can create multiple plans, which contain multiple semesters, which contain multiple courses. Each plan also corresponds to a single degree, which are versioned by year to account for handbook changes, and each degree contains multiple courses. Courses have “requisites”, which are a generalized structure to represent prerequisites and corequisites, and include the time period they are in effect (e.g. the recent addition of EECS 140 as a prerequisite to EECS 210 should not affect old plans).

The database will also store user data, including staff permissions, and a password that will only be used for development and testing (with KU SSO being the long-term plan for authentication). One other table included is for help text, which will allow staff to edit text and links displayed throughout the application to keep them up to date, such as information about the upper-level eligibility requirements or the link to the class search page (which has changed since I started in Fall 2018).

In addition to this high-level view of the database, all objects in the database will be versioned. This means they will also store the date and user they were last modified by, and old versions will be retained in the database. Deleted objects will be marked deleted with a boolean, rather than actually deleted. This will allow students to restore old versions of plans they’ve modified, and allow staff to view the changes that have been made to degrees. This reduces the risk of important information being lost.

Plan Editing Screen Concept

The screenshot displays the CourseCorrect interface for a student named Jayhawk Jayhawk, majoring in Computer Science with student ID 9999999. The main area shows a course plan for EECS 448, organized by semester:

- 2018 Fall (15 credit hours):** MATH 125 (4), GE 2.1(2) (3), GE 2.2 (3), EECS 168 (4) (highlighted), EECS 101 (1).
- 2019 Spring (15 credit hours):** MATH 126 (4), PHSX 210 (3), PHSX 216 (1), EECS 268 (4), Add SS (3).
- 2019 Fall (16 credit hours):** MATH 127 (4), PHSX 212 (3), PHSX 236 (1), EECS 210 (4), EECS 140 (4).
- 2020 Spring (15 credit hours):** MATH 290 (2), EECS 368 (3), MATH 526 (3), AE 4.1 (3), EECS 388 (4).
- 2020 Fall (13 credit hours):** AE 4.2 (3), EECS 448 (4), EECS 510 (3), EECS 645 (3).
- 2021 Spring (15 credit hours):** EECS 678 (4), EECS 665 (4), EECS 560 (4), Sen Elec 1 (3).

Navigation and utility options include: Home, Schedule of Classes, Visual Schedule Builder, Enroll & Pay, myKU, Drake Real Plan as of Fall 2020, Save, Export, and Print.

Figure 3: UI Mockup

This is a concept for our UI based on the EECS 448 GradPlan Graphic project. The main change so far is that the arrows are grayed out by default and become bright red when you hover over a course. Changes like these will make it easier for students to follow the arrows the prerequisites they are interested in. More changes will be made as we work on the project, such as replacing the manual name and student ID entries in the top right with a login system.

Design Constraints

Platforms <ul style="list-style-type: none"> Chromium family <ul style="list-style-type: none"> Chrome, Edge, and Opera Safari Firefox 	Screen sizes <ul style="list-style-type: none"> Desktop PC Laptop Tablet Cell phone
--	--

We want our application to function on common web-browsing capable platforms. To fulfill this goal we will target official application support on two different broad platforms; Desktop and Mobile. Within web platforms we will support the three currently most popular web browsers [according to Tech Advisor](#); Chrome, Safari, and Firefox. Edge and Opera, the two other browsers in this ranking, are powered by the same Chromium rendering engine as Chrome, so they should work as well. Modern mobile browsers use the same rendering engines as our supported desktop browsers. The libraries we are using, such as the Bootstrap framework, should handle the bulk of these devices with few changes necessary explicitly. Another important consideration is that we make sure our application appears clearly on a wide variety

of screen sizes from desktop PCs to smartphones. The Bootstrap framework also allows us to dynamically change the layout of our pages based on screen sizes.

With our EECS 448 project, we had used and implemented some libraries, which are due to change and alter as time passes. The current libraries in the EECS 448 project are; Bootstrap 4, jQuery, REDIPS.drag, SVG.js, Font Awesome, and Flaticon. These libraries are all open source, so usage of the libraries are okay in our situation, but as we go forward we must be diligent on what new libraries are being added as we continue to implement new features.

Ethical Issues

This project has the potential to significantly help or harm students' progress through their degrees based on the accuracy of the information provided. We must ensure that all errors in graduation plans are correctly reported to students so that nothing goes unnoticed until it is too late. This will be done through unit and integration testing of the software, and making the admin interface for setting degree requirements easy to use and understand. We will try to incorporate every requirement from the EECS handbooks, such as Upper Level Eligibility and acceptable senior electives, to ensure students are not caught off-guard by unknown expectations.

This project will be handling sensitive student data, potentially including the grades they received in their courses if we are able to implement degree progress report parsing. We need to ensure that this data is only accessible to those who are authorized to view it. The safest option would be to only use their grades in determining if a class was passed or not, and never store the information permanently. Login information will likely need to be stored as well. Only password hashes and usernames will be stored in this case. The webapp will be hosted with SSL.

Intellectual Property Issues

This project is being created based on an old code base of an EECS 448 project, with 3 members transferred over from the old project to this new project. The other two members, Rodrigo Figuro and Deepak Kumar had helped develop some of the designs and sections of the old 448 project but they aren't going to help develop this project. We must protect these two students' contributions to the project and their Intellectual Property through properly crediting their contributions to what will be the foundation of this project. The limitations and expectations of what should be proper crediting of the precursor project should be discussed within our team and with the two old team members.

Another potential concern for this project is the existing graduation planning application, gradplan.eecs.ku.edu. This application was developed by a previous senior design team and was adopted by the department as the official graduation planner after it was completed. The application does not contain any visible copyright or license notices. Although we do not have access to the source code for the backend or the admin interface, which are the primary areas we will be working for our project, it is still important that we do not copy any of the code we can access from this application, as it belongs to the original developers and/or KU IT, and it may not have an open source license.

Change Log

- The language used for the backend was changed from NodeJS to PHP as this language is easier to work with and used by the existing graduation planner.
- We have elected to use a Raspberry Pi for our development/testing server among our possible options, which was the only item in our project budget. Because it is something we already own, our project budget now does not contain any true expenses, as this is purely a software project.
- Updated Database diagram to reflect implemented database structure.
- We are now using a web hosting site, DigitalOcean, for the project instead of the Raspberry Pi.