Electronic Program of Study

DESIGN DOCUMENT

Team 15
Client: Tina Prouty
Academic Advisor: Maruf Ahamed

Saljooq Altaf (Coordinator/Full-Stack Developer), Nathan Marquardt (Front-End Developer), William Hunt (Back-End Developer), Carson Campbell (Back-End Developer), and Noah Nickel (Back-End Developer)

sddec22-15@iastate.edu
http://sddec22-15.sd.ece.iastate.edu/

Executive Summary

Development Standards & Practices Used

Work competence (SE Code of Ethics 2.01) - Since most of the students working on the project are primarily Software Engineers - their technical expertise are suited to a large scale web-development software that uses complex frameworks like Angular or React (Django for the current iteration).

Financial responsibility (SE Code of Ethics 4.04) - We have made sure that we only make use of open source software and libraries to ensure that the costs to the client is minimal - and have made all financial decisions transparently.

Communication Honesty (SE Code of Ethics 1.04) - There are likely to be a lot of hurdles in implementing the new features that the client has in mind which may not be easily integrated into the project and being honest and transparent about these hurdles is very important for our team to deliver a realistic product with certain limitations that our client understands.

Property Ownership (SE Code of Ethics 2.03) - Since we are being given access to virtual servers and resources for the purpose of creating this product for the client, we have restricted any use of such resources for personal use.

Summary of Requirements

- Supports the creation of a POS through an online application through an easy-to-navigate UI.
- The list of classes selectable should be up to date with what is currently offered by the university.
- A total credits and credits per semester check should be implemented on top of the existing program in order to provide a quicker check on the viability of a student’s plan in the long term.
- A reset button should be implemented that allows the user to re-start the process of creating a POS from scratch.
- A print button should be implemented to print out the POS that a student has created - which should be useful for in-person interactions.
- The prerequisite field should be completed and fully functional, showing a message if a course is selected without its prerequisite.

Applicable Courses from Iowa State University Curriculum

The most directly applicable courses were COM S 309, SE 339, and COM S 363. Other courses have assisted through furthering our understanding of software as a whole.

New Skills/Knowledge acquired that was not taught in courses

- Spring Boot
- Angular
- Self-Organization Skills
- Determining infeasibility of software maintenance

Table of Contents

List of figures/tables/symbols/definitions (This should be the similar to the project plan)

Figures:

Tables:

Definitions:

POS - Program of Study, a student's plan for the classes they will be taking during their enrollment at ISU to fulfill their degree requirements.

1 Team

1.1 Team Members

Our team consists of the following team members:

- Saljooq Altaf
- William Hunt
- Nathan Marquardt
- Carson Campbell
- Noah Nickel

1.2 Required Skill Sets for Your Project

Knowledge of the following coding languages and frameworks will be required:

- Angular
- Java
- SpringBoot
- SQL
- Git
- Bash

1.3 Skill Sets covered by the Team

All members have similar computing knowledge. the appropriate skill sets will be covered based upon the roles within the team. Below is a list of the skills and the team members who will cover these skill sets:

- Angular - Noah Nickel, Nathan Marquardt, Saljooq Altaf
- Java - Saljooq Altaf, William Hunt, Nathan Marquardt, Carson Campbell, Noah Nickel
- SpringBoot - William Hunt, Carson Campbell, Saljooq Altaf
- SQL - William Hunt, Carson Campbell, Saljooq Altaf
- Git - Saljooq Altaf, William Hunt, Nathan Marquardt, Carson Campbell, Noah Nickel
- Bash - Saljooq Altaf, William Hunt, Nathan Marquardt, Carson Campbell, Noah Nickel

1.4 Project Management Style Adopted by the team

Since the requirements are going to be updated regularly, our team has decided to adopt an Agile Management Style. We will have a 3-4 week sprint to incorporate the changes. Additionally, we will have regular weekly standup meetings on Tuesdays.

1.5 Initial Project Management Roles

- Saljooq Altaf - Meeting Coordinator, Code Reviewer, Full-Stack Team Developer
- William Hunt - Code Reviewer, Back-End Team Developer
- Nathan Marquardt - Code Reviewer, Front-End Developer
- Carson Campbell - Code Reviewer, Back-End Developer
- Noah Nickel - Code Reviewer, Front-End Team Developer

2 Introduction

2.1 Problem Statement

Incoming students are required to create a Program of Study which dictates what classes should be taken in the student's 4 years at ISU. The current method that Engineering intro classes use is a spreadsheet that must be manually updated and checked by both the student and advisor. This requires compiling resources from many different online sources that ISU provides which can be confusing and lead to significant problems that the advisors have to address. A better way of organizing this Program of Study would be to compile these resources into one application that can perform automatic checks for the validity of the plan. This would take a lot of pressure off of both the student and advisor as they would not have to compile and hand check the necessary resources.

2.2 Requirements & Constraints

The technical requirements of this project include:

- Supports the creation of a POS through an online application through an easy-to-navigate UI.
- The list of classes selectable should be up to date with what is currently offered by the university.
- A total credits and credits per semester check should be implemented on top of the existing program in order to provide a quicker check on the viability of a student's plan in the long term.
- A reset button should be implemented that allows the user to re-start the process of creating a POS from scratch.
- A print button should be available in order to print out the POS that a student has created - which should be useful for in-person interactions.
- The prerequisite field should be completed and fully functional, showing a message if a course is selected without its prerequisite.

Longer term goals include automating the checks for POS students have created, for instance:

- Some check to see if prerequisites of a selected course are met
- Some check to see if the core requirements of a selected major are met
- Check to see if technical elective requirements are met

Technical Constraints:

- Application is available to students on the internet (so long as they have a browser)
- Application should be able to serve possibly hundreds of students at a time as they are often all completing this plan in class simultaneously.
- Sensitive information like passwords etc. should be encrypted

2.3 Engineering Standards

Work competence (SE Code of Ethics 2.01) - Since most of the students working on the project are primarily Software Engineers - their technical expertise are suited to a large scale web-development software that uses complex frameworks like Angular or React (Django for the current iteration).

Financial responsibility (SE Code of Ethics 4.04) - We have made sure that we only make use of open source software and libraries to ensure that the costs to the client is minimal - and have made all financial decisions transparently.

Communication Honesty (SE Code of Ethics 1.04) - There are likely to be a lot of hurdles in implementing the new features that the client has in mind which may not be easily integrated into the project and being honest and transparent about these hurdles is very important for our team to deliver a realistic product with certain limitations that our client understands.

Property Ownership (SE Code of Ethics 2.03) - Since we are being given access to virtual servers and resources for the purpose of creating this product for the client, we have restricted any use of such resources for personal use.

2.4 Intended Users and Uses

The intended users of the website will be students and advisors. Following are the core use-cases that we will be focusing on:

- The main use for this website will be students creating a new Program of Study to help with planning which classes to take during their college career to meet the graduation requirements. To do this, students will be entering course information through the website and submitting their Program of Study in their “Student View”

- To standardize the input from students, they should have courses that are available from within the web application by a simple search
- Students should be able to see all the core requirements for their selected major
- Students should also be able to see detailed information of the course they selected, including the prerequisites
- Once POS is completed, a student should be able to print their POS,
- Student should be able to save their created POS for future reference
- Student should be able to submit their POS to the Advisor from within the application
- Advisors should be able to view student Programs of Study to assist students with completing required courses. They will be able to access their students’ Program of Study through an “Advisor View” mode to view
- Advisors should be able to reference current Programs of Study through appointment conversations with students

3 Project Plan

3.1 Project Management/Tracking Procedures

Project Management : We are making use of Agile - basically distributing jobs into independent-testable units for those with the required skill set and/or interest to work on them.

Communication: We will be using Discord for communicating with our team and our client and Email to communicate with our advisor.

Tracking: We will be using Git/GitLab to track changes in our project and plan to implement CI/CD using Gitlab runners to make sure there are no breaking changes / or re.

3.2 Task Decomposition

Note: each sprint is for 3 weeks

Sprint 1: Researching on alternative frameworks and technologies to use

Sprint 2: Starting to set up a basic bootstrap project and discussing more detailed structural concepts

Sprint 3: Have a basic structure with key libraries and technologies working. Also, breaking down the features/use-cases needed

Sprint 4: Adding the key use-cases - for the student side, including searching for courses, adding courses to semesters, adding new semesters, seeing course details etc. On the server side, we will need a database with key information regarding courses and a backend that can talk to our client application

Sprint 5: Get the round-trip working for sprint 4. Also, adding a log-in and sign up for students and a database for students and backend with the needed security. Additionally, an option to save the Program-of-study and retrieve it in the future.

Sprint 6: Adding a sign-up login for advisors/admin. Also, an option to share POS with advisors or print it out. Templating POS should be a stretch goal - this would require advisors to update the core-requirements each year. Another stretch goal would be to establish some communication i.e. chat.

Sprint 7: Working on sprint 6 stretch goals and refining and testing the progress so far. Stretch goals for Sprint 7 include creating pre-requisite checks and core-requirement checks.

Sprint 8: Working on deploying the project for use - working on the documentation for future teams to work on it

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Key milestones:

- Setting up a bootstrap application with key- technologies working
- Setting up a page where students can search for and drag-and-drop courses
- Detailed view to see course details and prerequisites
- Log-in and Sign-up screen for students with secured backend
- Option to save POS and retrieve it later
- Log-in and Sign-up for Advisors and Admins
- Page that allows advisors to enter updated core-requirements every year - to be used for templating
- An option for students to make use of templates
- Option to submit POS to advisors - option to print as well
- Prerequisite check and chat are key stretch goals

We will measure progress on these with each sprint being completed and creating detailed and proper documentation. Once a sprint is completed we will know how much progress has been made and what needs to be completed next. For example: Sprint 4 will require us to add key-use cases for the student side of the application. We will measure the sprint by checking if key features listed in that sprint are working as expected. In this case we will make sure that 90% accuracy the search button is working as expected and 99% accuracy adding and removing semesters is working as expected. After completion of a sprint we will create detailed documentation for what features were implemented and how to read it in the code. Once achieved we will be able to start sprint 5. Overall we want the new application to be working 90% accuracy all together and each feature should work with at least 95% accuracy.

3.4 Project Timeline/Schedule

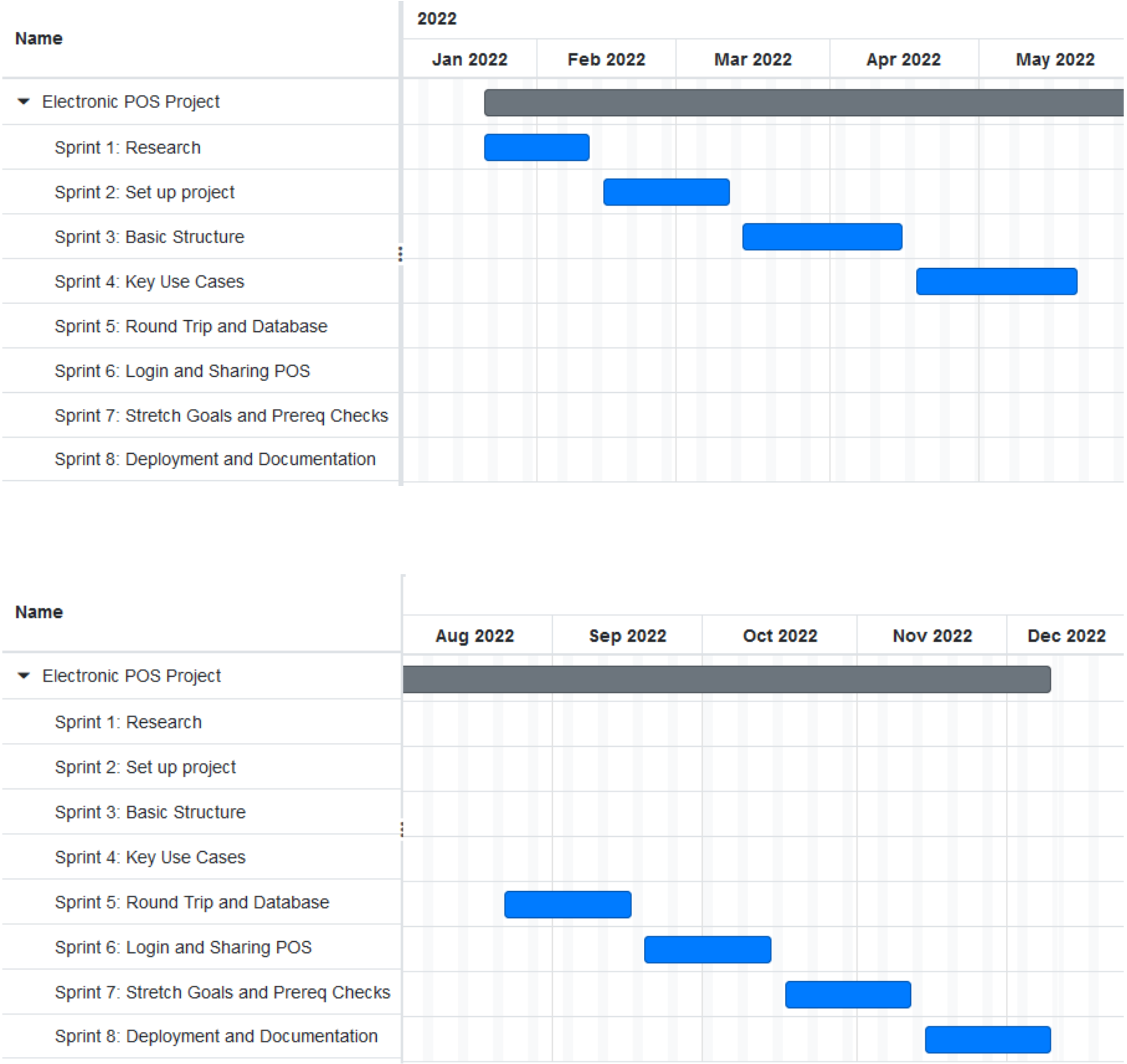

Figure 1: Gantt Chart

3.5 Risks And Risk Management/Mitigation

Sprint 1: There is a higher risk while we are determining technologies, as some may not have the tools we are looking for. This is a risk of 0.7. Since this task is researching other technologies and picking one we can easily eliminate technologies and find one that suits us.

Sprint 2: There is again high-risk since implementing a boot-strap with all the basics and getting it working as planned may require a lot of hours. A probability of 0.5 risk, we cannot eliminate this task since this task is risky when implementing.

Sprint 3: There will be a medium-risk since we have settled on the key technologies and frameworking and will be integrating the key libraries needed. A probability of 0.4, since this will mostly rely on us implementing framework and planning for the libraries we will need.

Sprint 4: There will be about a 0.4 risk for this sprint as well, since we will be using a database and adding key features to the application.

Sprint 5: This will also be a medium-risk, as this will be a probability of 0.3 for when we will be adding a login and sign up for the website.

Sprint 6: This is also a medium-risk with a probability of 0.3 as we will be adding a login page for admins and advisors and additional features.

Sprint 7: During this sprint we estimate a low-risk with a probability of 0.2, as we will be allowing advisors to change courses and requirements.

Sprint 8: Our last sprint will be our least risky at a probability of 0.1. This is because we will focus on deploying and documenting for future teams and users of the product.

3.6 Personnel Effort Requirements

NOTE: Sprint tasks are spanned for 3-week periods.

Task	Person-Hours (Estimated)
Sprint 1: Research	18 per team member (90 hours total)
Sprint 2: Implementation of Basic Structure and Frameworks	20 per team member (100 hours total)
Sprint 3: Basic Structure Prototype Completion/Research for More Complex Features	20 per team member (100 hours total)
Sprint 4: Implementation of Student-View Use Cases/Database	21 per team member (105 hours total)

Implementation to house Course Information	
Sprint 5: Prototype for Round-Trip Completed/Login Page Created/Database Implementation of Student Information/Add Security to Backend Structure/Save Feature for Program of Study Forms	22 per team member (110 hours total)
Sprint 6: Advisor & Admin Login Page Implemented/Share POS with Advisors Feature/Print Out POS Feature/Template Implementation Started	20 per team member (100 hours total)
Sprint 7: Finish Template Implementation/Start on Pre-Requisite and Core Requirement Checks/ Implement Testing of Prototype	22 per team member (110 hours total)
Sprint 8: Deployment Clean-Up of Prototype/Documentation	18 per team member (90 hours total)

Table 1: Personnel Effort Requirements

3.7 Other Resource Requirements

The only external resource we require is access to an ISU server for hosting the application.

4 Design

4.1 Design Context

4.1.1 Broader Context

Area	**Description**	**Examples**
Public health, safety, and welfare	Our project affects our various stakeholders (i.e. students and advisors) by allowing them to create a Program of Study that they can reference throughout their college career. Its ease of access allows students to create and access POS more easily than the current implementation of an excel sheet. This in turn allows students to plan more efficiently on their futures at school. It also helps with advisors in the fact that there could be significant time reduction in appointments needed to help plan out Programs of Study. This allows them to have more time for other appointments related to other topics.	Easy creation of Program of Study through simple web application. Better readability for both students and advisors for referencing a POS Increase creation of successful Program of Study with the necessary requirements needed to plan for completion of a degree at Iowa State University
Global, Cultural, and Social	Our project aims to reflect the values of responsible students and advisors. This project enhances the drive to plan for the future of students during their college careers. It also creates a social impact, since students will be able	Easier readability of POS means that students can share their POS with one another and plan to take the same classes at the same time

	to easily share readable templates with other students who are looking to take the same classes at the same time with friends.	Easier creation of POS can create a culture of students who look to be more prepared for the future.
Environmental	Based on our current design, our project has very little to no environmental impact. Our design of implementing a website means that there is only one server needed to run this application with Iowa State. This means that the environment will only be affected by the running of this server, as well as the running of computers or mobile devices needed to access the website. Since most people already have computers and mobile devices, the environmental impact of using them to access the website will stay the same as if we didn't create the website in the first place. The server from ETG was also already in place, so it had no increase in environmental impact in using it for our project.	Our website affects resources already in place and in use at Iowa State No extra hardware needed to implement the web application. Possible slight energy increase for using computers and servers more.
Economic	Our project currently will be available to all students and advisors for free. This allows Iowa State to take on no financial increase from using our solution. This can also affect budgets at Iowa State and allow for money to be spent on other solutions for other problems.	Free website for Iowa State (accessed via VPN or non proxied version) No increase in payment for using our web application Budgets can be focused in other areas for students instead of paying for a solution similar to

		ours. (i.e. pay for certain student resources instead of a POS creator)
Scalability	Our project is currently tailored to be used by students in a certain group of majors. However, we want to create a solution that can be scaled to a larger group of users (i.e. other students/advisors in other majors) if needed. So, our project needs to be scalable to accommodate this. To do so, we implement a design that is both stable for a smaller group of users, as well as being able to be scaled larger for the possible use of all students at Iowa State.	Access to the website via Iowa State VPN (which is available for all Iowa State Students/Advisors) as well as a non-proxy version Logic to gather course information can be applied to other majors if need be. Backend can be scaled to accommodate for larger user count and the information that it will bring.
Code Maintainability	For the future, we want to make sure that the code implemented will be easily maintainable. To do this, we want to make sure that we use common programming languages/frameworks that will be relevant for future use. We also want to make sure that the readability of the code is top notch for future reviewers. To do this, we will implement a lot of comments with the code, as well as a detailed readme file to show future workers how to run the project.	Using common programming languages/frameworks implemented for future reviewers of the project. Implement many comments within the code, as well as a detailed readme file on how to run the project.

Table 2: Broader Context

4.1.2 User Needs

Academic Advisors need a user-friendly interface to view student submitted POS forms in order to check if the forms are filled out correctly and keep students on track for graduation.

Students need a user-friendly interface to fill out and submit a POS form to schedule their future classes for their degrees.

4.1.3 Prior Work/Solutions

This was originally designed to be a continuation of POS designed by the previous team. So there has already been quite a bit of work done in this area. However, the group left very little documentation on how to expand the project. Also, their choice of language i.e. python limits what could be done to since most complex UI and complex routing - was done with heavily importing from other languages and frameworks - making the whole process very patchy and ad-hoc-ish. One member commented that it looked like a Frankenstein monster.

Thus we decided to look at other frameworks and technologies that solve similar problems - and we thought Angular was the best choice. It gives us type-checked language (TypeScript Configuration, n.d.) and routing and linking out of the box (Angular Routing, n.d.). Also, Materials library provides us the interface to build on the key UI components like drag and drop for courses (Drag and Drop, n.d.).

4.1.4 Technical Complexity

The current solution i.e. making POS on an Excel file and sharing it with the advisor via email is extremely cumbersome for the advisors and the students. Students need to find all the relevant course catalogs and the major requirement documents and figure out where to put what. Advisors too need to headcount credits and track requirements manually. The solution to this is simple i.e. a webapp, but also extremely complex. Here are some key areas of complexity:

1. Making sure the app is accessible, especially for freshmen, means that we cannot rely on VPN usage - which means a reverse-proxy would need to be set up with a public facing domain
2. The user personal data would need to be encrypted to prevent any fraudulent use
3. Modules would need to be created to allow easy-sharing of POS from students and advisors which would require a complex backend
4. The dragging and dropping functionality of courses along with complex UI functionality like quick-search results or adding and deleting semesters would require specialized front-end tools

5. The application would need to follow strict software engineering principles for instance ‘keeping it simple st*d’ to make sure our application can be expanded upon in the future by others.
6. The application would be serving thousands of requests a second - this would mean that there needs to be an eye towards scalability

4.2 Design Exploration

4.2.1 Design Decisions

For the frontend of our application we chose to go with a TypeScript based framework - Angular, which is a platform for building mobile and desktop web applications. For the backend of our application we decided to use the Java based framework Spring Boot and SQL for our database.

A few key decisions that we have made in relation to our proposed solution include many different features. This includes a drag and drop feature for students to drag classes and place them into rows of numbered semesters. A reset feature to clear the selections for classes students have made. A submit button to submit the form to a specified academic advisor. A drop down to select a major and display the core classes that are required by that major. A sign in feature for both students and academic advisors to allow saved work. Lastly we plan to implement a prerequisite check to ensure the students have filled out their planned courses successfully.

4.2.2 Ideation

When determining how we wanted to proceed with the project we thought the best way to create a user friendly website was to implement go with angular since we found it much easier to use to build our application. We then brainstormed ideas on what we could use it for and landed on a drag and drop feature along with a drop down menu to select the major the user is in and list core courses corresponding to it. With this in mind we considered implementing a prerequisite check, that checks the student’s schedule and makes sure they do not have a prerequisite class that is needed before taking the next class. We also thought of how to add blocks that are editable for students to fill in with electives that they may take. Another idea we had was to add a reset option if the student wants to reset their schedule and start over. Our final two ideas were to include an option to add additional semesters and displaying course details.

4.2.3 Decision-Making and Trade-Off

The current implementation uses a python based microservice based MVC framework (django) for backend and frontend - with sqlite to store data. This made it very difficult to identify and verify endpoints. Also sqlite is not scalable. The lack of independent frontend meant that adding simple features was very time consuming. Thus major design decisions included:

1. Making a separate backend and frontend. This will need more time at the beginning but should pay up with reduced development cost in the future.

2. We are using a reverse-proxy to serve public requests. This adds a degree of complexity to deployment but would allow students and advisors to access everything without a VPN.

3. We will use a Java-based Backend framework for servicing backend requests. This would allow students and researchers from ISU (where Java is still the dominant language) to easily add new end-points and test them.

4. We will use TypeScript based framework i.e. Angular for Front-end which requires a steeper learning curve but we wouldn't need patches of three other languages and frameworks to get the relevant UI features.

4.3 Proposed Design

4.3.1 Design Visual and Description

The design is made with our own design decisions in mind as well. To visualize the architecture, we can think of users making a request for the relevant information as the starting point - in our case users are expected to interface with a publicly accessible domain - and so our design starts off with a server listening at that domain (in port 443 for https).

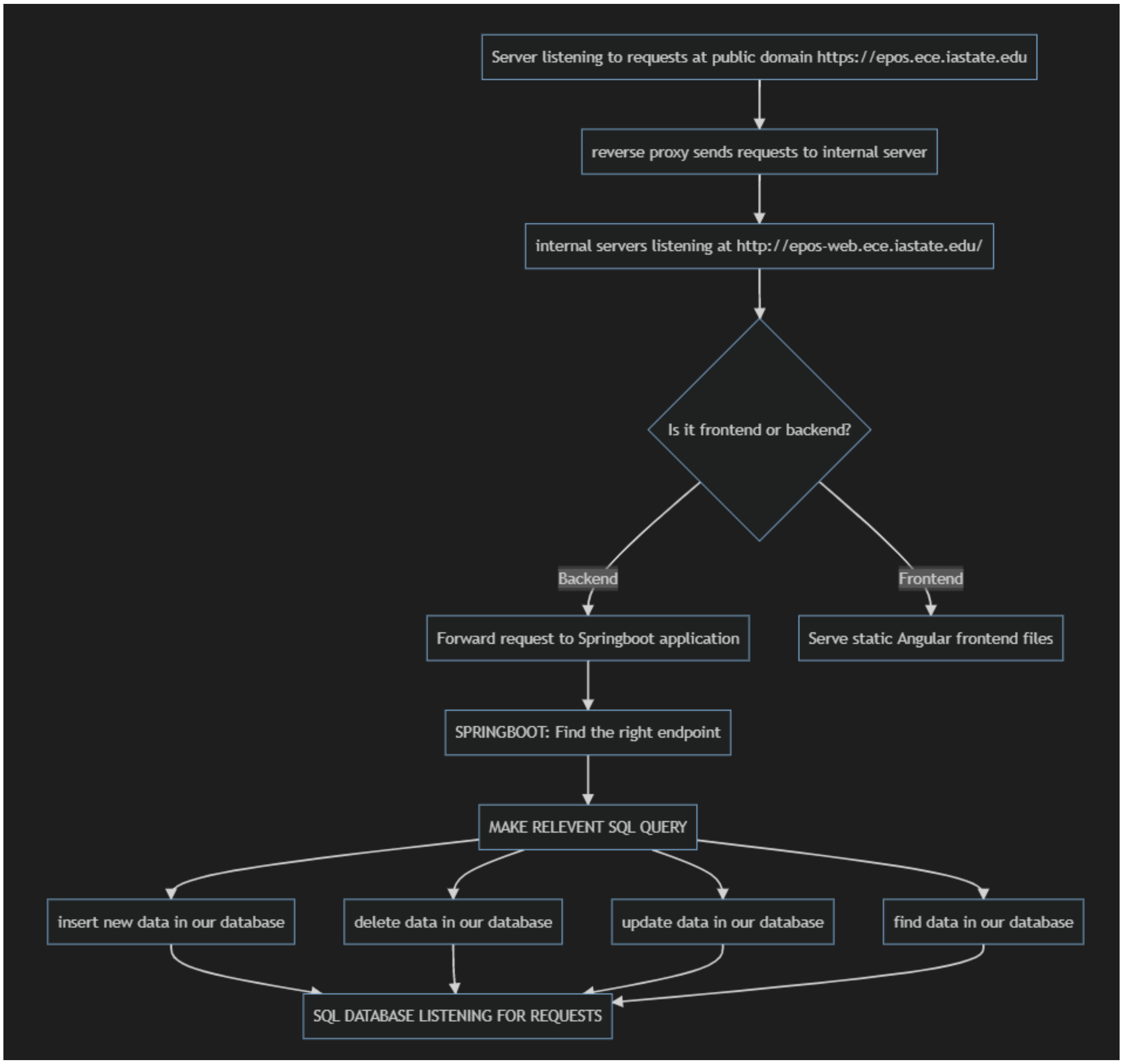
Server listening to requests at public domain https://epos.ece.iastate.edu
reverse proxy sends requests to internal server
internal servers listening at http://epos-web.ece.iastate.edu/
Is it frontend or backend?
Backend
Frontend
Forward request to Springboot application
Serve static Angular frontend files
SPRINGBOOT: Find the right endpoint
MAKE RELEVENT SQL QUERY
insert new data in our database
delete data in our database
update data in our database
find data in our database
SQL DATABASE LISTENING FOR REQUESTS

Figure 2: Design Visual

4.3.2 Functionality

The design of our Program of Study project is intended as a supplement to the normal 4 year plan process. As such, it should be designed in a way that puts as little stress on the students and advisors as possible while they focus on more important aspects of the class. The functionality is meant to limit the tedious nature of scheduling. This can be achieved with our currently proposed design which pulls scheduling information from official ISU sources, so that students and advisors don't have to seek them out from multiple sources. In the real-world this is very viable because all the information is out there we just need to make it readily available to students and advisors.

4.3.3 Areas of Concern and Development

The primary concern for our project is where we will be sourcing class information from and how that information may change in the future. Our design proposes sourcing class information from official ISU sources, but since we aren't in control of those resources we don't know how formatting or location of the data could change. We could address this concern with the client or faculty advisor as they are both ISU employees that could shed some light on possible changes, but they are not in control of those sources either. A change to the design could include having that class information stored within our project, but that would require changes to be manually implemented by administrators of this project.

4.4 Technology Considerations

One of the biggest considerations was separating the backend and frontend - this was to allow everyone to understand a very UI specific task for instance adding a button to a certain page in isolation of the backend. This does take away from not having a 'true' Model, View, Controller design - like Django would but it allows those more specialized in certain domains to focus on adding and testing features in it.

Also, Angular on the frontend is a great TypeScript based framework - but is less flexible and opinionated compared to something like React. Building an frontend application based on React is much much more time consuming - but it would have resulted in a more compact application and ultimately more scalable.

Another, competing design/technology choice was to have a backend built on more bleeding edge platforms and frameworks - something like Go or Rust. However, the time it would take to learn these languages vs. the improved efficiency was thought to be not worth it. Also, having a widely used, type checking language like java, and a tried and tested platform like Springboot would make it easier to be expanded upon in the future.

4.5 Design Analysis

This design is already tried and tested - since the previous team's design was the primary inspiration behind using something more maintainable and scalable. To the extent that we wanted the key UI features implemented, the adoption of Angular has worked - and we have been able to implement the Drag-and-Drop in a prototype created on GitHub (Altaf, 2022).

4.6 Design Plan

The project in itself is divided into 2 main parts - frontend and backend.

The Frontend uses AngularTS where routes are used to coordinate modules (which are called components). Each use-case should be tied to an independent and testable module, which adds to or reuse existing interfaces - and can be tested against the existing REST Api's on Spring Boot.

The Spring Boot will be divided into several modules - Controllers, Repositories, Models etc. And every use-case will be tied to new classes for each of these modules. This should help maintain the code in the long-run and avoid merge conflicts.

5 Testing

Testing is an extremely important component of most projects, whether it involves a circuit, a process, power system, or software.

The testing plan should connect the requirements and the design to the adopting test strategy and instruments. In this overarching introduction, given an overview of the testing strategy. Emphasize any unique challenges to testing for your system/design.

5.1 Unit Testing

Since our project is a programming one, we can apply unit testing in the conventional sense to test that each individual class and method conform to their intended uses. For unit-level testing, utilities such as JUnit or other similar tools can be used to write functional tests. For the backend, mocking tools can be used to test individual units even when their dependencies haven't been fully completed.

5.2 Interface Testing

The primary interface that we would be testing is how we are pulling class information into our program. We need to make sure that individual classes get pulled in with the correct format. Another problem that should be tested is the prerequisite and corequisite and making sure that those link certain classes together properly and display the proper errors from that.

Other interface testing could come in the form of the links to other official ISU resources that manage student information, but that would come in future versions of this project.

5.3 Integration Testing

For those working on the front-end, they can test if the endpoints for the backend are communicating properly with the front-end i.e. conduct end-to-end testing. Additionally, they can create mock-data and enable testing of front-end modules if backend controllers haven't been created.

Similarly, for backend testing, if the frontend apis are already defined, it should be simple to see if the new backend code is working - otherwise Postman can be used as proof of successful completion.

5.4 System Testing

Continuous Integration and Continuous Deployment (CI/CD) pipelines will be set up to make sure the codes are working properly. Additionally, Continuous Deployment will enable others to see the results of the commit on the deployed environment.

Gitlab allows for runners to be used for CI/CD pipelines which will need to be set-up on the server. Additionally, There are pre-build images that docker enables to see how a program might compile.

5.5 Regression Testing

To ensure that our new additions do not break old functionality, we will be using peer review of code through GitLab pushes/pulls. Team members will be working on their own separate branches and ask to have changes pushed up to the Master branch. A team member will look over the said changes and accept, deny, or discuss these changes before any action is taken to add more code to the master branch. If a push of code breaks the functionality of the project, then we can adjust back to the previous iteration found within our repository. Main critical features that we need to ensure do not break for our project include our User Interface, Class Information, and Save feature. This is driven by our requirements of wanting to create an application to help students create a Program of Study to help plan their semesters at Iowa State University. These three features are the features that are required for this requirement to take place. The main tool we will be using for this will be GitLab, as this will hold our project repository and our pipeline of changes from past and present versions.

5.6 Acceptance Testing

We will show that the design requirements are being met through demonstrations of the current state of the application to the client and advisor. Through these presentations they will have the opportunity to give their feedback and suggest changes to ensure that the final product is to their satisfaction.

5.7 Security Testing (if applicable)

The primary security concern for this application is the issue of privacy for class information. For this we just have to secure the end points and ensure accounts have the proper securities. That way there are no situations where the students' information can be accessed from unauthorized sources.

5.8 Results

Since the project is broken up into two parts - both backend and frontend will be tested separately. Postman results should suffice to show what an end-point gets done. Additionally, The backend can be tested with mock data if backend controllers corresponding to it haven't been created.

The two are tied together with a web-server - Apache2 or Nginx. This should be testable upon deployment. If we have a CI/CD pipeline set up, this can be tested quite easily by just going to the website.

6 Implementation

Our initial plan for next semester is to start by finishing the framework for a semester plan tracker. This consists of a website that will have drag and drop functionality and class information. From that point we will work towards more automation and functionality. This means implementing the necessary prerequisites and major requirements. This will take more time as it is a much more complicated process that requires cross-major information support. Then we will add additional support in various ways that will be beneficial to both advisor and student. Not all of these features have been fully decided, but some of them include printing and integration with other ISU resources.

7 Professionalism

This discussion is with respect to the paper titled "Contextualizing Professionalism in Capstone Projects Using the IDEALS Professional Responsibility Assessment", International Journal of Engineering Education Vol. 28, No. 2, pp. 416–424, 2012

7.1 Areas of Responsibility

Area of Responsibility	Definition	NSPE Canon	Chosen SE Code of Ethics	Description	Difference from NSPE Version
Work Competence	Perform work of high quality, integrity, timeliness, and professional competence.	Perform services only in areas of their competence; Avoid deceptive acts	2.01. Provide service in their areas of competence, being honest and forthright about any limitations of their experience and education.	Engineer should only engage in work of their own area of expertise and be honest about their limitation	NSPE is talking about using the knowledge you have and applying it to what you think is your best area. Where SE takes experience and education into account.
Financial Responsibility	Deliver products and services of realizable value and at reasonable costs.	Act for each employer or client as faithful agents or trustees.	4.04. Not engage in deceptive financial practices such as bribery, double billing, or other improper financial practices.	Engineers should be prudent with expenses that will be billed to the client and not engage in financially deceptive practices.	The SE code is more specific on the types of deception that are not permitted.
Communication Honesty	Report work truthfully, without deception, and understandable to stakeholders.	Issue public statements only in an objective and truthful manner; Avoid deceptive acts.	1.04. Disclose to appropriate persons or authorities any actual or potential danger to the user, the public, or the environment, that they reasonably believe to be associated with software or	If a project were to be dangerous to anyone or anything, report to the authorities.	NSPE describes being truthful about the work, while SE is more focused on disclosing potential

			related documents.		dangers that are related to the software. This means NSPE would be giving the whole truth when SE might not.
Health, Safety, Well-Being	Minimize risks to safety, health, and well-being of stakeholders.	Hold paramount the safety, health, and welfare of the public	1.03. Approve software only if they have a well-founded belief that it is safe, meets specifications, passes appropriate tests, and does not diminish quality of life, diminish privacy, or harm the environment. The ultimate effect of the work should be to the public good.	Software should only be approved if it meets specifications and doesn't harm the public or the environment.	The SE code is more specific in that the result of any project should help the public in some way, but also talks about meeting specifications.
Property Ownership	Respect property, ideas, and information of clients and others.	Act for each employer or client as faithful agents or trustees.	2.03. Use the property of a client or employer only in ways properly authorized, and with the client's or employer's knowledge and consent.	The property of a client should be handled in a professional way that does not violate trust between client and the project team.	SE Code of conduct is more specific in how to act as 'trustee' while NSPE is quite obscure

Sustainability	Protect the environment and natural resources locally and globally.		3.03. Identify, define, and address ethical, economic, cultural, legal, and environmental issues related to work projects.	Engineers should take notices of environmental issue related to the project that they engage in	Since NSPE doesn't have anything to say about this, it differs in that SE Code does have something to say
Social Responsibility	Produce products and services that benefit society and communities .	Conduct themselves honorably, responsibly, ethically, and lawfully so as to enhance the honor, reputation, and usefulness of the profession.	2.07. Identify, document, and report significant issues of social concern, of which they are aware, in software or related documents, to the employer or the client.	Notify clients of potential issues related to the product or the relationships within the team or with other clients.	The NSPE definition notes that the individuals should conduct themselves lawfully. The SE Code of Ethics does not say anything about notification of law breaking. Only of social concern.

Table 3: Areas of Responsibility

7.2 Project Specific Professional Responsibility Areas

Area of Responsibility	Definition	Does it apply? Why?	Performance and Justification
Work Competence	Perform work of high quality, integrity, timeliness, and professional competence.	Yes this does apply to our project as in every project we should be thinking about competence. Who is right for which job and who can perform with efficiency and quality in that specific area.	High, since work assignments are based on each individual's competence and areas on interest
Financial Responsibility	Deliver products and services of realizable value and at reasonable costs.	Yes, since we may possibly be using the university's funds to build our database and use other tools that may help us reach our end goal.	High, since we are using open-source tools that would be minimizing any continuous cost the university would incur outside of hosting the servers
Communication Honesty	Report work truthfully, without deception, and understandable to stakeholders.	Yes, since constant communication is the key to delivering a product that our client would be happy with	High, since we have established very frequently used channels of communication with our client as well as our advisor
Health, Safety, Well-Being	Minimize risks to safety, health, and well-being of stakeholders.	It doesn't apply since it's a web-app that would be replacing an already bad software based solution	N/A
Property Ownership	Respect property, ideas, and information of clients and others.	Yes, this does apply since we are using resources unique to the university as	High, since we are building on a previous phase of a project we have to

		well as code from a previous phase of the project. It would not be prudent to claim ownership of such things.	clearly differentiate our work from the previous code.
Sustainability	Protect the environment and natural resources locally and globally.	No, since we are mainly using virtual resources and do not associate any natural or environmental resources to our project	N/A, since our project does not incorporate any natural or environmental resources.
Social Responsibility	Produce products and services that benefit society and communities.	Yes, since our whole goal with the project is to produce a website to help the Iowa State Community with planning their college career paths.	High, our project focuses on a service that will benefit students and advisors who will need this to schedule their future courses at college.

Table 4: Project Specific Professional Responsibility Areas

7.3 Most Applicable Professional Responsibility Area

We chose Social Responsibility as it directly related with the end goal of our project. This would be a web application that helps students and advisors schedule their future courses in a user friendly and organized manner. Such an application would help these groups and make their lives easier, fulfilling the criteria for benefiting the community. Since this project is a continuation of a previous team we found the layout of the application to be poor. Once we saw this we immediately thought we could improve the design to make it even more user friendly and visually appealing.

8 Closing Material

8.1 Discussion

Through this semester of work, we have successfully created a plan for implementation that will guide us through our coding work in the coming semester. We were able to determine that it would be better to work on a new implementation rather than refine the original, due to issues with the original's documentation and technologies used. Our main result of the work this semester was a large quantity of knowledge regarding technologies to be used in our project, as well as a concrete implementation plan.

8.2 Conclusion

Throughout this semester, we've obtained valuable knowledge through research on technologies and the previous project phase. We were able to use this knowledge to create a plan for implementation, ensuring that the project's development will be smooth. Our goals for the coming semester include:

- Create a working prototype for the project by the time the SE 166 class does their program of study unit, so we can get feedback from them on our application.
- Complete a working final version of the application by the end of the fall semester that meets our listed requirements
- Work on the stretch goals of having a prerequisite check and a student-advisor chat feature

We plan to achieve these goals by adhering to our Agile sprint plan, and ensuring progress is made through discussion in our weekly meetings, and will strive to create a product that will make the lives of students and advisors easier.

8.3 References

List technical references and related work / market survey references. Do professional citation style (ex. IEEE).

Altaf, S. (2022, March 23). Saljooq / POSAngularExp. GitHub.
https://github.com/Saljooq/POSAngularExp

Angular Routing. (n.d.). Angular. Retrieved March 26, 2022, from
https://angular.io/guide/routing-overview

Drag and Drop. (n.d.). Angular Material. Retrieved March 26, 2022, from
https://material.angular.io/cdk/drag-drop/overview

TypeScript configuration. (n.d.). Angular. Retrieved March 26, 2022, from
https://angular.io/guide/typescript-configuration

8.4.1 Team Contract

Team Members:

1) Saljooq Altaf
2) William Hunt
3) Carson Campbell
4) Nathan Marquardt
5) Noah Nickel

Team Procedures

1. Day, time, and location (face-to-face or virtual) for regular team meetings:

Tuesday, 4pm (in person, Coover Hall)

Saturday 1pm (virtual)

2. Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):

Our team Discord server will be the primary method of communication.

3. Decision-making policy (e.g., consensus, majority vote):

We will take a vote and the majority will be implemented.

4. Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):

Meeting minutes will be archived in a Google Documents file located in our team Drive folder. (Saljooq Altaf will maintain the minutes of meetings)

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:

All hands meeting on Tuesdays at 4pm

Meeting as needed on Thursdays and Saturdays

2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:

Responsibility will be divided based on individual requirements of the client. The timeline of these will vary, but a deadline will be set based on the individual requirement and its necessary research/implementation.

3. Expected level of communication with other team members:

The expectation is that team members will be open during meetings and assignments via the Discord channel. If one were to miss or is struggling with a meeting or deadline, then he shall notify other team members in a timely manner.

4. Expected level of commitment to team decisions and tasks:

It is expected that team members will put forth the best effort they can in regard to tasks and other matters decided upon by the team.

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):

Saljooq Altaf - Meeting coordinator, Code Reviewer, Full-stack developer

William Hunt - Back-end Programer, Code Reviewer

Nathan Marquardt - Front-End Programmer, Code Reviewer

Carson Campbell - Back-End Programmer, Code Reviewer

Noah Nickel - Front-End Programmer, Code Reviewer

2. Strategies for supporting and guiding the work of all team members:

We will utilize code reviews through merge requests, which among other benefits allows the team to be aware of the work other members are doing and provide feedback if needed. Additionally, discord will be used to keep everyone up to date and see who wants to pair up to review latest changes.

3. Strategies for recognizing the contributions of all team members:

The weekly reports and weekly standup meetings will be the primary method of recording the individual work of each member. Additionally, their contributions will be recognised in the in-person weekly standup meetings. Also, all members will have a 3-4 week sprint to incorporate the current backlog.

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the team.

Saljooq Altaf - I have worked on multiple stacks and worked as a full stack developer. Additionally, I have been the team lead for several projects. Also, I have experience setting up CICD pipeline and working on Linux terminal.

Nathan Marquardt - Coding Curriculum from Iowa State, Soft Skills, Experience Coding from Danfoss Internship (still ongoing).

William Hunt - Iowa State coding curriculum, experience with back-end leadership, experience with CI/CD and the Linux terminal.

Carson Campbell - Experience with backend code and group work based around Iowa State classes, GIT, and organizational experience.

Noah Nickel - I have experience with front-end code, but was also able to help with back end on a group project for Iowa State. I have GIT experience as well.

2. Strategies for encouraging and support contributions and ideas from all team members:

Our weekly meetings serve as brainstorming sessions as well, which allows the team to amass ideas and hear the perspective of each member.

3. Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their opportunity or ability to contribute?)

Team members will be expected to bring the issue up to the team themselves, either through a message on the Discord server or by bringing it up during a meeting.

Goal-Setting, Planning, and Execution

1. Team goals for this semester:

Fulfill the client's requirements and document our work thoroughly to ensure future teams will know where to pick up from where we left off if need be. Additionally, keep the advisor and client up to date on our current progress.

2. Strategies for planning and assigning individual and team work:

Since the requirements are going to be updated regularly, the Agile strategy seems fit for the purpose of planning work. We will have a 3-4 week sprint to incorporate the changes. Additionally, we will have regular weekly stand ups on Tuesdays where tasks will be assigned at the beginning of the sprint.

3. Strategies for keeping on task:

We will be meeting regularly for weekly stand ups, and everyone will explain the progress made on the task/ feature taken up by them and any hurdles faced in the process.

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of any of the obligations of this team contract?

If a member is repeatedly infringing on the contract, it will be brought up during the next meeting and the team member will be asked to explain. The team can work to accommodate the member if there are issues impacting their work on the project.

2. What will your team do if the infractions continue?

If the issues with the team member cannot be resolved, we will appeal to the course instructors and advisor for assistance.

****

a) I participated in formulating the standards, roles, and procedures as stated in this contract.

b) I understand that I am obligated to abide by these terms and conditions.

c) I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.

1)______Saljooq Altaf____________________________________DATE ___2/12/2022_________

2) _____William Hunt__________________________________DATE ___2/12/2022_________

3) _____Nathan Marquardt____________________________DATE ___2/12/2022_________

4) _____Carson Campbell______________________________ DATE ___2/12/2022_________

5) _____Noah Nickel_____________________________________DATE ___2/12/2022_________