**Virtual Advisor**

A software development project by:

Danielle Romanoff

Fawaz Alturbaq

Kyle Aldrich

Sonali Thaker

**Dhruvil Parikh**

*(Team Lead)*

Siddhesh palav

Ekkasit Pinyoanuntapong

“ I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/ examination. I further pledge that I have not copied any materials from a book, article, the Internet or any other source except where I have expressly cited the source

**Purpose:**

Virtual Advisor is a web application that aims to ease the registration of students during their college experience. While students are the majority of our user base, we hope to solve the issues correlated with other users of a registration system as well. The goal of this project is to address any existing issues of the current registration system. Also, we hope to provide the most convenient and efficient user experience. An optimal user experience encompasses a central hub where a user can complete tasks or view information in regards to the registration process. This experience will be enhanced by the ability to efficiently complete those tasks and represent relevant information clearly. Virtual Advisor will allow students to choose career paths that will optimize their learning experience and help create impactful resumes for a life after higher education.

**Problem:**

There has been strong conversation amongst the student body of the constant struggles of class registration. This frustration not only applies to students but also extends to other users of the registration system. Such additional users would include professors/advisors and registrar staff. Feedback has shown the registration has multiple problems from a process and software perspective. Such process vulnerabilities are lack of availability of courses to students and yearly fluctuation of course requirements. Some software weaknesses include non-user friendly interface, disorganization or lack of data, and minimal features. These drawbacks are some of, but do not address all of the system’s limitations. As our team moves further along into this project we document and address theses problems

**Technologies:**

Our team was able to make use of several technologies. Each bit of technology used was part of open-source software. This brand of software was a big advantage to our team’s success. The inexpensiveness of open-source software was vital given our budget constraints. This concluded that open-source software would provide the resources to complete backend and frontend functionality of our web application at the right price.

While selecting open-source software was part of our decision making process, we did consider other factors. These factors were categorized around two main options, business and technology. Some business factors include budget, personnel, time constraints, and investor opinion. Some technology factors are, but not limited to, resource allocation, personnel, optimization, and scaling. As we break down the project’s software, we will also describe the advantages provided by each chosen technology.

**Node.js:** Despite many members of our team having little to no Node.js experience this backend tool was a viable option. Node.js employs the Javascript programming language and Javascript is predominantly used across which is predominately used across web applications. Since our team was building a web application, we thought it was a perfect fit. We ultimately decided on Node.js because of the added benefits it offers.

One of the added benefits of our chosen programming language was its ability to handle JSON files. Our database collections and http data are all using JSON, therefore using Node maintained the flow of data from frontend to backend and vice versa. In summary, Node.js will be used on the client, server, and in the database. An idea like this is even a unique feature amongst web applications.

Also Node.js had a great package manager called NPM. NPM has an abundance of modules, some of which we used in our project. Two examples of driver modules used were express and mongo. Express provided a web application framework with many additional features. Mongo module is a driver that worked with our Mongo database.

Lastly, Node.js is a programming language that works fasts. There are multiple reasons that help it push and pull data at an accelerated rate. One example of this is its concurrent connections. It operates on a single event loop that is able to handle multiple requests and does not block input/output calls. Such a feature helps it deal with events asynchronously. Also, it is able to execute the language very fast. It uses the V8 engine developed by Google, which in turn compiles the language into machine code. These features would be extremely helpful to students using our web application. Registering for classes can be stressful and we want to minimize that stress. The speed of our application should provide real-time and updated information, so students know what classes are still available.

**MongoDB:** MongoDB provided many opportunities for our backend developers. The database is great for scaling and converting into JSON like documents with the proper schema. It turns out that the data given to us, gave us the opportunity to complete both those tasks. These JSON like documents allow us to uniquely order our data in a manner best suited for our environment. Also, this feature would easily allow us to alter the data if our initial designs were not optimal or add additional data. MongoDB also works well with NodeJS. The reason for this is because Mongo and Node both employ Javascript and JSON making data easily interchangeable.

**Mongo Compass.** To coincide with MongoDB, our team utilized the graphical user interface Mongo Compass. This interface delivers the user with a visual organization of data that is easy to interpret and break down. Mongo Compass also provides users with querying capabilities. There are multiple ways to perform a query in Mongo Compass. Data can be extracted by either clicking/ dragging fields within the interface or physically inputting specific queries.

**RabbitMQ:**

**Github:** Github is one of the mostimportant tools used by software developers today. It provides a repository where programmers can work, store, and collaborate on a software project. On a web application, like ours, these features were invaluable. Github can track a project’s progress and allow developers to view and work on the most updated versions of software. Such real-time events prevent team members from completing similar tasks and limits mistakes during integration. The cloud-service also has some unique features such a bug tracking. This feature was very useful as we approached our project deadline. This feature helped our developers track and fix bugs to ensure our functionality would be running smoothly at the deadline.

**Github desktop.** Github desktopsignificantly simplified our ability to use Github. The desktop is a tool that provides all of the functionality of Github on an easy to use interface. It graphic interface helped us, as users, to make meaningful commits, track our team members and project history. Not all team members were familiar with using git commands in command line. Github desktop eliminated this process and allowed all team members to commit with a few clicks.

**HTML/CSS:** These two languages are commonly used across web pages. For this reason, we actively looked into using these languages and found out that HTML and CSS were a viable option for our frontend. HTML/CSS are trusted products and have a high level of frontend functionality that would be beneficial to achieving our goals. Also, several members of our team had prior experience with these languages.

**AngularJS:**

**Trello:** Trello was mostly used at the beginning of our project. This tool gave us the opportunity to organize and detail our requirements on a collaborative interface. The interface provided an enormous amount of features that allowed us to prioritize and divide tasks/requirements amongst team members.

**Gmail/Google Drive/ Hangouts:** Our team used a number of tools provided by Google. All of these tools simplified the collaboration process that is necessary to working on a software project. These tools permit the team to stay informed and communicate to ensure the project was moving in one direction. Lack of communication can be a huge detriment to a project’s success. Location, personal issues, events, and religion are some of the circumstances that can prevent a team from meeting in person. These tools allowed us to communicate under such difficult or different circumstances.

**Goals:**

* Provide extensive research on the class registration system from a software and user perspective
* Identify current issues in regards to class registration system and process
* Identify and create a list of additional features suggested by the majority of users
* Create solutions to registration difficulties

**Successes:**

Overall the project went rather successfully. No project is void of challenges or mishaps but such event always happen. Luckily, as a team we can say that zero events occurred that altered or stopped development. We have provided the area below to discuss what we thought were some success of the project. These ideas were helpful for us and we hope that our team and other groups can use them to their advantage in the future.

* Designated schedule. While team members were allowed some flexibility, the schedule structure was concrete and all members could attend.
* Designated roles. Some flexibility was allowed but generally team members were split up between frontend and backend that coincided with their strengths as a developer.
* Technology choices. Dedicated time to during the design process to thoroughly process the technologies chosen and why they were chosen.
* Managed expectations. It can be difficult to move a project in one direction when all members have opinions. It was important to keep in mind, that the ultimate goal of the project is a satisfied customer.

**Problems/ Solutions:**

Problems always happen, especially in dynamic environment. Our team was fortunate enough not to run into any major delays. Below, we have listed some events that potentially had an impact on the project.

* Problem: Github issue with one team member.
  + Solution. Used one developer who was more familiar with Github to resolve the problem.
* Problem: Learning curve with new technologies.
  + Solution: Dedicated our first sprint to familiarize members with technologies.

**Requirements:**

1. As a user, I want a secure login so other people are unable to view my personal information.
2. As a student, I want to see my study plan in my account which is updated semester wise (if required) by course advisor
3. As a student, I want to be able to check what required courses I have already completed and what is remaining.
4. As a student, I want to see my semester GPA, GPA in my major, cumulative GPA, and unofficial transcript
5. As a guest user, I want to see or access the list of courses(undergraduate and graduate) according to department (such as Computer Science, Mechanical Engineering, etc.), by major/ concentration, and/or graduation requirements for each major.
6. As a user, I want to see the list of all professors and advisors by majors.
7. As a student, I want to be able to view and register for classes.
8. As a student, I want to be able to submit or send my graduation application online so I do not have to do it manually.
9. As a user, I want rules and regulations related to OPT/CPT according to visa status. Also I want to be able to submit an application form to the graduate department to save time.
10. As a user, I want a link to see information about current market trends (such as new programming languages etc.) and events (such as conferences etc.)
11. As a student, I want to be able to see the professors that have received good reviews and recommendations from other students.
12. As a color blind, user I want dark text & light background so that I can easily read the text.
13. As a student user, I want to be stopped if I try to register for a course for which I do not have the prerequisites.
14. As a user, I want to see what courses that I have already taken that would count toward the requirements of other majors or certifications.
15. As a student, I want the system to put me in a waiting list if the class is full and notify me whenever my position in the queue has changed.
16. As a student, I want to be recommended the best classes or concentration for me based on my history with a percentage of success.
17. As student, I want a system to generate username and password fro me automatically and send that information via stevens email.
18. As a system user, I want to recover my username and password so that I can access the software
19. As a student or advisor, I want to be notified if my account has logged in a different computer for security reasons.
20. As a registrar, I would like the system to have a backup database in case of power failure. This database would have personal information related to students and faculties.
21. As a student, I would like to have a feature, which allows me to pre-select the class for future semesters and allows me to make a draft of my schedule before registration.
22. As an advisor, I want the system to automatically generate graduation candidacy forms for students who are graduating in a semester to optimize the process for students.
23. As an advisor, I want to have a report of all my advisees with contact information, completed and remaining courses so I can better help my advisees.
24. As a department planner, I want to have a summary of class enrollment including, the number of students in each class, the professor that is teaching each class.
25. As department head, I want to know the major of all students taking courses to understand opportunities for new students.
26. As an advisor, I want to know what courses and instructors other students have taken so I can avoid recommending bad courses to my students.
27. As department planner, I want a database repository so I can run arbitrary queries in reference to assigned students.
28. As an advisor, I want to know what relevant classes are available to recommend t students so I can better advise students.
29. As department head, I want to see a summary of course surveys by section, course and instructor so I can better understand how instructors are doing.
30. As department head, I want a report of how many people add and drop each class so I can better plan to meet the needs of the students.
31. As a department planner, I want to know who showed interest in a class, e.g. Tried to register but the class was full so I can better plan needs.
32. As department head, I want a report to show the number of students who have taken each class and section by semester so I can better plan students needs.
33. As an advisor, I want a report showing which required classes a student must take, so I can better advise the student on courses.
34. As an advisor, I want a recommendation system that considers the classes a student has take, the remaining required courses, and the student’s interest to help pick courses.
35. As a user (student, registrar or faculty), I would like to have built in email functionality, which allows emailing other people by just search of their name (don't need to remember Stevens email address for individuals).
36. As a user (student, registrar or faculty), I want to have user friendly and attractive interface of the system.
37. As a professor, I want to be able to enter available times for the students to schedule meetings.
38. As a student, I want to be able to sign up, based on the professor's availability, for an appointment or meeting.