

NYU Shanghai Attendance Tracker

Final Report

Daniela Oh

Supervised by: *Olivier, Marin*

Abstract

Many classes in NYU Shanghai have attendance as a requirement, however, there is no school provided method for professors to take attendance. Existing online solutions are available, but, they do not address NYU Shanghai's specific requirements for an attendance tracker. My project can take attendance in a similar way to these online solutions. However, it is made according to the school's database format, which makes it easy and simple to use. The project aims to facilitate importing data for each course. Professors can use the easily accessible rosters on NYU Classes to import the data and use the application to take attendance effortlessly. The data from taking attendance will also be saved for later reference. This way, calculating each student's attendance grade will not take much time nor effort. According to survey results, the project achieves its goal because all professors say they prefer to use this product over their current methods. It is "easy to understand", "very convenient" and saves them "a lot of time".



Contents

1 Introduction

- 1.1 Context
- 1.2 Objective

2 Related Work

- 2.1 Existing Solutions
 - 2.1.1 TeacherKit
 - 2.1.2 MyAT
 - 2.1.3 TrackCC
 - 2.1.4 Mitrefinch
- 2.2 Frameworks
 - 2.2.1 AngularJS
 - 2.2.2 ReactJS
 - 2.2.3 EmberJS
 - 2.2.4 VueJS
- 2.3 Summary

3 Solution

- 3.1 Preliminary Study
- 3.2 Architecture
- 3.3 Features
 - 3.3.1 Log in/ Sign up Page
 - 3.3.2 Welcome Page
 - 3.3.3 Home Page
 - 3.3.4 Import Page
 - 3.3.5 Roster Page
 - 3.3.6 View Attendance Page
 - 3.3.7 Take Attendance Page
 - 3.3.8 Student Profile Page

4 Evaluation

- 4.1 Survey Results
- 4.2 What it means

5 Lessons Learned

6 Conclusion

- 6.1 Contributions
- 6.2 Future Implementations

7 References



1. Introduction

1.1 Context

Currently, for NYUSH professors, there is no efficient way to take attendance since NYU does not provide an easily accessible platform to do so. This is problematic since in many NYUSH classes, attendance is part of the grade. Using online solutions such as TeacherKit, MyAT, and TrackCC are a possibility, however, these solutions are not compatible with the school's student management system (Albert/NYU Classes). This means that professors would have to manually add each student's names, pictures and create classes for each of the courses that they are teaching that semester. Not only that, but using online solutions also might go against privacy and commitment regulations from FERPA.

1.2 Objective

The goal of this project is to create an efficient, secure and reliable attendance tracker. The project aims to be useful for professors and save them time and energy. It is easily accessible as a web application with a mobile interface, so that app installations will not be necessary. It also aims to be easy to manage and use, with little setup required by the professors. Initially I aimed to limit the amount of setup required by directly calling from the school's database through an API. This way all information needed would automatically be setup in the product. However, due to school regulations, this is not a possibility as calling directly to the school's database is not allowed.

2. Related Work

In order to better understand how to design the project, several applications and a few professors were surveyed. The selection of existing applications is based on current user base size and differing features. Professors were selected based on if they take attendance in class. In order to obtain more diverse and significant feedback, one professor with no computer science background, one professor with a basic understanding of technology, one professor with extensive knowledge on user interface design and one neutral professor with a computer science background were chosen. In addition, several frameworks and libraries were compared to decide compatibility with the project. Selection of web development languages is based on the following factors: how widely they are used, ease of use, how they accomplish different tasks (UI rendering, architecture flexibility, community support, etc), and the extent of available documentation.

2.1 Existing Solutions

Table 1: Application Comparisons

Name	Subscription	Platforms	Security	Sync	Im/Export	Usage	Difficulty
TeacherKit	Free/Pro \$39.99	I,A,W(Pro)	Pro	Pro Only	Partial	1M+	Easy
MyAT	Free/Pro*	I,A,AT,W,M	SSL, Pro	Auto	Both	50,000	Hard
TrackCC	Free	I,A,AT	SSL, Login	Auto	Export	??	Moderate
Mitrefinch	Pro	I,A,AT,WP,W,M	Login	Auto	??	1M+	Moderate

iOS (I), Android (A), Android tablet (AT), Windows phone (WP) Windows (W), OS X (M) *Donation Based

2.1.1 TeacherKit

TeacherKit is one of the most popular attendance tracker applications with over 1 million downloads. When opening the application for the first time, it gives the user a tutorial explaining how to use the application. However, out of the four applications I compared, TeacherKit has the most limited free features available, and feature availability differs between platforms. Features such as attendance timetable and lesson addition are not available in Android phones nor the Windows web application. The web application is also available only to Pro users, which costs 39.99 USD monthly. Custom behavior, advanced gradebook, report features and syncing between devices are only for Pro users with Apple products and are unavailable otherwise. Import and export features are only available to Apple products, as well as data backups to Dropbox. Overall, TeacherKit is a product that has many great features that make attendance tracking easier and quicker, however, a lot of its interesting features are only available to users who pay the monthly subscription, and is even more limited than other products for free users [29].

2.1.2 MyAT

My Attendance Tracker is a great attendance taking tool. It is used by over 50,000 teachers and groups to track over 650,000 students. MyAT is available on most devices and is almost completely free. There are Pro features, however, Pro subscription is donation based, starting from as low as 1 USD per year. Pro features include student and parent login features, invoicing and priority customer support. Unlike other attendance trackers, all of MyAT's attendance features are free. It has student, class, grade, and parent managing features as well as messaging through the application and sms, import/export of csv files, student profiles, and attendance reports. Syncing between devices is automatic and security features include login and SSL encryption between the server and the client.

Overall, MyAT is the most complete attendance tracker application, but it is also more difficult to use than the other applications [5].

2.1.3 TrackCC

TrackCC is a completely free attendance tracker; however, it is currently only available on desktop and Apple devices. Many of the features available in MyAT are also available in TrackCC, except for the import feature. Classes, students, and all other information has to be manually added one by one, making it tedious and difficult to use. While reports are available for viewing, they cannot be downloaded unlike in TeacherKit and MyAT. Although the desktop application is not fully developed nor simple to use, the mobile application is easier to use than MyAT [30]. The interface for TrackCC requires significantly more user input than TeacherKit, which makes it harder to use.

2.1.4 Mitrefinch

Unlike the other three applications, Mitrefinch's attendance tracking software is for businesses. With over 1 million daily users, Mitrefinch's software is a popular option for both small and large businesses such as Nestle, Napa, Siemens, HSBC among others. It is a complete workforce management tool, which does not only track and manage employee hours, but also manages absences and workforce scheduling. The software provides real time built in reporting into HTML5 configurable Dashboards and has a configurable BI reporting tool. Its automated alerts and notification makes it convenient for both employers and employees. In addition absence profiles and timesheet bands allow for easy visualization of data [7]. It is only subscription based and no feature can be used for free. It can be used in many more platforms than the other attendance taking tools. Import and export features are not specified. On first use, it can be more difficult than TrackCC, however, it has a simpler interface than MyAT.

2.2 Frameworks

Table 2: Javascript Libraries/Frameworks

Name	Git Stars + Forks	Architecture Flexibility	Rendering Speed	Stack Overflow Tags	Binding	Type
Angular	90k+	No	Slow	250k+	Two way	Framework
React	112k+	Yes	Fast	84k+	One way	Library
Ember	22k+	No	Slow	22k+	Two way	Framework
Vue	106k+	Yes	Fast	16k+	One way	Library



2.2.1 Angular JS

With over 90,000 Git Stars and Forks, AngularJS is one of the most popular web development frameworks out there. Angular is a Google developed and maintained TypeScript-based Javascript MVW (Model-View-Whatever) framework. It is used by Google, Wix, Forbes, and many other companies. One of Angular's best features is the easy to test code. To create large and maintainable codebases, Angular heavily relies on dependency injections: providing objects the dependencies they need [3]. While this allows for components to be isolated and easily tested, it is relatively difficult for developers without a strong engineering background to properly utilize. In addition, although directives (its core feature) allow for HTML pages and scripts to be extremely organized, its mandatory use, on top of dependency injections, leads to complex syntax, which makes it hard to learn and debug scopes [23].

One of Angular's biggest advantages is its quick code production, quick prototyping and dynamic single page app creation. Due to the two way data binding with the DOM, Angular allows for quick and responsive websites [3]. The way that Angular achieves two way data binding is by having a digest cycle (aka. dirty checking). Any watched variable is scanned for changes on each digest cycle, and when a change occurs, it is rendered on the DOM [22]. The rendering speed of their websites is also relatively quick compared to other frameworks (although slower than React and Vue). However if there are many watched variables on one page, then the rendering speed will slow down significantly. A separate data binding is created for each rendered field in the DOM, and each field is checked before it is re-rendered [9].

Angular has three main components for its architecture which allow the code to be structured into building blocks: controllers, which work out the logic behind the UI'S, services, which take care of communications with the back end, and directives, which store useful and reusable HTML components [3]. These components make for a very specific architecture developers have to follow, that is, developers do not have freedom when deciding to build their product's architecture. This is good for developers that prefer to structure their architecture in one specific way, however, for developers that want more structural freedom, this becomes a problem [21]. In addition, Angular's native router allows for only one ng-view (a directive that renders the template that allows for switches between different views) and is relatively limited [3]. If developers want to include more than one view in the template, they are forced to use Angular's UI-Router, which provides a state machine and allows multiple views on a single page [4]. Angular has one of the largest communities, countless resources available to help developers, and over 250,000 questions tagged on StackOverflow [33].

2.2.2 React JS

React is a Facebook developed and maintained MVC (Model-View-Controller) Javascript library for building user interfaces. It has over 112,00 git stars and forks and is used by big companies such as Facebook, Airbnb, Uber, Twitter, in addition to many other companies. React's library is for the building of composable UI. It focuses on the creation of reusable UI components which present data that changes over time [8]. Components also do not affect each other since they are isolated [18]. These components can be reused at different levels which allows developers to easily update their code. React's high DOM rendering speed is due to the use of a virtual DOM or vDOM representation, and saves the DOM tree on the server side. This allows for a highly dynamic UI. React is commonly used with JSX, a Javascript extension that creates React elements [18]. React also has downward data binding which means that changes in child structures do not affect parents. This one directional data flow creates stable and scalable code [18]. This means that React first updates the model and then renders the UI elements [26]. Unlike Angular and Ember, React has a significantly higher flexibility regarding product architecture. Not having building blocks to structure their code into allows developers to freely design the architecture instead of following strict architectural rules.

However, despite React being a large sized library at 43K, it lacks full framework capabilities such as built-in database query tools, a state management system, a router, and model management libraries [27]. React on itself can be thought of as the view in MVC instead of a complete framework like Angular and Ember [18]. Because of this, React is usually complemented by Redux, and forces developers to learn another library in order to be able to create a complete product. In addition, documentation may be hard to understand for new developers. Despite having over 84,000 questions tagged in Stack Overflow, the development of React is also fast paced. The constant changes make it hard for new developers to keep up and many of these Stack Overflow answers are outdated [30].

2.2.3 Ember JS

Ember is a non-company developed Javascript framework currently owned and maintained by Facebook "for creating ambitious web applications" [17]. It has over 22,000 git stars and forks and it is used by Microsoft, Intercom, LinkedIn and many others. Ember is a high performance, heavy duty front-end Javascript framework that can build complex web applications. It focuses on creating scalable projects. Due to Ember's scalability goal, it is less flexible than Angular and has more strict rules, but guarantees for a scalable project. One of Ember's strongest feature is its full featured object model. It has an integrated development toolkit, Ember CLI, that allows for rapid iterations [11]. It has a strong built in routing and templating engine with Handlebars, and a data layer that makes it easy to query external APIs and manage application states [17]. When defining a router resource,

the routes' names and controllers are automatically determined. However, Ember Data's default Adapter has built-in assumptions of how a REST API should look. This means data that does not follow REST standards, API endpoints that return data differently, and ajax requests all have to be managed using jQuery middleware [17]. Ember's web components are very simple, such as UI widgets or components, which makes them easy to understand [23].

Ember also has two way data binding, which can be used with any object. The bindings are not updated immediately. Instead it waits until all the code runs, allowing for a bounded property to be changed freely [17] Ember also has its own testing and debugging tool (Ember Inspector), which allows for easier development and debugging [12]. The debugging tool is especially useful since Ember does not have "dirty checking" [9]. Getters and setters have to all be remembered in order to access object properties otherwise it may lead to bugs when developing [23]. Ember has a concept called The Run Loop, which allows for pages to be quickly rendered and improve application performance. The Run Loop is used to batch and order work efficiently and does so by scheduling work on specific queues, which lowers the number of expensive actions [17].

However, all the conventions and features that are built into Ember make it hard to use. Its steep learning curve is regarded to be higher than Angular's. Although the documentation is well made, the community is relatively small. The Stack Overflow tagged questions only amount to 22,000 [33]. Due to the several changes in the past few years, a lot of resources are outdated and new developers mainly have to rely on the official documentation. Most importantly, Ember has little space for customization and control of code. It prefers convention over configuration, which automatically generates resulting code as it favors following naming conventions [9]. In addition, it has a heavy framework, which is too large for small projects. This makes it not very suitable for small applications within websites, as well as the fact that Ember expects entire pages to be run. Furthermore, the mandatory use of Handlebars within the HTML may make the DOM look messy [9]. Although the templating engine allows for quicker development and reduces overall amount of written code, Ember should be mainly used for big projects that need scalability.

2.2.4 Vue JS

Vue is a non-company developed and maintained Javascript library for building interactive interfaces. It is used by Alibaba, Baidu, Expedia, Nintendo, GitLab and more. Although small sized at 23kb (after gzipping), its ecosystem is fast and consistently outperforms larger frameworks like Ember and Angular [21]. Vue has fast performance, partially due to its vDOM, and has deeper memory allocation than Angular and React [27]. It is easy to understand and develop, which makes it convenient for small developing teams to quickly create robust websites [27]. Due to the possibility of developing large scale templates, a lot of time can be saved by using Vue. It has the ability to create single page applications and



integrate components into already existing applications. Vue is very responsive and flexible as it allows for easy implementations of one way flows and creation of components which can be reused in several other projects. It can contain a component's structure, logic and style in one file [17]. Its flexibility allows for many different kinds of packages to be paired with the library [26].

Vue's one way data flow between components allow for easy understanding of the flow of data in applications. Because they are reactive, Vue variables are observable and can have watchers attached [21]. Its reactivity system is one of Vue's most distinct features. It keeps track of the state and automatically re-renders the DOM when mutated. However, property additions, deletions and array changes cannot be detected [15]. Data properties and computed properties are configured automatically when created, therefore, it saves development time and allows for terse code. However, this leads to bugs that are hard to spot and fix. Unlike Angular, Vue's directives and components have a clear distinction. Directives only encapsulate DOM manipulations and components are self contained units with their own view and data logic [21].

Although Vue has a well written and comprehensive documentation, Vue does not have the extensive community support React and Angular has. As seen on Stack Overflow, only 16,000 questions are tagged with Vue.js. Vue's creator Evan You, is a Chinese American who is very supportive of the Chinese development community [33][10]. Although it has over 106,000 git stars and forks, most of these are from the Chinese community. This sometimes creates a language barrier for English speaking developers since most of the code and support is written in Chinese [10].

2.3 Summary

There is no existing attendance tracker that can be easily implemented at NYU. Although all of the solutions have had extensive development, the biggest issue is the need to individually input information to take attendance. The application I aim to create is similar to these existing solutions, but will also address this issue.

Although there is no perfect framework or library, Vue would be the best fit for this type of project. Considering the size of the project, it does not seem to need a full framework such as Angular and Ember. These frameworks are also relatively hard to learn. In addition, Vue would allow for the quickest and simplest development of the project. Even though the community is not large, the documentation is simple to understand and detailed enough to not need other tutorials. React would also be a good fit for this project, however, its documentations and manuals are relatively more complicated to understand, which could delay the development process. The necessity of complementing React with Redux would also complicate the project's development process more than necessary.



3. Solution

3.1 Preliminary Study

The project has three major parts. The first part of the project focused on researching existing solutions and surveying professors. After extensive surveying of professors, I clearly know what professors need and what should be included in the project. I am able to see which aspects of existing solutions meet the requirements and which do not. With additional research, I am also able to determine what frameworks work best for the project with regards to front and back end development as well as security. I also worked with Registrars and IT in order to determine how to access the school's database and to ensure that the application can be adapted as necessary for future semesters. Then, I am able to design the architecture of the application.

The second part consists of creating a server side program which fetches information needed for each professor: the student's names, pictures and professor's current classes. Creating a secure database is also an important part in this step of development. For security reasons, there is a login and logout feature that allows professors to be the only ones accessing the attendance tracker's features and data.

The last part of the project consists of creating an easy to use interface (UI) that best fit the professors' needs. The project is largely based on the UI of the application. There is significant user testing by working directly with several professors. Professors tested out the application and provided feedback on several things: ease of use, time to take attendance, usability etc. These surveys are used for assessment of qualitative factors such as satisfaction of use with regard to previous methods, in addition to quantitative results such as time spent on taking attendance.

3.2 Architecture

The solution I designed is a full stack application with MEVN architecture, which stands for MongoDB, Express, Vue and Node [24][13][21][1]. For the UI framework, Vuetify.js is used [32]. CRUD (create, read, update, delete) operations are done with Express.js and MongoDB using mongoose. REST API calls are done using Axios, a http client library [6].

For the front end, the project uses Vue.js. Vue's reactive data binding system allows for a dynamic application. As shown on [21, Figure 1], Vue will bind the data to the DOM (document object model) view. Once the data is bound, the view will listen for changes in the data. Once the data is changed, the view will reflect that change in the data [21]. This way, the application does not have to be refreshed for data changes to be shown on the view.

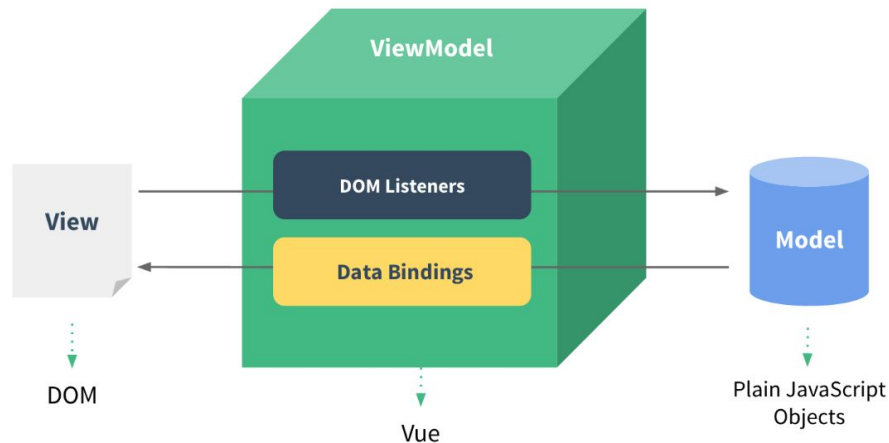


Figure 1: Vue Architecture

For the backend, as shown on [21, Figure 2], using a REST API call built with Express and Node will allow me to interact with the MongoDB Database [19]. Node.js is a runtime environment that runs the back-end application using Express. MongoDB is a document database which stores JSON documents in collections [23]. Node and MongoDB both work asynchronously, waiting for requests and returning requests. When a request occurs, Node will concurrently run other tasks until it receives a response. Once the response is received, Node will return the response to the front end [1]. Using Express' REST API, we can access the database using HTTP network calls [25].

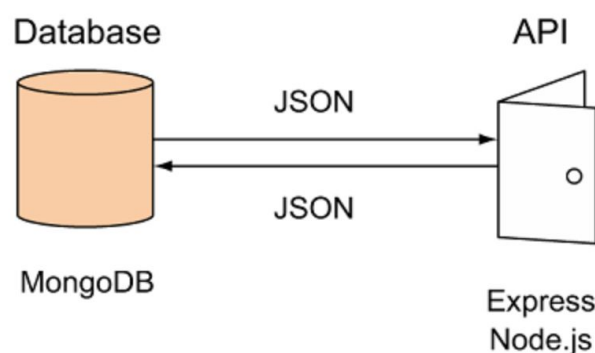


Figure 2: Express, Node + MongoDB interaction

[28, Figure 3] shows the whole architecture of the project. The front end side of the application will consist of several components where the HTML, CSS and JavaScript exist. Express will be running the back end within Node's environment, which will "handle the routing of requests" [25]. Using axios, I can call the Node API and then the API will post a json object to the database. The database will then return a json object. This is done using promises with async and await functions from the client side. In the server side, the request will be made to the database and return the data.

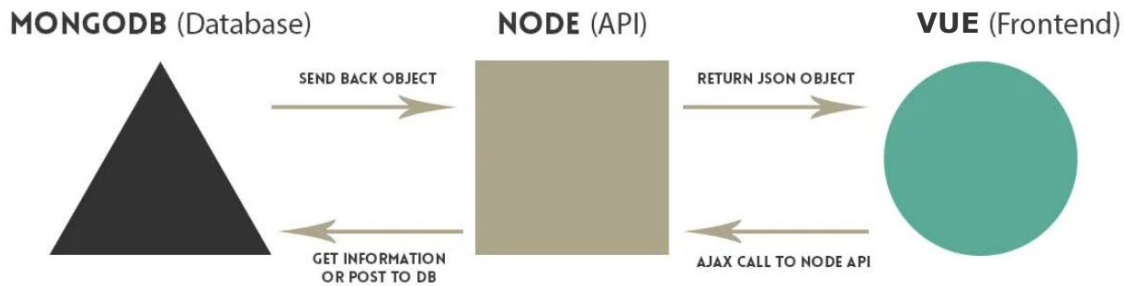
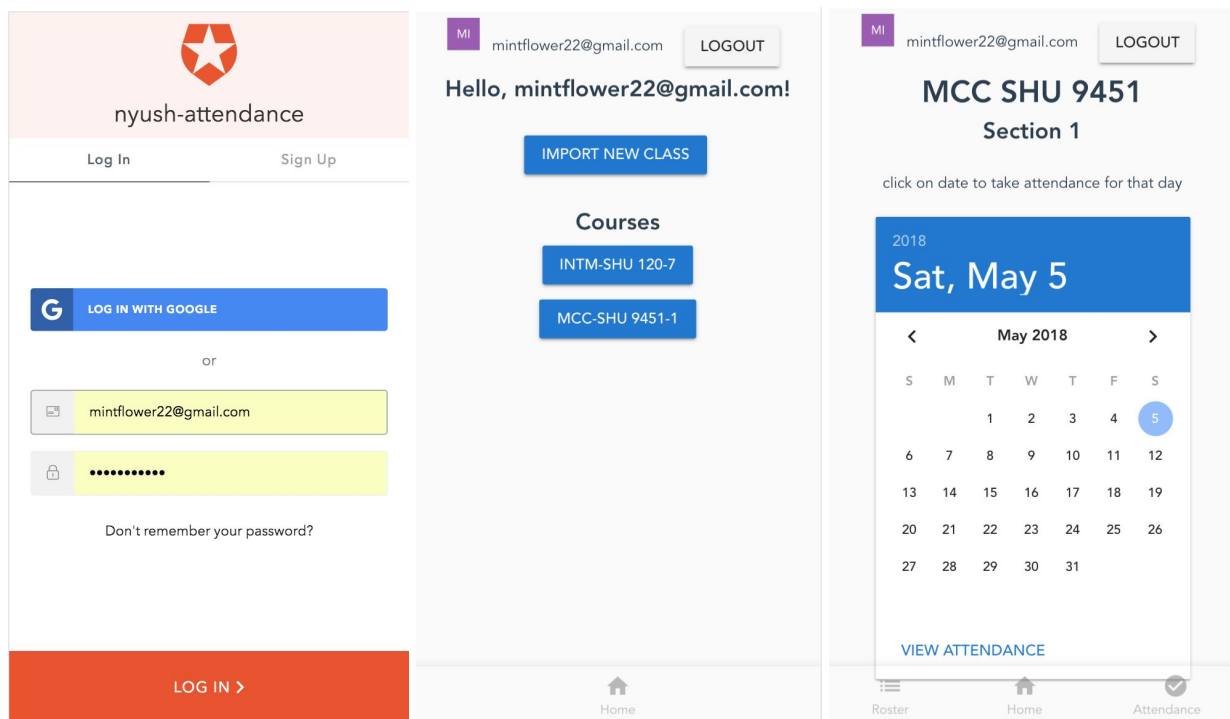


Figure 3: Application Architecture

3.3 Features

There are eight main views to this product: login/sign up page, welcome page, import page, home page, roster page, view today's attendance, take attendance page and a student profile page. A navigation bar and a menu bar persist throughout all the pages. Users are able to logout from the menu bar, and navigate around main pages with the bottom navigation bar.



3.3.1 Login/ Sign up Page

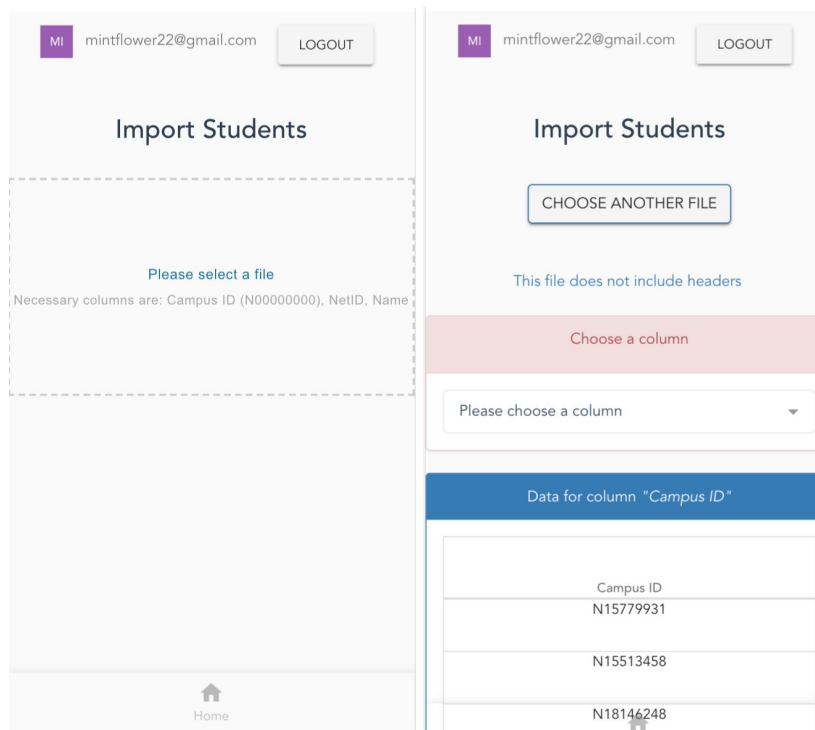
Users are able to login or sign up from this page. Unless a user logs in, they will not be able to access any of the contents of the product. Users can sign up with either their email or google account. Once they sign up, they will receive an email confirmation.

3.3.2 Welcome Page

This page welcomes the user and shows the list of classes that have been imported. If there are no existing classes, then the user will be directed to import a new class.

3.3.3 Home Page

If the user already has classes imported, then they can choose the class they want to take attendance for, and will be redirected to the home page. Here, the user will see a calendar where they can choose what date's attendance they want to take or view. Once they choose a date, they will be redirected to the attendance viewing page for that day.



MI mintflower22@gmail.com LOGOUT

Import Students

Please select a file

Necessary columns are: Campus ID (N00000000), NetID, Name

CHOOSE ANOTHER FILE

This file does not include headers

Choose a column

Please choose a column

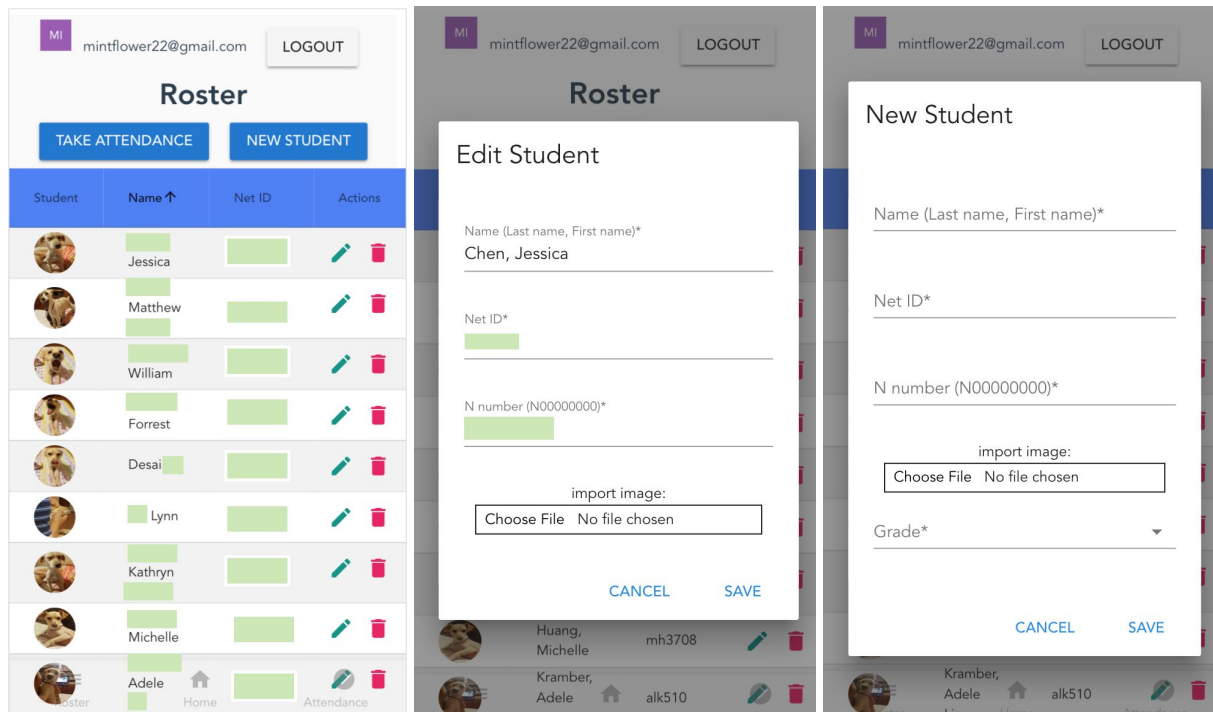
Data for column "Campus ID"

Campus ID
N15779931
N15513458
N18146248

Home

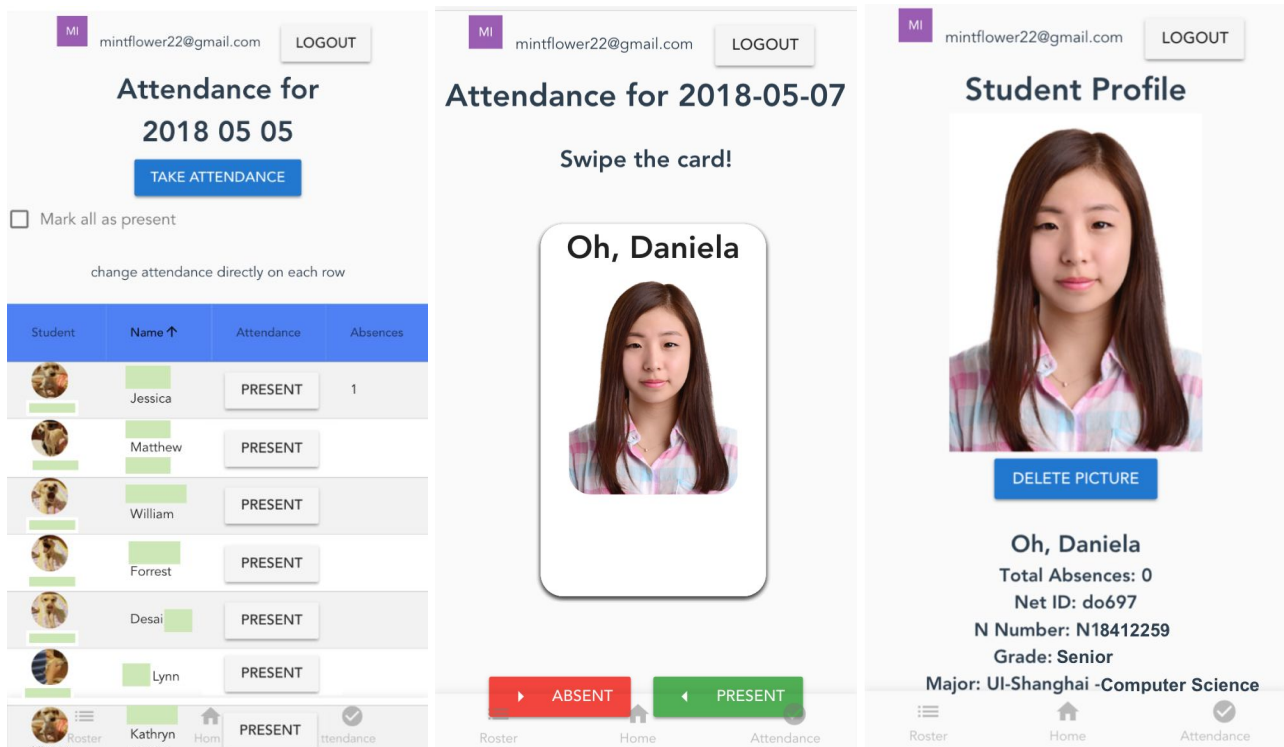
3.3.4 Import Page

This page allows professors to import a course by matching the excel or csv headers to the premade headers. The headers of the excel sheet will have similar names as the headers in the product. All professors need to do is match the names of these headers to each other. After an import is successful, the user will be redirected to the roster page for the newly imported course.



3.3.5 Roster Page

The roster page is the page with the most features. Here, users can view the roster for a specific class. Users will be able to edit basic student information and delete students. New students can also be added. Student pictures will be shown, however, if there are no images, users can upload one directly from their gallery, or take one with their devices.



3.3.6 View Attendance Page

In this page, users will be able to see a list of students of a specific class. Users will be able to quickly mark all students as present and edit each student's attendance type for that date in-line. From this page, users can also be directed to the attendance taking page.

3.3.7 Take Attendance Page

Students will be shown in cards with their names and pictures. Users can take attendance for each student by swiping left for present and right for absent.

3.3.8 Student Profile

From the roster page, users can click on a student's name or picture to be taken to their profiles. Here professors are able to upload images of the student, delete the current image and view student information.

4 Evaluation

Four different professors in different fields participated in user testing and surveying: Professor A, Professor B, Professor C and Professor D are surveyed. Professor A is a professor with extensive knowledge on UI design. Professor B is a middle aged professor with less technical knowledge. Professor C is a professor with no computer science background and basic technological knowledge. Professor D is a professor with extensive technological knowledge. In order to assure quantitative results from these surveys, all questions were answered in a score between 1 to 10. For qualitative results, all questions were asked to provide reasons for the scores given.

4.1 Survey Results

Table 3: Survey Results

	Professor A	Professor B	Professor C	Professor D	Average Score
Navigation Flow	8	8	9	7	8
Usability	7	10	9	6	8
Best Feature	attendance taking	attendance taking	attendance taking	attendance taking	9.25
Ease of Use	8	7	9	6	7.5
Clarity	6	8	8	8	7.5

*All questions are on a scale of 1 to 10 (1 = bad, 5 = average, 10 = excellent)



Question 1. How clear was the navigation flow (ie. can you find what you are looking for)? How easy was it to understand how to go from one page to another?

All answers suggest that there is a clear navigation flow. Feedback includes “easy to navigate”, “intuitive”, “useful navigation bar”, “easy to understand”. Professor D mentioned that he prefers to have a more highlighted navigation bar.

Question 2. How likely would it be for you use this product in your own class long term? Why? If no, what changes would make you want to use this product?

All professors prefer to use this product over their current methods. Professor D, without realizing the “mark all as present” feature, gave it a 6 because “it's okay if you have a small classroom”. But Professor D said he would use it “because it's better than what I'm using right now”. Professor A said the interface is easy to use and gives it a 7 because it would be easier to use if the information were synced with NYU Classes. Professor B mentioned that adding student pictures made the product a lot better to use than her current method.

Question 3. What features did you find most useful? Which features did you find not necessary? What features would you like to see in the future?

Not all professors answered part two and three of this question. All professors mentioned that the attendance taking feature is the most useful and gave that feature an average of 9.25. Professor A said “swiping to take attendance is very convenient”. Professor D also complimented the swiping feature. Professor C finds some headers in the import feature unnecessary, however, all these headers are used in the product and are as important as student ID.

Question 4. How much time, resources, and or energy does this product save you when taking attendance? How about saving attendance record (present, absent, etc)?

Professor C had internet connectivity issues but said that if the connectivity were not a problem that the product “will save [him] a lot of time”, thus giving it a 9. Same as question 2, Professor D said it will save time for smaller classrooms, without realizing the mark all as present feature. Professor B gave it a 7 because of the initial setup required at the beginning of the semester, but “after that it is a 10”.

Question 5. When you explored the product, were there any points where you became confused? If so, where and doing what/looking for what? (1 = very confused, 10 = not)

On average the professors do not find the product very confusing. The features they find slightly confusing are the import page and the navigation to the view attendance page from the home page. Professor C finds importing the excel sheet to the product confusing and says he does not know how to match the headers sometimes. Professor A said he is not sure why the home page's calendar leads to the view attendance page instead of directly to the take attendance page.



4.2 What it means

The aim of this product is to be useful for professors who take attendance. It is created to save time, energy and resources so that professor do not spend a lot of time taking attendance and recording that data after class. The product is meant to facilitate the attendance taking process such that it would not be cumbersome for professors to do so. The success of the product is evaluated by how well it accomplishes these goals and if professors would rather use this product than their current methods.

The survey results indicate that the product achieves its goal. Although some features did not get a 10, all professors said they would prefer to use this product than their current methods. With an average of 9.25 professors give the attendance taking feature a high score. Professors also find the product to save more time and energy than their current methods.

Although the average score for question 4 is not outstanding, professors explained the reason for their scoring. Professor D, not realizing that there is a mark all as present feature, led him to give a 6 for how much time is saved using this product. This is due to the fact that the feature is not noticeable enough because of poor UI design. If the feature were more noticeable, then the professor would have given it a higher score. Professor D says that this product would be good for small classes, but if the class size were over 50, then it would be too inefficient to swipe for every student. In addition, Professor A and B at first did not notice that in-line editing for attendance is possible, and suggest making this feature more noticeable too. Although the in-line editing feature was changed to be made more noticeable, due to the lack of time, an extra round of user testing was not possible.

Overall, small changes like these are simple to implement. These changes would not only improve the product's quality but also user experience. But according to survey results, the current product already proves to be more useful to professors than their current methods and make taking attendance less time consuming.

5 Lessons Learned

To improve the process of completing the project, I would first have more robust pre-planning. All the implementations needed for each page in this project would have been designed first. The architecture of each page would have been designed in detail to allow for more concise and clean code. Knowing each of the components and features needed for each page would allow me to plan what and how to code before having to code it. This way, a lot of repeated code would have been made into importable components. Designing the database before creating it would have allowed me to know how the data would look like from the beginning. By doing so, I would have prevented going back and forth adding attributes every step of the process. I would also make sure I have a domain name set up from the beginning, so the DNS propagation does not become a problem.



While making the product, I would have also focused on creating the login feature first so that future code would be built around the login. In my project, the login feature was implemented midway, so a significant amount of code refactoring was done. I would also want to test deployment from the beginning so that I can test my complete product on mobile instead of only on desktop. Most importantly, even before the minimum viable product is done, I would want to user test so I would know more clearly what the user's needs are at each step of the process.

6 Conclusion

6.1 Contributions

In this project, I attempt to ease the attendance taking process for professors in NYU Shanghai. Using a MEVN (Mongo, Express, Vue, Node) architecture for the website, I attempt to create a simple and easy to use product for professors. From the ability to take attendance quickly and painlessly to saving attendance data and getting basic student information, this project ideally makes attendance taking effortless. Although fetching from the school database as an API was not a possibility, professors are able to download excel files from NYU Classes very easily. Thus, the import function is set up to match this file in order to make class data import quick and painlessly. As opposed to existing solutions where professors have to match their data to their format, my project accommodates to the format of the professor's data. Unless the school allows for this whole project to be added to NYU Classes, not many changes can be made to speed up the initial setup. Overall, the feedback is positive and professors all say they would prefer to use this product over their current methods. Not only because it saves time but also because it is simple and "intuitive" to use. I therefore believe that this project has been successful and more importantly, useful.

6.2 Future Implementations

Several features can be added to further extend this product. An important feature is a name pronouncer and a nickname adder. Half of the student population in NYU Shanghai have Chinese names, and many professors do not know how to pronounce them. Having a feature where the professor can listen to the student's name will be useful. Also, adding student nicknames as a side note would help professors link the student's full name with their nicknames.

In order to improve the usability of the product, I would also like to add customizable features. For example, allowing professors to add notes or types of excused absences such as illness, travel, school event or family emergency will make it so that professors are not limited to the current attendance types. This way, professors will not have to remember why the student was excused for that particular day. Also, for professors that have trouble remembering student names, adding a customizable seating chart would help them



remember the students. In many classes, students sit in the same spot for the whole semester. Adding this feature where professors can recreate the seating of the classroom and see which student seats where, would increase usability.

In addition to simply being an attendance tracker, having grading features and linking student documentation and assignments would make the product more useful. Since in many courses, attendance is part of the grade, integrating grading according to how many absences a student has would save professors time. A customizable grading tool where they can add these rules and have the product automatically calculate the student's attendance grade would prove to be useful. Also, a feature where the professors can add a link to each student's documentation, home page, google drive shared folders page or NYU classes page. This way, instead of having to find their Net ID's or N Numbers and then finding that page, professors will be able to click one button and be redirected to that page automatically. This is especially convenient for professors that asks students to upload their assignments on blogs or the IMA documentation site.

Lastly, all the suggested features from professor feedback would be added. A more highlighted navigation bar, a less cumbersome import page, a more clear navigation from the home page to the attendance taking page and more a visible UI for important features such as "mark all as present". Due to the common problem of professors not noticing a feature, I would like to add a short in-app tutorial that will guide professors on how to use the product. This would solve a lot of the problems the professors mentioned in the feedback and would give a better user experience.



7 References

- [1] "About Node.js," *About Node.js*. [Online]. Available: <https://nodejs.org/en/about/>.
- [2] anaida07, "anaida07/MEVN-boilerplate," *GitHub*. [Online]. Available: <https://github.com/anaida07/MEVN-boilerplate>.
- [3] "Angular," *Angular Docs*. [Online]. Available: <https://angular.io/docs>. [Accessed: 07-Feb-2018].
- [4] angular-ui, "angular-ui/ui-router," *GitHub*. [Online]. Available: <https://github.com/angular-ui/ui-router/wiki>. [Accessed: 07-Feb-2018].
- [5] "Attendance Tracking Online: MyAttendanceTracker," *Attendance Tracking Software Online*. [Online]. Available: <https://www.myattendancetracker.com/>. [Accessed: 07-Feb-2018].
- [6] Axios, "axios/axios," *GitHub*. [Online]. Available: <https://github.com/axios/axios>.
- [7] "Best Time and Attendance System - Accurately Capture Employee Hours," *Advance Systems*. [Online]. Available: <https://advancesystemsinc.com/time-and-attendance/>. [Accessed: 07-Feb-2018].
- [8] C. Bryant, "Hello React and TypeScript," *React | Hello React and TypeScript*. [Online]. Available: <https://charleslbryant.gitbooks.io/hello-react-and-typescript/content/React.html>. [Accessed: 07-Feb-2018].
- [9] N. Bubniuk, "Comparison of JS Frameworks: AngularJS vs. React.js vs. Ember.js - DZone Web Dev," *dzone.com*, 03-Nov-2016. [Online]. Available: <https://dzone.com/articles/comparison-of-js-frameworks-angularjs-vs-reactjs-v>. [Accessed: 07-Feb-2018].



- [10] V. Cromwell, "Between the Wires: An interview with Vue.js creator Evan You," *Between the Wires: An interview with Vue.js creator Evan You*, 30-May-2017. [Online]. Available: <https://medium.freecodecamp.org/between-the-wires-an-interview-with-vue-js-creator-evan-you-e383cbf57cc4>. [Accessed: 07-Feb-2018].
- [11] "EmberCLI; User guide," *Ember CLI*. [Online]. Available: <https://ember-cli.com/user-guide/>. [Accessed: 07-Feb-2018].
- [12] Emberjs, "emberjs/ember-inspector," *GitHub*. [Online]. Available: <https://github.com/emberjs/ember-inspector>. [Accessed: 07-Feb-2018].
- [13] "Express - Fast, unopinionated, minimalist web framework for Node.js," Express - Node.js web application framework. [Online]. Available: <https://expressjs.com/>.
- [14] "FAQ: What is SSL?," *What is SSL?* [Online]. Available: <http://info.ssl.com/article.aspx?id=10241>. [Accessed: 30-Apr-2018].
- [15] A. Gore, "React or Vue: Which Javascript UI Library Should You Be Using?," *React or Vue: Which Javascript UI Library Should You Be Using?*, 22-Dec-2016. [Online]. Available: <https://medium.com/js-dojo/react-or-vue-which-javascript-ui-library-should-you-be-using-543a383608d>. [Accessed: 07-Feb-2018].
- [16] Goweijen, "goweijen/vue-swing," *GitHub*. [Online]. Available: <https://github.com/goweijen/vue-swing>.
- [17] "Guides and Tutorials Edit Page," *Ember.js - Guides and Tutorials*. [Online]. Available: <https://guides.emberjs.com/v3.0.0/>. [Accessed: 07-Feb-2018].
- [18] "Hello World - React," *A JavaScript library for building user interfaces*. [Online]. Available: <https://reactjs.org/docs>. [Accessed: 07-Feb-2018].
- [19] S. Holmes, "A Practical Guide to Planning a MEAN Stack Application," *A Practical Guide to Planning a MEAN Stack Application*, 17-Jul-2017. [Online].



Available: <https://www.sitepoint.com/planning-mean-stack-application/>.
[Accessed: 30-Apr-2018].

- [20] "Identity infrastructure, built for developers," *Auth0*. [Online]. Available: <https://auth0.com/>.
- [21] "Introduction - Vue.js," *Vue.js*. [Online]. Available: <https://vuejs.org/v2/guide>.
[Accessed: 07-Feb-2018].
- [22] A. Lerner, "The Digest Loop and \$apply," *Ng-Book -The Complete Book on AngularJS*. [Online]. Available: <http://www.ng-book.com/p/The-Digest-Loop-and-apply/>. [Accessed: 30-Apr-2018].
- [23] C. McKeachie, "AngularJS, Backbone.js or Ember.js – Which One to Choose for Your Project?," *AngularJS, Backbone.js or Ember.js – Which One to Choose for Your Project?*, 09-Dec-2014. [Online]. Available: <https://www.codementor.io/javascript/tutorial/which-javascript-mvc-angularjs-backbone-ember>. [Accessed: 07-Feb-2018].
- [24] "MongoDB for GIANT Ideas," *MongoDB*. [Online]. Available: <https://www.mongodb.com/>.
- [25] A. Morgan, "Introducing the MEAN and MERN stacks," *Introducing the MEAN and MERN stacks*, 26-Jan-2017. [Online]. Available: <https://www.mongodb.com/blog/post/the-modern-application-stack-part-1-introducing-the-mean-stack>. [Accessed: 30-Apr-2018].
- [26] J. Neuhaus, "Angular vs. React vs. Vue: A 2017 comparison ," *Angular vs. React vs. Vue: A 2017 comparison*, 28-Aug-2017. [Online]. Available: <https://medium.com/unicorn-supplies/angular-vs-react-vs-vue-a-2017-comparison-c5c52d620176>. [Accessed: 07-Feb-2018].
- [27] J. Schatz, "Why we chose Vue.js," *Why we chose Vue.js*, 20-Oct-2016. [Online]. Available: <https://about.gitlab.com/2016/10/20/why-we-chose-vue/>.
[Accessed: 07-Feb-2018].



- [28] C. Sevilleja, "Creating a Single Page Todo App with Node and Angular," *Creating a Single Page Todo App with Node and Angular*. [Online]. Available: <https://scotch.io/tutorials/creating-a-single-page-todo-app-with-node-and-angular>. [Accessed: 01-May-2018].

- [29] "TeacherKit," *TeacherKit*. [Online]. Available: <http://teacherkit.net/>. [Accessed: 07-Feb-2018].

- [30] "TrackCC | student attendance, behavior and grading," *TrackCC | student attendance, behavior and grading*. [Online]. Available: <https://www.trackcc.org/>. [Accessed: 07-Feb-2018].

- [31] Victorboissiere, "victorboissiere/vue-xls-csv-parser," *GitHub*. [Online]. Available: <https://github.com/victorboissiere/vue-xls-csv-parser>.

- [32] "Vue.js Material Component Framework - Vuetify.js," *Vue.js Material Component Framework - Vuetify.js*. [Online]. Available: <https://vuetifyjs.com/>.

- [33] "Where Developers Learn, Share, & Build Careers," *Stack Overflow*. [Online]. Available: <https://stackoverflow.com/>. [Accessed: 30-Apr-2018].