Greg Kaplowitz

# Mobile App Term Paper

When we first started this project we had the idea to make checking your class schedule more easily. We saw the problem where the only way to have a schedule that would check for day shifts(Monday schedules) was to log into SIS and view your schedule for the week. The problem is that SIS is a secure student database that contains all student records as well as student medical information. This requires SIS to have a lot of security measures to ensure that unauthorized parties aren't able to get access to this information; however, it also means that simply logging in to check your schedule in the morning takes a lot more time than many people are willing to do. Another problem we saw was that the academic calendar has a lot of useful information that could be important for students to see if they are checking their schedule at the beginning of the day/week. We had the idea to make an app that would solve both of these problems. By having the user enter in their class schedule and then using the school's academic calendar we could make an app that has all the information that students would want to see if their schedule. When we started thinking about how we were going to implement our app we started by using android studio and using jetpack to develop our user interface. We got the form set up in a way that we were happy with but then we decided to change our approach. After a class where we learned about progressive web apps which have the ability to be run on any computer operating system as well as being able to be downloaded onto any mobile device platform, we decided that we would try doing that instead. This proved to be much better because we were able to use a larger pool of tools like bootstrap to make the form that users would be entering their data into and javascript with JQuery which we could use to do form validation, it also has local storage which is a good database alternative to storing the users class data and academic calendar data so that the app can be used without having internet access. All these benefits made it pretty easy to switch our project over to a progressive web app from one made using android studio. After we started using the progressive web app we

were able to make a lot of progress on the main feature of our app which is being able to display the user's class schedule. Once we had the front end of our app built we started building up the backend this was made much easier by use of the window interfaces local storage. Using the local storage we are able to store the data entered into the form and store it in the web app's local storage so that it could be held for future sessions instead of having to use a database. When the user loads the app it will first check if there is any saved class data to check if this is the user's first time launching the app. If the user has no class data entered then it will automatically go to the enter class screen where the user can enter their class by giving name, school, section, and times. After the user has entered classes and on all future runs of the app, the user will see the calendar screen where it will show the day's classes as well as any useful information for the day on the school academic calendar. We did encounter some problems when developing the app, in the beginning, we were using android studio as previously mentioned which uses XML by default to build the user interface. XML is a very commonplace language for building interfaces but it can be very annoying to use, we tried to offset this by using android studio's jetpack plugin which completely changes the way that interfaces are built. Another problem we had was our worry about useability, since android studio apps only let you make apps that run on mobile devices running the android operating system we are alienating a large potential user base. Many college students don't have android devices or would simply prefer to be able to check their schedules on their computers in addition to their mobile devices. This is when we learned about the concept of a progressive web app that can be run in the browser and can be downloaded to work on any device. If a user wants to be able to use this app on their computer and then pick it up on their phone when they leave for class they are able to do this now. Even if they have a windows computer and an apple phone or a mac computer and a google pixel they will be able to use this app on all their devices. Once we had the progressive web app set up and running we encountered our next issue, how to save the user's data for future sessions. We thought about storing the data in json files and learned that

javascript doesn't have ways to interface well with json files without having a web host set up. We decided that the best way was to use the local storage from the window interface that javascript uses to store data locally to the browser which also gets copied over when being downloaded onto your device. Using the local storage we were able to set up a class that stores the data for each class and save that to the local storage file that can be easily obtained and used to render the calendar when the user opens the app. This also gave us a quick solution to how the app knows whether it is being run for the first time, by simply checking if the local storage file has any data in it already the app knows whether or not to try and render the4 calendar or take them to the class form where they can enter their classes in for the first time. One problem that we weren't really able to solve was the academic calendar. Even though all the dates and information are right there we weren't able to get a solution to automatically get the data off the website and present it to the user. Our solution to this ended up being to simply take the important dates by hand and enter them into the app so it knows when to inform the user. While the original goal was to be able to automatically pull up the dates from the current academic semester for now we were able to able to get the data from the current semester for the sake of time and only taking the data that we deemed important enough to be entered into our app. Our plan for implementing the academic calendar data was to make a new class structure to hold the date as well as the important information in the local storage. The idea is that when the user loads the app and the calendar screen is opened the app checks the local storage file to see if there are any objects for the current day and display the data if necessary. The best decision we made during this project was to make a progressive web app instead of a traditional one, it was important to us that users would be able to use our app on any device so that any student is able to use it. In the beginning, we worried that some users might want to be able to access their calendar on their pc or laptop. Had we ended up developing our app in the android studio it would only run on devices running the android operating system and would therefore not run on users' PCs. Luckily the progressive web app allows for all the

cross-platform compatibility we were hoping to add to our project. The problem that we started with spawned from the issues that we have had over the past few years of trying to view our schedules and share our schedules with fellow students. If we were to continue the development of this app after the semester is over another feature that was important to us was adding a social feature to help solve the issue of sharing our schedules with each other. At the start of semsters groups of students like to find out who is in the chase classes or have the same professors so they can make study groups at the beginning of the semester. We won' be able to complete the social features for this version of the app but my partner has mentioned that he does plan on continuing the development of the app in his next semester so this feature will hopefully be implemented in the next development cycle.