



THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

Student Organization Attendance Tracking

Benjamin Greenberg

Spencer Howell

Tucker Miles

Vicki Tang

Daniel Troutman

Preliminary Design Report
ECE401/COSC401 Senior Design Theory

Tickle College of Engineering
The University of Tennessee
Knoxville, Tennessee
November 20, 2020

Executive Summary

In this report, we discuss the need for an attendance tracking solution and how we can build one to fill this niche. Many student organizations on campus use a plethora of different websites, software, or even written paper to keep track of attendees. People almost always have their smartphones, so we can implement this with an app.

From the attendee's perspective, our app will be as simple as scanning an event QR code and receiving a checked-in notification. From the event manager's perspective, they will be able to generate new events, see who is checked in, and generate reports of the data. We hope that our app can enable both parties to maintain a reliable communication channel for events. Our app revolves around our five engineering characteristics to become a great solution for growing student organizations: security, usability, maintainability, adaptability, and aesthetics.

Table of Contents

Executive Summary	2
Problem Definition & Background.....	4
Requirements Specification	4
Technical Approach.....	5
Design Concepts, Evaluation & Selection	5
Project Deliverables	7
Project Management	7
Budget.....	7
References.....	8
Appendix.....	9

Table Number	Description	Page Number
0	List of Tables	3
1	List of Figures	3
2	Customer Requirements	5
3	Engineering Characteristics	5
4	Platform-Benefits Matrix	6
5	Framework-Benefits Matrix	6
6	Backend-Benefits Matrix	7

Table 0: List of Tables

Figure Number	Description	Page Number
1	Business Model Canvas	9
2	Milestone Timeline	9

Table 1: List of Figures

Student Organization Attendance Tracking

Benjamin Greenberg	Spencer Howell	Tucker Miles	Vicki Tang	Daniel Troutman
University of Tennessee	University of Tennessee	University of Tennessee	University of Tennessee	University of Tennessee
Knoxville, Tennessee	Knoxville, Tennessee	Knoxville, Tennessee	Knoxville, Tennessee	Knoxville, Tennessee
bgreenb3@vols.utk.edu	showel17@vols.utk.edu	tmiles7@vols.utk.edu	wph612@vols.utk.edu	dtroutm1@vols.utk.edu

I. PROBLEM DEFINITION AND BACKGROUND

A. What is the problem? Why is the current situation unsatisfactory?

Our technology is trying to solve a problem that exists in numerous places, but we are primarily focusing on student organizations in the beginning stages. The problem arises from the fact that there exists a great amount of inconsistency in the way that student organizations track event and meeting attendance. As an example, many organizations will have attendees scan a QR code which links to a Google form. Then, attendees will have to spend much more time than necessary just to prove that they have attended an event. This leads to valuable time being lost in what are already typically short meetings and workshops, and could also lead to lowering attendance in the future.

B. Who is having this problem? Who are the would-be customers for a solution?

As mentioned previously, student organizations are having this issue. If thought about much more broadly, these same issues exist in numerous places including workplaces, classrooms, and other similar events. The customers for our solution would be the leaders of these organizations. Our solution would allow them to easily track attendance.

C. What basic functions must the design perform?

The solution must allow organization officers to easily set up events, track attendance at these events, and view overall attendance statistics for each member. This process needs to be seamless, so organizations can quickly take attendance and move on with their meeting. The individual members need to be able to launch the app on their phone, scan the QR code, and receive a success message as they sign in. Other convenience functions, such as removing members from the roster, and marking members as “excused” from meetings, should be supported as well.

D. How will the design be used by the customer(s)?

The app will be used in meetings that occur on campus. Most of these meeting rooms will have projectors that display the slides for the meeting, but they might not be available in all rooms. The students who are present at the meeting will likely have either a phone or a laptop to use. In the case that a projector is not available, a QR code could be displayed on a laptop screen, or a link could be shared to an organization

group chat. Since meeting time is short and valuable, our solution should integrate seamlessly with the current practices of the organization and take less time than existing solutions do to take attendance.

E. What is the underlying theory or background that needs to be understood in order to address this problem?

To address this problem, a background in app development and/or UI/UX would be helpful for quickly and effectively building an app that is both secure and usable.

F. What prior work has been done on this problem?

There are several companies that have made attendance tracking software. Many of these only have paid options and offer features that can make the program seem complicated.

G. What products, currently available, were not designed or intended for this particular application but could be used to perform a similar function?

Other products that could be used to perform a similar functionality of our technology would be employee timesheet software, survey softwares, Google Forms, any type of spreadsheet software, VOLink, and clickers. However, the majority of the attendance tracker softwares made by other companies are more focused on employees and employers rather than student organizations and instructors.

II. REQUIREMENTS SPECIFICATION

The final product of our team’s development will be an attendance tracking app that works for both web and mobile devices. The primary user base will be student organizations, so we will target our initial efforts for them directly. However, our product should also be capable of supporting other use cases, such as professors and students in classrooms and other events where attendance is tracked.

Keeping in mind these varied user segments, our customer requirements are as follows, ranked from most important to least important:

Functionality	The application is able to successfully track attendance for participants at events. The event host can create an event, and participants may scan into the event with a generated QR code. The host receives a record that the participant is present, and the participant receives a success notification.
Security	The application must be as secure as possible in regards to confidentiality, integrity, and availability. Credentials such as usernames and passwords must be protected. The use of this product should not open any users up to security vulnerabilities, either on their devices or with their personal information.
Ease of Use	The users must be able to intuitively and easily perform the actions needed to utilize the app in a meeting. Since time is of the essence in group gatherings, this solution must be quicker and easier than existing systems for taking attendance (i.e. calling roll). The setup process should be intuitive and quick, and the interface should be clear and uncluttered.
Aesthetics	The app should be designed in an appealing and modern way.

TABLE 2
CUSTOMER REQUIREMENTS FOR THE ATTENDANCE TRACKING APP,
LISTED FROM MOST IMPORTANT TO LEAST IMPORTANT.

These customer requirements come from our own experiences as members and leaders in student organizations, as well as general knowledge of app usability and best practices. As we create prototypes in advance of our final version, we will observe our app in use by selected customers in order to determine if we are meeting these requirements in an effective way.

In addition to the customer requirements, we have generated a list of Engineering Characteristics, based off of these requirements, that will allow us to measure our progress and ensure that it aligns with the needs of the customer. Each of these characteristics is either a constraint, that we cannot change, or a variable that we can experiment with. Below in Table 2 is a list of our characteristics.

Security	Number of security vulnerabilities	Constraint: This number must remain at zero, or as close to zero as possible.
Usability	Number of actions required to perform desired tasks (including misclicks)	Variable: We will measure this characteristic when observing users with the app.
Maintainability	General ease / difficulty of adding new components or features to the application.	Constraint: Must be stable and upgradable whenever new features are wanted.
Adaptability	Number of devices able to run the application.	Variable: Must be flexible to as many different platforms and hardware options as possible.
Aesthetics	User feedback responses.	Variable: We will survey our users on their opinion of the interface, and seek to maximize their opinion of it.

TABLE 3
ENGINEERING CHARACTERISTICS

III. TECHNICAL APPROACH

In order to create an application that conforms to all of our stated criteria, the team will need to have a specified technical approach. This approach consists of what tools we use, what practices we employ, and what processes we follow during development.

The first choice that our team must make for our technical approach is what development platform we build our solution

on top of. As described in the Design Concepts, Evaluation & Selection section below, our team has decided to use the Flutter framework as it best suits our business needs, as well as aligning with our team's previous experience.

While building a Flutter application, our team will follow best practices as defined by Google in their Effective Dart guide [2]. This will ensure that our code is free from language-based vulnerabilities and is easily refactorable and maintainable, helping us achieve our desired Engineering Characteristic measurements. In addition, we will follow Flutter's defined best practices for performance [3] to ensure that our app is not bloated and will perform at acceptable levels on a wide range of devices.

Developing an app with a team of our size requires coordination and a defined strategy. Therefore, we will follow proven development processes in order to organize our efforts and respond to feedback. Our team is adopting an Agile development philosophy, as defined by the famous Manifesto for Agile Software Development [4]. Our team plans to utilize GitHub's Agile Board features to create tickets for each feature or implementation to be developed, which will then be divided up into defined "sprints" for development tracking. By breaking up our development into pieces, we have the opportunity to test out our ideas with customers and change our priorities if required. This will help us to serve the needs of the customer and remain flexible in our work.

As a part of this Agile process, our team will participate in a code review process, where all submitted code is vetted and learned from by other members of the team. This will help ensure our app is secure, as potential vulnerabilities are caught and corrected before even being merged into the codebase. Further, it will allow our team to learn from each other and develop a shared code style for the project through discussion. This process will help ensure the security and maintainability of our application.

By following this technical approach, our team can ensure that development efforts are well-directed and as productive as possible. We can write maintainable and secure code, iterate quickly, and change direction if needed. All of these benefits will directly serve our customers as outlined in the Customer Requirements and Engineering Characteristics.

IV. DESIGN CONCEPTS, EVALUATION AND SELECTION

There are many ways we can create an attendance tracking application. While we hope to expand our application and port it to as many platforms as possible, we have to start with one for our project. Once our first platform was decided, we needed to pick a language to develop in. Since our application must communicate and store shared information, we also needed to find a backend that integrates with all of our tools and software. For each decision, we have a description of each option as well as their strengths and weaknesses. We also have a decision matrix with ratings for several categories. The higher the rating, the better that option is to the user or us.

A. Platform

Since the customers will be using this application on their device, it is important for us to decide which type to target first. The first thought that comes to mind may be a website where people can sign into their accounts and check into events. This has the benefit of being extremely accessible as it doesn't matter what kind of device you have, as long as it can access the internet. There is no app to install so new people will be able to jump on board very quickly. As far as experience goes, not many of us are experienced in developing web applications so this option may be harder to release under a time constraint like this project. We will also address this platform a bit in the frameworks decision section.

Our next possible platform is a desktop application. Several of us have experience with desktop app development in Electron.JS from our COSC 340 class. A severe downside of this platform is the lack of portability and accessibility. Most of the time, people will not be bringing their laptops with them and if they do, it might be in a powered-off state. The installation of an app like this would also be the most intensive of the three on a desktop platform. Users would have to visit a website, download the installer, then install the application.

Our third option would be a mobile application. Like the desktop platform, we have members of our group with mobile app development experience gained from COSC 340. Since virtually everyone has a smartphone on them, a mobile app will have great accessibility. While there is something to install on the device, the process is not as burdensome as the desktop platform.

	Ease of installation	Availability	Easy access	Developer experience
Web	3	3	2	1
Desktop	1	1	2	2
Mobile	2	3	3	2

TABLE 4
PLATFORM BENEFITS MATRIX

Our final decision was to go with a mobile-first approach. We feel it has the best features to start off with and we have the necessary experience to implement it. While the other platforms can fill other niches, we wanted to pick one that has the best availability. Once the app is sufficiently fleshed out, we can start porting it to the web or make a desktop implementation.

B. Framework

For our group's current skill set, most of us have experience with Javascript and Dart from previous projects. Because the skill set of our group is specialized towards these two languages, we ended up narrowing down our options to React Native and Flutter as the frameworks for our app. Although React Native can be brought to the desktop thanks to libraries like Proton Native or support from Microsoft, if we wanted to bring it to the web in the future it would require building a

separate ReactJS project as React Native was built for mobile. While this is not necessarily too difficult to accomplish, it is an extra step that could be better accomplished in an alternative framework like Flutter that will have official native support for the web.

Aside from everyone in this group already being familiar with Dart and developing with the Flutter SDK, we can also use this framework to bring our app to the web and desktop in the future if we choose to. Another positive of Flutter over React Native is that Flutter has native support for Material UI design guidelines. This will allow us to more efficiently design a clean and efficient UI/UX for our app instead of having to install another library to a React Native project. For the reasons of familiarity, native Material UI support, cross-platform compatibility, and future platform development, we have decided to go with the Flutter SDK for the framework for our app.

	Database	Pricing	Time and Labor	Official Flutter Support
AWS	3	2	2	1
Firebase	1	3	3	3
Custom	3	1	1	1

TABLE 5
FRAMEWORK BENEFITS MATRIX

C. Backend

Along with the platform and framework for our project's app, we also needed to choose a suitable backend for authentication, hosting, storage, etc. The options we considered for this project were AWS, Google Firebase, and possibly a custom backend written in Django, Flask, etc. As we are developing our app for this project, a custom backend seemed unnecessary as we would have to handle hosting the backend as well as handle any scalability issues if the user base of the app grew in the future. Therefore our options for a backend were down to AWS and Firebase. The deciding factors between AWS and Firebase came down to the following options: database requirements, complexity, free services vs. paid services, and support for the Flutter SDK.

After comparing both AWS and Firebase and determining which backend is more suitable for our app, we decided to go with Firebase for our backend. Even though Firebase only offers NoSQL databases while AWS allows for a choice of which database to use, Firebase seemed more appropriate for our app's needs as we don't expect our database requirements to grow too complex. As we need a backend that's relatively easy to set up and communicate with, we found that with Firebase it was easier to both setup and interact with callable functions than in AWS. At the moment, Firebase provides more services for free such as user authentication than AWS does. Firebase also looks to be easier to set up than AWS as even though it may offer fewer services out of the box, it is enough for our app at this stage. Flutter even has official support for the Flutter SDK whereas AWS does not. Due to

these reasons, we will be using Firebase as the backend for our Flutter-based mobile app.

	Range of Platforms (Web, Mobile, Desktop, etc.)	Language Experience	Ease of Setup	Material UI Support
Flutter	2	3	3	3
React-Native	2	2	2	2
Ionic	3	1	1	2
Xamarin	2	1	2	2

TABLE 6
BACKEND BENEFITS MATRIX

Now that we have all the major decisions out of the way, we can get started with learning all about these services and software. Since we decided on a mobile platform first, we can create our wireframes and mock-ups to fit the cell phone format. We will still have to decide on the overall layout, but we at least know the limitations of how we can design it. Flutter is our chosen platform due to its simple integration with Android, iPhone, and our backend. Our development experience with the language is also a big plus and we can start brushing up on various concepts to hit the ground running when we start coding the app. Finally, we chose our backend due to its ease of setting up and overall experience. By getting these major decisions out of the way, we hope to have a smooth transition to implementation and these choices give us the best chance of success.

V. PROJECT DELIVERABLES

As a starting point our project has two main deliverables, a phone application and a web application. These products can be seen as our turn-in deliverables, but our true deliverable is to create a tool which provides users with our service through whatever means necessary. Fortunately, we have picked tools which will allow us to reach this objective, and our methods are described in detail within this proposal as well. These deliverables will be applications which possess the functionality that has been previously mentioned in the background sections of this proposal. Our first deliverable will be in the form of an Android mobile application. Developing a web application for our project is built into the timeline; however, in the event of setbacks, this portion of the project will be the first item to be withdrawn from the project requirements, as our main focus will initially be on completing our minimum viable product as a phone application. If all goes well and we are able to meet all of our milestones in a timely manner, we will complete this project with not only a phone application, but a web application that will allow our users to access our service through not only an application on their phone, but through the web. Lastly, our chosen framework allows us to create our mobile application in both an Android and an IOS environment without an entire codebase rewrite, and an IOS version of our application is one more item which we can add to our list of deliverables if time allows.

VI. PROJECT MANAGEMENT

As we are still in the planning phase of the project, we will be making a blueprint for the application in terms of user design and user experience by using Figma (vector graphics editor and prototyping tool). In addition, we will list out milestones of the project that are significant for the minimum viable project as we did for this semester. Once we complete a draft of it, we plan to begin development of a minimum viable product. The estimated start date of developing will be towards the month of January. Our development schedule will be done in a series of sprints with bi-weekly or weekly goals in mind while having one of our members assigned as the project manager. We will be assigning certain roles to each member based on how we section the various parts of the project. There will be a possibility that we will be rotating roles to expand each other's knowledge and improve the team's cooperation and teamwork. This will give each member a chance to experience different aspects of the application development. Due to the fact that we are developing an application, it is most likely that we will not be having many issues with going over the budget. We are using mostly free software to build our project and the cost will be based on maintaining our database and information within the application. However, any budget changes will be documented properly. Currently, the risk that we can predict for this application would be security protection. The most time consuming phase will be the development process. We plan to use the minimum viable product to gauge user feedback and make improvements based on it. Therefore, depending on those processes, it is possible that there will not be enough time to implement a proper security system on the application for database information. However, we will be implementing an university authentication system to protect user information. Overall, the project planning has been going smoothly in terms of our approach and what resources we will be utilizing to make this a successful project. We have a good idea in the direction we are heading, and only need more time to formulate the specific details of the application's features. The team has been completing milestones in documenting our plans and taking initiative for certain roles which has further helped us be more prepared for transitioning into the development phase.

VII. BUDGET

A. Google Firebase - \$100

Although we don't expect for our app to have enough traffic to warrant fees from Firebase initially, we have a budget of \$100 for any possible fees we may encounter should our app's traffic grow large enough.

B. Github - Free

To host our code and enable collaboration between group members, we will be using Github. By using Github, we will be able to work on different parts of the same project at the same time while keeping the repository private if we need to.

C. Flutter - Free

We will be basing our codebase on the Flutter SDK by Google as this will allow us to develop our app both at a low cost in terms of money and time. The Flutter SDK also has native support for Material design guidelines as they're both by Google, allowing us to easily create a desirable UI/UX. With Flutter, we will also be able to easily bring the app to the desktop or web if we decide to go in that direction.

D. Android Studio/Visual Studio Code - Free

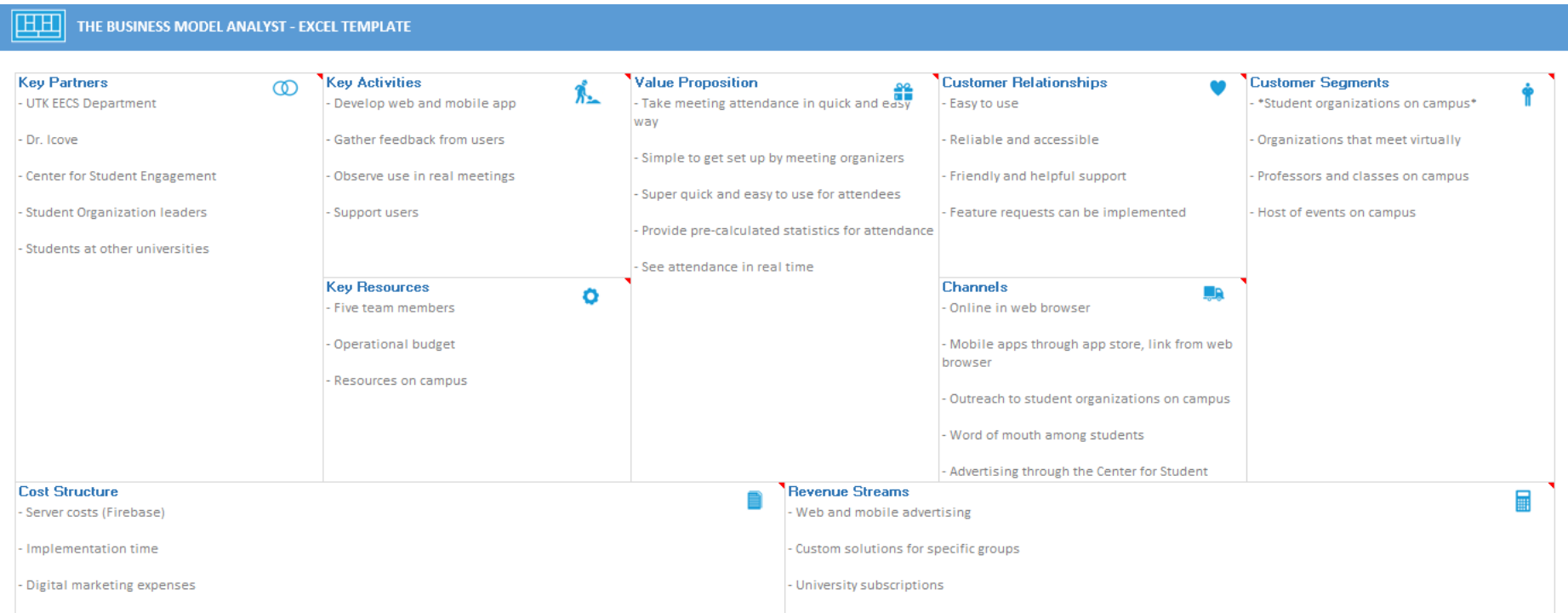
Our group will be using both Android Studio and Visual Studio Code to develop our app. Each developer will be able to choose which IDE they use as long as they can work on the app without affecting the codebase itself.

VIII. REFERENCES

- [1] B. Kopf, "The Power of Figma as a Design Tool," toptal.com.[Online]. Available: <https://www.toptal.com/designers/ui/figma-design-tool>. [Accessed November 20, 2020].
- [2] "Effective Dart," 2020. [Online]. Available: <https://dart.dev/guides/language/effective-dart>. [Accessed: 20-Nov-2020].
- [3] "Performance Best Practices," Flutter, 2020. [Online]. Available: <https://flutter.dev/docs/perf/rendering/best-practices>. [Accessed: 20-Nov-2020].
- [4] K. Beck, Manifesto for Agile Software Development, 2001. [Online]. Available: <https://agilemanifesto.org/>. [Accessed: 20-Nov-2020].

Appendix A: Business Model Canvas

This model helps give our team and project a direction in respect to our target audience and define its purpose.



Appendix B: Milestone Timeline

This visual shows the timeline of tasks we have completed throughout this semester as we plan out certain details of our project.

