Department of Computer Science & Engineering
The University of Texas at Arlington

Project Charter
CSE 4316: Senior Design I
Fall 2022

Passive Aggressive Solutions
Mapbox friend tracker

Nghia Lam

Trieu Nguyen
Colby Wyrick
Gia Dao Duy Duc
Matthew McNatt

Revision History

Revision	Date	Author(s)	Description
0.1	9.26.2022	MM, CW, GD, TN, NL	Document creation
0.2	11.30.2022	MM	Polishing

CONTENTS

List of Figures

1 Problem Statement

There is currently a diverse set of cultures, ideologies, and issues surrounding the time of arrival to events of both formal and informal nature. With the development of public roads unable to keep up with the rapid growth of rush hour commuting, and the presence of multiple global standards for what is and is not acceptably late concerning both time and cause, thousands of hours of time is wasted waiting on people that are late to events or setting up analog systems of attendance that often needlessly single out the unlucky commuter.

2 Methodology

We are going to build a cross platform application that allows groups to track the location of attendants heading to any event, record the timing habits of the attendees over multiple events, and keep all attendants informed in real time as to the status of other group members in travelling to an event.

3 Value Proposition

By creating a robust UI and adaptable back end of this application for real time updates that links to a road navigation-based API, two entirely different and equally profitable applications can be built. One for recreational use marketed as a pseudo social-media platform that can generate revenue with advertising or location information, and another subscription-based business application that allows employers to effortlessly track arrival time of individuals. Since the back end will make up most of the development phase, the only component that needs to be developed separately is the user interface.

4 Development Milestones

Provide a list of milestones and completion dates in the following format:

- Project Charter first draft - September 26, 2022
- System Requirements Specification - October 17, 2022
- Architectural Design Specification - November 7, 2022
- Demonstration of UI Skeleton - December 02, 2022
- Detailed Design Specification - February 2023
- Demonstration of Version 1 of UI - Month Year
- Demonstration of Database Functionality - Month Year
- CoE Innovation Day poster presentation - April 2023
- Demonstration of testable Application - March 2023
- Demonstration of Critical Components - April 2023
- Demonstration of Extra Features - April 2023
- Final Project Demonstration - April, 2023

5 Background

While the multitude of cultures for timeliness can prove complicated, especially for individuals trying to navigate more than one simultaneously, there is no detriment to removing assumptions and making expectations explicit where time is concerned. A great many arrival time issues are just miscommunications or cultural disconnects compounded by the presence of multiple standards. An informal meeting with a coworker could have an acceptable arrival window of 15 minutes while a group of friends needing to finalize theatre seating arrangements might need everyone present five minutes early. But while all these standards exist for what arrival time is acceptable, there is no consistent way to check what expectations are in play currently. This creates a myriad of uncomfortable and time costly situations such as particular individuals refusing to pivot standards, timely individuals letting everyone down on account of missing one critical message, or simple schedule shifts creating confusing and passive aggressive "*who dropped the ball?*" conversations. Meanwhile cooperate cultures experience the same issues, though somewhat deafened by the principle that the accepted standard of the individual with the greatest pay scale rules. But even this creates an unnecessarily harsh to navigate environment where employees must chart out their superior's punctuality expectations. Furthermore, with remote work creating a growing distance ideologically between the office and the job this only gets worse. This problem will continue to persist and even worsen as companies continue to hire for remote positions, inciting further cultural disconnects. While the customer initially wanted a solution to the individual and personal use application of this technology, the personnel and cooperate sides of this issue can both be eliminated with the central principles of explicit expectations, flexible interpretation of arrival time and clear communication before events. By creating a robust customize-able platform that takes all the guessing out of time standards, where individuals are, and what time they are arriving, the customer will have a tool to optimize their schedule and cut down on time loss due to miscommunication. This will streamline the otherwise tiresome process of explicitly scheduling meetings when there is too little time to allow for miscommunication.

6 Related Work

Currently there are no shortage of digital solutions that attempt to use location tracking to better handle employee attendance or specific utility applications. The presence of smartphones in all areas of life, however, has pushed all other options of digital tracking to the edge of feasibility. RFIDs are a good way to track location and have been presented as an enterprise level approach, but they require a dedicated server and devices to manage [3]. Biometrics while very secure and detached from privacy infringement issues have the same problem [5]. Simultaneously, there are several real issues with mobile application location trackers in individual and enterprise use. Suspicion of location tracking in general is becoming widespread as its presence grows and governments start to investigate these data points as tools for law enforcement [1]. Furthermore, Employee tracking applications are severely damaging to an employeeâs moral and general trust in the company [6]. The faceless nature of an algorithm determining company loyalty and lack of human element can end up doing far more harm than good. Meanwhile applications for individual use rely on explicit opt in models that still generate uneasiness as to when they stop processing location data. This perception worsened after the notorious national security breach caused by the fitness app Strava which exposed a Sensitive military information [2]. With the notable exceptions of applications for GPS guidance and child tracking, this has dictated that apps for individual use that utilize this technology rely on users to specify exactly when and with who to share their information. All of this is in addition to the currently expensive nature of location tracking on device battery which creates an annoying caveat to application usage [4]. For the recreational side of this application, the direct opt-in approach will not work since any individual that is currently late is likely not going to opt in to sharing their location with irritated acquaintances. Additionally, to maintain the social peer to peer

feel of any enterprise application, any app with hard and fast arrival deadlines and no ability to account for delays categorized by parties involved as reasonable or unavoidable is going to generate negative interactions. To avoid this a solution must function almost entirely independent of user interaction while pushing any judgment calls on acceptability back to other users, but both characteristics together are not widely available, especially for individual use. Ultimately a new model for opt in location tracking is necessary to accomplish the requirement of transparent tracking only when necessary that does not rely on case-by-case user approval. This self-aware tracking must also be limited to specific and concise windows so as not to perpetually drain the resources of a mobile device, ruling out any service that relies on constant information tracking.

7 System Overview

How our group is going to implement a solution to this problem is to first chart exactly when the application will track the location of its users. There are two diagrams that will be shown below, the state diagram and the actor diagram. The actor diagram will show the relationship between the users and the app's functionalities. The state diagram chart represents the steps to the two central functionalities of the system, with orange states representing periods where the location of the user is being tracked.

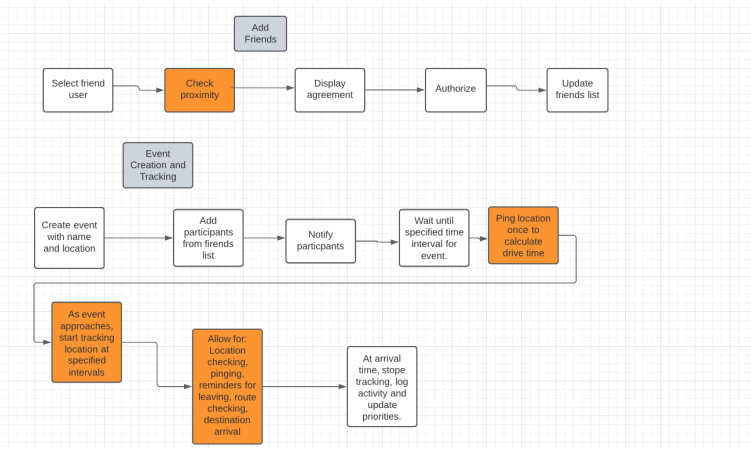

Figure 1: State Diagram

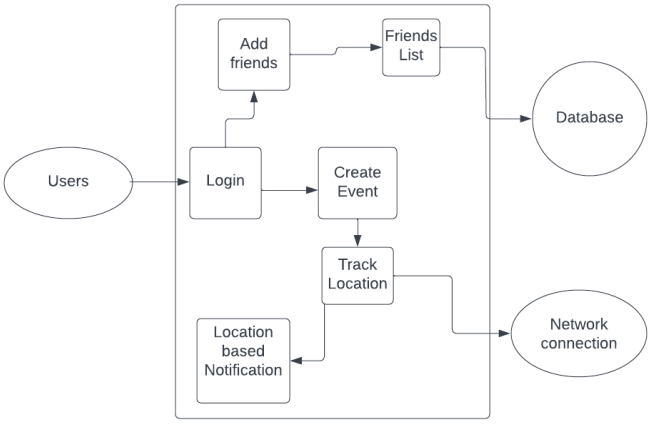

Figure 2: Actor diagram

The first part of this solution is to be able to add friends (or people who want to meet), and be able

to create events at a certain location and time that everyone in your group agrees to meet. Then, a certain amount of time before the event, the app will begin to track the locations of the people who agreed to the event in real time. The app should be able to ping to other people in the group where everyone is at and what distance they are from the event's location. This app should also be able to tell people in the group if somebody slows down sharply (whether because of an accident or large amounts of traffic) through a notification. Lastly, the application needs to be able to stop tracking once a person reaches a certain radius of the event's location. The major components of this app is the ability to track multiple people on a map, notifications based on location and time, a friends list, and event creation based on the location and time that the group wishes to meet.

8 Roles & Responsibilities

Name	Roles	Responsibilities
Trieu	Scrum Master	Burn down chart, Documentation
Nghia	Programmer	Documentation, Programming
Gia	Programmer	Documentation, Programming
Colby	Programmer	Documentation, Programming
Matthew	Programmer	Documentation, Programming

Table 1: Overview of team roles and responsibility

The stakeholders of our projects are Trieu Nguyen, Nghia Lam, Gia Dao Duy, Colby Wyrick, Matthew McNatt, and Shawn Gieser. The Team members, roles, and responsibilities are located in the table above and Shawn Gieser is our sponsor.

Roles and Responsibility:

Trieu will be managing the time of our group and documentation. Since Trieu is our Scrum Master he will be the first point of contact with our sponsors and customers. He will be the Scrum Master throughout this project year and this will not change periodically.

Nghia, Gia, Colby, and Matthew will all be programmers in this project. We will separate the program's responsibility into multiple sections and associate it with each team member. Our entire team will do documentation together.

9 Cost Proposal

Since the project is focusing on tracking location, the implementation phase requires heavily on the use of maps API, including navigation, routing, movement data,...Currently, we are considering using MapBox API, which is a free, open-source for online maps; however, if any bugs happen or the framework we chose to implement the application cannot utilize MapBox, then we could have to use our budget for Google Services.

9.1 Preliminary Budget

Currently, the project does not require preliminary budget.

9.2 Current & Pending Support

The funding source of the project is from our sponsor with the amount of 800 dollars. During the project's implementation and testing, we do not spend more than this amount.

10 Facilities & Equipment

Online meetings via Teams and Discord will be the group's main method of communication. For in-person group meetings, the lab space will be ERB 129 (Engineering Research Building - room 129), and for future meetings with our sponsor to discuss project modification, it will be in ERB 209. The Library is an alternative option for the group's meetings if ERB rooms are already occupied. For our equipment, we would need a laptop and a stable internet connection. For our application to run on multiple platforms, we need both IOS and Android mobile devices. Currently, the project does not require purchase or lease of any electronic equipment; however, we can contact our sponsor for the necessary equipment.

11 Assumptions

The following list contains critical assumptions related to the implementation and testing of the project.

- A meeting on teams or in person every week
- Our workspace in person is available
- Map box will be available
- All our documentation for the application will be finished by December
- All team members will have access to the internet and all developer applications
- No team member will drop out of the project at any phase

12 Constraints

The following list contains key constraints related to the implementation and testing of the project.

- Final prototype demonstration must be complete by May 1 2022
- Security will be questionable because of lack of knowledge
- Total development cost must not exceed 800 dollars
- Team has other classes, thus development will take longer
- Application will either need a stable internet or data source
- Will either have an android or apple device

13 Risks

The following high-level risk census contains identified project risks with the highest exposure. Mitigation strategies will be discussed in future planning sessions.

Risk description	Probability	Loss (days)	Exposure (days)
Getting sued because of tracking people's locations	0.50	50	25
Bots taking over app	0.30	14	4.2
Being hacked because of security issues	0.50	9	4.5
Unable to test because of weather issues	0.30	20	2.0
Information being leaked due to lack of security	0.5	10	5

Table 2: Overview of highest exposure project risks

14 Documentation & Reporting

14.1 Major Documentation Deliverables

14.1.1 Project Charter

This document will be maintained by updating the contents with any new content that comes up at least once every sprint. The initial version of the project charter will be delivered on September 26, 2022. The final version will be submitted in May 2023 in the final sprint.

14.1.2 System Requirements Specification

This document will be maintained by updating the system requirements anytime any major changes happen to the application in a given sprint. The initial version of the system requirements will be delivered on October 17, 2022. The final version will be submitted in May 2023 in the final sprint.

14.1.3 Architectural Design Specification

This document will not be updated unless something major needs to be changed. The initial version of the system requirements will be delivered on November 7th, 2022. The final version will be submitted in May 2023 in the final sprint.

14.1.4 Detailed Design Specification

This document will only be updated if something major changes in the application. The initial version date will be revealed at a later date. The final version will be submitted in May 2023 in the final sprint.

14.2 Recurring Sprint Items

The project charter and the system requirements will be updated every sprint.

14.2.1 Product Backlog

After team members are assigned with their tasks, they are to follow the system overview to implement what is required. Then the items will be added to the product backlog using programmers' reports or teams discussion, including what functionalities they had finished implementing and testing, and what functionalities are remain. Since the product backlog will be updated regularly, the prioritized items depend on what values should the project deliver each sprint. However, the project focuses on location tracking and users connection, therefore, these basic functions should be the priorities to be maintained and revised throughout the implementation. As prioritized items can be changed during implementing phase, weekly reports and burndown chart are essential tools to keep track of the project's progress. If any item needs to be delivered first or consumes more time than anticipated, the group will adjust the schedule accordingly and work on it to make sure the item is completed on time. In addition, a public repository is shared for all members with an attached README file, then all members can take note of what functions or items should be prioritized. The decision of items prioritizing will be made once

there is an agreement among all group members. Github, Overleaf, Excel are main software tools that will be used to maintain and share the product backlog with team members and stakeholders.

14.2.2 Sprint Planning

Each sprint will be planned through group meetings every week on Friday. There will be 8 sprints in total, 4 sprints for Senior Design I and 4 sprints for Senior Design II.

14.2.3 Sprint Goal

Everyone in the group will come together to come up with the sprint goals and we will involve the customer by meeting with them as needed.

14.2.4 Sprint Backlog

The team will decide together on the most important feature for the sprint backlog.

14.2.5 Task Breakdown

The individual tasks will be assigned based on the voluntary of team members. The group will be utilizing pair programming so that two members can work on one function to support each other and document the progress. The time spent on each tasks will be reported individually by all team members and submitted to the project manager to be summarized in the burndown chart.

14.2.6 Sprint Burn Down Charts

Trieu Nguyen will be responsible for generating the burndown charts for each sprint. We have a discord where we post the amount of time we worked on the project and what section we work on. Then Trieu will take the time and input it into the excel file where it will then create a burndown chart.

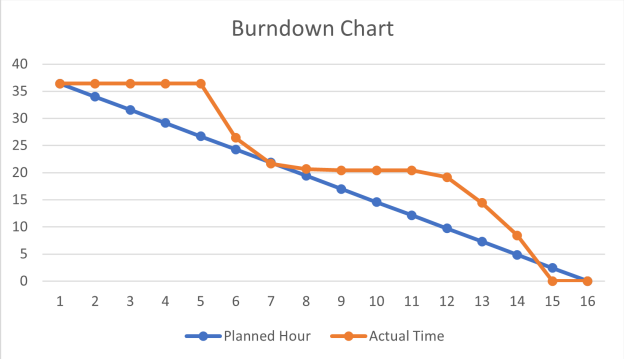

Figure 3: Example sprint burn down chart

14.2.7 Sprint Retrospective

We meet together every week to talk about what we've done that week and what we still need to complete before the end of the sprint or next meeting. It will be documented as a group and it will be due each week.

14.2.8 Individual Status Reports

There will be a report every Friday at 1 pm. We will either meet on Teams or at one of the in-person locations. Once a member has completed or has worked on a sprint backlog item they will post it on discord with the time it took the to do it. The key items that will be contained in each report are the progress and completion of anything they have done.

14.2.9 Engineering Notebooks

The engineering notebook will be updated by each team member every sprint and every time we meet up with the sponsor. We donât have a minimum number of pages required for each member and how long the interval will be. Each member will be accountable for the own work they put in. There will be no witness for ENB pages.

14.3 Closeout Materials

14.3.1 System Prototype

The final system prototype will include a working application with all the main features attached like event creation, friends list, and GPS tracking. Other features might be included if there is time before the deadline. This will be demonstrated through a presentation and a demo video. We will be doing both a PAT and FAT.

14.3.2 Project Poster

The project poster will include all the necessary information about the features on the app, how it was created, and what the app is used for. The poster size will be 36 inches tall and 48 inches wide. This will be delivered towards the end of the project when we are done with the project and it is ready to be shown to the public.

14.3.3 Web Page

On our project web page we will be including features, demo, download link, contact information, and tutorial. All of this information on the web page will be accessible to the public. The web page is going to be at the very end of the project once we get all the features and app running as intended. So that means the web page will be a closeout to the project.

14.3.4 Demo Video

The video demo would contain the application and people using it. As of this moment, the demo will not contain any B-reel footage. The video will be around 1 to 2 minutes long and the video will cover the main features of the application and how it works.

14.3.5 Source Code

At this current moment, our source will be maintained through GitHub. Source code will be provided to our customers but not to the general public. This project cannot be open source because of security risks. For the license terms we will be using MIT and it will be listed in every read-me file

14.3.6 Source Code Documentation

The documentation standard will be employed in JavaScript as comments. And then we would be able to use Java-docs in order to convert all the comments into documentation. Then it would be finalized as a PDF file.

14.3.7 Installation Scripts

The software will be kept up to date by an installation script, which will be provided by google play or the app store. If there is a new version of the app, we would send a notification to all users noticing that there is a new version of the app they need to download before accessing it.

14.3.8 User Manual

There will not be a user manual or setup video for the app. We will however provide a brief walk-through on how to use the app once the user first creates an account and logs in. The tutorial will go

over all the features of the app, such as creating an event, inviting people to an event, connecting with others, etc. There will also be a way for the user to access the tutorial again when needed.

REFERENCES

[1] Mike Fong. he unforeseen risks of sharing smartphone location data. *Security Magazine*, 2019.

[2] Marco Niemann Jan H. Betzing and C. Ingo Berendes. Privacy-friendly user location tracking with smart devices: The beat prototype. page 1, 2019.

[3] A. Kassem, M. Hamad, Z. Chalhoub, and S. El Dahdaah. An rfid attendance and monitoring system for university applications. In *2010 17th IEEE International Conference on Electronics, Circuits and Systems*, pages 851–854, 2010.

[4] Mikkel Baun Kjærgaard, Sourav Bhattacharya, Henrik Blunck, and Petteri Nurmi. Energy-efficient trajectory tracking for mobile devices. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, MobiSys '11, page 307â320, New York, NY, USA, 2011. Association for Computing Machinery.

[5] M. Olagunju, A. E. Adeniyi, and Oladele. Staff attendance monitoring system using fingerprint biometrics. international journal of computer applications. pages 8–15, 2018.

[6] Sandeep Trivedi and Nikhil Patel. Virtual employee monitoring: A review on tools, opportunities, challenges, and decision factors. *Empirical Quests for Management Essences*, 1(1):86â99, Dec. 2021.