Tech Spec - SafeRide

Team: #0000FF Thunder
Leon Chen
Frankie Htet
Andy Lee
An Ta
Orion Tang (Team Leader)

Date Issued:
October 6th, 2021

Introduction/Summary

At its core, SafeRide is a web application designed to run on mainly mobile devices where it can be utilized to its fullest capabilities, with real time GPS tracking. SafeRide includes features that require real time updates, where asynchronous updates are required. For example, map and route updates, where when the user moves around, their GPS location updates, will be sent to the server continuously and depending on their GPS location, the server will send back updates to specific views that updates the visual that represents where the user is.

Software Used

Communication

- Discord (latest version)
    - Primary way of communication between teammates, using both text and voice chat. Allows the team to schedule meetings and uses it for sprint and daily standups.

- Zoom (v5.7.7)
    - Zoom is the primary way of communication between teammates and the client, which provides the specs and various other information about the requirements

- Slack (web and desktop)
    - Slack is a backup form of communication used between teammates and the client which is used if other forms of communication is unavailable

Collaborative tools and VCS

- Jira
    - Jira is the preferred way of tracking tickets, issues, and tasks for the team. It is also used to start and track sprint progress, as well as various statistics and information about our group, such as burndown chart, and velocity charts that can be used to inform our team of how on target we are for our sprints.

- Github
    - Version control software used by our team to track differences and progress of our requirements, specifically for milestone 1 currently. We also use this to track and build our app.

- Google docs
    - Collaborative tool for teammates to work on the same document. Has real time updates to track edits done by teammates and can work on same document in real time

- Google drive
    - Collaborative file sharing tool, used to keep all of the documents together, can include many types of forms such as slides, docs and

excel documents. Used in conjunction with github because it is easier to access and it is real time.

- Lucidcharts
    - Collaborative whiteboard to build sitemap and wireframing to design desktop ui for site. Online diagramming tool that helps build appealing UI to allow for more accurate depictions of realized application

- Figma
    - Collaborative whiteboard to build mobile wireframe for site. Alternative online diagramming tool for building UI to allow for accurate depictions.

IDEs

- Visual Studio Code (for windows latest version (1.60.1))
    - Advanced text editor that supports many languages and linting via addons. Used to build apps of various sizes and scope.

- Jetbrains WebStorm (latest version (2021.3 eap build 213.4293.8))
    - IDE that specializes in building javascript and related applications that includes advanced features that makes developing easier and more convenient such as ml based autocomplete addons and live editing. Free to use when applying with school email.

- Jetbrains Pycharm (2021.2.2)
    - IDE that specializes in building python applications, used in our project to sort and clean data for possible machine learning models for our safety ratings. Linting and other features speed up development time and makes it more efficient.

Other frameworks and technologies

- Tensorflow (2.x.x)
    - Used to build and satisfy features that include deployment of machine learning models. Is useful since they provide many high level capabilities and functions that we can use to start building practically.
    - Alternatives
        - Pytorch
            - Lower level framework for machine learning model training. Can be used instead of tensorflow to allow for more modular and lower level manipulation of possible models. Typically more advanced and a steeper learning curve.

- Docker (for windows latest 4.1.0)
    - Used for application microservice architecture. Docker makes it easy to set up our microservice backend. Can be used to containerize database services, as well as make deployment and testing on different hardware easy.

- Google Colab or Kaggle
    - Used to train and manage datasets and models. Free plans include GPUs and TPUs that could be used to train. Links to google drive to convenient save and train models.

- MySQL Workbench (latest MySQL 8.0.26)
    - Essential in developing and troubleshooting SQL databases that will be used in our application.

- Postman
    - Allows testing of application apis. Convenient way to troubleshooting and developing for example restful apis. Able to collaborate with teammates through postman for easy analysis of application api.

Languages

- Typescript (latest 4.4.3)
    - Alternative to Javascript, could be used in certain cases if the team wants to. Most likely, if we use other libraries or frameworks in our frontend such as angular, typescript could be an option in contrast to javascript.

- Javascript (ECMAScript 2018)
    - First option in developing our application, most have some experience with this. Language will be used most often in our application due to the use of the planned react + node + express frameworks and libraries.
- HTML
    - Used to create the backbone of a website, useful in designing the structure of our web application and where everything goes.

- CSS (CSS3)
    - Used to customize UI. Used to design and make UI cleaner and more user friendly and appealing to look at.

- Python (> 3.7.x)

        - Used to build machine learning models and interact with tensorflow and pytorch libraries. Used in google colab and kaggle to clean and sort data.

- SQL
    - Used to develop and maintain our databases. Used in MySQL workbench in order to interact with our applications database.

- Shell scripting
    - Used if working with docker deployments for microservice architecture of our backend. Also used when deploying on cloud services.

Backend technology

- Node js + express js (LTS 14.18.0, and (4.17.1 or latest) respectively)
    - Preferred choice of backend framework. Popular with extensive documentation. Our main features will rely on SPA architecture since the view will need to be continuously and asynchronously updated with constant GPS change with a lot of different real time changes happening. Node + express + react should be ideal for this kind of application.
    - Alternative techs
        - Java spring for java backend development. Spring is multi-threaded which could be useful for our application where lots of real time updates are made. Spring is also a mature and well documented framework with a lot of support. Java could be a more preferred language to work with.

Frontend technology

- React (latest 17.0.2)
    - Popular and well known UI library to act as our frontend technology. Good ecosystem that is easy to learn and build. Works well with node.js and express that forms the MERN stack if used with MongoDB. Modular with ability to include third party libraries.
    - Alternative frontend framework
        - Angular
            - Good alternative that is more heavy weights and includes features which makes third party libraries less needed. Includes solutions in itself.
            - More straight forward development with previously well known paths to take with angular.

        - Thymeleaf

            - Could be a choice to use as a view if using java spring as a backend framework. Thymeleaf integrates well with java spring and acts as a web engine for our frontend.

Database

- SQL Server (2019)
    - Alternatives depending on whether we need a relational database or not
        - MongoDB
            - Document based database that goes well with react, node and express. Works well with dynamic web interfaces and json based communication between stacks such as with react, node and express.

OS's

- Windows 10
    - Primary operating system used by all of our teammates. Operating systems allow users to efficiently utilize the system's resources.

Hardware

- Personal computer
    - Provides hardware such as storage and CPU to be able to develop software. Provides the resources needed to perform complex tasks

Limitations of technology

- Our application will be a web app and there are currently no plans for mobile application development. Technologies that are needed for mobile development will not be included.
- Data limitations in the early stages will prevent us from making robust machine learning models, where libraries such as Pytorch and Tensorflow and languages such as python will not be utilized.
- Our application will not support offline mode. No technologies that work with offline mode will be supported.

Sources:

https://www.guru99.com/react-vs-angular-key-difference.html
https://en.wikipedia.org/wiki/Thymeleaf
https://medium.com/@devathon_/mean-vs-mern-stack-development-5ba3e517bc68
https://betterprogramming.pub/node-js-vs-spring-boot-which-should-you-choose-2366c2f76587
https://spring.io/microservices
https://www.w3schools.com/js/js_versions.asp
https://www.mongodb.com/mern-stack
https://en.wikipedia.org/wiki/Operating_systemhttps://en.wikipedia.org/wiki/Operating_system