TCA Carpool Overhaul

Team Members

1. Kelby Morrison
2. Jibas Poudel
3. Lucas Faupel
4. Matthew Long
5. Robert Mason
6. Marcus Langston
7. Andrew Smith

Background Info

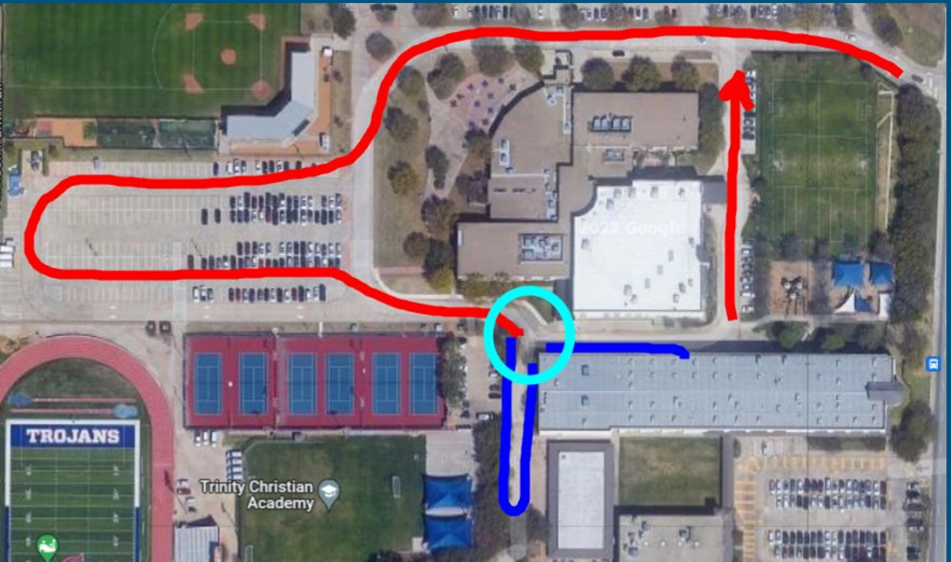
- Trinity Christian Academy
- Uses a numbering system for pickup
- Takes over half a dozen people just to run carpool

Problem

- The main problem is carpool is inefficient
- Secondary problems include
  - traffic
  - related costs for traffic
  - teacher safety while performing carpool duties

Objective

- Make carpool more efficient



Key concepts And Related work

- Cross-platform mobile application written in React Native
- Web application written in NextJS
- Firestore Database
- Numbers are assigned to families to pick up their kids
- The application should allow a parent to enter their families pickup number and that number be displayed in the pickup area so the child is released
- There are other apps out there that perform this, however
    - They are not tailored to our specific problem
    - They are too robust for the primary use case

Use Cases

- Primary use case
  - Let a parent pick up all children in a family in a designated pickup area
- Secondary use cases
  - Let a parent pick up multiple families children in a designated pickup area
  - Be used for after school care pick up (late pick up) so the parent doesn't have to walk into the building

Deliverable: Mobile App

Functions:

- Allow parent/guardian to check in
- Send notification to web front end for student to be picked up
- Timeout notification/allow parent to check out once they have their student(s)

Deliverable: Web Interface

Functions:

- Display which pickup location a student needs to go to
    - Needs to be updated when parents arrive and check in through mobile app
- Remove students names/carpool numbers as parents leave

Deliverable: Backend/Database

Functions/Specifications:

- Store parent login data, pickup location, and associated student carpool number(s)

Tasks

1. Configure Firestore Database
    a. Create specific tables needed for storing critical information
    b. Implement server authentication
2. Begin development of mobile app using React Native
    a. Design a user friendly mobile app UI/UX
    b. Configure mobile app logic
3. Design website Front End
    a. Ensure website is easily read and understood from a projected screen
4. Test Server Authentication
5. Final test demo