DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON

DETAILED DESIGN SPECIFICATION
CSE 4317: SENIOR DESIGN II
SUMMER 2021

TEAM CODEBENDERS
SCHOOL PICKUP AND DROP OFF APP

AVINASH ARAYAL
MANIZHA NOORI
NAVEEN KSHETRI
PRABHAT BACHAGAIN
SANTOSH BHANDARI
UJJWAL BAJAGAIN

Revision History

Revision	Date	Author(s)	Description
0.1	06.23.2021	SB	document creation
0.2	06.25.2021	AA, NK, MN, UB, SB, PB	editing and correction
0.3	06.28.2021	AA, NK, MN, UB, SB, PB	proofreading and finalization

CONTENTS

List of Figures

List of Tables

1 INTRODUCTION

During the school pick-up time, the traffic around the school becomes congested and sometimes would even block the whole neighborhood. It dramatically increases the safety risk of pedestrians and vehicles. At the time of COVID-19, students cannot stay in a group at a designated area and should be dismissed one by one when their parents come to pick them up. It requires much more coordination among teachers. Also, teachers should be on a constant lookout to check if students are on the school premises. We hope to automated the process by knowing whose student's parent has entered the premises and dismissed them in a designated picking spot. It will reduce the overall pickup time and decrease the safety risk of pedestrians and vehicles.

2 SYSTEM OVERVIEW

The application will be separated into 3 layer: Mobile Application layer, API Layer, and Backend Layer. The Mobile Application layer will be responsible for providing the interactive user interface. The API layer will be responsible for communicating between mobile and Backend layer. The backend layer will be used for business logic, database operations and will directly communicate with Website UI using HTTP requests .

The Mobile Application layer for the phone will support the android and the IOS. The mobile UI is for parents who will be picking up or dropping of their children from school. This layer interface with users and allow them to share data and location with the school system. The mobile application will be using geofencing to validate if the users is within the allocated radius from the school. The API layer is the bridge between the Mobile, 3rd party API layer and Backend layer. API layers is responsible for passing HTTP and JSON requests between the different layers.Backend layer will be responsible to adding and store information in the database, make database calls, perform various logical task such as user validation and authentication. Backend Layer is responsible for business logic and creating, updating, reading and deleting information from database. Here is the diagram that shows how data flow between different subsystem.

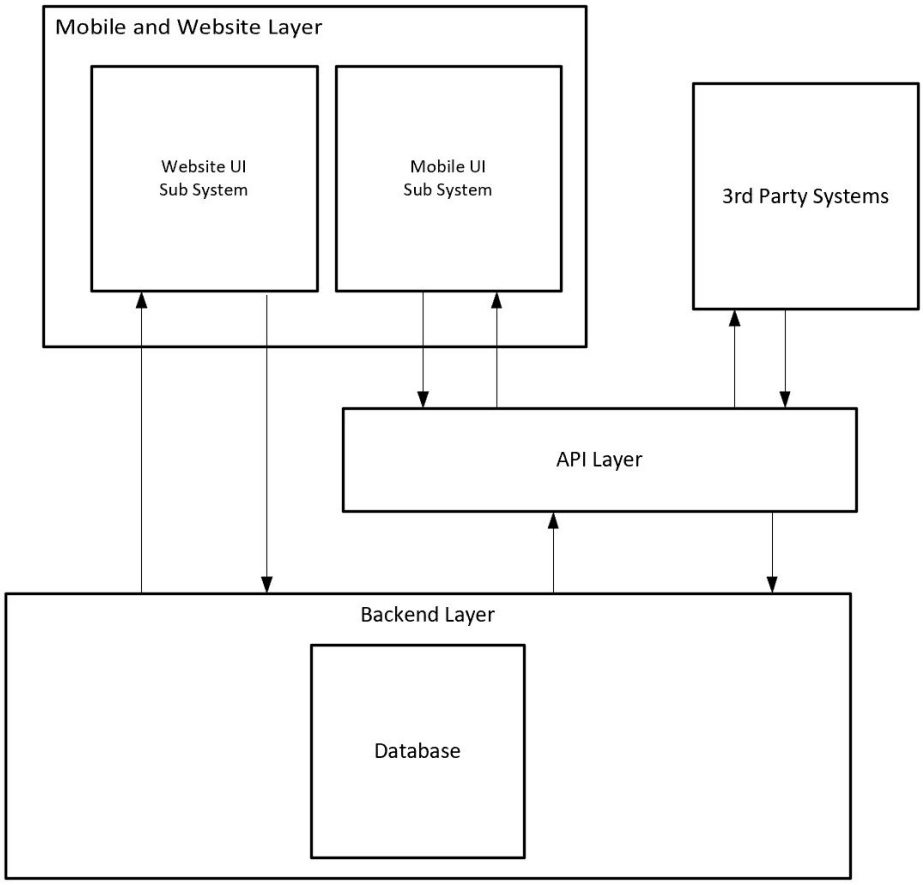

Figure 1: A simple data flow diagram

3 API Layer Subsystems

This layer is the bridge between the Mobile, 3rd party API layer and Backend layer. API layers is responsible for passing HTTP and JSON requests between the different layers.

3.1 Layer Hardware

Mobile devices and Heroku Infrastructures (cloud Infrastructures)

3.2 Layer Operating System

The Cloud server for this application is running Ubuntu image curatedand maintained by Heroku and the local development environment is running on Windows 10. iOS and Android OS for Mobile devices.

3.3 Layer Software Dependencies

Django Rest Framework,Node.js, React.

3.4 Authentication

This API service is responsible for authenticating users(log in) and authenticating their request.

3.4.1 Subsystem Hardware

Mobile devices and Heroku Infrastructures (cloud Infrastructures)

3.4.2 Subsystem Operating System

Ubuntu in Heroku. iOS and Android in mobile devices.

3.4.3 Subsystem Software Dependencies

Django Framework and React

3.4.4 Subsystem Programming Languages

Python and JavaScript

3.4.5 Subsystem Data Structures

The information are passed as JSON via Htpp post request from mobile devices. The data included are email and password. In return Backend will return token if the authentication are valid and error when token is invalid.

3.4.6 Subsystem Data Processing

Django authentication system is responsible for phrasing the request, hashing the password and generating the token.

3.5 Location

This API service is responsible for saving the parents location send from mobile to the database.

3.5.1 Subsystem Hardware

Mobile devices and Heroku Infrastructures (cloud Infrastructures)

3.5.2 Subsystem Operating System

Ubuntu in Heroku. iOS and Android in mobile devices.

3.5.3 Subsystem Software Dependencies

Django Framework and React

3.5.4 Subsystem Programming Languages

Python and JavaScript

3.5.5 Subsystem Data Structures

The information are passed as JSON via Htpp post request from mobile devices. The data included are latitude, longitude and timestamp.

3.5.6 Subsystem Data Processing

Djanjo serializer class is responsible for saving the information.

3.6 GetNearby Parents

This API service is responsible send nearby parents when staff send the get request.

3.6.1 Subsystem Hardware

Mobile devices and Heroku Infrastructures (cloud Infrastructures)

3.6.2 Subsystem Operating System

Ubuntu in Heroku. iOS and Android in mobile devices.

3.6.3 Subsystem Software Dependencies

Django Framework and React

3.6.4 Subsystem Programming Languages

Python and JavaScript

3.6.5 Subsystem Data Structures

The get request is send from Mobile devices. In response backend will send list of nearby parents who are there for pickup.

3.6.6 Subsystem Data Processing

Djanjo serializer class is responsible for saving the information.

3.7 Notification

This API service is responsible for sending notification.

3.7.1 Subsystem Hardware

Mobile devices and Heroku Infrastructures (cloud Infrastructures)

3.7.2 Subsystem Operating System

Ubuntu in Heroku. iOS and Androud in mobile devices.

3.7.3 Subsystem Software Dependencies

Django Framework and React

3.7.4 Subsystem Programming Languages

Python and JavaScript

3.7.5 Subsystem Data Structures

JSON response with pickup spot.

3.7.6 Subsystem Data Processing

N/A

3.8 Profile

This API service is responsible for sending information about user profile.

3.8.1 Subsystem Hardware

Mobile devices and Heroku Infrastructures (cloud Infrastructures)

3.8.2 Subsystem Operating System

Ubuntu in Heroku. iOS and Android in mobile devices.

3.8.3 Subsystem Software Dependencies

Django Framework and React

3.8.4 Subsystem Programming Languages

Python and JavaScript

3.8.5 Subsystem Data Structures

JSON response with profile deatials.

3.8.6 Subsystem Data Processing

N/A

4 MOBILE APPLICATION LAYER SUBSYSTEMS

The Mobile application layer will allow users to interact with the system and communicate the infraction with the backend using REST API and HTTP requests respectively. The Mobile Application layer consist of sub-layers like Login, Homescreen, Profile and Notification. The backend layer verifies user and connect user to their school system.

4.1 LAYER HARDWARE

This layer required the actual android smart phone or iPhone as the main hardware to have the application to be running and it must have network connection such as WiFi and internet connection.Also should allow application to track location within the geo-fencing range.

4.2 LAYER OPERATING SYSTEM

The layer will require Android Operating System or iOS

4.3 LAYER SOFTWARE DEPENDENCIES

This layer requires Android version or latest and iOS version

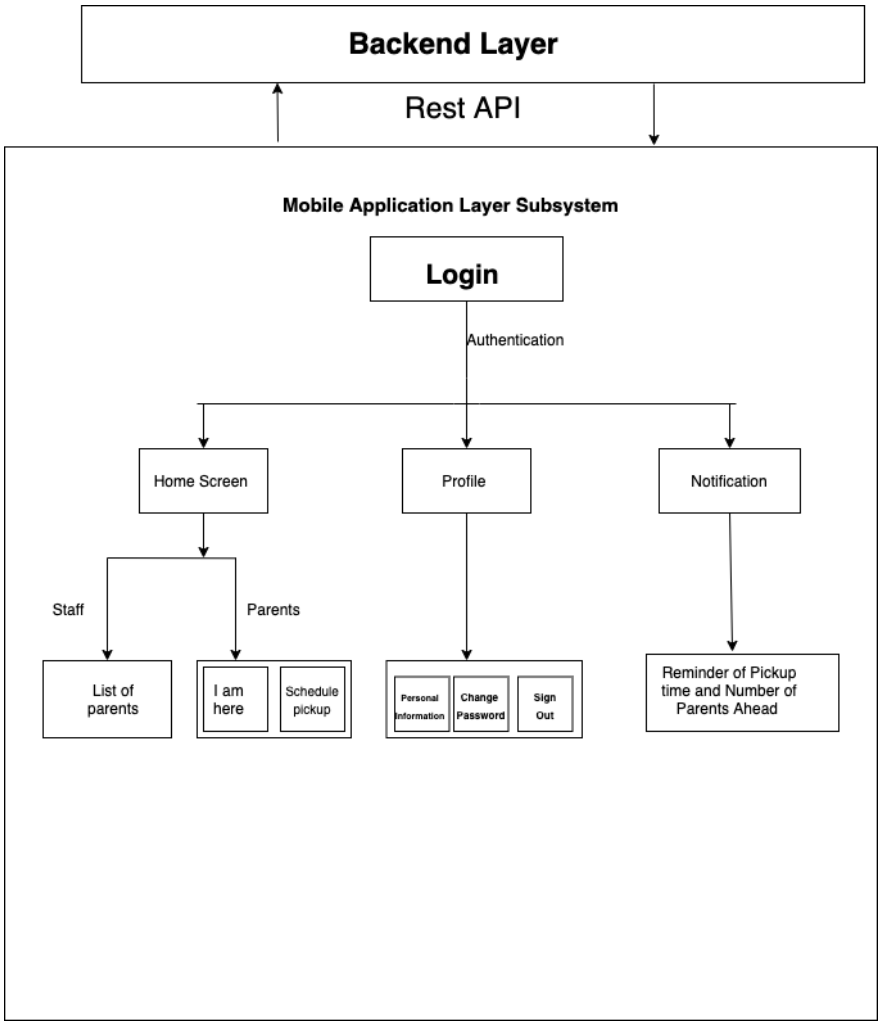

Figure 2: mobile application layer subsystem description diagram

4.4 LOGIN

The purpose of the sub-system is to allow user to get inside the system to access the services that are provided in our application. The parent will be able to logged in into the system using the Username and password given by the administration. The parent must enter the valid username and password otherwise the access will get denied.

4.4.1 SUBSYSTEM HARDWARE

The project is entirely based on software platforms, there are no hardware dependencies.

4.4.2 SUBSYSTEM OPERATING SYSTEM

The application will be running on Android or iOS operating system

4.4.3 SUBSYSTEM SOFTWARE DEPENDENCIES

React.js Libraries is used to create this layer User Interface.

4.4.4 SUBSYSTEM PROGRAMMING LANGUAGES

Java Script will be used on React Native framework to create this subsystem.

4.4.5 SUBSYSTEM DATA STRUCTURES

N/A

4.4.6 SUBSYSTEM DATA PROCESSING

N/A

4.5 HOME

The purpose of the sub-system is to allow staff to see the list of the parents that are near the pickup station. It gives list in the ascending order once staff drop off the children then parents list will go away and second parents will be at 1st position. In parents home screen they will be able to click I am here which they will do when they get near school and they can schedule pickup time for their children.

4.5.1 SUBSYSTEM HARDWARE

The project is entirely based on software platforms, there are no hardware dependencies.

4.5.2 SUBSYSTEM OPERATING SYSTEM

The application will be running on Android or iOS operating system

4.5.3 SUBSYSTEM SOFTWARE DEPENDENCIES

React.js Libraries is used to create this layer User Interface.

4.5.4 SUBSYSTEM PROGRAMMING LANGUAGES

Java Script will be used on React Native framework to create this subsystem.

4.5.5 SUBSYSTEM DATA STRUCTURES

N/A

4.5.6 SUBSYSTEM DATA PROCESSING

N/A

4.6 PROFILE

This subsystem will allow the parent to look up their personal information that they had got registered into the administration.The parent will be able to see their name, contact information and contact email address. Besides these the parent will also be able to change the password to their account and reset a new one. There will be one Sign-out button in case parent wants to logged out from the application.

4.6.1 SUBSYSTEM HARDWARE

The project is entirely based on software platforms, there are no hardware dependencies.

4.6.2 SUBSYSTEM OPERATING SYSTEM

The application will be running on Android or iOS operating system

4.6.3 SUBSYSTEM SOFTWARE DEPENDENCIES

React.js Libraries is used to create this layer User Interface.

4.6.4 SUBSYSTEM PROGRAMMING LANGUAGES

Java Script will be used on React Native framework to create this subsystem.

4.6.5 SUBSYSTEM DATA STRUCTURES

N/A

4.6.6 SUBSYSTEM DATA PROCESSING

The information that we get from backend first name,last name,email need to be rendered in the User Interface.

4.7 NOTIFICATION

In this subsystem,the parents will get notified about their pickup time and spot and also about the number of parents who are ahead of them waiting on the line.

4.7.1 SUBSYSTEM HARDWARE

The project is entirely based on software platforms, there are no hardware dependencies.

4.7.2 SUBSYSTEM OPERATING SYSTEM

The application will be running on Android or iOS operating system

4.7.3 SUBSYSTEM SOFTWARE DEPENDENCIES

React.js Libraries is used to create this layer User Interface.

4.7.4 SUBSYSTEM PROGRAMMING LANGUAGES

Java Script will be used on React Native framework to create this subsystem.

4.7.5 SUBSYSTEM DATA STRUCTURES

N/A

4.7.6 SUBSYSTEM DATA PROCESSING

N/A

5 Backend Layer Subsystems

The backend layer is responsible for performing business logic such as calculating distance between school and guardians and conducting database operations such as create, read, update and delete.

5.1 Layer Hardware

The backend layer sub system is running on a 3rd Party cloud platform as a service provider, so layer hardware is outside the scope of this project.

5.2 Layer Operating System

The backend layer can be run on Windows, Mac and Linux distribution as long as the operating system supports python 3.8 or higher. The Cloud server for this application is running Ubuntu image curated and maintained by Heroku and the local development environment is running on Windows 10.

5.3 Layer Software Dependencies

The backend layer can be further divided into Database and Web Application. The Database is running postgresql RDMS. The web application which perform most of the logical operations and connects to database is build on top of python as the programming language and using a wide range of dependencies and libraries. The backend web application is build on top of Django; which is a web development framework and utilities Django REST library to build REST APIs. A complete list of dependencies can be found in requirements.txt file which is located in project's GitHub repository.

5.4 Database Subsystem

Database Subsystem is responsible for storing various data related to the workings of this application such as but not limited to user details, schools detail and staff details. The database subsystem stores the data as key value pair.

5.4.1 Subsystem Hardware

Hardware requirement is outside scope of this project.

5.4.2 Subsystem Operating System

The database will be running on Linux, in addition it also supports other operating system such as windows and mac-OS.

5.4.3 Subsystem Software Dependencies

PostgreSQL

5.4.4 Subsystem Programming Languages

The programming language used for the database is outside scope of the project.

5.4.5 Subsystem Data Structures

N/A

5.4.6 Subsystem Data Processing

N/A

5.5 Business Logic Subsystem

The business logic Subsystem is part of web application and is responsible to making logical decision required for the application to provide its service. For example: this Subsystem will decide whether a parents location is near the school pick up point or not and notify accordingly.

5.5.1 Subsystem Hardware

Hardware requirement is outside scope of this project.

5.5.2 Subsystem Software Dependencies

Business logic Subsystem is build using python programming language and utilizes geopy library to calculate the distance between school and the parents current location.

5.6 Layer Operating System

The business logic subsystem is not Operating System falls within the Backend Layer, so Windows, Mac and Linux distribution can be used.

5.6.1 Subsystem Data Structures

N/A

5.6.2 Subsystem Data Processing

N/A

6 APPENDIX A

Include any additional documents (CAD design, circuit schematics, etc) as an appendix as necessary.

REFERENCES