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

ARCHITECTURAL DESIGN SPECIFICATION
CSE 4317: SENIOR DESIGN II
SPRING 2021 - SUMMER 2021

TEAM CODEBENDERS
SCHOOL PICKUP AND DROP OFF APP

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

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	04.02.2021	SB	document creation
0.2	04.09.2021	SB, PB, UB, NK, MN	First Draft Complete
0.2	04.11.2021	SB, PB, UB, NK, MN	Updated Document
1.0	04.12.2021	AA, UB	Final Review for V1
1.1	04.16.2021	MN, UB	Removed iOS and Android Subsystems
1.2	04.17.2021	MN, UB	Redesign Diagrams
2	04.17.2021	MN, UB	Minor design changes
2.1	05.03.2021	UB	Removed redundancy in requirement
3	08.11.2021	SB, PB, UB, NK, MN	Final Version

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, 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' parents 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.

1.1 PURPOSE AND USE

The purpose of the application is to reduce the overall pickup and drop off time of students. This will be done providing the school information on which student's parent has entered the premises and dismiss them in a designated parking spot.

1.2 INTENDED AUDIENCE

School Pickup/Drop Off App is designed for schools and will be purchased by schools. This App will be used by schools and student's family.

2 SYSTEM OVERVIEW

The application will be separated into 3 layer: Mobile and Website UI, API Layer, and Backend Layer. The Mobile and Website UI 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 .
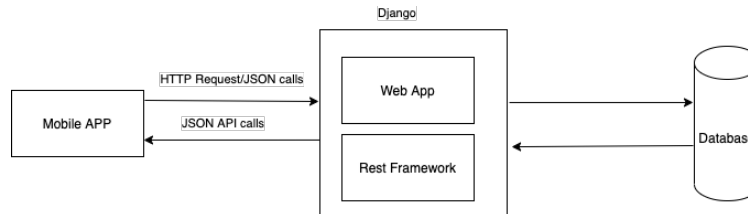

Figure 1: A simple architectural layer diagram

2.1 MOBILE AND WEBSITE SYSTEM LAYER DESCRIPTION

The Mobile UI 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 used to track live Geo-location and send coordinate to the faculty and staff. The Mobile UI will also have faculty and staff view for school system from where school staff can monitor parents locations.This system will receive data from the back end that parents device send. Once parent reach to the pickup location this will notify school staff. A separate website UI will be made available to school's administrative. The website UI can be used to make changes to user account information and other various administrative task.

2.2 API LAYERS

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.

2.3 BACKEND LAYER

This 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. It also handles the request for and from Website UI; The Django Rest Framework would be used to support RestAPI calls.

3 SUBSYSTEM DEFINITIONS & DATA FLOW

Here is the diagram that shows how data flow between different subsystem.
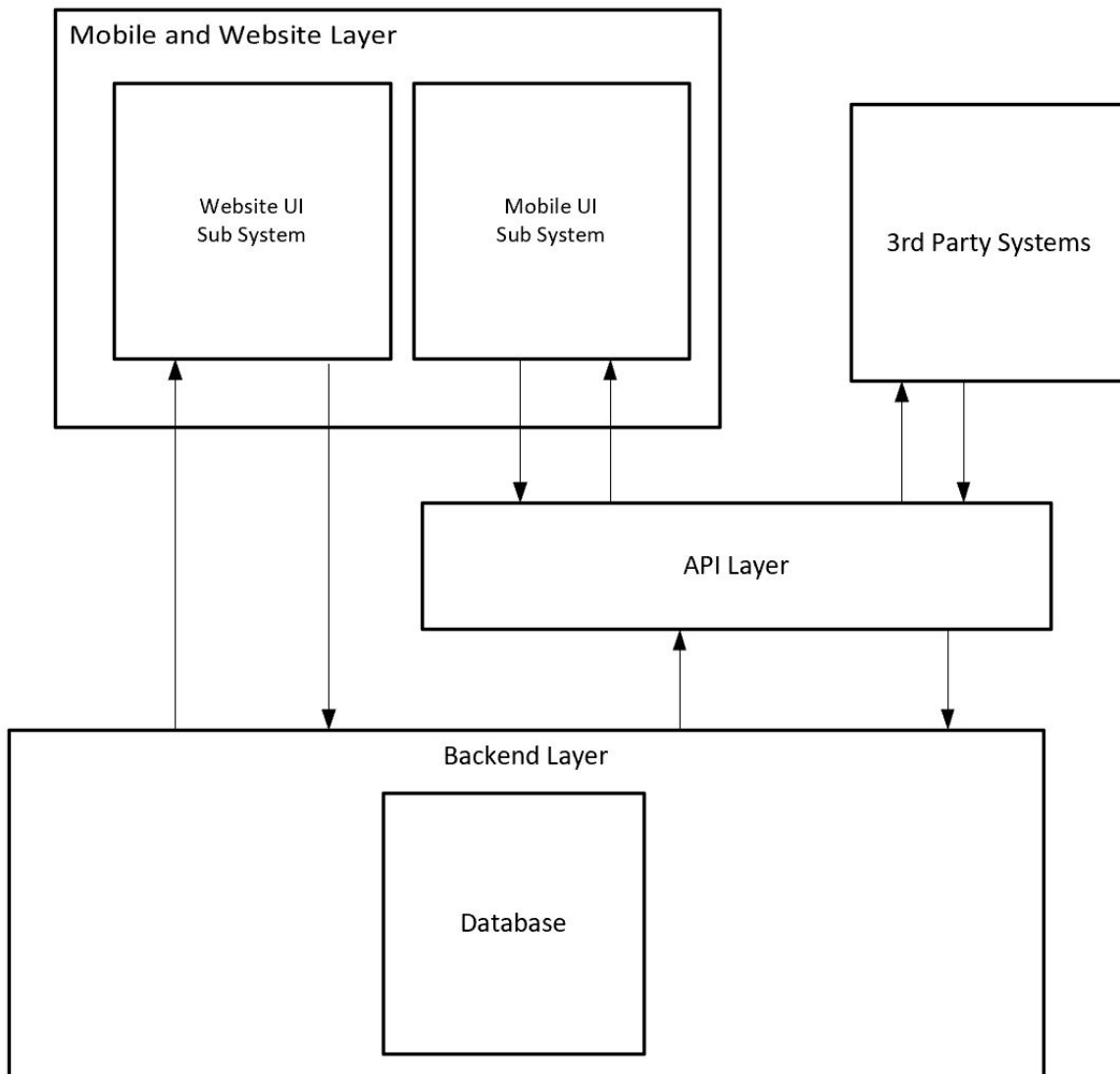

Figure 2: A simple data flow diagram

As shown in the figure above, Mobile UI and Website UI are part of Mobile and Website Layer. Any interaction by user will be communicated to backend. Website user interface will directly linked with backend using HTTP and Mobile UI will use API layer in the middle while communicating with the back end.

API layers primary objective will be to communicate information between Mobile layer and 3rd party system to backend and vice-versa.

Backend Layer is responsible for business logic and creating, updating, reading and deleting information from database.

4   MOBILE AND WEBSITE LAYER SUBSYSTEM

The mobile and website UI layer will allow uses to interact with the system and communicate the infraction with the backend using REST API and HTTP requests respectively. The mobile and website UI layer consist of sub-layer. The mobile UI allow user to register, login, share live location and receive notifications. The backend layer verifies user and connect user to their school system. The figure explain how data flow in this subsystem.
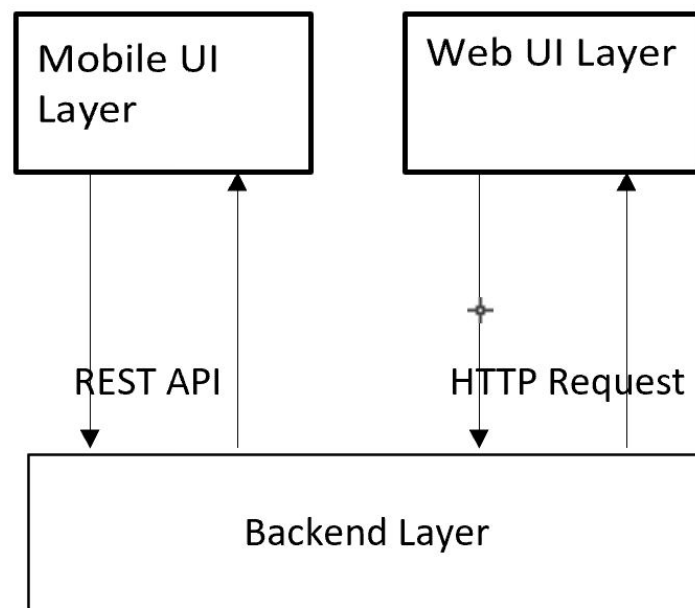

Figure 3: Overview of Mobile and Website Layer

4.1   MOBILE UI SUBSYSTEM

iOS and Android will be supported and Mobile UI SubSystem will be used to share live locations, provide information that are required to make pickup like pickup time, pickup location.
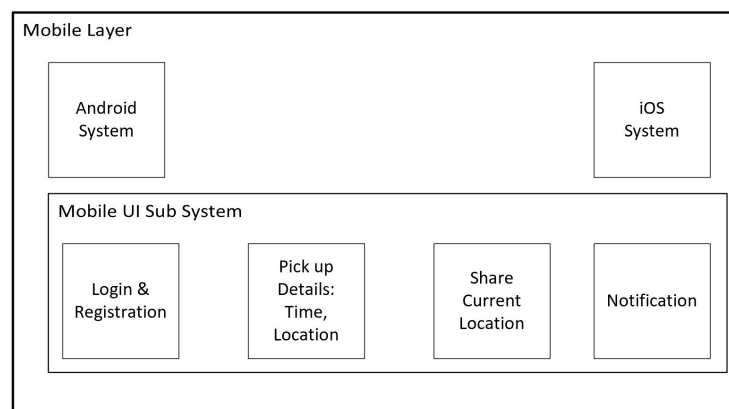

Figure 4: Example subsystem description diagram

4.1.1 ASSUMPTIONS

1) Users will allow to share their location to the System.
2) Users will have good internet connection to share and receive data from cloud.

4.1.2 RESPONSIBILITIES

This subsystem will be responsible for sending data to the database and receiving information provided by the servers and display it to the user.

4.1.3 SUBSYSTEM INTERFACES

The input will be text and geo location coordinate of the user device.

Table 2: Subsystem interfaces for Mobile UI

ID	Description	Inputs	Outputs
#1	Login and Registration	Text,Number	User created or logged in in system
#2	Schedule pickup time and Location	Text and time	Schedule created for pickup
#3	Share Live Location	GPS Coordinates	NULL
#4	Notification	message	Notification

4.2 WEBSITE UI SUBSYSTEM

This will be the user interface for school administrators to use. Website UI will used to perform administrative task such as add, edit and delete users. They make other changes such as add pickup locations and more.
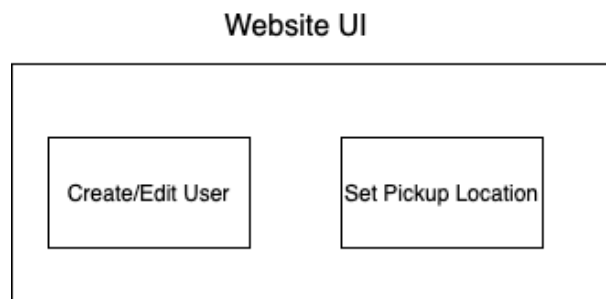

Figure 5: Example subsystem Website UI

4.2.1 ASSUMPTIONS

1) School have a accurate record of students currently enrolled, along with parents information.
2) School will have designated pickup location.

4.2.2 RESPONSIBILITIES

The school will keep the information such as student/parent record and pickup location up to date.

4.2.3 SUBSYSTEM INTERFACES

The input will be text and geo location coordinate of the user device.

Table 3: Subsystem interfaces for Website UI

ID	Description	Inputs	Outputs
#1	Create Parent User	Name, Username, Email, Password,Child Name, Address, Phone Number	Parent User Created
#2	Create Faculty/Staff User	Name, Username, Email, Password, Employee ID, Phone Number, Position	Faculty/Staff User Created
#3	Create Pickup location	Name of Location, Description, GPS Coordinates	New Location Created

5 API LAYERS

This layer works as mediator between the Mobile and the Backend interface and 3rd party systems. This layer will use REST architectural style for communication between the layers.
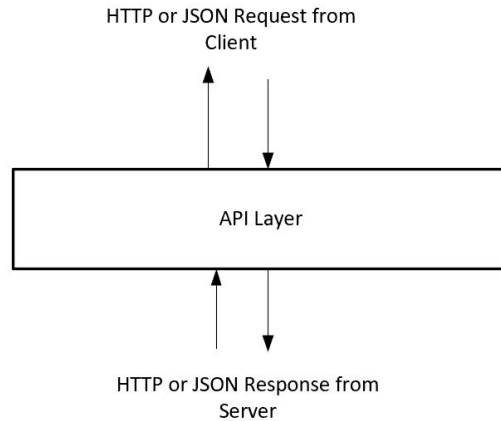

Figure 6: Example of API Layer's Communication

5.1 REST FRAMEWORK SUBSYSTEM

The Rest framework will take the JSON format request from the mobile interface. The framework will return the token object upon the successful authentication.

5.1.1 ASSUMPTIONS

The server will have fixed URL address and will be running 24/7.

5.1.2 RESPONSIBILITIES

The layers is responsible for communication between front end and back end.

5.1.3 SUBSYSTEM INTERFACES

Table 4: Subsystem interfaces for API Layer

ID	Description	Inputs	Outputs
#01	Authentication	{ "username":"ab" "password":"ab" }	{"token":"ab"}
#02	Authentication	{ "username":"abc" "password":"ab" }	{"Error":"Wrong Credentials"}
#03	Location	Coordinate	Null
#04	Notification	Input message	Message

6 BACKEND LAYER

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

6.1 DATABASE SUBSYSTEM

The database stores all the information such as user details, schools detail, staff detail etc.

6.1.1 ASSUMPTIONS

Database is always online and connected to backend system.

6.1.2 RESPONSIBILITIES

The database will create, store, and delete the user information.

6.1.3 SUBSYSTEM INTERFACES

Table 5: Subsystem interfaces for Database Layer

ID	Description	Inputs	Outputs
#01	Create, store, and delete user information	DB query	DB response

6.2 BUSINESS LOGIC SUBSYSTEM

The business logic Subsystem will be 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.

6.2.1 ASSUMPTIONS

The algorithm and values used in business Subsystem is correct.

6.2.2 RESPONSIBILITIES

Make business decisions using predefined values and algorithm.

6.2.3 SUBSYSTEM INTERFACES

Table 6: Subsystem interfaces for Business Logic

ID	Description	Inputs	Outputs
#01	Check if parent is near pickup area	GPS Coordinates	Boolean

REFERENCES