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

ARCHITECTURAL DESIGN SPECIFICATION
CSE 4317: SENIOR DESIGN II
SUMMER 2022

Bin Buddy

THE CLEAN TEAM
BIN BUDDY!

MOHAMMED AHMED
AMAY KADAKIA
SHUBHAYU SHRESTHA
NICHOLAS SOLIZ

Revision History

Revision	Date	Author(s)	Description
0.1	04.01.2022	MA , AK , SS , NS	Document Creation
0.2	04.10.2022	MA , AK , SS , NS	Complete Draft
0.3	04.25.2022	NS	Revision for Final Presentation
0.4	08.14.2022	MA , AK , SS , NS	Final Submission

CONTENTS

List of Figures

List of Tables

1 Introduction

Bin Buddy is a multi-platform application for homeowners and waste management companies that will be utilized to make the garbage experience far more admirable, including features such as garbage truck tracking and proper waste disposal methods.

1.1 Purpose and Use

The application will help track a garbage truck and inform the user if the truck has passed their home or they still have time to move their trash bins to the curb for the garbage truck to pick up. Additionally, the app will also be able to send reminder notification on a day prior to the scheduled garbage collection day.

1.2 Intended Audience

This application will be used by waste management companies and homeowners that utilize their services.

2 System Overview

This structure describes the four primary layers of the Bin Buddy application: UI, Data Processing, Database, and the Google Firebase (Server Backend) layer.

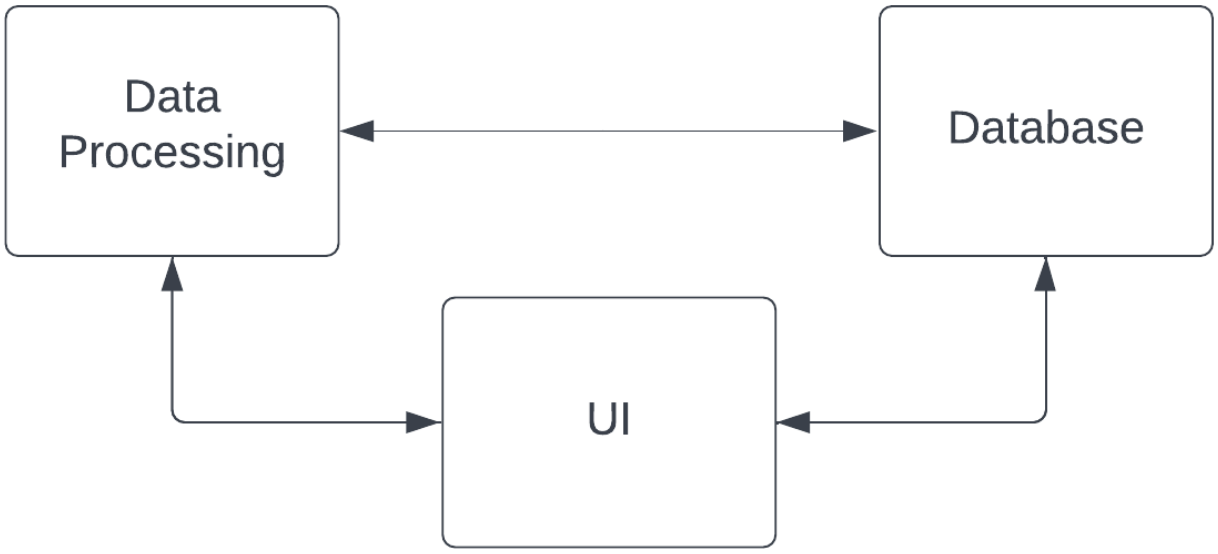

Figure 1: A simple architectural layer diagram

2.1 UI Layer

This is the layer that will handle much of the user input and display user output. This layer will allow homeowners to navigate through the application by utilizing the log in page, or they can register by entering their user details. The homeowner users can also view the location routes and maps as well as visit other information pages.

2.2 Data Processing Layer

This is the Data processing layer, which consists of two additional subsystems: Google Firebase Authentication and Route Processing and ETA. This layer will oversee the user authentication using Google Firebase and process ETA times based on the truck driverâs current location and their distance to the userâs residence.

2.3 Database Layer

This layer will act as the database for the application, where it will store login data, vehicle data, location data, and API keys.

3 SUBSYSTEM DEFINITIONS & DATA FLOW

This section breaks down your layer abstraction to another level of detail. Here you graphically represent the logical subsystems that compose each layer and show the interactions/interfaces between those subsystems. A subsystem can be thought of as a programming unit that implements one of the major functions of the layer. It, therefore, has data elements that serve as source/sinks for other subsystems. The logical data elements that flow between subsystems need to be explicitly defined at this point, beginning with a data flow-like diagram based on the block diagram.

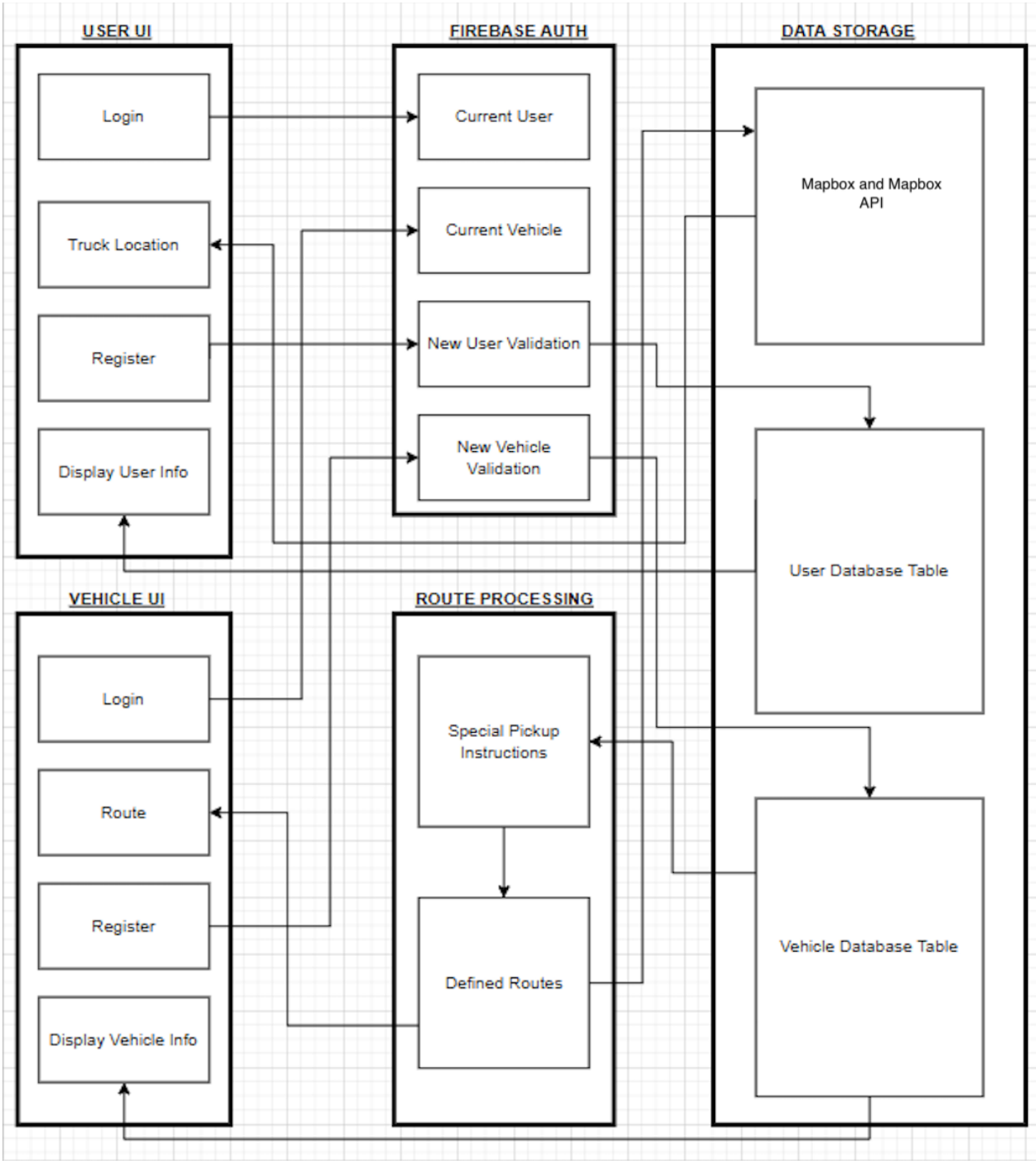

Figure 2: A simple data flow diagram

4 Application User Interface (UI)

4.1 Login

4.1.1 Assumptions

It is assumed that the user has an existing account and is not already logged in.

4.1.2 Responsibilities

The responsibility of this function is to take in the credentials from the user (ie username and password) and verify if they are correct. Then the user will be taken to their individual account.

4.1.3 Subsystem Interfaces

Table 2: UI Login Interfaces

ID	Description	Inputs	Outputs
#1	Login	Username Password	If the credentials are valid, take user to home page

4.2 Location and Route Map

4.2.1 Assumptions

It is assumed that the user has an existing account and is already logged in.

4.2.2 Responsibilities

The responsibility of this function is to display a map with the location of the garbage truck, as well as the Estimated Time of Arrival (ETA) of the truck.

4.2.3 Subsystem Interfaces

Table 3: Location Map Interfaces

ID	Description	Inputs	Outputs
#1	Location Map	Route Processing and ETA	- Map with the location of the garbage truck - Estimated Time of Arrival (ETA) of the truck.

4.3 Register User

4.3.1 Assumptions

It is assumed that the user does not already have an existing account.

4.3.2 Responsibilities

The responsibility of this function is to get information from the user and register their account.

4.3.3 Subsystem Interfaces

Table 4: UI Register User Interfaces

ID	Description	Inputs	Outputs
#1	Register User	- First Name - Last Name - Email - Address -Phone Number - Password	- User Database Table is updated - Account is created

4.4 User Information

4.4.1 Assumptions

The user information provided during registration is complete and correct.

4.4.2 Responsibilities

The responsibility of this function is to display user information that is stored in data storage for the user that is currently logged into the application. This functionality will not show any sensitive information of any user that is not the currently logged in user.

4.5 Register Vehicle

4.5.1 Assumptions

It is assumed that the vehicle being registered is not already registered under a different account.

4.5.2 Responsibilities

The responsibility of this function is to capture the information needed to register a new vehicle into a fleet of vehicles in the system database. The system will ensure that the data is entered fully by the user before submission of the vehicle registration.

4.6 Vehicle Information

4.6.1 Assumptions

The vehicle information provided during registration is complete and correct.

4.6.2 Responsibilities

The responsibility of this function is to display vehicle information that is stored in data storage for the vehicle that is currently logged into the application. This functionality will not show any information of any vehicle that is not currently logged in. df

5 Data Processing

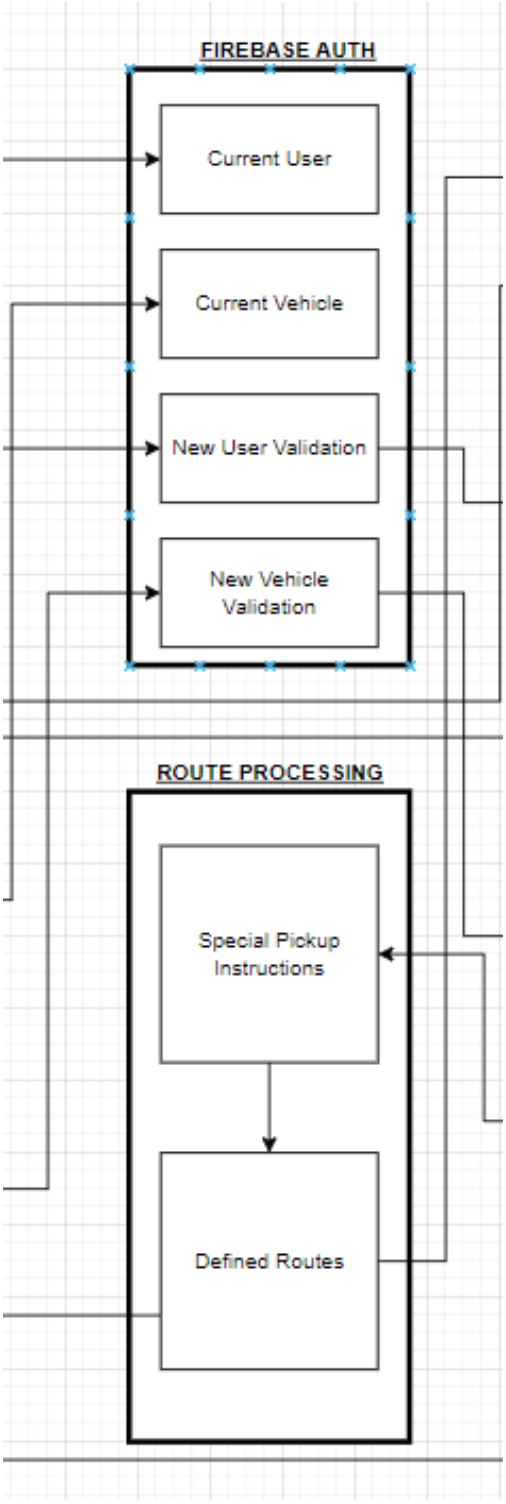

Figure 3: Data Processing

This is the Data processing layer, which consists of two additional subsystems: Google Firebase

Authentication and Route Processing and ETA. This layer will oversee the user authentication using Google Firebase and process ETA times based on the truck driver's current location and their distance to the user's residence.

5.1 Subsystem 1: Google Firebase Authentication

Google Firebase will let us Authenticate our users.

5.1.1 Assumptions

Google Firebase will have access to all user data, including username, Passwords etc.

5.1.2 Responsibilities

This subsystem will be responsible from validating the users into their accounts.

5.1.3 Subsystem Interfaces

This subsystem will have inputs from username, users passwords, users address. The output for this system will be to allow the user in to system.

Table 5: Login Data Processing

ID	Description	Inputs	Outputs
#1	Login	username password	If Valid, go to the next page if invalid, display invalid message
#2	User Address	Address	N/A

5.2 Subsystem 2: Route Processing and ETA

The subsystem will analyze the distance between a residential address and the current location of a truck driver to provide the user with a ETA.

5.2.1 Assumptions

The residential address needs to be on the route that the truck driver is following.

5.2.2 Responsibilities

Provide the user with the ETA for the truck driver.

5.2.3 Subsystem Interfaces

The inputs will be current location the driver and the users residential address. The output will be an ETA.

Table 6: Routing and ETA Data Processing

ID	Description	Inputs	Outputs
#1	ETA	Residential Address Truck's Current location	ETA

6 DATA STORAGE

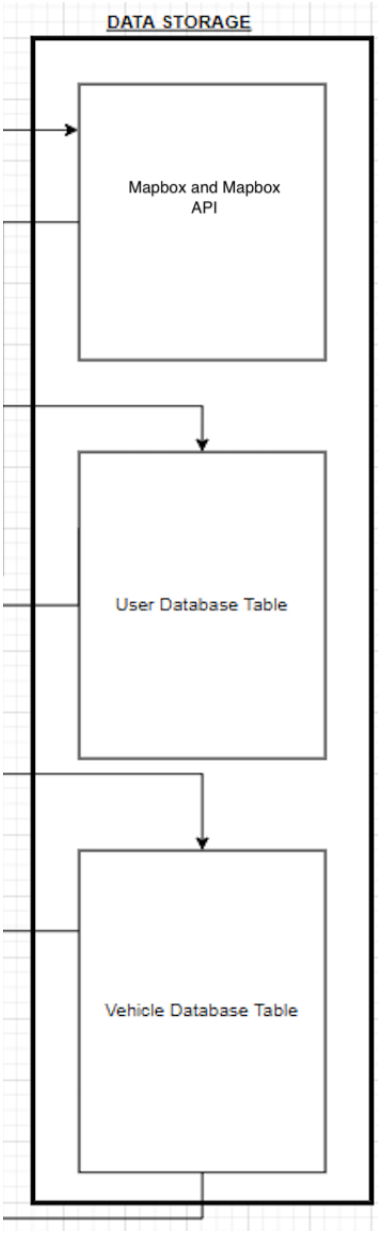

Figure 4: Data Storage Layer

This layer will act as the database for the application, where it will store login data, vehicle data, location data, and API keys.

6.1 API KEY TABLE

This subsystem will store all of the API keys for the application such as the Mapbox API and Geolocation API. These keys will provided to the code when called in order to limit accidental public exposure or leaks of API keys.

6.1.1 Assumptions

We assume that the database is able to communicate with the API key table and will be able to communicate with the other application layers (UI, Data Processing) in order to pass the API keys.

6.1.2 Responsibilities

This layer will store the API keys for the application.

6.1.3 Subsystem Interfaces

Table 7: API Key Table Interfaces

ID	Description	Inputs	Outputs
#1	API Key Storage	Mapbox API GeoLocation API Note: The above will be inputted utilizing the Database Manager	Output the API Keys utilizing the Data processing

6.2 User Information Table

This subsystem will store all of the user login credentials and account information.

6.2.1 Assumptions

We assume that the data processing layer is able to communicate with the database where it will be able to access the User Login Information Table to verify login credentials. This should also be able to communicate with the UI in order to make new accounts.

6.2.2 Responsibilities

This layer will store the user login credentials and account information.

6.2.3 Subsystem Interfaces

Table 8: User Information Table Interfaces

ID	Description	Inputs	Outputs
#1	User Information Table	Username Password Address Email Phone Number Name	Output information to the Data Processing layer

6.3 Vehicle Information Table

This subsystem will store all of the vehicle account information.

6.3.1 Assumptions

We assume that the data processing layer is able to communicate with the database where it will be able to access the Vehicle Information Table to verify credentials.

6.3.2 Responsibilities

This layer will store the organization's vehicle information.

6.3.3 Subsystem Interfaces

Table 9: Vehicle Information Table Interfaces

ID	Description	Inputs	Outputs
#1	Vehicle Information Table	VehicleID OrganizationID Pin	Output information to the Data Processing layer

References