Department of Computer Science & Engineering
The University of Texas at Arlington

Detailed Design Specification
CSE 4317: Senior Design II
Fall 2020

Dr. Coders
Pick-Bins

Utsav Dhungana
Himal Basnet
Sailesh Thapa
Suyash Ghimire
Bipul Karki

Revision History

Revision	Date	Author(s)	Description
0.1	09.24.2020	BK	document creation
0.2	09.24.2020	ST, SG, UD, BK, HB	complete draft
0.3	09.24.2020	UD	release candidate 1
1.0			official release

CONTENTS

List of Figures

List of Tables

1 Introduction

Pick-Bins aims to create a communication link between the waste management company and its users. We are going to build an mobile application which serves as a communication channel. The goal of this application is to inform its users about the schedule to collect the waste. The application will be cross-platform i.e it can run on iOS and android phones.

The application will be deployed in App Store and Google Play Store. The user can easily download the application and start using it. Application will receive a notification from a waste company server which includes information such as pickup time of the waste from user location. With this information, the users can dump the garbage and the pick up vehicle will collect it.

In case, the user miss the notification, they can send a request via the application to pick up the trash from their location, and the waste company will serve this request by sending the nearest or quickest driver from the location available to pick up the waste. Real time location of the driver will also be shown through the application in case one want to see how far is the driver. Main motive of creating the application is to make the task of waste management easier and efficient which can save time and make our environment cleaner.

2 System Overview

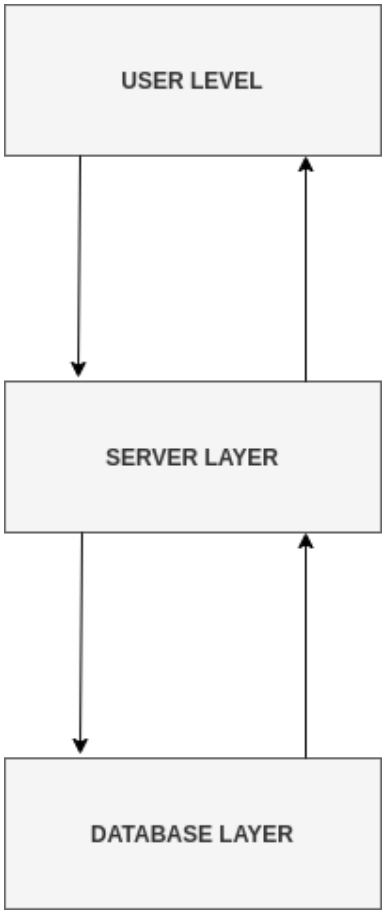

Figure 1: System architecture

2.1 User Layer

The user layer represents the user interface of the application. This layer handles taking input from the user and displaying the output on the screen. This layer serves various UI templates such as Start Screen, Login, Dashboard, Request Form, Schedule Display Screen, Task List Screen, Pick Up Confirmation Screen. This layer has two subsystems: User Authentication and User Interface.

2.2 Server Layer

Server layer is the bridge between user interface and database. It is also a control center for the application. Server layer includes Google Map API, Rest API as a subsystem for the application.

2.3 Database Layer

Database is the layer where all the information collected from the user or other important information will be stored for complete functioning of the application.

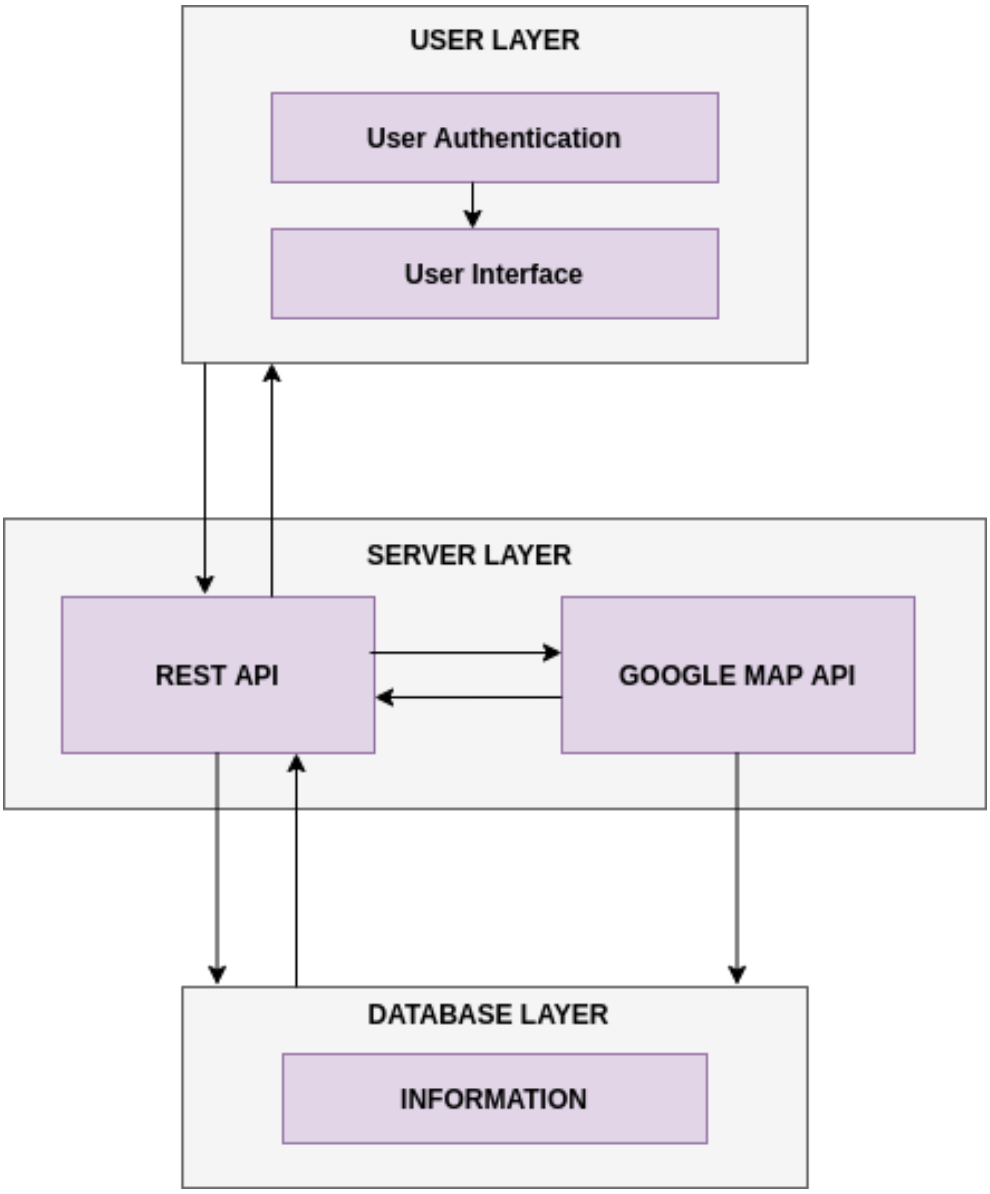

Figure 2: A simple data flow diagram

3 User Layer Subsystems

The user layer represents the user interface of the application. This layer handles taking input from the user and displaying the output on the screen. This layer serves various UI templates such as Start Screen, Login, Dashboard, Request Form, Schedule Display Screen, Task List Screen, Pick Up Confirmation Screen. This layer has two subsystems: User Authentication and User Interface.

3.1 Layer Hardware

Any device that runs on Andriod, iOS, and has webbrowers that supports java script and python will be able to run our software. For our project there are not any hardware designed or used.

3.2 Layer Operating System

Since this application runs in smartphones, the compatible operating system shall be Android OS or iOS. The web version of the application can run on any OS that supports some kind of web browser.

3.3 Layer Software Dependencies

Our software will be build under react native framework along with back-end support by Django. Libraries like npm, Django rest are used for this project.

3.4 User Authentication

The User Authentication subsystem closely interact with server layer to authenticate the user. This subsystem takes user credentials such as username and password and sends a login request to the server. If the login credentials are valid, user will be given access to the application. This layer also checks if the user is a customer or an employee. If the user is customer, the server will render Customer Interfaces. Else the Employee Interface will be rendered.

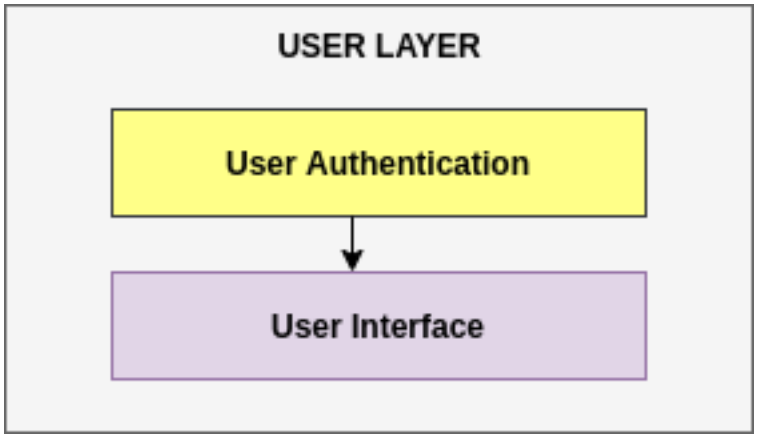

Figure 3: User Authentication subsystem description diagram

3.5 User Interface

This subsystem layer handles all the UI templates such as Dashboard, Request Form, Schedule Display Screen, Task List Screen, Pick Up Confirmation Screen.

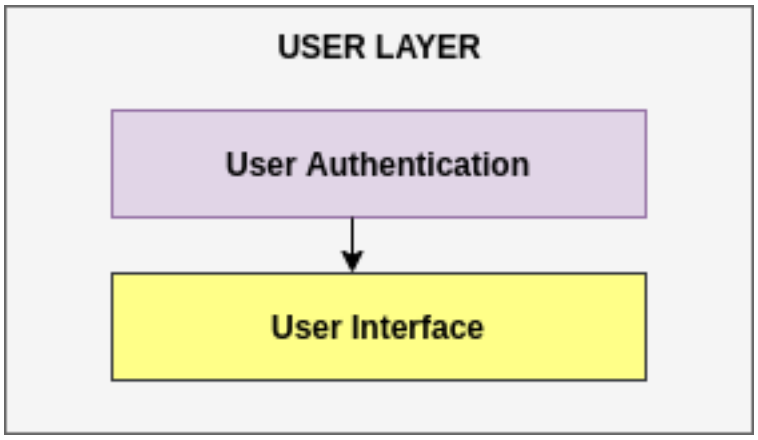

Figure 4: User Interface subsystem description diagram

4 Server Layer Subsystems

Server layer is the bridge between user interface and database. It is also a control center for the application. Server layer includes Google Map API, Rest API as a subsystem for the application.

4.1 Layer Hardware

Since we are using API which is based on cloud services there wont be any specific hardware involved for our system.

4.2 Layer Operating System

This layer can run on any OS that supports Django and React. The preferred OS is Linux.

4.3 Layer Software Dependencies

Our software will be build under react native framework along with back-end support by Django. Libraries like npm, Django rest are used for this project.

4.4 REST API

This subsystem act as an interface that supports communication between the user layer and the database layer. All the user input stored in the database via REST API. Similarly, all the required information are displayed on the user screen via REST API.

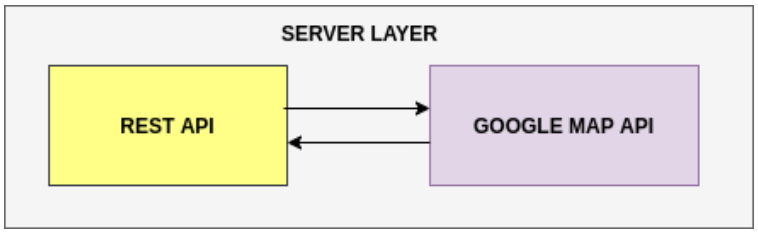

Figure 5: REST API subsystem description diagram

4.5 GOOGLE MAP API

This subsystem works to show location, search location, or calculate location distance through the google map interface. This API works closely with Rest API and will pass information to the database if needed in any case.

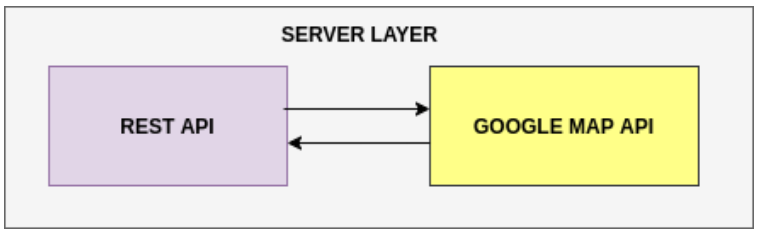

Figure 6: GOOGLE MAP API subsystem description diagram

5 Database Layer Subsystems

Database is the layer where all the information collected from the user or other important information will be stored for complete functioning of the application.

5.1 Layer Hardware

Since we are using API which is based on cloud services there wont be any specific hardware involved for our system.

5.2 Layer Operating System

This layer can run on any OS that supports Django and React. The preferred OS is Linux.

5.3 Layer Software Dependencies

Our software will be build under react native framework along with back-end support by Django. Libraries like npm, Django rest are used for this project.

5.4 Information

Rest API stores in and retrieves from the information, but the Google map API only stores the information in the database layer. Database layer acts as a storage option for any information that need to be stored within the application.

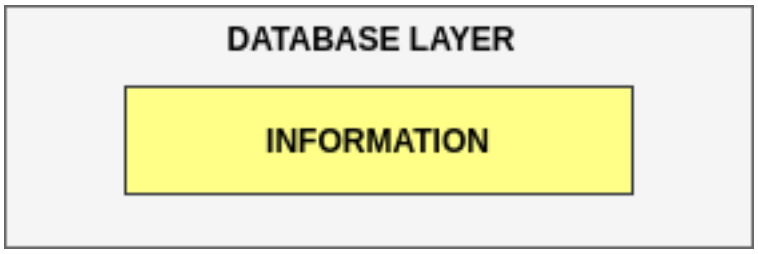

Figure 7: Information subsystem description diagram

6 APPENDIX A

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

References