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

Architectural Design Specification
CSE 4316: Senior Design I
Summer 2020

Dr. Coders
Pick Bins

Utsav Dhungana
Sailesh Thapa
Himal Basnet
Suyash Ghimire
Bipul Karki

Revision History

Revision	Date	Author(s)	Description
0.1	08.01.2020	BK	document creation
0.2	08.05.2020	ST, SG, UD, BK, HB	complete draft
0.3	08.13.2020	UD	release candidate 1
1.0			official release

CONTENTS

List of Figures

List of Tables

1 INTRODUCTION

The application 'PICK BINS' will help solve the waste management problems we've been facing. The confusion on specification between waste management company and customers not only leads to late service, time waste, diseases spread. With the help of this app we can create mutual understanding between customers and company to manage the waste product in our household. PICK BINS will help to create good understanding between customers and company. App is created in such a way that it will be user friendly. App provides user an interface through which they can navigate through various app features. App provides a display interface which shows the waste pick up schedule. App provides a request form through which user can immediately request driver for the pickup. There will be an interface which users will use to create a request which then is sent to waste pickup drivers later making a stop to pickup the waste. In this process there will be Google Map API and Rest API working in server layer while communicating with database to complete the task.

2 System Overview

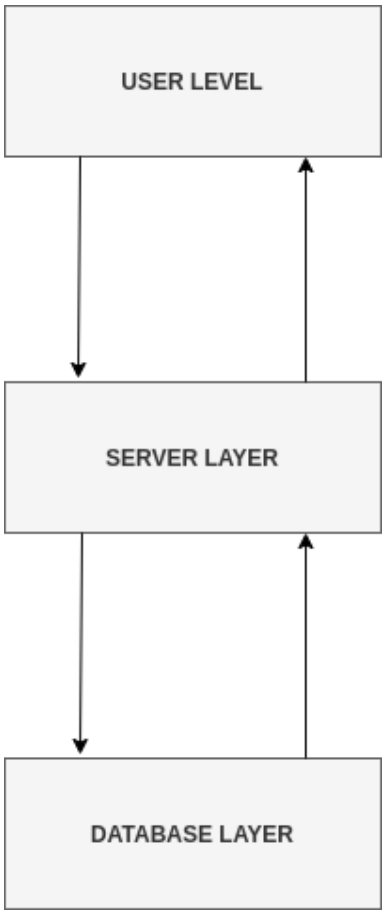

Figure 1: Architectural layer diagram

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.

3 Subsystem Definitions & Data Flow

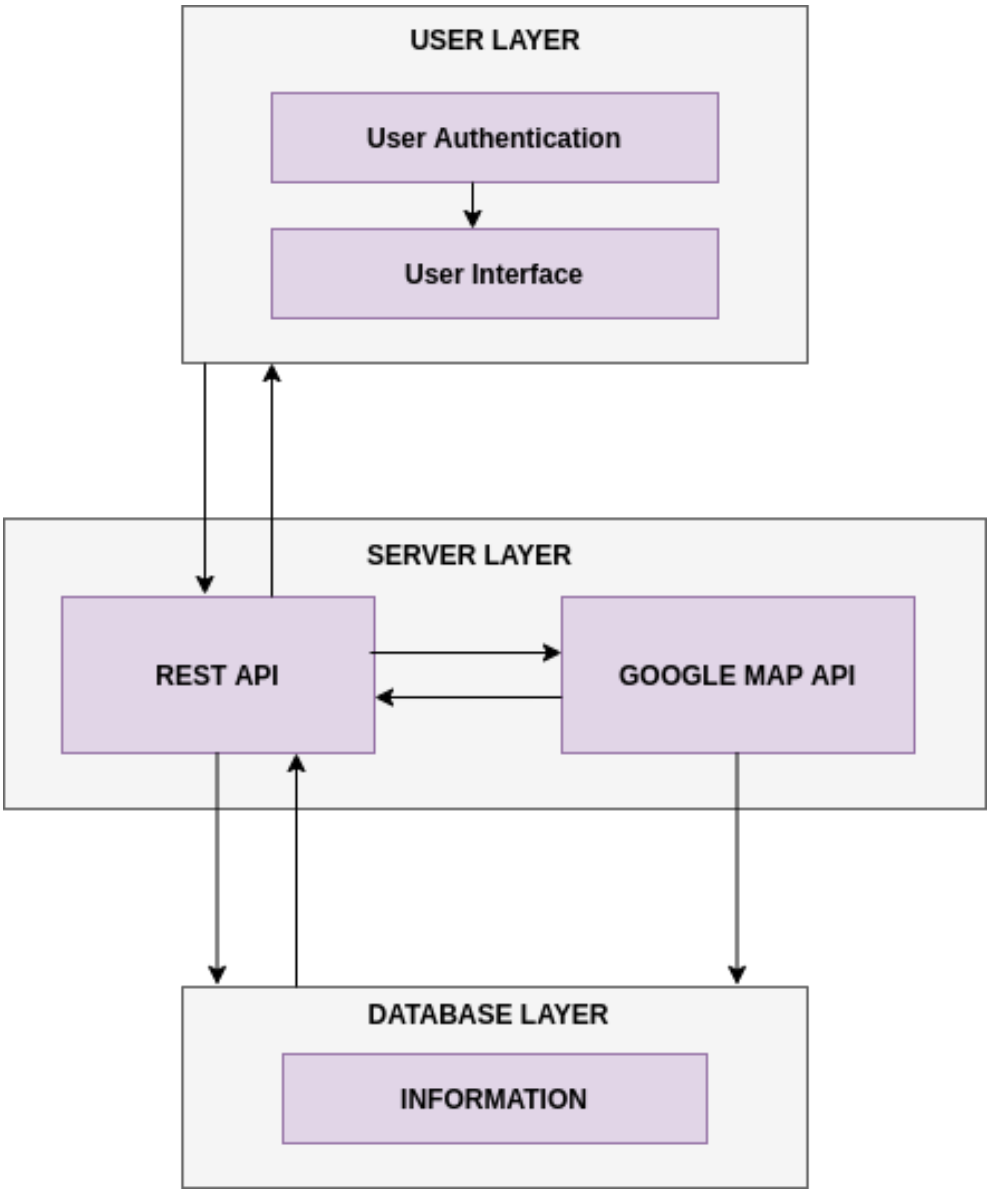

Figure 2: A simple data flow diagram

4 User Layer Subsystems

4.1 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.

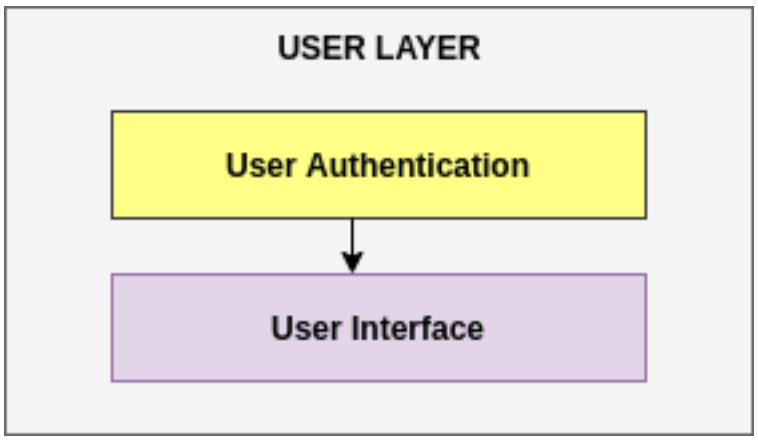

Figure 3: User Authentication subsystem description diagram

4.1.1 Assumptions

User Interface will be available in Web, Android, and IoS for initial release. Here authentication is required each time any type of user needs to access a user interface. All user function like request pickup, schedule pickup, support will be available through a simple user interface.

4.1.2 Responsibilities

The main responsibility of this layer is to validate user. It checks if the user is a customer or an employee. Depending upon the type of user, the specific templates will be displayed on the screen.

4.1.3 Subsystem Interfaces

Table 2: User Authentication Subsystem interfaces

ID	Description	Inputs	Outputs
1	Login Credential	username password	If invalid, prompt "Invalid credentials" message
2	Sign Up	Sign Up Button	Redirect to sign up screen to register user credentials

4.2 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.

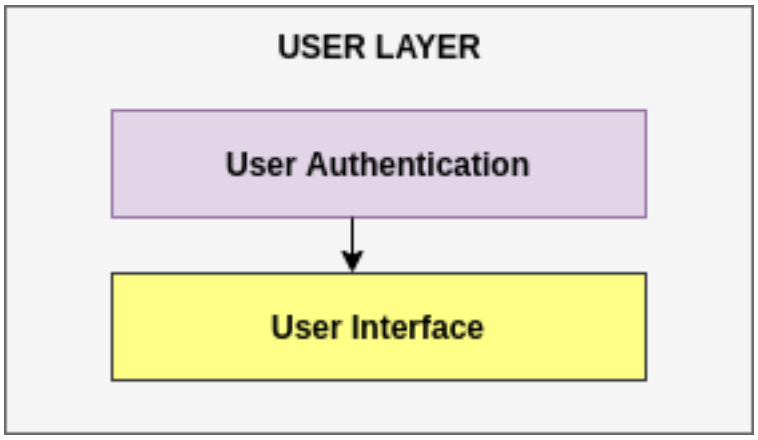

Figure 4: User Interface subsystem description diagram

4.2.1 Assumptions

User interface will include four main option first one is Dashboard where users basic and frequent information will be shown along with shortcut to other options, second one is Pickup Request where user can send their pickup request, third is support option to help user with various technical issues, last one is Account setting where users profile information and other settings will be saved and edited.

4.2.2 Responsibilities

The major responsibility of this layer is to display UI templates based the page requested by the user. For example, if the user press the dashboard icon, this layer is responsible for displaying the Dashboard Template on the screen.

4.2.3 Subsystem Interfaces

Table 3: User Interface Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Home Screen	Dashboard icon	Dash board template
#2	Pick Up Request	Form Data	Request Confirmation Message
#3	Support	Button for Contact Information, FAQs, Other options	List of FAQS, Text, Other option
#4	Account Setting	Button Click	Various account setting option to update

5 SERVER LAYER SUBSYSTEMS

5.1 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.

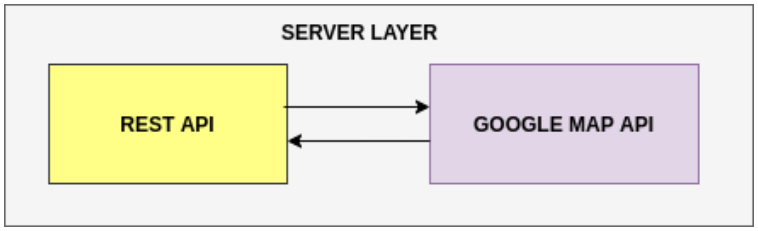

Figure 5: REST API subsystem description diagram

5.1.1 ASSUMPTIONS

Rest API will communicate with Google Map API with both way data flow within server layer. Also, it is will access and store information saved in database layer through multiple way data flow. Various interfaces like request button, login button, sign up button, history information will use Rest API to set and get information from database if necessary

5.1.2 RESPONSIBILITIES

This layer is an important part of the system which is responsible for handling user request and all the button events. For example, during the login process, REST API takes in user credentials and checks for the validation. This subsystem provides all the input parameters used in Google Map API. This layer is responsible for storing data in the database and access data from the database.

5.1.3 SUBSYSTEM INTERFACES

Table 4: Rest API Subsystem interfaces

ID	Description	Inputs	Outputs
1	Handle Request and button events	Request	Response to request or error message

5.2 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.

5.2.1 ASSUMPTIONS

Google Map API will be the subsystem under server layer that will be used to get location, calculate distance between two locations and supply those information to the Rest API. Google Map API will be transferring data to Rest API or getting data form Rest API as well. In case of communicating with other layer Google Map API will be sending data to Database Layer to store information.

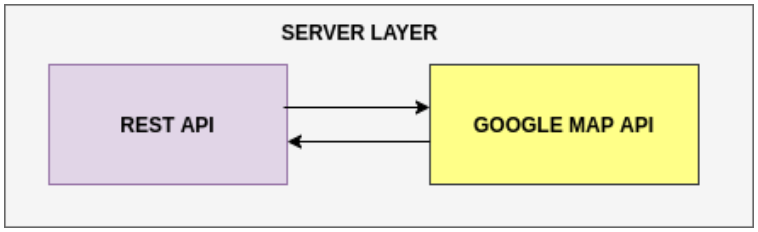

Figure 6: GOOGLE MAP API subsystem description diagram

5.2.2 Responsibilities

Google Map API is responsible for providing the interface of a map just like google map where user will be able to search location, pin location, find places, or find distance between two point and other information about location. Integrating this API in the application will give user a modern Each of the responsibilities/features/functions/services of the subsystem as identified in the architectural summary must be expanded to more detailed responsibilities. These responsibilities form the basis for the identification of the finer-grained responsibilities of the layer's internal subsystems. Clearly describe what each subsystem does.

5.2.3 Subsystem Interfaces

Table 5: Google Map API Subsystem interfaces

ID	Description	Inputs	Outputs
1	Distance Calculation	User location	shortest route and total distance of the route

6 Database Layer Subsystems

6.1 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.

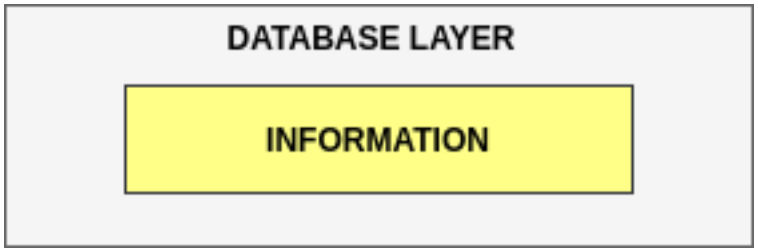

Figure 7: Information subsystem description diagram

6.1.1 Assumptions

The database will contain the information of customers, drivers, and other employees. The database layer will interact with the server layer subsystems and stores the required information accordingly.

6.1.2 Responsibilities

The main responsibility of this layer is to manage and store user information. This layer provides several query functions to allow the system to access the user data. The functions includes GetAllObjects, GetObjectById, UpdateAllObject, UpdateObjectById, DeleteObjectById.

6.1.3 Subsystem Interfaces

Table 6: Database Subsystem interfaces

ID	Description	Inputs	Outputs
1	Get data from API	user information	stores user data
2	Send data	user information	send data in object form

References