

Park In: An App-based Solution for Parking Vehicles

BY

ANIK GHOSH

ID: 191-15-12799

AND

MD. MOSTAIN BILLAH

ID: 191-15-12705

This Report Presented in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Computer Science and Engineering.

Supervised By

Dr. Sumit Kumar Banshal

Assistant Professor

Department of CSE

Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

JANUARY 2023

APPROVAL

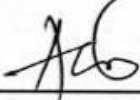
This Project titled “**Park In: An App-based Solution for Parking Vehicles**”, submitted by **Anik Ghosh** and **MD. Mostain Billah** to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 29th January 2023.

BOARD OF EXAMINERS

Dr. Touhid Bhuiyan
Professor and Head

Chairman

Department of CSE
Faculty of Science & Information Technology
Daffodil International University

 29-1-23

Arif Mahmud
Assistant Professor

Internal Examiner

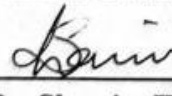
Department of CSE
Faculty of Science & Information Technology
Daffodil International University



Saiful Islam
Assistant Professor

Internal Examiner

Department of CSE
Faculty of Science & Information Technology
Daffodil International University

 29/01/2023

Dr. Shamim H Ripon
Professor

External Examiner

Department of CSE
East West University

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Dr. Sumit Kumar Banshal, Assistant Professor, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:



Dr. Sumit Kumar Banshal
Assistant Professor
Department of CSE
Daffodil International University

Submitted by:



Anik Ghosh
ID: 191-15-12799
Department of CSE
Daffodil International University



Md. Mostain Billah
ID: 191- 15-12705
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project successfully.

We really grateful and wish our profound our indebtedness to **Dr. Sumit Kumar Banshal** (Assistant Professor) Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “Mobile Application Development” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to **Professor Dr. Touhid Bhuiyan, Head**, Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

Our project is “**Park In**” An App-based solution for Parking Vehicles. This application helps people to park their cars in the nearest vacant parking space in a very short time. And there are many more benefits including a booking facility. On the other hand, homeowners who have vacant parking spaces can also come to our application and earn extra money by renting out their vacant parking spaces. By using our app, it is possible to reduce traffic congestion. Because most of the cities of our country, the main cause of traffic congestion is uncontrolled and illegal parking. We constantly see people parking their cars on the roadside, in front of various establishments. As a result, the road’s width is reduced and traffic jam is created. Finding the nearest parking space and parking there is also a troublesome task for car owners. That’s why our application will bring the right solution to the present day’s problems. Our app will show customers available parking spaces according to their location. And the customer can also book the parking space according to his choice. Promo codes are also available for customers. By using this, customers will get affordable parking facilities.

TABLE OF CONTENTS

CONTENTS	PAGE
Approval	i
Board of examiners	i
Declaration	ii
Acknowledgments	iii
Abstract	iv
CHAPTER	
CHAPTER 1: INTRODUCTION	1-3
1.1 Introduction	1
1.2 Motivation	1
1.3 Objectives	2
1.4 Expected Outcomes	2
1.5 Project Management and Finance	3
1.6 Report Layout	3
CHAPTER 2: BACKGROUND	4-6
2.1 Introduction	4
2.2 Preliminaries/Terminologies	4
2.3 Related Works	5
2.4 Comparative Analysis	5
2.5 Scope of the Problem	5
2.6 Challenges	6

CHAPTER 3: REQUIREMENT SPECIFICATION	7-17
3.1 Business Process Modeling	7-8
3.2 Requirement Collection and Analysis	9-10
3.3 Use Case Modeling and Description	11-15
3.4 Logical Data Model	15-16
3.5 Design Requirement	16-17
CHAPTER 4: DESIGN SPECIFICATION	18-21
4.1 Front-end Design	18
4.2 Back-end Design	19
4.3 Interaction Design and User Experience (UX)	19
4.4 Implementation Requirements	20-21
4.5 Permissions Required	21
CHAPTER 5: IMPLEMENTATION AND TESTING	22-35
5.1 Implementation of Database	22-24
5.2 Implementation of Front-end Design	24-33
5.3 Implementation of Interaction Design	33
5.4 Testing Implementation	34
5.5 Test Results and Report	35
CHAPTER 6: IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY	36-37
6.1 Impact on Society	36
6.2 Impact on Environment	36
6.3 Ethical Aspects	37
6.4 Sustainability Plan	37

CHAPTER 7: CONCLUSION AND FUTURE SCOPE	38
7.1 Conclusion	38
7.2 Limitation	38
7.3 Future Scope	38
REFERENCES	39

LIST OF FIGURES

FIGURES	PAGE NO
Figure 3.1: A business process model for the entire parking system	7
Figure 3.2: Business Process Model for User's Registration	8
Figure 3.3: ERD Model for the Entire Parking System	10
Figure 3.4: Use Case Diagram for User	11
Figure 3.5: Use Case Diagram for Parking Owner and Admin	12
Figure 3.6: Flowchart for user login	13
Figure 3.7: Flowchart for Parking Owner/ Admin Login	13
Figure 3.8: Flowchart for user	14
Figure 3.9: Logical Data Model for parking system	16
Figure 5.1: Firebase Authentication	22
Figure 5.2: Cloud Firestore Database	23
Figure 5.3: Cloud Firestore Database	23
Figure 5.4: Firebase Storage	24
Figure 5.5: Screenshot of splash screen	24
Figure 5.6: A Screenshot of Phone number verification	25
Figure 5.7: A Screenshot of OTP verification	25
Figure 5.8: A Screenshot of the Registration page	26
Figure 5.9: A Screenshot of the user's home screen	26
Figure 5.10: A Screenshot of the confirmation booking page	27
Figure 5.11: Screenshots of Online payment method	27
Figure 5.12: A Screenshot of the booking confirmation message	28
Figure 5.13: A Screenshot of the User's dashboard	28
Figure 5.14: A Screenshot of the Monthly parking system	29
Figure 5.15: A Screenshot of the promo code using	29

Figure 5.16: Screenshots of the User's Profile, Booking, Transaction	30
Figure 5.17: A Screenshot of the Admin/Parking Owner Login page	30
Figure 5.18: A Screenshot of the Parking owner's Registration Page	31
Figure 5.19: A Screenshot of the Admin's/Parking Owner's Dashboard	31
Figure 5.20: A Screenshot of the Admin's/Parking owner's dashboard	32
Figure 5.21: Screenshots of Add parking spaces	32
Figure 5.22: Screenshots of the profile, bookings, transactions, and parking spaces	33

LIST OF TABLES

TABLES	PAGE NO
Table 5.1: Testing Objective of our application	34
Table 5.2: Result of Test Objectives	35

CHAPTER 1

Introduction

1.1 Introduction

Park In is an App based solution for parking vehicles. This app will help car owners transform parking worries into relief. Most of the time car owners face various problems regarding parking their cars at a workplace. It is possible to solve these problems very easily by using our application. After registering on our app, customers can view parking spaces in their vicinity. After selecting the available parking spaces, the customer will go to the booking page. There he will see the booking time, cost, and address. He has to pay first to book the vacant parking space of his choice. After completing the booking, the customer can park the car in his designated parking space. On the other hand, parking owners can add their vacant parking spaces. Then they can track the bookings and transactions of their rented parking spaces. Besides, the authorities can rent any parking space and add it to the app. So that customers can easily get enough parking space in their vicinity. Through this entire work, parking will become easy as well as traffic congestion in every city will be eased.

1.2 Motivation

This application is used to find the nearest parking space and park the car at the specified location. It is also used to rent out vacant parking spaces. The purpose of this app is to make parking easy for those who face various problems while parking. Because finding parking space is very difficult in terms of Bangladesh. Also, due to parking the car everywhere, the car owner has to face various worries including damage to the car. At present, traffic jam is one of the problems of Bangladesh, the main causes of traffic congestion are uncontrolled and illegal parking. We constantly see people parking their cars on the roadside, in front of various establishments. As a result, the road's width is reduced and traffic jam is created. So, we have taken the initiative to create this app to create a well-equipped parking environment and eliminate the suffering of traffic jams in Bangladesh. We will update this system in the future.

1.3 Objectives

This application focuses on smart parking systems. This system is for those who want safe and hassle-free car parking. And those who want to make money by renting out their vacant parking spaces. Everyone can be registered there at any time.

The goals of our system are:

- User and Parking owner can register there at any time.
- Users can park their car at any time, any place if there have vacant parking space.
- User can use promo code to save money.
- Parking owners rent out their vacant parking spaces.
- This application helps to reduce traffic congestion.
- The system can save people valuable time and reduce wasting fuel.
- Helps to Reduce air pollution.
- Authority can manage the full system.
- Online registration and payment system.
- User and Parking owner can view bookings and transactions.

1.4 Expected Outcomes

The system is made to help people who want safe and hassle-free car parking. We hope all car owners will get a safe and hassle-free car parking facility by using this app. Also, they don't have to worry about vehicle damage. Through this well-equipped parking system, it is possible to create a traffic-free environment in every city in Bangladesh. The system can save people valuable time and reduce wasting fuel. Besides, homeowners who have vacant parking spaces can also come to our application and earn extra money by renting out their vacant parking spaces. By using our application, customers can easily get enough parking space in their vicinity. All in all, we think our application will help create a well-equipped parking system and reduce traffic jams in Bangladesh.

1.5 Project Management and Finance

The system requires very little staff for project management. Organizers can manage all processes and activities. The system requires less financial assistance. With less team management, the cost of this system is very low and maintenance costs are low.

1.6 Report Layout

A graduation degree allows us to learn a lot and apply it in real life. As a result, if we build this system, we will be able to put our ideas into action and use them to benefit humanity. The system is our concept, and it is brought to life through implementation. The first chapter includes an introduction to the task, goals, motivations, expected outcomes, and report design. The second chapter begins with an introduction to the project, related work, comparative studies, the scope of the problem, and the challenges of our task. The third chapter includes requirements specification, which includes use case modeling and description, a logical data model, and design requirements. The fourth chapter shows a complete site rendering that is identified by design specifications such as front-end design, back-end design, interaction design, user experience (UX), and implementation requirements. Implementation and testing are covered in chapter five. This covers front-end design, interaction, test execution, test execution outcomes, and test results and reporting. The complete task is finished in the last section. This report includes our framework's application, placement, and impact.

CHAPTER 2

Background

2.1 Introduction

As a developing nation, ours places a high priority on having a digital parking system. The parking system in our country is still not that developed. All it has is a traditional parking system. Everything from booking to payment is an old method. With the development of technology, people need an automatic parking system. Where everything from booking to payment can be done online. Break out of uncontrolled parking and find a fearless parking solution. In addition, Homeowners can rent out their vacant garages. Keeping in mind all the above needs we have developed this app which will be a blessing in real life. Data limit requests have been cleaned up. Using the power of the Firebase Database, the entire user interface is organized using an XML design.

2.2 Preliminaries/Terminologies

The system has three panels:

1. User panel
2. Parking owner panel
3. Admin panel

The user panel is a panel for all users. In this panel, users can find parking spaces, book parking spaces hourly or monthly, use promo codes, make payments, and view transactions and bookings. In the parking owner panel, an owner can add parking space for hourly, daily, or monthly, track his parking space's previous, ongoing, and upcoming bookings, and track transactions. In the admin panel, the admin can same as the parking owner. Admin can add parking space on rent from any owner through contract.

2.3 Related Works

There are some applications and websites available about this concept. We have study on these applications and websites. Such as Park Mobile, Spot Hero, Park me, Parking Koi, etc. Park Mobile, Park me, and Spot Hero are mainly U.S.A-based applications and make operation in other cities. Parking koi is a web-based application in Bangladesh.

2.4 Comparative Analysis

We researched several applications and websites concerning the concepts of online parking systems. The majority of them were created for people in the United Kingdom, the United States, and Australia. Unfortunately, there is no operational such application in our country. As a result, we attempted to concentrate on Bangladesh's prospects. Our primary goal is to maintain the concept of an App-based solution for parking vehicles. So, we're introducing an online parking system as an Android app. The system is not overly complex. The administration panel is simple to use, and a single administrator can manage the entire system. We attempted to back up my application.

2.5 Scope of the Problem

There are various types of problems that could arise in the system that we built.

- Web and Remote Services are not used.
- When the errand is in progress, there is a risk of errors.
- Without location service app doesn't work properly.

2.6 Challenges

We all face challenges in our daily lives. For this project, we faced multiple difficult situations. We must include various details about the system and its work processes, which are extremely difficult to calculate. The hardest part was creating a normal Android app. User satisfaction and trust issues are the main challenges. Sometimes users want to change their data. User-friendliness is also a challenge. In that case, implementing a payment method would be very difficult. In that case, collecting user opinions is much more difficult.

CHAPTER 3

Requirement specification

In this chapter, we will go over all of the requirements that will be required to complete the proposed system. This chapter describes the use case diagram, description, and other requirements.

3.1 Business Process Modeling:

A business process model is implemented to describe a project's data flow. It symbolizes the partnership between the program and the user and can be modified as the business sees fit.

Business Process Model for Entire parking system:

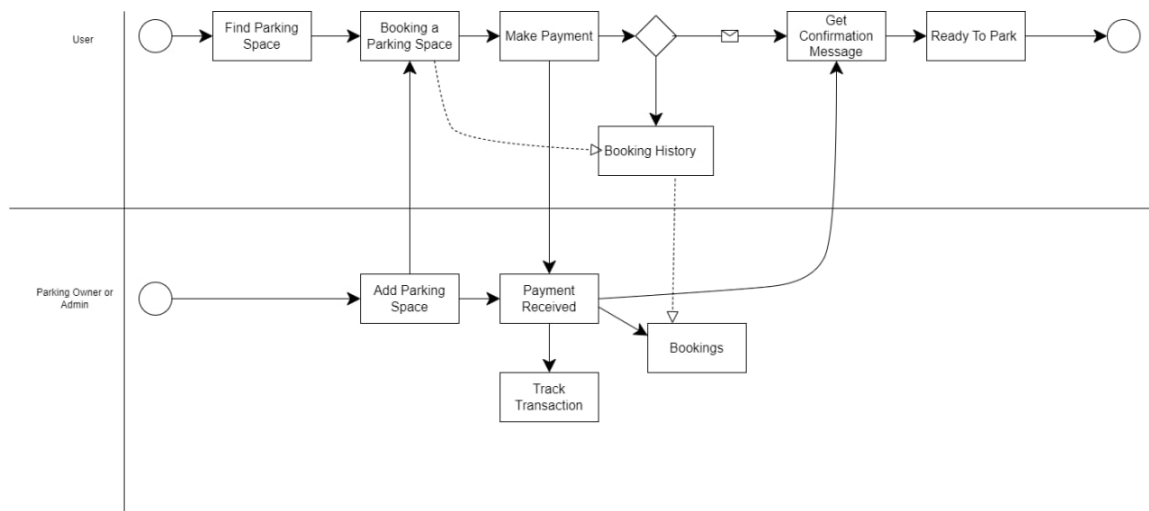


Figure 3.1: A business process model for the entire parking system

Business Process Model for User Registration:

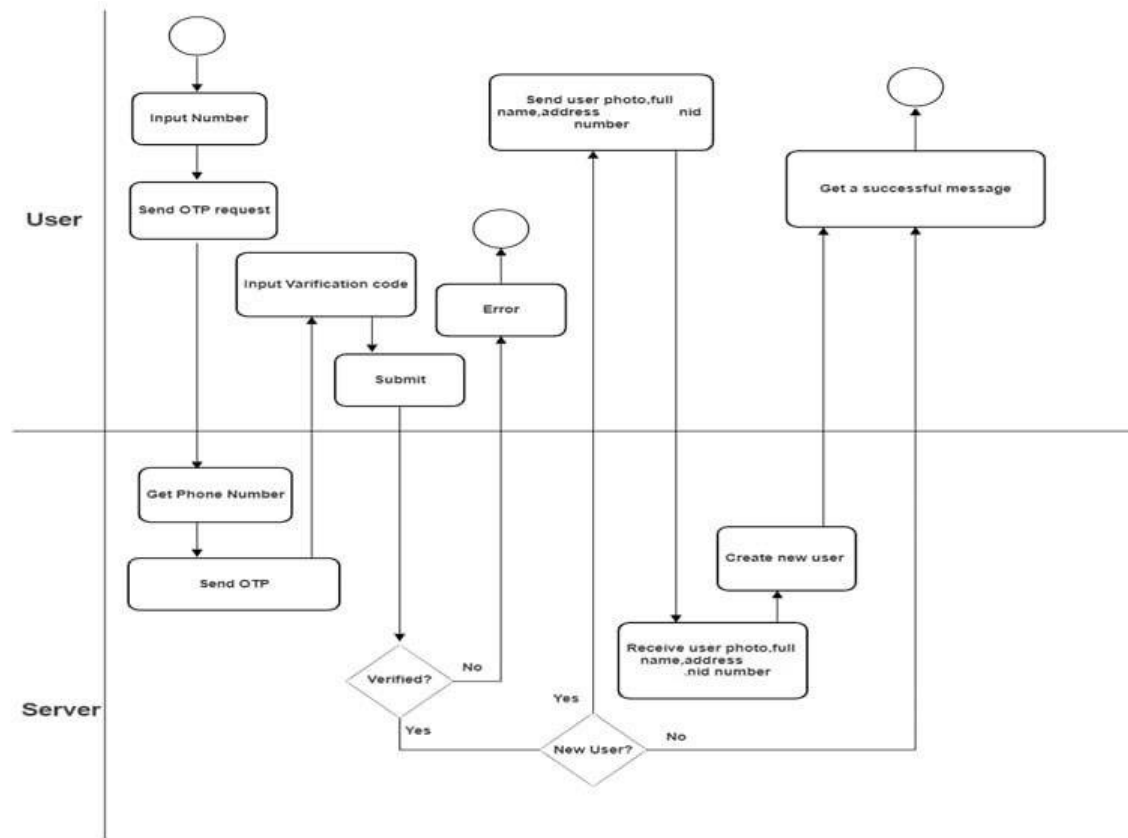


Figure 3.2: Business Process Model for User's Registration

3.1.1 Business Process Model Description:

First of all, the User inputs his phone no to verify, and he will get a verification code from the server and input the verification code. After that, he will register his/her self by providing all information. After that user goes to the home page where he/she can see the vacant parking space. Then the user can book a parking space for his/her suitable time. The user has to pay first for confirming his booking. After payment user can park his car in the parking space. On the other hand, the parking owner/admin can add parking space hourly or monthly. They can view bookings and transactions of his garage.

3.2 Requirement Collection and Analysis

Requirement Analysis, also known as Requirement Engineering, is the process of defining user expectations for a new software being built or modified. In software engineering, it is sometimes referred to loosely by names such as requirements gathering or requirements capturing. Requirements analysis encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, and analyzing, documenting, validating, and managing software or system requirements. In our application, there are two types of requirements. Such as non-functional and functional requirements.

Functional Requirements for User:

- Login
- Registration
- Find Parking Space
- Parking space booking
- Payment system
- Use promo code
- Monthly Parking

Non-Functional Requirements for User:

- View bookings
- View transaction
- View profile

Functional Requirements for Parking Owners:

- Login
- Registration
- Add daily parking space
- Add monthly parking space
- Total earning

Non-Functional Requirements for Parking owners:

- View bookings
- View transaction
- View profile
- View parking space

3.2.1 Entity Relationship (ER) Diagram Model

ER diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes, and relationships.

ERD Model for the Entire Parking System:

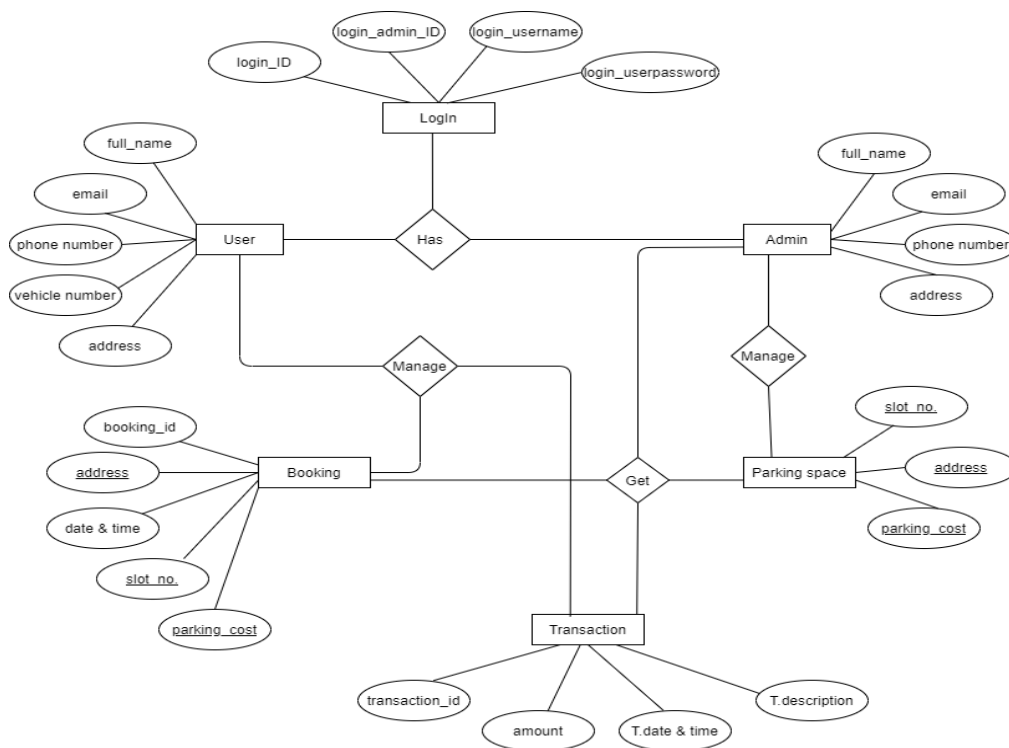


Figure 3.3: ERD Model for the Entire Parking System

3.3 Use Case Diagram

A use case diagram depicts all of the system's action flows with primary and secondary users. In general, primary users are human beings, while secondary users are technological creations. For administrators and users, we've attached the use case diagram.

Use Case Diagram for User:

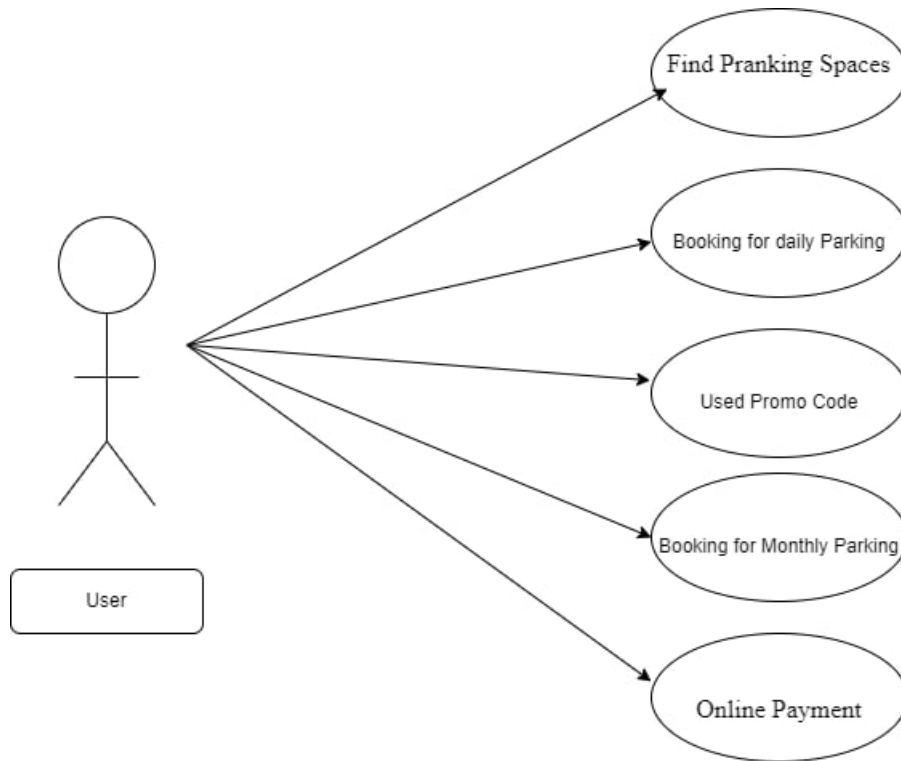


Figure 3.4: Use Case Diagram for User

Use Case Diagram for Parking Owner and Admin:

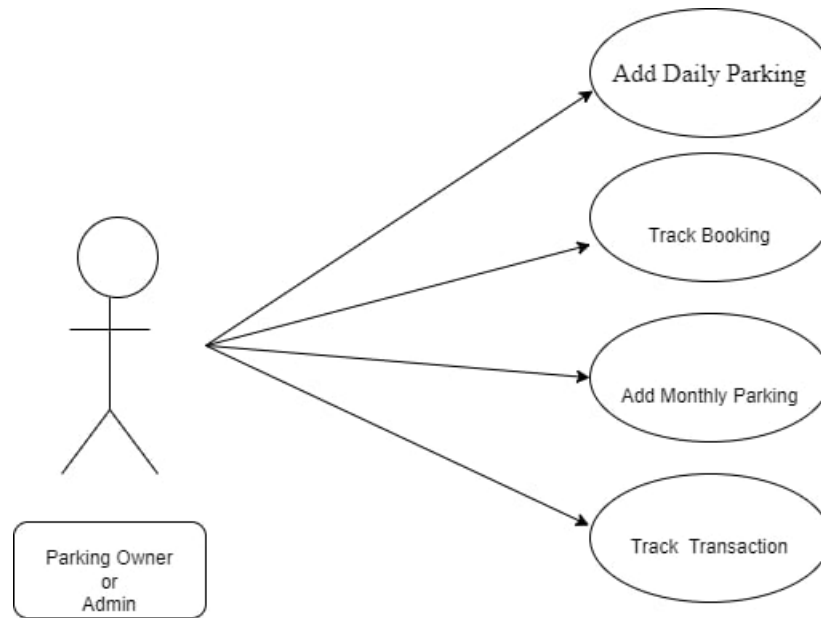


Figure 3.5: Use Case Diagram for Parking Owner and Admin

3.3.1 Use Case Description

Figure 3.4 Describe the project's use case diagram for the user. Users can find parking spaces, book the desired parking space, use promo codes for discounts, and book parking space for a month. The user has to pay online.

Figure 3.5 Describe the project's use case diagram for the parking owner or admin. Here Parking owner or admin can add daily parking space, add monthly parking space. And also, they can track their parking space bookings and transactions.

3.3.2 Flowchart:

A flowchart is a diagram that depicts an algorithm. To solve a problem, programmers frequently use it as a program-planning tool. It employs symbols that are linked together to represent the flow of information and processing. "Flowcharting" refers to the process of creating a flowchart for an algorithm.

Flowchart for user login:

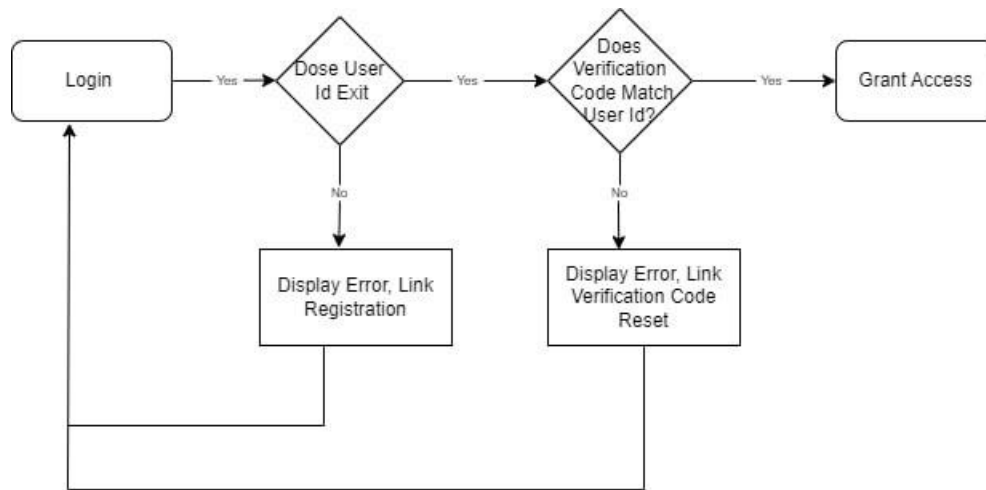


Figure 3.6: Flowchart for user login

Flowchart for Parking Owner/Admin Login:

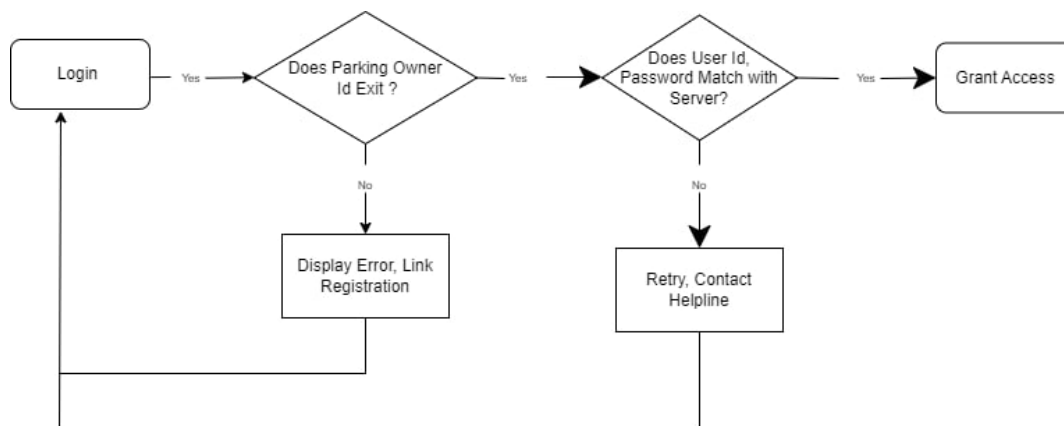


Figure 3.7: Flowchart for Parking Owner/ Admin Login

Flowchart for user:

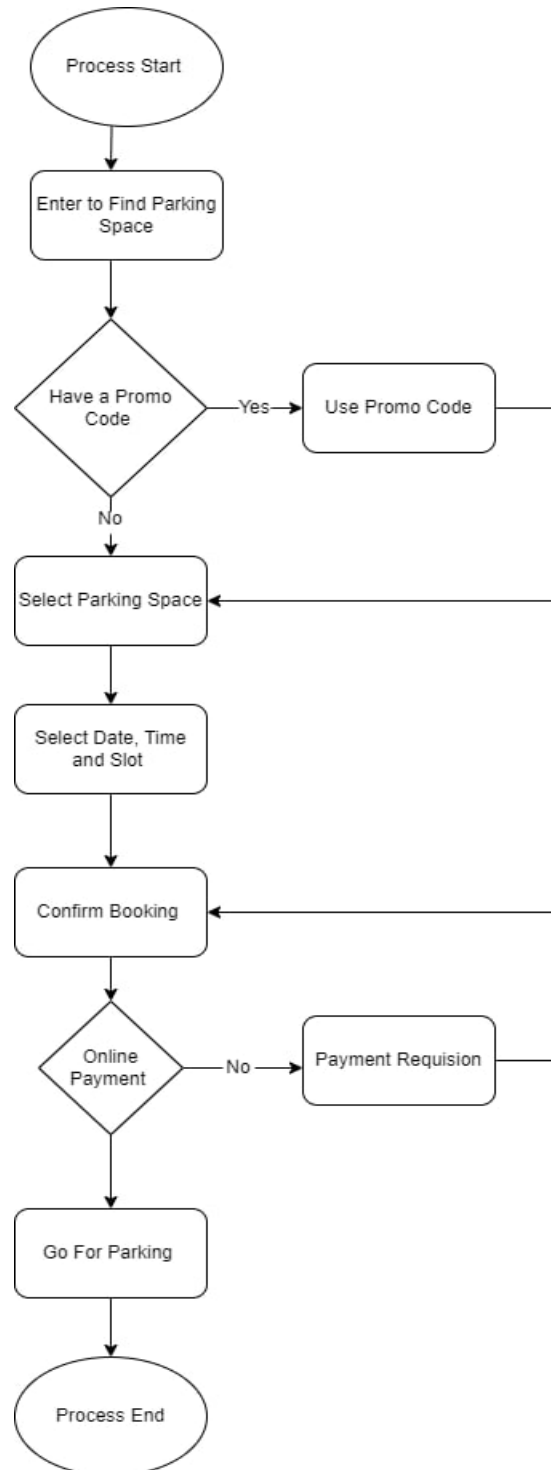


Figure 3.8: Flowchart for user

3.3.3 Flowchart Description:

Figure 3.6: Describes the project's flowchart for user login. Firstly, the user inputs his/her phone number. If the user id does not exist then he has to register his id and then go for the login. In the next stage, the user gets a verification code, and if the code match with user id, then the server gives access, and if does not match then the server resends the code.

Figure 3.7: Describes the project's flowchart for the parking owner/admin login. Firstly, the garage owner inputs his/her user id and password. If the user id does not exist then he has to register his id and then go for the login. In the next stage, the parking owner gives that information for login, and if the user id and password match with the server, it will give permission.

Figure 3.8: Describes the project's flowchart for users. After successful login, the user finds a parking space. If the user has a promo code, he can use it for discounts. Then the user selects the parking space and inputs the date, time, and slot for his/her booking. After confirming the booking, the user has to pay the amount for parking. If payment is completed user gets a document and goes for parking.

3.4 Logical Data Model

A logical data model establishes the structure of data elements and the relationships among them. It is independent of the physical database that details how the data will be implemented. The logical data model serves as a blueprint for used data. The logical data model takes the elements of conceptual data modeling a step further by adding more information to them.

Logical Data Model for parking system:

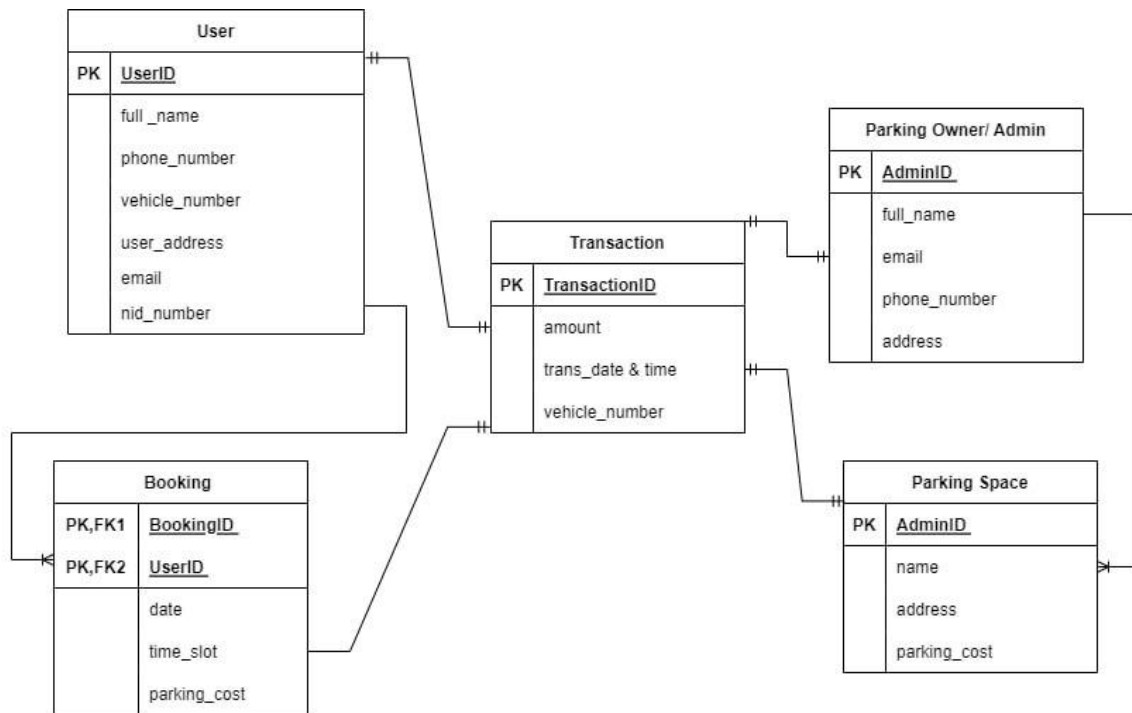


Figure 3.9: Logical Data Model for parking system

3.5 Design Requirement

Design requirements are usually the only thing that gives an idea of the overall working of a project. Our master plan is simple to implement. We are attempting to plan the application so that our customers can use it.

The design requirement for User:

- Login
- Find Parking Space
- Select parking space

- Book a parking space
- Online payment
- View Bookings
- View Transaction
- View Profile
- Use promo code
- Book monthly parking

The design requirement for Parking Owner/Admin:

- Login
- Add daily parking space
- Add monthly parking space
- View Bookings
- View Transaction
- View Parking Space
- View dashboard
- View profile

Chapter 4

Design specification

4.1 Front-end Design

The front-end design is where your application's users interact with it. We attempted to make the UI (user interface) as simple as possible. The front-end design of the application is done in XML, which stands for Extensible Markup Language. The application was prototyped using additional tools such as Adobe XD and Photoshop.

XML: Extensible Markup Language

Extensible Markup Language (XML) is a markup language with a set of rules for coding documents for human and machine readability. XML is defined by the W3C XML 1.0 configuration and several other system parameters. Unicode provides extensive support for a wide range of human languages. Due to Android's excellent XML support and user-friendliness, I used it in this application.

SVG: Scalable Vector Graphics

SVG is an XML-based vector image that represents 2D graphics and can be animated and interacted with. The World Wide Web Consortium, also known as the W3C, created SVG. SVG images and behaviours can be edited and changed using XML code because they are clearly defined in XML files. SVG files can be opened and edited in virtually any text editor, including vim, Nano, and others.

Adobe Photoshop:

Adobe Photoshop was only available for Windows and Mac computers. Photoshop is highly customizable and simple to use, with a wide range of tools. Newer terms like Photoshopping are also popular these days. Photoshop supports a variety of colour spaces, including RGB and CMYK. Though since Photoshop is so popular and simple to use, we used it for my preliminary sketches of the application design.

4.2 Back-end Design

Backend design is, in essence, what happens behind the scenes of your program. Java is the most widely used language worldwide for developing Android applications. Java thus functioned as the main backend language for this project. We used the real-time, user-friendly Firebase database as database, which is easy to link to your application.

Kotlin:

Kotlin is a general-purpose, cross-platform, statically typed programming language featuring type inference. The Kotlin standard library's JVM implementation uses the Java Class Library, which supports more concise syntax thanks to type inference. Although it can also be compiled into JavaScript or native code via LLVM, Kotlin primarily targets the JVM. The Kotlin Foundation is responsible for trademark protection, whereas JetBrains is in charge of language development expenditures.

Firebase:

Firebase is referred to as a real-time database since each alteration to the database is visible to the instance. Firebase databases are sometimes known as tree databases, because databases do not follow the traditional row and table format of databases to which we are used. Utilize NoSQL for querying and Volley for data transport.

4.3 Interaction Design and User Experience (UX)

Interaction design is the process of creating a dialogue between users of systems or services. This is one of the most important aspects of your application. Well-designed interactions lead to an excellent user experience. The user experience should be considered first before development.

4.4 Implementation Requirements

We utilized a variety of tools, properties, and components to properly construct this Android app. The tools, qualities, and elements used to create an Android application project and make it appealing to your users are all described in the Implementation Requirements section.

4.4.1 Android Studio

Google's official IDE for building Android applications is entitled Android Studio. This IDE allows you to design apps and launch them on the Android operating system. Numerous programming languages and development tools are supported by this IDE. contains rich libraries to aid developers in creating apps quickly. So, I made this project on Android Studio.

4.4.2 Basic Android Overview

There are several apps available on Android. To put it plainly, a complete software stack for mobile devices. It includes numerous upgrade libraries that will make creating this application easier and quicker. The program's design was done in XML and was built in Java. We used the Firebase Tools option in the IDE to connect to Firebase services. The project development process takes time.

4.4.3 The Emulator

An emulator also referred to as an Android Virtual Device, is required to run Android applications on your desktop computer (AVD). We have attempted the application several times with this emulator. You can create many Android devices and test your application on them with the aid of this emulator.

4.4.4 Android SDK

The software development kit is referred to as SDK. For the Android platform, the Android SDK is a collection of development tools that may be used to create various types of Android applications. This is a crucial tool for creating and running any Android application.

4.5 Permissions Required

- Internet permission.
- Media & storage permission
- Location permission
- SMS permission

CHAPTER 5

Implementation and testing

We'll provide a sample implementation of our project in this chapter and assess the project as a whole for efficiency.

5.1 Database Implementation

Data for applications is kept in databases. Secure databases are frequently used by developers to store data. Additionally, database users can add, update, and delete data. There are various kinds of databases, including Firebase, MySQL, and NoSQL.

5.1.1 Database Design

To store data in our Android application, we choose Firebase as a data manager. A real-time database is not available if you use a MySQL server instead time database. We used Firebase in our project to take advantage of the real-time database. We've attached screenshots of the project's backend database.

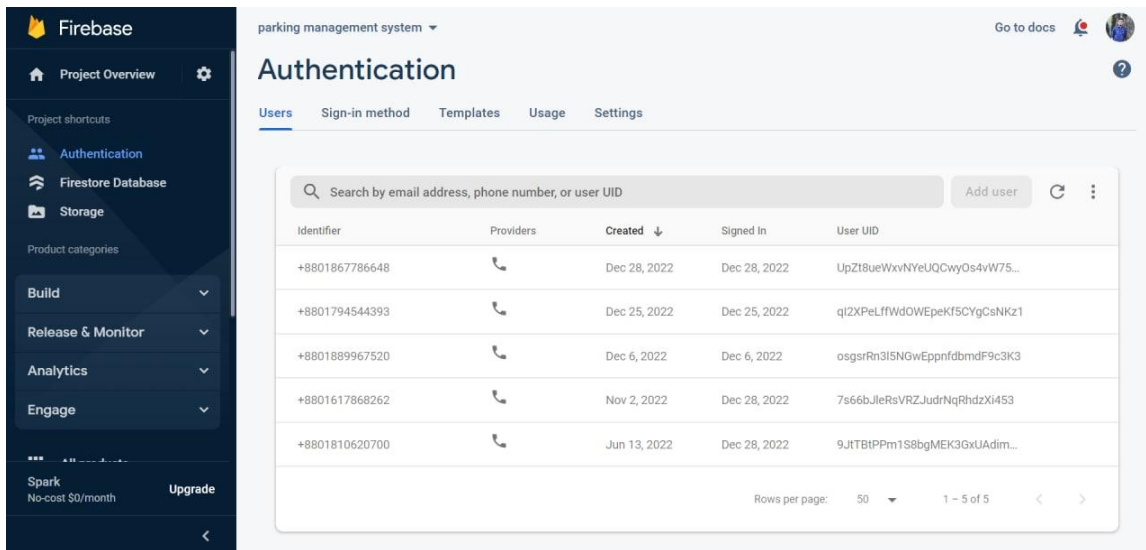


Figure 5.1: Firebase Authentication

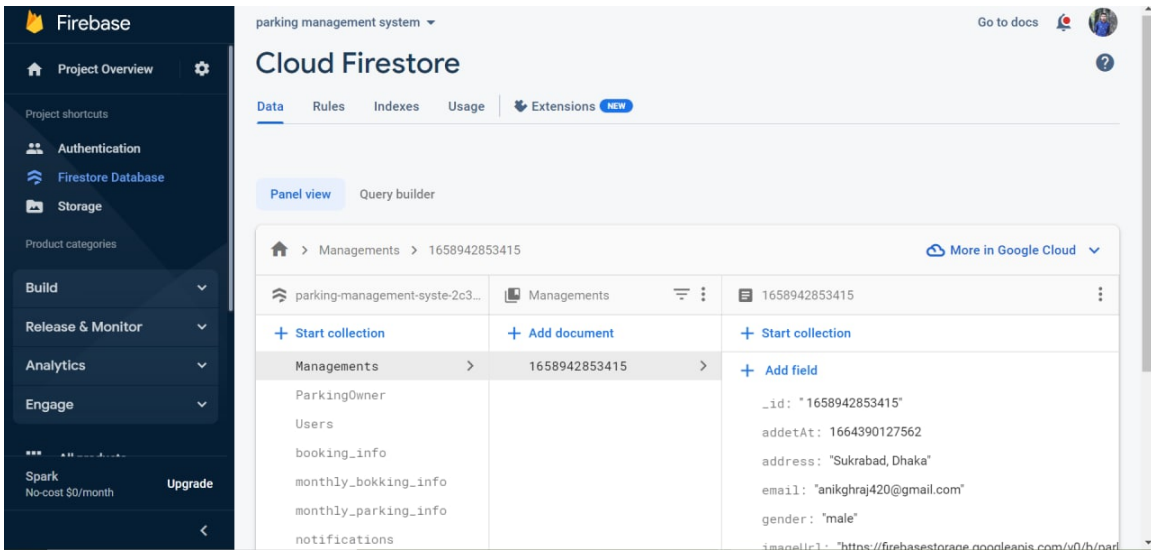


Figure 5.2: Cloud Firestore Database

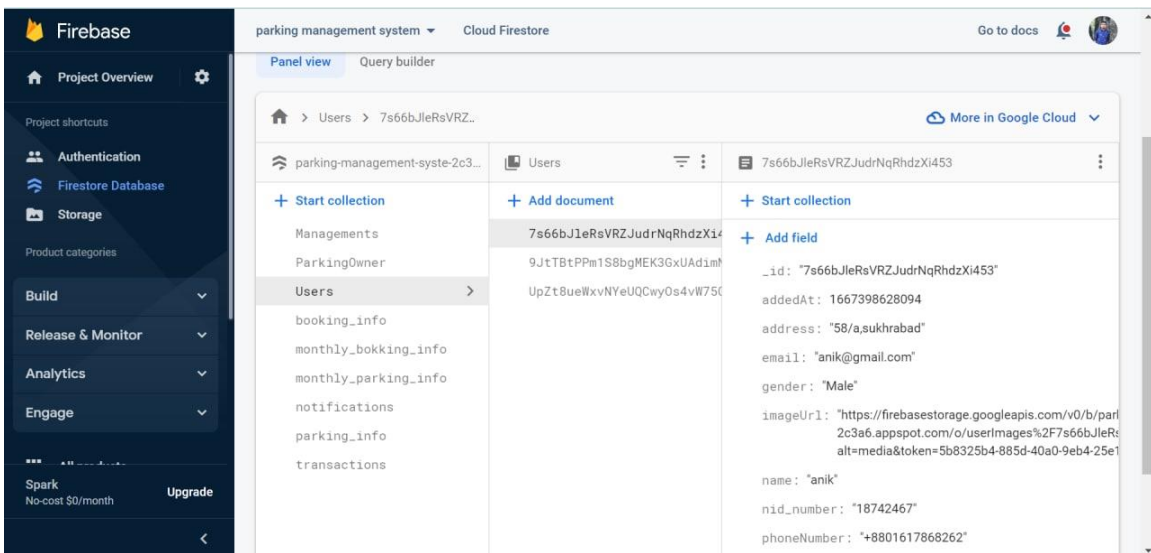


Figure 5.3: Cloud Firestore Database

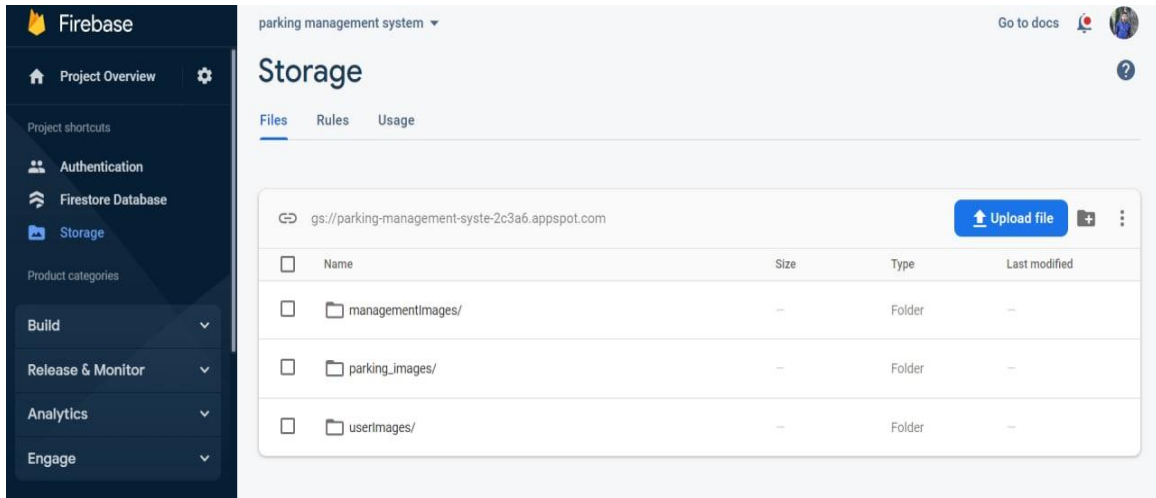


Figure 5.4: Firebase Storage

5.2 Implementation of Front-end Design

The main challenge is to develop a visually appealing and simple front-end design that enables users to immediately understand the functionality of the program and utilize it effectively. For the front-end design, we used XML.



Figure 5.5: Screenshot of the splash screen

In figure 5.5, We have attached the screenshot of the starting page as a Splash Screen of our application with the logo.

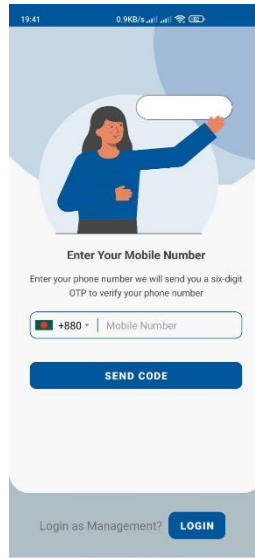


Figure 5.6: A Screenshot of Phone number verification

In figure 5.6, We have attached the screenshot of the Register/Login page of our application with the logo, tagline a login button. If a user wants to enter our application, he needs to verify OTP with his phone number.

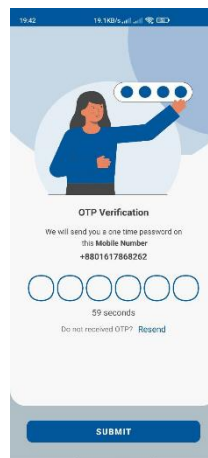


Figure 5.7: A Screenshot of OTP verification

Phone number verification activity has attached in figure 5.7. Our application will automatically fill out the verification code. The user can get a new OTP, by clicking on Resend code to receive the code again. After submitting the code, the user can verify their phone number. Phone number verification is needed while registering/Login into the application.



Figure 5.8: A Screenshot of the Registration page

In figure 5.8, a new user can register by giving full name, email address, address, NID number, vehicle number, and profile image.

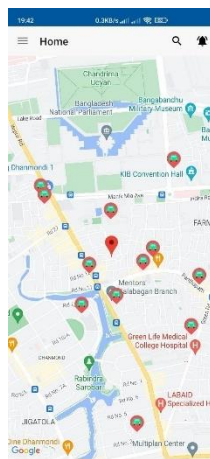


Figure 5.9: A Screenshot of the user's home screen

In figure 5.9, We have attached the home of a user. On this page, the user can see the nearest vacant parking space and choose one of those. They get different types of parking

spaces and different types of prices. It's could be easy to get a parking space at a cheaper price.

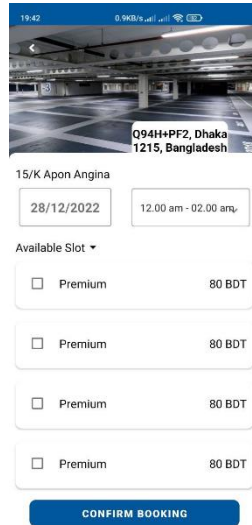


Figure 5.10: A Screenshot of the confirmation booking page

In figure 5.10, it's a confirmation booking page of a user. On this page, the user can see the parking space name, address, hourly cost, available slots, and parking space photo. And they can select the date, time slot, and slot price for the parking.

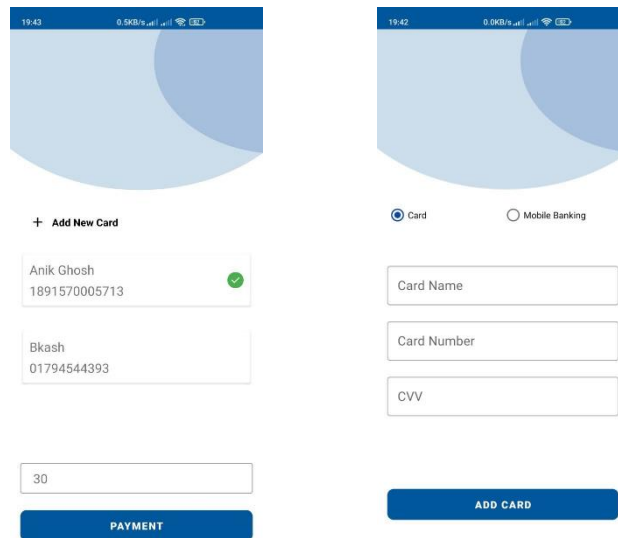


Figure 5.11: Screenshots of Online payment method

In figure 5.11 we've attached the payment method page of a user. Users can pay online for confirming their booking. And also, users pay with debit cards, credit cards, and mobile banking systems. Users can save a card or account for the next payments.

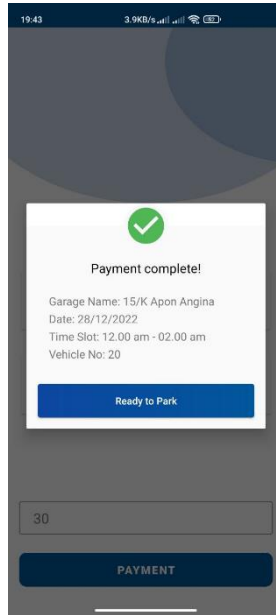


Figure 5.12: A Screenshot of the booking confirmation message

In figure 5.12 user gets a document to enter the parking space. This message provides the details about the booking. Such as parking space name, booking date, and time slot, user vehicle number, and payment status.

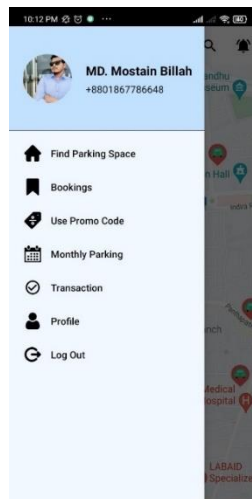


Figure 5.13: A Screenshot of the User's dashboard

Using the navigation drawer, they can find “Bookings”, “Use Promo Code”, “Monthly parking”, “Transaction”, “Profile”, and “Logout” options in figure 5.13.

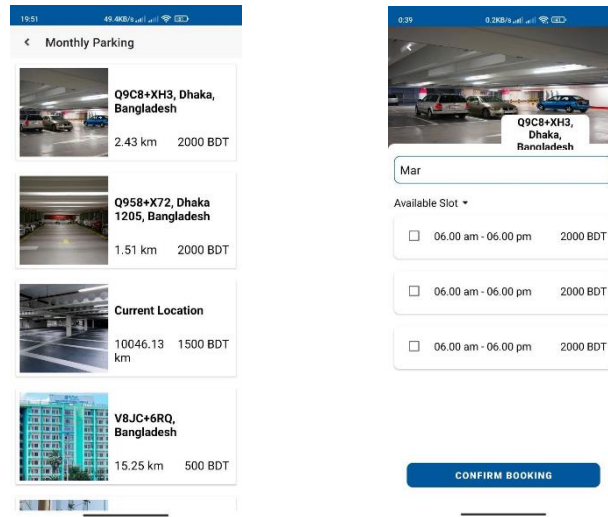


Figure 5.14: A Screenshot of the Monthly parking system

In figure 5.14 Users can see a list of parking spaces where users can choose and book a parking space for a month. In this slot, users can get an option where they can book a parking space for a month. After that, users can make payments for this in the same process as in figure 5.11.

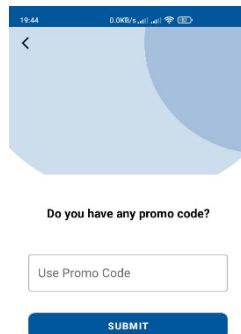


Figure 5.15: A Screenshot of the promo code using

In figure 5.15 Users can use a promo code to get discounts. Authority provides that promo code occasionally. Users can apply a promo code for once.

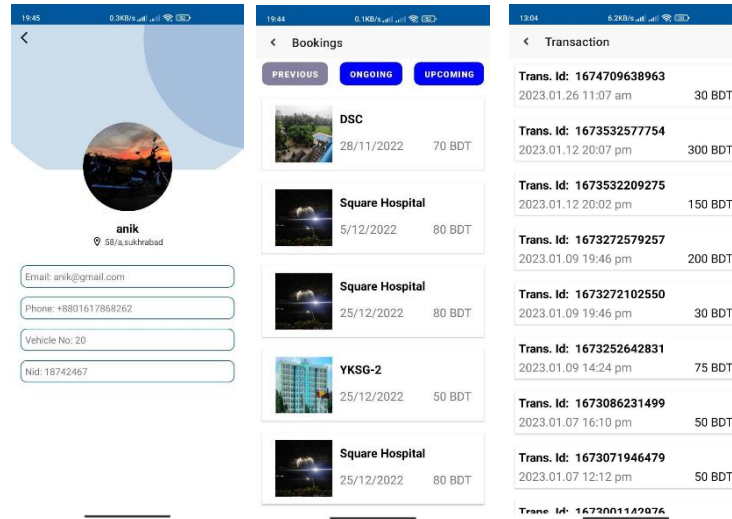


Figure 5.16: Screenshots of the User’s Profile, Booking, Transaction

Users can view their profile information, booking history, and transactions in figure 5.16. On the booking, page user gets his previous booking history, ongoing booking, and upcoming booking list. On the transaction page, user can view their payment history with transaction id, date, time, and cost.

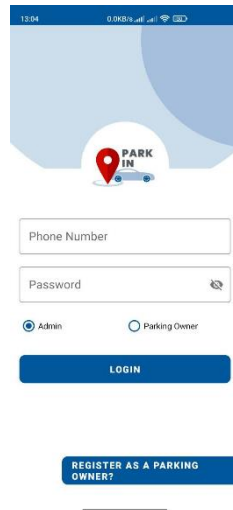


Figure 5.17: A Screenshot of the Admin/Parking Owner Login page

In figure 5.17 We have attached the screenshot of the Login page of our application with the logo, tagline a login button. If Admin or parking owner wants to enter our application and rent out their vacant parking spaces through our application, he needs to log in with a phone number and password. In our application parking owner and admin get the same user interface to add parking spaces.



Figure 5.18: A Screenshot of the Parking owner's Registration Page

In figure 5.18 A new parking owner can register by giving full name, gender, email address, address, phone number, and profile image. All information must have to valid otherwise authority can block this user.



Figure 5.19: A Screenshot of the Admin's/Parking Owner's Dashboard

In figure 5.19 Parking Owners can see their dashboards. Where they can able to see their total earnings by renting out their vacant parking spaces. And they can see their total parking space define by daily and monthly.

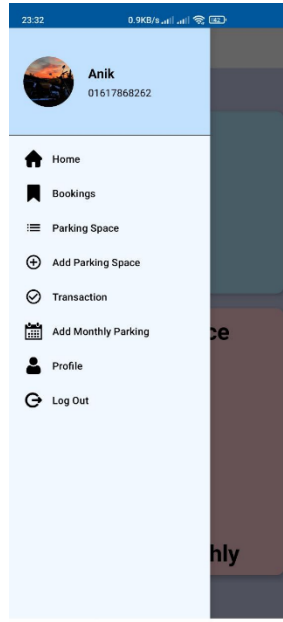


Figure 5.20: A Screenshot of the Admin's/Parking owner's dashboard

Using the navigation drawer, the Parking owner or admin can find “Bookings”, “Parking space”, “Add Monthly parking”, “Add parking space”, “Transaction”, “Profile”, and “Logout” options in figure 5.20.

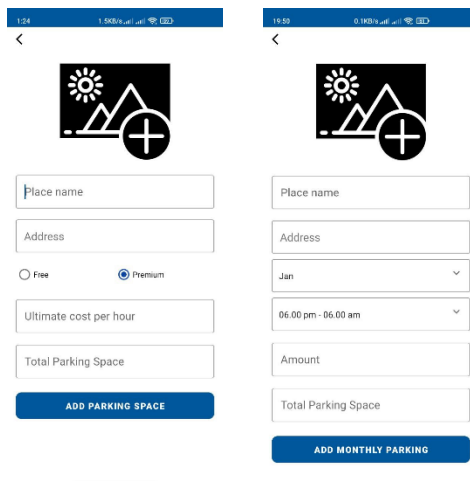


Figure 5.21: Screenshots of Add parking spaces

In figure 5.21 Admin and parking owner can add parking spaces for daily and monthly. They have to fill up the place name, address, date, time, month, amount, and total parking space field. Then they are able to earn money by renting out those spaces.

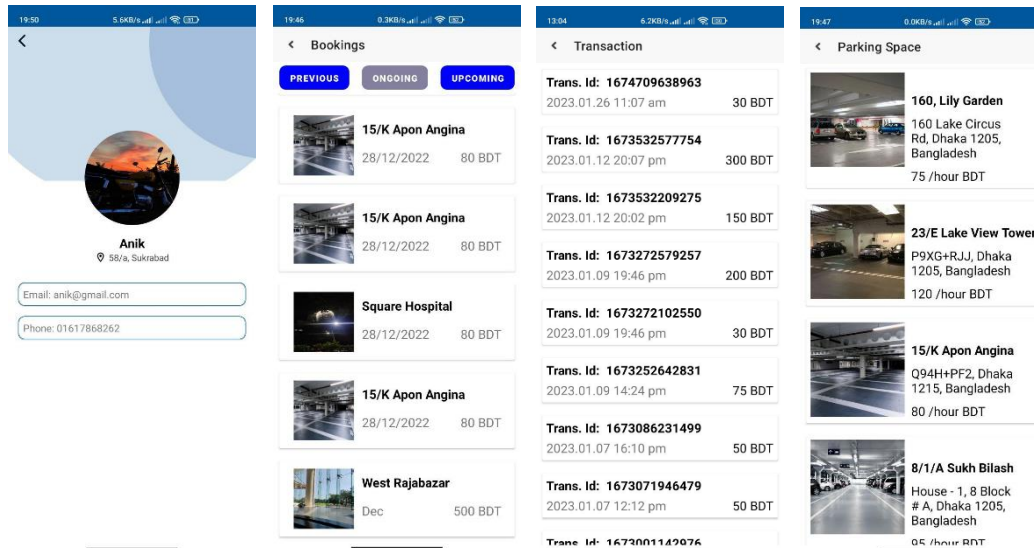


Figure 5.22: Screenshots of the profile, bookings, transactions, and parking spaces

Admin or parking owner can view their profile information, booking history, Parking space list, and transactions in figure 5.22. On the booking page, the parking owner gets his previous booking history, ongoing booking list, and upcoming booking list. On the transaction page, they can view their payment history with transaction id, date, time, and cost. And lastly, on the parking space page, garage owner can view their added parking space list with information.

5.3 Implementation of Interaction Design

Our application has a lot of interactivities. We created a welcoming user interface to provide users of this program with a richer user experience. I utilized an image view, a text button, and a recycler view to simplify things. The UI design of this program is user-friendly.

5.4 Testing Implementation

The tests for this project have been implemented. Testing is crucial for developing a complete, seamless, and error-free system. We tested it to find bugs, fix them, and make it easier to use. We'll come up with some unique requirements for this test.

Table 5.1: Testing Objective of our application

01	To check whether the program is running or not
02	To analyze register work
03	To analyze, whether the login is working or not
04	Check whether OTP sending successfully or not
05	Check Parking Owner can add parking space successfully or not
06	Check file was uploaded to firebase successfully or not
07	Explore the error of double action
08	Check Find Parking Space option is working or not
09	Check User can book a parking space or not
10	Check User can use the Promo code or not
11	Check Payment place successfully
12	Check Admin/Parking Owner/User can logout or not

5.5 Test Results and Reports

Table 5.2 displays the test results based on the test cases mentioned in the chapter's preceding section.

Table 5.2: Result of Test Objectives

01	To check whether the program is running or not	Success
02	To analyze register work	Success
03	To analyze, whether the login is working or not	Success
04	Check whether OTP sending successfully or not	Success
05	Check Parking Owner can add parking space successfully or not	Success
06	Check file was uploaded to firebase successfully or not	Success
07	Explore the error of double action	Success
08	Check Find Parking Space option is working or not	Success
09	Check User can book a parking space or not	Success
10	Check User can use the Promo code or not	Success
11	Check Payment place successfully	Success
12	Check Admin/Parking Owner/User can logout or not	Success

CHAPTER 6

Impact on society, environment, and sustainability

6.1 Impact on Society

Our application will make an important impact on society. Before doing any new work, it should be considered what kind of impact it will have on society. We have progressed with the impact on society while developing our application. Such as,

- Many unemployed people will get employment.
- The garage owner will benefit financially.
- Poverty level will be eradicated.
- People's precious time will be saved.
- Economic development of the country is possible through saving energy.
- Reducing noise pollution and air pollution will improve human health.
- Our system reduces stress while searching for a parking space.
- Increase in safety.

6.2 Impact on the Environment

We all know that climate change is not only a reality, it's an increasing threat to our lives as we know them. Our application will have some impact on the environment that is truly unimaginable. When you don't know where there's a parking space available, you drive around until you find one. According to calculations, cruising for a parking space adds somewhere between 3.5 and 14 minutes to a journey and that extra time means extra emissions. So, the fewer cars run, the less noise and air pollution, which helps balance the environment. We know that one of the main causes of climate change is carbon emissions. So, it will help us to control global warming by reducing carbon emissions. We all know that the amount of natural fuels is constantly decreasing on the earth. So, our system will prevent the wastage of fuel.

6.3 Ethical Aspects

We have designed our application from the very beginning with ethics in mind. Our app does not use anything that causes harm to the user or anyone around. If we come to ethics then false promotion comes first. Which will never be achieved by our app. Besides, if we need any access permission from the user, we will take it with the explanation. If we think beyond the app, people can escape this uncontrolled parking habit. Which will help to increase the moral values of people. Increasing usage of our app will increase the sense of responsibility and ethics among users.

6.4 Sustainability Plan

We have created this application given the problems that have arisen with Bangladesh parking. The users of our application are car owners and garage owners. So, the first condition of our sustainability plan is to provide better service to our users. Besides, improving the environment is also one of our essential tasks. Along with the garage owners we have to build our adequate parking system. So that in the future we can meet the parking needs of the users. Also, we need to update this system constantly according to the needs of the users. People spend an average of 15-20 minutes per day searching for parking in cities. Which causes traffic jams and unnecessary CO2 emissions. Through our application users reserve their parking so that they do not have to spend that additional time on the road. We will come up with new sustainability plans for our application to keep pace with the times.

CHAPTER 7

Conclusion

7.1 Conclusion

In reality, the majority of people in our country face several difficulties when they want to park their car. Because the parking facility in our country is still not so developed. As a result, traffic jam is created on the road. On the other hand, many house owners around us have empty parking garages. So, we hope our application will help those who have trouble parking their car. Our application will bring a hassle-free solution for them. Also, our application will play an important role in decongesting every city. On the other hand, parking lot owners will benefit financially. We made an all-in-one solution which can solve parking problems and traffic jams.

7.2 Limitation

1. After completing the booking user can't cancel the booking.
2. User and Parking owner can't do anything without the internet.

7.3 Future Scope

Our future goal is to implement this idea in our country and online to help those who want to park their car at their convenience and secure parking garage and those who want to rent out their vacant parking space. We want to expand not just our nation but also the world. We'll also add the rating system for each garage. So that the users get a complete idea about the garage. In the future, we want to give the user street navigation from the user's location to the garage. I'm working very hard to make using and administering this program easier. By using this application, we hope many car owners in our country will have no more worries about car parking and traffic congestion will also be reduced.

Reference:

[1] Wikipedia, “XML”, Available online: <https://en.wikipedia.org/wiki/XML>, Last access: August 2, 2022, 10.00 am.

[2] Wikipedia, “Scalable Vector Graphics”,

Available online: https://en.wikipedia.org/wiki/Scalable_Vector_Graphics, Last access: September 2, 2022, 10.10 am.

[3] Wikipedia, “Adobe Photoshop”,

Available online: https://en.wikipedia.org/wiki/Adobe_Photoshop, Last access: September 2, 2022, 10.20 am.

[5] Firebase, “Firebase Realtime Database”,

Available online: <https://firebase.google.com/docs/database/>, Last access: September 10, 2022, 8.30 pm.

[6] Wikipedia, “Android Studio”, Available online: https://en.wikipedia.org/wiki/Android_Studio, Last access: September 01, 2022, 9.00 am.

[7] Wikipedia, “Firebase - Realtime Database”,

Available online: https://en.wikipedia.org/wiki/Firebase#Realtime_Database, Last access:

September 12, 2022, 10.00 am.

[8] Learn about UI/UX design, from at <https://www.udacity.com>, Last access: July 5, 2022 at 10.30 pm.

[9] Learn about Firebase, from at <https://firebase.google.com/docs/>, Last access: July 10, 2022 at 8.30 pm.

[10] Firebase – Cloud Storage, Available online <https://firebase.google.com/docs/storage>, Last access: July 14, 2022 at 11.30 pm.

[11] Learn about Firestore Database, from at <https://www.udacity.com>, Last access: February 15, 2022 at 11.30 pm.

[12] Firebase – Authentication, from at <https://firebase.google.com/docs/auth>, Last access: September 15, 2022 at 11.30 pm.

[13] Learn about Figma, from at <https://www.figma.com/resources/learn-design/>, Last access: September 1, 2022 at 11.30 pm.

[14] Learn benefit of smart parking from at: <https://blog.getmyparking.com/2019/02/21/>, Last access: December 1, 2022 at 11.30 pm.

Ocean

ORIGINALITY REPORT

25%
SIMILARITY INDEX

19%
INTERNET SOURCES

1%
PUBLICATIONS

22%
STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	15%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	5%
3	blog.parkman.io Internet Source	1%
4	www.tibco.com Internet Source	1%
5	Submitted to University of Colombo Student Paper	1%
6	Submitted to Angeles University Foundation Student Paper	1%
7	Submitted to The University of Manchester Student Paper	<1%
8	Submitted to Tarlac Agricultural University Student Paper	<1%
9	Submitted to Green Hope High School Student Paper	<1%

10	Submitted to University of Greenwich Student Paper	<1 %
11	techrights.org Internet Source	<1 %
12	Submitted to Universiti Teknologi MARA Student Paper	<1 %
13	listenenglish.freevar.com Internet Source	<1 %

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off