Lincoln University College

Texas International College

A Final Year Project Report

On

Cleaner - Cleaning Service Booking Application

Under the Supervision of

Mr. Saroj Dhital

Lecturer

Texas International College

Submitted To:

Department of Information Technology Texas International College

In partial fulfillment of the requirement for the bachelor's degree in Information Technology

Submitted By:

Roshan Khatri

LC00017001252

September 27, 2023

SUPERVISOR'S RECOMMENDATION

I hereby recommend that this project report be prepared under my supervision by Roshan Khatri entitled **"Cleaner"** in partial fulfillment of the requirement for the degree of BIT (Bachelor in Information Technology) of Lincoln University College be processed for the evaluation.

……..

Mr. Saroj Dhital

Project Supervisor

Faculty

Texas International College

Chabahil, Kathmandu

LETTER OF APPROVAL

This is to certify that this project report prepared by Roshan Khatri in partial fulfillment of the requirement for the degree of BIT (Bachelor in Information Technology) has been well studied. In our opinion, it is satisfactory in the scope and quality as a project for the required degree.

…….. ……..

Mr. Saroj Dhital
Project Supervisor
Faculty
Texas International College
Shifal, Kathmandu

Mr. Suman Thapaliya
HOD, BIT
Texas International College
Shifal, Kathmandu

ACKNOWLEDGEMENT

I would like to express special thanks of gratitude to my supervisor Saroj Dhital as well as Texas College who gave me the golden opportunity to do this project on the topic of Cleaner. This also helped me in doing lots of research and I came to know so many new things.

Also, highly grateful to Texas College for guidance and mentorship. I appreciate the hard work and efforts of HOD Mr. Suman Thapaliya his suggestion plays an important role in my project. I got constant support from our college which helped me complete my project. So, I would like to thank all the technical and non-technical staff of the BIT department.

Last but not least I would like to thank everyone who helped and motivated me to work on this project. Also, I would like to show my gratitude toward all the faculty members, seniors, and colleagues for their direct or indirect support, especially Mr. Saroj Dhital for his great effort toward my project and documentation as well as for providing the necessary information. His constant guidance and willingness to share his vast knowledge made me understand this project and its manifestation.

I want to thank my family members, well-wishers, and seniors for being a part of the project and providing me with suggestions and feedback.

Sincerely,

Roshan Khatri

Abstract

In today's world, access to essential services like cleaning has become a critical need. The Cleaner application comes out as a transformative solution, redefining the way individuals and businesses interact with cleaning services. This project report shows the core features of Cleaner, an exceptionally user-friendly platform crafted to streamline the process of booking cleaning services.

Cleaner is designed for a small scale company with a primary focus on user convenience and operational efficiency. It empowers users to seamlessly book cleaning services, spanning from residential cleaning for homes to specialized solutions for businesses. The application simplifies the booking process, enabling users to effortlessly schedule cleaning services, selecting dates and times that align with their specific requirements.

By emphasizing the essential functions of service booking, Cleaner presents an intuitive and hassle-free approach to accessing cleaning services. Through this innovative solution, Cleaner aims to redefine the cleaning service landscape, making it more accessible and responsive to the evolving needs of modern society, ultimately enhancing the quality of life for its users.

Table Of Content

CHAPTER 1
INTRODUCTION

1.1 Introduction:

The Cleaner App serves as the ultimate solution to all cleaning service requirements. Its purpose is to offer a seamless and reliable platform for booking cleaning services. This application is developed considering an company which has experienced cleaning professionals. The company deals with the services booked in the application. This application serves as an medium for customers to book services and notify the company of the services booked and who booked them. Prioritizing convenience and high standards, the app suggests highly rated cleaning services who align with your personal preferences. The objective is to deliver cleanliness and excellent service directly to users' homes, enabling them to enjoy a clean and orderly space without any inconvenience.

1.2 Problem Definition

The traditional process of finding and booking cleaning services can be tedious and time-consuming for customers. Customers often struggle to find reliable and trustworthy cleaners. This project aims to address these problems. Existing options lack the method to book any cleaning service and the information on how well their professionals are trained or skilled.

1.3 Objectives

The objectives of developing Cleaner are:

- Simplify the service booking process for quick scheduling.
- Present a range of cleaning services options.

1.4 Scope and Limitation

The scopes of this project are as follows:

- The app will provide a range of cleaning service.
- The app will feature user-friendly interface to book cleaning services.

The limitations of this project are as follows:

- All cleaning service may not be available.
- Services may vary as per cleaning professional.
- May not be able to match all unique preference of cleaning.

1.5 Development Methodology:

The project will be following agile development methodology.The project will be divided into smaller tasks ,allowing for continuous integration.The iterative approach will enable to refine the system based on changing requirements.

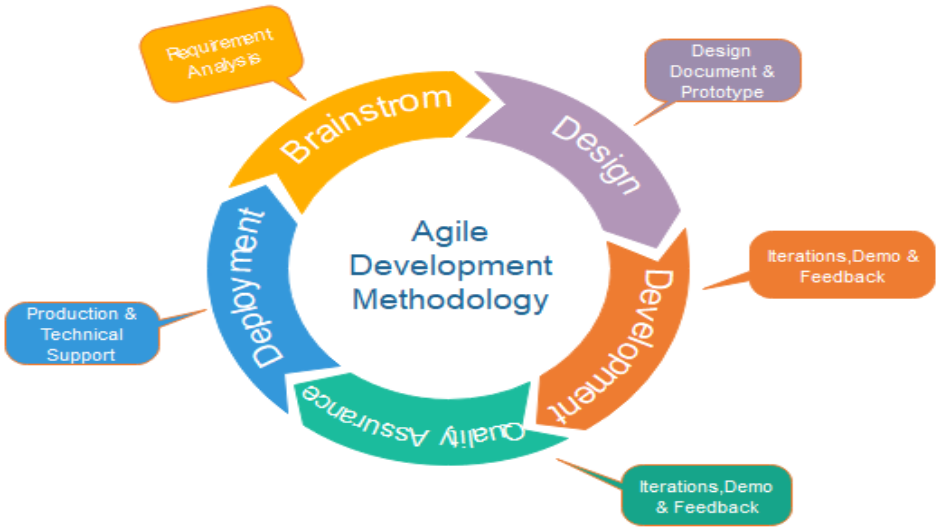


Fig. Agile Model

CHAPTER 2

BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Background Study

The background study reveals that the Cleaner App is grounded in:

I. **Market Demand:** The market lacks any online platform providing cleaning services booking system. People are seeking hassle-free ways to access trustworthy cleaning professionals. The rise of smartphone usage and app-based services has paved the way for innovative platforms like the Cleaner App to offer on-demand cleaning services with user-friendly interfaces.

II. **Service Diversity:** The app addresses varied cleaning requirements, from homes to businesses. Many places and people require different cleaning services which a untrained personnel cannot provide.

III. **Quality Assurance:** Checking and rating professionals to ensure consistent service quality. Prior experience indicates that consistent service quality is essential for user retention.

2.2 Literature Review

Online Cleaning Service (OCS) is a cleaning arrangement between a business and a property owner or representative made via the Internet through various channels such as web application, mobile application, social media messages, or etc. The customer/client experience usually starts from a client browsing for cleaning arrangements. After which, the client then uses the online platform of OCS to estimate for an instant quotation. [1]

ServeGo is an online service booking app offering a variety of services including electrical repair, plumbing, and home maintenance. It also serves as a platform for skilled professionals to showcase their expertise. The app connects users with service providers, addressing the challenge of finding reliable help quickly. ServeGo is developed using Android Studio and Firebase, with modules like notifications, requests, bookings, payments, ratings, and GPS tracking for a comprehensive user experience. This app

streamlines service access and empowers service providers, saving users valuable time. [2]

"Service Finder" is an Android app simplifying the search and booking of local services like plumbing and electrical help. It uses location-based services for proximity-based results. Developed with Android Studio and MySQL, the app consists of five modules: profile management, account handling, service search, appointment booking, and a rating/review feature. While most features work well, there might be some issues with the rating module. [3]

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

3.1 SYSTEM ANALYSIS

3.1.1 REQUIREMENT ANALYSIS

i. Functional Requirements

The system has the following functional requirements:

- Users can book cleaning services.
- They can specify the type of cleaning, date, and time.
- User can leave their contact and location details for visit.
- Dashboard is used of Firebase for database and admin control.
- Test all the modules whether it is working properly or not.

Use case diagram of system

In the figure 1, the interaction between user and admin is shown. The admin can login the Firebase console and view booked services and send receipts via email. They are responsible for managing location details for their staffs. Here, user can view the services available and book the services according to their needs, date and time and will receive the receipt of services via their email provided.

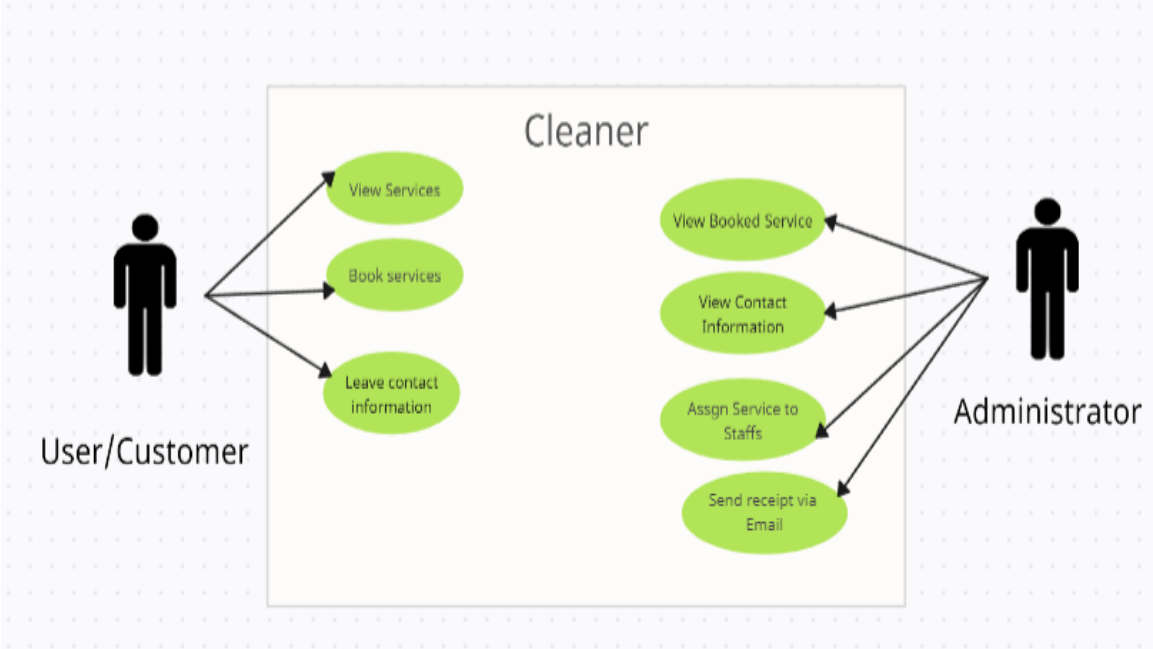


Figure 1: Use Case Diagram

ii. Non functional requirement

- The system should be compatible with different android devices.
- The application should be accessible by user at any time.
- The system should be maintained according to the requirement to increase the system performance and quality.
- The system should be user–friendly and it should be simple to understand

3.1.2 Feasibility Study

i. Technical feasibility

The system is developed and implemented for the small organizations that will hire cleaning professionals and cleaning equipment necessary for company operation and which will have minimum orders for daily basis.

ii. Operational feasibility

The system minimizes the cost of the software requirements. Only few developers are needed so, cost for the labor is minimized.

3.1.3 Gantt Chart

Figure 2: Gantt Chart

3.1.3 Data Modeling (ER Diagram)

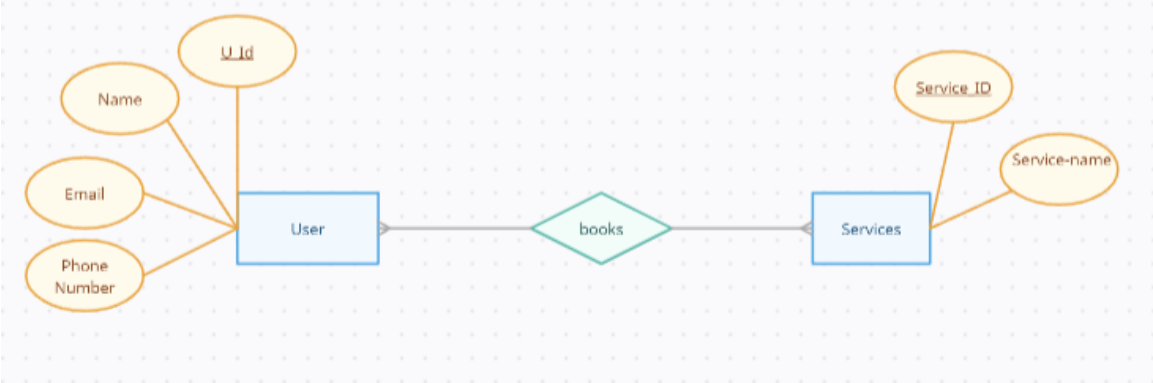


Figure 3: Data Modeling (ER Diagram)

3.1.3 Class Diagram

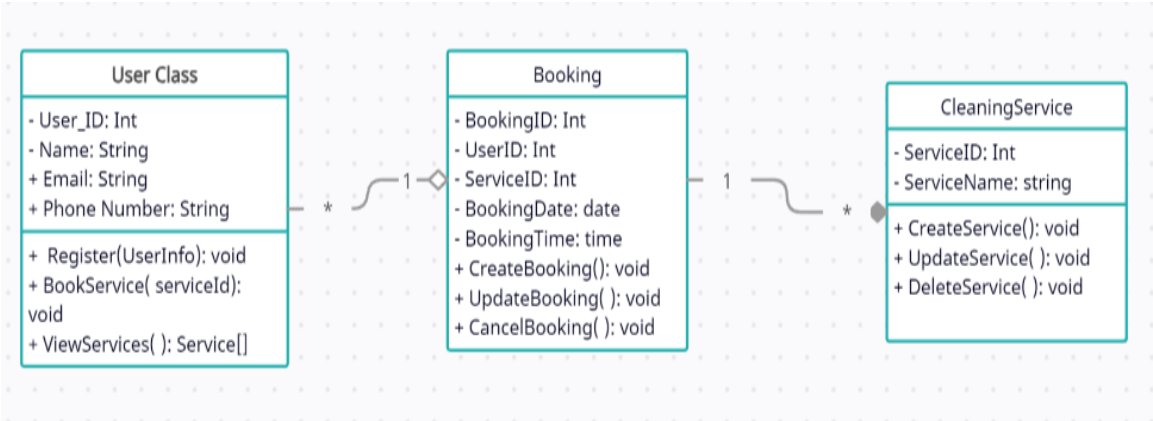


Figure 4: Class Diagram (UML diagram)

3.1.4 Activity Diagram

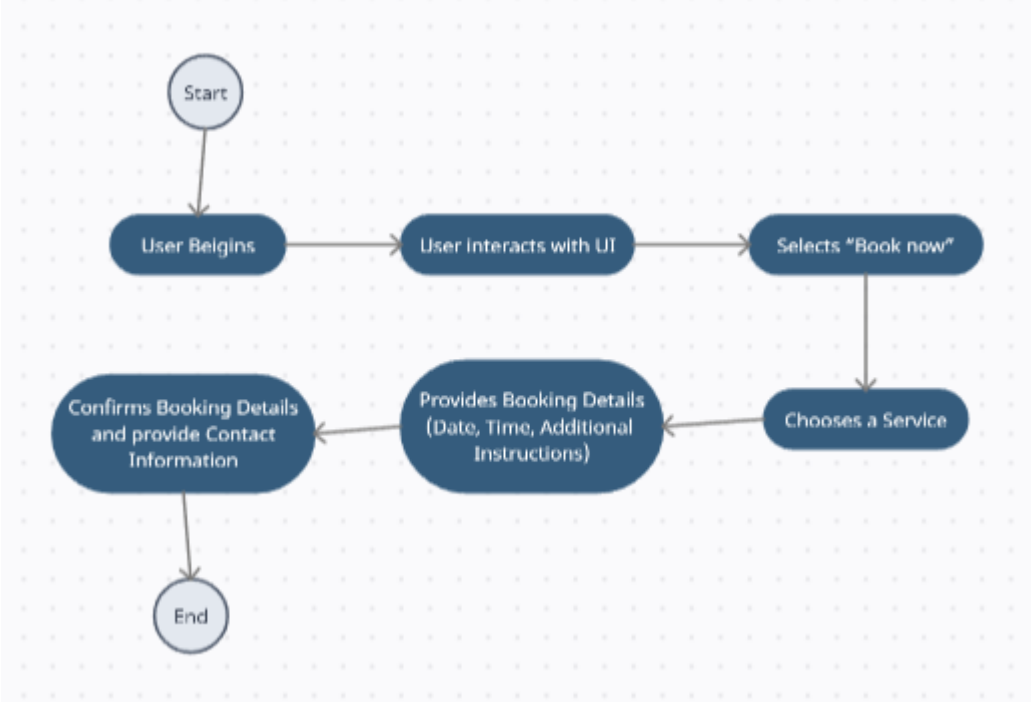


Figure 5: Activity Diagram

3.1.5 Data Modeling(DFD diagram)

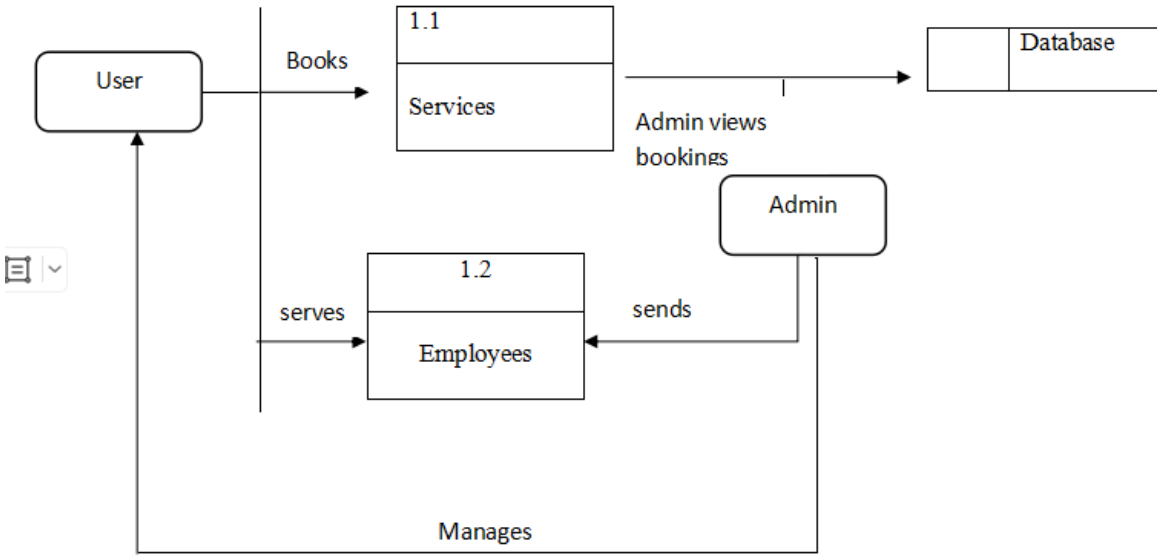


Figure 6: DFD Diagram

CHAPTER 4
IMPLEMENTATION AND TESTING

4.1. IMPLEMENTATION

- Identify entities: Users, Cleaning Services, and Bookings.
- Create database tables for each entity with attributes.
- Define relationships between entities.
- Choose a database system (Firebase Database).
- Design user-friendly app using a frontend framework(flutter dart).
- Perform thorough testing, including unit and user acceptance testing.
- Deploy the application to a production environment.
- Implement logging, error tracking, and ongoing maintenance.
- Optimize for scalability if needed (e.g., load balancing).
- Document the application's architecture and user guides.
- Launch the application to the public after testing and fine-tuning.

4.2 Tools Used

To implement the project, following tools are being used:

- Language: Dart 3.1.3
- Database: Firebase Database
- Frontend: Flutter 3.13.6 (to run: flutter pub get , flutter run)
- Backend: Firebase
- Text Editor: VS code
- Emulator: Android Studio- Giraffe 2023.1.1

4.3 Testing

4.3.1. Unit Testing

In this test, the software components will be combined to check the interaction among various models. The first module involves the proper functioning of the selecting the category of cleaning service. The second module involves the selection of the date and time of service implementation. The third module involves the input of contact information and location information of the user. The fourth module involve the company admin to monitor the booked service and contact information. The fifth module involves the testing of login and signup page.

Table 1: Testing of the Service Module

SN	Test Case	Input	Expected Output
1.	Selection of proper category of service necessary.	Initial Cleaning Daily Cleaning Deep Cleaning	User can select the type of cleaning service necessary.

Table 2: Testing of Date Time Module

SN	Test Case	Input	Expected Output
1.	Selection of appropriate date and time or the range of service period.	Date , Time , Bi-weekly, Weekly, Bi-Monthly, Monthly	User can select the appropriate date-time or the range of service period.

Table 3: Testing of the Contact Module

SN	Test Case	Input	Expected Output
1.	Users input the contact information and location for service.	Email, Phone Number, Location	Users can provide these information in the required field.

Table 4: Testing of the Admin Module

SN	Test Case	Input	Expected Output
1.	Manages the booked services by the admin through Firebase console.	Booked services, Contact Information	Admin can view the booked service and contact info necessary to deliver the service.

Table 5: Testing of the Login/Signup Module

SN	Test Case	Input	Expected Output
1.	Manages login of user with proper credentials.	Email, Password	User can login with their registered login email and password.
2.	Manages SignUp for new user.	Email, Password, Confirm Password	New user can set their email, password and confirm the password.

4.2.3. System Testing

This test is done to verify every component as a whole and each requirements is satisfied.

Table 6: System Testing of the Cleaner

SN	Test Case	Test Result
1.	Performance of the application using stable internet connection.	The application works totally fine with internet connection.
2.	Compatibility of the application shall be tested with the devices.	The app works properly in all android devices.

CHAPTER 5

CONCLUSION AND FUTURE RECOMMENDATION

5.1 Lesson Learnt and Outcome

The Cleaner application has demonstrated its effectiveness as a robust platform for managing cleaning services efficiently. Key insights and outcomes from the development and implementation of Cleaner include:

- Effective Task Management
- Progress Tracking
- Enhanced User Experience

5.2 CONCLUSION

In summary, the Cleaner application represents a significant advancement in simplifying the process of booking cleaning services for individuals and businesses. By focusing on key features like service exploration and booking, Cleaner offers an intuitive and user-friendly experience. It streamlines the search for cleaning services and makes scheduling convenient. With a commitment to user satisfaction, security, and scalability, Cleaner is poised to transform the cleaning service industry, making it more accessible and responsive to modern needs. As we move forward, our goal is to continually improve and expand Cleaner's capabilities, ensuring a top-notch cleaning service experience for all users.

5.3 Future Recommendations

Looking ahead, here are some key areas for potential enhancements:

- User Reviews and Ratings: Allows users to leave review and rating for services.
- Payment Integration: Integrate secure payment gateways for online payments.
- Login System: Create a system for login/Sign up.
- Develop app for IOS devices also.

REFERENCES

[1] [https://publisher.uthm.edu.my/periodicals/index.php/aitcs/article/view/479/288]

[2] [https://publisher.uthm.edu.my/periodicals/index.php/aitcs/article/view/7549/3853]

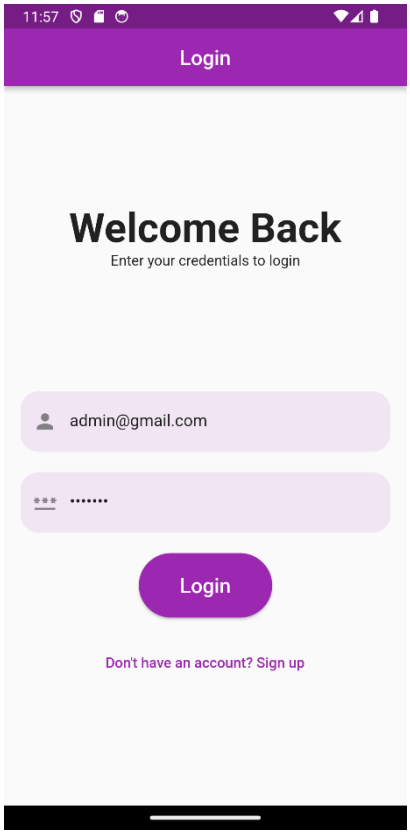
[3] [https://d1wqtxts1xzle7.cloudfront.net/63441338/TM_299_-_project_report_-_KING_-_2604202020200527-15250-1wzczy1-libre.pdf?1590581377=&response-content-disposition=inline%3B+filename%3DFuture_of_Philippine_Online_Residential.pdf&Expires=1695835288&Signature=bvQnKg7iRjOySxX94KvDKrsMwH1l-KhxulaDVF0Da5D3BLlUog5G4SkKOprouaSX~xcGV0~5-jCjcqntSTLhcR~zSf1zjZTk6ETvu6sDHZXmun4k7xexchklhzvVN4l4g7DSBeiAI8uJnHfqf~bht-vjeRA541RBFEum~m9NzP-FCICVyAoKACiRJSs9BOS6~CicweHhv~Kur42m9n2p74mGr4ZH1CpDwmxYklR148fv5Ru6cGyk4RPjHdeR6YFL3SUfJ9IMVm9bHVxrw3vj9ShZLOk-6I3X5rI4sYxylAknZZwDCyJ7uzVvC9xZz-b1sW2IiCeKgjSzU4TfdlLeGg__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA]

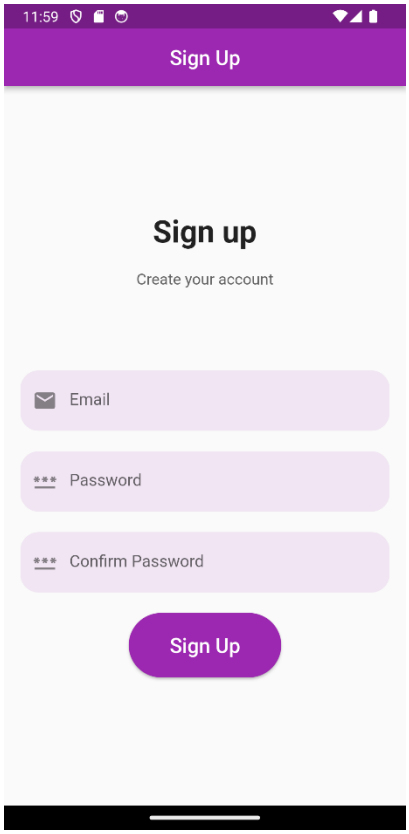
BIBLIOGRAPHY

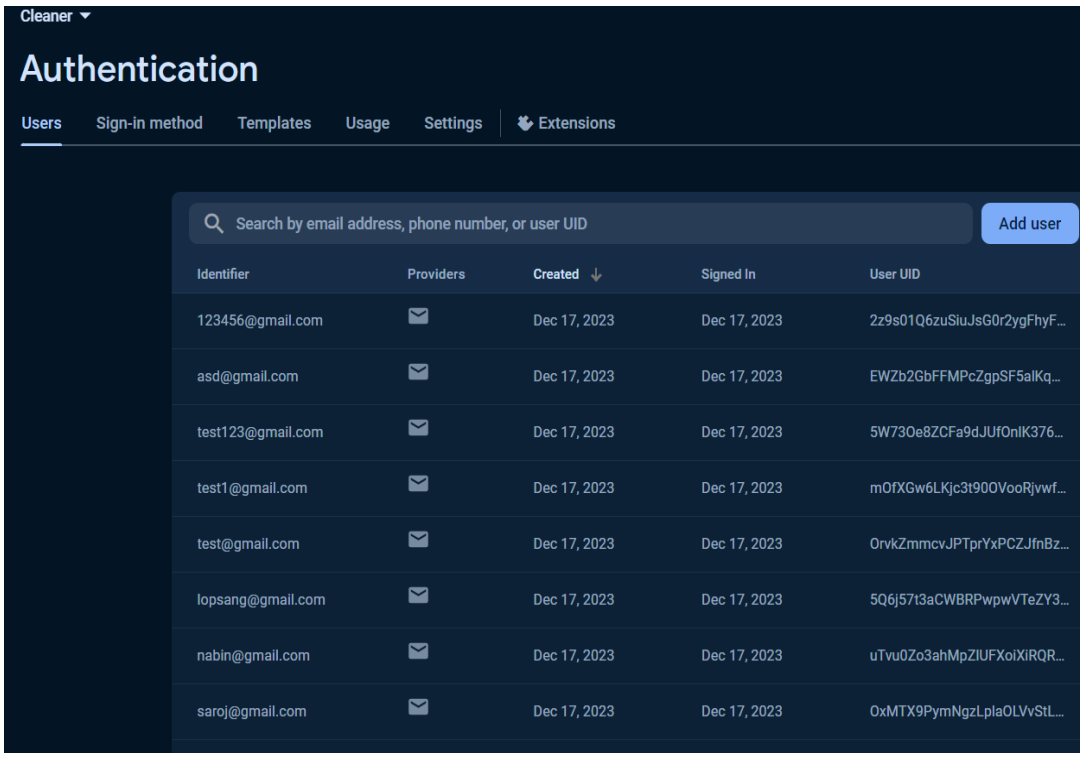
[1] *View of SERVICE ON THE GO (SERVEGO): Online service booking application*. (n.d.). https://publisher.uthm.edu.my/periodicals/index.php/aitcs/article/view/479/288

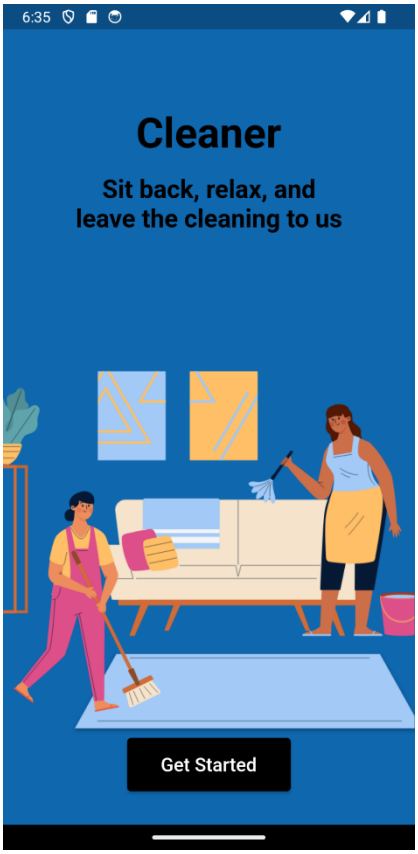
[2] *View of SERVICE FINDER: an Android-based service finding and booking application*. (n.d.). https://publisher.uthm.edu.my/periodicals/index.php/aitcs/article/view/7549/3853

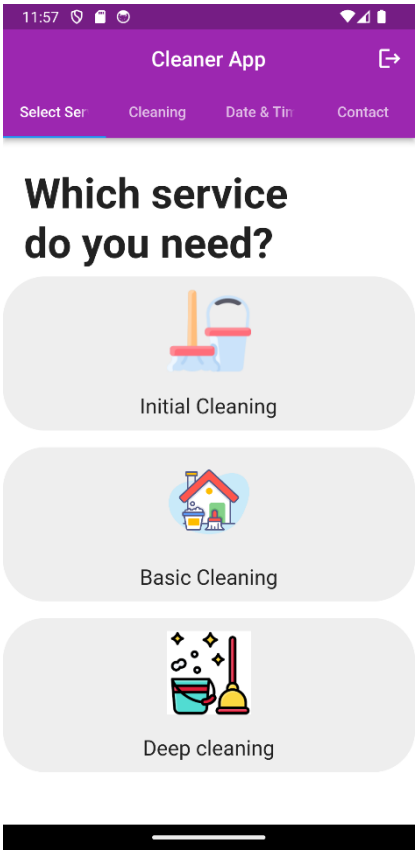
APPENDIX

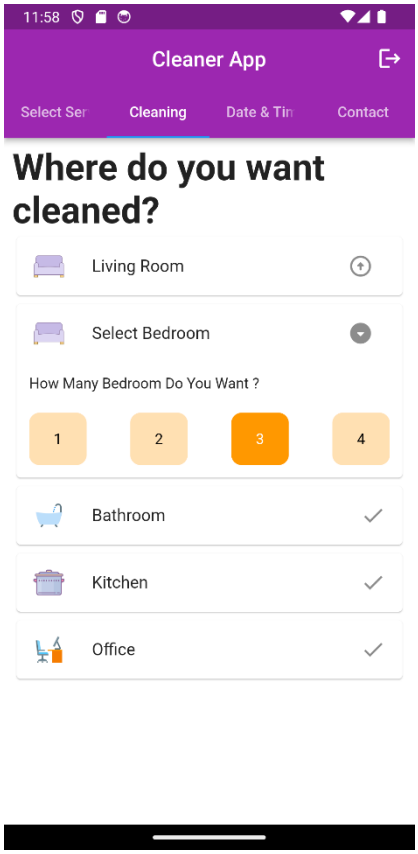

11:57
Login
Welcome Back
Enter your credentials to login
admin@gmail.com
Login
Don't have an account? Sign up

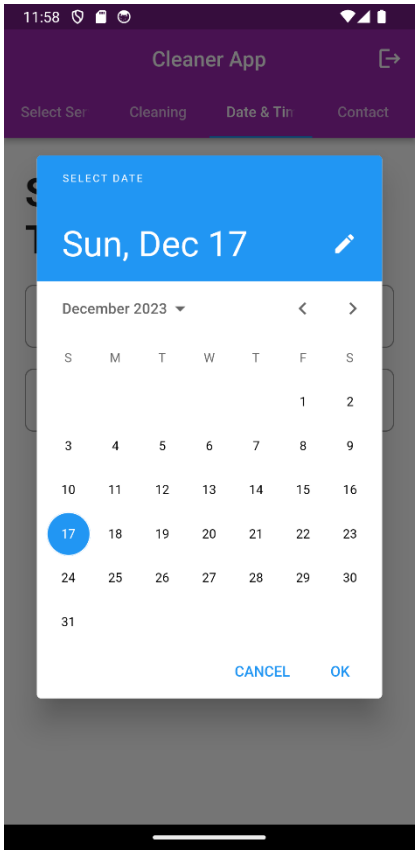

11:59
Sign Up
Sign up
Create your account
Email
Password
Confirm Password
Sign Up


Cleaner
Authentication
Users
Sign-in method
Templates
Usage
Settings
Extensions
Search by email address, phone number, or user UID
Add user
Identifier
Providers
Created
Signed In
User UID
123456@gmail.com
Dec 17, 2023
Dec 17, 2023
2z9s01Q6zuSiuJsG0r2ygFhyF...
asd@gmail.com
Dec 17, 2023
Dec 17, 2023
EWZb2GbFFMPcZgpSF5alKq...
test123@gmail.com
Dec 17, 2023
Dec 17, 2023
5W73Oe8ZCFa9dJUfOnlK376...
test1@gmail.com
Dec 17, 2023
Dec 17, 2023
mOfXGw6LKjc3t90OVooRjvwf...
test@gmail.com
Dec 17, 2023
Dec 17, 2023
OrvkZmmcvJPTprYxPCZJfnBz...
lopsang@gmail.com
Dec 17, 2023
Dec 17, 2023
5Q6j57t3aCWBRPwpwVTeZY3...
nabin@gmail.com
Dec 17, 2023
Dec 17, 2023
uTvu0Zo3ahMpZIUFXoiXiRQR...
saroj@gmail.com
Dec 17, 2023
Dec 17, 2023
OxMTX9PymNgzLplaOLVvStL...

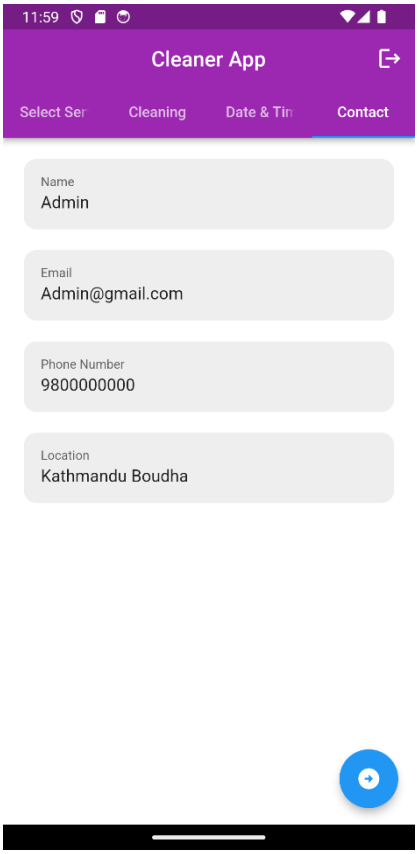

6:35
Cleaner
Sit back, relax, and
leave the cleaning to us
Get Started

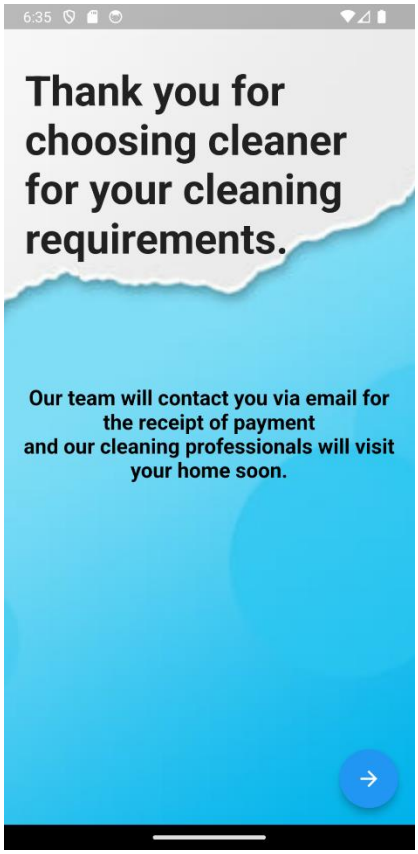

11:57
Cleaner App
Select Ser
Cleaning
Date & Tin
Contact
Which service
do you need?
Initial Cleaning
Basic Cleaning
Deep cleaning

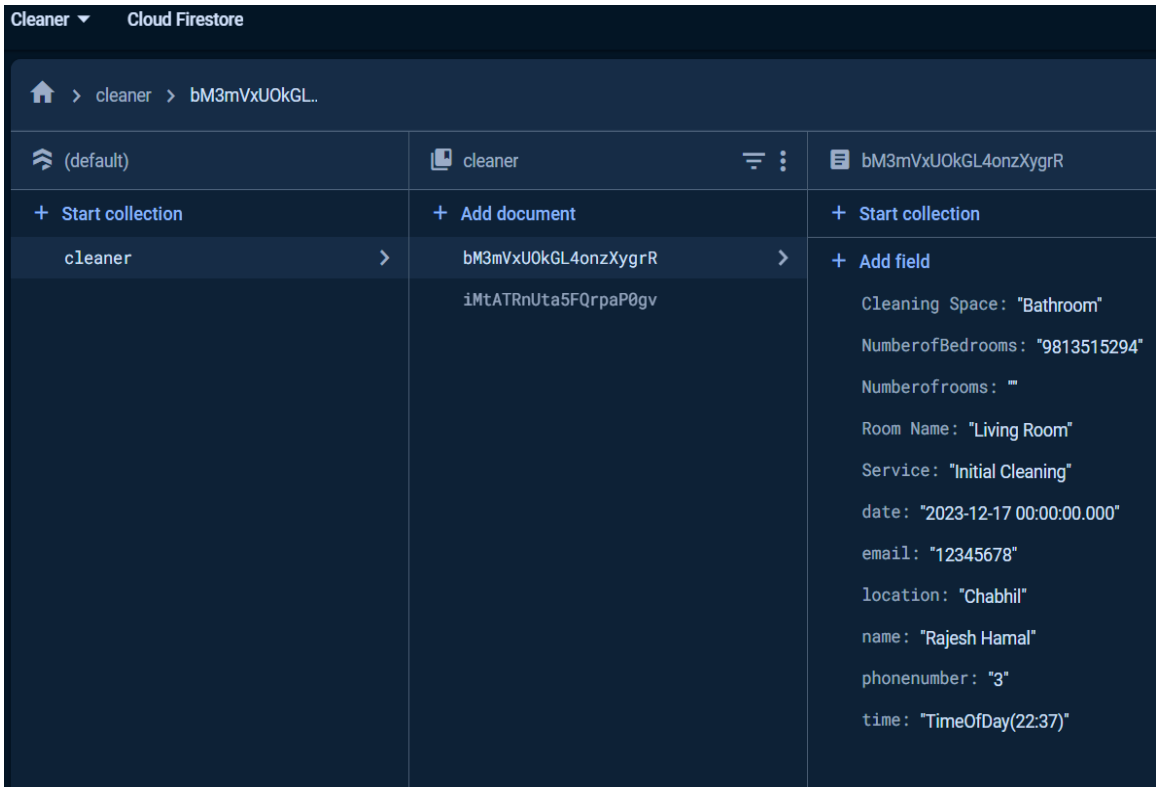

11:58
Cleaner App
Select Ser
Cleaning
Date & Tin
Contact
Where do you want
cleaned?
Living Room
Select Bedroom
How Many Bedroom Do You Want ?
1
2
3
4
Bathroom
Kitchen
Office

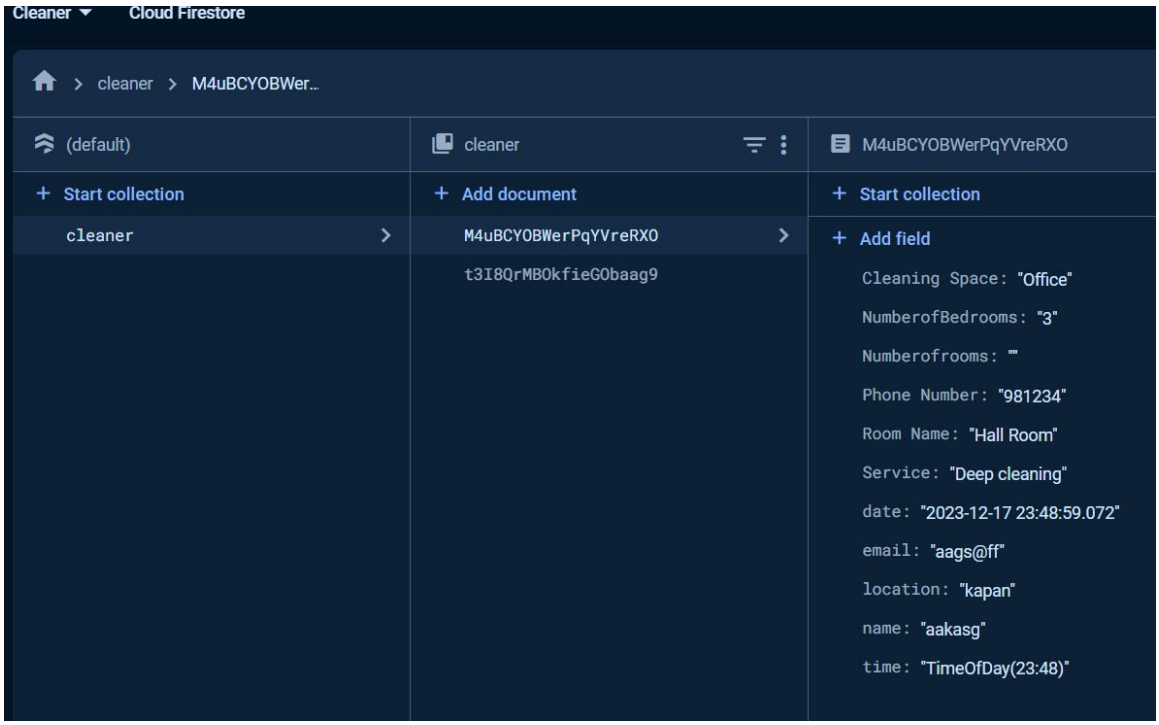

11:58
Cleaner App
Select Ser
Cleaning
Date & Tin
Contact
SELECT DATE
Sun, Dec 17
December 2023
S M T W T F S
1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
CANCEL
OK


11:58
Cleaner App
Select Ser
Cleaning
Date & Tin
Contact
SELECT TIME
11
58
AM
PM
12
1
2
3
4
5
6
7
8
9
10
11
CANCEL
OK


11:59
Cleaner App
Select Ser
Cleaning
Date & Tin
Contact
Name
Admin
Email
Admin@gmail.com
Phone Number
9800000000
Location
Kathmandu Boudha


6:35
Thank you for choosing cleaner for your cleaning requirements.
Our team will contact you via email for the receipt of payment and our cleaning professionals will visit your home soon.


Cleaner
Cloud Firestore
cleaner
bM3mVxUOkGL...
(default)
cleaner
bM3mVxUOkGL4onzXygrR
Start collection
Add document
Start collection
cleaner
bM3mVxUOkGL4onzXygrR
Add field
iMtATRnUta5FQrpaP0gv
Cleaning Space: "Bathroom"
NumberofBedrooms: "9813515294"
Numberofrooms: ""
Room Name: "Living Room"
Service: "Initial Cleaning"
date: "2023-12-17 00:00:00.000"
email: "12345678"
location: "Chabhil"
name: "Rajesh Hamal"
phonenumber: "3"
time: "TimeOfDay(22:37)"


Cleaner
Cloud Firestore
cleaner
M4uBCYOBWer...
(default)
cleaner
M4uBCYOBWerPqYVreRXO
Start collection
Add document
Start collection
cleaner
M4uBCYOBWerPqYVreRXO
Add field
t3I8QrMBOkfieGObaag9
Cleaning Space: "Office"
NumberofBedrooms: "3"
Numberofrooms: ""
Phone Number: "981234"
Room Name: "Hall Room"
Service: "Deep cleaning"
date: "2023-12-17 23:48:59.072"
email: "aags@ff"
location: "kapan"
name: "aakasg"
time: "TimeOfDay(23:48)"

main.dart

lib > main.dart > main

```dart
import 'package:cleaner/firebase_options.dart';
import 'package:cleaner/pages/login.dart';
import 'package:cleaner/provider/cleaning_request_provider.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';

Run | Debug | Profile
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  runApp(
    MultiProvider(
      providers: [
        ChangeNotifierProvider(create: (_) => CleaningRequestProvider()),
      ],
      child: MaterialApp(
        debugShowCheckedModeBanner: false,
        home: AuthenticationWrapper(),
      ), // MaterialApp
    ), // MultiProvider
  );
}
```

mytabs.dart 1

lib > mytabs.dart > ...

```dart
import 'package:cleaner/pages/cleaning.dart';

import 'package:cleaner/pages/contact.dart';
import 'package:cleaner/pages/date_time.dart';
import 'package:cleaner/pages/login.dart';
import 'package:cleaner/pages/select_service.dart';
import 'package:cleaner/pages/thank_you.dart';
import 'package:cleaner/provider/cleaning_request_provider.dart';
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';

class MyTabs extends StatefulWidget {
  @override
  _MyTabsState createState() => _MyTabsState();
}

class _MyTabsState extends State<MyTabs> with SingleTickerProviderStateMixin {
  late TabController _tabController;

  @override
  void initState() {
    super.initState();
    _tabController = TabController(length: 4, vsync: this);
    _tabController.index = 0;
```

mytabs.dart 1

lib > mytabs.dart > ...

```dart
@override
Widget build(BuildContext context) {
  return ChangeNotifierProvider(
    create: (context) => CleaningRequestProvider(),
    child: Consumer<CleaningRequestProvider>(
      builder: (context, value, child) {
        return Scaffold(
          appBar: AppBar(
            automaticallyImplyLeading: false,
            title: Text('Cleaner App'),
            centerTitle: true,
            backgroundColor: Colors.purple,
            bottom: TabBar(
              controller: _tabController,
              tabs: const [
                Tab(text: 'Select Service'),
                Tab(text: 'Cleaning'),
                Tab(text: 'Date & Time'),
                Tab(text: 'Contact'),
              ],
            ), // TabBar
            actions: [
              IconButton(
                icon: Icon(Icons.logout),
                onPressed: () async {
                  await FirebaseAuth.instance.signOut();
                  Navigator.pushReplacement(
                    context,
                    MaterialPageRoute(builder: (context) => LoginPage()),
                  );
                },
              ), // IconButton
```

mytabs.dart 1 | cleaningmodel.dart

lib > models > cleaningmodel.dart > Cleaningmodel

```dart
class Cleaningmodel {
  String servicename;
  String roomname;
  String numberofroom;
  String numberofbedroom;
  String other;
  String name;
  String date;
  String time;
  String email;
  String phonenumber;
  String location;

  Cleaningmodel({
    required this.numberofbedroom,
    required this.servicename,
    required this.roomname,
    required this.numberofroom,
    required this.name,
    required this.date,
    required this.other,
    required this.email,
    required this.location,
    required this.phonenumber,
    required this.time,
  });
  tojson() {
    return {
      'Service': servicename,
      'Room Name': roomname,
      'Cleaning Space': other,
      'name': name,
      'date': date,
      'location': location,
```

cleaning_request_provider.dart 1

lib > provider > cleaning_request_provider.dart > ...

```dart
import 'dart:math';

import 'package:cleaner/models/cleaningmodel.dart';
import 'package:cleaner/models/cleaningrequest.dart';
import 'package:cleaner/models/service.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';

class CleaningRequestProvider with ChangeNotifier {
  CleaningRequest cleaningRequest = CleaningRequest();
  TextEditingController nameofroom = TextEditingController();
  TextEditingController numberofroom = TextEditingController();
  TextEditingController kitchen = TextEditingController();
  TextEditingController office = TextEditingController();
  TextEditingController numberofbedroom = TextEditingController();
  TextEditingController expensescontroller = TextEditingController();
  TextEditingController nameController = TextEditingController();
  TextEditingController emailController = TextEditingController();
  TextEditingController phoneController = TextEditingController();
  TextEditingController locationController = TextEditingController();
  TextEditingController datecontroller = TextEditingController();
  TextEditingController timecontroller = TextEditingController();
  TextEditingController servicecontroller = TextEditingController();
  String _serviceName = '';

  String getServiceName() => _serviceName;
  bool choosenumber = false;
  bool selectedroom = false;
  bool selected = false;
  bool selectbedroom = false;
  bool selectedbedroom = false;
  List<bool> selectedrooms = [
```

cleaning_request_provider.dart 1

lib > provider > cleaning_request_provider.dart > ...

```dart
  String getServiceName() => _serviceName;
  bool choosenumber = false;
  bool selectedroom = false;
  bool selected = false;
  bool selectbedroom = false;
  bool selectedbedroom = false;
  List<bool> selectedrooms = [
    true,
    false,
    false,
    false,
  ];
  List<bool> selectedbeds = [
    true,
    false,
    false,
    false,
  ];
  List<bool> checked = [true, false, false];
  String select = 'Select Room';
  String bedroom = 'Select Bedroom';

  List<String> rooms = [
    '1',
    '2',
    '3',
    '4',
  ];
  List<String> bedrooms = [
    '1',
    '2',
    '3',
    '4',
```