SHAHEED ZULFIKAR ALI BHUTTO
INSTITUTE OF SCIENCE AND TECHNOLOGY

Final Year Project Report

UTechA

Project Team:
Malik Muhammad Basim Mansoor
1812156
Hasnain Haider
1812151

20th July, 2022

Project Supervisor:
Dr. Faraz Junejo

Submitted in the partial fulfillment of the requirements for the degree of

Bachelor of Science in Computer Science in
the

Faculty of Computing and Engineering Sciences

Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology (SZABIST)
Karachi Campus

Declaration of Authorship

We, Malik Muhammad Basim Mansoor (1812156) and Hasnain Haider (1812151), declare that this report "UTechA" and the work presented is our own. We confirm that: This is solely for the purpose of the completion of our bachelor's degree at Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology. This report has not been submitted to any university before this and where use of external text exists, that has been clearly stated. The source of references has been given where applicable. With the exception of such quoted texts, this report contains text from our own work. Where the report is based on work done by us jointly with others, we have clearly stated who has contributed to what area of the report.

Signed:

Malik Muhammad Basim Mansoor (1812156)

Hasnain Haider (1812151)

Date: 20th July '2022

Project Description

UtechA is an app that aims to connect small technical business owners to potential customers within their local operating areas. The app works similar to FoodPanda by using your location to show local technicians in your area; Connecting you to them and allowing you to book, rate, describe, discuss, and compare any technical services a customer might need.

Our aim is to provide a platform for both technical service providers and customers where both parties can communicate with and find each other without much hassle and hindrance. This App aims to facilitate users in finding the right man for the right job by allowing them to review customer feedback and book appointments, all without needing to set a single step outside the house.

Acknowledgement

In the name of ALLAH, the most beneficent and merciful, who gave us the knowledge and courage to work on this project.

We are grateful for the outcome and success of this project over the year are gratitude towards the people who have provided us with the guidance and assistance to be able to complete this project in such a difficult time.

We would like to thank our supervisor “Dr. Faraz Junejo” of the Mechatronics Engineering faculty at Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology. He was integral part in the project as he was always there to provide relevant feedback and motivate us with fresh ideas on how to proceed with our project as a product. His immense help in pinpointing usability and marketability of the application was detrimental to our success in the panel stages.

We would like to thank to the teachers at Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology, who guided us and taught us throughout our time in the university. We would also like to express our gratitude to our parents and family members who helped and encouraged us during this time. Furthermore, we would like to thank the staff at SZABIST for allowing us to use their labs and services to be able to complete the project.

Lastly, we would like to extend our gratitude to everyone at Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology for creating and environment for students to thrive in. The quality of education, the cooperative faculty members and the motivation provided by them.

Table of Contents

PROPOSAL

1. Introduction

UtechA is an app that aims to connect small technical business owners to potential customers within their local operating areas. The app works similar to FoodPanda by using your location to show local technicians in your area; Connecting you to them and allowing you to book, rate, describe, discuss, and compare any technical services a customer might need.

2. Objective

To provide a platform for both technical service providers and customers where both parties can communicate with and find each other without much hassle and hindrance. This App aims to facilitate users in finding the right man for the right job by allowing them to review customer feedback and book appointments, all without needing to set a single step outside the house.

3. Problem Description

Whenever technical work is required, it is quite difficult to find a technician without any hassle or financial drawbacks. Very few people have someone that they can trust and just the task of going out and seeking out technicians in the local markets can be quite difficult, most of the time people find themselves running around trying to find the right person for the right job all while comparing different prices, keeping track of material costs, labor costs and required material amounts and judging whether or not the person you have employed is capable enough to do the task you require. Once you do painstakingly find someone who seems right for the job, they might argue on payment and cost of materials only for you to find out afterwards that they didn’t do a substantial job because they resorted to cutting corners to minimize cost on their end. Our app aims to solve all of these problems by allowing users to browse services from within the comfort of their homes while featuring a rating system that will help users towards judging which service providers are considered to be the best ones by their local community.

4. Methodology

The app would feature 2 user sides, one for the service providers (Sellers) and another for the customers (Buyers). The Seller side would allow you to register along with details regarding the services that you provide. The Buyer side will be able to browse through different service providers that are registered and will be able to review their details and ratings before setting up an appointment with them. The technician would then visit the customer's home start work based on whatever terms discussed between both the parties via the app's communication feature until eventually asking the customer the rate/review the service once the work is done so that the rating may aid other potential customers in finding the right person for the job. We intent to integrate some sort of geo-location API such as Google maps to locate and connect buyers and sellers.

5. Project Scope

Although we have no current plans to monetize or market this application on a large scale, we are looking to bring it to a state that can affectively and accurately show off its capabilities in a way that makes all of its functions very clear to its users and testers. For the sake of this FYP and considering the time limit, our goal is not to commercially deploy this application but rather present a stable product that shows enough promise to hold potential for large scale deployment if ever considered for the future.

6. Feasibility Study

We should be able to meet our deadline as both team members are familiar with App development and understand the requirements of the project fairly well. That being said however, there are still some risks and uncertainties that we could run into:

i. While we are familiar with app development in general, more specific functionalities such as integrating locations and establishing a communication channel between two users is something that neither of the team members have firsthand experience with and so will have to learn and adapt to as we go along with the development process.

ii. Resource Requirements:

- Android/iOS Device
- Laptop or Desktop PC
- Stable Internet access

7. Solution Application Areas

This application aims to solve the gap between small business owners with technical expertise and customers who require their services. Most current apps on the market employ their own technicians and local small business owners are losing potential customers by lagging behind the digital age. This application and model aim to benefit both the customer; by giving them more options and security for their required work; And the seller/small business owner by connecting them to potential customers across their operating areas.

8. Tools/Technology

- VS Code text editor enabled with React Native

- Android/iOS supported mobile devices
- A third-party tool for UI design might be employed (to be decided)
- Google Maps API

9. Expertise of the Team Members

Both team members have a good grasp on the knowledge they need to complete this project including logic building, Android App development and Software Engineering however we do require courses on Web technologies and Web engineering to make two communicate with each other and store the data. Both members made sure to also enroll in the courses that could potentially aid us.

This project requires both team members to work on each component in parallel instead of building components separately and compiling them.

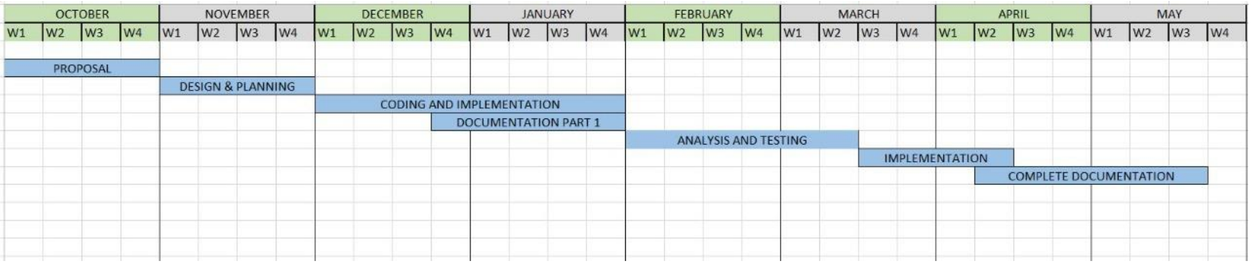
10.Milestones

1. Designing and planning (3 to 4 weeks)
2. Begin implementation and basic coding of the project while simultaneously working on the documentation (6 to 8 weeks)
3. Analyze and test the code for any potential issues/bugs etc. to make sure its ready (5 to 6 weeks)
4. Finally start implementation of the project once it’s in a presentable and stable state (3 to 4 weeks)

5. Actor/Goal List:

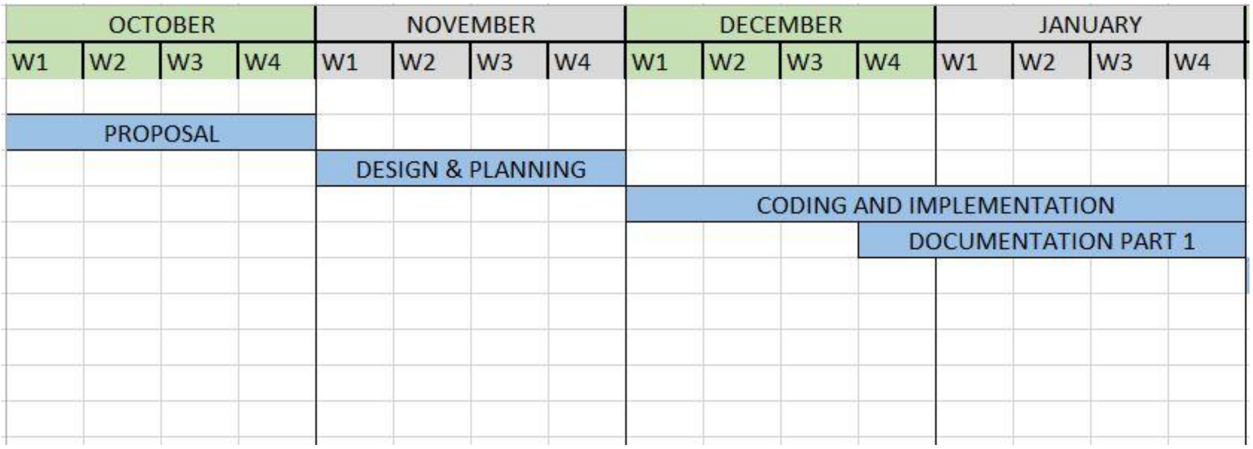
ACTOR	GOAL
CUSTOMER	• Create posting • Select vendor • Compare prices • Rate vendors • Book vendor
SELLER	• Receive posting • Send estimate • Accept/Reject booking • View requirements

11.Project Schedule

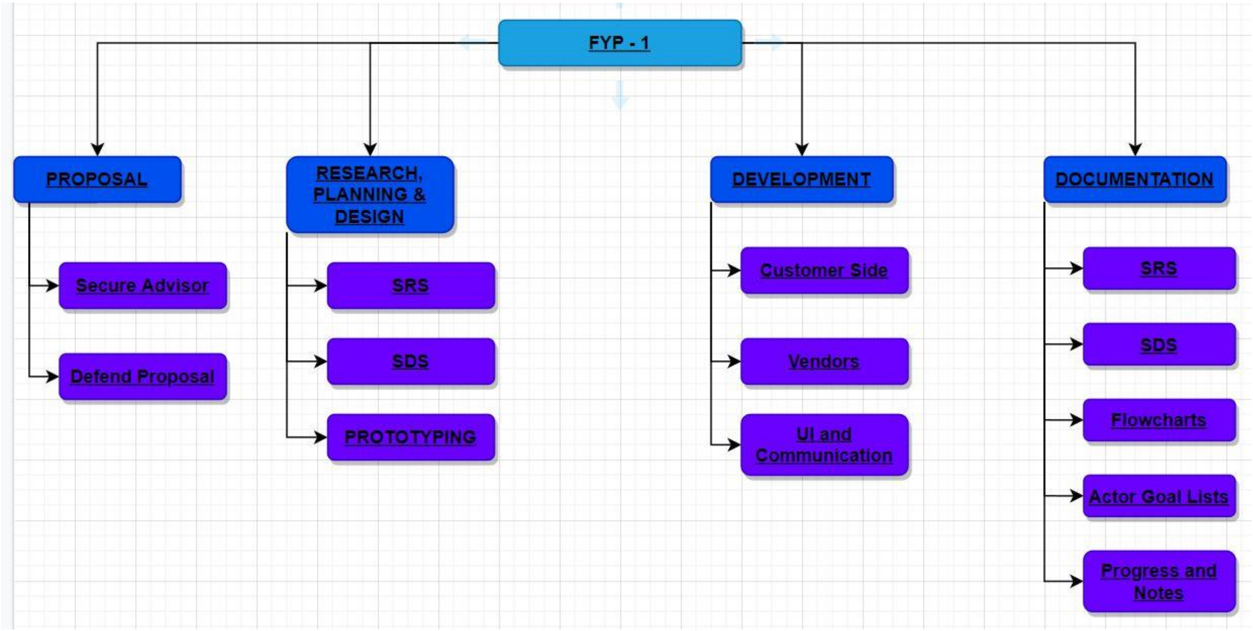


12.Work Breakdown Structure

Final Year Project – 1:



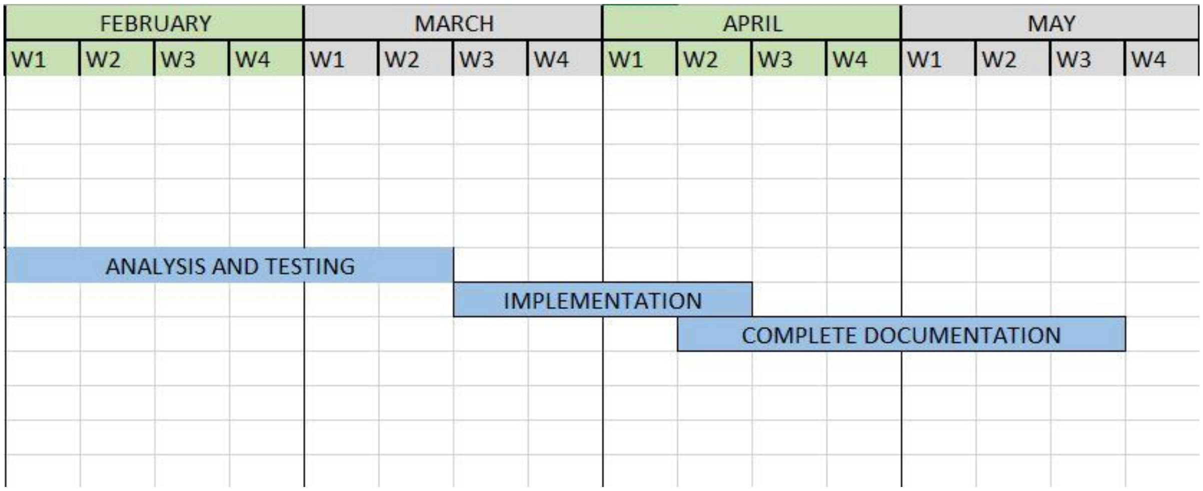
OCTOBER				NOVEMBER				DECEMBER				JANUARY			
W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
PROPOSAL															
				DESIGN & PLANNING											
								CODING AND IMPLEMENTATION							
											DOCUMENTATION PART 1				

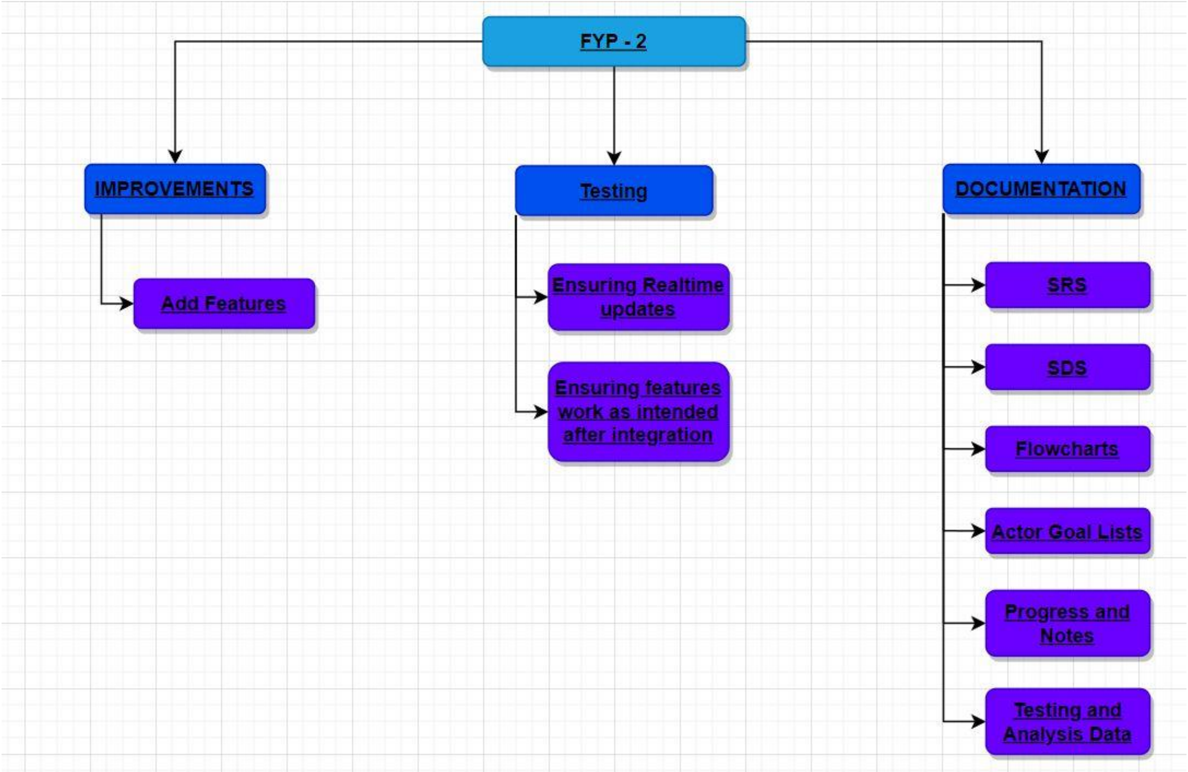


Feature-set for FYP-1:

- **Login and splash screen**
- **Basic UI design**
- **Customer and skilled worker portal**
- **Technician selection page**
- **General skeleton prototype and framework**

Final Year Project – 2:




FYP - 2
IMPROVEMENTS
Add Features
Testing
Ensuring Realtime updates
Ensuring features work as intended after integration
DOCUMENTATION
SRS
SDS
Flowcharts
Actor Goal Lists
Progress and Notes
Testing and Analysis Data

Feature-set for FYP-2:

- Booking feature
- Geo-Location
- Adding photos
- Contact
- Estimates

Software Requirement Specification

1. Introduction

1.1 Purpose

This SRS is for the documentation of the Final Year Project of Bs (Computer Science) students Malik Muhammad Basim Mansoor and Hasnain Haider of SZABIST Karachi under the supervision of Dr. Faraz Junejo. This document provides readers with information regarding the design and implementation of the Project UtechA which is a Utility Technical Application. This Document describes multiple subsystems of the entire application including buyer and seller portals.

1.2 Document Conventions

The standard 1999 IEEE SRS convention by Karl E. Wiegers was followed to compile the documentation of this project. Every requirement is to have its own priority which is set with 3 key levels, essential, important and desirable. Essential describes those features that the application is made for and cannot run without. Important is for those features that are needed for the project to be delivered as intended by the developer. Desirable is for those features that would be nice to have in the system but can be left out with no effect to other components.

1.3 Intended Audience and Reading Suggestions

Developers, testers, investment firms, evaluation experts and project managers are some of the people who might find this document helpful. This document contains the requirements, both hardware and software, the platforms used and any issues or difficulties faced during the development of this project.

1.4 Product Scope

The purpose of this project is to provide a platform for customers to connect to small technical business owners such as plumbers' electricians etc. within their local area. The system is meant to provide customers with information about the technicians including their rating and schedule of availability.

The system will deal with both customers and technicians on different views depending on who they login as. An interface will be provided for customers to select different types of technicians that they require. Another interface will allow them to locate and select technicians on a map in their local area.

In order to launch the project as a business, a business model similar to that of PayPal will be required with incentives set in place for both buyers and sellers to attract them to the platform. The usability of the project and features must be enough to keep the client base after the incentive program wares off.

2. Overall Description

2.1 Product Perspective

This system is an independent, self-contained product.

2.2 Product Functions

- Different home pages for Customers and Technicians

For Seller:

  - Give Estimate
  - Accept/Reject Job
  - Call Customer

For Buyer:

  - Technician Type Selection Page
  - Technician Selection Map
  - View Technician Rating
  - Post Requirements
  - Estimate
  - Schedule Appointment

2.3 User Classes and Characteristics

Demographics anticipated to use this product include smartphone users and homeowners between the ages of 20 to 60 to be the primary customer demographic while making orders and describing issues. We assume this demographic to have internet connections and have authority over home renovations/fixes. The 2nd demographic we expect to integrate into the product is technicians and small technical business owners. We expect them to have basic communication skills and be able to read and write with the English alphabet and be able to use a smartphone.

2.4 Operating Environment

The operating environment we chose is Android as it is open source and has enough documentation to smoothly integrate our needs. We have picked ReactNative as our development environment and RNFireBase as our database.

2.5 Design and Implementation Constraints

The UI and User Experience is the most important part of this application as it needs to be enough to keep the customer base going after the business model of incentive-based recruitment runs out. The biggest constraint we face is time as this project is part of a timed degree requirement which allows us 8 months to deliver and deploy the project in its most basic operational form. There are no regulatory policies as of yet as the project has not been picked up for widespread deployment as a business model.

2.6 Assumptions and Dependencies

Some assumptions that are necessary for the project to launch have more to do with technicians such as; we assume the technicians to be able to:

- Use a smartphone
- Text using the English alphabet
- Converse using local cellular services
- Have a schedule decided for work
- Be able to communicate with some efficiency

The only dependency we have identified as of yet for the project is Map integration which will most likely be done through Google maps API, however this could be subject to change.

3. External Interface Requirements

3.1 User Interfaces

The main buyer page has a technician selection page which is focused on user experience and aesthetics, when the user selects a technician that tile redirects to a map on which the technician can select the desired technician.

3.2 Hardware Interfaces

Smartphones that run on Android OS are our main hardware interface.

3.3 Software Interfaces

- ReactNative
- RNFireBase
- Cloud FireStore
- Google Maps API
- Nativebase
- Other Google Cloud Services

3.4 Communications Interfaces

Yet to be decided. (FYP-II)

4. System Features

4.1 Login

4.1.1 Description and Priority

Used to Login based off username and password. User will be authenticated through the database and will have access to the application if entered credentials are valid and authorized.

Priority: Essential

4.1.2 Stimulus/Response Sequences

User will enter correct username and password.

System will redirect User Type Selection page

User will enter incorrect username and password.

System will prompt an error message.

4.1.3 Functional Requirements

System should be able to verify and authenticate users based off the credentials they input on the login view. If credentials are valid, user should be redirected to User type selection page. If entered credentials are incorrect, an error message should pop up informing the user of invalid credentials.

4.2 Sign Up Page

4.2.1 Description and Priority

Used to Create accounts used for logging into the application. The user enter their details which will be used to authenticate them into the application and database including their email password etc.

Priority: Essential

4.2.2 Stimulus/Response Sequences

User will enter personal details: Name, Email, Password, Contact Number.

System will register user and push the new user data into the database and authorize the new user.

4.2.3 Functional Requirements

System should be able to take details the user has input and push them into the database. System should automatically authenticate, authorize and activate the user account.

4.3 User Type Selection Page

4.3.1 Description and Priority

First page that is shown after logging into the application. This page allows user to select either select seller or buyer services

Priority: Essential

4.3.2 Stimulus/Response Sequences

User selects “Looking for a Service”

System redirects them to technician selection page

User selects “Providing a service”

If user is an authorized seller:

Redirect to seller page

If user is not authorized as a seller:

Redirect to Technician Registration Page

4.3.3 Functional Requirements

System should be able to recognize what selection the user has made and redirect them to their desired destination. If user selects seller services, system should be able to verify if the user is an authorized seller or needs to register.

4.4 Technician Selection

4.4.1 Description and Priority

Used to select desired technician.

Priority: Essential

4.4.2 Stimulus/Response Sequences

User will click on desired technician tile.

System will redirect to map selection view, for example, user selects plumber, system will redirect to map selection page with plumbers in their local area.

4.4.3 Functional Requirements

System should be able to recognize and reflect the user’s selection by redirecting them to their desired destination.

4.5 Job Post page

4.5.1 Description and Priority

Used to describe the job requirements to the technicians

Priority: Essential

4.5.2 Stimulus/Response Sequences

User will enter job details such as issue, location, extent of work requires, picture etc.

User will press post job button.

System will send job details to technician that was selected

User will be redirected to home page after job post confirmation.

4.5.3 Functional Requirements

System must be able to push job details into database and create a job posting on both buyer and seller side in "Pending Orders" tab. This job card must include all Important details of a posted Job offer.

4.6 Estimate

4.6.1 Description and Priority

Used by buyer and seller to agree upon a price of service which includes cost of labor cost, material cost and time required to complete job.

Priority: Essential

4.6.2 Stimulus/Response Sequences

Seller Receives job offer.

Buyer contacts seller and discuss cost and requirements based off Job details

Buyer and Seller enters details into estimate view.

System shows buyer agreed upon price and provides them with the option to proceed with job or get other estimates.

4.6.3 Functional Requirements

System should be able to relay contact details of buyer and seller to each other.

4.7 Schedule Appointment

4.7.1 Description and Priority

Used by buyer to set appointment from technician to start the job.

Priority: Important

4.7.2 Stimulus/Response Sequences

Buyer and Seller both Accept cost estimate and job offer

Buyer and Seller enter agreed upon job schedule

System created pending job card with scheduled time and job details in pending jobs tab of buyer and seller orders.

4.7.3 Functional Requirements

System should be able to push job data into database and create pending job card in both Buyer and Seller Order tabs.

4.8 Comparison

4.8.1 Description and Priority

Used by seller to get estimates from different technicians on the same job

Priority: Important

4.8.2 Stimulus/Response Sequences

Buyer selects multiple technicians for one job posting.

Sellers receive job posting and submit individual offers based off 4.6 Estimate requirement.

System shows buyer estimates of multiple technicians and gives them option to accept/reject offer of choice.

4.8.3 Functional Requirements

To Be Decided

4.9 View Orders

4.9.1 Description and Priority

Used by both buyer and seller to view past, current, and pending orders

Priority: Essential

4.9.2 Stimulus/Response Sequences

User selects orders tab

User is navigated to current orders page that displays all current orders

System shows navigation path for Past and Pending Order tabs

System redirects to user selection

4.9.3 Functional Requirements

System must be able to fetch and display all orders in correct order tab from the database.

4.10 Other Functional Requirements To Be Decided

5. Other Nonfunctional Requirements To Be Decided

5.1 Safety Requirements

All users **MUST** accept jobs through the app and not through personal means after getting in contact with sellers through the app. This is the only way accountability can be guaranteed for both buyers and sellers.

5.3 Software Quality Attributes

The application and user interface must be easy to use and satisfying for customers. The app must be able to retain users even after incentives are discontinued to continue to keep its users through the functionalities of the application.

5.4 Business Rules

All Users can avail Buyer Services as soon as they log into the application however Buyers can only access Seller/Service Provider page after they have entered their details into the Technician Registration Form and have been verified and authorized by an Administrator, Only then will the user have access to the Seller Page.

6. Appendix A: Glossary

Buyer: Customer who needs technical services

Seller: Technician who provides technical services

Essential: Priority level for requirements that are essential to the operation of the project

Important: Priority level for requirements that are necessary for the operation of the project to be as intended

Desirable: Priority level for requirements that are desirable to improve the project but are not necessary for the project to operate.

7. Appendix B: To Be Determined List

- **3.4 Communication interfaces**
- **Functional Requirements:**
  - **4.8.3 Comparison**
- **4.10 onwards, Functional requirements later on in the project.**

8. Appendix C: Analysis Models

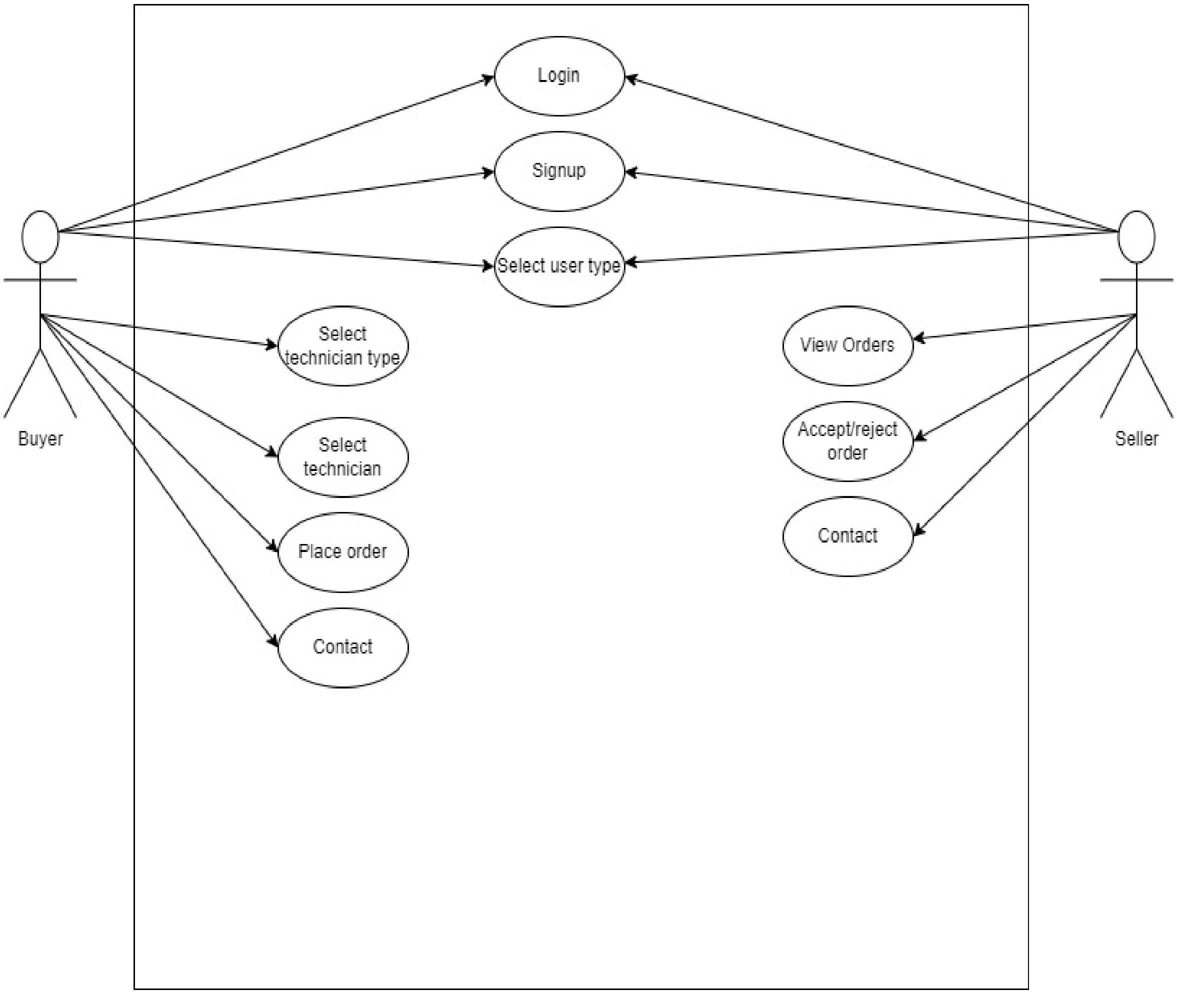
All Diagrams and Analysis Models are attached below along with titles.

9. Test Cases

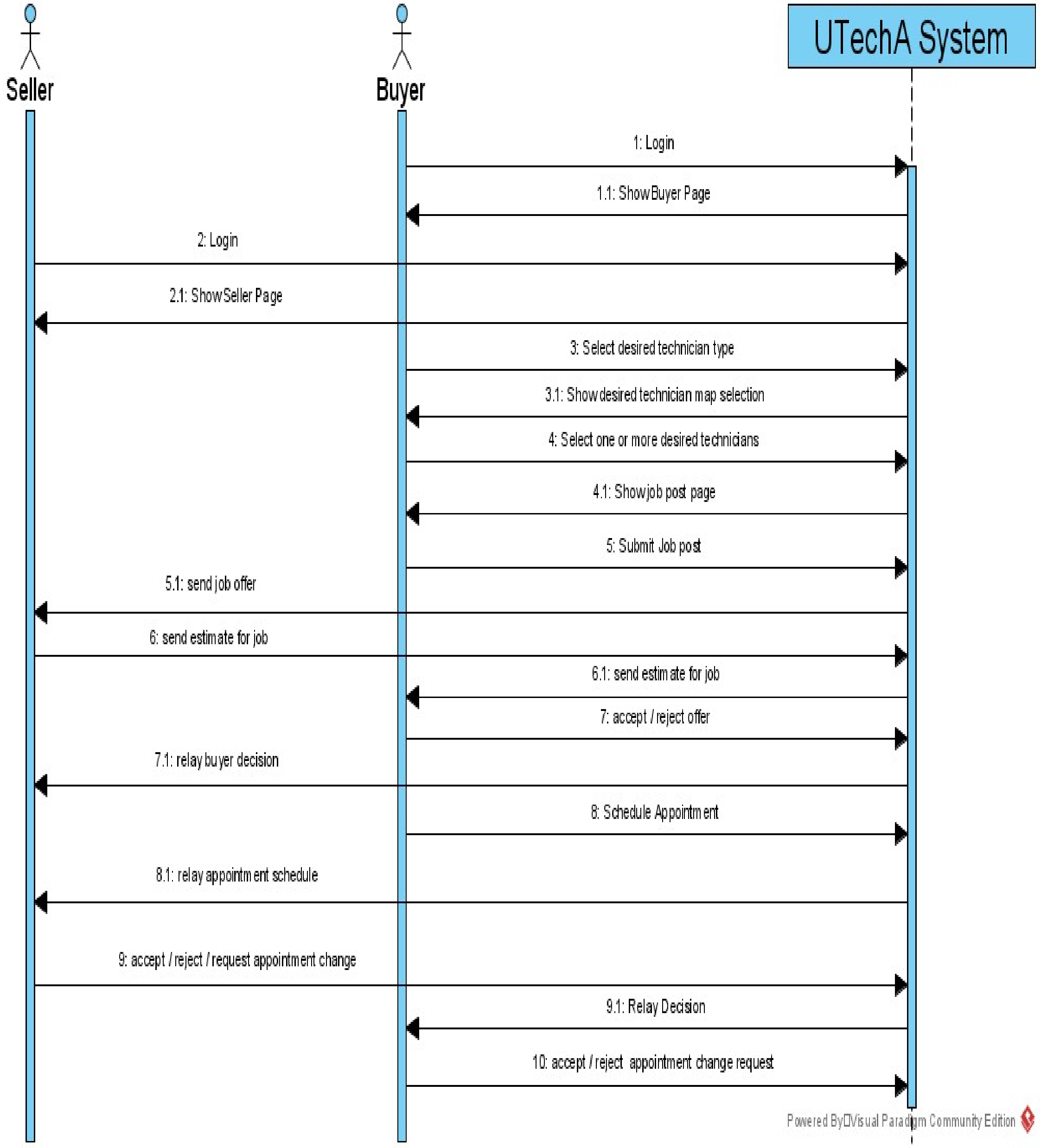
Test Case No.	Description	Input	Expected Result	Pass/Fail/Not yet tested
1	Login	Correct Buyer credentials	Show Buyer Page	PASS
2	Login	Correct seller credentials	Show Seller Page	PASS
3	Login	Incorrect buyer credentials	Login Failed message	PASS
4	Login	Incorrect seller credentials	Login Failed Message	PASS
5	Signup	Enter Details	Store data in database and create new user	PASS
6	User Type Selection	Select Looking for a service	System redirects to technician selection page	PASS
7	User Type Selection	Authorized Seller selects Providing a service	System redirects to Seller Orders Page	PASS
8	User Type Selection	Unauthorized Seller selects Providing a service	System redirects to Technician Registration Page	PASS
9	Technician Type Selection	Select required technician type	Redirect to selected technician type Map	PASS
10	Map Technician Selection	Select Desired Technician	Show Job Posting page	PASS

11	Job Posting	User enters job details and posts job	System pushes job data into database and presents it to seller	PASS
12	Contact	User Clicks Contact Button	System Redirects to device Dialpad and initiates Call	PASS
13	Dynamic Marker Selection	User Clicks Marker of Desired Shop	System Shows Shop And Map Details	PASS
14	Marker Routing	User Clicks Direction After Clicking Shop Marker	System Redirects to Google Maps with Directions to shop	PASS
15	Upload Photo	User Uploads Photo in Place Order Tab	System Uploads Photo to Firebase Storage	PASS
16	Offer Price	Seller Enters Price Offer for Job	System Sends Offer to Buyers Pending Job Tab	PASS
17	Accept Price	Buyer Accepts Price Offer	System Moves Order Card from Pending to Current for both Users	PASS
18	Reject Price	Buyer Rejects Price Offer	System Deletes Order Card for Both Users	PASS
19	Rate Button	Buyer Clicks Rate on Completed Order	System Moves to Rating Screen	PASS

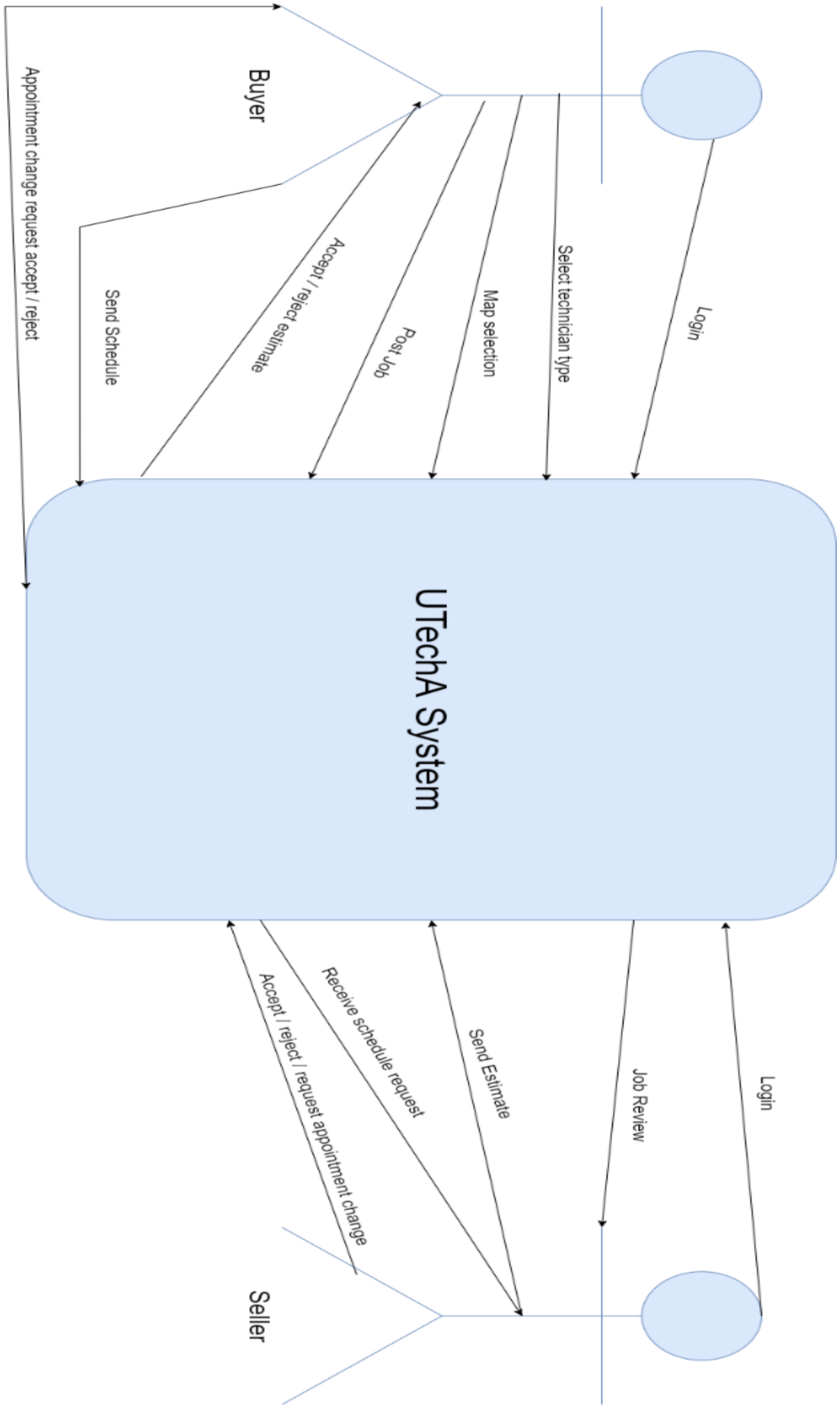
Use Case Diagram



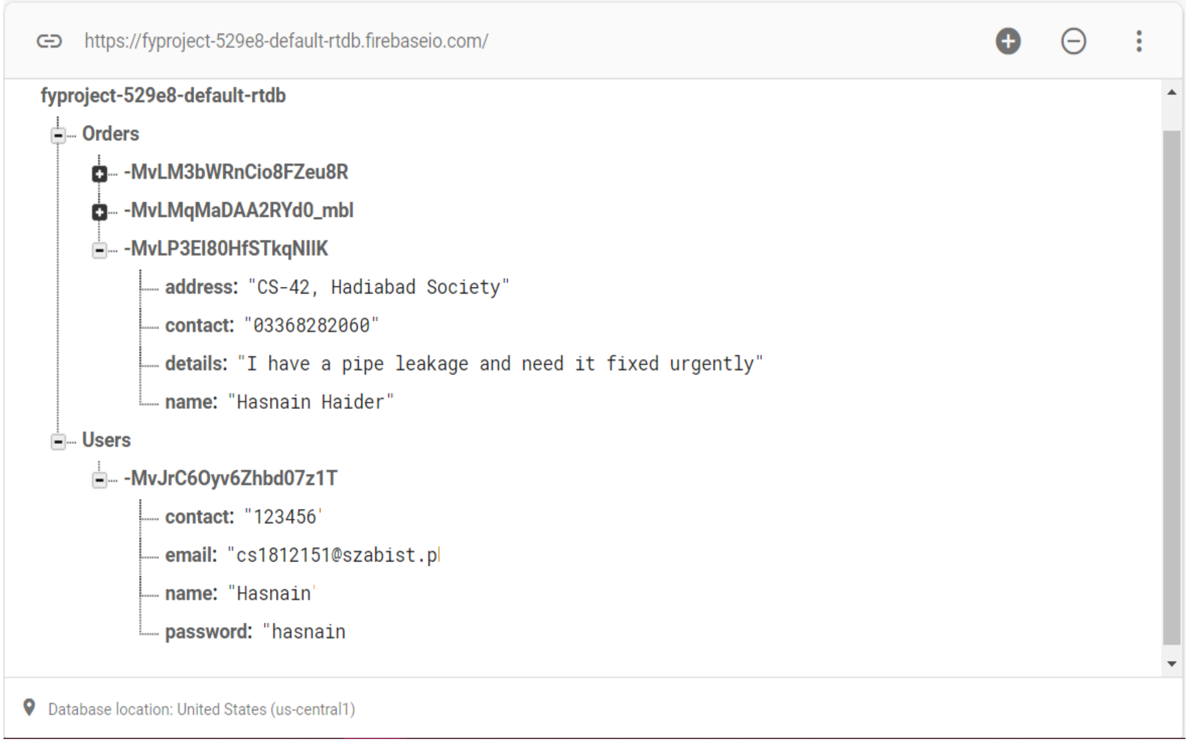
10. System Sequence Diagram


Seller
Buyer
UTechA System
1: Login
1.1: Show Buyer Page
2: Login
2.1: Show Seller Page
3: Select desired technician type
3.1: Show desired technician map selection
4: Select one or more desired technicians
4.1: Show job post page
5: Submit Job post
5.1: send job offer
6: send estimate for job
6.1: send estimate for job
7: accept / reject offer
7.1: relay buyer decision
8: Schedule Appointment
8.1: relay appointment schedule
9: accept / reject / request appointment change
9.1: Relay Decision
10: accept / reject appointment change request
Powered By Visual Paradigm Community Edition

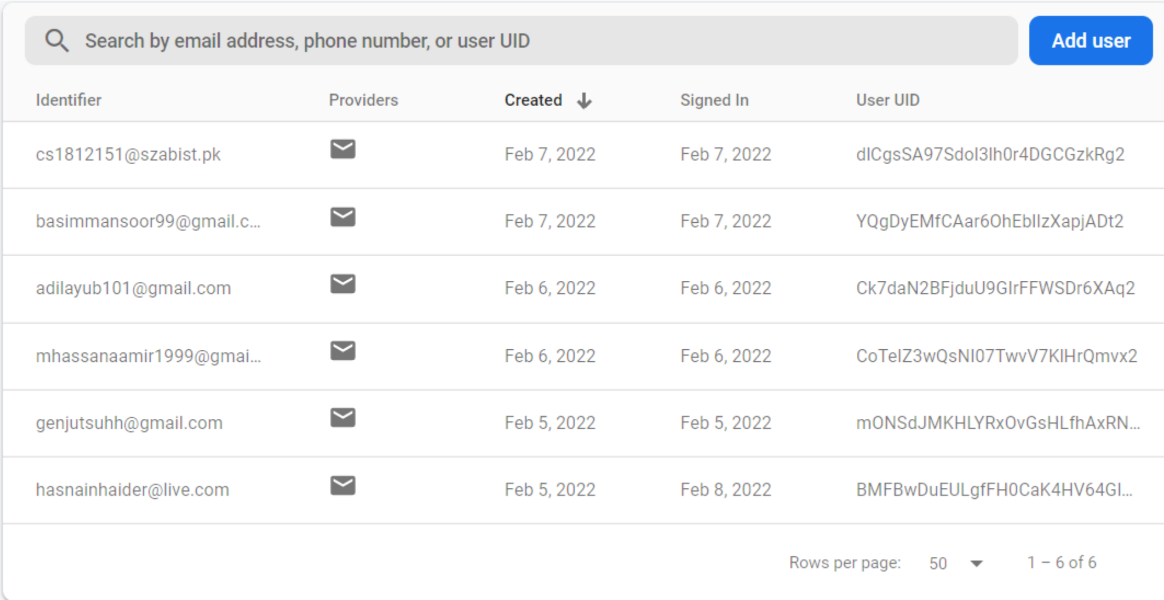
11. Context Diagram



12. ERD (Firebase Connections)



```
https://fyproject-529e8-default-rtdb.firebaseio.com/

fyproject-529e8-default-rtdb
  Orders
    -MvLM3bWRnCio8FZeu8R
    -MvLMqMaDAA2RYd0_mbl
    -MvLP3EI80HfSTkqNIIK
        address: "CS-42, Hadiabad Society"
        contact: "03368282060"
        details: "I have a pipe leakage and need it fixed urgently"
        name: "Hasnain Haider"
  Users
    -MvJrC6Oyv6Zhbd07z1T
        contact: "123456'
        email: "cs1812151@szabist.p
        name: "Hasnain'
        password: "hasnain

Database location: United States (us-central1)
```



Identifier	Providers	Created ↓	Signed In	User UID
cs1812151@szabist.pk	✉	Feb 7, 2022	Feb 7, 2022	dlCgsSA97Sdol3Ih0r4DGCGzkRg2
basimmansoor99@gmail.c...	✉	Feb 7, 2022	Feb 7, 2022	YQgDyEMfCAar6OhEbIIzXapjADt2
adilayub101@gmail.com	✉	Feb 6, 2022	Feb 6, 2022	Ck7daN2BFjduU9GIrFFWSDr6XAq2
mhassanaamir1999@gmai...	✉	Feb 6, 2022	Feb 6, 2022	CoTeIZ3wQsNI07TwvV7KIHrQmvx2
genjutsuhh@gmail.com	✉	Feb 5, 2022	Feb 5, 2022	mONSdJMKHLYRxOvGsHLfhAxRN...
hasnainhaider@live.com	✉	Feb 5, 2022	Feb 8, 2022	BMFBwDuEULgfFH0CaK4HV64GI...

Software Design Specification

1. Introduction

1.1 Purpose of this document

The main purpose of this document is to provide context to decisions made by the developers and to explain their choices during the applications design process. This document will go in depth into each functionality, component, UI, design choice and more that were made. This document will primarily focus on the components that were either created or used in the application to provide detailed descriptions of how the application was created.

1.2 Scope of the development project

The purpose of this project is to provide a platform for customers to connect to small technical business owners such as plumbers' electricians etc. within their local area. The system is meant to provide customers with information about the technicians including their rating and schedule of availability.

The system will deal with both customers and technicians on different views depending on who they login as. An interface will be provided for customers to select different types of technicians that they require. Another interface will allow them to locate and select technicians on a map in their local area.

In order to launch the project as a business, a business model similar to that of PayPal will be required with incentives set in place for both buyers and sellers to attract them to the platform. The usability of the project and features must be enough to keep the client base after the incentive program wares off.

1.3 Definitions, acronyms and abbreviations

Acronyms/Abbreviations	Definitions
User	Class of user type in main menu before service type selection
Buyer	User who is looking to purchase a service
Seller	User who is providing a service
Android	An open-source operating system used for smartphones and tablet computers
API	(Application Programming Interface) a set of functions and procedures that allows the creation of applications which access the features or data of an operating system, application or other service.
Database	A structured set if data held in a storage device, especially one that is accessible through multiple access points
Firebase	A collection of cloud services including databases, analytics services and much more as part of the Google Cloud Services Platform
React Native	React Native is an open-source UI Framework based on Javascript used to develop applications using the React Framework.

Native Base	NativeBase is an accessible, utility-first component library that helps you build UI across Android, iOS and Web
Google Maps	Google Maps is a web mapping platform and consumer application offered by Google
Google Cloud Services	Google Cloud Platform, offered by Google, is a suite of cloud computing services
SDK	Software Development Kit. A set of programs that allows software developers to create products that run on a particular platform or work with an API
SRS	Software Requirements Specification
SDS	Software Design Specification

1.4 References

Please Refer to the appendix to locate the references near the end of the document.

1.5 Overview of the document

This document provides details and context for the system’s major components and features. It also provides the interface, which goes in depth about the interaction between users and the system, stimulus and response details etc. Context is also provided for design and UI choices made by the developers while designing the aesthetics and usability of the application. We also explain the steps we took to focus on code reuse. We also explain the major decisions we had to make during the development of this project, why we made those decisions and we discuss what were the other alternatives available to us that we did not select and why. After that we provide the Pseudocode for provide any context/clarity for functionalities of the application.

2. System Architecture Description

2.1 Overview of Models/Components

UtechA is an application that intends to connect service providers and technicians to customers who require specialized services. The main purpose of the application is to build a database of local workforce and businesses that are not centralized on a platform, most of these businesses and service providers have no online presence at all for their businesses or services. Our goal is to create a centralized platform where technicians, business owners and service providers can register and be connected to customers in their local area. Our goal is also to create a system where accountability prevails above all else and quality of work can be assured and maintained. We also want to create a database of customers who can access this workforce through our application platform and get the services they require from the comfort of their own home. The framework we have designed uses a single application for both Buyers and Sellers rather than having two different applications for each user type, this will require only one app to be installed on a device and data can be consolidated much more easily by controlling access to Seller services for only Authorized sellers who need to be verified after applying. Our system allows us to easily add new services and technicians over the course of time.

2.2 General Constraints

Communication is a major issue we assume to happen between buyers and sellers as most of the local service providers do not have enough communication skills to effectively demonstrate the job requirements to customers through text either in English or Urdu. Although we could offer training sessions for such service providers, that would take them out of the workforce for too long and be unreasonable leading to technicians leaving our platform. Therefore, we decided to go with voice calls using local telecommunication. Almost all technicians already have a cell phone number registered in their name which makes buyer and seller contact simple and straight forward. However, this introduces an issue in itself, we must have check and balance for this by introducing a call masking service that can protect buyer contact information and forcing both parties to conduct all business through the application.

2.3 User Interface Issues

The user interface consists of 11 screens/activities, which are:

Login, Signup, User type selection, Technical service selection, Technician selection, Technician Details, Place order, Register as technician, Current Orders, Pending Orders, Completed Orders.

1. Login will allow authorized users to access the application features
2. Signup will allow users to register as authorized users
3. User type selection will allow user to select interacting with the app as a buyer or a service provider/seller
4. Technical service selection is a buyer component that allows buyers to select the type of service they desire
5. Technician selection is the view of the available shops in the user's local area of that specific service
6. Technician Details shows the details of the technician selected by the user
7. Place order will allow user to enter job details and place an order/get an estimate by contacting the technician

8. Register as technician will allow users to enter details to get authorization for seller services after admin approval
9. Current Orders allows user to view current orders
10. Pending Orders allows user to view pending orders that have not been accepted or rejected
11. Completed Orders allows users to view past completed orders

2.4 Section Overview

This section contains the system architecture including the overview of modules, components, general constraints, data design, program structure and alternatives that were considered.

2.5 Data Design

We decided to use Google cloud services for our database, specifically Google Firebase as it allows us to store and retrieve data online from the cloud without having to physically connect devices to a storage medium. Firebase also allows easy integration with React Native through packages such as RNFirebase. On our database we store user information and order information.

2.6 Program Structure

Refer to Appendix for Program Structure and navigation diagram.

2.7 Alternatives Considered

Many alternatives were considered for both the Maps API and the Database during the design process. For the Maps API an alternative that was considered was OpenLayers which is a free to use Maps API with features similar to what we needed for our project. However, due to the immense flexibility and documentation available with Google Maps API along with the unparalleled optimization of Google Maps API with Android devices, we decided on Google Maps for our Map API.

For the database, we considered many options such as SQLite, MariaDB, RealM and WatermelonDB. Most of these options however required us to have physical connections of the device to the database to update in real time. The other option available was to establish a server that would allow us to connect to these services over the internet. However, Firebase allowed us to do just that without any setup and worked instantly over the cloud allowing us to update, post and retrieve from the database over the internet without a physical link to the database. That coupled with the RNFirebase packages that allow seamless integration of React Native with Firebase made choosing Firebase an easy decision.

3. Detailed Description of Components

Database: We will be using Firebase connected to React Native through RNFirebase packages to store and retrieve user and order data on the cloud.

Client Software: This component is the Application Data that will be installed on the client device to allow them to interact with the app and use it as intended.

Google Maps API: This API component allows us to easily integrate GPS mapping and location services in our application.

RNFirebase: is a collection of packages that allows seamless integration of the Firebase database to the React Native environment

ReactNative Immediate Phone Call API: This API allows us to access the dialpad and cellular features of the device software and initiate the Call.

3.1 Database

Identification	Firebase
Type	Database Module
Purpose	Provides data management and storage for the application
Function	Stores, retrieves and updates database with user data
Subordinates	N/A
Dependencies	Google Cloud Services, RNFirebase
Interfaces	User Data: • Stores user data Order Data • Stores order data
Resources	Certificate fingerprints for database connection. Internet connection
Processing	Firebase Push and Retrieve functions will be used along with SHA 1 certificate fingerprint to store data in the database
Data	Forms in the application will be used to enter data of multiple data types into the database along with parity checks. Datatype used is String Firebase storage for images.

3.2 Client Software

Identification	Android User
Type	User Interface
Purpose	Provides access to application and features including the ability to navigate using the UI
Function	• Login • Signup • User type selection • Technical service selection • Technician selection • Technician Details • Place order • Register as technician • Current Orders • Pending Orders • Completed Orders.
Subordinates	N/A
Dependencies	Android OS, React Native, Native Base
Interfaces	Refer to Navigation Screenshots from appendix
Resources	Smartphone, internet connection, CPU & GPU, SHA certificate
Processing	CPU data processing, GPU render processing, Communication with database over the internet
Data	Input text and images through virtual smartphone keyboard

3.3 React Native

Identification	ReactNative
Type	Framework
Purpose	Used to create the application
Function	Allow development using the React Framework
Subordinates	Native Base
Dependencies	JDK, NODE, Android SDK platform
Interfaces	N/A
Resources	JDK, NODE
Processing	N/A
Data	N/A

3.4 RNFirebase

Identification	RNFirebase
Type	Collection package
Purpose	A collection of official React Native modules to connect the application to Firebase services
Function	Used to connect the application to Firebase services
Subordinates	N/A
Dependencies	React Native, Firebase
Interfaces	N/A
Resources	N/A
Processing	N/A
Data	N/A

3.5 React Native Immediate Phone Call API

Identification	Phone Call API
Type	API
Purpose	Used to integrate cellular features of smartphone to application
Function	Contact button
Subordinates	N/A
Dependencies	React Native, Cellular Service, SIM card
Interfaces	N/A
Resources	Phone Number required to initiate phone call
Processing	N/A
Data	11-digit phone number

3.6 Google Maps API

Identification	Google Maps API
Type	API
Purpose	Used to integrate map functionality into the application
Function	Tracking, Geolocation, Markers.
Subordinates	N/A
Dependencies	Google Cloud Services, API key
Interfaces	N/A
Resources	N/A
Processing	N/A
Data	Longitude & Latitude

4. User Interface Design

4.1 Section Overview

This section contains information about the user interface design including design decisions and GUI components and views.

4.2 Interface Design Rules

The interface has been designed adhering to the proposed design of Android Studio SDK and React Native's Native Base framework which provides buttons, cards, vector icons and many more GUI components. A user-friendly User experience was designed to easily navigate through the application. The color palette was specifically chosen from a survey that proved the colors White, Teal and Cyan (HEX codes: #ffffff, #52ab98 & #2b6777) which were proven to be trustworthy tones as we wanted to portray a reliable trustworthy application

4.3 GUI Components

The components used are those provided by React Native and Native Base. The API's provided are: Google Maps API, React Native Immediate Call API

4.4 Detailed Description

A clean, simple, easy to use and user-friendly interface has been implemented to make the user experience pleasant without needing a learning curve to adjust to the application.

Detailed Views are provided at the end of the document, refer to the diagrams in appendix labelled Views.

5. Reuse and relationships to other products

5.1 React Native Tools

From the very beginning we had decided to use and reuse open-source code and APIs that were already published rather than trying to reinvent the wheel. UI design, API integration, Call features, Maps etc. have all been created and published for the world to use to their advantage without having to start over from scratch.

An example of components we reused include:

Native Base UI components

We tweaked the predefined functions and components to meet our needs rather than creating UI components from scratch.

5.2 Database

We chose Firebase services for our database management due to its ease of use, analytical features and optimization with android OS along with its availability over the cloud.

5.3 API

Google Maps API was chosen due to its extensive documentation, ease of integration and availability along with google cloud services

6. Design Decisions and Tradeoffs

6.1 Android VS IOS

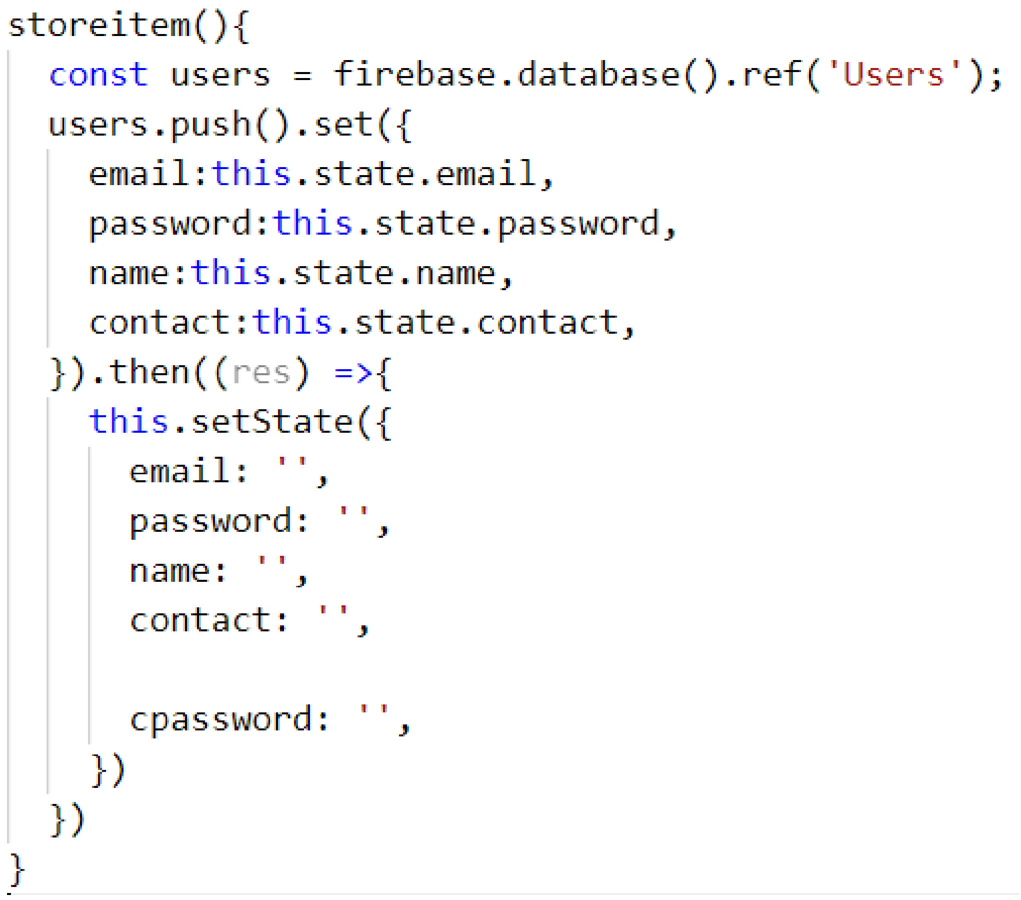
React Native was chosen as our development environment as it is a dynamic development environment that can be deployed on Android, IOS and Web with a few minor adjustments. However, while testing we optimized the application for Android as it had the easiest development and testing environment. On top of that both developers have no experience with IOS development but have extensive experience with Javascript and Android App Development.

6.2 Single Application Design

We decided to create a single application for both buyers and sellers to allow users to be able to access both aspects of the application if they have the proper authentication. We believe this will allow us to retain more users rather than if service providers had to download another application if they needed a service themselves. However, this choice did make UI design and navigation much more difficult as compared to what it could’ve been with two separate applications.

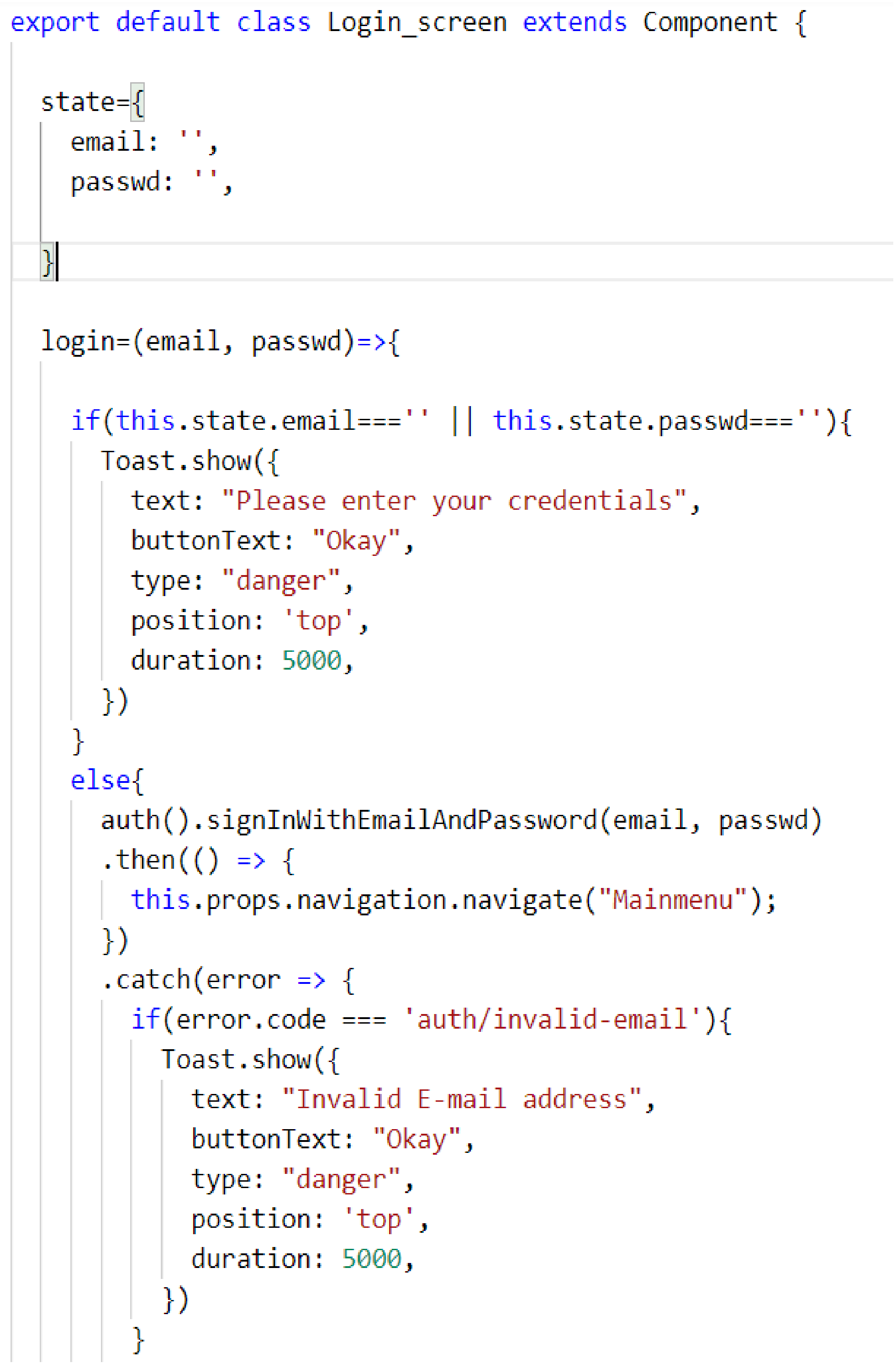
7. Pseudocode for Components

7.1 Signup



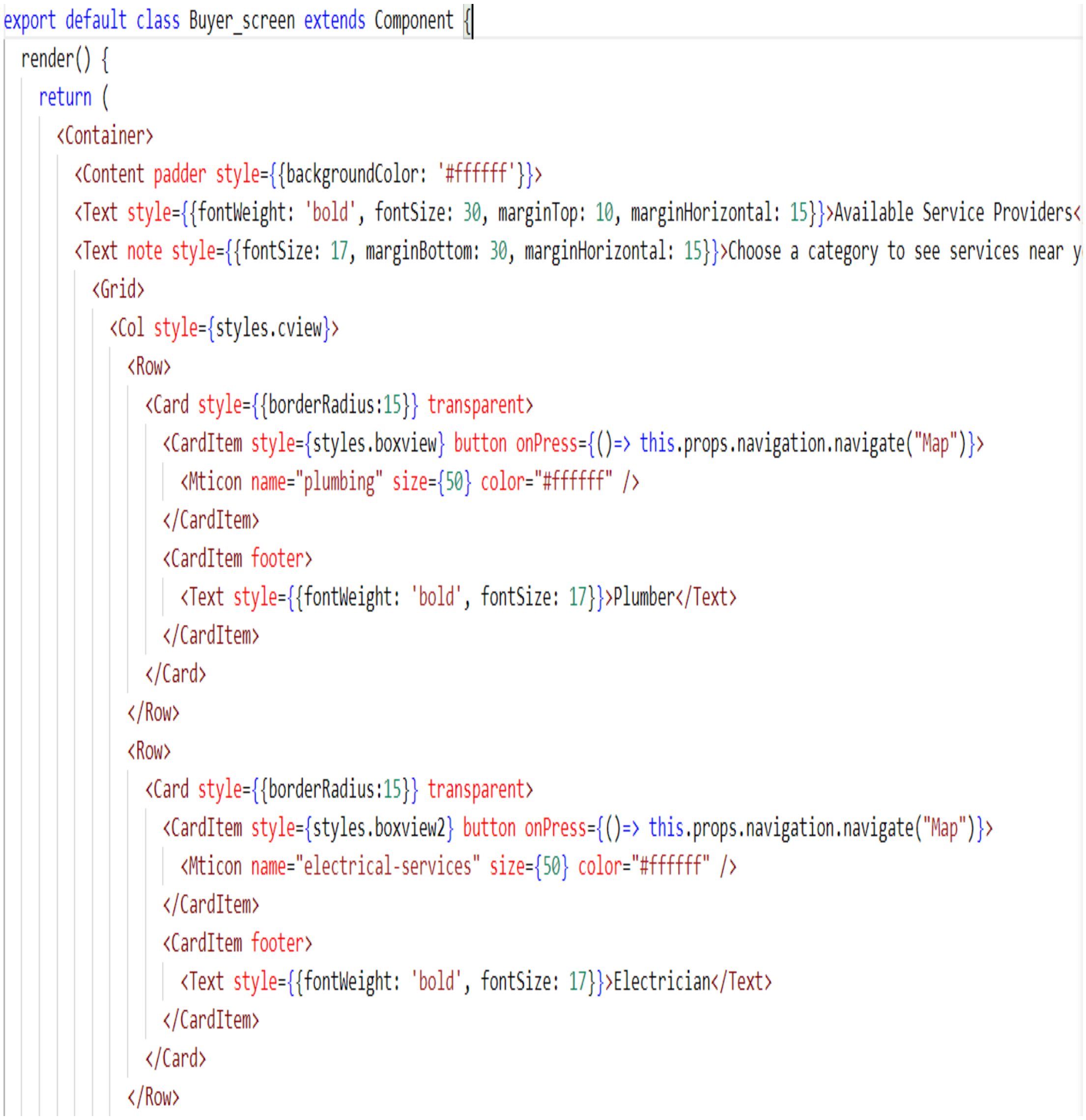
```
storeitem(){
  const users = firebase.database().ref('Users');
  users.push().set({
    email:this.state.email,
    password:this.state.password,
    name:this.state.name,
    contact:this.state.contact,
  }).then((res) =>{
    this.setState({
      email: '',
      password: '',
      name: '',
      contact: '',

      cpassword: '',
    })
  })
}
```

7.2 Login



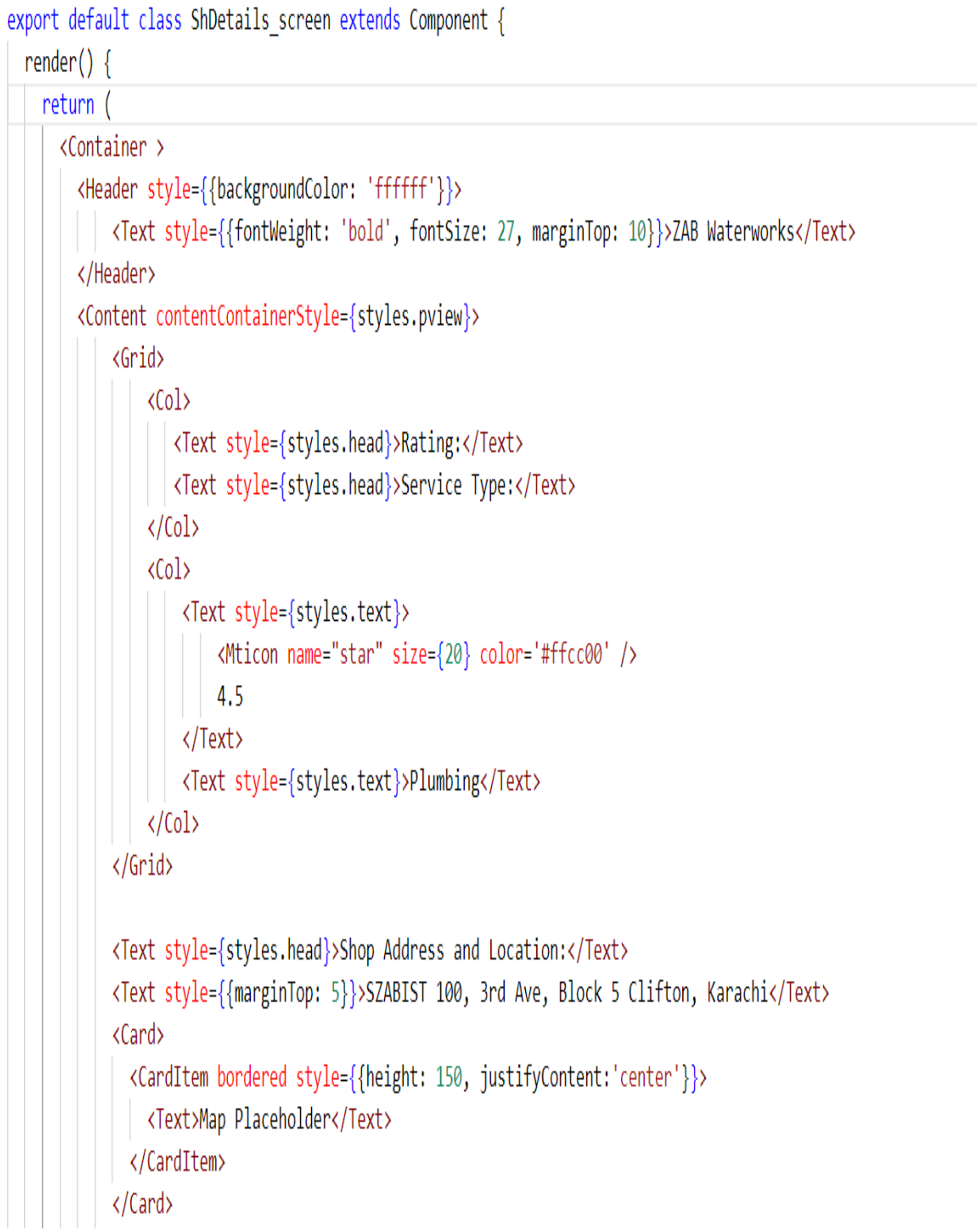
```
export default class Login_screen extends Component {

  state={
    email: '',
    passwd: '',

  }

  login=(email, passwd)=>{

    if(this.state.email==='' || this.state.passwd===''){
      Toast.show({
        text: "Please enter your credentials",
        buttonText: "Okay",
        type: "danger",
        position: 'top',
        duration: 5000,
      })
    }
    else{
      auth().signInWithEmailAndPassword(email, passwd)
      .then(() => {
        this.props.navigation.navigate("Mainmenu");
      })
      .catch(error => {
        if(error.code === 'auth/invalid-email'){
          Toast.show({
            text: "Invalid E-mail address",
            buttonText: "Okay",
            type: "danger",
            position: 'top',
            duration: 5000,
          })
        }
```

7.3 Buyer Service Tiles



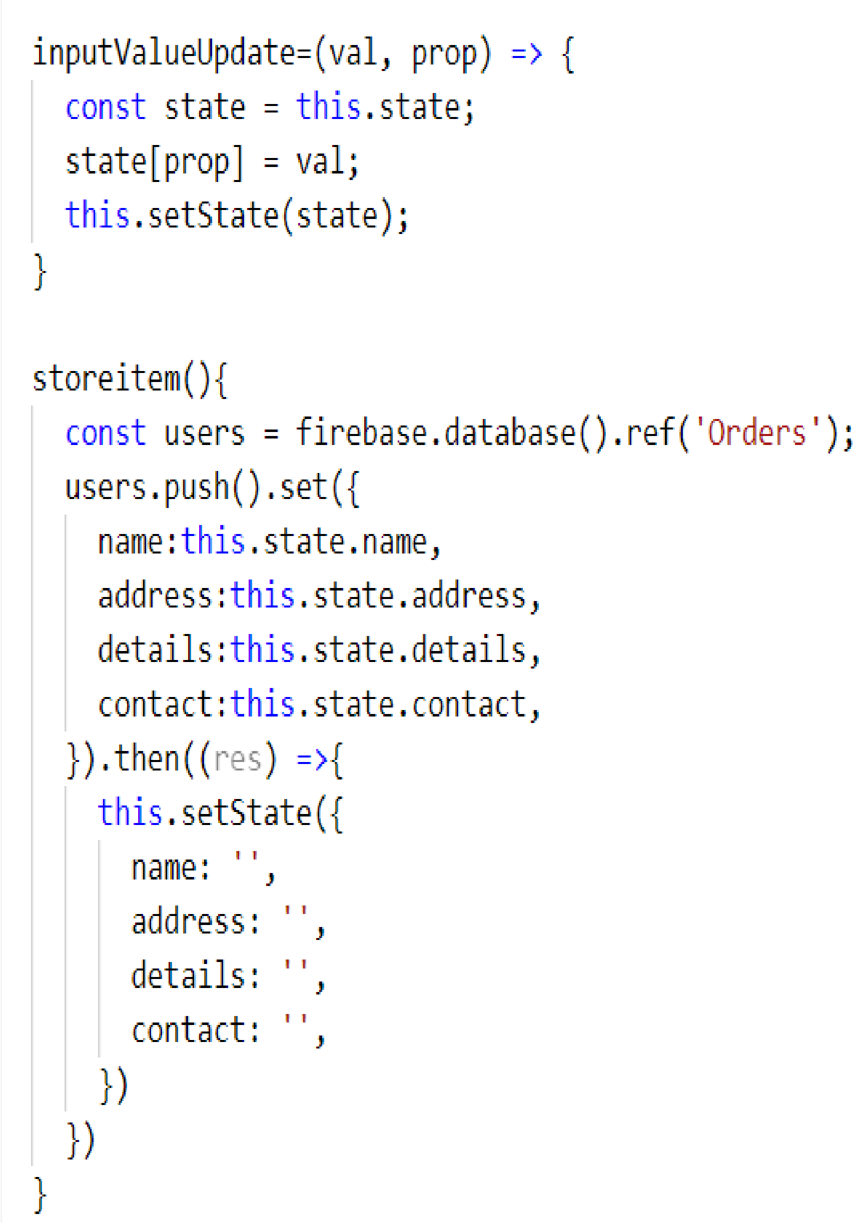
```
export default class Buyer_screen extends Component {
  render() {
    return (
      <Container>
        <Content padder style={{backgroundColor: '#ffffff'}}>
        <Text style={{fontWeight: 'bold', fontSize: 30, marginTop: 10, marginHorizontal: 15}}>Available Service Providers<
        <Text note style={{fontSize: 17, marginBottom: 30, marginHorizontal: 15}}>Choose a category to see services near y
          <Grid>
            <Col style={styles.cview}>
              <Row>
                <Card style={{borderRadius:15}} transparent>
                  <CardItem style={styles.boxview} button onPress={()=> this.props.navigation.navigate("Map")}>
                    <MtIcon name="plumbing" size={50} color="#ffffff" />
                  </CardItem>
                  <CardItem footer>
                    <Text style={{fontWeight: 'bold', fontSize: 17}}>Plumber</Text>
                  </CardItem>
                </Card>
              </Row>
              <Row>
                <Card style={{borderRadius:15}} transparent>
                  <CardItem style={styles.boxview2} button onPress={()=> this.props.navigation.navigate("Map")}>
                    <MtIcon name="electrical-services" size={50} color="#ffffff" />
                  </CardItem>
                  <CardItem footer>
                    <Text style={{fontWeight: 'bold', fontSize: 17}}>Electrician</Text>
                  </CardItem>
                </Card>
              </Row>
```

7.4 Shop Details



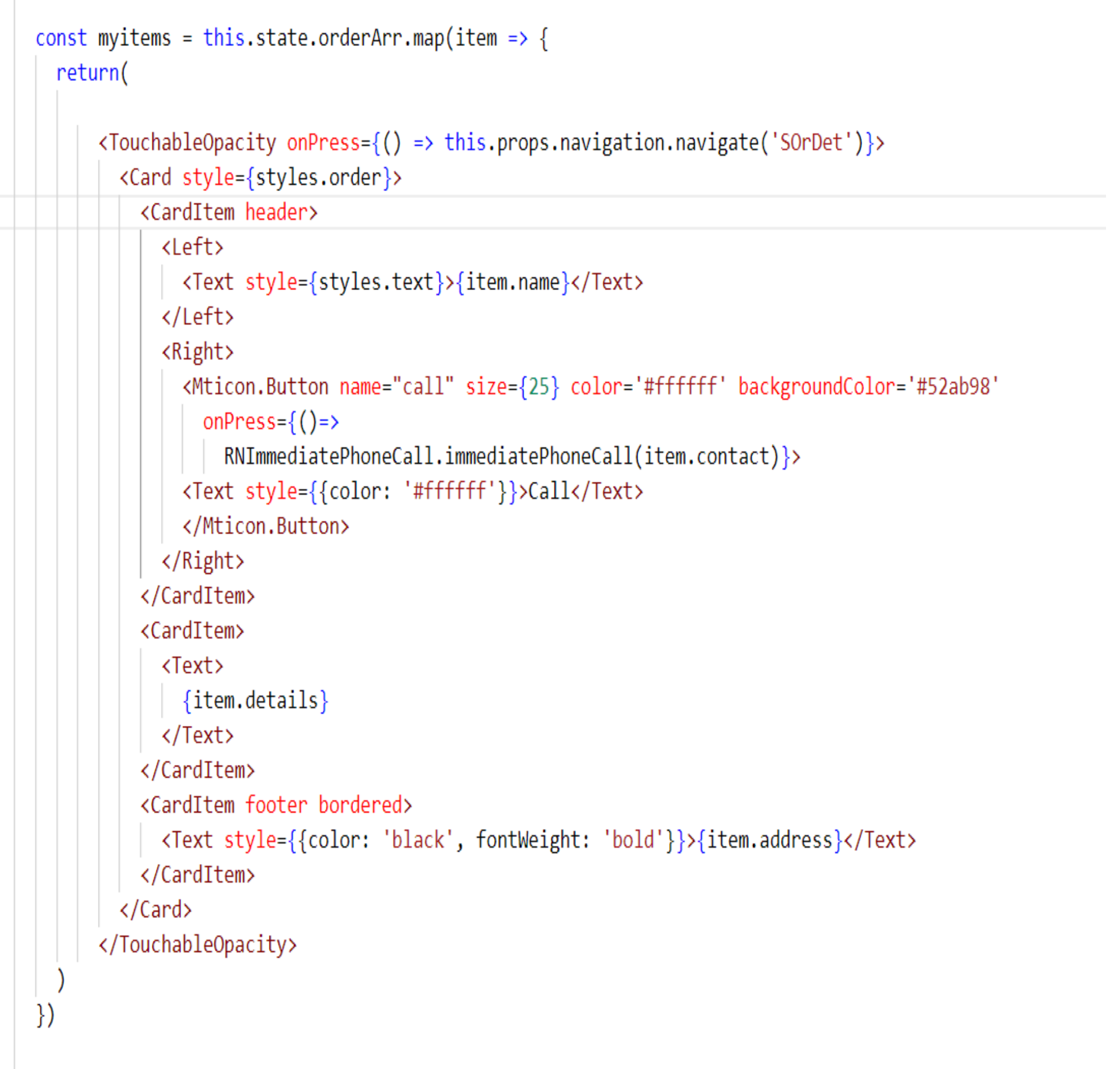
```
export default class ShDetails_screen extends Component {
  render() {
    return (
      <Container >
        <Header style={{backgroundColor: 'ffffff'}}>
            <Text style={{fontWeight: 'bold', fontSize: 27, marginTop: 10}}>ZAB Waterworks</Text>
        </Header>
        <Content contentContainerStyle={styles.pview}>
            <Grid>
                <Col>
                    <Text style={styles.head}>Rating:</Text>
                    <Text style={styles.head}>Service Type:</Text>
                </Col>
                <Col>
                    <Text style={styles.text}>
                        <Mticon name="star" size={20} color='#ffcc00' />
                        4.5
                    </Text>
                    <Text style={styles.text}>Plumbing</Text>
                </Col>
            </Grid>

            <Text style={styles.head}>Shop Address and Location:</Text>
            <Text style={{marginTop: 5}}>SZABIST 100, 3rd Ave, Block 5 Clifton, Karachi</Text>
            <Card>
              <CardItem bordered style={{height: 150, justifyContent:'center'}}>
                <Text>Map Placeholder</Text>
              </CardItem>
            </Card>
```

7.5 Place Order



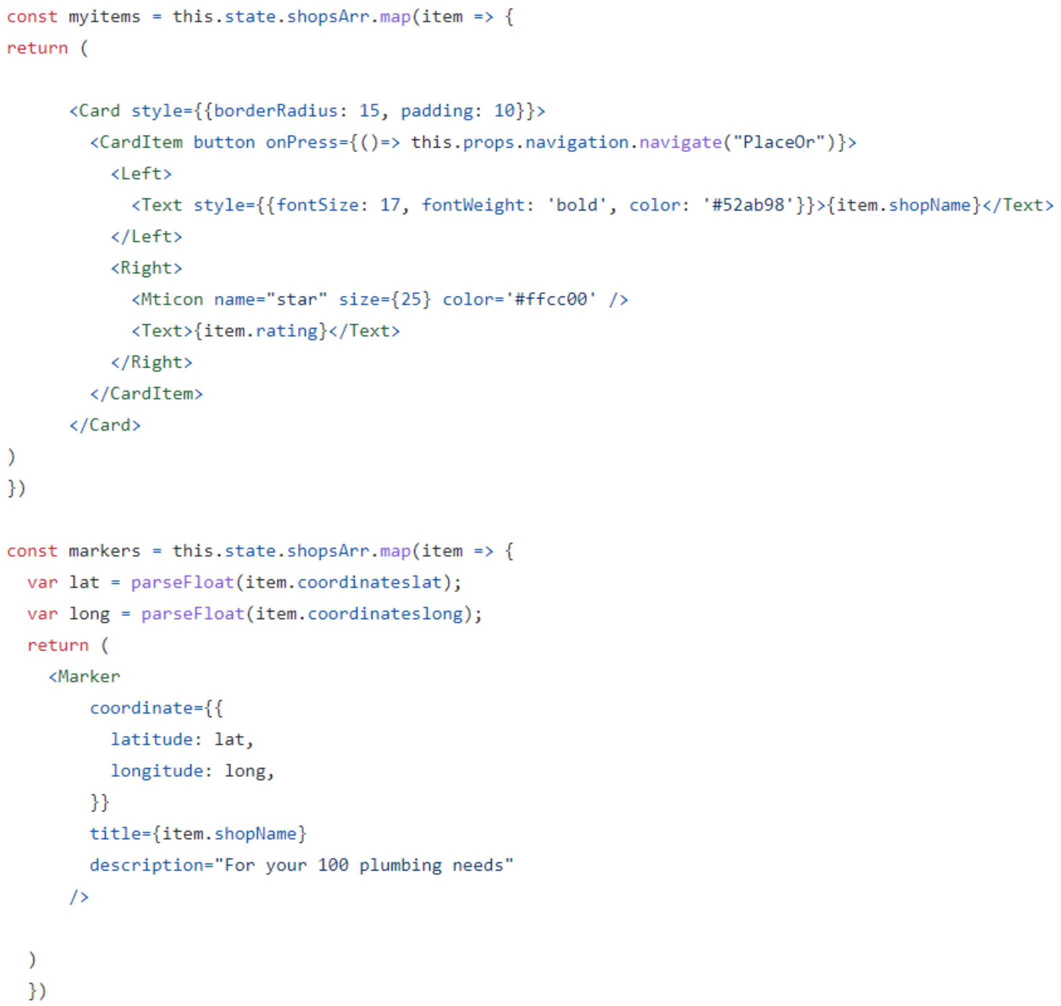
```
inputValueUpdate=(val, prop) => {
  const state = this.state;
  state[prop] = val;
  this.setState(state);
}

storeitem(){
  const users = firebase.database().ref('Orders');
  users.push().set({
    name:this.state.name,
    address:this.state.address,
    details:this.state.details,
    contact:this.state.contact,
  }).then((res) =>{
    this.setState({
      name: '',
      address: '',
      details: '',
      contact: '',
    })
  })
}
```

7.6 Pending Orders



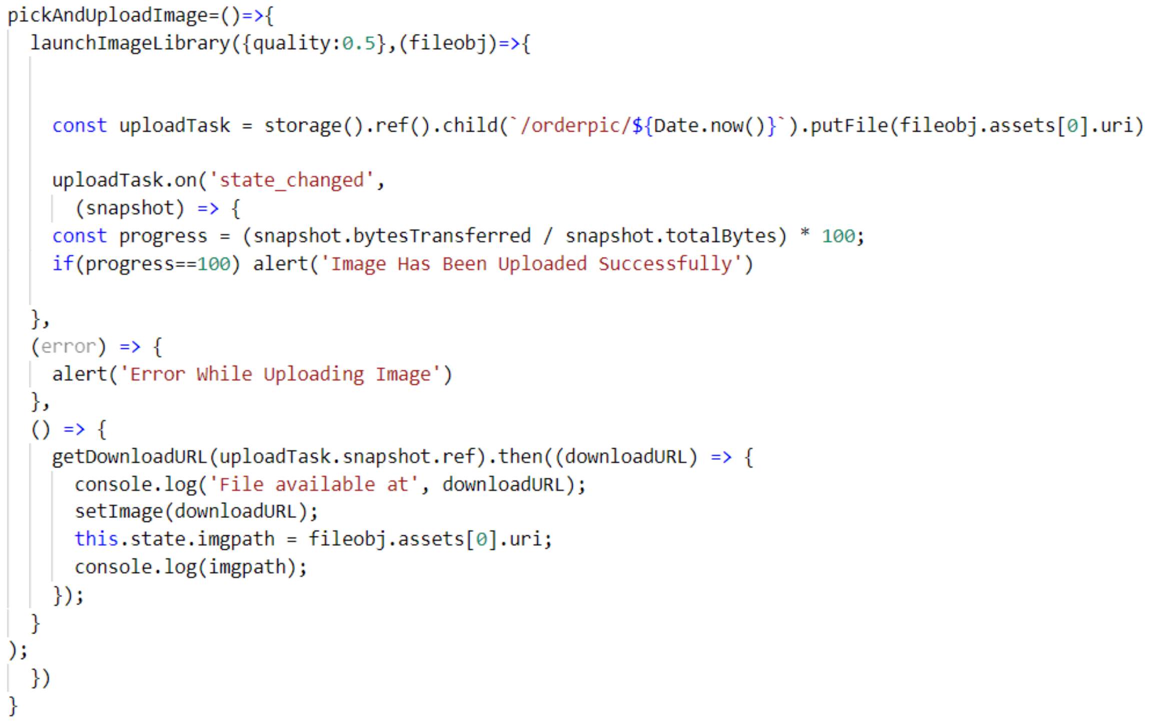
```
const myitems = this.state.orderArr.map(item => {
  return(

      <TouchableOpacity onPress={() => this.props.navigation.navigate('SOrDet')}>
        <Card style={styles.order}>
          <CardItem header>
            <Left>
              <Text style={styles.text}>{item.name}</Text>
            </Left>
            <Right>
              <Mticon.Button name="call" size={25} color='#ffffff' backgroundColor='#52ab98'
                onPress={()=>
                  RNImmediatePhoneCall.immediatePhoneCall(item.contact)}>
              <Text style={{color: '#ffffff'}}>Call</Text>
              </Mticon.Button>
            </Right>
          </CardItem>
          <CardItem>
            <Text>
              {item.details}
            </Text>
          </CardItem>
          <CardItem footer bordered>
            <Text style={{color: 'black', fontWeight: 'bold'}}>{item.address}</Text>
          </CardItem>
        </Card>
      </TouchableOpacity>
  )
})
```

7.7 Map



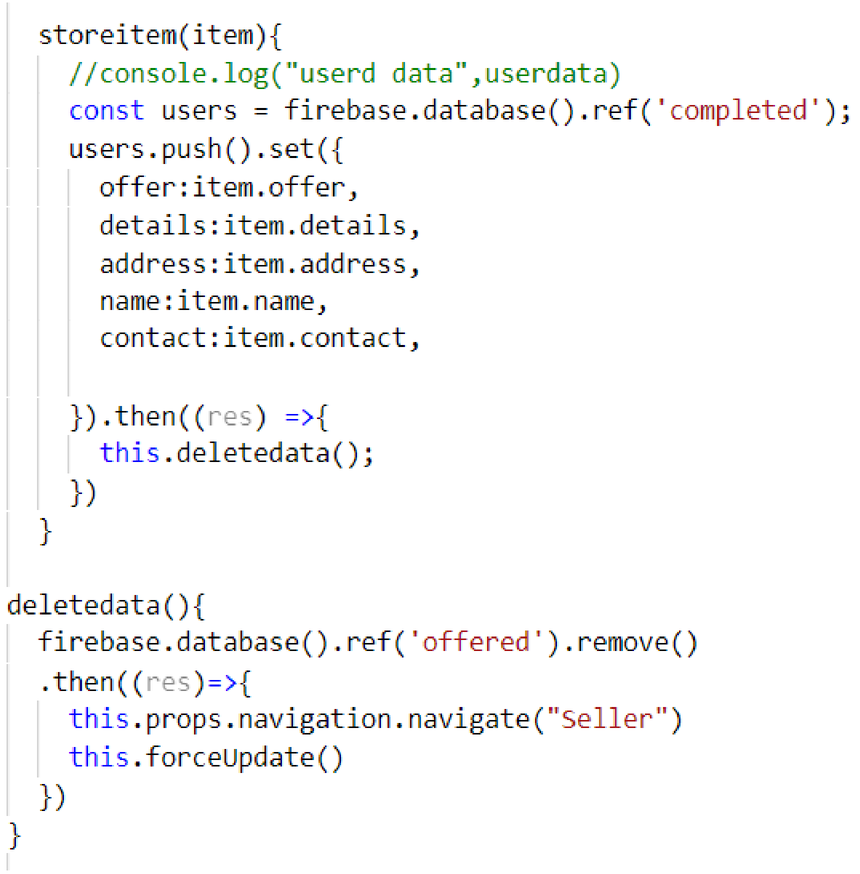
```
const myitems = this.state.shopsArr.map(item => {
return (

      <Card style={{borderRadius: 15, padding: 10}}>
        <CardItem button onPress={()=> this.props.navigation.navigate("PlaceOr")}>
          <Left>
            <Text style={{fontSize: 17, fontWeight: 'bold', color: '#52ab98'}}>{item.shopName}</Text>
          </Left>
          <Right>
            <Mticon name="star" size={25} color='#ffcc00' />
            <Text>{item.rating}</Text>
          </Right>
        </CardItem>
      </Card>
)
})

const markers = this.state.shopsArr.map(item => {
  var lat = parseFloat(item.coordinateslat);
  var long = parseFloat(item.coordinateslong);
  return (
    <Marker
        coordinate={{
          latitude: lat,
          longitude: long,
        }}
        title={item.shopName}
        description="For your 100 plumbing needs"
      />

  )
  })
```

7.8 Image Picker



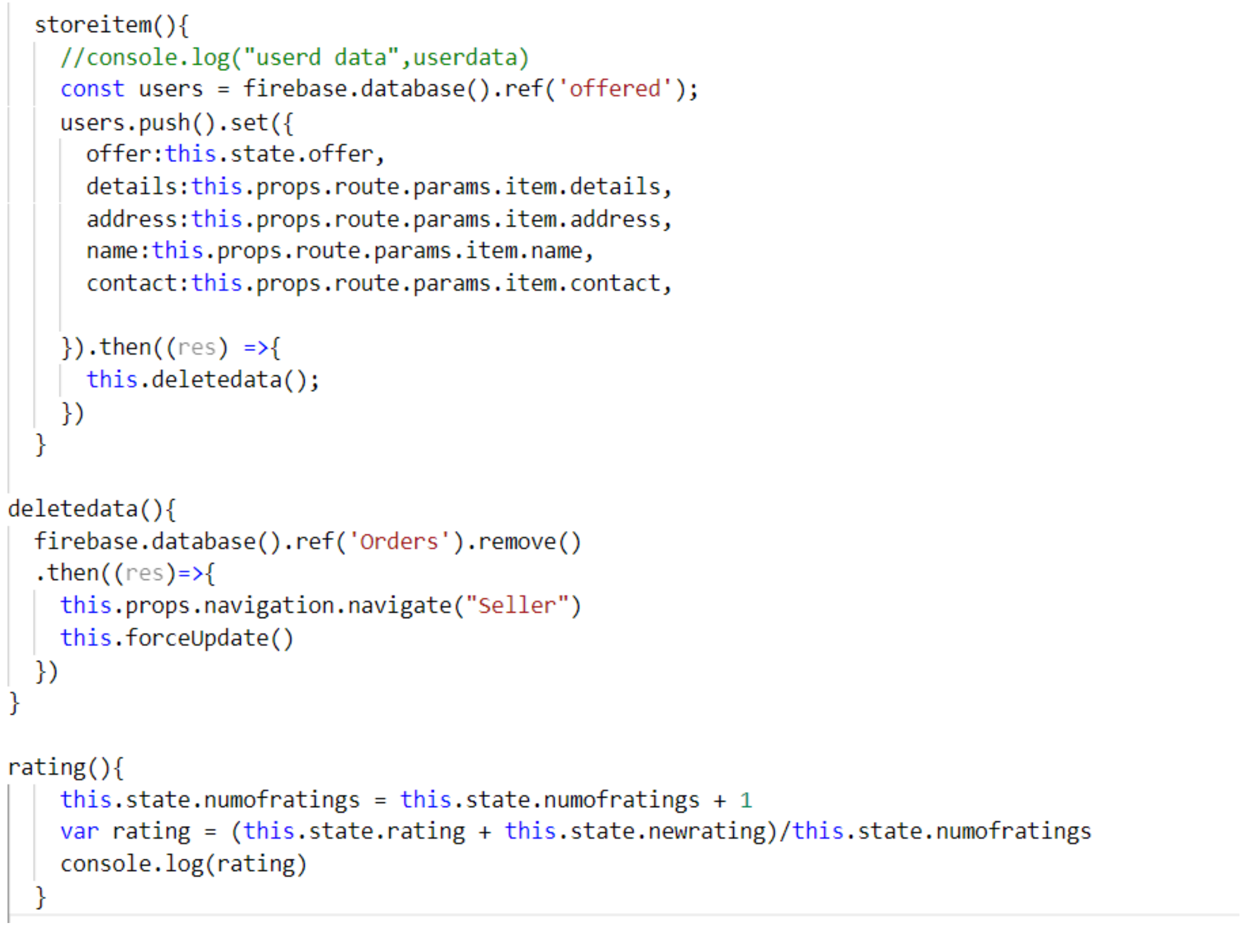
```
pickAndUploadImage=()=>{
  launchImageLibrary({quality:0.5},(fileobj)=>{


    const uploadTask = storage().ref().child(`/orderpic/${Date.now()}`).putFile(fileobj.assets[0].uri)

    uploadTask.on('state_changed',
      (snapshot) => {
    const progress = (snapshot.bytesTransferred / snapshot.totalBytes) * 100;
    if(progress==100) alert('Image Has Been Uploaded Successfully')

  },
  (error) => {
    alert('Error While Uploading Image')
  },
  () => {
    getDownloadURL(uploadTask.snapshot.ref).then((downloadURL) => {
      console.log('File available at', downloadURL);
      setImage(downloadURL);
      this.state.imgpath = fileobj.assets[0].uri;
      console.log(imgpath);
    });
  }
);
  })
}
```

7.9 Current Orders



```
    storeitem(item){
      //console.log("userd data",userdata)
      const users = firebase.database().ref('completed');
      users.push().set({
        offer:item.offer,
        details:item.details,
        address:item.address,
        name:item.name,
        contact:item.contact,

      }).then((res) =>{
        this.deletedata();
      })
    }

  deletedata(){
    firebase.database().ref('offered').remove()
    .then((res)=>{
      this.props.navigation.navigate("Seller")
      this.forceUpdate()
    })
  }
```

7.10 Rating



```
  storeitem(){
    //console.log("userd data",userdata)
    const users = firebase.database().ref('offered');
    users.push().set({
      offer:this.state.offer,
      details:this.props.route.params.item.details,
      address:this.props.route.params.item.address,
      name:this.props.route.params.item.name,
      contact:this.props.route.params.item.contact,

    }).then((res) =>{
      this.deletedata();
    })
  }

deletedata(){
  firebase.database().ref('Orders').remove()
  .then((res)=>{
    this.props.navigation.navigate("Seller")
    this.forceUpdate()
  })
}

rating(){
    this.state.numofratings = this.state.numofratings + 1
    var rating = (this.state.rating + this.state.newrating)/this.state.numofratings
    console.log(rating)
  }
```

8. Appendix A: Test Cases

8.1 Test case 1

Title	Incorrect Login
Pre-Condition	User must have initiated the application
Description	User should not be able to login with incorrect login details
Assumption	User enters incorrect credentials
Post Condition	Error message appears informing user of incorrect credentials
Pass/Fail	PASS
QA Testing Engineer	Hasnain Haider

8.2 Test case 2

Title	Correct Login
Pre-Condition	User must have initiated the application
Description	User should be able to login with correct login details
Assumption	User enters correct credentials
Post Condition	System redirects to Main menu (User type selection)
Pass/Fail	PASS
QA Testing Engineer	Hasnain Haider

8.3 Test case 3

Title	Looking for a service
Pre-Condition	User must be logged in
Description	User selects Looking for a service option
Assumption	User has basic know-how of the application
Post Condition	System redirects to Technician selection page (Technician type selection)
Pass/Fail	PASS
QA Testing Engineer	Hasnain Haider

8.4 Test case 4

Title	Providing a service
Pre-Condition	User must be logged in
Description	User selects providing a service option
Assumption	User has the basic know-how of the application
Post Condition	System redirects to Seller screen (Orders)
Pass/Fail	PASS
QA Testing Engineer	Hasnain Haider

8.5 Test case 5

Title	Place Order
Pre-Condition	User must have selected desired technician
Description	User enters job details and presses place order button
Assumption	User can communicate the issue through text
Post Condition	System posts job to database and creates an order card in seller pending orders tab with order details
Pass/Fail	PASS
QA Testing Engineer	Malik Muhammad Basim Mansoor

8.6 Test case 6

Title	Contact Customer
Pre-Condition	User must have pending order with correct contact details
Description	Seller presses contact customer button on order card
Assumption	User has an active cell phone number and basic know-how of the application
Post Condition	System redirects to Call feature of the android device and initiates a phone call with customer
Pass/Fail	PASS
QA Testing Engineer	Malik Muhammad Basim Mansoor

8.7 Test case 7

Title	Send Offer
Pre-Condition	User must have pending order
Description	Seller enters job price on order card
Assumption	Seller makes a professional and competitive offer
Post Condition	System stores offer and shows it on Buyer end
Pass/Fail	PASS
QA Testing Engineer	Malik Muhammad Basim Mansoor

8.8 Test case 8

Title	Accept Offer
Pre-Condition	User must have Received Offer of order
Description	Buyer Accepts job Offer price on order card
Assumption	Buyer Agrees with pricing offer from seller
Post Condition	System Moves card from Pending order to current Order
Pass/Fail	PASS
QA Testing Engineer	Malik Muhammad Basim Mansoor

8.9 Test case 9

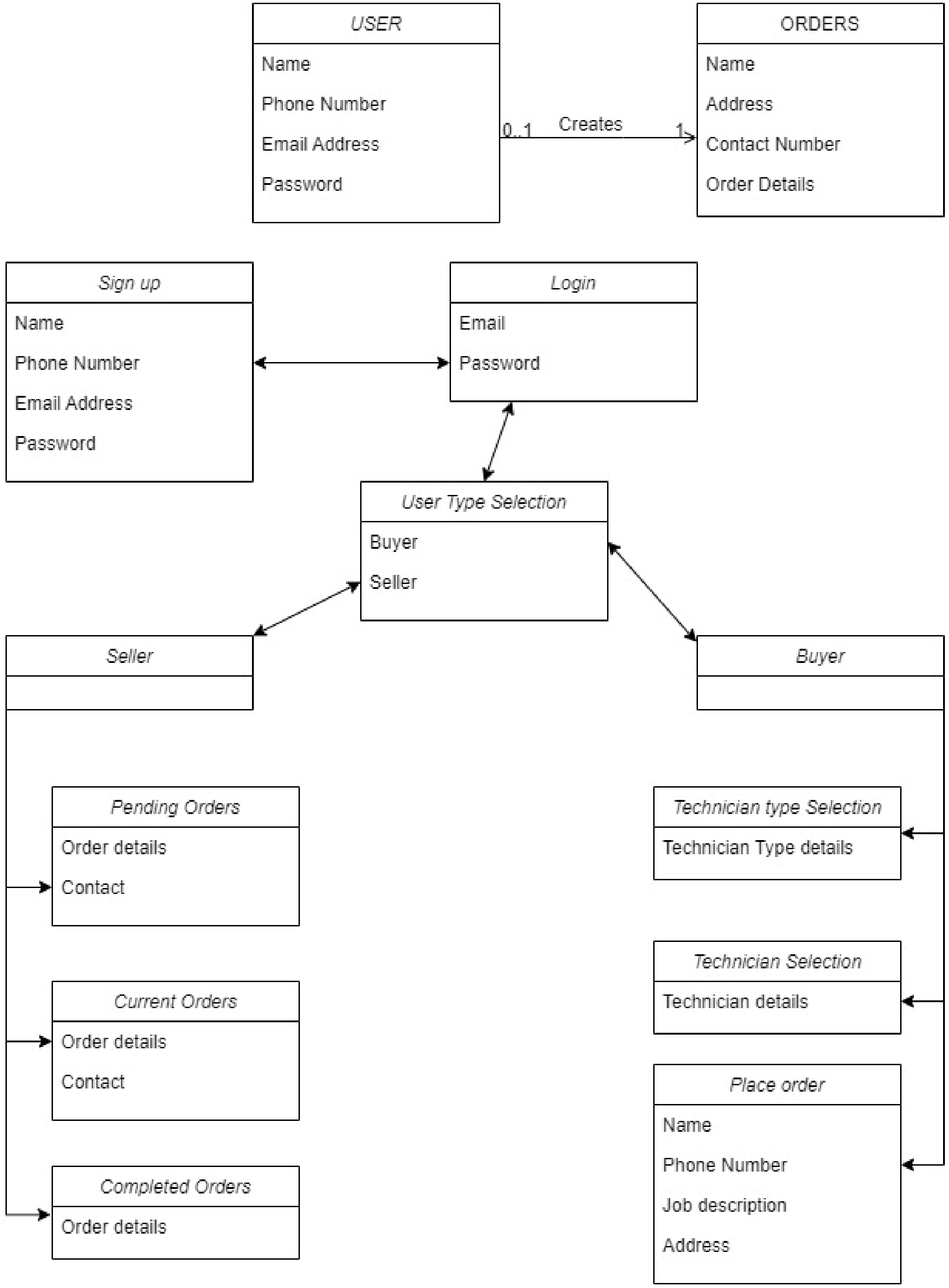
Title	Reject Offer
Pre-Condition	User must have Received Offer of order
Description	Buyer Rejects job Offer price on order card
Assumption	Buyer Disagrees with pricing offer from seller
Post Condition	System Deletes pending order from both users
Pass/Fail	PASS
QA Testing Engineer	Malik Muhammad Basim Mansoor

8.10 Test case 10

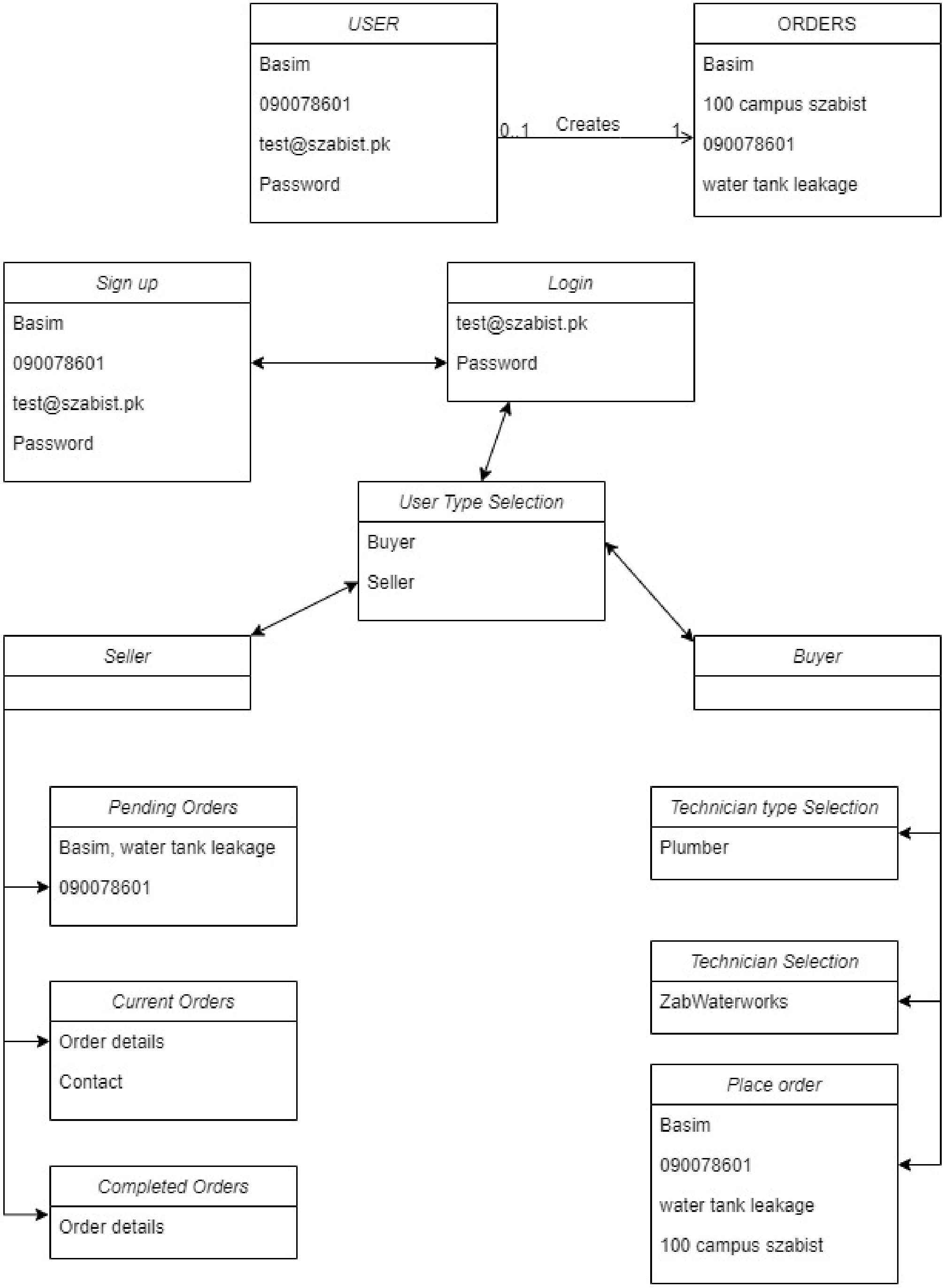
Title	Image Upload
Pre-Condition	User Must be on Place order Page
Description	User uploads image
Assumption	User has image to upload
Post Condition	System stores image in Firebase storage
Pass/Fail	PASS
QA Testing Engineer	Malik Muhammad Basim Mansoor

9. Appendix B: Diagrams

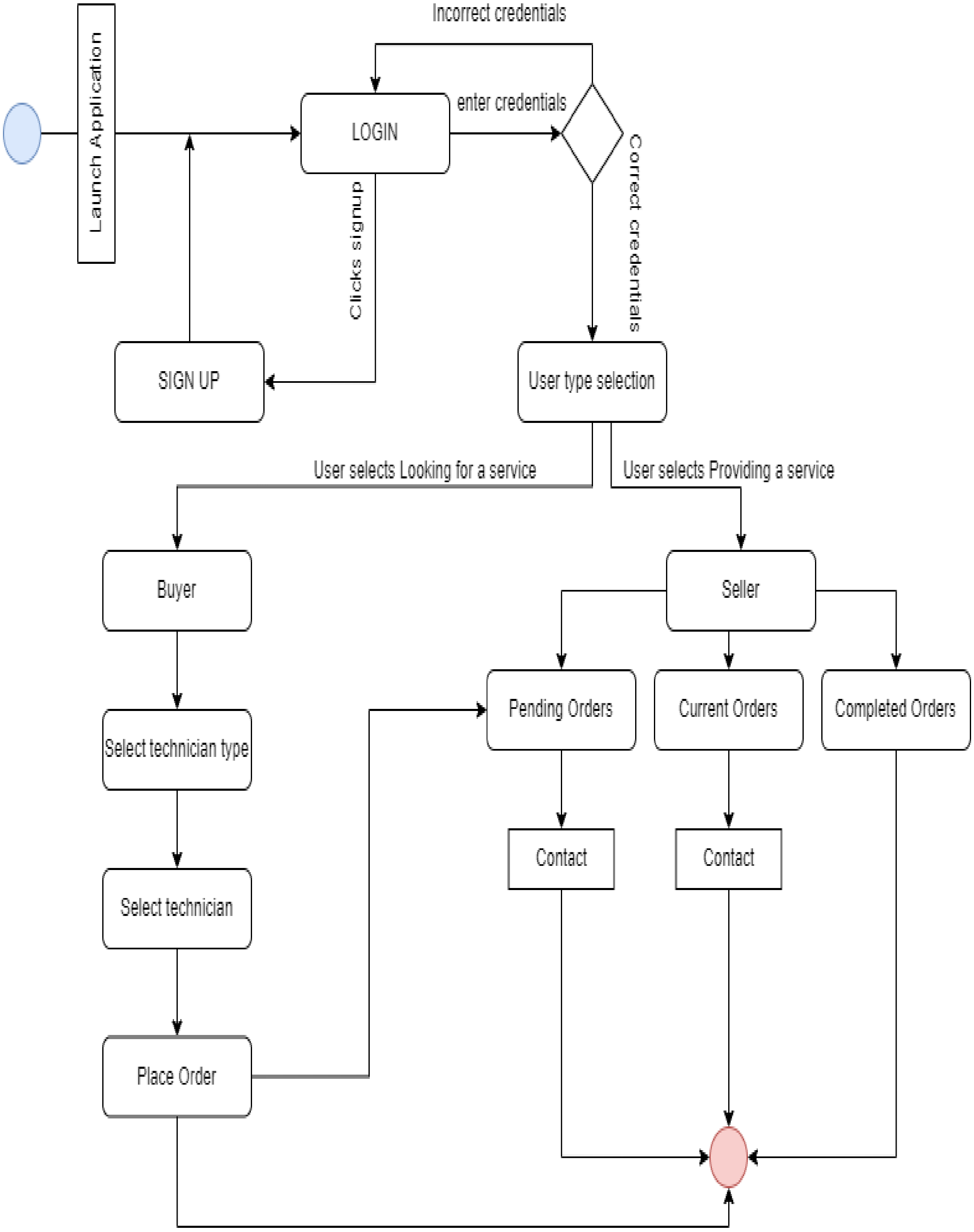
9.1 Use Case Diagram



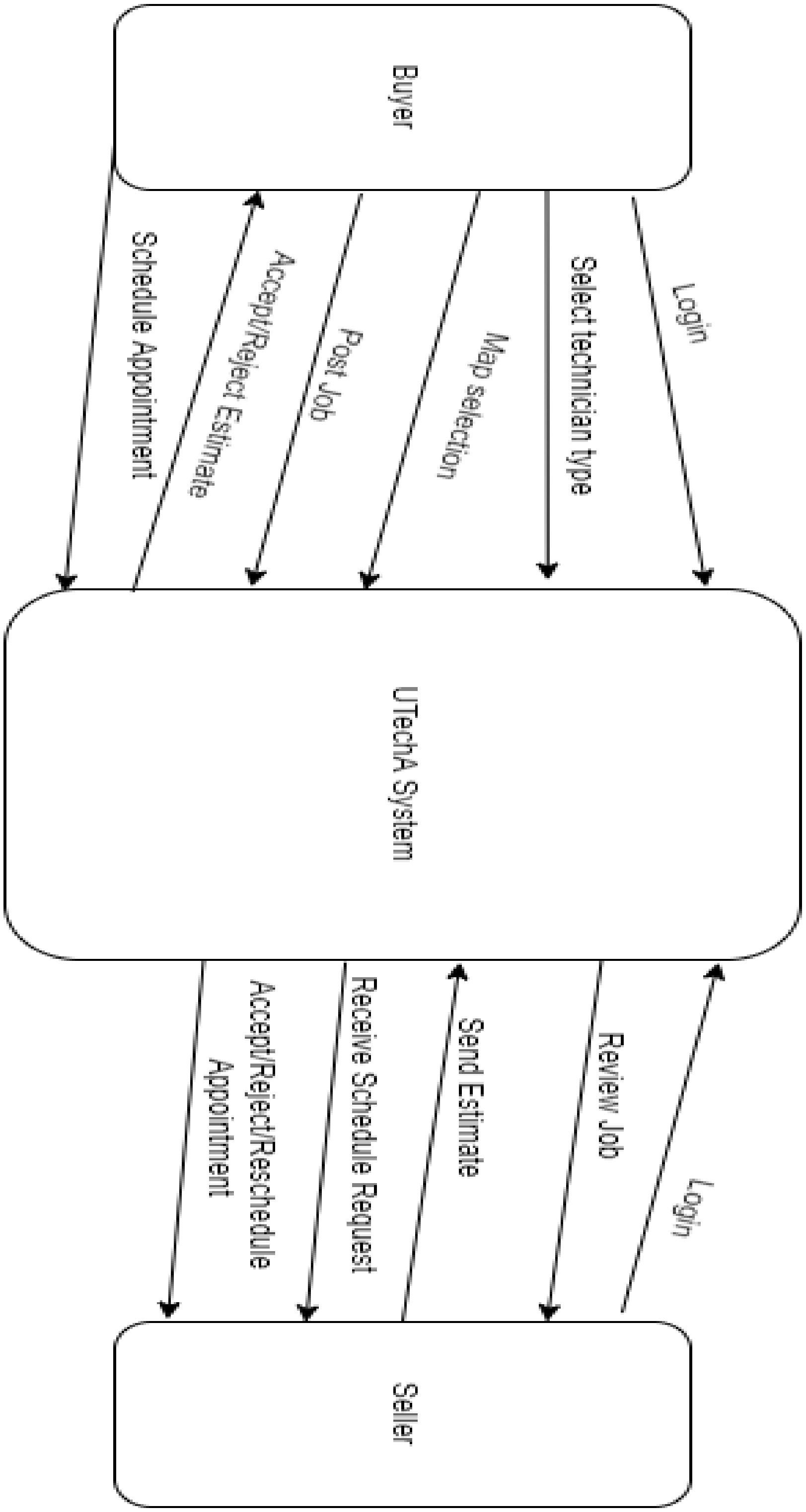
9.2 Object Diagram



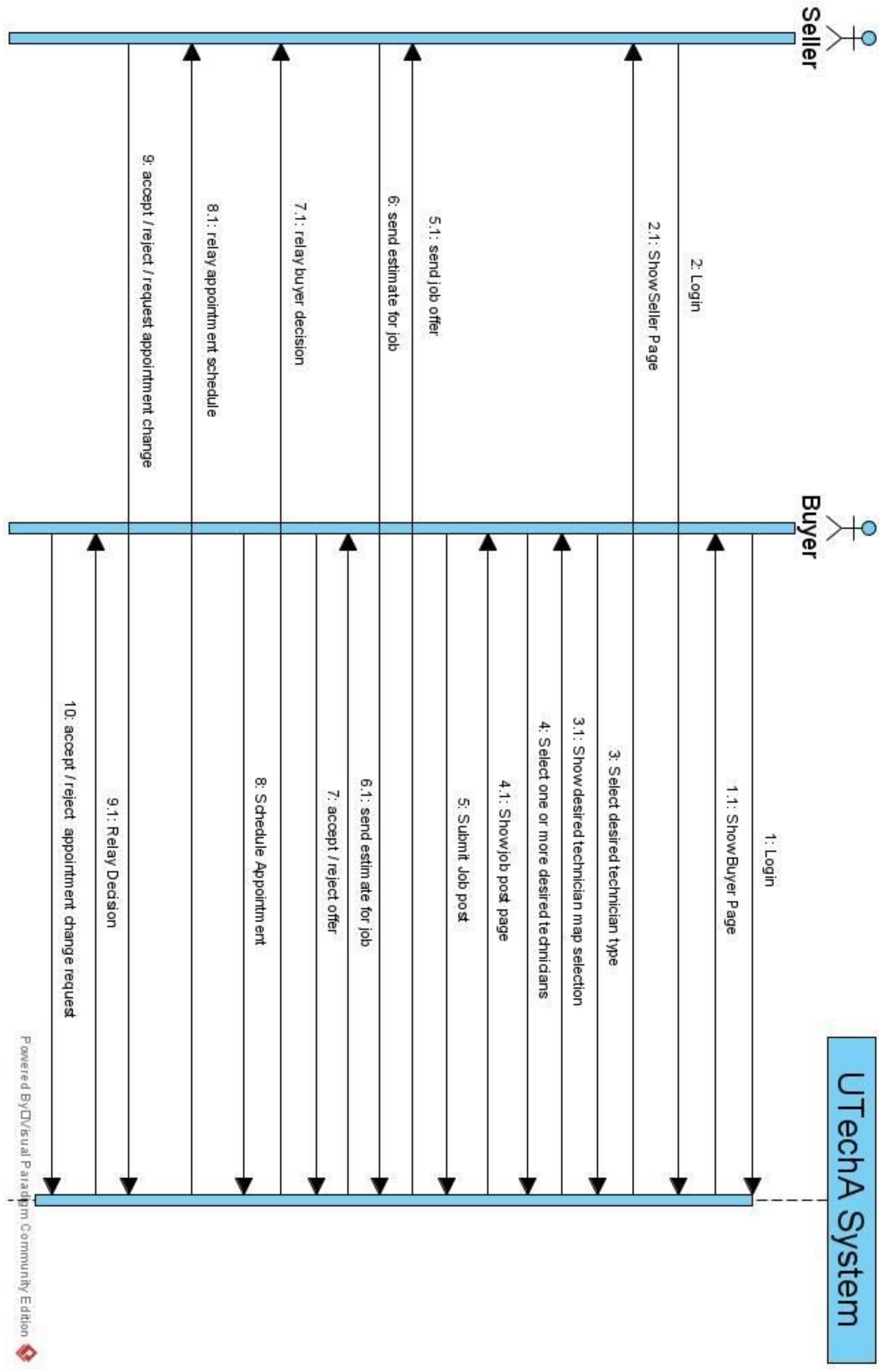
9.3 State chart diagram



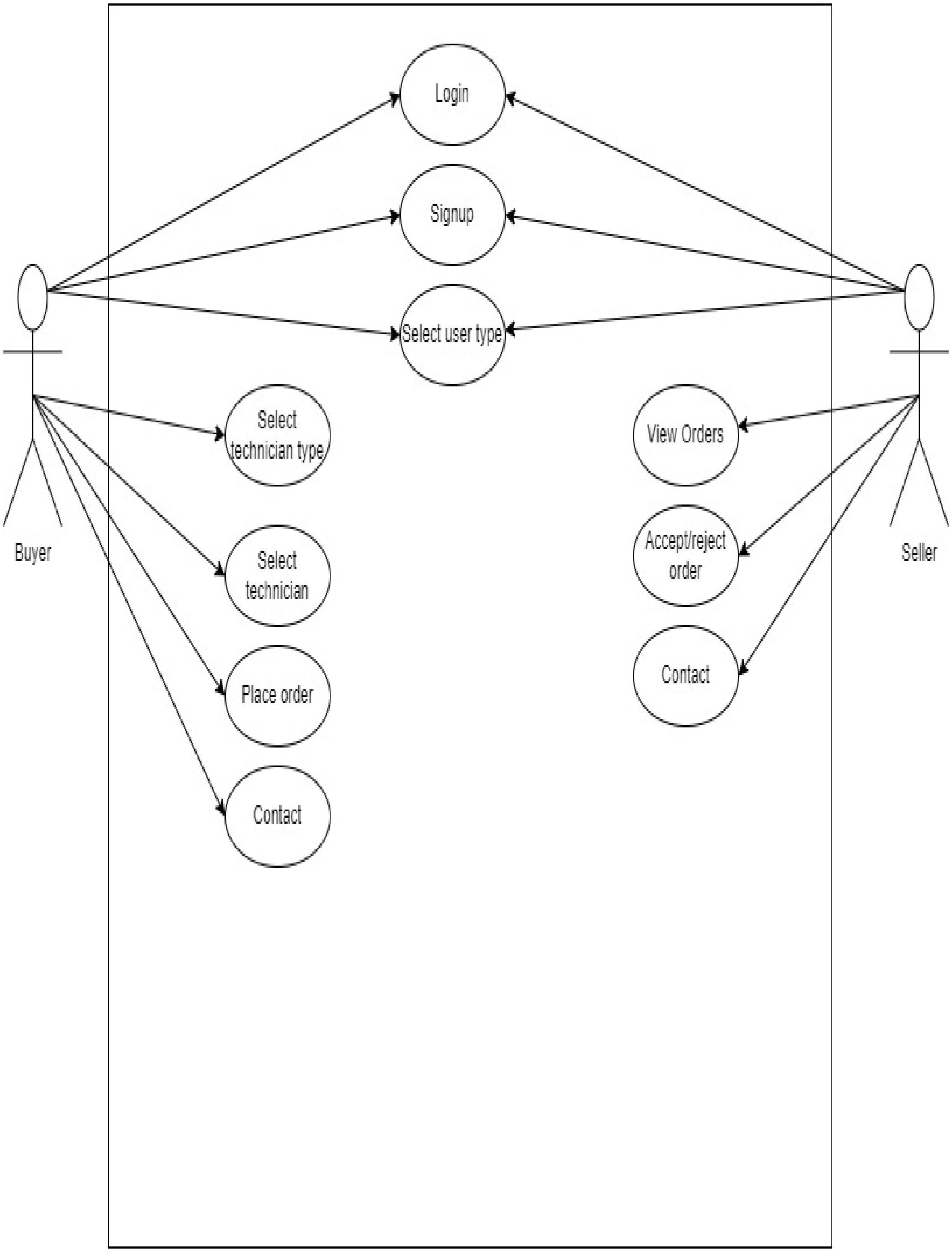
9.4 Context Diagram



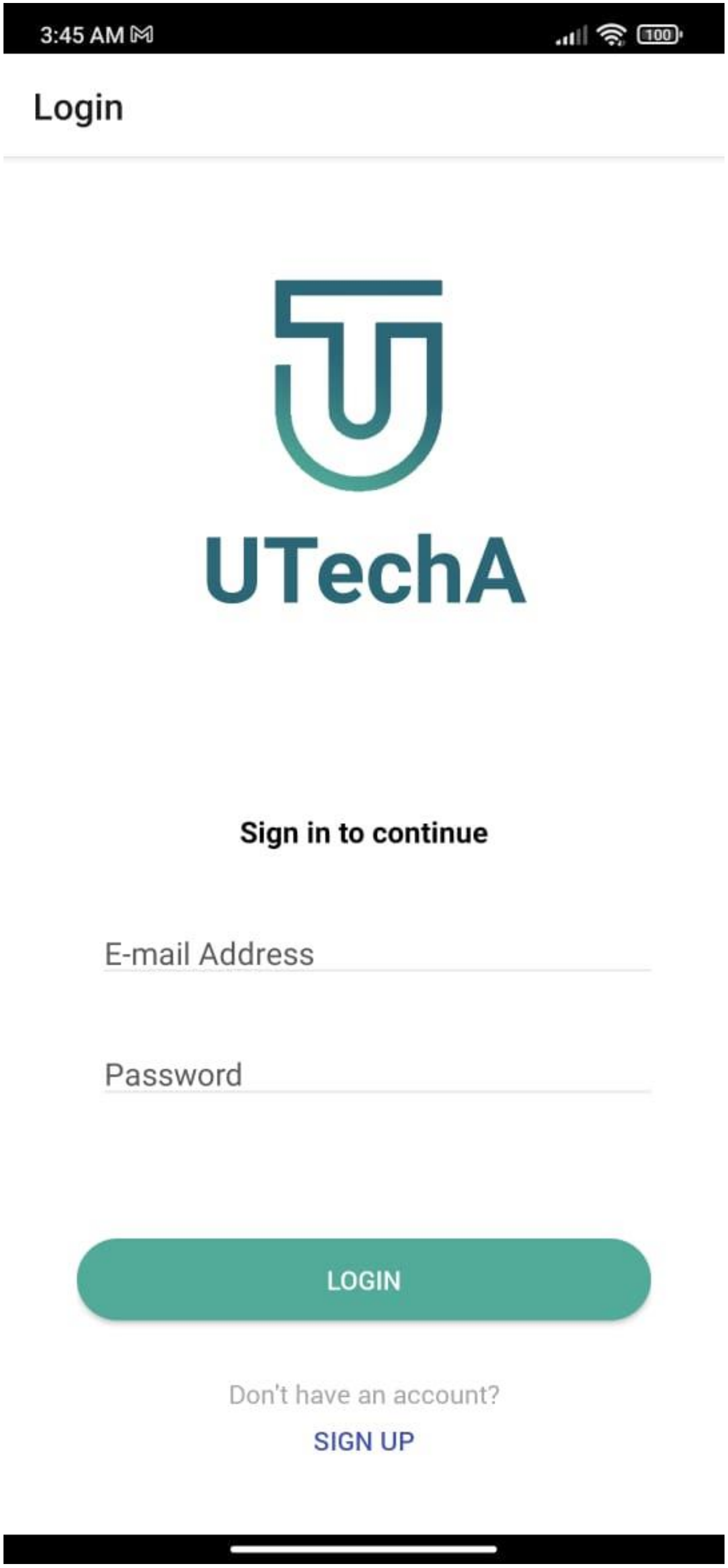
9.5 System Sequence Diagram


Seller
Buyer
UTechA System
1: Login
1.1: ShowBuyer Page
2: Login
2.1: ShowSeller Page
3: Select desired technician type
3.1: Show desired technician map selection
4: Select one or more desired technicians
4.1: Show job post page
5: Submit Job post
5.1: send job offer
6: send estimate for job
6.1: send estimate for job
7: accept / reject offer
7.1: relay buyer decision
8: Schedule Appointment
8.1: relay appointment schedule
9: accept / reject / request appointment change
9.1: Relay Decision
10: accept / reject appointment change request
Powered By Visual Paradigm Community Edition

9.6 Use Case


Login
Signup
Select user type
Select technician type
Select technician
Place order
Contact
View Orders
Accept/reject order
Contact
Buyer
Seller

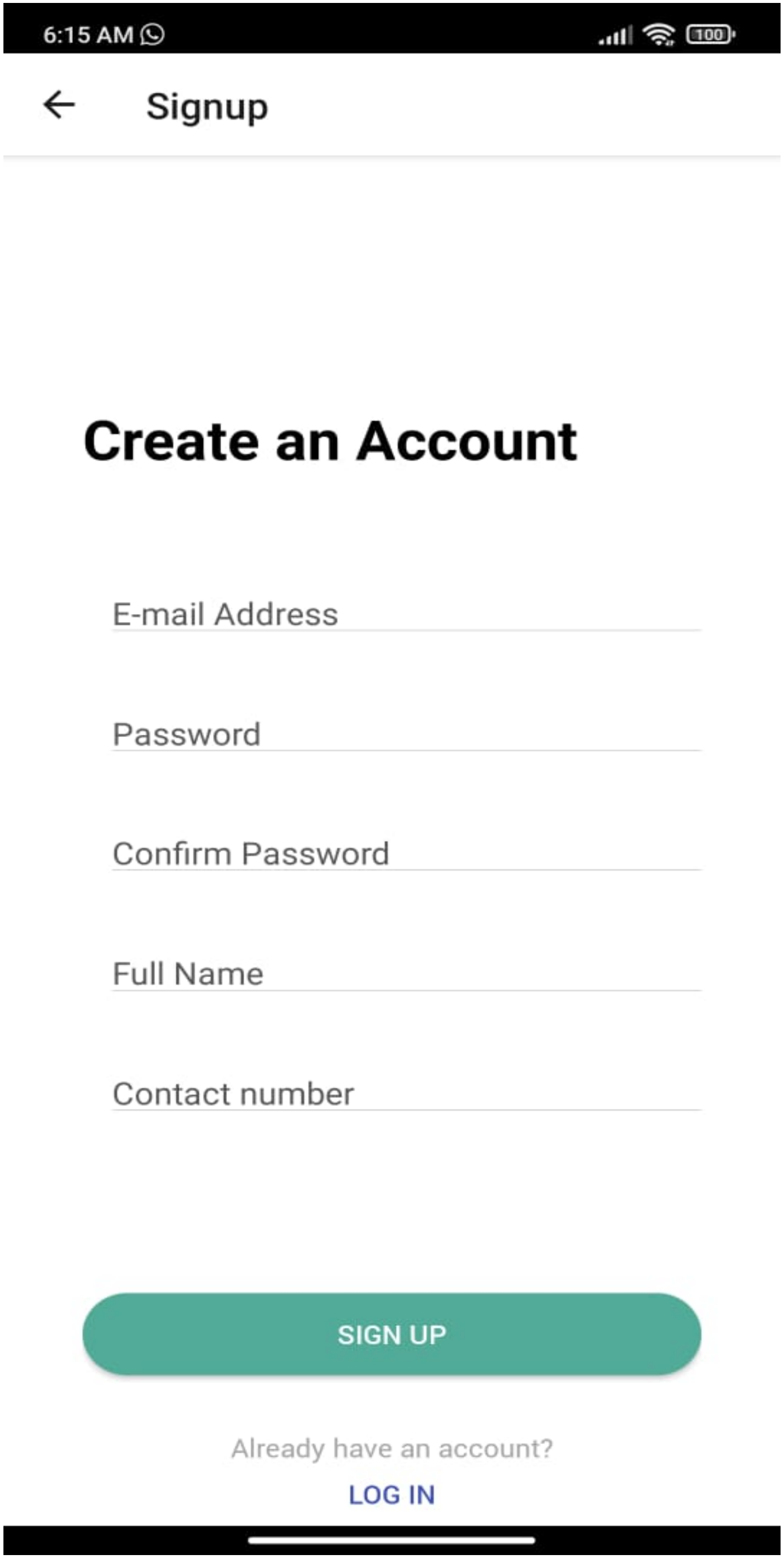
9.7 Login Screen



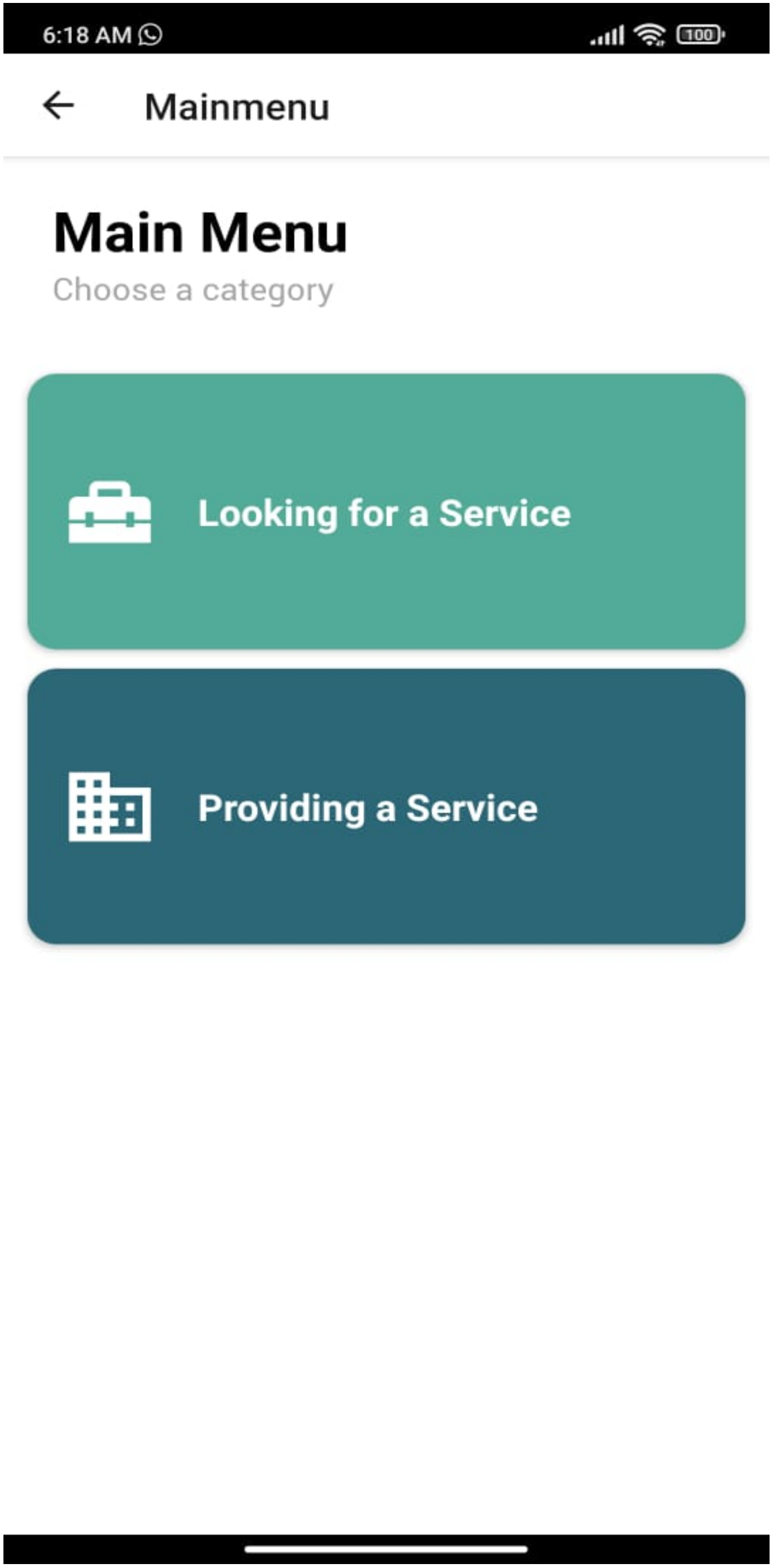
Sign in to continue

Don't have an account?

9.8 Sign Up Screen



9.9 User Type Selection Screen



6:18 AM

Mainmenu

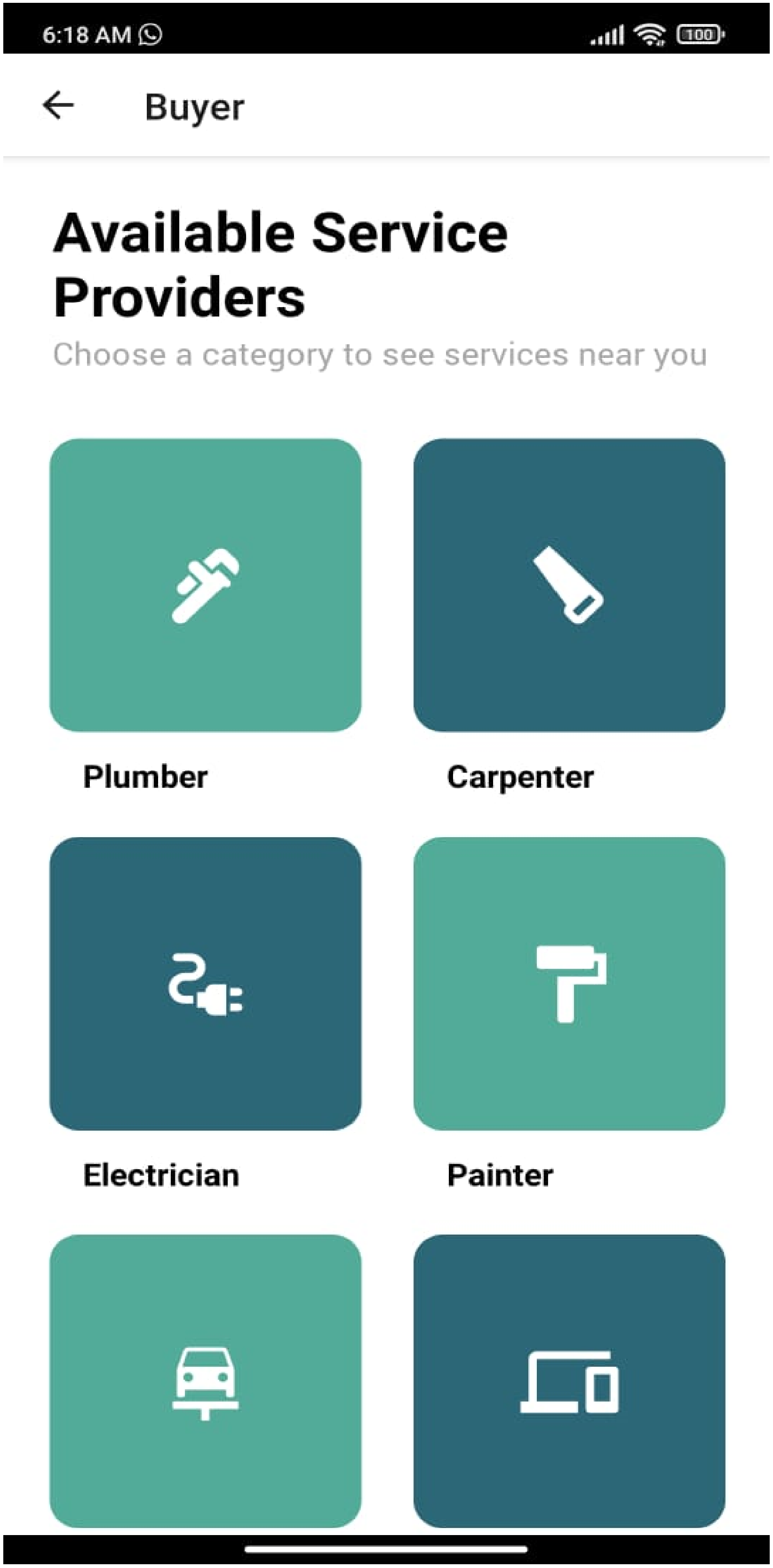
Main Menu

Choose a category

Looking for a Service

Providing a Service

9.10 Looking for a Service Screen



6:18 AM

Buyer

Available Service Providers

Choose a category to see services near you

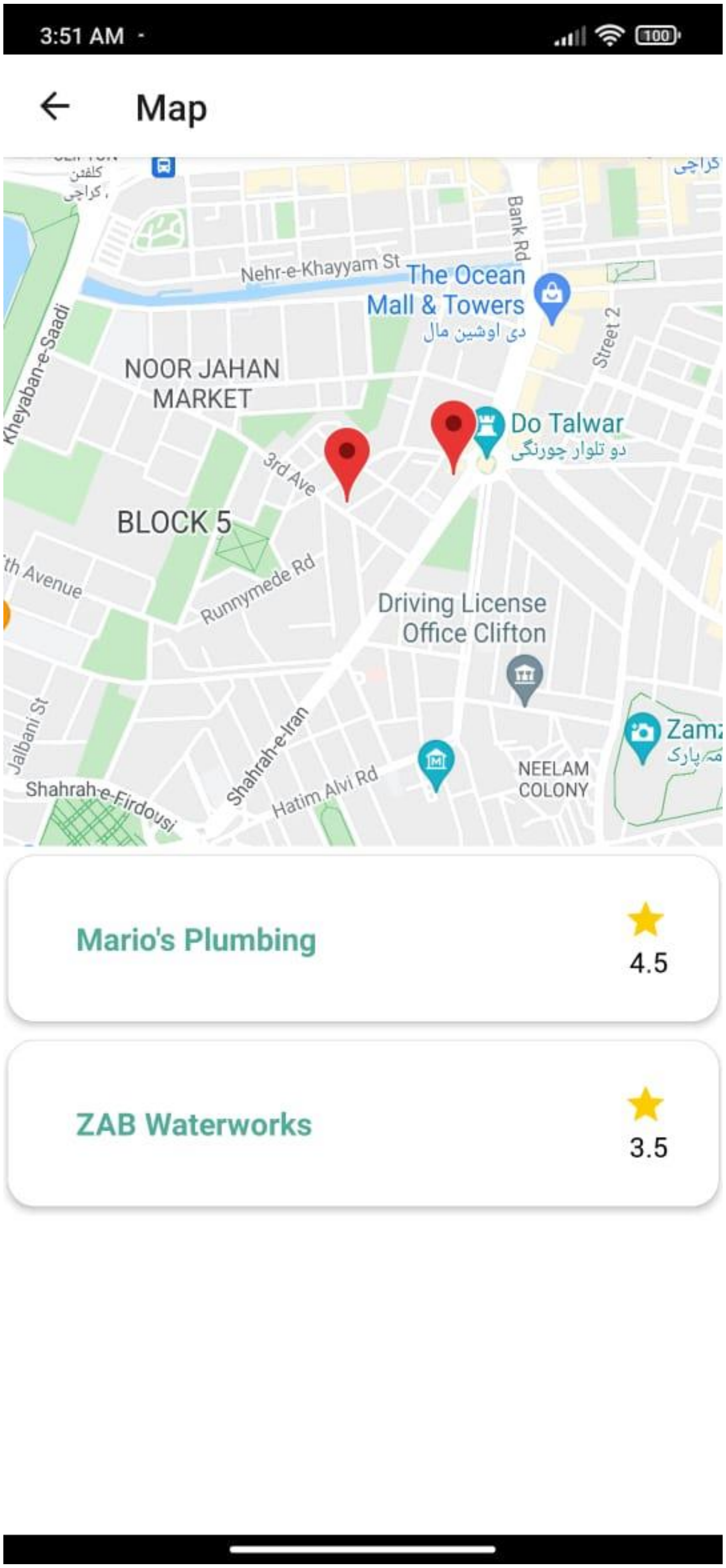
Plumber

Carpenter

Electrician

Painter

9.11 Technician Selection Screen

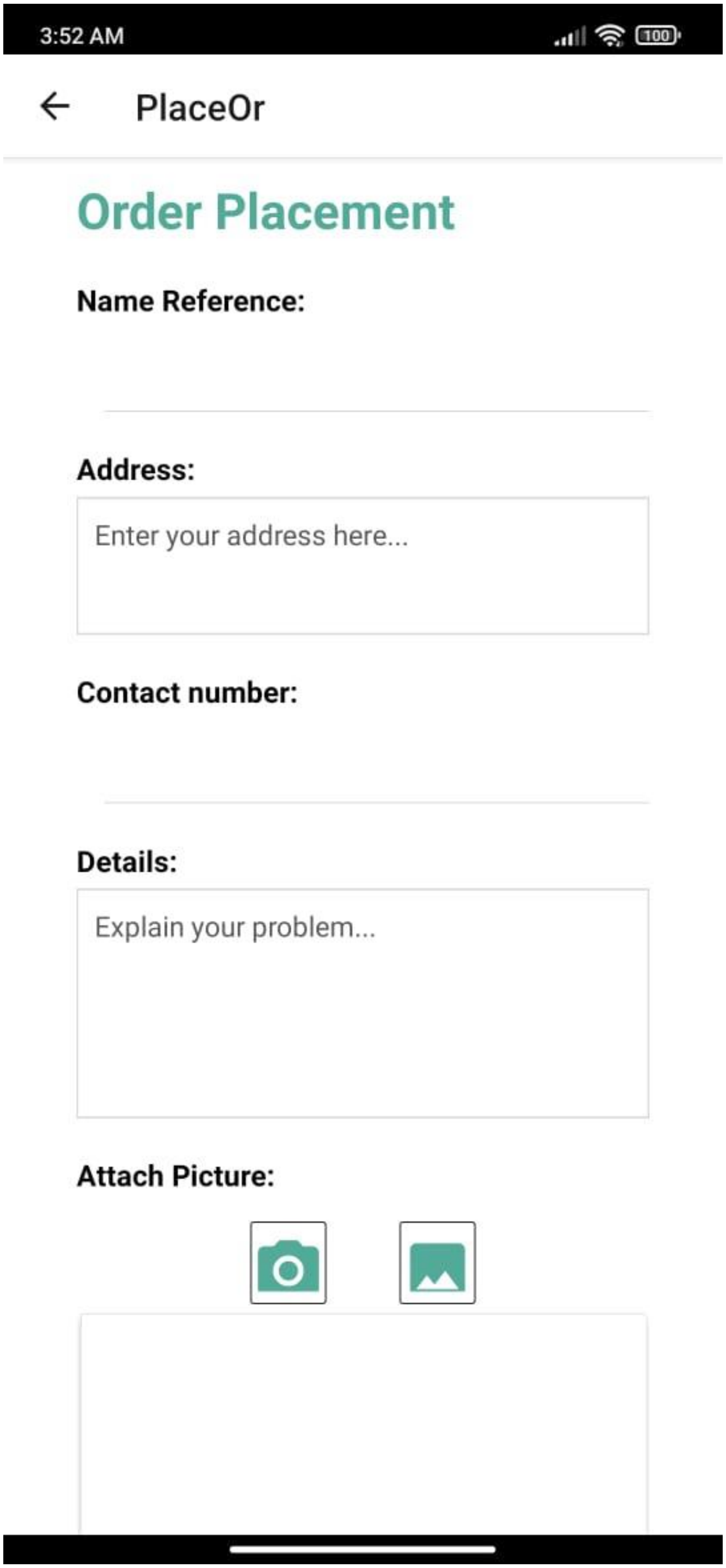


Map

Mario's Plumbing ★ 4.5

ZAB Waterworks ★ 3.5

9.12 Place Order Screen



Order Placement

Name Reference:

Address:

Enter your address here...

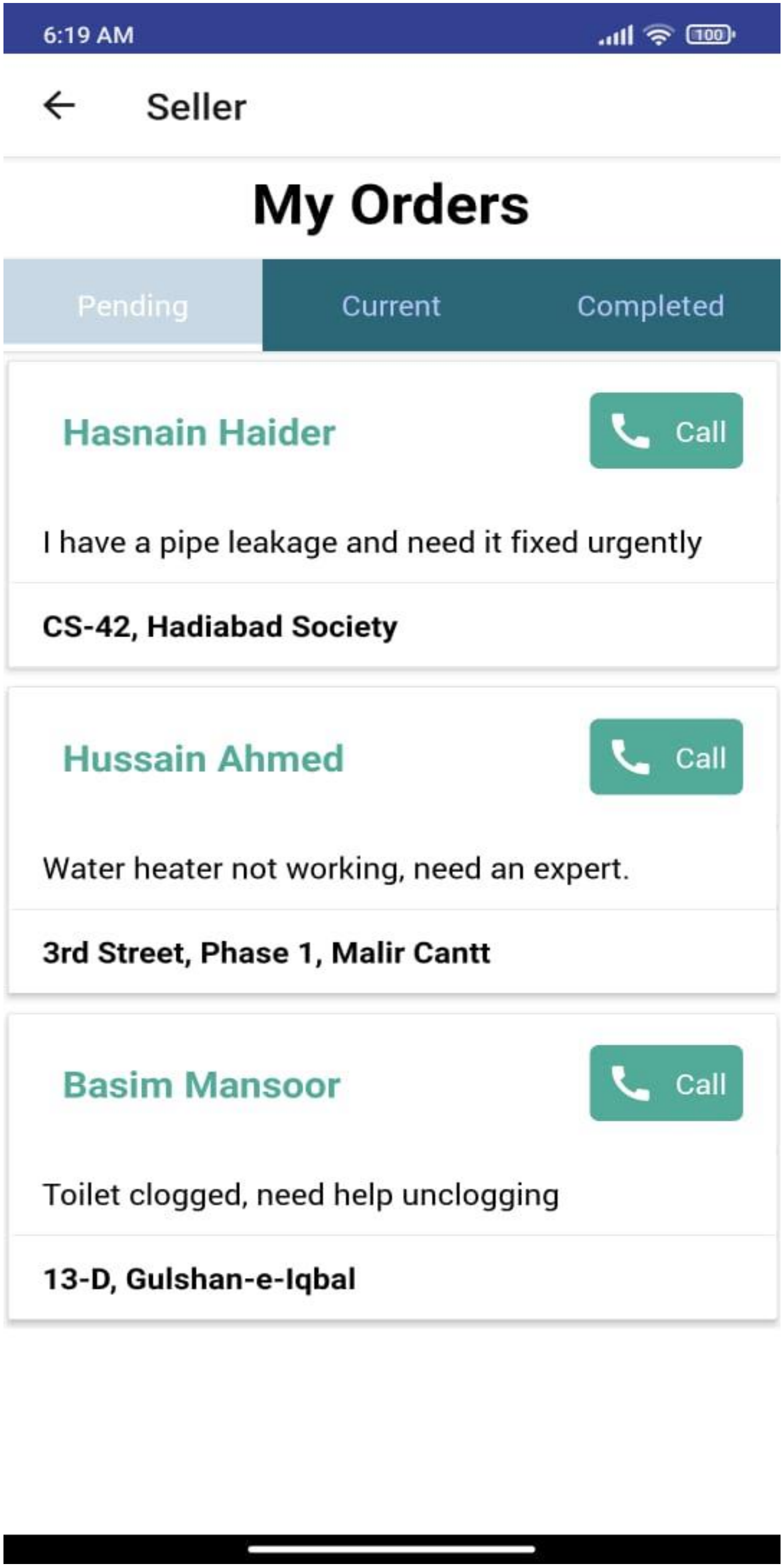
Contact number:

Details:

Explain your problem...

Attach Picture:

9.13 Providing a Service (Pending Orders Screen)



My Orders

Hasnain Haider

I have a pipe leakage and need it fixed urgently

CS-42, Hadiabad Society

Hussain Ahmed

Water heater not working, need an expert.

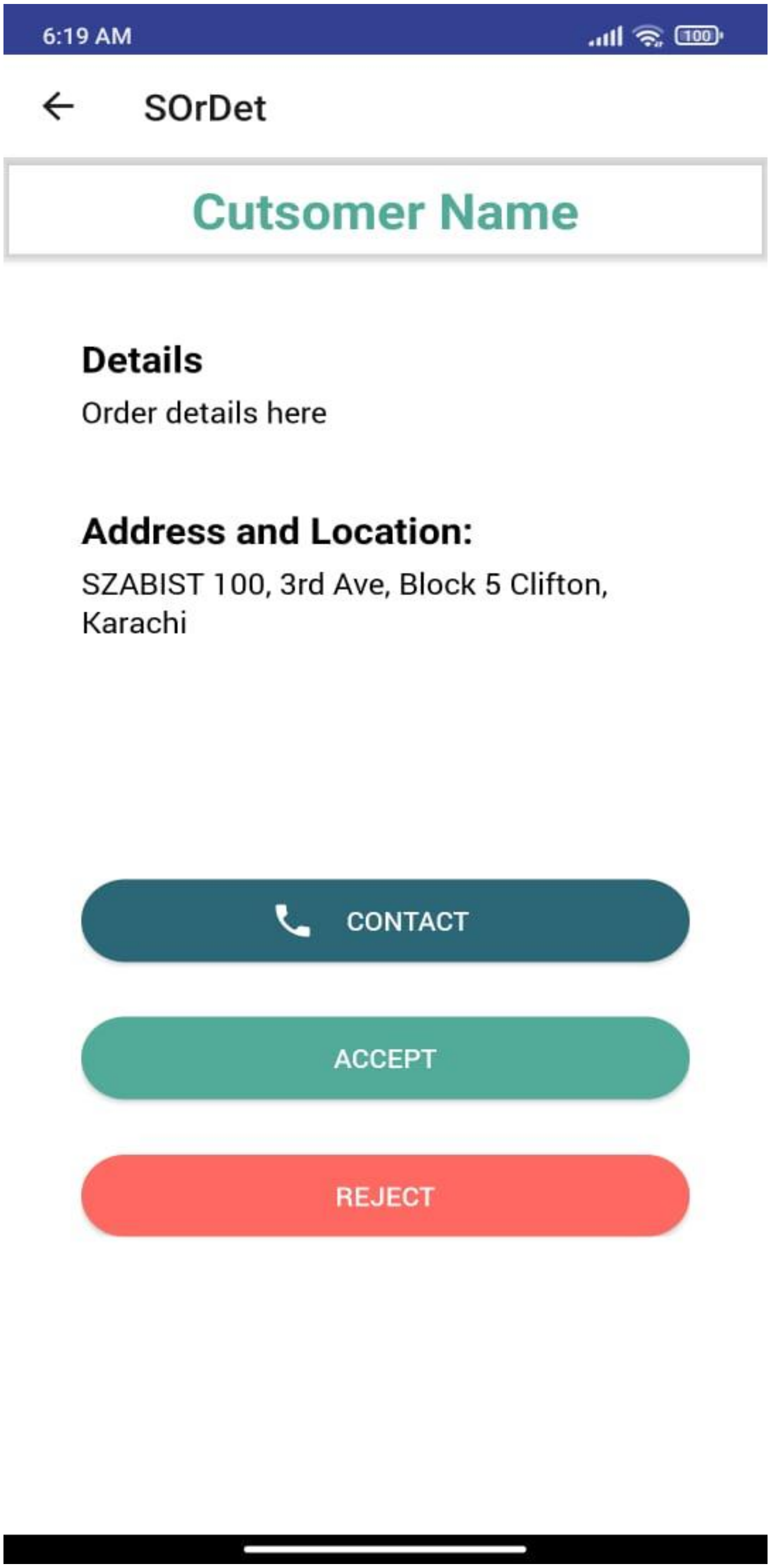
3rd Street, Phase 1, Malir Cantt

Basim Mansoor

Toilet clogged, need help unclogging

13-D, Gulshan-e-Iqbal

9.14 Seller Order Details Screen



Cutsomer Name

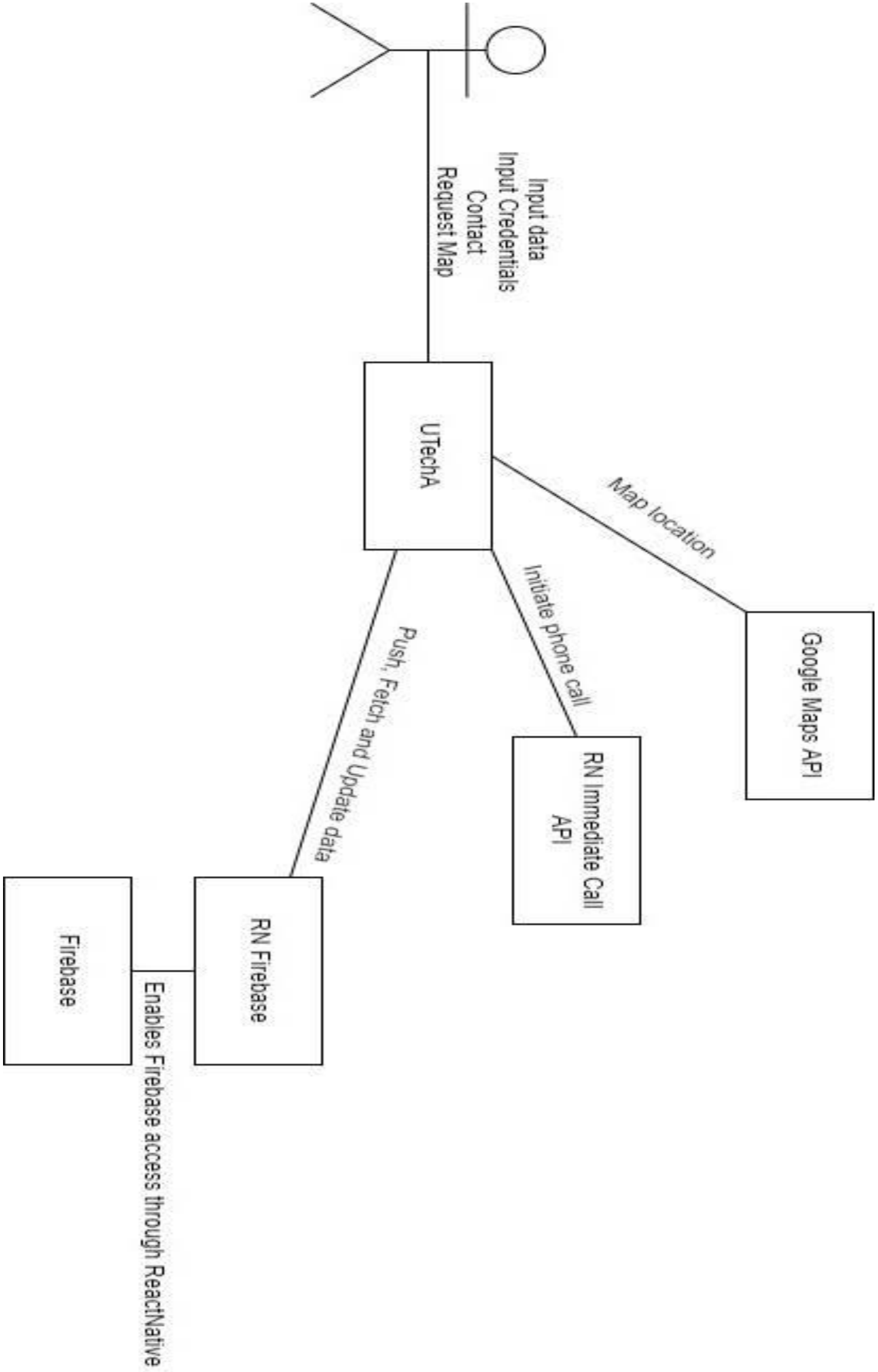
Details

Order details here

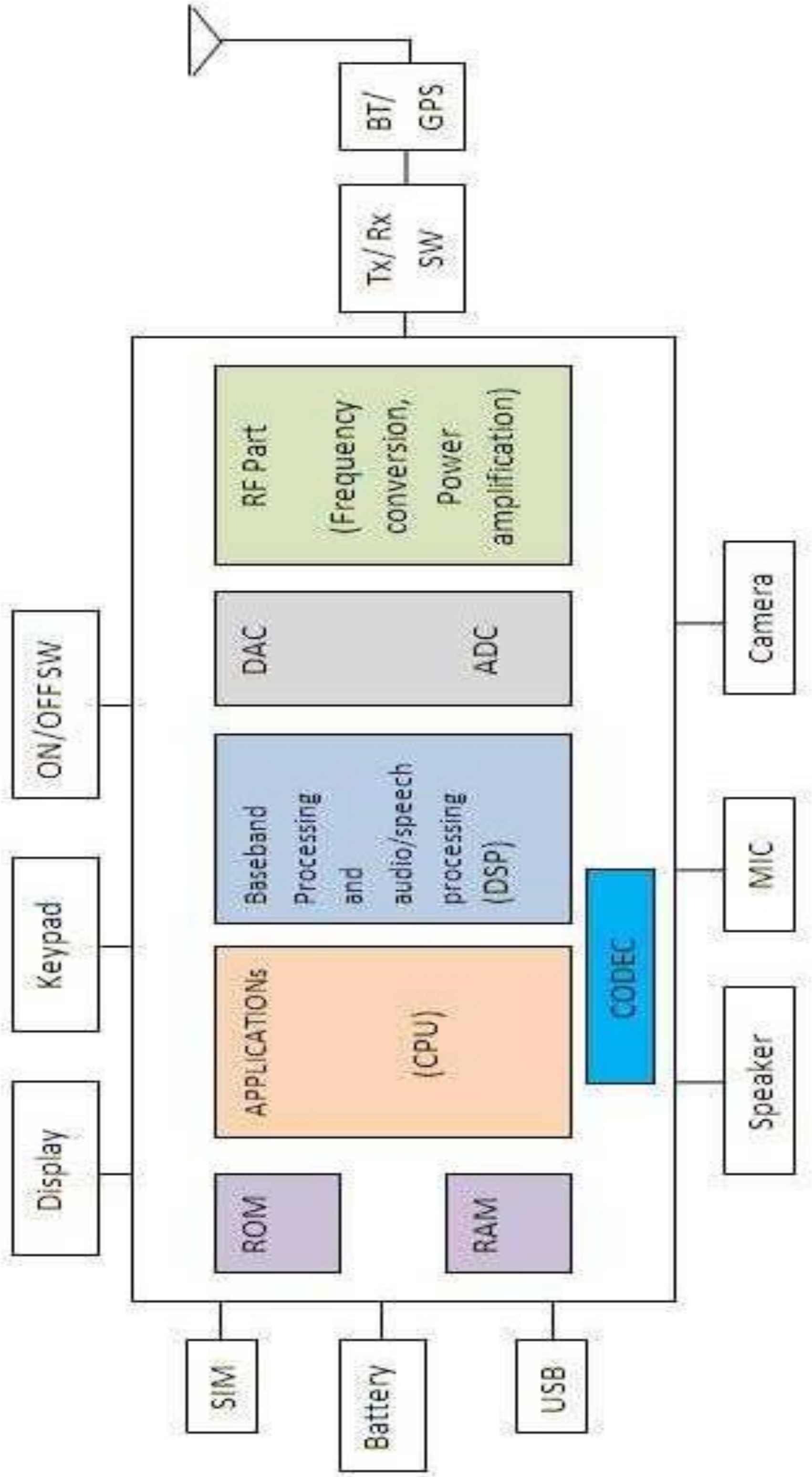
Address and Location:

SZABIST 100, 3rd Ave, Block 5 Clifton,
Karachi

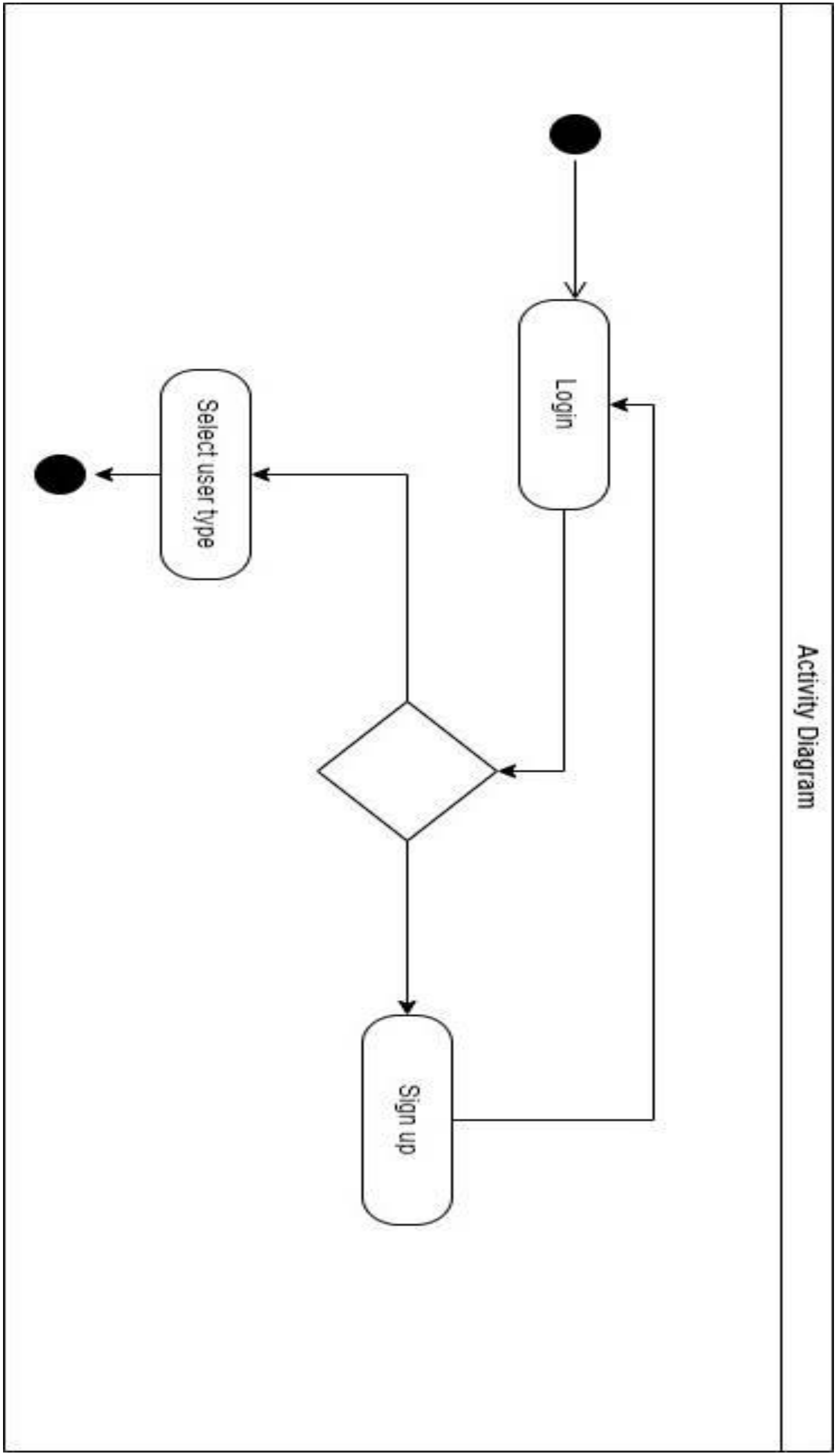
9.15 Collaboration Diagram



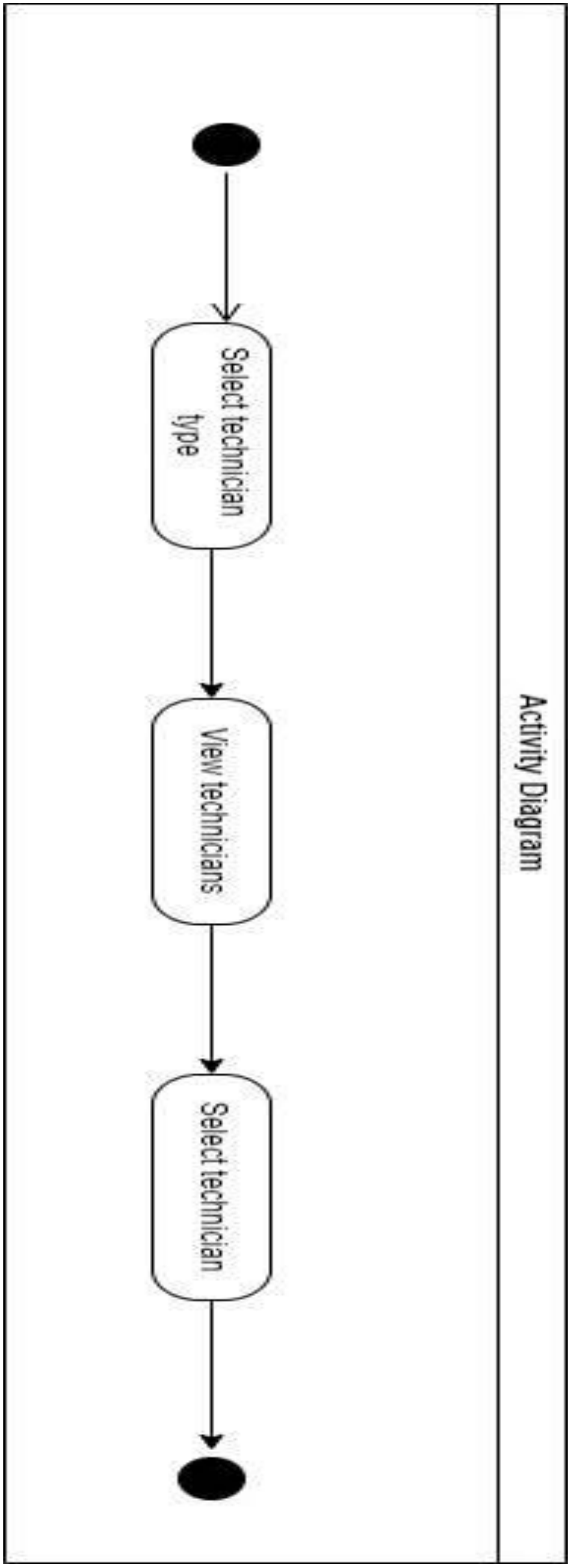
9.16 System Block Diagram (Smart Phone)


BT/ GPS
Tx/ Rx SW
RF Part (Frequency conversion, Power amplification)
DAC
ADC
Baseband Processing and audio/speech processing (DSP)
APPLICATIONs (CPU)
CODEC
ROM
RAM
ON/OFF SW
Keypad
Display
Camera
MIC
Speaker
SIM
Battery
USB

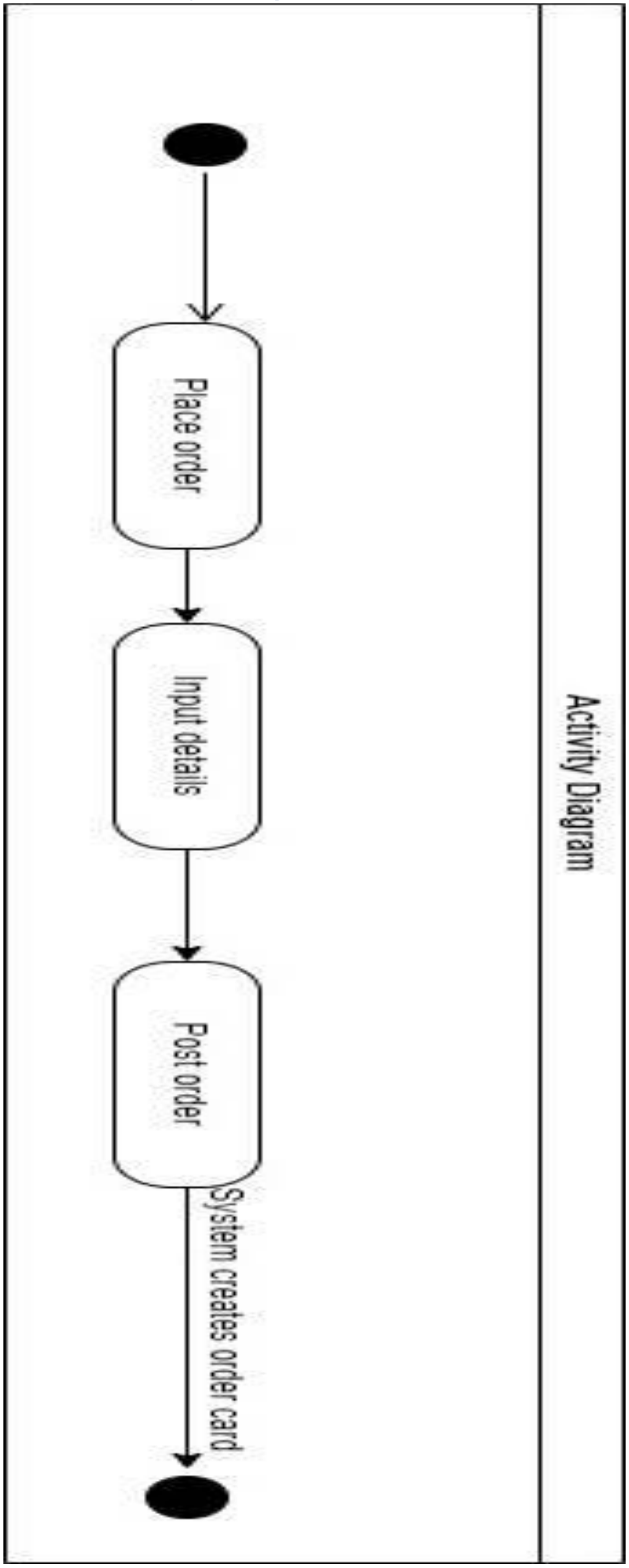
9.17 Activity Diagram – Login



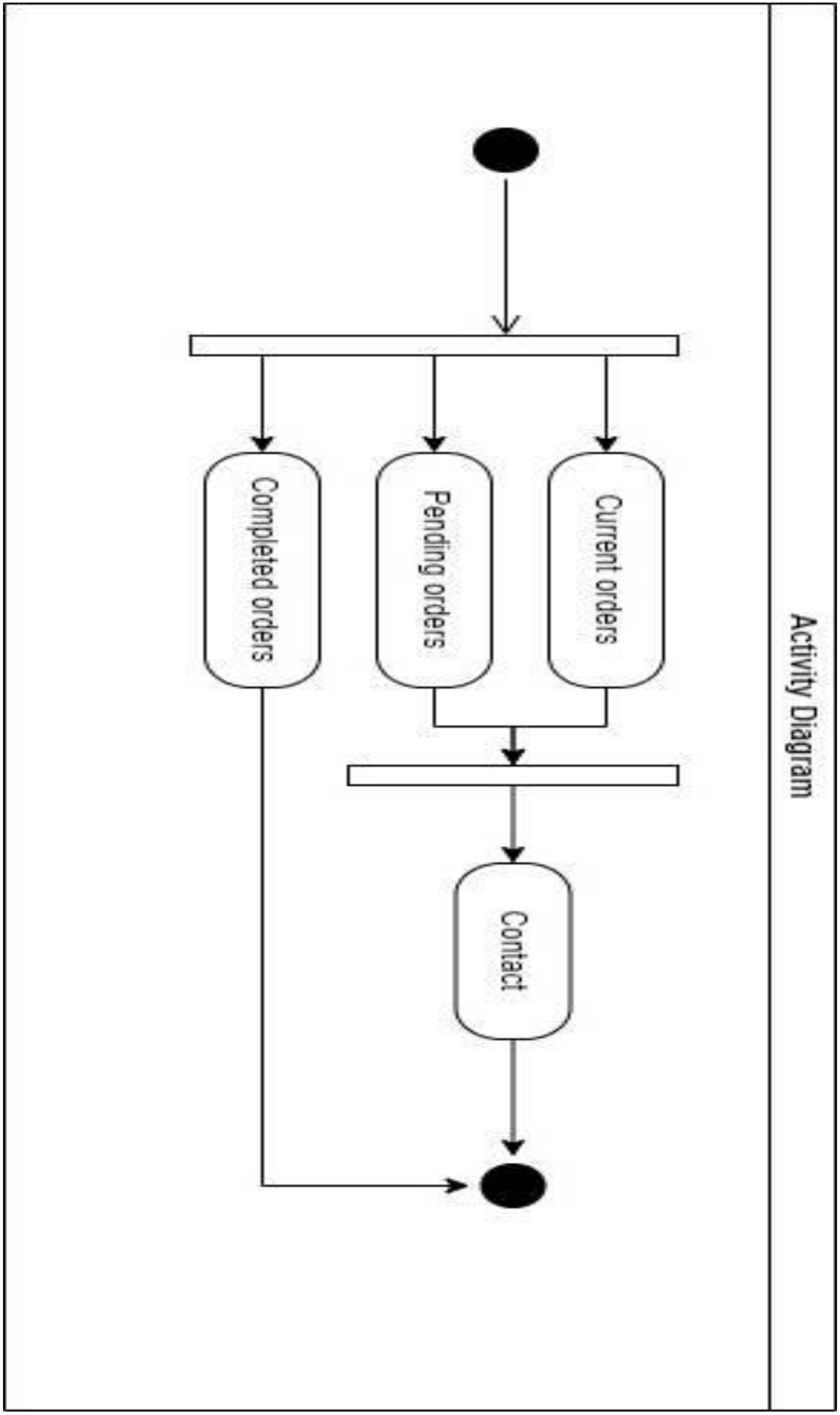
9.18 Activity Diagram – User type Buyer



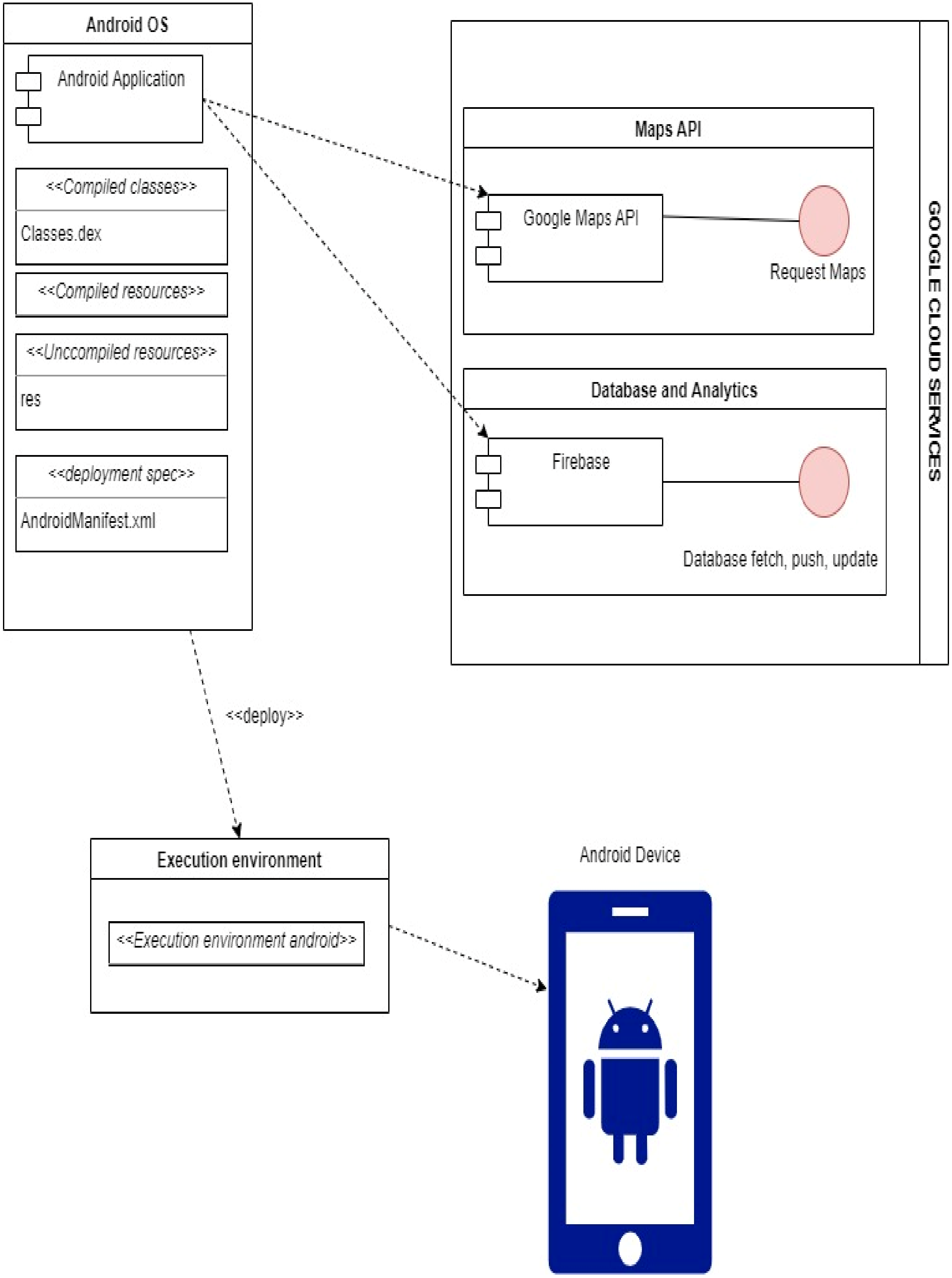
9.19 Activity Diagram – Place Order



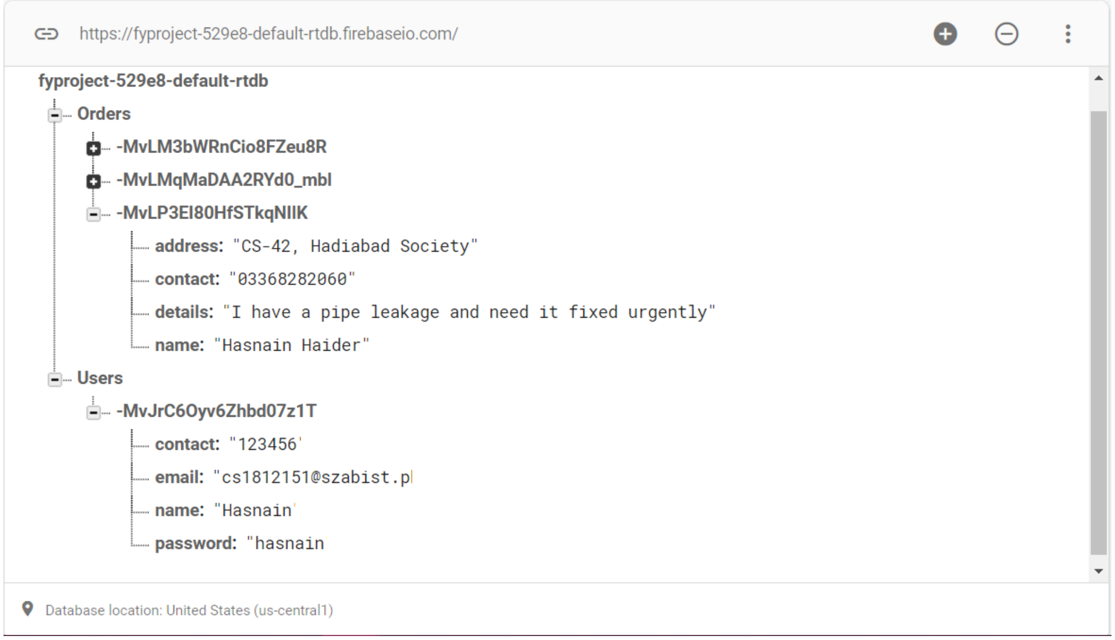
9.20 Activity Diagram – User type Seller (Orders)



9.21 Deployment Diagram



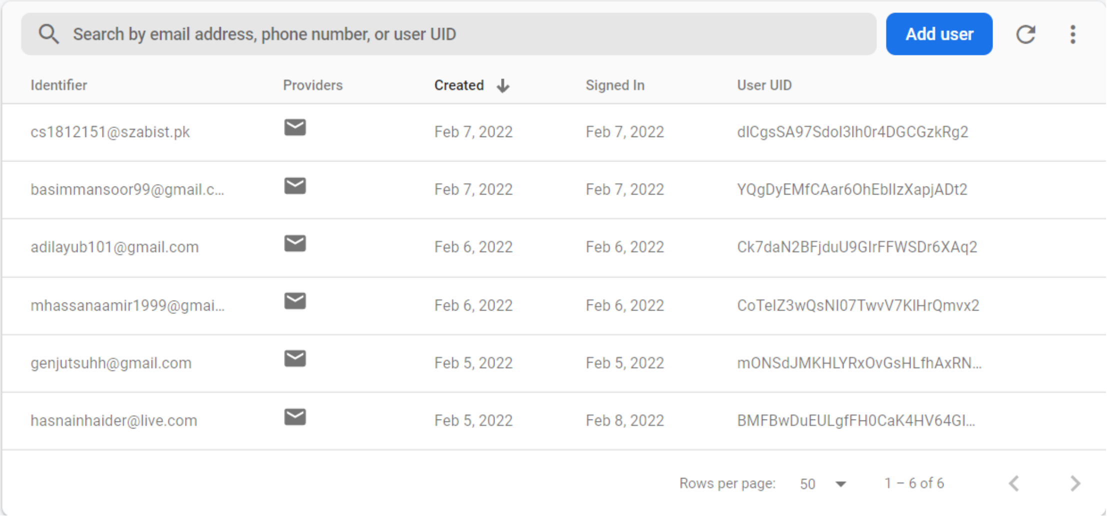
9.22 ERD (Firebase Connections)



https://fyproject-529e8-default-rtdb.firebaseio.com/

```
fyproject-529e8-default-rtdb
  Orders
    -MvLM3bWRnCio8FZeu8R
    -MvLMqMaDAA2RYd0_mbl
    -MvLP3EI80HfSTkqNIlK
      address: "CS-42, Hadiabad Society"
      contact: "03368282060"
      details: "I have a pipe leakage and need it fixed urgently"
      name: "Hasnain Haider"
  Users
    -MvJrC6Oyv6Zhbd07z1T
      contact: "123456'
      email: "cs1812151@szabist.pl
      name: "Hasnain'
      password: "hasnain
```

Database location: United States (us-central1)



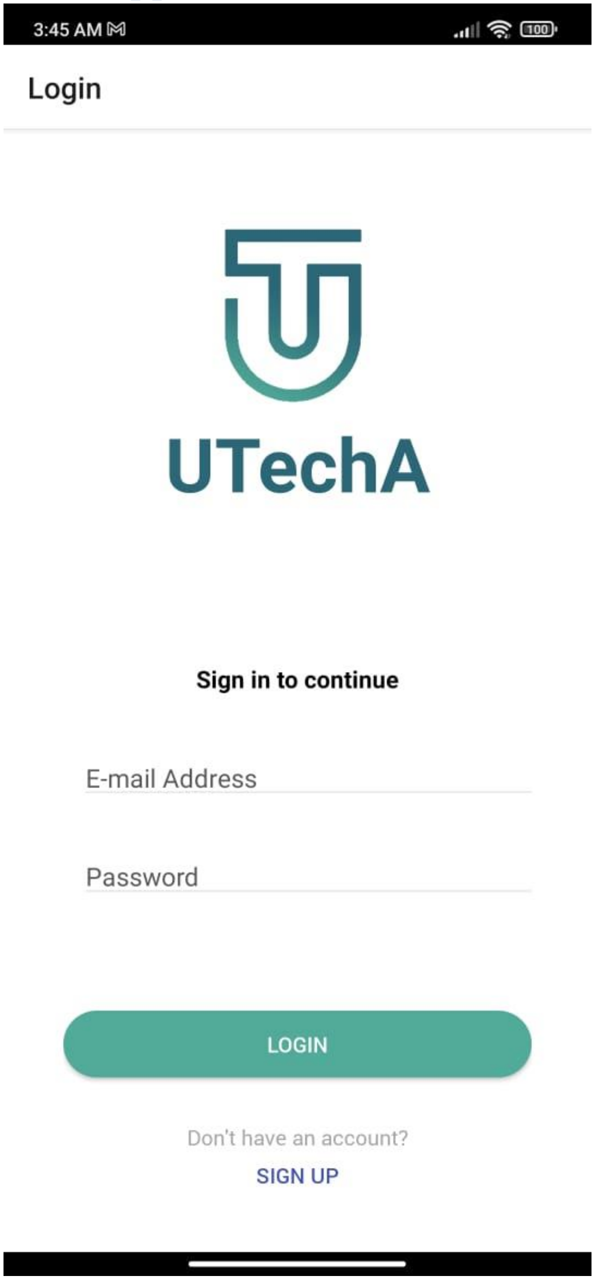
Search by email address, phone number, or user UID

Identifier	Providers	Created ↓	Signed In	User UID
cs1812151@szabist.pk	✉	Feb 7, 2022	Feb 7, 2022	dlCgsSA97Sdol3Ih0r4DGCGzkRg2
basimmansoor99@gmail.c...	✉	Feb 7, 2022	Feb 7, 2022	YQgDyEMfCAar6OhEbIlzXapjADt2
adilayub101@gmail.com	✉	Feb 6, 2022	Feb 6, 2022	Ck7daN2BFjduU9GIrFFWSDr6XAq2
mhassanaamir1999@gmai...	✉	Feb 6, 2022	Feb 6, 2022	CoTelZ3wQsNI07TwvV7KIHrQmvx2
genjutsuhh@gmail.com	✉	Feb 5, 2022	Feb 5, 2022	mONSdJMKHLYRxOvGsHLfhAxRN...
hasnainhaider@live.com	✉	Feb 5, 2022	Feb 8, 2022	BMFBwDuEULgfFH0CaK4HV64GI...

Rows per page: 50 1 – 6 of 6

USER MANUAL

1. Application Launch



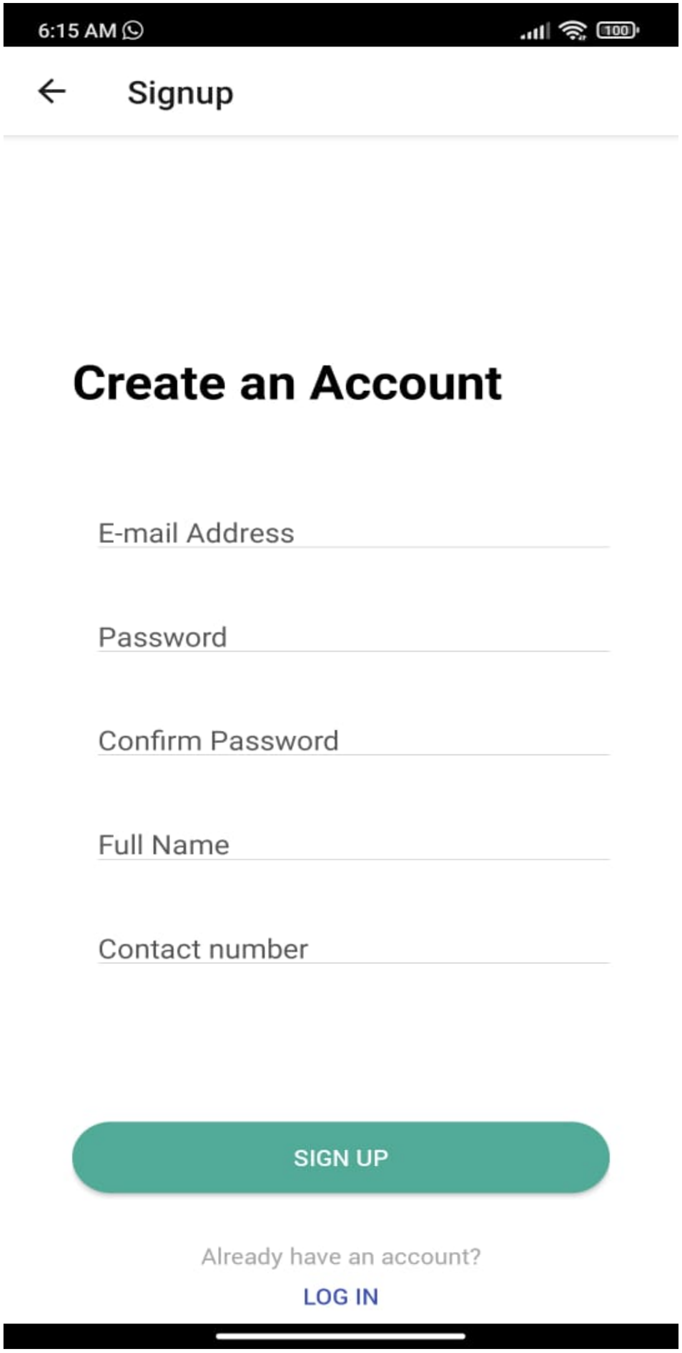
Sign in to continue

Don't have an account?

This is the screen that you are greeted with upon launching the application, from here you have two options:

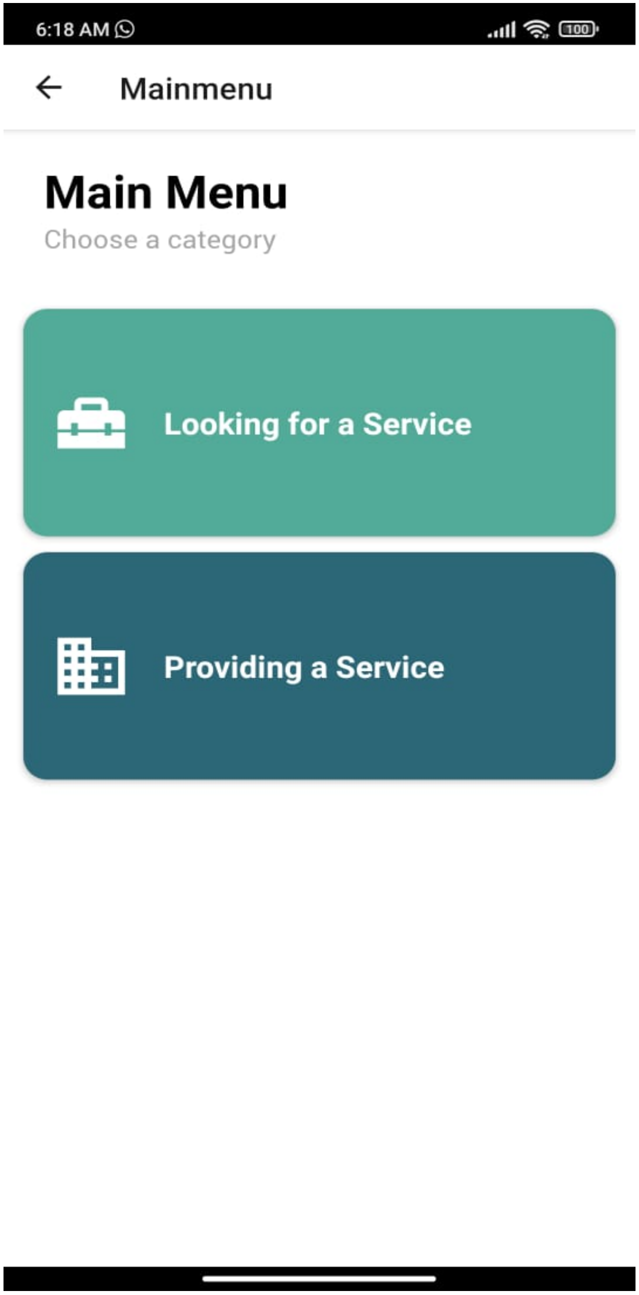
- Login using a registered Email Address
- Signup

2. Sign-up



If you do not have a registered account, after clicking on the signup button, you are redirected to this screen where you can register with your email address, Name, Contact Number, and Password. Once filled, you press the sign-up button and the application will notify you if you have been successfully registered. You can then press the back option on your phone or press log in at the bottom of the screen and login using your Registered email address and password.

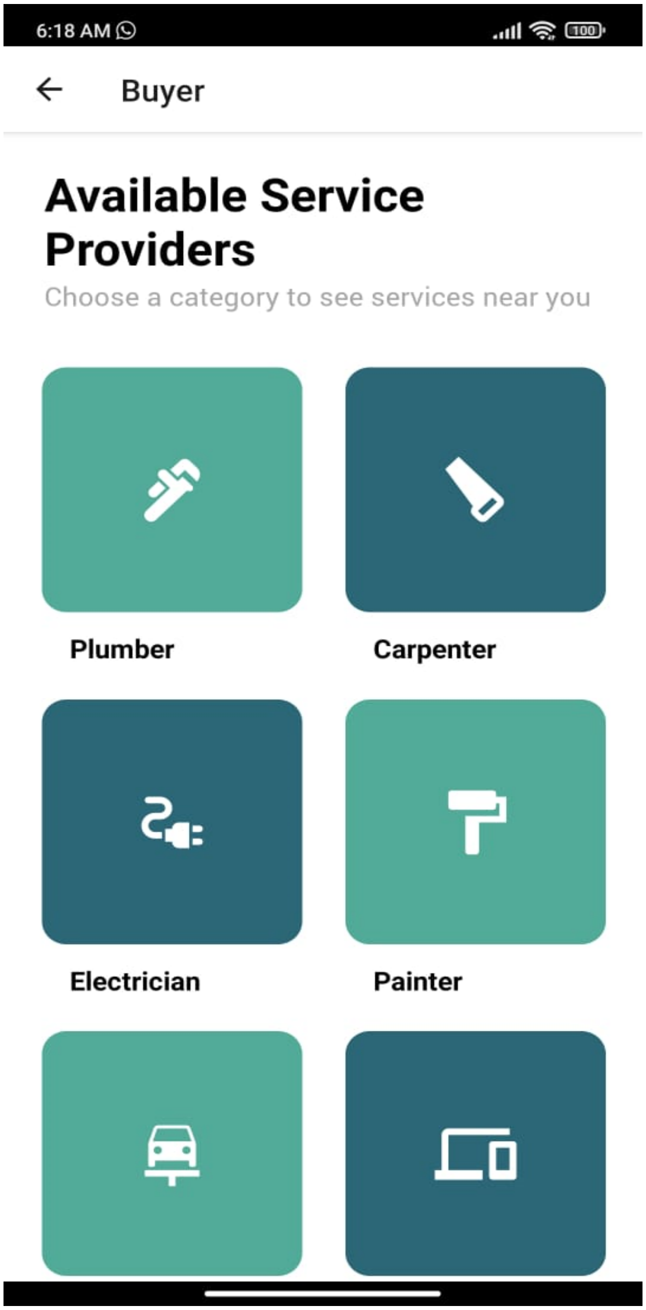
3. Customer Type Selection



Once you successfully login, you can select between two user types:

- Looking for a service: Customers who want technicians
- Providing a service: Technicians who want work
  Note: Providing a service option is only available to those registered users who are verified as sellers on the UtechA Admin.

4. Technician Type Selection



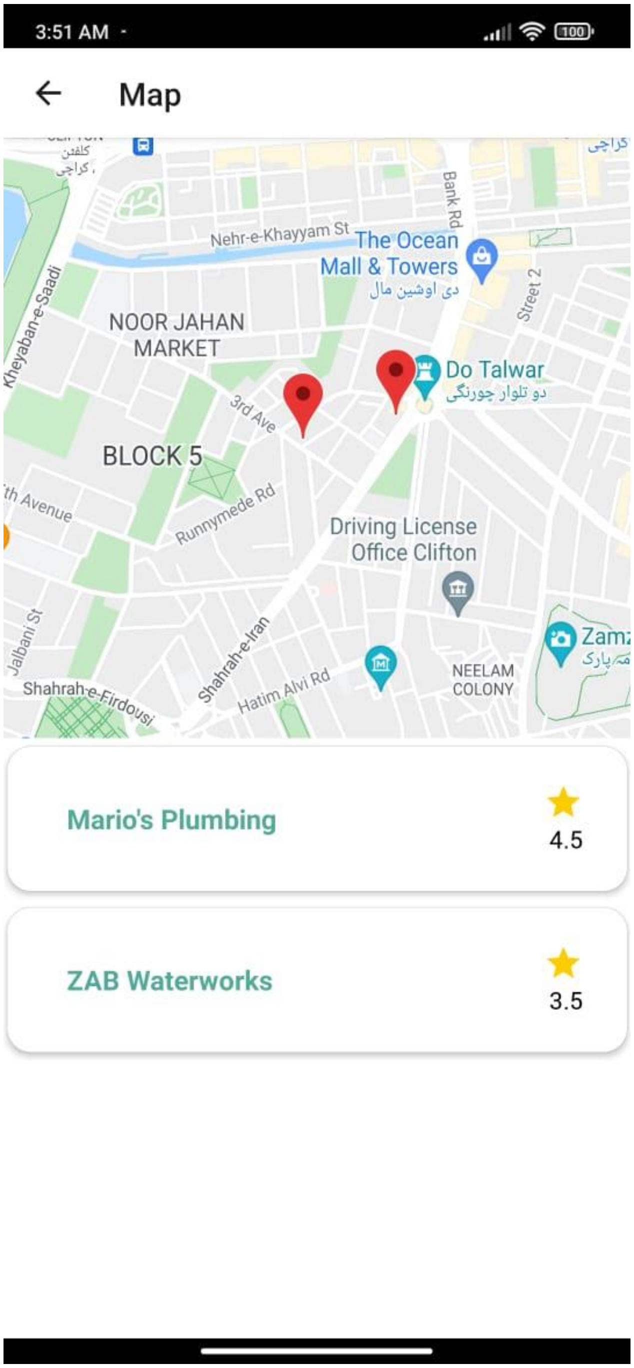
Buyer

Available Service Providers

Choose a category to see services near you

Once Looking for a service is selected, you are greeted with this screen which shows all available technician types according to services. Select the service you wish to avail by clicking on it.

5. Technician Selection

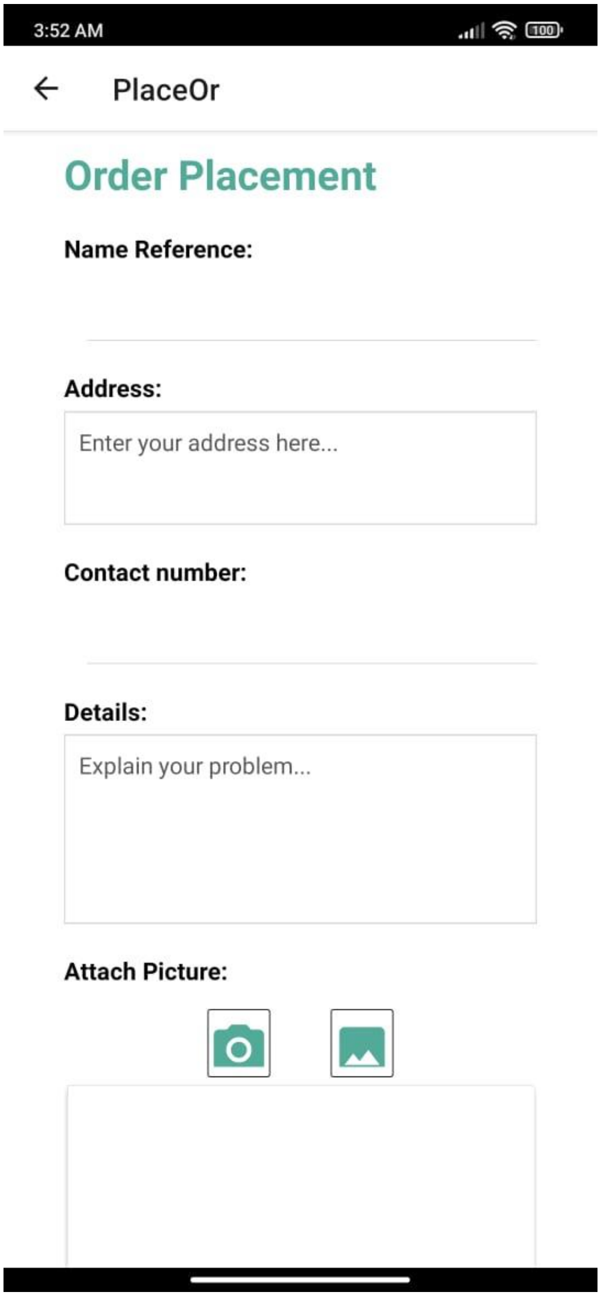


Mario's Plumbing ★ 4.5

ZAB Waterworks ★ 3.5

Once you have selected technician type, For example, plumber. You move onto this screen which shows all technicians of that service type in your area along with rating, you can click on the markers to show more details and even click on the directions button on the map which will show you directions to the technician by redirecting you to google maps.

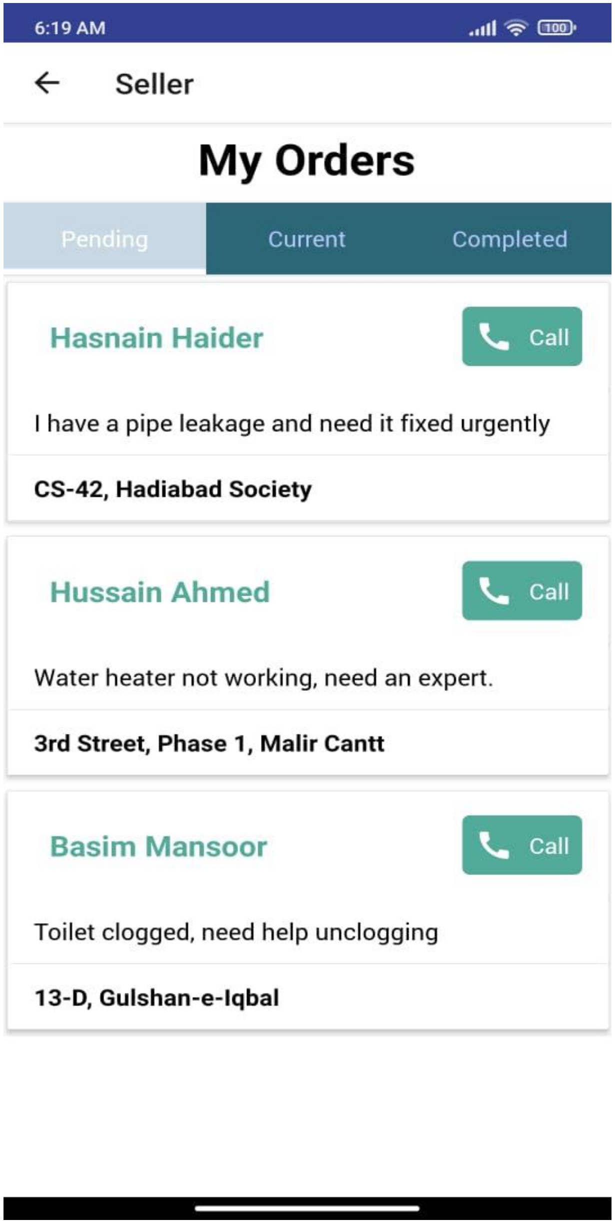
6. Order Placement



Once you select the desired technician, you place an order by filling out the details as explained in the UI and attach a picture either through your camera by using the button on the left, or your gallery by using the button on the right.

After pressing submit, this order will be sent to the technician and will be moved to your pending orders tab.

7. Pending Orders Tab (Providing a Service)



My Orders

Hasnain Haider ☎ Call

I have a pipe leakage and need it fixed urgently

CS-42, Hadiabad Society

Hussain Ahmed ☎ Call

Water heater not working, need an expert.

3rd Street, Phase 1, Malir Cantt

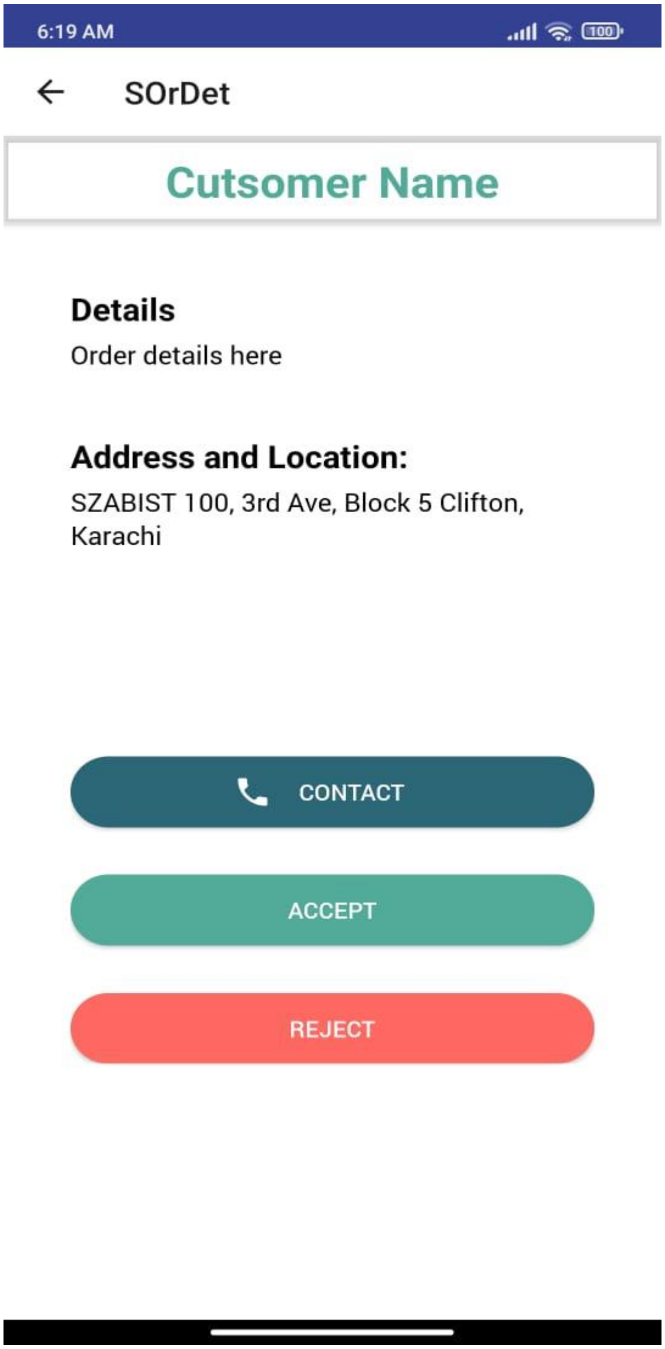
Basim Mansoor ☎ Call

Toilet clogged, need help unclogging

13-D, Gulshan-e-Iqbal

When you select providing a service, you are greeted with all pending orders that have been placed with you, you can see surface level information with these and you can select any order for more details. You can also directly call the customer from this screen to get a better understanding of the situation

8. Order Details



Cutsomer Name

Details

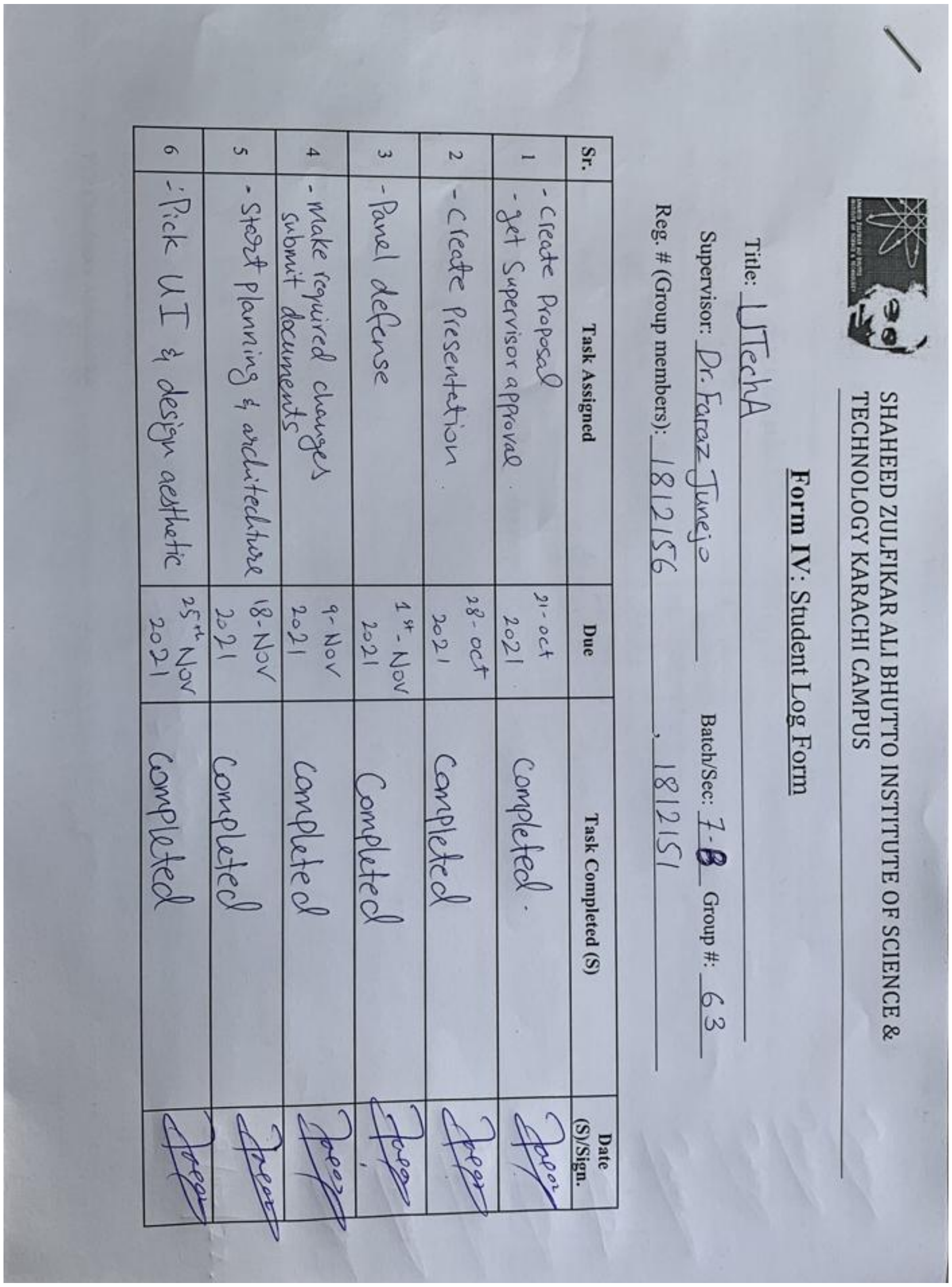
Order details here

Address and Location:

SZABIST 100, 3rd Ave, Block 5 Clifton,
Karachi

Once you select an order, you are shown order details and can either contact the customer, accept the order which will move the orders to your current orders tab or you can reject the order which will delete the order from your account.

10. Meeting Log Sheet



SHAHEED ZULFIKAR ALI BHUTTO INSTITUTE OF SCIENCE & TECHNOLOGY KARACHI CAMPUS

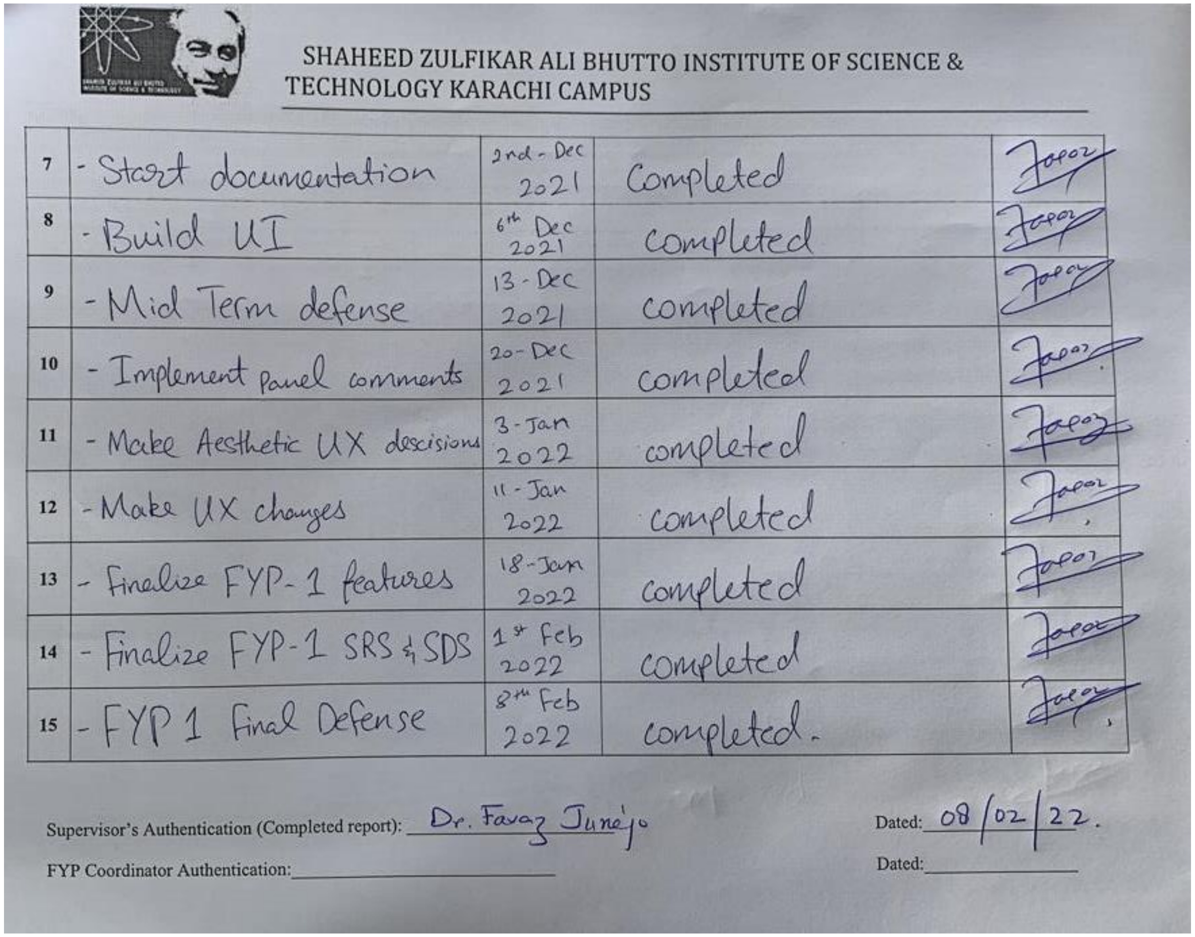
Form IV: Student Log Form

Title: UTechA

Supervisor: Dr. Faraz Junejo Batch/Sec: 7-B Group #: 63

Reg. # (Group members): 1812156 , 1812151

Sr.	Task Assigned	Due	Task Completed (S)	Date (S)/Sign.
1	- Create Proposal - get Supervisor approval	21-oct 2021	Completed.	
2	- Create Presentation	28-oct 2021	Completed	
3	- Panel defense	1st - Nov 2021	Completed	
4	- Make required changes submit documents	9-Nov 2021	Completed	
5	- Start planning & architecture	18-Nov 2021	Completed	
6	- Pick UI & design aesthetic	25th Nov 2021	Completed	

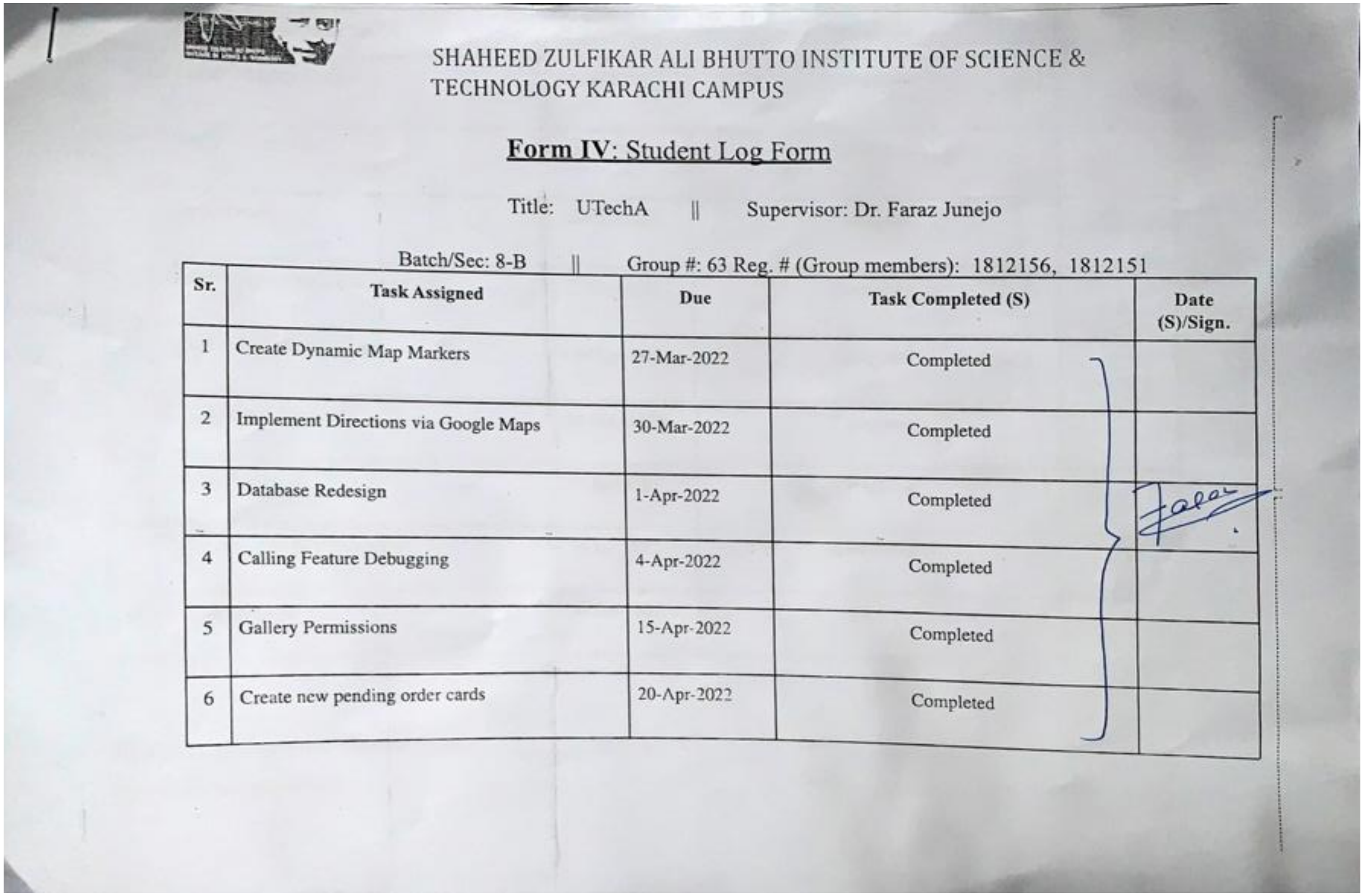


SHAHEED ZULFIKAR ALI BHUTTO INSTITUTE OF SCIENCE & TECHNOLOGY KARACHI CAMPUS

7	- Start documentation	2nd - Dec 2021	Completed	
8	- Build UI	6th Dec 2021	completed	
9	- Mid Term defense	13 - Dec 2021	completed	
10	- Implement panel comments	20 - Dec 2021	completed	
11	- Make Aesthetic UX descisions	3 - Jan 2022	completed	
12	- Make UX changes	11 - Jan 2022	completed	
13	- Finalize FYP-1 features	18 - Jan 2022	completed	
14	- Finalize FYP-1 SRS & SDS	1st Feb 2022	completed	
15	- FYP 1 Final Defense	8th Feb 2022	completed.	

Supervisor's Authentication (Completed report): Dr. Faraz Junejo Dated: 08/02/22.

FYP Coordinator Authentication:__________ Dated:__________



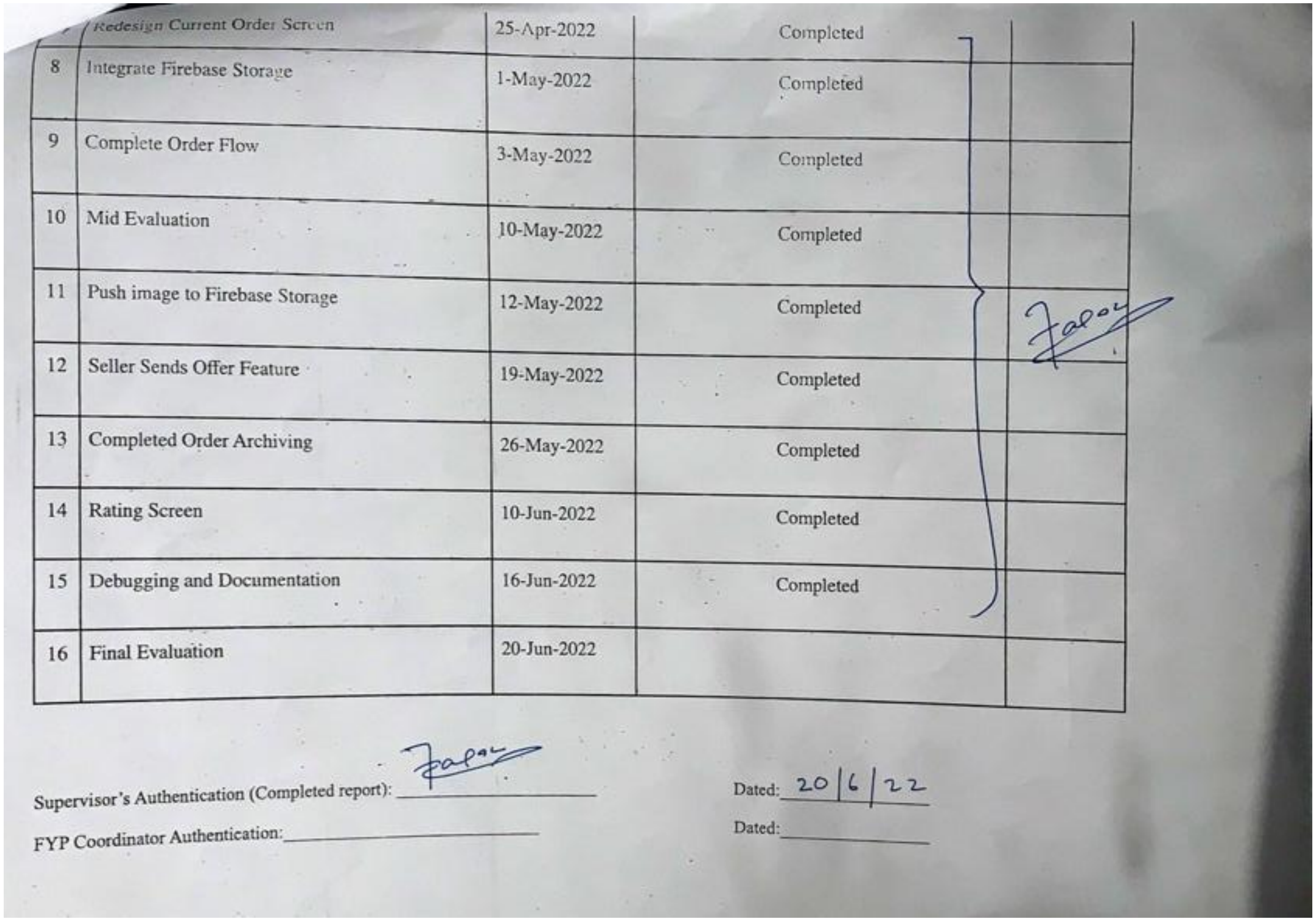
SHAHEED ZULFIKAR ALI BHUTTO INSTITUTE OF SCIENCE &
TECHNOLOGY KARACHI CAMPUS

Form IV: Student Log Form

Title: UTechA || Supervisor: Dr. Faraz Junejo

Batch/Sec: 8-B || Group #: 63 Reg. # (Group members): 1812156, 1812151

Sr.	Task Assigned	Due	Task Completed (S)	Date (S)/Sign.
1	Create Dynamic Map Markers	27-Mar-2022	Completed	
2	Implement Directions via Google Maps	30-Mar-2022	Completed	
3	Database Redesign	1-Apr-2022	Completed	
4	Calling Feature Debugging	4-Apr-2022	Completed	
5	Gallery Permissions	15-Apr-2022	Completed	
6	Create new pending order cards	20-Apr-2022	Completed	

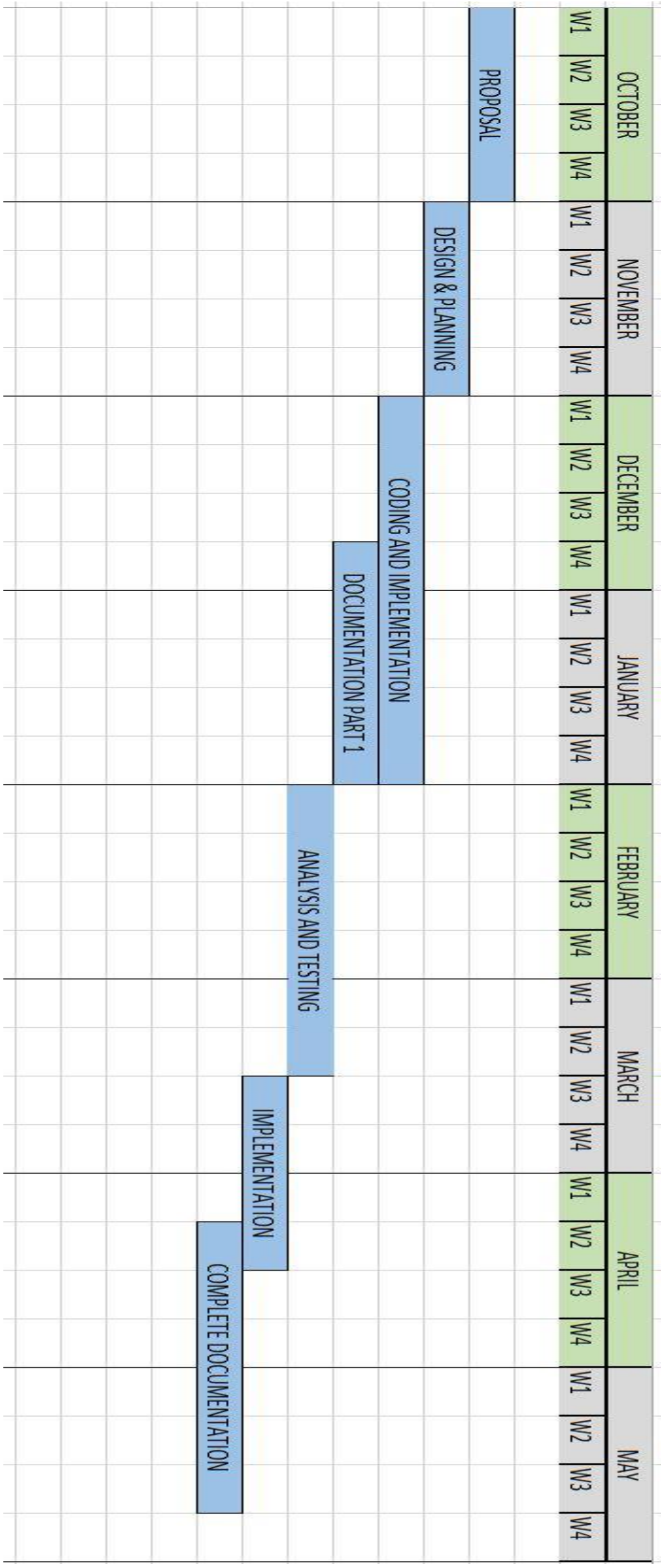


	Redesign Current Order Screen	25-Apr-2022	Completed	
8	Integrate Firebase Storage	1-May-2022	Completed	
9	Complete Order Flow	3-May-2022	Completed	
10	Mid Evaluation	10-May-2022	Completed	
11	Push image to Firebase Storage	12-May-2022	Completed	
12	Seller Sends Offer Feature	19-May-2022	Completed	
13	Completed Order Archiving	26-May-2022	Completed	
14	Rating Screen	10-Jun-2022	Completed	
15	Debugging and Documentation	16-Jun-2022	Completed	
16	Final Evaluation	20-Jun-2022		

Supervisor's Authentication (Completed report): ____________ Dated: 20/6/22

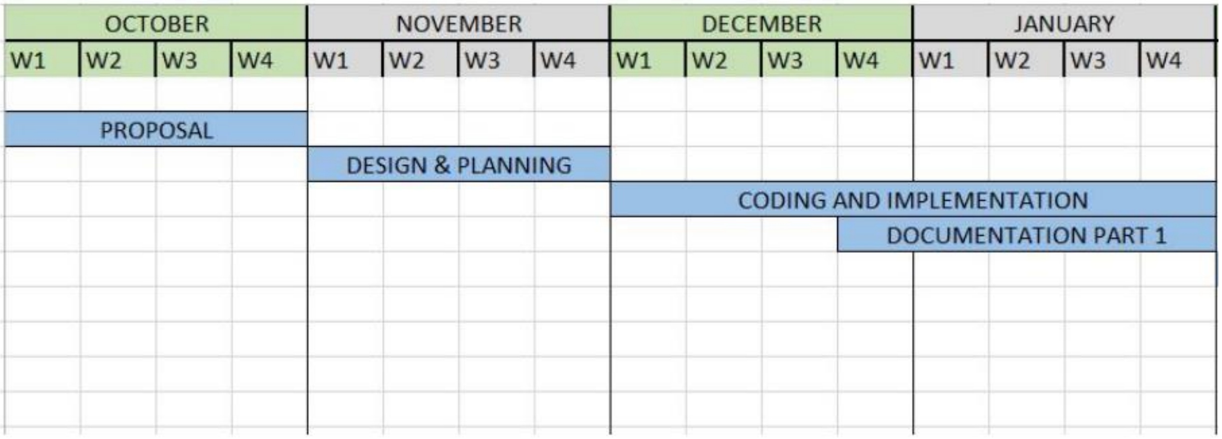
FYP Coordinator Authentication: ____________ Dated: ____________

11. GANTT Chart

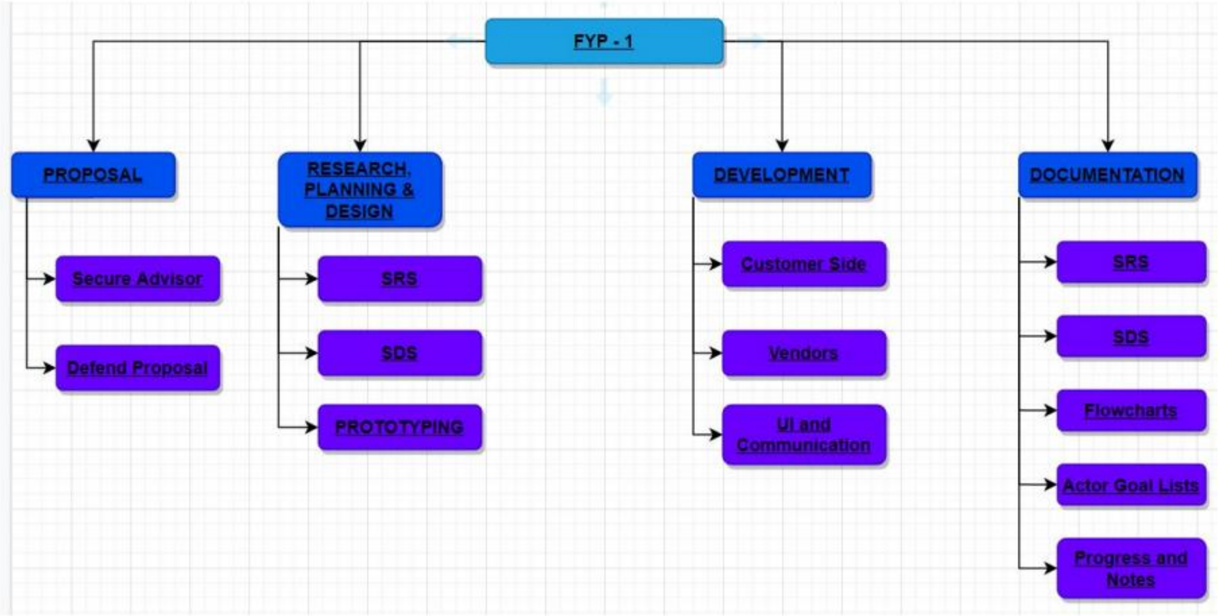


12. Iteration Plan

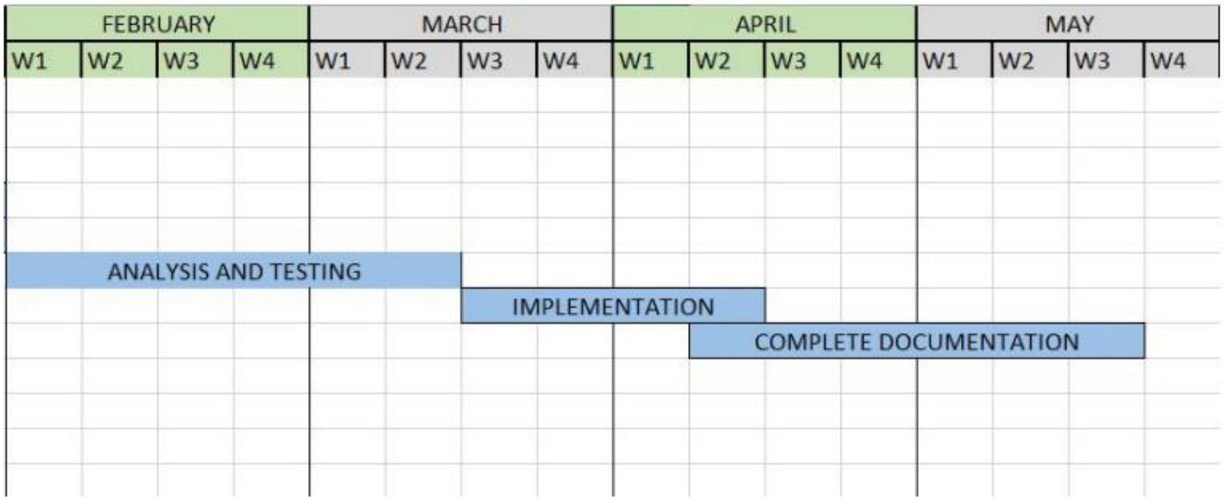
FYP-I:



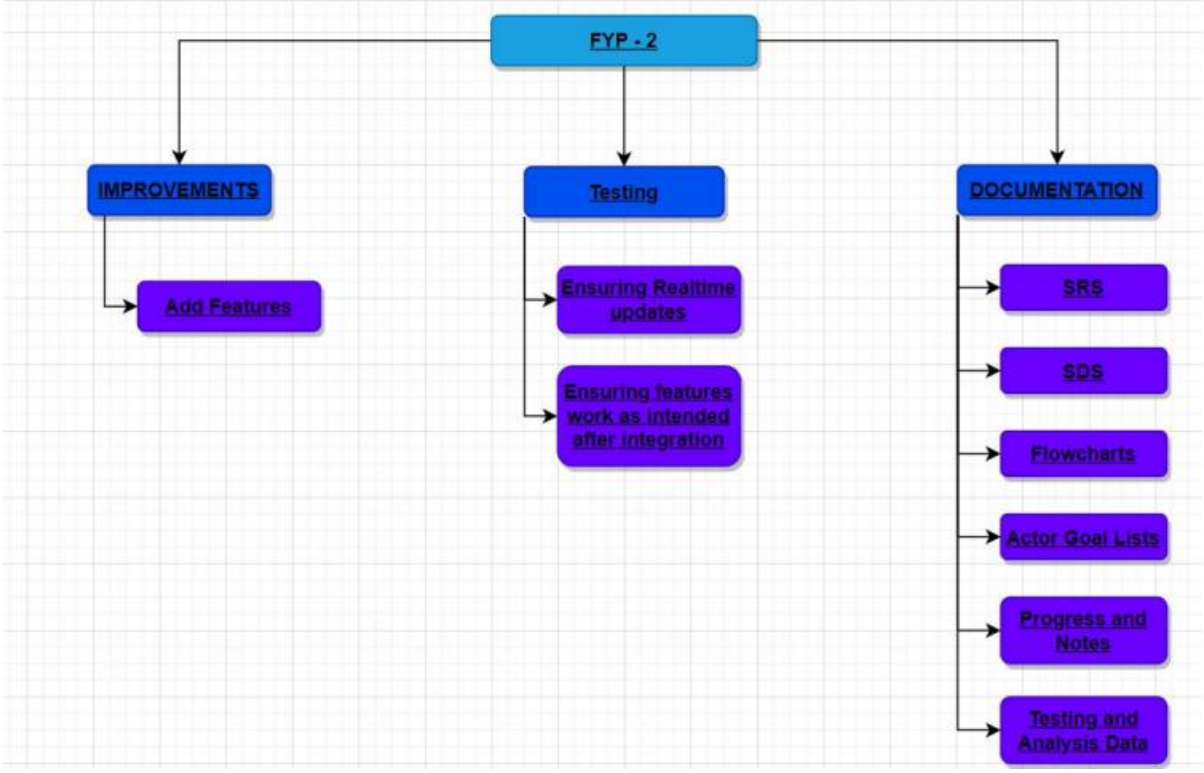
OCTOBER				NOVEMBER				DECEMBER				JANUARY			
W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
PROPOSAL															
				DESIGN & PLANNING											
								CODING AND IMPLEMENTATION							
											DOCUMENTATION PART 1				



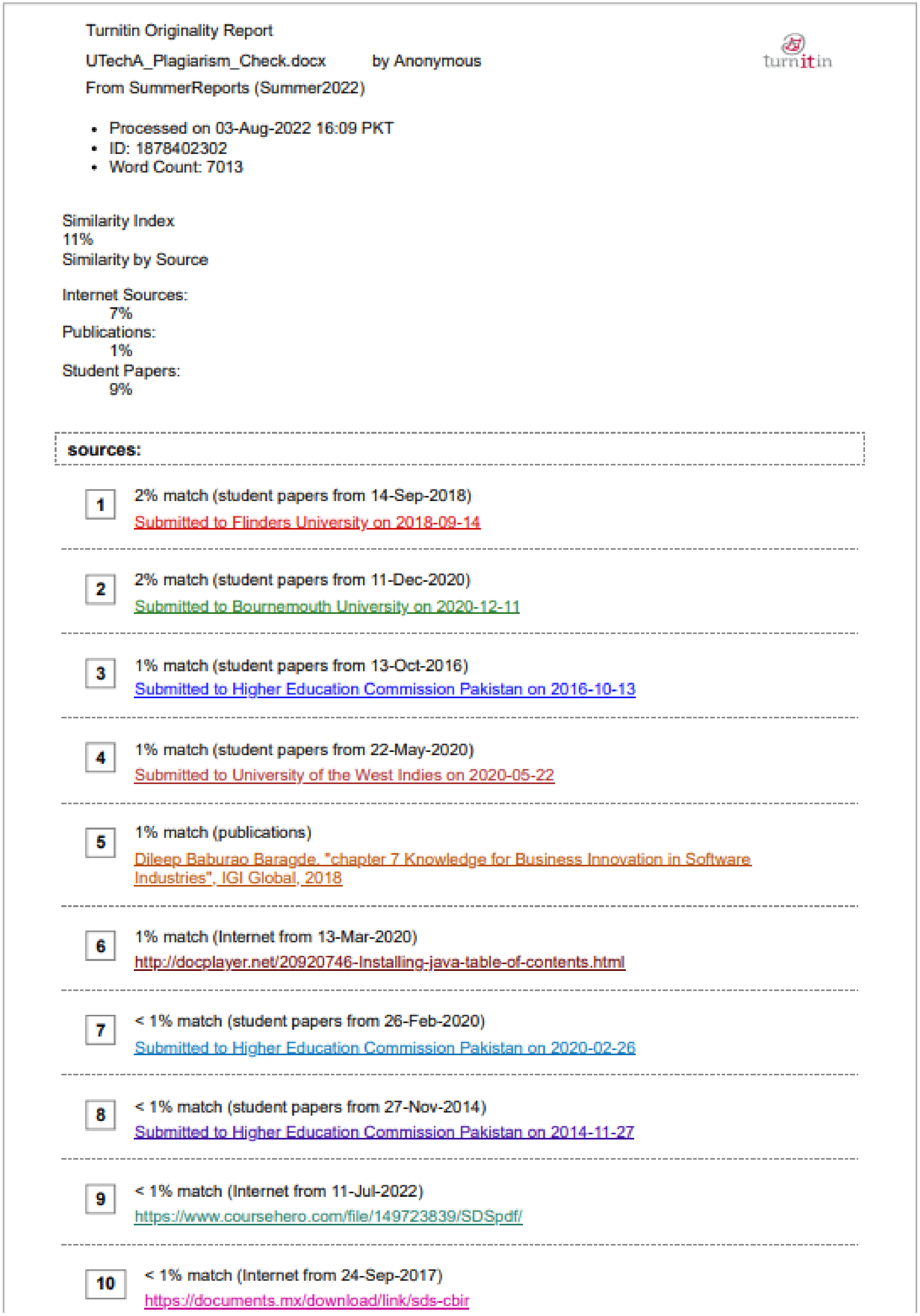
FYP-II:



FEBRUARY				MARCH				APRIL				MAY			
W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
ANALYSIS AND TESTING	▨	▨	▨	▨	▨										
						IMPLEMENTATION	▨	▨	▨						
									COMPLETE DOCUMENTATION	▨	▨	▨	▨	▨	


FYP - 2
IMPROVEMENTS
Testing
DOCUMENTATION
Add Features
Ensuring Realtime updates
Ensuring features work as intended after integration
SRS
SDS
Flowcharts
Actor Goal Lists
Progress and Notes
Testing and Analysis Data

13. Plagiarism Report



Turnitin Originality Report

UTechA_Plagiarism_Check.docx by Anonymous

From SummerReports (Summer2022)

- Processed on 03-Aug-2022 16:09 PKT
- ID: 1878402302
- Word Count: 7013

Similarity Index
11%
Similarity by Source

Internet Sources:
7%
Publications:
1%
Student Papers:
9%

sources:

1 2% match (student papers from 14-Sep-2018)
Submitted to Flinders University on 2018-09-14

2 2% match (student papers from 11-Dec-2020)
Submitted to Bournemouth University on 2020-12-11

3 1% match (student papers from 13-Oct-2016)
Submitted to Higher Education Commission Pakistan on 2016-10-13

4 1% match (student papers from 22-May-2020)
Submitted to University of the West Indies on 2020-05-22

5 1% match (publications)
Dileep Baburao Baragde. "chapter 7 Knowledge for Business Innovation in Software Industries", IGI Global, 2018

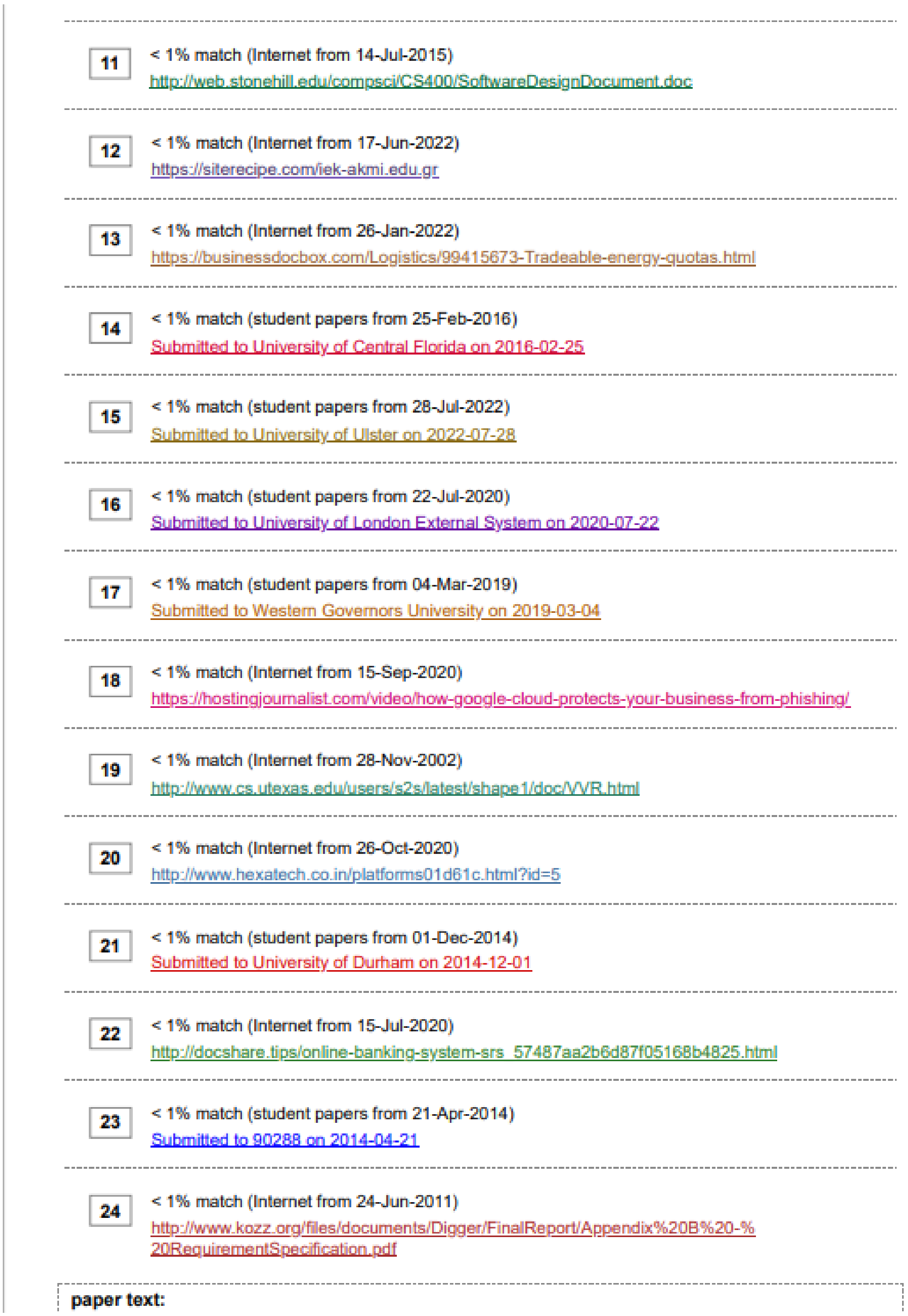
6 1% match (Internet from 13-Mar-2020)
http://docplayer.net/20920746-Installing-java-table-of-contents.html

7 < 1% match (student papers from 26-Feb-2020)
Submitted to Higher Education Commission Pakistan on 2020-02-26

8 < 1% match (student papers from 27-Nov-2014)
Submitted to Higher Education Commission Pakistan on 2014-11-27

9 < 1% match (Internet from 11-Jul-2022)
https://www.coursehero.com/file/149723839/SDSpdf/

10 < 1% match (Internet from 24-Sep-2017)
https://documents.mx/download/link/sds-cbir



11 < 1% match (Internet from 14-Jul-2015)
http://web.stonehill.edu/compsci/CS400/SoftwareDesignDocument.doc

12 < 1% match (Internet from 17-Jun-2022)
https://siterecipe.com/iek-akmi.edu.gr

13 < 1% match (Internet from 26-Jan-2022)
https://businessdocbox.com/Logistics/99415673-Tradeable-energy-quotas.html

14 < 1% match (student papers from 25-Feb-2016)
Submitted to University of Central Florida on 2016-02-25

15 < 1% match (student papers from 28-Jul-2022)
Submitted to University of Ulster on 2022-07-28

16 < 1% match (student papers from 22-Jul-2020)
Submitted to University of London External System on 2020-07-22

17 < 1% match (student papers from 04-Mar-2019)
Submitted to Western Governors University on 2019-03-04

18 < 1% match (Internet from 15-Sep-2020)
https://hostingjournalist.com/video/how-google-cloud-protects-your-business-from-phishing/

19 < 1% match (Internet from 28-Nov-2002)
http://www.cs.utexas.edu/users/s2s/latest/shape1/doc/VVR.html

20 < 1% match (Internet from 26-Oct-2020)
http://www.hexatech.co.in/platforms01d61c.html?id=5

21 < 1% match (student papers from 01-Dec-2014)
Submitted to University of Durham on 2014-12-01

22 < 1% match (Internet from 15-Jul-2020)
http://docshare.tips/online-banking-system-srs_57487aa2b6d87f05168b4825.html

23 < 1% match (student papers from 21-Apr-2014)
Submitted to 90288 on 2014-04-21

24 < 1% match (Internet from 24-Jun-2011)
http://www.kozz.org/files/documents/Digger/FinalReport/Appendix%20B%20-%20RequirementSpecification.pdf

paper text:

14. References

React Native:

https://reactnative.dev/

React Native Firebase:

https://rnfirebase.io/

Firebase:

https://firebase.google.com/docs

NativeBase:

https://docs-v2.nativebase.io/

React Native Vector Icons:

https://github.com/oblador/react-native-vector-icons

React Native Maps:

https://github.com/react-native-maps/react-native-maps

Google Cloud API:

https://cloud.google.com/apis/docs/overview

React Native Immediate Phone Call:

https://github.com/wumke/react-native-immediate-phone-call