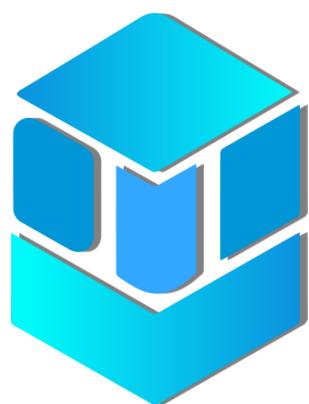


Mandob© | Business is never easier.



MANDOB
Business is never easier



Faculty of computers and Information Systems Department

Supervised by:
Prof. DR. Mona Nasr

Team Members



Hadeer Samir



Kareem Sameh



Omnia Ahmed



Khalil Emad



Mohraeel Salah



Abanoub Nagib

ACKNOWLEDGEMENT

First and foremost, we would express our deepest gratitude and appreciation to our advisor and great supervisor **Prof. Dr. Mona Nasr** for her support, outstanding guidance and encouragement throughout our graduation project. She is a great supporter for Mandob. Starting from Nile competition until today. Mandob really thank you for bringing it to light. Also we would like to thank **Eng. Amr Essam** for his outstanding guidance and path related instructions along the project phases.

We would also like to thank our families, especially our parents, for their encouragement, patience, and assistance over the years.

We are forever indebted to our parents, who have always kept us in their prayers and pushed us to Success.

Finally, for our faculty for providing the suitable environment that leaded us to represent the best image that computer science graduates of Helwan University are supposed to represent.

Page of Context

Chapter 1: Introduction	10
1.1 Overview	11
1.2 Objectives	11
1.3 Purpose	11
1.4 Scope	12
1.5 General Constraints	15
Chapter 2: Project “Planning and Analysis”	16
2.1 Project Planning	17
2.1.1 Feasibility Study	17
2.1.2 Estimated Cost	18
2.1.3 Gantt Chart	19
2.1.4 Business Model Canvas	21
2.1.5 Expertise Approval	22
2.1.6 Market study of similar apps	23
2.1.6.1 Analysis and Limitation of existing system	24
2.2 Analysis of the new system	29
2.2.1 User Requirements	29
2.2.2 System Requirements	30
2.2.3 Functional Requirements	30
2.2.4 Non Functional Requirements	42
2.5 Advantages of the new system	44
2.6 Risk and Risk Managements	45

Chapter 3: Software Design 49

3.1 Application Prototyping	50
3.2 ERD Diagram	53
3.3 Class Diagram	54
3.4 Use-case Diagram	55
3.5 Context Diagram	58
3.6 DFD Diagram	59
3.7 Sequence Diagram	63
3.8 Scenario	67

Chapter 4: Implementation 74

4.1 Users Models	75
4.2 Users Providers	76
4.3 SignUp	77
4.4 Login including (Facebook and Google)	78
4.5 Add service or Product	79
4.6 View Services and products details	81
4.7 Delete Function	82
4.8 Update Function	82
4.9 Get notifications	82

4.10 View services and products overview	83
4.11 Get latest added Products and services	83
4.12 View details of services and products	84
4.13 App Navigation	85
4.14 Get notifications about purchase	86
4.15 Navigate to cart and edit order	87
Chapter 5: Testing	88
5.1 Functional Testing	89
5.2 Non-Functional Testing	90
Chapter 6: Results and Discussion	108
6.1 Results	109
6.1.1 Expected Results	109
6.1.2 Actual Results	109
6.2 Discussion	110
Chapter 7: Conclusion	111
Chapter 8: Future Work	112

Abstract:

Main idea:-

In the last 6 years Egypt has witnessed a huge growth in business field. Due to the major renaissance that our president made available. so here in FC-AI Helwan. We came up with an idea of combining Ecommerce with offline commerce to ease the small to medium business and make it available for everyone. Mandob is that kind of technology that allow you to invest in internal business and create more flow in the Egyptian market. You can invest in places for selling and renting purposes. You can offer your services and products you can build a running, well planned business online.

Structure, Technologies and process.

Structure:

E-Commerce App with extendable categorical business field.

The application Cover the most important business placeholders that are very important for any business investment. Those are:

- Place to start your business : Where the placeholders here are brokers and places for sell/rent owners.
- Finishing works at your business place : Where the placeholders here are finishing work companies and individuals that offers instant finishing services that support business opening.

- Hardware to ready up your business : Where the placeholders here are Hardware suppliers companies and individuals that offers tools and equipments that are essentially needed for pre-product importing like lighting system installing, shelf, Place decorations...etc.
- Products suppliers : Where in Mandob we allow wholesalers and importers to offer their products in wholesale . or per piece so you can choose what is suitable for your business and order it within a click

So Here You go your Store is full and you can start your business now with no worries about anything. If u need installations or further products its always available you can re-order it in no time.

Technologies

Flutter Framework , Firebase real-time database and Firestore.

We used flutter technology due to its outstanding reputation in offering multi-platform applications so we can allow every user to use Mandob. You can use your android, i-phone or browser to start and manage your business. It is never easier than this.

And we also used Firebase/Firestore to save, retrieve and manipulate data so it's well delivered and processed throughout the application life cycle.

Firestore has offered a well-planned and structured algorithms. So we are willing a good use of it to deliver the touchable easiness for end-users.

Process:

(“The process has two main branches 1-Service Provider 2-Business Owner”)

Service provider process:

Register, Login, assign role and upload products and services

- First: you should register by entering your info (name, email ,address ,... and more authentication data).
- Choosing your role (Broker, Finishing work , hardware supplier , product supplier , working hand)
- Then you can login to the application by E-mail &password
- You can update all your services and products to be viewed by users
- Your home page contains all your uploaded products and services and you can edit, delete or see their details.
- You get notifications once anyone have ordered any of it.
- You can contact your customer once they place an order to complete their delivery

Chapter 1 : Introduction

In this chapter we're going to discuss and go deeper in the overview of the project and know more about its scope and limitations, and explain some terminologies we will find throughout the document.

1.1 Overview

Mandob is cross-platform applications that recruit a technology that allows users to start and manage their business in an easy way. Combining e-commerce with real life commerce. Where the two main users branches (service or product provider) and (new or existing business owners) can offer their services and products or buy services and products respectively.

Mandob has two main kind of users :

- Service and product provider
- Business owner

1.2 Objectives

- The app allows users to upload their services and products with all the essential and required details
- Customer navigate the application and search for the required product and service they need.
- The customer complete his purchase through verity of products and services
- The application connects the seller and buyer until the purchase process is completed

1.3 Purpose

The main purpose of Mandob is to ease the small to medium business construction and make it available for everyone. Which leads to an increase in

commerce field and create new investment opportunities in the Egyptian market.

1.4 Scope

The working scope that was required to finish the project was well designed and premade to achieve the highest quality aspect that were :

1- Planning :

- Studying the environment and collecting data about the idea so we came up with what the business need.
- Estimating application production cost
- Constructing Gantt's chart for the project to estimate requirements.
- Constructing Business model Canvas.
- Getting approval from experts in e-commerce field.
- Market study of similar applications.
- Need of new System study.
- Determining Functional and non-functional requirements.
- Risk and Risk Management.

2- Designing :

Determining the diagrams to be carried in the project that are :

- Application prototyping

- ERD Diagram
- Class Diagram
- Use-case Diagram
- Context Diagram
- DFD Diagram
- Sequence Diagram
- Scenario Diagram

3- Coding

Where the main functionalities to be coded in our project was estimated and pre-defined so clearly also the future development process is well estimated

The main requirements for project were :

--First category code requirements –

- Users Model
- Users Providers
- Signup
- Facebook and Google Signup
- Login
- Add Service or product
- View services and products details
- Update Service and product
- Delete Service or product
- Get notifications

-- Second category code requirements –

- View services and products overview.
- Get latest added services and products.
- Search for the required category
- View details of all listed services and products
- Buy service or product
- Get notification about purchase
- Navigate to cart and edit order

--Future plan for further development –

- Mandob statistics
- Cashier link
- Mandob expert consultant

4- Testing

-- Functional testing –

- Unit testing
- Regression testing
- Integration testing

--Non-functional testing –

- Performance testing
- Stress testing
- Security testing

5- Documentation

The documentation should mainly include these main chapters:-

- Introduction: includes an overview of the project and limitations
- Project Planning: includes the tools and technologies as well as tasks and timeline plan
- Project Requirements: includes the functional and non-functional requirements
- Project Design: includes the diagrams
- Project Implementation: includes user application and administrator system
- Project Testing: includes testing types listed previously

6- Results

- Expected application results
- Actual results
- Final App Discussion

1.5 General Constraints

- The application should be compatible with old and new mobile phones
- The application should work properly if the internet connection is poor
- The application should be easy to use and a predefined tutorial is premade
- Management of big data and future categories to add is essential.

Chapter 2 :

Project planning and analysis

In this chapter we're going to discuss and go deeper in the overview of the project and know more about its scope and limitations, and explain some terminologies we will find throughout the document.

2.1 Project Planning

2.1.1 Feasibility Study

A Feasibility study is used to determine if a business or a specific project is achievable, so for determining the achievability of our project we'll go deeper in the following points:

1. Market Analysis

- The market is in need for an e-commerce technology that makes business more easier for business owners
- Mandob is going on well with the technical market. Technical advisors took a great tribute upon it
- Mandob is so competitive as it is used by business owners and can buy services and products with the whole sale price
- The application is so extendable so it doesn't count on geo-location

2. Organizational Analysis

- Number of employees in the early Mandob :

4 Developers and 1 Tester, The revenue main stream will start by share throughout the development team and later will be set to monthly salary for the extended team.

The coverage we need to generate and support our sales: We upgrade the application by the developers. We make the application better from the rate and feedback from customers when they do the transactions.

App will get revenue in late deployment from sponsor ads . so if a user want to increase his sales opportunities he could sponsor his product so it's seen in the front page more frequent.

3. Operational Analysis

in our project we provide some rules and services to the developers that were :

- Good development environment
- Online and offline meeting to coop with the deadlines and task deliveries
- Good Team Leader support by team investing.

Mandob app delivery was to be held online, through Facebook marketing, Instagram Marketing and Google Ads.

Social media marketing is based on creating attractive visual illustrations and motion graphics video to start app marketing.

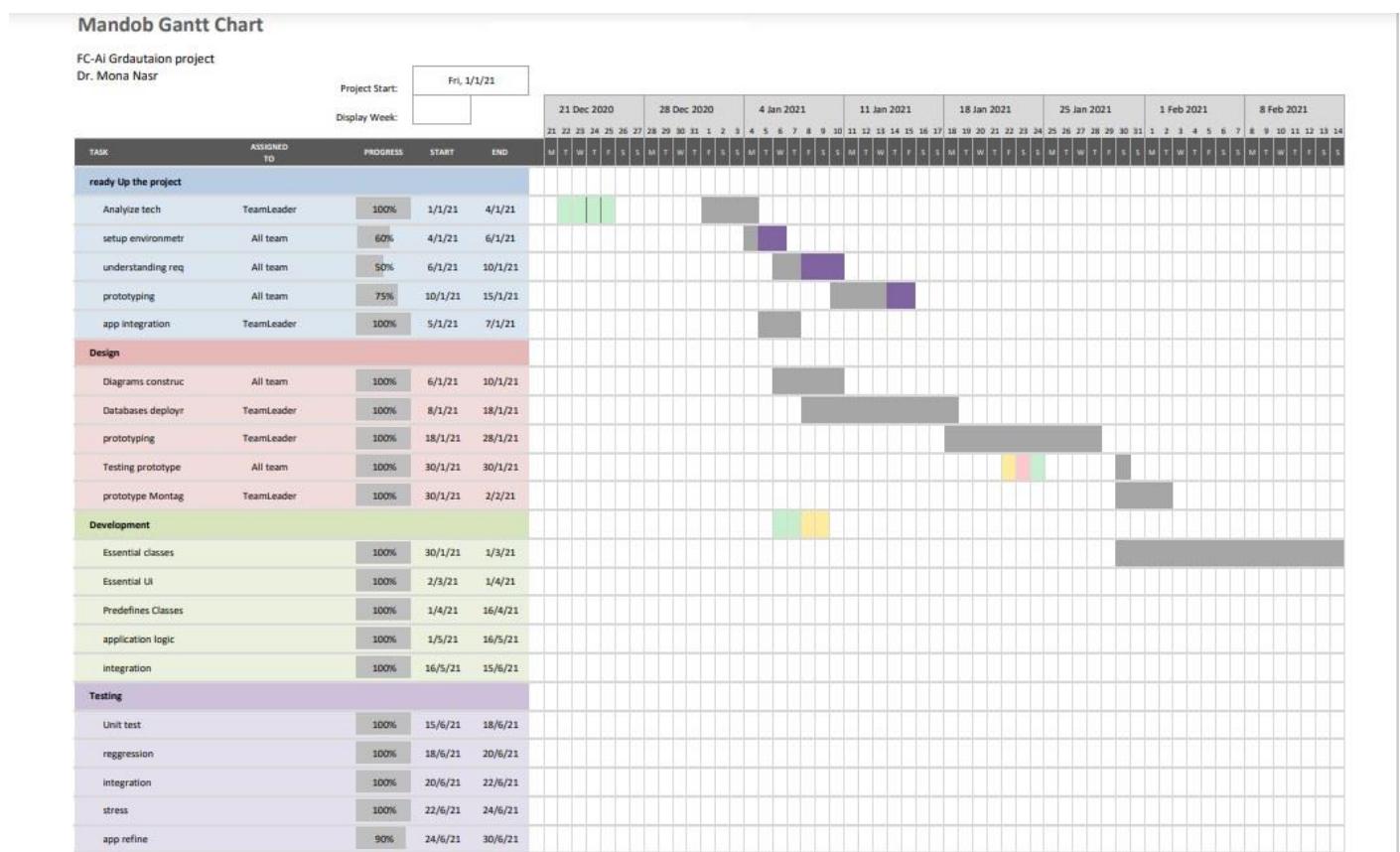
So for marketing and customer delivery we have used the following products with the following costs that will appended in estimated app cost

2.1.2 Estimated Cost

A cost estimate is approximation of the cost of a program, project or operation the cost estimate is the product of the cost estimating process and our estimated cost for this project comes as following :

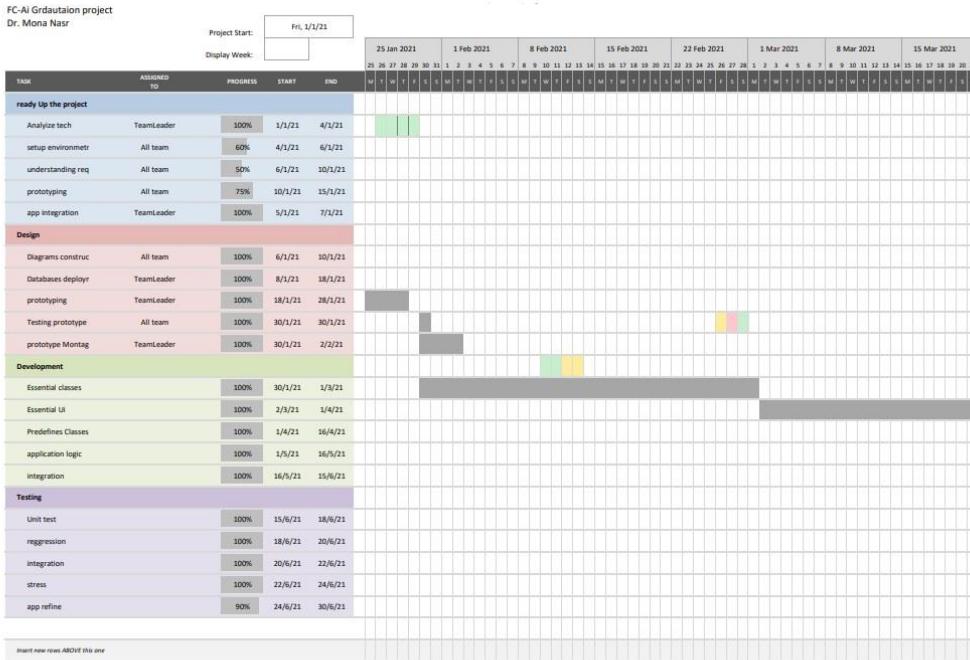
- Designing Social media banners => Cost 200 Per Design
- Design Motion Graphics Video => Cost 750 per Video
- Social Media account manager => Cost 2500 per month
- Offline meeting => Cost 1000 per month
- Salaries paid to new employees according to the extendable plan => 5K to 8K

2.1.3 Gantt's Chart



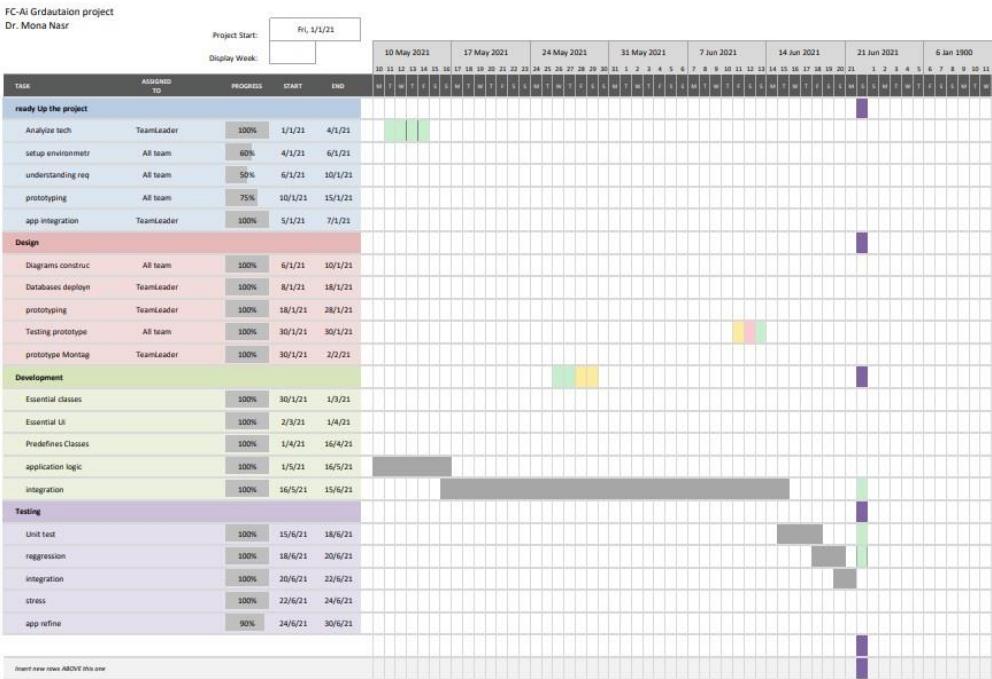
Mandob Gantt Chart

FC-AI Graduation project
Dr. Mona Nasr



Mandob Gantt Chart

FC-AI Graduation project
Dr. Mona Nasr

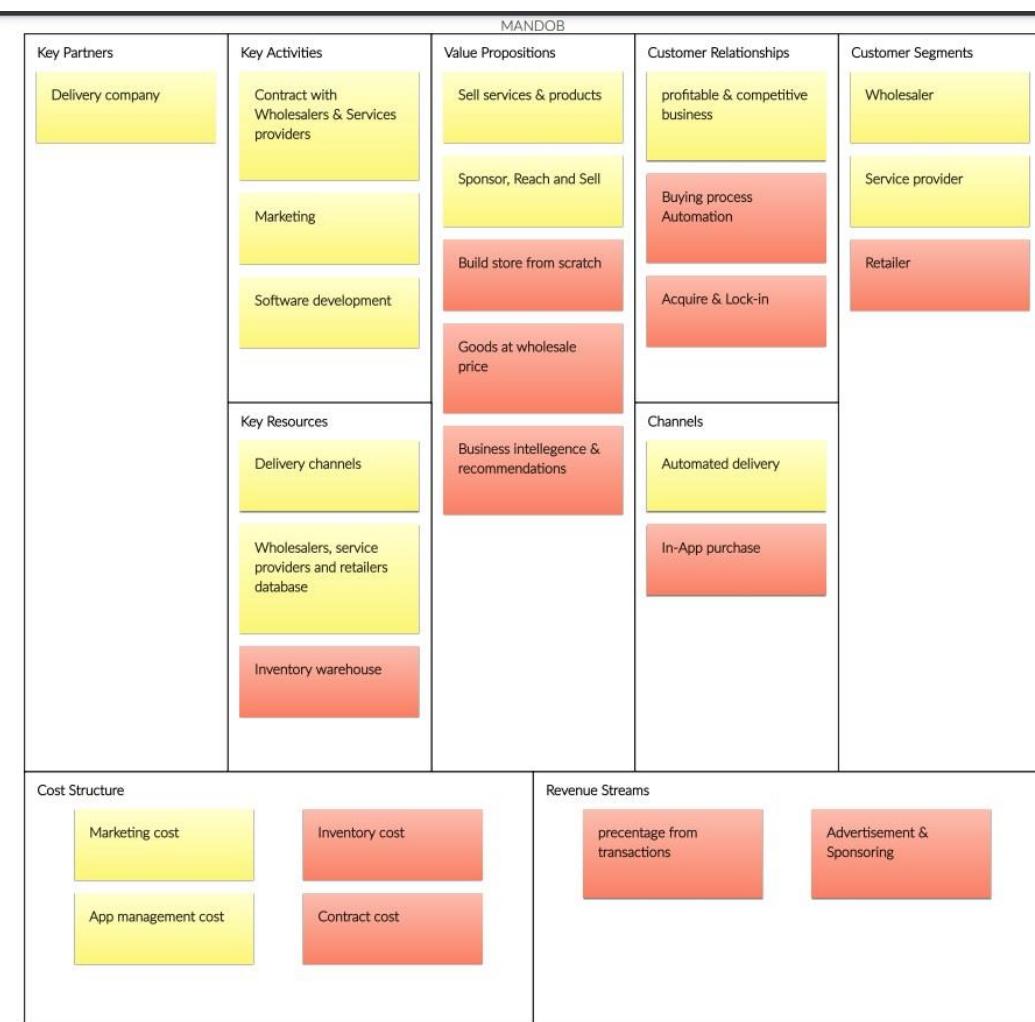


2.1.4 Business Model Canvas

The Business Model Canvas is a strategic management template used for developing new business models and documenting existing ones. It offers a visual chart with elements describing a firm's or product's value proposition, infrastructure, customers, and finances, assisting businesses to align their activities by illustrating potential trade-offs.

So Mandob Have a business model canvas has a firmly approved BMC approved by Prof. Dr. Mona Nasr and Nile University judges that follows :

Mandob BMC V1.0



Mandob BMC V2.0

Key Partners	Key Activities	Value Proposition	Customer Relationships	Customer Segments
Leading customer from each category	Contract with Wholesalers & Services providers	Sell easier and faster	profitable & competitive business	Wholesalers
Delivery company	Lock-in with retailers	Sponsor, Reach and Sell	Buying process Automation	Brokers / estate owners
Investors	Expand to new cities	Build store from scratch	Acquire & Lock-in	Finishing workers / offices
	Add new features	discounts on Goods at wholesale price	Ratings & feedback system	Hardware suppliers / offices
	Develop new BI algorithms & data analysis	Reduce buying cost & risk	Self-service	Retailers
	Key Resources	Upfront pricing	Channels	
	Technology team	Business intelligence & recommendations	Mobile app	
	Customers database	business monitoring	Website	
	BI and business analysis algorithms	safe packing & delivery	Social media	
	analytics expertise		Online & offline advertisements	
	Inventory warehouse			
	warehouse team			
Cost Structure				
Marketing	App management	Warehouse	Revenue Streams	
Contract & legal activities	warehouse team salaries	analytics & business expertise	Advertisement & Sponsoring	percentage from each transaction
				Business monitoring services subscription

2.1.5 Expertise Approval

One of the most important takeoffs in any application deployment in e-commerce field is the expertise consultancy before even starting the planning phase. So the app was set off by consulting the head of E-commerce Prof.Dr. Mona Nasr and Eng. Amr Essam the E-Commerce expert and was approved by them so it's' fair enough to make it come to the light.

2.1.6 Market study of similar apps

App Logo	App Name	Wholesale price	Statistics	Multi-purpose	Mandob Services
	Souq	NO	NO	YES	NO
	OLX	NO	NO	YES	YES
	opensooq	Yes	YES	YES	NO
	offerup	NO	NO	YES	NO
	Mandob	YES	YES	YES	YES

2.1.6.1 Analysis and limitations of existing system



App description :

Souq is a leading app in the field of selling products in different categories. Souq offers its existing in many places and was known for the best variety of products program in middle east as it's supervised by Amazon.com. the main user stream are the seller and the buyer it simply view the seller products so user can buy from them. Delivery is done by Souq and buyer must pay for it.

Features :

- Variety of products
- Discounts and promo codes
- Fast buying
- User rating

Downfalls :

- No wholesale price
- No direct contact between buyer and seller
- You can't buy too many items of same product
- No services available to buy



App description :

OLX is the largest marketplace to sell and buy with 35 million downloads and 260 million monthly active users in more than 45 countries. OLX connect sellers to buyers. OLX sellers can sell their personal and commercial products easily. Buyers can browse thousands of daily added listings for both new and second hand products and choose smartly.

Features :

- You can sell anything and everything.
- Direct contact with seller.
- Competitive prices.

Downfalls :

- Most of products are second hand.
- No wholesale prices.
- No services to buy or sell.
- You can't buy too many items of same product.
- User is limited with seller product amount.



App description :

OpenSooq is an online classified platform that allows users to freely trade property, cars, products, and services in the Middle East & North Africa region. Sellers can offer products and services to millions of users online without intermediaries. Buyers can browse a wide variety of goods and services, including automotive, Real Estate, Electronics, and much more.

Features :

- You can sell anything and everything.
- Direct contact with seller.
- Competitive prices.
- You can buy services or products

Downfalls :

- Statistics for business is not there
- Geo-location is missing



App Description :

OfferUp is a phone application that allows you to sell everything from clothing to cars. There is no fee to list. However, if you want your item to get more attention, you can pay to have it “bumped” or “featured”.

Features :

- You can sell anything and everything.
- Direct contact with seller.
- Competitive prices.

Downfalls :

- Offer up has a 9.9%, service fee
- Security of app block people randomly (according to sidehusl.com)
- Damaged items are delivered too
- No services to provide and buy



App Description :

Starting a new business or managing an existing one is becoming harder every day. Due to the major increment of Customers' requirements , Commodities Varity, supplying companies contend and the complexity of understanding the market as an individual. If an investor managed to start a new business, they will build it in view of a limited business knowledge and in best cases a business consultant will give a hand, But even if this experienced man gave you the most feasible market strategy. You won't be able to keep it up to date and robust to any market change, or any new customer requirement. Without breaking the bank.

So making the first platform that watch your business and give you the best options to improve your market potential , to keep strong presence between competitors, to be elastic to market changes , to ease business management and to understand the market of your interest as whole. Is very essential.

Features :

- Sell any service or product
- Rent and sell places with geo-location
- You can buy too many items of any product with wholesale price
- No fee for buying

2.2 Analysis of the new system

2.2.1 User Requirements

Mandatory requirements:

- 1- The user choose whether he is a seller, service provider or a buyer or someone looking for a job
- 2- The user must have a secured Signup and Login and account managing in anytime involving change his password and delete his account
- 3- Seller should EDIT or DELETE his products anytime , they can change price and quantity anytime without limitations
- 4- Buyer can view popular uploaded services and products
- 5- Buyer can make choices according to his desired category
- 6- Both seller and buyer have notifications for the products

Desirable requirements:

- 1- Mandob statistics that will be deployed in the future which support managing the store and providing information about what is sold and what is bought also have a system of recommending the best items in each category.
- 2- In-Application Chat with Seller and product
- 3- Sponsoring

Optional requirement:

- 1- Mandob cashier, Mandob is developed in cross-platform environment so it may be linked to cashier system in store to watch products and notify if anything is going to run out in the future and also automatically re-order it.

2.2.2 System Requirements

As the application works on the least versions of the ANDROID 11 because we use a working framework which is responsive to the lowest versions of Android.

IOS

The application works on IOS 9 to 14 as we use Flutter framework and it is the minimum stable version working on it.

Camera:

This app needs a mobile camera as it is very useful in taking pictures for products and services to make it easy to the user to choose the suitable Product or service and it doesn't matter which version of camera you will use, but it will be awesome if you use a good one. And if you don't have a camera it's okay! Mandob offers uploading images from Drive and internal mobile storage.

Internet connection:

This app uses a cloud server to store data so you need to be connected to the internet to fetch and view it. A part of it considered a Various categorical store so the whole data will be on the server [firebase] also you need it to enable the GPS service to upload photos

2.2.3 Functional Requirements

1- Signup:

Actor : All app Users

Precondition : open the application, client should enter a valid data in order to create an account such first name, last name, password, user name, email, and phone.

Description : open the application, client should enter a valid data in order to create an account such first name, last name, password, user name, email, and phone.

Post condition : the user should have an account and enter to home page and his profile that has its data which saved when the account created.

2- Login

Actor : All app Users

Precondition: open the application, enter login Button, and enter a valid e-mail and password.

Description : user should enter his email and his password for authorizing that he has an account.

Post condition : enter to home page then he can go to his profile that has its data which saved in the firebase.

3- Update profile

Actor: All app Users

Precondition: sign up or login, open his profile.

Description: help users that using the application for updating their information whenever they want.

post condition: the updated information will displayed in the profile and saved in the firebase

4- Log out

Actor: All app Users

Precondition: open the application, login, enter log out Button.

Description: when he make login he will go to home and then go to his profile then click on log out button. Also in any screen of application the user can access the drawer and logout successfully during application lifecycle.

post condition: user will log out successfully and could not use the application without doing login again.

5- Forget Password

Actor: All app Users

Precondition: open the application, user should has an account, click on forget password button .

Description: help the user to change password when he forget it, the user should click on forget password and then enter his e-mail to make sure that he is the correct user and sent the change password form to his mail .

Post condition : this password should be saved in the firebase to use it when you have made login before.

Other main functionalities are divided to the two types of the application users which are the service or product provider and the normal customer or seller

So we will note each separately starting with **Service or product provider** ;

6- Add service or product

Actor : service or product provider

Precondition : the user is logged in successfully and the application is connected to the internet

Description : the user can enter his products and services details and post them so any client can access it and order it if needed

Post condition : the service or product the user have entered will be cloud saved only for further access and further operations

7- Edit Service or product

Actor : Service or product provider

Precondition : the actor is logged in and they have an internet connection and a pre uploaded service or product are already posted before

Description : the user can access the pre uploaded products and services of his and edit it like editing the price or quantity...etc.

Post condition : the product or services are edited and the new values are assigned to it according to user choice.

8- Delete service or product

Actor : service or product provider

Precondition : the actor is logged in and they have an internet connection and a pre uploaded service or product are already posted before

Description : the user can access the pre uploaded products and services of his and they can delete it permanently and remove it completely from the firebase Firestore.

Post condition : The Service or product are permanently removed and can't be accessed by Buyers.

9- View Service or product

Actor : service or product provider

Precondition : the actor is logged in and they have an internet connection and a pre uploaded service or product are already posted before

Description : the user can access the pre uploaded products and services of his he can view them and scroll between them and have buttons to edit, delete or view product or service details.

Post condition : The Service or product are listed in a list view in Home page.

10 - View Details

Actor : service or product provider

Precondition : the actor is logged in and they have an internet connection and a pre uploaded service or product are already posted before and user is in his home page

Description : the user can access the pre uploaded products and services of his he can view them in details in a separate page

Post condition : The Service or product are created in a separated page with all details that will be viewed by the user.

11- View orders from regular user

Actor : service or product provider

Precondition : the actor is logged in and they have an internet connection and a pre product or service was uploaded before.

Description : the orders from the regular users are listed in the notifications page

Post condition : The orders are listed in list-view in-front of the user so he can choose to accept and start preparing them or decline it.

And Working Hand have a specific home for himself and it's only for him

1. Add work offer

Actor : Working hand

Precondition : the actor is logged in and they have an internet connection

Description : the working hand can upload his picture, cv and expected salary so the business owner can view them and ask for him later

Post condition : A new working hand is uploaded to Firestore firebase and will be viewed to business owners in the request service or product page

2. Get job notifications

Actor : Working hand

Precondition : the actor is logged in and they have an internet connection and a pre uploaded job profile is pre-uploaded

Description : the working hand will get a notification whenever a user request a working hand

Post condition : The job offer notification will appear in the notification section so he can contact the user who requested it

3. View Uploaded Job

Actor : Working hand

Precondition : the actor is logged in and they have an internet connection and a pre uploaded job profile is pre-uploaded

Description : the user can access the pre uploaded jobs and view them with buttons to edit or delete it.

Post condition : The jobs are listed in list view in home section.

4. Edit Working hand

Actor : Working hand

Precondition : the actor is logged in and they have an internet connection and a pre uploaded job profile is pre-uploaded and they are in Home page

Description : the user can access the pre uploaded jobs and Edit the data inside.

Post condition : the data is edited according to the user requirements and stored back in the Firestore.

5. Delete job

Actor : Working hand

Precondition : the actor is logged in and they have an internet connection and a pre uploaded job profile is pre-uploaded

Description : the user can delete the cv profile he previously uploaded

Post condition : The jobs is permanently deleted from the user list and from firebase Firestore.

Then we will be talking about the functionalities of the customer in according.

1. View Popular uploaded products and services

Actor : Regular user

Precondition : the actor is logged in and they have an active internet connection
Description : the user can view in a categorical list all the products and services uploaded previously according to his interests.

Post condition : The jobs are listed in list view in home section for the regular user

2. View Places for sale or rent

Actor : Regular User

Precondition : the Regular user is logged in successfully and navigated to Mandob Icon page in navigation bar and clicked on Places for sale / rent icon

Description : the user can list all places for rent and sell that the brokers previously have uploaded

Post condition : the places for sale and rent are listed in list-view in-front of the user so he can choose from them what is suitable for them.

3. View Finishing Work

Actor : Regular User

Precondition : the Regular user is logged in successfully and navigated to Mandob Icon page in navigation bar and clicked on Finishing work

Description : the user can list all finishing work providers previously have uploaded.

Post condition : The finishing works are listed in list-view in-front of the user so he can choose from them what is suitable for them.

4. View Hardworking

Actor : Regular User

Precondition : the Regular user is logged in successfully and navigated to Mandob Icon page in navigation bar and clicked on Hardware

Description : the user can list all Hardware providers previously have uploaded.

Post condition : The Hardware are listed in list-view in-front of the user so he can choose from them what is suitable for them.

5. View product for sale

Actor : Regular User

Precondition : the Regular user is logged in successfully and navigated to Mandob Icon page in navigation bar and clicked on products

Description : the user can list all products providers previously have uploaded

Post condition : The products are listed in list-view in-front of the user so he can choose from them what is suitable for them. According to the required category.

6. View Working hand

Actor : Regular User

Precondition : the Regular user is logged in successfully and navigated to Mandob Icon page in navigation bar and clicked on Working hands

Description : the user can list all working hands that need jobs that working hands previously have uploaded

Post condition : The working hands are listed in list-view in-front of the user so he can choose from them what is suitable for them. According to the required job.

7. View details

Actor : Regular User

Precondition : the Regular user is logged in successfully and navigated to Mandob Icon page in navigation bar and clicked on any category and it's listed in the screen

Description : the user can list all products and services that were uploaded lately and they can view everything that they want in more detailed page so they can order from it,

Post condition : the chosen service or product is listed in a page with more details and buttons for adding to cart.

8. Cart

Actor : Regular User

Precondition : the Regular user is logged in successfully and navigated to Mandob Icon page in navigation bar and clicked on products and viewed a service or products then they pressed add to cart

Description : the user can list all ordered products and services and view them before completing the purchase and order it.

Post condition : The products are listed in list-view in-front of the user in cart page so they can be navigated and furthermore choice will be taken.

9. Mandob Statistics

Actor : Regular User

Precondition : the Regular user is logged in successfully and navigated to Mandob statistics icon.

Description : The user can view everything he ordered and navigate the information collected by Mandob algorithms to overview his business

Post condition : all statistical data are viewed in a list-view in statistics screen.

2.2.4 Non Functional Requirements

Availability:

Our application can be used in any time and it always “uptime” is the amount of time that it is operational and available for use. This is specified because some systems are designed with expected downtime for activities like database upgrades and backups.

Efficiency:

Our application utilizes scarce resources: CPU cycles, disk space, memory, bandwidth, etc.

Flexibility:

If the team intends to increase or extend the functionality of the software after it is deployed, that should be planned from the beginning; it influences choices made during the design, development, testing, and deployment of the system.

Performance:

Our application is specify the timing characteristics of the software. Certain tasks or features are more time sensitive than others; the nonfunctional requirements should identify those software functions that have constraints on their performance.

Reliability:

Our application is capable of maintaining its performance over time. Unreliable software fails frequently, and certain tasks are more sensitive to failure.

Robustness:

Our application is able to handle error conditions gracefully, without failure. This includes a tolerance of invalid data, software defects, and unexpected operating conditions.

Scalability:

Our application is scalable has the ability to handle a wide variety of system configuration sizes. The nonfunctional requirements should specify the ways in which the system may be expected to scale up (by increasing hardware capacity, adding machines, etc.).

Usability:

Our application is Ease to use requirements address the factors that constitute the capacity of the software to be understood, learned, and used by its intended users.

2.5 Advantages of the new system

== As a Product or service provider ==

1. Mandob allows you to list all your products and services and assign prices and further more details about it
2. Thousands of regular users can access your listed products and services and order what they like from
3. You have direct contact with the customer who purchased your product so you can be carefree about delivery and no fees are applied

== As a Regular user ==

4. You can list surf the products and services from any category that meets your business need and purchase it.
5. You can request a working hand for your running project
6. All the popular items from different providers are listed in home screen according to category.

== As a working hand ==

7. You can post the required job you want to be hired for.
8. You can get job offers and accept them or refuse.

2.6 Risk and Risk Management

Risk Type	Possible Risks
Technology	The database used in the system cannot process as many transactions per second as expected. Software components that should be reused contain defects that limit their functionality.
People	It is impossible to recruit staff with the skills required. Key staff are ill and unavailable at critical times. Required training for staff is not available.
Organizational	The organization is restructured so that different management are responsible for the project
Tool	the code generated by CASE tools is inefficient. CASE tools cannot be integrated.
Requirements	Changes to requirements that require major design rework are proposed. Customers fail to understand the impact of requirements changes
Estimation	The time required to develop the software is underestimated. The rate of defect repair is underestimated. The size of the software is underestimated.

Risk	probability	Effects
Organizational financial problems force reductions in the project budget	Low	Catastrophic
It is impossible to recruit staff with the skills required for the project	High	Catastrophic
Key staff are ill at critical times in the project	Moderate	Serious
Software components that should be reused contain defects which limit their functionality.	Moderate	Serious
Changes to requirements that require major design rework are proposed	Moderate	Serious
The organization is restructured so that different managers are responsible for the project	High	Serious
The database used in the system cannot process as many transactions per second as expected	Moderate	Serious

The time required to develop the software is underestimated	High	Serious
CASE tools cannot be integrated	High	Tolerable
Customers fail to understand the impact of requirements change	Moderate	Tolerable
The rate of defect repair is underestimated	Moderate	Tolerable
The size of the software is underestimated	High	Tolerable

Risk	Strategy
Organizational financial problems	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business.
Recruitment problems	Alert customer of potential difficulties and the possibility of delays, investigate buying-in components
Staff illness	Reorganize team so that there is more overlap of work and people therefore understand each other's jobs.
Defective components	Replace potentially defective components with bought-in components of known reliability
Requirements changes	Derive traceability information to assess requirements change impact, maximize information hiding in the design
Organizational restructuring	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business.
Database performance	Investigate the possibility of buying a higher performance database

Chapter 3

Software Design

In this chapter we are going to discuss and go deeper in the internal design and coding and how the code is actually working with the database

3.1 Application Prototyping

Application prototyping is the process of creating an incomplete version of the software that is required. A prototype typically only simulates a few aspects of the final solution, and may be completely different from the final product.

Process of Proto typing :

- Identify the basic requirements, including the input and output of information desired
- Develop the initial prototype, this may only be the user interfaces
- Team Members reviews the prototype and provides feedback on additions or changes
- Revise and enhance the prototype based on Team members feedback
- Following along the SDLC with changes and editing it before production

So Mandob Team has followed a princely early prototyping to introduce a clear vision for the application



Customer: Places For ...



Customer Finishing



Customer Hardware



Customer Goods



Customer WorkingHa...



Customer cart



Mandob Statistics



Places Details



Finishing Details



Hardware Details



Goods details



WorkingHand details



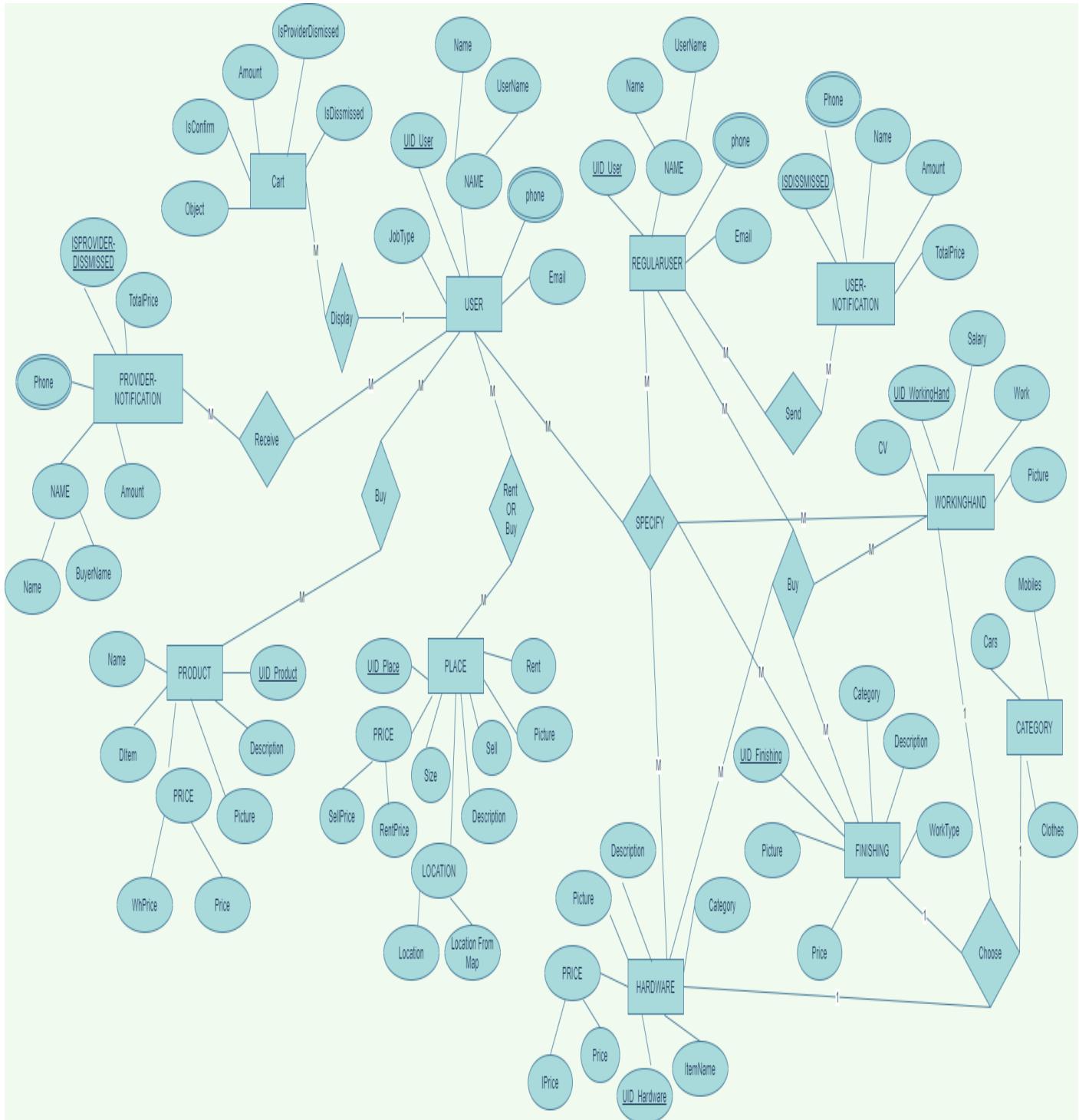
Mandob statistics 2



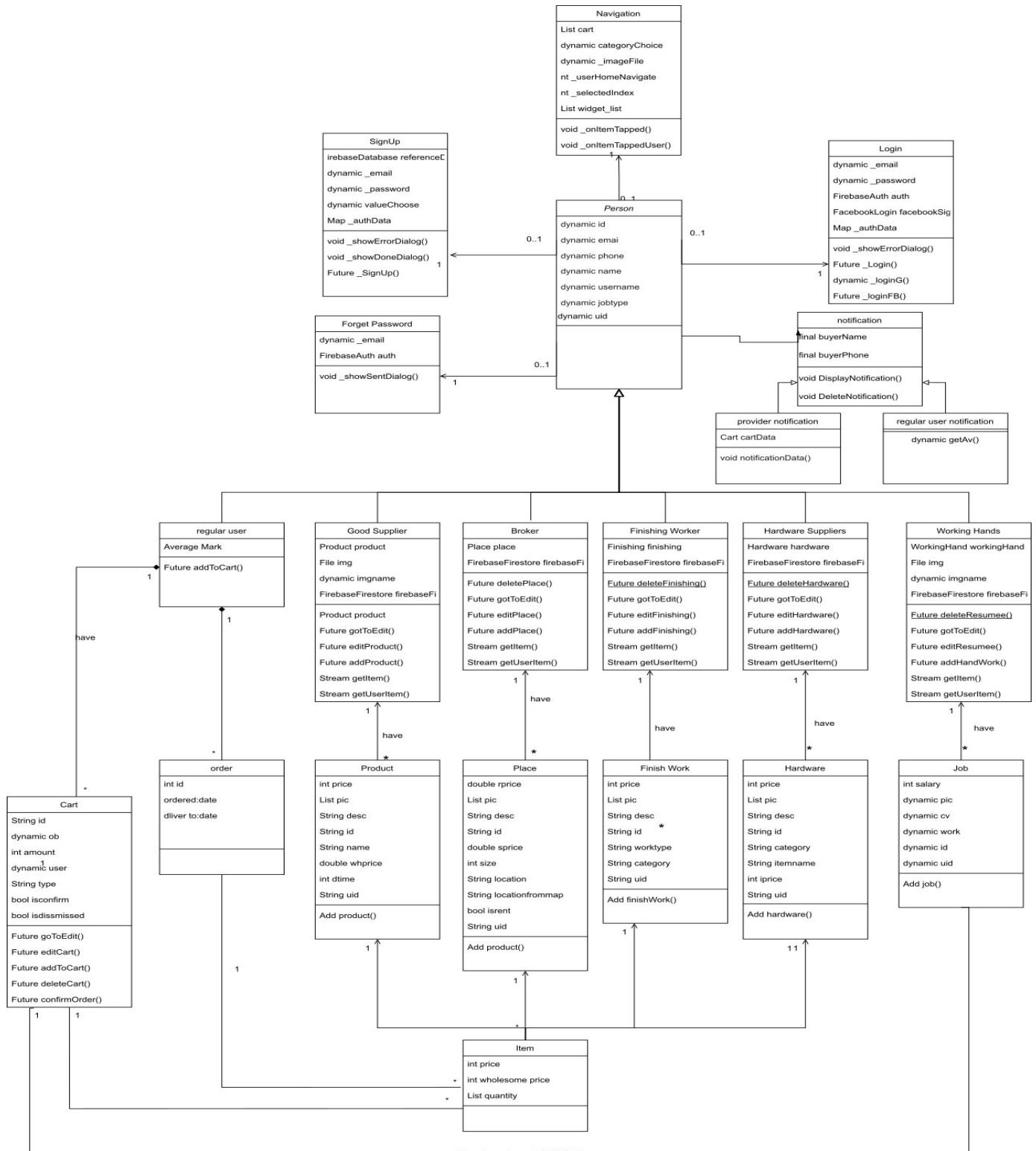
Edit Profile push screen



3.2 ERD Diagram



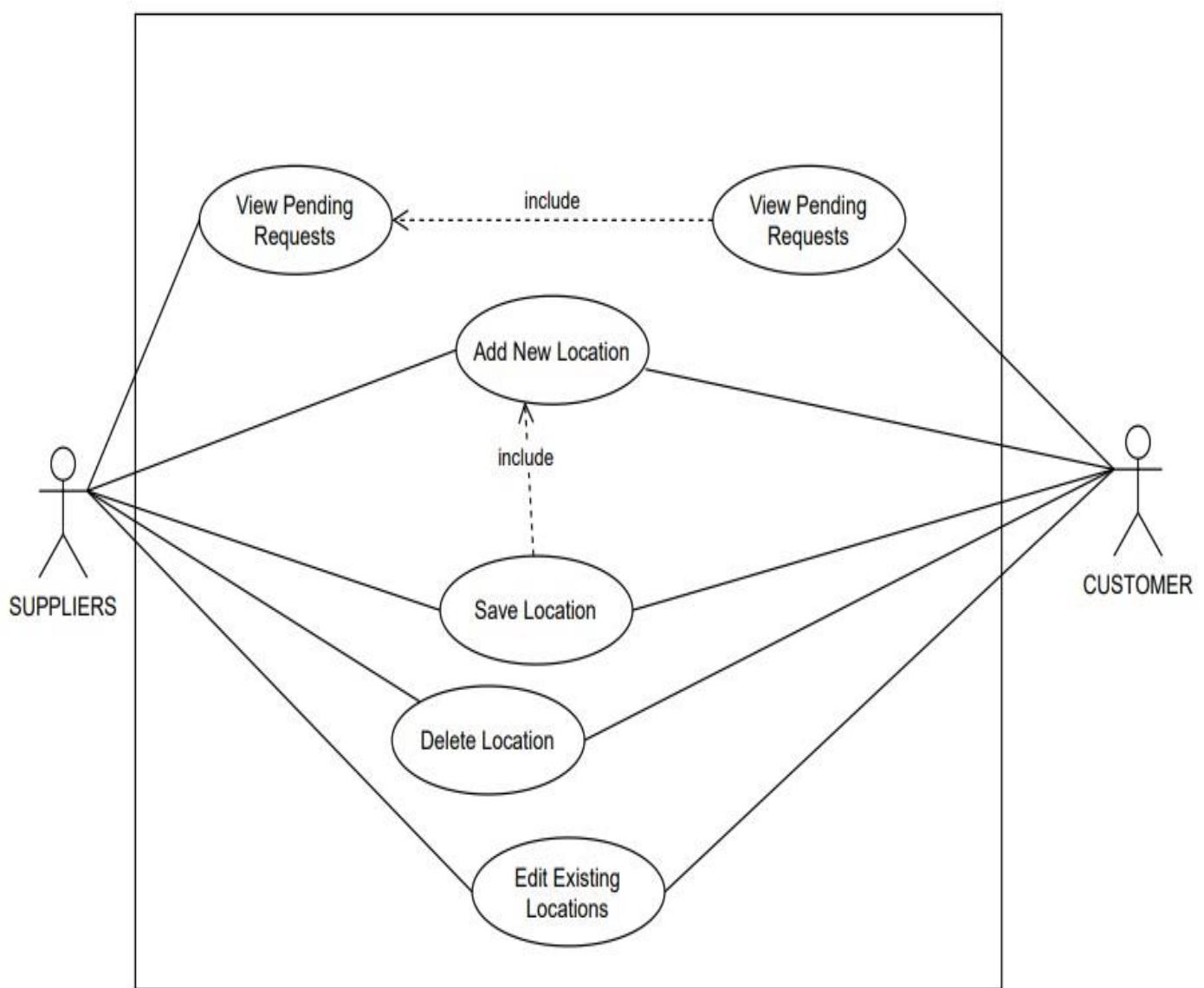
3.3 Class Diagram



3.4 Use Case Diagram

Use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how).

1-Customer & Supplier:



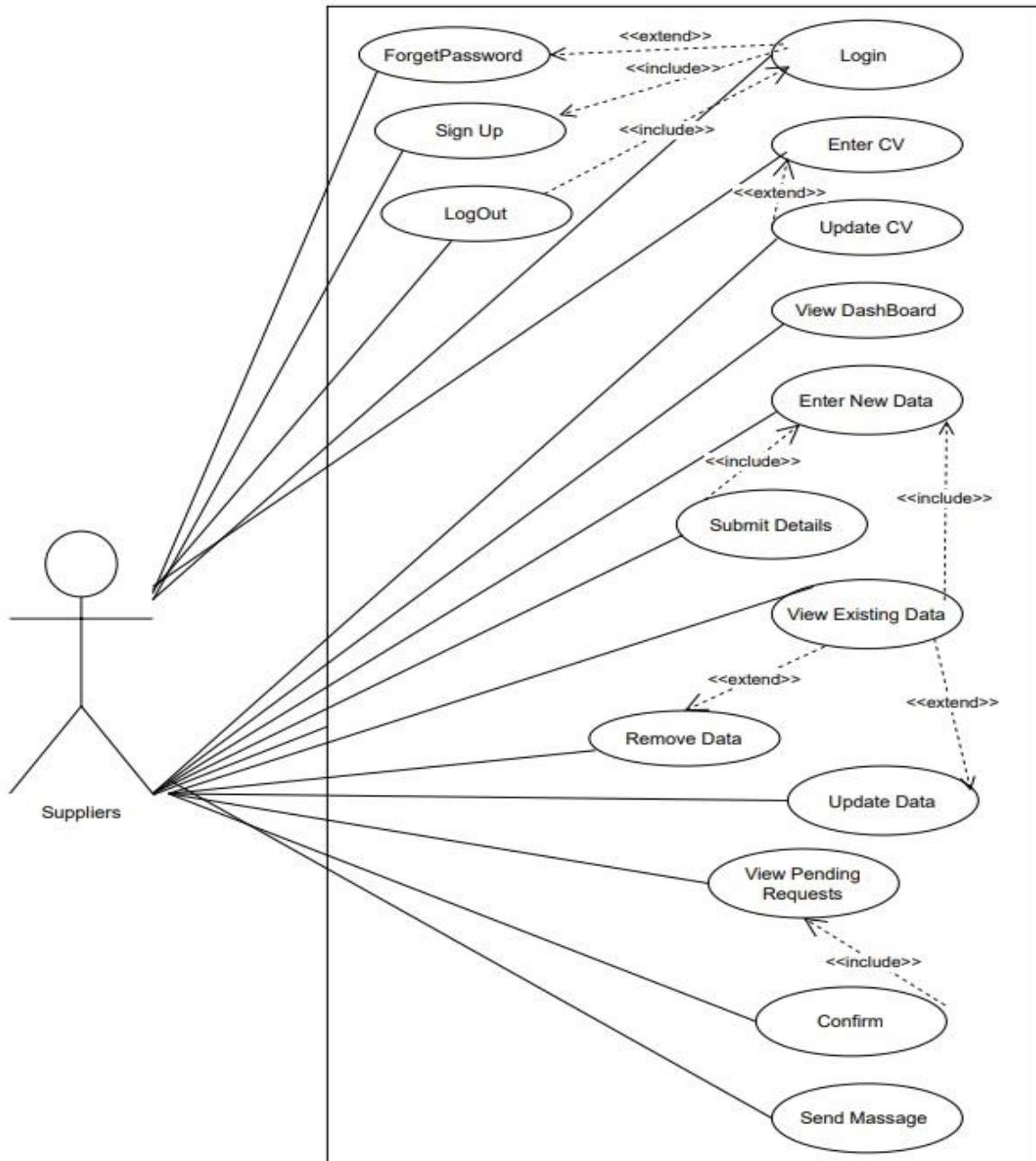
3.4 Use Case Diagram (cont.)

2-Customer:



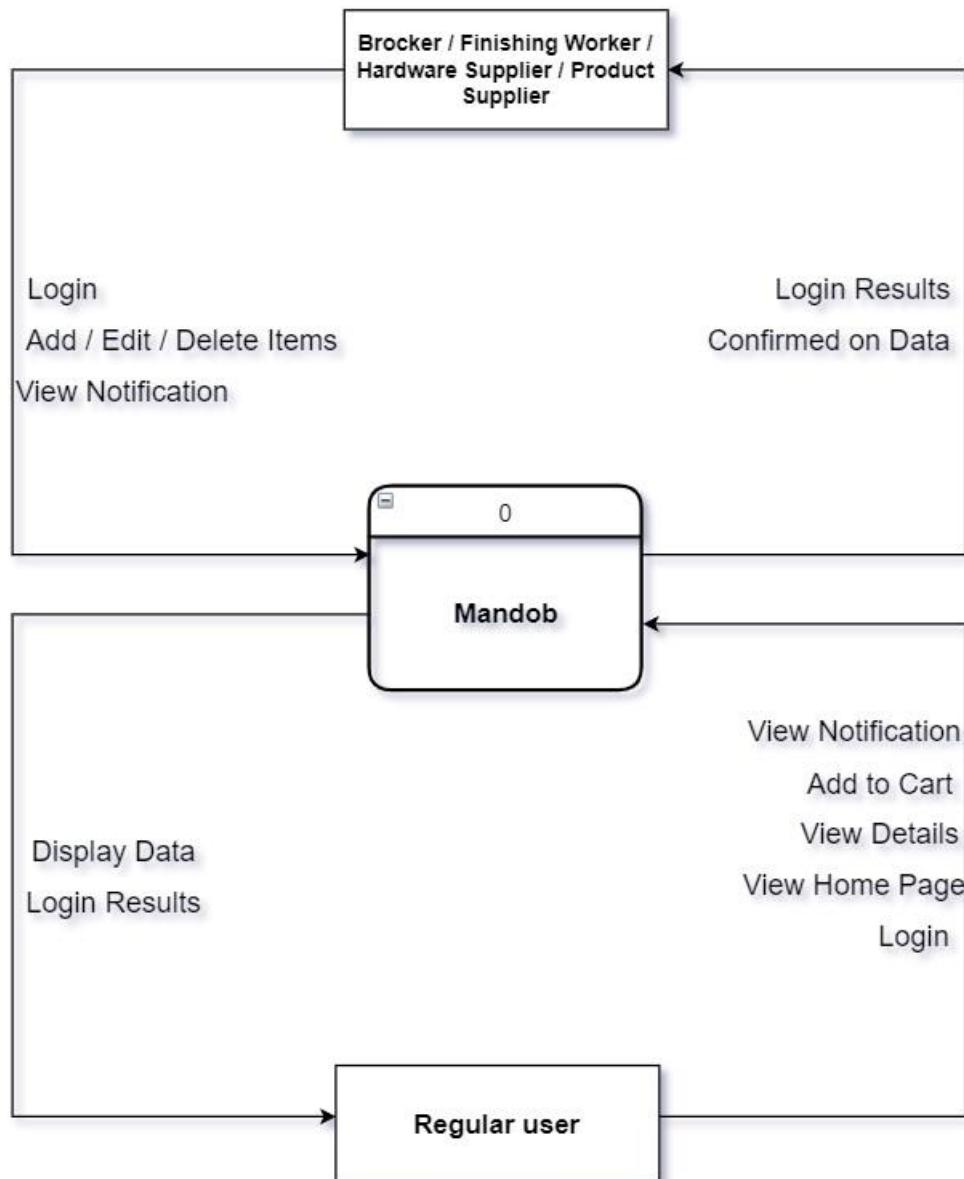
3.4 Use Case Diagram (cont.)

3- Supplier:



3.5 Context Diagram

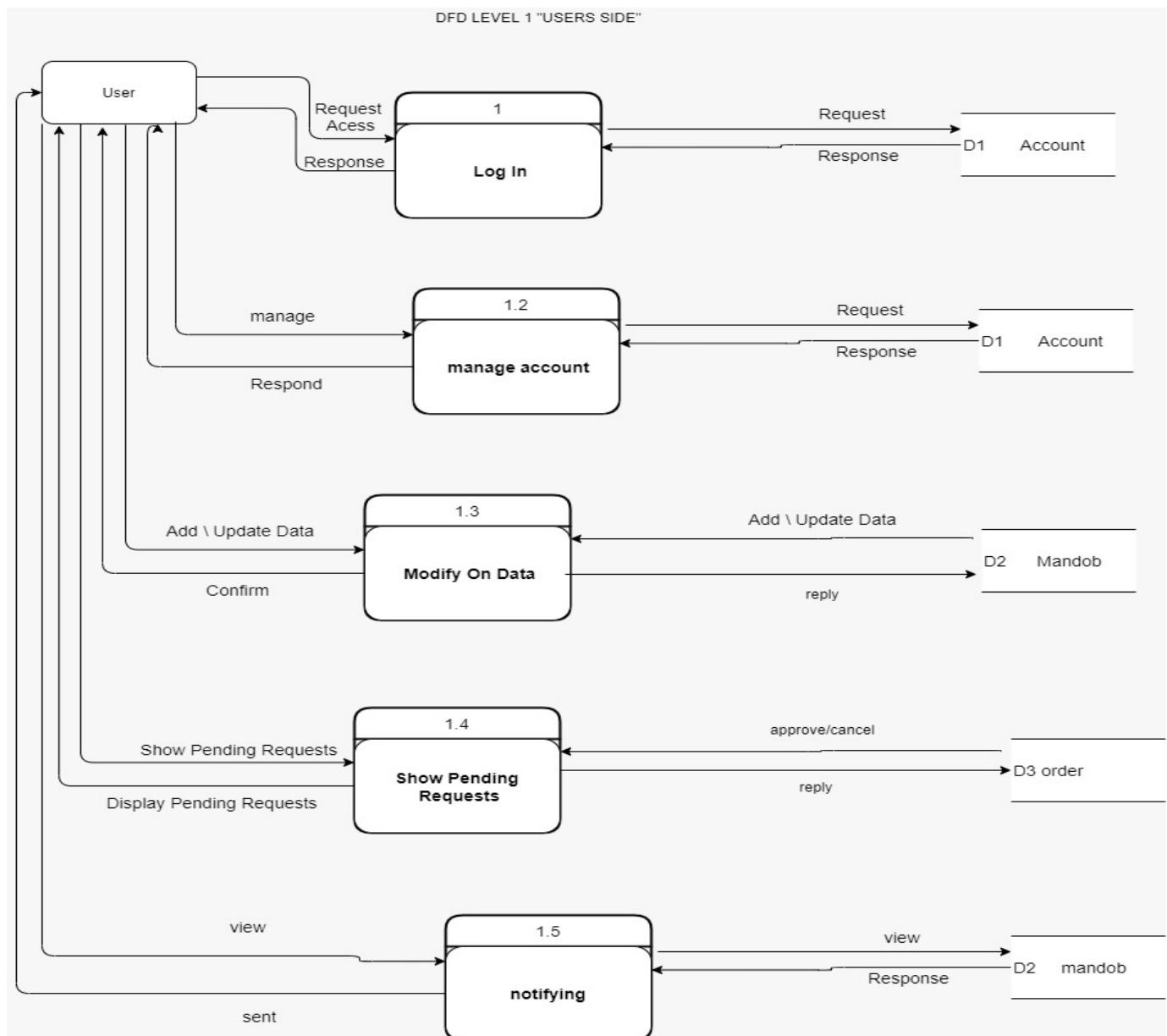
The Context Diagram shows the system under consideration as a single high-level process and then shows the relationship that the system has with other external entities (systems, organizational groups, external data stores, etc.).



3.6 DFD Diagram

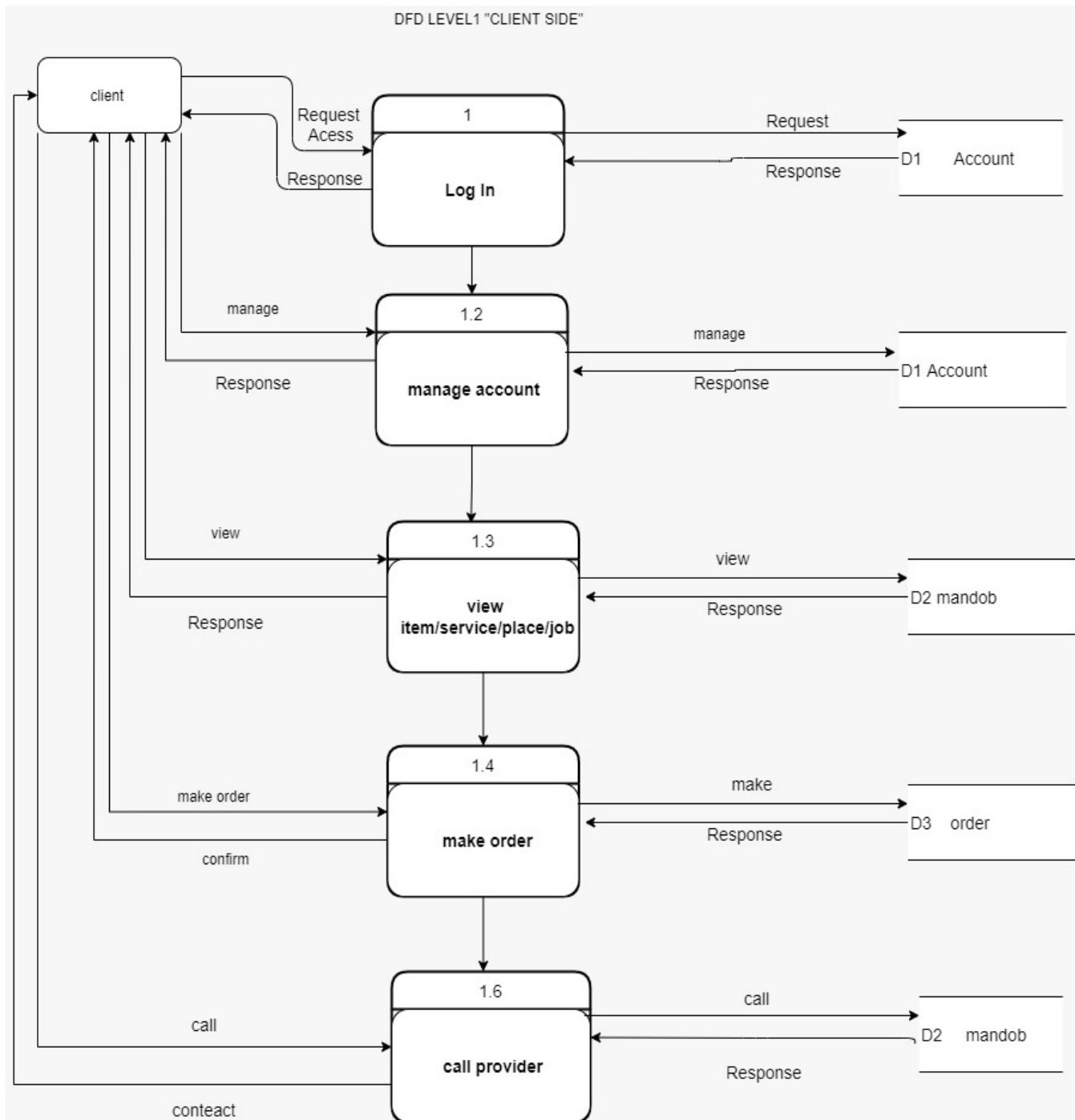
DFD graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system.

1- Level1 "Users Side":



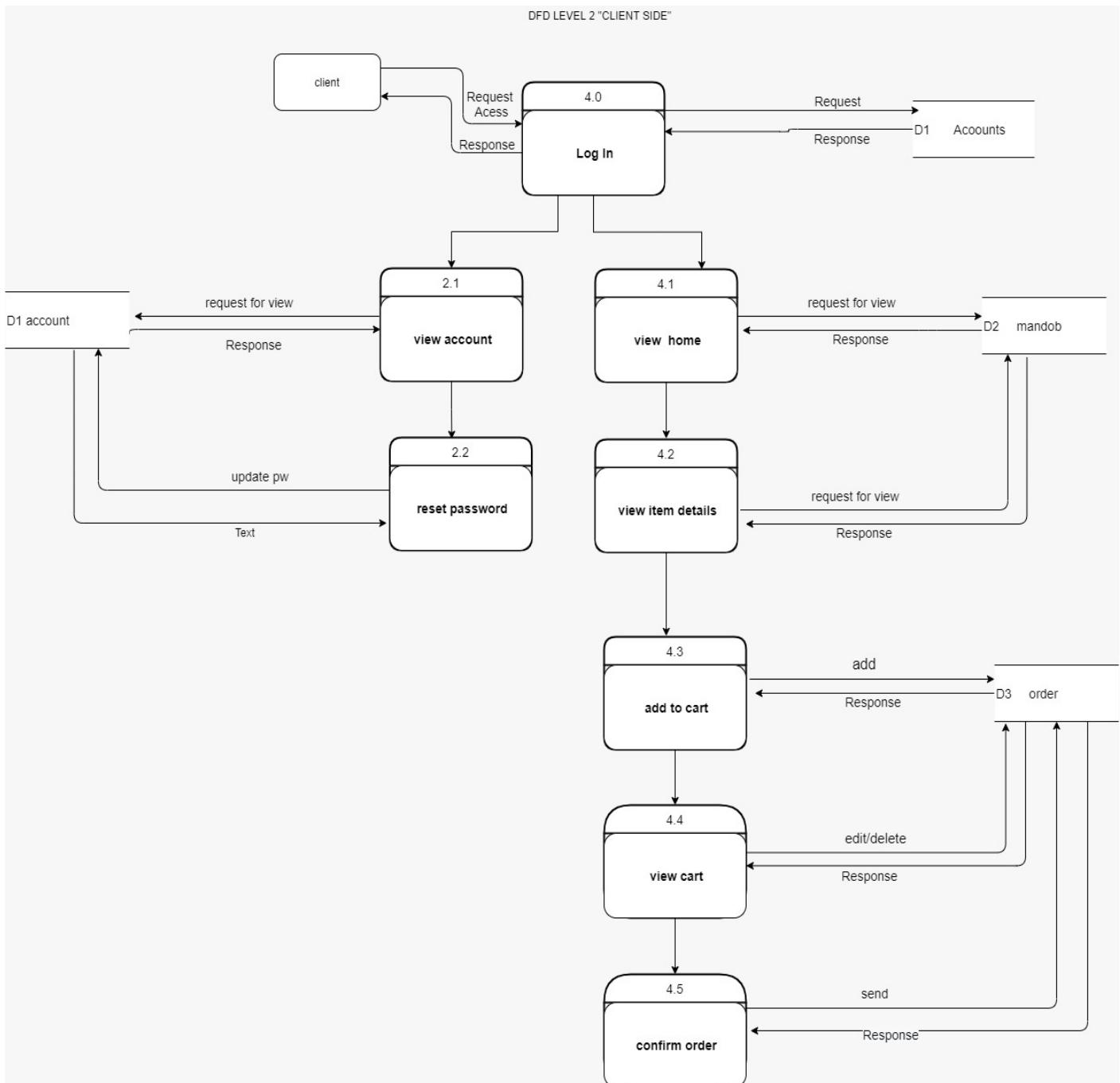
3.6 DFD Diagram (cont.)

2- Level1 "Client Side":



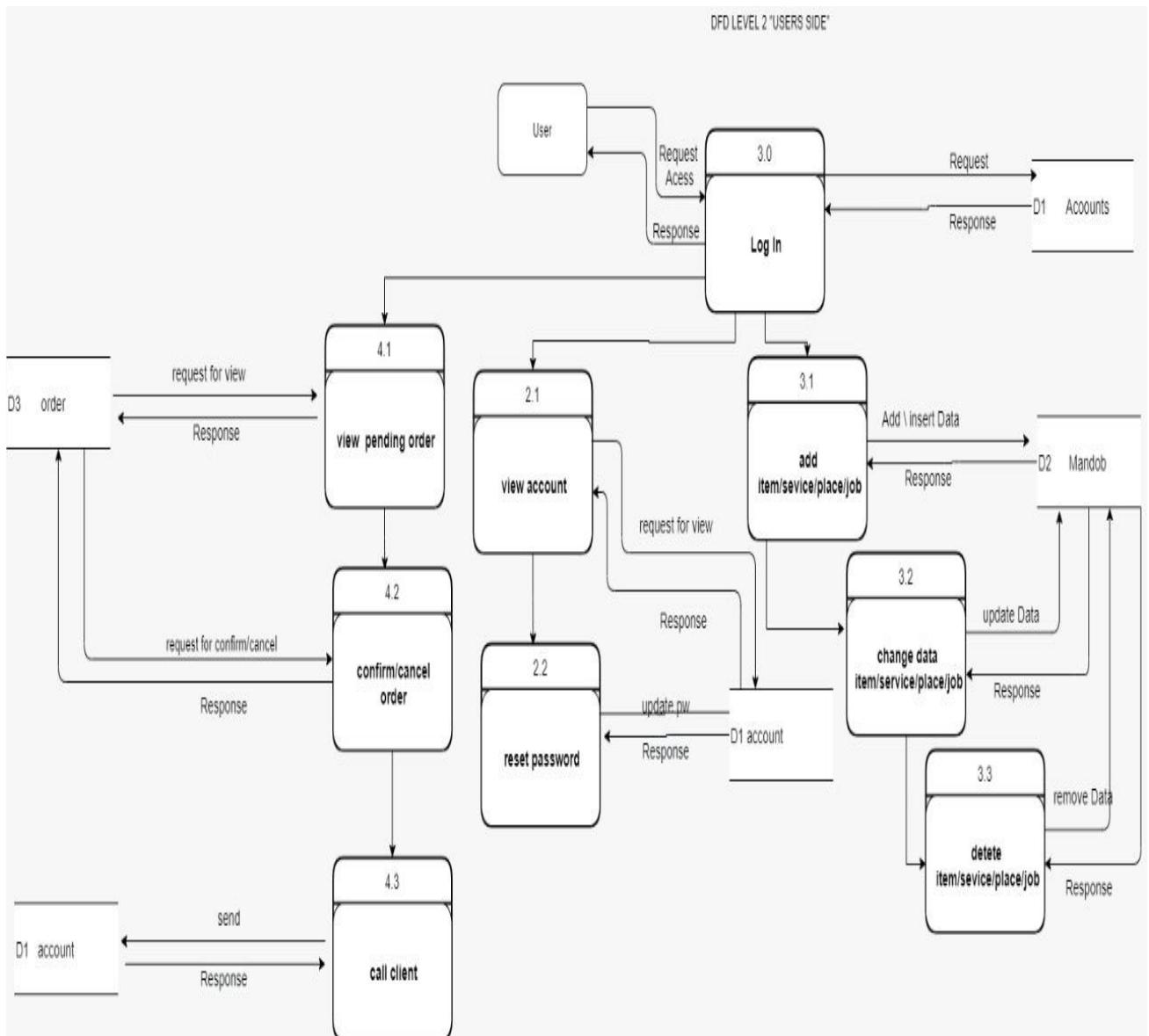
3.6 DFD Diagram (cont.)

3- Level2 “Client Side”:



3.6 DFD Diagram (cont.)

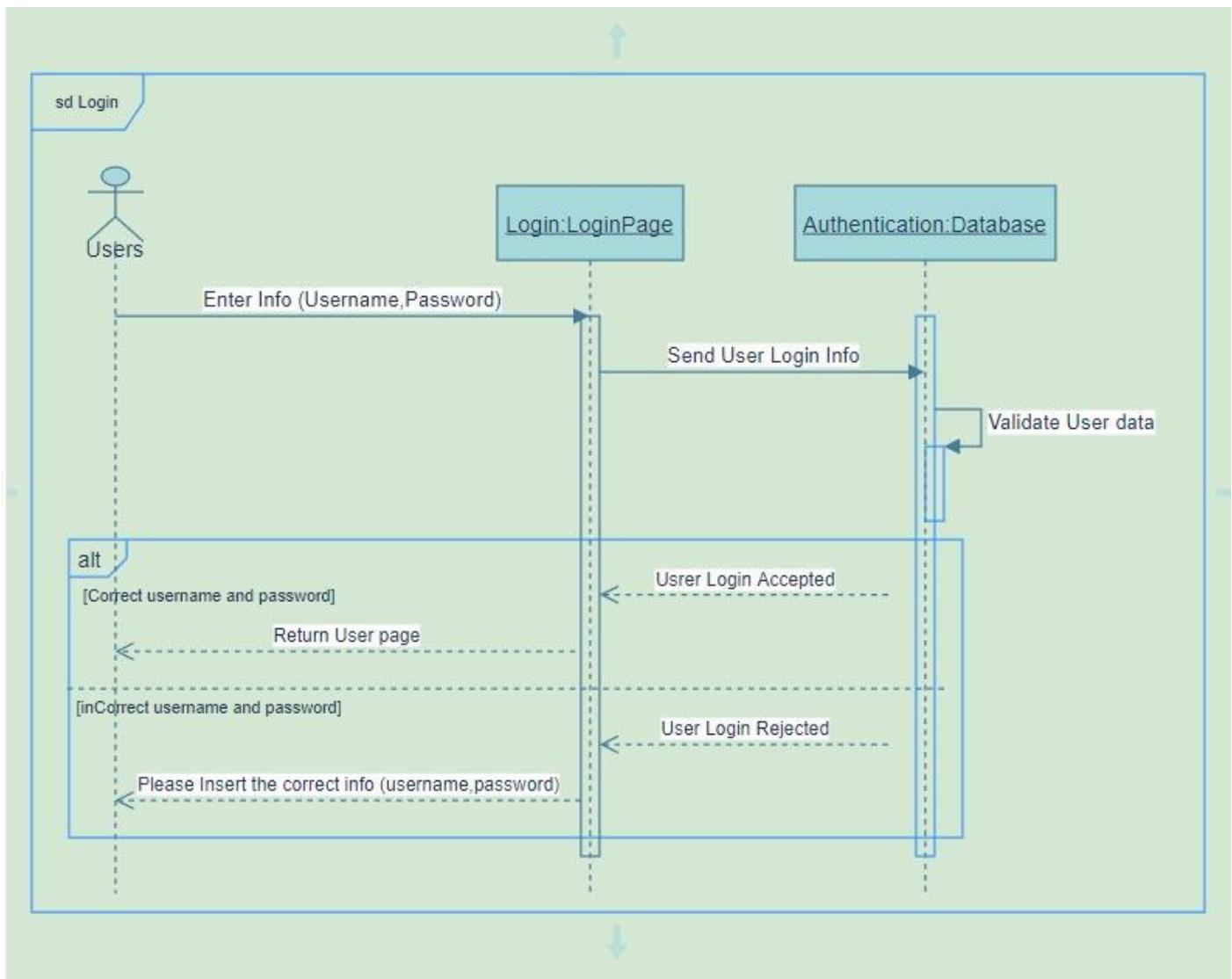
4- Level2 “Users Side”:



3.7 Sequence Diagram

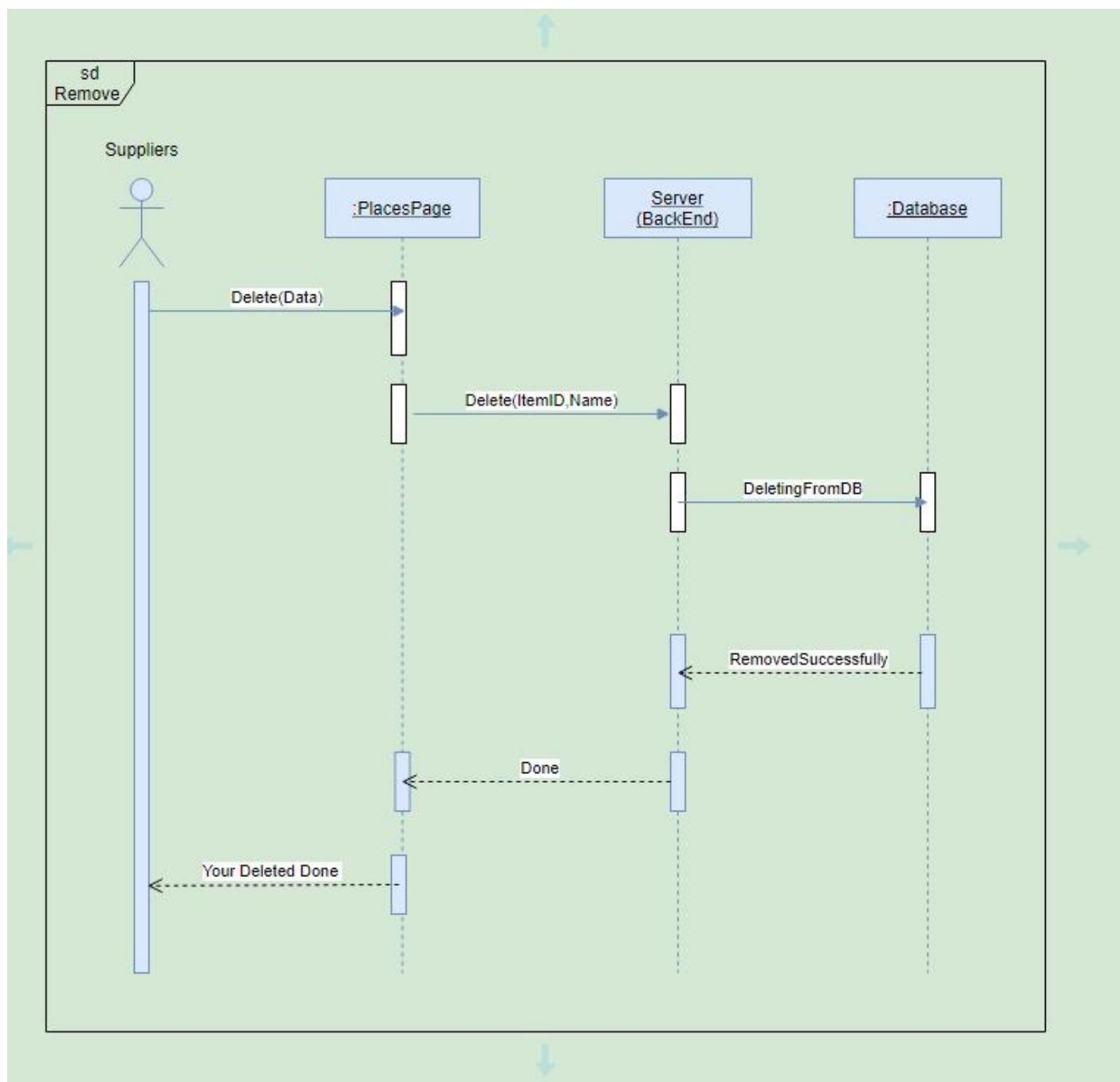
UML Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

1- Login:



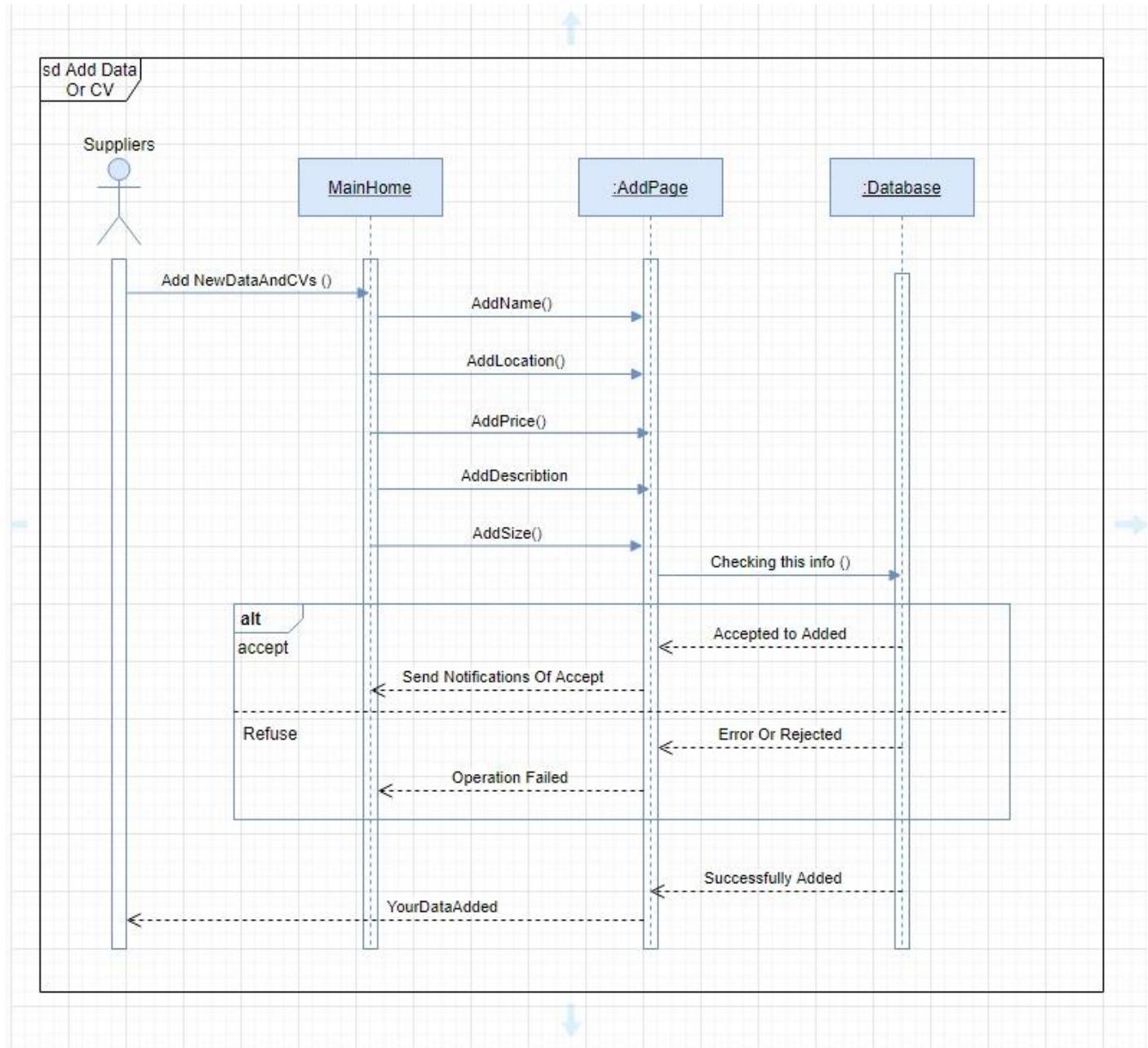
3.7 Sequence Diagram (cont.)

2- Remove:



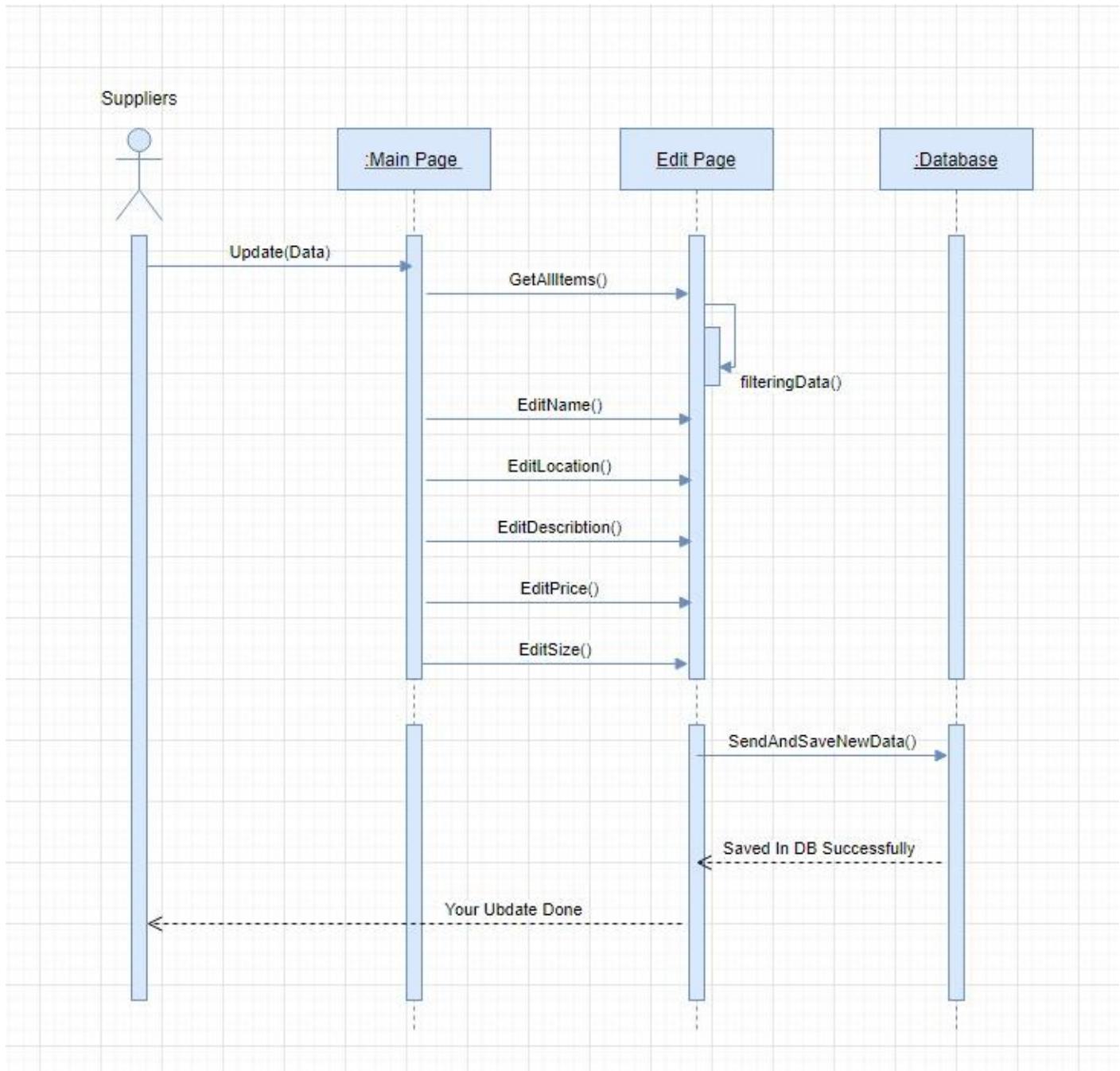
3.7 Sequence Diagram (cont.)

3- Add Data / CV:



3.7 Sequence Diagram (cont.)

4- Update:



3.8 Scenario

1) Register

- *Use case Name:* Register
- *Use case Description:* Regular user, Finishing worker, Broker, Hardware Supplier, Good Supplier and Working Hand register to app to access the system.
- *Actors:* Regular user, Finishing worker, Broker, Hardware Supplier, Good Supplier and Working Hand.
- *Pre Condition:* *user is not registered before.*
- *Post Condition:* *after successful register application open the user profile.*
- *Main scenario:*
 - 1-enter name.*
 - 2-enter username.*
 - 3-enter password.*
 - 4-enter email*
 - 5-enter phone number.*
 - 6-enter identity*
 - 7-if information valid, it is allow user to access system.*
- *Exceptions:* *if user enter invalid information system show an error message.*

2) Login

- *Use case Name:* login.
- *Use case Description:* Regular user, Finishing worker, Broker, Hardware Supplier, Good Supplier and Working Hand login to app to access it.
- *Actors:* Regular user, Finishing worker, Broker, Hardware Supplier, Good Supplier and Working Hand.
- *Pre Condition:* user have account.
- *Post Condition:* after successful login user will be entered to home page.
- *Main scenario:*

1-enter email

2-enter password

3-if email and password valid user allow access system.

- *Exceptions:* if user enters invalid email or password system show error message.

3) Add New Item

- *Use case Name:* Add New Item.
- *Use case Description:* Finishing worker, Broker, Hardware Supplier, Good Supplier and Working Hand can add new Item in page which

contains all the details about this Item such as photo, price and so on....

- **Actors:** Finishing worker, Broker, Hardware Supplier, Good Supplier and Working Hand.
- **Pre Condition:** user must be in the add page, this item won't be repeated before.
- **Post Condition:** item will be added successfully.
- **Main scenario:**

1-user must be logged in.

2-click on add new.

3-add all information about this item.

4-item will be added successfully.

4) Delete item

- **Use case Name:** delete item.
- **Use case Description:** Finishing worker, Broker, Hardware Supplier, Good Supplier and Working Hand want to delete an item.
- **Actors:** Finishing worker, Broker, Hardware Supplier, Good Supplier and Working Hand.
- **Pre Condition:** user must add item first.
- **Post Condition:** item is deleted successfully.
- **Main scenario:**

1-user must be in items page.

2-click on delete button.

3-item is deleted.

5) Edit item

- *Use case Name:* edit item.
- *Use case Description:* Finishing worker, Broker, Hardware Supplier, Good Supplier and Working Hand can edit any info of any of his items.
- *Actor:* Finishing worker, Broker, Hardware Supplier, Good Supplier and Working Hand.
- *Pre Condition:* user must be in the item he wants to edit info.
- *Post Condition:* new info will be edited successfully.
- *Main scenario:*

1-user must be in a specific item page.

2-click on the edit.

3-edit information about this item.

4-item info will be updated successfully.

6) Add to Cart

- *Use case Name:* add to cart.
- *Use case Description:* Regular user can buy any items by adding it to his cart.
- *Actor:* Regular user.
- *Pre Condition:* Regular user must be logged in.
- *Post Condition:* item added to cart successfully.
- *Main scenario:*
 - 1- Regular user *must be in a home page.*
 - 2-*click on view to see details.*
 - 2-*click on buy.*
 - 3-*item will be added to cart successfully.*

7) Show Notification

- *Use case Name:* show notification.
- *Use case Description:* Regular user, Finishing worker, Broker, Hardware Supplier, Good Supplier and Working Hand can show notification which sent to him.
- *Actor:* Regular user, Finishing worker, Broker, Hardware Supplier, Good Supplier and Working Hand.
- *Pre Condition:* user must make a move in app.
- *Post Condition:* notification showed successfully.

- *Main scenario:*
 - 1-user must be in a home page.
 - 2-click on notification button.
 - 3-notification showed successfully.

8) Confirm Order

- *Use case Name:* confirm order.
- *Use case Description:* Regular user have to confirm his order to be ready for preparing
- *Actor:* Regular user.
- *Pre Condition:* Regular user must make an order.
- *Post Condition:* order is preparing.
- *Main scenario:*
 - 1- Regular user must be in cart page.
 - 2-click on confirm button.
 - 3-item ready to prepare.

9) Delete Order

- *Use case Name:* delete order.
- *Use case Description:* Regular user want to delete an order from cart.

- *Actor:* Regular user.
- *Pre Condition:* Regular user must make an order.
- *Post Condition:* order is deleted successfully.
- *Main scenario:*
 - 1- Regular user *must be in cart page.*
 - 2-*click on delete button.*
 - 3-order is deleted.
- *Exceptions:* if Regular user confirmed this order before it won't be able to delete.

10) Logout

- *Use case Name:* Logout.
- *Use case Description:* this function it helps user to logout of the system when he click on the logout button.
- *Actors:* Regular user, Finishing worker, Broker, Hardware Supplier, Good Supplier and Working Hand.
- *Pre Condition:* user logged in the system.
- *Post Condition:* user logged out.
- *Main scenario:*
 - 1-*user clicked on logout button in any page.*
 - 2-*system returns user to the login page.*

Chapter 4 Implementation

In this chapter we're going to discuss and go deeper in Settle mobile application's implementation, and present its code and the algorithms used to build it.

• 4.1 Users Models :

Users Models are the most important kick start in the application life cycle, it gives dependencies and limitations to every user and assign what everyone should do and How it should be done.

(For The Full overview of models code Check => " Mandob .. /Models/ ")

In Mandob project we have followed MVC approach and used provider package for state management so we are introducing providers Next.

• 4.2 Users Providers

Providers for all expected users in the application to transmit data between models and firebase.

(For The Full overview of models code Check => " Mandob ..Providers/ ")

- 4.3 Signup

```
lib > Screens > SignupPage.dart > ...
1  import 'package:firebase_auth/firebase_auth.dart';
2  import 'package:firebase_core/firebase_core.dart';
3  import 'package:firebase_database.firebaseio_database.dart';
4  import 'package:flutter/material.dart';
5  import 'package:mandob/model/user.dart';
6  import 'package:mandob/provider/userprovider.dart';
7  import 'package:provider/provider.dart';
8  import 'LoginPage.dart';
9
10 class SignupPage extends StatefulWidget {
11   final FirebaseApp app;
12   SignupPage({this.app});
13
14   @override
15   _SignupPageState createState() {
16     return new _SignupPageState();
17   }
18 }
19
20 class _SignupPageState extends State<SignupPage> {
21   final referenceData = FirebaseDatabase.instance;
22   String _email, _password;
23   var valueChoose;
24
25   > List listItem = [...]
26
27   final auth = FirebaseAuth.instance;
28
29   > void _showErrorDialog(String msg) { ... }
30
31   > void _showDoneDialog(String msg) { ... }
32
33   > Map<String, String> _authData = { ... }
34
35   > Future<void> _SignUp() async {
36     ...
37
38     @override
39     Widget build(BuildContext context) { ... }
40
41   }
42
43 }
```

Signup page

(For The Full overview of Signup page code Check => “ Mandob .. /Screens/ “)

- **4.4 Login including (Facebook and Google)**

```
lib / Screens / LoginPage.dart / _LoginPageState  
1 import 'package:flutter/cupertino.dart';  
2 import 'package:flutter/material.dart';  
3 import 'package:flutter_facebook_login/flutter_facebook_login.dart';  
4 import 'package:google_sign_in/google_sign_in.dart';  
5 import 'package:firebase_auth/firebase_auth.dart';  
6 import 'package:mandob/Screens/ForgetPassword.dart';  
7 import 'package:mandob/Screens/SliderPage.dart';  
8 import 'dart:async';  
9 import 'SignupPage.dart';  
10 import 'package:flutter_session/flutter_session.dart';  
11  
12 class LoginPage extends StatefulWidget {  
13     @override  
14     _LoginPageState createState() {  
15         return new _LoginPageState();  
16     }  
17 }  
18 class _LoginPageState extends State<LoginPage> {  
19     String _email , _password;  
20     final auth = FirebaseAuth.instance;  
21  
22     void _showErrorDialog(String msg) { ... }  
23     Future<void> _Login() async { ... }  
24     static final FacebookLogin facebookSignIn = new FacebookLogin();  
25     GoogleSignIn _googleSignIn = GoogleSignIn(scopes: ['email']);  
26     _loginG() async { ... }  
27     Future<Null> _loginFB() async { ... }  
28  
29     @override  
30     Widget build(BuildContext context) { ... }  
31 }  
32  
33
```

Login page

(For The Full overview of models code Check => “ Mandob ..Screens/ ”)

• 4.5 Add service or Product

```
1 import 'package:firebase_auth/firebase_auth.dart';
2 import 'package:flutter/material.dart';
3 import 'package:image_picker/image_picker.dart';
4 import 'package:mandob/model/place.dart';
5 import 'package:mandob/provider/placeprovider.dart';
6 import 'package:mandob/provider/uploaddata.dart';
7 import 'package:mandob/widgets/customwidgets.dart';
8 import 'package:mandob/widgets/customtextfield.dart';
9 import 'package:mandob/theme/fonticon.dart';
10 import 'package:provider/provider.dart';
11 import 'package:google_maps_flutter/google_maps_flutter.dart';
12 import 'package:google_maps_place_picker/google_maps_place_picker.dart';
13
14 enum SingingCharacter { rent, sale }
15
16 class PlaceSceen extends StatefulWidget {
17   @override
18   _PlaceSceenState createState() => _PlaceSceenState();
19 }
20 class _PlaceSceenState extends State<PlaceSceen> {
21   TextEditingController describtion = TextEditingController();
22   TextEditingController sprice = TextEditingController();
23   TextEditingController rprice = TextEditingController();
24   TextEditingController size = TextEditingController();
25   TextEditingController location = TextEditingController();
26   TextEditingController locationfrommap = TextEditingController();
27   var nn;
28   final imgurl =
29     "https://image.freepik.com/free-photo/paperboard-texture_95678-72.jpg";
30   String img1;
31   String img2;
32   String img3;
33   String img4;
34   String img5;
35   final skey = GlobalKey<ScaffoldState>();
36   int i = 0;
37   SingingCharacter _character = SingingCharacter.rent;
38   bool isedit = false;
39   Place ob;
40
41   @override
42   Widget build(BuildContext context) {
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
59 }
```

Adding Services or products

(For The Full overview of Adding products and services code Check => “ Mandob ..Screens/ ”)

```
2  
3   import 'package:firebase_auth/firebase_auth.dart';  
4   import 'package:flutter/material.dart';  
5   import 'package:image_picker/image_picker.dart';  
6   import 'package:mandob/model/product.dart';  
7   import 'package:mandob/provider/productprovider.dart';  
8   import 'package:mandob/provider/uploaddata.dart';  
9   import 'package:mandob/widgets/customwidgets.dart';  
10  import 'package:mandob/widgets/customtextfield.dart';  
11  import 'package:mandob/theme/fonticon.dart';  
12  import 'package:provider/provider.dart';  
13  import 'package:textfield_search/textfield_search.dart';  
14  List cat = ["cars", "mobiles", "clothes"];  
15  class ProductScreen extends StatefulWidget {  
16    @override  
17    _ProductScreenState createState() => _ProductScreenState();  
18  }  
19  
20  class _ProductScreenState extends State<ProductScreen> {  
21    TextEditingController describtion = TextEditingController();  
22    TextEditingController category = TextEditingController();  
23    TextEditingController price = TextEditingController();  
24    TextEditingController name = TextEditingController();  
25    TextEditingController deliverytime = TextEditingController();  
26    TextEditingController wholesomeprice = TextEditingController();  
27    final imgurl =  
28      "https://image.freepik.com/free-photo/paperboard-texture_95678-72.jpg";  
29    String img1;  
30    String img2;  
31    String img3;  
32    String img4;  
33    String img5;  
34    final skey = GlobalKey<ScaffoldState>();  
35    int i = 0;  
36    bool isedit = false;  
37    Product ob;  
38    @override  
39  > Widget build(BuildContext context) { ...  
331  }  
332
```

- **4.6 View Services and products details**

```

1 import 'package:flutter/material.dart';
2 //import 'package:flutter_google_maps/flutter_google_maps.dart';
3 import 'package:google_maps_place_picker/providers/place_provider.dart';
4 import 'package:mandob/provider/placeprovider.dart';
5 import 'package:provider/provider.dart';
6 import 'package:google_maps_flutter/google_maps_flutter.dart';
7
8 class PlaceScreendetails extends StatelessWidget {
9   @override
10  Widget build(BuildContext context) {
11    final place = Provider.of<PlacesProvider>(context, listen: false);
12    var lp = place.place.pic;
13    var lat = place.place.locationfrommap.split("/")[1];
14    var lng = place.place.locationfrommap.split("/")[2];
15    var desc = place.place.desc;
16    var sp = place.place.sprice;
17    var rp = place.place.rprice;
18    var s = place.place.size;
19    var isr = place.place.isrent;
20    > List<Marker> x = [...];
21    place.place = null;
22    > return Scaffold( // Scaffold ...
23    );
24  }
25}
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123

```

(For The Full overview Viewing Details of items listed Check => “ Mandob .. /Screens/ ”)

```

1 import 'package:flutter/material.dart';
2 //import 'package:flutter_google_maps/flutter_google_maps.dart';
3 import 'package:mandob/provider/productprovider.dart';
4 import 'package:provider/provider.dart';
5 import 'package:google_maps_flutter/google_maps_flutter.dart';
6
7 class ProductScreendetails extends StatelessWidget {
8   @override
9   Widget build(BuildContext context) {
10    final product = Provider.of<ProductProvider>(context, listen: false);
11    var lp = product.product.pic;
12    var desc = product.product.desc;
13    var price = product.product.price;
14    var dt = product.product.dtime;
15    var name = product.product.name;
16    var wPrice = product.product.whprice;
17    product.product = null;
18    > return Scaffold(
19      appBar: AppBar(),
20      body: SingleChildScrollView(
21        child: Column(
22          children: [
23            > SizedBox( // SizedBox ...
24            > Padding( // Padding ...
25            > Padding( // Padding ...
26            > Padding( // Padding ...
27            > Padding( // Padding ...
28            > Padding( // Padding ...
29            > Padding( // Padding ...
30            > Padding( // Padding ...
31            > ],
32            > ), // Column // SingleChildScrollView
33            > ); // Scaffold
34          );
35        );
36      );
37    );
38  }
39}
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123

```

- **4.7 Delete Function**

- **4.8 Update Function**

Delete and Update functions are merged with Users Providers see **4.3.2** for more details

- **4.9 Get notifications**

```
6 import 'package:mandob/model/place.dart';
7 import 'package:mandob/model/product.dart';
8 import 'package:mandob/model/workinghand.dart';
9 import 'package:mandob/provider/cartprovider.dart';
10 import 'package:mandob/provider/userprovider.dart';
11 import 'package:mandob/theme/fonticon.dart';
12 import 'package:provider/provider.dart';
13 import 'package:flutter_slidable/flutter_slidable.dart';
14 import 'package:url_launcher/url_launcher.dart';
15
16 class ProviderNotification extends StatelessWidget {
17   FirebaseFirestore firebaseFirestore = FirebaseFirestore.instance;
18
19   final imgurl =
20     "https://image.freepik.com/free-photo/paperboard-texture_95678-72.jpg";
21
22   @override
23   Widget build(BuildContext context) {
24     var currentUserID = Provider.of<UserProvider>(context, listen: false)
25       .userprofile
26       .toJson()["uid"];
27     var currentUserName = Provider.of<UserProvider>(context, listen: false)
28       .userprofile
29       .toJson()["name"];
30     var currentUserPhone = Provider.of<UserProvider>(context, listen: false)
31       .userprofile
32       .toJson()["phone"];
33
34     print("$currentUserID $currentUserName $currentUserPhone");
35
36     return Scaffold(
37       appBar: AppBar( // AppBar ...
38       body: new Container( // Container ...
39     ); // Scaffold
40   }
41 }
```

- **4.10 View services and products overview**

```
lib > Screens > finishingdetails.dart > ...
1   import 'package:flutter/material.dart';
2   //import 'package:flutter_google_maps/flutter_google_maps.dart';
3   import 'package:mandob/provider/finishingprovider.dart';
4   import 'package:provider/provider.dart';
5   import 'package:google_maps_flutter/google_maps_flutter.dart';
6
7   class FinishingDetails extends StatelessWidget {
8     @override
9     Widget build(BuildContext context) {
10       final finishing = Provider.of<FinishingProvider>(context, listen: false);
11       var lp = finishing.finishing.pic;
12
13       var desc = finishing.finishing.desc;
14       var price = finishing.finishing.price;
15       var cat = finishing.finishing.category;
16       var worktype = finishing.finishing.worktype;
17       finishing.finishing = null;
18
19       return Scaffold(
20         appBar: AppBar(),
21         body: SingleChildScrollView( // Column // SingleChildScrollView ...
109       ); // Scaffold
110     }
111   }
112 }
```

(For The Full overview Viewing Details of any service or product Check => “ Mandob .. /Screens/ ”)

- **4.11 Get latest added Products and services**

(For The Full overview of Popular services and products check => “ Mandob .. /Screens/ ”)

```
1 > import 'package:flutter/material.dart'; ...
19   final imgurl =
20   | "https://image.freepik.com/free-photo/paperboard-texture_95678-72.jpg";
21   class UserHomePage extends StatefulWidget {
22     @override
23     _UserHomePageState createState() => _UserHomePageState();
24   }
25   class _UserHomePageState extends State<UserHomePage> {
26     final GlobalKey<ScaffoldState> _scaffoldKey = new GlobalKey<ScaffoldState>();
27     @override
28     Widget build(BuildContext context) { ...
487   >   Widget promoCard(image, String displayinfo, Function function) { ...
539   >   String getAv(List nn) { ...
548   }
549 }
```

- **4.12 View details of services and products**

```
1 import 'package:flutter/material.dart';
2 //import 'package:flutter_google_maps/flutter_google_maps.dart';
3 import 'package:google_maps_place_picker/providers/place_provider.dart';
4 import 'package:mandob/provider/placeprovider.dart';
5 import 'package:provider/provider.dart';
6 import 'package:google_maps_flutter/google_maps_flutter.dart';
7
8 class PlaceScreendetails extends StatelessWidget {
9   @override
10  Widget build(BuildContext context) {
11    final place = Provider.of<PlacesProvider>(context, listen: false);
12    var lp = place.place.pic;
13    var lat = place.place.locationfrommap.split("/")[1];
14    var lng = place.place.locationfrommap.split("/")[2];
15    var desc = place.place.desc;
16    var sp = place.place.sprice;
17    var rp = place.place.rprice;
18    var s = place.place.size;
19    var isr = place.place.isrent;
20  >    List<Marker> x = [...];
21    place.place = null;
22  >    return Scaffold( // Scaffold ...
23  >      body: ...
24  >    );
25  >  }
26  >
27  >
```

(For The Full overview Viewing Details of items listed Check => “ Mandob ..Screens/ ”)

• 4.13 App Navigation

```
1 > import 'package:convex_bottom_bar/convex_bottom_bar.dart'; ...
23
24 // This is the stateful widget that the main application instantiates.
25 class AppNaigation extends StatefulWidget {
26   @override
27   State<AppNaigation> createState() => _AppNaigationState();
28 }
29
30 class _AppNaigationState extends State<AppNaigation> {
31   int _userHomeNavigate = 2;
32   int _selectedIndex = 1;
33   Users users;
34   static const TextStyle optionStyle =
35     TextStyle(fontSize: 30, fontWeight: FontWeight.bold);
36 > static List<Widget> _list = <Widget>[ // <Widget>[] ...
40
41 > void _onItemTapped(int index) { ...
46
47   void _onItemTappedUser(int index) {
48     setState(() {
49       _userHomeNavigate = index;
50     });
51   }
52
53   @override
54   void initState() {
55 >     Future.delayed(Duration(seconds: 1), () async { // Future.delayed ...
105
106     super.initState();
107   }
108
109   @override
110 >   Widget build(BuildContext context) { ...
146
147 }
```

- **4.14 Get notifications about purchase**

```
> import 'package:cloud_firestore/cloud_firestore.dart'; ...

class ProviderNotification extends StatelessWidget {
  FirebaseFirestore firebaseFirestore = FirebaseFirestore.instance;

  final imgurl =
    "https://image.freepik.com/free-photo/paperboard-texture_95678-72.jpg";

  @override
  Widget build(BuildContext context) {
    var currentUserID = Provider.of<UserProvider>(context, listen: false)
      .userprofile
      .toJson()["uid"];
    var currentUserName = Provider.of<UserProvider>(context, listen: false)
      .userprofile
      .toJson()["name"];
    var currentUserPhone = Provider.of<UserProvider>(context, listen: false)
      .userprofile
      .toJson()["phone"];

    print("$currentUserID $currentUserNome $currentUserPhone");

    return Scaffold(
      appBar: AppBar( // AppBar ...
      body: new Container( // Container ...
        );
      ); // Scaffold
    }
}
```

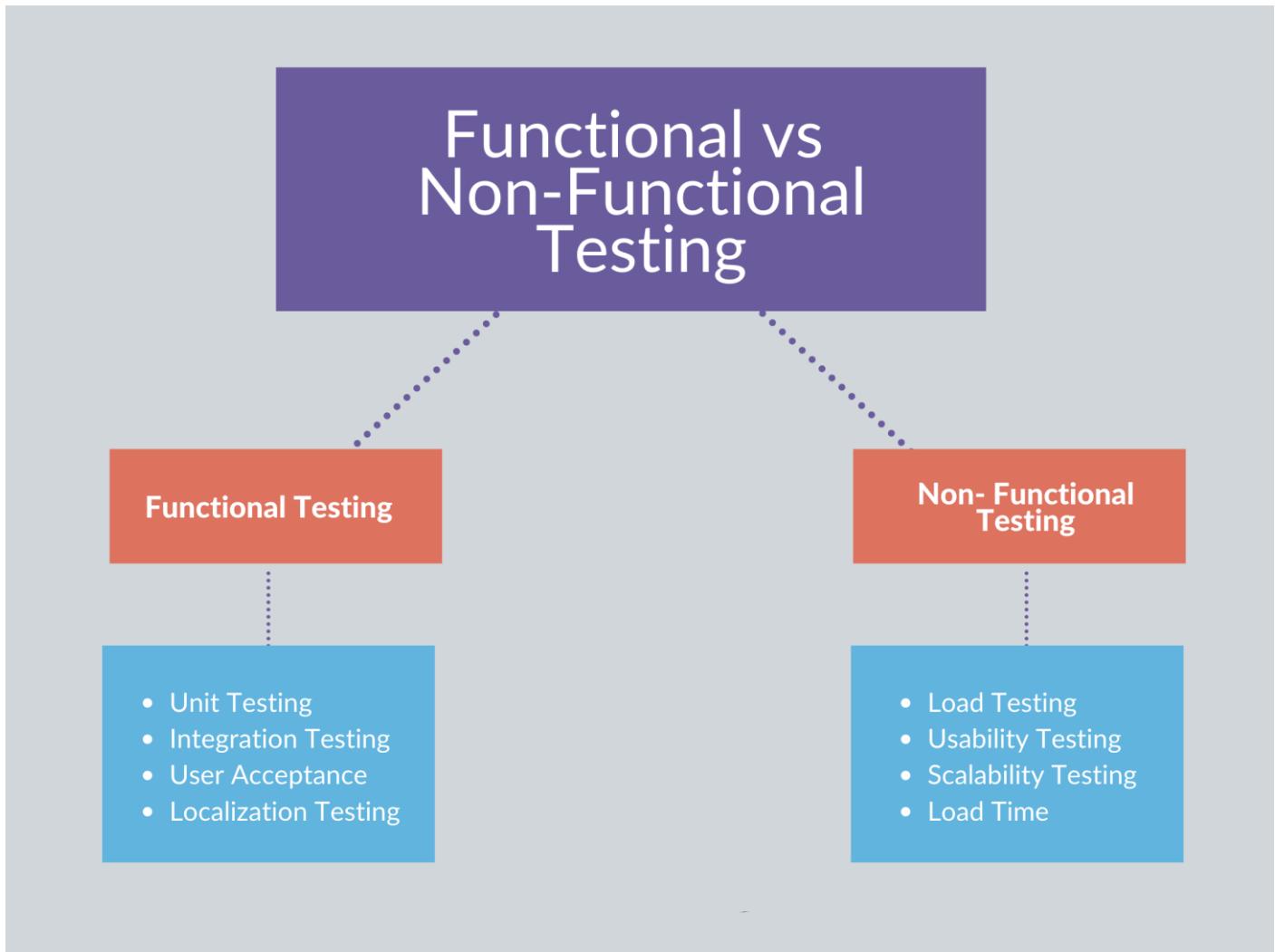
- **4.15 Navigate to cart and edit order**

```
1 > import 'package:cloud_firestore/cloud_firestore.dart'; ...
25
26   final imgurl =
27     "https://image.freepik.com/free-photo/paperboard-texture_95678-72.jpg";
28
29 class UserCart extends StatelessWidget {
30   @override
31   Widget build(BuildContext context) {
32 >     print(...);
33     return Scaffold(
34       appBar: AppBar(),
35 >       body: Container( // Container ...
36 >         child: Column( // Column ...
37 >           children: [
38 >             Container( // Container ...
39 >               child: Image.network(imgurl),
40 >             ),
41 >             Container( // Container ...
42 >               child: Text("Cart"),
43 >             ),
44 >           ],
45 >         ),
46 >       );
47 >     }
48 >   }
49 > }
50
51 > void main() {
52 >   runApp(MyApp());
53 > }
54 >
55 > class MyApp extends StatelessWidget {
56 >   @override
57 >   Widget build(BuildContext context) {
58 >     return MaterialApp(
59 >       title: 'Flutter Demo',
60 >       theme: ThemeData(
61 >         primarySwatch: Colors.blue,
62 >       ),
63 >       home: MyHomePage(),
64 >     );
65 >   }
66 > }
67 >
68 > class MyHomePage extends StatefulWidget {
69 >   @override
70 >   _MyHomePageState createState() => _MyHomePageState();
71 > }
72 >
73 > class _MyHomePageState extends State<MyHomePage> {
74 >   @override
75 >   void initState() {
76 >     super.initState();
77 >     FirebaseFirestore.instance
78 >       .collection('users')
79 >       .doc('user1')
80 >       .get()
81 >       .then((value) =>
82 >         setState(() {
83 >           user = value.data();
84 >         }));
85 >   }
86 >
87 >   User? user;
88 >
89 >   void updateCart() {
90 >     FirebaseFirestore.instance
91 >       .collection('users')
92 >       .doc('user1')
93 >       .update({
94 >         'cart': FieldValue.arrayUnion([
95 >           {'product': 'apple', 'quantity': 1, 'price': 100},
96 >           {'product': 'banana', 'quantity': 2, 'price': 200}
97 >         ])
98 >       });
99 >   }
100 >
```

Chapter 5

Testing

In this chapter we're going to discuss and go deeper in Mandob mobile application's testing, and present the types of testing to be used and test cases we examined our application through.



5.1 Functional Testing:

Unit testing

Testing of individual items (e.g. modules, programs, objects, classes, etc.) Usually as part of the coding phase, in isolation from other development item sand the system as a whole.

Integration testing

Testing the interfaces between major (e.g. systems level application modules) and minor (e.g. individual programs or components) items within an application which must interact with each other.

System testing

Testing a system behavior as a whole when development is finished and the system can be tested as complete entity.

Regression Testing

To check older functionality after integrating new functionality.

Acceptance testing

to ensure that a development is ready to be deployed into the business, operational or production environment.

5.2 Non-Functional Testing

Performance Testing

Accomplished a designated function regarding processing time and throughput rate.

Load Testing

Measuring the behavior of within creasing load which can be handled by the component or system.

Stress Testing

Evaluate a system or component at or beyond the limits of its specified requirements.

Security Testing

Testing how well the system protects against unauthorized internal or external access.

Test Cases

- **Test Scenario Objective:**
 - Verify signup successfully with valid Application Entries , Sign up for all possible users.

Assumptions / Dependencies :

Valid Name : Kareem Sameh

Kareem Ahmed

Kareem Mohamed

Kareem Youssef

Kareem moussa

Kareem ibrahim

Valid Username : Kareem – Kareem1 – Kareem2 – Kareem3 – Kareem4 –
Kareem5

Valid Password : 123456 (same for all users in Unit test)

Valid Email : kareem@gmail.com – kareem1@gmail.com –
kareem2@gmail.com – kareem3@gmail.com – kareem4@gmail.com –
kareem5@gmail.com

Valid Phone Number : 01065242355 – 01004252558 – 01000544655 –
01004877858 – 01116652558 - 01225588766

Valid Who are you radio : Broker – finishing work – hardware supplier – product
supplier – working hand – Regular user.

Description	Expected Result	Actual Result	Error Type
Navigate to sign up screen	App Navigate to signup screen		
Enter Name : kareem ahmed	User enter the required name		
Enter Username : kareem	User enter the required Username		
Enter Email : kareem@gmail.com	User enter the required Email		
Enter password :	User enter the		

123456	required password		
Enter phone number : 01065242355	User enter the required phone number		
Choose what is the category user working in : Broker	User enter the required category		
Press signup button	Sign up successfully and user navigate to broker Home Page	Passed	

Description	Expected Result	Actual Result	Error Type
Navigate to sign up screen	App Navigate to signup screen		
Enter Name : kareem mohamed	User enter the required name		
Enter Username : kareem1	User enter the required Username		
Enter Email : kareem1@gmail.com	User enter the required Email		
Enter password : 123456	User enter the required password		

Enter phone number : 01004252558	User enter the required phone number		
Choose what is the category user working in : Finishing worker	User enter the required category		
Press signup button	Sign up successfully and user navigate to Finishing Work Home Page	Passed	

Step Number	Expected Result	Actual Result	Error Type
Navigate to sign up screen	App Navigate to signup screen		
Enter Name : kareem yossef	User enter the required name		
Enter Username : kareem2	User enter the required Username		
Enter Email : kareem2@gmail.com	User enter the required Email		
Enter password : 123456	User enter the required password		
Enter phone number : 01000544655	User enter the required phone number		
Choose what is the	User enter the		

category user working in : Hardware supplier	required category		
Press signup button	Sign up successfully and user navigate to Hardware supplier Home Page	Passed	

Description	Expected Result	Actual Result	Error Type
Navigate to sign up screen	App Navigate to signup screen		
Enter Name : kareem moussa	User enter the required name		
Enter Username : kareem3	User enter the required Username		
Enter Email : kareem3@gmail.com	User enter the required Email		
Enter password : 123456	User enter the required password		
Enter phone number : 01004877858	User enter the required phone number		
Choose what is the category user working in : Product supplier	User enter the required category		

Press signup button	Sign up successfully and user navigate to Product supplier Home Page	Passed	
----------------------------	--	--------	--

Description	Expected Result	Actual Result	Error Type
Navigate to sign up screen	App Navigate to signup screen		
Enter Name : kareem ibrahim	User enter the required name		
Enter Username : kareem4	User enter the required Username		
Enter Email : kareem4@gmail.com	User enter the required Email		
Enter password : 123456	User enter the required password		
Enter phone number : 01116652558	User enter the required phone number		
Choose what is the category user working in : Working Hand	User enter the required category		
Press signup button	Sign up successfully and user navigate to Working hand	Passed	

	Home Page		
--	-----------	--	--

Description	Expected Result	Actual Result	Error Type
Navigate to sign up screen	App Navigate to signup screen		
Enter Name : kareem youssef	User enter the required name		
Enter Username : kareem5	User enter the required Username		
Enter Email : kareem5@gmail.com	User enter the required email address		
Enter password : 123456	User enter the required password		
Enter phone number : 01225588766	User enter the required phone number		
Choose what is the category user working in : Regular User	User enter the required category		
Press signup button	Sign up successfully and user navigate to Regular User Home Page	Passed	

- **Test Scenario Objective:**

- Verify Login successfully with valid Application Entries , Login for a User

Assumptions / Dependencies :

Valid Email : kareem[null – 5]@gmail.com

Valid Passwords : 123456

Login by facebook

Login by google

Description	Expected Result	Actual Result	Error Type
Navigate to Login screen	App Navigate to Login screen		
Enter Email : kareem@gmail.com	User enter the required name		
Enter password : 123456	User enter the required Username		
Press Login button	Login successful and application navigate to specific User's Page	Passed	

Login by Facebook

Description	Expected Result	Actual Result	Error Type
Navigate to Login screen	App Navigate to Login screen		
Press Facebook Icon	User can log in using his facebook email		
Choose your facebook profile	Login to required page and go to home page	Passed	

Login by Google

Description	Expected Result	Actual Result	Error Type
Navigate to Login screen	App Navigate to Login screen		
Press Gmail Icon	User can log in using his Gmail		
Choose your Gmail Icon	Login to required page and go to home page	Passed	

- **Test Scenario objective**
- Verify that user can upload his data successfully

Assumptions / Dependencies :

Valid Pictures : pictures from phone or drive or camera

Valid description : Text describing the place

Valid radio button choice : Rent or choice

Valid Location : a valid string with the main location of the place

Valid Geo-Location : Choose from the map the place of the place

Valid Size : number that describe the place dimensions in meters

Valid Rent price : number that describe how much would user pay each month

Valid Sell price : Number that describe how much user would pay to buy the place

== For Broker ==

Precondition : -- Look at System analysis section --

Description	Expected Result	Actual Result	Error Type
Navigate to Add New Place Icon	App navigates to the add place screen		
Enter a valid pictures	A suitable upload service is being chosen and		

	pictures uploaded		
Enter a valid description	A text is saved in the description box		
Enter a valid radio button choice Rent or sale	A radio button glow appear the required if place is for rent or sale		
Enter a location of the place	Location is saved for the place		
Enter a valid place for the place	Choose the location of the place from Google-maps		
Enter the size of the place	Enter a valid size of the place		
Enter the rent and sell Price	Enter the price of the place		
Press Confirm button	All the data should be saved to firebase for further operations	Passed	

== For Finishing Works ==

Precondition : -- Look at System analysis section –

Valid Pictures : pictures from phone or drive or camera

Valid description : Text describing the place

Valid Price/Meter size : a valid integer that describe the price / meter

Valid Radio button: Choose from lux – super Lux –ultra Lux

Description	Expected Result	Actual Result	Error Type
Navigate to Add New Finishing work Icon	App navigates to the add Finishing work screen		
Enter a valid pictures	A suitable upload service is being chosen and pictures uploaded		
Enter a valid description	A text is saved in the description box		
Enter a valid radio button choice to show finishing type	A radio button glow appear the required if finishing is lux, super lux, or ultra-lux		
Choose the category of the finishing work	Category is chosen		

Press Confirm button	All the data should be saved to firebase for further operations	Passed	
-----------------------------	---	---------------	--

Valid Pictures : pictures from phone or drive or camera

Valid Item name : enter the name of the hardware item

Valid description : Text describing the hardware type and details

Valid Unit Price : a valid integer that describe the Unit price

Valid Installation Price : a valid integer that describe the price of installing price

Valid Category: Choose what is the category of the item

== For Hardware suppliers ==

Precondition : -- Look at System analysis section --

Description	Expected Result	Actual Result	Error Type
Navigate to Add New Finishing work Icon	App navigates to the add Finishing work screen		
Enter a valid pictures	A suitable upload service is being chosen and pictures uploaded		

Enter a valid description	A text is saved in the description box		
Enter a valid radio button choice to show finishing type	A radio button glow appear the required if finishing is lux, super lux, or ultra-lux		
Choose the category of the finishing work	Category is chosen		
Press Confirm button	All the data should be saved to firebase for further operations	Passed	

== For Product suppliers ==

Precondition : -- Look at System analysis section –

Valid Pictures : pictures from phone or drive or camera

Valid Item name : Trousers

Valid description : Good Trousers for walking with variety of sizes

Valid Sell Price : 150

Valid Wholesome Price : 120

Valid Delivery Time: 3

Valid Category : Clothes

Description	Expected Result	Actual Result	Error Type
Navigate to Add New Product Icon	App navigates to the add Finishing work screen		
Enter a valid pictures	A suitable upload service is being chosen and pictures uploaded		
Enter a valid Item Name : Trouzers	Item name is saved in the item name box		
Enter a valid description : Good Trouzers for walking with variety of sizes	A text is saved in the description box		
Enter sell Price : 150	A sell price is saved		
Enter the wholesome price : 120	A wholesome price is saved		
Enter Valid Delivery Time : 3	3 days delivery time is saved in the box		

Enter the category : Clothes	Category is saved in the search list view		
Press Confirm button	All the data should be saved to firebase for further operations	Passed	

== For Working Hand ==

Precondition : -- Look at System analysis section –

Valid Picture : pictures from phone or drive or camera

Valid C.V : Upload a valid CV

Valid Category : Mobiles

Valid Expected Salary : 4500

Description	Expected Result	Actual Result	Error Type
Navigate to Add New Working Hand Icon	App navigates to the add Finishing work screen		
Enter a valid pictures	A suitable upload service is being chosen and pictures uploaded		

Enter a valid C.V.	Upload a c.v.		
Choose the category Work	Category is chosen		
Enter expected Salary: 4500	A required price is entered and saved in the box		
Press Confirm button	All the data should be saved to firebase for further operations	Passed	

Chapter 6

Results and Discussion

In this chapter we're going to find out the results of the project whether they're achieved or not and also the differences between the desired results and the actual ones

6.1 Results

- **6.1.1 Expected Results**

- The application help the e-commerce process by supporting small to medium size business investments
- Many users including many categories can sign up in Mandob and enjoy a limitless selling experience
- Regulars can easily sign up and choose a category to start a new business or manage the existing business
- Working hands can sign up in the application and can view job offers and accept what is suitable for them
- The purchasing process is fully automated and is being held by the users
- Sponsoring is the main revenue stream of the application

- **6.1.2 Actual Results**

- The application help the e-commerce process by supporting small to medium size business investments
- Many users including many categories can sign up in Mandob and enjoy a limitless selling experience
- Regulars can easily sign up and choose a category to start a new business or manage the existing business
- Working hands can sign up in the application and can view job offers and accept what is suitable for them
- The purchasing process is fully automated and is being held by the users
- Sponsoring was held to the late development process due to resources shortages

- **6.2 Discussion**

As the founders of Mandob we believe that E-Commerce is the future we had the vision with the help of our great supervisor who chose Mandob among many other applications. Mandob is believed to give a new Era of business marketing and investments it will change how the world look at investment. And in the future we believe that everyone can easily start his new business with great easiness and great opportunity to be on the top of business ladder.

Chapter 7

Conclusion

making such a platform will make a revolution in the business field. Allowing more investors to build their empires from scratch with efficiency in no time, And ease business management of existing ones will give them more time to invest more and more. And by canceling management difficulties, more investments will be noticed, decreasing unemployment and increasing market satisfaction.

Chapter 8

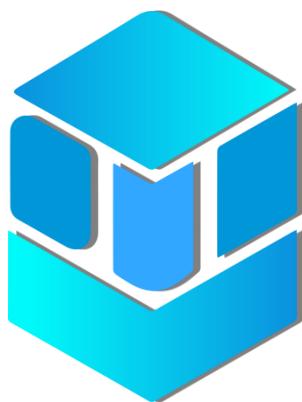
Future work

- Platform full connection with cashier system to view statistics, automatically order nearly finished products, and revoke unneeded products .
- Give suggestions to improve store by studying selling patterns, and customers preferences.
- Automatically generate Commodities combinations that will fit the business
- Budget based automated build
- Mandob Statistics will give briefs and details about the running business and consultancies of how it will be managed in the next few steps
- Cloud computing and big data managing will be a key future work and important for extending users database
- Most important thing is increasing the categorical services and products to be in everything soon

Mandob© | Business is never easier.



"Business opportunities are like buses, there's always another one coming."



Business is **MANDOB** never easier

THANK YOU