سمسارك
Semsark

كلية الحاسبات والذكاء الاصطناعي
جامعة القاهرة

سمسارك
Semsark

Dr.Dina ezzat

Eng.Dina amr

WITH YOU

SEMSARK

Documentation

Mohamed Saeed Ahmed	engmohamedsaeed19@gmail.com	IS
Islam Mohamed Abo Ouf	Eng.Islam.abououf@gmail.com	IS
Abdelrahman Ahmed Mohamed	abdelrahmn.ahmed119@gmail.com	IS
Youssef El Sayed Sultan	Youssefsoultan5@gmail.com	IS
Mohamed Abdelaziz Abdelmoien	Ma6707929@gmail.com	IS
Youssef Mounir Al Youssef	yousefalyousef321@gmail.com	IS

Under supervision

Dr. Dina Ezzat TA. Dina Amr

DECEMBER 12, 2022

Table of Contents

List of figures

Chapter 1

Introduction

1.1 Problem Statement:

There are several house rentals held by various proprietors, each with its own set of advantages and disadvantages. This process is difficult for them and for tenants specifically students because not all owners allow students to rent. As a renter, we'd want to give them the flexibility to choose their house by using Google Maps to specify the house location and by making the filtrations they want. So that we can make sure that they can decide on house renting.

1.2 Background:

As a student studying distant from home and without access to a hostel, we must find a nearby property to rent. House rentals near the university are frequently chosen by students. Students who wish to rent a property will have a difficult time finding one that is acceptable to them. The owner normally does not publicize their residence and instead spreads word of mouth among the residents. The student would have to ask around for available rental houses, which will take a long time. The goal of this system is to make it easier for students to obtain rental housing. This system will display various houses for rent and assist students in making their decision.

Another challenge with traditional rental processes and other applications is the limited access to information about properties. Often, the only way for potential renters to get a sense of a property is through in-person visits, which can be time-consuming and inefficient. This can lead to renters choosing properties that are not a good fit for their needs, leading to dissatisfaction and the need to move again in the future. By incorporating Google Maps into the rental application, renters can easily view the location of properties and see their surrounding areas. They can also see street-level imagery and use satellite view to get a better understanding of the property and its surroundings. This allows for more informed decisions and a more efficient rental process, reducing the number of unsuccessful rentals and increasing overall satisfaction with the process.

1.3 System Purpose:

The purpose of the system is to facilitate interaction between the user and the customer.

All manual papers are also converted into electronic transactions, in addition to the accuracy of the information that reaches the customer.

Where he can find out the address where he will live on the map, and all the details are explained in front of him, starting from the address on the map to the interior details of the apartment.

It also provides housing for all groups, where the user selects the type of category (family, student, individual worker, ... etc.), and based on his desire, a list of houses will be found based on his choice.

The main goal of this project is to facilitate the overall renting process for owners and renters.

1.4 System Scope:

In scope:

- The customer can prove his identity exclusively through the ID card.
- The customer can see ads outside his area.
- Show the apartment details to the customer in a simple way on the map.
- The customer's ability to modify the apartment's data.
- The customer's ability to update his personal data.
- The customer's ability to renew the ad.
- The customer's ability to complete the contract work in a simple manner.
- The client's ability to filter on the map for specific areas.
- Ensure the quality and credibility of the data and images used in the advertisement.

Out of scope:

- The customer's reservation is to view the details of the apartments in his area only.
- The client's inability to put pictures from his gallery (exclusively they must be taken from the camera to ensure that realistic pictures of the apartment are placed)

1.5 System objectives and acceptance criteria

1.5.1 System Objectives

□ We will provide a better experience for users.

□ We will try to make our application contain 1000 active users at the beginning without breaking down.

□ we will Provide a cross-platform application.

1.5.2 Success Criteria for the Project

• If our application is downloaded by 1000 users during the first 6 months and 80% of users use it, then our app is successful, and our app achieves its goal.

1.6 Life cycle model

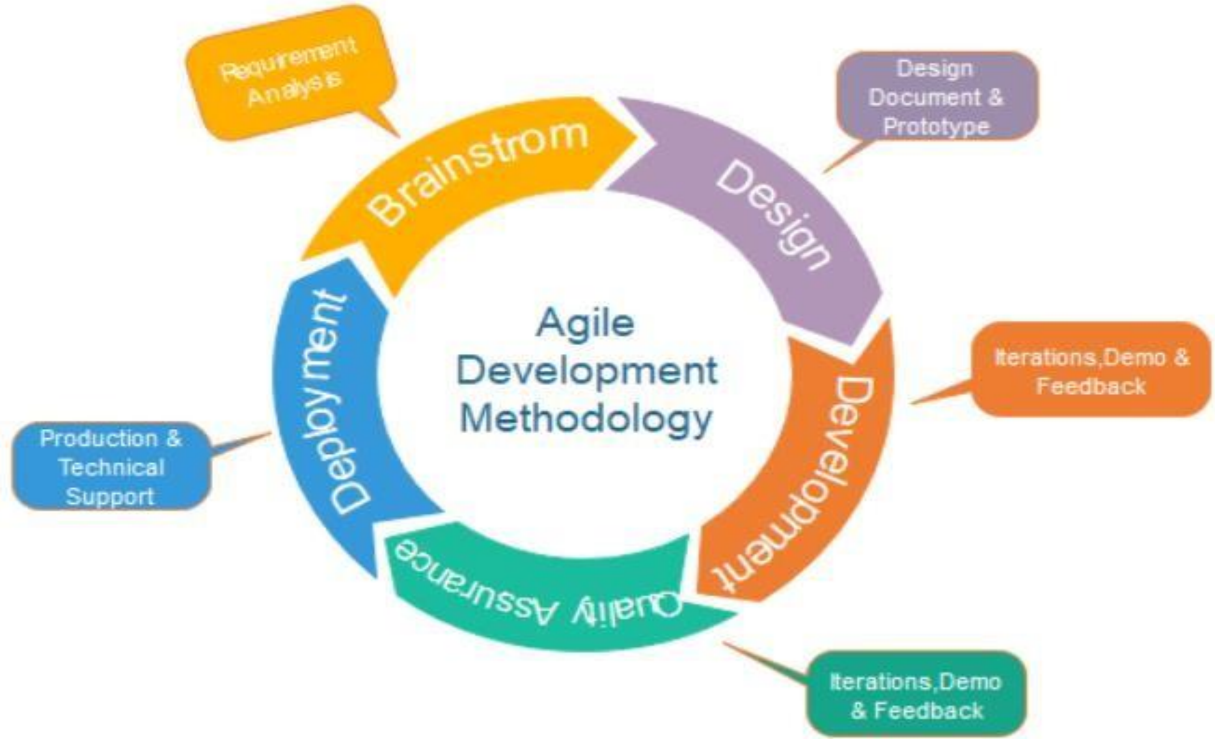

Requirement Analysis
Brainstrom
Design Document & Prototype
Design
Agile Development Methodology
Iterations,Demo & Feedback
Development
Production & Technical Support
Deployment
Quality Assurance
Iterations,Demo & Feedback

Figure 1

The Agile SDLC development method focuses on collaborative decision-making, customer satisfaction, and development over multiple short cycles or sprints, rather than a top-down process with a single series of stages. Your teams work in multiple cycles,which typically last between two and four weeks.

Developers use an iterative approach to focus on the most important features at any given moment, rather than going “according to plan.

1.7 Why Agile Methodology

Main reasons for choosing Agile methodology:

- There is a flexibility in the project as requirements are likely to change because during the development process, we discover new valuable features, so we decide to add them to the next sprint.
- It provides fast and visible results thanks to Sprint-based development life cycle.
- Immediate feedback as we make a several meetings with different people who have different point of view to get feedback from them after each sprint.

1.8 Solution statement

1.8.1 Project Overview

Business Case

- **What is the project?**

  The project is an application designed in an easy way to place advertisements in the field of architectural real estate, and the user can see the advertisements of other users in a new way, which is seeing the advertisements on the map directly and talking with the owners of the advertisements through the application, an easy window for students and non-students to know all the details of the advertisement.

- **The Value of project**
  1. This project will help student and people to find apartment – villa - … etc. easily.
  2. Ending the exploitation and contempt of brokers
  3. We guarantee high user protection.
  4. Certification at the location of the apartment

1.8.2 Limitation

1. Application used only in Egypt.
2. The user can add advertisements only when he documents himself by photographing his national ID or passport information page and photographing himself.
3. Account is unique on email; Email can be used to make 1 account only.

1.9 Expected results.

we expect from this project help the person easily find a suitable house rental using this system. By using Google Maps, this system can help the user to make a perfect choice of the house rental.

1.10 Related Work

- **OLX provide a similar service to SEMSARK.**

  But the difference in our service is that the user can get better services, as the location that appears in front of the user will be the real location of the apartment, and then he can know if the apartment is suitable and close to the services he needs.

  Also, dealing with SEMSARK is limited to a simple transaction between. your direct seller and direct beneficiary.

  This means that the system has no Broker, and this is what either party wish to reduce the cost.

- **Related work Like Aqar map**

  The application supports the provision of apartments available for rent and sale, but the difference between Aqar map and Semsark clarifies the real estate on the map so that the customer can view the coordinates and distinctive signs of the apartment and know all the details of the apartment.

  While the client finds the apartment he is looking for, he may face a problem with its location, as it is far from his interests, which makes him reject it.

  But in Semsark, the client can put his interests in the areas he wants and can see them on the map, which facilitates the process of renting and selling.

Chapter 2
PLANNING

2.1 Planning Basis

2.1.1 Scope

Our scope covers Buying, Renting and Selling properties without any intermediary between seller and buyer.

2.1.2 Milestones

Milestone	Description
Project approval	Approve our project idea by Dr and TA
Planning	Understand how we will reach to our goal? and show all critical details
Design approval	Approve our design by our Dr and TA
Review Requirements	Review All project requirements
Complete Tasks	Finish all assigned tasks
Test Functionalities	Test our functionalities if they achieve their purpose
Fix Bugs	Fix any bug will we meet in test phase
Deploy	Deploy the project on our server

2.1.3 Phases

Phase	Description	Sequence
Project Initiation	Defining the project by developing a business case, feasibility study and Project Charter as well as recruiting the project team	Phase # 1
Project Planning	Review All requirements, assign tasks to team members and specify due time	Phase # 2
Project Execution	Execute assigned tasks, test them, and migrate them together	Phase # 3
Project Closure.	Deploy final result on our server	Phase # 4

2.1.4 Tasks

Phase	Activity	Task	Sequence
Project Planning	Develop Quality Plan	Identify Quality Targets Identify Quality Assurance Techniques Identify Quality Control Techniques	1st 2nd 3rd
Project Execution	Develop Technical Plan	Create Authentication system Create CRUD operation for Ad Create View All buildings	1st 2nd 3rd
Project Execution	Test Technical Tasks	Test Authentication system Test CRUD operation for Ad Test View All buildings	1st 2nd 3rd
Project Closure	Migrate And Deploy	Fix Bugs Migrate Tasks Deploy	1st 2nd 3rd

2.1.5 Effort

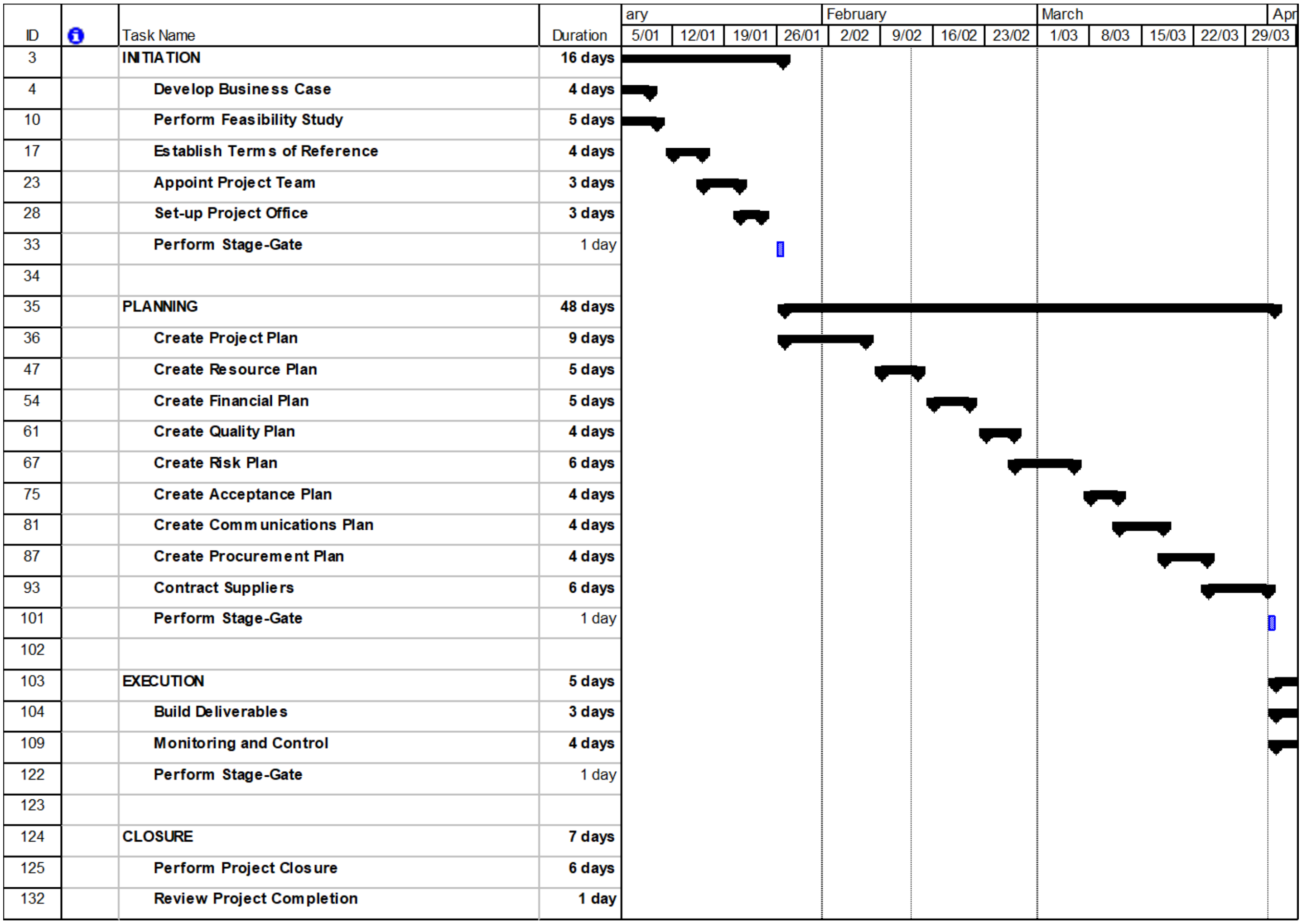
Task	Effort
Identify Quality Targets **Identify Quality Assurance Techniques** **Identify Quality Control Techniques**	*One. day* *One. day* *One. day*
Create Authentication system **Create CRUD operation for Ad** **Create View All buildings**	*Five. days* *Three. days* *no. days*
Test Authentication system **Test CRUD operation for Ad** **Test View All buildings**	*Two. days* *One. day* *One. day*
Fix Bugs **Migrate Tasks** **Deploy**	*Three. days* *Two. days* *One. day*

2.1.6 Resources

Task	Resource
Identify Quality Targets **Identify Quality Assurance Techniques** **Identify Quality Control Techniques**	*People as Project manager* *People as Project manager* *People as Project manager*
Create Authentication system **Create CRUD operation for Ad** **Create View All buildings**	*People as Backend developer* *People as Backend developer* *People as Backend developer*
Test Authentication system **Test CRUD operation for Ad** **Test View All buildings**	*People as Tester* *People as Tester* *People as Tester*
Fix Bugs **Migrate Tasks** **Deploy**	*People as developer* *People as Leader* *Money for Server*

2.2 Project Plan

2.2.1 Schedule



ID	ℹ	Task Name	Duration
3		INITIATION	16 days
4		Develop Business Case	4 days
10		Perform Feasibility Study	5 days
17		Establish Terms of Reference	4 days
23		Appoint Project Team	3 days
28		Set-up Project Office	3 days
33		Perform Stage-Gate	1 day
34			
35		PLANNING	48 days
36		Create Project Plan	9 days
47		Create Resource Plan	5 days
54		Create Financial Plan	5 days
61		Create Quality Plan	4 days
67		Create Risk Plan	6 days
75		Create Acceptance Plan	4 days
81		Create Communications Plan	4 days
87		Create Procurement Plan	4 days
93		Contract Suppliers	6 days
101		Perform Stage-Gate	1 day
102			
103		EXECUTION	5 days
104		Build Deliverables	3 days
109		Monitoring and Control	4 days
122		Perform Stage-Gate	1 day
123			
124		CLOSURE	7 days
125		Perform Project Closure	6 days
132		Review Project Completion	1 day

Figure 2

2.2.2 Dependencies

Activity	Depends on	Dependency Type
Develop Quality Plan	Business Plan	Finish-to-start
Develop Technical Plan	Develop Quality Plan	Finish-to-start
Test Technical Tasks	Develop Technical Plan	Finish-to-start
Migrate And deploy	Test Technical Tasks	Finish-to-start

2.2.3 Assumptions

It is assumed that:

- Correct number of handheld devices arrive on target delivery date with no delays.
- The application will utilize REST API architecture.
- Project will follow Agile methodology throughout development.
- The project scope will not change once the stakeholders sign off on the scope statement.

2.2.4 Constraints

- Our application must be compatible with IOS starting from IOS-11 and Android starting from Android – 6.0.
- Get permissions for camera and location.
- The apartment images must be taken in the exact location of the apartment.
- Internet must be stable.
- National ID is a must.

2.2.5 Risk managements

Risk management method which we use is **“Loss Prevention and Reduction.”** This method of risk management attempts to minimize the loss, rather thaneliminate it. While accepting the risk, it stays focused on keeping the loss contained and preventing it from spreading.

ID	Risk	Impact	Risk Level	Risk Owner
1	Design delay	Schedule	High	Designer
2	Bugs occur	Quality Functionality	High	Tester
3	Response Late	Quality	Medium	Developer
4	Project delay	Schedule	High	Project Manager

Chapter 3
ANALYSIS

3.1 Stakeholders Analysis

Primary stakeholders:

- **Property owners/landlords**: They want a platform to advertise their rental properties and reach potential tenants.
- **Renters:** They want an easy and convenient way to search for rental properties and make informed decisions

3.2 Proposed Solution

- Integrate Google Maps into the rental application to provide detailed and accurate information about the location of rental properties, including transportation, amenities, and safety features.
- Offer a user-friendly interface for renters to search for rental properties based on their preferences and budget.
- Allow property owners/landlords and real estate agents to easily advertise their rental properties on the platform, with the option to add photos, descriptions, and contact information.
- Implement verification procedures to ensure that all rental properties comply with local regulations and standards.
- Offer features to promote equitable access to rental properties, such as filters for low-income housing, accessible units, and more.

- Regularly gather feedback from renters, property owners/landlords, real estate agents, and other stakeholders to continually improve the platform.

  This proposed solution aligns with the needs and interests of all stakeholders and leverages the capabilities of Google Maps to provide accurate and reliable information about rental properties. It also considers local regulations and promotes equitable access to rental properties.

3.2.1 Functional Requirements:

1. **Sign-Up**: Just entering the sign-up page, the user can create an account through social methods or through the traditional method by entering basic data and personal information ... etc.
2. **Sign-In (Login):** Just entering the sign-in page, the user can login by his account through social methods or through the traditional method by entering email and password.
3. **Forget Password**: The user can recover his account, even if he forgot the password, by entering the Forget Password page and entering his email, and OTP will be sent to his email, and then he will be transferred to another page to create a new password.
4. **Create Advertisement:** The user can place an advertisement on the application by filling out a form with the data of the apartment, villa, whatever it is.
5. **See advertisements on the map:** The user, after logging in to the application, can see all the ads near his location on the map.
6. **Chatting:** The user can communicate with the owner of the advertisement through the application. The application provides an encrypted chat with two users.
7. **Taking a photo for apartment:** The user can take a photo by using his mobile camera only.
8. **Verify by using National Id:** The user can verify himself by uploading 2 pictures for his national id and a photo for himself and the application will detect the similarity between two photos.

3.2.2 Non-Functional Requirements:

3.2.2.1 Usability: we provide the following to enhance the Usability:

1- You can find all the information you are looking for

2- The user will find no difficulties in our application.

3- user-friendly application.

3.2.2.1 Reliability: The Application is available all the time

3.2.2.2 Performance: The landing page supporting 5,000 users per hour must provide seconds or less response time in an application, including the rendering of text and Map with Ads, over an LTE connection

3.3 System models

3.3.1 Scenario

1- Sign up scenario.

User story: **As a user, I want to sign up, so that I will be able to access all feature in this app**

Scenario	Sign up
Given	The user will be navigated to login page
When	At the first time when use our app or logout
And	Enter a valid email to receive an OTP
Then	System sends an OTP to the entered email
Given	The user will receive an OTP
When	System navigates him to OTP page
Then	The user enters received OTP
Given	System navigates him to final page to set his information
When	System checks on OTP if it is valid
Then	The user sets his information and click “signup” button
Given	System navigates him to home page

2- Forgot password scenario.

User story: **As a user, I want to recover my password, so that I will be able to access my account in case I forgot the password**

Scenario	Forgot password
Given	The user will be navigated to forgot password page
When	User clicks on “forgot password” option
And	Enter a valid email to receive an OTP
Then	System sends an OTP to the entered email
Given	The user will receive an OTP
When	System navigates him to OTP page
Then	The user enters received OTP
Given	System navigates him to final page to set a new password
When	System checks on OTP if it is valid
Then	The user sets his password and click “confirm” button
Given	System navigates him to login page

3- Create an advertisement scenario.

User story: **As a user, I want to create an advertisement, so that I will sell or rent my property**

Scenario	create an advertisement
Given	The user will be navigated to Add page
When	The user clicks on “add” button
And	Enter valid data of his property as (Area, price, number of rooms)
Then	System will get his Longitude and latitude after his permission to get exact location of user’s property
Given	User will receive “Your property is added successfully”
When	User clicks “finish” button

4- See apartments on map scenario.

User story: **As a user, I want to see apartments on the map, so that I will open the map.**

Scenario	See apartments on map
Given	The user will be navigated to home page
When	The user open the app and he logged in
And	Choose map view (map's icon)
Then	All apartments will be shown on the map

5- Chat scenario.

User story: **As a user, I want to chat with the property's owner, so that I will click on chat's icon which exists on advertisement.**

Scenario	Start a chat with the property's owner
Given	The user will be navigated to chat page
When	The user opens an advertisement
And	Clicks on chat's icon
Then	The chat will be start

6- Taking a photo for my apartment scenario

User story: **As a user, I want to take a photo for my advertisement, so that I will click on photo's button**

Scenario	Taking a photo for my apartment
Given	System opens user's camera after his permission
When	User clicks on "Choose an image button" on add ads page
And	User takes an apartment's photo
Then	Completes his property's information

7- Verify Account Scenario

User story: **As a user, I want to verify my account, so that I will be a subject of trust from other users**

Scenario	Verify my account
Given	System opens user's camera after his permission
When	User clicks on "Verify my account" on profile page
And	User takes a selfie photo with his National ID
Then	System checks if the two photos are matched to the same person using AI
Given	The user's account will verified

3.3.2 Use case models

3.3.2.1 use case actors

1- Admin

- Manage and monitor the whole system.

2- User

- Register and login to the system.
- Create and update house description.
- Search for the house with filtrations
- Book a house.
- Chat with house owner

3.3.2.2 use case narratives

Use case	Verify a user
Primary actor:	Customer
Goal in context:	Customer can verify his / her account.
Precondition:	Customer must have an Egyptian ID
Trigger:	When Customer need to verify his / her account.
Scenario:	1. System opens user’s camera after his permission. 2. Customer clicks on “Verify my account” on profile page. 3. Customer takes a photo for his National ID 4. System checks if the photo is matched with The photo, he uploaded of himself to the same person using AI 5. Then The Customer ’s account will verified
Exception:	The AI decides that they are not matched
Priority:	High
Frequency of use:	for one time
Channel to actor:	Application

Use case	Forgot password
Primary actor:	Customer
Goal in context:	Customer recover his / her password
Precondition:	Customer’s email must be valid
Trigger:	When Customer need to recover his / her password
Scenario:	1. The user will be navigated to forgot password page. 2. Customer clicks on “forgot password” option. 3. Enter a valid email to receive an OTP System sends an OTP to the entered email. 4. The Customer will receive an OTP. 5. System navigates him to OTP page. 6. The Customer enters received OTP. 7. System navigates him to final page to set a new password. 8. System checks on OTP if it is valid. 9. The Customer sets his password and click “confirm” button. 10. The user sets his password and click “confirm” button
Exception:	If the customer entered a different OTP number.
Priority:	High
Frequency of use:	If necessary
Channel to actor:	Application

Use case	Sign up
Primary actor:	Customer
Goal in context:	Customer wants to create an account.
Precondition:	Customer’s email must be first use.
Trigger:	When Customer need to create an account.
Scenario:	1- The user will be navigated to Sign up page. 2- Enter a valid email. 3- The Customer will receive an OTP. 4- System navigates him to OTP page. 5- The Customer enters received OTP. 6- System checks on OTP if it is valid. 7- System navigates him to form to enter his personal information. 8- The Customer sets his personal information and his password then click “confirm” button.
Exception:	If the customer entered a different OTP number or if the customer's email is already registered.
Priority:	High
Frequency of use:	For one time
Channel to actor:	Application

Use case	Create new advertisement
Primary actor:	Customer
Goal in context:	Customer want to add his / her advertisement
Precondition:	Customer's account must be verified
Trigger:	When Customer need to add his / her advertisement
Scenario:	1. The user will be navigated to Add page. 2. The user clicks on "add" button. 3. Enter valid data of his property as (Area, price, number of rooms) 4. System will get his Longitude and latitude after his permission to get exact location of user's property. 5. User will receive "Your property is added successfully." 6. User clicks "finish" button
Exception:	the customer loses internet connection
Priority:	High
Frequency of use:	Every time Customer add an advertisement
Channel to actor:	Application

Use case	See advertisements on the map
Primary actor:	Customer
Goal in context:	Customer can see advertisements on map
Precondition:	Customer must allow to application to take his location
Trigger:	When Customer need to see advertisement on map around his location
Scenario:	1- After user signed in, the user will be navigated to Home Page 2- Application will ask user to use his location. 3- Application will return the nearest advertisements around his location
Exception:	The user doesn't allow application to use his location.
Priority:	High
Frequency of use:	for All time
Channel to actor:	Application

Use case	Taking a photo for apartment
Primary actor:	Customer
Goal in context:	Customer can take photo to his advertisement.
Precondition:	Customer must allow to application to access his camera
Trigger:	When Customer need to create his advertisement
Scenario:	1- After user signed in, the user will be navigated to Create advertisement Page. 2- The user will press on button "take photo." 3- The user uses his camera to take photo and click save.
Exception:	The user doesn't allow application to use his camera.
Priority:	High
Frequency of use:	When the user needs to create advertisement.
Channel to actor:	Application

Use case	Chatting
Primary actor:	Customer
Goal in context:	Customer can take photo to his advertisement.
Precondition:	Customer must allow to application to access his camera
Trigger:	When Customer need to create his advertisement
Scenario:	1- After user signed in, the user will be navigated to home page. 2- The user will choose an advertisement. 3- The user clicks on chat icon. 4- The user can start chat with the advertisement's owner.
Exception:	The user was blocked by advertisement's owner.
Priority:	High
Frequency of use:	When the user needs to chat with advertisement's owner.
Channel to actor:	Application

Use case	Admins verify Customer account manually
Primary actor:	admin
Goal in context:	Admin Verifies manual the account comparison between the two photos if the AI rejects
Precondition:	AI decides that photos are not matched
Trigger:	When Admin need to Verify Customer account manually
Scenario:	1. Get all users who are not verified. 2. each user contains two images (personal and national id) 3. open and compare between them
Exception:	if one or both of user's images do not exist, Admin can't verify the user, the photos are not matched actually
Priority:	High
Frequency of use:	Every time AI decides that photos are not matched
Channel to actor:	Application

3.3.2.3 use case diagrams

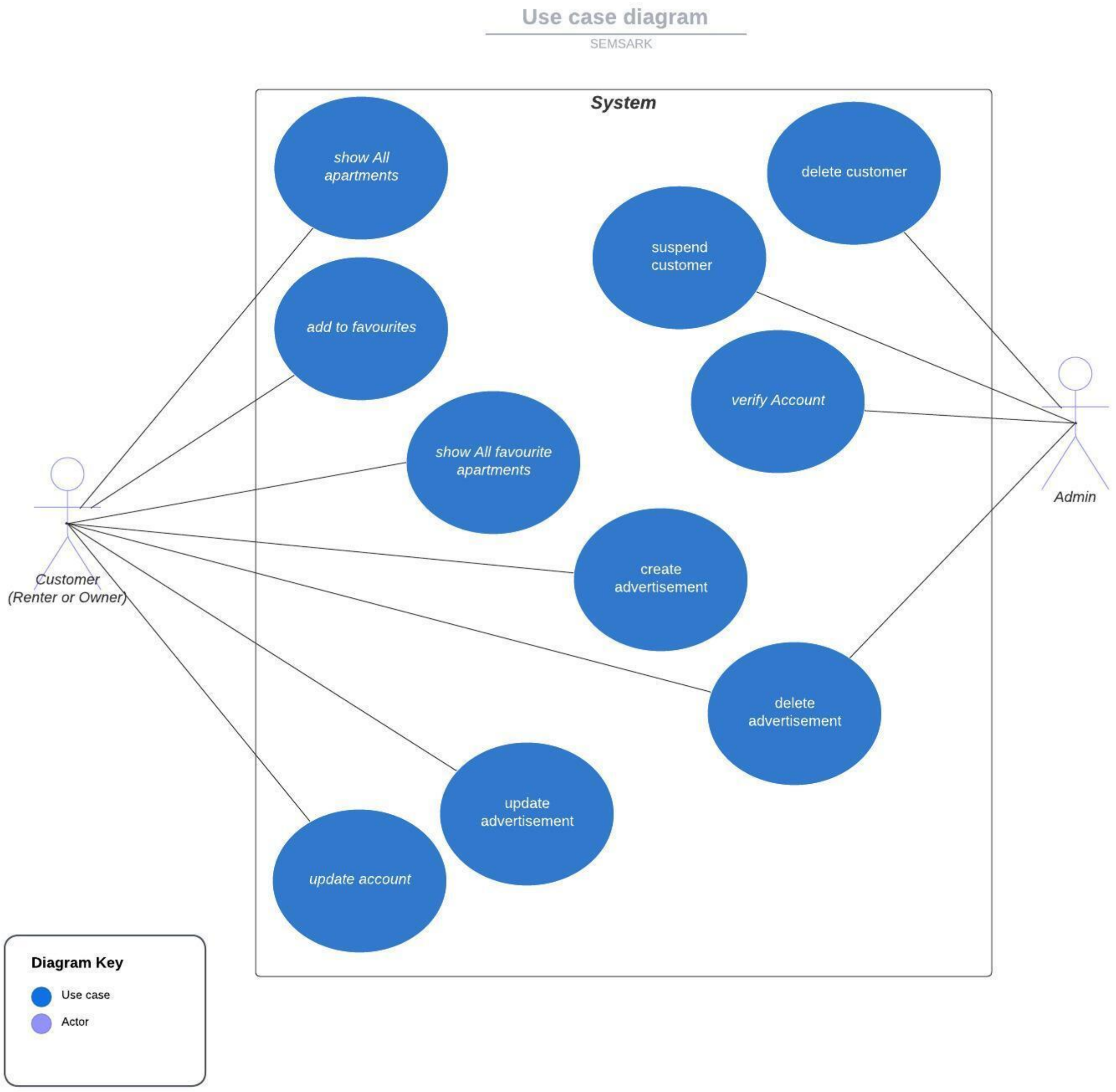


Figure 3

3.4 Storyboard

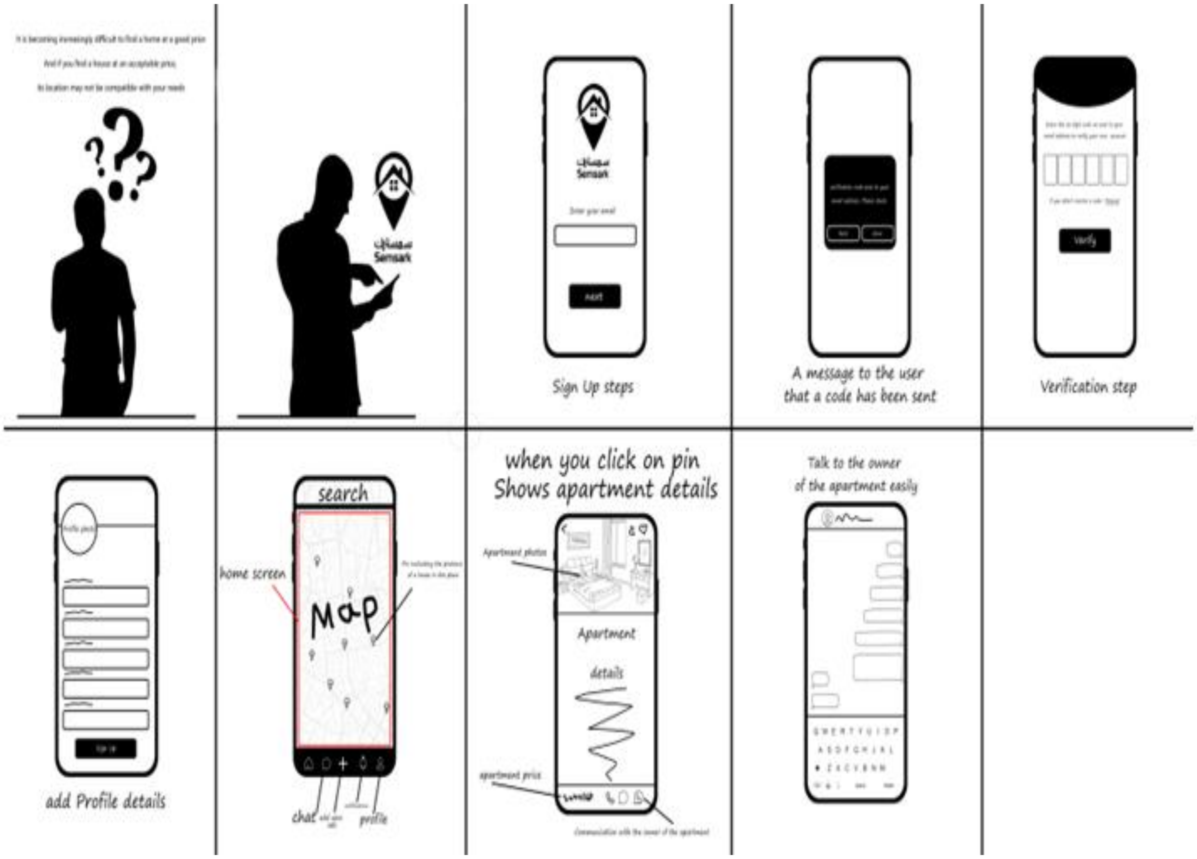

It is becoming increasingly difficult to find a home at a good price
And if you find a house at an acceptable price,
Its location may not be compatible with your needs
سمسارك
Semsark
Sign Up steps
A message to the user that a code has been sent
Verify
Verification step
add Profile details
search
home screen
Map
chat
profile
when you click on pin Shows apartment details
Apartment photos
Apartment
details
apartment price
Communication with the owner of the apartment
Talk to the owner of the apartment easily

Figure 4

Chapter 4
DESIGN

4.1 System decomposition

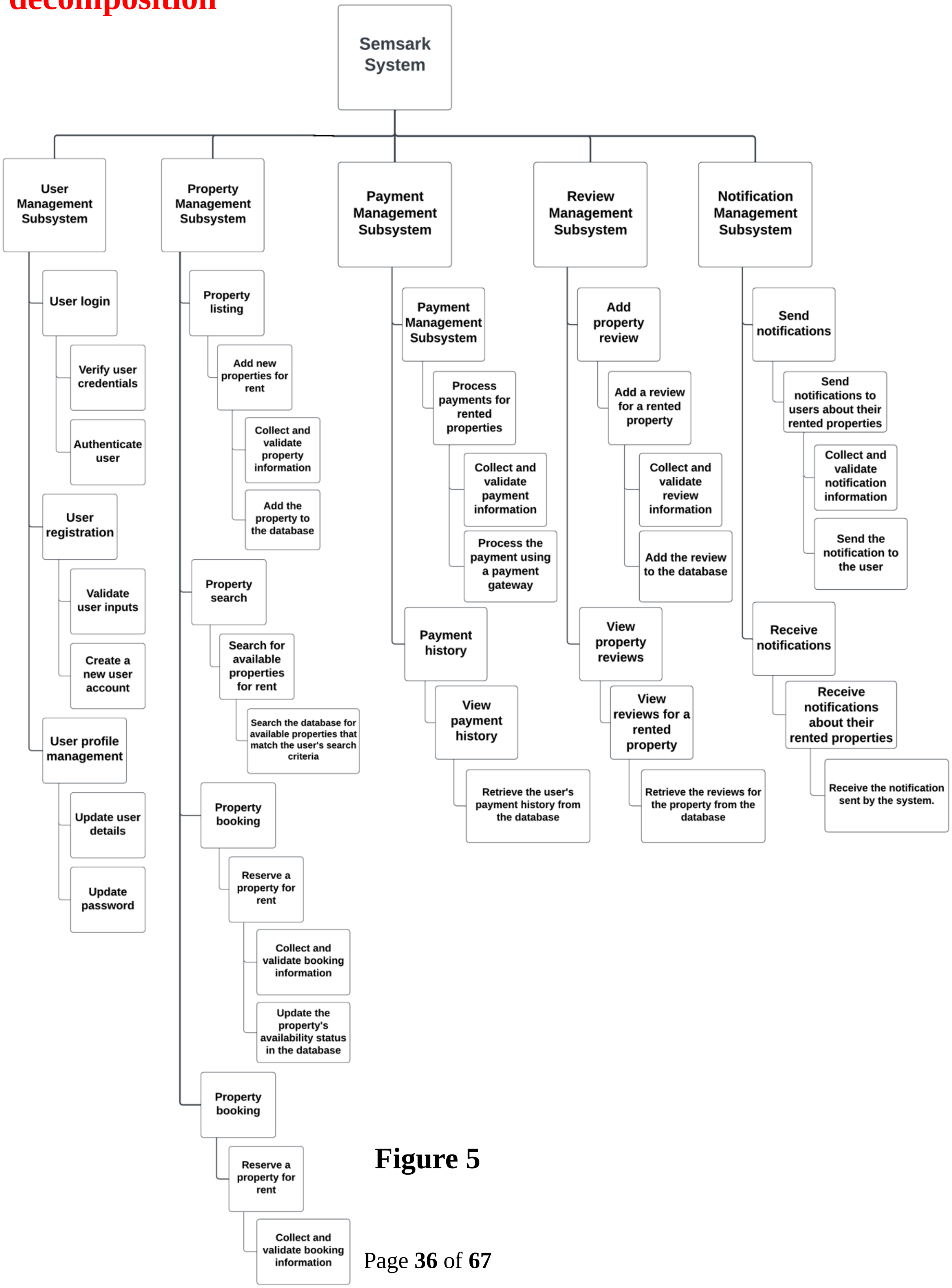

Semsark System
User Management Subsystem
User login
Verify user credentials
Authenticate user
User registration
Validate user inputs
Create a new user account
User profile management
Update user details
Update password
Property Management Subsystem
Property listing
Add new properties for rent
Collect and validate property information
Add the property to the database
Property search
Search for available properties for rent
Search the database for available properties that match the user's search criteria
Property booking
Reserve a property for rent
Collect and validate booking information
Update the property's availability status in the database
Property booking
Reserve a property for rent
Collect and validate booking information
Payment Management Subsystem
Payment Management Subsystem
Process payments for rented properties
Collect and validate payment information
Process the payment using a payment gateway
Payment history
View payment history
Retrieve the user's payment history from the database
Review Management Subsystem
Add property review
Add a review for a rented property
Collect and validate review information
Add the review to the database
View property reviews
View reviews for a rented property
Retrieve the reviews for the property from the database
Notification Management Subsystem
Send notifications
Send notifications to users about their rented properties
Collect and validate notification information
Send the notification to the user
Receive notifications
Receive notifications about their rented properties
Receive the notification sent by the system.

Figure 5

4.2 Object model

4.2.1 Class diagram

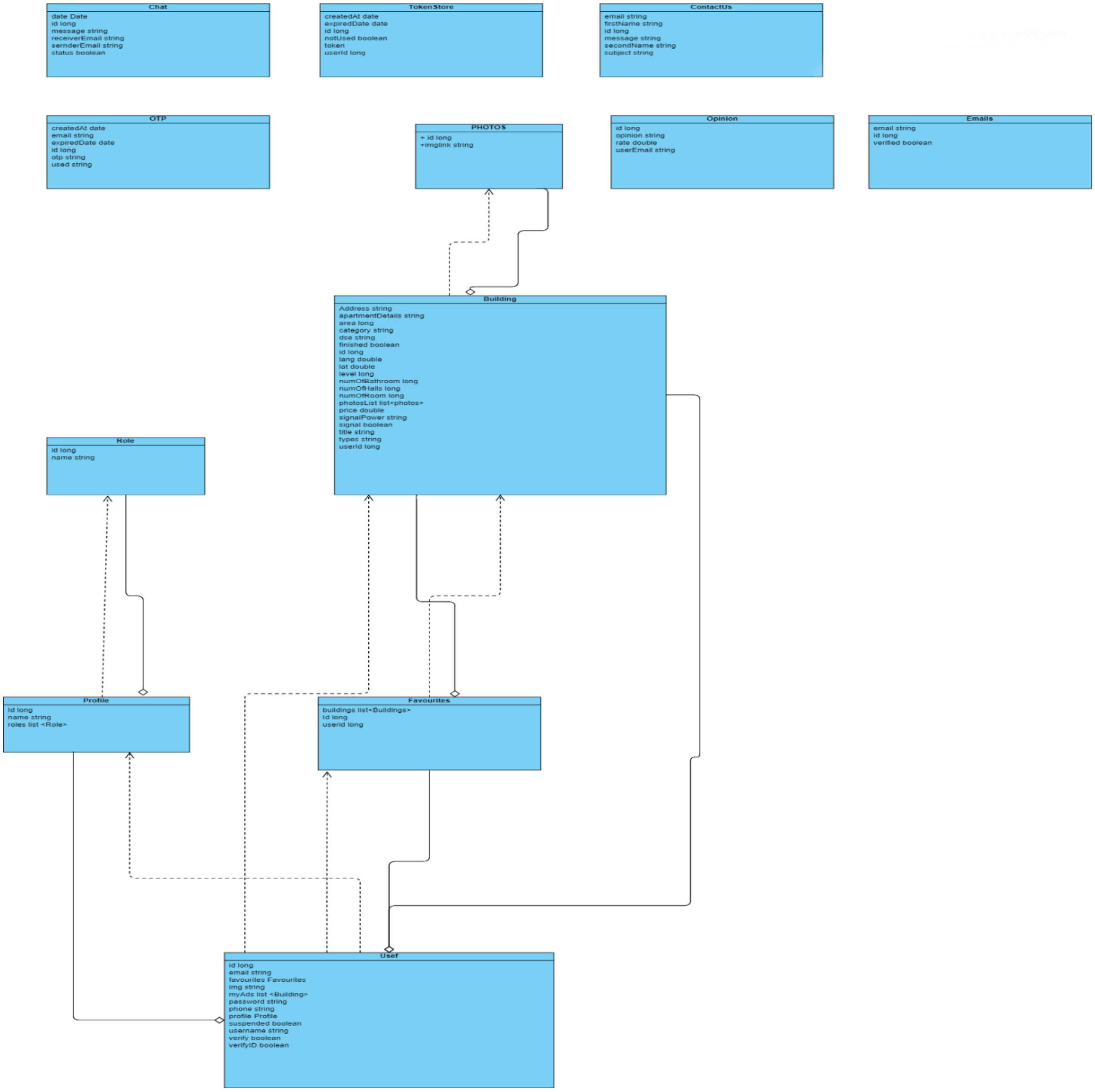


Figure 6

4.2.2 Class diagram data dictionary

- **User dictionary**

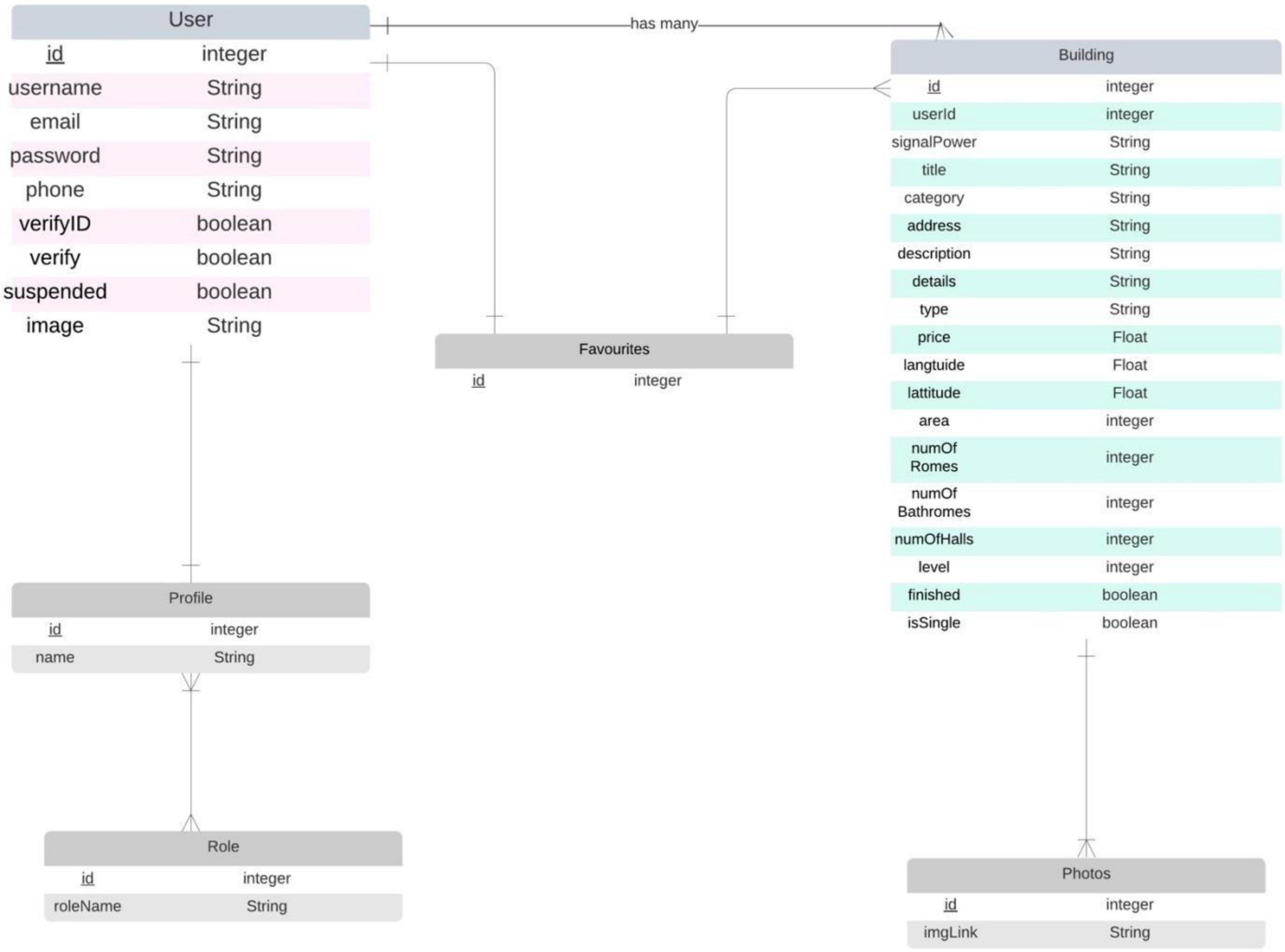
	Element or value display name	Description	Data type	Acceptable values	Required?	Accepts null value?
User	id	id that identifies the user	Integer	numbers	yes	no
User	userName	the name that is displayed at the profile	string	strings	yes	no
User	email	the email used to verify the account	string	strings	yes	no
User	phone	personal phone number of the user	Integer	numbers	yes	no
User	password	the password Which user use to login	Integer	numbers	yes	no
User	verifyID	Is used to check if the user verified his national id	bool	True or False	yes	no
User	verify	is used to check if the user entered the OTP properly	bool	True or False	yes	no
User	suspended	Is used to check if the user is suspended	bool	True or False	yes	no
User	image	the link of the image of the user that is stored in the server	string	strings	no	yes

- **Building dictionary**

	Element or value display name	Description	Data type	Acceptable values	Required?	Accepts null value?
Building	id	id that identifies the building	Integer	numbers	yes	no
Building	userID	id of the user who owns the building	Integer	numbers	yes	no
Building	signalPower	the phone signal that is available in the building	string	strings	yes	no
Building	title	the title of the building	string	strings	yes	no
Building	category	determines if the building is for renting or selling	string	strings	yes	no
Building	address	the address of the building	string	strings	yes	no
Building	description	mentioned the specification of the building	string	strings	yes	no
Building	details	the details of the building	string	strings	yes	no
Building	type	the type of the building	string	strings	yes	no

Building	price	the price of the building	Integer	numbers	yes	no
Building	longitude	the longitude of the building location in the maps	Float	numbers	yes	no
Building	latitude	the latitude of the building location in the maps	Float	numbers	yes	no
Building	area	area number where the building is located	Integer	numbers	yes	no
Building	numOfRooms	number of rooms in the building	Integer	numbers	yes	no
Building	numOfBathrooms	number of the bathrooms in the building	Integer	numbers	yes	no
Building	numOfHalls	number of the halls in the building	Integer	numbers	yes	no
Building	level	the level number of the building	Integer	numbers	yes	no
Building	finished	is used to check if the building is ready to live on it	bool	True or False	yes	no
Building	isSingle	is used to check if singles are eligible to live in the building	bool	True or False	yes	no

4.3 Entity relationship diagram



User	
id	integer
username	String
email	String
password	String
phone	String
verifyID	boolean
verify	boolean
suspended	boolean
image	String

Favourites	
id	integer

Building	
id	integer
userId	integer
signalPower	String
title	String
category	String
address	String
description	String
details	String
type	String
price	Float
langtuide	Float
lattitude	Float
area	integer
numOf Romes	integer
numOf Bathromes	integer
numOfHalls	integer
level	integer
finished	boolean
isSingle	boolean

Profile	
id	integer
name	String

Role	
id	integer
roleName	String

Photos	
id	integer
imgLink	String

Figure 7

4.4 User interface design

4.4.1 Prototype screenshots

4.4.2 Navigation hierarchy

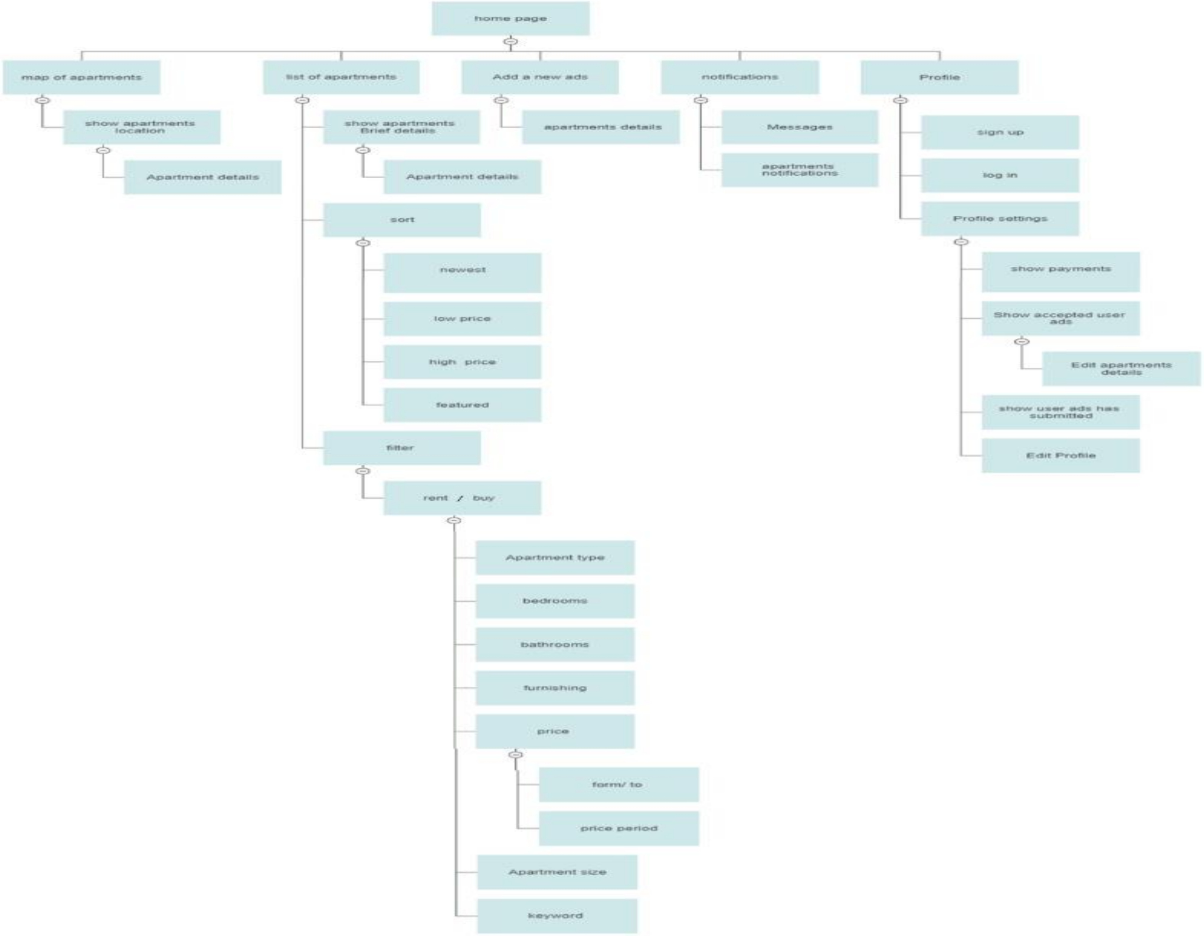


Figure 8

Chapter 5

IMPLEMENTATION

5.1 Tools used.

- **Back-end:**
  Framework: Java spring boot (v 2.7.5)
  Development tool: IntelliJ (v 2023.1)
  API: RESTful API and Google API

- **AI:**
  Libraries: OpenCV (v 4.8.0), Face Recognition (v 1.4.0) and Annoy (v 1.19.0)
  Development tools: Jupyter Notebook (v 6.5.4) and PyCharm (v 2023.1)

- **Database:** MySQL using Xampp (v 3.3.0)

- **UI and Design:** Adobe XD (v 56.1.12) and Lucid

- **Front-end:**
  Framework: React js (v 18.2.0)
  Development tool: PyCharm (v 2023.1)

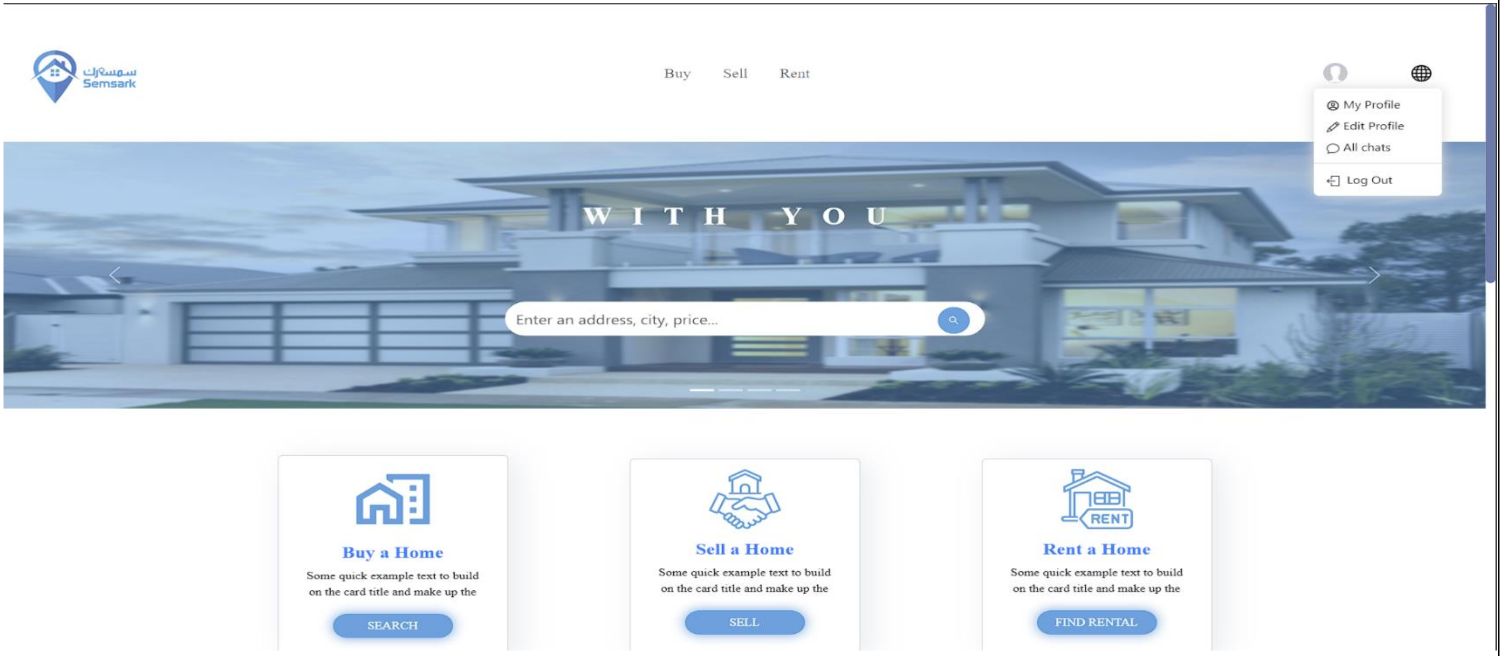
- **Mobile:**
  Framework: Flutter (v 3.7.12)
  Development tool: Android studio (v Flamingo | 2022.2.1 Patch 2)

- **Deployment**: Firebase (v 9.15.0)

5.2 GUI specification.

Website:

- **Navbar:** clickable items
  SEMSARK logo=> Home page
  Sell => App store to download SEMSARK app
  Rent => page contains on all apartments for rent
  Buy => page contains on all apartments for buy
  Profile menu contains on:
    my profile => profile page if logged in / login page if logged out
    Logout button => if the user is logged in)
  Language icon=> change the language (ar/en)



Buy a Home

Some quick example text to build on the card title and make up the

Sell a Home

Some quick example text to build on the card title and make up the

Rent a Home

Some quick example text to build on the card title and make up the

- **Home page:**

  Search input

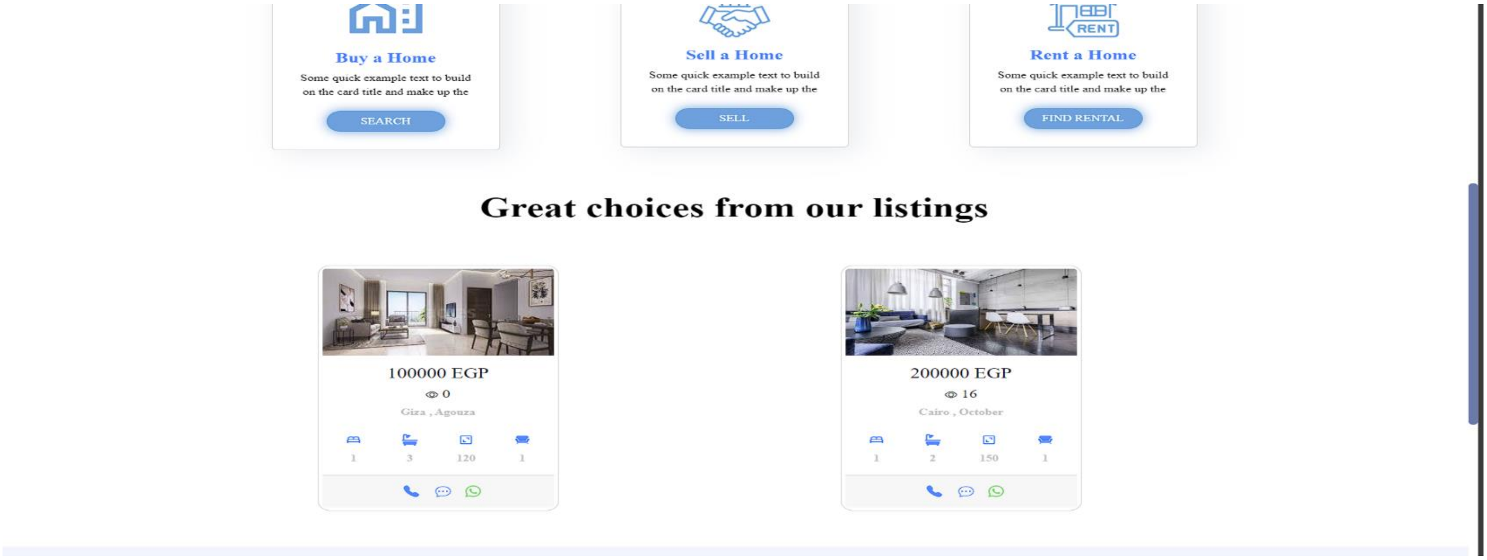
  Three cards:

  Sell => App store to download SEMSARK app

  Rent => page contains on all apartments for rent

  Buy => page contains on all apartments for buy

  List of Recommended Apartments: clickable items => Apartment’s details



Buy a Home

Some quick example text to build on the card title and make up the

Sell a Home

Some quick example text to build on the card title and make up the

Rent a Home

Some quick example text to build on the card title and make up the

Great choices from our listings

100000 EGP

0

Giza , Agouza

1 3 120 1

200000 EGP

16

Cairo , October

1 2 150 1

- **Apartment page:**

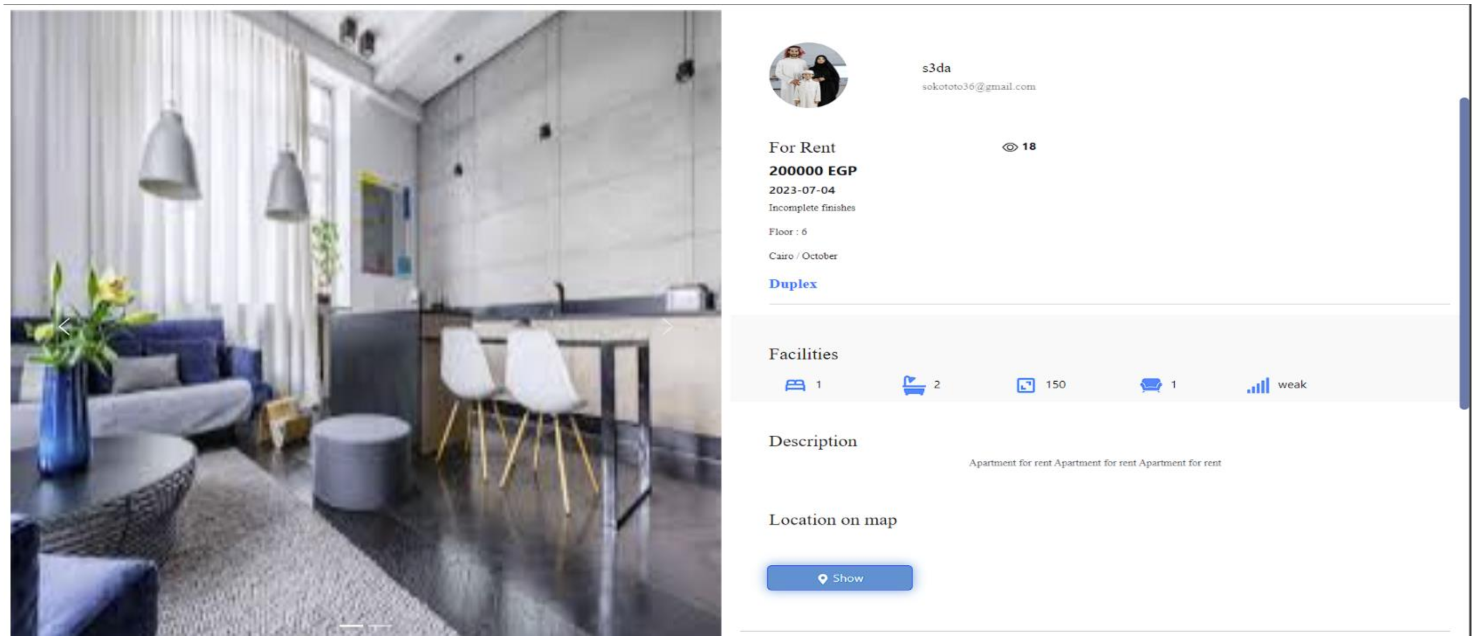
  Apartment's images

  Apartment's details

  Owner's details

  Chat icon: start chat room between the owner and the customer.

  WhatsApp icon: start chat between the owner and the customer using WhatsApp.



s3da
sokototo36@gmail.com

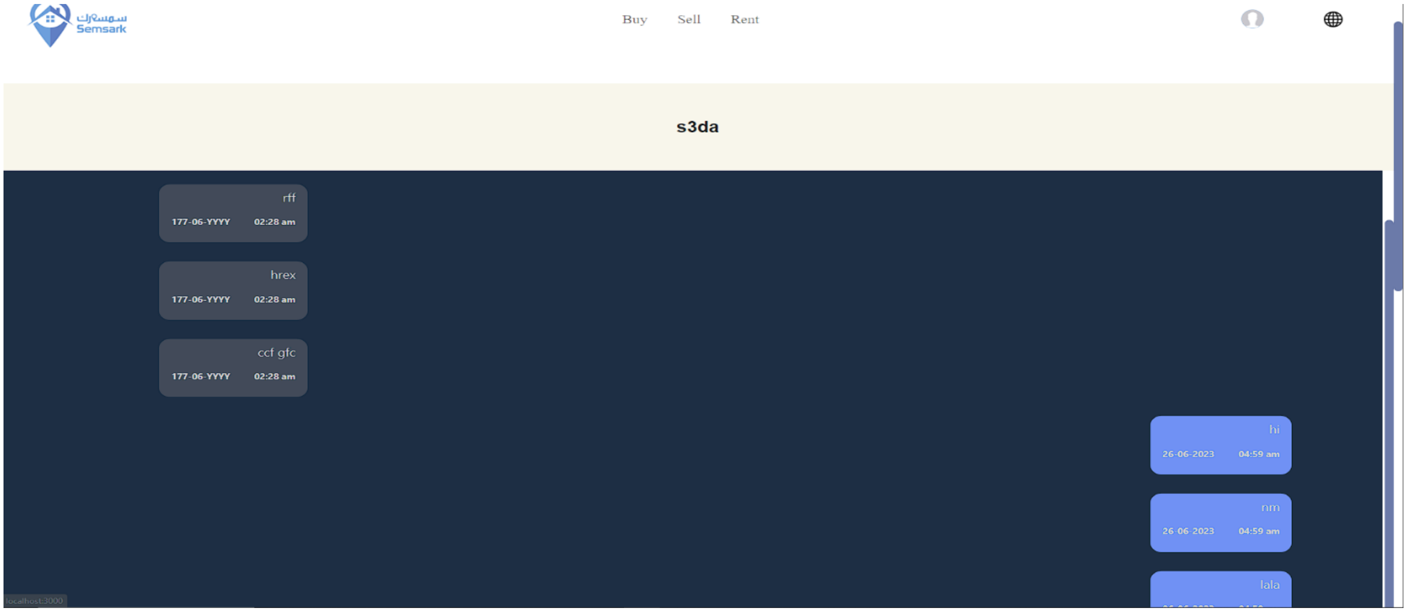
For Rent 👁 18
200000 EGP
2023-07-04
Incomplete finishes
Floor : 6
Cairo / October
Duplex

Facilities
1 2 150 1 weak

Description
Apartment for rent Apartment for rent Apartment for rent

Location on map

- **Chat page:**



s3da

rff
177-06-YYYY 02:28 am

hrex
177-06-YYYY 02:28 am

ccf gfc
177-06-YYYY 02:28 am

hi
26-06-2023 04:59 am

nm
26-06-2023 04:59 am

lala

- **Profile page:**

  User information

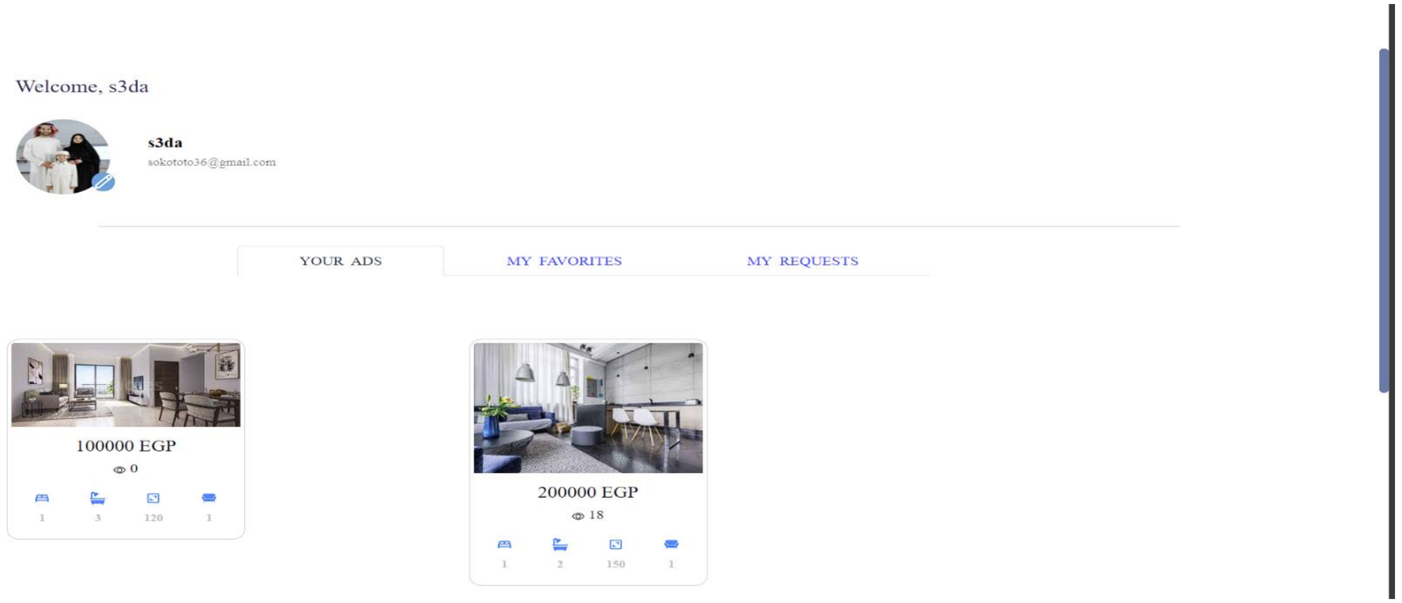
  Edit icon=> Edit my profile

  Three tabs:

  My Favorites: list of favorites apartments

  My ads: list of user’s apartments

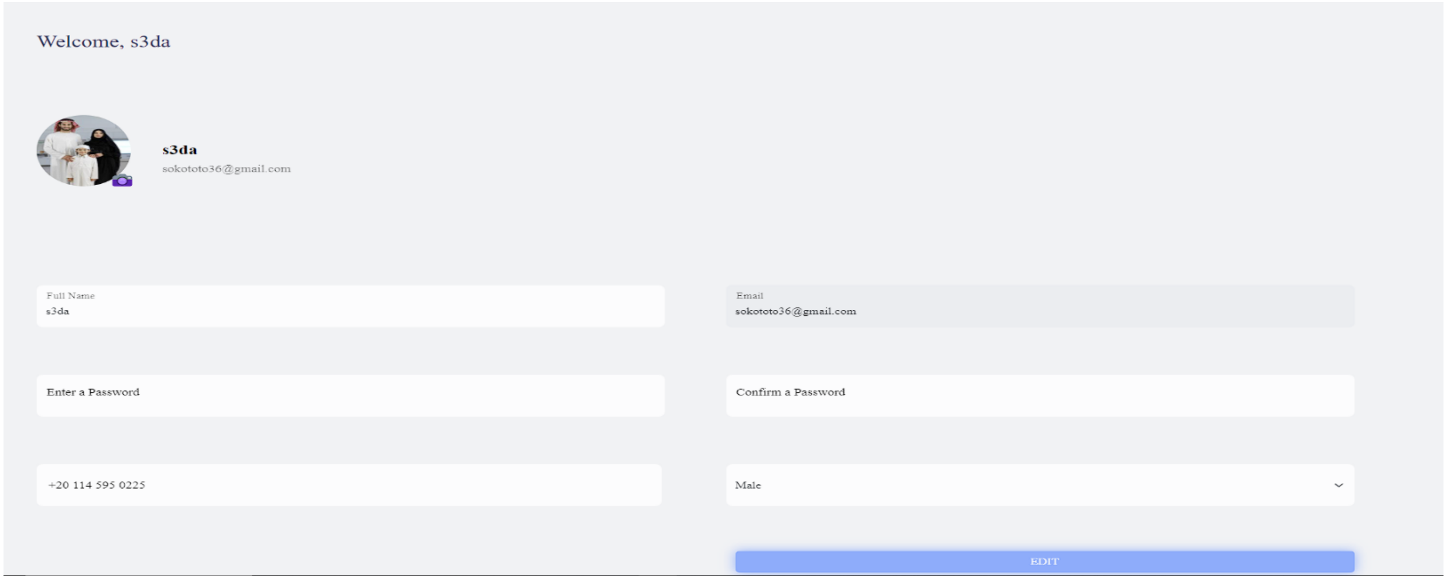
  Pending: List of user’s pending / suspended apartments



Welcome, s3da

s3da
sokototo36@gmail.com

- **Edit my profile page:** Contains user information fields which user can edit it.



Welcome, s3da

s3da
sokototo36@gmail.com

Full Name
s3da

Email
sokototo36@gmail.com

Enter a Password

Confirm a Password

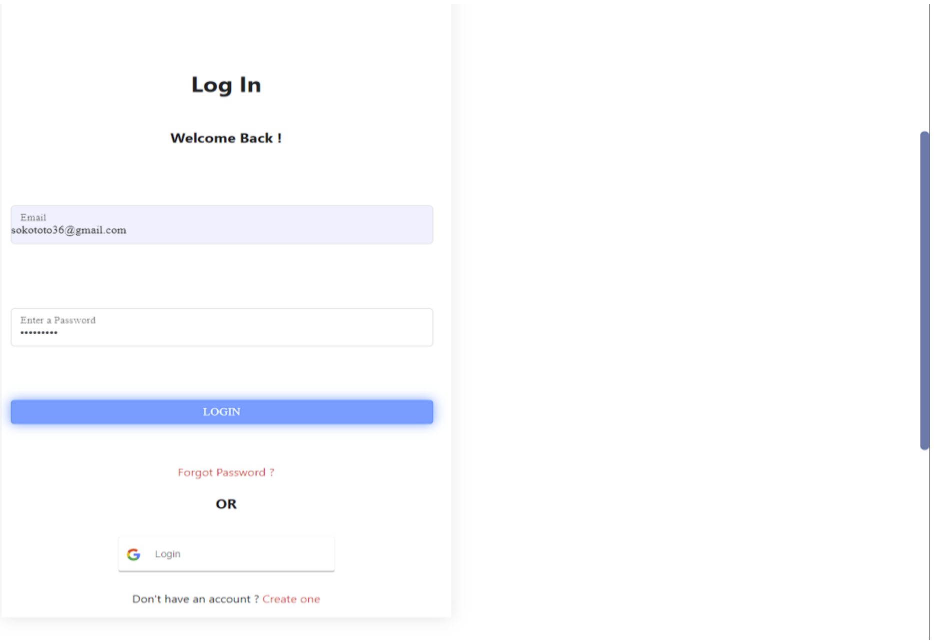
+20 114 595 0225

Male

- **Login page:**

  Two fields: Email and password

  Forget password => Forget password page



Log In

Welcome Back !

Email
sokototo36@gmail.com

Enter a Password
•••••••••

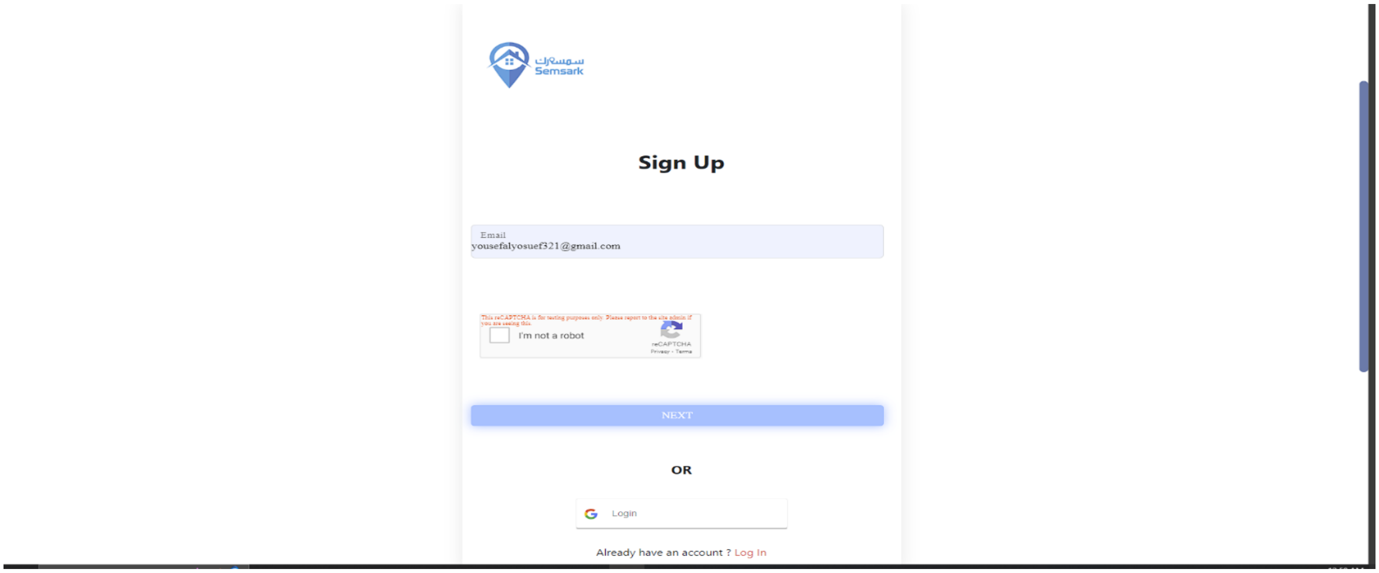
Don't have an account ? Create one

- **Signup page:**

  Signup with Google

  Email field

  Signup button => send OTP => Verification page (6 digits) =>Edit my profile



Sign Up

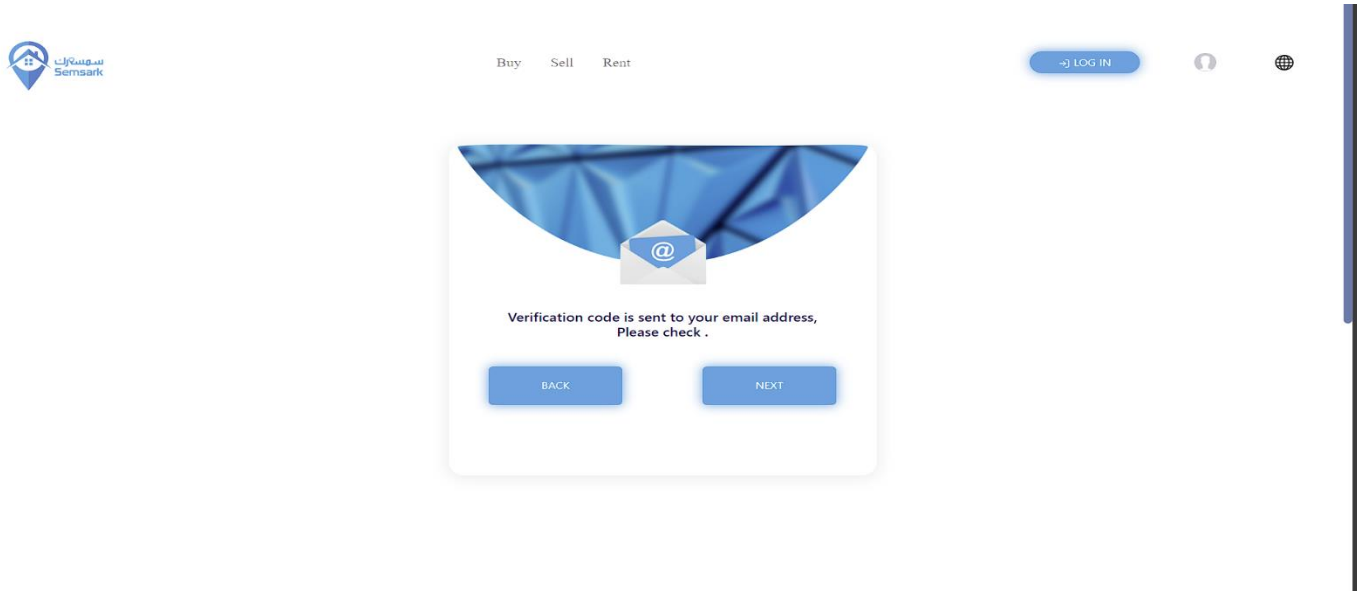
Email
yousefalyosuef321@gmail.com

This reCAPTCHA is for testing purposes only. Please report to the site admin if you are seeing this.

□ I'm not a robot

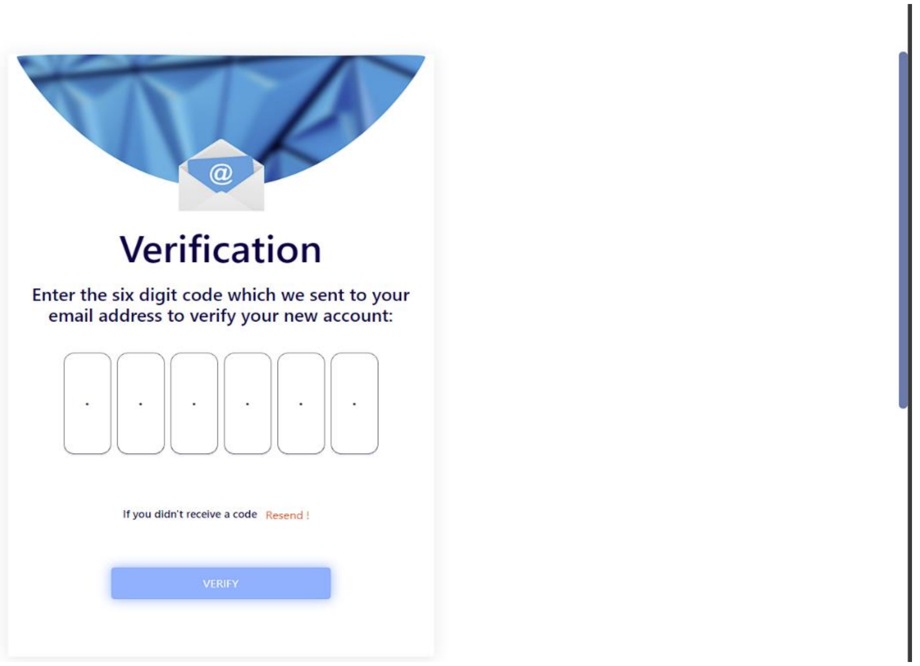
Already have an account ? Log In

- **Send OTP page:**



Verification code is sent to your email address,
Please check .

- **Verification page (6 digits):**



Verification

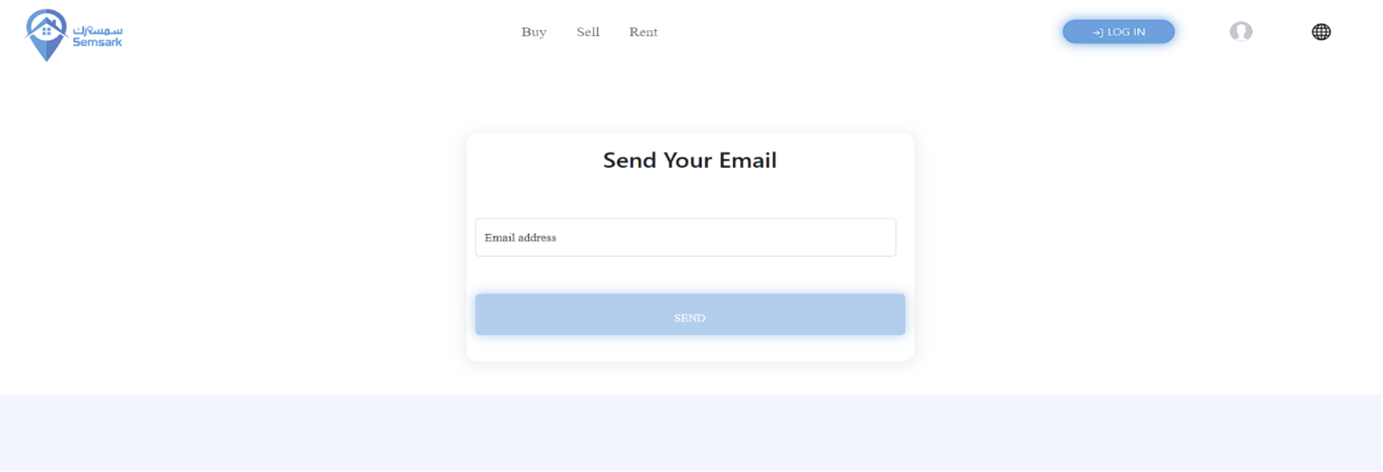
Enter the six digit code which we sent to your email address to verify your new account:

If you didn't receive a code Resend !

- **Forget password page:**

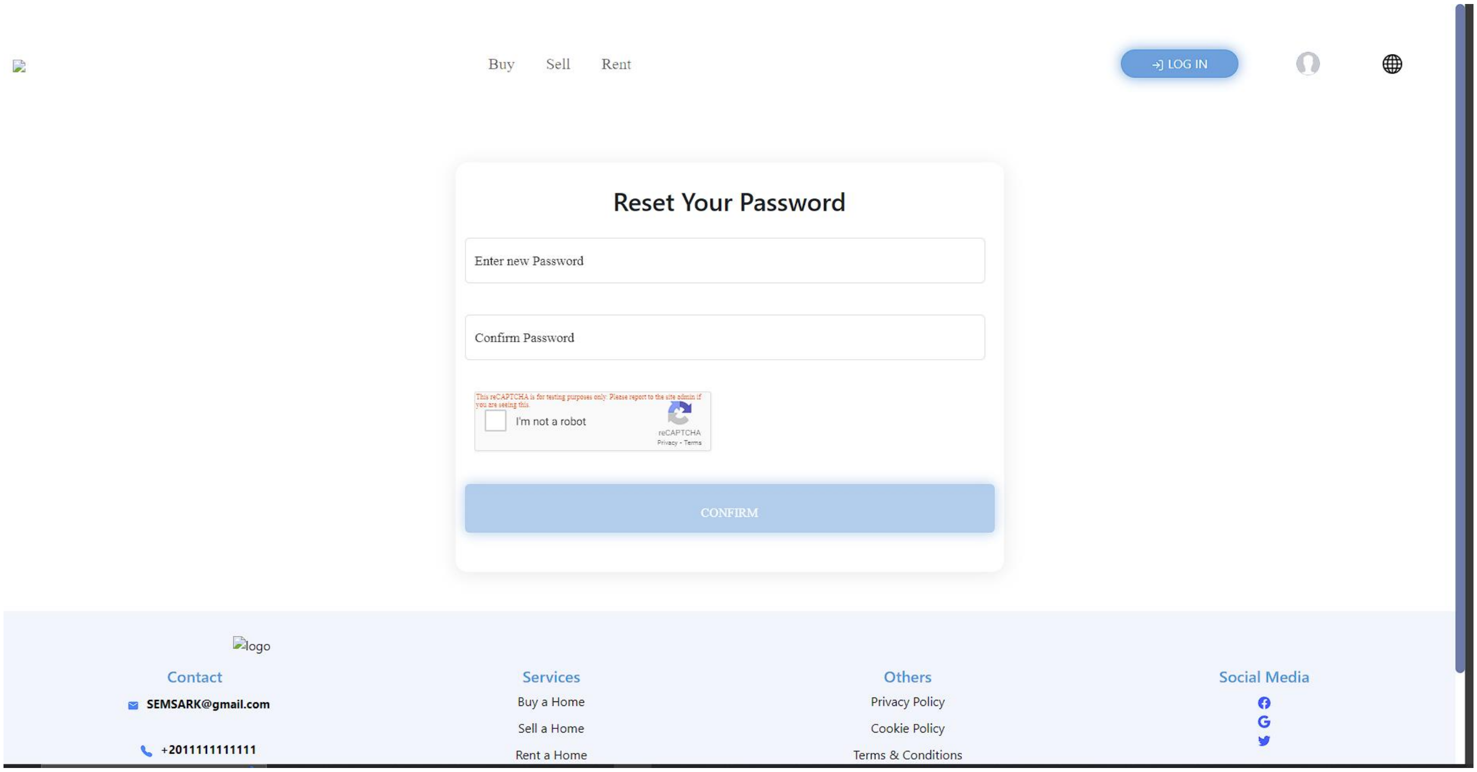
  Email field

  Send button => send OTP => Verification page (6 digits) =>Reset password page



Send Your Email

- **Reset password page:** Two password fields

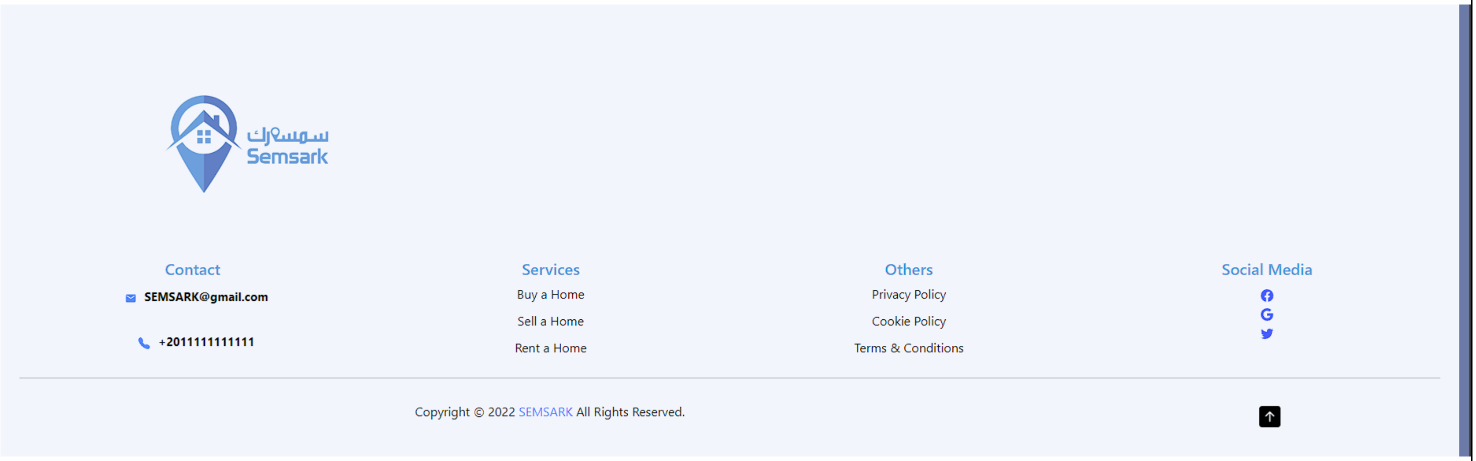


Reset Your Password

This reCAPTCHA is for testing purposes only. Please report to the site admin if you are seeing this.

- **Footer:**

SEMSARK logo
Sell => App store to download SEMSARK app
Rent => page contains on all apartments for rent
Buy => page contains on all apartments for buy
SEMSARK social media
Copyrights



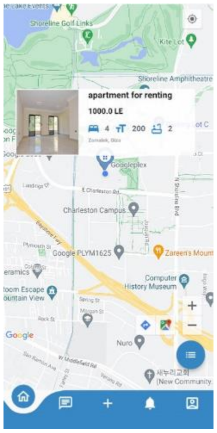
Contact	Services	Others	Social Media
SEMSARK@gmail.com	Buy a Home	Privacy Policy	
+2011111111111	Sell a Home	Cookie Policy	G
	Rent a Home	Terms & Conditions	

Copyright © 2022 SEMSARK All Rights Reserved.

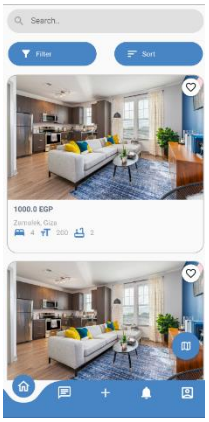
Mobile:

- **Home page:**

  Map Page: showing the nearest Advertisement as marker on map.
  List Page: showing the nearest Advertisement as a list**.**
  Search input
  Floating Action Button: that swap view to list of Apartments**.**



apartment for renting
1000.0 LE
4 200 2
Zamalek, Giza

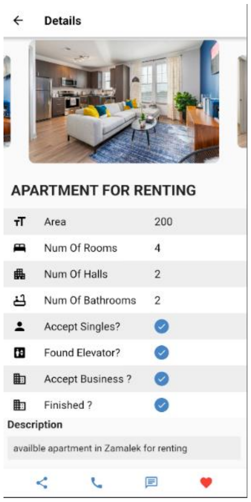


Search..

1000.0 EGP
Zamalek, Giza
4 200 2

- **Advertisement page:**

  Apartment's images
  Apartment's details
  Owner's details
  Chat icon: start chat room between the owner and the customer.
  Call icon: start call between the owner and the customer.
  WhatsApp icon: start chat between the owner and the customer using WhatsApp.



Details

APARTMENT FOR RENTING

Area	200
Num Of Rooms	4
Num Of Halls	2
Num Of Bathrooms	2
Accept Singles?	✓
Found Elevator?	✓
Accept Business ?	✓
Finished ?	✓

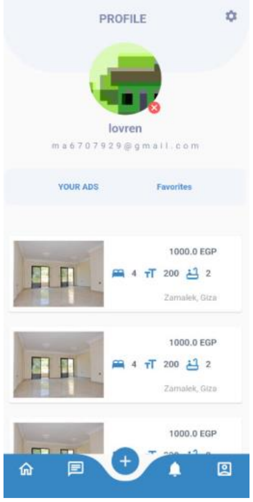
Description

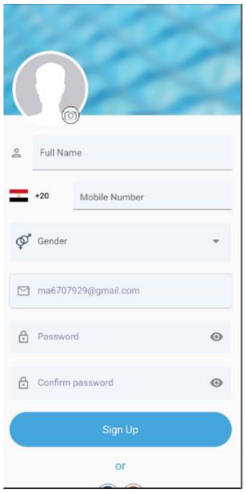
availble apartment in Zamalek for renting

- **Profile page:**

  User information
  Edit icon=> Edit my profile
  Three tabs:

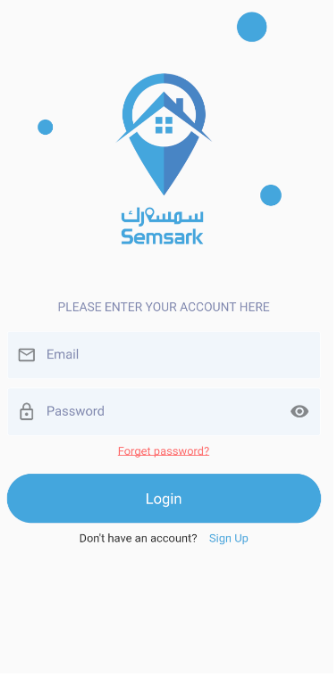
  My Favorites: list of favorites apartments
  My ads: list of user's apartments





- **Login page:**

  Two fields: Email and password
  Forget password => Forget password page

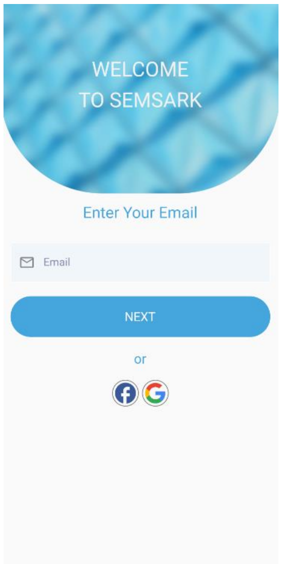

سمسارك
Semsark
PLEASE ENTER YOUR ACCOUNT HERE
Email
Password
Forget password?
Login
Don't have an account? Sign Up

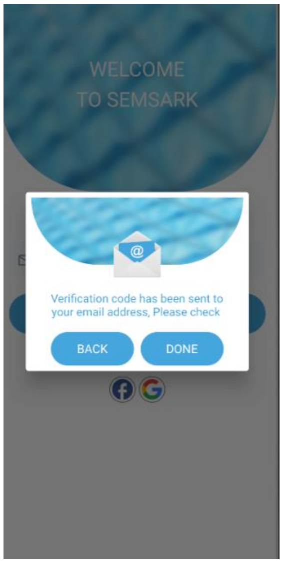
- **Signup page:**

  Signup with Google

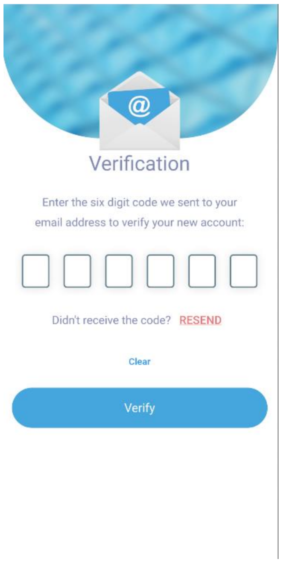
  Email field

  Signup button => send OTP => Verification page (6 digits) =>Edit my profile



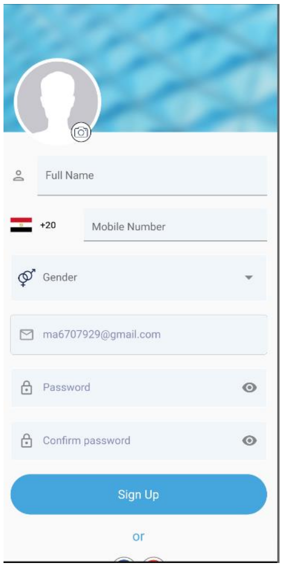


Verification code has been sent to your email address, Please check



Enter the six digit code we sent to your email address to verify your new account:

Didn't receive the code? RESEND

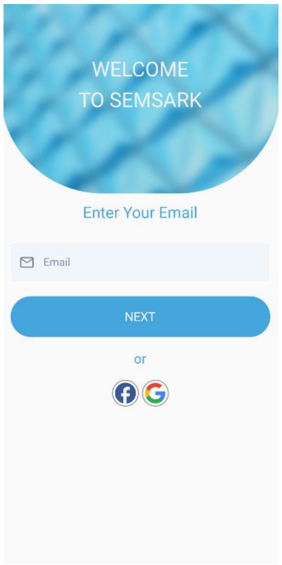


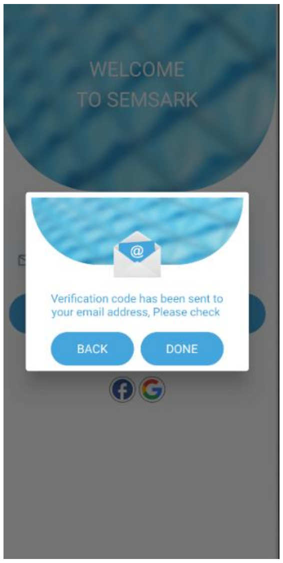
ma6707929@gmail.com

- **Forget password page:**

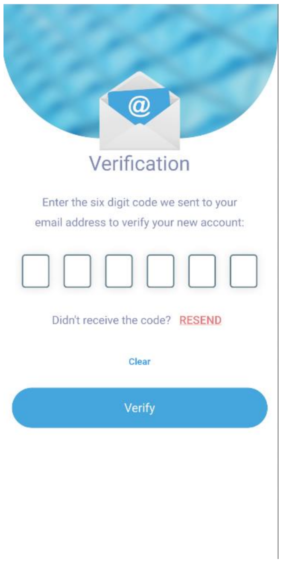
  Email field

  Send button => send OTP => Verification page (6 digits) =>Two password fields



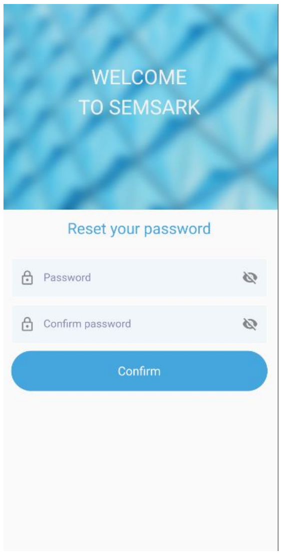


Verification code has been sent to your email address, Please check

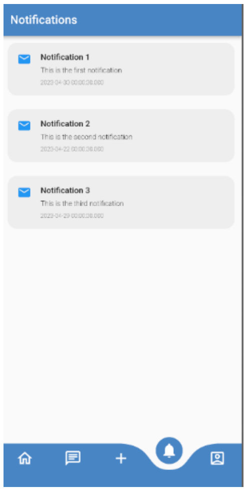


Enter the six digit code we sent to your email address to verify your new account:

Didn't receive the code? RESEND



- **Notifications page: List of Notifications**



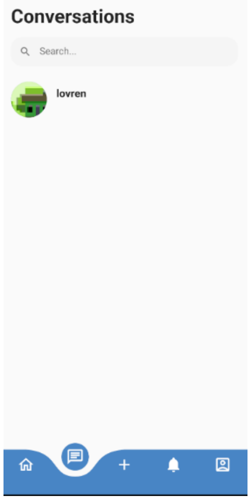
Notifications

Notification 1
This is the first notification
2023-04-30 00:00:00.000

Notification 2
This is the second notification
2023-04-22 00:00:00.000

Notification 3
This is the third notification
2023-04-29 00:00:00.000

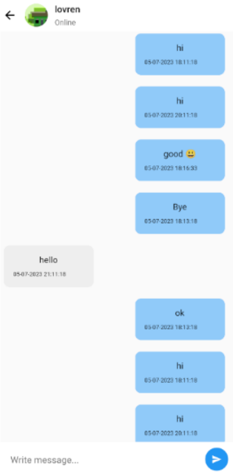
- **Chat page: List of Chats**



Conversations

Search...

lovren



lovren
Online

hi
05-07-2023 18:11:18

hi
05-07-2023 20:11:18

good 😃
05-07-2023 18:16:33

Bye
05-07-2023 18:13:18

hello
05-07-2023 21:11:18

ok
05-07-2023 18:13:18

hi
05-07-2023 18:11:18

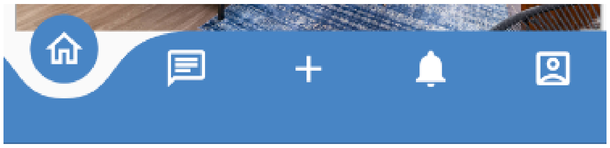
hi
05-07-2023 20:11:18

Write message...

- **Bottom Navigation Bar:**

  Contain 5 Tabs:

  Home Page
  Chat Page
  Add Advertisement Page
  Notifications page
  Profile Page



Chapter 6

TESTING

6.1 Test Cases

Test Case ID:	1
Test Case Name:	Login
Input Data:	Email, Password
Test Case Description:	Precondition: user must provide email and password Action: user can login to system
Expected Result:	**Success**: Token return and user can login to system **Failed**: Token return and user can login to system

Test Case ID:	2
Test Case Name:	Sign up
Input Data:	Username, email, password, gender, phone, image
Test Case Description:	**Precondition**: user must provide email, password, username, gender, phone, and image **Action**: user can create account on system
Expected Result:	**Success**: Account added to database system **Failed**: Account not added to database system

Test Case ID:	3
Test Case Name:	Forget Password
Input Data:	email
Test Case Description:	**Precondition**: user must provide email **Action**: user can change his password
Expected Result:	**Success**: Password changed successfully in database system **Failed**: Password not changed, and user can’t login to system.

Test Case ID:	4
Test Case Name:	Create Advertisement
Input Data:	Title, price, number of bathrooms, number of halls, number of rooms, area, latitude, longitude, description, image of (apartment – duplex – villa), type of advertisement, category (Rent – sell), accept business, advertisement allow for single not family, elevator
Test Case Description:	**Precondition**: User must provide Title, price, number of bathrooms, number of halls, number of rooms, area, latitude, longitude, description, image of (apartment – duplex – villa), type of advertisement, category (Rent – sell), accept business, advertisement allow for single not family, elevator. **Action**: user create advertisement on system
Expected Result:	**Success**: advertisement added to database system and another user can see it. **Failed**: advertisement not added to database system and another user can’t see it.

Test Case ID:	5
Test Case Name:	Verify user
Input Data:	Image in his national id and personal image
Test Case Description:	**Precondition**: User must provide two images for national id and personal image. **Action**: user will be verified and can create advertisements if there is similarity between two images.
Expected Result:	**Success**: user will be verified and can create advertisements **Failed**: user will not be verified and can’t create advertisements

Test Case ID:	6
Test Case Name:	Recommend advertisements
Input Data:	Id of clicked advertisement
Test Case Description:	**Precondition**: System detect id of advertisement when user click on it **Action**: System recommend similar advertisement.
Expected Result:	**Success**: System recommend similar advertisement. **Failed**: System can't find similar advertisements.

Test Case ID:	7
Test Case Name:	Update profile
Input Data:	Updated data like (username – password – image)
Test Case Description:	**Precondition**: user provide new data **Action**: user profile updated in database system and all users can see updated data.
Expected Result:	**Success**: profile updated successfully in database system. **Failed**: updated data may be the same as another user and profile not updated.

Test Case ID:	8
Test Case Name:	Update advertisement
Input Data:	Updated data like (title – elevator – advertisement image – signal power - price)
Test Case Description:	**Precondition**: user provide new data of his advertisement **Action**: advertisement updated in database system and all users can see updated advertisement.
Expected Result:	**Success**: advertisement updated successfully in database system. **Failed**: updated data may not meet the constraints and advertisement not updated.

Test Case ID:	9
Test Case Name:	Delete advertisement
Input Data:	Id of advertisement
Test Case Description:	**Precondition**: user chooses what advertisement wants to delete it. **Action**: advertisement deleted from database system and other users can’t see access to this advertisement.
Expected Result:	**Success**: advertisement deleted from database system. **Failed**: advertisement not deleted from database system.

Test Case ID:	10
Test Case Name:	Chat with advertisement creator
Input Data:	Id of user
Test Case Description:	**Precondition**: user clicked on chat icon and system detect id of user **Action**: user can chat with advertisement creator.
Expected Result:	**Success**: user can chat with advertisement creator.

6.2 Testing results

Test case ID	Test case name	Actual result	Defect?	Defect description	Fixed?
1	Login	Token retuned and login to system	No	_	_
2	Sign up	User created in system	No	_	_
3	Forget password	Password updated	No	_	_
4	Create advertisement	Advertisement created	No	_	_
5	Verify user	The similarity has been found	Yes	Image not cleared	Yes
6	Recommend advertisements.	The advertisement recommended	Yes	The evaluation needs user interaction	Yes
7	Update profile	Profile updated	Yes	Updated data already another user choose it	Yes
8	Update advertisement	Advertisement updated	No	_	_
9	Delete advertisement	Advertisement deleted	No	_	_
10	Chat with advertisement creator	Chat is worked and no delay	Yes	There are some challenges with performance	Yes

Chapter 7

EVALUATION

7.1 Project overview

- Semsark is an innovative application developed as a graduation project. It serves as a platform for renting and selling apartments, villas, and other properties. The application allows users to showcase advertisements in both list and map views, add their own advertisements, and incorporates face recognition technology and a recommendation system. The project was implemented using Spring Boot, Flutter, machine learning, deep learning, and React JS, ensuring a seamless user experience across different platforms and devices.

7.1.1 Achievements Made:

- During the development of the "Semsark" application, several achievements were made. These include:

- Designing an intuitive user interface: The project team successfully created a user-friendly interface that enables users to navigate the application easily, search for properties efficiently, and manage their advertisements effectively.

- Implementing property search functionality: The application incorporates a robust property search feature that allows users to specify their preferences and find suitable properties based on location, price range, property type, and other criteria.

- Developing advertisement display options: The project team implemented two viewing options for advertisements - a list view and a map view. This allows users to browse through properties in a compact list format or visually explore properties on a map.

- Integrating face recognition technology: The application successfully incorporates face recognition technology, enabling users to securely verify their identities during the rental or sales process. This enhances trust and security within the app.

- Creating a recommendation system: The team developed a recommendation system that analyzes user preferences, browsing behavior, and historical data to provide personalized property suggestions. This feature assists users in discovering relevant listings based on their interests and requirements.

7.1.2 Lessons Learned:

- Throughout the project, the team learned valuable lessons that can be applied to future endeavors. These lessons include:

- Learning new technologies during the project we learned How to deal with challenges and know very useful technologies as Spring boot which we use it to develop back-end side and know how to make apis, how to build database of project and define the relations between tables and integrate with another platform as firebase features and we learned a flutter framework to implement our application

designs and we learned to add machine learning and deep learning in our project else we learned React Js to implement Website to project.

- Proper planning and project management: Adequate planning and effective project management are crucial for the success of a project. The team realized the importance of setting clear goals, establishing a timeline, and allocating resources efficiently to ensure a smooth development process.

- Regular communication and collaboration: Open and regular communication among team members is essential for coordinating efforts, sharing progress, and addressing challenges effectively. The project team learned the importance of maintaining clear lines of communication throughout the development cycle.

- User feedback and iterative improvements: Incorporating user feedback and continuously improving the application based on user needs and expectations is vital. The team recognized the value of user testing and incorporating iterative improvements to enhance the user experience.

7.2 Possible Future Work in This Area:

There are several potential areas for future work and enhancements in the domain of the "Semsark" project. These include:

Payment integration: Integrating secure and convenient payment options within the application can streamline the rental and sales process, allowing users to make transactions directly through the app.

Electronic contracts: the user can make electronic contracts with another user.

Enhanced security features: Continuously improving the security measures within the application, such as implementing two-factor authentication and encrypted data storage, can provide users with peace of mind and protect their sensitive information.

Integration with third-party services: Collaborating with third-party services, such as real estate agencies, property evaluation platforms, or home staging services, can enrich the user experience and provide additional value-added services to users.

Machine learning and AI-based features: Leveraging machine learning and AI algorithms can enable the application to provide more accurate property recommendations, predictive pricing models, and intelligent search capabilities. This can enhance the user experience and offer personalized suggestions based on individual preferences and behaviors.

Expansion to other property types: While the current focus of the application is on apartments, villas, and similar properties, future work can involve expanding to other property types such as commercial spaces, land plots, or vacation rentals. This can cater to a broader range of users and diversify the available listings.

In summary, the "Semsark" project has achieved significant milestones in developing an application for renting and selling properties. The project team has learned valuable lessons in planning, communication, and user-centric design. By considering possible future work areas, such as payment integration, advanced analytics, and augmented reality, the application can continue to evolve and provide an enhanced experience to its users while staying at the forefront of the real estate industry.