Helwan University

Faculty of computers and information
Information Systems Department

Supervised by:

DR. Laila Abdel-Hamid

SAFE ZONE

Graduation Project Presented by:

Ahmed Abdel-Sattar Othman Awed [20180052]
Omar Youssef Farouk Ahmed [20180395]
Ahmed Saeed Hassan Gouda [20180040]
Tawfiq makram Mikael Shafiq [20180171]
Mohamed Samir Abdel-Aziz Abdel-Rahman [20180503]

Submitted in fulfillment of the requirements for the degree of Bachelor of Science in Computers & Artificial Intelligence, at Information Systems Department, the Faculty of Computers & Artificial Intelligence, Helwan University

ACKNOWLEDGEMENT

First and foremost, we would express our deepest gratitude and appreciation to our advisor **Dr. Laila Abdel-Hamid** for her support, outstanding guidance, and encouragement throughout our graduation project. As well as **Engineer. Helal Ahmed** for his continuous support and help as he was always there for us providing more than the needed time and effort from him since the very beginning of our project.
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.

Table of Content:

Lists of Figures

Abstract

This program solves a daily social problem in all societies, it helps people who have lost their things to get them back by finding a way to communicate between them and the people who found these things, and the application also helps you know how safe or dangerous the places you want to visit are. To obtain information to warn of any potential risks.

Chapter one: 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. Introduction

1.1 Overview

First, our application Safe Zone is a social moral Application that works on all platforms for helping people who lost or stolen something from them.

We help them by matching the posts that they share. The application will match between the lost items posts and the found items post automatically. And will help users to contact each other, and get their things back.

On the other side, we will help others who have things stolen from them by trying to make a small social media that users can share the posts that will help them by providing them with any information that's will help them to reach for their stolen things. these posts will warn others from dealing with thieves who have stolen these things too.

The app name comes from the warning feature that will provide users with information about the security of places which they will visit.

The app name comes from the warning feature that is will provide users with information about the security of places which they will visit.

1.2 Problem statement

In our usual daily life, many people during their day may lose their things. too they may be stolen.so they will need help to find the things which they lost.

when people found any lost things they don't know who were lose these things so, they can't return these things to owners. Also, when we are robbed, these things can
be used in unethical ways such as fraud, so others must be warned.

It is also possible that we want to go to a place that we do not know well. We may need information about how safe that place is, so whoever needs something to tell us how safe that place is.

1.3 Objectives

- Use the application on many platforms and almost possible vices.
- The best possible rate of accuracy in matching between different posts.
- We will try to achieve the previous goal by trying to divide things into many categories so that it is easier for us to match them in any effective way.
- Use English as the main language in the application to help all different people with many cultures to use the application easily.
- We will try to achieve the previous goal by trying to use the icons on the buttons, facilitating the number of steps, and reducing the number of navigations between pages.
- Trying to make the user interface easy to use and comfortable makes the fewest number of steps to do a particular job.

- Attempting to build a secure application while fully preserving the privacy of users and trying to prevent fraud and sabotage methods while trying to match the missing items with the ones that were found.
- Achieving the goal by achieving the matching automatically in the background without the need for human intervention.
- Achieving the goal by achieving the matching automatically in the background without the need for human intervention.
- Users will find matching items through their notification list and thus we will enable them to communicate using the users' information.
- Users can try to find their lost items multiple times by republishing their posts that did not match as trying to find others.
- Users can communicate with others to try to warn or obtain any information about their stolen things. This is done through small social networking.

- Users can verify safe areas through the information we collected from places where the theft occurred.
- We will definitely maintain the security level while verifying users when logging in.
- Ensure and maintain the application's work inside Egypt, not outside it, and we will achieve this by identifying specific sites previously.

1.4 Scope of work

Our application is an application that works on all platforms and operating systems and is limited to people inside Egypt in pre-defined areas, and the search for certain things is limited by categorizing them to facilitate and speed up the matching process and ensure its accuracy. The system stores the information it needs in the matching process and in the process of obtaining safety information for different places in a huge database. The system helps users communicate with each other to retrieve their lost items.

1.5 Overview of remainder of document

1.5.1 General constraints:

In this section, we will introduce some of the general constraints in our project

Time: This is the main constraint in our project because we should set a specific time to finish the project, also we break down the project into tasks, and the task into smaller tasks, and so on, we set time constraints for each task.

Learning new technologies: Like flutter, Text similarity, firebase

Internet access: The users should have a device connect to the internet and create an account on the application to write reports or search in past reports, also to get a notification if a match occurs.

Matching posts: We should find and use an appropriate mechanism that produces the highest accuracy ratio in the matching process.

Indiscipline “Human factor”: like being late in delivering tasks or attending meetings.

1.5.2 Project description:

it's an application help people to retrieve their stolen or lost Personal belongings, user can post information about the item and the place and an image (if it exists), users also can post images that show robbery to alert other people and that help to find the thieves, the user chooses the category of the lost item then the system matches the report with other reports at the same category by text analysis and semantic analysis then the user receives a notification with the matched reports with information (phone number) to contact with the person who found the lost item.

when someone is robbed, he can open the maps and rate the street as not safe if many people rate the street as not safe the application will send this data and users can search for information of all available places.

Users can post something they lost and put a financial reward if someone has found it.

1.5.3 Assumptions and constraints

The use of our application is very easy because most users have been using social media apps and realize that there are some software techniques used to appear in future ads and notifications about events near to them or result of search according to things, they search for that match the written statement.

Context and background

1.5.4 Background

The user just needs to know how to use smartphones and have been using social media applications.

1.5.5 Related works

(Troov) mobile application.

The app idea is close to our idea, but we differentiate by an additional category of safe and unsafe place, and we provide more security and work hard to make the very good, best version that any other application can't compete with us. Also, some people use Facebook groups on purpose to search about lost thing owners or ask if anyone has seen their lost thing before anywhere.

1.5.6 Project benefits

1. create an environment include.

2. The easiest way to search about lost and stolen things and know about unsafe places.

3. The most suitable platform to present information about something has been found.

4. Make the process of a search easier because the post includes as many details as possible about the lost or found thing.

5. Make the process of search more efficient by reducing the search area depending on input location details written in a post.

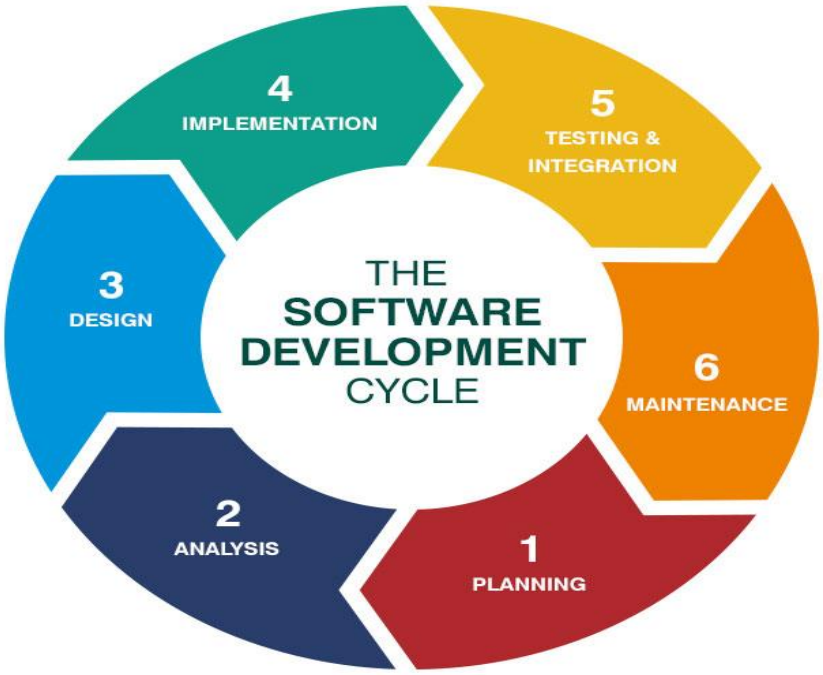
Chapter two: Planning and analysis

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. project “planning and analysis”

System Development Life Cycle

We followed system development life cycle approach in developing this project with its 7 phases, planning, analysis, design, development, testing, deployment and maintenance.



Methodology

We followed Agile analysis and design methodology in developing, this project and developed all related UML diagrams, represented in this document later.

Software development phases:

2.1 Planning

Very important phase through the phases to build our software. In this phase we will define the problem and show the following:

- Define problem.
- Define Project Scope and purpose.
- Estimate the cost, schedule.
- Identify the project team, team leader, and project manager.
- How will the software be used?
- What data will serve as the input of the software?
- What will be the output of the software?
- Who is going to use the software (who the stakeholders)?

2.1.1 Define problem

Many people during their day may lose their things. too they may be stolen. so, they will need help to find the things which they lost. when people found any lost things, they don't know who were lose these things so, they can't return these things to owners. Also, when we are robbed, these things can be used in unethical ways such as fraud, so others must be warned. It is also possible that we want to go to a place that we do not know well. We may need information about how safe that place is, so whoever needs something to tell us how safe that place is

2.1.2 Estimate the cost, schedule

We may need some cost to buy software and datasets to help our project expand its scope. The development of any project must be organized and have a clear time plan for the entire work of the project.

2.1.3 How will the software be used?

You can use the system by installing it on your smartphone then you should sign up in our system. You can make any report (lost, found, and stolen report) by filling out the information form. you will wait for the matching process if the matching is done successfully, you will need contact information to communicate with the other user.

2.1.4 What data will serve as the input of the software?

we need many inputs information through the use of the app. like user information, reports inputs that user's input, and other information we need it.

2.1.5 What will be the output of the software?

system output will be the result of the matching process through making a notification message to users. also, many features we will explain later will be sub outputs

2.1.6 Project Stakeholders

- Project Supervisor (Prof. Dr. Laila Abdel Hamid).
- System developers (the five of us).
- Teaching assistant responsible for guiding us.
- Any user of the system.

2.1.7 Gantt Chart

	Task Name	Duration	Start	End
1	Total Project	245	2021-10-1	2022-6-15
2	) Documentation chapter one (introduction	15	2021-10-1	-10-15 2021
3	Problem statement	4	2021-10-1	2021-10-4
4	objective	4	2021-10-5	2021-10-8
5	Scope of work	4	2021-10-9	-10-12 2021
6	Overview of remainder of document	3	-10-13 2021	-10-14 2021
7	) Documentation Chapter 2 (Background	10	-10-15 2021	-10-25 2021
8	) Documentation chapter three (Methodology	70	-10-26 2021	2022-1-5
9	Development approach and methodology	5	-10-26 2021	2021-11-1
10	Implementation of text similarity and image processing	15	2021-11-2	-11-17 2021
11	Planning and analysis	10	-11-18 2021	-11-28 2021
12	Requirements	7	2021-12-1	2021-12-7
13	Class diagram	4	2021-12-7	-12-11 2021
14	Use cases and scenarios	4	-12-12 2021	-12-16 2021
15	Sequence diagrams	4	-12-17 2021	-12-21 2021
16	Activity Diagrams	4	-12-22 2021	-12-26 2021
17	State Diagrams	4	-12-30 2021	2022-1-3
	Logo and slogan	4	2022-1-4	2022-1-8
18	Implementation	125	2022-2-25	2022-6-14
19	Front start	10	2022-2-25	2022-3-6
20	Frond end	10	2022-3-7	2022-3-17
21	Backend firebase	45	2022-3-18	2022-5-3
22	Backend firebase 2	10	2022-5-4	2022-5-14
23	Backend finish	10	2022-5-15	2022-5-25
24	testing	7	2022-5-26	2022-6-2
25	Generating android version	3	2022-6-3	2022-6-6
26	Launching app	3	2022-6-7	2022-6-10
27	Generating project demo	3	2022-6-11	2022-6-14

2.1.8 Feasibility study

Feasibility is defined as the practical extent to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software. Information such as resource availability, cost estimation for software development, benefits of the software to the organization after it is developed and cost to be incurred on its maintenance are considered during the feasibility study.

- **Operational feasibility**

  The application will be easy to use and make it easier for users to find their lost things and communicate with others to get their things back. It will also be safe for users and will preserve their privacy.

- **Technical feasibility**

  The hardware and software for developing, maintaining, and launching the application will be reliable and available for us. the application will not be the need for training users also. The system will be maintainable in the future and easy to connect with external systems and work on many different platforms.

- **Economic feasibility**

  The software will be developed by ourselves, and the cost of not developing the app affects us. the cost of firebase service if the project expanded in the future. we don't need any cost of the fees and license. also, we don't have any training cost for users.

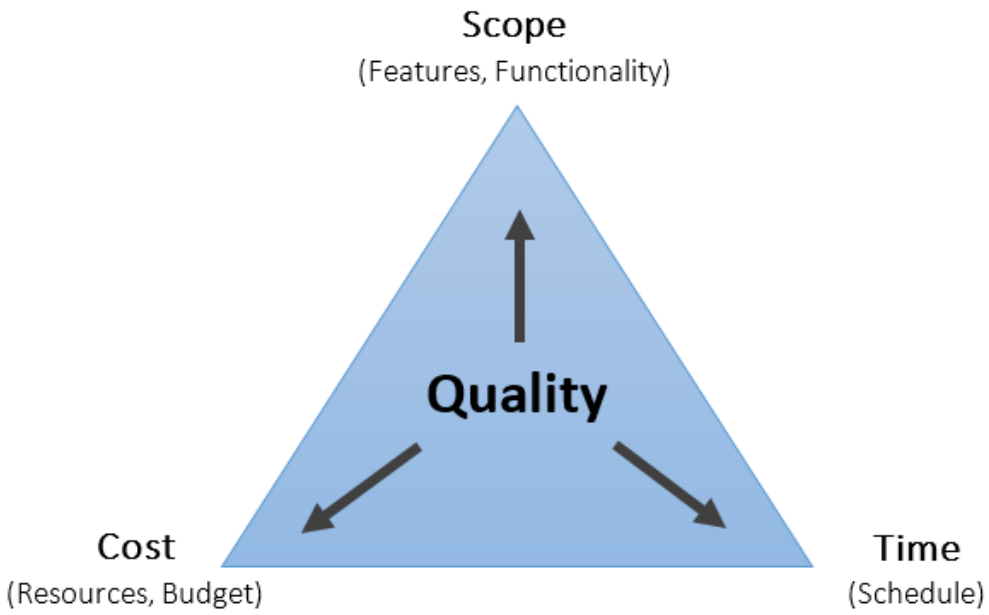
- **Schedule feasibility**

  Of course, we divided the application development stages into small tasks with a specific working duration with specific deadline.

2.2 Application constraints

The triple constraints of project management also known as the project management triangle or the iron triangles are scope, cost, and time. You'll need to balance these three elements in every project, and doing so can be challenging because they all affect one another.

- **Scope Constraints:** Project scope refers to a project's magnitude in terms of quality, detail, and deliverables. Time and money are dependencies of project scope. So we identify the project scope carefully as we explain before, because as the project scope grows, the project will require more time and money to complete
- **Time Constraints:** we have time constraints that we should finish the app completely, it should be ready at June, we have to make a buffer to avoid risks so we have to finish at half May maximum.



- **Cost Constraints:** Maybe we'll need to buy software which will give us more options than the free ones. Maybe we will need to buy datasets if we couldn't get it for free to expand our storage capacity. We will need budget to buy servers that will make the app resists multiple number of users and the parallel processing operations. We maybe some budget for upload projects on app store and google store. Also, we may need some budget for marketing. In addition to the cost of repair and maintenance.
- **Resource constraints:** we will need to use some tools to build our software like using flutter and dart using visual studio code. We need to use tools like firebase to support our back-end system. People resource is available.
- **Risk constraints:** Before the start of the project, we prepared a set of potential risks and tried to analyze and manage these risks, like operational risks, Scope creep, High costs, Time crunch, and Low performance.

- **Quality constraints:** all other constraints have some effect on quality constraints so, any defect in these constraints will result low-quality application. We try to assurance from application quality by making many testing and discovering the defects and errors to solve them before delivering the application.

2.3 Risk and risk management

We expected to face some problems during the development stages of our application.

We made a list of some expected risks and built a risk management plan to avoid these expected risks.

Some of these risks that we are expected to face are the risk of lack of time as a result of overcrowding the tasks and a lot of exams and projects during the semester.

We solve this risk by trying to make good time management. And also the risk of choosing the type of technology we will use to implement our application. We needed to learn new technology like Flutter to help us to make our project work on all platforms and operating systems. This was a risk for us.

Also, while working on our project, we faced some expected technical problems and tried to solve them correctly through research, experiment, and inquiries from experienced people to reach the best possible solution.

We faced some unexpected risks like the lack of communication between team members and between team members and the project supervisor, but we resolved it. another big unexpected problem that almost affected the progress of the project, and was a managerial error during the registration of the project, but we're getting organized and completed the project successfully.

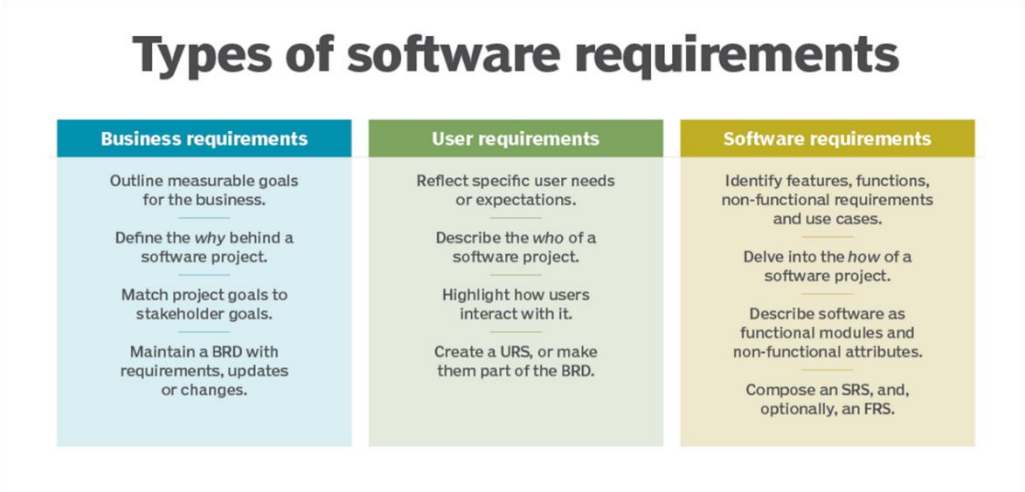
2.4 Analysis

Software requirement

- ### 2.4.1 Business requirements

  The Safe Zone application is software that will help people to search for their lost things to find them and get them back.

  The system must search for lost objects. Using some input data from users and then matching that data with data extracted from the posts of the found items.



- ## 2.4.2 User requirements

User needs are represented in several actions that in turn require several responses from the application in response to the desired. Users need help from applications to find their lost items. The user makes a report with some needed information. Then the application response whether it was found or not. Users need to communicate with others to get their lost items back so, the application will provide them with this information.

Users need to see all stealing posts to get a warning from them and help others with any information by a comment on these posts.

In addition to some other features that we add in case you get robbed, you can submit a stealing report. Also users can search to get some information about the security of specific places they will visit.

- ## 2.4.3 Software requirement

➢ **Domain requirements:** Authorized users should access all application features at any time except in case of maintenance of the software.

Users should have a response when they make an action on the application. The application should support all security and privacy rules. And support data integrity.

their lost things to find them and get them back.

- **Functional requirements:** In software engineering and systems engineering, a functional requirement defines a function of a system or its component, where a function is described as a specification of behavior between inputs and outputs.

  Functional requirements may involve calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish. The following are some of the functional requirements that the system must achieve.

  - ✓ **The authentication process of registering and logging into the system:** The system must allow users to register in the application after requesting several necessary information. The system must also allow users to log into the application after the authentication process has been performed.
  - ✓ **Authorization:** The system should maintain authorization levels by making users see what they own.
  - ✓ **Report post:** The system must allow users to prepare a report, whether lost, found, or stolen, after filling in all the necessary information.
  - ✓ **Edit and delete posts:** Users can select their previous report and then can edit or delete it.
  - ✓ **Get Notification:** Users must be informed of the outcome of the matching process, whether it was successful or unsuccessful.
  - ✓ **Edit profile:** Users can edit their profiles especially their contact information.
  - ✓ **See all stolen posts:** Users can see all stolen report.
  - ✓ **Comment on stolen reports:** User can comment on the others stolen posts.
  - ✓ **Log out:** Users can logout after they finish using their account and can login in any other time from any device.

- ✓ **Nonfunctional requirements:** Non-Functional Requirements are the constraints, or the requirements imposed on the system. They specify the quality attribute of the software. Non-functional requirements deal with issues like scalability, maintainability, performance, portability, security, reliability, and many more. Non-Functional Requirements address vital issues of quality for software systems. If Non-Functional Requirements are not addressed properly, the results can include:
  - Users, clients, and developers are unsatisfied.
  - Inconsistent software.
  - Time and cost overrun to fix the software which was prepared without keeping NFRs in mind.

  There are many types of non-functional requirements we will explain some of them that we applied to our application.
    - ✓ **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.
    - ✓ **Performance:** The performance of our application is high in terms of the quality of the result, the speed of response, and the speed of recovery from any failure.
    - ✓ **Maintainability:** Is the ability to maintain system bugs and issues with less loss that doesn't affect the services or stop it. With ability of detecting these bugs and solving it.

Nonfunctional requirements: (cont.)

- ✓ **Usability:** Our application is easy to use, does not require training, is understandable for all users, and is also comfortable during use.
- ✓ **Capacity:** The maximum number of the concurrent users is unlimited users.

- ✓ **Reliability:** Reliability is closely related to availability which is typically described as the ability of a component or system to function at a specified moment or interval of time. Our application has the ability of fast recovering from failure.
- ✓ **Security:** This is security for such as security audits, cryptography, user data, system identification, system authentication, resources utilization... etc.
- ✓ **Scalability:** Our project can be expanded and developed in addition to the ability to increase its features. We can increase of decrease users and features.
- ✓ **Portability:** For sure, our application supports portability it is developed by Flutter so, the application can work on many different platforms.

Chapter three: Design

In this chapter we're going to discuss and go deeper in how we plan the project and show the steps and the instructions that we've followed to plan the application.

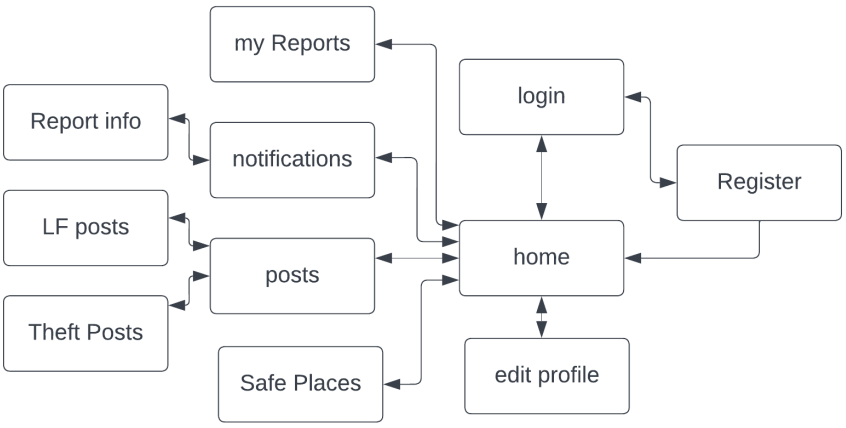
3 Design

• 3.1 Hardware design

- ✓ During the development, we will need laptops or pcs.
- ✓ After deployment to use the application you should use smartphones, pcs, or laptops.

• 3.2 Interface design

- ✓ We tried to create a commercial identity for our application and chose the appropriate colors and fonts and made a logo expressive of our application.
- ✓ We designed how the users flow while using the application.
- ✓ We made a design for all the pages of the program to make a UI design using adobe XD.
- ✓ **User Flow:**



- ✓ **Brand identity**

  Brand identity is the visible elements of a brand, such as color, design, and logo, that identify and distinguish the brand in consumers' minds. Brand identity is distinct from brand image. The former corresponds to the intent behind the branding and the way a company does the following—all to cultivate a certain image in consumers' minds.

  We try to make a special brand identity for our application and follow all possible ways to create our visual identity.

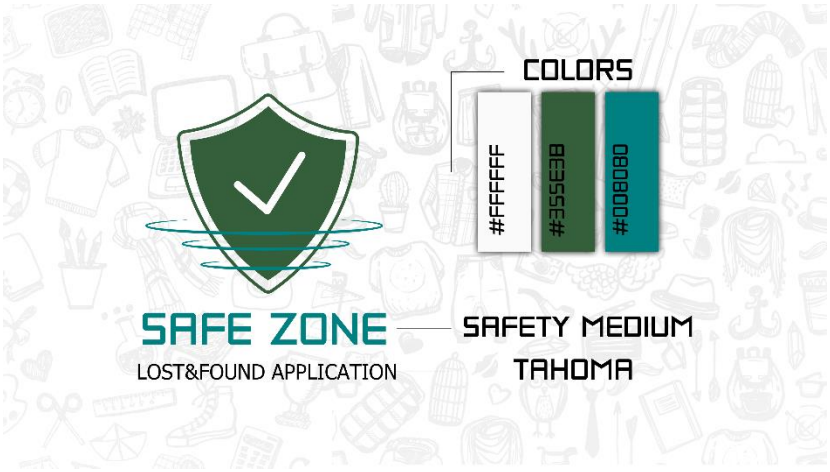
  First, we select our name inspired from our idea (Safe Zone).

  Then we choice the suitable colors and fonts which reflect the application purpose.

  Finally, we can make the logo for our application with use shapes describe the application.

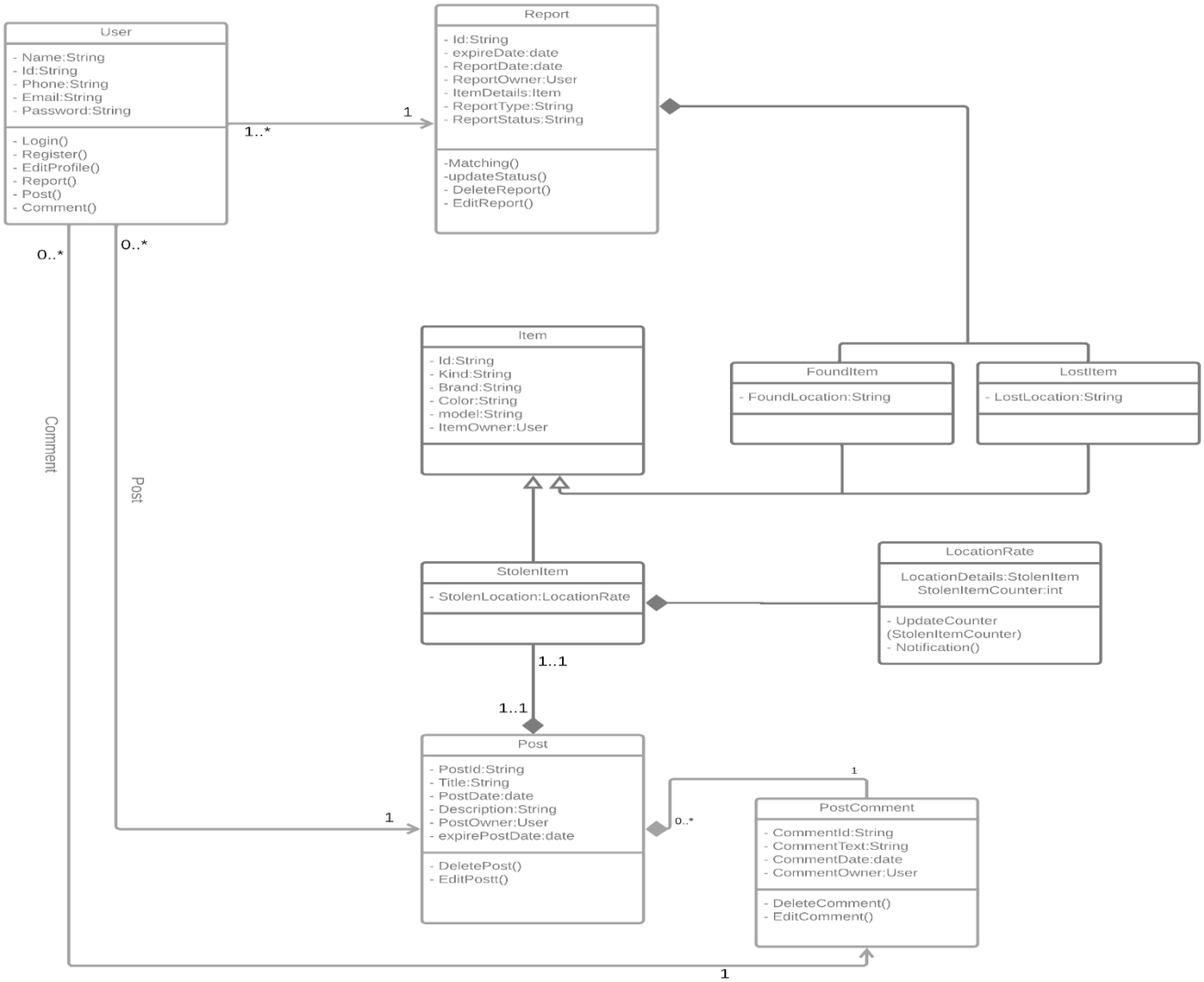
  We select the colors like teal green to reflect the safety feeling that our application should reflect it.

  Also, we select the shield to sign the safety which application provide it.

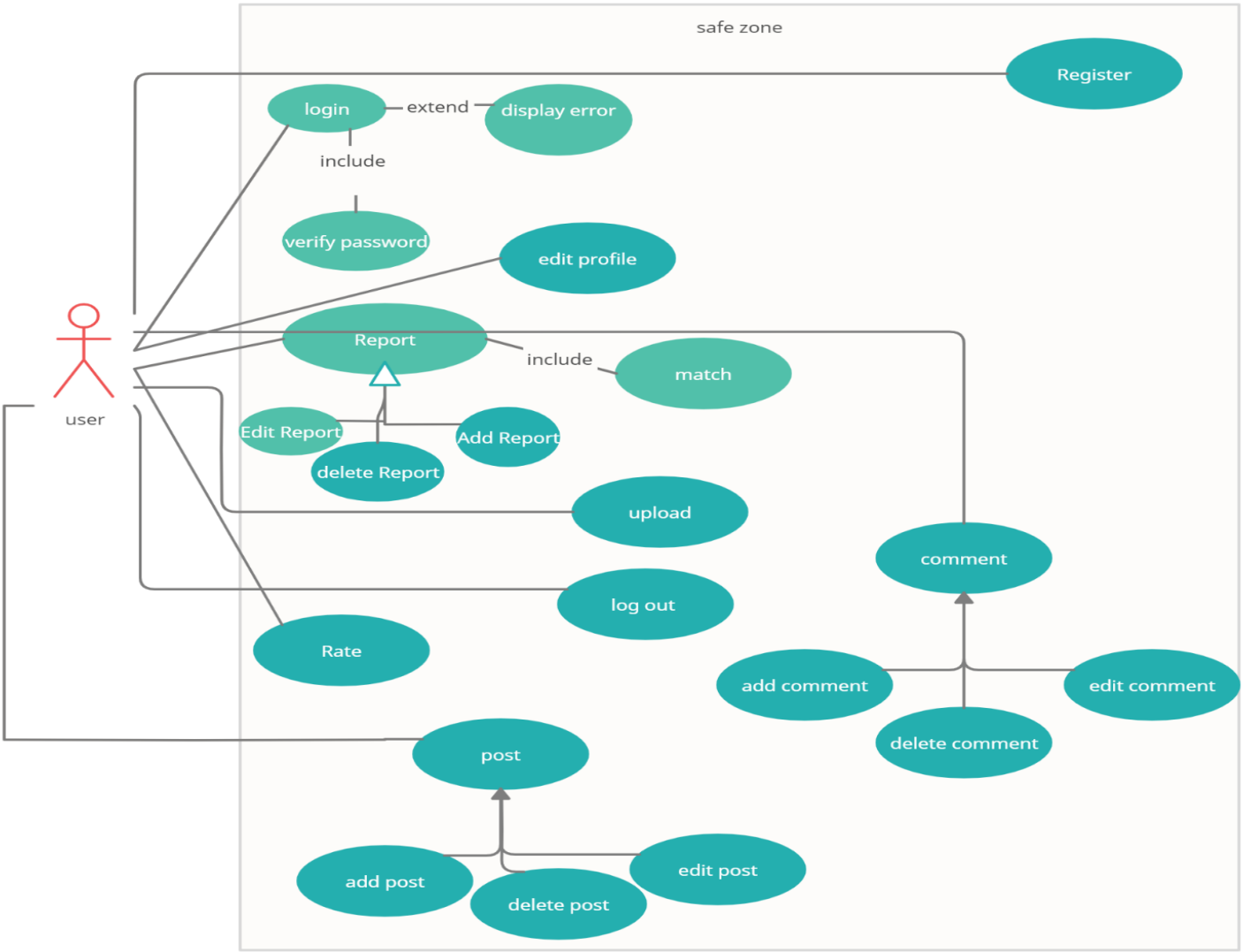


3.3 Software design

3.3.1 Class diagram



3.3.2 Use Case diagram



Use Case Diagram description

Identifier and Name:	safe zone
Initiator:	user
Goals:	help people to find their lost things, and find the safe ways
Precondition:	register in the application
Postcondition:	user may get what he lost

1-use case name: login

Precondition: user has registered, enter e-mail and password

Postcondition: verify password

Actor: user

Main success scenario: log into the application

Main unsuccess scenario: display error

2-use case name: register

Precondition: enter his data with unique phone number and e-mail

Postcondition: send data to the database

Actor: user

Main success scenario: create an user account

Main unsuccess scenario: display error for incorrect date

3-use case name: edit profile

Precondition: user has an account

Postcondition: change users' data

Actor: user

Main success scenario: update profile correctly

Main unsuccess scenario: display error for incorrect data

4-use case name: add Report

Precondition: user has an account

Postcondition: the system matches the report with other reports

Actor: user

Main success scenario: the system starts a matching with another reports

Main unsuccess scenario: The report has not been added

5-use case name: delete report

Precondition: user uploaded report before

Postcondition: report deleted and cannot match with other reports

Actor: user

Main success scenario: the report deleted successfully from the application

Main unsuccess scenario: the report still appears for the user

6-use case name: edit Report

Precondition: user uploaded report before

Postcondition: refresh the page

Actor: user

Main success scenario: the report edits successfully

Main unsuccess scenario: the report still appears as before edit

7-use case name: upload

Precondition: user add a report

Postcondition: add a photo to the report

Actor: user

Main success scenario: upload photo correctly

Main unsuccess scenario: The photo was not uploaded due to the format or image size

8-use case name: logout

Precondition: user logged in

Postcondition: back to log in page

Actor: user

Main success scenario: logged out from the application

Main unsuccess scenario: the system still running with the same profile

9-use case name: Rate

Precondition: user enter his location

Postcondition: calculate the average of safety for the street

Actor: user

Main success scenario: add the users rate to the street

Main unsuccess scenario: Wrong average safety calculation from the system

10-use case name: add post

Precondition: user logged in

Postcondition: post into the application

Actor: user

Main success scenario: add the post and other users watch it

Main unsuccess scenario: the post didn't upload

11-use case name: edit post

Precondition: user added post before

Postcondition: the post shows up with new editing

Actor: user

Main success scenario: edit the post and other users watch it

Main unsuccess scenario: the comment still shows up as its before editing

12-use case name: delete post

Precondition: user added post before

Postcondition: Post and accompanying comments will be deleted

Actor: user

Main success scenario: delete the post successfully

Main unsuccess scenario: the post still shows up for other users

13-use case name: add comment

Precondition: user watched a post or add post

Postcondition: the comment appears in the post

Actor: user

Main success scenario: add the comment into the post

Main unsuccess scenario: the comment didn't appear in the post

14-use case name: edit comment

Precondition: user add a comment before

Postcondition: the comment appears in the post

Actor: user

Main success scenario: edit the comment and every one see the new comment

Main unsuccess scenario: the comment still shows up as its before editing

15-use case name: Delete comment

Precondition: user added a comment before

Postcondition: the comment must disappear from the post

Actor: user

Main success scenario: delete the comm successfully

Main unsuccess scenario: the comment didn't appear in the post

Extensions:

1 a- use case name: verify password

Precondition: user enter password and e-mail

Postcondition: display an error message or login to home page

Actor: system

Main success scenario: determine inputs are valid or not

Main unsuccess scenario: user login with wrong data

1 b- use case name: Display Error

Precondition: user entered wrong password or e-mail

Postcondition: user receive notification to login again with valid data

Actor: system

Main success scenario: display notification to login again with valid inputs and user stay in login page

Main unsuccess scenario: user transfer to home page, or stay at login page with no notification for the reason

4 a- use case name: Match

Precondition: user add post

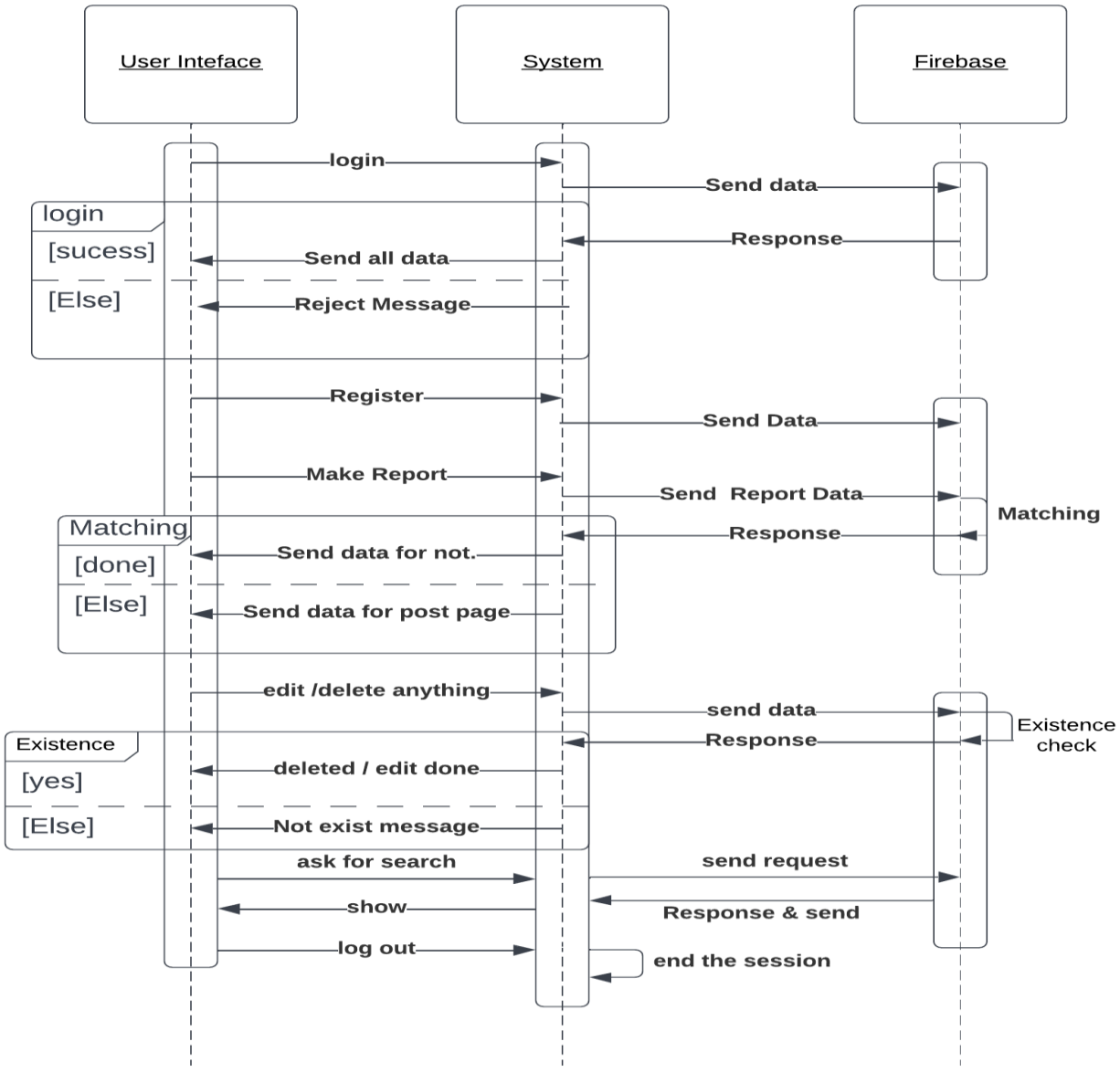
Postcondition: user receive notification to, and Similar reports appear in notification page

Actor: system

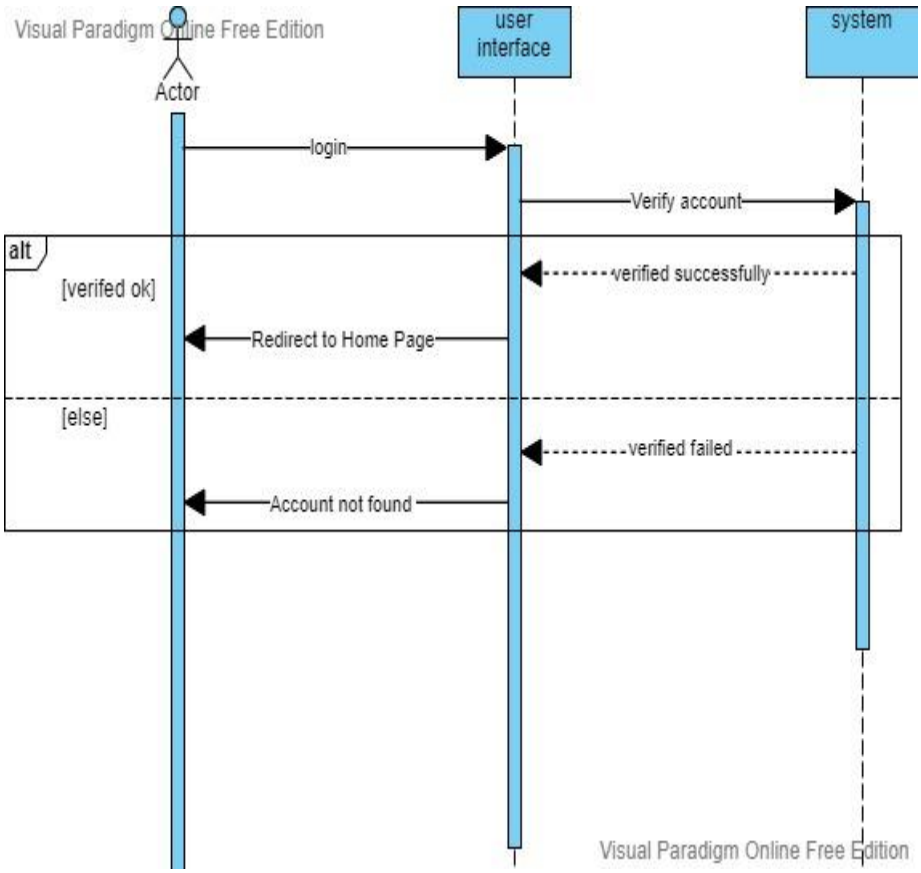
Main success scenario: find reports like the lost item, or send notification item not found

Main unsuccess scenario: cannot find the correct report

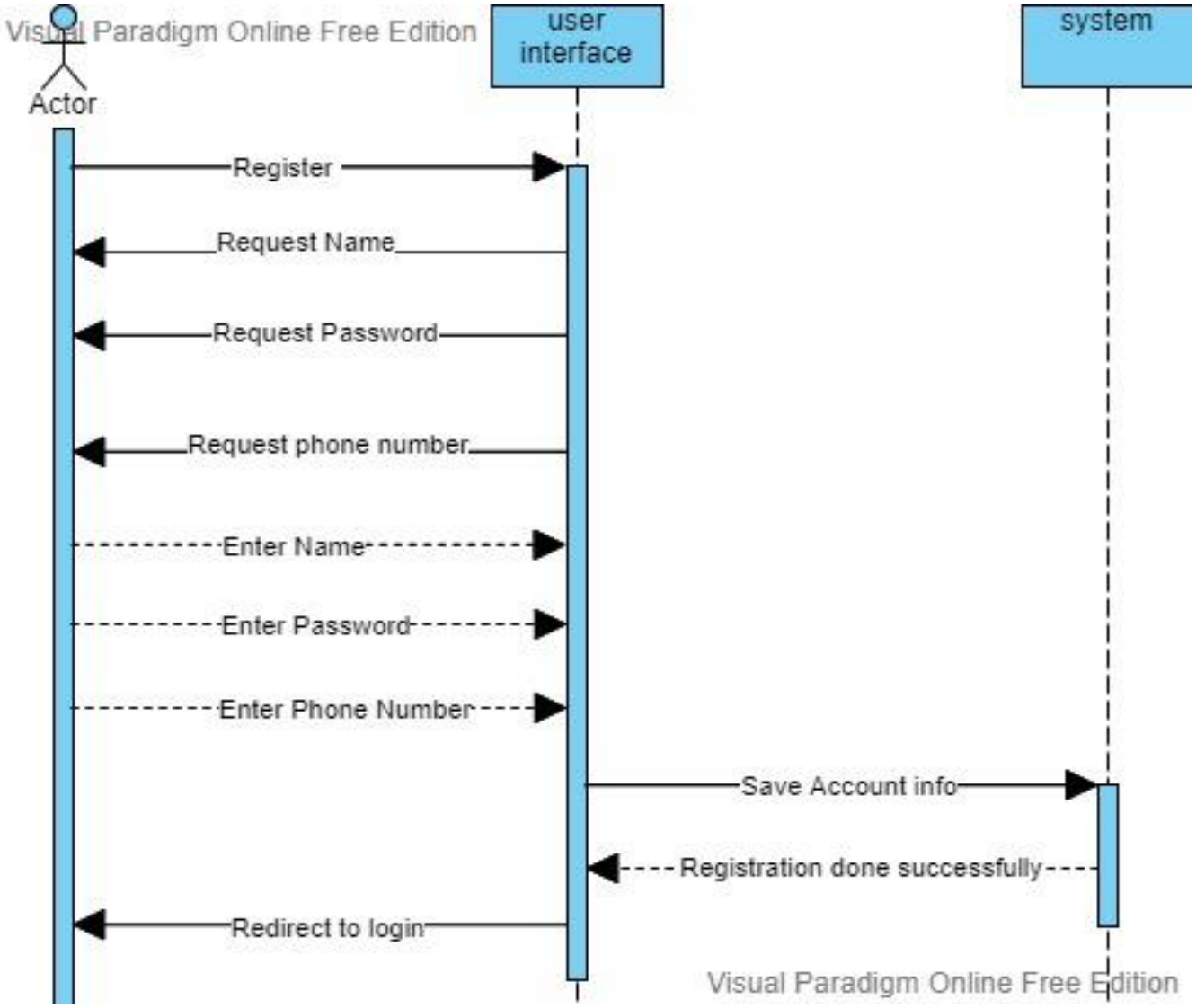
3.3.3 Sequence diagram



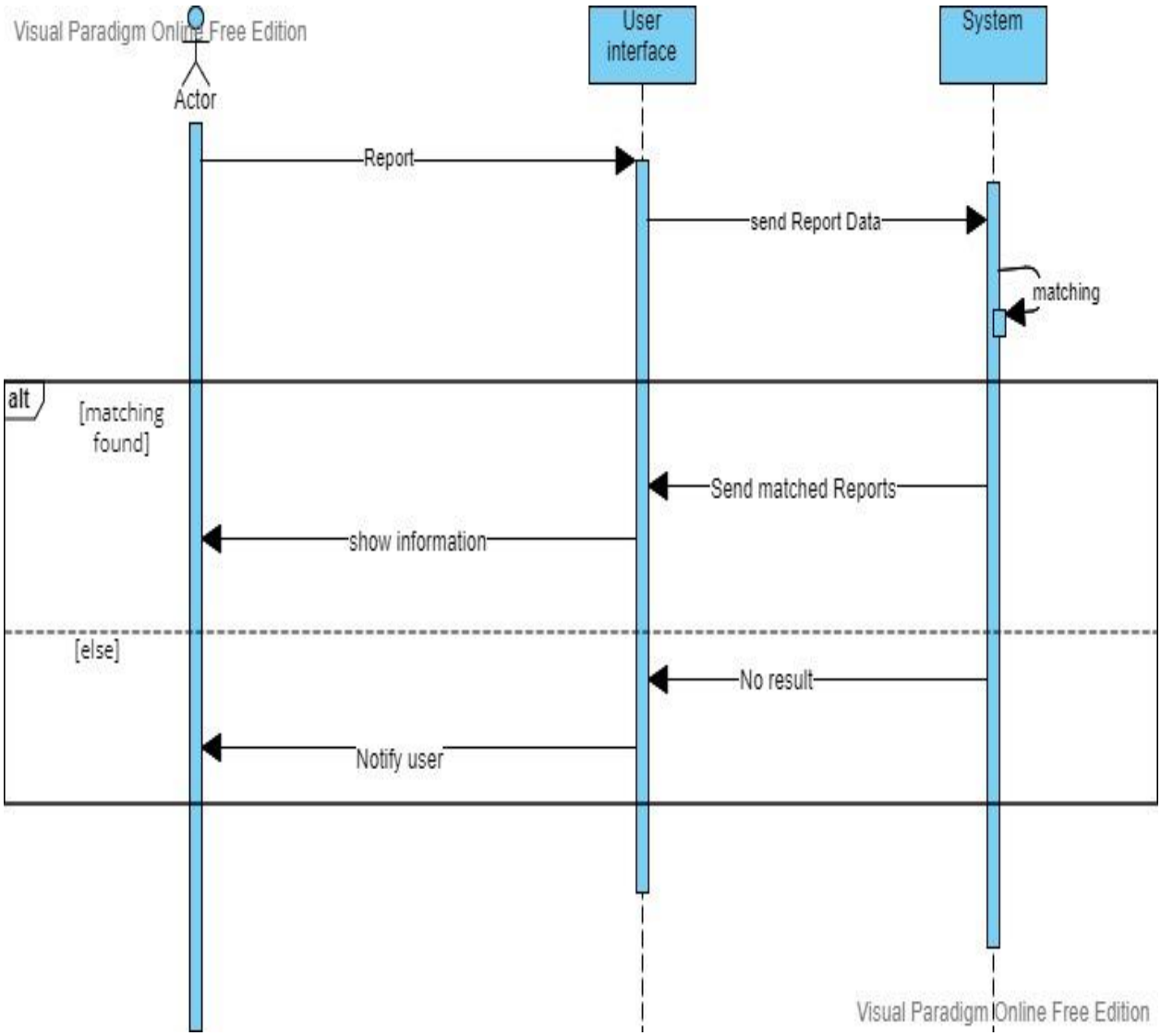
Login:


Visual Paradigm Online Free Edition
Actor
user interface
system
login
Verify account
alt
[verifed ok]
verified successfully
Redirect to Home Page
[else]
verified failed
Account not found
Visual Paradigm Online Free Edition

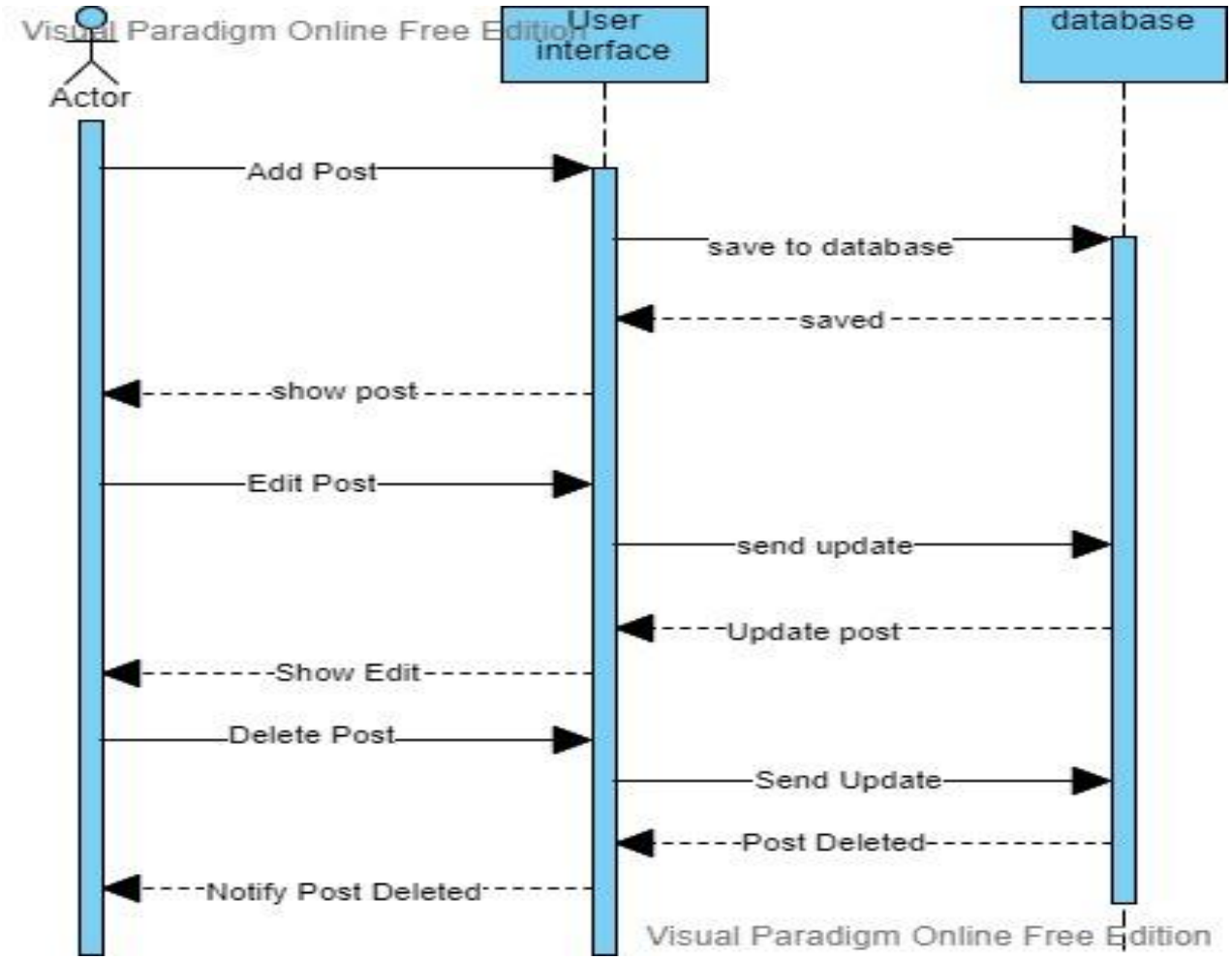
Register:


Visual Paradigm Online Free Edition
Actor
user interface
system
Register
Request Name
Request Password
Request phone number
Enter Name
Enter Password
Enter Phone Number
Save Account info
Registration done successfully
Redirect to login
Visual Paradigm Online Free Edition

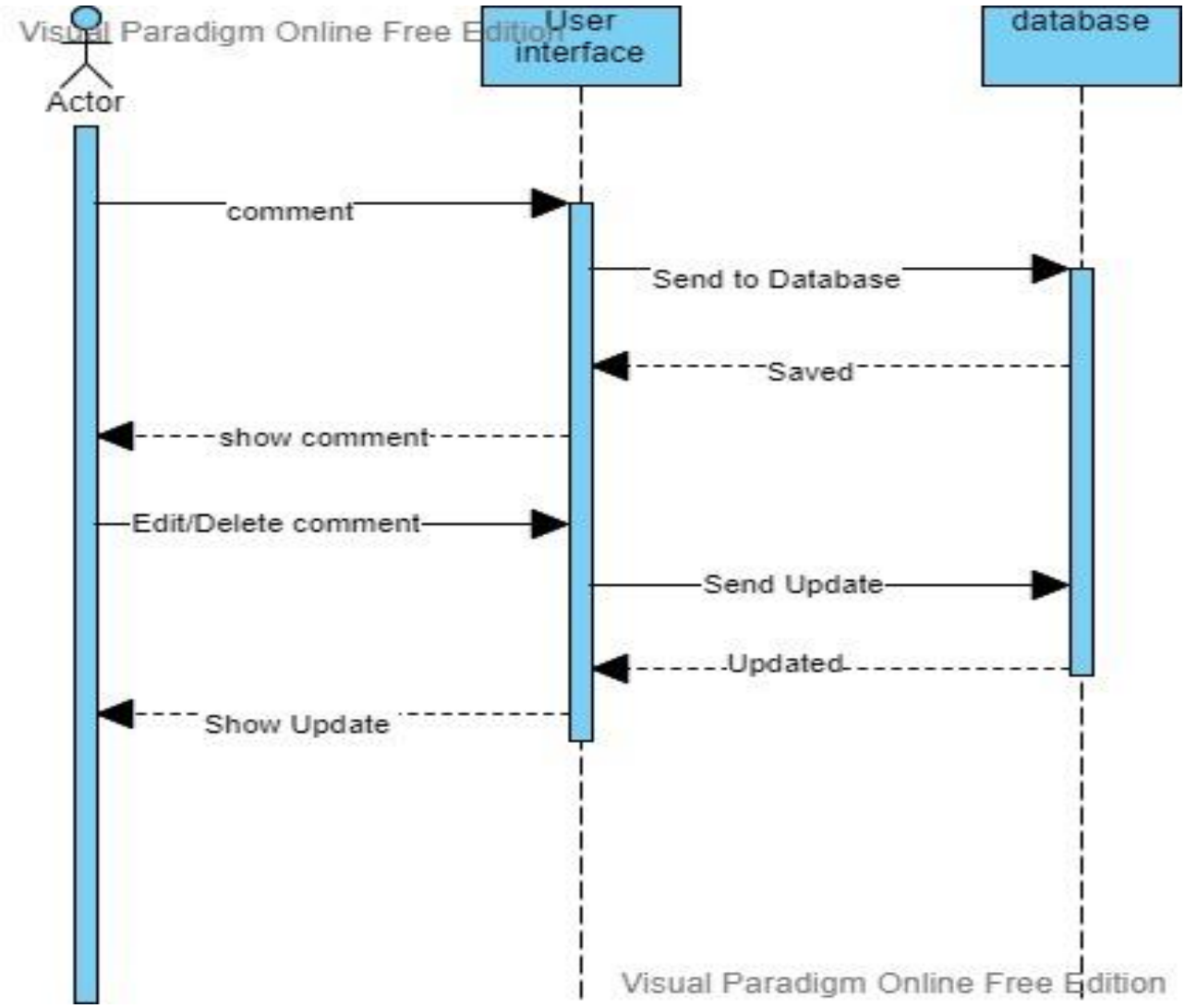
Report:


Visual Paradigm Online Free Edition
Actor
User interface
System
Report
send Report Data
matching
alt
[matching found]
Send matched Reports
show information
[else]
No result
Notify user
Visual Paradigm Online Free Edition

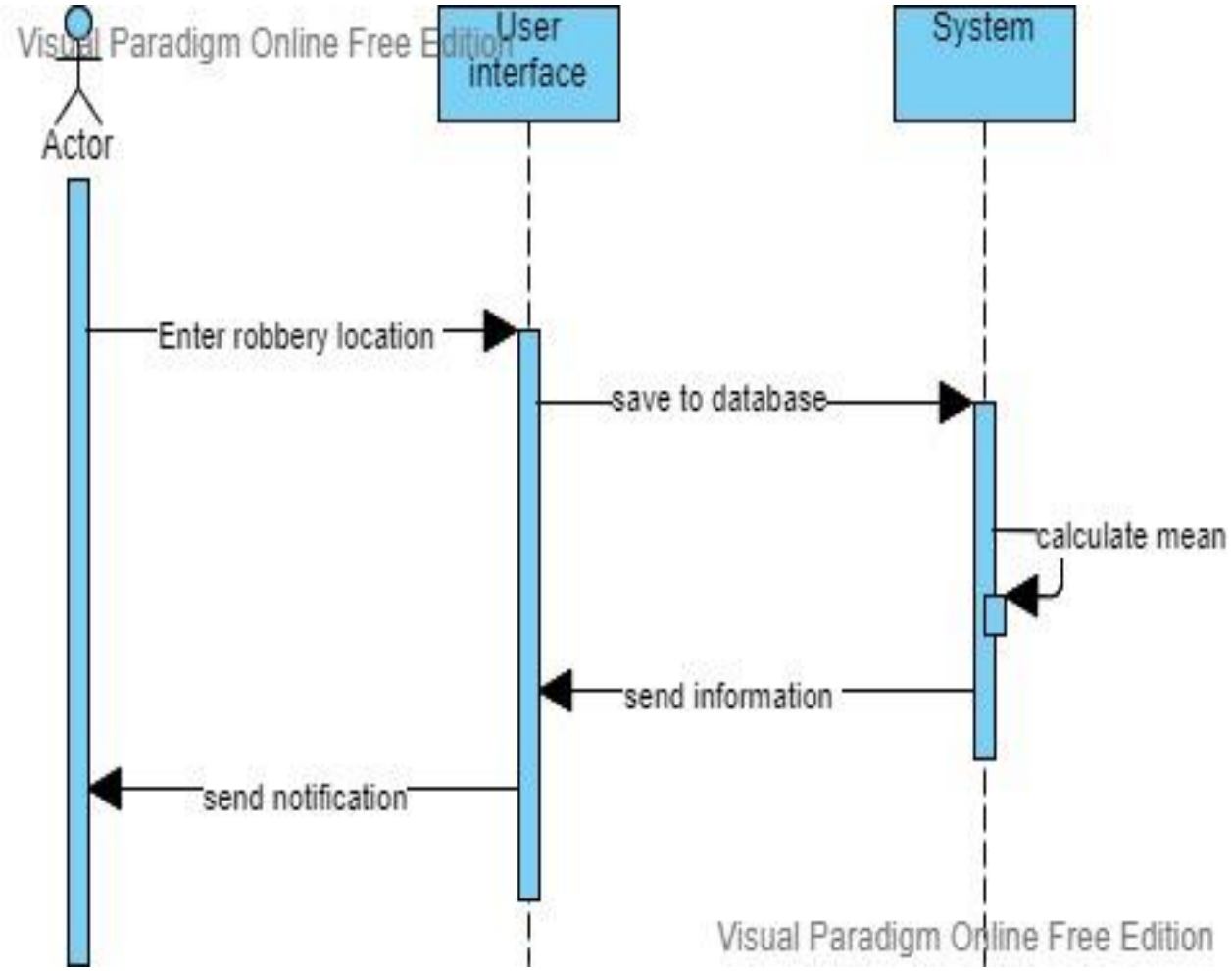
Post:


Visual Paradigm Online Free Edition
Actor
User interface
database
Add Post
save to database
saved
show post
Edit Post
send update
Update post
Show Edit
Delete Post
Send Update
Post Deleted
Notify Post Deleted
Visual Paradigm Online Free Edition

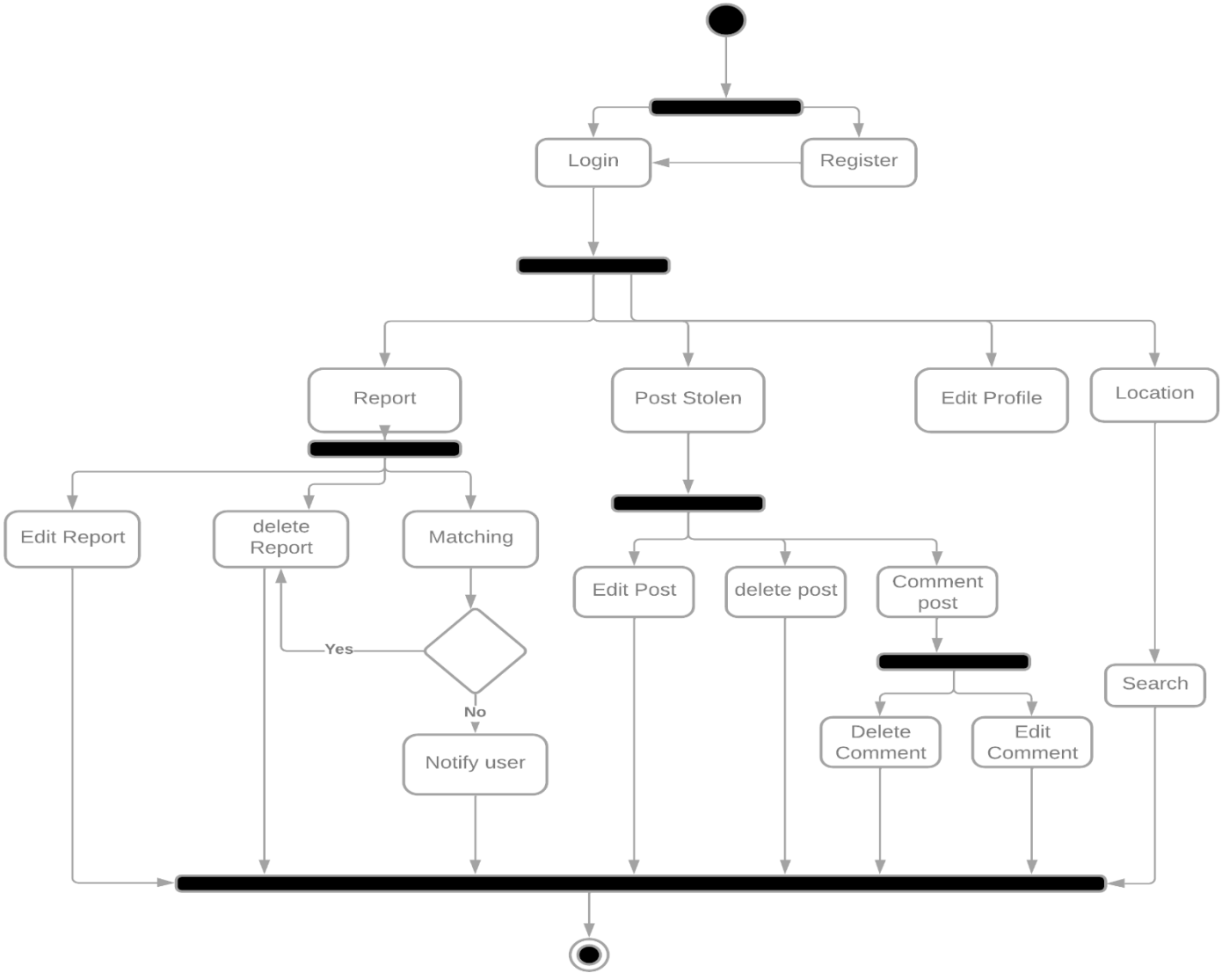
Comment:


Visual Paradigm Online Free Edition
Actor
User interface
database
comment
Send to Database
Saved
show comment
Edit/Delete comment
Send Update
Updated
Show Update
Visual Paradigm Online Free Edition

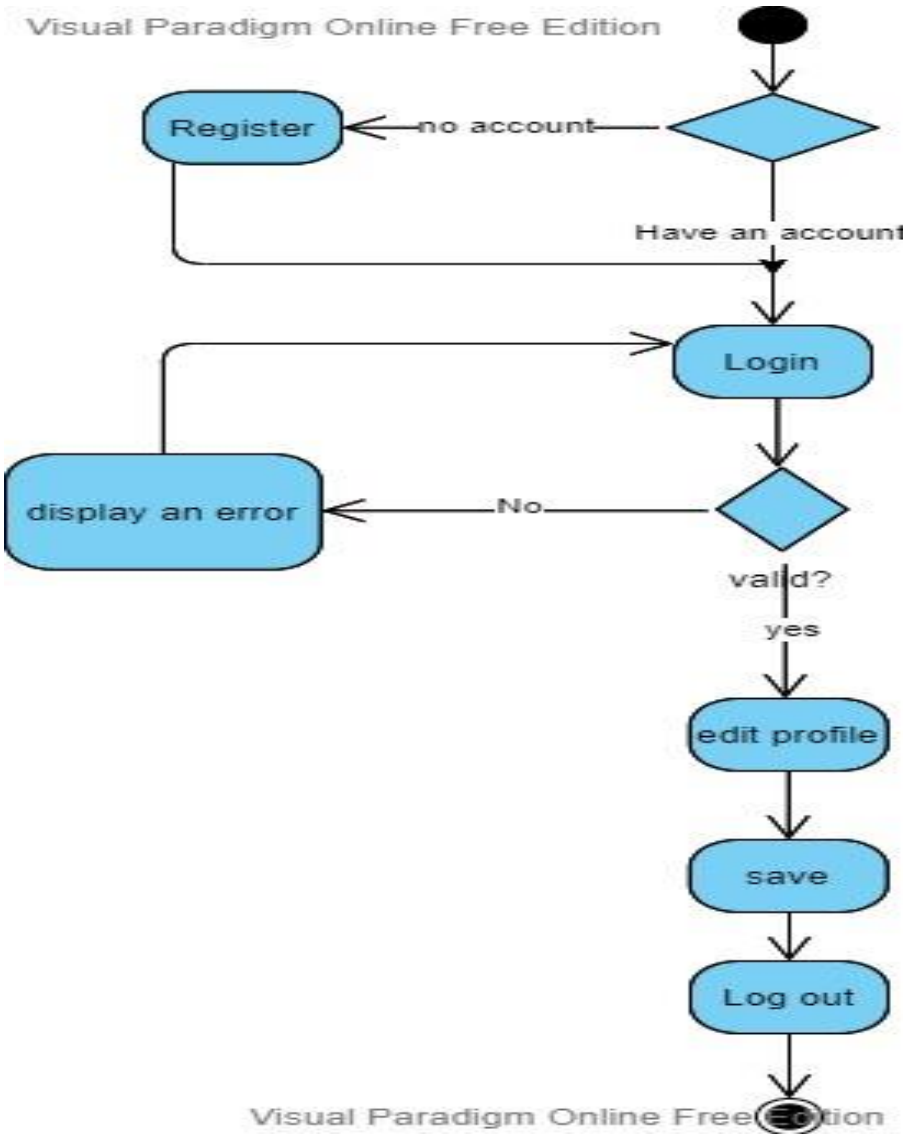
Safe zone:


Visual Paradigm Online Free Edition
Actor
User interface
System
Enter robbery location
save to database
calculate mean
send information
send notification
Visual Paradigm Online Free Edition

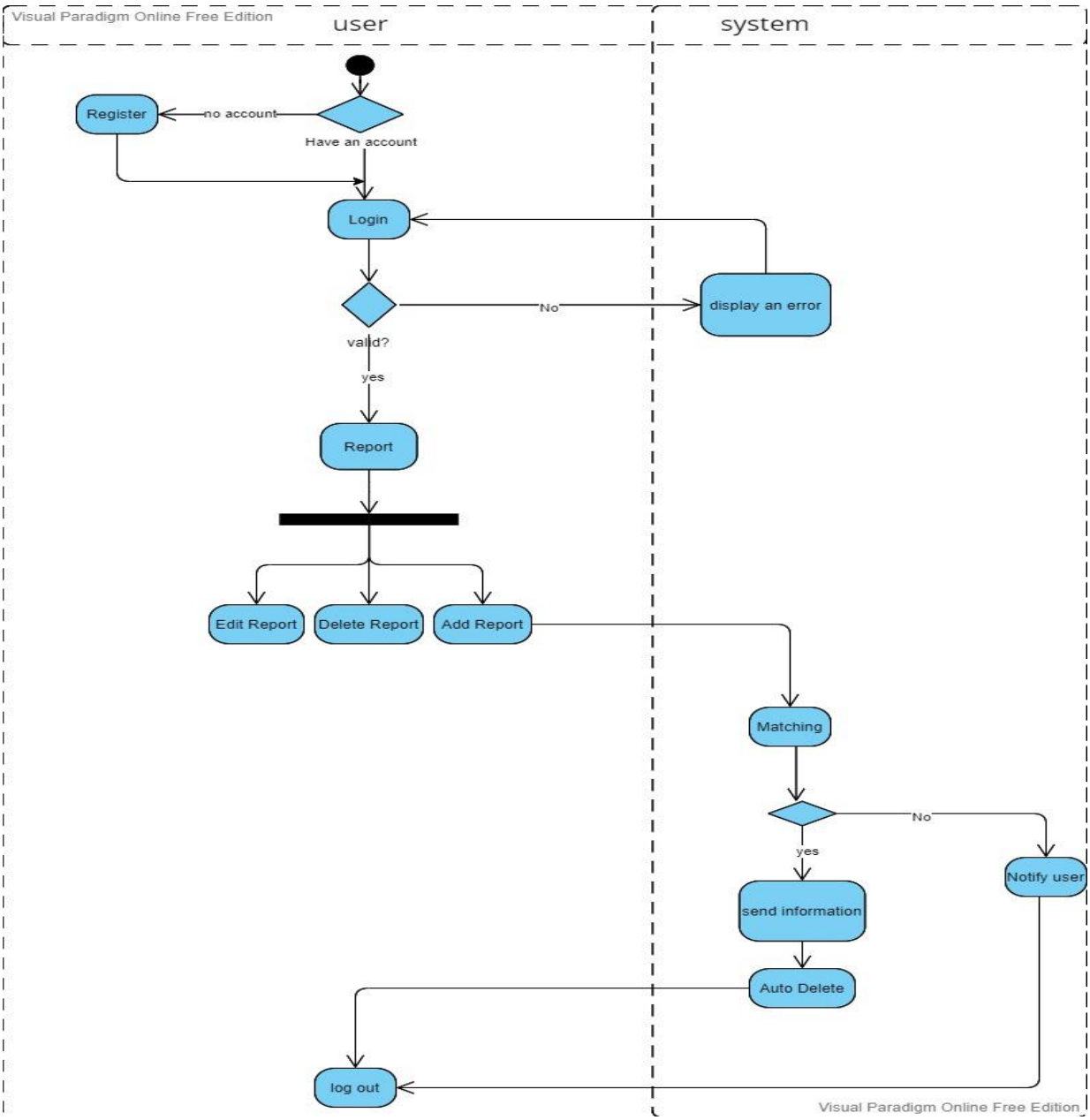
3.3.4 Activity Diagrams



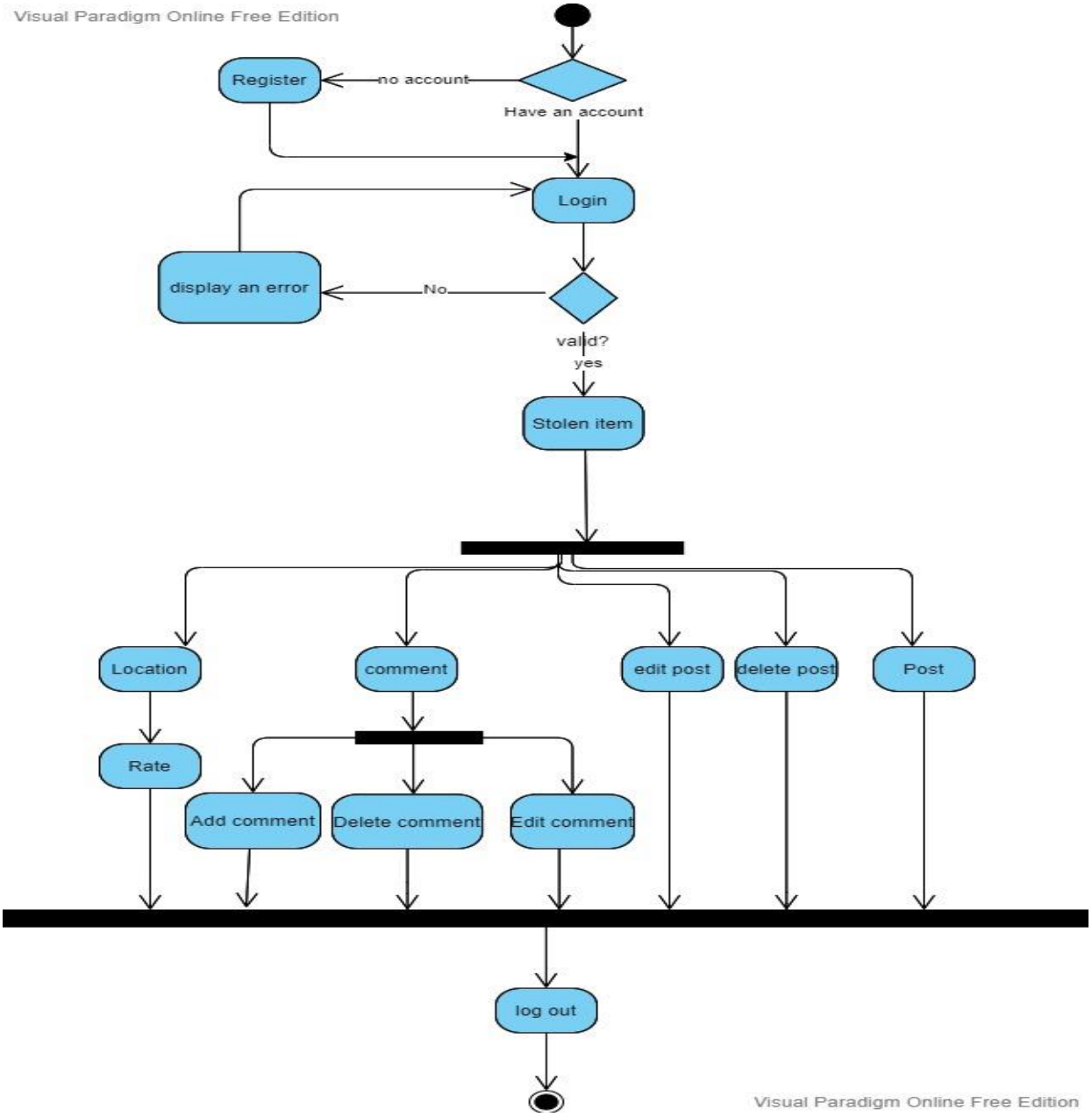
Login/register:



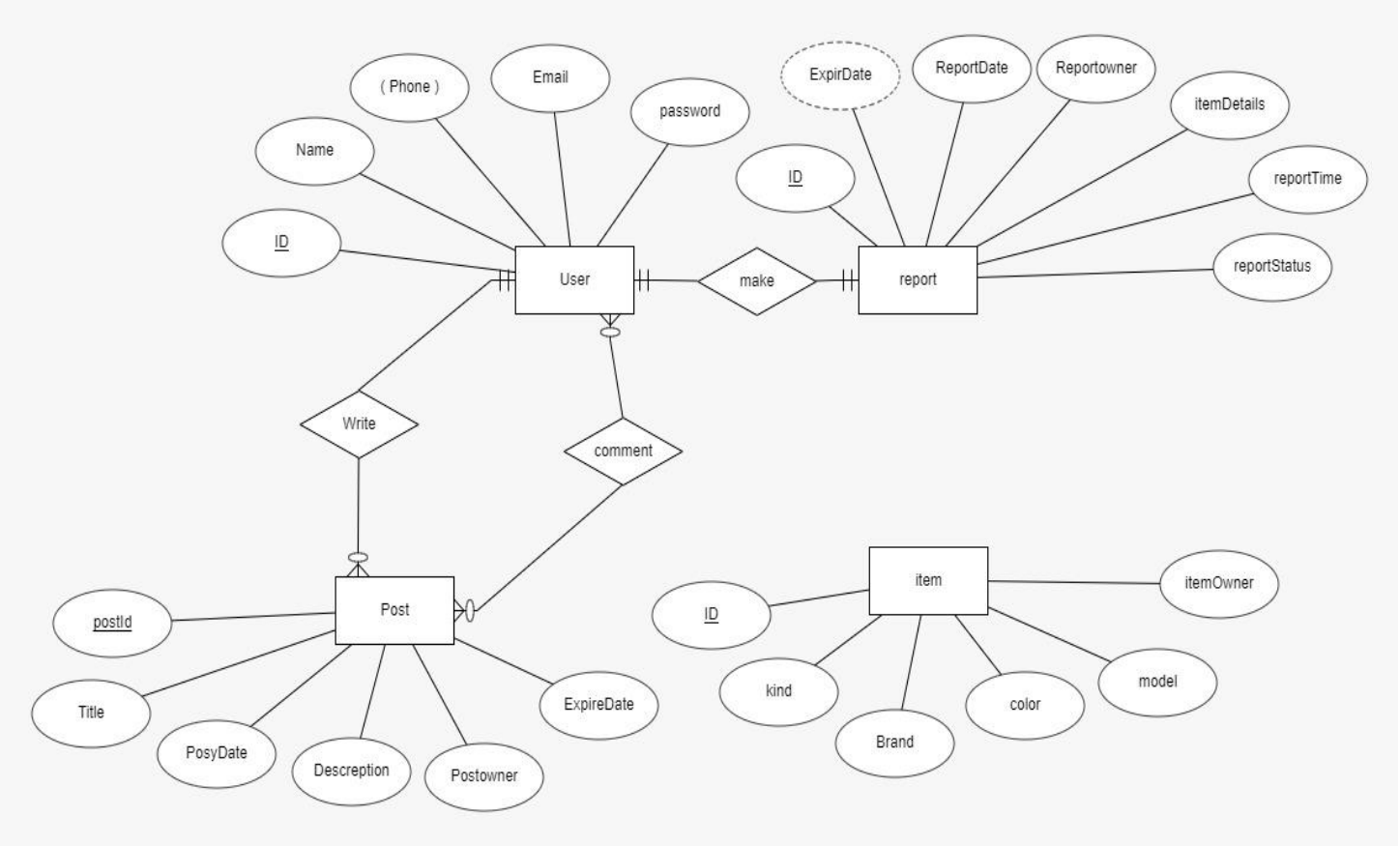
Report:


Visual Paradigm Online Free Edition
user
system
Register
no account
Have an account
Login
No
display an error
valid?
yes
Report
Edit Report
Delete Report
Add Report
Matching
No
yes
Notify user
send information
Auto Delete
log out
Visual Paradigm Online Free Edition

Stolen items:



3.3.5 ERD diagram:



Chapter four: Implementation

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

4. implementation

4.1 Software Architecture:

4.1.1 -Login:

```
void signIn(String email, String password) async {
  if (_formKey.currentState!.validate()) {
    try {
      await _auth
          .signInWithEmailAndPassword(email: email, password: password)
          .then((uid) => {
                Fluttertoast.showToast(
                    msg: "Login Successful",
                    webBgColor: "linear-gradient(to right, #2e8b57, #2e8b57)",
                    timeInSecForIosWeb: 2),
                Navigator.of(context).pushReplacement(
                    MaterialPageRoute(builder: (context) => HomeScreen())),
              });
    } on FirebaseAuthException catch (error) {
      switch (error.code) {
        case "invalid-email":
          errorMessage = "Your email address appears to be malformed.";

          break;
        case "wrong-password":
          errorMessage = "Your password is wrong.";
```

4.1.2-Registration:

```
void signUp(String email, String password) async {
  if (_formKey.currentState!.validate()) {
    try {
      await _auth
          .createUserWithEmailAndPassword(email: email, password: password)
          .then((value) => {postDetailsToFirestore()})
          .catchError((e) {
        Fluttertoast.showToast(msg: e!.message);
      });
    } on FirebaseAuthException catch (error) {
      switch (error.code) {
        case "invalid-email":
          errorMessage = "Your email address appears to be malformed.";
          break;
        case "wrong-password":
          errorMessage = "Your password is wrong.";
          break;
        case "user-not-found":
          errorMessage = "User with this email doesn't exist.";
          break;
        case "user-disabled":
          errorMessage = "User with this email has been disabled.";
          break;
        case "too-many-requests":
          errorMessage = "Too many requests";
          break;
        case "operation-not-allowed":
          errorMessage = "Signing in with Email and Password is not enabled.";
          break;
        default:
          errorMessage = "An undefined Error happened.";
      }
```

4.1.3-Add report as lost/found

```
matchingAlgorithem match = matchingAlgorithem();
postDetailsToFirestoreLost() async {
  if(formKey2.currentState!.validate()){
    FirebaseFirestore firebaseFirestore = FirebaseFirestore.instance;
    User? user = _auth.currentUser;

    // writing all the values
    lostReport.useremail = user!.email;
    lostReport.userid = user.uid;
    lostReport.username = loggedInUser.firstName;
    lostReport.userphone = loggedInUser.phone;
    lostReport.itemtype = selectedType + ' ' + CategorySelected;
    lostReport.itembrand = selectedBrand;
    lostReport.serialnumber = idNumber.text;
    lostReport.date = selectedDate.day.toString() +
        ' / ' +
        selectedDate.month.toString() +
        ' / ' +
        selectedDate.year.toString();
    lostReport.location = selectedregion + ' ' + selectedGovernorate;
    lostReport.itemcolor = selectedColor.toString();
    lostReport.status = false;
    lostReport.matchesLids;
    lostReport.Reportkind = 'lost';

    final collRef = await firebaseFirestore.collection('lostReports');
    var docReference = collRef.doc();
    lostReport.reportId = docReference.id;
    docReference.set(lostReport.toMap());
    Fluttertoast.showToast(
        msg: "Report added  successfully :) ",
        webBgColor: "linear-gradient(to right, #2e8b57, #2e8b57)",
        timeInSecForIosWeb: 5);
```

4.1.4-Matching:

```
Future<void> matchingFound(foundReportModel report) async {
 List<String?> ids = [];
 bool mainstatus = false;

 FirebaseFirestore.instance
     .collection('lostReports')
     .get()
     .then((QuerySnapshot querySnapshot) {
   querySnapshot.docs.forEach((doc) {
     if (doc["status"] == false &&
         doc["reportId"] != report.reportId &&
         doc["userid"] != user!.uid &&
         doc["itemtype"] == report.itemtype &&
         doc["itembrand"] == report.itembrand &&
         doc["serialnumber"] == report.serialnumber &&
         doc["location"] == report.location) {
       ids.add(doc["userid"]);
       mainstatus = true;
       //update all equals
       FirebaseFirestore.instance
           .collection('lostReports')
           .doc(doc["reportId"])
           .update({'status': true, 'matchesLids': report.userid});
     }
   });
   FirebaseFirestore.instance
       .collection('foundReports')
       .doc(report.reportId)
       .update({'status': mainstatus, 'matchesFids': ids});
 });
}
```

4.1.5- delete Report:

```
class deleteReportClass{
 static deleteReport(BuildContext context,String reportId ,String reportKind){
   if ( reportKind == 'lost'){
     FirebaseFirestore.instance.collection('lostReports').doc(reportId).delete();
     Fluttertoast.showToast(
         msg: "Report deleted  successfully :) ",
         webBgColor: "linear-gradient(to right, #2e8b57, #2e8b57)",
         timeInSecForIosWeb: 5);

   }else
     {
       FirebaseFirestore.instance.collection('foundReports').doc(reportId).delete();
       Fluttertoast.showToast(
           msg: "Report deleted  successfully :) ",
           webBgColor: "linear-gradient(to right, #2e8b57, #2e8b57)",
           timeInSecForIosWeb: 5);

     }


}
```

4.1.6-Add post:

```
postData(BuildContext context,var locationCity,var locationTown, String description, File? ir
  if (_formKey.currentState!.validate()) {
if (image != null){
  print('the image link is :$image');
  String fileName = basename(image.path);
  Reference firebaseStorageRef =
  FirebaseStorage.instance.ref().child('images/$fileName');
  await firebaseStorageRef.putFile(image);
  _url =await firebaseStorageRef.getDownloadURL();

  FirebaseFirestore firebaseFirestore = FirebaseFirestore.instance;
  postModel post = postModel();
  User? user = FirebaseAuth.instance.currentUser;
  DocumentSnapshot<Map<String, dynamic>> data =await firebaseFirestore
      .collection("users")
      .doc(user!.uid).get();
  //set values
  post.postOwner = data.get('firstName');
  post.postOwnerId=user.uid;
  post.postDescription = description;
  post.locationTown =locationTown;
  post.locationCity = locationCity;
  post.location=locationTown+' '+locationCity;
  post.postTime =DateTime.now();
  post.imageUrl=_url;
  post.comments =[];
  post.likes;

  final collRef = await firebaseFirestore.collection('posts');
  var docReference = collRef.doc();
  post.postId = docReference.id;
  docReference.set(post.toMap());
  Fluttertoast.showToast(
```

4.1.7- add comment:

```
addCommint(String postid,String postContent) async{
  if(_FieldKey.currentState!.validate()){

    //comment model call
    commentModel comment = commentModel();
    User? user = FirebaseAuth.instance.currentUser;
    comment.postId=postid;
    comment.commentContent=postContent;

    var snap= await FirebaseFirestore.instance.collection('users').doc(user!.uid).get();
    var ownerName =snap.get('firstName');
    comment.commentOwnerId =user.uid;
    comment.commentOwner=ownerName;
    final collRef = await FirebaseFirestore.instance.collection('comments');
    var docReference = collRef.doc();
    comment.commentId = docReference.id;
    docReference.set(comment.toMap());
    await FirebaseFirestore.instance.collection('posts').doc(postid).update({'comments': FieldValue.arrayU
  }
}
```

4.1.8- safe places:

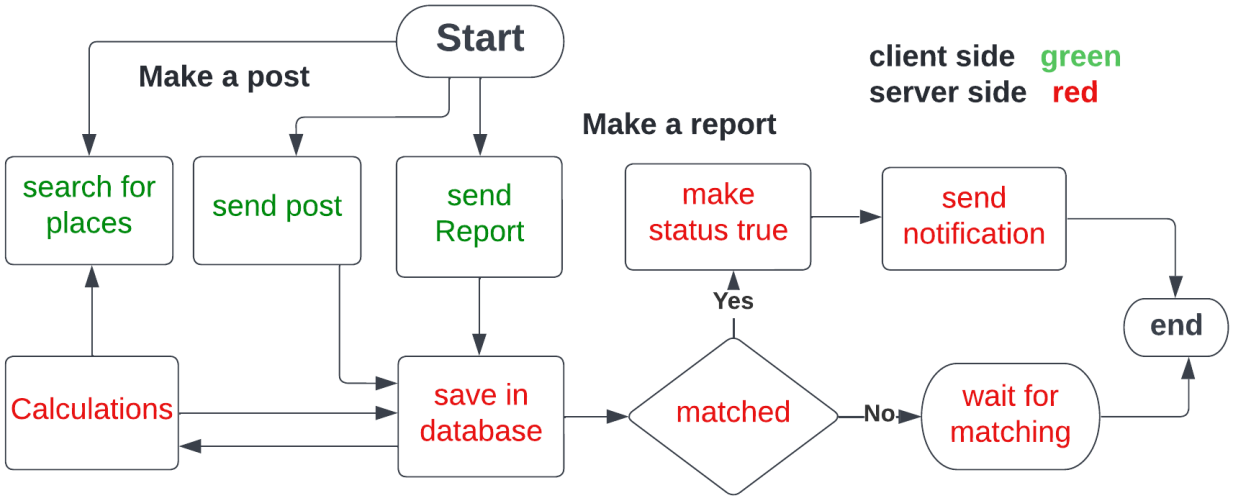
```
safePlaces(BuildContext context , var gov , var reg) async{
    int safety=0;
    QuerySnapshot<Map<String, dynamic>> snap;
    if(reg == null){
      snap = await FirebaseFirestore.instance.collection('posts').where('locationTown',isEqualTo:gov).get();
      safety=snap.size;
    }else{
      snap = await FirebaseFirestore.instance.collection('posts').where('location',isEqualTo:gov+' '+reg).get();
      safety=snap.size;
    }
    var all = await FirebaseFirestore.instance.collection('posts').get();
    double safeRatio = 0.0;
    safeRatio = (safety/all.size)*100;
    return safeRatio;
  }
}
```

4.1.9-log out:

```
Future<void> logout(BuildContext context) async {
  await FirebaseAuth.instance.signOut();
  Navigator.of(context).pushReplacement(
      MaterialPageRoute(builder: (context) => LoginScreen()));
}
```

4.2 Flow Chart

In this Figure we show the communication between client side and server side.



Chapter Five: Testing

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

5. Testing

5.1 Functional Testing:

5.1.1 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.

5.1.2 Integration testing:

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

5.1.3 System testing:

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

5.2 Non-Functional Testing:

5.2.1 Performance testing:

Accomplished a designated function regarding processing time and through put rate.

5.2.2 Load testing:

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

5.2.3 Security testing:

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

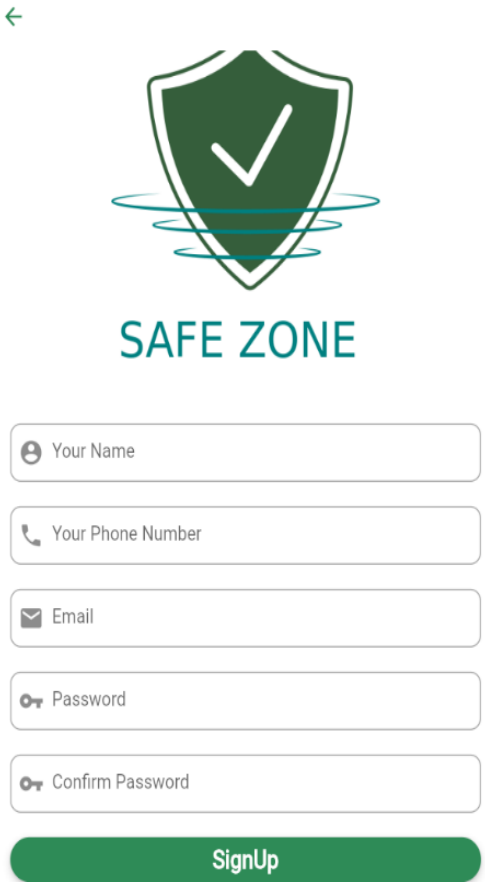
5.3 Additional testing

Project Name: safe zone

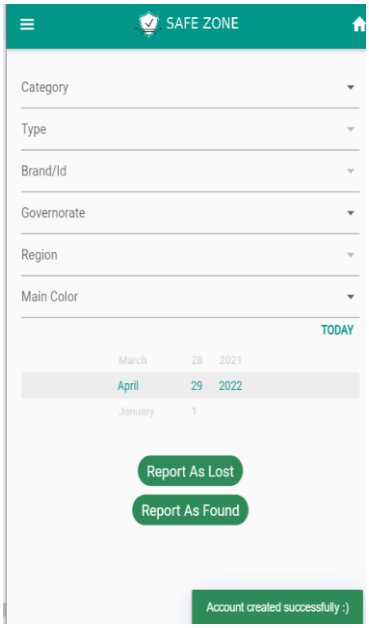
Module Name: safe zone Register

Test case id: Register_1

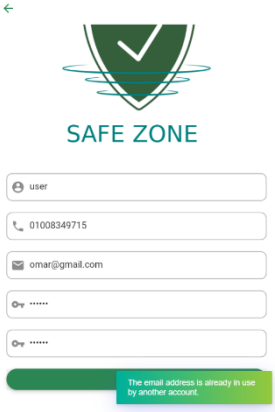
Test case scenario: try to registration with valid and invalid data

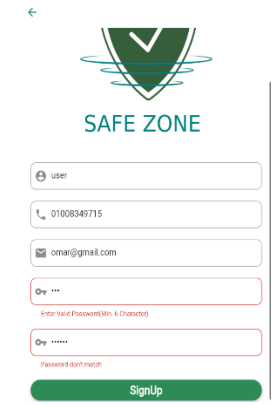


Test case	Test steps	Test data	Result
Register with complete and valid data	1-press sign up 2-enter data 3-press sign up	Personal data(name ,phone number, e-mail, password)	1-Account created successfully -logged in

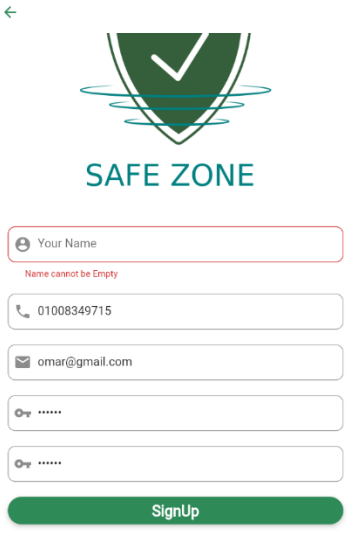


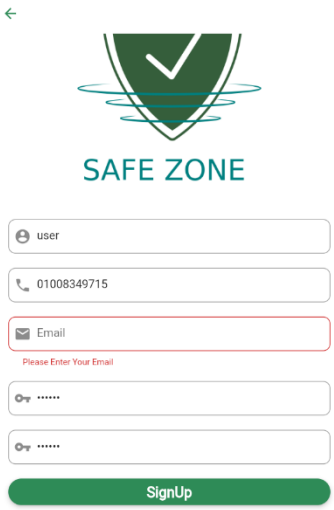
Test case	Test steps	Test data	Result
Register with complete and invalid data	1-press sign up 2-enter data 3-press sign up	Personal data(name ,phone number, e-mail(already exist) ,password(invalid))	1-display message that e-mail in use 2-display error message invalid password





Test case	Test steps	Test data	Result
Register with incomplete and valid data	1-press sign up 2-enter data 3-press sign up	Personal data(,phone number, e-mail, password)	Display error message to co complete empty fields





Test case	Test steps	Test data	Result
Register with incomplete and invalid data	1-press sign up 2-enter data 3-press sign up	Personal data(,phone number, e-mail, password(invalid))	1-Display an error message to co complete empty fields 2- Display an error message to enter valid password

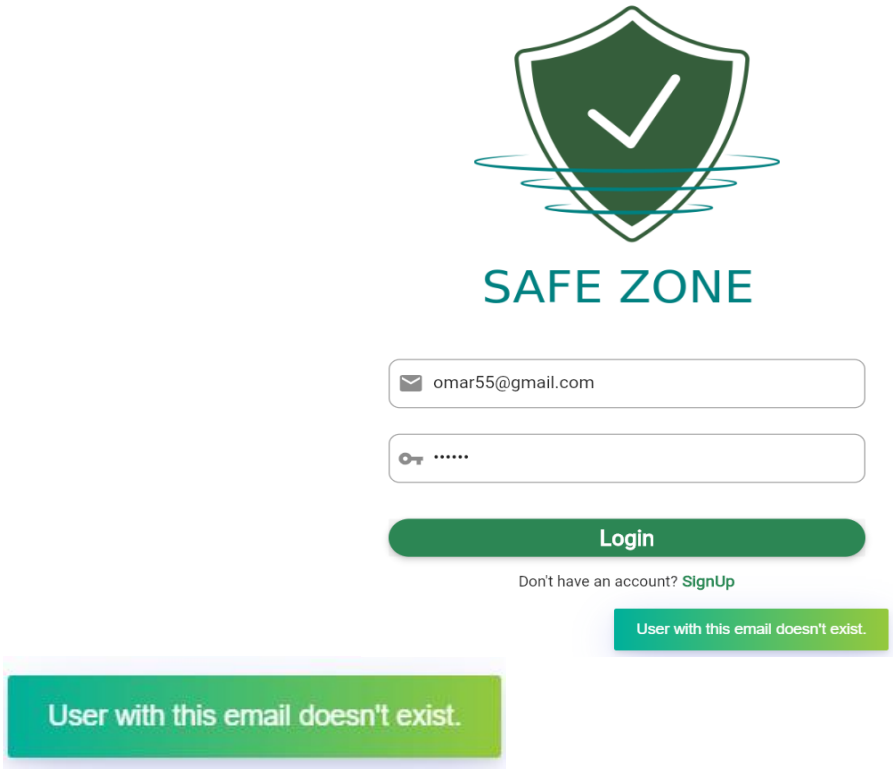
Module Name: safe zone login

Test case id: login_2

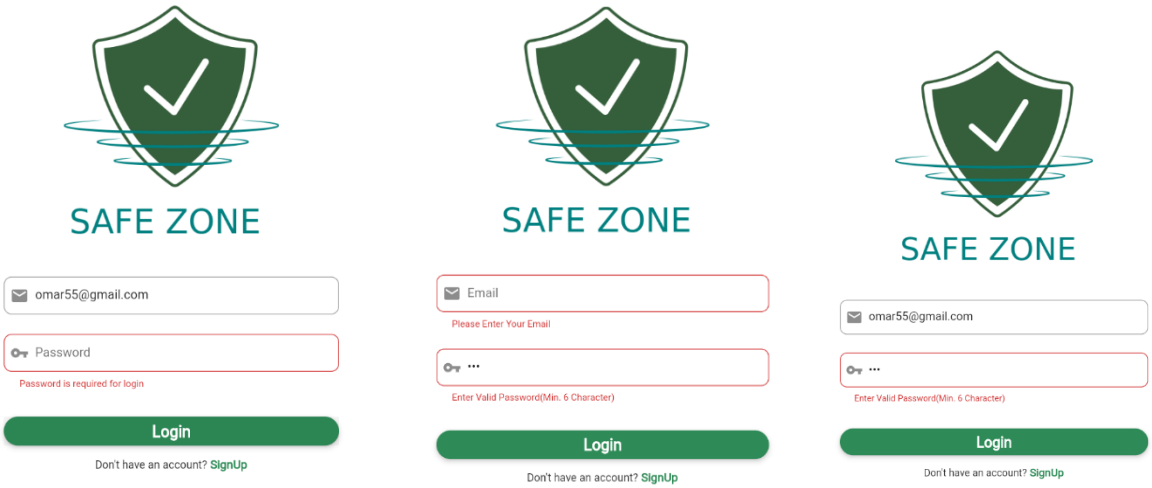
Test case scenario: try to login with valid and invalid email and password

Test case	Test steps	Test data	Result
login with valid email and valid password	1-enter email 2-enter password 3-press login	Valid Email Valid password	Logged successfully

Test case	Test steps	Test data	Result
login with invalid email and valid password	1-enter email 2-enter password 3-press login	Invalid email Valid password	Display message email doesn't exist



Test case	Test steps	Test data	Result
login with valid email and invalid password	1-enter email 2-enter password 3-press login	valid email invalid password	Display error message to enter valid password



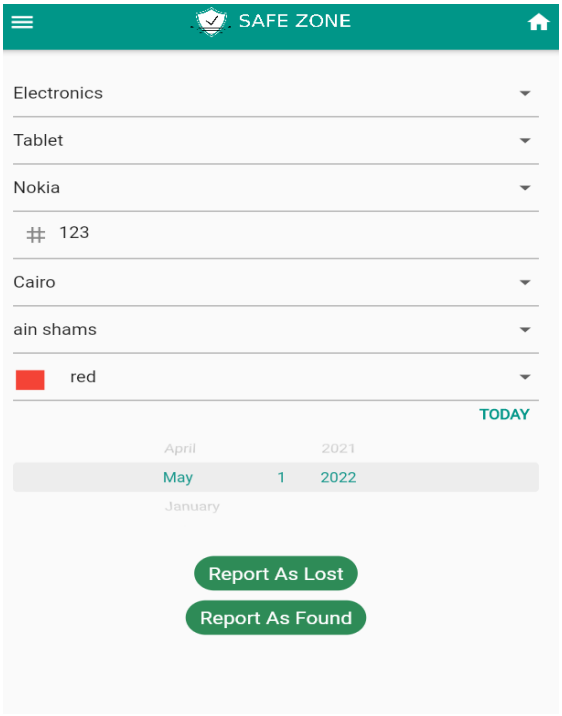
Test case	Test steps	Test data	Result
login with invalid email and invalid password	1-enter email 2-enter password 3-press login	invalid email invalid password	Display error message to enter valid email and password

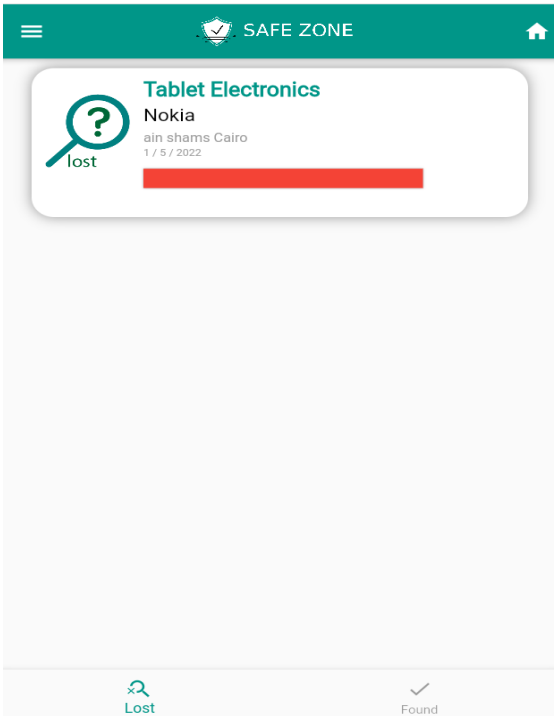
Module Name: safe zone Report

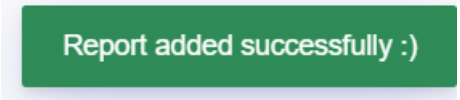
Test case id: Report_1

Test case scenario: try to Report as lost

Test case	Test steps	Test data	Result
Report about Item as a lost item	1- enter item details 2-press Report as lost	Categories Type brand/id governorate region color	1-display message Report added successfully 2-Report added to my Reports and wait for matching





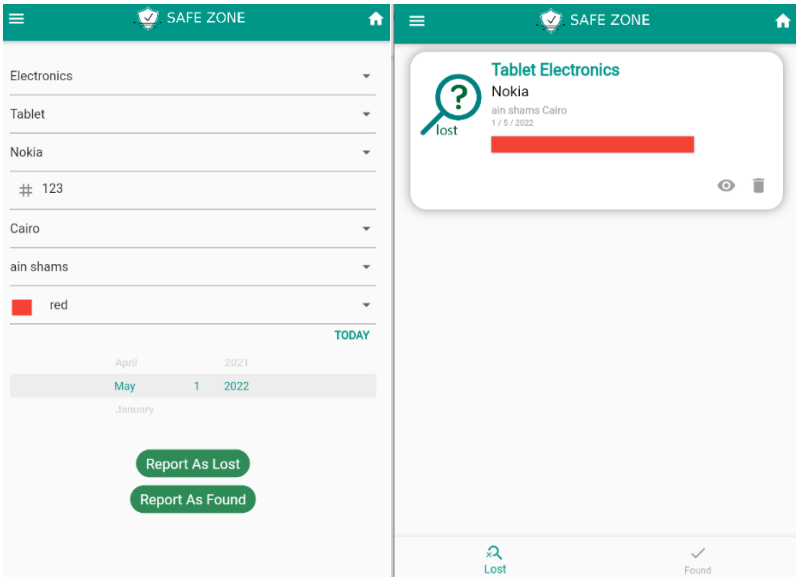


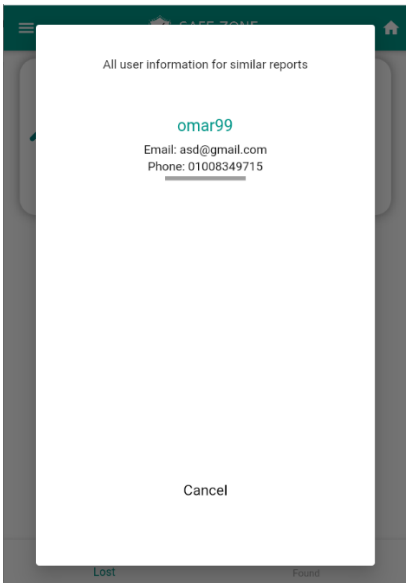
Module Name: safe zone Report

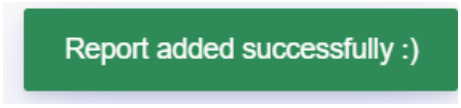
Test case id: Report_2

Test case scenario: try to Report as found

Test case	Test steps	Test data	Result
Report about Item as a found item	1- enter item details 2-press Report as found	Categories Type brand/id governorate region color	1-display message Report added successfully 2-send notification to the owner of the lost item







Module Name: safe zone Report

Test case id: matching_1

Test case scenario: test the matching between reports

Test case	Test steps	Test data	Result
Matching between lost and found report	1- open the owner account add report ass lost 2-open the other user account report as found	e-mail password Categories Type brand/id governorate region color	1-display message Report added successfully 2-send notification to the owner of the lost item

Module Name: safe zone Report

Test case id: posts_1

Test case scenario: try to add post

Test case	Test steps	Test data	Result
Add post with image	1- choose zone 2-write the text in text area 3-choose image 4-click Post	Zone Text image	The post shows up for other users

Module Name: safe zone Report

Test case id: posts_2

Test case scenario: try to comment on post

Test case	Test steps	Test data	Result
Comment on posts	1- click on the post 2-write the comment in text area 3-click Post	Text	The comment show up for other users

Module Name: safe zone Report

Test case id: posts_3

Test case scenario: try to delete comment on post

Test case	Test steps	Test data	Result
Delete Comment on posts	1- click on the post 2- determine the comment 3- click at the x Sign 4-click yes	Text	a flash message show up to make sure of deleting

Module Name: safe zone Report

Test case id: posts_4

Test case scenario: try to delete post

Test case	Test steps	Test data	Result
Delete posts	1- determine the post 2- click at the x Sign 3-click yes	Text	a flash message show up to make sure of deleting

Module Name: safe zone Report

Test case id: safe places_1

Test case scenario: try the rate of safety at different places

Test case	Test steps	Test data	Result
Safe places rate	select place	List of places	The rate show up

Module Name: safe zone Report

Test case id: logout_1

Test case scenario: try to sign out

Test case	Test steps	Test data	Result
Sign out	1-click on the main menu 2-click sign out		Back to login page

Chapter Six:

Results and Discussion

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

6. Results and Discussion

6.1 Results

6.1.1 Expected Result:

- the app help people to retrieve their lost item.
- the user gives a detailed report about lost/found item.
- the application help people to report in confidentiality and safety.
- the application provides a social communication between users.
- users determine the range of safety of each zone.
- application send notification if user entered unsafe zone.

6.1.2 Actual Result:

- the user gives a detailed report about lost/found item.

the app help people to retrieve their lost item.

- the user gives a detailed report about lost/found item.
- the application help people to report in confidentiality and safety.
- user can add posts and comment on another users' posts.
- user determine the unsafety zone on his post.
- the system shows the range of safety for each zone.

6.2 Discussion

We managed to get the same expected result except sending notification for the user if he exists in an unsafe zone and changed some features as the way of determining safety of zones.

Chapter seven:

Conclusion

Finally, here we are writing the very last words and putting the last lines of our story and adventure at FCAI-HU, working in this project was really different from any other project we worked on during the last 4 years, this project was full of the feeling of responsibility towards our society and ourselves, that why we tries to do our best in it, regardless the poor resources we had and the limited time.

We consider this project as a thanks to all our professors and teacher assistance and to our country as well, trying to solve the social problems through what we learned and know was a real challenge, a cool challenge that was worth trying.

We hope that safe Zone will play its role in society as we are expecting.

We are looking forward to being the first local solution that encourages people to help others to retrieve their lost things and to increase the awareness how to protect themselves from all crimes types.

We wish that we had more resources to publish Safe Zone with the highest quality possible, and hopefully we are going to expand this application and enhance its performance and make it popular.

Finally, we hope that everyone realizes how “few clicks '' can affect society and make a huge difference to people’s life, and start to pay more attention to the technology and try to use it in solving our daily big and small problems.

Chapter eight:

Future Work

8. Future work

1. instead of showing all the posts in the home page we will restrict those post so the user can mainly see his/her friend's posts then the other posts will be shown at the bottom of the page as the user scroll. We will add that by giving the user an (add-friend) feature so his home page will be like any social media app where he can first see his/her friend's posts

2. we will add a chat bot to answer the most common questions about the app or a guideline into the app

3. We plan to launch the beta version for close people and friends to test the project at the advanced operating level to fix any errors, and then launch the project on all platforms after completing any repairs.

4. At the advanced level of the project, artificial intelligence can be added to the project with many goals that are scheduled to develop the application in a way. For example, you can add nature language processing which will help in expanding the categories and locations which users use in reports. also, the operation of monitor the posts for any unacceptable work.

Bibliography

https://www.techtarget.com/searchsoftwarequality/answer/What-are-requirements-types

https://en.wikipedia.org/wiki/Governorates_of_Egypt

https://troov.com/

https://docs.flutter.dev/deployment/android

https://firebase.google.com/

https://www.lucidchart.com/pages/

https://bigwater.consulting/2019/04/08/software-development-life-cycle-sdlc/

SAFE ZONE

LOST&FOUND APPLICATION

THANKS