



MISSING PERSON (Ukraine Finder)

Software Engineering for Automation (SE4A)

ABSTRACT

Requirement, design, implementation, and testing of a website with the goal to make efficient the searching for a missing person

TEAM MEMBERS (ARS)

- Gabriele Ribolla
10617369

PROFESSOR

Prof.

Table of Contents

TABLE OF CONTENTS	1
CHAPTER 1 : FEASIBILITY AND STUDY DOCUMENT.....	1
➤ Introduction	1
➤ Literature Survey	1
➤ Problem Statement	1
➤ Existing solutions.....	5
Telegram channel	5
National bureau	6
Police Website.....	7
Instagram/Facebook	7
➤ Proposed Solution	8
Ukraine Finder	8
Type of users	8
Objectives.....	8
➤ Business model.....	10
Business Canvas	10
Marketing Strategy.....	11
➤ Assumptions and Constraints	13
Assumptions.....	13
Constraints	13
➤ Tools and Technologies	13
CHAPTER 2 : REQUIREMENTS ANALYSIS AND SPECIFICATION DOCUMENT.....	14
➤ Requirements given by form's responses	14
➤ Stake Holders	15
➤ Requirements Analysis	15
Description of the system	15
User Functionalities.....	16
Admin Functionalities.....	18
➤ Non-functional Requirements	19
Usability.....	19
Reliability.....	19
Performance.....	19
Interface	19
Implementation.....	19
Security.....	19

Installation.....	19
➤ Use Case Descriptions (Scenarios)	20
Sign Up	20
Sign In	21
Sign Out	22
Update Username	23
Update Password	24
Add Post	25
View Post.....	26
Delete Post	27
Edit Post	28
Search User	29
Contact Details	30
Make a Call	31
View Notification.....	32
Add Location	33
Admin Sign In	34
Admin Update Password	35
Admin View Post	36
Admin Delete Post.....	37
Admin Search User	38
Admin Delete User	39
Search Missing Person.....	40
➤ Activity diagrams.....	41
Sign up (Use Case ID 001)	41
Sign in (Use Case ID 002)	41
Sign Out (Use Case ID 003)	42
Add Post (Use Case ID 007)	42
Make a Call (Use Case ID 014)	43
View post (Use Case ID 008).....	43
Edit profile (Use case ID 004/006/013)	44
Delete Post (Use Case ID 009)	44
Admin sign in (Use Case ID 019).....	45
Admin View Post (Use Case ID 021)	45
Admin Delete Post (Use Case ID 022).....	45
Admin Search User (Use Case ID 023)	46
Admin Update Password (Use Case ID 020)	46
➤ Flow Chart	47
➤ Specifications	48
General Definitions	48
➤ UML.....	50
CHAPTER 3 : SOFTWARE DESIGN	51
➤ The Software Design Activity.....	51
➤ Interactions Between Modules.....	53
Interface Design: Interface UML	53

Runtime-level Sequence Diagram	54
CHAPTER 4 : IMPLEMENTATION.....	59
➤ Technology used.....	59
➤ Project structure.....	59
➤ Actual and Future developments.....	61
➤ Implementation Class diagram UML.....	64
CHAPTER 5 : TESTING.....	65
➤ Scope.....	65
➤ Testing Strategy.....	65
➤ Test Cases.....	65
Sign in with Username and password	65
Sign up with Username and password and email	67
Sign Out	70
Add post	71
Notify user.....	74
Image Matching.....	76
Admin Sign in with Email password	78
Admin update Password	80
Admin view post.....	82
Admin Delete post.....	83
Admin Delete user.....	84
Admin View/search user	86
Admin Add user	87
CHAPTER 6 : USER MANUAL.....	90
➤ Requirements.....	90
➤ Getting started	90
CHAPTER 7 : CONCLUSION	93
➤ Problem Faced.....	93
➤ Project Summary	93
➤ Future Work	93
CHAPTER 8 : REFERENCES	94

Chapter 1 : FEASIBILITY AND STUDY DOCUMENT

➤ Introduction

Ukraine finder is a web-based project. It provides a platform through which people around the world, in general, but particularly in Ukraine, can find their lost ones. The category of lost/missing persons includes mostly missing due to the current war situation in Ukraine, abducted or run-away children, mentally retarded and elderly people with memory loss, etc. Other than this, not only in Ukraine but also worldwide we can witness tons of such cases.

➤ Literature Survey

The term '**missing Ukrainians**' referred to those persons who have been missing from home due to war issues, mainly. According to the research, over 7.1 million people have been internally displaced since the invasion of Ukraine, according to the second Ukraine Internal Displacement Report issued by the International Organization for Migration (IOM). This represents a 10% increase in number the of internally displaced persons (IDPs) in Ukraine since the first round of the survey on 16 March 2022 (Msehli & Dillon, 2022). Since the spring of 2014 when conflict ("counter-terrorism", as it is more often described by parties on both sides) began in Eastern Ukraine, thousands are believed to have gone missing because of refugee flight, fighting, reprisal and abductions. Following Russia's annexation of Crimea in March 2014, a marked deterioration in the human rights environment on the peninsula has been reported. Because of the conflict, it is estimated that 86,000 people have been internally displaced, over 6,000 have been injured and more than 3,000 have been killed. The exact number of those who have gone missing is unknown. The continuing violence involves a variety of actors, including Ukrainian military forces, pro-Ukrainian volunteers, pro-Russian separatists, and allegedly Russia itself. The conflict has resulted in a general breakdown of the rule of law in contested parts of the country. Estimates assembled from a variety of sources suggest that since the beginning of 2014 more than 2,500 people may have been kidnapped, abducted, forcibly disappeared, or have gone missing from all levels of Ukrainian society, as well as from all sides of the political and conflict divide. According to sources interviewed by ICMP in Ukraine in September 2014, of this number, the fate of almost 1,000 persons is unknown. As fighting has continued, these numbers have continued to rise, which means that the discovery of more mass or clandestine graves like those found in Slovyansk in July 2014 is to be expected. It is a big challenge for Police and NGOs as they provide food and shelter to these missing people/children, but it is not the ideal solution. Tracing their families and sending them home is the foremost task. But, reuniting these missing persons and elderly with their families is a mammoth task in Ukraine, a country with a 44.13 million population and poor coordination between police departments within different cities due to the war situation (Morvan, 2022).

➤ Problem Statement

Being out of touch with a loved one is concerning, especially during the war, and not hearing from them is terrifying. Several cases of missing people have been reported for many years, but most of the searches turn out unsuccessful. To quickly reunite families and friends with their missing loved ones, a solution for effectively searching for the missing people is presented. Many studies conclude that we need to leverage mobile-based technology to devise a more efficient and rapid method of finding

missing persons more easily and quickly. The website will enrich the communication activity among users and police. The compliant holder and admin will get an appropriate result. The admin will continuously update the database and delete unnecessary data. In our evaluation of this solution, a form (Ribolla) was created to understand what nowadays people think about a possible solution that helps them find a person. From it, we obtain useful results to be used to define our solution feasibility of it, in detail:

5. Have you ever heard of a story of a missing person or do you know anyone who got missing from his/her home or Has he/she been reported missing? (0 point)

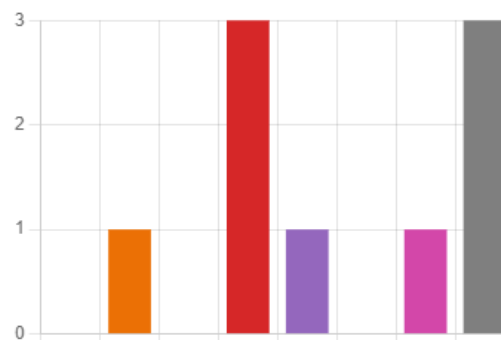
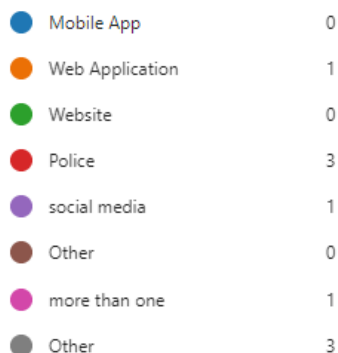
[More Details](#)



The responses underlying that 26% of people had lived in a situation in which a person was missed. Thus, this situation is even present in the normal life of people, so it is a real problem to face, and a solution can be developed to solve it.

6. what was **used to find** the missing person? (0 point)

[More Details](#)



In this question, people highlight that nowadays the most used way to find a person is the police, but the services related to the internet (web application, social media) are at the same level as it. So, an internet solution can be developed to find a person.

14. Based on the option you selected for question 6, Has the missing person been found? (0 point)

[More Details](#)

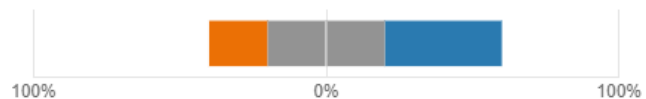


17. How long did it take? (0 point)

[More Details](#)

■ Hours ■ Days ■ Weeks ■ Months ■ Years

Select the duration



Considering the previous question, it is possible to understand that the actual solution does not solve the problem to find people swiftly. In fact, 20% of people answer days, 40% of people answer weeks such as months, and no one answers hours (that is our goal for our solution)

18. if a person is missing are you interested to (0 point)

find a person through ?

(you can select more than 1 option)

[More Details](#)



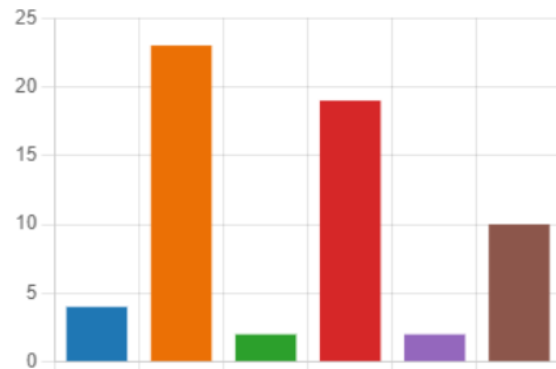
This question highlights that the internet solutions are the most selected by the people, so creating a new internet solution, can help us to have a lot of users that use our solution.

27. According to you mostly **persons are missed** from their homes in situations like
(you can choose more than one option)

(0 point)

[More Details](#)

war	4
kidnapping	23
Eathquakes	2
mental illness	19
tsunami	2
got lost	10



30. Does mobile/web application **beneficial to find** missing people that are
(you can choose more than one option)

(0 point)

[More Details](#)

dramatic events (Ukraine-Russia...	14
in desolate places (in mountains...	6
Earth events (earthquakes, tsuna...	9
people that suffer of mental illn...	18
people that are kidnapped	15



The above 2 questions show that the war situation such as kidnapping and mental illness are the possible situations in which an internet solution can be used to find a person

33. Does the mobile/web application need to get missing person **data from?**
(you can choose more than 1 option)




[More Details](#)

Government	21
ONG/NGO	7
user	12



36. The missing person application should also include (0 point)
additional features for users, like
(you can choose more than one option)

[More Details](#)

	Medical help	26
	Clothing help	14
	Food help	12

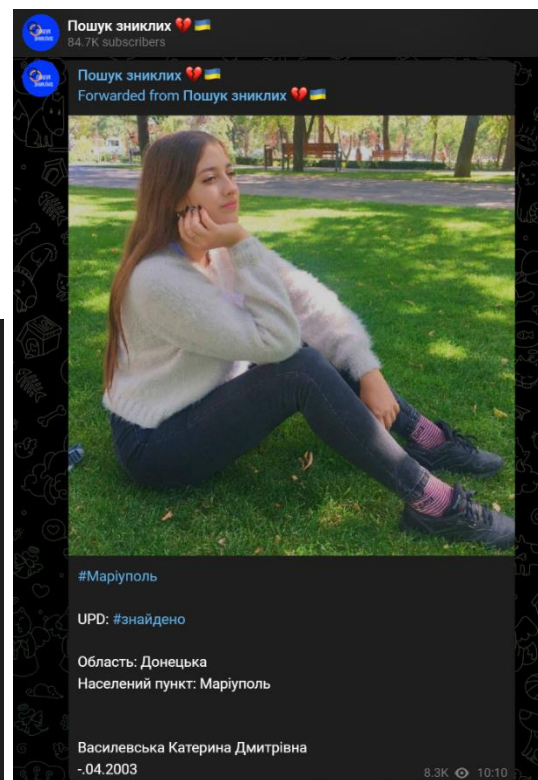
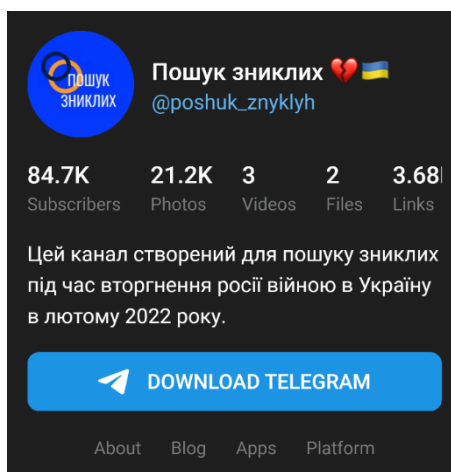


This question helps us to understand what could be implemented in future versions of the application, focusing on medical help.

➤ Existing solutions

Telegram channel

To find the missing persons, there is a specific telegram channel named Пошук зниклих [@poshuk_znyklyh](#) with 84.7K subscribers, having 21.2K photos and data of missing persons. The people here post photos and details including age, physical appearance, and the place from where they are missing. This group is growing fast, and this group is in the local Ukrainian language. The only problem with this type of platform is that because of so many posts, a person can neglect some important posts.



If we compare this with our project, we are thinking to develop a system which will automatically image match the lost person with the finder person's details and send notifications. So, in this way, our method is more feasible and effective and less time-consuming.

Missing Children Europe is the European Federation for Missing and Sexually Exploited Children, connecting 31 grassroots organizations in 26 countries across Europe. Committed to the same goals: to prevent children to go missing, and to protect children from any violence, abuse or, exploitation that leads to or results from them going missing.

Each year, 250, 000 children slip through the net in the European Union. Children go missing for a wide range of reasons, including conflict, violence, abuse, and exploitation. Missing Children Europe strives to protect and empower children to prevent them from going missing, and to keep them from harm when they do go missing.



If we compare this with our project, we find that our proposed project will not only focus on children but on people of all ages. It will automatically image match the lost person with the finder person's details and send notifications. So, in this way, our method is more feasible and effective.

National bureau

The main objective of the National Information Bureau for searching is to gather and summarize information on the Ukrainian war's internee, missing, confined by the colonizers, deceased, etc.

Wars have always been a cause of disuniting people's connectedness. Many people get lost due to intractable circumstances. Connecting with them or finding them will be a challenge for ordinary citizens. Therefore, Ukraine has been setting up a National Information Bureau for searching.

A list is made of all categories above-mentioned. This will assist in exchanging recorded internet. It allows civilians to be updated on their related ones.

However, this will not be swift and could cause mismatches. This issue is solved in the proposed project where the app will a rapid tool that supports all civilians, and it provides instant collected data based on the database. They are working on developing it into a website (<https://nib.gov.ua/en/>).



Until the website's completion, the traditional method of writing manually will be in use. In addition, some external bodies corporate in developing some tools like Telegram Channels, and WhatsApp Groups. In brief, civilians are utilizing social media which is not nongovernmental to cope with situations temporarily.

Police Website

The police departments of all provinces have a website on which they have uploaded the details of missing persons, but these are the details of the cases that are only reported to the police. And most of the time police data cannot be shared in a more effective way due to privacy reasons. The delay can also be expected because we don't know Police will start pursuing the case at which time.

Instagram/Facebook

They are social networks. People can post missing person data in different ways, such as in groups or feeds etc. And there are possibilities that people can receive information about the missing person through messages, comments etc. They are not designed solely for the purpose of finding the missing person. So, the effectiveness of our purpose cannot be fulfilled up to 100%.

➤ **Proposed Solution**

Ukraine Finder is a website that speeds up the process of searching the missing person by helping people through an online platform.

Basically, our solution is targeted at the following type of users:

Type of users

Finder/Helper will be the one who will find the lost person and will access our Ukraine finder website to upload the details including the NGO (non-government organization) current location of the lost person along with the picture. The finder can be a common person, police, or any NGO. They all can upload the details of the missing person on this online platform.

Lost/Affected (Missing person) is the one who has unfortunately lost someone close and will use our website to upload the details of the lost person along with the lost person's picture and his own contact details. Lost/Affected (Missing person) can constantly view the newsfeed which will contain the news related to any successful match that has been made by the system.

Our website will timely match the pictures uploaded by the finders and the pictures uploaded by the Affected (Missing person). If any match is found, it will notify both the Affected (Missing person) and the finder. Image matching will play an important role in tracing the family of the missing person and sending them back to their home.

Objectives

The main objective of this service is to help the people who are really depressed in the process of searching for their missing ones. Also, the goal here is to increase the pace of the search process in the war zone area of Ukraine (it can also be used worldwide, not only in a war situation) and facilitate the police department as well. Overall, we have built this platform as an act of charity and to get something positive out of our skills.

The first few minutes and hours of a search for someone who has gone missing are crucial. An aggressive, well-planned initial response is necessary for the effectiveness of the search and rescue process. However, responders on the scene of a missing person search often lack the specialist search and rescue teams' intensive training and preparation of initial search plans. The app provides a simple way to report lost persons in case of war, such as civilians or soldiers. In a nutshell, the app is designed to provide an easy way to reduce the search and rescue efforts.

Comparison of our product with existing solutions

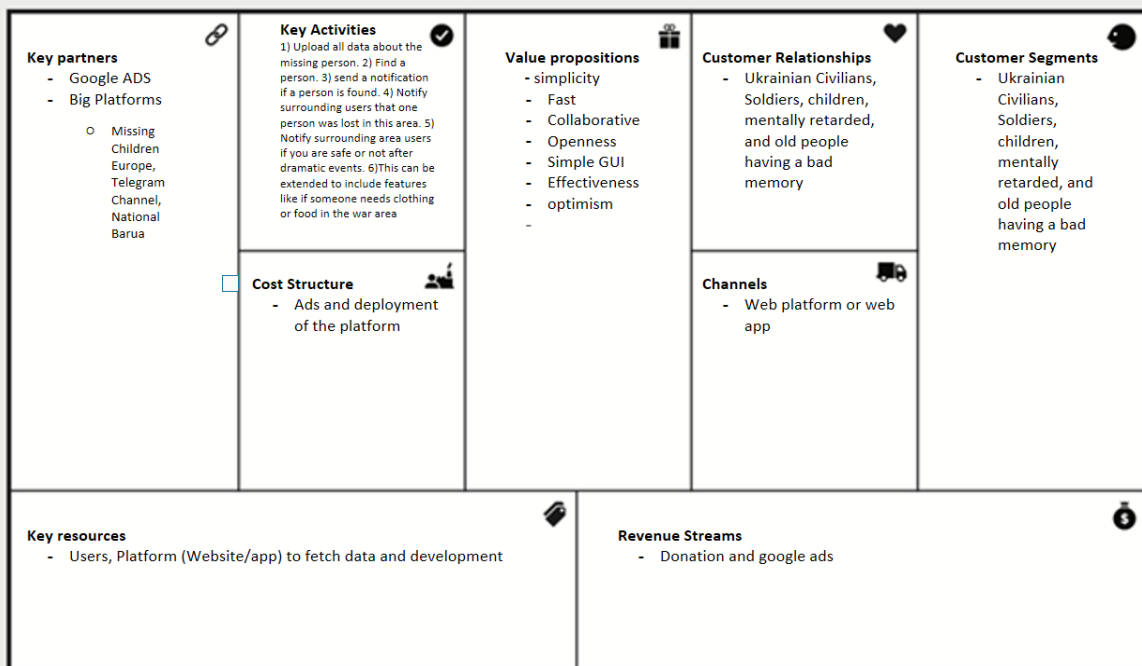
Name	Searching Method	Easy to Use	Effectiveness	Time	Social Media	Access
Ukraine finder (this project)	*Publish = upload the photo and data *Be advertised with latest updates *Easy possibility to become a finder	Yes, there are: *Feed (published post) * Visual buttons with actions to do	At first it will not be popular, but it is expected to be popular and more effective in the future	As fast as possible	Posts can be shared on others social media, saying to join to help the missing person to find him/her	free access to everyone that pass a verification test/check and internet connection
Telegram Channel	It is possible only to Publish a message with photo and data, and wait for a possible answer	Yes, it is a social media app	No, due the fact that messages could be ignored	It depends if the messages are ignored or not, not equal time for all	The message is possible to be shared only on the telegram app and not to the other	Free to everyone, not a real verification test of person identity
Missing Children Europe	the method it is to work with national cops	Yes, through the contact details given on the website	An organization which works in Europe and focuses on children	Time can vary from case to case	It is not possible directly to share data on social media	Through the website or calling the provided phone number
Website Police	Go to police station and inform police to about details of a missing person	It depends on the working days and office hours of a police force	Due to other duties of Police a missing person complain can be ignored for some time	Maybe can take a lot of time if police are not pursuing that case	The data of police station cannot be shared on social media due to privacy	Local police station has access to everyone, but family should go and appear in police station to register a complain.
Instagram /Facebook	It is possible only to Publish a message in different groups with photo and	Yes, it is a social media app	Can be effective, but not so much if we see practically	The time can vary from case to case	Posts can be shared on others social media, requesting to join to help the missing	free access to everyone who have internet connection having Instagram

	data, and wait for a possible answer				person to find him/her	and Facebook account
National Bureau	Complain can be registered by contacting with National Bureau	Yes, one can use by using the contact details given on their website	It is operational in Ukraine, but it also focuses on other matters. Not only to find missing persons	Time can vary from case to case	The data cannot be shared on social media due to privacy reasons	Can be contacted having internet connection through their website or calling on the provided phone number

➤ Business model

Business Canvas

The Business Model Canvas



Marketing Strategy

Short-term goal (next 1-3 years):

- To be used in the Ukrainian situation.
- Having at least one million visits on the website or downloads on Google play and App Store within one year after the release of the official version.
- Help to find at least 1000 people.

Long-term goal (after 3 years up to 8):

- To provide each worldwide user with a sense (by marketing and user experience) that our platform is one of the most usable, less time-consuming, easy, and more effective to search for a missing person.
- It includes that people should give preference to using our platform in the top 10 platforms or organizations.
- And to be expanded also worldwide, not only in the Ukrainian situation.
- We will launch additional features like “ask for help” (i.e., food, clothing, etc).
- At least, 5 million users worldwide.

Main strategy

Our strategy emphasizes four factors, product effectiveness, less time, communication, and branding. The concept of the product aims to satisfy the need of a missing person's family who is in pain. The features will be as simple as possible to reduce users' actions. Furthermore, a minimal user interface design will be applied to the platforms, they will have a minimal pleasant appearance and contain only essential functions to put all details about a missing person.

Organization and staffing

- Software developers to implement all the stuff that this website requires
- User experience design to improve the user interaction to make it more effective to find a missing person.
- Consultation with locals about the proper survey of current situations
- Consultation with professional charity worker about the proper settings for the recommendations we are going to provide for missing people

Financial resources

Funding from friends, a n d family: In the first quarter, during the 1-Phase development, the team members are responsible for all costs.

Google accelerator, Google AdWords: After the 1st-Phase functions have been developed, we will introduce the app to Google accelerator and Google AdWords.

Crowdfunding: In case joining Google, Accelerator is not successful or more fund is required, we will start a Kickstarter campaign through Crowdfunding.

Start-up Business Roadmap

Launch of the project on market

	Phase (I) 2022	Phase (II) 2022	Phase (III) 2023	Phase (IV) 2023
Product	<ul style="list-style-type: none"> 1-Phase software development Missing person database Starting from simple platform 	<ul style="list-style-type: none"> 2-Phase software development Extend the solution along Ukraine Notifying surrounding area 	<ul style="list-style-type: none"> Review and fix issues from users' comment 	<ul style="list-style-type: none"> Release official and improved version
Marketing	<ul style="list-style-type: none"> Marketing research 	<ul style="list-style-type: none"> Introduce product to the market Marketing site launch" (Ukraine) 	<ul style="list-style-type: none"> Focus on B2C promoting digital marketing e.g., Facebook ads, Googles Ads 	<ul style="list-style-type: none"> Digital marketing: B2C and B2B promoting Expand users base from B2C to B2B (other missing person organizations, food, clothing etc.)
Finance	<ul style="list-style-type: none"> Funding from friends, Family crowdfunding 	<ul style="list-style-type: none"> Google accelerator Google AdWords 	<ul style="list-style-type: none"> Crowdfunding: after the launching of the beta version, we will start a kick-starter campaign 	<ul style="list-style-type: none"> Revenue comes from Google Ads and other digital advertising, and charity work.

➤ Assumptions and Constraints

Assumptions

- The scope of the project is defined (help people to find missing person).
- Front-end for the user is easy.
- The user must have an email and Username to register an account. (For Privacy Reasons)
- People who have lost their dear ones will use this website to find them in Ukraine.

Constraints

- This website cannot become popular in people who don't use internet.
- To create an account the main elements are:
 1. A valid email
 2. A secure password
 3. Username
 4. Pass a verification test
- There are no constraints about how many posts can be added when the user creates an account to put details about lost or found person.

➤ Tools and Technologies

- Python
- HTML
- SQLite
- Visual Studio
- Virtual Machine

Chapter 2 : REQUIREMENTS ANALYSIS AND SPECIFICATION DOCUMENT

➤ Requirements given by form's responses

To understand what nowadays people think about a possible solution that helps them find a person, a form was created. Below, a collection of questions related to requirements are collected and analysed to define the possible requirement of our solution:

21. To find a person through mobile/Web application, **How much** it is **important** to (0 point)
have missing person picture

[More Details](#)[Insights](#)

30

Responses



4.67 Average Rating

22. To find a person through mobile/Web application, How much it is important (0 point)
to protect the privacy

[More Details](#)[Insights](#)

30

Responses



3.53 Average Rating

23. To find a person through mobile/Web application, How much it is important to (0 point)
get latest updates

[More Details](#)[Insights](#)

30


Responses



4.57 Average Rating

24. To find a person through mobile/Web application, How much it is important (0 point)
to notify surrounding area people that person is missing here

[More Details](#)

 Insights

30

Responses



4.23 Average Rating

The responses to the above questions underline the importance of some features that our solution should focusing on. They are:

- Include picture of the missing person
- Protect the privacy
- Get the latest update
- Notify surrounding are users

Knowing them, our solution must implement them and include them in possible future versions.

➤ Stake Holders

- Admin
- Lost/Affected (Missing person)
- Helper/Finder (One who finds a missing Person)

Description of stake holders is described in section in proposed solution.

➤ Requirements Analysis

Description of the system

The **sign-up** process must be initiated whenever a new user clicks on the “add member” button located on the welcoming page. Clicking the “add member” button must invoke a pop-up sign-up form. A new user must have a valid email, Username and Password, to complete the sign-up form. The user can later **sign in** by using a Username and the correct password, then the system will sign in the user.

The two users are the affected (Lost button) and the helper (Find button). Both users can do the functions described below.

The user will enter the user panel of the Ukraine finder system that allows different functionalities such as **view posts, add posts, delete a post, search user, edit posts, call, contact details, update password, and update username**. On the other hand, the admin, after signing in **can view, delete**

posts, update passwords, search and delete a user from the admin panel. To add a new post the user should add the photo (at least), the name, address, height, colour complexion, age, and other details of a missing person.

In case the users are helpers/Lost, a notification will be sent to them if the photo or other information corresponds. At this level, the user can confirm if he found the victim and other public users can comment with extra information. Whereas in case the users are the affected (Missing person)/victims, the post will be available to the public to read and comment it.

User Functionalities

Sign up:

FR01.1	Ukraine finder system allows its users to register their selves.
FR01.2	After registration helper, affected (Missing person) will be able to use the Ukraine finder functionalities.

Sign in:

FR02.1	Ukraine finder system allows its users to sign into the website.
FR02.2	Helper and Affected (Missing person) will be able to use the Ukraine finder functionalities, only after signing into the website.

Sign Out:

FR03.1	The system allows user to sign out.
---------------	-------------------------------------

Update Username:

FR04.1	Ukraine finder allows helper to update his/her username.
FR04.2	Ukraine finder allows affected (Missing person) to update his/her username.

Update Password:

FR06.1	Ukraine finder system allows helper to update his/her password.
FR06.2	Ukraine finder system allows affected (Missing person) to update his/her password.

Add Post:

FR07.1	Ukraine finder allows helper to add a post about any found person.
FR07.2	Ukraine finder allows affected (Missing person) to add a post about a missing person.

View Post:

FR08.1	The system allows user to view post of missing persons on the news feed.
---------------	--

Delete Post:

FR09.1	Ukraine finder system will automatically delete the post of both helper and affected (Missing person) when the system will find the missed person and hand over it to affected (Missing person) family.
FR09.2	The Ukraine finder system allows the helper or affected (Missing person) to delete their post.

Edit Post:

FR10.1	The system allows the Affected (Missing person) to edit their post.
FR10.2	The system allows the helper to edit their post.

Searching User:

FR12.1	The Ukraine finder system has easy searching of their users.
---------------	--

Contact Details:

FR13.1	The system shall allow the affected (Missing person) to add his or her contact details.
FR13.2	The system shall allow helper to add his or her contact details.

Call:

FR14.1	The system shall allow the finder to call affected (Missing person) in case of a successful match.
FR14.2	The system shall allow affected (Missing person) to call helper if there is successful match.

View Notification:

FR16.1	Ukraine finder system generates the notification for the helper and affected (Missing person) .
FR016.2	The helper and affected (Missing person) can view notification.

Add Location:

FR17.1	Ukraine finder allows the helper to add to location where the helper/finder find some missed person.
FR17.2	Ukraine finder allows the affected (Missing person) to add to location of their house.

Admin Functionalities

Sign in:

FR19.1	Ukraine finder system allows its admins to sign into the app.
FR19.2	Admin will be able to use the Ukraine finder admin panel, only after signing into the admin panel.

Update Password:

FR20.1	Ukraine finder admin panel allows admin to update his/her password.
---------------	--

View Post:

FR21.1	Ukraine finder allows admin to view posts done by the finder or affected (Missing person) .
---------------	---

Delete Post:

FR22.1	Ukraine finder admin panel allows admin to delete any post done by finder or affected (Missing person) .
---------------	--

Search User:

FR23.1	Ukraine finder admin panel allows admin to search for any specific User profile.
---------------	---

Delete User:

FR24.1	Ukraine finder admin panel allows admin to delete any User profile.
---------------	--

➤ **Non-functional Requirements**

Usability

- The workflow for the user must be as smooth as possible
- The interaction with the user during the search for the person should be the most efficient one, not time-consuming
- Prompt notifications
- Search of a post and view for notifications should be as fast as possible
- Add a post should be easy and fast
- Take/ upload a photo is simple

Reliability

- The posts/notifications should be up to date every time the user interacts with the app.
- The app allows importing a person's data to a personal phone to access data also offline.
- The app allows making a post draft offline.

Performance

- There are no requirements for the performance

Interface

- The information provided looking at the post should be enough and clear

Implementation

- No special requirements for implementation

Security

- Restricted data for not authenticated users
- User data cannot be accessible to other users, only what the user selects as public, such as phone number, email etc ...
- Identity user test verification

Installation¹

- Application must be available in the corresponding application library
- Application might have a web platform to support those users who do not want to download any Application
- Possibility to download a lite version for a very old phone with not updated hardware and software.

¹ It is only needed if an application will be developed in future, after the website spread and increase their user volume, so the following points are related to that, because there is not an installation for a website

➤ Use Case Descriptions (Scenarios)

Sign Up

Use case ID 001		Use case Name: Sign up	Functional requirements ID: FR01
Priority	High		
Actors: Users (Helper and Affected (Missing person))			
Use Case Summary	Helper and Affected (Missing person) register their selves by require information/data for system		
Pre-condition:	Data should be valid which is to be added in the system.		
Normal Flow of Events		Alternative Path	
1. The use case starts when the helper and Affected (Missing person) will add their information (Username, and email).			
2. The helper and Affected (Missing person) will confirm the registration by submitting the data (to fill a form).		2b. The system will not add the records if all require information is not present.	
3. The system will respond by checking the information is valid or not (using also a verification test).			
4. If the information is valid system will add the information.			
5. Use case ends			
Exceptions / Alerts			
The system will not allow the user to add information if the data is not valid.			
Post Conditions			
Information will add correctly in the system.			

Sign In

Use case ID 002		Use case Name: Sign In	Functional requirements ID: FR02
Priority	High		
Actors:users (Helper and Affected (Missing person))			
Use Case Summary	Helper and Affected (Missing person) sign into the system by providing their credentials.		
Pre-condition:	Username should be valid, and password should be correct.		
Normal Flow of Events		Alternative Path	
1. The use case starts when the user wants to use Ukraine finder website.			
2. The user will enter the username and password.			
3. The system will respond by checking the information is valid or not.		4a. If the Username or password is not valid the system will show an error message to re-enter the details.	
4. System will check if the password is correct		5b. If the password is incorrect the system will show an error message to re-enter the password.	
5. Use case ends			
Exceptions / Alerts			
The system will not allow the user to add information if the data is not valid.			
Post Conditions			
Sign In can be successful if username and password is correct			

Sign Out

Use case ID 003		Use case Name: Sign Out	Functional requirements ID: FR03.1
Priority	High		
Actors: users (Helper and Affected (Missing person))			
Use Case Summary	Helper and Affected (Missing person) can sign out from the system.		
Pre-condition:	Helper and Affected (Missing person) must be logged In.		
Normal Flow of Events		Alternative Path	
1. The use case starts when the user wants to Sign out Ukraine finder website.			
2. The user will click the Sign out button.			
3. The system will respond by displaying a message on the top that you have sign out from the system.			
4. The system will direct user to homepage.			
5. Use case ends			
Exceptions / Alerts			
Post Conditions			
The system will direct user to home page of our website.			

Update Username

Use case ID 004 Use case Name: Update Username Functional requirements ID: FR04	
Priority	Medium
Actors:	Users (Helper and Affected (Missing person))
Use Case Summary	Update username allows the user to update his/her name.
Pre-condition:	User previously exists.
Normal Flow of Events	
1. The use case starts when the user wants update username.	
2. The user opens his/her profile setting, then clicks the update profile.	2b. User Does not open any click on any option.
3. The system responds by asking the user with the details want to update.	
4. User edit his/her name.	
5. The user confirms updating.	5a. Cancel Action
6. A system responds by updating the post.	
7. This use case ends.	
Exceptions / Alerts	
The system will not allow a user to Update if name not having alphabets only.	
Post Condition	
Update information edit in all the system.	

Update Password

Use case ID: 006		Use case Name: Update Password		Functional requirements ID: FR06	
Priority	Medium				
Actors	Users (Helper and Affected (Missing person))				
Use Case Summary	Update Password allows the user to change their old password to new password.				
Pre-condition	User must remember their previous password to change password.				
Normal Flow of Events			Alternative Path		
1. The use case starts when the user wants to change their password.					
2. Users select the change password option and enter the old password.					
3. The system will check if the old password is correct.			3a. If the old password is wrong an error message is displayed to retype the old password		
4. User enters the new password and again types it to confirm password.					
5. The system validates the new password.			5a. If the new password is invalid the system show error message to retype the password.		
6. The new password is saved into the system.					
7. Use case ends.					
Exceptions/Alerts					
The system will not allow the user to share the post if that post is not available in the system.					
Post Condition					

Add Post

Use case ID: 007		Use case Name: Add Post	Functional requirements ID: FR07
Priority	High		
Actors	users (Helper and Affected (Missing person))		
Use Case Summary	Add post will allow the user to add post about missing person.		
Pre-condition	Previously post is not available for same missing person.		
Normal Flow of Events		Alternative Path	
1. The use case starts when the user wants to add new post about a missing person.			
2. The user will enter the Name, address, height, color complexion, age and other details of the missing person.			
3. The user will add his own details including his name, contact number and address.			
4. The System validates the given fields.		4a. If the given information is not valid system shows error message to re-enter information.	
5. The System asks the user to confirm the addition.		5a. The user cancels the addition.	
6. This use case ends.			
Exceptions/Alerts			
The system will not allow the admin to add route if there is already			
Post Condition			
Record will be saved to system.			

View Post

Use case ID: 008		Use case Name: View Post	Functional requirements ID: FR08.1
Priority	Medium		
Actors	Users (Helper and Affected (Missing person))		
Use Case Summary	View post allow user to go to view the current missing persons.		
Pre-condition	The user must by logged In.		
Normal Flow of Events		Alternative Path	
1. The use case starts when user want to view the post of missing person.			
2. The user will enter password and email address/ phone number.			
3. The System validates the given fields.		3a. If the given information is not valid system shows error message to re-enter information.	
4. The system directs user to home page with new posts about missing persons and view successful matches.			
5. This use case ends.			
Exceptions/Alerts			
The system will not allow guest user to view posts on the website.			
Post Condition			
The user will be able to view posts of missing persons.			

Delete Post

Use case ID 009		Use case Name: Delete Post		Functional requirements ID: FR09	
Priority	Low				
Actors:	Users (Helper and Affected (Missing person)), System				
Use Case Summary	Deleting post allows the user to permanently remove post from the system.				
Pre-condition:	Post previously exists.				
Normal Flow of Events			Alternative Path		
1. The use case starts when the user wants to delete the post or System deleted it					
2. The user selects the post that he wants to delete.					
3. The system responds by asking the user to confirm deletion.					
4. The user confirms deletion.					
5. A system responds by deleting the post.			5a. Cancel Action		
6. This use case ends.					
Exceptions / Alerts					
The system will not allow a user to delete the information that is being used in the system.					
Post Condition					
Deleted post not exist in the system,					

Edit Post

Use case ID 010		Use case Name: Edit Post		Functional requirements ID: FR10	
Priority	Medium				
Actors:	Users (Helper and Affected (Missing person))				
Use Case Summary	Edit post allows the user to Update post.				
Pre-condition:	Post previously exists.				
Normal Flow of Events			Alternative Path		
1. The use case starts when the user wants to edit the post					
2. The user selects the post want to edit.					
3. The system responds by asking the user for the details (Password, username, Email, Phone number) want to update.					
4. Users edit the information and edit data.					
5. The user confirms updating.			5a. Cancel Action		
6. A system responds by Updating the post.					
7. This use case ends.					
Exceptions / Alerts					
Post Condition					

Search User

Use case ID: 012		Use case Name: Search user	Functional requirements ID: FR12
Priority	Low		
Actors	Users (Helper and Affected (Missing person))		
Use Case Summary	Search users allow the user to searching the other users by their usernames		
Pre-condition	The information of user was previously saved in the system user want to search out.		
Normal Flow of Events		Alternative Path	
1. The use case starts when the user wants to search the other user.		1a: If the user does not want to search user, so not need to search, or type in search bar.	
2. The user searches the other user by their username.			
3. The system responds by asking the user to confirm searching an information.			
4. The user confirms the information about user on which he/she wants to search.		4a: The user can cancel the action.	
5. A system responds by searching user and notify the user with the require user profiles link as their username.			
6. The user then views the user profile			
7. This use case ends.			
Exceptions/Alerts			
1. The system will not search the user which is not register in the system.			
Post Condition			
1. The information which is searched is available anywhere in the system.			
2. Every user can view the require search detailed of user profile.			

Contact Details

Use case ID: 013 Use case Name: Add Contact Details Functional requirements ID: FR13	
Priority	Medium
Actors	Users (Helper and Affected (Missing person))
Use Case Summary	Add Contact allows both Affected (Missing person) and helper to add their contact details.
Pre-condition	User must be logged In.
Normal Flow of Events	
1. The use case starts when the user wants to add his or her contact details	
2. User must provide Username and password to log in the system.	
3. The system will check if the provided credentials are valid or not.	3a. If the credentials are not valid the system will display an error message.
4. The user will click on add contact details button.	
5. The system will direct user to text fields where he or she can add mailing address, phone number, email address etc.	
6. The new contact details are saved in the system.	
7. Use case ends.	
Exceptions/Alerts	
1. The system will not allow the user to add details if he or she is not registered.	
Post Condition	

Make a Call

Use case ID 014		Use case Name: Call	Functional requirements ID: FR14
Priority	low		
Actors:	Users (Helper and Affected (Missing person))		
Use Case Summary	It allows Affected (Missing person) and finder to contact each other through call.		
Pre-condition:	Both users must be registered with provided contact details e-g phone number.		
Normal Flow of Events		Alternative Path	
1. The use case starts when the Affected (Missing person) or finder must contact each other through call, they should be logged in.			
2. The finder will click on the phone button below the details of missing person.			
3. The system shall make a call to Affected (Missing person) or vice versa by clicking phone number and then click dial.			
4. This use case ends.			
Exceptions / Alerts			
The system will not allow unregistered user to make calls.			
Post Condition			
The call shall be directed to the respective user by the system.			

View Notification

Use case ID: 016		Use case Name: View Notifications		Functional requirements ID: FR16	
Priority	High				
Actors	Users (Helper and Affected (Missing person))				
Use Case Summary	Generate notifications allow the user to view his/her inform about the profile details and about their post.				
Pre-condition	The notification was generated that will view by users.				
Normal Flow of Events			Alternative Path		
1. The use case starts when the user opens the notification tab			1a. If the user does not want to know about the notification so, he/she not open the notification tab.		
2. Click on the notification and system responds by opening new page with the corresponding notification			2a. If the user does not click on notification so no action performs.		
3. Then user perform certain action according to notify by the system.			3a. user can't perform the certain require action and cancel it.		
4. User come back to home page.					
5. This use case ends.					
Exceptions/Alerts					
Notification generated will not remain for long time.					
Post Condition					
Notification views out.					

Add Location

Use case ID 018		Use case Name: Add location	Functional requirements ID: FR17
Priority	Medium		
Actors:	Users (Helper and Affected (Missing person))		
Use Case Summary	Users will add their locations.		
Pre-condition:	Location must be true.		
Normal Flow of Events		Alternative Path	
1. The use case starts when the user add location in its post.			
2. The system responds by asking the user to add the location from map.			
3. Users add the location.			
4. The user confirms location			
5. A system responds by adding location.			
6. This use case ends.			
Exceptions / Alerts			
The system will not allow a user to add location without map.			
Post Condition			
Location added successfully and must be in range.			

Admin Sign In

Use case ID: 019		Use case Name: Admin Sign In	Functional requirements ID: FR19
Priority	High		
Actors: Admin			
Use Case Summary	Admin signs into the system by providing their credentials.		
Pre-condition:	Username should be valid, and password should be correct.		
Normal Flow of Events		Alternative Path	
1. The use case starts when the admin wants to use Ukraine finder admin panel.			
2. The admin will enter the username and password.			
3. The system will respond by checking the information is valid or not.		3a. If the Username or password is not valid the system will show an to reenter the details.	
4. System will check if the password is correct, if yes then successfully logged in.		4a. If the password is incorrect the system will show to reenter the password.	
5. Use case ends			
Exceptions / Alerts			
The system will not allow the admin to add information if the data is not valid.			
Post Conditions			
Information will add correctly in the system.			

Admin Update Password

Use case ID: 020		Use case Name: Update Password	Functional requirements ID: FR20.1
Priority	High		
Actors	Admin		
Use Case Summary	Update Password allows admin to change their old password to new password.		
Pre-condition	Admin must remember their previous password to change password.		
Normal Flow of Events		Alternative Path	
1.	The use case starts when the admin wants to change their password.		
2.	Admin selects settings and then change password option and enters the old password.		
3.	The system will check if the old password is correct.	3a.	If the old password is wrong an error message is displayed to retype the old password
4.	Admin enters the new password and again types it to confirm password.		
5.	The system validates the new password.	5a.	If the new password is invalid the system show error message to retype the password.
6.	The new password is saved into the system.		
7.	Use case ends.		
Exceptions/Alerts			
The system will not allow the user to share the post if that post is not available in the system.			
Post Condition			

Admin View Post

Use case ID 021		Use case Name: View Post	Functional requirements ID: FR21.1
Priority	Medium		
Actors:	Admin		
Use Case Summary	View Post will allow the admin to view posts done by finders and Affected (Missing person).		
Pre-condition:	The user and post exist in the system.		
Normal Flow of Events		Alternative Path	
1. The use case starts when the admin wants to view posts done by the admin or Affected (Missing person)			
2. The admin opens the profile of the finder/Affected (Missing person)			
3. The admin then clicks on the posts section.			
4. The system responds by showing all the posts of that user to the admin.			
5. This use case ends.			
Exceptions / Alerts			
Post Condition			

Admin Delete Post

Use case ID 022		Use case Name: Delete Post	Functional requirements ID: FR22.1
Priority	Medium		
Actors:	Admin		
Use Case Summary	Delete post allows the admin to permanently remove post from the system of any user		
Pre-condition:	Post previously exists.		
Normal Flow of Events		Alternative Path	
1. The use case starts when the admin wants to delete any post of user.			
2. The admin selects the post of user that he wants to delete.			
3. The system responds by asking the admin to confirm deletion.			
4. The admin confirms deletion.		4a. Cancel Action	
5. The system responds by deleting the post.			
6. The system notifies the user that his post is deleted by the admin.			
7. This use case ends.			
Exceptions / Alerts			
Post Condition			
Deleted post does not exist in the system.			

Admin Search User

Use case ID: 023		Use case Name: Search user		Functional requirements ID: FR23.1	
Priority	Medium				
Actors	Admin				
Use Case Summary		Search users allow the admin to search user profiles by their names or usernames.			
Pre-condition		The information of user was previously saved in the system that admin wants to search out.			
Normal Flow of Events			Alternative Path		
1. The use case starts when the admin wants to search any specific user.			1a: If the admin doesn't want to search user, then no need to search or type in search bar.		
2. The admin searches the user by entering their username Open the search bar.					
3. The system responds by asking the user to confirm searching an information.					
4. The admin confirms the information about user on which he/she wants to search.			4a: The admin can cancel the action.		
5. System responds by searching user and notifies the admin with the required user profiles.					
6. The admin can then view the user profile					
7. This use case ends.					
Exceptions/Alerts					
The system will not search the user which is not register in the system.					
Post Condition					
The information which is searched is available anywhere in the system.					

Admin Delete User

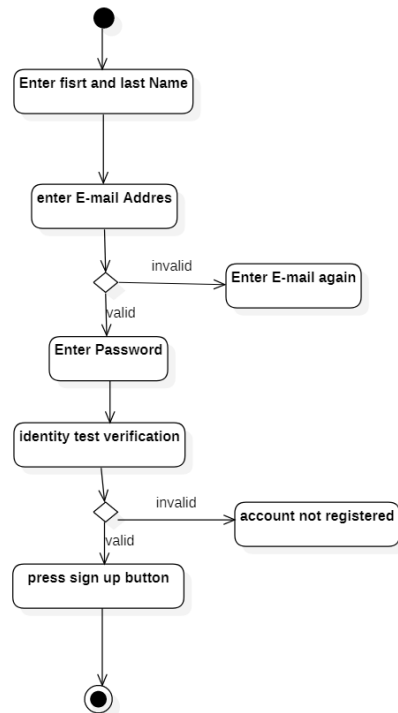
Use case ID 024		Use case Name: Delete User	Functional requirements ID: FR24.1
Priority	Medium		
Actors:	Admin		
Use Case Summary	Delete User allows the admin to permanently remove any user account from the system.		
Pre-condition:	User previously exists.		
Normal Flow of Events		Alternative Path	
1. The use case starts when the admin wants to delete any user from the system.			
2. The admin selects the user that he wants to delete.			
3. The system responds by asking the admin to confirm deletion.			
4. The admin confirms deletion.		4a. Cancel Action	
5. The system responds by deleting the user from the system.			
6. This use case ends.			
Exceptions / Alerts			
Post Condition			
Deleted User does not exist in the system.			

Search Missing Person

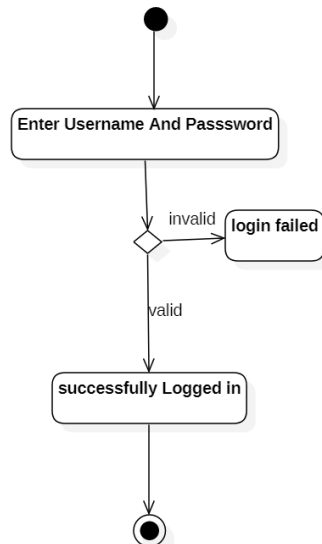
Use case ID: 025		Use case Name: Search missing person	
Priority	Medium		
Actors	Users (Helper and Affected (Missing person)) /Admin		
Use Case Summary	Search missing person allow the users/admin to search the missing person by image matching or by bio data.		
Pre-condition	The users or admin is signed in and there is no missing person match before.		
Normal Flow of Events		Alternative Path	
1.The use case starts when the user wants to search the missing person.			
2. The user/admin searches the other missing person by image.		2a. The user/admin searches the other missing person by bio data.	
3. The system loads the image and apply image processing.			
4. The system applies filters and find similar facial features to get different processed images.		4a: If no filter is applied then retrieve data from preprocessed images data base	
5. A system matches the images to view results.			
6. If image matches, then send notification		6a. If image does not match then go back to optional filters to add details more.	
7. This use case ends.			
Exceptions/Alerts			
8. The system will not search the missing person whose image or bio data is missing.			
Post Condition			
1.The missing person who is searched is available anywhere in the system.			
2.Every user/admin can view the required missing person details at user/admin profile.			

➤ Activity diagrams

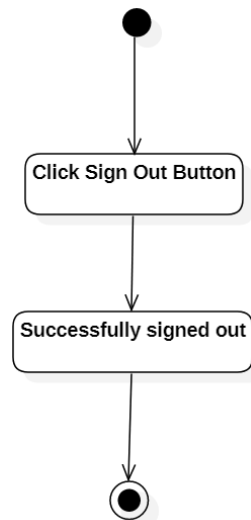
Sign up (Use Case ID 001)



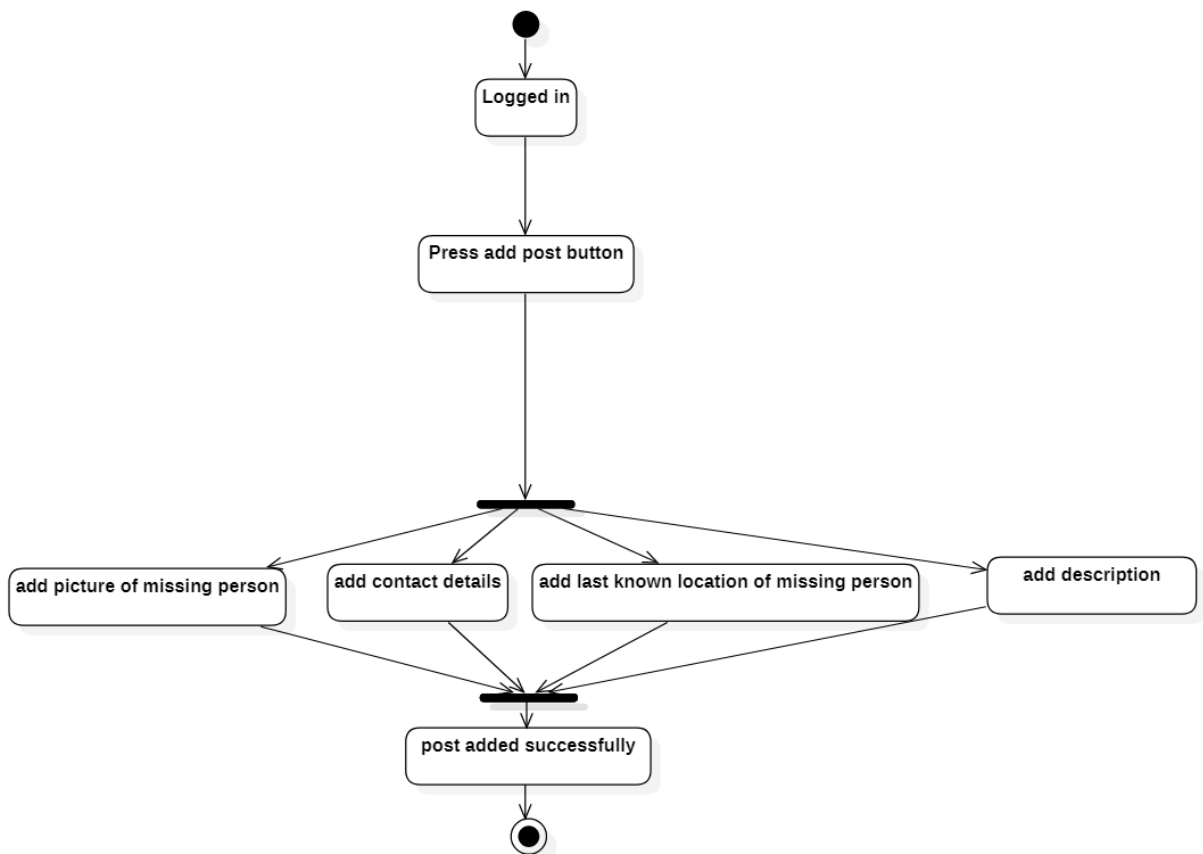
Sign in (Use Case ID 002)



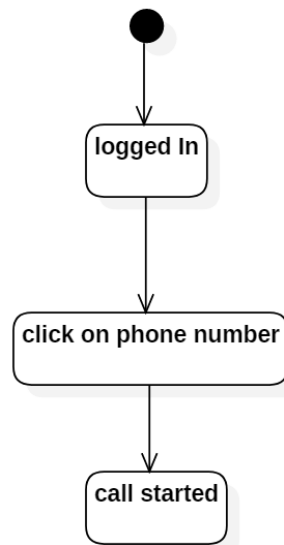
Sign Out (Use Case ID 003)



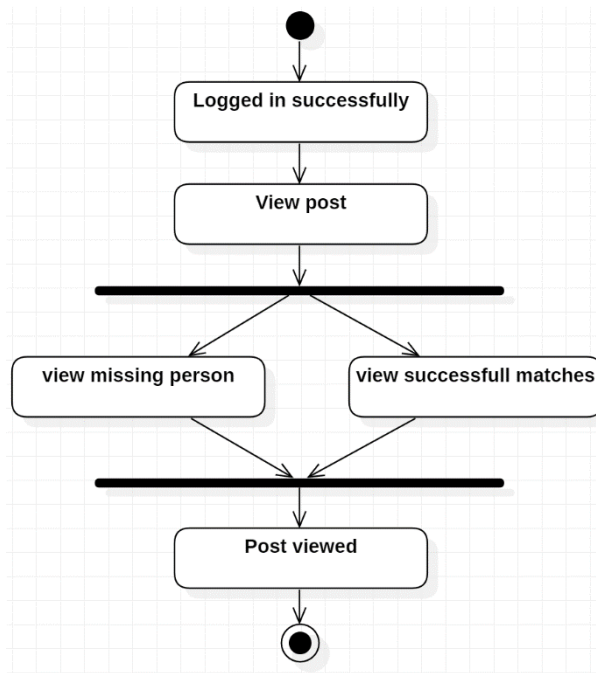
Add Post (Use Case ID 007)



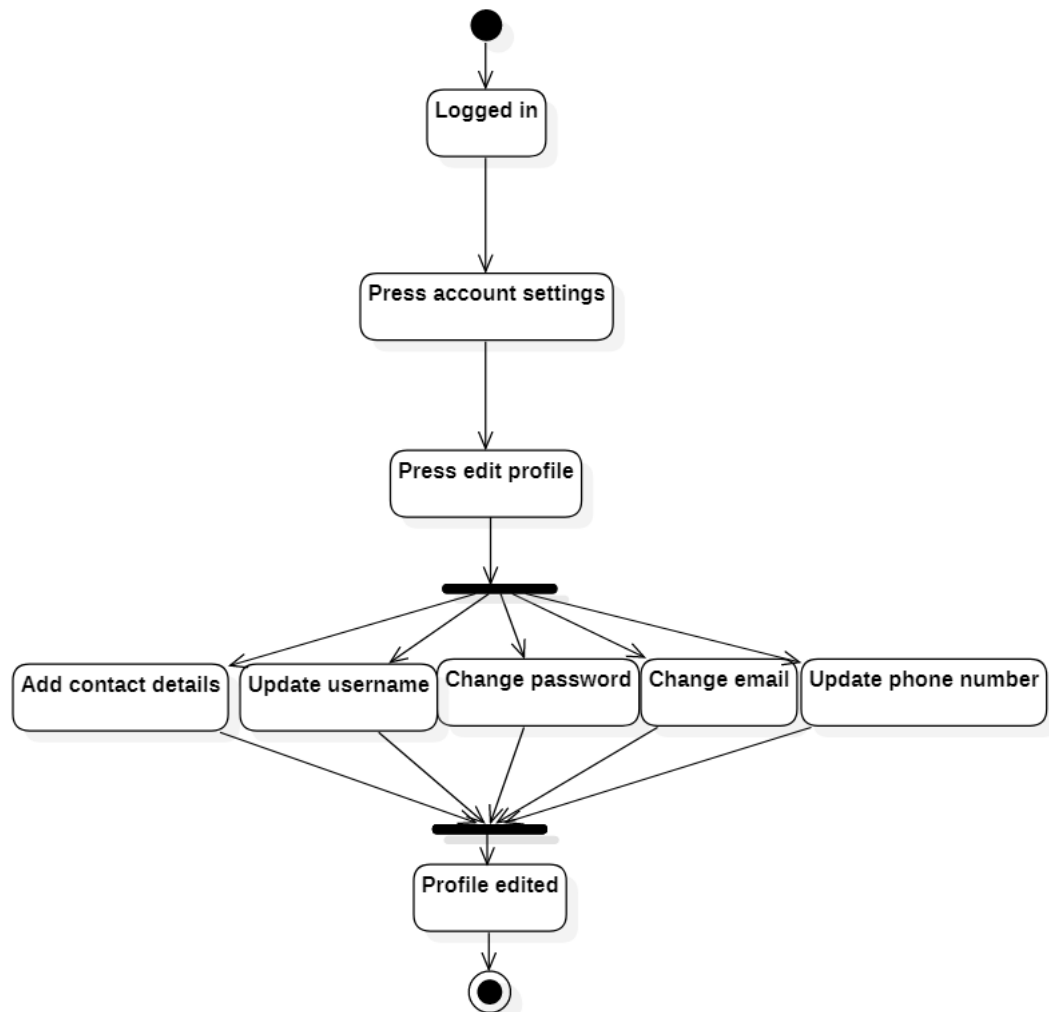
Make a Call (Use Case ID 014)



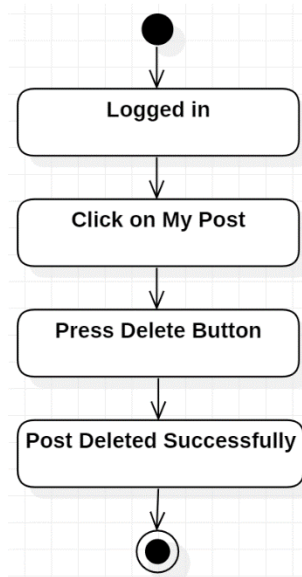
View post (Use Case ID 008)

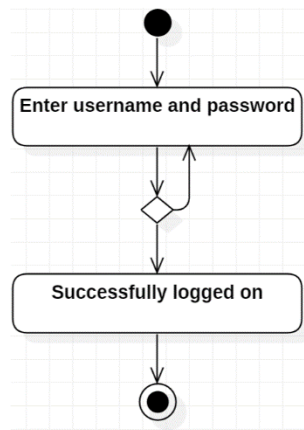
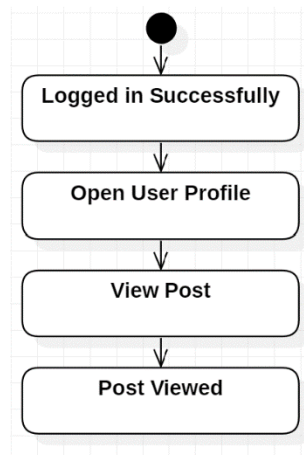
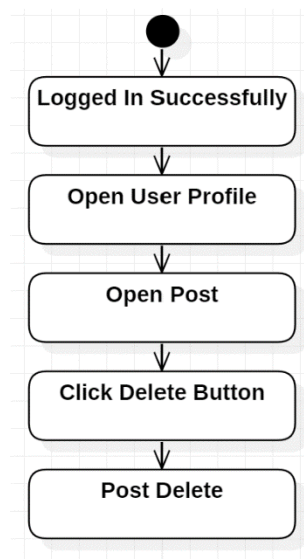


Edit profile (Use case ID 004/006/013)

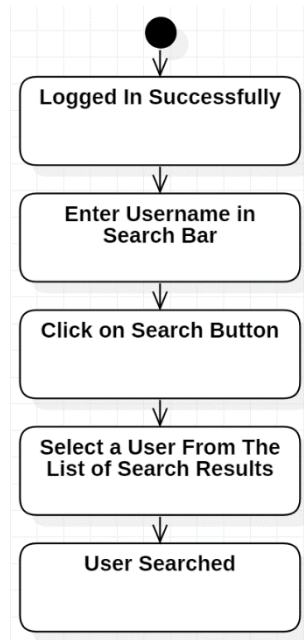


Delete Post (Use Case ID 009)

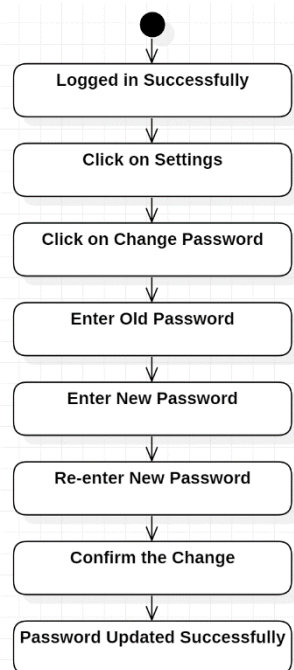


Admin sign in (Use Case ID 019)**Admin View Post (Use Case ID 021)****Admin Delete Post (Use Case ID 022)**

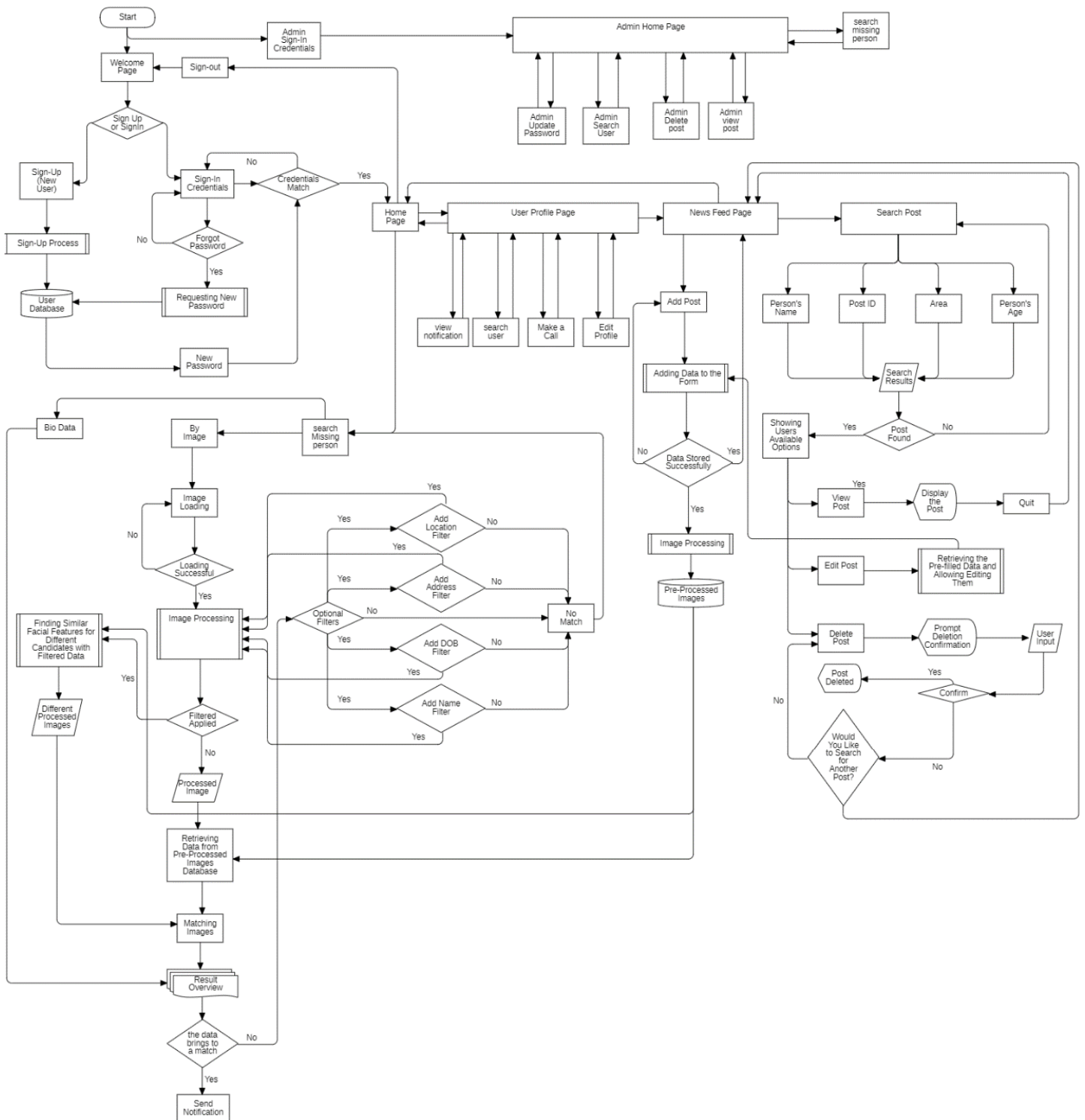
Admin Search User (Use Case ID 023)



Admin Update Password (Use Case ID 020)



➤ Flow Chart



➤ Specifications

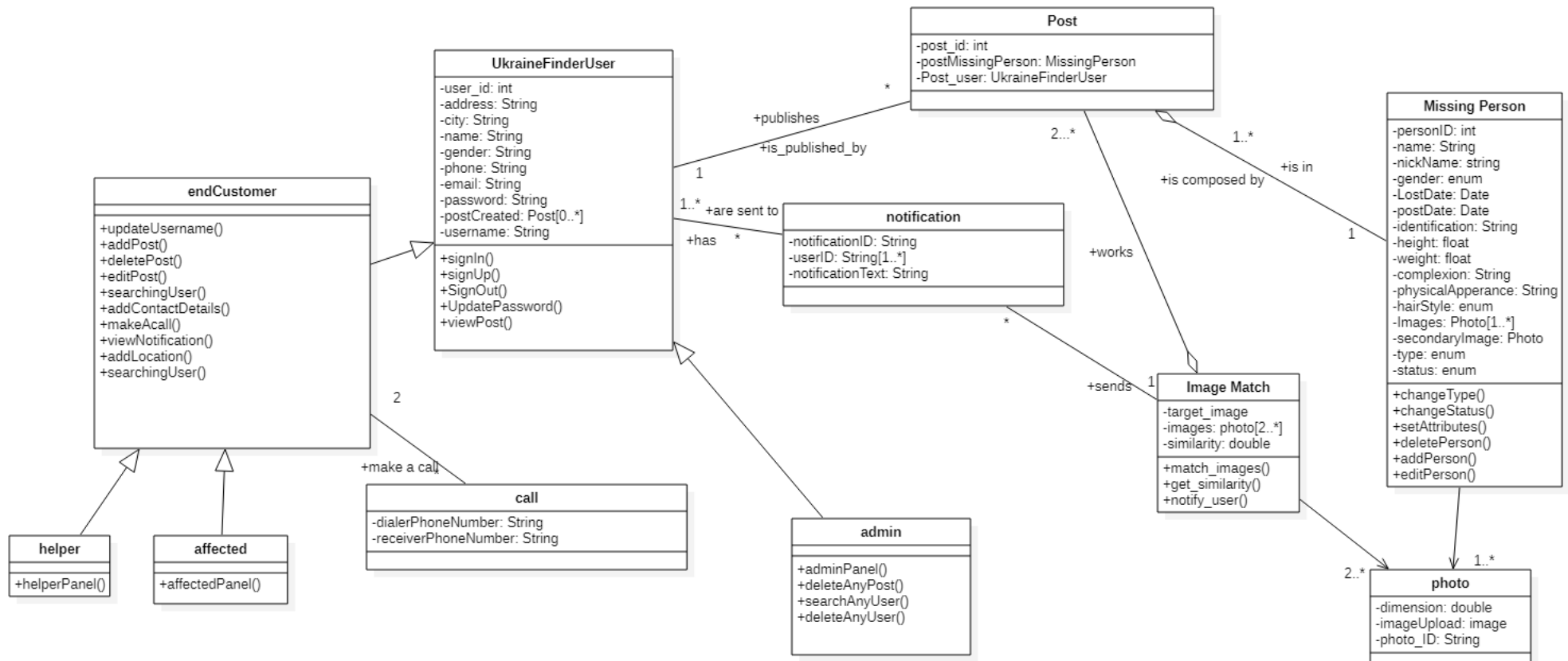
General Definitions

- The missing person can be the subject of different posts (at least 1)
- The Ukraine finder user can publish zero or more than one post
- Each post is related to only one missing person
- Each post is related to only one user
- Posts can be filtered either by missing person data (i.e., Name) or by image
- In the website it is allowed a single email per user
- The username must be unique
- User can publish as many posts as desired for a different missing person but, cannot post a specific/same post again.
- When the user is creating the post, must insert the mandatory data
- The posts are a part of the image matching process.
- The image matching process uses images to find some correspondence among posts' images
- A single post must have only one missing person data
- Every post must have at least one photo
- Each post made by a user can be edited by that user only. Nevertheless, the user can still view other users' posts without editing them.
- A notification can be sent to one or more user
- A user can receive zero or more notifications
- A call must have only 2 users
- A user can have zero or more calls
- The image match can send zero or more notification
- A notification is sent by only one image matching
- The image matching process ends with a notification if a match is found
- The image match can work if there are at least 2 or more posts (2 photos), to be compared

- The missing person can have at least 1 or more photos

➤ UML

Specification Class diagram UML



Chapter 3 : Software Design

➤ The Software Design Activity

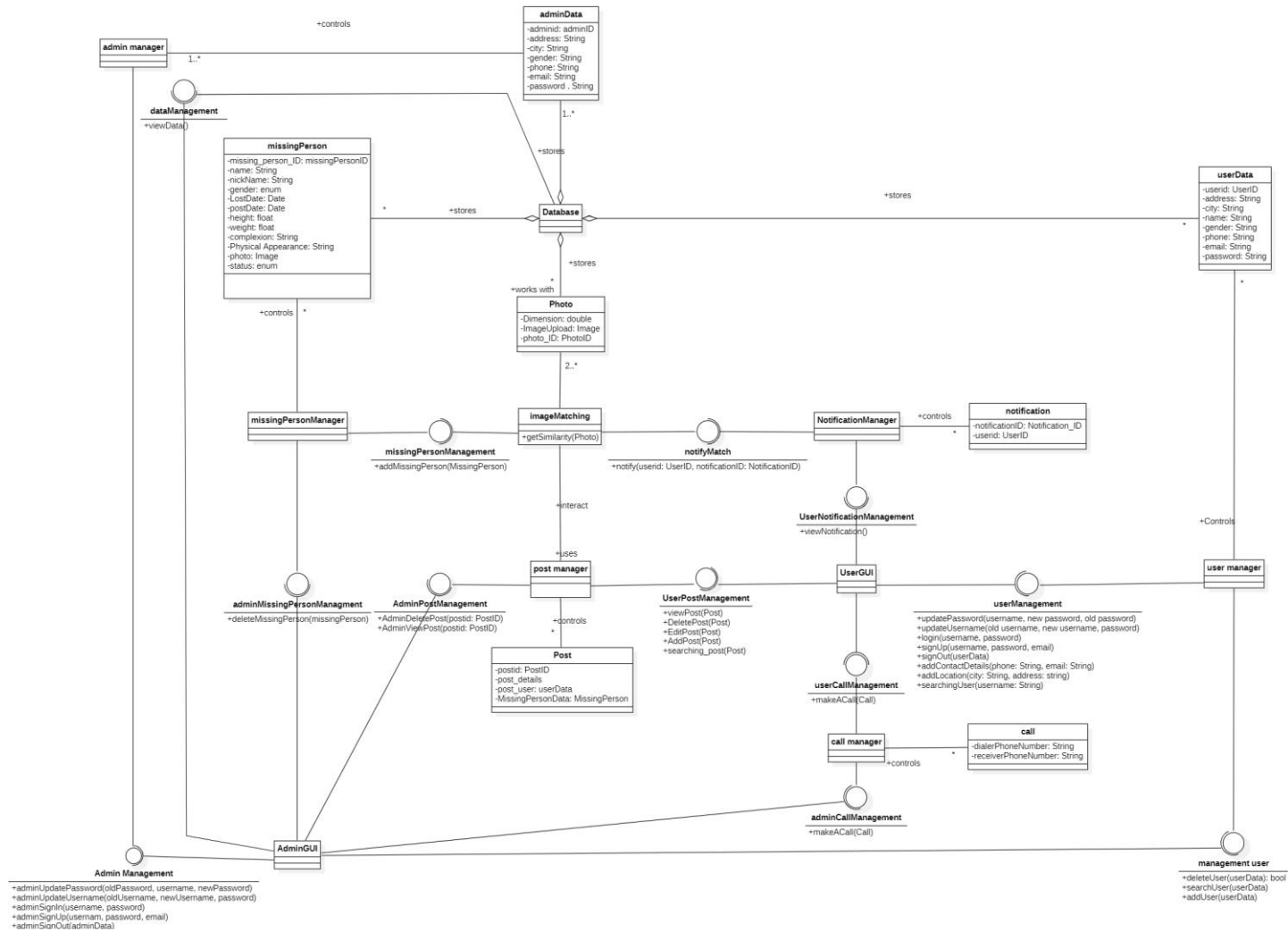
Modules	Description	Interface - Provided Service(s)	
userGUI	Graphic User Interfaces		
AdminGUI			
User Manager	Provides the services to manage the operations that the user can do	Management user (AdminGUI → User Manager)	<i>adminDeleteUser (userData)</i>
			<i>adminfSearchUser (userData)</i>
			<i>adminAddUser(userData)</i>
		User management (userGUI → User Manager)	<i>updatePassword(username,new password,old password)</i>
			<i>updateUsername(old username, new username, password)</i>
			<i>login(username, password)</i>
			<i>signUp(username, password,email)</i>
			<i>signOut(userData)</i>
			<i>addContactDetails(phone:String, email:String)</i>
			<i>SearchUser(userData)</i>
			<i>addLocation(city:String, address:string)</i>
Admin Manager	Provides the services to manage the operations that the admin can do including exclusive permissions	Admin Management (adminGUI → Admin Manager)	<i>adminUpdatePassword (oldPassword, username, newPassword)</i>
			<i>adminUpdateUsername(old Username, newUsername, password)</i>
			<i>adminSignIn(username, password)</i>
			<i>searchingUser(username)</i>
			<i>adminSignUp(username, password,email)</i>
			<i>adminSignOut()</i>
Post Manager	Coordinates the task requested by the user to add information about the lost part, view them, edit them, or delete the post	UserPostManagement (userGUI → Post Manager)	<i>viewPost(Post)</i>
			<i>DeletePost(Post)</i>
			<i>EditPost(Post)</i>
			<i>AddPost(Post)</i>
			<i>searching_post(Post)</i>
			<i>AdminDeletePost(postid :PostID)</i>

		AdminPostManagement (<i>adminGUI</i> → <i>PostManager</i>)	<i>AdminViewPost</i> (postId:PostID)
Call Manager	Allows the user to have a call with the associated phone number recorded	<i>userCallManagement</i> (<i>userGUI</i> → <i>call manager</i>)	<i>makeACall</i> (Call)
Notification Manager	Send the notification of the image matching and to the user in case of found	UserNotificationManagement (<i>userGUI</i> → <i>Notification Manager</i>)	<i>viewNotification</i> ()
Image matching	Provides the possibility of matching images	notifyMatch (<i>NotificationManager</i> → <i>imageMatching</i>)	<i>notify</i> (userid:UserID, notificationID:NotificationID)
Missing person manager	Provides the missing's person services	missingPersonManagement (<i>imageMatching</i> → <i>missingPersonManager</i>)	<i>addMissingPerson</i> (MissingPerson)
		adminMissingPersonManagement (<i>AdminGUI</i> → <i>missingPersonManager</i>)	<i>deleteMissingPerson</i> (missing Person)

Data module	Description
Database	Collects the data (reported below in the table)
Photo	This module represents the data related to the photo
userData	This module represents the data related to the user
missingPerson	This module represents the data related to the missing person
adminData	This module represents the data related to the admin
call	This module represents the data related to the call
notification	This module represents the data related to the notification

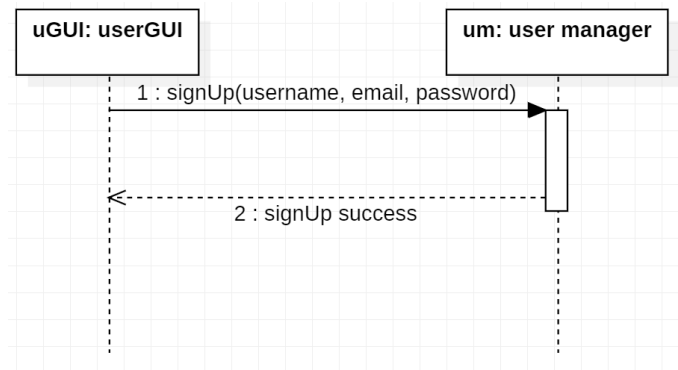
➤ Interactions Between Modules

Interface Design: Interface UML

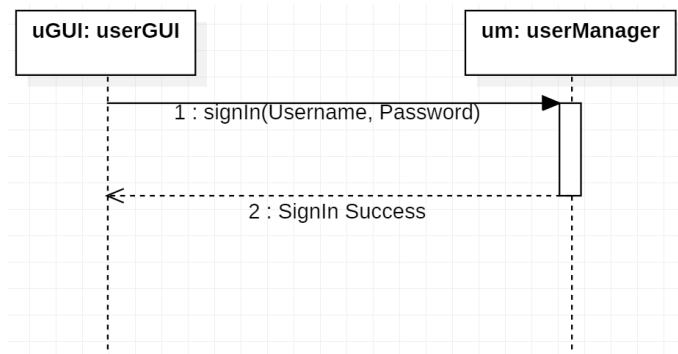


Runtime-level Sequence Diagram

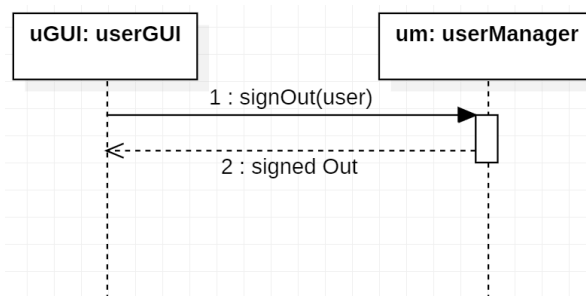
Sign Up (FR01.1)



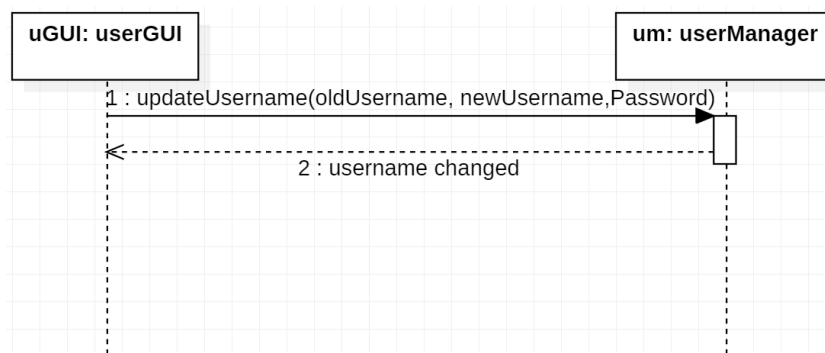
Sign In (FR02.1)



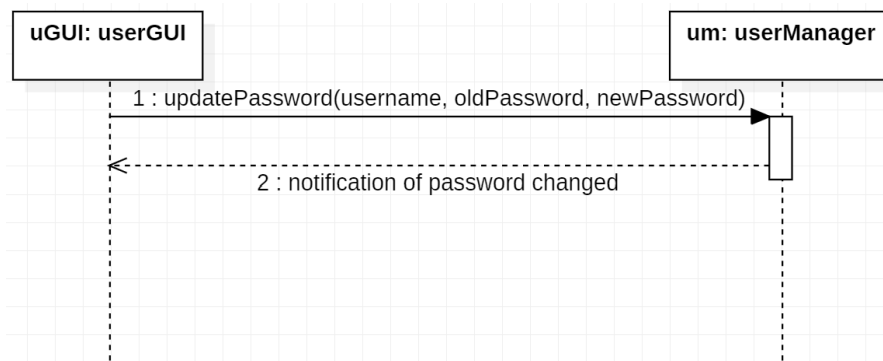
Sign Out (FR03.1)



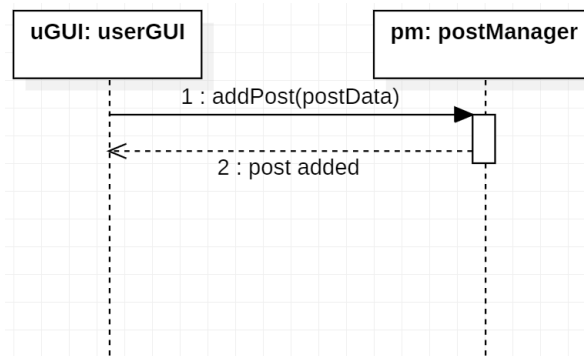
Update Username (FR04.1 / FR04.2)



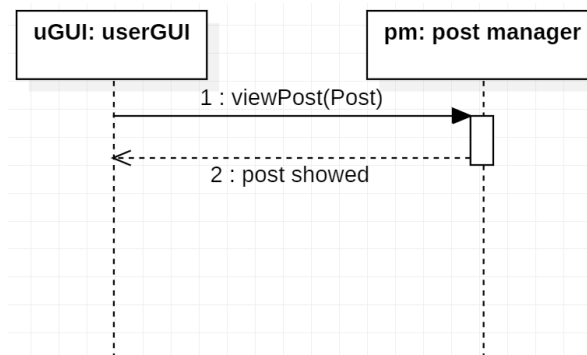
Update Password (FR06.1 / FR06.2)



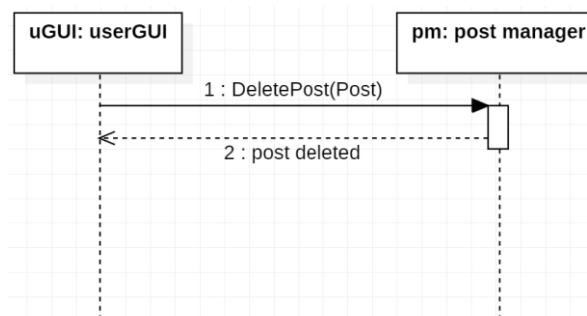
Add Post (FR07.1 / FR07.2)



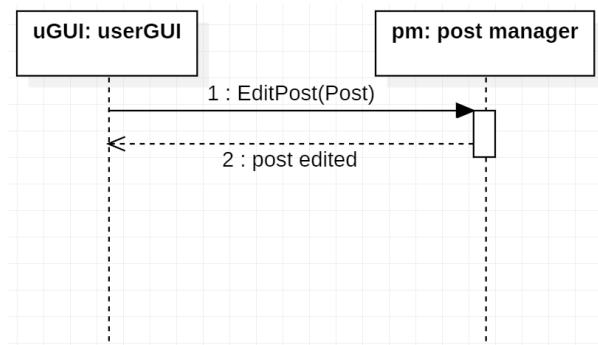
View Post (FR08.1)



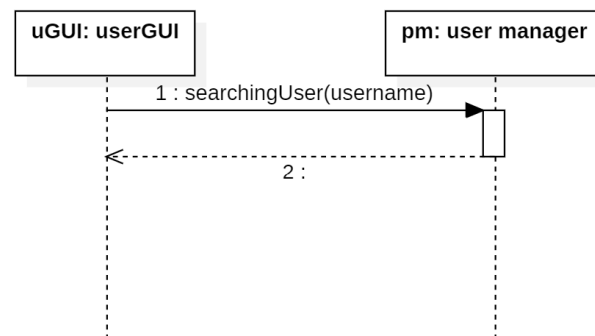
Delete Post (FR09.1)



Edit Post (FR10.1 / FR10.2)



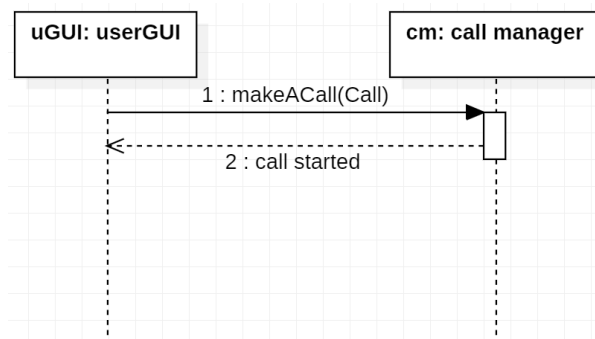
Search User (FR12.1)



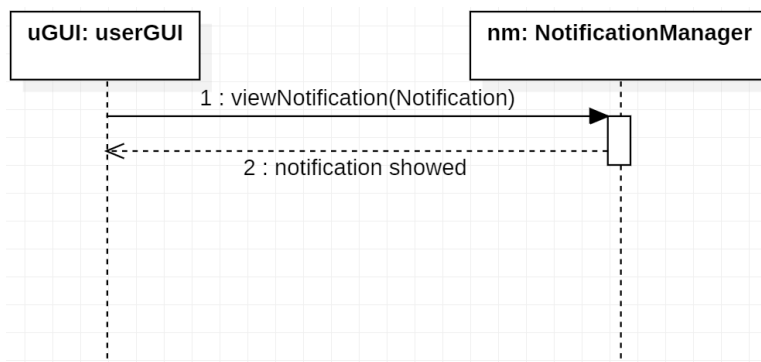
Add contact details (FR13.1)



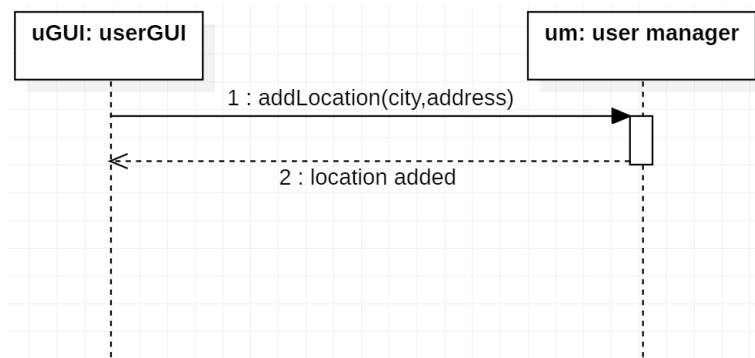
Make a Call (FR14.1)



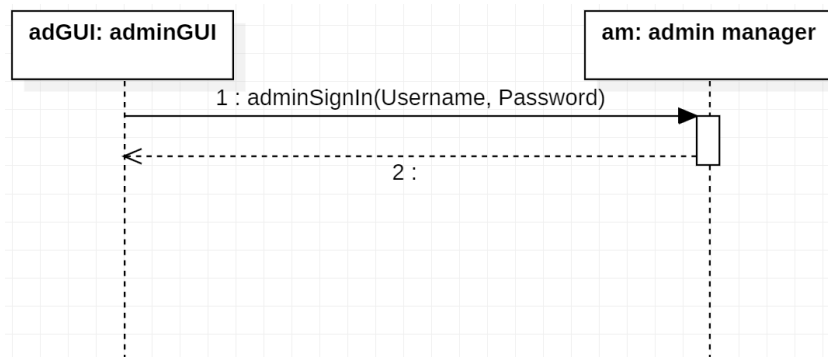
View notification (FR16.1 / FR16.2)



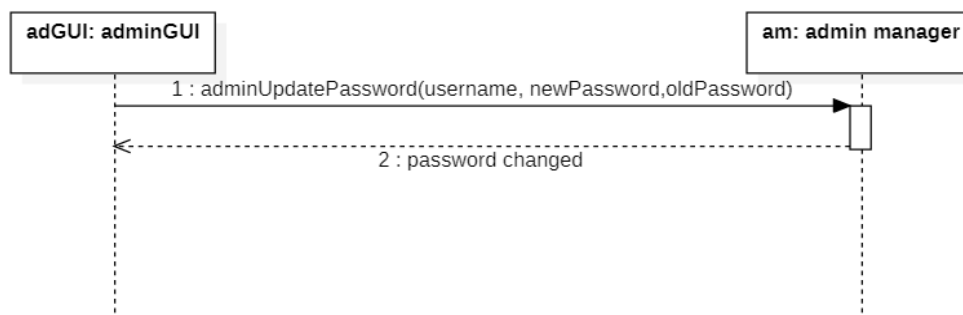
Add Location (FR17.1 / FR17.2)



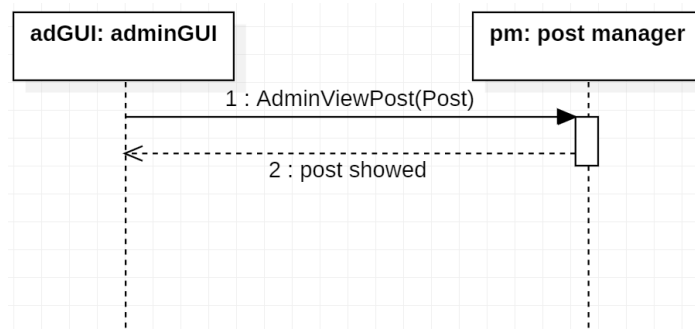
Admin Sign In (FR19.1 / FR19.2)



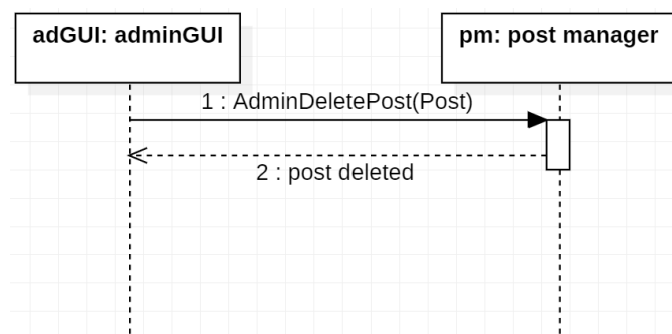
Admin Update Password (FR20.1)



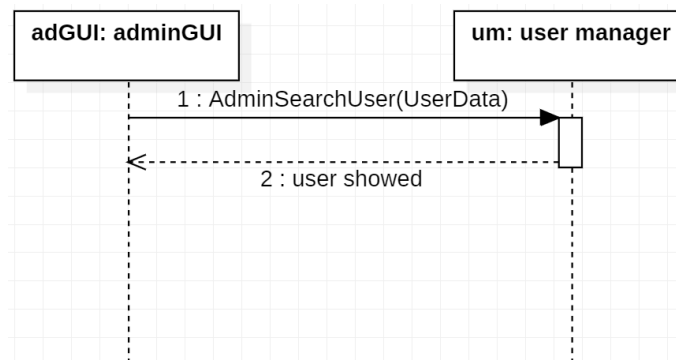
Admin View Post (FR21.1 / FR21.2)



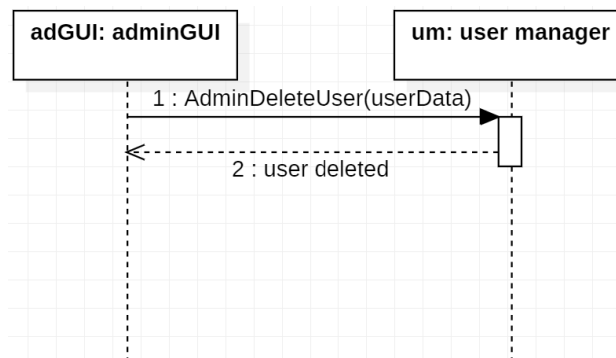
Admin Delete Post (FR22.1)



Admin Search User (FR23.1)



Admin Delete User (FR24.1)



Chapter 4 : Implementation

➤ Technology used

The IDE used is Visual Studio, in which the HTML and Python languages are used to create all the files needed to develop the application. The solution is defined by a back-end and a front-end part. The first has the goal to do the operations with the collected data, and the second has the objective of giving the graphical shape to the application and defining for each page what are the data that the application collects.

➤ Project structure

Front-end: .html Files	
File	Description
add.html	It implements the sign-up form, to add a new user to the database.
find.html	It implements the "Search missing persons" form, where the user needs to upload a picture of the missing person and click the button search to see if a match is found
index.html	It implements the function to give a unique identification number for the users
login.html	It implements the "Ukraine Finder" login page, where the user needs to correctly insert the Username and Password to login and the possibility to be remembered and to add a new member (following the sing up form, given by the add.html file)
Lost.html	It implements the "Report Missing Persons" page. The user can put the name and upload the picture of the missing person to be added in the database
main.html	It implements the "kindly Select the option" page, in which the user reaches after the login and where the user can select the lost (related to Lost.html) or find (related to find.html) option
result_file.html	It implements the database structure, that shows the id, Username, Password and Email of the users

Back-end: .py (python) files	
File	Description
admin.py	It implements the admin functionalities to how see the data for the find, lost and members users. It depends on the admin class from django.contrib and from Lost ,find and members classes taken by "models.py"
apps.py	This module implements the function of configuring the application by defining AppConfig inside it.
forms.py	This module provides tools and libraries to build forms to accept input from site visitors (member/users) and then process and respond the input.
models.py	It defines different classes (and their details) related to the application that are Members (Username, Password, Email), Find (Name, Photo, date) and Lost (Photo, date).

tests.py	It can be used to create different tests for the application, trying different scenario with different testing data
urls.py	It designs URLs for the Ukraine Finder web application and it defines the mapping between the URL path expressions to Python functions (view.py). It is used for the users
views.py	It is a module that takes the web request and returns a web response. In our case it takes the requests of index (index.html), add record (related to data storage), add (add.html) , login (login.html), lost (lost.html), Find (Find.html). It returns the http response to users/members.
settings.py	It is used to configure the options for the Django installation, and it explains how settings work, and which are available. It includes admin utility, server, security.
urls.py	This does the same thing described above for the urls.py but related to the admin portal
wsgi.py	It is the application callable which the application server uses to communicate with our code. It is a Django's deployment platform for web servers and application
manage.py	It is automatically created in Django project. It is used to launch the application.

Database	
File	Description
db.sqlite3	A SQLITE3 file is a database file stored in the SQLite 3 format. It contains structured data records, which contain data types and values. In our case that contains the data about the Find, Lost and Members.

Modules	Code file
userGUI	main.html
AdminGUI	admin.py
Database	db.sqlite3
User Manager	views.py
Admin Manager	admin.py
Post Manager	views.py
Notification Manager	views.py
Image matching	views.py
Missing person manager	views.py

➤ Actual and Future Developments

In the following table there is an analysis of what is developed and what is not with respect to what is described in the above report.

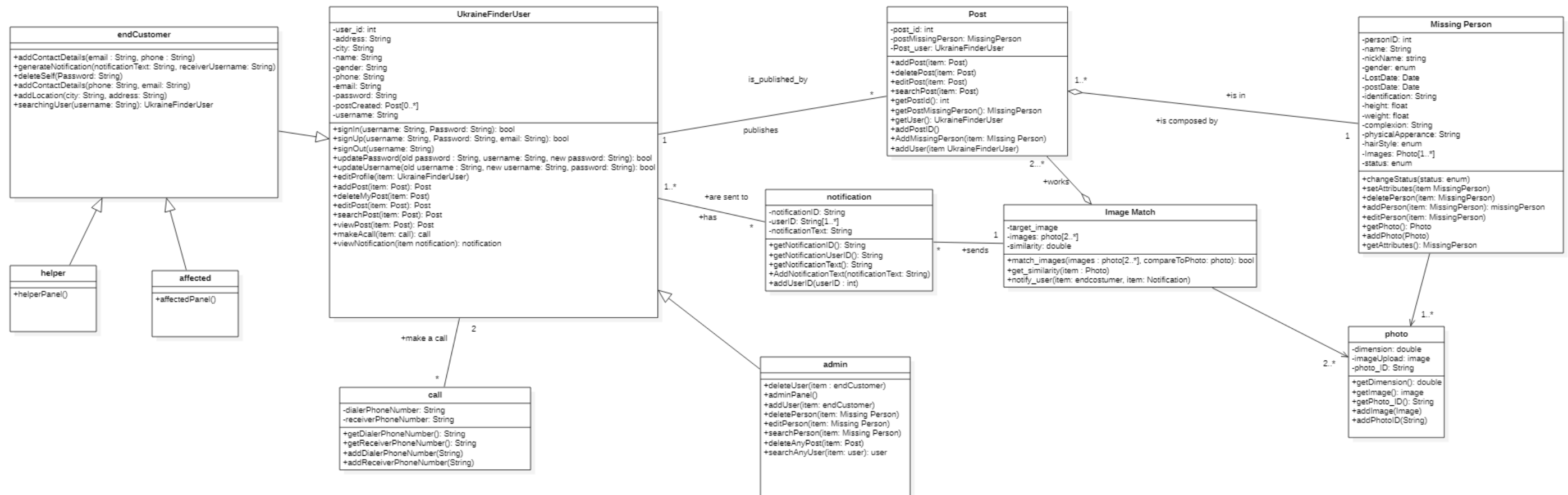
ID	Priority	Function	Description	Status	Notes
FR01.1	High	Sign In	Users can register themselves.	Developed	
FR01.2	High		After registration helper and missing person will be able to use the app's functionalities.	Developed	
FR02.1	High	Sign Up	Users can sign into the website.	Developed	
FR02.2	High		Helper and missing person will be able to use the app's functionalities, only after signing in.	Partially Developed	Because the pass verification task is not implemented
FR03.1	High	Sign Out	The system allows user to sign out.	Partially Developed	
FR04.1	Medium	Update Username	Ukraine finder allows helper to update his/her username.	Not Developed	Because we focus on a more important task
FR04.2	Medium		Missing person can update his/her username.	Not developed	Because we focus on a more important task
FR06.1	Medium	Update Password	Helper can update his/her password.	Not developed	Because we focus on a more important task
FR06.2	Medium		Missing person can update his/her password.	Not developed	Because we focus on a more important task
FR07.1	High	Add Post	Helper can add a post about any found person.	Partially developed	It is a primal version w.r.t. what it is defined above in the report
FR07.2	High		Missing person to add a post about a missing person.	Partially developed	It is a primal version w.r.t. what it is defined

					above in the report
FR08.1	Medium	View Post	The system allows user to view post of missing persons on the news feed.	Not developed	Because we focus on a more important task
FR09.1	Low	Delete Post	The app will automatically delete the post of both helper and missing person when the system finds the missed person and hand over it to their family.	Not developed	Because we focus on a more important task
FR09.2	Low		The app allows the helper or missing person to delete their post.	Not developed	Because we focus on a more important task
FR10.1	Medium	Edit Post	The system allows the missing person to edit their post.	Not developed	Because we focus on a more important task
FR10.2	Medium		The system allows the helper to edit their post.	Not developed	Because we focus on a more important task
FR12.1	Low	Searching User	The app has easy searching of users.	Not developed	Because we focus on a more important task
FR13.1	Medium	Contact Details	The system shall allow the missing person to add his or her contact details.	Not developed	Because we focus on a more important task
FR13.2	Medium		The system shall allow helper to add his or her contact details.	Not developed	Because we focus on a more important task
FR14.1	Low	Call	The system shall allow the finder to call missing person in case of a successful match.	Not developed	Because we focus on a more important task
FR14.2	Low		The system shall allow missing person to call helper if there is successful match.	Not developed	Because we focus on a more

					important task
FR16.1	High	View Notification	The app generates the notification for the helper and missing person .	Partially developed	It is a primal version w.r.t. what it is defined above in the report
FR016.2	High		The helper and missing person can view notification.	Partially developed	It is a primal version w.r.t. what it is defined above in the report
FR17.1	Medium	Add Location	The app allows the helper to add to location where the helper/finder find some missed person.	Not developed	Because we focus on a more important task
FR17.2	Medium		The app allows the missing person to add to location of their house.	Not developed	Because we focus on a more important task
FR19.1	High	Admin Sign in	The app allows its admins to sign into the app.	Developed	
FR19.2	High		Admin will be able to use the Ukraine finder admin panel, only after signing into the admin panel.	Developed	
FR20.1	High	Admin Update Password	The app admin panel allows admin to update his/her password.	Developed	
FR21.1	Medium	Admin View Post	The app allows admin to view posts done by the finder or missing person .	Developed	
FR22.1	Medium	Admin Delete Post	The app admin panel allows admin to delete any post done by finder or missing person .	Developed	
FR23.1	Medium	Admin Search User	The app admin panel allows admin to search for any specific User profile.	Developed	
FR24.1	Medium	Admin Delete User	The app admin panel allows admin to delete any User profile.	Developed	

The not developed functions will be implemented in future website versions

➤ Implementation Class diagram UML



Chapter 5 : TESTING

➤ Scope

The tests are implemented for the function we developed, and we can test it.

➤ Testing Strategy

The black box testing strategy is considered to test the implemented functions-

➤ Test Cases

Sign in with Username and password

Test Case ID: TUK_FR01.1

Function tested: FR01.1 sign-in

Product/Version/Module:

Product name: Ukraine finder

Test Priority (Low/Medium/High): High

Test Case Version TUK_US_ FR01.1

Use Case References: ID 002

Test Design Date: 26-07-2022

Test Execution Date: 28-07-2022

Test Title: User Sign In

Test Case Description: Test the Sign in Functionality.

Pre-conditions/Pre-requisites:

- The user must have an account on Ukraine finder.

Dependencies: Internet connection, already have account

Steps	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Enter Sign in Credentials	Username, Password	User must be able to enter Sign in credentials	User can enter Sign in credentials	pass	Username: shoaib Password:1234

2	Click on Sign in/login button.		User is Signed In	User is signed In.	pass	Click on login button
3	Validate Sign in Credentials			Validated Username and password	Pass	Cursor moving in circle to validate credentials
4	Verify User's Sign in Credentials		Verification is successful	User can Sign In.	pass	success
5	User redirect to home page		User redirect to home page	User redirect to home page	pass	Lost and found page appeared
6	Result				Passed	Passed

Post Conditions:

- Users enter in Ukraine finder main page

Results with different data:

Test case no.	Test title	Test data	Test result	note
1	Normal behaviour	userName = 'shoaib' password = '1234'	Login success	Login successful with correct existing username and password
2	Pre-conditions violation test	userName = null, password= null	No login	Invalid username or password
3	Pre-condition to check password	userEmail='shoaib@gmail.com' userName = 'shoaib' password= '12'	Not login	Wrong password

Ukraine Finder

Your access

Username

Password

Login

☒ Remember me

[Add member](#)

Figure 5.1 Login Page

Kindly Select the Option

Tip: Select the relevant option!:

Stay Safe

Found & Lost

Figure 5.2 Page after the login

Sign up with Username and password and email

Test Case ID: TUK_ FR02.1

Product/Version/Module:

Function tested: FR02.1 sign-up

Product name: Ukraine finder

Test Priority (Low/Medium/High): High

Test Case Version: TUK_ FR02.1

Use Case References: ID 001 (Sign up)

Test Design Date: 26-07-2022

Test Execution Date: 28-07-2022

Test Title: User Sign up with Username and password

Test Case Description: Test the Sign-up Functionality.

Pre-conditions/Pre-requisites:

Dependencies: Internet connection

Steps	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Enter Sign up Credentials	Email Address, Password, Username	User must be able to enter Signup credentials	Users enter all the credentials	Pass	Username: gabriele Password: 5678
2	Click on Sign up button.		Validate sign up Credentials	Validate all fields.	pass	After clicking on sign up data is saved
3	All data successfully saved in Database		Data saved	Data saved	pass	
4	User successfully signup		User redirect to home page	User redirect to home page	pass	
5	Result				passed	

Post Conditions:

- Data correctly save in Database

Results with different data:

Test case no.	Test title	Test data	Test result	note
1	Normal behavior	userEmail = 'gabriel@gmail.com' userName = 'Gabriele' password = '5678'	User data is saved	Userdata is saved in database and they can be viewed from admin panel in "memberss" or from sqlite3 datasbase.
2	Pre-conditions violation test	userEmail = null userName = null password= null	User data not saved because Email is empty Name is empty Password is empty	Not saved in system because username and password is must to store data,

3	Pre-conditions violation test	userEmail='gabriele@gmail.com' userName = 'Gabriele' password= '5678'	User data not saved because Password is lower than 3 charts	Was not able to store because of pre condition that password should be longer than 3 characters -
---	-------------------------------	---	---	--

Ukraine Finder

Your access

Username

shoaib

Password

Login

☒ Remember me

[Add member](#)

Add member

UserName:

gabriele

Password:

Email:

gabriele@gmail.com

Submit

Figure 5.3 signUp page (after the click on the "add member" button)

Add member

UserName:

test2

Password:

..

Please use at least 3 characters (you are currently using 2 characters).

Email:

test2@gmail.com

Submit

Figure 5.4 SignUp page shows error for uncorrect Password

Sign Out

Test Case ID: TUK_FR03.1

Function tested: sign-out

Product/Version/Module:

Product name: Ukraine Finder

Test Priority (Low/Medium/High): High

Test Case Version: TUK_FR03.1

Use Case References: Sign out

Test Design Date: 26-07-2022

Test Execution Date: 28-07-2022

Test Title: Sign out

Test Case Description: Test the Sign out Functionality.

Pre-conditions/Pre-requisites:

- User must be sign in

Dependencies: Sign in

Steps	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Click on logout button		User redirect to Sign in page	User logout successfully	pass	
					pass	

Post Conditions:

- User now have to sign in again

Results with different data:

Test case no.	Test title	Test data	Test result	note
1	Normal behaviour	Click sign out	Logged out	-
2	Pre-conditions violation test	Click some other button	Not logged out	-



Figure 5.5 signOut page

Add post

Test Case ID: TUK_FR07.1_FR07.2

Function tested: FR07.1/FR07.2 Add post

Product/Version/Module:

Product name: Ukraine finder

Test Priority (Low/Medium/High): High

Test Case Version: TUK_FR07.1_FR07.2

Use Case References: ID 007 Add post

Test Design Date: 26-07-2022

Test Execution Date: 28-07-2022

Test Title: Verify Is post added by user successfully

Test Case Description: Test objective is to ensure that post is added with all the require details and information.

Pre-conditions/Pre-requisites:

1. User must login
2. All credentials of post should be complete first.

Dependencies: Login

Steps	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Select Option Lost	Add missing person	selected option Lost	Selection of the option Lost	pass	
2	Fill all required fields.	Name	The field must be fill and picture must be attached	The field are filled correctly	Pass	No fields left empty
3	Attach Picture	Attach picture	Picture must attach	Picture attached	Pass	
4	Click on the upload button		All data must be verified and check neither of the field is empty	All credentials are provided	pass	
5	Post added		Post added	Post added in the timeline at missing person side	pass	
					All data added is correct so test the will be pass	

Post Conditions: Picture and Name is stored in database

Results with different data:

Test case no.	Test title	Test data	Test result	note
---------------	------------	-----------	-------------	------

1	Normal behavior	Name and picture	Name and picture uploaded successfully	-
2	Pre-conditions violation test	Name = null Picture uploaded	Failed to store the information	-
3	Pre-conditions violation test	Name given Picture not uploaded	Failed to store information	-

Report Missing Persons

Name:

Browse...

No file selected.

Upload

Figure 5.6 Report Missing Persons page

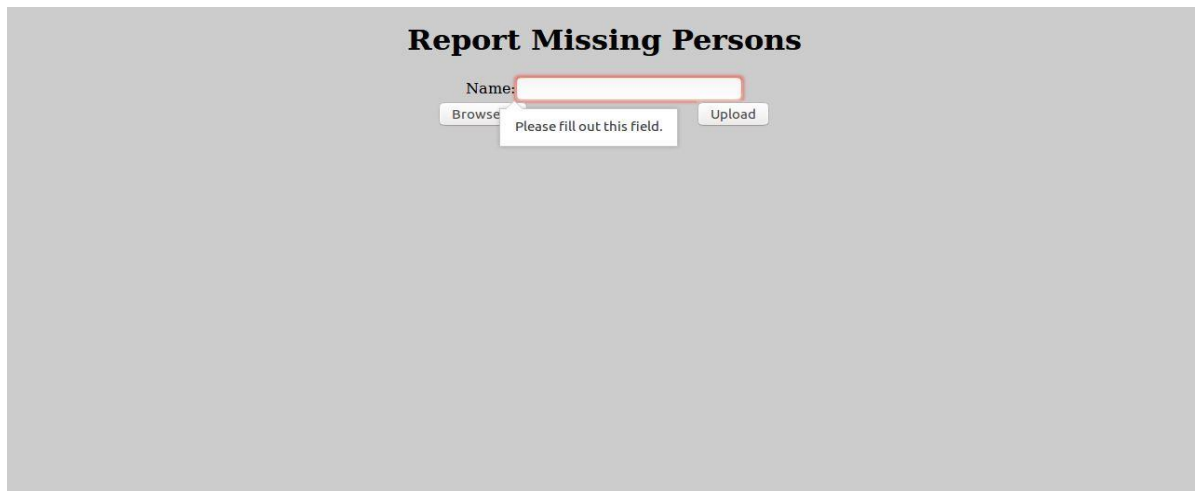


Figure 5.7 Report Missing Persons page shows no name error

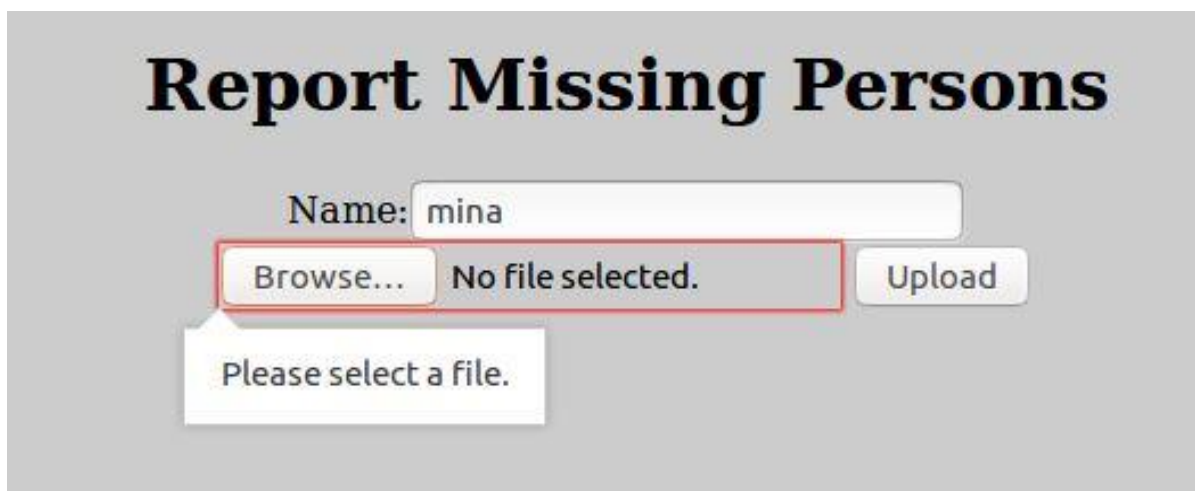


Figure 5.8 report missing page shows to select a file

Notify user

Test Case ID: TUK_US_FR16

Function tested: FR16.1/FR16.2 view notification

Product/Version/Module:

Product name: Ukraine Finder

Test Priority (Low/Medium/High): High

Test Case Version: TUK_US_FR16

Use Case References: Notify user

Test Design Date: 26-07-2022

Test Execution Date: 28-07-2022

Test Title: System will notify user by text message

Test Case Description: Test is user notify after image will matched

Pre-conditions/Pre-requisites:

- Missing and found Picture must be in Database

Dependencies: Picture should be clear

Steps	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	System will auto fetch all the picture and compare	Pictures in missing and found section	Same image will match	Same image matched	PASS	
2	System will take both users numbers who have posted the pictures	Text message that notifies user missing person is found and will send the phone number of user who found that person	Message deliver to the user	Message received by the user successfully	pass	
3	A notification is sent		Notification will be sent	Notification sent	pass	
					Test pass	

Post Conditions:

Matched image post must be deleted

Results with different data:

Test case no.	Test title	Test data	Test result	note
1	Normal behavior	Upload picture	Match found or not found	



Figure 5.9 notification of match found

Image Matching

Test Case ID: TUK_FR16

Function tested: FR16.1/FR16.2

Product/Version/Module:

Product name: Ukraine finder

Test Priority (Low/Medium/High): High

Test Case Version: TUK_FR16

Use Case References: Image Matching

Test Design Date: 26-07-2022

Test Execution Date: 28-07-2022

Test Title: System will match image

Test Case Description: Test any image in missing section will match with picture in found section

Pre-conditions/Pre-requisites:

- Missing and found Picture must be in Database

Dependencies: Picture must of human being

Steps	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
-------	------------	-----------	-----------------	---------------	--------------------	-------

1	Select the Find option					
2	Add same persons picture already inserted in the Lost option	Picture in Lost section	Picture in Database	Picture in database	Pass	
3	Fill the form		Photo inserted	Photo inserted		
4	Click on button search		Data is passed to Image matching	Data are passed to image matching		
5	System will automatically find a match		Same image will match	Same image matched	PASS	
					Test pass	

Post Conditions:

Return match status

Results with different data:

Test case no.	Test title	Test data	Test result	note
1	Normal behavior	Upload picture and hit search	Found or not found	-
2	Pre-conditions violation test	Not uploading a picture	Error	-

Admin Sign in with Email password

Test Case ID: TUK_FR19

Function tested: FR19.1/FR19.2 (admin Sign In)

Product/Version/Module:

Product name: Ukraine finder

Test Priority (Low/Medium/High): High

Test Case Version: TUK_FR19

Use Case References: ID 019 Admin Sign In

Test Design Date: 26-07-2022

Test Execution Date: 28-07-2022

Test Title: Admin Sign In

Test Case Description: Test the Sign in Functionality.

Pre-conditions/Pre-requisites:

- The user must have an account on Ukraine finder.

Dependencies: Sign Up

Steps	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1.	Enter Sign in Credentials	Username Password	Admin must be able to enter Sign in credentials		pass	On apge Dgango administration Username: admin Password: admin
2	Click on Sign in button.		Admin is Signed In	Admin is signed In.	pass	Click on Sign in button.
2	Validate Sign in Credentials			Validated Email Address and password	Pass	
3	Verify User's Sign In Credentials		Verification is successful	Admin can Sign In.	pass	
4	Admin redirect to		Admin redirect to	Admin redirect to	pass	Admin page appeared

	home page		home page	home page		
					pass	

Post Conditions:

- The Sign in button changes its text to Sign out when user is signed in.

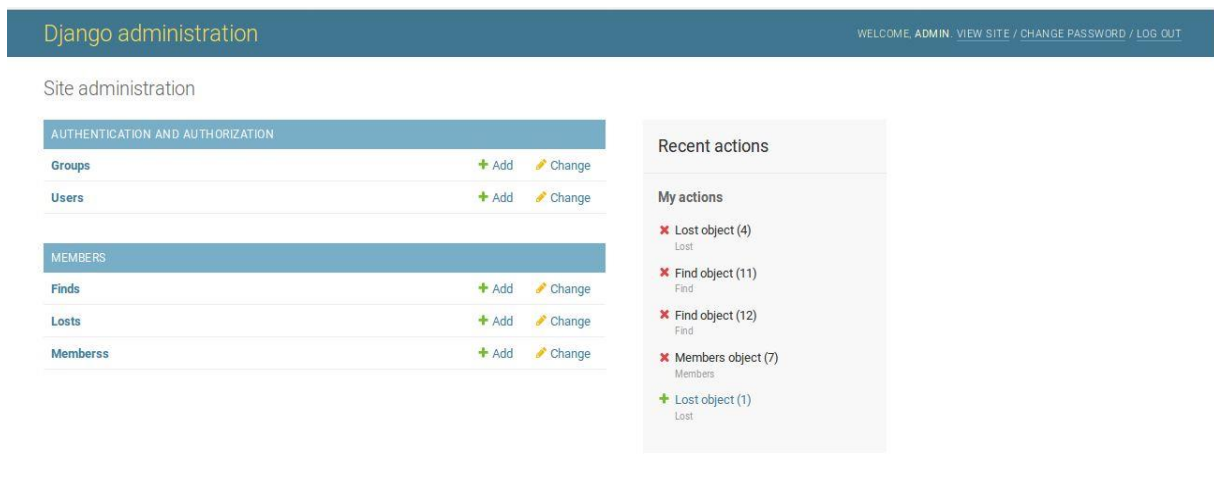
Results with different data:

Test case no.	Test title	Test data	Test result	note
1	Normal behavior	userName = 'admin' password = 'admin'	Login success	-
2	Pre-conditions violation test	userName = null password= null	Not login	-
3	Pre-conditions violation test	userName = wrong, or password= wrong	Not login	-



The image shows the Django administration login page. It has a dark blue header with the text "Django administration". Below the header, there are two input fields: "Username:" with the value "admin" and "Password:" with masked characters "*****". A blue "Log in" button is centered below the password field.

Figure 5.10 Admin Sign in page



The image shows the Django administration home page. The header is dark blue with "Django administration" on the left and "WELCOME, ADMIN. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)" on the right. Below the header, the page is divided into two main sections. The left section, titled "Site administration", contains two tables. The first table, "AUTHENTICATION AND AUTHORIZATION", has rows for "Groups" and "Users", each with "Add" and "Change" links. The second table, "MEMBERS", has rows for "Finds", "Losts", and "Members", each with "Add" and "Change" links. The right section, titled "Recent actions", contains a list of actions under the heading "My actions". The actions are: "Lost object (4) Lost", "Find object (11) Find", "Find object (12) Find", "Members object (7) Members", and "Lost object (1) Lost".

Figure 5.11 Admin home Page

Admin update Password

Test Case ID: TUK_FR20

Function tested: FR20.1

Product/Version/Module:

Product name: Ukraine finder

Test Priority (Low/Medium/High): High

Test Case Version: TUK_FR20

Use Case References: ID 020 Admin update password

Test Design Date: 26-07-2022

Test Execution Date: 28-07-2022

Test Title: Admin Sign In

Test Case Description: Test the Sign in Functionality.

Pre-conditions/Pre-requisites:

- The user must have an account on Ukraine finder.

Dependencies: Sign In

Steps	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Click change password on admin page	Password	Admin must be able to enter on change password page	Admin entered in change password page	pass	
2	Enter old password	Old password should be correct	Admin must enter old password correctly	Admin entered old password correctly	pass	
2	Enter new password	New password should follow the constraints		New password entered correctly	Pass	
3	Enter new password confirmation	It should be same as previous		Password is entered same/matched	pass	
4	Click on change password			Password changed	pass	
	result				pass	

Post Conditions:

- The Sign in button changes its text to Sign out when user is signed in.

Results with different data:

Test case no.	Test title	Test data	Test result	note
1	Normal behaviour	Enter old password Enter new password Confirm new password	Password changed	-

2	Pre-conditions violation test	Wrong old password	Password not changed	-
3	Conditions violation test	Not confirmation of entering new password	Password not changed	-

Figure 5.12 Admin Update Password page

Admin view post

Test Case ID: TUK_ FR08

Function tested: FR08

Product/Version/Module:

Product name: Ukraine Finder

Test Priority (Low/Medium/High): High

Test Case Version: TUK_ FR08

Use Case References: ID 021 Admin View post

Test Design Date: 26-07-2022

Test Execution Date: 28-07-2022

Test Title: Admin view all the detail description added to the post

Test Case Description: Test objective is to ensure that user can view post clearly

Pre-conditions/Pre-requisites:

Admin must login

Dependencies: Login

Steps	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Admin view all the post at timeline		All post appears in both found and missing section	Post is there	Pass	
2	Click on post at timeline		Post will expand and all the post details will appear	User able to view all detail of added post		
						Post viewed successfully

Post Conditions:

- Post should appear in the current timeline and in user's profile.

Admin Delete post**Test Case ID:** TUK_FR22**Function tested:** FR22.1 Admin Delete Post**Product/Version/Module:**

Product name: Ukraine finder

Test Priority (Low/Medium/High): High**Test Case Version:** TUK_FR22**Use Case References:** Admin Delete post**Test Design Date:** 26-07-2022**Test Execution Date:** 28-07-2022**Test Title:** Verify Is post Delete by admin successfully**Test Case Description:** Test objective is to ensure that post is delete**Pre-conditions/Pre-requisites:**

1. Admin must login

Dependencies: Login

Steps	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Click on delete post button		System asks really want to delete?	System asks really want to delete?		
2	Press Confirm deletion				pass	
						Post deleted successfully

Post Conditions: the post is deleted

Admin Delete user

Test Case ID: TUK_FR24

Function tested: FR24.1

Product/Version/Module:

Product name: Ukraine finder

Test Priority (Low/Medium/High): High

Test Case Version: TUK_FR24

Use Case References: Admin Delete post

Test Design Date: 26-07-2022

Test Execution Date: 28-07-2022

Test Title: Admin delete user successfully

Test Case Description: Test objective is to ensure that user is delete

Pre-conditions/Pre-requisites:

1. Admin must login

Dependencies: Login

Steps	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
-------	------------	-----------	-----------------	---------------	--------------------	-------

1	Click on delete user button		System asks really want to delete?	System asks really want to delete?		
2	Press Confirm deletion				pass	
					Test Pass	User deleted successfully

Post Conditions:

- User will not login with the email he/she has been deleted

Results with different data:

Test case no.	Test title	Test data	Test result	note
1	Normal behaviour	Delete user	User deleted	-
2	Pre-conditions violation test	Not selecting the user	User not deleted	-

Django administration WELCOME, ADMIN [VIEW SITE](#) [CHANGE PASSWORD](#) [LOG OUT](#)

[Home](#) > [Members](#) > [Members](#)

Select members to change ADD MEMBERS +

Action: Delete selected members Go 1 of 4 selected

<input type="checkbox"/>	ID	USERNAME	PASSWORD	EMAIL
<input type="checkbox"/>	5	khalid	6789	kh@gmail.com
<input type="checkbox"/>	4	gabriele	5678	gabriele@gmail.com
<input checked="" type="checkbox"/>	3	Test3	1234	Test3@gmail.com
<input type="checkbox"/>	1	Shoalb	1234	-

4 members

Figure 5.13 Admin Delete User Page

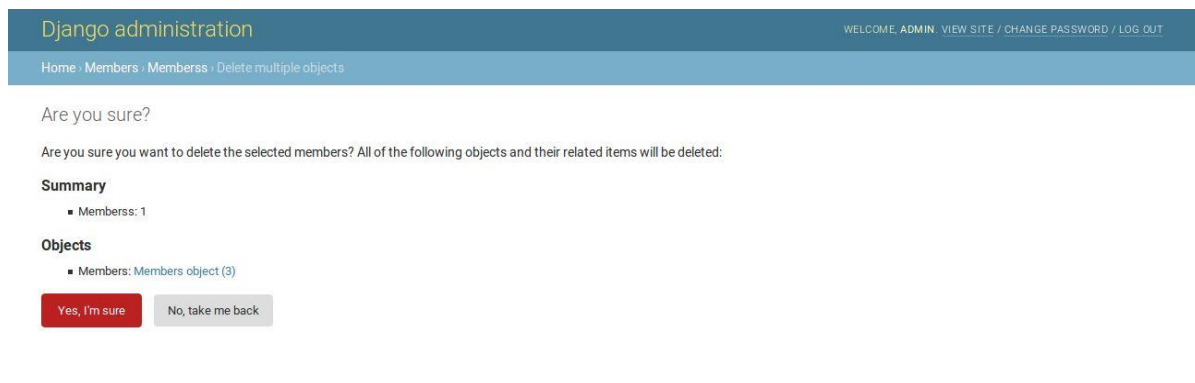


Figure 5.14 Admin Delete User asks to confirm the deletion

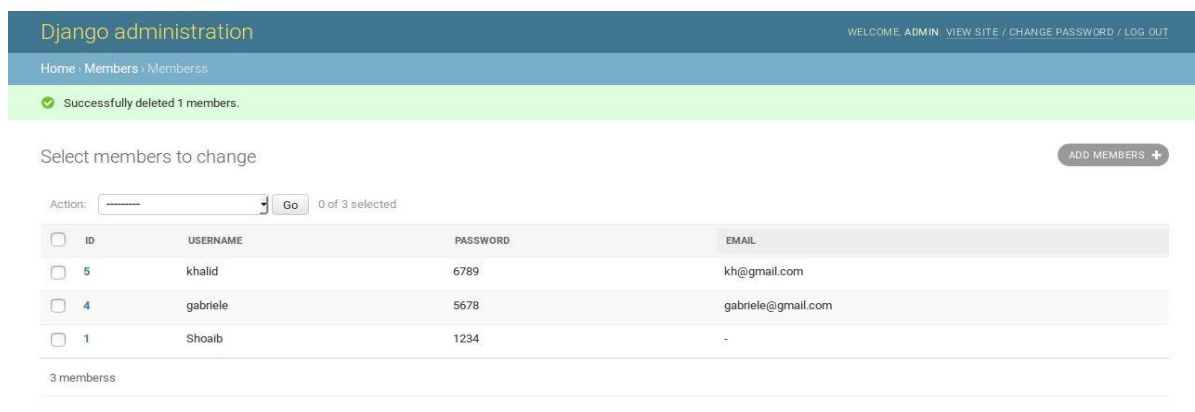


Figure 5.15 Step after the deletion of test3 User

Admin View/search user

Test Case ID: TUK_FR23

Function tested: FR23.1 admin search user

Product/Version/Module:

Product name: Ukraine finder

Test Priority (Low/Medium/High): High

Test Case Version: TUK_FR23

Use Case References: ID 023 Admin search user

Test Design Date: 26-07-2022

Test Execution Date: 28-07-2022

Test Title: Admin view all the details of the user

Test Case Description: Test objective is to ensure that Admin is able to view all user's data

Pre-conditions/Pre-requisites:

1. Admin must login

Dependencies: Login

Steps	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Admin click on view user button		All user data will appear	All user data will appear	Pass	
					Test Pass	

Post Conditions:

Nothing

Admin Add user

Test Case ID: TUK_FR24

Function tested: FR24.1

Product/Version/Module:

Product name: Ukraine finder

Test Priority (Low/Medium/High): High

Test Case Version: TUK_FR24

Use Case References: Admin Add user

Test Design Date: 26-07-2022

Test Execution Date: 28-07-2022

Test Title: Admin add user successfully

Test Case Description: Test objective is to ensure that user is added

Pre-conditions/Pre-requisites:

- Admin must login

Dependencies: Login and on admin page

Steps	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Click on add member/user button		System takes to fill details for new user	Same as expected	Pass	

2	Add username, password and email				pass	
3	result				Test Pass	User added successfully

Post Conditions:

- User will not login with the email he/she has been deleted

Results with different data:

Test case no.	Test title	Test data	Test result	note
1	Normal behavior	Fill all fields	User added	-
2	Pre-conditions violation test	Not filling all fields	User not added	-

Django administration
WELCOME, ADMIN · [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home · Members · Members · Add members

Add members

Username:

Password:

Email:

Figure 5.16 Admin add user page

Django administration WELCOME, ADMIN. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home » Members » Membersss » Add members

Add members

Username:

Password:

Email:

[Save and add another](#) [Save and continue editing](#) [SAVE](#)

Figure 5.17 Admin add user form filled for user test1

Django administration WELCOME, ADMIN. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home » Members » Membersss

✓ The members "Members object (8)" was added successfully.

Select members to change ADD MEMBERS +

Action: Go 0 of 4 selected

<input type="checkbox"/>	ID	USERNAME	PASSWORD	EMAIL
<input type="checkbox"/>	8	test1	1234	test1@gmail.com
<input type="checkbox"/>	5	khalid	6789	kh@gmail.com
<input type="checkbox"/>	4	gabriele	5678	gabriele@gmail.com
<input type="checkbox"/>	1	Shoalb	1234	-

4 membersss

Figure 5.18 Admin search user page, note test1 is added (user filled above)

Django administration WELCOME, ADMIN. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home » Members » Membersss » Add members

Add members

Please correct the errors below.

Username:

This field is required.

Password:

This field is required.

Email:

[Save and add another](#) [Save and continue editing](#) [SAVE](#)

Figure 5.19 Admin user page shows advertisements

Chapter 6 : User Manual

➤ Requirements

- Web browser: Firefox
- Visual studio with python environment
- Virtual machine image:
 - https://archive.org/download/DeepLearningVMForVirtualBox.tar/DeepLearningVM_for_VirtualBox.tar.gz
 - The virtual machine image has Ubuntu Linux Desktop 16.04 LTS 64-bit pre-installed with the following machine learning tools:
 - Python 3.5.
 - OpenCV 3.2 for Python 3 bindings.
 - Dlib 19.9 for Python 3 bindings.
 - TensorFlow 1.5 for Python 3 bindings.
 - Keras 2 for Python 3 bindings.
 - face_recognition for Python 3 bindings.

➤ Getting started

Go to: The code in the visual and in Terminal and type “python3 manage.py runserver”

And run the server by holding ctrl key on keyboard while pressing on this link
[“https://127.0.0.1:8000/”](https://127.0.0.1:8000/)

6.1 Account registration (Add member)

The account registration is compulsory before access Ukraine finder’s features.

Step 1: Insert username, password and email address and press submit to register

Step 2: If registered, enter the username and password to login.

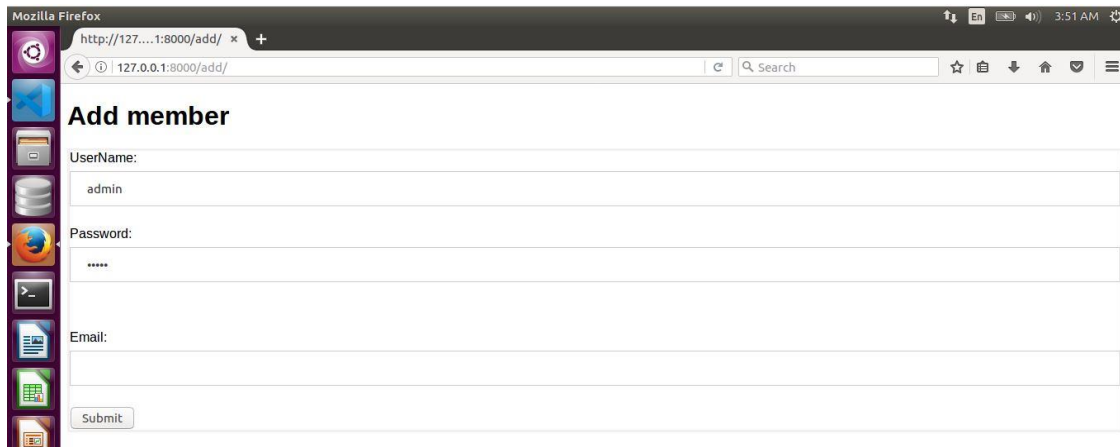
6.2 Lost: click on lost it will take you to the page of missing persons

Step 3: To publish a post about lost person write name and browse and then upload picture

6.3 Find click on find after login. It will take you to find missing persons page.

Step 4: Browse and upload the picture of missing person and click search

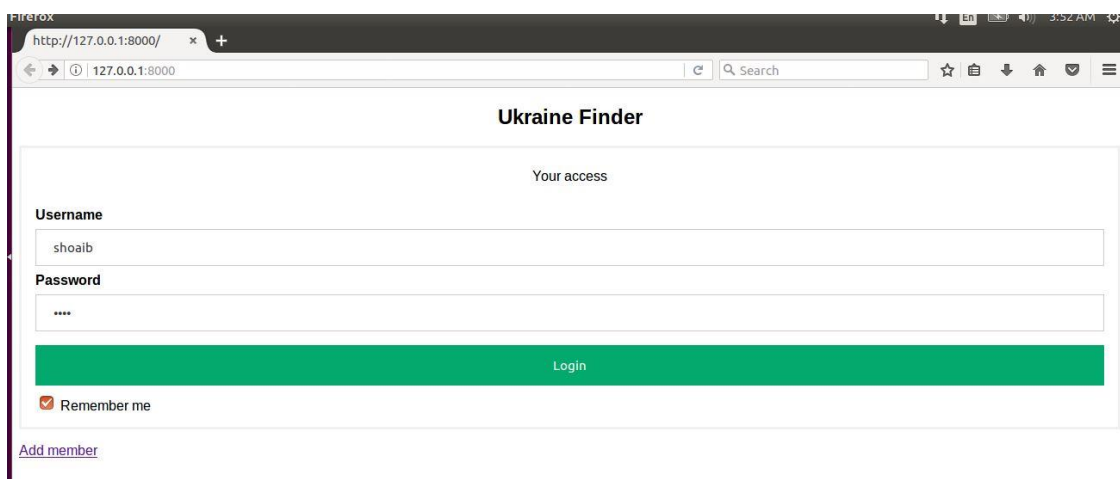
It will show you after some seconds that match has been found or not



The screenshot shows a web browser window with the address bar displaying `http://127.0.0.1:8000/add/`. The page title is "Add member". The form contains the following fields and elements:

- UserName:** A text input field with the value "admin".
- Password:** A password input field with masked characters "*****".
- Email:** An empty text input field.
- Submit:** A button at the bottom of the form.

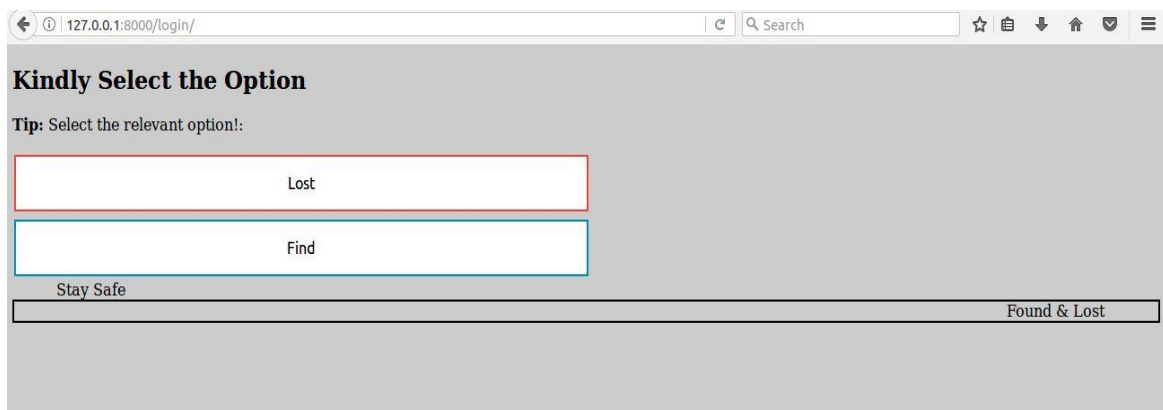
Step 1: Add member / Account registration



The screenshot shows a web browser window with the address bar displaying `http://127.0.0.1:8000/`. The page title is "Ukraine Finder". The page content includes:

- Your access:** A section header.
- Username:** A text input field with the value "shoaib".
- Password:** A password input field with masked characters "****".
- Login:** A large green button.
- Remember me:** A checked checkbox.
- Add member:** A link at the bottom left.

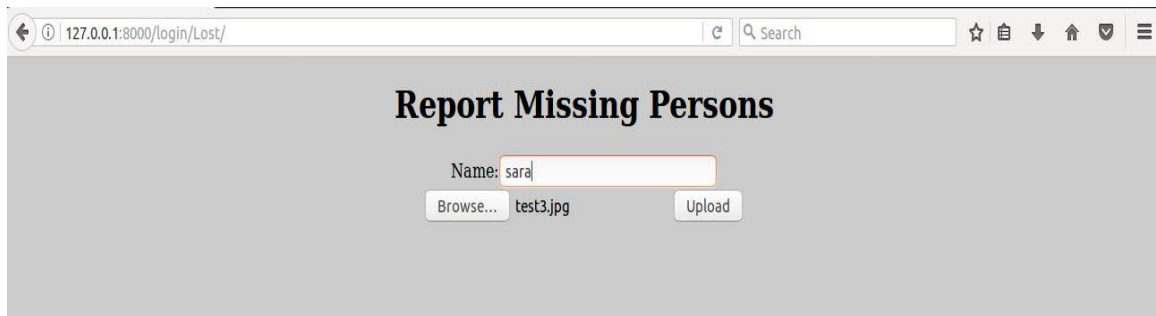
Step 2: Login



The screenshot shows a web browser window with the address bar displaying `http://127.0.0.1:8000/login/`. The page title is "Kindly Select the Option". The page content includes:

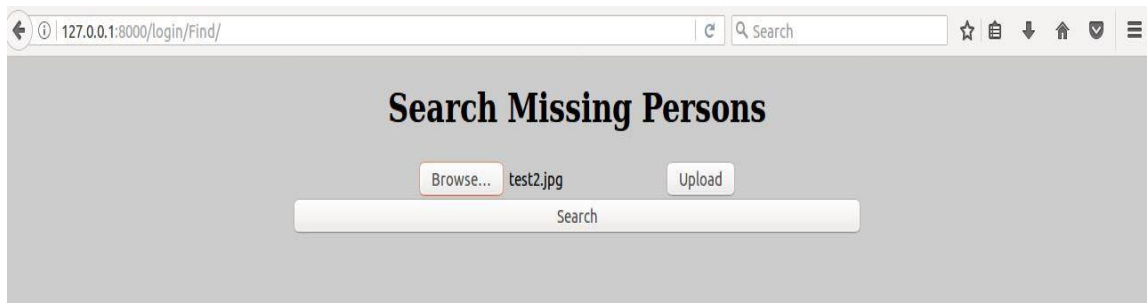
- Tip:** Select the relevant option!:
- Lost:** A button with a red border.
- Find:** A button with a blue border.
- Stay Safe:** A button at the bottom left.
- Found & Lost:** A button at the bottom right.

For lost select on lost



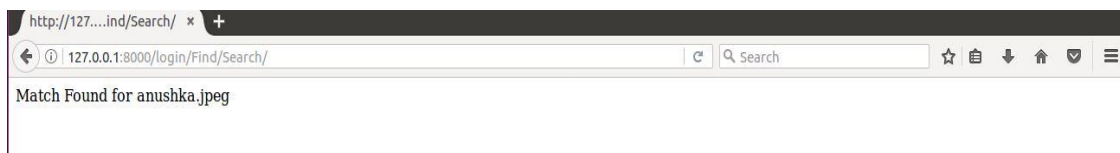
A web browser window showing a form titled "Report Missing Persons". The form has a text input field labeled "Name:" with the value "sara". Below the input field are two buttons: "Browse..." and "Upload". A file named "test3.jpg" is displayed between the two buttons.

Step 3 write name and upload picture



A web browser window showing a form titled "Search Missing Persons". The form has a "Browse..." button, a file named "test2.jpg", and an "Upload" button. Below these is a large "Search" button.

Step 4 to find someone upload a picture and click search



A web browser window showing the search results page. The address bar shows "http://127.0.0.1:8000/login/Find/Search/". The page displays the message "Match Found for anushka.jpeg".

Notification if missing person is found



A web browser window showing the search results page. The address bar shows "http://127.0.0.1:8000/login/Find/Search/". The page displays the message "Match not Found for test.jpeg".

Notification if person is not found

Chapter 7 : CONCLUSION

➤ Problem Faced

The feature of image matching differentiates our application from other available platforms in Ukraine. At this moment, it was difficult for us to choose the best image recognition library. Due to the current pandemic, we are unable to contact the Police department if they can give access to data on missing children in Ukraine.

➤ Project Summary

The aim of this project is to develop a solution which will enable people to find their lost ones by speeding up the process of finding them. The use of computer vision techniques of face recognition, image matching and face ageing will be used to make the process of locating a missing person quicker. An ordinary person can use the website to help find someone.

➤ Future Work

The project can be extended for accident cases and documents lost by people. Further, the system can be integrated with the police department database, which will make it more useful. Regarding the user experience, a modified version of the website is possible to be done considering the implementation of the undeveloped features mentioned in the [Actual and Future Developments](#) section.

Moreover, the design of the website is subject to be upgraded and utilized based on its usage and its significance. A mobile app could be developed to facilitate the utilization experience for the user and to hold swift usage and accessibility for everyone. A recurring optional survey could be asked by some individuals to enhance and soothe the process of using the website or the mobile application.

Chapter 8 : REFERENCES

- [1] Morvan, O. (2022). *Ukraine's forgotten missing and disappeared*. (ICMP Ukraine's Forgotten Missing and Disappeared) Retrieved from <https://www.icmp.int/news/ukraines-forgotten-missing-and-disappeared/>
- [2] Msehli, S., & Dillon, P. (2022, April 05). *7.1 million people displaced by the war in Ukraine: IOM survey*. Retrieved from International Organization for Migration: <https://www.iom.int/news/71-million-people-displaced-war-ukraine-iom-survey>
- [3] Missing Person Form
<https://forms.office.com/Pages/ResponsePage.aspx?id=K3EXCvNtXUKAjjCd8ope66TT3f7DJaRbiWaiN6A0RN1UOU4wWjIVWloxV0pKNDkxVzZLTjlxOERJSi4u>