Gabriele Ribolla
ID.10617369

Ukraine Finder

SE4A Project – 12 \ 08 \ 2022

Problem Statement

- Ukraine War
- Internet-based Technology

Feasability Survey

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?

More Details

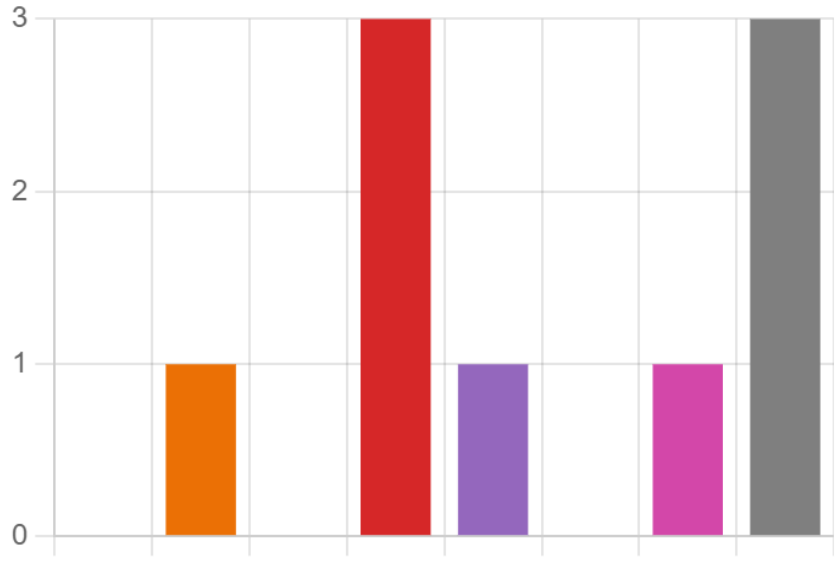
Yes	9
No	25

Feasability Survey

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

More Details

Mobile App	0
Web Application	1
Website	0
Police	3
social media	1
Other	0
more than one	1
Other	3



Feasability Survey

3. if a person is missing are you interested to (0 point)
find a person through ?
(*you can select more than 1 option*)

More Details

Social Media	24
Police	20
Mobile applications	8
web Sites	2
Non Goverment Organization (...	7

Feasability Survey

. Does the mobile/web application need to get missing person (0 point)
data from?
(*you can choose more than 1 option*)

More Details

Government	21
ONG/NGO	7
user	12

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

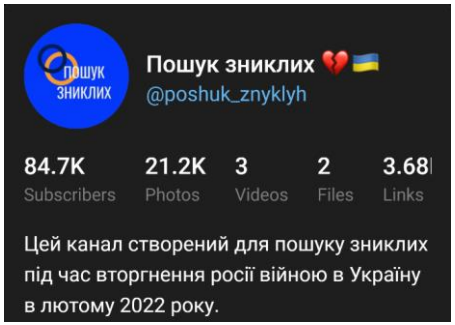
More Details

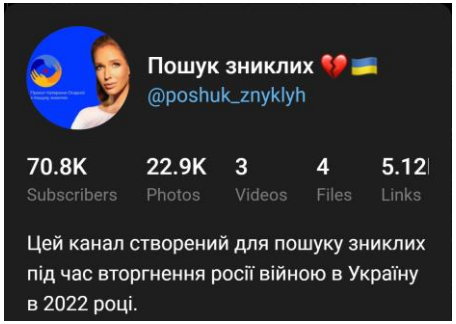
Medical help	26
Clothing help	14
Food help	12

Existing Solutions \ *Telegram channel*

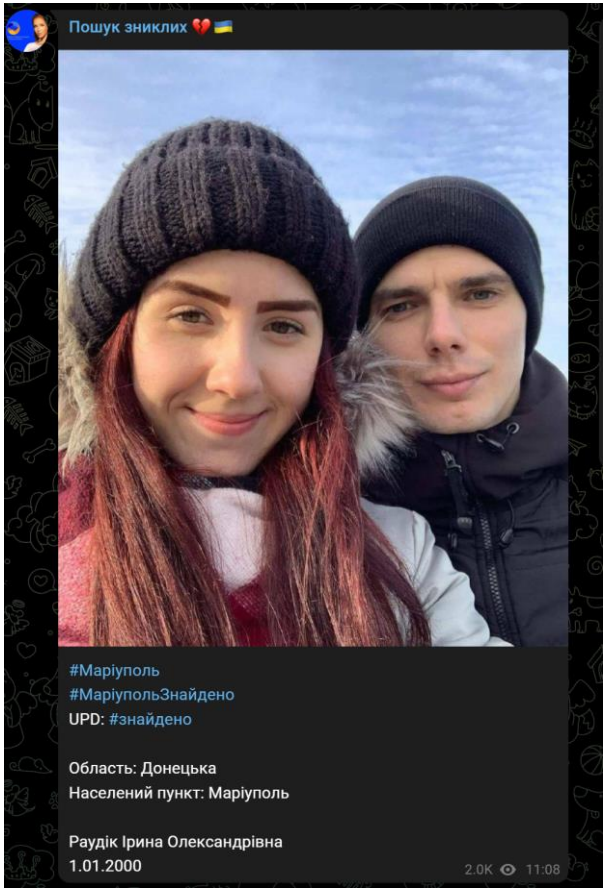
- Telegram channel: [Пошук зниклих @poshuk_znyklyh]
- Activity
- Cons

Early Months of War

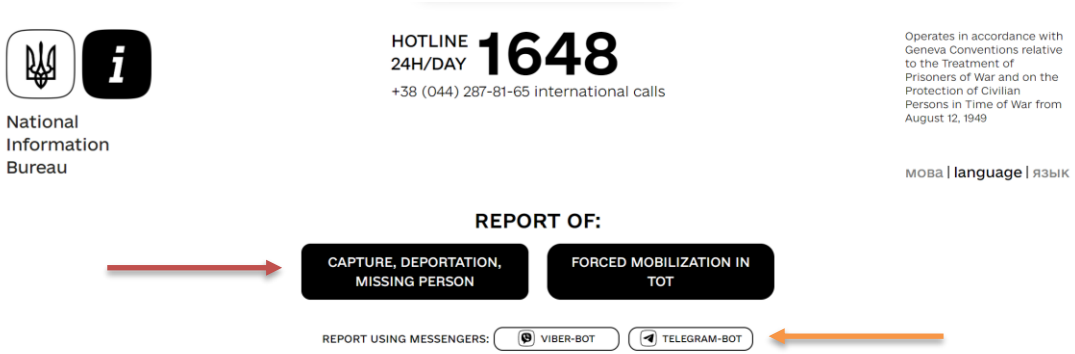


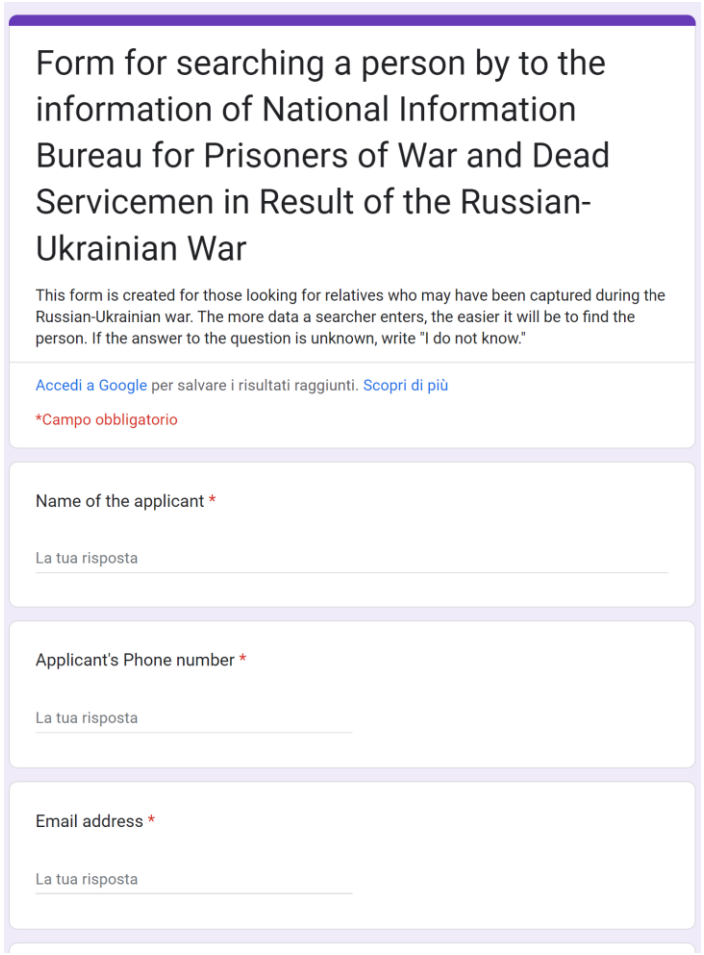


After Few Months



Existing Solutions \ *National Information Bureau*





- Website
- Cons

Existing Solutions \ *Police – Social Media*

- Police
- Instagram \ Facebook

Proposed Solution

MISSING PERSON

(Ukraine Finder)

- Introducing
- Objectives
  - Provide an assistance service
  - Speed up the searching process.
  - Collaborate with police and the responsible entities.
  - Introduce a new technique (Image Matching).

Comparison of our Product with Existing Solutions

Name	Searching Method	Effectiveness	Time
Ukraine finder	Uploading the photo and data	Effective due to the utilized technology and simplicity of use by everyone	Fast
Telegram	Send a message with photo and data	No, messages could be ignored	If the messages are ignored or not, not equal time for all
Police	Filling a report	Depends on the priority	Slow especially during wars
Instagram \ Facebook	Post with a photo and data	Can be effective, but not so much if we see practically	The time can vary from case to case
National Bureau	Filling online form	Mismatches Occurs	Slow

Type of Users

- Finder \ Helper

- Lost \ Affected (Missing Person)

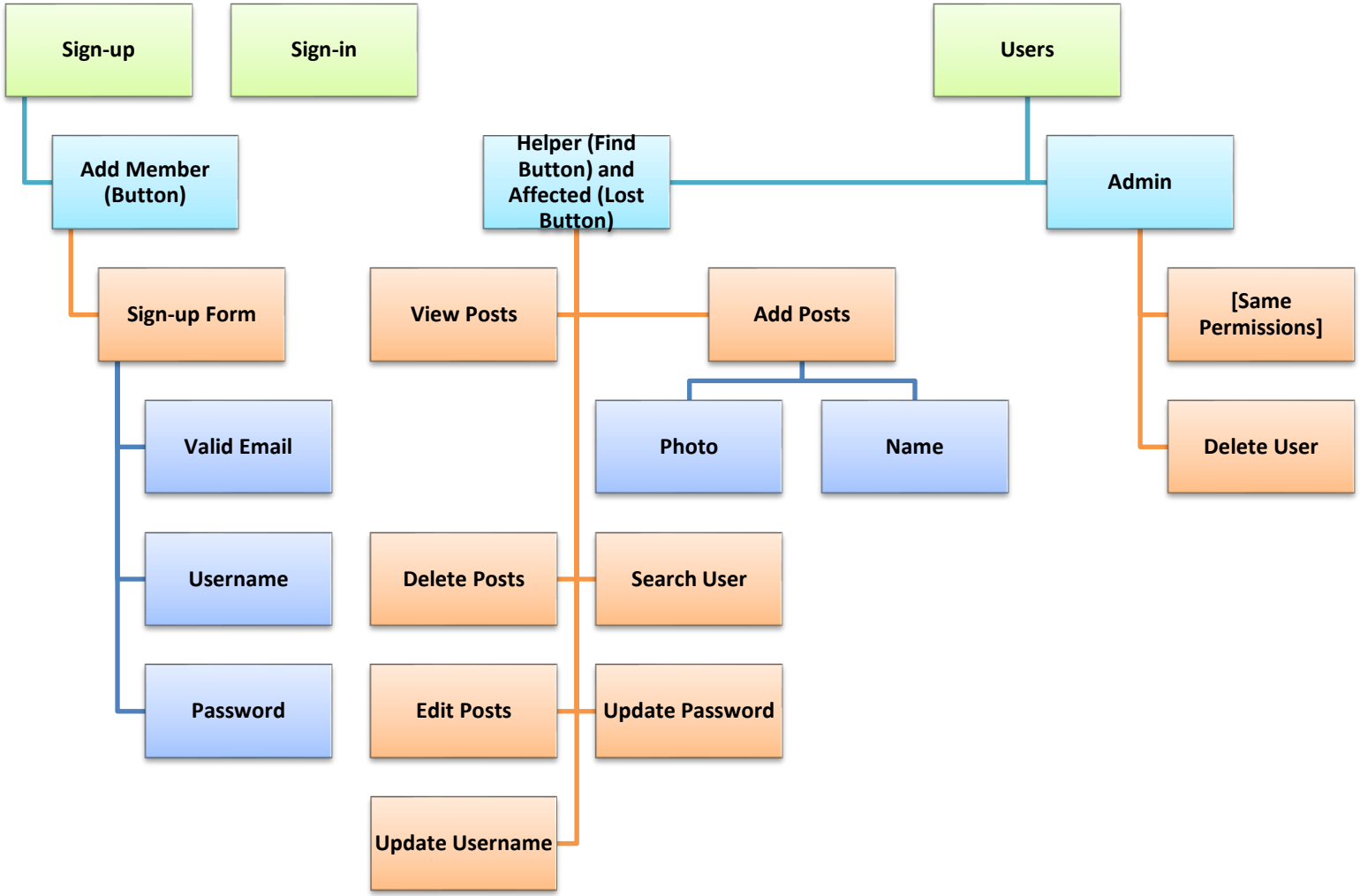
Assumptions and Constraints

- Assumptions
- Constraints

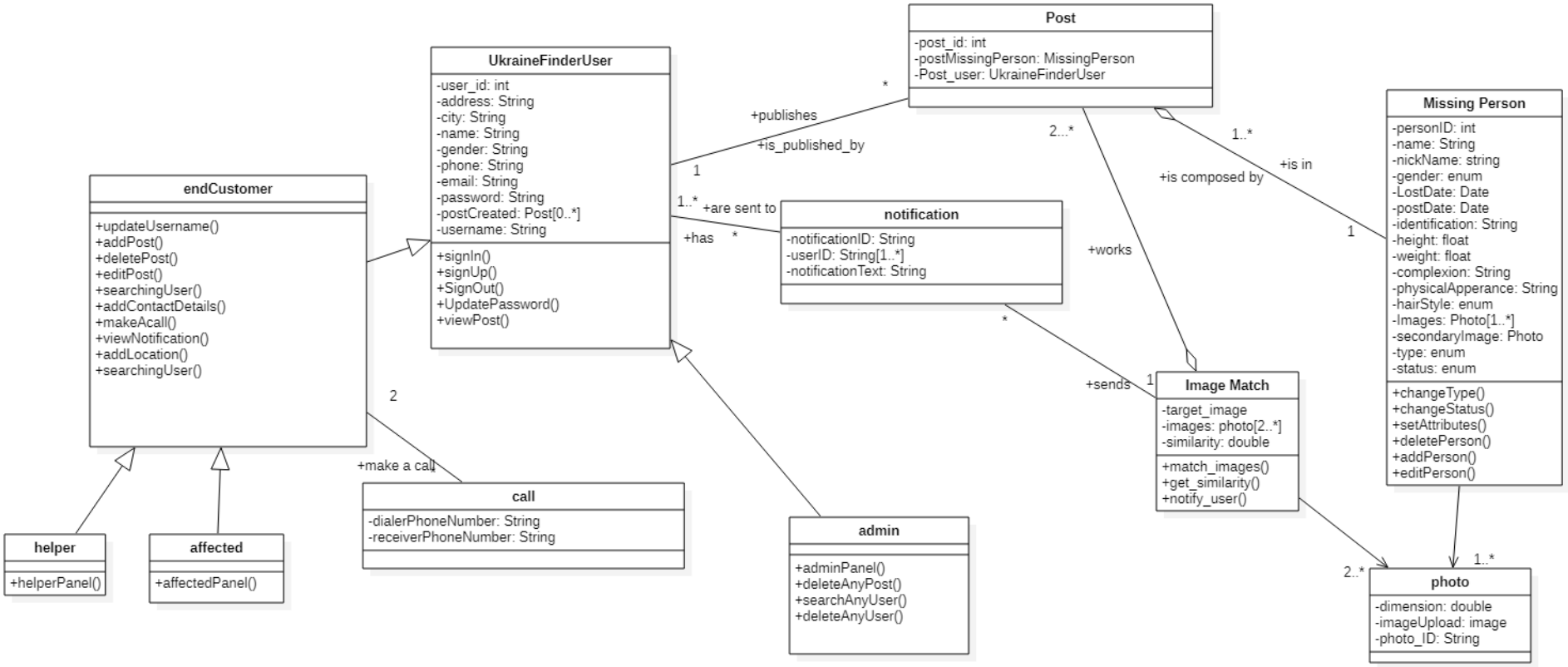
Requirements Analysis \ Requirements given by form’s responses

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

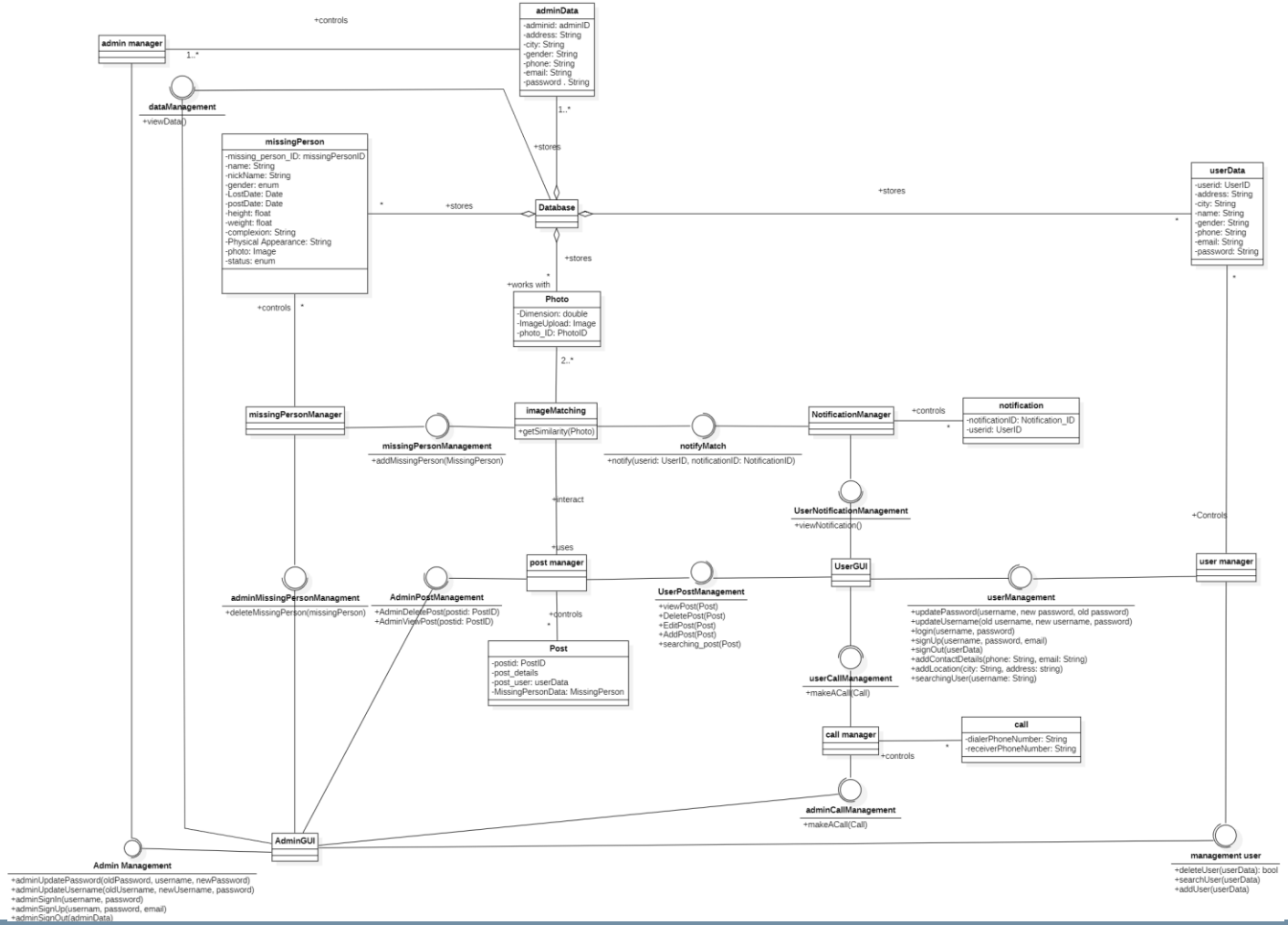
Requirements Analysis \ Description of the System



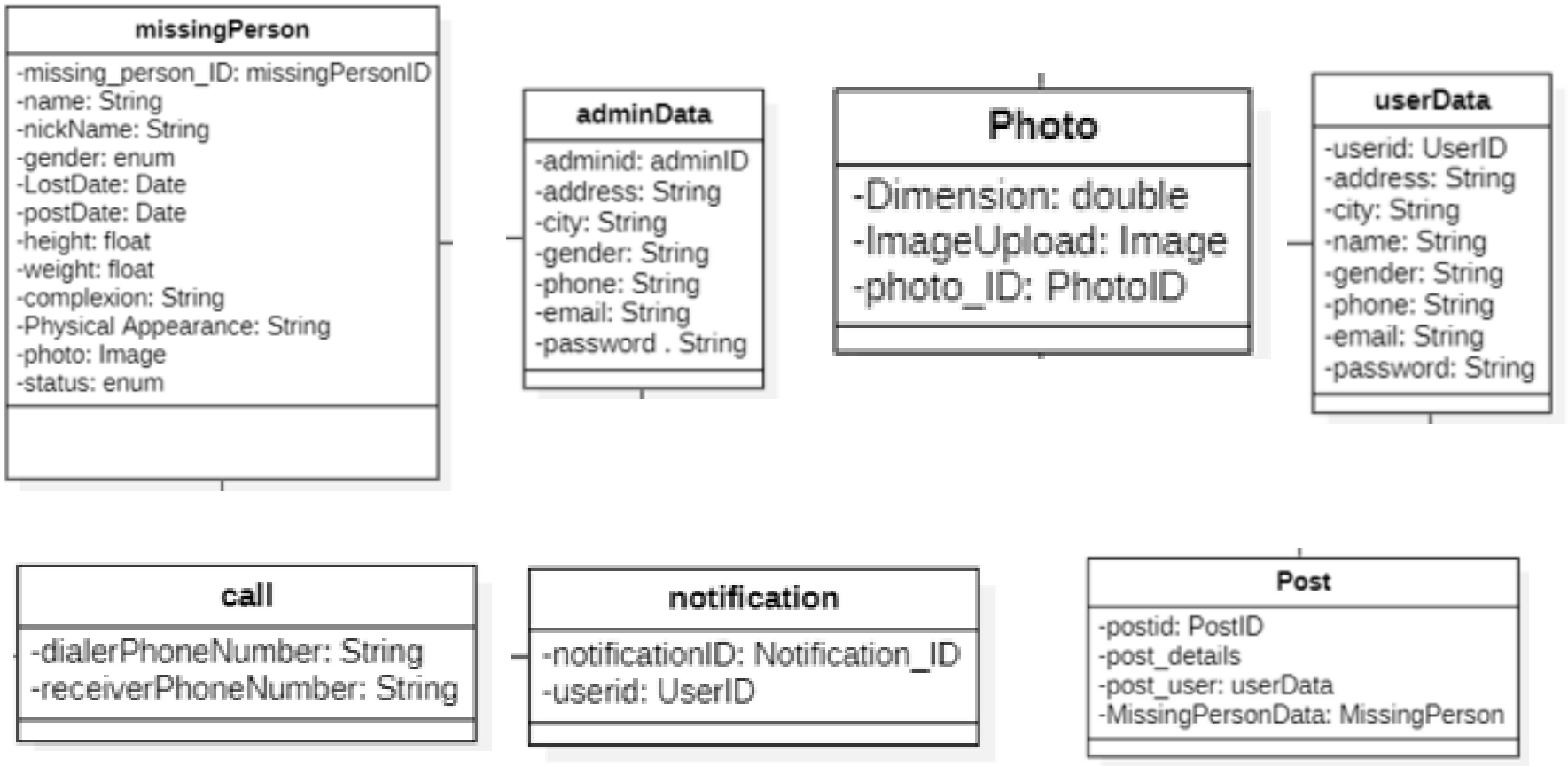
UML \ Specifications UML Class diagram



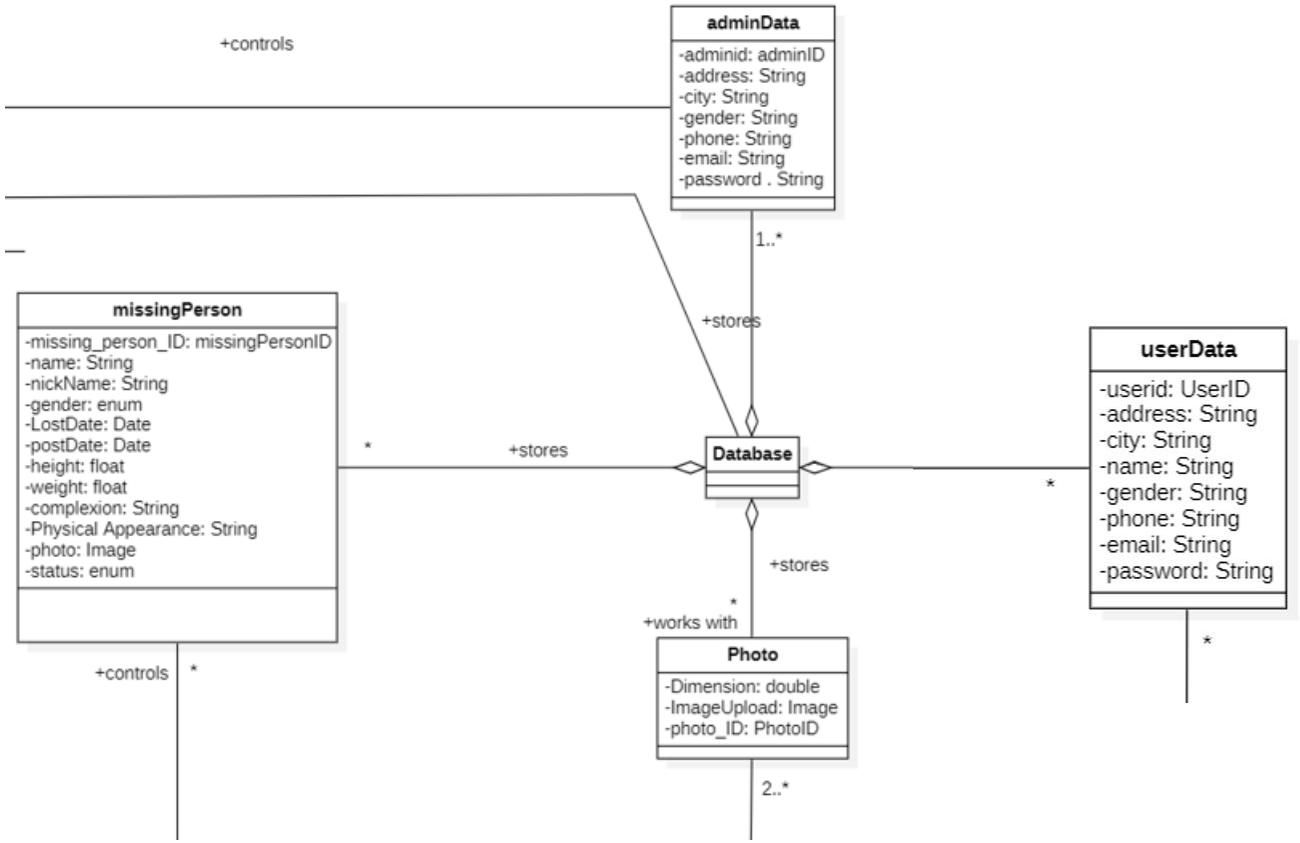
Software Design \ Interface UML



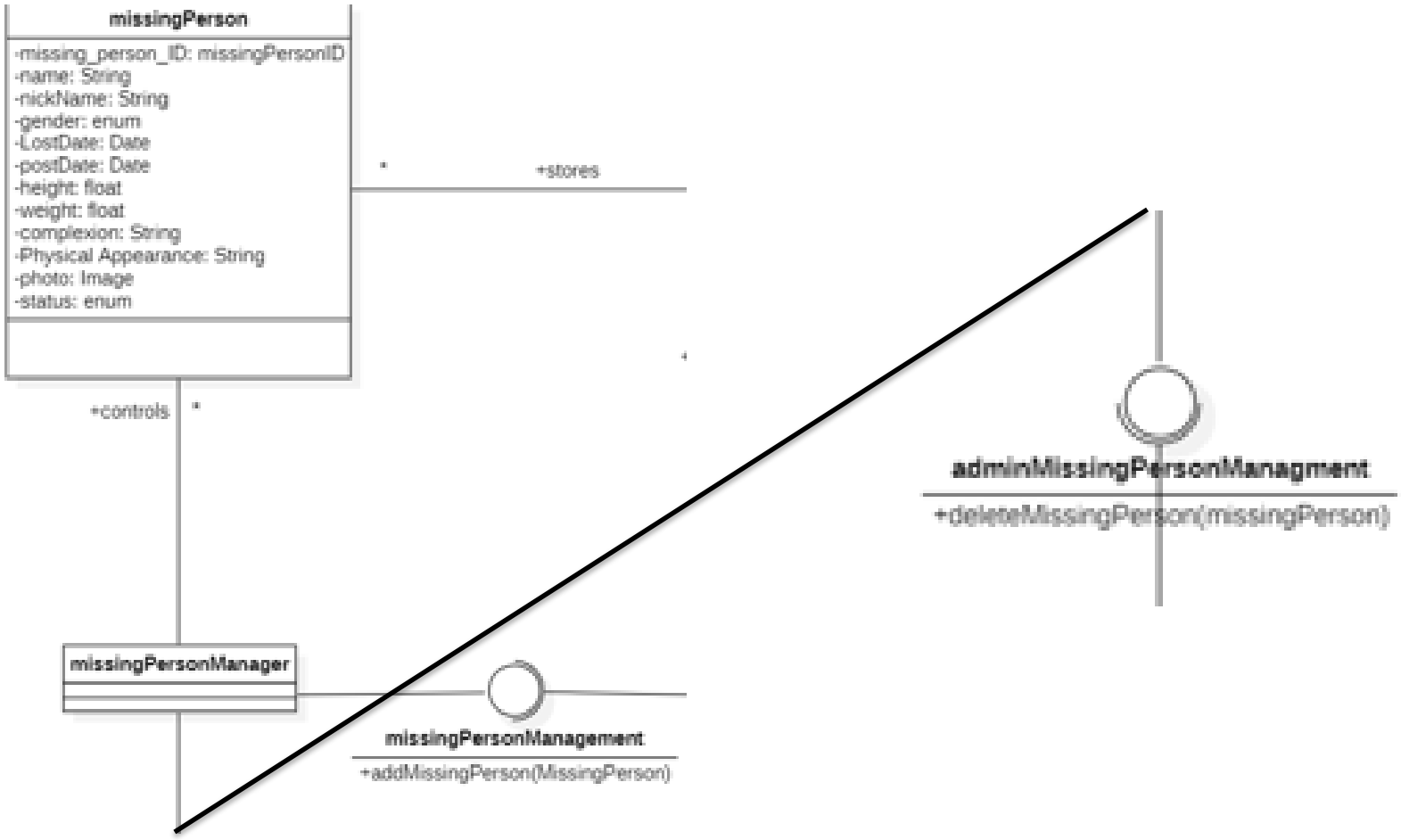
Software Design \ Interface UML \ Data module



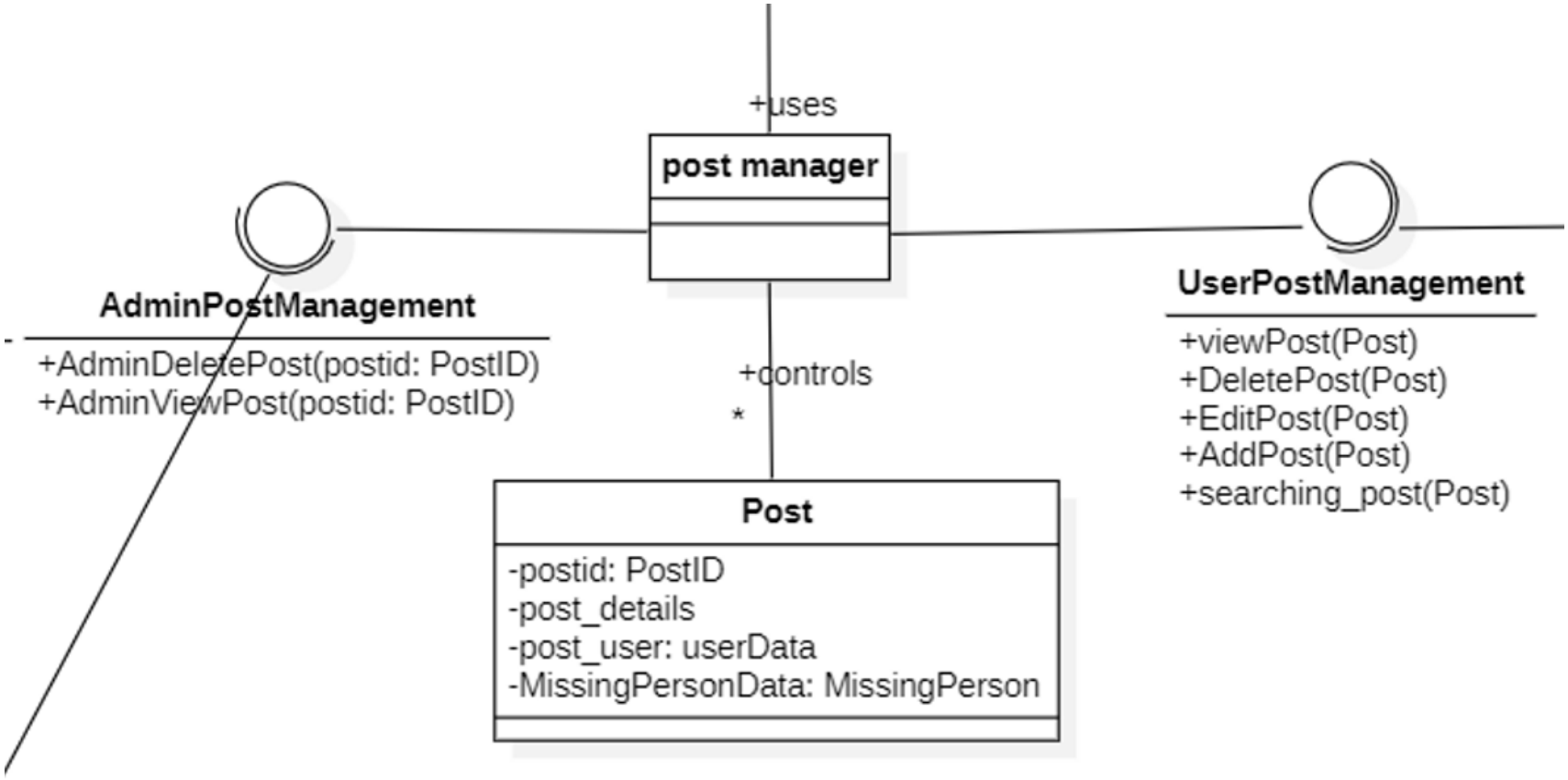
Software Design \ Interface UML \ Database



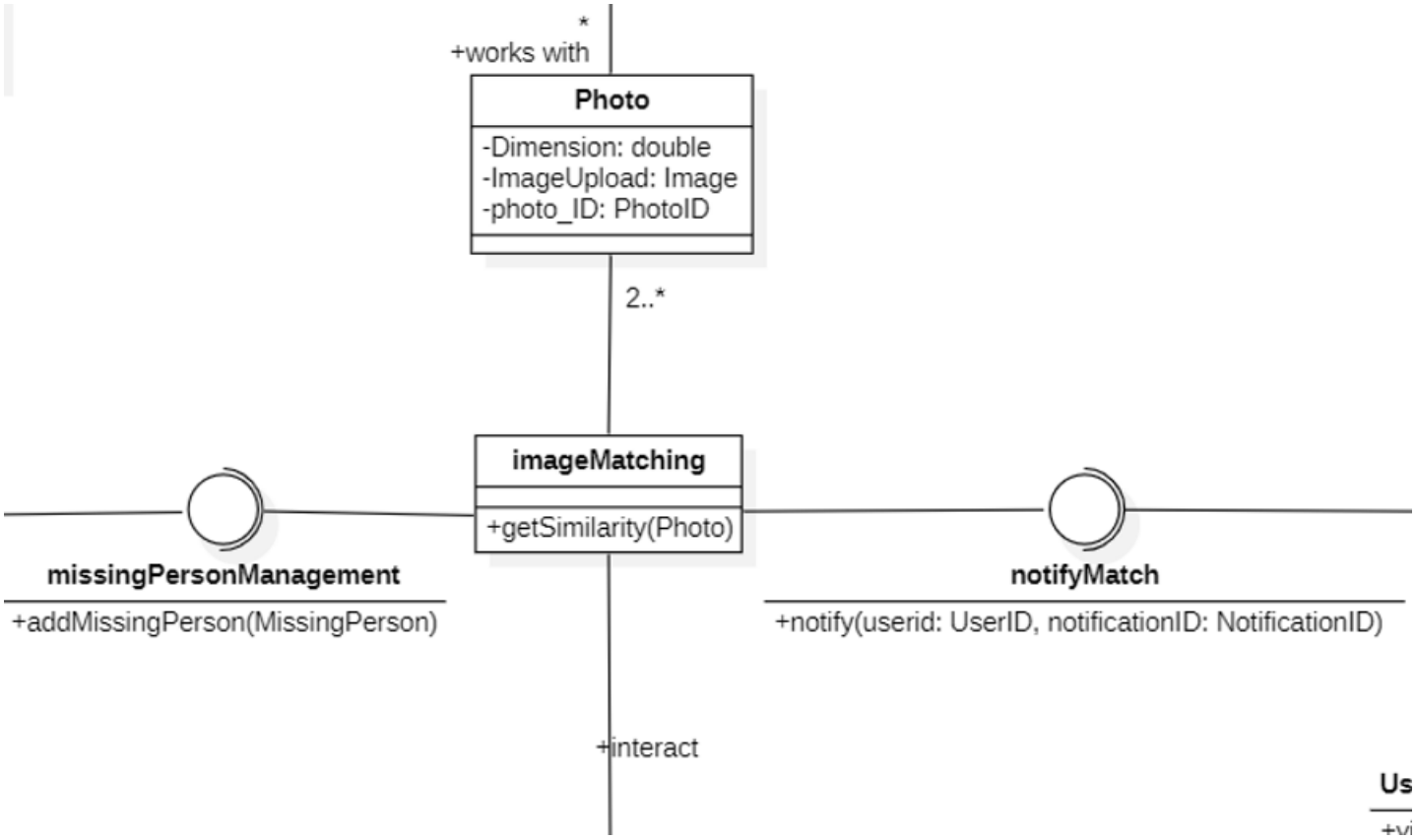
Software Design \ Interface UML \ missin person manager



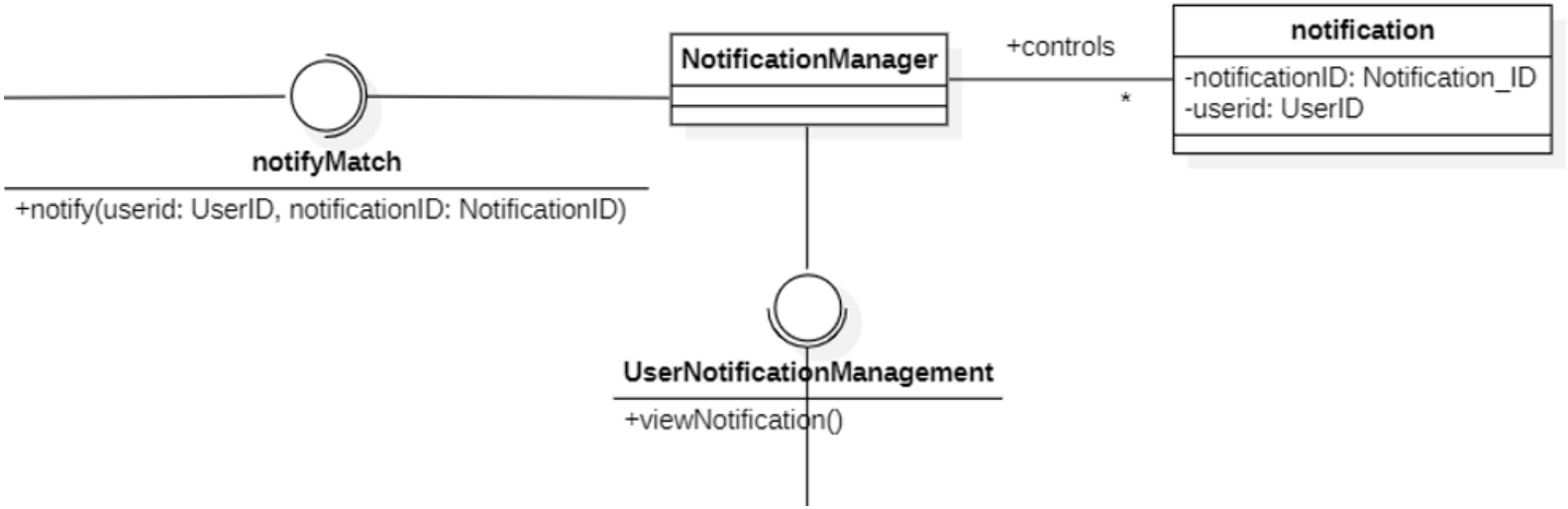
Software Design \ Interface UML \ post manager



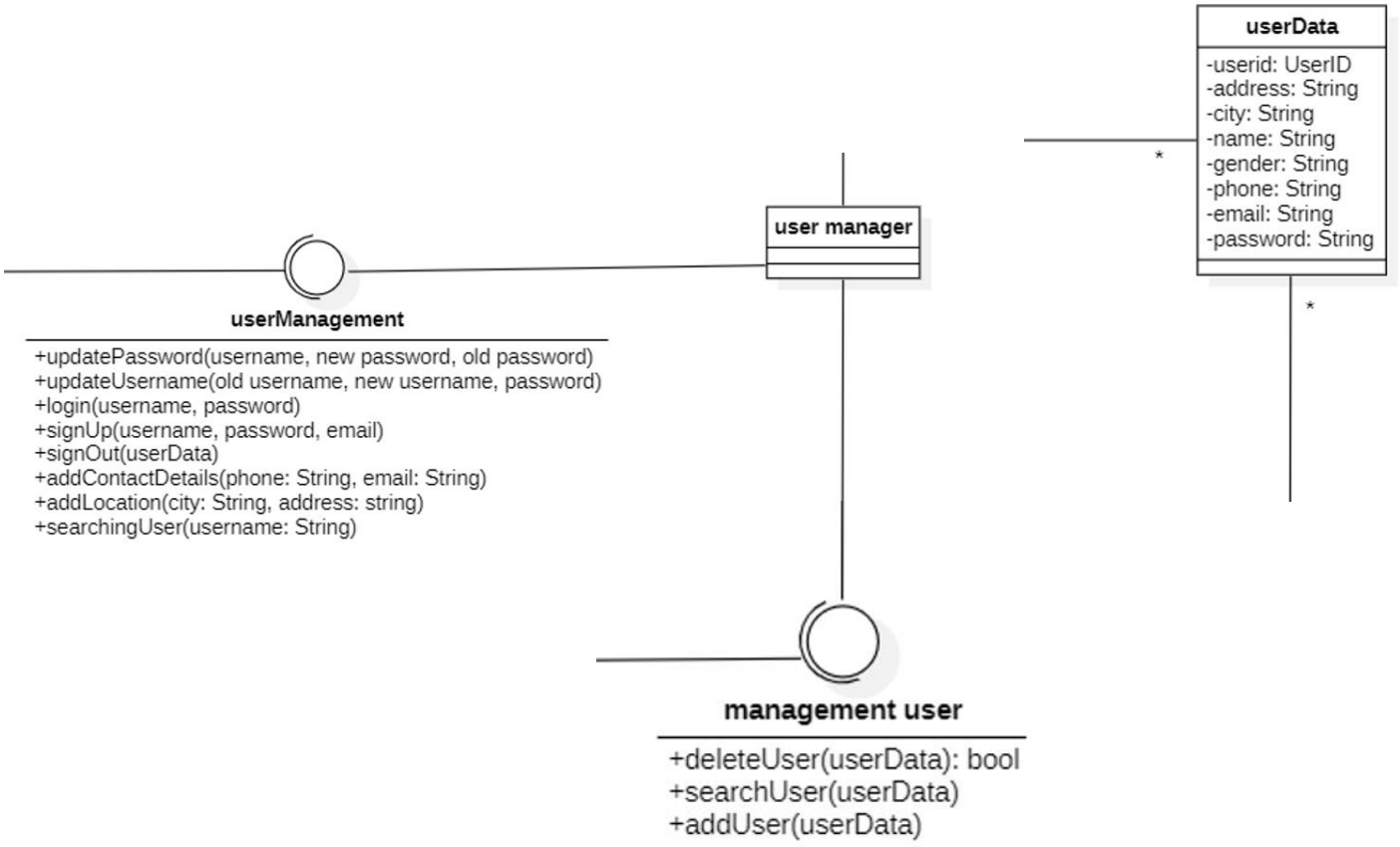
Software Design \ Interface UML \ image matching



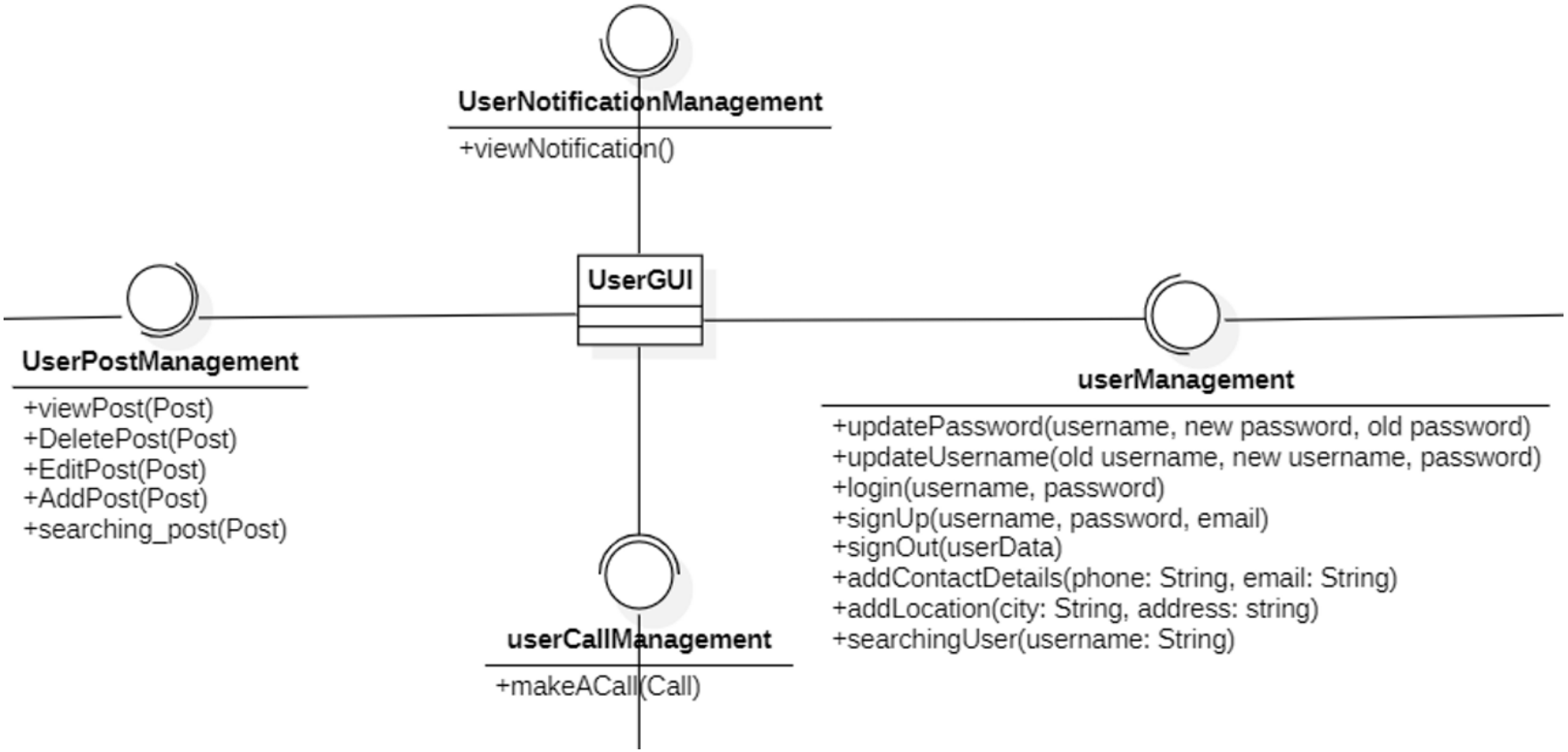
Software Design \ Interface UML \ notification manager



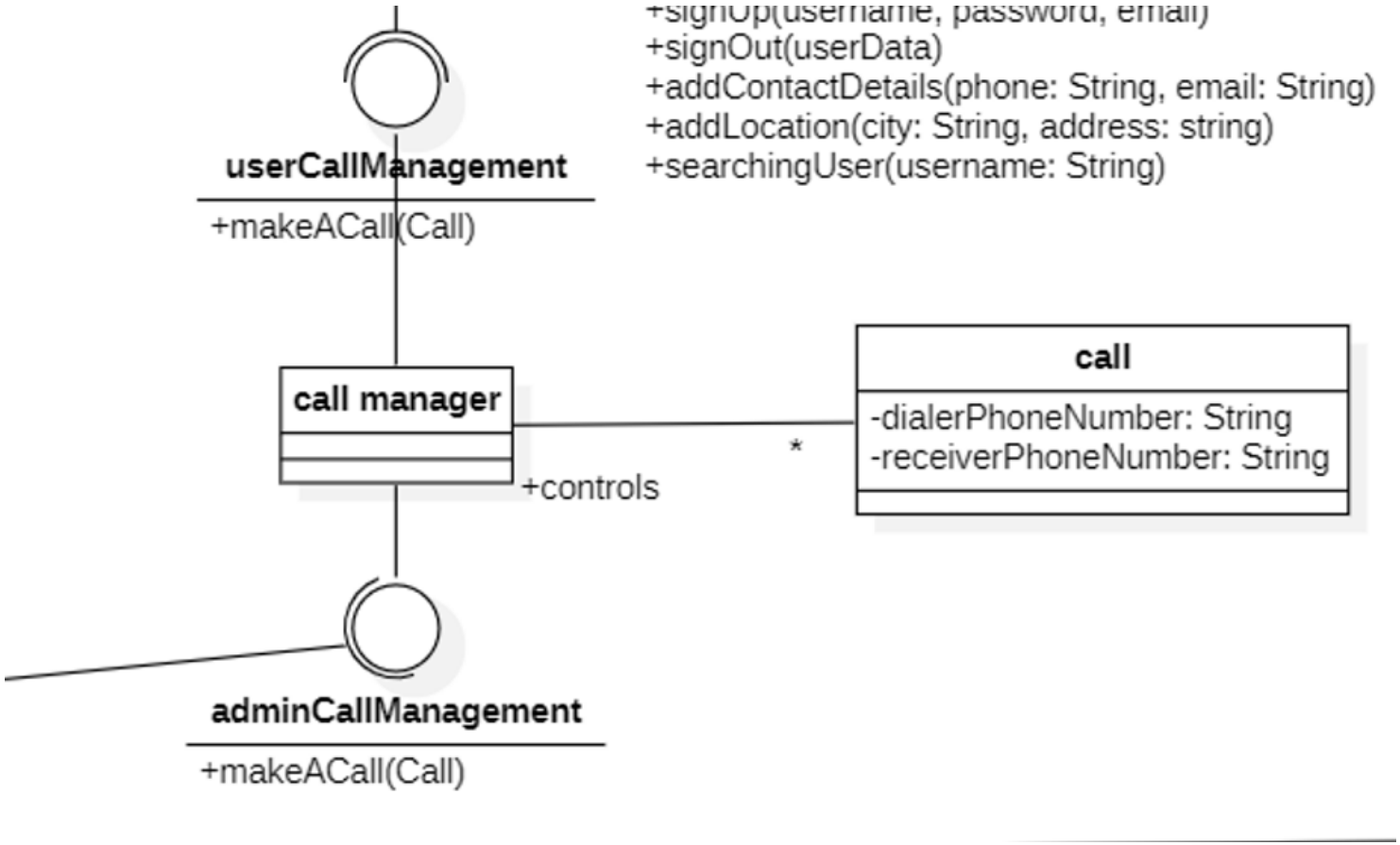
Software Design \ Interface UML \ user manager



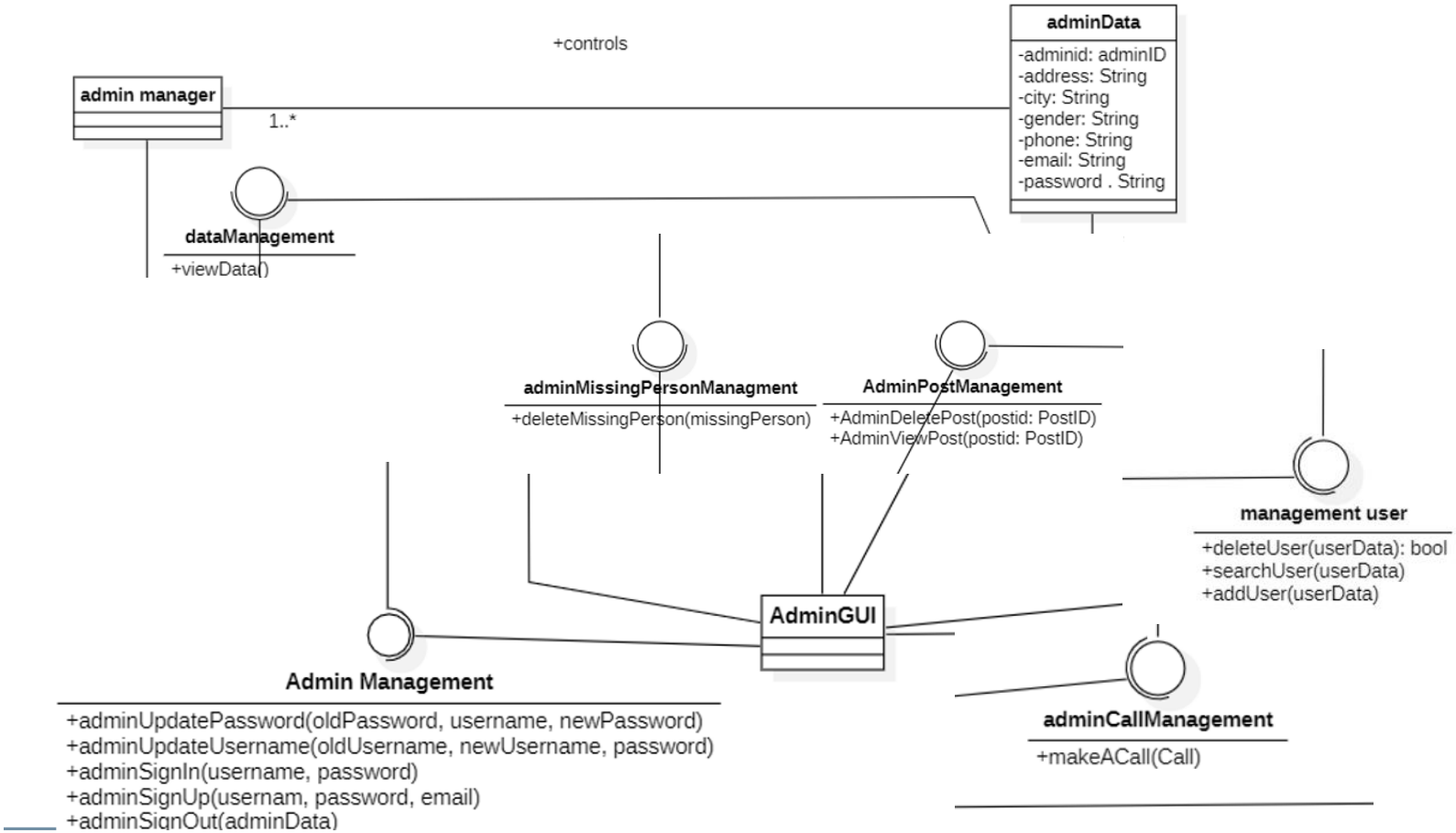
Software Design \ Interface UML \ userGUI



Software Design \ Interface UML \ Call Manager

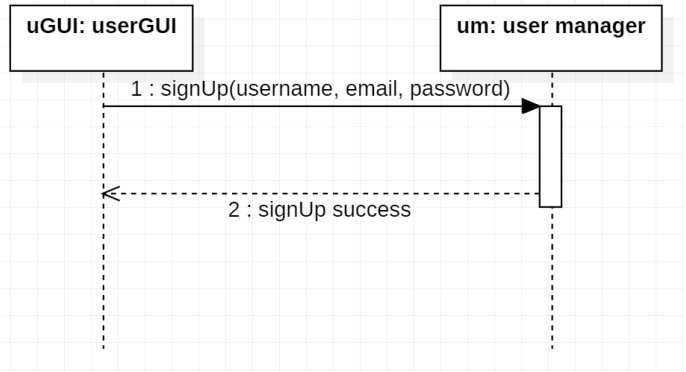


Software Design \ Interface UML \ admin GUI

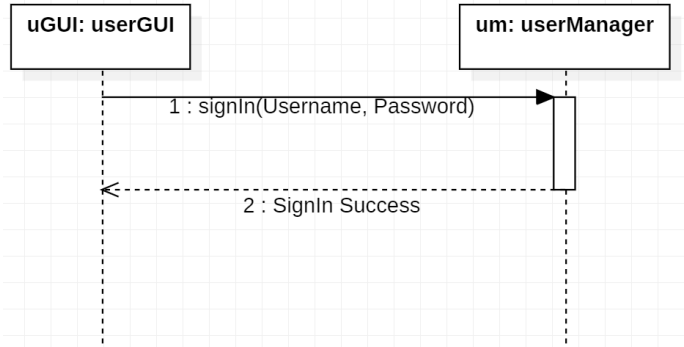


Software design \ *runtime-level sequence diagram* →

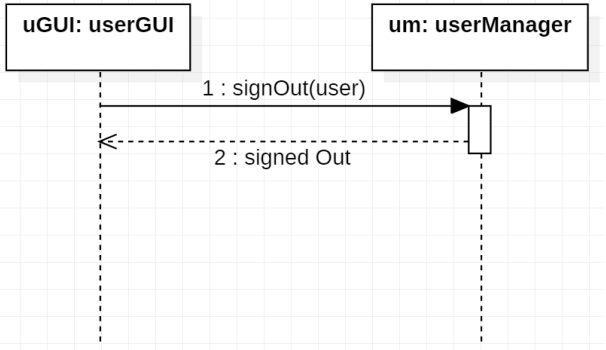
Sign Up (FR01.1)



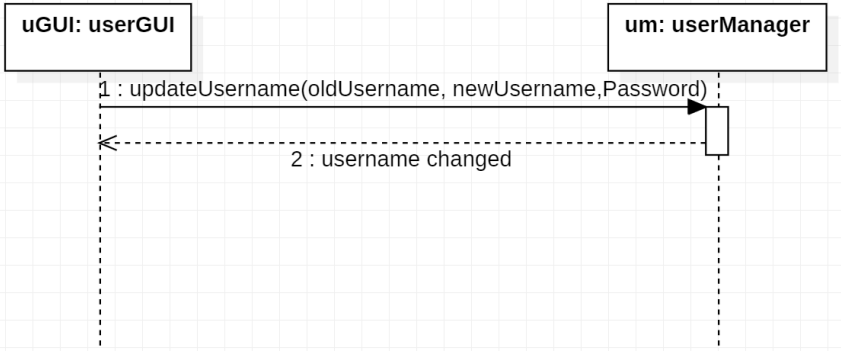
Sign In (FR02.1)



Sign Out (FR03.1)

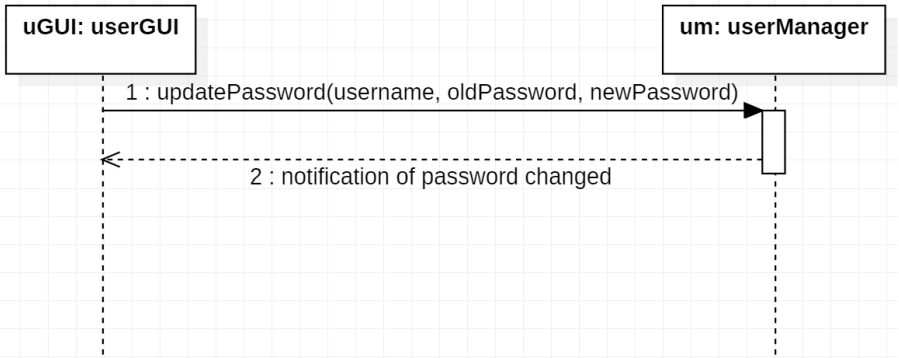


Update Username (FR04.1 \ FR04.2)

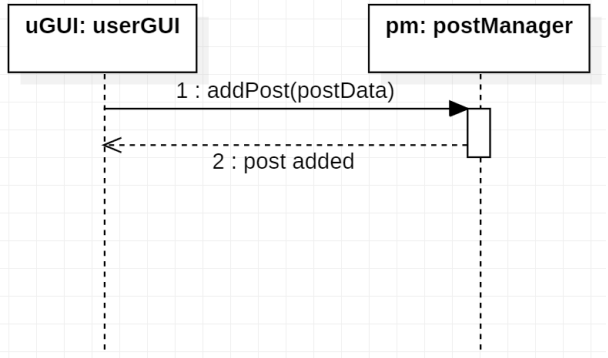


Software design \ *runtime-level sequence diagram* →

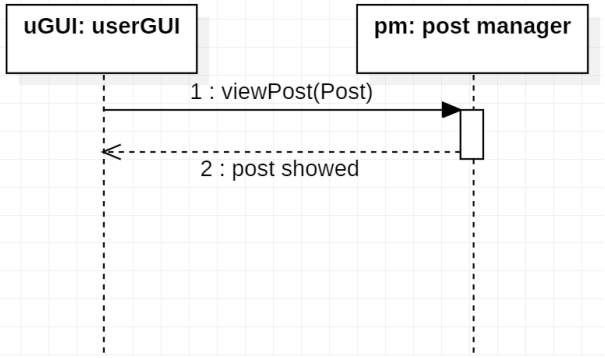
Update Password (FR06.1 \ FR06.2)



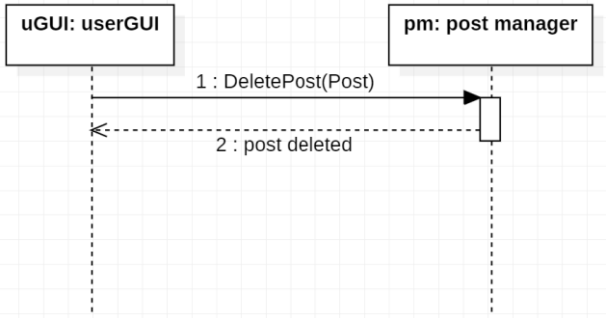
Add Post (FR07.1 \ FR07.2)



View Post (FR08.1)

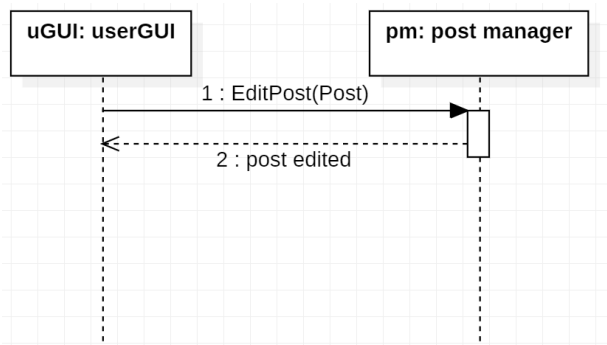


Delete Post (FR09.1)

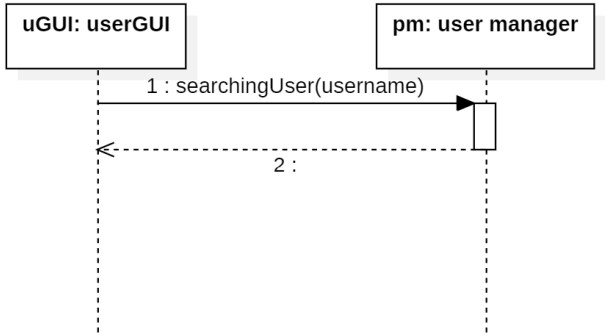


Software design \ *runtime-level sequence diagram* →

Edit Post (FR10.1 \ FR10.2)

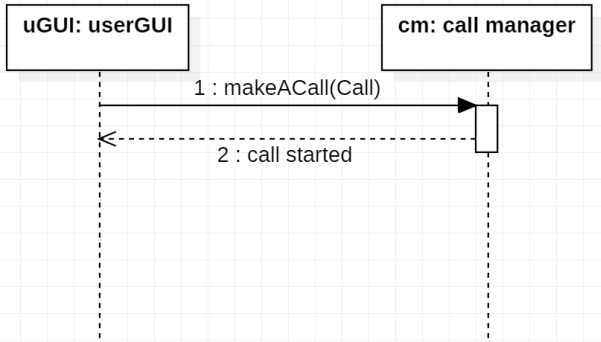


Search User (FR12.1)



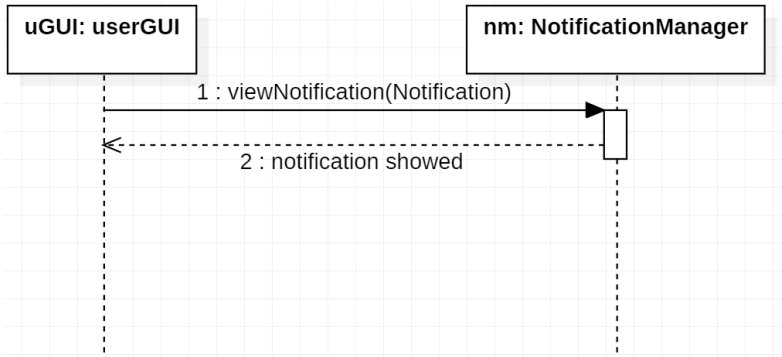
Add contact details (FR13.1)

Make a Call (FR14.1)

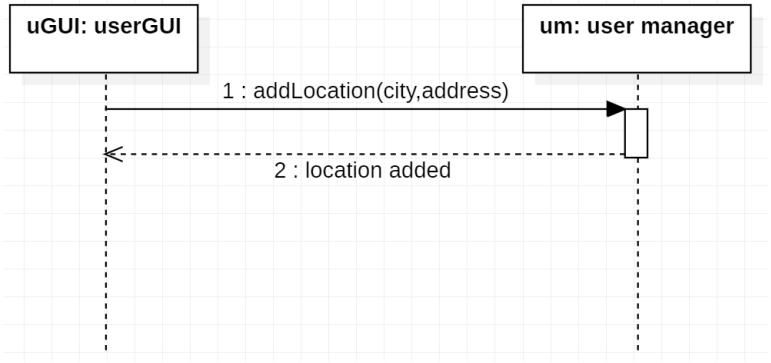


Software design \ *runtime-level sequence diagram* →

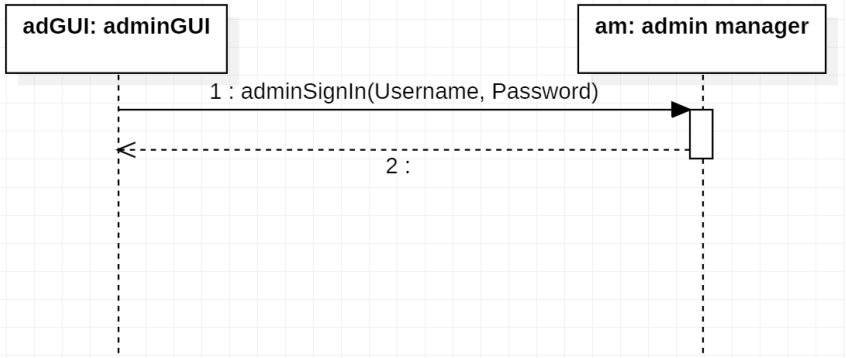
View notification (FR16.1 \ FR16.2)



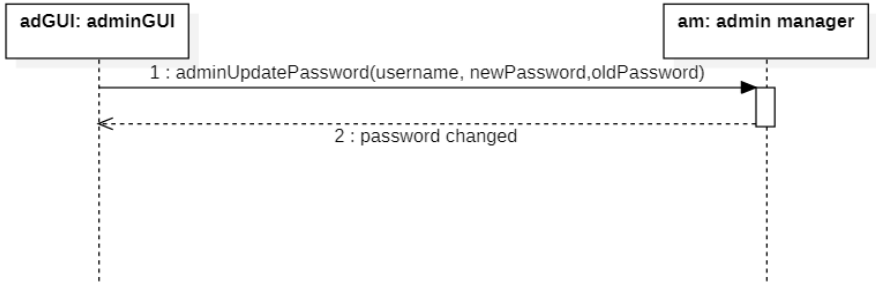
Add Location (FR17.1 \ FR17.2)



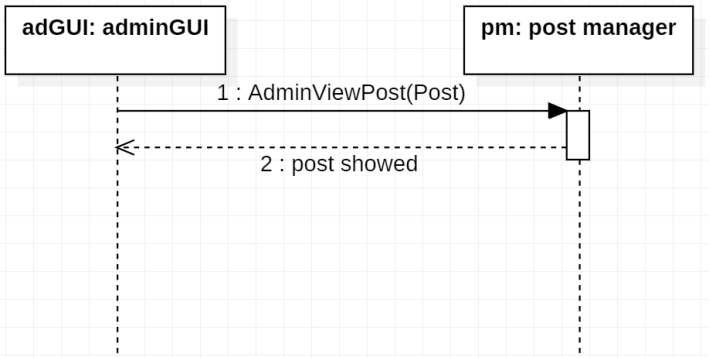
Admin Sign In (FR19.1 \ FR19.2)



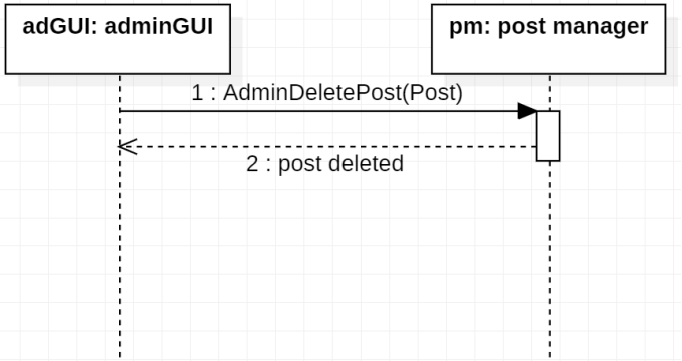
Admin Update Password (FR20.1)



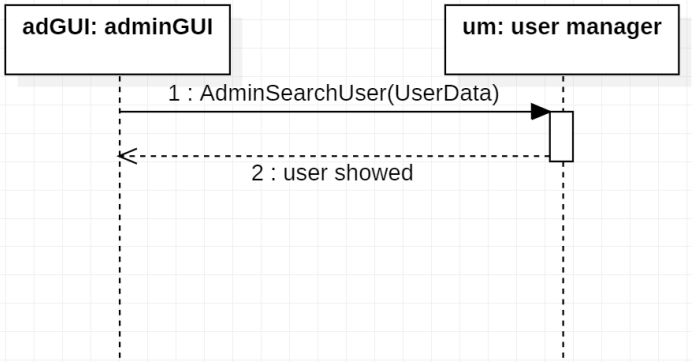
Admin Delete Post (FR22.1)



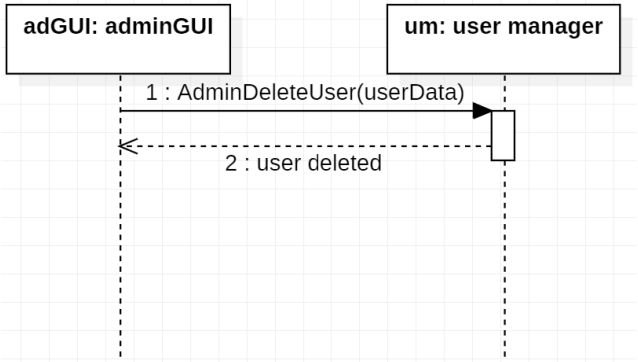
Admin Delete Post (FR22.1)



Admin Search User (FR23.1)



Admin Delete User (FR24.1)



Implementation \ *technology used*

Tools	Use
Visual studio	As IDE to write code
Python	Language for Back-end
Html	Language for Front-end
Sqlite	To collect Data

Implementation \ *project structure* \ *front end*

Front-end: .html Files	
File	**Description**
add.html	sign-up form
find.html	“Search missing persons” form
index.html	Give a unique identification number for the users
login.html	“Ukraine Finder” login page
Lost.html	“Report Missing Persons” page
main.html	“kindly Select the option” page
result_file.html	Database structure (id, Username, Password and Email of the users)

Implementation \ *project structure* \ *back end*
→

Back-end: .py (python) files	
File	**Description**
admin.py	It implements the admin functionalities
apps.py	Configuration of the application (Appconfig)
forms.py	build forms to accept input from site visitors
models.py	Members (Username, Password, Email), Find (Name, Photo, date) and Lost (Photo, date).
tests.py	tests for the application
urls.py	User part of the mapping between the URL path expressions to Python functions (view.py)
manage.py	It is automatically created in Django project. It is used to launch the application.

Implementation \ *project structure* \ *back end*
X

views.py	takes the web request and returns a web response. For users \ members.
settings.py	to configure the options for the Django installation,.
urls.py	Urls related to the admin portal
wsgi.py	Application server uses to communicate with our code.

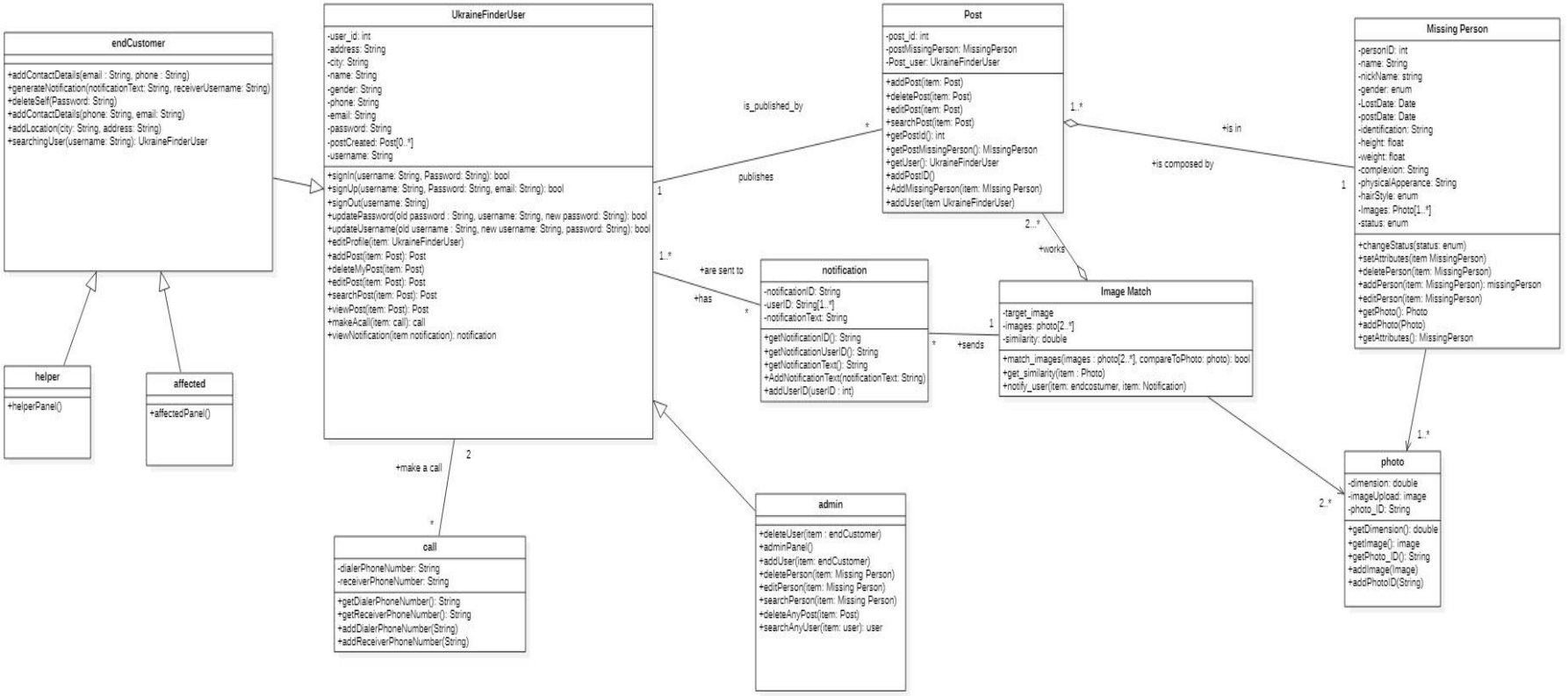
Database	
File	Description
db.sqlite3	A SQLITE3 file is a database file stored in the SQLite 3 format. It contains structured data records, which contain data types and values. In our case that contains the data about the Find, Lost and Members.

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

Implementation \ Developed Functions

User	Admin
• Sign in • Sign up • Sign out • Add post • View notification	• Sign in • Sign up • Sign out • Update password • Update username • View post • Delete post • Search user • Delete user

Implementation \ *uml class diagram*



Testing \ Sign up

Pre-conditions \ Pre-requisites: Run server

Dependencies: Internet connection

Enter : Email Address, Password, Username

Violation Test

userEmail = null

userName = null

password= null or password shorter than 3 characters

Post Conditions:

- Data correctly save in Database

Testing \ Sign out

Pre-conditions \ Pre-requisites:

- User must be sign in

Normal behaviour: Click sign out

Pre-conditions violation test: Click some other button

Post Conditions:

- User now have to sign in again

Testing \ Add post

Pre-conditions \ Pre-requisites:

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

Normal behavior:

Name and picture

Violation Test:

Name = null , Picture uploaded

Or Name given, Picture not uploaded

Post Conditions:

Picture and Name is stored in database

Testing \ Notify user

Pre-conditions \ Pre-requisites:

- Missing and found Picture must be in Database

Dependencies: Picture should be clear

Normal behavior: Match Found Notification sent

Violation Test: Not uploading picture in find => Invalid

Post Conditions:

Matched image post must be deleted

Testing \ Image Matching

Pre-conditions \ Pre-requisites:

- Missing and found Picture must be in Database

Dependencies: Picture must of human being

Normal behavior: Upload picture and hit search

Pre-conditions violation test: Not uploading a picture => Error

Post Conditions:

Return match status

Testing \ Admin Sign in with Email password

Pre-conditions \ Pre-requisites:

- The user must have an account on Ukraine finder.

Normal behavior: userName = ’admin’ password = ’admin’

Pre-conditions violation test:

userName , password= null,

userName or password= wrong

Post Conditions:

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

Testing \ Admin update Password

Pre-conditions \ Pre-requisites:

- The user must have an account on Ukraine finder.
- Signed in

Normal behaviour: Enter old password=> new password =>Confirm new password

Pre-conditions violation test: Wrong old password, Not confirmation of entering new password

•**Post Conditions:**

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

Testing \ Admin view post

Pre-conditions \ Pre-requisites:

Admin must login

Normal Behaviour: Click on post at timeline

Violation Test: Not clicking on specific post

Post Conditions:

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

Testing \ Admin Delete post

Pre-conditions \ Pre-requisites:

- Admin must login

Normal Behaviour: Click on delete post button and confirm delete

Violataion Test: Not login from admin page Click on delete post button but not on confirm delete

Post Conditions: the post is deleted

Testing \ Admin Delete user

Pre-conditions \ Pre-requisites:

- Admin must login

Normal Behaviour: Click on delete user button and confirm delete user

Precondition violation test: Not login from admin or not confirming to delete user

Post Conditions:

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

Testing \ Admin View \ search user

Pre-conditions \ Pre-requisites:

- Admin must login

Normal Behaviour: Admin click on view user button

Precondition Violation test: Not login from admin account or not clicking on view user button

Post Conditions:

Should be able to view \ search user details

Testing \ Admin Add user

Pre-conditions \ Pre-requisites:

- Admin must login

Normal Behaviour: Fill all fields

Precondition Violation test: Not login from admin page or not filling all fields

Post Conditions:

- User will not login with the email he \ she has been deleted
- Details should be added in database

Demo of Ukraine Finder Website

Thank You for Your Attention

Team **ARS**

Q&A