



Faculty of Engineering and Architecture

Computer Engineering Department

Software Engineering Project



June 2017



THIS PROJECT IS PREPARED BY:

EGI PAPA

ERGERT POPA

GERI ZEQIRI

LEDIO BAHJA

REI MANDIJA

VARIS JORGANXHI

Abstract

Technology Over Police Department(TOPD) is a software approach to provide means of communication and organization to the police department. In doing so, our aim is to provide a software in order to ease the communication between different departments in the State Police, a technical approach to the structure of the organization, provide technological means in order to exchange information between links provided between different departments of the organization. The system will provide a categorized user interface, meaning that the software will strictly follow the hierarchy of the employees, and by doing so we will provide several categories of user that shall login in the system to perform their duties.

The software interface shall be easy to access, even though this will strongly depend in the user category actions that need to be perform and the privileges of that category. The implementation will be in the format of an web application, the language of programming will be PHP, and the software will be implemented using an PHP framework, Symfony. We are aiming to provide an API format to the software in order to use JavaScript libraries and tools, in order to provide an interactive application and balance the load of each web page. Another reason for this type of implementation is to provide a mean for the system to communicate also with mobile applications, which will be useful, not only to the police department, but we will make possible a communication, between the police and the citizens. Another link will be between police stations or departments and the police employees that are in duty in the streets and need information from the department or from the headquarters.

This is a basic description of the software system that we are striving to build, and new features will be added incrementally to the software specification and development phase.

Table of Contents

Abstract.....	3
Executive Summary	Error! Bookmark not defined.
Project Overview	Error! Bookmark not defined.
Purpose and Scope of this Specification	Error! Bookmark not defined.
Product/Service Description	8
Product Context.....	8
User Characteristics.....	8
Assumptions.....	9
Constraints	9
Dependencies.....	9
Apportioning of requirements	9
Requirements.....	Error! Bookmark not defined. 0
Functional Requirements.....	10
Non- functional requirements.....	12
Usability	13
Performance.....	14
Capacity.....	14
Availability.....	15
Latency.....	15
Protection.....	15
Authorization and Authentication.....	15
Manageability/Maintainability.....	16
Monitoring	16
Maintenance	16
Operations.....	16
System Interface/Integration.....	17
Network and Hardware Interfaces	17
Systems Interfaces.....	17
Portability.....	17
User Scenarios/Use Cases.....	18
User Stories	19

Scenarios.....	21
General Use cases	33
DIAGRAMS	40
Implementation.....	Error! Bookmark not defined.

*THIS PAGE IS INTENTIONALLY LEFT
BLANK*

Executive Summary

Project Overview

Technology Over Police Department(TOPD) is a software approach to provide means of communication and organization to the police department. In doing so, our aim is to provide a software in order to ease the communication between different departments in the State Police, a technical approach to the structure of the organization, provide technological means in order to exchange information between links provided between different departments of the organization. The system will provide a categorized user interface, meaning that the software will strictly follow the hierarchy of the employees, and by doing so we will provide several categories of user that shall login in the system to perform their duties.

The software interface shall be easy to access, even though this will strongly depend in the user category actions that need to be perform and the privileges of that category. The implementation will be in the format of a web application, the language of programming will be PHP, and the software will be implemented using an PHP framework, Symfony. We are aiming to provide an API format to the software in order to use JavaScript libraries and tools, in order to provide an interactive application and balance the load of each web page. Another reason for this type of implementation is to provide a mean for the system to communicate also with mobile applications, which will be useful, not only to the police department, but we will make possible a communication, between the police and the citizens. Another link will be between police stations or departments and the police employees that are in duty in the streets and need information from the department or from the headquarters.

This is a basic description of the software system that we are striving to build, and new features will be added incrementally to the software specification and development phase.

Purpose and Scope of this Specification

The purpose of this project is to come to the help of the police department, in organizing and to make it easier to communicate inside the dep. Another purpose would be to provide a way for the citizen to report in real time.

Product/Service Description

Product Context

How does this product relate to other products? Is it independent and self-contained? Does it interface with a variety of related systems? Describe these relationships or use a diagram to show the major components of the larger system, interconnections, and external interfaces.

This product (software) aims to help in communication the police departments with each other and citizens with the police.

User Characteristics

For the user it will be easy to access the software.

There will be different level users according to their position in the administrate.

- Citizen
- Police patrol
- District operator
- Police Station operator
- General operator

Assumptions

There will be a limited access of information between the users.

If a document is intended to be shared restrictively, it won't be visible to all of the users.

List any assumptions that affect the requirements, for example, equipment availability, user expertise, etc. For example, a specific operating system is assumed to be available; if the operating system is not available, the Requirements Specification would then have to change accordingly.

Constraints

There are many common examples of technical constraints:

- *Programming language* - often times a specific programming language will be required for various reasons. For example, the customer may be a Java or Ruby or Microsoft shop. You might simply prefer a certain language over another, or have specific expertise that dictates using a

particular programming language. Nearly always, once you've picked a language you are stuck with that choice for the remainder of the project.

- *Operating system or platforms supported* - It must work on Windows, or Linux, or iOS, or Qt on Solaris, or IE 6 on Windows XP, or ... building software that does not satisfy the platform constraint means you have failed to design a software system that satisfies stakeholders' key concerns.
- *Use of a specific library or framework* - Sometimes a specific library might be required to be used. The specific origin might come from the business but the influence is very technical. A common example at many companies is the use of specific open source libraries. Some companies might require that open source always be used. Other companies, might have an approved list indicating which open source software may be used. An interesting example at IBM is that we are required to target the "Blue" JVM, [IBM's JDK](#) for all JVM-based projects.

Dependencies

List dependencies that affect the requirements.

- Reporting to a higher hierarchical level every time there are new information.
- Tasks given in real time by a certain level of user.
- In order to take an action, first it should be reported.
- Official releases should be done in an exact given time.

Requirements

Functional Requirements

Req#	Requirement	Comments	Priority	Date Rvwd
R_01	Different level users according to their position in the administrate.	According to their position, each will have their own profile	1	22/05/17
R_02	Privileges according to their position.	-	1	22/05/17
R_03	Exchange of information	Information between levels of hierarchy	2	22/05/17
R_04	Limited access of information	The information will be visible only to the profile of intended user.	1	22/05/17
R_05	Interactivity between the department and the polices on duty.	Giving tasks and reporting in real time	1	22/05/17
R_06	Interactivity with the citizens.	The citizens can report with the help of the android app	1	22/05/17
R_07	Reporting to a higher hierarchical level.	-	1	22/05/17

R_08	Sharing the information restrictively.	The information shared will be highly protective.	1	22/05/17
R_09	Official release of the information.	Releasing the infos at the end of the day.	2	22/05/17
R_10	Location with Google Maps	Location of each police station at each district.	2	22/05/17
R_11	Assigning tasks	X user assigns task to Y user.	1	22/05/17
R_12	Task problems	If a task is not completed or is facing difficulties they should report to the higher up organization.	1	22/05/17
R_13	Account validation (QR Code)	Once a user creates an account, in order to activate it, a QR code is provided to the web app, in order to verify its account.	2	22/05/17
R_14	Flow of information	Flow of information between different levels of the hierarchy, directly or indirectly.	1	22/05/17
R_15	Tasks between levels of hierarchy	Assigning tasks to different level of hierarchy to the lower tier level of this hierarchy.	1	22/05/17
R_16	Reporting citizens for country scale emergencies	There should be a special purpose section in the software, in order to inform the citizens in real time for the country scale emergencies.	1	22/05/17
R_17		Once the task is finished and reported as finished,		

	Detailed report after terminating tasks	the head of operational unit is required to provide a detailed report to the operational office, in order to formally persist the report in the software.	2	
R_18	Units not responding	If the higher up organization that assigned a specific task to an organizational unit, does not receive information on the situations, it communicates with the unit, in order to provide help by assigning more officers or units, or if the task is not any more valid to inform the unit to interrupt its operation.	1	22/05/17
				22/05/17
R_19	Violation of citizen's rights	There should be a specific section for citizens in order to denounce the police officers, in case of violence, or if the police worker violate the citizens' rights, this denunciation is to be reported to the appropriate section of the organization.	1	22/05/17

3.2 Non-functional Requirements

1. Product Requirements:

A. Usability:

- i. The software interface shall be easy to access, even though this will strongly depend in the user category actions that need to be perform and the privileges of that category. There is going to be a help page on how to properly use the software. Each of the options to use will be described in details.
- ii. It should be practical and easy to cope with the abilities of the user, namely the police department workers and also the citizens, even though the citizens access is restricted only to the android app.
- iii. It shall be easy to train the users, the training is concentrated more on the police department workers as here is supposed to happen the heavy load of information and operations which in interdependable with different categories, and this training is supposed to take place in less than one week, and the users must not exceed two minor mistakes per hour of work.
- iv. As the software is supposed to be easy to work with, we do not intend to create a specific tutorial book, but we should provide the user the tutorial based on presentations and practice.

B. Efficiency

- i. Performance Requirements: The performance of the software is going to depend on the server used. JavaScript and Ajax will be used which means that it will be light weighted and the browser won't crash.
- ii. Space Requirements: We are striving to build a good, strong server. It will be able to support a number of simultaneous users.
- iii. As the software is supposed to perform a country size scale, the servers that will be used at least 1 Terabyte of memory, fast, preferable with SSD, and the operating system is thought to be Linux Ubuntu.
- iv. The central server, which is located to the Central General Directory, which will operate and access the information from all subdirectories will require at least 100 Terabyte of memory.

- v. For every police station or department there should be one server operating with android application requests and response.
- vi. As the software is planned to be fast, there should be an internet connection of at least 20 dedicated Megabytes of internet, both in download and upload.

C. Security:

- i. Specify the factors that will protect the system from malicious or accidental access, modification, disclosure, destruction, or misuse. For example:
 - o *encryption*
 - o *activity logging, historical data sets*
 - o *restrictions on intermodal communications*
 - o *data integrity checks*
- ii. Security provided to the access of user private information.
- iii. Security provided to access of data.
- iv. Restriction provided to access of data.
- v. There should also be provided a term of agreement to the user, in order to access their information and the data posted to the server.
- vi. The android application should also include a list of modules and components that will be required to be accessed from the application, as known as permissions.

D. Dependability

- i. Even though in the current required version of the software there is no specific requirement to be dependable on other modules or applications, in the next version, if required it will be interactive with the State office, in order to access the information of each user, in order to provide consistency and reliability, and also it might be required to have an agreement and API to communicate with the telecommunication agencies.

2. Organizational Requirements.

A. Environmental Requirements:

- i. The software is going to operate 24/7. Its availability is measured by its performance when a subsystem fails, its ability to resume service in a state close to the state of the system at the time of the original failure, and its ability to perform other service-affecting tasks (such as software upgrade or configuration changes) in a manner that eliminates or minimizes down time. We are planning to add coverage for the whole country area by using Google maps.

B. Operational Requirements:

- i. Reporting in real time.
- ii. Sharing info restrictively.
- iii. Backup and recovery information.
- iv. Assigning tasks

C. Developments Requirements:

- i. The team is divided in two sub teams, one responsible for the front-end development and the other responsible for the back-end development.
- ii. For the front-end part, the sub team is required to have intermediate knowledge on html, CSS, and JavaScript technologies.
- iii. Back end sub team is required to have intermediate knowledge and at least a year of experience in PHP, and is required to acquire necessary knowledge and practice in PHP framework Symfony.
- iv. The team will use an agile approach to the development of the software, namely the XP programming, or extreme programming model.
- v. The sub teams should organize meetings based on a regular basis, and also it is required that the team to organize weekly meetings, in order to organize

the work, to make the necessary changes to the software, to discuss about new features, and to deliver new duties to sub teams.

3. External Requirements

A. Regulatory Requirements: According to law regulations

B. Ethical Requirements

C. Legislative Requirements:

- i. Accounting Requirements: Verification with Id card no.
- ii. Safety/Security Requirements: System encrypted.

*THIS PAGE IS INTENTIONALLY LEFT
BLANK*

User Scenarios/Use Cases

Nr	User Story Name	Description
1.	Register	Creates a new user for accessing the software. It requires name, surname, username, password, id card/phone no.
2.	Citizen's News Feed Page	All the news is published automatically. Is refreshed by swiping down.
3.	Citizen's Emergency Page	All the emergency news is published.
4.	Citizen's Virtual Denunciation Page	A new page with a tabular form opens, and you fill it with Title, short description and upload a photo of what you are reporting. Then you send it.
5.	Police Patrol's News Feed Page	All the news is published automatically. Is refreshed by swiping down.
6.	Police Patrol's Emergency Page	All the emergency news is published.
7.	Police Patrol's Virtual Orders	A new page is shown with all of your orders published. He can select which of the orders are executed. In case of any problems, the user can contact with the department. Then clicks Send .
8.	Operational Room's News Feed Page	where all the news is published automatically.
9.	Operational Room's Orders Page	From a higher level of hierarchy. ->From orders you are send to another tab: Management of orders : list of orders.
10.	Operational Room's HRM Page	List of all the police officers.

11.	Operational Room's Report's Page	All the reports from the police officer after completing the task.
12.	DISTRICT OPERATIONAL ROOM's HomePage	It has a list of other tabs:
13.	a. News Feed	where all the news is published automatically.
14.	B. New Denunciations	List of denunciations.
15.	c. Orders	->From a higher level of hierarchy. ->Send the orders to the lower level of hierarchy. ->Checks all the orders if they are completed or not.
16.	d. New User	Creates accounts for human resources. (Dep. Operational Room)
17.	e. Human Resources Management	Creates and edits users of the police dep.
18.	f. Management of reports.	After gathering the reports from the departments send them to the General Operating Room.
19.	GENERAL OPERATIONAL ROOM's Home Page	It has a list of other tabs:
20.	a. News Feed	After gathering the information, edits and then updates.
21.	b. Emergency Page	Publish national emergencies and gives order directly to the whole police dep.
22.	c. Human Resources Management	Creates and edits users of the district operating room.
23.	d. Orders for district operating room	Daily orders.
24.	Settings	User can edit his attributes. Clicks Send and waits for confirmations if it's valid or not.

Scenarios

Citizen: Scenario 1

1. Firstly installs the app
2. Clicks on the register tab.
3. An empty form appears for completing.
4. The user enters his/her:
 - Name, Surname
 - Username
 - Password
 - Id Card/Pass no. OR Phone no.
5. The user clicks Sign In.
6. User is now logged in.
7. Opens his profile
8. Goes to **News Feed**
9. Checks all the news published. Refresh it by swiping down.

Citizen: Scenario 2

1. Firstly installs the app
2. Clicks on the register tab.
3. An empty form appears for completing.
4. The user enters his/her:
 - Name, Surname
 - Username
 - Password
 - Id Card/Pass no. OR Phone no.
5. The user clicks Sign In.
6. User is now logged in.
7. Opens his profile

8. Goes to **Emergency Page**
9. Checks all the emergency news published.

Citizen: Scenario 3

1. Firstly installs the app
2. Clicks on the register tab.
3. An empty form appears for completing.
4. The user enters his/her:
 - Name, Surname
 - Username
 - Password
 - Id Card/Pass no. OR Phone no.
5. The user clicks Sign In.
6. User is now logged in.
7. Opens his profile
8. **Virtual Denunciation:**
9. Completes a tabular form with Title, short description and upload a photo of what you are reporting. Then you **send it**.

Police Patrol Scenario 4

1. The user enters his/her:

- Name, Surname
- Username
- Password
- Id Card/Pass no. OR Phone no.

2. The user clicks Sign In.
3. User is now logged in.
4. Opens his profile.
8. Goes to **News Feed**
9. Checks all the news published. Refresh it by swiping down.

Police Patrol Scenario 5

1. The user enters his/her:
 - Name, Surname
 - Username
 - Password
 - Id Card/Pass no. OR Phone no.
2. The user clicks Sign In.
3. User is now logged in.
4. Opens his profile.
8. Goes to **Emergency Page**
9. Checks all the emergency news published.

Police Patrol Scenario 6

1. The user enters his/her:
 - Name, Surname

- Username
- Password
- Id Card/Pass no. OR Phone no.

2. The user clicks Sign In.

3. User is now logged in.

4. Opens his profile.

8. Goes to Virtual Orders Page

9. All of his orders published. He can select which of the orders are executed. In case of any problems, the user can contact with the department. Then clicks **Send**.

Police Dep. Operational Room Scenario 7

1. The user enters his/her:

- Name, Surname
- Username
- Password
- Id Card/Pass no. OR Phone no.

2. The user clicks Sign In.

3. User is now logged in.

4. The user clicks Home Page

5. Then clicks News Feed

6. Checks all the emergency news published.

Police Dep. Operational Room Scenario 8

1. The user enters his/her:

- Name, Surname
- Username

-Password

-Id Card/Pass no. OR Phone no.

2. The user clicks Sign In.

3. User is now logged in.

4. The user clicks Home Page

5. Then clicks Orders

6. Sees a list of orders from higher level of hierarchy.

Police Dep. Operational Room Scenario 9

1. The user enters his/her:

-Name, Surname

-Username

-Password

-Id Card/Pass no. OR Phone no.

2. The user clicks Sign In.

3. User is now logged in.

4. The user clicks Home Page

5. Then clicks Human Resources Management

6. List of all the police officers.

Police Dep. Operational Room Scenario 10

1. The user enters his/her:

-Name, Surname

-Username

- Password
 - Id Card/Pass no. OR Phone no.
2. The user clicks Sign In.
 3. User is now logged in.
 4. The user clicks Home Page
 5. Then clicks Reports
 6. All the reports from the police officer after completing the task

District Operational Room Scenario 11

1. The user enters his/her:
 - Name, Surname
 - Username
 - Password
 - Id Card/Pass no. OR Phone no.
2. The user clicks Sign In.
3. User is now logged in.
4. The user clicks Home Page
5. Then clicks News Feed
6. Checks all the news published

District Operational Room Scenario 12

1. The user enters his/her:

- Name, Surname
- Username

- Password
- Id Card/Pass no. OR Phone no.

2. The user clicks Sign In.
3. User is now logged in.
4. The user clicks Home Page
5. Then clicks New Denunciations
6. Checks a list of all denunciations made

District Operational Room Scenario 13

1. The user enters his/her:
 - Name, Surname
 - Username
 - Password
 - Id Card/Pass no. OR Phone no.

2. The user clicks Sign In.
3. User is now logged in.
4. The user clicks Home Page
5. Then clicks Orders
6. Checks a list of all the orders.

Sends order to the lower level of hierarchy.

Checks if the orders are completed or not.

District Operational Room Scenario 14

1. The user enters his/her:
 - Name, Surname
 - Username

- Password
 - Id Card/Pass no. OR Phone no.
2. The user clicks Sign In.
 3. User is now logged in.
 4. The user clicks Home Page
 5. Then clicks New User
 6. Creates accounts for human resources

District Operational Room Scenario 15

1. The user enters his/her:
 - Name, Surname
 - Username
 - Password
 - Id Card/Pass no. OR Phone no.
2. The user clicks Sign In.
3. User is now logged in.
4. The user clicks Home Page
5. Then clicks Human Resources Management
6. Checks the list of all HR and edits them

District Operational Room Scenario 16

1. The user enters his/her:
 - Name, Surname
 - Username

- Password
 - Id Card/Pass no. OR Phone no.
2. The user clicks Sign In.
 3. User is now logged in.
 4. The user clicks Home Page
 5. Then clicks Management of Reports
 6. After gathering the reports from the departments send them to the General Operating Room.

General Operational Room Scenario 17

1. The user enters his/her:
 - Name, Surname
 - Username
 - Password
 - Id Card/Pass no. OR Phone no.
2. The user clicks Sign In.
3. User is now logged in.
4. The user clicks Home Page
5. Then clicks News Feed
6. Gathers information and edit and updates them.

General Operational Room Scenario 18

1. The user enters his/her:
 - Name, Surname
 - Username

- Password
 - Id Card/Pass no. OR Phone no.
2. The user clicks Sign In.
 3. User is now logged in.
 4. The user clicks Home Page
 5. Then clicks Emergency Page
 6. Publish national emergencies and gives order directly to the whole police dep.

General Operational Room Scenario 19

1. The user enters his/her:
 - Name, Surname
 - Username
 - Password
 - Id Card/Pass no. OR Phone no.
2. The user clicks Sign In.
3. User is now logged in.
4. The user clicks Home Page
5. Then clicks Human Resources Management
6. Creates and edits users of the district operating room

General Operational Room Scenario 20

1. The user enters his/her:

- Name, Surname
- Username

-Password

-Id Card/Pass no. OR Phone no.

2. The user clicks Sign In.

3. User is now logged in.

4. The user clicks Home Page

5. Then clicks Orders for district operating room

6. Uploads daily orders

User Scenario 21

1. The user enters his/her:

-Name, Surname

-Username

-Password

-Id Card/Pass no. OR Phone no.

2. The user clicks Sign In.

3. User is now logged in.

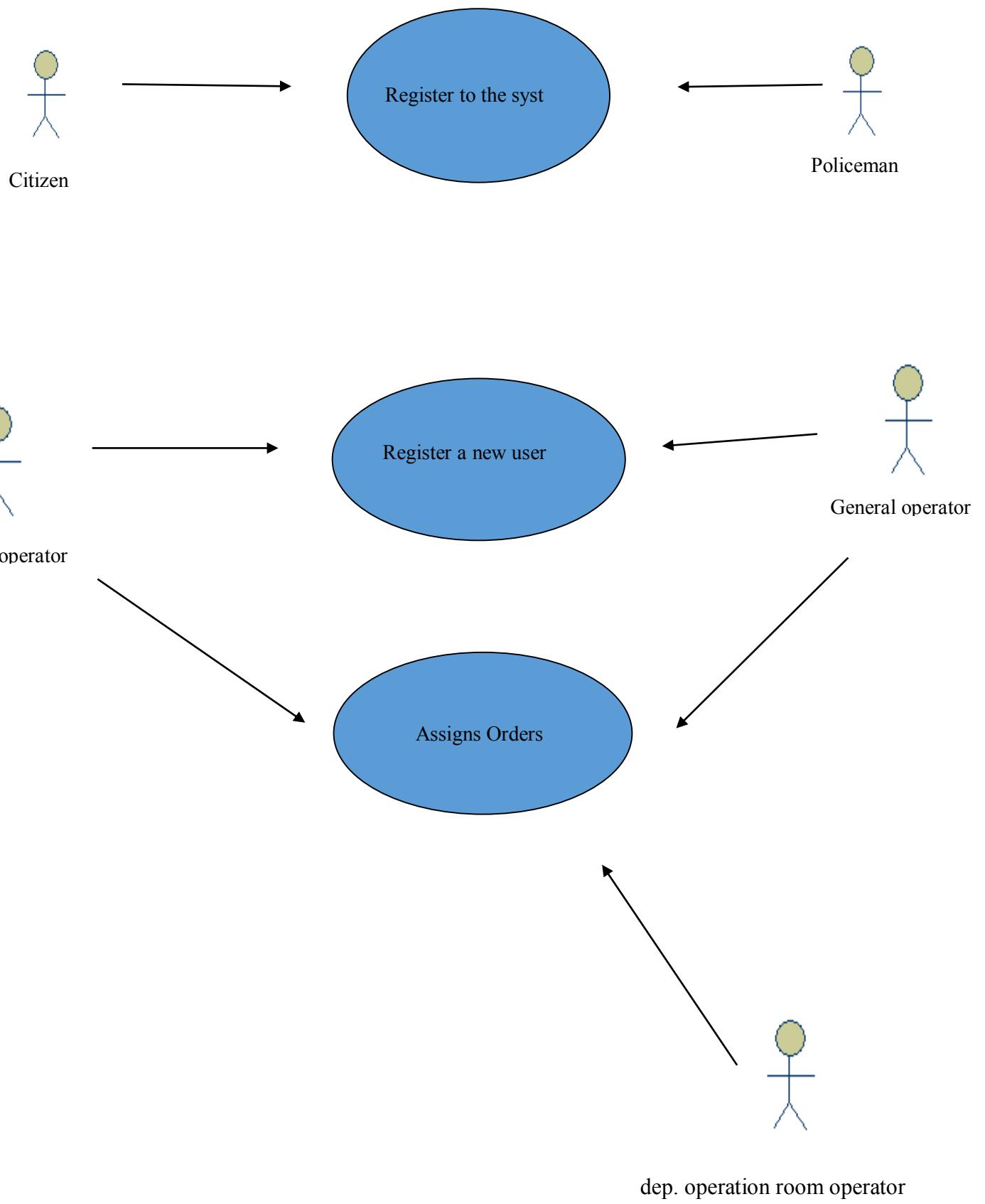
4. The user clicks Settings

5. User can edit his attributes. Clicks Send and waits for confirmations if it's valid or not.

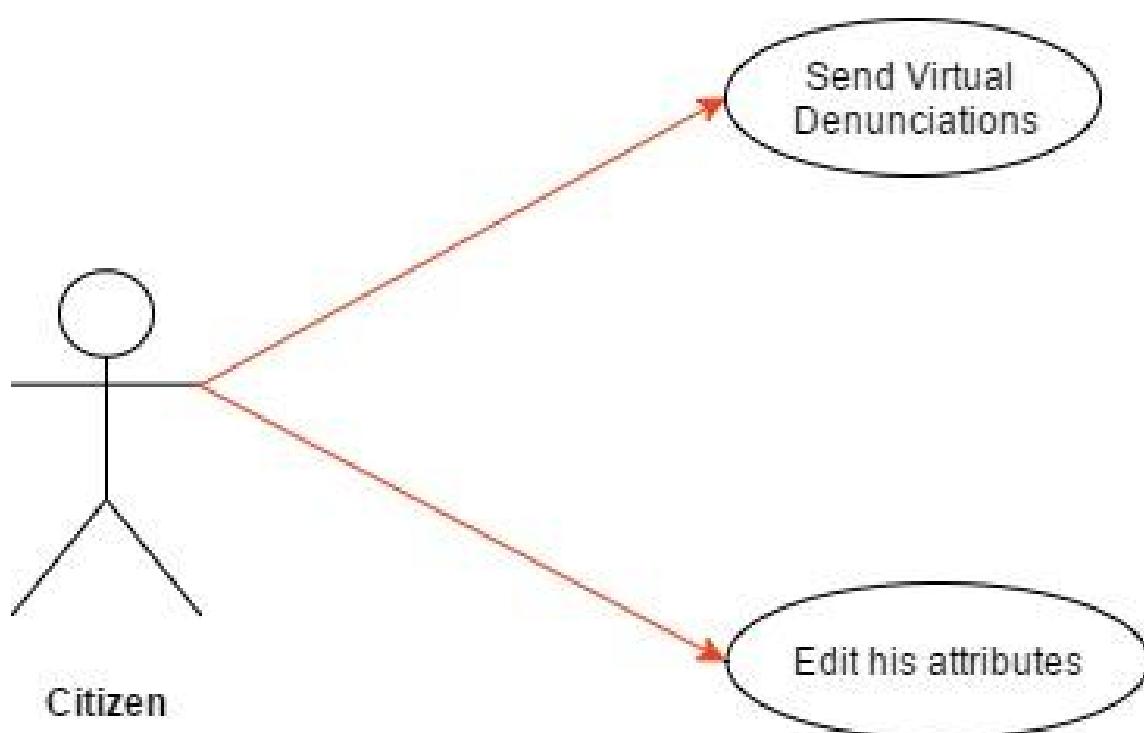
General Use cases

Use Case No.	1	2	3
Use Case name	Log in	Register a new user	Assigning Orders
Overview	Provides the user of TOPD system the right to access the system's functionalities. Access to each of the Home Tabs .	Provides a user of a higher level of hierarchy to register a new user of a lower level of hierarchy.	Provides a user of a higher level of hierarchy assign orders to a user of a lower level of hierarchy.
Actors	Users (citizens, police officers, operators).	Higher level hierarchy users (district operator, general operator).	Higher level hierarchy users (dep. operation room operator, district operator, general operator).
Pre-conditions	User must have valid name, surname, id no., phone no., internet connection.	The actors should be firstly registers and should have subordinates.	The actors should be firstly registers and should have subordinates.
Scenario flow	1. Log in form. 2. User enters the info required. 3. System checks the user's attributes. 4. System displays his/her profile.	1. Log in to his account. 2. Goes under Home Tab . 3. Selects Human Resources Management Tab . 4. Fills out a tabular form for creating a new user.	1. Log in to his account. 2. Goes under Home Tab . 3. Selects Orders Tab . 4. Sends the orders to the lower level of hierarchy.
	-Wrong user data. -> Access denied.	1. Fills the data for an already registered user. -> Registration failed.	1. Sends the order to a higher level of hierarchy. -> Sending Order Failed
Post conditions	After logged in the user access it's profile.	After registering the new user, he/she should be able to access his/her profile.	After sending the orders, subordinate, starts executing it.

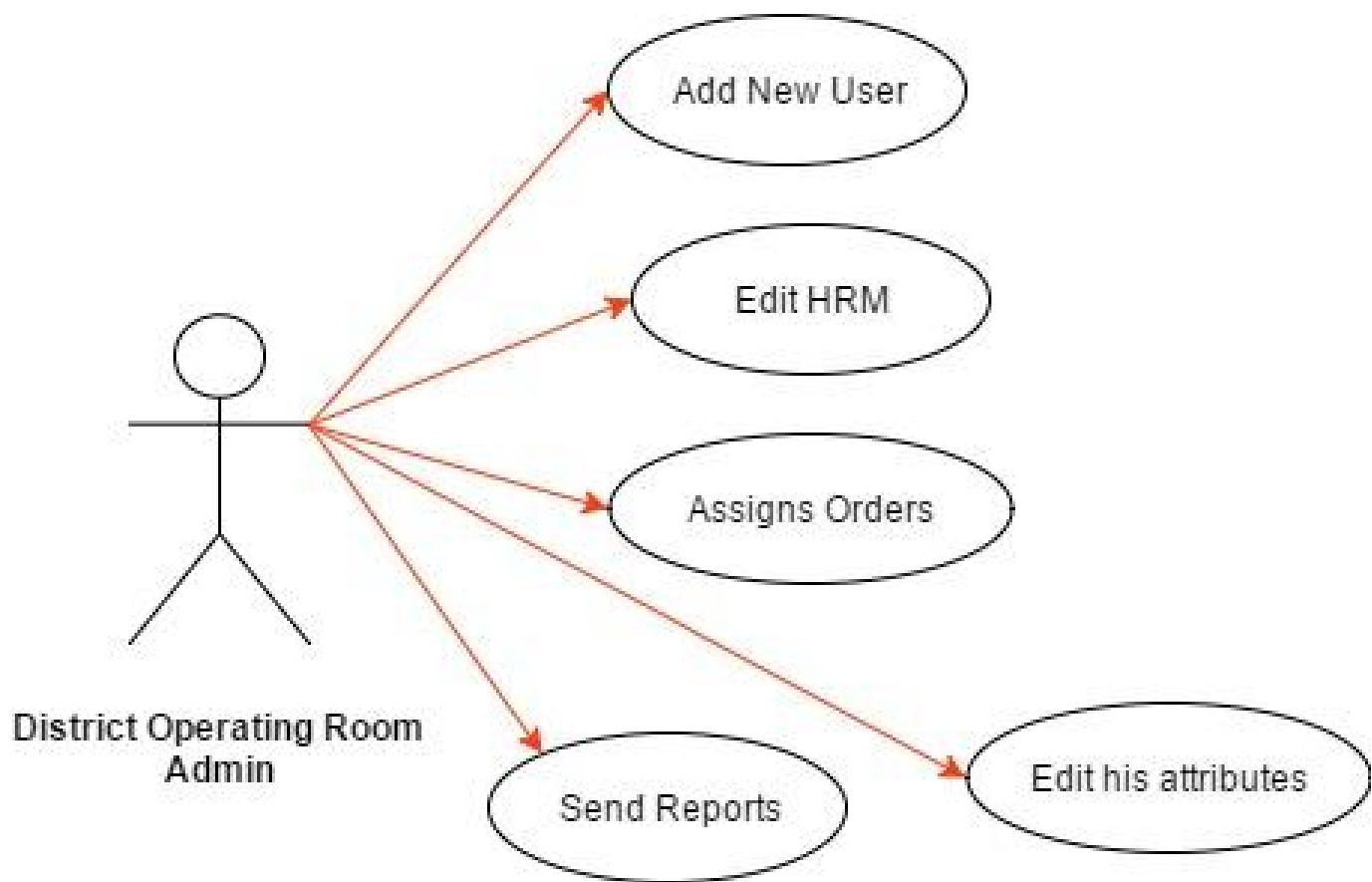
Use Case No.	4	5	6
Use Case name	Edit	Gathering information	Virtual Denunciations
Overview	Provides the user of TOPD system the rights to edit his/her attributes.	User gathers the news and updates the News Feed Page	Provides the citizen to denounce.
Actors	Users (citizens, police officers, operators).	Police Dep. Operational Room	User (citizen).
Pre-conditions	User must have been registered and logged in	The actors should be firstly logged in.	The user should firstly install the app, register, log in.
Scenario flow	1. Log in form. 2. User enters the info required. 3. Selects Settings Tab 4. Fill out a tabular form for editing the attributes	1. Log in to his account. 2. Goes under Home Tab . 3. Selects News Feed	1. Log in to his account. 2. Selects Virtual Denunciations . 3. Fills out a tabular form (title, description, photo).
	-Wrong user data. -> Access denied.		Not all fields are filled.
Post conditions	The user is still logged in his profile with new attributes.	After accessing the News Feed Page, the user edits the news gathered and upload them.	After sending the denunciations, the user can log out.



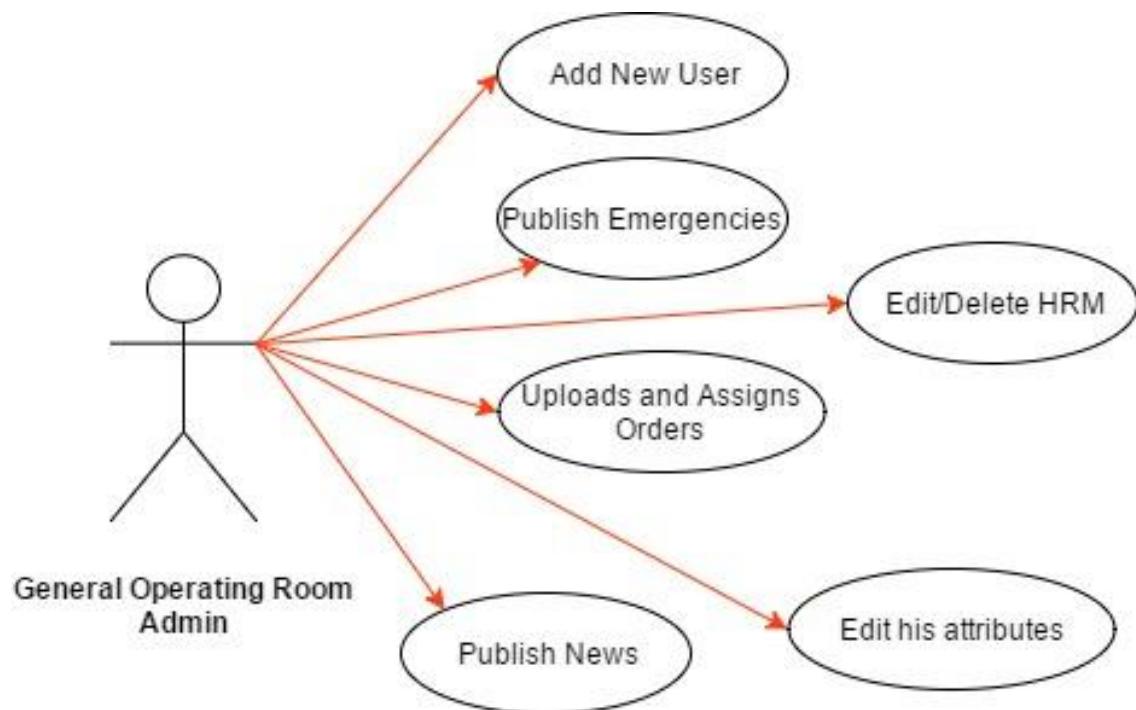
CITIZEN USE CASE



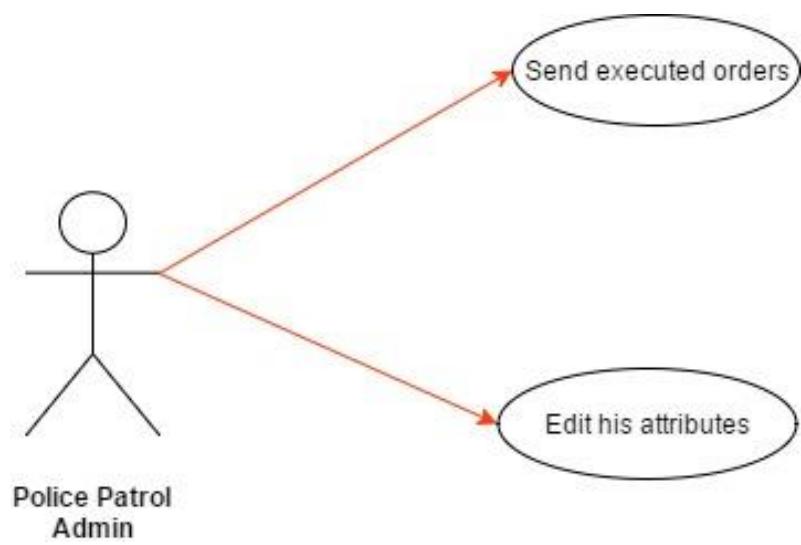
DISTRICT OPERATING ROOM USE CASE



GENERAL OPERATING ROOM USE CASE

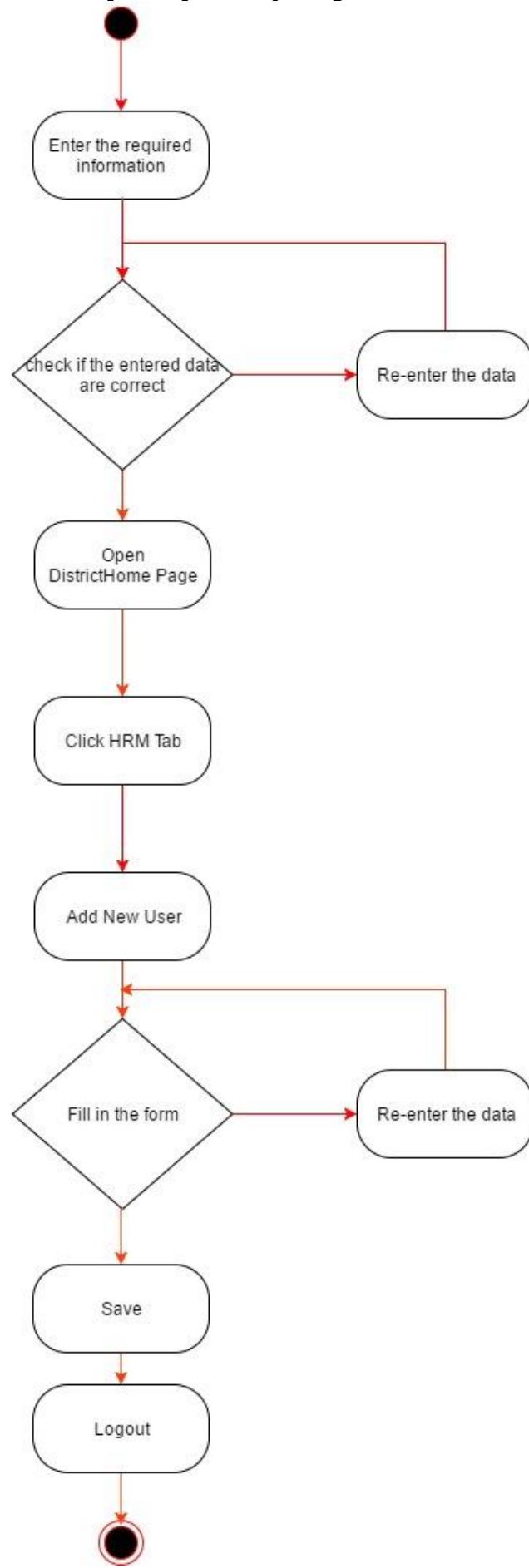


POLICE PATROL USE CASE

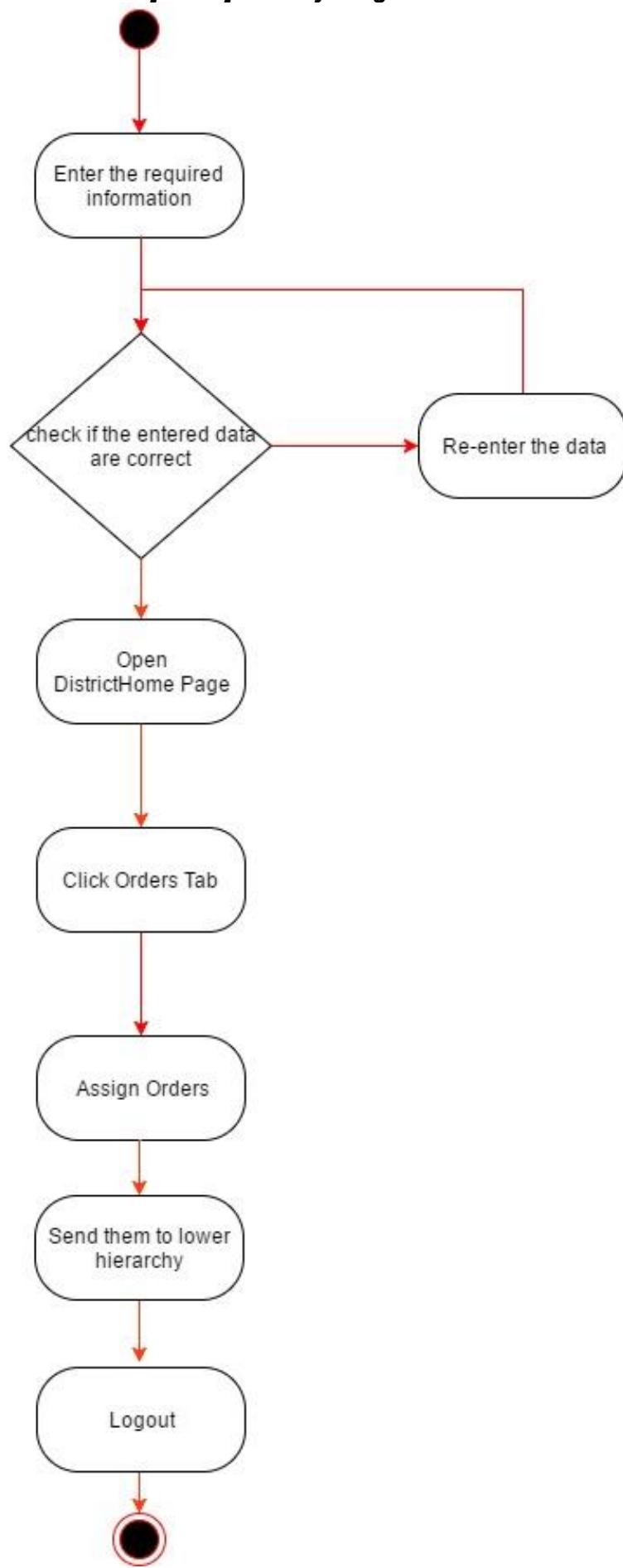


DIAGRAMS

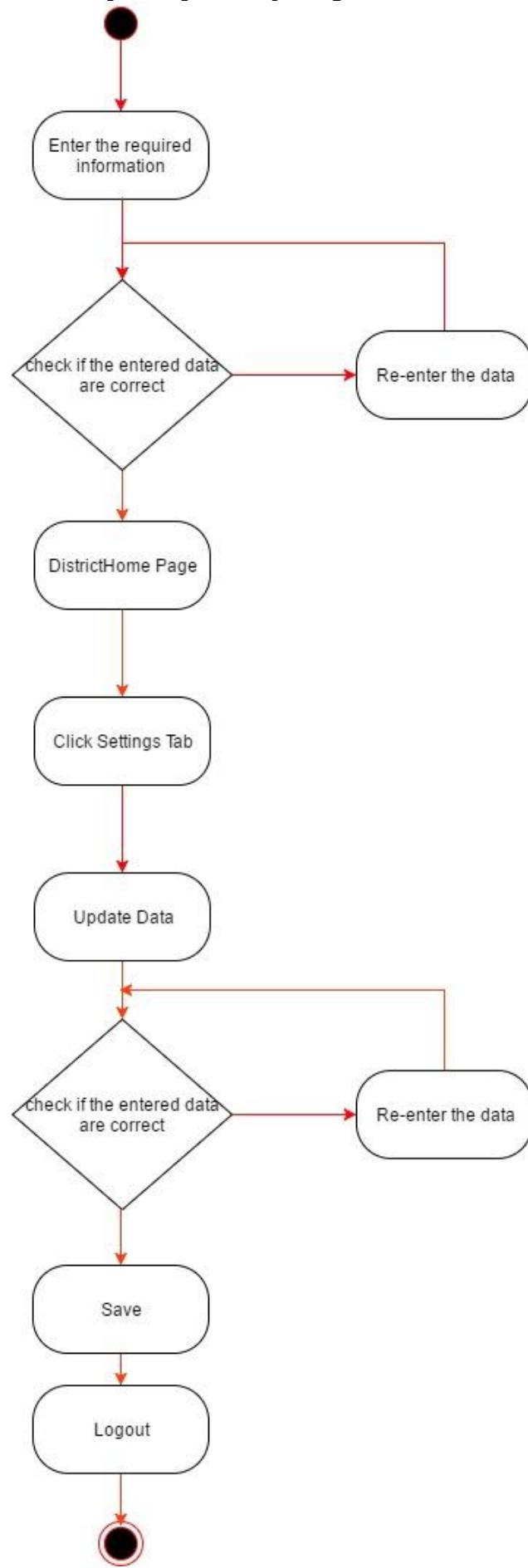
[TOPD] Activity Diagram



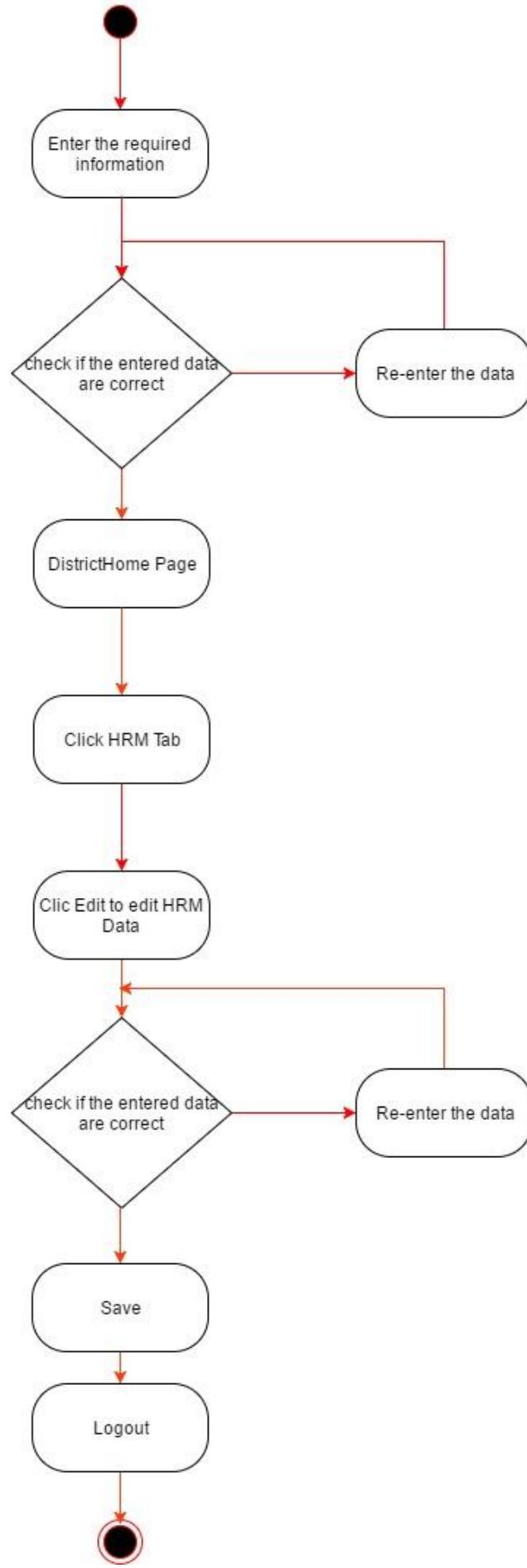
[TOPD] Activity Diagram



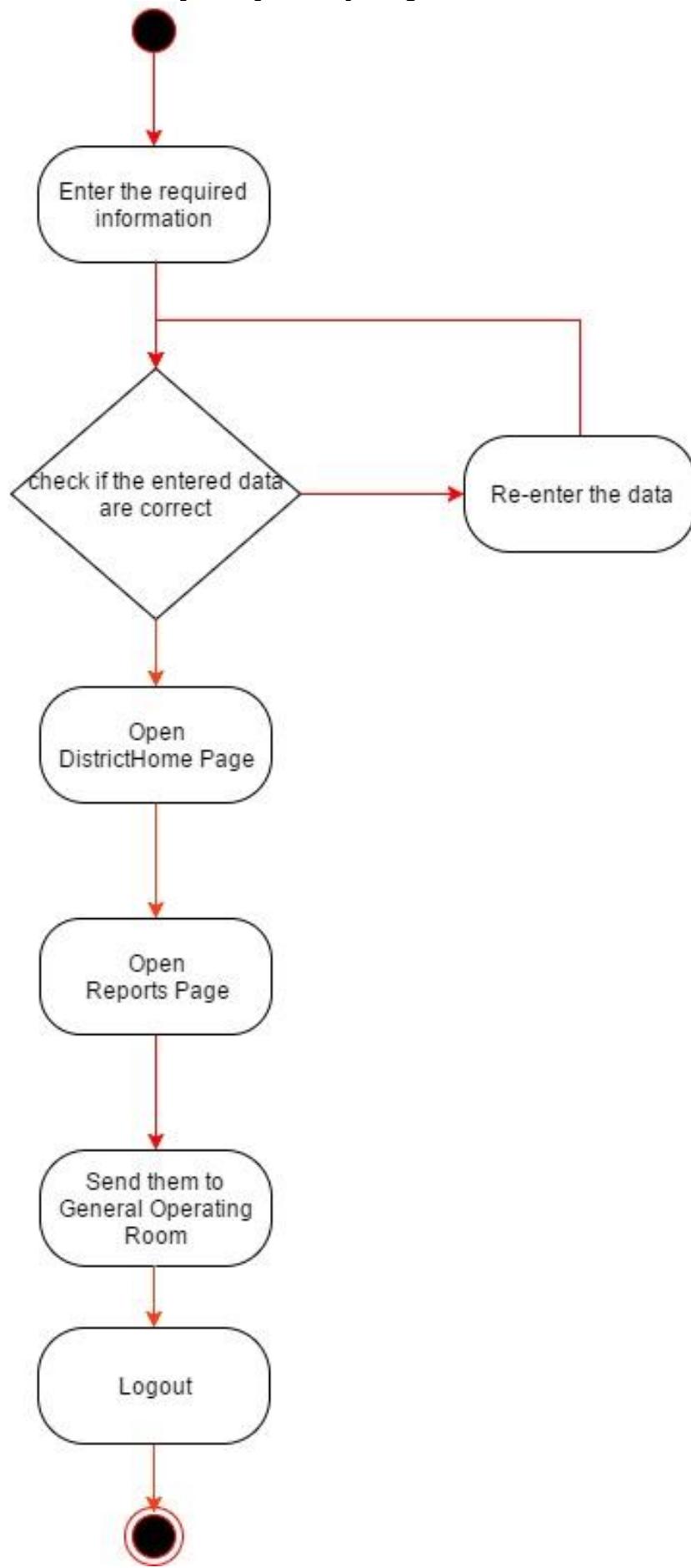
[TOPD] Activity Diagram



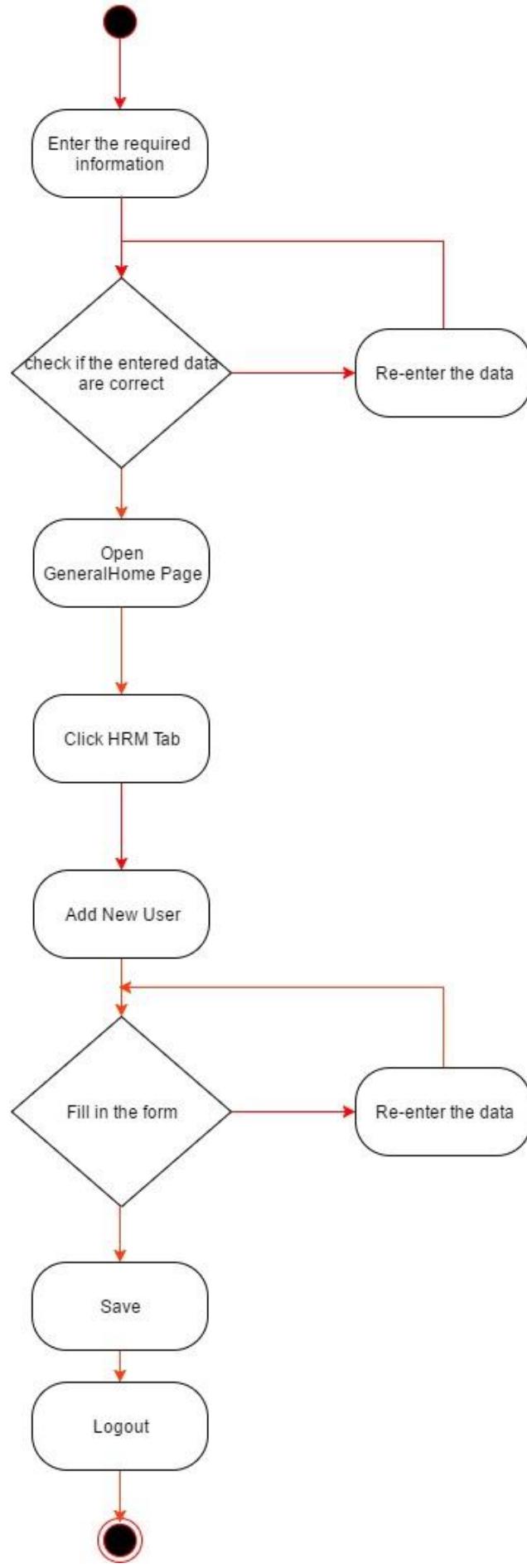
[TOPD] Activity Diagram



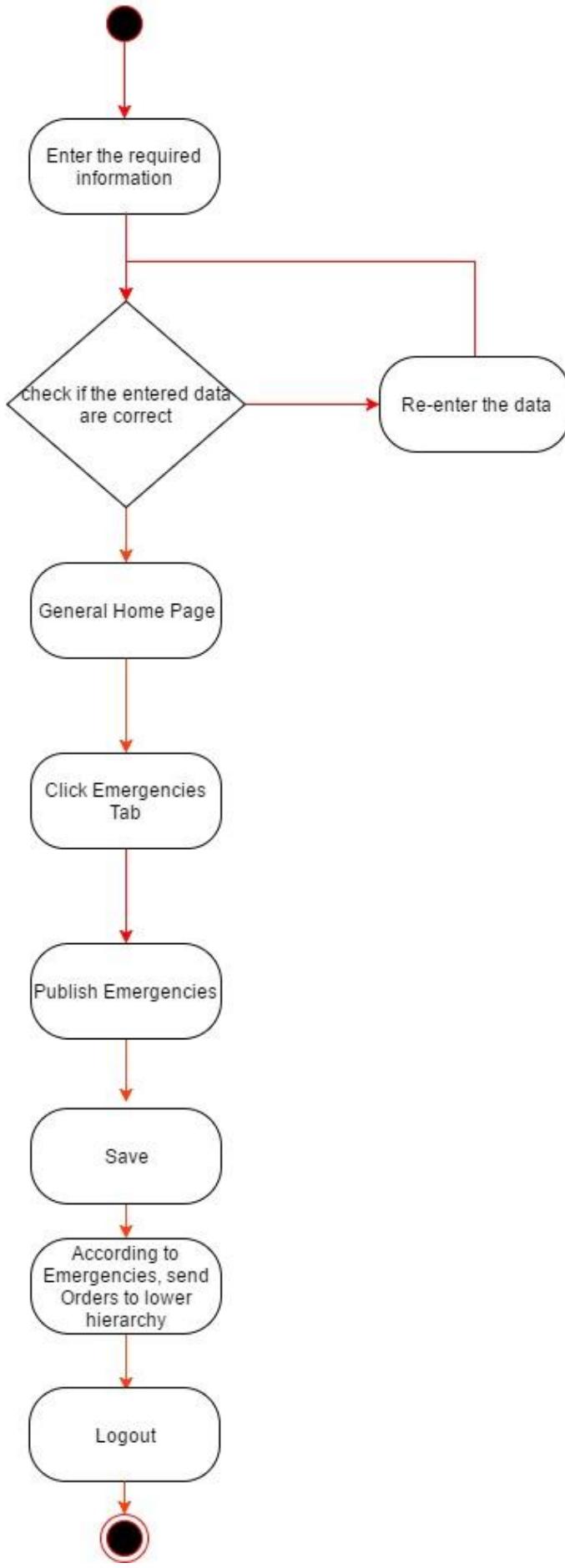
[TOPD] Activity Diagram



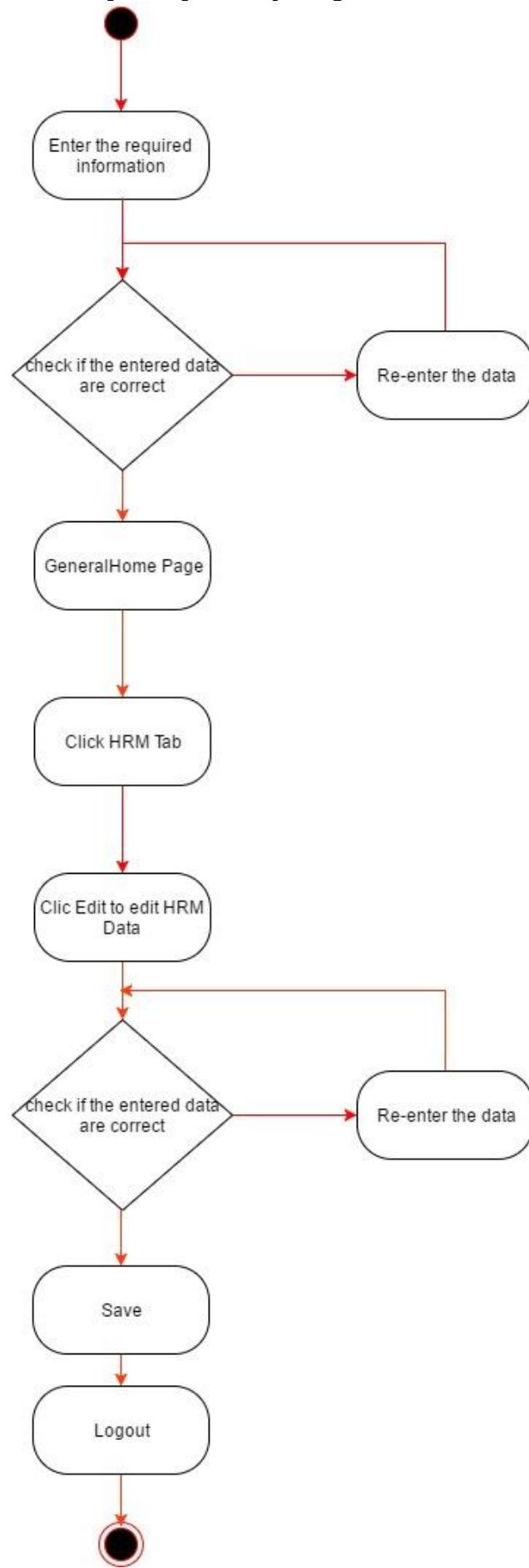
[TOPD] Activity Diagram



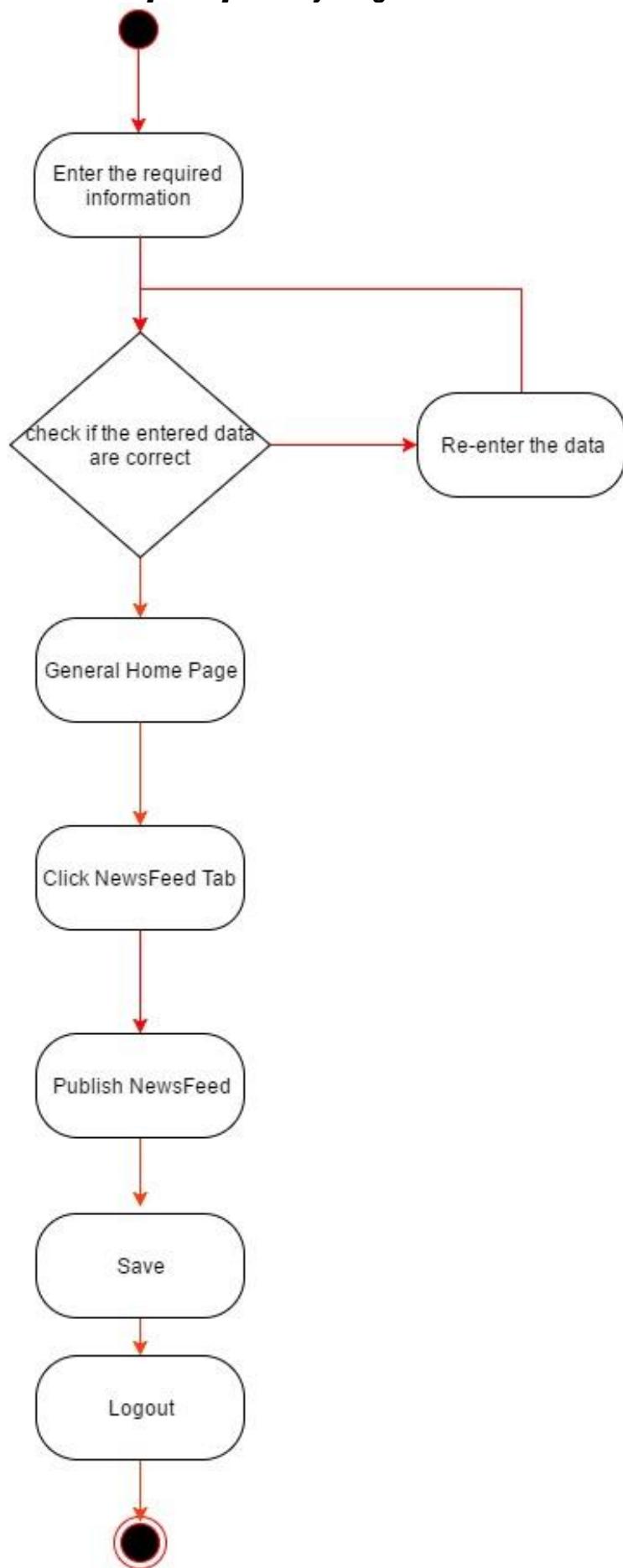
[TOPD] Activity Diagram



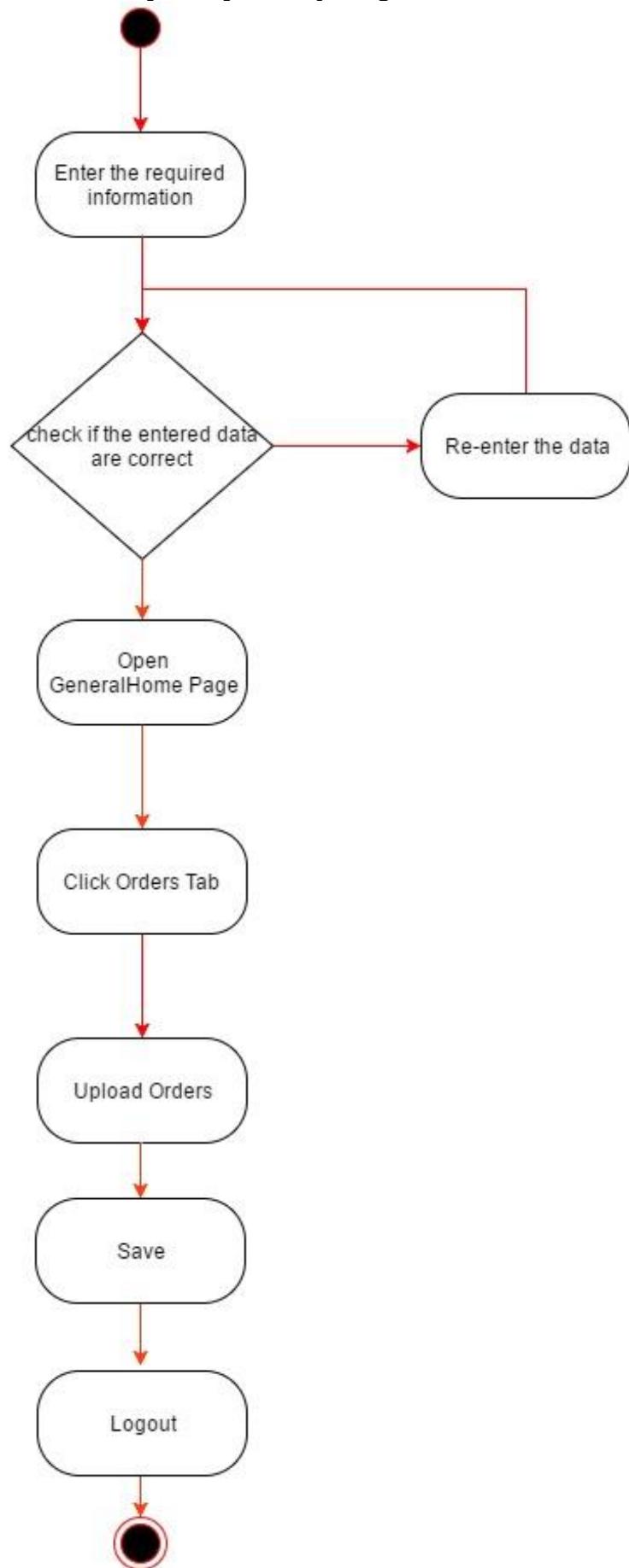
[TOPD] Activity Diagram



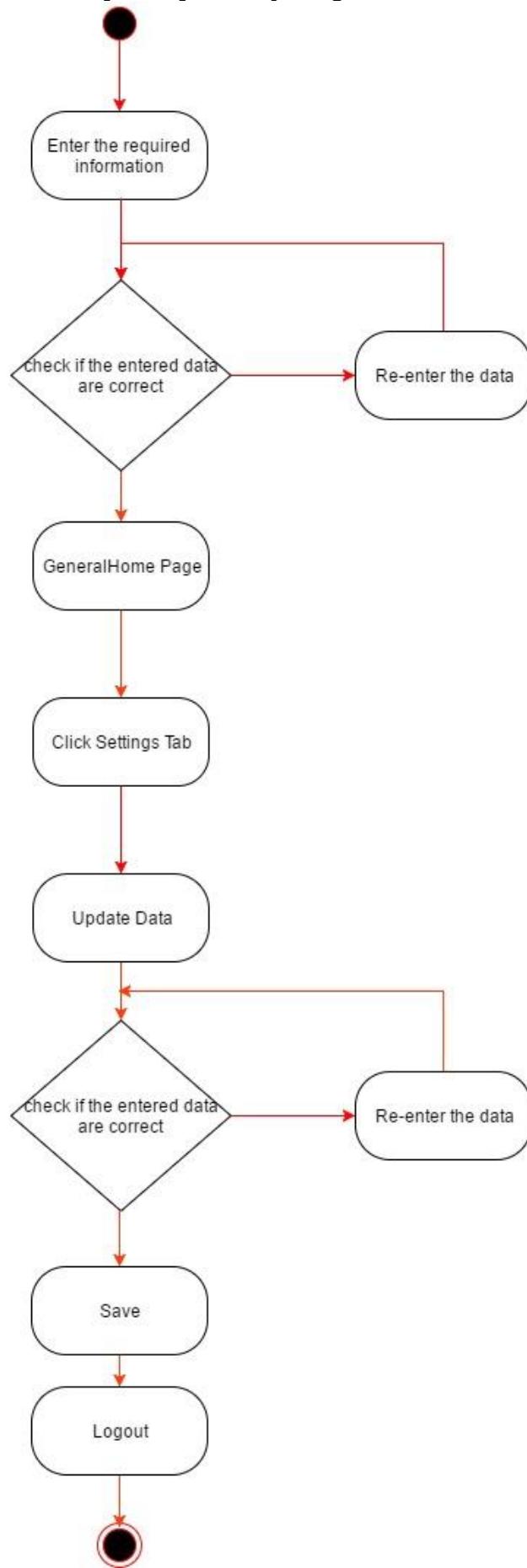
[TOPD] Activity Diagram



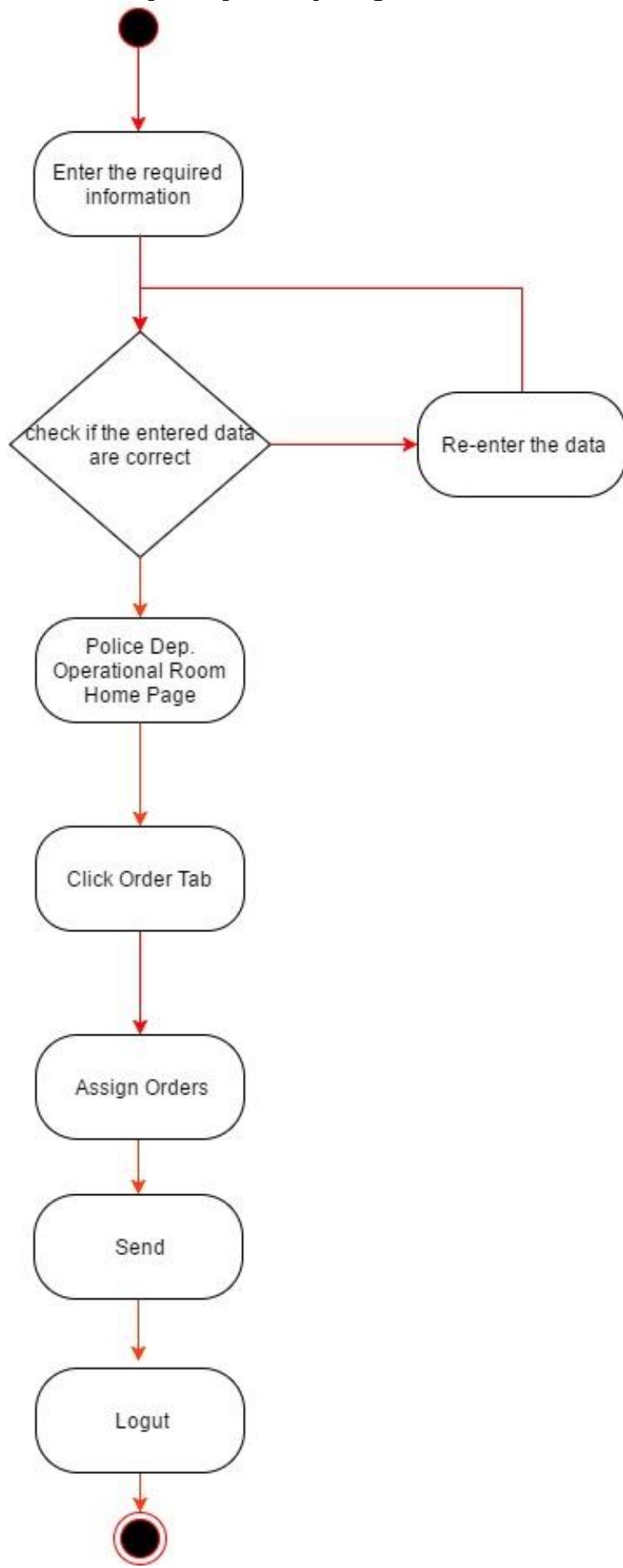
[TOPD] Activity Diagram



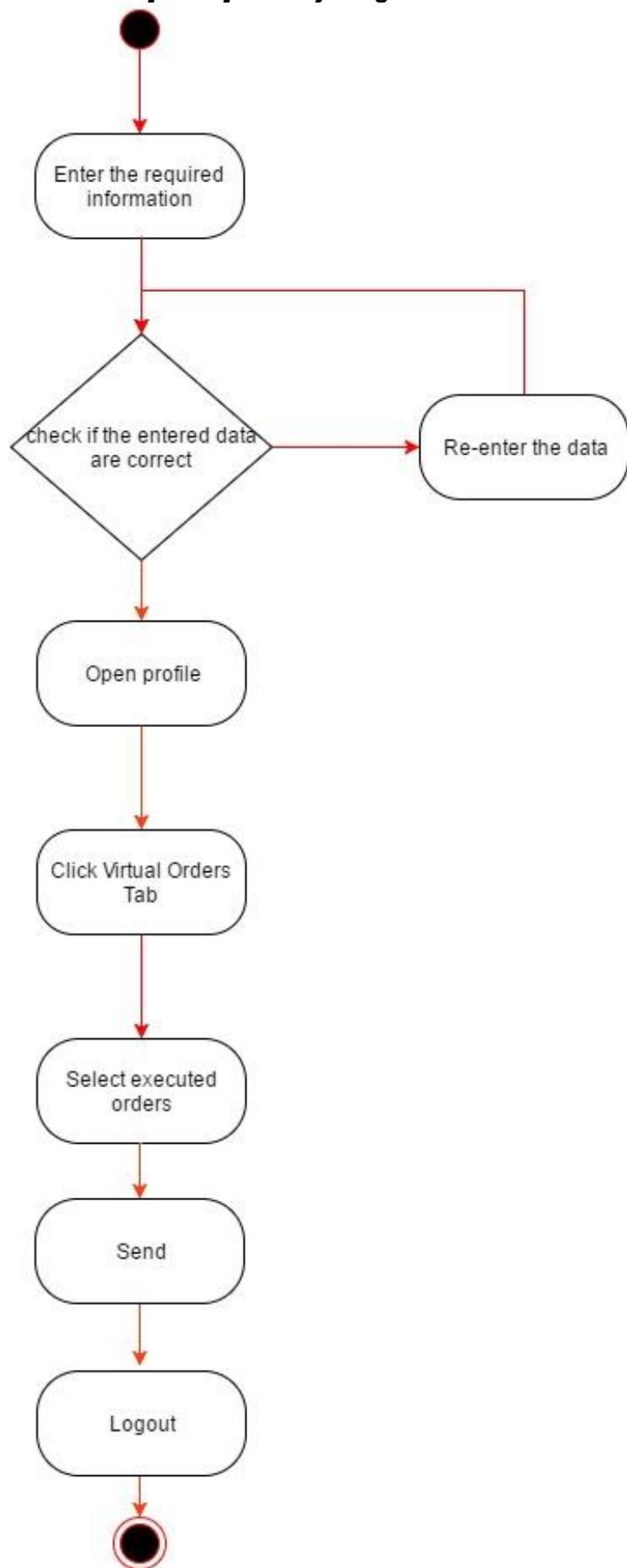
[TOPD] Activity Diagram



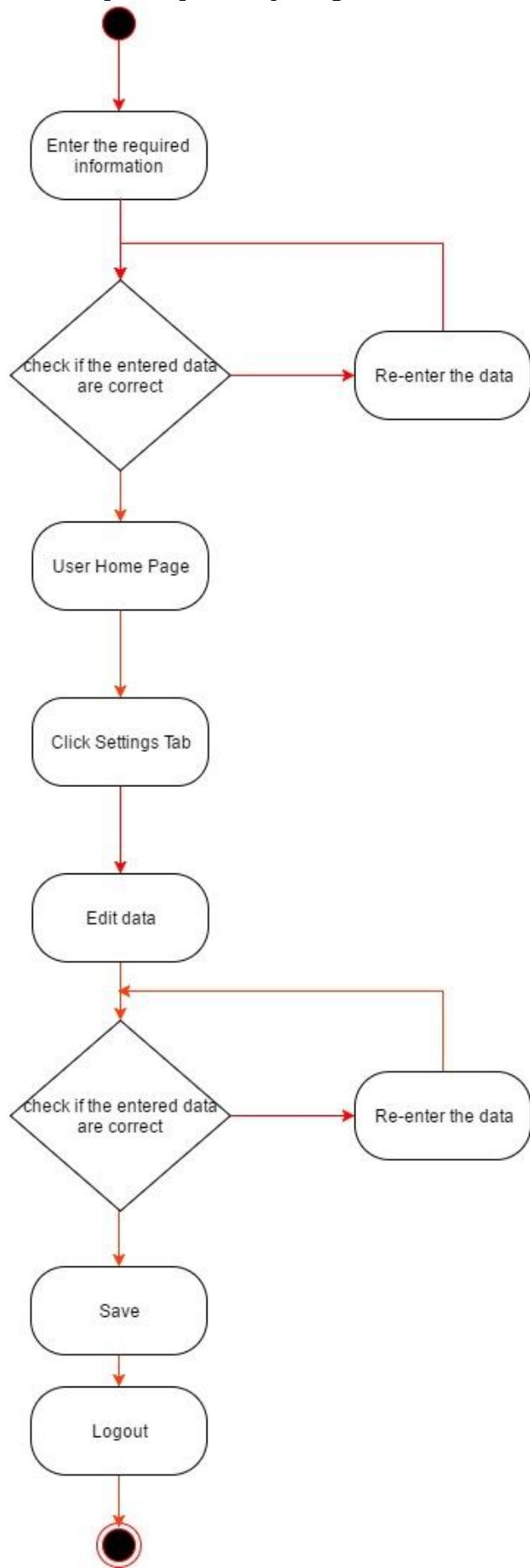
[TOPD] Activity Diagram



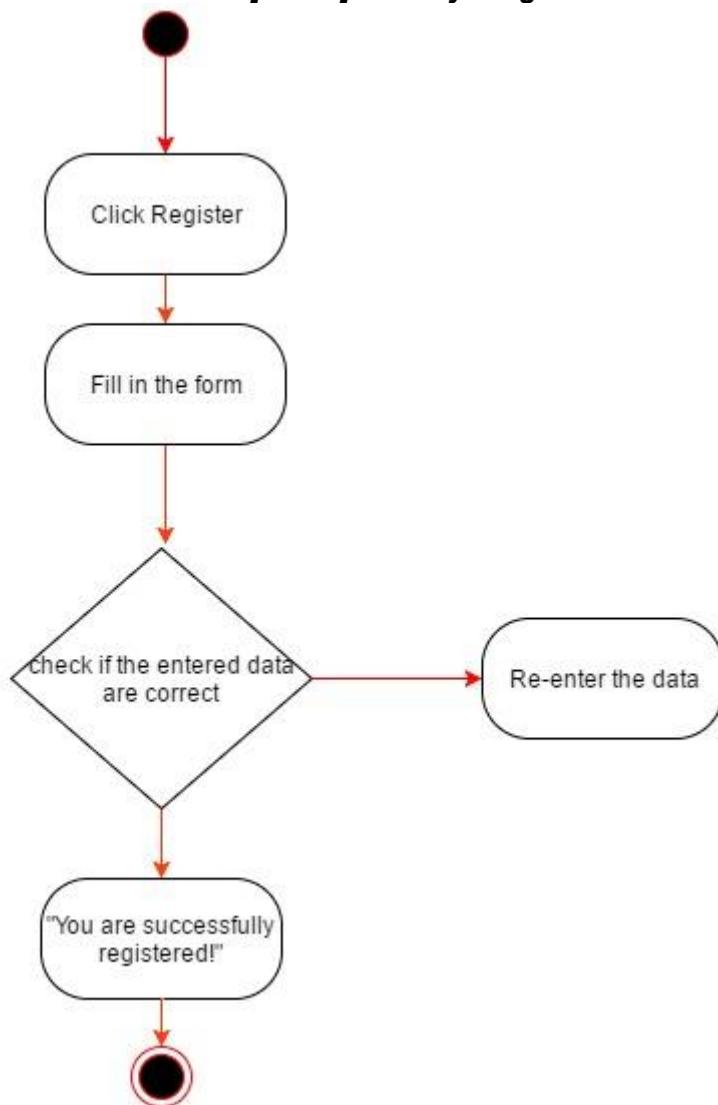
[TOPD] Activity Diagram



[TOPD] Activity Diagram

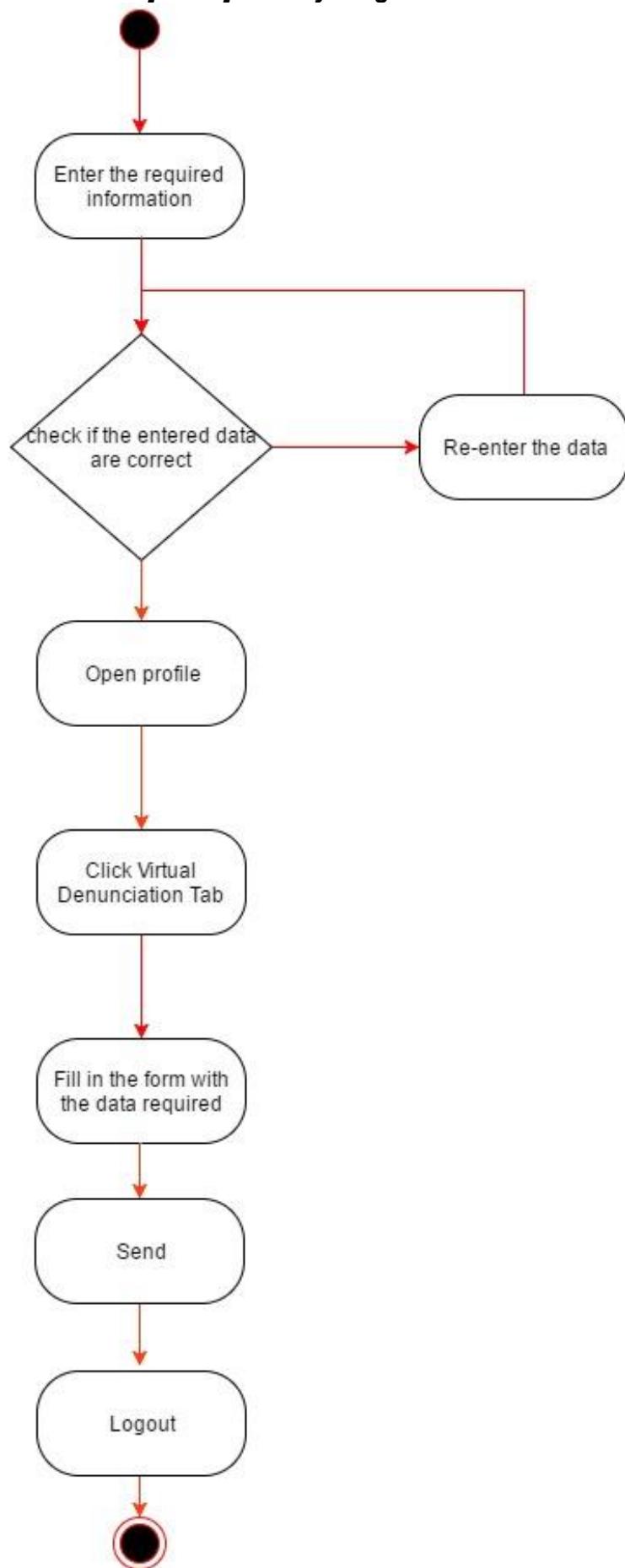


[TOPD] Activity Diagram

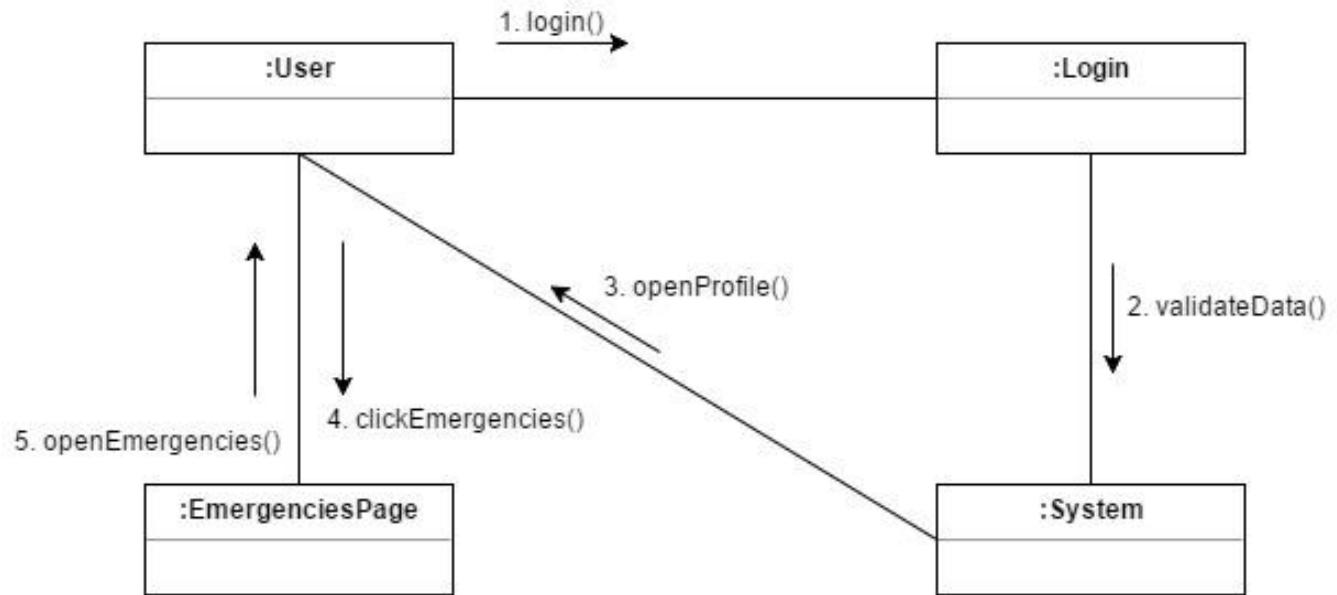


[TOPD] Activity Diagram

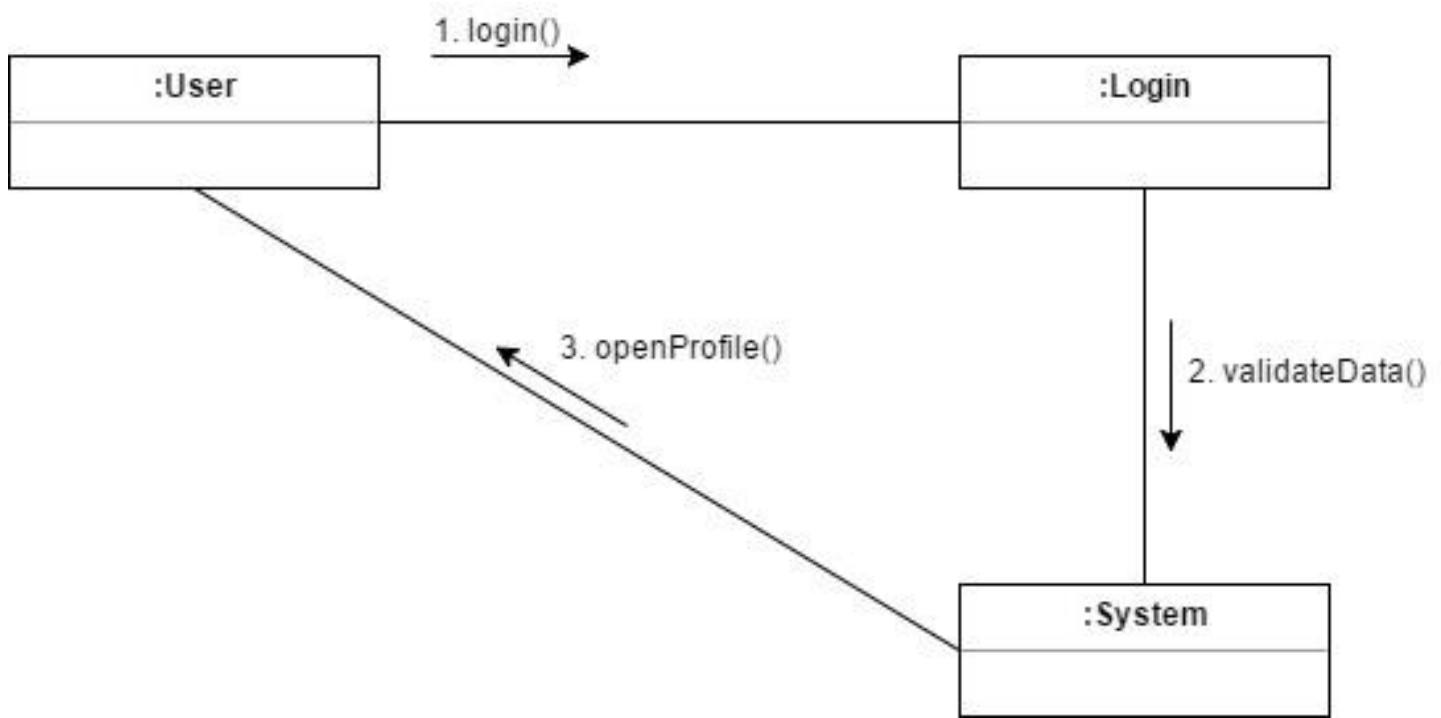
[TOPD] Activity Diagram



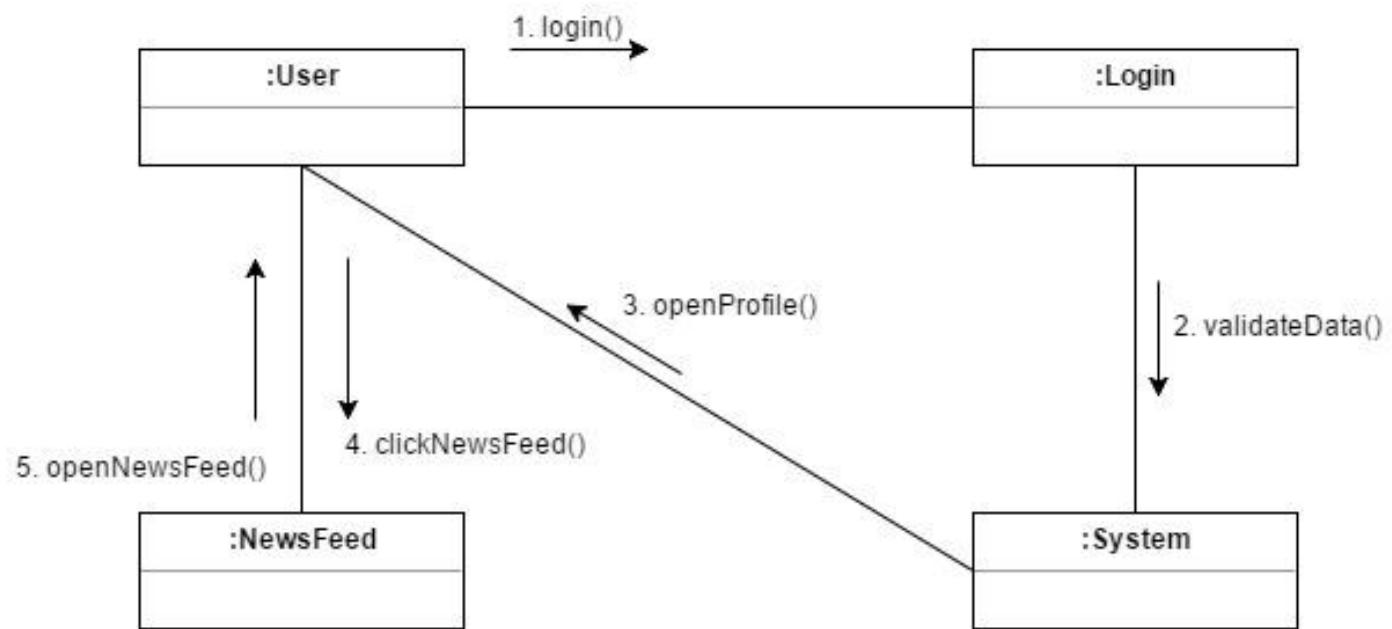
[TOPD] Collaboration Diagram



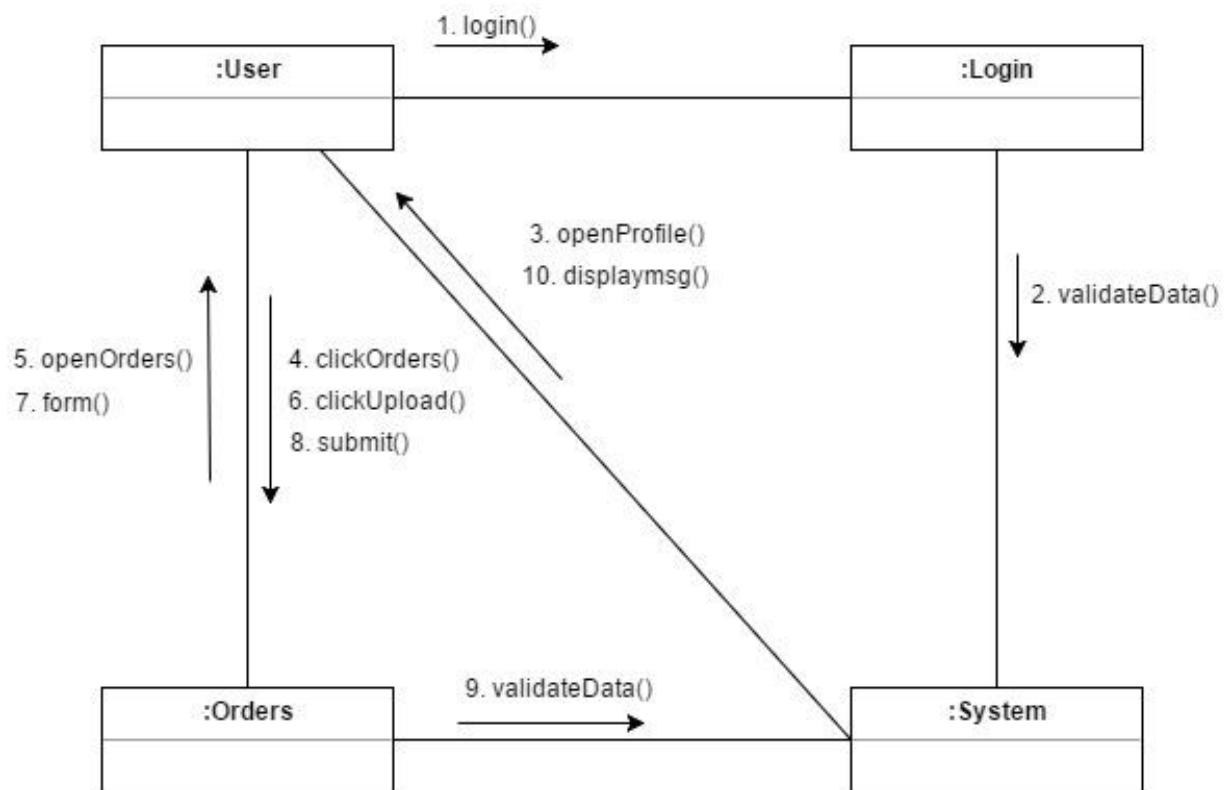
[TOPD] Collaboration Diagram



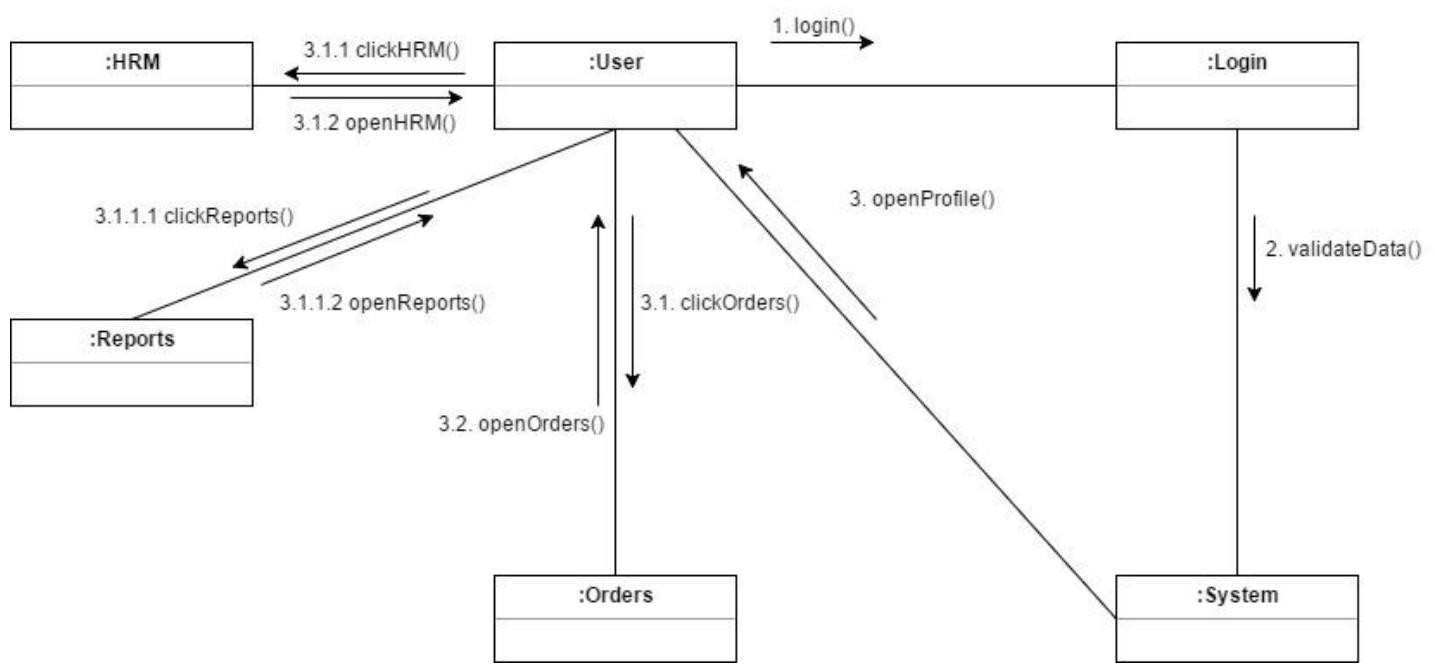
[TOPD] Collaboration Diagram



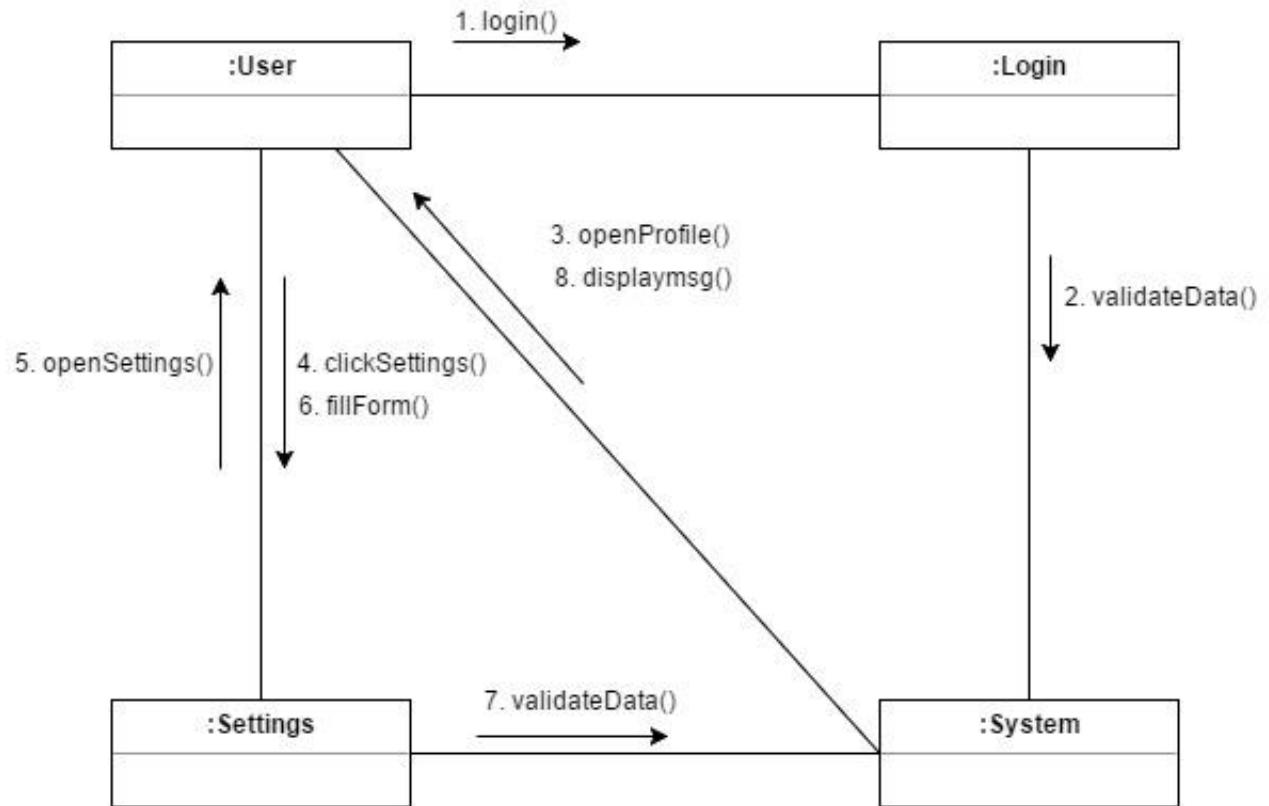
[TOPD] Collaboration Diagram



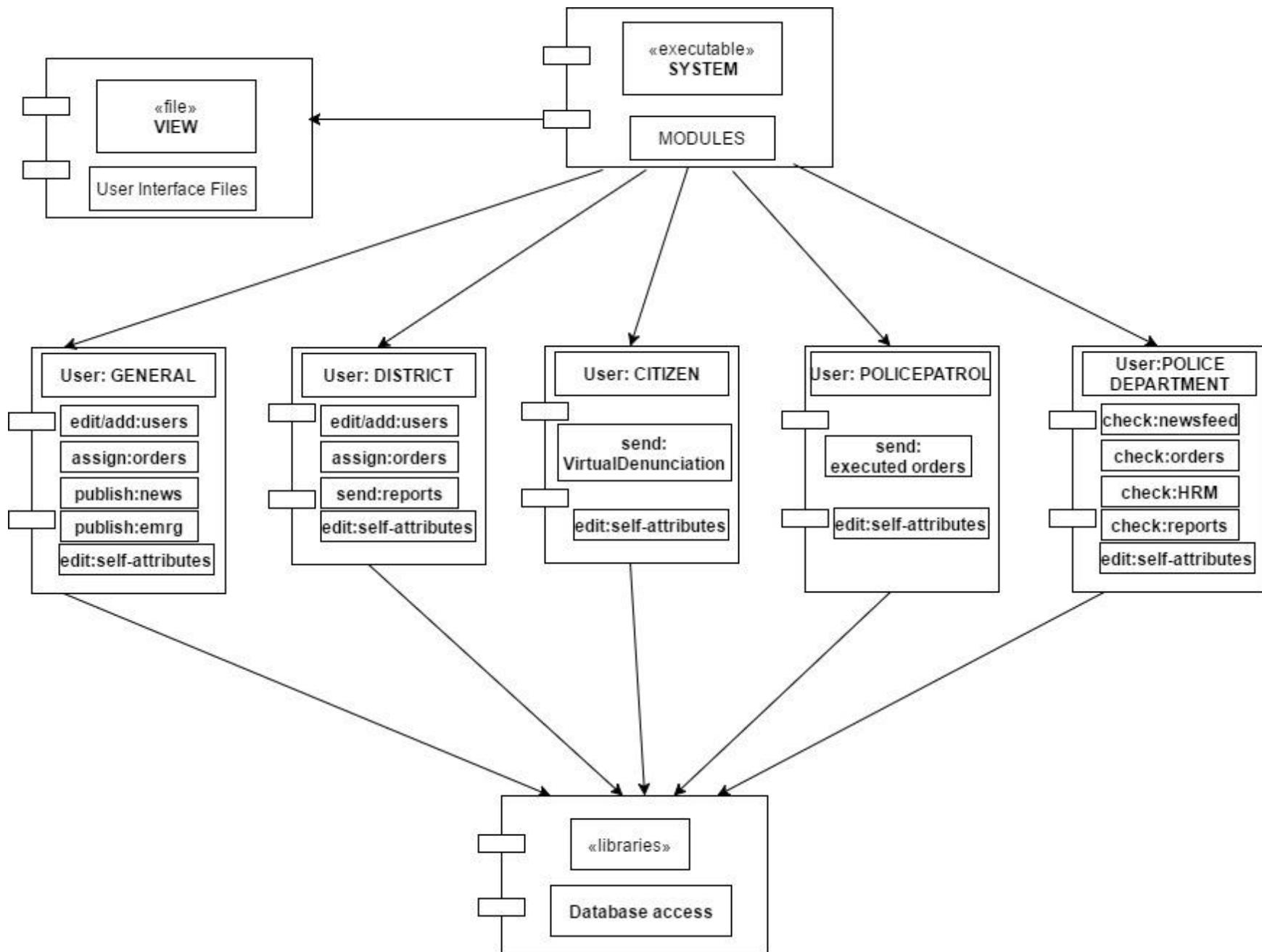
[TOPD] Collaboration Diagram



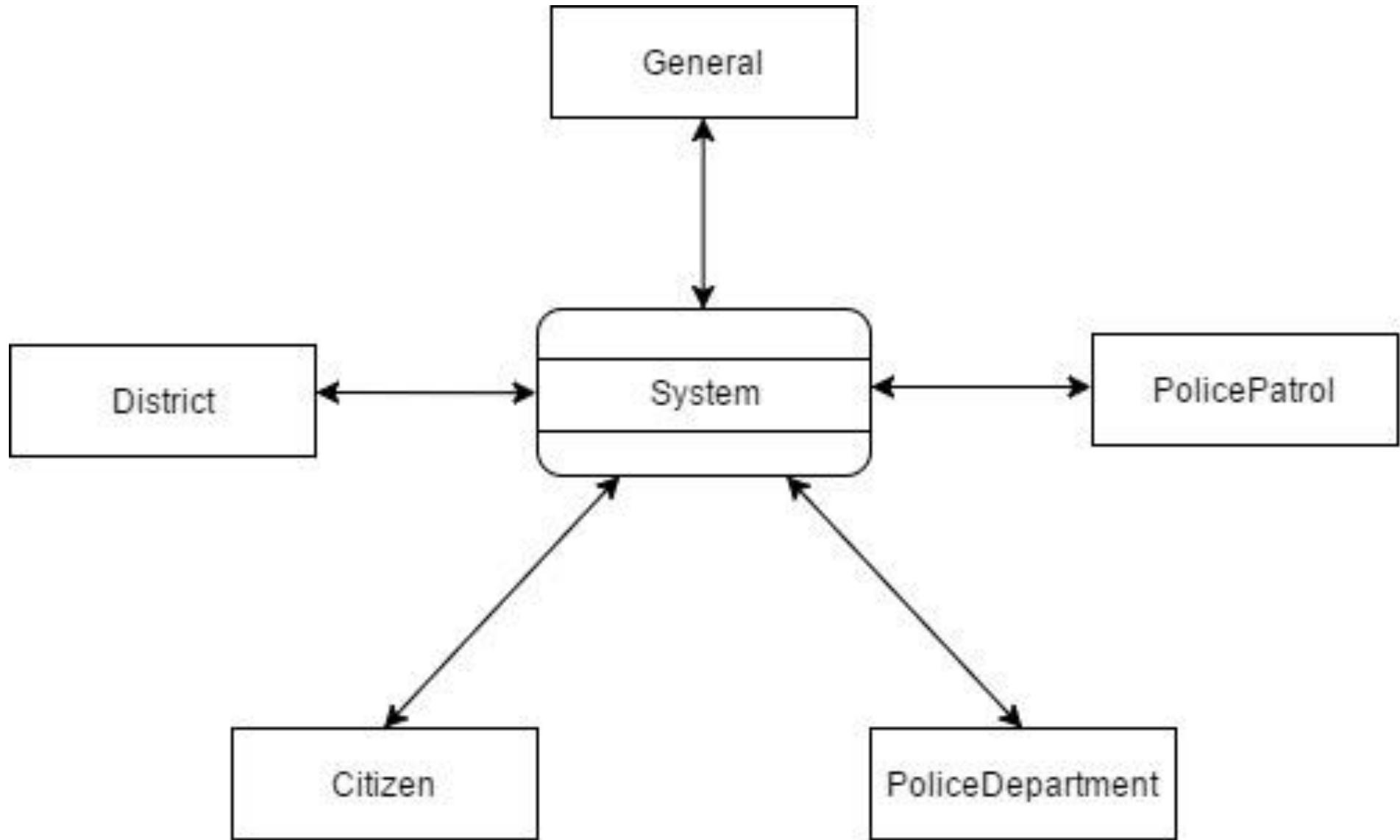
[TOPD] Collaboration Diagram



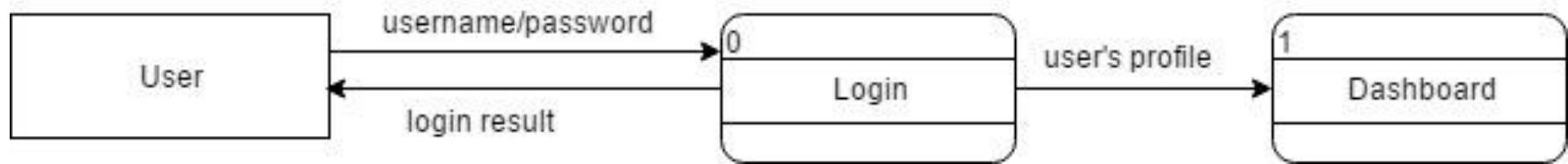
[TOPD] Component Diagram



[TOPD] Data Flow Diagram

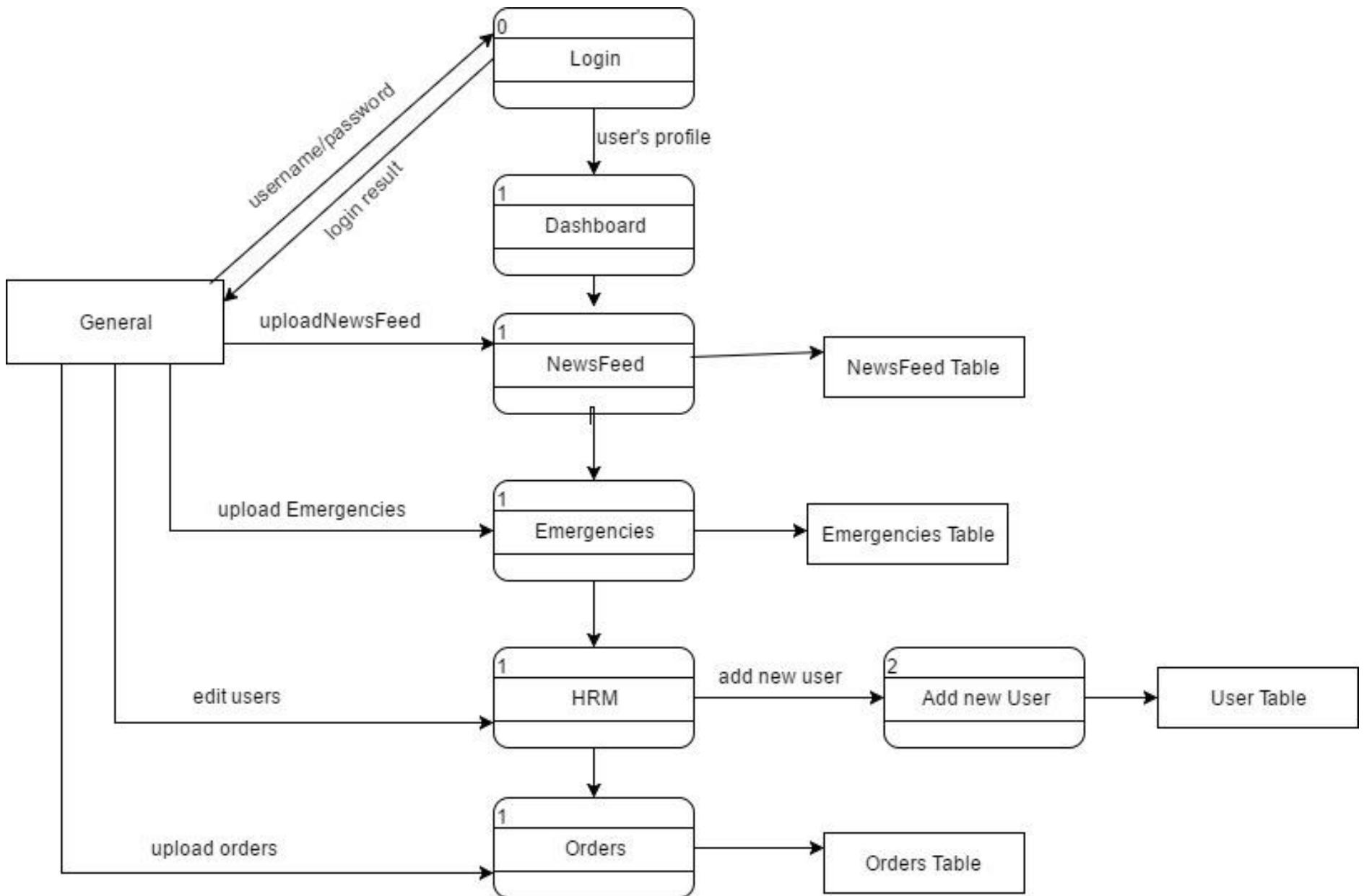


[TOPD] Data Flow Diagram



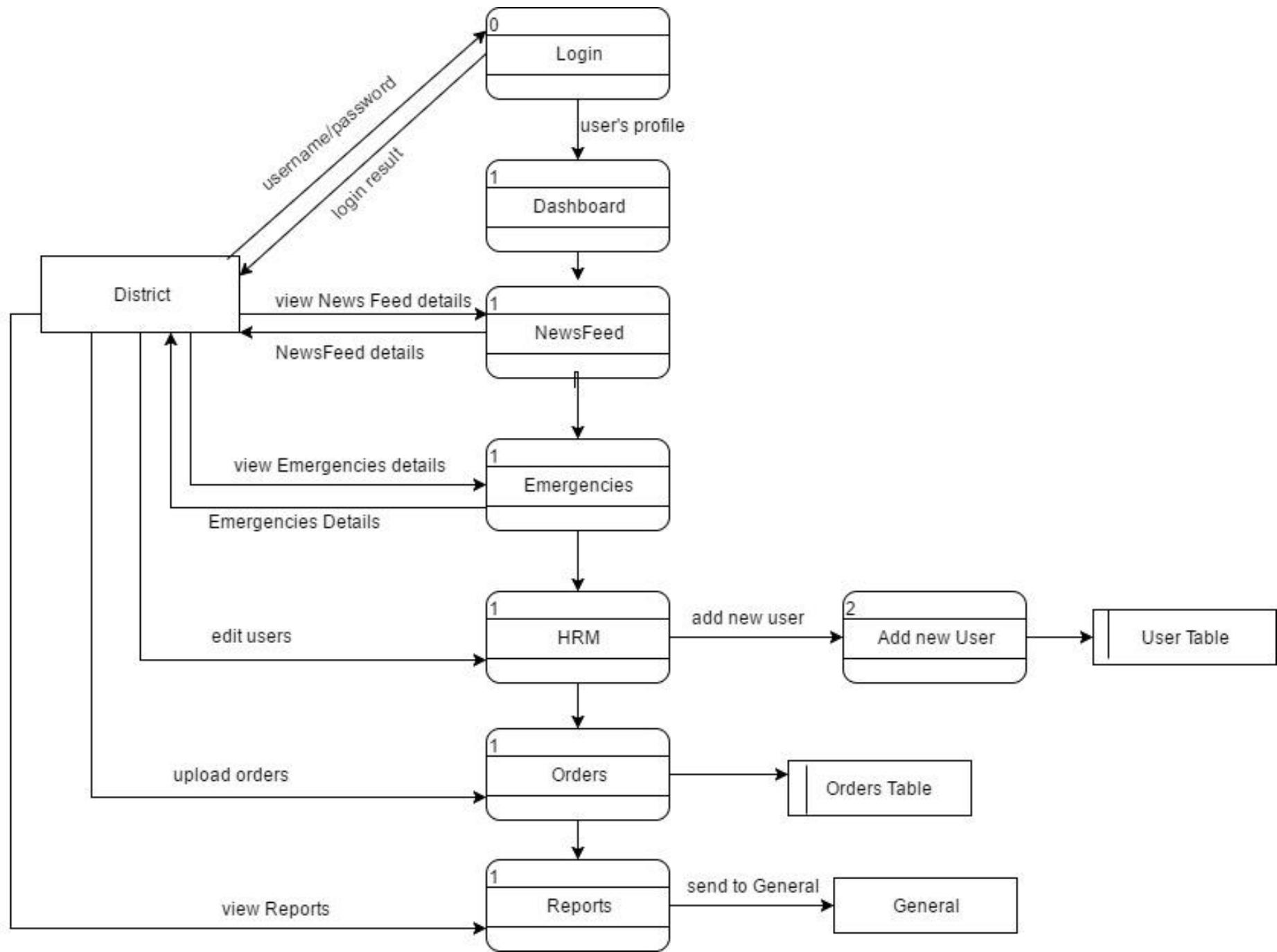
[TOPD] Data Flow Diagram

[TOPD] Data Flow Diagram

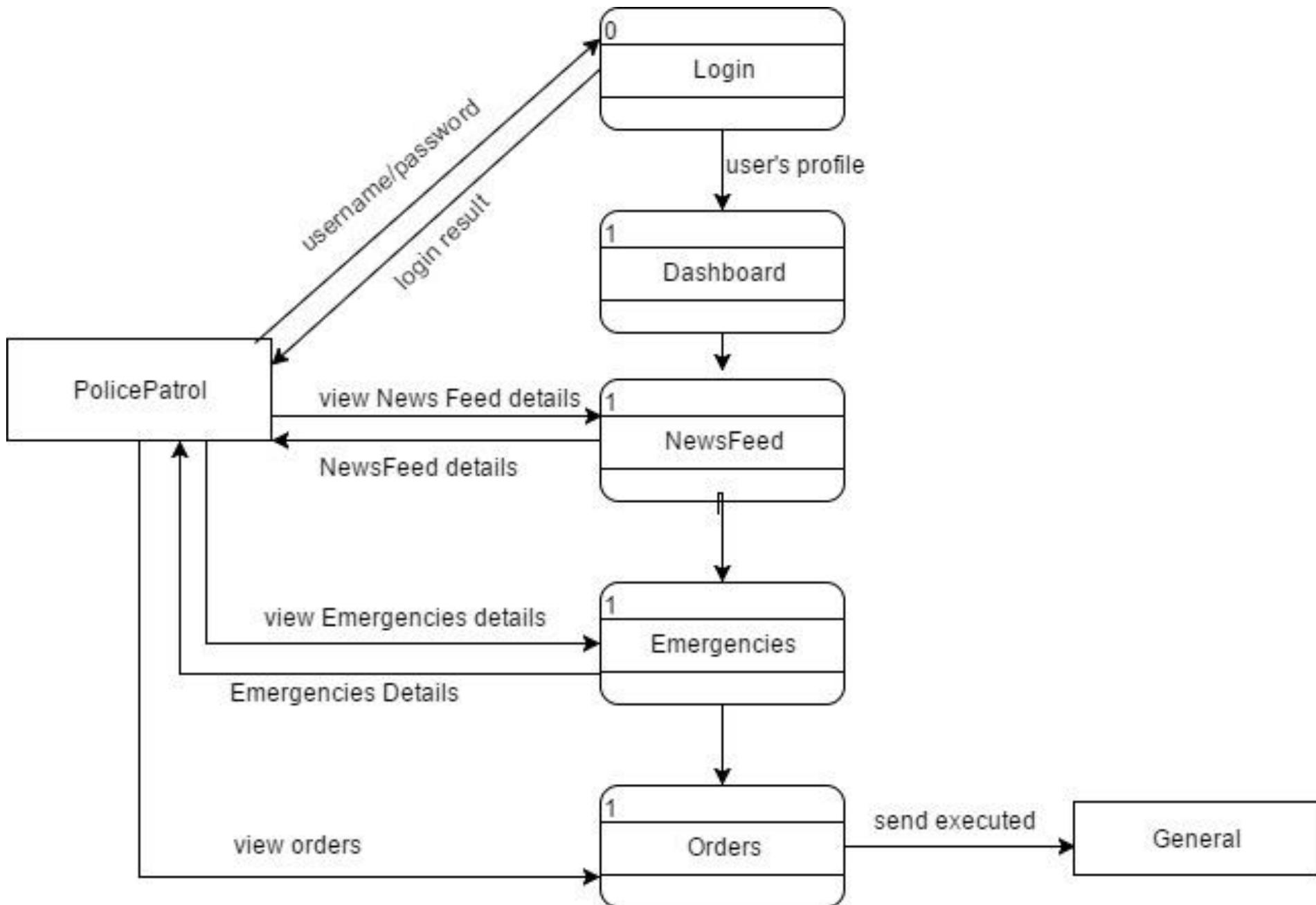


[TOPD] Data Flow Diagram

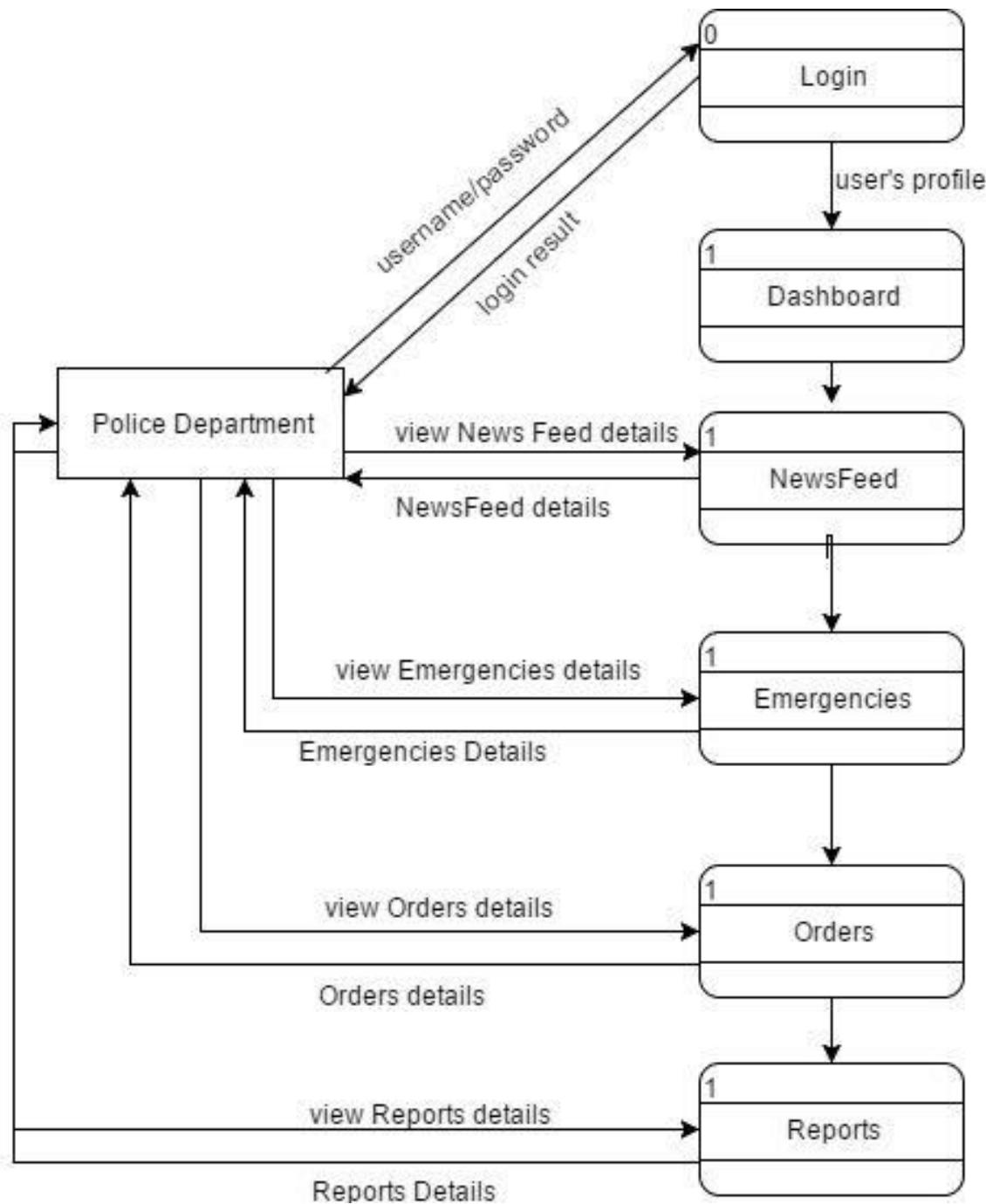
[TOPD] Data Flow Diagram



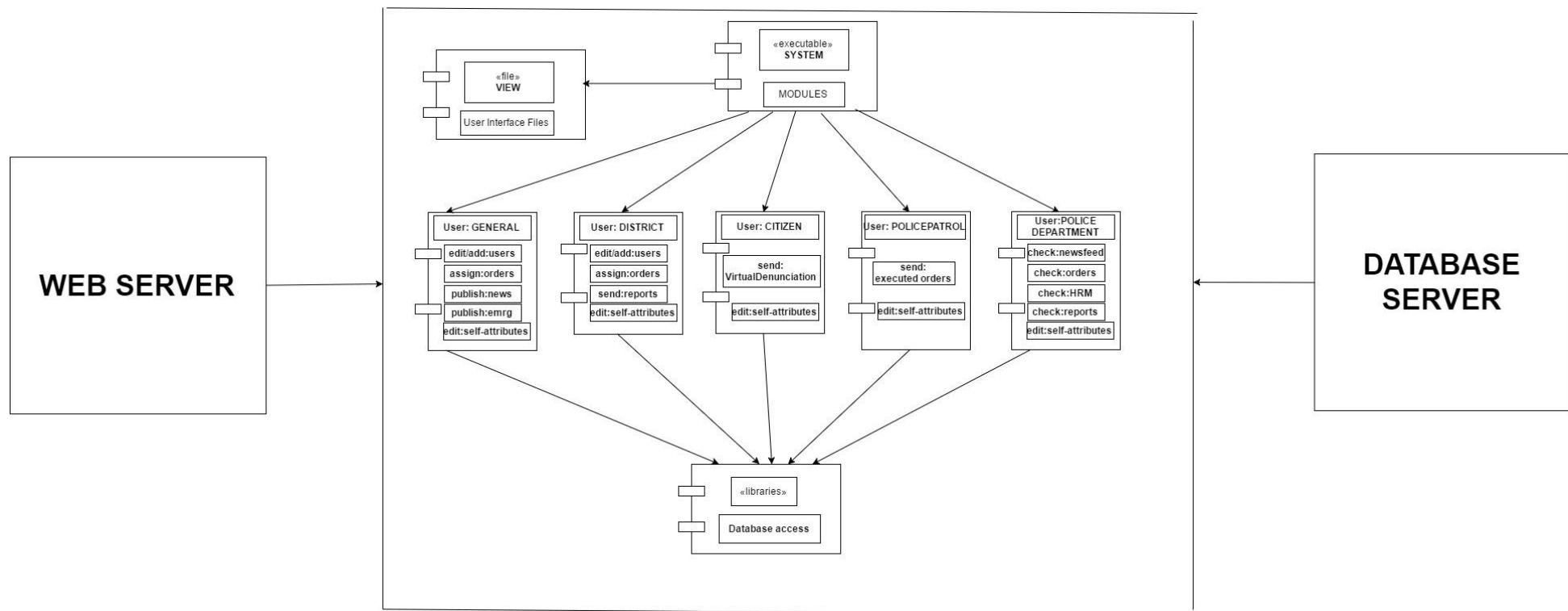
[TOPD] Data Flow Diagram



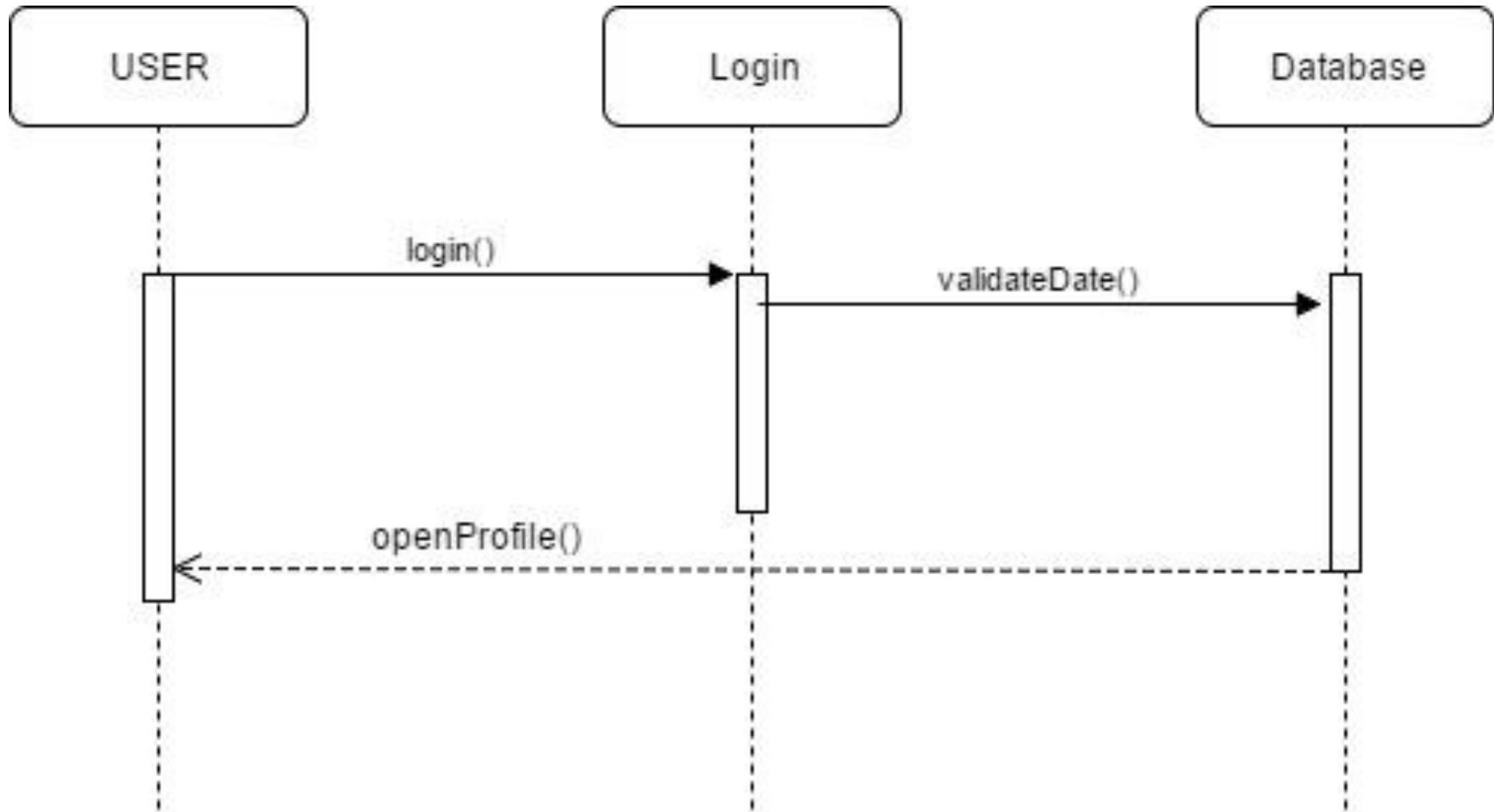
[TOPD] Data Flow Diagram



[TOPD] Deployment Diagram

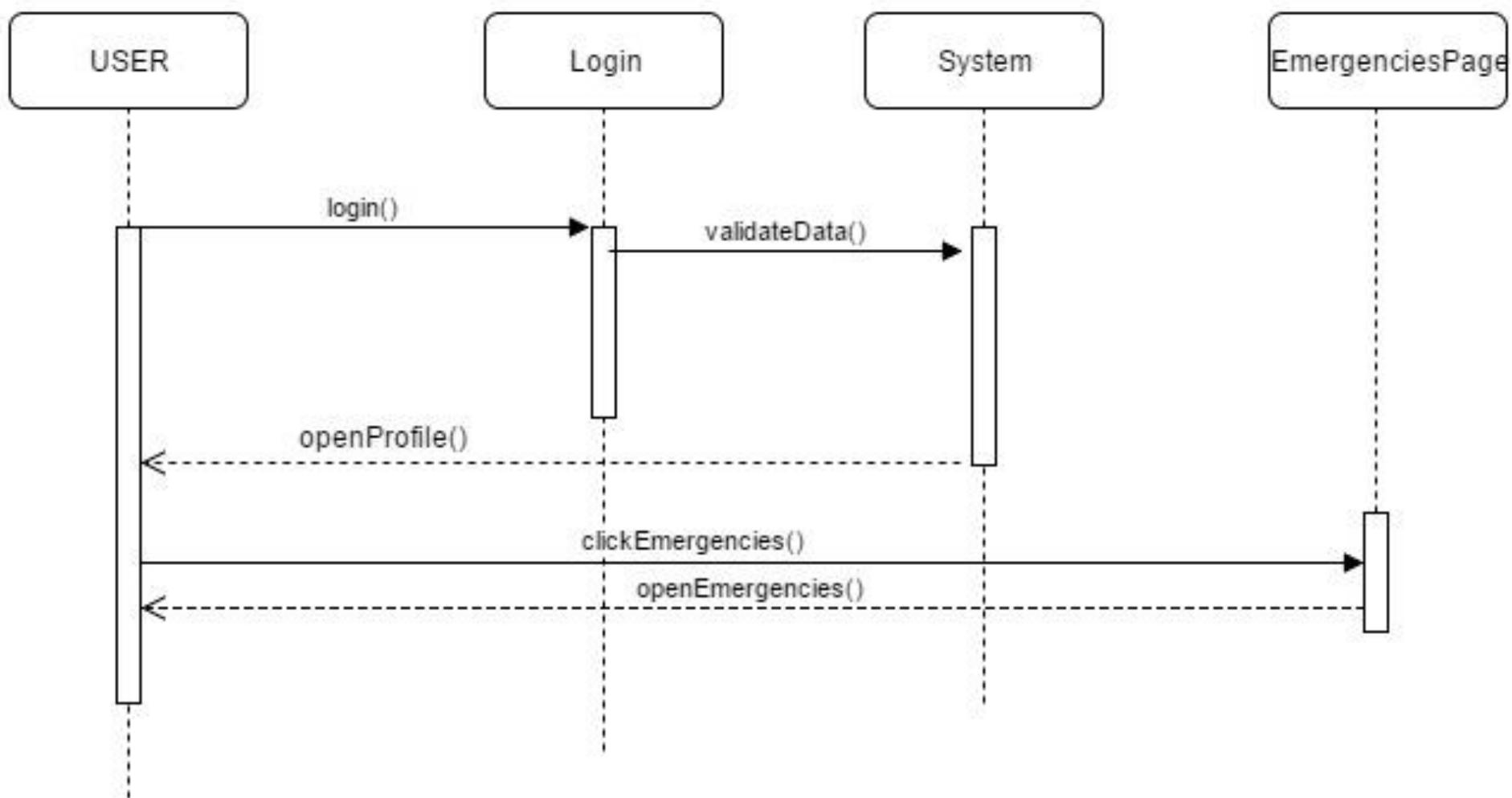


[TOPD] Sequence Diagram



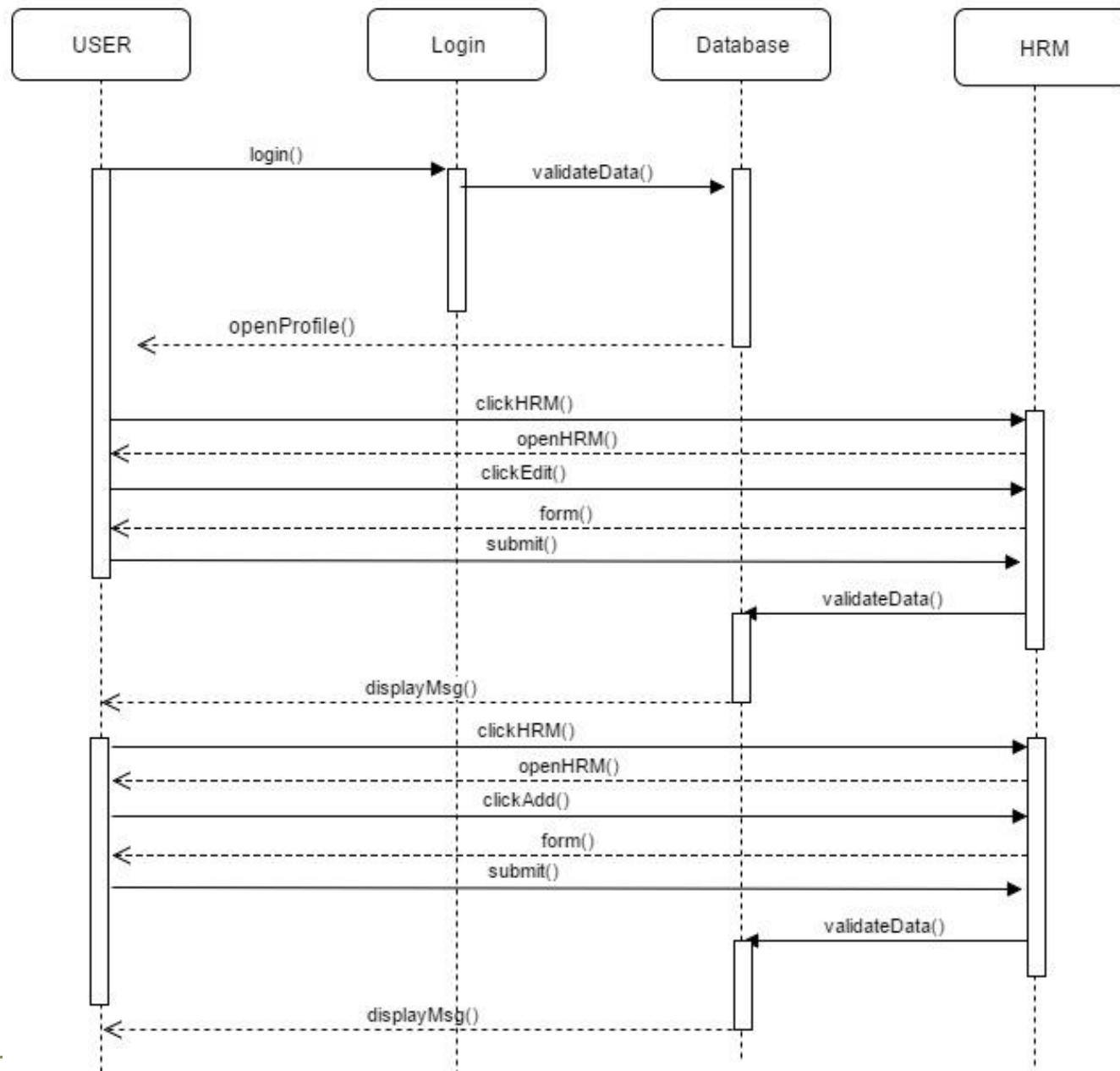
[TOPD] Sequence Diagram

User(Citizen,District,PD,PP)



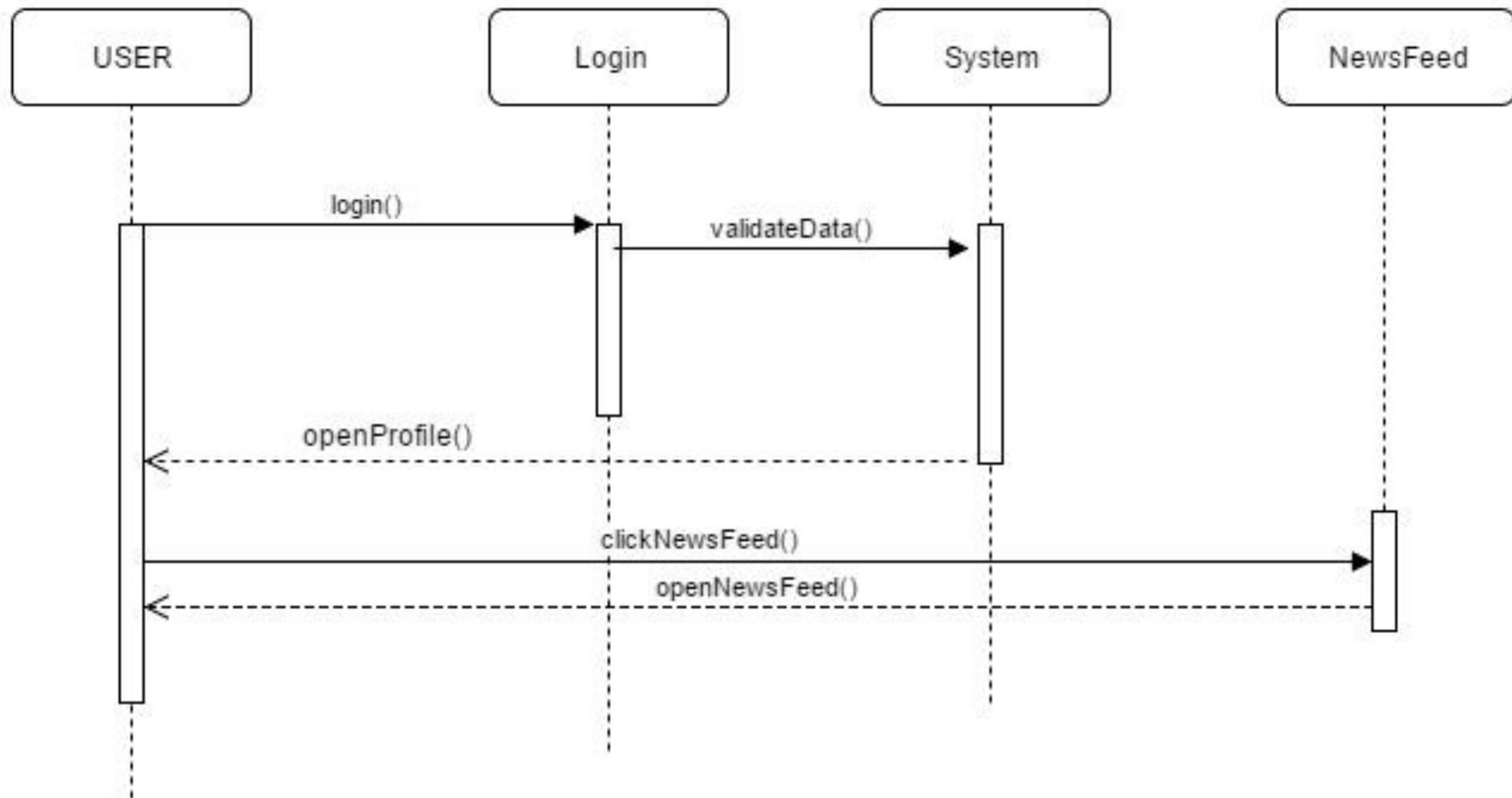
[TOPD] Sequence Diagram

User(District,General)



[TOPD] Sequence Diagram

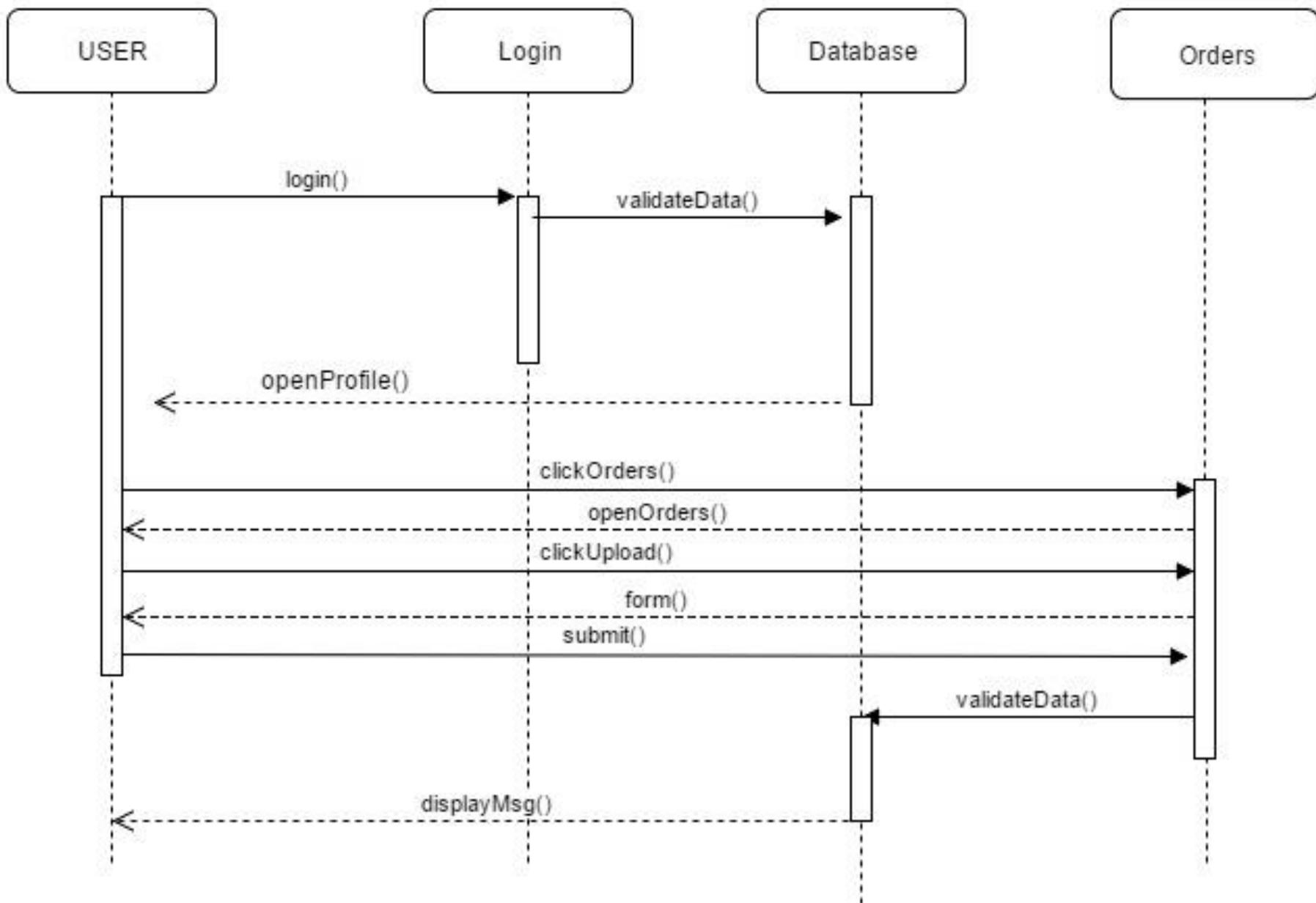
User(Citizen,District,PD,PP)



[TOPD] Sequence Diagram

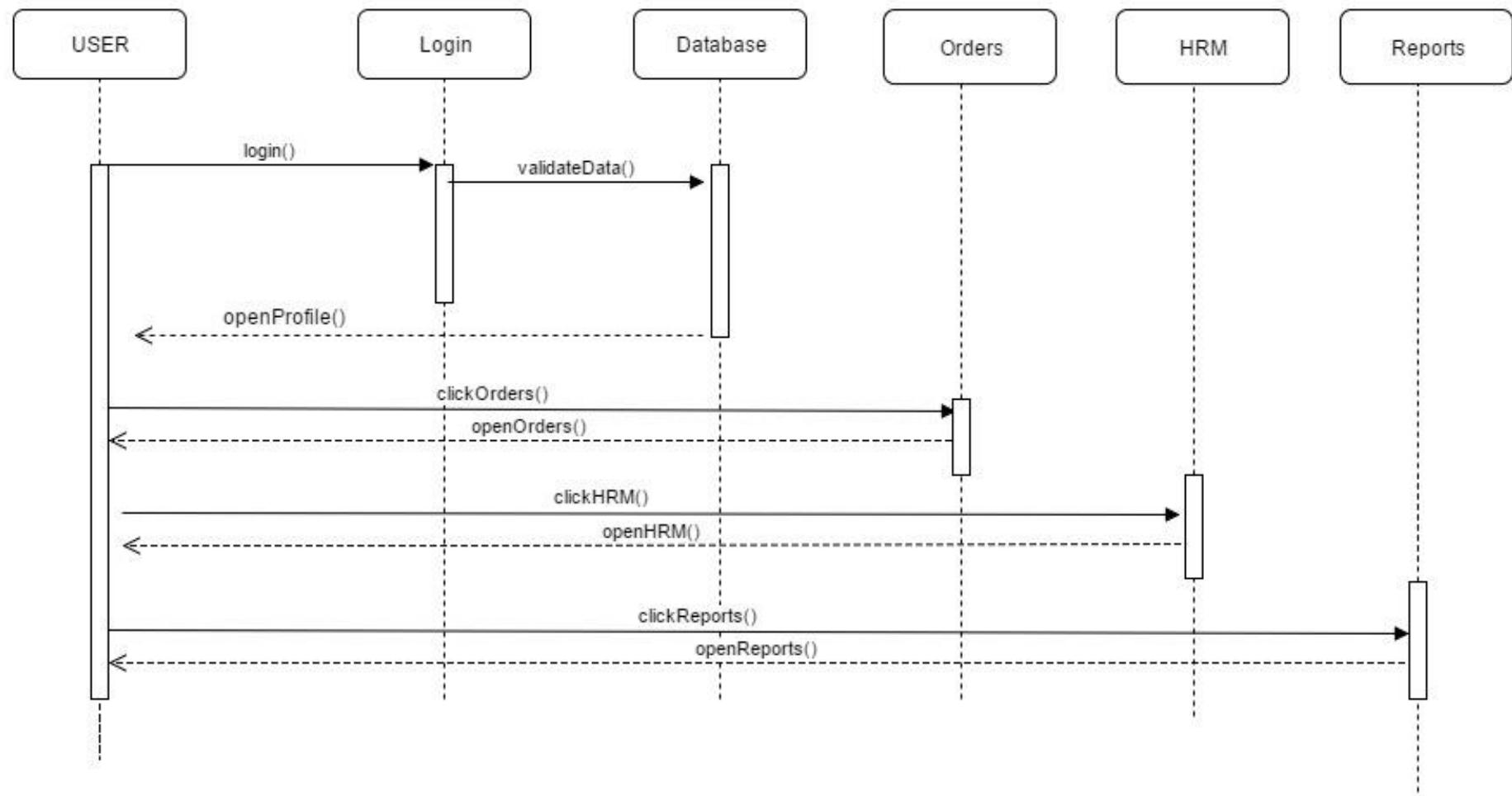
[TOPD] Sequence Diagram

User(District,General)

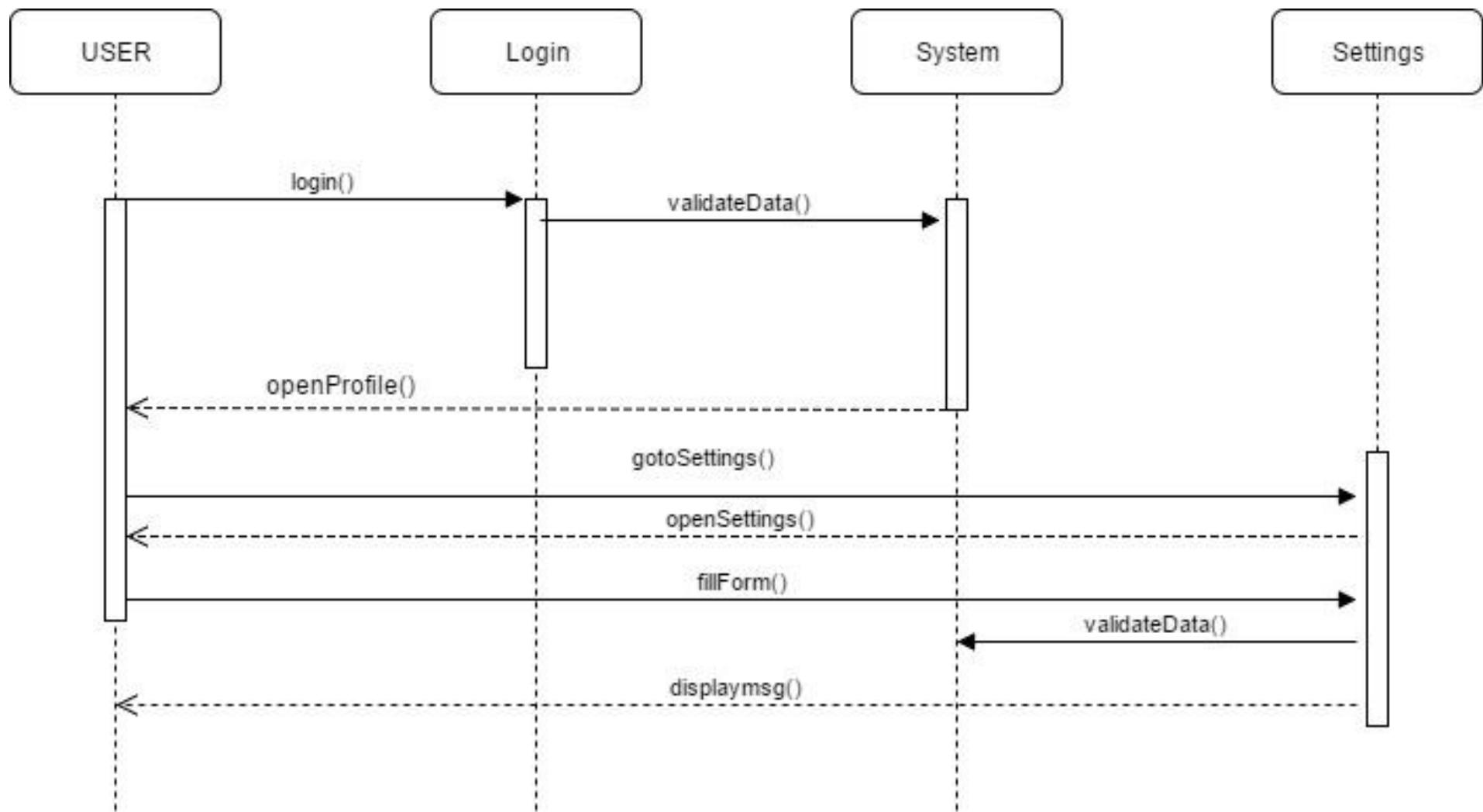


[TOPD] Sequence Diagram

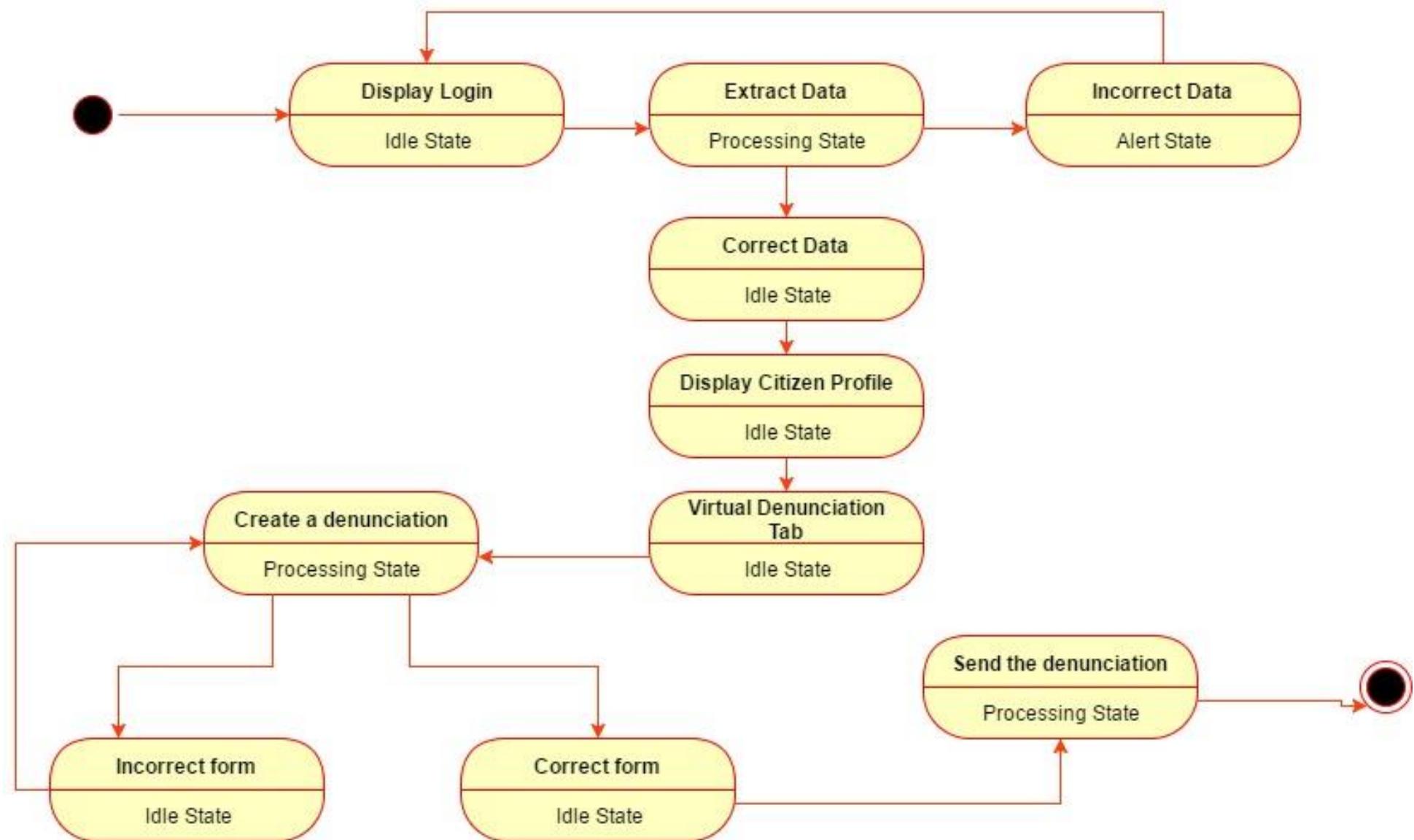
User(PoliceDepartment)



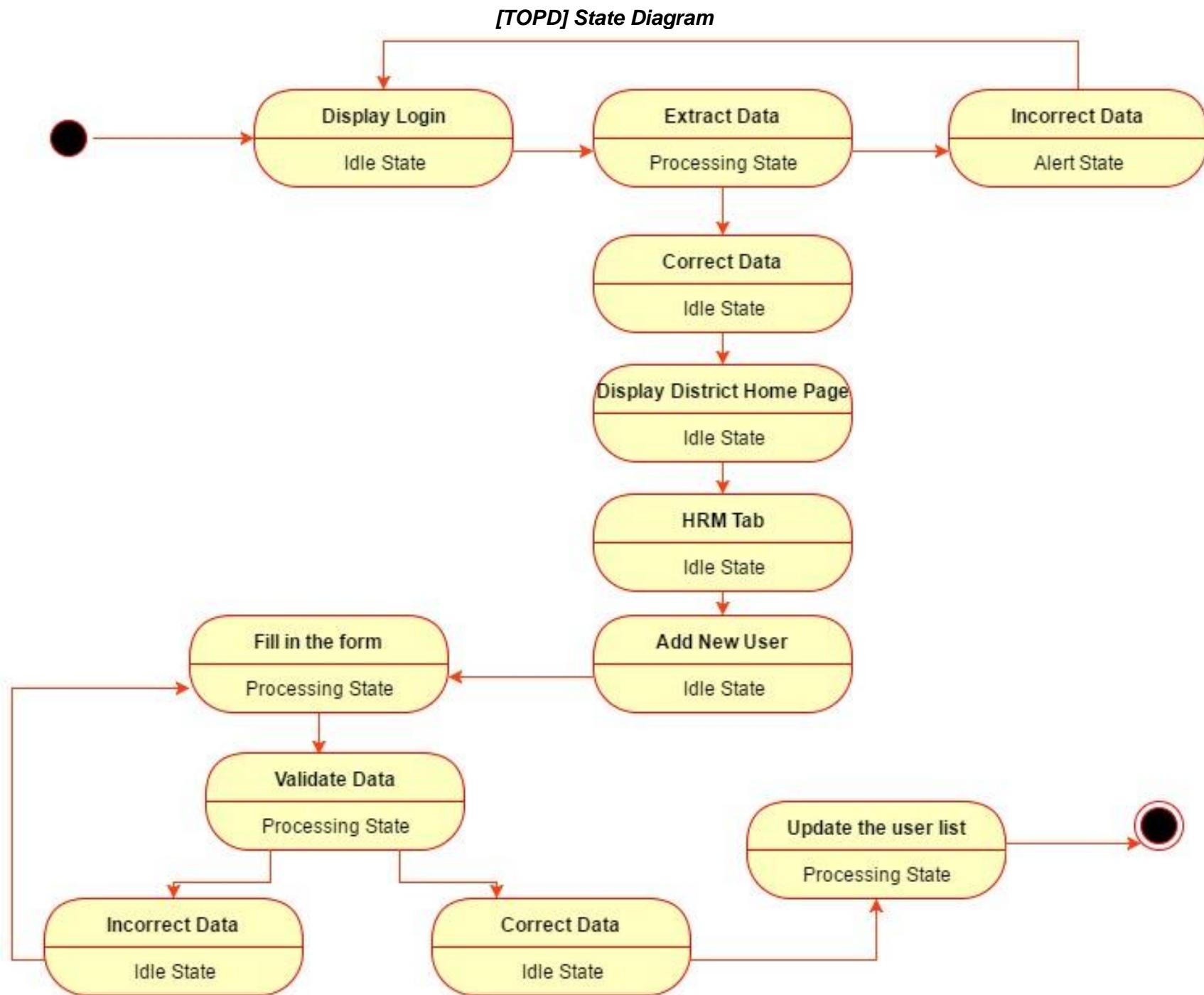
[TOPD] Sequence Diagram

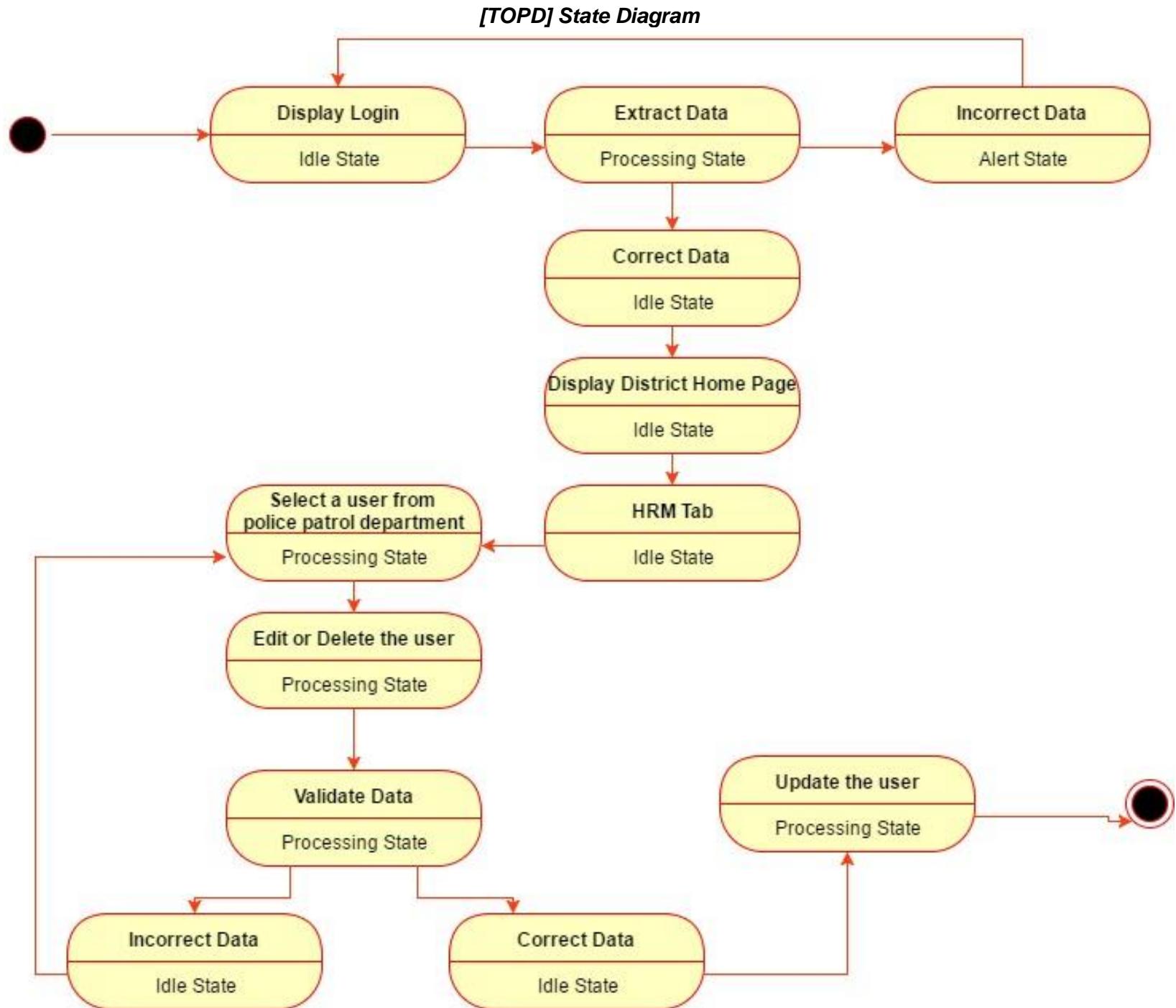


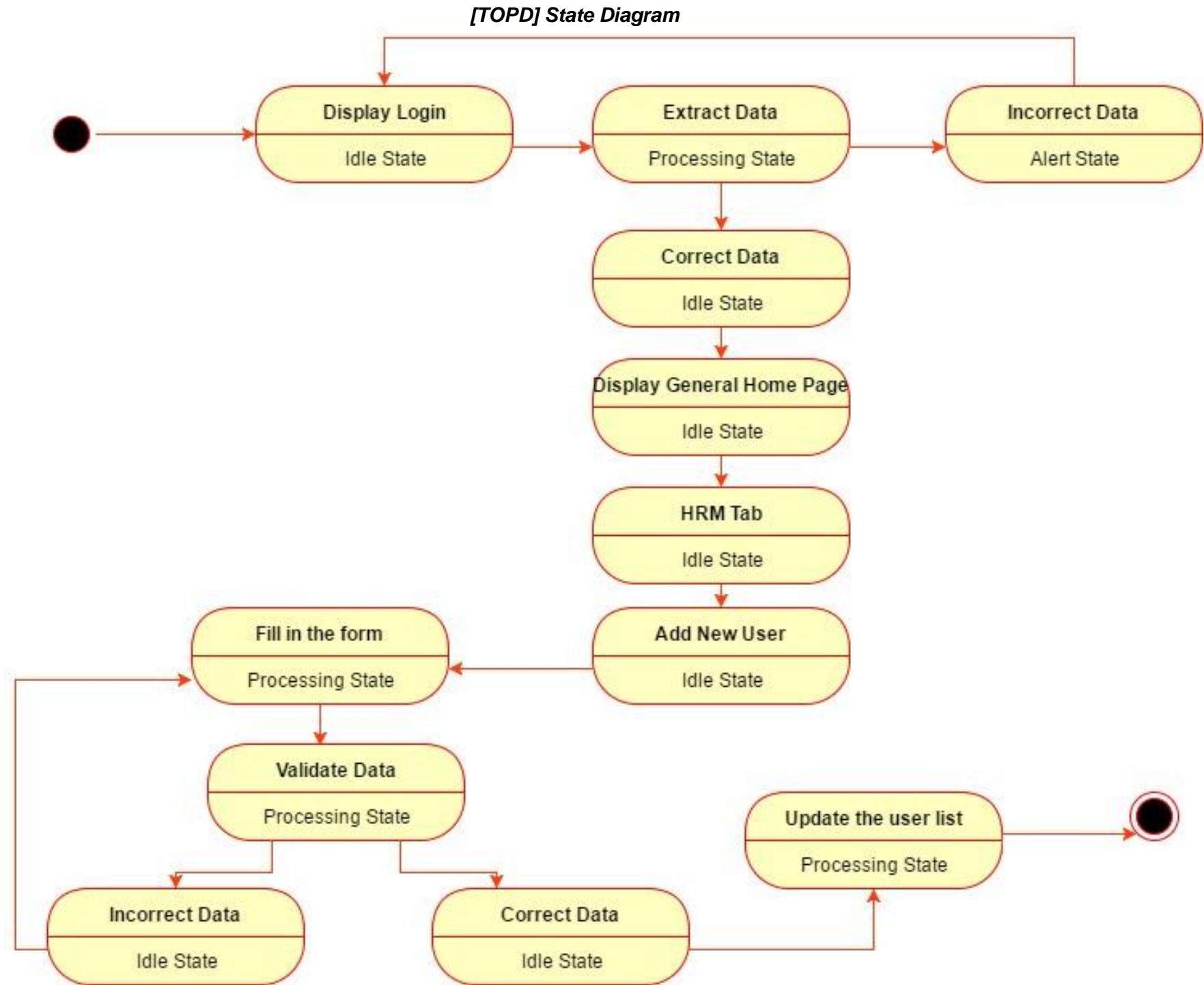
[TOPD] State Diagram



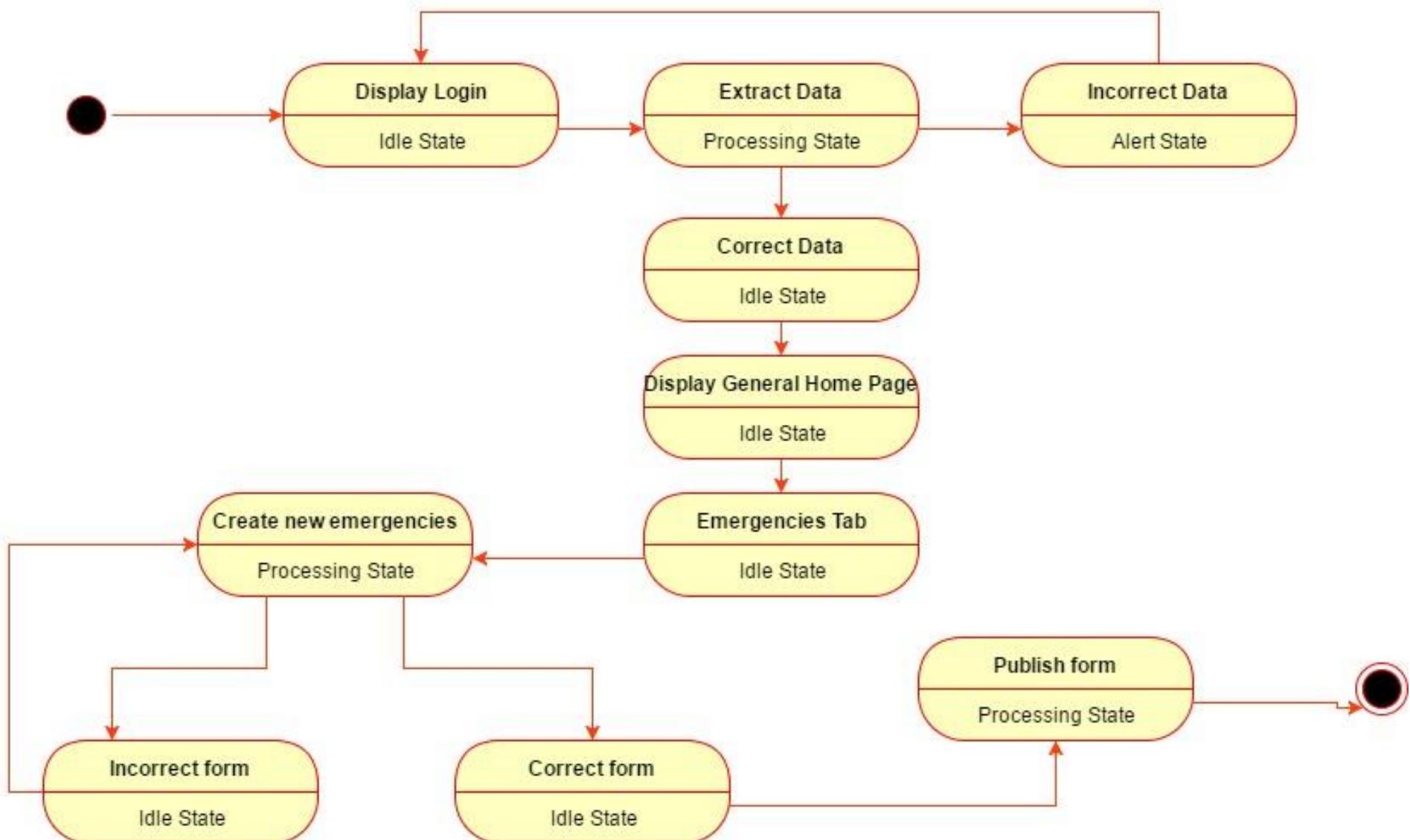
[TOPD] State Diagram

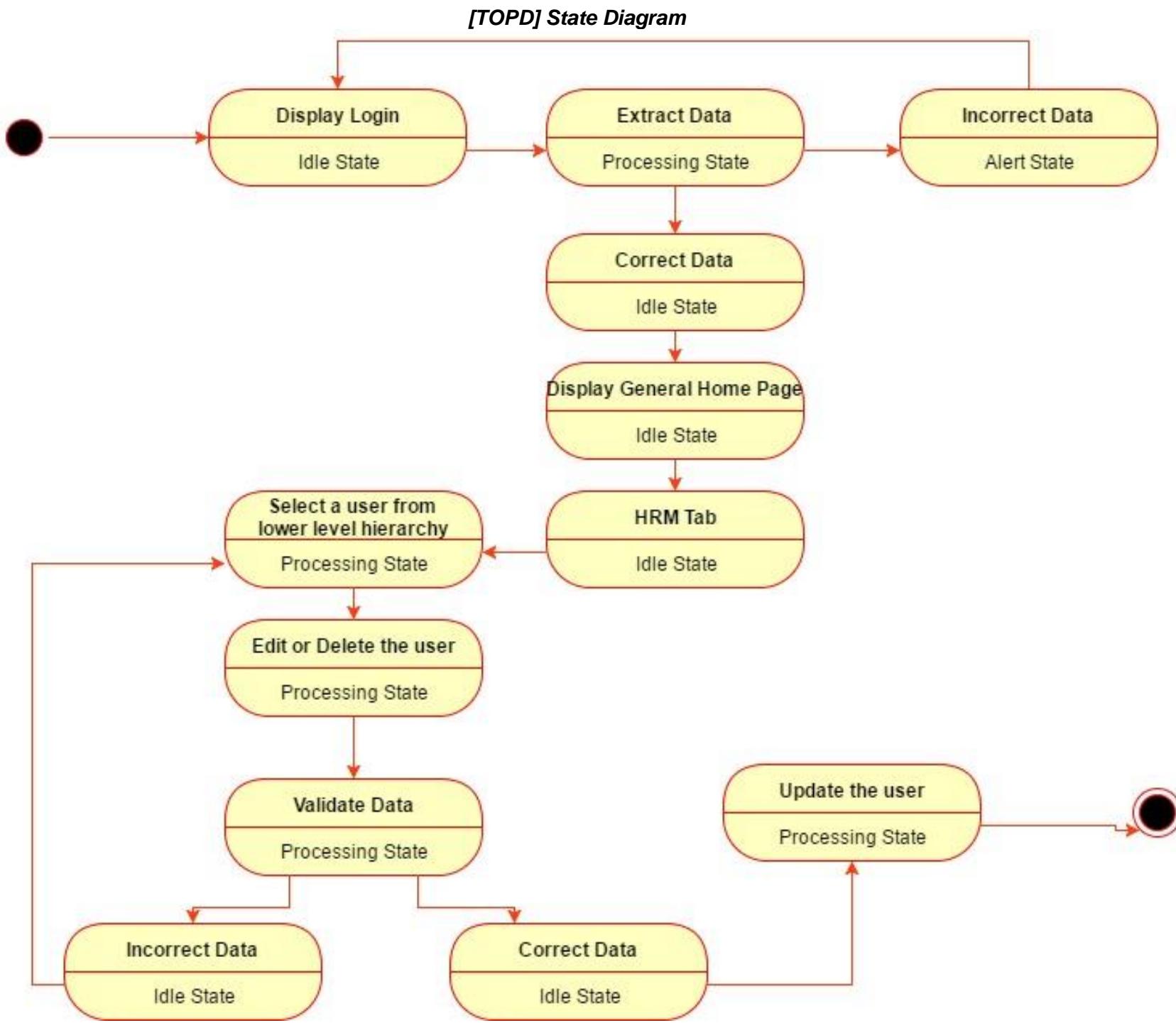




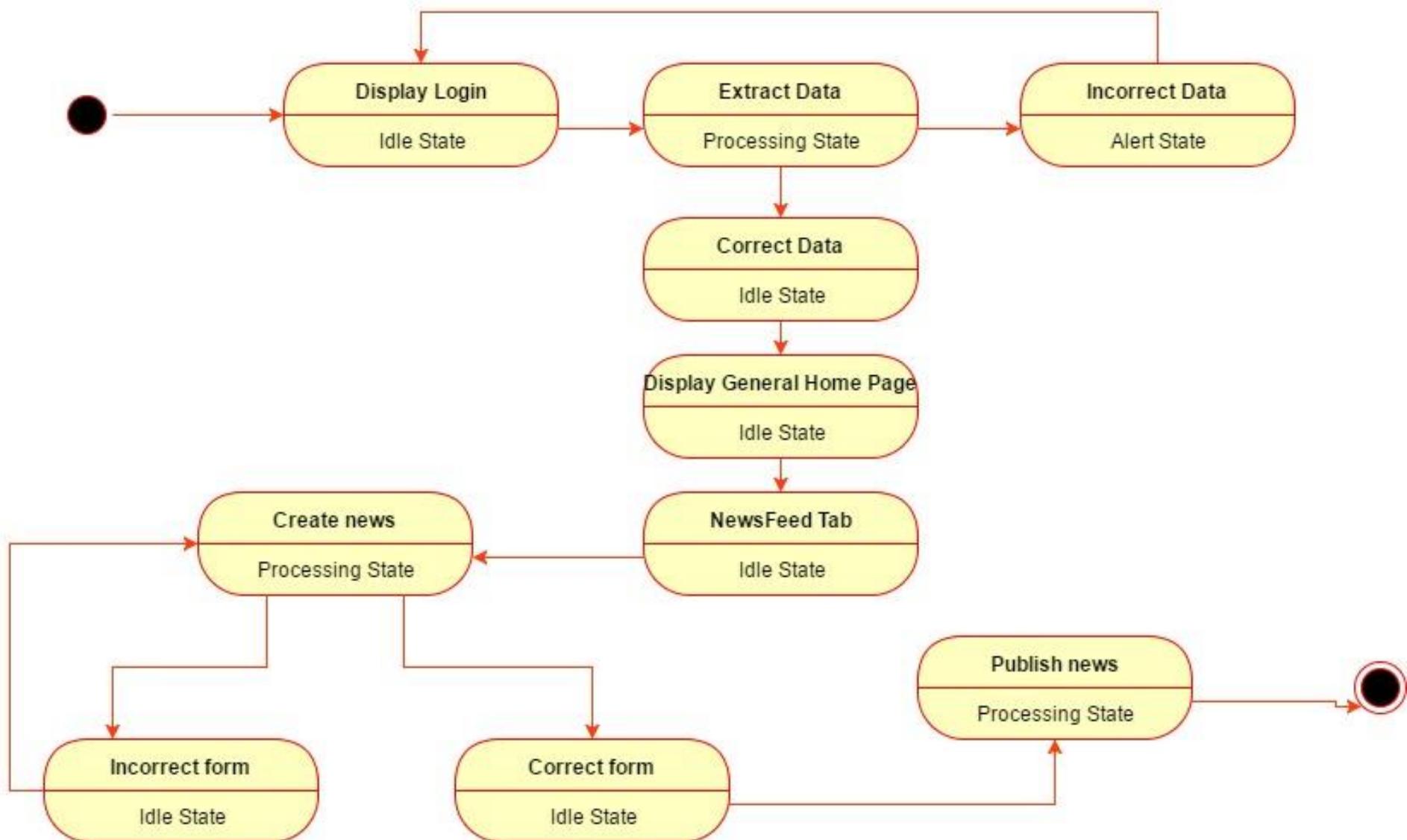


[TOPD] State Diagram

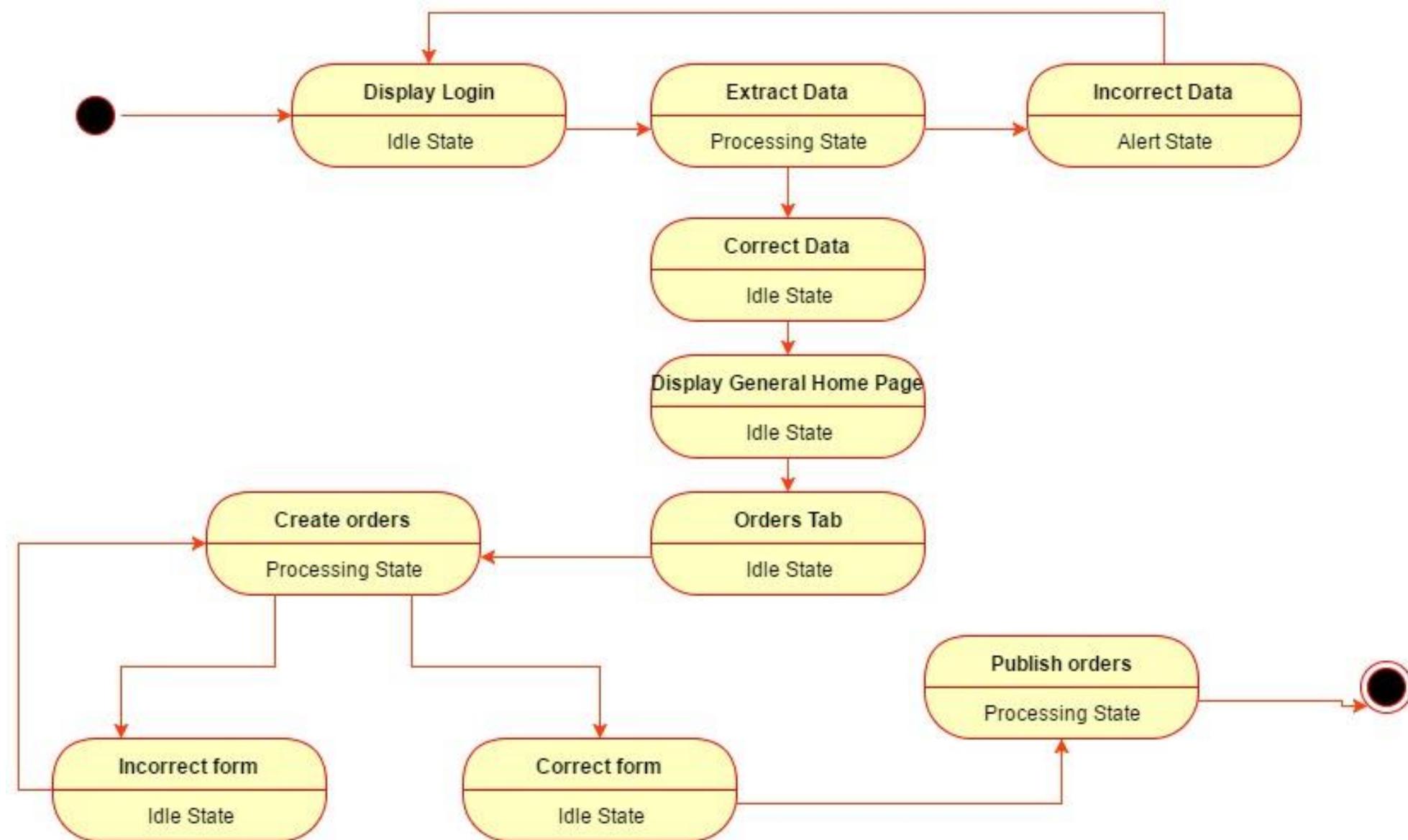




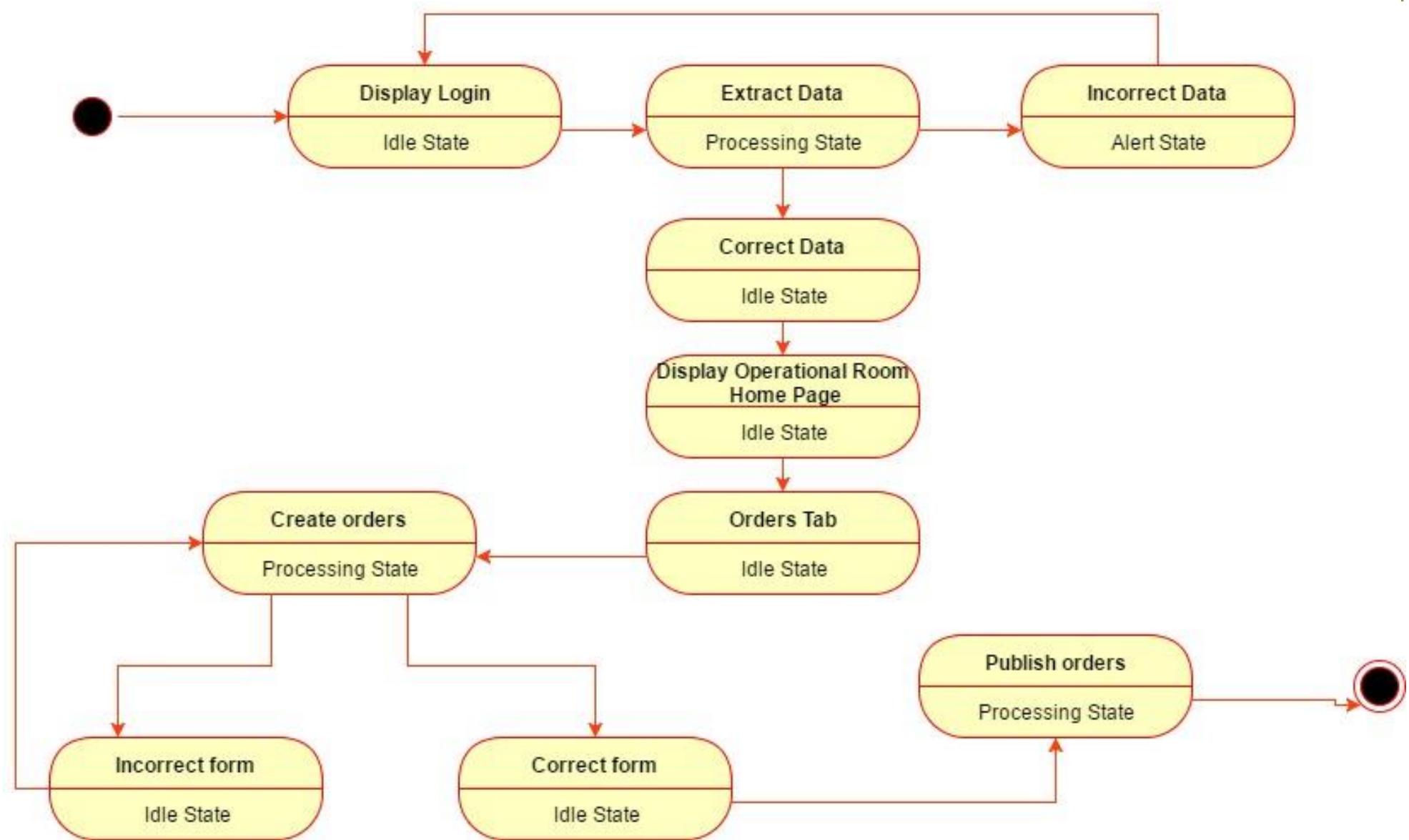
[TOPD] State Diagram



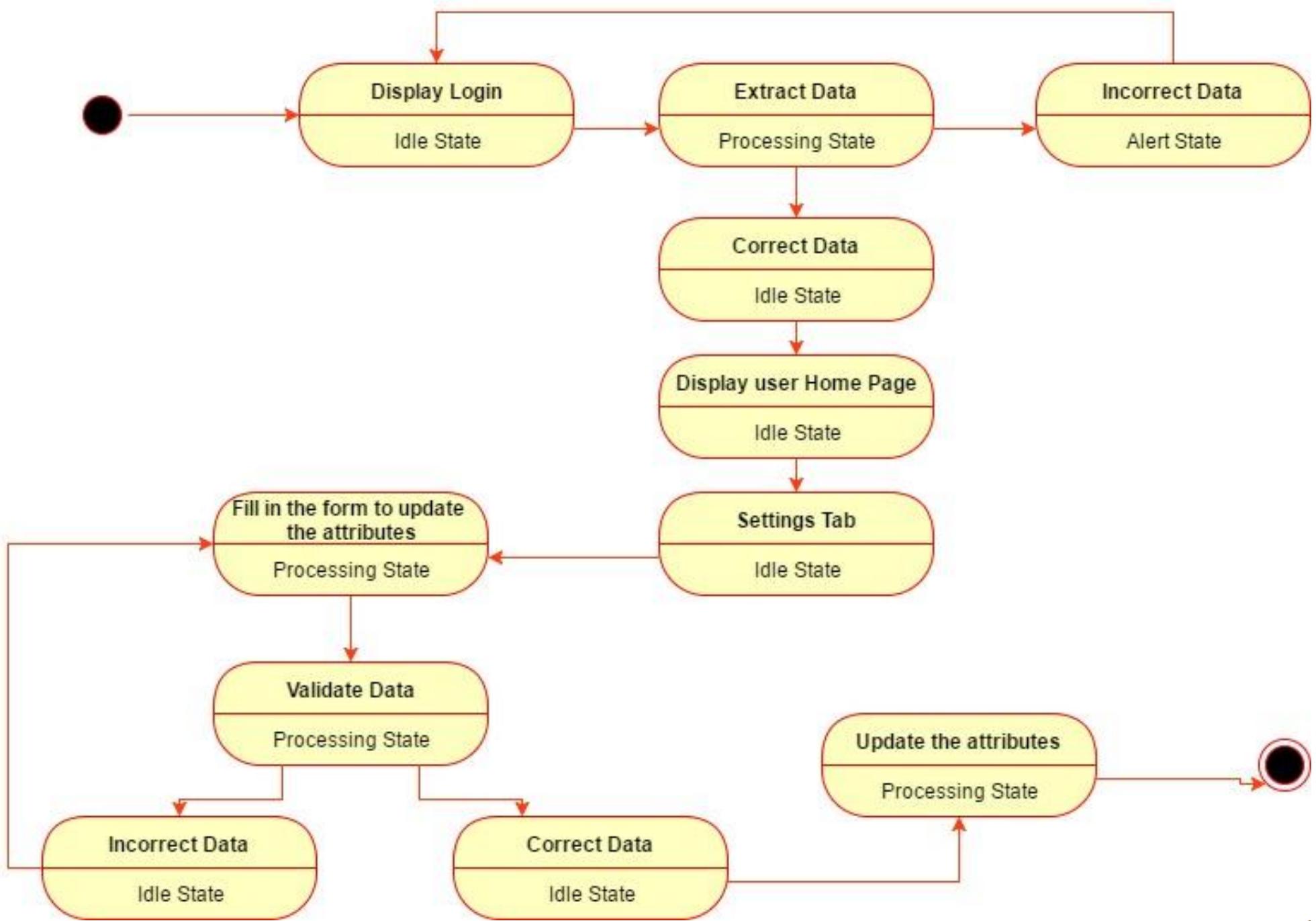
[TOPD] State Diagram



[TOPD] State Diagram

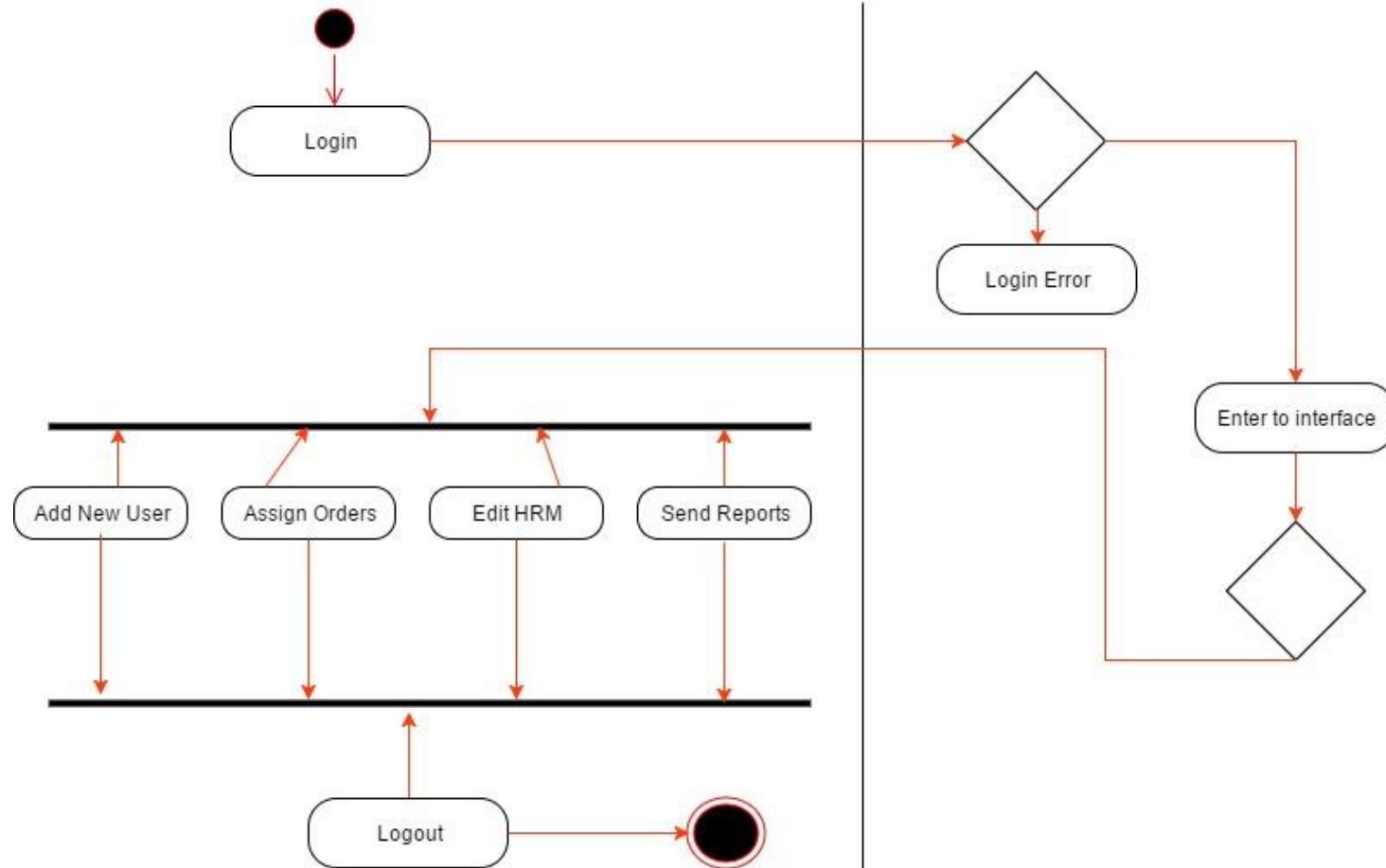


[TOPD] State Diagram



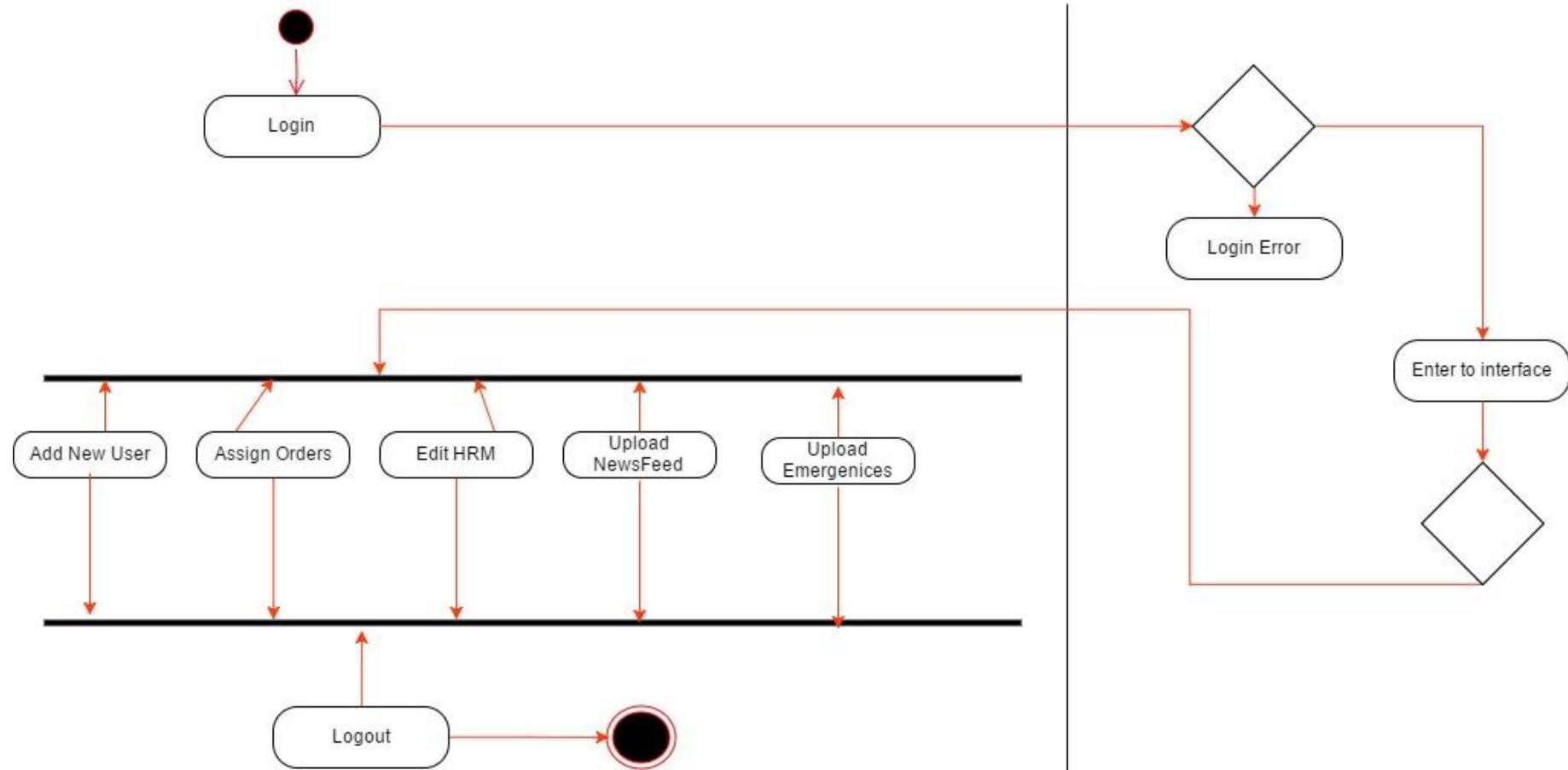
[TOPD] Swim Lane Diagram

District Operational Room Admin

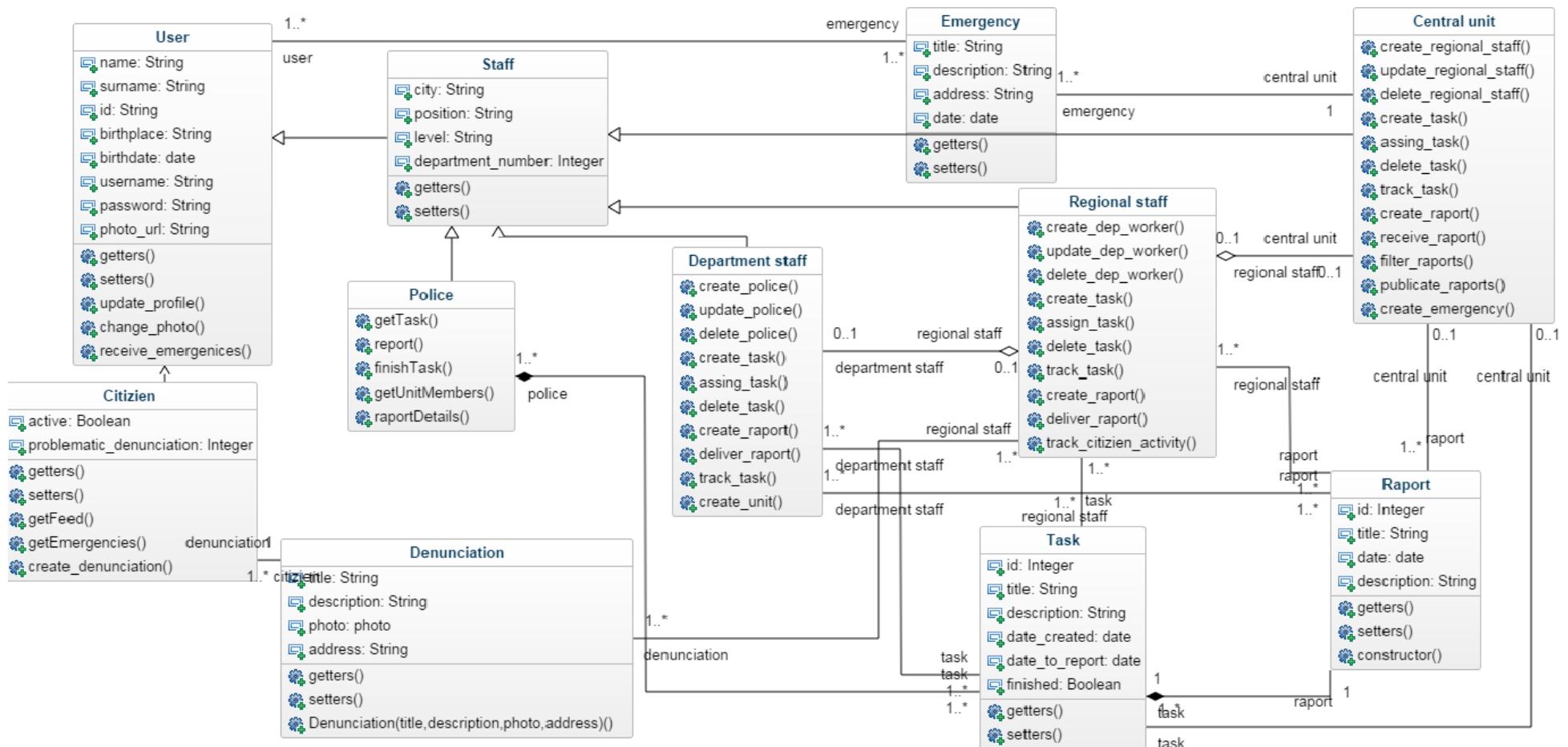


[TOPD] Swim Lane Diagram

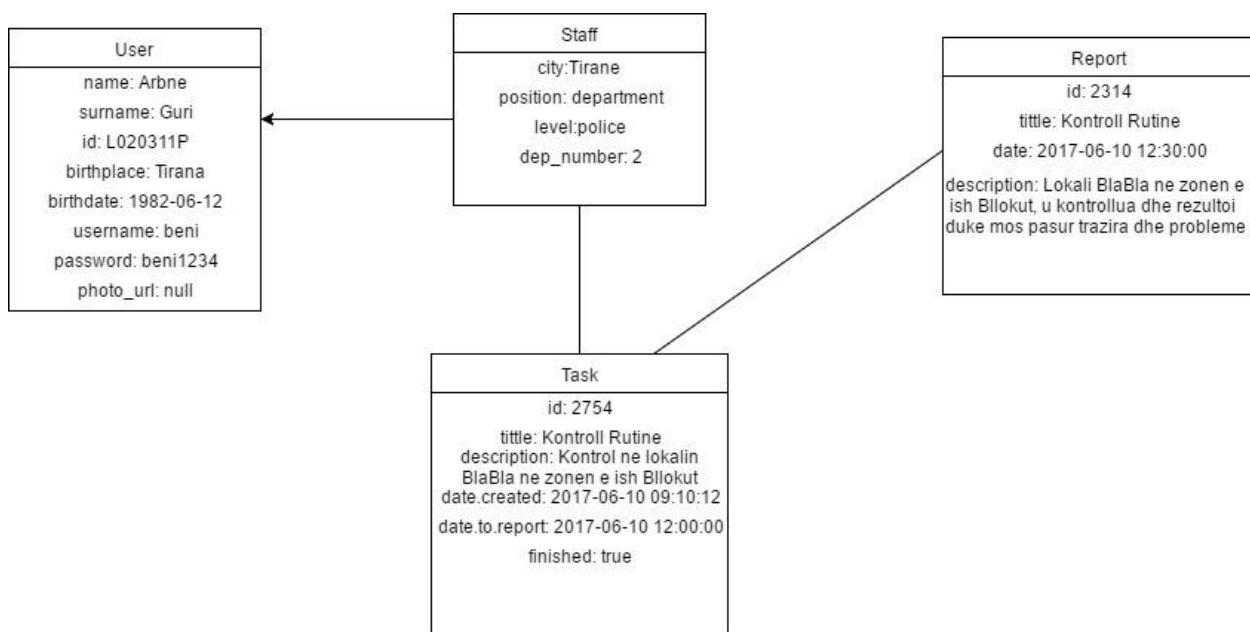
General Operational Room Admin



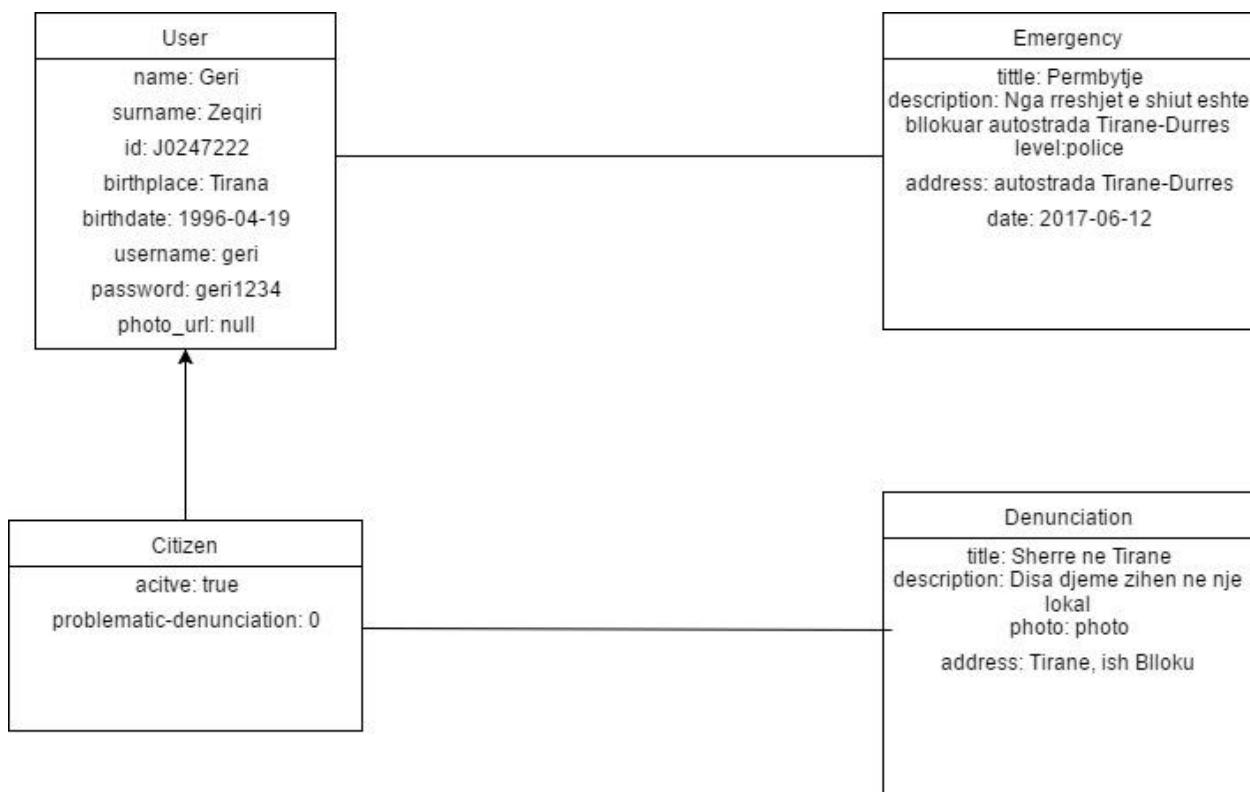
[TOPD] Class Diagram



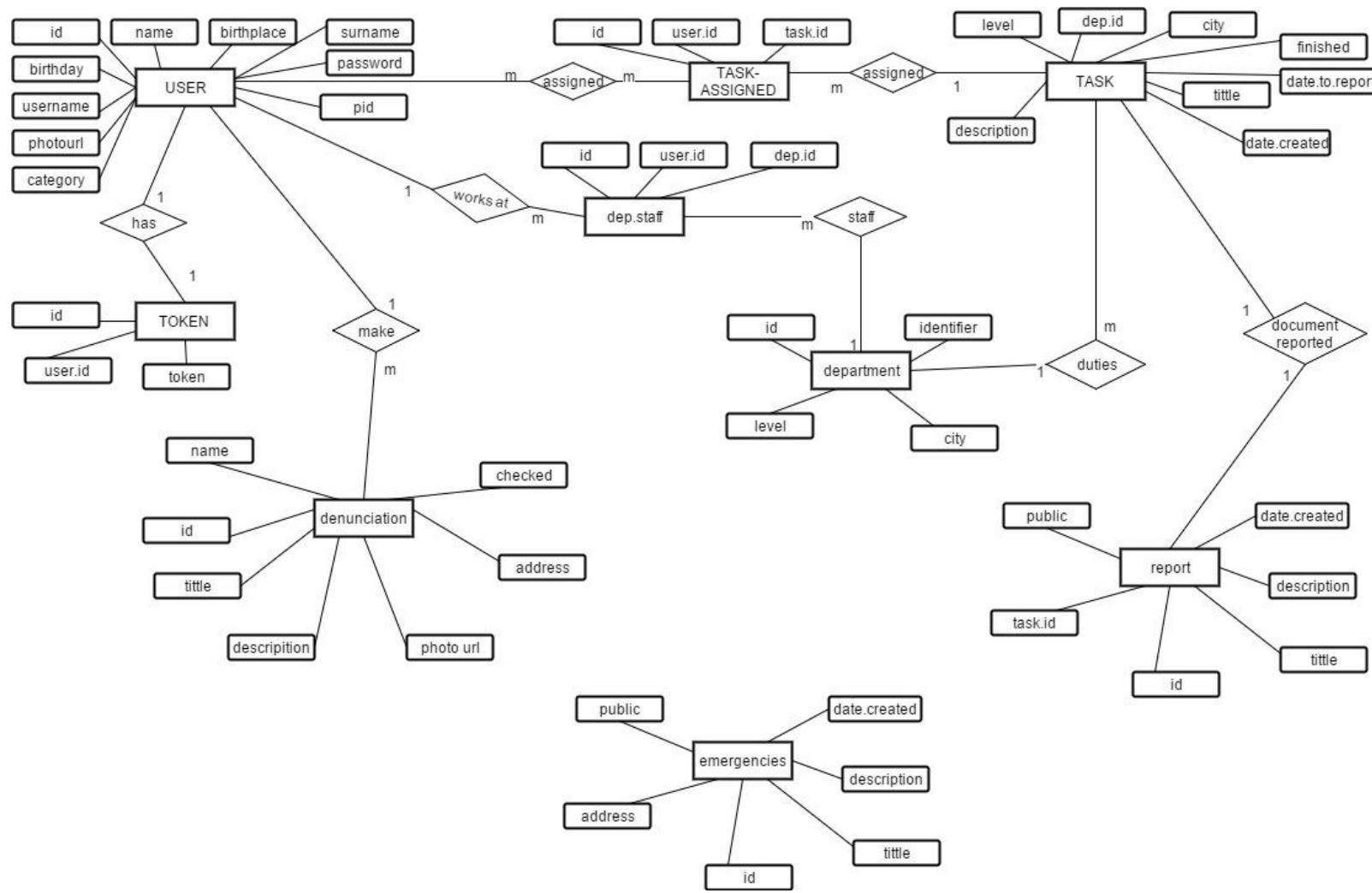
[TOPD] Object Diagram



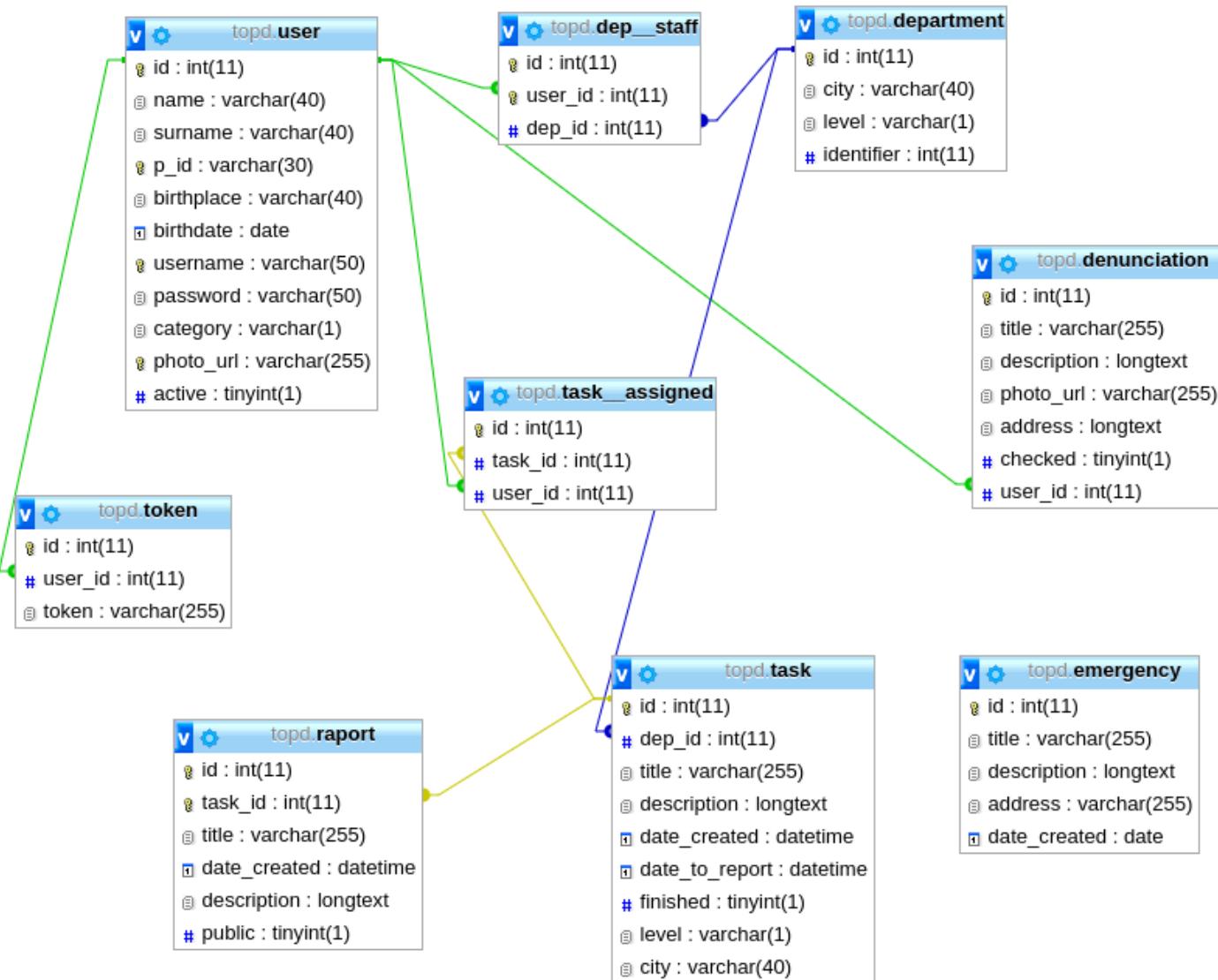
[TOPD] Object Diagram



[TOPD] ER Diagram



[TOPD] Database Schema



Implementation

In order to implement our software project, which includes a web project and also a companion android application, we decided to take in consideration the use of PHP framework Symfony and the use of android programming language.

Symfony is a MVC PHP framework, where MVC stands for model view controller framework. In the model part are included the classes which will interact with the database, using the Doctrine approach, that is integrated to the Symfony framework. By using Doctrine the data residing in the database are represented in the shape of objects, which is a form of mapping from database entities characteristics and parameters, to classes and their parameters and their associated methods in order to access, update and create instances of those classes. This mapping is very useful in the sense that it excludes the need to use traditional sql queries, and it provides an enhanced approach towards security and access of information, eliminating traditional threats such as sql injections and cross-site scripting css. By using the methods provided by each class, that are residing under the directory Entity of src directory of the project, it is relatively simply to perform action on the data, because the entire process is based on the object model approach and techniques. Also the Doctrine feature of Symfony provides useful methods, meaning the doctrine is included as an object in the construction of Symfony, by which it provides the way to create, update, delete and perform any other action that are part of MySql programming language, in the easiest way possible, using object and class methods. By using the annotations elements provided by Symfony and Doctrine, it is also possible to define complex connection and relationships between entities, without using sql codes, and retrieving those relationships by calling the class methods. In order to define the entities required for the success of our software implementation, we took advantage of the features offered by Symfony and Doctrine in order to create the databases and model entities, using solely the command line and specifying the characteristics of each entity, which we immediately displayed in the code, meaning it automatically generated the PHP classes in terms of entity, with the specified methods to access the parameters of the class. The entity directory which contains the entity, which are the classes representing the entities residing in the database, represents the core of our software projects, containing the most important information which will be used by the application and users, in order to accomplish their needs.

The controller part of the Symfony, is represented by using the Controller component of this framework, which is an objected integrated to this framework, providing necessary methods in order to offer the means of manipulating the information that resides in the database, and to provide a mapping of this information to the view part, which is rendered through the methods of the controller class. All that is required to access the required methods, is to extends the user defined classes with the Controller class, and the framework provides the necessary methods in order to perform from the most basic to the most complicated action regarding to the access of web pages, manipulating the information and rendering the necessary information to the web page, meaning providing the dynamic content of the web pages. In our project we took advantage of this technology and approach, in order to specify what is called as the class diagram

methods and actions, to the controller which is responsible for each category of user, and each action of each user level. By doing so we mean that the predefined actions and methods that are specified during the diagram creating development process are mapped and associated to each of the controller representing the user or entity which is responsible to perform actions that are specified in the requirements process, of the software specification. Each of the actions, which in the old plain php coding process would require a whole php scripting file, in Symfony takes just a method, with a specified Route, that represents the url of the page, and the necessary url parameters, in case there are required parameters. What is important to note is the ability that we used in order to create an API, meaning a interface for mobile to communicate with the server using the technology of JSON format, by using only controller methods and routes, and also HTTP object to specify the method to be used, which is POST.

Concerning to the view part of the framework, it is important to note that the frontend part of the project uses the technologies used by every Web application which are HTML, JAVASCRIPT and CSS, and their elements. The special feature that Symfony uses, is the format of TWIG, which is a HTML format page, but that does not allow php tags and scripting in its content. We also took advantage of open forums and communities that provide html, css and javascript templates such as BootStrap.

For the mobile application, we provide an android application implementation which is based on Java and XML. Java provides the necessary methods and classes in order to implement the core code and backend of the android application. Xml is the format that is used by the android in order to render the templates and layout of the application. As mention earlier in this section, we provided an API, that the mobile uses in order to communicate with the server. In order to make use of this API, we use the Retrofit technology, in order to produce Plain Old Java Object models in order to provide the necessary mapping from JSON to JAVA object and from Java Object to JSON format, in order to establish the means of communication between server and mobile client.

Below we provide a touch to the API implementation:

```
<?php
/**
 * Created by PhpStorm.
 * User: geri
 * Date: 17-05-30
 * Time: 9.55.MD
 */
```

```
namespace AppBundle\Controller\App;
```

```
use AppBundle\Entity\Denunciation;
use AppBundle\Entity\Token;
```

```

use AppBundle\Entity\User;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Method;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route;
use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Symfony\Component\HttpFoundation\JsonResponse;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\Validator\Constraints\Date;

/**
 * Class AppController
 *
 * @package AppBundle\Controller\App
 * @Route("/app")
 */
class AppController extends Controller
{
    /**
     * @Route("/login", name="app_login")
     * @Method("POST")
     * @param Request $request
     * @return JsonResponse
     */
    public function login(Request $request){

        //validate data

        $data = $request->getContent();
        if($request->getContentType() != 'json'){
            return new JsonResponse(array('status' => 403, 'message' => "Content should be in json format", 'token'=>null, 'category'=>null));
        }

        $json = json_decode($data, true);

        if(json_last_error() !== JSON_ERROR_NONE){
            return new JsonResponse(array('status' => 401, 'message' => 'Post method is required to send data in json format', 'token' =>null, 'category'=>null));
        }

        $username = '';
        $password = '';

        if(array_key_exists('username', $json) && !empty($json['username']) &&
array_key_exists('password', $json) && !empty($json['password'])){
            $username = $json['username'];
        }
    }
}

```

```

        $password = $json['password'];
    }
    else{
        return new JsonResponse(array('status'=>402, 'message'=>'Please provide a valid
username and password', 'token'=>null, 'category'=>null));
    }

//retrieve data

$em = $this->getDoctrine()->getRepository("AppBundle:User");

$user = $em->findOneBy(array('username' => $username, 'password' => $password));

if(!$user){
    $array = array('status' => 400, 'message' => "No user with these credentials", 'token'
=> null, 'category'=>null);
    return new JsonResponse($array);
}

if($user->getCategory() != 'u' && $user->getCategory() != 'p'){
    return new JsonResponse(array('status'=>404, 'message'=>"This account cannot
receive a token", 'token'=>null, 'category'=>null));
}

//check password and token

$tokens = $user->getTokens();
$category = $user->getCategory();

$size = sizeof($tokens);
if($size != 0){
    $array = array('status' =>300, 'message' => "You have already been assigned a token",
'token' => null, 'category'=>null, 'category'=>null);
    return new JsonResponse($array);
}

//create token

$token = new Token();
$time = date('Y/m/d');
$token->setToken(sha1($user->getUsername().$time));
$token->setUser($user);

$em1 = $this->getDoctrine()->getManager();

```

```

$em1->persist($token);
$em1->flush();

if(is_numeric($token->getId())){
    $array = array('status'=>200, 'message'=>"Ok", 'token'=>$token->getToken(),
    'category'=>$category);
    return new JsonResponse($array);
}
else{
    return new JsonResponse(array('status'=>500, 'message'=>"Problem connecting to
server", 'token'=>null, 'category'=>null));
}

}

/***
 * @param Request $request
 * @Route("/emergency", name="app_emergency")
 * @Method("POST")
 * @return JsonResponse
 */
public function getEmergencies(Request $request){

    if($request->getContentType() != 'json'){
        return new JsonResponse(array('status'=>403, 'message'=>"Content should be in json
format", 'emergency'=>null));
    }

    $data = json_decode($request->getContent(), true);
    if(json_last_error() !== JSON_ERROR_NONE){
        return new JsonResponse(array('status' => 401, 'message'=>'Post method is required to
send data in json format', 'emergency' =>null));
    }

    $token = "";
    if(array_key_exists('token', $data)){
        $token = $data['token'];
        if(empty($token) || is_null($token)){
            return new JsonResponse(array('status'=>402, 'message'=>"please provide a valid
token format", 'emergency'=>null));
        }
    }
    else{
        return new JsonResponse(array('status'=>402, 'message'=>"please provide a token in
the json request", 'emergency'=>null));
    }
}

```

```

    }

    //check token

    $token = $this->getDoctrine()->getRepository("AppBundle:Token")-
>findOneByToken($token);
    if(!$token){
        return new JsonResponse(array('status'=>300, 'message'=>"invalid token",
'emergency'=>null));
    }

    //retrieve data

    $emergencies = $this->getDoctrine()->getRepository("AppBundle:Emergency")-
>findAll();

    $em = array();
    foreach($emergencies as $emergency){
        $em[] = array(
            'title'=>$emergency->getTitle(),
            'description'=>$emergency->getDescription(),
            'address'=>$emergency->getAddress(),
            'date'=>$emergency->getDateCreated());
    }

    //return response

    return new JsonResponse(array('status'=>200, "message"=>"ok", 'emergency'=>$em));
}

/**
 * @param Request $request
 * @Route("/logout", name="logout")
 * @Method("POST")
 * @return JsonResponse
 */
public function logout(Request $request){

    if($request->getContentType() != 'json'){
        return new JsonResponse(array('status'=>403, 'message'=>"Content should be in json
format"));
    }

    $data = json_decode($request->getContent(), true);
    if(json_last_error() !== JSON_ERROR_NONE){

```

```

        return new JsonResponse(array('status' => 401, 'message'=>'Post method is required to
send data in json format'));

    }

    $token = '';
    if(array_key_exists('token', $data)){
        $token = $data['token'];
        if(empty($token) || is_null($token)){
            return new JsonResponse(array('status'=>402, 'message'=>"please provide a valid
token format"));
        }
    }
    else{
        return new JsonResponse(array('status'=>402, 'message'=>"please provide a token in
the json request"));
    }

    //check token

    $token = $this->getDoctrine()->getRepository("AppBundle:Token")-
>findOneByToken($token);
    if(!$token){
        return new JsonResponse(array('status'=>300,'message'=>"invalid token"));
    }

    $em = $this->getDoctrine()->getManager();
    $em->remove($token);
    $em->flush();
    return new JsonResponse(array('status'=>200, 'message'=>'ok'));
}

/***
 * @param Request $request
 * @Route("/register", name="app_register")
 * @Method("POST")
 * @return JsonResponse
 */
public function register(Request $request){
    if($request->getContentType() != 'json'){
        return new JsonResponse(array('status'=>403, 'message'=>"Content should be in json
format", 'token'=>null));
    }
    $data = json_decode($request->getContent(), true);
    if(json_last_error() !== JSON_ERROR_NONE){
        return new JsonResponse(array('status'=>401, 'message'=>'Post method is required to
send data in json format','token'=>null));
    }
}

```

```

}

if(array_key_exists('name', $data) && !empty($data['name'])){
    $name = $data['name'];
}
else{
    return new JsonResponse(array('status'=>402, 'message'=>'please provide a name in
the json request','token'=>null));
}
if(array_key_exists('surname', $data) && !empty($data['surname'])){
    $surname = $data['surname'];
}
else{
    return new JsonResponse(array('status'=>402, 'message'=>'please provide a surname in
the json request','token'=>null));
}
if(array_key_exists('pid', $data) && !empty($data['pid'])){
    $pid = $data['pid'];
}
else{
    return new JsonResponse(array('status'=>402, 'message'=>'please provide a personal id
in the json request','token'=>null));
}
if(array_key_exists('birthplace', $data) && !empty($data['birthplace'])){
    $birthplace = $data['birthplace'];
}
else{
    return new JsonResponse(array('status'=>402, 'message'=>'please provide a birthplace
in the json request','token'=>null));
}
if(array_key_exists('birthdate', $data) && !empty($data['birthdate'])){
    $birthdate = $data['birthdate'];
}
else{
    return new JsonResponse(array('status'=>402, 'message'=>'please provide a birthdate
in the json request','token'=>null));
}
if(array_key_exists('username', $data) && !empty($data['username'])){
    $username = $data['username'];
}
else{
    return new JsonResponse(array('status'=>402, 'message'=>'please provide a username
in the json request','token'=>null));
}
if(array_key_exists('password', $data) && !empty($data['password'])){
    $password = $data['password'];
}

```

```

        }
    else{
        return new JsonResponse(array('status'=>402, 'message'=>'please provide a password
in the json request', 'token'=>null));
    }

//check username

$user = $this->getDoctrine()->getRepository("AppBundle:User")-
>findOneBy(array('username'=>$username));
if($user){
    return new JsonResponse(array('status'=>'333', 'message'=>"Username is taken",
'token'=>null));
}

$us = new User();
$us->setName($name);
$us->setSurname($surname);
$us->setPId($pid);
$us->setBirthplace($birthplace);
$us->setBirthdate(new \DateTime($birthdate));
$us->setUsername($username);
$us->setPassword($password);
$us->setCategory('u');
$us->setActive(true);

$token = new Token();
$token->setUser($us);

$tk = $us->getUsername().".".$us->getPId();
$tk = sha1($tk);
$token->setToken($tk);

$em = $this->getDoctrine()->getManager();
$em->persist($us);
$em->persist($token);
$em->flush();
return new JsonResponse(array('status'=>200, 'message'=>'ok', 'token'=>$token-
>getToken(), 'category'=>'u'));

}

```

```

    /**
     * @param Request $request
     * @Route("/feed", name="app_feed")
     * @Method("POST")
     * @return JsonResponse
     */
    public function getFeed(Request $request){

        if($request->getContentType() != 'json'){
            return new JsonResponse(array('status'=>403, 'message'=>"Content should be in json
format", 'feed'=>null));
        }

        $data = json_decode($request->getContent(), true);
        if(json_last_error() !== JSON_ERROR_NONE){
            return new JsonResponse(array('status' => 401, 'message'=>'Post method is required to
send data in json format', 'feed' =>null));
        }

        $token = "";
        if(array_key_exists('token', $data)){
            $token = $data['token'];
            if(empty($token) || is_null($token)){
                return new JsonResponse(array('status'=>402, 'message'=>"please provide a valid
token format", 'feed'=>null));
            }
        }
        else{
            return new JsonResponse(array('status'=>402, 'message'=>"please provide a token in
the json request", 'feed'=>null));
        }

        //check token

        $token = $this->getDoctrine()->getRepository("AppBundle:Token")-
>findOneByToken($token);
        if(!$token){
            return new JsonResponse(array('status'=>300,'message'=>"invalid token",
'feed'=>null));
        }

        $feed = $this->getDoctrine()->getRepository("AppBundle:Raport")->findBy(array('public'
=>true));
        $feed_res = array();

        foreach ($feed as $f){

```

```

$feed_res[] = array(
    'title'=>$f->getTitle(),
    'description'=>$f->getDescription(),
    'date_created'=>$f->getDateCreated(),
);
}

return new JsonResponse(array('status'=>200, 'message'=>'ok', 'feed'=>$feed_res));
}

/**
 * @param Request $request
 * @Route("/update_profile", name="app_update_profile")
 * @Method("POST")
 * @return JsonResponse
 */
public function update_profile(Request $request){

if($request->getContentType() != 'json'){
    return new JsonResponse(array('status'=>403, 'message'=>"Content should be in json
format"));
}

$data = json_decode($request->getContent(), true);
if(json_last_error() !== JSON_ERROR_NONE){
    return new JsonResponse(array('status' => 401, 'message'=>'Post method is required to
send data in json format'));
}

}
$token = "";
if(array_key_exists('token', $data)){
    $token = $data['token'];
    if(empty($token) || is_null($token)){
        return new JsonResponse(array('status'=>402, 'message'=>"please provide a valid
token format"));
    }
}
else{
    return new JsonResponse(array('status'=>402, 'message'=>"please provide a token in
the json request"));
}

//check token

```

```

$token = $this->getDoctrine()->getRepository("AppBundle:Token")-
>findOneByToken($token);
if(!$token){
    return new JsonResponse(array('status'=>300, 'message'=>"invalid token"));
}

$user = $token->getUser();
if(array_key_exists('username', $data) && !empty($data['username'])){
    $username = $data['username'];
    $us = $this->getDoctrine()->getRepository("AppBundle:User")-
>findOneBy(array('username'=>$username));
    if(!$us){
        $user->setUsername($username);
    }
    else{
        return new JsonResponse(array('status'=>302, 'message'=>"username is taken"));
    }
}

if(array_key_exists('password', $data) && !empty($data['password'])){
    $password = $data['password'];
    $user->setPassword($password);
}

$em = $this->getDoctrine()->getManager();
$em->flush();

return new JsonResponse(array('status'=>200, 'message'=>"ok"));
}

/**
 * @param Request $request
 * @Route("/denunciation", name="app_denunciation")
 * @Method("POST")
 * @return JsonResponse
 */
public function denunciation(Request $request){

if($request->getContentType() != 'json'){
    return new JsonResponse(array('status'=>403, 'message'=>"Content should be in json
format"));
}

$data = json_decode($request->getContent(), true);
if(json_last_error() !== JSON_ERROR_NONE){

```

```

        return new JsonResponse(array('status' => 401, 'message'=>'Post method is required to
send data in json format'));

    }
    $token = "";
    if(array_key_exists('token', $data)){
        $token = $data['token'];
        if(empty($token) || is_null($token)){
            return new JsonResponse(array('status'=>402, 'message'=>"please provide a valid
token format"));
        }
    }
    else{
        return new JsonResponse(array('status'=>402, 'message'=>"please provide a token in
the json request"));
    }

    //check token

    $token = $this->getDoctrine()->getRepository("AppBundle:Token")-
>findOneByToken($token);
    if(!$token){
        return new JsonResponse(array('status'=>300,'message'=>"invalid token"));
    }

    if(array_key_exists('title', $data) && !empty($data['title'])){
        $title = $data['title'];
    }
    else{
        return new JsonResponse(array('status'=>402, 'message'=>"please provide a title in the
json request"));
    }
    if(array_key_exists('description', $data) && !empty($data['description'])){
        $desription = $data['description'];
    }
    else{
        return new JsonResponse(array('status'=>402, 'message'=>"please provide a
description in the json request"));
    }
    if(array_key_exists('address', $data['address']) && !empty($data['address'])){
        $address = $data['address'];
    }
    else{
        return new JsonResponse(array('status'=>402, 'message'=>"please provide an address
in the json request"));
    }
}

```

```

if(array_key_exists('photo', $data) && !empty($data['photo'])){
    $photo = $data['photo'];
}
else{
    return new JsonResponse(array('status'=>402, 'message'=>"please provide a base64
image in the json request"));
}

$user = $token->getUser();

$den = new Denunciation();
$den->setTitle($title);
$den->setUser($user);
$den->getAddress($address);
$den->setDescription($desription);
$den->setChecked(false);

$photo1 = explode(", ", $photo);
$den->setPhotoUrl($photo1[1]);

$em = $this->getDoctrine()->getManager();
$em->persist($den);
$em->flush();

return new JsonResponse(array('status'=>200, 'message'=>'ok'));
}

<?php
/**
 * Created by PhpStorm.
 * User: geri
 * Date: 17-06-03
 * Time: 10.44.MD
 */
namespace AppBundle\Controller\App;

```

```

use Sensio\Bundle\FrameworkExtraBundle\Configuration\Method;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route;
use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Symfony\Component\HttpFoundation\JsonResponse;
use Symfony\Component\HttpFoundation\Request;

/**
 * Class PoliceController
 * @package AppBundle\Controller\App
 * @Route("/app", name="app")
 */
class PoliceController extends Controller
{

    /**
     * @Route("/task", name="app_task")
     * @param Request $request
     * @Method("POST")
     * @return JsonResponse
     */
    public function getTask(Request $request){
        if($request->getContentType() != 'json'){
            return new JsonResponse(array('status'=>403, 'message'=>"Content should be in json format", 'task'=>null));
        }

        $data = json_decode($request->getContent(), true);
        if(json_last_error() !== JSON_ERROR_NONE){
            return new JsonResponse(array('status' => 401, 'message'=>'Post method is required to send data in json format', 'task' =>null));
        }

        $token = "";
        if(array_key_exists('token', $data)){
            $token = $data['token'];
            if(empty($token) || is_null($token)){
                return new JsonResponse(array('status'=>402, 'message'=>"please provide a valid token format", 'task'=>null));
            }
        }
        else{
            return new JsonResponse(array('status'=>402, 'message'=>"please provide a token in the json request", 'task'=>null));
        }

        //check token
    }
}

```

```

$token = $this->getDoctrine()->getRepository("AppBundle:Token")-
>findOneByToken($token);
if(!$token){
    return new JsonResponse(array('status'=>300, 'message'=>"invalid token",
'task'=>null));
}
$user = $token->getUser();
if($user->getCategory() != 'p'){
    return new JsonResponse(array('status'=>302, 'message'=>"permission denied", 'task'-
=>null));
}

$tasks = $user->getTaskAssigned();

$array = array();
foreach($tasks as $task){
    $task = $task->getTask();
    $array[] = array(
        'id'=>$task->getId(),
        'title'=>$task->getTitle(),
        'description'=>$task->getDescription(),
        'date_created'=>$task->getDateCreated(),
        'date_to_report'=>$task->getDateToReport(),
        'finished'=>$task->getFinished(),
    );
}

return new JsonResponse(array('status'=>200, 'message'=>"ok", 'task'=>$array));
}

}

```

[TOPD] Project Management

20-Mar-17	4-Apr-17	11-Apr-17	18-Apr-17	25-Apr-17	2-May	9-May-17	16-May-17	23-May-17	30-May-17	10-Jun-17
Discussing different ideas										
	Assignin the tasks									
		Writing abstract and description								
			Writing software requirements							
				Writing user stories						
					Writing Use Cases					
						Writing the code				
							Writing Scenerios			
								Use Case&Activity Diagram		
									State & Sequence Diagram Class&Object&Component	
										Class&Deploy Diagram
										Data Flow Diagram
										ER DIAGRAM Database Schema
										Software Testing Project Manager Installation Manu Final Review