DU ALERT

A safety & awareness social platform for DU

SE-3105: Software Project Lab 2

Submitted by:

Md. Israfil Hossain
Roll: BSSE 1508

Sabbir Ahmed
Roll: BSSE 1530

Supervised by:

Dr. Emon Kumar Dey
Associate Professor
Institute of Information Technology
University of Dhaka

Table of Contents

1. Inception

1.1 Identify All Stakeholders:

Stakeholder	Role
Student	Registers using university details, reports incidents, submits public alerts (anonymous or non-anonymous) and receives safety notifications.
Proctor & Proctorial Team Member	Responds to emergency SOS alerts, views and investigates student complaints, updates complaint status (Received, In Progress, Resolved), monitors suspicious activities, and uses AI-generated summaries to handle cases efficiently.
Push Notification Service	Sends real-time notifications to students, proctors, admins, and security teams for emergency SOS alerts, public alerts, complaint updates, and important safety announcements.
Location Service (GPS/Geolocation API)	Collects and provides real-time location data of students during Emergency SOS situations to ensure fast and accurate response.
OTP Verification Service	Generates and verifies One-Time Passwords during registration, login recovery, and identity verification to ensure only genuine Dhaka University students can access the system.
Email Generator	Automatically generates and sends context-aware emails to user or authorities.
Summary Generator AI	Generate summary from the complaint for the proctor

1.2 Understanding the Existing System

This section summarizes how the current manual or semi-manual system works for student safety and communication at Dhaka University and highlights pain points or areas for improvement.

1.2.1 Existing System

The existing safety and reporting system at Dhaka University relies primarily on manual or semi-manual processes. The primary components of the current system are as follows:

1. Emergency Response

- Students in danger contact the university proctor or security personnel by phone or in person.
- The location of the student is communicated verbally or via text.
- The proctor or security team manually decides which unit will respond.
- Response times depend heavily on human availability and communication efficiency.

2. Incident Reporting

- Students report incidents by submitting paper forms or emails to the university authority
- Reports include basic information: student name, department, registration number, incident description.
- Multiple complainants must submit separate forms.
- Media evidence, if any (photos or videos), must be physically submitted or emailed.

3. Complaint Tracking

- Complaint status is tracked manually using notebooks, registers or spreadsheets.
- Students must call or visit proctors to check updates on their complaints.
- Proctors manually update the status of each complaint: Received, In Progress or Resolved.

4. Public Alerts

- Students warn peers about suspicious activities or risks via word-of-mouth, notice boards, or informal social media groups.
- No structured review or verification process exists.
- Anonymous reporting is limited or non-existent.
- False information may spread due to lack of moderation.

5. Notification System

- Notifications about incidents or alerts are communicated manually via announcements, SMS or emails.
- Students, proctors, and admin may not receive real-time updates.
- Many updates are delayed or missed due to manual handling.

6. Proctorial Monitoring

- Proctors maintain separate logs for emergencies, complaints and suspicious activities.
- Data consolidation requires manual collection from registers, phone calls and emails.
- AI-based analysis or summary generation is absent.
- Proctors rely solely on memory or notes for quick decision-making.

7. Administrative Oversight

- Admins manually verify student identities during registration or for issuing access.
- Approval of public warnings and management of proctor accounts is paper-based.
- Analytics on complaint trends, emergency response times and campus safety are prepared manually or using spreadsheets.

1.2.2 Observations & Problems in the Existing System

- Manual tracking of emergencies and complaints is **time-consuming and error-prone**.
- Response to SOS or critical incidents is **slow due to lack of real-time location tracking**.
- No unified digital platform for submitting complaints, public alerts or anonymous posts.
- **Status updates for complaints** are not visible to students in real-time.
- **Verification and authentication** of students is cumbersome and prone to misuse.
- Notifications are **delayed, inconsistent or incomplete**.
- **Data redundancy** exists due to separate paper logs, emails and registers.
- No analytics or automated reporting to **help identify campus safety trends**.
- Lack of **role-based access control**—any manual error can compromise security or confidentiality.
- Public warnings often **lack moderation**, leading to misinformation or false alarms.

2. Elicitation

2.1 Quality Function Deployment (QFD)

This maps user needs to system functions, helping prioritize features in the upcoming digital system for DU Alert.

2.1.1 Normal Requirements

- **Incident Reporting:** Students can report incidents with category, description, and optional media.
- **Complaint Tracking**: Students can view complaint status: Received, In Progress, or Re-solved.
- **Public Alert Posting:** Students can create public alerts after admin approval.

- **Admin Approval System:** Admin reviews and approves or rejects public posts before publishing.
- **Proctor Dashboard:** Proctors can view and manage complaints, SOS alerts, and communicate with students.
- **Admin Dashboard:** Admin manages users, posts, and tracks statistics and complaints.
- **Timeline Feed:** Displays approved alerts, updates, and resolved cases with reactions/comments.
- **Notification System:** Students get notified about complaint updates and admin responses.
- **Media Storage:** Images and videos are securely stored in cloud platforms (e.g., Firebase,Cloudinary).
- **Monthly Statistics & Analytics:** Generates monthly reports showing complaint and safety trends.
- **Logout Functionality:** Secure logout option for all users to end sessions safely.

2.1.2 Expected Requirements

- The interface must be intuitive, simple and easy for students, proctors and admins to navigate.
- The system should be mobile-friendly and responsive across devices (mobile, tablet, desktop).
- Login credentials must be securely stored using encrypted password methods.
- Sensitive user data (location, contact info, complaint content) must be protected.
- Students and proctors should be able to search complaints or alerts by keywords such as category, name, date or location.
- Students should be able to track complaint progress in real-time via dashboard.
- Notifications should be delivered immediately for emergencies, public alerts and updates.
- The system must support anonymous reporting while still ensuring admin moderation.
- Role-based access control must ensure only authorized users can perform specific tasks (student, proctor, admin).

2.1.3 Exciting Requirements

- **AI Summary Generator for Proctors:** Proctors can view a quick AI-generated summary of reported incidents.
- **Smart Alert Prioritization:** The system automatically flags and prioritizes urgent SOS or high-risk reports.
- **Auto Location Sharing:** When the SOS button is pressed, the system automatically captures and sends the student's latitude and longitude.

3. User Story (DU Alert)

DU Alert is a modern, mobile-friendly safety and communication platform created to protect the students of Dhaka University and improve emergency response across the entire campus. The application focuses on three core goals: helping students during emergencies, allowing them to report incidents easily, and providing a secure way to share important safety updates with the university community. The story of DU Alert begins from the moment a student installs the app, and continues through registration, daily use, emergency handling, reporting, and notifications.

The first experience a user has with the system is the registration process. When a student opens the app for the first time, they are shown a clean and user-friendly registration form. The student enters their full name, department, registration number, university email address, and phone number. To make registration easier and more accurate, the department list includes auto-suggestions. As soon as the student starts typing, the system suggests the name of the department from the full list of Dhaka University departments, helping the student avoid spelling mistakes and speeding up the process.

After filling in the form, the student clicks the "Register" button. Immediately, the system sends an OTP (One-Time Password) to the student's registered phone number. This OTP is used to verify that the student truly belongs to Dhaka University. The OTP must be entered within a certain time limit. If the student enters the correct code, the system confirms their identity and allows them to move to the next step: creating their own unique username and password. This username and password become the student's permanent login credentials. The app stores these securely using encrypted password storage methods, ensuring that no unauthorized person can view or misuse the login information.

Once registration is complete, the student can log in anytime using their username and password. If a student forgets their password, the app includes a simple recovery system. They just enter their registration number, and the system sends another OTP to their registered phone or email. After verifying the OTP, they can set a new password and regain access to their account. This allows only genuine and verified students to use DU Alert while preventing outsiders from accessing the platform.

After successful login, the student enters the DU Alert dashboard. This dashboard is designed to be clear, simple, and action-focused. The most important feature is the "Emergency SOS" or "Need Help" button, placed prominently at the top. This button is specially designed for life-threatening or serious situations. If a student faces danger—such as harassment, assault, or any medical emergency they can press this single button. When the button is pressed, the system automatically collects the student's real-time location using the phone's GPS or browser's geolocation service. This location, along with the student's name, registration number, and phone

number, is immediately sent to the university's proctorial team. The system identifies which proctor or security unit is responsible for the student's area and sends the emergency alert directly to them. This ensures extremely fast response and allows the authorized team to reach the student as quickly as possible.

In addition to the emergency button, the DU Alert dashboard also includes a Report an Incident section. This section is used for non-emergency situations that still require attention and investigation. When a student wants to report an incident, they see a well-organized complaint form. The first field is the category selector. The user can choose from categories such as harassment, theft, property loss, suspicious activity, fraud, cyber-related issues, or others. This helps in classifying the incident and ensures that the proper authorities can respond. After selecting the category, the student enters the complainant details. This feature allows a single complaint to include multiple complainants. For example, if a group of students experiences the same issue, they can submit the complaint together by adding their names and registration numbers. Next, the form includes a section for entering accused information. If the student knows who is responsible such as a fellow student, an outsider, or an unknown individual they can enter that person's name, department, or description. If the accused person is unknown, this section can be left blank or marked as "Unknown." Student's then use the description box to explain the incident in detail. They describe what happened, when it happened, who was involved, and what actions were taken. They can also upload optional media files such as images or videos. These files help investigators understand the situation better and provide strong evidence.

Once the complaint is submitted, the system automatically stores it in the database with status "Received." As the proctors and administrators handle the complaint, they update the status to "In Progress" and later to "Resolved." The student can track their complaint through the Complaint Tracking section. This section shows a timeline of updates so the student always knows what is happening with their report.

Another major part of DU Alert is the Public Alert system. Sometimes, students want to warn others about theft, suspicious people, risky areas, or certain dangers around campus. For this purpose, DU Alert allows users to create public posts. The Public Alert form includes category selection, complainant information, accused information, a description box, and media upload just like the complaint form. However, public posts cannot directly appear on the global timeline. To prevent misinformation, false rumors, and harmful content, every public alert goes through admin approval. When a student submits a new public alert, its status becomes "Pending Approval." The admin reviews the post carefully, checking for accuracy, relevance, clarity, and appropriateness

DU Alert has another important feature anonymous posting. Some students may want to warn others but may not feel comfortable revealing their identity. To protect these students, the system

allows anonymous posts for public alerts. When a student chooses anonymous mode, the system hides their name, registration number, and personal details. The admin still reviews the post, but once approved, it appears publicly without showing who submitted it. This feature increases the confidence of students and helps them share important information without fear.

The app also includes a universal notification system. Whenever a student submits a public alert, posts anonymous content or sends an emergency signal, notifications are sent to all users of the DU Alert system. This ensures that everyone students, proctors, and admins stays informed about important events and safety issues in real time. Notifications help the entire community remain alert and prepared. Although DU Alert is mainly built for students, the proctorial team has a special place in the story. To access the system, proctors need verified accounts. These accounts are created by the admin. When a proctor logs in for the first time, they are asked to change their temporary password to secure their account. Proctors can view emergency alerts, track complaints, monitor suspicious activities, and update complaint statuses.

For the proctorial team, DU Alert provides a dedicated and powerful dashboard designed to support quick decision-making and efficient case handling. When a proctor logs in, they are taken to a dashboard where they can instantly view key statistics, including the total number of complaints, how many are currently *In Progress*, and how many have been *Resolved*. These numbers are shown clearly at the top, giving the proctor an immediate overview of the current safety situation on campus. Below this, the proctor can access a complete list of all complaints submitted by students, with filtering and sorting options based on category, status, date, or urgency. Each complaint can be opened to view full details, attached media, and previous updates, and the proctor can change the complaint status as the investigation progresses. In addition, the proctor dashboard includes an AI-powered Summary Generator, which allows the proctor to quickly generate a concise summary of individual complaints or multiple complaints at once. This feature helps proctors understand the core issues without reading long descriptions, saving time and enabling faster response and reporting.

Administrators hold the highest authority in the system. They can manage registrations, verify users, approve or reject public alerts, monitor complaint activity, and assign proctor accounts. The admin dashboard includes analytics that show monthly complaint statistics, resolved vs. pending cases, and overall system performance. This helps the university identify problem areas, understand trends, and improve campus safety policies. Together, these features create a complete safety ecosystem. DU Alert connects students with authorities in real time, encourages responsible reporting, protects anonymity, spreads awareness through notifications, and ensures that every voice is heard safely and securely. It transforms traditional university safety into a modern, digital, and community-driven system.

4. Use Case Diagram (DU Alert)

A use case is a list of actions or event steps typically defining the interactions between a role (actor) and a system to achieve a goal. The actor can be a human or other external system. In this modeling, use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. Use case diagrams are a blueprint for the system. Due to their simplistic nature, use case diagrams can be a good communication tool for stakeholders. The drawings attempt to mimic the real world and provide a view for the stakeholder to understand how the system is going to be designed. Use case diagrams consist of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

Primary Actor: Primary actors interact to achieve required system function and derive the intended benefit from the system. They work directly and frequently with the software.

Secondary Actor: Secondary actors support the system so that primary actors can do their work. They either produce or consume information.

Level 0: DU Alert System

Name: DU Alert System

Primary Actors: Student, Proctor, Proctorial_team, System

Secondary Actors: OTP Generator , Location API,Email Generator,Push Notification, Summary Generator,Graph Generator.

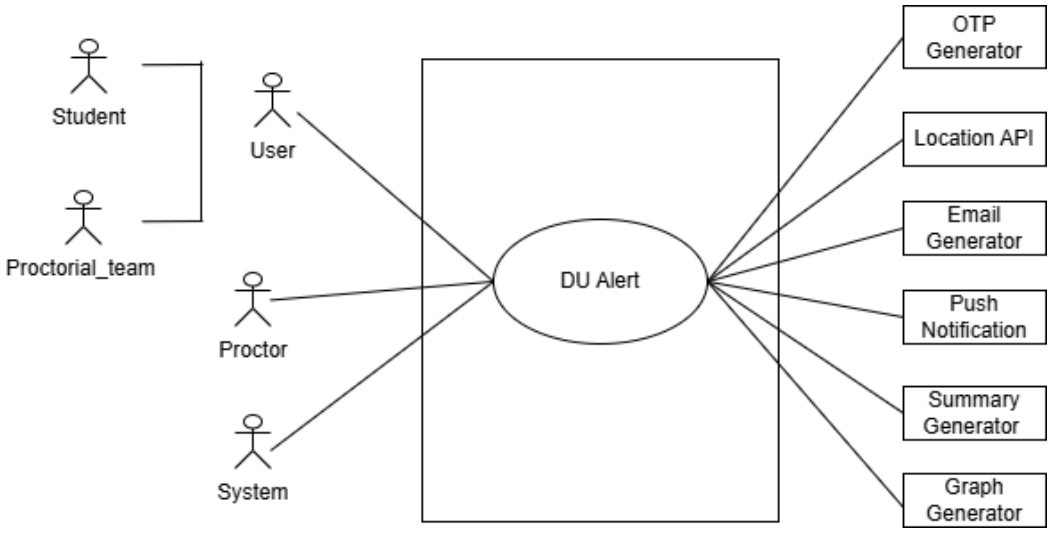


Figure 1: DU Alert Use Case Diagram

Level 1: DU Alert System

Name: DU Alert System

Primary Actors: Student, Proctor, Proctorial_team, System

Secondary Actors: OTP Generator , Location API,Email Generator,Push Notification, Summary Generator,Graph Generator.

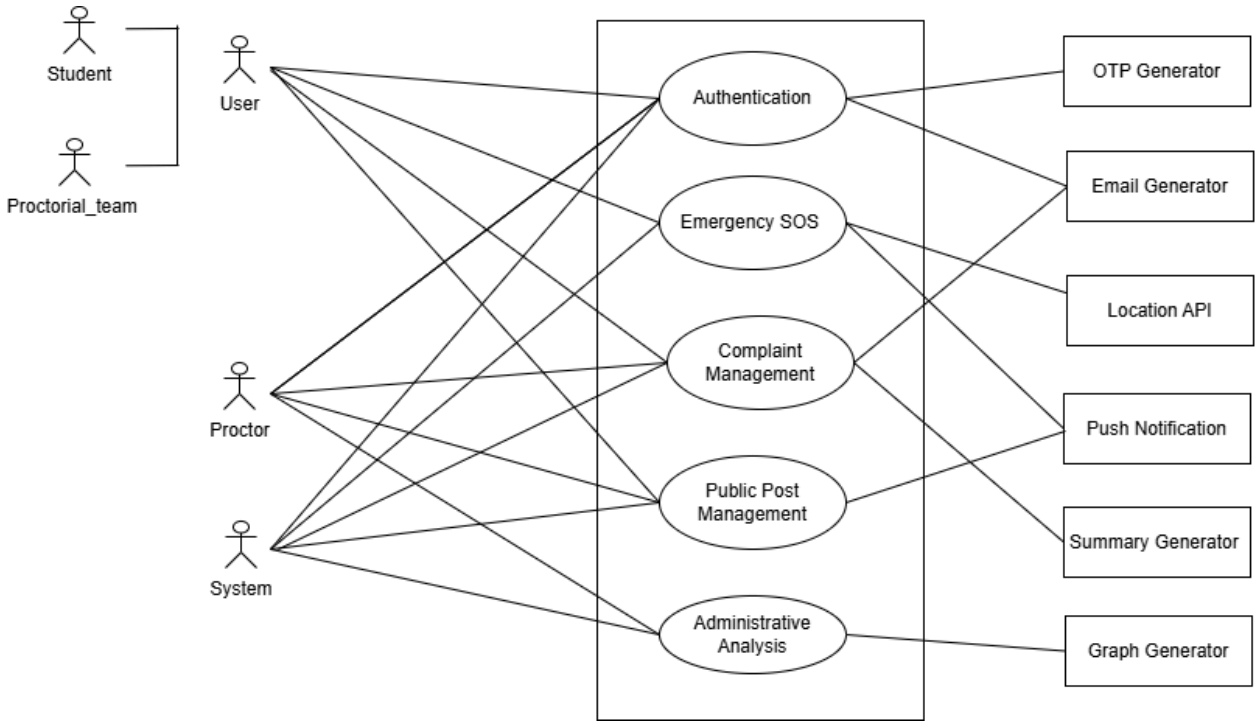


Figure 2 : DU Alert Use Case Diagram

Description:

1.Registration & Login

Users can register by selecting their role as Student, Faculty, or Staff and providing personal details like university email and phone number. The system verifies their identity through OTP to ensure security. After verification, users can create a unique User ID and password for login and password recovery.

2.Emergency SOS

Users can send an emergency SOS with their live location directly to Admin and Proctors for immediate assistance. The system tracks the user's location in real-time to ensure quick response. Emergency services are also notified instantly to handle the situation efficiently.

3.Incident Reporting

Students can report various incidents, such as crime, harassment, or accidents, directly through the app. The system allows uploading media like photos or videos to provide evidence and context. Users can track the status of their reports, and anonymous reporting is also supported to maintain privacy.

4.Public Alerts

Students can submit public alerts that require Admin approval before broadcasting to the entire user base. These alerts include essential details like location, severity, and description of the incident. Approved alerts ensure the campus community stays informed about important safety issues.

5.Notifications

The system sends real-time notifications for emergencies, incident updates, and other important announcements. Users receive timely alerts to stay aware and take necessary action. Notifications are an essential part of keeping everyone informed and safe on campus.

6.Analytics & AI Summary

The Admin dashboard provides comprehensive statistics on incidents, response times, and safety trends. AI-powered summaries analyze reports to give quick insights for better decision-making. This helps improve campus safety management and emergency preparedness.

Level 1.1: Authentication

Name: Registration & Login

Primary Actors: Student, Proctor, Proctorial_team, Student,System

Secondary Actors: OTP Generator , Email

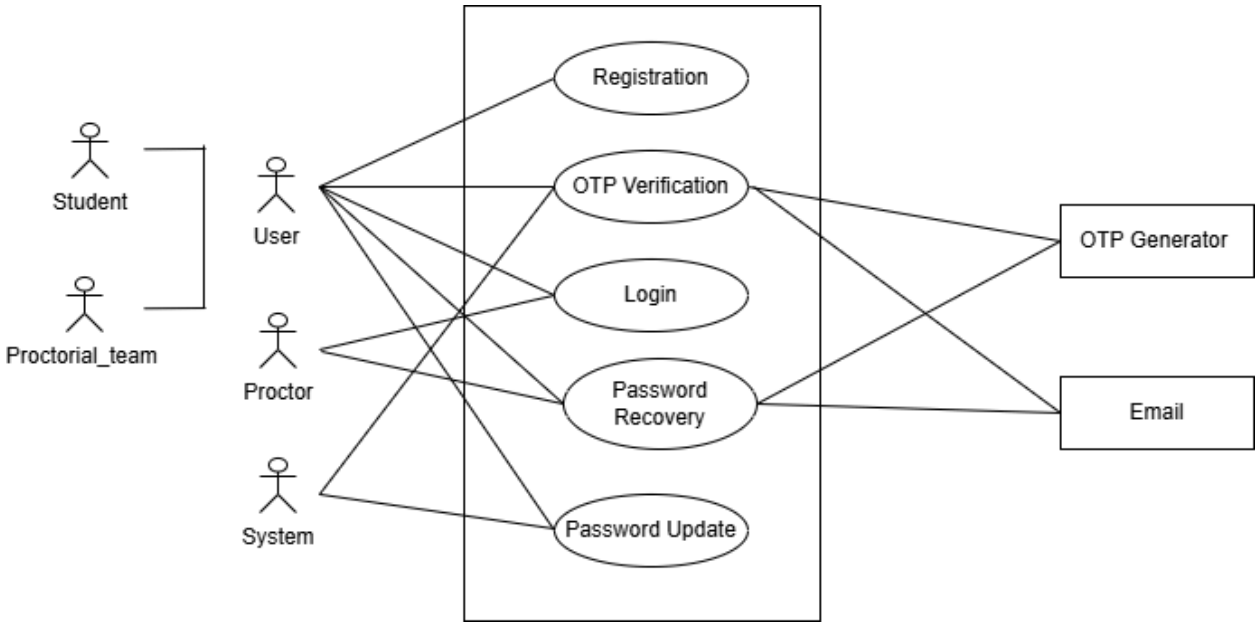


Figure 3 : Registration & Login Use Case Diagram

Description:

1. Registration

New users, including students, Proctorial team begin by selecting their user type and providing their phone number along with required role-specific details. For students, this includes roll number, registration number, institutional email, and address.

2. OTP Verification

An OTP is sent via the OTP Generator to the user's registered phone number to confirm their identity. This process is used both during initial registration and for password recovery to ensure that only authorized users can access or modify their accounts. Successful verification allows the process to continue.

3. Login

After successful registration, users log in using their User ID and password. The system validates credentials and grants access to the role-specific dashboard.

4. Password Recovery

If a user forgets their password, they can enter their username and registered phone number to receive an OTP for identity verification. After verifying the OTP, they can set a new password. The Email Generator then sends a confirmation email to ensure account security.

Level 1.2 : Emergency SOS

Name: Emergency SOS

Primary Actors: Student, Proctorial_team, Student,System

Secondary Actors: Push Notification , Location API

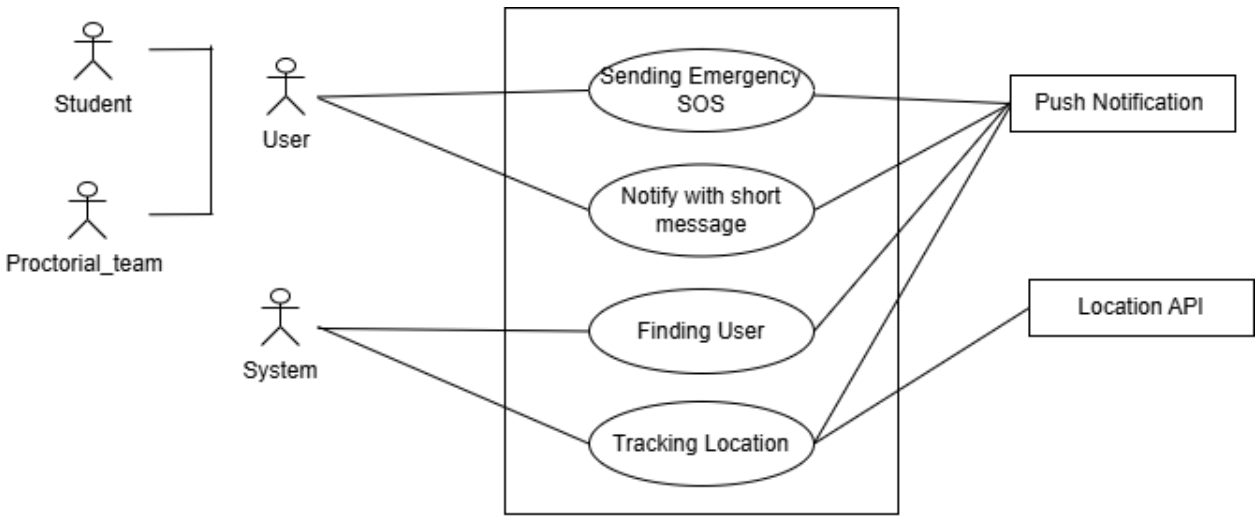


Figure 4: Emergency SOS Use Case Diagram

Description :

1. Sending Emergency SOS

Users can send an emergency SOS directly from their dashboard or mobile interface by pressing the SOS button. This action immediately notifies the system of a critical situation. The SOS request includes the user's role, location, and any additional details provided.

2. Tracking Location

The system captures the real-time GPS location of the user sending the SOS. This location is continuously updated to ensure accurate tracking. Administrators, proctors, and emergency responders can view the user's live location for rapid assistance.

3. Notifications

Upon receiving an SOS, the system automatically sends notifications to designated authorities, such as the campus admin, proctors, or emergency services. Users are also notified about the status of their SOS request. This ensures that help is mobilized immediately and efficiently.

4. Finding User

The system enables authorized personnel to locate the user quickly using the live location data. It also provides maps and directions for responders to reach the user promptly. This feature helps reduce response time and improves safety during emergencies.

Level 1.3 : Complaint Management

Name: Complaint Management

Primary Actors: Student, Proctorial_team, Student,System , Proctor .

Secondary Actors: Push Notification

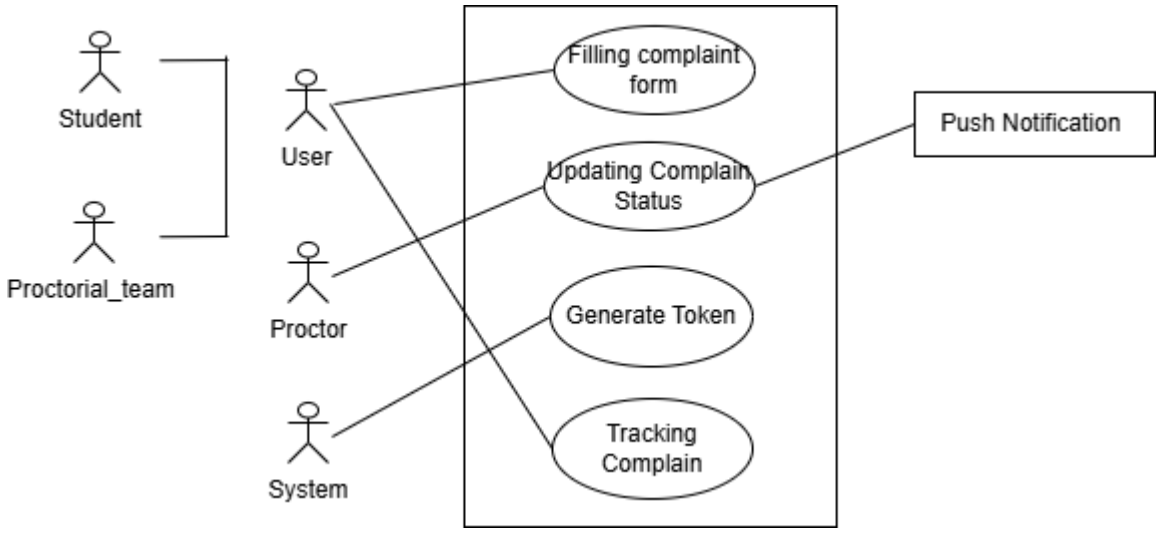


Figure 4: Complaint Management Use Case Diagram

Description :

1. Filling Complaint

Users can submit complaints regarding incidents, harassment, accidents, or other concerns through the platform. The complaint form allows adding details, selecting complaint type, and uploading supporting media such as photos or videos. Once submitted, the system generates a unique complaint record for tracking.

2. Updating Complaint

Users can update their complaints with additional information or corrections after submission. The system ensures that updates are recorded chronologically for transparency. Administrators and assigned personnel are notified of any changes to ensure proper handling.

3. Generate Token

After a complaint is submitted, the system generates a unique token ID for the user. This token allows the user to track the status of their complaint securely. It also serves as a reference for administrators to process and resolve the complaint efficiently.

4. Tracking Complaint

Users can track the status of their complaints in real-time through the dashboard using the generated token. Status updates include pending, in-progress, and resolved stages. Notifications are sent to keep users informed about actions taken and final resolution.

Level 1.4 :Public Post Management

Name: Public Post Management

Primary Actors: Student, Proctorial_team, Student,System , Proctor .

Secondary Actors: Push Notification

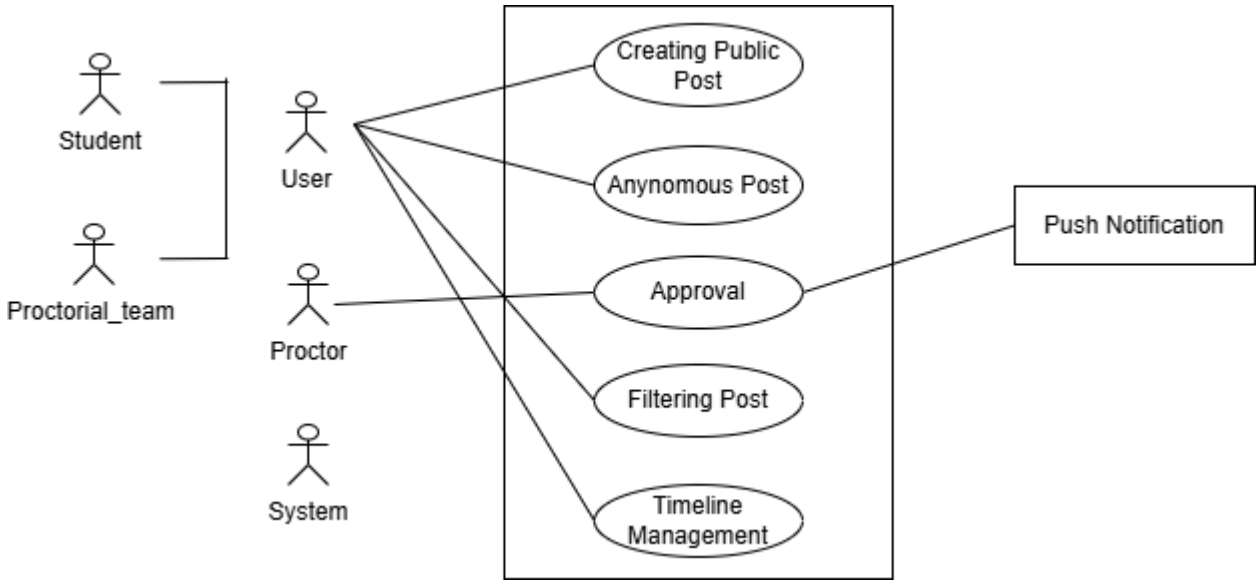


Figure 5: Public Post Management Use Case Diagram

Description:

1. Creating Public Post

Users can create public posts to share important updates, safety tips, or announcements with the university community. Each post includes a title, description, and optional media attachments such as images or videos. Once submitted, the post enters the system for processing and visibility on the platform.

2. Anonymous Post

Users have the option to submit posts anonymously to protect their identity. Anonymous posts allow safe sharing of sensitive information, concerns, or tips. The system ensures that the creator’s identity remains hidden while maintaining post accountability for moderation.

3. Approval

All public posts undergo approval by the admin before being broadcasted to the wider community. The approval process ensures that posts are relevant, appropriate, and free of harmful or misleading content. Approved posts are then made visible on the platform’s timeline for all users.

4. Filtering

The system provides filtering options for posts based on categories, severity, date, or location. Users can easily find relevant updates without being overwhelmed by unrelated content. Filtering ensures efficient access to important safety information and alerts.

5. Timeline Post

Approved posts are displayed on a timeline for easy viewing and interaction. Users can see the latest updates in chronological order and engage with them via likes, comments, or shares if allowed. The timeline ensures that critical information is visible in real-time to all users.

Level 1.5 :Administrative Analysis

Name: Administrative Analysis

Primary Actors: System , Proctor .

Secondary Actors: Summary Generator , Graph Generator

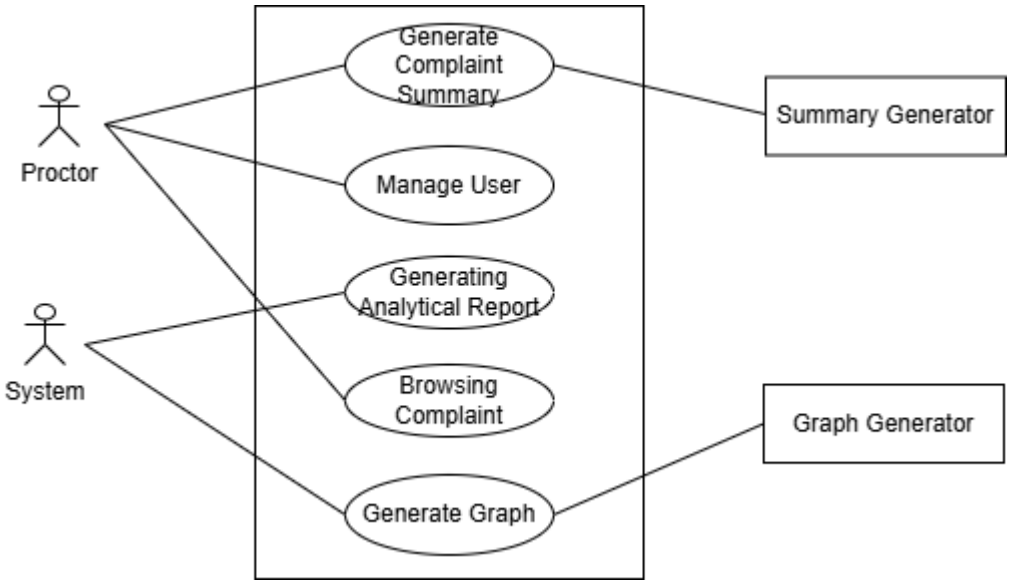


Figure 6:Administrative Analysis Use Case Diagram

Description :

1. Generate Complaint Summary

Admins can generate summaries of complaints received across the platform. These summaries include key details such as type, location, severity, and status of each complaint. This helps in quickly understanding trends and prioritizing urgent issues.

2. Manage Users

The system allows administrators to manage registered users, including students, faculty, and staff. Admins can view user details, update roles, suspend accounts, or handle flagged accounts. Effective user management ensures platform security and accountability.

3. Generating Analytical Report

Administrators can generate detailed analytical reports based on complaints, incidents, or SOS alerts. Reports include statistics, trends, and summaries that assist in decision-making and improving emergency response. These reports can be exported for record-keeping or sharing with authorities.

4. Browsing Complaint

The platform provides a searchable interface for admins to browse complaints by category, date, severity, or user. This enables quick access to relevant complaints for review or follow-up actions. Efficient browsing helps in faster resolution and monitoring.

5. Graph Generator

Admins can visualize complaint and incident data through graphs and charts. Graphs display trends, response times, and the distribution of incidents across different categories. This visual analysis aids in quick insights and informed administrative decision

5. Activity Diagram

Activity diagram is an important behavioral diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flowchart that models the flow from one activity to another activity.

Level 1.1: Registration & Login

Name: Registration & Login
Reference: Use Case Diagram Level 1.1

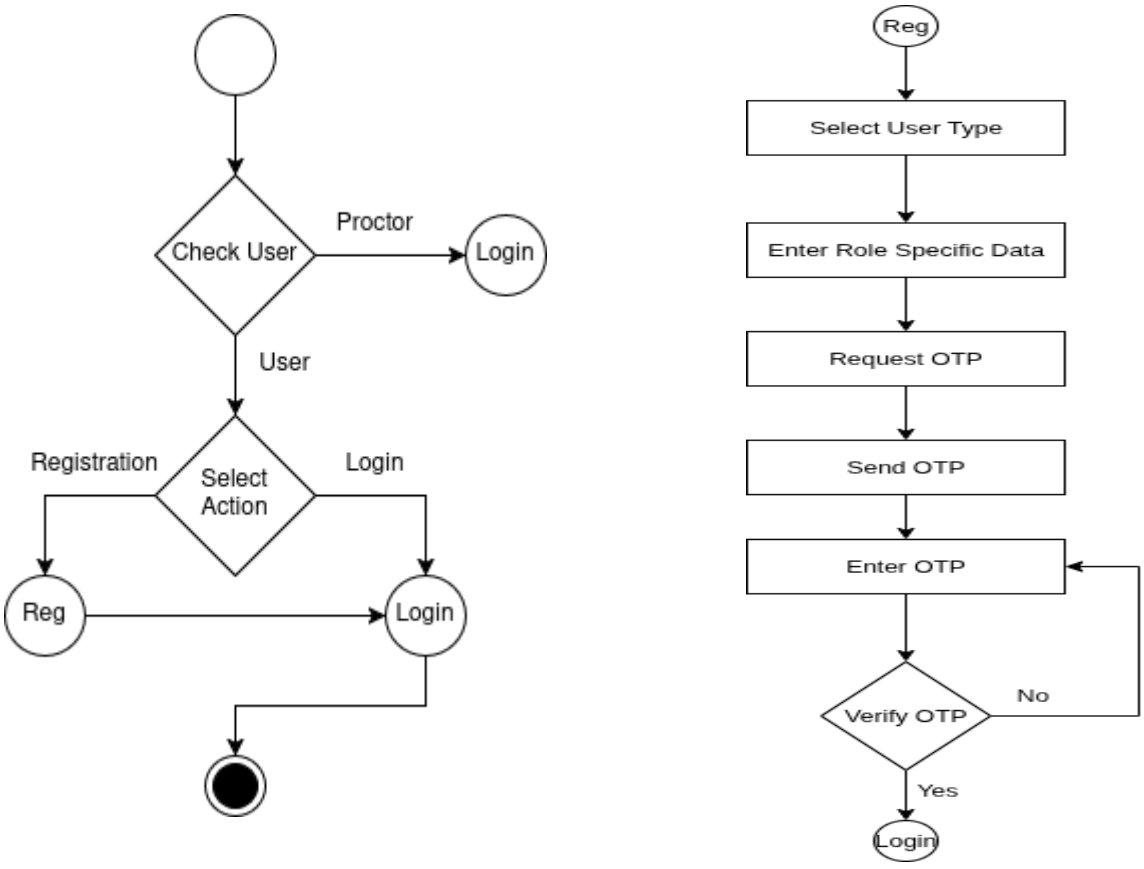


Figure 7 : Registration & Login Activity Diagram

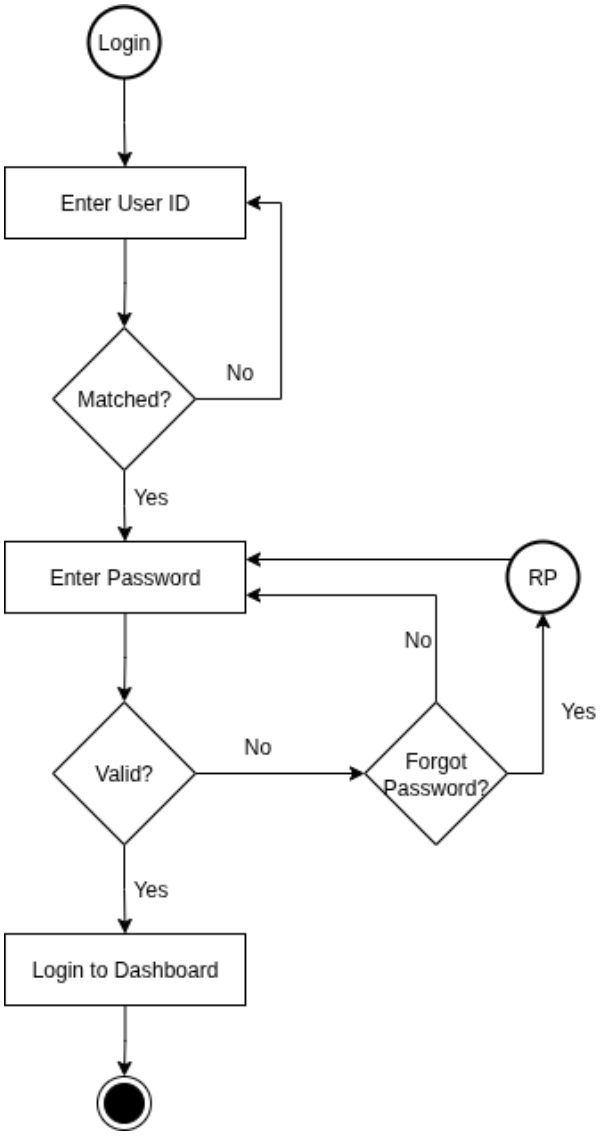


Figure 8 : Login Activity Diagram

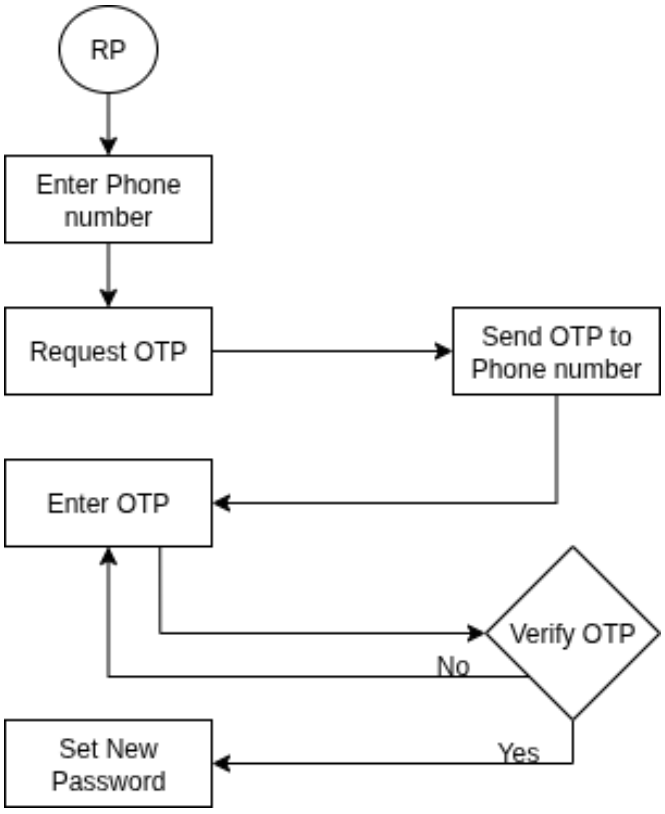


Figure 9 : Recovery Password Activity Diagram

Level 1.2: Emergency SOS

Name: Emergency SOS
Reference: Use Case Diagram Level 1.2

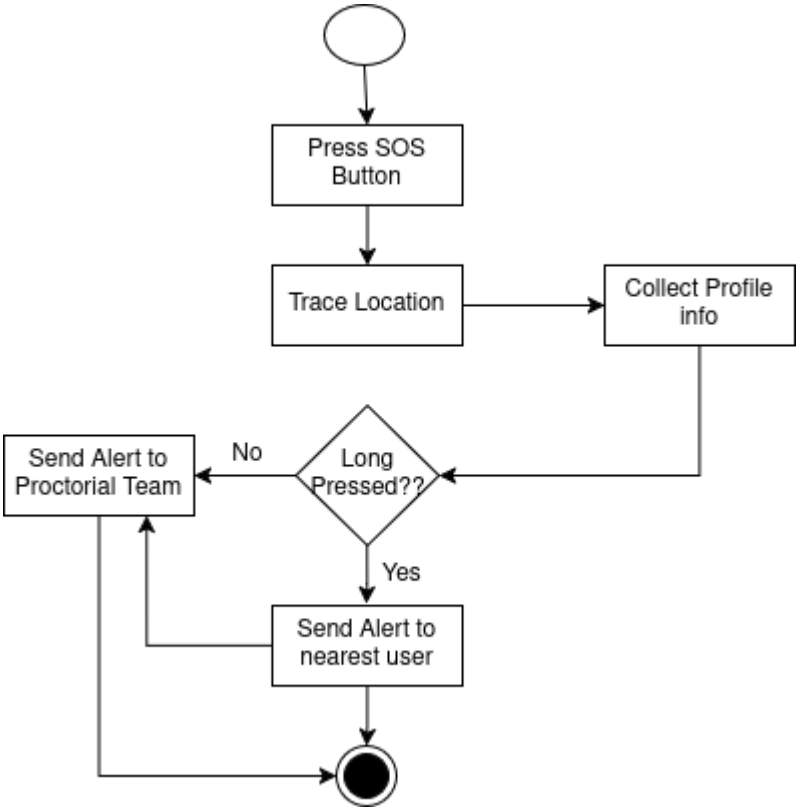


Figure 10 : Emergency SOS Activity Diagram

Level 1.3: Complain Management

Name: Complaint Management
Reference: Use Case Diagram Level 1.3

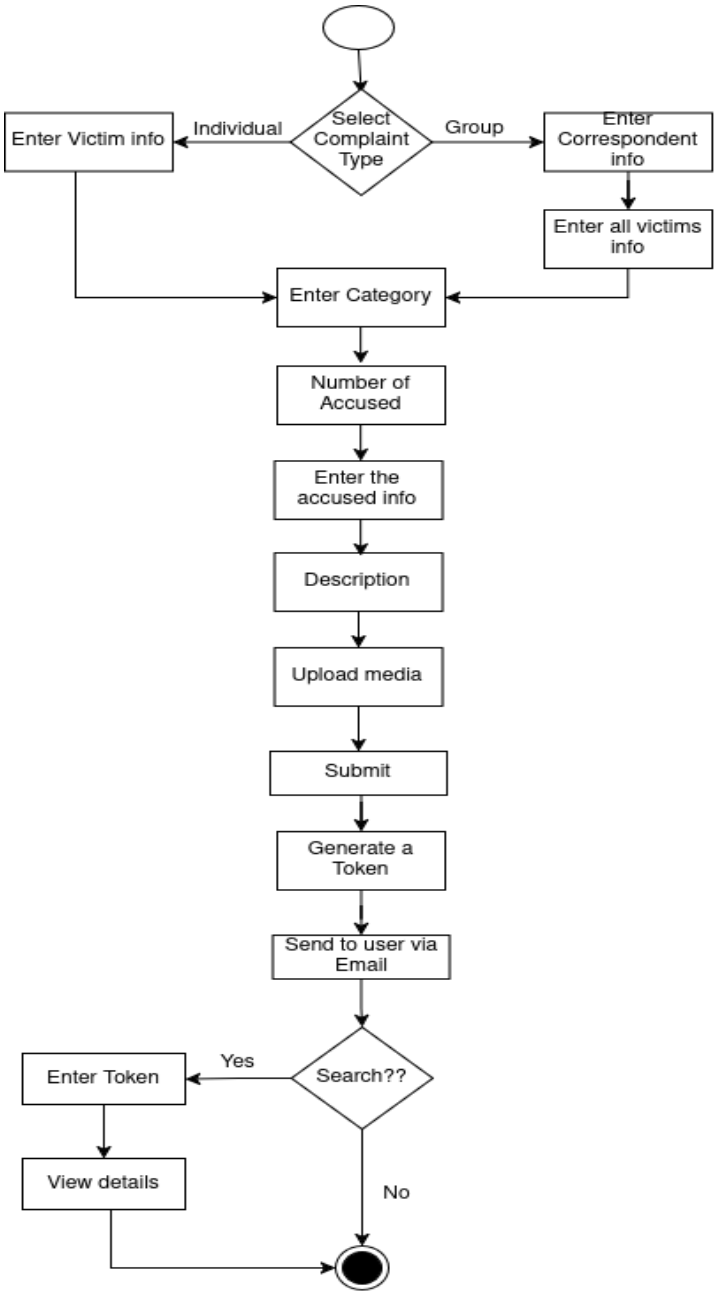


Figure 11 : Complaint mangemnet Activity Diagram

Level 1.4: Public Post Management

Name: Public Post Management
Reference: Use Case Diagram Level 1.4

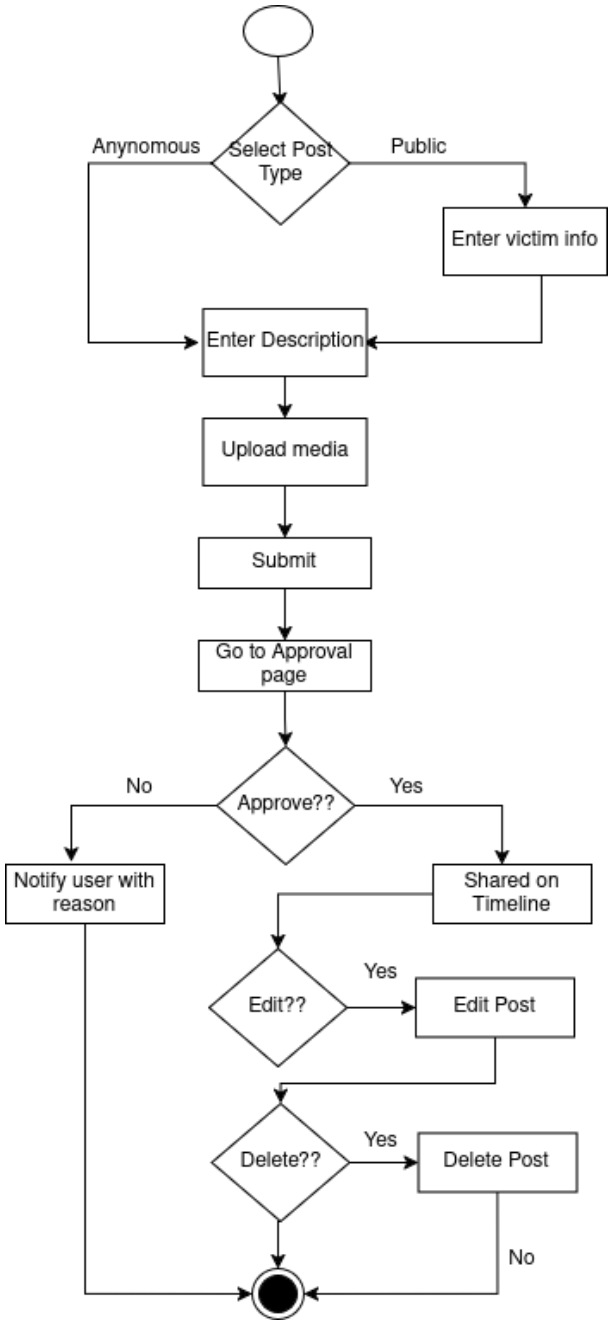


Figure 12 : Public post mangemnet Activity Diagram

Level 1.5: Administrative

Name: Public Post Management
Reference: Use Case Diagram Level 1.5

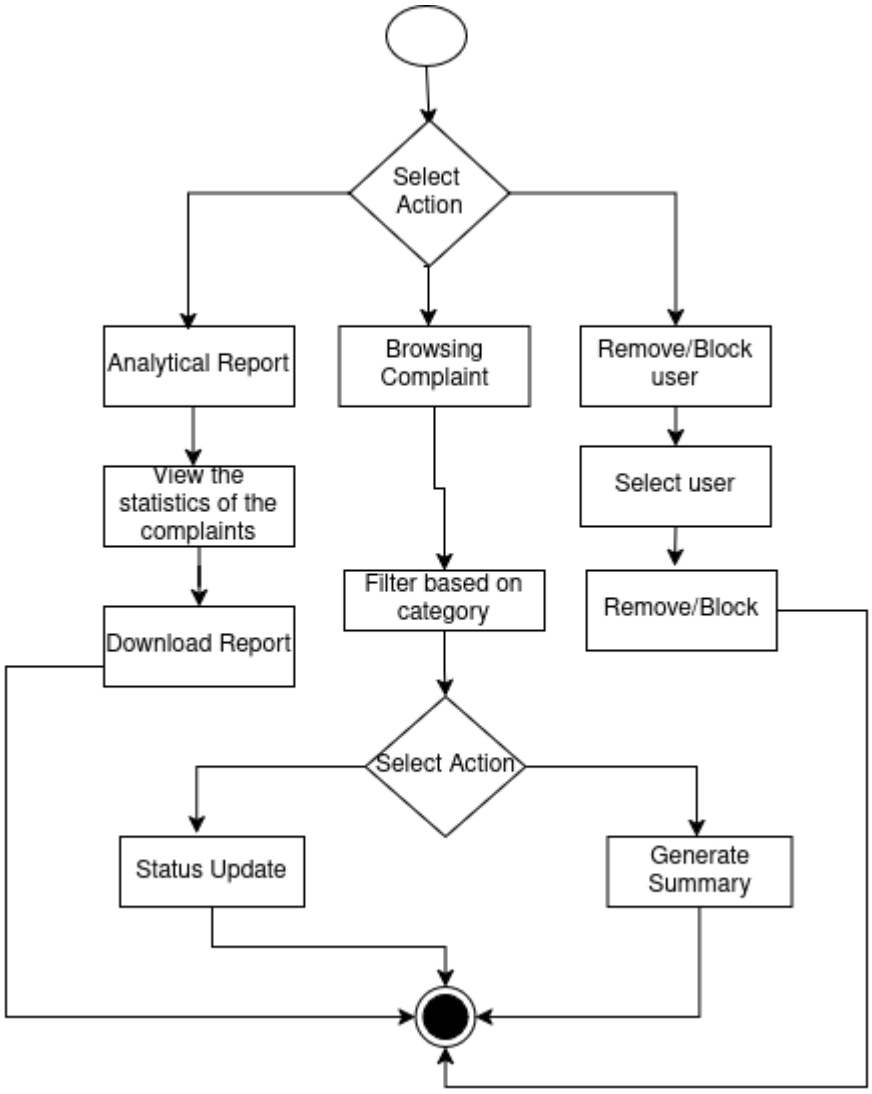


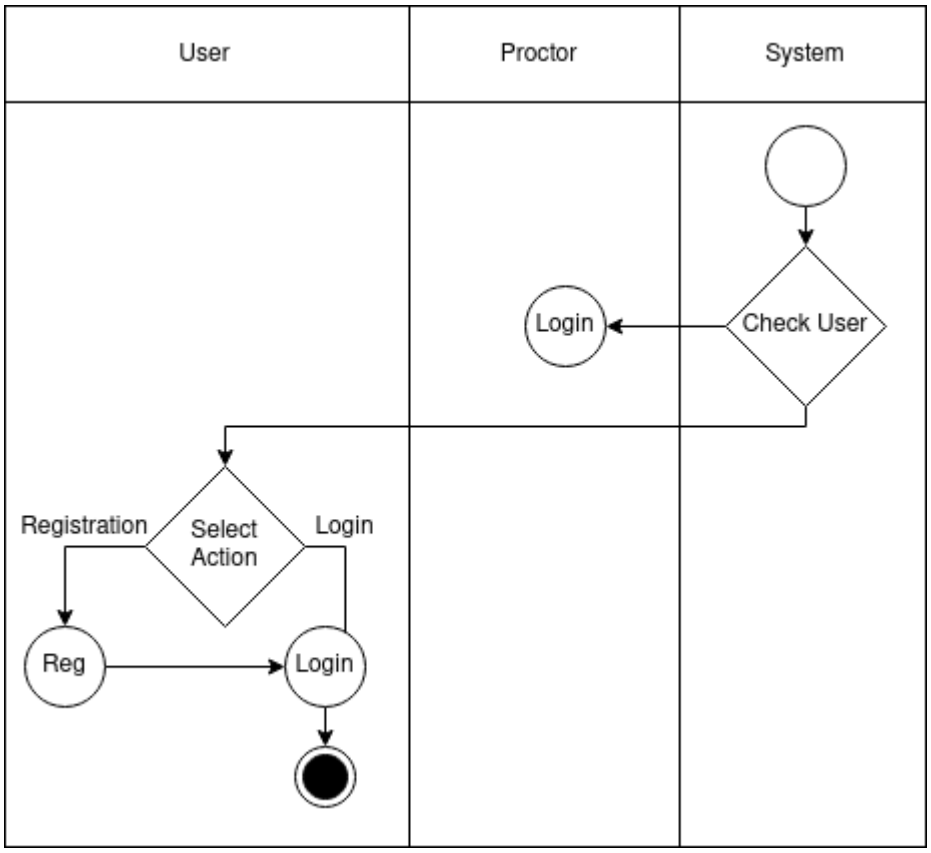
Figure 13 : Administrative Activity Diagram

6. Swimlane Diagram :

A Swimlane diagram is a type of flowchart, which diagrams a process from start to finish, but it also divides these steps into categories to help distinguish which departments or employees are responsible for each set of actions. It is based on the analogy of lanes in a pool, as it places process steps within the horizontal or vertical “Swimlanes” of a particular department, work group, or employee, thus ensuring clarity and accountability.

Level 1.1: Registration & Login

Name: Registration & Login
Reference: Activity Diagram Level 1.1



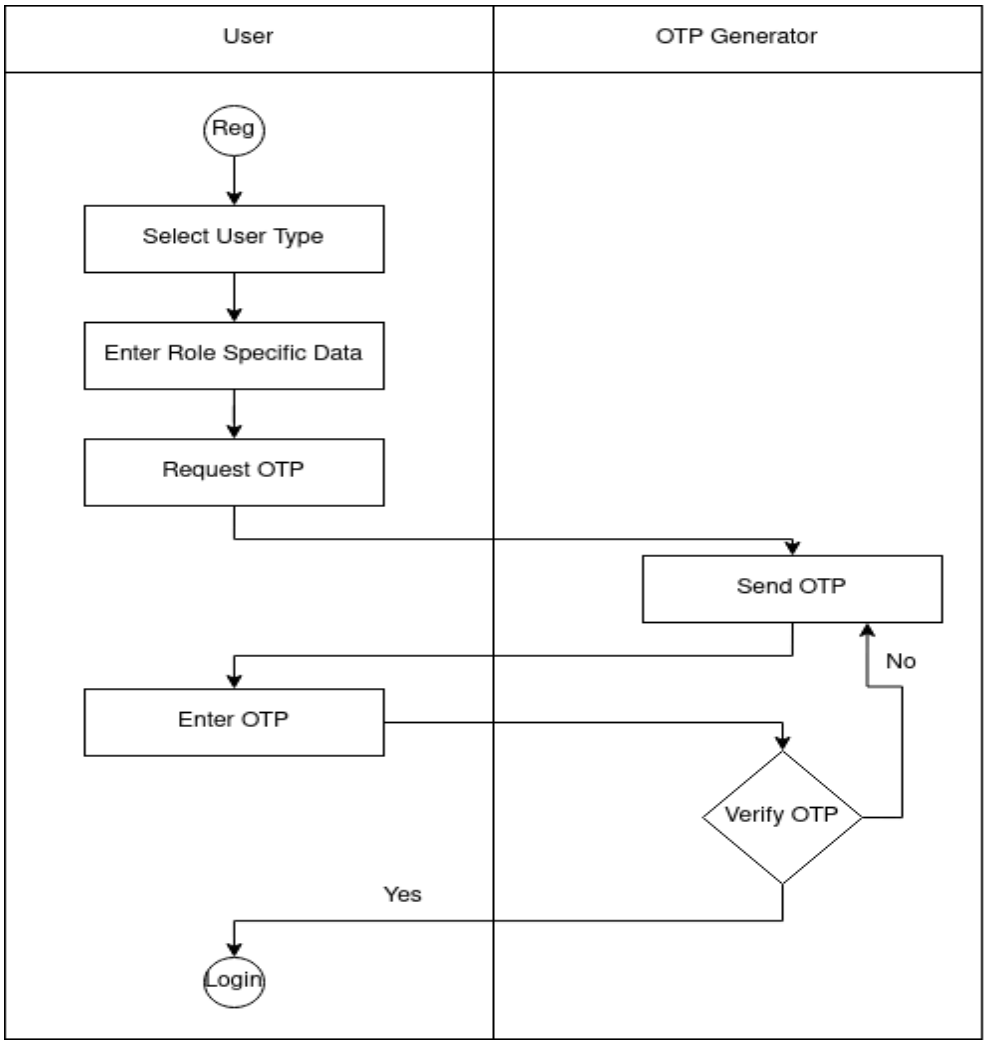


Figure 14 : Registration & Login Swimlane Diagram

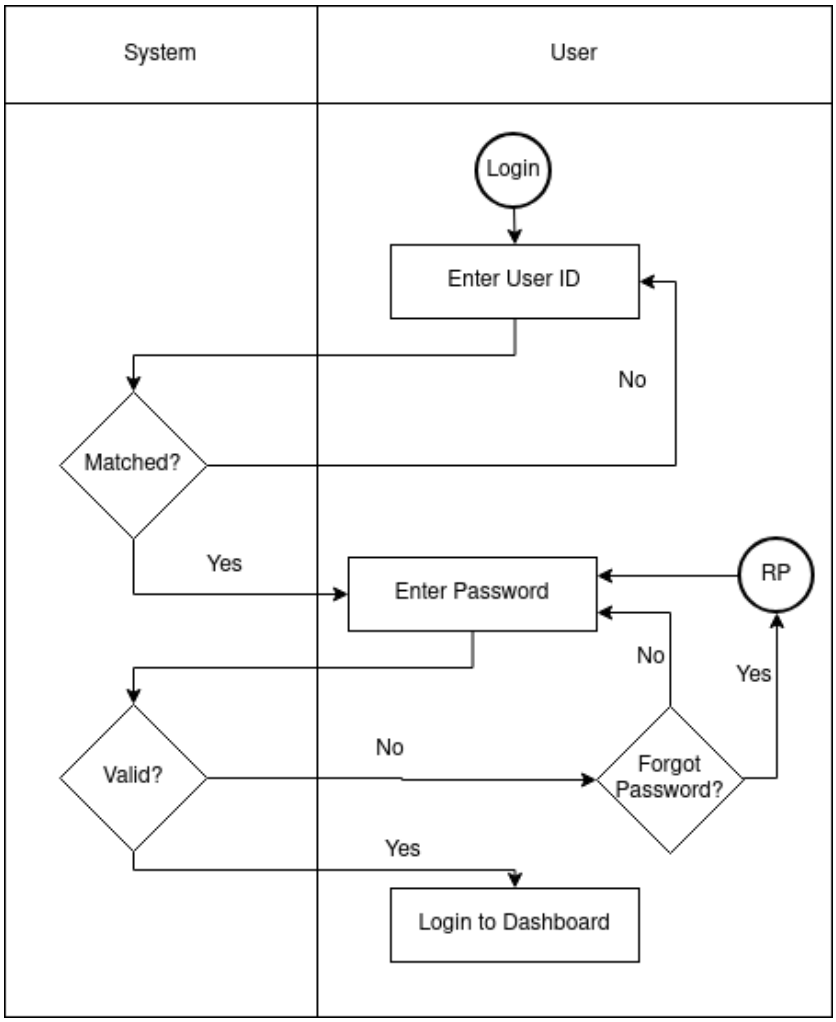


Figure 15 : Registration & Login Swimlane Diagram

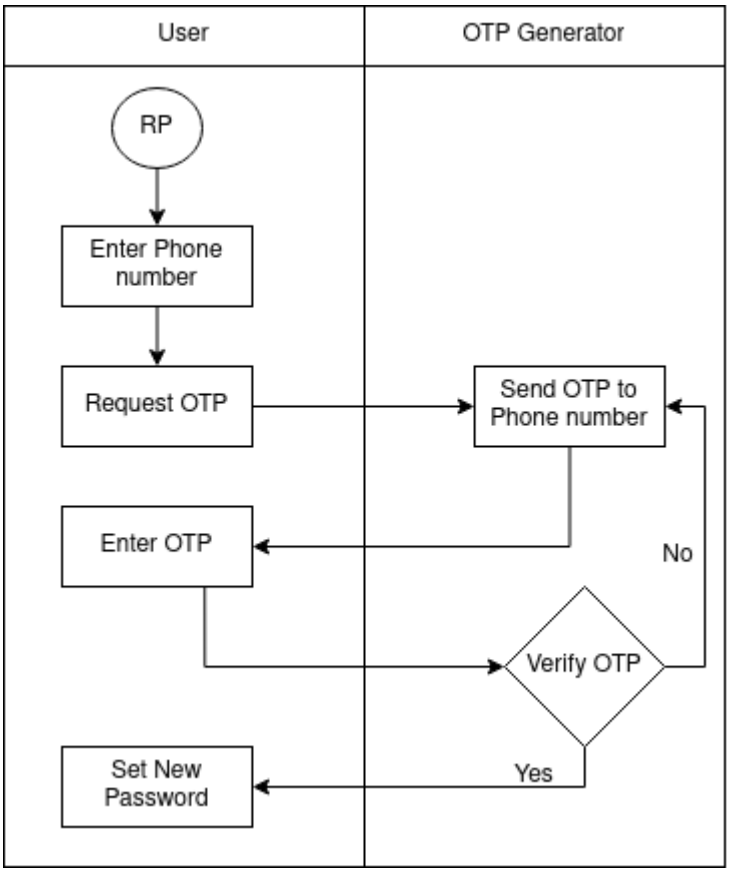


Figure 16 : Recovery password Swimlane Diagram

Level 1.2:Emergency SOS

Name: Emergency SOS
Reference: Activity Diagram Level 1.2

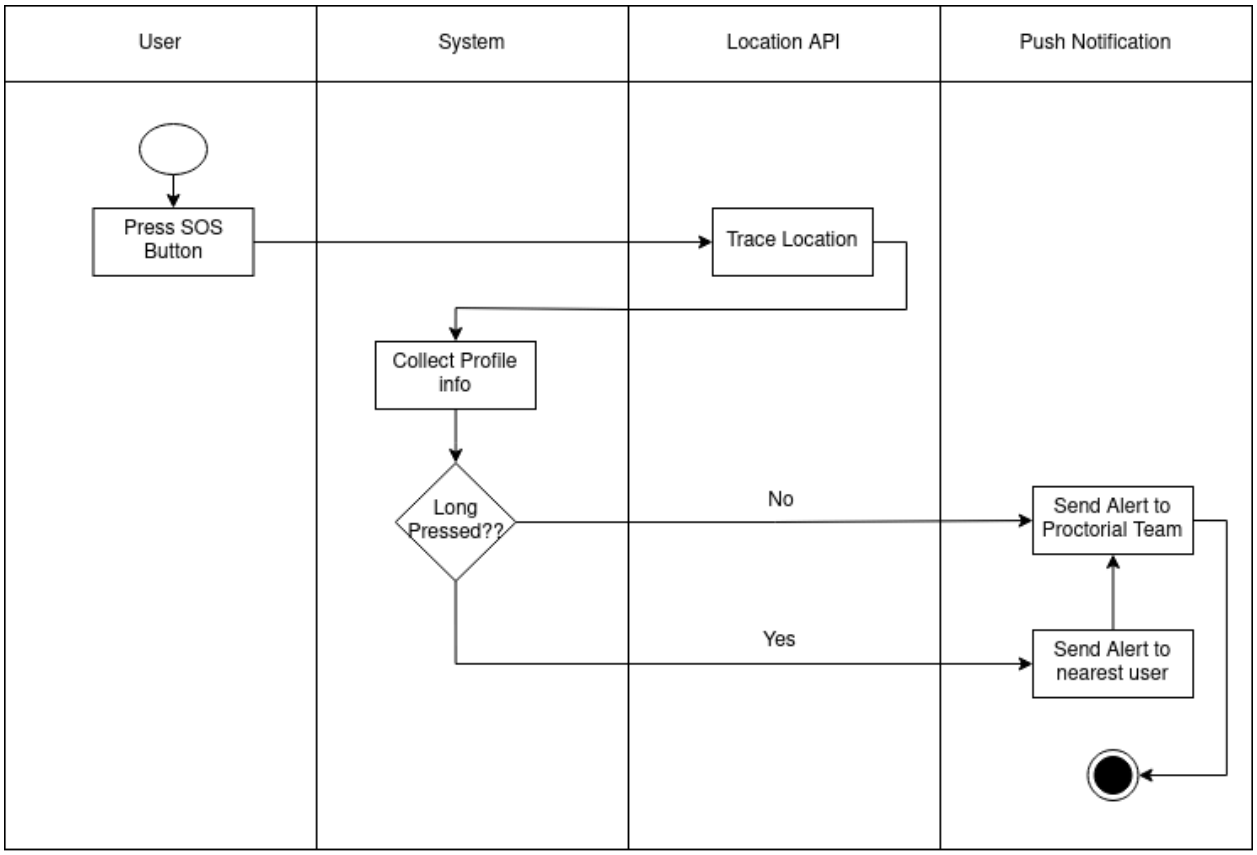


Figure 17:: Emergency SOS Swimlane Diagram

Level 1.3:Complaint Management

Name: Complaint Management
Reference: Activity Diagram Level 1.3

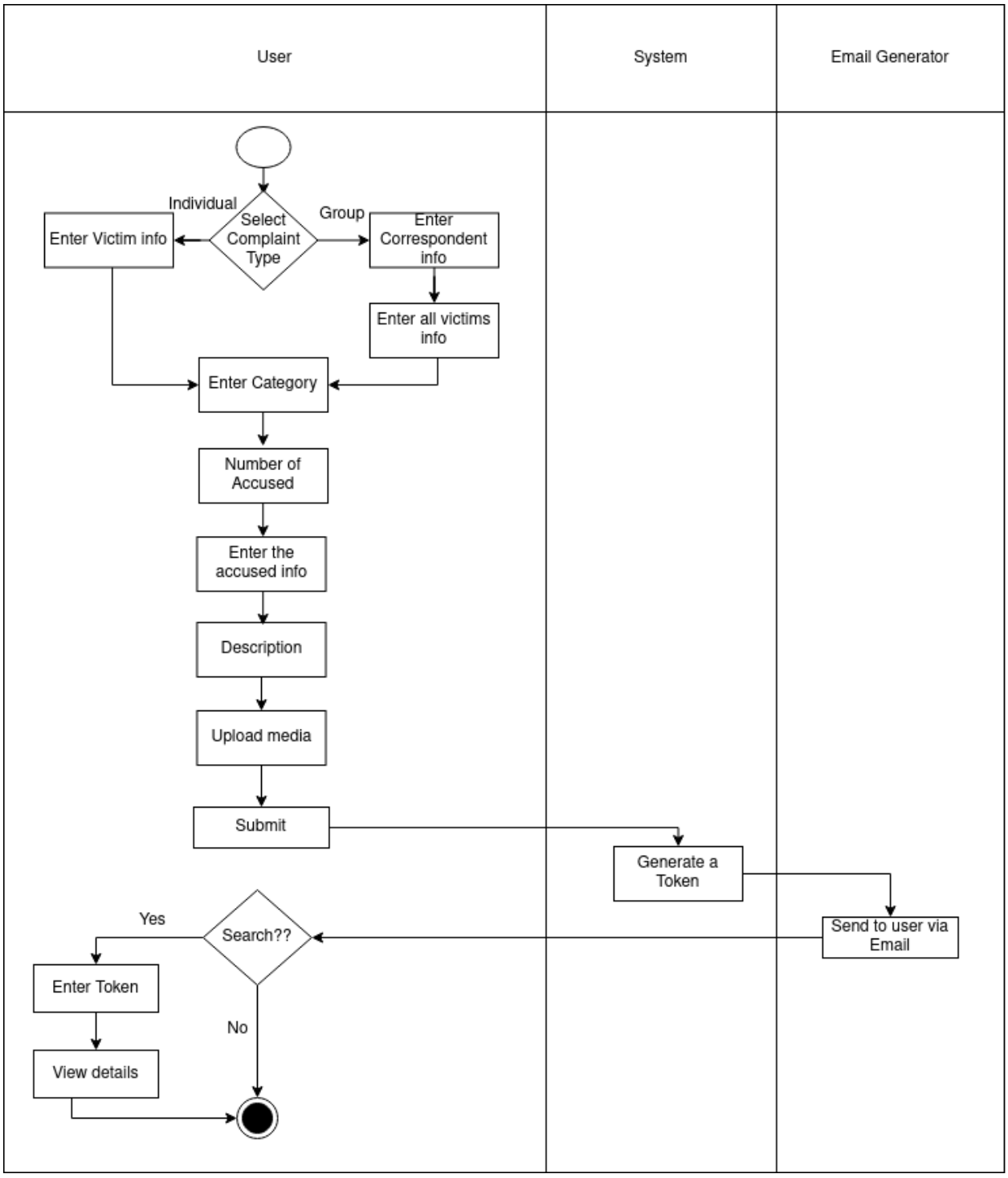


Figure 18 : Complaint management Swimlane Diagram

Level 1.4:Post Management

Name: Post Management
Reference: Activity Diagram Level 1.4

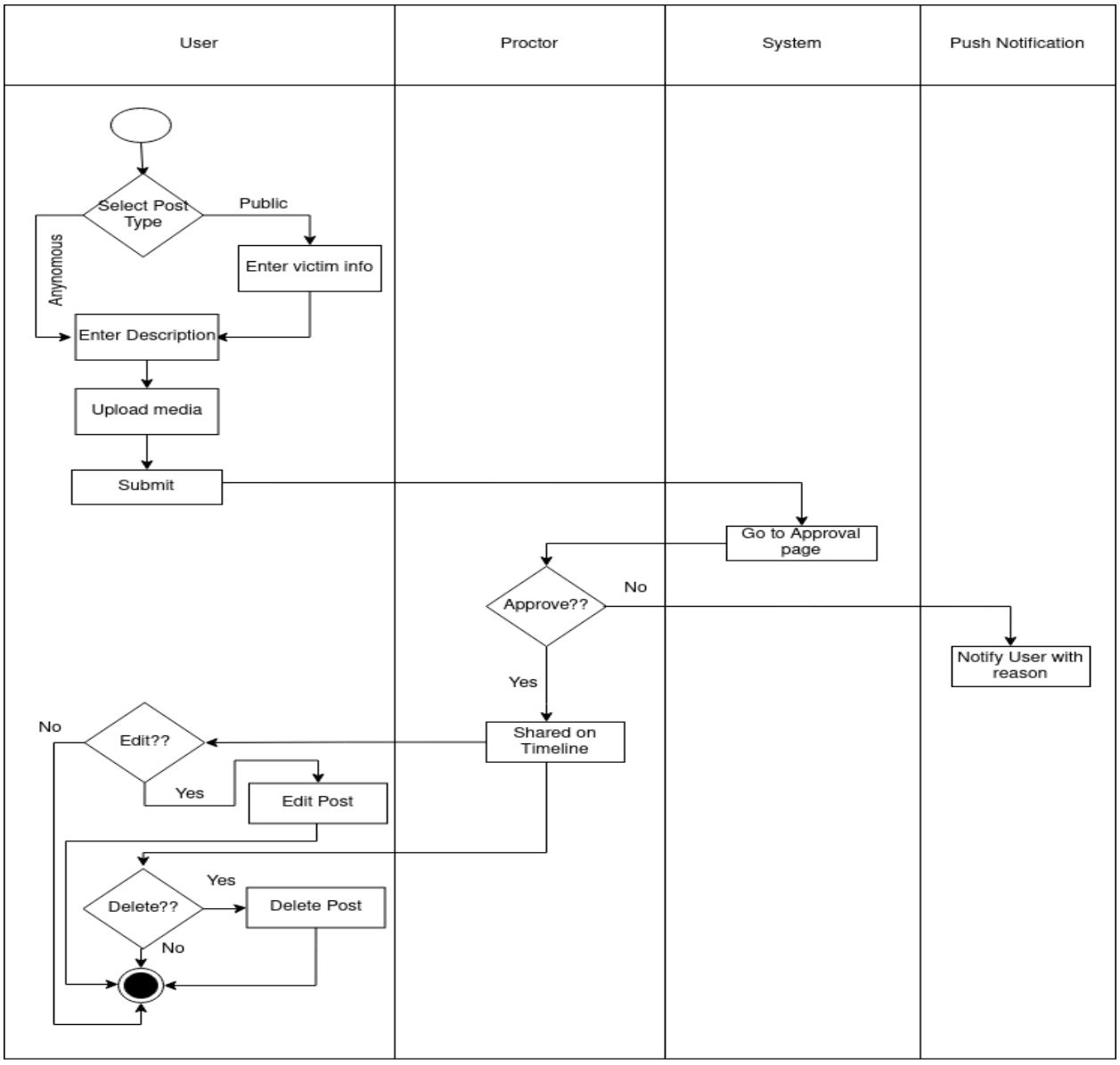


Figure 18: POST management Swimlane Diagram

Level 1.5:Administrative

Name: Administrative
Reference: Activity Diagram Level 1.5

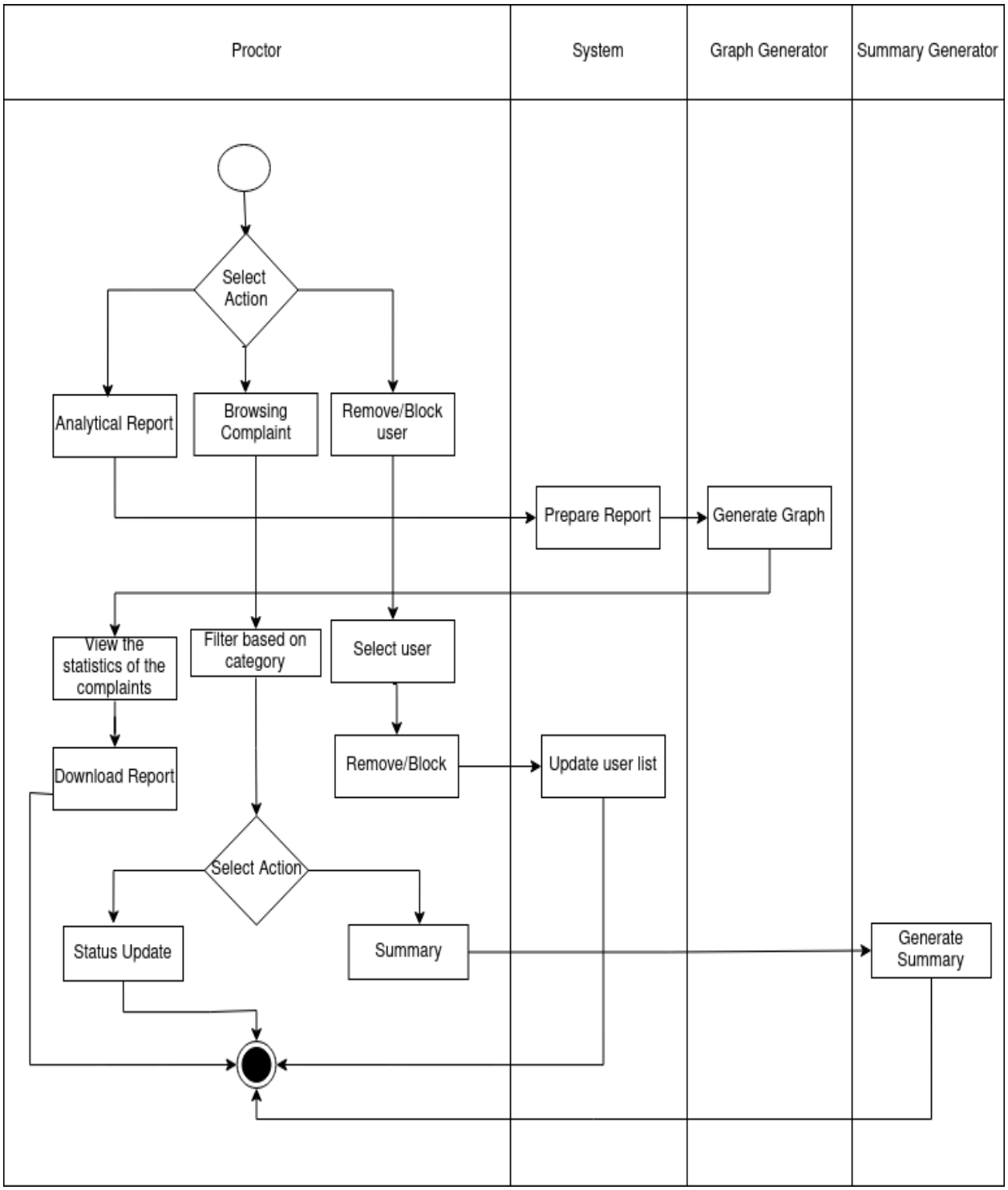


Figure 18: Administrative Swimlane Diagram

7. Data Based Modeling

Data-based modeling focuses on identifying, organizing, and defining the data objects used in the DU Alert system. It helps in understanding how data is created, stored, accessed, and managed throughout the system, ensuring data consistency, integrity, and security.

7.1 Data Object Identification

No.	Noun	P/S	attribute
1.	User	S	12,13,14,15,16,23,67
2.	Student	S	
3.	OTP	S	31,32,36
4.	account	S	
5.	complaint	S	18,21,22,33,37,44,45
6.	Public Post	S	14,18,22,38,39,40,41,72,74
7.	notification	S	30,31,32,41
8.	proctor	S	
9.	Admin	S	
10.	Emergency SOS	S	22,26,34,35
11.	Name	S	
12.	User id	S	
13.	E-mail	S	
14.	Phone Number	S	

15.	Username	S	
16.	Password	S	
17.	Student	S	12,13,14,15,16,23,42,67,75
18.	Media	S	
19.	Location	S	20,27,28,29,77
20.	Location id	S	
21.	Status	S	
22.	Timestamp	S	
23.	Role	S	
24.	Category	S	
25.	Registration Number	S	
26.	Security unit	S	
27.	Latitude	S	
28.	Longitude	S	
29.	Area Name	S	
30.	Notification id	S	
31.	Time	S	
32.	Date	S	

33.	Description	S	
34.	SOS id	S	
35.	Action Status	S	
36.	OTP number	S	
37.	Complaint id	S	
38.	Alert approval	S	
39.	Anonymous post	S	
40.	Alert approval	S	
41.	Message	S	
42.	Department	S	
43.	Complainant id	S	
44.	Accused info	S	
45.	Complaint category	S	
46.	Emergency response	S	
47.	Proctorial Team	S	29,49
48.	AI summary	S	
49.	Designation	S	
50.	Verification token	S	

51.	Forgotten password request	P	
52.	Campus map	P	
53.	Student group	P	
54.	Incident log	P	
55.	Feedback	P	
56.	Activity log	P	
57.	System setting	P	
58.	Password history	P	
59.	Public notification	P	
60.	Safety tip	P	
61.	University Event	P	
62.	Contact info	S	
63.	User preference	P	
64.	Alert history	S	
65.	Student profile	P	
66.	Security incident	P	
67.	Anonymous user	S	
68.	Event notification	P	

69.	Report file	P	
70.	Warning sign	P	
71.	Threat alert	P	
72.	Summary	S	
73.	System log	P	
74.	Post_id	S	
75.	Session	S	
76.	Alert priority	S	
77.	Location history	S	

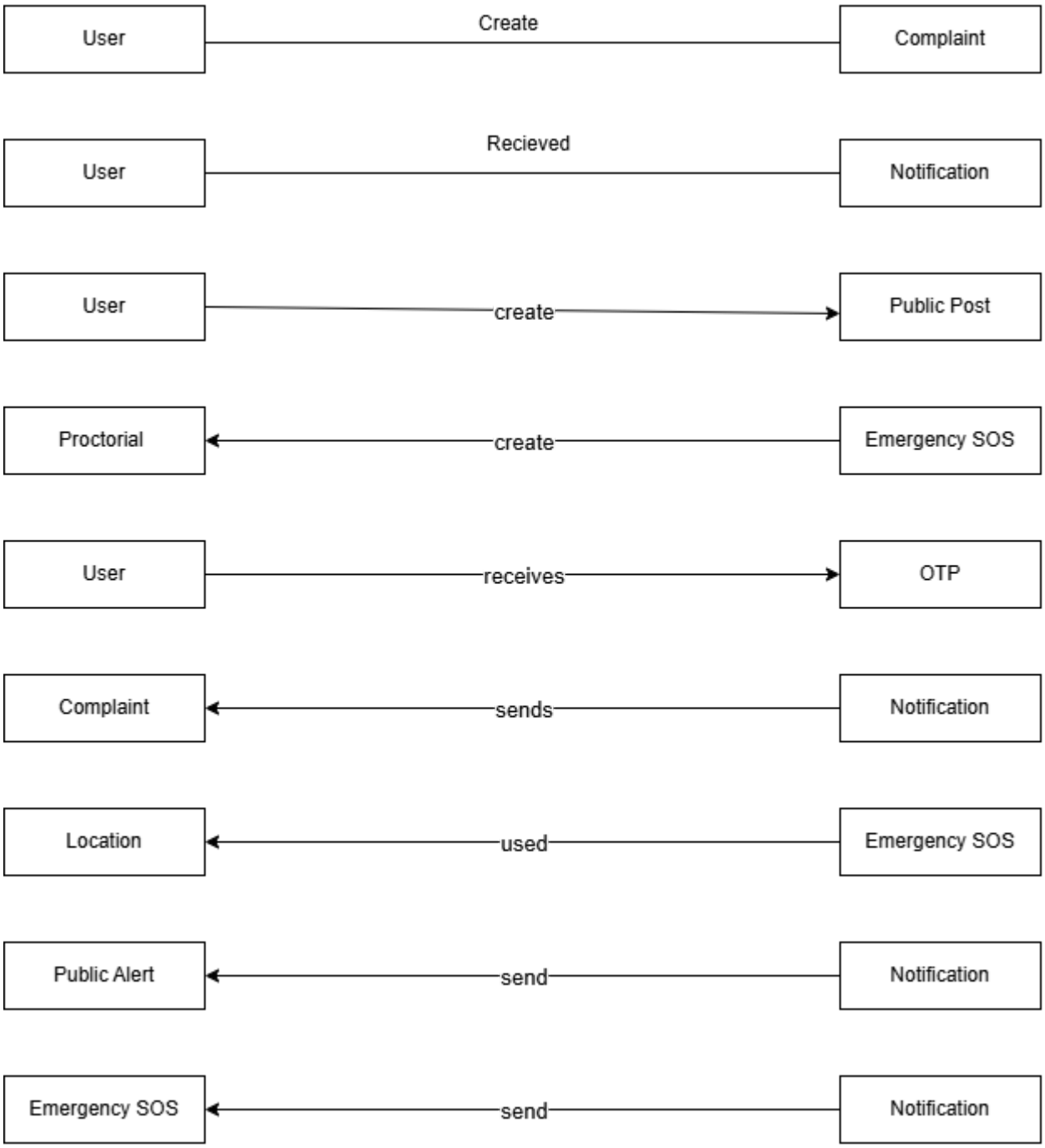
7.2 Data Object

A data object is a representation of composite information that must be understood by the software. Here composite information means information that has a number of different properties or attributes. A data object can be an external entity, a thing, an occurrence, a role, an organizational unit, a place or a structure.

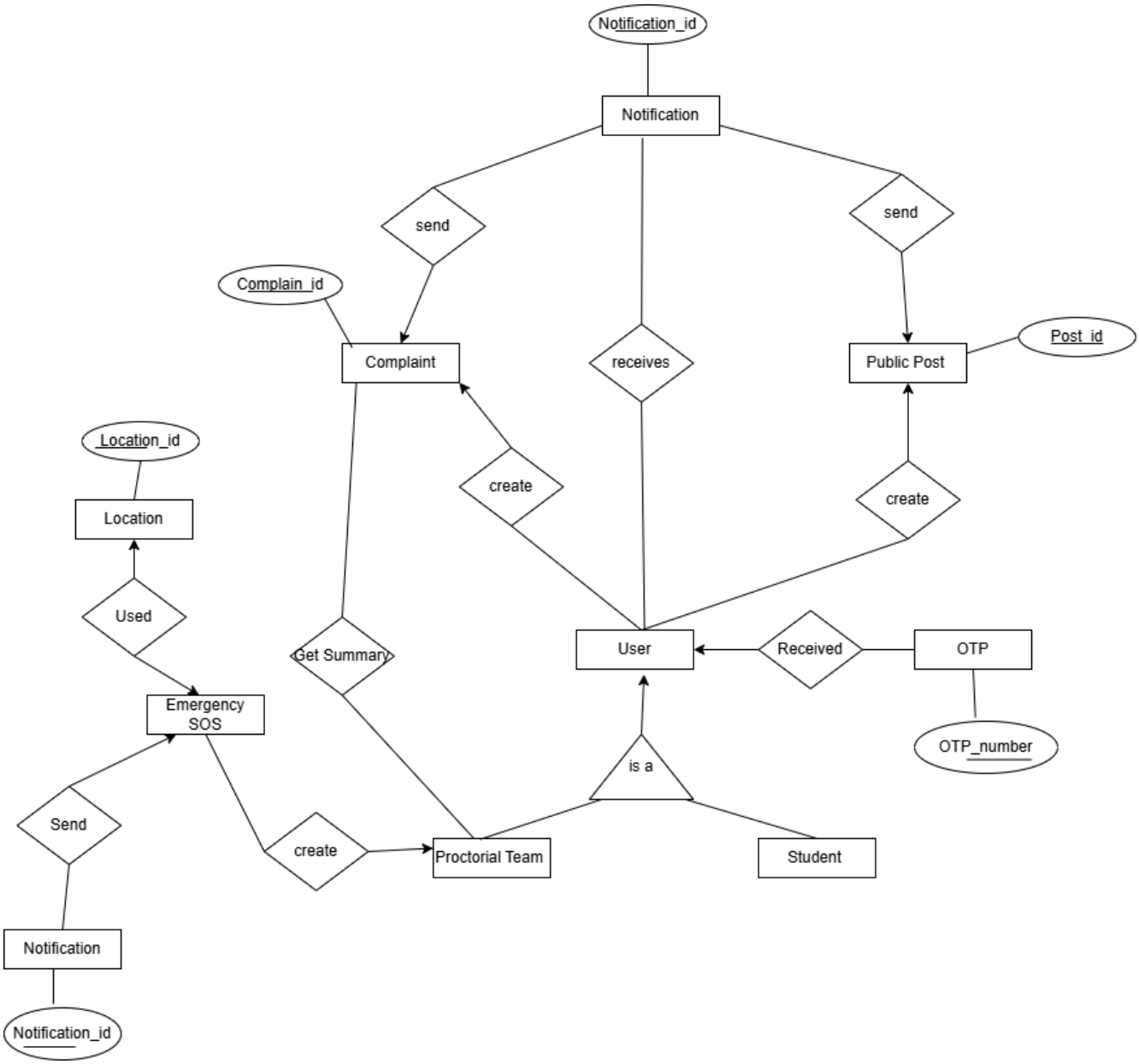
7.3 Selected Data Object:

Serial no.	Data objects	Attributes
1	User	User_id, name , email, phone, password, role
2	Student	Registration no.,Department name, Hall name,Session
3	Proctorial Team	Designation, Area
4	Complaint	Complaint_id, description, timestamp, Media
5	Location	Location_id, latitude, longitude, area_name
6	notification	Notification_id, message, status, timestamp
7	Public Post	Post_id, category, priority, anonymous, description, status, timestamp, Media
8	Emergency SOS	SOS_id, response_time, action_status, timestamp
9	OTP	OTP_number, time, date
10	Proctor_ *Complaint*	Complain id, status

7.4 Relation :


User
Create
Complaint
User
Recieved
Notification
User
create
Public Post
Proctorial
create
Emergency SOS
User
receives
OTP
Complaint
sends
Notification
Location
used
Emergency SOS
Public Alert
send
Notification
Emergency SOS
send
Notification

7.5 ERD :


Notification_id
Notification
send
send
Complain_id
Complaint
receives
Public Post
Post_id
Location_id
Location
create
create
Used
Get Summary
User
Received
OTP
Emergency SOS
OTP_number
is a
Send
create
Proctorial Team
Student
Notification
Notification_id

7.6 Echema Table :

No.	Entity Name	Arrtibute	Data Type	Description
1	User	user_id(PK)	INT	Unique identifier for each user
		Name	VARCHAR	FULL Name For the User
		email	VARCHAR	User's email address
		Phone	VARCHAR	User's phone number
		password	VARCHAR	Login Password
		role	ENUM('Student','Proctor','Admin')	Defines user type (Student, Proctorial team)
2	Student	User_id(FK)	INT	References User
		Registration No.(PK)	INT	User's Registration number
		Department name	VARCHAR	User's Department name
		Hall	VARCHAR	User's Hall name
		Session	INT	User's Session

3	Proctorial Team	Designation	VARCHAR	User's Designation name
		Area	VATCHAR	User's Area name
4	Complaint	complaint_id(pk)	INT	Complaint ID NUmber
		user_id(FK)	INT	User id Number
		location_id	INT	User Location
		Description	TEXT	About Complaint
		status	ENUM('received','in progress','resolved')	Complaint status
		timestamp	DATETIME	Date and time when the compliant was created
		category	VARCHAR	Complaint Category
5	Location	location_id(PK)	INT	Unique identifier for each location
		latitude	DECIMAL	Latitude coordinate of the location
		longitude	DECIMAL	Longitude coordinate of the location
		area_name	VARCHAR	Name of the area or place
6	Notification	notification_id(PK)	INT	Unique identifier for each notification
		user_id(FK)	INT	References the user who receives the notification
		alert_id(FK)	INT	References the related public alert

		sos_id(FK)	INT	References the related emergency SOS
		message	TEXT	Notification message content
		status	ENUM('sent','read')	Read/Unread or delivery status
		timestamp	DATETIME	Date and time when the notification was created
7	Public post	post_id(PK)	INT	Unique identifier for each public post
		user_id(FK)	INT	References the user who created the post
		category	VARCHAR	Type of alert (e.g., theft, warning, danger)
		priority	ENUM('Low','Medium','High')	Severity level of the alert
		anonymous	BOOLEAN	Indicates whether the post is anonymous (Yes/No)
		description	TEXT	Detailed description of the alert
		status	TEXT	Approval status (Pending, Approved, Rejected)
		timestamp	DATETIME	Date and Time when the post was created
8	Emergency SOS	sos_id(PK)	INT	Unique identifier for each SOS request
		user_id(FK)	INT	References the user who triggered the SOS
		location_id(FK)	INT	References the location of the emergency

		status	ENUM('Pending','Responded')	Current SOS status (Sent, In Progress, Resolved)
		timestamp	DATETIME	Date and time when the SOS was triggered
		response_time	DATETIME	Time taken to respond to the SOS
		action_taken	TEXT	Action Performed By which User
9	OTP	otp_no(PK)	INT	Unique OTP Number
		user_id(FK)	INT	References the user associated with the OTP
		date	DATETIME	Date when the OTP was generated
		time	TIMESTAMP	Time when the OTP was generated
10	User_Notification	User_id(FK)	INT	References the user who receives the notifiction
		notification_id(FK)	INT	References the notification who receives the notification
11	Proctor_complaint	User_id(FK)	INT	References the user who files or is associated with the complaint
		complaint_id(Fk)	INT	References the specific complaint submitted by the user

8. Class Based Modeling

Class-based modeling defines the structure of the entire system by identifying the static structure of objects in that system. A class model defines attributes and operations for the objects of each class and also the relationship between the objects, and the collaborations that occur between the classes of the systems. The elements of a class-based model include classes and objects, attributes, operations, Class-Responsibility-Collaborator (CRC) models, collaboration diagrams, and packages.

8.1 General Characteristics

Candidate classes are categorized based on the seven general classifications. The analysis classes manifest themselves in one of the following ways:

1. External Entities
2. Things
3. Occurrence/Events
4. Organizational Unit
5. Role
6. Places
7. Structure

Serial	Noun	P/S	General Classification
1.	DU Alert	S	4
2.	system	S	2
3.	application	S	2
4.	platform	S	
5.	student	S	4,5,7
6.	university	S	6
7.	campus	S	6
8.	emergency	S	
9.	response	S	
10.	goal	S	
11.	update	S	
12.	community	S	
13.	Registration No.	S	2
14.	process	S	
15.	user	P	4,5,7
16.	form	S	
17.	name	P	2
18.	department	P	2
19.	Roll number	P	2
20.	Email Generator	P	1
21.	address	P	6
22.	Phone number	P	2
23.	list	S	
24.	suggestion	S	
25.	OTP	S	1,2,3,7
26.	password	S	2
27.	identity	P	
28.	login	S	
29.	account	S	
30.	dashboard	S	
31.	button	S	
32.	danger	P	
33.	harassment	P	
34.	assault	P	

Serial	Noun	P/S	General Classification
35.	location	S	
36.	GPS	S	1
37.	service	S	
38.	alert	S	
39.	proctor	S	4,5,7
40.	Proctorial team	S	4,5,7
41.	incident	S	2
42.	report	S	2
43.	complaint	S	2
44.	category	S	2
45.	theft	S	5
46.	fraud	S	5
47.	activity	S	
48.	description	S	2
49.	image	S	2
50.	video	S	2
51.	evidence	S	2
52.	database	S	2
53.	status	S	2
54.	timeline	S	2
55.	administrator	S	4,5,7
56.	admin	S	4,5,7
57.	approval	S	
58.	Push Notification	S	1
59.	anonymity	S	
60.	analytics	S	2
61.	statistics	S	2
62.	Summary Generator	S	1
63.	investigation	P	
64.	case	P	
65.	safety	P	
66.	policy	P	

Probable Classes:

1. User
2. Student
3. Proctor
4. Proctorial Team
5. OTP
6. Administrator
7. Email Generator
8. Summary Generator
9. Push Notification
10. LocationAPI

8.2 Selection Criteria:

1. Retained Information
2. Needed Service
3. Multiple Attributes
4. Common Operation
5. Common Attributes
6. Essential Requirements

Serial No.	Classes	Selection Criteria
1.	User	1,3,5,6
2.	Student	1,3,5,6
3.	Proctor	1,3,5,6
4.	Proctorial Team	1,3,5,6
5.	OTP	6
6.	Administrator	1,3,6
7.	Email Generator	6
8.	Summary Generator	6
9.	Push Notification	6
10.	Location API	6

Analysis: We can remove “Student, Proctor, Proctorial Team” class because all attributes and responsibilities can be found in “User” class

8.3 List of verbs:

1. protect	2. improve
3. focus	4. help
5. allow	6. report
7. provide	8. share
9. begin	10. continue
11. install	12. open
13. show	14. enter
15. include	16. start
17. type	18. suggest
19. avoid	20. speed
21. fill	22. click
23. send	24. use
25. verify	26. belong
27. confirm	28. move
29. create	30. become
31. store	32. ensure
33. view	34. misuse
35. log in	36. forget
37. recover	38. regain
39. prevent	40. access
41. design	42. place
43. face	44. press
45. collect	46. identify
47. reach	48. require
49. choose	50. classify
51. respond	52. add
53. know	54. mark
55. explain	56. describe
57. happen	58. involve
59. upload	60. understand
61. submit	62. update
63. track	64. warn
65. appear	66. review
67. check	68. hide
69. reveal	70. increase
71. post	72. stay
73. inform	74. remain
75. build	76. manage
77. monitor	78. assign
79. generate	80. save
81. enable	82. support
83. connect	84. encourage
85. spread	86. transform

8.4 Selected Class:

1. **User**
   a. Attributes:
      - Name
      - Phone No
      - Email
      - Address
      - User Id
      - Password
      - Role

   b. Methods:

      - register()
      - login()
      - recoverPassword()
      - updateProfile()
      - receiveNotification()
      - createEmergencySOS()
      - createPost()
      - createComplaint()

2. **OTP**
   a. Attribute
      - OTP_id
      - Phone Number
      - Email Address
      - Time
      - Date

   b. Methods
      - generateOTP()
      - sendOTP()
      - verifyOTP()
      - expireOTP()

3. **Administrator**

   a. Attribute
      - User id
      - Password
   b. Methods
      - verifyUser()

- removeUser()
- approvePost()
- rejectPost()
- complaintStatusUpdate()
- createStatisticalAnalysis()
- archivedComplaint()

4. Email Generator

a. Attribute
- Email Address
- Date
- Time

b. Methods
- sendAccountEmail()
- sendNotificationEmail()
- sendComplaintEmail()

5. Summary Generator

a. Attribute
- Complaint Description

b. Methods
- generateSummary()
- saveSummary()

6. Push Notification

a. Attribute
- Message
- Date
- Time
- User id
- Location id

b. Methods
- generateNotificationId()
- sendPushNotification()
- findUser()

7. Location API

a. Attribute
- Date

- Time
- User id

b. Methods
   - traceLocation()
   - generateLocationId()
   - sendLocationDetails()

8.5 CRC (Class-Responsibility-Collaborator) Models:

1. **User**

Attributes	**Methods**
Name Phone No Email Address User Id Password Role	register() login() recoverPassword() updateProfile() receiveNotification() createEmergencySOS() createPost() createComplaint()
Responsibilities	**Collaborator**
Register and log in with role-specific details	Administrator
Recover password using OTP	OTP
Request for Public Post	Administrator
Emergency SOS send	Push Notification
Create Complaint	Administrator

2. **OTP**

Attributes	Methods
OTP_id Phone Number Email Address Time Date	generateOTP() sendOTP() verifyOTP() expireOTP()
Responsibilities	**Collaborator**
Generate and Send OTP for registration and recovery	User
Verify OTP for authentication	Administrator

3. **Administrator**

Attributes	Methods
User id Password	verifyUser() removeUser() approvePost() rejectPost() complaintStatusUpdate() createStatisticalAnalysis() archivedComplaint()
Responsibilities	**Collaborator**
Verify user information & remove user	User
Approve/Reject Post request	User
Complaint status update, statistical analysis & archived resolved complaints	User

4. **Email Generator**

Attributes	**Methods**
Email Address Date Time	sendAccountEmail() sendNotificationEmail() sendComplaintEmail()
Responsibilities	**Collaborator**
Send account related emails	User, OTP
Send post rejection emails	User, Administrator
Send Complaint related emails	User, Administrator

5. **Summary Generator**

Attributes	**Methods**
Complaint Description	generateSummary() saveSummary()
Responsibilities	**Collaborator**
Generate summary of complaint description	User

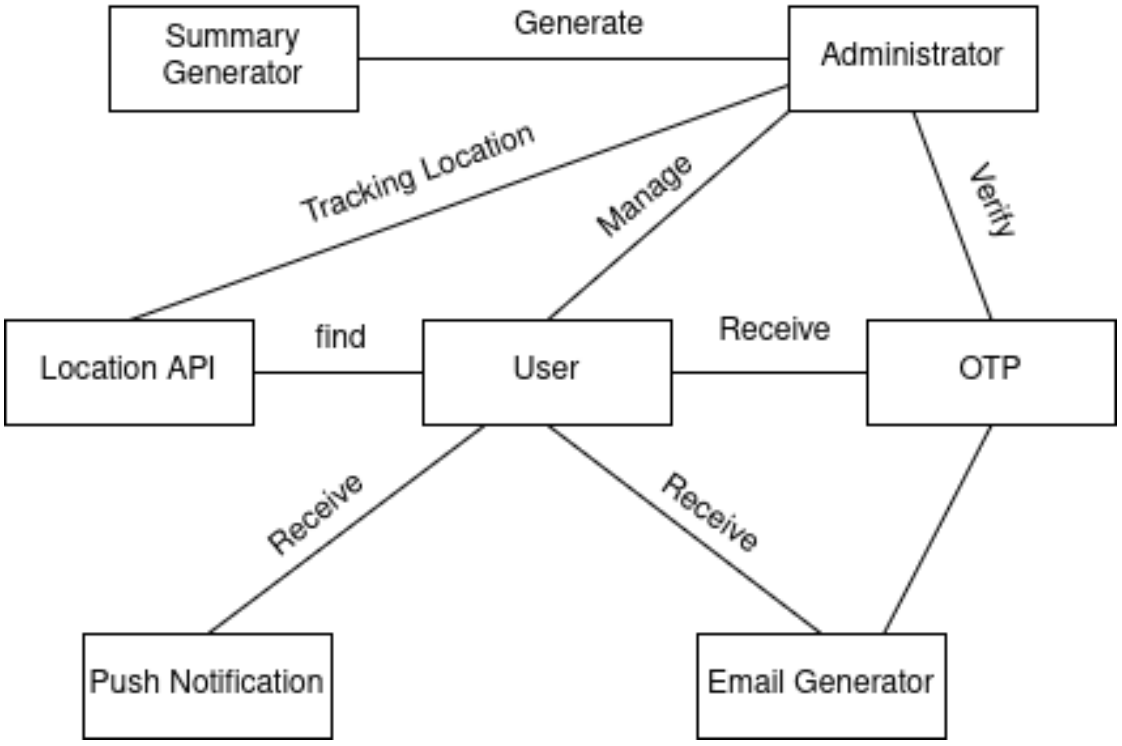
6. **Push Notification**

Attributes	**Methods**
Message Date Time User id Location id	sendPushNotification() generateNotificationId() findUser()
Responsibilities	**Collaborator**
Send push notifications	User
Find users	Location API

7. **Location API**

Attributes	**Methods**
Date Time User id	traceLocation() generateLocationId() sendLocationDetails()
Responsibilities	**Collaborator**
Trace accurate location	User
Send the location details	User, Administrator

8.6 CRC Diagram:



9. Behavioral Modeling

9.1 Introduction:

The behavioral model predicts how the software will react to events or stimuli outside its control.
Two distinct characterizations of states must be taken into account while modeling behavior: (1) the state of each class as the system executes its function, and (2) the state of the system as seen from the outside as the system executes its function.

9.2 State Transition:

List of event :

No	Initiator	Event	Event Name	Collaborator
01	User	Enter name, department, registration no, email, phone	User Registration	OTP
02	User	Enter username and password	Create Login Credentials	Administrator
03	User	Enter username and password	User Login	Administrator
04	User	Enter registration number for recovery	Password Recovery Started	OTP
05	OTP	Generate and send OTP	OTP Sent	User
06	OTP	Verify OTP code	OTP Verified	User
07	User	Enter new password after OTP verification	Password Reset	Administrator

08	User	Press Emergency SOS button	Emergency SOS Created	Push Notification, Location API
09	Location API	Collect real-time GPS location	Location Traced	User, Administrator
10	Location API	Send location with user details to proctor	Location Details Sent	Administrator, Push Notification
11	User	Select category, enter complainant details	Complaint Created	Administrator
12	User	Enter accused information	Accused Details Added	Administrator
13	User	Write incident description	Incident Description Added	Administrator
14	User	Upload images/videos	Media Evidence Uploaded	Administrator
15	User	Submit complaint form	Complaint Submitted	Administrator
16	Administrator	Receive complaint and set status "Received"	Complaint Status Updated	User, Email Generator
17	Administrator	Update complaint status to "In Progress"	Complaint In Progress	User, Email Generator
18	Administrator	Update complaint status to "Resolved"	Complaint Resolved	User, Email Generator
19	User	View complaint timeline	Complaint Tracking Viewed	Administrator
20	User	Select category for public alert	Public Post Created	Administrator

21	User	Enter details and submit public post	Public Post Submitted	Administrator
22	User	Choose anonymous mode for public alert	Anonymous Post Created	Administrator
23	Administrator	Approve public alert	Post Approved	User, Push Notification
24	Administrator	Reject public alert	Post Rejected	User, Email Generator
25	Push Notification	Send notification to all users	Push Notification Sent	User
26	User (Proctor)	View emergency alerts	Emergency Alert Viewed	User, Location API
27	User (Proctor)	View dashboard statistics	Dashboard Statistics Viewed	Administrator
28	User (Proctor)	Filter and sort complaints	Complaints Filtered	Administrator
29	User (Proctor)	Open complaint details	Complaint Details Viewed	Administrator
30	User (Proctor)	Request AI summary of complaints	Summary Generation Requested	Summary Generator
31	Summary Generator	Generate complaint summary	Summary Generated	User
32	Summary Generator	Save generated summary	Summary Saved	User
33	Administrator	View monthly complaint statistics	Analytics Viewed	None

34	Administrator	Generate system performance report	Statistical Analysis Created	None
35	Administrator	Verify and approve user registration	User Verified	User, Email Generator
36	Administrator	Remove user from system	User Removed	User
37	Administrator	Archive resolved complaints	Complaint Archived	None
38	Email Generator	Send account creation confirmation	Account Email Sent	User
39	Email Generator	Send notification emails	Notification Email Sent	User
40	Email Generator	Send complaint update emails	Complaint Email Sent	User
41	User	Update personal profile information	Profile Updated	Administrator
42	Push Notification	Find users in affected location	Users Found	Location API
43	Push Notification	Generate notification ID	Notification ID Generated	User
44	System	Auto-store submitted complaint in database	Complaint Stored	Administrator
45	User	Log out from system	User Logged Out	Administrator

9.2.1 User:

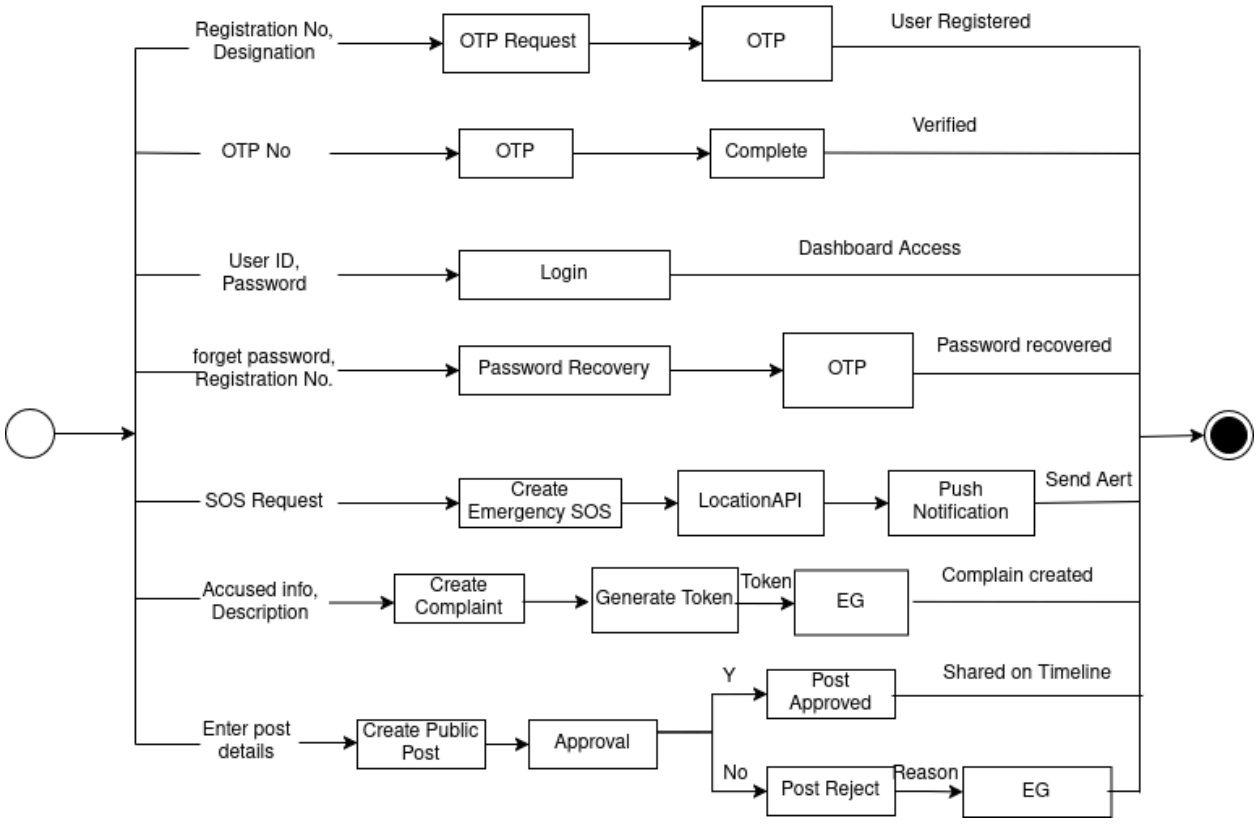


Figure: State Transition of User`s

9.2.2 OTP:

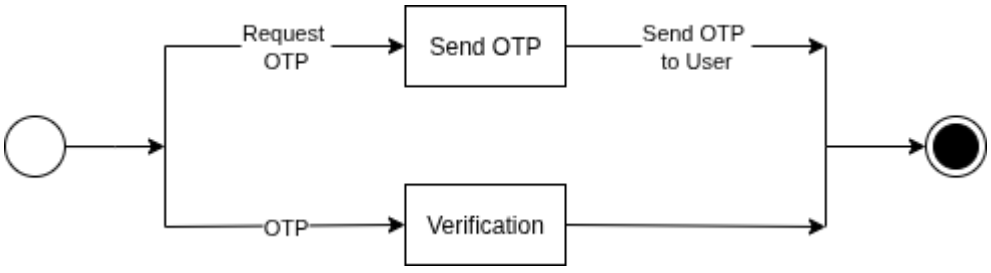


Figure: State Transition of OTP

9.2.3 Administrator :

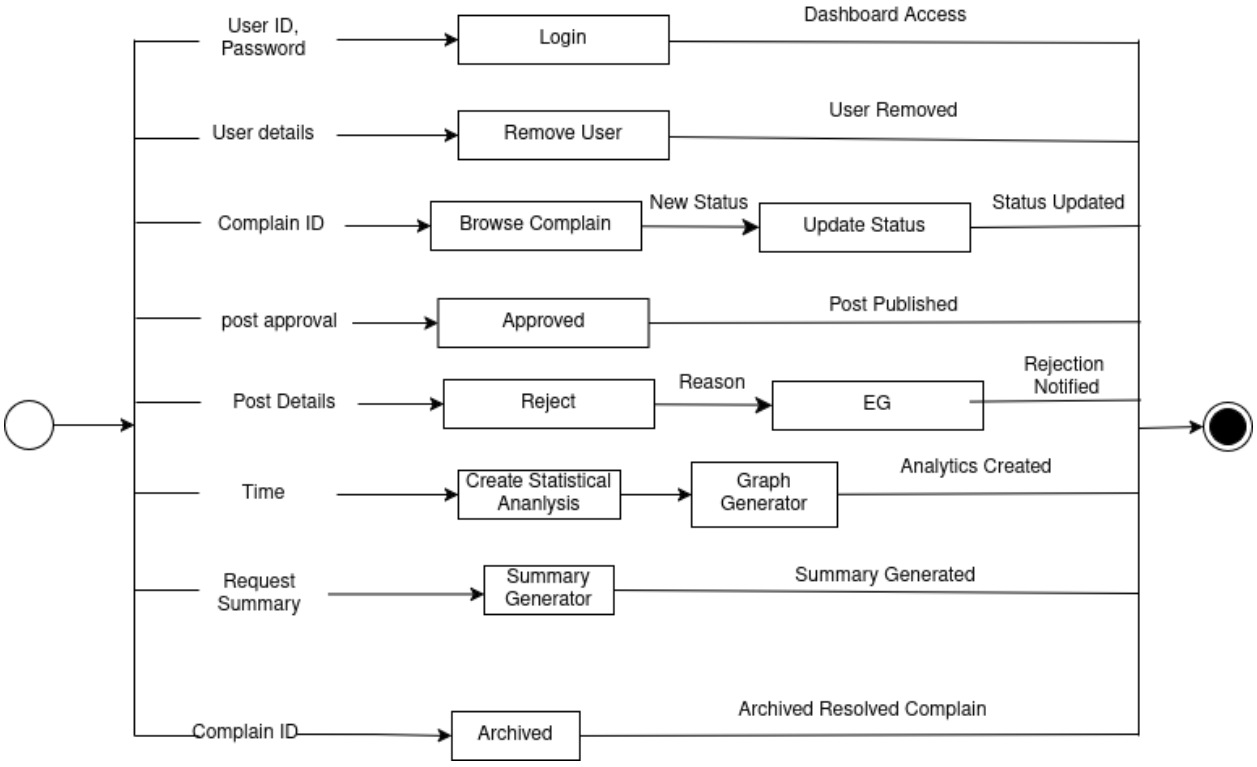


Figure: State Transition of administrator

9.2.4 Location:

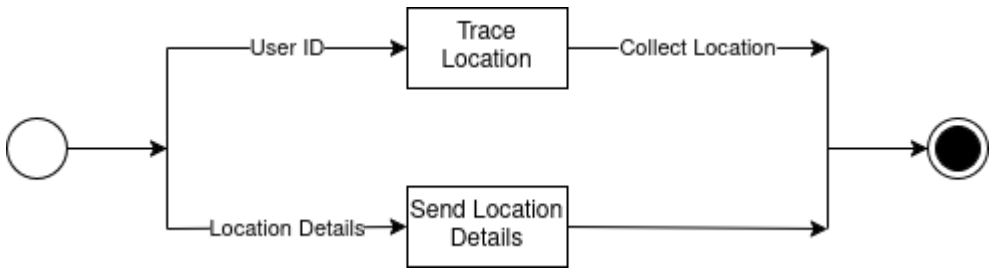


Figure: State Transition of Location

9.2.5 Push Notification:

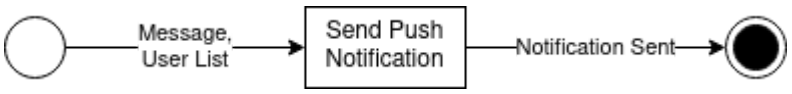


Figure: State Transition of Push Notification

9.2.6 Summary Generator:

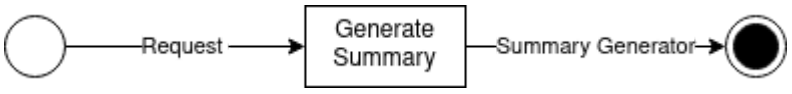


Figure: State Transition of Summary Generator

9.2.7 Email Generator:

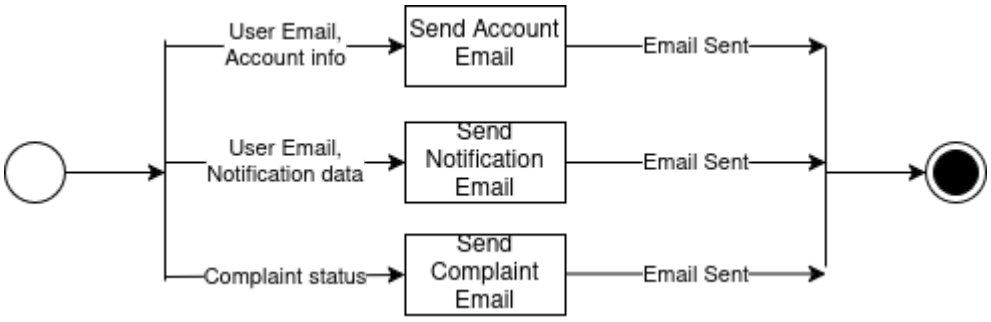


Figure: State Transition of Email Generator

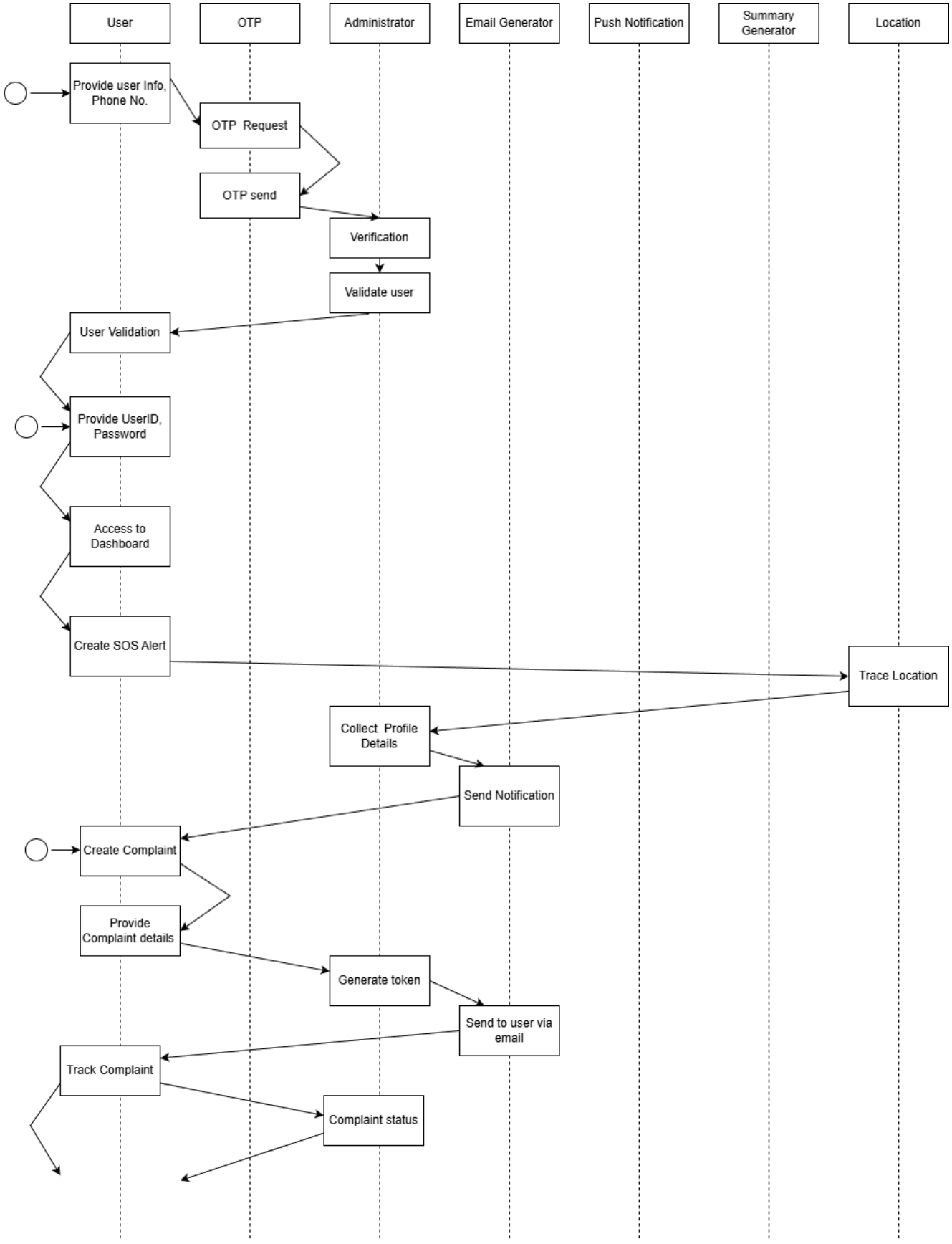
9.3 Sequence Diagram

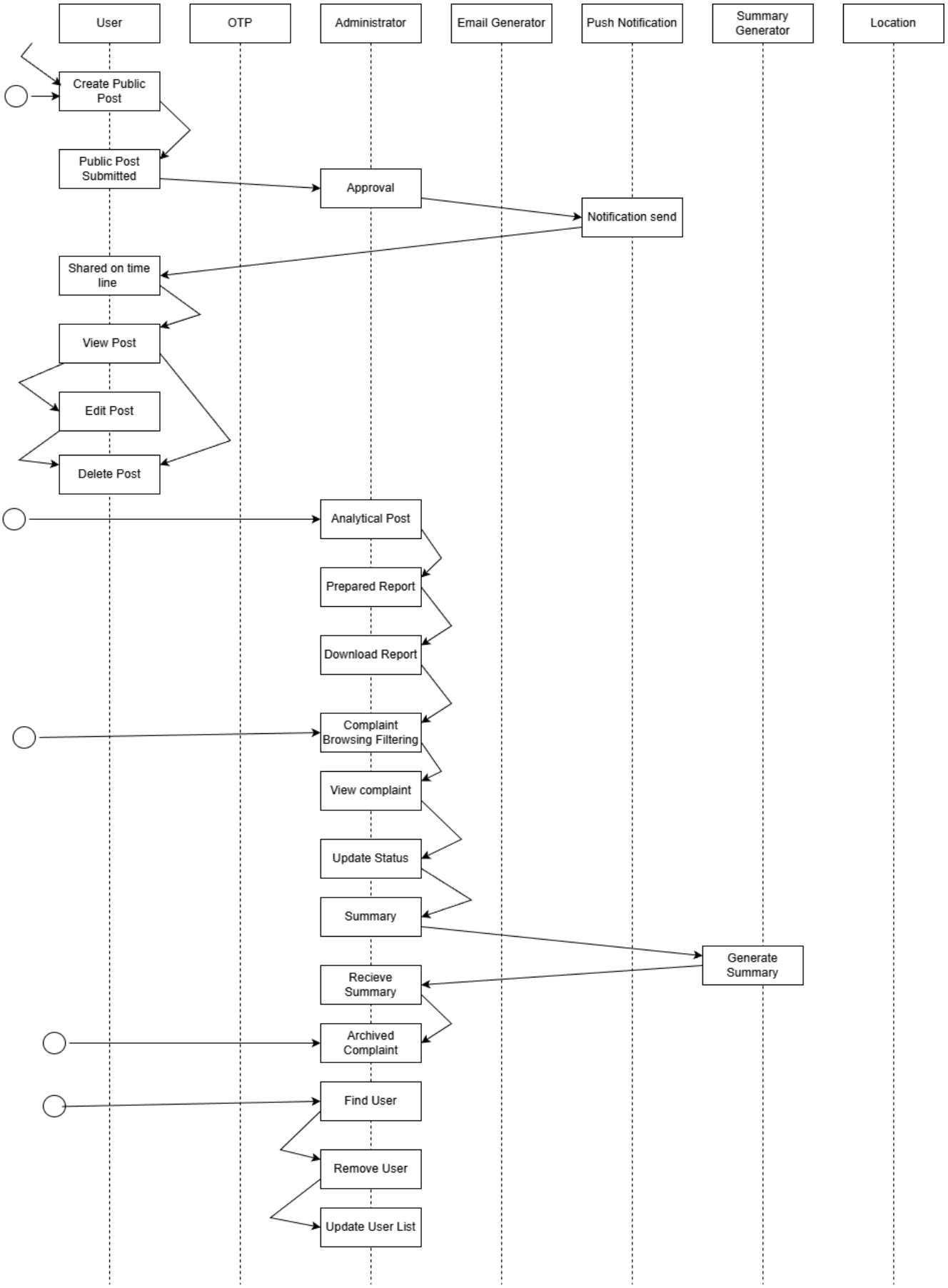
The second type of behavioral representation, called a sequence diagram in UML, represents how events cause flow from one object to another as a function of time. The sequence diagram is a shorthand version of the use case. It represents vital classes and the events that cause behavior to flow from class to class.

9.3.1 Purpose of Sequence Diagram

● High-level interaction between active objects in a system
● The interaction between object instances within a collaboration that realizes a use case
● The interaction between objects within a collaboration that realizes an operation
● Either model generic interactions (showing all possible paths through the interaction) or specific instances of an interaction (showing just one path through the interaction)

9.3.2 Sequence Diagram of LMS




User
OTP
Administrator
Email Generator
Push Notification
Summary Generator
Location
Create Public Post
Public Post Submitted
Approval
Notification send
Shared on time line
View Post
Edit Post
Delete Post
Analytical Post
Prepared Report
Download Report
Complaint Browsing Filtering
View complaint
Update Status
Summary
Generate Summary
Recieve Summary
Archived Complaint
Find User
Remove User
Update User List