SOFTWARE ENGINEERING AND PROJECT MANAGEMENT

LAB FILE

NAME – SAGNIK ROY
SAP ID – 500109927

ENROLLMENT NO. – R2142221403

COURSE – B.Tech CSE(Big Data Spz)
BATCH – 2(Non honors)

Sagnik Roy
500109927

INDEX

Sr.no.	Experiment	Date	Signature
1	USE CASE DIAGRAM	16.01.24	16/1
2	CLASS DIAGRAM	23.01.24	23/1
3	DATA FLOW DIAGRAM	30.01.24	30/1
4	SOFTWARE REQUIREMENT SPECIFICATION	09.02.24	9/2
5	ACTIVITY DIAGRAM	15.02.24	15/2
6	SEPM RECAP	20.02.24	20/2
7	SEQUENCE DIAGRAM	11.03.24	11/3
8	STATECHART DIAGRAM	21.03.24	21/3
9	COLLABORATION DIGRAM	09.04.24	9/4
10	DEPLOYMENT DIAGRAM	09.04.24	9/4

Use Cases

for

Mobile Location Alarm

Prepared by Sagnik Roy

16.01.2024

Revision History

Name	Date	Reason For Changes	Version

1. Use Case Identification

1.1. Use Case ID

Give each use case a unique numeric identifier, in hierarchical form: X.Y. Related use cases can be grouped in the hierarchy. Functional requirements can be traced back to a labeled use case.

1.2. Use Case Name

- View part number information.
- Manually mark hypertext source and establish link to target.
- Place an order for a CD with the updated software version.

1.3. Use Case History

1.3.1 Created By

Sagnik Roy initially documented this use case

1.3.2 Date Created

The date of the creation when it is initially documented is 16.01.2024.

1.3.3 Last Updated By

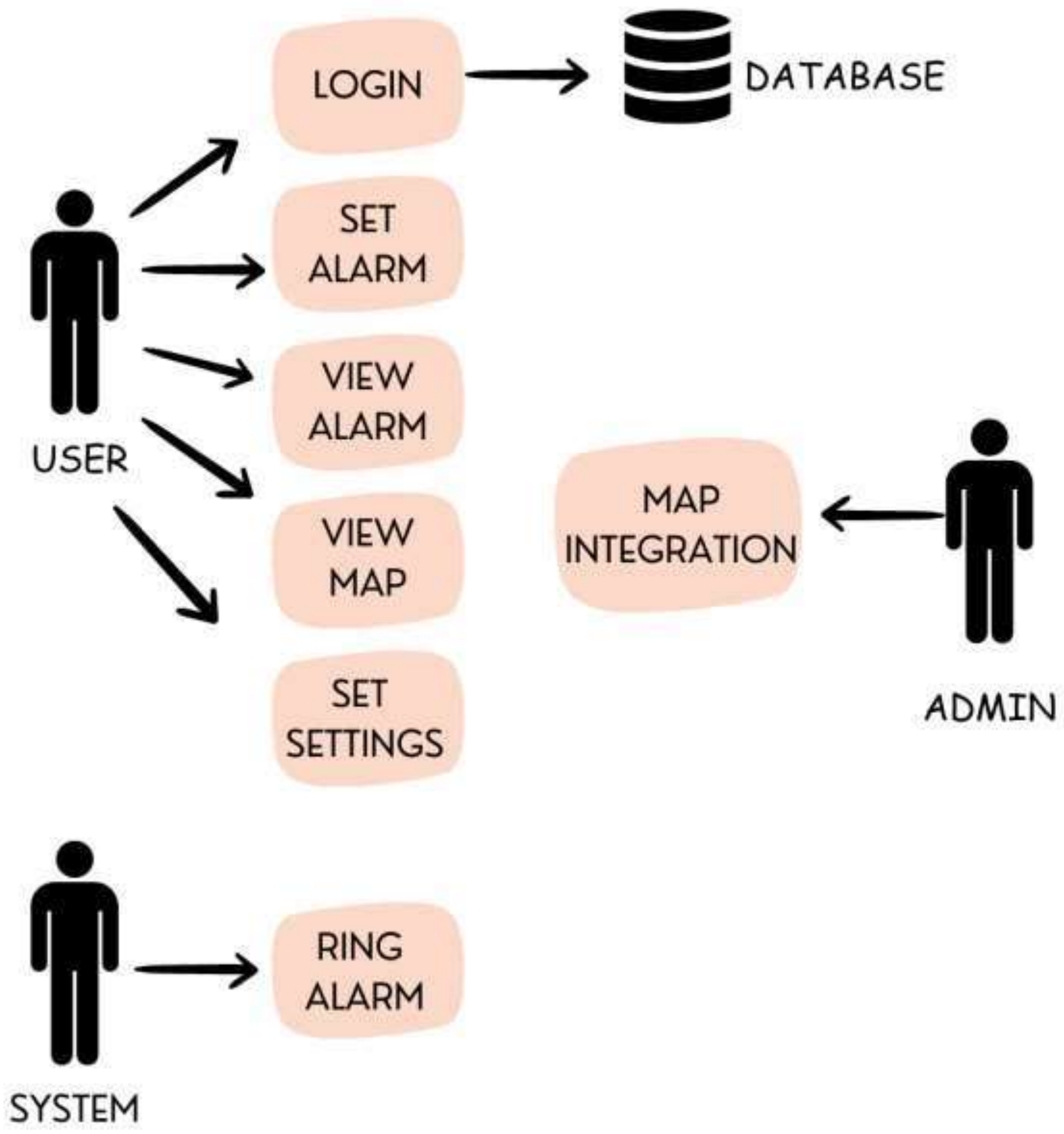
Sagnik Roy performed the most recent update to the use case des

1.3.4 Date Last Updated

This use case was recently updated on 20.01.2024

2. Use Case Definition

2.1. Use Case Diagram



2.2. Actors

- USERS
- ADMIN
- SYSTEM

2.3. Description

The Mobile Location Alarm System is an innovative and practical project designed to enhance user experience by providing a personalized and location-aware alarm solution. This project leverages the capabilities of mobile devices to offer users a dynamic alarm system that triggers based on their specific location, ensuring timely and contextually relevant reminders.

2.4. Preconditions

1. The user needs to be authenticated and logged into the mobile application.
2. The mobile device must have GPS or location services enabled.
3. The mobile device must have an active and stable internet connection.
4. For use cases like editing, deleting, or enabling/disabling alarms, there must be at least one previously set alarm.
5. User preferences for alarm sounds and vibration settings should be configured.
6. The user should be actively using the application or have the application running in the background.

2.5. Postconditions

1. **Set Location-Based Alarm:** A new location-based alarm is successfully added to the system.
2. **Edit Location-Based Alarm:** The details of the existing location-based alarm are successfully updated.
3. **Delete Location Based Alarm:** The specified location-based alarm is successfully removed from the system.
4. **View Alarms:** The user can view a list of all active location-based alarms in the system.
5. **Notification Handling:** A notification is successfully sent to the user's device when a location-based alarm is triggered.
6. **Alarm Sound/Vibration Settings Updated:** Changes to alarm sound and vibration settings are successfully applied.

2.6. Priority

The highest priority is assigned to the Login and Registration Use Case, as it serves as the primary mechanism for user authentication and session management. This functionality is crucial for establishing the identity of users, ensuring that only authorized individuals can access the system. The Login and Registration Use Case plays a pivotal role in assisting administrators in identifying legitimate users, thereby enhancing security and accountability within the Mobile Location Alarm System.

2.7. Frequency of Use

The system allows users an unlimited number of login attempts until the entered username and password credentials successfully match those stored in the database. Once a match is verified, the user gains access to utilize the features of the application.

2.8. Normal Course of Events

In the typical course of events for a Mobile Location Alarm System, the user logs in, navigates to the alarm-setting section, and sets a location-based alarm by specifying a geofenced area along with customized preferences. The application continuously monitors the user's location in the background and triggers the alarm when the predefined area is entered or exited. A notification is sent to the user's device, prompting them to acknowledge and take appropriate action. Users can view, edit, or delete alarms as needed. The system allows for seamless management of location-based alerts, enhancing user awareness and responsiveness to specific geographic contexts.

3. Alternative Courses

3.1.1 Invalid User

- If the validation of the user does not succeed, the use case concludes with a failure condition.

3.1.2 Offline Mode

- If the mobile phone lacks an internet connection, it ceases to receive GPS updates, rendering the application ineffective.

3.2. Special Requirements

- **Real-Time Updates:** The application must provide real-time updates on the user's location to ensure the timely triggering of location-based alarms.
- **Internet Connectivity:** The application requires Internet connectivity for triggering the alarm based on the GPS positioning system.

3.3. Notes and Issues

- Running the application in the background can cause battery degradation.
- If the mobile phone lacks an internet connection, it ceases to receive GPS updates, rendering the application ineffective.

Use Case Template

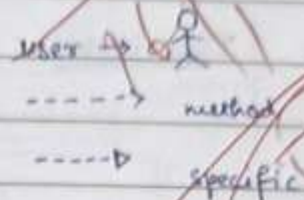
Use Case ID:			
Use Case Name:			
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actor:	
Description:	
Preconditions:	
Postconditions:	
Priority:	
Frequency of Use:	
Normal Course of Events:	
Alternative Courses:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

16/01/24

Software Engineering

[SAGNIK ROY]
(500109927)



Q Who are the actors?

Soln 1) user

2) admin

3) system

Requirement Analysis :

① To retrieve the user's current geological coordinates

② Allow user to enter new alarm for a particular location

③ Allows user to put a reminder text with the alarm

④ When near the location, the alarm should ring and display any reminder text to the user

⑤ The user should be able to view the already set alarms and can edit/delete or update alarms

⑥ User should be able to see the map to find how far is the user from expected location.

⑦ User should be able to set settings for the applications

CLASS DIAGRAM

For

Mobile Location Alarm System

Prepared by Sagnik Roy

UPES Dehradun

23.01.2024

Step 1: - Keeping the right class

Good Classes: -

- LOGIN
- SET ALARM
- SEND ALARM
- LOCATION

Bad classes: -

- SETTINGS
- LOGOUT

Redundant Class: -if the classes express the same information the most descriptive name should be kept. For example, SET ALARM and SEND ALARM

Irrelevant Classes: -if a class has little or nothing to do with the problem, it should be eliminated. In the mobile location alarm system, the system shall get executed without providing any Settings.

Vague Classes class should be specific. Some tentative classes may have ill-defined boundaries or be too broad in scope. For example, the Settings class here is a vague class,

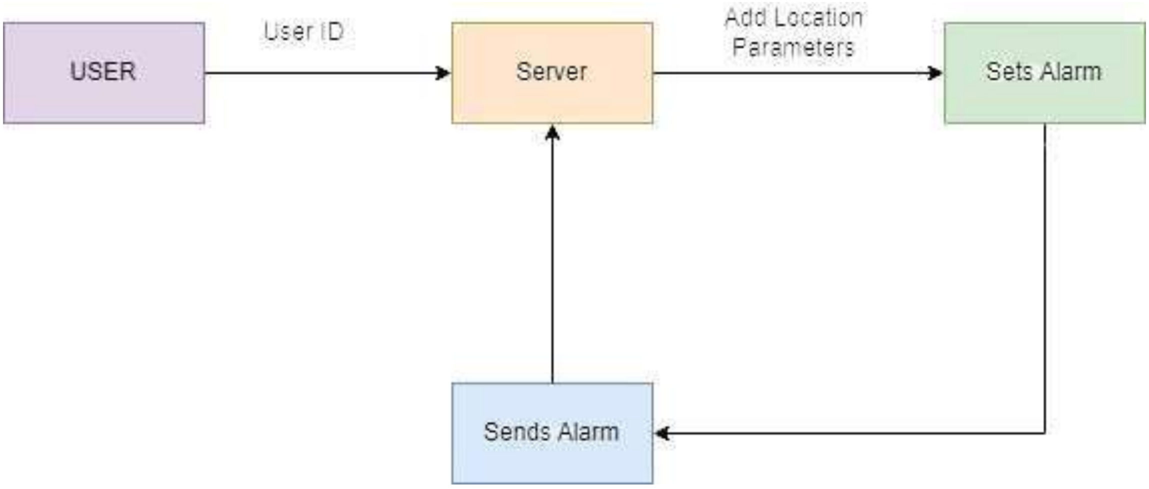
Attributes: - Names that primarily describe individual objects should be restarted as attributes. For example, login, and logout are usually attributes.

Step 2: - Keeping the right association

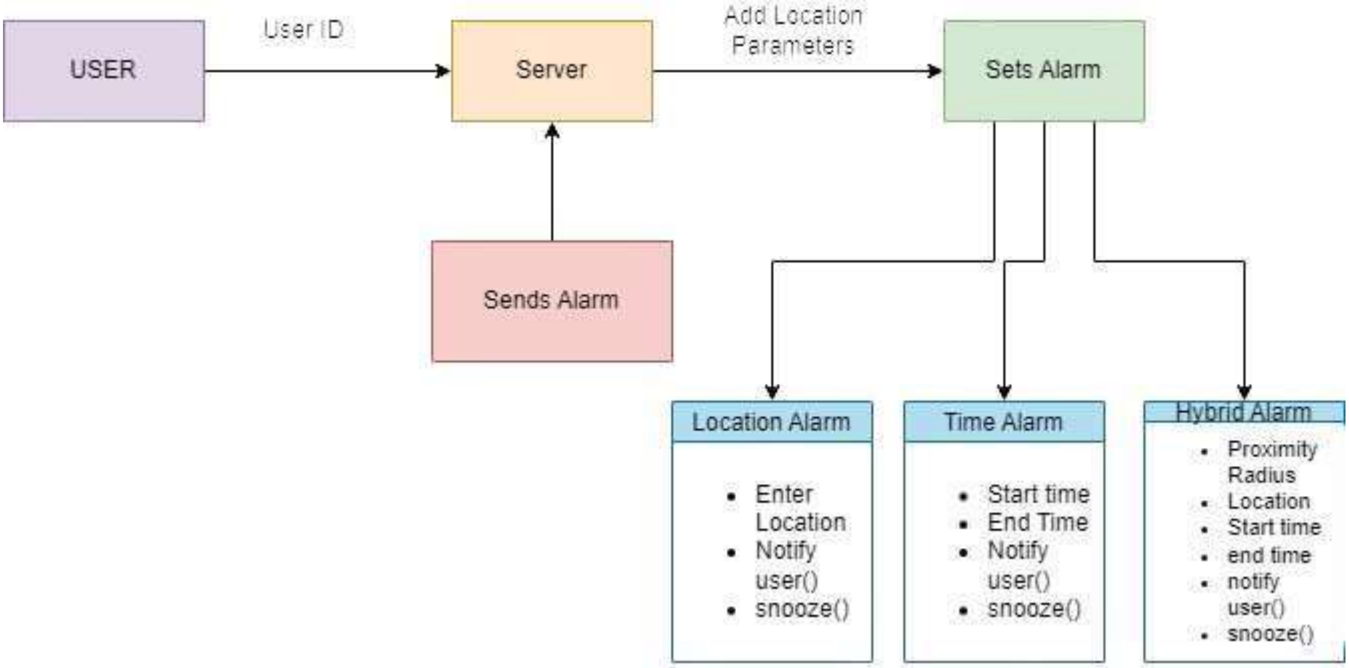
1. User enters the login credentials
2. The user enters the desired location
3. Based on the entered location the alarm shall ring upon reaching the proximity of the desired location
4. The user can check the alarms set and also manage them by modifying them

5. The system offers convenience to the user by giving the freedom to set notifications and alarm ringtones.

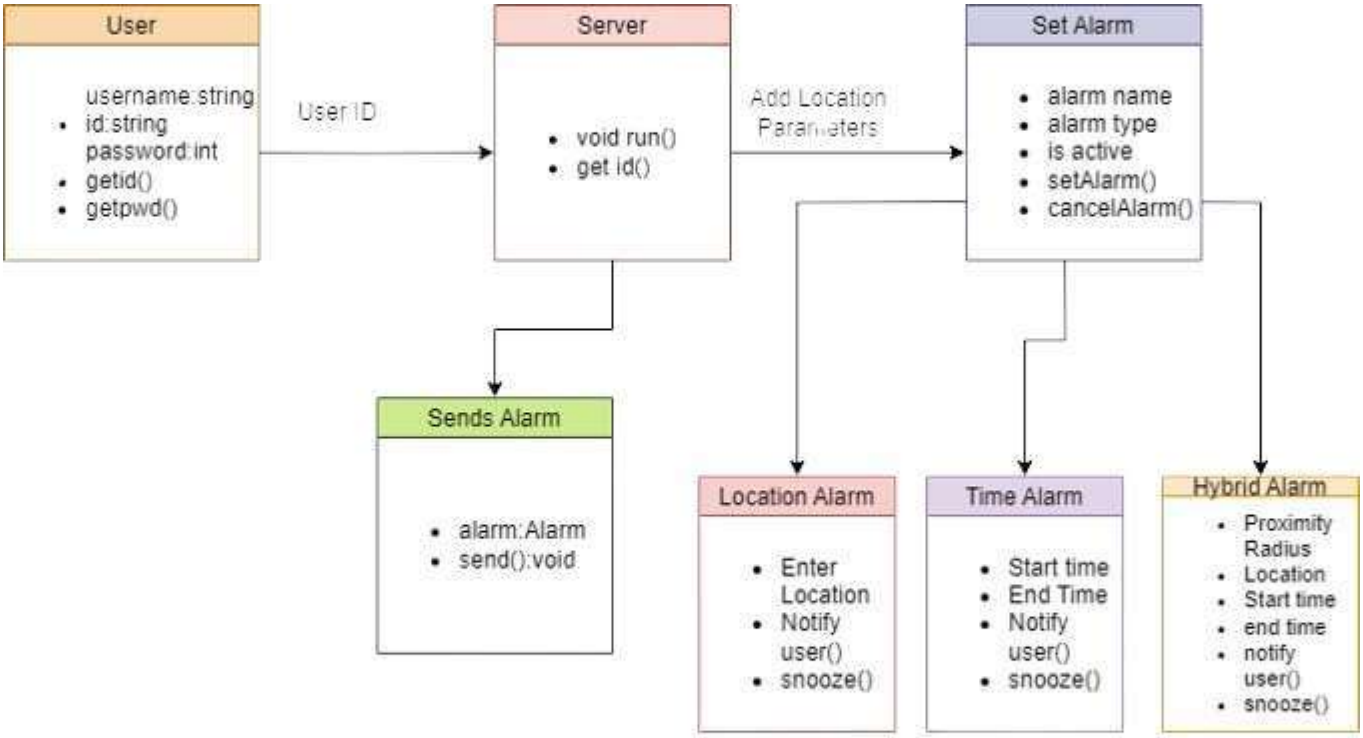
Step 3: - Initial diagram for Internet chatting



Step 4: - Refining with Inheritance



Step 5: - Class diagram


User
username:string
id:string
password:int
getid()
getpwd()
User ID
Server
void run()
get id()
Add Location
Parameters
Set Alarm
alarm name
alarm type
is active
setAlarm()
cancelAlarm()
Sends Alarm
alarm:Alarm
send():void
Location Alarm
Enter Location
Notify user()
snooze()
Time Alarm
Start time
End Time
Notify user()
snooze()
Hybrid Alarm
Proximity Radius
Location
Start time
end time
notify user()
snooze()

[SAGNIK ROY]
(500109927)

23/1/24

23/01/24

SEPM Lab-2

Class Diagram:

↳ Static structure of application / system

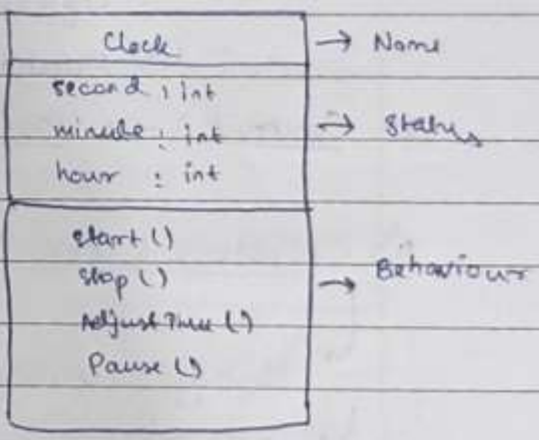
↳ Representation ---> JAVA (oops)

General	CD
* NAME	Name
* METHOD	Behaviour
* VARIABLE	Status

Example →

```
Class Clock {
        sec
        min
        hour
    }
      method
      {
        start();
        stop();
        adjust Time ();
        pause();
```

Class Diagram



Data Flow

For

Mobile Location Alarm

Prepared by Sagnik Roy

UPES, Dehradun

30.01.2024

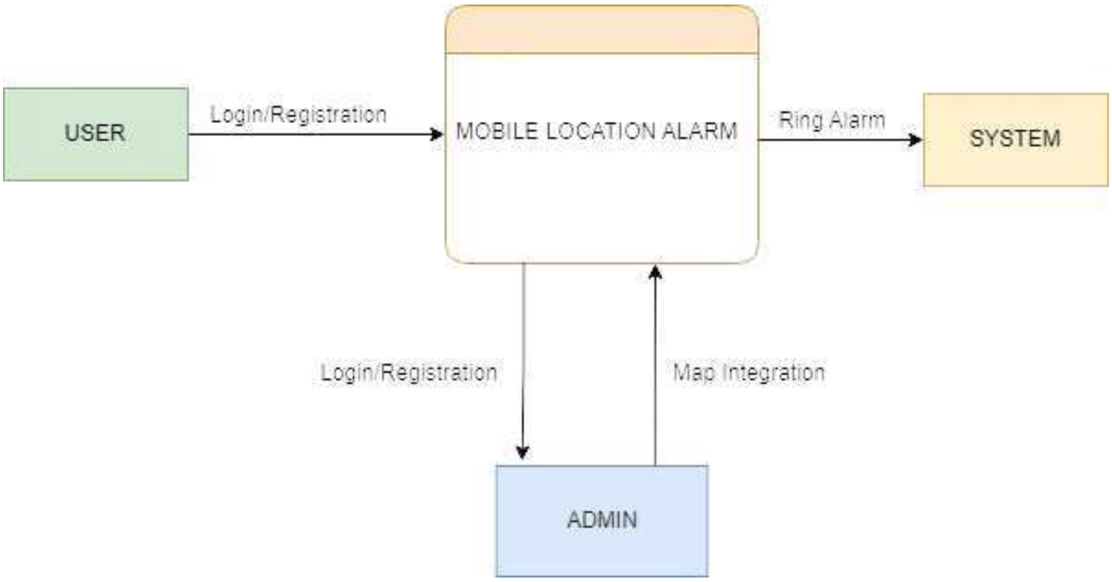
Revision History

Name	Date	Reason For Changes	Version

DFD DIAGRAM:

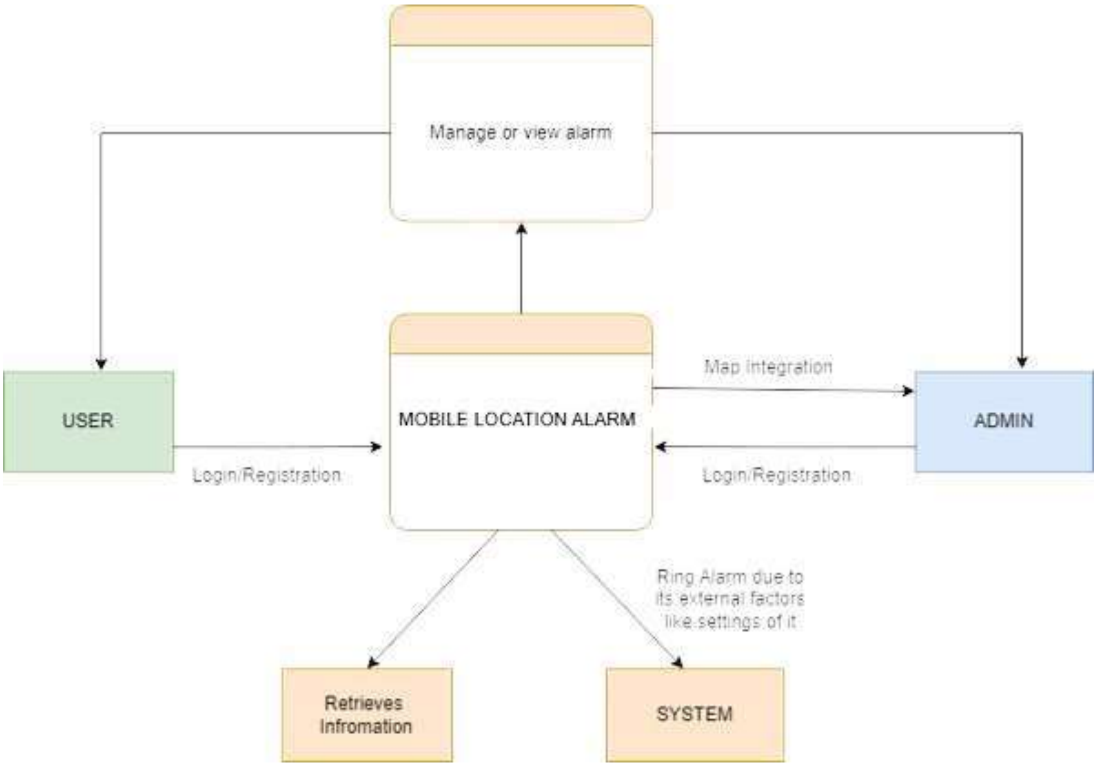
LEVEL 0: -

- In **level 0** the user requests login/registers to the system.
- The Server grants the request through a response service.
- The Mobile Location Alarm System works by leveraging the GPS (Global Positioning System) technology in smartphones or other GPS-enabled devices
- Admin establishes map integration.



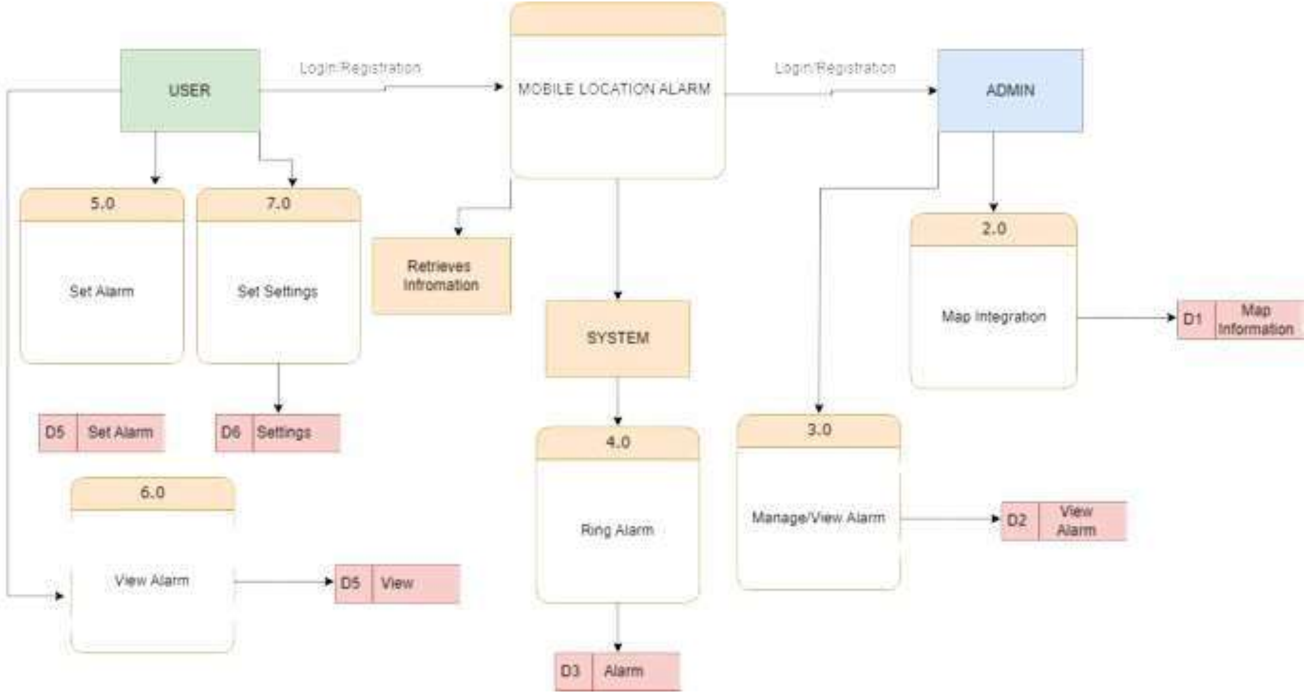
LEVEL 1: -

- In **Level 1** user must log in/register to use the system.
- The user can manage or view alarms set.
- The system retrieves the desired location and sets the alarm accordingly



LEVEL 2: -

- In Level 2 user can set settings where the user can select the alarm ringtone and notification settings.
- In Level 2, users can establish alarms, whether it be for singular or multiple destinations.



[SAGNIK ROY]
500109927
__/__/__

30/01/24 SEPM Lab-III 30/1/24

Data Flow Diagram (DFD)

It consists of four symbols-

- Entity → □
- Process → ▢
- Data Store → D1 | NAME
- Data Flow → $\xrightarrow{\text{NAME}}$

Topic → Mobile Location Alarm

For this topic -

Entity → 1) Mobile Device - User's Mobile Device

2) Location Service - Provides Location Service Information.

Process → 1) Alarm Triggering Process - The process monitors the location data and triggers the alarm when the certain criteria/ conditions are met.

2) Location Data Retrieval - Retrieves the current location data ~~and triggers~~ from the location service.

3) User Settings Management - Allows the user to set & manage alarm settings.

Software Requirements Specification

for

Mobile Location System

Version 1.0 approved

Prepared by Sagnik Roy

UPES

09.02.2024

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of the Mobile Location Alarm System is to provide users with a reliable and efficient tool for location-based alarms and notifications on their mobile devices. The system aims to enhance personal safety and security by allowing users to set up geographic boundaries (geofences) and receive timely alerts when they enter or exit these predefined areas.

1.2 Document Conventions

We use a bold font for the header. The size of the header font is 18. The size of the sub font is 16 and the size of the font is 14 for all the text documents. The description can be down in the points so that it is easy for read purposes. The font style is Times New Roman

1.3 Intended Audience and Reading Suggestions

This SRS is designed especially for developers, project managers, testers, and documentation writers. This document is intended for all individuals participating in and/or supervising the INTRANET CHATTING project. Readers interested in a brief overview of the product should focus on the rest of Part 1 (Introduction), as well as Part 2 of the document (Overall Description), which provide a brief overview of each aspect of the project. These readers may also be interested in Part 6 which lays out a concise timeline of the project.

1.4 Product Scope

The Mobile Location Alarm System aims to provide users with a comprehensive solution for enhancing personal safety and security through location-based alarms and notifications on their mobile devices. The key components and functionalities within the product scope include:
User Registration and Authentication:

- *Users can create accounts securely with unique identifiers.*
- *Authentication mechanisms ensure the protection of user data and account access.*

Location Tracking:

- *The system utilizes mobile device location services to track real-time geographical information.*
- *Continuous monitoring enables accurate positioning for effective geofencing.*

Geofencing:

- *Users can define geographical boundaries (geofences) on the map.*
- *The system triggers alarms when a user enters or exits the predefined geofenced areas.*

Alarm Configuration:

- *Customizable alarm settings allow users to define preferences for notifications.*
- *Alarm parameters include sound, vibration, and notification frequency.*

Notification Alerts:

- *Instant alerts are sent to users when they enter or exit specified geofences.*
- *Notifications include relevant information, such as the geofence name and location details.*

User Interface:

- *An intuitive and user-friendly mobile application interface facilitates easy interaction.*
- *Screens display real-time location information and enable seamless configuration of alarms.*

Privacy and Security:

- *Robust privacy measures are implemented to safeguard user location data.*
- *Security protocols ensure secure data transmission and storage.*

Compatibility:

- *The system is designed to run on major mobile platforms, ensuring compatibility with a wide range of devices.*

Documentation:

- *User manuals and technical documentation provide comprehensive guidance on system usage and implementation.*

1.5 References

- *www.chatgpt.com*
- *www.w3schools.com*

2. Overall Description

2.1 Product Perspective

The Mobile Location Alarm System operates as an integral component within the framework of modern mobile technology, seamlessly interfacing with mobile platforms and leveraging built-in location services. It functions primarily as a standalone mobile application, yet it retains the flexibility for potential integration with third-party services or applications. Central to its operation are several key facets within the product perspective. Firstly, the system interfaces directly with the mobile device's operating system to access and utilize location services, ensuring the accuracy of geographical data. Additionally, it presents users with an intuitive interface through a dedicated mobile application, facilitating tasks such as user registration, geofence creation, alarm configuration, and real-time location tracking. Moreover, the system relies on the underlying hardware components of the mobile device, particularly GPS sensors, to obtain precise location data crucial for its functionality.

Communication interfaces play a vital role, utilizing internet connectivity and standard communication protocols to transmit data, including location information and notifications. Dependencies on consistent internet connectivity and the availability of accurate location services from the mobile device underscore the system's reliance on external factors.

Finally, the potential for integration with external services, ranging from cloud storage for user data to social media platforms for location sharing, highlights its versatility and potential for expansion.

2.2 Product Functions

The main functionalities of the system are as follows:

User Registration and Authentication:

- *User Registration: Create accounts securely with unique identifiers.*
- *Authentication: Ensure secure access, protecting user data.*

Location Tracking:

- *Real-time Location Monitoring: Use mobile device location services for accurate, real-time geographical information.*
- *Continuous Tracking: Monitor locations constantly for up-to-date data.*

Geofencing:

- *Geofence Creation: Allow users to define geographic boundaries on the map.*
- *Alarm Triggering: Initiate alarms promptly upon entering or exiting predefined geofenced areas.*

Alarm Configuration:

- *Customizable Settings: Permit users to configure alarm preferences (sound, vibration, notification frequency).*
- *Multiple Alarms: Support the setup of multiple alarms with distinct configurations.*

Notification Alerts:

- *Instant Alerts: Send immediate notifications to users upon entering or exiting specified geofences.*
- *Informative Notifications: Include relevant details such as the geofence name and location information.*

User Interface:

- *Intuitive Design: Provide an easy-to-use and visually appealing interface for seamless interaction.*
- *Functional Screens: Display real-time location information and facilitate straightforward alarm configuration.*

Privacy and Security:

- *Privacy Measures: Implement robust protocols to protect user location data.*
- *Secure Data Transmission: Ensure secure transmission and storage of sensitive information.*

Compatibility:

- *Multi-Platform Support: Design to run on major mobile platforms for compatibility with various devices.*
- *Adaptability: Adjust to different screen sizes and resolutions for a consistent user experience.*

Documentation:

- *User Manuals: Offer comprehensive guides for users on system functionality.*
- *Technical Documentation: Provide in-depth technical documentation for developers and administrators.*

2.3 User Classes and Characteristics

The Mobile Location Alarm System caters to diverse user classes, each with distinct characteristics reflecting their needs and interaction with the system:

End Users:

Characteristics:

- *Non-technical individuals seeking a user-friendly safety solution.*
- *Varied age groups, requiring an intuitive interface for efficient navigation.*

Interaction:

- *Use the application for setting up geofences, configuring alarms, and receiving notifications.*

Administrators:

Characteristics:

- *Technical proficiency to manage system settings and user accounts.*
- *Responsible for system maintenance and addressing technical issues.*

Interaction:

- *Access administrative controls for user management, system configurations, and troubleshooting.*

Developers:

Characteristics:

- *Technical expertise in software development and system integration.*
- *Involved in system enhancements, bug fixes, and integration with external services.*

Interaction:

- *Refer to technical documentation for system integration and contribute to system development.*

2.4 Operating Environment

The Mobile Location Alarm System operates within a dynamic and interconnected environment, involving both hardware and software components. Key aspects of the operating environment include:

Mobile Platforms:

- *The system is designed to run on major mobile operating systems, such as iOS and Android, ensuring compatibility with a wide range of smartphones and tablets.*

Device Hardware:

- *Utilizes hardware components of mobile devices, including GPS sensors, for accurate location tracking.*

Internet Connectivity:

- *Requires a stable internet connection for real-time data transmission, notifications, and updates. The system may adjust its functionality based on the quality and availability of the internet connection.*

Location Services:

- *Relies on mobile device location services to obtain accurate geographical information.*

External Services and APIs:

- *May integrate with external services, such as map data providers, to enhance location accuracy and provide additional functionalities.*

2.5 Design and Implementation Constraints

Several constraints may influence the design and implementation of the Mobile Location Alarm System, impacting its development and functionality:

1. ***Mobile Platform Limitations:***
   - *Adherence to the constraints imposed by different mobile operating systems (iOS, Android) may limit certain functionalities or require platform-specific implementations.*
2. ***Hardware Variability:***
   - *Varied hardware configurations among mobile devices may affect the system's performance and necessitate optimization for different specifications.*
3. ***Internet Connectivity:***
   - *The system heavily relies on internet connectivity; limitations in network availability or speed may affect real-time data transmission and responsiveness.*
4. ***Location Service Accuracy:***
   - *The accuracy of location services provided by mobile devices may vary, impacting the precision of geofencing and location tracking functionalities.*
5. ***Battery Consumption:***
   - *Continuous use of location services and background processes may impact device battery life; the system needs to optimize power consumption.*
6. ***Data Privacy Regulations:***
   - *Compliance with data privacy regulations may impose constraints on data collection, storage, and transmission, influencing system design and user data handling.*

2.6 Assumptions and Dependencies

Assumptions:

1. ***Internet Availability:***
   - *It is assumed that users will have consistent access to the internet for real-time data transmission and seamless functionality.*
2. ***Device Compatibility:***
   - *Users are expected to have mobile devices that support the necessary hardware components, such as GPS sensors, for accurate location tracking.*
3. ***User Cooperation:***
   - *Assumption is made that users will actively engage with the application, configure alarms, and enable necessary permissions for optimal system functionality.*
4. ***Data Accuracy:***
   - *The accuracy of location data provided by mobile device services is assumed to be reliable for effective geofencing and alarm triggering.*
5. ***User Understanding:***
   - *Users are assumed to have a basic understanding of the application's features, including geofence creation and alarm configuration.*

Dependencies:

1. ***Mobile Platform APIs:***
   - *The system relies on APIs provided by mobile platforms (iOS, Android) for access to device functionalities, such as location services.*
2. ***External Map Data Providers:***
   - *Dependency on external map data providers or mapping APIs for accurate geographical information and mapping functionalities.*
3. ***Network Infrastructure:***

- *Dependencies on stable and secure network infrastructure for data transmission and communication between the mobile application and external services.*

4. ***Third-Party Services:***
    - *Integration with third-party services, if applicable, is a dependency, and the availability and reliability of these services may impact the system's functionality.*
5. ***Regulatory Compliance:***
    - *Compliance with data privacy regulations and legal requirements is a dependency, and any changes in regulations may necessitate adjustments to the system.*
6. ***Mobile Platform Updates:***
    - *Dependencies on updates and features provided by mobile operating systems, requiring periodic adjustments to maintain compatibility.*

3. External Interface Requirements

3.1 User Interfaces

User Registration and Authentication:

- *Registration Screen for account creation.*
- *Authentication Screen for logging in with credentials or biometric options.*

Dashboard:

- *Overview with active alarms and recent location history.*
- *Map View for visual representation of the user's current location.*

Geofence Configuration:

- *Map-based interface for creating geographic boundaries.*
- *Geofence List displaying created geofences for editing or deletion.*

Alarm Configuration:

- *Customizable settings for sound, vibration, and notification frequency.*
- *Overview of active alarms with toggle options.*

Notifications:

- *Alerts Screen displaying real-time notifications.*
- *Includes details such as geofence name and location.*

User Profile:

- *Profile Information for viewing and updating details.*
- *Options for changing passwords and account settings.*

Help and Documentation:

- *User Manual for system features and functionalities.*
- *Support/Contact Us section for customer support and feedback.*

3.2 Hardware Interfaces

The Mobile Location Alarm System interacts with various hardware components to ensure accurate location tracking and seamless functionality. Key hardware interfaces include:

1. ***Mobile Device:***
    - *The primary hardware interface is the user's mobile device (smartphone or tablet), serving as the platform for the Mobile Location Alarm System.*
2. ***GPS Sensors:***
    - *Utilizes the device's GPS sensors to obtain precise location data for accurate tracking and geofence functionality.*
3. ***Network Connectivity:***
    - *Relies on the device's network interfaces (Wi-Fi, cellular data) to establish and maintain internet connectivity for real-time data transmission.*
4. ***Speaker and Vibration Motor:***
    - *Interfaces with the device's speaker and vibration motor to deliver audible alarms and haptic feedback based on user-configured settings.*

3.3 Software Interfaces

The Mobile Location Alarm System relies on various software interfaces to function seamlessly and integrate with external services. Key software interfaces include:

1. ***Mobile Operating Systems (iOS, Android):***
    - *Adapts to the software specifications and requirements of major mobile platforms, ensuring compatibility and optimal performance.*
2. ***Mobile Platform APIs:***
    - *Utilizes APIs provided by mobile operating systems for access to device functionalities, including location services, notifications, and system controls.*
3. ***Location Services API:***
    - *Interfaces with the device's location services API to obtain accurate and real-time geographical information for location tracking.*
4. ***Mapping APIs (e.g., Google Maps, Mapbox):***
    - *Integrates with mapping APIs to provide detailed maps, geocoding, and additional mapping functionalities.*
5. ***Notification Services (e.g., Firebase Cloud Messaging):***
    - *Interfaces with notification services to deliver real-time alerts to users' devices.*

3.4 Communications Interfaces

The Mobile Location Alarm System relies on various communication interfaces to facilitate data exchange between its components and external services. These interfaces ensure real-time updates, notifications, and seamless interactions. Key communication interfaces include:

1. ***Mobile Device to Server:***
    - *Utilizes secure communication protocols (HTTP/HTTPS) to transmit user data, location information, and configuration settings between the mobile application and the server.*
2. ***Server to Mobile Device:***
    - *Sends real-time updates, notifications, and relevant information from the server to the mobile device, ensuring timely communication with users.*
3. ***Mobile Device to Mapping API:***
    - *Interacts with mapping APIs (e.g., Google Maps, Mapbox) to retrieve map data and geospatial information for accurate representation on the device.*
4. ***Mobile Device to Location Services API:***
    - *Communicates with the device's location services API to obtain precise and real-time location data for tracking and geofencing.*

5. ***Mobile Device to Notification Service:***
    - *Utilizes notification services (e.g., Firebase Cloud Messaging) to push real-time alerts and notifications to users' devices.*

4. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

4.1 User Registration and Authentication:

4.1.1 User-friendly registration process with unique identifiers.

4.1.2 Secure authentication methods, including biometric options.

4.2 Real Time Location Tracking:

4.2.1 Utilizes mobile device location services for accurate and continuous tracking.

4.2.2 Provides real-time updates on the user's geographical position.

4.3 Geofencing Functionality:

4.3.1 Intuitive geofence creation through a map-based interface.

4.3.2 Configurable geofence parameters, including name and radius.

4.3.3 Alarms triggered upon entering or exiting defined geofenced areas.

4.4 Alarm Configuration:

4.4.1 Customizable alarm settings for sound, vibration, and notification frequency.

4.4.2 Support for multiple alarms with distinct configurations.

4.5 Notification Alerts:

4.5.1 Instant alerts sent to users upon entering or exiting specified geofences.

4.5.2 Notifications include relevant details such as geofence name and location information.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The performance requirements of the Mobile Location Alarm System are crucial to ensure optimal functionality, responsiveness, and efficiency. Key performance metrics and requirements include:

1. ***Location Tracking Accuracy***
2. ***Real-time Updates***
3. ***Geofence Response Time***
4. ***Application Responsiveness***
5. ***Notification Delivery Time***
6. ***Scalability***
7. ***Data Transmission Speed***
8. ***Battery Efficiency***
9. ***Platform Compatibility***
10. ***Offline Functionality***

5.2 Safety Requirements

Ensuring the safety of users and the proper functioning of the Mobile Location Alarm System is paramount. The safety requirements are designed to mitigate potential risks and provide a secure user experience:

1. ***Data Privacy and Protection:***
   - *Requirement: Implement robust encryption protocols to protect user data during transmission and storage, adhering to relevant data protection laws and regulations.*
2. ***Secure User Authentication:***
   - *Requirement: Employ secure user authentication methods, including multi-factor authentication, to prevent unauthorized access to user accounts.*
3. *Geofence Validation:*
   - *Requirement: Implement mechanisms to validate and verify user-defined geofences to prevent false alarms or inaccurate location-based triggers.*
4. *User Education and Awareness:*
   - *Requirement: Provide educational materials within the application to inform users about the proper use of location-based alarms, geofencing, and potential safety considerations.*
5. *Notification Clarity:*
   - *Requirement: Ensure that notification alerts clearly convey the nature of the triggered alarm, including the geofence name and location, to avoid confusion or misunderstanding.*

6. *Battery Usage Safety:*
    - *Requirement: Optimize power consumption to prevent excessive draining of the device's battery, avoiding potential safety concerns associated with unexpected shutdowns.*
7. *User Location Privacy:*
    - *Requirement: Allow users to control and customize the level of location sharing, emphasizing user privacy and providing options for selective sharing.*

5.3 Security Requirements

Security is a critical aspect of the Mobile Location Alarm System to protect user data, ensure system integrity, and prevent unauthorized access. The following security requirements are essential for maintaining a secure environment:

1. ***User Authentication:***
    - *Requirement: Implement strong user authentication mechanisms, such as secure passwords or biometric authentication, to ensure only authorized users access the system.*
2. ***Data Encryption:***
    - *Requirement: Use encryption protocols (e.g., TLS) for secure data transmission between the mobile application and the server to protect sensitive information from unauthorized interception.*
3. ***Secure Storage:***
    - *Requirement: Safeguard user data, geofence configurations, and alarm settings by securely storing them in the database using encryption and access controls.*
4. ***Authorization Controls:***
    - *Requirement: Enforce proper authorization controls to restrict access to sensitive functionalities and user data, ensuring that users can only access information relevant to them.*
5. ***Secure APIs:***
    - *Requirement: Implement secure APIs with proper authentication and authorization mechanisms to prevent unauthorized access to system functionalities.*

5.4 Software Quality Attributes

The Mobile Location Alarm System is designed with various software quality attributes to ensure a robust, reliable, and user-friendly application. These attributes contribute to the overall performance, usability, and maintainability of the system:

1. ***Reliability:***
    - *Attribute: The system consistently performs its intended functions accurately, ensuring users can rely on accurate location tracking and timely alarms.*
2. ***Availability:***
    - *Attribute: The system is available for use whenever users require it, minimizing downtime and ensuring uninterrupted access to location-based services.*
3. ***Performance Efficiency:***
    - *Attribute: The system optimizes resource usage, providing efficient location tracking, geofencing, and alarm functionality while minimizing energy consumption and response times.*
4. ***Usability:***
    - *Attribute: The user interfaces are intuitive and user-friendly, ensuring that users can easily configure alarms, set up geofences, and navigate the application without confusion.*
5. ***Scalability:***

- *Attribute: The system can handle increased user loads and data volumes without a significant decrease in performance, allowing for scalability as the user base grows.*

6. ***Maintainability:***
    - *Attribute: The system is designed with clean and modular code, facilitating easy maintenance, updates, and enhancements by developers.*
7. ***Security:***
    - *Attribute: The system incorporates robust security measures to protect user data, prevent unauthorized access, and ensure secure communication between the mobile application and the server.*
8. ***Portability:***
    - *Attribute: The system can run seamlessly on different mobile platforms (iOS, Android) and adapt to varying screen sizes and resolutions.*
9. ***Flexibility:***
    - *Attribute: The system allows for easy customization of alarm settings, geofences, and other configurations, providing flexibility to meet diverse user needs.*
10. ***Accuracy:***
    - *Attribute: The system delivers accurate location tracking and geofencing, ensuring that alarms are triggered precisely when users enter or exit predefined areas.*

6. Other Requirements

In addition to functional, performance, safety, and security requirements, there are other miscellaneous requirements that contribute to the overall success and usability of the Mobile Location Alarm System:

1. ***Documentation:***
    - *Requirement: Provide comprehensive documentation, including user manuals, developer guides, and system architecture documentation, to facilitate understanding, usage, and maintenance.*
2. ***Localization:***
    - *Requirement: Support localization to accommodate users from different regions, ensuring that the application is available in multiple languages.*
3. ***User Feedback Mechanism:***
    - *Requirement: Implement a user feedback mechanism within the application to gather user suggestions, identify issues, and continuously improve the system based on user input.*
4. ***Backup and Recovery:***
    - *Requirement: Implement regular backups of user data and system configurations to prevent data loss. Develop a recovery plan to restore the system in case of unforeseen incidents.*
5. ***Versioning and Updates:***
    - *Requirement: Establish a versioning system for the application and provide timely updates with bug fixes, feature enhancements, and security patches to ensure a continuously improving user experience.*

Appendix A: Glossary

To ensure clarity and understanding while interpreting the Software Requirements Specification (SRS) for the Mobile Location Alarm System, the following terms, acronyms, and abbreviations are defined:

1. ***SRS:*** *Software Requirements Specification*
   - *Definition: A document that outlines the detailed requirements and specifications for the development of a software system.*
2. ***GPS:*** *Global Positioning System*
   - *Definition: A satellite-based navigation system that provides real-time location and time information.*
3. ***API:*** *Application Programming Interface*
   - *Definition: A set of rules and protocols that allows different software applications to communicate with each other.*
4. ***UI:*** *User Interface*
   - *Definition: The visual elements and interactive components through which users interact with a software application.*
5. ***DBMS:*** *Database Management System*
   - *Definition: A software system that manages the organization, storage, and retrieval of data in a database.*
6. ***HTTP/HTTPS:*** *Hypertext Transfer Protocol/Secure*
   - *Definition: Protocols used for transmitting data over the internet, with HTTPS providing a secure, encrypted version of HTTP.*
7. ***SDK:*** *Software Development Kit*
   - *Definition: A set of tools and resources that developers use to create applications for specific software platforms.*

ACTIVITY DIAGRAM

For

Mobile Location Alarm

Prepared by Sagnik Roy

UPES

15.02.2024

Revision History

Name	Date	Reason For Changes	Version

1. ACTIVITY DIAGRAM DESCRIPTION

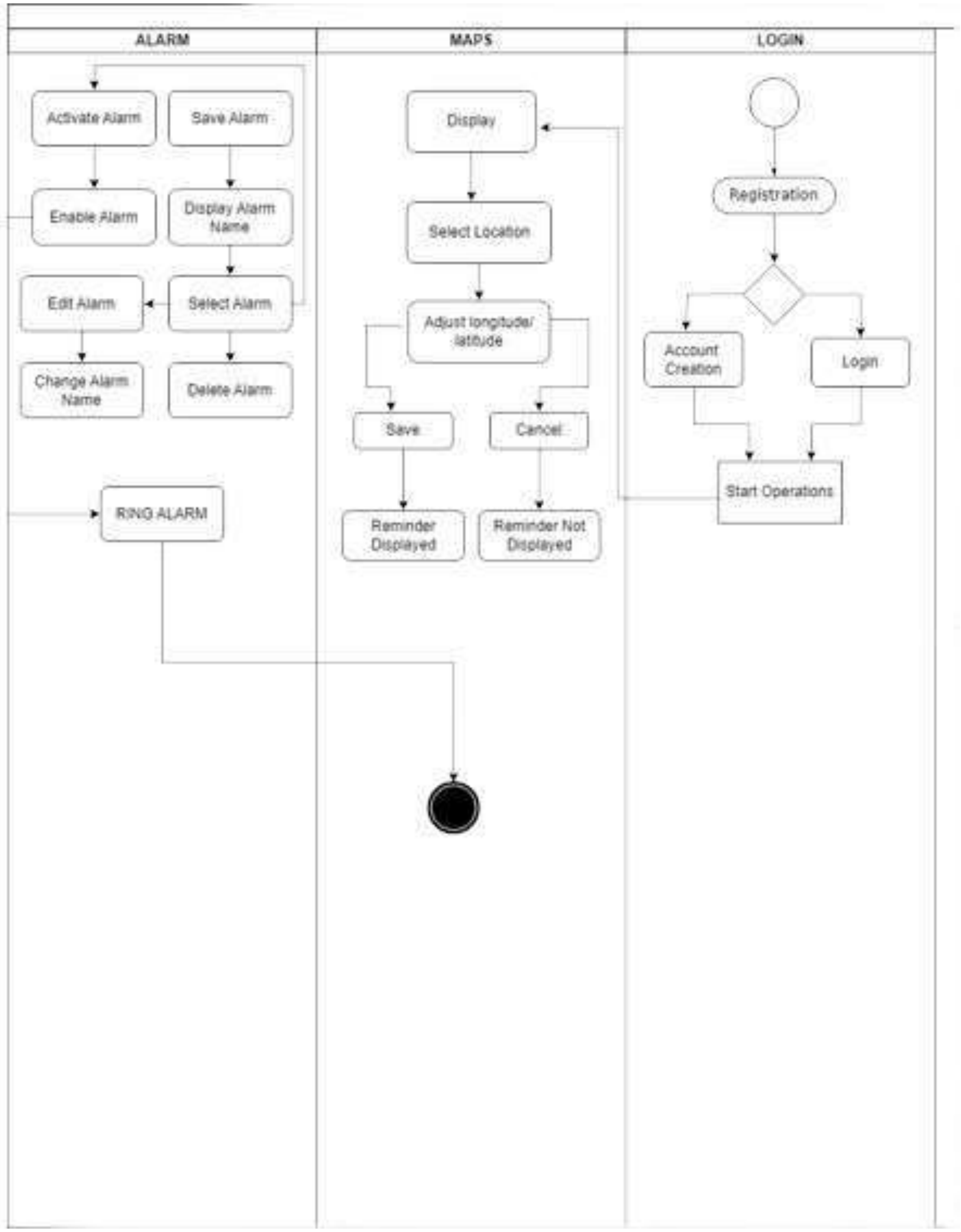
An activity diagram is a visual representation of any system's activities and flows of data or decisions between activities.

- Activity diagrams provide a comprehensive view of a business process.
- They represent the dynamics of a system.
- They are flow charts used to show a system's workflow.
- They show the control flow from activity to activity in the system.
- They show what activities can be done in parallel, and any alternative paths through the flow.
- Purpose
- Model business workflows
- Model operations
- Activity diagrams commonly contain
- Activity states and action states
- Transitions
- Objects

2. CONCLUSION

From this practical, we have learnt how to design an activity diagram using draw.io.

3. Diagram with and without Swimlane



[SAGNIK ROY]

[500109927]

13/2/24

12/02/24

LAB-05

Topic → Mobile Location Alarm System

Movement of process →

Step 1 → Identify the process

↳ How many pages we have

↳ Login ↳ Registration ↳ Working Page

Step 2 → Design the flowchart for every pages

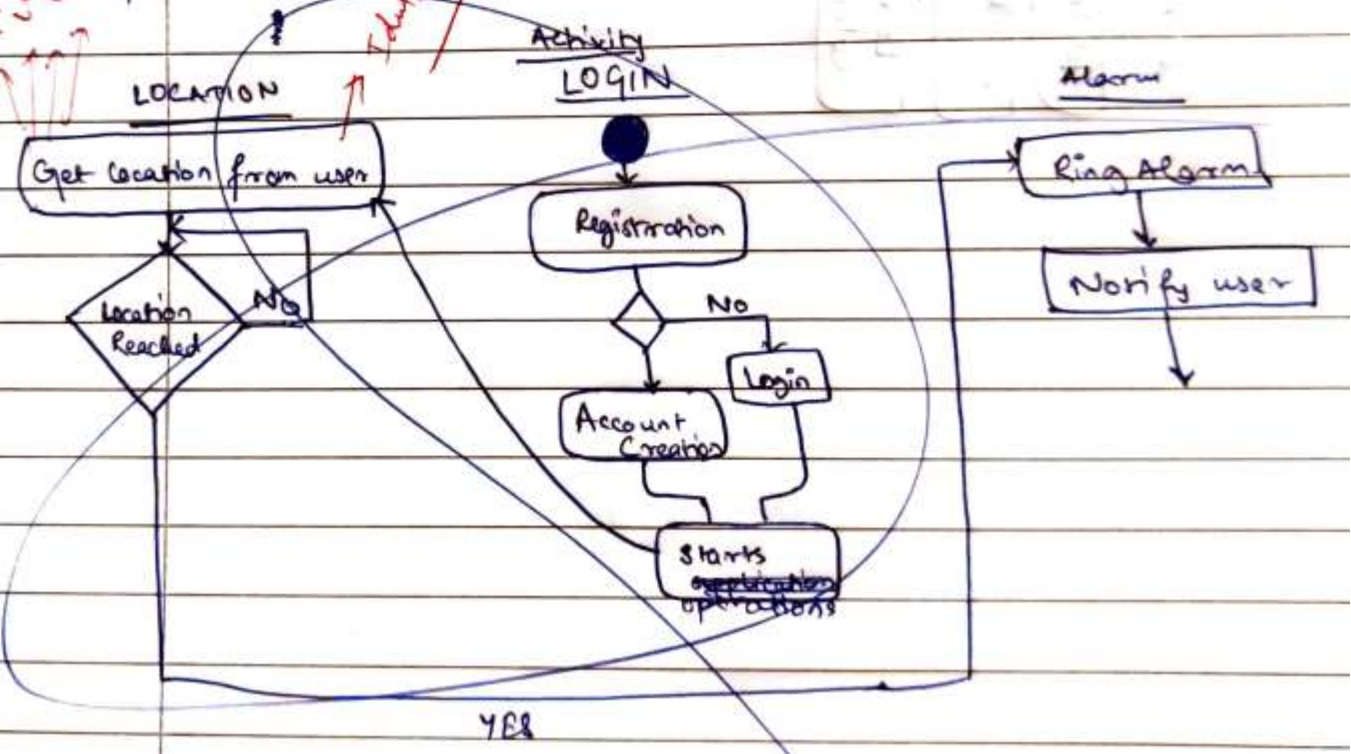
↳ Use the notation of UML (Activity Diagram)

Step 3 → Use swimlane to project the activity.

i) Location co-ordinates

ii) Landmarks Landmark only

iii) Reference



[SAGNIK ROY]
[500109927]
[Team Red]

20/2/24

__/__/__
20/02/24

Topic → Online Task Management System

I] Requirement Analysis:

- User-Friendly Interface: The system must feature an intuitive and user friendly interface, ensuring effortless task management through clear navigation, concise labelling and visually appealing design
- Task Assignment: User should be able to assign task to specific individuals and monitor their progress until compilation.
- Collaboration: The system must facilitate seamless collaboration among user, allowing for real time updates, communication and sharing of files and comments.
- Reminders and Notification: A strong and powerful notification system should be in place to alert users of upcoming tasks and deadline, leveraging email notifications, pop-up reminders or both.
- Reporting: Managers require a comprehensive reporting module to generate insights on task completion rates, time tracking & overall team performance aiding in productiving assessment and identifying areas ~~of~~ for enhancement
- User Authentication: Implementing a secure user authentication and authorisation mechanism is crucial to safeguard sensitive task related information & ensure that only authorized users have access.

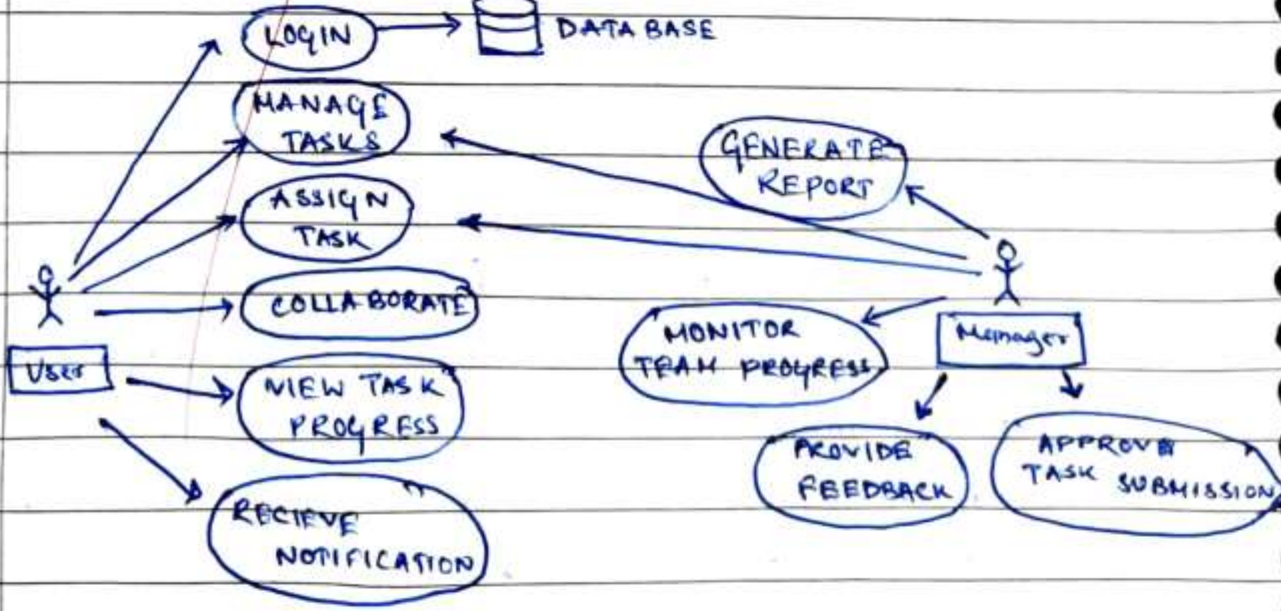
- Task Management Features : Users should have the ability to create, edit and delete tasks.
- Mobile Responsiveness : The system must be accessible across various devices allowing user to conveniently manage tasks on the go.
- Data Security & Privacy : Ensuring the security & privacy of user data is necessary , requiring measures to protect sensitive information.

II] USE CASE DIAGRAM

II.i) Actors

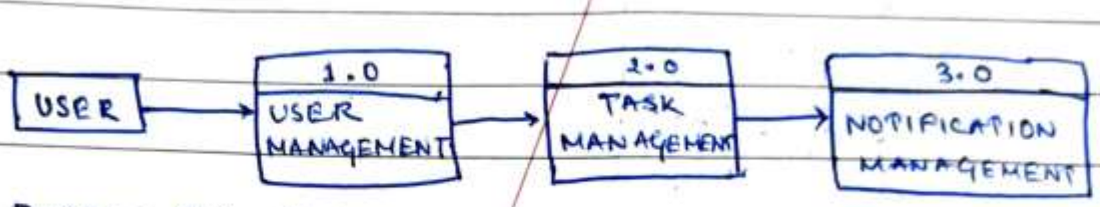
→ User

→ Manager



III] DFD Diagram

III.1) Level 0 →



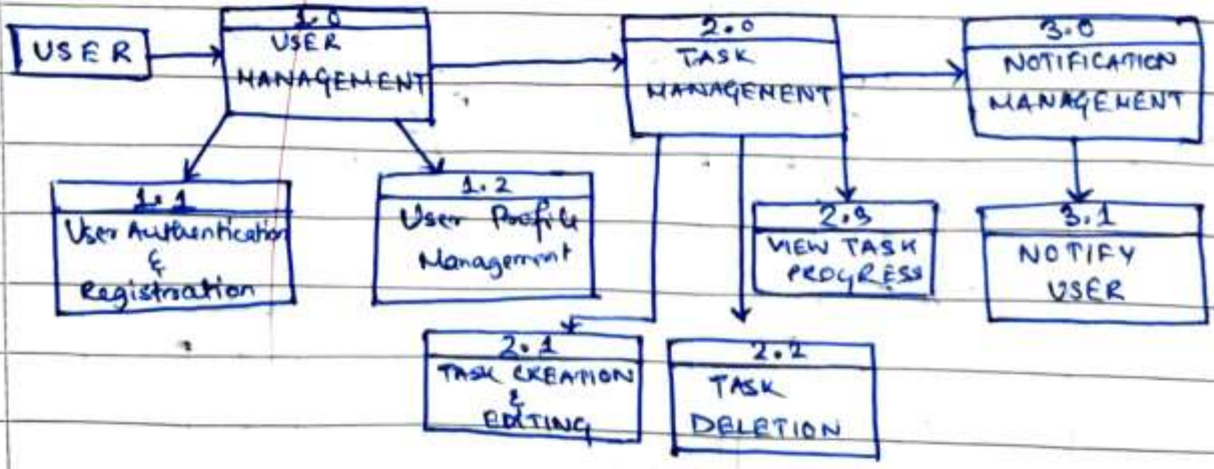
Entity - Online Task ~~Management~~ System or USER (here)

Processes - ① Task Management - Handles processes related to creating, editing and deleting tasks

② User Management - Handles processes related to ~~creating &~~ user authentication, registration & profile management.

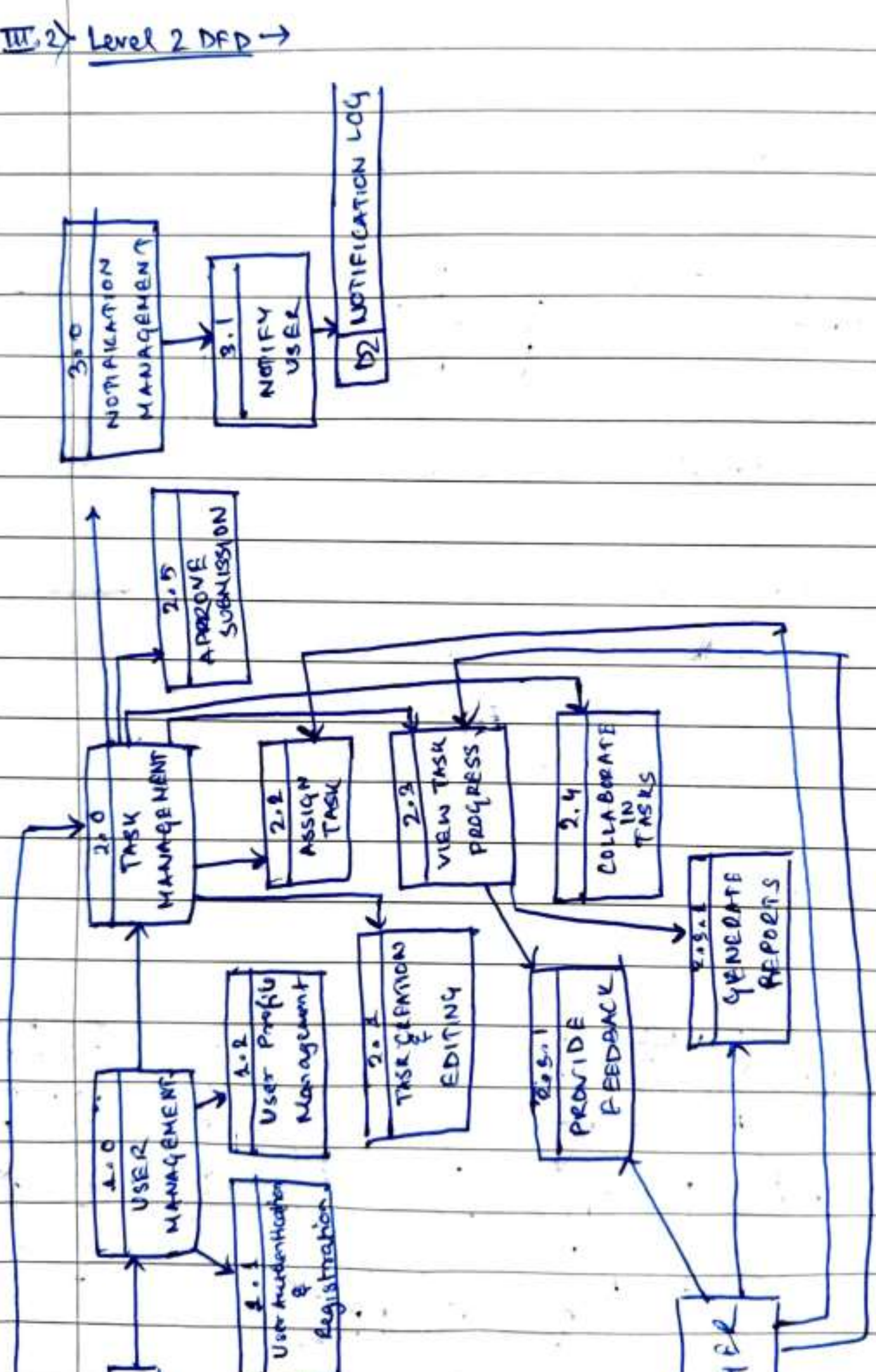
③ Notification Management - Manages processes related to sending notification for task assignment, update & deadline.

III.2) Level 1 →



__/__/____

III.2) Level 2 DFD →



INTERACTION DIAGRAMS

For

Mobile Location Alarm System

Prepared by Sagnik Roy

UPES

11.03.2024

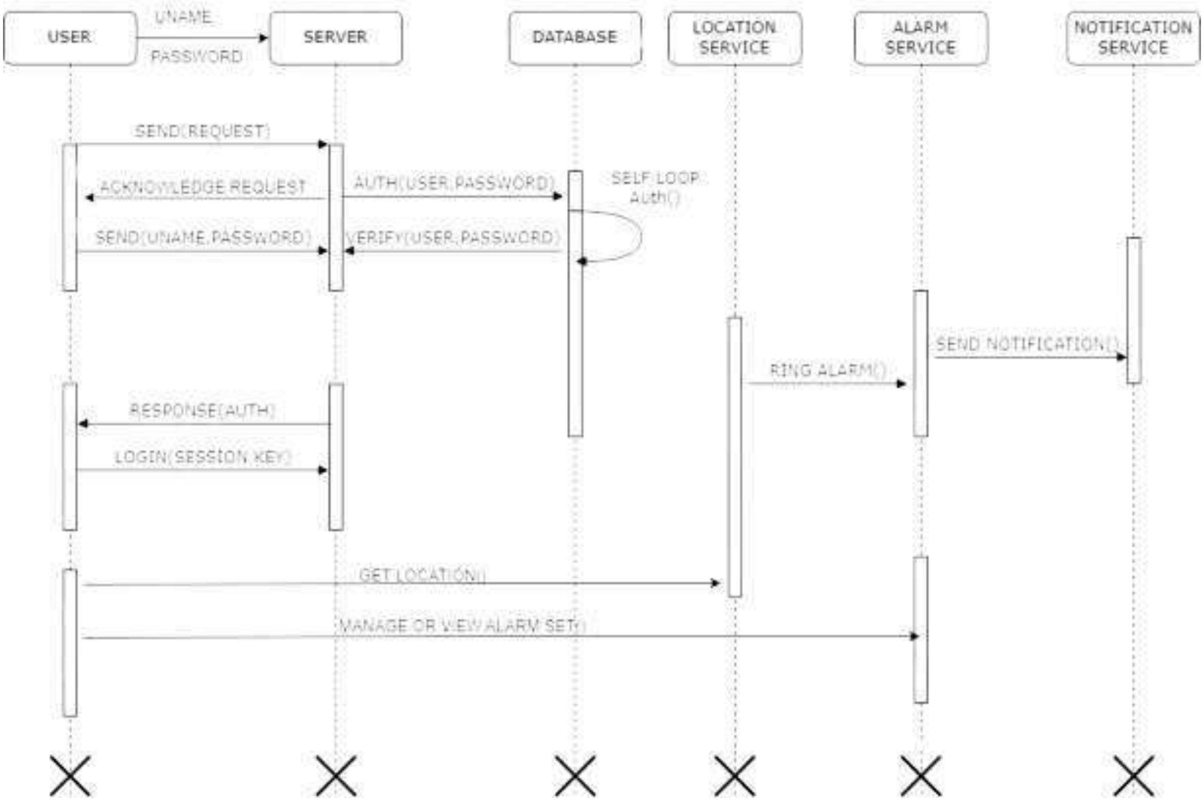
SEQUENCE DIAGRAM –

A sequence diagram is an interaction diagram that details how operations are carried out -- what messages are sent and when. Sequence diagrams are organized according to time. The time progresses as you go down the page. The objects involved in the operation are listed from left to right according to when they take part in the message sequence.

Sequence diagrams contain the following: –

- Class roles (subsystem/object/class, actor, and external system roles in the interaction). These are (usually) drawn across the top of the diagram.
- Lifelines (subsystem/object/class existence). These (usually) extend down the diagram.
- Activations (show when the subsystem/object/class is doing something)
- Messages (communication between roles)

SEQUENCE DIAGRAM FOR MOBILE LOCATION ALARM SYSTEM:



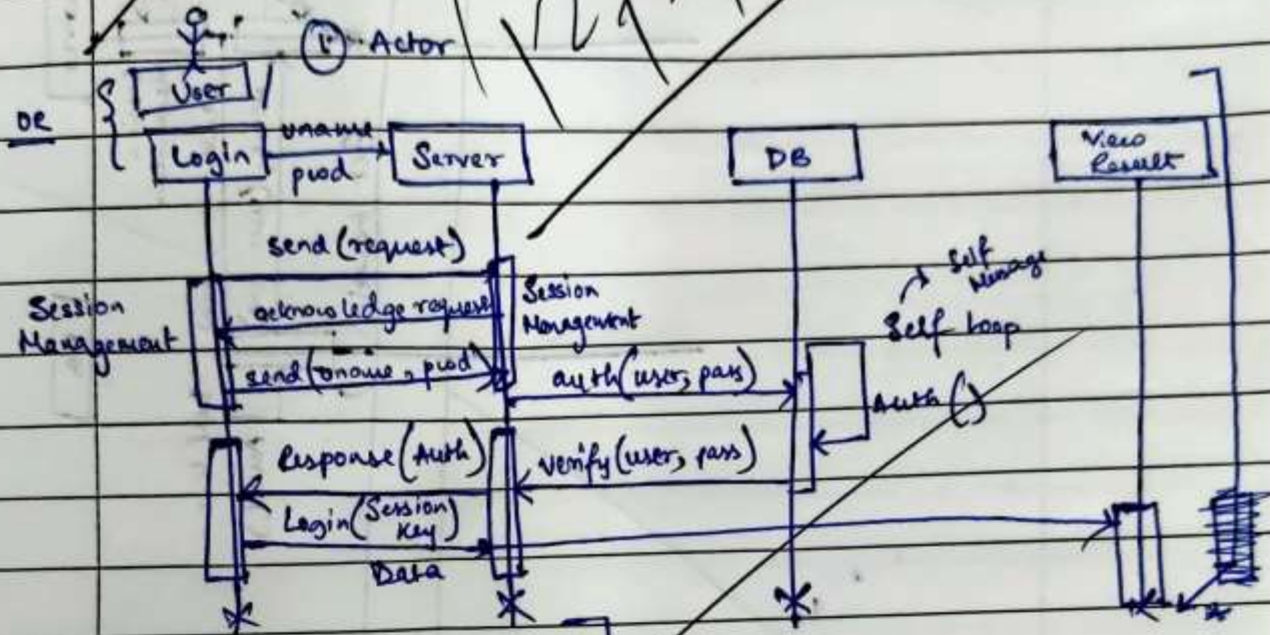
SAYNIK ROY
500109927

__/__/__

12/03/24

12/3/24

Lab-07 { Sequence Diagram } → Flow of Information



1. Actor
2. Lifeline
3. Message → Synchronous, → Asynchronous
4. Create Message
5. Delete Message
6. Self Message
7. Replay Message
8. Found Message
9. Lost Message
10. Guard

Use :-

- Behaviours of s/w
- Design and architecture
- Communication
- Requirement
- Debuging

Challenges—

- Dynamic System [HTML5]
- Real- time application
- Abstraction
- Concurrency
- Complexity

STATECHART DIAGRAM

For

Mobile Location Alarm

Prepared by Sagnik Roy

UPES

21.03.2024

Introduction

The name of the diagram itself clarifies the purpose of the diagram and other details. It describes different states of a component in a system. The states are specific to a component/object of a system. A Statechart diagram describes a state machine. Now to clarify it state machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.

Purpose:

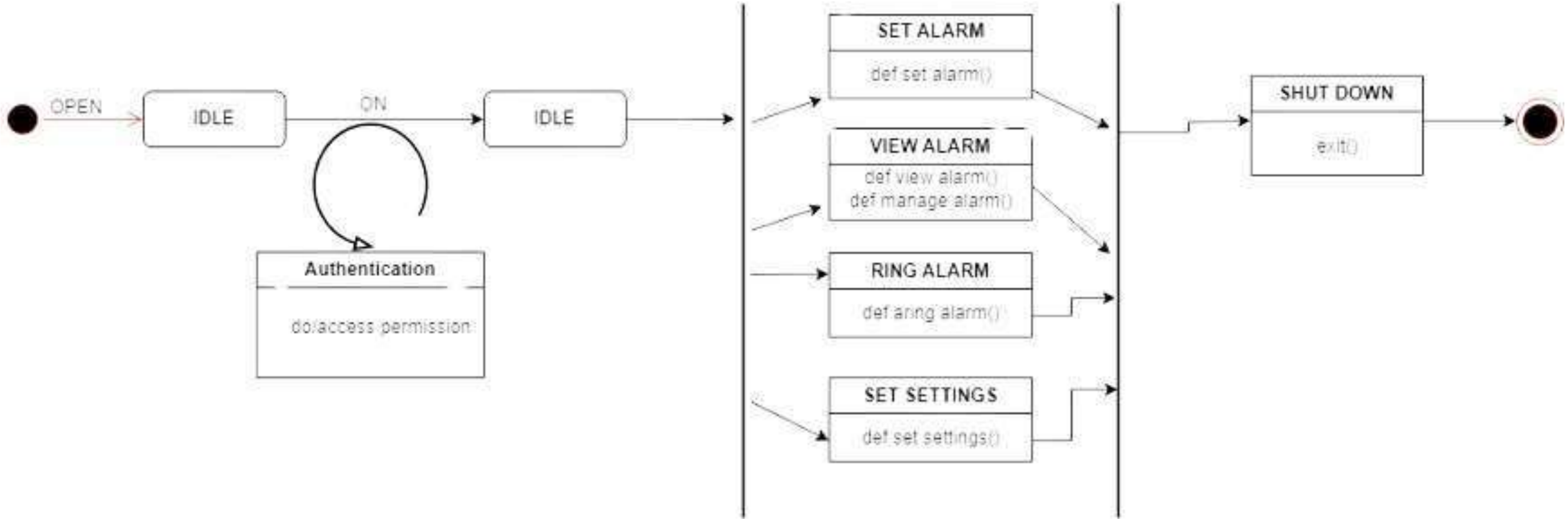
Statechart diagram is one of the five UML diagrams used to model dynamic nature of a system. They define different states of an object during its lifetime. And these states are changed by events. So Statechart diagrams are useful to model reactive systems. Reactive systems can be defined as a system that responds to external or internal events. Statechart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. So the most important purpose of Statechart diagram is to model life time of an object from creation to termination. Following are the main purposes of using Statechart diagrams:

- To model dynamic aspect of a system.
- To model life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model states of an object.

Conclusion:

From this practical we have learnt how to make State Chart Diagram using Rational Rose software in UML

- **Statechart diagram for MOBILE LOCATION ALARM-**



[SAGNIK ROY]
(500109927)

17/3/24

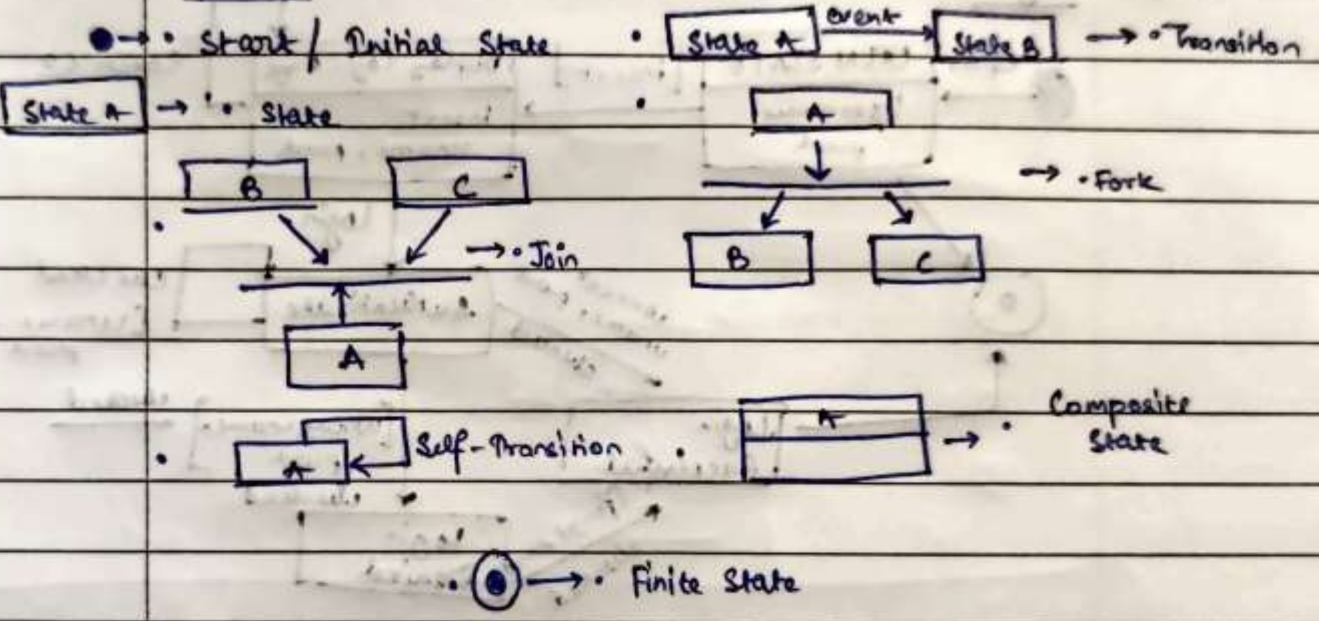
__/__/____

19/03/2024

LAB - 08

- State Chart / Machine Diagram
  ↳ It represents condition of system or part of the system at finite Instance of time.

Notations



Steps to Draw -

1) Identify the system.
2) Identify the initial and final states
3) Identify the possible states
4) Label Triggering Events
5) Draw with appropriate notation

COLLABORATION DIAGRAM

For

Mobile Location Alarm

Prepared by Sagnik Roy

UPES

09.04.2024

UML Collaboration Diagram

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.

When to use a Collaboration Diagram?

Collaborations are used when it is essential to depict the relationship between the objects. Both the sequence and collaboration diagrams represent the same information, but the way of portraying it is quite different. The collaboration diagrams are best suited for analyzing use cases. Following are some of the use cases enlisted below for which the collaboration diagram is implemented:

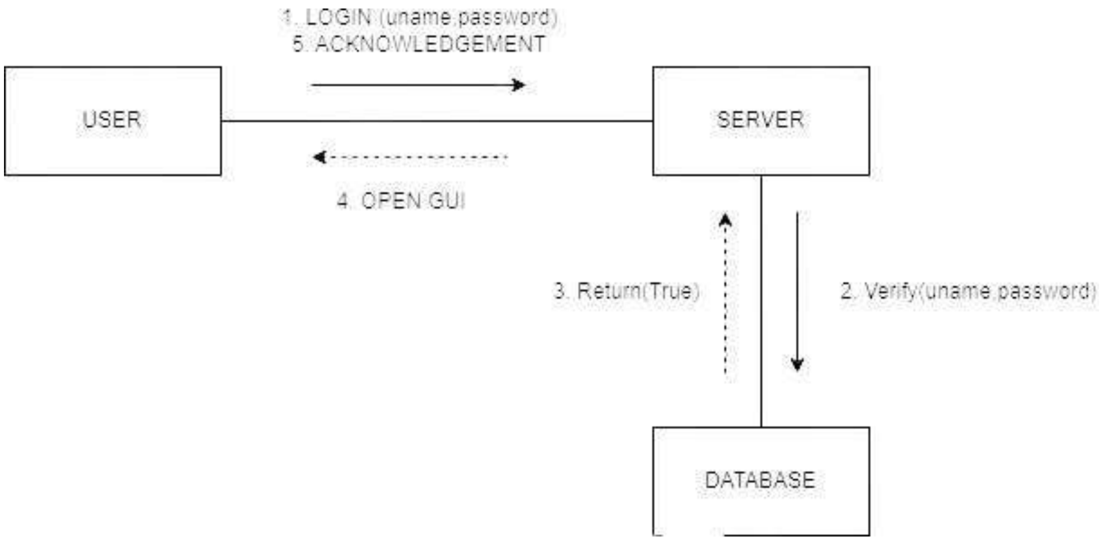
- To model collaboration among the objects or roles that carry the functionalities of use cases and operations.
- To model the mechanism inside the architectural design of the system.
- To capture the interactions that represent the flow of messages between the objects and the roles inside the collaboration.
- To model different scenarios within the use case or operation, involving a collaboration of several objects and interactions.
- To support the identification of objects participating in the use case.

In the collaboration diagram, each message constitutes a sequence number, such that the top-level message is marked as one and so on. The messages sent during the same call are denoted with the same decimal prefix, but with different suffixes of 1, 2, etc. as per their occurrence.

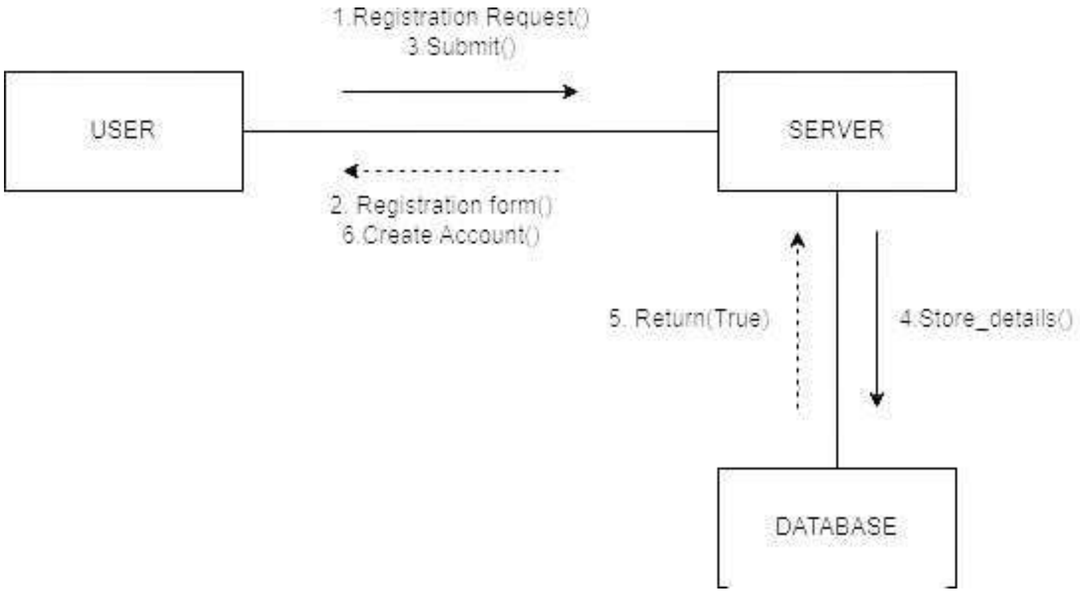
Steps for creating a Collaboration Diagram

- Determine the behavior for which the realization and implementation are specified.
- Discover the structural elements that are class roles, objects, and subsystems for performing the functionality of collaboration.
- Choose the context of an interaction: system, subsystem, use case, and operation.
- Think through alternative situations that may be involved.
- Implementation of a collaboration diagram at an instance level, if needed.
- A specification level diagram may be made in the instance level sequence diagram for summarizing alternative situations.

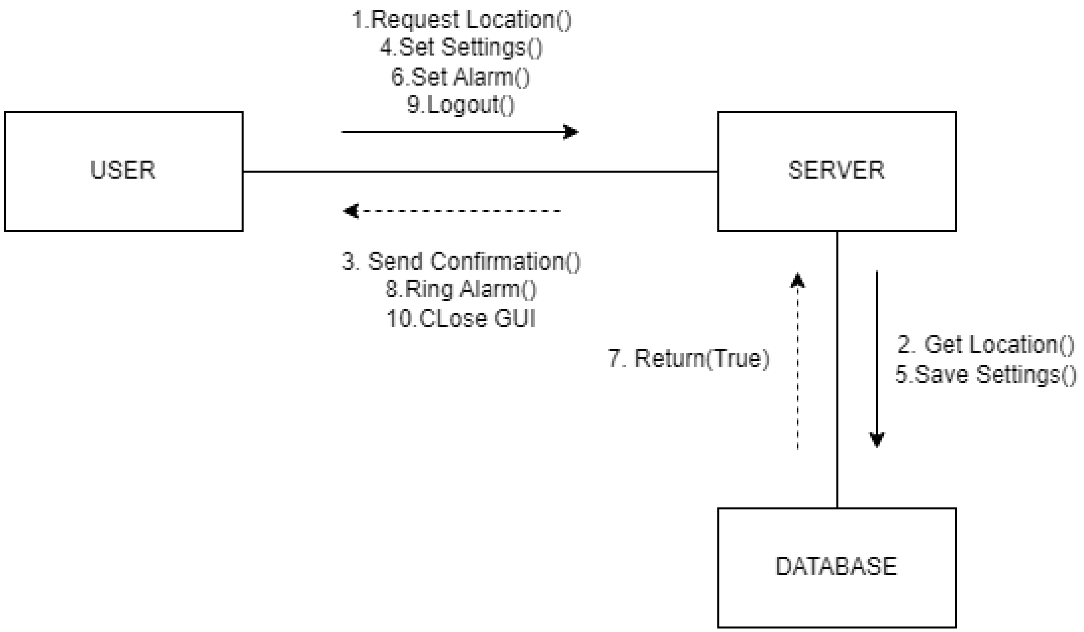
- **Collaboration diagram for LOGIN:**



- **Collaboration diagram for Registration:**



- **Collaboration diagram for Mobile Location Alarm:**

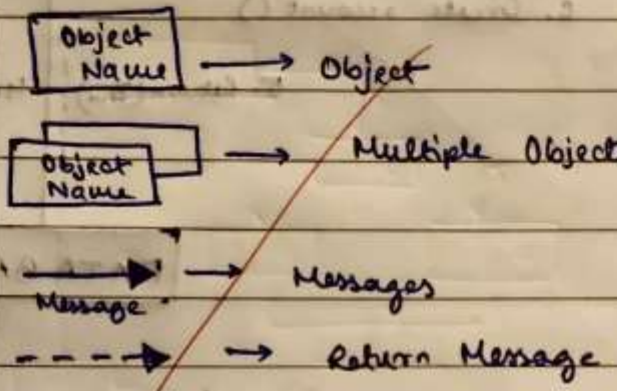


[SAGNIK ROY] Lab-9
(500109927)

09/04/24

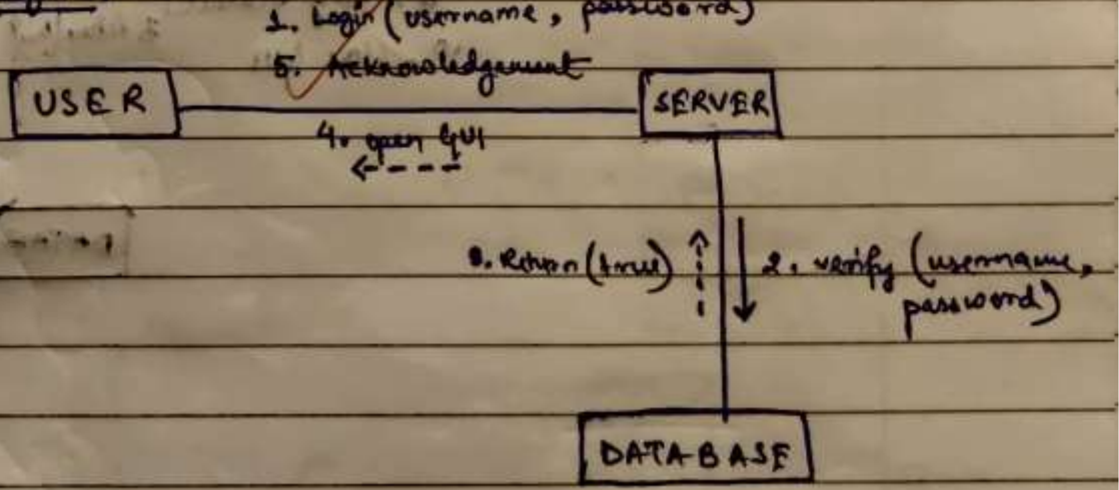
Collaboration Diagram

Representations :-

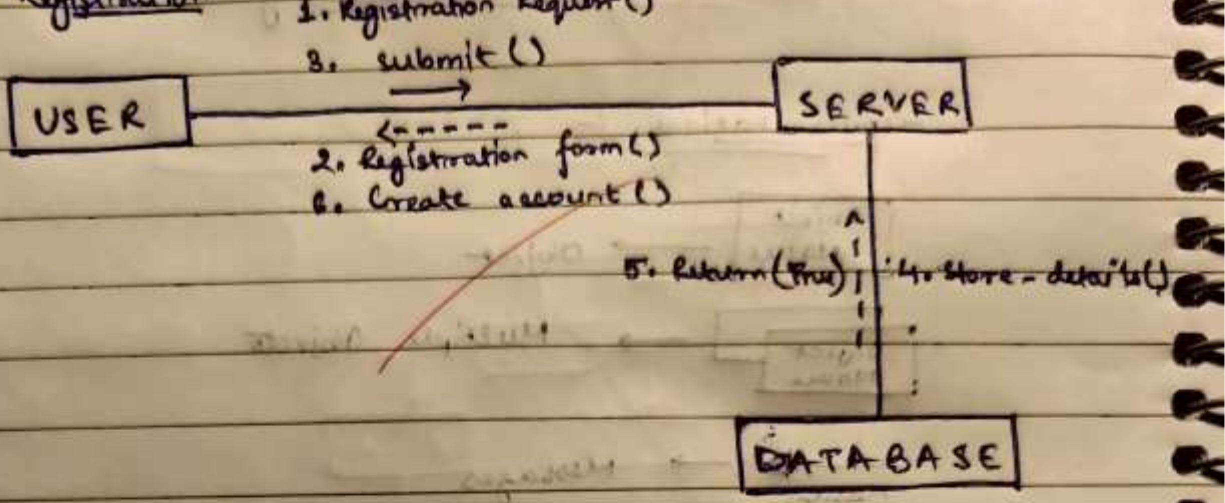


Topic → MOBILE LOCATION ALARM

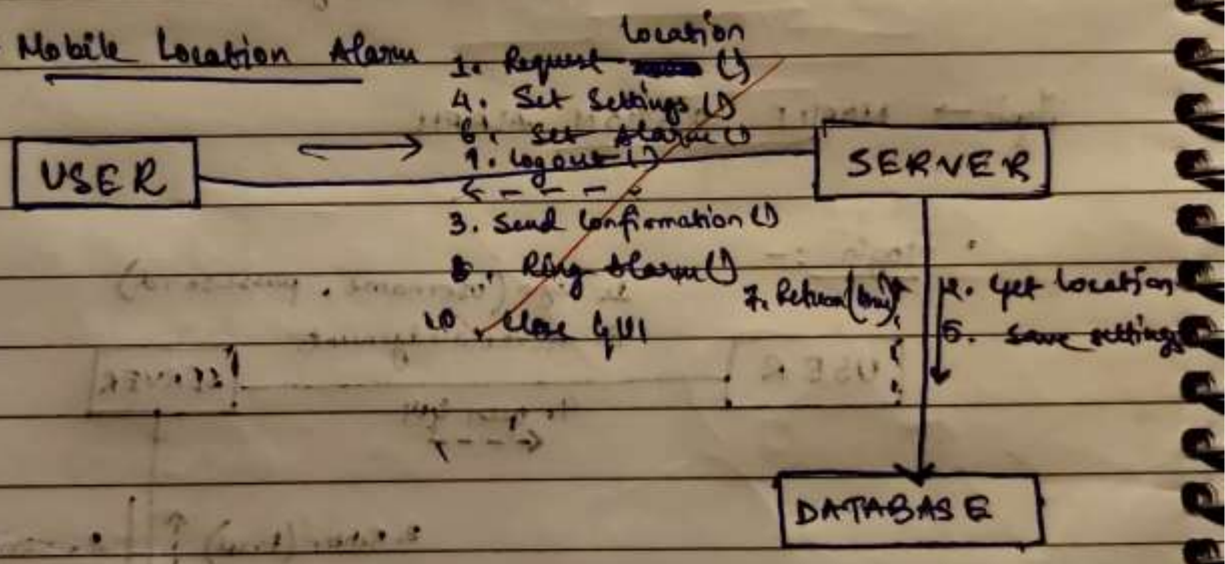
- Login :-



- Registration



- Mobile Location Alarm



DEPLOYMENT DIAGRAM

For

Mobile Location Alarm

Prepared by Sagnik Roy

UPES

09.04.2024

UML Deployment diagrams

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

Purpose:

The name Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components where software components are deployed. Component diagrams and deployment diagrams are closely related. Component diagrams are used to describe the components and deployment diagrams show how they are deployed in hardware. UML is mainly designed to focus on software artifacts of a system. But these two diagrams are special diagrams used to focus on software components and hardware components. Most of the UML diagrams are used to handle logical components but deployment diagrams are made to focus on hardware topology of a system. Deployment diagrams are used by the system engineers.

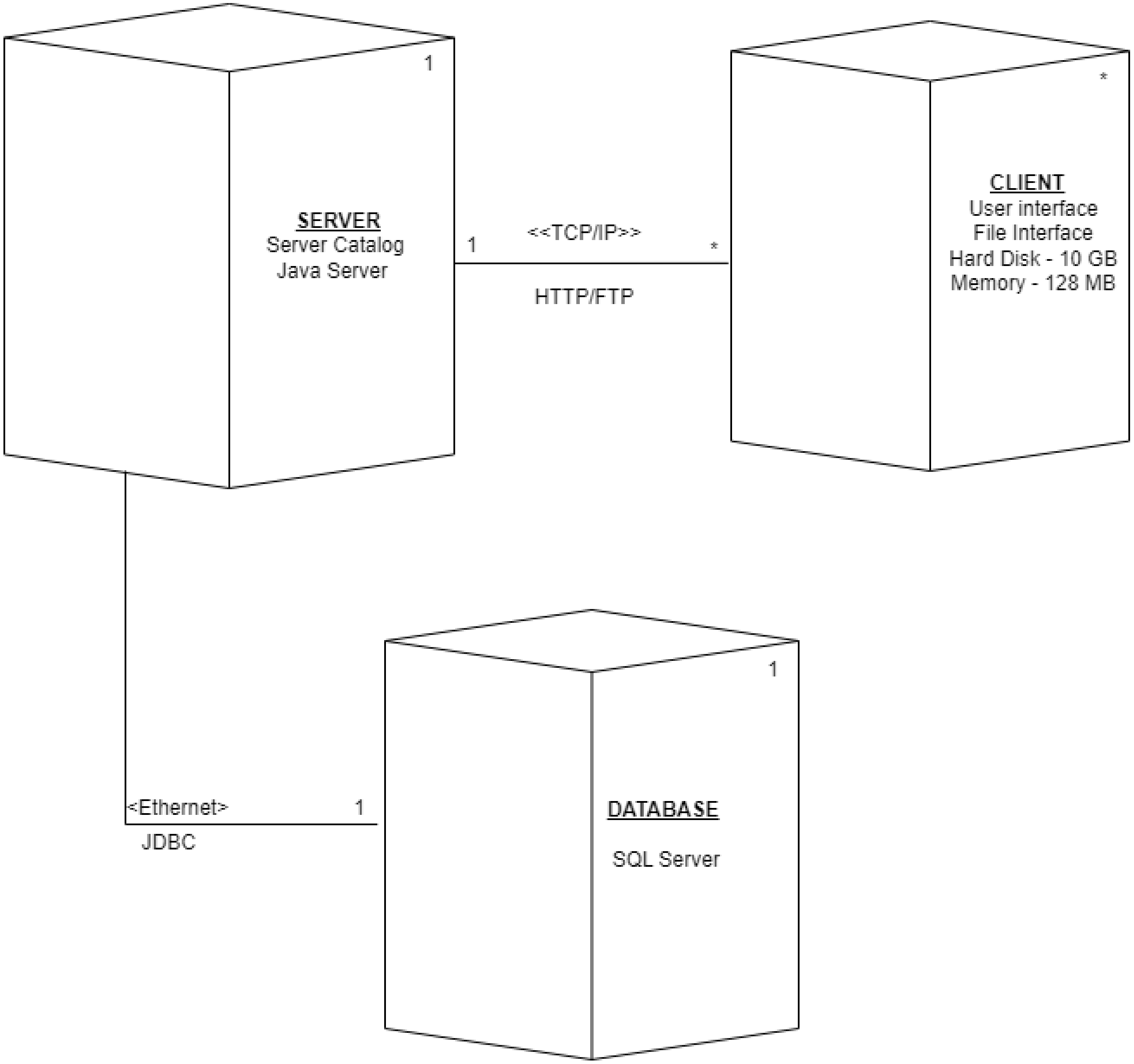
The purpose of deployment diagrams can be described as:

- Visualize hardware topology of a system.
- Describe the hardware components used to deploy software components.
- Describe runtime processing nodes.

Conclusion:

From this practical we have learnt how to make Deployment Diagram using Rational Rose software in UML.

- **DEPLOYMENT diagram for Mobile Location Alarm:**



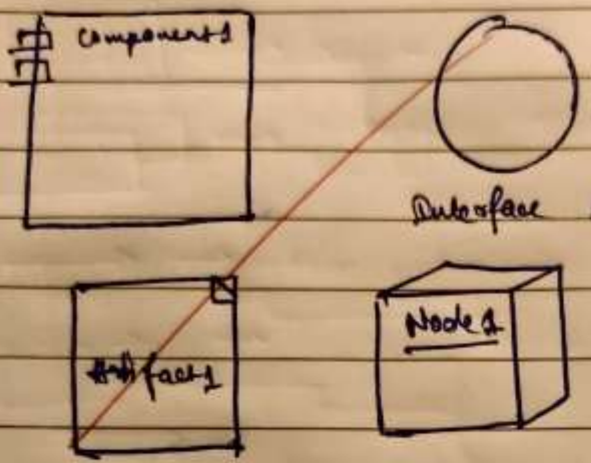
[SAGNIK ROY]
(500109927)

09/04/24

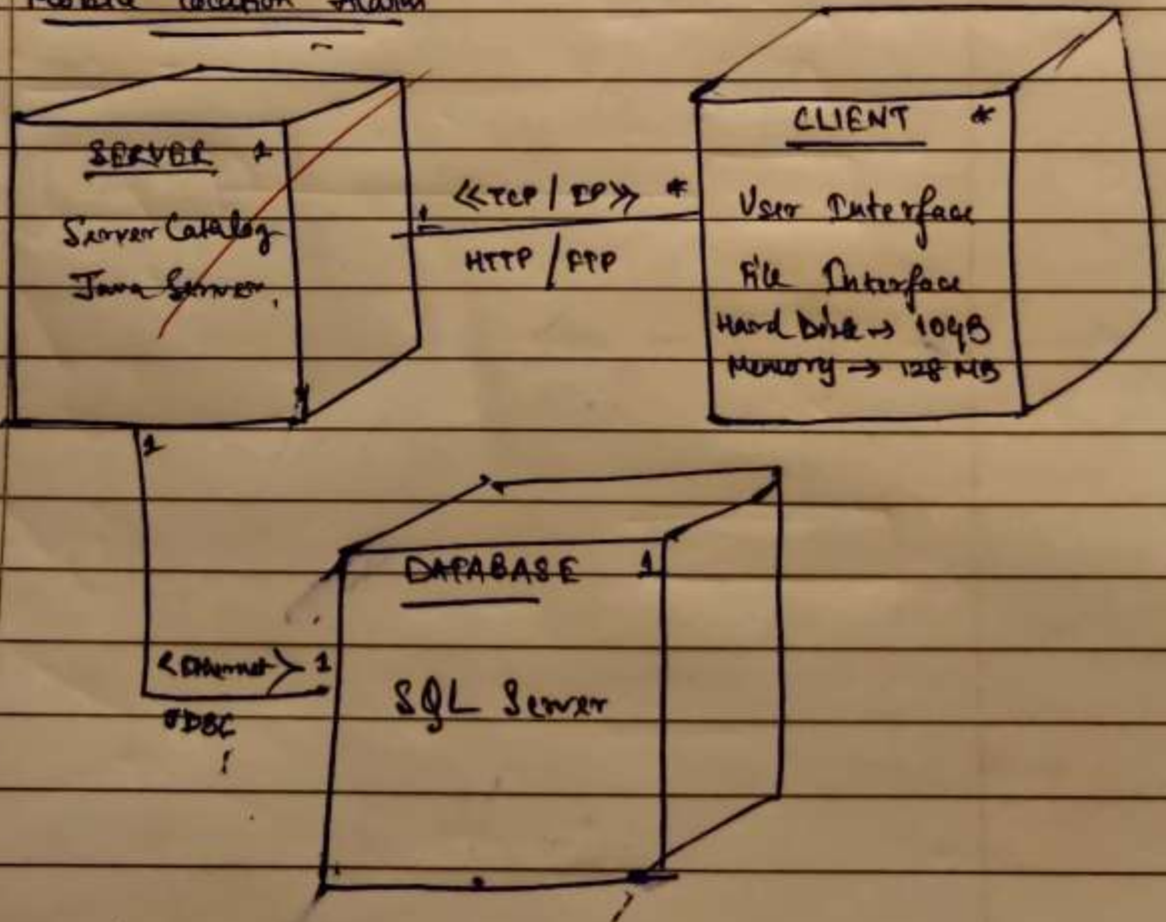
LAB - 10

Deployment Diagram

Representation:



Mobile location Alarm


SERVER 1
Server Catalog
Java Server
1
«TCP / IP» *
HTTP / FTP
CLIENT *
User Interface
File Interface
Hard Disk → 10GB
Memory → 128 MB
1
<Ethernet> 1
JDBC
DATABASE 1
SQL Server