

System Design & Planning

SUST SWE HACKATHON 2023

Submitted by

Team DU_Sylhet_Tour

Md Siam

Nazmus Sakib Ahmed

Ashraful Islam Shanto Sikder

Dardaul Hoque

Date: 27 January, 2023

Use Case Diagrams:

The high-level requirements of **!Blink** are represented with UML Use Case diagrams. The high-level critical modules are refined further along with user roles. The levels are illustrated below:

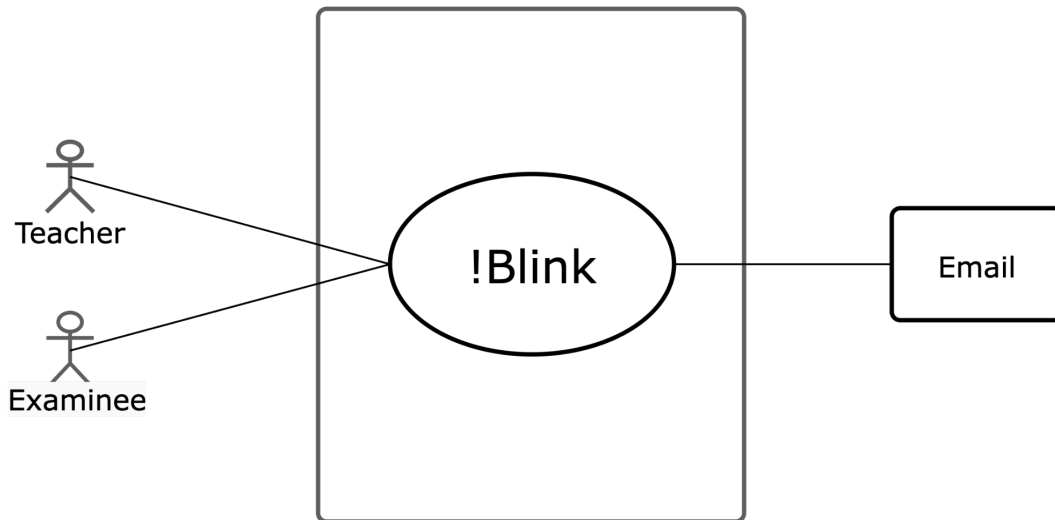


Figure 1: Use Case Diagram Level 0

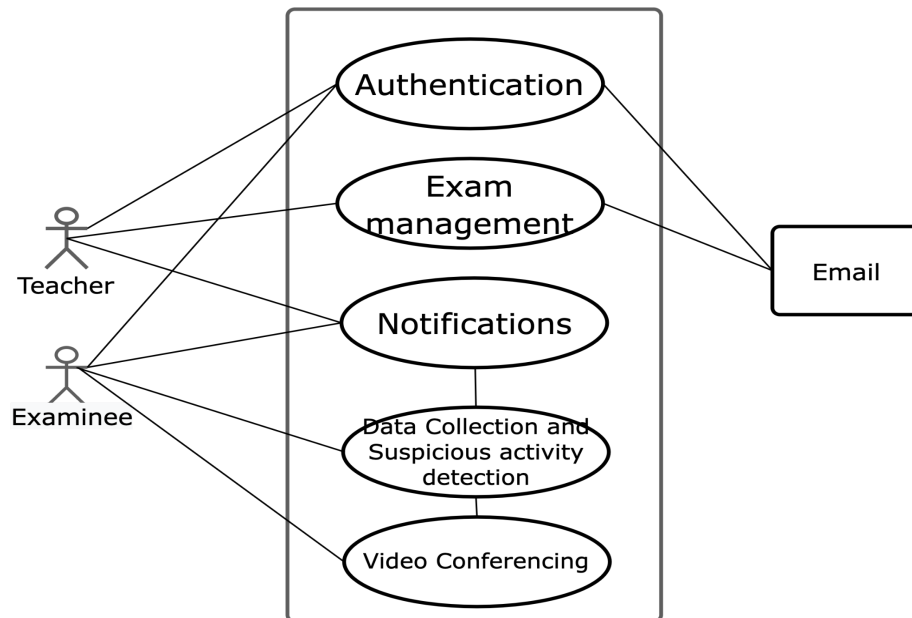


Figure 2: Use Case Diagram Level 1

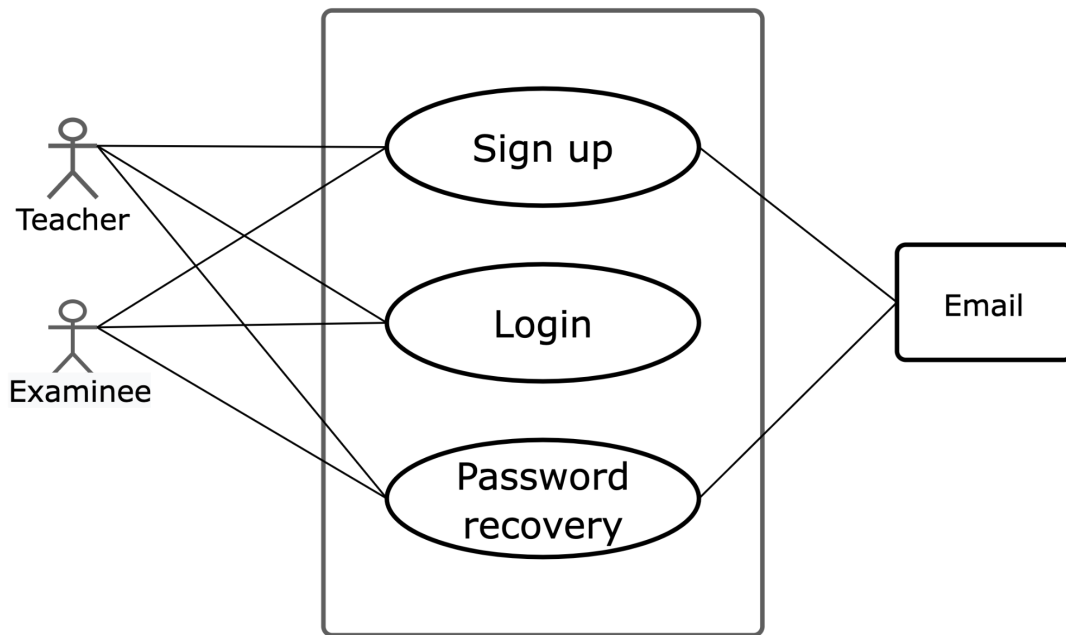


Figure 3: Use Case Diagram Level 1.1

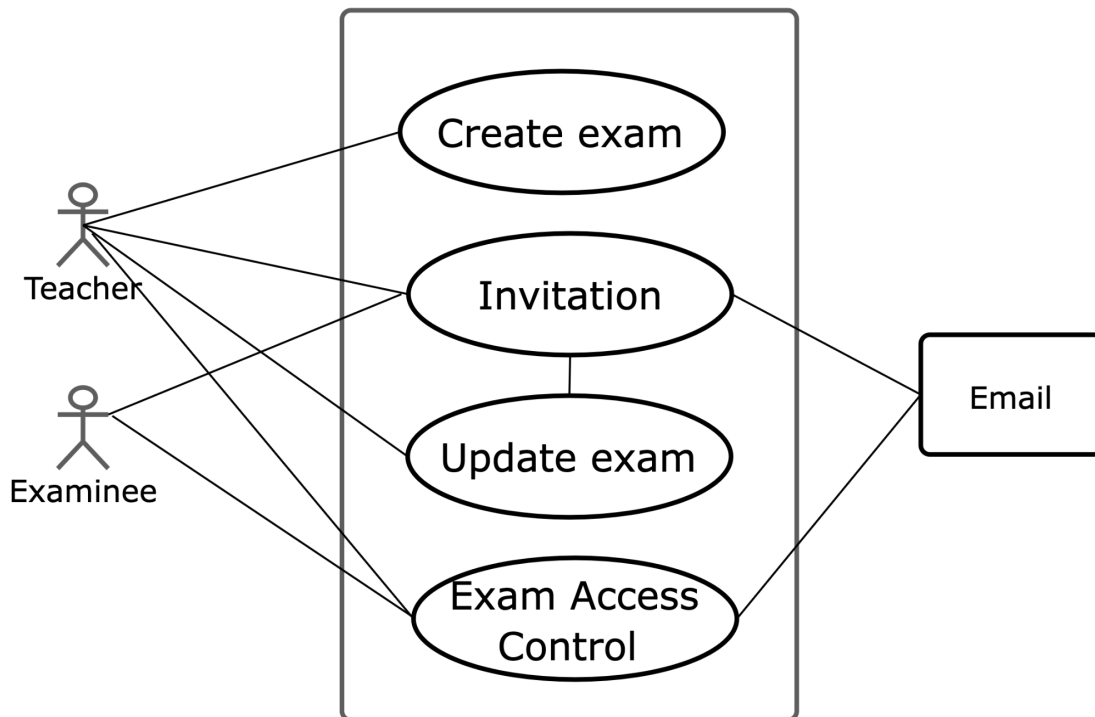


Figure 4: Use Case Diagram Level 1.2

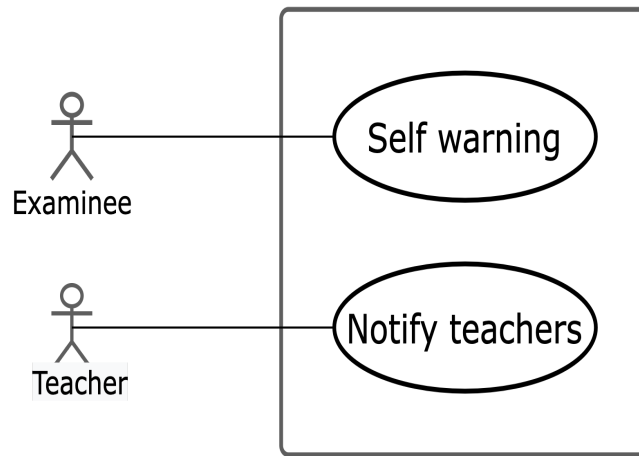


Figure 5: Use Case Diagram Level 1.3

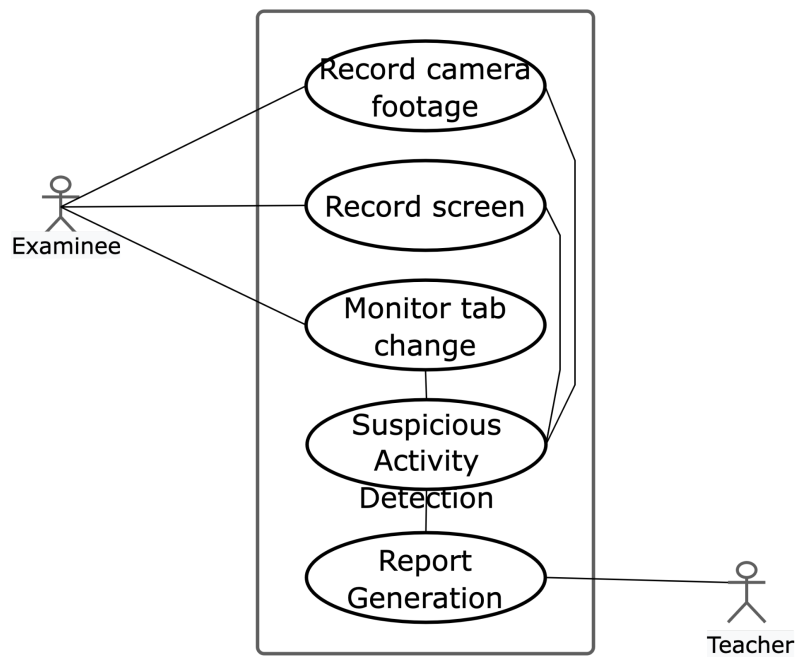


Figure 6: Use Case Diagram Level 1.4

Swimlane Diagrams:

A swimlane diagram is a type of flowchart that delineates who does what in a process. The swimlane diagram of !Blink is given below:

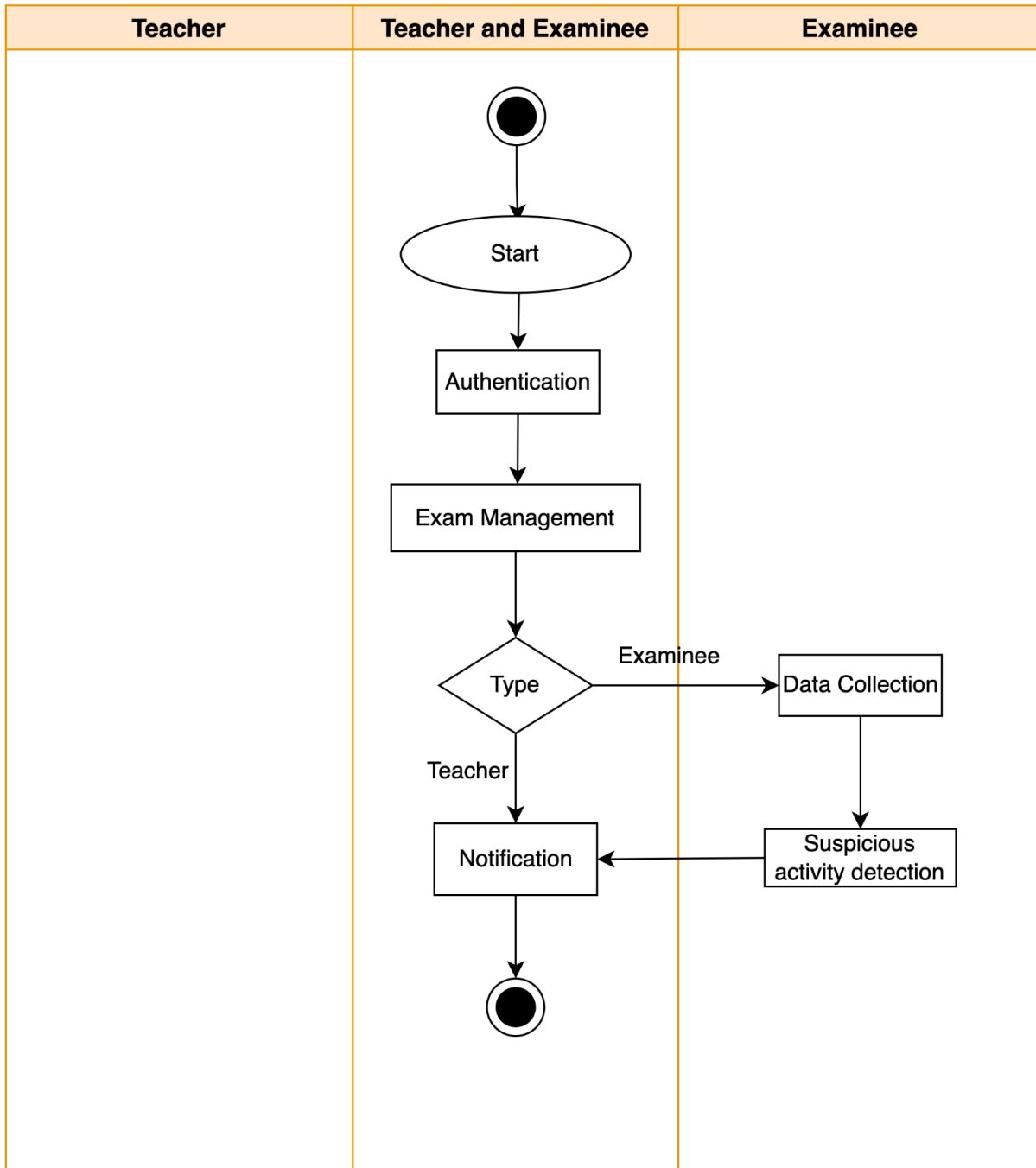


Figure 7: Swimlane Diagram of Use Case Level 1

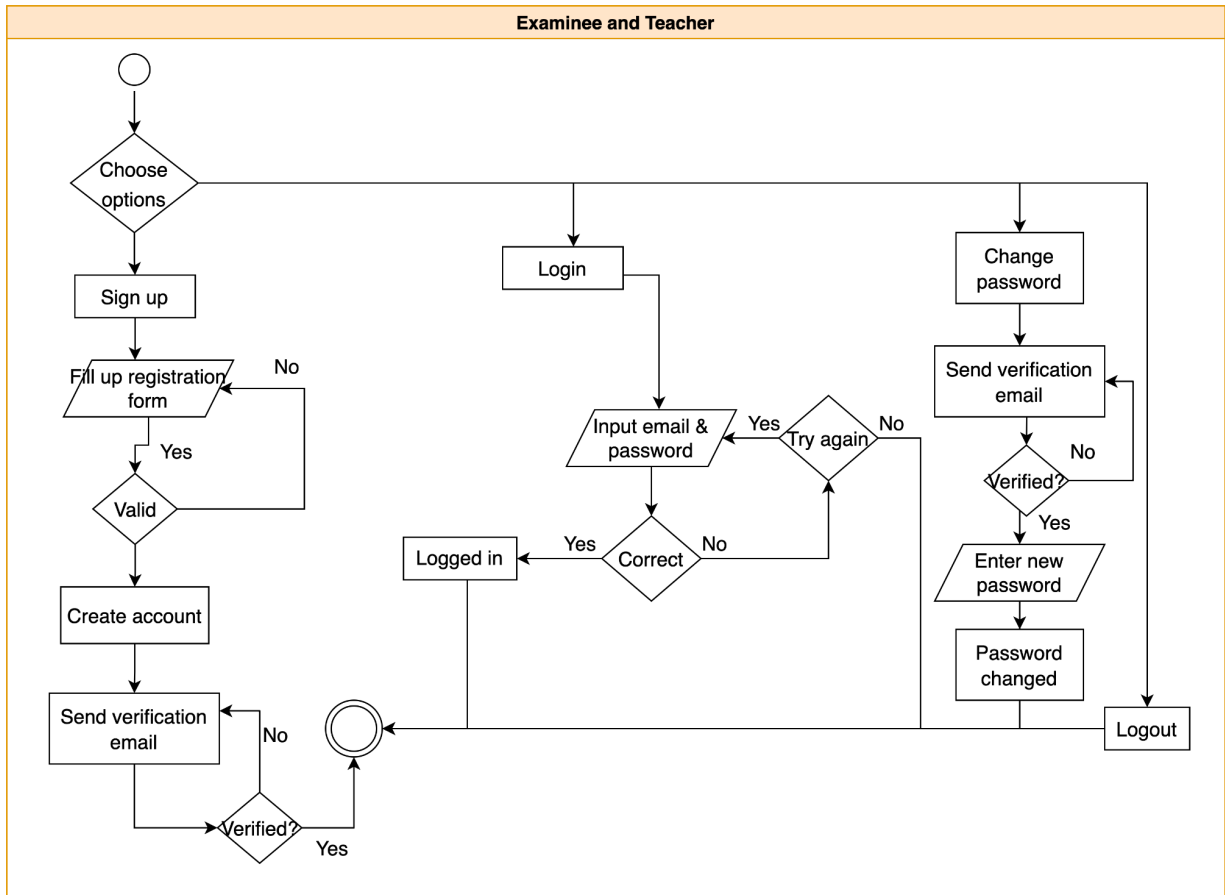


Figure 8: Swimlane Diagram of Use Case Level 1.1

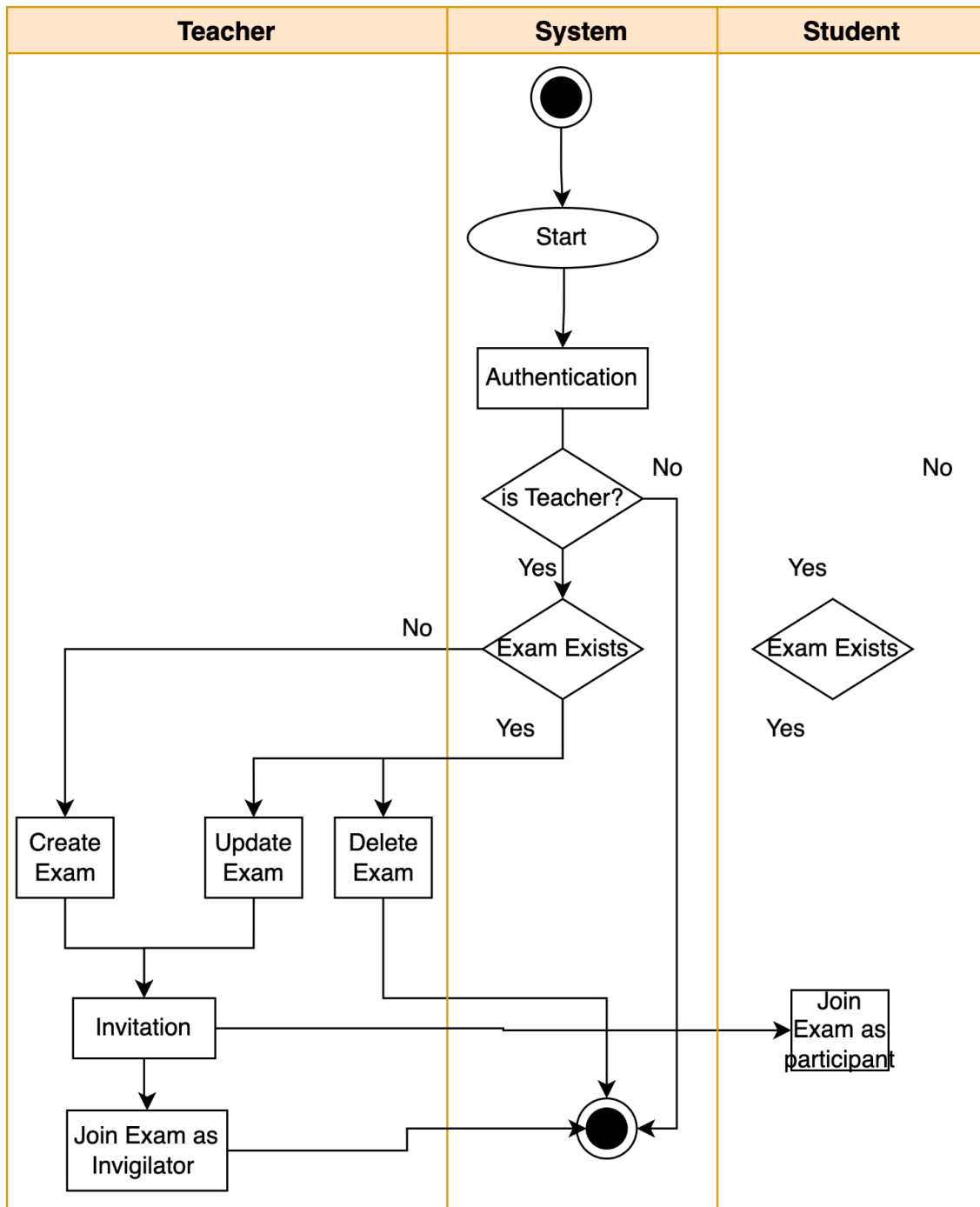


Figure 9: Swimlane Diagram of Use Case Level 1.2

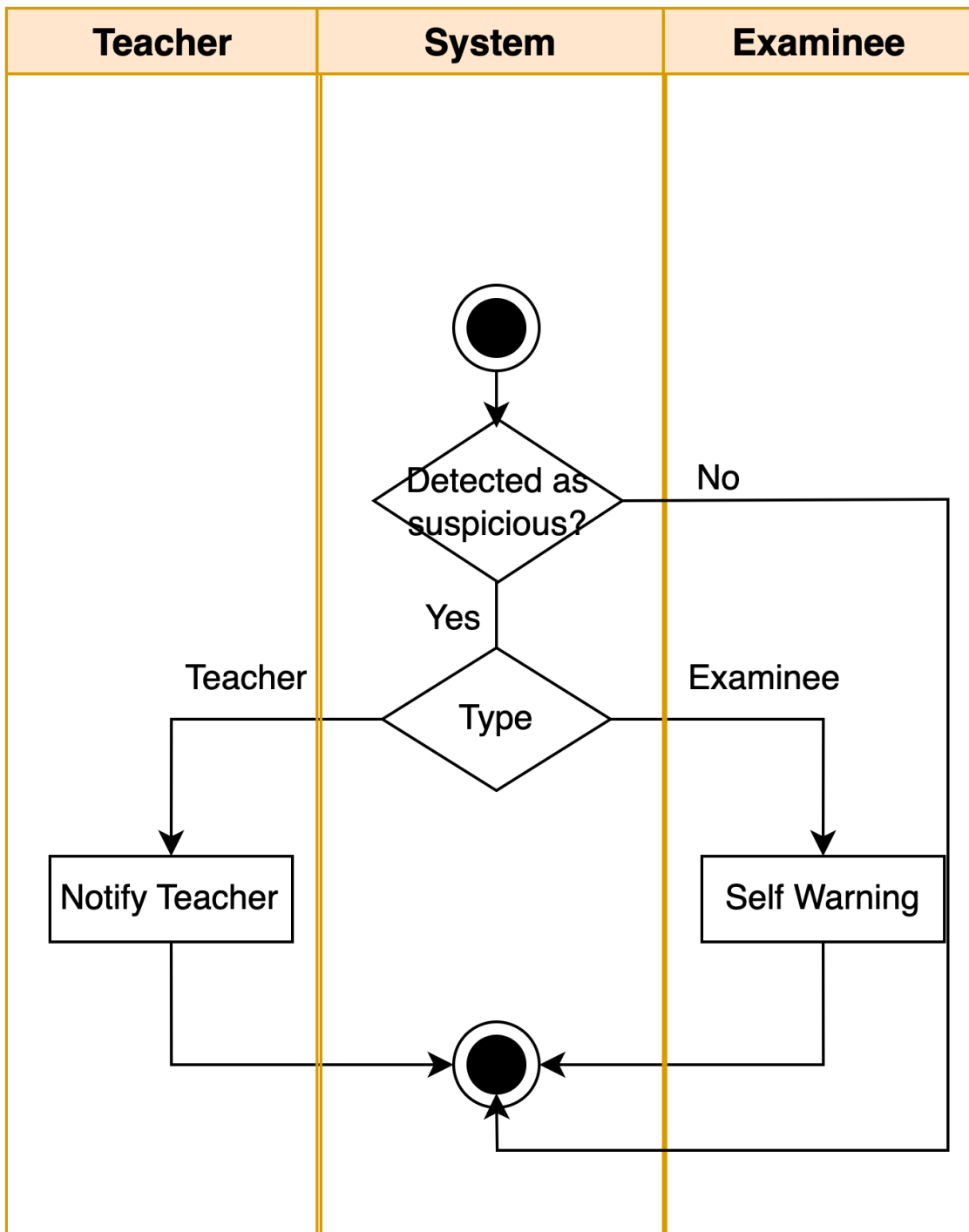


Figure 10: Swimlane Diagram of Use Case Level 1.3

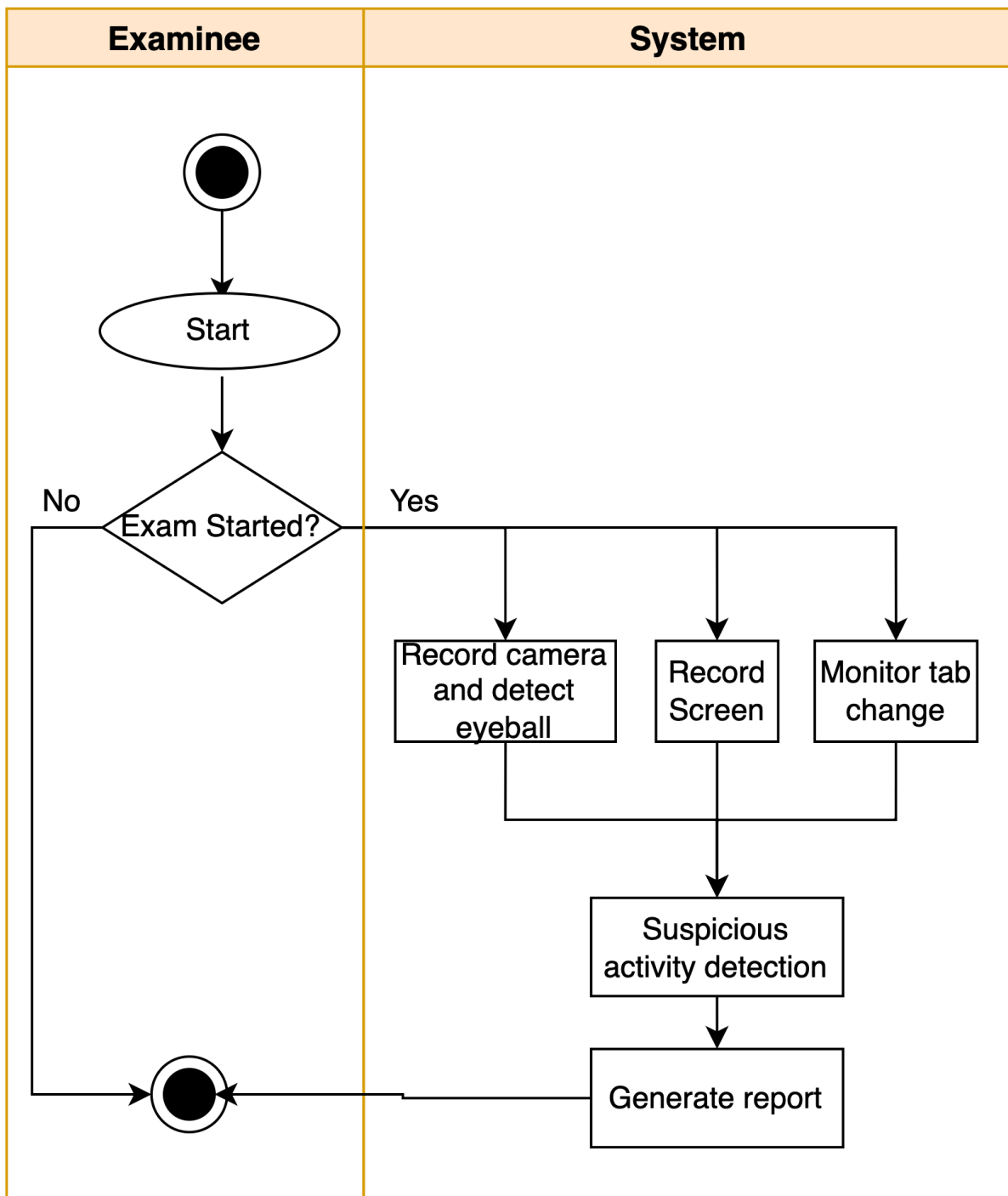


Figure 11: Swimlane Diagram of Use Case Level 1.4

Architectural Diagram:

Based on the requirements, the business logic processing can be done based on the defined architecture. It consists of broadly of -

1. Authoriser: Handles JWT token generation and validation. This will solely be responsible for authorization and user role assignment. Further the permissible operation for the obtained roles are matched with current request. Redis/any other cache storage should be used to make this frequently needed operation efficient.
2. Realtime Notification Sender: This module is responsible for sending the evidences and notifications of suspicious activities to teachers in real time.
3. DB Change Observer: The observer is notified about the event of changes in the project (receiving of new notifications). The observer further notified the Real Time Notification Sender to make changes in the estimation.
4. REST API: The REST paradigm should be used to decouple resources and make CRUD operations easier. The request is processed in the Controller and redirected to appropriate resource Repositories. After running the DB transactions the results are returned to the Client after Serialization.

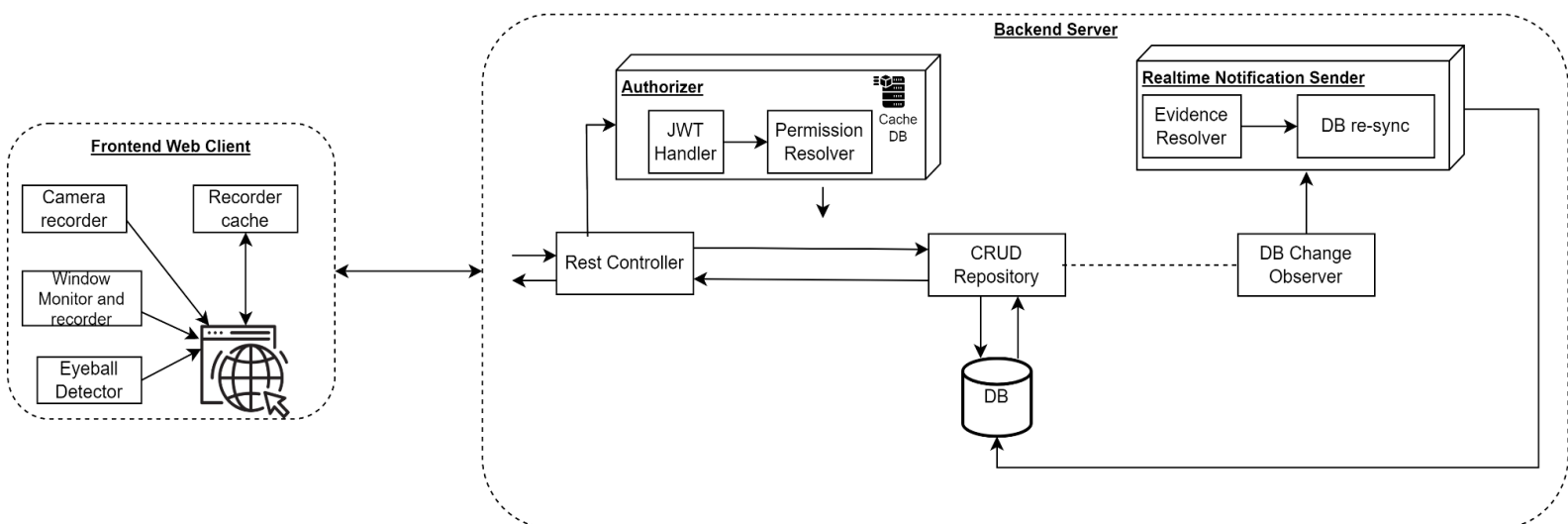
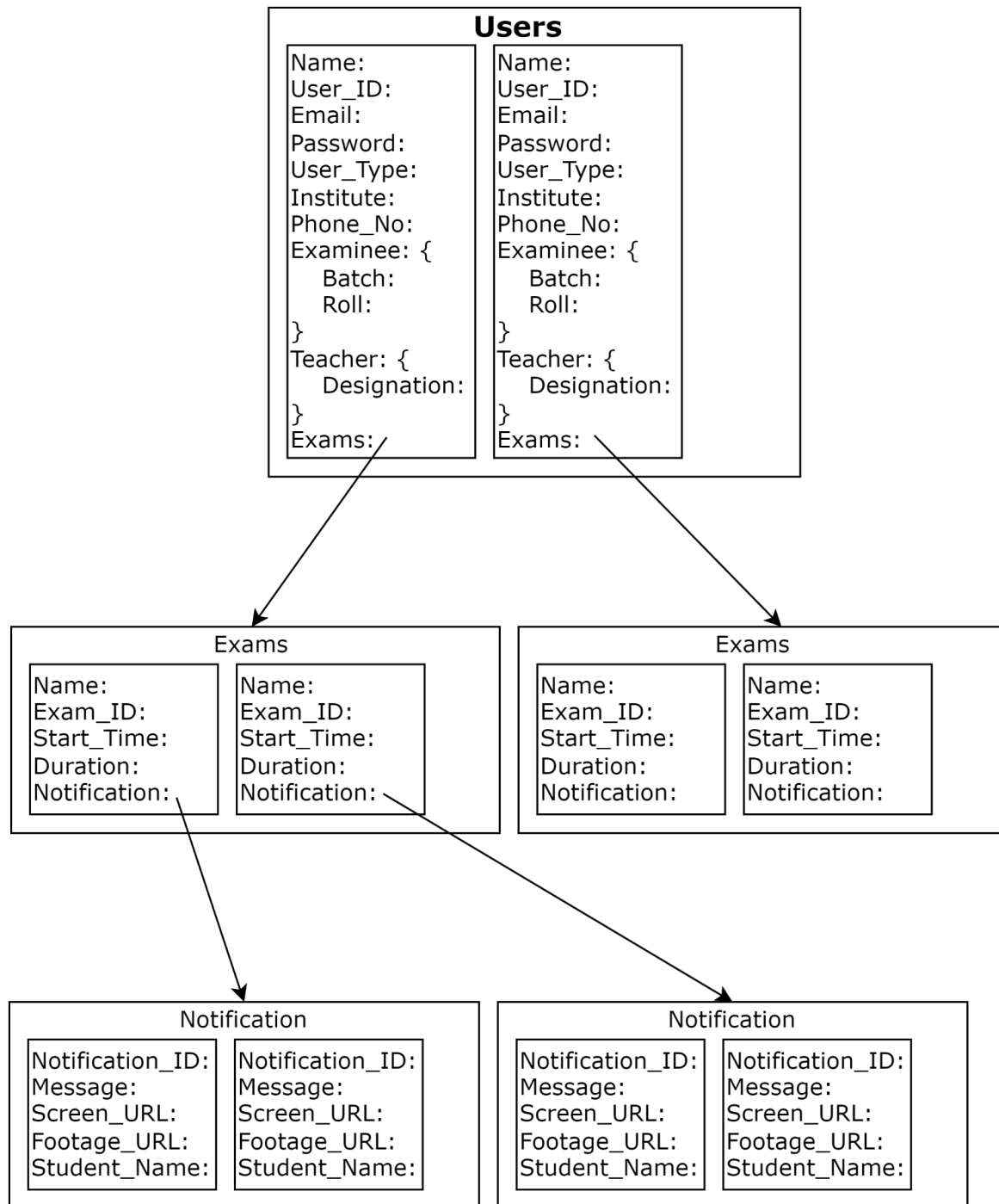


Figure 12: High Level Architectural Overview

Database Design



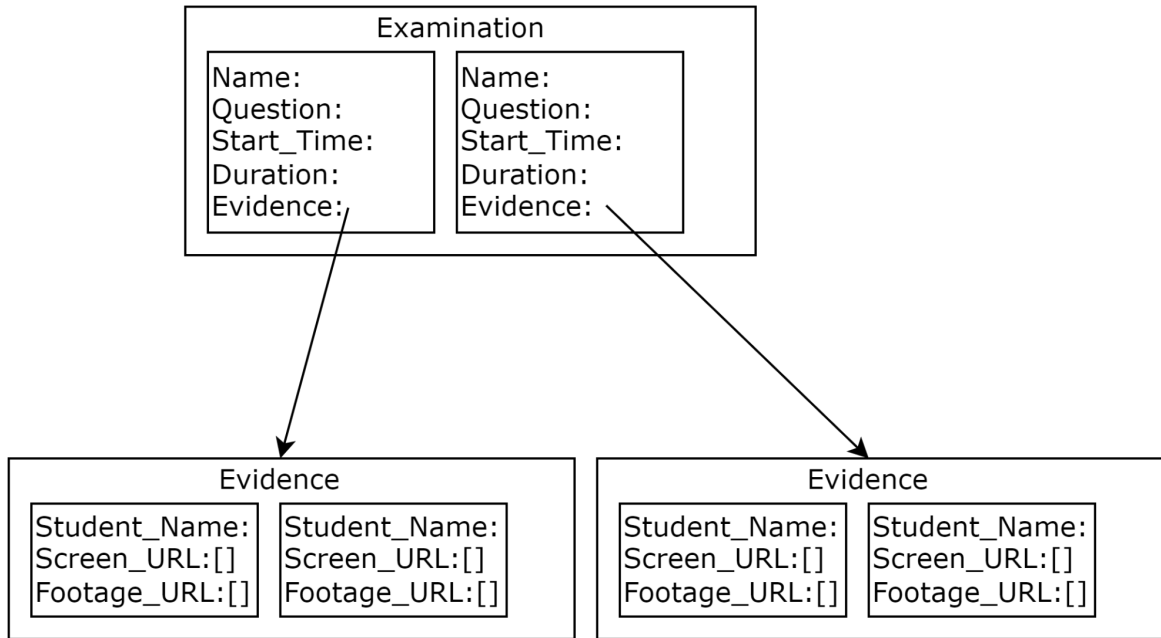


Figure 13: Non Relational Database Schema

Data flow Diagram

A data-flow diagram is a way of representing the flow of data through a process or a system.

The DFD provides information about the outputs and inputs of each entity and the process itself.

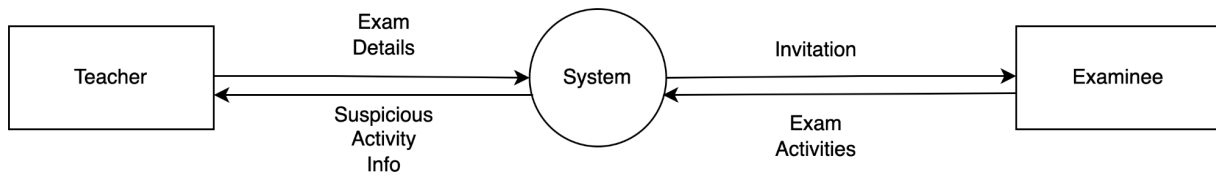


Figure 14: Level-0 DFD

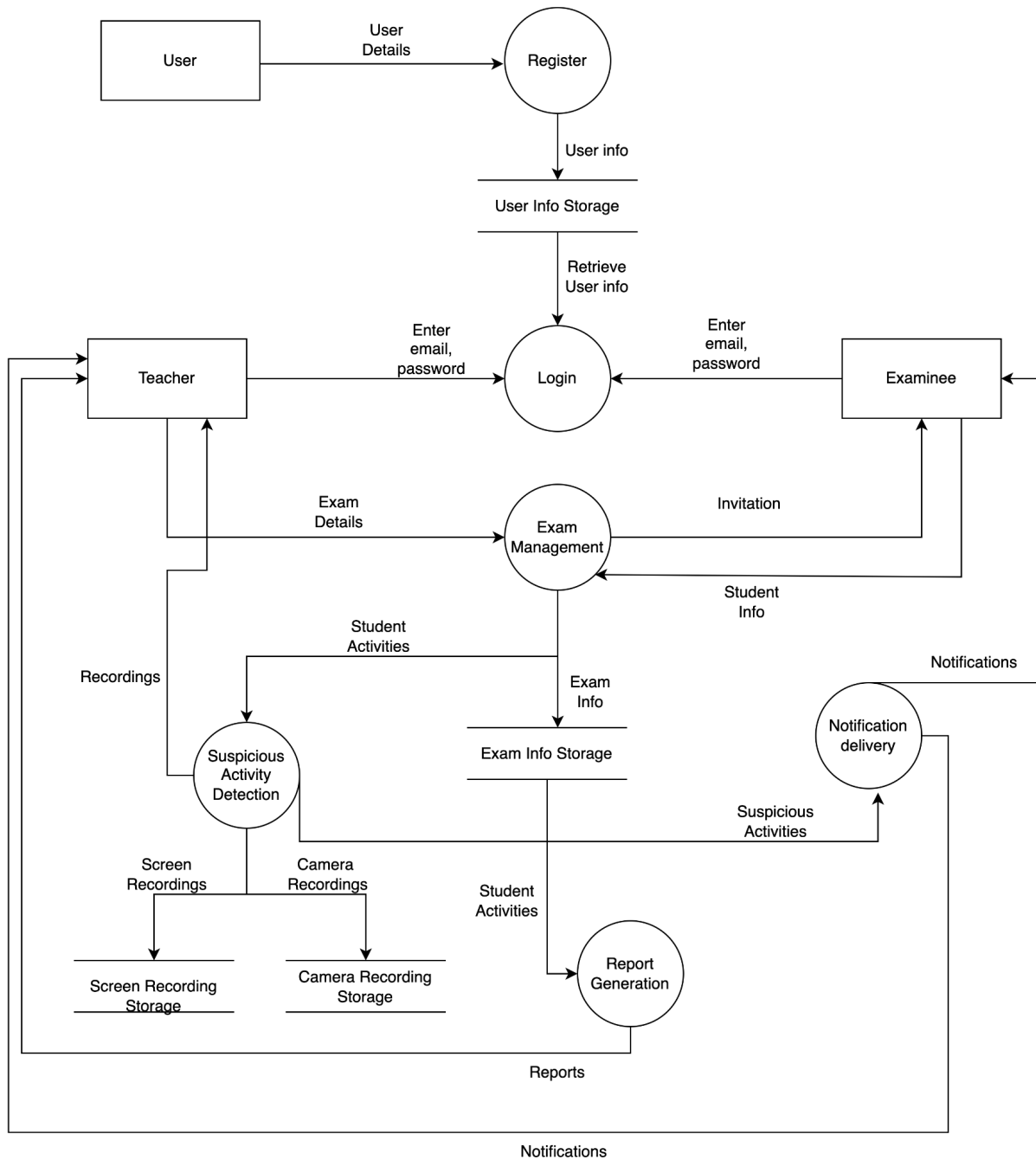


Figure 15: Level-1 DFD

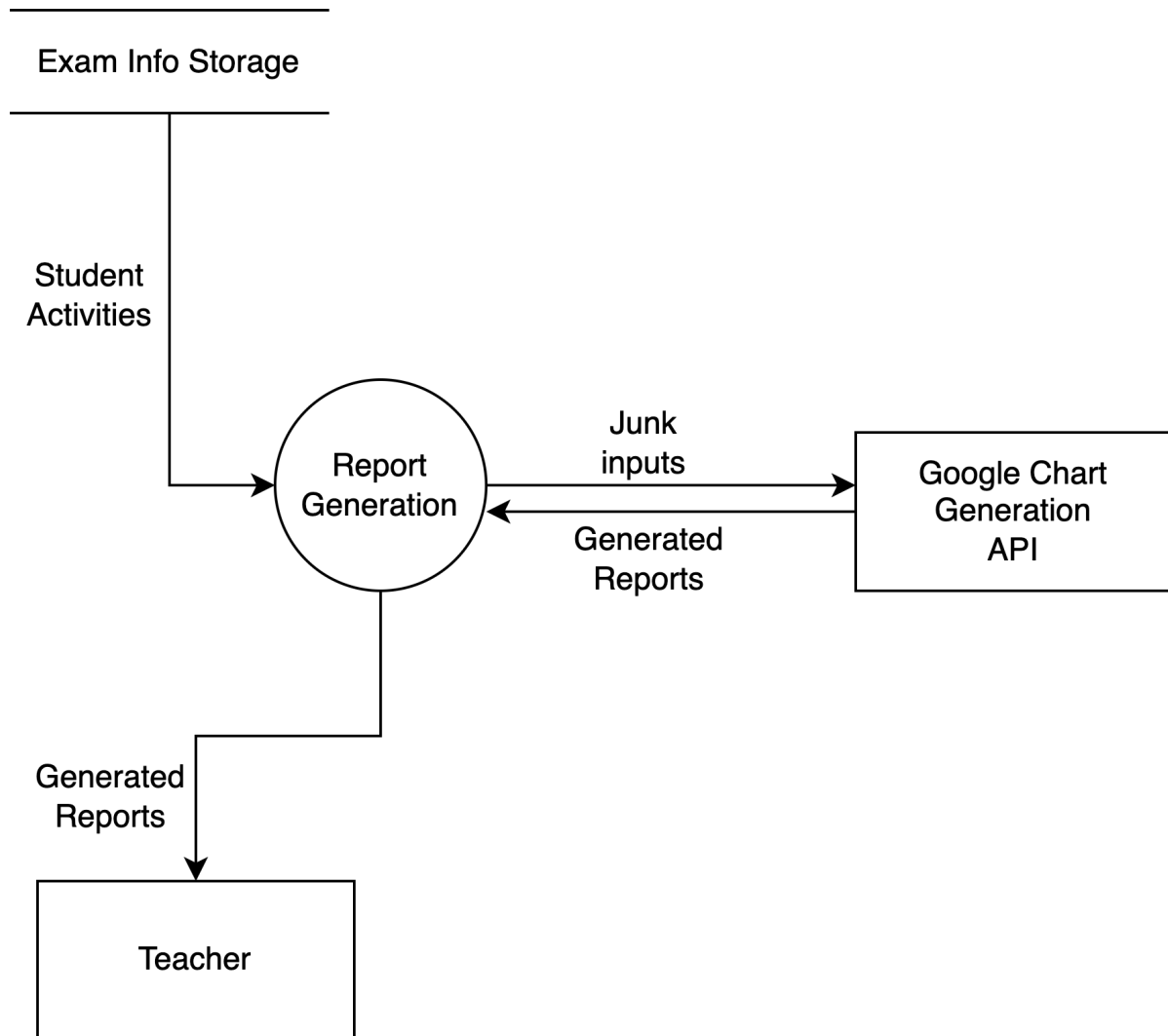


Figure 16: Level-2 DFD (Report Generation)

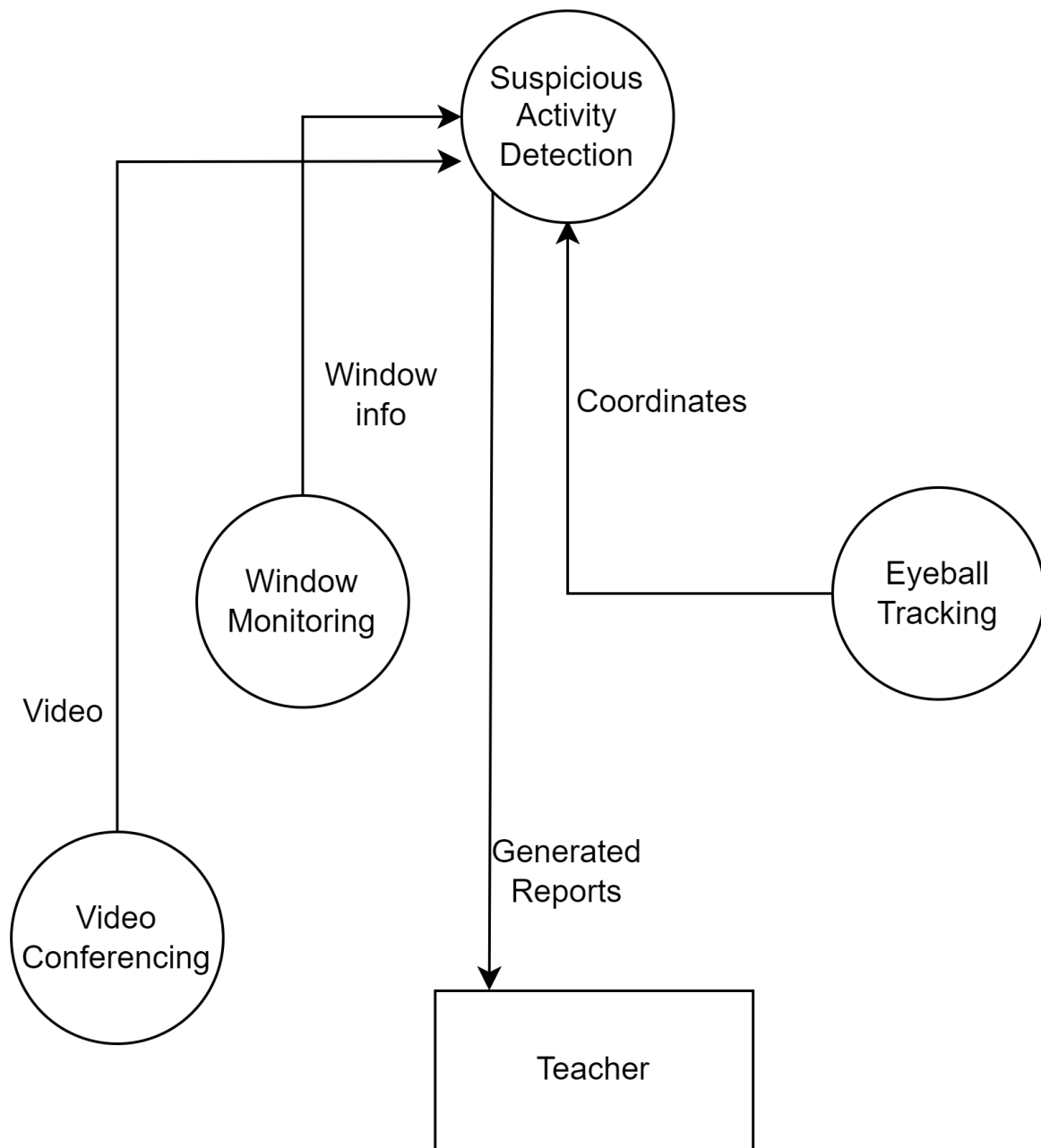


Figure 17: Level-2 DFD (Suspicious Activity Detection)

State Diagrams:

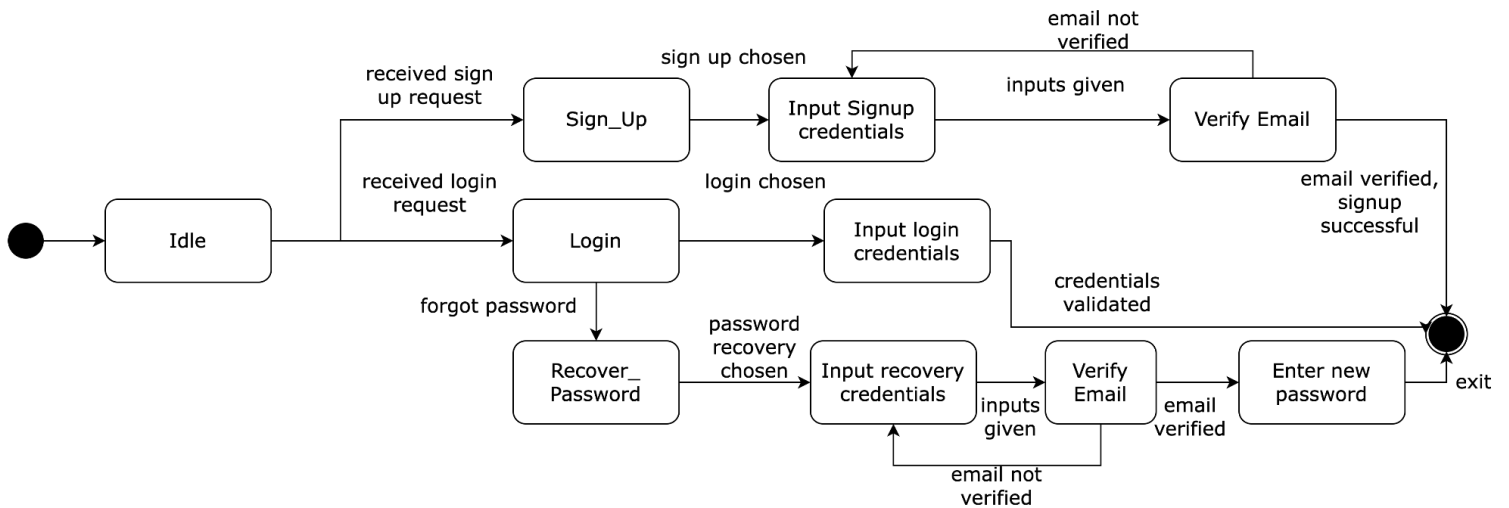


Figure 18: State diagram of User

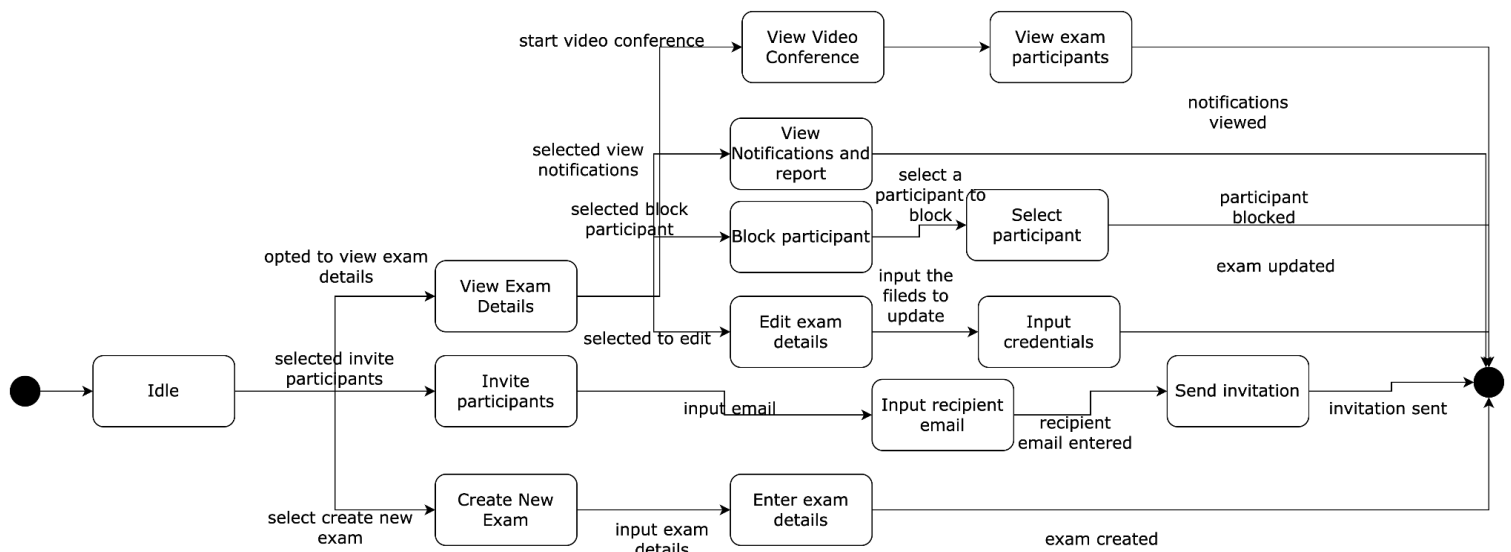


Figure 19: State diagram of Teacher

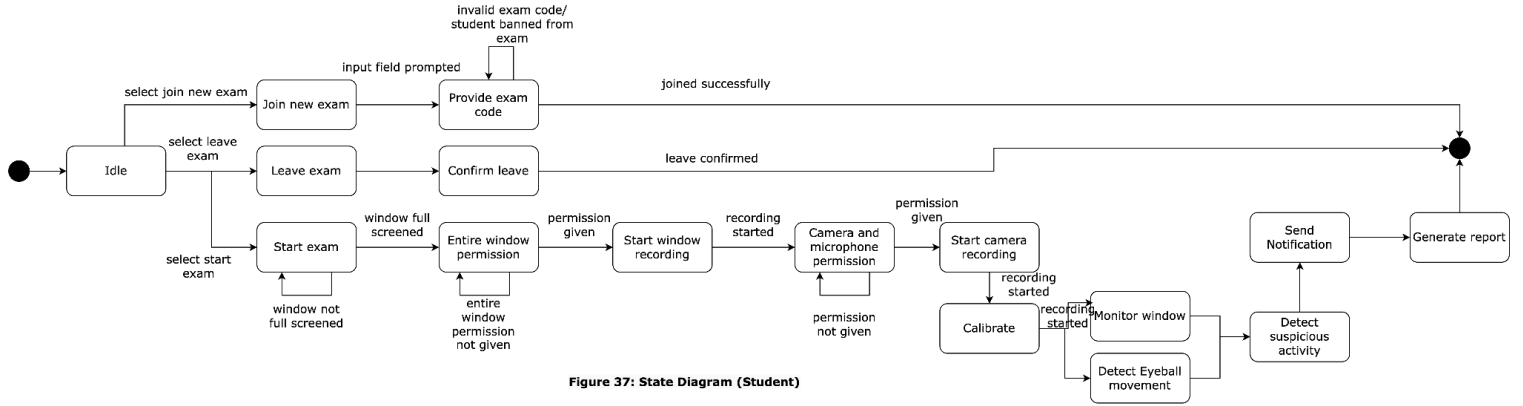


Figure 20: State diagram of Examinee