

BSSE FINAL PROJECT

Software Design Specification

INVIGILEYE



Project Advisor

Ms. Saher Zia

Presented by:
Group ID: S25SE044

Registration #

L1S22BSSE0011
L1S22BSSE0151
L1F21BSSE0238

Name

Muhammad Faizan Ashraf
Zieshan Ali
Uzair Ali

Faculty of Information Technology & Computer Science
University of Central Punjab

Software Design Specification

SDP Phase II

INVIGILEYE

Advisor: Ms. Saher Zia

Team S25SE044

Member Name	Primary Responsibility
Muhammad Faizan Ashraf	Prototype & Diagrams
Zieshan Ali	Prototype & Document
Uzair Ali	Research & Document

Table of Contents

Table of Contents	i
Revision History	ii
Previous Phases Feedback.....	ii
Abstract.....	1
1. Introduction.....	2
1.1 Product	2
1.2 Background	2
1.3 Objective(s)/Aim(s)/Target(s).....	3
1.4 Scope	3
1.4.1 In-Scope Features:	3
1.4.2 Out-Scope Features:.....	4
1.5 Document Conventions	4
2. Overall Description	5
2.1 Product Features.....	5
2.2 Functional Description	5
2.3 User Classes and Characteristics.....	6
2.4 Design and Implementation Constraints	6
2.5 Assumptions and Dependencies.....	7
2.5.1 Key Assumptions.....	7
2.5.2 Dependencies	7
3. Technical Architecture	8
3.1 Application and Data Architecture.....	8
3.2 Component Interactions and Collaborations	10
3.3 Design Reuse and Design Patterns.....	11
3.3.1 Design Reuse in InvigilEye	11
3.3.2 Design Patterns in InvigilEye	12
3.4 Technology Architecture.....	12
3.4.1 Platform & System Hosting.....	12
3.4.2 Hardware Components.....	13
3.4.3 Connectivity Requirements	13
3.4.4 Modes of Operation	13
4. Screenshots/Prototype	13
4.1 Workflow	13
4.2 Screens	14
5. Other Design Details	32
5.1 Core System Focus.....	32
5.2 Hardware Dependency	32
5.3 Reporting.....	32
5.4 Limitations	32
6. Revised Project Plan	33
7. References	33
8. Appendix A: Glossary.....	34
Appendix B: IV & V Report	35

Revision History

Name	Date	Reason For Changes	Version

Previous Phases Feedback

Idea Defense Feedback (Screenshot)

Abstract

InvigilEye is an AI-powered invigilation system for physical examinations. It enhances exam integrity and fairness through facial recognition for automated attendance and unauthorized face detection, posing estimation to flag suspicious behavior and instant alert to invigilators. The system will also be able to save snapshots (visual evidence) for post exam reports, while admins will be able to configure exam parameters (e.g., venues, students' data etc.) and be able to download reports. InvigilEye reduces 70-80% of manual workload by applying ML models and integrating cameras with desktop interface, ensures unbiased monitoring and provides insights-modernizing exam security from student authentication to detailed reporting.

1. Introduction

1.1 Product

InvigilEye is basically an AI-powered invigilation system mainly designed for physical examinations. It majorly enhances exam integrity and fairness through advanced facial recognition, which automates attendance and also detects unauthorized faces in real time. The system also uses pose estimation to flag overall suspicious behavior and sends quite instant alerts to invigilators, ensuring very timely action during exams.

With the ability to simply save snapshots as visual evidence, InvigilEye also supports post-exam analysis and reporting. Admins can easily configure essential exam parameters, such as venues and student data, and download detailed reports whenever they are needed.

InvigilEye reduces about 70–80% of manual workload just by integrating machine learning models with a camera-enabled desktop interface. This ensures unbiased, consistent monitoring and provides highly valuable insights—modernizing exam security from overall student authentication to complete reporting. It's a really smart solution for institutions seeking a highly reliable and tech-driven approach to physical exam supervision.

1.2 Background

1. Online Proctoring Platforms

- [1]**Proctorio** mainly offers an AI-powered, browser-based monitoring system designed for conducting remote exams. It generally supports features like identity verification and screen tracking.
- [2]**ProctorU** combines live and automated proctoring with core functionalities such as ID checks and session recording for all sorts of online assessments.

Limitations: Both platforms are simply limited to online exams only. They offer absolutely no support for physical exam halls, lack pose estimation to analyze behavior, and cannot even generate real-time alerts within in-person environments.

2. Traditional and CCTV-Based Invigilation

- **Manual Supervision:** Heavily depends on invigilators that are physically monitoring the exam hall, which is highly prone to human fatigue, limited coverage, and unintentional oversight.
- **CCTV Monitoring:** While some institutions use some sort of fixed cameras during exams, these setups mainly lack AI integration. As a result, they cannot automatically detect cheating behaviors or generate instant reports or alerts.

How InvigilEye Differs:

InvigilEye is developed specifically for physical examination scenarios. It simply combines facial recognition for attendance, pose estimation to detect cheating behaviors, and real-time alerts for invigilators. Additionally, it generates detailed automated reports with timestamped visual evidence. Unlike earlier solutions, it offers a pretty complete package for in-person exam monitoring, using AI to enhance overall accuracy, reduce workload, and improve academic integrity.

1.3 Objective(s)/Aim(s)/Target(s)

The aim of this project is to build a system that:

1. cut down manual invigilation work by 70–80%.
2. automatically detects and highlights suspicious activities like passing notes or unusual movements during exams.
3. automatically verifies attendance and identifying unauthorized students during exams.
4. sends instant real-time alerts to invigilators during any sort of suspicious actions.
5. helps the examination department by providing post-exam reports highlighting attendees and exam details.
6. provides an interactive desktop-based dashboard for both admins and invigilators where invigilators can easily monitor live alerts, call UMC, ask for material (e.g., extra sheets), and review exam reports.

1.4 Scope

The following features which are in-scope and out-scope of InvigilEye are:

1.4.1 In-Scope Features:

- **Automate Attendance:** It automatically verifies the student's attendance during exam using facial recognition system.
- **Cheating Detection:** It determines suspicious activity through pose estimation they may include cheating.
- **Alert System:** It alerts the invigilators about suspicious activities and about unknown students.
- **Snapshot Capturing:** It automatically captures the snapshots of suspicious actions for review and evidence.
- **Admin Dashboard:** It provides features for admins to login, set venues, manage exam sessions, and students' records.
- **Invigilator Dashboard:** It also provides features of invigilators to login, ask for UMC and demand material.
- **Exam Report:** It automatically generates downloadable exam and attendance report.

1.4.2 Out-Scope Features:

- Support for web/mobile platform.
- Integration with LMS.
- Grading System.

1.5 Document Conventions

The abbreviations used in this document are given below:

AI	Artificial Intelligence
GPU	Graphics Processing Unit
ML	Machine Learning
SDS	Software Design Specification
RTSP	Real Time Streaming Protocol
OpenCV	Python library of face scanning
YOLOv8	You Only Look Once Version 8 (Python library for pose estimation)
UMC	Unfair Means Case
PDF	Portable Document Format

CSV	Comma Separated Values
ERD	Entity Relationship Diagram

Table 1: Document Conventions

2. Overall Description

2.1 Product Features

InvigilEye is specifically an AI-based system mainly designed to enhance the overall physical exam monitoring. It automatically marks student attendance using facial recognition and also detects cheating through real-time pose estimation. The system sends highly instant alerts to invigilators upon detecting any sort of suspicious behavior and captures snapshots as visual evidence.

After each exam, it simply generates downloadable PDF reports containing attendance data and incident records. Admins can also manage exam setups, student data, and system configurations through a dedicated dashboard, while invigilators mainly use their dashboard to monitor alerts, request materials, and access reports.

2.2 Functional Description

InvigilEye is an AI-powered system for physical exam invigilation. It ensures exam integrity through automated processes.

2.2.1.1 Core Functions

1. Facial Recognition

- Automates attendance by verifying students.
- Detects and alerts invigilators about unauthorized faces.

2. Pose Estimation

- Flags suspicious behaviors like unusual movements.
- Sends instant alerts to invigilators.

3. Evidence & Reporting

- Saves snapshots of flagged incidents as visual proof.
- Generates post-exam reports with timestamps.

4. Admin Controls

- Configures exam settings (venues, student data).
- Provides downloadable reports for record-keeping.

5. Desktop Interface

- Integrates with cameras for live monitoring.
- Displays alerts and allows invigilator actions.

2.3 User Classes and Characteristics

The system is designed for the following user types:

1. Invigilators

- **Use:** Invigilators will be able to handle alerts about unverified or cheating students. They are also able to download exam reports, demand material and call for UMC.
- **Expertise:** Moderate technical knowledge.

2. Admin

- **Use:** Admin will be able to manage examinations. They upload students' data, set up exam details (such as venue and timing) and review post-exam reports and snapshots. Admins can also respond to material demand and UMC, raised by invigilators.
- **Expertise:** Moderate technical skills.

3. Students (Passive Users)

- **Use:** No direct interaction with the system - their faces and behavior are analyzed for verification and detection
- **Expertise:** Not required.

The most critical users to satisfy are invigilators and admins, as they rely on the system for smooth and effective monitoring during exams.

2.4 Design and Implementation Constraints

Here are the risk factors which are holding us back (and how we will oversee it):

- Bad Lighting:

Challenge: Bad Lighting might affect the accuracy of the system – might face difficulty in scanning student's faces.

Solution: Use WDR cameras + Train models on low light data as well.

- Budget Issues:

Challenges: Schools cannot afford expensive cameras and GPUs.

Solution: Support affordable (mid-range) cameras and consumer GPUs.

- Privacy Concerns:

Challenges: Students' trust regarding their privacy.

Solution: No raw images are saved, and snapshots can be deleted easily with a one-click option.

2.5 Assumptions and Dependencies

2.5.1 Key Assumptions

It is highly assumed that all invigilators who will be using this system will be trained before using our InvigilEye system by simply reading the overall user manual and watching a few tutorial videos. The exam environment is mainly expected to have a highly stable power supply and a properly working Wi-Fi connection. Cameras should be correctly installed at the recommended heights and positions to simply ensure that our system works 100% correctly.

2.5.2 Dependencies

The system mainly depends on IP cameras that also support RTSP, and such computers with NVIDIA GPUs to run all the AI models smoothly. It will use all these software tools like Python, OpenCV, and YOLOv8 for its main functions. The system also majorly requires overall access to the school's database for all important student information and a reliable internet connection to work properly.

Technical Architecture

2.6 Application and Data Architecture

2.6.1 Class Diagram

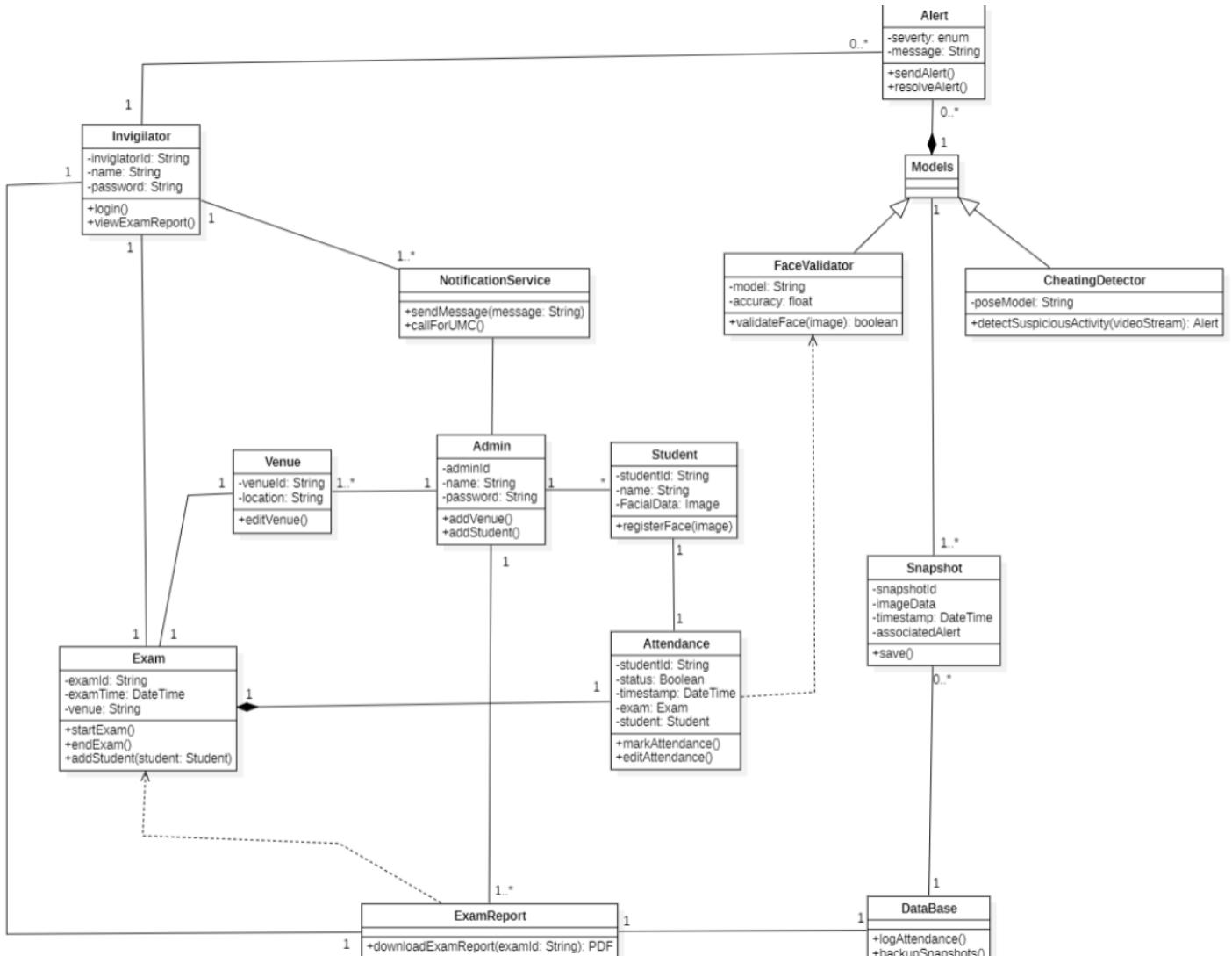


Fig.1: Class Diagram

2.6.2 Entity Relationship Diagram

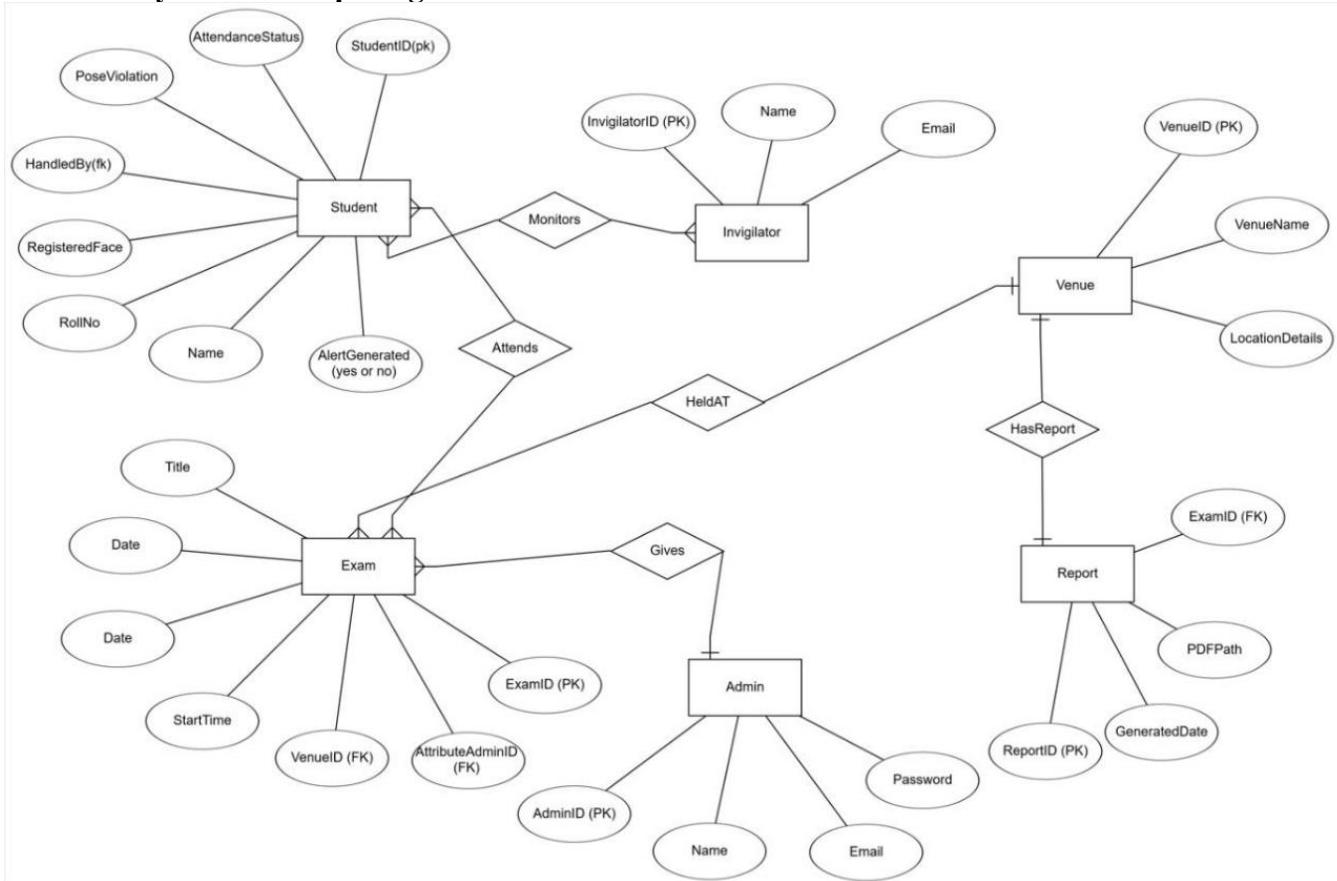


Fig.2: ERD

2.7 Component Interactions and Collaborations

2.7.1 Data Flow Diagram

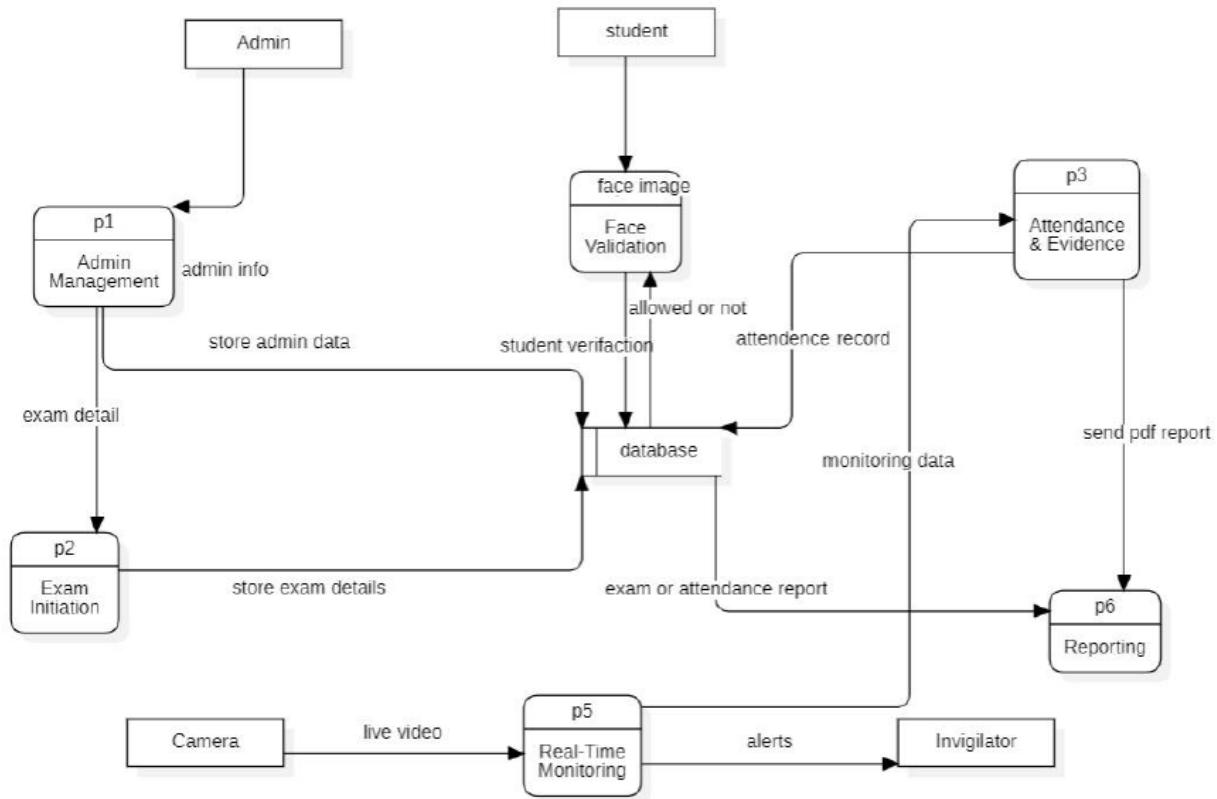
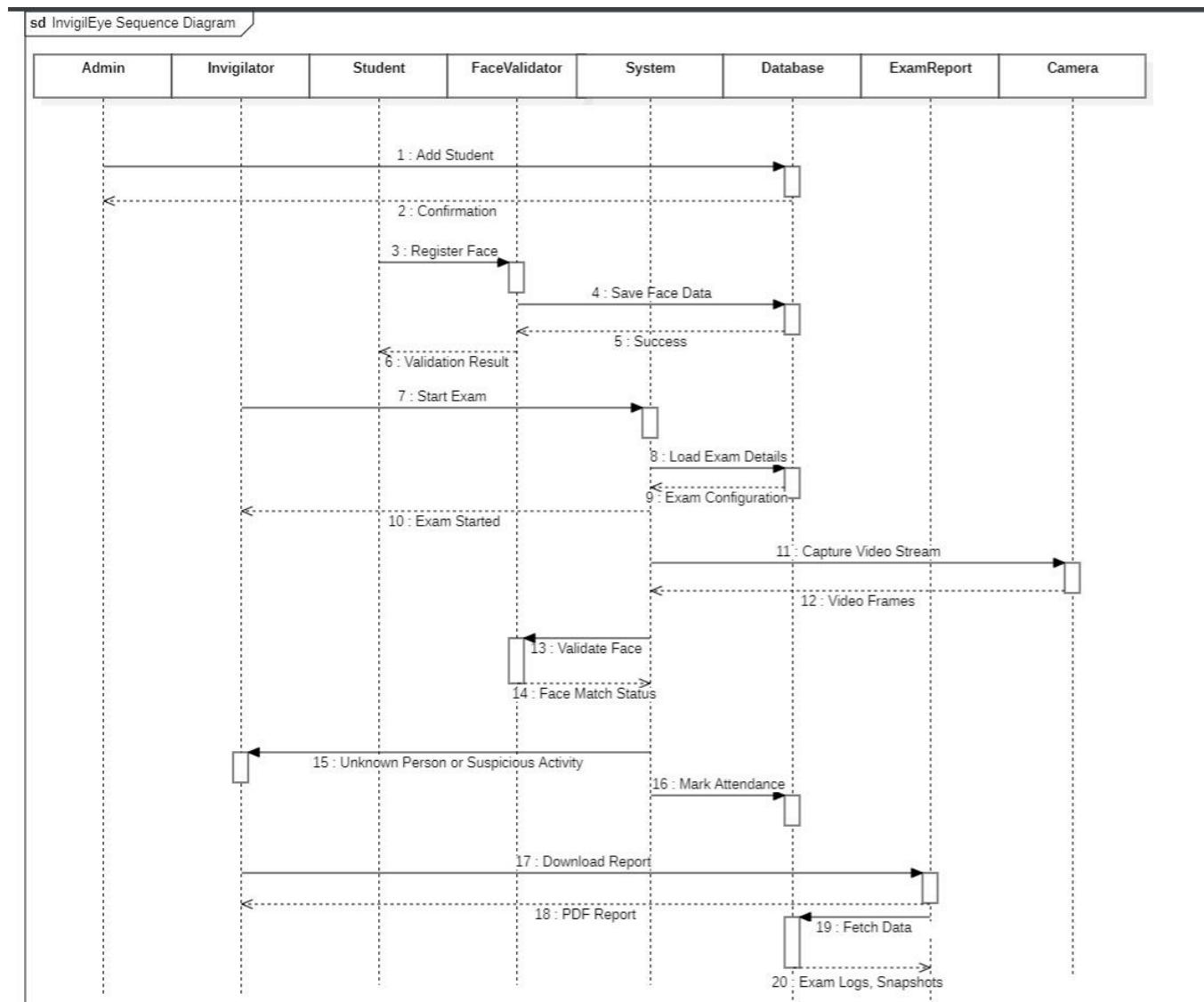


Fig.3: Data Flow Diagram

2.7.2 Sequence Diagram



2.8 Design Reuse and Design Patterns

2.8.1 Design Reuse in InvigilEye

The system utilizes the following design reuse:

1. Facial recognition technology is reused for both:
 - o Automated student attendance
 - o Unauthorized face detection

2. Pose estimation technology is reused for:
 - Detecting suspicious behaviors
 - Triggering real-time alerts
3. The alert system is reused for:
 - Unauthorized face notifications
 - Suspicious behavior notifications
4. The reporting module is reused for:
 - Live monitoring data
 - Post-exam reports

2.8.2 Design Patterns in InvigilEye

The system inherently uses these patterns based on described functionality:

1. Observer Pattern:
 - For real-time alert notifications to invigilators
 - When unauthorized faces or suspicious poses are detected
2. Singleton Pattern:
 - For the admin configuration interface
 - Ensures single control point for exam parameters
3. Facade Pattern:
 - Simplifies the complex interaction between:
 - i. Camera systems
 - ii. AI processing modules
 - iii. User interface

2.9 Technology Architecture

2.9.1 Platform & System Hosting

- **Desktop Application** – Primary interface for invigilators/admins (Windows/macOS/Linux support implied).
- **Local Server/On-Premises Hosting** – Processes facial recognition, pose estimation, and alert generation.

2.9.2 Hardware Components

- **Cameras** – Integrated for real-time video feed in exam halls.
- **Workstation/PC** – Hosts the AI models and desktop application.

2.9.3 Connectivity Requirements

- **Local Network** – Connects cameras to the processing workstation.

2.9.4 Modes of Operation

- **Real-Time Monitoring** – Continuous analysis of exam hall footage.
- **Alert Mode** – Instant notifications to invigilators upon detection.
- **Report Generation** – Post-exam evidence compilation.

3. Screenshots/Prototype

3.1 Workflow

1. Exam Setup (Admin)

- Admin logs in, sets up a new exam and enters details (venue, time, date, invigilator etc.).
- Data is stored in the database with preload facial images of students.

2. Exam Initiation (Invigilator)

- The invigilator logs in, sees the exam time and date, and starts the exam by clicking on ‘start’ button.
- The system activates the camera and starts video streaming.

3. Face Recognition

- Students sit at their seats; the system starts capturing their faces for attendance.
- The system starts matching captured faces with stored facial images.
- The system flags unauthorized or unmatched faces.

4. Monitoring

- The system starts monitoring students and pops up red alerts on screen on suspicious students.
- For evidence, the system stores the snapshot of suspected students.

5. Communication

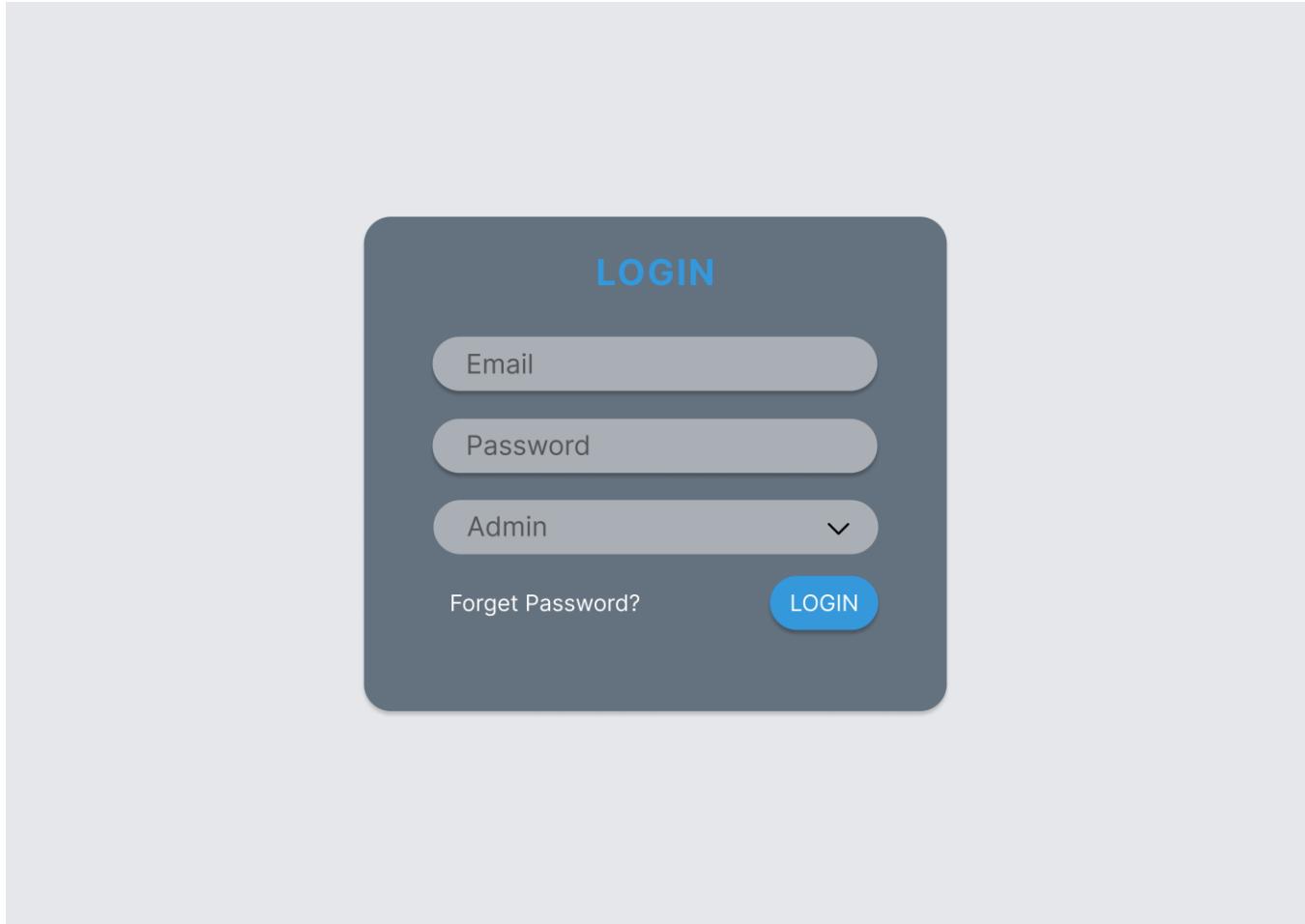
- Invigilators can call the UMC or ask for help/material through one way messaging with admin.

6. Post Exam Reporting

- Admin and invigilator can see attendance record and snapshots by clicking on ‘Details’ button of respected exam.

3.2 Screens

Login Page for Admin:



InvigilEye Admin Dashboard:

InvigilEye Admin Dashboard

[Sign Out](#)

Admin Dashboard - InvigilEye

Create Exam

Setup new exam session with invigilator and student info.

View Exams

Check, edit or delete any previously created exam.

View UMC Reports

Review unfair means cases with attached evidence.

View Requests

See invigilator requests for UMC, IT or materials.

Download Reports

Download exam attendance and activity logs.

IT Requests

Manage technical issues reported by invigilators.

Create New Exam:

The screenshot shows a web-based application for creating a new exam. At the top, there's a blue header bar with a back arrow labeled "Dashboard", the title "Create New Exam" in the center, and a red "Sign Out" button on the right. Below the header is a large white input form with rounded corners. The form contains several input fields with placeholder text and validation examples:

- Course Name:** Placeholder: e.g. Data Structures
- Course ID:** Placeholder: e.g. BSSE2311
- Room No:** Placeholder: e.g. Lab 304
- Invigilator Name:** Placeholder: e.g. Sir Ali
- Invigilator Email:** Placeholder: e.g. ali@example.com
- Section:** Placeholder: e.g. B
- Date & Time:** Placeholder: mm/ dd/ yyyy , -- : -- -- (with a small calendar icon)
- Upload CSV (Student List):** A file input field showing "Choose File" and "No file chosen".

At the bottom right of the form are two buttons: a grey "Cancel" button and a blue "Create Exam" button.

View & Manage Exams

INVIGILEYE

← Dashboard

View & Manage Exams

Sign Out

Search exam by course name or ID...

Exam Name	Course ID	Section	Date & Time	Actions
Mid Term - Data Structures	BSSE2311	N5	2025-06-22 10:00 AM	<button>View</button> <button>Edit</button> <button>Delete</button>
Final - OS	BSCS2703	Q8	2025-06-25 2:00 PM	<button>View</button> <button>Edit</button> <button>Delete</button>
Mid Term - Information Security	BSSE6311	P4	2025-06-22 10:00 AM	<button>View</button> <button>Edit</button> <button>Delete</button>
Final - OPP	BSCS2803	S8	2025-06-25 2:00 PM	<button>View</button> <button>Edit</button> <button>Delete</button>
Mid Term - ML	BSSE2011	B7	2025-06-22 10:00 AM	<button>View</button> <button>Edit</button> <button>Delete</button>
Final - PF	BSCS8203	A6	2025-06-25 2:00 PM	<button>View</button> <button>Edit</button> <button>Delete</button>

UMC Reports:

← Dashboard UMC Reports Sign Out

Exam Title	Student ID	Description	Snapshot	Time
Mid Term - Data Structures	BSSE001	Looking repeatedly to the left side during exam		10:05 AM
Final - OS	BSCS014	Attempted to exchange answer sheets during the exam		2:22 PM
Mid Term - Information Security	BSSE6311	Looking repeatedly to the left side during exam		10:45 AM
Final - PF	BSCS8203	Whispering answers to a nearby student		2:42 PM

View Requests:

← Dashboard		View Requests				Sign Out
Invigilator	Time & Exam	Room	Type	Status	Request Info	
Sir Waqas	10:05 AM – Data Structures	A-009	Material Request	Pending	See Request	
Miss Sana	2:10 PM – Operating Systems	A-006	IT Request	Resolved	See Request	
Sir Ali	10:05 AM – Data Structures	A-005	UMC Request	Pending	See Request	
Miss Laraib	2:10 PM – OPP	B-009	Material Request	Resolved	See Request	
Sir Moshin	10:05 AM – Math	B-008	UMC Request	Pending	See Request	
Miss Sana	2:10 PM – Operating Systems	A-004	IT Request	Resolved	See Request	

Download Reports:

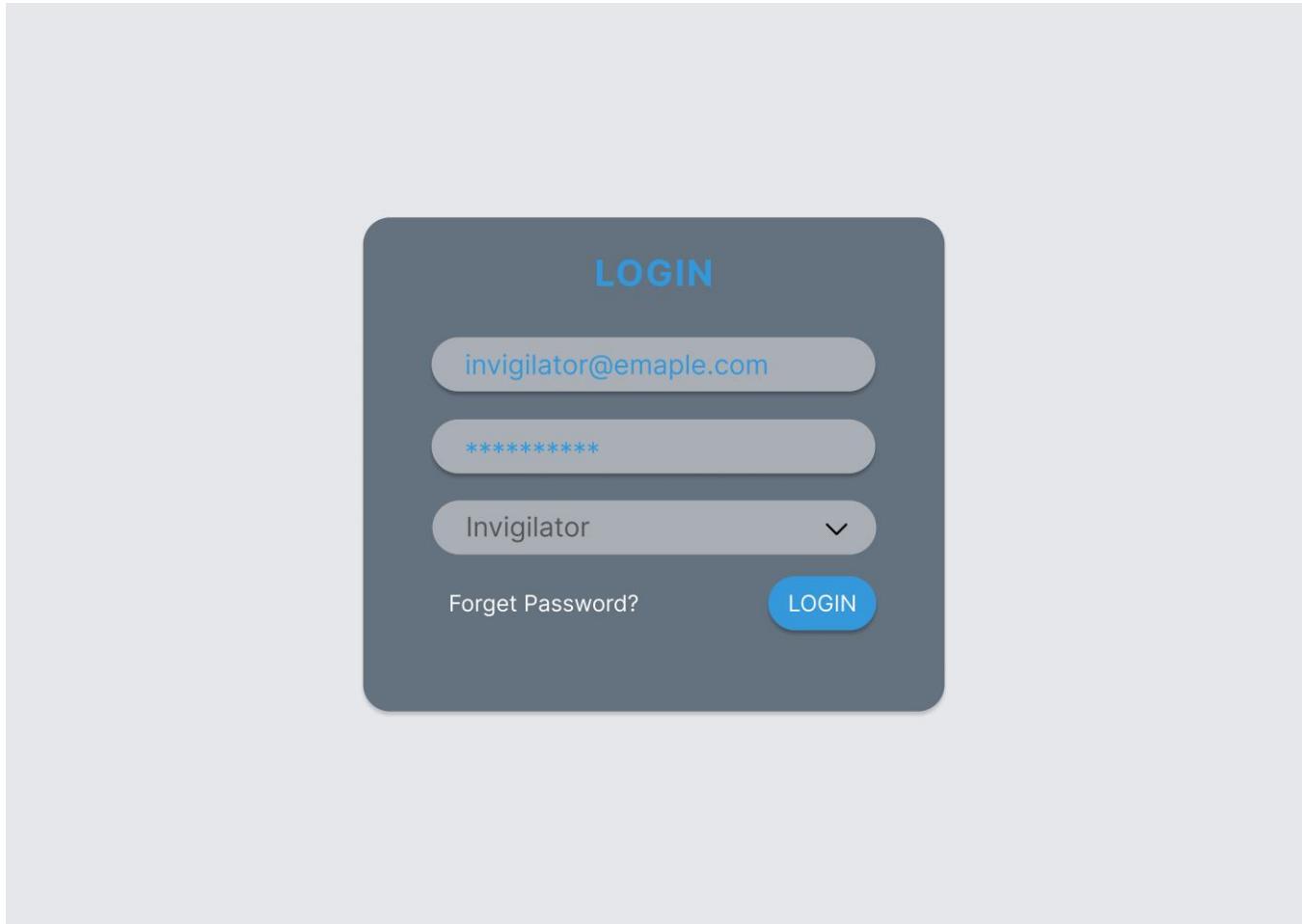
← Dashboard Download Reports Sign Out

Exam Title	Course ID	Date	Downloads
Mid Term - Data Structures	BSSE2311	2025-06-22	PDF CSV Attendance
Final - OS	BSCS2203	2025-06-25	PDF CSV Attendance
Mid Term - OPP	BSSE6311	2025-06-22	PDF CSV Attendance
Final - IS	BSCS2273	2025-06-25	PDF CSV Attendance
Mid Term - PF	BSSE2511	2025-06-22	PDF CSV Attendance
Final - CNN	BSCS2903	2025-06-25	PDF CSV Attendance
Mid Term - Math	BSSE2911	2025-06-22	PDF CSV Attendance
Final - HCI	BSCS2403	2025-06-25	PDF CSV Attendance

IT Request:

The screenshot shows a web-based IT Request form. At the top, there is a blue header bar with the text "← Dashboard" on the left, "IT Request" in the center, and "Sign Out" on the right. Below the header is a white form area with three input fields: "Issue Type" (a dropdown menu with "Select an Issue" placeholder), "Problem Description" (a text area with "Briefly explain the issue..." placeholder), and "Urgency Level" (a dropdown menu with "Select Urgency" placeholder). At the bottom of the form is a large blue button labeled "Submit IT Request".

Invigilator Side Login Page:



InvigilEye Invigilator Dashboard:

The dashboard features a dark blue header bar with the text "InvigilEye Invigilator Dashboard" on the left and a "Sign Out" button on the right. Below the header is a light gray header section with the text "Invigilator Dashboard - InvigilEye". The main content area is divided into six colored boxes arranged in a grid-like pattern:

- View Attendance** (Light Blue): Mark or update attendance manually in real-time.
- View Snapshots** (Light Green): Browse captured evidence for suspicious behavior.
- View Stream** (Yellow): Monitor live student feed with red alert highlights.
- UMC Request** (Pink): Report unfair means case with snapshot and note.
- IT Request** (Purple): Raise technical issues faced during exam monitoring.
- Material Request** (Light Purple): Request extra sheets or question papers instantly.

View & Edit Attendance:

[← Dashboard](#)
View & Edit Attendance
Sign Out

Student Name	Roll No	Status	Timestamp
Ali Raza	BSSE001	Present ▾	10:01 AM
Maria Khan	BSSE002	Absent ▾	10:03 AM
Usman Tariq	BSSE003	Present ▾	10:05 AM
Sarah Iqbal	BSSE004	Present ▾	10:06 AM
Hassan Javed	BSSE005	Present ▾	10:07 AM
Anam Asif	BSSE006	Absent ▾	10:08 AM
Zeeshan Shah	BSSE007	Present ▾	10:09 AM
Mehwish Bano	BSSE008	Present ▾	10:10 AM
Junaid Saleem	BSSE009	Present ▾	10:11 AM
Iqra Ali	BSSE010	Select ▾	10:12 AM

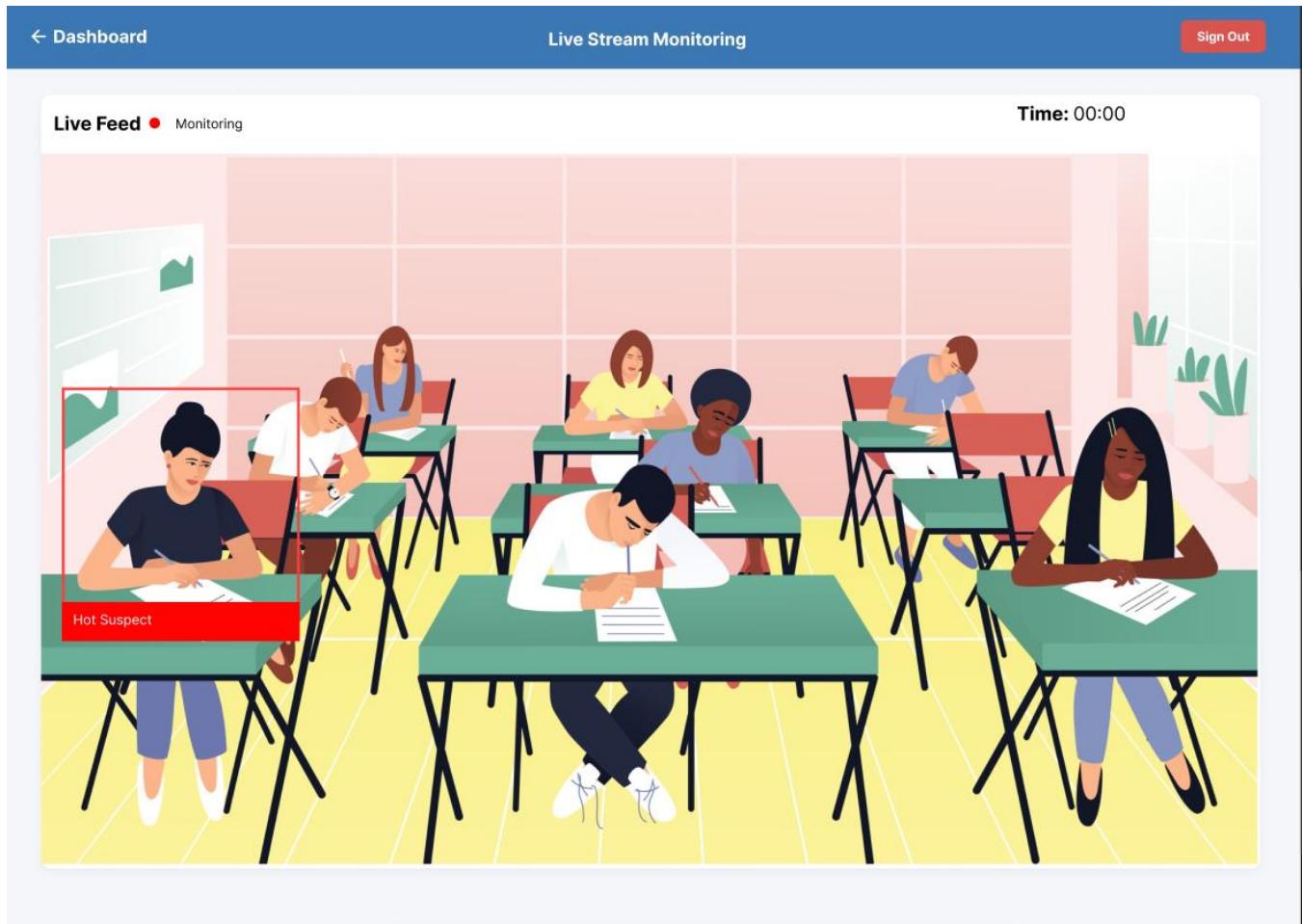
Save Attendance

Snapshots:

← Dashboard Snapshots Sign Out

The image shows a user interface for monitoring student activity. At the top, there's a blue header bar with navigation links: '← Dashboard', 'Solutions', and 'Sign Out'. Below the header are four square thumbnails, each depicting a different scene of students at desks. The top-left thumbnail shows two students, one with a patterned shirt and one with glasses, both writing. The top-right thumbnail shows two students, one in an orange shirt and one in a green shirt, both looking towards the right. The bottom-left thumbnail shows two female students, one in a pink top and one in a red top, both looking down at their papers. The bottom-right thumbnail shows two male students, one in a blue shirt and one in a green shirt, both looking towards the right. The overall theme is classroom monitoring.

Live Stream Monitoring:



UMC Request:

The screenshot shows a web-based application for submitting a UMC (Unproctored Medium Check) request. The interface is divided into several sections:

- Header:** A blue header bar with navigation links: "← Dashboard" on the left, "UMC Request" in the center, and "Sign Out" on the right.
- Form Fields:** A white rectangular form area containing the following fields:
 - Course ID:** An input field labeled "Enter Course ID".
 - Student ID:** An input field labeled "Enter Student Roll No".
 - Room No:** An input field labeled "Enter Room No".
 - Description:** A text area labeled "Describe what was observed...".
 - Attach Snapshot:** A file upload field with a "Choose File" button and the message "No file chosen".
- Buttons:** At the bottom of the form area are two blue buttons: "View Previous Request" on the left and "Submit UMC Request" on the right.

IT Request:

The screenshot shows a web-based IT Request form. At the top, there is a blue header bar with the text "← Dashboard" on the left, "IT Request" in the center, and "Sign Out" on the right. Below the header is a white form area with three main sections: "Issue Type:", "Problem Description:", and "Urgency Level:". The "Issue Type:" section contains a dropdown menu with the placeholder "Select an Issue". The "Problem Description:" section is a text input field with the placeholder "Briefly explain the issue...". The "Urgency Level:" section contains a dropdown menu with the placeholder "Select Urgency". At the bottom of the form are two blue buttons: "View Previous Request" on the left and "Submit IT Request" on the right.

← Dashboard

IT Request

Sign Out

Issue Type:

Select an Issue

Problem Description:

Briefly explain the issue...

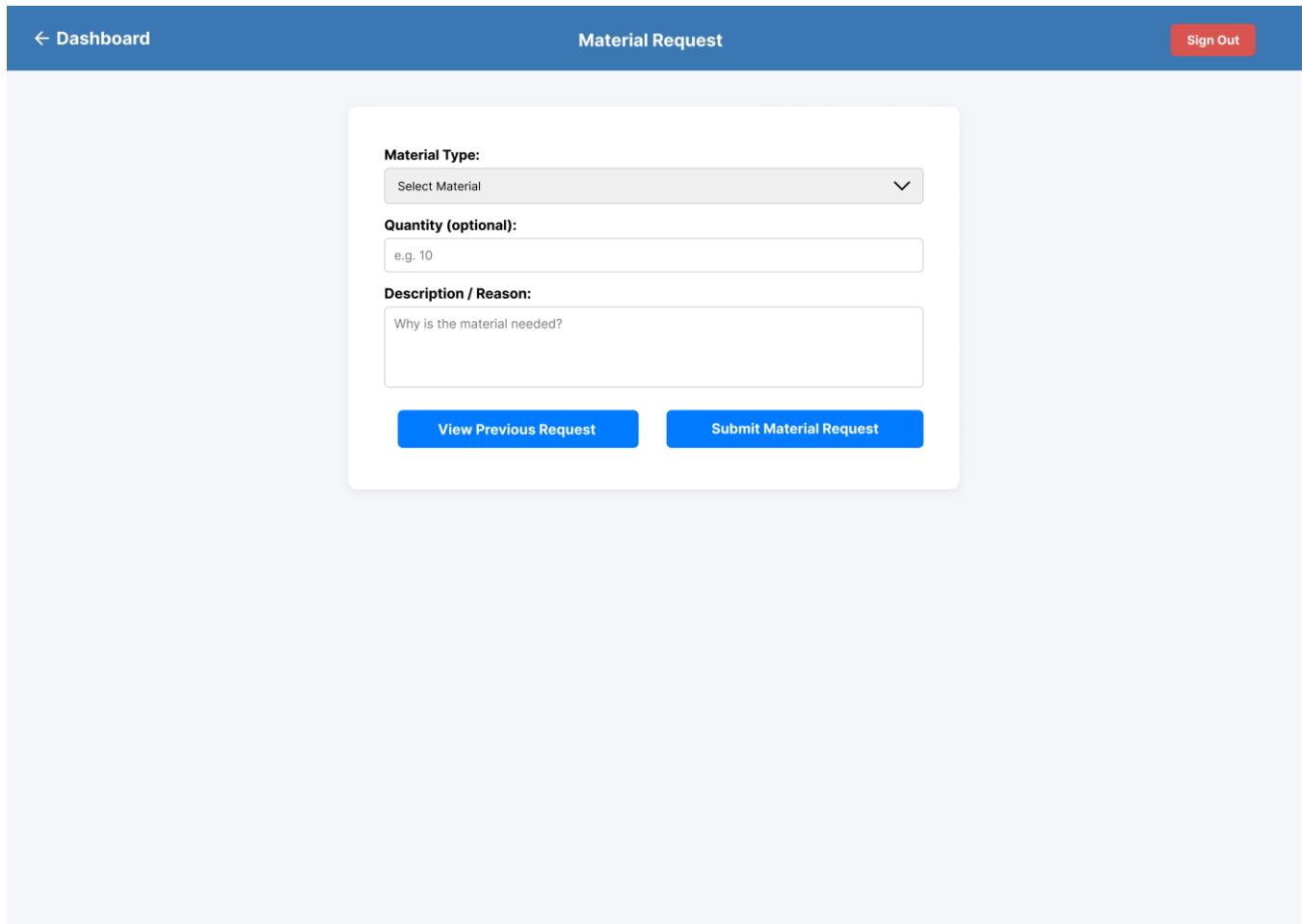
Urgency Level:

Select Urgency

View Previous Request

Submit IT Request

Material Request:



A screenshot of a Material Request form. At the top, there's a blue header bar with '← Dashboard' on the left, 'Material Request' in the center, and 'Sign Out' on the right. Below the header is a white form area with three input fields: 'Material Type' (a dropdown menu with 'Select Material'), 'Quantity (optional)' (a text input field with 'e.g. 10'), and 'Description / Reason' (a text area with placeholder text 'Why is the material needed?'). At the bottom of the form are two blue buttons: 'View Previous Request' on the left and 'Submit Material Request' on the right.

Material Type:
Select Material

Quantity (optional):
e.g. 10

Description / Reason:
Why is the material needed?

[View Previous Request](#) [Submit Material Request](#)

View Previous Requests:

[← Dashboard](#) **View Previous Requests** [Sign Out](#)

Request Type	Time	Status	Action
IT Request	10:22 AM	Pending	Re-send
UMC Request	10:45 AM	Resolved	<input checked="" type="checkbox"/>
Material Request	11:10 AM	Pending	Re-send
UMC Request	10:22 AM	Pending	Re-send
IT Request	10:45 AM	Resolved	<input checked="" type="checkbox"/>
Material Request	11:10 AM	Pending	Re-send

4. Other Design Details

4.1 Core System Focus

- **Exam Hall Monitoring Only** – Designed specifically for physical exam supervision (no online/remote capabilities).
- **Desktop-Based** – No mobile or web access.

4.2 Hardware Dependency

- **Requires Cameras** – Must be connected to the processing PC.
- **No Specialized Hardware** – Uses standard desktop setup.

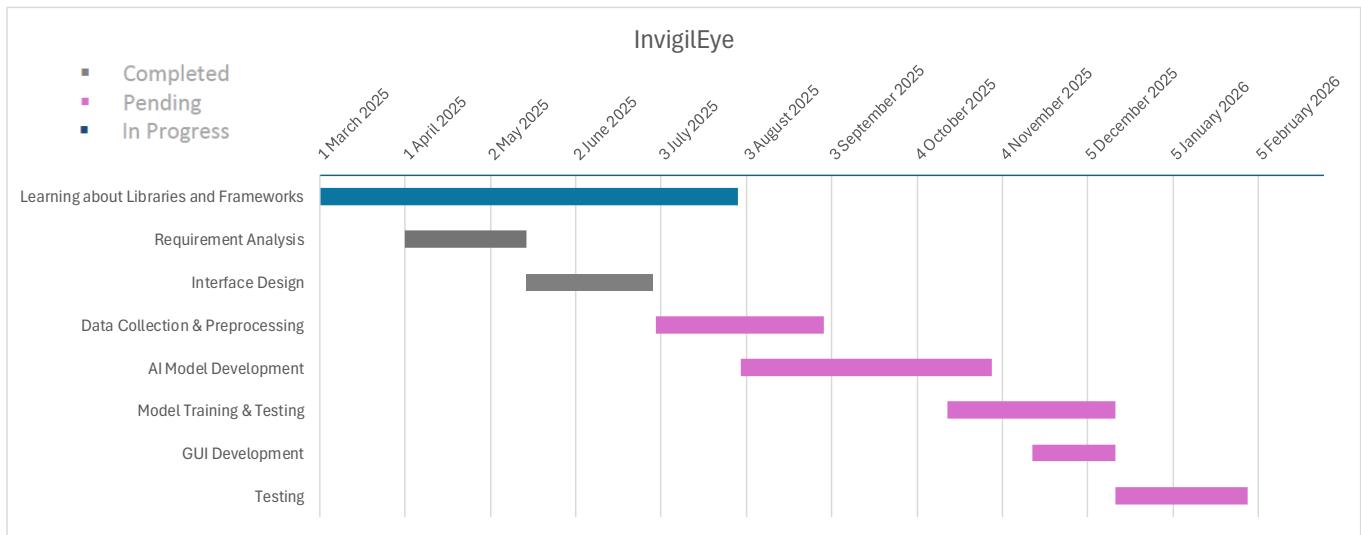
4.3 Reporting

- **Automated Reports** – Generated post-exam.

4.4 Limitations

- **No Offline Mode Mentioned** – Require active camera feed.
- **Need Active Internet Connection**

5. Revised Project Plan



6. References

- [1] Proctorio. “*Online Proctoring | Proctorio.*” [Online]. Available: <https://proctorio.com>. Accessed: June 17, 2025.
- [2] ProctorU. “*Remote Proctoring & Live Online Exam Proctoring.*” [Online]. Available: <https://proctoru.com>. Accessed: June 17, 2025.

7. Appendix A: Glossary

This glossary defines the key terms used throughout the Software **Design** Specification (**SDS**) for the InvigilEye.

Term	Definition
Admin	An admin is a primary system user with administrative privileges responsibilities.
GPU	A specialized processor designed to accelerate graphics rendering and is suitable for AI & ML tasks.
Invigilator	A person who is responsible for managing the system during exam session and has every right to take suitable action against suspicious activities.
Facial Recognition	AI technology that identifies and verifies identities through face and posture of body.
UMC	A formal action again illegal activities during exam session.
Snapshot	An image captured at a specific point of incident.
RTSP	Network protocol to stream video data from camera in real time.
Hot suspect	A label for students for highly suspicious.

Appendix B: IV & V Report

(Independent verification & validation) IV & V Resource

Name	Signature
------	-----------

S#	Defect Description	Origin Stage	Status	Fix Time	
				Hours	Minutes
1					
2					
3					
...					

Table 1: List of non-trivial defects

This document has been adapted from the following:

1. Previous project templates at UCP
2. High-level Technical Design, Centers for Medicare & Medicaid Services. (www.cms.gov)