Digital Twin

Software Requirement Specifications 2
presented to the academic faculty
by

Nehal Naeem Haji	nh07884
Manal Hasan	mh08438
Eeshal Khalidnadeem Qureshi	eq08433
Muhammad Shawaiz Khan	mk07899

In partial fulfillment of the requirements for
Bachelor of Science
Computer Science

Dhanani School of Science and Engineering

Habib University
Fall 2025

Copyright © 2019 Habib University

1 System Block Diagram

This diagram provides a high-level architectural overview of the system and its key components. It illustrates how user authentication, IoT-based energy management, and security modules interact to enable efficient facility monitoring, real-time crowd management, and emergency response.

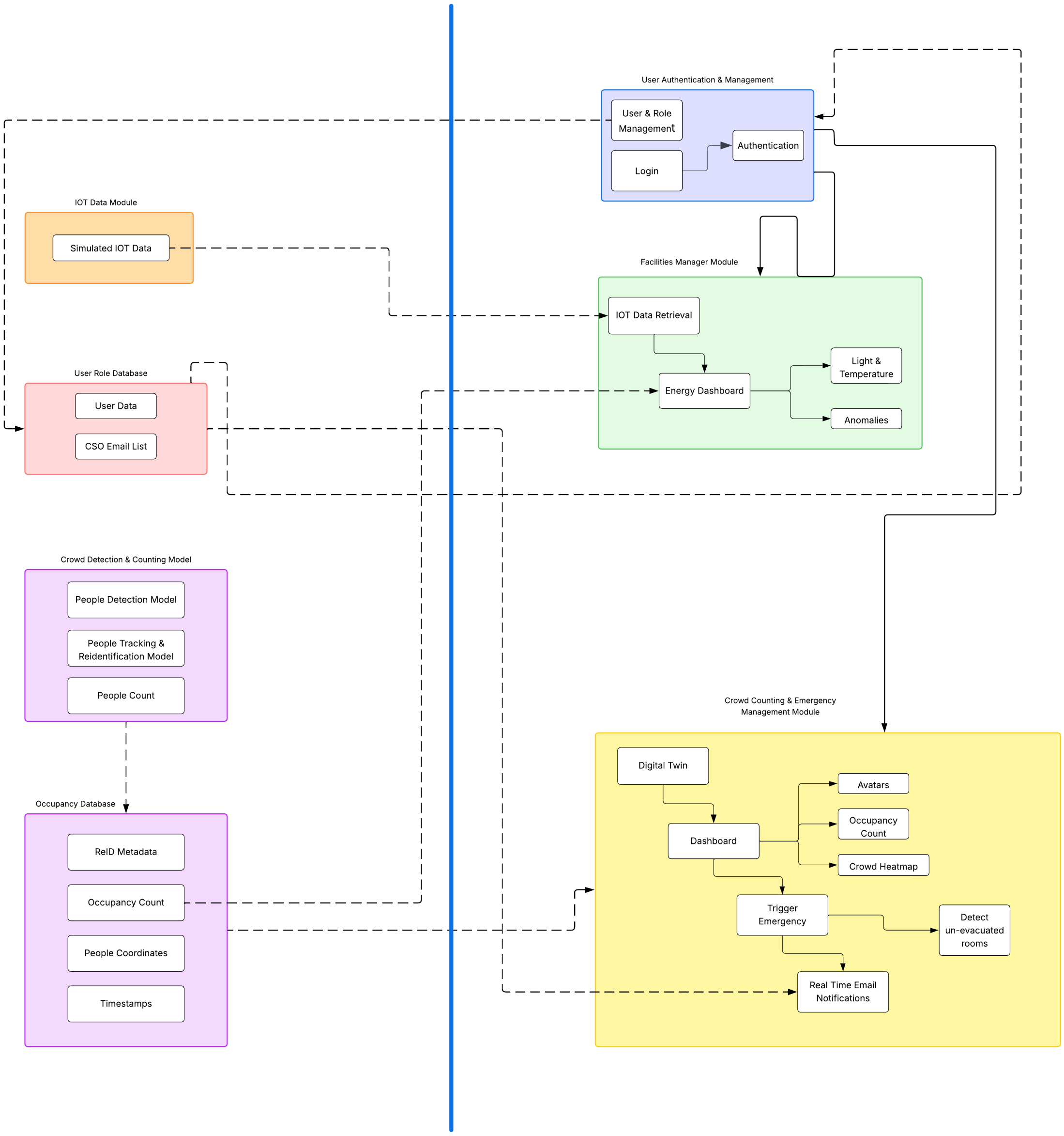


Figure 1: System Block Diagram

1.1 System Components

The system is divided into several major modules, each serving a distinct operational and user-specific purpose.

1.1.1 User Authentication and Role Management

This module enforces **Role-Based Access Control (RBAC)** to ensure that only authorized personnel can access specific functionalities.

- **User & Role Management:** Enables Admin to register user, assign system roles and manage users.
- **Authentication:** Verifies credentials during login before granting access to role-specific dashboard. It also allows the Security Admin to assign CSOs as notification recipients.

1.1.2 User Role Database

Stores user-related information and role-specific information.

- **User Data:** Contains verified user credentials and assigned role permissions.
- **CSO Email List:** Maintains a list of security personnel who receive real-time alert notifications.

1.1.3 IoT Data Module

Retrieves simulated IoT sensor data of laboratories.

- **Simulated IoT Data:** Includes temperature and lighting status. This data is continuously fetched and shared with the Energy Management module for monitoring and anomaly detection.

1.1.4 Facilities Manager Module (Energy Management Module)

This module focuses on energy efficiency and environment monitoring using simulated IoT sensor data.

- **IoT Data Retrieval:** Continuously collects simulated sensor data from the IoT Data Module.

- **Energy Dashboard:** Displays occupancy data, temperature, and lighting status for laboratories. Correlates occupancy data with energy usage to flag rooms unoccupied for more than 25 minutes or when temperature thresholds are violated.

1.1.5 Crowd Detection and Counting Model

Implements computer vision techniques for real-time crowd monitoring and people tracking.

- **People Detection Model:** Detects individuals from CCTV camera feeds using YOLOv8.
- **People Tracking & Re-identification:** Tracks individuals and maintains consistent IDs across frames using DeepSort/ByteTrack.
- **People Count:** Calculates the occupancy count per laboratory and stores it in the Occupancy Database.

1.1.6 Occupancy Database

Stores metadata related to crowd movement and occupancy.

- **ReID Metadata:** Maintains unique identifiers (embeddings) for tracked individuals and stores person-specific tracking features for re-identification.
- **Occupancy Count:** Holds people count data.
- **People Coordinates / Timestamps:** Logs spatial and temporal movement information for each detected person.

1.1.7 Crowd Counting and Emergency Management Module

This module integrates crowd data with emergency protocols to manage crowd visualization, evacuation tracking, and emergency response.

- **Digital Twin:** Represents a 3D virtual model of laboratories showing avatar movements.
- **Dashboard:** Displays occupancy counts, and crowd heatmaps.

- **Trigger Emergency:** Enables the Security Admin to declare or end emergencies and automatically identifies un-evacuated rooms when an emergency is declared.
- **Real-Time Email Notifications:**.Sends alerts to designated CSOs with details of occupied rooms and locations.

1.2 System Flow

The system's operational flow ensures that user access, monitoring, and emergency management function seamlessly.

1. **User Login → Authentication:** Verifies user credentials for secure access.
2. **Authentication → Role-Based Access:** Directs users to dashboards based on assigned roles.
3. **Facilities Manager Module:** Displays IoT-based energy and occupancy insights.
4. **Crowd Detection & Counting Model:** Processes live video feeds for real-time occupancy tracking.
5. **Digital Twin Visualization:** Updates avatars and heatmaps to reflect current crowd activity.
6. **Emergency Management Module → CSO Alerts:** Detects un-evacuated areas and sends automated emergency notifications.

1.3 Data Flow

The data flow within the system ensures accurate and synchronized exchange between all modules.

- **User Role Database → User Authentication & Management:** Provides user credentials and role information for authentication and directs users to their respective role-specific dashboards.
- **Crowd Detection Model → Occupancy Database:** Supplies real-time crowd and movement metadata for storage and further analysis.

- **Occupancy Database → Crowd Counting & Emergency Management Module:** Real-time occupancy data from the database updates the Digital Twin with live avatars and heatmaps, while during emergencies, un-evacuated areas trigger automated email alerts to CSOs.

- **Occupancy Database → Facilities Management Module:** Sends occupancy data to assist in decision-making for energy and safety management. If a room remains unoccupied beyond a threshold time, the module flags those rooms.

- **IoT Data Module → Facilities Management Module:** Provides continuous environmental sensor data (e.g., temperature and lighting status) for real-time facility monitoring.

- **User Role Database → Crowd Counting & Emergency Management Module:** Supplies contact details of designated CSOs (Chief Safety Officers) to enable automated email notifications and emergency alerts.

2 Project Plan

The Project Plan provides an outline of the project, detailing the objectives, tasks, timeline, required resources, and potential risks. This section serves as a roadmap to guide the project from inception through completion, ensuring each phase aligns with the overall goals of providing accurate and hyper-local weather forecasts. By clearly defining milestones, team roles, and project phases, this plan aims to identify any risks that may affect project outcomes and suggest appropriate mitigation strategies.

2.1 Project Objectives

The Digital Twin project aims to:

- Develop an anonymity-preserving digital twin for two university labs - the Projects Lab and the Digital Instrumentation Lab.

- Implement real-time anonymous crowd counting and occupancy monitoring via object detection and object tracking models.

- Provide emergency alerts for evacuation management, identifying uncleared spaces in case of emergencies.

- Incorporate simulated IoT data, particularly lighting and temperature, to optimize energy usage based on real-time occupancy.

2.2 Tasks

Key tasks involved in the project are as follows:

- **Video Recording & Consent**
  - **Video Data**: Collect 1 hour long synchronized video footage recorded by already existing CCTV cameras inside aforementioned labs to create video model
  - **Consent**: Get consent from students in footage, Program Director, Security Team and Project Supervisor via IRB forms.
- **Model and Algorithm Development**
  - **Model Experimentation**: Test various multi-camera object detection, tracking and reidentification models (YOLO v8, Bytetrack, DeepSORT, CSRNet).
  - **Duplicacy Mitigation**: Ensure people count does not include duplicates as a result of overlapping footage.
  - **Model Population**: Accurately populate model with person positions from multiple streams.
- **Frontend Development**
  - **Role-Based Interface Design**: Design separate UI/UX wireframes for Admin and Facilities Manager.
  - **Role-Specific Layouts**: Develop user-specific dashboards with role-based functionality.
  - **Data Visualization Integration**: Add heatmap and people count for visualizing crowds. Add temperature and light readings to visualize energy usage.
- **Backend Development**
  - **SMTP Server**: Set up SMTP server so that CSOs can get real-time notifications during emergencies.

  - **Database Design**: Design schemas to store user credentials (encrypted), ReID feature embeddings, occupancy count, people coordinates and timestamps.
  - **API Development**: Develop APIs to return output of object detection and tracking models.
  - **Model Integration**: Integrate object detection and tracking models into the backend for APIs to call.
  - **User Authentication**: Implement secure user authentication.

- **Modeling**
  - **2D Twin Development**: Initially develop a 2D digital twin to validate the extraction and placement of people based on coordinates. Ensure that these detected person positions from multi-camera input are correctly mapped and synchronized in the 2D environment.
  - **Coordinate Validation**: Verify that coordinate data from the backend accurately represents real-world spatial positions within the 2D layout.
  - **3D Twin Transition**: Once 2D validation is complete, extend the twin into a 3D environment where each detected individual will be represented by a 3D avatar instead of a dot.
  - **Environment Visualization**: Model and render the lab environment in 3D to provide spatially realistic visualization of people movement and occupancy.

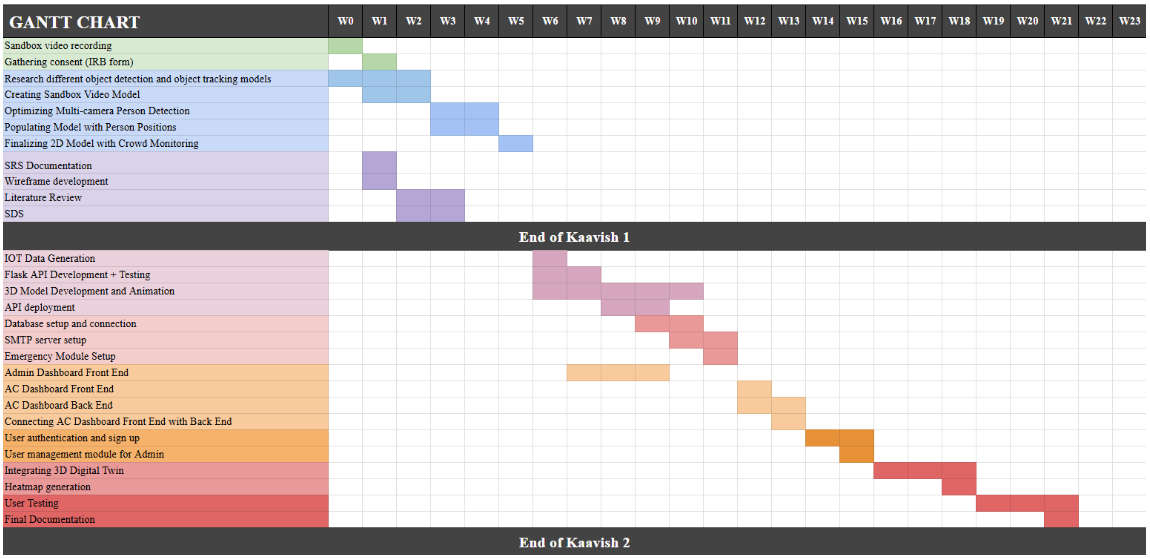
- **Testing and Validation**
  - **Unit Testing**: Test individual components (e.g., person identification models, re-identification, API responses).
  - **Integration Testing**: Verify interactions between components, such as frontend-backend communication and people coordinates with avatar position.
  - **System Testing**: Conduct end-to-end testing of the application across roles.
  - **Model Validation**: The performance of multi-camera person detection and re-identification will be validated by comparing model outputs with manually counted results.

- **Usability Testing**: Gather user feedback on the app interface, role-based functionalities, and features.
- **Final Validation**: Perform a comprehensive test of the system for reliability. Resolve any bugs encountered.

- **Documentation**
  - **Technical Documentation**: Entails system architecture, backend functions, use case diagrams and other such documents.
  - **User Guide**: A document which will explain app features, role-specific functionalities and navigation to intended end-users.
  - **Kaavish Documentation**: Detailed documentation satisfying Final Year Project requirements.

2.3 Timeline

The Gantt Chart below outlines the timeline laid out for the project.



GANTT CHART	W0	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16	W17	W18	W19	W20	W21	W22	W23
Sandbox video recording	■																							
Gathering consent (IRB form)		■																						
Research different object detection and object tracking models	■	■	■																					
Creating Sandbox Video Model		■	■																					
Optimizing Multi-camera Person Detection				■	■																			
Populating Model with Person Positions				■	■																			
Finalizing 2D Model with Crowd Monitoring						■																		
SRS Documentation		■																						
Wireframe development		■																						
Literature Review			■	■																				
SDS			■	■																				
End of Kaavish 1																								
IOT Data Generation							■																	
Flask API Development + Testing							■	■																
3D Model Development and Animation							■	■	■	■	■													
API deployment									■	■														
Database setup and connection										■	■													
SMTP server setup											■	■												
Emergency Module Setup												■												
Admin Dashboard Front End								■	■	■														
AC Dashboard Front End													■											
AC Dashboard Back End													■	■										
Connecting AC Dashboard Front End with Back End														■										
User authentication and sign up															■	■								
User management module for Admin																■								
Integrating 3D Digital Twin																	■	■	■					
Heatmap generation																			■					
User Testing																				■	■	■		
Final Documentation																						■		
End of Kaavish 2																								

Figure 2: Gantt Chart Timeline Visualization

2.4 Resources

- **Technology Stack**: For the visualization of the Digital Twin, as well as displaying live occupancy and crowd density heatmaps, we will make use of Three JS. Real-time, accurate crowd counting will be achieved using the YOLO v8 model, given its capability of detecting multiple objects in a single frame. And in order to track people across multiple frames, the latest version of the DeepSort algorithm will be employed.

- **Backend**: The APIs, namely the /get_people_count and /get_people_positions, will be implemented using Flask, as integrating our Object Detection and Tracking models will be simpler.

- **Frontend**: ReactJS will be used to create the various dashboards and visualization tools.

- **Database**: To store occupancy data and real-time event records, MongoDB shall be used by our system.

- **Hardware**: To perform real-time, local inference for testing purposes, we will require access to the NVidia RTX Titan GPU that is available on campus via remote connection.

2.5 Team Roles

The team formation leverages the members' individual expertise in deep learning, data handling, software engineering, and user experience design to ensure a well-rounded and efficient development process for the Digital Twin system. Each member's role is designed to align with their technical strengths and project experience.

- **Nehal Naeem Haji (Project Lead)** Nehal leads the development of the system's deep learning components, including CCTV data processing, real-time object detection, and tracking using YOLOv8 and ByteTrack. With expertise in deep learning and ISC2-certified data handling, he ensures privacy compliance and system reliability. He also contributes to backend integration and model optimization for stable, high-performance inference.

- **Manal Hasan (Computer Vision Frontend Development Lead)** Manal specializes in implementing and refining CCTV-based identification and people tracking using advanced computer vision and digital image processing techniques. Leveraging her strong foundation in deep learning and prior backend development experience, she ensures accurate detection and crowd analysis. Additionally, with her UI/UX certification and design background, she contributes to developing intuitive, role-specific interfaces that enhance user experience across the system.

- **Eeshal Khalidnadeem Qureshi (Spatial Modeling & Data Integration Lead)** Eeshal develops the 3D spatial models of the labs using Unity 3D, ensuring realistic visualization of occupancy and movement data. Combining her background in deep learning and data science with her full-stack development experience, she bridges the integration between computer vision outputs, sensor data, and the digital twin environment for synchronized and accurate real-time visualization.

- **Muhammad Shawaiz Khan (UI/UX & Backend Infrastructure Lead)** Shawaiz leads the design of an accessible, user-centric interface for different system roles, emphasizing clarity and usability. Utilizing his expertise in UI/UX design and Flask-based backend development, he is responsible for developing a robust data pipeline that connects the frontend dashboards with backend APIs, ensuring seamless and reliable real-time performance across the system.

2.6 Risks & Mitigation Strategies

- **GPU Contention:** Shared GPU resources may cause delays in model inference during real-time crowd detection, especially when multiple video streams are processed simultaneously.

  Mitigation:

  - Reserve dedicated GPU slots during peak times.
  - Fall back to CPU or reduced-resolution input if GPU capacity is exceeded.

- **API Latency & Rate Limits:** Latency in data retrieval from external services (e.g., Google Sheets API) or rate limits can slow down API responses.

  Mitigation:

  - Implement exponential backoff to handle API rate limits as recommended in Google's documentation.
  - Optimize API calls to reduce unnecessary requests and speed up data retrieval.

- **Three.js Performance:** The browser engine might struggle to handle intensive rendering of 3D models, especially with live data updates.

  Mitigation:

  - Start with static meshes and low-resolution textures to optimize the initial setup.
  - Optimize rendering performance by reducing unnecessary details and using efficient rendering techniques in Three.js.

- **Data Synchronization:** Synchronizing video streams, occupancy data, and 3D models in real-time could lead to misalignments.

  Mitigation:

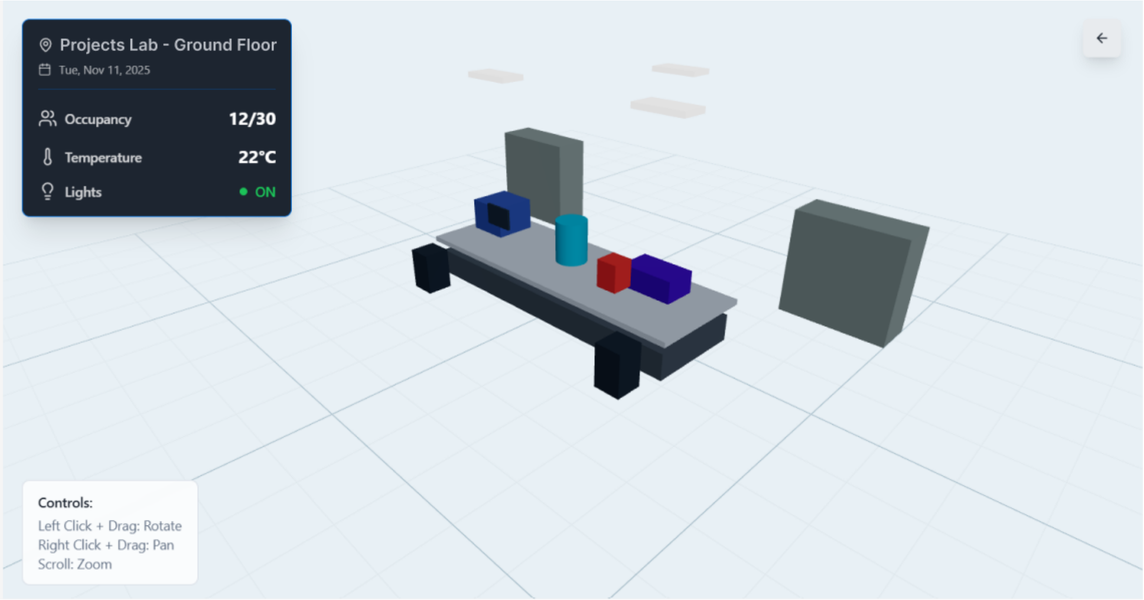
  - Use timestamped JSON data to ensure synchronized updates across video streams, 3D models, and real-time analytics.
  - Poll data at fixed intervals to maintain consistency.

- **IT Delays:** Delays in provisioning cloud resources or IT support for the project may disrupt progress.

  Mitigation:

  - Use alternative cloud services (such as Vercel or OneDrive) if GCP provisioning faces delays.
  - Plan for potential fallback systems and ensure that the team is ready for alternative cloud deployment options.

3 Wireframes



Projects Lab - Ground Floor

Tue, Nov 11, 2025

Occupancy 12/30

Temperature 22°C

Lights ● ON

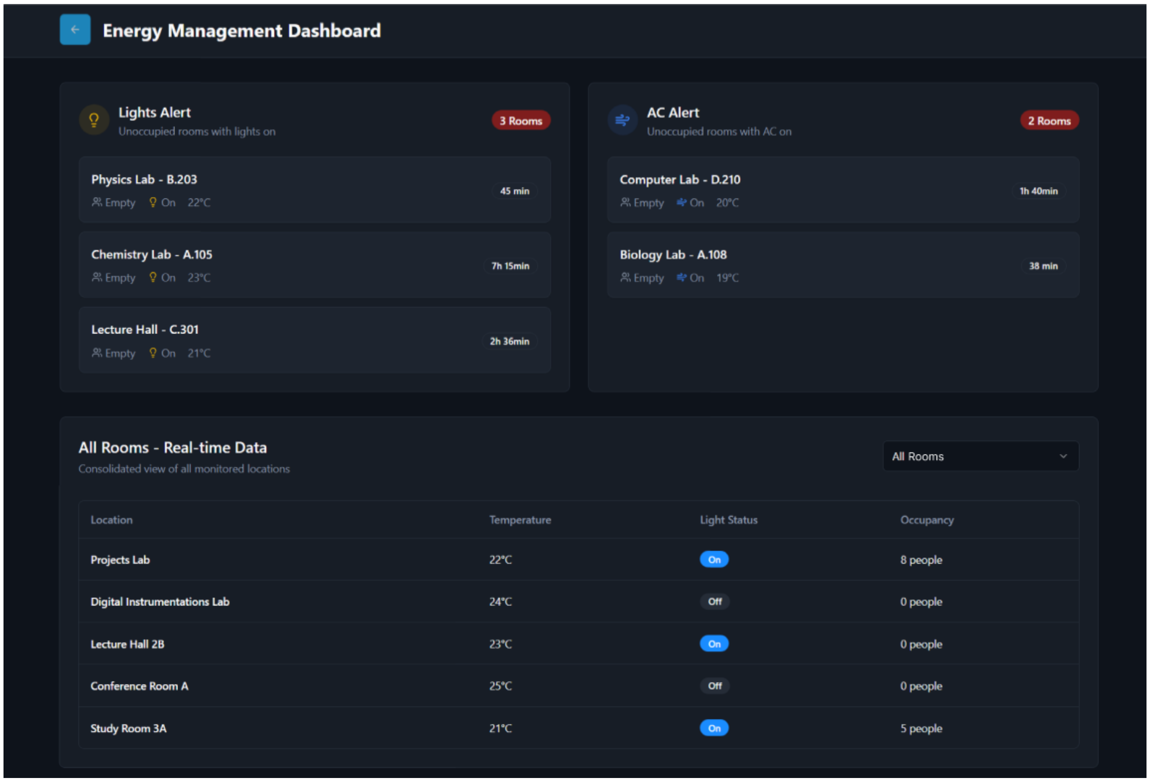
Controls:

Left Click + Drag: Rotate

Right Click + Drag: Pan

Scroll: Zoom

Figure 3: Main Digital Twin view page with data displayed on the top left widget of the current lab being viewed



Energy Management Dashboard

Lights Alert
Unoccupied rooms with lights on
3 Rooms

Physics Lab - B.203
Empty On 22°C
45 min

Chemistry Lab - A.105
Empty On 23°C
7h 15min

Lecture Hall - C.301
Empty On 21°C
2h 36min

AC Alert
Unoccupied rooms with AC on
2 Rooms

Computer Lab - D.210
Empty On 20°C
1h 40min

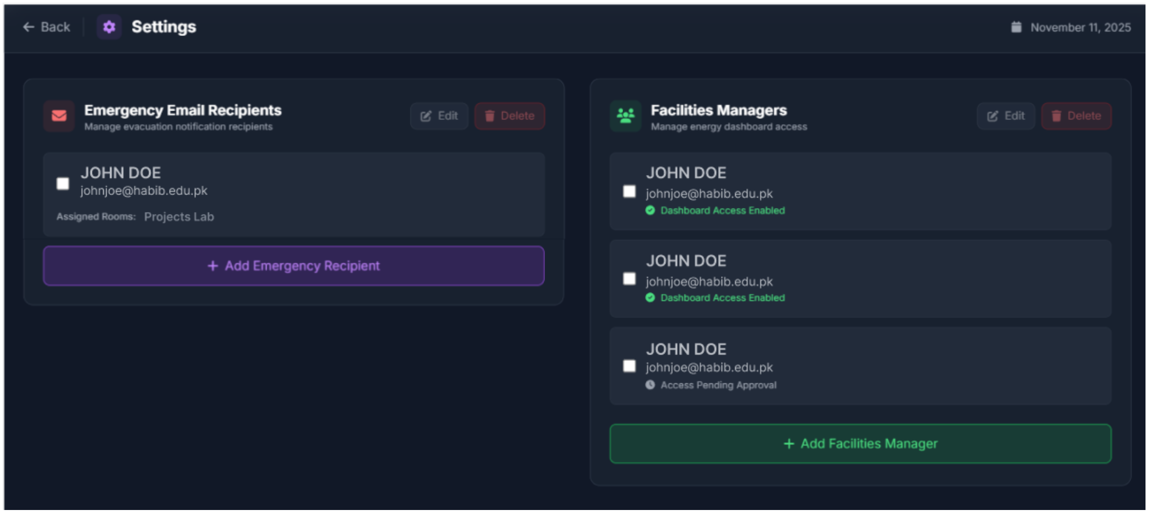
Biology Lab - A.108
Empty On 19°C
38 min

All Rooms - Real-time Data
Consolidated view of all monitored locations

All Rooms

Location	Temperature	Light Status	Occupancy
Projects Lab	22°C	On	8 people
Digital Instrumentations Lab	24°C	Off	0 people
Lecture Hall 2B	23°C	On	0 people
Conference Room A	25°C	Off	0 people
Study Room 3A	21°C	On	5 people

Figure 4: Energy Management Dashboard - includes tables that shows flagged room due to lights status and temperature, with a consolidated table below with all rooms/labs data including occupancy count



Emergency Email Recipients
Manage evacuation notification recipients

□ JOHN DOE
johnjoe@habib.edu.pk
Assigned Rooms: Projects Lab

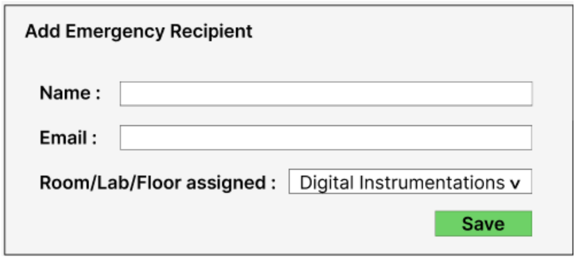
Facilities Managers
Manage energy dashboard access

□ JOHN DOE
johnjoe@habib.edu.pk
✓ Dashboard Access Enabled

□ JOHN DOE
johnjoe@habib.edu.pk
✓ Dashboard Access Enabled

□ JOHN DOE
johnjoe@habib.edu.pk
Access Pending Approval

Figure 5: Settings page for admins for assigning access to facilities manager and add CSO’s for emergency evacuation email recipient



Add Emergency Recipient

Name :

Email :

Room/Lab/Floor assigned : Digital Instrumentations v

Figure 6: Widget displayed to add Emergency email recipient