TED UNIVERSITY

CMPE492

Computer Engineering

Library Occupancy Detector

Final Report

21.12.2024

Team Members

Burak Koç

Buse Öner

Furkan Safa Altunyuva

Table of Contents

1. Introduction 3
   1.1 Definitions, acronyms, and abbreviations 5
2. Requirements Detail 6
   2.1 Functional Requirements 6
      2.1.1 Real-time Seat/Desk Occupancy Detection 6
      2.1.2 Occupancy Rate Calculation 6
      2.1.3 Detection of Unoccupied Areas 6
      2.1.4 Item Detection 6
      2.1.5 Past Occupancy Data and Reporting 7
      2.1.6 Hold Exceed Alerts 7
      2.1.7 Dynamic Hold Duration Management 7
      2.1.8 Multi-Feed Support 7
      2.1.9 Integration with Existing Applications and Administrative Web Interface 7
   2.2 Non-Functional Requirements 8
      2.2.1 Accuracy and Adaptability 8
      2.2.2 Performance 8
      2.2.3 Privacy 8
      2.2.4 Security 8
      2.2.5 Scalability 9
      2.2.6 Portability 9
3. Final Architecture and Design Details 10
4. Development/Implementation Details 14
5. Test Cases and Results 16
6. Other Project Elements 20
   6.1 Contemporary Issues and Additional Considerations 20
      6.1.1 Global Considerations 20
      6.1.2 Cultural Considerations 20
      6.1.3 Social Considerations 21
      6.1.4 Environmental Considerations 21
      6.1.5 Public Safety Considerations 22
   6.2 Ethics and Professional Responsibilities 23
      6.2.1 User Privacy and Data Protection 23
      6.2.2 Equity and Social Justice 23
      6.2.3 Responsible Use of Technology 24
      6.2.4 Commitment to the Professional Standards 24
   6.3 New Knowledge Acquired and Applied 25
      6.3.1 Description of Tools Used 25
      6.3.2 Integration of Technologies in the System 25
7. Conclusion and Future Work 27
   7.1 Conclusion 27
   7.2 Future Work 27
8. References 29

1. Introduction

Libraries are vital spaces for students seeking a conducive learning environment. However, students often waste their limited and valuable time when they come to campus without knowing the library's occupancy rate, only to find it at full capacity. To address this issue, we developed the Library Occupancy Detector, a solution that allows students to check the library's occupancy rate remotely from their current location before heading to the library. This system helps students access real-time occupancy data and prevents time-consuming trips to an already overcrowded library.

At the core of our project lies advanced vision detection technology, which utilizes Python, YOLOv10, and OpenCV to classify specific Regions of Interest (ROIs). Our system distinguishes between three critical states for each seat:

1. Occupied: Presence of a person
2. Hold: Presence of objects that indicate space reservation where there is no person
3. Empty: Absence of both people and objects

The system has been developed with two principal objectives in mind. Firstly, it is designed to provide students with real-time access to library occupancy data through existing platforms such as TEDU App and myTeduPortal. Secondly, it has been developed to offer a specialized web interface for library management. Students can view both the total number of people occupying study areas and the combined data for "hold" and "occupied" seats, allowing them to make informed decisions about when to visit the library.

The system offers a user-friendly interface designed to assist library management in monitoring space usage and occupancy patterns. The web interface has been developed and customized to meet the specific needs and requests of the library management team, who are the primary stakeholders in this project. In addition to managing occupancy data, the system enables administrators to view detailed occupancy rates and generate reports. These reports can be used by library staff for future planning or to adjust hold durations based on current occupancy rates.

The system also allows library staff to get information about seats that exceed waiting times. Based on the overall occupancy rate, these seats can be made available for students, ensuring that the library space is optimized and accessible. Depending on their roles and permissions, staff members can dynamically adjust hold durations. The combination of real-time data and dynamic hold management has the dual effect of improving student experience and streamlining operations for library staff, thereby rendering space management more efficient and effective.

The system's robustness has been a primary area of focus, ensuring accurate detection under varying conditions, such as changes in lighting, diverse camera angles, and object variations. Additionally, the utilization of Firebase for data storage and visualization enables library administrators and students to readily access real-time occupancy information, thereby enhancing the library experience for all stakeholders.

The Library Occupancy Detection System was designed with extensibility and long-term usability in mind. The capacity to accommodate multiple video feeds and adapt to future requirements is a key feature of the system. This report provides an overview of the Library Occupancy Detector project, covering its design, implementation, testing and evaluation. It also highlights the challenges encountered, the lessons learned and the potential areas for future improvement.

1.1 Definitions, acronyms, and abbreviations

- KVKK: Personal Data Protection Law, the Turkish legislation governing the protection of personal data, in accordance with “Kişisel Verilerin Korunması Kanunu”.
- myTEDUPortal: The website where students, faculty, and staff can access academic and administrative services, such as course registration, grades, and schedules.
- TeduAPP: The "Tedu App" is an application designed and developed by Tedu students, aiming to guide students about school activities, class schedules, and communities within the school under the slogan of "by students, for students.
- Occupancy Rate: The ratio of occupied seats to the total seating capacity in the library, basically representing the percentage of students currently utilizing seats.
- Hold Status: A situation where a seat or table is considered full due to the presence of personal items left by a student.
- YOLOv10: You Only Look Once, is an object identification method known for its high accuracy and real-time detection capabilities.
- Real-Time Detection: The ability of the system to process/analyze and show data almost instantly.
- QA: Quality assurance is defined as a component of quality management that aims to provide assurance that quality requirements will be fulfilled.
- ROI: The region of interest denotes a specific, predefined area within the camera's field of view that is monitored for detection purposes. In our system, each ROI corresponds to an individual seating area that is analyzed for occupancy status.

2. Requirements Detail

2.1 Functional Requirements

2.1.1 Real-time Seat/Desk Occupancy Detection

In real-time, the system must determine whether the tables and chairs in the library are occupied or not, using computer vision technology.

2.1.2 Occupancy Rate Calculation

The system shall be responsible for calculating and displaying two distinct occupancy rates.

- Firstly, the total occupancy rate, which considers all areas that are occupied, including those that are currently on hold.
- Secondly, the occupancy rate, that is based solely on the number of students who are currently present in the library, excluding areas marked as 'hold'.

2.1.3 Detection of Unoccupied Areas

The system shall accurately identify empty chairs and desks in addition to occupied areas, providing real-time updates on the number of available spaces for students.

2.1.4 Item Detection

The system has been designed to detect items left on tables or chairs with the objective of accurately classifying seating areas. In order to achieve this, the system must be capable of distinguishing between three states: areas with a person present ("occupied"), areas without a person but with belongings present ("hold"), and completely empty areas ("empty"). The system is designed to ensure the precise detection and classification of seating areas based on the presence of not only persons, but also personal items.

2.1.5 Past Occupancy Data and Reporting

The system shall retrieve occupancy data from the database to provide information to library administrators. This historical occupancy data will be analyzed and formatted into comprehensive reports, enabling library managers to make data-based decisions based on occupancy rates

2.1.6 Hold Exceed Alerts

The system shall automatically share the locations of areas held by items when the hold duration exceeds the specified limit, allowing library administrators to utilize these areas.

2.1.7 Dynamic Hold Duration Management

Library administrators shall have the ability to dynamically adjust hold durations based on current occupancy trends and reports provided by the system.

2.1.8 Multi-Feed Support

The system shall support multiple video feeds, enabling the monitoring of occupancy across different library areas simultaneously. This ensures scalability for larger implementations.

2.1.9 Integration with Existing Applications and Administrative Web Interface

The system shall integrate with existing platforms such as TEDU App and myTEDUPortal to share real-time library occupancy data from Firebase for student access. Furthermore, a web interface is to be provided for library administrators, with the aim of enabling them to access detailed occupancy data through the same Firebase integration.

2.2 Non-Functional Requirements

2.2.1 Accuracy and Adaptability

It is imperative that the system demonstrates a high level of accuracy in detecting seat, desk, and table occupancy within the library environment. In order to provide students with reliable information regarding available seating areas and accurately distinguish between occupied and unoccupied spaces, the system must be capable of adapting to changing lighting conditions and effectively identifying items placed on tables and chairs. This adaptability is necessary to ensure reliable and accurate occupancy detection in various environmental conditions.

2.2.2 Performance

In order to function effectively, the system must display rapid responsiveness in terms of processing and displaying occupancy data, with minimal latency. Additionally, it is crucial that the system does not necessitate an excessively extensive computing capacity to ensure continuous operation of the detection system.

2.2.3 Privacy

The system has been designed to prioritize user privacy by implementing a range of measures. These include the anonymization of data, the refraining from the recording of images, and the avoidance of the storage of personally identifiable information. These measures have been implemented in order to comply with privacy regulations, such as the KVKK. Additionally, individuals will be treated as objects in order to ensure compliance and minimize legal issues.

2.2.4 Security

It is imperative that the system implements robust security measures to prevent data breaches and unauthorized access.

2.2.5 Scalability

It is imperative that the system is scalable to facilitate the implementation of future updates and expansions without necessitating a substantial amount of rework.

2.2.6 Portability

In order to fulfil its function in a range of contexts, the system must be capable of deployment across different hardware configurations and operating environments without the necessity for extensive modifications. The system should support:

- Various camera systems and network setups.
- Integration with both web-based and mobile platforms to ensure optimal user accessibility.
- Facilitating seamless migration to institutional infrastructure when scaling or transitioning.

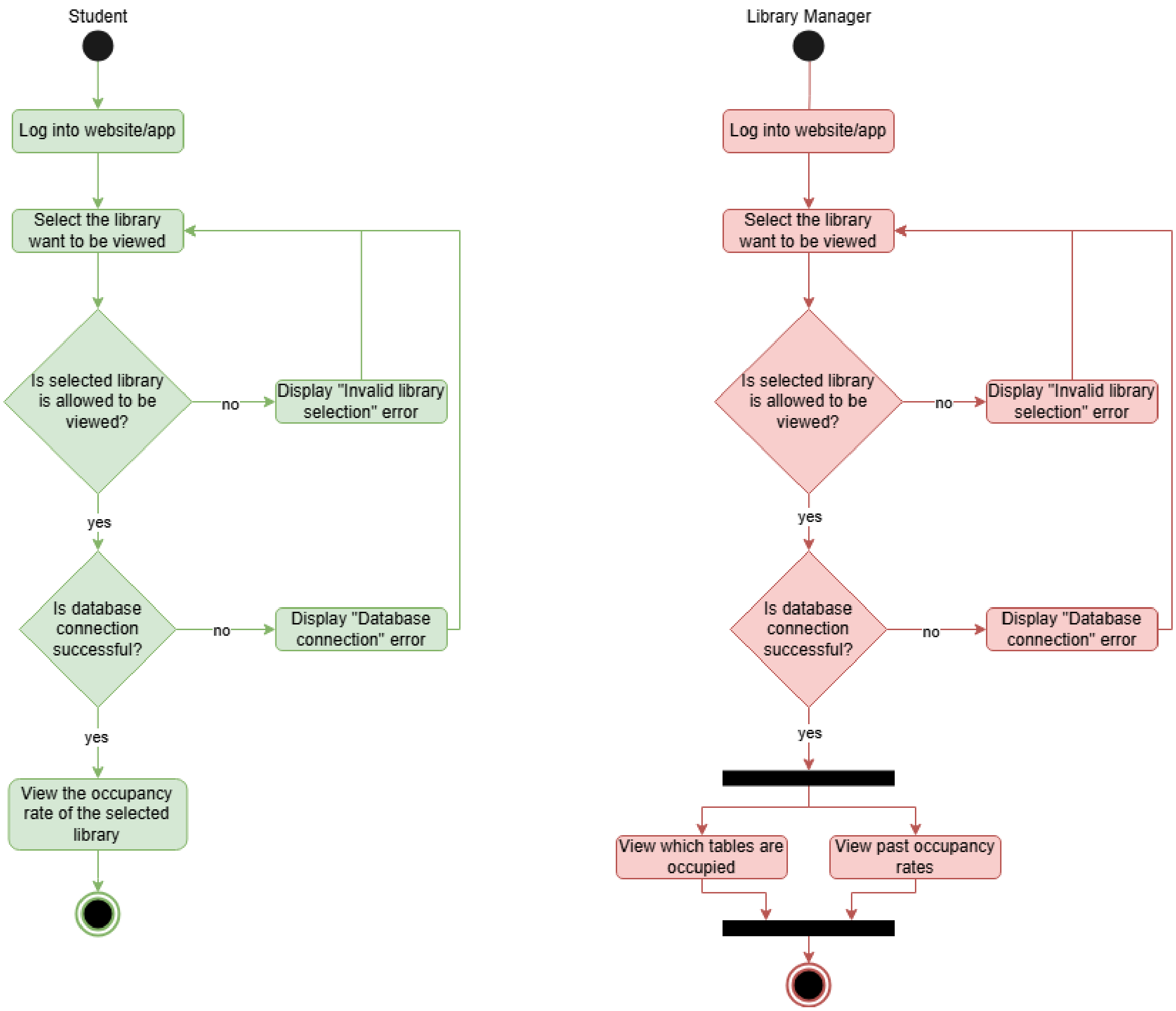
3. Final Architecture and Design Details

The overall system operates through a series of interconnected components that work in harmony to provide real-time occupancy monitoring and management capabilities. The architecture follows a distributed model with four main components: data acquisition, image processing, data management, and user interface. These components are integrated through a pipeline that processes visual information and converts it into actionable occupancy data.

The data acquisition layer consists of ceiling-mounted cameras strategically positioned above the library spaces. These cameras continuously capture frames of the monitored areas, including desks and chairs. The camera placement ensures optimal coverage while maintaining user privacy through top-down viewing angles. The captured data feeds directly into the image processing component, which implements the YOLO (You Only Look Once) object detection algorithm using the COCO dataset. This layer processes the captured frames through continuous extraction from the video feed, followed by object detection using the YOLO algorithm to identify and classify objects. The system then performs zone analysis by evaluating predefined spatial zones corresponding to desk positions, ultimately assigning occupancy states based on detection results. These states include "occupied" when a person is detected in the designated zone, "hold" when objects are detected without human presence, and "empty" when no objects or persons are detected.

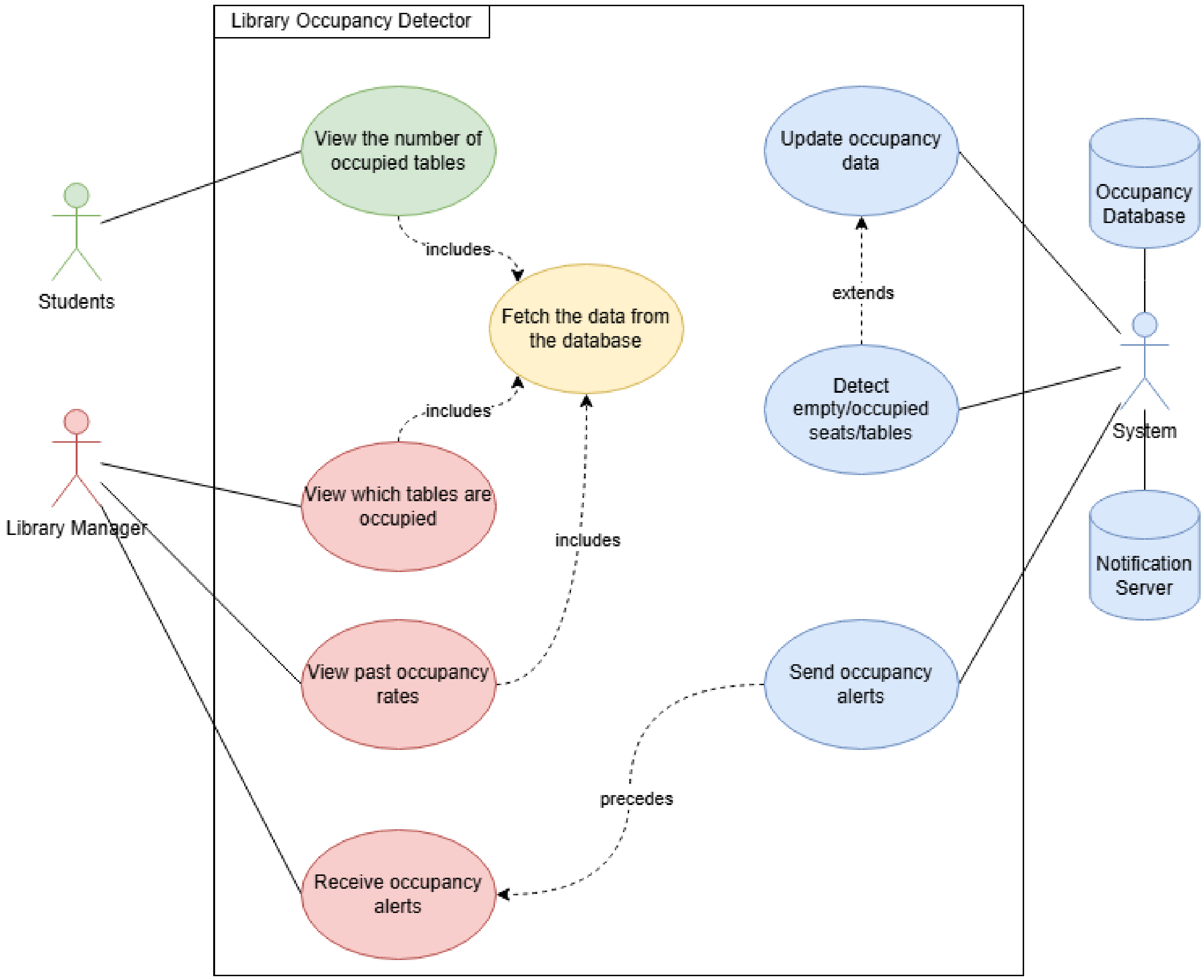
The data management layer utilizes Google Firebase Real-Time Database to store and manage occupancy information. This layer maintains real-time occupancy status for each monitored position, stores configuration parameters for system settings, manages threshold values for hold-time alerts, and ensures data consistency and availability across all system components.

The system provides two distinct interfaces implemented using Flutter. The public interface displays real-time occupancy status of library spaces, provides intuitive visualization of available, occupied, and held spaces, and offers responsive design for multiple device types. The administrative interface enables library staff to configure system parameters, provides hold-time threshold management, and implements a notification system for extended hold situations. Both interfaces are designed with user experience in mind, ensuring efficient access to relevant information.



Activity Diagram for the Overall System (Created by Using draw.io)

As can be seen in the activity diagram, our Flutter application provides the library information for the library manager and students. Even though we thought that we would implement the occupancy data to myTEDUPortal and TEDU App, we could not finish the implementation on those 3rd party applications. We must point that this process is not complicated because we have the data in our database and all that needs to be done is to send the necessary queries to the database from third-party applications.



Use Case Diagram of the Overall System (Created by Using draw.io)

As can be seen in the case diagram, the system serves two primary user groups: Students and Library Managers. Students can view real-time information about the number of occupied tables, while Library Managers have additional capabilities including viewing specific table occupancy status and analyzing historical occupancy rates. The system's core functionality is managed by automated processes that detect empty or occupied chairs and tables, update the occupancy database accordingly, and send relevant alerts. This information is retrieved from a central database that maintains current and historical occupancy data. The system also integrates with a

Notification Server to send occupancy alerts to Library Managers, helping them make informed decisions about space management.

Our current implementation of the system satisfies the needs of the library employees and managers. All the actions stated in the use case diagram except the notifications are available and tested with necessary test cases. Also, a demonstration and basic usage scenarios are presented to the library manager and approved by them.

We collaborated with our supervisor, library management, the IT department, and the administrative affairs directorate to address key topics, including the placement, installation, and costs of the cameras. Throughout the project and demo phases, we maintained regular discussions with the IT and administrative affairs teams to exchange ideas and ensure progress.

4. Development/Implementation Details

The Library Occupancy Detector system employs multi-layered architecture integrating computer vision, cloud services, and applications to provide real-time monitoring of library space utilization.

To state the hardware implementation, the primary hardware component consists of strategically positioned ceiling cameras overlooking library desks and chairs. These cameras were carefully installed to maximize coverage while maintaining optimal viewing angles for accurate detection. The camera positioning ensures comprehensive coverage of designated study areas while minimizing occlusion and blind spots. We used only two cameras to test our system and show the process with the customer, namely the library administration.

The core of the detection system utilizes the YOLO object detection algorithm, implemented through the following process:
Frame Extraction: The system continuously captures video feeds from the ceiling-mounted cameras and extracts individual frames for processing.
Object Detection: Each frame is processed through the YOLO algorithm utilizing the COCO dataset, which enables the detection of common objects including people and personal belongings.
Space Classification: The system implements a spatial mapping algorithm where predefined areas corresponding to desk positions are monitored for occupancy. These spaces are classified into three distinct states, "Occupied": When a person is detected within the designated area, "Hold": When personal belongings are detected without a person present, "Empty": When no objects or people are detected in the space.
Database Integration: The system utilizes Google Firebase Real-Time Database for data storage and management. This implementation enables real-time updates of space occupancy status and persistent storage of historical occupancy data.

We also developed a Flutter application to provide an intuitive interface for accessing occupancy information. We chose Flutter to port the application to website, desktop application, and even mobile application easily. This application has direct integration with Firebase Real-Time Database, and it also allows dynamic configuration options for system parameters by also having the previous occupancy reports. The application also includes an automated alert mechanism for monitoring extended "hold" states. By doing so, library employees can see which places are held for some time duration and move away their belongings to somewhere else to let other students study there. The duration can be set by the library administrators as we stated (changing the threshold according to the need for space in the library).

For other needs of the system, the complete system integration involves secure communication channels between cameras and processing units, real-time data synchronization between the detection system and Google Firebase, two-way communication between the database and mobile application, and implementation of error handling and recovery mechanisms. For the performance optimization of the system, we did some frame processing optimization to maintain real time performance such as capturing the frames every 10 seconds instead of every second and recording the results for a minute. Then updating the database according to the results recorded in one minute. By doing this, we are not overwhelming the database server and the YOLO algorithm.

We also considered both the local and server-based implementation options and concluded that if the system is to be run locally (either by placing computers in libraries that will be used only for this purpose, or by integrating them into the computers already used by librarians), it will need a computer that is always on when the system is needed. This computer does not need high-end system requirements, but it needs to be connected to always power and connected to the cameras (regardless of the type used). Using this type of system can cause problems if too many USB/Ethernet cables are used. The more cable inputs, the more power required to process the image.

Also, we tested the cameras and YOLO and concluded that the accuracy of YOLO will decrease for cameras with a viewing angle of more than 120 degrees or fisheye cameras. However, the distortions caused by the angle of fisheye cameras can be corrected with methods such as undistortion techniques or model training.

5. Test Cases and Results

Test ID: 00101

Test Type/Category: Functional Testing

Summary/Title/Objective: Proper Receiving of the Video Stream from the Camera

Procedure of testing steps:

- Check that the connected camera is available.
- View the video stream through the application.
- Ensure that the video stream is indefinite and clear.

Expected Results/Outcomes: The camera video stream should be displayed correctly and without interruption on the application.

Actual Results/Outcome:

The video stream was received and displayed without interruptions or lag. The image quality remained clear and stable throughout the test.

Priority/Severity: Critical

Test ID: 00102

Test Type/Category: Functional Testing

Summary/Title/Objective: Detection of Table Occupancy (Empty/Occupied)

Procedure of testing steps:

- Simulate empty and occupied tables in the camera feed.
- Check that the application can identify that status of each table - empty/occupied.

Expected results/outcomes: The system should identify correctly which tables are empty or occupied and display the same.

Actual Results/Outcome:

The image processing algorithm correctly detected the status of each table in various scenarios (empty, hold, occupied) but some objects are still cannot detected clearly. So there are some false positives.

Priority/Severity: Critical

Test ID: 00103

Test Type/Category: Functional Testing

Summary/Title/Objective: Display of Table Status in the Application Interface

Procedure of testing steps:

- Simulate different table statuses (empty or occupied) in the application interface.
- Ensure that each table's status displays correctly on the interface.

Expected results/Outcome: The table status (empty/occupied) should reflect its true state within the application interface.

Actual Results/Outcome:

The table status displayed on the interface was accurate and reflected the true state of each table. The interface updated in real-time without delay.

Priority/Severity: High

Test ID: 00201

Test Type/Category: Non-functional testing

Summary/Title/Objective: System Latency and Performance test

Procedure of testing steps:

- Capture the response time of the system with various video streams
- No latency should be more than 1 second expected.

Expected results/Outcome: The table status will be shown by the system towards 2 seconds without having realizable delay.

Actual Results/Outcome:

The system displayed table status within 1 seconds with negligible latency, even under multiple video feed conditions.

Priority/Severity: Critical

Test ID: 00202

Test Type/Category: Non-functional Testing

Summary/Title/Objective: System Load and Performance Test

Procedure of testing steps:

- Start live feeds from different television cameras and watch how well the system performs under load.

Expected results/Outcome: The system should be able to process multiple video feeds simultaneously without performance degradation or crashing.

Actual Results/Outcome:

The system successfully handled multiple video feeds simultaneously without any degradation in performance or system crash. The processing remained stable and consistent.

Priority/Severity: Critical

Test ID: 00301

Test Type/Category: Hardware Testing

Summary/Title/Objectives: Stability of Camera Connections

Procedure of Testing Steps:

- Perform tests on cable connections of the cameras.
- Finally, observe that both cameras are left connected and functioning without disconnection.

Expected Results/Outcome: The connection to camera should be stable without system interruption for continuous video feed.

Actual Results/Outcome:

The camera connections remained stable throughout the test. No disconnections or interruptions were observed, ensuring continuous video feed.

Priority/Severity: Critical

Test ID: 00302

Type/Category of Test: Hardware Testing

Summary/Title/Objectives: Camera Area of View and Angle Adjustment

Procedure of testing steps:

- Test the camera's field of view by tilt and pan.
- Check whether the camera can view the entire area of interest, for example multiple tables or seats in each library.

Expected Results/Outcome: The camera should have an adequately wide field of view to not require additional cameras.

Actual Results/Outcome:

The webcams with low field of view used in test which proved below 120 degrees cameras are insufficient for covering a single 12-person table.

Priority/Severity: High

Test ID: 00401

Test Type/Category: User Interface Testing

Summary/Title/Objective: Table Status Display on User Interface

Procedure of testing steps:

- Launch the application and ensure the interface loads correctly.
  Check that each table's occupancy status (empty/hold/occupied) is clearly displayed on the interface.
- Verify that the table status is updated in real-time.

Expected Results/Outcome:

The table statuses (empty/hold/occupied) should be clearly displayed on the user interface. The interface should update the status of each table in real-time as the occupancy changes.

Actual Results/Outcome:

The table statuses were clearly displayed, and updates occurred in real-time. However, the UI design could be improved.

Priority/Severity: High

6. Other Project Elements

6.1 Contemporary Issues and Additional Considerations

6.1.1 Global Considerations

The Library Occupancy Detector has a clear potential for application on a worldwide scale. Every library faces the same problems; and therefore, such a system can be applied to any library of any region. It comprises very basic and available components, such as ceiling-based cameras and cloud storage solutions; thus, it will function with minimum customization in various regions. The system contributes to the global educational equity of library resource utilization. Hence, it allows footfall optimization concerning students and researchers in libraries, regardless of whether those libraries are in developed or developing countries. Aside from that, the open-source technologies used, like YOLO and Firebase, allow the solution to reach a wider global audience. Such implementations put no caps on budgets, allowing libraries to use the system however rich or poor the institution is, succumbing to the easy global adaption and collaboration of the system.

6.1.2 Cultural Considerations

Sensitivity to culture is core to the design and deployment of the system. As academic and cultural hubs, libraries become invaluable to the system because of their nature of intrusiveness-free monitoring; instead of tracking people personally, ceiling cameras are used. This ensures privacy, an important issue in individual-rights-dependent societies, as well as data protection.

Additionally, it will also be designed for those varying cultural values. The system could allow, for example, one to adjust the duration of "hold" for unattended personal items to span between usages to align with local sensibilities on unattended personal stuff. Thanks to the flexibility of the alert mechanism, the system is maintained across different cultures.

6.1.3 Social Considerations

In a societal perspective, the system encourages equal chances to access library resources and takes care of inequity coming from disturbance or vacancy duration. It would use such dynamic spaces to enable these libraries to accommodate more persons, especially in their peak times. It would portray a sense of fairness and community among fellow students and library patrons.

The system helps library personnel by automating tiresome monitoring duties, allowing them to devote time to more significant user dealings. Furthermore, historical occupancy usage allows libraries to plan for their future expansions or configurations based on actual usage patterns, strengthening their role as sociocultural learning spaces.

6.1.4 Environmental Considerations

Libraries can avoid unnecessary, usually resource-expensive, extension or remodeling by making better use of the available space. Such extensions reduce underused spaces. Besides energy expenditure savings, better heating, cooling, and lighting systems can be matched with real occupancy.

The hardware specifications of this system are low because of its low-energy camera and algorithm. Furthermore, the capturing of a frame at interval distances reduces the computation demand, thus reducing the local and cloud energy consumption.

6.1.5 Public Safety Considerations

Public safety is at the forefront of the system's development. The use of ceiling-mounted cameras ensures that no personal data is directly captured, safeguarding user privacy. All communication to Firebase Real-Time Database is encrypted so that it cannot fall into unauthorized hands.

Indirectly contributes to safety in that it identifies unattended belongings, which can be sources of safety risks in public areas. Prolonged "hold" states create alerts, which can stimulate staff to act, hence reducing the chances that lost or suspicious objects disturb library operation

6.2 Ethics and Professional Responsibilities

6.2.1 User Privacy and Data Protection

This system has been designed to respect the privacy of users at every level. A video feed from the ceiling-mounted cameras is processed in real-time occupancy states, without saving any information which is personal or identifiable. By omitting to keep raw video data and concentrating solely on occupancy metrics, the system ensures compliance with tight privacy regulations like GDPR or KVKK.

The communication between camera/cameras- processing unit and Firebase Real-Time database is also put on secure encryption protocol so that it can safeguard against unauthorized access to obtain data and breach, thus further strengthening trust and privacy for a user.

6.2.2 Equity and Social Justice

This system has been designed to respect the privacy of users at every level. A video feed from the ceiling-mounted cameras is processed in real-time occupancy states, without saving any information which is personal or identifiable. By omitting to keep raw video data and concentrating solely on occupancy metrics, the system ensures compliance with tight privacy regulations like GDPR or KVKK.

The communication between camera/cameras- processing unit and Firebase Real-Time database is also put on secure encryption protocol so that it can safeguard against unauthorized access to obtain data and breach, thus further strengthening trust and privacy for a user.

6.2.3 Responsible Use of Technology

Through computer vision and cloud computing technologies, the initiative would be responsible for such uses, maintaining efficiency and resource effectiveness; truly, the project would be green and ethically integrated engineering tools. Besides, the system does not engage or practice invasive methods such biometric recognition or tracking, consistent with ethical guidelines for AI and machine learning applications.

6.2.4 Commitment to the Professional Standards

The team strictly abided by the ethical principles stipulated among others by the IEEE and ACM during all the development phases. Such standards directed all decisions with respect to user safety, data integrity, and social concern, standing tall for professional responsibility at all stages of the project.

In that sense, the Library Occupancy Detector system discusses ethical and professional issues with an understanding approach and accordingly resolves the critical issues of privacy, equity, and responsible use of technology in favor of society.

6.3 New Knowledge Acquired and Applied

6.3.1 Description of Tools Used

- **YOLO Algorithm:** A real-time object detection framework used to identify people and personal belongings in library spaces.
- **Google Firebase:** A cloud-based database solution that supports real-time updates and efficient data storage.
- **Flutter Framework:** A versatile UI toolkit enabling cross-platform development of the application for desktop, web, and mobile devices.
- **OpenCV:** OpenCV allows the handling of frames more effectively and minor image corrections, like resizing and undistortion, which makes it compatible for YOLO processing of video frames.

6.3.2 Integration of Technologies in the System

Video Feed Processing

- Continuous video feed was provided from the camera mounted on ceiling.
- OpenCV snapped frames at specified intervals (10 seconds) as opposed to processing every frame to avoid site computation and database load.
- YOLO analyzed each frame to detect objects and classified desk areas into "Occupied", "Hold", or "Empty" statuses.

Data management with Firebase

- Occupancy detected were uploaded to the Firebase real-time database for instant updates.
- The same database was used as a repository for historical data to allow trend analysis and reports through the user application.

User Application

- Flutter-based applications provided real-time occupancy data report to users.
- Firebase SDK was the bridge to connect the application to the database for real-time update and configuration.

System optimization

- Capture and processing frame rates were optimized based on testing balances between real-time performance versus resources consumed.
- Robust error-handling mechanisms will ensure proper operation during the down time of network or hardware failures.

7. Conclusion and Future Work

7.1 Conclusion

The library occupancy detector enables optimization of utilization by using advanced technologies such as cloud-based data management, computer vision, and highly responsive user interfaces. Through effectively detecting and classifying occupancy states, the detector provides better accessibility and efficiency of study areas while safeguarding the privacy and security of the users. The pilot phase using two cameras proved the concept and has laid a solid foundation for future development and deployment.

7.2 Future Work

The Future holds several improvements meant to enhance and extend the system covering various technical advances and the requirements of the institutions. The following can be highlighted:

1. **Algorithm Tuning for Better Accuracy**
   The detection algorithm will also be fine-tuned further to yield better accuracy in detection under extreme conditions or for wide-angle cameras. Hyperparameter tuning, more model training on custom datasets, and improved preprocessing will be tried to minimize errors in detection.
2. **Planning for the Transition to Institutional Infrastructure**
   The development will be directly migrated from Firebase to the institutional servers for better control, security, and reliability. MySQL shall replace the current database solution to provide advanced relational database features that correlate with those of the institution and facilitate more efficient data query and management.

3. **Backend System Development and API Integration**

   This will set up a strong backend to smoothen processes for the data and provide central control over the whole system. The APIs allow a smooth connection between detection system, database, and user interfaces for integration with any of the library systems, such as reservation platforms or resource tracking.

4. **Enhanced UI with Advanced Features**

   The upgrade includes the user interface functionality, such as peak time analysis, use history trends, and customizable reporting. These improvements will help library administrators make data-backed decisions, assist in optimizing space allocation and predicting future demand patterns.

5. **Scalability and Deployment of the System**

   The expansion of this project would include the entire library, including more cameras, and would add functionalities to other study areas. The system architecture will be modified to accommodate increased database loads in the larger deployment and ensure performance consistency all the way.

6. **Feasibility and Planning**

   A feasibility study has been prepared in detail for the transition and scaling process without technical, operational, and financial concern. This document will help in aligning future development with the objectives of the institution.

The enhancement has been built to ensure that the Library Occupancy Detector system will also be able to further enhance accuracy, scalability, and institutional integration. In fact, these improvements exhibit an ongoing commitment for evolving towards greater application of the model as a system tool intended for the optimization of library operations and enhancing user experience.

8. References

1. YOLOv10 Algorithm: YOLOv10
2. Software Architecture for Computer Vision: ResearchGate - Software architecture for computer vision: Beyond pipes and filters
3. Proposed ICPS Real-Time Monitoring System: ResearchGate - Architecture of the proposed ICPS real-time monitoring system
4. diagrams.net [For other consumer use]. diagrams.net
5. Real-Time Architecture Considerations: Upsolver Blog - Building a Real-Time Architecture: 8 Key Considerations