Wild Commons

Course Name: Integrative Project

Rodolfo Abdiel Macias Marez 4176

Jared Rodriguez de la Cruz 4371

Gael Esau Ruiz Abundez 4790

Jesus Javier Santos Cervantes 4578

Rodrigo Zamacona de la Concha 4256

29/09/2024

Index

Project Description

The project revolves around developing a mobile application that uses two tablets for managing the entry and exit of users into Learning Commons, built with Android Studio using Java and Object-Oriented Programming (OOP) principles. The backend is supported by a MySQL relational database.

Objective

The project aims to improve the management and usage of university library resources through a mobile application.

Real-Time Registration and Monitoring:

- Implement a system that allows users to register and reserve library resources in real-time (general room, computers, cubicles).
- Monitor the use of resources via the application so the administrator can see in real time which resources are busy, free, or under maintenance.

Administrator Resource Control:

- Provide the librarian with tools to manage resources, such as forced session closure, viewing detailed usage reports, and controlling resources when students fail to release them properly.
- Offer an administrative interface for accessing reports or putting resources under maintenance as needed.

Improvement of User Experience:

- Create an intuitive mobile application that allows users to easily reserve and release spaces, using an interactive map that shows the status of the resources.

Boundaries of the Project

Included in the Project:

Student Resource Reservation System:

- The application will allow users to reserve resources like the general study room, specific computers and cubicles.

- A real-time map will be provided, indicating the availability status of each resource (free, busy, or under maintenance).

Session Management for Students:

- Users will be required to check in and check out.

Administrator Control Panel:

- The administrator will have a dashboard with access to resource availability, usage reports, and user logs.
- Administrators can override reservations, manage maintenance schedules, and force logouts if resources are being misused.

Database Management:

- A relational MySQL database will be implemented to store user information, reservations, resource statuses, and session logs.
- The system will support multi-table relationships for users, resources, reservations and complaints.

Report Generation:

- Provide the users with the ability to generate detailed usage reports, facilitating decision making regarding preventative maintenance, adjustments to resource capacities, or improvements to library infrastructure.

Not Included in the Project:

Hardware Integration (IoT Devices):

- The project does not include hardware components such as sensors or IoT devices for real-time tracking of student presence in rooms or at computers. The system will rely solely on user interaction with the application.

Multi-Branch Library System:

- The project focuses on a single library. If the university has multiple branches or locations, those will not be covered under this initial version of the application.

Detailed Inventory Management:

- While resource availability (occupied/free/maintenance) is managed, the project does not extend to inventory control for physical assets (e.g., tracking library books, printers, or equipment) beyond the designated study resources.

Complex User Roles Beyond Student/Teacher/Administrator/External:

- The system will support four primary user roles: students, professors, administrators and externals. Other specialized roles (e.g., IT staff, library assistants) will not be included.

Resource Usage Metrics Analytics:

- The project will generate basic usage reports for the administrator, but advanced data analytics or predictive maintenance features will not be part of this scope.

Security Measures:

- This system is for informational purposes only. It is not intended for use in a production environment and does not include any security measures. Authentication is not implemented beyond basic user role differentiation.

Tasks and Milestones

1. Planning and Initial Setup

- Clearly define the project requirements and scope.
- Set up the development environment in Android Studio and establish the MySQL database.

2. Creation of the Database

- Create database tables for users, resources, reservations, and sessions.

- Develop a role-based system for students, professors, administrators and externals.

3. Development of the Reservation System for Students

- Build a system allowing students to reserve resources such as general room, computers, and cubicles.

- Implement an interactive real-time map showing the availability of resources.

- Add check-in and check-out features to record resource usage by students.

4. Development of the Administrator Control Panel

- Create an admin panel to monitor resource usage and generate usage reports.

- Include tools for forced session closures and resource maintenance management.

5. Testing and Debugging

- Perform unit tests on each system component.

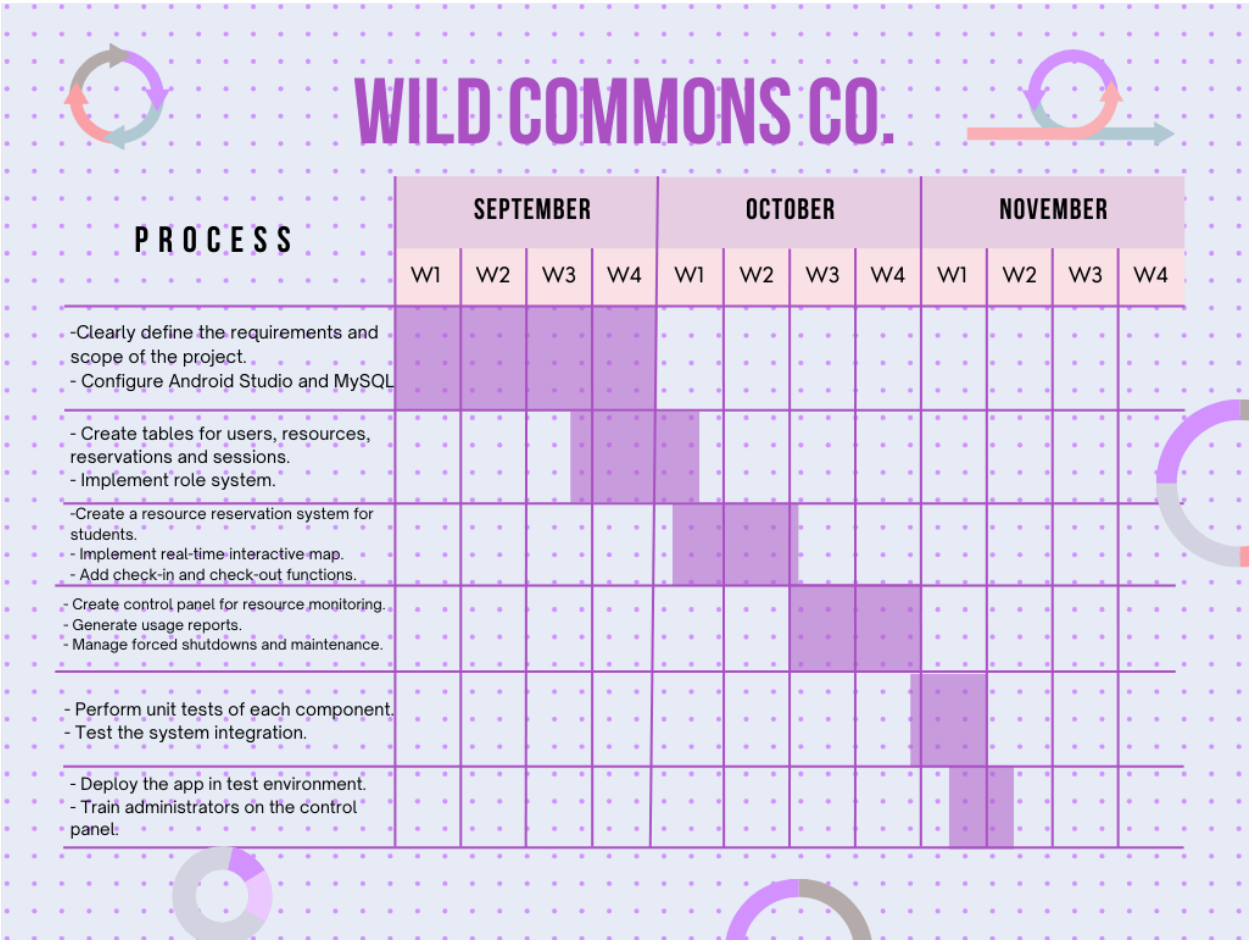
- Test the integration between the database, authentication, and reservations.

6. Deployment and Training

- Deploy the app in a testing environment to gather feedback.
- Train administrators on how to use the control panel.

- Completion of planning and database design.
- Fully functional database.
- Operational reservation and student check-in/check-out system.
- Implemented and functioning admin control panel.
- Active group reservation system for professors.
- Operational automated notifications and resource release.
- Completed testing and debugging, application ready for release.
- Final deployment and training completed, ready for general use in the library.

Gantt Chart


WILD COMMONS CO.
PROCESS
SEPTEMBER
OCTOBER
NOVEMBER
W1 W2 W3 W4 W1 W2 W3 W4 W1 W2 W3 W4
-Clearly define the requirements and scope of the project.
- Configure Android Studio and MySQL
- Create tables for users, resources, reservations and sessions.
- Implement role system.
-Create a resource reservation system for students.
- Implement real-time interactive map.
- Add check-in and check-out functions.
- Create control panel for resource monitoring.
- Generate usage reports.
- Manage forced shutdowns and maintenance.
- Perform unit tests of each component.
- Test the system integration.
- Deploy the app in test environment.
- Train administrators on the control panel.

List of Resources Needed

Software:

- Android Studio: IDE for Android application development.
- Java Development Kit (JDK): To compile and run Java applications.
- Android SDK: Tools required for development in different Android versions.
- Android Emulator: For testing the app in various resolutions.
- MySQL: Database management system for the backend.
- MySQL Workbench: Visual administration tool for databases.
- Visual Studio Code: A versatile and lightweight code editor with excellent extensions for web development.
- XAMPP: A free software distribution that includes MySQL as its default database server, enabling developers to create and manage databases for their web applications.

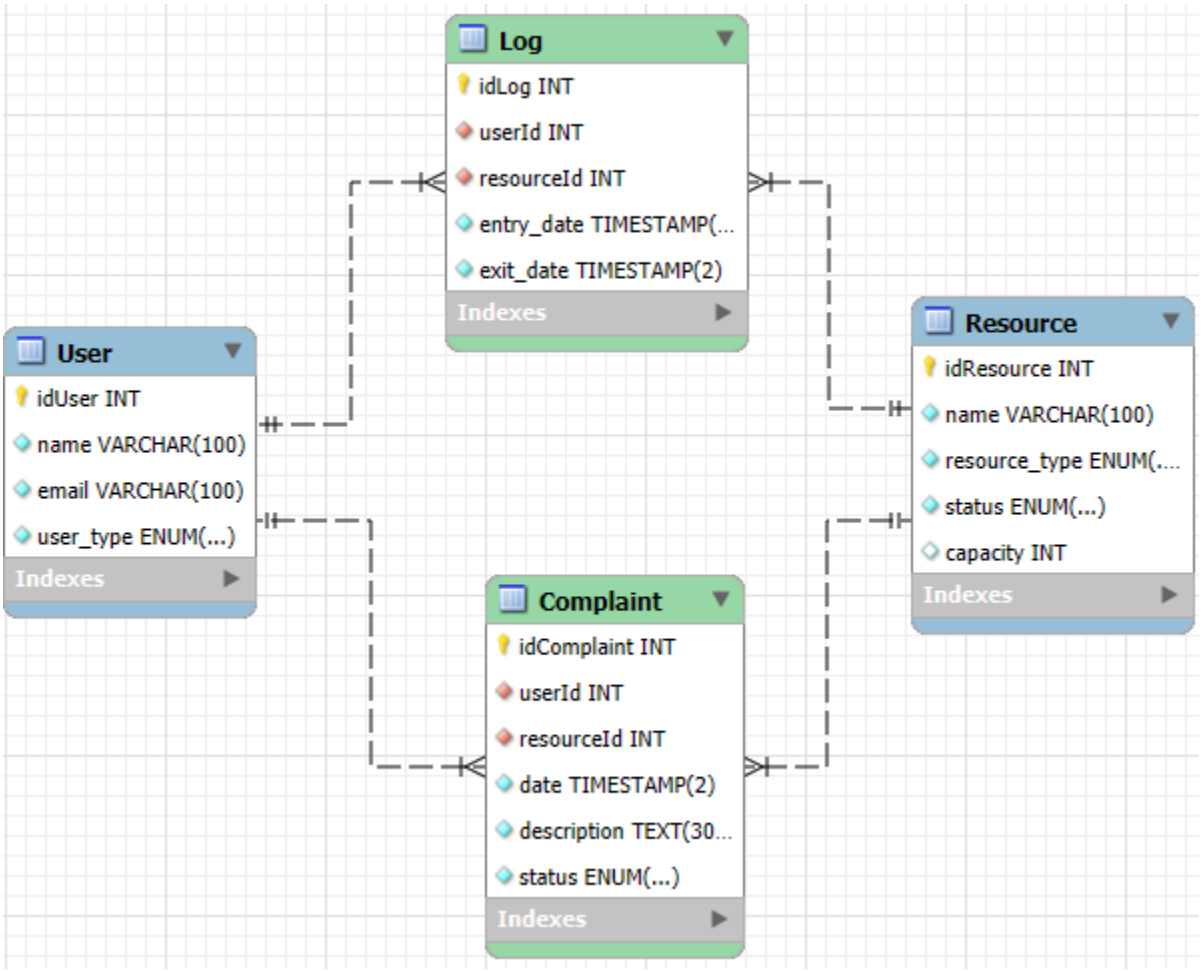
Hardware:

- Computers: Devices with i5/i7 or AMD Ryzen processors, 8 GB of RAM, and SSD.
- Physical Android Devices: For real device testing.

Programming Languages

- JAVA: Development of register system and additional database configurations. "Java's platform independence, combined with its extensive libraries and security features, makes it a reliable choice for developing enterprise-grade applications. [1]" .
- MySQL Server: Main database development. "Relational databases enable efficient data management by structuring information into tables and ensuring relationships between datasets through referential integrity. [2]".
- PHP: Server-side scripting for web application development.
- SQL: Database query language for managing and manipulating data.
- JavaScript/HTML: To design and develop a web page for the admin.

Database Schema and Relationships



Tables and Attributes

- User
  - idUser: Primary key, unique identifier for each user.
  - name: User name.
  - email: User email.
  - user_type: User type, stored as an ENUM that could include values like 'student', 'teacher', 'administrative', 'external'.
- Resource
  - idResource: Primary key, unique identifier for each resource.
  - name: Resource name.
  - resource_type: Resource type, stored as an ENUM that could include values like 'general_room', 'computer', 'cubicle'.
  - status: Resource status, stored as an ENUM that could include values like 'free', 'busy', 'maintenance'.
  - capacity: Resource capacity, relevant for resources like cubicles.

- Log
    - idLog: Primary key, unique identifier for each log record.
    - userId: Foreign key referencing User(idUser).
    - resourceId: Foreign key referencing Resource(idResource).
    - entry_date: Date and time the user entered the resource.
    - exit_date: Date and time the user left the resource.
- Complaint
    - idComplaint: Primary key, unique identifier for each complaint.
    - userId: Foreign key referencing User(idUser).
    - resourceId: Foreign key referencing Resource(idResource).
    - date: Date and time the complaint was logged.
    - description: Textual description of the complaint.
    - status: Status of the complaint, stored as an ENUM that could include values like 'pending', 'resolved'.

Relationships

- User to Log: One-to-many relationship, where a user can have multiple log entries but each log entry is associated with a single user.
- Resource to Log: One-to-many relationship, where a resource can be associated with multiple log entries, but each log entry refers to a single resource.
- User to Complaint: One-to-many relationship, where a user can have multiple complaints, but each complaint is filed by a single user.
- Resource to Complaint: One-to-many relationship, where a resource can be associated with multiple complaints, but each complaint is associated with a single resource.

Network Setup

All the devices are going to be connected to the local internet network from the Learning Commons to fully leverage the system functionality.

- Tablets: We need two tablets, one to register the user entrance and another one to register the exit from the library. Both tablets will connect to the library's local network via Wi-Fi.
- SQL Server: The SQL Server is located on the library's local network and will connect to the network via Ethernet. This will ensure a continuous connection and minimize interruptions and interference.
- PC: The admin PC will have access to a special web page to administrate the system. This device already counts with a functional Ethernet connection which will remain.

System architecture

The software architecture described is a distributed software architecture, specifically a client-server architecture. "Client-server architecture is essential in distributed computing as it separates user interfaces from data storage, ensuring scalability and manageability. [3]"

In this architecture, there are several components that communicate with each other:

Clients: The input and output tablets, which are the points of interaction with users.
Server: The application server, which processes client requests and manages the database.
Database: The MySQL server, which stores user information, resources, and usage logs.
Communication between components is done through the library's internal network, using the HTTP protocol and the Android Volley library for communication between tablets and the server.

The software architecture can be described as follows:

Presentation layer: The input and output tablets, which provide the user interface and allow users to interact with the system.
Application layer: The application server, which processes client requests and manages the database.

Data layer: The MySQL server, which stores user information, resources, and usage logs.

Infrastructure layer: The internal network of the library, which provides connectivity between components.

Detailed Description

1. Tablet Application (Entry and Exit)
    - Entry Application: This application will be installed on the tablet located at the entrance of the library. It will allow users (students, teachers, administrative, and externals) to check in by selecting the resource they wish to use. Users will enter their ID, and the application will display in real time the availability of the resources (general room, computers, cubicles) for them to choose what to use.
    - Check Out Application: Installed on the exit tablet, this application will allow users to check out of the library. In addition, it will offer the option to register complaints or comments about the resources or the service received, feeding directly into the complaints database.

2. Administrator Web Interface
    - The web interface will be accessible from any computer within the library network, providing the administrator with a real-time view and full control over resources and records. They will be able to see who is using which resource, manage the status of the resources (free, occupied, under maintenance), and review complaints and historical usage records.
3. Back-end (Application Server)
    - Server: Hosted on a central library computer, it processes all requests from both entrance and exit tablets, handling user authentication, resource booking, session tracking, and complaint management.
    - Volley Implementation: Android's Volley library is used for network communication between the tablets and the server. Volley handles HTTP requests efficiently and manages the network queue, making it ideal for operations like sending user data and receiving updates from the server.

4. Database (MySQL Server)
    - Structure: Hosts tables for users, resources, logs of resource usage, and complaints. Ensures data integrity and handles concurrent access to resource records.
    - Location: Located on the same network as the application server and admin’s computer, potentially hosted on a dedicated server or cloud environment but within the library’s secure network to ensure data security and quick access.
    - Management: Administered by the library’s IT staff, with regular backups and maintenance to ensure data availability and integrity.
5. Network Infrastructure
    - Configuration: Both tablets and the administrator's computer are connected to the library's internal network. This setup facilitates fast communication with the back-end server and database.}

Potential Risks and Strategies to Mitigate Them

1. **User Authentication Failures**: Issues with logging in or registering users, particularly students and teachers, could lead to delays in accessing the services.
    a. Provide clear error messages and troubleshooting steps for users experiencing login issues.
2. **Hardware Malfunction**: The tablets at the entrance or exit might face technical failures, causing inaccurate tracking of entries/exits or unavailability of services.
    a. Use high-quality, durable tablets and equipment to minimize the risk of malfunction.
3. **System Overload**: High demand during peak hours might overload the system, causing it to become unresponsive or slow.
    a. Optimize database queries and system performance to minimize response times.
4. **Data Loss or Corruption**: Power outages or system crashes could result in lost or corrupted data, such as service usage records or group registrations.

    a. Implement data validation and correction mechanisms to detect and correct data inconsistencies.
    b. Implement system monitoring to detect system crashes or power outages quickly.
5. **User Errors**: Users (students/teachers) may accidentally select the wrong service or input incorrect data when registering groups or using the system.
    a. Develop a user-friendly interface that minimizes the risk of user error.
6. **Inaccurate Tracking**: Issues with tablet synchronization between entry and exit could lead to inaccurate usage tracking.
    a. Implement a robust synchronization mechanism between entry and exit tablets.
7. **System Maintenance**: Regular updates or maintenance could temporarily disrupt service availability, impacting the user experience.
    a. Develop a maintenance schedule that minimizes service disruptions.
    b. Implement rolling updates and deployments to minimize downtime.

Testing Approach

1. **Unit Testing**: Each function or component of the system (e.g., user registration, service selection) will be tested individually to ensure that it behaves as expected in isolation.
2. **Integration Testing**: After unit tests, components will be combined and tested together, ensuring that interactions between tablets (entrance and exit), service selections, and group registrations work smoothly.
3. **System Testing**: A full system test will simulate real-world usage to check overall functionality, including tracking entry/exit, service allocation, and group registrations.
4. **User Acceptance Testing (UAT)**: End-users (students, teachers, librarians) will test the system to validate that it meets their requirements and works effectively in the library environment.
5. **Performance Testing**: The system will be tested under varying loads (e.g., peak hours) to ensure it can handle high traffic without crashing or slowing down.

List of Test Cases with Expected Outcomes

1. **User Registration (Student)**
    - **Input**: Valid student ID, personal details.
    - **Expected Outcome**: User successfully registers and can access services.
2. **Service Selection (Computer)**
    - **Input**: A student selects "Computer."
    - **Expected Outcome**: A computer station is assigned, and usage is tracked.
3. **Service Selection (Cubicle)**
    - **Input**: A student selects "Cubicle."
    - **Expected Outcome**: A cubicle is assigned, and the system reflects availability status.
4. **Tablet Sync (Entry/Exit)**
    - **Input**: Student checks in at the entrance and checks out at the exit.
    - **Expected Outcome**: Accurate tracking of entry and exit times, with a complete record stored in the system.

5. **Error Handling (Invalid Student ID)**
    - **Input**: An invalid or incorrect student ID is entered.
    - **Expected Outcome**: System displays error message and prevents access.
6. **System Overload During Peak Hours**
    - **Input**: High volume of simultaneous service selections and group registrations.
    - **Expected Outcome**: System remains responsive, with no delays or crashes.
7. **Security Test (Unauthorized Access)**
    - **Input**: Attempt to access the system without valid credentials.
    - **Expected Outcome**: System denies access and logs the attempt for security review.
8. **Data Backup and Recovery**
    - **Input**: Simulate a power outage.
    - **Expected Outcome**: System automatically backs up data, and no information is lost once the system is restored.
9. **Accessibility Test**

- **Input**: Attempt to use the system with accessibility tools (screen readers, large text).
- **Expected Outcome**: System remains fully usable by individuals with disabilities.

Team Roles

Jared Rodriguez de la Cruz: Responsible for designing the database and system architecture, as well as generating reports and implementing the authentication system.

Gael Esau Ruiz Abundez: Responsible for identifying project risks and defining the testing approach (unit and integration tests).

Jesus Javier Santos Cervantes: Responsible for defining roles, coordinating team communication, summarizing key points of the project plan, and establishing future steps.

Rodrigo Zamacona de la Concha: Responsible for network setup and risk management.

Rodolfo Abdiel Macias Marez: Responsible for the student reservation system and coordinating the progress steps.

Communication

Personal meetings: These will be in-person to discuss progress, identify obstacles, and adjust the plan as needed.

Emails: Used to share important documents, project status updates, and coordinate between meetings.

Instant messaging (WhatsApp): For quick communications and real-time problem solving.

Collaborative documents (Google Drive): To store and share files in a centralized manner, ensuring that everyone has access to the latest version.

Discord: For team communications and real-time collaboration, including text and voice channels to discuss specific topics and share resources.

Costs Estimation

1. Infrastructure Costs:

- **Tablets (for entry, exit, and admin interfaces):**
  - Estimate: 3 tablets (touchscreens) at MXN $7,000 each (good quality tablets for the job)
  - Total for Tablets: 2 × $7,000 = $14,000 MXN
- **Wall Mounts and Chargers:**
  - Cables, chargers, and mounting systems for wall installation.
  - Estimate: MXN $2,000 per tablet (including high-quality wall mounts and charging systems)
  - Total for Accessories: 2 × $2,000 = $4,000 MXN
- **Server and Networking Setup:**
  - Assuming the central library computer will act as the server, we only need to consider the cost of potential upgrades, cables, and configuration.
  - Estimate: MXN $5,000 for basic networking infrastructure and configuration.

- **Total Infrastructure Costs:**
    - $14,000 (tablets) + $4,000 (mounts/chargers) + $5,000 (server setup) = $23,000 MXN

2. Software & Services:

- **Software Development Tools**:
    - **Android Studio** and **Java** are free, so no cost here.
- **MySQL Database**:
    - **MySQL Community Edition** is open-source and free for this type of project.
- **Backup & Security**:
    - If basic **data backup** and security services are required, such as cloud backup, it could range from **$1,000 - $5,000 MXN** for initial setup.

3. Full Stack Developer Salaries:

- Estimate for all programmers: **MXN $30,000** for the entire project.
- For **5** programmers: **5 / $30,000 = $6,000 MXN**

4. Additional Costs:

- **Miscellaneous expenses** (unexpected hardware upgrades, maintenance, software licenses, etc.): Estimate around **$8,000 MXN**.

Total Estimated Cost:

- **Infrastructure**: $23,000 MXN
- **Programmer Salaries**: $30,000 MXN
- **Miscellaneous**: $8,000 MXN

Grand Total:

$23,000 + $30,000 + $8,000 = $61,000 MXN

Summarize

The **Wild Commons** project aims to develop an application for resource management in the university library, implementing a real-time reservation and monitoring system. Tablets will be used at entry and exit points, with resources managed through a MySQL database and Android Studio.

Key points include:

- Authentication system for students, professors, and administrators.
- Real-time resource reservations with interactive maps.
- Admin control panel to manage resource usage and generate reports.

Future Plans

Once the project is completed, the following steps are proposed for its evolution:

- Functionality improvements: Add more user roles, integrate new features such as resource usage prediction through advanced data analysis and new systems integrating an easiest access as RFID cards for the members of the UTR.
- Securit: Implement a biometric fingerprint security system to ensure secure access and management of library resources across multiple locations, allowing for institutional-level control and monitoring.
- Hardware integration (IoT): Include sensors to track resource usage in real time, optimizing automatic monitoring without relying on user interaction.
- Monitoring and maintenance: Implement advanced monitoring tools that help predict failures or the need for preventive maintenance to improve resource availability.

References

[1] J. Gosling, B. Joy, G. Steele, and G. Bracha, The Java Language Specification, 3rd ed. Addison-Wesley, 2005.

[2] C. J. Date, An Introduction to Database Systems, 8th ed. Pearson, 2003.

[3] M. van Steen and A. S. Tanenbaum, Distributed Systems, 3rd ed. Pearson, 2017.