

10/07/2024



Integrative Project

WILD COMMONS

Team Members:

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

- **Teacher:** Oliver Aaron Velasco Cárdenas
TIDSM 3º “A”

WHAT IS WILD COMMONS?

● 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.



OBJECTIVES

- **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.

PROJECT BOUNDARIES

Included:

- Student resource reservation system.
- Administrator control panel.
- Real-time resource map and reports.
- Basic session and complaint management.

Not Included:

- IoT devices for real-time tracking.
- Multi-branch library support.
- Inventory management for physical assets (books, printers)
- Advanced analytics for resource usage.
- Security measures beyond role-based access.

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.



MySQL Server | Main database development. Relational databases enable efficient data management by structuring information into tables and ensuring relationships between datasets through referential integrity.



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.

SYSTEM ARCHITECTURE

The system is based on a distributed client-server architecture.

● Client Device (Entry Tablet)

This tablet, located at the Learning Commons entrance, allows users to log in by providing their ID. Upon login, the tablet connects to the SQL server to record the user's session start and the resource they will use.

● Client Device (Exit Tablet)

This tablet, located at the Learning Commons exit, requires users to check out, updating the SQL server to indicate they are no longer using Learning Commons resources.

● Database (MySQL)

Database stores all system entities, including users, complaints, entry/exit logs, resource status, and usage history, and is located on the admin PC for fast, direct access.

● Administrator Page

The administrator accesses a web application from his computer that connects directly to the SQL server.

DETAILED COMPONENT DESCRIPTIONS .

● Tablets

Frontend: Upon startup, a welcome message will be displayed followed by a screen where the user can select their role (student, teacher, administrative or external) and then enter their ID. After validating this data, the user will be redirected to an interactive map that will show in real time the status of the available resources (general room, computers and cubicles), indicating whether they are free, occupied or under maintenance. From here, they can select the resource they wish to use.

Backend: The system utilizes PHP scripts on the XAMPP server as an intermediary between the client app and the database. These scripts handle HTTP requests from the app, process data, and update the database accordingly.

The Android app employs the Volley library to make HTTP requests to the PHP scripts, enabling seamless communication with the MySQL database.

● Administrator PC

The admin portal features three primary sections:

- View logs: Displays who is using which resource, how long they've been using it, and which resources are free or occupied.
- Manage complaints: Administrators can review complaints logged by users about resources.
- Monitor resource status: Real-time updates on resource availability, status changes, and lock/unlock functionalities for cubicles.

NETWORK SETUP

● Tablets:

We require two tablets, one for user check-in and another for check-out, both connecting to the library's local network via Wi-Fi.

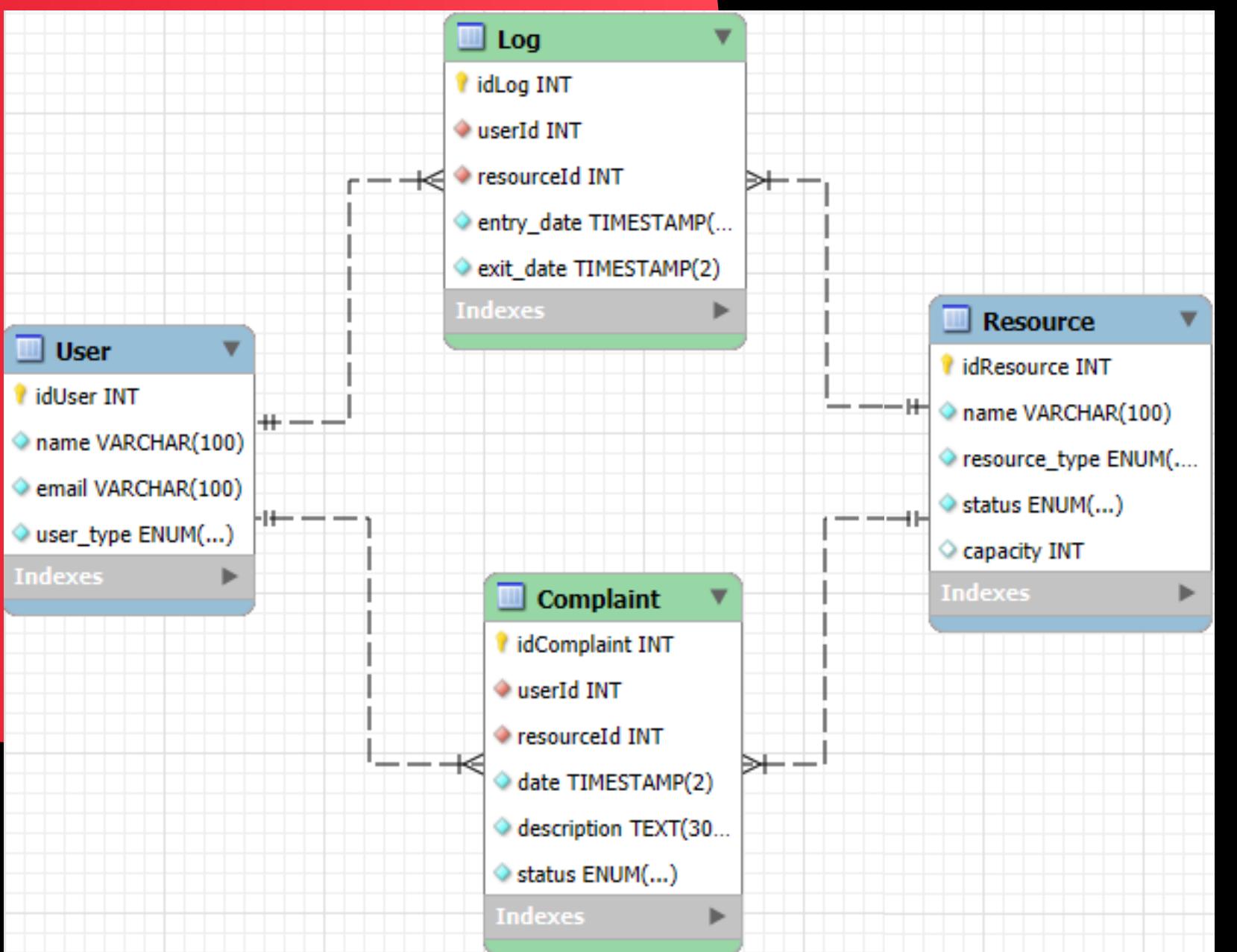
● SQL Server

The SQL Server, situated on the library's local network, will establish a wired Ethernet connection, guaranteeing a stable and uninterrupted link while reducing interference.

● PC

The admin PC, already equipped with a functional Ethernet connection, will access a dedicated web portal for system administration, ensuring continued connectivity.

DATABASE SCHEMA



- **User**
 - user_type: User type, stored as an ENUM that could include values like 'student', 'teacher', 'administrative', 'external'.
- **Resource**
 - 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'.

User: 1 → many Logs

Resource: 1 → many Logs

User: 1 → many Complaints

Resource: 1 → many Complaints

POTENTIAL RISKS AND HOW TO MITIGATE

User Authentication Failures

- Issue: Problems logging in or registering can delay access to services.
- Strategy: Provide clear error messages and troubleshooting guides for quick resolution.

Hardware Malfunction

- Issue: Technical issues with tablets may lead to inaccurate tracking or unavailability of services.
- Strategy: Use high-quality hardware to minimize technical failures.

System Overload

- Issue: High demand during peak hours could cause the system to slow down or become unresponsive.
- Strategy: Optimize system performance to reduce response times.

POTENTIAL RISKS AND HOW TO MITIGATE

Data Loss or Corruption

- Issue: Power outages or system crashes could result in data loss or corruption.
- Strategy: Implement data validation mechanisms and monitor the system for quick recovery.

User Errors

- Issue: Users may input incorrect data or select the wrong service.
- Strategy: Design a user-friendly interface to reduce errors.

Inaccurate Tracking

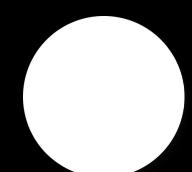
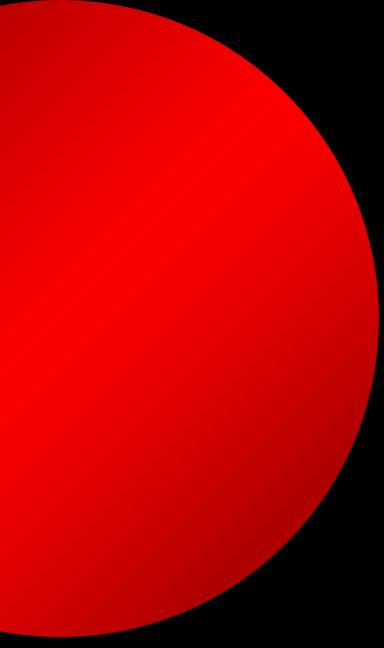
- Issue: Synchronization problems between entry and exit tablets could lead to inaccurate tracking.
- Strategy: Ensure a reliable synchronization mechanism.

System Maintenance

- Issue: Maintenance could disrupt service availability.
- Strategy: Plan updates and maintenance to minimize downtime.

TESTING STRATEGY FOR THE WILD COMMONS PROJECT: ENSURING SYSTEM QUALITY AND FUNCTIONALITY





UNIT TESTING

Unit tests are fundamental in this project to verify the correct functioning of each component in isolation. In this case, they focus on individual system modules, such as functions within the mobile app developed in Android Studio with Java, PHP scripts handling HTTP requests, or interactions with the MySQL database.



PURPOSE

Ensure that each function or method in the app performs its specific task without errors.

Identify issues early in the development cycle.

Testing if the user registration function correctly stores information in the database.

Verifying that resources (cubicles, computers, etc.) are marked as occupied or available in real time based on user interactions.

Examples in the project

● INTEGRATION TESTING

Integration tests are key to ensuring that different system components communicate and work together properly. This includes the interaction between the mobile app, PHP scripts, the XAMPP server, and the MySQL database.

PURPOSE

Guarantee that the interaction between the Android app and the MySQL database, via the API managed by the PHP scripts, functions correctly.

Evaluate the proper synchronization of data between the entry and exit tablets, as well as the admin portal.

Examples in the project

Verifying that when a user logs in at the entry tablet, the system correctly updates the resource's status in the database.

Ensuring that administrators can manage resources (e.g., release a cubicle or put it under maintenance) and that these changes are reflected in real time in the app.



FUNCTIONAL TESTING

These tests ensure that the system meets the functional requirements set for the application, meaning that all designed features for the end-user work properly.

PURPOSE

Verify that users can register and reserve resources, that the interface is intuitive, and that the administrator can access real-time reports.

Examples in the project

Testing if a user can make a reservation and if the reservation is correctly reflected in the interactive map in real time.

PERFORMANCE TESTING

These tests are crucial to ensure that the system can handle a high number of users and requests in real-time without performance issues. Since the system is designed to be used by multiple users simultaneously, these tests are essential.

PURPOSE

Evaluate the system's capacity to handle peak loads (e.g., during times of high resource usage in the library).

Examples in the project

Simulating multiple users interacting with the app at the same time and measuring the response times of the MySQL server and tablets.

● TEST CASES AND EXPECTED RESULTS FOR THE WILD COMMONS SYSTEM

This table presents the test cases and expected results for the Wild Commons system, a mobile application for managing user entry and resource reservations. Each case is designed to verify key functionalities, such as user registration and real-time updates. The expected results serve as benchmarks to ensure the system operates effectively and meets user needs, facilitating early identification of any issues.

Test Case	Description	Expected Result
1. User Registration	Verify the registration of a new user.	Successful registration and confirmation on screen.
2. Login	Validate login with correct ID.	User accesses the resource screen.
3. Resource Reservation	Test reserving an available resource.	Resource marked as occupied with reservation confirmation.
4. Release Resource	Verify release of a reserved resource.	Resource marked as free and user exit recorded.
5. Real-Time Map	Check update of resource map in real time.	Resource status updates in real time on the map.
6. Complaint Management	Validate review of complaints by the admin.	Admin correctly views and manages complaints.
7. Force Logout	Test forced logout by admin.	Resource is released, and status is updated.
8. Tablet Synchronization	Evaluate synchronization between entry and exit.	Correct synchronization between both tablets.
9. Real-Time Update on Panel	Verify real-time update of resource usage.	Admin sees resource usage in real time.
10. Authentication Failure	Test login failure with incorrect credentials.	System shows error for incorrect ID.
11. Performance Under Load	Evaluate performance with multiple users.	System responds efficiently without delays.
12. Loss of Connection	Test behavior when connection is lost.	Shows error and does not allow continuation until connection is restored.

TEST CASES AND EXPECTED RESULTS FOR THE WILD COMMONS SYSTEM

Abdiel:

- Role: Frontend Developer
- Responsibilities:
 - Design and implement the user interface of the mobile application.
 - Ensure the usability and accessibility of the system.
 - Collaborate with the team to integrate the frontend with the backend.
 - Conduct usability tests and gather user feedback.

Javier:

- Role: Backend Developer
- Responsibilities:
 - Develop and maintain PHP scripts for database interaction.
 - Implement the business logic of the system.
 - Ensure proper communication between the frontend and the database.
 - Perform unit and integration testing to verify backend functionality.

Gael:

- Role: User Experience (UX) Designer
- Responsibilities:
 - Research and analyze user needs to improve the application experience.
 - Design wireframes and prototypes of the user interface.
 - Collaborate with Abdiel and Jared to ensure the design is correctly implemented.
 - Conduct usability testing and gather user feedback for continuous improvements.

Rodrigo:

- Role: Quality Assurance Specialist
- Responsibilities:
 - Develop and execute test cases to verify the system.
 - Perform functionality, performance, and security testing.
 - Document test results and report issues.
 - Collaborate with the team to ensure identified problems are resolved.

Jared:

- Role: Database Administrator
- Responsibilities:
 - Design and manage the MySQL database.
 - Ensure data integrity and security.
 - Optimize queries and database performance.
 - Collaborate with the backend developer to integrate the database with the system.

TEAM COMMUNICATION FOR WILD COMMONS

Weekly Meetings:

Weekly meetings will be held to discuss project progress, review pending tasks, and plan next steps. These meetings will also serve to address questions and share ideas.

Instant Messaging:

A group will be established on an instant messaging platform (such as Slack, WhatsApp, or Microsoft Teams) for quick and effective communication. This will facilitate problem resolution and real-time collaboration.

Shared Documentation:

Online documentation tools (like Google Docs or Notion) will be used to maintain a record of decisions, meeting notes, and project progress. All members will have access to contribute and review the information.

Code Reviews:

Periodic code review sessions will be conducted to ensure the quality of work and encourage learning among team members.

Continuous Feedback:

A culture of continuous feedback will be encouraged, where each team member can express their opinions and suggestions on each other's work. This will help improve collaboration and the overall quality of the project.

ESTIMATED COST

INFRAESTRUCTURE | \$23,000 MXN

TABLETS | $2 * \$7,000 = \$14,000$

WALL MOUNTS & CHARGERS | Cables, chargers, and mounting systems for wall installation(per tablet) $2 * \$2,000 = \$4,000$

SERVER & NETWORKING | Basic networking infrastructure, configuration and computer upgrade. \$5,000

SOFTWARE & SERVICES | \$5,000 MXN

DEVELOPMENT TOOLS | Most softwares (java, Android Studio) are FREE

DATABASE | MySQL is Open-source and free

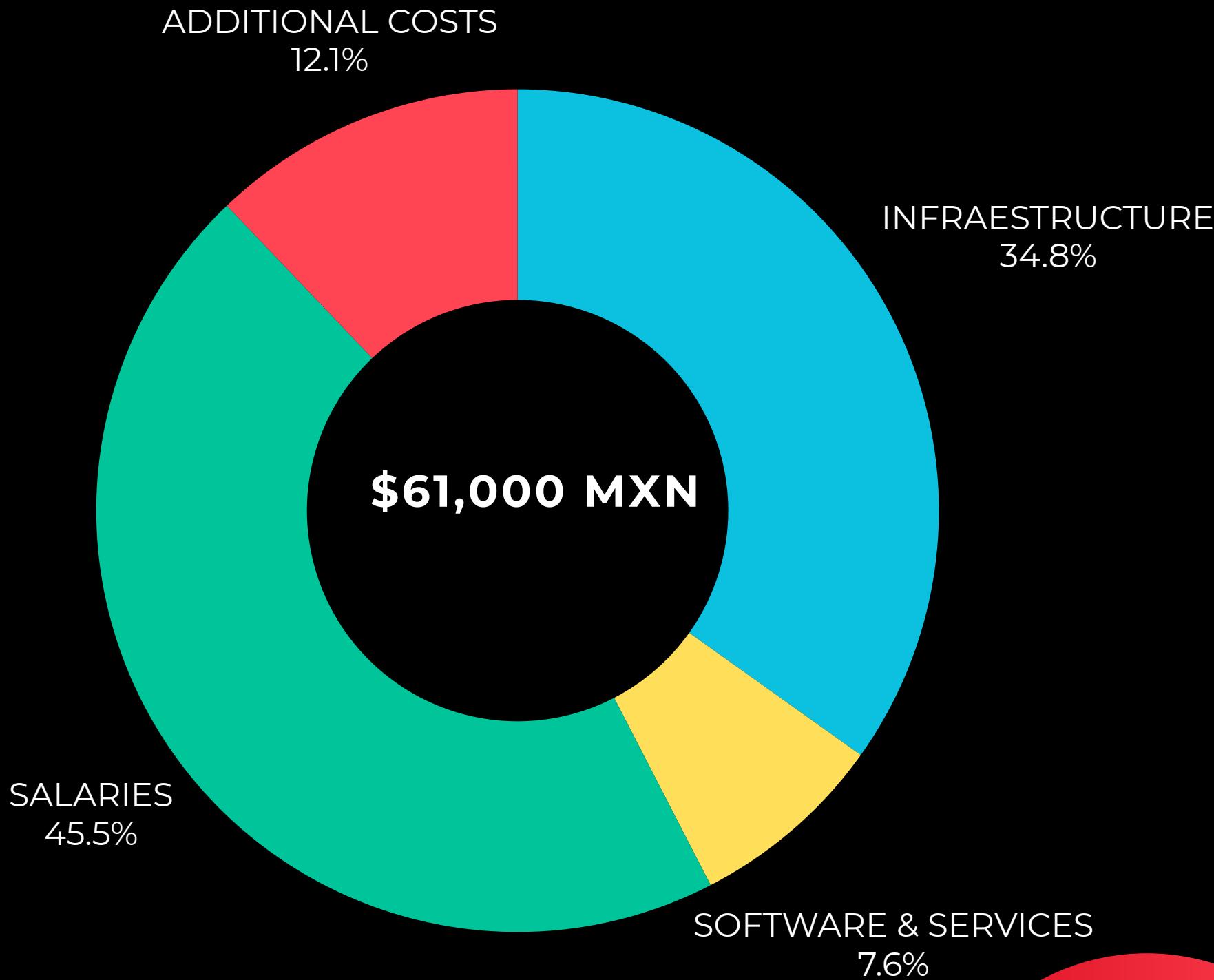
BACKUP - SECURITY | Cloud backup is in range of \$1,000 - \$5,000

SALARIES | \$30,000 MXN

PROGRAMMERS | $\$6,000 * 5 = \$30,000$

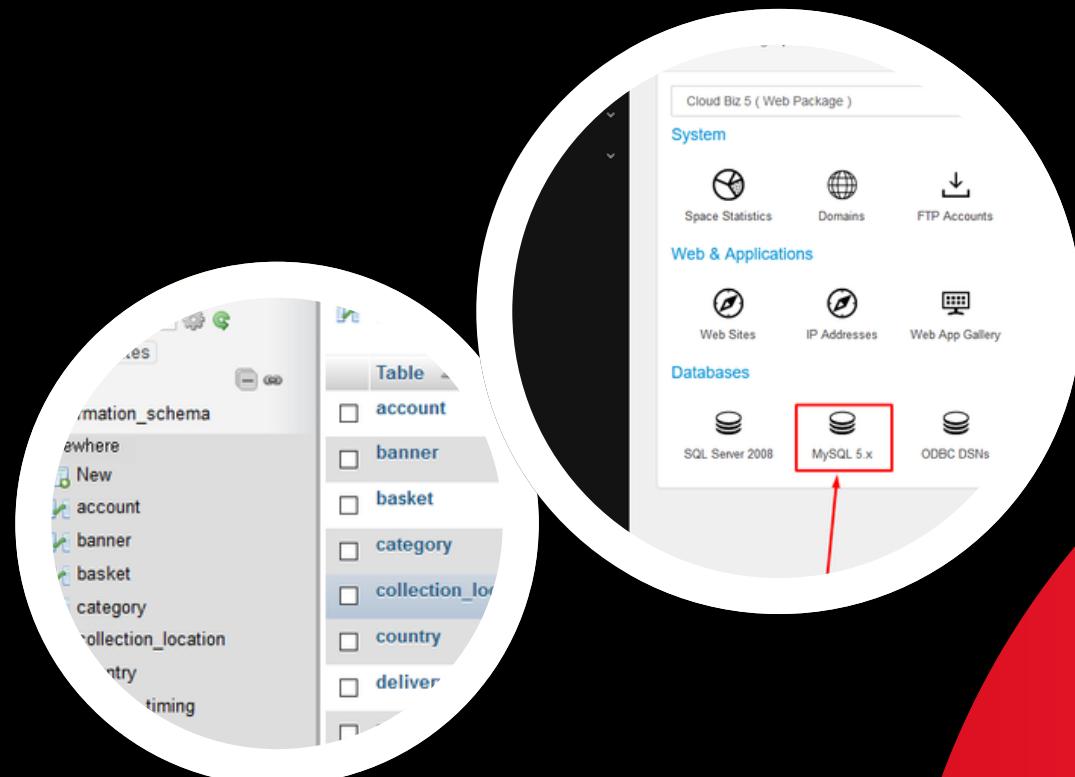
ADDITIONAL COSTS | \$8,000 MXN

MISCELLANEOUS EXPENSES | unexpected hardware upgrades, maintenance, software licenses, etc. \$8,000



KEY MILESTONES

- Complete design of the database.
- Fully functional reservation system
- Operational admin control panel.
-



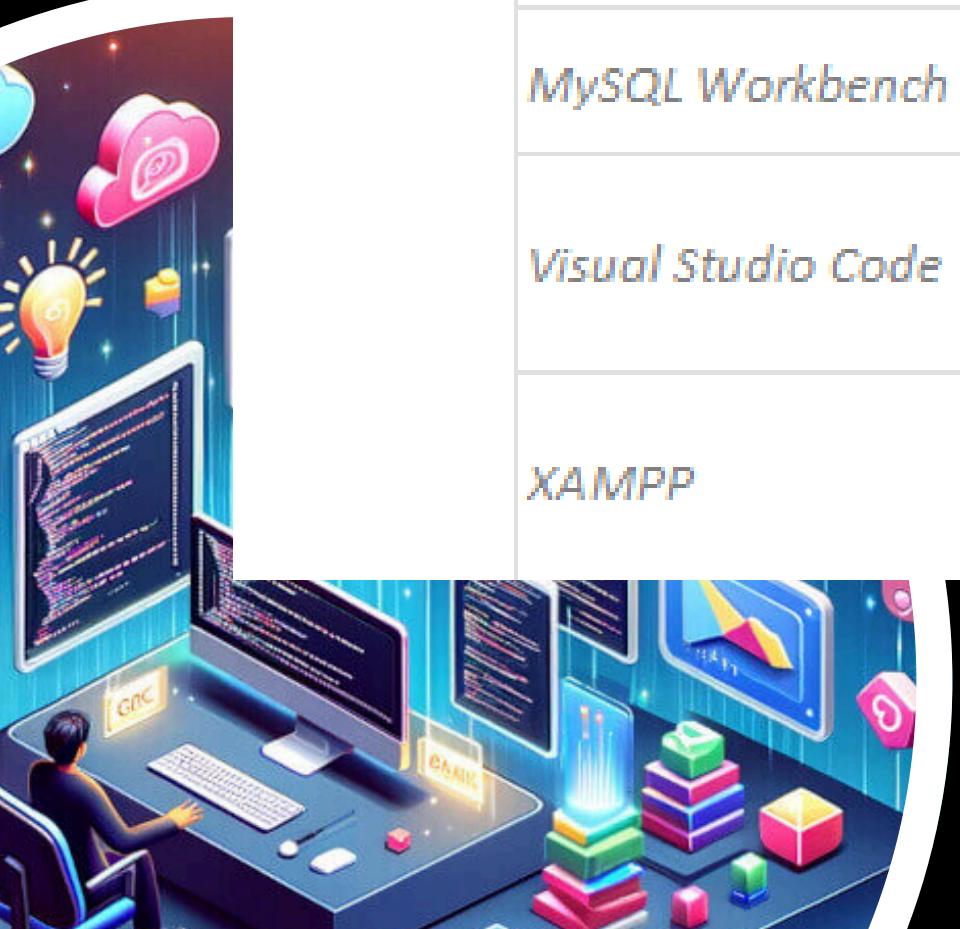
WILD COMMONS CO.



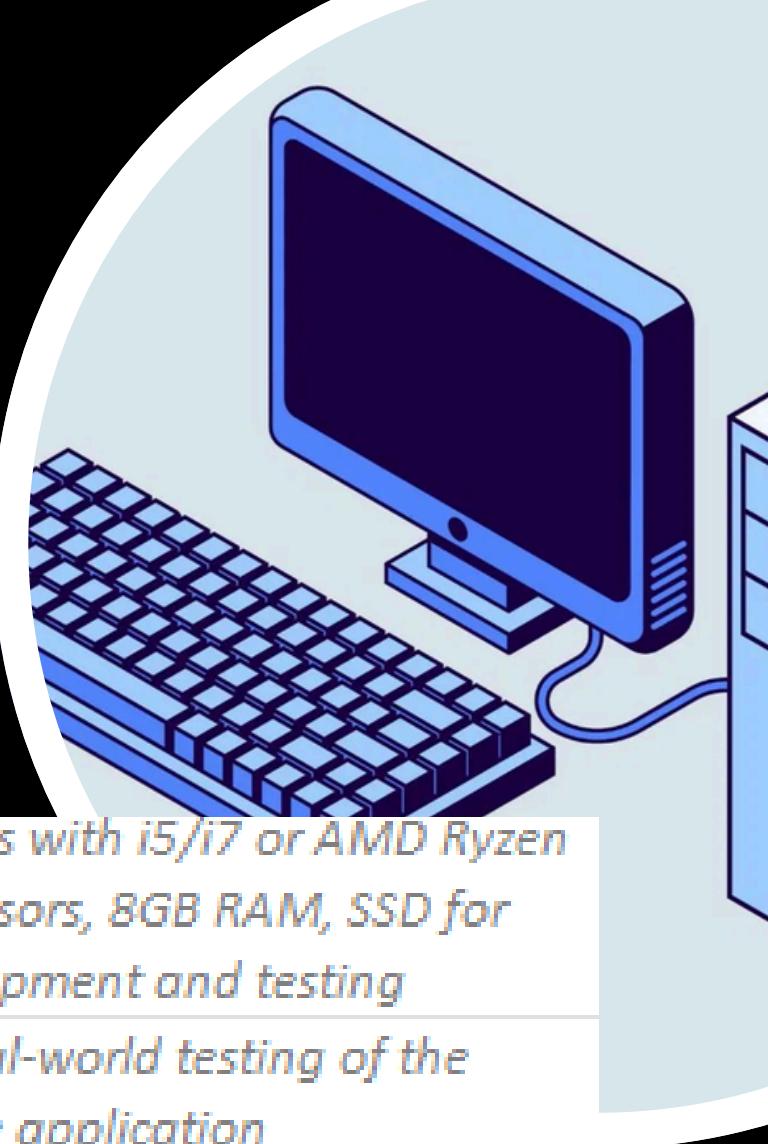
PROCESS

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.												



	<i>Android Studio</i>	<i>Integrated Development Environment (IDE) for Android application development</i>
	<i>Java Development Kit (JDK)</i>	<i>Tools to compile and run Java applications</i>
	<i>Android SDK</i>	<i>Software Development Kit for different Android versions</i>
	<i>Android Emulator</i>	<i>Virtual devices for testing the app on different screen resolutions</i>
	<i>MySQL</i>	<i>Relational database management system for backend</i>
	<i>MySQL Workbench</i>	<i>Visual tool for managing MySQL databases</i>
	<i>Visual Studio Code</i>	<i>Lightweight code editor with extensions for web development</i>
	<i>XAMPP</i>	<i>Free software that includes MySQL for database management</i>

<i>Hardware</i>	<i>Computers</i>	<i>Devices with i5/i7 or AMD Ryzen processors, 8GB RAM, SSD for development and testing</i>
	<i>Physical Android Devices</i>	<i>For real-world testing of the mobile application</i>





Wild Commons

THANK YOU