**Guide 3. APT Project Final Report**

**Capstone Subject**

|  |
| --- |
| **1. APT Project Final Report** |
| The objective of this report is for you to describe the most relevant aspects of your APT Project. It is important that you justify the decisions you had to make throughout the process.  Below, you will find different fields that you must complete with the requested information, which will give a summary of your APT project and its main results. |

|  |  |
| --- | --- |
| Student name | [Nibaldo Andrés Quezada Garay](mailto:nib.quezada@duocuc.cl) |
| ID | **19.116.353-2** |
| Career | **Computer Engineering** |
| headquarters | **Saint Bernard** |

|  |  |
| --- | --- |
| Student name | **Nicolás Gabriel Peña Muñoz** |
| ID | **18.795.757-5** |
| Career | **Computer Engineering** |
| headquarters | **Saint Bernard** |

|  |  |
| --- | --- |
| Student name | [Cristian Mauricio Arroyo Bossardt](mailto:cri.arroyo@duocuc.cl) |
| ID | **15.893.730-1** |
| Career | **Computer Engineering** |
| headquarters | **Saint Bernard** |

|  |  |
| --- | --- |
| **Contents of the final report** | |
| 1. Relevance of the APT project | Our client is in the Vista Parque 1 and 2 condominium, which is located in the commune of San Bernardo, at Almirante Riveros #630. The condominium has 570 apartments and 30 houses, with a total of 600 homes. The condominium has two major problems, which are described below.   1. Due to the large number of vehicles on the premises and the manual work done by the concierges on a daily basis, traffic jams are generated both at the entrance to the condominium and on the avenue on which it is located. 2. Due to the high crime rates, the administration needs a system that restricts vehicle access, ensuring exclusive access to residents and authorized personnel, with the intention of preventing theft and "portonazos", which have recently affected the condominium.   At TechApps we put the quality of life and work of people as our main focus, so faced with the problem posed by the condominium, we decided to propose a complete and efficient solution that solves their vehicular access management problem, since as a company we believe that a project of this magnitude could apply all the knowledge acquired during the training process at Duoc, from project management and product quality, to software development and implementation. |
| 2. Goals | Project Objectives  General Objective: Develop a comprehensive vehicle access control system that improves security and streamlines the entry and exit of vehicles in the Vista Parque condominium, ensuring exclusive control by residents and authorized personnel.  Specific Objectives:   1. Implement an automatic vehicle license plate reader using Python. 2. Develop a mobile application in Flutter so that residents can register visits in advance. 3. Create a REST API in Spring Boot for managing vehicle access permits and integration with the external API. 4. Design a database in SQL Server that supports the management of residents, visitors and commercial vehicles. 5. Generate vehicle access reports and real-time monitoring. 6. 6. Ensure that the system complies with computer security standards. |
| 3. Methodology | **Project Methodology**  The methodology followed is **Scrum**, with **9 sprints of two weeks each**. The activities include **requirement planning, modular development** (license plate reader, API, mobile application, database), **integration testing**, and the future **system deployment**.  **1. User Stories**  User stories are essential for defining system requirements and ensuring that the developed functionalities align with the needs of the end users. Below are the prioritized user stories for the mobile application, focused on improving vehicular access in the Vista Parque condominium.  **01. Resident Vehicle Registration**   * **Description**: As a resident, I want my vehicle to be registered and able to enter the condominium automatically to avoid congestion at the entrance during peak hours. * **Priority**: High * **Justification**: This functionality is critical for streamlining vehicle flow and reducing wait times at the entrance, enhancing the daily experience of residents.   **02. Visitor Vehicle Registration**   * **Description**: As a resident, I want to be able to register visits in advance so that they can enter without delays, at any time. * **Priority**: High * **Justification**: Facilitating the access of pre-authorized visitors improves security and efficiency, allowing residents to manage their visits autonomously.   **03. Visit History**   * **Description**: As a resident, I want to view the history of my entries and exits, as well as those of my visitors, to monitor vehicle movements and avoid fines. * **Priority**: Medium * **Justification**: This feature enables residents to have better control over vehicle and visitor access, contributing to transparency and the prevention of infractions.   **2. Product Backlog**  The Product Backlog defines the key functionalities and modules that the system must include to optimize vehicular access control and security in the Vista Parque condominium. Below are the prioritized functionalities for the development team.  **01. Real-Time Monitoring**   * **Description**: As security personnel (Gatekeeper), I want to monitor vehicle flow in real-time at the entrance to speed up access and handle incidents in emergency situations. * **Priority**: High * **Justification**: This functionality is essential for security personnel to have a clear view of vehicular access and to make quick decisions in emergencies.   **02. Fine Management**   * **Description**: As security personnel (Gatekeeper), I want the system to automatically manage and record fines for visits that exceed the authorized time. * **Priority**: Medium * **Justification**: Automating fine management helps reduce administrative workload and ensures compliance with access rules.   **03. Commercial Vehicle Inquiry**   * **Description**: As security personnel (Gatekeeper), I want to quickly check the access of commercial vehicles (Uber, deliveries, etc.) to grant them fast access to the condominium. * **Priority**: Medium * **Justification**: Optimizes service vehicle access, enhancing efficiency without compromising condominium security.   **04. Emergency Access Control**   * **Description**: As security personnel (Gatekeeper), I want full control of access during emergencies to ensure the safety of residents and the condominium. * **Priority**: High * **Justification**: This functionality is critical to ensure controlled access during emergencies, preventing unauthorized persons from entering during crises.   **Development Focus**: The development of the functionalities described in the user stories and backlog is centered on enhancing vehicular access efficiency, ensuring resident safety, and facilitating the work of security personnel. By prioritizing the needs of both residents and administrative staff, we aim to create a **comprehensive and scalable system** that addresses the challenges of the Vista Parque condominium.  **Project Plan Overview**  The **TechApps SCAV** project has been planned to be developed in a total of **9 sprints**, each lasting **2 weeks**. Below is a detailed breakdown of the activities to be carried out in each phase of the project, as well as the specific objectives of each sprint.  **Sprint Breakdown**   1. **Sprint 1 & 2: Project Management**    * **Activities**: Planning sprints, creating the backlog, and defining priorities.    * **Duration**: Weeks 1 to 4 2. **Sprint 3: License Plate Recognition System Development**    * **Activities**: Implementing the license plate recognition system using Python and OpenCV to capture and validate vehicle plates.    * **Duration**: Weeks 5 and 6 3. **Sprint 4: Database Analysis and Design**    * **Activities**: Designing the database structure and relational model using Google Cloud SQL.    * **Duration**: Weeks 7 and 8 4. **Sprint 5: REST API Development with Spring Boot**    * **Activities**: Implementing the REST API using Spring Boot to manage communication between the frontend and backend, deployed using Google Kubernetes Engine (GKE).    * **Duration**: Weeks 9 and 10 5. **Sprint 6 & 7: Mobile Application Development (Flutter)**    * **Activities**: Creating a mobile application in Flutter that connects to the API, allowing residents to register visits, manage vehicles, and receive notifications.    * **Duration**: Weeks 11 to 14 6. **Sprint 8: Software Quality Testing**    * **Activities**: Conducting functional, integration, and security tests to ensure the system is robust and reliable.    * **Duration**: Weeks 15 and 16 7. **Sprint 9: Implementation, Training, and Soft Launch**    * **Activities**: Conducting a soft launch, training end-users (security personnel and administrators), and final deployment to production.    * **Duration**: Weeks 17 and 18   **Work Plan**  The schedule structured into **9 sprints**, each lasting **2 weeks**, is designed to ensure that the **TechApps SCAV** project meets the set deadlines and is delivered with the expected quality. The integration of **cloud technologies** and the use of **agile methodologies** have been key to optimizing both the development and implementation of the system.  This approach will improve the **security and efficiency** of vehicular access control at the Vista Parque condominium, benefiting both the residents and the security personnel. |
| 4. Development | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Activity | Status | Difficulties | Materials/Physical Resources | Languages/Environment | | Patent Reader | Completed | Difficulties in doing live testing with the camera on the street due to weather conditions and traffic | Video camera, PC | Python, OpenCV | | Database Design | Completed | N/A | PC, SQL Server SQL | PostgreSQL | | API REST | In progress | N/A | PC, GCP (Google Cloud Platform) | Java, Spring Boot | | Mobile Application in Flutter | In progress | N/A | PC, smartphone for Flutter, Dart testing | Flutter, Dart | | Integration Testing | Not started | Coordination for real-time camera testing in a controlled | environment Video camera, PC, local network for integration | Python, Java, SQL, Flutter |   Future Activities   * Perform integration tests between modules. * Prepare the system for launch and training of end users.   Factors that have facilitated the development of the work plan:   * Team coordination: The implementation of daily meetings using the Scrum methodology has allowed for better communication and quick problem resolution. * Adequate technological tools: We have a well-configured development environment, including Python, SQL Server, and Spring Boot, which has facilitated the work of the developers. * Early functional prototype: The patent reader was completed in the early stages of the project, which allowed for rapid progress in the integration with other modules. Factors that have hindered the development of the work plan:  **Change in Application Structure**  * **Identified Difficulty**: As mentioned in previous changes, the plan was to develop two separate applications: one for **administration** and another for **residents**. However, this separation created redundancy, complicated management, and increased the maintenance workload. * **Action Taken**: Instead of continuing with two separate applications, the decision was made to **unify** all functionalities into a **single mobile application**. * The new application is **dynamic** and adapts its functionalities based on the **user role** (Administrator or Resident) upon login.   .   1. Error in the application of the Scrum methodology:    * Difficulty: Although it was planned to work with Scrum, in the first few weeks a traditional approach was mistakenly adopted, affecting planning and deliverables.    * Action taken: An immediate correction was made to resume the Scrum methodology, establishing 2-week sprints with daily meetings and incremental deliverables. 2. Adjustment in the sprint schedule:    * Difficulty: The initial planning of the sprints did not allocate enough time for some key modules, such as the application in Flutter.    * Action taken: Tasks were redistributed into 9 sprints, giving 2 full sprints for development in Flutter and adjusting the dates of the rest of the modules.  Adjusted or deleted activities Adjustments made:   1. Unification of applications into two versions (Administrative and Resident):    * Reason: Initially, it was planned to develop two separate applications: one for **administration** and another for **residents**. However, during development, it was identified that this separation created redundancies and increased system complexity.    * Adjustment: The decision was made to **return to the original idea** of developing a **single unified application** that integrates all functionalities, adapting its interface and options based on the **user role** (Administrator or Resident) upon login.    * Impact: This unification simplifies **maintenance and updates**, enhances the user experience, and reduces the workload for the development team. Now, all efforts are focused on optimizing the unified application, which ensures more efficient management and scalability of the system. 2. Adjustment in the sprint schedule:    * Reason: The initial planning allocated one sprint per module, which did not provide adequate time for the development of some more complex components, such as the mobile app in Flutter.    * Adjustment: A redistribution of the schedule was made, increasing the time for the development of the application in Flutter to two sprints and maintaining one sprint for the simplest modules. This new structure ensures better planning and avoids delays.   Deleted activities: At the moment, no activities have been deleted, as all the initially defined phases are still relevant to achieve the project objectives. |
| 5. Evidence | Patent Reader Module  **Vehicle Access Control System Overview**  Below, we describe the functionality of the **vehicle access control system** developed to optimize security in private areas such as condominiums, offices, or parking lots. The system allows **automated access** for authorized vehicles by detecting license plates and logs both entry and exit times in a logbook via an API. Additionally, it integrates a **servo motor controlled by an Arduino** to operate a barrier that automatically raises upon detecting a valid license plate.  **General System Description**  The system consists of several interacting components:   1. **Camera**: Captures real-time images of the vehicle as it approaches the control point. 2. **License Plate Reader (Python)**: Detects the vehicle’s license plate from the video stream and performs API queries to verify if the vehicle is registered. 3. **API (Backend)**: Manages information about authorized vehicles and logs access events in a database. 4. **Barrier Control (Arduino)**: Uses a servo motor to control the physical barrier, which is activated by commands sent from the Python system.   **System Functionality**  **1. Vehicle Entry Flow**  The entry process follows these steps:   1. **Image Capture**:    * The camera continuously captures real-time images. 2. **License Plate Detection**:    * Using the **OpenCV library in Python**, the presence of a license plate is detected in the video stream.    * The license plate number is extracted using **Optical Character Recognition (OCR)** techniques. 3. **API Query**:    * The detected license plate number is sent to a specific endpoint (/api/v2/vehicle/license/{license}) to verify if the vehicle is registered.    * The API returns information about the vehicle, including its status (resident, visitor, etc.). 4. **Logbook Entry**:    * If the vehicle is registered, a new entry is created in the logbook using a **POST request** to the API (/api/v2/logbook).    * The logbook records the entry time and leaves the exit time field as **null** until the vehicle leaves. 5. **Barrier Activation**:    * If the license plate is valid, a command is sent to the Arduino via serial communication to raise the barrier using the servo motor.   **2. Vehicle Exit Flow**  The exit process follows a similar logic with a few differences:   1. **Capture and Detection**:    * The camera detects the license plate when the vehicle attempts to exit. 2. **Verification and Logbook Update**:    * The license plate number is queried in the API to obtain the corresponding entry record ID.    * A **PUT request** is sent to the API (/api/v2/logbook/exit/{vehicleId}) to update the exit time field. 3. **Duration Calculation**:    * Once the exit time is recorded, the total time the vehicle spent in the area is calculated. 4. **Barrier Opening for Exit**:    * Upon confirming that the logbook has been updated successfully, a command is sent to the Arduino to open the barrier and allow the vehicle to exit.   **Integration with Arduino for Barrier Control**  The system uses an **Arduino** connected to a **servo motor** to control the barrier. The Arduino communicates with the Python system via a **serial port (COM6)**. The integration is implemented as follows:   1. **Initialization**:    * The Arduino is set up to listen for commands from the serial port. 2. **Open Command**:    * When a valid license plate is detected, a **"raise" command** is sent to the Arduino, which activates the servo motor to lift the barrier. 3. **Automatic Closing**:    * After a brief interval (5 seconds), the Arduino automatically lowers the barrier.   This system enhances **security and efficiency** by automating vehicle access while maintaining detailed records of entries and exits, making it ideal for managing high-traffic environments like condominiums and office complexes.  REST API Module    Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto  Descripción generada automáticamenteInterfaz de usuario gráfica, Texto  Descripción generada automáticamenteInterfaz de usuario gráfica  Descripción generada automáticamente con confianza mediaMódulo Flutter  Interfaz de usuario gráfica  Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación  Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Correo electrónico  Descripción generada automáticamente  Interfaz de usuario gráfica, Texto, Aplicación  Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación  Descripción generada automáticamente  Interfaz de usuario gráfica, Aplicación  Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación  Descripción generada automáticamente  Interfaz de usuario gráfica, Aplicación  Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación, Word  Descripción generada automáticamente  Interfaz de usuario gráfica, Aplicación  Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto  Descripción generada automáticamente  Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto  Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación  Descripción generada automáticamente  Interfaz de usuario gráfica, Texto, Aplicación  Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación  Descripción generada automáticamente  Interfaz de usuario gráfica, Aplicación  Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto  Descripción generada automáticamente  Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto  Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación  Descripción generada automáticamente  MER    Architecture diagram |
| 6. Professional interests and projections | **1. Implementation of Artificial Intelligence (AI)**   * **Description**: The plan is to incorporate **artificial intelligence algorithms** to enhance the accuracy of **license plate detection** and predict vehicle access patterns. This will further optimize security and traffic flow within the condominium. * **Expected Impact**:   + **Improved Detection**: Increase the accuracy of license plate recognition, reducing errors in identifying unauthorized vehicles.   + **Predictive Analysis**: Utilize AI to **predict traffic peaks** and dynamically adjust barrier opening times, enhancing efficiency during peak hours.   + **Intelligent Automation**: Apply AI to **automatically manage access** based on the history and behavior of registered vehicles.   **2. Payment System for Fines**   * **Description**: A **payment module** is planned to allow residents to pay fines quickly and securely through the mobile app. * **System Features**:   + **Integration with Payment Platforms**: Enable the use of **credit/debit cards** and other electronic payment methods.   + **Automatic Notifications**: Send automatic reminders to residents regarding **outstanding fines** and payment deadlines.   + **Payment History**: Users will be able to view a **payment history** and download receipts directly from the app.   **3. Parking Management Module**   * **Description**: Implement a module for **parking management**, allowing residents and visitors to **reserve and manage parking spaces** within the condominium. * **Key Features**:   + **Real-Time Reservations**: Allow residents to **reserve parking spots** in advance through the app, preventing space conflicts during peak hours.   + **Monitoring of Available Spaces**: Use sensors and real-time data to **monitor the availability** of parking spaces.   + **Visitor Space Allocation**: Manage the allocation of specific spaces for visitors, improving control and security within the condominium. |