**Guide3. 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 substantiate the decisions you had to make throughout the process.  Below, you will find different fields that you must fill in with the requested information, which account for the summary of your APT project and its main results. |

|  |  |
| --- | --- |
| Student Name | [Nibaldo Andrés Quezada Garay](mailto:nib.quezada@duocuc.cl) |
| Ruth | **19.116.353-2** |
| Race | **Computer Engineering** |
| Headquarters | **Saint Bernard** |

|  |  |
| --- | --- |
| Student Name | **Nicolás Gabriel Peña Muñoz** |
| Ruth | **18.795.757-5** |
| Race | **Computer Engineering** |
| Headquarters | **Saint Bernard** |

|  |  |
| --- | --- |
| Student Name | [Cristian Mauricio Arroyo Bossardt](mailto:cri.arroyo@duocuc.cl) |
| Ruth | **15.893.730-1** |
| Race | **Computer Engineering** |
| Headquarters | **Saint Bernard** |

|  |  |
| --- | --- |
| **Contents of the final report** | |
| 1. Relevance of the APT project | Our client corresponds to the Vista Parque 1 and 2 condominium, which is located in the San Bernardo commune, at Almirante Riveros #630. The condominium has 570 apartments and 30 houses, having a total of 600 homes. The condominium presents two major problems, which are described below.   1. Due to the large number of vehicles on the premises and the manual daily operation on the part of the janitors, traffic jams are generated both at the entrance to the condominium, and on the avenue on which it is located. 2. As a result of the high crime rates, the administration needs a system that restricts vehicular access, ensuring exclusive access to residents and authorized personnel, with the intention of preventing robberies and "slamming doors", which has affected the condominium lately.   At TechApps we put as our main focus the quality of life and work of people, so in the face of the problems 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 in Duoc, From project management and product quality, to software development and implementation. |
| 2. Objectives | Project Objectives  General Objective: To develop a comprehensive vehicular access control system that improves security and speeds up the entry and exit of vehicles in the Vista Parque condominium, ensuring the exclusive control of residents and authorized personnel.  Specific Objectives:   1. Implement an automatic vehicle license plate reader using Python. 2. Develop a mobile app in Flutter so residents can register visits in advance. 3. Create a REST API in Spring Boot for vehicular access permission management and integration with the external API. 4. Design a PostgreSQL database that supports resident, visitation, and commercial vehicle management. 5. Generate reports of vehicular access and monitoring in real time. 6. Ensure that the system complies with computer security standards. |
| 3. Methodology | The methodology followed is Scrum, with 9 two-week sprints. Activities include requirements planning, modular development (patent reader, API, mobile app, database), integration testing, and future system commissioning.  1. User Stories  User stories are essential to define system requirements and ensure that the functionalities developed align with the needs of 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 automatically enter the condo to avoid congestion at the driveway during peak hours. * Priority: High * Rationale: This functionality is critical to streamline vehicular flow and reduce wait times at the entrance, improving the daily experience of residents.   02. Registration of Visitor Vehicles   * Description: As a resident, I want to be able to register visitors in advance so that they can enter without delay, at any time. * Priority: High * Rationale: Facilitating access for previously authorized visitors improves safety and efficiency, allowing residents to manage their visits autonomously.   03. Visit History   * Description: As a resident, I want to be able to check the history of my entries and exits, as well as visits to the condo, to control my vehicular movements and avoid fines. * Priority: Medium * Rationale: This feature allows residents to have greater control over the access of their vehicles and visitors, contributing to transparency and the prevention of violations.   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. The following describes the prioritized capabilities for the development team.  01. Real-Time Monitoring   * Description: As a security personnel (Porter's Office), I want to monitor in real time the vehicular flow at the entrance to speed up access and attend to incidents in emergency situations. * Priority: High * Rationale: This functionality is essential for security personnel to have a clear view of vehicular access and to be able to make quick decisions in case of emergencies.   02. Fines Manager   * Description: As a security (Porter) staff, I want the system to automatically manage and record fines for visits that exceed the authorized time. * Priority: Medium * Rationale: Automating fine control helps reduce administrative burden and ensures that access rules are complied with.   03. Vehicle Inquiry   * Description: As a security personnel (Porter's Office), I want to be able to quickly consult the access of commercial vehicles (Uber, mail, orders) to give them quick access to the condominium. * Priority: Medium * Justification: Optimizes access for service vehicles, improving efficiency without compromising the security of the condominium.   04. Emergency Access Control   * Description: As security personnel (Porter), I want to have full control of access in emergency situations to ensure the safety of the residents and the condo. * Priority: High * Rationale: This functionality is critical to ensure controlled access during emergencies, preventing unauthorized persons from entering in times of crisis.   The development of the functionalities described in the user stories and the backlog has a clear focus on improving the efficiency of vehicular access, ensuring the safety of residents and facilitating the work of security personnel. By prioritizing the needs of both residents and administrative staff, it seeks to create a comprehensive and scalable system that responds to the challenges of the Vista Parque condominium.  The TechApps SCAV project has been planned to be developed in a total of 9 sprints, with a duration of 2 weeks each. The detailed breakdown of the activities to be carried out in each phase of the project is described below, as well as the specific objectives of each sprint.  **Summary of the Work Plan**  The TechApps SCAV project has been planned to be developed in a total of 9 sprints, with a duration of 2 weeks each. The detailed breakdown of the activities to be carried out in each phase of the project is described below, as well as the specific objectives of each sprint.  **1. Sprint 1 & 2 Project Management**   * Activities: Planning the sprints, creating the backlog and defining priorities. * Duration: Weeks 1 to 4   **2. Sprint 3 Development of the Patent Recognition System**   * Activities: Implement the license plate recognition system using Python and OpenCV to capture and validate vehicle licenses. * Duration: Week 5 & 6   **3. Sprint 4 Database Analysis and Evaluation**   * Activities: Design the structure of the database and its relational model using Neon Tech SQL. * Duration: Week 7 and 8   **4. Sprint 5 REST API Development in Spring Boot**   * Activities: Implement the REST API in Spring Boot to manage communication between the frontend and the backend, using Google Kubernetes Engine (GKE) for deployment. * Duration: Week 9 & 10   **5. Sprint 6 & 7 Mobile App Development (Flutter)**   * Activities: Create a mobile app in Flutter that connects to the API so residents can log visits, manage vehicles, and receive notifications. * Duration: Week 11 to 14   **6. Sprint 8 Software Quality Testing**   * Activities: Perform functional, integration, and security tests to ensure the system is robust and reliable. * Duration: Week 15 & 16   **7. Sprint 9 Implementation, Training, and White March**   * Activities: Carry out the white march, training end users (security personnel and administrators) and final deployment in production. * Duration: Week 17 & 18   **Work Plan**  The schedule structured in 9 sprints, lasting 2 weeks each, is designed to ensure that the TechApps SCAV project meets the established 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 in the Vista Parque condominium, benefiting both residents and security personnel. |
| 4. Development | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Activity | State | Difficulties | Physical Materials/Resources | Languages/Environment | | Patent Reader | Completed | Difficulties in doing live tests with the camera on the street due to weather conditions and traffic | Video Camera, PC | Python, OpenCV | | Database Design | Completed | N/A | |  | | --- | | PC, Neon Tech PostgreSQL |  |  | | --- | |  | | PostgreSQL | | API REST | Current | N/A | PC, GCP (Google Cloud Platform) | Java, Spring Boot | | Mobile App in Flutter | Current | N/A | PC, Smartphone for 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 commissioning and training 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 solving. * Suitable technological tools: We have a well-configured development environment, including Python, postgreSQL and Spring Boot, which has made the developers' work easier. * Early functional prototype: The patent reader was completed in the early stages of the project, allowing rapid progress in integration with other modules.   Factors that have hindered the development of the work plan:   1. **Change in Application Structure**   **Identified Difficulty**: Mentioned as a previous change, it was planned to develop two independent applications: one for **administration** and one for **residents**. However, this separation created redundancy, complicated management and increased the maintenance burden.  **Action Taken**: Instead of continuing with two separate apps, the decision was made to **unify** all functionalities into **a single mobile app**.  The new application is **dynamic** and adapts its functionalities according to the **role of the user** who logs in (Administrator or Resident).   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 in two versions (Administrative and Resident):    * Reason: Originally, it was planned to develop two separate applications: one for **administration** and one for **residents**. However, during development, it was identified that this separation generated redundancies and increased the complexity of the system.    * Adjustment: It was decided to return to the original idea of developing a single unified application that integrates all the functionalities, adapting its interface and options according to the role of the user (Administrator or Resident) who logs in.    * Impact: This unification simplifies maintenance and upgrades, improves the user experience, and reduces the workload of 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  Below, we will describe the operation of the vehicular access control system developed to optimize security in private premises, such as condominiums, offices or parking lots. The system allows automated access of authorized vehicles through the detection of license plates (vehicle plates), and records both entry and exit in a log through an API. In addition, an Arduino-controlled servo motor is integrated to operate a barrier that automatically rises upon detecting a valid patent.  **System Overview**  The system consists of several components that interact with each other:   1. **Camera**: Captures real-time images of the vehicle as it approaches the checkpoint. 2. **License Plate Reader (Python):** Detects the vehicle's license plate from the video stream and queries an API to verify if the vehicle is registered. 3. **API (Backend):** It is responsible for handling information about authorized vehicles and registering accesses in a database. 4. **Barrier Control (Arduino):** Uses a servo motor to control the opening and closing of a physical barrier, triggered by commands sent from the Python system.   **How the System Works**  **1. Vehicle Input Flow**  The process of entering the venue follows these steps:   1. **Image Capture**: The camera continuously captures images in real time. 2. **Patent Detection**:    * Using the OpenCV library in Python, the presence of a patent is detected in the video stream.    * The patent 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/patent/{patent}) to verify if the vehicle is registered.    * The API returns information about the vehicle, including its status (resident, visiting, etc.). 4. **Log Log**:    * If the vehicle is registered, a new log entry record is created via a POST request to the API (/api/v2/log).    * The log stores the clock-in time and leaves the check-out field as null until the vehicle leaves. 5. **Barrier Activation**:    * If the patent is valid, a command is sent to the Arduino via serial communication for the servo motor to raise the barrier.   **2. Vehicle Outflow**  The output process follows a similar logic, with a few differences:   1. **Capture and Detection**:    * The camera detects the vehicle's license plate when it tries to leave the premises. 2. **Verification and Update in the Logbook**:    * The patent number is queried in the API to get the corresponding entry record ID.    * A PUT request is sent to the API (/api/v2/log/exit/{vehicleId}) to update the dateout field with the departure time. 3. **Calculation of the Dwell Time**:    * Once the departure time has been recorded, the total duration of the vehicle's stay on the premises is calculated. 4. **Opening of the Exit Barrier**:    * Upon confirming that the register has been successfully updated, a command is sent to the Arduino to open the barrier and allow the vehicle to exit.   **Connecting with the Arduino for Barrier Control**  The system uses an Arduino connected to a servo motor that controls the barrier. The Arduino communicates with the Python system using a serial port (COM6, in this case). The integration is carried out as follows:   1. **Initialization**:    * The Arduino is configured to listen for commands from the serial port. 2. **Aperture Command**:    * When a valid license plate is detected, the command is sent to climb the Arduino, which activates the servo motor to raise the barrier. 3. **Automatic Closing**:    * After a short interval (5 seconds), the Arduino automatically lowers the barrier.   REST API Module    Graphical User Interface, Text, App, Chat, or Text Message  Auto-generated descriptionGraphical User Interface, Text  Auto-generated descriptionGraphical user interface  Description automatically generated with medium confidenceFlutter Module  Graphical user interface  Auto-generated descriptionGraphical User Interface, Application  Auto-generated descriptionGraphical User Interface, Text, Application, Email  Auto-generated description  Graphical User Interface, Text, Application  Auto-generated descriptionGraphical User Interface, Application  Auto-generated description  Graphical User Interface, Application  Auto-generated descriptionGraphical User Interface, Application  Auto-generated description  Graphical User Interface, Application  Auto-generated descriptionGraphical User Interface, Application, Word  Auto-generated description  Graphical User Interface, Application  Auto-generated descriptionGraphical User Interface, Text, App, Chat, or Text Message  Auto-generated description  Graphical User Interface, Text, App, Chat, or Text Message  Auto-generated descriptionGraphical User Interface, Application  Auto-generated description  Graphical User Interface, Text, Application  Auto-generated descriptionGraphical User Interface, Application  Auto-generated description  Graphical User Interface, Application  Auto-generated descriptionGraphical User Interface, Text, App, Chat, or Text Message  Auto-generated description  Graphical User Interface, Text, App, Chat, or Text Message  Auto-generated descriptionGraphical User Interface, Application  Auto-generated description  MER    Architecture diagram |
| 6. Professional interests and projections | **1. Payment System for Fines**   * **Description: A** payment module **is planned to be developed**  that allows residents to **pay fines** quickly and securely via the mobile app. * **System Features**:   + **Integration with Payment Platforms**: Allow the use of credit/debit cards and other electronic payment methods.   + **Push Notifications**: Sending **automatic reminders** to residents about outstanding fines and payment deadlines.   + **Payment History**: Users will be able to view a **history of payments** and receipt downloads from the app.   **2. 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 Bookings**: Allow residents to **book parking from** the app in advance, avoiding space conflicts during peak hours.   + **Available Space Monitoring**: Use sensors and real-time data to **monitor the availability** of parking spaces.   + **Allocation of Spaces for Visitors**: Manage the allocation of **specific spaces for visitors**, improving the control and security of the condominium. |