**CHAPTER III: METHODOLOGY**

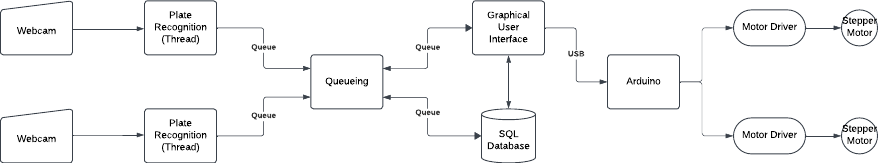
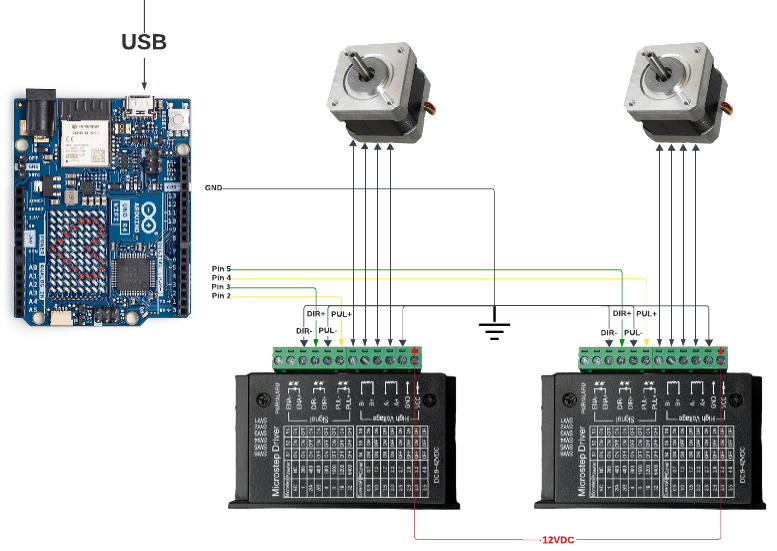
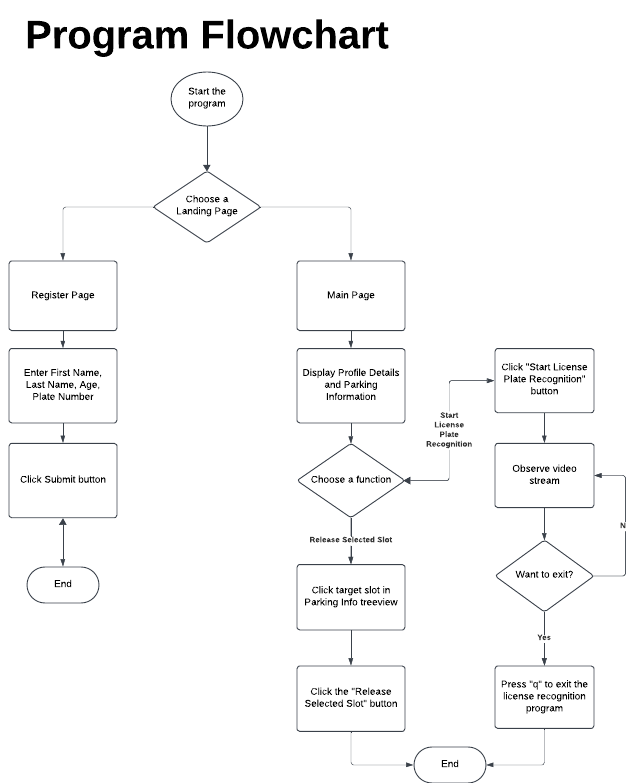
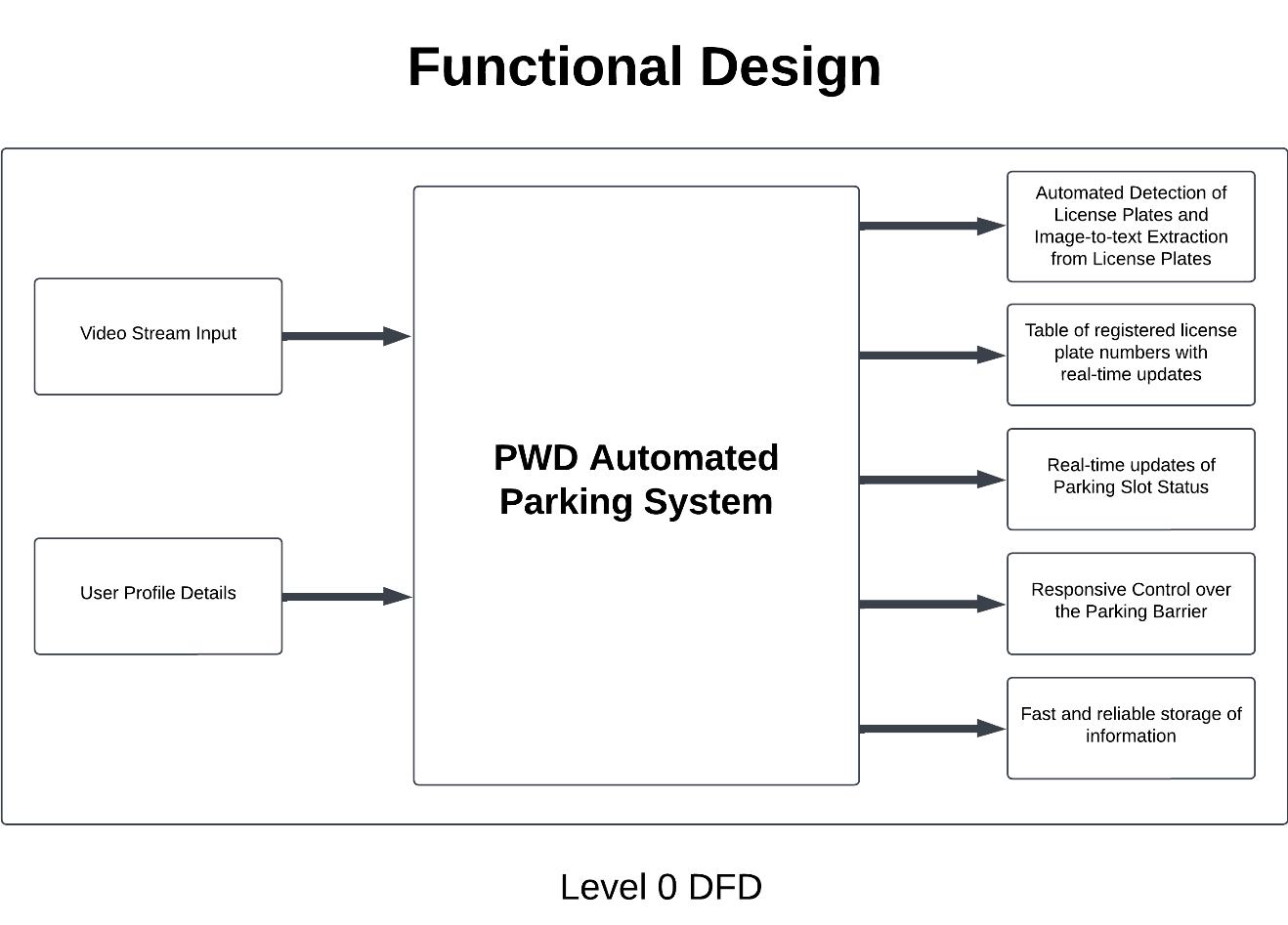
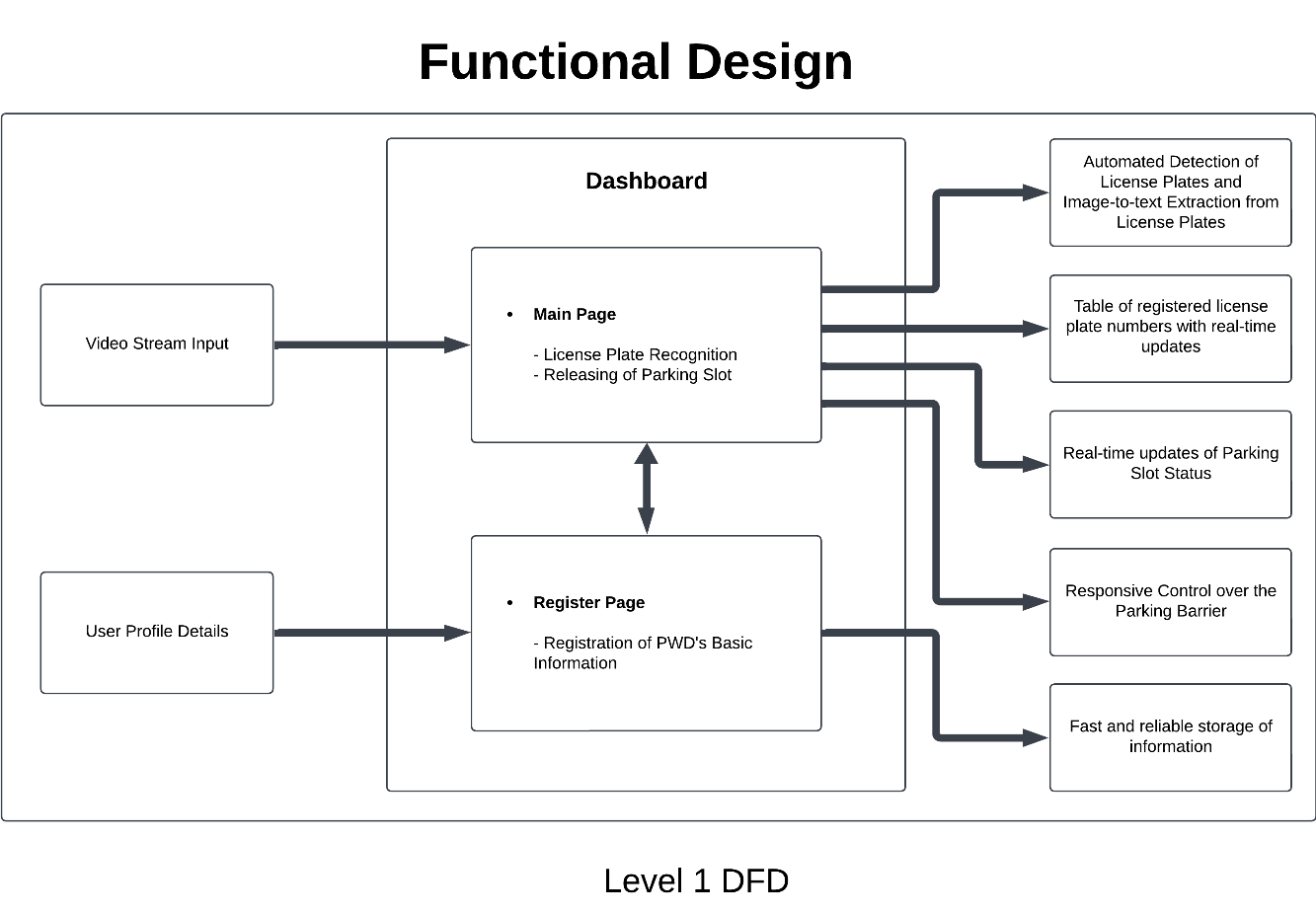


Figure # illustrates the end‐to‐end architecture of the automated license‐plate recognition and enforcement system. Two network‐attached webcams continuously capture live video streams, each of which is processed by an independent Python thread running our custom plate‐recognition algorithm. Detected plate data are immediately enqueued via Python’s built‐in queue module, enabling safe, asynchronous communication between the recognition threads, the graphical user interface (GUI), and the central database. As soon as new information arrives in the queue, the GUI component retrieves and displays the plate number, while simultaneously committing this record to an embedded SQLite database for persistent storage and later audit. In parallel, a serial signal will be dispatched over a USB connection to an Arduino microcontroller, which in turn activates the appropriate motor‐driver circuits to position two independent stepper motors. This hardware interface ensures real‐time actuation, such as raising a barrier arm or rotating a camera mount, based on recognition events. By leveraging Python’s concurrency primitives, a lightweight SQL engine, and a modular hardware interface, the system achieves reliable, low‐latency performance suitable for an automated parking system.









Test Cases:  
Testing of Prototype

Items to test:  
1. License Plate Detection

- Accuracy of License Plate Detection

- Inference of the Detection System

2. Optical Character Recognition

- Accuracy of extracted text with respect to the actual value

- Persistence of the extraction

3. Stepper Motor Control

- Performance of the Serial Communication

- Responsiveness of the motors

4. Profile Registration

- Ensure connection with the database

- Real-time update on the dashboard

Testing Methodology:  
- Conduct unit testing for hardware components such as the microcontroller, motor drivers, and stepper motors.

- Test communication protocols to ensure reliable transfer of data between devices.

- Perform integration testing to validate interactions between hardware and software.

- Execute registration testing in the Registration Panel to ensure a reliable connection with the database.

Unit Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Action | Expected Result | Actual Result | Date of Testing |
| Object-Detection-1 | Detection of vehicle license plates. | The system detected all visible license plates with high accuracy. | Visible license plates are detected successfully. |  |
| Optical-Character-Recognition-1 | Image-to-text extraction from vehicle license plates. | The system should successfully extract text from the image with high precision. | The system extracted the text as long as it was legible and had good lighting conditions. |  |
| Motor-Control-1 | Control of Stepper Motors using Serial Signals. | The stepper motors should respond quickly as soon as they receive a serial signal from the main computer. | The stepper motors responded successfully. |  |
| Profile-Registration-1 | Navigate to the Register Page and fill in the client’s basic information. | The system should successfully add the information to the database, and it should appear on the profile details inside the main page panel. | The system responded quickly and successfully added the information to the database. |  |