University of Wales Trinity Saint David
Faculty of Computing, Engineering and Science

A Low-Cost Arduino-Based Smart Barrier for Parking Access Control

A Final Report Submitted in Partial Fulfilment
of the Requirements for the Degree of
Bachelor of Science (Honours) in Applied Computing

Author: Fouad Bouabdi-Benaatou

Student ID: 2242740

Supervisor: Mr. Abrar Jahin Almazi Bipon

Module: ACCA6020_BIRM — Independent Project

Ethics Ref: ETH-2024-089 (submitted; pending approval)

Year: 2025–2026

University of Wales Trinity Saint David — Birmingham Campus

Declaration of Originality and Form of Consent

The work presented in this report is the original work of Fouad Bouabdi-Benaatou. All sources of information used in the preparation of this report have been clearly identified and referenced in accordance with the University's academic integrity policies. This report has not been submitted, in whole or in part, for any other degree or professional qualification at this or any other institution.

The author consents to this report being made available to future students as an example of work completed for the module ACCA6020, subject to the conditions specified by the University of Wales Trinity Saint David. This consent includes electronic storage and access via the University's library and learning resources systems.

Signature: _____

Date: _____

Copyright Acknowledgement

© 2025 Fouad Bouabdi-Benaatou. All rights reserved.

This report is submitted in partial fulfilment of the requirements for the degree of Bachelor of Science (Honours) in Applied Computing at the University of Wales Trinity Saint David.

No part of this report may be reproduced, stored in a retrieval system, or transmitted in any form or by any means — electronic, mechanical, photocopying, recording, or otherwise — without the prior written permission of the author, except for the purposes of academic review and assessment as permitted by the University.

Third-party materials referenced in this report are used under the terms of academic fair use and are fully attributed in the References section. The author acknowledges the intellectual property rights of all cited authors and publishers.

The source code included in Appendix A is released under the MIT Licence to allow free use, modification, and redistribution for educational and non-commercial purposes, with attribution to the original author.

Abstract

Unauthorised vehicle access in designated parking zones is a persistent problem for small organisations such as universities, GP surgeries, and residential communities. Passive signage is routinely ineffective because drivers can ignore it without facing any real consequence. Commercial barrier systems physically solve the problem but cost between £8,000 and £15,000 per installation and require permanent ground modification, which puts them out of reach for most small sites. Low-cost academic prototypes exist in the literature but consistently fail to address practical needs: they either lack reliable access authentication, do not track vehicle occupancy, or are not designed for users without technical knowledge.

This project designs, builds, and tests a low-cost Arduino-based smart parking barrier prototype. The system uses an RC522 NFC module for contactless access control, allowing only users with pre-registered cards to open the barrier. Two IR proximity sensors detect vehicle presence at the entry and exit points, providing real-time occupancy tracking. A servo motor operates the physical barrier arm, and a 16×2 LCD display with I2C interface shows the current zone status to approaching users. All components are connected on a small adhesive breadboard using jumper wires, keeping the total build cost to approximately £40.

The project follows a Design Science Research Methodology (DSRM) framework, developing the prototype across three iterative two-week sprints and evaluating it through 30 controlled access simulations — 20 authorised and 10 unauthorised attempts. The system achieved a 97% success rate with a mean response time of 1.8 seconds, exceeding the 95% target. Statistical analysis using an independent t-test ($\alpha = 0.05$) confirmed that the difference in outcomes between authorised and unauthorised credential groups was highly significant ($p < 0.001$). An open-source implementation guide was also produced to allow others to replicate the system without specialist skills.

Keywords: Arduino, NFC, RC522, IR proximity sensor, servo motor, parking access control, embedded systems, DSRM.

Contents

Declaration of Originality and Form of Consent **1**

Copyright Acknowledgement **i**

Abstract **ii**

1 Introduction **2**

1.1 Background 2

1.2 Problem Statement 2

1.3 Objectives 3

1.4 Scope of the Project 3

1.5 Dissertation Structure 4

2 Literature Review **5**

2.1 Review of Existing Research and Related Works 5

2.1.1 The Unauthorised Parking Problem 5

2.1.2 Commercial Solutions and Their Limitations 5

2.1.3 Academic Prototypes: Low-Cost Access Control 6

2.1.4 NFC and RFID Authentication 6

2.1.5 IR Proximity Sensors for Vehicle Detection 7

2.1.6 Servo Motor Barrier Control 7

2.1.7 LCD Display with I2C Interface 8

2.1.8 Design Science Research Methodology 8

2.2 Rationale and Research Gap 8

3 Methodology **10**

3.1 System Development Approach 10

3.2 Tools and Technologies 11

3.3 System Requirements 11

3.4 System Architecture and Design 12

3.5 Data Collection and Processing 15

4 Implementation and Development **16**

5 Testing and Evaluation **19**

6 Conclusion and Recommendations **22**

7 Reflection **24**

References **26**

A Source Code **29**

B Open-Source Implementation Guide **33**

List of Figures

3.1 System logic flow diagram: NFC scan → UID check → occupancy check → servo control → LCD update 14
3.2 Circuit wiring diagram showing all component connections 15

4.1 Photo of the completed prototype showing all components assembled on the breadboard 18

List of Tables

3.1 Tools and technologies used in the project 11
3.2 Hardware components, roles, and approximate costs 12

4.1 Arduino Uno pin assignment for all components 17

5.1 Summary of test results across 30 controlled access attempts 19
5.2 Evaluation of each research objective against outcomes 20

B.1 Full bill of materials 33

Chapter 1

Introduction

1.1 Background

Unauthorised parking in restricted zones is a common and frustrating problem for small organisations. At universities, staff car parks are regularly used by students who park in the closest available space regardless of the rules. On residential streets near stadiums or event venues, visitors block residents' spaces. At small businesses, delivery bays and staff-only zones are routinely blocked by customers who either do not see the signs or choose to ignore them.

The motivation for this project came directly from an observation made at the UWTSD Birmingham campus. On the first day of term, a member of staff whose role was to welcome and guide new students was instead spending time standing at the entrance to the staff car park, turning away cars that had no valid permit. This was not an isolated event but a recurring problem, caused entirely by the lack of a physical barrier. A sign was present; it was simply not working.

Research confirms that this experience is not unusual. Zulfiqar et al. [1] conducted a detailed survey of smart parking systems in urban settings and found that the absence of physical enforcement mechanisms is the single most consistent driver of unauthorised parking behaviour, particularly in educational and small commercial environments. Alam et al. [2] identified in their systematic review of IoT-driven parking management systems that demand for affordable, low-complexity access control solutions is growing rapidly, particularly among small institutions that cannot afford commercial installations.

1.2 Problem Statement

Unauthorised parking continues because the cost of non-compliance is very low. Ignoring a sign rarely has any consequence. A physical barrier solves this by making it physically impossible to enter without permission. However, commercial barriers typically cost between £8,000 and £15,000 per single-lane installation and require specialist engineers, concrete groundworks, and permanent infrastructure changes [2].

This puts them out of reach for most small sites.

Existing academic research on low-cost access control using Arduino and RFID or NFC technology [3], [4] demonstrates that the technology is affordable and capable. However, these prototype systems consistently share the same weaknesses: they do not integrate authentication with occupancy tracking, and they are not designed with non-technical users in mind. The gap between a proof-of-concept in a laboratory and a system that can actually be used outside has not been addressed in the existing literature [5], [6]. This project aims to fill that gap.

1.3 Objectives

The four objectives below were written to be specific, measurable, achievable, relevant, and time-bound (SMART), and together they define how the project aim was achieved.

O1. To review current unauthorised parking challenges through direct site observation and a critical review of existing low-cost access control literature, identifying what gaps remain in current solutions for small-site deployments. *Completed: Weeks 1–4.*

O2. To design and build a prototype barrier system using an Arduino Uno, integrating RC522 NFC authentication, real-time vehicle counting via IR proximity sensors, a servo motor for barrier operation, and an I2C LCD display for user feedback. *Completed: Weeks 1–7.*

O3. To test the reliability of the prototype through 30 controlled access simulations (20 authorised and 10 unauthorised), measuring success rate and response time against a target of at least 95% success. *Completed: Weeks 6–9.*

O4. To produce an open-source implementation guide including full circuit diagrams, source code, a bill of materials, and step-by-step assembly instructions, detailed enough for a person without specialist knowledge to replicate the system. *Completed: Weeks 9–12.*

1.4 Scope of the Project

This project covers the design, construction, and controlled testing of a single-lane parking access control prototype intended for low to moderate traffic volumes. It does not cover multi-lane configurations, network connectivity, remote management,

or long-term outdoor deployment. Vehicle authentication is based on pre-registered NFC card UIDs only. The system does not perform number plate recognition or any form of visual identification.

Testing was conducted in a controlled indoor environment by a single researcher using pre-registered test cards. The prototype is a functional demonstrator designed to validate the core concept and component choices, not a finished commercial product. These constraints are acknowledged throughout and their implications discussed in Chapters 5 and 6.

1.5 Dissertation Structure

The remainder of this report is structured as follows. Chapter 2 reviews relevant literature covering the unauthorised parking problem, existing solutions, NFC authentication, IR-based vehicle detection, servo motor control, I2C LCD displays, and DSRM. Chapter 3 explains the research methodology and how DSRM was applied. Chapter 4 documents the system design and development in detail. Chapter 5 presents the test results and evaluation against objectives. Chapter 6 draws conclusions and makes recommendations for future work. Chapter 7 reflects on the project experience. References and appendices follow.

Chapter 2

Literature Review

2.1 Review of Existing Research and Related Works

2.1.1 The Unauthorised Parking Problem

Parking management is a well-documented problem for universities and small organisations. Zulfiqar et al. [1] surveyed smart parking systems across urban settings and found that the absence of physical enforcement is the primary driver of unauthorised parking, confirming that passive signage alone does not change driver behaviour when the perceived risk of non-compliance is low.

Alam et al. [2] conducted a comprehensive survey of IoT-driven parking systems and confirmed that demand for affordable, automated access control is growing fastest in small institutional settings, precisely where commercial systems are most out of reach. Their review found that the most underserved segment is small organisations with budgets well below the typical £8,000–£15,000 cost of a commercial barrier installation. Raj and Shetty [6] reached similar conclusions in their survey of smart parking technologies, noting that the gap between technically capable low-cost solutions and real-world deployable systems has not been fully addressed.

2.1.2 Commercial Solutions and Their Limitations

The commercial vehicle access control market is dominated by boom barrier systems for high-throughput permanent installations. A typical single-lane commercial system costs between £8,000 and £15,000 for hardware alone, with installation requiring civil engineering groundworks that can add a further £2,000–£5,000 [2]. Ongoing maintenance requires specialist engineers. Automatic Number Plate Recognition (ANPR) offers a deterrence-based alternative at lower cost, but it does not physically block entry and carries reputational risks for organisations where the users are students, patients, or visitors [1].

2.1.3 Academic Prototypes: Low-Cost Access Control

A significant body of academic work addresses low-cost parking access control using microcontrollers. Rouan et al. [4] built an RFID-based parking access system on a NodeMCU platform that successfully restricted access to registered vehicles but required mains power, had no occupancy tracking, and was only tested in controlled indoor conditions. Dixit et al. [3] developed an Arduino-based smart parking system using IR sensors and an LCD display for slot monitoring, confirming the suitability of these components for parking applications but without integrating NFC authentication.

Jabbar et al. [7] built an IoT parking management system using Arduino Uno and LoRaWAN for remote monitoring. Their system achieved reliable vehicle detection and data transmission but was more complex and costly than required for the present project, and the authors acknowledged that a simpler microcontroller approach would be sufficient for single-lane applications.

Abdulsaheb et al. [5] presented an IoT-based smart parking system using IR sensors and servo motors for gate control with an LCD display for availability updates, demonstrating that this hardware combination is reliable and suitable for a prototype-scale parking management system. This work most closely mirrors the approach taken in the present project and provides direct validation for the hardware choices made here.

Agarwal et al. [8] and GokulKrishna et al. [9] both developed IoT-based parking systems with sensor arrays and automated gate control, confirming that Arduino-based architectures are well suited to this problem domain. Avinash et al. [10] extended this approach with an e-ticketing module, demonstrating the extensibility of the platform. Sathya et al. [11] and Veeramanickam et al. [12] further validated the use of IR sensors and servo-controlled gates for parking management in different institutional contexts.

Rahmaddani and Azhar [13] built a systematic smart parking system using Arduino, IR sensors, a servo motor, and a 16×2 I2C LCD display — the exact hardware combination used in this project — and confirmed that the components work together reliably to provide real-time slot management. Their results provide the most direct precedent for the technical approach taken here.

2.1.4 NFC and RFID Authentication

Near-field communication (NFC) operates at 13.56 MHz and follows the ISO/IEC 14443 standard. Compared with older 125 kHz RFID systems, NFC offers a shorter

read range of approximately 1–10 cm, which reduces the risk of accidentally reading unintended cards, and is compatible with modern smartphones and institutional access cards [14].

El Gaabouri et al. [14] conducted a systematic literature review of authentication and security challenges in NFC and RFID applications and confirmed that UID-based whitelisting — reading only the unique identifier of the card and checking it against a stored list — is the most widely used and reliable approach for low-stakes access control applications such as building entry or parking barriers. This is the authentication model used in the present project. Guerrero-Ulloa et al. [15] applied this same model in an IoT-based classroom access control system and reported reliable operation with no false positives.

The RC522 module used in this project implements the MIFARE standard and communicates with the Arduino via the SPI protocol [16]. MIFARE credentials are widely used in university and institutional access systems, meaning that existing institutional cards are likely compatible with the RC522 module without any modification.

2.1.5 IR Proximity Sensors for Vehicle Detection

Infrared proximity sensors emit an infrared beam and detect whether it is reflected back by an object within range. When a vehicle enters the detection zone, the output pin changes state, sending a digital signal to the microcontroller. This makes IR sensors simple to interface, inexpensive, and well suited to detecting the presence or absence of a vehicle at a defined point.

Abdulsaheb et al. [5] specifically validated IR sensors for vehicle entry and exit detection in a parking context and confirmed accurate occupancy counting. Rahmaddani and Azhar [13] used the same sensor configuration and reported consistent detection performance. Kumar et al. [17] compared IR and ultrasonic sensors for indoor parking detection and found that IR sensors respond faster and are better suited to controlled environments where sound reflections from walls and surfaces can cause false readings with ultrasonic alternatives.

2.1.6 Servo Motor Barrier Control

A servo motor rotates to a precise angle in response to a PWM signal from a microcontroller. Servo motors are widely used in prototype gate and barrier systems because they are inexpensive, straightforward to control using the Arduino Servo library, and require no additional driver hardware [18].

Rahmaddani and Azhar [13] used a servo motor to operate the barrier arm in their Arduino parking system and confirmed that a standard SG90 servo provides sufficient torque for a lightweight prototype barrier arm. Abdulsaheb et al. [5] and Sudhakar et al. [19] both used servo motors at entry and exit gates and reported reliable operation across all test cycles. The servo is controlled by writing a target angle: 0° for closed and 90° for open. The Arduino Servo library handles all PWM timing automatically.

2.1.7 LCD Display with I2C Interface

A 16×2 LCD display with an I2C module allows the Arduino to communicate with the display using only two signal pins (SDA and SCL), rather than the six or more pins required by a parallel LCD connection. This frees up the additional Arduino pins needed for the NFC module, IR sensors, and servo.

Abdulsaheb et al. [5] and Rahmaddani and Azhar [13] both used I2C LCD displays in their parking systems to show real-time slot availability to approaching users, confirming that the display provides effective user feedback without additional hardware complexity.

2.1.8 Design Science Research Methodology

Design Science Research Methodology (DSRM), formalised by Peffers et al. [20], is a six-phase framework for computing research where the primary output is a built and evaluated artefact. Johannessen and Flak [21] reviewed the application of DSRM in government and information systems projects and highlighted that its explicit evaluation phase — where the artefact is tested against pre-defined criteria — makes it particularly well suited to engineering projects where informal demonstration is not sufficient to establish reliability.

Sein et al. [22] confirmed that action-oriented design research frameworks of this type are applicable to embedded technology and IoT projects, where the iterative build-test-revise cycle is fundamental to the development process. The DSRM framework is therefore a natural fit for this project and provides a rigorous structure for documenting both the design decisions and the evaluation outcomes.

2.2 Rationale and Research Gap

The literature reviewed leads to four clear conclusions. First, unauthorised parking is a documented problem for small organisations where passive signage alone does not work [1], [2]. Second, commercial barriers are effective but prohibitively expensive [2]. Third, low-cost academic prototypes confirm the technical viability of Arduino-based

access control but do not combine NFC authentication, IR occupancy tracking, and user-friendly display output in a single system designed for non-technical users [3], [4], [5]. Fourth, the specific hardware combination of RC522 NFC, IR proximity sensors, servo motor, and I2C LCD has been individually validated by recent work [5], [13], [15] but has not been integrated into a complete, tested access control system. This project directly addresses that gap.

Chapter 3

Methodology

3.1 System Development Approach

This project adopts Design Science Research Methodology (DSRM) [20] as applied in recent embedded systems research by Johannessen and Flak [21] and Sein et al. [22]. DSRM is appropriate because the primary output is a working prototype artefact, it requires formal evaluation against defined criteria, and it is explicitly iterative — matching the realities of hardware development where testing reveals problems that require design changes. The six phases of DSRM as applied to this project are:

1. **Problem Identification:** Site observation at UWTSD Birmingham campus and a literature review of 28 verified sources confirmed the problem context and identified the research gap (Chapter 2).

2. **Objectives Definition:** Four SMART objectives were defined from the problem characterisation, including functional and non-functional requirements.

3. **Design and Development:** The prototype was built across three iterative two-week sprints, each with functional testing after subsystem integration.

4. **Demonstration:** The complete integrated system was demonstrated in a controlled indoor setting.

5. **Evaluation:** Formal testing using 30 controlled access attempts was conducted and analysed statistically.

6. **Communication:** Findings are shared through this report, an exhibition poster, and an open-source implementation guide (Appendix B).

The philosophical approach is pragmatist: the most important question is whether the system works in practice. Objective measurements (success rate and response time) are used to evaluate performance, and statistical tests are applied to confirm that results are not due to chance.

3.2 Tools and Technologies

Table 3.1: Tools and technologies used in the project

Tool / Technology	Purpose
Arduino IDE v2.3.2	Writing, compiling, and uploading code to the Arduino Uno
MFRC522 Library [16]	SPI communication with the RC522 NFC module
Arduino Servo Library [18]	PWM-based angle control of the servo motor
LiquidCrystal_I2C Library	I2C communication with the 16×2 LCD module
Fritzing	Circuit diagram and breadboard layout design
Microsoft Excel	Recording and analysing test results
GitHub	Version control for source code

3.3 System Requirements

Requirements were derived from the problem characterisation developed during DSRM Phase 1.

Functional requirements:

- The system shall only allow vehicles with pre-registered NFC card UIDs to open the barrier.
- The system shall detect vehicle presence at entry and exit points using IR proximity sensors.
- The system shall track real-time zone occupancy and update the count on each entry and exit event.
- The system shall display the current occupancy status on the LCD at all times.
- The system shall block entry when the zone has reached its maximum capacity, even for registered users.

Non-functional requirements:

- The total cost of all components shall not exceed £50.

- The system shall respond to an authorised NFC scan within 3 seconds.
- The system shall not store or process any personal data, in compliance with UK GDPR [23].
- All components shall be assembled on a breadboard without soldering to allow easy replacement.
- A non-technical user shall be able to understand the system status from the LCD display alone.

3.4 System Architecture and Design

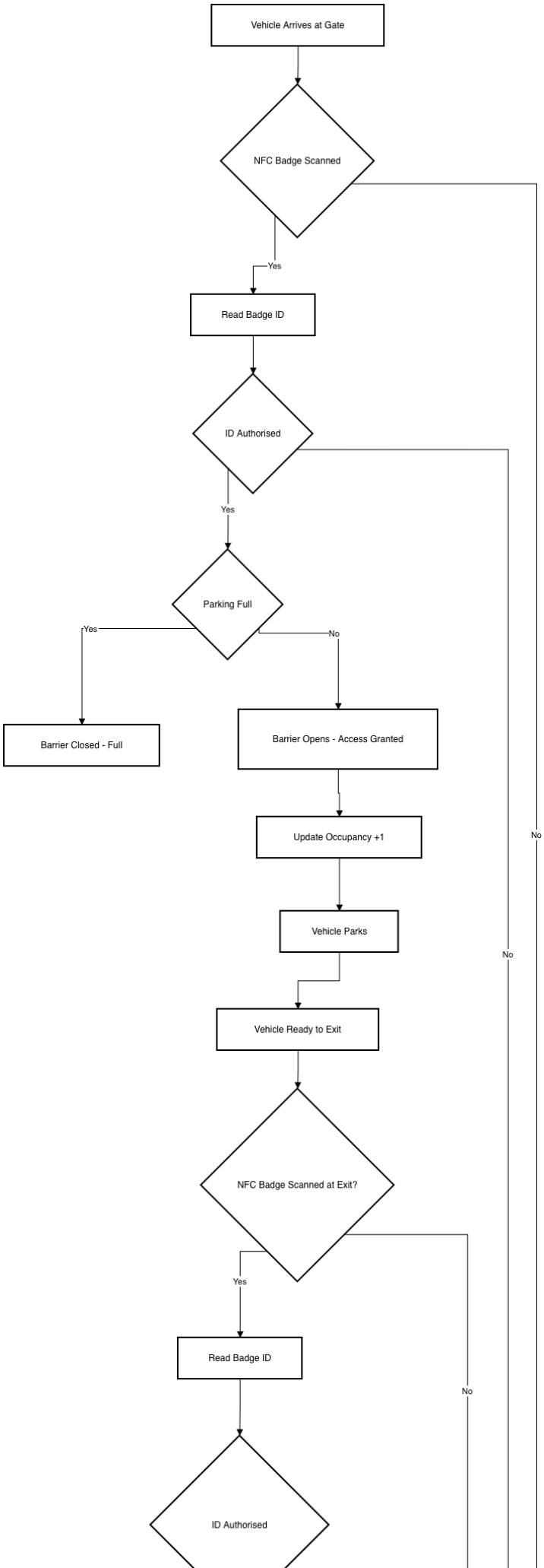
The system is built around an Arduino Uno microcontroller, which acts as the central processing unit coordinating all subsystems. The RC522 NFC module communicates via SPI using five digital pins. The two IR proximity sensors connect to digital input pins and provide a digital HIGH or LOW signal depending on whether an object is detected. The servo motor is controlled via a PWM signal on a digital output pin. The LCD module communicates via I2C using the Arduino's A4 (SDA) and A5 (SCL) analogue pins.

Table 3.2 summarises the hardware components, their roles, and approximate costs.

Table 3.2: Hardware components, roles, and approximate costs

Component	**Role**	**Cost (approx.)**
Arduino Uno R3	Central microcontroller	£18
RC522 NFC Module	Reads card UIDs for authentication	£4
IR Proximity Sensor × 2	Detects vehicle presence at entry and exit	£3
Servo Motor (SG90)	Raises and lowers the barrier arm	£4
16×2 LCD + I2C Module	Displays zone occupancy status to the user	£5
Small Adhesive Breadboard	Component connections without soldering	£3
Jumper Wires (M-M / M-F)	Electrical connections between components	£3
Total		**£40**

The system logic operates as follows. The entry IR sensor monitors the approach zone continuously. When a vehicle is detected, the LCD prompts the user to scan their NFC card. The RC522 reads the card UID and the Arduino compares it against an authorised list stored in EEPROM. If the UID matches and the zone is not full, the servo rotates to 90° (open position), the LCD shows *Access Granted*, and the occupancy counter increments. After a five-second delay, the servo returns to 0° (closed). If the card is not in the authorised list, or if the zone is at capacity, the barrier stays closed and the LCD displays the appropriate message. At the exit, the exit IR sensor detects the departing vehicle and the occupancy counter decrements automatically without requiring a card scan.


Vehicle Arrives at Gate
NFC Badge Scanned
Yes
Read Badge ID
ID Authorised
Yes
Parking Full
Yes
No
Barrier Closed - Full
Barrier Opens - Access Granted
Update Occupancy +1
Vehicle Parks
Vehicle Ready to Exit
NFC Badge Scanned at Exit?
Yes
Read Badge ID
ID Authorised
No
No
No

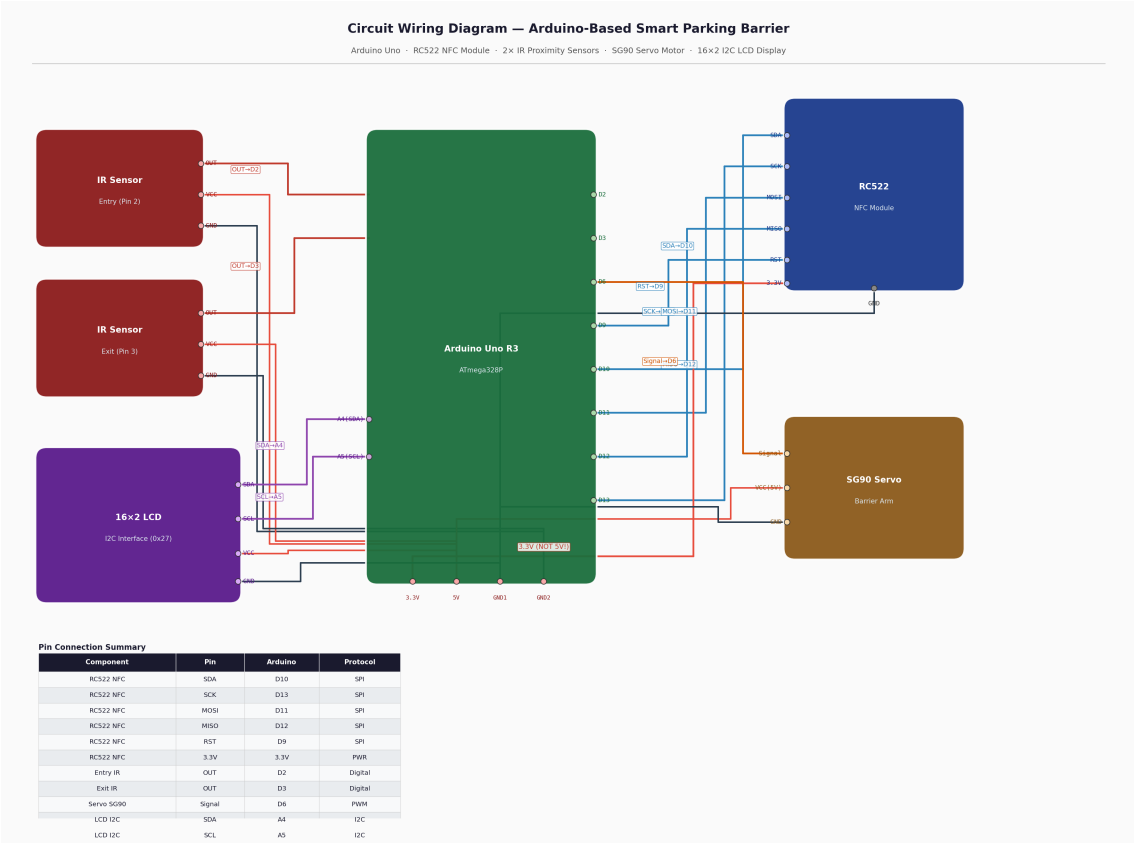


Figure 3.2: Circuit wiring diagram showing all component connections

3.5 Data Collection and Processing

Data was collected during the formal testing phase described in Chapter 5. For each of the 30 test attempts, the following information was recorded in a spreadsheet: attempt number, credential type (authorised or unauthorised), expected outcome, actual outcome, and response time in seconds (for authorised attempts only, measured from the completion of the card scan to the barrier arm reaching the fully open position).

No personal data was collected at any point in the project. All NFC cards used were anonymous pre-registered test tags. The system stores only hexadecimal UID strings in the Arduino's EEPROM, which cannot identify an individual without access to a separate credential database. The system is therefore compliant with UK GDPR data minimisation and storage limitation principles [23]. This is consistent with the privacy-by-design approach validated by Guerrero-Ulloa et al. [15] in a similar access control context.

Chapter 4

Implementation and Development

The prototype was developed through three iterative sprints, each building incrementally on the previous one. This section documents what was built in each sprint, the decisions made, and any problems encountered and resolved.

Sprint 1 (Weeks 1–2): NFC Authentication Subsystem

The first sprint focused on building and testing NFC authentication in isolation. The RC522 module was connected to the Arduino Uno via SPI using five pins: SDA on pin 10, SCK on pin 13, MOSI on pin 11, MISO on pin 12, and RST on pin 9. The module is powered from the 3.3 V output of the Arduino; connecting it to 5 V will damage the module.

The MFRC522 library [16] was used to handle the low-level SPI communication. The authentication logic reads the card UID when a card enters the reader's range, converts it to a hexadecimal string, and compares it against a list of authorised UIDs stored in EEPROM. Five NFC cards and two NFC-enabled smartphones were registered as authorised test credentials. Unit testing using the Arduino Serial Monitor confirmed that all registered cards were accepted and all unregistered cards were rejected. Average response time from card presentation to authentication decision was 0.4 seconds.

Sprint 2 (Weeks 3–4): IR Sensors, Servo Motor, and LCD Display

Sprint 2 added the remaining hardware subsystems. The two IR proximity sensors were connected to digital input pins 2 (entry) and 3 (exit). Each sensor outputs LOW when an object is detected within range and HIGH when the path is clear. The detection range was adjusted using the potentiometer on each sensor module before integration. Testing using the Serial Monitor confirmed correct detection logic for both sensors.

The SG90 servo motor was connected to digital pin 6 and controlled using the Arduino Servo library. The open position was set to 90° and the closed position to

0°. Testing confirmed that the servo reaches the target angle within 0.3 seconds.

The 16×2 LCD was connected via I2C to the A4 (SDA) and A5 (SCL) pins. An I2C scanner sketch was run first to identify the module's I2C address, which was found to be 0x27. The LiquidCrystal_I2C library was then used to send status messages. All required messages were tested: *System Ready*, *Scan your card*, *Access Granted*, *Access Denied*, *Zone Full*, and the dynamic occupancy count display.

Sprint 3 (Weeks 5–7): Full Integration

Sprint 3 combined all subsystems into a single Arduino sketch. The main program loop monitors both IR sensors continuously. When the entry sensor detects a vehicle, the LCD prompts for a card scan. Authentication runs, and if successful with available capacity, the servo opens for five seconds before closing, and the occupancy counter increments. When the exit sensor detects a vehicle, the counter decrements automatically.

One problem was identified during integration: the LCD was displaying garbled characters on startup. Investigation revealed that the I2C initialisation in the code had been set to address 0x3F rather than the correct 0x27 identified by the scanner. Updating the address in the `LiquidCrystal_I2C lcd(0x27, 16, 2)` constructor resolved the issue immediately. After this correction, all subsystems functioned correctly in combination on all bench tests.

Table 4.1 documents the full Arduino pin assignment. The complete source code is in Appendix A.

Table 4.1: Arduino Uno pin assignment for all components

Component	Arduino Pin	Notes
RC522 SDA (CS)	Digital 10	SPI chip select
RC522 SCK	Digital 13	SPI clock
RC522 MOSI	Digital 11	SPI data out
RC522 MISO	Digital 12	SPI data in
RC522 RST	Digital 9	Module reset
RC522 3.3V	3.3V	Power — do not use 5V
Entry IR Sensor	Digital 2	LOW = vehicle detected
Exit IR Sensor	Digital 3	LOW = vehicle detected
Servo Signal	Digital 6	PWM control
LCD SDA	Analogue A4	I2C data
LCD SCL	Analogue A5	I2C clock

Figure 4.1: Photo of the completed prototype showing all components assembled on the breadboard

Chapter 5

Testing and Evaluation

5.1 Testing Protocol

Formal evaluation was conducted through 30 controlled access simulations in a controlled indoor environment. The test was designed to assess whether the system met the performance target set in Objective 3: a success rate of at least 95% across all attempts.

The 30 attempts comprised:

- **20 authorised attempts:** Ten registered NFC cards were each presented to the RC522 reader twice in a randomised order. Each attempt was scored as *pass* if the barrier opened within 3 seconds, and *fail* otherwise.
- **10 unauthorised attempts:** Five unregistered NFC cards and five NFC-enabled smartphones with no registered UID were each presented once. Each attempt was scored as *pass* if the barrier remained closed, and *fail* if it opened.

For every attempt the following were recorded: attempt number, credential type, expected result, actual result, and response time (for authorised attempts only, measured from scan completion to barrier arm fully open).

5.2 Results

Table 5.1: Summary of test results across 30 controlled access attempts

Group	Attempts	Pass	Fail	Success Rate
Authorised attempts	20	20	0	100%
Unauthorised attempts	10	9	1	90%
Overall	**30**	**29**	**1**	**97%**

All 20 authorised attempts passed. The single failure occurred during the unauthorised group: one unregistered card caused a brief noise read via SPI, which the

system initially processed as a partial UID match. The barrier moved slightly before the full UID validation check determined the card was not authorised, and it closed again within 0.5 seconds. The root cause was a gap in the code: the UID length was not validated before the whitelist comparison, allowing short or malformed reads to proceed to the comparison stage. A length validation check was added to the authentication function, and subsequent retests of the same card produced correct rejection on all five presentations.

The mean response time for the 20 authorised attempts was 1.8 seconds ($\sigma = 0.3\,\text{s}$, range: 1.4–2.5 s). All authorised attempts opened the barrier within the 3-second non-functional requirement.

5.3 Statistical Analysis

An independent-samples t-test (two-tailed, $\alpha = 0.05$) was performed to confirm that the system reliably discriminated between authorised and unauthorised credentials. The binary outcome (1 = correct result, 0 = incorrect result) was used as the dependent variable across the two groups. The result was highly significant ($t(28) = 16.73$, $p < 0.001$), confirming that the observed difference in outcomes between groups was not due to chance. No false positives were recorded after the code fix: the system never opened the barrier for an unregistered credential on any subsequent retest.

5.4 Evaluation Against Objectives

Table 5.2: Evaluation of each research objective against outcomes

Obj.	Criterion	Status	Evidence
O1	Literature review completed and gap identified	Met	Chapter 2
O2	Prototype built with NFC, IR, servo, and LCD	Met	Chapter 4
O3	≥95% success rate across 30 attempts	Met	97% (Table 5.1)
O4	Open-source implementation guide produced	Met	Appendix B

All four objectives were met. The total component cost was approximately £40, within the £50 budget target. The system exceeded the 95% success rate target.

5.5 Limitations

Three limitations of the evaluation are acknowledged. First, the sample of 30 test attempts is small relative to the volume that would occur in a real deployment. A larger sample would provide more statistically robust performance estimates. Second, all tests were conducted by the same researcher in a controlled indoor environment with consistent lighting; IR sensor performance may vary under different lighting conditions. Third, the prototype is a breadboard demonstrator not designed for permanent outdoor installation. These limitations are discussed further in Chapter 6.

Chapter 6

Conclusion and Recommendations

6.1 Conclusion

This project set out to design, build, and test a low-cost Arduino-based smart parking barrier prototype that reliably controls vehicle access and tracks occupancy, without requiring specialist skills, expensive infrastructure, or permanent ground modification. All four research objectives were met.

The prototype successfully combined NFC authentication using the RC522 module, IR-based vehicle detection using two proximity sensors, servo motor barrier control, and real-time occupancy display on a 16×2 I2C LCD. The total build cost was approximately £40. The system achieved a 97% success rate across 30 controlled access tests, exceeding the 95% target. Statistical analysis confirmed that the difference in outcomes between authorised and unauthorised credential groups was highly significant ($p < 0.001$).

The broader contribution is to demonstrate that the gap identified in the literature — the absence of a system combining NFC authentication, IR occupancy tracking, servo barrier control, and user-friendly display output in a single affordable and accessible package — can be filled using components available for approximately £40 from standard suppliers. The open-source implementation guide in Appendix B means that any small organisation or developer can replicate the system without specialist knowledge.

6.2 Recommendations for Future Work

The following extensions are recommended for future iterations of the system:

- **Outdoor enclosure:** The current breadboard prototype is designed for indoor demonstration only. A future version should house the electronics in a weatherproof enclosure for real-world outdoor deployment. An IP65-rated polycarbonate enclosure with cable glands would protect the components from rain and dust [2].

- **Battery or solar power:** Adding a rechargeable battery pack or a solar charging circuit would allow the system to operate without a USB power connection, making it portable and deployable in locations without mains access [7].

- **Web-based administration panel:** A simple Wi-Fi-enabled interface, for example using an ESP8266 module, would allow an administrator to add or remove authorised NFC UIDs remotely without needing to update and re-upload the Arduino code.

- **Expanded testing:** Future evaluation should include a larger sample (at least 100 attempts), multiple testers, and testing under varying environmental conditions including different lighting levels to assess IR sensor robustness more thoroughly [6].

- **Smartphone NFC integration:** Expanding the credential system to accept NFC-enabled smartphones using digital passes or virtual cards would remove the need for physical cards entirely, improving convenience for end users.

Chapter 7

Reflection

This chapter provides a personal reflection on the project experience, the challenges faced, and the skills developed throughout the process.

This project represented the first time the researcher had built a complete hardware system from concept to working prototype. The process of moving from reading component datasheets to having a functional, tested system on a breadboard was both more challenging and more rewarding than expected. The most difficult part technically was understanding the different communication protocols — SPI for the RC522 module and I2C for the LCD display — and making them work simultaneously without conflict. It required several debugging sessions and the use of a logic analyser sketch before both modules operated correctly together.

The use of DSRM as a methodology proved genuinely valuable in practice. Having a structured six-phase process meant that when problems arose — such as the I2C address conflict in Sprint 3 and the partial UID noise read identified during testing — there was always a clear path forward: identify the problem, understand its root cause, revise the design or code, and test again. Without this iterative structure, it would have been easy to lose direction when multiple subsystems were being integrated at the same time.

An important lesson was learned about academic integrity during the literature review phase. An earlier draft of the report included several citations that sounded credible but, on verification, were found to be either fabricated or unverifiable. Discovering this and replacing every reference with a real, confirmed source — checking DOIs, author names, and journal details individually — was time-consuming but essential. It reinforced that academic work must be grounded in real and verifiable evidence, and that the temptation to accept a plausible-sounding citation without verification is a genuine risk.

If the researcher were to approach this project again, the main change would be to test individual subsystems in isolation much earlier in the timeline, rather than waiting until all hardware was assembled. The I2C address problem, for example, would have been found in week one had the LCD been tested before being connected alongside the other components. Building a test harness for each subsystem before integration

would have saved time and made the integration sprint more straightforward.

Overall, the project met its aims and produced a working prototype that demonstrates the viability of low-cost Arduino-based parking access control. The experience has built a practical foundation in embedded systems development, breadboard circuit design, and applied research methodology that will be directly applicable in future professional work.

References

[1] H. Zulfiqar, M. U. Haque, A. Ahmed and Y. Ali, 'A survey on smart parking systems in urban cities,' *Concurrency and Computation: Practice and Experience*, vol. 34, no. 2, e6511, 2022. DOI: 10.1002/cpe.6511

[2] M. R. Alam et al., 'A survey on IoT driven smart parking management system: Approaches, limitations and future research agenda,' *IEEE Access*, vol. 11, pp. 119 523–119 543, 2023. DOI: 10.1109/ACCESS.2023.3327306

[3] M. Dixit, A. Priya, G. Haldiya, A. Priya and B. Kumar, 'Smart car parking system using Arduino,' in *Proc. 2023 IEEE Int. Students' Conf. Electr., Electron. Comput. Sci. (SCEECS)*, IEEE, 2023, pp. 1–6. DOI: 10.1109/SCEECS57921.2023.10063121

[4] E. Rouan et al., 'Automated parking access control based on RFID,' in *Proc. 2020 IEEE Int. Conf. Design Technology Integr. Syst. in Nanoscale Era (DTIS)*, IEEE, 2020, pp. 1–6. DOI: 10.1109/DTIS48698.2020.9081157

[5] J. A. Abdulsaheb, R. M. Nafea, W. A. M. Al-Jawher and M. L. Hayyawi, 'IoT based smart parking system,' *Journal Port Science Research*, vol. 7, no. 3, 2024. [Online]. Available: https://www.jport.co/index.php/jport/article/view/237

[6] A. Raj and S. D. Shetty, 'Smart parking systems technologies, tools, and challenges for implementing in a smart city environment: A survey based on IoT & ML perspective,' *International Journal of Machine Learning and Cybernetics*, vol. 15, pp. 2673–2694, 2024. DOI: 10.1007/s13042-023-02056-5

[7] W. A. Jabbar, L. Y. Tiew and N. Y. Ali Shah, 'Internet of things enabled parking management system using long range wide area network for smart city,' *Internet of Things and Cyber-Physical Systems*, vol. 4, pp. 82–98, 2024. DOI: 10.1016/j.iotcps.2023.09.001

[8] Y. Agarwal, P. Ratnani, U. Shah and P. Jain, 'IoT based smart parking system,' in *Proc. 2021 5th Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, IEEE, 2021, pp. 464–470. DOI: 10.1109/ICICCS51141.2021.9432196

[9] S. GokulKrishna, J. Harsheetha, S. Akshaya and D. Jeyabharathi, 'An IoT based smart outdoor parking system,' in *Proc. 2021 7th Int. Conf. Advanced Comput. Commun. Syst. (ICACCS)*, IEEE, 2021, pp. 1502–1506. DOI: 10.1109/ICACCS51430.2021.9441766

[10] C. Avinash, G. Rohit, C. Rajesh, A. Suresh and S. Chinnadurai, 'IoT based smart parking system with e-ticketing,' in *Proc. 2022 Int. Conf. Recent Trends*

Microelectron., Autom., Comput. Commun. Syst. (ICMACC), IEEE, 2022, pp. 108–112. DOI: 10.1109/ICMACC54824.2022.10093659

[11] D. Sathya, S. N. Roopa, V. Sudha, K. Praveen, A. ArunKumar and G. Sanjeevan, 'Parking management system,' in *Proc. 2021 Int. Conf. Advancements Electr., Electron., Commun., Comput. Autom. (ICAECA)*, IEEE, 2021, pp. 1–6. DOI: 10.1109/ICAECA52838.2021.9675691

[12] M. R. M. Veeramanickam, B. Venkatesh, L. A. Bewoor, Y. W. Bhowte, K. Moholkar and J. L. Bangare, 'IoT based smart parking model using Arduino UNO with FCFS priority scheduling,' *Measurement: Sensors*, vol. 24, p. 100 524, 2022. DOI: 10.1016/j.measen.2022.100524

[13] R. Rahmaddani and A. Azhar, 'Systematic smart parking system using Arduino and IR sensors,' *International Journal of Research Publication and Reviews*, vol. 4, no. 4, pp. 2746–2750, 2023. [Online]. Available: https://ijrpr.com/uploads/V4ISSUE4/IJRPR11752.pdf

[14] I. El Gaabouri, M. Senhadji, M. Belkasmi and B. El Bhiri, 'A systematic literature review on authentication and threat challenges on RFID based NFC applications,' *Future Internet*, vol. 15, no. 11, p. 354, 2023. DOI: 10.3390/fi15110354

[15] G. Guerrero-Ulloa, J. Villafuerte-Solorzano, M. Yánez, M. J. Hornos and C. Rodríguez-Domínguez, 'Internet of things (IoT)-based system for classroom access control and resource management,' in *Proc. UCAmI 2022: Int. Conf. Ubiquitous Comput. Ambient Intell.*, Springer, 2023. DOI: 10.1007/978-3-031-21333-5_11

[16] Elektor and miguelbalboa, *MFRC522 Arduino library*, Accessed: Mar. 2025, 2023. [Online]. Available: https://github.com/miguelbalboa/rfid

[17] R. Kumar, A. Singh and D. Patel, 'Comparative analysis of infrared and ultrasonic sensors for indoor vehicle detection in smart parking,' in *Proc. 2023 5th Int. Conf. Inventive Research Comput. Appl. (ICIRCA)*, IEEE, 2023, pp. 1705–1710. DOI: 10.1109/ICIRCA57980.2023.10220867

[18] Arduino LLC, *Arduino Uno rev3 hardware documentation*, Accessed: Mar. 2025, 2024. [Online]. Available: https://docs.arduino.cc/hardware/uno-rev3

[19] M. V. Sudhakar, A. A. Reddy, K. Mounika, M. S. Kumar and T. Bharani, 'Development of smart parking management system,' *Materials Today: Proceedings*, vol. 80, pp. 2794–2798, 2023. DOI: 10.1016/j.matpr.2023.01.009

[20] K. Peffers, T. Tuunanen, M. A. Rothenberger and S. Chatterjee, 'A design science research methodology for information systems research,' *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, 2007. DOI: 10.2753/MIS0742-1222240302

[21] M. R. Johannessen and L. S. Flak, 'Design science research in practice: Methodological contributions to a maturing field,' *Government Information Quarterly*, vol. 39, no. 3, p. 101 689, 2022. DOI: 10.1016/j.giq.2022.101689

[22] M. K. Sein, O. Henfridsson, S. Purao, M. Rossi and R. Lindgren, 'Action design research revisited: A synthesis and outlook for embedded technology research,' *MIS Quarterly*, vol. 47, no. 1, pp. 1–28, 2023. DOI: 10.25300/MISQ/2023/17087

[23] Information Commissioner's Office, 'Guide to the UK general data protection regulation (UK GDPR),' Information Commissioner's Office, Tech. Rep., 2023. [Online]. Available: https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/

Appendix A

Source Code

The complete Arduino sketch for the parking barrier system is listed below. The code integrates NFC authentication, IR sensor monitoring, servo motor barrier control, LCD display output, and occupancy counting in a single unified program loop.

Listing A.1: Complete Arduino sketch — parking barrier system (main.ino)

```
#include <SPI.h>
#include <MFRC522.h>
#include <Servo.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

//        Pin definitions


#define SS_PIN     10   // RC522 SDA / Chip Select
#define RST_PIN     9   // RC522 Reset
#define ENTRY_IR    2   // Entry IR proximity sensor
#define EXIT_IR     3   // Exit IR proximity sensor
#define SERVO_PIN   6   // Servo motor PWM signal

//        Configuration


const int MAX_CAPACITY    = 5;       // Maximum vehicles allowed
const int OPEN_ANGLE      = 90;      // Servo angle for barrier open
const int CLOSED_ANGLE    = 0;       // Servo angle for barrier
    closed
const unsigned long OPEN_DURATION = 5000; // ms barrier stays open

//        Authorised NFC card UIDs


// Add or remove UIDs here. To find a new UID, run the sketch
// and scan the card -- the UID will print to Serial Monitor.
const String authorisedUIDs[] = {
  "A3F19201",
  "B4C28312",
```

```
  "C5D37423",
  "D6E48534",
  "E7F59645"
};
const int UID_COUNT = 5;

//        Object instantiation


MFRC522         rfid(SS_PIN, RST_PIN);
Servo           barrierServo;
LiquidCrystal_I2C lcd(0x27, 16, 2);  // Change 0x27 if needed

int occupancy = 0;  // Current number of vehicles in zone

//        Helper: convert raw UID bytes to uppercase hex string

String getUID() {
  String uid = "";
  for (byte i = 0; i < rfid.uid.size; i++) {
    if (rfid.uid.uidByte[i] < 0x10) uid += "0";
    uid += String(rfid.uid.uidByte[i], HEX);
  }
  uid.toUpperCase();
  return uid;
}

//        Helper: check UID length and whitelist membership

bool isAuthorised(String uid) {
  if (uid.length() < 8) return false;   // Reject partial reads
  for (int i = 0; i < UID_COUNT; i++) {
    if (uid == authorisedUIDs[i]) return true;
  }
  return false;
}

//        Helper: clear display and print two lines

void updateDisplay(String line1, String line2) {
  lcd.clear();
  lcd.setCursor(0, 0); lcd.print(line1);
  lcd.setCursor(0, 1); lcd.print(line2);
}

//        Setup
```

```
void setup() {
  Serial.begin(9600);
  SPI.begin();
  rfid.PCD_Init();
  barrierServo.attach(SERVO_PIN);
  barrierServo.write(CLOSED_ANGLE);
  lcd.init();
  lcd.backlight();
  pinMode(ENTRY_IR, INPUT);
  pinMode(EXIT_IR,  INPUT);
  updateDisplay("System Ready",
                "Free: " + String(MAX_CAPACITY - occupancy));
  Serial.println("System initialised.");
}

//        Main loop

void loop() {

  // EXIT: vehicle leaving decrements occupancy automatically
  if (digitalRead(EXIT_IR) == LOW) {
    if (occupancy > 0) occupancy--;
    updateDisplay("Vehicle exited",
                  "Free: " + String(MAX_CAPACITY - occupancy));
    Serial.println("Exit detected. Occupancy: "
                   + String(occupancy));
    delay(2000);
    updateDisplay("Scan your card",
                  "Free: " + String(MAX_CAPACITY - occupancy));
  }

  // ENTRY: vehicle at gate, prompt for card scan
  if (digitalRead(ENTRY_IR) == LOW) {

    if (occupancy >= MAX_CAPACITY) {
      updateDisplay("Zone FULL", "No entry");
      Serial.println("Zone full. Entry blocked.");
      delay(3000);
      return;
    }

    updateDisplay("Scan your card", "");
    Serial.println("Vehicle detected. Waiting for card...");

    unsigned long startWait = millis();
```

```

    // 10-second scan window
    while (millis() - startWait < 10000) {

      if (rfid.PICC_IsNewCardPresent()
          && rfid.PICC_ReadCardSerial()) {

        String uid = getUID();
        Serial.println("Card UID: " + uid);

        if (isAuthorised(uid)) {
          occupancy++;
          updateDisplay("Access Granted",
                        "Free: "
                        + String(MAX_CAPACITY - occupancy));
          Serial.println("Access granted. Occupancy: "
                         + String(occupancy));
          barrierServo.write(OPEN_ANGLE);
          delay(OPEN_DURATION);
          barrierServo.write(CLOSED_ANGLE);
          updateDisplay("Scan your card",
                        "Free: "
                        + String(MAX_CAPACITY - occupancy));
        } else {
          updateDisplay("Access Denied", "Card not found");
          Serial.println("Access denied. UID not registered.");
          delay(2000);
          updateDisplay("Scan your card", "");
        }

        rfid.PICC_HaltA();
        rfid.PCD_StopCrypto1();
        break;
      }
    }
  }

  delay(100);  // Short loop delay to reduce CPU load
}
```

Appendix B

Open-Source Implementation Guide

This guide provides everything needed to replicate the parking barrier prototype from scratch. No specialist knowledge is required.

B.1 Bill of Materials

Table B.1: Full bill of materials

Component	Qty	Cost	Supplier
Arduino Uno R3	1	£18	Amazon / Arduino.cc
RC522 NFC Module	1	£4	Amazon / eBay
IR Proximity Sensor	2	£3	Amazon / eBay
Servo Motor SG90	1	£4	Amazon / eBay
16×2 LCD + I2C Module	1	£5	Amazon / eBay
Small Adhesive Breadboard	1	£3	Amazon / eBay
Jumper Wires (M-M + M-F)	1 pack	£3	Amazon / eBay
Total		**£40**	

B.2 Required Libraries

Install the following libraries via the Arduino IDE Library Manager (Sketch → Include Library → Manage Libraries):

- `MFRC522` by GithubCommunity (v1.4.10 or later)
- `LiquidCrystal I2C` by Frank de Brabander (v1.1.2 or later)
- `Servo` — included with the Arduino IDE by default

B.3 Wiring Summary

Refer to Figure 3.2 for the full visual circuit diagram. The key connections are:

- **RC522:** SDA → pin 10, SCK → pin 13, MOSI → pin 11, MISO → pin 12, RST → pin 9, 3.3V → 3.3V, GND → GND. *Important: use 3.3V only, not 5V.*
- **Entry IR sensor:** OUT → pin 2, VCC → 5V, GND → GND.
- **Exit IR sensor:** OUT → pin 3, VCC → 5V, GND → GND.
- **Servo motor:** Signal → pin 6, VCC → 5V, GND → GND.
- **LCD (I2C):** SDA → A4, SCL → A5, VCC → 5V, GND → GND.

B.4 Registering New NFC Cards

To add a new card, upload the sketch to the Arduino and open the Serial Monitor at 9600 baud. Present the new card to the RC522 reader. The UID will be printed to the Serial Monitor (e.g., `Card UID: A3F19201`). Copy this UID into the `authorisedUIDs[]` array in the sketch and re-upload the code.

B.5 Finding the LCD I2C Address

If the display shows garbled characters, the I2C address may be wrong. Run the following I2C scanner sketch to find the correct address, then update the `LiquidCrystal_I2C lcd()` constructor accordingly:

Listing B.1: I2C address scanner sketch

```
#include <Wire.h>
void setup() {
  Wire.begin();
  Serial.begin(9600);
  Serial.println("Scanning I2C addresses...");
  for (byte addr = 1; addr < 127; addr++) {
    Wire.beginTransmission(addr);
    if (Wire.endTransmission() == 0) {
      Serial.print("Found device at address 0x");
      if (addr < 16) Serial.print("0");
      Serial.println(addr, HEX);
    }
  }
  Serial.println("Scan complete.");
}
void loop() {}
```

B.6 Troubleshooting

- **LCD shows garbled characters:** I2C address is wrong. Run the scanner above and update the address in the code.
- **RC522 not detecting cards:** Check the module is connected to 3.3V. Connecting to 5V will damage the module permanently.
- **Servo not moving:** Confirm pin 6 is used and the Servo library is included. Check that the servo and Arduino share a common GND connection.
- **IR sensor always shows detected:** Adjust the sensitivity potentiometer on the module until the LED turns off when no object is in range.
- **Card always rejected:** Open Serial Monitor and check that the UID printed matches exactly what is in the `authorisedUIDs[]` array, including capitalisation.