A black background with white text

Description automatically generated with low confidence **Assignment Cover Sheet**

| **Subject Code: CSIT 321** | |  | |
| --- | --- | --- | --- |
| **Subject Name: Project** | | | | |  |
| **Submission Type: Group report** | | | | |  |
| **Assignment Title: Software Requirements Specification** | | | | |  |
| **Student Name: Afraaz Ali, Mohammed Maarij Uddin Khan, Muhammad Ibrahim Jafar, Hassanen Alshami, Ali Imad** | | | | |  |
| **Student Number: 7066600, 6222560, 7001915, 6675220, 6255103** | | | | |  |
| **Student Phone: +971 56 101 6265, +971 50 6966480,+971 50 350 7469, +971 56 427 4710, +971 50 764 6157** | | | | |  |
| **Student Email:** [**amfa964@uow.edu.au**](mailto:amfa964@uow.edu.au)**,** [**mmuk953@uowmail.edu.au**](mailto:mmuk953@uowmail.edu.au)**,** [**mij990@uowmail.edu.au**](mailto:mij990@uowmail.edu.au)**,** [**hamas400@uowmail.edu.au**](mailto:hamas400@uowmail.edu.au)**, ais808@uowmail.edu.au** | | | | |  |
| **Lecturer Name: Dr Mai El Barachi** | | | | |  |
| **Due Date: 13 / 02 / 2024** | | | | |  |
| **Date Submitted: 29 / 01 / 2024** | | | | |  |
|  | **PLAGIARISM:** The penalty for deliberate plagiarism is FAILURE in the subject. Plagiarism is cheating by using the written ideas or submitted work of someone else. UOWD has a strong policy against plagiarism.  The University of Wollongong in Dubai also endorses a policy of non-discriminatory language practice and presentation.  **PLEASE NOTE:**STUDENTS MUST RETAIN A COPY OF ANY WORK SUBMITTED | | **DECLARATION:** I/We certify that this is entirely my/our own work, except where I/we have given fully-documented references to the work of others, and that the material contained in this document has not previously been submitted for assessment in any formal course of study. I/we understand the definition and consequences of plagiarism.  **Signature of Student: Maarij, Hassanen, Afraaz, Ibrahim, Ali** | | |



| | ***Optional Marks:*** | | | --- | --- | | ***Comments:*** | |
| --- | --- | --- | --- |

| | **Lecturer Assignment Receipt**(To be filled in by student and retained by Lecturer upon return of assignment) | | | --- | --- | | **Subject:** | **Assignment Title:** | | | **Student Name:** | **Student Number:** | | | **Due Date:** | **Date Submitted:** | | | **Signature of Student:** | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

| | **Student Assignment Receipt** (To be filled in and retained by Student upon submission of assignment) | | | | --- | --- | --- | | **Subject:** | | **Assignment Title:** | | | **Student Name:** | **Student Number:** | | | **Due Date:** | **Date Submitted:** | | | **Signature of Lecturer** | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

[Introduction 4](#_mbtclhurz0wv)

[Background 4](#_257ih1yho4bu)

[Problem Statement 4](#_9vyg2ia8bux3)

[Our solution 4](#_fu8s3zyeod1y)

[Main problems we will solve: 4](#_tyjcwt)

[Target niche 5](#_uu2ifkgb5b13)

[Technologies 6](#_bn85xj43rl8v)

[Gantt Chart 7](#_9p9n312truqq)

[Tools 7](#_2rgny4a0rnu2)

[Feasibility 8](#_40rxifbcqmno)

[Project charter 8](#_2sl1mppv6nww)

[Scope statement 8](#_6bqbd37x3ptu)

[Goals and objectives 8](#_mzb0vp1xn5x9)

[Competitive analysis 9](#_ylms06kbzzs2)

[Pre-requisites 9](#_nj2blyqr4pna)

[Milestones and division of work among team members 10](#_g8kkuw4bc0o)

[Work Breakdown Structure 11](#_ipfhna2rsa36)

[Risk analysis and mitigation plan 11](#_e0xez2ktu0br)

[High level system architecture 13](#_j0vzfn1ard79)

[High level system operation 13](#_hoqzmqxtjkr6)

[Requirements Analysis 14](#_fkwrsxq14va7)

[Design constraints 14](#_1d66ozcczzw4)

[Product functions 15](#_blly2pg2svhn)

[Non-Functional requirements 16](#_4vl2damya6cj)

[Performance Requirements 16](#_qh5gendkycei)

[Safety Requirements / Security Requirements 16](#_17t9z8w7v05)

[Software Quality Attributes 18](#_ek0cidkms9ln)

[Other Requirements 18](#_t0ikmey6e49w)

[Requirements Matrix 18](#_izzyplaqcijy)

[High-Level Design 19](#_ohztagmgh1ug)

[System overview and design considerations 19](#_c0y3v06kvv9l)

[Registration and User Profiles 19](#_tkbtinfpdn54)

[QR Code Scan 20](#_pw84sqqfradu)

[Rewards System 20](#_zk3n9wilquu)

[Texting Feature 20](#_9dswxqt4t5il)

[System architecture and use cases 20](#_ibyfkhrdtdl3)

[Create Account 24](#_xcsmekw4twyt)

[Delete Account 24](#_zf27jyj6equg)

[Login 25](#_90hd7kq15d2k)

[Logout 26](#_punhfdp61zr)

[Change user account settings 27](#_p72lichv6qmx)

[Generate unique QR code for account 27](#_81pqhmwvhsiq)

[Print your QR code 28](#_4ojbu7j274eg)

[Save your QR code to phone gallery 29](#_c23wcc15d4q2)

[Scan a QR code 30](#_4hihs3orrub9)

[Rewards tab 30](#_rv1u3a6qfsob)

[Message feature 31](#_le8rpzi8obt0)

[Pick the reason for QR code scan 32](#_4spdk9wnh9l5)

[View notifications 33](#_365kv3c5t2qu)

[Sell your car feature 34](#_kdmusycbjb9)

[Decomposition Description 34](#_7jwoebtvdpi0)

[Design Rationale 35](#_m5qzl2e4k063)

[Data Description 37](#_hdku441i9ui3)

[Data Dictionary 38](#_bn1o7jc7eqta)

[Component Design 38](#_ka4zvituo5oq)

[Design Components (Detailed) 39](#_8h2v1dqsanvn)

[User Interfaces 41](#_2g6jpb236w4y)

[Hardware Interfaces 41](#_voz65ne5qkpc)

[Software Interfaces 42](#_k89sqs26vwn1)

[Communication Interfaces 42](#_fpx9bltzyafb)

[Low-Level Design 43](#_c0y3v06kvv9l)

[Overview of User Interface 43](#_fsectwmg1h1x)

[Screen Images 43](#_fsectwmg1h1x)

[Low-Fidelity Hand-drawn Sketch 43](#_j01fuqokrcjc)

[Low Fidelity Software Designed Screens 44](#_j62fmxxlctdn)

[High Fidelity Software Created Screens 44](#_t7n503vthfmf)

[Screen Image Description Table 46](#_wdxiaxr0927b)

[Testing 47](#_fsectwmg1h1x)

[Testing goals 47](#_go5o9q4ogh7z)

[Test plan scope 48](#_o0yqhicfh5ht)

[Test forms and test results 49](#_drysqi8lu57n)

[Conclusion and Future Work 62](#_go5o9q4ogh7z)

[Conclusion 62](#_ko869phliptc)

[Strengths and weaknesses 62](#_ko869phliptc)

[Future improvements 64](#_48ed0uoivkcw)

### Introduction

#### Background

Our application aims to address four common automobile-related issues. Firstly, the frustration of being blocked by another car in a parking lot is a significant concern for drivers. Secondly, the hindrance to smooth traffic flow caused by improperly parked cars presents a safety issue. Thirdly, in the event of an accidental collision, locating the car owner for insurance claims becomes a challenge. Lastly, the inconvenience of charging an electric car is exacerbated when fully charged vehicles occupy charging stations for an extended period.

To resolve these issues, we plan to utilize QR codes affixed to cars, allowing users to scan them and notify owners about the specific problems. Additionally, we intend to implement a "sell the car" feature for those interested in selling their vehicles through the application.

#### Problem Statement

Our application aims to address four common automobile-related issues. Firstly, the frustration of being blocked by another car in a parking lot is a significant concern for drivers. Secondly, the hindrance to smooth traffic flow caused by improperly parked cars presents a safety issue. Thirdly, in the event of an accidental collision, locating the car owner for insurance claims becomes a challenge. Lastly, the inconvenience of charging an electric car is exacerbated when fully charged vehicles occupy charging stations for an extended period.

To resolve these issues, we plan to utilize QR codes affixed to cars, allowing users to scan them and notify owners about the specific problems. Additionally, we intend to implement a "sell the car" feature for those interested in selling their vehicles through the application

#### Our solution

A car community software, based on QR code usage where the person causing the block may be directly notified by the person being blocked through an Application message being received after scanning the attached QR code on the blocker’s car. The main motive of using this software for the user is the convenience and pace of direct notification instead of calling the traffic authorities which will therefore result in time consumption and expected financial loss from the paid fine.

#### Main problems we will solve:

* Incorrectly parked vehicles obstructing the parked car of the user.
* Improper spacing in parking areas leading to disruptions in traffic flow.
* Fully charged electric cars occupying charging stations for an extended period.
* Absence of immediate communication between the car causing obstruction and the affected vehicle.
* Accidents occurring in parking lots without a means to communicate with the affected party.

#### Target niche

The target niche for our comprehensive automotive solution encompasses a diverse range of users who value efficiency, convenience, and problem-solving capabilities in their daily experiences with vehicles. This includes:

1. Urban Dwellers and Commuters:

- Individuals living in urban areas with limited parking space.

- Commuters facing challenges such as blocked parking or traffic flow obstructions.

2. Electric Vehicle Enthusiasts:

- Owners of electric vehicles who are passionate about eco-friendly transportation.

- Users committed to using electric charging stations efficiently.

3. Safety-Conscious Drivers:

- Drivers who prioritize safety and wish to efficiently handle accidents in parking lots.

- Individuals seeking a quick and reliable means to communicate with other parties involved in accidents.

4. Tech-Savvy and Problem-Solving Users:

- Tech-savvy individuals who appreciate the convenience of using technology to address common automotive issues.

- Users who prefer immediate and direct communication solutions over traditional methods.

5. Frequent Parkers:

- Individuals who regularly park their cars in various locations, such as workplaces, shopping centers, or public spaces.

- Users looking for a hassle-free way to manage parking-related challenges.

6. Vehicle Owners in High-Traffic Areas:

- Individuals residing or working in areas with high traffic density.

- Users looking for tools to navigate and address traffic-related issues effectively.

7. Community and Environment-Focused Individuals:

- Drivers who want to contribute to a positive community experience by responsibly managing their vehicle-related activities.

- Individuals with an environmental consciousness, particularly those using electric vehicles for sustainability.

8. Car Sellers and Buyers:

- Users interested in selling their cars who appreciate the convenience of an integrated "sell the car" feature.

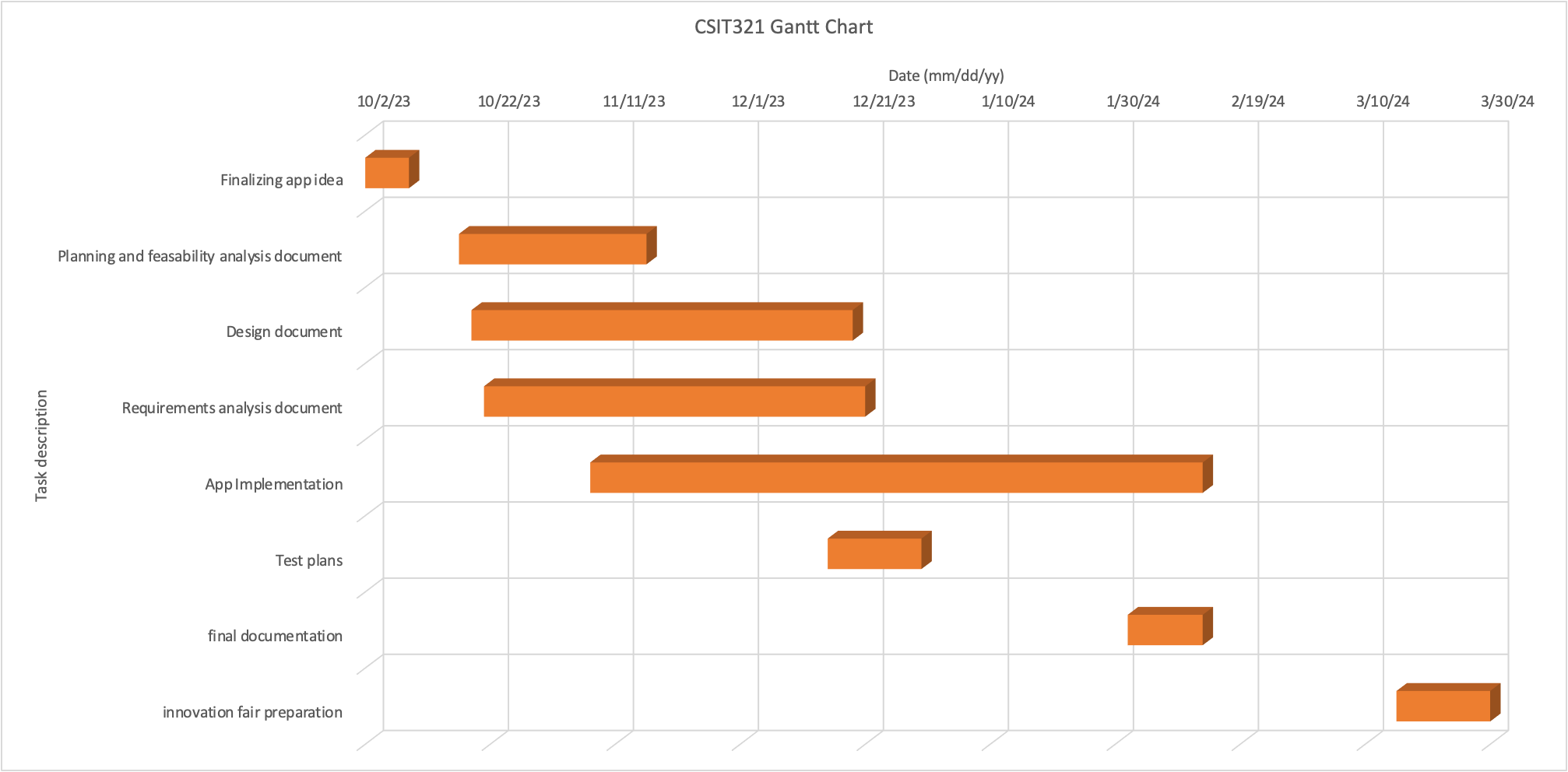
- Potential car buyers looking for a platform that provides additional functionalities beyond traditional selling platforms.

By targeting this diverse audience, our application aims to cater to the multifaceted needs of modern drivers, offering a holistic solution to enhance their overall automotive experience.

#### Technologies

Our app makes use of QR code technology for features such as scanning the QR code that is attached on the other user’s car via the smartphone’s in-built camera technology to notify them on specific scenarios such as I cannot get out of parking, someone hit a vehicle, I want this vehicle and so on as well as QR code generation that the user can perform and either print or save their QR code to their phone gallery to print the QR code later to stick it on their car. Also, Wi-fi technology is used in our app to allow access to the internet for smooth and uninterrupted usage of the app.

#### Gantt Chart



#### 

#### Tools

The Integrated Development Environment (IDE) chosen for crafting this application is Android Studio, a robust platform tailored for Android app development. In the process of app creation, the programming language of choice is Java, requiring installation as an extension to Android Studio to facilitate seamless development. This strategic combination of Android Studio and Java ensures a cohesive and efficient workflow, enabling developers to harness the full potential of the Android ecosystem.

Moreover, the application's backend will be seamlessly integrated with the Firebase Console, a hosting service provided by Google. This integration is pivotal for enhancing the application's functionality and scalability. Leveraging the Firebase Console adds a layer of versatility to the app, empowering it with robust features such as real-time database management, secure user authentication, and efficient cloud storage solutions.

The utilization of Java as the programming language further contributes to the application's versatility and compatibility within the Android Studio environment. Java, a widely adopted and versatile language, ensures that developers can leverage a rich ecosystem of libraries and frameworks, streamlining the development process and enhancing the overall quality of the application.

The Firebase Console, hosted by Google, serves as a centralized hub for managing and optimizing various backend functionalities. This integration is instrumental in fostering a seamless and dynamic user experience. From real-time data synchronization to secure user authentication mechanisms, the Firebase integration aligns with our commitment to providing users with a robust and feature-rich application.

In essence, the selection of Android Studio, coupled with the incorporation of Java and Firebase Console, represents a strategic approach aimed at maximizing the application's potential. This comprehensive toolset ensures not only a smooth development process but also positions the application to offer a sophisticated and user-friendly experience, underpinned by the latest advancements in Android app development and cloud-based solutions.

### Feasibility

#### Project charter

The purpose of this app is for the users to not get frustrated if they get blocked in by a car or try to find a parking spot only to realize someone double parked, and instead of having to call authorities or wait for the person to come back they would instead rather simply scan a QR code on the person’s car and it will notify the other party that they need to move their car in hopes of relieving frustrations and creating a way where people can communicate with those that block them in or park incorrectly without being spammy to the other party and without having to involve the authorities. Other features and scenarios where this application can be used for will be for: if you can't pass safely, hit the other person’s vehicle by mistake, if an electric car is fully charged, requesting to buy a person’s car, and a rewards system that would be used as an incentive for people to use the application.

#### Scope statement

The app will be easy to navigate and will be made to be used as a mobile application. The mobile phone’s camera will be used as the technology to scan the QR code placed on the car. The app will help the user get the person that blocked them to be notified as soon as possible to move the car as there is not much typing or selecting that needs to be done by the user.

#### Goals and objectives

The primary objective of this undertaking is to establish a decentralized communication notification system. This system is designed to facilitate communication between the user and an individual who has been adversely affected or become a victim of an incident such as an accident or blockage. The aim is to create a seamless and efficient platform that fosters direct interaction, allowing users to connect with those directly impacted by an issue. Through this decentralized communication channel, we aspire to enhance the effectiveness of communication in situations involving accidents or blockages, thereby promoting a more responsive and collaborative approach to problem resolution.

The primary advantage that this software offers to users lies in the convenience and speed of receiving direct notifications. This eliminates the need to contact traffic authorities via phone, a process that can be time-consuming and may entail additional financial implications in the form of expected fines. By opting for direct notification through the software, users can expedite the communication process, ensuring prompt awareness of any issues related to traffic violations or concerns. This not only saves valuable time but also mitigates the anticipated financial losses associated with potential fines, contributing to a more efficient and user-friendly experience.

#### Competitive analysis

Our application stands out in a market where there are limited competitors offering similar functionalities. While some existing applications in the automotive domain focus on creating communities for car enthusiasts, facilitating meet-ups, sharing events, and engaging in the exchange of news and parts/cars for sale, our application takes a distinctive approach. Rather than emphasizing social and community aspects, our platform addresses a critical need in the market by providing assistance in situations where individuals typically find themselves stranded or facing inconveniences.

Unlike other applications, our primary focus is on resolving issues arising from blocked parking scenarios. When compared to conventional car community apps, our unique selling proposition lies in the provision of practical solutions to common problems. For instance, in situations where users might otherwise be compelled to wait for the owner of a blocking vehicle or resort to involving authorities, our application steps in to offer a more immediate and user-centric resolution. This sets us apart as a problem-solving application that addresses the practical challenges faced by users in everyday situations.

By differentiating ourselves from existing competitors, we aim to establish our application as an indispensable tool for individuals dealing with real-time issues related to parking and traffic. This strategic positioning aligns with our commitment to providing not just a social platform, but a reliable and efficient solution to everyday problems encountered by drivers.

#### Pre-requisites

Users should ensure they possess all the following requirements :

1. Smartphone Compatibility:

- A mobile device capable of downloading and running the application seamlessly.

2. QR Code Scanning Capability:

- A functional camera on the smartphone capable of scanning QR codes efficiently.

3. QR Code Application:

- QR code stickers provided by the application, which users will affix to their vehicles for identification purposes.

4. Global Mobile Number:

- A registered mobile number from any country, enhancing the application's accessibility for users worldwide.

5. Stable Internet Connection:

- Access to a reliable and stable internet connection to ensure uninterrupted usage of the application.

6. Operating System Compatibility:

- A smartphone with a compatible operating system to guarantee the application's functionality.

These prerequisites collectively ensure that users have the necessary tools, connectivity, and settings to maximize their experience with the application.

#### 

#### Milestones and division of work among team members

Milestones and division of work among team members played a pivotal role in the successful execution of our project. Recognizing the urgency of the situation, wherein we needed to completely overhaul the project within a significantly condensed time frame compared to our initial projections, we adopted a strategic approach to task allocation and collaboration.

The team was divided into two distinct groups, each assigned with specific responsibilities tailored to their expertise and project requirements. One group focused on redoing the documentation of project requirements, specifications, and design considerations, ensuring clarity and coherence in our project roadmap. Meanwhile, the other group delved into the implementation phase, where they translated the outlined plans and specifications into tangible code, features, and functionalities.

Throughout this process, regular and structured meetings were held to monitor progress, address challenges, and facilitate seamless coordination between the two groups. These meetings served as crucial checkpoints, allowing team members to share updates, solicit feedback, and provide assistance where necessary. By fostering an environment of collaboration and mutual support, we ensured that each team member had the resources and guidance needed to excel in their respective roles.

Furthermore, the division of work was carefully orchestrated to capitalize on individual strengths and expertise within the team. Assignments were distributed based on skill sets, experience levels, and domain knowledge, maximizing efficiency and productivity. This strategic approach not only expedited the project timeline but also facilitated a more comprehensive and well-rounded development process.

In essence, our approach to milestones and division of work exemplifies our commitment to adaptability, collaboration, and efficiency in the face of evolving project requirements and constraints. Through effective communication, strategic task allocation, and a shared commitment to excellence, we successfully navigated the challenges inherent in project redevelopment and emerged with a robust and refined solution.

#### Work Breakdown Structure



#### Risk analysis and mitigation plan

1. Security Risks:

- Data Breaches: Potential unauthorized access to sensitive user data, such as personal information and communication logs, could result in data breaches.

- Vulnerabilities in Authentication: Weaknesses in authentication mechanisms may lead to account hijacking or unauthorized access.

- Malicious Activities: QR code tampering or interception of communication channels could facilitate malicious activities, such as phishing or spoofing.

2. Performance Risks:

- Network Dependency: Reliance on internet connectivity for core functionalities may result in performance issues in areas with poor network coverage or during network outages.

3. Compatibility Risks:

- Older Devices: Limited support for older Android versions may exclude users with outdated devices from accessing the application.

6. Operational Risks:

- Service Outages: Downtime or service interruptions in the backend infrastructure, such as Firebase, could disrupt application functionality and user access.

- Insufficient Maintenance: Lack of regular maintenance and updates may lead to software bugs, security vulnerabilities, or performance degradation over time.

Mitigation Strategies:

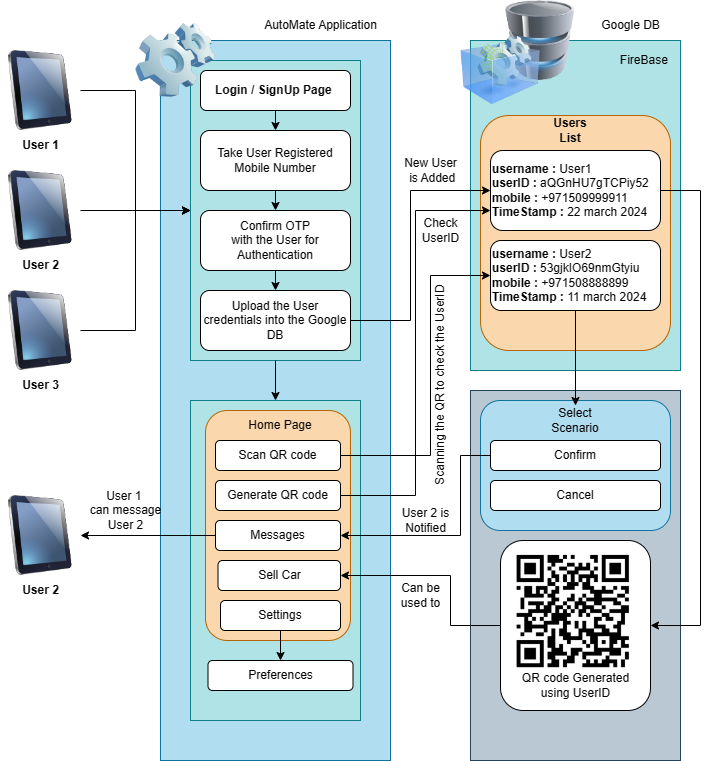
- Implement robust security measures, such as data encryption, secure authentication, and continuous monitoring for suspicious activities.

- Establish contingency plans for operational risks, such as backup systems and disaster recovery protocols.

- Enhance usability through user testing, feedback collection, and design improvements.

- Prioritize compatibility testing across various devices and operating system versions to ensure a consistent user experience.

#### High level system architecture



#### High level system operation

The high-level system operation for your automotive application involves several interconnected components working seamlessly to provide users with a comprehensive solution to common automobile challenges. Here's an overview:

**1. User Registration and Authentication:**

- Users download the application and undergo a registration process using their registered mobile numbers.

- Authentication involves a secure one-time password (OTP) confirmation to validate user identity.

**2. Homepage Navigation:**

- Users are directed to a dynamic homepage offering multiple features and choices.

- Options include scanning a QR code, generating a QR code, messaging, selling a car, and accessing settings.

**3. QR Code Scanning and Generation:**

- Users can scan QR codes affixed to other cars using their smartphone cameras, initiating various functionalities (e.g., reporting a parking issue).

- The application also allows users to generate QR codes for personal identification or sharing with others.

**4. Messaging and Community Interaction:**

- Users engage in real-time messaging to communicate with other drivers, fostering a sense of community.

- Interaction features include reporting issues, providing assistance, or discussing car-related topics.

**5. Car Selling Feature:**

- Users can list their cars for sale within the application, streamlining the selling process and connecting potential buyers.

**6. Settings and Preferences:**

- The settings section provides users with options to personalize their experience, configure preferences, and manage account details.

- Users can log out or switch between mobile numbers as needed.

This high-level system operation encapsulates the key functionalities and interactions within the application, emphasizing a user-centric design, real-time communication, and adaptability to future enhancements. The integration of Firebase as the backend infrastructure ensures a reliable and scalable foundation for the application's seamless operation.

### Requirements Analysis

#### Design constraints

Our primary constraints revolve around the timeline for application development and the logistical challenge of distributing QR code stickers to our user base. However, it's important to note that the creation and implementation of the application itself present manageable tasks that align with our strategic goals.

1. Development Timeframe:

- The temporal factor in crafting the application poses a significant consideration. Balancing the need for swift development with ensuring a high-quality product demands careful planning and resource allocation.

2. Distribution Logistics:

- Overcoming the challenge of efficiently distributing QR code stickers to all users requires strategic coordination. Exploring innovative and streamlined distribution methods will be crucial to ensuring widespread accessibility.

3. Application Management:

- It's worth emphasizing that the application's core development and implementation aspects are well within our control. The robustness and user-friendliness of the application are key focal points, and our team is confident in its ability to bring these aspects to fruition.

4. Strategic Planning:

- Given the constraints, our team is actively engaged in strategic planning to optimize the development process. This involves resource allocation, project management, and a keen focus on meeting deadlines without compromising on the application's quality.

5. User Engagement Strategy:

- Recognizing the importance of user participation, we are formulating a comprehensive strategy to engage our user base effectively. This includes communication channels, feedback mechanisms, and incentivizing user involvement in the application's evolution.

6. Quality Assurance:

- Ensuring the application's reliability and functionality is paramount. Our quality assurance processes involve rigorous testing, user feedback loops, and continuous improvement to guarantee a seamless and dependable user experience.

7. Iterative Development Approach:

- Adopting an iterative development approach allows us to incrementally enhance the application's features. This approach not only accommodates evolving user needs but also enables us to address any unforeseen challenges during the development lifecycle.

In navigating these considerations, our team remains committed to delivering an application that not only meets but exceeds user expectations. Despite the limitations posed by time and distribution logistics, we are confident that our strategic approach will result in a successful and impactful solution for our users.

#### Product functions

* ⁠Create Account
* Generate unique QR code for account.
* Print your QR code (For sticking on car)
* Save your QR code to the phone gallery.
* ⁠ Change user account settings (change the mobile number and password, edit user profile, change language etc)
* ⁠ Login (mobile number and password)
* ⁠Logout
* ⁠Scan a QR code (to notify the person blocking you by their car being incorrectly parked / notify the person that you can’t pass safely / notify the person you accidentally hit their car / notify the person that their car is fully charged / notify the car owner that you are interested in buying their car)
* ⁠Rewards tab (to redeem vouchers and discounts and view your points).
* Message feature between someone interested in buying a car and its seller.
* ⁠Pick the reason for QR code scan so the user gets the proper notification accordingly.
* View notifications
* Sell your car feature (Upload car image and enter car details such as make, model, year, mileage, price)

#### Non-Functional requirements

##### Performance Requirements

As previously mentioned, all that is required to run the application with great performance would be any basic mobile with a camera and is supported for Android 10 or later.

##### Safety Requirements / Security Requirements

1. Advanced User Data Protection Measures:

- Implement sophisticated data encryption protocols to fortify the safeguarding of user data, encompassing all facets of personal information and communication logs.

2. Enhanced Authentication Protocols:

- Deploy robust authentication mechanisms, including multi-factor authentication, to fortify the application against unauthorized access attempts, thus fortifying user account security.

3. Elevated Communication Security:

- Embrace cutting-edge secure communication protocols such as HTTPS to establish a secure conduit for the transmission of user data between the application and backend servers, mitigating potential interception risks.

4. Adherence to Privacy Standards and Regulations:

- Strictly adhere to pertinent privacy regulations and standards, instilling confidence in users through responsible and transparent handling of their data. This involves obtaining explicit consents and adhering to evolving privacy compliance requirements.

5. Rigorous QR Code Security Measures:

- Institute stringent security measures encompassing the generation, scanning, and overall handling of QR codes to forestall any malicious activities, thereby fortifying the integrity of QR code functionalities.

6. Educational Initiatives on Security Best Practices:

- Integrate in-app guidance and educational materials to enlighten users on security best practices. This includes the significance of creating secure passwords and judiciously utilizing QR code functionalities to optimize security awareness.

7. Frequent Security Audits and Vulnerability Assessments:

- Conduct periodic security audits and vulnerability assessments to systematically identify and remediate potential security risks within the application, thus fortifying its overall security posture.

8. Stringent Access Controls:

- Institute stringent access controls to judiciously regulate user access, ensuring that individuals only have access to information commensurate with their designated roles, thereby minimizing the risk of unauthorized access.

9. Secure User Messaging through End-to-End Encryption:

- Institute end-to-end encryption for user messaging functionalities, reinforcing the confidentiality of communications and securing sensitive user-generated content.

10. Community Guidelines, Content Moderation, and Reporting Mechanisms:

- Establish unambiguous community guidelines to encourage positive interactions. Institute robust content moderation mechanisms to promptly address any inappropriate or harmful user-generated content. Additionally, provide users with a straightforward reporting mechanism for suspicious activities or security concerns within the application.

By meticulously incorporating these extensive safety requirements into the fabric of the application's development and operational frameworks, a robust commitment to user trust, the protection of sensitive information, and the establishment of a secure user environment is unequivocally affirmed.

##### Software Quality Attributes

The application's user interface is meticulously crafted to offer an intuitive and easily navigable experience, ensuring that users can seamlessly access and comprehend its features. This design philosophy is harmoniously balanced with a clean and professional aesthetic, prioritizing visual appeal. The commitment to design excellence extends beyond mere functionality to embody a user-centric ethos that values simplicity and efficiency.

A rigorous quality assurance process underpins the development, with a meticulous evaluation of the application's final look and service offerings. This evaluation involves a comparative analysis against other applications providing similar services, particularly in the realm of car transactions. By scrutinizing the functionality and visual appeal of competing applications, the quality assessment serves as a benchmark to ascertain the effectiveness and aesthetic appeal of the developed application.

In essence, the design philosophy revolves around creating an application that not only meets high standards of usability but also stands out in terms of visual sophistication. This commitment to excellence is evidenced in the meticulous attention to detail, ensuring that the application not only functions seamlessly but also possesses an aesthetic allure comparable to, if not surpassing, other applications in the domain of automotive services.

#### Other Requirements

A fundamental requirement of this application is for users to affix the provided QR code onto their vehicles. This singular directive ensures the seamless integration of the application's functionalities into the user's daily experience. By adhering to this straightforward requirement, users contribute to the efficacy of the application's core features, such as QR code scanning for reporting issues or engaging in community-driven interactions.

This step not only enhances the application's overall functionality but also fosters a sense of shared responsibility among users, creating a cohesive community where the adoption of the QR code becomes a pivotal aspect of leveraging the application's capabilities. The act of affixing the QR code serves as a tangible and essential contribution to the application's operational integrity, affirming the user's commitment to actively participating in the community-driven solutions facilitated by the innovative features of the automotive application.

#### Requirements Matrix

| System components →  Functional requirements | User Authentication | Rewards System | QR Code Scanning | Direct Messaging | Notifications | User Profile Management | Selling your car | Your QR Code |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Create Account | Yes | No | No | No | No | No | No | No |
| Delete Account | No | No | No | No | No | No | No | No |
| Generate unique QR code for the account | No | No | No | No | No | No | No | Yes |
| Print your QR code | No | No | No | No | No | No | No | Yes |
| Save QR code to phone gallery | No | No | No | No | No | No | No | Yes |
| Change user account settings | No | No | No | No | No | No | No | No |
| Login | Yes | No | No | No | No | No | No | No |
| Logout | No | No | No | No | No | No | No | No |
| Scan a QR code | No | Yes | Yes | No | No | No | No | No |
| Rewards tab | No | Yes | No | No | No | No | No | No |
| Message feature | No | No | Yes | Yes | No | No | No | No |
| Pick the reason for QR code scan | No | No | Yes | No | No | No | No | No |
| View notifications | No | No | No | No | Yes | No | No | No |
| Sell your car feature | No | No | No | No | No | No | Yes | No |

### 

### High-Level Design

#### System overview and design considerations

The application consists of various components and functions that work together to provide a seamless experience to the users. Here is the overview of the main components:

##### Registration and User Profiles

Users can create their own profiles on the app and provide their vehicle information, contact details and preferences. The registration process ensures that users can access personalized features. The users will also get assigned their QR code during this process, it will be unique to their account and they will have to print it and stick it on their car.

##### QR Code Scan

Users can scan a QR code through the application which will then show the options the user will have to pick as to why they scanned the code, options are as follow:

* Your vehicle is blocking me since I cannot get out of my parking space.
* I accidentally hit your vehicle.
* I cannot pass safely
* Your electric vehicle is fully charged.
* I am interested in buying your vehicle.

The app will then notify the other user through in-app notification and an SMS in case the user doesn’t have Wi-Fi coverage or a data plan.

##### Rewards System

A rewards system is planned to be added to the app in the future as a future implementation as we can’t really just get discounts and vouchers without partnerships from companies that are willing to give out such coupons. The rewards system will act as an incentive for the users to use the app more often, and to reward users that respond in a timely manner when their code is scanned.

##### Texting Feature

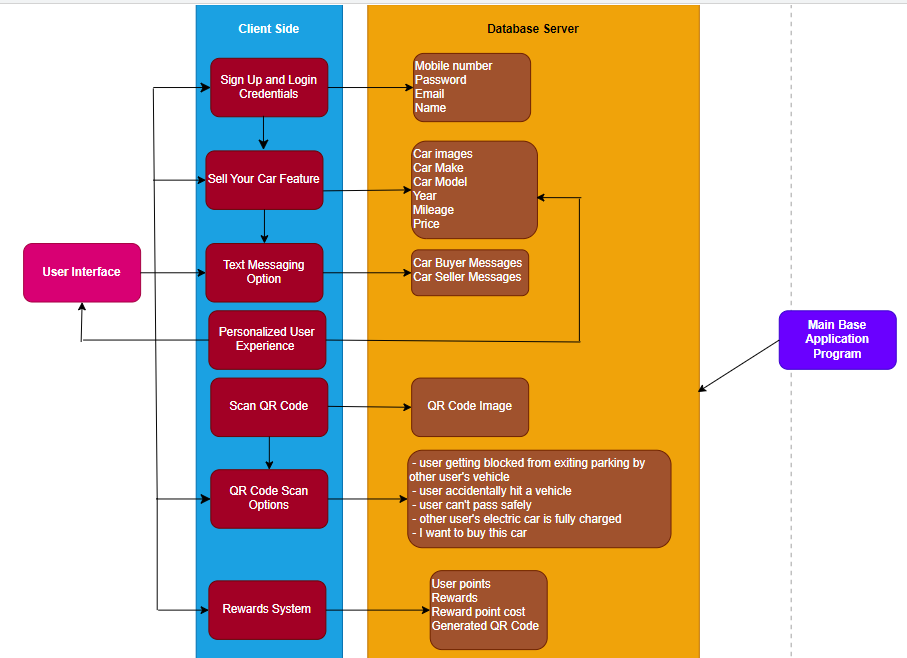
The app will also allow users that are interested in buying a car from another user to text the other user inside of the app itself but due to privacy reasons, the user with the car that has garnered the interest of the user trying to buy it will need to set the option for selling the car and being able to be contacted to “on”.

By incorporating these components and functionalities, the app provides users with a comprehensive and convenient solution for managing their vehicle-related needs, enhancing their knowledge about cars.

### 

#### System architecture and use cases

The system follows a modular program structure, dividing the responsibilities into several high-level subsystems that collaborate to achieve the desired functionality. Each subsystem has specific roles and responsibilities assigned to it, working together to provide a seamless user experience. Here is a high-level overview of the major subsystems:

****

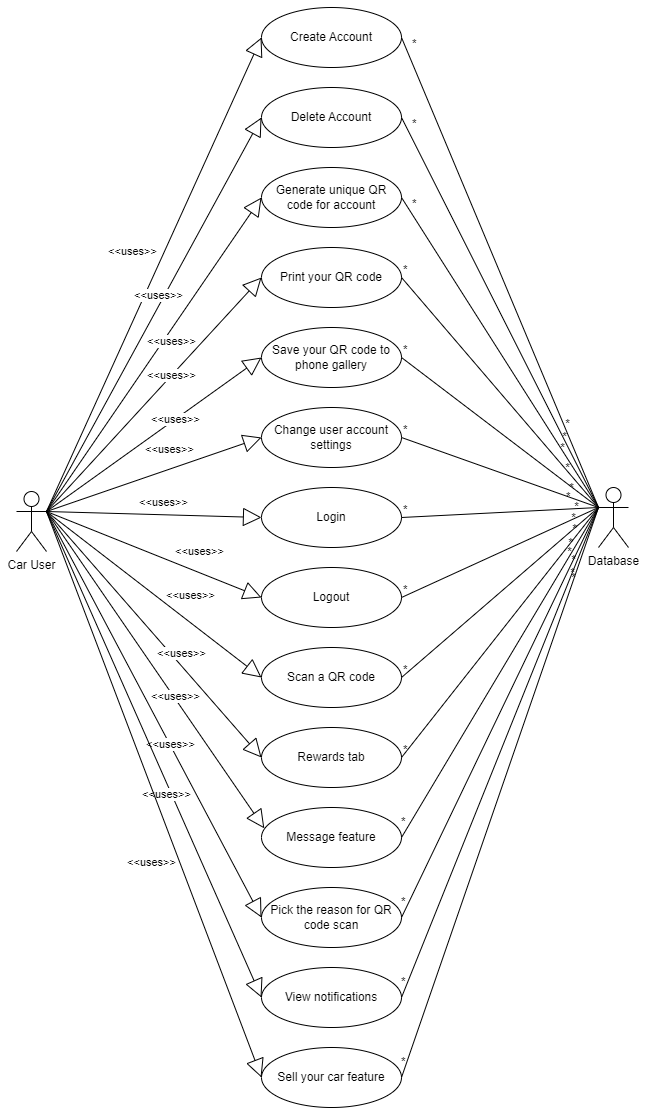
The user interface is the screen where the user interacts with the client-side functionalities. The user enters signup and login credentials, which is saved in the database server. The application's main feature is scanning the QR Code on the other user’s car to notify them on the following scenarios:

* User getting blocked from exiting parking by other user’s vehicle
* User accidentally hit a vehicle
* User can’t pass safely
* Other user’s electric car is fully charged
* I want to buy this car

The QR Code scanned by the user as well as the scenario chosen by the user will be accessed by the database server.

Other subfeatures for a personalized user experience include a sell your car feature that the user can use if they wish to sell their car. This would be done by uploading images of the car and their car details such as the make of the car, its model, year, mileage, price on the app which would be managed by the database server. Moreover, there is also a text messaging option that allows the user interested in buying the seller’s car to communicate with the seller about their car with the messages between car buyer and seller managed by the database server. Main base application program will be the database server since it is the main app component where the data is stored, managed and accessed. A rewards system will also be present that contains user points that users can use to redeem the various rewards such as free oil top up, discounts etc. The redemption process will happen based on how many user points are there and point cost of the reward. A generated QR code can then be scanned to avail the redeemed reward. The user points, rewards, rewards point cost and generated qr code will all be accessed by the database server.

The system's modular design allows for flexibility, scalability, and ease of maintenance. Each subsystem focuses on specialized responsibilities, promoting code reuse and concern separation. The collaboration of these subsystems allows the system to present users with the ease of contacting another car user of a specific scenario via scanning the QR Code attached on the other user’s car, providing the users with a platform to sell their car, messaging between car buyer and seller and a rewards system to create a strong and unified car community.

****

##### Create Account

Name: Create Account

Goal: To allow the user to create a new account

Input: User details (Mobile number, email, name, password and confirm password)

Output: An account will be created

Main Scenario: A new user is willing to use the app and use many of the functions that the app provides, which requires the user to create an account

Pre-condition: Email, mobile number and name should not be existing

Post-condition: The user should now have full access to the app

Steps:

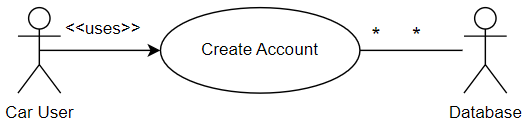
Step 1: User selects create account

Step 2: User adds in details (Mobile number, email, name, password, confirm password)

Step 3: User clicks on sign up

Exceptional Scenario: The email, mobile number and name used may already be existing

Example :



## 

##### Delete Account

Name: Delete an Account

Goal: The user should be able to delete an existing Account

Input: The user selects an account he wants to delete

Output: The user can know if the account was deleted after successful deletion

Main Scenario: The user doesn’t need the account in the application, so the user will simply select the option to remove the account he doesn’t need and the application allows this on the accounts he has logged in.

Pre-condition: The account should be existing

Post-condition: User gets a message in the app stating that their account has been successfully deleted

Steps:

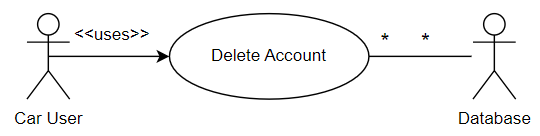
Step 1: The User selects the Account he wants to remove

Step 2: The application gives a confirmation message

Step 3: The user selects yes option, the Account is then successfully removed

Exceptional Scenario: The user doesn’t confirm the selection to remove, and the Account is not removed

Example :

****

##### Login

Name: Login

Goal: To enable the user to access the app features

Input: User can input mobile number and password

Output: User is either told that they were successful in logging into the app or login was invalid due to incorrect mobile number and/or password

Main Scenario: A user opens the car community app wanting to use its features. Before they can use the app, they must log into the app using a valid account

Pre-condition: The user must access the app with a supported smartphone and must have a valid account

Post-condition: User should be on the homepage of the app.

Steps:

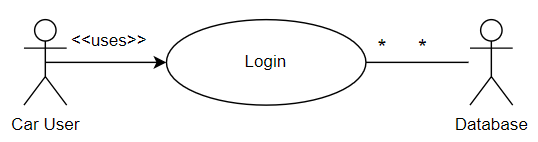
Step 1: The user is asked to sign up or login into a existing account

Step 2: The application allows new users and will alert if they already have an existing account with the same credentials

Step 3: The user is then logged into the account with the account details saved in the database

Exceptional Scenario: The user may use the forgot password option to log back to the existing account

Example :

****

##### Logout

Name: Logout

Goal: To sign out from an logged in account

Input: Select the logout button

Output: The user is logged out

Main Scenario: The user may be done using the app or is handing the phone over to someone for some task so the user may wish to log out from the account

Pre-condition: The user must be logged in

Post-condition: The user must log back in to use the application

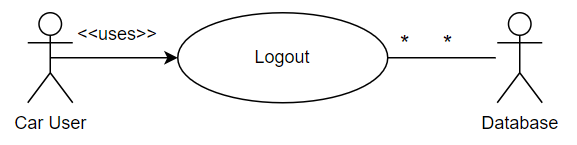
Steps:

Step 1: Go to settings

Step 2: Choose logout

Exceptional Scenario: N/A

Example :

****

## 

##### Change user account settings

Name: Change user account settings

Goal: To manage the account settings

Input: User must select the settings button

Output: The user is now able to change account settings

Main Scenario: The user may want to change the mobile number and password or edit user profile or change language to arabic or english (maybe they were entered or set incorrectly when the account was created)

Pre-condition: The user must be logged in

Post-condition: The user must now see the updated version of the account

Steps:

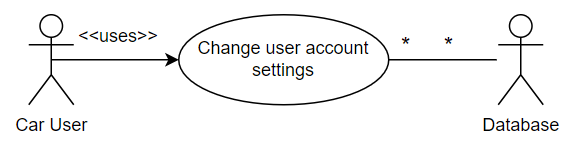
Step 1: Select Settings

Step 2: Go to the desired section that must be changed

Step 3: Change the details/options

Exceptional Scenario: N/A

Example :

****

##### Generate unique QR code for account

Name: Generate unique QR code for account

Goal: To generate a unique QR code as identification for the user’s account

Input: User must select the generate QR code button

Output: A unique QR code is generated

Main Scenario: Users can generate a unique QR code to print it and stick it on their car, save it to the phone gallery and share it with other car users.

Pre-condition: User must be logged in to their account

Post-condition: N/A

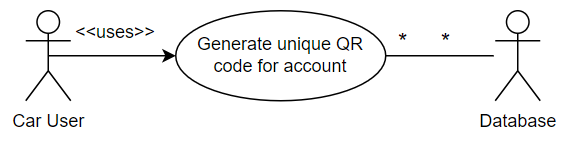
Steps:

Step 1: Log in to your account

Step 2: Select My QR code to generate QR code

Exceptional Scenario: N/A

Example:



## 

##### Print your QR code

Name: Print your QR code

Goal: To print the QR code generated by the user

Input: The user selects the print QR code button

Output: The Qr code will be printed for the user

Main Scenario: the user can print their generated QR code and stick it on their car to be notified on various scenarios when their QR code is scanned

Pre-condition: The user must generate their unique QR code

Post-condition: The user gets a message in the app stating that their QR code has been printed successfully

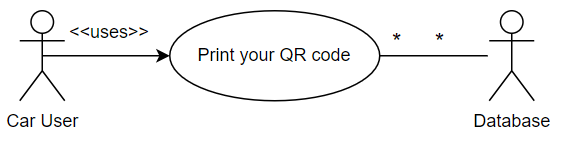
Steps:

Step 1: User must click on My QR code to generate unique QR code

Step 2: Select the print option

Exceptional Scenario: The QR code may not be printed using the app since the printer may have run out of ink or paper.

Example :

****

## 

##### Save your QR code to phone gallery

Name: Save your QR code to phone gallery

Goal: To store the QR code generated by the user in their phone gallery.

Input: The user selects the save your QR code to gallery button

Output: The user’s QR code is stored in their phone gallery

Main Scenario: The user may want to keep a copy of their QR code in their phone in case the app malfunctions and fails to display the generated QR code for the user’s account

Pre-condition: The user must have their QR code generated

Post-condition: The user gets a message stating that the QR code has been saved in the phone gallery successfully.

Steps:

Step 1: Select My QR code to generate unique QR code

Step 2: Click on Save QR code to gallery

Exceptional Scenario: N/A

Example :

****

## 

## 

##### Scan a QR code

Name: Scan a QR code

Goal: To scan the QR code attached on the user’s car to to notify the person blocking you by incorrectly parking their car / notify the person that you can’t pass safely / notify the person you accidentally hit their car / notify the person that their car is fully charged / notify the car owner that you are interested in buying their car

Input: Select Scan QR code

Output: The in-app camera feature is displayed to scan the QR code attached on the user’s car

Main Scenario: The user may want to scan the QR code on the other user’s car to notify them on a specific scenario such as I cannot pass safely, the car is fully charged and so on.

Pre-condition: User must be logged in to their account

Post-condition: User is directed to the scenarios section of the app.

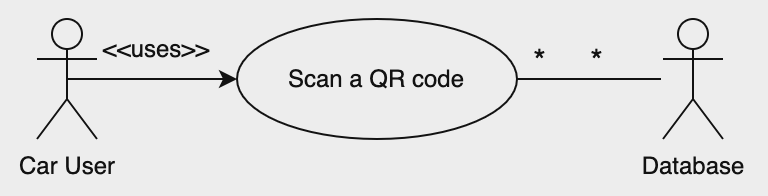
Steps:

Step 1: Select Scan QR code

Step 2: Use the in-app phone camera to scan the QR code attached on the user’s car

Exceptional Scenario: QR code was not scanned due to incorrect camera positioning and an error message is displayed.

Example :



##### Rewards tab

Name: Rewards tab

Goal: To redeem vouchers and discounts and view points

Input: Select My Rewards

Output: The rewards page is displayed

Main Scenario: The user may want to view the rewards page to see the various rewards on offer and redeem them using the points they accumulated which they can view in the rewards page as well as make use of the rewards.

Pre-condition: The user must be logged in to their account.

Post-condition: A unique QR code for the user is generated that can be scanned so that the user can make use of the specific reward

Steps:

Step 1: The user selects My Rewards

Step 2: The user selects the reward to redeem using points

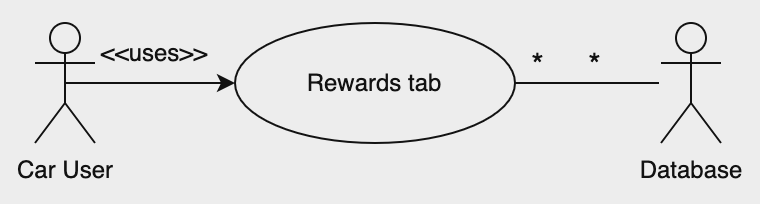
Step 3: The user gets a confirmation message

Step 4: The user selects yes option, and user redeems the reward

Step 5: A QR code is then generated by the app which the user can get scanned at a specific place to make use of the reward.

Exceptional Scenario: The user doesn’t confirm the selection to redeem the reward, and the reward is not redeemed.

Example :



##### Message feature

Name: Message feature

Goal: To allow the user interested in buying the car to communicate with its seller.

Input: The user selects messages option

Output: The messages section is displayed

Main Scenario: The user interested in buying a car may want to communicate with its seller to have more information related to the car being sold

Pre-condition: User must be logged in to the account

Post-condition: N/A

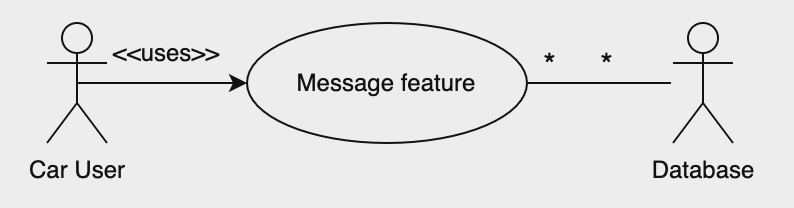
Steps:

Step 1: The User selects the messages option

Step 2: The user can now view the messages section where they can communicate with the seller of the car to know more about the car to decide whether to buy the car.

Exceptional Scenario: N/A

Example :



##### Pick the reason for QR code scan

Name: Pick the reason for QR code scan

Goal: To choose a reason after scanning QR code attached on user’s car to notify them on a specific scenario

Input: The user needs to choose a specific scenario to inform the other car user such as if they are interested in buying the other user’s car, if the car is fully charged and so on.

Output: The other user receives a notification from another user about a specific scenario

Main Scenario: The user may want to notify the other car user on a specific scenario so can pick the desired scenario to notify the other user on.

Pre-condition: User must scan the QR code attached on the other user’s car

Post-condition: The user gets a confirmation of the message the user wants to send to the other user on a certain scenario

Steps:

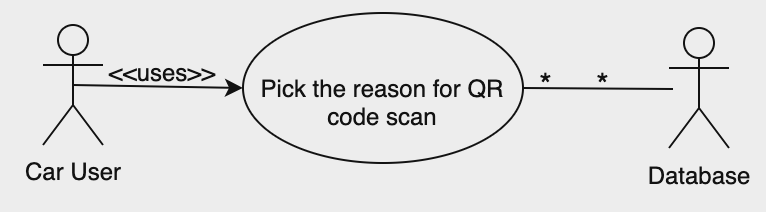
Step 1: The user scans the QR code attached to the other user’s car

Step 2: The user picks the reason for QR code scan to notify the other user

Step 3: The user receives the confirmation of the message that they wish to use to notify the other user on a specific scenario and selects send message

Exceptional Scenario: The user selects the cancel message if they decide against sending a message to the other user regarding a certain scenario.

Example :



##### View notifications

Name: View notifications

Goal: The User can view any notifications related to a specific scenario that another user has sent to them

Input: The user selects My Messages

Output: The User Can view the notifications

Main Scenario: The user may want to check any notifications they have received from other users related to a certain scenario such as if another user is interested in their car, if the car is fully charged and so on.

Pre-condition: The user must be logged in to their account

Post-condition: N/A

Steps:

Step 1: The user selects My Messages

Step 2: The user is directed to the My Messages section where they can view any notifications

Exceptional Scenario: N/A

Example :



##### Sell your car feature

Name: Sell your car feature

Goal: The User can sell their car using the sell your car feature

Input: The user selects the sell your car feature

Output: The User is directed to a section where they enter their car details and its images and upload them

Main Scenario: The user may want to sell their car

Pre-condition: The user must be logged in to their account

Post-condition: Another user must be notified if the car is up for sale based on a different color of QR code attached to the car

Steps:

Step 1: The user selects the sell your car feature

Step 2: The user is directed to a section where they can enter their car details and insert their car images

Step 3: User selects the upload button

Exceptional Scenario: N/A

Example :

****

#### Decomposition Description

Login and Sign-up

The app will have a personalized user experience where the user can sell their car. Private information such as the car’s model, make, year, mileage, price and car images will all be uploaded so for security purposes, there will be a login system.

QR Code Features

The QR code feature will help the user efficiently contact other users of the app for various reasons without the need to contact the local authorities over small matters that can be easily solved through our application.

The QR code features are divided into 5 parts:

* Scan because the user is blocked from getting out of parking by other user’s vehicle
* Scan because the user accidentally hit the other user’s vehicle.
* Scan because the user can’t pass safely.
* Scan because the other user’s electric vehicle is fully charged.
* Scan because a user is interested in buying the other user’s car.

Scan Timeout: Once a user’s QR code has been scanned, there will be restrictions put in place that will block spam scanning the same QR code to prevent harassment and abuse of the application’s features. A timer of around 5-10 mins should be suitable enough for a user to get back to their vehicle.

Sell / Buying Vehicles

This feature will allow the user to text another user directly to buy their car, only if certain conditions are met which are that the “seller” should have the car as up for sale and allow the option to be directly contacted by other users within the app.

Rewards Feature

This feature allows users to redeem various rewards such as free oil top up, discounts etc based on user points and cost of the reward. A unique qr code will then be generated to avail the redeemed reward.

#### Design Rationale

The rationale behind selecting the architecture above was to have a modular program structure where the responsibilities are divided into several high-level subsystems that collaborate to achieve the desired functionality. Moreover, this architecture shows that each subsystem has specific roles and responsibilities assigned to it that work together to provide a seamless user experience. The architecture’s modular design enables flexibility, scalability and ease of maintenance. Each subsystem in the architecture focuses on specialized responsibilities, promoting code reuse and concern separation. The architecture shows the user interface which is the screen where the user interacts with the client-side functionalities. The database server is clearly depicted in the architecture showing all the data stored in it and arrows have also been used to clearly show the interconnections between various parts of the architecture. The color scheme makes the architecture design visually appealing as well as making the user clearly understand how the architecture looks.

Critical issues that were considered for this architecture include:

* Dependency Management- The interactions between the subsystems, like the user interface, scanning the QR code and text messaging option, may have complex dependencies. Making sure that these subsystems communicate effectively without creating tight coupling or excessive dependencies could be vital.
* Scalability- While the modular design enables flexibility, ensuring scalability can be difficult. As the user base grows, mainly with features like car selling and messaging, the system’s ability to tackle increased traffic and data volume needs consideration.
* Performance- The real-time nature of scenarios like user notifications or choosing the scenarios based on QR code scanning requires efficient and responsive subsystems. Balancing responsiveness with the number of subsystems and the interactions is vital for a seamless user experience.
* Security- Handling user data, mainly sensitive information like login credentials, messages between car buyer and seller, and car details, needs robust security measures. Each subsystem’s interfaces and data handling processes need to be fully secured to avoid data breaches or unauthorized access.
* Maintenance and Updates- As the system evolves, maintaining and updating various subsystems without hampering the entire system is a challenge. Making sure that changes in one subsystem does not negatively affect others while providing new features or improvements needs careful planning
* User Experience and Usability- While the modular design aims for specialized responsibilities, ensuring a cohesive and user-friendly experience across all subsystems is vital. Users interacting with various features like QR code scanning, car selling, or text messaging should face consistency and intuitive usability.

Trade/offs that were considered for this architecture include:

* Flexibility vs. Complexity- Adding new functionalities may improve flexibility but could also increase system complexity, affecting maintainability.
* Scalability vs. Performance- Scaling the system for more users may impact performance. Balancing scalability while maintaining a responsive system is vital.
* Security vs. Usability- Implementing stringent security measures may sometimes affect user convenience or ease of use.
* Centralized Database Server- Having a centralized database server simplifies data management but could introduce a single point of failure.
* User points and Rewards System Performance- Maintaining a rewards system may introduce performance considerations, especially as the user base grows.
* Integration complexity- Integrating features like QR code scanning, messaging and the rewards system may bring integration complexities.

#### Data Description

Users Table

| **ID** | **int** | ***Not null*** |
| --- | --- | --- |
| **EntryDate** | **datetime** | ***Not null*** |
| **Name** | **nvarchar(80)** | ***null*** |
| **Email** | **nvarchar(100)** | ***null*** |
| **MobileNo** | **nvarchar(15)** | ***null*** |
| **Password** | **nvarchar(20)** | ***null*** |
| **User\_GUID** | **Uniqueidentifier** | ***null*** |

Notifications Table

| **ID** | **int** | ***Not null*** |
| --- | --- | --- |
| **User\_GUID** | **Uniqueidentifier** | ***Not null*** |
| **Description** | **nvarchar(max)** | ***null*** |
| **EntryDate** | **Datetime** | ***null*** |
| **Status** | **Int** | ***null*** |

Notes:

The system's unique identifiers, such as ID and EntryDate, are automatically generated. Additionally, the User\_GUID is automatically created and serves as a unique identifier, crucial for the functioning of the QR code. This unique identifier is utilized to retrieve information from the database, allowing the system to identify the specific user and deliver notifications accordingly.

When a QR code is scanned, it provides the GUID of the user, which is then stored in the notification table with a status of 1. This status signifies that the user has not yet received or read the message. Subsequently, when a notification is sent and the intended user receives it, the built-in spGetNotification function in the MSSQL Database examines the User\_GUID to locate the corresponding entry and updates the status to 0. This change confirms that the notification has been successfully received by the targeted user, completing the notification delivery process.

#### 

#### 

#### 

#### Data Dictionary

The table below will give an idea about the type of data that will be stored in the database.

|  | ***Field*** | ***Data Type*** | ***Null*** | ***Default*** |
| --- | --- | --- | --- | --- |
| ***user*** | **Email** | NVARCHAR(100) | Yes | *NULL* |
| **ID** | INT | No |  |
| **MobileNo** | NVARCHAR(15) | Yes | *NULL* |
| **Name** | NVARCHAR(80) | Yes | *NULL* |
| **password** | NVARCHAR(20) | Yes | *NULL* |
| ***Users*** | **EntryDate** | DATETIME | No |  |

|  | ***Field*** | ***Data Type*** | ***Null*** | ***Default*** |
| --- | --- | --- | --- | --- |
| ***user*** | **Email** | NVARCHAR(100) | Yes | *NULL* |
| **ID** | INT | No |  |
| **MobileNo** | NVARCHAR(15) | Yes | *NULL* |
| **Name** | NVARCHAR(80) | Yes | *NULL* |
| **password** | NVARCHAR(20) | Yes | *NULL* |
| ***Notifications*** | **EntryDate** | DATETIME | No |  |

#### Component Design

User Authentication

* Mobile number and password login
* Sign-up with OTP confirmation

Rewards System

* Display user points
* Show available rewards
* Redeem rewards with QR codes

QR Code Scanning

* Use camera for QR code scanning
* Report issues with predefined options

Direct Messaging

* Communicate directly with other users
* Specific messaging options for buying/selling cars

Notifications

* Display all previous notifications with timestamps

User Profile Management

* Edit name and profile picture
* Update mobile number with OTP confirmation
* Change password

Selling Your Car

* Upload car image
* Enter car details (make, model, mileage, year, engine, price)
* Update QR code with a notification message

Your QR Code

* Display QR code along with a unique number
* Save to gallery or print options

#### 

#### Design Components (Detailed)

Sign Up Flow

* User Authentication:

User enters mobile number and password.

If the user doesn't have an account, they can select the "Sign Up" option.

* Sign Up Page:

User enters mobile number, name, password, and confirms the password.

Clicking on "Sign Up" triggers OTP confirmation.

* OTP Confirmation:

User receives OTP on the registered mobile number.

User enters OTP to confirm the sign-up process.

Main Page Navigation

* Main Page:

Logo and main options (Rewards, Scan QR Code, Notifications).

Hamburger icon for side menu.

* Side Menu:

Edit Your Profile

Update Your Mobile Number

Change Your Password

Sell Your Car

Your QR Code

Messages

About AutoMate

About Us

Rewards Flow

* Rewards Page:

Displays user points and available rewards.

User can select a reward (e.g., AED40 Fuel Discount).

* Redeem Rewards:

Displays points cost for the selected reward.

User can redeem the reward, and a QR code is generated.

QR Code Scanning Flow

* Scan QR Code Page:

User uses the camera to scan a QR code.

Options for issue reporting (e.g., cannot get out of parking) are displayed.

* Confirm Page:

Displays the chosen case and a message to be sent.

User can choose to send or cancel the message.

Direct Messaging Flow

* Direct Messaging:

User selects the option "I want to buy this car."

Confirm page is displayed with a predefined message.

User can send the message or opt for direct messaging.

Notifications Flow

* Notifications Page:

Displays all previous notifications with timestamps.

User Profile Management Flow

* Edit Your Profile:

User can change their name and profile picture.

* Update Mobile Number:

User enters a new mobile number and OTP for confirmation.

* Change Your Password:

User enters the current password, sets a new password, and confirms.

Selling Your Car Flow

* Sell Your Car Page:

User uploads a car image and enters car details.

After submission, a notification is displayed, indicating the QR code has been updated.

* Your QR Code Page:

Displays the updated QR code.

Users can print or save the QR code to the phone gallery.

#### User Interfaces

The user interfaces (UI) within our application are meticulously designed to offer a seamless and intuitive experience, reflecting a user-centric approach. Each element of the UI is thoughtfully crafted to ensure accessibility, ease of navigation, and clarity. The design is characterized by a clean and modern aesthetic, fostering a visually engaging environment for users. Intuitive navigation menus and well-defined visual hierarchies guide users through various features, promoting a sense of familiarity and ease in interaction. The UI elements are strategically arranged to optimize user flow, emphasizing the essential functionalities related to QR code scanning, messaging, car selling, and other core features. Responsiveness and consistency across different screens and devices are prioritized, ensuring a uniform and enjoyable experience for users. The application's user interfaces are not merely functional; they encapsulate a design philosophy that prioritizes user understanding, engagement, and a visually appealing experience throughout their interaction with the application.

#### Hardware Interfaces

The application seamlessly integrates with a variety of hardware interfaces to enhance its operational capabilities. The primary hardware interface revolves around the users' smartphones, serving as the central hub for interaction. A modern smartphone with a built-in camera becomes an essential component, enabling users to effectively scan QR codes attached to vehicles for various functionalities within the application. This interaction between the smartphone's camera and the QR code exemplifies the harmonious synergy between hardware and software.

The hardware interface extends beyond smartphones to encompass any additional components or devices related to electric vehicles, ensuring a comprehensive and inclusive approach to technological integration. The seamless interplay between the application and various hardware interfaces forms the backbone of its functionality, providing users with a versatile and user-friendly experience across different devices.

#### 

#### Software Interfaces

The software interfaces within this application play a pivotal role in orchestrating seamless interactions between different components, ensuring a harmonious user experience. The user interacts with the application through a well-designed graphical user interface (GUI), which serves as the visual gateway to various functionalities.

Additionally, the application integrates with external services, such as Firebase, to facilitate real-time data synchronization, user authentication, and cloud-based storage. The software interfaces are carefully crafted to not only enhance the user's interaction with the application but also to establish robust connections with backend services, creating a cohesive and responsive ecosystem. This emphasis on seamless software interfaces is fundamental to the application's ability to deliver a reliable, user-friendly, and feature-rich experience to its diverse user base.

#### 

#### Communication Interfaces

The application necessitates an internet connection to access cloud-based data seamlessly. However, for enhanced user convenience and ease of use, we aspire to minimize the dependency on internet connectivity for specific features. Notably, we aim to eliminate the requirement for an internet connection to receive critical notifications, such as alerts for blocking or incorrect parking. By implementing a system that delivers these notifications via SMS, users can seamlessly receive alerts even in scenarios where a data package or network coverage may be unavailable. This strategic enhancement ensures that users remain informed and engaged with the application's essential functionalities, fostering a more inclusive and reliable user experience, particularly in situations where internet connectivity might pose limitations.

### Low-Level Design

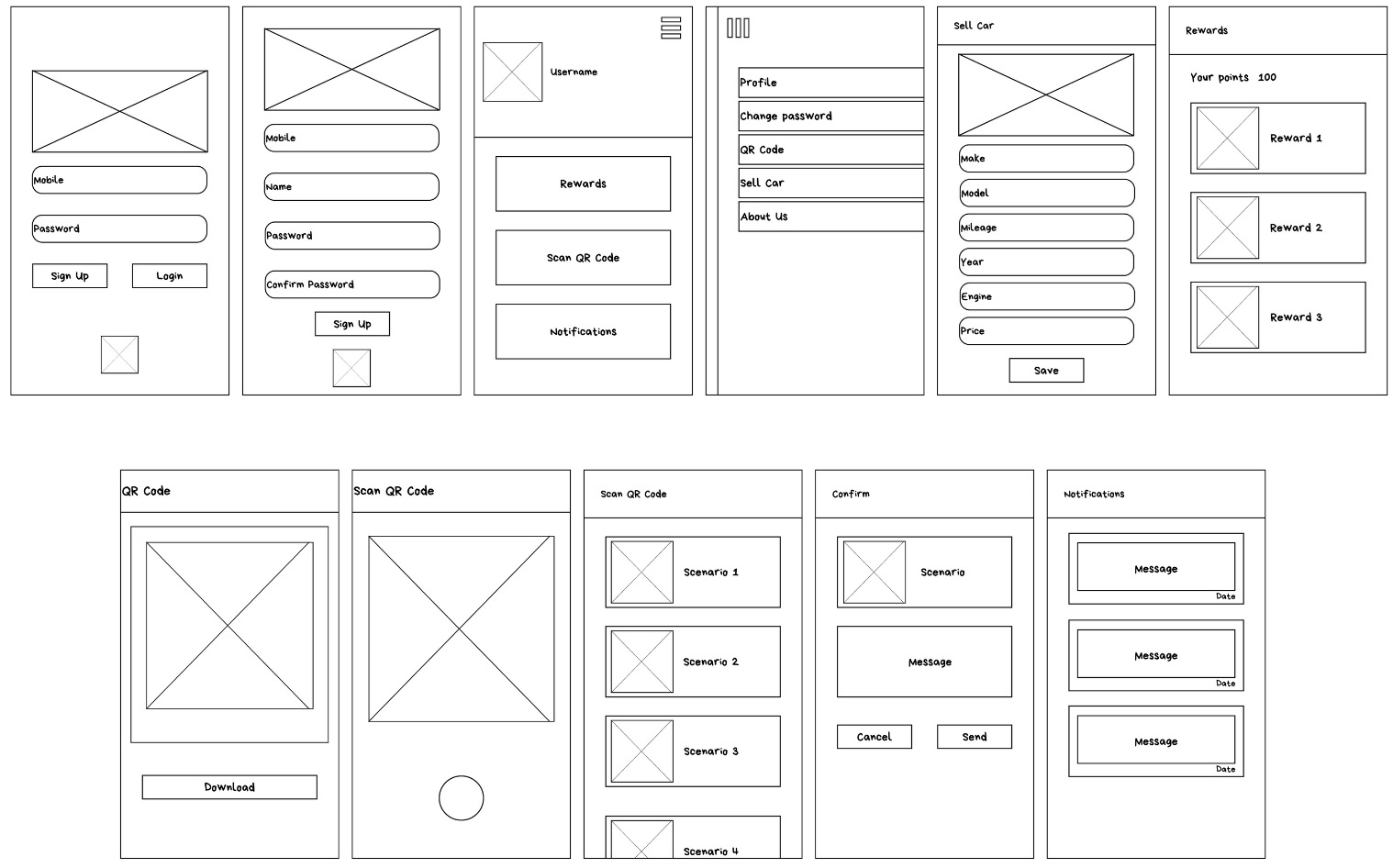
#### Overview of User Interface

The AutoMate Application, designed to foster a car community, provides users with a seamless experience in addressing issues related to improperly parked or obstructing vehicles. Upon downloading the application, users gain the ability to effortlessly notify others using a QR code. The user-friendly interface ensures a straightforward process. To leverage this feature, users are prompted to sign up for a new account, or log in if they already have one, to access all application features.

Navigating the app is made easy as users can generate their unique QR code by simply clicking on "My QR Code." This code serves as a personalized identifier for the user. Users can further enhance communication by scanning the QR codes of other community members, facilitating the exchange of information regarding parking issues and more. The application's intuitive design enhances user interaction, providing a convenient and efficient platform for addressing car-related concerns within the community.

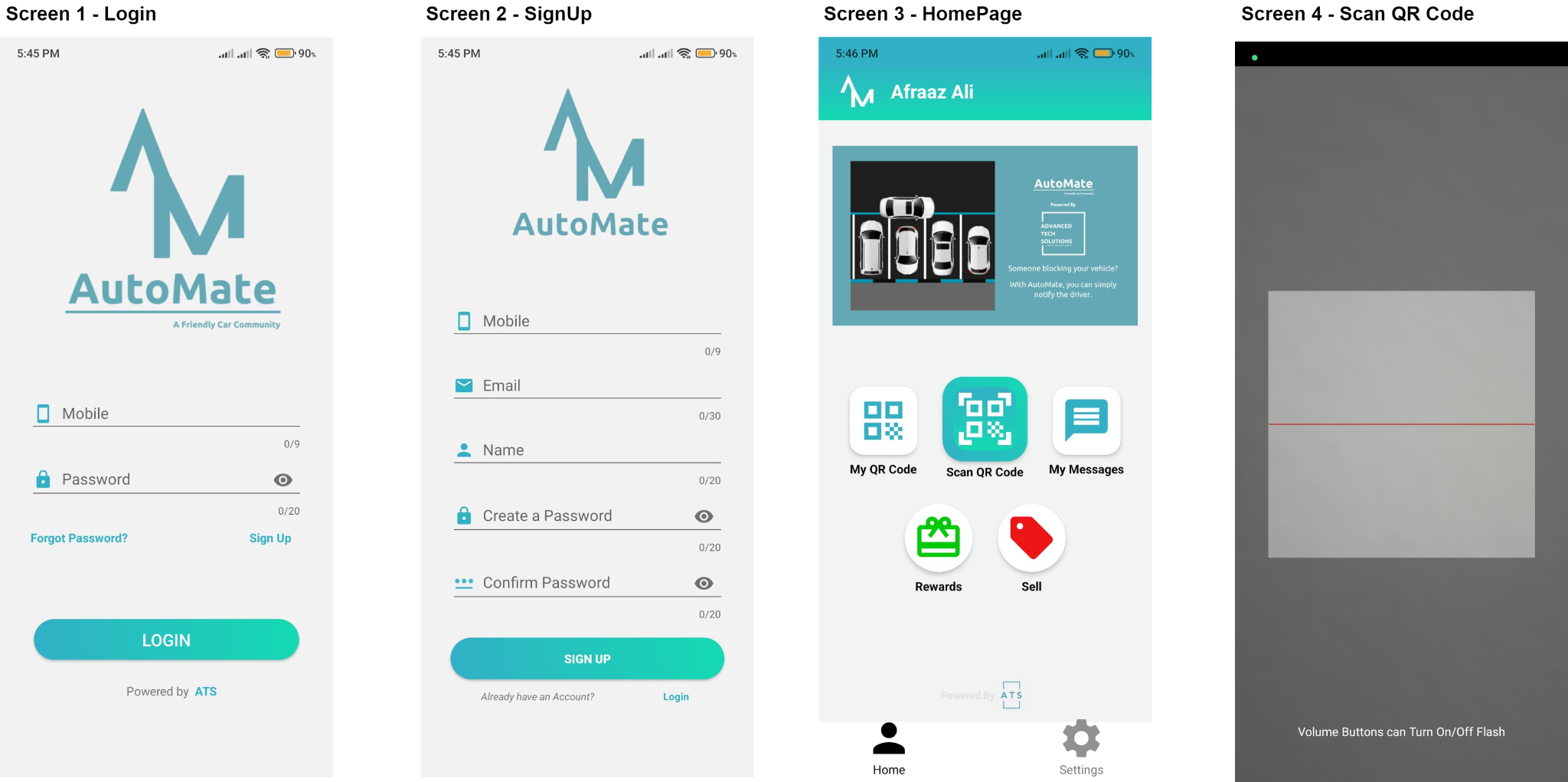
#### Screen Images

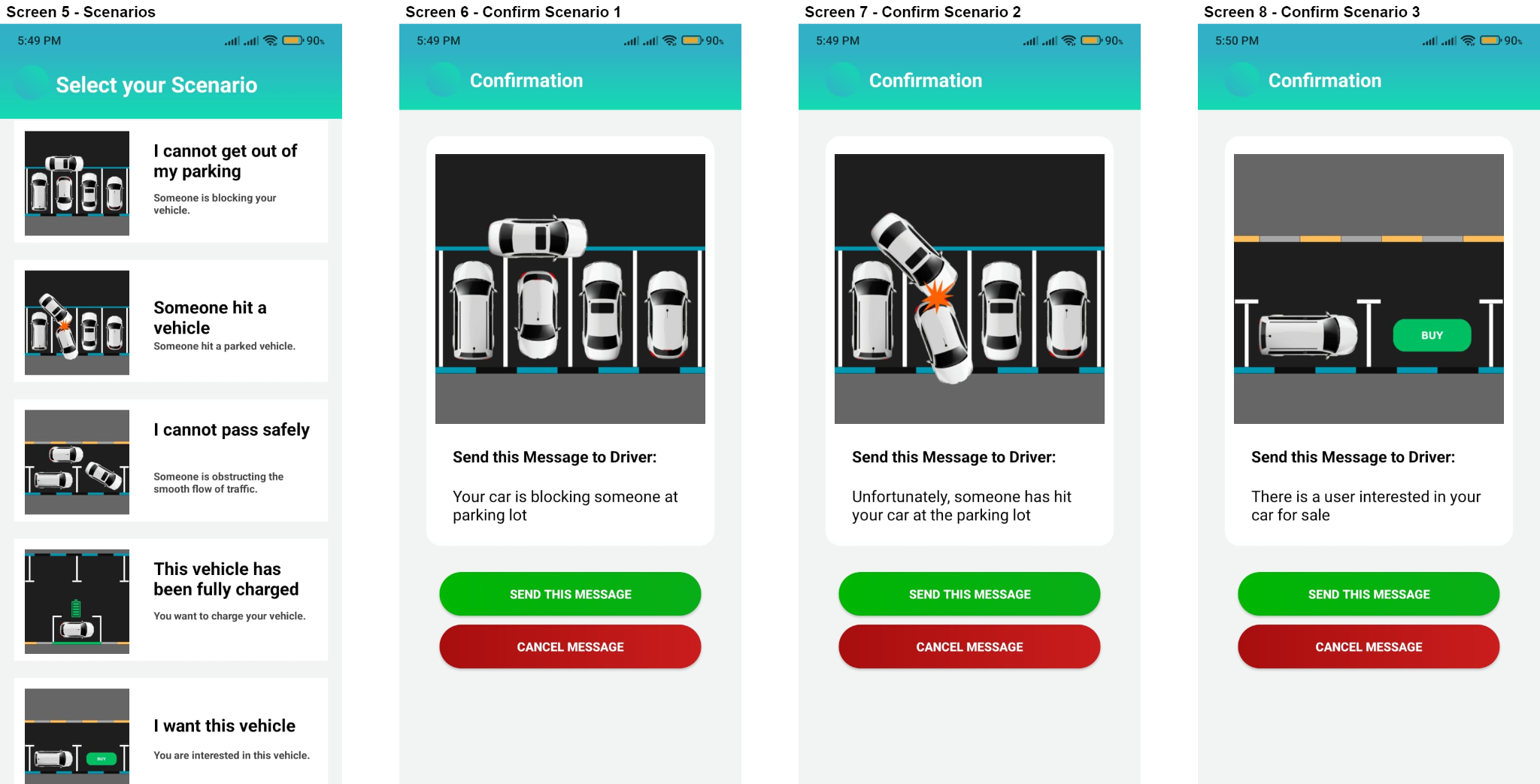
##### Low-Fidelity Hand-drawn Sketch

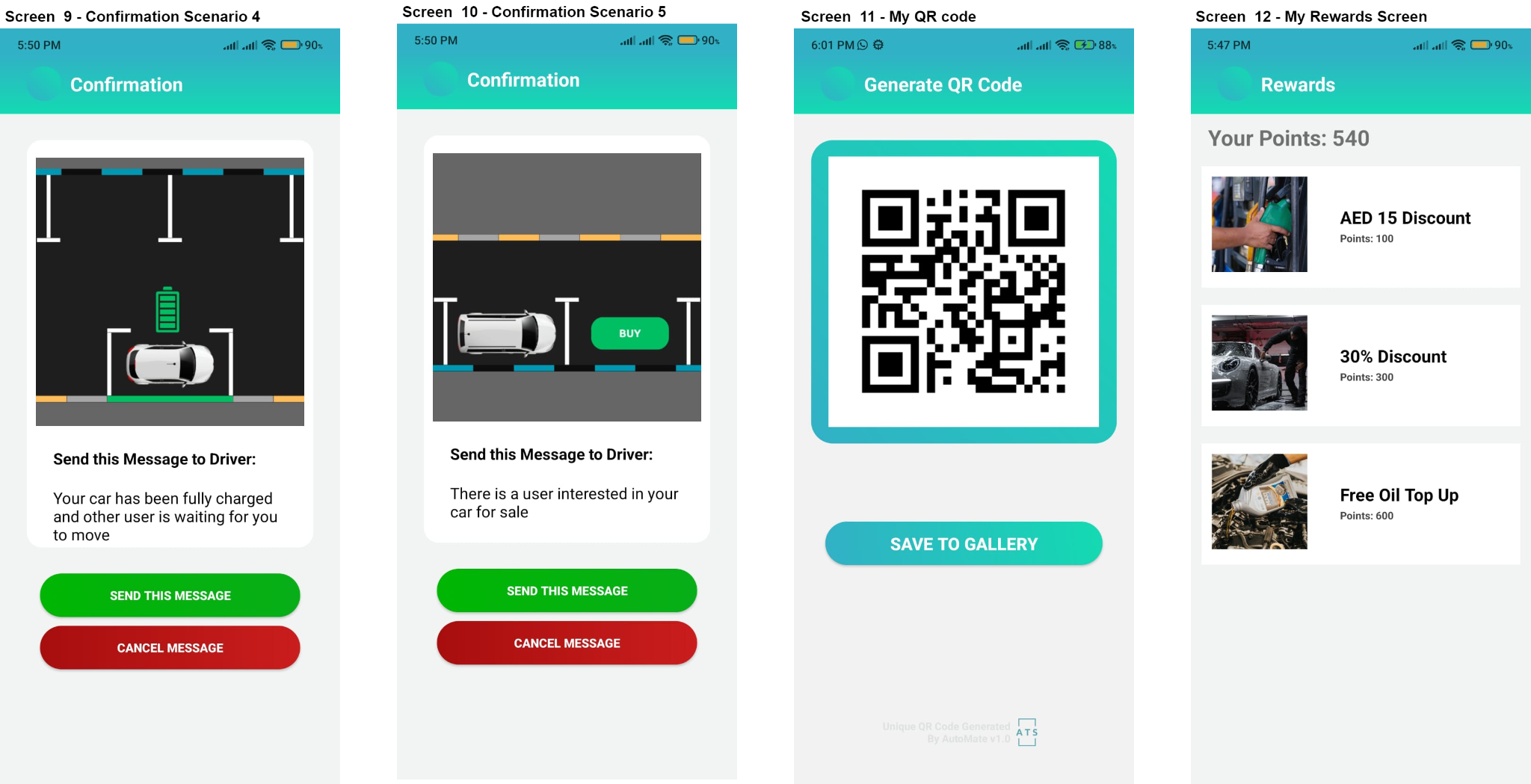


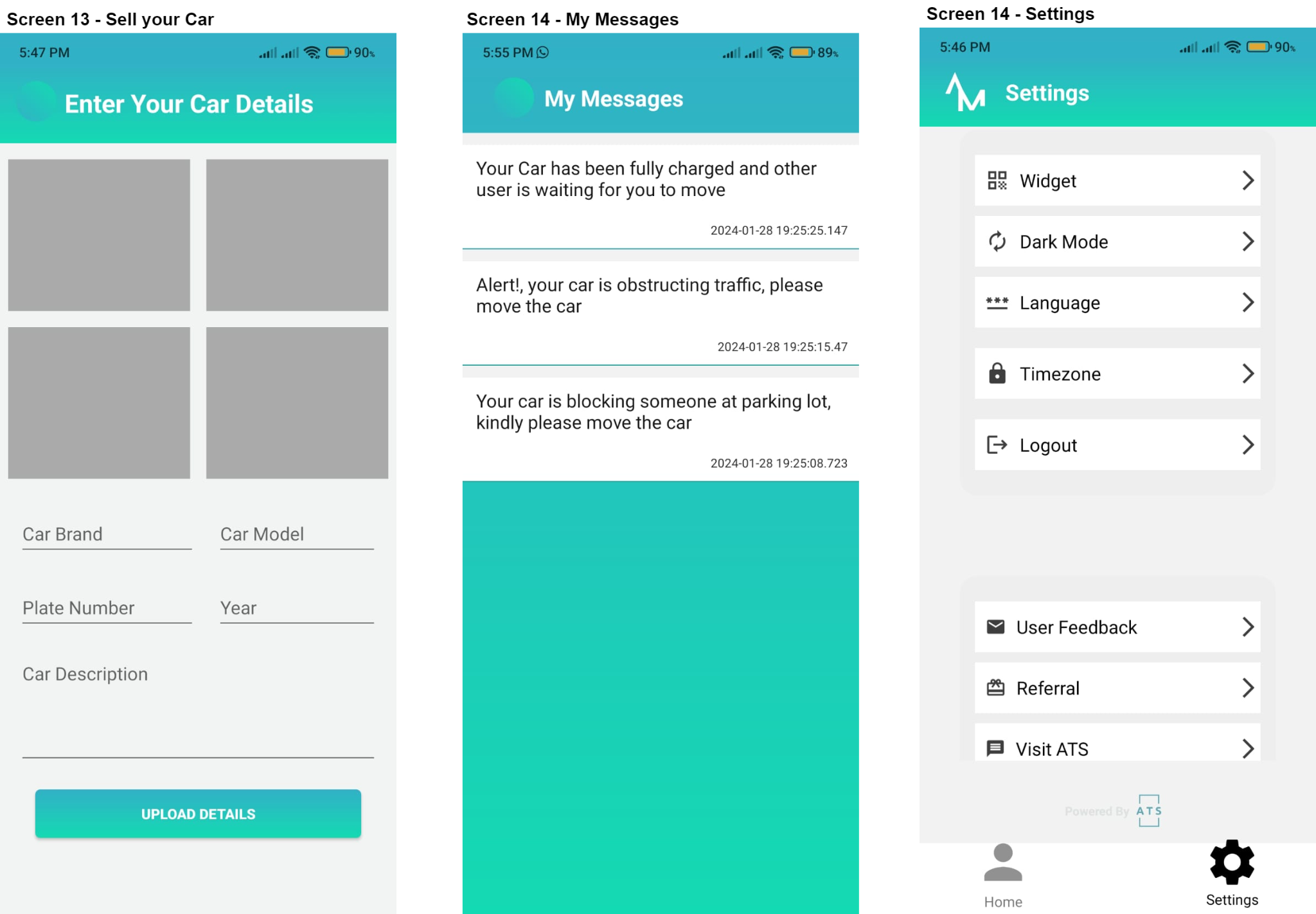
##### Low Fidelity Software Designed Screens

##### High Fidelity Software Created Screens









##### Screen Image Description Table

| Screen 1 | Login Page | Here the user is expected to enter his/her Phone Number and Password, a forget password option is provided considering human errors, A sign-up option is presented below the password field. The Powered By ATS is a link to the official website of Advance Tech Solutions. |
| --- | --- | --- |
| Screen 2 | Sign-up Page | Here the user is expected to enter his/her new credentials for a new account, Mobile number, name, email, and password, A login option is presented below, and after the sign up is successful the user is taken to the login Screen |
| Screen 3 | Home Page | This is the Home Page where the user’s name is on the Top bar, The User is provided with 6 neatly placed buttons.  Each of them have their own Functionality. The Newsletter screen is also present above the buttons to show the most recent news and updates from the ATS company. |
| Screen 4 | Scan QR code | This Scan QR Button takes you to Scanning by opening the camera directly. The user may use the volume buttons to turn on and off the flash lights. Once the User Scans, he is taken to the Scenario Screen |
| Screen 5 | Scenarios | The User is presented with 5 options of each having their own unique message. The user can Select any one of them to send the other user the issue related to his vehicle. |
| Screen 6 | Confirm  Scenario 1 | This Screen Confirms with the User that he is about to send a Notification that contains the description as ‘Your car is blocking someone at the parking lot.’ |
| Screen 7 | Confirm  Scenario 2 | This Screen Confirms with the User that he is about to send a Notification that contains the description as ‘Unfortunately, Someone has hit your car at the parking lot.’ |
| Screen 8 | Confirm  Scenario 3 | This Screen Confirms with the User that he is about to send a Notification that contains the description as ‘Warning, your car is obstructing traffic, please move the car’ |
| Screen 9 | Confirm  Scenario 4 | This Screen Confirms with the User that he is about to send a Notification that contains the description as ‘Your Car has been fully charged and another user is waiting for you to move.’ |
| Screen 10 | Confirm  Scenario 5 | This Screen Confirms with the User that he is about to send a Notification that contains the description as ‘There is a User interested in your car for sale.’ |
| Screen 11 | My QR code | Here the user can generate his own unique special QR code where he has the option to print or save to your gallery for printing it later and sticking it to your vehicle |
| Screen 12 | My Rewards Screen | In this Screen the user can get promo codes for fuel-top, car wash and other types of promo codes using a reward system that is yet to be implemented. |
| Screen 13 | Sell my Car Screen | This is for a future implementation where the user can insert his details on the car and its images and can put it for sale, there should be a hub of different cars up for sale from this screen. |
| Screen 14 | My Messages | All the received Notifications will be displayed here on this screen for the user to see his previous issues and if a user is interested in their car. |
| Screen 15 | My Settings | This page allows the user to log out and change preferences according to personal use. About us is also provided for the user to view our Official website |

### Testing

#### Testing goals

**1. Functionality Testing:**

Ensure that all core functionalities of the application, including QR code scanning, messaging, car selling, and notifications, operate as intended across various scenarios and devices.

**2. Compatibility Testing:**

Verify the application's compatibility across different Android versions (e.g., Android Oreo and above) ensuring consistent performance and user experience.

**3. Usability Testing:**

Assess the application's user interface (UI) for intuitiveness, accessibility, and overall user experience. Gather feedback to identify areas for improvement in terms of navigation, layout, and user interaction.

**4. Performance Testing:**

Evaluate the application's performance under different conditions, including varying network speeds and device specifications. Measure factors such as loading times, responsiveness, and resource utilization to ensure optimal performance.

**5. Security Testing:**

Conduct comprehensive security assessments to identify and mitigate potential vulnerabilities, ensuring the protection of user data, authentication mechanisms, and communication channels.

**6. Stability Testing:**

Validate the application's stability and reliability by subjecting it to prolonged usage and stress testing. Identify and address any potential crashes, freezes, or performance degradation over extended periods.

**7. Localization Testing:**

Verify the accuracy and effectiveness of language translations and localization efforts, ensuring that the application is accessible and culturally appropriate for users worldwide.

**8. Regression Testing:**

Perform regression testing to confirm that recent code changes or updates have not introduced new defects or affected existing functionalities adversely.

#### Test plan scope

The test plan encompasses the following key areas:

**1. Feature Testing:**

* QR code scanning
* Qr Code Generation
* Notifying Users
* My Messages

**2. Compatibility Testing:**

* Android devices (versions 10 and above)

**3. Usability Testing:**

* User interface (UI) design
* Navigation and user interaction

**4. Performance Testing:**

* Loading times
* Responsiveness
* Resource utilization

**5. Security Testing:**

* Data encryption
* Authentication mechanisms
* Secure communication protocols

**6. Stability Testing:**

* Prolonged usage
* Stress testing

**7. Regression Testing:**

* Validation of recent code changes
* Impact assessment on existing functionalities

#### Test forms and test results

**Function 1**

| **Requirement name:** | Login |
| --- | --- |
| **Test case description 1 :** | Verify that a user can successfully log in with valid credentials. User needs to enter his Phone number and his password. |
| **Pre-requirements:** | The User has to have his account signed up before hand |
| **Input:** | Username as Mobile Number and Password of the User  Mobile: 561016265  Password: Testing@123 |
| **Expected output:** | The Mobile Number and Password matches and the User is Logged In to the home screen, The User GUID is passed to the Homepage for the QR code generation. |
| **Observed output:** | The Login button after its pressed takes the User to the HomePage |
| **Verdict:** | Pass: The acceptance test has been successful, and the system behaves as expected. It meets the defined criteria, and there are no critical issues |
| **Comments:** | The Login Page is designed specifically for the user to enter his details so that the application recognizes his account and  gives him his personalized experienced |

| **Test case description 2 :** | Verify that an error is displayed when trying to log in with an incorrect password. |
| --- | --- |
| **Pre-requirements:** | The User has to have his account signed up before hand |
| **Input:** | Username as Mobile Number and Password of the User  Mobile: 561016265  Password: Testing@345 (Incorrect Password) |
| **Expected output:** | The Mobile Number and Password does not match and an error message is displayed, indicating that the password or Mobile Number is incorrect. |
| **Observed output:** | The error message for an incorrect password or Mobile Number is displayed. |
| **Verdict:** | Pass |
| **Comments:** | The system correctly detected the incorrect password. |

| **Test case description 3 :** | Verify that an error is displayed when trying to log in with an unregistered Mobile No. |
| --- | --- |
| **Pre-requirements:** | The User has to have his account signed up before hand |
| **Input:** | Username as Mobile Number and Password of the User  Mobile: 561016266 (Wrong number)  Password: Testing@123 |
| **Expected output:** | The Mobile Number and Password does not match and an error message is displayed, indicating that the password or Mobile Number is incorrect. |
| **Observed output:** | The error message for an incorrect password or Mobile Number is displayed. |
| **Verdict:** | Pass |
| **Comments:** | The system correctly detected the incorrect Mobile Number. |

**Function 2**

| **Requirement name:** | Sign Up |
| --- | --- |
| **Test case description 1 :** | Verify that a user can successfully register for a new account. |
| **Pre-requirements:** | The user is on the signup page. |
| **Input:** | Mobile : 509999999  Full Name: John Doe  Email: john.doe@example.com  Password: Test@123  Confirm Password: Test@123 |
| **Expected output:** | The user is registered successfully, and a confirmation message is displayed. After the message is displayed the User is Automatically taken back to the Login Page |
| **Observed output:** | The registration confirmation message is displayed and the Login Page is Displayed after it. |
| **Verdict:** | Pass |
| **Comments:** | The system correctly registered the user. |

| **Test case description 2 :** | Verify that an error is displayed when trying to register with an existing email. |
| --- | --- |
| **Pre-requirements:** | The user is on the signup page. |
| **Input:** | Mobile: 501016265  Full Name: Jane Smith  Email: john.doe@example.com (an existing email)  Password: Test@456  Confirm Password: Test@456 |
| **Expected output:** | An error message is displayed, indicating that the email is already in use. |
| **Observed output:** | The error message for existing Mobile No. is displayed. |
| **Verdict:** | Pass |
| **Comments:** | The system correctly detected the existing Mobile No. |

| **Test case description 3 :** | Verify that password and confirm the password must match during registration. |
| --- | --- |
| **Pre-requirements:** | The User is on the SignUp Page |
| **Input:** | Mobile: 561016275  Full Name: Mary Johnson  Email: mary.johnson@example.com  Password: Test@789  Confirm Password: Test@123 (mismatch) |
| **Expected output:** | An error message is displayed, indicating that the passwords do not match. |
| **Observed output:** | The error message for password mismatch is displayed. |
| **Verdict:** | Pass |
| **Comments:** | The system correctly detected the password mismatch. |

| **Test case description 4 :** | Verify that the password must meet the minimum complexity requirements. |
| --- | --- |
| **Pre-requirements:** | The User is on the SignUp Page |
| **Input:** | Mobile: 561017275  Full Name: Robert Brown  Email: robert.brown@example.com  Password: test (below complexity requirements)  Confirm Password: test (below complexity requirements) |
| **Expected output:** | An error message is displayed, indicating that the password does not meet complexity requirements. Minimum 6 characters and a special Character |
| **Observed output:** | The error message for password complexity is displayed. |
| **Verdict:** | Pass |
| **Comments:** | The system correctly detected the password complexity issue. |

**Function 3**

| **Requirement name:** | Scan QR Code |
| --- | --- |
| **Test case description 1 :** | Verify that the user can successfully scan a valid QR code and get its Contents. |
| **Pre-requirements:** | The User needs to be on the Scan QR screen and his Phone needs to have a camera capable of reading QR codes |
| **Input:** | The user uses the device's camera to scan a valid QR code. |
| **Expected output:** | The application recognizes the QR code and retrieves information associated with it to use it for the Scenario page Screen which opens up |
| **Observed output:** | The application successfully recognizes the QR code and uses its contents for the Scenario Purpose |
| **Verdict:** | Pass |
| **Comments:** | The system correctly processes a valid QR code. |

| **Test case description 2 :** | Verify that an error is displayed when attempting to scan an invalid QR code. |
| --- | --- |
| **Pre-requirements:** | The User scans a QR code using his Camera from Scan QR code Screen |
| **Input:** | The user attempts to scan a QR code that is not associated with any valid information. |
| **Expected output:** | An error message is displayed after the user selects the scenario he wants to send to another User, indicating that the scanned QR code is invalid. |
| **Observed output:** | An error message is displayed for an invalid QR code. |
| **Verdict:** | Pass |
| **Comments:** | The System correctly detects an invalid QR code that wasn’t generated from the Application itself |

| **Test case description 3 :** | Verify that the system prompts for permission to access the device's camera. |
| --- | --- |
| **Pre-requirements:** | The User is on the Scan QR code Screen |
| **Input:** | The system prompts the user to grant camera permissions. |
| **Expected output:** | The system prompts the user to grant camera permissions. |
| **Observed output:** | A prompt requesting camera permissions is displayed. |
| **Verdict:** | Pass |
| **Comments:** | The system correctly requests necessary permissions for QR code scanning. |

| **Test case description 4 :** | Verify that the user receives feedback upon a successful scan. |
| --- | --- |
| **Pre-requirements:** | The user is on the Scan QR Code screen, ready to Scan a Valid QR code |
| **Input:** | The User Scans the Valid QR code |
| **Expected output:** | Auditory feedback is provided, indicating a successful scan. |
| **Observed output:** | The system provides audio feedback upon successful QR code recognition. |
| **Verdict:** | Pass |
| **Comments:** | The system provides appropriate feedback for a successful scan. |

**Function 4**

| **Requirement name:** | Generate User’s Unique QR code |
| --- | --- |
| **Test case description 1 :** | Verify that the user can generate their QR code. |
| **Pre-requirements:** | The user is logged into the application. |
| **Input:** | The user navigates to the "My QR Code" section. |
| **Expected output:** | The application generates a unique QR code associated with the user's account. |
| **Observed output:** | A unique QR code is displayed on the user's screen |
| **Verdict:** | Pass |
| **Comments:** | The system successfully generates a QR code for the user. |

| **Test case description 2 :** | Verify that the generated QR code is scannable by another user. |
| --- | --- |
| **Pre-requirements:** | Two users are using the application, and one user has generated a QR code. |
| **Input:** | Another user scans the generated QR code. |
| **Expected output:** | The scanning user receives information related to the user who generated the QR code to send a personalized Notification |
| **Observed output:** | The scanning user successfully retrieves information from the scanned QR code and the User is able to send a Notification to the User whose QR code was scanned |
| **Verdict:** | Pass |
| **Comments:** | The generated QR code is scannable and provides accurate information to system for the other user to send notifications discreetly |

**Function 5**

| **Requirement name:** | My Messages |
| --- | --- |
| **Test case description 1 :** | Verify that the user can access the "My Messages" screen. |
| **Pre-requirements:** | The user is logged into the application and navigates to the My Messages screen. |
| **Input:** | The user navigates to the "My Messages" section. |
| **Expected output:** | The application displays a screen showing the user's previous notifications. |
| **Observed output:** | The "My Messages" screen is accessible and shows the user's notification history. |
| **Verdict:** | Pass |
| **Comments:** | The user can successfully access the "My Messages" screen. |

| **Test case description 2 :** | Verify that the user's previous notifications are displayed on the "My Messages" screen. |
| --- | --- |
| **Pre-requirements:** | The user has received notifications in the past. |
| **Input:** | The user views the "My Messages" screen |
| **Expected output:** | The screen displays a list of the user's previous notifications, including details such as the notification message and date. |
| **Observed output:** | The "My Messages" screen shows a list of the user's past notifications. |
| **Verdict:** | Pass |
| **Comments:** | The system successfully retrieves and displays the user's previous notifications. |

| **Test case description 3 :** | Verify that the notifications are listed in chronological order, with the latest notification at the top. |
| --- | --- |
| **Pre-requirements:** | The user has received multiple notifications. |
| **Input:** | The user checks the "My Messages" screen. |
| **Expected output:** | Notifications are arranged in descending order based on their timestamp, with the latest notification appearing at the top. |
| **Observed output:** | Notifications are listed chronologically, with the newest notification at the top. |
| **Verdict:** | Pass |
| **Comments:** | The system correctly orders notifications based on their timestamp. |

**Function 6**

| **Requirement name:** | View Rewards Page (Future Implementation) |
| --- | --- |
| **Test case description:** | Verify that the user can access the "Rewards" section only for view purpose. |
| **Pre-requirements:** | The user is logged into the application. |
| **Input:** | The user navigates to the "Rewards" section. |
| **Expected output:** | The application provides a screen with available promo codes for fuel and car wash offers. |
| **Observed output:** | The "Rewards" screen is accessible, and promo codes for fuel and car wash offers are displayed. |
| **Verdict:** | Pass |
| **Comments:** | The user can successfully access the "Rewards" section for View purpose only. |

**Function 7**

| **Requirement name:** | Sell my Car (Future Implementation) |
| --- | --- |
| **Test case description:** | Verify that the user can access the "Sell a Car" section only for view purposes. |
| **Pre-requirements:** | The user is logged into the application. |
| **Input:** | The user navigates to the "Sell my Car" section. |
| **Expected output:** | The application provides a screen with options to view and potentially enter details for selling a car in the future. |
| **Observed output:** | The "Sell a Car" section is accessible, displaying relevant information about selling a car |
| **Verdict:** | Pass |
| **Comments:** | The user can successfully access the Sell my Car Screen for view Purpose |

| **Requirement name:** | Scenario Page |
| --- | --- |
| **Test case description:** | Verify that the system displays the list of scenarios after successfully scanning a QR code. |
| **Pre-requirements:** | The user has successfully scanned a QR code. |
| **Input:** | The user views the screen after scanning the QR code. |
| **Expected output:** | The application presents a list of 5 scenarios to the user. |
| **Observed output:** | The system displays a list of 5 scenarios on the screen. |
| **Verdict:** | Pass |
| **Comments:** | The user can see the available scenarios after scanning the QR code. |

| **Requirement name:** | Scenario Confirmation (5 pages) |
| --- | --- |
| **Test case description:** | Verify that the user can select a scenario from the list and go to its confirmation page where the user can either send or cancel the Notification to be sent.. |
| **Pre-requirements:** | The list of scenarios is displayed and user clicked on his desired scenario |
| **Input:** | The user taps on one of the scenarios. |
| **Expected output:** | The selected scenario takes the User to a confirmation page where he is given a send and a cancel option |
| **Observed output:** | The screen shows the user whether to confirm to send or to cancel the notification to be sent |
| **Verdict:** | Pass |
| **Comments:** | The chosen scenario is visually distinguished from the others. The user can successfully select a scenario from the presented list and confirm to send a notification |

| **Requirement name:** | Scenario Confirmation 1 to send the Notification |
| --- | --- |
| **Test case description:** | Verify that the system proceeds to the confirmation page upon selecting a scenario and allows the User to send a notification regarding the scenario 1 he is facing. |
| **Pre-requirements:** | Scenario Confirmation is selected to be confirmed. |
| **Input:** | The user clicks on send the Message button |
| **Expected output:** | The User successfully sends the notification to the other user whose QR code was scanned that his car has been blocked by the other user |
| **Observed output:** | The User receives a message sent button and the other User receives the notification with the scenario mentioned |
| **Verdict:** | Pass |
| **Comments:** | The user can send the other User the notifications he wants to send exclusively for only the scenarios he is facing. |

| **Requirement name:** | Scenario Confirmation 2 to send the Notification |
| --- | --- |
| **Test case description:** | Verify that the system proceeds to the confirmation page upon selecting a scenario and allows the User to send a notification regarding the scenario 2 he is facing. |
| **Pre-requirements:** | Scenario Confirmation is selected to be confirmed. |
| **Input:** | The user clicks on send the Message button |
| **Expected output:** | The User successfully sends the notification to the other user whose QR code was scanned that the users car has been hit by someone while it was parked |
| **Observed output:** | The User receives a message sent button and the other User receives the notification with the scenario mentioned |
| **Verdict:** | Pass |
| **Comments:** | The user can send the other User the notifications he wants to send exclusively for only the scenarios he is facing. |

| **Requirement name:** | Scenario Confirmation 3 to send the Notification |
| --- | --- |
| **Test case description:** | Verify that the system proceeds to the confirmation page upon selecting a scenario and allows the User to send a notification regarding the scenario 3 he is facing. |
| **Pre-requirements:** | Scenario Confirmation is selected to be confirmed. |
| **Input:** | The user clicks on send the Message button |
| **Expected output:** | The User successfully sends the notification to the other user whose QR code was scanned that the user’s car is dangerous parked in such a way it obstructs the flow of traffic |
| **Observed output:** | The User receives a message sent button and the other User receives the notification with the scenario mentioned |
| **Verdict:** | Pass |
| **Comments:** | The user can send the other User the notifications he wants to send exclusively for only the scenarios he is facing. |

| **Requirement name:** | Scenario Confirmation 4 to send the Notification |
| --- | --- |
| **Test case description:** | Verify that the system proceeds to the confirmation page upon selecting a scenario and allows the User to send a notification regarding the scenario 4 he is facing. |
| **Pre-requirements:** | Scenario Confirmation is selected to be confirmed. |
| **Input:** | The user clicks on send the Message button |
| **Expected output:** | The User successfully sends the notification to the other user whose QR code was scanned that his electric car has been fully charged and other user is waiting |
| **Observed output:** | The User receives a message sent button and the other User receives the notification with the scenario mentioned |
| **Verdict:** | Pass |
| **Comments:** | The user can send the other User the notifications he wants to send exclusively for only the scenarios he is facing. |

| **Requirement name:** | Scenario Confirmation 5 to send the Notification |
| --- | --- |
| **Test case description:** | Verify that the system proceeds to the confirmation page upon selecting a scenario and allows the User to send a notification regarding the scenario 5 he is facing. |
| **Pre-requirements:** | Scenario Confirmation is selected to be confirmed. |
| **Input:** | The user clicks on send the Message button |
| **Expected output:** | The User successfully sends the notification to the other user whose QR code was scanned that he likes and is interested to buy the car for sale |
| **Observed output:** | The User receives a message sent button and the other User receives the notification with the scenario mentioned |
| **Verdict:** | Pass |
| **Comments:** | The user can send the other User the notifications he wants to send exclusively for only the scenarios he is facing. |

| **Requirement name:** | Scenario Confirmation 1 back |
| --- | --- |
| **Test case description:** | Verify that the user can go back and change the selected scenario. |
| **Pre-requirements:** | A scenario is selected. |
| **Input:** | The user taps on the "Cancel" button. |
| **Expected output:** | The system returns to the scenario selection screen, allowing the user to choose a different scenario. |
| **Observed output:** | The application takes the user back to the scenario selection screen. |
| **Verdict:** | Pass |
| **Comments:** | Users can revisit and modify their scenario selection if needed. |

| **Requirement name:** | Scenario Confirmation 2 back |
| --- | --- |
| **Test case description:** | Verify that the user can go back and change the selected scenario. |
| **Pre-requirements:** | A scenario is selected. |
| **Input:** | The user taps on the "Cancel" button. |
| **Expected output:** | The system returns to the scenario selection screen, allowing the user to choose a different scenario. |
| **Observed output:** | The application takes the user back to the scenario selection screen. |
| **Verdict:** | Pass |
| **Comments:** | Users can revisit and modify their scenario selection if needed. |

| **Requirement name:** | Scenario Confirmation 3 back |
| --- | --- |
| **Test case description:** | Verify that the user can go back and change the selected scenario. |
| **Pre-requirements:** | A scenario is selected. |
| **Input:** | The user taps on the "Cancel" button. |
| **Expected output:** | The system returns to the scenario selection screen, allowing the user to choose a different scenario. |
| **Observed output:** | The application takes the user back to the scenario selection screen. |
| **Verdict:** | Pass |
| **Comments:** | Users can revisit and modify their scenario selection if needed. |

| **Requirement name:** | Scenario Confirmation 4 back |
| --- | --- |
| **Test case description:** | Verify that the user can go back and change the selected scenario. |
| **Pre-requirements:** | A scenario is selected. |
| **Input:** | The user taps on the "Cancel" button. |
| **Expected output:** | The system returns to the scenario selection screen, allowing the user to choose a different scenario. |
| **Observed output:** | The application takes the user back to the scenario selection screen. |
| **Verdict:** | Pass |
| **Comments:** | Users can revisit and modify their scenario selection if needed. |

| **Requirement name:** | Scenario Confirmation 5 back |
| --- | --- |
| **Test case description:** | Verify that the user can go back and change the selected scenario. |
| **Pre-requirements:** | A scenario is selected. |
| **Input:** | The user taps on the "Cancel" button. |
| **Expected output:** | The system returns to the scenario selection screen, allowing the user to choose a different scenario. |
| **Observed output:** | The application takes the user back to the scenario selection screen. |
| **Verdict:** | Pass |
| **Comments:** | Users can revisit and modify their scenario selection if needed. |

### 

### Conclusion and Future Work

#### Conclusion

In conclusion, our application represents a comprehensive solution to common automotive challenges, offering users an intuitive and feature-rich platform designed to enhance their driving experience. Through meticulous testing and refinement, we have ensured that the application delivers on its promises of functionality, usability, performance, and security. With its innovative features such as QR code scanning, messaging, car selling, and notifications, coupled with a robust backend powered by SQL, our application stands poised to revolutionize the way users navigate and interact within the automotive landscape. By prioritizing user feedback and continuous improvement, we are committed to maintaining the application's excellence and relevance in meeting the evolving needs of our diverse user base. Join us on this journey towards safer, more efficient, and environmentally-conscious driving experiences.

#### Strengths and weaknesses

Strengths:

1. Comprehensive Functionality: The application addresses a wide range of automotive challenges, including parking issues, traffic flow disruptions, and electric vehicle charging concerns, providing users with a holistic solution.

2. User-Friendly Interface: With an intuitive and visually appealing user interface, the application offers ease of navigation and accessibility, ensuring a seamless user experience.

3. Real-Time Communication: The messaging feature enables users to communicate with fellow drivers in real-time, fostering a sense of community and facilitating quick resolution of issues.

4. Integration with MSSQL: Leveraging SQL as the backend infrastructure enhances the application's scalability, reliability, and data synchronization capabilities, contributing to a seamless user experience.

Weaknesses:

1. Dependency on Internet Connectivity: While efforts have been made to minimize the reliance on internet connectivity for certain features, the application's overall functionality still requires an internet connection, which may pose limitations in areas with poor network coverage.

2. Compatibility Constraints: Limited compatibility with older Android versions may restrict access for users with outdated devices, potentially excluding a portion of the target audience.

3. Security Considerations: Despite implementing security measures, such as data encryption and secure communication protocols, vulnerabilities may still exist, necessitating continuous monitoring and updates to mitigate potential risks.

4. Localization Challenges: Ensuring accurate language translations and cultural appropriateness across diverse regions and languages may pose challenges, impacting the application's accessibility and user experience in non-English-speaking regions.

5. Performance Optimization: While efforts have been made to optimize performance, further enhancements may be required to address issues related to loading times, responsiveness, and resource utilization, particularly on devices with lower specifications.

By leveraging strengths and addressing weaknesses, the application can evolve into a more robust and user-centric solution, better equipped to meet the needs of its diverse user base.

#### Future improvements

1. Enhanced QR Code Features: Explore advanced QR code functionalities, such as incorporating augmented reality (AR) elements or expanding the types of information QR codes can convey.

2. Machine Learning Integration: Implement machine learning algorithms to improve QR code recognition accuracy, making the scanning process more robust and adaptable to diverse environments.

3. Integration with Smart Parking Systems: Collaborate with smart parking systems to provide users with real-time information on parking availability, optimizing the overall parking experience.

4. Multi-Language Support: Expand language support to cater to a more diverse user base, making the application accessible to users worldwide and fostering inclusivity.

5. Continuous User Education: Implement ongoing user education initiatives to inform users about new features, best practices, and the benefits of using the application. This can be achieved through in-app tutorials, newsletters, or video content.

6. Gamification Elements: Introduce gamification elements to encourage user engagement, such as rewarding users for active participation, achieving milestones, or contributing to the community.

7. Vehicle Telemetry Integration: Collaborate with vehicle manufacturers to integrate telemetry data, providing users with insights into their vehicle's health, performance, and usage patterns.

8. Offline Functionality: Enhance the application's offline functionality, allowing users to access critical features even in areas with limited connectivity.

9. Advanced Analytics Dashboard: Develop an analytics dashboard for users to gain insights into their driving and parking behavior, fostering a sense of awareness and encouraging responsible driving practices.

10. Adaptability to Offline Scenarios: The application's ability to deliver critical notifications via SMS ensures that users remain informed even in scenarios where internet connectivity is limited or unavailable.

11. Texting feature: Develop a texting feature in the app that allows the person interested in buying the car to interact with the seller of the car for any queries and details they may need related to the car.

11. Firebase: Firebase is a BaaS (Backend-as-a-Service) hosted by Google. It can be used for all kinds of authentication and storage of data (it is a database system). Firebase is faster and can sync data more efficiently for Android compared to other databases like MySQL.

12. Car Selling Feature: Develop a car selling feature where the user can upload his car images and enter their car details and put it for sale. The user is then directed to a hub of different cars up for sale from the car selling page.

13. Print option integration for user-generated QR code: A printing option can be added to the My QR code page where the user has generated their QR code so they can print it and stick it on their car.