

---

# Final Project Documentation

for

## Gramac Auto Online

Version 3.4

Prepared by

Brianna Beharrie	620155911	briannabeharrie2003@gmail.com
Saaeen Grant	620155794	saaeengrant.uwi@gmail.com
Tashanya Martin	620149877	tashanyamartin@gmail.com
Brianna Roper	620150889	brianna.roper17@gmail.com
Roshae Sinclair	620155876	sinclairroshae@gmail.com
Jada Walters	620154361	jada.walters@mymona.uwi.edu

**Course Instructor:** Dr. Claudine Allen

**Course:** COMP2140 – Introduction to Software Engineering

**Studio Facilitator:** Mr. Eyton Ferguson

**Date:** December 1, 2023

## TABLE OF CONTENTS

Overall Description.....	4
1.1 Overview.....	4
1.2 Product Context and Need.....	6
1.3 Product Functionality.....	6
1.4 Stakeholders and Users Characteristics.....	6
1.5 Operating Environment.....	7
1.6 Design and Implementation Constraints.....	8
1.7 Assumptions and Dependencies.....	8
2 Specific Requirements.....	9
2.1 External Interface Requirements.....	9
2.1.1 Hardware Interfaces.....	9
2.1.2 Software Interfaces.....	9
2.1.3 Communications Interfaces.....	10
2.2 Functional Requirements.....	11
Requirement 1: Create an account.....	11
Requirement 2: Update Existing Account Information.....	13
Requirement 3: Efficient Appointment Scheduling.....	15
Requirement 4: Invoice Generation.....	17
Requirement 5: Create/Add an account from the Database.....	18
Requirement 6: Update Existing Account Information from Database.....	20

Requirement 7: Delete Customer Account from Database.....	22
Requirement 8: Manage Appointments.....	23
Requirement 9: Report Generation.....	24
2.3 Behaviour Requirements.....	25
2.3.1 Use Case View.....	25
3 Other Non-functional Requirements.....	26
3.1 Performance Requirements.....	26
3.2 Safety and Security Requirements.....	27
3.3 Software Quality Attributes.....	27
<b>4 Architecture Design.....</b>	<b>29</b>
4.1 General Constraints.....	29
4.2 Alternatives Considered.....	30
4.3 System Architecture Diagram.....	32
4.4 Architecture Description.....	33
4.4.1 Architecture Justification.....	41
4.5 Architecture Decomposition.....	42
4.5.1 Component Decomposition.....	42
4.5.2 Structural Design – Class Diagram.....	45
4.5.3 Design Notes.....	46

## Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1	Brianna Beharrie, Saaeen Grant, Tashanya Martin, Brianna Roper, Roshae Sinclair, Jada Walter	Completed SRS document	11/04/23
2	Brianna Beharrie, Saaeen Grant, Tashanya Martin, Brianna Roper, Roshae Sinclair, Jada Walter	Completed SDS document	15/11/23
3	Roshae Sinclair	Completed merge of SRS and SDS document	30/11/23
3.1	Roshae Sinclair	Added test cases	30/11/23
3.2	Brianna Beharrie	Added screenshots for test cases 1-9	30/11/23
3.3	Jada Walters	Added screenshots for test cases 10-16. Changes has been made to the UI but the functionality remains the same.	1/12/2023
3.4	Roshae Sinclair	Added screenshots for test cases 17-22	1/12/2023
4	Roshae Sinclair	Updated Requirement 1 to only allow user to enter their license plate and make and model of the vehicle	1/12/2023
4.1	Roshae Sinclair	Discontinued Requirement 2 as customer users are not able to update their info from the customer side of the application	1/12/2023

## GitHub Repository

This project is an appointment scheduler created for Gramac Auto Repair and Parts Limited. The application also allows users to store vehicle information, customer details, and inquire about the parts needed for their vehicles.

The branch containing the finalized code is named "function-css." To view the application, please follow these steps:

1. Add the project folder to the **htdocs** folder in the XAMPP directory.
2. Navigate to <http://localhost/comp2140-project/public/> in your preferred web browser.
3. Select the "public" folder.

You now have access to the web application.

This project was successfully completed by the following team members:

- Brianna Beharrie
- Saaeen Grant
- Tashanya Martin
- Brianna Roper
- Roshae Sinclair
- Jada Walters

Feel free to explore the application and provide any feedback.

You can view GramacAuto by clicking the link [here](#).

# Overall Description

## 1.1 Overview

Gramac Auto Repair Services and Parts Ltd. is a family-owned auto repair shop facing operational challenges due to its size. The absence of an efficient appointment system has led to difficulties in managing customer bookings, resulting in overcrowding, delays, and internal confusion. Additionally, maintaining comprehensive vehicle information is an ongoing challenge. The company aims to develop software to address these issues efficiently. The client is seeking a software solution to improve its operations. The primary problem to be solved is the lack of an efficient appointment system, comprehensive vehicle information management, and streamlined diagnosis processes. These issues result in customer dissatisfaction, operational inefficiencies, and potential revenue loss. The intended users of the software include customers who want to schedule appointments, provide and update their vehicle information, and receive detailed invoices. Additionally, administrative staff members need access to the system to manage customer and vehicle data, view upcoming appointments, generate reports, and ensure the system's security.

Customers who visit the shop for the first time or those who have work done on their vehicles are given the option to sign up on their own through a user-friendly interface or by visiting the admin staff, who can create an account for them. This software solution aims to improve the operations of this family-owned auto repair shop by providing a comprehensive system that allows customers to easily access and manage their vehicle information, schedule appointments, and receive detailed invoices. Administrative staff members also have access to the system to efficiently manage (add, update or delete) customer and vehicle data, view upcoming

appointments, and generate reports for informed decision-making. The software not only enhances the customer experience but also streamlines internal operations, ultimately improving the overall efficiency of Gramac Auto Repair Services and Parts Ltd.

The system must ensure efficient appointment scheduling, customer information management, and data security while meeting non-functional requirements such as performance, safety, and security. This context highlights the importance of the software's ability to handle concurrent users, generate reports quickly, and protect sensitive customer data, as well as the need for a secure authentication system. The software must also be scalable to accommodate a growing number of customers and vehicle records without performance degradation. Overall, the software aims to enhance the customer experience, streamline internal operations, and improve the overall efficiency of Gramac Auto Repair Services and Parts Ltd.

## 1.2 Product Context and Need

Gramac Auto Repair Services and Parts Ltd., a family-owned auto repair shop, faces significant operational challenges due to its size. One of the most pressing issues is the absence of an efficient appointment system, resulting in difficulties managing customer bookings. This leads to overcrowding, delays, and internal confusion within the business. Moreover, maintaining comprehensive vehicle information is an ongoing challenge. Therefore, software needs to be created in order to execute these tasks efficiently.

## 1.3 Product Functionality

- a. **Invoicing and Service Summary:** Generate detailed invoices summarizing selected services, parts, and costs.
- b. **Efficient Appointment Scheduling:** Streamline appointment scheduling for smoother workflow management.
- c. **Detailed Vehicle Information Documentation:** Record and store comprehensive vehicle information in a database. Include insurance and vehicle fitness data. Enable clients to access this information, including service due dates, using their own credentials.
- d. **Informative Reports:** Generate reports from the customer and vehicle information stored in a database on the administration side to provide valuable insights and data analysis.

## 1.4 Stakeholders and Users Characteristics

1. **Internal stakeholders** primarily comprise the **auto repair shop manager, the administrative staff and the team of skilled mechanics**.
  - a. The manager of Gramac plays a huge role in the development of this software as their support in providing the key requirements of this software is necessary.
  - b. The mechanics depend on accurate job details in order to carry out their repair. Therefore, it is important that the software aligns with their requirements and is critical to maintaining repair quality and overall effectiveness.

- c. The admin staff are integral to daily operations. Their efficient use of the management system directly impacts customer experience as they will be the ones mainly interacting with the system.
2. **External stakeholders**, on the other hand, encompass the diverse clientele served by Gramac Auto Repair Services and Parts Ltd.
- a. Customers are the lifeblood of the business, expecting reliable service scheduling, clear progress updates, and seamless communication. Meeting these expectations is essential for customer satisfaction, loyalty, and the shop's reputation.

The most important stakeholders for the software product are the Auto Repair Shop Manager, Administrative Staff, Customers, and Mechanics. These stakeholders have direct involvement in the shop's daily operations, and the software significantly impacts their roles and interactions. The success of the software in improving efficiency, customer satisfaction, and operational effectiveness hinges on meeting the needs and expectations of these primary stakeholders. **While other stakeholders like Suppliers and Parts Vendors, Insurance Companies, and Regulatory Authorities are also important, their satisfaction is indirectly linked to the software's primary objectives and may be secondary in terms of day-to-day software utilization.**

## 1.5 Operating Environment

The vehicle management and appointment scheduling website for the auto repair shop will operate in a typical web-based environment. The software will primarily be a web application accessible through standard web browsers such as Google Chrome, Safari, Mozilla Firefox, or Microsoft Edge, and it will require a minimum platform of modern hardware and software components. The system should work seamlessly on popular operating systems like Windows (versions 7 and above), macOS (versions 10.10 and above), and Linux. It will rely on a server-side scripting language like PHP for server-side processing. The front-end will use HTML5, CSS3, and JavaScript for the user interface. Database management can be achieved with relational database systems like MySQL. The application should also be responsive and

user-friendly, catering to various devices and screen sizes, including desktop computers, laptops, tablets, and smartphones. To ensure optimal performance, a reliable internet connection will be necessary for both the clients and the web server. Additionally, the system should be compatible with the latest versions of common web technologies to ensure a secure and efficient user experience.

## **1.6 Design and Implementation Constraints**

- The system must be able to handle an increasing number of users and appointments as the business grows and handle peak booking times.
- The client will be responsible for the maintenance or further evolution of the system. Therefore, updates to the system will be made on the part of the client.
- The software must be transferable to personal computers and mobile devices.
- The client must have appropriate resources to be able to run the software on their system.
- The system should be highly available and reliable to prevent downtime and ensure users can make appointments at any time.
- Employees and customers must be trained to use the system effectively as this is essential for its success.

## **1.7 Assumptions and Dependencies**

The following are assumptions that could possibly affect the requirements and performance of software stated in this SRS document.

- The users have sufficient literacy and analytical skills to understand how to efficiently use the software.
- The users and employees have the proper requirements to efficiently use the software. For example, internet connectivity.
- The system has real-time data synchronization so that both customers and employees see up-to-date information about appointment availability and changes.
- The system depends on a reliable and secure database to secure customer information, appointment details and historical data.

- The system depends on timely updates, bug fixes and technical support to function effectively

## 2 Specific Requirements

### 2.1 External Interface Requirements

#### 2.1.1 Hardware Interfaces

The software is accessible and fully functional on a variety of devices, including desktop computers, laptops, tablets, and smartphones, ensuring broad compatibility for users. In addition, it's designed to perform optimally on devices meeting specific minimum specifications, including a single processor, 4GB of RAM, and a minimum 10Mbps download speed. This approach enhances the user experience and accessibility by accommodating a wide range of device types while ensuring smooth performance.

#### 2.1.2 Software Interfaces

The software's compatibility extends to a range of client devices and operating systems, ensuring widespread accessibility. Specifically, it shall function seamlessly on Windows versions 7 and above, macOS versions 10.10 and above, Linux, Android, and iOS, making it adaptable to the majority of commonly used operating systems.

Moreover, to maintain a consistent and high-quality user experience, the software has been designed to be compatible with various web browsers. This includes the latest versions of Google Chrome, Mozilla Firefox, Apple Safari, and Microsoft Edge, assuring that users can access the software across different browsers without any compatibility issues.

In addition to compatibility with operating systems and web browsers, the software will interface with crucial third-party software components. One such component is the Database Management System (DBMS), where the software is set to interact with MySQL for data storage and retrieval.

This interaction is designed to support SQL queries, facilitating smooth communication with essential third-party software for efficient data storage and retrieval.

### **2.1.3 Communications Interfaces**

Since users and admin staff will be entering some personal information onto the system over an internet connection the use of HTTPS to add a layer of encryption on the user's information is vitale. After the service is selected, the system must send an email to the customer as confirmation that an appointment was scheduled online. The email should contain the invoice, which can be printed or downloaded onto the device. SMTP (Simple Mail Transfer Protocol) is the underlying communication standard that will be used for sending these email based communications.

## 2.2 Functional Requirements

### Requirement 1: Create an account

**Use Case:** Add account

**Rationale:** To provide customers of the auto repair shop with an account for convenient access to detailed information about their vehicles.

**User Requirements:** The system shall allow users to sign up to the website and upload comprehensive information about their vehicles, including make, model, tire specifications, insurance information, and other relevant details.

#### System Requirements:

**1.1** The system shall provide a user registration interface on the website.

**1.2** Users shall be required to provide the following information during registration:

- Full Name
- Email Address
- Password
- Contact Number

**1.3** After successful registration, the system shall allow users to add comprehensive information about their vehicles.

**1.4** Users shall be able to add multiple vehicles to their account. 2.3 The following vehicle details shall be captured for each vehicle:

- **1.4.1** Make and model of the customer's vehicle.
- **1.4.2** License plate number

**Acceptance Criteria:**

1. 100% of the customer's vehicle information added on the system should be saved and reflected in system

**Dependencies:** None

**Priority:** High

**Team Owner:** Brianna Beharrie

**Requirement 2: Update Existing Account Information (Requirement Discontinued.)**

**Customer is NOT able to edit their own information. Only Admins can.)**

**Use Case:** Update account

**Rationale:** To provide customers of the auto repair shop with the option to update their previous saved information conveniently and easily.

**User Requirements:** The system shall allow users to update their customer and/or vehicle information that was saved on the website.

**System Requirements:**

**2.1** The system shall require users to authenticate themselves before they can update their account information.

**2.2** Users shall log in using their registered email and password.

**2.3** Upon successful authentication, the system shall allow users to update their customer information, including but not limited to:

- Full Name
- Contact Number

**2.4** Users shall be able to modify their customer information at any time.

**2.5** The system shall provide users with the capability to update vehicle information associated with their account.

**2.6** Users shall be able to modify the following vehicle details:

- **2.6.1** Make and model of the customer's vehicle.

**2.7** The system shall validate user inputs during the update process and display clear error messages in case of incorrect or missing data.

**2.8** Users shall receive feedback confirming the successful update of their information.

**Acceptance Criteria:**

2. 100% of the customer's vehicle information updated on the system should be saved and reflected in system

**Dependencies:** None

**Priority:** High

**Team Owner:** Brianna Beharrie

**Requirement 3: Efficient Appointment Scheduling**

**Use Case:** Schedule Appointments

**Rationale:** To streamline the appointment scheduling process and enhance customer experience, the system should enable customers to request, schedule, and manage appointments online.

**User Requirements:** The system shall allow users to schedule the appointments

**System Requirements:**

**3.1** The system shall maintain a dynamic and up-to-date schedule of staff availability, their assigned tasks, and expected workloads. This schedule should be easily accessible for appointment scheduling.

**3.2** The software should continuously update and manage appointment slots in real-time to reflect the current availability of staff members. This includes accounting for any new appointments, cancellations, or rescheduling.

**3.3** The system should offer a user-friendly interface for customers to schedule appointments online. This interface should include features such as:

- **3.3.1** Browsing available time slots based on staff availability.
- **3.3.2** Selecting preferred appointment dates and times.
- **3.3.3** Inputting specific appointment requirements or notes.
- **3.3.4** Providing confirmation of the selected appointment details.

**3.5** The system should enable customers to reschedule or cancel appointments through the online platform.

**Acceptance Criteria:**

1. Whenever an appointment is booked, the time slot should become unavailable. There should be no discrepancies between the admin-side and the customer side.

**Dependencies:** None

**Priority:** High

**Team Owner:** Roshae Sinclair

## **Requirement 4: Invoice Generation**

**Use Case:** Generate Invoice

**Rationale:** To improve transparency and customer experience, the system should allow customers to select services, generate detailed invoices, and maintain a secure record of service details.

**User Requirements:** The system shall allow users to select repair preferences and/or diagnostic services and print/download a detailed invoice summarizing selected services, associated parts, and costs

### **System Requirements:**

**4.1** The system must securely store customer service selections and generated invoices, ensuring that customer data is protected and maintained in a secure environment.

**4.2** Invoices should be easily accessible and retrievable for future reference, both by customers and authorized staff, to facilitate post-service inquiries and record-keeping.

**4.3** The software must accurately calculate and display costs for services and associated parts, ensuring that customers receive a precise summary of their expenses.

### **Acceptance Criteria:**

1. The system should securely store 100% of customer service selections, generate detailed invoices with accurate cost summaries, and allow customers to review, approve, request modifications to, and download/print their invoices.

**Dependencies:** None

**Priority:** High

**Team Owner:** Jada Walters

**Requirement 5: Create/Add an account from the Database****Use Case:** Create Account**Rationale:** To provide the administration staff the option to create accounts for customers from the administration end of the system.**User Requirements:** The system shall allow the admin staff to login to the system using admin credentials and upload comprehensive information about customer's vehicles, including make, model, tire specifications, insurance information, and other relevant details.**System Requirements:**

- **5.1** The system shall provide a secure login interface for administrative staff to access the system.
- **5.2** Admin users shall be required to enter their unique admin credentials (e.g., username and password) for authentication.
- **5.3** The system shall grant access to the account creation functionality only to users with valid admin credentials.
- **5.4** Non-admin users shall not have access to the admin account creation feature.
- **5.5** Upon successful admin authentication, the system shall allow admin staff to create customer accounts.
- **5.6** Admin staff shall be able to add comprehensive information about customer vehicles, including:
- **5.7** Admin staff shall have the option to create accounts for multiple customers.
- **5.8** The system shall validate admin input for accuracy and completeness during the account creation process.
- **5.9** Clear error messages shall be displayed in case of incorrect or missing data.

**Acceptance Criteria:**

100% of the customer's vehicle information added on the system should be saved and reflected in the system's database

**Dependencies:** None

**Priority:** High

**Team Owner:** Tashanya Martin

## **Requirement 6: Update Existing Account Information from Database**

**Use Case:** Update customer vehicle info from Database

**Rationale:** To provide administration staff of the auto repair shop with the option to update search for accounts and update the customer or customer's vehicle information.

**User Requirements:** The system shall allow administration staff users to update customer accounts and/or vehicle information that was saved on the website.

### **System Requirements:**

**6.1** Admin users shall be required to enter their unique admin credentials (e.g., username and password) for authentication.

**6.2** The system shall grant access to the account update functionality only to users with valid admin credentials. Non-admin users shall not have access to the admin account update feature.

**6.3** After successful admin authentication, the system shall allow admin staff to search for and select customer accounts.

**6.4** Admin staff shall have the capability to search by customer name, customer ID, or other relevant criteria.

**6.5** The system shall provide admin staff with the capability to update vehicle information associated with customer accounts.

**6.6** The system shall validate admin input for accuracy and completeness during the account update process.

**6.7** Clear error messages shall be displayed in case of incorrect or missing data.

### **Acceptance Criteria:**

100% of the customer's vehicle information updated on the system should be saved and reflected in system

**Dependencies:** None

**Priority:** High

**Team Owner:** Saaeen Grant

## **Requirement 7: Delete Customer Account from Database**

**Use Case:** Delete account from database

**Rationale:** To allow administration staff to delete account/s from the database if requested by the customer

**User Requirements:** The system shall allow administration staff users to delete a customer's account

### **System Requirements:**

**7.1** Admin users shall be required to enter their unique admin credentials (e.g., username and password) for authentication.

**7.2** The system shall grant access to the account update functionality only to users with valid admin credentials. Non-admin users shall not have access to the admin account update feature.

**7.3** After successful admin authentication, the system shall allow admin staff to search for and select customer accounts for deletion.

**7.4** Once admin staff initiates the account deletion process and provides confirmation, the system shall begin the account deletion procedure.

**7.5** The system shall permanently delete the customer account and associated vehicle information.

### **Acceptance Criteria:**

100% of the customer's vehicle information deleted on the system should be permanently deleted from the database and website.

**Dependencies:** None

**Priority:** High

**Team Owner:** Roshae Sinclair

## **Requirement 8: Manage Appointments**

**Use Case:** Manage Appointments

**Rationale:** To provide administrators with a centralized and accessible database of pending appointments.

**User Requirements:** The system shall allow user (administrators) to view upcoming appointments.

### **System Requirements:**

**8.1** The database shall be able to efficiently store and manage customer appointment history.

**8.2** The database shall enable administrators to search for appointments by name, contact information or vehicle information.

**8.3** The system shall generate reports on appointment history and vehicle information within a specified time period.

### **Acceptance Criteria:**

1. System should store and manage 100% of customer appointment history and allow administrators to search appointments by name, contact info, or vehicle details, and generates reports on appointment history and vehicle info within a set time period.

**Dependencies:** Requirement 1 & 2

**Priority:** High    **Team Owner:** Brianna Roper

**Requirement 9: Report Generation**

**Use Case:** Generate Reports

**Rationale:** To provide administrators with the ability to extract and analyze essential information from the customer vehicle database, facilitating data-driven decision-making and insights for the business. This feature enhances the overall management of customer information and vehicle service history.

**User Requirements:** The system shall allow users (administrators) to generate reports based on the customer vehicle information stored in the database.

**System Requirements:**

**9.1** The system shall have the capability to generate reports from the customer vehicle database.

**9.2** Administrators shall be able to select various report parameters, such as date range, customer name, vehicle information, or service history.

**9.3** Reports shall include detailed customer information, vehicle details, and service history for the specified criteria.

**Acceptance Criteria:**

1. Administrators should be able to generate comprehensive reports based on various criteria, ensuring that relevant customer information and vehicle service history are available for analysis and decision-making.

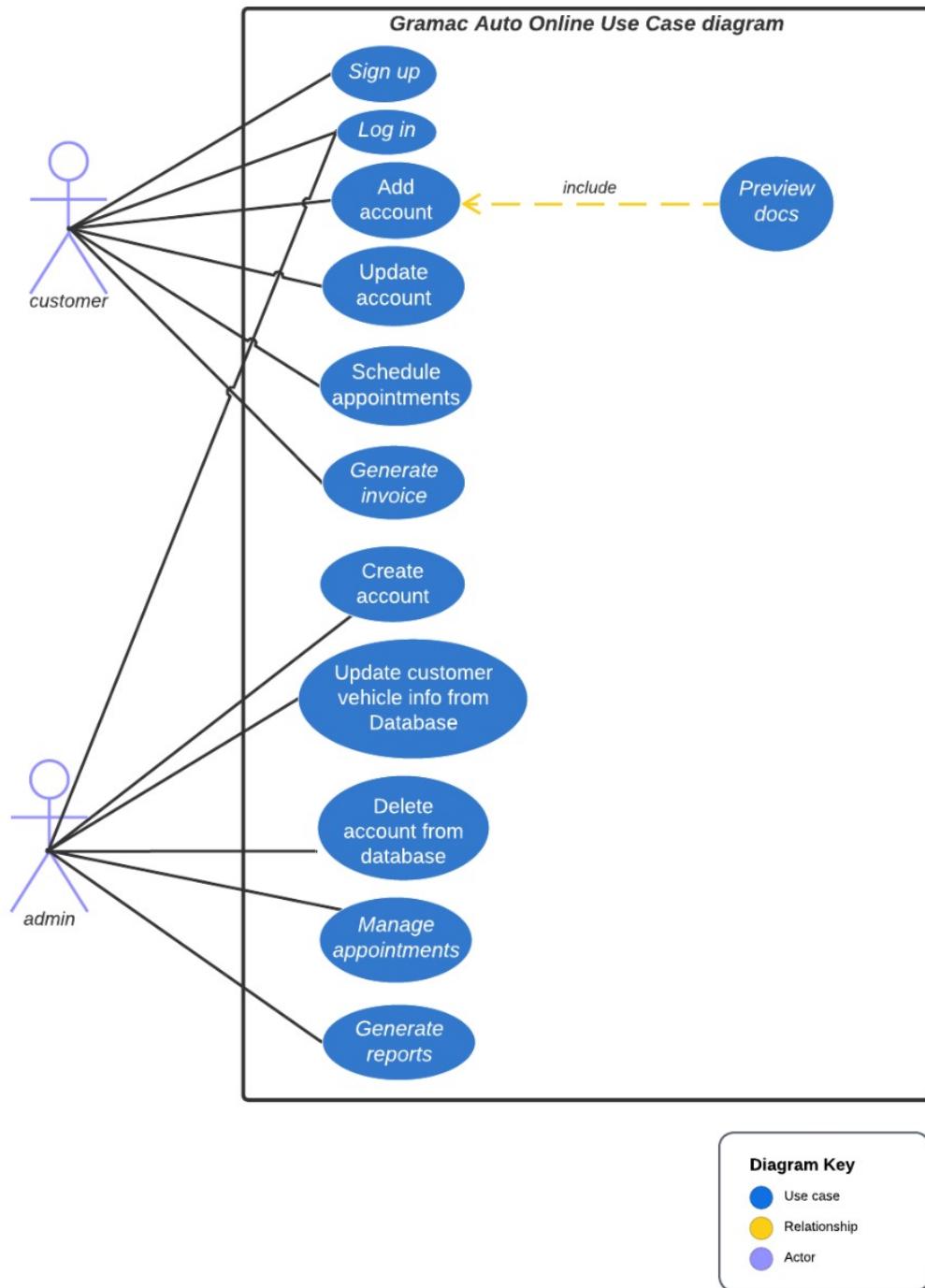
**Dependencies:** Requirement 1 & 2

**Priority:** High

**Team Owner:** Roshae Sinclair

## 2.3 Behaviour Requirements

### 2.3.1 Use Case View



## 3 Other Non-functional Requirements

### 3.1 Performance Requirements

**3.1.1. Requirement 1:** The desired data must be accessed within 3 seconds of an administrator submitting a query.

**Rationale:** Administrators need to be able to access data quickly and easily to perform their jobs efficiently.

**3.1.2. Requirement 2:** The system must be able to handle 50 users concurrently without any noticeable delays in response times.

**Rationale:** Ensures good user experience for all users by minimizing performance degradation.

**3.1.3. Requirement 3:** Reports must be generated within 10 seconds of an administrator making a report request.

**Rationale:** Reports must be generated quickly so that administrators are able to make quick and informed decisions.

**3.1.4. Requirement 4:** The vehicle information dashboard must load within 2 seconds of successful log in.

**Rationale:** Quick loading time is essential for the customer experience, otherwise it is disruptive and frustrating leading to the process being abandoned.

**3.1.5. Requirement 5:** The system must not exceed 200 MB of RAM.

**Rationale:** The system should use memory efficiently otherwise it will impede the general performance including the other performance requirements.

## 3.2 Safety and Security Requirements

### 3.2.1. Safety Requirements:

- The system will inhibit customers from having administrators privileges such as adding, deleting, editing or viewing other customer information.
- The systems shall encrypt all customer data when not being viewed by the customer or administrator.
- The system shall not store any cookies on the customer's computer containing personal information.

### 3.2.2. Security Requirements:

- The systems must verify the identity of the users by the use of passwords, usernames and two step verification before granting system access.
- The systems must keep a record of user activity to investigate u
- The system must comply with applicable security regulations such as the General Data Protection Regulation.

## 3.3 Software Quality Attributes

### 3.3.1. Maintainability:

The system will be developed with a modular structure, making it easier to update or add new features when we see fit. The code base will be comprehensive, and each module should contain only 250 lines of code in order to uphold maintainability.

### 3.3.2. Security:

The system shall be secured with the implementation of proper authentication and authorisation so that customer login and vehicle information access are secure, allowing only authorised users to access this data. For adequate data protection sensitive customer data, such as personal information and service history, will be securely stored and protected from unauthorised access.

### 3.3.3. Scalability:

The system shall be scalable by accommodating an increasing number of users, appointments, and vehicle records without performance degradation. There should be limited errors and failure as the number of customers and customer data grows.

## **4 Architecture Design**

### **4.1 General Constraints**

There are several general constraints associated with this software system for Gramac Auto Online which are integral to its functionality. Firstly, the system must possess the capability to adapt to a growing user base and increasing appointment demands, particularly during peak booking periods, ensuring scalability and efficiency as the business expands. Another significant constraint lies in the client's role, as they are entrusted with the maintenance and further development of the system, requiring them to manage all updates. Additionally, the software's versatility, being transferable to personal computers and mobile devices, demands that the client possesses adequate resources to run it seamlessly on their system. Furthermore, the client's personal computer or mobile device must meet specified criteria, including sufficient memory, for optimal software performance. Lastly, the effectiveness of the system is contingent on the training of both employees and customers, emphasizing the importance of proficiency to ensure the system's success. These constraints collectively shape the parameters within which the appointment scheduling system for Gramac Auto Online must operate to meet the dynamic needs of the business and its users.

## 4.2 Alternatives Considered

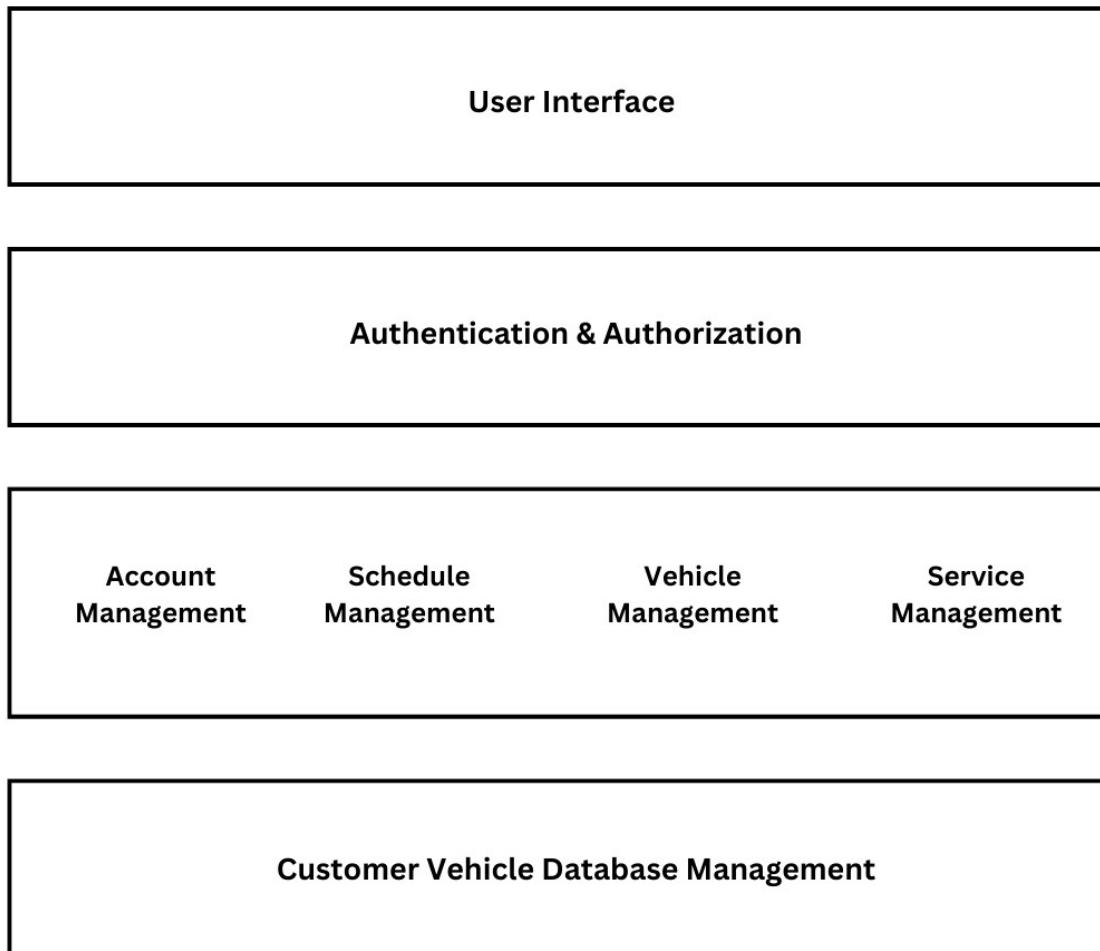
Other models considered for the system architecture were the "Model-View-Controller" (MVC) and "Client-Server" models. The MVC model breaks down the system into model, controller, and view components, providing a structured approach that accommodates multiple ways to interact with data. This separation is particularly beneficial when facing uncertain future requirements, enabling data and representation to evolve independently. However, in larger systems handling distributed tasks, the MVC model may introduce additional code complexity, potentially impeding team efficiency.

On the other hand, the Client-Server model organizes the system into services, each delivered by a separate server. This approach facilitates access to a shared database from different locations, but it comes with drawbacks. Each service becomes a single point of failure, and performance becomes unpredictable due to reliance on the network and the system as a whole.

The repository pattern was considered in the evaluation of system architectures, but it ultimately wasn't chosen for several reasons. The repository pattern primarily focuses on abstracting the data access layer, which can be beneficial for decoupling the application from the underlying database implementation. However, in the context of this system, the layered model was favored for its ability to enforce a clear separation of concerns, support simplified unit testing, and enhance security measures at different layers. The repository pattern, while effective for certain scenarios, was deemed less suitable for the specific requirements of the project, as it didn't provide the comprehensive advantages offered by the chosen layered architecture.

Ultimately, the chosen approach was a layered model for several reasons. This model enforces a clear separation of concerns, allowing for simplified unit testing and bolstering security by implementing measures at each layer. The decision was influenced by the drawbacks of the MVC model, which could introduce unnecessary complexity, repository, as it created a separation with the data access components and the components, and the potential unreliability of the Client-Server model due to its dependence on network performance. In light of these considerations, the layered model emerged as the preferred choice for the implementation of the system.

### **4.3 System Architecture Diagram**



## 4.4 Architecture Description

The layered architecture design was selected for our software because of its self-independent nature since all components are interconnected but not dependent on each other. This allows for greater efficiency since modifications can be made to a module without affecting another, stronger security for each layer and the database and, also enables more simplified unit testing to be done. There are four layers in the design: *the presentation layer, business layer, persistence layer and database layer*. The presentation layer involves the **user interface component** which manages user interactions, including account creation, updates, appointment scheduling, and invoice generation. The business layer involves **authentication and authorization component** which handles authentication, authorization, appointment management, and invoice generation logic. The persistence layer involves **the account management, schedule management, vehicle management and service management** components. This layer manages data storage and retrieval for account details, vehicle information, appointments, and invoices. The database layer involves **the customer vehicle database management** component which stores and manages customer vehicle information.

Table 1: **Interaction Overview:**

<b><i>Interaction</i></b>	<b><i>Description</i></b>
Presentation Layer to Business Layer:	User inputs from the presentation layer (e.g., account creation, updates) are forwarded to the business layer. Processed results (e.g., schedules, invoices) are returned.
Business Layer to Persistence Layer:	Business logic execution triggers data storage and retrieval operations in the persistence layer.
Persistence Layer to Database Layer:	The persistence layer interacts with the database layer to store and retrieve data (account, vehicle, appointments, invoices).
Database Layer to Persistence Layer:	Efficient data storage and retrieval operations are ensured by communication between the database and persistence layers.

### **Presentation Layer-User Interface:**

#### *Functional Interaction:*

**User Interactions:** The presentation layer provides interfaces for customers and administrators, enabling them to interact with the system. It facilitates actions such as account creation, updates, appointment scheduling, and invoice generation.

**Data Input and Display:** Collects user inputs during account creation and updates, then displays relevant information to users.

#### *Non-Functional Interaction:*

**Quick Response Time:** Ensures a responsive and user-friendly interface, meeting performance requirements by minimizing delays in data presentation.

**Security Measures:** Enforces secure user interactions, with the layer inhibiting unauthorized access and ensuring that sensitive data is encrypted.

### **Business Layer- Authentication and Authorisation:**

#### *Functional Interaction:*

**Authentication and Authorization:** Validates user credentials during login, ensuring only authorized users can access and perform actions within the system.

**Business Logic Execution:** Executes business logic for appointment management and invoice generation, enforcing rules and processes.

#### *Non-Functional Interaction:*

**Efficient Logic Execution:** Ensures that business logic executes efficiently, contributing to the system's overall performance.

**Security Measures:** Verifies user identity, records user activity, and adheres to security regulations, meeting safety and security requirements.

**Persistence Layer- The Account management, Schedule management, Vehicle management and Service management:**

*Functional Interaction:*

**Data Storage and Retrieval:** Manages the storage and retrieval of account details, vehicle information, appointments, and invoices.

**Supports Business Functions:** Enables the execution of business functions related to account management, schedule management, vehicle management, and service management.

*Non-Functional Interaction:*

**Efficient Data Management:** Ensures efficient storage and retrieval of data, contributing to quick response times and meeting performance requirements.

**Data Security:** Safeguards customer and system data, aligning with safety and security requirements.

### **Database Layer- The Customer vehicle database management:**

#### *Functional Interaction:*

**Data Storage:** Stores and manages customer vehicle information, serving as the persistent data store for the system.

#### *Non-Functional Interaction:*

**Efficient Database Operations:** Optimizes database operations for quick access to customer vehicle information, supporting performance requirements.

**Data Security:** Ensures the secure storage of sensitive customer information, aligning with safety and security requirements.

These interactions create a cohesive flow, ensuring that user actions lead to the execution of business logic, efficient data management, and secure storage of information, thereby meeting both functional and non-functional requirements.

Table 1.2: Component Overview:

<i>Component</i>	<i>Interaction</i>
Presentation Layer:	<ul style="list-style-type: none"><li>- Facilitates user interactions (account creation, updates).</li><li>- Displays processed results.</li></ul>
Business Layer:	<ul style="list-style-type: none"><li>- Validates user credentials.</li><li>- Executes business logic (appointment, invoice).</li></ul>
Persistence Layer:	<ul style="list-style-type: none"><li>- Manages data storage and retrieval for accounts, vehicles, appointments, invoices.</li></ul>
Database Layer:	<ul style="list-style-type: none"><li>- Stores and manages customer vehicle information.</li></ul>

Table 1.3: Non-functional Considerations:

Non-Functional Aspect	Consideration
Quick Response Time:	<ul style="list-style-type: none"><li>- Achieved through efficient communication between layers and optimized logic execution.</li></ul>
Security Measures:	<ul style="list-style-type: none"><li>- Enforced across layers to verify user identity, secure data storage, and restrict unauthorized access.</li></ul>
Efficient Data Management:	<ul style="list-style-type: none"><li>- Optimized operations between persistence and database layers to ensure efficient data storage</li></ul>

#### **4.4.1 Architecture Justification**

The system will implement a layered architecture pattern as it possesses several advantageous characteristics that occur as a consequence organizing the software into distinct layers. This architecture enforces separation of concerns, due to the partitioning of functionalities. This modularity allows for simplified unit testing as the components are not as strongly interleaved compared to other architecture patterns.

Layered architecture strengthens security by enabling the implementation of security measures at each layer. This layered security approach protects sensitive data and system resources from unauthorized access. By having this multi-layer approach to security implementation can have more robust security of the customer and admin database.

Furthermore this pattern allows modifications to be made within specific layers without significantly impacting the other functionalities. This modularity also facilitates collaboration among team members. Different layers can be assigned to separate individuals, enabling parallel development and reduced conflicts. Each team member can focus on their assigned layers without interfering with the work of others, improving communication and overall development efficiency.

## 4.5 Architecture Decomposition

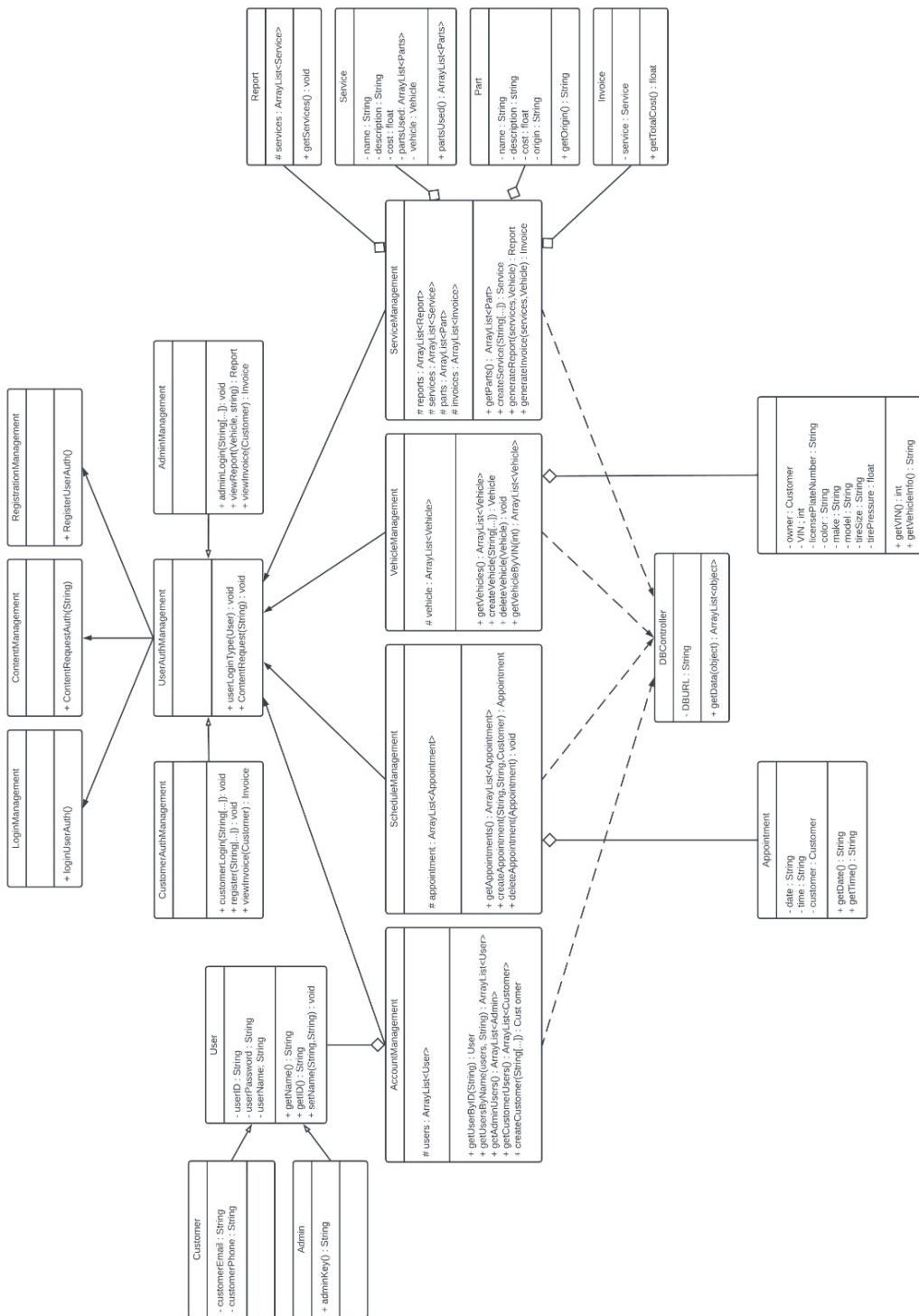
### 4.5.1 Component Decomposition

Requirement ID	Architecture Component	Class Name	Description
Req 1	User Interface	LoginManagement ContentManagement  RegistrationManagement  UserAuthManagement	<p>Displays the user log-in page</p> <p>After authorized to view the content, the interface display and give the users access to other pages such as the ScheduleMangement , Vehicle Management, Service Management and the Account Management</p> <p>Displays the interface for the user to register</p> <p>Control all the page content displayed to the user</p>
Req. 1	Authentication & Authorization	UserAuthManagement	Enables a user (Customer or Admin class) with an account to gain access to the system with proper credentials

Req 1 Req 2	Account Management	AccountManagement  User  Admin  Customer	Allows user (based on their role) to perform tasks on their account based on their role With proper authorization (UserAuthManagement), the user can login into a Customer or Admin account Allows administration of the site to login after successfully passing the UIManagement and UserAuthManagement and perform tasks such as search for users based on ID or name or create a customer account (Admin) Contains the customer's email and their phone number
Req 1 Req. 5	Vehicle Management	Vehicle  Vehicle Management	Each vehicle has an owner, VIN, license plate number, color make, model, tire size and tire pressure. Users are able to view the details of the vehicle and search based on the VIN Allows user to get a list of vehicles (Vehicle class), search or modify a Vehicle.

Req 3 Req 8	Schedule Management	ScheduleManagement Appointment	Stores the appointments (Appointment class)  Allows user to set Appointment with a date and time or search for appointment and delete.
Req 4 Req 9	Service Management	Report  Service  Part  Invoice	Only available to the admin after authorization (UserAuthManagement). Shows a list of services.  Allows users to view the services available and other details Allows user to view the parts and other details Available to the Customer after a Service is chosen.
Req. 5 Req. 6 Req. 7	Vehicle Database Management	DBController	Stores the accounts, schedules, vehicles and services.

## 4.5.2 Structural Design – Class Diagram



### 4.5.3 Design Notes

The UserAuthManagement is the parent class for the AccountManagement, ScheduleManagement, VehicleManagement and ServiceManagement classes. This class restricts the direct access to certain information due to prevent users who do not have an account in the database (managed by the DBController class) to have access to information they don't have permission to. The AccountManagement class, which is dependent on the DBController class, is made up of many users (User class) which contains information about their user id, user password and their username. The AccountManagement and the User class uses assessors and mutators to get the information and to manipulate data stored in the variable. The User class has two child classes which are the Customer and Admin class which provides the customer's email and phone number (customer) and the adminKey(admin).

Another child class of UserAuthManagement is the ScheduleManagement which controls the appointments sent by the user. This class contains mutators and accessors which allows the user to view, create and destroy. These appointments fall under the class, Schedule which stores the customer and the date and time they selected.

Furthermore, the VehicleManagement, which is dependent on the DBController, is also a child class of the UserAuthManagement. The VehicleManagement class controls the viewing of the vehicles (Vehicle class) and their information as well as creating and editing a vehicle from the Vehicle class.

Finally the Parts, Service, Report and Invoice classes all inherit from the class

ServiceManagement. From the ServiceManagement, the user is able to view the parts' name, description, cost and origin, select a service, generate a invoice or generate the report.

The backbone of all of these classes is the DBController which stores databases for the schedules, customers, admins and vehicles.

The constraints that were taken in the considerations are:

1. The UserAuthManagement class imposes restrictions on direct access to certain information, ensuring that only users with valid accounts in the database can access relevant data.
2. The AccountManagement class controls user access by verifying credentials through the use of user id and password.
3. The classes Customer and Admin inherit from the User class, indicating an inheritance hierarchy where specific user types share common attributes and functionalities.
4. The backbone of the system, the DBController class, serves as the storage and management system for databases related to schedules, customers, admins, vehicles, parts, services, reports, and invoices, therefore the system cannot function without this database.



## 5 Test Cases

<u>Test Case</u>	<u>Test Data</u>	<u>Associated Requirement</u>	<u>Expected Result</u>	<u>Actual Result</u>	<u>Pass/Fail</u>	<u>Comment</u>
Case 1: Check user register with correct format for first and last name, email address and matching passwords.	“Bob” “Brown” Bob.brown@gm ail.com “876-757-3450” “Iamthebob” “Iamthebob”	Req #1.2	Registration Successful	Registration Successful	Pass	After the user registers, they will be redirected to the Login Page.
Case 2: Check the user register with email	“Bob” “Brown” Bob.brown@gm ail.com	Req #1.2	Email has already been used	Email has already been used	Pass	After the user registers with the same email, the ‘Email has

already in database	“876-757-3450” “Iamthebob” “Iamthebob”					already been used’ message comes up and prevents them from registering again.
Case 3: Check user register with incorrect format for email but correct name format and matching passwords.	“Jane” “Doe” “janedoe.yahoo.com” “JAneYdoit?” “JAneYdoit?”	Req #1.2	Invalid Email Address	Invalid Email Address	Pass	System will ask user to input appropriate email address
Case 4: Check user vehicle	“2001” “Toyota”	Req #1.4	Vehicle Information	Vehicle Information	Pass	If other appointment

input with correct format for year, make, model and additional model information.	“RAV4” “Burgundy, black interior, dent in the right side door”		Added Successfully	Added Successfully		information is entered, the system saves the vehicle (license plate, year, make, model) into the vehicles table in the database.
Case 5: Check user vehicle input with no format for year, make, model and additional model	- - - -	Req #1.4	Selection required	Selection required	Pass	User will be given a message saying that all the details are required.

information inputted.						
Case 6: Check user login to system with admin email + password correct	“mechanicnumber1@example.com” “password1”	Req #2.2	Access Granted	Access Granted	Pass	User will be given access to the full system
Case 7: Check user login to system with incorrect admin email + correct password	“mechanicnumber1@example.com” “p012393e”	Req #2.2	No Access Granted	No Access Granted	Pass	The password textbox is cleared, however no message shown
Case 8: Check user login to system with	“bob.brown@example.com” “bobthebuilder”	Req #2.2	No Access Granted	No Access Granted	Pass	The password textbox is cleared,

incorrect admin email + incorrect password						however no message shown
Case 9: Check user login with no email or account added.	-	Req #2.2	No Access Granted	No Access Granted	Pass	Page refreshes but no message shown. User not allowed to the appointments page
Case 10: Check appointment scheduling with date and time selected as well as one or more services	User chooses “December 5, 2023”  User selects time “8:00 - 10:00”	Req #3.3	Appointment Saved	Appointment Saved	Pass	After the appointment is saved, the user is sent back to the homepage

	User selects “Braking” and “Oil Change”					
Case 11: Check appointment scheduling with no date or time, drop off or service selected		Req #3.3	Selection required	Selection required	Pass	User will be given a message saying that all the details are required.
Case 12: Check if customer can download their invoice after appointment is made.		Req #4	Invoice downloaded successfully	Invoice download failed	Fail	The code to implement the invoice is not functional but the user can review their

						order after setting up an appointment
Case 13: Check if customers are able to purchase parts	“John Kennedy” “john.kennedy@po.com”  “Good day. Do you have engine for 2014 Mazda 5?”	Req #4.3	Inquiry sent	Inquiry not sent	Fail	There is no email address connected to this function at the moment therefore the email was not sent, however a message was shown that it was sent.

						The original purpose of this function was a display page of all the parts but this is too vast therefore the requirements was changed to an inquiry page instead.
Case 14: Check if admin can add information from database		Req #5	Account Created	Account Created	Pass	While a proper UI is not implemented, the user is still able to add a

						vehicle, customer, appointments and user entries to the database from phpmyAdmin.
Case 15: Check if admin can update the information of a user such as their first name, last name, user type, email, phone number	Req #6	Update successfully	Update successfully	Pass		While a proper UI is not implemented, the user is still able to update vehicles, customers, appointments and user entries

and password from database						to the database from phpmyAdmin.
Case 16: Check if user (admin) can delete customer accounts from database		Req # 7.4	Account deleted successfully	Account deleted successfully	Pass	While a proper UI is not implemented, the user is still able to delete vehicles, customers, appointments and user entries to the database from phpmyAdmin.

Case 17: Check if admin user can view all appointments and their appointment history.		Req #8.1	All Appointments Displayed	All Appointments Displayed	Pass	Admin is able to view all the appointments entered into the system from phpmyAdmin database, however it is not reflected on the website.
Case 18: Check if admin user can search for appointments by licence plate number and date		Req #8.2	Filtered successfully	Filtered successfully	Pass	Admin is able to search for a license plate number and access their appointments

from website and database						from phpmyAdmin and the website
Case 19: Check if admin user is able to change the date and time of the appointment from database		Req #8	Updated successfully	Updated successfully	Pass	Admin is able to edit all the appointments entered into the system from phpmyAdmin
Case 20: Check if admin user is able to generate a report of a customer's and their vehicle's		Req #8	Report generated successfully	Report generated successfully but does not show the respective customer's report	Fail	The admin is able to view the vehicle and its plate number, as well as the owner's

information and previous services done to their vehicle(s)						information, their previous appointments, the date the appointment was added and the service chosen. However at the moment, the report does not reflect the vehicle that was selected.
--	--	--	--	--	--	--

## 6 Appendix

During our requirements elicitation process, Brianna Beharrie was the representative that interviewed the Managing Director and owner of Gramac Auto Repair Shop. The business is located at 21 Montgomery Avenue, Kingston 10. The director, Ingid Graham McFarlane, was interviewed via WhatsApp chat on September 25, 2023. Our ideas for the project were shared and the director was inquired about the operations of the business and the issues they faced. The interview enabled us to acquire valuable information from the stakeholders and provided insight so we could design our project to meet their needs. The interview log is provided below and summarises the conversation held between both parties.

**Brianna:** Good morning, could you provide this information for us whenever you're free please? So far, we have some ideas like having customers make appointments online and filling in the kind of work they would like to be done on their car, along with a tracking system to keep up with all the appointments and remind the customers when they can pick up their car after the work is complete. We also think a small inventory of some car parts that would be used on the cars could be beneficial. There could be an admin section for workers to input their information, but we'd like to know more so we can create something that's well suited for your business.

**Director:** Those are good. There could be a section where they can see the actual cost of the parts, so u have 3 categories of parts: Japanese, Genuine parts, Non genuine. So those could be in it, where the customer would say that the car is coming in for service to which we would input in

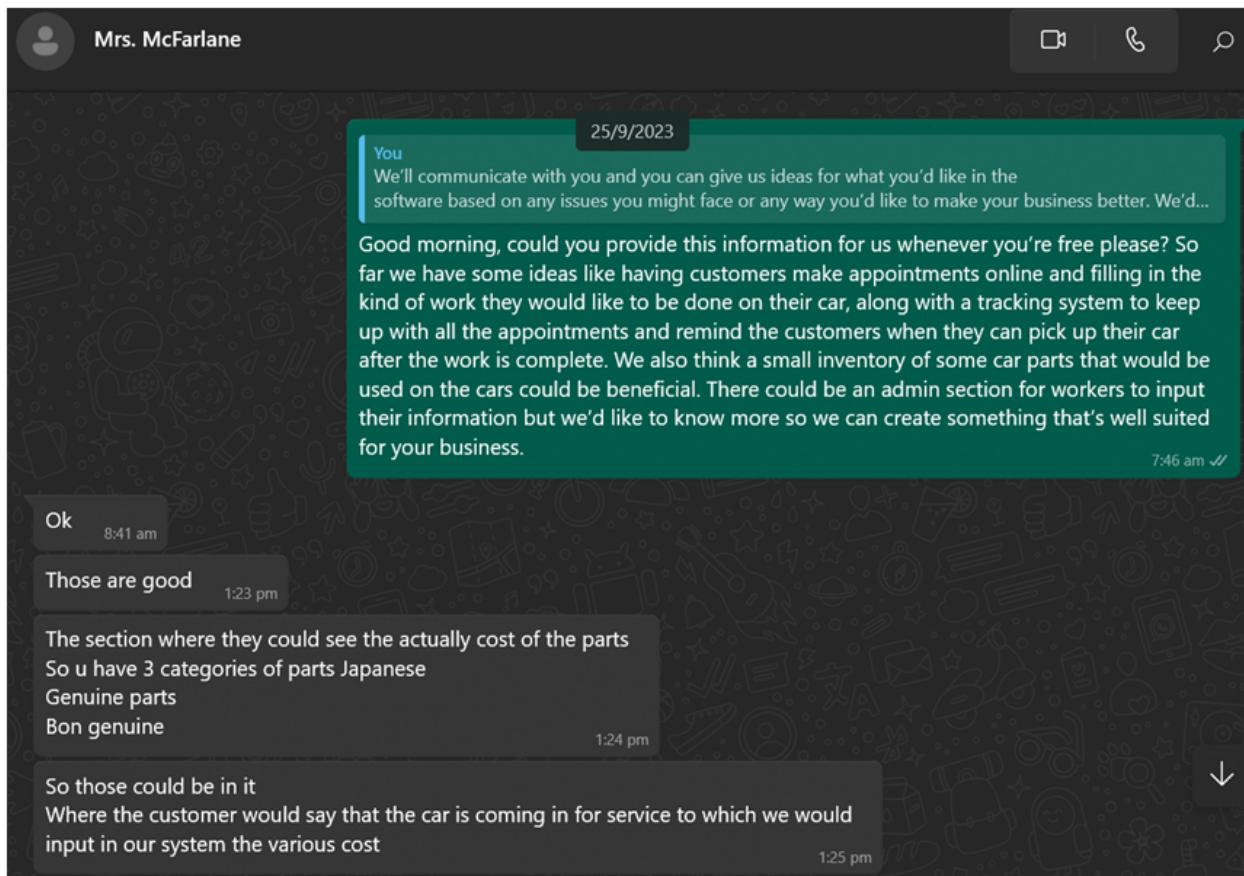
our system the various cost. This will allow the customer to select what they require, and we would use from that.

**Director:** Another is that they could put a brief description of the issue and it would give various problems associated with those symptoms and even when there is a light on the dash, they could send those from we could send feedback of what it could be. For those it would entail a cost. For the bookings it would be just to schedule.

**Director:** They could even generate their own invoice base on what is online. Also, we want the programto be that we the company create a car detail for all customers that way they can log in with a unique login to see what has been done over the years and what parts have changed as well as reminders about: Insurance, Fitness, when to service.

**Brianna:** So, you want the cost to automatically come up or for the mechanic to input it based on what they want? Also, regarding the input of the description of the issue could be in a chat assistance section. For the invoice generation, we are able to do that, but payments will have to be in person since payment processing is quite difficult to do.

**Director:** Yes, basically so u would know what it is beforehand, and it would be difficult.

**Screenshots of the conversation:**

This will allow the customer to select what they require and we would use from that 1:25 pm

Another is that they could put a brief description of the issue and it would give various problems associated with that symptoms and even when there is a light on the dash they could send those from we could send feed back of what it could be 1:27 pm

But dor those it would entail a cost 1:27 pm

But for bookings it would be just to schedule 1:27 pm

They could even generate their own invoice base on what is online 1:28 pm

Also the programs to be that we the company create a car details for all customers that way they can log in with a unique login to see what has been done over the years and what parts have chnage as well as reminders about Insurance  
Fitness  
When to service 1:30 pm

**Mrs. McFarlane**  
The section where they could see the actually cost of the parts  
So u have 3 categories of parts Japanese... 1:47 pm ↴

Okay 1:47 pm ↴

**Mrs. McFarlane**  
So those could be in it  
Where the customer would say that the car is coming in for service to which we would input in...  
So you want the cost to automatically come up or for the mechanic to input it based on what they want? 1:49 pm ↴

Yes basically so u would know what it is before hand 1:49 pm

**Mrs. McFarlane**  
Another is that they could put a brief description of the issue and it would give various problems associated with that symptoms and even when there is a light...  
Oh maybe this could be in a chat assistance section 1:50 pm ↴

Yes 1:50 pm

**Mrs. McFarlane**  
They could even generate their own invoice base on what is online  
Yes we can do that but the payment would be in person since payment processing is quite difficult to do 1:51 p ↴

Yes it would 1:51 pm

Screenshots of the Test Cases:

Test Case 1:

The screenshot shows the 'Register' page. At the top, there is a navigation bar with the 'GramacAuto' logo on the left and 'Login'、'Register'、'Contact Us' links on the right. The main title 'Register' is centered at the top. Below the title are six input fields arranged vertically. The first two fields are grouped together with a separator line. The third field is grouped with the fourth. The fifth field is grouped with the sixth. Each group has a rounded rectangular border. The first field contains 'Bob', the second 'Brown'. The third field contains 'Bob.brown@gmail.com'. The fourth field contains '8767573450'. The fifth field contains 'Iamthebob'. The sixth field contains '.....'. Below these fields is a large red rectangular button with the word 'Register' in white. At the bottom of the page, there is a small link 'I already have an account'.

The screenshot shows the 'Login' page. At the top, there is a navigation bar with the 'GramacAuto' logo on the left and 'Login'、'Register'、'Contact Us' links on the right. The main title 'Login' is centered at the top. Above the input fields, there is a message 'Account successfully created'. Below the message are two input fields: one for 'Enter Email Address' and one for 'Enter Password', both with rounded rectangular borders. Below these fields is a large red rectangular button with the word 'Login' in white. At the bottom of the page, there is a small link 'I do not have an account'.

Test Case 2:

GramacAuto

Login Register Contact Us

Bob Brown

Bob.brown@gmail.com

Email has already been used

8767573450

Iamthebob

Confirm Password

Register

I already have an account

Test Case 3:

GramacAuto

Login Register Contact Us

Jane Doe

janedoe@yahoo.com

8765559999

JAneydoit?

.....

Register

I already have an account

## Register

Jane Doe

janedoe@yahoo.com

8765559999

JAneydoit?

.....

Register

I already have an account

# Register

Jane Doe

janedoe@yahoo.com

Email is not valid

8765559999

JaneYdoit?

Confirm Password

**Register**

[I already have an account](#)

## Test Case 4:

Your Information		Vehicle Information	
Bob		2001	
Brown		Toyota	
Bob.brown@gmail.com		RAV4	
8767573450		Burgundy, black interior, dent in the right side door	
		7865BO	
<b>Review</b>			
<b>Submit</b>			

	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	<b>id</b>	<b>user_id</b>	<b>vehicle_plate_number</b>	<b>year</b>	<b>make</b>	<b>model</b>	<b>add_model</b>	<b>processed</b>	<b>in_shop</b>	<b>date_added</b>
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	1	6	J6SJ5R	2023	Honda	Civic	Type-R	no	no	2023-11-23
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	2	6	ABC123	2012	Toyota	Wish	base	no	no	2023-11-23
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	3	6	B4J11K	2006	Mazda	Speed3	Touring	no	no	2023-11-23
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	4	7	WASH99	2016	Toyota	GT86	The lights are too dim and sometimes flicker	no	no	2023-11-30
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	6	10	OPKA55	2022	Honda	Accord	Black	no	no	2023-11-30
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	7	11	7865BO	2001	Toyota	RAV4	Burgundy, black interior, dent in the right side d...	no	no	2023-11-30

## Test Case 5:

Your Information

Vehicle Information

Name: Nate	Year: Enter Year
Address: Richards	Make: Enter Make
Email: NateR01@gmail.com	Model: Enter Model
Phone: 8762345678	Additional Info: Enter Additional Model Info
	Plate Number: Enter Plate number

Review

Submit

## Test Case 6:

GramacAuto

Login Register Contact Us

# Login

[I do not have an account](#)

## Find Critical Vehicle Information

Welcome to Gramac Auto Repair and Service Limited, where excellence is our standard. Utilize our website to schedule appointments, purchase genuine parts online, and receive detailed reports when necessary. Our dedicated professionals ensure top-notch service, guaranteeing your vehicle's longevity and performance. Drive confidently, knowing your vehicle is in the hands of professionals dedicated to quality service.



### Test Case 7:

GramacAuto

Login Register Contact Us

# Login

[I do not have an account](#)

The screenshot shows the login interface for Gramac Auto Online. At the top, there is a dark header bar with the text "GramacAuto" on the left and "Login Register Contact Us" on the right. Below the header, the word "Login" is centered in a large, bold, black font. Underneath the title are two input fields: the first is labeled "mechanicnumber1@example.com" and the second is labeled "Enter Password". Below these fields is a red "Login" button with white text. At the bottom of the form, there is a link underlined in blue that reads "I do not have an account".

Test Case 8:

This screenshot is identical to the one above it, showing the Gramac Auto Online login page. It features the same header, "Login" title, input fields for email and password, the red "Login" button, and the "I do not have an account" link. However, the email input field now contains the text "bob.brown@example.com" and the password input field contains "\*\*\*\*\*".

GramacAuto

Login Register Contact Us

# Login

The image shows a login form with two input fields and a red "Login" button. The first field contains the placeholder text "bob.brown@example.com". The second field is labeled "Enter Password". Below the fields is a red "Login" button with white text.

bob.brown@example.com

Enter Password

Login

I do not have an account

Test Case 9:

GramacAuto

Login Register Contact Us

# Login

The image shows a login form with two input fields and a red "Login" button. The first field is labeled "Enter Email Address". The second field is labeled "Enter Password". Below the fields is a red "Login" button with white text.

Enter Email Address

Enter Password

Login

I do not have an account

## Test Case 10

GramacAuto      Home Contact Us Parts Inquiry      Hi, Jada Logout

1 Appointment Details    2 My Information    3 Review & Submit

**Preferred Date of Request**

05/12/2023

**Preferred Date of Request**

08:00AM - 10:00AM  
10:00AM - 12:00AM  
12:00AM - 02:00AM  
02:00AM - 04:00AM  
04:00AM - 05:00AM

**Vehicle Drop off?**

Drop Off     Wait For

**Select Service(s)**

Alignment     Battery     Exhaust     Oil Change     Tires  
 Brakes     Suspension     Transmission     Engine     Heating & Cooling  
 Interior     Exterior     Electrical     Accessories     Body Work  
 Glass     Lights     Paint     Wheels

GramacAuto      Home Contact Us Parts Inquiry      Hi, Jada Logout

1 Appointment Details    2 My Information    3 Review & Submit

**Your Information**

Jada  
Walters  
jada@gmail.com  
8761112222

**Vehicle Information**

2015  
Toyota  
GT86  
White paint  
1234AB

GramacAuto

Home Contact Us Parts Inquiry

Hi, Jada Logout

**1 Appointment Details**    **2 My Information**    **3 Review & Submit**

---

<b>Your Information</b>	<b>Vehicle Information</b>
Jada Walters jada@gmail.com 8761112222	2015 Toyota GT86 White paint 1234AB
<b>Date &amp; Time</b>	<b>Service(s) Needed</b>
2023-12-05 08:00AM - 10:00AM	Oil Change    Brakes

**Back**    **Submit**

## Test Case 11

GramacAuto      Home Contact Us Parts Inquiry      Hi, Jada Logout

**1 Appointment Details**      **2 My Information**      **3 Review & Submit**

---

**Preferred Date of Request**

**Preferred Date of Request**

**Vehicle Drop off?**

Drop Off     Wait For

**Select Service(s)**

<input type="checkbox"/> Alignment	<input type="checkbox"/> Battery	<input type="checkbox"/> Exhaust	<input type="checkbox"/> Oil Change	<input type="checkbox"/> Tires
<input type="checkbox"/> Brakes	<input type="checkbox"/> Suspension	<input type="checkbox"/> Transmission	<input type="checkbox"/> Engine	<input type="checkbox"/> Heating & Cooling
<input type="checkbox"/> Interior	<input type="checkbox"/> Exterior	<input type="checkbox"/> Electrical	<input type="checkbox"/> Accessories	<input type="checkbox"/> Body Work
<input type="checkbox"/> Glass	<input type="checkbox"/> Lights	<input type="checkbox"/> Paint	<input type="checkbox"/> Wheels	

GramacAuto      Home Contact Us Parts Inquiry      Hi, Jada Logout

**1 Appointment Details**      **2 My Information**      **3 Review & Submit**

---

<b>Your Information</b>	<b>Vehicle Information</b>
Jada Walters jada@gmail.com 8761112222	Vehicle year required Vehicle make required Vehicle model required Additional model info required Vehicle licence plate number required
<b>Date &amp; Time</b>	<b>Service(s) Needed</b>
Date selection is required Date selection is required	Service(s) selection is required

Test Case 12

The screenshot shows a web-based application for booking an appointment at GramacAuto. The top navigation bar includes links for Home, Contact Us, Parts Inquiry, and a user session (Hi, Jada | Logout). Below the navigation, a horizontal navigation bar indicates the current step: 'Appointment Details' (marked with a red circle), 'My Information' (green circle), and 'Review & Submit' (blue circle).

**Your Information:**  
Jada Walters  
jada@gmail.com  
8761112222

**Vehicle Information:**  
2015  
Toyota GT86  
White paint  
1234AB

**Date & Time:**  
2023-12-05  
08:00AM - 10:00AM

**Service(s) Needed:**  
 Oil Change  
 Brakes

**Action Buttons:**  
Back (left) | Submit (right)

Test Case 13



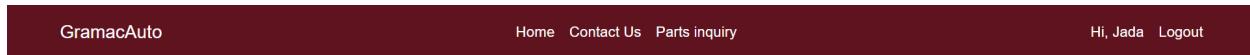
**Need a Part? Ask us!:**

John Kennedy

john.kennedy@po.com

Good day. Do you have engine for 2014 Mazda 5

**Submit**



## Test Case 14

Screenshot of MySQL Workbench showing the execution of an SQL query and its results.

**Query Execution:**

```
INSERT INTO `vehicles` (`id`, `user_id`, `vehicle_plate_number`, `year`, `make`, `model`, `add_model`, `processed`, `in_shop`, `date_added`) VALUES ('5', '8', '9874BB', '2018', 'Toyota', 'Crown', 'red paint', 'no', 'no', '2023-12-01');
```

**Table Structure:**

id	user_id	vehicle_plate_number	year	make	model	add_model	processed	in_shop	date_added
----	---------	----------------------	------	------	-------	-----------	-----------	---------	------------

**Result Set:**

id	user_id	vehicle_plate_number	year	make	model	add_model	processed	in_shop	date_added
5	8	9874BB	2018	Toyota	Crown	red paint	no	no	2023-12-01

**Console Output:**

```
Showing rows 0 - 4 (5 total, Query took 0.0013 seconds.)
```

```
SELECT * FROM `vehicles`
```

Screenshot of MySQL Workbench showing the 'appointments' table.

**Top Panel:**

- Server: MySQL:3306 » Database: gramac\_db » Table: appointments
- Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, Triggers tabs.
- Message: 1 row inserted.
- SQL Query: `INSERT INTO `appointments` (`id`, `vehicle_plate_number`, `appointment_date`, `time`, `drop_off`, `date_added`) VALUES ('5', '9874BB', '2023-12-13', '08:00AM - 10:00AM', 'drop off', '2023-12-01');`
- Buttons: [Edit inline], [Edit], [Create PHP code].

**Bottom Panel:**

- Run SQL query/queries on table gramac\_db.appointments:
- SQL Editor: `1 INSERT INTO `appointments` (`id`, `vehicle_plate_number`, `appointment_date`, `time`, `drop_off`, `date_added`) VALUES ('5', '9874BB', '2023-12-13', '08:00AM - 10:00AM', 'drop off', '2023-12-01');`
- Table Definition: 

<code>id</code>	<code>vehicle_plate_number</code>	<code>appointment_date</code>	<code>time</code>	<code>drop_off</code>	<code>date_added</code>
-----------------	-----------------------------------	-------------------------------	-------------------	-----------------------	-------------------------
- Action Buttons: SELECT\*, SELECT, INSERT, UPDATE, DELETE, Clear, Format, Get auto-saved query.
- Checkboxes: Bind parameters, Delimiter: ;, Show this query here again, Retain query box, Rollback when finished, Enable foreign key checks, Go button.

**Second Screenshot:**

Showing rows 0 - 4 (5 total, Query took 0.00006 seconds.)

SELECT \* FROM `appointments`

Query results operations: Print, Copy to clipboard, Export, Display chart, Create view.

	<code>id</code>	<code>vehicle_plate_number</code>	<code>appointment_date</code>	<code>time</code>	<code>drop_off</code>	<code>date_added</code>
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	1 J6SJ5R	2023-11-09	08:00AM - 10:00AM	drop off	2023-11-23
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	2 ABC123	2023-11-08	12:00AM - 02:00AM	wait for	2023-11-23
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	3 B4J11K	2023-11-09	08:00AM - 10:00AM	wait for	2023-11-23
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	4 1234AB	2023-12-05	08:00AM - 10:00AM	drop off	2023-12-01
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	5 9874BB	2023-12-13	08:00AM - 10:00AM	drop off	2023-12-01

Action Buttons: Show all, Number of rows: 25, Filter rows: Search this table, Sort by key: None.

Extra options: Check all, With selected: Edit, Copy, Delete, Export.

Action Buttons: Show all, Number of rows: 25, Filter rows: Search this table, Sort by key: None.

## Test Case 15

Screenshot of MySQL Workbench showing the 'users' table.

**Table Structure:**

	<b>id</b>	<b>first_name</b>	<b>last_name</b>	<b>user_type</b>	<b>email</b>	<b>phone_number</b>	<b>password</b>	<b>date_created</b>
<input type="checkbox"/>	1	Mechanic	Number1	admin	mechanicnumber1@example.com	8760000000	\$2y\$10\$XoN9oYG9/R7Zrlyn0T0St.9dxxylY7C6TWqq09lCo...	2023-11-22
<input type="checkbox"/>	2	Mechanic	Number2	admin	mechanicnumber2@example.com	8760000001	\$2y\$10\$6EGCNLY5NZhkVLtg0HLB.2643gkvFgKRqJbMCkB1...	2023-11-22
<input type="checkbox"/>	3	Mechanic	Number3	admin	mechanicnumber3@example.com	8760000002	\$2y\$10\$zEsLZ3tv2lSH6v//POPQesq1ppif.QWyYelbxef7F7...	2023-11-22
<input type="checkbox"/>	4	Mechanic	Number4	admin	mechanicnumber4@example.com	8760000003	\$2y\$10\$c/O0xHPCGoekJqLM97.g2OtersJY6pYr6P0Z1sOZe.S...	2023-11-22
<input type="checkbox"/>	5	Mechanic	Number5	admin	mechanicnumber5@example.com	8760000004	\$2y\$10\$BpV24almGBw9brqQ4qpJqOk8X8ZoiQRKz5irgL3brX...	2023-11-22
<input checked="" type="checkbox"/>	6	Saaeen	Grant	customer	asdadsds@gmail.com	8761112222	\$2y\$10\$pIsQD8CTIL/eIeatKePAOlw5zlRh3m4AzamkHIdzld...	2023-11-23
<input type="checkbox"/>	7	Jada	Walters	customer	jada@gmail.com	8761112222	\$2y\$10\$n0v4G0oKM4iXKlinncD6GuqSfcgmRL0R.GR1lZyscbJ...	2023-12-01
<input type="checkbox"/>	8	Brianna	Beharie	customer	bribri@gmail.com	8769998888	bowbowbow	2023-12-01

**Query Results:**

```
UPDATE `users` SET `email` = 'saaeen@gmail.com', `password` = 'gibberishGibberish' WHERE `users`.`id` = 6;
```

**Console Output:**

```
1 row affected.
```

## Test Case 16

Screenshot of MySQL Workbench showing two sessions for the 'users' table.

**Session 1 (Top):**

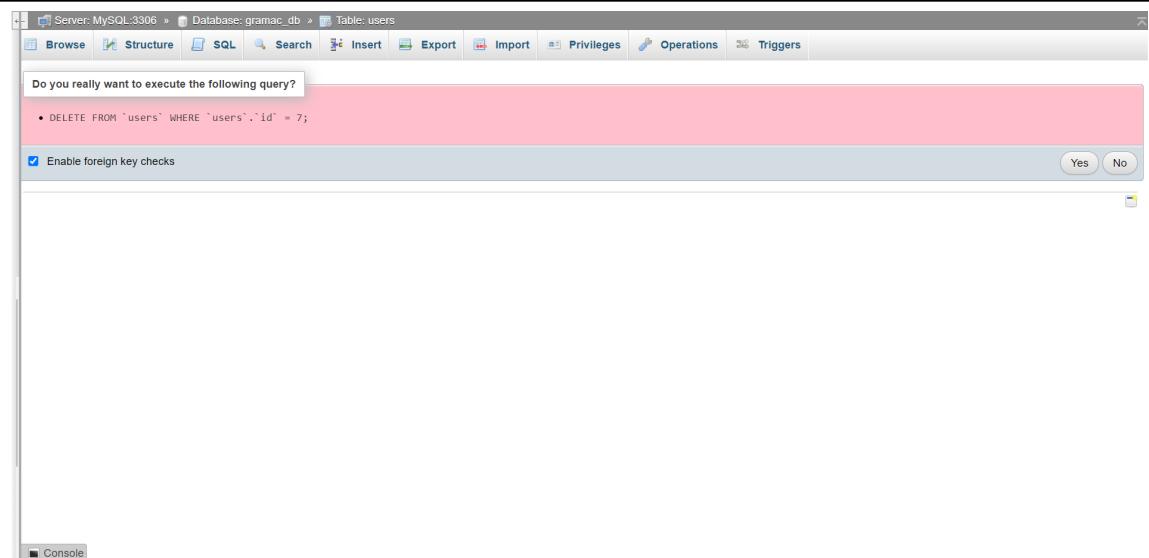
- Showing rows 0 - 7 (8 total, Query took 0.0004 seconds.)
- SQL query: `SELECT * FROM `users``
- Table data:

	<b>id</b>	<b>first_name</b>	<b>last_name</b>	<b>user_type</b>	<b>email</b>	<b>phone_number</b>	<b>password</b>	<b>date_created</b>
<input type="checkbox"/>	1	Mechanic	Number1	admin	mechanicnumber1@example.com	8760000000	\$2y\$10\$XoN9oYG9/R7Znyn0T0St.9dxyyyY7C6TWqq09i5lCo...	2023-11-22
<input type="checkbox"/>	2	Mechanic	Number2	admin	mechanicnumber2@example.com	8760000001	\$2y\$10\$6EGCNLY5NZHKVLtg0HLB.2643gkvFgKRqubMCKBi2...	2023-11-22
<input type="checkbox"/>	3	Mechanic	Number3	admin	mechanicnumber3@example.com	8760000002	\$2y\$10\$zEslZ3tV2ISH6v0/POPQesq1ppif.QWYyElbxef7F...	2023-11-22
<input type="checkbox"/>	4	Mechanic	Number4	admin	mechanicnumber4@example.com	8760000003	\$2y\$10\$c0OxHPCGoekJdLM97.g2OtersJY6pYr6P0Z1sOZe.S...	2023-11-22
<input type="checkbox"/>	5	Mechanic	Number5	admin	mechanicnumber5@example.com	8760000004	\$2y\$10\$BpV24almGBw9brqQ4qpJqOk8X8Xz0jQRkz5irgl3brX...	2023-11-22
<input checked="" type="checkbox"/>	6	Saaeen	Grant	customer	saaeen@gmail.com	8761112222	gibberishGibberish	2023-11-23
<input checked="" type="checkbox"/>	7	Jada	Walters	customer	jada@gmail.com	8761112222	\$2y\$10\$n0v4G0oKM4XKlinncD6GuqSfcgmRL0R.GR1ZyscbJ...	2023-12-01
<input type="checkbox"/>	8	Brianna	Beharie	customer	bribri@gmail.com	8769998888	bowbowbow	2023-12-01

**Session 2 (Bottom):**

- Showing rows 0 - 6 (7 total, Query took 0.0004 seconds.)
- SQL query: `SELECT * FROM `users``
- Table data:

	<b>id</b>	<b>first_name</b>	<b>last_name</b>	<b>user_type</b>	<b>email</b>	<b>phone_number</b>	<b>password</b>	<b>date_created</b>
<input type="checkbox"/>	1	Mechanic	Number1	admin	mechanicnumber1@example.com	8760000000	\$2y\$10\$XoN9oYG9/R7Znyn0T0St.9dxyyyY7C6TWqq09i5lCo...	2023-11-22
<input type="checkbox"/>	2	Mechanic	Number2	admin	mechanicnumber2@example.com	8760000001	\$2y\$10\$6EGCNLY5NZHKVLtg0HLB.2643gkvFgKRqubMCKBi2...	2023-11-22
<input type="checkbox"/>	3	Mechanic	Number3	admin	mechanicnumber3@example.com	8760000002	\$2y\$10\$zEslZ3tV2ISH6v0/POPQesq1ppif.QWYyElbxef7F...	2023-11-22
<input type="checkbox"/>	4	Mechanic	Number4	admin	mechanicnumber4@example.com	8760000003	\$2y\$10\$c0OxHPCGoekJdLM97.g2OtersJY6pYr6P0Z1sOZe.S...	2023-11-22
<input type="checkbox"/>	5	Mechanic	Number5	admin	mechanicnumber5@example.com	8760000004	\$2y\$10\$BpV24almGBw9brqQ4qpJqOk8X8Xz0jQRkz5irgl3brX...	2023-11-22
<input type="checkbox"/>	6	Saaeen	Grant	customer	saaeen@gmail.com	8761112222	gibberishGibberish	2023-11-23
<input type="checkbox"/>	8	Brianna	Beharie	customer	bribri@gmail.com	8769998888	bowbowbow	2023-12-01



## Test Case 17

A screenshot of the GramacAuto web application. The header includes "GramacAuto", "Mechanic Number", and "Logout". A sidebar on the left lists "Appointments", "Jobs", "Vehicles", and "Home". The main content area is titled "Appointments" and displays a table of appointment records:

#	vehicle_plate_number	appointment_date	time	drop_off	date_added	related_job
1	J6SJ5R	2023-11-09	08:00AM - 10:00AM	drop off	2023-11-23	<a href="#">Jobs</a>
2	B4J1K	2023-11-09	08:00AM - 10:00AM	wait for	2023-11-23	<a href="#">Jobs</a>

The screenshot shows the phpMyAdmin interface for the 'gramac\_db' database. The 'appointments' table is selected, displaying 5 rows of data. The columns are: id, vehicle\_plate\_number, appointment\_date, time, drop\_off, and date\_added. The data is as follows:

	id	vehicle_plate_number	appointment_date	time	drop_off	date_added
<input type="checkbox"/>	1	J6SJ5R	2023-11-09	08:00AM - 10:00AM	drop off	2023-11-23
<input type="checkbox"/>	3	B4J11K	2023-11-09	08:00AM - 10:00AM	wait for	2023-11-23
<input type="checkbox"/>	5	1299VT	2023-12-08	10:00AM - 12:00AM	drop off	2023-12-01
<input type="checkbox"/>	2	ABC123	2023-11-08	12:00AM - 02:00AM	wait for	2023-11-23
<input type="checkbox"/>	4	7890RT	2023-12-08	12:00AM - 02:00AM	drop off	2023-12-01

## Test Case 18

The screenshot shows the GramacAuto web application. The user is logged in as Mechanic Number1. The main menu on the left includes Appointments, Jobs, Vehicles, and Home. The current page is 'Vehicle'. A search bar at the top right contains 'B4J11K' and a 'Lookup' button. The main content area displays a table of vehicles:

#	user_id	vehicle_plate_number	year	year	model	add_model	processed	in_shop	date_added	vehicle_report
1	6	J6SJ5R	2023	Honda	Civic	Type-R	no	no	2023-11-23	<button>See report</button>
2	6	ABC123	2012	Toyota	Wish	base	no	no	2023-11-23	<button>See report</button>
3	6	B4J11K	2006	Mazda	Speed3	Touring	no	no	2023-11-23	<button>See report</button>
4	1	7890RT	2001	Toyota	RAV4	Blue exterior with scratches on the left front door	no	no	2023-12-01	<button>See report</button>
5	1	1299VT	2003	Suzuki	Vitara	Grey model	no	no	2023-12-01	<button>See report</button>

The screenshot shows the Gramac Auto Online application interface. At the top, there is a red header bar with the text "GramacAuto". On the right side of the header, there are links for "Mechanic Number" and "Logout". Below the header, on the left, is a sidebar with navigation links: "Appointments", "Jobs", "Vehicles", and "Home". The main content area has a title "Vehicle" and a search bar with a "Lookup" button. Below the search bar is a table with the following data:

#	user_id	vehicle_plate_number	year	year	model	add_model	processed	in_shop	date_added	vehicle_report
1	6	B4J11K	2006	Mazda	Speed3	Touring	no	no	2023-11-23	<a href="#">See report</a>

The screenshot shows the phpMyAdmin interface connected to a MySQL server at 127.0.0.1. The database selected is "gramac\_db" and the table is "appointments". The top navigation bar includes "Browse", "Structure", "SQL", "Search", "Insert", "Export", "Import", "Privileges", "Operations", "Tracking", and "Triggers". The main content area displays the following SQL query results:

```

SELECT * FROM `appointments` ORDER BY `time` ASC

```

Showing rows 0 - 4 (5 total, Query took 0.0003 seconds.) [time: 08:00AM - 10:00AM... - 12:00AM - 02:00AM...]

The table data is as follows:

	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	<a href="#">Insert</a>	<a href="#">Export</a>	<a href="#">Import</a>	<a href="#">Privileges</a>	<a href="#">Operations</a>	<a href="#">Tracking</a>	<a href="#">Triggers</a>
#	id	vehicle_plate_number	appointment_date	time	drop_off	date_added				
1	1	B4J11K	2023-11-09	08:00AM - 10:00AM	wait for	2023-11-23	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	<a href="#">Insert</a>
2	2	B4J11K	2023-11-09	08:00AM - 10:00AM	wait for	2023-11-23	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	<a href="#">Insert</a>
3	3	B4J11K	2023-11-09	08:00AM - 10:00AM	wait for	2023-11-23	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	<a href="#">Insert</a>

## Test Case 19

Showing rows 0 - 4 (5 total, Query took 0.0004 seconds) [time: 08:00AM - 10:00AM... - 12:00AM - 02:00AM...]

```
SELECT * FROM `appointments` ORDER BY `time` ASC
```

	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	<a href="#">1</a> J6SJ5R	2023-11-09	08:00AM - 10:00AM	drop off	2023-11-23
	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	<a href="#">3</a> B4J11K	2023-11-09	08:00AM - 10:00AM	wait for	2023-11-23

## Test 20

Mechanic Number1 [Logout](#)

#	user_id	vehicle_plate_number	year	year	model	add_model	processed	in_shop	date_added	vehicle_report
1	6	J6SJ5R	2023	Honda	Civic	Type-R	no	no	2023-11-23	<a href="#">See report</a>
2	6	ABC123	2012	Toyota	Wish	base	no	no	2023-11-23	<a href="#">See report</a>
3	6	B4J11K	2006	Mazda	Speed3	Touring	no	no	2023-11-23	<a href="#">See report</a>
4	1	7890RT	2001	Toyota	RAV4	Blue exterior with scratches on the left front door	no	no	2023-12-01	<a href="#">See report</a>
5	1	1299VT	2003	Suzuki	Vitara	Grey model	no	no	2023-12-01	<a href="#">See report</a>

The screenshot shows a software application window titled "GramacAuto". The top navigation bar includes links for "Mechanic Number" and "Logout". The left sidebar contains links for "Appointments", "Jobs", "Vehicles", and "Home". The main content area displays a "Vehicle report" for a "Honda Civic Type-R" with plate number "J6SJ5R". Below the vehicle details, there is a section for "User/Owner Info" with the following contact information:

id: 6  
Name: Saaeen Grant  
Phone: (876) 111 - 2222  
Email: asdadss@gmail.com

Below this is a section for "Appointment Info" with the following details:

Id: 1  
Date: 2023-11-09  
Time: 08:00AM - 10:00AM  
Drop off action: drop off

At the bottom of the main content area, there is a small note: "RA- 1".