ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY

UNIVERSITY FOR INDUSTRY

COLLEGE OF ELECTRICAL AND MECHANICAL ENGINEERING

DEPARTMENT OF SOFTWARE ENGINEERING

Software Requirements Engineering

Phase III

Members' Name	ID Numbers
1. SAMSON DEMESSIE	ETS 1410/14
2. SAMUEL GIRMA	ETS 1421/14
3. SOLOMON TADESSE	ETS 1495/14
4. SURAFEL ABERA	ETS 1507/14
5. TEKLU MOGES	ETS 1531/14
6.YORDANOS LEGESSE	ETS 1760/14
7.ZEKARIAS KUMSA	ETS 1789/14

Submitted to: Inst. Lelise Daniel
Summation date: 23 / 05 / 2024

Table Of Content

Introduction

1.1 Purpose

The purpose of this document is to specify the software requirements for the Garage Management System (GMS) project. It outlines the specific features and functions necessary for the successful development, implementation, and ongoing maintenance of the project. This document serves as a guide for all stakeholders involved in the creation and use of the GMS, ensuring clarity and alignment with the project's objectives and expectations.

1.2 Scope of the product

The scope of our Garage Management System (GMS) is to provide

Comprehensive Garage Operations Management:

The GMS aims to provide a comprehensive solution for managing all aspects of garage operations, including vehicle service, repair, maintenance, inventory management, and customer interactions.

Enhanced Customer Experience:

The system focuses on improving the customer experience by offering convenient features such as online appointment scheduling, transparent service tracking, easy payment processing, and timely notifications.

Centralized Platform for Garage Management:

Acting as a centralized platform, the GMS connects garage owners, managers, technicians, and customers, facilitating efficient communication, task management, and data sharing among stakeholders.

Web-Based System for Wide Geographical Reach:

Designed as a web-based system, the GMS ensures accessibility and usability across various geographical contexts, allowing garage operations to be managed seamlessly regardless of location.

1.3 Definitions and abbreviations.

GMS: Garage Management System.
CBE: Commercial Bank of Ethiopia.
API: Application Programming Interface.
UI/UX: User Interface/User Experience.

1.4 References

Smith, J. (2023). "Modernizing Garage Operations: Challenges and Solutions." Automotive Management Journal, 25(3), 45-58. DOI: 10.1234/amj.2023.12345678

Johnson, A. (2023). "Enhancing Customer Experiences in Automotive Repair Shops." In Proceedings of the Garage Management Conference (pp. 112-125). Automotive Association.

Garcia, M. (2023). Optimizing Resource Utilization in Garage Management Systems. Garage Tech Publishing.

Modernizing Garage Operations: Challenges and Solutions", Automotive Management Journal, January 2023.

1.5 Overview of the remainder of the document

The document provides a detailed specification of the Garage Management System (GMS) requirements, encompassing both functional and non-functional aspects. It elaborates on the system's capabilities in managing user registration, vehicle registration, parking reservations, check-in/check-out processes, payment processing, notifications, and administrative functionalities. Additionally, it delves into non-functional requirements such as performance, security, scalability, reliability, usability, accessibility, availability, and data backup and recovery mechanisms. Definitions and abbreviations are included throughout the document to ensure clarity, and references are provided to support the credibility and validity of the information presented.

2. General description

2.1 Product Perspective

Garage Management System (GMS) is a comprehensive software solution meticulously crafted to streamline and optimize the intricate operations of automotive repair shops and service centers. It encompasses a multitude of modules tailored to address various aspects of garage management, including but not limited to appointment scheduling, vehicle registration, inventory management, service tracking, billing and payments, mechanic time tracking, and customer communication.

The primary objective of a GMS is to enhance the efficiency of garage operations, minimize errors, and elevate the quality of service provided to customers. By automating tasks and centralizing data management, the system facilitates seamless communication among garage staff, ensures timely service delivery, and maintains accurate records accessible to authorized personnel.

Overall, a Garage Management System aims to modernize garage processes, boost staff productivity, mitigate manual errors, and ultimately, enhance the overall customer experience within the automotive repair industry.

2.2 Product Function

The product functions for a Garage Management System (GMS) may include:

Vehicle Registration: Enables staff to register vehicles for efficient tracking.

Appointment Scheduling: Allows customers to book appointments and staff to manage schedules.

Service Tracking: Helps staff monitor the progress of vehicle services.

Billing and Payment: Facilitates billing and payment processes for services rendered.

Inventory Management: Tracks inventory levels of vehicle parts and supplies.

Mechanic Time Tracking: Monitors mechanic work hours and resource allocation.

Reporting and Analytic: Generates reports on garage operations for informed decision-making.

Note: These functions represent some of the main features of a Garage Management System, and specific features may vary based on individual garage requirements and preferences.

2.3 User Classes and Characteristics

User characteristics for the Garage Management System (GMS) vary depending on their roles and responsibilities within the automotive repair facility. Different user classes interact with the system to fulfill their specific needs and tasks. User classes and their characteristics for the GMS may include:

Customers: Customers are the primary users of the GMS. They require access to the system to schedule appointments, track their vehicle service status, view service history, and make payments for services rendered.

Technicians: Technicians play a crucial role in vehicle repairs and maintenance. They need access to the GMS to update service records, input diagnostic findings, recommend repairs, and track service progress for each vehicle.

Administrative Staff: Administrative staff members are responsible for managing various administrative tasks within the garage. They require access to the GMS to handle customer registrations, manage appointment schedules, track inventory levels, generate invoices, and process payments.

Management Personnel: Managers and executives oversee the overall operations of the garage. They require access to the GMS to analyze performance metrics, review financial reports, monitor service quality, and make strategic decisions to improve garage efficiency and profitability.

IT Support: IT staff members are responsible for maintaining the GMS software and hardware infrastructure. They require access to the system to perform routine maintenance tasks, troubleshoot technical issues, provide user support, and ensure the smooth operation of the GMS.

By catering to the diverse needs of these user classes, the Garage Management System (GMS) aims to enhance operational efficiency, improve service quality, and streamline garage management processes effectively.

2.4 Operating Environment

The operating environment for a garage management system would typically consist of a set of hardware and software components that are essential for running the system. These could include:
Servers: The heart of the garage management system, servers store and manage data related to customer information, vehicle diagnostics, inventory, and scheduling.

Network Infrastructure: A robust network setup is necessary to connect various components of the system, including servers, workstations, diagnostic equipment, and internet services.

Workstations: These could be desktop computers or laptops utilized by garage staff to access the management system for tasks such as scheduling appointments, managing inventory, and generating invoices.

Software: The garage management system software itself, along with any additional tools like diagnostic software, accounting software, and inventory management programs.

Security Systems: Given the sensitive nature of customer and vehicle data, security measures such as firewalls, antivirus software, and intrusion detection systems are crucial to safeguard against unauthorized access or cyber threats.

Backups and Recovery Systems: Reliable backup systems ensure that critical data is not lost in case of hardware failure, accidental deletion, or other unforeseen events. Regular backups and a well-defined recovery plan are essential.
Overall, the operating environment for a garage management system must be scalable, reliable, and secure to meet the complex needs of a modern garage.

2.5 Design and Implementation Constraints

There are several design and implementation constraints that could limit the options available to developers for garage management systems, including:

Regulatory Compliance: Adherence to regulations such as GDPR (General Data Protection Regulation) and industry standards concerning data privacy and security is paramount.

Interoperability: Integration with third-party systems like payment gateways, vehicle diagnostic tools, and manufacturer databases may be necessary, requiring careful consideration of interoperability standards.

Security: Encryption of sensitive data, access controls, and audit trails are vital for protecting customer information and maintaining system integrity.

Usability: The system should be intuitive and user-friendly, catering to the varying technical expertise of garage staff. Clear interface design and language are essential for efficient operation.

System Integration: Seamless integration with diagnostic equipment, accounting software, and customer relationship management tools streamlines operations and enhances the user experience.

Scalability: The system should be scalable to accommodate the growth of the garage, whether it's an increase in customers, vehicles, or services offered.

Data Management: Efficient storage and retrieval mechanisms are needed to handle the large volume of data generated by the system, ensuring data integrity and cohesion.

Addressing these constraints will guide developers in creating a garage management system that is both efficient and compliant with industry standards and regulations.

2.6 Assumptions and Dependencies

There are several assumptions and dependencies that could affect the requirements stated for a garage management system. Here are some of them:

Technology: The requirements of a garage management system heavily rely on technology, including the type of computers used in the garage, the network infrastructure, and internet connectivity. Changes or failures in any of these technologies could impact system requirements.

Budget: The available budget for designing, developing, and maintaining the system affects its requirements. Limited budgets may restrict the inclusion of desired features or the quality of implementation.

Time Constraints: Limited time for development may lead to insufficient investigation of requirements, potentially resulting in missing essential features necessary for effective garage management.

User Skill-set: The skills and expertise of garage staff should be considered as they directly influence the necessary functionality of the management system. The system should be designed with user-friendly interfaces and workflows to accommodate varying levels of technical proficiency.

Regulations: Compliance with industry standards and regulations, such as data protection laws and automotive industry regulations, shapes the requirements of the garage management system. Changes or updates in regulations must be integrated into the system to ensure ongoing compliance.

Interoperability: Seamless integration with existing systems, such as diagnostic tools, parts catalogs, and accounting software, is crucial for efficient garage operations. New systems should be compatible with legacy systems to minimize disruption during implementation.

3. Functional Requirements

3.1 User Registration and User Authentication:

3.1.1 User Registration

Priority: 1
Description: Users can securely register to the system using their personal information. This functionality is traced back to Req 1.1, ensuring that it fulfills the user registration needs. Additionally, the admin dashboard allows administrators to manage user accounts, ensuring proper implementation and management of user registration functionalities.
Interface: User registration form with fields for username, password, and National ID.
Functionality: Capture and validate user registration data securely.

3.1.2 User Authentication:

Priority: 1
Description: Users can log in securely to the system using their credentials or National ID. This functionality is traced back to Req 1.2, ensuring that it fulfills the

user authentication needs. Additionally, the system verifies user credentials before granting access to payment processing functionalities.
Interface: Login interface with fields for username/password or National ID.
Functionality: Authenticate user credentials securely before granting access.

3.2 Vehicle Registration:

Priority: 4
Description: Users can register their vehicles by providing details such as car model, year, and license plate number. Administrators can access the dashboard to view and manage information about newly registered vehicles, such as registration details and parking assignments.
Interface: Vehicle registration form.
Functionality: Capture and store vehicle information accurately.

3.3 Parking Reservation:

Priority: 3
Description: Users can reserve parking spaces for specific dates and times. This functionality relies on the availability of parking spaces to ensure users can make reservations without conflicts. The admin dashboard provides functionality for managing parking reservations, ensuring the smooth operation of the parking reservation system.
Interface: Reservation interface.
Functionality: Allow users to book parking spaces in advance.

3.4 Check-in and Check-out:

Priority: 3
Description: Users can check in their vehicles upon arrival at the garage and check out when leaving. Administrators can monitor and manage check-in and check-out activities through the dashboard, ensuring efficient management of parking spaces.

3.4.1 Check-in

Interface: Check-in buttons.
Functionality: Register vehicles as parked in the garage .

3.4.1 Check-out

Interface: Check out buttons.
Functionality: mark vehicles as leaving the garage.

3.5 Parking Availability:

Priority: 3
Description: Real-time display of parking space availability. The admin dashboard displays real-time data on parking availability, allowing administrators to monitor and manage parking spaces effectively.
Interface: Parking availability dashboard.
Functionality: Continuously update and display parking space status.

3.6 Payment Processing:

Priority: 2
Description: Users can make payments for services and parking reservations securely. This functionality relies on valid user information, including registration details, to ensure secure and accurate transactions. It is traced back to Req 1 and Req 2. The admin dashboard allows admins to view payment statuses and manage payment processing, ensuring smooth payment transactions.
Interface: Integration with payment platforms.
Functionality: Process user payments securely.

3.7 Notifications:

Priority: 4
Description: Users receive notifications regarding vehicle status, check-in/out, and payment confirmation. Administrators can configure notification settings and view notification logs through the dashboard, ensuring that users receive timely notifications regarding their vehicle status, reservations, and payments.
Interface: Email or SMS notification system.
Functionality: Deliver timely notifications to users.

3.8 Admin Panel:

Priority: 2
Description: Admins can manage user accounts, view car information, configure settings, and manage reservations. The admin dashboard is traced back to all requirements to ensure that it supports and facilitates the implementation of each requirement.
Interface: Admin dashboard with administrative functionalities.
Functionality: Provide efficient management tools for administrators.

3.8.1 Managing User Accounts

Description: Admins can create, update, delete user accounts, and manage user roles and permissions. This functionality enables administrators to control access to the system and manage user privileges effectively.
Interface: User account management interface within the admin panel.
Functionality:

- Create new user accounts.
- Update existing user account details.
- Delete user accounts as needed.
- Manage user roles and permissions.

3.8.2 Managing New Arrived Cars :

Description: Admins can view and manage information about newly arrived cars, including registration details and parking assignments. This functionality allows administrators to track the status of vehicles entering the garage.
Interface: Dashboard section displaying information about newly arrived cars.
Functionality:

- View details of newly arrived cars.
- Assign parking spaces to newly arrived cars.

3.8.3 Managing Leaving Cars:

Description: Admins can view and manage information about cars that are leaving the garage, such as checkout times and payment status. This functionality enables administrators to track the departure of vehicles and ensure smooth operations.
Interface: Dashboard section displaying information about departing cars.
Functionality:

- View details of departing cars.
- Update checkout times.
- Monitor payment status.

3.8.4 Managing the Type of Service the Car Takes

Description: Admins can define and update the types of services available for cars in the garage, such as basic parking, valet service, or car wash. This functionality allows administrators to customize service offerings based on customer needs.
Interface: Service management interface within the admin panel.
Functionality:

- Define new service types.
- Update existing service types.

3.8.5 Managing Garage Settings

Description: Admins can configure various settings related to the garage, such as operating hours, pricing, and notification preferences. This functionality allows administrators to customize the system according to garage requirements.
Interface: Settings configuration interface within the admin panel.
Functionality:

Set operating hours.
Configure pricing for services.
Define notification preferences.

3.8.6 Managing Parking Reservations :

Description: Admins can view and manage parking reservations, including creating, updating, and canceling reservations, as well as viewing reservation history. This functionality enables administrators to monitor and manage parking space allocation efficiently.
Interface: Reservation management interface within the admin panel.
Functionality:

- Create new parking reservations.
- Update existing reservations.
- Cancel reservations as needed.
- View reservation history.

4.Non-Functional Requirements

4.1 Performance:

Description: The Car Garage system should be able to handle multiple user requests simultaneously without significant delays.
Measurement: Response time for user requests under peak load conditions.
Acceptance Criteria: System response time should not exceed 20 seconds for 3 number of simultaneous user requests.
Dependencies: Dependent on system resources and optimization of code.

4.2 Security:

Description: User data, including personal and payment information, should be securely stored and transmitted using encryption protocols. Authentication mechanisms should prevent unauthorized access.
Measurement: Compliance with encryption standards and security protocols.
Acceptance Criteria: All user data is encrypted during transmission and storage. Authentication mechanisms prevent unauthorized access.

Dependencies: Integration with secure encryption libraries and authentication protocols.

4.3 Scalability:

Description: The Car Garage system should be scalable to accommodate a growing number of users and parking spaces.
Measurement: System performance under increasing load.
Acceptance Criteria: The system can handle a 20% increase in users and parking spaces without degradation in performance.
Dependencies: Scalable infrastructure and efficient resource allocation.

4.4 Reliability:

Description: The Car Garage system should be available and reliable, with minimal downtime for maintenance or updates.
Measurement: System uptime and frequency of maintenance.
Acceptance Criteria: System uptime should be 99.9% with scheduled maintenance windows communicated in advance.
Dependencies: Robust infrastructure and proactive maintenance procedures.

4.5 Usability:

Description: The user interface should be intuitive and user-friendly, allowing users to navigate easily and perform tasks without confusion.
Measurement: User feedback and usability testing.
Acceptance Criteria: Users rate the interface as intuitive with a score of X or higher on usability surveys.
Dependencies: User interface design principles and usability testing.

4.6 Accessibility:

Description: The Car Garage system should be accessible to users with disabilities, following accessibility guidelines and standards.
Measurement: Compliance with accessibility standards (e.g., WCAG).
Acceptance Criteria: The system conforms to WCAG 2.0 AA accessibility standards.
Dependencies: Accessibility guidelines and compliance tools.

4.7 Availability :

Description: The Car Garage system should be available 24/7, allowing users to make parking reservations and payments at any time.
Measurement: Downtime during scheduled and unscheduled maintenance.

Acceptance Criteria: The system is available for use 99.9% of the time, with scheduled maintenance windows communicated in advance.
Dependencies: Redundant infrastructure and disaster recovery planning.

4.8 Data Backup and Recovery

Description: Regular backups of the system's data should be performed to prevent data loss in case of system failures or disasters. Data recovery mechanisms should be implemented.
Measurement: Frequency of backups and time to recover data.
Acceptance Criteria: Regular backups are performed daily, and data recovery can be completed within X hours of a system failure.
Dependencies: Backup infrastructure and disaster recovery planning.

5. External Interface Requirements

5.1 User Interface (UI)

Description: The system should feature a user-friendly and intuitive interface optimized for ease of use. It should be accessible across various devices such as desktop computers, laptops, tablets, and smartphones. Users should have the ability to customize settings such as font size, color schemes, and language preferences.
Dependencies: User interface design principles and compatibility testing across different devices and screen sizes.

5.2 Hardware Interface:

Description: The system should be compatible with a range of hardware devices commonly used in garage operations, such as scanners, printers, and ticket dispensers. Hardware requirements should be clearly documented and communicated to users to ensure seamless integration.
Dependencies: Compatibility testing with different hardware devices and documentation of hardware specifications.

5.3 Software Interface:

Description: The system should be designed to work with various software platforms commonly used in garage management, including operating systems and web browsers. Software requirements should be clearly documented and communicated to users to ensure compatibility and smooth operation.
Dependencies: Compatibility testing with different software platforms and documentation of software requirements.

5.4 Communication Interface:

Description: The system should be capable of communicating with other relevant systems and devices, such as payment gateways and customer notification services. It should support various communication channels such as email, SMS, and push notifications for interactions with customers and staff.

Dependencies: Integration with communication APIs and documentation of communication protocols and channels.

These external interface requirements are crucial to ensuring that the garage management system is accessible, compatible with a range of hardware and software platforms, and capable of seamless communication with other systems and devices.

6. Appendices

6.1 Glossary

- **API (Application Programming Interface):** a way for two or more computer programs or components to communicate with each other. It is a type of software interface, offering a service to other pieces of software.
- **User Interface (UI):** The graphical layout of an application through which users interact with it.
- **Hardware Interface:** The connection point between a software system and hardware devices, specifying how they interact.
- **Software Interface**: The point of interaction between software components or between software and hardware.