

Software Requirements Specification (SRS)

Document Title: Garage Connection Application - Software Requirements Specification (SRS)

Project Name: Garage Connection Application

Version: 1.0

Author(s): Osama Akbari 60307447 and Saeed Abdullah Dar 60307149

Reviewer(s): Technical Review Committee

Approver(s): Project Manager, Product Owner

Creation Date: 2025-05-27

Last Update: 2025-06-01

Status: Final

Document Reference: GCA-SRS-001

Scope: This SRS covers the functional and non-functional requirements for the Garage Connection Application that connects vehicle owners with local garage services.

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) documents the functional and non-functional requirements for the Garage Connection Application - a comprehensive digital platform that connects vehicle owners with local garage services. The intended audience includes:

- System Analysts and Developers responsible for designing and implementing the system
- QA/Test Engineers for validating the system against these requirements
- Stakeholders/Project Managers overseeing development and ensuring alignment with business objectives
- UX/UI Designers creating user interfaces and experiences
- Database administrators and system architects

1.2 Document Conventions

- **Shall** indicates a requirement that is mandatory
- **Should** indicates a requirement that is recommended (but may have valid exceptions)
- **May or Optional** indicates a feature that is not mandatory
- **Admin** refers to system administrators with full access privileges
- **GarageOwner** refers to garage business owners managing their services
- **VehicleOwner** refers to end-users seeking automotive services
- **System** refers to the Garage Connection Application platform

1.3 Intended Audience and Reading Suggestions

- **Developers:** Pay close attention to Section 3 (System Features), Section 4 (External Interface Requirements), and Section 5 (Non-Functional Requirements)
- **Testers:** Use the requirements and use cases in Section 3 to develop test plans and acceptance criteria
- **Project Managers:** Review the entire document for scope, assumptions, and dependencies
- **UI/UX Designers:** Focus on Section 4 (External Interface Requirements) and usability aspects in Section 5

1.4 Scope

The scope of this SRS covers the complete Garage Connection Application system, specifically:

- User management for three user types: Admin, GarageOwner, and VehicleOwner
- Garage management and service catalog maintenance
- Appointment booking and management system
- Review and rating system
- Geolocation-based garage search functionality
- Promotional campaign management
- User authentication and account management

The system excludes payment processing, which will be handled by third-party services, and direct vehicle diagnostic capabilities.

1.5 References

- ISO/IEC/IEEE 29148:2018 – International standard for software and systems requirements
- GDPR (General Data Protection Regulation) for data privacy compliance
- OAuth 2.0 specification for secure authentication
- RESTful API design principles
- Google Maps API documentation for geolocation services

2. Overall Description

2.1 Product Perspective

The Garage Connection Application is a standalone web and mobile application that serves as an intermediary platform connecting vehicle owners with garage services. The system:

- Operates as a comprehensive service marketplace for automotive repair and maintenance
- Integrates with external geolocation services for location-based searches
- Interfaces with third-party authentication providers for secure user management
- Connects to mapping services for navigation and route optimization

- Supports multi-tenant architecture serving different user types with role-based access

2.2 Product Functions

- **User Management:** Registration, authentication, and profile management for all user types
- **Garage Management:** Complete garage profile creation, service catalog management, and operational information maintenance
- **Service Discovery:** Advanced search and filtering capabilities for garages and services
- **Appointment System:** Real-time booking, scheduling, and appointment management
- **Review System:** Interactive feedback mechanism with ratings and responses
- **Promotional Tools:** Campaign creation and targeted promotion management
- **Geolocation Services:** Location-based search and navigation assistance
- **Administrative Tools:** System-wide management and oversight capabilities

2.3 User Classes and Characteristics

1. **Vehicle Owners (Primary Users):**
 - End consumers seeking automotive services
 - Minimal technical knowledge required
 - Expect intuitive, mobile-friendly interfaces
 - Require reliable service information and easy booking processes
2. **Garage Owners (Business Users):**
 - Small to medium business owners managing automotive services
 - Need efficient tools for service management and customer interaction
 - Require business intelligence and appointment management capabilities
 - Expect professional-grade administrative interfaces
3. **System Administrators (Technical Users):**
 - Technical personnel managing the overall system
 - Require comprehensive system oversight and management tools
 - Need access to all system functions and user management capabilities
 - Expect advanced reporting and analytics features

2.4 Operating Environment

- **Client-side:** Web browsers (Chrome, Firefox, Safari, Edge) and mobile applications (iOS, Android)
- **Server-side:** Cloud-based infrastructure with scalable architecture
- **Database:** Relational database management system with data replication
- **External Services:** Integration with geolocation APIs, mapping services, and authentication providers
- **Network:** Secure HTTPS connections with encrypted data transmission

2.5 Design and Implementation Constraints

- **Security Requirements:** Data encryption, secure authentication, and privacy protection compliance
- **Performance Requirements:** Response times under 3 seconds for standard operations
- **Scalability Requirements:** Support for concurrent users and growing data volumes

- **Compatibility Requirements:** Cross-platform compatibility for web and mobile devices
- **Regulatory Compliance:** GDPR compliance for data protection and local business regulations

2.6 User Documentation

- Interactive tutorials and onboarding guides for new users
- Comprehensive help documentation and FAQ sections
- Video tutorials for complex operations
- In-app contextual help and tooltips
- Customer support contact information and channels

2.7 Assumptions and Dependencies

- Reliable internet connectivity for all system operations
 - External geolocation services availability and accuracy
 - Third-party authentication services operational status
 - Mobile device GPS capabilities for location-based features
 - User willingness to share location data for optimal service delivery
-

3. System Features (Functional Requirements)

3.1 Use Case: Manage Garage (Admin)

3.1.1 Description

Allows system administrators to add, edit, and manage garage registrations within the platform.

3.1.2 Trigger

Admin user selects "Manage Garages" from the administration dashboard.

3.1.3 Preconditions

- User has valid admin credentials and is authenticated
- Admin dashboard is accessible
- Database connectivity is established

3.1.4 Main Flow

1. Admin opens the app
2. System displays login page
3. Admin logs in as Admin
4. System validates credentials
5. System displays Admin dashboard

6. Admin clicks "Manage Garages"
7. System shows list of registered garages
8. Admin clicks "Add Garage"
9. System displays "Add Garage" form
10. Admin enters garage name
11. Admin enters garage address
12. Admin enters contact info
13. Admin enters operating hours
14. Admin submits the form
15. System validates input fields
16. System saves new garage to database
17. System shows success message "Garage added successfully"
18. System refreshes and displays updated garage list

3.1.5 Alternative Flows

- **Invalid Input (AF1):**
 - Steps 1–9 as in Main Flow
 - Admin leaves garage name missing
 - System validates input fields
 - System shows error message (e.g., "Address is required")
 - Admin enters missing or corrected info
 - Admin submits the form
 - System continues with steps 15–18 in Main Flow
- **Admin Enters Invalid Credentials (AF2):**
 - Steps 1–3 as in Main Flow
 - Admin enters incorrect username or password
 - System displays "Invalid credentials, please try again"
 - Admin enters correct credentials
 - System continues with steps 4–18 in Main Flow
- **Admin Leaves a Required Field Blank (AF3):**
 - Steps 1–13 as in Main Flow
 - Admin leaves a required field blank
 - Admin submits the form
 - System displays "This field is required"
 - Admin fills in the missing field
 - System continues with steps 14–18 in Main Flow
- **Admin Enters Invalid Data (Address, Contact Info, or Operating Hours) (AF4):**
 - Steps 1–13 as in Main Flow
 - Admin enters invalid data in a field (e.g., address, contact info, operating hours)
 - Admin submits the form
 - System displays "Invalid [field] format" (e.g., "Invalid address format")
 - Admin corrects the data
 - System continues with steps 14–18 in Main Flow
- **Admin Enters Duplicate Garage Name or Address (AF5):**
 - Steps 1–14 as in Main Flow
 - System checks for duplicate garage (same name or address)
 - System displays "A garage with this name/address already exists"

- Admin edits garage details
 - System continues with steps 14–18 in Main Flow
- **Admin Cancels Before Submitting the Form (AF6):**
 - Steps 1–13 as in Main Flow
 - Admin closes the app or navigates away before submitting
 - Garage creation process aborted; user returns to garage list on next open
- **Session Times Out Before Submitting (AF7):**
 - Steps 1–13 as in Main Flow
 - Admin is inactive for 2 hours
 - System displays "Session timed out, please log in again"
 - Admin logs in again
 - System continues with steps 1–18 in Main Flow
- **Admin Tries to Edit or Delete a Non-Existent Garage (AF8):**
 - Steps 1–7 as in Main Flow
 - Admin selects a garage to edit or delete that no longer exists
 - System displays "Garage not found"
 - Admin returns to garage list
 - System shows updated list

3.1.6 Postconditions

- New garage is successfully added to the system database
- Updated garage list is displayed to admin
- Garage is available for vehicle owner searches

3.1.7 Special Requirements

- Input validation for all required fields
- Duplicate garage name detection within same geographical area
- Audit logging of all garage management activities

3.2 Use Case: Manage Services (Admin)

3.2.1 Description

Enables administrators to create and maintain the master list of automotive services available in the system.

3.2.2 Trigger

Admin user selects "Manage Services" from the administration dashboard.

3.2.3 Preconditions

- Admin authentication completed
- Admin dashboard accessible
- Service management privileges verified

3.2.4 Main Flow

1. Admin opens the app
2. System displays login page
3. Admin enters username
4. Admin enters password
5. Admin submits credentials
6. System validates credentials
7. System displays Admin dashboard
8. Admin clicks "Manage Services"
9. System shows list of services
10. Admin clicks "Add Service"
11. System displays "Add Service" form
12. Admin enters service name
13. Admin enters service description
14. Admin enters default price range
15. Admin submits the form
16. System validates input fields
17. System saves new service to database
18. System shows success message "Service added successfully"
19. System refreshes and displays updated services list

3.2.5 Alternative Flows

- **Admin Enters Wrong Username (AF1):**
 - Admin opens the app
 - System displays login page
 - Admin enters wrong username
 - Admin enters password
 - Admin submits credentials
 - System validates credentials (username not found)
 - System displays "Invalid username, please try again"
 - Admin enters correct username and password
 - System continues with steps 6–19 in Main Flow
- **Admin Enters Wrong Password (AF2):**
 - Admin opens the app
 - System displays login page
 - Admin enters username
 - Admin enters wrong password
 - Admin submits credentials
 - System validates credentials (password incorrect)
 - System displays "Invalid password, please try again"
 - Admin enters correct password
 - System continues with steps 6–19 in Main Flow
- **Admin Leaves a Required Field Blank (AF3):**
 - Steps 1–14 as in Main Flow
 - Admin leaves a required field blank
 - Admin submits the form

- System displays "This field is required"
 - Admin fills in the missing field
 - System continues with steps 15–19 in Main Flow
- **Admin Enters Invalid Data (e.g., Price Range Format) (AF4):**
 - Steps 1–14 as in Main Flow
 - Admin enters invalid data (e.g., price range format)
 - Admin submits the form
 - System displays "Invalid input format"
 - Admin corrects the data
 - System continues with steps 15–19 in Main Flow
- **Service Name Already Exists (AF5):**
 - Steps 1–15 as in Main Flow
 - System checks for duplicate service name
 - System displays "Service already exists"
 - Admin edits service name
 - System continues with steps 16–19 in Main Flow
- **Admin Cancels Before Submitting the Form (AF6):**
 - Steps 1–14 as in Main Flow
 - Admin closes the form or navigates away before submitting
 - Service creation process aborted; returns to services list
- **Session Times Out Before Submitting (AF7):**
 - Steps 1–14 as in Main Flow
 - Admin is inactive for 2 hours
 - System displays "Session timed out, please log in again"
 - Admin logs in again
 - System continues with steps 1–19 in Main Flow
- **System/Database Error on Submission (AF8):**
 - Steps 1–15 as in Main Flow
 - Admin submits the form
 - System/database error occurs
 - System displays "An error occurred while saving. Please try again later."
 - Admin retries submission or contacts support
 - If resolved, system continues with steps 16–19 in Main Flow

3.2.6 Postconditions

- New service is added to master service catalog
- Service is available for garage owners to add to their offerings
- Updated service list is displayed

3.2.7 Special Requirements

- Service categorization and tagging capabilities
- Price range validation and formatting
- Service description character limits and formatting

3.3 Use Case: Create Promotion (Admin/Garage Owner)

3.3.1 Description

Allows administrators to create system-wide promotional campaigns and garage owners to create garage-specific promotions.

3.3.2 Trigger

Admin selects "Create Promotion" from the administrative dashboard, OR Garage owner clicks "Create Promotion" from garage owner dashboard.

3.3.3 Preconditions

- User authentication verified (Admin or Garage owner)
- For Garage owners: Garage profile is verified and active
- Promotional management access granted
- Active garages and services available in system
- Promotions module is available

3.3.4 Main Flow

1. System displays user dashboard (Admin or Garage owner)
2. User clicks "Create Promotion"
3. System displays promotion creation form
4. User enters promotion details (title, description, discount %, start date, end date, validity dates)
5. For Admin: User selects applicable garages and services
6. User submits the form
7. System validates input fields and date ranges
8. System creates promotion in database
9. System shows success message "Promotion created successfully"
10. System refreshes and displays updated promotions list
11. Promotion becomes visible to vehicle owners

3.3.5 Alternative Flows

- **Invalid Date Range (AF1):**
 - System detects invalid or past dates, or end date earlier than start date
 - System displays error message "Invalid date range"
 - User corrects dates and resubmits
- **Missing Required Fields (AF2):**
 - User submits the form with empty mandatory fields
 - System highlights missing fields and displays an error message
- **Duplicate Promotion (AF3):**
 - System detects a similar promotion already active
 - System asks whether to overwrite or cancel
- **Backend/System Error (AF4):**
 - System fails to save data due to an internal issue
 - System shows "Unable to create promotion. Please try again later."

3.3.6 Postconditions

- Promotion is active and visible to applicable users
- Targeted notifications sent to relevant vehicle owners
- Promotion tracking initiated for analytics

3.3.7 Special Requirements

- Date validation ensuring future start dates
- Promotion duration limits and scheduling conflicts
- Automated notification system integration

3.4 Use Case: Edit Services (Garage Owner)

3.4.1 Description

Enables garage owners to manage their service offerings, pricing, and availability.

3.4.2 Trigger

Garage owner selects "Edit Services" from their dashboard.

3.4.3 Preconditions

- Garage owner authentication completed
- Garage profile exists and is verified
- Garage has existing services listed
- Master service catalog available
- Service management module is working

3.4.4 Main Flow

1. System displays garage owner dashboard
2. Garage owner clicks "Edit Services"
3. System shows list of services currently offered
4. Garage owner selects a service to edit OR clicks "Add Service"
5. For new service: System displays service selection form with master service list
6. For editing: System shows service details
7. Garage owner selects service from master list (if adding) OR updates service name, price, description, or estimated duration
8. Garage owner enters/updates specific price and estimated duration
9. Garage owner clicks "Save"
10. System validates input fields
11. System saves service to garage profile/updates the record
12. System shows success message and confirmation message
13. System refreshes and displays updated services list

3.4.5 Alternative Flows

- **Invalid Price Format/Invalid Input (AF1):**
 - System detects invalid price format, illegal characters, or negative price
 - System displays error message "Invalid price format" or validation error
 - Garage owner corrects price and resubmits
- **No Changes Made (AF2):**
 - Garage owner tries to save without modifying anything
 - System displays "No changes to save"
- **Service Not Found (AF3):**
 - Service selected is deleted or doesn't exist anymore
 - System shows "Service not found" and returns to service list
- **Update Failure (AF4):**
 - Database fails to update due to server issue
 - System shows "Update failed. Try again later."

3.4.6 Postconditions

- Service is added to or updated in garage's offerings
- Service is visible to vehicle owners in searches
- Pricing and duration information updated

3.4.7 Special Requirements

- Price validation and formatting standards
- Service duration estimation guidelines
- Capacity management for service scheduling

3.5 Use Case: Manage Appointments (Garage Owner)

3.5.1 Description

Allows garage owners to view, update, and manage customer appointments.

3.5.2 Trigger

Garage owner selects "Manage Appointments" from their dashboard.

3.5.3 Preconditions

- Garage owner authentication verified/logged in
- Garage profile active
- Appointment system operational

3.5.4 Main Flow

1. Garage owner opens the app

2. System displays login page
3. Garage owner enters username
4. Garage owner enters password
5. Garage owner submits credentials
6. System validates credentials
7. System displays GarageOwner dashboard
8. Garage owner clicks "Manage Appointments"
9. System retrieves and displays list of upcoming and past appointments
10. Garage owner selects an appointment to view, edit, or cancel
11. System displays appointment details with options to edit or cancel
12. Garage owner makes changes (e.g., reschedules, updates details, or cancels)
13. Garage owner submits changes
14. System validates changes (e.g., date/time availability, required fields)
15. System saves changes to database
16. System shows success message (e.g., "Appointment updated successfully" or "Appointment cancelled")
17. System displays updated appointments list

3.5.5 Alternative Flows

- **User Enters Invalid Username (AF1):**
 - Steps 1–3 as in Main Flow
 - Garage owner enters invalid username
 - System displays "Invalid username, please try again"
 - Garage owner retries or exits
- **User Enters Wrong Password (AF2):**
 - Steps 1–4 as in Main Flow
 - Garage owner enters wrong password
 - System displays "Invalid password, please try again"
 - Garage owner retries or exits
- **No Appointments Available (Present or Past) (AF3):**
 - Steps 1–8 as in Main Flow
 - System retrieves appointments returns empty
 - System displays "No appointments found"
 - Garage owner returns to dashboard or menu
- **User Selects Invalid or Unavailable Appointment (AF4):**
 - Steps 1–10 as in Main Flow
 - Garage owner selects an appointment that has been deleted or is no longer available
 - System displays "Selected appointment not found"
 - Garage owner returns to appointments list
- **User Leaves Required Fields Blank When Editing Appointment (AF5):**
 - Steps 1–12 as in Main Flow
 - Garage owner leaves required fields blank
 - Garage owner submits changes
 - System displays "All fields are required"
 - Garage owner completes all fields
 - System continues with steps 13–17 in Main Flow
- **User Enters Invalid Data (e.g., Invalid Date/Time) (AF6):**

- Steps 1–12 as in Main Flow
 - Garage owner enters invalid data (e.g., date/time in the past, invalid format)
 - Garage owner submits changes
 - System displays "Invalid input" or "Date/time not available"
 - Garage owner corrects the data
 - System continues with steps 13–17 in Main Flow
- **User Cancels Editing or Managing Appointment Before Submitting (AF7):**
 - Steps 1–12 as in Main Flow
 - Garage owner clicks "Cancel" or navigates away before submitting changes
 - Changes are discarded; system returns to appointments list
- **Session Times Out Before Submitting Changes (AF8):**
 - Steps 1–12 as in Main Flow
 - Garage owner is inactive for a long time
 - System displays "Session timed out, please log in again"
 - Garage owner logs in again
 - System continues with steps 1–17 in Main Flow
- **System/Database Error on Submission (AF9):**
 - Steps 1–13 as in Main Flow
 - Garage owner submits changes
 - System/database error occurs
 - System displays "An error occurred while saving. Please try again later."
 - Garage owner retries submission
 - If resolved, system continues with steps 14–17 in Main Flow

3.5.6 Postconditions

- Appointment status updated in system
- Vehicle owner notified of any changes
- Appointment history maintained for records

3.5.7 Special Requirements

- Real-time appointment status updates
- Automated customer notification system
- Appointment conflict detection and resolution

3.6 Use Case: Answer Review (Garage Owner)

3.6.1 Description

Enables garage owners to respond to customer reviews and feedback.

3.6.2 Trigger

Garage owner selects "View Reviews" from their dashboard.

3.6.3 Preconditions

- Garage owner authentication completed
- Customer reviews exist for the garage
- Review response system active

3.6.4 Main Flow

1. Garage owner opens the app
2. System displays login page
3. Garage owner enters username
4. Garage owner enters password
5. Garage owner submits credentials
6. System validates credentials
7. System displays GarageOwner dashboard
8. Garage owner clicks "View Reviews"
9. System displays list of recent reviews for their garage
10. Garage owner selects a review to answer
11. System displays review details and "Answer Review" form
12. Garage owner writes a response
13. Garage owner submits the response
14. System validates input
15. System saves response to database
16. System shows success message "Response submitted successfully"
17. System refreshes and displays updated review with response

3.6.5 Alternative Flows

- **Garage Owner Enters Wrong Username (AF1):**
 - Garage owner opens the app
 - System displays login page
 - Garage owner enters wrong username
 - Garage owner enters password
 - Garage owner submits credentials
 - System validates credentials (username not found)
 - System displays "Invalid username, please try again"
 - Garage owner enters correct username and password
 - System continues with steps 6–17 in Main Flow
- **Garage Owner Enters Wrong Password (AF2):**
 - Garage owner opens the app
 - System displays login page
 - Garage owner enters username
 - Garage owner enters wrong password
 - Garage owner submits credentials
 - System validates credentials (password incorrect)
 - System displays "Invalid password, please try again"
 - Garage owner enters correct password
 - System continues with steps 6–17 in Main Flow
- **Garage Owner Leaves Response Field Blank (AF3):**
 - Steps 1–12 as in Main Flow

- Garage owner submits the response without entering text
 - System displays "Response field cannot be empty"
 - Garage owner writes a response
 - System continues with steps 13–17 in Main Flow
- **Garage Owner Cancels Before Submitting Response (AF4):**
 - Steps 1–12 as in Main Flow
 - Garage owner closes the form or navigates away before submitting
 - Answer process aborted; user returns to reviews list
- **No Reviews Available to Respond To (AF5):**
 - Garage owner opens the app
 - System displays login page
 - Garage owner enters username
 - Garage owner enters password
 - Garage owner submits credentials
 - System validates credentials
 - System displays GarageOwner dashboard
 - Garage owner clicks "View Reviews"
 - System checks for available reviews
 - System displays message "No reviews available to respond to at this time"
 - Garage owner returns to dashboard
- **Session Times Out Before Submitting Response (AF6):**
 - Steps 1–12 as in Main Flow
 - Garage owner is inactive for a long time
 - System displays "Session timed out, please log in again"
 - Garage owner logs in again
 - System continues with steps 1–17 in Main Flow
- **System/Database Error on Submission (AF7):**
 - Steps 1–13 as in Main Flow
 - Garage owner submits the response
 - System/database error occurs
 - System displays "An error occurred while saving. Please try again later."
 - Garage owner retries submission or contacts support
 - If resolved, system continues with steps 14–17 in Main Flow

3.6.6 Postconditions

- Response is published with the review
- Customer receives notification of response
- Review interaction history updated

3.6.7 Special Requirements

- Response character limits and content guidelines
- Automated notification to review author
- Review moderation and reporting capabilities

3.7 Use Case: View History (Vehicle Owner)

3.7.1 Description

Allows vehicle owners to view their appointment and service history.

3.7.2 Trigger

Vehicle owner selects "View History" from their dashboard.

3.7.3 Preconditions

- Vehicle owner authentication verified
- User account contains historical data
- History system accessible

3.7.4 Main Flow

1. Vehicle owner opens the app
2. System displays login page
3. Vehicle owner enters username
4. Vehicle owner enters password
5. Vehicle owner submits credentials
6. System validates credentials
7. System displays VehicleOwner dashboard
8. Vehicle owner clicks "View History"
9. System retrieves user's service and appointment history
10. System displays history list (services, appointments, reviews)
11. Vehicle owner scrolls and selects a history item for details
12. System displays detailed information about selected history item

3.7.5 Alternative Flows

- **User Enters Wrong Username (AF1):**
 - Steps 1–3 as in Main Flow
 - Vehicle owner enters wrong username
 - Vehicle owner enters password
 - Vehicle owner submits credentials
 - System displays "Invalid username, please try again"
 - Vehicle owner enters correct username and password
 - System continues with steps 6–12 in Main Flow
- **User Enters Wrong Password (AF2):**
 - Steps 1–4 as in Main Flow
 - Vehicle owner enters wrong password
 - Vehicle owner submits credentials
 - System displays "Invalid password, please try again"
 - Vehicle owner enters correct password
 - System continues with steps 6–12 in Main Flow
- **User Has No History Available (AF3):**

- Steps 1–8 as in Main Flow
 - System retrieves history returns empty
 - System displays message "No history available"
 - Vehicle owner returns to dashboard or other menu
- **Session Times Out Before or During Viewing History (AF4):**
 - Steps 1–9 as in Main Flow
 - Vehicle owner inactive for a long time
 - System displays "Session timed out, please log in again"
 - Vehicle owner logs in again
 - System continues with steps 6–12 in Main Flow
- **System/Database Error Retrieving History (AF5):**
 - Steps 1–8 as in Main Flow
 - Error occurs retrieving history data
 - System displays "Unable to retrieve history at this time. Please try again later."
 - Vehicle owner retries or returns to dashboard
- **User Selects Invalid or Unavailable History Item (AF6):**
 - Steps 1–11 as in Main Flow
 - Vehicle owner selects a history item that no longer exists or is invalid
 - System displays "Selected history item not found"
 - Vehicle owner returns to history list

3.7.6 Postconditions

- Historical information displayed to user
- User can access detailed service records
- Data available for future reference

3.7.7 Special Requirements

- Chronological sorting of historical data
- Detailed service and payment information
- Export capabilities for personal records

3.8 Use Case: Create Review (Vehicle Owner)

3.8.1 Description

Enables vehicle owners to create reviews and ratings for completed services.

3.8.2 Trigger

System prompts user to create review after appointment completion, or user manually selects review option from history.

3.8.3 Preconditions

- Vehicle owner authentication completed

- Completed appointment exists
- Review system operational

3.8.4 Main Flow

1. Steps 1–11 from View History (Vehicle owner selects a completed service/appointment)
2. Vehicle owner clicks "Create Review"
3. System displays review form
4. Vehicle owner enters review text and rating
5. Vehicle owner submits review
6. System validates input
7. System saves review to database
8. System shows success message "Review submitted successfully"
9. System displays updated history with new review

3.8.5 Alternative Flows

- **Review Already Exists for Selected Service/Appointment (AF1):**
 - Steps 1–2 as in Main Flow
 - System checks for existing review
 - System displays "You have already reviewed this appointment/service"
 - Vehicle owner returns to history list
- **User Leaves Review Text or Rating Blank (AF2):**
 - Steps 1–4 as in Main Flow
 - Vehicle owner leaves review text or rating blank
 - Vehicle owner submits review
 - System displays "All fields are required"
 - Vehicle owner completes all fields
 - System continues with steps 5–9 in Main Flow
- **User Types Inappropriate Review Text (AF3):**
 - Steps 1–4 as in Main Flow
 - Vehicle owner enters invalid data in review
 - Vehicle owner submits review
 - System displays "Inappropriate language detected"
 - Vehicle owner corrects the data
 - System continues with steps 5–9 in Main Flow
- **User Cancels Before Submitting Review (AF4):**
 - Steps 1–4 as in Main Flow
 - Vehicle owner clicks "Cancel" or navigates away before submitting
 - Review creation process aborted; system returns to history list
- **Session Times Out Before Submitting Review (AF5):**
 - Steps 1–4 as in Main Flow
 - Vehicle owner is inactive for 2 hours
 - System displays "Session timed out, please log in again"
- **System/Database Error on Submission (AF6):**
 - Steps 1–5 as in Main Flow
 - Vehicle owner submits review
 - System/database error occurs

- System displays "An error occurred while saving. Please try again later."
 - Vehicle owner retries submission
 - If resolved, system continues with steps 6–9 in Main Flow
- **Selected Service/Appointment is Deleted or Unavailable Before Review is Submitted (AF7):**
 - Steps 1–2 as in Main Flow
 - System cannot find the selected service/appointment (deleted/unavailable)
 - System displays "Selected item not found"
 - Vehicle owner returns to history list

3.8.6 Postconditions

- Review is published and visible to other users
- Garage's overall rating updated
- Review added to user's history

3.8.7 Special Requirements

- Review authenticity verification
- Rating aggregation and display algorithms
- Review moderation and content filtering

3.9 Use Case: Create Account (Vehicle Owner)

3.9.1 Description

Allows new vehicle owners to register and create accounts in the system.

3.9.2 Trigger

New user selects "Sign Up" from the welcome screen.

3.9.3 Preconditions

- Application is accessible
- Registration system operational
- Email/SMS verification services available

3.9.4 Main Flow

1. Vehicle owner opens the app
2. System displays "Welcome to Garage Go." "Sign in or Sign Up?"
3. Vehicle owner selects Sign Up
4. System displays registration form "username, password, DOB, email, phone number and optionally; vehicles they own"
5. Vehicle owner provides valid details
6. Vehicle owner submits the form
7. System validates fields

8. System creates user
9. System displays "Welcome to Garage Go, user"
10. System displays "Verify details by phone / email"
11. System sends an automated OTP to the user's email or phone number to verify
12. Vehicle owner verifies by entering the OTP
13. System validates OTP
14. System unlocks features for the user
15. System displays home page and asks for optional user preferences (vehicle brand, which area they're close to, or vehicle production year)

3.9.5 Alternative Flows

- **User Tries to Sign In Before Registration (AF1):**
 - User selects "Sign In" but account doesn't exist
 - System displays "User doesn't exist, would you like to try again? You can try to sign up"
 - User proceeds with registration process (steps 3–15 in Main Flow)
- **Username Already Exists (AF2):**
 - Steps 1–6 as in Main Flow
 - System detects duplicate username
 - System displays "User already exists, try using a different username"
 - User enters alternative username
 - System continues with steps 6–15 in Main Flow
- **Invalid Email Format (AF3):**
 - Steps 1–6 as in Main Flow
 - System detects invalid email format
 - System displays "Email doesn't exist" with format example
 - User corrects email format
 - System continues with steps 6–15 in Main Flow

3.9.6 Postconditions

- New user account created and verified
- User preferences optionally configured
- User can access all vehicle owner features

3.9.7 Special Requirements

- Strong password requirements and validation
- Email and phone number format validation
- Privacy policy acceptance and GDPR compliance
- Optional vehicle information for personalized services

3.10 Use Case: Manage Appointments (Vehicle Owner)

3.10.1 Description

Allows vehicle owners to view, edit, or cancel their appointments.

3.10.2 Trigger

Vehicle owner selects "Manage Appointments" from their dashboard.

3.10.3 Preconditions

- Vehicle owner authentication completed
- User has existing appointments
- Appointment management system accessible

3.10.4 Main Flow

1. Vehicle owner opens the app
2. System displays login page
3. Vehicle owner enters username
4. Vehicle owner enters password
5. Vehicle owner submits credentials
6. System validates credentials
7. System displays VehicleOwner dashboard
8. Vehicle owner clicks "Manage Appointments"
9. System retrieves and displays list of upcoming and past appointments
10. Vehicle owner selects an appointment to view, edit, or cancel
11. System displays appointment details with options to edit or cancel
12. Vehicle owner makes changes (e.g., reschedules, updates details, or cancels)
13. Vehicle owner submits changes
14. System validates changes (e.g., date/time availability, required fields)
15. System saves changes to database
16. System shows success message (e.g., "Appointment updated successfully" or "Appointment cancelled")
17. System displays updated appointments list

3.10.5 Alternative Flows

- **User Enters Invalid Username (AF1):**
 - Steps 1–3 as in Main Flow
 - Vehicle owner enters invalid username
 - System displays "Invalid username, please try again"
 - Vehicle owner retries or exits
- **User Enters Wrong Password (AF2):**
 - Steps 1–4 as in Main Flow
 - Vehicle owner enters wrong password
 - System displays "Invalid password, please try again"
 - Vehicle owner retries or exits
- **No Appointments Available (Present or Past) (AF3):**
 - Steps 1–8 as in Main Flow
 - System retrieves appointments returns empty
 - System displays "No appointments found"

- Vehicle owner returns to dashboard or menu
- **User Selects Invalid or Unavailable Appointment (AF4):**
 - Steps 1–10 as in Main Flow
 - Vehicle owner selects an appointment that has been deleted or is no longer available
 - System displays "Selected appointment not found"
 - Vehicle owner returns to appointments list
- **User Leaves Required Fields Blank When Editing Appointment (AF5):**
 - Steps 1–12 as in Main Flow
 - Vehicle owner leaves required fields blank
 - Vehicle owner submits changes
 - System displays "All fields are required"
 - Vehicle owner completes all fields
 - System continues with steps 13–17 in Main Flow
- **User Enters Invalid Data (e.g., Invalid Date/Time) (AF6):**
 - Steps 1–12 as in Main Flow
 - Vehicle owner enters invalid data (e.g., date/time in the past, invalid format)
 - Vehicle owner submits changes
 - System displays "Invalid input" or "Date/time not available"
 - Vehicle owner corrects the data
 - System continues with steps 13–17 in Main Flow
- **User Cancels Editing or Managing Appointment Before Submitting (AF7):**
 - Steps 1–12 as in Main Flow
 - Vehicle owner clicks "Cancel" or navigates away before submitting changes
 - Changes are discarded; system returns to appointments list
- **Session Times Out Before Submitting Changes (AF8):**
 - Steps 1–12 as in Main Flow
 - Vehicle owner is inactive for a long time
 - System displays "Session timed out, please log in again"
 - Vehicle owner logs in again
 - System continues with steps 1–17 in Main Flow
- **System/Database Error on Submission (AF9):**
 - Steps 1–13 as in Main Flow
 - Vehicle owner submits changes
 - System/database error occurs
 - System displays "An error occurred while saving. Please try again later."
 - Vehicle owner retries submission
 - If resolved, system continues with steps 14–17 in Main Flow

3.10.6 Postconditions

- Appointment successfully modified or cancelled
- Garage owner notified of changes
- Updated appointment list displayed

3.10.7 Special Requirements

- Cancellation policy enforcement
- Notification system for changes

- Appointment modification time limits

3.11 Use Case: Search Garage (Vehicle Owner)

3.11.1 Description

Enables vehicle owners to search for garages based on location and other criteria.

3.11.2 Trigger

Vehicle owner selects "Search Garage" from their dashboard.

3.11.3 Preconditions

- Vehicle owner authentication completed
- Geolocation services available
- Garage database populated
- Geolocation system is operational

3.11.4 Main Flow

1. System displays vehicle owner dashboard
2. Vehicle owner clicks "Search Garage"
3. System prompts for location or offers geolocation
4. Vehicle owner chooses current location or enters manual address
5. System connects to geolocation services
6. System fetches nearby garages
7. System displays list of garages with details, ratings, and filters (rating, services, distance)
8. Vehicle owner selects a garage
9. System shows garage profile and services

3.11.5 Alternative Flows

- **Geolocation Not Available (AF1):**
 - Location services disabled or inaccessible
 - System prompts user to enable location or enter manually
- **No Garages Found (AF2):**
 - System finds no garages in specified area
 - System displays "No garages found in this area" or "No garages available near you"
 - System prompts for new location search
- **Invalid Address (AF3):**
 - User enters non-existent or malformed address
 - System displays "Invalid location input"
- **External API Error (AF4):**
 - Geolocation system fails or times out
 - System shows fallback message or cached garages

3.11.6 Postconditions

- Relevant garages displayed to user
- User can view detailed garage information
- User can proceed to book services

3.11.7 Special Requirements

- GPS accuracy and location permission handling
- Distance calculation and sorting algorithms
- Advanced filtering options (ratings, services, price range)

3.12 Use Case: Search Services (Vehicle Owner)

3.12.1 Description

Allows vehicle owners to search for specific automotive services across available garages.

3.12.2 Trigger

Vehicle owner selects "Search Services" from their dashboard.

3.12.3 Preconditions

- Vehicle owner authentication completed/logged in
- Service catalog populated/available
- Garage service offerings updated

3.12.4 Main Flow

1. System displays vehicle owner dashboard
2. Vehicle owner clicks "Search Services"
3. System prompts for service type
4. Vehicle owner enters service name or selects from categories
5. System searches garages offering the service/filters and returns matching services and garages
6. System displays garages and service details/list of results
7. Vehicle owner selects a garage
8. System shows garage profile and booking option

3.12.5 Alternative Flows

- **Service Not Available/No Results (AF1):**
 - System finds no garages offering requested service
 - System displays "No garages offer this service" or "No garages found offering this service"
 - System prompts for new service search
- **Typo or Vague Search (AF2):**

- Keyword too vague or mistyped
- System suggests corrected or more precise terms
- **System Timeout (AF3):**
 - Backend service list cannot be fetched
 - Error message shown to user

3.12.6 Postconditions

- Relevant service providers displayed
- User can compare service offerings
- User can proceed to book appointment

3.12.7 Special Requirements

- Service matching algorithms
- Price comparison capabilities
- Service availability and scheduling integration

3.13 Use Case: Book Appointment (Vehicle Owner)

3.13.1 Description

Enables vehicle owners to book appointments for automotive services.

3.13.2 Trigger

Vehicle owner selects booking option after finding desired garage and service.

3.13.3 Preconditions

- Vehicle owner authentication completed/logged in
- Garage and service selected
- Appointment scheduling system operational
- Appointment booking system is online

3.13.4 Main Flow

1. Vehicle owner searches and selects garage/service OR clicks "Book Appointment"
2. System displays matching garages/services
3. Vehicle owner selects garage and service
4. System shows available appointment slots
5. Vehicle owner chooses date, time, and desired service
6. Vehicle owner enters required details and submits
7. System validates availability and checks garage availability
8. Vehicle owner confirms booking
9. System creates appointment and sends confirmation/saves to database
10. System shows booking details and adds to appointments list

11. Garage owner is notified

3.13.5 Alternative Flows

- **Time Slot Unavailable/Slot Unavailable (AF1):**
 - System detects selected slot is no longer available or double booking
 - System displays "Slot unavailable, please choose another" or prompts user to select a different time
 - Vehicle owner selects alternative slot
- **Incomplete Details (AF2):**
 - Required fields (e.g., date/service) are empty
 - System highlights missing fields
- **Garage Not Accepting Bookings (AF3):**
 - Selected garage has disabled appointments
 - System shows message: "Garage currently not accepting bookings"
- **Booking for Past Date (AF4):**
 - System prevents choosing past date/time
 - Displays validation error
- **User Not Logged In (AF5):**
 - Unauthenticated user attempts booking
 - Redirected to login page
- **System Error (AF6):**
 - Backend issue prevents saving
 - Displays: "Could not book appointment. Try again later."

3.13.6 Postconditions

- Appointment successfully created and confirmed
- Both user and garage owner receive confirmations
- Appointment appears in both parties' management systems

3.13.7 Special Requirements

- Real-time availability checking
- Automated confirmation and reminder system
- Calendar integration capabilities
- Booking conflict prevention

4. External Interface Requirements

4.1 User Interface

4.1.1 General UI Requirements

- Responsive design supporting desktop, tablet, and mobile devices
- Intuitive navigation with consistent layout patterns
- Accessibility compliance (WCAG 2.1 AA standards)
- Multi-language support capability
- Dark mode and light mode options

4.1.2 Vehicle Owner Interface

- Clean, user-friendly design with minimal learning curve
- Location-based garage discovery with map integration
- Service filtering and comparison tools
- Appointment booking calendar interface
- Review and rating submission forms
- Personal dashboard with appointment history

4.1.3 Garage Owner Interface

- Professional business management dashboard
- Service catalog management interface
- Appointment scheduling and management tools
- Customer review response system
- Business analytics and reporting interface
- Profile and operational information management

4.1.4 Administrator Interface

- Comprehensive system administration dashboard
- User and garage management interfaces
- System-wide promotional campaign tools
- Advanced reporting and analytics
- System configuration and maintenance tools

4.2 Hardware Interfaces

- **Mobile Devices:** GPS capability for location services
- **Camera Integration:** For profile photos and service documentation
- **Push Notification Support:** For appointment reminders and updates
- **Biometric Authentication:** Fingerprint and face recognition support where available

4.3 Software Interfaces

4.3.1 Geolocation Services

- **Google Maps API:** For location search and mapping functionality
- **GPS Integration:** Real-time location tracking and navigation
- **Address Validation:** Postal address verification and standardization

4.3.2 Authentication Services

- **OAuth 2.0:** Third-party authentication integration
- **Multi-Factor Authentication:** SMS and email verification
- **Social Media Login:** Google, Facebook authentication options

4.3.3 Notification Services

- **Email Service:** Automated email notifications and confirmations
- **SMS Gateway:** Text message notifications and verification
- **Push Notification Service:** Real-time mobile app notifications

4.3.4 Payment Integration (Future Enhancement)

- **Payment Gateway APIs:** Secure transaction processing
- **Digital Wallet Integration:** Support for mobile payment systems

4.4 Communication Interfaces

- **HTTPS Protocol:** Secure data transmission for all communications
 - **RESTful APIs:** Standard API architecture for system integration
 - **WebSocket Connections:** Real-time updates and notifications
 - **JSON Data Format:** Standardized data exchange format
-

5. Non-Functional Requirements

5.1 Performance

- **Response Time:** Standard operations complete within 3 seconds
- **Search Performance:** Location-based searches return results within 5 seconds
- **Concurrent Users:** System supports minimum 1000 concurrent users
- **Database Performance:** Query response time under 2 seconds for standard operations
- **Mobile Performance:** Optimized for mobile devices with efficient resource usage

5.2 Security

- **Data Encryption:** All sensitive data encrypted in transit and at rest using AES-256
- **Authentication:** Strong password requirements with multi-factor authentication
- **Authorization:** Role-based access control with principle of least privilege
- **Data Privacy:** GDPR compliance with user consent management
- **Audit Logging:** Comprehensive logging of all system activities and user actions
- **Vulnerability Management:** Regular security assessments and updates

5.3 Reliability and Availability

- **System Uptime:** 99.5% availability target with planned maintenance windows
- **Data Backup:** Automated daily backups with point-in-time recovery
- **Disaster Recovery:** Recovery time objective (RTO) of 4 hours
- **Error Handling:** Graceful error handling with user-friendly error messages
- **Failover Capability:** Automatic failover for critical system components

5.4 Usability

- **User Experience:** Intuitive interface requiring minimal training
- **Accessibility:** Full compliance with WCAG 2.1 AA accessibility standards
- **Mobile Optimization:** Touch-friendly interface optimized for mobile devices
- **Help System:** Comprehensive in-app help and documentation
- **User Onboarding:** Guided tutorials for new user registration and first use

5.5 Scalability

- **Horizontal Scaling:** Architecture supports adding additional servers
- **Database Scaling:** Database design supports sharding and replication
- **Load Balancing:** Traffic distribution across multiple server instances
- **Caching Strategy:** Implemented caching for frequently accessed data
- **API Rate Limiting:** Prevents abuse and ensures fair resource usage

5.6 Maintainability

- **Modular Architecture:** Clean separation of concerns following SOLID principles
- **Code Quality:** Comprehensive unit testing with minimum 80% code coverage
- **Documentation:** Detailed technical documentation and API specifications
- **Version Control:** Git-based version control with branching strategy
- **Continuous Integration:** Automated testing and deployment pipelines

5.7 Portability

- **Cross-Platform:** Web application compatible with major browsers
- **Mobile Apps:** Native iOS and Android applications
- **Database Independence:** Database abstraction layer for flexibility
- **Cloud Deployment:** Containerized deployment supporting multiple cloud providers

5.8 Regulatory Compliance

- **Data Protection:** GDPR compliance for European users
 - **Accessibility Standards:** Section 508 and ADA compliance
 - **Industry Standards:** Following automotive industry best practices
 - **Local Regulations:** Compliance with local business and consumer protection laws
-

6. Other Requirements

6.1 Analytics and Reporting

- **User Analytics:** Comprehensive tracking of user behavior and system usage
- **Business Intelligence:** Dashboard reporting for garage owners and administrators
- **Performance Metrics:** System performance monitoring and reporting
- **Custom Reports:** Configurable reporting system for various stakeholder needs

6.2 Integration Requirements

- **Third-Party APIs:** Extensible architecture for future service integrations
- **Data Import/Export:** Standard formats for data migration and backup
- **Webhook Support:** Real-time notifications to external systems
- **API Documentation:** Comprehensive API documentation for developers

6.3 Internationalization

- **Multi-Language Support:** Framework for supporting multiple languages
- **Localization:** Region-specific formatting for dates, currencies, and addresses
- **Time Zone Support:** Proper handling of different time zones
- **Cultural Adaptation:** UI/UX considerations for different cultural contexts

6.4 Future Enhancements (Out of Scope for Current Version)

- **Payment Processing:** Integrated payment system for service fees
 - **Advanced Analytics:** Machine learning for service recommendations
 - **IoT Integration:** Connected vehicle diagnostic integration
 - **Loyalty Programs:** Customer reward and loyalty management system
 - **Fleet Management:** Enterprise features for vehicle fleet management
-

7. Appendices

7.1 Glossary

- **Admin:** System administrator with full system access and management privileges
- **API:** Application Programming Interface for system integration
- **Garage Owner:** Business user managing garage services and appointments
- **Geolocation:** Technology for determining geographical location
- **GPS:** Global Positioning System for location services
- **OAuth:** Open standard for access delegation and authentication
- **OTP:** One-Time Password for secure verification
- **REST:** Representational State Transfer architectural style

- **SRS:** Software Requirements Specification document
- **Vehicle Owner:** End user seeking automotive services
- **Webhook:** HTTP callbacks for real-time notifications

7.2 Assumptions

- Users have access to reliable internet connectivity
- Mobile devices have GPS capabilities enabled
- Third-party services (maps, authentication) maintain acceptable uptime
- Users are willing to share location data for service optimization
- Garage owners maintain accurate service and availability information

7.3 Dependencies

- External geolocation and mapping services availability
 - Third-party authentication service reliability
 - SMS and email service provider functionality
 - Cloud infrastructure and hosting service stability
 - Mobile app store approval and distribution processes
-

End of SRS Document

This Software Requirements Specification provides a comprehensive foundation for the development of the Garage Connection Application, ensuring all stakeholder needs are addressed while maintaining professional standards for system development and implementation.