Software Requirements Specification (SRS) for AutoClean BD

1. Introduction

1.1 Purpose
This document outlines the requirements for *AutoClean BD*, a web-based platform for on-demand car and bike wash services in Bashundhara R/A, Dhaka. It defines the system's scope, features, interfaces, and constraints to guide development.

1.2 Document Conventions
Standards: IEEE SRS format.
- Abbreviations:
  - R/A: Residential Area
  - bKash/Nagad: Mobile payment platforms
  - GUI: Graphical User Interface
- Formatting:
  - Requirements in bullet points

1.3 Intended Audience
- Developers: Implement features .
- Testers: Verify requirements .
- Admin/Managers: Review scope .

1.4 Project Scope
AutoClean BD enables:
- Customers can book car/bike washes online.
- Admins to manage bookings, workers, and payments.
- Workers to receive assignments (via admin).
Exclusions: Real-time worker GPS tracking, automated payment reconciliation.

1.5 References
- IEEE 830-1998 SRS Guidelines.

2. Overall Description

2.1 Product Perspective
Standalone web platform integrating:
- Customer portal (frontend).
- Admin dashboard (backend).
- Third-party payment gateways (bKash/Nagad).

2.2 Product Features
1. Customer booking with vehicle/package selection.

2. Admin dashboard for booking/worker management.
3. Payment processing (cash, bKash, Nagad).
4. Booking status tracking and notifications.

2.3 User Classes and Characteristics

User	**Characteristics**
Customer	Tech-savvy residents; requires intuitive UI for booking/payment.
Admin	Manages operations; needs data visualization (bookings, reports).
Service Worker	Receives assignments via phone/app; no system login required.

2.4 Operating Environment
- Frontend: Responsive web (Chrome, Firefox, Safari).
- Backend: Linux/Windows server, PHP/Python, MySQL/PostgreSQL.
- Hardware: Standard web servers; mobile/desktop client access.

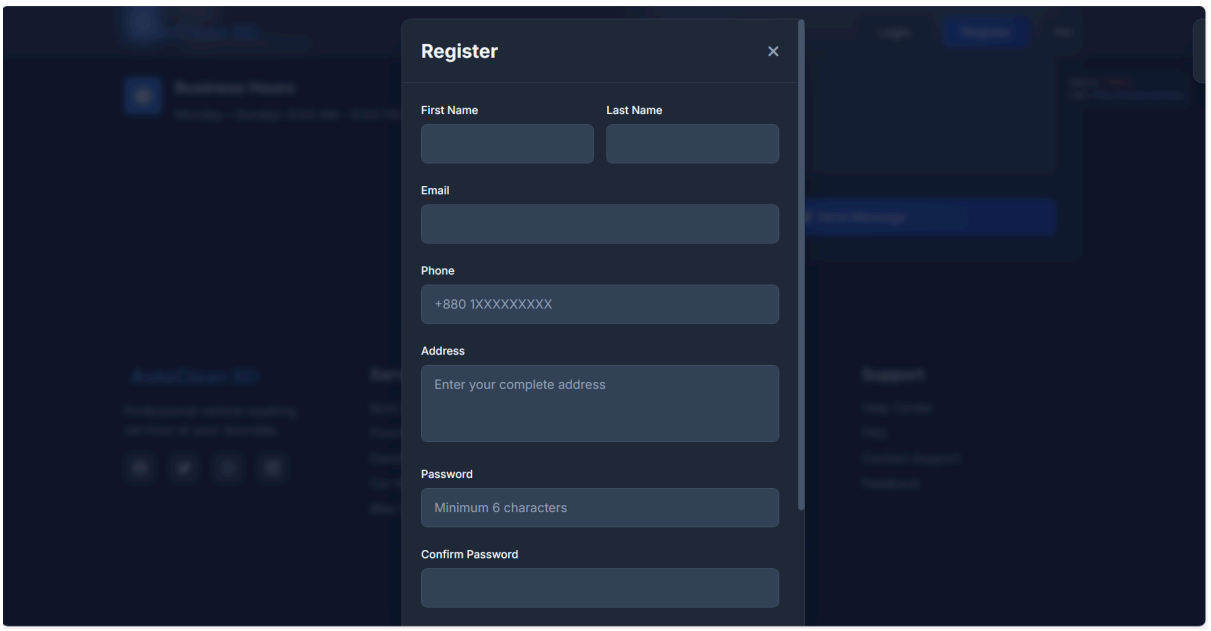
2.5 Design Constraints
- Must support cashless payments (bKash/Nagad APIs).
- Responsive design for mobile/desktop.
- Compliance with Bangladesh data privacy laws.

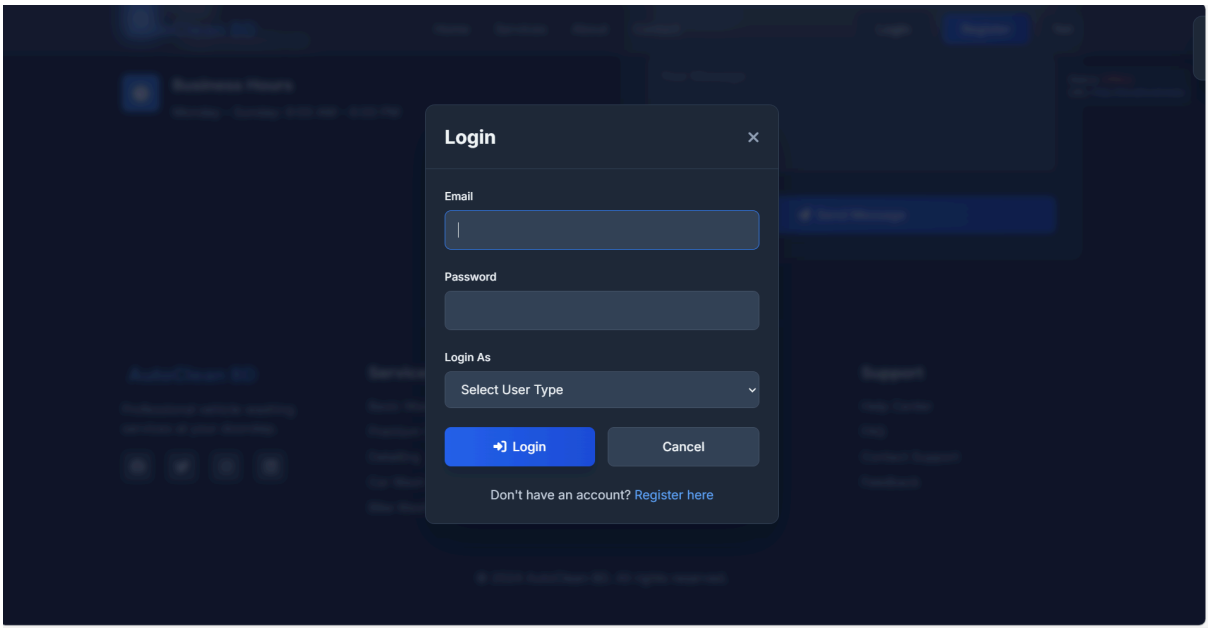
2.6 User Documentation
- Customer: Booking/payment guides (web).
- Admin: Dashboard manual (PDF).

2.7 Assumptions & Dependencies
- Assumptions: Customers have internet access; workers own phones.
- Dependencies: Payment gateway APIs (bKash/Nagad), SMS/email services.

3. System Features

3.1 Feature: Customer Booking
- Description/Priority: Core feature (High).
- Stimulus/Response:
  - Customer selects: Vehicle type → Package → Time → Payment → Confirmation.
  - *System responds: Real-time slot validation → Booking ID → Email/SMS confirmation.





- Functional Requirements:
  - `FR-01`: User registration/login (email/phone).
  - `FR-02`: Display service packages (Basic, Premium, Detailing) with pricing.
  - `FR-03`: Calendar/time slot selection based on worker availability.
  - `FR-04`: Payment method choice (Cash, bKash, Nagad).

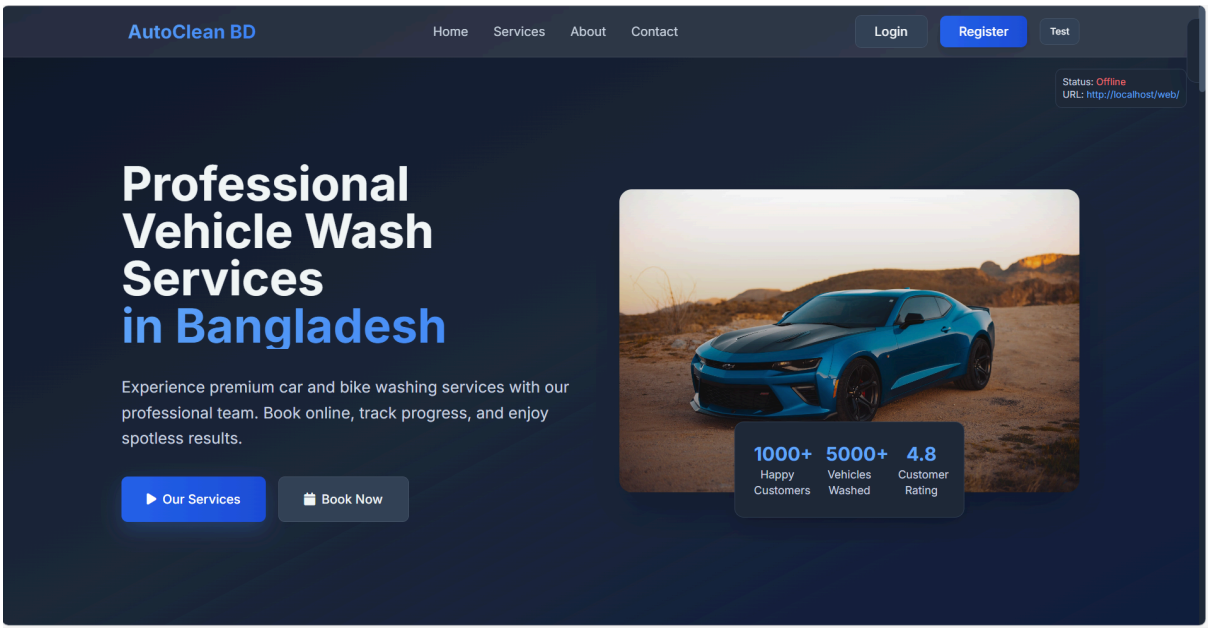
3.2 Feature: Admin Dashboard
- Description/Priority: Core feature (High).
- Stimulus/Response:
  - Admin assigns worker → Worker notified via SMS.
  - Admin updates status → Customer notified.
- Functional Requirements:
  - `FR-05`: Secure admin login.
  - `FR-06`: View/manage bookings (filter by date/status).
  - `FR-07`: Assign workers to bookings manually.

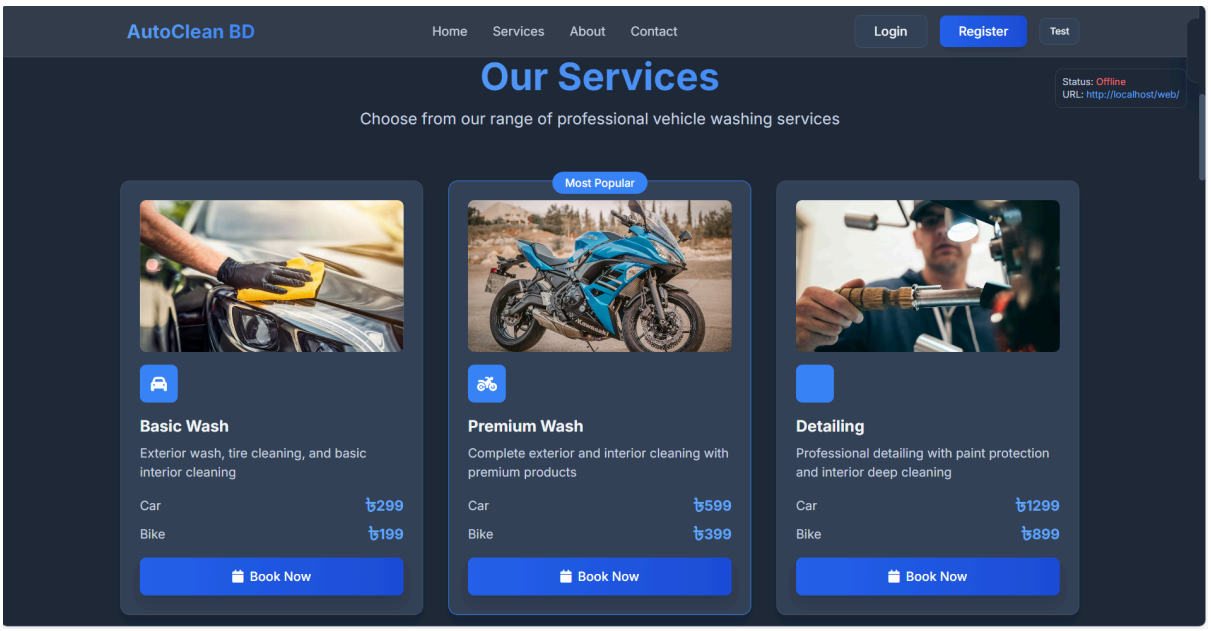
- `FR-08`: Generate reports (earnings, worker productivity).

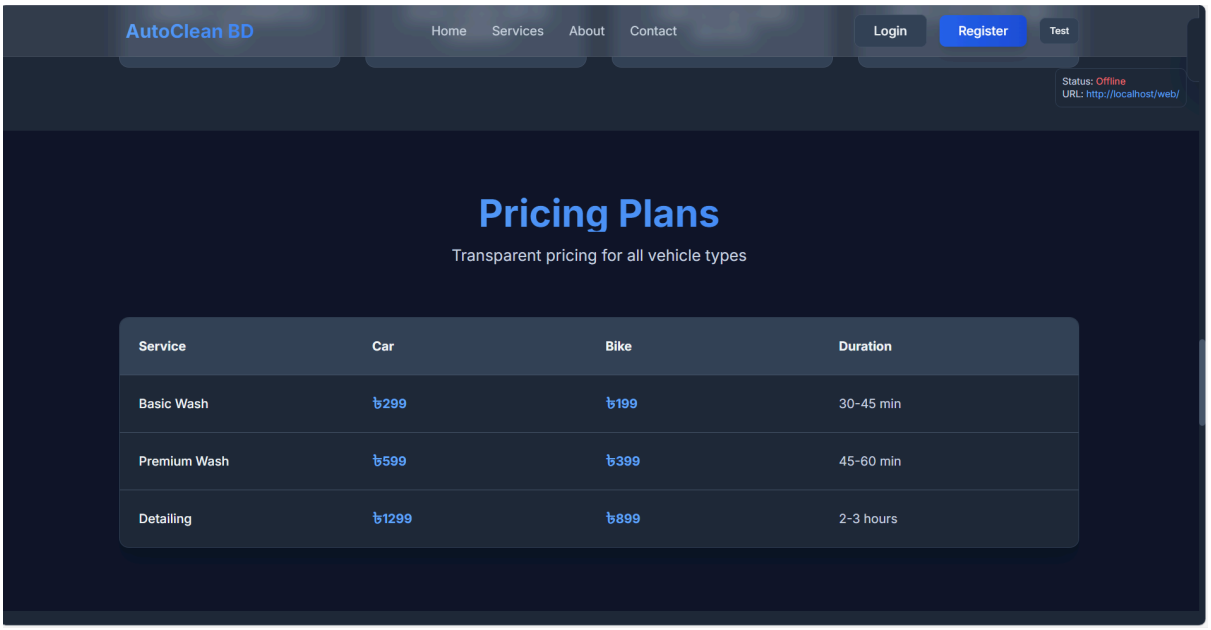
3.3 Feature: Feedback & Reviews
- Functional Requirements:
  - `FR-09`: Customers submit ratings/reviews post-service.
  - `FR-10`: Admin views feedback in dashboard.

4. External Interface Requirements

4.1 User Interfaces
- Customer Portal:
  - Homepage: Service packages, booking form.
  - Dashboard: Booking history, status tracking.
- Admin Dashboard:
  - Tables: Bookings, workers, feedback.
  - Charts: Earnings, completion rates.







4.2 Hardware Interfaces
- None beyond standard servers/devices.

4.3 Software Interfaces
- Payment Gateways: bKash/Nagad APIs for transactions.
- SMS/Email: Twilio/SendGrid for notifications.

4.4 Communications Interfaces
- HTTPS for data security.
- RESTful APIs for payment processing.

5. Nonfunctional Requirements

5.1 Performance
- `NF-01`: Booking completion in < 5 seconds.
- `NF-02`: Support 500 concurrent users.

5.2 Safety
- `NF-03`: Backup database daily.

5.3 Security
- `NF-04`: Encrypt passwords .
- `NF-05`: PCI-DSS compliance for payments.

5.4 Quality Attributes
- Usability: Mobile-friendly UI (Google Lighthouse score > 90).
- Maintainability: Modular code (documented APIs).

6. Other Requirements

- Localization: Bengali/English language support.

Appendices

A. Glossary
- COD: Cash on Delivery.
- Detailing: Premium cleaning package.

B. Analysis Models
(To be added during design phase)
- Use case diagram: Customer booking flow.

- ER Diagram: User-booking-payment entities.

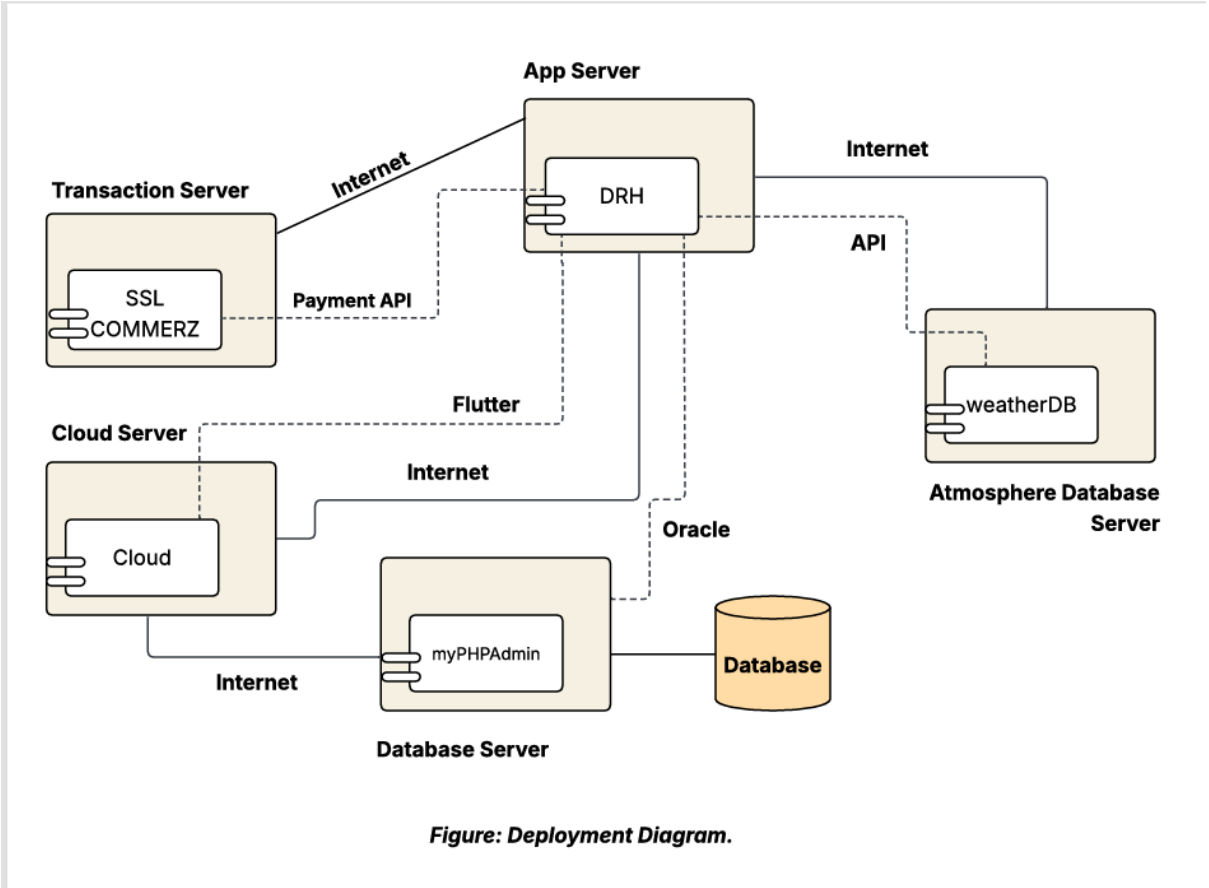
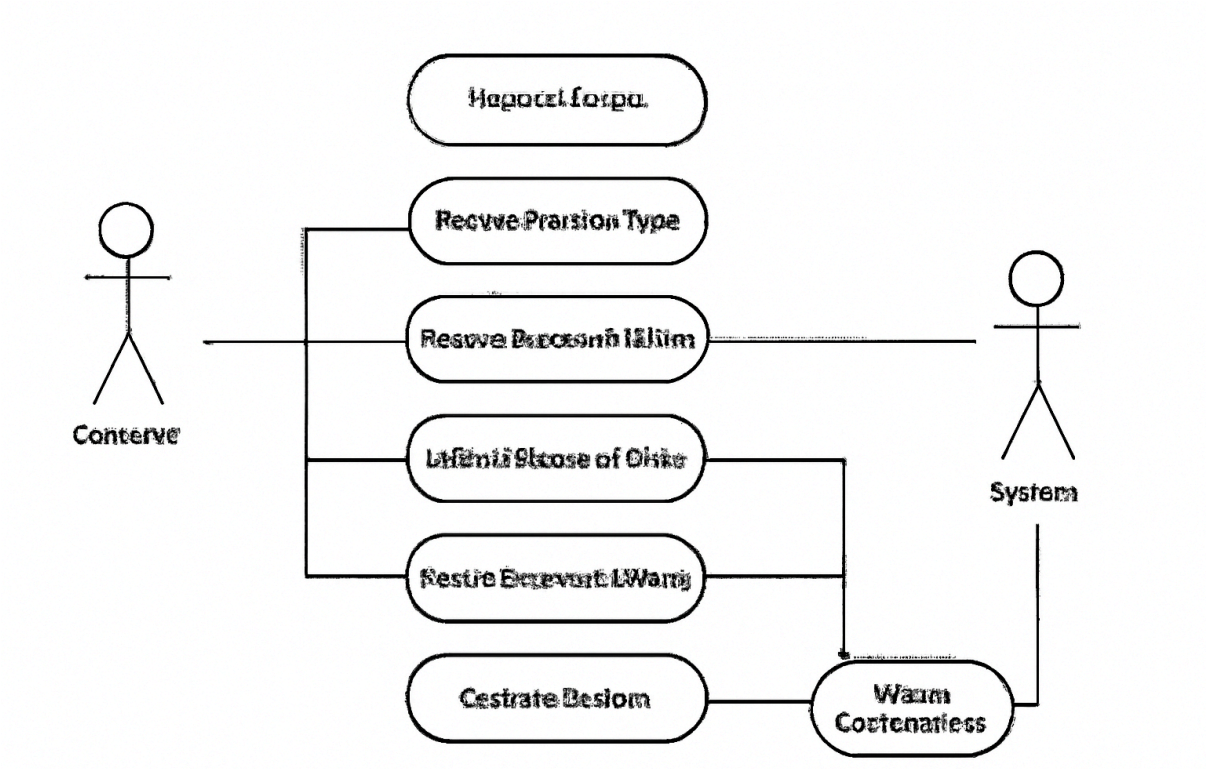


Figure: Deployment Diagram.



C. Issues List

Integration with bKash/Nagad APIs pending vendor approval.