**Chapter 4**

**Methodology, Results, and Discussions**

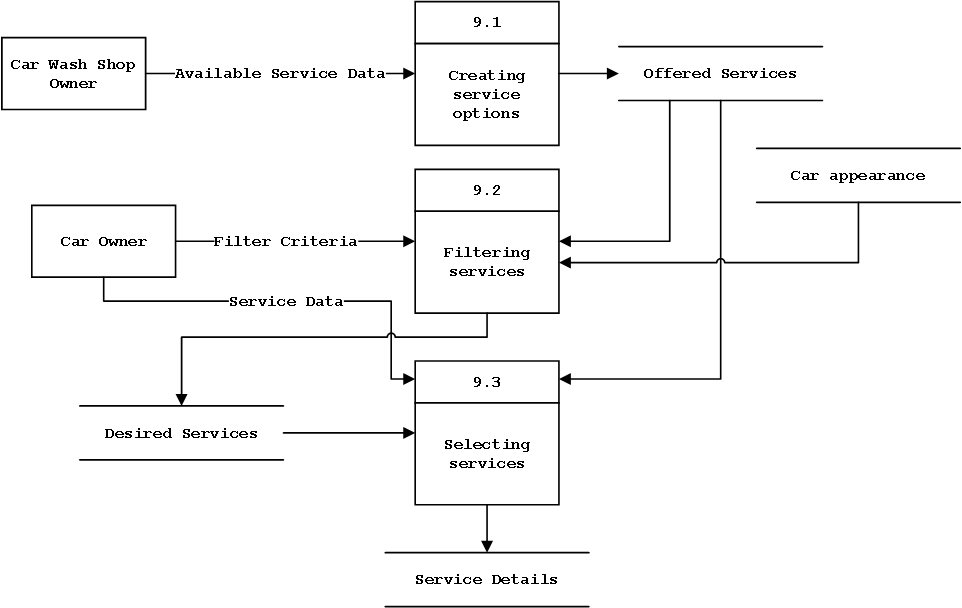
This chapter presents the testing and implementation results of Spark Mobile, an innovative mobile application designed to provide convenient car wash services for busy car owners in the comfort of their homes. The primary goal is to outline the step-by-step execution of the capstone project, addressing identified gaps and validating the system's functionality.

The chapter details key development processes, including requirements analysis, which utilizes data flow diagrams to illustrate information flow within the system. It also covers essential aspects such as functionality, data management, user interface design, and technical requirements to ensure optimal performance and usability.

Security and legal compliance are also examined to safeguard user data and adhere to relevant regulations. Additionally, the user requirements section identifies the specific needs of both car owners and service providers, ensuring Spark Mobile delivers an efficient and user-friendly experience.

The development process follows a structured methodology within a defined environment, enabling a systematic approach to building and refining the application. Rigorous testing procedures verify system reliability, while project scheduling, resource allocation, and risk management strategies help keep development on track and address potential challenges effectively.

**Requirements Analysis**

**

*Figure 4.1 Level 1 Data Flow Diagram (Selecting Services)*

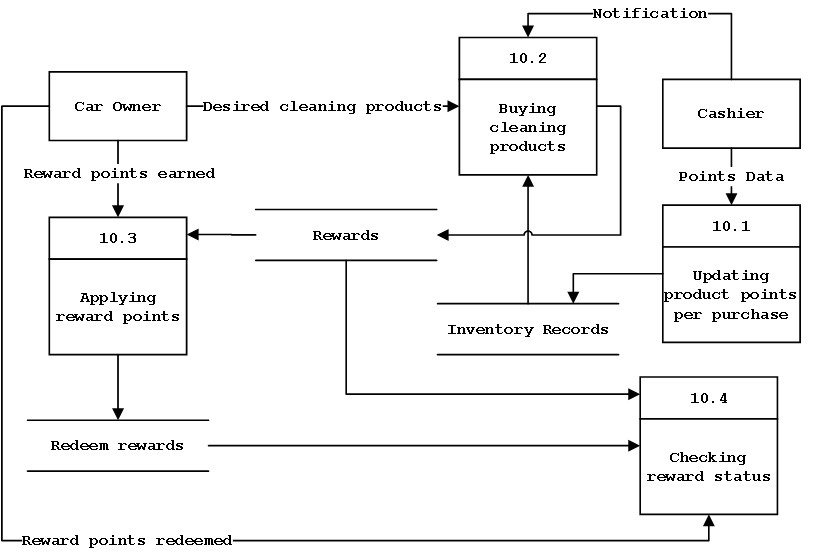
The **Level 1 Data Flow Diagram (DFD) for Selecting Services** illustrates the process of how a car owner selects a car wash service through the application. It details the interaction between the **Car Wash Shop Owner,** the **Car Owner,** and the system's internal processes that handle service selection.

The process begins with the **Car Wash Shop Owner,** who provides the system with **Available Service Data**. This data includes the different car wash services offered by the shop, such as basic washes, interior detailing, waxing, and other related services. The system processes this information through the **Creating Service Options (Process 9.1),** where the available services are structured into a list of **Offered Services.**

Once the service options are created, the **Car Owner** interacts with the system by entering **Filter Criteria**—these could include preferences based on price, service type, or duration. The system processes this input through the **Filtering Services (Process 9.2),** refining the offered services according to the given criteria. Additionally, **Car Appearance** may play a role in filtering, considering factors like vehicle size, level of dirt, or specific cleaning requirements.

After filtering, the system presents the **Service Data** to the **Selecting Services (Process 9.3)** stage, where the **Car Owner** makes a final decision on which service(s) they want to avail. The selected services are then recorded as **Desired Services**, which ultimately generate the final **Service Details**. These details may include specifics such as service name, price, estimated duration, and any special requests.

This structured approach ensures that the car owner can efficiently browse, filter, and select the most suitable car wash service while allowing the car wash shop to manage and offer their services effectively.



*Figure 4.2 Level 1 Data Flow Diagram (Purchasing Cleaning Products)*

The **Level 1 Data Flow Diagram (DFD) for Purchasing Cleaning Products** illustrates how a car owner buys cleaning product through the system and how reward points are earned and applied during the process. The diagram outlines the interactions between the **Car Owner,** the **Cashier,** and various system processes that handle product purchases and reward points.

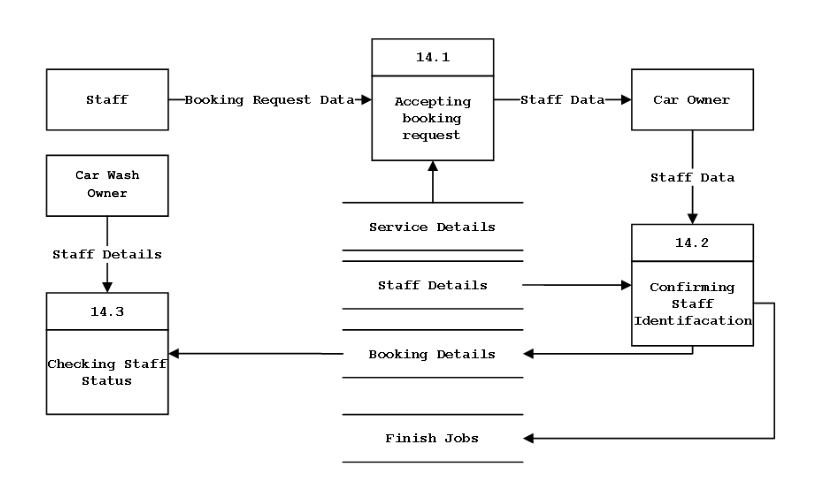
The process begins when the **Car Owner** selects the **Desired Cleaning Products** they wish to purchase. This request is processed by the **Buying Cleaning Products (Process 10.2)** module. Once the purchase is initiated, the system sends a **Notification** to the **Cashier,** ensuring the transaction is recorded. Simultaneously, **Inventory Records** are updated to reflect the change in stock.

To manage the reward points, the system includes the **Updating Product Points per Purchase (Process 10.1)** module. This process logs the purchase transaction and updates the **Points Data**, which tracks the reward points earned based on the products bought. The accumulated points are then associated with the **Car Owner’s** account.

The **Car Owner** can later redeem their reward points through the **Applying Reward Points (Process 10.3)** module. This process allows the car owner to use their earned points as **Rewards**, which can be applied to future purchases or special offers. The system also supports checking reward eligibility through the **Checking Reward Status (Process 10.4)** module, ensuring that the car owner knows how many points they have and whether they can be redeemed.

Once the car owner decides to use their points, the system processes the **Redeem Rewards** request, deducting the appropriate points from their balance and confirming that the **Reward Points Redeemed** have been successfully applied.

This structured approach ensures a seamless purchase experience, allowing car owners to buy cleaning products while benefiting from a reward system that encourages customer loyalty and continued engagement with the platform.



*Figure 4.3 Level 1 Data Flow Diagram (Cleaning the vehicle)*

The Level 1 Data Flow Diagram (DFD) for Booking Staff Services illustrates how a Car Owner requests a service, how Staff members are assigned, and how their identification and job status are confirmed within the system. The diagram highlights the interactions between the Staff, Car Owners, and various system processes that facilitate the booking and completion of services.

The process begins when either the Staff or the Car Owner submits a **Booking Request**. This request is processed by the **Accepting Booking Request (Process 14.1)** module, which records the request and retrieves **Staff Data**. The system then generates **Service Details, Staff Details, and Booking Details** to manage the request efficiently.

Once the booking is recorded, the **Confirming Staff Identification (Process 14.2)** module ensures that the assigned Staff member’s information is verified before proceeding. This verification step guarantees that the correct Staff is dispatched to perform the service. Upon confirmation, **Booking Details** are finalized, and Staff are assigned specific jobs. As services are completed, the system updates the **Finish Jobs** data to reflect task completion.

The **Checking Staff Status (Process 14.3)** module allows the system to track and verify **Staff Details**, ensuring they are available and qualified for assignments. This process ensures that only eligible Staff are assigned to bookings, improving efficiency and service reliability.

This structured approach ensures a smooth and transparent booking process for Car Owners while enabling Staff to efficiently manage service requests.

**Functional Requirements**

**Functionality**

The system shall provide essential features to facilitate convenient and efficient car wash services for users. It shall ensure seamless booking, payment processing, and service tracking.

F1: The system shall allow users to create an account and log in using their email and password.

F2: The system shall allow users to browse available car wash services, including exterior wash, interior detailing, and full detailing.

F3: The system shall allow users to schedule car wash appointments by selecting a preferred date, time, and location.

F4: The system shall provide users with real-time service availability and allow them to reschedule bookings if needed.

F5: The system shall enable users to view pricing details for each service before confirming a booking.

**F6:** The system shall allow users to track the status of their car wash service in real-time.

**F7:** The system shall allow the users to browse car wash shop available near at their location.

**F8:** The system shall generate digital receipts and booking confirmations for completed transactions.

**F9:** The system shall allow users to rate and review car wash providers based on their service experience.

**F10:** The system shall send notifications and reminders about upcoming or completed appointments via email or in-app alerts.

***Data***

To ensure a seamless and efficient user experience, Spark Mobile must handle various types of data, including user profiles, service details, bookings, and payments. The system will securely store and process this data to maintain service accuracy, protect user privacy, and facilitate smooth transactions. Below are the key data elements that Spark Mobile will manage:

**User Data**

User data, including full name, email, mobile number, address, profile picture, and password, shall be securely stored in a relational database.

**Format:** Text fields (VARCHAR) for names and contact details, DATETIME for timestamps, and file paths for storing profile pictures.

**Storage:** Encrypted within the users table, with passwords hashed using bcrypt.

**Validation:** User inputs will be checked for completeness, proper formatting (e.g., valid email and phone number), and uniqueness (e.g., email must be unique).

**Privacy:** Sensitive information like passwords will be encrypted, and access will be restricted based on user roles.

**Service Data**

Service details, including service name, description, price, and estimated duration, shall be stored in a structured database.

**Format:** Text fields (VARCHAR) for service names and descriptions, DECIMAL for pricing, and TIME for estimated duration.

**Storage:** Stored in the services table, with each service linked to providers and availability schedules.

**Integrity:** Service pricing and details will be regularly updated to maintain accuracy.

**Booking Data**

Booking details, including booking ID, user ID, selected service, date, time, and location, shall be recorded in the system.

**Format:** VARCHAR for booking IDs, DATETIME for scheduling, and text fields for addresses.

**Storage:** Stored in the bookings table, linked to users and service providers through foreign keys.

**Validation:** The system will verify service availability before confirming bookings.

**Privacy:** Only users and assigned service providers can access booking details.

**Payment Data**

Transaction details, including transaction ID, user ID, payment method, amount, and status, shall be securely processed and stored.

**Format:** DECIMAL for payment amounts, VARCHAR for transaction IDs and methods, and DATETIME for payment timestamps.

**Storage:** Stored in the payments table, linked to bookings and users via foreign keys.

**Security:** Payments will be processed through an encrypted gateway (e.g., PayPal, Stripe).

**Validation:** Transactions will be verified before confirming payment status.

**Shop Data**

Shop details, including shop name, location, contact details, and available services, will be stored to provide users with provider information.

**Format:** Text fields (VARCHAR) for names, locations, and contact details.

**Storage:** Stored in the shops table, linked to available services and employees.

**Integrity:** Data updates will be managed by authorized personnel to maintain service accuracy.

**Car Data**

Car details, including car ID, user ID, brand, model, year, color, and license plate, will be stored to help users manage their vehicle information.

**Format:** Text fields (VARCHAR) for brand, model, and license plate, INT for year.

**Storage:** Stored in the cars table, linked to user accounts.

**Validation:** Users must input correct vehicle details for accurate service recommendations.

**Privacy:** Only users can access and update their car details.

**Employee Data**

Employee information, including employee ID, name, contact details, role, assigned shop, and work schedule, will be securely maintained.

**Format:** Text fields (VARCHAR) for names and roles, DATETIME for work schedules.

**Storage:** Stored in the employees table, linked to shops and bookings.

**Validation:** Only authorized admins can add or modify employee records.

**Privacy:** Employee information will only be visible to shop administrators and system managers.

**Review and Rating Data**

User feedback, including ratings and review comments, will be collected to improve service quality.

**Format:** INT for ratings, text fields (VARCHAR) for review comments.

**Storage:** Stored in the reviews table, linked to users and service providers.

**Validation:** Reviews must follow content guidelines to prevent spam or abuse.

**Notification Data**

Notifications, including alerts for bookings, payments, and promotions, will be managed within the system.

**Format:** Text fields (VARCHAR) for message content, DATETIME for timestamps.

**Storage:** Stored in the notifications table, linked to user accounts.

**Privacy:** Only intended recipients can view their notifications.

**Data Processing Methods**

**Validation:** All user inputs will be checked both client-side and server-side to ensure accuracy.

**Integrity Checks:** Database constraints like foreign keys, unique constraints, and transactional updates will be used to maintain consistency.

**Encryption:** Sensitive data, including passwords and payment details, will be encrypted using secure algorithms like AES and bcrypt.

**Data Security and Privacy**

**Access Control:** Role-based permissions will be implemented to restrict access to sensitive information.

**Encryption:** Secure communication (e.g., HTTPS) will be enforced, and sensitive fields will be encrypted at rest.

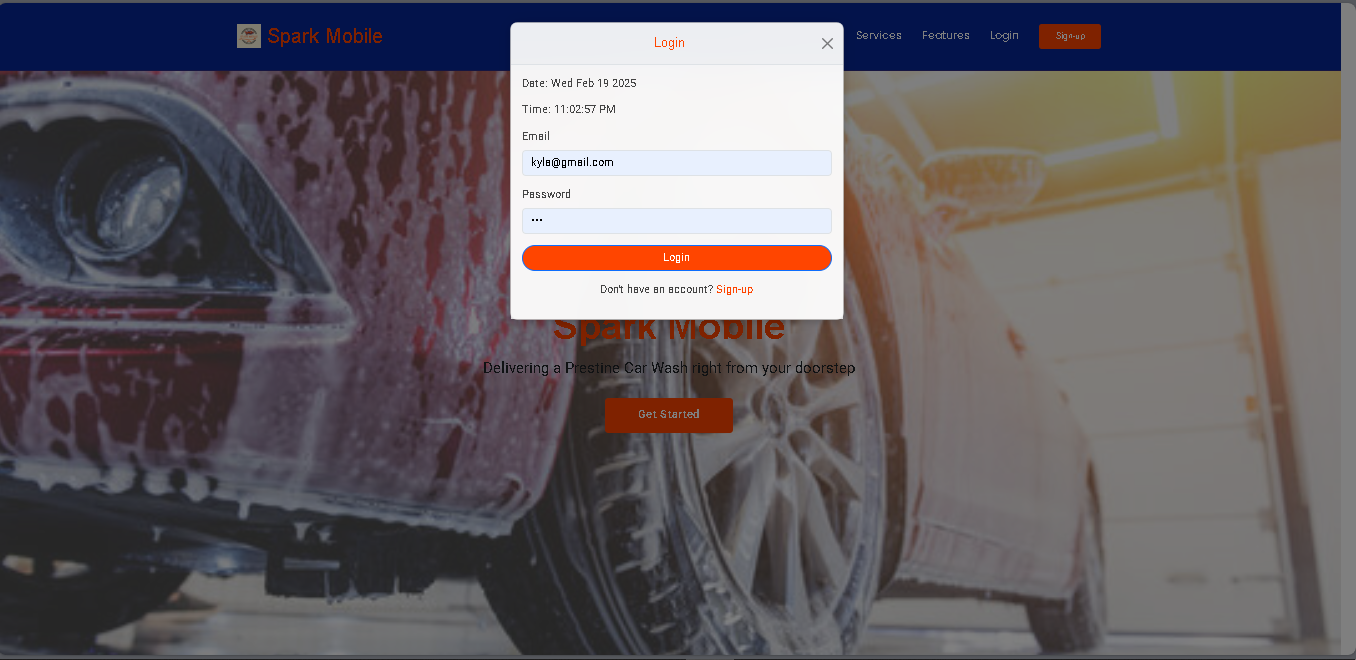
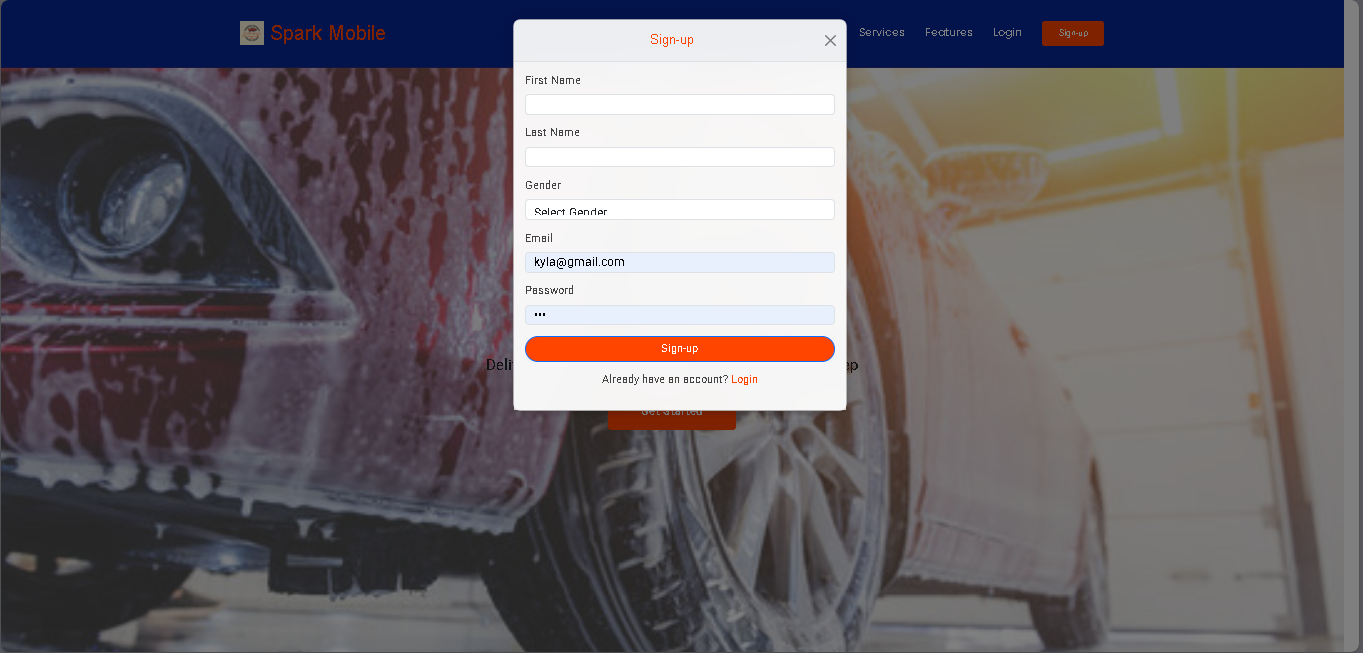
**Audit Logging:** Changes to financial transactions and user data will be logged for accountability.

**Compliance:** The system will follow data protection laws, including the **Data Privacy Act of the Philippines** and **GDPR**, ensuring user data is handled responsibly.

***User Interface***

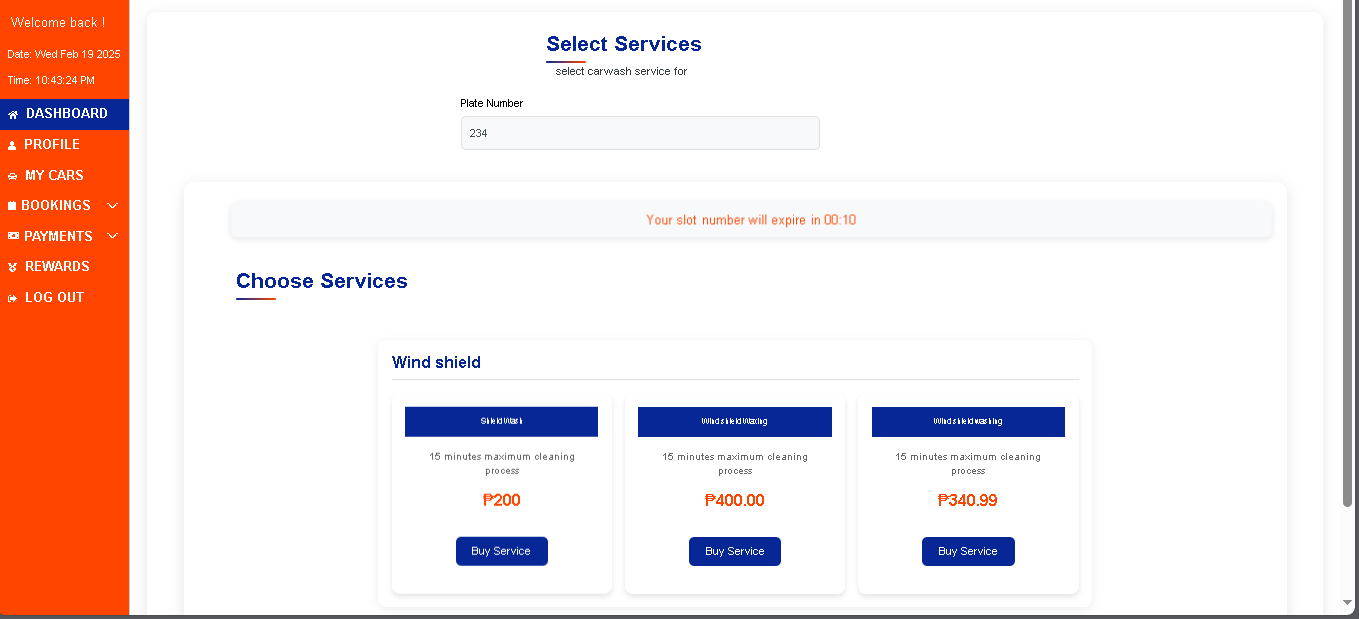
Define the interactions between the system and its users, focusing on how they engage with the interface. Describe user interface elements, such as buttons, menus, and forms, and specify the interactions these elements support, like clicking, typing, or dragging. Clearly outline usability requirements, including ease of use and accessibility, and consider user experience (UX) factors like aesthetics and visual appeal. Select only the UI components that align with the solution's goals, objectives, and innovative features.

UI1: User friendly and easy to use Registration and Log In Interfaces.



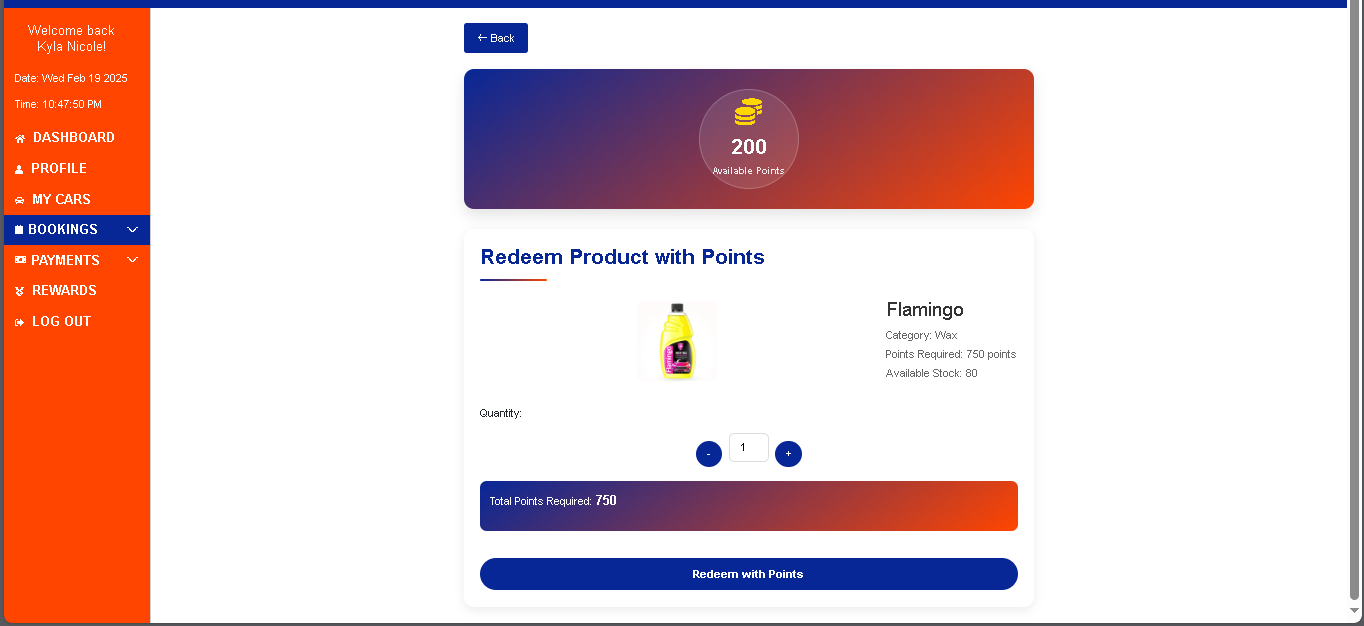
The registration page will include clearly labeled forms needed for the information of the user.

UI2: Automatic service filtering.



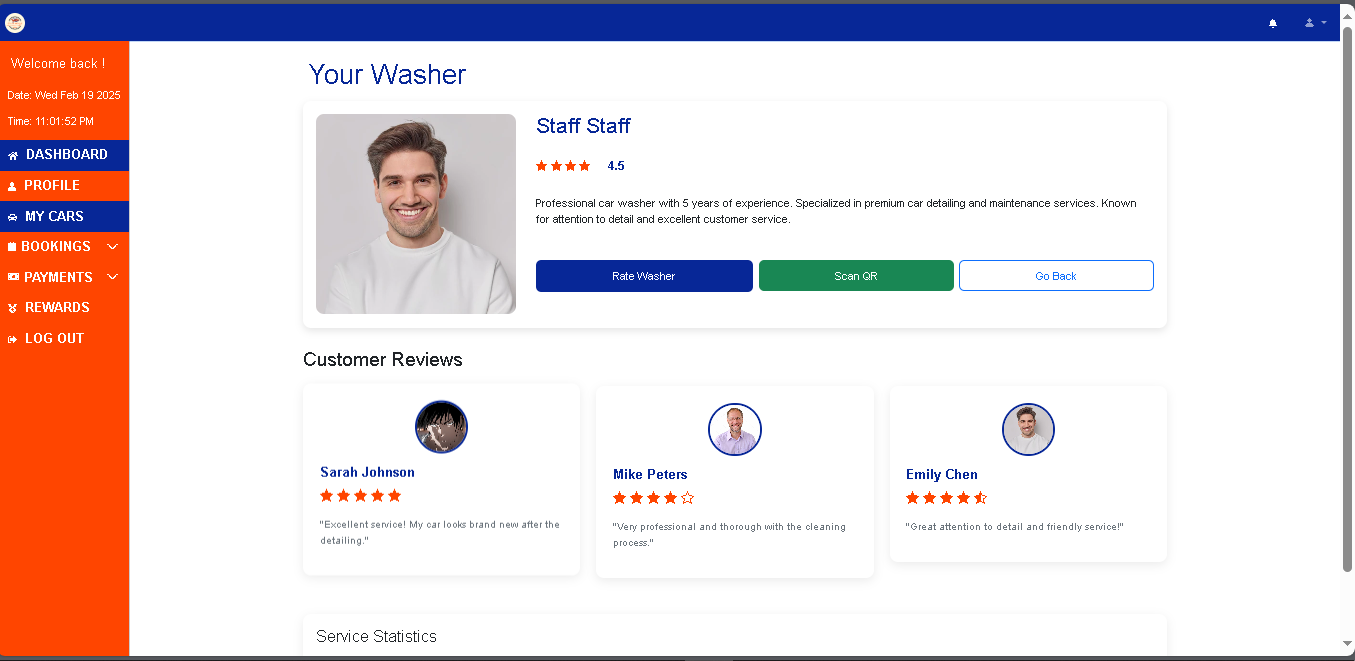
The selecting services page has a detailed services description based on the car appearance of the user’s vehicle so the user can save time from browsing too long finding the best services.

UI3: Cleaning products reward system.



The Cleaning Products Reward System allows users to redeem cleaning products using earned points. The interface displays the available points, product details, required points for redemption, and an option to select quantity before confirming the redemption.

UI4: Feedback and review.



**Non-Functional Requirements**

**Technical Requirements**

T1: Response Time

The system shall ensure an average response time of less than 3 seconds for all key operations, including booking requests, service searches, and payment processing.

T2: Peak Load Capacity

The platform shall be capable of handling a peak load of 1,000 concurrent users without performance degradation.

T3: Browser Compatibility

The system shall be compatible with major web browsers, including Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari. Internet Explorer will not be supported.

T4: Platform Compatibility

The system shall be fully operational on both desktop and mobile web browsers. A native iOS and Android app version may be developed in future updates.

T5: Scalability

The platform shall be hosted on a scalable cloud infrastructure (e.g., AWS, Azure, GCP), allowing seamless expansion to support increasing bookings, transactions, and service providers.

T6: Data Throughput

The system shall be able to process a minimum of 50 transactions per second, ensuring smooth payment processing and booking confirmations.

T7: Integration with Third-Party APIs

The system shall integrate with payment gateways (e.g., Gcash, Maya, PayPal) for secure transactions. Google Maps API for accurate location services. SMS/Email notification APIs to send booking confirmations and updates.

T8: Data Backup and Recovery

Automatic daily backups of critical data (user profiles, bookings, transactions) shall be implemented. In case of failure, the system's Recovery Time Objective (RTO) shall not exceed 2 hours.

T9: Security Measures

The system shall implement HTTPS encryption, ensure secure password hashing with bcrypt, and enforce role-based access control (RBAC) to protect sensitive user data.

T10: Resource Optimization

The platform shall minimize CPU and memory usage to ensure smooth operation, especially for users accessing the system via mobile devices.

T11: Localization

The system shall support local time zones and currency formats based on the user’s location. Multilingual support may be considered in future updates.

**Security Requirements**

S1: Data Encryption

All sensitive user data, including personal details, booking history, and payment information, shall be encrypted both in transit (using TLS 1.3) and at rest (using AES-256) to prevent unauthorized access.

S2: Strong Password Policies

Users shall be required to create strong passwords containing uppercase and lowercase letters, numbers, and special characters. Passwords shall be securely hashed using bcrypt with a high work factor to enhance security.

S3: Role-Based Access Control (RBAC)

The system shall enforce role-based access control, ensuring that customers, car wash staff, and administrators have access only to the necessary system functionalities relevant to their role.

S4: Intrusion Detection and Prevention (IDPS)

An intrusion detection and prevention system shall be in place to monitor system activity, detect unauthorized access attempts, and mitigate security threats such as brute-force attacks and malware.

S5: Regular Security Audits and Penetration Testing

SparkMobile shall undergo periodic security audits and penetration testing, conducted at least quarterly, to identify vulnerabilities and apply necessary security patches.

S6: Secure Session Management

The system shall use secure session management practices, including encrypted cookies, automatic session timeouts, and protection against session hijacking to prevent unauthorized access.

S7: Access Logging and Anomaly Detection

All user activities, including login attempts, payment transactions, and service bookings, shall be logged for security auditing. Anomalous behaviors shall trigger real-time alerts for further investigation.

S8: Protection Against Common Cyber Threats

SparkMobile shall implement security measures to defend against common threats, including SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), and brute-force attacks.

S9: Secure Payment Transactions

All financial transactions shall be processed through trusted and secure payment gateways (e.g., GCash, Maya, PayPal) with end-to-end encryption to prevent fraud and unauthorized access.

S10: Data Backup and Disaster Recovery Plan

Automatic encrypted backups shall be performed regularly to prevent data loss. A disaster recovery plan shall ensure service continuity in case of system failures or cyber incidents.

**Legal Requirements**

To ensure SparkMobile operates in compliance with legal and regulatory frameworks, the platform shall adhere to data privacy laws, consumer protection policies, and financial regulations. These measures will protect user rights, ensure lawful operations, and maintain trust in the system.

L1: Compliance with Data Privacy Laws

SparkMobile shall comply with all applicable data privacy laws, including the Data Privacy Act of 2012 (Philippines) and international standards such as GDPR and CCPA, to protect user data and ensure proper data handling, storage, and consent management.

L2: Consumer Protection Compliance

The platform shall adhere to consumer protection laws to ensure transparent pricing, clear service agreements, and fair treatment of customers, in accordance with the Philippine Consumer Act (RA 7394) or similar applicable regulations.

L3: Intellectual Property Compliance

All content, including branding, images, and software used within SparkMobile, shall comply with copyright and trademark laws. Unauthorized use of third-party intellectual property shall be strictly avoided.

L4: Compliance with E-Commerce and Digital Service Regulations

As an online service provider, SparkMobile shall comply with e-commerce and digital service regulations in the country of operation, ensuring secure transactions, proper invoicing, and compliance with online business operation guidelines.

L5: Tax Compliance

SparkMobile shall adhere to applicable tax regulations, ensuring proper documentation and reporting of revenue, including VAT compliance where necessary.

L6: Secure Payment and Financial Compliance

All transactions processed through SparkMobile shall comply with anti-fraud and financial security regulations, including Anti-Money Laundering (AML) laws, and ensure safe and legal online payments via verified payment gateways (e.g., GCash, Maya, PayPal).

L7: Liability and Service Agreements

Users, including both car owners and service providers, shall agree to SparkMobile’s terms and conditions, which outline liability, dispute resolution, and service warranties to ensure clear expectations and legal protection for all parties involved.

***User Requirements***

To ensure SparkMobile meets the needs and expectations of car owners and service providers, the system shall be designed for usability, accessibility, and convenience. Continuous user feedback shall be incorporated to improve the overall experience.

U1: Ease of Use

SparkMobile shall have a user-friendly and intuitive interface, making it easy for car owners and service providers to navigate, book services, and manage appointments.

U2: Accessibility

The platform shall be accessible to all users, including mobile and desktop users, and comply with web accessibility standards to accommodate users with disabilities.

U3: Flexible Booking System

Users shall be able to schedule, reschedule, or cancel car wash services at their convenience, with real-time availability updates.

U4: Secure and Multiple Payment Options

The system shall provide secure payment methods, including e-wallets (e.g., GCash, Maya), debit/credit cards, and cash-on-service options.

U5: Real-Time Notifications

Users shall receive real-time updates on booking confirmations, service status, and payment transactions through SMS, email, or in-app notifications.

U6: Ratings and Feedback System

Customers shall be able to rate and review car wash providers after each service to help maintain service quality and trust.

U7: Service Provider Profile and Selection

Users shall have access to detailed profiles of car washers, including their ratings, experience, and service history, enabling informed decision-making.

U8: Customer Support and Assistance

SparkMobile shall provide 24/7 customer support through chat, email, or a support hotline to address user concerns and service issues.

U9: Location-Based Service Matching

The platform shall use GPS-based service matching, ensuring customers are connected to the nearest available car wash provider for faster service.

U10: Loyalty and Rewards System

Users shall earn reward points for each booking, which can be redeemed for discounts or free services, encouraging user engagement.

**Development and Testing**

***Development Methodology***

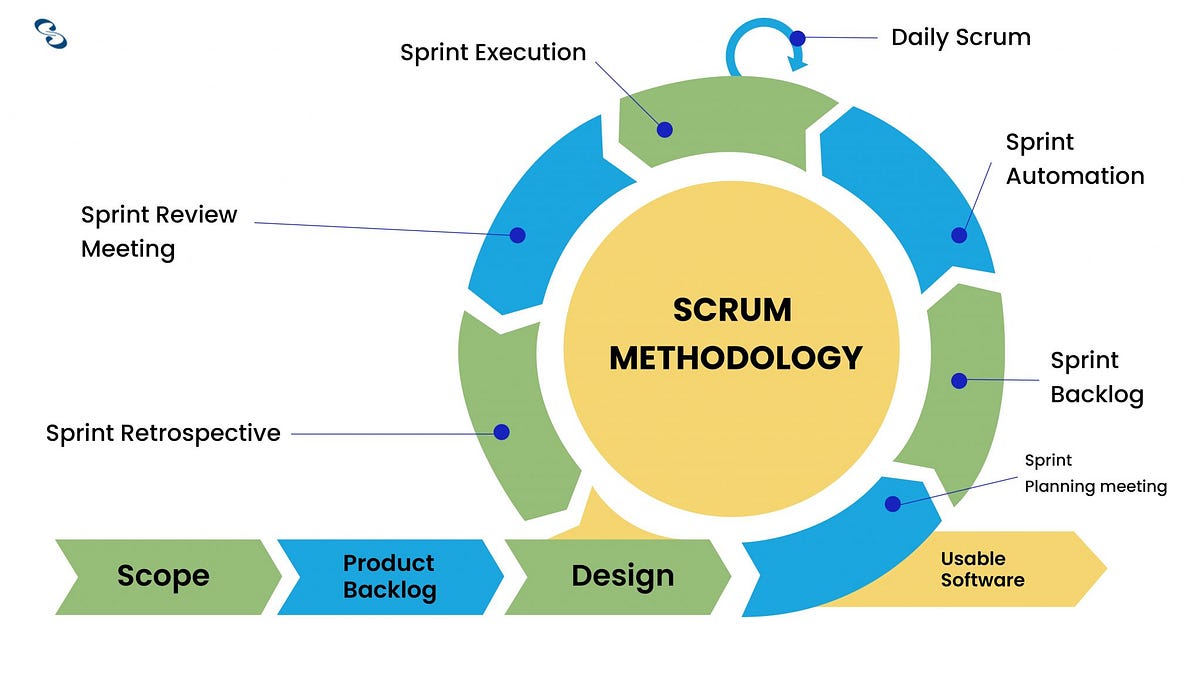


Figure 4.4 Agile Methodology (Scrum)

The development of SparkMobile followed the Agile (Scrum) methodology, which allowed for flexibility, iterative development, and continuous user feedback. Agile was chosen due to the evolving nature of the project and the need to incorporate user-driven improvements throughout development. This approach ensured that SparkMobile could adapt to user needs efficiently while maintaining high system performance and reliability.

***Development Phases***

1. Requirements Gathering:

Requirements were gathered iteratively, focusing on key processes such as user registration, booking system, payment integration, and service provider management. Regular feedback from car owners and service providers helped refine and enhance the platform’s features.

2. Design:

The system was designed with a modular and scalable architecture, ensuring that SparkMobile could support future expansions. User interface (UI) designs were improved during each sprint to enhance usability, accessibility, and responsiveness across different devices.

3. Implementation:

Development was conducted incrementally, with each sprint delivering a functional feature. Key modules such as service booking, real-time tracking, and notification systems were implemented iteratively, allowing for continuous improvements throughout the development cycle.

4. Testing:

Testing was conducted at every sprint to ensure each process functioned as intended. The testing procedures included:

* Unit Testing: To validate individual components, such as the booking system and payment processing.
* Integration Testing: To ensure seamless interaction between modules (e.g., booking, notifications, and payments).
* System Testing: To assess the overall functionality, performance, and security of SparkMobile.
* User Feedback Testing: Conducted during each sprint to identify usability issues and make improvements based on real-world usage.

5. Deployment:

Deployment was carried out in stages after successful sprint completion. Early releases allowed real users to test the platform, provide feedback, and help ensure a smooth transition to the fully functional system.

6. Maintenance:

SparkMobile underwent continuous maintenance and updates during and after development. Security enhancements, bug fixes, and feature updates were incorporated based on user feedback. Regular system monitoring ensured optimal performance and data security.

***Development Environment***

The development of Spark Mobile utilized a range of tools and technologies, ensuring a streamlined and efficient process.

1. ProgrammingLanguages**:**
   * PHP: For server-side scripting and backend development.
   * HTML, CSS: For structuring and styling web pages.
   * JavaScript: For interactive and dynamic user interface functionalities.
2. Development Tools:

* Notepad++: A lightweight and efficient text editor used for coding PHP, HTML, CSS, and JavaScript.
* XAMPP: A local development environment incorporating Apache for the web server, MySQL for database management, and PHP for backend processing.

1. Database**:**

* MySQL: A robust and reliable relational database system used to manage and store platform data, including user profiles, service bookings, and transaction records.

1. CloudDeployment**:**

* InfinityFree: A free cloud hosting platform used to deploy Spark Mobile, making it accessible online for testing and demonstration purposes.

1. OperatingSystem**:**

* Windows: The development and deployment processes were conducted on a Windows-based system, providing compatibility with the chosen tools and technologies.

***Testing Procedures***

1. To ensure the quality and reliability of Spark Mobile, the following testing activities were conducted:
2. Unit Testing**:** Individual components such as user registration, booking system, and payment processing were tested in isolation.
3. Integration Testing**:** The interaction between different modules, such as linking customer bookings with available car wash providers, was tested.
4. System Testing**:** The entire platform was tested to ensure all functionalities, including scheduling, notifications, and payment, worked seamlessly.
5. User Acceptance Testing (UAT): Test users provided feedback to validate usability, efficiency, and performance.
6. Test Data**:** Sample data and real-world booking scenarios were used for validation.
7. Test Metrics**:** Defect density and test pass rate were tracked to measure system stability and reliability.

**Implementation Plan**

***Project Schedule***

A structured timeline was followed using a Gantt chart for efficient tracking of milestones:

* Week 1: Requirements gathering and UI/UX design.
* Week 2: User authentication and profile setup module development.
* Week 3: Booking system and scheduling integration.
* Week 4: Payment gateway implementation and security enhancements.
* Week 5: System testing, debugging, and initial user feedback.
* Week 6: Deployment and user acceptance testing.
* Week 7: Final adjustments and live deployment.

***Resource Allocation***

1. Human Resources:
   * Project Manager
   * Developer
   * Tester
   * UI/UX Designer
2. Hardware Resources:
   * Windows-based development machines with an Intel i3 processor or higher.
   * Local server setup using XAMPP.
   * Internet connection for online deployment and testing.
3. Software Resources:
   * Notepad++
   * XAMPP
   * Cloud hosting (InfinityFree)
   * Browsers for cross-platform testing
4. Financial Resources:
   * Cloud hosting and contingency funds estimated at PHP 2,000.

***Risk Management***

1. **Risk Identification**

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Likelihood** | **Impact** | **Priority** |
| Bugs or errors in booking system | High | High | Critical |
| Data breaches | Medium | High | Critical |
| Budget overruns | Medium | Medium | High |
| Compliance issues | Low | High | High |
| Schedule delays | Medium | Medium | Medium |

1. **Risk Mitigation**
2. Technical Risks:
   * Conduct thorough testing to resolve bugs early.
   * Implement encryption and security best practices.
3. Financial Risks:
   * Allocate contingency funds for unexpected expenses.
4. Operational Risks:
   * Set service level agreements (SLAs) for provider response times.
5. Compliance Risks:
   * Regularly audit for regulatory compliance.
6. **Risk Monitoring**

* Regular Risk Reviews: Bi-weekly meetings to assess risks.
* User Feedback: Continuous feedback collection to address concerns.
* Performance Monitoring: Automated tracking of system performance.

***Communication Plan***

1. **Communication Channels**

* Meetings: Virtual and in-person for team discussions.
* Email: Formal updates and documentation sharing.
* Project Management Software: Trello/Asana for tracking progress.
* Instant Messaging: Quick updates via Gmail/Messenger.
* Documentation Repositories: Google Drive/GitHub for storing project files.

1. **Communication Protocols**

* Clear meeting objectives and documentation.
* 24-hour response time for emails.
* Immediate reporting of critical issues.
* Regular milestone updates to stakeholders.

**Capstone Project Deployment Results**

State the period of the actual deployment of the system. State also the number of respondents and the location. You may also state the experiences during the deployment.

General Objective: *<Objective Statement>*

<Table of Results>

*Table 4.1 <Name of the table>*

*<Image>*

*Figure 4.x <UI in the system where the results are presented>*

Provide a detailed narrative that explains the image, helping readers who may not be familiar with interpreting tables or figures. Begin by citing a related study that employed the same method to achieve similar objectives. Clarify how the objective aligns with the problem statement and its relevance. Describe the method used to assess the achievement of the quantifiable objectives in detail. Discuss whether the objective was met or not, and provide justification for the results.

Specific Objective 1: *<Objective Statement>*

<Table of Results>

*Table 4.1 <Name of the table>*

*<Image>*

*Figure 4.x <UI in the system where the results are presented>*

Provide a detailed narrative that explains the image, helping readers who may not be familiar with interpreting tables or figures. Begin by citing a related study that employed the same method to achieve similar objectives. Clarify how the objective aligns with the problem statement and its relevance. Describe the method used to assess the achievement of the quantifiable objectives in detail. Discuss whether the objective was met or not, and provide justification for the results.

Specific Objective 2: *<Objective Statement>*

<Table of Results>

*Table 4.1 <Name of the table>*

*<Image>*

*Figure 4.x <UI in the system where the results are presented>*

Provide a detailed narrative that explains the image, helping readers who may not be familiar with interpreting tables or figures. Begin by citing a related study that employed the same method to achieve similar objectives. Clarify how the objective aligns with the problem statement and its relevance. Describe the method used to assess the achievement of the quantifiable objectives in detail. Discuss whether the objective was met or not, and provide justification for the results.