Software Requirements Specifications

FUEL XPRESS

Team: Hackelite

Uday Vandanapu

Sahithi Regalla

Kavya Jangapalli

Durga Devi Pampani

Pendyala Pranay

Hemanth Janapala

Karthik Setti

Vaishnavi Adapa

Mounika Eddala

Table of Contents

1. Introduction

1.1 Purpose:

The purpose of the project is to design and develop a web application for delivering fuel to the users through an online platform, making easier and convenient to people to get the fuel they need.

1.2 Scope:

The scope of our project is to development of web application known as "**Fuel Xpress**" for online fuel delivery. This application enables users to place fuel orders online and have them delivered to their exact location. Based on the user's input location, the application assigns the order to the nearest gas station and dispatches a qualified driver to the user's location to fulfill the delivery.

1.3 Goals: efficient fuel delivery, reduced waiting times.

1.4 Benefits: Convenience for users, time savings, useful for users in remote areas

1.5 Objectives: Develop the "Fuel Xpress" application, integrate GPS, payment gateways, order assignment, ensure security, manage drivers.

1.6 Definitions:

Online Fuel Delivery: The process of ordering and receiving fuel through a digital platform without visiting a physical gas station.

Gas Station: A retail facility where customers can purchase fuel for their vehicles.

User Location: The geographical coordinates or address where a user requests fuel delivery.

Order Assignment Algorithm: A set of rules and calculations to determine which gas station fulfills a user's fuel order.

1.7 Acronyms and abbreviations:

FD App**:** Abbreviation for "Fuel Xpress" web application.

GPS**:** Acronym for Global Positioning System.

UI**:** Abbreviation for User Interface, referring to the visual design and layout of the application.

API**:** Acronym for Application Programming Interface, used for integrating location services and payment gateways.

1.8 Reference Documents:

We have referred to the following sources: The IEEE document distributed across modules and the Belitsoft website.

1.9 Overview:

In the next chapter we have written about the Production function, User characteristics, system structure and description.

In third chapter, we have mentioned about the Interfaces, functional and non-functional requirements.

Fourth chapter described about Development phase plan in detail.

2. Overall Description:

2.1 System Structure:

Fuel Xpress website serves as platform for users, gas station and drivers to fulfil fuel needs. It Provides a User Interface where users can Sign up using a form, Login to the website post successful authentication. Several modules such as Order creation, View/Track Orders, Process/Cancel orders, pay for orders, Update Inventory are integrated with the UI using development framework and Database. The website runs on MVC Architecture, where UI components are represented as View, Business logic for several modules represents Model and data that is viewed, processed represents Controller.

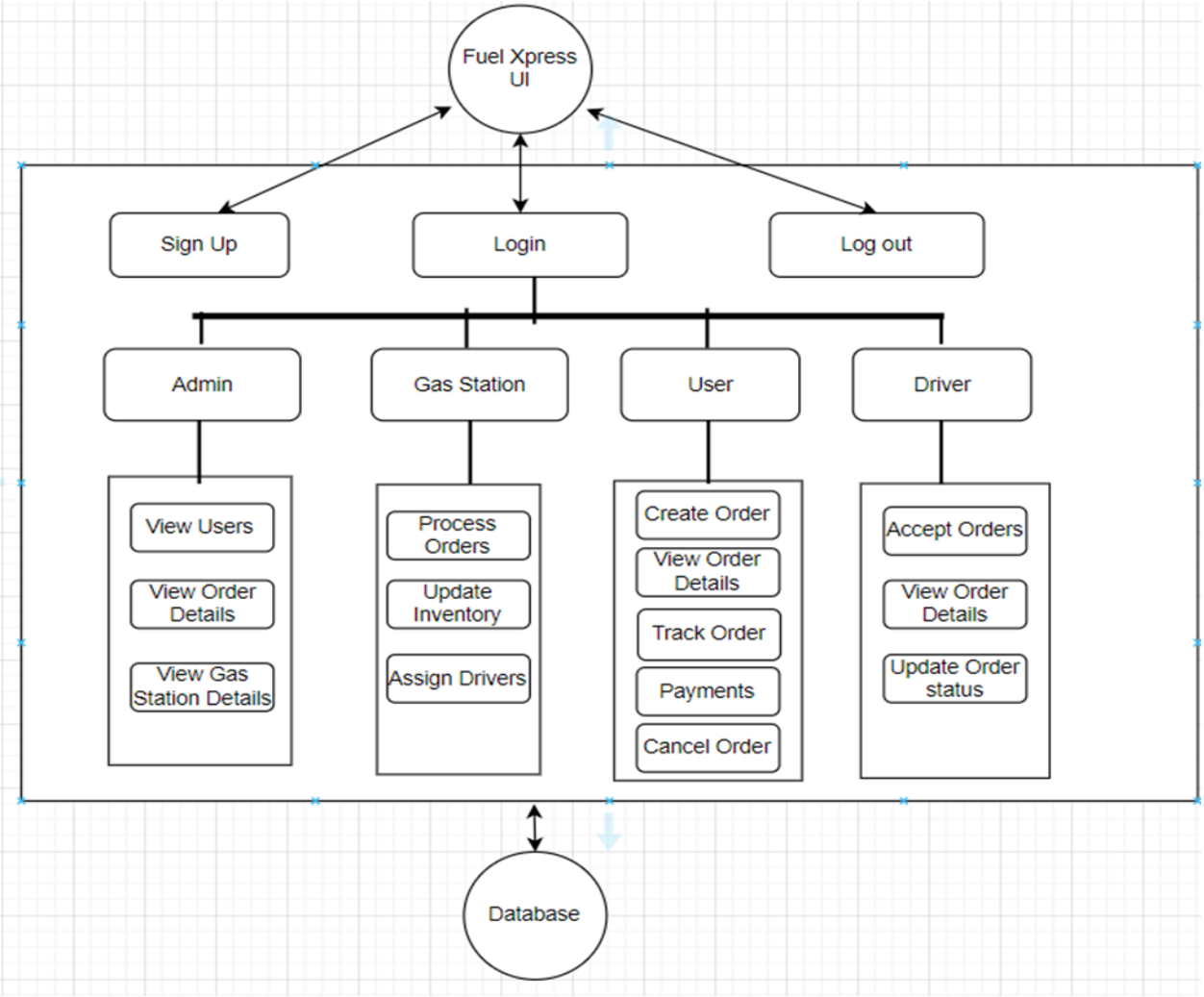

Fig 2.1 System Structure Diagram for Fuel Xpress Website

2.2 Product Functions:

Fuel Xpress is an online fuel delivery platform that serves three primary user groups: Users, Gas Stations, and Drivers. The software offers a range of functions to facilitate fuel delivery, order management, and user interactions:

Sign Up: This module is useful to successfully Onboard Users, gas station and Drivers using a web form.

Login: This module allows Admin, User, Gas Station, and Drivers to login and access the website.

Logout: This module successfully logs out users from the current session.

View Users: This module is used by admin to monitor the list of users present on the website.

View Order Details: This module is useful to view the details mentioned in any fuel order. It can contain details such as Delivery address, Customer details, Order Summary, Order status etc., This module is used by Admin, Users and Drivers.

View Gas Station Details: This module is used by the admin to monitor the list of gas stations present on the website.

Update Inventory: This module is used by Gas station to maintain the Inventory, fuel prices etc.

Process Orders: This Module is used by Gas station to accept or reject the fuel orders placed by customers based on stock, order location etc.

Assign Drivers: This Module is used by Gas station to assign drivers based on their availability to any pending fuel orders.

Create Order: This module is used by customers to create a fuel order. Details such as delivery location, payment method, customer details need to be specified here.

Track Order: Once the order is successfully placed, it can track using this module. This module is integrated with Google API to track the location and provide estimated time to deliver.

Cancel Order: This module is used by the user to cancel the fuel orders within the permitted time.

Process Payments: This module is integrated with payment Gateway and is used to safely authorize the user payments. Acceptable payment modes are set by admin.

Accept Orders: This module is used by drivers to accept the orders assigned to them by Gas stations.

Update Order status: This module is used by drivers to update order status in instances where it is fulfilled or cancelled.

2.3 User Characteristics:

The Fuel Xpress platform is designed to accommodate the following user groups:

1. **Admin:** Admin has comprehensive control and oversight of the entire platform, managing users, gas stations, and orders.
2. **Users:** These are individuals who require fuel delivery services. They can place orders, track deliveries, and manage their accounts.
3. **Gas Stations:** Gas station owners can register their businesses on the platform, manage inventory, process orders, and assign drivers.
4. **Drivers:** Drivers are responsible for fulfilling fuel orders assigned to them. They can accept orders, navigate to delivery locations, and update order statuses.

2.4 Constraints:

- The development must adhere to the chosen MVC (Model-View-Controller) architecture.
- Integration with external systems, such as the payment gateway and Google API, must be implemented securely and reliably.
- The system must handle sensitive user data securely to ensure privacy and data protection.
- Availability of drivers and gas station inventory may affect order processing and delivery times.

2.5 Assumptions and Dependencies:

- The system assumes that users will provide accurate and valid information during the registration process.
- It depends on external services, such as the payment gateway and Google API, to provide essential functionality.
- The platform assumes that gas stations will keep their inventory up to date and manage orders efficiently.
- Admin assumes the responsibility of managing and maintaining the overall functionality of the platform, including user and gas station registration.

3. Requirements Specification

3.1 Functional Requirements

Functional Requirement 1:

Title: **User Registration**

Description: To order fuel the first step is user Registration, and the user can create an account by providing First Name, Last Name, Mobile number, Email, Password, and Delivery Address. Users can create an account with a valid email and password.

Functional Requirement 2:

Title: **Authentication**

Description: User can log in safely by using their email and password. Here the password should be confidential.

Functional Requirement 3:

Title: **Profile Management**

Description: Here User can be able to change the personal information first name, Last name, Delivery address, and contact details.

Functional Requirement 4:

Title: **Admin Page**

Description: The Monitoring Orders part of the admin interface gives administrators the opportunity to analyze, manage, and track orders by grouping them according to the location, dates, and status that users have booked.

Monitoring of users and petrol stations: The system administrator keeps a close eye on user behavior and makes sure that the petrol stations are running well.

Functional Requirement 5:

Title: **Drivers Page**

Description: For quick service, drivers accept incoming orders and place them in a queue.The driver will be given complete delivery information, including the client's location, client data, and vehicle identification number. using the Google Maps API to navigate to the user's location. A record of the hours worked is available to drivers.

Functional Requirement 6:

Title: **Gas station**

Description: With this feature, you can view and control incoming fuel orders, get alerts for new orders, and access order information including

delivery addresses, fuel kinds, and quantity requests. The petrol station is looking for experienced drivers. For usage in the delivery procedure, petrol stations can register their delivery cars online. Orders may be cancelled if there is not enough gasoline available.

Functional Requirement 7:

Title: **Fuel Selection**

Description: Here the user can select the type of fuel and quantity of the fuel.

Functional Requirement 8:

Title: **Delivery Location**

Description: The user needs to indicate the delivery location by adding an Address.

Functional Requirement 9:

Title: **Order Confirmation**

Description: The user can review order confirmation on the user interface page including order details and payment information.

Functional Requirement 10:

Title: **Order Tracking**

Description: Once the order is placed the User can Track the order status.

Functional Requirement 11:

Title: **Order History**

Description: The user can access the complete order history that includes the type of fuel, cost, and payment data.

Functional Requirement 12:

Title: **Payment Processing**

Description: The system securely processes payments from customers using many sorts of payment alternatives like debit cards, credit cards, or wallets.

Functional Requirement 13:

Title: **Route Optimization**

Description: The system optimizes the delivery routes to various customers to deliver the fuel, it saves time and fuel for the delivery agent.

Functional Requirement 14:

Title: **Quantity Measurement**

Description: The system calculates the quantity of fuel delivered to the users.

Functional Requirement 15:

Title: **Data Security**

Description: The System must protect the user's data like the user's personal information and payment Details.

Functional Requirement 16:

Title: **Privacy**

Description: A privacy statement describes the system’s procedures for gathering, using as well and protecting collected data.

Functional Requirement 17:

Title: **Support Assistance**

Description: If the user facing any problem while placing an order, payment issues or any other issues user has the option to contact the support assistance team.

3.2 Non-Functional Requirements

1. Performance:

Response Time: All the pages of the website should be loaded in maximum of 3 seconds, and placing an order should be done in 5 seconds.

Scalability: The system needs to handle a growing number of simultaneous users, Especially during periods of high demand, the system should be able to accommodate a larger number of simultaneous users without experiencing a noticeable performance decline.

2. Availability and Reliability:

Uptime: The website should be available to users around the clock, aiming for a minimum uptime of 99.9%

Fault Tolerance: The system must continue to function even when hardware failures or minor software issues occur.

Data Backup: Scheduled automated backups must be conducted to safeguard against data loss in case of system failures.

3. Security:

Data Encryption: All data exchanged between clients and servers, especially sensitive information like user credentials and payment details, must undergo robust encryption, employing protocols like HTTPS to ensure security.

Authentication: Implement strong authentication mechanisms to guarantee that only authorized users are able to access the system.

Authorization: Establish access control policies to limit users to specific areas and functionalities based on their assigned roles.

Penetration Testing: Conduct routine security evaluations, penetration tests, and code reviews to detect and rectify potential vulnerabilities.

4. Usability and Accessibility:

User-Friendly Interface The website should include a user-friendly and visually appealing interface designed to cater to the needs of all user roles, including administrators, gas station staff, drivers, and customers.

Accessibility: Ensure that the website complies with WCAG guidelines, making it accessible to users with disabilities.

Responsiveness: The website should be responsive and perform effectively on a variety of devices, including smartphones and tablets.

5. Logging:

Logging: Implement thorough logging of system activities, including user actions, error occurrences, and security events.

6. Payment Gateway Performance:

Payment Gateway Reliability: Ensure that the chosen payment gateway is highly dependable, offering minimal downtime and swift transaction processing.

3.3 Interfaces:

3.3.1 User Interface:

User Registration and Login: New users can sign up with their credentials and subsequently log in using the provided details.

Fuel Selection: Users can choose the fuel type, quantity, and delivery location.

Order Placement: After making selections, users can proceed to order fuel.

Order Cancellation: Users have the option to cancel their orders.

Order Tracking: Users can track the status of their orders.

Payment Processing: users are directed to the payment section, enabling them to utilize an API payment gateway for online payment processing.

Admin Interface:

Monitoring Orders: Within this section, administrators have the capability to review, handle, and track orders by sorting them based on location, dates, and status booked by users.

Monitoring users and gas stations: The administrator actively supervises user activities and ensures the efficient operation of gas stations within the system.

Driver Interface:

Order Acceptance: Drivers accept incoming orders and add them to a queue for efficient service.

Delivery Information: The driver will receive comprehensive delivery details, including the customer's location, customer information, and vehicle number. Traveling to the user's location with the assistance of the Google Maps API. Drivers can access a record of the hours they have worked.

Gas station Interface:
Order Handling: This functionality allows to view and manage incoming fuel orders, receive notifications for new orders, and access order details such as delivery addresses, requested fuel types, and quantities. The gas station recruiting skilled drivers. Gas stations can register their delivery vehicles online for use in the delivery process.
Inventory Management: Gas station administrators can adjust inventory levels in response to orders, reducing fuel quantities accordingly.
Order Cancellation: Orders can be cancelled when there is an inadequate supply of fuel

3.3.2 Hardware Interface:

- Laptop
- Fuel Dispensing equipment

3.3.3 Software Interface:

- Payment gateway Interface: This interface facilitates secure online payments through a payment gateway.
- Google API for precise location tracking and seamless data integration, our system ensures accurate geo-location information, enabling advanced features and functionalities that enhance user experiences and streamline operations.

3.3.4 Communication Interface:

- Communication protocols - System interfaces often specify the communication protocols or rules that govern how data is transmitted between components. This includes details such as data format, message structure, encoding, and network protocols (e.g., HTTP, TCP/IP)
- Live chat. Support, email notifications, SMS alerts, in-app messaging, order status updates

3.3.5 Memory Constraints:

- Supports 32-bit, 64-bit systems.
- Random Access Memory (RAM) - 8,16 GB
- Read Only Memory (ROM)
- Mass storage devices (like hard drives, SSDs)
- External memory (memory cards)

3.3.6 Operations:

Security Considerations: Security measures related to the operation interface, such as authentication, authorization, and encryption, are often described to ensure the secure usage of the functionalities.

Error Handling: The interface provides a clear plan for dealing with errors or unexpected issues during an operation. It lays out the error codes and messages, along with what should happen when errors or failures occur.

4. **Development Phase Plan**

Phase 1: Core Functionality Development

In Phase 1, the focus is on establishing core functionalities and security measures.

Module 1: User Registration and Authentication

- Submodule 1.1: User Registration
    - Description: Create a user-friendly registration form to capture user details securely.
- Submodule 1.2: User Authentication
    - Description: Implement a robust authentication system for secure user logins.
- Submodule 1.3: Database Setup
    - Description: Design and create a secure database schema for user profiles and authentication data.

Module 2: Admin Interface Development

- Submodule 2.1: Admin Dashboard
    - Description: Build an admin dashboard for order monitoring and management.
- Submodule 2.2: Order Management
    - Description: Develop order tracking and management features for administrators.
- Submodule 2.3: User Management
    - Description: Create tools for admin access and management of user profiles.
- Submodule 2.4: Gas Station Management
    - Description: Enable admin oversight of affiliated gas stations.
- Submodule 2.5: Database Enhancements
    - Description: Extend the database to include admin-specific data.

Phase 2: User and Gas Station Interfaces

Phase 2 expands the platform by introducing interfaces for users, gas stations, and drivers.

Module 3: Gas Station Interface Development

- Submodule 3.1: Gas Station Dashboard
    - Description: Provide a user-friendly dashboard for gas stations to manage inventory and orders.
- Submodule 3.2: Inventory Management
    - Description: Develop inventory tools for efficient management.
- Submodule 3.3: Order Processing
    - Description: Enable gas stations to process orders seamlessly.
- Submodule 3.4: Database Enhancements
    - Description: Enhance the database for gas station-specific data.

Module 4: Driver Interface Development

- Submodule 4.1: Driver Dashboard
    - Description: Create a driver dashboard for order acceptance and navigation.
- Submodule 4.2: Order Acceptance
    - Description: Develop features for drivers to accept and manage orders.
- Submodule 4.3: Navigation and Delivery
    - Description: Implement navigation tools for secure fuel delivery.
- Submodule 4.4: Database Enhancements
    - Description: Extend the database for driver-specific data.

Phase 3: Integration and Testing

Phase 3 focuses on integration, testing, debugging, optimization, and final preparations.

Module 5: Integration of User, Gas Station, and Driver Modules

- Submodule 5.1: Integration of User and Gas Station Modules
    - Description: Combine user, gas station, and driver modules for seamless integration.
- Submodule 5.2: Integration of Driver Modules
    - Description: Integrate driver-specific functionalities.

- Submodule 5.3: Integration Testing
    - Description: Verify cohesive operation through integration testing.

Module 6: Final Debugging and Optimization

- Submodule 6.1: Debugging and Issue Resolution
    - Description: Identify and resolve integration issues and defects.
- Submodule 6.2: Performance Optimization
    - Description: Optimize the platform for performance and efficiency.

This structured approach ensures a systematic development of the online fuel delivery platform, focusing on core functionality, user interfaces, integration, and quality assurance.

5. Member Contribution Table:

Team Members	**Responsibilities**
Karthik Setti	Requirement Specifications (Functional Requirements)
Durga Devi Pampani	Introduction, System Interfaces
Uday Vandanapu	Meeting Minutes
Vaishnavi Adapa	Development Phase Plan
Sahithi Regalla	System Structure
Kavya Jangapalli	Requirement Specifications (System Interfaces)
Mounika Eddala	Note-Deliverable 2
Pranay Pendyala	Member Contribution Table
Hemanth Janapala	Requirement Specifications (Non-Functional Requirements