

Requirement Specification

Version 1.0

Form Number: BS23-QMS-1.0

Dated: 05/11/2019

REVISION HISTORY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ver. No** | **Date of Release** | **Prepared By** | **Reviewed / Approved By** | **List of changes from Previous Version** |
| 1.0 | 20-08-2020 | Md. Rakibul Alam Tanvi | MJ Ferdous | Draft Creation |
| 1.2 | 02-09-2020 | Md. Rakibul Alam Tanvi |  | Incorporate Anesh feedback/update. |
| 1.3 | 03-09-2020 | Md. Rakibul Alam Tanvi | Anesh Roopnarine |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |



Requirement Specification

Version 1.0

<PROJECT NAME>

Document Number: <Project Name>-SRS-1.0

QMS Document Reference: BS23-QMS-1.0

QMS Document Version Reference: 1.0

Dated: <dd/mm/yyyy>

REVISION HISTORY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ver. No | Date of Release | Prepared By | Reviewed / Approved By | List of changes from Previous Version |
| 1.0.0 | <dd-mm-yy> |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Table of Contents

[1. Introduction 2](#_Toc49967843)

[2. Purpose of the Document 3](#_Toc49967844)

[3. Definitions, Acronyms and Abbreviations 4](#_Toc49967845)

[4. References 4](#_Toc49967846)

[5. Role and responsibility 5](#_Toc49967847)

[6. Overview of the system 5](#_Toc49967848)

[7. Current System 5](#_Toc49967849)

[1.1.1 Bespoke Software Focus 5](#_Toc49967850)

[1.1.2 Ecosystem - Partial 6](#_Toc49967851)

[8. Scope of the Proposed System 7](#_Toc49967852)

[Release 1 8](#_Toc49967853)

[Release 2 8](#_Toc49967854)

[Release 3 8](#_Toc49967855)

[9. Proposed Tools and Technology 9](#_Toc49967856)

[10. Software Requirements Details 9](#_Toc49967857)

[Functional Requirements 9](#_Toc49967858)

[11. Non-Functional Requirements 73](#_Toc49967859)

[12. Performance Requirements 73](#_Toc49967860)

[Security Requirements 73](#_Toc49967861)

[13. Third-Party Integration or requirements 73](#_Toc49967862)

[14. External Interface Requirements 73](#_Toc49967863)

[15. User Interfaces 74](#_Toc49967864)

[16. Hardware Interfaces 74](#_Toc49967865)

[17. Software Interfaces 74](#_Toc49967866)

[18. Communication Interfaces 74](#_Toc49967867)

[19. Environment considerations 74](#_Toc49967868)

[Development and Production environment 74](#_Toc49967869)

[Backup and Recovery Plan 75](#_Toc49967870)

[20. Assumptions and Dependencies 75](#_Toc49967871)

[21. Maintainability Requirements 75](#_Toc49967872)

[22. High Availability (HA) Requirements 75](#_Toc49967873)

[23. Appendix A: Glossary 76](#_Toc49967874)

# Introduction

Unipet is interested in developing an electronic payment platform for the purchase of fuel, as well as other products retailed by the Company, as part of its digital transformation strategy roadmap. Currently the Company utilizes a payment solution which is primarily geared towards companies which own and manage a fleet of vehicles and require fuel management support. This solution is owned and managed by a third-party vendor and has several inherent issues such as unreliable equipment, lack of point of sale integration, a weak driver verification process etc.

In the first quarter of 2020, Unipet subscribed to a white label fleet software solution which addresses many of the issues described above. However, this solution also has its limitations which prevents the Company from achieving its strategic objectives. This has led to the decision to develop a platform customized to satisfy the Company's needs.

As a key component of the roadmap, the electronic payment platform will be developed in three (3) phases which are detailed below:

Phase 1: Fleet fuel platform geared towards fleet companies

Phase 2: Consumer platform and enhanced fleet platform

Phase 3: Additional enhancements of both consumer and fleet platforms

Due to the subscribed white label software's ability to integrate to Unipet point of sale system currently, the platform to be developed must initially integrate with the white label software using Application Programming Interfaces (APIs - available from the white label solution provider) in Phases 1 and 2. When Phase 3 is implemented, the white label software shall be removed from the ecosystem.

# Purpose of the Document

This document describes high level software requirements This document describes high level software requirements of Bespoke Software Development – Fleet Web & Mobile Application Development. It contains functional, non-functional, technical requirements as well as features and UI screenshots. Readers of the document will be able to know Overview of the system, Software Requirements Details, Third-Party Integration or requirements, System Interfaces, Environment considerations, Performance Requirements.

# Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| Item | Description/ Expansion |
| Admin Panel | Term used in this document to provide and highlight CMS Admin Panel coverage. |
| API | Application Programming Interface |
| BS or BS23 or BS-23 | Brain Station 23 Ltd. Company carrying out the development of the software |
| Dev. Team | Development Team |
| IDP | Identity Provider, a centralized application built by BS23 to facilitate SSO functionalities of different applications serving end-customers (e.g. mobile number) |
| LAMP | Linux, Apache, MySQL, PHP/Perl/Python, a software bundle. |
| MSISDN | Mobile Station International Subscriber Directory Number. It defines unique number for each subscriber in terms of mobile number. |
| PM | Project Manager |
| PO | Product Owner |
| POC | Point of Contact |
| SEO | Search Engine Optimizations |
| SM | Scrum Master |
| Sr. | Senior |
| SRS | Software Requirement Specification |
| SSO | Single Sign On |
| TL | Tech Lead |
| PIN | Personal Identification Number Code |
| PUK | Personal Unblocking Key code |
| SIM | Subscriber Identification Number |

# References

*N/A*

# Role and responsibility

|  |  |  |  |
| --- | --- | --- | --- |
| Sl | Member | Organization | Role & Responsibility |
| 1 | Anesh Roopnarine,  Business Transformation Programme Manager | Unipet | **POC**  Acting as a point of contact between Unipet team and Brain Station Team.  Responsible for arranging meeting, workshop, discussion session etc. |
| 2 | Md. Rakibul Alam, Project Manager | Brain Station 23 | **PM**  As a PM, he will be responsible to manage the project team. Will prepare and manage project plan and ensure project completion within the budget and time. |
| 3 | Jahangir Alam,  Lead Engineer | Brain Station 23 | **TL**  As a Technical Lead, he will ensure all the technical support required for the team. He will prepare and manage Solution Architecture Design Document, Database Design Document etc. |

# Overview of the system

As part of Unipet’s digital strategy roadmap, several software solutions will be developed concurrently. The diagrams below are intended to provide a high-level view of the focus for the software development in this request for proposal.

# Current System

## Bespoke Software Focus

Focus for this proposal

Future developments

Bespoke Software

Phase 3

Phase 2

Phase 1

Release 3

Release 2

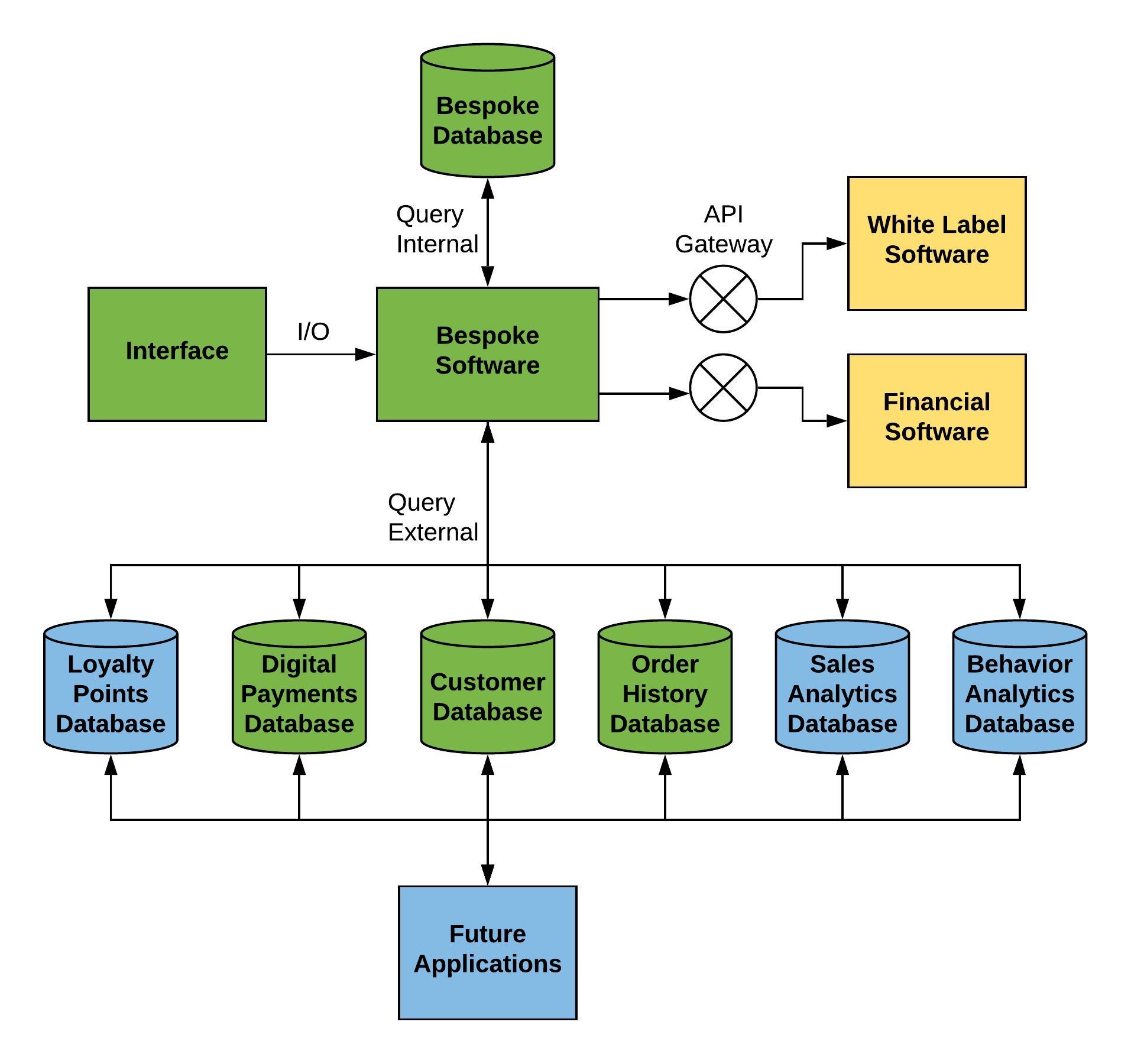
Release 1

## Ecosystem - Partial

To be developed in Phase 1

External software integration in Phase 1

Future developments



Note: “Future Applications” development is not only limited to Phases 2 and 3 of the bespoke software products but also to digital transformation solutions that will share common databases.

# Scope of the Proposed System

This document focuses only on Phase 1 of the development of the platform. Phase 1 concentrates on the development and implementation of a fuel management system to cater for various sizes and types of fleet companies and consists of three (3) releases which are expanded on below:

## Release 1

The initial specifications for Release 1 are outlined in the document entitled “Specifications for Convenience Pay Interface Software - Release 1” attached with the request for proposal.

Some of the key outputs of Release 1 are:

1. Web Interfaces for Fleet Company Managers, Fleet Drivers, Dealer/Merchant (Gas Station Owner), Unipet Administration Department and Unipet Finance Department
2. Integration with white label software through APIs.
3. Digital wallet for management of cash and fuel deposits, as well as transfer of funds.

## Release 2

The initial specifications for Release 2 are outlined in the document entitled “Specifications for Convenience Pay Interface Software - Release 2” attached with the request for proposal.

Some of the key outputs of Release 2 are:

1. Migration of database from white label software: At this point, everything captured in the white label software should duplicated in the bespoke software hosting environment.
2. Added control features on web interfaces.
3. Mobile application for Fleet Company Manager and Fleet Drivers.

## Release 3

The initial specifications for Release 3 are outlined in the document entitled “Specifications for Convenience Pay Interface Software - Release 3” attached with the request for proposal.

Some of the key outputs of Release 3 are:

1. Mobile application for Unipet Administration Department, Finance Department and Dealer/Merchant
2. Capture of and notifications for vehicle demerit points, driver's license renewal, vehicle insurance renewal, etc.

# Proposed Tools and Technology

|  |  |  |
| --- | --- | --- |
| **SL** | **Item Nam** | **Description** |
| 1 | Front End | Angular 7.0/8.0 |
| 2 | Back-End | Java, Spring Boot, AKS (Azure Kubernetes Service) |
| 3 | Database | MS Sql |
| 4 | Admin Portal | Custom Admin Panel |
| 5 | Operating System | Azure Cloud |
| 6 | Web Server | Tomcat |
| 7 | Cache | Redis |
| 8 | Others |  |
| 9 | Development Methodology | Microservice, Cloud Native |

# Software Requirements Details

## Functional Requirements

|  |  |
| --- | --- |
| **Requirement ID** | **2.0** |
| **Requirement Title** | **Administrative Interface** |
| **Requirement ID** | **2.1** |
| **Requirement Title** | **Customer Contract Management** |
| **Requirement ID** | **2.1.1** |
| **Requirement Title** | **Customer Enrolment** |

|  |  |
| --- | --- |
| **Requirement ID** | **2.1.1.1** |
| **Requirement Title** | Automated Customer Enrolment |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | New Fleet Card customers will be able to apply for registration through the New Bespoke Software Public website. Where all the necessary data and documents of the customer and contact will be captured. All information will both store in New Bespoke Database and as well existing ATIOnet/White Label Software database.  The enrolment interface must ensure the following accuracy:   * Clear and brief information on each field. * Customers can Upload any documents required in the enrolment process. * The provision of values from which to select. If required. * Rules governing the completion of fields. * Data type validation. * Ensuring mandatory fields are filled.   After completing the enrolment process Unipet Administrator will get alert/notification through the system. |
| **API** |  |
| **Data Definition** | |
| |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Mandatory** | **Default value (for inaccessible Field)** | **Remarks (Show/Hide Condition)** | | Customer Code | Yes |  |  | | Active | Yes | No | Options: Yes or No | | Customer Type (Fleet | Consumer (Not Used)) | Yes | Fleet | *<e.g. Only Display the following If Customer Type == Fleet>* | | *1. Industry* | | *2. Primary Contact for Fleet* | | *3. Vat Registration Number* | | Customer Name – Company Name in the case of a Fleet Customer | Yes |  |  | | Industry | Yes | Please Select Industry | Drop Down Menu with the following list to select from: | | For Consumer or primary contact for Fleet Customer: (Customer Surname | First Name) | Yes |  |  | | Customer Address (Address line1,2 | City/Town | Country) | Yes |  |  | | Address Line 1 | Yes |  |  | | Address Line 2 | Yes |  |  | | City/Town | Yes |  |  | | Zip Code |  |  |  | | Country | Yes | Please Select Country |  | | Customer Email Address | Yes |  |  | | Customer Telephone Number 1 | Yes |  |  | | Customer Telephone Number 2 |  |  |  | | VAT Registration Number | Yes |  |  | | LinkedIn Account |  |  |  | | Facebook Account |  |  |  | | Instagram Account |  |  |  | | Banking Information ( Bank; Account Number) |  |  |  | | |
| **Database Storage** | * Bespoke Software * White Label Software   **Data Store both in Bespoke & White label Software.** |
| **Dependency** | N/A |

|  |  |
| --- | --- |
| **Requirement ID** | **2.1.1.2** |
| **Requirement Title** | Application Acceptance |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | After submitting customer enrollment application, the following steps will be followed for application acceptance:  Steps:   * Step 01: Unipet Administrator will get an alert/notification. As well shown on the Administrator interface. * Step 02: Unipet Administrator will able to review/check the application for completeness and accuracy. * If all the data and documents are correct then, Enrolment will happen automatically. * But if the data and documents are incomplete/inaccurate, the Unipet Administrator will update information or send back to the customer for correction or completion. |
| **API** |  |
| **Database Storage** | * Bespoke Software * White Label Software   **Data Store both in Bespoke & White label Software.** |
| **Dependency** | 2.1.1.1 Automated Customer Enrolment |

|  |  |
| --- | --- |
| **Requirement ID** | **2.1.1.3** |
| **Requirement Title** | Contract Preparation |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Contract document will auto generate after automatic acceptance of customer enrollment   * Contact (Terms and Conditions) document will auto generate using the enrollment information provided. The customer selects ‘Agree to be enrolled’ * Customer get successfully registered email. * Also get the document (Terms and Conditions) in customer fleet interface. |
| **API** |  |
| **Data Definition** | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Field** | **Mandatory** | **Default value (for inaccessible Field)** | **Remarks (Show/Hide Condition)** |  | | Active |  | No | Options : Yes or No |  | | Company Name | Yes |  |  | Select from drop down list or while company is registering automatically add based on previous value | | Contract Code | Yes |  |  |  | | Description | Yes |  |  |  | | Contract Start Date | Yes |  |  |  | | Contract Duration | Yes | 3 | years | Options: Months, Years | Set for 3 Years on System | | Current Account Mode | Yes | Money |  | Both options should be available - Money & Product. The end user will never see this field | | Currency | Yes | TTD |  | List of options from Ationet | | Payment Mode (Credit | Debit | Cash) | Yes | Debit |  |  | | Balance Mode (Disperse | Do not Disperse | Autofill) | Yes | Disperse |  |  | | Sites Validation | Yes | No | Options: Yes or No | Yes turns on the feature. If Yes, 'Site Information' option below becomes available | | Validate fuels | Yes | No | Options: Yes or No | Yes turns on the feature. If Yes, 'Fuel Information' option below becomes available | | Fuel Information |  |  |  |  | | Fuel |  |  |  |  | | Price |  |  |  |  | | Money |  |  |  |  | | Volumn Unit |  |  |  |  | | Site Information |  |  |  |  | | Site Name |  |  |  |  | | Fuel (Value) |  |  |  |  | | Reload Maximum |  |  |  |  | | Reload Minimum |  |  |  |  | | |
| **Database Storage** | * Bespoke Software * White Label Software   **Data Store both in Bespoke & White label Software.** |
| **Dependency** | 2.1.1.1 Automated Customer Enrolment 2.1.1.2 Application Acceptance |

|  |  |
| --- | --- |
| **Requirement ID** | **2.1.1.4** |
| **Requirement Title** | Contract Execution |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Customer able to add electronic signature through the application.  Steps:   * Step 01: Customer will upload/attach signature through the system. Agree to Terms and Conditions will suffice. |
| **API** |  |
| **Database Storage** | * Bespoke Software * White Label Software   **Data Store both in Bespoke & White label Software.** |
| **Dependency** | 2.1.1.1 Automated Customer Enrolment 2.1.1.2 Application Acceptance 2.1.1.3 Contract Preparation |

|  |  |
| --- | --- |
| **Requirement ID** | **2.1.1.5** |
| **Requirement Title** | Enrolment Invoice |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | After getting the signed copy from the customer, Unipet Administrator will able to send information finance interface for enrollment invoice generation.  Steps:   * Step 01: Enrollment invoice will be calculated based upon information entered in completing the application form. ***e.g*. Fleet Size.** * Step 02: Enrolment Invoice should be automatically emailed to the customer as well as shown on the customer fleet interface. * Step 03: The customer will get the following information/instructions with the invoice   + Unipet's account number.   + Instructions for direct deposit.   + Instruction for payment by Credit Card. |
| **API** |  |
| **Database Storage** | * Bespoke Software * White Label Software   **Data Store both in Bespoke & White label Software.** |
| **Dependency** | 2.1.1.1 Automated Customer Enrolment 2.1.1.2 Application Acceptance 2.1.1.3 Contract Preparation 2.1.1.4 Contract Execution |

|  |  |
| --- | --- |
| Requirement ID | **2.2** |
| Requirement Title | **Card Management** |

|  |  |
| --- | --- |
| **Requirement ID** | **2.2.2** |
| **Requirement Title** | Card Generation Request |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | The Unipet Admin able to request for card generation for customer:  Steps:   * Step 01: Admins select customer from customer list. The white label software Contract Number will also be displayed here. * Step 02: Admin also define number of cards required. * Step 03: Log will generate and Submit ATIONet for card generation. * Step 04: Generate * Step 05: ATIONet system prepares Card information, as requested. The activation code on the Cards is set on. The Card is assigned to the Customer Contract. * Step 06: Log will generate based on ATIONet response. * Step 07: A notification will send to Fleet/driver of card generation. * Step 08: Admin also get a notification of card generation. * Step 09: Admin will receive email in an excel format that can be downloaded and used to print physical cards. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Data Definition** | |
| |  |  |  | | --- | --- | --- | | **Field** | **Mandatory** | **Comments/Note** | | Type | Yes | This is the creation of cards. Can bulk create also and automactially be done by acceptance of a Fleet company request | | Model | Yes |  | | User Type |  |  | | Community |  |  | | Program |  |  | | Contract |  |  | | Label | Yes | This should be generated | | Track Number | Yes | This should be generated | | PAN |  | This should be generated | | Expiration |  |  | | |
| **Dependency** | 2.1.1.1 Automated Customer Enrolment 2.1.1.2 Application Acceptance 2.1.1.3 Contract Preparation 2.1.1.4 Contract Execution |
| **Flow Diagram** | |
|  | |

|  |  |
| --- | --- |
| **Requirement ID** | **2.2.3** |
| **Requirement Title** | Disable Card |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | The Unipet administrator will be able to disable the card based on the customer request or if any fraudulent found. Step would be following:   * Step 01: Disable card * Step 02: The card available balance/fund will automatically be added back to the **customer contact account.** |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 2.2.2 Card Generation Request |

|  |  |
| --- | --- |
| **Requirement ID** | **2.2.4** |
| **Requirement Title** | Enable Card |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | The Unipet administrator will be able to re-enabling/activate a disable card  **Steps:**   * Step 01: Enable/Activate card and assign to customer contract. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 2.2.2 Card Generation Request 2.2.3 Disable Card |

|  |  |
| --- | --- |
| **Requirement ID** | **2.2.5** |
| **Requirement Title** | Reassign Card |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | **The Unipet administrator will be able to reassignment card to different contacts**  Steps:   * Step 01: First, transfer card balance account to customer contract account. * Step 02: Remove from customer contract * Step 03: Assign to new account |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 2.2.2 Card Generation Request |

|  |  |
| --- | --- |
| **Requirement ID** | **2.4** |
| **Requirement Title** | **User Administration** |

|  |  |
| --- | --- |
| **Requirement ID** | **2.4.1** |
| **Requirement Title** | Create User |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Unipet administrator able to create other user through this system.  Steps:   * Step 01: Unipet administrator create user with require information * Step 02: The role assign option also will be available. If the Unipet administrator wants, the role will be able to assign when creating the user. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **2.4.2** |
| **Requirement Title** | Update User |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Unipet administrator able to update user from existing user list.  Steps:   * Step 01: Unipet administrator select user from existing user list. * Step 02: The role assign option also will be available. If the Unipet administrator wants, the role will be able to assign/reassign a role when updating the user. * Step 03: Update user associate information and save data with new information. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 2.4.1 Create User |

|  |  |
| --- | --- |
| **Requirement ID** | **2.4.3** |
| **Requirement Title** | Assign User Role |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Unipet administrator able to assign role for user. In this case, the administrator will select a role from a predefined list, as defined in the White Label software currently.  Steps:   * Step 01: Unipet administrator select user from existing user list. * Step 02: **Assign role from user role** list. OR **Update with new role** Or Removal role of the user. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 2.4.1 Create User |

|  |  |
| --- | --- |
| **Requirement ID** | **2.4.4** |
| **Requirement Title** | Delete User |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Unipet administrator able to remove/delete user from existing user list.  Steps:   * Step 01: Unipet administrator delete a user or multiple users from existing user list by clicking delete button. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 2.4.1 Create User |

|  |  |
| --- | --- |
| **Requirement ID** | **2.4.5** |
| **Requirement Title** | User List |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | The Unipet Administrator will see a list of all associated Users with companies. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 2.4.1 Create User |

|  |  |
| --- | --- |
| **Requirement ID** | **2.6** |
| **Requirement Title** | **Reporting** |

|  |  |
| --- | --- |
| **Requirement ID** | **2.6.1** |
| **Requirement Title** | Transaction Reports |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Generate a statement of the transaction of **Site, fleet, vehicle and Driver.** |
| **Filter Criteria:** | * Auth. Code * Shift Number * Vehicle * Driver * Site * Terminal / Controller * Fleet * Fuel * Contract * Date:   + Date From     - Time From   + Date To     - Time To |
| **Sort By:** |  |
| **Data Field/JSON Field** | |  |  |  |  | | --- | --- | --- | --- | | **Site** | **Fleet** | **Vehicle** | **Driver** | | Site | Fleet | Vehicle | Driver | | Tax Payer Id | Auth. Code | Auth. Code | Auth. Code | | Auth. Code | Site | Site | Site | | Trn. Number | Tax Payer Id | Tax Payer Id | Tax Payer Id | | Date | Trn. Number | Trn. Number | Trn. Number | | Time | Date | Date | Date | | SubAccount | Time | Time | Time | | Identification | SubAccount | Identification | Identification | | Fuel | Identification | Fuel | Fuel | | Volume | Fuel | Volume | Volume | | Amount | Volume | Amount | Amount | | Unit Price | Amount | Unit Price | Unit Price | |  | Unit Price |  |  | |
| **Group By** |  |
| **Database Source** | White Label Software |
| **API** |  |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **2.6.2** |
| **Requirement Title** | Financial Reports |
| **UI/Prototype/Mock Screen** | | |
|  | | |
| **Requirement Type** | Output |
| **Description/Business Logic** | | |
|  | | |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **2.6.3** |
| **Requirement Title** | Schedule Reports |
| **UI/Prototype/Mock Screen** | | |
|  | | |
| **Requirement Type** | Output |
| **Description/Business Logic** | | |
|  | | |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **2.7** |
| **Requirement Title** | **Wallet** |

|  |  |
| --- | --- |
| **Requirement ID** | **2.7.1** |
| **Requirement Title** | Digital Wallet |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Unipet Administrator able to deposit/withdraw money customer contract account.  **Deposit Money:**   * Administrator check customer available balance in Financial solution through API. * If balance is available, the amount will be transferred to the customer contract account. After transfer balance also would be updated in financial solution through API as transactions are carried out.   **Withdraw Money:**   * Based on request, Administrator able to withdraw/send back money from customer contract account. * Withdraw balance also would be updated in financial solution through API. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 2.4.1 Create User |

|  |  |
| --- | --- |
| **Requirement ID** | **3.0** |
| **Requirement Title** | **Fleet Interface** |
| **Requirement ID** | **3.1** |
| **Requirement Title** | **Vehicle Management** |

|  |  |
| --- | --- |
| **Requirement ID** | **3.1.1** |
| **Requirement Title** | Add Vehicle |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Input |
| **Description/Business Logic** | Fleet administrator able to create vehicle through this system.  Steps:   * Step 01: Fleet administrator create vehicle with require information. * Step 02: Fleet administrator assign the vehicle to a driver. Also, able to add multiple drivers with that vehicle. * Step 03: Vehicle Code and Vehicle Registration Number at the fuel dispenser, this field must be front loaded with leading zeroes.   For example, Vehicle Registration Number PDN 4942 is captured as 00004942. Similarly, Vehicle Registration Number PCN 422, is captured as 00000422.   * Step 04: Must select a fleet during vehicle creation. By default, there should a fleet for all vehicle. E.g.- Basic. |
| **API** |  |
| **Data Definition** | |
| |  |  |  | | --- | --- | --- | | **Field** | **Mandatory** | ***Comments/Note*** | | Vehicle Pin | Yes | The number used at the prompt for fuel purchase usually 8 numbers with the first 4/5 being 0's at minimum and the remaining being the vehicle Plate numbers | | Vehicle Registration Number (Plate) | Yes | Eg: PDS9540, PA2031, PDT101. Numbers from these used as Vehicle Pin | | Make of Vehicle | Yes | This should be a drop-down list with all the different vehicle brands | | Model of Vehicle | Yes | This should be all the different vehicle models available. List a 'other option' also | | Color | Yes |  | | Vehicle Class |  |  | | Manufacture Date |  |  | | Engine Number |  |  | | Chassis Number |  |  | | Fleet | Yes | Should be linked to Fleet type - Basic, Management etc. Drop down list to select. | | Drivers |  | This is when drivers are assigned to the vehicle | | Vehicle Rules |  | This is when rules are assigned to the vehicle | | Fleet Rules |  | Contract level rule assignment | | Fuel Type |  | This should be a drop-down list with Super, Premium and Diesel. If Diesel is selected, a rule should be applied to the vehicle where no super/premium can be used on the vehicle and vice versa. | | Fuel Tank Size |  |  | | Insurance Certifying Agency |  | Release 3 | | Last Certification Date |  | Release 3 | | Next Certification Date |  | Release 3 - Automatic Reminders - Expired once date is past. | | Certified Copy (Attachment) |  | Release 3 | | Demerit Points |  | Release 3 - Demerit Points build up with warning of approaching penalties which should also be listed on the page | | |
| **Database Storage** | White Label Software |
| **Dependency** | 3.3.1 Create Fleet |

|  |  |
| --- | --- |
| **Requirement ID** | **3.1.2** |
| **Requirement Title** | Update Vehicle |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet administrator able to update associated vehicle information.  Steps:   * Step 01: Fleet administrator select vehicle from existing list. * Step 02: Update vehicle information and save data with new information. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 3.1.1 Add Vehicle |

|  |  |
| --- | --- |
| **Requirement ID** | **3.1.3** |
| **Requirement Title** | Enable/Disable Vehicle |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet administrator able to enable or disable an associated vehicle from list:  Steps:   * Step 01: Fleet administrator enable/disable vehicle from existing list by clicking enable/disable button.   Note: Driver is not be able to assign with disable vehicle. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | * + 1. Add Vehicle |

|  |  |
| --- | --- |
| **Requirement ID** | **3.1.4** |
| **Requirement Title** | Remove Vehicle |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet administrator able to remove/delete vehicle from existing list.  Steps:   * Step 01: Fleet administrator select the vehicle that want to remove. * Step 02: First disassociated that vehicle from driver. * Step 03: Delete/remove vehicle from list.   Step 04: Adjust Subscription fee. (**The Subscription Fee, which is dependent upon the Fleet size, will be calculated based upon the number of Vehicles defined to the system for a particular Customer, at the time of calculating the fee. This will mean, as Fleet Customers keep their Fleet information current, adding and removing vehicles, the Subscription Fee is accurately calculated.**) |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 3.1.1 Add Vehicle |

|  |  |
| --- | --- |
| **Requirement ID** | **3.1.5** |
| **Requirement Title** | Assign a Vehicle to a Driver |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet administrator able to assign/tag a driver or multiple drivers with a vehicle.  Steps:   * Step 01: Fleet administrator select vehicle from existing list. * Step 02: Search driver from driver list and add (can add multiple driver one by one). * Step 03: add/tag with the vehicle.   Same way can add multiple drivers for a single vehicle. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 3.1.1 Add Vehicle  3.2.1 Add Driver |

|  |  |
| --- | --- |
| **Requirement ID** | **3.1.6** |
| **Requirement Title** | Vehicle List |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | The Fleet Administrator will see a list of all associated vehicles. For details view for a particular vehicle will need to select that vehicle. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 3.1.1 Add Vehicle |

|  |  |
| --- | --- |
| **Requirement ID** | **3.2** |
| **Requirement Title** | **Driver Management** |

|  |  |
| --- | --- |
| **Requirement ID** | **3.2.1** |
| **Requirement Title** | Add Driver |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet administrator able to add new driver through this system.  Steps:   * Step 01: Fleet administrator add a driver with require information. * Step 02: Fleet administrator **assign a vehicle** to a driver. * Step 02.1: Also, able to **add multiple vehicles** with the driver. * Step 02.2: Fleet administrator have also able to assign vehicle or add more vehicle from existing Driver list. |
| **API** |  |
| **Data Definition** | |
| |  |  |  | | --- | --- | --- | | **Field** | **Mandatory** | **Comments/Note** | | Driver Identifier | Yes |  | | Driver Last Name | Yes |  | | Driver First Name | Yes |  | | Driver Address (Address Line 1 | Address Line 2 | City/Town | Country) |  |  | | Email Address |  |  | | Telephone Number 1 |  |  | | Telephone Number 2 |  |  | | Date of Birth |  |  | | Driver's Permit # |  |  | | Vehicles |  | This is when vehicles are assigned to the drivers | | Rules |  | This is when rules are assigned to the drivers | | Driver's Permit Expiry Date |  | Notification should warn of expire and past due to Driver. Fleet Manager Interface shoul state whether drivers are eligbile to drive or not | | Driver Top Up Category (Fleet Identifier) |  | This is directly linked to Fleet type: Basic, Management Etc | | |
| **Database Storage** | White Label Software |
| **Dependency** | 3.1.1 Add Vehicle |

|  |  |
| --- | --- |
| **Requirement ID** | **3.2.2** |
| **Requirement Title** | Update Driver |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet administrator able to update associated driver information.  Steps:   * Step 01: Fleet administrator select driver from existing list. * Step 02: Update driver information and save data with new information. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 3.2.1 Add Driver |

|  |  |
| --- | --- |
| **Requirement ID** | **3.2.3** |
| **Requirement Title** | Enable/Disable Driver |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet administrator able to enable or disable an associated driver from list:  Steps:   * Step 01: Fleet administrator enable/disable driver from existing list by clicking enable/disable button.   Note: Disable driver is not be able to assign with a vehicle. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 3.2.1 Add Driver |

|  |  |
| --- | --- |
| **Requirement ID** | **3.2.4** |
| **Requirement Title** | Remove Driver |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet administrator able to remove/delete vehicle from existing list.  Steps:   * Step 01: Fleet administrator select the driver that want to remove. * Step 02: First disassociated that driver from tagged vehicle and card. * Step 03: Delete/remove driver from list. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 3.2.1 Add Driver |

|  |  |
| --- | --- |
| **Requirement ID** | **3.2.5** |
| **Requirement Title** | Assign a Driver to a Vehicle |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet administrator able to assign a vehicle or multiple vehicles for a driver.  Steps:   * Step 01: Fleet administrator select driver from existing list. Or, also able when add new driver. * Step 02: Search vehicle from list and select. * Step 03: assign for the driver   Same way can add multiple vehicles for a single driver. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 3.1.1 Add Vehicle  3.2.1 Add Driver |

|  |  |
| --- | --- |
| **Requirement ID** | **3.1.6** |
| **Requirement Title** | Vehicle List |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | The Fleet Administrator will see a list of all associated Driver. For details view for a particular Driver will need to select that driver. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 3.1.1 Add Vehicle |

|  |  |
| --- | --- |
| **Requirement ID** | **3.3.1** |
| **Requirement Title** | **Fleet Management** |

|  |  |
| --- | --- |
| **Requirement ID** | **3.3.1** |
| **Requirement Title** | Create Fleet |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet administrator able to create fleet for the company. This will allow for the more effective management of the Fleet, including the dispersal of funds in different ways to different fleets, or the assignment of rules to different fleets.  Steps:   * Step 01: Create fleet with Name and code. * Step 02: Would be an option to for administrator able to add/assign rules with that fleet.   Same way can add multiple vehicles for a single driver. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 3.6.1 Add Rules |

|  |  |
| --- | --- |
| **Requirement ID** | **3.3.2** |
| **Requirement Title** | Add Vehicle to a Fleet |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet administrator able to add vehicle to a fleet.  Steps:   * Step 01: During vehicle creation or modification Fleet administrator select/assign fleet for particular vehicle. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 3.1.1 Add Vehicle |

|  |  |
| --- | --- |
| **Requirement ID** | **3.3.3** |
| **Requirement Title** | Remove Vehicle from a Fleet |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet administrator able to remove/delete vehicle from existing fleet.  Steps:   * Step 01: Fleet administrator select from a vehicle list from an existing vehicle list. * Step 02: Remove vehicle by click remove button. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 3.1.1 Add Vehicle |

|  |  |
| --- | --- |
| **Requirement ID** | **3.3.4** |
| **Requirement Title** | Remove Fleet |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet administrator able to remove/delete vehicle from existing list.  Steps:   * Step 01: First Fleet administrator remove all tagged vehicle from fleet. * Step 02: Remove fleet |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 3.3.1 Create Fleet 3.3.2 Add Vehicle to a Fleet |

|  |  |
| --- | --- |
| **Requirement ID** | **3.4** |
| **Requirement Title** | **Card Management** |

|  |  |
| --- | --- |
| **Requirement ID** | **3.4.1** |
| **Requirement Title** | Assign a Card to a Driver |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet administrator will able to assign/reassign a card to a driver.  Steps for **Assign** card:   * Step 01: Assign a card to a driver. * Step 02: Driver will get a link through email/ notification and also can able to access from driver interface. * Step 03: For first time use, Driver request for reset PIN. **The PIN may be system generated or User selected.** * Step 04: Fleet Administrator get the request in fleet interface and set a PIN for the card. * Step 05: Driver again get a notification. * Step 06: Driver reset with new PIN. In this case, **Driver will be allowed to reset the PIN.**   Steps for **reassign** card:   * Step 01: Remove card from existing driver. * Step 02: Generate a new PIN. * Step 03: For first time use, Driver request for reset PIN. **The PIN may be system generated or User selected.** * Step 04: Fleet Administrator sees the request in fleet interface. * Step 05: Driver gets a notification of automatic pin change with options. * Step 06: Driver reset with new PIN. In this case, **Driver will be allowed to reset the PIN.** |
| **API** |  |
| **Database Storage** | * White Label Software * Bespoke Software |
| **Dependency** | 2.1.1.1 Automated Customer Enrolment  2.1.1.2 Application Acceptance  2.1.1.3 Contract Preparation  2.1.1.4 Contract Execution  2.1.1.5 Enrolment Invoice  2.2.2 Card Generation Request  3.2.1 Add Driver  3.3.1 Create Fleet |

|  |  |
| --- | --- |
| **Requirement ID** | **3.4.2** |
| **Requirement Title** | Reset PIN |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Prior to first use, or in the event that a Driver forgets their PIN, the Driver or the Fleet Administrator may request the reset of the PIN.  Step:   * Step 01: Driver/Fleet administrator request for reset PIN. * Step 02: Request will go to ATIONet for change PIN. * Step 03: In ATIONet PIN will change. A log will generate on ATIONet response. * Step 04: Driver/Fleet admin will get the notification with new PIN. * Step 04: For first time login Driver will need to reset the PIN. In this case, Driver will be allowed to reset the PIN. The PIN may be system generated or User selected. * Step 05: Fleet Admin will get notification of Card change PIN. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 2.1.1.1 Automated Customer Enrolment  2.1.1.2 Application Acceptance  2.1.1.3 Contract Preparation  2.1.1.4 Contract Execution  2.1.1.5 Enrolment Invoice  2.2.2 Card Generation Request  3.2.1 Add Driver  3.3.1 Create Fleet  3.4.1 Assign a Card to a Driver |

|  |  |
| --- | --- |
| **Requirement ID** | **3.4.3** |
| **Requirement Title** | Deactivate Cards |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet administrator able to deactivate card based on any incidence. Such as: card stolen or lost.  Steps:   * Step 01: Fleet administrator select card from card list. * Step 02: Fleet administrator deactivate card. * Step 03: Fleet administrator will get alert/notification in Admin interface. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 2.2.2 Card Generation Request  3.4.1 Assign a Card to a Driver |

|  |  |
| --- | --- |
| **Requirement ID** | **3.5** |
| **Requirement Title** | **Fleet Administration** |

|  |  |
| --- | --- |
| **Requirement ID** | **3.5.1** |
| **Requirement Title** | ACH Payment Advice and Approval |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet Administrator acknowledge ACH payment and request for approval.  Steps:   * Step 01: Fleet administrator update payment information with Reload Amounts and other Administrative Fees, providing the Payment Receipt details. * Step 02: UNIPET (Finance) will get a notification and also get in finance interface in this solution. * Step 03: Finance team check with bank payment transection. * Step 04: If payment would have completed, finance team confirm the request and update status “Approved”. Finance can select multiple companies. Also update   + Customer contract account balance will be increased to reflect the amount deposited and update in ATIONet company contract balance based on a minimum threshold Total amount deposited will not go on Ationet system. Total amount will be shown on interface with available amount to assign to fleet.   + Conversion of payment to U-liters.   + Update Fleet company U-liters balance.   + Fleet company get a notification with payment approval with update information. * However, if the payment was not made, the finance team declined the request. Sending a notification to the fleet administrator. Able to see in fleet interface as well. |
| **API** |  |
| **Database Storage** | Bespoke Software |
| **Dependency** | 2.1.1.1 Automated Customer Enrolment  2.1.1.2 Application Acceptance  2.1.1.3 Contract Preparation  2.1.1.4 Contract Execution  2.1.1.5 Enrolment Invoice  2.2.2 Card Generation Request  3.3.1 Create Fleet  3.3.2 Add Vehicle to a Fleet |
| **Flow Diagram** | |
|  | |

|  |  |
| --- | --- |
| **Requirement ID** | **3.5.2** |
| **Requirement Title** | Contract and Card Balance Check |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet administrator Able to check contract balance as well card balance.  Steps:   * Step 01: Fleet administrator Select Customer contract or card. * Step 02: available Balance will display for card or customer contract. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 2.1.1.1 Automated Customer Enrolment  2.1.1.2 Application Acceptance  2.1.1.3 Contract Preparation  2.1.1.4 Contract Execution  2.1.1.5 Enrolment Invoice  2.2.2 Card Generation Request  3.2.1 Add Driver  3.3.1 Create Fleet  3.4.1 Assign a Card to a Driver  3.5.1 ACH Payment Advice and Approval |

|  |  |
| --- | --- |
| **Requirement ID** | **3.5.3** |
| **Requirement Title** | Disperse funds |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet administrator will able to dispersal of funds to Drivers. Funds shall be dispersed in one of a number of ways   * Equate to – Enter an amount that will be equally shared among the selected Cards. (Ad hoc). * By Amount – Select cards to disperse to, and the individual amounts to each card. (Ad hoc) * Balance to – Select Cards to disperse to. Indicate an amount that all cards should balance to. Take funds from the Contract, and balance all selected cards to the amount set in Balance to. If a card has a higher balance, this is not adjusted. If the Contract does not have sufficient funds to bring the selected cards to the indicated Balance, the system should advise of the insufficient funds before dispersing any funds. (Scheduled) |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 3.1.1 Add Vehicle |

|  |  |
| --- | --- |
| **Requirement ID** | **3.5.4** |
| **Requirement Title** | Current Account Management |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet admin will able to movement of funds from   * the Fleet Contract to the subaccounts (Card), or * from subaccount to subaccount.   In this case**, a description of the transaction must be entered.** This transaction description may be selected from a standard set of transaction descriptions provided in a Drop-Down list, or it may be a free form transaction description. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 3.1.1 Add Vehicle |

|  |  |
| --- | --- |
| **Requirement ID** | **3.5.5** |
| **Requirement Title** | Digital Wallet |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet administrator will be able assignment of Cash/Ulitres to digital wallets. The Fleet Administrators must be able to transfer funds to drivers based on their need. This will be in the form of Ulitres- (Slitres, Plitres, Dlitres). |
| **API** |  |
| **Database Storage** | Bespoke Software |
| **Dependency** | 3.1.1 Add Vehicle |

|  |  |
| --- | --- |
| **Requirement ID** | **3.6** |
| **Requirement Title** | **Rule Management** |

|  |  |
| --- | --- |
| **Requirement ID** | **3.6.1** |
| **Requirement Title** | Add Rules |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet Administrator a will able to create Rules for Vehicle, Driver, Fleet etc. |
| **Rules Definition** | |
| |  |  |  | | --- | --- | --- | | **Function** | **Rule** | **Description** | | Network | Prompting | Used to turn on rule at fleet contract level to ask all driver fulling at the pump to input the driver pin, vehicle pin or current odometer level for example. It turns on the secondary track of a card to be read also.  Retries can be limited here. | | Fleet Company | Prompting | Used to turn on rule at fleet subaccount level to ask individual/group drivers (assigned to fleet) fuelling at the pump to input the driver pin, vehicle pin or current odometer level for example. It turns on the secondary track of a card to be read also. Retries can be limited here | | Fleet Company | Fuel | Limit driver group of drivers (assigned to fleet) type of fuel used at the sub account level. | | Regular Diesel | | Unleaded Super | | Unleaded Plus = Premium (on new interface) | | Fleet Company | Date Range | Limits driver/group of drivers (assigned to fleet) dates and time fuelling can take place at the sub account level | | Fleet Company | Location | Limits location a driver/group of drivers can fuel at the sub account level | | Fleet Company | DaysTime | Limit driver/group of drivers (assigned to fleet) dates and times fuelling can take place by individual dates at the sub account level | | Fleet Company | Limit Per Transaction | Limit driver/group of drivers (assigned to fleet) amount they can fuel at a given time. Eg Volume (10 Litres) or Money - $ 50.00 | | |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **3.6.2** |
| **Requirement Title** | Update Rules |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet administrator able to update existing rules driver information.  Steps:   * Step 01: Fleet administrator select a rule from existing list. * Step 02: Update associated rules information and save data with new information.   Rules would be applied to Drivers, Vehicles for Fleets. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 3.1.1 Add Vehicle |

|  |  |
| --- | --- |
| **Requirement ID** | **3.6.3** |
| **Requirement Title** | Rule Assignment |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet administrator able to be applied new/existing rule to Drivers, Vehicles or Fleets.  Steps:   * Step 01: Fleet administrator select a rule from existing list. * Step 02: Search Driver, Vehicle or fleet by code or name. The fleet administrator may be assigned rule only for driver/vehicle/fleet or for all. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 3.6.1 Add Rules |

|  |  |
| --- | --- |
| **Requirement ID** | **3.6.4** |
| **Requirement Title** | Remove Rules |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet administrator able to remove/delete existing rule from list.  Steps:   * Step 01: Fleet administrator select the one or multiple rules from list. that want to remove. Removal rules which are no longer valid. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 3.1.1 Add Vehicle |

|  |  |
| --- | --- |
| **Requirement ID** | **3.7** |
| **Requirement Title** | **Reporting** |

|  |  |
| --- | --- |
| Requirement ID | **3.7.1** |
| Requirement Title | Transaction Views |
| UI/Prototype/Mock Screen | | |
|  | | |
| Requirement Type | Output |
| Description/Business Logic | | |
| **Details will be added after receiving the information.** | | |
| API |  |
| Database Storage | White Label Software |
| Dependency | 3.1.1 Add Vehicle |

|  |  |
| --- | --- |
| **Requirement ID** | **3.7.2** |
| **Requirement Title** | Transaction Reports |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet admin able to generate a statement of the transaction of **fleet, vehicle and Driver.** |
| **Filter Criteria:** | * Auth. Code * Shift Number * Vehicle * Driver * Site * Terminal / Controller * Fleet * Fuel * Contract * Date   + Date From     - Time From   + Date To     - Time To |
| **Sort By:** |  |
| **Data Field/JSON Field** | |  |  |  | | --- | --- | --- | | **Fleet** | **Vehicle** | **Driver** | | Fleet | Vehicle | Driver | | Auth. Code | Auth. Code | Auth. Code | | Site | Site | Site | | Tax Payer Id | Tax Payer Id | Tax Payer Id | | Trn. Number | Trn. Number | Trn. Number | | Date | Date | Date | | Time | Time | Time | | SubAccount | Identification | Identification | | Identification | Fuel | Fuel | | Fuel | Volume | Volume | | Volume | Amount | Amount | | Amount | Unit Price | Unit Price | | Unit Price |  |  | |
| **Group By** |  |
| **Database Source** | White Label Software |
| **API** |  |
| **Dependency** |  |

|  |  |
| --- | --- |
| Requirement ID | **3.7.3** |
| Requirement Title | Schedule Reports |
| UI/Prototype/Mock Screen | | |
|  | | |
| Requirement Type | Output |
| Description/Business Logic | | |
| **Details will be added after receiving the information.** | | |
| API |  |
| Database Storage | White Label Software |
| Dependency | 3.1.1 Add Vehicle |

|  |  |
| --- | --- |
| **Requirement ID** | **3.8** |
| **Requirement Title** | **Request Management** |

|  |  |
| --- | --- |
| **Requirement ID** | **3.8.1** |
| **Requirement Title** | Create Request |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | This feature is actually a different kind of request facility that different types of users can do.  **Request by Fleet Administrator to Unipet Administrator**   * Steps:   + Step 01: Select request type. e,g.- card (add more in next release)   + Step 02: Add reason/comments/description for the request.   + Step 03: Submit request.   + Step 04: Unipet admin will notify through admin interface   **Request by Driver Administrator to Fleet Administrator**   * Steps:   + Step 01: Select request type. e,g.- Top up (add more in next release)   + Step 02: Add reason/comments/description for the request.   + Step 03: Submit request.   + Step 04: Fleet admin will notify through admin interface   + Step 05: System will generate a deduction amount which will reduce their balance once fleet company agrees agree. |
| **API** |  |
| **Database Storage** | Bespoke Software |
| **Dependency** |  |

|  |  |
| --- | --- |
| Requirement ID | **3.9** |
| Requirement Title | Fleet User Administration |

|  |  |
| --- | --- |
| **Requirement ID** | **3.9.1** |
| **Requirement Title** | Create User |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet administrator able to create user through this system.  Steps:   * Step 01: Fleet administrator create user with require information. * Step 02: The role assign option also will be available. If the fleet administrator wants, the role will be able to assign when creating the user. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **3.9.2** |
| **Requirement Title** | Update User |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet administrator able to update user from existing user list.  Steps:   * Step 01: Fleet administrator select user from existing user list. * Step 02: The role assign option also will be available. If the fleet administrator wants, the role will be able to assign/reassign a role when updating the user. * Step 03: Update user associate information and save data with new information. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 3.9.1 Create User |

|  |  |
| --- | --- |
| **Requirement ID** | **3.9.3** |
| **Requirement Title** | Assign User Role |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet administrator able to assign role for user. In this case, the administrator will select a role from a predefined list, as defined in the White Label software currently.  Steps:   * Step 01: Fleet administrator select user from existing user list. * Step 02: **Assign role from user role** list. OR **Update with new role** Or Removal role of the user. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 3.9.1 Create User |

|  |  |
| --- | --- |
| **Requirement ID** | **3.9.4** |
| **Requirement Title** | Delete User |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet administrator able to remove/delete user from existing user list.  Steps:   * Step 01: Fleet administrator delete a user or multiple users from existing user list by clicking delete button. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 3.9.1 Create User |

|  |  |
| --- | --- |
| **Requirement ID** | **3.9.5** |
| **Requirement Title** | User List |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | The Fleet Administrator will see a list of all associated Users. For details view for a particular vehicle will need to select that vehicle. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 2.4.1 Create User |

|  |  |
| --- | --- |
| **Requirement ID** | **4.1** |
| **Requirement Title** | **ERP Interface** |

|  |  |
| --- | --- |
| **Requirement ID** | **4.1.1** |
| **Requirement Title** | Company Data |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | System able to update Merchant or Fleet company information to ERP.  Steps:   * When Fleet administrator or Merchant admin create or update information it would be update in ERP system through API. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **4.1.2** |
| **Requirement Title** | Transaction Data |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | System will able to transfer information between the software and the ERP, to allow for the accurate invoicing and settlement of Fleet Companies and Merchants.  For Fleet Companies transactional information will update in Software as well ERP. Such as:   * information relating to Reload Amounts, will need to be sent from the software to the ERP. * The invoice will be created in the ERP, and will be a Receivable on the ERP, until the payment is received.   For Merchants information also able to update System to ERP and vice versa. Such as:   * The Card transaction details must be sent daily. * Based upon this information, the payment will be created on the ERP, and will be a Payable on the ERP until the Merchant is paid. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 3.1.1 Add Vehicle |

|  |  |
| --- | --- |
| **Requirement ID** | **4.1.3** |
| **Requirement Title** | Financial Updates – Merchant |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | System will be able to automatic update of the Merchant Current Account on the software when a Merchant Clearing transaction is completed. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **4.1.4** |
| **Requirement Title** | Financial Updates – Fleet Company |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | System will be able to automatic update of Fleet Company Current Account on the software when a Fleet Company's remittance of Reload Amount is confirmed received. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **4.2** |
| **Requirement Title** | **Finance Reports** |

|  |  |
| --- | --- |
| Requirement ID | **4.2.1** |
| Requirement Title | Transaction Report |
| UI/Prototype/Mock Screen | | |
|  | | |
| Requirement Type | Output |
| Description/Business Logic | | |
| **Details will be added after receiving the information.** | | |
| API |  |
| Database Storage | White Label Software |
| Dependency | 3.1.1 Add Vehicle |

|  |  |
| --- | --- |
| Requirement ID | **4.2.2** |
| Requirement Title | Financial Reports |
| UI/Prototype/Mock Screen | | |
|  | | |
| Requirement Type | Output |
| Description/Business Logic | | |
| **Details will be added after receiving the information.** | | |
| API |  |
| Database Storage | White Label Software |
| Dependency | 3.1.1 Add Vehicle |

|  |  |
| --- | --- |
| Requirement ID | **4.2.3** |
| Requirement Title | Schedule Reports |
| UI/Prototype/Mock Screen | | |
|  | | |
| Requirement Type | Output |
| Description/Business Logic | | |
| **Details will be added after receiving the information.** | | |
| API |  |
| Database Storage | White Label Software |
| Dependency | 3.1.1 Add Vehicle |

|  |  |
| --- | --- |
| **Requirement ID** | **4.3** |
| **Requirement Title** | **ACH Payment Advice and Approval** |

|  |  |
| --- | --- |
| **Requirement ID** | **4.3.1** |
| **Requirement Title** | ACH Payment Advice and Approval |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Fleet Administrator acknowledge ACH payment and request for approval.  Steps:   * Step 01: Fleet administrator update payment information with Reload Amounts and other Administrative Fees, providing the Payment Receipt details. * Step 02: UNIPET (Finance) will get a notification and also get in finance interface in this solution. * Step 03: Finance team check with bank payment transection. * Step 04: If payment would have completed, finance team confirm the request and update status “Approved”. Also update   + Customer contract account balance will be increased to reflect the amount deposited and update in ATIONet company contract balance.   + Conversion of payment to U-liters.   + Update Fleet company U-liters balance and cash balance   + Fleet company get a notification with payment approval with update information. * However, if the payment was not made, the finance team declined the request. Sending a notification to the fleet administrator. Able to see in fleet interface as well. |
| **API** |  |
| **Database Storage** | Bespoke Software |
| **Dependency** | 2.1.1.1 Automated Customer Enrolment  2.1.1.2 Application Acceptance  2.1.1.3 Contract Preparation  2.1.1.4 Contract Execution  2.1.1.5 Enrolment Invoice  2.2.2 Card Generation Request  3.3.1 Create Fleet  3.3.2 Add Vehicle to a Fleet |
| **Flow Diagram** | |
|  | |

|  |  |
| --- | --- |
| **Requirement ID** | **5.1.1** |
| **Requirement Title** | **Merchant User Administration** |

|  |  |
| --- | --- |
| **Requirement ID** | **5.1.1** |
| **Requirement Title** | Create User |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Merchant Manager will able to create **other merchant user**  Steps   * Steps01: Merchant Manager create user with require information. * Step 02: Could be assigned relevant roles with entities. Or can be assigned later. |
| **API** |  |
| **Data Definition** | |
| |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Mandatory** | **Default value (for inaccessible Field)** | **Comments/Note** | | Active | Yes | No |  | | Merchant Identifier | Yes |  |  | | Merchant Name – Service Station Name |  |  |  | | For primary contact for Merchant (First Name | Last Name) |  |  |  | | Merchant Address (Address Line 1 | Address Line 2 | Zip Code | City/Town | Country) | Yes |  | Optional 'Address Line 2' and Zip Code | | Merchant Email Address |  |  |  | | Merchant Telephone Number 1 |  |  |  | | Merchant Telephone Number 2 |  |  |  | | VAT Registration number | Yes |  |  | | Ownership Status (Dealer Owned, Unipet Owned) | Yes |  | Drop down list | | Business Type |  |  | Drop down list | | Banking Information ( Bank, Account Number) |  |  |  | | |
| **Database Storage** | White Label Software |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **5.1.2** |
| **Requirement Title** | Update User |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Merchant Manager able to update user from existing user list.  Steps:   * Step 01: Merchant Manager select merchant user from existing user list. * Step 02: Update user associate information and save data with new information. |
| **API** |  |
| **Data Definition** | |
|  | |
| **Database Storage** | White Label Software |
| **Dependency** | 5.1.1Create User |

|  |  |
| --- | --- |
| **Requirement ID** | **5.1.3** |
| **Requirement Title** | Assign User Role |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Merchant Manager able to assign role for user. In this case, the Merchant Manager will select a role from a predefined list, as defined in the White Label software currently.  Steps:   * Step 01: Merchant Manager select user from existing user list. * Step 02: **Assign role to and existing user role** list. OR addition, modification or removal of roles. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 5.1.1Create User |

|  |  |
| --- | --- |
| **Requirement ID** | **5.1.4** |
| **Requirement Title** | Delete User |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Merchant Manager able to remove/delete user from existing user list.  Steps:  Step 01: Merchant Manager delete user from existing user list by clicking delete button. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 5.1.1Create User |

|  |  |
| --- | --- |
| Requirement ID | **5.2** |
| Requirement Title | **Reporting** |

|  |  |
| --- | --- |
| Requirement ID | **5.2.1** |
| Requirement Title | Transaction Views |
| UI/Prototype/Mock Screen | | |
|  | | |
| Requirement Type | Output |
| Description/Business Logic | | |
| **Details will be added after receiving the information.** | | |
| API |  |
| Database Storage | White Label Software |
| Dependency |  |

|  |  |
| --- | --- |
| **Requirement ID** | **5.2.2** |
| **Requirement Title** | Transaction Reports |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Generate a statement of the transaction of **Site.** |
| **Filter Criteria:** | * Auth. Code * Shift Number * Site * Terminal / Controller * Fuel * Date:   + Date From     - Time From   + Date To     - Time To |
| **Sort By:** |  |
| **Data Field/JSON Field** | * Site * Tax Payer Id * Auth. Code * Trn. Number * Date * Time * SubAccount * Identification * Fuel * Volume * Amount * Unit Price |
| **Group By** |  |
| **Database Source** | White Label Software |
| **API** |  |
| **Dependency** |  |

|  |  |
| --- | --- |
| Requirement ID | **5.2.3** |
| Requirement Title | Financial Reports |
| UI/Prototype/Mock Screen | | |
|  | | |
| Requirement Type | Output |
| Description/Business Logic | | |
| **Details will be added after receiving the information.** | | |
| API |  |
| Database Storage | White Label Software |
| Dependency |  |

|  |  |
| --- | --- |
| **Requirement ID** | **5.2.4** |
| **Requirement Title** | Schedule Reports |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | |
| **Details will be added after receiving the information.** | |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** |  |

|  |  |
| --- | --- |
| Requirement ID | **6.1** |
| Requirement Title | **Fleet Reports** |

|  |  |
| --- | --- |
| **Requirement ID** | **6.1.1** |
| **Requirement Title** | Fleet Fueling Report |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Generate a report of all fueling transactions.  **Summary totals are to be provided.** |
| **Filter Criteria:** | Fleet, Specified period (e.g. week, month, quarter, etc.) |
| **Sort By:** | Chronological order, |
| **Data Field/JSON Field** | * Date; * Time; * Transaction ID (a unique identifier on the system for the transaction); * Card Number (Identification); * Driver ID – This can be the Driver Name (Subaccount); * Vehicle ID; * Merchant ID (Site); * Fuel Type; * Unit Cost; * Ulitres * Slitres * Dlitres * Plitres * Volume Dispensed (Litres); * Total; * VAT. |
| **Group By** | Driver, Vehicle, Date, Merchant (Site), Fuel Type, *etc*. |
| **Database Source** | White Label Software, Bespoke software |
| **API** |  |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **6.1.2** |
| **Requirement Title** | Activity Detail by Vehicle (Card Usage Report) |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | **For each Card in a Fleet, generate a report or statement of Card Usage over a specified period of time.**  Summary totals should:   * Total dispersal funds * Total fuel amount in Liters * Total spends.   Grand totals at the end of the report should provide a tally across all Cards reported on. |
| **Filter Criteria** | Select Fleet à Select Card/Cards, Specified period (e.g. week, month, quarter, etc.) |
| **Sort By** | Chronological order in the selected period |
| **Data/JSON Field** | * Driver ID – This can be the Driver Name; * Vehicle ID; * Opening Card Balance; * One line each, for each fueling transaction, comprising:   + Date;   + Time;   + Transaction ID (a unique identifier on the system for the transaction);   + Merchant ID (Site);   + Fuel Type;   + Odometer Reading;   + Unit Cost;   + Volume Dispensed (Liters);   + Total; * One line each, for each fund dispersal, comprising:   + Date;   + Time;   + Disperse Amt; * Closing Card Balance; * Number of Transactions |
| **API** |  |
| **Database Source** | White Label Software |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **6.1.3** |
| **Requirement Title** | Fleet Contract Balance Statement |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | **For a specified period, generate a statement of the movement in and out of the Contract.** |
| **Filter Criteria** | Select Fleet à Select Card/Cards, Specified period (e.g. week, month, quarter, etc.) |
| **Sort By** | Chronological order in the selected period |
| **Data/JSON Field** | * Driver ID – This can be the Driver Name; * Vehicle ID; * Opening Card Balance; * One line each, for each fueling transaction, comprising:   + Date;   + Time;   + Transaction ID (a unique identifier on the system for the transaction);   + Merchant ID (Site);   + Fuel Type;   + Odometer Reading;   + Unit Cost;   + Volume Dispensed (Litres);   + Total; * One line each, for each fund dispersal, comprising:   + Date;   + Time;   + Disperse Amt; * Closing Card Balance; * Number of Transactions |
| **API** |  |
| **Database Source** | White Label Software |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **6.1.4** |
| **Requirement Title** | Fleet Request Status Report |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | **For all requests made to the Administrator, which are open at the beginning of the reporting period, or which were created during the reporting period, generate a report of the requests and their status.** |
| **Filter Criteria** | Specified period (e.g. week, month, quarter, etc.) |
| **Sort By** | Chronological order in the selected period |
| **Data/JSON Field** | * Date Request Submitted; * Time Request Submitted; * Request ID; * Requestor Name; * Description of Request; * One line per response to request:   + Date;   + Time;   + Responder Name;   + Response (e.g. Acknowledgement, Request for more information, Quotation, Confirmation of Completion, etc.) * Status – Open, Pending, On Hold or Closed; * If Closed: * Date Request Closed; * Time Request Closed. |
| **API** |  |
| **Database Source** | White Label Software |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **6.2** |
| **Requirement Title** | **Merchant Reports** |

|  |  |
| --- | --- |
| **Requirement ID** | **6.2.1** |
| **Requirement Title** | Daily Transaction Report |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | **Generate a statement of the Merchant's current financial standing with the Company, including all payments to, and all Card Transactions performed at, the specified Merchant.** |
| **Filter Criteria** |  |
| **Sort By** | Chronological order in the selected period |
| **Data /JSON Field** | * Date Request Submitted; * Time Request Submitted; * Request ID; * Requestor Name; * Description of Request; * One line per response to request:   + Date;   + Time;   + Responder Name;   + Response (e.g. Acknowledgement, Request for more information, Quotation, Confirmation of Completion, etc.) * Status – Open, Pending, On Hold or Closed; * If Closed: * Date Request Closed; * Time Request Closed. |
| **API** |  |
| **Database Source** | White Label Software |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **6.2.2** |
| **Requirement Title** | Merchant Remittance Statement |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | **For a specified period, generate a statement of remittances to the Merchant's Contract.** |
| **Filter Criteria** | Specified period (e.g. week, month, quarter, etc.) |
| **Sort By** | Chronological order in the selected period |
| **Data Field/JSON Field** | * Opening Account Balance; * One line each, for each activity against the Account, comprising:   + Date;   + Activity Type:     - Day's Fuel Sales;     - Reimbursement;   + Transaction Amount; * Closing Account Balance. |
| **API** |  |
| **Database Source** | White Label Software |
| **Dependency** |  |

|  |  |
| --- | --- |
| Requirement ID | **6.3** |
| Requirement Title | Administrator Reports |

|  |  |
| --- | --- |
| **Requirement ID** | **6.3.1** |
| **Requirement Title** | Fleet Contract Balance Report |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | For all Fleet Companies, generate a report for a specified reporting period, providing following information:   * the movement of funds/points into and out of the Contract for the period * Closing Balance. * A Flag is to be set on if the Reload Threshold has been reached. This will enable the Administrator to reach out to Fleet Administrators who are approaching their Threshold. |
| **Filter Criteria** | Specified period (e.g. week, month, quarter, etc.) |
| **Sort By** | Chronological order in the selected period |
| **Data Field/JSON Field** | * Date; * Reporting Period; * One line for each Fleet Company, comprising:   + Fleet Company Name;   + Contract Maximum;   + Cumulative Opening Contract Balance (i.e. Contract Balance plus Card Balances);   + Total Reload Amount for Period;   + Total Fueling Amount for Period;   + Total Fees Deducted for Period;   + Over Limit details (if applicable) this will be the amount applied and used.   + Cumulative Closing Contract Balance;   + Reload Threshold Flag – If the Cumulative Closing Contract Balance is less than or equal to the Reload Threshold, this flag will be set to "Yes". |
| **API** |  |
| **Database Source** | White Label Software |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **6.3.2** |
| **Requirement Title** | Merchant Account Balance Report |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Administrator able to generate a for a specified reporting period, showing following information   * the amounts owed * the amounts paid to the Merchant Contract for the period * Closing Balance.   The Closing Balance will signal whether there is a need to reach out to a Merchant, to assure that payment is forthcoming, or that a reconciliation matter under investigation is receiving attention, etc. |
| **Filter Criteria** | Specified period (e.g. week, month, quarter, etc.) |
| **Sort By** | Chronological order in the selected period |
| **Data Field/JSON Field** | * Date; * Reporting Period; * One line for each Merchant, comprising:   + Merchant Name;   + Opening Account Balance;   + Total Remittance to Merchant for Period;   + Total Fuelling Amounts for Period     - Regular – Litres and Cost     - Super – Litres and Cost     - Premium – Litres and Cost   + Total Fees Deducted for Period;   + Cumulative Closing Account Balance. |
| **API** |  |
| **Database Source** | White Label Software |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **6.3.3** |
| **Requirement Title** | Request Status Report |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Administrator able to generate the request, which are open at the beginning of the reporting period, or which were created during the reporting period, generate a report of the requests and their status. |
| **Filter Criteria** | Specified period (e.g. week, month, quarter, etc.) |
| **Sort By** | Chronological order in the selected period |
| **Data Field/JSON Field** | * Date Request Submitted; * Time Request Submitted; * Request ID; * Requestor Name; * Description of Request; * One line per response to request:   + Date;   + Time;   + Responder Name;   + Response (e.g. Acknowledgement, Request for more information, Quotation, Confirmation of Completion, etc.) * Status – Open, Pending, On Hold or Closed; * If Closed:   + Date Request Closed;   + Time Request Closed. |
| **API** |  |
| **Database Source** | Bespoke Software |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **6.4** |
| **Requirement Title** | Card Production |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | **Generate a report on the Card production activity for a specified period.**   * **Summary totals are to be provided.** * **Categorized by Fleet Company.** |
| **Filter Criteria** | Specified period (e.g. week, month, quarter, etc.) |
| **Sort By** | Chronological order in the selected period |
| **Data Field/JSON Field** | * Fleet Company Name; * Request Date; * Request Type:   + New Enrolment batch;   + Replacement Cards (Lost or Stolen, Damaged);   + Additional Cards; * No. of Cards requested; * Date produced; * No. of Cards produced; * No. of Cards printed and encoded; * No. of Cards dispatched; * Date batch sent out; * Outstanding (if applicable); * Costing (optional). |
| **API** |  |
| **Database Source** | Bespoke Software |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **6.5** |
| **Requirement Title** | Lost or Stolen Cards |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | **Generate a report on the number of cards reported lost or stolen for a specified period.**   * **Summary totals are to be provided.** * **categorized by Fleet Company.** |
| **Filter Criteria** | Specified period (e.g. week, month, quarter, etc.) |
| **Sort By** | Chronological order in the selected period |
| **Data Field/JSON Field** | * Date reported * Fleet Company Name * Fleet Name * Account number * Driver's name * Status (lost or stolen) * Action taken (recovered, replaced, etc) * Date Action Taken * Authorized by |
| **API** |  |
| **Database Source** | Bespoke Software |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **7.0** |
| **Requirement Title** | **Security** |

|  |  |
| --- | --- |
| **Requirement ID** | **7.1** |
| **Requirement Title** | Restricted System Access |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Access control list (ACL) will be included to manage service and domain level permission. |
| **API** |  |
| **Database Storage** |  |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **7.2** |
| **Requirement Title** | Password security |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Use standard encryption & hashing mechanism. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 3.1.1 Add Vehicle |

|  |  |
| --- | --- |
| **Requirement ID** | **7.3** |
| **Requirement Title** | Restricted Access to different categories of information |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Access control list (ACL) will be included to manage service and domain level permission. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 3.1.1 Add Vehicle |

|  |  |
| --- | --- |
| **Requirement ID** | **7.4** |
| **Requirement Title** | User Administration |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Will be available in based on user role. |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** | 3.1.1 Add Vehicle |

|  |  |
| --- | --- |
| **Requirement ID** | **7.5** |
| **Requirement Title** | Access Control |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Will be available based on requirement. |
| **API** |  |
| **Database Storage** |  |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **7.6** |
| **Requirement Title** | Firewall & VPN |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Will provide as needed. |
| **API** |  |
| **Database Storage** |  |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **8.0** |
| **Requirement Title** | **Audit Controls & Monitoring** |

|  |  |
| --- | --- |
| **Requirement ID** | **8.1** |
| **Requirement Title** | Administrative/Configuration Transaction Monitoring |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Administrative/Configuration Transaction Monitoring log will available in the system |
| **API** |  |
| **Database Storage** | White Label Software |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **8.2** |
| **Requirement Title** | Security Maintenance Monitoring |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Will be available security log that can be monitor by date, time, user id, device and location, |
| **API** |  |
| **Database Storage** | Bespoke Software |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **8.3** |
| **Requirement Title** | Audit Log |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Will be available but restrict in the system. |
| **API** |  |
| **Database Storage** | Bespoke Software |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **9.1** |
| **Requirement Title** | **General Requirements** |

|  |  |
| --- | --- |
| **Requirement ID** | **9.1** |
| **Requirement Title** | Transaction Logging |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | System will track/transactional log will be store. |
| **API** |  |
| **Database Storage** | Bespoke Software |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **9.2** |
| **Requirement Title** | Operating Environment |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Hosted in Azure Cloud service used as server. |
| **API** |  |
| **Database Storage** |  |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **9.4** |
| **Requirement Title** | User Interface |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Will provide user friendly, lightweight and responsive UI. |
| **API** |  |
| **Database Storage** |  |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **9.5** |
| **Requirement Title** | Networking & Telecommunications |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | System will be lightweight and easily accessible through mobile data as well as public internet. |
| **API** |  |
| **Database Storage** |  |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **9.6** |
| **Requirement Title** | Documentation |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Will provide Following document:   * Software Specification document (SRS) * User Manual * System design Document |
| **API** |  |
| **Database Storage** |  |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **9.7** |
| **Requirement Title** | Training |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | * Provide training as per requirement. * Will provide video and user manual of this solution |
| **API** |  |
| **Database Storage** |  |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **9.8** |
| **Requirement Title** | Back Ups |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | We will store using MS cloud storage. |
| **API** |  |
| **Database Storage** |  |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **9.9** |
| **Requirement Title** | Retention |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | As per following requirement we will store using MS cloud storage   * Daily backups for thirty (90) calendar days, * Monthly backups for five (5) years |
| **API** |  |
| **Database Storage** |  |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **9.10** |
| **Requirement Title** | Scalability |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | As per our development approach this solution would be scalable. |
| **API** |  |
| **Database Storage** | N/A |
| **Dependency** |  |

|  |  |
| --- | --- |
| **Requirement ID** | **9.11** |
| **Requirement Title** | System Integration |
| **UI/Prototype/Mock Screen** | |
|  | |
| **Requirement Type** | Output |
| **Description/Business Logic** | Ability to export in csv, MS excel format.  For other format we will discuss finalize based on both team discussion. |
| **API** |  |
| **Database Storage** | Bespoke Software |
| **Dependency** |  |

# Non-Functional Requirements

*<We will provide.>*

# Performance Requirements

*<We will provide.>*

## Security Requirements

*<We will provide.>*

# Third-Party Integration or requirements

*<We will provide.>*

# External Interface Requirements

*<We will provide.>*

# User Interfaces

*<We will provide.>*

# Hardware Interfaces

*<We will provide.>*

# Software Interfaces

*<We will provide.>*

# Communication Interfaces

*<We will provide.>*

# Environment considerations

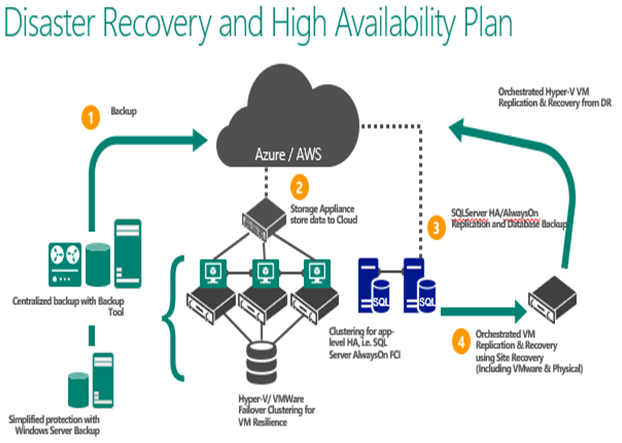
## Development and Production environment

|  |  |  |  |
| --- | --- | --- | --- |
| Mobile App  {(Android/iOS/ Windows)-React Native} | Web Portal  {Angular} {D3}{ Bootstrap}{CSS} | Service {Spring Boot}{Log4}{Elasticsearch}{Eureka}… | Database  {PostgreSQL}{Memcached} |
| Configuration |  | | |
| Dev. Env |  | | |
| Report |  | | |
| Build |  | | |
| Testing |  | | |
| OS |  | | |

## Backup and Recovery Plan

Following are required as backup and recovery:

* Synchronized replication of data wherever applicable and upon agreement
* Point-in-time-recovery (PITR)
* Nightly full back

**

# Assumptions and Dependencies

*N/A*

# Maintainability Requirements

*N/A*

# High Availability (HA) Requirements

*N/A*

# Appendix A: Glossary

*N/A*