# Document management

|  |  |
| --- | --- |
| Property | Notes |
| Title | Statement of work for “Q-Box Project” development engagement. |
| Service | Product development of “Q-Box Project” software. |
| Purpose | To provide implementation and support for “Q-Box Project” software. |
| Document owner | SWOMB Technologies and Services private limited. |
| Revision history | 1.0. |

Authors

The following authors contributed to the creation of this deliverable.

|  |  |
| --- | --- |
| PM | Delivery |
| Vengatesan Kalimuthu | Rajanandini Venkataraman |

Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Change description | Updated by | Date |
| 1.0 | Document creation | Vengatesan Kalimuthu | 12-Feb-2024 |
| 2.0 | Changes on   * Non-Functional Requirements * Pricing * Support Activities SLA | Vengatesan Kalimuthu | 26-Feb-2024 |

Contents

[Document management 1](#_Toc331061123)

[Section -1 Executive summary 4](#_Toc1996988765)

[Project overview: 4](#_Toc507712738)

[Objectives: 4](#_Toc1424506674)

[Section -2 Scope of work 4](#_Toc576395407)

[Overview: 4](#_Toc2129082611)

[In-scope: 4](#_Toc619503646)

[Out of scope: 4](#_Toc1074599718)

[Dependencies 4](#_Toc850989002)

[Section -3 Deliverables 4](#_Toc773037755)

[Section -4 Functional requirements 5](#_Toc2052218311)

[High Level Modules 5](#_Toc1408617223)

[Section -5 Non-functional requirements 5](#_Toc1960830778)

[Section -6 High level architectural considerations 6](#_Toc2092442835)

[Key Considerations 6](#_Toc1202261270)

[Components & technologies 6](#_Toc92721619)

[Project resources 6](#_Toc1980118142)

[Section -7 Solution Architecture 6](#_Toc263502996)

[Technical Implementation Landscape 6](#_Toc1217667760)

[Solution Design 6](#_Toc470152388)

[Hosting Infrastructure Architecture 6](#_Toc60736386)

[Technical Architecture 6](#_Toc2083382292)

[Major Use Case Flow 6](#_Toc1890711545)

[Data Backup process 6](#_Toc1099010726)

[Data Archiving: 6](#_Toc2088694965)

[Restoring from Archive: 6](#_Toc2058661793)

[API Contract 6](#_Toc662367656)

[Section -7 Delivery Strategy 6](#_Toc1908994985)

[Section -8 Initial Support Engagement 6](#_Toc45715907)

[Section -9 Delivery Commitments 6](#_Toc864193499)

[Section -10 Product development plan 6](#_Toc1884093953)

[Development Timeline: 6](#_Toc906241065)

# Section -1 Executive summary

## Project overview:

“OwlTech Pvt Ltd, a startup established in 2016 specialised in setting up IoT enabled robotic retail outlets under the name FRSHLY for fresh food aggregated from popular restaurants within the city. At its peak OwlTech operated as much as 20 such outlets across the country in cities like Chennai, Bangalore, Mumbai, Kolkata and Hyderabad and had over 100+ restaurant partners. Now with its understanding of the market and experience in launching IoT enabled platforms,

OwlTech is looking to solve a major problem that the online delivery companies face at the kitchen to rider hand over process.

## Objectives:

* Define the in-scope / out of scope of the work towards this project.
* Define deliverables of the project to the client.
* Define technologies and architecture.
* Define timelines of the deliverables.
* Define configuration management / source code management.
* Define communication methods.
* Define development methodologies to achieve the deliverables.

# Section -2 Scope of work

## Overview:

Based on the information provided by the customer “OwlTech Pvt Ltd in the engagement and as mutually agreed the below are the scope of the development and initial support.

## In-scope:

* Product Development
  + UI development.
  + Data model & data structure design.
  + API design & development.
  + Web app design & development.
  + Integration with the backend (3rd party) APIs.
* Version Maintenance
  + Maintaining the project artifacts in client preferred SCM tool.
  + Maintaining the project binaries and their versions in client preferred SCM tool.
* Application Deployment
  + Building the deployable binaries.
  + Deploying the application in the cloud.
* Initial Support
  + As per the agreement the initial support covers
  + Knowledge transition to client personal
  + Handling application incident tickets.
  + Stabilization of the application.
  + Training to administrator user of the application – limited.
* Documentation
  + As part of the deliverables, the below documents will be prepared and handed over to the customer
    - Technical design document
    - Infra arch document.
    - Use case document.
    - Test case document.
    - Mini user manual / training document.
* Cloud Infrastructure Architecture
  + Assessment and planning.
  + Design and architecture.
  + Security planning.
  + Cost optimization.
  + Integration.
  + Network design.
  + Automation and orchestration.
  + Compliance and governance.
  + Documentation.
* Cloud Infrastructure Support
  + Monitoring and alerting.
  + Incident response.
  + Patch and update management.
  + Performance optimization.
  + User support.
  + Backup and recovery.
  + Access control.
  + Cost monitoring and reporting.
  + Documentation and knowledge sharing.

## Out of scope:

* Application content and multimedia.
* End user customer support engagement system.
* Marketing & media contents.
* Multilingual content.
* Play store App Store Content if any

## Dependencies

* Development & testing infrastructure in cloud for deploying the application to be provided by Owl Tech Pvt Ltd.
* Third party integration licenses, access keys like AWS whatever needed should be provided by Owl Tech Pvt Ltd.

# Section -3 Deliverables

|  |  |
| --- | --- |
| **List of deliverables** | |
| Product source code & software licenses |  |
| Implementation & configuration |  |
| Handover & high-level documentation |  |
| Support |  |
| Training |  |
| Professional services |  |

# Section -4 Functional requirements

## High Level Modules

**User Profile Management**

* This module will serve as the master module to add, remove, update user profiles based on designation.
* Access to functions and modules will be managed by hierarchical integrity.
* This module will be the only one which allows the Company to perform all CRUD operations for profile management, which includes addition of customers, remote location, infrastructure addition across geography.

**Customer Management**

* This module is meant for customers to self-manage their operations from our portal, like view total inventory available, delivered and in Q-Box, it should also provide with features to manage reallocation, returns re-order etc.

**Remote location management**

* This module will be an installable application which will be uniquely identifiable ·
* Easy to install / uninstall/re-install ·
* Data synchronization as near to live as possible without costing high ·
* Track update and persist stock and SKU’s ·
* Ability to capture code identifier and make necessary process based on application flows. Q- Box, Delivery partner, Inward stock and outward stock ·
* Track other logistical information required at Remote location

**Infrastructure Management**

* A single viewpoint to locate and drill down all infrastructure related metrics. ·
* Facilities available,
* Number of boxes
* In operation/ out of operation/ Needing maintenance / replacement

**Database Management**

* This module should have the ability persist every capable data about Infrastructure, transactions with timestamp, Should Realtime and offline data charts along with pivots to showcase the data flowing in and thru the systems.
* There needs to be a system with which data can be used to predict /extrapolate and show trend charts over a period of time.

**Integration Management**

* This module will be an integral portion of the MVP, which will allow the application to integrate with Delivery company infrastructures. Every Integration with Q-Box application and Delivery Company should be managed from this module.
* All remote configuration to Q-Box Remote locations will be routed thru this module.
* Integration with other client systems maybe on any Tech stake, this module should have the capability to showcase real time traffic and connectivity to ensure the systems work seamlessly.
* System should have the ability to send and receive any kind of data from Partner. All such data should be traceable and persisted

**Occupancy Management**

* This Server side – View of all Q-boxes and ability to drill down to see utilization of infrastructure – Q- Box, Mass storage, duration of occupancy, item wise occupancy, date and time wise occupancy based on item etc.
* Remote Side – Display available Q-boxes to load, Occupied Q- boxes, Inventory status, delivered and time inside box status,
* Scan, persist and open to load.
* Scan, soft delete and open door to pick up.
* Inventory Inward tracking in the remote location.

**Metrics and Statistics Management.**

* This module is a one stop shop to retrieve all historical data based on filters and selection dropdowns. This module is accessible. And should be to generate MIS reports on a periodic basis and publish to a defined group of people, this facility can be configured to add individuals or a group of people

**Order Tracking Management**

* This module is for tracking Order generation from Delivery Partner to Hotel and to Q-Box Inventory,
* This module will host the SKU generation for every individual menu item that will be attached along with every Order and Menu item generated.
* This will also host the SKU management, that will include addition deletion and updating of SKU’s.
* This module will be pivotal for connecting the Restaurant Order to the Q-Box and then to the Customer order.
* Every SKU and individual pack will have to be associated to an order number {both the customer and delivery company}.
* Any unmapped SKU pack will have to be reconcilable with returns.

**Revenue Tracker**

* This module is an acquired intelligence module.
* The key function of this module is to retrieve the cost of a menu item from the delivery company application on a daily basis against every SKU, collate the order details and derive the sale price of every order passing through the system and show case the Transactional value of all orders.
* A report of total sale value transacted through every Q-box should be consolidated to prepare an EOD report at every level of the Pecking order.

# Section -5 Non-functional requirements

|  |  |
| --- | --- |
| Performance | Ensure the applications are responsive and perform efficiently, even with a large volume of data and users. |
| Scalability | Design the system to handle the potential growth of the company and accommodate additional units or functionalities. |
| Usability | Develop an intuitive and user-friendly interface to facilitate easy adoption by employees with varying levels of technical expertise. |
| Availability | The Application will be up and running all the time 24/7 and the availability will be 99.9% other than the unexpected situation with the Third-Party Service Providers like Cloud Vendors, Internet Service Providers. |
| Data backup and recovery | Implement regular data backup and disaster recovery mechanisms to prevent data loss. |
| Compliance | Ensure that the Q-BOX applications comply with industry-specific standards and regulations pertaining to iso certifications. |
| Integration | Enable integration with existing and future systems or APIs |
| Security | Implement robust security features to protect sensitive business data and prevent unauthorized access. |
| Reporting and analytics | Provide comprehensive reporting and analytics tools to help management make data-driven decisions. |
| Hierarchical access control | Implement a hierarchical access system to control user access based on roles and responsibilities. |
| Documentation | Maintain detailed technical documentation for the applications to support iso compliance and facilitate future maintenance and updates. |
| Training and support | Provide training and support to ensure that the application users can effectively use and maintain the application. |
| Cloud-based | Ensure that the applications are hosted in a secure and reliable cloud environment to facilitate accessibility and scalability. |
| Scalable architecture | Design a modular and scalable architecture that can accommodate future expansions and changes in business requirements. |
| User training and onboarding | Develop comprehensive training materials and provide onboarding support to ensure all users can effectively utilize the application. |

# Section -6 High level architectural considerations

## Key Considerations

* Component based distributed architecture.
* Reactive in nature.
* Containerized builds.
* Microservice / API based.
* Clear separation of functional modules and
* Plug & play integrations
* The solution is being loosely coupled model, which can easily be integrated with any system, which needs data engineering services.
* Role based customization
* Rule driven configurations
* Easily extendable.

## Components & technologies

**Web Ui** – Reactjs

**Mobile Application –** Flutter ( built for iOS & Android )

**Middleware** – Vert.X

**Authentication** – JWT / OAuth2

**Database** - Postgres

**Security** - DMG with web application firewalls

**Data encryption** - AES / RSA / SHA 256-bit encryption

**Hosting environment –** Cloud / On Premises

## Project resources

**Technology resources** -

* + most preferred Open-source software licenses / purchase licenses if required for the technology listings.

**Human resources** -

* + initial development 3 – 4 ft and 1 pt. later support based on mutual agreement

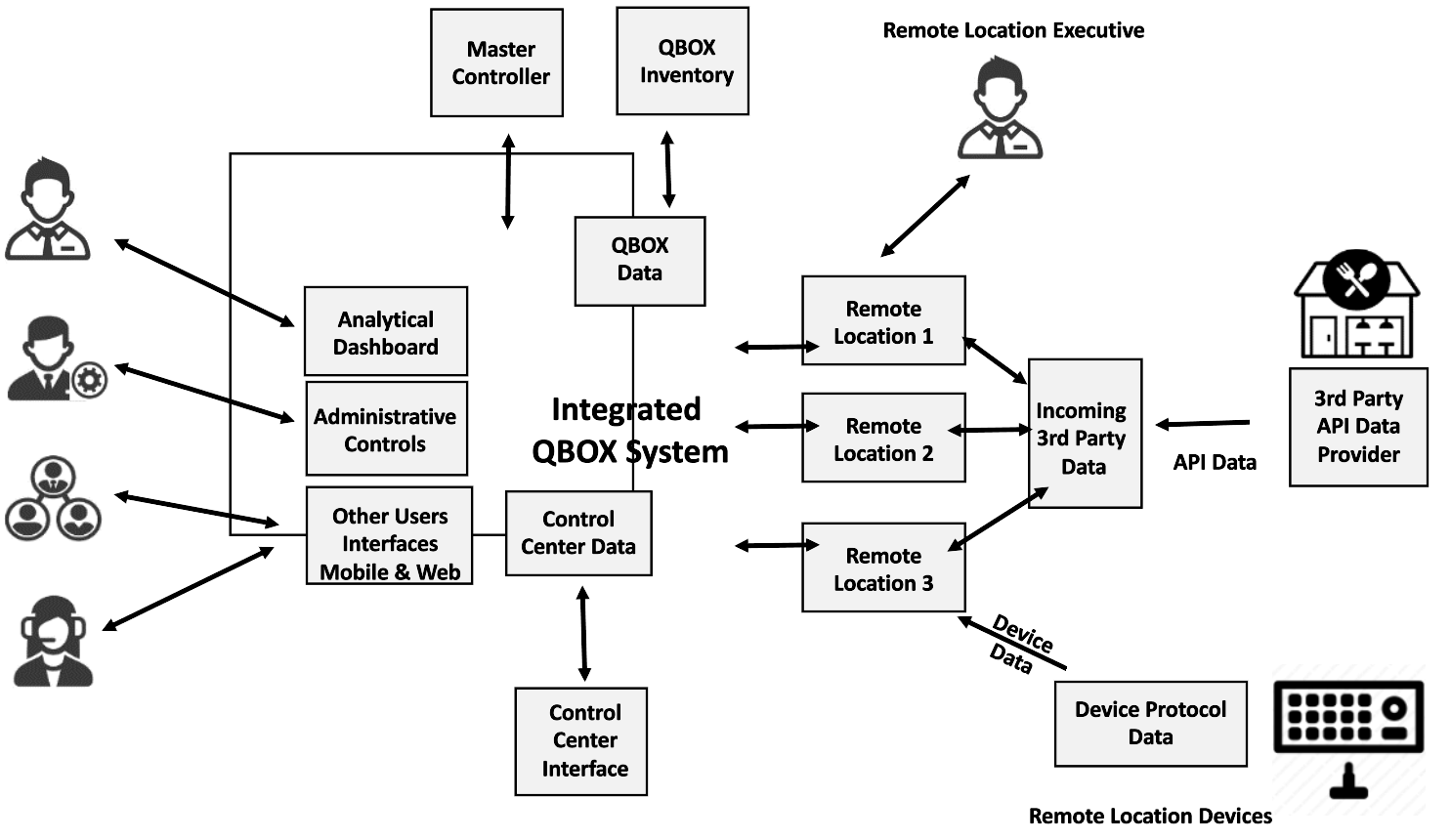
**Communication resources** -

* + over emails / teams / or other tele or video calls. in person meetings

**Scrum based development model**

# Section -7 Solution Architecture

## Technical Implementation Landscape

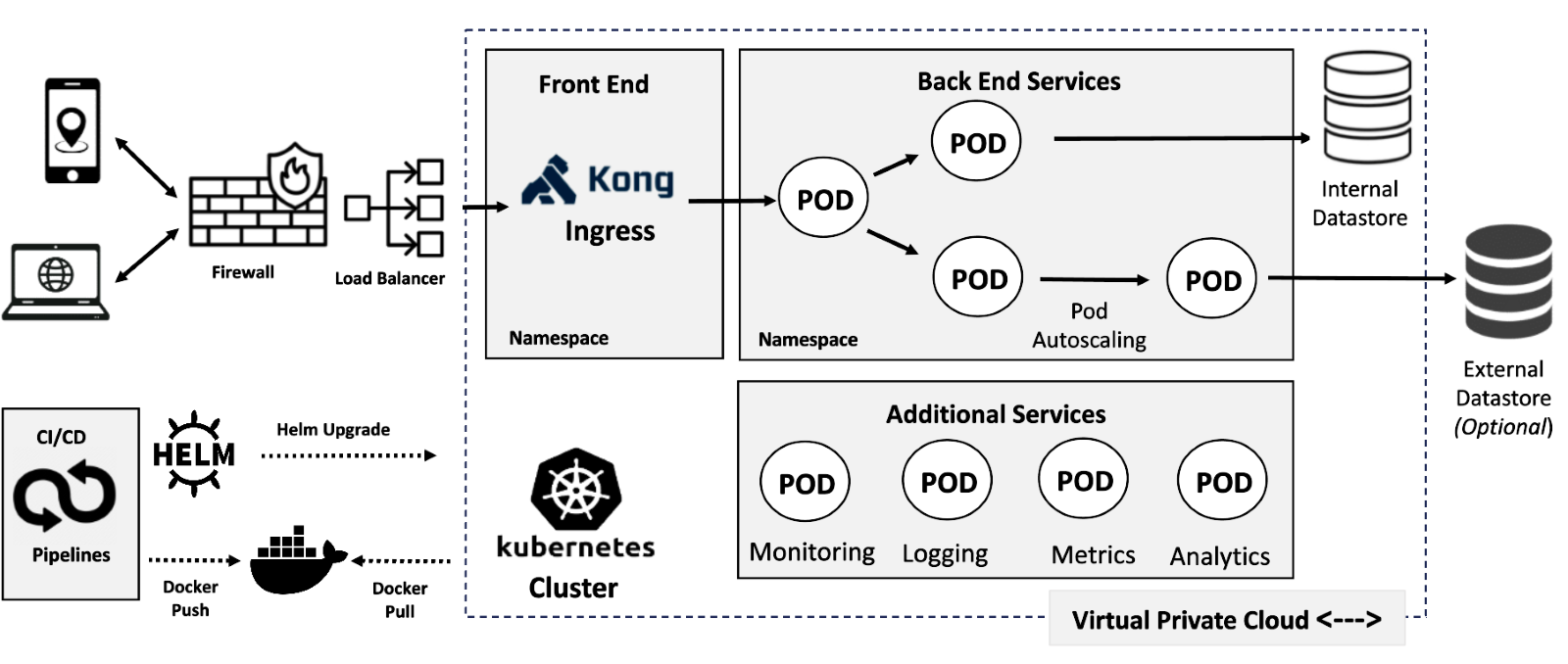


* Master Controller Dashboard
* Remote Location Dashboard
* Restaurant / Food Supplier / Delivery Company API Integration
* Device Data Integration
* Overall Process Orchestration

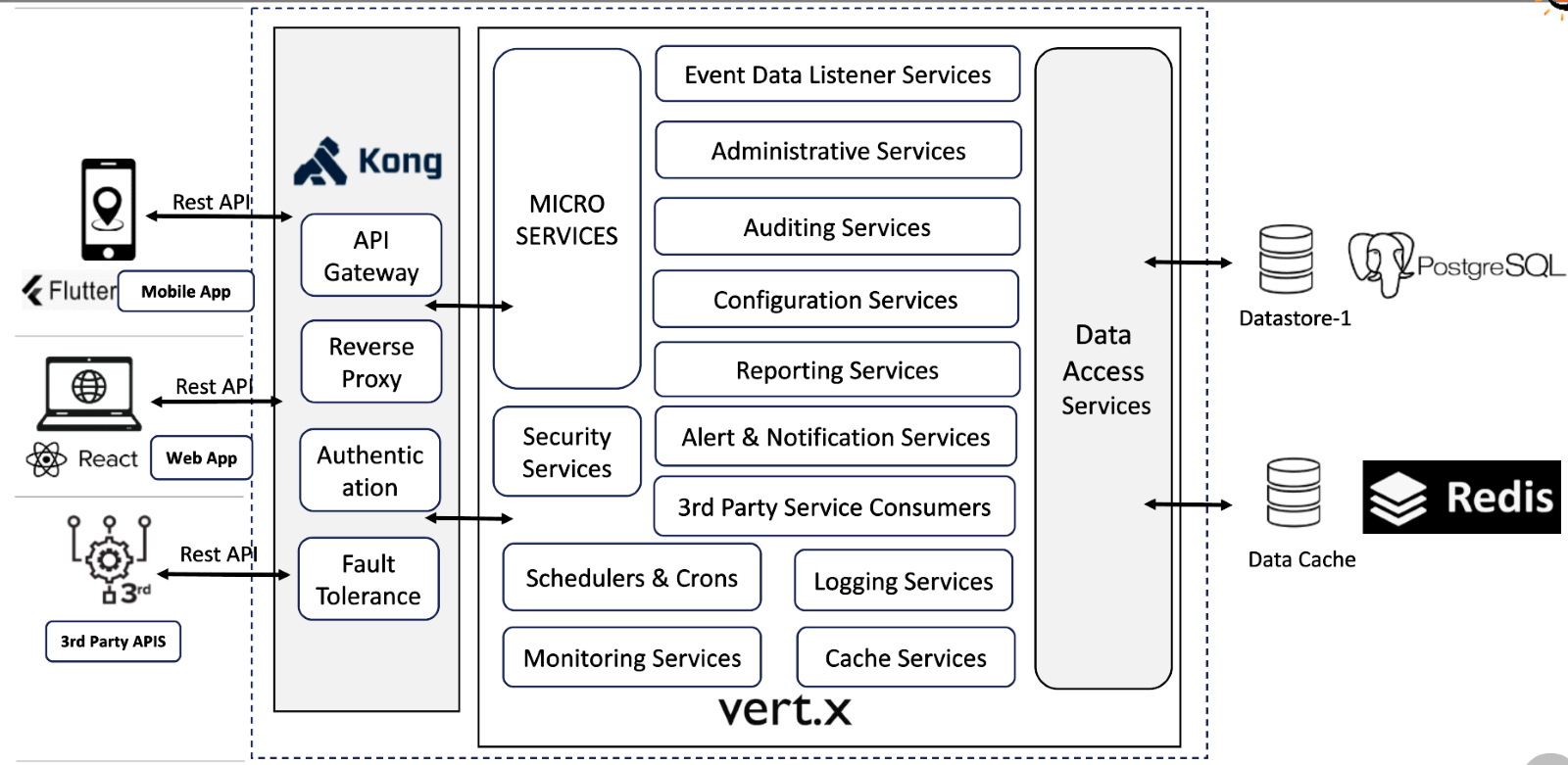
## Solution Design

## 

## Hosting Infrastructure Architecture

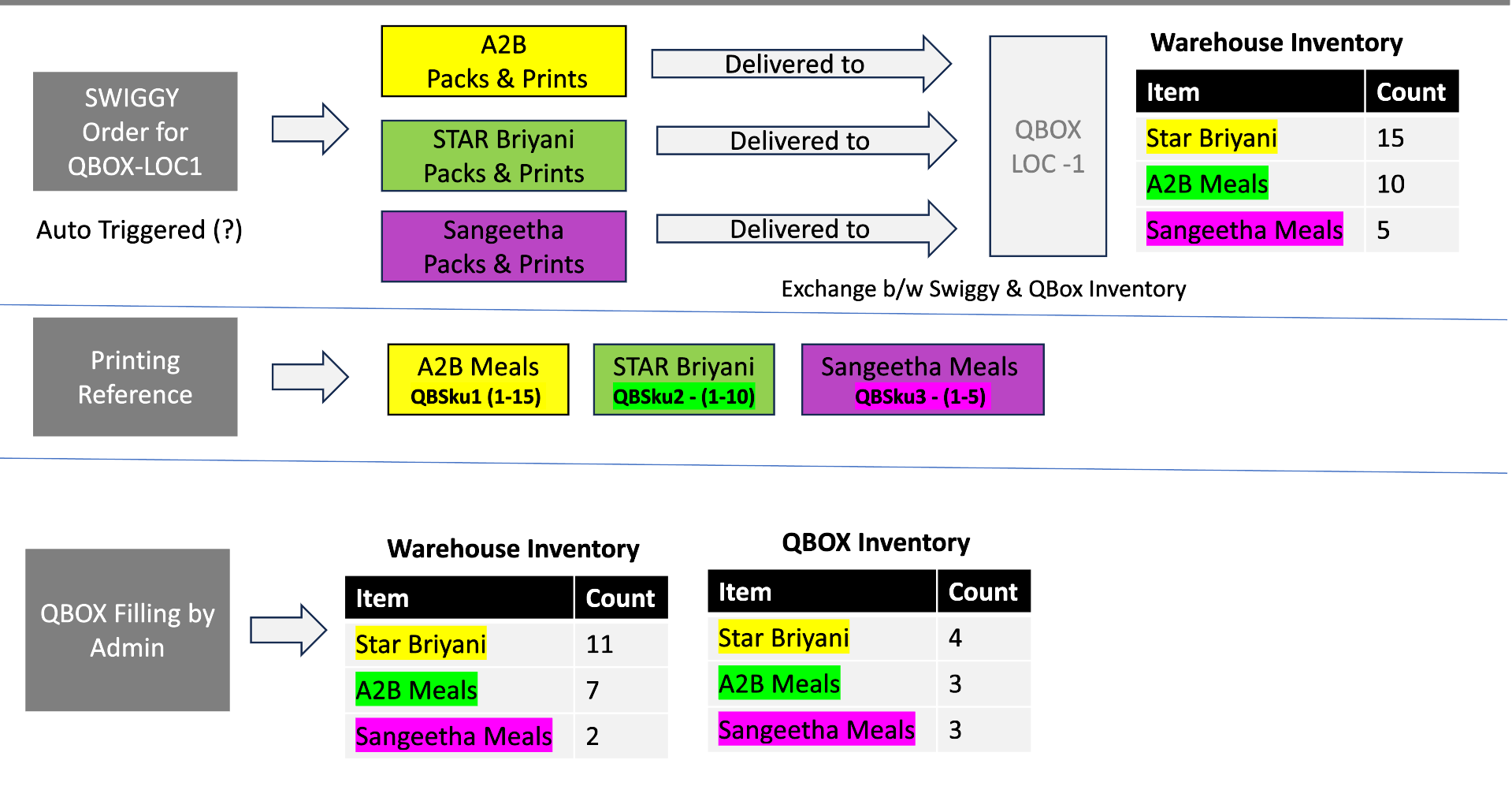


## Technical Architecture

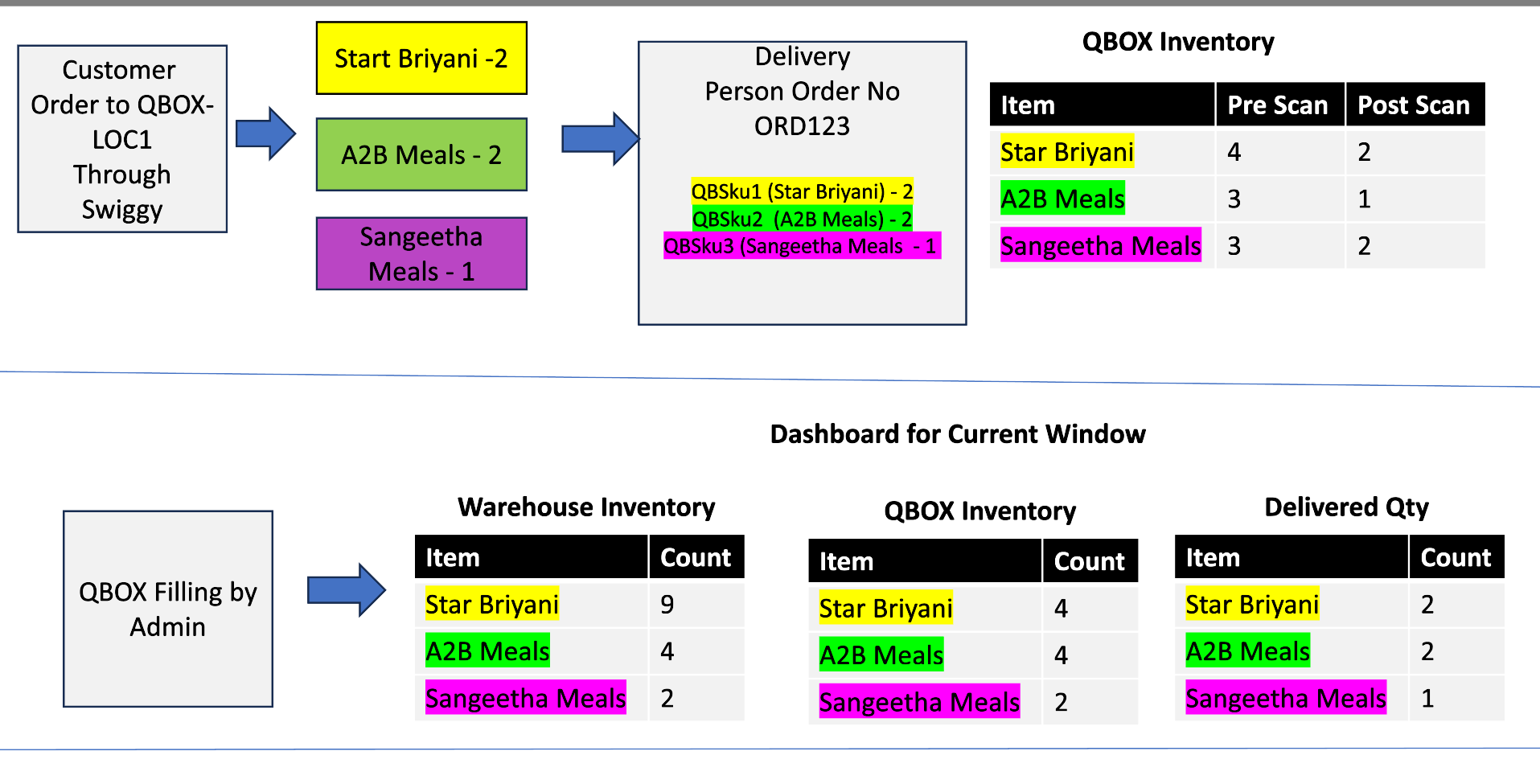


## Major Use Case Flow

**Inward & Internal QBOX-Flow**



**Outward Flow**



## Data Backup process

**Option –1**

* + Using Cloud PAAS (Postgres PAAS)
  + Continues Expense
  + Reliable,
  + No Manual intervention is needed when the Database server goes down.

**Option –2**

* + Writing Procedure to Automate the Data backup
  + One time development expense.
  + Manual Restoration needed when server Goes down

## Data Archiving:

**Purpose:**

* + Move inactive data to separate storage for long-term retention, reducing active database size and improving performance.

**Methods:**

* + Manual archiving: Export data using SQL COPY commands or other tools, store in compressed files or another database.
  + Automatic archiving: Set up automated scripts or use built-in features like pg\_dump to regularly archive data based on criteria such as age, size, or usage.

**Capabilities:**

* + Flexibility: Various options for archiving data, allowing administrators to choose the most suitable method.
  + Scalability: Handles large volumes of data efficiently, suitable for high-volume applications.
  + Security: Implement access controls and encryption mechanisms to secure archived data.

## Restoring from Archive:

**Purpose:**

* + Retrieve archived data and restore it to the active database when necessary for data recovery, analysis, or auditing.

**Capabilities:**

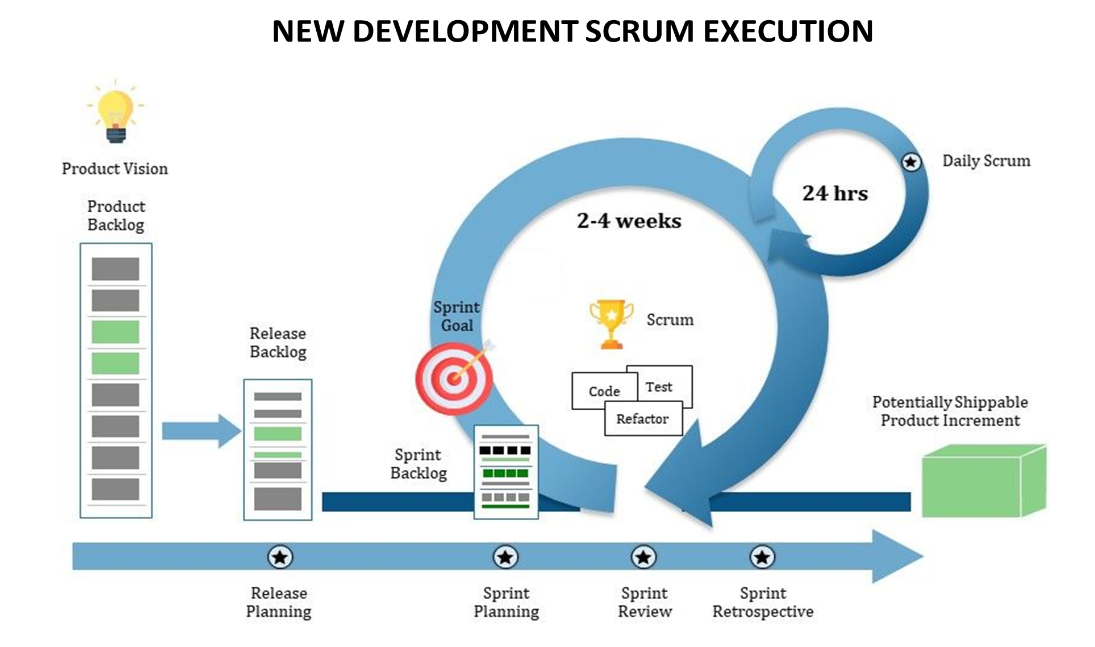
* + Point-in-time recovery: Supports restoring the database to a specific point using archived WAL files.
  + Granular restoration: Restore specific tables, schemas, or individual records from archived data.
  + Rollback: Supports rollback mechanisms to revert changes and maintain data consistency.

# 

## API Contract

* + To Maintain the system and updates better, we need to build and handshake the APIS to the external parties wherever applicable rather than the core consumption APIs of the third party.
  + Example: Sequence number generation API must manage by Q-BOX and consumed by Swiggy / Zomato.

# Section -7 Delivery Strategy



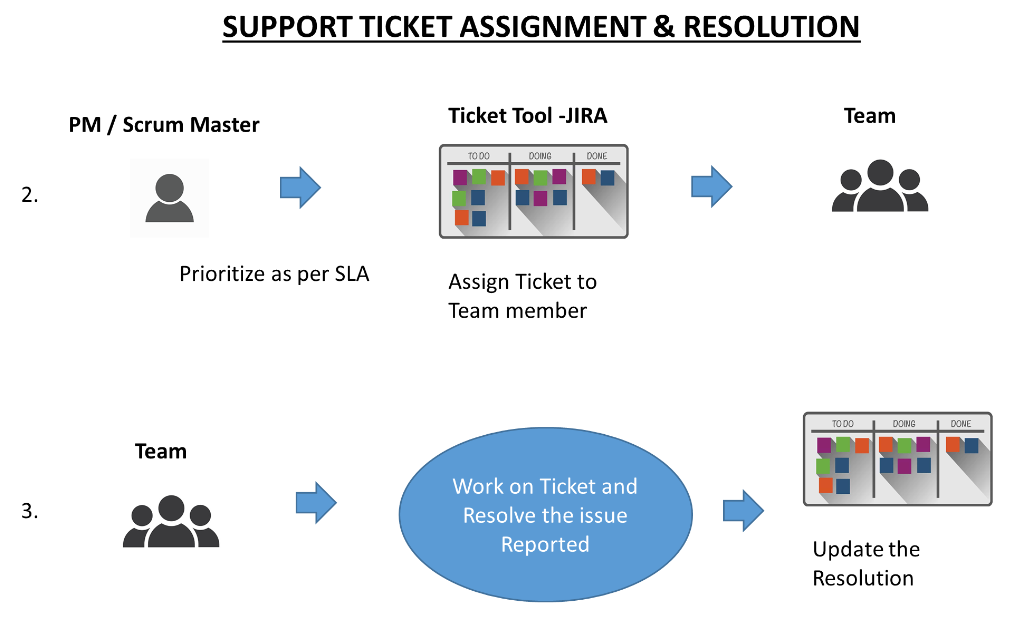
* **Product Owner / Scrum Master** collects all stockholder’s new requirements and enhancements and then they are converted into product backlog based with priority.
* Based on initial estimation and availability, Scrum Master will provide the number of sprints and release date.
* Each sprint will be planned based on capacity of team and timeline.
* End of the each sprint, team will demo the completion and review followed by retrospective.
* Based on the feedback, the team will improve the sprint velocity to deliver best productivity on each sprint.
* Team will also encourage to share the best practice and automation in technology based on domain knowledge and the experience over the project execution.

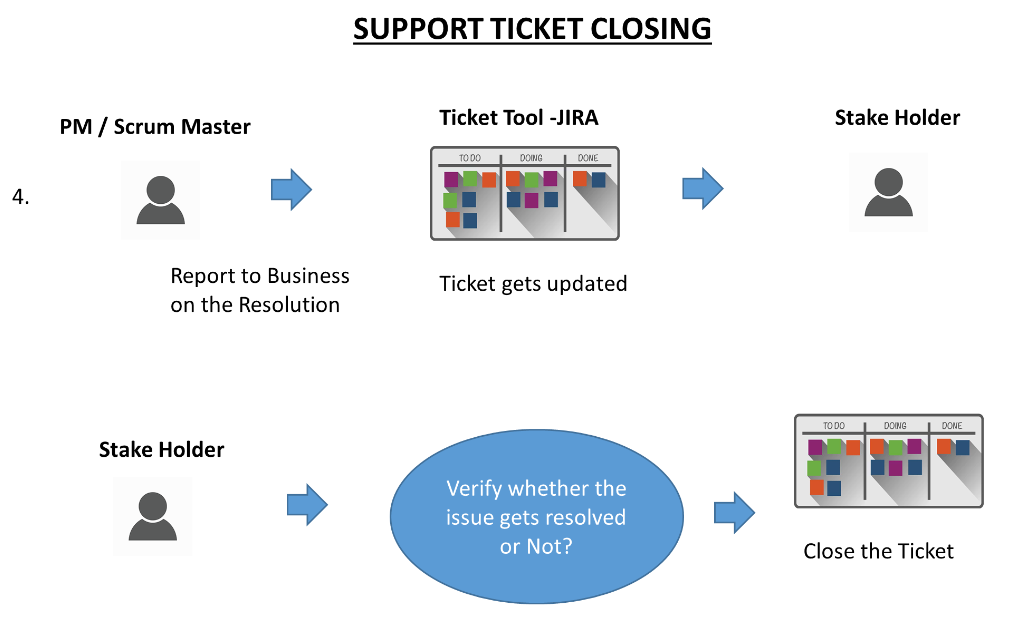
# Section -8 Initial Support Engagement

**Supporting Services Approach after Product Launch**

**SCRUM –KANBAN BOARD Approach.**

* Ticketing tool (JIRA) will be configured for issue logging (Ticket).
* Issue Ticket will be entered / communicated to the team, and it will be prioritized as per the SLA.
* Ticket will be allocated to the Specific Point of Contact from the Team.
* Ticket will be analysed / resolved as per the SLA by the Team member and updated against the ticket.
* PM/Scrum Master will verify and update to resolution the Business SPOC
* Weekly / Monthly reports will be published with statistics.





# Section -9 Delivery Commitments

**Capability for customization, extensibility and maintainability of the platform**

* Features are Role based.
* Features are Rule based.
* Configurable System Parameters.
* External Adaptor factory implementation.

**Interface with third Party applications**

* API based Secured Integration and / or other secured hand shaking methods to access and use third party services, systems and devices.

**Scalability, Throughput, Performance of the platform**

* Elastic Solution Architecture,
* Elastic Technical and Deployment Architecture to attain higher Scalability, Throughput & Performance.

**Administrative Capabilities**

* Role based access controls
* Page level access
* Field level access

**Support for on premise and private cloud deployment model**

* Containerized hosting model to ensure hosting platform independent.

**Capability and scope for IT knowledge Transfer**

* User Manual
* Technical Design Documents
* Functional Documents
* KT Videos & Audios

**Security and Risk Compliance**

* Adhering Risk Assessment Practices
* Risk Analysis and evaluation
* Security compliance testing
* Risk mitigation Plans

**Logging and Monitoring**

* Secured and scoped Logging and Log visualization with custom built or third-party tools
* Application / API / utility Health check measures

**Strong AES-256 encryption for data security at rest**

* AES 256 encryption
* Separate &Unique key store for various applications
* Vault stored encryption file stored keys
* In-memory encryption processing – Self-construction & destruction of encryption logics at service start and stop.

**Mobile security to protect data on mobile devices**

* JWT & OATH2 Token based authentication
* Device Unique, mobile number unique and OTP unique based security key construction

**Vulnerability Management Program**

* Security Measures for OWASP Top 10 Vulnerabilities

1. Injection

2. Broken Authentication

3. Sensitive Data Exposure

4. XML External Entities

5. Broken Access Control

6. Security Miss-configuration

7. Cross-Site Scripting

8. Insecure De-serialization

9. Using Components with Known Vulnerabilities

10. Insufficient Logging and Monitoring

**Implement segregation of duties and role-based access control with authentication and authorization, using federated CLIENT identities.**

* Security Headers with Application and Role
* ID & access Management
* Application & Role based features access
* Page level Access
* Control Level Access

**Absence of obsolete or known insecure technologies / products**

* As part of OSWAP security measures and Security Scans, this will be eliminated at the time of choosing the tools and products.

**Solution deployment in multi-tier architecture model**

* Our solution is compliant with this model as we follow
* Micro-service-based approach
* Container based approach
* Container Orchestration for scalability

**Segregation of solution layers logically and physically**

* Every Individual module is built as a separate Micro-service with DBs associated with it.
* Multiple DB vendors can be used based on need.
* Scaling, clustering can be done separately.

**Functionality of software be operational behind Web Application Firewall**

* Web Application Firewall will be configured with DMZ & Application gateway
* Reverse proxy API gateway will be used
* Auth and App layers be separated

**Audit logging**

* Centralized and Customized Audit
* Dynamic Configuration
* Actor, Action & Time-based Audit Configuration
* History Based Configuration
* Generic utility tool to retrieve Audit Data
* Switch on / off capability

**QMS Strategy**

**Quality Manual**

* Create a quality guide to follow quality management that reflects the organizational values of Customer.
* Outline of the organization’s quality objectives and policies
* Highlight the quality procedures to achieve through the implementation.
* Use a flowchart to show the documentation of essential quality management processes.

**Quality Objectives**

* We define the quality management system to make sure we define every person’s responsibilities are clearly defined.

**Data management**

* Define clear data management to avoid scenarios of operational inefficiencies, poor customer experience, and other compliance risks.
* Data management procedures should address documentation and records, collection methods, sources, disposal, and storage.

**Internal Processes**

* Each process involved in transforming inputs to finished products and services their links and relations and flows should be clearly defined, understood, and communicated.

**User Satisfaction from the implemented services**

* We will evaluate and manage the quality of user satisfaction.
* Define different methods and tools you intend to use to measure the levels of user satisfaction by continuously reviewing satisfaction surveys, complaint procedures, or analytical tools to assess satisfaction trends.

**Documentation**

* We need to record and design clear documentation of intentions to meet QMS continuous improvement standards.
* The documentation should contain safety designs, corrective action, quality planning procedures, and compliance requirements.

**Quality Instruments**

* We will be using quality tools to continuously measure the progress and success of the support services to ensure the project objectives are met as per the organization standard.

**Training plan and schedule**

* Schedule starts once the UAT is over and one month training to the people as per their role in this system.
* Parallel support for a month with trained executives.

# Section -10 Product development plan

six months product s

## Development Timeline:

|  |  |  |  |
| --- | --- | --- | --- |
| **Initial construct** | 4 months | **Support & evolve** | 3 months |

|  |  |  |
| --- | --- | --- |
| **No** | **Project task item** | **Delivery timeline** |
| 1 | * High level design, * System architecture, * Data model, * UX to UI conversion, * Test strategy & test plan * Project plan. * Requirement Iteration & finalization. | 1 month |
| 2 | * Low level design * Middleware API development * Business logic development * Ui-API wiring * Unit testing * System testing * Test automation phase -1 | 1 month |
| 3 | * INFRA readiness * 3rd party API integration * Unit testing * System testing * Integration testing * Test automation phase –2 * Test data readiness * Build and deployment readiness. | 1 month |
| 4 | * Test execution * Bug fixes * Build and deployment. * Complete testing (system, integration, security, load testing). * User acceptance testing * Production roll-out for current scope | 1 month |
| 5 | * Production support * Application hardening measures * Documentation * Knowledge-based construct * Bug fixes * Small enhancement development * Professional services. | 3 months |