Inventory Management Portal

Contents

1.0 Problem statement

The purpose of the requirements document is to systematically capture requirements for the project and the system “**Inventory Management Portal**” to be developed. The application should be Cloud Native Architecture with Microservices. Both functional and non-functional requirements are captured in this document. It also serves as the input for the project scoping.

About the System

The client would like to develop an inventory management Portal (IMP) application for different locations to view the list of products available in specific location and the products available in which location. Will have login page for user and admin. Admin page has access to edit the inventory detail.

Scope of the System

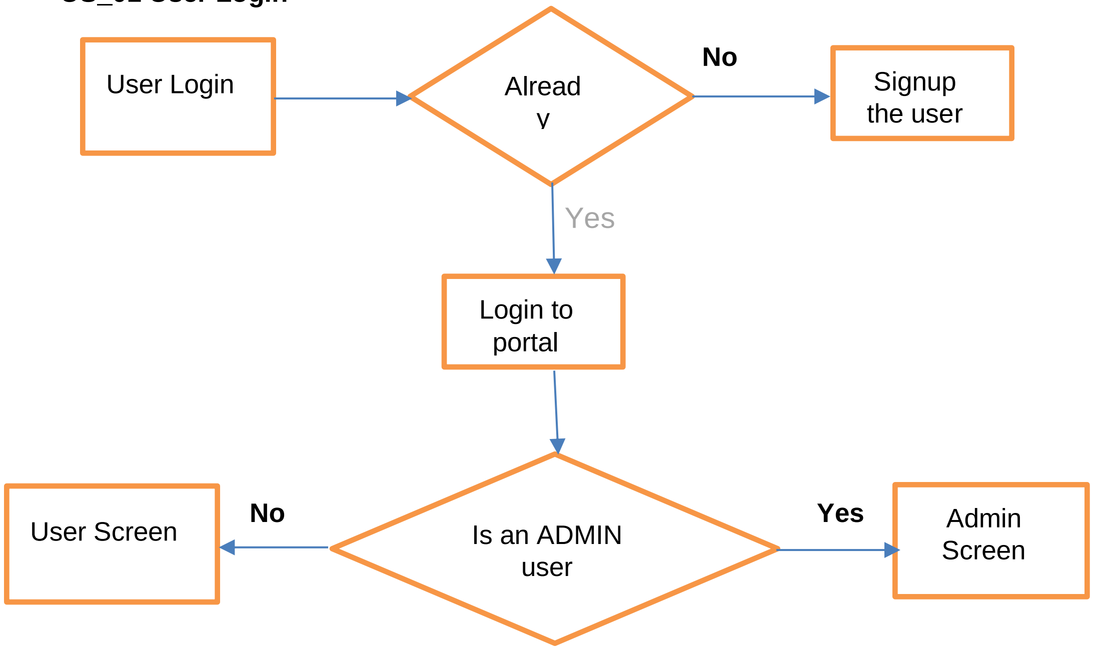
The scope of the system is explained through its modules as follows

- Registration – used by user to register the details of self-information into the system. The system stores the details of the user in the system and able to edit it.
- Edit Inventory Detail Admin – Admin will have access to add inventory during the day.
- List Available Quantity – User and Admin can view the available inventory based on the selected location/product.

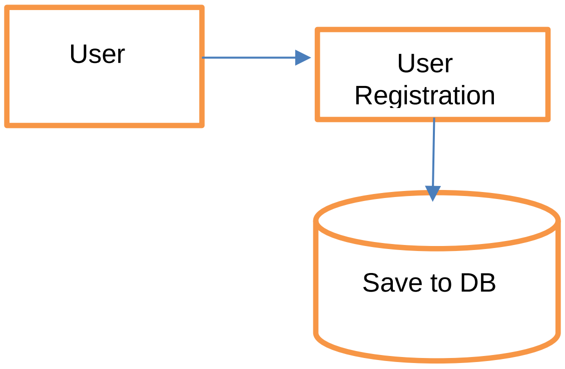
2.0 Architecture Diagram for the Problem Statement

Use case Diagram

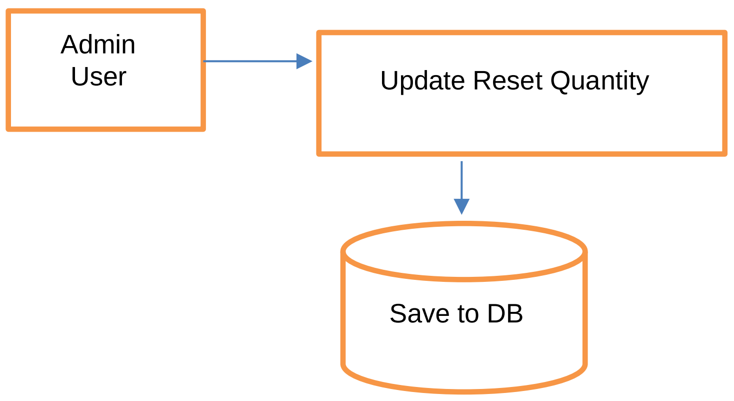
US_01 User Login



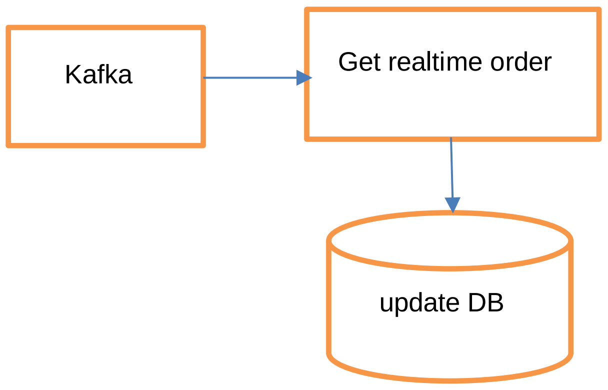
US_02 User Registration



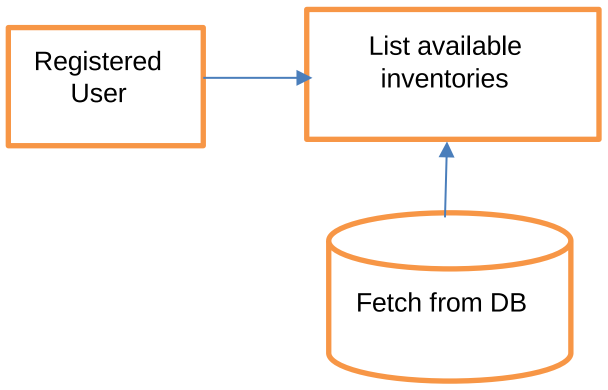
US_03 Edit Inventory



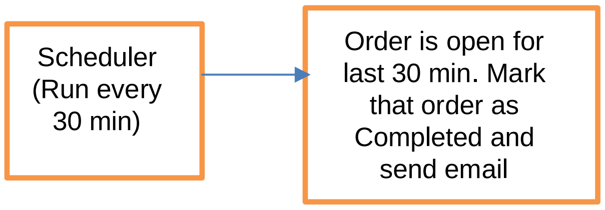
US_04 Accept Orders



US_05 View Available Quantities



US_06 Trigger Email


Scheduler (Run every 30 min)
Order is open for last 30 min. Mark that order as Completed and send email

3.0 Database Diagram for the Problem Statement

Below table in MySQL/SQLServer

Table 1 : User in MySQL for authentication

- UserId(Varchar(100), NotNull)
- Name(Varchar(100), NotNull)
- Email Address(Varchar(100), NotNull)
- Contact No(Number, NotNull)
- DOB(Varchar(100), NotNull)
- UserType(Varchar(100), NotNull)

Below table in CB/Cassandra

Table 1 : Order

- OrderId(varchar(10), Not Null, unique)
- OrderDateTime(DateTime2, NotNull, Defaullt currtime)
- LocationNbr(Number(6), Not Null)
- MaterialId(varchar(10), Not Null)
- OrderQty(Number, Not Null)
- OrderStatus(varchar(100), Not Null)

Table 2: ResetInventory

- LocationNbr(Number(6), Not Null)
- MaterialId(varchar(10), Not Null)
- ResetQty(Number, Not Null)
- ResetDateTime(DateTime2, NotNull, Defaullt currtime)

Table 3: Inventory Table :

- Id(Pk, int, identity)
- LocationNbr(Number(6), Not Null)
- MaterialId(varchar(10), Not Null)

- ResetQty(Number, Not Null)
- OrderQty(Number, Not Null)
- AvailableQuantity(Number, Not Null)
- UpdateDateTime(DateTime2, NotNull, Defaullt currtime)

4.0 Technical Scope

Compute & Integration	As an application developer, develop the application as a microservice architecture. And implement as follows: 1. Follow the Single Data Store per microservice practice 2. Document REST endpoints with OpenAPI or Swagger
Security & Identity	As an application developer: 1. Restrict the access over all write operation (secured operations) by adding authentication
Database & Storage	As an application developer: 1. Ensure Application is implementing ORM with .Net Core 2. Restructure the same as per microservice architecture 3. Use SQL SERVER for maintaining data of the microservice 4. Optimize existing implementation for Transactions
Governance & tooling	As an application developer: 1. Containerize the complete application 2. Perform unit testing of your application and do proper CI/CD
Code Quality/Optimizations	1. Optimize the DB implementation using index search 2. Use SonarQube to scan the backend application for security vulnerabilities 3. Should have written clean code that is readable 4. Should have written testable code

5.0 Use case details

User Story #	User Story Name	User Story
US_01	Login	As a user, I should be able to login to the portal if already registered. If not, I should be able to register as a new user.

US_02	User Registration	As a User, I should be able to register my details in the system. Acceptance criteria: User should be able register the details in the system and it should be saved in the database. Capture the details like Name, Email Address, Contact No, DOB, UserType.
US_03	Edit Inventory	As an admin, I can update the reset quantity for any material in any location. **ResetInventory Table:** LocationNbr, MaterialId, ResetQty, ResetDateTime - Accept Reset Inventory - Log the data in ResetInventory table. - Recalculate Available Inventory and update Inventory Table Formula : AvailableInventory = ResetInventory – OrderQuantity
US_04	Accept Order	As a realtime application, it should support to get order in realtime and update the available quantity Get realtime order from kafka topic. **Order Table :** OrderId, OrderDateTime, LocationNbr, MaterialId, OrderQty, OrderStatus - Accept order - Log the data in order table - Recalculate Available Inventory and update Inventory Table Formula : AvailableInventory = ResetInventory – OrderQuantity
US_05	View Available Quantity	As a User, I should be able to view the available quantity for any location and Material. As an admin, I should be able to view the pending orders and available quantity and order fulfilled in last 3 hours from given location and the top 5 material which has more order in the last 3 hours. **Inventory Table :** Id, LocationNbr, MaterialId, ResetQuantitty, OrderQty, AvailableQuantity and UpdateDateTime
US 06	Trigger Email	As an Admin, I should get alert if the order is in-progress for more than 30 min.

6.0 Functional/Non-Functional Requirement of the Problem Statement

US -01 Login

The user should be able to login to the portal using the registered username and password.

For new users, provide a signup link which will enable the user to register to the system.

Validations

1. Username should not contain (Min. 5 character and Max. 8 character)
2. Password should be greater than 8 characters long and should contain atleast one uppercase,one numeric and a special character
3. New user should be able to signup by providing his username, dob, userType(ADMIN, USER)and valid email id and password . All fields are mandatory.
4. DOB(Date of Birth) should not be greater than system date.
5. If the user enters an invalid username/password show error message appropriately.
6. On Successful login, USER/ADMIN can be redirected to appropriate screens.
7. User/Admin will take to view available quantity page and Admin will have option to go to Edit Inventory page.

US -02 User Registration

Business Rules & Validations

1. User id should be generated automatically during the time of registration and should be shown in the success message.
2. The user name should contain only alphabets and space.
3. All fields are mandatory.
4. Contact number should be 10 digits.
5. Email id should contain @ and. symbols.
6. User id should be in the format of ‘R-XXX’.XXX should be random numeric of 3 digits.
7. Based on the DOB, age will be calculated.
8. Age should be greater than 18.

Validations

- All fields are mandatory.
- Age should be greater than 18.

Edit Inventory

US_03	Edit Inventory
Description Admin should be able to edit the reset inventory.	

Input Parameters Below are the input parameters. Locationnumber, Material Id and Reset Quantity
Business Rules & Validations • All fields are mandatory. • Locationnumber should contains 6 digit • MaterialId should have length of 10 number. It can start with '0' as well. • Reset Quantity should be greater than the existing reset quantity for given location and material. • During available inventory recalculation, consider only in-progress order quantity for that location and Material • Reset quantity should not be greater than 1000. • If no entry in order table, then make order quantity to 0 in inventory table. • Formula : AvailableInventory = ResetInventory - OrderQuantity

Accept Order

US_04	Accept Order
Description Backend Application should accept order in realtime	
Input Parameter OrderId, OrderDateTime, LocationNbr, MaterialId, OrderQty, OrderStatus • All fields are mandatory. • Locationnumber should contains 6 digit • MaterialId should have length of 10 number. It can start with '0' as well. • OrderStatus can be in-progress or Completed	

Business Rules & Validations
• Consumer order data from kafka topic. • If LocationNumber and Material is not available in inventory table, then just log in order table and no need to recalculate in Inventory. • If Order quantity is greater than AvailableQty in Inventory, then mark the status as NOT FULFILLED. Don't accept partial order. • Formula : AvailableInventory = ResetInventory - OrderQuantity

US_05	View Inventory
Description GET Available Inventory and metrics data	
Input Parameter Scenario 1 : Get By LocationNumber Scenario 2 : Get By LocationNumber and Material Id Scenario 3 : Get all the pending orders Scenario 4 : Get completed orders in the last 3 hours Scenario 5 : Get the top 5 materials which sold more quantity in last 3 hours.	
Business Rules & Validations • All input fields are mandatory	

Accept Order

US_06	Trigger Email
Description Trigger email based on scheduler	
Input Parameter • Scheduler and Order data	
Business Rules & Validations • Get the order which is in-progress for more than 30 mins. And send email to admin.	

Service Requirements

US_01 Login

When the user logins by providing username, password, the details should be sent to the POST method and checked in the db if the user exisits. If the user does not exist, send exception response. If user found return 200 with success message.

When the user enters the details in signup page, they should be sent to the POST method and saved in db.

Mandatory validation should be done for username, dob, userType ,email id and password

User Type can be presented as dropdown with ADMIN/USER options to configure Access privilege to the logged in user.

US_02 User Registration

Once the user enters the details, they should be sent to the POST method and saved in the db.

Mandatory fields should be validated as mentioned in the rules above and 400 exception response should be sent with the missing field details.

When the details are saved successfully, the service should response 200 ok along with success message.

If there are any exceptions while connecting/saving to db , the service should throw corresponding error with error status as 500.

US_03 Edit Inventory

Add Post Method to save the inventory data first in ResetInventory and then Inventory table.

Once the user enters the details, they should be sent to the POST method and saved in the db.

Mandatory fields should be validated as mentioned in the rules above and 400 exception response should be sent with the missing field details.

When the details are saved successfully, the service should response 200 ok along with success message.

If there are any exceptions while connecting/saving to db , the service should throw corresponding error with error status as 500.

US_04 Accept Orders

Create Kafka Consumer to get the order data.

If the mentioned validations fail, then log and ignore the

- Log the data in order
- Recalculate Available Inventory as per the formula and update in Inventory.

US_05 View Inventory

Create GET API to get the available quantity based on:

- Location number
- Location number and material id

Create GET API for admin :

- To get the pending orders
- To get Available quantity based on Location number and Material Id
- To get Order fulfilled in last 3 hours
- To get the top 5 material which sold fast in the last 3 hours.

US_06 Trigger Email

Have a scheduler for every 30 min to alert admin if the order is in-progress for more than 30 min.

Expected Deliverables

The following deliverables are expected as outcomes:

- Application Code base
- Readme document on the complete application
    - Setup of the application
    - How to run the application
    - Any inference
    - Screenshot of UI results
- Reports:
    - Unit/Functional Test Report

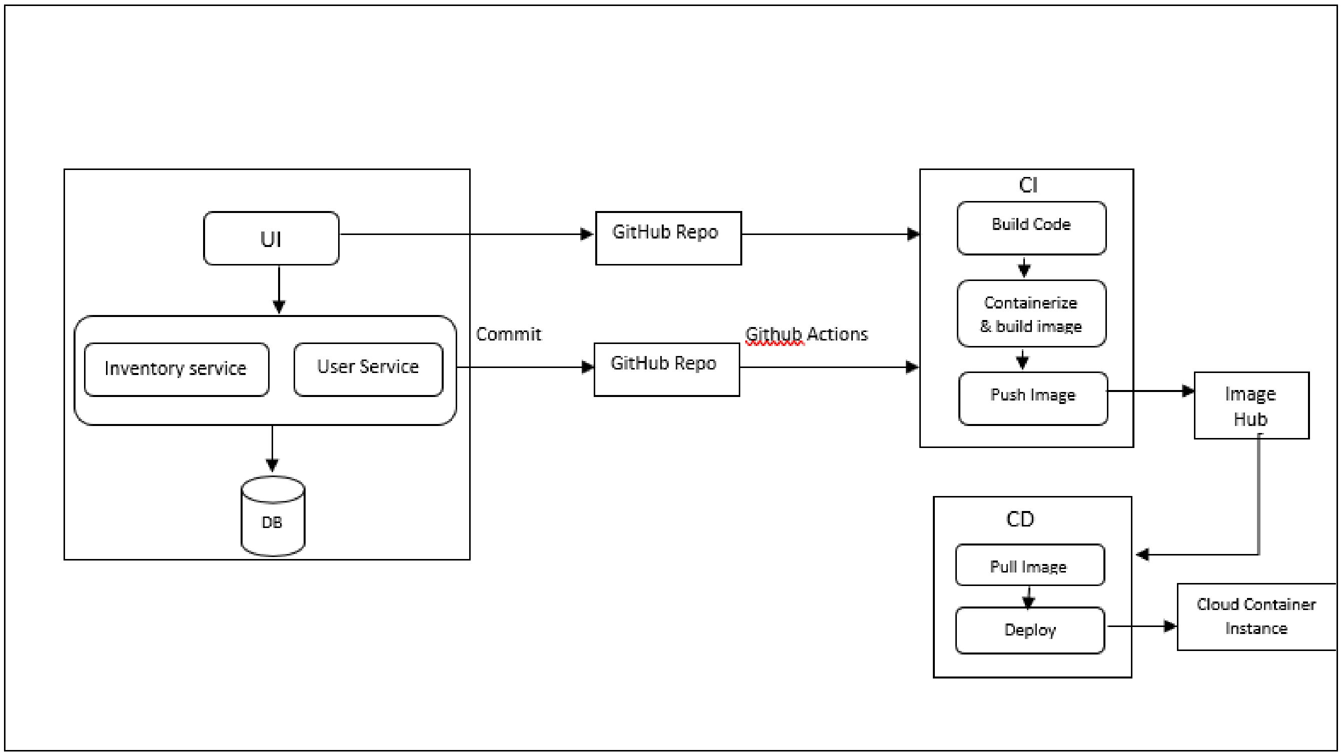
7.0 Skills to develop the project

List the Technology based on your respective technology stack, that will be used to develop the project.

Skill Stack	Java FSE
Front end	Angular, CSS, Typescript/JavaScript, Karma/ Jasmine
Service End	Java, Spring boot, Spring Security Framework
Database	SQL Server/MySql, Couchbase/Cassandra
Source Control	GIT, GitHub, GitHub Actions for CI & CD.

Cloud	Azure
Unit testing	Junit, Mockito framework

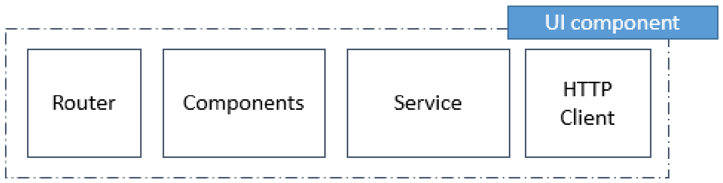
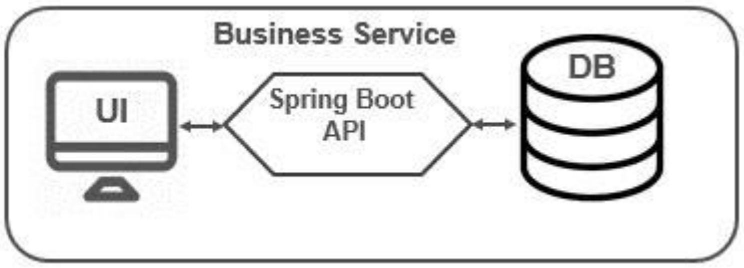
8.0 Deployment Architecture



9.0 Milestone

The milestone for the project use is given below

Milestone	Duration	Topic
Milestone – 1	7 days	Design and develop the UI for the application

		 UI component Router Components Service HTTP Client • Implement user-stories using Angular framework • Design application with minimum backend or mock backend as the focus for milestone-1 in on frontend skills • Implement forms, data binding, validations Use appropriate unit test framework
Milestone -2	12 days	Develop the required APIs for the application  • Using Microservices architecture • Follow coding standards • Follow standard project structure. • Log all request details. • Log errors. • Message input/output format should be in JSON (read the values from the property/input files, wherever applicable). Input/output format can be designed as per the discretion of the participant • Database connections and web service URLs should be configurable. • Use browser / POST Man to invoke APIs • Run SonarQube for code quality. • Implement Junit Testing
Milestone -3	7 days	• Integrate service layer with UI component. • Setup CI & CD pipelines. • Dockerize the application.

10.0 Evaluation rubrics

Angular	• Associate must have used Angular Components, Modules, Databinding, data validation, CLI commands. • Associate must have used Forms and Forms validation • Associate must have used Directives • Associate must have developed Reusable Components • Associate must have followed coding standards Why Angular is chosen to design this application? Explain the application requirement along with use case Distinguish the speed and performance of using Angular for this design over other frameworks Distinguish the advantages of using IDE and Templates of Angular for this design Distinguish the scalability and accessibility over other frameworks Distinguish the data binding of angular over other frameworks Distinguish the security features of Angular over other frameworks
Microservice	• REST controller • Follow controller ->service->Dao model • Entity and Model classes • Recommended JPA Data binding • Appropriate logging statements • Exception handling
Java	• SOLID Principles • Design patterns for microservices • Cloud Design Patterns • 12 Factor Principles • Design and create applications with proper design principles
Spring Boot	• Spring Boot should be used in the application. • Overview of Adjacent skills of SpringBoot : (Should be evaluated during Project Demo) • Spring MVC, Jersey,Micronaut ,Dropwizard ,Eclipse MicroProfile • Advantages of using SpringBoot over other frameworks • Disadvantages of using SpringBoot compared to other frameworks • Why SpringBoot is choosed to design this application?

Docker	• Dockerize the application • Build docker containers • Push your Docker images to an Amazon ECR repository with the docker push command
Azure	• Deploy application to azure namespace • Create deployment file