

## **Asian Paint safe painting Application: (Employee perspective)**

**Project overview:** The Asian Paint Safe Painting Service application is designed to offer customers a safe, end-to-end solution for their painting needs, including consultations, colour selection, and professional painting services. The application streamlines the entire process, from booking services to receiving updates and finalizing payments. My Role and Responsibilities are:

### **1) MVC Architecture Implementation:**

- Applied the **Model-View-Controller (MVC)** pattern to organize the code structure, separating concerns between the front-end, business logic, and data layers.
- This ensured the application was modular, scalable, and easier to maintain.

### **2) Backend Development using Spring Boot:**

- Designed and developed the **backend** with **Spring Boot**, handling business logic and services for the application.
- Managed user interactions and service workflows efficiently.

### **3) Repository Development with Spring Data JPA:**

- Created **repositories** to interact with the **MySQL** database, performing **CRUD operations** efficiently.
- This allowed smooth handling of customer data, service bookings, and transactions.

### **4) Spring Boot Controllers:**

- Developed **controllers** to manage the interaction between the front-end (views) and backend.
- Ensured smooth data flow between the user interface and backend services.

### **5) Email Integration:**

- Integrated **email-sending services** that allowed the application to automatically send notifications to customers regarding booking confirmations and service updates.
- Enhanced customer communication through timely updates.

### **6) PDF Document Generation:**

- Implemented functionality to generate **PDF documents** using APIs, allowing the system to send detailed service reports to customers.
- This Helped streamline reporting and record-keeping.

### **7) Database Setup and Schema Design:**

- Installed **MySQL** and designed the **database schema** for efficient data storage and retrieval.
- Defined **entities** for handling customer, service, and booking-related data.

### **8) Dynamic Views with JSP and JSTL:**

- Created dynamic views using **JSP** and **JSTL tags** to display backend data effectively on the front-end.
- Enabled users to view real-time data related to their bookings and service status.

### **9) API Testing with Postman:**

- Thoroughly tested APIs using **Postman** to ensure correctness and reliability.
- Documented the APIs for future reference and use by other developers.

### **10) Collaborative Development:**

- Worked closely with the team, ensuring that the backend, front-end, and database components were well integrated.
  - Ensured smooth deployment of new features and improvements based on feedback.
- 

### **Profinch Solutions Pvt Ltd:**

At Profinch, my contribute to building and enhancing fintech ecosystems. My role involves providing consulting services to improve core operations and processes, I'm part of a team that's passionate about making fintech solutions more efficient and effective.

### **Firstly, Worked on TISA project (Teachers Savings and Loans Society)**

TISA's main activity is to provide financial services and products to the retail and commercial segment, including Small and Medium Enterprises (SME). And support clients grow their individual and business wealth.

### **My roles and responsibilities in this project are:**

Here I am using as Atumverse application this is a product of profinch solution. Atumverse application is run through workflow system for that we are using Camunda BPMN workflow.

**Step 1** - Camunda is an open-source platform that uses a workflow engine and decision engine to automate business processes.

**Step 2** - Creating a Json form schema for all the stages of activities going to create as 3 files are: View, review, and summary and configure to all Json schema properties into the Json file.

**Step 3** - Once all the Json files are completed then configure into the database, here we are using dbeaver database we configured our Json file from all the stages.

**Step 4** - For the front-end part we are using TypeScript and Angular js and IDE using as Visual studio code. From here we configure our Atumverse application and configure exact BPMN diagram name inside the source folder.

**Step 5** - We are transfer our all files local into the server, for the server we are using WinSCP run the server using putty. Once this process done will check our UI, based on the business needs we need to modify or add any feature into the application. we have written a code into the **component.ts** file in our angular application. Also, we need change those required fields in properties file as well.

### **Then I moved to the Uzbek cards domain:**

In this project I have used tools like core java, spring boot, hibernate microservices and restful webservice. for database used as dbeaver and testing the APIs using postman tool and swagger UI.

### **My roles and responsibilities are:**

Here developed Atumverse microservices, in this we have four modules are:

Uzbek-card-entities -> This module act as a pojo class.

Uzbek-card-core -> This module act as a service which will have a dependency of entities.

Uzbek-card-web -> This module like a controller- which will have a dependency of service layer.

These are the three is basics of modules and based on the requirement we must add a spring batch module as well. For the database we are using dbeaver. Here,

- Creating a FTL (FreeMarker Template Language) files and populated with them with necessary hardcoded values.
- Based on the Camunda BPMN Diagram triggers workflows and services within the microservices architecture.
- Adding core business objects into the entity module. In repository layer interacts with the database and provides CRUD operations for the entity.

- The core module contains business logic. In this case, it processes data coming from the BPMN workflow. The Camunda BPMN flow would trigger service methods as per the business rules.
- The web module contains handles incoming requests and sends responses back to the client.
- Using Eureka Server, it is used to register the microservices for service discovery, allowing services to communicate with each other dynamically. Feign Client is used to simplify communication between microservices. Using Spring Boot Actuator can be used to monitor the performance of the microservice.
- Once the all the service up and run test all the APIs using postman, as well as conducted UI testing with the Automverse application.

## **Digital Flex (Fifth Third Bank)**

### **Worked Core Transfer Domain**

**Project Overview:** "Following my time at Profinch Solutions, I transitioned to the **Digital Flex team** for our second client, **Fifth Third Bank**, where I worked within the **Core Transfer Domain**. This project followed an **Agile Scrum** methodology, enabling us to deliver high-quality solutions adaptively."

Skills used in this project Core Java, Spring Boot, Hibernate JPA, Restful Web services, API development, Docker, Splunk, Dynatrace, Digital.ai, Postman tool and swagger UI, PostgreSQL, Splunk etc

**My Role and Responsibilities:** In this project Developed and maintained microservices using Core Java and Spring Boot for core banking operations.

- I designed and implemented **APIs** for various functionalities, ensuring they met both client requirements and industry standards. Also documented all those APIs in swagger UI.
- Developed functionality for creating **Certificate of Deposit (CD) accounts** that integrates with the account domain to transfer core funds to internal or external savings accounts and implemented a process to track transfers via an ID, enabling fund retrieval after a 5-6 working day period.
- Conducted through API Testing to validate their functionality, reliability, and security.
- Conducted **code reviews** and **debugging sessions**, adhering to best practices and coding standards.
- I also provided **technical support** and solutions to resolve issues that in the core transfer domain, ensuring system stability and performance.
- Working within an **Agile** framework, I actively participated in sprints and utilized **Digital.ai** to manage and track user stories effectively. This enabled our team to remain organized and focused on delivering features that added value to our client.