

EVALUATING THE APP & PROCESS USED

CONTENT

SECTION 1: INTRODUCTION.....	3
1.1: THE APPLICATION – PARCEL DELIVERY SYSTEM	3
SECTION 2: SOFTWARE ARCHITECTURE	4
2.1: WHY I USED MVC ARCHITECTURE PATTERN?	4
2.3: ADVANTAGES OF USING MVC	5
SECTION 3: SOFTWARE DESIGN PROCESS	5
3.1: USDP APPROACH AND IT’S ADVANTAGES.....	5
3.2: SOFTWARE DESIGN PATTERN USED	6
SECTION 4: BUILDING THE DELIVERY APP	7
SECTION 5: RECOMMENDATION	ERROR! BOOKMARK NOT DEFINED.

SECTION 1: INTRODUCTION

The design and development of an enterprise software solution come with numerous challenges; therefore, the need to leverage an industry-standard design blueprint, architectural best practices, suitable design patterns will help mitigate complexities and address practical enterprise concerns such as productivity, agility, testability, and flexibility. In addition, issues like maintainability and scalability could be quickly addressed without affecting efficiency and application security (Nitelius, 2021).

This report demonstrates the steps taken in specifying, designing, implementing, and testing a *Parcel Delivery System* using with the Java Platform, Enterprise Edition (Java EE) using widely adopted best practices to handle the separation of concerns (i.e., separating business logic from view code in a dynamic application), layering and other important factors discussed in this report.

1.1: THE APPLICATION – Parcel Delivery System

The developed application, a parcel delivery system, is a Java EE based application built with NetBeans leveraging MVC (Model–view–controller) for parcel delivery and tracking. The presentation layer uses XHTML, and the application runs on SQL (Structured Query Language) powered by JDBC (Java Database Connectivity).

The application uses three use cases (customer, manager, and driver) and five events (book a collection, deliver a parcel, track, and collect a parcel).

SECTION 2: SOFTWARE ARCHITECTURE

Architectural pattern entails a general, reusable solution to a commonly occurring problem in software architecture within and given context (Wikipedia, 2021).

In practice, software architecture focuses on relationships and how the elements and components of an application interact with each other. There are numerous architectural patterns such as Layered, Master-slave, Event-Bus, Blackboard, Pipe-Filter and MVC.

Even though the aforementioned patterns are used to simplify complex applications into more manageable partitions, MVC was used in this project.

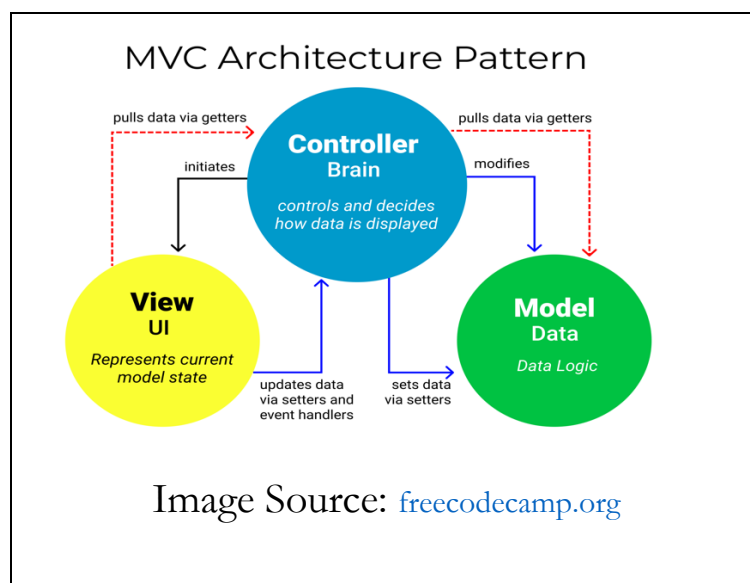
2.1: WHY I USED MVC ARCHITECTURE PATTERN?

From my experience in this module, I realised that MVC is easy to understand and apply to enterprise applications as it makes an application easy to manage (without interference) and collaborate on by using separation of concerns thus:

Model – Contains the backend data logic.

View – Takes care of the graphical user interface.

Controller – Connects Model to view to control data display.



2.3: ADVANTAGES OF USING MVC

1. Easy to understand and apply to enterprise applications
2. Wide usage, and less complex when used for collaborating project
3. Effective decoupling
4. Allows of scalability
5. Ensure ease of maintainability and ease of expansion.

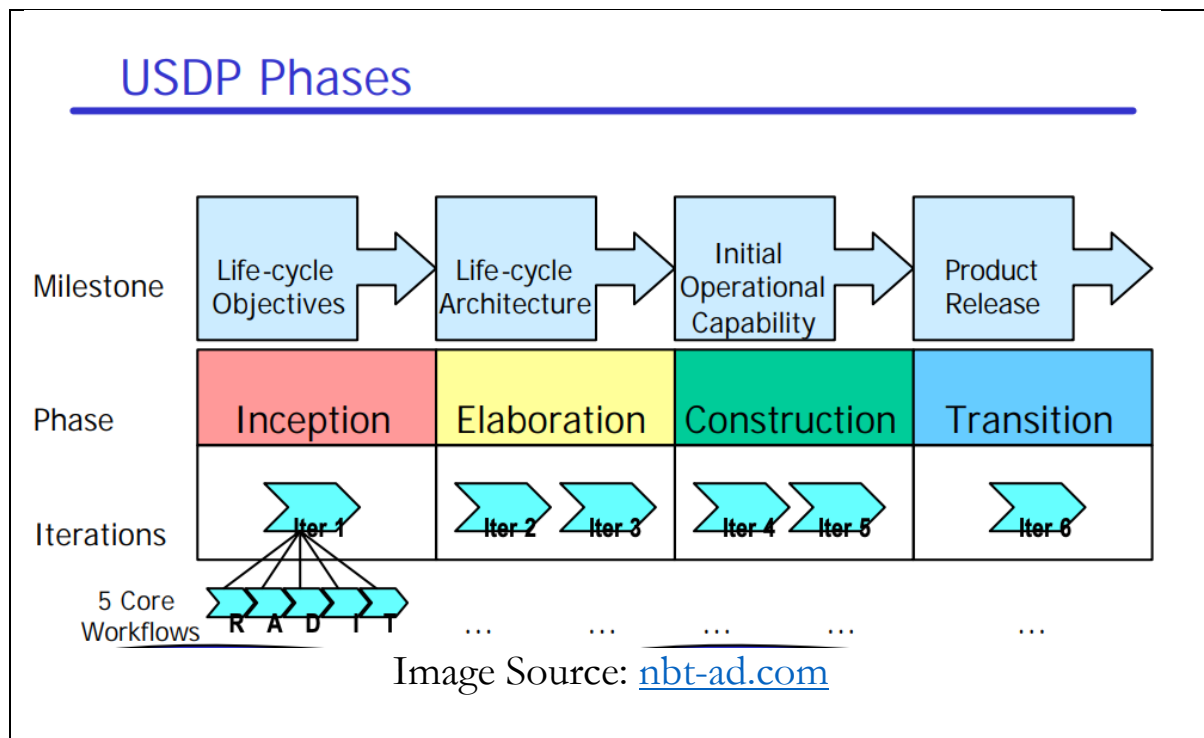
SECTION 3: SOFTWARE DESIGN PROCESS

Applying design processes is considered best practice, as the design of the application is quite tried and tested, resulting in better readability of the end code. They are basically created for and used by Object Oriented Programming languages such as Java - in which the code for my project was written. Some design processes are USDP, Rational Unified Process, OpenUp, and Agile Unified. The USDP approach was used in my design (NBT AD, 2021).

3.1: USDP APPROACH AND IT's ADVANTAGES

I used the Unified Software Development Process (USDP) for iterative and incremental workflow due to the following advantages:

1. It makes the development of JSF applications easy to implement
2. It allows for a smooth transition between different stages of software development.
3. It provides a standard graphical language for expressing and discussing design ideas and requirements, especially in parallel development.
4. It allows professionals to allow us to adjust the quantity and quality of project documentation as required by the project time, volume, budget, and future expansion needs.



Submitted my report in the copy of a design file showing the following:

1. Use case diagram
2. UML Activity diagram
3. Analysis diagram
4. Class diagram
5. Entity diagram

3.2: SOFTWARE DESIGN PATTERN USED

Software design patterns are designed to level solutions, and they describe how to address recurring software engineering problems faced by software engineers. It explains how to tackle the issues encountered by creating a solution.

In my project, I used the following design patterns:

1. Facade
2. Factory
3. Singleton
4. Builder
5. Decorator

SECTION 4: BUILDING THE DELIVERY APP

The roadmap I used in building the application is highlighted below:

JSF Web Pages

This shows the definition of the web interface using XHTML pages. Here, my data model was mapped to my forms using JSF Expression Language.

Modelling

My data table properties were included in my managed beans files to allow for easy instantiation of the class objects. I ensured that each instance of the class corresponds to a row in the database table.

Java Beans Annotations

These were used to map entity classes to their respective database table; therefore, eliminating the need for XML configurations.

Database

I configured my JDBC to run on derby. The connection to the database instance was created and added to my class directory.

REFERENCES

- [1] Nitelius, SK (2021), Tracker. Available at: REF: <https://sknitelius.gitbooks.io> (Accessed: 7th August 2021).
- [2] Oracle (2021), Java Documentation. Available at: REF: <https://docs.oracle.com/javase/7/firstcup/java-ee001.htm> (Accessed: 12th August 2021).
- [3] Josh, J. (2014) Java Faces – Introduction by Example. New York: Apress Media.
- [4] New Bulgarian Technologies AD (2021), USDP/UML. Available at: REF: http://www.nbt-ad.com/USDP_presentation.pdf (Accessed: 16th August 2021).