LEARMONTH Anders

l018160a@student.staffs.ac.uk

Parcel tracking system: Design, implementation, testing and Review

Contents

[Introduction 1](#_Toc58533407)

[Design 1](#_Toc58533408)

[Choices made 1](#_Toc58533409)

[Design patterns 1](#_Toc58533410)

[Alternative design strategies 1](#_Toc58533411)

[Build 1](#_Toc58533412)

[Testing & Results 2](#_Toc58533413)

[JUnit tests 2](#_Toc58533414)

[Integration tests 2](#_Toc58533415)

[Future development opportunities 2](#_Toc58533416)

[Conclusion 2](#_Toc58533417)

[Appendices 2](#_Toc58533418)

[References 2](#_Toc58533419)

Write a report

Write a report (up to 1,000 words) in which you critically evaluate your application. You should consider:

• the design, describing alternatives and justifying your selections;

• the results of testing, making recommendations for further development.

Critically reflect on and draw conclusions about the fitness for purpose of an application

You should ensure that your report presents evidence to support the awarding of marks in each criterion of the rubric

# Introduction

This report details the design, build, testing and review of a parcel tracking system which allows users of multiple pre-assigned roles to perform differing actions against parcels, orders, and their current stage in the delivery process. This could be used for an online shop looking to migrate to a single enterprise system with all data accessible in one location.

The application uses Java EE to allow interaction with a Java database and includes JUnit to enable automated testing at all stages of development and to safely add features to the system in the future without the risk of unknown breakages elsewhere.

# Design

## Choices made

Justification for the choices/selections of tools/things I used/did with this project

Db is complex but manageable and highly scalable allowing future additions to complement the overall system without negatively impacting the existing areas.

### Design patterns used

Which 5 design patterns did I use, why did I use them and how did I implement them?

DTO pattern – data transfer object

Table gateway pattern –

Command pattern –

Factory pattern –

Façade pattern –

## Alternative design strategies

Describe alternative methods available that could have been used in this project, including their pros and cons.

# Build

Any troubles when building, libraries, dependency injection?, …

# Testing & Results

Include results from testing, using JUnit tests, and why? (allow modular improvements while ensuring the overall system still functions as expected against the already outlines unit tests, nothing breaks)

My test plan… including edge cases in my testing, checking expected pass and fail results

Using black box testing with known inputs, do we get the expected output(s)?

## JUnit tests

Benefits of automated testing, how did I use this, why did I use this, alternative(s) to this are manual testing and… pros/cons

## Integration tests

With my use cases

# Future development opportunities

What features would be nice to add?

Any features that can be optimised/re-designed to provide more functionality or be better performance/more scalable or reliable?

# Conclusion

Critically evaluate the application

Does the application meet the original specification, does it perform all required tasks?

What were the results from testing?

Are all of the original 12 use cases satisfied?

What could be improved or done differently if this project were to be completed again?

# Appendices

# References

**There are no sources in the current document.**