Applied software engineering

Hamza Bhatti (21223241)

Contents

[Introduction 2](#_Toc468206980)

[Requirements From Case Study 2](#_Toc468206981)

[UML Modelling 2](#_Toc468206982)

[Use Case Description 2](#_Toc468206983)

[Class Diagram 2](#_Toc468206984)

[Sequence Diagram 2](#_Toc468206985)

[Architecture Diagram 2](#_Toc468206986)

[Implementation 3](#_Toc468206987)

[Software Testing 4](#_Toc468206988)

[Evaluation 5](#_Toc468206989)

[UML 5](#_Toc468206990)

[Implementation 5](#_Toc468206991)

[Software Testing 5](#_Toc468206992)

[Appendix 6](#_Toc468206993)

# Introduction

# Requirements From Case Study

# UML Modelling

## Use Case Description

Talk about the functionality of the use case diagram – What’s going to happen in the software

## Class Diagram

What are the elements in the class diagram – classes (attributes, operations), associations, pattern and abstraction.

How would you convert the diagram into code.

## Sequence Diagram

What functions where used? – Just list them I guess.

Behaviour of the system – Talk about what happens at each step.

## Architecture Diagram

Talk about the sections of the architecture and how it fits with the other UML.

# Implementation

Talk about how the system meets the use case requirements

Features of the code – interfaces, abstraction how classes interact, the design pattern (why is it there?)

Quick instruction on how to run the program – what is included with the package

# Software Testing

What is the purpose of testing?

Talk about the tests you did - How did they help meet the requirements?

Explain what happens during the tests to get the result.

Give instructions on how to run

# Evaluation

## UML

Why was the UML abstract in certain places – in the use case diagram and class diagrams?

How can the UML change to be better?

## Implementation

How did the code follow the uml – is the code abstract too?

How is the code reusable etc? – Software principles along with maintainability etc

How could the code change to be better?

## Software Testing

How did the tests help?

Why were the tests written the way they were?

How could the tests change to be better?

# Appendix

PUT DIAGRAM AND CODE HERE