|  |
| --- |
| **Project Title Subtitle** |

**[Name(s) of student(s), student number, photographs]**

**Supervisor: [Name of supervisor(s)]**

**[Name and logo of educational institution]**

**[Logo of companies included]**

**[Number of characters]**

**[Study program]**

**[Semester]**

**[Date]**

**Table of content**

[Abstract iii](#_Toc531774235)

[1 Introduction 1](#_Toc531774236)

[2 Requirements 2](#_Toc531774237)

[2.1 Functional Requirements 2](#_Toc531774238)

[2.2 Non-Functional Requirements 3](#_Toc531774239)

[3 Analysis 4](#_Toc531774240)

[4 Design 5](#_Toc531774241)

[5 Implementation 6](#_Toc531774242)

[6 Security 6](#_Toc531774243)

[7 Test 7](#_Toc531774244)

[7.1 Test Specifications 7](#_Toc531774245)

[8 Results and Discussion 8](#_Toc531774246)

[9 Conclusions 9](#_Toc531774247)

[10 Project future 10](#_Toc531774248)

[11 Sources of information 11](#_Toc531774249)

[12 Appendices 1](#_Toc531774250)

**List of figures and tables**

Optional

# Abstract

The following report will present the process of introducing the Zinema – Cinema Management Distributed Software System to the market and the conclusions that have been drawn in the end. The project followed the Unified Software Development Process and SCRUM. At the start, the focus was on setting up why the system should be created. As it advanced, the functional and non-functional requirements were defined and use cases were created accordingly. A Domain Model was used to analyse the problem and understand how different objects would interact with each other. Next, the design of the system was defined using Design Class diagrams based on the existing Domain model. The Design Class diagrams acted as blueprints for the implementation of a 3-tier architecture system. In the last phases of the project, the application was tested based on the Use Cases. Once finalized, the results were subject to discussion and analysis to bring further improvements in the future.

# Introduction

The purpose of the introduction is to provide background information and set the scene for your project. Within which business or organization are you doing the project? Who are the stakeholders and who is the customer?

The background information is adapted from your project description where you have already described the problem domain. Describe the current situation and existing context. Your statements must be supported by references to reliable and relevant sources.

This should lead to why this project is relevant and outline your aim and objectives. Which technical problems and challenges will be presented in this report, again taken from your project description. System illustrations and rich pictures are welcome here.

State delimitations relevant for your project in the introduction. Delimitations include what the project will not cover in relation to your project description, i.e. what could have been expected in your project. Remember that you can only make delimitations to aspects mentioned in the project description and you must argue well for your delimitations.

The last sentences of the introduction should be an overview of the sections to follow. This will be a good transition to the next sections.

Remember: You must ensure a clear connection between sections in the project report, from Project Description, Requirements, Analysis, Design, Implementation to Test. This means that everything that is implemented can be found in design, everything that is designed is based on the analysis, and anything that is found in analysis has a clear link to requirements, etc.

# Requirements

The purpose of the requirement section is to define functional and non-functional requirements. Requirements are perceived as a contract with the stakeholders (customer), and are specified to ensure a common understanding.

Identify the users and describe their roles (e.g. actor descriptions, personas and scenarios).

Note: Remember that all requirements must be precise and testable.

Use the SMART principle (YourCoach n.d.) and MoSCoW (Business Analyst Learnings 2013).

Present a numbered and prioritised list of all the requirements of the users, customer and stakeholders for the project.

## Functional Requirements

1. The administrator should be able to create a flight.
2. The administrator should be able to remove a flight.
3. The administrator should be able to search for flight by flightID.
4. The administrator should be able to see actions performed on the system.
5. The administrator should be able to see all the flights.
6. The system should be able to store a flight’s flightID, origin, destination, date and time of departure and arrival, price, number of tickets.
7. The system should be able to store tickets including its ticketID, seat, price.
8. The system should be able to store a customer including his/hers/its customerID, first name, last name, email, telephoneNumber, passportNumber.
9. The system should be able to store credentials for the customer that include username and password.
10. The user should be able to book a flight and choose the seat for it.
11. The user should be able to cancel a booking for a flight.
12. The user should be able to search flights by origin, destination and by a certain time frame.
13. The user should be able to get a list of cheapest flights and today’s flights.
14. The user should be able to create an account.
15. The user should be able to use the credentials linked to his account to log in and use the system.
16. The user should be able to view his current and past bookings.

## Non-Functional Requirements

1. The system should be coded in JAVA.
2. The system should use a database.
3. The system must be compatible with Microsoft Windows 7,8,10, Mac OS.
4. The system should not contain hard coded values.

# Analysis

The purpose of the analysis section is to outline an understanding of the problem domain and specifically WHAT the stakeholders want. Here, you elaborate on your background description.

You identify objects in the problem domain that will be involved in the solution and how these objects cooperate. The result of this analysis is a Domain Model (Larman 2004, chap.9) and other relevant diagrams.

Use the UML standard for all diagrams where relevant.

Note: Remember that all implementation dependent objects are not part of the domain model only conceptual classes related to the requirements and the domain.

# Design

The purpose of the design section is to outline HOW the system is structured; i.e. to transform the artefacts of the analysis into a model that can be implemented. The design section is relevant for the programmer, whereas the analysis is relevant for the stakeholder.

Elements that may be relevant in this section:

* Architecture: Find architecture patterns here (Leszek Maciaszek 2004, chap.9).
* Technologies: Describe technologies used, also alternative technologies. Argue for choice of technology according to the project aim.
* Design Patterns: Describe which design patterns (GoF (Gamma et al. 2002) etc.) you are using and why.
* Class Diagrams
* Interaction Diagrams
* UI design choices
* Data models, persistence, etc.

You must explain all diagrams in the report. These diagrams including descriptions are the blueprints for the implementation.

Hint: One way to figure out which objects/classes are needed in the design is to apply the General Responsibility Assignment Software Patterns/principles (GRASP) (Larman 2004, chap.17).

Hint: Consider how to design your system to make it testable.

# Implementation

The purpose of the implementation section is to explain interesting code snippets. An idea is to explain the complete path through your system from UI to database etc.

Remember that your implementation must be consistent with your design (Larman 2004, chap.20).

Which standard libraries are used? How are design patterns implemented, etc.

Hint: Implement your code in a testable manner.

# Security

# Test

The purpose of the test section is to document the result of your testing; to verify if the content of the requirements section has been fulfilled. How is the system tested, which strategy has been used; e.g. White Box (Unit Test), Black Box, etc.

## Test Specifications

For functional requirements, test specifications must be listed. These test specifications can be described as soon as the functional requirements have been completed (Use Cases including descriptions).

IEEE can be used as a template for test specification (IEEE Computer Society 2008). VIA Library can give you access to this standard.

# Results and Discussion

The purpose of the results and discussion section is to present the outcome and achieved results of the project.

# Conclusions

The purpose of the conclusion section is to compile the results from each section in the report. What is the conclusion? Did the project fulfil the requirements? Etc.

You can only comment on report contents, no new topics or content can be introduced in this section.

# Project future

Reflect on your project from a technical viewpoint and describe what you would change if you could.

Suggest how the project could be improved or made ready for production. Discuss scalability, suggest possible spin offs, what is needed, missing, etc.?

# Sources of information

**Note: Use the standard reference method: Harvard Anglia. A very good reference tool is Mendeley** (Mendeley.com 2016), **ask VIA Library if you need help.**

Banger, D., 2014. A Basic Non-Functional Requirements Checklist « Thoughts from the Systems front line.... Available at: https://dalbanger.wordpress.com/2014/01/08/a-basic-non-functional-requirements-checklist/ [Accessed January 31, 2017].

Business Analyst Learnings, 2013. MoSCoW : Requirements Prioritization Technique — Business Analyst Learnings. , pp.1–5. Available at: https://businessanalystlearnings.com/ba-techniques/2013/3/5/moscow-technique-requirements-prioritization [Accessed January 31, 2017].

Dawson, C.W., 2009. *Projects in Computing and Information Systems*, Available at: http://www.sentimentaltoday.net/National\_Academy\_Press/0321263553.Addison.Wesley.Publishing.Company.Projects.in.Computing.and.Information.Systems.A.Students.Guide.Jun.2005.pdf.

Gamma, E. et al., 2002. *Design Patterns – Elements of Reusable Object-Oriented Software*, Available at: http://books.google.com/books?id=JPOaP7cyk6wC&pg=PA78&dq=intitle:Design+Patterns+Elements+of+Reusable+Object+Oriented+Software&hl=&cd=3&source=gbs\_api%5Cnpapers2://publication/uuid/944613AA-7124-44A4-B86F-C7B2123344F3.

IEEE Computer Society, 2008. *IEEE Std 829-2008, IEEE Standard for Software and System Test Documentation*,

Larman, C., 2004. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*,

Mendeley.com, 2016. Homepage | Mendeley. Available at: https://www.mendeley.com/ [Accessed February 2, 2017].

YourCoach, S.M.A.R.T. goal setting | SMART | Coaching tools | YourCoach Gent. Available at: http://www.yourcoach.be/en/coaching-tools/smart-goal-setting.php [Accessed August 19, 2017].

# Appendices

The purpose of your appendices is to provide extra information to the expert reader. List the appendices in order of mention.

Examples of appendices

* Project Description
* User Guide
* Source code – source documentation
* Diagrams
* Data sheets
* Etc.

**Appendix A Project Description**

Insert the original Project Description here