|  |
| --- |
| **Exam Scheduling System VIA University College** |

**Eva Nikolaeva, 293164**

**Marton Pentek, 293649**

**Miladin Asenov, 240054**

**Sabina Elena Baghiu, 293119**

**Supervisor: Steffen Vissing Andersen**

**Mona Wendel Andersen**

**Via University College**

**[Number of characters]**

**Software Technologies Engineering**

**1st Semester**

**[Date]**

**Table of content**

[Abstract iii](#_Toc18659739)

[1 Introduction 1](#_Toc18659740)

[2 Analysis 2](#_Toc18659741)

[2.1 Requirements 2](#_Toc18659742)

[2.2 Functional Requirements 2](#_Toc18659743)

[2.3 Non-Functional Requirements 3](#_Toc18659744)

[3 Design 4](#_Toc18659745)

[4 Implementation 5](#_Toc18659746)

[5 Test 6](#_Toc18659747)

[5.1 Test Specifications 6](#_Toc18659748)

[6 Results and Discussion 7](#_Toc18659749)

[7 Conclusions 8](#_Toc18659750)

[8 Project future 9](#_Toc18659751)

[9 Sources of information 10](#_Toc18659752)

[10 Appendices 1](#_Toc18659753)

**List of figures and tables**

Optional

# Abstract

*This report serves the purpose of providing a detailed overview through the development stages of a Single User Exam Booking System, developed in Java. The system was created for VIA University College, that used excel sheets to book exams, but this was causing too many problems, due to human errors. The solution to the problem was found by analysing the requirements given by the university. Creating use case descriptions for the use cases, activity diagram, sequence diagram as well as a class diagram.*

*Based on the requirements the important classes were implemented with necessary functionality and tested successfully against the use cases. A functional GUI was designed with an objective of making the daily tasks of the user less complicated. The use cases were again tested against the program, successfully. The result is a functional program that fulfils all the requirements, making the everyday work of the user easier.*

# Introduction

We need to create a program for VIA University College that will correctly be able to show available dates for exams, choose a room, date and time for a given exam, implement the data needed ( student data, examiner data, available dates ) The program will be used by the secretary, so she needs to be able to get implement data, and if needed change anything implemented by the user at any given time. Once an exam is booked, there will be a calendar showing all the exams booked and all rooms that will be still available, then save all the data for further use. The program’s design should give the user a simpler way of booking an exam, see the exam period start and end of it and allow them to search for already booked exams.  
  
The secretary in VIA University College needs a single user booking system for the hotel. The lack of an exam booking system causes problems for choosing correct rooms and dates. Currently the university is using excel sheets where they write all exams that need to be booked. Using excel sheets is also very time-consuming since they need to go through it every morning to find the available rooms and figure out the equipment needed for a given exam. The needed booking system should be a single user system, since the program will be used only by the secretary. The program’s design should give the user a simpler way of booking an exam, see the exam period start and end of it and allow them to search for already booked exams.

Furthermore, the program should allow the user to change and delete booked exams, as well as changing the room needed. In addition, the program should minimize the number of mistakes that the university is currently struggling with. The following pages explains in depth how the program has been developed.

# 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.

## 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

Functional requirements could be described with Use Cases, Use Case descriptions and Actor descriptions. Use Case descriptions can be detailed with different types of UML diagrams.

## Non-Functional Requirements

There are no standards for describing non-functional requirements. You can find a useful checklist here (Banger 2014). For content see Appendix 3 “Project Report – VIA Engineering Guidelines”.

# 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.

# 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