1.0.0

|  |
| --- |
| **PROJECT REPORT ICT Engineering - 2017**  **VIA University College** |

**Version history**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Description** | **Version** | **Init** | **Revised** | **Approved** | **Date** |
| Start of document | 0.1.0 | IHA | <INIT> | <INIT> | 2017-01-31 |
| Small corrections | 0.2.0 | IHA | MWA |  | 2017-02-01 |
| Revisions made in workshop meeting | 1.0.0 | MWA/IHA | All | ICT | 2017-08-19 |

**Group: XXXX**

|  |  |
| --- | --- |
| **Student Name** | **Student No** |
|  |  |
|  |  |
|  |  |

# Contents

[1 Introduction 2](#_Toc490902148)

[2 Requirements 2](#_Toc490902149)

[2.1 Functional Requirements 2](#_Toc490902150)

[2.2 Non-Functional Requirements 2](#_Toc490902151)

[3 Analysis 2](#_Toc490902152)

[4 Design 2](#_Toc490902153)

[5 Implementation 2](#_Toc490902154)

[6 Test 2](#_Toc490902155)

[6.1 Test Specifications 2](#_Toc490902156)

[7 Results and Discussion 2](#_Toc490902157)

[8 Conclusion 2](#_Toc490902158)

[9 Project future 2](#_Toc490902159)

[10 References 2](#_Toc490902160)

[11 List of Appendixes 2](#_Toc490902161)

[Appendix A Project Description 2](#_Toc490902162)

**Abstract**

*An abstract is a shortened version of the report and should contain all information necessary for the reader to determine:*

1. *What are the aim and objectives of the project*
2. *What are the main technical choices*
3. *What are the results*

*Frequently, readers of a report will only read the abstract, choosing to read at length those reports that are most interesting to them. For this reason, and because abstracts are frequently made available to engineers by various computer abstracting services, this section should be written carefully and succinctly to have the greatest impact in as few words as possible.*

*Although it appears as the first section in a paper, most report writers write the abstract section last.*

Cf. (Dawson 2009, p.195).

# 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

## Functional Requirements

1. The user should be able to create a new event
2. The events can be of four different types (lectures, seminars, workshops and trips)
3. The system should store the following information about the lecturers: a title, start date, start time, duration, lecturer, 1 subject, sponsor name, sponsored or not, number of available tickets, price, finalized or not, total number of tickets, discount
4. The system should store the following information about the seminars: a title, start date, start time, duration, lecturers, subjects, sponsor name, number of available tickets, price, finalized or not, total number of tickets, discount
5. The system should store the following information about the workshops: a title, start date, start time, duration, lecturers, food included (breakfast, lunch, dinner? – vegan or not), number of available tickets, price, finalized or not, total number of tickets, discount
6. The system should store the following information about the trips: a title, start date, start time, duration, locations, available tickets, price, finalized or not, total number of tickets, discount
7. The user should be able to search events by: finalized or not, time period, subject, price, available tickets, lecturers, sponsors
8. The user should be able to modify every aspect of non-finalized events
9. The user should be able to store members’ information
10. Members are defined by name, email, address, phone, payment year, date of registration, newsletter subscription, attended events
11. The user should be able to search members by name, payment year, date of registration, attended events
12. The user should be able to update the information of each member
13. The user should be able to store lecturers’ information
14. Lecturers are defined by name, email, phone, sponsor or not, subject
15. The user should be able to search lecturers by name, subject
16. The user should be able to update the information of each lecturer

## Non-Functional Requirements

1. The system has to be implemented in Java
2. The system has to be compatible with Microsoft Windows 7,8,10
3. The system needs to be able to run indefinitely
4. The system should use files for secondary storage only

# The system should use files for secondary storage only 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.

# 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.

# Conclusion

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.?

# References

**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].

# List of Appendixes

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.

1. Project Description

Insert the original Project Description in here.