1.0.0

|  |
| --- |
| **Vipasanna Management System**  **Subtitle** |

**Group: 6**

|  |  |
| --- | --- |
| **Student Name** | **Student No** |
| Daniela Koch | 266502 |
| Matej Michalek | 266827 |
| Michaela Golhova | 266099 |
| Michal Karol Pompa | 266494 |
| Remedios Pastor Molines | 266100 |

# 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

Nowadays due to the fact that people are living in a rush and that the professional life has taken advantage on the other aspects of life they are often exposed on a huge amount of stress (WHO, 2017). Together with the popular unhealthy lifestyle- both physically (the sedentary lifestyle) and mentally (continuous stress)- it often causes depression. Globally, more than 300 million people suffer from this disease. That is one of the biggest issues of the 21st century (WHO, 2017).

One way of solving the issue of spreading of depression is provided by Vipassanā ‐ Insight Awareness (VIA). VIA is a center for spiritual events originally with a base in the Buddhist principles of meditation as an insight with awareness of what is happening as it happens. Today events at  VIA also include spiritual practices not directly linked to any religion. Such practices, to take an example meditating, can reduce stress (Psychiatry online, 2006).  Furthermore taking part in lectures, seminars or workshops which are provided by VIA keeps one’s mind away from work and lets it rest and relax. What helps people even more are the trips promoted by the organization. Not only do they allow one to forget about the everyday routine, but also force one to move, breathe fresh air and spend time close to the nature.

However Vipassanā ‐ Insight Awareness is not adapted to the today’s world based on new technology. As the world depends more and more on the work of computers, keeping a paper-based system is hardly possible. The organization is in a need of keeping track of events, members, lecturers and the work done by its council in an easy way. It needs a system that would store data and provide both an easy access to them (to take an example to simplify sending emails to the members) and an easy way of sharing them (to ease promoting organized events). Due to the fact that today the internet is one of the most popular sources of information (Taylor & Francis, 2017), a computerized system would increase the range of attendance on the events. Moreover it would solve the complication of searching for proper lecturers. Even though similar systems already exist they do not meet the needs of our client. That is due to the fact that VIA expects a simple server and the existing ones are complicated.

# Requirements

Bearing in mind the needs of VIA presented in the introduction, the requirements concerning the needed system are as follows:

## 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.

1. The system has to search for lecturers of a given category
2. The administrator must be able planning events on specific date, about specific topic, and with specific lecturer.
3. The system must count the price for members, having considered the discount for events depending on the event type.
4. The system must store information about members (name, address, phone, email, date for membership and the payment year).
5. The system must collect finalized events for newsletter.
6. The system must collect list of lecturers who wants a fee or advertisement (and sponsors) for newsletter.
7. Members must be informed about the payment via e-mail by the system.
8. The system must allow the administrator to search for not finalized events.
9. The administrator must be able to sign up a new member.
10. The administrator must be able to sign up a new lecturer.
11. The administrator must be able to sign up participants and members to specific events.
12. The system must be able to return the amount of available places for every event.

## Non-Functional Requirements

There are no standards for describing non-functional requirements. You can find a useful checklist here (Banger 2014).

1. The system must be implemented in Java
2. Secondary storage needs to be done with files
3. The parts of the system must be accessible among them (feedback with events/sponsors/event calendar).

# 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.