1.0.0

|  |
| --- |
| **Vipassanā ‐ Insight Awareness (VIA) Event Management System** |

**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** |
| Dragos Chirtoaca | 253742 |
| Pascari Liviu | 266094 |
| Oskars Arajs | 266534 |
| Taha Mohamed Alzein | 269055 |

# 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**

*The main objective of the project is to develop an event handler(event management) system for Vipassanā ‐ Insight Awareness (VIA), in order to help the company to solve their current problems related to management of the events and their members. This system should be capable to make the work easier for the event manager of their company. For the event manager it is important to see the status of their events, is this an event in the past or in the future, to create an event, to edit, or to delete. On the other hand the event manager need to be able to manage the participants of the events such as members and non-members. For the members event manager should be able to add a member, to see the status of members membership, and to be able to send a reminder outside of the system to this members that has to pay, so for this it is necessary to identify other contact information of this members. It is very important to handle a list of contacts, is it a member or non-member, in order to send manually newsletter about current events based on the preferences of the event participants.*

*In order to satisfy the customer needs and to develop an end use product, it is needed to divide the project into objectives that should be reached, that are:*

* *based on interview we identified the main requirements of the customer;*
* *design the diagrams that are describing the main problems that the system should solve;*
* *create the interface of the system that will make the work of event manager easier;*
* *develop the system functionality using Java;*

*All the information that is related to the events, members, non-members, sponsors and lecturers, is stored using binary files, on the computer on which is running.*

*As a result, the project has an event handling system related to members, non-members, sponsors and lecturers management, which meets-up the customer’s requirements. This system is easy to modify and add to it more features if this will appear in customer’s needs.*

# Introduction

For any organisation that is organising events it is very hard to manage the events, especially if there are many and especially if you are non-profit organisation. Nowadays people are very opened in studying new things, and as a result of increasing the number of customers that are interested in Buddhism, meditation, alternative self healing, there is very hard to keep track of all the events. For keeping the track of events, members, lecturers, sponsors, participants needs there is a need for a very good event manager, and there is very hard if this event manager decides to leave the company, but by using a system that is holding all this information in digital format and is making creation of an event with almost just a click, it will save not just the time, but it will make the company more secure and safe in the event handling.

In the traditional way of handling events, there were a pencil and a piece of paper (notebook) and the event manager has to search through all of that paper in order to find the information of any specific event. In our fast developing word and in digitalized era this is possible by writing everything in excel, access or other software that are helping a little bit the user. This is still not enough because this does not fully satisfy our needs and the users have to learn how to use this software that are doing too many things that we don’t need. That’s why as a professionals companies need to create something more user-friendly and more suitable for their needs.

VIA is a centre for spiritual events originally with a base in the Buddhist principles of meditation as an insight with awareness of what happens when it happens. Today events at VIA also include spiritual practices not directly related to any religion like dream interpretations, healing, astrology, reincarnation, karma, alternative health care and similar events. VIA helps organize these events and journeys for people with interest in different categories starting from exploring one’s mind and learning how to deal with inner conflicts. Up until now, they used pen and paper to keep track of their lectures, lecturers and upcoming events. But that system is only viable if the number of lectures and events are low. Otherwise, the system gets overwhelmed and it gets hard to keep track of all of the data and needed resources to keep the organization going. That’s why they decided to implement a software in their company that will make their work easier and it will help them to not disappoint the current customers.

The developed event handling software system allows the event manager to: create, edit, delete an event, by entering the category, type, name, date, time, duration, lecturer and price, on the other hand it will help to keep track of events, is this a future event, or this is an event in the past. The event manager will be able to add, edit, delete a member, by entering name, address, phone number, email and preferences of the member, on the other hand the event manager will be able to see the members and non-members in a specific category, and to see which members has to pay their membership by choosing the interested date. Every time when the event manager is adding a member or a non-member the system is generating automatically an ID for each of them, that is starting with “M” if it is a member and with “N” if it is non-member. The event manager will be able to add needed member or non-member to the needed event only by finding the specific member or non-member by their ID, or to create new ones if they are not in their lists. The system is not able to calculate the total income of an event and it is not able to send automatically newsletters to the potential participants.

The system purpose is to help non-profit organization with their hard lifetime in managing events, and on the other hand to manage their sponsors, lecturers, and of course members.

**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

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).

## Use case diagram

The following diagram shows what the event manager must be able to do, in order to satisfy the client needs based on the interview. Each use case that you can see in the diagram is an action that can be performed by the event manager.

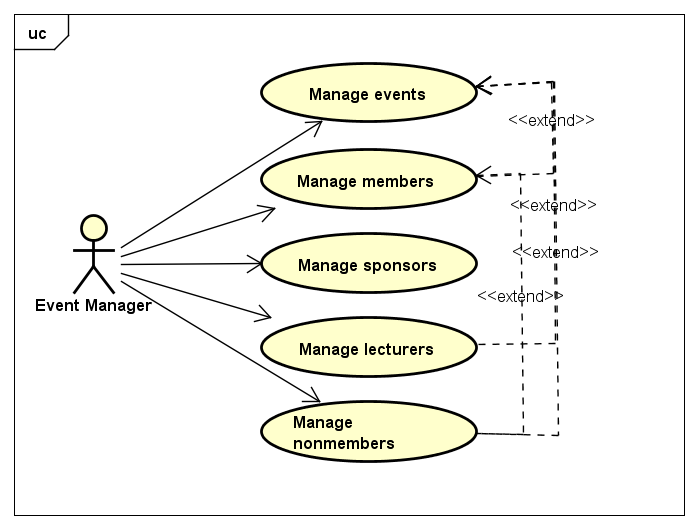


Figure 1, Use case diagram

In the figure above, we have a short version of the use case diagram, that is showing what the user will be able to do by using the developed system.

See the full version of the use case diagram in the Appendix A

## Use case diagram description

## Class diagram

A class diagram models the static structure of a system. It shows relationships between classes, objects, attributes, and operations. Classes represent an abstraction of entities with common characteristics. Associations represent the relationships between classes.[[1]](#footnote-2)

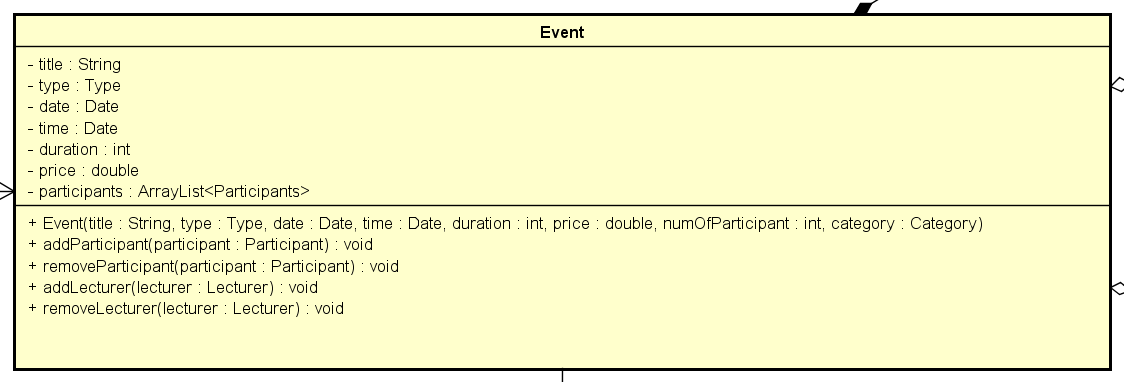


Figure 2, Event class diagram

The figure above is an example of UML class diagram for the Event class.

See full version of class diagram in Appendix C

# 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.

1. SmartDraw, LLC, definition of class diagram, <https://www.smartdraw.com/class-diagram/>, visited 18/12/2017 [↑](#footnote-ref-2)