|  |
| --- |
| **Project Report ViaFit** |



Mihail Alexandru Teodosiu

299111

Michal Pupák, 299128

Tamás Péter, 299124

Supervisors:Mona Wendel Andersen, Allan Rune Henriksen

**VIA University College**



**[Number of characters]**

**Software Engineering**

**Semester 1**

**[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

*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

ViaFit is a small fitness centre in Denmark. Its owner is preparing for retirement and leaves the daily management to his son. The owner, who was running the fitness centre for many years, did not bother to keep up with the new technology and everything was done in an old-fashioned way.

Until now, all the relevant information about the members, instructors and classes is stored on the paper. The schedule is drawn on a whiteboard inside of the fitness centre at the start of every month and is the only source of information about the upcoming classes.

There are many practical flaws and difficulties associated with this old-fashioned way of running the fitness center. It is a small fitness center with not too many members, but as the costumer mentioned during the interview, he wants to expand the business and attract a lot of new members. If that is the case, it will become much more difficult to store and manage all the information on the paper. (Mr. Sixpack, Interview)

Accidents such as kids drawing on the whiteboard with the schedule must also be taken into consideration. Because it is the only place where the schedule exists, it had to be redrawn from the owner’s memory. The “Candle incident of 1997” must also be mentioned. The fitness center lost information about a half of their members during this accident. Both problems could have been prevented if the relevant information were stored in a digital form.

To avoid all these unnecessary problems, the owner’s son wishes to move the fitness centre into the digital age and equip his business with a new management software system. He requires a GUI (graphical using interface) application, operated by a receptionist at the front desk. It must be able to manage all the information of their members, instructors, and classes. He specifically asked for no password login.

The purpose of this project is to design and develop a management software, which fulfils costumer’s needs and makes the business run more efficiently and with less risks involved.

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, 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.

# 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.

**High priority**

1. As a receptionist, I want to register the name, address, phone number, email, and membership status of all users, so that it is more secure and easy via search by functions.
2. As a receptionist, I want to delete members, manage their information, and manage their membership status. So that it is easy to change a user’s status from normal to premium and vice versa.
3. As a receptionist, I want to register the instructor’s name, address, phone number, email and the classes taught by all instructors, so that it is easier and faster accessible via search by functions.
4. As a receptionist, I want to delete instructors, manage their information, and manage the classes that they teach so that I can easily manage my employees' information.
5. As a receptionist, I want to register new classes to the system, an instructor, a date, and a time slot, so that I can easily add new classes if members become interested.
6. As a receptionist, I want to modify the date, time, and instructor of classes, so that users easily see what is on the schedule and our instructors know when to show. One of the modifying options will be to delete a class from the system.
7. As a receptionist, I would like to be able to add and remove classes from the system. If a class does not have enough participants, I will remove it and if people have new interests, I will add this class.
8. As a receptionist, I want to make a schedule which will be displayed on the VIAFit website so that my clients can access the class schedule online, which is more secure and prevents problems like data-loss.
9. As a receptionist, I want to be able to access a list of all registered members, so I can tell who is registered as a member of the fitness center.
10. As a receptionist, I want to be able to access a list of all registered instructors, so I can find the desired instructor easier and manage his or her information if needed.
11. As a receptionist, I want to be able to access a list of all registered classes, so I can easier access the targeted class and manage its data

**Low priority**

1. As a receptionist, I want to set more than one instructor for a class, so that I may have more than one instructor teaching a specific class.
2. As a receptionist, I want to set a minimum number of members in a class, as I do not want classes with too few participants.
3. As a receptionist, I would like to offer the possibility of free trials for members, so I can boost our popularity. This can only be done once.
4. As a receptionist, I want to find a specific instructor, member, class easily using a search bar, so that I can easily view and manage data.

**Non-Functional Requirements**

1. The system must be implemented in Java.
2. The system should have buttons and text fields for the user to interact with, operated with keyboard and mouse.
3. The system must have fields to enter data, buttons to save and search for data.

## 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”.

## Use Case Diagram

## 

**Use Case Descriptions**

|  |  |
| --- | --- |
| **Use case** | **Register Member** |
| **Summary** | Register a new member with his/her information (name, e-mail, address, phone number and member status) into the system. |
| **Actor** | Receptionist |
| **Precondition** | None |
| **Postcondition** | New member has been created and stored in the system with information provided by the receptionist in step 3. |
| **Base sequence** | 1. Choose register member 2. The system displays a form with empty text fields. 3. Enter the new members information (name, email, address, phone number and membership status) 4. Save the new member information. New member will be registered in the system. |
| **Exception**  **sequence** | If the user does not fill all text fields, the system will not register a new member and will not proceed. The receptionist must return to step 3. |
| **Note** | A member can be created at any time.  The process can be cancelled at any time. |

|  |  |
| --- | --- |
| **Use case** | **Register Instructor** |
| **Summary** | Register a new instructor with his/her information (name, e-mail, address, phone number and class) |
| **Actor** | Receptionist |
| **Precondition** | None. |
| **Postcondition** | New instructor has been created and stored in the system with information provided by the receptionist in step 3. |
| **Base sequence** | 1. Choose register instructor 2. The system displays a form with empty text fields to be filled 3. Enter person’s information (name, e-mail, address, phone number and class) 4. Save the new instructor information. New instructor is created and registered in the system. |
| **Exception**  **sequence** | If the receptionist does not fill all text fields, the system will not create a new instructor and will not proceed. The user must return to step 3. |
| **Note** | An instructor can be created at any time.  The process can be cancelled at any time. |

|  |  |
| --- | --- |
| **Use case** | **Register Class** |
| **Summary** | Register a new class with it’s basic information (short description, instructor’s name, maximum number of participants, date and time) provided by the receptionist in step 4. |
| **Actor** | Receptionist |
| **Precondition** | There must be at least one instructor available. |
| **Postcondition** | New class has been registered and stored in the system with given information |
| **Base sequence** | 1. Choose classes 2. The system displays a list of all registered classes alphabetically. 3. Choose a new class. 4. The system displays a form with all information the actor must provide in order to successfully register a new class (name of the class, short description, name of the instructor, maximum number of participants, date and time slot) 5. Fill all text fields with relevant information 6. Save and confirm information. New class with provided information will be registered in the system. |
| **Exception**  **sequence** | If the receptionist does not fill all text fields, the system will not register a new class and will not proceed. The receptionist must return to step 5. |
| **Note** | Class can be registered at any time.  The process can be cancelled at any time. |

|  |  |
| --- | --- |
| **Use case** | **Manage Member** |
| **Summary** | Modify the name, phone, email or membership type, and remove member from the system. |
| **Actor** | Receptionist |
| **Precondition** | A member has already been created and stored in the system. |
| **Postcondition** | The member is either deleted or is stored in the system with the new information provided by the receptionist. |
| **Base sequence** | 1. Choose members 2. The system displays an alphabetical list of all members. 3. Search for the member’s name in the search bar. 4. The system displays all members with the searched name. 5. Choose desired member 6. Choose to modify member 7. The system displays a form with all information about the member. 8. Modify the desired fields or delete a member. 9. Save and confirm information. |
| **Exception sequence** | If any text fields have been left empty, the system will not save the changes and will not proceed. The receptionist must return to step 8. |
| **Note** | Member can be modified at any time.  The process can be cancelled at any time. |

|  |  |
| --- | --- |
| **Use case** | **Manage Instructor** |
| **Summary** | Modify the name, phone, email or the classes taught, and remove instructor from the system. |
| **Actor** | Receptionist |
| **Precondition** | An instructor has already been created and stored in the system |
| **Postcondition** | The instructor is either deleted or stored in the system with the new information provided by the receptionist in step 9. |
| **Base sequence** | 1. Choose Instructors 2. The system displays a list of all instructors in an alphabetical order. 3. Choose the Search Bar and type the name of an instructor, or search for the targeted instructor on the list. 4. The system displays all instructors with the same name as provided in the search bar. 5. Choose desired instructor 6. Choose to modify instructor 7. The system displays a form filled with all the information about the instructor 8. Modify the desired fields or cancel 9. Save and confirm information. |
| **Exception**  **sequence** | If any text fields have been left empty, the system will not save the changes and will not proceed. The receptionist must return to step 8. |
| **Note** | Instructor can be modified at any time.  The process can be cancelled at any time. |

|  |  |
| --- | --- |
| **Use case** | **Manage Class** |
| **Summary** | Changing information about already registered class, and removing a class from the system |
| **Actor** | Receptionist |
| **Precondition** | Class must be already registered |
| **Postcondition** | The class is either deleted from the system or is stored in the system with new information provided by the receptionist in step 6. |
| **Base sequence** | 1. Choose classes 2. The system displays a list of all registered classes ordered      alphabetically 3. Look for the targeted class 4. Choose to modify class 5. The system will display a form with text fields filled with currently stored information 6. Change any information by typing in to the corresponding text field 7. Save and confirm information. |
| **Exception**  **sequence** | If any of the text fields are left empty, the system will not save the changes and will not proceed. The receptionist must return to step 6. |
| **Note** | The process can be cancelled at any time. |

|  |  |
| --- | --- |
| **Use case** | **Search Data** |
| **Summary** | Searching for a specific member, instructor or class. |
| **Actor** | Receptionist |
| **Precondition** | The receptionist is inside one of the member, instructor or class folder |
| **Postcondition** | The data matching the search will be displayed on the screen. |
| **Base sequence** | 1. Choose the search bar 2. Type keywords 3. The system displays all relevant information |
| **Exception**  **sequence** | If none of the data matches the search, display an error message explaining that there are no search results and prompt the receptionist to try again. |
| **Note** | You can easily find a specific member, instructor or class |

# 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