|  |
| --- |
| **Final Exam Planner** |

**Jan Lishak – 294322**

**Lenka Orincakova - 293085**

**Juan Iglesias Trebolle – 293143**

**João Bernardo Baptista Vieira Dias – 293133**

**Supervisors: Astrid Hanghøj, Michael Viuff**

**[Name and logo of educational institution]**

**[Logo of companies included]**

**[Number of characters]**

**Software Technology Engineering**

**1st semester**

**16.12.2019**

**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 juan

*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- juan

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 lenka

appendices

* User Guide
* Source code – source documentation
* Diagrams
* Data sheets

(Summary of analysis)

VIA University College needs to plan exam schedule before the examination starts. The schedule plan must meet many specific criteria. Scheduling in general is simple task but when there are so many conditions for the right plan, the task can become very complex. Computers might possibly help to speed up the process of scheduling.

Our system is able to do: 1) the administrator collects the information, 2) the administrator is uploading the information and system checks the requirements, 3) the schedule is created.

The system contains information about students, co-examiners and examiners. The schedule must be flexible and delivered to the students in a way that the information is always updated and easy to access. It would be better if the schedule could be viewed by a student or a teacher in a way that it filters all irrelevant information.

*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).*

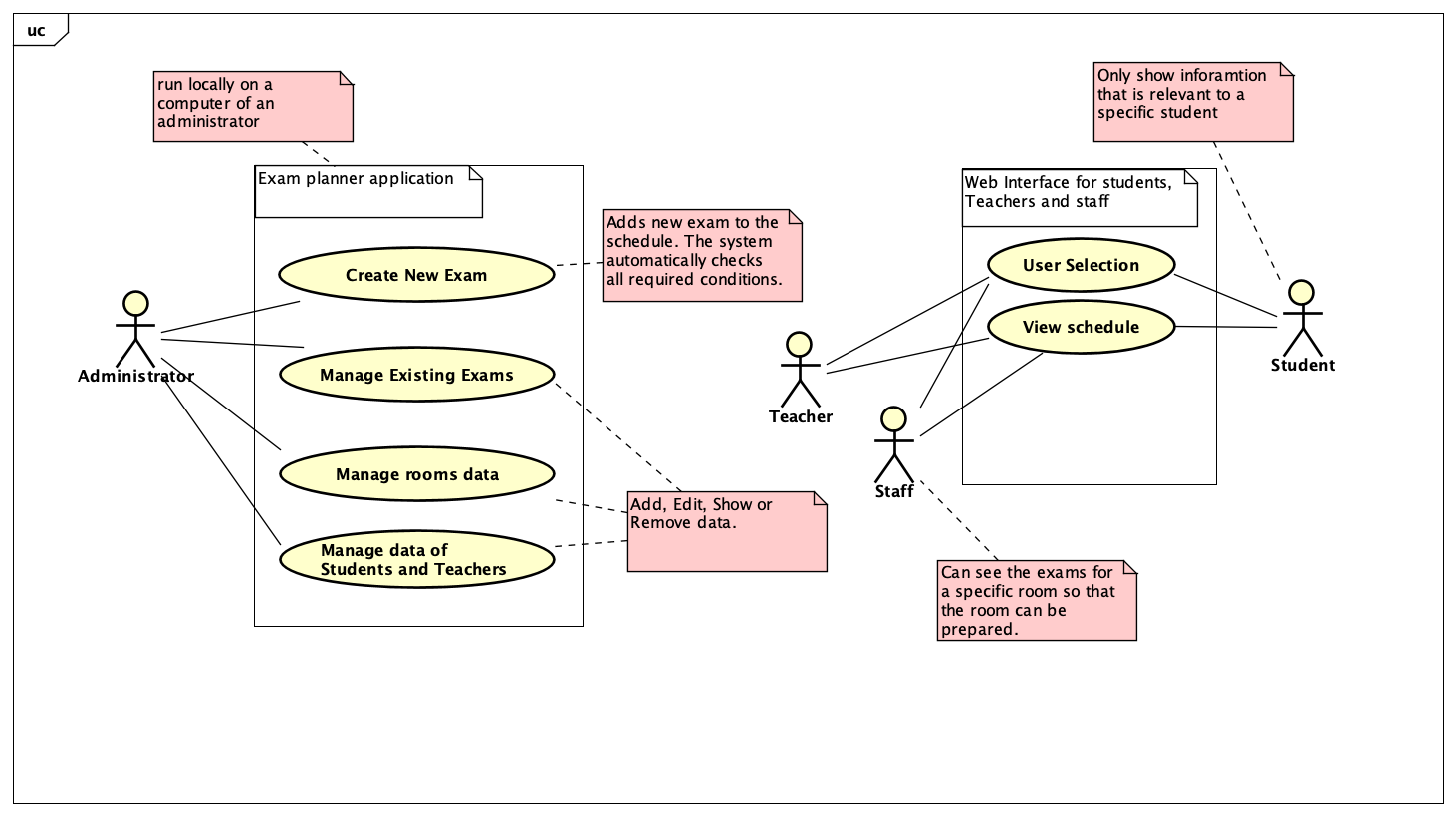
## Functional Requirements

1. As an administrator, I want to be registered as only administrator with access to all information.
2. As an administrator, I want to be able to upload information about teachers, students and classrooms.
3. As an administrator, I want the system to check if there are two exams in the same day, with the right amount of time between them.
4. As an administrator, I want the system to check if the rooms are suitable for exams and if the examiners and co-examiners are available.
5. As an administrator, I want the system to check that the examiners’ and co-examiners’ workday is 8 hours max.
6. As an administrator, I want the system to check if the written exam is before oral exam.
7. As an administrator, I want the system to check if the time of the written exam is 3 hours and the time of the oral exam is 20 minutes.
8. As an administrator, I want the system to check if the exam on 7th semester is at least 3 days before graduation.
9. As an administrator, I want the system to check if the class for the exam is the same as class used during the semester.
10. As an administrator, I want the system to check if the service department is informed about the exam at least one day in advance to prepare the room for exam.
11. As an administrator, I want to be able to add, remove, edit and show information about exam.
12. As an administrator, I want to be able to add, remove, edit and show information about student.
13. As a student, I want to be registered as student with access to the schedule.
14. As a student, I want to be able to see the schedule.
15. As a teacher, I want to be registered as a teacher with access to the schedule.
16. As a teacher, I want to be able to see the schedule.

## Non-Functional Requirements

1. Every update in the system includes writing to a file.

## Use case Diagram



## Use case descriptions

|  |  |
| --- | --- |
| **Use case** | **Manage room data** |
| **Summary** | Add a new classroom, show the list of all classrooms, edit classroom’s properties or delete a classroom. |
| **Actor** | Administrator. |
| **Precondition** |  |
| **P Postcondition** | A new classroom has been added to the list of classrooms, data for an existing classroom has been shown or updated or an existing classroom has been deleted from the list. |
| **Base sequence** | ADD:   1. If a new classroom needs to be added specify the values for    1. Name of the classroom    2. The capacity (numbers of places to sit for students)    3. Subject taught in this classroom    4. Capabilities for projection (HDMI, VGA)   Show:   1. Show a list of all classrooms recorded in the system. For each classroom show the defined values as in step number 1.   Edit:   1. The system shows all classrooms as in step number 2. 2. Select which of the classroom properties defined in step number 1 are required to change 3. Update the property with a new valid value   Remove:   1. Show a list of all classrooms recorded in the system. For each classroom show the defined values as in step number 2. 2. Select which of the existing classroom needs to be deleted. 3. Remove the specified classroom from the list of classrooms. |

|  |  |
| --- | --- |
| **Use case** | **Create new exam.** |
| **Summary** | Creates a new exam in the schedule specifying all the required information. |
| **Actor** | Administrator |
| **Precondition** | The Administrator has to be logged in and the student’s, room’s and teacher’s data must be already set up. |
| **Postcondition** | Exam is created with all necessary information. |
| **Base sequence** | 1. The system shows every program which is taught at VIA UC. For each program, teachers, students, rooms are set. Choose the program for exam. 2. Each program is divided in several semesters. Select the semester you want to have an exam in. 3. Every course from the semester is shown. Choose the course (SDJ, RWD, MSE, etc.). 4. System filters the classes. Choose the class for exam (X, Y, Z, etc.). 5. Choose the type of exam. The system will show the classroom which is assigned to specific class (E 301). If the exam is of type written and is before oral exam, list all the rooms which minimum size is 40 and list all the features of the classroom (HDMI, etc.). 6. If the specific classroom is not available, go to step 5 and choose another one. 7. Enter the date of exam. 8. If the date is not available go to step 7. 9. Approve the selected exam or go to step 5 to select another. 10. System shows a list of examiners. Choose the examiner. 11. If the examiner has reached maximum hours (8 hours) and is not available on the specific date, go to step 10 to choose another one. 12. Approve the selected examiner or go to step 10 to select another. 13. System shows a list of co-examiners. Choose the co-examiner. 14. If the co-examiner has reached maximum hours (8 hours) and is not available on the specific date, go to step 13 to choose another one. 15. Approve the selected co-examiner or go to step 13 to select another. 16. The exam is uploaded with assigned date, examiner, co-examiner, time, room and list of the students. 17. If the exam is oral, the order of students is uploaded. |

|  |  |
| --- | --- |
| **Use case** | **Manage the data of students** |
| **Summary** | Add, edit or remove users of the system. |
| **Actor** | Administrator |
| **Precondition** | The classrooms must be defined. |
| **Postcondition** | Information about students is set. |
| **Base sequence** | Show:   1. show a list of all students recorded in the system. 2. For each student show the first name, last name, ID, subject.   ADD:   1. If adding new student, then set up this information:    1. First name    2. Last name    3. ID    4. subject 2. System validates data and prompts for illegal values 3. If the values are valid, the system adds a new student with given information.   Edit:   1. System shows a list of all students and all information as defined from step 1. to step 2. 2. Enter or edit values for one of the fields of the student. 3. System updates the selected field of the student.   Remove:   1. System shows a list of all users and all information as defined from step 1. to step 2. 2. Select a specific student from the list to be deleted 3. System removes the student from the list. |

|  |  |
| --- | --- |
| **Use case** | **Manage the data of teachers.** |
| **Summary** | Add, edit or remove users of the system. |
| **Actor** | Administrator |
| **Precondition** | The classrooms must be defined. |
| **Postcondition** | Information about teachers is set. |
| **Base sequence** | Show:   1. show a list of teachers recorded in the system. 2. For each teacher show the name, ID, teaching subject, classes they teach and email.   ADD:   1. If adding new teacher, then set up this information:    1. First name    2. Last name    3. ID    4. subjects 2. System validates data and prompts for illegal values 3. If the values are valid, the system adds a new teacher with given information.   Edit:   1. System shows a list of all users and all information as defined from step 1. to step 2. 2. Enter or edit values for one of the fields of the teacher. 3. System updates the selected field of the teacher.   Remove:   1. System shows a list of all teachers and all information as defined from step 1. to step 2. 2. Select a specific teacher from the list to be deleted 3. System removes the teacher from the list. |

|  |  |
| --- | --- |
| **Use case** | **Manage exam data** |
| **Summary** | Add, edit, remove or view exam data. |
| **Actor** | Administrator |
| **Precondition** | The student’s, room’s and teacher’s data must be already set up. |
| **Postcondition** | The data in the schedule is replaced with the one specified. |
| **Base sequence** | ADD:   1. If adding new exam, then set up this information:    1. Subject    2. Room    3. Date    4. Teacher    5. Type of exam    6. Format    7. ECTS    8. Examiners 2. If the date and time steps over existing date and time, go to step 1 and choose another date and time. 3. If the semester is 7th and the date is not three days before graduation, go to step 1 and choose another date. 4. System validates data and prompts for illegal values. 5. If the input is valid then the system adds a new exam with given information.   SHOW:   1. System shows a list of exams with all information from step 1.   EDIT:  Step 1-5 as above and then   1. Enter or edit values for one of the exam fields from step 1. 2. System updates the selected field in the exam.     REMOVE:  Step 1-5 as above and then   1. Verify deleting the selected exam. 2. System approves the deletion and removes the exam from the list. |

|  |  |
| --- | --- |
| **Use case** | **User Selection** |
| **Summary** | User is asked to enter an ID number to view the schedule. |
| **Actor** | Staff, Students and Teachers. |
| **Precondition** | Open the webpage. |
| **P Postcondition** | The user can view all the relevant information about the exams. |
| **Base sequence** | 1. System asks for type of user (student, teacher or stuff). |

|  |  |
| --- | --- |
| **Use case** | **View schedule** |
| **Summary** | Showing the schedule to students, teachers and stuff. |
| **Actor** | Students, teachers, stuff. |
| **Precondition** | User must select the type of user. |
| **Postcondition** | The user can view information about the exams. |
| **Base sequence** | 1. System checks, what type of user is logged in 2. Accordingly, to the type of user any irrelevant information is filtered out so that the schedule is easily readable. (the data is displayed as a table) |

## Link between requirements and use cases

**Use case Covered requirements**

Administrator Login 1,13,15,

Create new exam 2,3,4,5,6,7,8,9,10

Manage students’ and teacher’s data 12

Manage exam data 11

View schedule 14,16

User

## 

# Design jano diagrams, Lenka description

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 jano

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 joao

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 joao

The purpose of the results and discussion section is to present the outcome and achieved results of the project.

# Conclusions lenka

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 juan

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**

**Final Exam Planner**

**Jan Lishak – 294322 Lenka Orincakova - 293085**

**Juan Iglesias Trebolle – 293143**

**João Bernardo Baptista Vieira Dias - 293133**

**Software Technology Engineering 1st Semester**

**18th September 2019**

Supervisor: Astrid Hanghøj, Michael Viuff | Number of words : 1338

**Table of content**

[Background Description 3](#_bookmark0)

[Problem Statement 4](#_bookmark1)

[Definition of purpose 5](#_bookmark2)

[Delimitation 6](#_bookmark3)

[Methodology 7](#_bookmark4)

[Time schedule 8](#_bookmark5)

[Risk assessment 9](#_bookmark6)

[Source of information 10](#_bookmark7)

[Group contract 11](#_bookmark8)

# Background Description

At the end of every semester, the examiners at the university need to plan exam schedule before the examination starts. This task may become very complex and time-wasting due to the number of students and the subjects in which the students are being examined.

The schedule plan must meet many specific criteria. For instance, one student should have one exam in consecutive days. Other external factors need to be considered as well. The rooms must be prepared beforehand so they correspond to the format of the examination. All those conditions make the process of planning more difficult and a real issue to be solved.

Some schools accomplish the planning of exams by making a spreadsheet and manually choose all the times for specific exams. The disadvantage is that the person that makes the plan must check if every exam is scheduled at a time that does not clash with any other exam.

Due to human factor, errors in planning might occur and the schedule might change at a time close before the examination. Those changes can make the exam stressful, both for students and professors. Such problem can be that the supposed room where the examination is held does not have the right technical functionalities such as HDMI port. This could be avoided if there was a system that could check all the criteria automatically.

Scheduling in general is simple task but when there are so many conditions for the right plan, the task can become very complex. Computers might possibly help to speed up the process of scheduling. Some solutions using computers already exist, but every university have different conditions and needs to plan their exams. It is hardly possible to make a system that will consider every need of all the different Universities. Therefore, it would be ideal if every university made their own system that meets their needs.

The whole problem also consists of the delivery of the exam plan. Even if the schedule plan was created perfectly without any errors, external factors might affect the schedule. A professor can become unable to come to work at the specified time or a student might become sick. The schedule must be flexible and delivered to the students in a way that the information is always updated and easy to access. If the schedule was delivered in a PDF file all the people being involved in the schedule would have to download a new file when a new version is published. It would be better if the schedule could be viewed by a student or a teacher in a way that it filters all irrelevant information.

With the use of modern technologies, it is possible to make the task much more efficient and easier than by using spreadsheets or just a pen and paper. Therefore, the problem is relevant to solve and open to new solutions.

# Problem Statement

Main problem:

How can the system help the user to check if all conditions are met?

Sub-problems:

1. How can the process be made faster?
2. How can the system help the user to check if all conditions are met?
3. How should the schedule be shown?
4. Could software solve this problem? What are the benefits?
5. How can the system be made more automatic?
6. What kind of user interface would be the easiest to use?

# Definition of purpose

The purpose is to improve the administrator’s efficiency. Making his job easier and faster.

# Delimitation

1. The service department will be contacted by the third party like the examiner itself.

# Methodology

The methodology chosen for this project is Waterfall, which is divided in 7 phases:

Firstly, **requirements phase**. We met the customer to gather all the requirements needed for the project. This is a key part for the success of the project, because every other phase will be planned without further customer involvement.

Secondly, **analysis**. We analyzed the system in order to properly generate the models that will be used in the application.

Thirdly, the **design** phase. This is broken up into 2 sub-phases:

Logical design sub-phase. Here we brainstormed theoretical possible solutions.

Physical design sub-phase. Here those theoretical ideas and schemas are made into concrete specifications.

Fourth, **implementation.** We assimilated the requirements and specifications from the previous phases and produce actual code.

Fifth, **testing**. Here the testers will discover and report issues with the application. The code from previous phases will be repeated and improved in order to eliminate those bugs.

Sixth, **verification**. The customer will review the product to make sure that it meets all the requirements laid out at the beginning of the project. The product shown to the customer is a “final product”.

Seventh, **maintenance**. In this phase, the customer will be using the product regularly, discovering bugs, inadequate features, etc. The team will work on those problems until the customer is satisfied.

# Time schedule

This timeline is based on our plan how to efficiently manage our time.

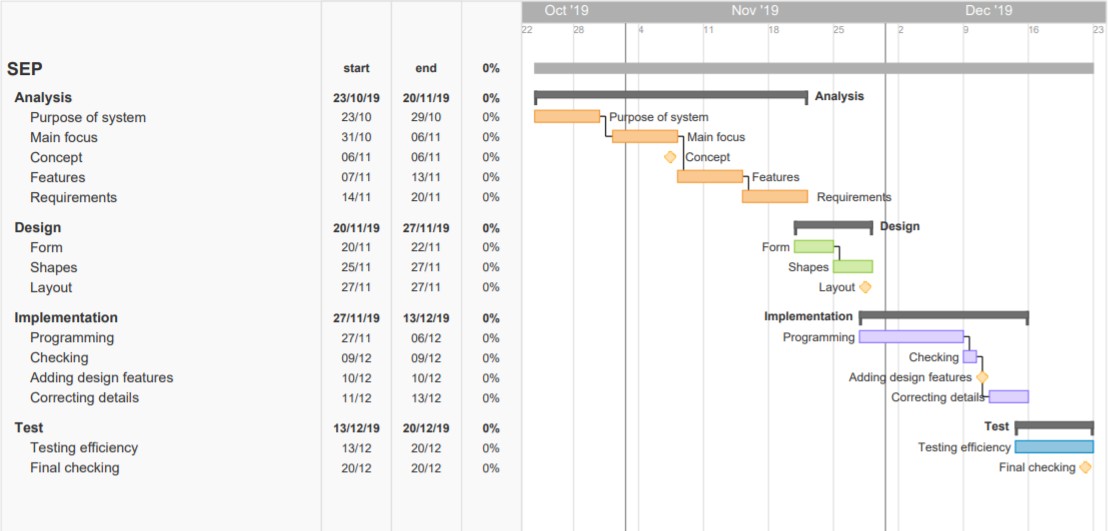
Firstly, focus will be on analysis of the project, where we will discuss what is the purpose. Also, this section includes discussing features and requirements given by the customer. Time spent on this part will be approximately 70 hours per person.

Secondly, we will discuss the design of final product. This would mean 20 hours per person.

Implementation of all information is one of the most important parts. Therefore, 35 hours per person is expected in the third part.

Testing is the last part of our project. Efficiency and clarity will be inspected. Expected time is 15 hours per person.

Total amount of hours is 140 hours per person.



# Risk assessment

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Risks | Likelihood Scale: 1-5  5 = high risk | Severity Scale: 1-  5  5 = high risk | Product of likelihood and severity | Risk mitigation e.g.  Preventive- & Responsive  actions | Identifiers | Responsible |
| The software could crash due to the number of  students. | 2 | 2 | 4 | Separate the students/clas ses into smaller  groups. | Crash of the software. | João Dias |
| If the information is not handled properly there could be security  fails. | 3 | 5 | 15 | Make the software run offline. | Spills in information. | Jan Lishak |
| There may be an overlap in the classroom  reservation. | 1 | 3 | 3 | Leave an available room for written and one for oral  exams. |  | Lenka Orincakova |
| There might be some student information incorrectly written | 4 | 4 | 16 | Put a button that will let the student suggest a fix. | Student having exams in time / subjects they should not | Juan Trebollle |

# Source of information

Andrew Powell-Morse, 2016, *Waterfall Method, What Is It and When Should I Use It?*...Available at: <https://airbrake.io/blog/sdlc/waterfall-model>

IEEE Computer Society, 2008. IEEE Std 829-2008, IEEE Standard for Software and System Test Documentation,

*Waterfall Methodology in Project Management. Available at:* [*https://www.projectmanager.com/software/use-cases/waterfall-methodology*](https://www.projectmanager.com/software/use-cases/waterfall-methodology)

# 

# Group contract

Group Name: **Group 8** Date: **07/10/2019**

These are the terms of group conduct and cooperation that we agree on as a team.

## Participation:

We agree on equally participating in each project, being active and responsible. We will work effectively and be concentrated.

## Communication:

We agree on communicate openly about issues and making decisions. We are expected to tell new ideas, opinions or disagreements with respect, please.

## Meetings:

We agree to participate in every meeting we agreed on, unless the person informs the group in advance.

## Conduct:

We agree on being polite, taking our responsibilities seriously, with good mood. 😊

## Conflict:

We agree on discussing possible conflicts, solving them and making final agreement.

## Deadlines:

We agree to finish our project on a deadline.

## Other Issues:

Bringing snacks when you cook!



|  |  |  |
| --- | --- | --- |
| **Group member’s name** | **Student number** | **Signature** |
| Lenka Orincakova | 293085 |  |
| Juan Iglesias Trebolle | 293143 |  |
| Joao Bernardo Dias | 293133 |  |
| Jan Lishak | 294322 |  |