**Requirements**

**Discipline Document**

**INF-F Analysis and Software Design**

**Project: Schedule App Hague University**

**Date:** 3**-**4**-2014**

**Owner: *Jansen, Ryanne. Zygnarowski, Kacper***

**Version: 1.**4

**Document History**

**Revision History**

|  |  |  |
| --- | --- | --- |
| **Version** | **Revision date** | **Summary of changes** |
| 1.1 | 7-3-2014 | Added chapter 3: Description of Requirements Based on Case Materials |
| 1.2 | 14-3-2014 | Chapter content is updated. Chapter 4 and 5, 10, 13 partially added. |
| 1.3 | 28-3-2014 | Chapter structure is changed. Chapters are updated.Chapter 6, 7, 8 ,9 added. |
| 1.4 | 3-4-2014 | Chapter content is updated. Chapter 11, 12, 13.2 added. |

**Distribution**

This document is intended for:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Function** | **Date of issue** | **Version** |
| van Dordrecht, Eric | Team manager | 28-03-2014 | 1.4 |
| Josine | Course manager | 28-03-2014 | 1.4 |
| Maas, Juul | Expert RE/BM | 28-03-2014 | 1.4 |
| Thea | Teacher | 28-03-2014 | 1.4 |
| van Aalten, Jos | Client | 28-03-2014 | 1.4 |
| Lustenhouwer, Wim | Project Manager | 28-03-2014 | 1.4 |
| van Damme, Djastin | Project Co-Manager | 28-03-2014 | 1.4 |

**Table of Contents**

[1. Introduction](#h.u5azoms8zogo)

[2. System Scope](#h.1fob9te)

[3. Description of Requirements Based on Case Materials](#h.3znysh7)

[3.1. User Requirements based on Case Materials](#h.cl7hs0lq99i2)

[3.2. Business Rules](#h.3dy6vkm)

[3.3. Business Requirements](#h.1t3h5sf)

[4. Description of User Requirements Based on Interviews](#h.4d34og8)

[5. Use Cases](#h.2s8eyo1)

[5.1. Use Case Diagram](#h.mfbpj26gi0ve)

[5.2. Actor List](#h.3rdcrjn)

[5.3. Use Case List (User Requirements)](#h.26in1rg)

[5.4. Use Cases Without Priority](#h.ygmj96y0pui7)

[5.4. Additional Information](#h.lnxbz9)

[6. Prioritized Requirements (based on 3, 4 and 5)](#h.35nkun2)

[7. Functional User Requirements not in Use Cases](#h.1ksv4uv)

[8. Functional and Non-Functional Software Requirements](#h.44sinio)

[8.1. Functional Software Requirements](#h.tjmztphmc63b)

[8.2. Non-Functional Software Requirements](#h.vfjracvlrywj)

[9. Technical Constraints](#h.2jxsxqh)

[10. Use Case Specification Conform Priority](#h.3j2qqm3)

[11. Additional Information](#h.41mghml)

[11.1. Requirements](#h.2vjyty9gi4ix)

[11.2. Use Cases](#h.5gmg57371s31)

[12. Verification Report](#h.2grqrue)

[13. Navigation Map](#h.m0mutnqrhwd)

[13.1. Navigation Map](#h.laln33snnjx)

[13.2. User Interface Convention](#h.1v1yuxt)s

# 1. Introduction

The Academy for IT and Media has asked the project group to analyze the current system and come up with a proposal to improve this system, resulting in an overall process that is faster, more efficient and cost effective. This document intends to describe the requirements needed to realize that goal.

The requirements are based on the supplied business case and various interviews, as described in the PID. They will serve as the baseline for the design of the software.

# 2. System Scope

The system will be used by the IT academy for planning and designing courses, managing the deployment of teachers per term (which courses run in which term is specified in the year plan) and their non-course specific responsibilities. Other academies of the HHS will not be using our system.

The business case can be divided into several processes as described in the business modeling discipline. The system will primarily deal with the two most important business use cases, specifically the term planning, and course design.

The output of our system will be input for the scheduling department, which will then compile a complete schedule from this information. No alterations will be made to this process because it falls outside of the scope of the system.

# 3. Description of Requirements Based on Case Materials

Each requirement is identified by a unique code. This code consists of a requirement type and a requirement number.

Requirement type definition is as follows:

B Business Rule

BR Business Requirement

U User Requirement

S Functional Software Requirement

NS Non-Functional Software Requirement

T Technical Constraint

Source definition is as follow:

CASE Case Description

BCASE Business Case

INTERVIEW Interview Records

CONSENSUS Choice made during process

## 3.1. User Requirements based on Case Materials

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Description | Stakeholder | Source | BR | BUC | Priority |
| U02 | Ability to manage constraints of a course. | Course manager | BCASE | 1,4 | 2 | C |
| U05 | Ability to manage the course parameters. | Course manager | BCASE | 1,5 | 2 | M |
| U22 | Ability to see the PTO per teacher(team manager exclusive). | Team manager | BCASE | 1 | 5 | S |
| U23 | Ability to see the total costs (in hours) of a course after a run. | Team manager | BCASE | 7 | 5 | S |
| U37 | Wants the system to be more user friendly and accessible. | Team manager  Course manager | BCASE | 1,2 | 2,3,4 | S |
| U01 | Ability to generate management reports. | Client | BCASE | 7 | 5 | C |
| U03 | Ability to manage course specific tasks of a course. | Course manager | CASE | 2 | 2 | M |
| U04 | Ability to see the available budget for a course. | Course manager | CASE | 2 | 1,2 | S |
| U13 | Ability to indicate how many student groups are acceptable for a course  (based on the number of students that have signed up or the course and the group size). | Course manager | CASE | 4,5 | 2 | S |
| U17 | Ability to manage teachers availability in school hours. | Team manager | CASE | 1,2,5 | 3 | S |
| U18 | Ability to see the current working hours of the teachers. | Team manager | CASE | 1 | 3 | S |
| U19 | Ability to assign teachers to non course specific tasks. | Team manager | CASE | 1,2,4 | 3 | M |
| U20 | Ability to manage non course specific tasks. | Team manager | CASE | 1,2,5 | 3 | M |
| U21 | Ability to assign teachers to course specific tasks. | Team manager | CASE | 1,2,4 | 3 | M |
| U38 | Wants the system to be more time-saving. | Client | PID | 1,2 | 2,3,4 | M |
| U39 | Wants the system to be less error prone. | Client | PID | 1,2,4 | 2,3,4 | M |
| U40 | Wants the system to be extendable. | Client | PID | 1,2 | 2,3,4 | W |

## 

## 

## 

## 

## 

## 3.2. Business Rules

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Stakeholder | Source |
| B01 | Fulltime teachers can give lessons during the day and in the evening. | Team manager | CASE |
| B02 | Teachers specific to part-time students can only teach in the evening. | Team manager | CASE |
| B03 | In a PTO teacher tasks are inventorised for 10 weeks. | Team manager | CASE |
| B04 | Assignment of annual tasks to teachers happens in May / June. | Team manager | CASE |
| B05 | Each teacher is assigned to tasks according to their working hours. | Team manager | CASE |
| B06 | Budget in hours for the main phase is 15 \* 0.6 \* number of enrolled students. | Course manager | CASE |
| B07 | An activity is carried out by one or more teachers. | Scheduling department | CASE |
| B08 | An activity cannot be divided into sub-activities, due to a limitation of the Iris program. | Scheduling department | CASE |
| B09 | An activity can be linked to a type of classroom, but this is not a necessity. | Scheduling department | CASE |
| B10 | An activity must be bound to one or more groups of students. | Scheduling department | CASE |
| B11 | A group should not be smaller than two students. Groups larger than eight are not allowed. | Scheduling department | CASE |
| B18 | A group of students follow a specific course during a specific period in time. | Scheduling department | CASE |
| B12 | A teacher can be only scheduled for an activity on a day or an hour (hours) on the basis of scheduling preferences. | Scheduling department | CASE |
| B13 | Team managers should make schedule preferences known before the scheduling process begins. | Scheduling department | CASE |
| B16 | Every room has a specific room type. | Scheduling department | CASE |
| B17 | A room type describes the most important properties of a specific room. | Scheduling department | CASE |
| B18 | Each course has a designated term. | Team manager | Interview 17-3 |
| B19 | A teacher is allowed to see its own PTO. | Team manager | Interview 18-2 |
| B20 | A team manager is allowed to see everyones PTO. | Team manager | Interview 18-2 |
| B21 | A teaching unit consists of course specific tasks. | Team manager | Interview 18-2 |

## 

## 

## 3.3. Business Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Description | Stakeholder | Source | UR | BUC |
| BR01 | Wants the desired system to be more productive, so the system’s users will spend less hours on planning a term. | Team manager | BCASE | U22,U18,U26,U02,U05,U14,U15,U19,  U21,U10,U11,U12,U17,U20,U33,  U34,U35,U24,U27,U29,U37,U38,U39,U40,U41 | 2,3 |
| BR02 | Wants less faults being made in the planning process by inaccuracy. | Team manager | BCASE | U03,U04,U07,U19,U21,U10,U11,  U12,U17,U20,U33,U34,U35,U24,U27,  U29,U37,U38,U39,U40,U41 | 2,3 |
| BR03 | The system should include the year planning as a basis for course planning. | Client | Interview | U32 | 1,2,3 |
| BR04 | Wants improvement in the way how the team manager gets the data required to complete the planning process  (not by e-mail anymore). | Team manager,  Course manager | BCASE | U02,U07,U13,U16,U30,U19,U21,U08,U39 | 2,3,5 |
| BR05 | Maintenance of course and terms has to be centralized in one system. | Team manager  Course manager | Interview | U05,U14,U15,U13,U16,U30,U25,U28,  U10,U11,U12,U17,U20,U33,U34,  U35,U24,U27,U29,U08 | 2,3,5 |
| BR06 | Wants improvement in communication with scheduling department through the system. | Team manager | Interview | U06,U42 | 4 |
| BR07 | During or after a term, the business wants an overview of effects and feedback on that term in order to improve  planning of reused courses. | Client | Interview | U01,U23,U31,U08 | 5 |



# 

# 

# 

# 4. Description of User Requirements Based on Interviews

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Description | Stakeholder | Source | BR | BUC | Priority |
| U08 | Ability to add feedback/comments to a task,course or activity. | Course manager | Interview 4-3 | 4,5,7 | 5 | C |
| U09 | Ability to make changes to a course, which will be saved separately as a mutation. | Course manager | Interview 4-3 | 1,2,5 | 5 | S |
| U10 | Ability to change the student group size in a course parameter. | Course manager | Interview 4-3 | 1,2,5 | 2 | S |
| U11 | Ability to indicate a dependency between activities that need to be given together  (e.g.: lab after corresponding workshop). | Course manager | Interview 4-3 | 1,2,5 | 2 | C |
| U12 | Ability to make changes to a course, which will be saved separately as a mutation. | Course manager | Interview 4-3 | 1,2,5 | 2,3,5 | S |
| U16 | Ability to see own (temporary) PTO. | Teacher | Interview 4-3 | 4,5 | 5 | M |
| U28 | Ability to have several screens open to group data and get a quick overview thereof. | Team manager | Interview 4-3 | 5 | 2,3,5 | M |
| U29 | Ability for team manager to lock and unlock a course in order to prevent changing the  course data during the scheduling process (some courses are given by the same teachers each year). | Team manager | Interview 4-3 | 1,2,5 | 4 | C |
| U41 | Wants the system view-language to be Dutch. | Client | Interview 21-3 | 1,2 | 2,3,4 | M |
| U31 | Ability to see the given task history per teacher. | Team manager | Interview 19-3 | 7 | 3 | S |
| U42 | Wants the ability to export all data. | Team manager | Interview 18-3 | 6 | 3,4 | M |
| U06 | Ability to request a specific classroom for an activity. | Course manager | Interview 18-2 | 6 | 2,4 | C |
| U07 | Ability to request a specific teacher for an activity. | Course manager | Interview 18-2 | 2,4 | 2 | S |
| U24 | Ability to manage personal details teachers. | Team manager | Interview 18-2 | 1,2,5 | 3 | W |
| U25 | Ability to see all mutations of a course. | Team manager | Interview 18-2 | 5 | 3 | S |
| U26 | Ability to see the student workload per week in a course. | Team manager | Interview 18-2 | 1 | 3 | C |
| U27 | Ability to manage roles (teacher, course manager etc.). | Team manager | Interview 18-2 | 1,2,5 | 2,3,5 | M |
| U30 | Ability to see the meaning of abbreviations that are in course templates(if abreviations are used). | Team manager | Interview 18-2 | 4,5 | 2,3,4,5 | W |
| U14 | Ability to manage teaching units for a course. | Course manager | Interview 17-3 | 1,5 | 1,2 | S |
| U15 | Ability to manage activities. | Course manager | Interview 17-3 | 1,5 | 2 | M |
| U32 | Ability to see a list of courses per term. | Team manager | Interview 17-3 | 3 | 3 | S |
| U33 | Ability to manage specialisations per teacher. | Team manager | Interview 17-3 | 1,2,5 | 3 | M |
| U34 | Ability to manage specialisations. | Team manager | Interview 17-3 | 1,2,5 | 3 | M |
| U35 | Ability to manage specialisations per task. | Team manager | Interview 17-3 | 1,2,5 | 2,3 | M |
| U36 | Ability to manage the history of the teachers availability. | Team manager | Interview 17-3 | 1,4 | 2 | C |

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

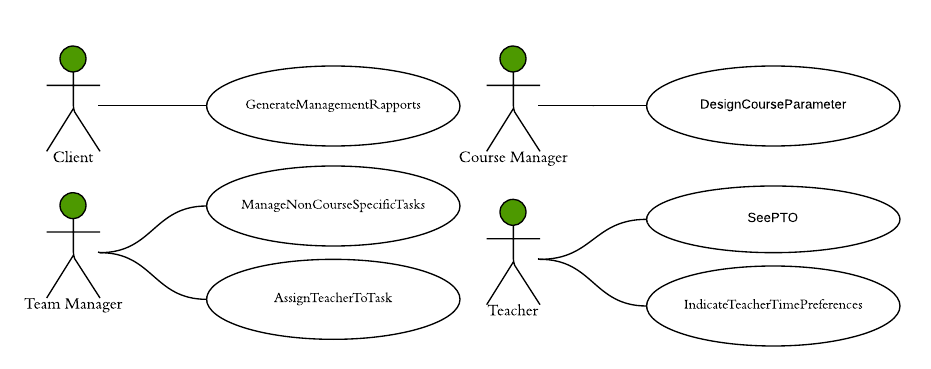
# 

# 

# 

# 5. Use Cases

## 5.1. Use Case Diagram



## 5.2. Actor List

|  |  |
| --- | --- |
| P**rimary** actor | **Description** |
| **Course manager** | The course manager creates or modifies course parameters. |
| **Team manager** | The team manager adds teachers to tasks defined in the course parameter and sends it to the scheduling department.  The team manager also manages the teachers data. |
| **Teacher** | The teacher is able to indicate his availability and view his own PTO |

|  |  |
| --- | --- |
| S**econdary actor** | **Description** |
| **Webber** | Webber provides students and teachers with their schedule. |
| S**chedul**ing **department** | The scheduling department schedules all activities obtained from the mastersheet, received from the team manager.  The scheduling department uses Webber as a view on the schedule. |
| **Client** | The client receives generated reports with data of the course, the teachers and students. |
| **Student** | Students can see their schedule using Webber. |

## 

## 5.3. Use Case List (User Requirements)

|  |  |  |  |
| --- | --- | --- | --- |
| Use  Case ID | Use Case Name | Actor | Source Requirements |
| 1 | Generate management report | Client | U01 |
| 2 | Assign teachers to tasks | Team manager | U17, U18, U19, U20, U21, U31, U32, U33, U34, U35, U36 |
| 3 | Create course parameter | Course manager | U02, U03, U04, U05, U06, U07, U10, U11, U12, U13, U14, U15, U25 |

## 5.4. Use Cases Without Priority

\*For all use cases the actor is logged in.

|  |  |
| --- | --- |
| ID / Name | 1 / Generate management reports |
| Description | Generates the management reports. |
| Primary actor | Client |
| Secondary actor | None |
| Precondition(s) | 1 None |
| Main flow | 1. The actor chooses one of the following reports: [1.1]  - teacher hours  - summary of the costs of a course  - course history  - number of students registrations  - available expertise and expertise that has to be present  3. The system shows the data on the screen.  4. The actor chooses to save the report. [1.2] |
| Postcondition(s) | The report is generated on the screen and saved. |
| Alternative flow | [1.1] Gives error: ‘Data not available’.  [1.2] PrintReport: See Alternative flow |

|  |  |
| --- | --- |
| ID / Name | 1.2 / PrintReport |
| Description | The client chooses to print the report. |
| Primary actor | Client |
| Secondary actor | None |
| Precondition(s) | Data is generated on the screen. |
| Alternative flow | 1. 1.The use case starts after step 4 of the Main Flow.  . 2. The actor chooses to print the report.  3. 3. The system sends the report to the printer software.  4. 4. The use case resumes at step 1 of the Main Flow. |
| Postcondition(s) | The report is printed. |

|  |  |
| --- | --- |
| ID / Name | 2 / AssignTeacherToTask |
| Description | Assign a teacher to task. |
| Primary actor | Team manager |
| Secondary actor | None |
| Precondition(s) | N None |
| Main flow | 1. The use case starts when the team manager wants to assign a teacher to a task.  2. The system shows a list of teachers and their expertises.  3. The actor assigns a teacher to a task. [2.1]  4. The system confirms that the teacher has been added. [2.2] |
| Postcondition(s) | 1. 1. The system has saved the links between tasks and teachers. |
| Alternative flow | [2.1] WorkloadExceeded: The system gives an error about the teachers workload being exceeded.  [2.2] NotAssigned: The system gives an error because the teacher has not been added to the task. |

|  |  |
| --- | --- |
| ID / Name | 3 / DesignCourseParameter |
| Description | Designing the course parameter. |
| Primary actor | Course manager |
| Secondary actor | None |
| Precondition(s) | e. None |
| Main flow | 1. 1. The use case starts when the course manager chooses to design a course parameter.  2. 2. The system creates a new course parameter [3.1] or imports an existing one.  ffff 3. The actor creates a task.  3.1. The actor assigns activities to the task.  4. The system saves the course parameter. |
| Postcondition(s) | 1. 1. The course parameter is filled in.  2 |
| Alternative flow | [3.1] Course does not exist |

|  |  |
| --- | --- |
| ID / Name | 3.1 / Course does not exist |
| Description | The course that needs to be designed does not exist yet. |
| Primary actor | Course manager |
| Secondary actor | None |
| Precondition(s) | None |
| Alternative flow | 1. The use case starts after step 2 of the main flow.  2. The actor creates teaching units.  3. Done. |
| Postcondition(s) | The new course is created with teaching units. |

## 

## 5.4. Additional Information

The glossary has been designed specifically for the stakeholders and for the project team. It establishes a mutual reference point, clarifying terms used in the project.

The system’s database will be updated automatically when an edit is made. When the client wants to view a management report, he can decide which kind of report.

# 

# 6. Prioritized Requirements (based on 3, 4 and 5)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Description | Stakeholder | Source | BR | BUC | Priority |
| U38 | Wants the system to be more time-saving. | Client | PID | 1,2 | 2,3,4 | M |
| U39 | Wants the system to be less error prone. | Client | PID | 1,2,4 | 2,3,4 | M |
| U16 | Ability to see own (temporary) PTO. | Teacher | Interview 4-3 | 4,5 | 5 | M |
| U28 | Ability to have several screens open to group data and get a quick overview thereof. | Team manager | Interview 4-3 | 5 | 2,3,5 | M |
| U41 | Wants the system view-language to be Dutch. | Client | Interview 21-3 | 1,2 | 2,3,4 | M |
| U42 | Wants the ability to export all data. | Team manager | Interview 18-3 | 6 | 3,4 | M |
| U27 | Ability to manage roles (teacher, course manager etc.). | Team manager | Interview 18-2 | 1,2,5 | 2,3,5 | M |
| U15 | Ability to manage activities. | Course manager | Interview 17-3 | 1,5 | 2 | M |
| U33 | Ability to manage specialisations per teacher. | Team manager | Interview 17-3 | 1,2,5 | 3 | M |
| U34 | Ability to manage specialisations. | Team manager | Interview 17-3 | 1,2,5 | 3 | M |
| U35 | Ability to manage specialisations per task. | Team manager | Interview 17-3 | 1,2,5 | 2,3 | M |
| U03 | Ability to manage course specific tasks of a course. | Course manager | CASE | 2 | 2 | M |
| U19 | Ability to assign teachers to non course specific tasks | Team manager | CASE | 1,2,4 | 3 | M |
| U20 | Ability to manage non course specific tasks. | Team manager | CASE | 1,2,5 | 3 | M |
| U21 | Ability to assign teachers to course specific tasks | Team manager | CASE | 1,2,4 | 3 | M |
| U05 | Ability to manage the course parameters. | Course manager | BCASE | 1,5 | 2 | M |
| U09 | Ability to make changes to a course, which will be saved separately as a mutation. | Course manager | Interview 4-3 | 1,2,5 | 5 | S |
| U10 | Ability to change the student group size in a course parameter. | Course manager | Interview 4-3 | 1,2,5 | 2 | S |
| U12 | Ability to make changes to a course, which will be saved separately as a mutation. | Course manager | Interview 4-3 | 1,2,5 | 2,3,5 | S |
| U31 | Ability to see the given task history per teacher. | Team manager | Interview 19-3 | 7 | 3 | S |
| U07 | Ability to request a specific teacher for an activity. | Course manager | Interview 18-2 | 2,4 | 2 | S |
| U25 | Ability to see all mutations of a course. | Team manager | Interview 18-2 | 5 | 3 | S |
| U14 | Ability to manage teaching units for a course. | Course manager | Interview 17-3 | 1,5 | 1,2 | S |
| U32 | Ability to see a list of courses per term. | Team manager | Interview 17-3 | 3 | 3 | S |
| U04 | Ability to see the available budget for a course. | Course manager | CASE | 2 | 1,2 | S |
| U13 | Ability to indicate how many student groups are acceptable for a course  (based on the number of students that have signed up or the course and the group size). | Course manager | CASE | 4,5 | 2 | S |
| U17 | Ability to manage teachers availability in school hours. | Team manager | CASE | 1,2,5 | 3 | S |
| U18 | Ability to see the current working hours of the teachers. | Team manager | CASE | 1 | 3 | S |
| U22 | Ability to see the PTO per teacher(team manager exclusive). | Team manager | BCASE | 1 | 5 | S |
| U23 | Ability to see the total costs (in hours) of a course after a run. | Team manager | BCASE | 7 | 5 | S |
| U37 | Wants the system to be more user friendly and accessible. | Team manager  Course manager | BCASE | 1,2 | 2,3,4 | S |
| U08 | Ability to add feedback/comments to a task,course or activity. | Course manager | Interview 4-3 | 4,5,7 | 5 | C |
| U11 | Ability to indicate a dependency between activities that need to be given together  (e.g.: lab after corresponding workshop). | Course manager | Interview 4-3 | 1,2,5 | 2 | C |
| U29 | Ability for team manager to lock and unlock a course in order to prevent changing the  course data during the scheduling process (some courses are given by the same teachers each year). | Team manager | Interview 4-3 | 1,2,5 | 4 | C |
| U06 | Ability to request a specific classroom for an activity. | Course manager | Interview 18-2 | 6 | 2,4 | C |
| U26 | Ability to see the student workload per week in a course. | Team manager | Interview 18-2 | 1 | 3 | C |
| U36 | Ability to manage the history of the teachers availability. | Team manager | Interview 17-3 | 1,4 | 2 | C |
| U01 | Ability to generate management reports. | Client | BCASE CASE | 7 | 5 | C |
| U02 | Ability to manage constraints of a course. | Course manager | BCASE | 1,4 | 2 | C |
| U40 | Wants the system to be extendable. | Client | PID | 1,2 | 2,3,4 | W |
| U24 | Ability to manage personal details teachers. | Team manager | Interview 18-2 | 1,2,5 | 3 | W |
| U30 | Ability to see the meaning of abbreviations that are in course templates (if abbreviations are used). | Team manager | Interview 18-2 | 4,5 | 2,3,4,5 | W |

# 7. Functional User Requirements not in Use Cases

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Description | Stakeholder | Source | BR | BUC | Priority |
| U16 | Ability to see own (temporary) PTO. | Teacher | Interview 4-3 | 4,5 | 5 | M |
| U23 | Ability to see the total costs (in hours) of a course after a run. | Team manager | BCASE | 7 | 5 | S |
| U24 | Ability to manage personal details teachers. | Team manager | Interview 18-2 | 1,2,5 | 3 | W |
| U26 | Ability to see the student workload per week in a course. | Team manager | Interview 18-2 | 1 | 3 | C |
| U27 | Ability to manage roles (teacher, course manager etc.). | Team manager | Interview 18-2 | 1,2,5 | 2,3,5 | M |
| U28 | Ability to have several screens open to group data and get a quick overview thereof. | Team manager | Interview 4-3 | 5 | 2,3,5 | M |
| U29 | Ability for team manager to lock and unlock a course in order to prevent changing the  course data during the scheduling process (some courses are given by the same teachers each year). | Team manager | Interview 4-3 | 1,2,5 | 4 | C |
| U30 | Ability to see the meaning of abbreviations that are in course templates(if abbreviations are used). | Team manager | Interview 18-2 | 4,5 | 2,3,4,5 | W |
| U08 | Ability to add feedback/comments to a task,course or activity. | Course manager | Interview 4-3 | 4,5,7 | 5 | C |
| U09 | Ability to make changes to a course, which will be saved separately as a mutation. | Course manager | Interview 4-3 | 1,2,5 | 5 | S |

# 8. Functional and Non-Functional Software Requirements

## 8.1. Functional Software Requirements

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Description | UR | Source | BR | BUC | Prio | Stakeholder |
| S01 | The system could be able to generate management reports from a specific term. | 1 | BCASE CASE | 7 | 5 | C | Client |
| S02 | The system must be able to create course specific tasks. | 3 | CASE | 2 | 2 | M | Course manager |
| S03 | The system must be able to view course specific tasks with no mutations. | 3 | CASE | 2 | 2 | M | Course manager |
| S04 | The system must be able to edit course specific tasks if no teacher or activity is assigned. | 3 | CASE | 2 | 2 | M | Course manager |
| S05 | The system must be able to delete course specific tasks if no teacher or activity is assigned. | 3 | CASE | 2 | 2 | M | Course manager |
| S06 | The system must be able to view course specific tasks with the latest mutation. | 3 | CASE | 2 | 2 | M | Course manager |
| S07 | The system should display the available budget for a course. | 4 | CASE | 2 | 2 | S | Course manager |
| S08 | The system should calculate the available budget for a course based on the formula described in the case | 4 | CASE | 2 | 2 | S | Course manager |
| S09 | The system can create a course | 5 | BCASE | 1,5 | 2 | M | Course manager |
| S10 | The system must be able to edit specific course | 5 | BCASE | 1,5 | 2 | M | Course manager |
| S11 | The system must be able to view specific course | 5 | BCASE | 1,5 | 2 | M | Course manager |
| S12 | The system could have an option to request a specific classroom for an activity. | 6 | Interview 18-2 | 6 | 2,4 | C | Course manager |
| S13 | The system could have an option to undo the request for specific classroom for an activity. | 6 | Interview 18-2 | 6 | 2,4 | C | Course manager |
| S14 | The system can create a room type | 6 | Interview 18-2 | 6 | 2,4 | C | Course manager |
| S15 | Ability to assign a room type to activities | 6 | Interview 18-2 | 6 | 2,4 | C | Course manager |
| S16 | Ability to assign room type to task (resulting in all activities within that task) | 6 | Interview 18-2 | 6 | 2,4 | C | Course manager |
| S17 | Ability to add room type to classroom | 6 | Interview 18-2 | 6 | 2,4 | C | Course manager |
| S18 | When requesting a classroom for an activity, the activities room type must be 0 or the same as the activity | 6 | Interview 18-2 | 6 | 2,4 | C | Course manager |
| S19 | Ability to unassign a room type from activities | 6 | Interview 18-2 | 6 | 2,4 | C | Course manager |
| S20 | Ability to unassign room type from task (resulting in all activities within that task) | 6 | Interview 18-2 | 6 | 2,4 | C | Course manager |
| S21 | Ability to remove room type from classroom | 6 | Interview 18-2 | 6 | 2,4 | C | Course manager |
| S22 | Ability to delete a room type (nulling places where room type was assigned) | 6 | Interview 18-2 | 6 | 2,4 | C | Course manager |
| S23 | The system should have an option to request a specific teacher for a whole task. | 7 | Interview 18-2 | 2,4 | 2 | S | Course manager |
| S24 | The system should have an option to undo the request for specific teacher for a whole task. | 7 | Interview 18-2 | 2,4 | 2 | S | Course manager |
| S25 | The system should have an option to request a specific teacher for one activity. | 7 | Interview 18-2 | 2,4 | 2 | S | Course manager |
| S26 | The system should have an option to undo the request for specific teacher for one activity. | 7 | Interview 18-2 | 2,4 | 2 | S | Course manager |
| S27 | The system could have the ability to add comments (textstrings) to courses, tasks and activities. | 8 | Interview 4-3 | 2,4,5 | 5 | C | Course manager |
| S28 | The system could have the ability to edit comments (textstrings) in courses, tasks and activities. | 8 | Interview 4-3 | 2,4,5 | 5 | C | Course manager |
| S29 | The system should be able to create activity mutations | 9 | Interview 4-3 | 1,2,5 | 5 | S | Course manager |
| S30 | The system should be able to display activity mutations | 9 | Interview 4-3 | 1,2,5 | 5 | S | Course manager |
| S31 | The system must provide a way to edit the group size for a course. | 10 | Interview 4-3 | 1,2,4 | 2 | M | Course manager |
| S32 | The system will set default group size to 2 | 10 | Interview 4-3 | 1,2,4 | 2 | M | Course manager |
| S33 | The system can create a link between activities(indicating schedule order) | 11 | Interview 4-3 | 2,6 | 2 | C | Course manager |
| S34 | The user can create a text saved with the link between activities(explanation) | 11 | Interview 4-3 | 2,6 | 2 | C | Course manager |
| S35 | The system can delete a link between activities(which also deletes the explanation text) | 11 | Interview 4-3 | 2,6 | 2 | C | Course manager |
| S36 | The system can view a link between activities. | 11 | Interview 4-3 | 2,6 | 2 | C | Course manager |
| S37 | The system should be able to edit the maximum amount of students that can enroll in a specific course. | 13 | CASE | 4,5 | 2 | S | Course manager |
| S38 | The system should be able to display the maximum amount of students that can enroll in a specific course. | 13 | CASE | 4,5 | 2 | S | Course manager |
| S39 | The system should be able to create teaching unit | 14 | Interview 17-3 | 1,5 | 1,2 | S | Course manager |
| S40 | The system should be able to assign a teaching units to a specific course. | 14 | Interview 17-3 | 1,5 | 1,2 | S | Course manager |
| S41 | The system should be able to view teaching units per course | 14 | Interview 17-3 | 1,5 | 1,2 | S | Course manager |
| S42 | The system should be able to remove teaching units from a specific course. | 14 | Interview 17-3 | 1,5 | 1,2 | S | Course manager |
| S43 | The system should be able to delete teaching units | 14 | Interview 17-3 | 1,5 | 1,2 | S | Course manager |
| S44 | The system should be able to create constraint to an activity | 2 | BCASE | 1,4 | 2 | C | Course manager |
| S45 | The system should be able to edit the constraint of an activity | 2 | BCASE | 1,4 | 2 | C | Course manager |
| S46 | The system should be able to delete the constraint of an activity | 2 | BCASE | 1,4 | 2 | C | Course manager |
| S47 | The system should be able to view the constraint of an activity | 2 | BCASE | 1,4 | 2 | C | Course manager |
| S48 | The system must be able to create activities. | 15 | Interview 17-3 | 1,5 | 2 | M | Course manager |
| S49 | The system must be able to assign activities to tasks. | 15 | Interview 17-3 | 1,5 | 2 | M | Course manager |
| S50 | The system must be able to edit activities. | 15 | Interview 17-3 | 1,5 | 2 | M | Course manager |
| S51 | The system must be able to unassign activities from tasks. | 15 | Interview 17-3 | 1,5 | 2 | M | Course manager |
| S52 | The system must be able to delete activities (if assigned, unassign first) | 15 | Interview 17-3 | 1,5 | 2 | M | Course manager |
| S53 | The system must provide a way to indicate whether a teacher wants to receive temporary PTO's. | 16 | Interview 4-3 | 1,3 | 4,5 | M | Teacher |
| S54 | The system should provide a way to view the working hours of a teacher. | 18 | CASE | 2,4 | 3 | S | Team manager |
| S55 | The system must be able to assign teachers to non course specific tasks. | 19 | CASE | 1,2,5 | 3 | M | Team manager |
| S56 | The system must be able to unassign teachers from non course specific tasks. | 19 | CASE | 1,2.,5 | 3 | M | Team manager |
| S57 | The system must be able to create non course specific tasks. | 20 | CASE | 1,2,5 | 3 | M | Team manager |
| S58 | The system must be able to edit non course specific tasks if no teacher is assigned. | 20 | CASE | 1,2,5 | 3 | M | Team manager |
| S59 | The system must be able to display non course specific tasks. | 20 | CASE | 1,2,5 | 3 | M | Team manager |
| S60 | The system must be able to assign teachers to course specific tasks. | 21 | CASE | 1,2,5 | 3 | M | Team manager |
| S61 | The system must be able to unassign teachers from course specific tasks. | 21 | CASE | 1,2,5 | 3 | M | Team manager |
| S62 | The system should generate the PTO's of teachers. | 22 | BCASE | 1 | 5 | S | Team manager |
| S63 | The system should provide a way for the team manager to view all the PTO's of teachers separately. | 22 | BCASE | 1,4 | 5 | S | Team manager |
| S64 | The system should be able to show the total cost in hours of a designed course parameter. | 23 | BCASE | 4,7 | 5 | S | Team manager |
| S65 | The system will provide a way to edit the personal details of teachers. | 24 | Interview 18-2 | 1,2,5 | 3 | W | Team manager |
| S66 | The system will provide a way to create teachers. | 24 | Interview 18-2 | 1,2,5 | 3 | S | Team manager |
| S67 | The system will provide a way to delete teachers. | 24 | Interview 18-2 | 1,2,5 | 3 | S | Team manager |
| S68 | The system should provide an ability to view all mutations of a course. | 25 | Interview 18-2 | 5 | 3 | S | Team manager |
| S69 | The system could show the student workload per week. | 26 | Interview 18-2 | 1 | 3 | C | Team manager |
| S70 | The system must provide a way to log in. | 27 | Interview 18-2 | 2 | 2,3,5 | M | Team manager |
| S71 | The system must provide a way to log off. | 27 | Interview 18-2 | 2 | 2,3,5 | M | Team manager |
| S72 | The system must provide a way to add rights to roles | 27 | Interview 18-2 | 2 | 2,3,5 | M | Team manager |
| S73 | The system must create rights (which screens are accessible to roles) | 27 | Interview 18-2 | 2 | 2,3,5 | M | Team manager |
| S74 | The system must provide a way to remove rights from roles | 27 | Interview 18-2 | 2 | 2,3,5 | M | Team manager |
| S75 | The system must provide a way to delete rights | 27 | Interview 18-2 | 2 | 2,3,5 | M | Team manager |
| S76 | The system must provide a way to assign roles to teachers | 27 | Interview 18-2 | 2 | 2,3,5 | M | Team manager |
| S77 | The system must provide a way to edit rights | 27 | Interview 18-2 | 2 | 2,3,5 | M | Team manager |
| S78 | The system will provide an ability for team manager to lock a course in order to prevent changing the  course data during the scheduling process | 29 | Interview 4-3 | 1,2,5 | 3 | C | Team manager |
| S79 | The system could provide an ability for team manager to unlock a course in order to make the changing of the  course data possible during the scheduling process | 29 | Interview 4-3 | 1,2,5 | 3 | C | Team manager |
| S80 | The system will provide a way for displaying the full written word, instead an abbreviation. | 30 | Interview 18-2 | 2,3 | 2,3,4,5 | W | Team manager |
| S81 | When a teacher has run a task or activity with a specialisation he or she does not possess, this specialisation  will be added to the teacher's specialisations automatically | 31 | Interview 19-3 | 7 | 3 | S | Team manager |
| S82 | The system must provide a way to assign specialisations to teachers. | 33 | Interview 17-3 | 1,2,4 | 3 | M | Team manager |
| S83 | The system must provide a way to unassign specialisations from teachers. | 33 | Interview 17-3 | 1,2,4 | 3 | M | Team manager |
| S84 | The system must provide a way to see teachers connected to specific specialisations. | 33 | Interview 18-2 | 1,2,4 | 3 | M | Team manager |
| S85 | The system must have the ability to create specialisations. | 34 | Interview 17-3 | 1,2,4 | 3 | M | Team manager |
| S86 | The system must have the ability to delete specialisations if not assigned to teachers, tasks or activities. | 34 | Interview 17-3 | 1,2,4 | 3 | M | Team manager |
| S87 | The system must have the ability to view specialisations. | 34 | Interview 17-3 | 1,2,4 | 3 | M | Team manager |
| S88 | The system must have the ability to assign a specialisation to a task. | 35 | Interview 17-3 | 1,2,4 | 3 | M | Team manager |
| S89 | The system must have the ability to unassign a specialisation from a task. | 35 | Interview 17-3 | 1,2,4 | 3 | M | Team manager |
| S90 | The system should edit the teachers working hours. | 18 | Interview 18-2 | 1,2 | 3 | S | Team manager |
| S91 | The system should provide a way to indicate the minimal working hours (per day and per week). | 18 | CASE | 1,2 | 3 | S | Scheduling department |
| S92 | The system should provide a way to indicate the maximal working hours (per day and per week). | 18 | CASE | 1,2 | 3 | S | Scheduling department |
| S93 | The system should give a notification when a teachers has exceeded his maximal working hours | 18 | Interview 18-2 | 1,2 | 3 | S | Team manager |
| S94 | The system should provide a way to see the teachers workload. | 18 | Interview 18-2 | 1,2 | 3 | S | Team manager |
| S95 | The system will be able to show tasks mutations | 9 | Interview 17-3 | 1,2,5 | 3 | S | Team manager |
| S96 | The system should be able to create task mutations. | 9 | Interview 17-3 | 1,2,5 | 3 | C | Team manager |
| S97 | The system should be able to delete roles. | 27 | Interview 17-3 | 2 | 1,2,3 | C | Team manager |
| S98 | Let teachers view their own PTO's. | 16 | Interview 18-2 | 1,3 | 4,5 | M | Team manager |
| S99 | The system should be able to assign teachers to activities. | 19 | BCASE | 1,2 | 2,3 | M | Team manager |
| S100 | The system should show a list of courses that are to be given in a specific term. | 32 | Interview 17-3 | 1,5 | 2,3 | C | Team manager |
| S101 | Ability to create a term | 32 | Interview 17-3 | 1,5 | 2,3 | C | Team manager |
| S102 | Ability to display a term with it's courses | 32 | Interview 17-3 | 1,5 | 2,3 | C | Team manager |
| S103 | The system should be able to add new roles. | 27 | Interview 17-3 | 2 | 1,2,3 | C | Team manager |
| S104 | The system will export course parameters to scheduling department as Microsoft Excel files. | 42 | CASE | 6 | 3,4 | M | Team manager |
| S105 | The system will export all teacher data to scheduling department in Microsoft Excel files. | 42 | CASE | 6 | 3,4 | M | Team manager |
| S106 | The system will export terms excluding course parameters to scheduling department as Microsoft Excel files. | 42 | CASE | 6 | 3,4 | M | Team manager |
| S107 | The system will make exports to scheduling department automatically. | 42 | CASE | 6 | 3,4 | S | Consensus |
| S108 | Course parameters data should be read and write only to course managers. | 5 | Interview 18-3 | 2,5 | 2 | S | Course manager |
| S109 | Edits to a specific teacher's availability will be saved as a mutation | 36 | Interview 17-3 | 1,4 | 2,3 | C | Team manager |
| S110 | Course parameter will be read-only for team managers. | 5 | Interview 18-3 | 2,5 | 2 | S | Course manager |
| S111 | The system can provide a way for team manager to see the all availability mutations of a specific teacher | 36 | Interview 17-3 | 1,4 | 2,3 | C | Team manager |
| S112 | The user can edit teacher's availability | 36 | Interview 17-3 | 1,4 | 2,3 | C | Team manager |

## 8.2. Non-Functional Software Requirements

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Description | UR | Source | BR | BUC | Priority | Stakeholder |
| NS01 | All screens will have a title in the top describing the content in that screen in maximally three words | 28 | Interview 4-3 | 1,2,4 | 2,3,4 | C | Team manager |
| NS02 | Make the GUI user friendly so that there is no need to call in an expert (design). | 37 | BCASE | 1,2,4 | 2,3,4 | S | Team manager  Course manager |
| NS03 | The system needs to be time-saving. | 38 | PID | 1,2,4 | 2,3,4 | M | Client |
| NS04 | The system needs to be user friendly and accessible. | 37 | PID | 1,2,4 | 2,3,4 | S | Client |
| NS05 | The system needs to be less error prone. | 39 | PID | 1,2,4 | 2,3,4 | M | Client |
| NS06 | The system needs to be extendable. | 40 | PID | 1,2,4 | 2,3,4 | W | Client |
| NS07 | The system must be multi screenable. | 28 | Interview 4-3 | 1,2,4 | 2,3,5 | M | Team manager |
| NS08 | The system view-language should be Dutch. | 41 | Interview 21-3 | 1,2 | 2,3,4 | M | Client |

# 

# 

# 

# 

# 

# 9. Technical Constraints

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Stakeholder | Source |
| T01 | The scheduling department system cannot be changed. | Scheduling department | CASE |
| T02 | One system to assign teachers to tasks. | Team manager | BCASE |
| T03 | The system has to have a central database. | Summary | BCASE |
| T04 | Employees need to be able to work from home when needed. | Client | Interview 4-3 |
| T05 | The scheduling department system accepts Microsoft Excel files only. | Scheduling department | Consensus |
| T06 | Excel files sent to the scheduling department have no specific order of content | Scheduling department | Consensus |
| T07 | The application will be approachable from external locations. | Client | Interview 17-2 |

# 

# 

# 

# 

# 10. Use Case Specification Conform Priority

\*For all use cases the actor is logged in.

|  |  |
| --- | --- |
| Generate management reports | Mainflow |
| Priority | Must |
| ID | 1 |
| Description | Generates the management reports |
| Primary actor | Client |
| Secondary actor | None. |
| Preconditions | 1 None. |
| Main flow | 1. The actor chooses one of the following reports: [1.1]  - teacher hours  - summary of the costs of a course.  - course history  - number of students registrations  - available expertise and expertise that has to be present.  3. The system shows the data on the screen.  4. The actor chooses to save the report. [1.2] |
| Postconditions | The report is generated on the screen and saved. |
| Alternative flow | [1.1] Gives error: ‘Data not available’.  [1.2] PrintReport: See Alternative F low |

|  |  |
| --- | --- |
| Alternative Flow Name:PrintReport |  |
| Priority | Must |
| ID | 1.2 |
| Description | The client chooses to print the report. |
| Primary Actor | Client |
| Secondary Actor | None |
| Precondition(s) | Data is generated on the screen. |
| Alternative Flow | 1. 1.The use case starts after step 4 of the Main Flow.  2. 2. The actor chooses to print the report.  3. 3. The system sends the report to the printer software.  4. 4. The use case resumes at step 1 of the Main Flow. |
| Postcondition(s) | The report is printed. |

|  |  |
| --- | --- |
| AssignTeacherToTask | Mainflow |
| Priority | Must |
| ID | 2 |
| Description | Assign a teacher to task. |
| Primary actor | Team manager |
| Secondary actor | None |
| Preconditions | N None |
| Main flow | 1. The use case starts when the team manager wants to assign a teacher to a task.  2. The system shows a list of teachers and their expertises.  3. The actor assigns a teacher to a task. [2.1]  4. The system confirms that the teacher has been added. [2.2] |
| Postconditions | 1. 1. The system has saved the links between tasks and teachers. |
| Alternative flow | [2.1] WorkloadExceeded: The system gives an error about the teachers workload being exceeded.  [2.2] NotAssigned: The system gives an error because the teacher has not been added to the task. |

|  |  |
| --- | --- |
| DesignCourseParameter | Mainflow |
| Priority | Must |
| ID | 3 |
| Description | Designing the course parameter. |
| Primary actor | Course manager |
| Secondary actor | None. |
| Preconditions | e. None. |
| Main flow | 1. 1. The use case starts when the course manager chooses to design a course parameter.  2. 2. The system creates a new course parameter [3.1] or imports an existing one.  ffff 3. The actor creates a task.  3.1. The actor assigns activities to the task.  4. The system saves the course parameter. |
| Main flow |  |
| Postconditions | 1. 1. The course parameter is filled in.  2 |
| Alternative flow | [3.1] Course does not exist |

### 

|  |  |
| --- | --- |
| Alternative Flow name: Course does not exist |  |
| Priority | Must |
| ID | 3.1 |
| Description | The course that needs to be designed does not exist yet. |
| Primary Actor | Course manager |
| Secondary Actor | None |
| Precondition(s) | None |
| Alternative Flow. | 1. The use case starts after step 2 of the main flow.  2. The actor creates teaching units.  3. Done. |
| Postcondition(s) | The new course is created with teaching units. |

|  |  |
| --- | --- |
| **Generate management reports** |  |
| **ID** | 1 |
| **Description** | Generates the management reports |
| **Primary actor** | Client |
| **Secondary actor** | None. |
| **Preconditions** | 1.None. |
| **Main flow** | 1. Actor choses which report he wants to generate.  2. The system collects one of the following data: [1.1]  - teacher hours  - summary of the costs of a course.  - course history  - number of students registrations  - available expertise and expertise that has to be present.  3. The system shows the data on the screen.  4. The actor choses to save the report[1.2] |
| **Postconditions** | The report is generated on the screen and saved. |
| **Alternative flow** | [1.1] Gives error: ‘Data not available’.  [1.2] ReportToPrinter: See Alternative Flow |

|  |  |
| --- | --- |
| **Alternative Flow Name:ReportToPrinter** |  |
| **ID** | 1.2 |
| **Description** | The client chooses to print the report. |
| **Primary Actor** | Client |
| **Sec**on**dary Actor** | None |
| **Preconditi**on**(s)** | Data is generated on the screen. |
| **Alternative Flow** | 1. 1.The use case starts after step 4 of the Main Flow.  2. 2. The actor chooses to print the report.  3. 3. The system sends the report to the printer software.  4. 4. The use case resumes at step 1 of the Main Flow. |
| **Postcondition(s)** | The report is printed. |

|  |  |
| --- | --- |
| **AssignTeachToTasks** |  |
| **ID** | 2. |
| **Description** | Assign teachers to tasks |
| **Primary actor** | Team manager |
| **Secondary actor** | None |
| **Preconditions** | N None |
| **Main flow** | 1. The use case starts when the team manager wants to assign teachers to tasks.  2. The system shows a list of expertises and teachers.  3. While team manager has not assigned teachers to all tasks  3.1 The actor chooses how the data has to be sorted in the screen.  3.2 The system shows the teachers according to the sorted data.  3.3 The actor assigns teachers to a task and indicates how many groups of students for a project are acceptable.[2.2]  4. The system saves the links between tasks and teachers.[2.1] |
| **Postconditions** | 1. 1. The system has saved the links between tasks and teachers. |
| **Alternative flow** | [2.1] NotAssigned:The system gives an error about a teacher not being assigned to a task.  [2.2] WorkloadExceeded: The system gives an error about the teachers workload being exceeded. |

### 

|  |  |
| --- | --- |
| **DesignCourseParameter** |  |
| **ID** | 3 |
| **Description** | Designing the course parameter. |
| **Primary actor** | Course manager |
| **Secondary actor** | None. |
| **Preconditions** | 1. None. |
| **Main flow** | 1. 1. The use case starts when the course manager chooses to design a course parameter.  2. 2. The system creates a new course parameter or imports an existing one.  3. While the tasks don‘t cover the learning objectives.  3.1 The actor fills in the education activities with hours and possible preferences(teacher, classrooms).[3.1]  3.2. The system saves the course parameter.[3.2] |
| **Postconditions** | 1. 1. The course parameter is filled in.  2 |
| **Alternative flow** | [3.1] HoursNotFilled: The system gives an error about the hours that are not filled in.  [3.2] SendToTM: |

|  |  |
| --- | --- |
| **Alternative Flow name: SendToTM** |  |
| **ID** | 3.2 |
| **Description** | The course manager sends the course parameter to the team manager. |
| **Primary Actor** | Course manager |
| Secondary **Actor** | Team manager |
| **Precondition(s)** | The course parameter is filled in. |
| **Alternative Flow.** | 1. The use case starts after step 6 of the main flow.  2. The course manager sends the course parameter to the team manager.  3. The use case resumes at step 1 of the Main Flow. |
| **Postcondition(s)** | The course parameter is sent to the team manager. |

# 11 Additional Information

The Business Analysis Discipline document and the Glossary form the basis for this document. Choices made concerning the business have been based on our viewpoints described in these documents.

## 11.1 Requirements

Requirements are divided into several levels:

* Business requirements are requirements the business has given to improve the entire process. All business requirements should include content indicating the improvement made.
* User requirements are requirements from users for the new system. User requirements should be traceable to business requirements.
* Software requirements are based on the user requirements. They translate the wishes of the users (user requirements) into features and functionalities the new system should have. These requirements should be singular in form.
* Technical constraints are limits to the environment of the system.
* Business rules are rules employed by the business which have effect on the software.

## 

## 

## 11.2. Use Cases

Only a selection of important use cases have been fully documented due to time constraints.

Requirement prioritisation has been realized in accordance with the MoSCoW method. Letters shown stand for

* Must Have (item necessary in the system at all costs)
* Should Have (item necessary in the system after Must Haves)
* Could Have (item wanted but may be skipped depending on time spend on higher prioritised items)
* Would have (item with lowest priority, not needed in current version, would like in a later version)

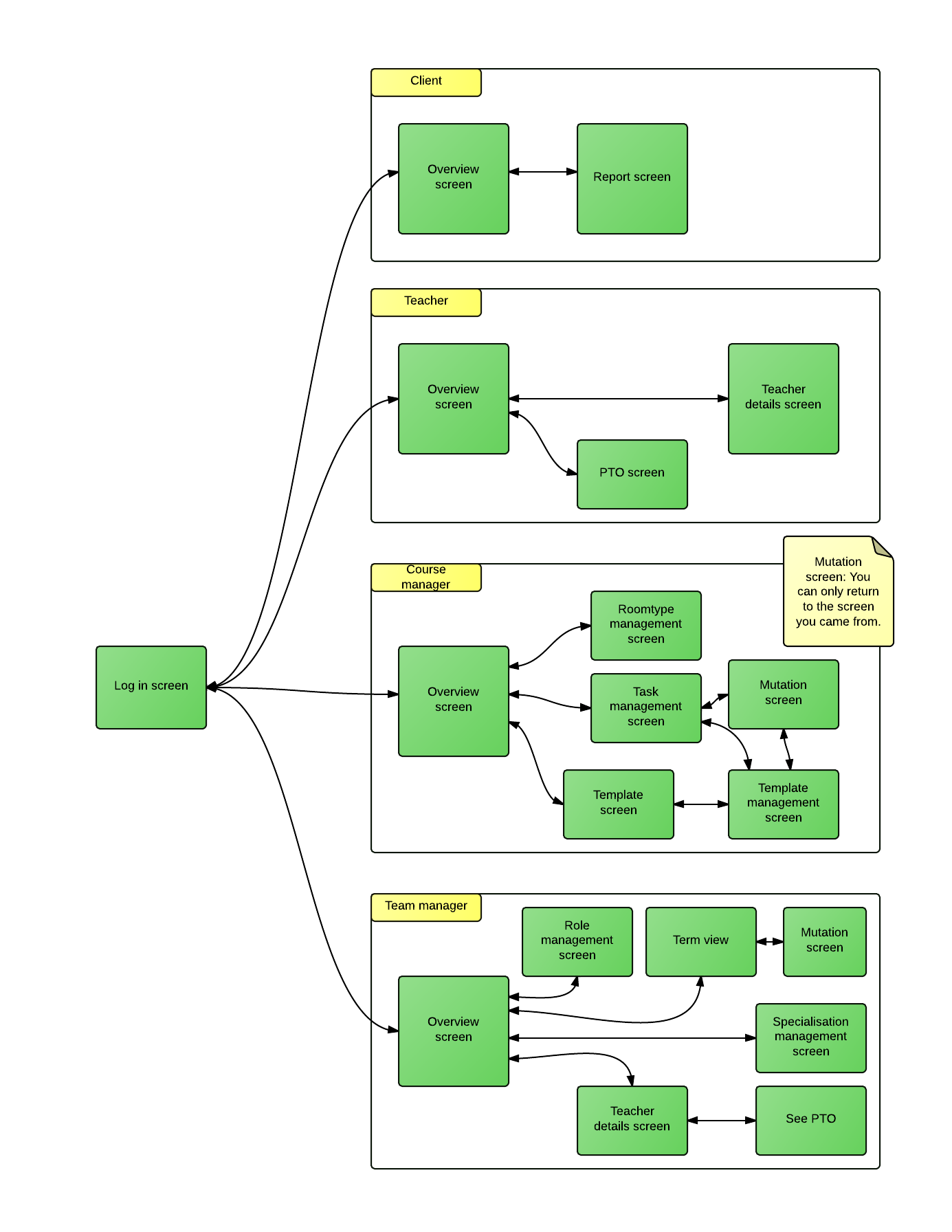
12. Verification Report

Requirements are gathered from the case materials and interviews with the stakeholders. These requirements are fully processed by the requirements analysts. Thereafter, the requirements were transformed from the business perspective into the software perspective and made SMART, as far as possible.

Lastly, traceability has been added to the requirements. This indicates the connections between: user-, software-, business requirements, business rules and business use cases. It is important to make use of traceability to make sure that all of the requirement have a business value. Otherwise, the system would not fully support the business processes.

# 13. Navigation Map

## 13.1. Navigation Map



## 13.2. User Interface Conventions

This chapter contains user interface and screen layout conventions for the application.

* **User Interface Standards**

Keep the style simple and neutral, for users not be overwhelmed by the visual noise.

This implies the use of dimmed background screens, so that the user’s primary focus will be aligned with the specific application window in use.

Error messaging will be handled by the use of more conspicuous colours, like red and orange. Also a matching error sound will ensure the user is aware of their faulty act.

* **Screen Layout**

The home- or starting screen will always be the ‘Log In’-screen, and when successfully logged in, followed by the overview screen of the corresponding role. On the overview screen there are several options, each of them displaying a new primary window. All primary windows must be resizable and movable.

* **Operation of Common Buttons**

Each individual option on the overview screen will have a letter underlined, which imply the ability to use shortcuts. These are performed by pressing and holding ALT, while selecting the letter corresponding to your option.

Buttons:

- The ok-button is always called: ‘OK’

- The cancel-button is always called: ‘Cancel’

- The refresh button will always be accompanied by an icon consisting of two arrows rotating next to each other, the general refresh icon. These button will retrieve the latest data from the database and replace the current one.

- The closing-button will always be accompanied by an ‘X’ icon. When some changes are being made to the data and not saved, a confirmation dialog will be prompted on the screen.

- The back-button will always be accompanied by a, to the left pointing arrow, icon. When some changes are being made to the data and not saved, a confirmation dialog will be prompted on the screen.

* **List Sorting**

Each list with more than 2 attributes or with excess data to display in a panel, must be equipped with the ability to scroll and sort. Sorting will be done in alphabetical order. A single click will order them from A to Z, double clicking will order them from Z to A. In case of numbers, single click will display 0 on top, increasing. Double clicking will display the highest number on top, decreasing.

The teacher’s ‘Preferences’ screen is an example of correct scrolling placement, because certain teachers have more expertises than the panel can display.

The sorting capability is necessary for the ‘LinkTeacherToTask’ screen, as it handles one of the main functionalities. This screen has multiple attributes in its table, all with too much data to fit in one window.

* **Heading Displa**y

The complexity of the user interface is relatively low compared to high end technology companies, so the need for tooltips is non-existent.

When input is entered and is too long for the input field, the cursor will be reinitiated at the last letter.

For input fields that have a maximum letter capacity

* **List of Admitted and Special Characters**

The system will be used by The Academy of IT & Media in the Netherlands. Dutch names do not contain special characters, but it is common to see Non-Dutch teachers working inside the Academy of IT & Media. Because of this, the system must support some special characters, thus UTF-8 character encoding will provide support for it.

* **Exceptions Display**

All of the errors and warning within the system will be prompt in a message box on the screen. This message box will contain name and specification of an error and name of the window it occurred in. If the warning is related to the uncertain user decision, it will result in a message box containing Continue and Abort button. Continue button will ignore the message box and continue performing an operation, and abort button will abort the operation.

Errors related to the validation of the user input will result in highlighted input form an validation error occurred. This form will be highlighted in red and tooltip will be displayed containing specification of an error. After the error is corrected, the highlighting and tooltip will disappear.

Technical errors will be stored in the database for further processing by the IT-desk of The Hague University. If for some reason, the error data can not be stored in the database, it will be stored in a log file on a workstation of an user.