S

Jan Polfers

Exposee

Software requirements specification document about a high school scheduling system.

SRS – Jan Polfers

High school scheduling system

Inhaltsverzeichnis

[1.Introdution 2](#_Toc53953401)

[1.1 Purpose 2](#_Toc53953402)

[*1.2 Scope* 2](#_Toc53953403)

[*1.3. References* 2](#_Toc53953404)

[*1.4. Overview* 3](#_Toc53953405)

[2. Overall description 3](#_Toc53953406)

[*2.1 Product perspective* 3](#_Toc53953407)

[*2.2 Product functions* 3](#_Toc53953408)

[*2.3 User characteristics* 3](#_Toc53953409)

[*2.4 Constraints* 4](#_Toc53953410)

[*2.5 Assumptions and dependencies* 4](#_Toc53953411)

[3. Specific Requierments 4](#_Toc53953412)

[*3.1. External Interface Requirements* 4](#_Toc53953413)

[*3.1.1 User Interfaces* 4](#_Toc53953414)

[*3.1.2. Hardware Interfaces* 5](#_Toc53953415)

[*3.1.3. Software Interfaces* 5](#_Toc53953416)

[*3.1.4 Communication Interfaces* 5](#_Toc53953417)

[*3.2. Functional requirements* 5](#_Toc53953418)

[*3.3 Design constraints* 6](#_Toc53953419)

[*3.4 Performance requirements* 7](#_Toc53953420)

[*3.5 Software system attributes* 7](#_Toc53953421)

[*3.5.1 Reliability* 7](#_Toc53953422)

[3.5.2 Availability 8](#_Toc53953423)

[*3.5.3. Maintainability* 8](#_Toc53953424)

[*3.5.4 Portability* 8](#_Toc53953425)

[*4. Use Cases* 9](#_Toc53953426)

# 1.Introdution

The Fontys Venlo has the problem that changes to the schedule don’t get adapted fast enough. Currently the schedule gets adapted by hand, which takes a lot of time. Often the teachers want to make changes for the upcoming week or even the next day. That results in a lot of chaos, because not everybody got informed about the changes. Or the changes couldn’t be done on time.

### 1.1 Purpose

The goal of the project is to develop a software which allows Fontys Venlo to plan the schedules. It should be possible to assign a lecturer, course and a room. The system should adapt to changes, like absent lecturers. The changes made should be delivered to the students fast enough.

## 1.2 Scope

The „Fontys Scheduler“ application lets users create schedules and displays schedules for students and lecturers. Schedules should be created for a given class. The schedule for the lecturers should get created automatically. The application is located on a webserver alongside the Fontys website. Therefore the schedule should get published on a website and on paper. Every user has already an account for the Fontys website. The account handles user authentication to view the schedules.

## 1.3. References

The use cases are mentioned in Chapter 4.

## 1.4. Overview

This document contains three chapters. The first one is an Introduction into the document and provides an overview about the software. In the second chapter provides an overview over the applications functionality and interactions with other systems. The different users (maybe stakeholders. Because PRJ3 is about stakeholder engagement) get introduced in this chapter. The third chapter defines the requirement specifications and the application interfaces. The fourth chapter explains the priorisation of the different requirements. The Appendix shows a release plan for the different requirements.

# 2. Overall description

This section gives you an overview over the application. This section will show how the application will interact with other systems and the basic functionalities of the software. It will also describe the different users and the different use cases of the given users.

## 2.1 Product perspective

The application defines the main business process of the university. Although the software defines the main business process of the customer, it doesn’t really interact with it. But the application is part of the universitys website. There is already an website in place. This website gives students access to course materials, news from the university and more. The application isn’t part of the website but can be accessed from it. Both share the same database for user authentication.

## 2.2 Product functions

The application is a website which can be entered via the already existing login credentials, from the Fontys website. But the scheduler is not part of Fontys website, it can only accessed from it. After login the student can see the current weeks schedule, for his class. If he forgot his password, he gets redirected to the already in place site to change it. He can also change the week or semester to see the schedules which lay in the future (Use case 1). After login he can also access the yearly overview, to see when exams or holiday take place (Use case 2) When a lecturer logs in he can access the same functionality’s as the student. On top of that he can report ill (Use case 3) or fill in a vacation request (Use case 4). A vacation request has to be accepted by the lecturers Boss. When a lecturer is absent his courses get cancelled for the time. When a lecturer is part of the planning committee he can apply changes to the schedule (Use cases 5-8), create a new schedule (Use case 10), apply changes to the yearly overview (Use case 13) and create a yearly overview (Use case 11).

## 2.3 User characteristics

There are three types of users who interact with the system: the student, the lecturer and the lecturer who is part of the planner group. All users manly use the same functionality, but some functionalities have restricted access. The student can see the schedule from his class and other classes. Additionally the lecturer can report ill and fill in a vacation request. Furthermore, he can see his schedule and the schedule of every class and other lecturer. Some lecturers are part of the planner group. They can create a schedule, edit it, create a yearly overview and so on.

## 2.4 Constraints

Because the scheduling system is a website, the user needs an internet connection to access it. Some courses are hosted online and some are on campus. Some students don’t live near the campus that’s why there has to be a block of travel time between the courses.

## 2.5 Assumptions and dependencies

We assume that the user has already an account on the Fontys website. Also we assume that the user has an internet connection to access the application. Furthermore we assume that the user accesses the site mostly on his computer.

# 3. Specific Requierments

This section contains all of the functional and quality requirements of the application. It gives a detailed description of the scheduling system and all its features.

## 3.1. External Interface Requirements

This section provides a detailed description about all inputs into and all outputs from the system. It also gives a description of the software and communication interfaces. It also shows the basic prototypes of the user interface.

## 3.1.1 User Interfaces

There are three types of users and the functionalities the users have access to build up on another. The order is the following: student, lecturer and planner.

The user interacts with the system over a web portal. There the user sees his personal schedule for this week. If he clicks on a week in the calendar he can view a schedule from another week. For a student the menu on the right only has the option to choose a class from a drop down menu or select a week from a calendar.

If the user is a lecturer he can report ill and fill in a vacation request, on top of the base functionalities. If he clicks on the schedule drop down menu, he can access other lecturers schedules from there. When he reports ill or fills in a vacation request a pop up menu opens up. When he fills in a vacation request he can enter the start date and the end date. When the lecturer reports ill, he can either report ill for today or for some amount of days. Another one of these extra functionalities is a site where he can see his vacation request history. He accesses the site from a button on the left menu. The site displays all vacation requests and the status of them. If the lecturer is allowed to process vacation requests he has the option to see vacation requests of his employees. He then can accept or decline these requests. If the user is part of the planner group he can create a schedule, edit a schedule, create a yearly plan, change room capacity, change the lecturer of a course, change the room of a course, make lecturer a planner, plan an exam and change the time of an course. When the user clicks on the “create a schedule” button on his menu. He gets redirected to a site displaying a schedule template. There he selects a semester from a drop down menu. He can select a class in an drop down menu on the left side. In another drop down menu he can select a course. When left clicks and holds on the schedule template, he marks several hours of a day. After releasing the hours are still marked and contain the name of the course. When the user right clicks on the marked course a menu pops up. There he can enter the room in a drop down menu and tick a box if the course is online or offline. He repeats this steps until the schedule is finished. To publish the schedule the user clicks on submit. When he wants to edit a schedule, he clicks on “edit schedule” in his menu. He gets redirected to another site. There he selects a class and semester in a drop down menu. Then the schedule from the given class from the selected semester gets loaded. When the user clicks on a scheduled course a pop up menu opens. In this menu he can change lecturer, room and time of the course. To create a course the user clicks on “create a course” on his menu. Then a pop up menu opens where he can enter a lecturer from a drop down menu, enter a name and description for the course. After that he clicks on submit to publish the schedule.

On the menu on the left the user can click on “create a yearly overview”. Then the user gets redirected to another site. When the user clicks on the three bars a menu slides up to the right. On this menu the user can select a year from a calendar. This calendar pops up on the site. There the user can select days or weeks with a click and hold of the mouse. The user can now assign these marked sections with a right click. He either can assign them as holiday or exam time. This is done in a pop up menu that pops up when the user right clicks on a marked section.

## 3.1.2. Hardware Interfaces

The web application is not bound to any specific hardware. Therefore, it does not have any designated hardware interfaces.

## 3.1.3. Software Interfaces

The scheduling system communicates with the universities website and the database. When transitioning form the universities website to the scheduling system, the login credentials get transferred. The scheduling system can only run read operations on the university’s database. To store the information’s from the scheduling system, there is a database attached to it. There the scheduling system has rights to both read and write.

## 3.1.4 Communication Interfaces

The communication between the university’s website and the scheduling system important, since the authentication is handled that way.

## 3.2. Functional requirements

|  |  |
| --- | --- |
| ID | FR1 |
| Title | Transfer from universitys website. |
| Description | The login credentials get transferred from the universitys website to the scheduling website. |
| Rational | In order to login to the scheduling site |
| Dependency | None |
| Corresponding  Use Cases | All |
| Priority | High |

|  |  |
| --- | --- |
| ID | FR2 |
| Title | Functionalitys menu |
| Description | The menu on the left side shows the different functionalitys available. Based on the user there are more functionalities. For example all the functionalities the planner has access to. |
| Rational | In order to access all functionalities |
| Dependency | None |
| Corresponding  Use Cases | UC3, UC4, UC5, UC6, UC7, UC8, UC10, UC13, UC11 |
| Priority | High |

|  |  |
| --- | --- |
| ID | FR3 |
| Title | Lecturers schedule |
| Description | Only lecturers can see the schedule from other lecturers. |
| Rational |  |
| Dependency | None |
| Corresponding  Use Cases | UC1 |
| Priority | Low |

|  |  |
| --- | --- |
| ID | FR4 |
| Title | Overwrite existing schedule |
| Description | When changes get applied for a schedule or then a schedule gets overwritten. These changes should get published onto the viewable schedule. |
| Rational | In order to apply changes. |
| Dependency | None |
| Corresponding  Use Cases | UC5, UC6, UC7, UC8, UC10 |
| Priority | High |

|  |  |
| --- | --- |
| ID | FR5 |
| Title | Cancel courses |
| Description | When a lecturer reports ill or when his vacation request gets granted. The courses for the time he is absent, should get cancelled. |
| Rational | In order to handle it when a teacher is absent. |
| Dependency | None |
| Corresponding  Use Cases | UC3, UC4 |
| Priority | High |

|  |  |
| --- | --- |
| ID | FR6 |
| Title | Block of travel time |
| Description | When there is a switch between online and offline courses in the schedule. There should be a block of travel time added. |
| Rational | In order to help student and lecturers travel home and be on time for online courses. |
| Dependency | FR4 |
| Corresponding  Use Cases | UC5, UC6, UC7, UC8, UC10 |
| Priority | Medium |

## 3.3 Design constraints

This section includes the design constraints on the software.

|  |  |
| --- | --- |
| ID | DC1 |
| Title | Scalable hard drive space. |
| Description | Because we can know how much space the system takes on a disk. The storage should be scalable. |
| Rational | In order to pay to much for storage. |
| Dependency | None |
| Corresponding  Use Cases | None |
| Priority | Medium |

|  |  |
| --- | --- |
| ID | DC2 |
| Title | Ensure connection capacity. |
| Description | Because the university is growing, the website and database should handle at least 500 simultaneous connections. |
| Rational | In order to ensure that enough students can access the website at the same time. |
| Dependency | None |
| Corresponding  Use Cases | None |
| Priority | Medium |

|  |  |
| --- | --- |
| ID | DC3 |
| Title | Font type. |
| Description | The same font type should be used across the website. A few exceptions are tolerated. |
| Rational | In order to ensure readability and a consistent design. |
| Dependency | None |
| Corresponding  Use Cases | None |
| Priority | Medium |

## 3.4 Performance requirements

This section contains the constraints that ensure the performance of the website.

|  |  |
| --- | --- |
| ID | PR1 |
| Title | Low response time |
| Description | The response time should be below two milliseconds. |
| Rational | In order to ensure that the website works properly. |
| Dependency | None |
| Corresponding  Use Cases | None |
| Priority | Medium |

|  |  |
| --- | --- |
| ID | PR2 |
| Title | Easy to understand sites. |
| Description | The sites should be easy to understand for every user. |
| Rational | In order to ensure that every user knows how to use the website. |
| Dependency | None |
| Corresponding  Use Cases | None |
| Priority | Medium |

## 3.5 Software system attributes

The requirements in this section specify the reliability, availability, maintainability and portability.

## 3.5.1 Reliability

|  |  |
| --- | --- |
| ID | R1 |
| Title | Load the right schedule |
| Description | The system should always load the right schedule. |
| Rational | In order to ensure that the website works properly. |
| Dependency | None |
| Corresponding  Use Cases | None |
| Priority | High |

|  |  |
| --- | --- |
| ID | R2 |
| Title | Display the right time and courses. |
| Description | The system should always load the right courses on the right time. |
| Rational | In order to ensure that the website works properly. |
| Dependency | None |
| Corresponding  Use Cases | None |
| Priority | High |

### 3.5.2 Availability

|  |  |
| --- | --- |
| ID | DC3 |
| Title | The website should be available 24/7 |
| Description | The users should be able to access the website 24/7. |
| Rational | In order to ensure that the website works properly. |
| Dependency | None |
| Corresponding  Use Cases | None |
| Priority | High |

## 3.5.3. Maintainability

|  |  |
| --- | --- |
| ID | M1 |
| Title | Extendable |
| Description | New features or other functionalities should get added easily. To ensure that the code has to allow a high level expandability. |
| Rational | In order to ensure new features to get added later on. |
| Dependency | None |
| Corresponding  Use Cases | None |
| Priority | Medium |

## 3.5.4 Portability

|  |  |
| --- | --- |
| ID | P1 |
| Title | Access on mobile |
| Description | The website should get displayed properly on mobile phones and tablets. On android and IOS. |
| Rational | In order to ensure a consistent design. |
| Dependency | None |
| Corresponding  Use Cases | None |
| Priority | Medium |

|  |  |
| --- | --- |
| ID | P2 |
| Title | Access from all browsers. |
| Description | The website should get displayed properly on the most commen browsers. |
| Rational | In order to ensure a consistent design. |
| Dependency | None |
| Corresponding  Use Cases | None |
| Priority | Medium |

# 4. Use Cases

|  |  |
| --- | --- |
| Name | View schedule in the future. |
| Actor | User |
| Description | The user wants to see a schedule from another week. |
| Precondition | The user is logged in. |
| Scenario | 1. The system displays the users personal schedule. 2. The user accesses the calendar and selects a week. 3. The system displays the schedule from the selected week. |
| Result | The user sees a schedule from another week. |

Use case 1

|  |  |
| --- | --- |
| Name | View yearly overview. |
| Actor | User |
| Description | The user wants to see the yearly overview. |
| Precondition | The user is logged in. |
| Scenario | 1. The system displays the users personal schedule. 2. The user wants to see the yearly overview. 3. The system displays the yearly overview. 4. The user selects a different year. 5. The system displays the yearly overview from a different year. |
| Result | The system displays the yearly overview from another year. |

Use case 2

|  |  |
| --- | --- |
| Name | Report ill |
| Actor | Lecturer |
| Description | The lecturer reports ill for one day or multiple days. |
| Precondition | The lecturer is logged in. |
| Scenario | 1. The system displays the users personal schedule. 2. The lecturer reports ill. 3. The system cancels the lecturers courses for the time he is ill. |
| Result | The courses from the lecturer are cancelled. |

Use case 3

|  |  |
| --- | --- |
| Name | Take holidays |
| Actor | Lecturer and lecturers’ boss. |
| Description | The lecturer creates a vacation request. |
| Precondition | Both actors are logged in. |
| Scenario | 1. The system displays the users personal schedule. 2. The lecturer fills in a vacation request. 3. The system displays a possibility to fill in a vacation request. 4. The lecturer enters the time when he wants to go on vacation. 5. The lecturer submits the request. 6. The system checks if the date is two weeks in advance. 7. The system sets the status on waiting for approval. |
| Boss | 1. The lecturers boss wants to see request waiting for approval. 2. Thy system displays every request waiting for approval. 3. The lecturers boss clicks on approve. 4. The system marks the request as approved. |
| Exceptions | * 1. When the date is not more than two weeks away, the system throws a message informing the user about it. |
| Result | The lecturer created a vacation request and it got accepted. |

Use case 4

|  |  |
| --- | --- |
| Name | Change the room of a course. |
| Actor | Lecturer who is part of the planner group. |
| Description | The lecturer changes the room from a course. |
| Precondition | The lecturer is logged in. |
| Scenario | 1. The system displays the users personal site. 2. The user wants to apply changes to a schedule from a given class. 3. The system displays the schedule in a mode to apply changes. 4. The user selects a different room for a course. 5. The system checks the class size and displays possible rooms. 6. The user selects a room. 7. The system applies the changes to schedule in the “changes” mode. 8. The user clicks on publish schedule. |
| Result | The room of the selected course gets changed. |

Use case 5

|  |  |
| --- | --- |
| Name | Change the time of a course. |
| Actor | Lecturer who is part of the planner group. |
| Description | The lecturer changes the time of a course. |
| Precondition | The lecturer is logged in. |
| Scenario | 1. The system displays the users personal site. 2. The user ´wants to apply changes to a schedule from a given class. 3. The system displays the schedule in a mode to apply changes. 4. The user selects a different time for a course. 5. The system checks if the selected room is occupied on the selected time. 6. The system applies the changes to the temporary schedule 7. The system displays the temporary schedule. 8. The user publishes the schedule. 9. The changes get applied to the schedule |
| Exception | The room is occupied  5.1. The system gets the class size.  5.2. The system checks available rooms with the given capacity and on the given  time.  5.3. The user selects a new room. |
| Result | The time of a course got changed. |

Use case 6

|  |  |
| --- | --- |
| Name | Change the capacity of a room. |
| Actor | Lecturer who is part of the planner group. |
| Description | The lecturer changes capacity of a room. |
| Precondition | The lecturer is logged in. |
| Scenario | 1. The system dispalys the users personal site. 2. The user wants to change the capacity of a room. 3. The system displays all rooms. 4. The user selects a room. 5. The system displays the name and the capacity. 6. The user changes the capacity. 7. The system saves the new capacity. |
| Result | The selected room has now a different capacity. |

Use case 7

|  |  |
| --- | --- |
| Name | Change the assigned lecturer of a course |
| Actor | Lecturer who is part of the planner group. |
| Description | The user changes the lecturer from a course. |
| Precondition | The lecturer is logged in. |
| Scenario | 1. The system displays the users personal site. 2. The user wants to see all courses. 3. The system displays all courses. 4. The user selects a course. 5. The system displays the course. 6. The user wants change the lecturer. 7. The system changes the lecturer of the given course. |
| Result | The course has a different lecturer now. |

Use case 8

|  |  |
| --- | --- |
| Name | Create a new course |
| Actor | Lecturer who is part of the planner group. |
| Description | A new course gets created. It contains a name, description and a lecturer. |
| Precondition | The lecturer is logged in. |
| Scenario | 1. The system displays the users personal site. 2. The user wants to create a new course. 3. The system displays the course template. 4. The user enters a name. 5. The system saves the name. 6. The user enters a description. 7. The system saves the description. 8. The user enters a lecturer. 9. The system saves the lecturer. 10. The user clicks on save. 11. The system saves the new course. |
| Result | A new course got created. |

Use case 9

|  |  |
| --- | --- |
| Name | Create a schedule |
| Actor | Lecturer who is part of the planner group |
| Description | The lecturer creates a schedule for a given class. |
| Precondition | The lecturer is logged in. |
| Scenario | 1. The system displays the users personal site. 2. The user wants to select a new schedule. 3. The system displays an empty schedule template. 4. The system displays the courses.   3.1 The course can get edited. (See use case)   1. The user selects a date (Monday to friday) and a time. 2. The system checks class size and compares it to the room size. 3. The system displays free rooms for the given time and day. 4. The user selects a room. 5. Repeat steps 3 – 8. 6. The user has added a course. 7. The system adds course to the temporary schedule. 8. Repeat steps 3-11. 9. The user selects semester to publish the schedule. 10. The system publishes the schedule.   13.1 If the current semester is selected, the system marks the current schedule as old. |
| Result | A new schedule is created and is published. |

Use case 10

|  |  |
| --- | --- |
| Name | Plan yearly overview. |
| Actor | Lecturer who is part of the planner group |
| Description | The planner creates a yearly overview |
| Precondition | The user is logged in. |
| Scenario | 1. The system displays the users personal schedule. 2. The user wants to create a yearly overview. 3. The system displays the yearly overview template. 4. The user marks a week. 5. The system selects a week. 6. The user can either mark the week as holiday or exam. 7. The system marks the selected as holiday or exam. 8. Repeat steps 3 to 7. 9. The user publishes the yearly overview. 10. The system deletes all courses for the given weeks. |
| Result | A yearly overview got created and all courses for the selected weeks got cancelled. |

Use case 11

|  |  |
| --- | --- |
| Name | Make lecturer a planner |
| Actor | Lecturer who is already a planner. |
| Description | A planner gives another lecturer the right to plan schedules. |
| Precondition | The user is logged in |
| Scenario | 1. The system displays the users personal schedule. 2. The user wants to add another planner. 3. The system displays the lecturers. 4. The user selects a lecturer. 5. The system adds the lecturer to the planner group. |
| Result | A lecturer got added to the planner group. |

Use case 12

|  |  |
| --- | --- |
| Name | Plan exams |
| Actor | User who is part of the planning group |
| Description | During exam weeks are no courses scheduled. Exams need to get scheduled. |
| Precondition | User is logged in and part of the planner group. |
| Scenario | 1. The system displays the users personal schedule. 2. The user wants to plan exams 3. The system displays all classes. 4. The user selects a class. 5. The system displays the different exam weeks. 6. The user selects a week. 7. The system displays a scheduling template. 8. The user selects the day and time. 9. The system marks the day and time in the template. 10. The user enters the name of the exam and the lecturer. 11. The system checks if the lecturer is available 12. The system displays a list of available rooms. 13. The user selects a room. 14. The system checks if the room is big enough for the class. 15. The user submits changes. 16. Repeat steps 7-16. 17. The user clicks publishes the schedule. 18. The system enters the schedule into the system. |
| Exceptions | The lecturer is not availabile  11.1. The system throws a message that the lecturer is not availabil.  11.2. The system displays the menu to enter an exam.  11.3. The user enters a different time.  The room is to small  14.1. The system throws a message that the room is too small.  14.2. The system displays the exam menu.  14.3. The user enters a new room |
| Result | The exams for a class got scheduled. |

Use case 13