



# Software Requirements Specification



Anders Deliens  
4.0

# 1 Introduction

## 1.1 Purpose

The purpose of this SRS is to provide a detailed description of all the requirements for project Xiast. It will include a list of constraints, features and (user)interfaces.

This document is intended for the members of the development team behind Xiast as well as Professor Ragnhild Van Der Straeten and assistant Jens Nicolay.

## 1.2 Scope

The application that will be discussed in this document is an online scheduling-tool for university classes, named Xiast (short for: Xiast is a scheduling tool). It will compute the optimal distribution of classes, based on certain constraints, which are given by administrators and/or teachers. Students and teachers will be able to check their personal schedule online and via android smartphones. Teachers will be able to add certain requirements to the classes and will be able to cancel a class last-minute. The goal of this project is to make a scheduling-tool that computes rosters and displays them in a user-friendly way.

## 1.3 Definitions, acronyms and abbreviations

- program-administrator = person who assigns courses to teachers, is able to configure every detail of the scheduling constraints (courses, classrooms, teacher, number of students, theory or practical classes...)
- teacher = person who defines the details of all the classes assigned to him ( assistants, required resources and facilities, assistants... )
- student = someone who can register for programs and courses
- user = program administrator, teacher or student
- guest = someone who is not yet logged in and identified as user
- program = combination of multiple courses (provided by the university)
- WPO = lab sessions or exercise lessons
- HOC = theory lessons

## 1.4 References

IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications, *IEEE Computer Society*, 1998

## 1.5 Overview

Section 2 will cover an overview of the general requirements. Using scenarios it will describe how the application should work. The functions that will be implemented and the user characteristics will be portrayed there, next to the constraints the application is bound by. The requirements will be further analysed and described in-depth in section 3.

This document largely follows the IEEE Std 830-1998.

## 1.6 Revision history

Version	Date	Description
0	28/10/2013	First draft of SRS
1	13/11/2013	Completion of initial version
2	08/12/2013	Adapted to feedback of initial version
3	03/03/2014	Update of SRS + images
4	04/05/2014	Update Final Version

## 2 Overall description

### 2.1 Product perspective

Xiast is an independent open-source product and will be free to use by anyone. Xiast aims to provide more personally modifiable rosters than other applications on the market, using a user-friendly interface. Alongside the web-application, there will be a mobile android application with some specific features and support for mobile users. Both the web and mobile application will use data stored on a database-server (Wilma).

## 2.2 Product functions

Xiast will accommodate three types of users: students, teachers and program-administrators. Each of these types has their own rights and own functions.

Every user will have access to the home screen, which will contain information about Xiast, the team, the VUB... Users will also be able to change the language of Xiast. After filling in their username and password, Xiast will check the authenticity of the user and will determine whether the user is a student, a teacher or a program-administrator and give the user the appropriate rights.

Students can register (and unregister) for a program (or different courses). After a certain amount of time, no further registration will be allowed. The students will be able to check their personal schedule online and on their smartphone. They will be notified of any last-minute changes made to their schedule.

Teachers get certain courses assigned to them by program-administrators. They will be able to check their roster online and on their smartphone. They will be able to submit their specifications regarding their courses (like maximum number of students, or the need of an overhead-projector, days they won't be able to teach...). Teachers can perform scheduling for their own courses and can mark these with a flag, so the program-administrator can see that they approve of this roster. If a teacher, for some reason, cannot make it to his scheduled class, he can notify the application, which will update the rosters for all students attending said class. Teachers can request certain changes to their personal roster, but only a program administrator is able to actually change the roster and the details of it.

Program administrators are in charge of a bundle of courses, which combined together form a standard program. They specify which courses are part of the program, how many courses there will be, which courses are obligatory and which aren't... They also assign a teacher to each course. They can add or delete existing courses and are able to modify every constraint regarding the scheduling. They can create new courses as well.

The program administrator can ask Xiast to perform a scheduling. Xiast will compute and display the best roster possible given said constraints. If there are any overlaps, Xiast will highlight them and the program administrator has the choice to either make a change manually or let Xiast try come up with alternative solutions. After any (manual) changes are made, administrators have to give their fiat before the roster gets picked and is

made visible for all other users. When a new schedule is being made and an overlap between courses occurs, Xiast will try to adjust the scheduling of certain courses by ignoring their lowest priority constraints. By ignoring these low priority constraints, a new roster without overlaps may emerge. The administrator of the course which has had a change in constraints or which needs to be rescheduled will receive a notification. A program administrator may choose to change the scheduling or constraint of the courses manually or even keep the overlap.

### **2.3 User Characteristics**

Users don't need any particular experience or expertise to use this application, just some basic knowledge of how to work with a computer and internet (on smartphones).

### **2.4 Constraints**

Since the application must be able to run on Android devices, there is a limitation to memory. Xiast must be able to store its data in an efficient way, so that it runs smoothly, even on devices with little internal memory and low processor speeds. The schedules should still be nicely displayed on smaller smartphone screens. If the mobile network connection is unstable, the application should still be able to display the schedule, without having to reload everything.

The application has to have a Wilma back-end and a browser (computer or android smartphone) front-end.

Only Java, JavaScript, HTML, CSS, SQL and associated open-source frameworks and libraries may be used as programming language. Only free and open-source software may be used for this product.

The user interface must be appealing and simple.

### **2.5 Assumptions and dependencies**

To use this application, an internet connection is required. This connection is necessary for the application to fetch data from the server. If a user doesn't have access to internet, he will still be able to check previously saved schedules.

## 3 Specific Requirements

### 3.1 External interfaces

The only link with an external system is the one to the Wilma back-end server, which contains all the scheduling data.

### 3.2 Functional Requirements

This next section will contain a detailed list of all functional requirements that need to be implemented, divided into sections for every user class. Every requirement will have a unique ID, which makes it easier to make references. The ID consist of FR (=Functional Requirement) followed by the first letter of the user class (G=Guest; U=User; S=Student; T=Teacher and P=Program-administrator) and then a number.

#### User Class 1: Guest

##### Functional Requirement FRG1

**TITLE:** Log in

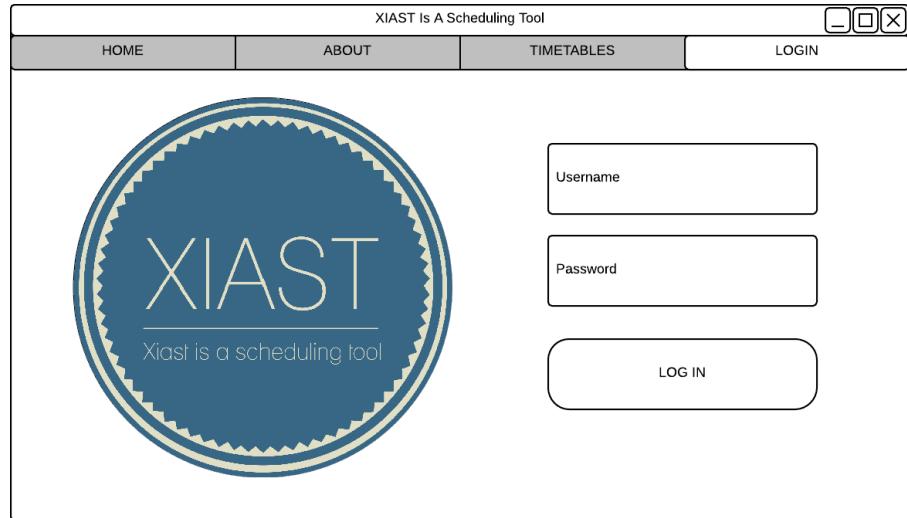
**DESCRIPTION:** When guests correctly enter the username and password of an existing account, they will be logged in, giving them the appropriate rights.

**SCENARIO:** Guests click on the log in tab. The log in screen is showing. Guests enter their username and password and click the "log in" button.

The system logs the guest in as user, gives the appropriate rights and shows the appropriate "Home" screen.

**EXCEPTIONS:** At any point in this scenario, the guest can click on any of the other tabs on the home screen. In this case, the guest will be redirected to the appropriate screen. If a guest enters the wrong password or username, the system will display an error message stating that either the password was incorrect or the username doesn't exist. Guests will then be redirected to the log in screen, where they can try to log in again.

**POSTCONDITION:** The guest is now logged in as a user and the system is showing the user's "Home" screen.



## Functional Requirement FRG2

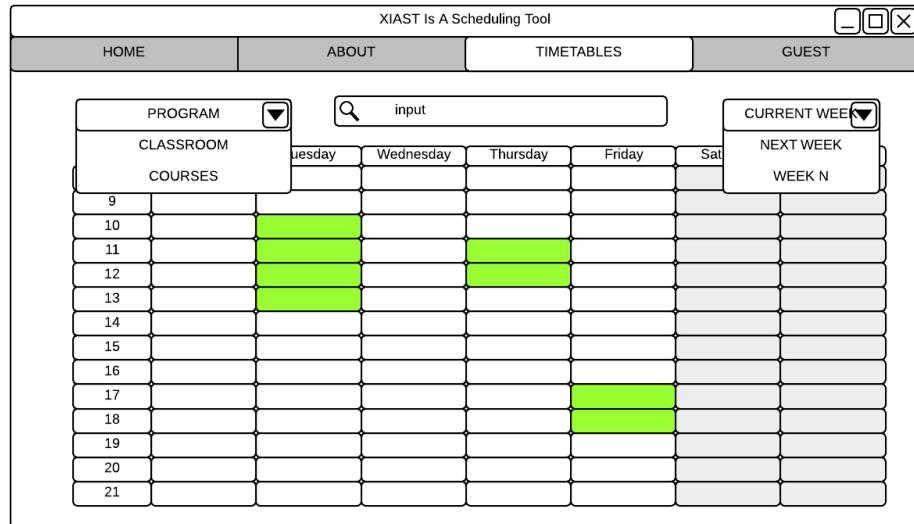
**TITLE:** Timetables

**DESCRIPTION:** Guests can search through all existing timetables (courses, programs and classrooms) and display their schedule

**SCENARIO:** Guests can click on the timetable-tab, choose a certain program, course or classroom - and a certain period, after which Xiast will display the proper schedule.

**EXCEPTIONS:** None

**POSTCONDITION:** The appropriate schedule is showing.



### Functional Requirement FRG3

**TITLE:** Choosing language

**DESCRIPTION:** Guests can choose their language by clicking on a the language button, representing the language of their choosing.

**SCENARIO:** Guests click on a language button.

**POSTCONDITION:** All pages of the application will be translated to the selected language.

**User Class 2: User** All the functional requirements that follow, will apply to all three types of users: students, teachers and program-administrators.

### Functional Requirement FRU1

**TITLE:** Log out

**DESCRIPTION:** Users can log out, becoming guests.

**PRECONDITION:** User is logged in.

**SCENARIO:** Whenever users click the log out button, they will be asked to confirm this decision and after confirmation, they will be logged out and returned to the home screen as a guest.

**EXCEPTIONS:** If guests click cancel after clicking the log out button, they

stay logged in, remaining on the same page they were before clicking the log out button. If the user clicks another tab after clicking the log out button, the system will interpret this as a cancel log out command.

**POSTCONDITION:** The user is logged out and is now a guest.

### **Functional Requirement FRU2 (FRG2)**

**TITLE:** Timetables

**DESCRIPTION:** Users can search through all existing timetables (courses, programs and classrooms) and display their schedule.

**SCENARIO:** Users can click on the timetable-tab, choose a certain program, course or classroom - and a certain period, after which Xiast will display the proper schedule.

**EXCEPTIONS:** None

**POSTCONDITION:** The appropriate schedule is showing.

### **Functional Requirement FRU3**

**TITLE:** Viewing personal schedule.

**DESCRIPTION:** Users can view their personal schedule and filter the display of the schedule with different modifiers.

**PRECONDITION:** None

**SCENARIO:** Users click on the home button and will be redirected to their home page. Their personal schedule will be displayed here. They can add modifiers to denote the period they want to display.

**EXCEPTIONS:** When there is no schedule to be shown, a message will inform the users. When no specifications are made for the time to be displayed, the current week will be shown.

**POSTCONDITION:** The correct personal schedule is being displayed.

XIAST Is A Scheduling Tool							
HOME	ABOUT	TIMETABLES		CURRICULUM INFO		MY PROFILE	
<b>CURRENT WEEK ▾</b>							
NEXT WEEK	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
WEEK N							
10		COURSE 1					
11		ROOM 52					
12							
13							
14							
15				COURSE 3			
16				ROOM 42			
17							
18							
19							
20							
21							

### Functional Requirement FRU4

**TITLE:** Set default language

**DESCRIPTION:** Users can set their default language.

**SCENARIO:** A user clicks on the language button. This choice will be saved, so users don't have to change the language every time they log in.

**POSTCONDITION:** All pages of the application will be translated to the selected language. When the user logs in, after being logged out, his preferred language will automatically be used.

### User Class 3: Student

### Functional Requirement FRS1

**TITLE:** View profile + courses

**DESCRIPTION:** Students can take a look at their profile.

**PRECONDITION:** None

**SCENARIO:** When students click on the "My Profile" button, they will be redirected to a page where a list of their personal details will be displayed, alongside with a list of all their registered courses.

**EXCEPTIONS:** If the student should not have any registered courses, a

message will notify the student.

**POSTCONDITION:** The correct list of registered courses and personal details of the student are displayed.

The wireframe shows a top navigation bar with tabs: HOME, ABOUT, TIMETABLES, CURRICULUM INFO, and MY PROFILE. The MY PROFILE tab is active. Below the navigation is a placeholder for a user profile picture. To the right of the picture is a section labeled "PERSONAL DATA". Inside this section are two checked checkboxes: "COURSE 1" and "COURSE 3". At the bottom right of the page is a "SAVE CHANGES" button.

### Functional Requirement FRS2

**TITLE:** Search for course or program

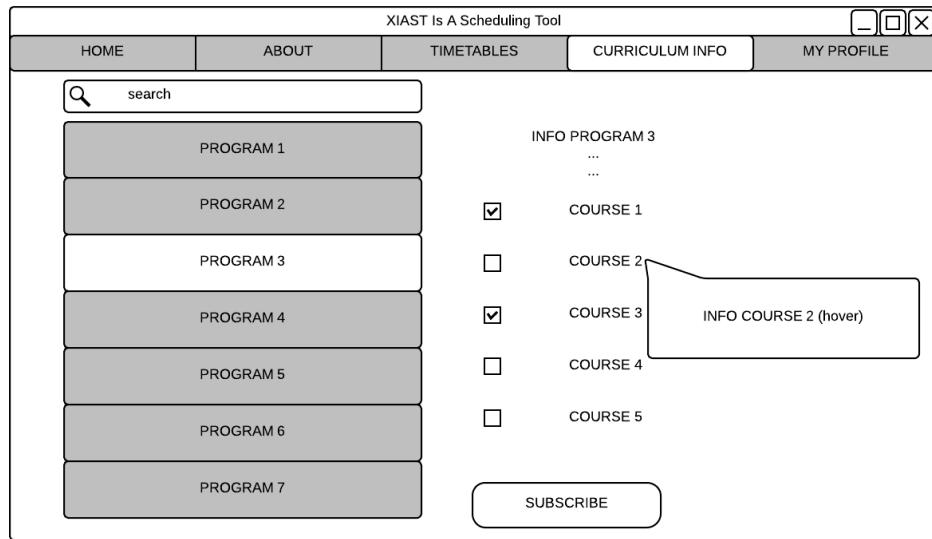
**DESCRIPTION:** Students can search a course or program

**PRECONDITION:** Students must be on the "Curriculum Info" page in order to perform a search.

**SCENARIO:** The student can type in a query inside the search box and a list of all programs matching the query will be displayed. After selecting the desired program, a list of all courses inside this program will be displayed as well as a brief description of the program. When selecting a course, XIAST will display the course information.

**EXCEPTIONS:** When there are no courses/programs that match the search criteria, a notification will alert the student.

**POSTCONDITION:** A list matching the search criteria will be displayed.



### Functional Requirement FRS3

**TITLE:** Registering for a course or program

**DESCRIPTION:** Students can register for courses and programs

**PRECONDITION:** Students must be on the "Curriculum Info" page in order to register for courses or programs.

**SCENARIO:** If students want to register for a program, they search for this program using the search box. When the program is found, the students can select the courses of this program they want to attend. A pop-up will ask for confirmation and after the confirmation the student will be registered.

**EXCEPTIONS:** When there are no courses that match the search criteria, a notification will alert the student. If a student presses cancel he will not be registered for the course.

**POSTCONDITION:** The student is now registered for the course/program.

### Functional Requirement FRS4

**TITLE:** Unregister for a course or program

**DESCRIPTION:** Students can unregister for courses and programs.

**PRECONDITION:** Students must be on their profile page and must be registered to at least one course.

**SCENARIO:** If students want to unregister for a course, they search for this course in the list of their registered courses. When the course is found, students can uncheck boxes next to the courses. A pop-up will ask for confirmation and after the confirmation the students will be unregistered.

**EXCEPTIONS:** When a student presses cancel he will not be unregistered.

**POSTCONDITION:** The student is no longer registered for the course/program.

#### User Class 4: Teacher

##### Functional Requirement FRT1

**TITLE:** View profile + courses

**DESCRIPTION:** Teachers can take a look at their profile.

**PRECONDITION:** None

**SCENARIO:** When teachers click on the "My Profile" button, they will be redirected to a page where a list of their personal details will be displayed, alongside with a list of all the courses assigned to them.

**EXCEPTIONS:** If there are no courses assigned to teachers, a message will notify them of that.

**POSTCONDITION:** The correct list of all assigned courses and personal details of the teacher are displayed.

XIAST Is A Scheduling Tool

HOME ABOUT TIMETABLES SEMI-SCHEDULING MY PROFILE

PERSONAL DATA

COURSE 1  
COURSE 2  
COURSE 3

INFO COURSE 2  
EDITABLE COURSE INFO  
FACILITIES HOC/WPO

beamer  
 overhead projector  
 ...

WEEK AGENDA TO SPECIFY HOURS AND DAYS AVAILABLE FOR TEACHING

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							

SAVE CHANGES

### **Functional Requirement FRT2**

**TITLE:** Last-minute cancelling

**DESCRIPTION:** Teachers can cancel a scheduled course last-minute.

**PRECONDITION:** A course must be scheduled and the teacher must go to his "Home" screen.

**SCENARIO:** When teachers can't make it to a certain class due to unexpected circumstances, they can cancel that course by going to the "Home" screen. They search for the course that needs to be cancelled in the schedule and click it. A pop-up will appear asking the teacher to confirm cancellation or to go back without cancelling. This cancellation is immediately visible for all the students attending this course.

**EXCEPTIONS:** When teachers abort the cancellation, nothing happens and they are returned to the "Home" screen.

**POSTCONDITION:** The course is no longer scheduled and this change is visible in all schedules containing this course.

### **Functional Requirement FRT3**

**TITLE:** Edit details of course.

**DESCRIPTION:** Teachers can change the details of a course.

**PRECONDITION:** A course must be assigned to the teacher.

**SCENARIO:** Teachers go to the "My Profile" screen and can click on any of the courses assigned to them. This will bring forth another screen with different sections. One of these sections will allow teachers to adjust the description of the course. Another section will contain a list of possible facilities needed for the course. Teachers select whether they want to adjust the facilities of the WPO or the HOC (since these can differ quite a lot). After making changes to either the description or the facilities, the teachers have to hit a save button.

**EXCEPTIONS:** The teacher can hit the cancel adjustments button, so no changes will occur.

**POSTCONDITION:** The course description, facilities and preferences are changed and updated correctly.

### **Functional Requirement FRT4**

**TITLE:** Edit preferences.

**DESCRIPTION:** Teachers can change their preferences.

**PRECONDITION:** None.

**SCENARIO:** Teachers go to the "Semi-Scheduling" screen. There they will find a form where they can fill in their preferences for the scheduling of their courses: the days they are available and their preferred classrooms.

**EXCEPTIONS:** The teacher can hit the cancel button, so no changes will occur.

**POSTCONDITION:** The teacher's preferences are changed.

### Functional Requirement FRT5

**TITLE:** Teacher semi-scheduling

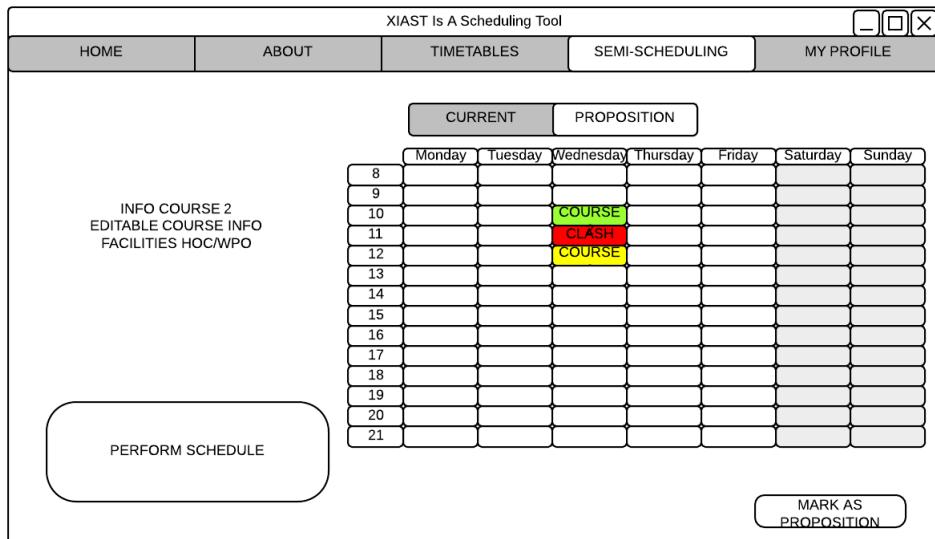
**DESCRIPTION:** Teachers can perform a scheduling of their own courses and mark one as a "proposal".

**PRECONDITION:** Courses must be assigned to the teacher.

**SCENARIO:** Teachers can go to their "Semi-Scheduling" screen, where their current schedule will be shown. When the teacher presses the "perform scheduling" button, Xiast will perform a scheduling of all the courses assigned to the teacher, taking the preferences and facilities of each course into account. The new roster will be displayed and the teacher can cycle between his current schedule and the new schedule by clicking on either the "current" button or the "proposal" button. If the scheduling doesn't contain any conflicts or overlaps, the teacher can mark this schedule as a "proposal".

**EXCEPTIONS:** If the newly scheduled roster contains overlaps, the teacher will not be able to mark it as a "proposal". Instead, it will be marked with "conflicting proposal" so that it remains clearly visible that there are still conflicts in this schedule. The conflicts will be highlighted in order to make them more visible.

**POSTCONDITION:** A newly scheduled roster is marked as ""proposal" and this proposal is visible for the program-administrator.



## Functional Requirement FRT6

**TITLE:** Teacher manual adjustments

**DESCRIPTION:** Teachers can manually adjust their schedule.

**PRECONDITION:** Courses must be assigned to the teacher. The teacher has done a scheduling.

**SCENARIO:** The teacher can change the times when the courses are being scheduled in order to remove certain existing overlaps by clicking and dragging the courses around the schedule. He can also edit the facilities needed for a course by clicking on the course, after which he can perform another scheduling. If conflicts are resolved, the courses lose their highlighted state. The teacher can mark the schedule as "proposal".

**EXCEPTIONS:** Conflicts occur, the teacher cannot mark the schedule as "proposal". Instead, it will be marked with "conflicting proposal".

**POSTCONDITION:** Manual changes have been made and the new roster is marked as a "proposal".

## User Class 5: Program-administrator

### Functional Requirement FRP1

**TITLE:** View programs and courses

**DESCRIPTION:** Program-administrators can look at all the programs that they have to control.

**PRECONDITION:** None

**SCENARIO:** When program-administrators click on the "Edit Programs" button, they will be redirected to a page where a list of the programs that they are in control of, will be displayed. If they press on one of the programs, a list containing all the courses of that program will be displayed.

**EXCEPTIONS:** If there are no programs or if there are no courses in a program, a message will notify the program-administrator.

**POSTCONDITION:** The correct list of all the programs under control of the program-administrator will be displayed.

### Functional Requirement FRP2

**TITLE:** Add Program

**DESCRIPTION:** Program-administrators can add a program to their repertoire.

**PRECONDITION:** None

**SCENARIO:** When program-administrators want to add a program, they go to their "Edit Programs" view and press on the "Add Program" button. A pop-up window shows up, where they have to fill in a name for the program. After hitting the save button, the program is created.

**EXCEPTIONS:** When the program-administrators press cancel, no program is created and they are redirected to the "Edit Programs" view.

**POSTCONDITION:** A new program is created.

### Functional Requirement FRP3

**TITLE:** Add Course

**DESCRIPTION:** Program-administrators can add a course to one of their programs.

**PRECONDITION:** The program-administrators should have at least one program under their control.

**SCENARIO:** When program-administrators want to add a course, they go to their "Edit Program" view and press on the program where they want

to add the course. Then they press on the "Add Course" button. A form appears on the screen, where they can fill in the name of the course, assign a teacher to it, state how many hours the course has, note if its an obligatory course or an optional one... After hitting the save button, the course is added to the program.

**EXCEPTIONS:** When program-administrators press cancel, the course isn't added and they are returned to the "Edit Program" screen.

**POSTCONDITION:** The new course is added to a program.

#### **Functional Requirement FRP4**

**TITLE:** Edit program/course

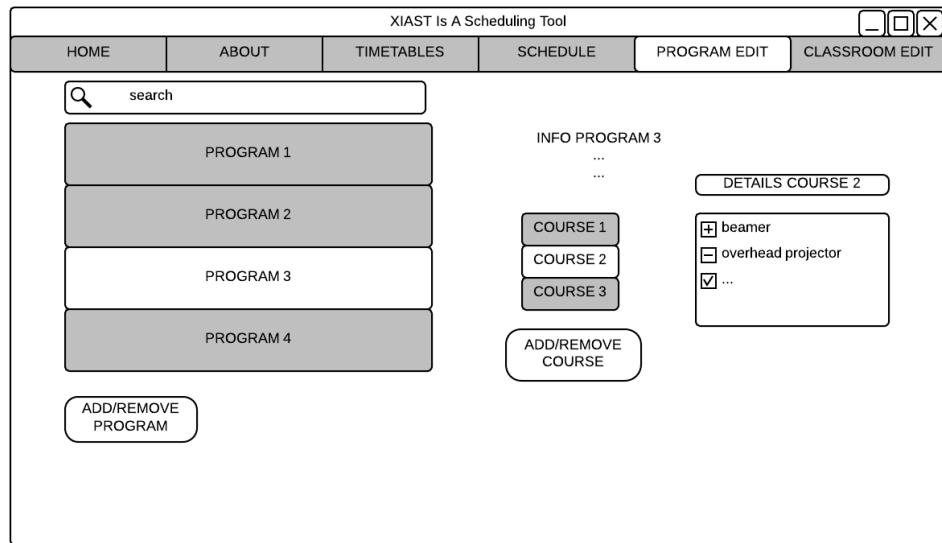
**DESCRIPTION:** Program-administrators can edit the courses inside their programs.

**PRECONDITION:** The program-administrators should have at least one program under their control, containing a minimum of one course.

**SCENARIO:** The program-administrators go to their "Edit Programs" screen and click on the program they want to edit. The list of courses of this program is shown. The program-administrators can remove the course from the program, or they can click on the course. Then they can change the details of the course. After hitting save, the form is updated.

**EXCEPTIONS:** When the program-administrators press cancel, the course isn't changed and they are returned to the "Edit Programs" screen.

**POSTCONDITION:** Course details are changed and/or courses are removed from a program.



### Functional Requirement FRP5

**TITLE:** Add/remove and edit classrooms

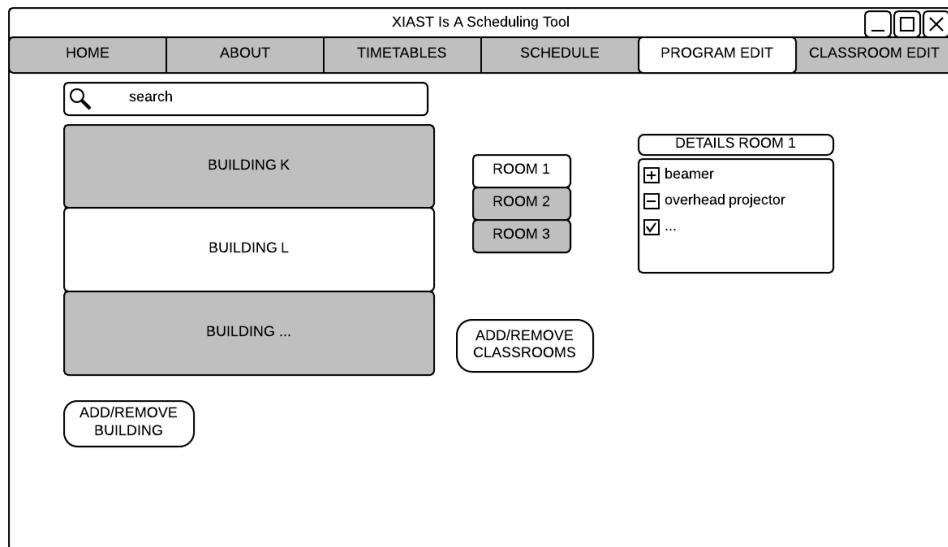
**DESCRIPTION:** Program-administrators can add/remove and edit classrooms.

**PRECONDITION:** None.

**SCENARIO:** The program-administrators go to their "Edit Classrooms" screen and click on the building where they want to add/remove or edit a classroom. The list of rooms in this building is displayed. The program-administrators can remove the building/classroom from the system, or they can click on the classroom to edit its details. After hitting save, the form is updated.

**EXCEPTIONS:** When the program-administrators press cancel, no changes are made.

**POSTCONDITION:** Classrooms are added/removed or edited.



### Functional Requirement FRP6

**TITLE:** View preferences/facilities courses

**DESCRIPTION:** Program-administrators can see the preferences and facilities required for each course in their program.

**PRECONDITION:** The program-administrators should have at least one program under their control, containing a minimum of one course.

**SCENARIO:** The program-administrators go to the "Edit Programs" screen and click on the program which contains the course of which they want to see the details of. They press on the course and then Xiast shows the preferences and facilities put forward by the teacher of the course. By hitting back, the program-administrator can return to the "Edit Program" view.

**EXCEPTIONS:** When the teacher of the course hasn't filled in the preferences and facilities of the course, the program-administrator will be alerted with a notification.

**POSTCONDITION:** Course preferences and facilities are shown on the screen.

### Functional Requirement FRP7

**TITLE:** Semi-scheduling

**DESCRIPTION:** Program-administrators can command Xiast to perform a scheduling on the courses of their programs and mark it as "final" or "frozen".

**PRECONDITION:** The program-administrators should have at least one program under their control, containing a minimum of one course.

**SCENARIO:** The program-administrators go to their "Schedule" screen and click on the program they want to schedule. He clicks on "Schedule" to perform a scheduling. Xiast shows the best schedule it could compute. If the program-administrator approves of this schedule, it can be marked as "frozen". Now this schedule will become visible for every user.

**EXCEPTIONS:** If the newly scheduled roster contains overlaps, the schedule cannot be marked as a "final". Instead, it will be marked with "conflicting proposal" so that it remains clearly visible that there are still conflicts in this schedule. The conflicts will be highlighted in order to make them more visible.

**POSTCONDITION:** The new "final" schedule is available for everyone.

XIAST Is A Scheduling Tool						
HOME	ABOUT	TIMETABLES	SCHEDULE	PROGRAM EDIT	CLASSROOM EDIT	X
<input style="width: 100px; height: 20px; border: none; padding: 2px; margin-bottom: 5px;" type="button" value="PROGRAM 1"/> <div style="border: 1px solid black; width: 15px; height: 15px; display: inline-block; vertical-align: middle;"></div>			<input style="width: 50px; height: 20px; border: none; padding: 2px; margin-right: 10px;" type="button" value="CURRENT"/> <input style="width: 50px; height: 20px; border: none; padding: 2px;" type="button" value="PROPOSITION"/>			
	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
8						
9						
10						
11			COURSE 1			
12			CLASH			
13			COURSE 2			
14						
15						
16						
17						
18						
19						
20						
21						
<input style="width: 100px; height: 20px; border: none; padding: 2px; margin-bottom: 5px;" type="button" value="PERFORM SCHEDULE"/>				<input style="width: 100px; height: 20px; border: none; padding: 2px;" type="button" value="FREEZE ROSTER"/>		

### Functional Requirement FRP8

**TITLE:** Manual scheduling

**DESCRIPTION:** Program-administrators can manually perform changes to a schedule.

**PRECONDITION:** The program-administrators should have at least one program under their control, containing a minimum of one course.

**SCENARIO:** The program-administrators can change the times when the courses are being scheduled in order to remove certain existing overlaps by clicking and dragging the courses around the schedule. If conflicts are resolved, the courses lose their highlighted state. They can mark the schedule as "final".

**EXCEPTIONS:** Conflicts occur, the schedule cannot be marked as "final". Instead, it will be marked as "conflicting proposal".

**POSTCONDITION:** Manual changes have been made and the new roster is marked as a "final". Now this schedule will become visible for every user.

### 3.3 Performance requirements

No particular performance related objectives were put forward for this project, however the aim of this project is to make Xiast as performant as possible. The number of students should not be of any concern, since students don't communicate with each other. The number of teachers or program-administrator also doesn't influence the performance of the application, however time to negotiate between different teachers and program-administrators to come up with a solution for overlaps will increase, since they have to reach a consensus about the subject. The biggest performance issue will probably be the automatic scheduling. This scheduling should take place in a reasonable amount of time, after which the roster should be displayed immediately.

### 3.4 Design constraints

The application must be able to run on personal computers and on android smartphones. For these devices we must consider the limited memory, smaller screen and slower processors. The application should still be able to run decently under these constraints. Only free, open-source software/programming languages may be used.

### 3.5 Software system attributes

**Reliability** The application should work the way it is intended to work at all times. The outcome of using the scheduling algorithm should be consistent with the data provided. Similar scheduling problems should result in similar solutions. Users should be able to have access to all functionalities described in this SRS. We will provide support for this project at least until May 21st of the year 2014.

**Availability** The application is available for the user as long as they have access to a web-browser, using a personal computer or android smartphone. Since this project works with an external server (Wilma), availability of the database cannot be regulated by any of the team-members. However, the goal is to have as less down-time as possible. Should there be a server failure, the team will contact the person responsible for the server and try to help get it back up as soon as possible.

Personal schedules will be saved offline, so users will have access to their schedule at all time.

**Security** The database will be protected from SQL-injections. Users will only have access to their own data.

**Portability** The application will be available for android devices and any web-browser.

### 3.6 Other requirements

The application should at least be available in Dutch and English.