Bachelor Technische Informatica

Kroket

Kroket

User Requirements Document

Project Manager:
Sebastiaan Candel

Quality Assurance Manager:
Ronald van Zon

Senior Management:
Mark van den Brand, MF 7.0.96
Lou Somers, MF 7.145

Advisor:
Erik Scheffers, MF 7.146

Customer:
Lex Lemmens, HG -01.27

Authors:
Peter van Heck (0649886)
Peter Koymans (0748876)
Kay Lukas (0758084)
Astrid Pieterse (0743073)
Robbert Raats (0675661)
Willem Sonke (0752229)
Roby Visser (0654428)

October 26, 2012    Eindhoven
Abstract

This document contains the user requirements for the kroket application, which aids students in choosing electives for the new Bachelor College. This application is made as part of the Software Engineering Project at Eindhoven University of Technology.

The user requirements were established by Group kroket based on discussions with our customer, Lex Lemmens, the dean of the Bachelor College. The document complies with the User Requirements Document (URD) of the Software Engineering Standards, as set by the European Space Agency (ESA) [1].
Contents

Document Status Sheet 4

Document Change Records 5

1 Introduction 6
   1.1 Purpose .................................................. 6
   1.2 Scope ................................................... 6
   1.3 List of definitions and abbreviations .......................... 6
      1.3.1 Definitions ........................................... 6
      1.3.2 Abbreviations ......................................... 7
   1.4 List of references ......................................... 7
   1.5 Overview ................................................. 7

2 General description 8
   2.1 Product perspective ....................................... 8
   2.2 General capabilities ...................................... 8
      2.2.1 Searching for subjects ................................. 8
      2.2.2 Creating schedules .................................... 8
      2.2.3 Login ................................................ 9
   2.3 General constraints ....................................... 9
   2.4 User characteristics ..................................... 9
      2.4.1 Students .............................................. 9
      2.4.2 Education and Student Service Centre .................. 9
      2.4.3 Dienst ICT .......................................... 10
      2.4.4 Bureau Computer Facilities ............................ 10
   2.5 Environment description .................................. 10
   2.6 Assumptions and dependencies .............................. 10

3 Specific requirements 12
   3.1 Capability requirements .................................... 12
   3.2 Constraint requirements ................................... 14
   3.3 Performance requirements .................................. 15
A Use cases

A.1 Abstract use cases .......................................................... 16
  A.1.1 Register ................................................................. 16
  A.1.2 Log in to the website ................................................. 16
  A.1.3 Log out of the website ............................................... 17

A.2 Use cases ................................................................. 17
  A.2.1 Search for subjects ................................................... 17
  A.2.2 Add subject to schedule ............................................ 17
  A.2.3 Remove subject from schedule ................................... 18
  A.2.4 Save schedule ....................................................... 18
  A.2.5 Load schedule ....................................................... 18
  A.2.6 Print schedule ...................................................... 19
  A.2.7 Validate schedule .................................................. 19

B Bachelor College ......................................................... 20
# Document Status Sheet

## General

<table>
<thead>
<tr>
<th>Document title</th>
<th>User Requirements Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification</td>
<td>Documentatie.URD.2.0.0</td>
</tr>
<tr>
<td>Authors</td>
<td>Peter Koymans, Robbert Raats, Roby Visser</td>
</tr>
<tr>
<td>Document status</td>
<td>Approved by Lex Lemmens</td>
</tr>
</tbody>
</table>

## Document history

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author</th>
<th>Reason of change</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.0</td>
<td>3-9-2012</td>
<td>Willem Sonke</td>
<td>Initial version</td>
</tr>
<tr>
<td>0.0.1</td>
<td>6-9-2012</td>
<td>Peter Koymans, Robbert Raats, Roby Visser</td>
<td>First version for internal review</td>
</tr>
<tr>
<td>0.1.0</td>
<td>7-9-2012</td>
<td>Peter Koymans, Robbert Raats, Roby Visser</td>
<td>First concept for review</td>
</tr>
<tr>
<td>0.2.0</td>
<td>10-9-2012</td>
<td>Robbert Raats</td>
<td>Version after first meeting</td>
</tr>
<tr>
<td>0.2.1</td>
<td>11-9-2012</td>
<td>Robbert Raats</td>
<td>Version after student meeting</td>
</tr>
<tr>
<td>0.3.0</td>
<td>11-9-2012</td>
<td>Peter Koymans</td>
<td>Second version for internal review</td>
</tr>
<tr>
<td>0.3.1</td>
<td>11-9-2012</td>
<td>Peter Koymans</td>
<td>Version after second internal review</td>
</tr>
<tr>
<td>1.0.0</td>
<td>14-9-2012</td>
<td>not applicable</td>
<td>Accepted by Lex Lemmens</td>
</tr>
<tr>
<td>1.0.1</td>
<td>1-10-2012</td>
<td>Peter Koymans</td>
<td>User requirements and use cases added</td>
</tr>
<tr>
<td>2.0.0</td>
<td>5-10-2012</td>
<td>not applicable</td>
<td>Accepted by Lex Lemmens</td>
</tr>
</tbody>
</table>
Document Change Records

General

Date 5-10-2012
Document title User Requirements Document
Identification Documentatie.URD.2.0.0

Document history since 1.0.0

<table>
<thead>
<tr>
<th>Sections</th>
<th>Reason of change</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>UCR12 added</td>
</tr>
<tr>
<td>3.1</td>
<td>UCR26 added</td>
</tr>
<tr>
<td>3.1</td>
<td>UCR29 added</td>
</tr>
<tr>
<td>3.1</td>
<td>UCR34 added</td>
</tr>
<tr>
<td>3.1</td>
<td>UCR35 added</td>
</tr>
<tr>
<td>3.2</td>
<td>UCR41 added</td>
</tr>
<tr>
<td>A.1.1</td>
<td>New section Register added</td>
</tr>
<tr>
<td>A.2.5</td>
<td>New section Load schedule added</td>
</tr>
<tr>
<td>A.2.7</td>
<td>New section Validate schedule added</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Purpose

The User Requirements Document (URD) contains the requirements for kROKET. These requirements are a negotiated agreement between the Bachelor College and kROKET. All of the listed requirements, and only these, will be implemented in kROKET, according to their priorities. Any changes to these requirements require the full consent of both parties.

1.2 Scope

kROKET is an application designed and developed by kROKET group for the Bachelor College at the Eindhoven University of Technology. The application is designed to aid students in determining their choice for electives.

1.3 List of definitions and abbreviations

1.3.1 Definitions

Bachelor College The result of a reform of bachelor education at the TU/e. See appendix B for a description.

Capability requirement Describes the functions and operations needed by users.

Constraint requirement Places restrictions on how software can be built and operated.

KROKET Software engineering team developing the application.

Microsoft BizTalk Microsoft Biztalk is a Microsoft product: it connects systems inside and across organizations for data exchange and process orchestration.

MoSCoW A system to classify the priority of user requirements. Details are explained in chapter 3.
### 1.3.2 Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCF</td>
<td>Bureau Computer Facilities</td>
</tr>
<tr>
<td>Dienst ICT</td>
<td>Information and Communication Technology Service</td>
</tr>
<tr>
<td>ECTS</td>
<td>European Credit Transfer System</td>
</tr>
<tr>
<td>ESA</td>
<td>European Space Agency</td>
</tr>
<tr>
<td>KROKET</td>
<td>Kies niet Roekeloos maar Objectief Keuzevakken Efficiënt en Tevreden</td>
</tr>
<tr>
<td>OASE</td>
<td>Online Administrative Study Environment</td>
</tr>
<tr>
<td>SEP</td>
<td>Software Engineering Project</td>
</tr>
<tr>
<td>STU</td>
<td>Education and Student Service Centre</td>
</tr>
<tr>
<td>TU/e</td>
<td>Eindhoven University of Technology</td>
</tr>
<tr>
<td>URD</td>
<td>User Requirements Document</td>
</tr>
<tr>
<td>USE</td>
<td>User, Society and Enterprise</td>
</tr>
</tbody>
</table>

### 1.4 List of references

1. ESA PSS-05-0 Issue 2 (February 1991) The User Requirement Definition Phase


### 1.5 Overview

The remainder of this document will give a general discussion in chapter 2 and specific requirements in chapter 3.

Section 2.1 and 2.2 discuss the product perspective and general capabilities respectively. Section 2.3 discusses the general constraints KROKET has to comply with. Section 2.4 on User Characteristics describes the different parties that will use KROKET in the future. In section 2.5 the environment in which KROKET will be used is discussed together with user restrictions that comply with the wishes of the customer.

Chapter 3 lists the requirements and their respective priorities. Section 3.1 deals with capability requirements, while section 3.2 deals with constraint requirements.
Chapter 2

General description

2.1 Product perspective

The reform of the bachelor at the TU/e gives greater flexibility to students. We describe the details of this reform in appendix B. Students can choose electives to accentuate their study under mild constraints.

The greater flexibility requires good planning from students. *KROKET* will allow students to search for and schedule electives. *KROKET* is a new project: it has no predecessors.

2.2 General capabilities

The capabilities of *KROKET* will be described below, divided in several categories.

2.2.1 Searching for subjects

Students can search for electives based on different filters. Possible filters are:

- Year and quartile.
- Interests, such as sports, health and energy.
- Coherent packages and USE-packages.
- Difficulty.
- Broadening versus deepening.

After a search request, a list of matching subjects will be displayed. A combination of filters can also be used. In this case, only subjects matching all filters will be displayed. Students can then ask for detailed course information.

2.2.2 Creating schedules

After searching for electives, students can allocate subjects to a timeslot in some quartile and year. It is possible to allocate several subjects in the same timeslot. *KROKET* will give a
warning in this case: the student can then decide to go ahead or abort. Students can also remove subjects from timeslots. In this way the student is able to create a schedule. It is also possible to print schedules. When a student has chosen all its electives, kroket can check if it meets the requirements.

2.2.3 Login

Students can also login on kroket, which gives them several extra options. When logged in, students will also see their major subjects. This allows them to make a complete studyplan. Furthermore, students can save and load schedules for future use.

2.3 General constraints

The system will be located at the servers of the TU/e, so bandwidth should not be a problem. Furthermore the system should be designed to be fast, reliable, and easy to be maintained. The last constraint is especially important, as kroket may be altered by another team in the future.

The end-users of the system are students of the TU/e who are following the Bachelor College. Due the nature of students the system should be easy and intuitive to use, otherwise they will have little interest in actually using the system.

2.4 User characteristics

This section will address the different User Groups and the main purpose the application will have for them.

2.4.1 Students

Students will use the application in order to create schedules, which is an important part of making studyplans. The student has to choose which USE and elective subjects he or she wants to follow. Students can search for subjects and place it in their schedule.

kroket is able to give students interactive feedback about their schedule. Firstly, it gives a warning if the same timeslot is used twice. It can also check for students if their schedule meets the requirements.

kroket also can suggest students to take a whole packet, when a certain subject of that packet is chosen. The student hereby is able to choose an entire packet. The packet then is imported in the schedule.

2.4.2 Education and Student Service Centre

Future changes in the major, USE and elective subjects are possible. The information used by the application is managed by the Education and Student Service Centre (STU). This means that STU has to provide the information needed by the application. When the development of kroket is complete, kroket will be run on the STU servers. Maintenance of kroket after development will be the respnsibility of STU.
2.4.3 Dienst ICT

While STU is responsible for maintaining student information, Dienst ICT takes care of online access to student information. Dienst ICT will provide kroket a link with access to student information. Furthermore, the nature of kroket might require changes to the underlying datastructures. This will also be the responsibility of Dienst ICT.

2.4.4 Bureau Computer Facilities

The Bureau Computer Facilities (BCF) maintains the server on which the application is installed and running. This server will only be available during the development of kroket for testing purposes. After development, kroket will run on the STU servers as mentioned earlier. BCF is responsible for maintaining the server environment such that the application operates properly and securely.

2.5 Environment description

We will now analyze the environment that kroket operates in. Here we consider kroket as a black box. We first look at the domain of the software. The domain of a system is defined as the area of business and technology in which it operates.

In the case of kroket, the domain is a web-based tool to support students. From now on we will refer to kroket as the system. The main users for the system are therefore students. Students can make schedules by sending requests to the system. The system runs on one of the BCF servers. It will be waiting for incoming requests.

When the system receives a request, the system will send an immediate response. This is possible, because the system locally stores the information needed for the application. The local information will be updated on a regular basis. The servers of Dienst ICT need to be contacted to update this local information. These servers can be queried with Microsoft BizTalk.

As mentioned earlier, STU is responsible for maintaining the information. The complete domain model is displayed in figure 2.1.

2.6 Assumptions and dependencies

This section contains a list of assumptions for kroket to operate properly.

- A year consists of four quartiles.
- A quartile consists of five timeslots.
- kroket allows students to plan up to ten years.
- Students have an username and password, provided by the TU/e. kroket must be able to check student usernames and passwords, this information is provided by STU.
- STU provides kroket with correct subject information.
• STU provides kroket with complete subject information.
• BCF provides kroket a working server.
Chapter 3

Specific requirements

In this chapter all requirements and constraints of the product to be developed are specifically stated. The product will adhere to these requirements. Furthermore any requirements resulting from additional requests are included here.

The requirements are based on the use cases, that are analyzed in appendix A. For prioritizing the specific requirements for the program, the MoSCoW model will be used [2]. The capital letters in MoSCoW stand for:

M Must have; these requirements are essential for the product.
S Should have; these requirements are not critical for the product to work, but are nearly as important as the must haves, meaning they must be implemented if at all possible.
C Could have; requirements which are not critical to the product’s success. If they can be implemented with little development costs, they can increase the Client’s satisfaction.
W Won’t have; these requirements will not be implemented in this Project. However, they would be nice to have in future versions of the product.

3.1 Capability requirements

<table>
<thead>
<tr>
<th>UCR₁</th>
<th>must have</th>
</tr>
</thead>
<tbody>
<tr>
<td>There is an English version of the application.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UCR₂</th>
<th>could have</th>
</tr>
</thead>
<tbody>
<tr>
<td>There is a Dutch version of the application.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UCR₃</th>
<th>should have</th>
</tr>
</thead>
<tbody>
<tr>
<td>A user can use its TU/e account to log on.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UCR₄</th>
<th>must have</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users can see a list with all possible electives.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UCR₅</th>
<th>must have</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electives can be filtered on the year it is given in.</td>
<td></td>
</tr>
<tr>
<td>UCR</td>
<td>Requirement Details</td>
</tr>
<tr>
<td>-------</td>
<td>-------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>UCR6</td>
<td>Electives can be filtered on the quartile it is given in.</td>
</tr>
<tr>
<td>UCR7</td>
<td>Electives can be filtered on the coherent packages it is in.</td>
</tr>
<tr>
<td>UCR8</td>
<td>Electives can be filtered on the USE-packages it is in.</td>
</tr>
<tr>
<td>UCR9</td>
<td>Electives can be filtered on the interest of the student.</td>
</tr>
<tr>
<td>UCR10</td>
<td>Electives can be filtered on the difficulty of the subjects.</td>
</tr>
<tr>
<td>UCR11</td>
<td>Electives can be filtered on broadening subjects or deepening subjects.</td>
</tr>
<tr>
<td>UCR12</td>
<td>Electives can be searched on name, subject code and subject description with keywords.</td>
</tr>
<tr>
<td>UCR13</td>
<td>Users can add subjects to the schedule.</td>
</tr>
<tr>
<td>UCR14</td>
<td>Users can remove subjects from the schedule.</td>
</tr>
<tr>
<td>UCR15</td>
<td>A warning is shown when the user tries to schedule two subjects in the same timeslot.</td>
</tr>
<tr>
<td>UCR16</td>
<td>There is shown how much ECTS is still needed.</td>
</tr>
<tr>
<td>UCR17</td>
<td>There is shown how many subjects of a certain difficulty are still needed.</td>
</tr>
<tr>
<td>UCR18</td>
<td>Users can schedule more than three subjects per quartile.</td>
</tr>
<tr>
<td>UCR19</td>
<td>Additional subject information is shown when the user clicks on the subject.</td>
</tr>
<tr>
<td>UCR20</td>
<td>Users can see the difficulty of the subject in UCR19.</td>
</tr>
<tr>
<td>UCR21</td>
<td>Users can see the prerequisites of the subject in UCR19.</td>
</tr>
<tr>
<td>UCR22</td>
<td>Users can see the coherent packages, of which the subject is a part, in UCR19.</td>
</tr>
<tr>
<td>UCR23</td>
<td>Users can check whether the chosen subjects form a valid packet for the Bachelor College.</td>
</tr>
<tr>
<td>UCR24</td>
<td>Users can use drag-and-drop to schedule subjects, if it is supported by their browser.</td>
</tr>
<tr>
<td>UCR25</td>
<td>Users can save their schedules.</td>
</tr>
<tr>
<td>UCR26</td>
<td>Users can load a saved schedule.</td>
</tr>
</tbody>
</table>
### 3.2 Constraint requirements

We will not state explicit constraints on the computer and tablet itself, but only on the browser. Since our application should not be performance intensive, we believe there is no gain in stating these constraints explicitly. Any modern computer or tablet should be able to use the application.

<table>
<thead>
<tr>
<th>UCR36</th>
<th>should have</th>
<th>The application works in Internet Explorer version 9 and above.</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCR37</td>
<td>must have</td>
<td>The application works in Chrome version 20 and above.</td>
</tr>
<tr>
<td>UCR38</td>
<td>must have</td>
<td>The application works in Firefox version 13 and above.</td>
</tr>
<tr>
<td>UCR39</td>
<td>should have</td>
<td>The application works in Safari version 5 and above.</td>
</tr>
<tr>
<td>UCR40</td>
<td>should have</td>
<td>The application works in Opera version 12 and above.</td>
</tr>
<tr>
<td>UCR41</td>
<td>could have</td>
<td>Users with color blindness can use KROKET.</td>
</tr>
</tbody>
</table>
3.3 Performance requirements

<table>
<thead>
<tr>
<th>UCR42</th>
<th>Each page loads in 10 seconds.</th>
<th>must have</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCR43</td>
<td>Each page loads in 5 seconds.</td>
<td>should have</td>
</tr>
<tr>
<td>UCR44</td>
<td>Each page loads in 2 seconds.</td>
<td>could have</td>
</tr>
<tr>
<td>UCR45</td>
<td>Each page loads in 1 second.</td>
<td>won’t have</td>
</tr>
<tr>
<td>UCR46</td>
<td></td>
<td>must have</td>
</tr>
</tbody>
</table>

The website should be able to handle up to twenty concurrent users.
Appendix A

Use cases

We will first describe the so called abstract use cases. Abstract use cases are included in, or generalize other use cases. After discussing the abstract use cases, we will discuss ‘ordinary’ use cases.

A.1 Abstract use cases

A.1.1 Register

Goals: To register an account.
Preconditions: None.
Summary: User is registered.
Priority: Should have.
Steps:

*Actor actions:*

1. Click on the register option.
2. Display register page.
3. Enter username.
4. Enter password.
5. Enter confirmation for password.
6. Click on the register button.

*KROKET response:*

7. Create a new account.

Alternatives: There are two alternatives for step 7. Firstly, the username may already be in use. Secondly, the password may not match the confirmation. In both cases, step 7 will not be executed. An appropriate warning is given instead.

A.1.2 Log in to the website

Goals: To log in to the website.
Preconditions: User has an account.
Summary: User is logged in.
Priority: Should have.
Steps:
Kroket – USER REQUIREMENTS DOCUMENT

Actor actions:  KROKET response:
1 Click on the login option.  2 Display login page.
3 Enter credentials.
4 Click on the login button.  5 Log the user in to the website.

Alternatives: If the credentials are invalid, the user will not log in to the website in step 5: an error will be displayed instead.

A.1.3 Log out of the website

Goals: To log out of the website.
Preconditions: User is logged in.
Summary: User is logged out.
Priority: Should have.
Steps:

Actor actions:  KROKET response:
1 Click on the logout button.  2 Log the user out of the website.

3 Display log on page.

A.2 Use cases

A.2.1 Search for subjects

Goals: To search for subjects.
Preconditions: None.
Summary: A list of subjects is displayed matching the search filters.
Priority: Must have.
Steps:

Actor actions:  KROKET response:
1 Set filters for search.  2 Display a list of matching subjects.
2 Click search.

A.2.2 Add subject to schedule

Goals: To add a subject to the schedule.
Preconditions: A list of subjects is displayed after some search request.
Summary: The subject is added to the schedule.
Priority: Must have.
Steps:

Actor actions:  KROKET response:
1 Click on subject in list of subjects.  2 Highlight timeslots in the schedule.
3 Click on one of the timeslots.  4 The subject is added in selected timeslot and deleted from list of subjects.
Alternatives: There are two possible alternatives for step 4. Firstly, it is possible that an user has selected a valid timeslot in step 3, but the timeslot is already filled. In this case, KROKET will warn the user that the two subjects might have overlapping exam times. The user can then decide to proceed or abort.

Secondly, it is possible that an user has selected an invalid timeslot for the subject, i.e. a non-highlighted timeslot. KROKET will cancel the operation.

A.2.3 Remove subject from schedule

Goals: To remove a subject from the schedule.
Preconditions: At least one subject must be in some timeslot.
Summary: The subject is removed from the schedule.
Priority: Must have.
Steps:
   Actor actions:  
   1 Move mouse to a non-empty timeslot.  
   2 Click on remove.
   KROKET response:  
   2 Show remove option.  
   4 The subject is removed from the schedule and added to list of subjects.

A.2.4 Save schedule

Goals: To save the current schedule.
Preconditions: User is logged in.
Summary: The schedule is saved on the server.
Priority: Should have.
Steps:
   Actor actions:  
   1 Click on account, save schedule.
   KROKET response:  
   2 Save the schedule on the server.  
   3 Display success message.

Alternatives: If for some reason step 2 fails, a failure message is displayed in step 3 instead.

A.2.5 Load schedule

Goals: To load the stored schedule.
Preconditions: The user is logged in and saved a schedule on that account.
Summary: The schedule is loaded from the server.
Priority: Should have.
Steps:
   Actor actions:  
   1 Click on account, load schedule.
   KROKET response:  
   2 Load the schedule from the server.
**A.2.6 Print schedule**

**Goals:** To print the current schedule.

**Preconditions:** None.

**Summary:** The schedule is printed.

**Priority:** Should have.

**Steps:**

<table>
<thead>
<tr>
<th>Actor actions</th>
<th>KROKET response</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Click on the print button.</td>
<td>2 Open print display.</td>
</tr>
<tr>
<td>3 Choose printer and print options.</td>
<td>4 Print schedule page.</td>
</tr>
</tbody>
</table>

**A.2.7 Validate schedule**

**Goals:** To validate the current schedule.

**Preconditions:** None.

**Summary:** The schedule is validated based on several checks.

**Priority:** Must have.

**Steps:**

<table>
<thead>
<tr>
<th>Actor actions</th>
<th>KROKET response</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Click on the validate button.</td>
<td>2 Check for major courses.</td>
</tr>
<tr>
<td></td>
<td>3 Check for 180 ECTS.</td>
</tr>
<tr>
<td></td>
<td>4 Check for USE package.</td>
</tr>
<tr>
<td></td>
<td>5 Check for elective packages.</td>
</tr>
<tr>
<td></td>
<td>6 Display result page.</td>
</tr>
</tbody>
</table>
Appendix B

Bachelor College

Bachelor students at the TU/e study within the Bachelor College. Every student chooses a major and besides that it follows a common part and several electives as well. In short, the student has 90 ECTS in the major, 30 in the common basis and 60 credits in the electives. Our applications will focus on those 60 credits in the electives.

Choosing the electives is quite a task, since there are several constraints. At least 15 credits should be in a USE-package, three connected courses that teach a student about User, Society and Entrepreneurship. The remaining 45 credits should be sufficiently challenging, something that is ensured by assigning difficulties to the subjects.

To help the students there are coherent packages, 3 courses (15 ECTS) that together ensure the student reaches a sufficient level to gain a academic bachelor’s degree. If the electives contain two coherent packages, it will automatically satisfy the earlier mentioned constraints. The chosen electives will also satisfy the constraints if the student takes enough subjects of high difficulty. Precise criteria are however not known yet.

Furthermore, the Bachelor College uses timeslots. Every subject will be in one of the timeslots. There are five timeslots in total: A, B, C, D and E. The teaching hours for these timeslots are fixed. This is done in such a way that subjects in different timeslots have different teaching hours and exam times.