

Internship Report

**Enhancement of a web based
student self service application**

Marius Politze

Internship Supervisors:

Dr. Frank Thuijsman

Dr. Alexander Voß

Maastricht University
Faculty of Humanities and Sciences
Department of Knowledge Engineering
Master Artificial Intelligence

March 2011 - January 2012

Abstract

The Center for Computing and Communication (CCC) of RWTH Aachen University wants to further automate its administration processes regarding the training of Mathematical and Technical Software Developers (MATSE). Therefore two parts of the existing web application *MATSE-Dienste* were enhanced: A part of the current web based application that assigns courses to the students in a fair way was extended and a new functionality that allows the instructors to look at application papers of candidates that apply for the programme was implemented. Due to the great success of the online application service for MATSEs it was unexpectedly introduced for all the applications on the about 750 apprenticeship vocations of RWTH Aachen University. This report describes the course of the internship at the CCC, the implementation and the introduction of the products in RWTH Aachen university.

This internship was performed at
RWTH Aachen University
Center for Computing and Communication
Department for MATSE-Training
Seffenter Weg 32, 52074 Aachen, Germany

This internship was supervised by:

Dr. Frank Thuijsman
(Maastricht University)

Dr. Alexander Voß
(RWTH Aachen University)

Contents

1	Introduction	3
1.1	Internship Setting	3
1.2	Assignment	4
1.2.1	Assignment Part 1: Course Election	4
1.2.2	Assignment Part 2: Online Applications	5
1.3	Planning	6
1.4	Current State	6
2	Technical Background	7
2.1	Technical Setup	7
2.2	Structure of the <i>MATSE-Dienste</i>	8
2.3	Project Management	10
3	Course Election	12
3.1	Election Process	12
3.2	Distribution	13
3.3	Data Structures	17
4	Online Applications	17
4.1	Placement Process	19
4.2	Implementation	21
4.2.1	Data Access	21
4.2.2	Web Interface	23
4.3	Introduction of the System	23
5	Conclusion	24

1 Introduction

1.1 Internship Setting

The internship was carried out at the department for MATSE-Training at the Center for Computing and Communication (CCC) of RWTH Aachen University. The department for MATSE-Training co-organizes the program of study Scientific Programming at FH Aachen University of Applied Sciences. This task includes, but is not limited to, booking of rooms, organization of lectures and electives, evaluation of lectures and student application.

Scientific Programming is a dual program of study. This means that every student has a mandatory apprenticeship contract with a company that obliges the students to work for the company if they have no educational activity during working hours or semester breaks. To apply for such an apprenticeship the student sends his application form to the CCC where it is processed and distributed to the companies. The companies then decide with whom they contract.

The companies also define a student advisor who tutors the student. Furthermore this apprenticeship gives the the students unique opportunity to gain knowledge about practical applications during their studies.

In the last three years the department for MATSE-Training at the CCC developed a web based application, the *MATSE-Dienste*, to support these processes. The *MATSE-Dienste* offer access for students, lecturers, administrative personnel and the companies student advisors. Serveral modules within the *MATSE-Dienste* offer different functionalities to organize different facets of the studies such as:

Target group	Activities
Administrative Staff	<ul style="list-style-type: none">• Set up electives and other courses• View grades and evaluate exams• Document exchange• Plan exams
Students	<ul style="list-style-type: none">• Choose electives• View Grades• Document exchange

Lecturers	<ul style="list-style-type: none"> • View courses and participants • Evaluate exams • View exam participants • View student details (ID number, email address, ...)
Company Advisors	<ul style="list-style-type: none"> • View electives of apprentices • Document exchange

1.2 Assignment

The assignment of this internship was to enhance the existing web application *MATSE-Dienste* in two parts. The first was to extend a part of the current web based application that assigns courses to the students in a fair way and the second to implement a completely new functionality that allows the companies to look at application papers of candidates that apply for the program.

The two extensions to the *MATSE-Dienste* offer different possibilities: The implementation of the online application service is a greenfield development and as such can be completely influenced and has to be newly designed. The extension of the course election is a brownfield development, thus is based on an already running system that has to continue working during the development process. Both projects together give complete insight in the software development processes of the *MATSE-Dienste*.

Since both modules are based the same application framework, but their development itself does not have much in common, they will be discussed separately in Sections 3 and 4.

1.2.1 Assignment Part 1: Course Election

Before the enhancement, the places of the electives are mainly given based on the registration date and some minor rules. This leads to an unfair distribution of the available places since students that register early for a lot of electives get mostly all of them and students that register late get few or even none. Nevertheless late registrations are usually not caused by the

students but occur due to organization issues in the external companies and departments.

This behaviour of the application makes many students choose more electives than they need in order to ensure they get at least one of them, accepting the risk of having to take all of them. On the other hand students that only register for few electives were not always able to get all the electives they need during their studies. These cases had to be handled manually to meet the students' needs and cannot be done any more for more than 300 participants. Thus the registration process has to be enhanced.

This should be done in the following way: Instead of assigning places by registration time find a better distribution of students into the courses, taking into account:

- Number of electives already taken by this student
- Number of electives during this semester
- Number of places available in the courses
- Semesters left within standard period of study
- Priority assigned to electives by the students

In a nutshell a web-interface has to be developed that offers the possibility to the students to register for and rank the courses. For the staff members this interface offers the possibility to see the final algorithmic outcome.

1.2.2 Assignment Part 2: Online Applications

The second task was to develop a web based application that allows companies, that take part as an employer in the dual program of study, to access the application papers of the candidates that apply for the program. Each year there are about 200 applications for the 110 places. To be able to distribute the applications over all of the 40 companies the application papers are collected centrally at the department for MATSE-Training at the CCC.

Before the implementation of this web based application the companies had to send their human resources managers to view the applications at the CCC. Since the application phase lasts from September to June usually multiple visits are necessary for the companies to find the right applicant. These visits generate a vast overhead.

This overhead should be reduced in the following way: instead of coming to the CCC the companies get a secure web access to view the application papers online. Since the application is dealing with personal data a special focus is put on the security of the data.

1.3 Planning

Before the start of the internship an initial planning phase was carried out with the head of the department for MATSE-Training. This was condensed in a Gantt chart as displayed in Figure 1.

Due to several reasons, mostly the early adoption of the online application system for the whole university, the initial plan could not be followed exactly but had to be altered. The current status of the two parts of the assignment is shown in Section 1.4.

The actual project phases as they were carried out are displayed as a Gantt chart in Figure 2. As one can see here the development of the course election module was suspended until the online application module could be released university wide. This lead to a delay of about one month in the finishing of the course election module. Nevertheless since the course election module was planned to be used for the first time in week 4 of 2012 this delay had only a neglectable influence on the operations of the department.

1.4 Current State

At the current point in time both developed modules are fully operational and are used by the students and administrative personnel of the department for MATSE-Training and the whole University.

The course election module was released into the live system in week 4 of 2012. A course election phase from January 30th to February 10th will

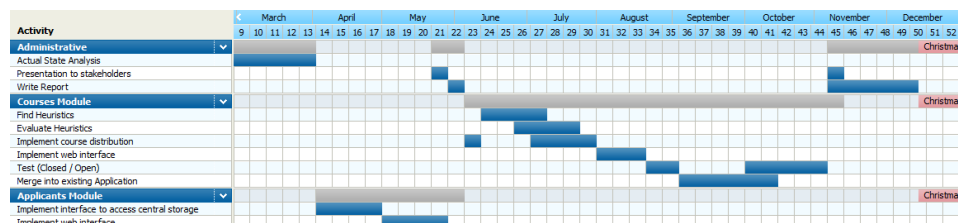


Figure 1: Initial planning prior to the internship.

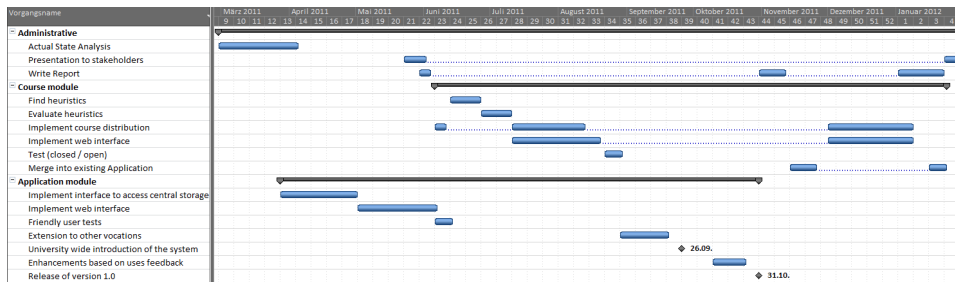


Figure 2: Actual distribution of the different tasks during the internship.

be completely using the new user interface and the new course distribution algorithm for the first time.

For the MATSE applicants, the online application module was released during the internship, in July 2011. The release then was also demonstrated to the university administration and Division 8.4 (Vocational Training) who decided to use the system starting in September 2011 for all of the 25 apprenticeship vocations with more than 750 jobs in total. With currently about 550 page impressions per week the online application module became one of the central components of the *MATSE-Dienste* as well as the university wide application process.

2 Technical Background

Before the development of the two modules is discussed the technical background that is used by the *MATSE-Dienste* is introduced. This technical background consists of two parts: the technical setup describing the server architecture and data sources of the *MATSE-Dienste* and the program structure of the software itself.

2.1 Technical Setup

The *MATSE-Dienste* web application is set up on a virtual machine (VM) hosted at the CCC. This VM can be accessed directly via the internet. The usage of a VM to host a single web service is a quite common approach since it has several benefits:

- The application is independent of a physical machine and can be easily transferred without risking long downtimes in which the service is not available.

- The technical administration can be outsourced and the software development team does not have to configure the physical set-up of the host.
- If the amount of services offered or the amount of users increases a VM can be either enhanced in terms of memory and processing power or can be mirrored and then distribute the load on multiple VMs.
- By duplication of the VM it is easy to produce a test system that has exactly the same properties as the live machine but does not interfere with the live system.

The application itself is programmed in PHP and therefore mainly run by an Apache web server that is installed on the VM. The *MATSE-Dienste* application has two data sources: an internal MySQL Database and an XML interface to Genesis World Azubi (GW) the university's central system for management of the about 750 apprentices and their advisors.

The MySQL Database can be read-write accessed by the *MATSE-Dienste* and, for example, stores information about the uploaded documents and the course registration. The XML interface can only be read and delivers personal information like name and email address, the associations between the companies advisors and the students but also the grades of the students.

During this internship the information that can be obtained from GW was extended by the application papers and personal information about the applicants.

2.2 Structure of the *MATSE-Dienste*

The *MATSE-Dienste* consist of different modules accessing the core library. This core library offers functionality to access different data sources: the MySQL database and the XML interfaces from the GenesisWorld (GW) system. Figure 3 shows a diagram of the structure described in this section.

The *MATSE-Dienste* core clearly separates the modules from the data storage. This is necessary because the modules usually have to combine data from both, the MySQL database and the XML interface. One module offers the functionality for a single administrative process such as the management of courses and participants or the exam scheduling and grades. A module then may offer different functionalities for different groups of users already mentioned in Section 1.1. An administrative staff member, for example, has a different set of possibilities than a student. Figure 4 shows two views of the *MATSE-Dienste* for Administrative staff and Students.

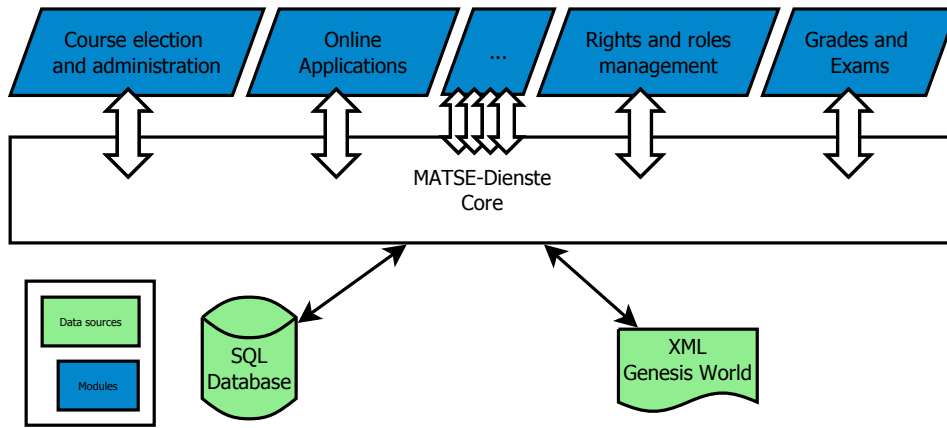



Figure 3: Structure of the MATSE-Dienste.

Auszubildende



- [Kurse](#)
 - [Hausaufgabengruppen](#)
 - [Anmeldung an Kursen](#)
 - [Hinweise zum Anmeldeverfahren](#)
 - [Seminare](#)
- [Notenansicht](#)
- [Dokumentenupload](#)
 - [Ausbildungsnachweise](#)
 - [Praxisberichte](#)
- [Dokumente](#)
 - [Prüfungsvorbereitung](#)
- [Häufig gestellte Fragen \(FAQ\)](#)

Abteilungsverwaltung

MATSE Ausbildung

- [Notenbescheinigungen](#)
- [Notenübersicht: Pflichtfächer / Wahlfächer](#)
- [Kursadministration](#)
 - [Pflichtkurse anlegen](#)
 - [Hausaufgabengruppen](#)
 - [Seminare](#)
- [Dokumentenupload](#)
 - [Ausbildungsnachweise](#)
 - [Praxisberichte](#)
 - [Protokolle der Betreuerversammlung](#)
- [Benutzerverwaltung](#)
 - [Alle Benutzer](#)
 - [Studenten](#)
 - [Dozenten](#)
- [Klausurplanungstabelle](#)
- [Abschlussprüfung - Aufsichtsplan](#)
- [Dokumente](#)
- [Häufig gestellte Fragen \(FAQ\)](#)
- [Gäste eintragen](#)
- [Fehler und Verbesserungsvorschläge melden](#)

Figure 4: Different views of the MATSE-Dienste for different users: Students (left), administrative staff (right).

2.3 Project Management

With approximately 8000 site impressions per month the *MATSE-Dienste* are a very frequently used application with approximately 450 distinct users. Regarding these numbers it is crucial that the *MATSE-Dienste* offer a very high availability since even short downtimes produce a considerable effort for the staff members of the department for MATSE-Training.

This is why the development of the *MATSE-Dienste* is also supported by an elaborated software engineering process. At the time of the internship the *MATSE-Dienste* are developed by a team of five developers. To coordinate the development two tools are used:

Version Control: Git To ensure that multiple persons can work on the source code of the application the version control system Git is used. Like every other version control system Git allows to track changes made by the developers and makes it possible to switch between recent and older versions. Furthermore Git allows every developer to change parts of the source code independently from the other developers. In a fixed schedule a release version containing the changes of all developers is then merged and tested. Figure 5 shows a screenshot of the log of the *MATSE-Dienste* repository, displaying the branches.

Issue Tracking: Trac The coordination of the different steps of development are important to be able to divide tasks among several developers. This coordination of the different tasks is provided by the software Trac. Trac therefore offers two main features: Release management and issue tracking. Using the release management part of Trac multiple issues can be bundled together to plan them for an upcoming milestone / release. This includes deadlines and priorities that give a clue about the impact of the issues. Apart from the releases Trac also allows to check the status of single issues, assigning them to a developer and see recent changes made to the source code concerning this issue. Figure 6 shows a screenshot of the ticket view showing the status of the different tickets.

Software Testing: VMWare Every developer uses a local copy of the Virtual Machine that hosts the *MATSE-Dieste*. During the development of a module every developer can test changes to the code without touching the productive system. Nevertheless since all developers are running a copy of the real host machine the changes are interchangeable and will produce the same results on the local copy of each developer and finally on the live system.

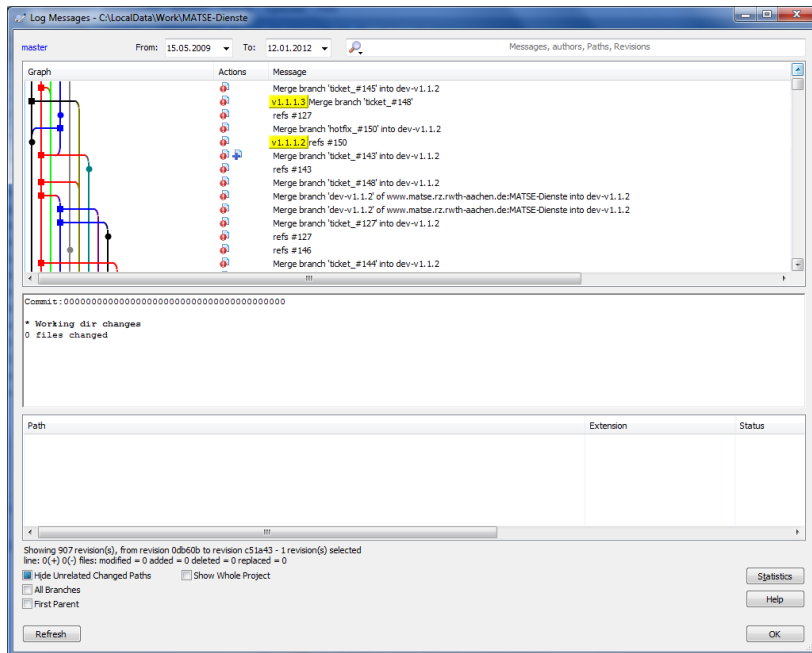


Figure 5: Screenshot of the Git versioning system.

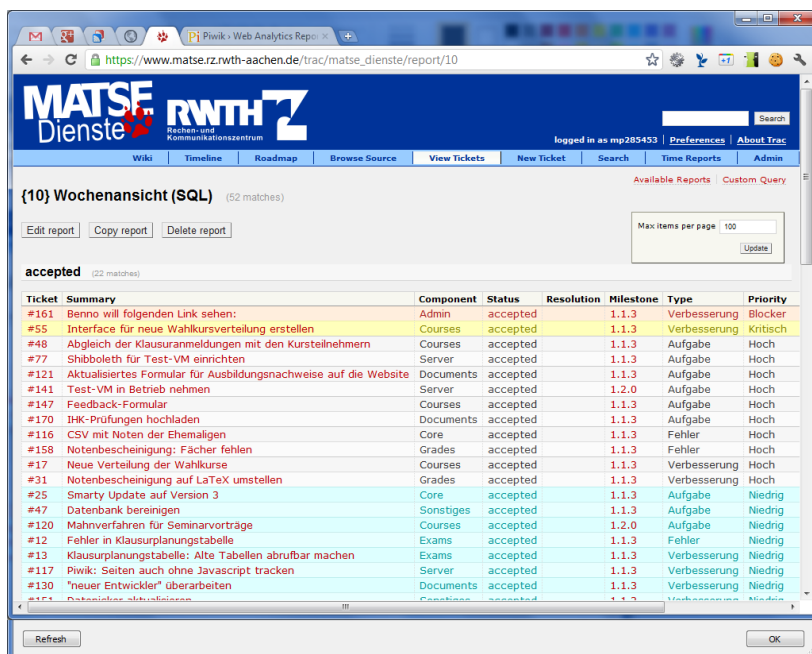


Figure 6: Screenshot of the Trac system.

3 Course Election

The development of the new Course election system is split into two parts: First a new way of the distribution of the seats had to be found. Then the existing application had to be changed in a way that the new distribution can be used. This process is described in Section 3.2.

To be able to implement the distribution process the election process performed by the administrative staff and the students was redesigned. This was necessary since the new distribution process needs more information than the old process had provided. Section 3.1 compares the old and the new process from the users' view.

3.1 Election Process

This part of the internship assignment deals with the enhancement of the existing application that takes care of the process of course election which consists of the following steps:

1. A member of the administrative staff creates several single electives for the upcoming semester
2. The students may then choose one or more of the electives
3. The students advisor now has to approve the choice of the student
4. A member of the administrative staff then distributes the places of the course among the registered and approved students

For the enhancement it was not admissible that the general process would be changed but rather minor steps of the election process were altered. Furthermore it was mandatory that the changes would guarantee that the results are better than the ones of the existing system. Generally speaking the following changes were made:

- Instead of single courses that are elected by themselves, the courses are now combined in an election set.
- From each election set the student may only choose courses once.
- The electives from one election set are ordered by the student.
- To control the overall study load, the student may define a maximum number of electives that shall not be exceeded.

- The advisor may approve single courses from one election set and may also reset the complete election.
- The places of the courses in one election set are automatically distributed to the approved students.

Figure 7 shows the user interface for the students, that was developed during the internship. Figure 8 shows user interfaces for the advisors respectively.

The course election web interface now offers a system that works similar to most online shopping systems. The student selects a subset of courses, these selected courses are then stored in a «basket». These courses may then be ordered according to their priority by the student. Once the student is finished with the election, it can be submitted to the system. The Advisor is now informed that he has to approve the course election. The following actions can be performed by the advisor: Accept or reject single courses or reject the complete election. In the latter case the student may choose a new set of courses as displayed in the flowchart in Figure 9.

3.2 Distribution

The part that was changed the most during the internship was the actual distribution of the available seats of the courses. This part of the course module distributes the courses by taking the following information into account:

1. Number of places available in the courses (places are available or not):
 $\{no, yes\}$
2. Number of electives already taken by this student (more than maximum):
 $\{yes, no\}$
3. Number of electives during this semester (more than maximum):
 $\{yes, no\}$
4. Semesters left within standard period of study:
 $\{1, 2, 3, 4, 5\}$
5. Priority assigned to electives by the students:
 $\{1, 2, 3, 4, 5\}$

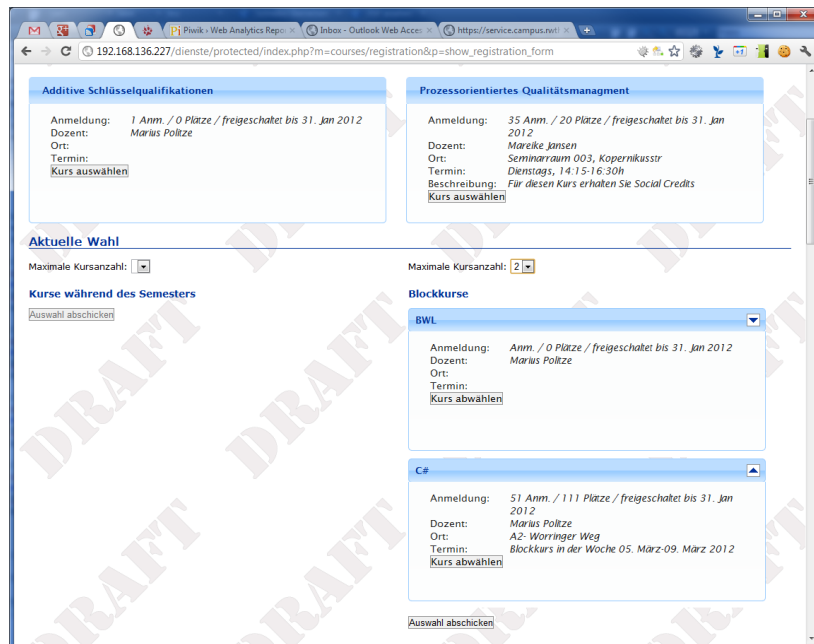


Figure 7: Screenshot of the course election web interface for the students after the enhancement.

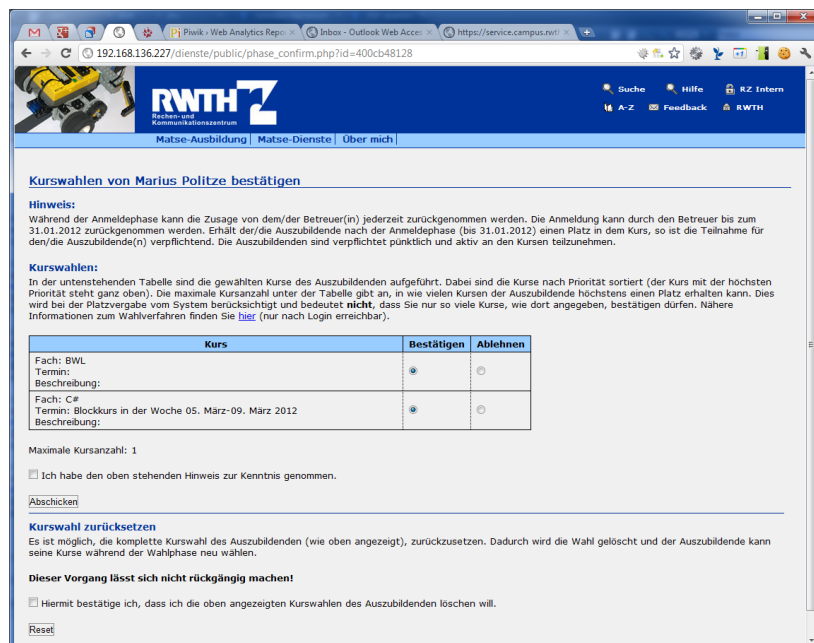


Figure 8: Screenshot of the course election web interface for the advisors after the enhancement.

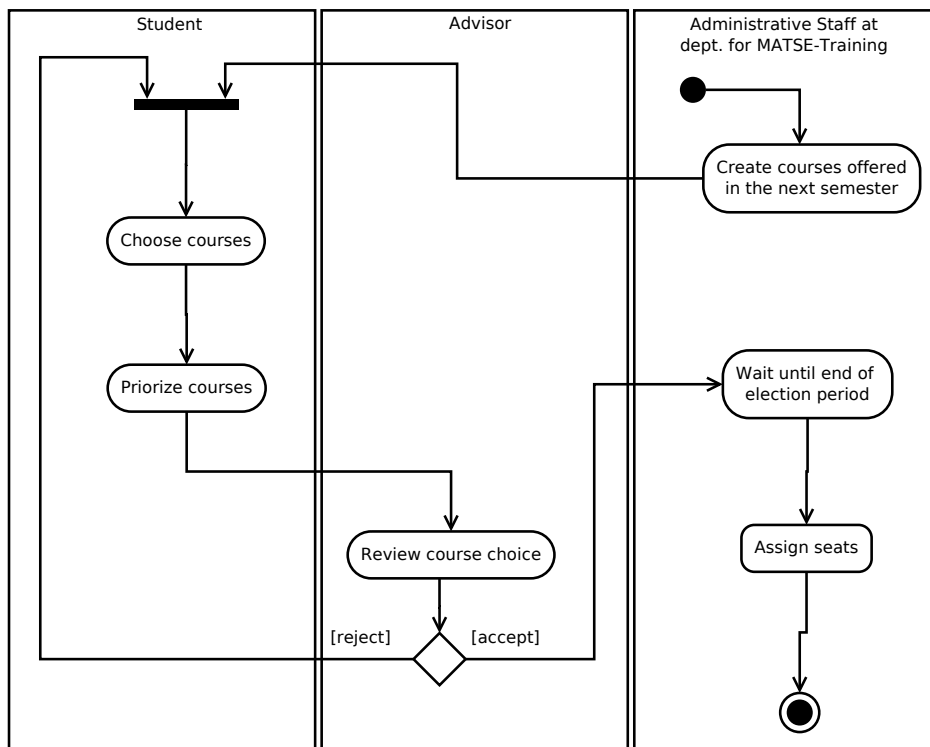


Figure 9: Activities taken to chooodiaporse and distribute seats for electives.

The values in braces show the range of the Attributes. The values of the attributes can be ordered and are given in ascending order.

To distribute the free seats in a course an iterative approach was chosen. The registered students are sorted by the criteria above starting with the first. If the criterion is tied the next one is used. If all criteria tie, a random number is used to order the students. Given this ordered list the first student gets a place in the course with the highest priority. If the course with the highest priority is already full, the second highest prioritized course will be taken. A pseudo code showing the distribution algorithm is displayed in Listing 1.

```
1 function performAssginment (l : registeredStudents)
2   while sizeof(l) > 0 do
3     first := pop(l);
4     if placesLeft (courseOf(first)) > 0 then
5       assignSeat (first);
6     else
7       needUpdate := list ();
8       for each d in l do
9         if needUpdate contains studentOf(d) then
10          incrementPriority (d)
11        else if courseOf(d) = courseOf(first) then
12          remove(d, list);
13          add (studentOf(d), needUpdate);
14        fi
15      od
16    fi
17    sort(l);
18  od
19 end
```

Listing 1: Pseudocode of the course distribution algorithm.

While some alternatives to the displayed algorithm existed it was chosen due to two reasons:

- It does not involve randomness: Even though other algorithms e.g. using monte carlo methods were investigated this iterative solution was chosen even though it might deliver a worse result. This was due to the reason that it produces replicable results that can be explained to students and companies advisors.

- It runs reasonably fast: Other algorithms, not involving randomness, would have to perform a search on the course registrations to find a good distribution. Nevertheless it turned out that these algorithms, given the limitations of PHP, are too slow to be used in a web application.

3.3 Data Structures

The data that the course module deals with is mainly stored in the MySQL database of the *MATSE-Dienste*. Figure 10 shows an entity relationship diagram of the schemes used by the module. The most central scheme for the course election is `md_course_registration` as it stores the election, its priority and its status. The other schemes store information about the courses like the lecturer and the semester as well as the election period during which the course may be elected. The scheme `md_user` stores the user data such as the email address and the users ID.

The list l of registered students used in Listing 1 is created from the data in `md_course_registration` as it contains all current and all former course registrations.

4 Online Applications

The second part of the *MATSE-Dienste* that was tackled during the internship was the development of a tool that allows supervisors of companies and institutes to view application documents via the internet. For this project not only the CCC and the department for MATSE-Training were involved but also Division 8.4 (Vocational Training) of the Central University Administration of RWTH Aachen.

Apart from the implementation the software had to be developed from scratch. Therefore the placement process had to be modelled and optimized in a way that allows automatic processing. This will be described in Section 4.1.

The next step was to implement the software and the web interface that allow users to access the application data. Since the applications contain personal data this step put a vast priority on security. In short this means that only authorized persons may access the data. The implementation, the security requirements and how they were dealt with is shown in Section 4.2.

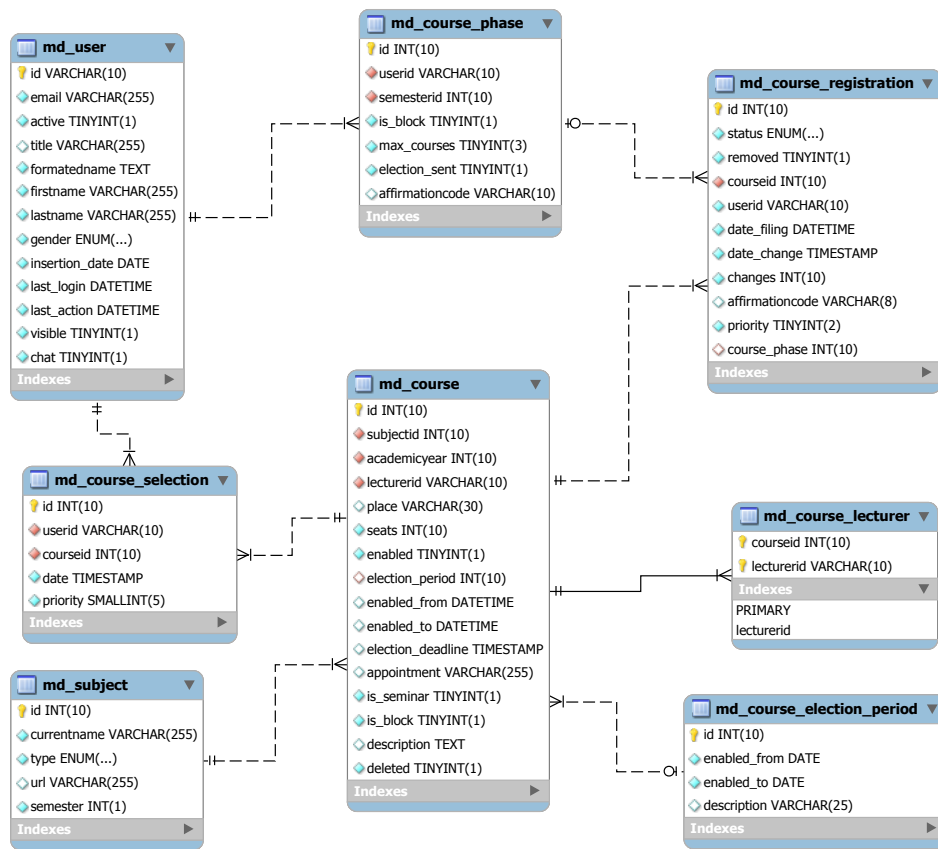


Figure 10: Entity relationship diagram displaying the database structures used in the course election module.

The last step was the introduction of the system within the University. This was done in two steps the system was made available early during development to a group of selected friendly users. These users tested the application and gave immediate feedback. Then the application was made available to all companies and institutes that participate in the MATSE-Training. A brief overview of the course of the introduction of the system at the University is given in Section 4.3.

4.1 Placement Process

The placement process was first formalized. This formal process is displayed in Figure 11. The actions taken in the shown steps are explained below:

View Applicants at CCC One of the companies' representatives, usually the future advisor of the applicant, is coming to the offices of the department of MATSE-Training. They select one or more application papers which are then copied for them.

Select Applicant At the companies' office one or more applicants are selected from the copies. These applicants are then invited for an interview.

Invite Applicant The applicant is invited for a job interview.

Hire Applicant If a company wants to hire the applicant they make an apprenticeship contract.

Notify department for MATSE-Training Throughout the process the company has to inform the department for MATSE-Training. This information is then used by the department to maintain the pool of applicants. For every applicant the department maintained a list of interested companies.

Notify other companies that selected the applicant In this step the department for MATSE-Training notifies companies that had also selected the applicant of the fact that he / she was hired.

Using the online application system this process was optimized and automated as far as possible. The biggest flaw in the initial process was the need for the companies representative to repetitively come to the department for MATSE-Training in order to find an appropriate applicant. Furthermore the process involved multiple notifications that had to be carried out and that were often forgotten about. If these notifications from the companies are faulty, the notification of the department will not reach everyone concerned.

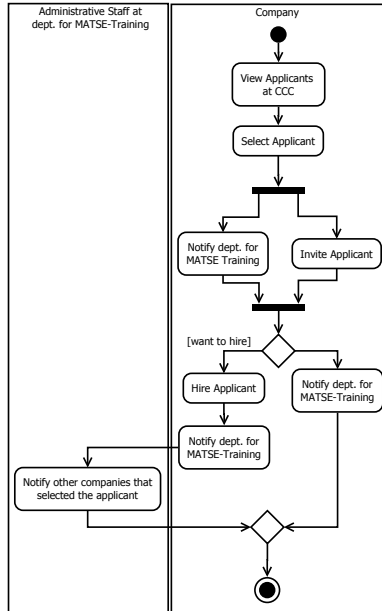


Figure 11: Activities taken to hire a new apprentice, before the introduction of the online application system.

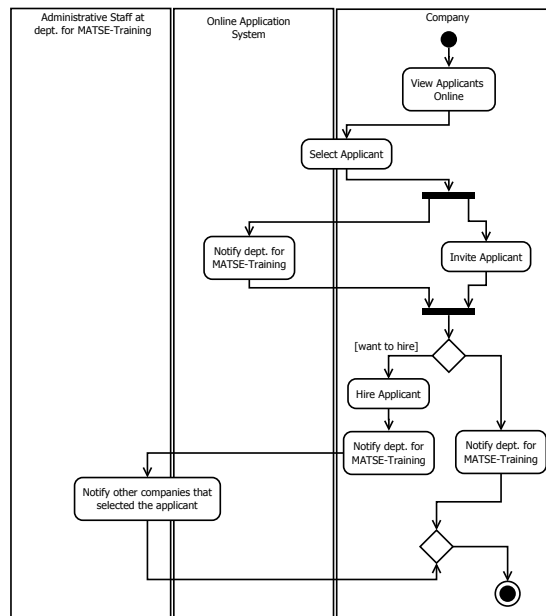


Figure 12: Activities taken to hire a new apprentice, after the introduction of the online application system.

The online application system will support the following steps in the process as displayed in Figure 12:

Select applicant The online application system helps the companies' representative to select applicants from the pool by sorting and filtering. Once the applicant is selected, the system automatically informs the department.

Notify other companies that selected the applicant The list of interested companies is still maintained manually but in future implementations the system will automatically maintain this list.

4.2 Implementation

After the analysis of the application process the implementation of the online application system was performed in three steps:

1. Gain (electronic) access to application data such as name of the applicant, application date and application papers
2. Implement a functional prototype that is provided to a group of friendly users
3. Given the feedback of the friendly users improve the system before it is introduced generally

Fortunately the application papers of nearly all MATSE applicants are submitted electronically so they do not have to be manually scanned by a staff member.

4.2.1 Data Access

The first step was dedicated to finding an automated transform of the data needed by the software. In general all the application data is stored in a central customer relationship system. The system used by RWTH Aachen University is a modified version of CAS genesisWorld (GW). It contains all personal and contact data of the applicants and a set of documents like the application papers. Apart from the data of the applicants GW also contains the data of the companies and their advisors. The data stored cannot be accessed directly by the web service but has to be accessed via a predefined interface.

The interface specified for the online application system is an XML interface that uses password protection and HTTPS. Since the XML interface was implemented by the department CAMPUS of the CCC it had to be specified in advance. The CAMPUS department administers the GW servers and is responsible for different extensions made to the system.

Using an XML based interface clearly separates the data from the processing part, which is a desired property in general. Nevertheless this interface is quite static and offers less possibilities than an SQL interface. Furthermore the XML interface has the drawback that it cannot access files saved within GW. The choice for this interface was also made since other modules of the *MATSE-Dienste* use similar interfaces as displayed in Figure 3.

To give the online application system the possibility to access the files saved in GW the Component-Object-Model-Server (COM) offered by GW is used. The COM server is accessed by a client that searches the documents associated with the applicants. The online application system accesses this client using a SOAP interface. Again this approach clearly distinguishes between the processing and the data access.

Both the XML and the SOAP interface provide another significant benefit for the software: Since the data is clearly separated from the application itself there is no possibility that the data is accidentally changed by an error in the system. Furthermore this approach allows the online application system to go without any storage or database. This issue made the implementation of the system very attractive to the university administration and the commissioner for data protection.

Listing 2 shows an example of the data provided by the new XML interface. This data mainly consists of personal data such as name of the applicant and date of the application. additionally it contains the GGUID, a unique identifier for the dataset. This GGUID is also used to identify the applicant when using the SOAP interface to access the application papers.

```
<entry Zugehoerig="" Name="Sample" Vorname="Susan"
  Datum_Kenntnistest="2011-12-13_00:00:00"
  Note_Kenntnistest="45" Bewerberberuf="Mathematisch-
  technische/r_Softwareentwickler/in" Geburtsdatum="
  1992-09-23_00:00:00" BewerberGGUID="0
  xB8EBBFD2E473CA48AC661D9FCB878744"
  AusbilderCampusGGUID="" AusbilderGGUID="0
  x49DC3B052B528E4B9EF3BADAC3D0945A" Zustimmung_Firma
  ="" MATA_Nummer="2011023" Status="Bewerber"
  Zeitrahmen="2011" Rueksendegrund="" />
```

Listing 2: Example of a line of the XML interface describing one applicant.

4.2.2 Web Interface

The web interface that was designed for the online application module aimed at two goals:

- **Easy usage:** The system is mainly used by non-technically experienced users. That is why it should be easy to use and display clearly what happens when the user takes a specific action.
- **Security:** The system deals with personal data of about 3000 applicants. Even though the users have to accept a disclosure agreement before they are allowed to use the software, it is important that the software only allows actions that are compliant with this agreement.

Figure 13 shows a screenshot of the developed web interface. On the left the applicants for a certain vocation are listed. Furthermore the user can select other vocations if he is allowed to. On the right side the selected application is displayed. To be able to control the display and printing capabilities of the application paper preview, it is rendered on the server and sent to the client as an image. Using CSS style sheets and JavaScript these images are prevented from being downloaded or printed and thus can only be displayed on the screen. Furthermore this approach has the advantage that the images can be displayed on portable devices such as a tablet or a smartphone.

The separate download button automatically triggers a notification that the applicant was selected by the user. Once the applicant was marked as selected the application papers may also be printed. This procedure allows the system to track the users that are interested in an applicant and thus to inform them if the applicant was hired elsewhere.

4.3 Introduction of the System

One month before the actual introduction of the application module it was tested with a group of friendly users. This group of users consisted of about 10 people, mostly technically experienced advisors. Based on their feedback and experiences with the system it was further improved and adapted to support their selection and hiring process as much as possible.

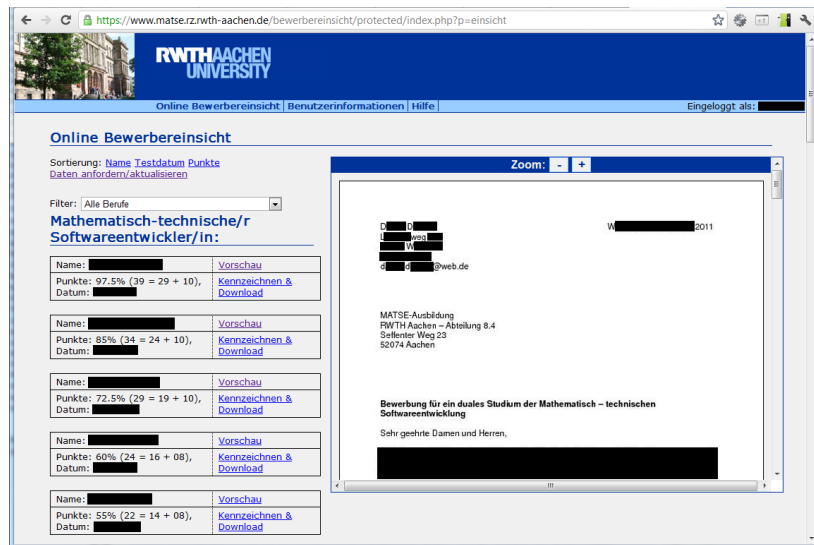


Figure 13: Screenshot of the web interface of the online application module.

The online application module then was officially introduced in July 2011 to the university administration and to all the companies who wanted to hire MATSE apprentices. During this phase the module was available to a wide variety of users for the first time. Especially employees of the personnel departments are not always able to cope with minor technical issues. Nevertheless most of the issues in the module they found could be improved.

During this phase Division 8.4 (Vocational Training), the central institution that coordinates the vocational training at RWTH Aachen University and therefore also the MATSE-Training, tested the system. Even though it was initially planned to have a longer phase where only the applications for MATSE would be handled by the online application module, Division 8.4 decided to introduce the system also to the other vocations starting in September 2011.

As already mentioned in Section 1.3 this decision was in conflict with the initial planning of the internship. Nevertheless it was incorporated with great success. The system is used since September 2011 by the entire university to find applicants for the 25 apprenticeship vocations and the total 750 apprenticeships jobs.

5 Conclusion

During the internship at the department for MATSE-Training I could gain insight in the software engineering process of a web based student self service

application, the *MATSE-Dienste*. This process was mainly driven by the given deadlines and the requirement, that the live system had to be available at all times. This requirement to the availability put restrictions on the way the software could be tested as fixed release cycles had to be satisfied.

Two modules for the *MATSE-Dienste* were developed: the course election module and the online application module. The course election module was extended and as such my implementations had to comply with the existing parts of the module whereas the online application module had to be developed completely from scratch and thus allowed more freedom. By participating in the development of both modules the complete software engineering process and life-cycle management could be observed during the internship.

Apart from the technical implementation I was able to take part in the process of requirements engineering for both modules. This included regular meetings with the customers and the other developers. While the course election primarily had internal stakeholders, for the online application module the stakeholders were the administrative staff of the university and the companies.

Last but not least both of the modules were finished in time and especially the online application module has been very successfully introduced, not only for MATSE applications, but for all apprenticeship applications of the University. Currently the online application module offers a total of 3000 applications. Furthermore the course election module will be used by the 300 students for the first time in week 4 of 2012.