Faculty Elektrotechnik / Informatik

UNIKASSEL
VERSITÄT

Master Thesis

Homework Manage System with Git-Support and hierarchical File database for Data management

Submitted by: Hao Gao
Matriculation number: 33101387

Supervised by: Prof. Dr. Albert Zündorf
Universität Kassel
Prof. Dr. Gerd Stumme
Universität Kassel

Kassel, December 3, 2015

Declaration of Authorship

Deutsch oder English?

Wenn dann English

Kassel, December 3, 2015

Hao Gao

Summary

zusätzliche Zusammenfassung über Deutsch nötig?

Contents

Bitte nach hinten

List of Figures

Bitte nach Hinten

weg

Besser alle COdesnipsel und Bilder als List of Figure zusammenfassen

List of Tables

1 Introduction

Chapter 1

motivation, why we need this kind of hms?

Weil cool
Hier gerade was der Wissenschaftliche Gedanke ist. Was ist neu und wurde noch nie betrachtet

Chapter 2
2 Basics

what were used in this project?explain the basics of the techniques

2.1 techniques like framework and jpa?

or?

2.2 Related works

moodle?

Studentenprojekt HDS, Fohry Abgabesystem

3 Chapter 3
Design

Als Erstes sollen die Inhalte zu einer Vorlesung veröffnetlicht werden und die Hausaufgaben inklusiv Abgaben und Bewertungen verwaltet werden.

Home Management System

The HMS is a web platform to manage the activities related to the homework. In this chapter the detailed design of functions to support the activities will be discussed.

The functions of the "HMS" system are divided into two parts. The first part is to develop the common features which are similar to other web platform,for instance "register a user account" . The second part of the design is to develop the special core features to distinguish the HMS system from other platform, the dynamic database and the git based file management.

3.1 Common functions

The main task of "HMS" is handling the process of handing in and handing out the homework between the students and teachers.

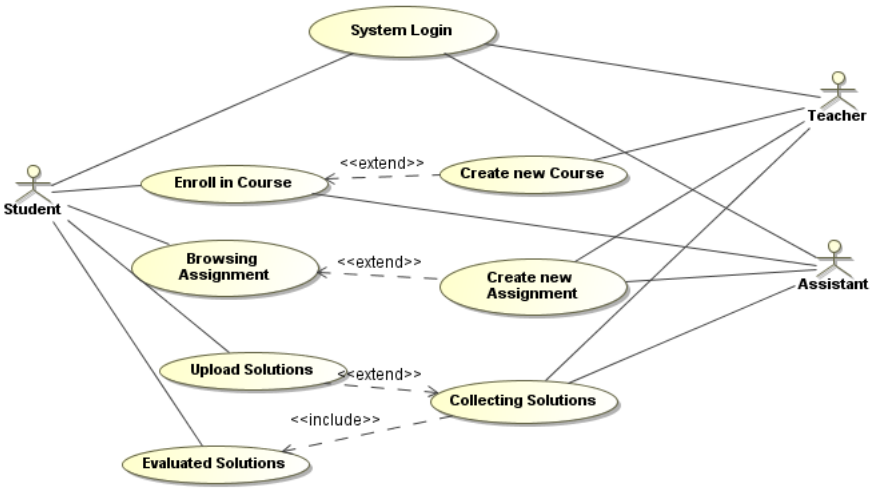


Figure 3.1: The use case from the perspective of different roll of user.

As the Figure 3.1 shows,there are several steps within this process from the prospective of the user roll. First considering the user roll of students,the student should be able to

register a account of "HMS", after logging into the system the students can browsing all the available courses in the system and subscribing the target course. Then it is possible for the students to view the homepage of the course and download the available homework. and finally uploading the solutions to the homework accordingly. the process of uploading a homework to the system is finished at the student side. now take a look at the side of Teacher and Assistant, besides the registration and logging process, the user group of teacher can create new course and new assignment also collecting the handed in homework and give them back to the students once the evaluation is finished. The assistants however can not creating new course, but should be able to add new assignment and evaluate the handed in homework as well. In summary, in order to realize the main task of the "HMS", the "HMS" system should have following capabilities:

1. User management including "Registration" , "Role based access control","Self management".
2. Course management including "Creation of Course","Modification of Course".
3. Assignment management including "Creation of Assignment","Distribution of Assignment","Collecting of Assignment","Evaluation of Assignment".

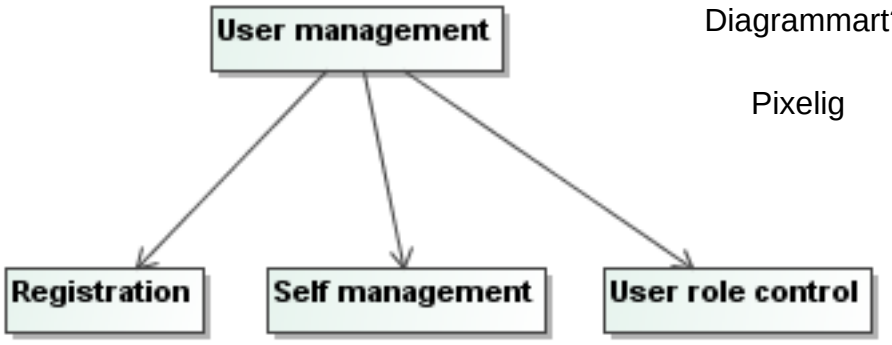
Besides the above listed capabilities, the HMS should also provide a way that the students and teachers can communicate with each other. firstly a private message system will be needed for the students and teachers to exchange the information about the problem of assignment or evaluations individually,also the new message system should work as a instant message system,in this way the questions or the problems between the students and teachers can resolve more efficiently. secondly, a course forum is also not a bad idea, a common scenario is that more students may have a same question for a new assignment, if a students write a new post about this question, and the teacher gives a answer to the question, the other students with the same question can also get the answers and avoiding asking the same questions again. This saves the time from both side. So the communications system for the HMS has two parts:

1. Instant private message system.
2. a public course forum for each course.

3.1.1 User management

The user management of the "HMS" system consists of several modules(Figure 3.2), first is the registration module, with this module the user can use their email address to register a account in HMS system. second is the self management module. after the user has logged into the system, they should be able to change their emails and

password or other personal details.third is the user role control module, this is necessary for system to arrange the proper functions to the current user based on their user role, the user obtains a startup role at the registration, later on the user role can be changed by the system admin. In the rest of this section, all the modules will be detailed discussed.

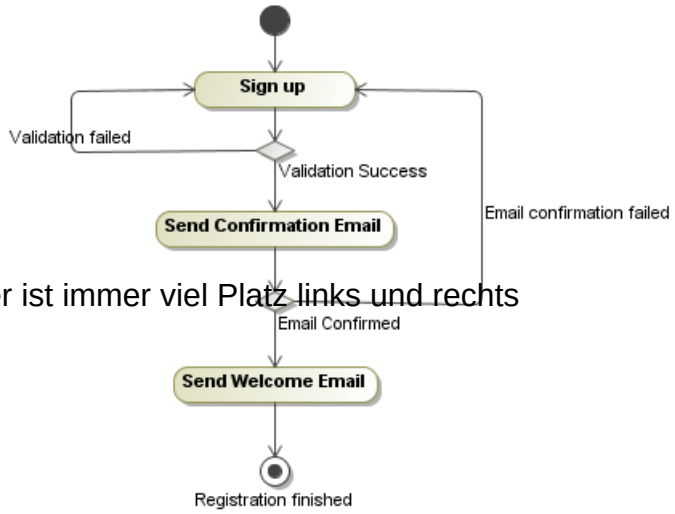


Diagrammart???

Pixelig

Figure 3.2: The Modules of user management

a. Registration



Bei solchen Bilder ist immer viel Platz links und rechts

Figure 3.3: Registration activity

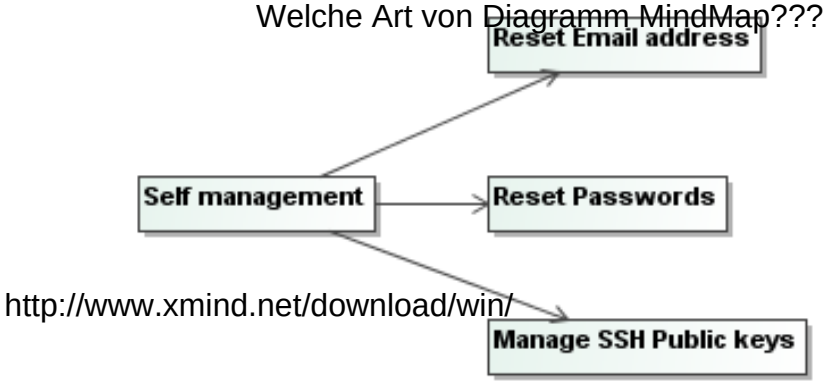
Figure 3.3 shows the workflow when a user register a new account. the user first use the registration form to fill in all the relevant information , for instance the password, email, and students number, after clicking the sign up button, the whole process begins, first the validation of the registration form will be performed to check whether the user has fill in all the required field and without error, if user passed the form validation, the

HMS system will then send a confirmation email with a confirmation URL to the email address from the registration form, if the user click the confirmation link, the user will be redirect to the website and can directly starting using the account. Otherwise the user has to start over the registration process. The step of email address confirmation is important because this procedure allows the system to check that the user actually signed up for the account and guarantee the email of this user is valid and ready to receive the system information.

b. Self Management

https://gitlab.se.cs.uni-kassel.de/homework-management-system/hmsplay/blob/master/HomeworkManagementSystem/docs/HDS.xmind

Welche Art von Diagramm MindMap???



http://www.xmind.net/download/win/

Figure 3.4: The functions of self management

It is very common that user may forget their passwords or even worse their registered email becomes invalid, so it is necessary to develop the functions, user can use to reset their email and passwords. The resetting of email address or the passwords works similar as registration. If user choose to reset their email address, first they will be asked to type in the new email address,after that user click the reset button, the system should send a new confirmation email to the new address, when user click the hyper link in the email, a new web page will be generated and user can confirm the address change. if user choose to reset their passwords, they just need to click the reset password button, the system will send another confirmation email with a hyper link, the user can use this link to type in their new passwords. Besides the modification of passwords and email address,there is another function should be added to self management, thus the HMS system use git server to control all the homework files, and this git server use ssh to authenticate the connections, the system should also provide a function that user can add their ssh public key to the git server, so that they can connect their computer direct to the git server. Figure 3.4gives a overview for all the components of the self management.

c. Role based access control

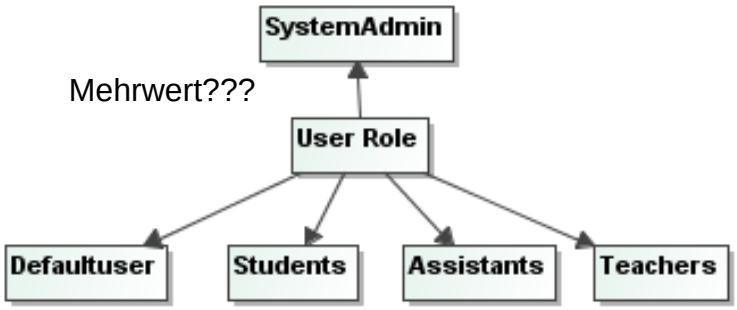


Figure 3.5: Different user role in HMS system

The user of the HMS system has different roles, and they need different functions. for instance a teacher can create new course but students can not, so it is important for the system to distinguish the user based on their user roles, so that the system can serve different functions to different user.

Figure 3.5 shows all the user role in HMS system,from the top is the system admin, the system admin account will be generated at the first boot up of the HMS,the user name and password of the system admin will be saved under the home directory of the current server admin. it means only the admin of the server can access the username and passwords of HMS system admin account, this can ensure that only trusted personal can access the system admin account of HMS.The job of system admin is to manage other user's role and the system database(backup the database after semester ends). At the bottom are the normal user roles. first is the default user,every user obtains this role after registration.if you don't give a students number.The user with this role can not do much things within the system other than updating their email address or passwords. if the default user wants to promote their user role to another level, they have to contact system admin. second user role is students,if user register the account with their students number, they will automatically obtain a student account, with student account the user can browse all the available course and join the course, download and upload homework, using the communication system like chat and forum. third user role is assistants, besides all the functions of students, assistants can create new assignment and review all the handing in homework and make a evaluation,last one is the teacher, the teacher account has all the functions of assistant account and additionally the teacher can also create new course.

Geht definitiv kürzer. Viel Blabla
Wenig mehrwert

3.1.2 Course management

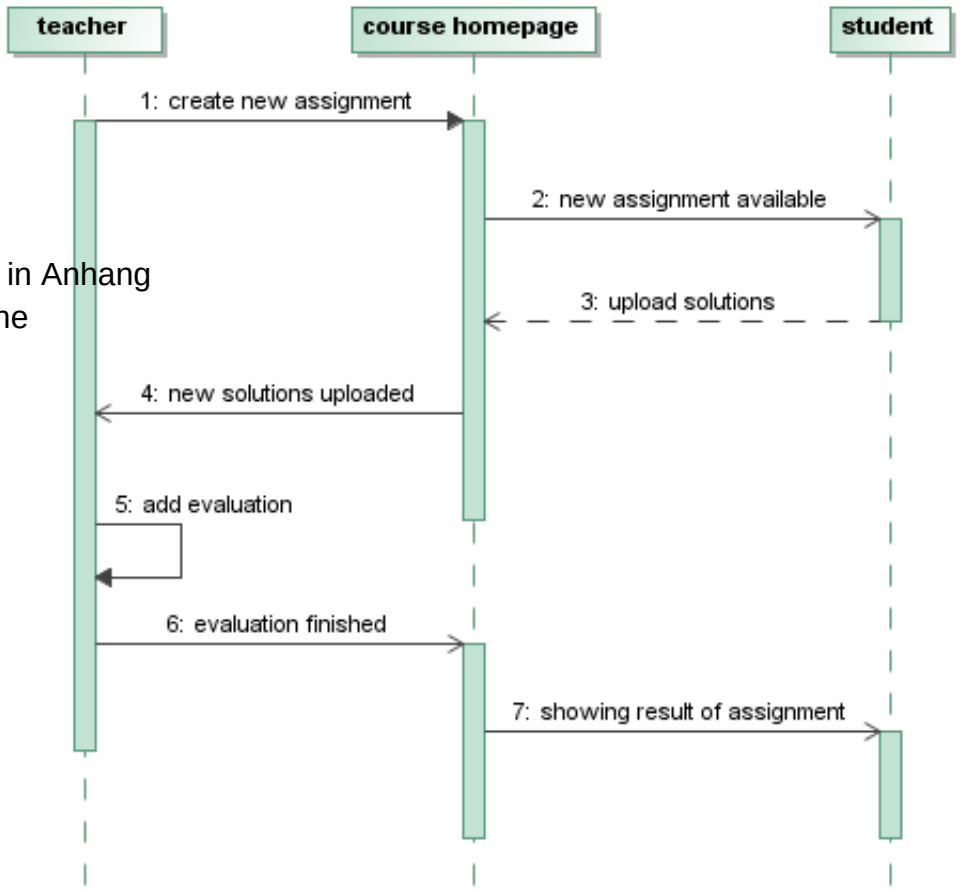
There are two types of course for informatics students, in the first type of course the students have to hand in various homework, and the points gained from those homework are usually used as a prerequisite for the final exam. in the second type of course the students will get a semester assignment, normally a whole software project, the points gained from this project usually is the final points for this course. additionally every students will get a git repository after user had signed up the course,this repository will also worked in two modes according to the type of course,the details of git working modes will be discussed in more detail.

Features	Precondition to final exam	Git repository Mods	Performance Evaluation
Type I	1. Students have to hand in at least amount of valid homework. 2. A valid homework requires normally for students to gain more than 50% points of a assignment. 3. A student should gain at least 50% of total points for final exam	Git repository works under local modus (Student can only hand in the homeworks through course homepage)	need detail evaluation of assignments (number of valid handin, percentage of gained points)
Type II	None,Students just need to hand in the final project	Git repository works under remote modus (Student can use the course repository as any remote git repository)	Only final evaluation Ausblick: Dort gibt es auch ggf. Zwischenbewertungen

Table 3.1: Features of different types of course

The Table 3.1 shows the different features of different types of course.The function of "creating course" should take all the features from above into consideration.

Bachelor PM: Klausurzulassung
Bachelor SE1: Studentenprojekt außerhalb kiene Abdeckung im HMS
Master SE2: Endnote mit 1. Abgabe
Master Compilerbau: Endnote mit n-Abgaben
Master Graph-Model: Klausurzulassung
Bachelor Deisgn-Pattern: Endnote mit n-Abgaben



Kürzer oder in Anhang
weniger Höhe

Figure 3.6: Sequence diagram for assignment management

3.1.3 Assignment management

Figure 3.6 shows the sequence of handing in and handing out a assignment, after creating a new course, teacher can start adding assignments to the course, when a new assignment has been successfully added to the course, the course homepage for students will show this new assignment and with a download link, student can download the assignment and start working on the new assignment. later when students finished the assignment,the solution will be uploaded to the according assignment and teacher can review them at the evaluation part of the course's homepage and add a evaluation to the solution. When the evaluation is successfully saved to the assignment, the student will get the results directly at course homepage.

3.1.4 Communication system

a. Forum

The HMS system has a standard client-server structure, the client and server communicate with each other over internet using HTTP protocol.[2] HTTP has a typical "Request-Response" pattern, the web client sends a request to the web server, web server serves a response according to the web client request. It is a simple but powerful solution to provide a two-way conversation for two party over one channel.[3] The forum function within the HMS system works also after this pattern, Figure 3.7 shows the communication between the clients and HMS server based on the different forum action request.for instance: a client sends a "create new post" request to the server, the HMS server accepts the request and save this new post to the related course forum, and return the content of this post to the client again.

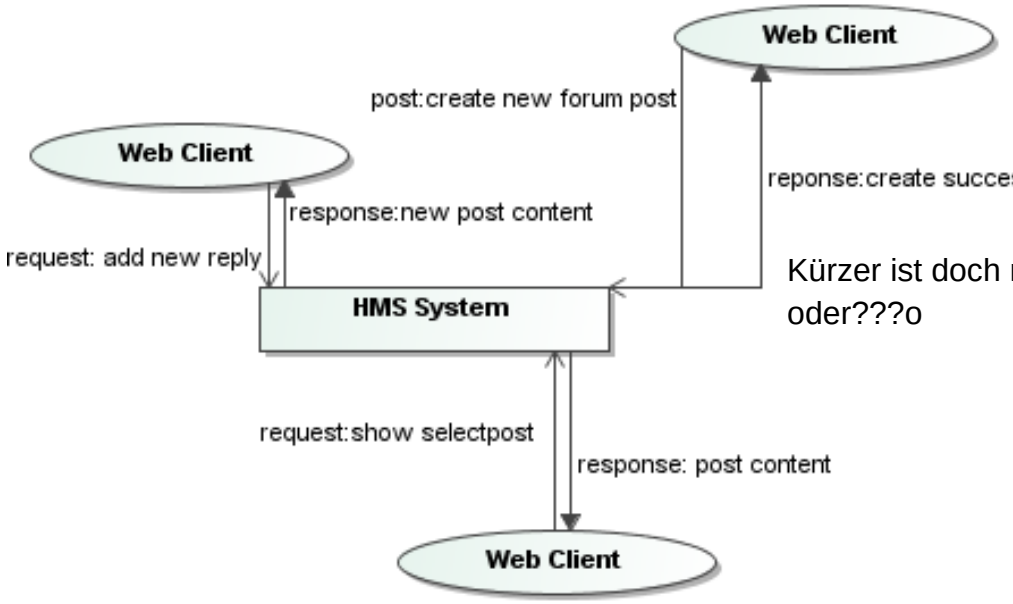


Figure 3.7: Forum message pattern

b. Instant message

The standard HTTP protocol however is not suited for the instant message system, because of the "Request-Response" message pattern, user has to manually ask for a content upgrade. But a instant message system needs automatically update the chat contents on both side of a conversations while a new message has been added[4]. therefore a full-duplex communication system "Web Socket"[5]will be used to back up the message module. Figure 3.8 illustrate a simple use case of web socket,after logging

into their HMS account user A and user B are both connecting to the HMS web socket server, later on user A send a new message to user B,first the request from user A are passing to the web socket server,the server processed the request accordingly and served the response not only to the user A but also automatically served the response to the user B,this ensure the both side of the conversation can have their message received in real time.

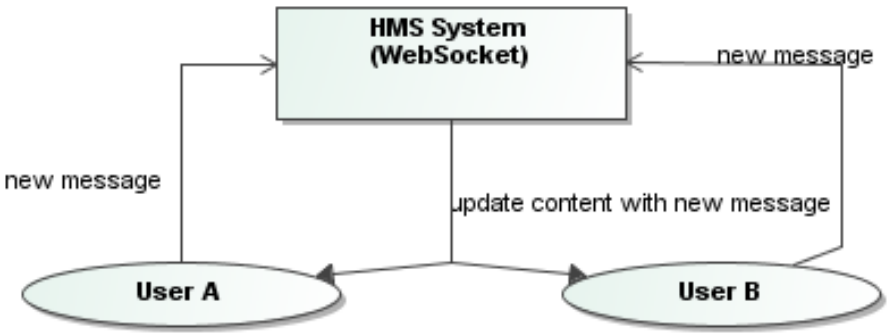


Figure 3.8: Web socket message pattern

3.2 Specific functions

The functions from last section needs two critical preconditions to work:

1. The files of assignment and student's submissions can be saved.
2. The result of the activities can be persistent in a database.

There are already plenty solutions to those problems,for instance the file upload and the database support are the standard functions within the play framework. The developer just need to decide where to put the files and configure the configuration file for database to work. However there are still some disadvantages with the standard approach,this will be discussed in the rest of this section and the new approaches for both will be introduced.

3.2.1 Submission management

The normal approach for the file upload in Play framework is pretty simple,the developer only need to defined the extra file attribute for a HTML form and send the form through POST methode of HTTP protocol,and defined a java controller on the server side to save the file to a desired location. Also if the user want to retrieve this file again, the developer just need to save the path of this file under the name of the user in a

database. But when considering this file is a homework submission,there is still need a lot work to do than just upload a file to the server. The documentation of the moodle system has suggests several submission type related to file submission[6]:

1. Student submit a work and teacher download it later.
2. Student submit a work multiple times.
3. Student submit a work with a response.

The first type can direct use the default file upload function with a database recording the file path. For second type there are two ways to solve it, first way is that every time when students submit a new version, the old version will be deleted, and teacher will always get the latest version,the downside of this solution is that the previous upload will be all gone,there is no way for teacher to track the submission history. another is that when a new version is submitted,the old version will not be deleted,however with this solution the teacher do can track the history but will taken two much hard drive capacity.For the last type it is just need to put a extra column in the database to save the response with the file path.

All these submission type can be solved just using the standard techniques. But this is still too complicated, not only for development but also for the user of the system. At this point the GIT will be introduced to resolve the problem.

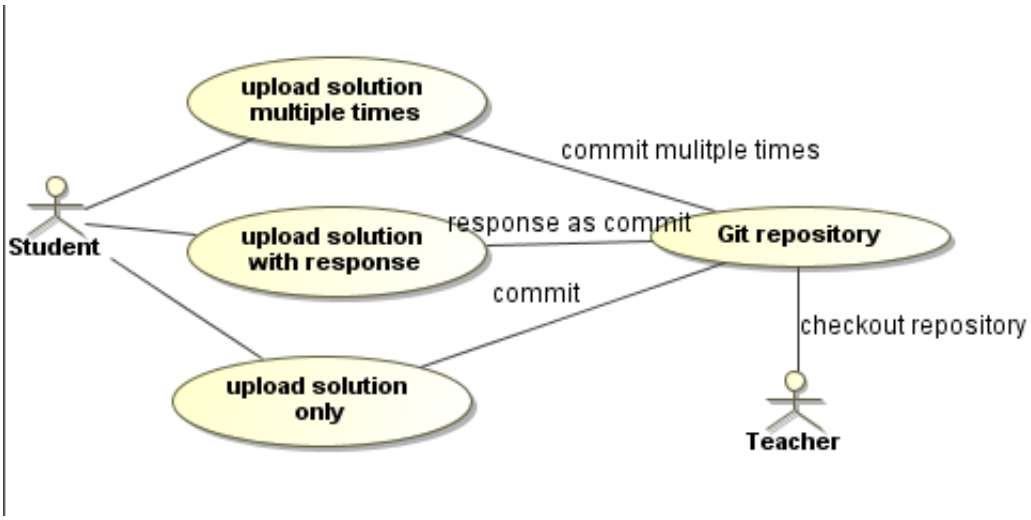


Figure 3.9: Workflow of git

Figure 3.9 shows the workflow for this 3 type of submission when using the git repository. Every students when they join a course, they will automatically get a git course repository from the system,all the submissions of the student will be written into this repository. all this scenario from above can be solved just using the git commit. no matter what submission type was chosen by the students,the only difference how many commits

were made. on the teacher side they just need to checkout the students git repository and read the commit history. naturally the repository address is also needed to be saved into the database.

3.2.2 Database

Play frameworks support several database engine:

- H2
- SQLite
- PostgreSQL
- My SQL

Memory oder File???

only H2 and SQLite support the embedded mode,the embedded database runs directly in the application that uses them,it requires no extra server and no maintenance for the database itself.Another advantage of embedded database is the speed, because all the database operations happen inside the application process.[7]In this project the H2 database engine will be used as the default database engine. first h2 support the embedded mode and it is purely written in java, the java API of h2 can be directly used in the play framework. another reason is that the embedded database runs within the application, it is normally hidden from the end user[7], so there is no way the user can side load the database, but in the real life of maintaining the HMS system, direct accessing and manipulating the database using a SQL query tool sometimes is more efficiency than the usual routine.in this case h2 support a mixed mode(Figure 3.10),mixed mode is a combination of the embedded mode and server mode, the first application (in this case the HMS) will use the database as embedded mode, but it also starts a server so that the other application(a SQL query tool) can still side load the database.It is also importent to notice that if the application is shut down, the server mode will also close all the connections[1],therefore side loading a database using remote mode can only take place when HMS is still rung. However the database file generated by the h2 engine can still be loaded with the h2 web based manage tools.

Play framework uses Ebean ORM to access the database,ORM is a technique to convert objected oriented programming language(in this case Java) into its persistence as a relational data base, so the data can freely exchange between a java object and database table.[8]By default the configuration of Ebean and the database is done by editing the configuration file of play framework. After configuration of the database engine and the Ebean, the system is ready for developer to design the database for the web application.

This is the standard approach when developing a web application using play framework. But it also has a problem, the database configuration has to be done before the

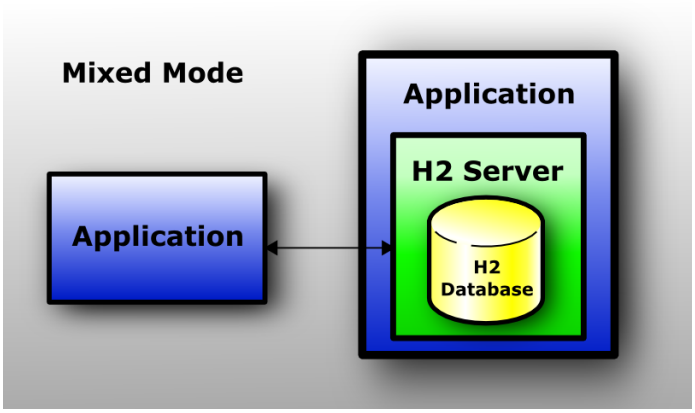


Figure 3.10: Mixed Mode of h2 server[1]

application started, in the run time of the application all the database related actions can only take place within the preconfigured database. This approach is convenient for the developer, because the developer don't need to take care of the programmatically details of bring database ,ORM and application together.But it sometimes cause trouble for the end user. Usually there will be only one predefined database, all the data from the web application will be saved within this database. In this project, HMS is designed for managing the homework for teacher,and the system will be used in a university,every once in a while the university need to archive or backup the old data from the past term. A possible way is query out all the related data based on the semester, then dump the data into file and save the file some where else. Moodle also use this approach to make a course backup[9].The disadvantage of this approach is strait forward,the amount of data will increasing rapidly after years of use, it is not only complicated but also consume a lot of time.

Genau das ist das spannende

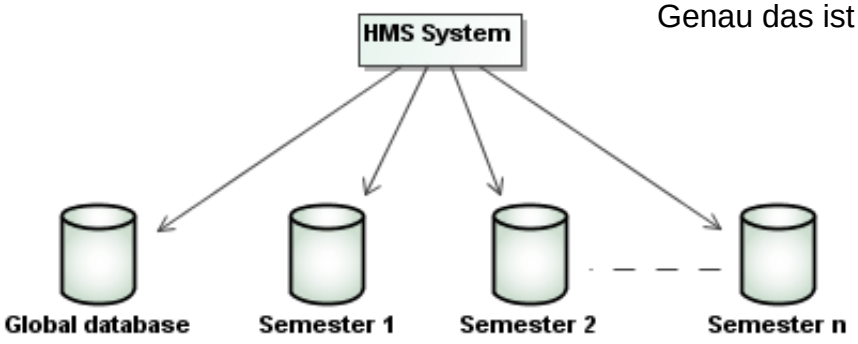


Figure 3.11: multiple database based on semester

Figure 3.11 shows a better approach used in the HMS project. all the data can be separate saved on different database based on the semester, the back up process will a lot easier.First the data structure of every semester is the same, so the database design for semester can be reused in every new semester,therefor no need to query the data.

second the database in this project working under the embedded modes,all the relevant data of the database are saved in a single file. so each semester will be a single file, backup the semester data only need to copy the database file to some where else.

aside from the semester data, there is a additional database to manage all the data related to the authentication process, for instance the email address,password,and ssh public keys.

This approach requires the system to create a database in the run time, therefore the standard database configuration of play framework can not be used. The details of implementing the dynamic database function will be demonstrated in the next chapter.

Mehr über dieses Thema
Wo genau werden die Daten gespeichert, was passiert mit alten Studenten
wird die globale Datenbank aufgeräumt etc. werden die Basiszugänge verdoppelt

Chapter 4

4 Implementations

how to implement the functions from chapter4

4.1 User management

4.2 Course management

4.3 Assignment management

4.4 Communication system

4.5 Integration of Git

4.6 Database design

Chapter 5

5 System usage

Test run?

5.1 First run

Chapter 6

6 Evaluation

Chapter 7

7 Summary and Outlook

Zukünfig komplett inklusiv upload Videos etc. Plugin-System

Nomenclature

HMS Homework Management System

ORM Object-relational mapping

Bibliography

[1] *H2 Documentation*. http://www.h2database.com/html/features.html#embedded_databases

[2] Microsystem, Sun: *Distributed Application Architecture*. 06 2009

[3] Hohpe, G. ; Woolf, B.: *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Pearson Education, 2012 (Addison-Wesley Signature Series (Fowler)). https://books.google.de/books?id=qqB7nrrna_sC. – ISBN 9780133065107

[4] Day, Mark ; Rosenberg, Jonathan ; Sugano, Hiroyasu: A model for presence and instant messaging. 2000. – Forschungsbericht

[5] Mozilla: Glossary:WebSockets. (2015)

[6] *Documentation of moodle*. : *Documentation of moodle*

[7] Chaudhri, A.B. ; Rashid, A. ; Zicari, R.: *XML Data Management: Native XML and XML-enabled Database Systems*. Addison-Wesley, 2003 https://books.google.de/books?id=7LNhdOeQulQC. – ISBN 9780201844528

[8] K, S.K.: *Spring And Hibernate*. McGraw-Hill Education (India) Pvt Limited, 2009 https://books.google.de/books?id=NfNbbhBRcOkC. – ISBN 9780070077652

[9] *Moodle documentation:Course Backup*. https://docs.moodle.org/30/en/Course_backup