

WebTUTOR – A Web-based Personal Study Plan Tool

Harri Hämäläinen, Jari Porras, Kimmo Koskinen

*Laboratory of Communications Engineering, Department of Information Technology,
Lappeenranta University of Technology
PO Box 20, 53851 Lappeenranta, Finland
Tel.+358 5 621 2801; Fax.+358 5 621 2899; E-mail. {harri.hamalainen, jari.porras,
kimmo.m.koskinen}@lut.fi*

Abstract

Student guidance from the beginning of their studies until the graduation has proven to be rather multi-phased process. Tutors take care of students by giving them general advice in the beginning of the studies. Later on they will have to make the decision between major and minor studies. Before graduation the curriculum needs to be accepted by the administration and all the papers need to be delivered in the right places. It can be said that guiding a student from the beginning till end requires quite a lot of human resources.

We have been developing a tool that should decrease the required amount of supervision and administrative work. Study planning in early phases of the studies and its constant updates will also decrease the unnecessary work by automatically guiding the student to make reasonable choices while selecting his studies. This should finally decrease the time required for graduation.

1. Introduction

Currently we are living an era when many PSP-systems are being designed. We are no longer too far from the situation when

software based guidance will be a natural part of student guidance at different universities. This not only gives opportunities but also creates challenges for those maintaining and participating on the process.

Personal study plans can roughly be divided into two different categories: the one where students set goals for their studies and knowledge (the more humanistic approach) and the one where students plan which kind of studies will they include in their curriculum (the technical approach). We have been developing our application more for the latter usage. From a tutors' point of view it provides more concrete tool compared to the humanistic view that is more likely aimed for students' personal use only. On the other hand the technical approach does not necessarily support setting an overall goal of improving yourself. Moreover it should also be considered, what kinds of benefits can those students gain who are in their first steps of academic studies. Their knowledge of the profession may still be in such a basic level, that setting reasonable goals is impossible.

Our solution, WebTUTOR, is a web-based application for creating and maintaining student's Personal Study Plan (PSP). By using the application student can create himself a personal plan that is based

on the rules defined in the curriculum by his university and department. Besides selecting the correct courses to be included into his plan, the uncompleted ones will also be scheduled. At this point, the application is implied for being used as a supporting tool, helping also to create a path where all the required prerequisites are completed before taking a single course.

The tool is designed for students who are in different phases of their studies. Junior students can use the tool mostly for scheduling and selecting the proper courses, while those at the end of their studies can use it for the final approval of the curriculum. You may say that the difference between these phases becomes more transparent – a plan becoming the final transcript.

Many Finnish instances have been and still are developing systems for their own or public use. At the moment one of the most widely known is eHopo [1], developed by the University of Joensuu and publicly available for testing. It includes characteristic partly from both of the planning approaches.

2. System structure

WebTUTOR is a web-based application running on LAMP (Linux, Apache, MySQL, PHP) platform. All the information and the curriculum structures are stored on database. The currently used version of the application is installed on Linux-based server running Apache HTTP-server.

The application supports the usage of both local and external database. Installing the database on different server allows the same database to be used by multiple web-interfaces. One example of an occasion would be a public application server having multiple applications installed and running on it. In such an environment the usage of a separate database makes the WebTUTOR application more secure and more reliable.

While designing the system we have paid attention to independency of the application requiring no external libraries. Therefore it does not matter which kind of system is used e.g. for storing students' records. The architecture is described in figure 1.

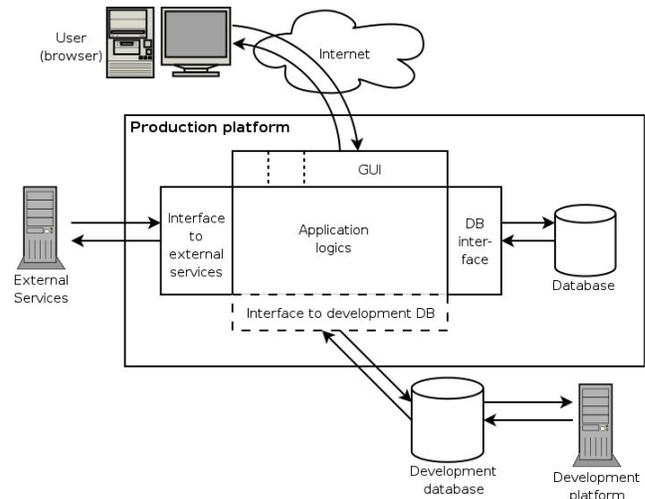


Figure 1. System architecture.

2.1. System Interfaces

Examinations and information of courses can be exported from the application either through the web interface or as an XML file. Different kind of reports can also be handled using the web interface or XML files that are based on the defined schema. Students can also import their transcript of records into the system as far as it fills the requirements of the defined schema.

Since the current systems containing the data that is needed in our application are missing public interfaces, there has not been use to complete one to ours so far. Nevertheless the fact that we are using XML to provide information enables the usage of e.g. Web Services easily in the future if needed. In this case the application naturally requires some minor changes to be done. It also has to be considered that some systems might even not want to share their information for others for some reason.

2.2 Data modeling

All the data is stored in a database that can be divided into three major groups: curriculum data model, student curriculum data model and system's internal data model. Curriculum data includes all the curriculum bases and course information. Student model includes the tables where all the user's curriculums are stored. System tables are used to handle data that is necessary only for the system itself such as user accounts and sessions.

The curricula are modeled in a tree-hierarchy. A simplified example of this model is presented in figure 2. The examination is located at the top of the model. It consists of separate groups that may include more subgroups and/or courses. All these units have certain parameters, such as amount of credits, that are used to model the real-world as well as possible and to make the structure work on different kinds of occasions.

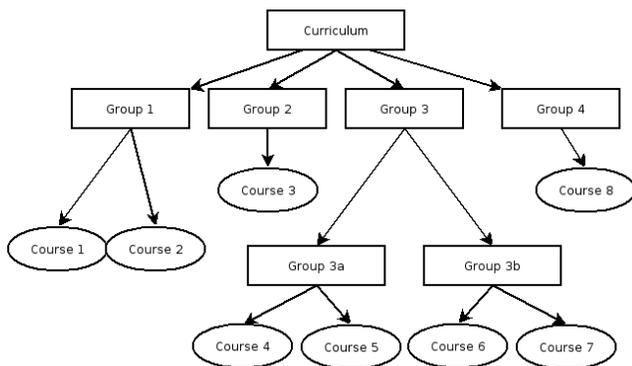


Figure 2: The model of curriculum structure.

2.3. User Interface

The application is used through a web-browser. Users are divided in three different roles: students, tutors and administrators. Each of these groups has different rights on the system, which basically means own interface and differing access for all to the information stored in the application.

All the produced data is validated HTML that follows the definitions set by W3C [2]. Cascading Style Sheets are used to define the layout and style of application. Also the layout and logical functions are to be kept apart from each other to allow easier layout customizing for each possible instance using it.

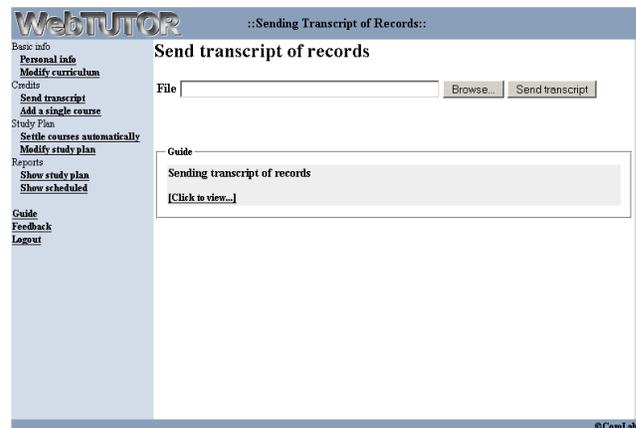


Figure 3: Screenshot of student's view.

The design of user interface in general includes mapping the end user and how the system will be used to support the actual task to be done. [3] In our case dividing the users in three major groups was easy: students, tutors and administrators. Students' group can still be divided into minor subgroups. Since the use is mostly likely to be quite periodical for most of the users, the goal was clear: to create a rapidly adoptable interface that is easy to use even in the first time. In this case efficiency of committing actions is not that important.

To acquire the settled objectives we chose the traditional navigation-centric layout model represented in figure 3. The navigational functions are placed on the left while the content still has the most of the available space. Some less important information, such as logo and copyright notice, are placed on their own cells.

2.4 Authentication and sessions

Currently users can be authenticated using either the local database or RADIUS authentication service. In most cases using RADIUS, if available, is more preferable because of centralizing the user accounts that leads to decrease required work for maintaining the WebTUTOR. After a successful authentication a new session is created for the user.

Session management is done with the help of the session functions included in PHP. Each session has a personal 25-character-long identification value that is used to validate every incoming request and to identify each user. The session ID is stored in a cookie, if possible, or included in each POST and GET request as a variable. Each session expires after 30 minutes of inactivity. Session create a potential security risk so the functions and model have to be designed carefully.

2.5 Security

Being a publicly available service on the Internet that requires authentication and stores user's private information, security has to be considered seriously. Security can be divided in three different subgroups: network architecture and transmission security, platform security and security of the application. Transmission and platform security are to be considered to be tasks for server administration and do not affect a lot to application development. The application security instead has to be considered by the programmer himself while planning and developing the application.

WebTUTOR is protected against the most common attacks used against web-based applications including Cross-Site Scripting, Cross-Site Request Forgeries and SQL injection based attacks. Therefore all the user input is controlled before handling it or executing any actions that include user submitted data. Chris Shiflett on his article

[4] has pointed out the five most important things while handling the data sent by users:

- Filter all foreign data
- Use existing functions
- Only allow safe content
- Use strict naming convention
- Be creative

2.6 Double Submit Problem

As a program that is manipulating the data in database, some kind of anti double submit solution has to be implemented. This kind of a problem occurs for example, when users refresh their web-browser's view after executing an action that changes the state of the database (such as inserting, updating or deleting data). The solution for this was to use personal tickets for each session. This parameter is included in all the manipulating actions that a user can perform. After being used once, ticket is outdated and the next forms will include a new value of the ticket. Now if user resends once submitted and filled form, actions will not be performed and the database won't get corrupted.

3. WebTUTOR

3.1. Student tasks

Students are the main target group. They design their plans by themselves. First they select the structure, which will be used as a base for the plan. They have a possibility to do some limited modifications to it during the planning. After the structure is chosen, students import their current studies into the system and place them into the structure where those should be. Then the plan is filled with missing studies and those courses will be scheduled by the student. After all this is done, he should have a complete plan for his studies ready to be printed or sent to the tutor that can comment on it. After spending some time with studies, students can update their existing curriculum and do

some changes if necessary. The flow of procedures to be committed is presented in figure 4.

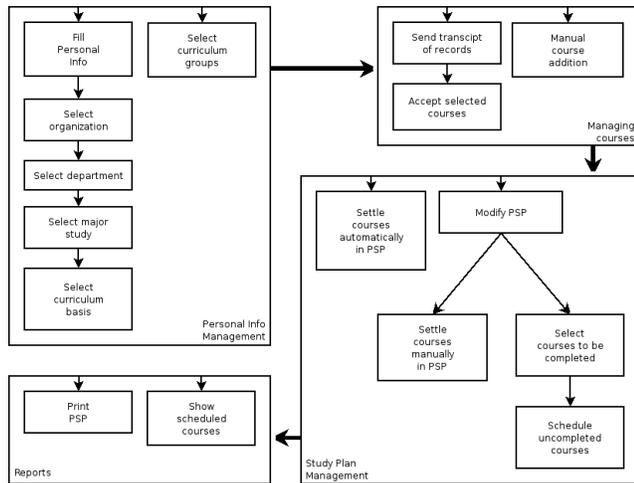


Figure 4: Navigational and action model for students.

In figure 5 is a student's view to his plan. For those courses that the system was not able to settle automatically user has a change to do it manually by selecting the proper group from drop-down box and submit the change. Finally when all the courses are included into the plan, it is easily updateable and ready to be sent to evaluation.

Nro	Opintojakson nimi	Arvosana	Ov	Huom.
070905000	Itäia 1	H	2	[Muokkaa] [Siirä]
070915000	Itäia 2	H	2	[Muokkaa] [Siirä]
PISA_ISS	Sistema informativi	H	6.5	Anotaan hyväksyttäväksi. [Muokkaa] [Siirä]
PISA_IT	Italiano Lingua Strazera	H	3.5	Anotaan hyväksyttäväksi. [Muokkaa] [Siirä]
Suositettu: 14, Suosittelua: 0, Yhteensä: 14 Opintovaihtokorvaus: 5.5				
PAKOLLINEN HARJOITTELU 6 ov (Tietotekniikan osasto)				
Opetussuunnitelman 2000 - 2001 mukaan.				
Nro	Opintojakson nimi	Arvosana	Ov	Huom.
H10	Harjoittelu	H	10	Valitse opintoryhmä... [Muokkaa] [Siirä] •Pienopinnot •Aineopinnot •Pakolliset opinnot •Aineopinnot
OPINTOSUUNNITELMAAN SIIJOITAMATTOMAT OPINTOJAK				
Nro	Opintojakson nimi	Arvosana	Ov	Huom.
010236000	Lineaarinen optimointi	1	3	[Muokkaa] [Siirä] •Sivuaineopinnot •Tietotekniikan sivuaineopintoryhmä
010765000	Tietokonegrafikan perusteet	2	3	[Muokkaa] [Siirä] •Tuotantotekniikan sivuaineopintoryhmä •Väpääsi valittavat opinnot •Pakollinen harjoittelu
020117000	Teknillisen piirustuksen peruskurssi	2	2	Valitse opintoryhmä... [Muokkaa] [Siirä]
030147000	Johdon laskentatoimen peruskurssi	1	3	Valitse opintoryhmä... [Muokkaa] [Siirä]
030370000	Laatujohtamisen peruskurssi	1	2	Valitse opintoryhmä... [Muokkaa] [Siirä]
R	Ruotsin kielen taito	T	0	Valitse opintoryhmä... [Muokkaa] [Siirä]
Suositettu: 13, Suosittelua: 0, Yhteensä: 13				

Figure 5: Placing a course as a part of student's plan.

In figure 6 we present a single page as an example of ready-made curriculum ready to

be printed. In our case it is divided in separate subgroups that include course information or more subgroups if necessary. The figure contains two groups each of which includes some courses.

Nro	Opintojakson nimi	Ov	Arvosana	Muuta
010113010	Matematiikka A	4	4	Korvaa opintojakson 010113011 Matematiikka A, 5 ov.
010113020	Matematiikka B	4	4	Korvaa opintojakson 010113021 Matematiikka B, 5 ov.
010307000	Tietotekniikan perusteet	2	5	
010511000	Ohjelmoinnin perusteet	3	3	
010961000	Johdatus korkeakouluopintoihin	0.5	H	
070304000	Tjasteveysla	1	3	
070801000	Puhevestintä	(2)		Kevät 2005 (Harjoitukset tarkastettavana.)
070925000	Itäia 3	2	4	
070940000	Itäia-elämästä	2	3	
080310010	Fysiikka T, osa 1	3	1	
080310020	Fysiikka T, osa 2	3	1	
080310030	Fysiikka T, osa 3	2	4	
080802000	Elektronikan perusteet B	2	2	
Yhteensä		30.5		
AINEOPINNOT 37.5 ov				
Opetussuunnitelman 2000 - 2001 mukaan.				
Nro	Opintojakson nimi	Ov	Arvosana	Muuta
010155000	Kompleksianalyysi ja integraalilaskennat	3	2	Korvaa opintojakson 010125000 Diskreetit mallit ja menetelmät, 3 ov. Aineopinnot 1997, 1998.
010200000	Numerinen analyysi	3	2	
010214000	Tilastotietotekniikan perusteet	2.5	2	
010506010	Tietotekniikan perusteet, osa 1	3	4	
010506020	Tietotekniikan perusteet, osa 2	3	3	
010533000	Tietotekniikka ja C-kieli	3	5	
010534000	Algoritmit ja suunnittelu	2	4	
010539000	Käyttöjärjestelmät	2	4	
010602000	Tietotekniikan perusteet 1	2	4	
010604000	Tietotekniikan harjoitukset	2	H	
010740000	Tietotekniikan perusteet	3	2	
010747000	Työväestien käyttö perusteet	1	H	
010751000	Makrotaloustieteen perusteet	3	2	
010758002	Ohjelmistotietotekniikka	3	2	
080810000	Digitaalitekniikka	2	2	
Yhteensä		37.5		

Figure 6: A part of a printable study plan.

The final study plan may be printed out and evaluated traditionally. Our application also allows electric evaluation where student submits the plan to the tutor from the application. This reduces the need of private guidance, which has both positive and negative affects.

3.2 Tutor tasks

To ease up the validation in many cases and to allow a real electric evaluation, also the tutors have their own interface to student's PSP. Compared to students' view, tutors can have some additional information available of student's plan. Each course and group has it's own information. This information may for example represent the state, if course is accepted into the

curriculum or not and how the course was initially placed into the group in the plan. Once accepted, the state will only be changed if student does some modifications for the course or the group. This means that once something is accepted, tutor does not have to go through all the plan again after a minor change, but only the committed change. Figure 7 represents a view for a part of plan to be evaluated. Different colors describe different states, including accepted and rejected as well as automatically and manually settled courses.

BASIC STUDIES 33.5 credits						
Based on 2000 - 2001 curriculum.						
ID	Course name	Vote	Credits	Note	Accept	Reject
010113010	Mathematics A		4	4		<input type="checkbox"/>
010113020	Mathematics B		3	4		<input type="checkbox"/>
010135000	Matrix calculus		5	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
010511000	Fundamentals of programming		4	3	<input checked="" type="checkbox"/>	<input type="checkbox"/>
010961000	Becoming an academical student	H	0.5		<input checked="" type="checkbox"/>	<input type="checkbox"/>
070210000	German 3		5	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
070240001	Finnische Industrie	H	1		<input checked="" type="checkbox"/>	<input type="checkbox"/>
070253000	Finland als Partner		3	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
070302000	Swedish exam	H	0		<input checked="" type="checkbox"/>	<input type="checkbox"/>
070801000	Finnish		3	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
080310010	Physics T, part 1		3	3	<input checked="" type="checkbox"/>	<input type="checkbox"/>
080310020	Physics T, part 2		5	3	<input checked="" type="checkbox"/>	<input type="checkbox"/>
080318000	Physics T, Laboratory works		4	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
080802000	Fundamentals of electronics B		5	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
1507A-96	Fundamentals of Computer science		5	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Accept group				32.5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figure 7: Tutor validating student's single group of studies.

3.3 Administrator tasks

The administrative interface is required for keeping the system up to date. New study structures are published every year, students graduate and tutors change. Some of the administrative tasks, such as maintaining structures and course information, can and most probably will be included for tutors' tasks as well.

The view for creating structure is presented in figure 8. Step by step a relational curriculum is built. The view includes a group with two subgroups. These groups include a couple of courses that may be set to be either obligatory or voluntary. They can also be alternative with each other.

Name	Credits	Remove	Mandatory/Optional
No extra information available. Add...			
Mandatory studies [03B1] (Laboratory of Datacommunications)	45		[Modify] [Remove]
Add course...			
Add alternativity between groups...			
Mandatory: 11 Optional: 0			
Name	Credits	Remove	Mandatory/Optional
No extra information available. Add...			
Basic Studies [03B11]	11		[Modify] [Remove]
010607000	Communications Software and Architecture	3 <input type="checkbox"/>	<input type="checkbox"/>
010608000	Communication Software Laboratory Works	3 <input type="checkbox"/>	<input type="checkbox"/>
010673000	Computer Networks and Data Communications	2 <input type="checkbox"/>	<input type="checkbox"/>
010674000	Basics of TCP/IP	3 <input type="checkbox"/>	<input type="checkbox"/>
Add course...			
Add alternativity between groups...			
Name	Credits	Remove	Mandatory/Optional
No extra information available. Add...			
Special Studies [03B12]	34		[Modify] [Remove]
010626000	Local Area Networks, Special Course [Ⓔ]	5 <input type="checkbox"/>	<input type="checkbox"/>
010666001	Broadband Communications [Ⓔ]	5 <input type="checkbox"/>	<input type="checkbox"/>
010680000	Seminar on Data Communications	2 <input type="checkbox"/>	<input type="checkbox"/>
010696000	Group Work Course on Communications Software [Ⓔ]	5 <input type="checkbox"/>	<input type="checkbox"/>
010715000	Information Technology Project	7 <input type="checkbox"/>	<input type="checkbox"/>
misc	Master's Thesis	20 <input type="checkbox"/>	<input type="checkbox"/>

Figure 8: Administrator's curriculum creation view.

4. Future work

Currently the system has been taken in use in the department of Information Technology at Lappeenranta University of Technology. We are planning to share the source code for interested instances. We are paying attention for the impact of the Bologna process [5] and the changes being caused by it. We are looking forward to have some feedback from the students. We have become aware that the needs of first year students and those close to their graduation vary from each other from this tool also. Combining these will be an interesting challenge for us.

References

- [1] eHopo. Homepage for OVI-portal. Available at <http://ovi.joensuu.fi/>. Referred 14.2.2005.
- [2] W3C. World Wide Web Consortium. Homepage available at <http://w3c.org/>. Referred 14.2.2005.
- [3] Deborah Hix and H Rex Hartson. Developing user interfaces: Ensuring usability through product and process, 1993.
- [4] Chris Shiflett. Foiling Cross-Site Attacks, October 2003. PHP Architect.
- [5] Ministry of Education. The Bologna Process. Available at: <http://www.minedu.fi/minedu/education/bolognaproces.html>. Referred 29.4.2005.