Ville Mikkola

Designing administrative features for social learning application

Helsinki Metropolia University of Applied Sciences
Bachelor of Engineering
Media Engineering
Thesis
14.5.2012

Metropolia

Abstract

Author	Ville Mikkola
Title	Designing administrative features for social learning application
Number of Pages	38 pages
Date	14 May 2012
Degree	Bachelor of Engineering
Degree Programme	Media Technology
Specialisation option	Digital Media
Instructors	Antti Keurulainen, CEO Ilkka Kylmäniemi, Lecturer

The purpose of this thesis was to design user interfaces of three different administrator features for social learning platform. The goal of this thesis was to design user interfaces that would offer high usability and functionality.

The Constantly growing amount of social and networking web services and the informal learning taking place in those services have changed traditional e-learning, as more formal learning style to be old-fashioned. Nowadays, many e-earning companies have started to focus more on informal learning and ways that it could be used in a corporate environment. The client company has started developing Soclet in 2009 and in 2010 the first version was ready with the idea of combining learning management systems with social features that provide informal style of learning.

The learning platform is mainly built with Ruby on Rails -framework together with HTML and JavaScript technologies. The platform's development process is managed with a variation of agile development methodologies called Scrum. Scrum is a iterative development process that is very flexible, because of the short development periods and constant prioritization of requirement tasks. Scrum was included in this thesis to keep the design work connected with the core process of project development.

Features chosen for this thesis were user account removal, user role management and the dashboard. All three features were designed starting with paper sketches and continuing with wireframe models to ensure that the final design was functional and possible to implement. The outcome was a highly usable user interface plan for each feature. User management was improved with easy way to disable user accounts and a new role management feature that makes the access control more flexible in the service. The new modular dashboard design opens new possibilities for developers to extend the administrator panel in the future.

Keywords	e-learning, Scrum, user interface design

Metropolia

Tiivistelmä

Tekijä Otsikko Sivumäärä Aika	Ville Mikkola Sosiaalisen oppimispalvelun ylläpito-ominaisuuksien suunnittelu 38 sivua 14.5.2012
Tutkinto	insinööri (AMK)
Koulutusohjelma	mediatekniikka
Suuntautumisvaihtoehto	digitaalinen media
Ohjaajat	toimitusjohtaja Antti Keurulainen lehtori Ilkka Kylmäniemi

Insinöörityön tarkoituksena oli suunnitella kolme eri käyttöliittymän osaa e-learning-alalla toimivan yrityksen kehittämään sosiaalisen oppimisympäristön ylläpito-näkymään. Työn tavoitteena oli suunnitella helppokäyttöiset ja toimivat käyttö-liittymät valituille toiminnallisuuksille.

Erilaisten sosiaalisiin toimintoihin keskittyvien palveluiden kasvava määrä on johtanut arkioppimisen yleistymiseen perinteisen kouluopetustyylin sijasta. Muutos on johtanut perinteisen verkko-oppimisen vanhentumiseen, minkä myötä alan yritykset ovat aloittaneet arkioppimisen tutkimisen yrityskäytössä.

Insinöörityön tilaajayritys aloitti oppimispalvelun kehittämisen vuonna 2009, ja ensimmäinen versio saatiin valmiiksi vuonna 2010. Sen ajatuksena oli yhdistää perinteisistä oppimistavoista tutut järjestelmät sosiaalisten toimintojen kanssa.

Palvelu on kehitetty pääasiassa käyttäen Ruby on Rails -ohjelmistokehystä ja hyödyntäen HTML- ja JavaScript-teknologioita. Palvelun kehittämisen aikana projektinhallintaan käytetään ketterien kehitysmenetelmien pohjalta luotua viitekehystä nimeltä Scrum. Se perustuu toistavaan ja lisäävään työskentelytapaan, jossa ohjelmistokehitystä tehdään lyhyissä jaksoissa. Se sisällytettiin osaksi insinöörityötä, jotta suunnittelu ja varsinainen ohjelmistokehitys pystytään yhdistämään helposti työn jälkeen.

Suunniteltaviksi toiminnoiksi valittiin käyttäjätunnuksien poisto, käyttäjäroolien hallinta ja hallintapaneeli. Kaikki toiminnot suunniteltiin aluksi luonnostelemalla paperille ja siirtymällä rautalankamallien avulla suoritettavaan käytettävyyden testaamiseen. Lopputuloksena saatiin yhtenäiset käyttöliittymät valituille toiminnallisuuksille, joiden toteuttaminen parantaa palvelun käyttäjienhallintaa, mahdollistaa monipuolisemman käyttäjäroolien hallinnan ja mahdollistaa ylläpito-näkymän eri osien jatkokehityksen suunniteltujen toimintojen pohjalta.

Avainsanat	e-learning, Scrum, käyttöliittymäsuunnittelu

Contents

1 Introduction

E-learning has been a viable option for medium and large-sized companies to run their continuous employee trainings cost efficiently across the world. It reduces expenses and allows companies to control the training process efficiently. [1.] However, today social aspects of learning have increased, thus resulting formal traditional e-learning to be old fashioned.

Soclet is a social learning platform developed by Bitville Oy to offer an e-learning 2.0 type of learning solution, which is an alternative for currently widely used LMS (Learning Management System) platforms. Soclet combines the traditional way of e-learning with social features that offer communality and non-formal learning style. Soclet allows companies to share their training materials in one place where their employees can study independently and also share their knowledge. [1; 2.]

The purpose of this thesis is to determine the currently most important features that should be implemented to Soclet, design a user interface for those functionalities, document how the functionalities would work and build wireframe models of each feature to illustrate the actual usage. The selected features will focus on administrative functionalities, as that side of Soclet has not been developed very far yet. This study should give a good base plan for Bitville Oy to develop the designed features. The thesis focuses on user-centered design and is influenced by agile development framework called Scrum that is currently used within Soclet's project team.

To improve the e-learning experience for modern day learning practices, the term e-learning 2.0 has been launched to meet the new demands for new kinds of platforms. These platforms should support traditional type of e-learning and also offer the possibility for learners to choose their own methods to study and allow them to socialize and collaborate together with other learners. Figure 1 illustrates how the e-learning 2.0 as a term can be placed between the traditional e-learning and social web services in the field of e-learning in general. [1; 2.]

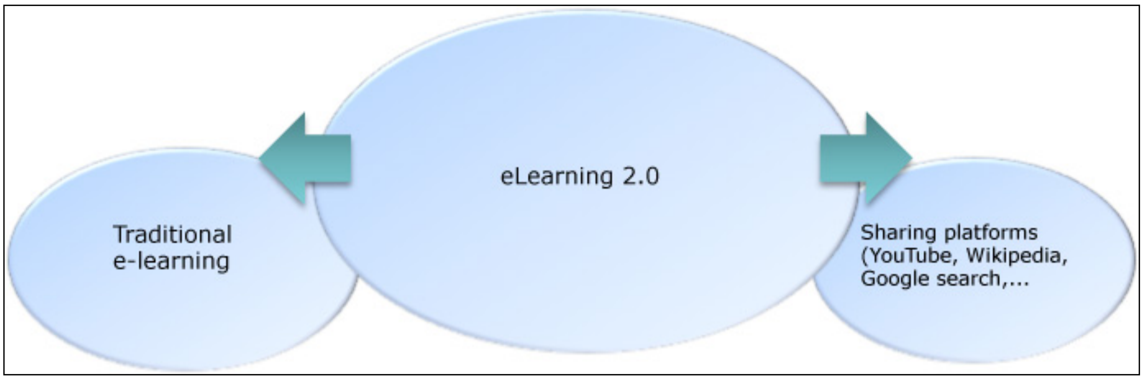


Figure 1. Visualization on how e-learning 2.0 has developed between formal and informal learning.

Nowadays a lot of learning happens through new channels such as YouTube, Facebook and other social environment where anyone can teach others new things, which are related to interesting topics. There is not necessarily a need for having a traditional teacher-learner type of relationship, because users have adapted their own ways of using the services to match their needs.

When Soclet was introduced, an administrator user interface was built to handle only minimum although mandatory requirements. As the service has been growing, there is now a high demand for more advanced administration user interface.

2 Soclet

The philosophy of Soclet is to be a service for a corporate use to improve existing competence of the company’s employees. The main reason to make the platform more social and user-centered is to allow users to share their knowledge easier, as in many companies there is usually a huge amount of knowledge among the employees, but it is not shared or used efficiently inside the organization. Soclet’s objective is to remove the obstacles that are preventing companies to increase their organizations maximum potential by offering a service to share tacit knowledge, expertise, interests and content across the company. [2; 3.]

2.1 History of Soclet

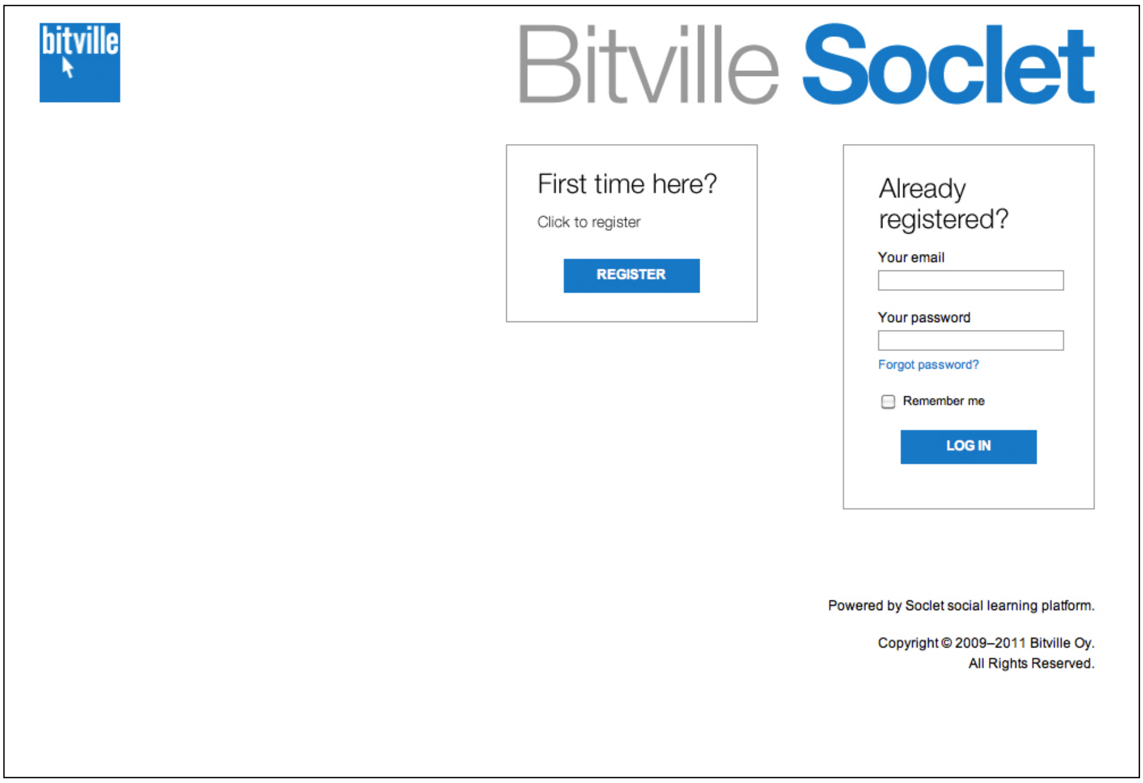
First steps in Soclet’s development were done in 2009 with Joomla!, which is an open source content management system. Joomla! offers a quick and easy way to setup a service that includes working login procedure and customizable components. The decisive reason to use Joomla! for the demo purposes was a wide range of different plugins that were used to build support for video and animation content which are crucial for an e-learning platform. Joomla! allowed Bitville Oy to build a working demo application fast and affordably in order to evaluate the concept idea of Soclet before starting the actual development phase. [4, p. 22.]

After the demo application was made and the concept idea was tested and refined, Soclet’s development was started in 2010 and at the time Ruby on Rails was chosen as the main technology to build it. Ruby on Rails is an open source web application framework for the Ruby programming language. After the first development phase with Ruby on Rails was concluded, Bitville Oy continued developing Soclet with a small team taking advantage of agile development framework called Scrum. [4, p. 22.]

2.2 Current state

Before starting to design a new user interface features for Soclet's administrator panel, it is good to make an overview of the current version. Current version works similarly to any common business intranet web-service. Only registered users are allowed to enter the system and registration can be done only within limited domains, which are defined in the system.

The latest version of Soclet's login page is shown in figure 2 with options to either login to the system or register as a new user. Registration and login are possible only from predefined domains that are set by system administrators in the administrator panel.

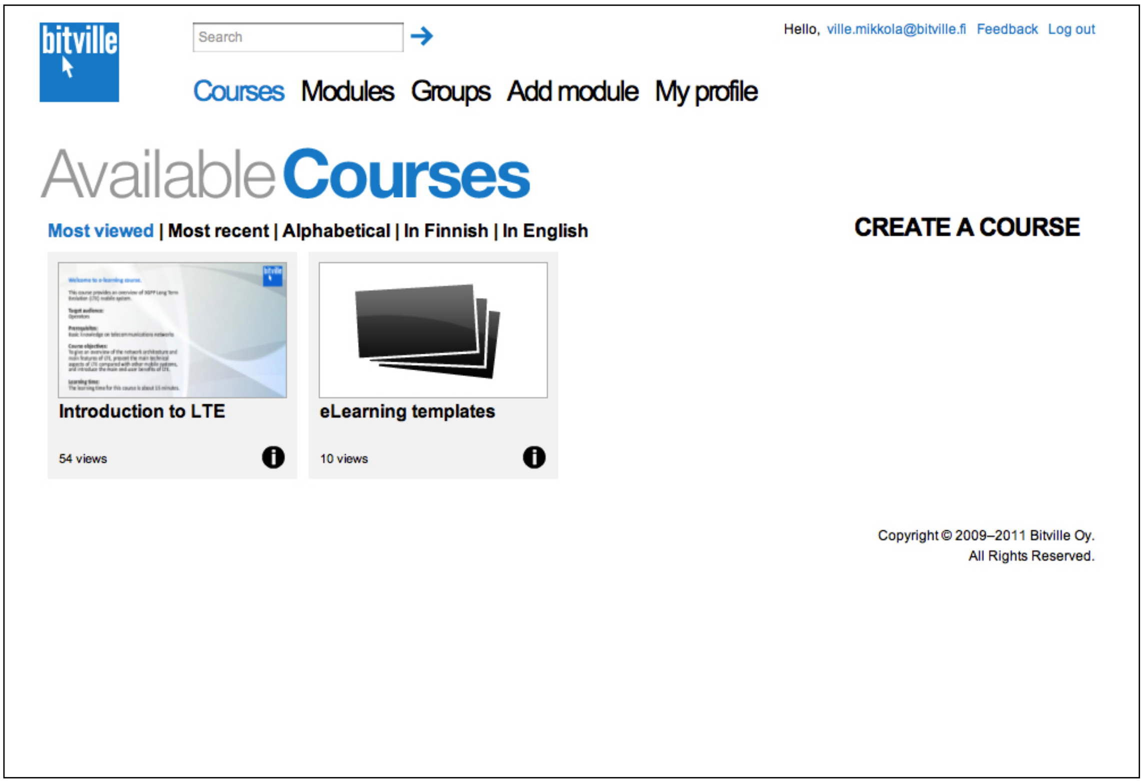


Powered by Soclet social learning platform.

Copyright © 2009–2011 Bitville Oy.
All Rights Reserved.

Figure 2. Login page with an option to choose registration or login.

The design of Soclet's user interface was highly influenced by widely used social web services such as YouTube and Facebook. The plan was to combine best practices from those services together with the course oriented structure used in learning management systems. That led the design to be a thumbnail-based list of content similar to YouTube that would be hierarchically separated to individual modules while still keeping the course concept supported. This makes the transition easier and more suitable for companies that are used to LMS solutions. Figure 3 illustrates the landing page of the current version of Soclet's main user interface after successful login.



bitville

Search

Hello, ville.mikkola@bitville.fi Feedback Log out

Courses Modules Groups Add module My profile

Available Courses

Most viewed | Most recent | Alphabetical | In Finnish | In English

CREATE A COURSE

Introduction to LTE

54 views

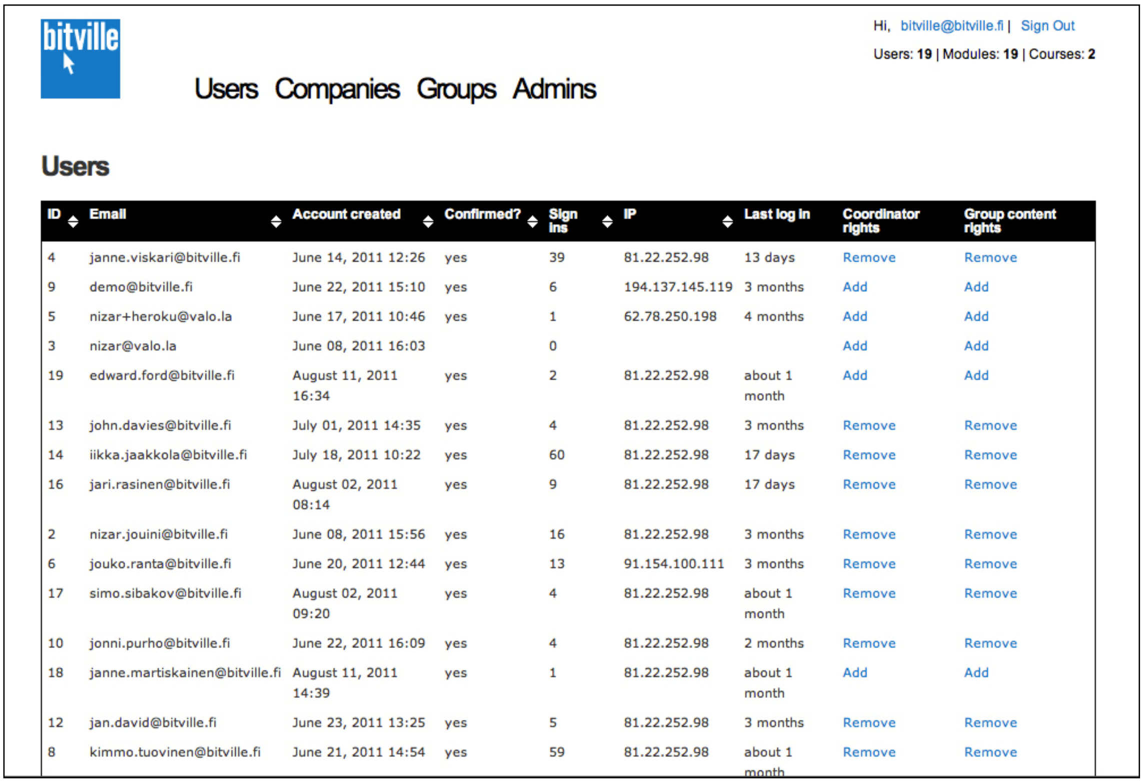
eLearning templates

10 views

Copyright © 2009–2011 Bitville Oy.
All Rights Reserved.

Figure 3. Soclet's main user interface after successful login.

In the current version, user roles are separated into two groups: basic users and administrator users. Basic users have only access to features that are available through the normal login procedure shown previously. Administrator users are separated from basic users, so that they can only access administrator panel, which uses separate login procedure. Figure 4 shows the current landing page for administrator panel, which by default shows the list of registered users in the Soclet instance.



Hi, bitville@bitville.fi | Sign Out

Users: **19** | Modules: **19** | Courses: **2**

Users

ID	Email	Account created	Confirmed?	Sign ins	IP	Last log in	Coordinator rights	Group content rights
4	janne.viskari@bitville.fi	June 14, 2011 12:26	yes	39	81.22.252.98	13 days	Remove	Remove
9	demo@bitville.fi	June 22, 2011 15:10	yes	6	194.137.145.119	3 months	Add	Add
5	nizar+heroku@valo.la	June 17, 2011 10:46	yes	1	62.78.250.198	4 months	Add	Add
3	nizar@valo.la	June 08, 2011 16:03		0			Add	Add
19	edward.ford@bitville.fi	August 11, 2011 16:34	yes	2	81.22.252.98	about 1 month	Add	Add
13	john.davies@bitville.fi	July 01, 2011 14:35	yes	4	81.22.252.98	3 months	Remove	Remove
14	iikka.jaakkola@bitville.fi	July 18, 2011 10:22	yes	60	81.22.252.98	17 days	Remove	Remove
16	jari.rasinen@bitville.fi	August 02, 2011 08:14	yes	9	81.22.252.98	17 days	Remove	Remove
2	nizar.jouini@bitville.fi	June 08, 2011 15:56	yes	16	81.22.252.98	3 months	Remove	Remove
6	jouko.ranta@bitville.fi	June 20, 2011 12:44	yes	13	91.154.100.111	3 months	Remove	Remove
17	simo.sibakov@bitville.fi	August 02, 2011 09:20	yes	4	81.22.252.98	about 1 month	Remove	Remove
10	jonni.purho@bitville.fi	June 22, 2011 16:09	yes	4	81.22.252.98	2 months	Remove	Remove
18	janne.martiskainen@bitville.fi	August 11, 2011 14:39	yes	1	81.22.252.98	about 1 month	Add	Add
12	jan.david@bitville.fi	June 23, 2011 13:25	yes	5	81.22.252.98	3 months	Remove	Remove
8	kimmo.tuovinen@bitville.fi	June 21, 2011 14:54	yes	59	81.22.252.98	about 1 month	Remove	Remove

Figure 4. Soclet's current Administrator Panel.

Administrator panel is used to handle all higher-level system maintenance tasks including user, domain and group management. Administrator panel's user interface is separated from the main user interface to emphasize the difference between the each interface and to prevent accidental login attempts to administrator panel. As the service grows, the administrator panel needs to be more versatile and include new features to keep up with the main user interface. First step of the design process is to define new features that are needed and other requirements that affect the administrator panel's design.

3 Agile development with Scrum

Agile development is a popular development methodology that Soclet's development team is currently using, thus this thesis will cover some basic information related to agile development. Agile development is a set of software development methods based on incremental and iterative process that allow developers to respond quickly to changes in the project and work as a self-organized team, thus enhancing the work efficiency. The basic workflow of agile development is to have short development periods called sprints and during those periods, development team works as a self-organized unit on the preselected tasks by the team and stakeholders. Splitting development to short sprints allows developers to focus fully on selected tasks without any interruption and it helps stakeholders to keep track of the project state better than in normal project management, as development results are reviewed after each iteration together with the development team. [5, p. 10-16; 6, p. 11, 136.]

After first agile methods were introduced, many variations were also created based on the original idea to support different use cases and environments that each organization uses. One of those variations is called Scrum, which is one of the most used agile development frameworks nowadays. Compared to agile development, Scrum is not really a set of strict methods and rules to work with, but instead a framework that gives tools and guidelines for development team to create their own rule set. Scrum introduces some new methods on top of the basic agile development idea and defines the hierarchy of the development team, also called Scrum-team that contains different roles for each project participant in order to keep the team organized. [6, p. 38-40.]

3.1 Terminology

Scrum is built around a few terms that describe participants in the project and their roles as well as the process itself and its phases. Even though Scrum is very agile and customizable for different projects, there are some key rules that should be followed in order to have a successful work environment. This thesis will not cover all the Scrum rules, but the most important aspects are explained to give a general overview of the process itself.

3.1.1 Scrum roles

In the Scrum, there are three different roles: Product owner, ScrumMaster and the Scrum-team. All of these roles have a specific purpose in the Scrum process that should be clear to all participants when starting a new project.

Product owner is a person who is responsible for keeping the project on right track towards wanted goal. Product owner is usually a person in the project team who has the best knowledge of the high-level objectives and has the vision of what the end product should be. Product owner can also be a person from the client's side, but in that case good communication with the development team and devotion to the project is required to keep the project successfully proceeding. [5, p. 125-127; 6, p. 127-128.]

ScrumMaster is a person who is in charge of the development team in terms of making sure that project is following Scrum methods. ScrumMaster's primary task is to act as a buffer or a shield for development team, removing possible impediments that the development team might face during their sprints. ScrumMaster also keeps track of the project progress and communicates with the Product owner about possible problem situations or changes to the development plan. In some cases ScrumMaster can be partially part of the development team if the team itself contains only few developers. After all, ScrumMaster's main assignment is to lead the development team to achieve their sprint goals and help in the development if needed. [5, p. 117-118.]

Scrum-team is the third fundamental role in Scrum, which consists of three to ten members depending on the project scale. Development team is usually gathered of members with various types of skills and abilities such as programmers, graphic designers, testers and other industry specific experts that are needed for the current project. Cross-functional members are required for the, as the team to be able to work independently as a self-organized group. [6, p. 107.]

3.1.2 Product backlog

Product backlog is an ordered list of requirements for the project from where the project team chooses new items to work on when new development iteration called sprint starts. Typically product backlog items are written down in story format to describe requirements or actions that end user should be able to perform in the developed product. Anyone in the project team including product owner, ScrumMaster and the development team can add items to the backlog, but only the product owner can arrange added items to form a prioritized list of tasks that will generate the best business value with minimum time effort and risk possibility. To help the Product owner to make the decisions on how important and fast each item is to implement, development team members give their own estimation on each item, which product owner can use as a part of the prioritization process. [5, p. 235-236; 6, p. 88; 7, p. 167-168, 173.]

After selecting wanted items from the product backlog to be developed for the next sprint, items are transferred to new list called sprint backlog. Sprint backlog is only meant for the development team to work as a more technical to-do-list with feature requirements that combined together form original product backlog items that were requested. [6, p. 88; 7, p. 169.]

3.1.3 User story

A user story is a description of a feature that the developed system should have or an operation that the end user should be able to perform when using the system. User stories are usually short including three key components that describe wanted feature: target user who will user the feature, action that is performed by the user and the business value of the feature to justify need for it. Proper formatting and writing style are an essential part of user story descriptions to keep the product backlog clean and understandable for stakeholders and other project participants who are not required to see the technical aspects of each story. In this thesis user stories are extended to cover each feature in more detail, but stories are still kept non-technical. [5, p. 238-241; 7, p. 4-8]

3.1.4 Sprint

Sprint is a time period for development specifically defined for each project, usually lasting from one to four weeks depending on how big the project is. Before each sprint starts, Product owner, ScrumMaster and the development team holds a Sprint planning meeting, where the whole group selects the most important tasks to work during the following sprint. Each day during the sprint, ScrumMaster and the development team have a short meeting called daily scrum. Daily scrum or sometimes referred daily standup is a status meeting to keep everyone updated on possible changes and the progression of the sprint.

In the end of each sprint, whole team comes together again to review past sprint. Review is divided into two different meetings: Sprint review meeting and Sprint retrospective. During a sprint review meeting, all the completed tasks are presented to stakeholders. Reviewing is continued after sprint review meeting with development team, ScrumMaster and Product owner in Sprint retrospective meeting. During retrospective meeting, team goes through past sprint in more detail trying to see what when well during the sprint and what could be improved for the next sprint. That way the Scrum process develops inside the team and future sprints become more efficient and agile. [5, p. 257-258; 6, p. 90-91.]

3.2 Scrum in this thesis

Some of the Scrum methods were taken into use for this study in order to connect the thesis with on going development of Soclet, but most of the Scrum methods cannot be used as the team size was reduced to a single person. This thesis mostly focuses on user stories as a part of scrum to help the actual development that will be done in the future for Soclet. User stories are also explained in more detail compared to regular Scrum, user stories are short descriptions of what user should be able to do, but as this thesis combines preplanned specifications with agile development, user stories are described as a fully working features so that development can be started right after design process is completed.

4 User story definitions

User story is a description of a functionality requirement that system should provide for the end user. The story describes in the everyday or business language, what the end user should be able to do or what he needs from the system in order to perform his job tasks. User story also defines the benefits of the feature to clarify what is the business value of that functionality. Usually the stakeholder or an other person outside of the development team writes the stories to get requirements that are matching the actual end user needs. Developers can also write user stories if they come up with good ideas while developing the system as developers might have wider knowledge on non-functional features that are affecting system performance, security or other technical aspect of the system. [7, p. 5-9.]

As this study is not fully following agile development workflow, some parts of the development process are changed to suit better for one-person team. Instead of collecting information from stakeholders to define user stories, I am writing user stories myself in co-operation with Bitville's project management persons. User stories in this thesis are also defined more comprehensively compared to user stories defined normally in agile development, so that this thesis can be used as documentation to build designed features.

User stories generally follow a pattern that contains three elements. First is described who needs the feature as a user, then the action is defined that the user should be able to perform and lastly the gained value of the feature is explained to assess its importance. [7, p. 6-7.]

4.1 Disabling user account

A definition of the user story could look like the following: As an administrative user, I want to be able to delete or disable user accounts in the service, so that I can deny access rights for those who are not allowed to use the Soclet or to clean up the service to have accurate count of active users and usage rates of the service.

The process of deleting or disabling a user account is divided into several distinct cases that need to be considered before designing the user interface and implementing the functionality. There can be many reasons for removing or disabling a user account from the service such as a person leaving the organization, a user behaving inappropriately in the service or when user account is allowed to access the service only during a specific time period. Also, Soclet’s pricing conditions can be based on the number of active users in the service, hence a functionality to disable users is required to calculated count of active users for corresponding pricing.

When users are disabled in the Soclet, it is important that the system keeps track of possible contents that disabled user could have uploaded to the system. That way the ownership of those contents can be moved from the disabled user to an administrator or other active user. After active use of the service, user might have also participated in numerous discussions where his or her posts could be an important part of the discussion, thus removing user’s posts could render the whole discussion thread fragmented.

In some cases, an individual person might own the copyrights for certain content files in the Soclet instead of the company that he or she is working for. When that person leaves the company or wants to remove his or her content from the Soclet, only the content files should be removed. Therefore, the content description, posted comments and the page structure are left untouched to ensure that all of the material that was linked to the removed content are kept available for later use.

If the user does not own copyrights to the content, it is important that the ownership is transferred to someone else. This is necessary incase the person who has the ownership decides to leave the company or his or her user account is disabled. To avoid situations where materials that were uploaded become unmanageable due to a loss of content ownership, administrative users must be able to move the ownership between the users. Soclet should also automatically require the administrative user to transfer the ownership when deleting a user who has ownership for content in the Soclet.

4.2 User role management

When defining the user story of role management in Soclet it could be done in the following way: As an administrative user, I want to grant different roles for users with specific access rights to Soclet's features and materials, so that I can distribute general management tasks to other administrative persons and maintain overall user control in the service.

For large web services that contain a lot of users, role management becomes an essential part of the service, as it has to be dynamic and scalable for administrative users to keep the service running smoothly. Usually two user roles are enough for small services, as management tasks are few, but in Soclet, roles must be seen as individual rights to enable more versatile and customizable access rights.

Administrators must be able to grant predefined user roles and also customize those roles to have dynamic access right management. It is important to have predefined roles, as during the registration process, default role needs to be assigned for each user and when multiple users need to be assigned to the same role, predefined roles can be used to get the same rights for everyone. Predefined roles also minimize possible errors that could occur if individual rights for each task would be added separately. Role customization is also important as the service is further developed and new features are introduced that require new user rights. In order to keep the role base in the Soclet clear, administrative users must be able to save new user roles to be used later on.

An important part of the user management is to restrict the creation of the high-level administrative roles to secure that only Soclet's development team is able to create new accounts that have rights to change system level settings. The basic principal is to allow users to create new roles with access rights to only those features that they can access themselves. This ensures that the high-level control stays within the hands of selected system administrators, thus enhancing the security of the service.

Administrative users must also be able to edit existing user roles in order to avoid creation of multiple similar user roles. Editing should be possible only if editing user has sufficient user rights compared to selected role. Soclet should also automatically prevent creation of new user roles if a user role with the same rights already exists.

4.3 Administrator panel dashboard

The design requirements for dashboard can be described in the following way: As an administrative user, I want to be able to see important information about the service from the landing page of Soclet's Administrator site, so that I can get a quick overview of the service without browsing through multiple different pages to find the same information.

Dashboard is the front page of the administrator panel. On dashboard, administrators can see an overview of the Soclet instance and perform general maintenance tasks. Data on dashboard are displayed uniform modules that can be swapped between similar modules to decide what data are wanted to show. Modular layout structure makes dashboard to be easily customizable and fast to update or add new module types.

The first version of the dashboard design should have a flexible grid layout with predefined set of modules that demonstrate different ways to use modular layout. Predefined modules should offer information from a variety of different data sources such as user and content statistics, online users, hosting information, backup status, error logging and messaging capabilities. All modules should be designed as uniform block-style elements with a few different sizes depending on module purpose.

5 User story designs

Designing user stories starts with high-level hierarchy planning that consists of layout design, analysis of different ways to implement selected user stories and going through possible common user interface elements that can be used for other upcoming features in the future.

5.1 User-centered Web Design

User-centered web design is a design philosophy and a process that highly emphasizes the importance of end users' needs, opinions and limitations at each phase of the design process. Thereby the designers are not making decision based only on their understanding of the needs but instead, they collect information from the end users and make final decisions based on all the data that is collected. User-centered web design term is closely related to web usability, which means creation of web sites that are easy to use from the first user experience to continuous routine usage. Focusing on the user needs during the design process helps designers to make decisions that will more likely please the majority of the end users, thus maximizing the amount of potential users to become active users who use the service daily because it is easy and enjoyable to use. Service with good user experience and gratified user base will also lead to easier growth of the service as the current users will promote the service for their friends, coworkers and other potential new users. [8, p. 2-3; 9, p. 12-13; 10, p. 80. 11, p. 4.]

5.2 Layout design

Before starting to sketch user interface ideas on paper, it is good to think about different layout models for each user story and how those models can be used on this particular user interface. When building user interfaces for different features that belong to the same service, it is important to keep the look and feel simple and let users focus on the content instead of complicated user interface around it. Good user experience comes from predictability and consistency meaning that maintaining the position of frequently used buttons and other interface elements keeps the layout clean and easily memorable. If layout varies a lot between each feature, users have to learn each one separately. [12, p. 265-267, 471-472; 13, p. 67.]

5.2.1 User behavior

User behavior is an important aspect of the design process. Knowing how users view the web page, where are they looking at and what are they thinking while they perform different operations in the service helps designers to understand the real usage of the service and that way focus on highlighting important areas of the interface resulting in clearer browsing. A common assumption is that every time when a new user comes to a new page, he or she reads each elements to get an idea what he or she can do and then makes the decision, but in real situations users tend to just glance at the page and pick up certain elements that have more visual impact for their eye. [9, p. 4-5; 14, p. 21-23.]

There are some important things to focus on when designing user interface to make it easier for users to browse. The first one is to create a consistent visual hierarchy on every page and clearly point out different interaction possibilities for each element to avoid hesitation in users behavior as shown in figure 5. The second way is to group layout elements that are linked together to form easily recognizable sections and at the same time minimize all the material that is not important. This helps users to find important things faster and avoid getting distracted by non-important elements. Lastly, it is also important to take advantage of commonly known conventions that users recognize immediately. [14, p. 31-34.]

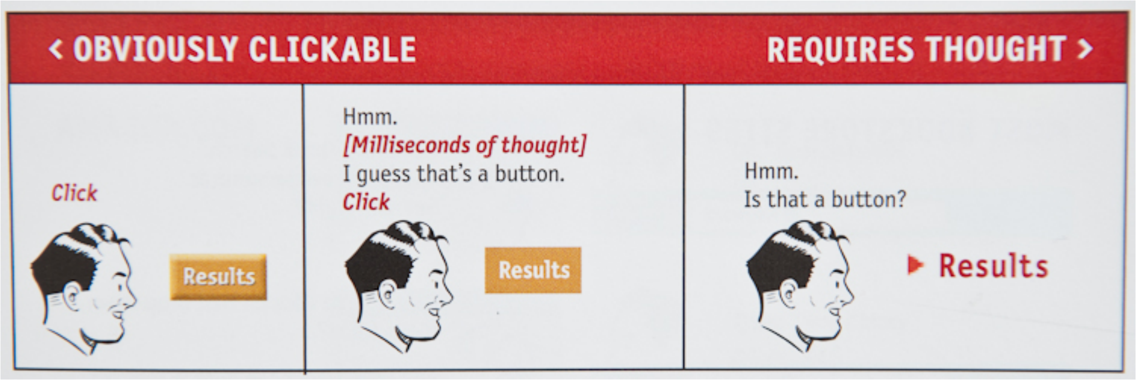
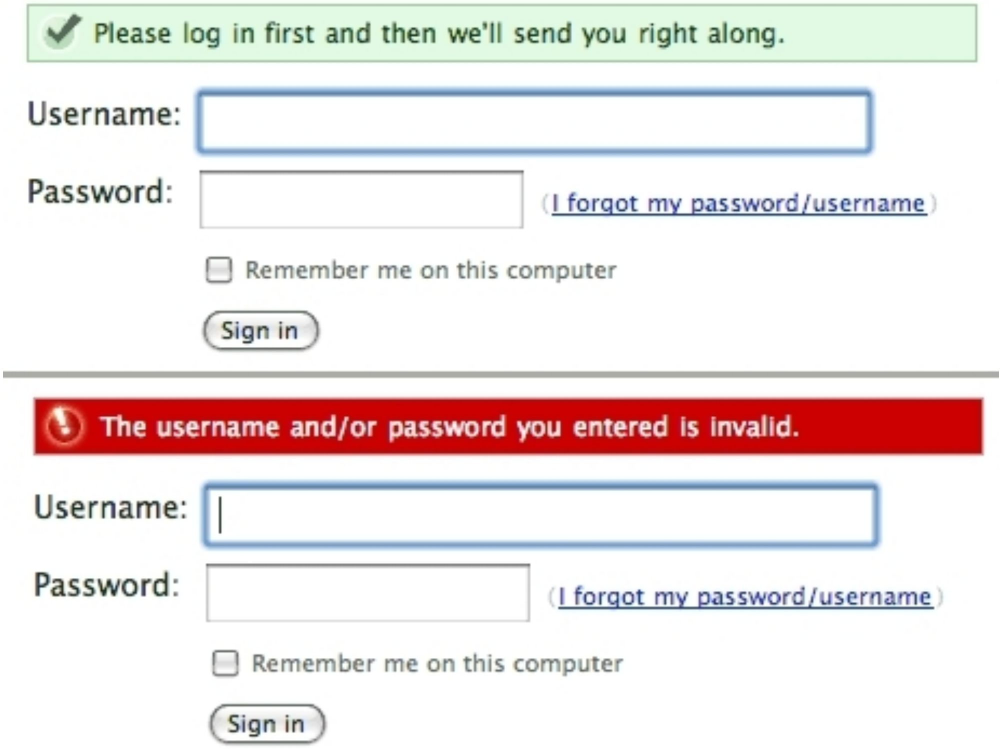


Figure 5. Changing the look of buttons can have a huge impact on how easily users recognize them as clickable buttons. [14, p. 15.]

5.2.2 Consistency and design conventions

Consistency is a crucial part of the usability of a user interface as with consistent elements, new users will learn the user interface easier and be able to quickly adapt to it. Page elements that are used on multiple pages for same functionality should always be located in the same place within the user interface to ensure consistency. Appearance and behavior of those elements should also remain the same throughout the whole user interface, to allow users to memorize where elements are and how they work, thus enhancing the user experience. In some cases it is hard to make complex user interface highly usable and at the same time appealing for users, but with consistent components a user interface can be build to be highly functional and at the same time aesthetically pleasing as the component looks and functions the same way on each page. Even though this thesis covers only three different features in Soclet, it is important to design layouts for those features to be reusable, as new features might contain similar elements. [12, p. 88-89; 13, p. 64, 67.]

An other way to enhance the usability of web applications is to use widely known design conventions that are proven to be understandable for majority of people and well functioning in different use cases. Design conventions include common ways to implement features such as sorting, filtering, pagination, status visualization and notifications for successful and failed operations. Different colors, shapes and other visualization elements that are taken from the real life situations, help users to understand indicated actions as the user can refer to those visualizations from earlier experience without being forced to learn new ways to process information. Figure 6 demonstrates an example of input form where system gives a feedback for user either with green or red colored box depending on whether the operation was successfully completed. [12, p. 483-487; 14, p. 34, 60-61.]



✓ Please log in first and then we'll send you right along.

Username:
Password:
(I forgot my password/username)
□ Remember me on this computer
Sign in

The username and/or password you entered is invalid.

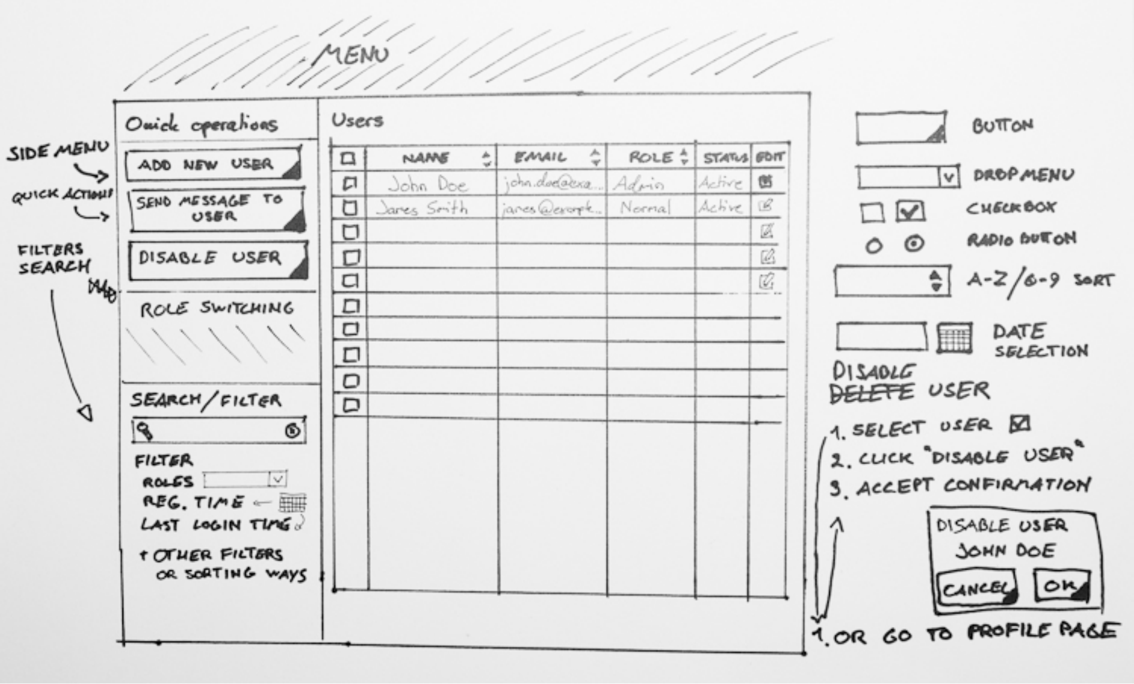
Username:
Password:
(I forgot my password/username)
□ Remember me on this computer
Sign in

Figure 6. Color-coding user interface elements helps user to quickly understand if performed operation was successful or if something went wrong. [http://ui-patterns.com/collections/input-feedback/entry/180.]

5.3 Sketching

Sketching is the first step to start building the user interface based on the ideas and functionalities that designers have in mind. The primary goal of sketching is to quickly draw different user interface ideas and evaluate them in order to understand and decide how the interface should be designed and developed. It is important to keep the sketching process simple and not focus too much on details, as the idea is to roughly estimate how different pieces fit together. Fast sketches help designers to eliminate non-working solutions before the final design process starts. [15; 16, p. 59.]

In this thesis project the first step was to start sketching user deletion and user role management –stories as both of those are linked to the user management page on the administrator panel. I wanted the user table to undergo as few changes as possible so that new features would mostly merge with the current view. As an addition to the original layout, I decided to add a side menu to the administrator panel as shown in figure 7. Side menu contains links and other components for the most important and frequently used operations in the administrative panel. This helps administrators to find core features easier and make the maintenance duties faster in general.



□	NAME	EMAIL	ROLE	STATUS	EDIT
☑	John Doe	john.doe@exa...	Admin	Active	✎
□	James Smith	james@exampk...	Normal	Active	✎

DISABLE ~~DELETE~~ USER

1. SELECT USER ☑
2. CLICK "DISABLE USER"
3. ACCEPT CONFIRMATION

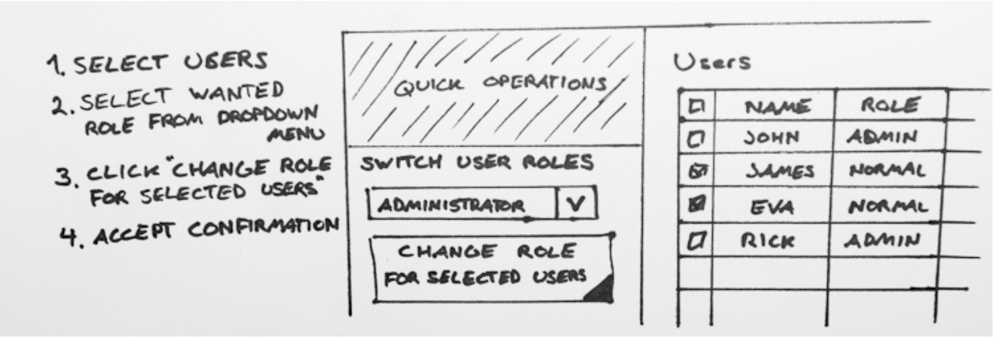
DISABLE USER JOHN DOE

1. OR GO TO PROFILE PAGE

Figure 7. Sketch of layout for new user management view.

The first link in the side menu is to add new users, as in some situations user accounts have to be created in advance. This can be done for demonstration purposes or if users are not allowed to create their own user accounts. The second quick link is for sending a private message to an other user in the Soclet instance. Private messaging features have not yet been implemented, but in the future communication between users will greatly improve the communality. The third quick link introduces a way to disable selected users, by selecting them from the user table with corresponding checkboxes and then clicking "disable user". Other content that I added to the user management view was options to search for users and filter given results, as growth of the service will soon make the administration of multiple users very hard without being able to narrow down shown results.

The first sketch of the user management view did not include an option to change user role, but the role was added to the table as one of the columns. Next phase was to come up with role switching component, as shown in figure 8. The original idea for role switching functionality was to show a small role selection box with radio buttons when current role was clicked, but that idea was rejected as it did not allow change of a role for multiple users at once. Also, the dropdown menu that was used in the final sketch was clearer and easier to use.



1. SELECT USERS
2. SELECT WANTED ROLE FROM DROPDOWN MENU
3. CLICK "CHANGE ROLE FOR SELECTED USERS"
4. ACCEPT CONFIRMATION

□	NAME	ROLE
□	JOHN	ADMIN
☑	JAMES	NORMAL
☒	EVA	NORMAL
□	RICK	ADMIN

Figure 8. Sketch of user role management elements on the side menu.

The next step was to design a profile page for users that would include same functionalities that user management view had and more versatile tools for role management. Flexibility of switching and adding roles was a priority thing for user management, so I decided to allocate a big area for role management in the profile view as shown in figure 9. The idea was to allow administrators to switch user's role quickly and preview assigned user rights instantly.

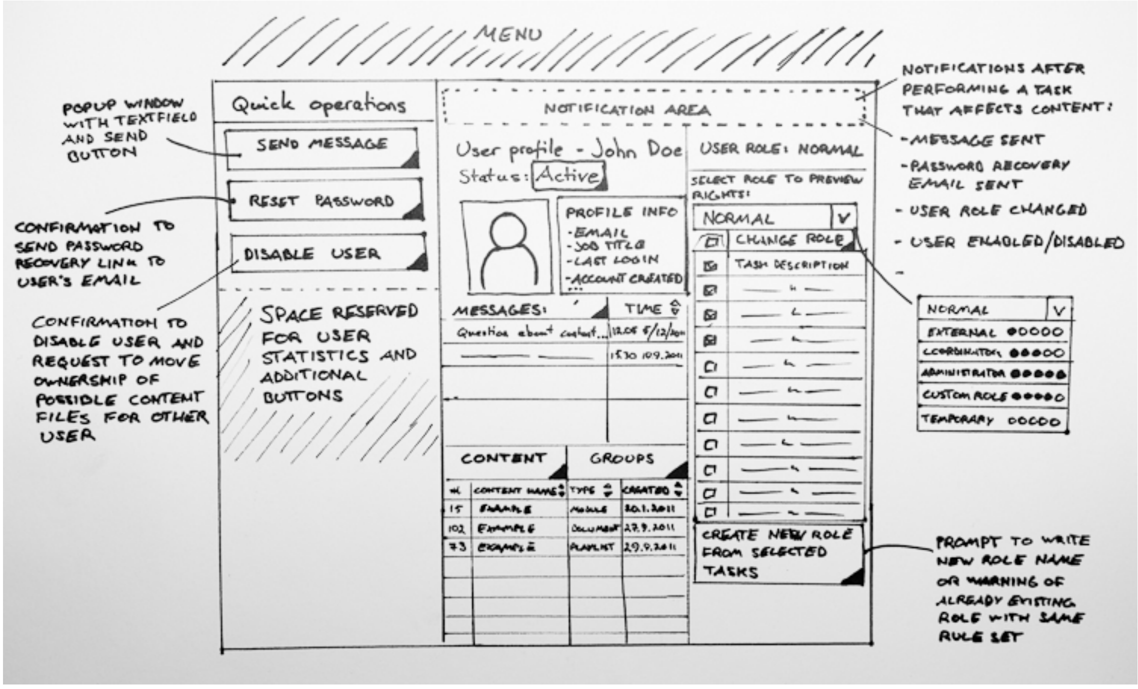
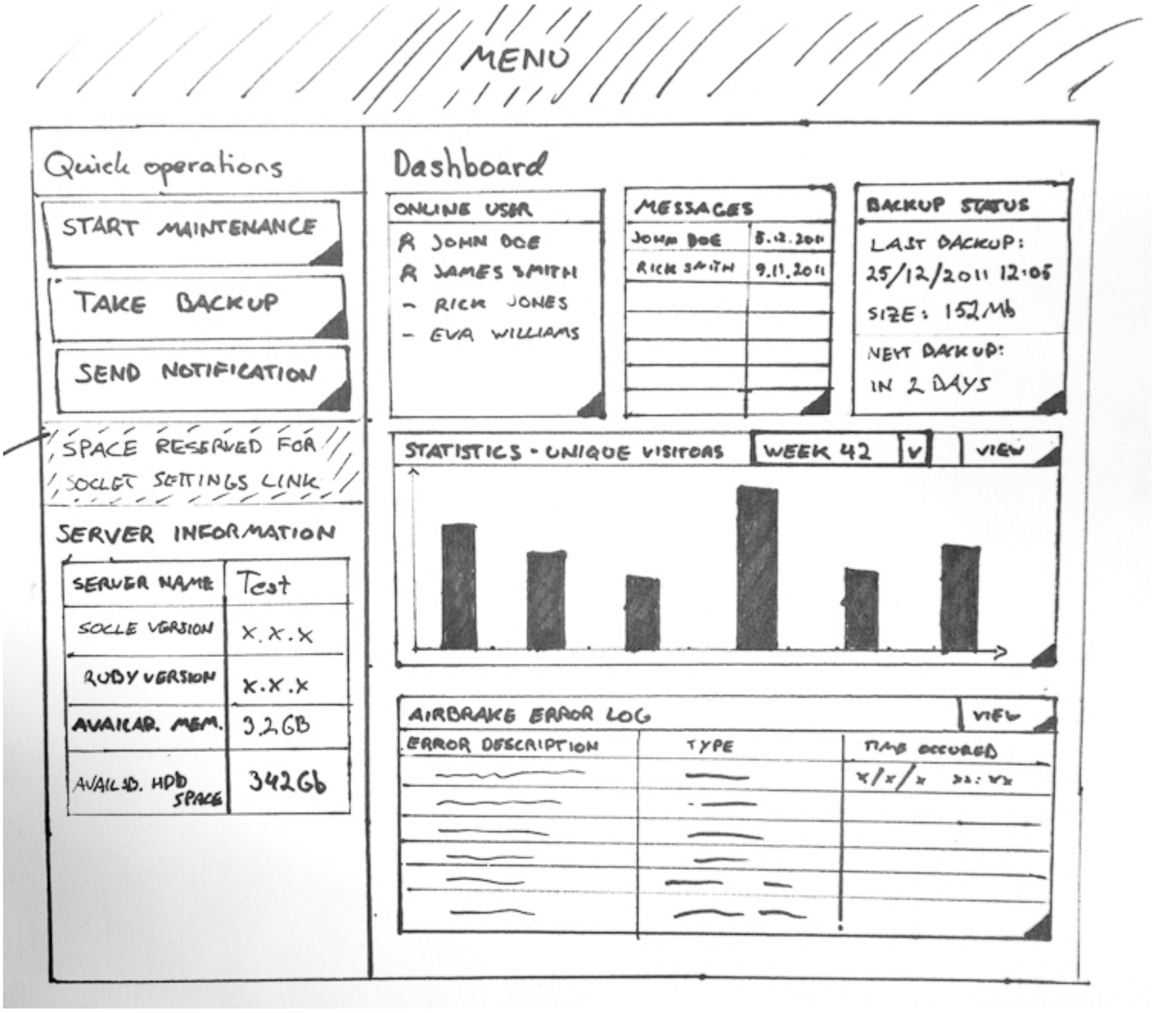


Figure 9. Sketch of user profile page including role management and preview for user rights.

The third user story is to create a dashboard layout, which would be functioning as a landing page for administrator panel. I started sketching with the same layout order I used for user management views and collected possible data sources that could be used for different modules. The final idea was to have two different sizes for modules that would build the modular grid layout as show in figure 10. Two different sizes were chosen as some of the modules would require a lot of space to be usable such as statistics and error logs. However, other modules such as backup information do not require large space to be usable. After sketching a mockup for each user story, I started building the wireframes with prototyping application called Axure RP.



JOHN DOE	5.12.2011
RICK SMITH	9.11.2011

SERVER NAME	Test
SOCLE VERSION	X.X.X
RUBY VERSION	X.X.X
AVAILAB. MEM.	3.2GB
AVAILAB. HDD SPACE	342Gb

ERROR DESCRIPTION	TYPE	TIME OCCURED
		x/x/x xx:xx

Figure 10. Final sketch of the dashboard including layout idea and module design.

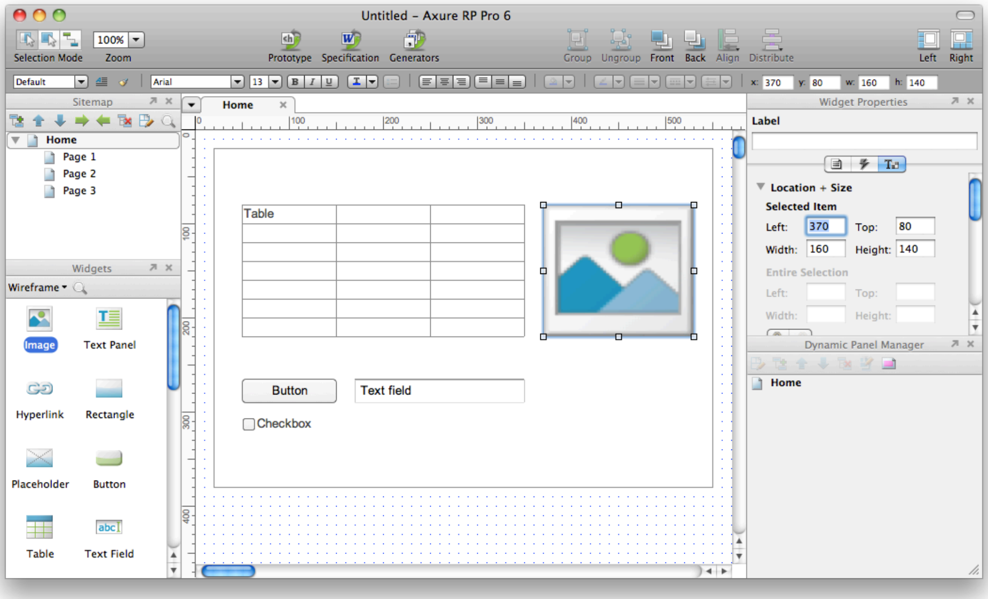
5.4 Wireframing

After sketching user interface ideas on paper, I started building wireframes to make those sketches come alive and to see the actual usability of the interface with fully functioning components that simulate the designed interaction. Wireframe versions should give the user an idea of what the final user experience would be by allowing the user to interact with the system and test it for possible usage problems and conflicting components. Wireframing can be done on paper just like sketching or with computer software that is specifically designed to build wireframes for websites and application user interfaces. Depending on the complexity of the user interface, a wireframing tool can be chosen based on the designer’s own preference. The criterium is whether those tools can provide needed interaction components and functionalities that reflect the actual end users experience of the system. [17, p. 269-271.]

For simple websites and other projects that contain only shallow page structure, simple data management and basic interactive components, paper or drawing software is usually the best option to use as it is fast and easy to edit later. For more complex user interfaces, there is a wide range of different user interface tools to choose from which allow users to create complicated and highly customizable user interfaces that can be built to interact almost like the final version the system. For this thesis, I used software called Axure RP, which is a wireframing and prototyping software made to create user interfaces for web and desktop applications.

5.4.1 Axure RP

Axure RP is a software to build wireframes and prototypes to design user interfaces for web and desktop application. Axure offers easy and quick tools to build user interfaces with premade widgets like textboxes, dropdown menus, buttons, checkboxes, table elements and other commonly used interface components that can be dragged and dropped to the user interface easily from the menu as shown in figure 7. All widgets have properties such as position, size, text formatting and widget specific settings that allow user to customize widgets. [18.]

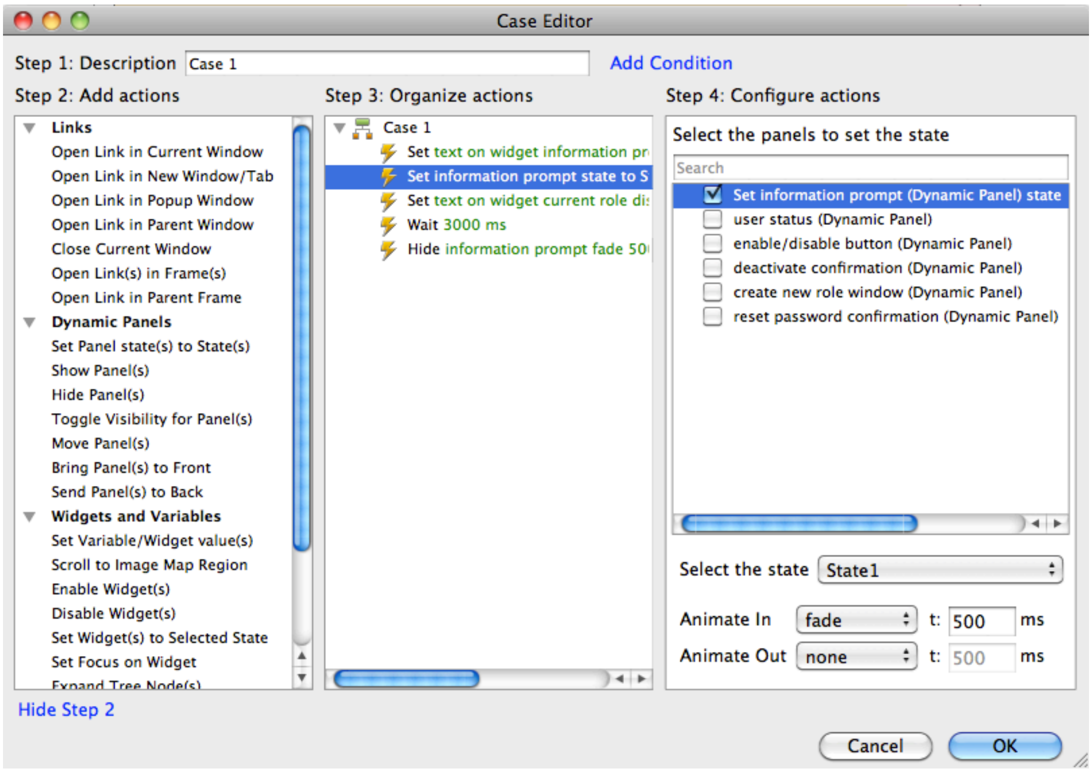


x: 370 y: 80 w: 160 h: 140

Left: 370 Top: 80
Width: 160 Height: 140

Figure 11: Axure RP user interface with common web application components.

For more advanced user interfaces, Axure offers comprehensive tools to build custom widgets and link those widgets to create interactive components that communicate with other components and pages to create pop up elements, transition effects, fully functioning forms and other html components that can be rendered as a html prototype for use case simulation. Figure 12 illustrates a case editor for widget where user can add his or her own rules to trigger different actions. [18.]



Step 1: Description Case 1

Step 3: Organize actions

- Case 1
  - Set text on widget information pr
  - Set information prompt state to S
  - Set text on widget current role di:
  - Wait 3000 ms
  - Hide information prompt fade 50(

Step 4: Configure actions

Select the panels to set the state

☑ Set information prompt (Dynamic Panel) state
☐ user status (Dynamic Panel)
☐ enable/disable button (Dynamic Panel)
☐ deactivate confirmation (Dynamic Panel)
☐ create new role window (Dynamic Panel)
☐ reset password confirmation (Dynamic Panel)

Select the state State1

Animate In fade t: 500 ms

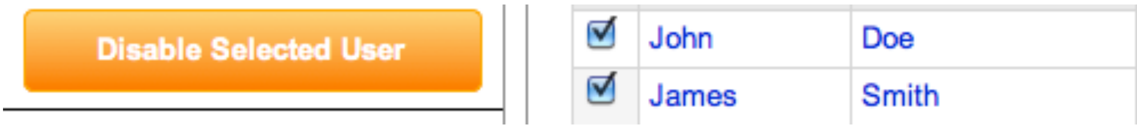
Animate Out none t: 500 ms

Figure 12. Widget case editor used to open notification prompt when button is clicked.

5.5 Story 1: Disabling user account

At first, the idea was to delete user accounts from the service, but after thinking about the process of removing a user and handling all the things that could be linked to the deleted user, the plan was changed to soft deletion. Disabling user accounts when access needs to be denied ensures that user’s content or other contributions are not lost.

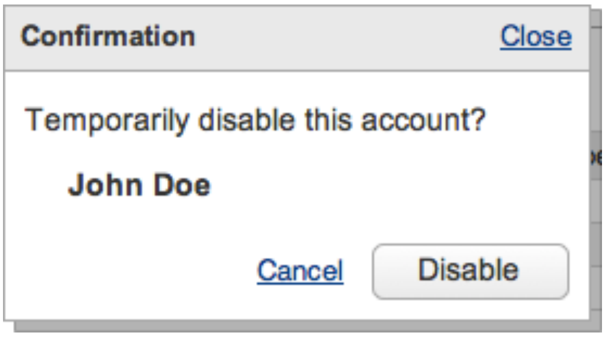
A functionality to disable a user was added into two different views. In user management view, one or more users can be disabled at the same time by marking users with checkboxes and clicking “disable selected users” as show in figure 13. In user profile page, the administrator can disable viewed user by clicking “disable user” similarly to the user management view. To avoid disabling or enabling users accidentally, the system asks for confirmation of the action after the button has been clicked as shown in figure 14.



Disable Selected User

☑	John	Doe
☑	James	Smith

Figure 13. User management view allows administrator to disable multiple user simultaneously.



Confirmation Close

Temporarily disable this account?

John Doe

Cancel Disable

Figure 14. Confirmation to disable selected user.

5.6 Story 2: User role management

User role management was originally planned to include only predefined roles and the possibility to edit rights for those roles. However, after a few meetings about the role management where scalability and flexibility of the role management was set as a priority requirement, I decided to design the role management to be more customizable for different use cases. Predefined roles were kept to have a role hierarchy to start with, which then can be extended with additional roles in the future.

The original idea with predefined roles was to have four different users levels in Soclet. Two of those roles are administrative users, System administrator for high-level maintenance with access rights to the administrator panel and Coordinator with access rights to the administrator panel, but restricted to only manage users, content and view statistics. Other two roles, Normal and External are regular users that do not have access rights to the administrator panel or any of its features. Normal user is the default user role in Soclet that is given to all users when they register. It has rights to only access the main user interface of the Soclet where he or she can watch and comment the existing content. External user role was created to allow administrator to create accounts for visitors, external employees, business partners and other users that will not use the Soclet in regular basis, but need a temporary account to review the service.

Filter component is placed in the side menu to make searching specific user from wide user base easier. Filter component is created to reserve space for future development of the actual component and it does not include all the necessary options, as it is only an example of possible filter options that can be included in the design. User table is also updated to include user status and user role. User profile can be viewed by clicking the user specific row or edit button in the last column.

Role management components are included in both user management and user profile views to allow administrator to make changes from multiple places. In user management view, only existing roles can be assigned for selected users by activating a checkbox for the wanted user and selecting a role from role selection dropdown menu as shown in figure 14 and also clicking "Change role for selected user". Figure 15 shows a notification prompt that is displayed for the user to inform if the role was successfully changed or if change was not possible.

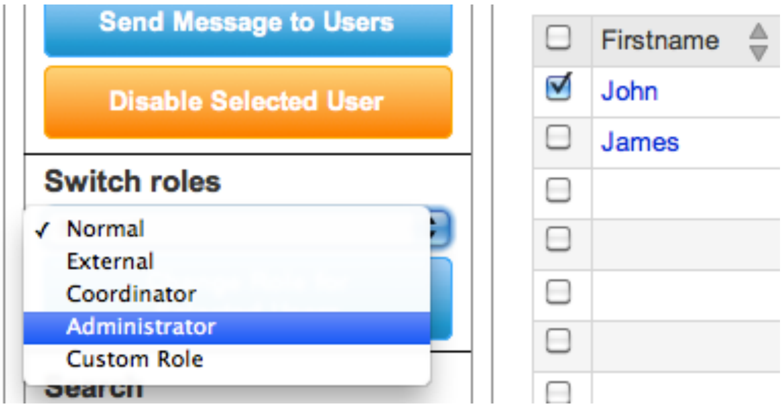
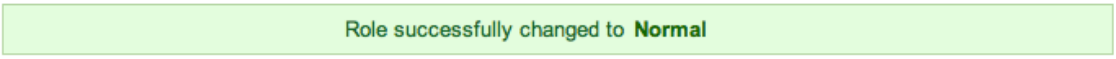


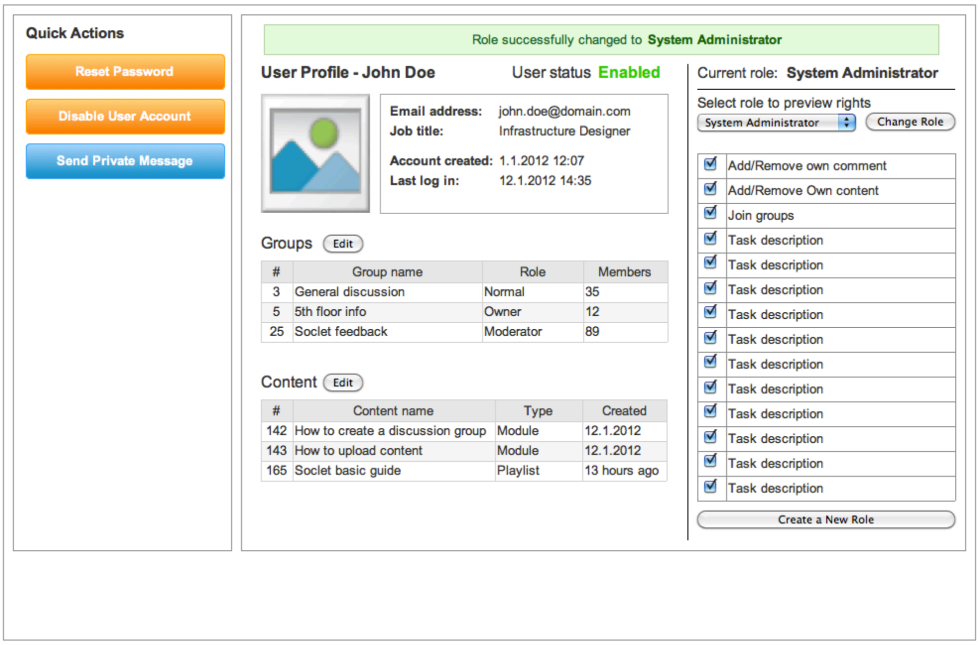
Figure 15. Selecting user role from the dropdown menu.



Role successfully changed to **Normal**

Figure 16. System notification shown for the user after successfully changing user role.

The main features to manage user roles are included in the user profile view. In profile view administrators can edit user rights of existing roles, create new roles with different access rights and preview any role to see the list of granted rights for that role. Figure 17 illustrates a view of profile page where user role has successfully changed to "System Administrator" with access rights to all available features in the service illustrated by the checkboxes.



Role successfully changed to **System Administrator**

User Profile - John Doe User status **Enabled**

Email address: john.doe@domain.com
Job title: Infrastructure Designer
Account created: 1.1.2012 12:07
Last log in: 12.1.2012 14:35

Groups

#	Group name	Role	Members
3	General discussion	Normal	35
5	5th floor info	Owner	12
25	Soclet feedback	Moderator	89

Content

#	Content name	Type	Created
142	How to create a discussion group	Module	12.1.2012
143	How to upload content	Module	12.1.2012
165	Soclet basic guide	Playlist	13 hours ago

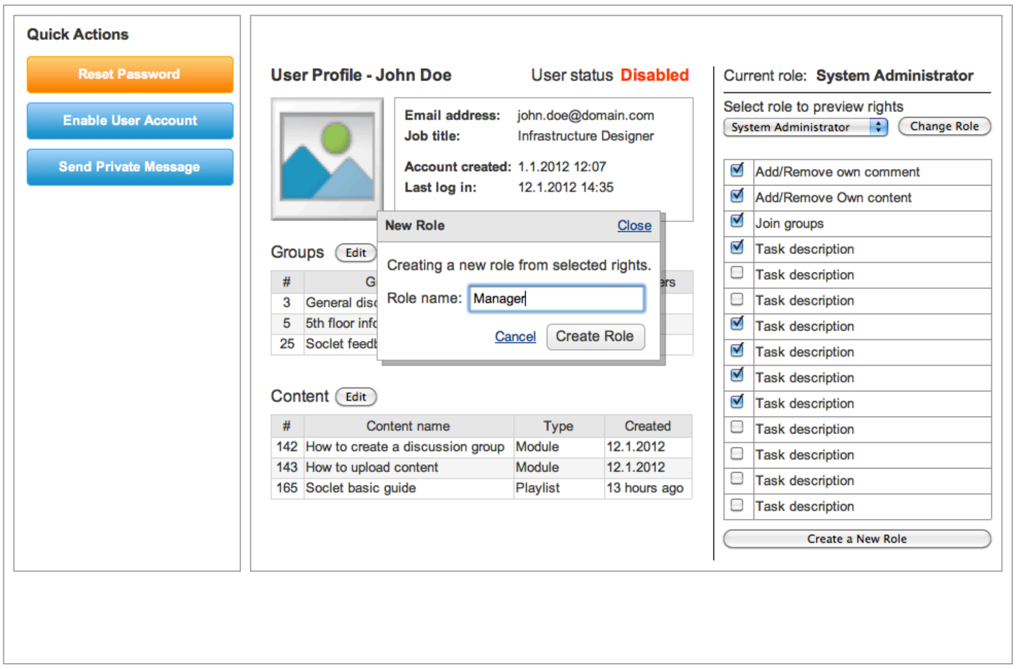
Current role: **System Administrator**

Select role to preview rights
System Administrator

- ☑ Add/Remove own comment
- ☑ Add/Remove Own content
- ☑ Join groups
- ☑ Task description
- ☑ Task description
- ☑ Task description
- ☑ Task description
- ☑ Task description
- ☑ Task description
- ☑ Task description
- ☑ Task description
- ☑ Task description
- ☑ Task description
- ☑ Task description

Figure 17. Final user interface for user profile page where user roles can be modified.

A functionality to create a new role is included in the same component that is used to change the user role. To create a new role, the administrator selects desired access rights from the list and clicks "Create a New Role". System opens a form for administrator to write the role name and accept as shown in figure 18. If the service already has a role with identical access rights, a confirmation form is shown to avoid duplicate user roles with different names.



User Profile - John Doe

User status **Disabled**

Email address: john.doe@domain.com
Job title: Infrastructure Designer

Account created: 1.1.2012 12:07
Last log in: 12.1.2012 14:35

New Role

Creating a new role from selected rights.

Role name: Manager

Groups

#	G	rs
3	General disc	
5	5th floor info	
25	Soclet feedb	

Content

#	Content name	Type	Created
142	How to create a discussion group	Module	12.1.2012
143	How to upload content	Module	12.1.2012
165	Soclet basic guide	Playlist	13 hours ago

Current role: **System Administrator**

Select role to preview rights
System Administrator

☑ Add/Remove own comment
☑ Add/Remove Own content
☑ Join groups
☑ Task description
□ Task description
□ Task description
☑ Task description
☑ Task description
☑ Task description
☑ Task description
□ Task description
□ Task description
□ Task description
□ Task description

Figure 18. System asks for name when creating a new user role.

Similarly to user management view, a profile view includes elements that reserve space for new features and illustrate possible development direction for later use. Side menu includes buttons for password reset and private messaging, which both will be useful functionalities in the future. The content area also includes new features ideas that could make administrative tasks easier to perform. Displaying content, groups and comments on the profile page that are linked to the user keeps data well organized and builds connections and reputation between users, which is a highly wanted outcome for social web sites.

5.7 Story 3: Administrator panel dashboard

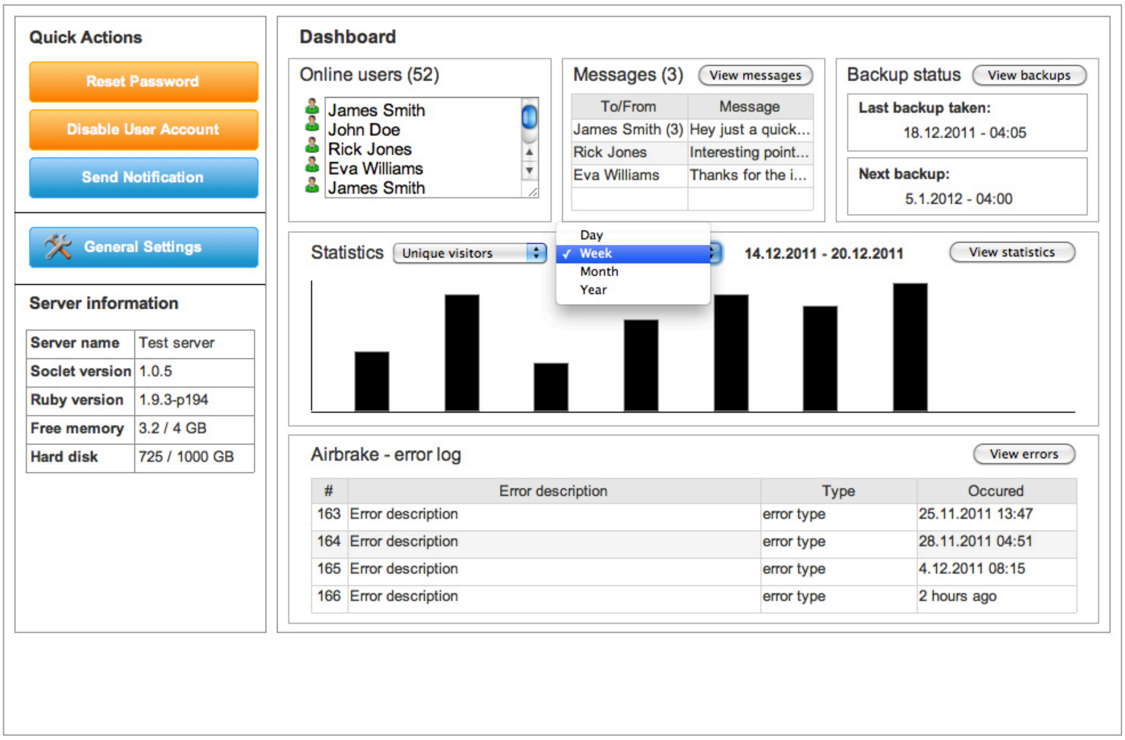
Dashboard is a completely new feature for Soclet to act as a landing page for the administrator panel. Dashboard combines different modules to one single view offering a quick and easy way for administrators to see an overview of the service and perform important daily maintenance tasks. Building layout out of uniformly shaped boxes allows the content to be swapped easily without changes to the structure. Box modules are reusable and make the page look organized and easily understandable. [19, p. 5.]

Dashboard user story was defined to only design the layout of the page and not include data module design, but to get a clear idea of how elements could be placed, I had to go through possible source options and base the design decisions on those modules. Example-modules were chosen to represent the variety of different information that administrators need to see in order to get get the overall understanding of the service.

Statistics are one of the most important information to be shown on dashboard especially for stakeholders and other people who are only interested about the business value of service like Soclet. Good statistics offer reliable data of general usage of the service and specific information related to individuals and their behavior in the system. Statistics also help developers to see which parts of the service are functioning properly and which areas are causing problems and require reconsideration. [10, p. 327.]

Currently Soclet uses an error management application called AirBrake to collect all the errors that occur in the system, it is very important that Soclet is running reliably and when an error occurs, developers will notice it as soon as possible. AirBrake offers a diverse API for developers to build their own error system, which makes it a good source for error logging module. A variety of different server information, such as backup status and network traffic can also be used for dashboard modules.

Dashboard layout is split in 5 different modules. Top area of the dashboard is for small modules that show limited amount of content. Small modules are useful to display backup status info, online users, daily statistics or private messages. For larger data sets, the layout contains two big modules to display graphs, big tables for error reports or bigger messaging module, if administrators are used to communicate regularly with users through the Soclet. Figure 19 shows an example of different modules combined to create a comprehensive overview of the service.



Server information

Server name	Test server
Soclet version	1.0.5
Ruby version	1.9.3-p194
Free memory	3.2 / 4 GB
Hard disk	725 / 1000 GB

Online users (52)

James Smith
John Doe
Rick Jones
Eva Williams
James Smith

Messages (3)

To/From	Message
James Smith (3)	Hey just a quick...
Rick Jones	Interesting point...
Eva Williams	Thanks for the i...

Backup status

Last backup taken:
18.12.2011 - 04:05

Next backup:
5.1.2012 - 04:00

Statistics Unique visitors Week **14.12.2011 - 20.12.2011**

Airbrake - error log

#	Error description	Type	Occured
163	Error description	error type	25.11.2011 13:47
164	Error description	error type	28.11.2011 04:51
165	Error description	error type	4.12.2011 08:15
166	Error description	error type	2 hours ago

Figure 19. Final version of the dashboard layout with modules in place.

6 Future development

Connecting more features with each other and adding social functionalities to all administrative features can make the administrative user interface a more dynamic tool to manage users and content in the Soclet. In the future, role management should go towards roles that are inheriting access rights from parent roles to keep the amount of roles as low as possible.

Currently administrative functionalities are focused on user management, thereby the next features should focus on content management and social aspects around it. Developing only one area of the service at a time makes prediction of upcoming problems and new ideas difficult, as already built features might have to be changed in order to connect them with new features that were not done in the first place. After content management is built to the same level with user management, developers can start implementing social features that combine users and content together strengthening the communality of Soclet. Large community with active users allows developers to build new social features such as reputation systems to emphasize active users.. When communality starts growing in the service, reputation and other social factors can be made an essential part of core features. [20, p. 172-173.]

Statistics are one of the most important things for developers to focus on. Saving statistics is important, but even more important is to build tools that can display statistics and export them for business decision and marketing. An easy way to understand and get benefit of collected statistics is to use analysis methods such as Funnel analysis, which follows users through their lifecycle in the service from a new user to an experience user that has used the service for a long time [21, p. 164.]. Understanding statistics helps the whole service to grow, as the development team can quickly see how the system is performing on different areas and make development decisions based on that data. Dashboard modules are an excellent place to display statistics, as different modules can be used depending on the style of the wanted statistics.

7 Conclusion

Nowadays there is a growing number of web services that are utilizing social and networking features. There is also a growing number of people accustomed to share their interests and be a part of different communities. Many popular web services, which rely on informal learning have also proven to be efficient learning environments. E-learning industry has noticed this change and many companies have started analyzing the possibilities that informal learning.

Bitville Oy started experimenting with informal learning in 2009 and in 2010 the first version of social learning platform called Soclet was released. Soclet was meant to be a combination of traditional learning management system with social features that offer communality and informal learning style. Soclet's development continued in 2011 by a small team that created a basic administration panel with simple functions to manage users and general settings.

The purpose of this thesis was to further develop the existing administrator panel by designing more features mainly focused on the user management. The first plan was to redesign the whole administrative user interface with focus on user management. However, this idea had to be rejected as it was too massive to be carried out in this project. Finally, three different administrator features were chosen for this study. The goal was to first define the selected features and then to design a user interface for each of those features.

As a result of this thesis, selected features were defined as user stories for Scrum compatibility. Then these features were designed based on the user story definitions by first creating a paper sketch of each feature and finally building wireframe versions to test the actual usage of the designed user interface. All designed features and user interfaces are relatively easy to implement and the new layout design offers logical places for old and new components. The design plan increases the usability of the user management view by allowing administrators to disable users and to manage user roles with precise task control. The dashboard layout offers flexible and customizable user interface that can be further extended with different module designs. All user interface components were designed to be extendable for further development, as the development work still continues.

References

1 Keurulainen Antti. CEO, Bitville Oy, Espoo. Conversation. 15.2.2012.

2 Soclet - Tool for corporate competence development. 2011. Project document. Bitville Oy.

3 Viskari Janne. Director, operations, Bitville Oy, Espoo. Conversation. 1.8.2011.

4 Tuovinen Kimmo. 2011. E-learning-järjestelmän kurssikonseptin kehittäminen. Thesis. Metropolia Ammattikorkeakoulu.

5 Cohn Mike. 2010. Succeeding with Agile : software development using Scrum. Upper Saddle River, NJ: Addison-Wesley.

6 Settler George. 2009. The truth about Agile software development with Scrum : the facts you should know. Milton Keynes, England: Lightning Source.

7 Cohn Mike. 2004. User stories applied : for Agile software development. Boston, MA: Addison-Wesley.

8 Lazar Jonathan. 2006. Web usability: a user-centered design approach. 1st ed. Boston, MA: Pearson Addison Wesley.

9 Rubin Jeff, Chisnell Dana. 2008. Handbook of usability testing: how to plan, design and conduct effective tests. 2nd ed. Indianapolis, IN: Wiley.

10 Bell Gavin. 2009. Building social web applications. Sebastopol, CA: O´Reilly Media, Inc.

11 Tullis Tom, Albert Bill. 2008. Measuring the user experience. Amsterdam: Elsevier/Morgan Kaufmann.

12 Shneiderman Ben, Plaisant Catherine. 2010. Designing the user interface: strategies for effective human-computer interaction. 5th ed. Upper Saddle River: Addison Wesley.

13 Lawrence Dave, Tavakol Soheyla. 2007. Balanced website design: optimising aesthetics, usability and purpose. London, England: Springer.

14 Krug Steve. 2006. Don’t make me think. 2nd ed. Indianapolis, IN: New Riders.

15 Rohde Mike. 2011. Sketching: the Visual Thinking Power Tool. Web document. <http://www.alistapart.com/articles/sketching-the-visual-thinking-power-tool/> Updated 25.1.2011. Read 15.12.2011.

16 Jovanovic Janko. 2011. Professional Web Design. Indianapolis, IN: Wiley.

17 Brown Daniel. 2007. Communicating design: developing web site documentation for design and planning. Berkeley, CA: Peachpit Press.

18 Axure RP. 2011. Web document. Axure Software Solutions. <http://www.axure.com/features>. Updated 18.4.2011. Read 15.12.2011.

19 Curtis Nathan. 2010. Modular Web Design. Berkeley, CA: New Riders.

20 Farmer F. Randall, Glass Bryce. 2010. Building Web Reputation Systems. Sebastopol, CA: O'Reilly Media, Inc.

21 Porter Joshua. 2008. Designing for the social web. Berkeley, CA: New Riders.