Last Planner

TDT4290 Customer Driven Project

Group 2

by

Sibte-Haider Syed
Atiyeh Seifvand
Kishore Kosuri
Eirik D. Haukedal
Aleksander L. Waage
Tobias Laupsa Nilsen
Christian Skar
Ole Kristian Ekseth

NTNU
Norwegian University of
Science and Technology

Contents

List of Figures

Part I

PlanningRequirements

1 Project Directive

This section discusses about the administrative plans of the project. This includes the results of the planning phase: project mandate, project plans and various templates, tools and routines involved with the project.

1.0.1 Chapter Overview

The following sections can be found in this chapter:

1. Project mandate : This presents details about the our customer and project.
2. Project organization :This states about our team organization and distribution of tasks and responsibilities.
3. Project plan: This part discusses on overall plan for the project, The choices we made for development and various pahses involved in the development process.
4. Templates and standards : This section describes about the various templates used while working with the project.

1.1 Project Purpose

The purpose of this project is to develop and deliver the Last Planner (planning the work at construction sites) application for the Client "Iterate and Belief". This report serves as an overview of development process involved with the project and guides the user to use the application.

1.2 Project Mandate

1.2.1 Project name

The name of the project is ”Last Planner”.

1.2.2 Project sponsor

The project sponsor is Iterate AS and Belief AS, one of Norway’s leading consultancy companies.

1.2.3 Stakeholders

Representatives from Iterate AS and Belief AS

1. Simen Fure Jørgensen
2. Lars Gullaksen

Group 2 in the course TDT4290 Customer-driven project

1. Sibte Haider Syed
2. Atiyeh Seifvand
3. Kishore Kosuri
4. Eirik D. Haukedal
5. Aleksander L. Waage
6. Tobias Laupsa Nilsen
7. Christian Skar
8. Ole Kristian Ekseth

Representatives from The Department of Computer and Information Science/Norwegian University of Science and Technology

1. Gustav Aagesen
2. Ole Andreas Alsos

1.2.4 Client: Iterate

Iterate is an international firm which provides software solutions for various businesses and organizations [?]. The main focus of iterate company is to find the best possible software solution fitting to needs of the customers. This is mainly achieved by becoming acquainted with the domain in which company is working on. The Lean construction is one such project in that direction.

1.2.5 Project Background

The consultancies Iterate and Belief are working together for developing software products for building industry. The building projects are getting benefited by following a method known as Last Planner(i.e. This is a framework for planning the building activities.)[?]. Presently, the Last Planner method is not fully developed. It is still in experimental stage.Iterate and Belief are exploring the possibilities to apply it to practical scenarios of construction thus increasing the productivity and saving lot of needless expenditures on waiting for activities to be done in the construction site.

./Images/PlanningProcess.jpg

Figure 1: Planning process of Last Planner

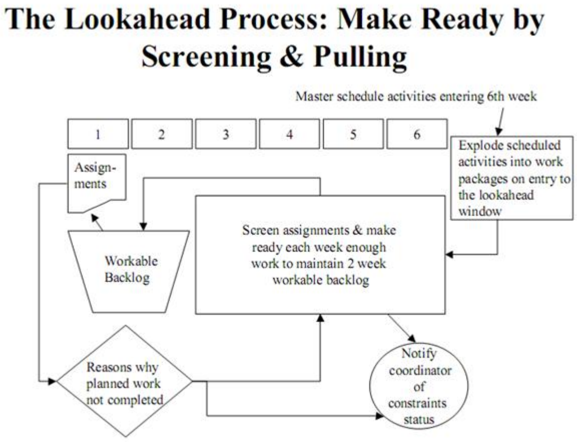

The Lookahead Process: Make Ready by Screening & Pulling
Master schedule activities entering 6th week
1
2
3
4
5
6
Assign-ments
Explode scheduled activities into work packages on entry to the lookahead window
Screen assignments & make ready each week enough work to maintain 2 week workable backlog
Workable Backlog
Reasons why planned work not completed
Notify coordinator of constraints status

Figure 2: Look ahead process involved in Last Planner

1.2.5.1 Last Planner The Last Planner system was developed by Ballard[?] . The framework was initially developed for improving the quality management and productivity in construction industry. The processes involved in the construction process are to schedule tasks, make ready plans and working on the backlogs and finally weekly work plans and its execution. The following figure 1 illustrates the processes involved in the planning system.

Last Planner research focuses on improving the quality of assignments in weekly work plans (Koch Refinery Mid-Plants Project, 1993-45). The research process of last planner continued by adding a look ahead process to shape and control work flow (PARC, 19956 ; DMOS-6, 19967). This was extended further from construction to design (Nokia 8 and Hewlett-Packard9 , 1996). The look ahead process is way of assimilating the tasks for the next period for instance, 3-12 weeks. The weeks are decided upon the project characteristics and requirements such as materials, labor, and equipment etc. The figure 2 illustrates the look ahead process showing work flow from right to left. The activities for 6 weeks are looked ahead of its execution. Then they move into workable backlog, which means that all the constraints have been fulfilled and ready for execution. In other words, The Last planner pull system includes what ŞcanŤ we do and what we ŞshouldŤ do in the planning process such that it ŞWillŤ produce the desired results. The figure 3 illustrates the pull mechanism involved in the last planner.

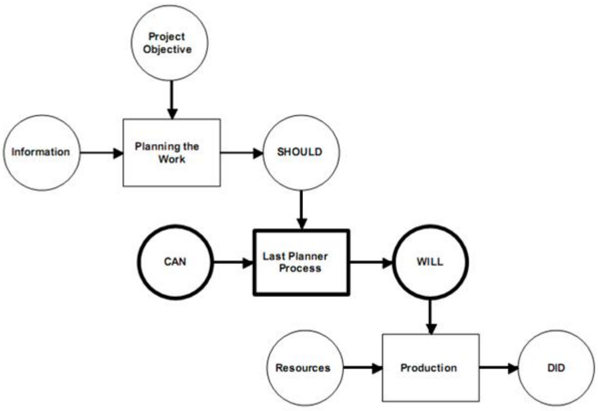
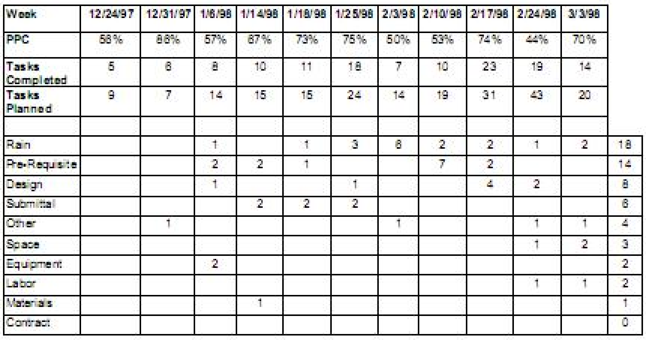


Figure 3: Pull process of Last Planner



Week	12/24/97	12/31/97	1/6/98	1/14/98	1/18/98	1/25/98	2/3/98	2/10/98	2/17/98	2/24/98	3/3/98	
PPC	56%	86%	57%	67%	73%	75%	50%	53%	74%	44%	70%	
Tasks Completed	5	6	8	10	11	18	7	10	23	19	14	
Tasks Planned	9	7	14	15	15	24	14	19	31	43	20	
Rain			1		1	3	6	2	2	1	2	18
Pre-Requisite			2	2	1			7	2			14
Design			1			1			4	2		8
Submittal				2	2	2						6
Other		1					1			1	1	4
Space										1	2	3
Equipment			2									2
Labor										1	1	2
Materials				1								1
Contract												0

Figure 4: Pull process of Last Planner

The last planner provides various possibilities to find the PPC ('percent plan complete'; i.e., percentage of assignments completed) , weekly progress The following are the graphs generated from CCSR project by implementing last planner process. The CCSR Project was to build laboratory for Stanford University for which the general contractor was Linbeck Construction.

1.2.6 Project Goal

(The application should provide features for planning and reporting the activities.support tasks like planning and reporting, but must be easy to use. Usability is an important aspect for this application. Based on findings in the early stages of the project, the students will further develop an existing software

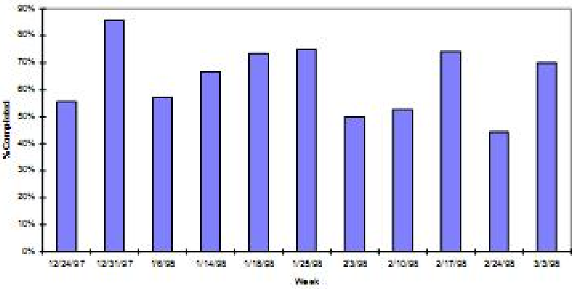


Figure 5: CCSR Process Weekly Report

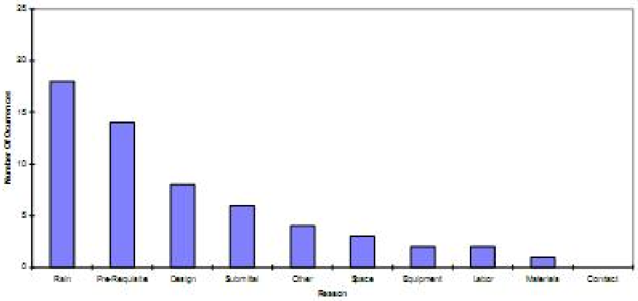


Figure 6: CCSR Reasons

prototype.). . . . To be done

1.2.7 Similar Products

Presently, the activities at the construction sites are manually maintained by recording the activities on paper. It becomes difficult for Foreman to arrange and attend the meetings with all the workers when working in the construction sites. It is rather less efficient way of recording and maintaing the activities as most of the time is wasted in waiting for others.The commonly followed procedure is Sticky notes or Papers on wall.

1.2.8 External Conditions

People and Companies The client: A representative from a group should regularly (i.e. Weekly or twice in a week)contact the client via email or any video conference. The meetings should focus upon updating the status of the project and taking inputs and requirements needed for the project. The supervisors: The team should meet the supervisors weekly and update the status of the report and take suggestions if needed. The group: See section xxxxx for details about the roles of the team members

Computers : The team members will use their personal laptops for the development. The code will be stored in the already existing GIT repository provided by the client. Each team member will use their personal laptop for the project. The final product will be used in the construction field. The application can be used either weekly or daily (depending upon the users who are using it)to monitor the activities at the construction site.

1.2.9 Duration

The work load for each member is expected to be 312 working hours. In our case, we have 8 members in our team this constitutes 2496 working hours.

- Project start: 31st of August, 2010.
- Project end and Final presentation: 25th of November, 2010.

2 Planning

The purpose of this chapter is to give reader an overview on the Project planning. The chapter begins with a Project plan in section 2.1 , which includes development model , sprints . In section 2.2 , we discuss on Orgaization. In section 2.3 we discuss about the risk management which includes Risk Treatment, Risk Identification and Risk Asessment. At the end of the chapter in section 2.4, we discuss on the quality assurance of the project.

2.1 Project Plan

The total duration for the project is 13 weeks . The initial two weeks shall be used for configuring our machines with the development server and existing project , Team building activities and project planning. After this we will start our sprints of two weeks duration. The first sprint will be prestudy and next sprints will be code development and testing iteratively.

2.1.1 Planning

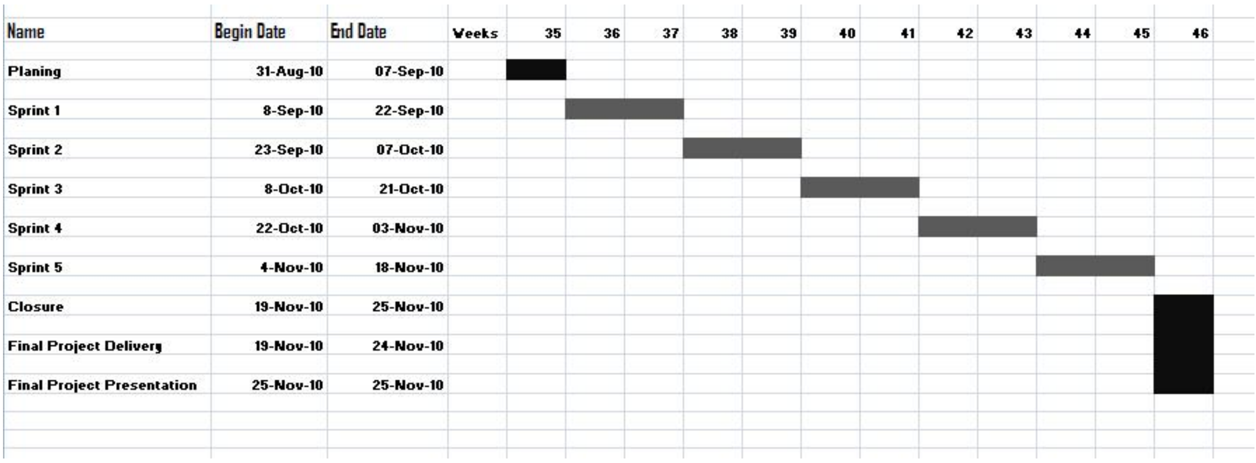
The team organization started after the first customer meeting on Tuesday ,August 31. The team sugggested the following tasks in Planning phase.

Team sheet	
1	Understanding the requirements of project
2	Getting overview on tools such as Latex and Git
3	Brainstorming and Addressing Usability requirements for project domain
4	Team organization
5	Planning schedule for Development
6	Team Building
7	Preparing Use case diagrams

We shall have a meeting at the end of this phase to discuss about the usability issues and project development plan.

2.1.2 Development model

The team has decided to use the Scrum development method (Pls. refer section 3.2 for more details). In our first meeting with customer , customer suggested us to use scrum based development methodology as they wanted to check the progress regularly. The Team agreed with customers to use scrum based model and we felt that we can use sprints to solve issues from the product backlog. We have decided to use customers feedback after each sprint in order to address the usability issues. The following figure shows the schedule of our project.



Name	Begin Date	End Date	Weeks	35	36	37	38	39	40	41	42	43	44	45	46
Planing	31-Aug-10	07-Sep-10		■											
Sprint 1	8-Sep-10	22-Sep-10			■	■									
Sprint 2	23-Sep-10	07-Oct-10					■	■							
Sprint 3	8-Oct-10	21-Oct-10							■	■					
Sprint 4	22-Oct-10	03-Nov-10									■	■			
Sprint 5	4-Nov-10	18-Nov-10											■	■	
Closure	19-Nov-10	25-Nov-10													■
Final Project Delivery	19-Nov-10	24-Nov-10													■
Final Project Presentation	25-Nov-10	25-Nov-10													■

The total workload, as given from the project description, is 2496 hours. We have decided to use scrum methodology for development and have 5 sprints comprising of two weeks each. This will help us to develop the project iteratively. The main challenge for team is to address the usability concerns of the project, using sprints of two weeks is the good way to get feedback on usability from customers. In each sprint, we will take some of the tasks from the product backlog both technical issues and also the usability. In case of technical issues , we priortise the functional requirements and try to solve them as per priority. For usability problems, we will develop different prototypes or sketches of the application and get the feedback from the customers. Upon approval from the customers , the usability tasks(i.e. the approved user interface sketches) will be solved. The following table illustrates the activities

Sprint Workflow		
Sprint 1	Product Backlog Usability proposals (i.e. UI sketches) Project Plan Testing	Week 36 and 37
Sprint2	Product Backlog Usability proposals (i.e. UI sketches) Implementation Testing	Week 38 and 39
Sprint 3	Product Backlog Usability proposals (i.e. UI sketches) Implementation Testing	Week 40 and 41
Sprint 4	Product Backlog Usability proposals (i.e. UI sketches) Implementation Testing	Week 42 and 43
Sprint 5	Product Backlog Usability proposals (i.e. UI sketches) Implementation Testing	Week 44 and 45
Closure Week 46	Project presentation (i.e. UI sketches)	Product delivery

./Images/organization.jpg

Figure 7: Project Organization Chart

First Sprint

The first sprint includes getting familiar with the product backlog and planning the project development. The main concern of the project is Usability i.e. the application should be easy to use. In order to address the usability concerns , team has decided to use 4 fold method suggested for usability tests. Accordingly, the first step includes the objectives and second step includes the designs or sketches . Third step is getting feedback from the customer about the sketches. The final step includes the implementing the selected sketch in the application.

Sprint two to five The sprints 2 to 5 consists are general sprints. The process being followed for these sprints are same as first sprint. In each of these sprints, the requirements are generated from the usability sketches and product backlog. At the end of each sprint, The sprint review meetings are conducted both with team members and customers. The code changes and sprint progress are discussed with the customers. The meeting will produce some feedbacks and requirements for the new sprint. As it is very common in scrum development that requirements will change after each sprint. So the requirements from the customers are taken at each sprint to generate new requirements.

2.2 Organization

This section discusses about the roles and responsibilities of team members team members. The following figure illustrates the organization of project.

2.2.1 Roles

(To be done)

Roles		
Project Manager:	Product Backlog Usability proposals (i.e. UI sketches) Project Plan Testing	Week 36 and 37

3 Priliminary Studies

The aim of the preliminary studies is to give an overview of the information needed before the development of the project starts. The following elements are covered by this preliminary study:
Limitations: The observed limitations the frame of this project, in a general term, sets.
Scrum: The methodology of the agile project development used.
Study of technologies and software: Analyzing the tools, and the constraints they imposes.

3.1 Limitations

As a software to be developed for a comercial ïňĄrm, there are certain limitations set. This, in order to avoid work who is outside the scope of the project. Those limitations are:
Responstime: Complexity of queries to the database system must be less than 0.1 seconds.
Browsers: Will not support backward compatibility for browsers who is not updated with the newst release. (Exception is for Micosoft Ecplorer, whom the web page will be working from 7 and newer).
Handheld devices: Handheld devices with a touch screen will not be speciïňĄcally supported.

3.2 Scrum

Scrum is the methodology of the agile project development used. Intitially, scrum was proposed as the prefered methodoly by the customer, and, during the intial stages of this project, veriïňĄed as the prefered methodology in order to reach the goals, set in the guidlines for this project.

3.2.1 Why is scrum the prefered methodology?

The main focus is this project is usability. Does usability corresponds to a developmen cycle, speciïňĄed by the scrum Methodology? âĂŞ The scrum methodology focuses on short development cycles, giving workable program for every sprint, making the road open for a user-feed-back-responed track of development, implying that, ref ïňĄgure above, every sprint adds some functionality to the system.

3.2.2 Daily Scrum

Gaing understanding of the other team members work, and the motivating eïňĂect about making commitments to tasks, is, by the group, regarded as a key factor in a project development. Therefore, a daily scrum meeting isto be held. The tree questions given to each team member is: 1. What did you do yesterd, 2. What will you do today?, 3. Are there any impediments in your way?.

3.2.3 Scrum Review Meeting

At the end of each sprint, a presentation will be given to the customer by Skype, togheter with the resulting code, uploaded to their test facility.

3.3 Study of technologies and software

Analyzing the tools, and the constraints they imposes, opens the horizons, and enhances the understanding. Sleceting the best mathing tools for the task, reduces the risks of uncertainty, increases the productivity and opens a path thorugh the uncertainty.

3.3.1 Model View Controller

In the limit of requirements for the system, speciïňĄed by the client, was the speciïňĄcation of using the pattern of a âĂIJModel View ControllerâĂİ(MVC) principle. The logic behind the MVC principle is to layer the code, and thereby enbale modiïňĄcations on one layer, without the need of cascading changes in the other parts of the code. (Ref http://martinfowler.com/eaaDev/uiArchs.html). The isolation of the data, as described above, enables:
Model: Storage keeping of data, in order to render information in the view.
View: ModiïňĄcations of the user interface (i.e. how the data is displayed), without a change in the functionality, or the structure of the model used, itself.
Controller: ModiïňĄcations of the functionality (i.e. the speed of execution), without a change in the user interface, or the structure of the model used, itself.
Making a strong seperation between the model and the presentation (view and controller), decreases the complexity, whom this model causes, enables rapid changes in code, due to the seperation caused by the layers. Using the MVC principle, a well known, de facto, standard, eases the time-to-understand curve for future developers, reducing the cost for reusing the code for future applications.

3.3.2 Python

The framework given to the team was written in Python, a programming language. This section includes a discussion around why the team concluded that the language would solve the task well.

3.3.3 Curve of learning

Due to Pythons semantic structure, the curve of learning is higher, compared to other languages, like PHP. This potential upside might, not neccassary, be a good thing, due to its potential of creating excessive maintanenace costs in the future. Thereby, in order to avoid this pithole, documentation would need some high light during the process of working.
The history has it that Python is not aWeb language, compared, for instance, to PHP. In recent years, this has changed, thanks to the introduction of framworks, removing the hinders, opening the road, and thereby the ability creating complex web pages with the language Python.

3.3.4 The risk of an open source language languish

Python is an open source language. Most open source languages has a tendency of languishing. Due to the widespread usae of the language, among developers on the web, the tendency is that it will last for some years to come. Thereby, the team evaluates the risk in the language python to languish as negligeble.

3.3.5 Library, the availability of tools

Due to the widespread usage of the language, libararies, like load testing tools and web framworks, has been developed, with the implication that a community, giving repons the question arising, exists.

3.3.6 Django Framwork

As for Python, the Django Framwork was chosen by the customer as the content management system (CMS). In order to verify that Django ïňĄlled the needs whom the requirements of the project builds upon, a review was made. This section includes an a discussion about why the Django environment fulïňĄlls the need of the tasks set during this project.

3.3.7 MVC support in the Django environment

Django imposes the MVC principle on the programmer, or rather MTV framevord, as deïňĄned in (ref http://docs.djangoproject.com/en/dev/faq/general/, thereby preventing the programmer from the pitfalls, whom normally delays a software development project. The implementation of the MVC principle, may be deïňĄned by the following stages :
Model: The data, whom the model represents, is deïňĄned in the âĂIJmodels.pyâĂİ ïňĄle. This is done by using the Django ORM models (FIXME: Whats this??).
Template: A set of HTML templates. Those templates are based on the CSS description language. As the defenition in the section describing the MVC princple above states, only lighweight processing- and decision making is legal at this stage. The link between the view and the controller (below), is deïňĄned in the âĂIJurls.pyâĂİ ïňĄle, mapping the URLs to the view function.
View: Describes the presentation of the data for the user, but not how it is presented. Insted, it view describes wich data is presented. Therby, the view is a python callback function for a particular URL. Also deïňĄned as the âĂIJview functionâĂİ, is the stage in responsibility of the processing, making use of the models, and methods in the models. In addidtion, the Django framwork provides a large and active community, includes an automatically build admin and thorough documentation. The django framwork supports the model view controller ...

3.3.8 Version Control

Working on multiple ïňĄles creates the need for a ïňĄle merging- and conïňĆict handling tool. The git tool was the chosen tool from the customers point of view, but in order to verify that it as well fulïňĄlled the need of this project, a discussion was made. The conclusion, comparing the svn and the git tools, ref http://www.looble.com/git-vs-svn-which-is-better/, of the subsections below, were that software called Git ïňĄts the need of the project, and therefore became the version control system in use. Looking ïňĄrst upon the alternative system, the SVN system, it is described as a system with one central repository, thereby creating the need of connecting to the server in order to commit their changes. Compered to SVN, git, ref http://git-scm.com/about, is a distributed system, implying that every group member has a local copy of the repository, using the server as a synchronization tool. Git was verïňĄed as the best tool to use, due to the factors listed below:

Backup: The delay of getting a backup, if the system crashes, is avoided by keeping the same backup of the entire repository on several machines. Therby, the status of the server will not cause uncertainty during the development of the project.
Brancing and merging: The support Git gives of rapid branching and merging, enables multiple users working on the same ïňĄles.
Offline working: The opportunity of working oïňĎine, without loosing the advantages of the features of having a version control tool, is provided by Git.

3.3.9 Text Editors

The text editor for the report writing chosen, was latex.

3.3.10 Document Compiling

As a methodology for producing, editing and compiling the documentation, the make utiliy of linux were chosen.

3.3.11 Naming Conventions

In order to follow hte standards, the naming conventions used by the code are those speciïňĄed for the phyton language, ref http://www.python.org/dev/peps/pep-0008/.

Part II

Sprints

The first sprint is short sprint lasting for two weeks. In this sprint , we focus mainly on understanding the first impressions of the application. The first impressions are the look and feel of the application in the current state. As this is the first sprint , our focus was to understand the application and improvements required for application rather than implementation. The sprint 2 and later discusses mostly implementation part of the project.

4 Sprint 1

4.1 First Impressions

The following section contains tables which contains feedback on the last planner system during sprint 1 both functional, usability and appearance. Feedback has been given for every web-page and pop-up window of Last Planner and the most important output is first impression, negatives and positives, and suggestions for improving the current system.

State diagram of Last Planner Sprint 1 The following figure is a general overview of the system and shows the different states you can access from each web-page.

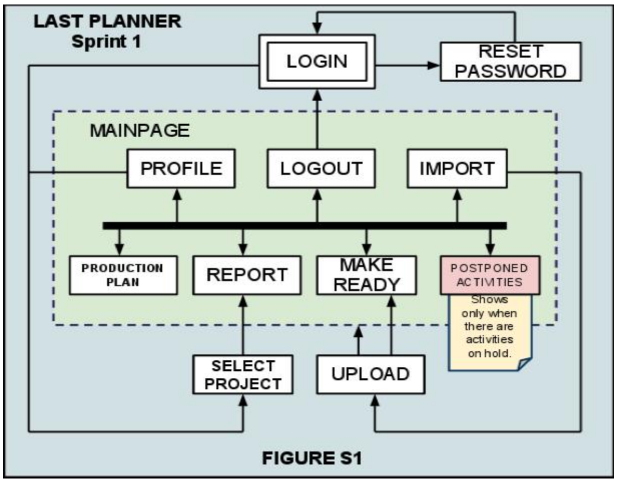
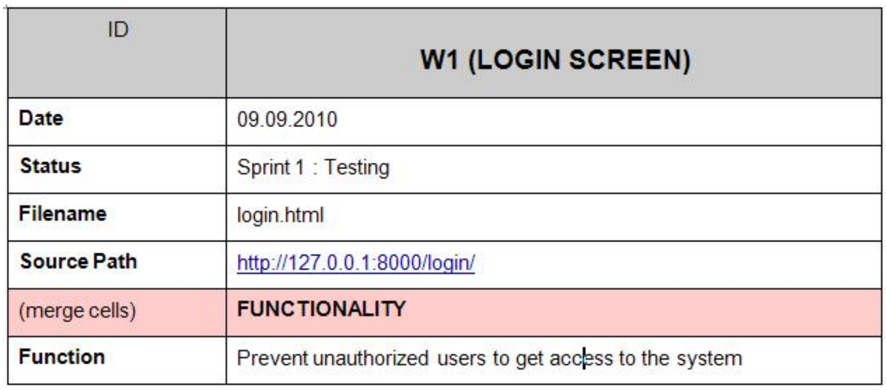


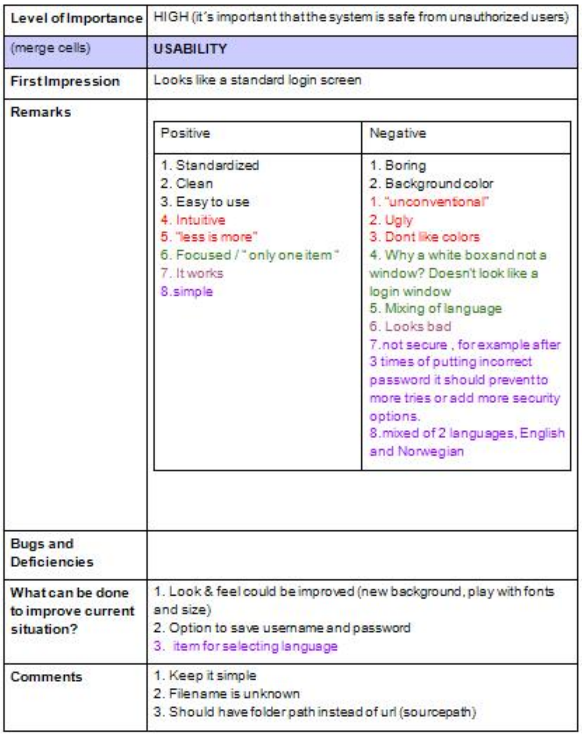
Figure 8: Overview of the system



ID	W1 (LOGIN SCREEN)
Date	09.09.2010
Status	Sprint 1 : Testing
Filename	login.html
Source Path	http://127.0.0.1:8000/login/
(merge cells)	**FUNCTIONALITY**
Function	Prevent unauthorized users to get access to the system

Figure 9: Login screen of the system

The following are the feedback tables for various screens of current system.



Level of Importance	HIGH (it´s important that the system is safe from unauthorized users)
(merge cells)	**USABILITY**
First Impression	Looks like a standard login screen
Remarks	Positive: 1. Standardized 2. Clean 3. Easy to use 4. Intuitive 5. "less is more" 6. Focused / " only one item " 7. It works 8.simple Negative: 1. Boring 2. Background color 1. "unconventional" 2. Ugly 3. Dont like colors 4. Why a white box and not a window? Doesn't look like a login window 5. Mixing of language 6. Looks bad 7.not secure , for example after 3 times of putting incorrect password it should prevent to more tries or add more security options. 8.mixed of 2 languages, English and Norwegian
Bugs and Deficiencies	
What can be done to improve current situation?	1. Look & feel could be improved (new background, play with fonts and size) 2. Option to save username and password 3. item for selecting language
Comments	1. Keep it simple 2. Filename is unknown 3. Should have folder path instead of url (sourcepath)

Figure 10: Login screen of the system

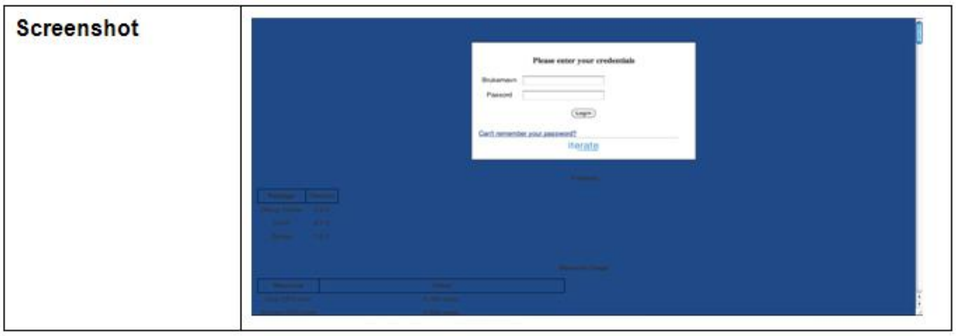


Figure 11: Login screen of the system

4.2 Spring Backlog

4.3 Implementation

4.4 Sprint Evaluation

5 Sprint 2

5.1 Spring Backlog

5.2 Implementation

5.3 Testing

5.4 Burndown Chart

5.5 Client Feedback

5.6 Sprint Evaluation

6 Sprint 3

6.1 Spring Backlog

6.2 Implementation

6.3 Sprint Testing

6.4 Testing

6.5 Burndown Chart

6.6 Client Feedback

6.7 Sprint Evaluation

7 Sprint 4

7.1 Spring Backlog

7.2 Implementation

7.3 Sprint Testing

7.4 Testing

7.5 Burndown Chart

7.6 Client Feedback

7.7 Sprint Evaluation

8 Sprint 5

8.1 Spring Backlog

8.2 Implementation

8.3 Sprint Testing

8.4 Testing

8.5 Client Feedback

8.6 Sprint Evaluation

Part III

Evaluation

9 Overall System Architecture

9.1 Overall Architecture

9.2 Further Development

9.3 Limitations

10 Project Evaluation

10.1 Using Scrum

10.2 Group Dynamics

10.3 Learning Python.........?

10.4 Risks

10.5 Time Estimation

11 Future Work

12 Appendix

13 Bibliography