

Final report

Workload forecast

Kazarnikov Alexey

Lavrentyeva Galina

Fedotenkova Maria

Lappeenranta University of Technology

CT30A9300 Code Camp on Agile Java Development by Capgemini

1 Task

This is the final report of the project for the Code Camp on Agile Java Development by Capgemini. During three weeks our task was to implement the application Workload forecast for Capgemini in the web framework Grails by using Agile development and Scrum. The full description of our final project and stages of the development process will be presented further in this report

2 Scrum

2.1 Main features

Scrum is one of the most popular agile software development methods for managing project development processes. One of the main reasons for its popularity is its simplicity. Scrum focuses on project management institutions where it is difficult to plan ahead. It allows to provide new software features with highest priority every fixed and small-time iteration (called sprints). Strictly fixed duration of each sprint makes the development process predictable and flexible.

The whole scrum process can be seen on the picture below. There are only 3 roles in Scrum methodology:

1. Scrum master
2. Product Owner
3. Team

Scrum Master - is the most important role. Scrum Master is responsible for the success of a Scrum project. In fact, the Scrum Master is the interface between the management and the team. Typically, the role in the project is the project manager or timlid. It is important to emphasize that the Scrum

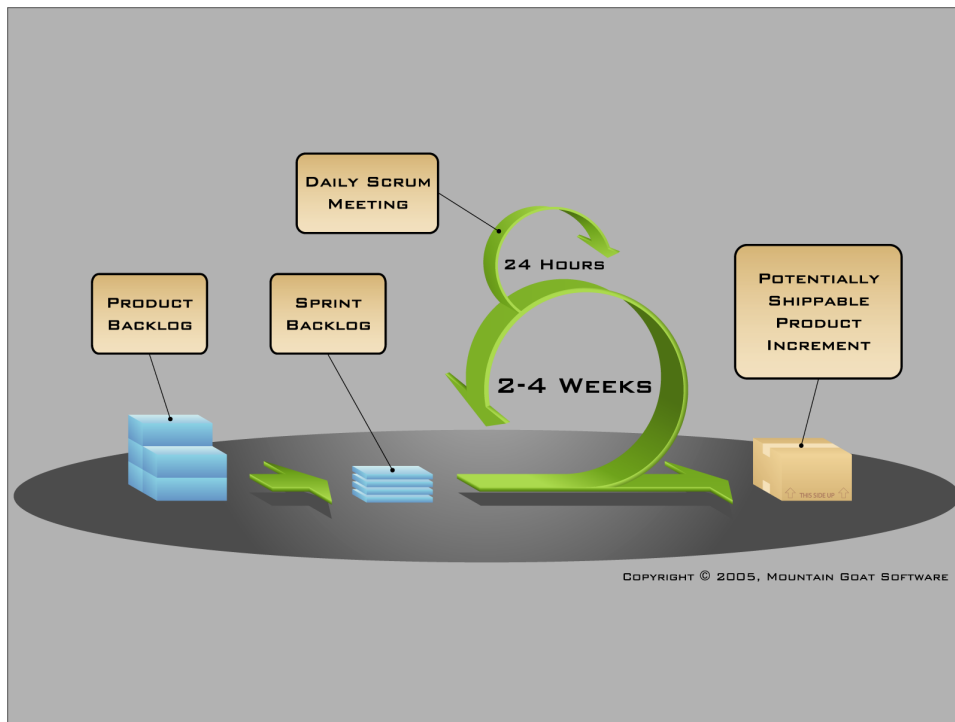


Figure 1: Scrum lifecycle

Master does not give the task team members.

Main responsibilities Scrum Masters are:

- Creates an atmosphere of trust
- Participates in meetings as a facilitator
- Removes obstacles
- Makes problems and questions open to discuss
- Creates Daily Scrum Meeting and monitors the progress of the team with the Sprint Backlog, noting the status of all tasks in the sprint.

Product Owner is responsible for the product development. Typically, this is product manager for product development, project manager for interior design and a customer representative for custom development. Product

Owner is a single point of the final decision for the team in the project, that is why product owner is one person.

Product Owner responsibilities are as follows:

- Responsible for building product vision
- Manage expectations of customers and other interested parties
- Coordinates and prioritizes Product backlog
- Provides a clear and testable requirements team
- Interact with the team and the customer
- Responsible for the acceptance of the code at the end of each iteration

Product Owner sets the problem for the team, but not for the concrete team member during the sprint.

In Agile *Team* is self-organizing and self-managed. Team is responsible for implementing the work of current sprint. In Scrum, the contribution of individual team members is not measured, as it destroys the self-organizing team.

Responsibilities of the team are:

- Responsible for evaluating the elements of backlog
- Decide on the design and implementation
- Develops software and provides it to the customer
- Monitor their own progress (with Scrum Master).
- Responsible for the result to the Product Owner

Team size is limited to the size of a group of people who can effectively communicate face to face. The typical size of the team - 7 plus or minus 2.

To facilitate communications team should be in one place (colocated). Preferably in a common room in order to reduce barriers to free communication. Team should provide all necessary things for comfortable operation, provide boards and flip charts, provide all the necessary tools and environment to work.

2.2 Sprints

In Scrum iteration is called Sprint. Its duration is 1 month (30 days).

Sprint is the result of the finished product, which can be transmitted to the customer (at least, the system should be ready to be shown to the customer).

Short sprints provide rapid feedback to the project team from the customer.

The customer is able to flexibly control the scope of the system, evaluating the results and offering a sprint to create a functional improvement. Such improvements fall into Product Backlog, prioritized along with other requirements and can be scheduled for the next (or one of the following) sprints.

Each sprint is a small waterfall. During the sprint all the work connected with gathering requirements, design, coding and testing of the product is being done.

Scope sprint should be fixed. This allows the team to make commitments to the amount of work that must be done in the sprint. This means that the Sprint Backlog may not be changed by anyone other than the team.

At the beginning of each sprint, *sprint planning* is held. During this meeting the following steps are discussed:

- Select what work is to be done
- Prepare the Sprint Backlog that details the time it will take to do that work, with the entire team
- Identify and communicate how much of the work is likely to be done

during the current sprint

Every day during the sprint, a project status meeting (called *daily scrum* or *daily standup*) is held. It takes about 15 minutes and usually occurs before team starts the work. All members of the Development Team should come prepared with the updates for the meeting. On this meeting all are welcome, but normally only the core roles speak. Each team member should answer 3 questions:

- What have you done since yesterday?
- What are you planning to do today?
- Any impediments/stumbling blocks?

Any impediment/stumbling block identified is documented by the Scrum Master and are resolution of this meeting. One of the most important points is that no detailed discussions shall happen in this meeting.

At the end of a sprint cycle, two meetings are held: the *Sprint Review Meeting* and the *Sprint Retrospective*. During the Sprint Review Meeting team review the work that was completed and not completed and present the completed work demo to the stakeholders. On the Sprint Retrospective all team members are analyzing the past sprint(What went well during the sprint? What could be improved in the next sprint?) and make continuous process improvements.

2.3 Sprints in Workforecast project

Table 1: Sprints

Sprint 1
Collecting client needs
Collecting and analyzing requirements
Project design
Web-pages design
Cabinet interfaces
DataBase architecture
DataBase realization
Login
Testing
Product presentation
Sprint 2
HTML pages
Ablity to see personal worforecast for consultant
Searching consultant function
Ablity to see worforecasts of any consultant
Administrator(for security needs) functionality
Testing
Sprint 3
HTML pages (from sprint 2)
Adding new project and new client to DataBase
Searching free consultant function
Calculating summary information for director
Sending messages functionality
Testing
Bug fixing
Project documentation

2.4 User stories

Here are user stories and corresponding features, that we planned in sprints.

Figure 2: User stories and features

Priority	Role	User stories	Features	Status
1	As a consultant	I can login	Login logic and realization	Done
1		I can see my workforecast for any month	Logic, Interface, realization	Done
1		I can edit my workforecast for any month	Edit button, edit interface and edit logic	Done
3		I can add new activities to DataBase	Add button, add interface, add logic and realization	Done
3		I can edit activities, stored in Database	Add button, add interface, add logic and realization	Done
2		I can see messages from controller	Messages logic, view, realization	Done
1	As a manager	I can login	Login logic and realization	Done
3		I can add new activities to DataBase	Add button, add interface, add logic and realization	Done
3		I can edit activities, stored in Database	Add button, add interface, add logic and realization	Done
1		I can see workforecast of any consultant	Searching consultant function logic, interface, realization	Done
1	As a controller	I can login	Login logic and realization	Done
1		I can see workforecast of any consultant	Searching consultant function logic, interface, realization	Done
2		I can send messages with comments to any consultant	Messages logic, messages interface, realization	Done
Priority	Role	User stories	Features	Status
1	As a resourcing representative	I can login	Login logic and realization	Done
1		I can see workforecast of any consultant	Searching consultant function logic, interface, realization	Done
2		I can search for free consultant in any month	Searching free consultant function logic, interface, realization	Done
1	As a director	I can login	Login logic and realization	Done
2		I can see summary information from the workforecasts	Calculating summary information, interface, realization	Done
3		I can see workforecast of any consultant	Searching consultant function logic, interface, realization	Done
1	As an administrator	I can login	Login logic and realization	Done
1		I can see personal information of any consultant (login, password)	Searching user function logic, interface, realization	Done
1		I can edit consultant's personal information	Edit button, edit interface and edit logic	Done

3 Program structure and features

Our Workload forecast application is useful web-tool, that can be used as a usual web-site through any web-browser. This application allows every one in the company (from consultant to director) get information about the workload forecast for any month and year of concrete project or concrete consultant and change that information according to their positions.

First page contains short introduction to the project, that can be useful for new users. It also allows users to login. After input login and password user can push the button 'Login' and if input data was correct user will be identified according to role of user name. According to the role user has its own abilities.

Consultant, for example, have easy tool to view and edit his own workload forecast for concrete month, that is performed as a color table. Manager can see personal forecast of any consultant. Searching functionality is very useful and clear: manager can search concrete consultant by name or using the alphabeth with drop-down list for each letter

Controller can also see any personal forecast and send messages to consultant with any remarks very fast using simple functionality for that.

Director apart from viewing forecasts capability can have the summary information about projects or consultants. That is very useful because directors mostly need only the total information as average hours for project and others.

Resource representative also has the viewing forecast capability, but the most demanded feature for this role is searching free consultant for some week in concrete month. Searching functionality is made that way, that not only free consultant can be searched, but also consultant with some minimum percentage of working time, this rate is set by resource representative as a searching criteria.

To make user registration easier to organize we introduced the position of

administrator, who can see all users information and create new users with passwords.

In general terms, Workload forecast, that was developed especially for Capgemini company is colorful, clear, fast in use and safe from cheating time management tool. There is no need to spend time for teaching all staff how to use it, because all the functionality is obvious.

4 Description of functionalities

The solution works with individual work forecasts of the employees. Consultants can enter and edit their forecasts. Managers can view the forecast of every consultant and edit it. Other users of the system (resourcing, controller, director) can only view the forecasts of the employees but not edit them.

However, one more role exists in the system. An administrator manages the account data of the users. He does not have access to the forecast data, but he can create, edit and delete accounts of any other user.

The Grails framework was used for the development of application. For data storage MySQL database was used.

The system is organized in a form of web application and consists of several pages. Here, the purposes of each page are stated:

Authentication page

This is the starting point of the application. Here user can enter the system and after that they are redirected to the page, designed for the corresponding role. The user can only view the pages, designed for his role.

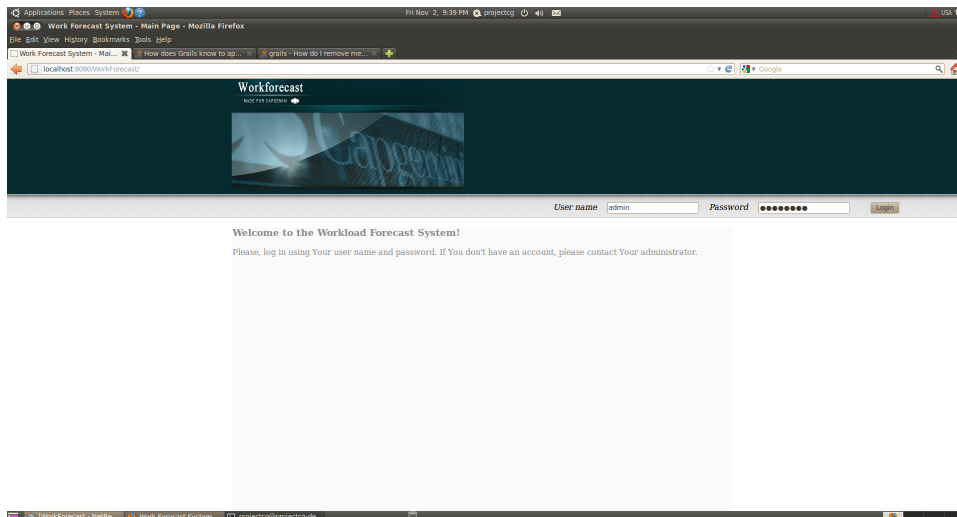


Figure 3: Authentication page

Administration page

Only administrator can enter this page. Here user account data can be edited, new users can be added to the system, or existing ones can be deleted. The following information is stored in the database:

1. Account name.
2. Password.
3. Real name.
4. Role.
5. Employment year. (is needed for more accurate visualization of personal forecasts)

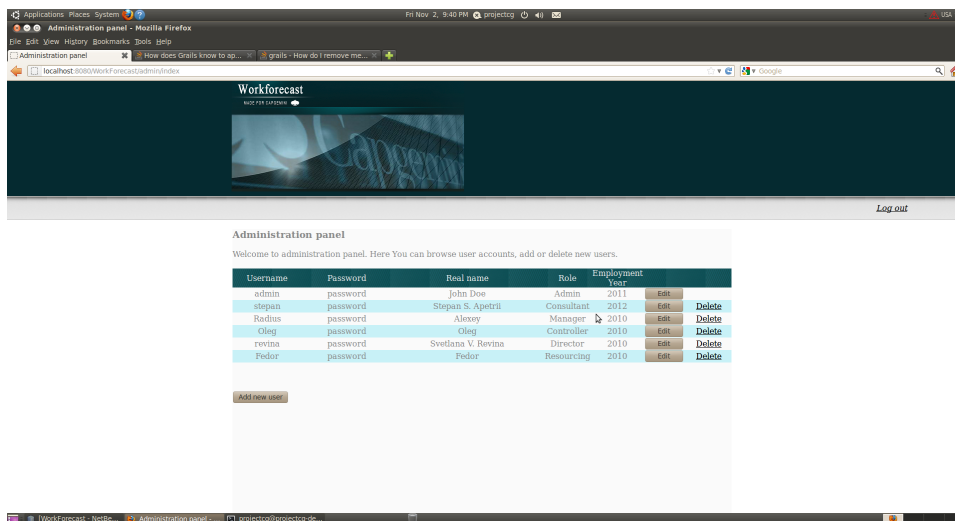


Figure 4: Administration page

Consultant page

The personal cabinet of a consultant. The consultant can enter the forecast data here, add or remove the projects, in which he or she participates and create or edit a personal forecast. This can be done by two ways: by editing

the forecast of a single activity, or by editing the full forecast, which contains the whole activities in a single page. Also, in the full forecast page ARVE, URVE and COR values are shown for each month. In the personal cabinet the messages, sent by controller to the consultant are shown.

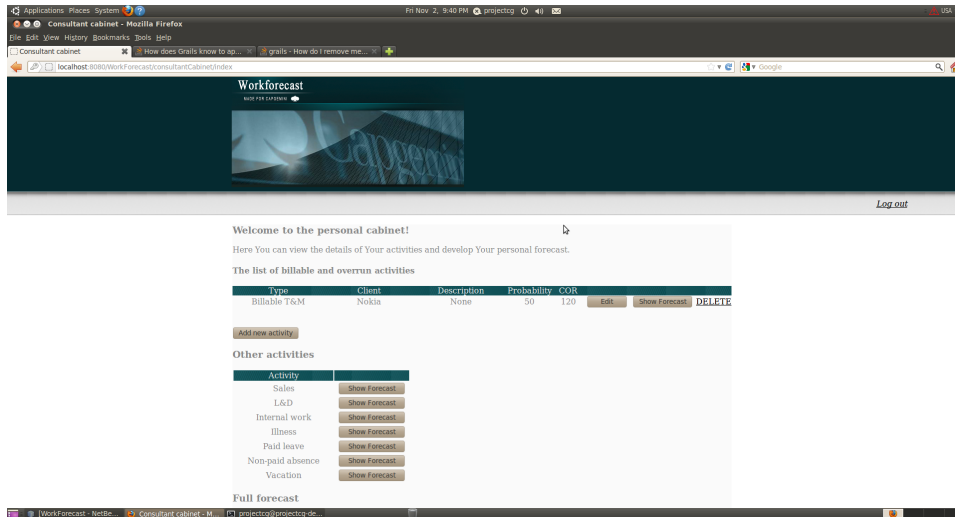


Figure 5: Consultant page

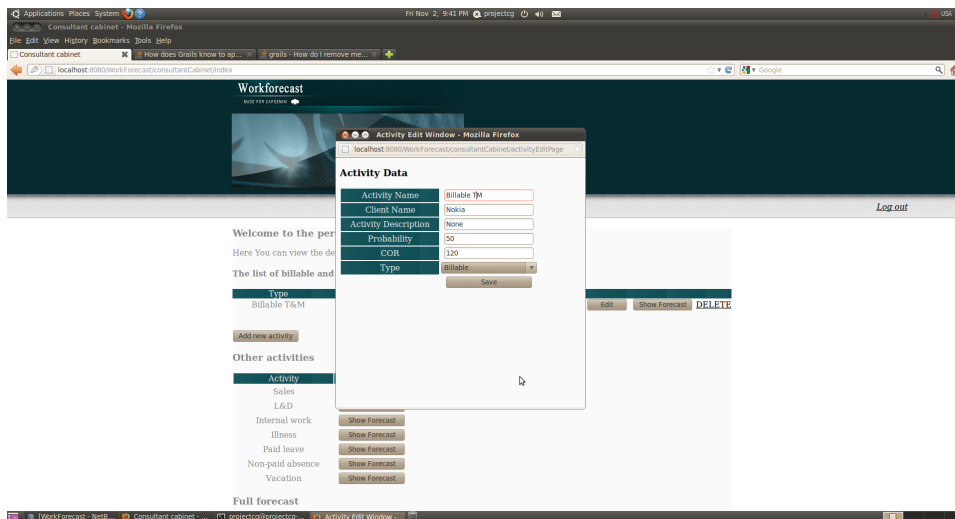


Figure 6: Consultant page

Manager page

By using this page, manager can view the forecast of any consultant and edit

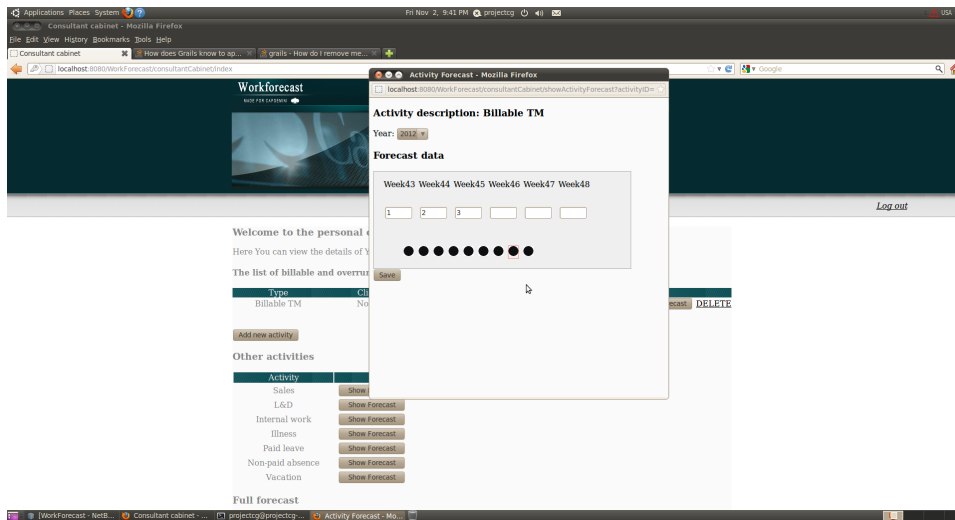


Figure 7: Consultant page

it. He can also view the forecast for single activity, of full forecast.

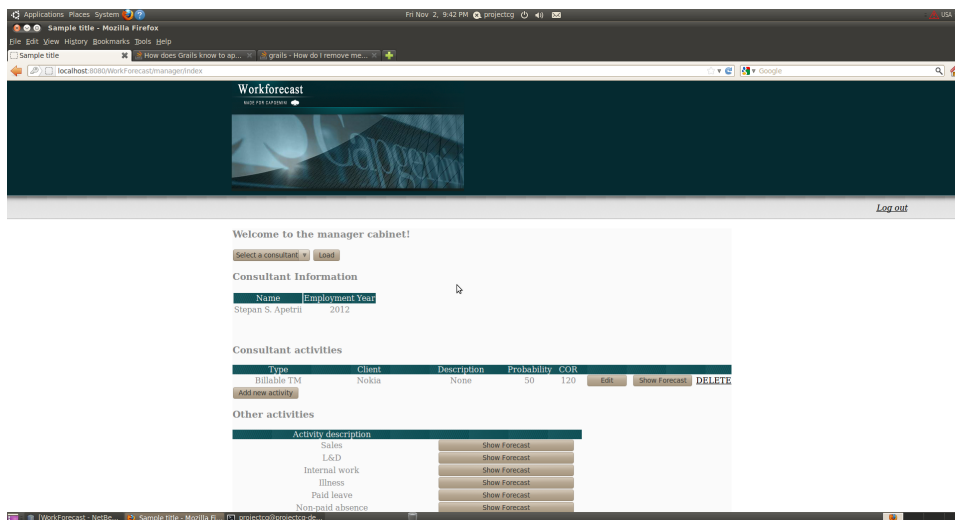


Figure 8: Manager page

Controller page

Here controller can view (but not edit) a forecast of any consultant and comment the forecast. He can also comment the forecast of the employee.

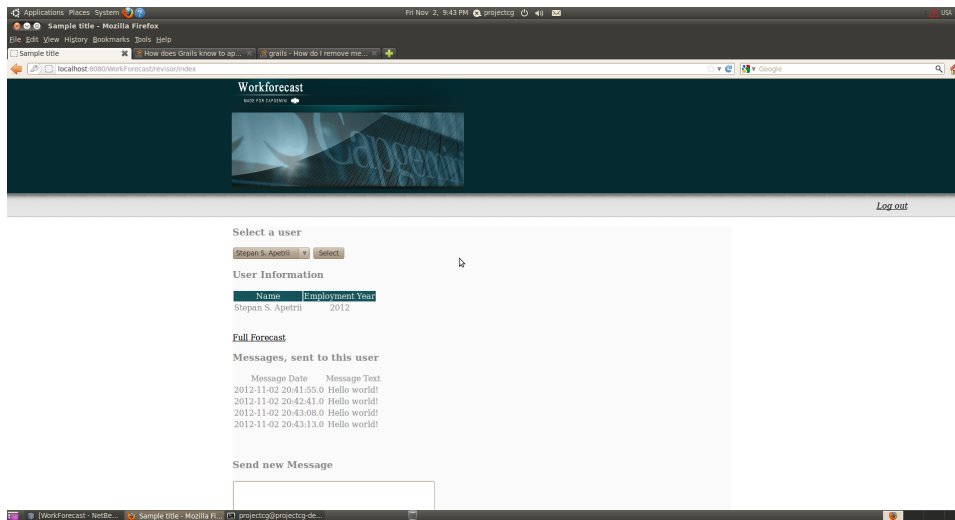


Figure 9: Controller page

Resourcing page

By using this page, resourcing can browse the forecast for a single consultant of search for consultants, who match certain criteria.

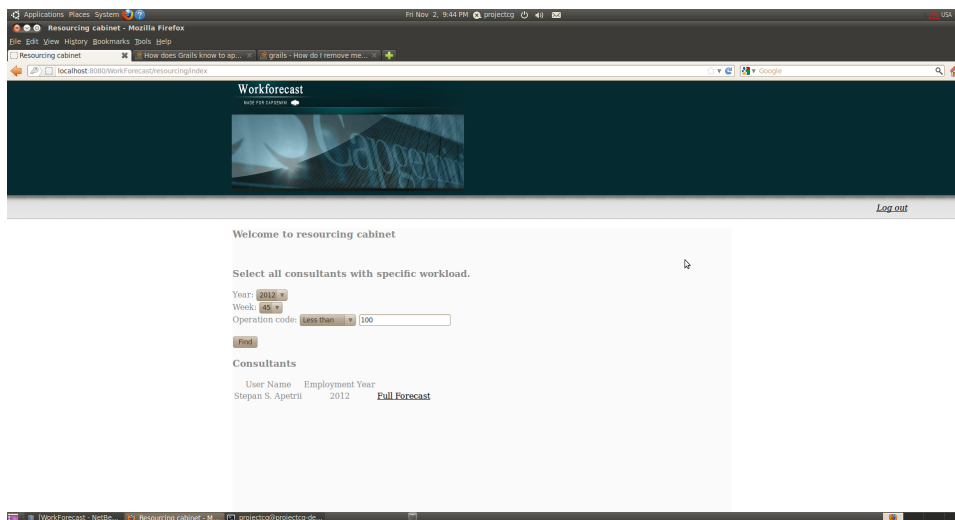


Figure 10: Resourcing page

Director page

Director can login and view the information of any consultant. He cannot

edit the information. He can view the full forecast of any employee.

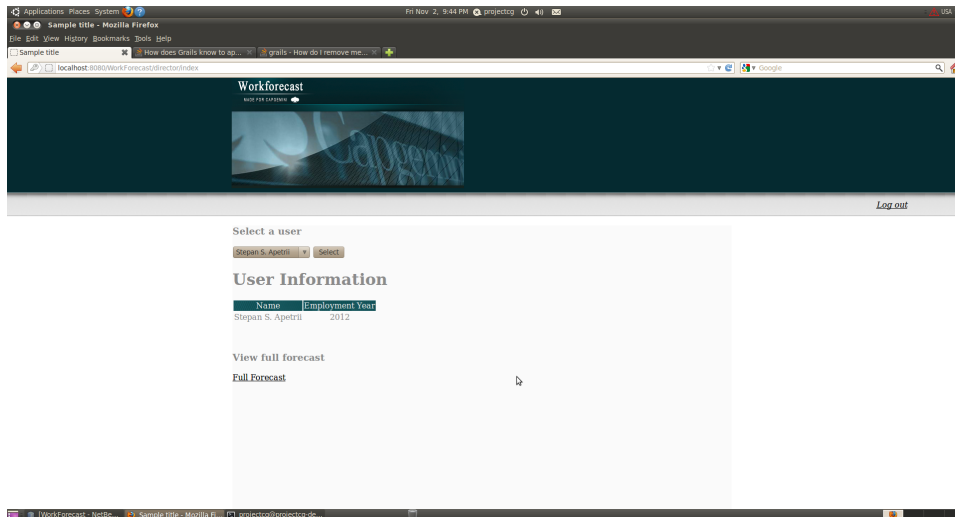


Figure 11: Director page

5 The development process and lessons learned

5.1 Development process

The main task of this Code Camp was to develop the full project for concrete client with special requirements.

Firstly, we read the task carefully and analyzed the lecture of client-company introducing. That helped us to know what is Capgemini and what they need from the product project, thus we collected client needs. After that, we clearly defined user stories, features corresponding to them and planned all sprints to finish work in time.

Almost the whole first day of Code Camp we were focusing of the project design and DataBase design. We were trying to discuss all details and future possible problems, that we can avoid before we meet them. We also set the priority for all defined features in order to get what we need to do for the first sprint. After that we decided to devide the development process into several parts:

1. Project implementation
2. View design
3. Documentation

Each member of the team was responsible for his own part. As we were doing all this step by step, it gives everyone the opportunity to controll the step of the deveopment process and management skills.

5.2 Lessons learned

5.2.1 SCRUM

In this Code Camp we were developing out project by using SCRUM Agile Project Management. By applying Scrum we understood how we can organize the development process, espesially when we are limited by short time, by deviding all work into several sprints, using the priority of features we

need to implement. Based on the experience we got, now we can access our abilities more reliably and plan work reasonably. Using sprint meeting we learned how to discuss actual problems, what have been done and what still need to be implemented.

5.2.2 Work with real client

During this Code Camp we were working with Capgemini representatives. We have improved our skills in working with real clients, needs, requirements. We also got the experience in presenting product for sale.

5.2.3 Team work

Working in team taught us how to divide development process in parts, listen to each other and trying to find appropriate solution in disputed situation, when each one have their own opinion. This opportunity was good because each of us was thinking about this project in different aspects, and thus we make the whole view of the project.

5.2.4 Work with other teams

There was one task for several teams. Almost none of us was familiar with Grails and it's facilities. We were all in the same situation and were learning together. Because of that we were learning from each other, asking and explaining others how to solve the concrete problem, despite the fact that we have a competition. That was new for us.

5.2.5 Grails