

Project report - group 5

- Smaranda Dungeanu
- Marek Furak
- Cristian Nita

Introduction

The project of our choice was a school's IT System, where teachers can add other users of different types (so, with different level of access), set up semesters, periods, tasks and study points. More details about the functional and non-functional requirements will be presented further down, in the product backlog.

project link: <http://massacre.azurewebsites.net>

web server's GitHub link: <https://github.com/SmarandaDungeanu/Semester3Project>

java server's GitHub link:

<https://github.com/theIncredibleMarek/3rdSemesterProjectJavaServer>

Scrum Methodologies and Agile Planning

During the development we tried to follow as much as possible the agile methodologies, and try to learn from applying them into practice. Our project was divided into 3 sprints, with a variable number of working days, as you will see further on.

Every daily scrum meeting had between 15-20 minutes. They helped us divide the work in a very constructive way and keep our workflow and development speed constant.

We worked both from school(when we were together) and at home, followed by a detailed presentation of what we did by ourselves. On the end of the project period, we focused more on testing what we did previously, as we didn't follow a test-driven development for most of the project.

Product backlog:

ID	User Story	Status	Sprint number	Size	Priority	How to demo
1	As a teacher, I want to be able to add new different types of user accounts	completed	3	19.5	100	click on All Teachers - Add More Teachers; click on Classes - select a class - select a semester - click See students - click Add More Students
2	As a teacher, I want to be able to log in	completed	3	6	98	write your username and password on the right top of the page, then click Login or press Enter
3	As a teacher, I want to be able to reset my user password	0%	3		60	
4	As a student, I want to be able to log in	completed	3	3	96	write your username and password on the right top of the page, then click Login or press Enter
5	As a student, I want to be able to reset my user password	0%	3		58	
6	As a teacher, I want to be able to add a new class into the system	completed	1	14	95	click Classes - Your Classes - Add more classes
7	As a teacher, I want to be able to set up a new semester(with all its details) into the system	completed	1	1.5	90	click Classes - Your Classes - click one of the classes - Add More Semesters
8	As a teacher, I want to be able to set up a new period(with all its details) for any active semester	completed	1	1.5	89	click Classes - Your Classes - click one of the classes - click one of the semesters - Add more periods
9	As a teacher, I want to be able to set up a new task(with all its details) for an active period	completed	1	2.5	88	click Classes - Your Classes - click one of the classes - click one of the semesters - click one of the periods - Add More Tasks
10	As a teacher, I want to be able to see any class/semester/period/task's details	completed	1	4	85	click Classes - Your Classes - click one of the classes to see its details; click one of the semesters to see its details; click one of the periods to see its details; click one of the tasks to see its details
11	As a teacher, I want to be able to modify periods details	0%	3		40	
12	As a teacher, I want to be able to modify task details	0%	3		38	
13	As a teacher, I want to be able to remove a period from an active semester	0%	3		35	
14	As a teacher, I want to be able to remove a task from an active period	0%	3		33	
15	As a teacher, I want to be able to see a list of all classes/students and search for a specific student	completed	2	5	60	click All Students - use the searchbox; click Classes
16	As a teacher, I want to be able to add/edit scores for students for each task	completed	2	5	75	click All Students - click one of the students - click See profile - click one of the semesters - click one of the periods - click one of the tasks - edit Achieved Points in the right form and click Save
17	As a teacher, I want to be able to see scores for each student per semester/period/task	completed	2	6.5	70	click All Students - click one of the students - click See profile - click one of the semesters - click one of the periods
18	As a teacher, I want to be able to see reports for each class	completed	3	8.5	80	click Classes - click one of the classes - click one of the semesters - See students
19	As a student, I want to be able to see a semester/period/task's details	completed	3	13	83	click My Profile - select one of the semesters
20	As a student, I want to be able to add a photo with me	completed	2	13	10	click All Students - click one of the students - click See profile - Choose Files - Upload

Sprint 1 backlog:

User Story	Tasks	Day 1	Day 2	Day 3	Day 4
As a teacher, I want to be able to add a new class into the system	Design the database of the overall system, to match further needs	6	0	4	0
	Created the necessary functions in all the layers	0	0	3	0
	Created the forms for the webpage	0	0	1	0
As a teacher, I want to be able to set up a new semester(with all its details) into the system	Created the necessary functions in all the layers	0	0	1	0
	Created the forms for the webpage	0	0	0.5	0
As a teacher, I want to be able to set up a new period(with all its details) for any active semester	Created the necessary functions in all the layers	0	0	1	0
	Created the forms for the webpage	0	0	0.5	0
	Created the necessary functions in all the layers	0	0	0	1
As a teacher, I want to be able to set up a new task(with all its details) for an active period	Created the forms for the webpage	0	0	0	0.5
	Set up the deployment on Azure	0	0	1	0
	Created the necessary functions in all the layers	0	0	0	2
As a teacher, I want to be able to see any class/semester/period/task's details	Created the forms for the webpage	0	0	0	2
As a teacher, I want to be able to add new different types of user accounts	Started working on the java API	0	0	0	4

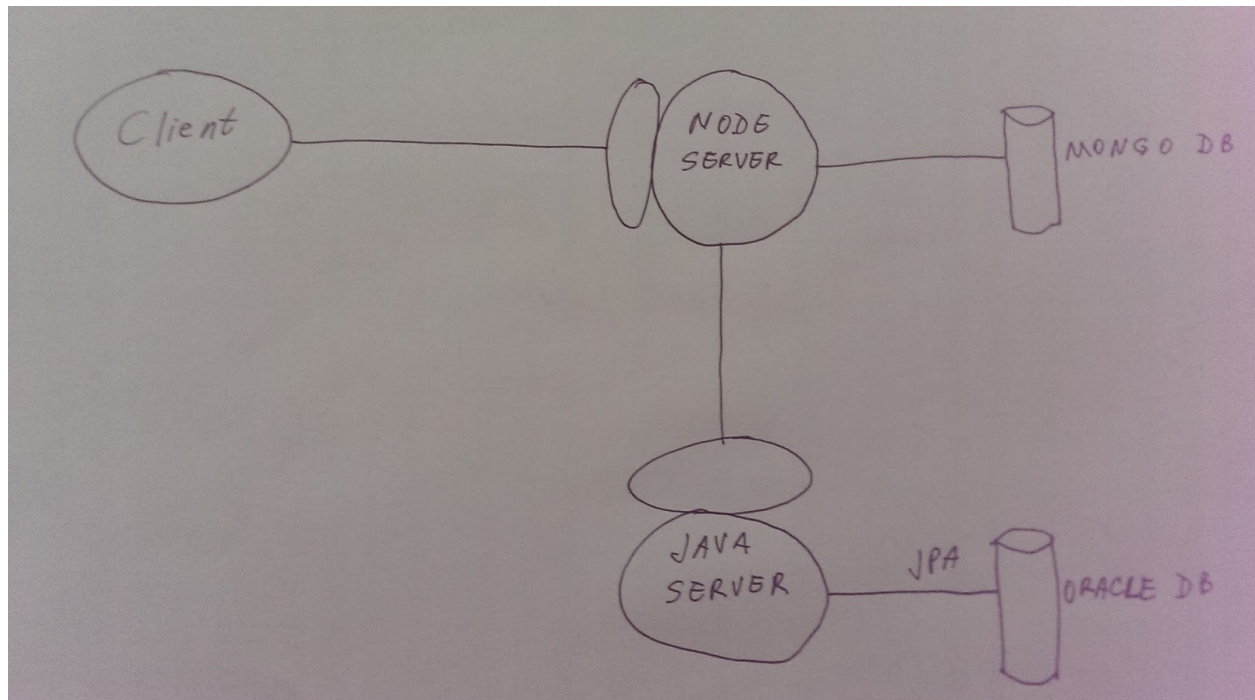
Sprint 2 backlog:

User Story	Tasks	Day 1	Day 2	Day 3	Day 4
As a teacher, I want to be able to add a new class into the system	Design the database of the overall system, to match further needs	6	0	4	0
	Created the necessary functions in all the layers	0	0	3	0
	Created the forms for the webpage	0	0	1	0
As a teacher, I want to be able to set up a new semester(with all its details) into the system	Created the necessary functions in all the layers	0	0	1	0
	Created the forms for the webpage	0	0	0.5	0
As a teacher, I want to be able to set up a new period(with all its details) for any active semester	Created the necessary functions in all the layers	0	0	1	0
	Created the forms for the webpage	0	0	0.5	0
As a teacher, I want to be able to set up a new task(with all its details) for an active period	Created the necessary functions in all the layers	0	0	0	1
	Created the forms for the webpage	0	0	0	0.5
	Set up the deployment on Azure	0	0	1	0
As a teacher, I want to be able to see any class/semester/period/task's details	Created the necessary functions in all the layers	0	0	0	2
	Created the forms for the webpage	0	0	0	2
As a teacher, I want to be able to add new different types of user accounts	Started working on the java API	0	0	0	4

Sprint 3 backlog:

User Story	Tasks	Day 1	Day 2	Day 3	Day 4
As a teacher, I want to be able to add new different types of user accounts	Finished working on the java API	4	0	0	0
	Added the options in the web pages and controllers	0	4	0	0
	Testing	0	0	3	0
As a teacher, I want to be able to log in	Created the necessary functions in all the layers	3	0	0	0
	Created the html for the webpage	0.5	0	0	0
	Testing	0	0	2.5	0
As a student, I want to be able to log in	Created the necessary functions in all the layers	1	0	0	0
	Created the forms for the webpage	0.5	0	0	0
	Testing	0	0	1.5	0
As a teacher, I want to be able to see reports for each class	Created the necessary functions in all the layers	0	4	0	0
	Created the forms for the webpage	0	2	0	0
	Testing	0	0	0	2.5
As a student, I want to be able to see a semester/period/task's details	Implemented uploading a file to the web server	3	4	0	0
	Created the forms for the webpage	0	2	0	0
	Testing	0	0	0	4

Architecture diagram

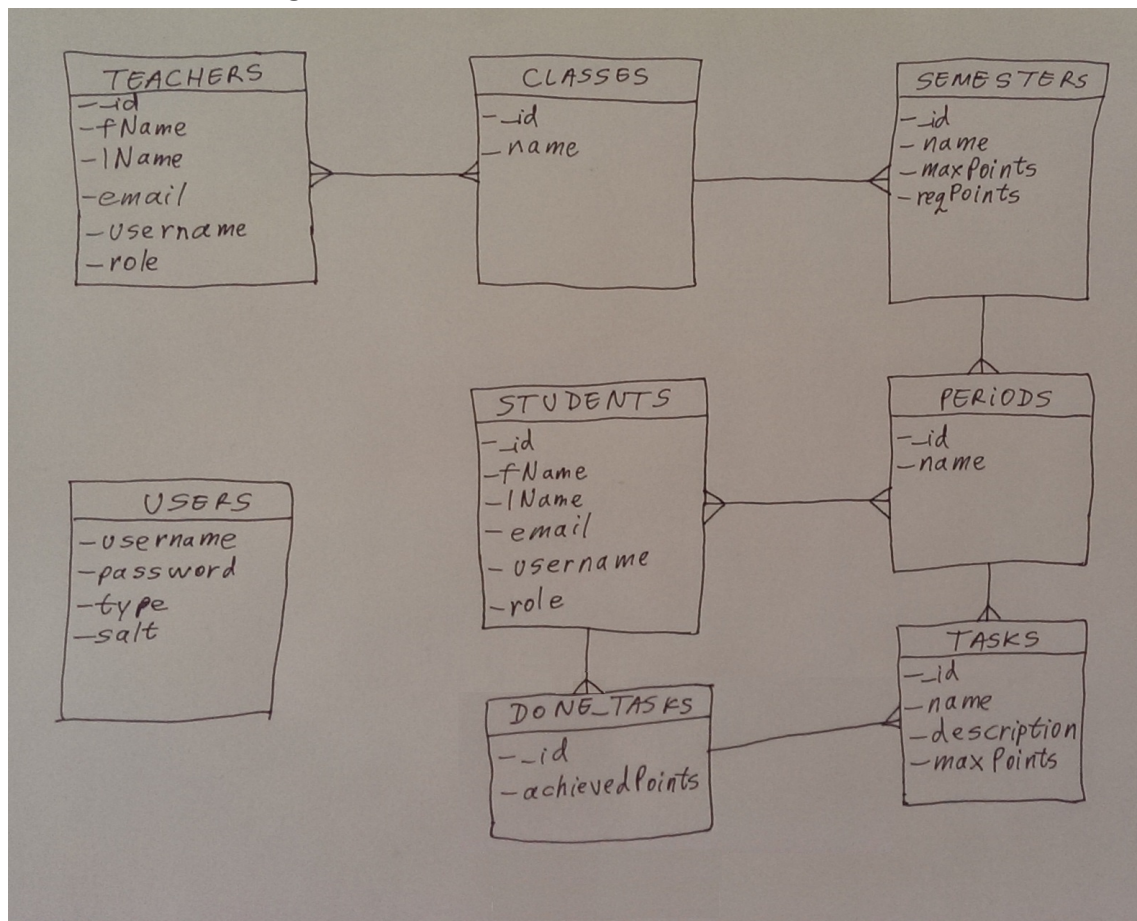


Site diagram

We tried to improve the user experience as much as we could, by keeping it simple: there are lots of links on the top bar for quick access to main pages, and, from there, a very simplified, suggestive menu in each page. The sitemap options will be displayed according to each user type's (guest, student, teacher) level of access to the actual website.

- index
 - Semester Project
 - All Students
 - Student name - See profile
 - Classes
 - Your Classes - Class name
 - Semester name - Add more periods
 - Add More Semesters
 - Assign New Teacher
 - Your Classes - Add more classes
 - My Profile
 - Semester name
 - All Teachers
 - Teacher name
 - Add more teachers
 - Sitemap
 - GitHub
 - Report

Database diagram



In this image you can see both our databases: the simple one, including only the users table is an Oracle database and it is only used by the login server (Java server) and the other one, with all the IT System's data - on mongolab, used in Node.

Wireshark sample

Wireless Network Connection - Graph Analysis			
Time	192.168.1.236	104.45.81.79	Comment
4.026225000	65492→80 [SYN]	80	TCP: 65492→80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
4.072015000	80→65492 [SYN, ACK]	65492	TCP: 80→65492 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1440 WS=256 SACK_PERM=1
4.072107000	65492→80 [ACK]	80	TCP: 65492→80 [ACK] Seq=1 Ack=1 Win=66240 Len=0
4.072602000	GET / HTTP/1.1		HTTP: GET / HTTP/1.1
4.145514000	80→65492 [ACK]	65492	TCP: 80→65492 [ACK] Seq=1 Ack=542 Win=131328 Len=0
4.158187000	HTTP/1.1 304 Not Modified		HTTP: HTTP/1.1 304 Not Modified
4.294811000	65493→80 [SYN]	80	TCP: 65493→80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
4.295067000	GET /app/app.js		HTTP: GET /app/app.js HTTP/1.1
4.295749000	65494→80 [SYN]	80	TCP: 65494→80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
4.296632000	65495→80 [SYN]	80	TCP: 65495→80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
4.338086000	80→65493 [SYN, ACK]	65493	TCP: 80→65493 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1440 WS=256 SACK_PERM=1
4.338346000	65493→80 [ACK]	80	TCP: 65493→80 [ACK] Seq=1 Ack=1 Win=66240 Len=0
4.338550000	80→65494 [SYN, ACK]	65494	TCP: 80→65494 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1440 WS=256 SACK_PERM=1
4.338550000	80→65495 [SYN, ACK]	65495	TCP: 80→65495 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1440 WS=256 SACK_PERM=1
4.338635000	65494→80 [ACK]	80	TCP: 65494→80 [ACK] Seq=1 Ack=1 Win=66240 Len=0
4.338705000	65495→80 [ACK]	80	TCP: 65495→80 [ACK] Seq=1 Ack=1 Win=66240 Len=0
4.339500000	GET /app/view2/		HTTP: GET /app/view2/view2.js HTTP/1.1
4.340074000	GET /app/view5/		HTTP: GET /app/view5/view5.js HTTP/1.1
4.340556000	GET /vendor/ui-		HTTP: GET /vendor/ui-bootstrap-tpls-0.12.0.min.js HTTP/1.1
4.356561000	HTTP/1.1 304 Not Modified		HTTP: HTTP/1.1 304 Not Modified
4.406825000	80→65493 [ACK]	65493	TCP: 80→65493 [ACK] Seq=1 Ack=534 Win=131328 Len=0
4.415728000	HTTP/1.1 404 Not Found		HTTP: HTTP/1.1 404 Not Found (text/html)
4.450623000	80→65494 [ACK]	65494	TCP: 80→65494 [ACK] Seq=1 Ack=535 Win=131328 Len=0

As requested, we made a wireshark capture of the communication between one of the clients (IP 192.168.1.236) and the server (IP 104.45.81.79), ignoring all other routers IPs in between. On the first 3 lines you can see the initial three way handshake:

- 1st - the synchronize, with sequence number 0 sent to the server;
- 2nd - the response from the server, with sequence number 0, acknowledgment number 1 (value 1 because the server already received the 1st request, which is counted with size 1);

3rd - the acknowledgment from the client to the server, with sequence number 1 and acknowledgment number 1.

Persistence strategies

In the JPA server for the initial setup of the server which includes the port and the address we use the Properties file as our persistence strategy for the server startup information. The pros of this strategy include quick setting of the server properties. They can be quickly edited even without having to go to Netbeans, it is enough to use a simple text editor, and when it's saved and the server is restarted these setting will be applied. It also saves us the need to hardcode values into the program. One of the cons is that these properties file is just based on keys and values therefore you cannot put any logic into this file. Also if the program is getting the resources from an external file that could be a security issue in case the attackers would get hold of this file and alter it to their needs since it's not compiled just copied.

We are connecting both the JAVA and Node server to a database to store data as our second persistence strategy. In JPA we use an Oracle database and in the case of the Node server we are using MongoDB. The pros of using a database include a permanent way to store data. This data is not kept in the memory of the program at runtime so that it is not erased when the program finishes. It can be accessed by multiple different programs therefore the data is reusable. Its cons include security issues and connectivity problems if you are trying to read from a DB that is hosted on another server, since you have to go through the Internet. Another issue might be that there are different types of DB so that you might have to use a REST API like we did in the project which takes care of the communication between the databases.