

Sprint Retrospective Document

Date: 13/03/2019

Project acronym: COW

Members:

- İdil Zeynep Alemdar - e2098721@ceng.metu.edu.tr
- Sevim Seda Çokoğlu - e2008662@ceng.metu.edu.tr
- Ozan İncesulu - e2099711@ceng.metu.edu.tr
- Muhammed Kerim - e2001535@ceng.metu.edu.tr
- Volkan Gülen - e2171601@ceng.metu.edu.tr

Supervisor: Dr. Onur Tolga Şehitoğlu.

Sprint 6 summary:

Item ID	Workpackage ID (from the Kick-off doc)	Status	Description
9	WP4	Not started	- Implementing course creation To complete the development of newsgroup and search module, unrelated tasks were not started. This issue is left till related tasks are being developed.
10	WP2	Complete	- Editable table creation and updates Description: We had created an interface for editable components that can be extended and customized, however they have lacked key bindings for editing, and their implementation was missing for some lists that we have. Thus we have created a modular key binding function and used this to create display tables for

			<p>course outline, course instructors, course list and instructor list in general.</p> <p>Problems faced:</p> <p>We had empty strings when pulling the outline from ODTUClass to prepare the demonstration of concept. This has prevented the logic from identifying rows, which were indexed by their values.</p> <p>Solutions found:</p> <p>We have created static week indexes to be the identifiers of rows to prevent multiple cells being updated and we have replaced empty values with (empty) to allow editing to happen on those rows.</p> <p>Conclusion:</p> <p>The table implementation that we had was based on being dynamic identification of columns however this proved to become buggy for real-life cases, thus we have realized that creating static read only fields is important not only DBMS.</p>
13	WP2	Dropped	<p>- Creating admin view for role assignments</p> <p>The instructor has asked for a demonstration of simple CMS methods thus we had to concentrate on allowing instructors to edit courses more than creating a front-end view for role assignments.</p>
15	WP2	In progress	<p>- Create dependency injection strategies on CMS</p> <p>Description:</p> <p>We need to allow extensions to inject different ways of performing actions on different paths on frontend. We have created a way to create these in database and created places on front-end to add these.</p>

			<p>Problems faced:</p> <p>We have realized that creating the strategies inside the DB will create synchronization problems while deleting the components.</p> <p>Solutions found:</p> <p>We have to create the dependency injection configs inside the cluster instead of the DB.</p> <p>Conclusion:</p> <p>We have successfully created front-end injections however we need to create a better mechanism for configuration injection.</p>
3	WP4	In progress	<p>- Create a global search service for different indexable entities</p> <p>Description:</p> <p>Back-end part of the global search is finished. According to our meetings with our supervisor, we decided not to implement global search for nested documents in elasticsearch.</p> <p>Problems faced:</p> <p>If the search pattern is in the nested part of a document, elasticsearch would not respond to that search.</p> <p>Solutions found:</p> <p>Instead of implementing the search for nested documents, we will flatten them.</p> <p>Conclusion:</p>

			Backend part of the global search is now ready and will be made compatible with the front-end.
17	WP5	In progress	<p>- Creating event scheduling</p> <p>Description:</p> <p>Further development and debugging preferences API & further development of remaining scheduling entities</p> <p>Problems faced:</p> <p>Turns out that static entities like time slots and rooms shall not be newly added but we shall give reference to the existing ones. In this case input validation etc should be modified.</p> <p>Solutions found:</p> <p>Before debugging preferences I decided to first implementing CRUD operations of static entities.</p> <p>Conclusion:</p> <p>Due to my illness as well I couldn't spend as much time as I would like to. In the next sprint I plan to make more improvements in this MF.</p>
18	WP8	In progress	<p>- Create grading tool backend API</p> <p>Description:</p> <p>We have created a grading backend API that handles basic crud operations, csv import and export, composite grade generation.</p> <p>Problems faced:</p>

			<p>Being unfamiliar with TypeScript and Node framework derived most of the problems. Course sectioning and generic composite grading with a formula is still uncertain.</p> <p>Actions taken:</p> <p>Most of the time spent building the API was on research and test solutions.</p> <p>Solutions found:</p> <p>Apart from crud operations weighted average of the gradables are implemented.</p> <p>Conclusion:</p> <p>Grading is still incomplete because of the arising factor of course sectioning arising on the last minute and being unable to implement the generic grade generation. However if not needed to divide grading policy of sections the endpoint is almost complete.</p>
19	WP6	Complete	<p>- Create designs for newsgroup frontend:</p> <p>Description:</p> <p>The front-end development of the newsgroup module was addressed in this sprint and is finalized.</p> <p>Problems faced:</p> <p>The system that we are designing that resembles COW of CENG department have multiple newsgroup requirements that need to be addressed. To develop</p>

			<p>the newsgroup features and design, we are required to understand teachers preferences. And reasons behind favoring ODTUClass, Piazza and other systems over COW.</p> <p>Actions taken:</p> <p>A full feature lists have been designed based on the COW newsgroup and the favored features from the other mentioned course management systems.</p> <p>Solutions found:</p> <p>We have actually designed a standard (default) template for newsgroup. This does not eliminate the possibility of having another interface for the newsgroup based on the course type and the teachers prospectives.</p> <p>Conclusion:</p> <ul style="list-style-type: none"> - The design of the newsgroup template is complete but it is still subject to change and modifications based on teachers feedback. - The front-end logic of the newsgroup is yet to be developed.
20	WP6	Was not on initial plan	<p>- Create back-end newsgroup module</p> <p>Description:</p> <p>The very basic part of the newsgroup workpackage is implemented by using python nntplib. Additionally, mongoDB configurations are done for database requirements by using python pymongo.</p> <p>Problems faced:</p> <p>There are not sufficient examples on the Internet about how to make a newsgroup with NNTP.</p>

			<p>Solutions found:</p> <p>Documentations helped a lot for both pymongo and nntplib.</p> <p>Conclusion:</p> <p>It is the very first implementation of this workpackage. It will surely evolve by the next sprint.</p>
--	--	--	---

Sprint 7 plan

Item ID	Workpackage ID (from the Kick-off doc)	Description	Status
15	WP4	- Create dependency injection strategies on CMS	Leftover from Sprint 6
3	WP4	- Create a global search service for different indexable entities	Leftover from Sprint 6
17	WP5	- Creating event scheduling	Leftover from Sprint 6
18	WP8	- Create grading tool backend API	Leftover from Sprint 6
20	WP6	- Create back-end newsgroup module	Leftover from Sprint 6
21	WP5	- Development of scheduler front-end	New

Overall progress

	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Sprint 7	Sprint 8	Sprint 9
MF1	5%	60%	80%	85%	85%	85%	90%		
MF2	0%	30%	50%	60%	60%	75%	75%		
MF3	10%	35%	40%	55%	55%	60%	70%		
MF4	0%	0%	0%	0%	0%	5%	20%		
MF5	5%	5%	5%	5%	5%	10%	25%		
MF6	0%	0%	0%	0%	0%	0%	0%		
MF7	10%	10%	20%	45%	45%	55%	60%		
MF8	0%	10%	10%	10%	10%	20%	35%		
MF9	20%	30%	30%	40%	40%	50%	50%		
MF10	0%	0%	0%	15%	15%	20%	35%		
MF11	0%	10%	50%	70%	70%	70%	75%		
MF12	0%	0%	0%	0%	0%	0%	0%		
MF13	0%	0%	0%	0%	0%	0%	0%		
MF14	0%	0%	0%	0%	0%	0%	0%		