Sprint Retrospective Document

Date: 03/04/2019

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 7 summary:

Item ID	Workpackage ID (from the Kick-off doc)	Status	Description
15	WP4	Complete	Create dependency injection strategies on CMS Description: We had a way for injecting configurations on the system, however the requirements of the project that we had an access controlled and template rendering (ex. adding courseID automatically to configs with {cid}) way to inject configurations on the systems, furthermore the available system was not tested Problems faced: We realized creating template tags and access control (which are service scoped) on the injections (which are infrastructure scoped) would create coupling between different scopes, thus would prevent scalability.

			Solutions found: We have made the choice to move whatever customization requirement exist, to the binding services. This approach would require each service to handle their own config, however considering the extended capabilities introduced to the extension developers, we believe that this is the correct way to go. Conclusion: We have only applied bug fixes for the bugs when the component were being tested against the core endpoint and change nothing else. However, we have changed the core endpoint in order to demonstrate the capabilities that the developers will benefit from a simple API
3	WP4	Dropped	Create a global search service for different indexable entities Description: The initial plan to use elastic search was failed because resources requşred to provision an elastic search cluster appears to be too high. Therefore, in the future we will create text based search using mongodb however, upon discussing with the instructor, we have concluded that this work can be tackled in the future.
17	WP5	In progress	Creating event scheduling Description: Started implementing collision checking features and testing thereof. Problems faced: There were some bugs in the library code mainly resulting from Express. js, which resulted in errors while testing. Besides, I didn't know some library features, as a result I was thinking of complex ways to solve some issues.

			Solutions found: I took a further look into the features provided by the library, which made my development process easier. Also, as I am getting better at working with TS and Mongoose, I can search for errors and debug them better. Conclusion: Two main conflict checks are implemented, two more to go. Semester scheduling is about to be completed.
18	WP8	Complete	Create grading tool backend API Description: Grading backend API that handles basic crud operations, csv import and export, composite grade generation is completed. Problem faced: Evaluation expression without using reflection was struggling. Solutions found: At first gradables in the formula were kept as an object(field-value pair) {abbreviation:grade} that needed further tricks with reflection. However, expression evaluation tool had substitude functionality, therefore no need to use reflection. Conclusion: There may need little tweaks to make backend compatible with the frontend.
20	WP6	In progress	Create back-end newsgroup module Description: The newsgroup back-end module is actually finished but it should be tested. NNTP server is now reachable, and MongoDB is following it.

			Problems faced: Finding a free, public newsgroup server was a bit difficult. Solutions found: I used the newsgroup server which is used in the python nntplib documentation. Conclusion: The module is almost finished except that it needs to be tested.
21	WP5	Complete	Development of scheduler front-end Description: The interface of the scheduler was analyzed first and divided into three main views: the scheduler notifier view, teacher preferences view, one shot scheduler view Problems faced: Defining the components and their descriptions on the interfaces is treaky since the rooms and the times shall be seen at once for every day. In addition, teacher preferences design which has depended on drop down input selection was not approved by the instructor since it creates an overhead and bad user experience. Solutions found: The view of the events has been built according to the weekly distribution of classes where classes have been categorized and assigned colors. For teacher preferences table the supervisor has suggested using color inputs which can be selected if a user clock on some table cell. Conclusion: The scheduler frontend is almost ready but it has to

			include notification mechanism to show collisions and the teacher preferences table shall include the suggested solution.

Sprint 8 plan

Item ID	Workpackage ID (from the Kick-off doc)	Description	Status
17	WP5	Creating event scheduling	Leftover from Sprint 7
20	WP6	Create back-end newsgroup module	Leftover from Sprint 7
22	WP5	Implement a notifier interface for scheduler	New
23	WP5	Develop a colorful input preferences interface for teachers	New
26	WP6	Create front-end integration for scheduling	New
27	WP2	Develop notification end-point	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%	90%	
MF2	0%	30%	50%	60%	60%	75%	75%	75%	
MF3	10%	35%	40%	55%	55%	60%	70%	75%	
MF4	0%	0%	0%	0%	0%	5%	20%	35%	
MF5	5%	5%	5%	5%	5%	10%	25%	40%	
MF6	0%	0%	0%	0%	0%	0%	0%	0%	
MF7	10%	10%	20%	45%	45%	55%	60%	70%	
MF8	0%	10%	10%	10%	10%	20%	35%	35%	
MF9	20%	30%	30%	40%	40%	50%	50%	55%	
MF10	0%	0%	0%	15%	15%	20%	35%	55%	
MF11	0%	10%	50%	70%	70%	70%	75%	75%	
MF12	0%	0%	0%	0%	0%	0%	0%	0%	
MF13	0%	0%	0%	0%	0%	0%	0%	0%	
MF14	0%	0%	0%	0%	0%	0%	0%	0%	