Drivers Evaluation

We set clear project specification, and had regular meetings on weekends since we all had 54-60 units and shared some time constraints. We all expected to experience web development as full-stack engineers. Therefore, on the meeting we would divide features into small tasks that were not so dependent on each other and decide who would complete which task according to personal perference and strengths. So there was no misunderstanding when implementing.

We didn’t set the explicit time limit for each task. But once we found the task was difficult to accomplish in a resonable time period, we would try to settle it together or switch role. And sometimes we would talked about using another approach.

All members were present on every meeting. Everyone was punctual, and we did not waste time waiting. Our meetings were very efficient.

Between sprints, we modified features according to audience’s feedback.

Contributions:

Guangyao Xie <guangyax>:
Lauched and scratched most reports and presentation slides.
Reseached on the API for course.
Wrote and debug date/time picker component for submiting time to server.
Implemented front-end for deadline stream.
Implemented add due ajax using plugins for date time selection.
Designed and implemented Admin site with Django user group and permission.
Wrote unit test suites with 62% line coverage.
Implemented Jenkins Continuous Integration server with build result notification

Pimeng Fu <pimengf>:
Drew the wire-frames for prototype of duelink.
Implemented calendar features at the home page.
Implemented task features with all Ajax .
Implemented team features.
Build and deployed DueLink docker containers on AWS EC2.
Deployed Postsql and Postfix docker container using existing docker image.
Rearched and adopt the plugins for auto complete selection (courses, friends)
Purged and reformated the course data for duelink database.

Yilong Chang <yilongc>:
Designed and scratched DueLinks’s model at proposal stage
Researched on the django-allauth and tried to implement email verification and facebook login

Set up POSTFIX email server backend and used it to implement password reset
Implemented ‘friend recommendation’ feature
Implemented ‘add task and delete task’ using Ajax
Implemented ‘profile’
Lay the foundation for basic front-end CSS style and adjusted CSS on each page
Used the cmu-course-api to get all courses of CMU and load data into postgres