DOSE Project report (Group 3)

The backend is developed using the Eiffel Web Framework.
Frontend team provided the original REST API, and then was adjusted by the backned team, that is documented in REST_API.pdf file. You can find this file in the doc folder.
Backend consists of four main parts:
1) APPLICATION class
2) Controllers
3) DB models
4) The DB
1) APPLICATION class creates controllers, DB models and maps URLs to the appropriate controllers. It's simple. Thanks to Eiffel Web Framework.
2) Most controllers functionality is implemented in the CONTROLLER_BASE class. This class is inherited by all controllers. Some of them block some functionality, some provides new custom functionality. Some DB model is binded to each controller.
3) Common database interaction functionality is imlemented in the DB_MAPPER_BASE class that is the base class for all DB models. This class stores a table name and provide typical functionality such as get, update and delete. If it's needed some special functionality (such as in login), it's implemented in the appropriate inherited class.
4) The DB model have tables that reflects the project requirements. There are projects, users, sprints, stories, tasks and comments tables and the relations between them in the database.
In our project we decided to have one controller and one DB model for each table in the backend database. It makes the code clear and simple.

We discussed our REST API with the frontend team and implemented what they need.
The problem with the backend is that we hadn't got enough time to add adequate error-handing. Thus if someting wrong happens backend may raise an exception and nobody will catch it. That's a very bad behaviour but we are going to correct it just after the exam.

Unfotunately the integration was no completed instead the frontend team was waiting for implementation of the whole API functionality. Backend team was implemented several controllers very early to give frontend team a possibilty to test integration on them. Several weeks before the deadline backend team discussed the API with the frontend team and agrreed on them. Several days after final API documentation was commited to the SVN. By the deadline full API implementation was commited to the SVN.

The frontend was developed using HTML, and Javascript with AngularJS.

It has some requirements functioning on its own without backend, because integration was not possible due to most of backend controllers being commited on the last moment. The requirements implemented are FR1, FR2, FR5, and the one not implemented are FR3, FR4, and all non-functional requirements. Details are showed in the demo.

Note: On the video demo we are only showing the independent frontend.