Team 41
Bi-Weekly Report
UCLH Peach & Microsoft : Tools

German Mikulski, Nancy Amelia, Yuan Pan

10/03/2017 - 24/03/2017

Overview

Progress summary

Over the past two weeks, we have been working on main components of the final system. At this point of time, we have completed the core components for the User Interface which includes the sidebar, header-bar, and an upload form functionality of the form builder. Furthermore, we are currently working on parsing a JSON object in order to remove meta-data and other dependencies that are not required for form generation. We have managed to complete tree-rendering process, although detailed features such as specification of the data types need to be added in order to produce a functionally complete component.

Successes

- Completed the majority of the user interface components of our system
- Managed to successfully fetch data from EHRScape and initiate parsing processes
- Completed the tree rendering process of EHRScape backend

Integrating the front-end and back-end might be problematic. We are anticipating this by performing unit tests on each of our component.

Issues

- Implementing the customisation part of our form builder is tricky due to the restrictions of data format.
- Manipulating the JSON object might needs the use of RegEx, which increases the complexity of the parsing method.
- Integrating certain parts of front-end and back-end might prove problematic in the future.

Meetings

17/03/2017

During this meeting, we showcased our work to the project director to receive constructive feedback regarding our project. Earlier in the week, we prepared a presentation which includes our project brief, client details and a comparison of MosCoW analysis and the progress of our project. We managed to complete most of the basic functionalities of our project, however we are still working on completing additional features and integrating our work.

24/03/2017

In the meeting, we discussed the approach of integrating drag-and-drop components in our system. We decided to first start with a simple up and down button for the customisation part of our form and further integrate the drag-and-drop functionalities. Furthermore, we discussed with our academic leader additional aspects of template parsing. He suggested the use of RegEx in order to simplify our work instead of using a "for-loop" approach to iterate through the JSON object. We are carefully considering his advices to improve on the functionality of our form builder.

Future plans

- Complete our project website which includes user manuals of the system, testing stages, system diagrams and architecture.
- Implement additional functionality of our form builder such as the customisation part
- Integrate the frontend and backend components of the system

Individual reviews

German

My core task during this period of time is to simplify the JSON object that I previously managed to receive from EHRScape in order to make the process of markup generation simpler for the rest of my team. Due to the complex structure of openEHR templates the task is not complete yet. At the moment, I still have to make child JSON object leaner, despite main body of the object already being prepared for markup generation. In addition, I tried out several form generation libraries that the rest of our team asked me to use and lead the presentation of our progress to a TA on the last Friday session.

Nancy

Over the past two weeks, I have been developing the UI components for the project. This includes the upload button and functions to validate the type of the files uploaded. Furthermore, I experimented on form generation using React form builder libraries. I also have a look at the React Drag and Drop functionalities in order to enhance user experience in using our form builder. Moreover, I experimented on parsing the JSON object to equivalent arrays and render it using a JSON React-Schema libraries.

Angela

During the past two week, Ive been doing the rendering of the form. The input of the form generator should be a tree structure with JSON array containing children node and properties. So far I have been experimenting with only title property with a toggle functionality. Next step is to implement the rendering of different controls according to the property of the children node.