

Stage B Deliverable

- Members: Ryan Guard, Elizabeth Gieske, Ronnie Hoover, Liam Tiemon, Christopher Groppe
- Date: 04/19/2018

Schedule

Schedule Overview

Work on stage B began on March 29th. This was our first meeting for deciding our feature and modularization of it. After this we had several meetings on how to split up our work and by April 7th we had designed our feature documentation and clearly established each person's objectives. From there till April 13th we met several times to provide us a place to do pair programming which ensured each person was informed in how our implementations were shaping up and provided us opportunities to further bring clarity to our initial design. April 16th onward we finalized our B objectives and began the forming for stage A and uploading the project to PiPy. Meetings had varying attendance but most attended the sessions and those who didn't kept in contact online.

Schedule and milestones (ver 1.0)

- 03/29/2018: Select a feature for the group to implement
- 04/02/2018: Assign tasks
- 04/07/2018: Finish architecture and initial design of project.
- 04/13/2018: Code of modules finished.
- 04/19/2018: All of Stage B/A is finished.

Meetings

- 1st meeting: March 29, 2018
 - Everyone Attended
 - Decided on Feature to cover
- 2nd meeting: April 2, 2018
 - Everyone Attended
 - Online figuring out how we would break up our feature (storing content history for the website and allowing for reverts and recovery of pages).
 - Delegated tasks out for the project and decided to meet on the 7th to start working together to allow for pair programming if necessary and maintain a consistent vision for how our feature would look.
- 3rd meeting: April 5, 2018
 - Everyone Attended
 - Online discussion on how we were planning on lining up our development, architecture, schedules, etc.
- 4th meeting: April 7, 2018
 - Ryan Guard, Elizabeth Gieske, Christopher Groppe attended
 - Finished feature architecture and how we would communicate between the web application and the database. Finished feature documentations. Further established scope of our individual modules of the feature.
- 5th meeting: April 9, 2018
 - Ryan Guard, Elizabeth Gieske, Liam Tiemon, Christopher Groppe(online) attended
 - Worked on Stage B in pair programming. Started scoping out what we'd need to do for Stage A of the assignment.
- 6th meeting: April 13, 2018
 - Ryan Guard, Elizabeth Gieske, Liam Tiemon, Christopher Groppe(online) attended
 - Completed our developments of our modules in stage B. If modules weren't done by end of meeting it was agreed to have them done by monday where we'd put it all together and test end-to-end.
 - Further documentation made.
- 7th meeting: April 16, 2018
 - Everyone attended
 - Integrated modules. Finalized testing and documentation for stage B.
 - Start working towards completing stage A (hopefully finished).
- 8th meeting: April 17, 2018
 - Completed Stage A requirements
 - Completed all testing
 - Finished documentation
 - Group will do final checks and submissions tomorrow
- Retrospective: April 17, 2018

- Everyone Attended
- Final review of Deliverables
- Assessment on current sprint.

Rules collection

We collected 10 SE rules in stage.

- Collaboratively came up with them. All get credit.

Results

Results Overview

Features:

- All features(Document archive system, Document retrieval system, UI, Database, and testing suite) were completed on time all members get credit for their work and collaborating to help each other accomplish their goals.
- Documents on edit or save will be stored into the postgresql database which stores info about the document and who edited/created it.
- Users can look at all pages made and choose to restore or recreate pages based on selection of it.
- System is tested end to end guaranteeing the quality of the feature.
- Database is hosted in the cloud allowing for it to be remotely accessed by the web service be it local host or on a server somewhere.
- Documentation has been created explaining the business reasons and intricacies of each module
- Elizabeth Gieske created the UI for the restoration feature using Flask, Jinja2, and python
- Ryan Guard created the Archive feature, implemented it so that it stored on creation or edit of a page, and developed queries for data storage, retrieval for UI, and overall KT in how to use the psycopg2 library to interface with the database.
- Ryan Guard created the standalone version adapting the entire projects code to be compatible as an argparser and reformatting code to be able to be built as a module and uploaded to PiPY
- Ryan Guard created architecture diagram.
- Liam Tiemon created the retrieval feature, worked with Elizabeth to implement it with the UI.
- Christopher Groppe created postgresql database on AWS using a linux box to allow for the database to be better accessed and hosted on a robust server that we wouldn't have to worry about going down.
- Ronnie Hoover created a test suite to ensure the quality of our code.