

Personal Page Oscar

Jan 10 (3h):

Read all of the provided documents for the EPL project and made notes regarding questions or things that were not clear to me.

Jan 11 (3h):

After having met everyone at the team I began exploring if Django would fit our needs. I created a sample app to test it out. I now have a Branch model which gives us a free management page to modify DB data. Also have unit tests and feature tests using Selenium.

Jan 17(3h):

Had our client meeting and the target infrastructure will allow us to deploy a Django backend. Added field validation to the previously created model.

Jan 19 (3h):

We now have a git repo so I merged the code that I have for the back-end into the repository. Also wrote up wiki pages documenting the steps required to setup a development environment. Also created my first milestone which is to finish a POC for the back-end by Friday 25th.

Jan 20 (2h):

Added tastypie to the back-end in order to provide a REST interface into the back-end. This will allow us to decouple the front-end from the back-end. Also wrote documentation detailing at a high level how to accessed the exposed data.

Jan 25 (1h):

Read up on user profiles and Oauth2

Jan 29 (2h):

Did most of the work for the Stories model.

Jan 30 (1h):

Braeden proposed we manage the branches in our repo better. I had pushed some stuff to master while he was doing that so I synced the branches and fixed the merge conflicts. Sent a quick email to everyone explaining a simple workflow with the branches.

Jan 31 (5h):

- Created wiki page outlining git workflow
- Made a couple of scripts to automate DB management to make it easier on everyone
- Created a git-hook to automatically update DB after any merge
- Finished most of the story model
- Exposed story model through front end JSON API
- Wrote up quick summary on how to create DB entries through POST requests
- Added testing module from tastypie 0.9.12-alpha version (We use 0.9.11-beta) to the repository to allow for easier testing of the API.
- Created sample DB with a couple of branches and a story
- Did more investigation on file management for the backend. I now know how the files will be served, but I'm still wondering how the upload will work.
- Researched security concerns with allowing file-uploads. Several security risks become apparent. We need to validate the files that are uploaded.
- Researched how Django handles Cross Site Request Forgery protection (CSRF).

Feb 02 (10h):

- Had team meeting to work together for the day
- Setup an IRC server, couldn't get github hook services as only Noorez has admin and he left early
- Finished initial backend setup for file upload

Feb 03 (1h):

- Finished setting up the IRC channel and learned how the modes work so that the github hooks would work
- Figured out why the front end code by Tim to upload files wasn't working.

Feb 04 (1h):

- Setup the github service hooks with Noorez and made sure notifications were sent to IRC server.
- Experimented with help text in the models to provide better error messages.

Feb 09 (7h):

- Project Plan and Requirements Document Work
- Created script to extract the wiki section from the github wiki and generate html ready to be printed to pdf as we could not find a markdown compiler that supports github table syntax.

Feb 16 (8h):

- Import of given branch and map csv data
- Generate fixtures to allow quick loading of the data for testing
- Updated branch model to allow currently opened branches
- Created map model
- Investigated file mime types for file uploads
- Investigated different approaches for exposing filtered data through tastypie resources

Feb 17 (8h):

- Finished file type validation through Mime types
- Improved response error messages of the backend API
- Added django-tagging app to track keywords on stories
- Adapted django-tagging to allow filtering through tastypie
- Added filtering for branches and stories through tastypie
- Updated Timemap API wiki page with summary of new features

Feb 18 (6h):

- Investigated different options for handling user sessions, logins, and access control both for Django and tastypie. (Spent a significant amount of time doing this to make sure our implementation is safe and robust)
- Implemented user session requirements for story creation

Feb 19 (8h):

- Added login and logout views
- Found bug with uploadStory page where media files were not being uploaded. Tim fixed it
- Fixed tests for story creation and added test to make sure anon users can't create stories
- Reviewed Noorez's pull request, commented on it, and later merged it.
- Story creations through the API now return the new object details encoded

- Changed TagManager to use keywords instead of tags as the name for keywords
- Added users to the story model. When a story is created the logged in user is set as the creator.
- Updated test fixtures and removed old migrations as we do not need to rollback that far back, they were helpful however when keeping the fixtures up to date.

Feb20 (8h):

- Fixed AES import from Noorez' change
- Wrote docs on some methods that were not very clear, the probably need to be refactored in some cases
- Fixed a small bug when stories with no media files attached were deleted
- Filtered stories so that only publicly approved stories are exposed through the JSON api
- Restrict file types on the front end to the allowed ones
- Added content_type to stories
- Added proper file validation using the file's mimetype

Feb 21 (8h):

- Add content_type filtering on stories
- Improved media file saved locations
- Updated fixtures to remove migration data
- Finished user profiles that Noorez started

March 7 (2h):

- Worked on finalizing the search logic
- More on file management for stories

March 14 (3h):

- Fixed duplicate stories on search
- Prepared for demo with Braeden
- Debugged account logins with Braeden

March 16 (9h):

- Added preferences to timemap for global settings
- Stories are cleaned when media type is not valid
- Uploaded files are assigned unique names
- Added Map resource for tastypie
- Created featured stories
- Finished all tickets assigned to me! o.O

March 17 (2h):

- Added more preferences
- Added indexes to timemap model
- Refactored error messages (Not pushing until after demo to avoid breaking things)

March 29 (8h):

- Closing all the things
- Updated timemap API to use anonymous flag for uploaded stories
- Changed API to properly expose user names
- Made error messages more consistent

- Added more preferences

March 30 (13h):

- Expanded unit tests
- Timemap and HYQ code polish

April 1 (11h):

- Added ability to have global stories
- Repo cleanup
- Added branch types (Dickensrolly >:())
- Updated sidebar to say Featured instead of Features
- Removed migrations

April 2 (9h):

- Deployed full production environment