



Sluggit

A Computer Science blog and resource

Team Members: Jasmine Dupree, Maximilian Brain, Tarum Fraz, Diego Garcia, Hope Ashcraft(SM), Chris Gradwohl(PO)

Document Name: Sprint 2 Plan

Sprint 2 Completion Date: 05/15/2017

Revision Number: 2.2

Revision Date: 05/05/17

Goal

1. Finish building out backend for posts and for Google, Facebook, GitHub Auth.
2. Finish the Sluggit minimum viable product.

Task Listing

Sprint 2 (story points)

- A. As a Sluggit user, I want to signup and login, so that I can post content to Sluggit.
(40 story points)
 - a. Create signup form/ UI
 - b. create angular signup service to connect data to the backend
 - c. Finish passport auth strategy for google, facebook, github

- B. As a Sluggit user, I want to be able to post a blog to share my ideas to the Sluggit community. (40 story points)
 - a. Finish the blog post angular component and UI template
 - i. Edit, text box, title, body, picture etc.
 - b. Add a Post model to the mogodb with mongoose
 - i. Also need to add CRUD api methods and DB methods to accomplish this.
 - ii. Test this with postman
 - c. Create a blog post angular service
 - i. Store the post to the db, based on the author
 - d. Add a hashed url for each blog that is posted to the db
 - i. Or figure out a good way to ID the blog posts
 - e. Update the home feed with new post components from other users
 - i. Want to display posts based on timestamp
 - ii. Note that the home public feed requires no styling it is just a container of blog post objects

This is probably for sprint 3, but maybe we can get there!

C. As a Sluggit user I want to be able to be able to 'like' blogs so that I can shape the landscape of the sluggit community. (30 story points).

- a. Add 'like' functionality to the blog object
- b. Update the Post model to store likes
- c. Update the DB methods and API to update the likes field in the Post model

Team Roles

Jasmine Dupree: Developer

Max Brain: Developer

Tarum Fraz: Developer

Diego Garcia: Developer

Hope Ashcraft: Developer, Scrum Master

Chris Gradwohl: Product Owner, Developer

Initial Task Assignment

Jasmine Dupree:

- A. Research angular services
- A. Add service to store blog posts
- B. Add 'like' service on blog object
- C. Sorting posts by upvotes, downvotes, etc...
- D. Get backend database to hold comments and votes

Max Brain:

- A. Set up external user authentication
 - a. Create a Google Passport auth stragetgy
 - b. Create a Facebook passport auth strategy
 - c. Create a Github passport auth strategy
 - d. Add all strategy endpoints to the login view via buttons
 - e. Get Project working across all developers

Tarum Fraz:

- A. Set up external user authentication:
 - f. Create a Google Passport auth strategy
 - g. Create a Facebook passport auth strategy
 - h. Create a Github passport auth strategy
 - i. Add all strategy endpoints to the login view via buttons
 - j. Get Project working across all developers

Diego Garcia:

- A. Get backend to hold posts:
 - a. Add a Post object to the mongo model
 - b. Add DB methods for CRUD operations on the Posts
 - c. Add DB methods for CRUD operations on the Posts

Hope Ashcraft:

- A. Show all users posts on the public feed/ Home page
 - a. Create blog post component form
- B. Add ability for user to add, delete, and edit post via angular services
- C. Write Mongo queries/methods for CRUD operations on posts
- D. Learn http

Chris Gradwohl:

- A. J. MAKE SURE EVERYBODY IS ON THE SAME PAGE AND HAS THEIR QUESTIONS ANSWERED!
- B. Finnish native login
 - a. Create and add login form
 - b. Confirm everything is working
- C. B. Show all users posts on the public feed/ Home page
 - a. Create blog post component form
- D. Get backend to hold posts:
 - a. Add a Post object to the mongo model
 - b. Add DB methods for CRUD operations on the Posts
 - c. Add DB methods for CRUD operations on the Posts
- E. Finish the blog post angular component and UI template
 - i. Edit, text box, title, body, picture etc.
- F. Add a Post model to the mogodb with mongoose
 - i. Also need to add CRUD api methods and DB methods to accomplish this.
 - ii. Test this with postman
- G. Create a blog post angular service
 - i. Store the post to the db, based on the author
- H. Add a hashed url for each blog that is posted to the db
 - i. Or figure out a good way to ID the blog posts
- I. Update the home feed with new post components from other users
 - i. Want to display posts based on timestamp
 - ii. Note that the home public feed requires no styling it is just a container of blog post objects

Initial Burnup Chart

[Trello Burndown](#)

Initial Scrum board

[Sluggit Scrum Board](#)

Scrum Times

Wednesday: 1:00pm - 2:00pm

Thursday: 12:00pm - 2:00pm

Friday: 1:00pm - 2:00pm

TA: Kavya Jha

Meetings: Wednesday at 3:40pm