Sprint Planning Document

for the

Hotspotter Bug Prediction Software

CS 425 / CS 499 Senior Project

by

Nathan Reinhardt

Spencer Smith

Dylan Williams

of

Team HotSpotter

Sprint Planning Document

Revision 1.0

As Of: 29 September 2015

Change Log:

Revision	**Change Note(s)**
1.0	• Initial release

Goal: Allow to log in.

The client wants to log in to access account features

- Code the database user table (2 p-h)
- Code the user domain object (1 p-h)
- Code user security options (4 p-h)
- Code login UI (2 p-h)
- Code test cases (1 p-h)

Goal: Create Account

The client can create an account for other users to view the hotspotter results

- Code admin server-side API (2 p-h)
- Code admin panel UI (2 p-h)
- Code test cases (1 p-h)

Goal: Add Repository

The client adds a new repository to the database via web interface

- Code the database table to store repo info (2 p-h)
- Code repo domain object (1 p-h)
- Code git services (8 p-h)
- Code repo entry UI (3 p-h)
- Code test cases (1 p-h)

Goal: Manage Repository

The client can update/remove a repository from the database and change the automatic sync timeframe

- Code repo admin server API (5 p-h)
- Code repo admin UI (4 p-h)
- Code test cases (1 p-h)

Goal: System Sync Repository

The system automatically retrieves and update the repo in database based off the current git repo

- Code batch job (10 p-h)
- Code server API to work with batch job (5 p-h)
- Code front end UI to manage batch jobs (5 p-h)
- Code test cases (2 p-h)

Goal: System Retrieve Repo

The System clones a copy of the git repo on server

- Code batch job to clone repo (5 p-h)
- Code server service to quality check repo (5 p-h)
- Code test cases (2 p-h)

Goal: System analyze repo

The system will store the repo into the database for persistent history

- Code database tables to hold specific aspect (3 p-h)
- Code batch job to massage data and store in database (5 p-h)

Goal: View Hotspot results

The client can filter the results based off of different criteria

- Code server side API to retrieve data from database (15 p-h)
- Code client side to call for the data (15 p-h)
- Code UI that will show the repository in a tree format (20 p-h)
- Code searchable facets (10 p-h)
- Code test cases (5 p-h)

Goal: Save Result History

The system automatically saves certain user defined results in the database

- Code server side API save functions (10 p-h)
- Code test cases (2 p-h)

Goal: View previous results and history

The client can view previous results from specific repositories

- Code server side history API (5 p-h)
- Code test cases (1 p-h)

Goal: Save Result Snapshot

The client can permanently save results locally

- Code sever side API to export results (10 p-h)
- Code UI for saving export results (5 p-h)
- Code test cases (3 p-h)

Goal: Export Result

The client can download the text/visual results and save them locally

- Code saved results to database (5 p-h)
- Code server api to export into excel format (10 p-h)
- Code save UI to search saved for saved results (5 p-h)
- Code test cases (2 p-h)