Sprint 2 Plan

Product name: Rootify
Team name: Rootify

Sprint Completion: 11/5/17

Revision number: 0
Revision date: 10/25/17

High-level Goal: Show related tracks based on a Spotify user's top tracks; Selection of a track or artist will populate the 'Details' tab with associated information.

Tasks:

User story 1: "As a developer, I want to be comfortable with JavaScript, d3.js, jQuery, and the Spotify API so that I can be skilled for later development." (**2**; Continued from Sprint 1)

- Continue learning node.js, JavaScript, D3.js and jQuery (two members still need to complete this from last sprint): **5 hours**

Total: 5 hours

User acceptance: N/A

User story 2: "As a Spotify user, I want to see related tracks based on what I listen to so that I can discover new tracks." (**2**)

- Fetch related tracks using the Spotify Web API - **0.5 hours**
- Store results in a children array of the selected node (which represents a track) with associated track information (and updating the tree) - **1 hour**
- Add "up and down" triangles to expand or collapse related tracks (and likewise for related artists) - **2 hours**

Total: 3.5 hours

User acceptance criteria:

- Must have an icon for each track and artist to click to find related related tracks or artists
- Must be able to see the related artists or tracks, and must see the connection
- Most related artists should go from left to right; For tracks, the most popular from left to right

---

User story 3: “As a Spotify user, I want the ability to select an artist and preview their top songs or preview a single track.” (**3**)

- Handling selection of the artist or track in D3.js and calling function to load its associated data - **1 hour**
- Displaying an artist’s top tracks (up to 5, if possible) using Spotify embedded HTML - **1 hour**
- Displaying a track using Spotify embedded HTML - **0.5 hours**

Total: 2.5 hours

User acceptance criteria:

- Must be able to select an artist or track easily
- Must be able to preview or play a Spotify track, either by pressing a play button or playing it through my local and loaded Spotify application

---

User story 4: “As a Spotify user, I want to be able to view artist details (associated genres and popularity) or track details (danceability, energy, popularity, valence, key, and major/minor) so I can learn more about my taste in music and specific artists and tracks.” (**8**)

- (See above—the first task of user story 3)
- Add header image of artist or track with text overlayed and styled properly - **1 hour**
- Implement the ‘Popularity’ bar for both artist and track - **1.5 hours**
- Implement the ‘Danceability’ progress bar (from 0 to 100) using a DIV and overlayer text and style it - **0.5 hours**
- Implement the ‘Energy’ progress bar - **0.5 hours**
- Implement the ‘Valence’ bar - **0.5 hours**
- “ “ Key (tonic key) - **0.5 hours**
- “ “ Major or minor - **0.5 hours**
- Associated genres for artist - **0.5 hours**

Total: 5.5 hours

User acceptance criteria:

- Must able to see a header image of a track’s album art or an artist background.
- Should be able to see the name of the track (and its artist) or the artist
- Should see popularity, danceability, energy, valence (happiness), key of song, and major or minor **for a track.**
- Associated genres and popularity bar **for artist**

Total hours: 16.5 hours

Team roles:

Tristan Iverson:	**Product Owner**, Developer
Jonathan Ortiz:	Developer
Kristine Nguyen:	Developer
Chloe Jiang:	**ScrumMaster**, Developer
Brian Tran:	Developer

Initial task assignment:

Tristan Iverson: **User story 2** - Selection of a track or artist, Storing results in children array with associated information

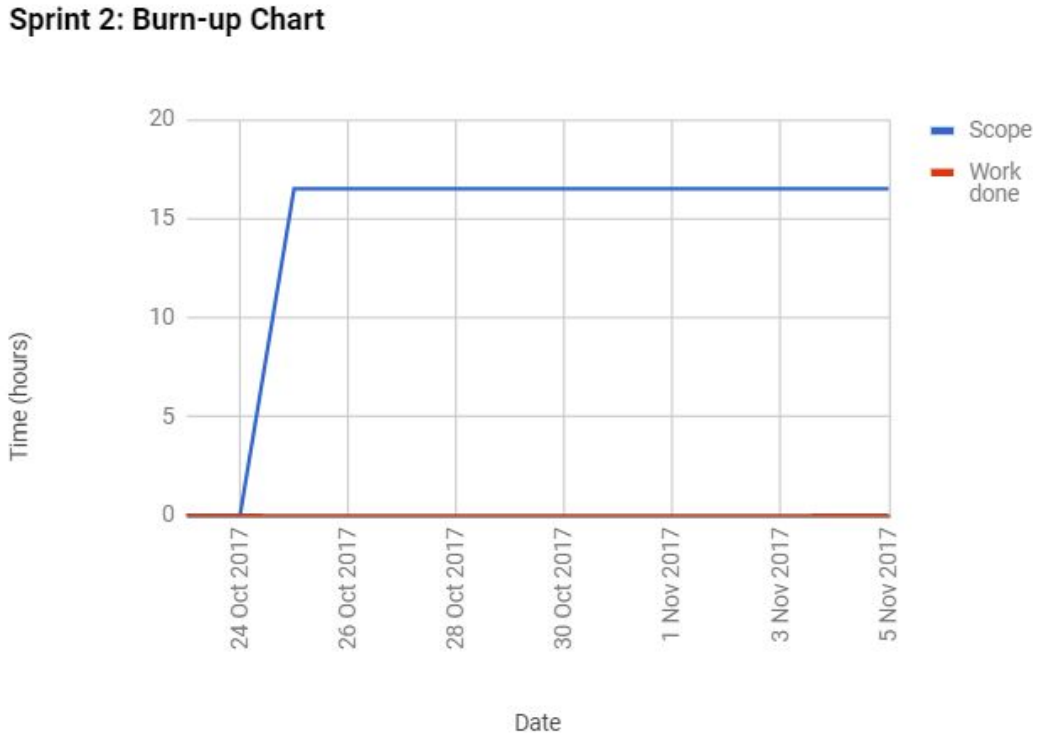
Jonathan Ortiz: **User story 1**. **User story 4** - Creating the bars in HTML and setting them up for styling; **User story 2** - Triangle buttons to collapse and uncollapse subtrees

Kristine Nguyen: **User story 1** - Re-familiarizing with node.js, JS, D3.js, jQuery; **User story 4** - Add header image or artist or track album cover with overlayed text on top of gradient, Styling the bars

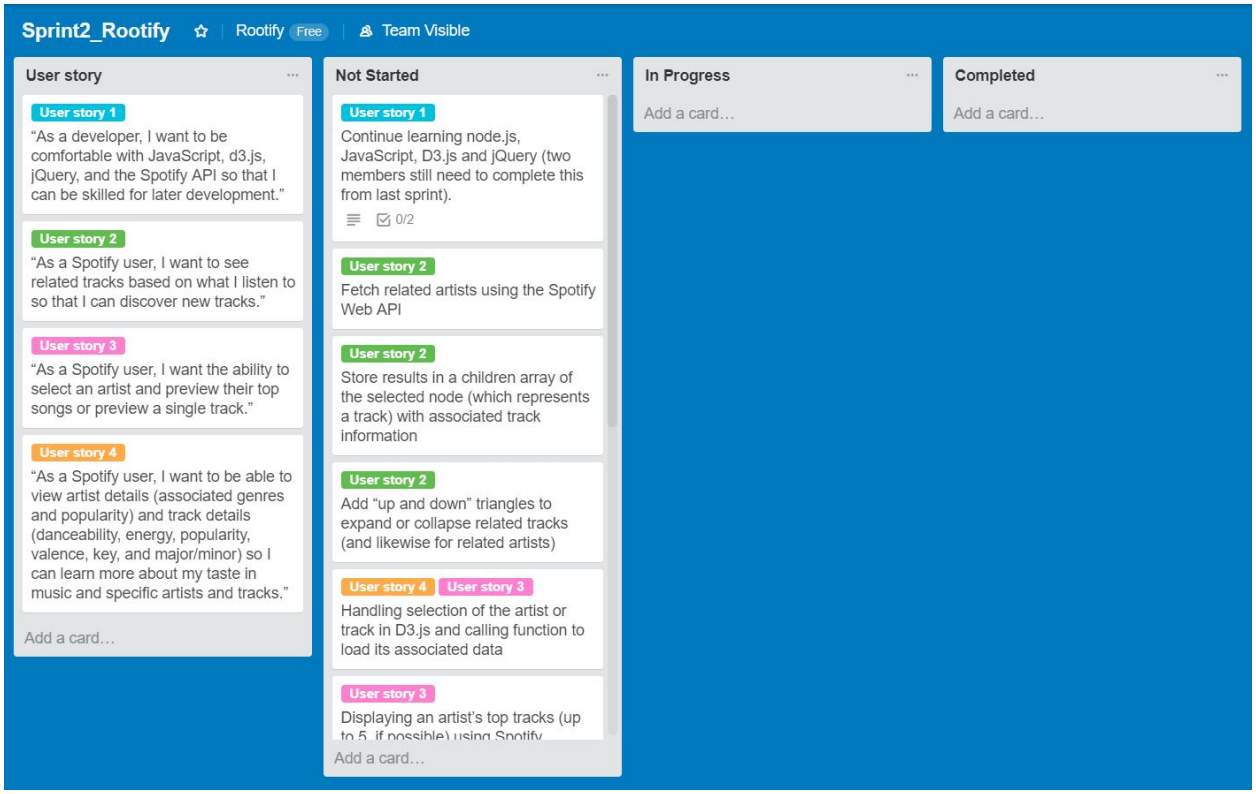
Chloe Jiang: **User story 1. User story 3** - Handling the selection of a track and artist properly

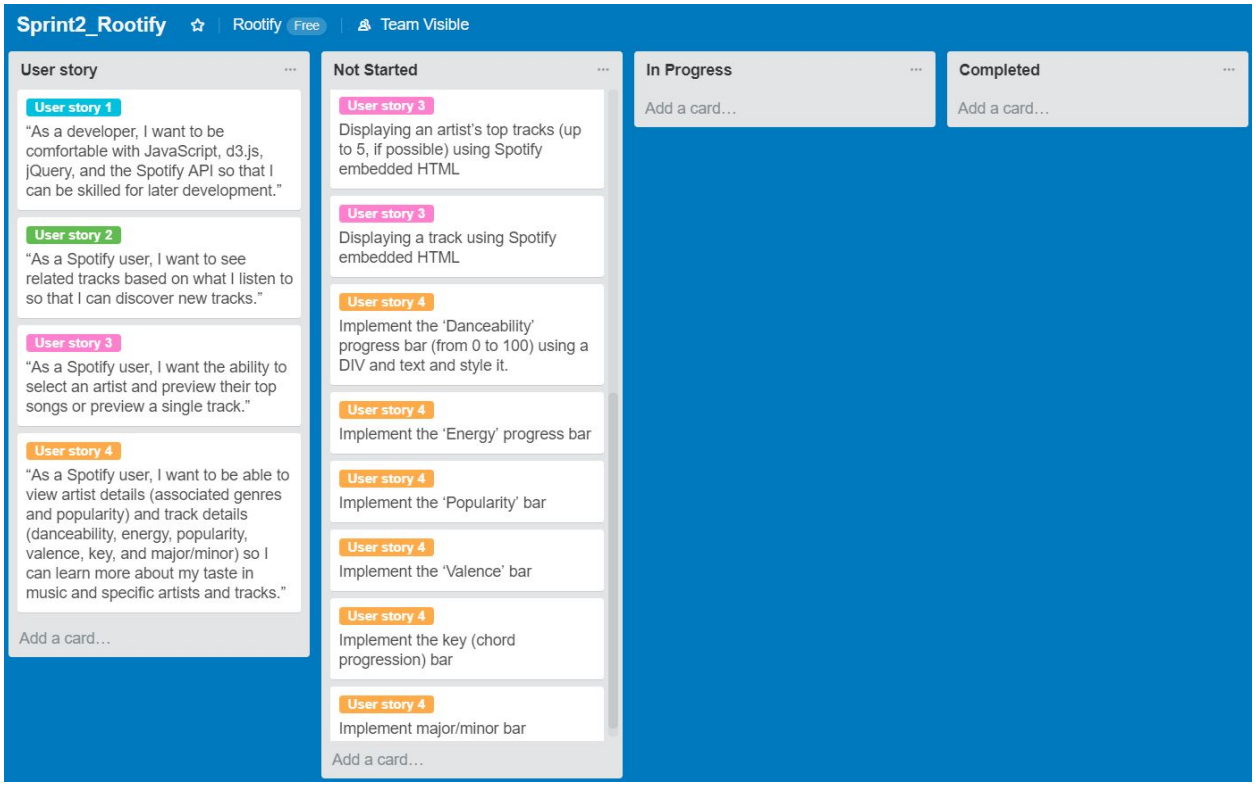
Brian Tran: **User story 1** - Learning node.js, JS, d3.js, jQuery; **User story 3 & 4**: Working on getting the Spotify previews to show up properly (given the track ID or artist ID).

Initial burn-up chart:



Initial scrum board:





Scrum meeting times:

Monday, 7:00 PM - BE 302
Wednesday, 10:00 AM - BE 302 (w/ TA Reihaneh)
Thursday, 4:00 PM - BE 302

Definition of Done:

- Code committed/checked into the GitHub repository
- Code is up to our agreement upon standards
- Over 80% of test coverage
- All unit tests pass