CMPS115 Sprint 3 Plan
LoLStats

Goal: Provide a website where a player can view twitch streams, get to interesting articles, and finalize the overall user experience.

User stories divided into tasks:

- As a frequent LoLStat user, I want to view live Twitch Streams in a section of the website, so that I don’t have to load a new page.
  Tasks:
  1. Implement twitch api to pull in top twitch streams and play them in website.

- As a LoLStat user, I would like to watch a Twitch Stream on the community section of the website so I can also view stat analysis of the players.
  Tasks:
  2. Make simplified player card to display next to popular streams
  3. Map summoner names to twitch streams

- As a trendy person, I would like to read interesting and noteworthy articles bookmarked and catalogued so that I can keep up-to-date with news.
  Tasks:
  4. Pull in reddit and other LoL news articles

- Finalize product deployment strategies (branding and domain), build community endpoints
  Tasks:
  5. Create graphic assets for the brand
  6. Implement company graphics into website frontend design

List of Tasks:

Last sprint (moved to product backlog)

- Task 7: Create a UI that displays the data
    - 2 hours
- Task 8: Pull the ward data from Riot’s API
    - 1 hour
- Task 9: Implement D3 on our website
    - 1 hour
- Task 10: Display the ward data in a useful way as D3 chart
    - 4 hours
- Task 11: Pull the location of kills information from Riot’s API
    - 1 hour
- Task 12: Load a image of the in game map
    - 1 hour
- Task 13: Display the kills on the map
    - 2 hours
- Task 15: Build frontend view UI to display these stats
    - 4 hours
- Task 17: Filter database queries based on attached win/loss info to display whether an item build/talent build/champion matchup resulted in a win
    - 4 hours

This sprint

1. Implement twitch api to pull in top twitch streams and play them in website.
    - 4 hours
2. Make simplified player card to display next to popular streams
    - 4 hours
3. Map summoner names to twitch streams
    - 4 hours
4. Pull in reddit and other LoL news articles
    - 6 hours
5. Create graphic assets for the brand
    - 6 hours
6. Implement company graphics into website frontend design
    - 3 hours

Team roles:

Logan Collingwood: Product Owner, Backend Lead
Griffin Meyer: Backend {full stack as needed}
Brandon Chai: ScrumMaster (Sprint 2), Frontend Lead
Johannes Pitz: ScrumMaster (Sprint 2), Frontend {full stack as needed}
Michael Le: Frontend {full stack as needed}

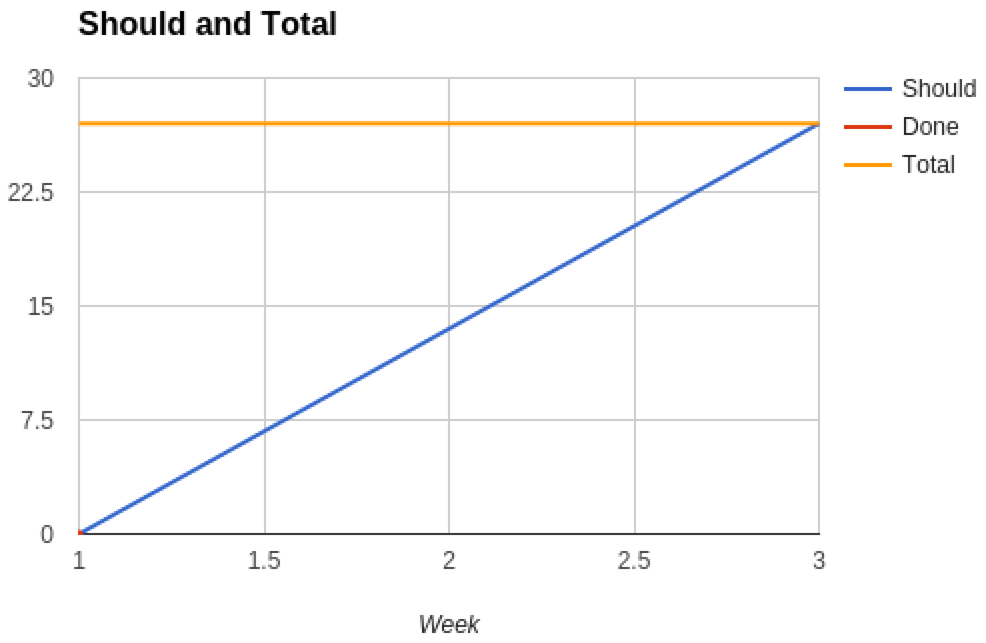
Initial task assignment:

Logan Collingwood: Task 1: Implement twitch api to pull in top twitch streams and play them in website.
Griffin Meyer: Task 1: Implement twitch api to pull in top twitch streams and play them in website.
Brandon Chai: Task 5: Create graphic assets for the brand
Johannes Pitz: Task 3: Map summoner names to twitch streams
Michael Le: Task 3: Map summoner names to twitch streams

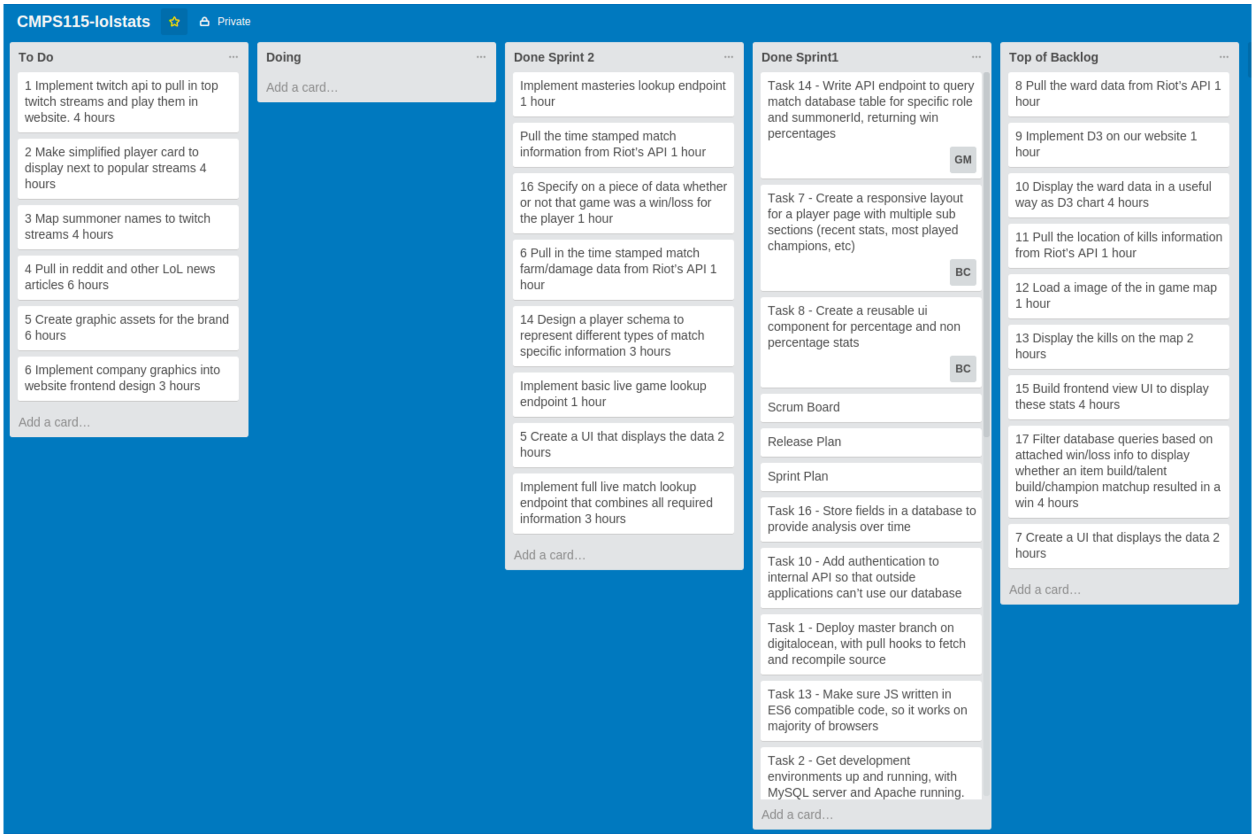
SCRUM Meeting times:

MoWeFr 12:15-12:25
Shobhit will attend our Wednesday Meetings.

Initial Burn-up Chart



Initial SCRUM Board



CMPS115-lolstats

To Do

- 1 Implement twitch api to pull in top twitch streams and play them in website. 4 hours
- 2 Make simplified player card to display next to popular streams 4 hours
- 3 Map summoner names to twitch streams 4 hours
- 4 Pull in reddit and other LoL news articles 6 hours
- 5 Create graphic assets for the brand 6 hours
- 6 Implement company graphics into website frontend design 3 hours

Doing

Done Sprint 2

- Implement masteries lookup endpoint 1 hour
- Pull the time stamped match information from Riot's API 1 hour
- 16 Specify on a piece of data whether or not that game was a win/loss for the player 1 hour
- 6 Pull in the time stamped match farm/damage data from Riot's API 1 hour
- 14 Design a player schema to represent different types of match specific information 3 hours
- Implement basic live game lookup endpoint 1 hour
- 5 Create a UI that displays the data 2 hours
- Implement full live match lookup endpoint that combines all required information 3 hours

Done Sprint1

- Task 14 - Write API endpoint to query match database table for specific role and summonerId, returning win percentages GM
- Task 7 - Create a responsive layout for a player page with multiple sub sections (recent stats, most played champions, etc) BC
- Task 8 - Create a reusable ui component for percentage and non percentage stats BC
- Scrum Board
- Release Plan
- Sprint Plan
- Task 16 - Store fields in a database to provide analysis over time
- Task 10 - Add authentication to internal API so that outside applications can't use our database
- Task 1 - Deploy master branch on digitalocean, with pull hooks to fetch and recompile source
- Task 13 - Make sure JS written in ES6 compatible code, so it works on majority of browsers
- Task 2 - Get development environments up and running, with MySQL server and Apache running.

Top of Backlog

- 8 Pull the ward data from Riot's API 1 hour
- 9 Implement D3 on our website 1 hour
- 10 Display the ward data in a useful way as D3 chart 4 hours
- 11 Pull the location of kills information from Riot's API 1 hour
- 12 Load a image of the in game map 1 hour
- 13 Display the kills on the map 2 hours
- 15 Build frontend view UI to display these stats 4 hours
- 17 Filter database queries based on attached win/loss info to display whether an item build/talent build/champion matchup resulted in a win 4 hours
- 7 Create a UI that displays the data 2 hours