Project Plan Document

Jesse Millar & Derek Hill

IT 350

October 20, 2016

Table of Contents

Project Description

Baseball is an awesome sport. There's a lot of strategy involved and, when you know what's going on, a lot of excitement. Unfortunately, there's a decently-steep learning curve for anyone new to the sport, especially when it comes to statistical terminology. Statistics have become a huge part of the game that can make or break a franchise. We aim to build a fantasy baseball app that not only allows people to compete against their friends to build the best baseball teams possible, but educates users on how baseball is played and why certain statistics are important.

Stories

General Cases

1. Generic Competition
    a. As the baseball season starts, a group of coworkers wants to have a challenge to see who can create the most successful fantasy baseball team. They download our app/visit our site, create accounts, start a competition, and set to work building their teams and tracking stats throughout the season.
2. Educational Competition
    a. A man wants to challenge his younger brother to a fantasy match at the start of a season. His brother is reluctant until he discovers our app and agrees to compete since he feels confident he will understand what is happening with the help of our explanatory product.

Specific Cases

1. Tom is playing our game and finds he doesn't understand the importance of a batting average or how it is calculated. He is able to quickly pull up a short help section and learn the reason behind the important statistic. Tom is now more prepared to build a better team.
2. The application will pull live data from a number of sources to keep the database populated with the most up-to-date stats. Push notifications can be sent when a player does well or general fantasy team stats change by a defined amount.

Stakeholder List

Direct

1. Baseball fans - Long time baseball fans will have yet another venue to celebrate their love of the game.
2. Soon-to-be baseball fans - Though they may not know it yet, our app will help regular people become fans of baseball and its many beautiful complexities.
3. Developers - Anyone maintaining the open-source system will have to fix problems as they arise.

Indirect

1. Baseball players - More exposure means more money.
2. Owners - Exposing their players means more revenue for the franchise.
3. Management - More exposure leads to the ability to sign better players.

Management Plan

The vast majority of the tasks involved in our project naturally fall into smaller categories and should only take a couple hours to complete. Example tasks are implementing our database schema, writing code for HTTP call handling, interfacing with the database, etc. Any larger, unforeseen tasks can be broken up into more bite-sized chunks. Our task list will be managed through GitHub's issue system which has built-in commenting and completion tracking.

Overall progress will be tracked through GitHub's milestone system. We create issues for small tasks and assign each issue to a member of the team and then classify those issues inside of a GitHub milestone. We can then visit the page for this milestone and view progress, make comments, and modify tasks involved in each milestone if needed.

Our critical path will be defined and managed by GitHub's project features. After creating issues, we can drag those issues into visual swimlanes representing various stages of task completion. Over time, this view will afford us a great overview of what we have left to accomplish before release.

Source code control is handled via git and synced to GitHub. Integration testing will be handled by CircleCI utilizing the Go programming language's built-in unit testing framework. Code coverage reports and unit testing statistics will be uploaded automatically to Codecov for easy analysis.

If a bug is found during development or after pushing to production, we will create a new issue on GitHub and label it accordingly. By using GitHub's issue system, we can assign the fix to a member of our team, have a discussion surrounding the fix, and log the steps we took to fix it all in a transparent, preserved fashion.

We plan to meet remotely once a week on Thursday mornings to discuss progress and any issues we've been struggling with. As mentioned above, our tracking of tasks and bugs through GitHub issues will largely negate the need for in-person meetings as most of our conversations will happen as time permits on any and all days.

Initial Task List

Jesse Millar

1. Install Go, Git, and become acquainted with GitHub
2. Configure an Ubuntu server for use
3. Set up continuous build pipeline
4. Put together boilerplate code to build off of

Derek Hill

1. Learn the basics of Go
2. Install Go, Git, and become acquainted with GitHub
3. Implement the database schema

Glossary

Codecov

A free service for keeping track of unit test coverage and test reports.

CircleCI

A free continuous integration system that allows for easy building and deploying of our code after each pushed change.

Git

Our version control system for keeping track of code changes. Git is industry standard.

GitHub

GitHub is a highly popular cloud storage solution for hosting and managing Git repositories.

Go

The programming language we will be using for our project. Go is written by Google and geared heavily towards web applications.

Issue

GitHub terminology for a ticket or todo list item.

Milestone

GitHub terminology for a section of a project. In SCRUM terminology, a milestone would represent and be used to track progress through a sprint.

Unit Test

A bit of programming code that exists to test that the functionality of other code is following the specification laid out at the beginning of the project. Unit tests usually run each time a change is made to the code.