

QUEUEING THEORY

Team Aphrodite

Nora Allison (efa91)

Bryan Djunaedi (dju90)

Stephen LaPlante (laplansk)

Evan Leon (ejl6)

Thomas Rothschilds (tgr4)

Simone Schaffer (simone09)

Nicholas Shahan (nshahan)

Evan Whitfield (evanw2)

*Requirements:
Process Description*

Software Toolset

Programming languages:

- Javascript with BackboneJS
- Scriptaculous Javascript library
- Python / Flask for server side scripting
- SQLite, on server side.

We have chosen to have our application be use Javascript for the underlying dynamic content visible to the user, since this is dominant language of the web for client side scripting.

BackboneJS is an MVC framework (C is for collection, not controller here.) in Javascript that automatically updates the page when the underlying view changes. Python / Django will be used as our server side scripting framework because some of our team members have knowledge of Python and prefer it to PHP.

Version control:

- Git

We are choosing to use Git for our version control because more of our team is familiar with Git, and we will be using GitHub for some of our other requirements, so it will be helpful to have the version control tightly integrated with the website and other tools. Some of our team members also believe that git is more widely used, and is a better tool to learn for that reason.

Bug tracking:

- Git Issue Tracker
- Google doc of known issues

Task Management:

- Google doc showing the weeks assignments.
- Team members should update their own page after every team meeting, indicating what they completed in the last week, what they are working on, and what they will be working on the next week.

Group Dynamics

Our project manager will be Bryan.

Other members will be flexible with their project roles.

At a high level, roles will be separated into front end and back end. Initially, there will be a group focused on UI design that will merge with front end after the UI design is mostly complete. Within these two groups, there will be design, implementation, and testing, but each

member will most likely overlap. Any member of the team should feel comfortable jumping into whatever task is currently our highest priority.

Initial assignments are as follows:

Simone: UI design, testing.

Bryan: Product Manager, UI design, front end implementation.

Thomas: front end implementation, testing.

Evan Whitfield: UI design, front end design, testing.

Evan Leon: back end implementation.

Stephen: front end implementation, testing.

Nicholas: back end design, implementation.

Nora: back end implementation.

These roles are subject to change as the project continues to develop.

Design decisions are to be resolved by majority. If no majority can be reached, then there will be a team discussion for at most 15 minutes, at which point the PM will have the deciding vote in the case of a continued tie. A majority vote is useful because it will ensure that each person is involved in the success of the project.

Disagreements about what a team member wants to work on can be resolved at the next team meeting. Each team member should recognize that there may be things they may not have wanted to work on, and should be willing to compromise.

Team members that are struggling with any task should ask for assistance from the PM. The most important thing with a blocked task is to get it “unblocked”, so that the team member can continue to be productive.

Schedule

This high level schedule outlines what will be completed by what days. It is up to individual groups to determine specifics. A very important milestone is the Zero Feature Release: At this point, there should be basic queue functionality from front end to back end. This includes being able to create, add to, and remove from a queue, with the queue state being stored on the back end.

Zero feature is important to keep in mind. Things like dynamically keeping the queue up to date on the front end and analytics on the back end are not of the highest priority. Once basic views, models, functionality, and communication exist we will pursue advanced features. We have a little over two weeks to get the basics. We will have a month afterward to work on features and testing.

Due April 18:

- * UI team: prepare for Paper prototype. Have first draft of prototype. This may require more team members. Recruit them if needed.
- * Front end: make final Javascript framework decision. Help with paper prototype. Prepare for Architecture with initial views, models, and paths for state diagram.
- * Back end: make final server framework decision. Help with paper prototype. Prepare for Architecture with proposed database type and format (ex. SQLite, with these tables: users, queues, administrators, etc.)

Due April 21: Paper Prototype.

- * UI team: Paper Prototype due. Recruit other team members as needed.
- * Front end: first pass at Architecture.
- * Back end: first pass at Architecture.

Due April 25: Architecture

- * UI team: begin HTML. UI team and front end design begin to merge, as the HTML and Javascript begin to interact.
- * Front end design: “final” Architecture due.
- * Back end: “Final” architecture. Database design, queries, and state diagram to handle front end paths.

Due April 28:

- * Front end: Be able to traverse the website, with basic Javascript functionality. Be able to display queues, create/add to/remove from/delete them. Use a “fake” server, if necessary hard-coding data.
- * Back end: Be able to serve the Zero-Feature front end interface, with database interaction.. Recruit more team-members if necessary.

Due May 2:

Zero-Feature Release.

Basic queue functionality should exist. Be able to create a queue and get a url to access it, where people can add/remove themselves from the queue.

By this point, front end and back end teams will work on clearly identifying what extra features will be implemented, as well as develop their own schedules for implementing them. The future schedule is up to each team to implement the additional features.

Also at this point, review use cases to confirm we are on the right track. Customer input is very important at this stage to determine features, basic layout.

Due May 5: feature proposals.

Due May 9: testing plans for feature releases.

Beta Release: May 16

Feature Complete Release: May 23

Final Release: June 4

Demo/Presentation June 6

Risk Analysis

1. Learning curve for Javascript, BackboneJS framework.

Mitigation: Nicholas has some experience with BackboneJS, and some of our team is involved with early research into the framework to determine its uses for our product.

Failure plan: Explore AngularJS. Thomas has explored AngularJS and the learning curve seems do-able.

2. How much analytics is really possible with PHP?

Mitigation: Explore possibility of a framework for managing creation/modification of queues. Explore Python and Django frameworks. The Django framework allows for a server to be set up rather quickly, allowing initial exploration into applications and analytics.

Failure plan: Fall back on PHP, which will be less elegant, but is better understood by most of the team.

3. Database integration

A risk here is that we could have trouble allowing our application to synchronize the information it is receiving for the position of people in the queues from multiple sources. Most of our team is not especially familiar with different database technologies, so we might have some trouble figuring out the right type of schema to use. In order to mitigate this risk, we could have a teammate to specifically focus on this portion of the application, so that at least one person could invest themselves in thoroughly learning the technology and feel comfortable in this domain. In order to explore the seriousness of this risk, we will probably perform some prototyping of a simple usage of a database system with the type of data we will store.

Feedback from an external user will probably be most useful to us early on in the process, after we have completed initial prototypes, and have not committed too much infrastructure to a particular design. We will hopefully solicit this feedback from the customer group we have in class, as well as talking to potential real world customers, such as CLUE tutors.