

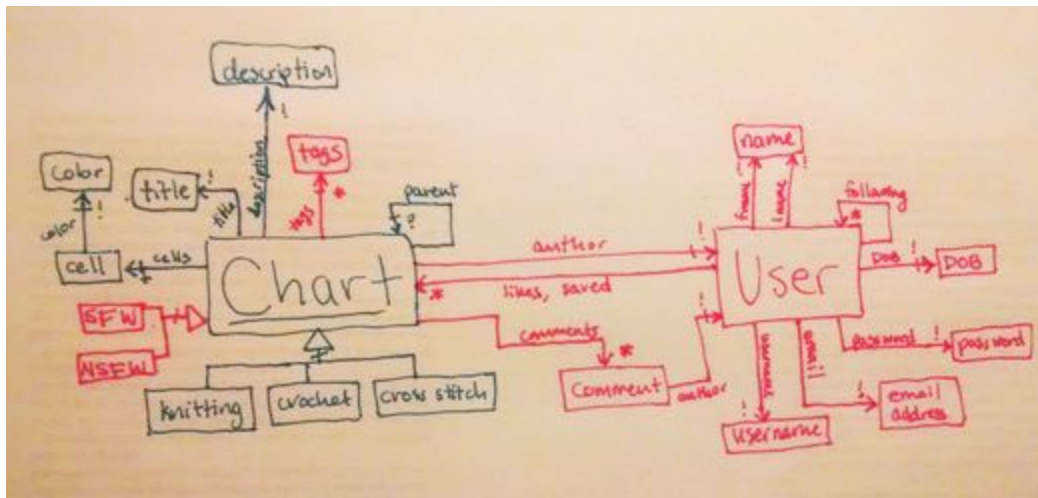
6.170 Final Project Team Plan

Minimum Viable Product

Overview

For the minimum viable product, we aim to provide users with all functionality except for traditional social aspects such as commenting/liking/following. We will instead focus on remixing, our innovative concept of sharing design patterns. In fact, we chose not to implement user accounts for the MVP. Every visitor to the site will be able to remix the charts. Because there are no users, we will also not implement commenting, following, or liking.

There plans were detailed in data model:



Pictured above: our data model. Blue elements represent components in MVP; pink elements represent components that we will add for the final version.

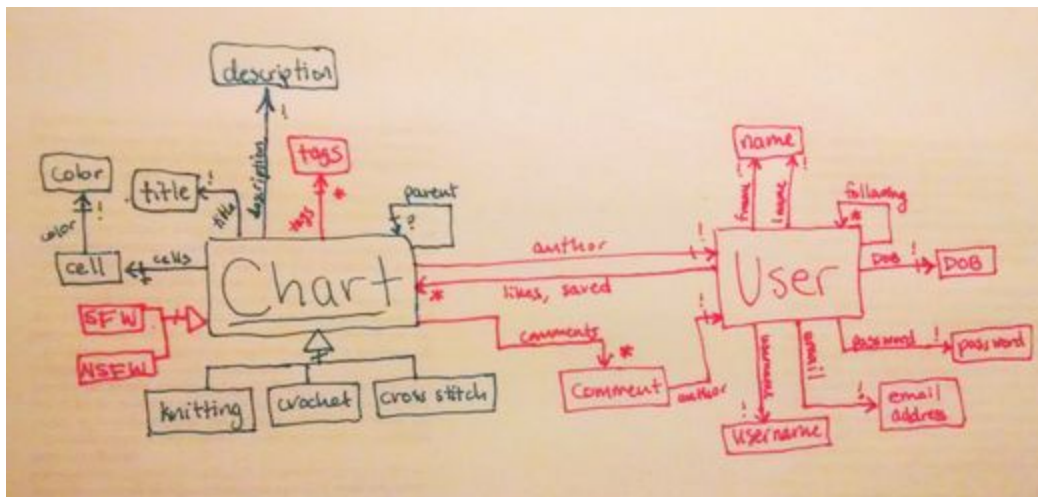
Specific Functionality and Concepts Included

Users will be able to create and remix charts. This means that a user can initiate a chart design project by setting features such as size, chart type, chart description, and colors in the grid. Additionally, all users will be able to see a feed of all other charts created on the site. Users will have the option to remix any of these existing charts. This means that a user can take an existing design, modify it as they see fit, and then share that modified design in the same way that they share original designs — these remixed designs will appear in the public feed and users can remix these as well.

Issues Postponed to Final Product

As mentioned earlier, in terms of functionality, we have postponed some colloquial social aspects of our application, ie: the User component. This also means that we will not be working on the related security components for those. For example, since there are no users, we do not need to concern ourselves with password hashing and salting. Likewise, there are no CSRF attack concerns at the moment. Since searching for charts is possible and requires user input, we will need to handle injection attacks. Similarly, we need to review our project to ensure that we are using the best practices which guard against XSS attacks.

Our user interface will allow the user to perform all functionality specified as part of the MVP. However, we make no guarantees regarding the aesthetics. This means there is no CSS styling. We do not even guarantee that HTML elements will be positioned in intuitive fashion, although we will try our best. We will save much of the styling for the final product.



Pictured above: our data model. Blue elements represent components in MVP; pink elements represent components that we will add for the final version.

Tasks

Server-Side (Karleigh and Val)

Task	Estimate	Assigned to
Implement schema for Chart in MongoDB.	2 hours	Karleigh and Val

Handle user request to create a Chart. Involves responding with a blank Chart and corresponding HTML page where user can edit the Chart.	3 hours	Val
Handle user request to save a completed Chart.	2 hours	Karleigh
Handle request to get a collection of Charts. The must also handle the option to sort by date.	3 hours	Karleigh
Handle user request to remix a chart. Involves fetching a chart, responding with a chart, and responding with HTML page where user can edit the Chart.	2 hours	Val
Launch application on Heroku	1.5 hours	Karleigh and Val

Client-Side (Denis and James)

Task	Estimate	Assigned To
Implement a client-side Chart model for representing the chart that users color on.	3 hours	James
Implement controller and view for Chart, so that a user can see and edit the Chart locally.	3 hours	Denis
Implement HTML for the Chart creation page.	2 hour	Denis
Implement logic (JavaScript) for Chart creation page. This includes interaction with the server and handling user input.	3 hours	Denis
Implement HTML for the Chart remixing page.	3 hours	James
Implement logic (JavaScript) for Chart remixing page. This includes interaction with the server and handling user input.	2 hours	James

Milestones

By 11/21 Monday: Complete a first-pass implementation of the MVP

By 11/22 Tuesday Evening: Go through code review and confirm correctness to aim for a final implementation

By 11/23 Wednesday Noon (Official MVP deadline): Ideally should have to do nothing, but will iron-out remaining bugs if they exist.

Final Product

Overview

For the final product, we will add in the User component. Users will be able to create and log in to personal accounts, as well as follow other users, like and comment on charts, remix charts, and save charts.

Additionally, our UI will be more polished, as well as having CSS styling and intuitive layout. In addition, security issues (potential XSS/CSRF attacks) will be fully addressed.

Tasks

Note that since we have yet to finish our MVP, we do not yet know how work will be allocated for the final project. After the MVP, we can assess our pace and better make judgments on who should work on what, as long as how long tasks should take. Still, we have already made considerations of what we would need to do for the final product. The list of tasks below is not exhaustive, but gives us an idea of important functionality that we wish to implement for the final product.

Server-Side

Task
Implement schema for User in MongoDB
Implement authentication of a user. This is basically the security aspect on the server side.
Implement inappropriate content mitigation. Basically block underage users from viewing mature content.
Implement schema and request handling for reporting system.
Implement sessions and creation for user accounts.
Handle search request by user to search for Charts related tags.
Extend Chart schema and update handling of Chart creation and remixing to handle tags.
Extend the Chart schema and handle like requests from users.

Handle save requests from users.
Handle requests from users to follow other users as well as request for viewing all charts of users followed.
Extend the Chart schema and handle commenting requests from users.

Client-Side

Task
Update the HTML page for Chart creation and remixing so that users can create tags.
Implement client-side signing up and logging in functionality.
Implement the forms for searching and commenting.
Implement the buttons and request-sending for liking and saving.
Implement button and request-sending for flagging mature content.
Implement the form for doing a report. Also the button for bringing the user to the reporting screen.
Go through all existing client-side code and performing validations on data to avoid useless/unreasonable requests.
Go through all existing client-side code and perform necessary
Implement CSS styling for all pages. This will also include any artwork to include in the HTML.

Milestones

By 11/29:

- Implement User schema
- Implement login, account creation
- Make pages/modal boxes for above

By 12/6:

- Implement liking/saving/following/mature content/tags
- Implement comments
- Implement report content button

By 12/11 (feature freeze is 12/7 so this will be just polishing):

- Make logo, put on site
- Polish interface
- Clean up code
- Fix any remaining bugs

Handling Mis-estimations

There are two primary concerns we have with mis-estimations. The first of these is unbalanced workload. We want everyone on the team to get a fair share of the work. If a task is too short or too long, we will move tasks around so that everyone is working for approximately the same amount of time.

The second concern we have with mis-estimation is that some tasks might take significantly longer than expected. We have already accounted for this possibility. Therefore we plan to have final implementations be due an entire day before the official deadline. We also aim have to a complete implementation a day before that, but we assume there will may be bugs needed to straighten out.