StitchHub Design Documentation

By National Badger: Karleigh Moore, Val Healy, Denis Li, James Usrey

Overview

Motivation (Val)

In our project, we have set out to build a social web application that allows users to create, share, save, and remix colorwork charts (design patterns) for knitting, crocheting, and cross stitching.

While many existing chart-making apps allow users to share patterns through other forms of social media, as well as email, none of them really acted as a social network in and of themselves. Many of these sites have clunky or aesthetically displeasing interfaces. Furthermore, only one of the applications allowed users to customize existing patterns on the site. We hope to resolve these issues in our application.

What does our application do?

Our application will focus on two primary purposes: the creation of designs and the creation of derivative designs based on existing designs.

The function of a design is to allow users to plan their stitching projects in advance so that they might use the finished design for reference while they are stitching project. Other sites allow the creation of designs, but we will improve the user interface. We will also introduce some new social features, such as comments, likes, and saving. This will allow other users to give feedback to designers and allow designers to connect with each other through their designs.

We will also allow users to create derivative works based on existing charts. A user can use this feature to create custom designs based on existing designs on the site. Users can collaboratively mutate designs and then trace the genealogy of each design. This will supplement the social aspect of our app as well as allow users inspiration for their design and a chance to customize their favorite patterns to suit their own needs.

We also allow users to follow other users. This lets a user filter out to see only the charts of those that he/she follows.

Concepts (All members contributed)

Name	Purpose	Operational Principle
Chart	Provide a visual representation of the design for the user	When a user creates a new chart to share on StitchHub, the user initializes it by specifying a size and title. The user can then select cells in the chart in order to assign colors to the particular squares in the chart and create a design. The user saves the chart when finished and this saves the chart to the database and allows the chart to appear in the public feed so other users can see (and choose to remix) the design. Users can comment on charts and like or save any chart but their own.
Remixing	Users can take existing designs and edit them to their liking and create a new design to share.	When a user creates a design, it appears in a feed visible to all users. Other users can click to remix an existing design. By remixing an existing design, a user can edit the colors on the original design and make and save a new design (similar to a retweet you can edit).

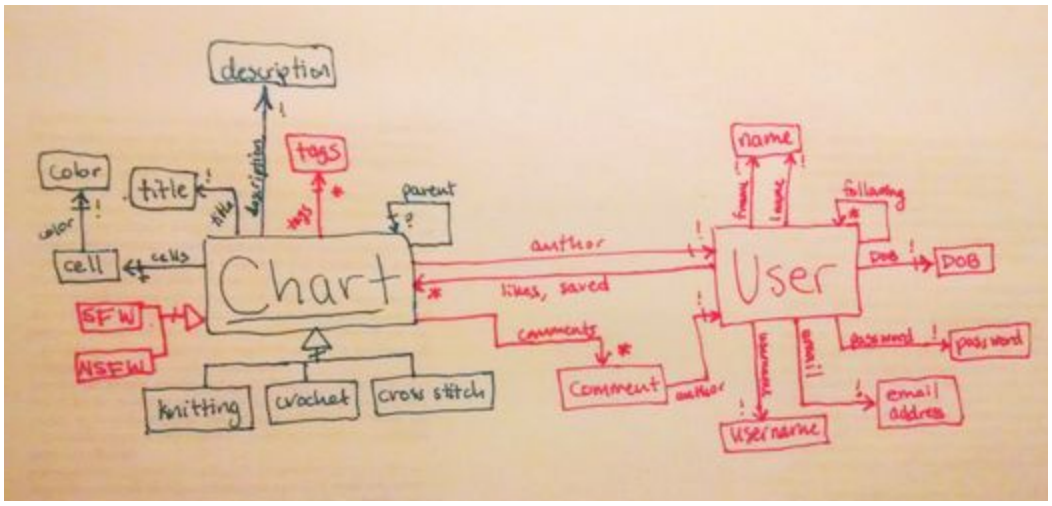
Additional concepts we plan to include that are self-explanatory are

- Description
- Tags
- Following
- Liking
- Save
- Searching
- Sorting (by date/most liked/etc.)
- Comments

Anticipated Misfits:

- One misfit found in almost all of the existing apps is incorrect cell aspect ratio for different chart types. Knitted stitches have a different aspect ratio from crocheted stitches or cross stitches. We will remedy this by allowing for different aspect ratios, each paired to a chart type (knitting, crocheting, cross stitching).

Data Model (Val)



Our data model is pictured above. The blue elements represent the sets and relations of our MVP. The pink elements will be added for our final project.

Textual Constraints:

- A User cannot follow themselves
- A Chart cannot be its own parent
- Title cannot be empty string
- Username cannot be empty string and must be comprised of only numbers, letters, and underscores
- Password must be a minimum of 6 characters and can comprise of numbers, letters, and ascii special characters
- E-mail should be a valid email address

Design Choices (All members contributed)

Resolved Design Choices

Concern	Choice Made

Can remixes be exactly the same as their parent? Allowing this may allow one user to plagiarize another's chart.	We have decided that this is acceptable, since the new chart will still have its parent connected to it, and the app will have a report button for issues such as these.
Should Users be able to resize their Charts? If so, where should additional rows be added? Should we allow the user to choose to append rows to the top and bottom? Similarly for columns?	We have decided to allow users to determine chart size, but this will be a fixed size after the decision is finalized and cannot be changed on the fly. Additionally, when remixing, chart size cannot be changed.

Unresolved Design Choices

Concern	Options to Consider
What happens when a User deletes one of their Charts? In particular, how will this affect the children/remixes?	● All charts are derived from the deleted chart via remixing would be deleted. ● Descendant charts of a deleted chart would still mention it as an ancestor with the name "[Deleted]". However, there is no link associated with that chart.
Can users edit charts that they have created?	● Users are allowed to edit the chart, but this affects a number of things like remixes no number resemble the parent. ● Users are not allowed to edit the chart, but this does not give the user flexibility.
What is the maximum size (# rows/columns) of a Chart?	● Have no limit so that users can have flexibility. ● Maximum size, if implemented, must not bog down our database or make the app run too slowly.
What information should we display with each Chart on the dashboard? How should we display this information (text, icons, etc)?	*There are a number of options to this, which we will consider later when actually implementing the UI.

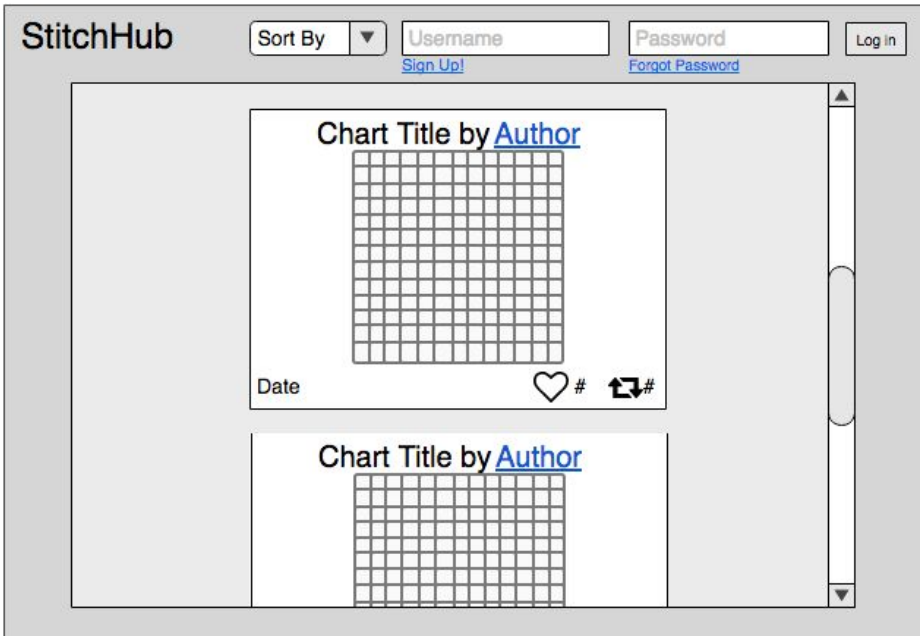
Can users zoom in/out?	• No zoom. This would be easier to implement. • Discrete zoom options (selection?) • Continuous zoom options (slider?)

Design Risks (Denis)

Risk	Solution
Plagiarism	A user submits a report via a reporting system that we have. Our team will manually look through and handle those reports.
Copyright (e.g. creating content like Pokemon)	Companies that find copyright-infringing material will report the material to us.
Inappropriate content	Users can flag content as inappropriate in which case we will not display it to younger users.

User Interface (Karleigh)

Landing Page (Not logged in)

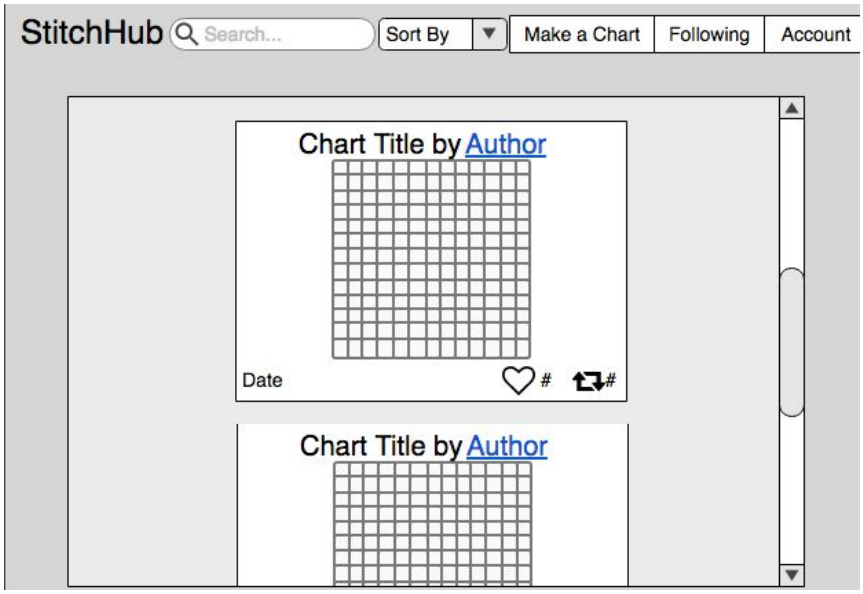


If a user clicks “Sign up!” they’re redirected to the following page.

Sign Up Page

Once the user logs in, they are directed to the home feed.
A user will immediately see the public feed but can click on a particular user to see only that user’s posts by going to that user’s profile.

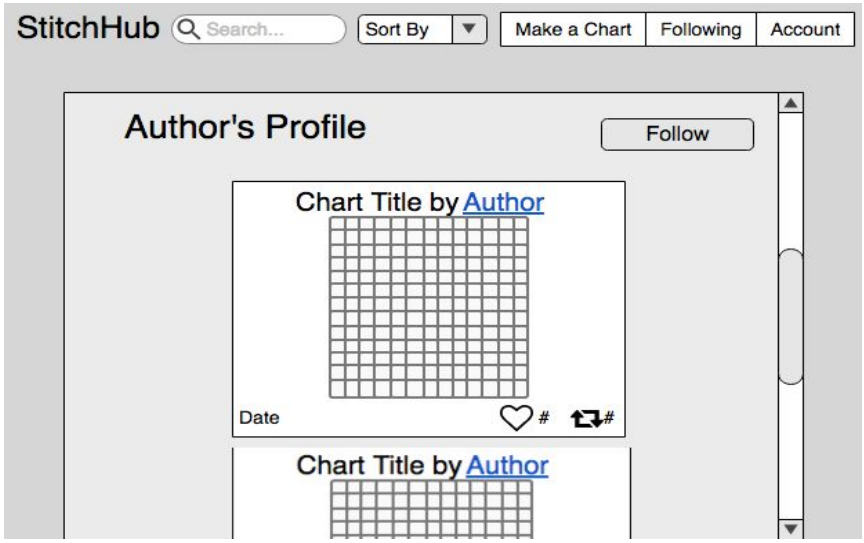
Landing Page (Logged in)



Clicking on “Following” will show users a feed populated by posts made only by users they are following. This will look similar to the home feed view-wise — we will just have some backend stuff filtering which posts make it to this feed.

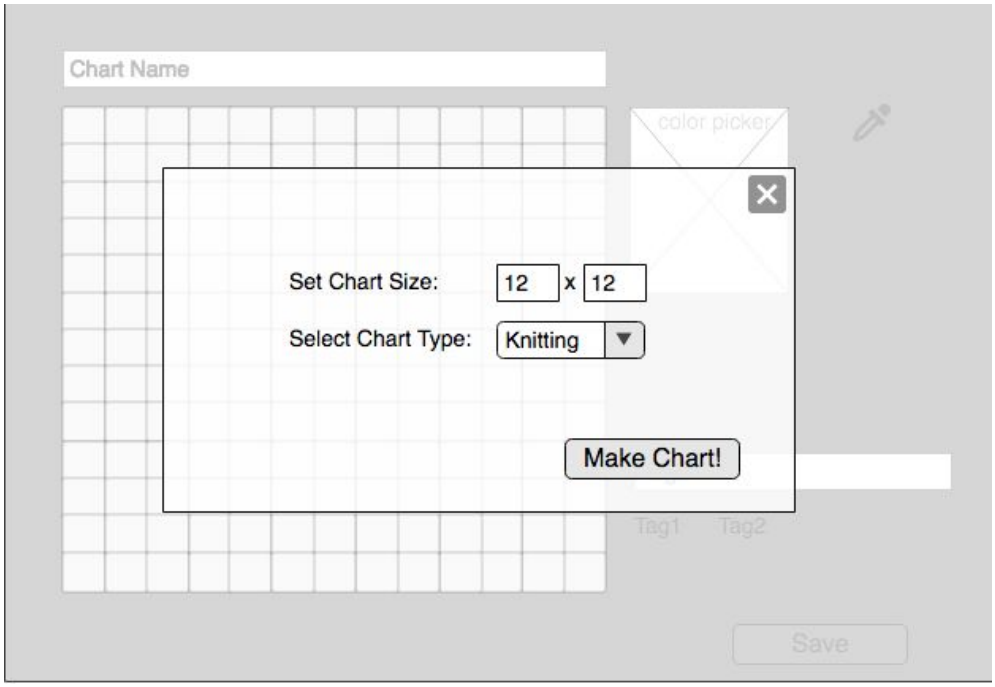
We can get to the user profile page by clicking on the author’s name.

User Profile Page

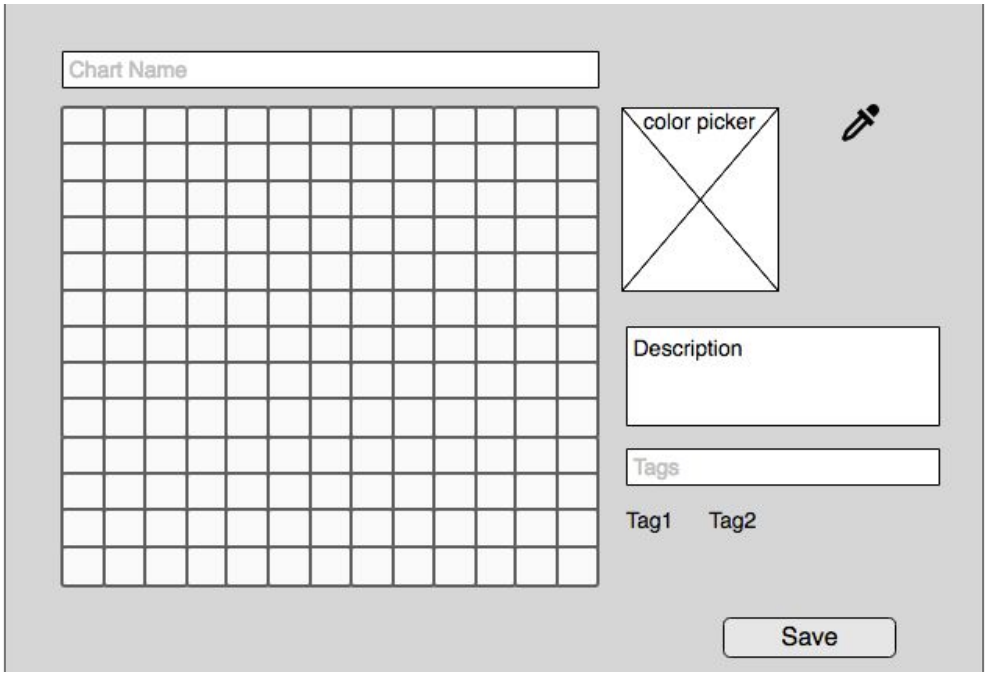


When users click “Make a Chart”, they are taken to the following page where they need to specify the type and size of the chart they want.

Chart Creation Page



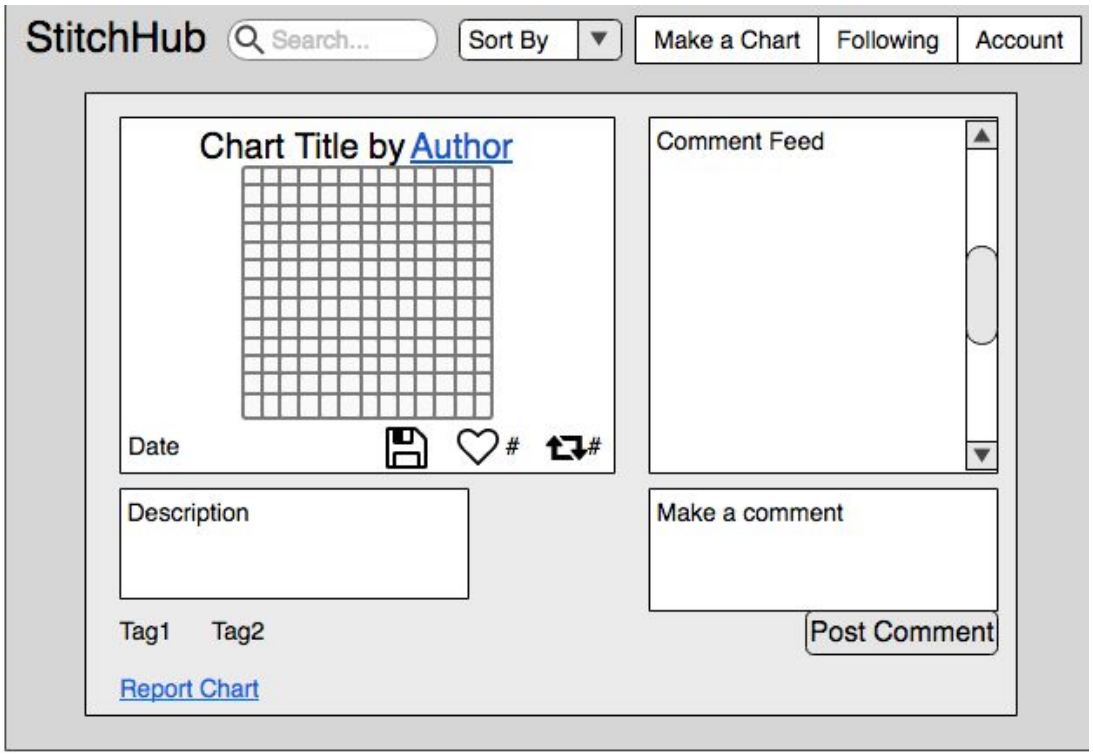
Once the user specifies the size and type of their chart, they are taken to the following screen. If they click the “X” button, they are taken back to the home feed.



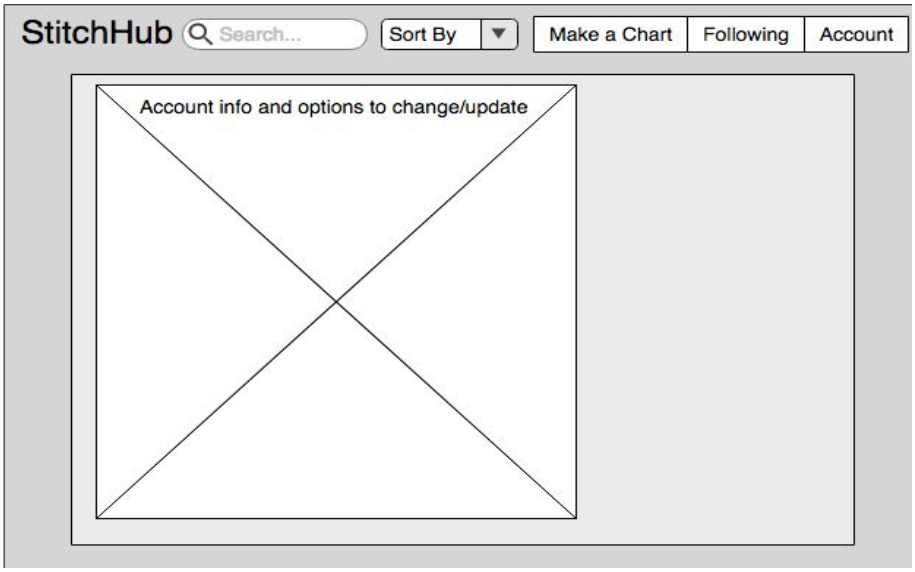
The user can modify the chart by selecting colors from a color picker and using an eyedropper-type tool to determine existing colors they've used (in case they forget). Once a chart is posted, the user is taken to the chart's page.

If the user clicks on an existing chart (that they find on someone's profile or in their feed), they are taken to a page where they can see the chart more closely and add comments.

Chart Page

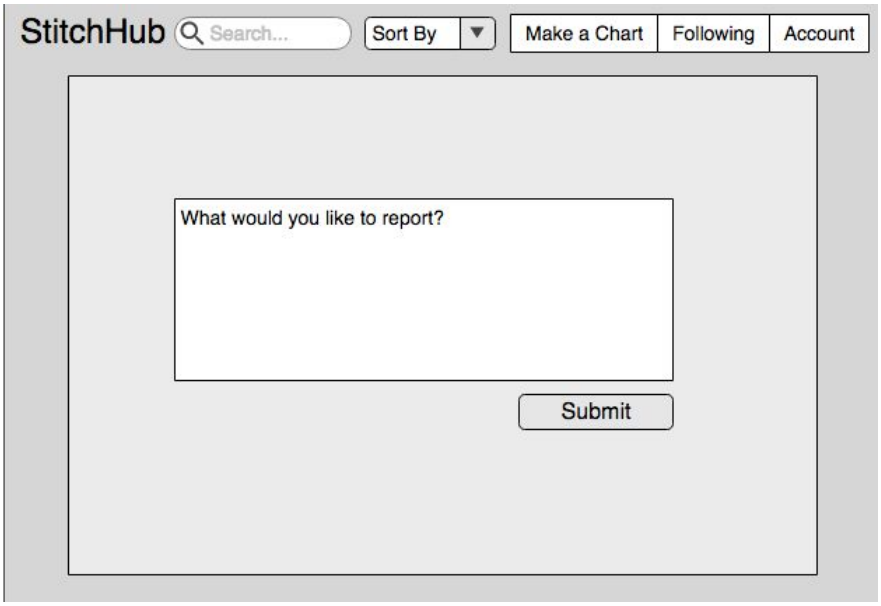


Account page



Report page

If a user clicks to report a chart, they are taken to the following page.



Security Concerns (James)

Key Security Requirements

Requirement	How to satisfy requirement
Protect user information like email, DOB, and password	We will hash and salt the password via reputable libraries like crypto.
Users should be able to log in using the username and password they specified when they registered their account.	The server should properly authenticate these credentials, and use them to enforce security constraints.

Mitigating Standard Web Attacks

Cross-Site Scripting (XSS) and **Cross-Site Request Forgery (CSRF)** attacks are mitigated using best practices and prevention methods described by OWASP at the following links.
https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet
https://www.owasp.org/index.php/CSRF_Prevention_Cheat_Sheet

Injection attacks are mitigated via the use of MongoDB, as it uses fixed calls rather than constructed queries.

Threat Model

What assumptions we're making about attackers

- Assume they want passwords to take over user accounts.
- Assume they want to delete the database.
- Assume they want access to email addresses (can be used to spam, or used in addition with passwords to potentially hack other sites).
- Assume that members of the team want the passwords of users.