[ArtRank]

[Mike McCarthy, Co Giang, Daniel Lipeles, Senh Wang,
Adonis Familia, Pat Beagan, Sean Luo]

System Design Specification and Planning Document

Draft 1
2/25/15

ArtRank

Mike McCarthy, Co Giang, Daniel Lipeles, Senh Wang, Adonis Familia, Pat Beagan, Sean Luo

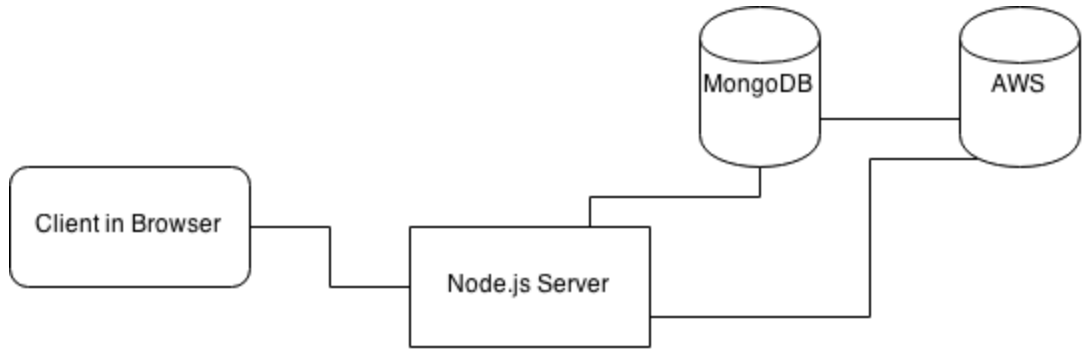
CSRocks Inc.

Version	Primary Author(s)	Description of Version	Date Completed
1	Mike McCarthy, Co Giang, Daniel Lipeles, Senh Wang, Adonis Familia, Pat Beagan	This is the first draft of ArtRank.	02/16/15

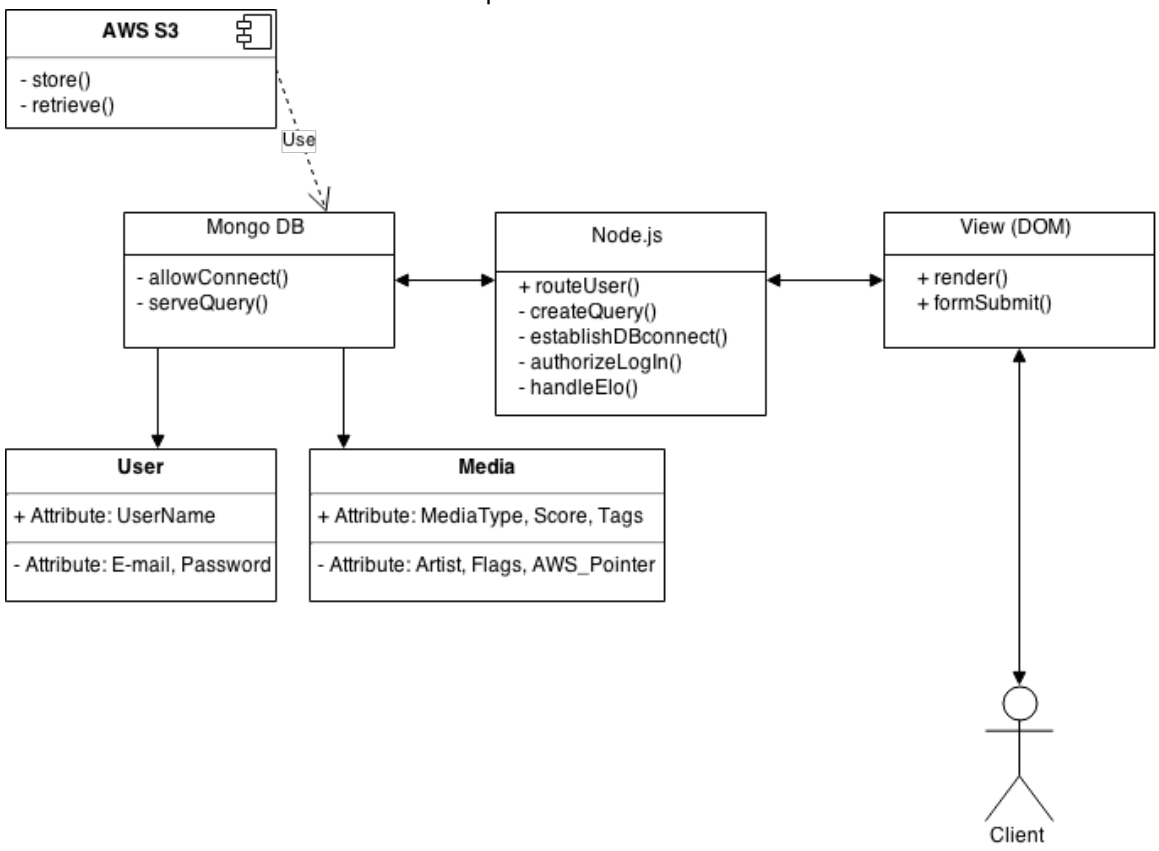
System Architecture

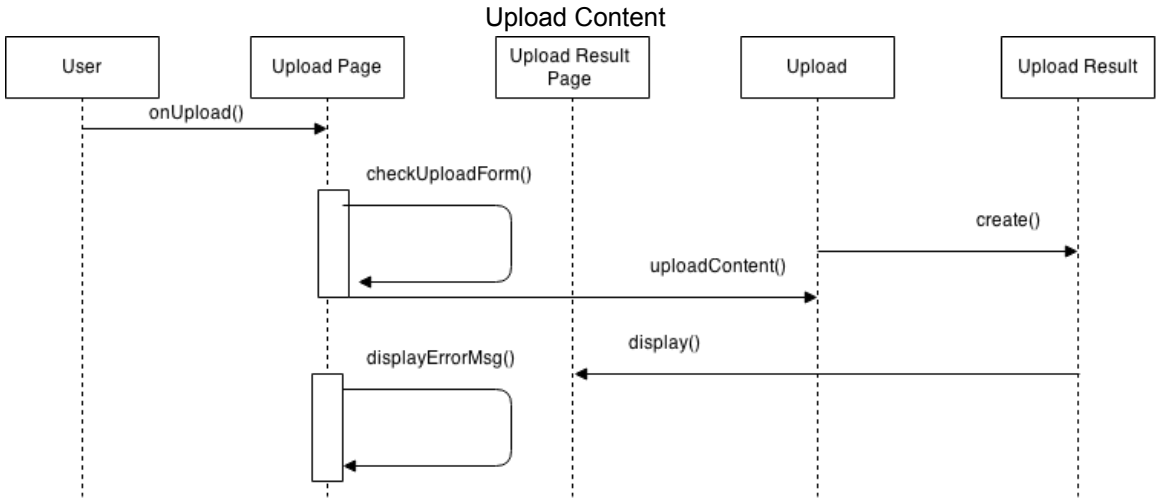
The system architecture has been organized around a fully javascript stack in order to target the browser as the platform and keeping the entire team working in one language. The architecture and design decisions were focused on making the development logical but also to parallel the philosophy of the site as a comparison of pieces rather than as an artist's promotional gallery. The decisions and organization of the database schema is probably the most definitive element of the system architecture as it enables all of our actions. It was crucial to define and our solution fits with our core thinking for the project. Overall the system architecture is set to create a great in browser experience with a solid core back end that serves content properly and efficiently according to our algorithm.

Implementation Overview

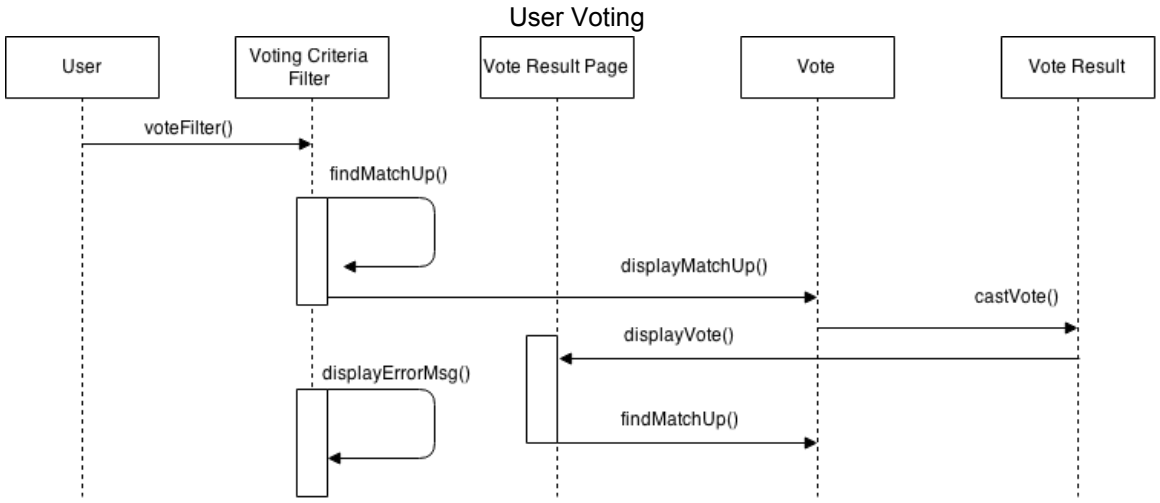


Implementation UML





In this process view, a logged in user goes through the steps to upload a piece of content.



In this process view a user goes through the steps of

The role of the database will be filled in with a combination of MongoDB and Amazon Web Services. MongoDB will house the user account information as on of the main objects to be stored. The user will have a username, password, and email as properties. The actual content storage will be offloaded to Amazon Web Services, with MongoDB holding the meta-data and location of the content on AWS. The meta data consists of the type or format of the media, it's genre defining tags to compare it in relation to other content, a count for the number of times the piece has been flagged as inappropriate, and the artist that uploaded the piece.

For our database schema, we had considered the potential of storing the art objects as elements within the user object. While this makes it easy to keep track of a users portfolio and contributions to the site, this did not follow with our vision of treating each individual piece as competitor or

contender with a ranking. Having them be separate allowing each pieces to stand on it's own as presented on the site but also simplifies the sorting by rank with the entirety of the library of pieces sorted by rank.

Another decision we had to make was weather we wanted to handle the storage of media ourselves. While we ultimately ended up choosing Amazon Web Services, we considered storing the media directly in our own database. This gives us the flexibility of having full control of all of our components and not have to worry about the outside third party solution. The benefits outweigh this as we now do not have to handle the stress and load of uploading and serving high quality images and audio files directly and can focus on other important aspects of the project. The assumption is that this setup and interaction between MongoDB and AWS will function in the ways that we expect.

1. Team Structure

Our team is divided into sub-groups of 2 people apiece. This structure both allows easy coordination within a subgroup and encourages oversight for each task. One group will be responsible for the front end, one will be responsible for the back end, and one will be responsible for testing and documentation. Each one of these sub-groups will be accountable first to the project manager, and then to each of the other teams in turn. The teams are organized as follows: the front end team is Pat Beagan and Mike McCarthy, the back end team is Daniel Lipeles and Adonis Familia, and the testing and documentation team is Co Giang and Senh Wang.

The front end team is responsible for ensuring that the UI/UX of the website is both functional and attractive, and the user sees appropriate content. The back end team is responsible for the database of the website, and interactions with web servers (Amazon, DigitalOcean). The testing and documentation team is responsible for the bulk of testing (unit testing, system testing, and usability testing), as well as maintaining and creating the website's documentation both for developers (requirements specification and design specification) and users (FAQ, integrated text walkthrough).

Our team will remain in constant communication through Slack and post all information on due dates and weekly updates on Trello. Major group decisions are made during our weekly meetings at 4 PM on Fridays in the USpace, as well as during class discussion sessions on Wednesdays.. Communication between members of the same sub-group will be done through texting and email.

2. Project Schedule

Task	Estimated Effort	Due Date	Responsible party
Requirements Documentation	1 day	2/17/15	ALL
Design Documentation	1 day	2/25/15	ALL
Setting up initial database	1-2 days	3/2/15	Back
Setting up initial UI	1-2 days	3/2/15	Front
Initial testing framework and updates to existing documents	1-2 days	3/2/15	Test
ALPHA RELEASE		**3/5/15**	**ALL**

Submitting Content	2 days	3/13/15	Back
Supporting Image media	1 day	3/13/15	Front
Voting on Content	1 day	3/13/15	Front
Elo ranking system	2 day	3/13/15	Back
Comments on pieces of art	2 days	3/23/15	Front/Back
Tagging pieces of art	1 day	3/25/15	Front/Back
User choosing which tag they want to vote under	2-3 days	3/27/15	Front
FAQ section	1 day	3/27/15	Front/Test
Testing of all new features and updates to existing documents	4 days	4/1/15	Test
BETA RELEASE		**4/2/15**	**ALL**
User Login	2-3 days	4/10/15	Back
User Profile / Gallery	2-3 days	4/14/15	Front
Support Video, Text, alternative types of media	3 days	4/19/15	Front
Integrated help text for new users	1-2 days	4/21/15	Test/Front
Ability for users to favorite artists	2 days	4/23/15	Front/Back
Ability to report inappropriate content	1 day	4/25/15	Front/Back
Testing of all new features and updates to existing documents	4 days	4/30/15	Test
FINAL RELEASE		**5/1/15**	**ALL**
Gamification (how to incentivize users to vote)	2 days		Front/Back
Suggestion algorithm based on user data	3 days		Back

Ability to message/view profiles of other users	1 day		Front
Ability for users to post ideas of what kind of media they'd like to see which artists can view for project ideas	1 day		Front

3. Risk Assessment

The top 5 high risk areas in our project are a lack of users, legal liability for copyrighted content, moderation of inappropriate content, the security of user accounts and the accuracy of our ranking algorithm. This risk set has been narrowed down since the SRS, as some issues were solved through design, and the ones which were still an issue were made more specific. Each of these risk areas is either unlikely to occur or is easily mitigated if it does occur, making the project relatively low risk.

The two greatest risks for the site will be security and the size of the user base. Security will be one of our main concerns after the beta release, so we will be able to solve the issue with extensive testing. The size of the user base is much more in contention because it isn't something we can directly solve. It will become the main issue for the project as it nears its final release. To mitigate this, we will attempt to seed the site with artwork from students around campus and promote the site on other forms of social media.

Risk	**Chance of occurring**	**Impact if it occurs**	**Prevention plan**	**Mitigation plan should it occur**
Lack of Users	Med	High	Seeding the site with content to encourage user participation.	Promote site on social media and around campus to form a userbase.
Legal Liability	Med	Med	Include a waiver passing all liability to the user.	Immediately remove questionable content and cite the waiver the user signed when they registered.
Moderation of Offensive Content	Low	High	Include a "Report" button to flag content. Have a separate algorithm taking into account the ranking of flagged content.	Have a human moderator giving the final say on the offensiveness of material.
Security of Accounts	Med	High	Use an encryption library to encrypt accounts and passwords.	Lock the compromised account until recovery efforts have been made by the user.
Algorithm Accuracy	Low	High	Extensive testing of accuracy and refining algorithm based on testing.	Disabling the algorithm and rewriting it until we get the desired effect.

1. Test Plan

All these tests will be developed with QUnit, a Javascript unit testing framework. They should all be run at least once a day to ensure the website if functioning properly. When testing, we will pay particular attention to edge cases, such as special characters. The mechanism we will use to achieve these tests is passing in an expected value, running the function we want to test, and comparing the result to the expected value.

Unit tests

User log in - This test will make sure the user logs in successfully if he enters his email and password correctly. It will also check to make sure the user does not log into the system if he does not enter his email or password correctly.

Input sent correctly to server - This will check to see if what the user inputs is what the server actually gets. For example, if she enters an ‘A’, then the server receives an ‘A’. Also, if the user votes for a piece of art, then the vote is registered correctly in the server.

Head to head Algorithm - This test will verify that in a head to head match up, the piece of art that wins will increase in ELO, while the one that lost will decrease in ELO. It will also verify that in a tie, the higher one loses ELO while the lower one gains ELO. If both pieces of art have the same ELO in a tie, both their ELOs will remain the same. Also, this test will have to verify that if a piece of art beats an opponent with a higher ELO than it, then it will gain more ELO than if it beat an opponent with a lower ELO than it.

File upload is correct - The test makes sure each content added is placed in the correct category. A mock file will be used to test file uploading capabilities. Also test to see that we don’t accept files that are too big correctly.

Links working properly - This test will check to ensure that the links direct the users to the correct location.

Connection lost - This will test to see that if a user loses connection, after a certain amount of time, say 60 seconds, then the connection times out and an error page is displayed. If the connection is slow and not adequate for using the website, then a dialog will warn the user. It will also test to make sure the user is where the were once they log back in i.e(saved their vote history, favorites, etc.)

Ladder showing top pieces of art - This checks to see that the ladder is correctly displaying the top pieces of art in the correct order.

User favorite button - This checks to see that if a user favorites an artist, then the server is saving that correctly.

Flag inappropriate content button - This will test to see that if a user flags a piece of art as inappropriate, then the server receives it.

System tests

Archiving correctly - This test will make sure everything is getting archived correctly (user vote history, favorite artists, ladder, pieces of art, ELO score).

Sign Up - This will test to see that the user is successfully signed up and added to our server if he enters a valid email that is not already in our system, his password meets the minimum security requirements, and his user ID is not already taken.

ELO algorithm - This will check to see that a piece of art is moving up the ladder correctly. For example, if two pieces of art are both 10-0, the one that has faced higher ELOed opponents (higher opponents' average ELO) should be ahead of the other one on the ladder.

Usability tests

A/B testing **-** A/B testing is comparing two versions of a web page, a controlled version and a varied version. We will measure how each version does in terms of user traffic and performance to determine which web page is better.

Watching users go through use cases - We will watch our customer relations group (SaveYour), go through use cases to see that everything is working properly.

Feedback button - There will be a submission form for user feedback.

Check which buttons are pressed more (Heat Map) - There will be a data logger that keeps track of which pages are popular.

Adequacy of test strategy

Our plan is to make sure each of the test cases are met for every criteria. We will ensure every functionality of the site works properly as designed. Also in cases where there are many active users at once voting or uploading onto the website concurrently, we will perform stress tests to make sure we can handle the increase in traffic.

Bug tracking mechanism and plan of use

A database will be used to keep track of bugs that arise. Developers will have the capability to tag bugs in terms of severity and record how to reproduce that bug which will make it easier to fix it. Since we will be using GitHub, we will also track bugs using the GitHub issue tracker.

2. Documentation Plan

There will be integrated help text to assist first time users to help them navigate through the site. It is important to make sure they are not obtrusive and users can have the option of turning off the them if they don't need it. Therefore, it will not appear for users who have used the site before. An example walkthrough of the help text could be:

- user logs in for the first time and is greeted with a "Welcome to ArtRank" message
- the user will then be prompted with two pieces of art and then asked to choose which one he prefer

- once he has gone through one round of voting, we will show him both art pieces' artists and ELO rating and provide a brief explanation of how the ELO rating system works as well as how he can view both artists' profile page.
- he will then be directed to the gallery/main page, which has a display of the top pieces of art for each category.
- finally, he will be shown the site's other features such as favoriting an artist, viewing an artist's profile, the comments section, the report button, user vote history.

There will also be a FAQ page that will answer some questions that we anticipate users will have such as:

- What is the ELO rating system/how does it work?
- How do I upload a file?
- How do I report an offensive image?
- I really like this artist! How can I find out more about them?
- How do I find interesting art to vote on?
- Why don't I have access to all the artist's images/videos when I visit their profile page?
- Do I get anything for voting a lot?