Open Paint Online

Carlos Sandoval
Brandon Ruhl
Gabriela Martins
Adam Omar
Richard Strobel

1. Project goal:

- To create an online paint editor that does not require users to download or install any software.
- Accessible from multiple devices and allows for multiple devices to be in sync.

2. Project description:

Open Paint Online is a basic graphics/painting utility that is hosted on the web. This utility consists of a simple canvas and editing tools that allows users to express their artistic skills. The tools implemented include a free-drawing pencil tool, spray brush, text tool, line tool, ellipse tool, rectangle tool, and a canvas clearer. Some of these tools can have their properties manipulated using the color picker and brush size options. The following filters are also available: “grayscale”, “pixelate”, “invert”, “brightness”. These manipulating features are used in combination with one another at the user’s discretion, e.g. the brush can be green, and the user can still adjust brush size.

Users are able to register an account. Once registered, users can login from any device with a web browser. Logging in to Open Paint Online gives users the ability to save and load their art. The art that they save is linked to their personal account and does not require the user to maintain storage. The advantage of maintaining the users storage for them gives them the ability to save work on one computer or device and pick up where they left off on another computer or device. Open Paint Online is the paint editor for the twenty first century.

3. User specifications:

a. Functional requirements

Requirement ID	Requirement Statement	Must/Want	Comments	Contributed
1	Users shall be able to draw on a canvas.	Must	Complete	Carlos
2	Users shall be able to login.	Must	Complete	Brandon
3	Users shall be able to log out.	Must	Complete	Brandon
4	Users shall be able to save their work.	Must	Complete	Brandon
5	Users shall be able to load their work.	Must	Complete	Brandon
4	Users shall be able to change color of tool.	Must	Complete	Gabriela
5	Users shall be able to use the pen tool to free draw.	Must	Complete	Carlos
6	Users shall be able to use the spray brush tool for for free draw.	Must	Complete	Carlos
7	Users shall be able to use the line tool to draw straight lines.	Must	Complete	Gabriela
8	Users shall be able to change the size of the brush tool.	Must	Complete	Gabriela
9	Users shall be able to use the erase to clear the canvas.	Must	Complete	Carlos
10	Users shall be able to use the text to insert text into the canvas.	Must	Complete	Carlos

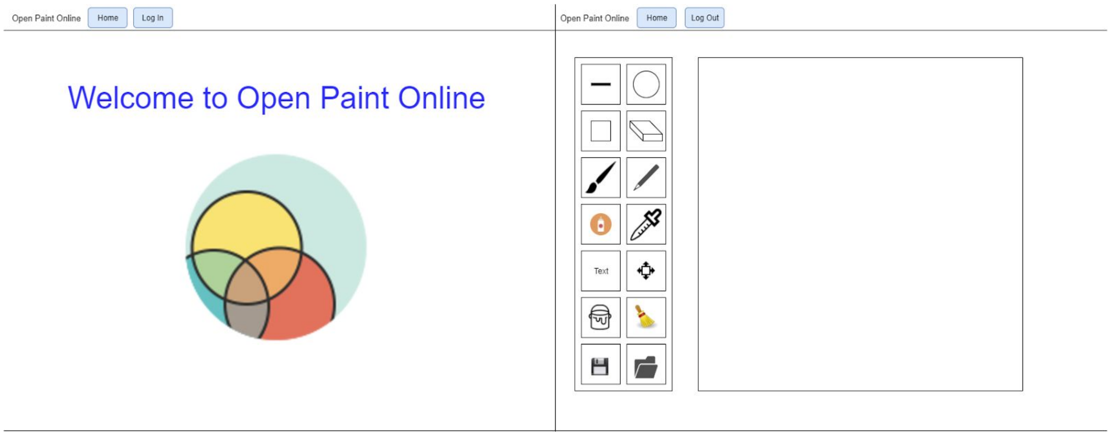
11	Users shall be able to use the shapes tool to make shapes easy.	Must	Complete	Adam
12	Users shall be able to use the fill tool to fill shapes in with a solid color.	Old Must	Dropped for filters	Richard
13	Users shall be able to use the Image filtering tools on the whole canvas.	New Must	Added; Complete	Richard

b. Non-functional requirements

- **Performance**: should run smoothly while maintaining responsiveness.
- **Usability**: should be easily accessible from all devices and simple to use, requiring no installation.
- **Security**: Authentication and authorization should be implemented. The application should keep user data secure and private.
- **Maintainability**: should follow S.O.L.I.D. principles in order to ease flexibility and/or maintainability.
- **Extensibility**: Should support a wide variety of libraries all at the same time.
- **Accessibility**: Users are able to navigate throughout the application and correctly use the capabilities of the app from the context provided.

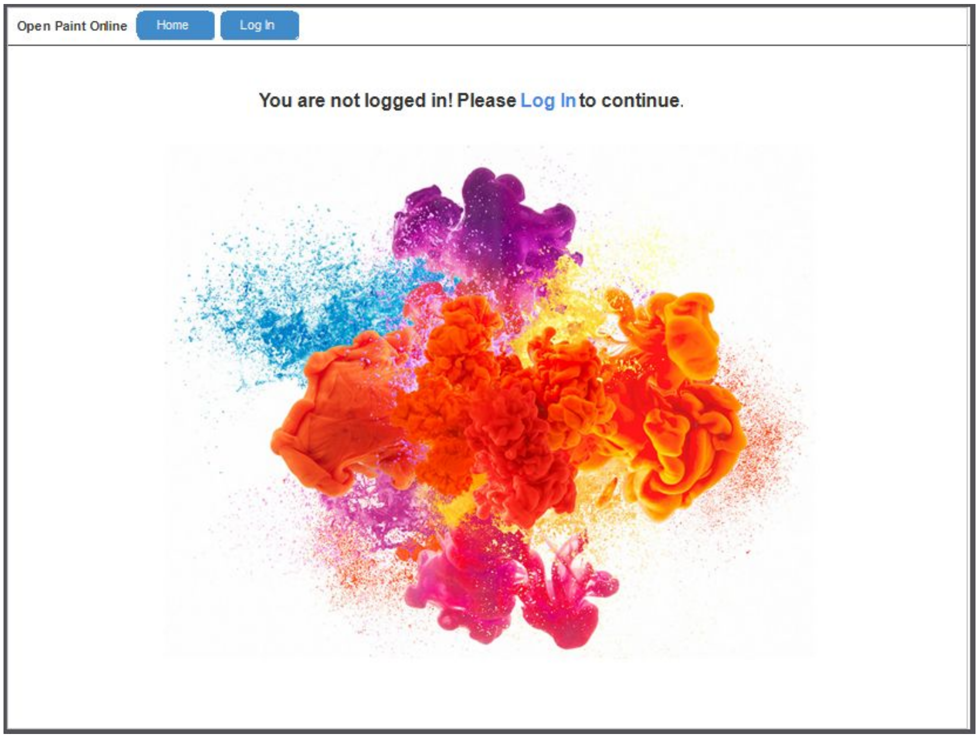
c. Wireframes

Version 1.0:

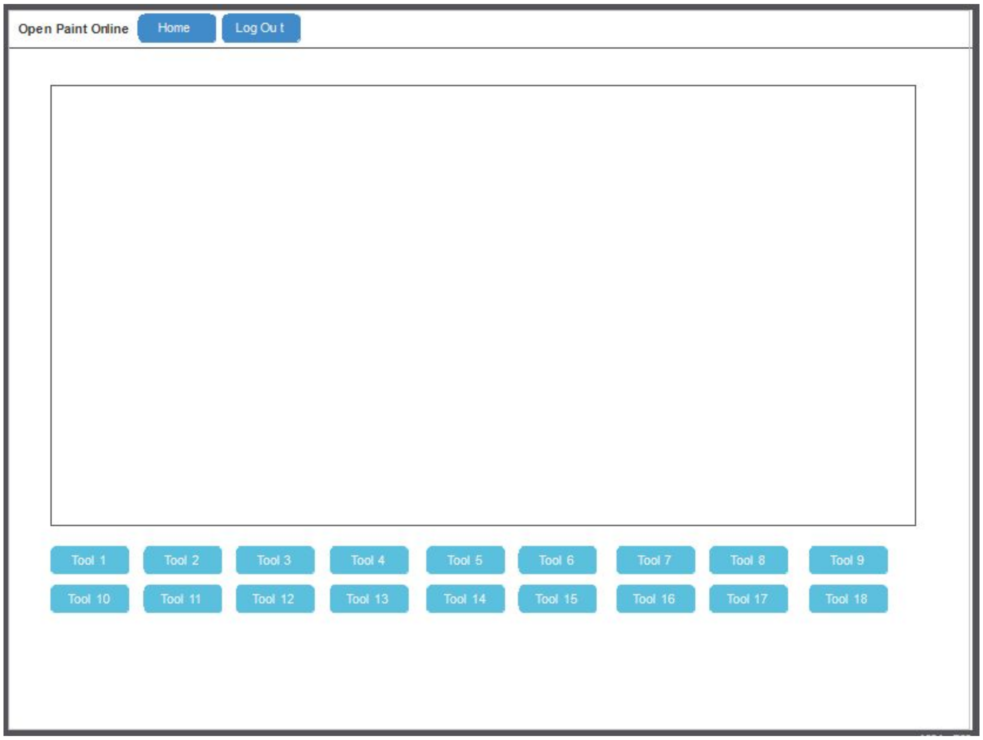


Welcome to Open Paint Online

Version 2.0



You are not logged in! Please Log In to continue.



d. Functional features

- Brush ,Pen, Line, Erase,
  Text, Fill, Brush color, Brush size, Filters,
  Shapes, Save ,Open, Login, Logout.

e. UI (user interface)

- Above are pictures of the User Interface.

f. UX (user experience)

- Open Paint Online is designed to be easy to use. The user is presented with a blank canvas and a variety of tools to choose from. The tools are a series of buttons located below the canvas and can readily be toggled on and off. Some buttons take immediate effect on the canvas, such as clearing the canvas.

g. Execution performance

- All tools now function fast enough so that the user feels that the program is highly responsive.
- Filters: Initial difficulties in performance with filters involved the use of the slow HTML function, putImageData(), which displayed an array of rgba pixel values to the Canvas. This function commonly took more than 10

seconds to execute on a 1000x500 pixel Canvas size. This issue was resolved with the use of Fabric JS functions that utilize the GPU for near immediate response time.

h. Usability

The application is available online from any device with a modern web browser. This includes mobile devices, traditional pc’s, and cross platform Mac, Windows, Android, IOS, Windows Phone, Chrome OS, plus many more.

i. Stability

Strictly speaking code and not IT infrastructure the code uses tested and proven languages (Typescript, JavaScript, CSS, HTML, and Java) as well as tested and proven Frameworks (Angular 4, and Spring Boot). Maven is used for dependency management on the Java backend and Node Package Manager is used for dependency management on the Angular 4 frontend. Both of these allow for libraries to be updated easily making our code scalable and maintainable.

j. Hardware

Users can use any hardware that has a modern web browser. As for IT infrastructure, our project currently uses Amazon’s AWS Ec2 for hosting both the frontend and backend. The MySQL database is located on Amazon’s AWS RDS a relational database service. This infrastructure can be changed very easily to dedicated servers or some other AWS competitor in the future.

k. Maintainability

The code is easy to maintain due to the decision made for both hardware and software. The code is also has better maintainability from using version control, best practices, and documentation. Features can be added independently as code is generally decoupled.

l. Security

This project was built from the ground up with security in mind. It was built around token authentication and authorization. Auth0 is used to secure all backend REST api endpoints. Auth0 was also used on the front end to authenticate users which the backend authorizes using the roles passed in the token. In doing so, the passwords and authentication methods are completely separated from both frontend and backend. Maintaining security such as revoking users, user statistics, and utilizing Auth0's trusted and proven credential storage are just some of the great features we inherit in Open Paint Online.

m. Extensibility

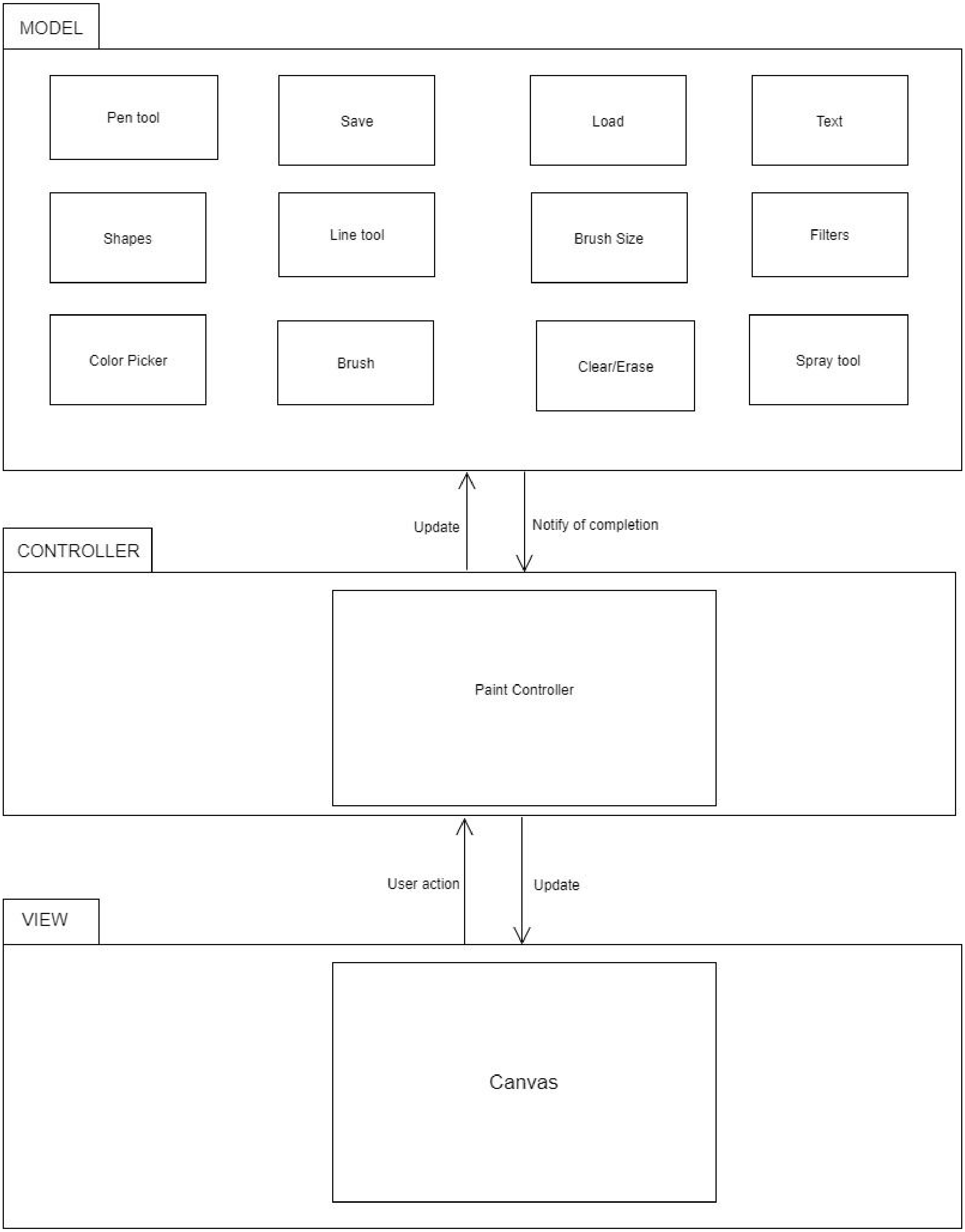
Libraries we use are independent of one another. Both frontend and backend use multiple libraries which are easily updated or replaced without affecting each other.

4. Architecture:

a. Design patterns (+ MVC)

The front end uses Angular 4, which is a model view controller implementation. This allows the view (html) to interact with the controller (typescript) using two way data binding. The model results in two-way data binding used by both the controller and the view to communicate with each other. The backend also uses a form of model view controller where the view is JSON and the controller is the Rest Controller classes. The model is the MySQL database which is accessed through Hibernate's object relational mapping.

b. Diagrams (MVC)



5. Coding guidelines:

a. Naming conventions

- **Class name conventions**: camelCase, TypeScript language class name conventions.
- **Variable name conventions**: camelCase, TypeScript language variable name conventions.
- **Function name conventions**: camelCase, TypeScript language function name conventions

b. Usability guides

- **Understandability** (attributes of the software that bear on the users' effort for recognising the logical concept and its applicability)
- **Learnability -** The web application is intuitive and has labeled buttons for the user's understanding.
- **Operability -** The operations work with a quick response time and has buttons to give the user complete control.
- **Attractiveness -** The style and placement of the various feature of the web application are meant to be easily found look appealing.

c. Menu

There are two menus. One is located at the bottom of the canvas and it has a button for each feature for drawing on the canvas, as well as a save and a clear the canvas button. The other menu is located at the top of the canvas and it consists of the login/logout button and a home button.

d. Organization

Each member of the team was to choose 2 features to implement. We ended up doing at least 3 features each. We communicated through Slack and used Github to collaborate on the project and for version control.