Collaborative Cloud Wireframe

(A cloud based wire framing tool)

Arun Malik (*Author*)
Software Engineering Dept.
San Jose State University,
San Jose,California, United States
malikarun86@gmail.com;

Pravin Agrawal (*Author*)
Software Engineering Dept.
San Jose State University,
San Jose,California, United States
agrawalpravin19@gmail.com;

Yaopeng Wu (*Author*)
Software Engineering Dept.
San Jose State University,
San Jose,California, United States
go.gyoho@gmail.com

Uday Mankena (*Author*)
Software Engineering Dept.
San Jose State University,
San Jose,California, United States
udaymankena91@gmail.com

***Abstract*—Our aim is to create a web application that will enable software development teams to create system design artifacts like paper prototypes, wireframe etc that can be shared within team members, customer and other stakeholders to get firsthand feedback on the initial design.**

Keywords—design artifact, online collaboration, paper prototypes, wireframe

I. Introduction

In this day and age of software development, software firms have their presence all over the world. Also, the development teams, customers and other stake holders are not pertained to a country or a region. Teams are globally distributed. So a team may have members from different countries, who may not know each other physically, but work together, towards a common goal. Members of the team need hands on information about the work of other members. So a platform is required for the development teams to share the documents. Our web application solves this problem, by providing provisions for development teams to share and collaborate the design artifacts like wireframe, paper prototype design etc. As it is a web based application, geographically distributed teams can access it and seamlessly collaborate with each other..

II. Scope of the Project

A. Fast, Intuitive and Iterative Prototyping

Simple drag and drop UI elements to the page to go from idea to prototypes in minutes. Since prototypes requires little investment in time and effort, you'll revise your designs more often and refine them sooner.

B. Collaboration

Share a link and your clients and teammates can edit and provide feedback. No more emailing images back and forth. All stakeholders can collaborate irrespective of their location constraints. It also allows multiple stakeholders to edit and collaborate on a single file or document. You don't have to worry about tracking the latest version or who has made what changes.

C. Cloud Storage

Users can crate any number of artifacts on cloud providing easy accessibility from tablets to smartphones, netbooks to desktops.

D. Security

Storing confidential or sensitive information projects is often more secure than storing it locally, especially for businesses. With online storage services, data is encrypted both during transmission and while at rest, ensuring no unauthorized users can access the files

III. Implementation and Technology Stack

When considering Tech Stack, we decided to choose both cutting edge and handful technology. We tried to take advantage of technologies with a compelling use case and obtain practical experience from them. Starting from the top on down to the data layer, follow us through our stack.

Resources : *https://github.com/devmaven/CMPE272Project_WireframeTool*

A. Frontend

Canvas :

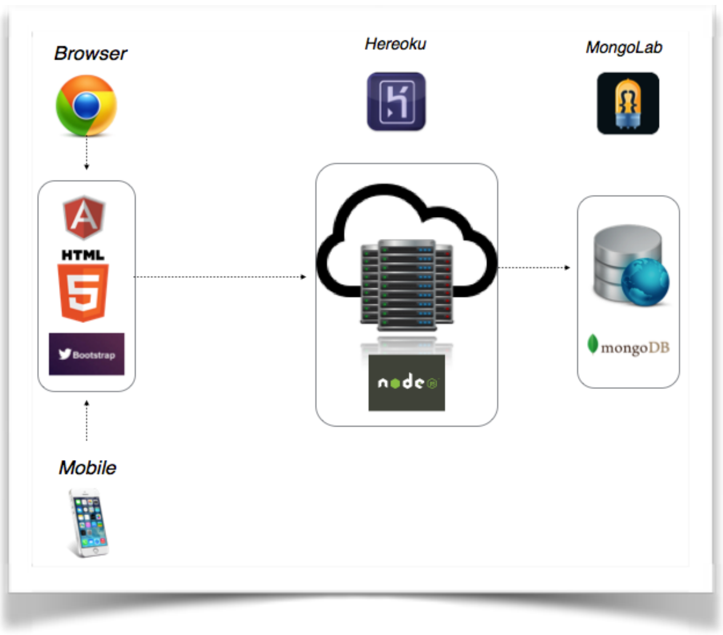
"The canvas element provides scripts with a resolution-dependent bitmap canvas, which can be used for rendering graphs, game graphics, art, or other visual images on the fly."[1]

Since we were developing a free prototyping app, we realized that HTML5's canvas was the most convenient tool for our app. This is the core of our app with greatly extended by Kenetic.js.

KineticJS:

"KineticJS is an HTML5 Canvas JavaScript framework that enables high performance animations, transitions, node nesting, layering, filtering, caching, event handling for desktop and mobile applications, and much more."[2]

We employed the KineticJS canvas library to enhance basic canvas features further. The library of theKineticJS is well documented with comprehensive tutorials; http://

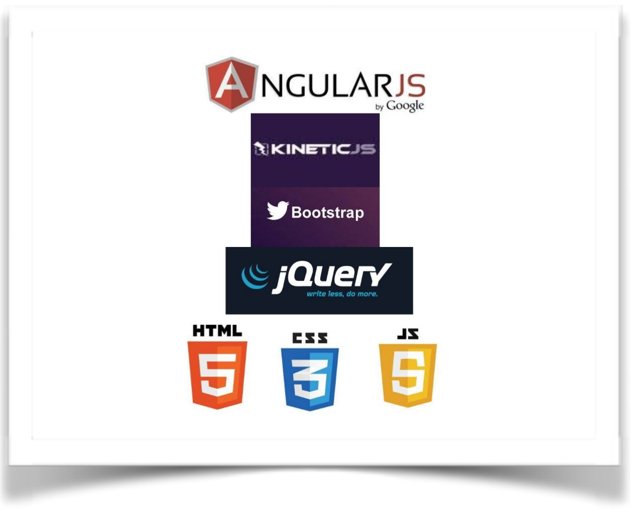


www.html5canvastutorials.com/. The KineticJS provides useful features to add shapes, images, and text into a canvas object very easily and quickly.

AngularJS

"HTML is great for declaring static documents, but it falters when we try to use it for declaring dynamic views in web-applications. AngularJS lets you extend HTML vocabulary for your application. The resulting environment is extraordinarily expressive, readable, and quick to develop."[3]

In order to properly design a MVC based application, we



introduced AngularJS. AngularJS is a frontend web framework to assist in rapid app development. It helped us organize our code properly while adding more complexity. Moreover, this helped us discover the power of the KineticJS using with AngularJS. [4]

B. Application Layer

NodeJS:

"Node.js is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices."[5]

For the backend, we decided to use NodeJS as our server implementation. Our motivation is largely came from curiosity. Node is a recently emerging new backend stack. Moreover, as a backend with the goal of rapid prototyping, node was suitable for our app.[6]

Node application architectures are event driven which allows node to scale to handle many network events. Since our application needs to communicate to the backend for authentication, save/restore operations and collaboration, which largely involves I/O requests, latency is crucial factor. With its fast runtime environment, event driven, Node's fully satisfies our application's needs.

ExpressJS:

"Express is a minimal and flexible node.js web application framework, providing a robust set of features for building single and multi-page, and hybrid web applications." [7]

Express is a well-known web framework for node.js. It is not strongly opinionated and thus easy to start with. Express is originally built upon Connect.js, a middleware extends node.js, but obviously it now exceeds the abilities of Connect.js. Express supports many middleware to enhance functionality of our application. In particular, we used the templating engine module ejs so that we can easily support combining templates with JSON for generating dynamic page. Express further supports route management which makes GET or POST requests executed smoothly.

PassportJS:

"Passport is authentication middleware for Node.js. Extremely flexible and modular, Passport can be unobtrusively dropped in to any Express-based web application." [8]

One of the key component of our app is PassportJS. It is a middleware that makes us handle authentication easily. PassportJS takes an authentication strategy from the frontend such as username / password, and adds a user object to every request. This is incredibly useful when implementing the express' session handler. The significant convenience is that once a user is authenticated the passport middleware adds user context to every request. PassportJS also worked very well as a middleware with user data models, which are abstracted in higher layers and translated to the database with Mongoose.

Mongoose:

"Mongoose provides a straight-forward, schema-based solution to modeling your application data and includes built-in type casting, validation, query building, business logic hooks and more, out of the box."[9]

Mongoose enhances the functionality of Mongodb. It provides support for caching individual data documents on the app side and adds some schema to the largely free-form mongodb. Mongoose allowed us to take the raw document model of mongodb and organize it to match the MVC of our frontend.

Heroku:

"Write apps in your language – we support Ruby, Node.js, Clojure, Java, Python and Scala. Use technologies you already love and discover new ones through our Add-ons platform. Add Postgres, New Relic, Papertrail, Redis, Mailgun, or pick from dozens of other cloud services. Deploy instantly from the command line using Git."[10]

We used Heroku as our PaaS provider because it's free tier functionality matched our demand. By connecting our app in GitHub repository with Heroku, we can deploy our app to Heroku immediately and directly through single command. Since our app involves many UI features, it was important for us to test some features and monitor the progress on live. Furethermore, Heroku supports plugins for integration which enabled mongodb support to our heroku instance.

C. Data Layer

MongoDB:

Passport
Simple, unobtrusive authentication for Node.js.
bcrypt.js bcrypt in plain JavaScript
EJS
<% Embedded JavaScript %>
express
High performance, high class web development for Node.js
mongoose

"MongoDB (from "humongous") is an open-source document database, and the leading NoSQL database. Written in C++, MongoDB features: Document-Oriented Storage, Full Index Support, Replication & High Availability, Auto-Sharding, Querying, Fast In-Place Updates, Map/Reduce, Flexible aggregation and data processing, GridFS.[11]

MongoDB is a "Document" oriented database. Instead of using tables to store data, it applies a new concept, which interpret data as collections of documents. This abstraction introduced more flexible schema style and thus enables each document be more complex than an traditional single table row. Mongo aims to have most of the reliability of traditional RMDBs while at the same time being more complex than simpler NoSQL solutions such as MEMCACHED or other Key/Value solutions. Another useful feature we observed was the data format. MongoDB stores data in BSON format, binary of JSON, which enables our app to communicate with single language, JavaScript, through all layers.

MongoLabs:

MongoLab is a fully-managed cloud database service featuring highly-available MongoDB databases, automated backups, web-based tools, 24/7 monitoring, and expert support. [12]

Mongolabs was one of two Mongo hosting solutions supported by Heroku. MongoLabs also has a free "sandbox" tier which is meant for prototyping. This allowed us to deploy our entire app including our data persistence(database) layer to "the cloud".

IV. Bottlenecks

When using a new language and technology, there is a steep learning curve, especially when switching from a Object Oriented language to a mostly functional language with closure and callbacks(JavaScript) . As Dijkstra famously said "C++ damages the brain ..." so it was definitely worth it, but sometimes wrapping one's brain around the alternative design was boggling.

Kinetic.js does not properly serialize data in some cases(images and functions). We traversed this limitation by manually loading images and functions after the kinetic restore function had executed.

V. Future Scope

A. Collaboration

Currently collaboration is a limited. In the future we could add live collaboration offering by Mozilla's TogetherJS.[13]

Together.JS also support live audio chat and text chat.

B. Comments

We implemented the comment feature by letting a user directly posting a sticky note into the canvas. However, this will take a good portion of space and interrupt the work. The best way to overcome this problem is to introduce a threaded comment section separate from the canvas object.

C. OAUTH (Passport)

Passport.js support more login strategies such as Facebook, Google or Twitter. Nowadays, Oauth logins are very common. So It might improve user interaction if we were to support these login systems.

D. User Interface

At this time the User Interface is very functional but more time can be spent by adding CSS (transitions and decorations) as well as improving upon the Kenetic Canvas decorations.

Some specific areas for improvement might be making the buttons more attractive. As well as adding a global color scheme. We also did not spend a lot of time picking fonts specialized.

E. Image Upload

The image upload feature is common for most Mockup apps. However, because of the limited storage space for Mongolab sandbox in Heroku, we did not include the function. In the future, we should find a way to circumvent this limitation. One way is to use GridFS supported by MongoDB, in order to do so we would be to convert images directly into data objects and convert them on the fly on the frontend but this is a somewhat complex way to solve the problem and will also allow users to quickly ballon past mongo's document size limit.

F. Mobile App

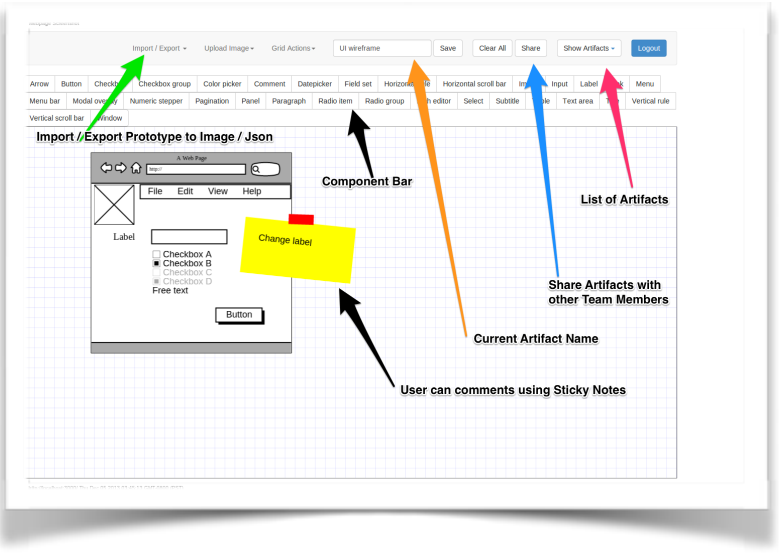
Since the market for mobile app is growing very rapidly, the next step for our app is to support mobile version. Although our app is responsive for all devices, there is a definite speed and market penetration advantage to a presence in Mobile application stores. One potential stopgap would be to use Adobe's PhoneGap to create application containers for a web view. This solution could use the same API/Backend that we created for our app.

G. Figures and Tables

I. APPLICATION SCREENSHOT

ACKNOWLEDGMENT

We are very thankful to Prof. Rakesh Ranjan and Teaching Assistant (Krunal Vohra and Harnika Sahni) for their assistance and guidance in helping us to materialize this project



CONCLUSION

Our application makes it easy and quick for software development team, customers and other stakeholders involved to create, share and seamlessly collaborate different design artifacts like wire-frames, paper prototype, mock-up screens with each other.

REFERENCES

1. The canvas element. Living Standard - Last Updated 3 December 2013, from http://www.whatwg.org/specs/web-apps/current-work/multipage/the-canvas-element.html#the-canvas-element.
2. KinectJs , from http://kineticjs.com/
3. Angular Js, from http://angularjs.org/
4. Agido-Mockups, from https://github.com/it-crowd/agido-mockups
5. Node Js, from http://nodejs.org/
6. Blog on Express Js, from http://blog.mediumequalsmessage.com/understanding-expressjs-and-nodejs-as-a-medium-for-prototyping
7. Express Js, from http://expressjs.com/
8. Passport Js, from http://passportjs.org/
9. Mongoose Js, from http://mongoosejs.com/
10. Heroku, from https://www.heroku.com/
11. MongoDb, from http://www.mongodb.org
12. MongoLab, from https://mongolab.com/company
13. Together Js, from https://togetherjs.com/