ITWS 4500
Team CAATY
Project Proposal

Vercipe

There is a better recipe

The Team

Anjin Lima
Chelsea Carlson
Andrew Leaf
Yuze Ma
Thomas MacIntosh

Project Summary

Finding a quality recipe can be very challenging. Taking into account the number of calories for a recipe, the many options for ingredients and the time it takes to cook, it is often easier to ignore the effort of cooking and go for takeout instead. This project is meant to make the process easier by consolidating recipes into an easy to use site. The user can then search through recipes to find the types that they want as well as allowing them to update the recipes if they find a better mix of ingredients or need to change how long to cook it for. All copies of that recipe will be saved.

Users

There are many users for this program. People who have been cooking for years and have their own recipes can add them in to the system and enable faster searching. Users who love experimenting with existing recipes can use the editing feature to keep track of changes they've made to each recipe. Doing this, allows those users to step back in time if they like an older recipe or want to experiment in a different direction. New cooks would also benefit from this system, as more recipes are being found online, this system can be used to consolidate 'good' recipes into a single point of access. Vercipe will move with the users, giving them access to their own recipe books without having the weight associated with those recipe books.

Stakeholders

As team members, we are the closest stakeholders to the project. Because of this the success of the project generates the value. Other stakeholders holding this same value are Peter Fox, the teacher and Rahul Prajapati, the TA.
The users are also stakeholders. A working site for them means that they can work with new recipes and experiment with their culinary skills. If they change a recipe, they'll be able to see it in their version history so that they can cook it again. Keeping a working site is beneficial for them. Their value lies in the ability to carry a potentially massive library of recipes with them as they travel.

Technologies

Frontend: Angular 5
Angular will be the primary front-end technology to display the HTML. The data will be filtered through the Angular, via client requests.

Backend: ExpressJS, Docker, Elasticsearch (if time allows)
Based on the apache server, Express will be the go-between from the AngularJS front end and MongoDB backend with a restful API.

Database: MongoDB
A simple database system meant to hold several different aspects of the data within the site.

Functional Requirements

Requirement	Details	Priority
Use the MEAN stack	MongoDB, Angular , ExpressJS	Must
Mobile Phone integration	This project has to be available for use on a mobile device, as people should be able to quickly access a recipe on the go	Must
Ability to edit the recipe fields	As the consumer experiments and finds better versions of the recipe, they should be able to add changes to those recipes	Must
View older versions of the recipe	Sometimes a user may want to share the original recipe or start a new	Must

	experimental path, in order to do this the user must be able to view the original recipe or several steps back in the recipe versions	
Must be able to create account	Create a login name and password to save data on the site	Must
Users can sign in	Using already created accounts, the user should be able to access further features of the site	Must
Works on a variety of browsers	Chrome, Firefox, Internet Explorer, Safari	Must
Fork a recipe	Allow user to copy a resume and edit it	Must
Update a recipe	Make change to the recipe	Must
Searching	Use Elasticsearch to allow user to search for recipes by ingredients or names	Should

Non-Functional Requirements

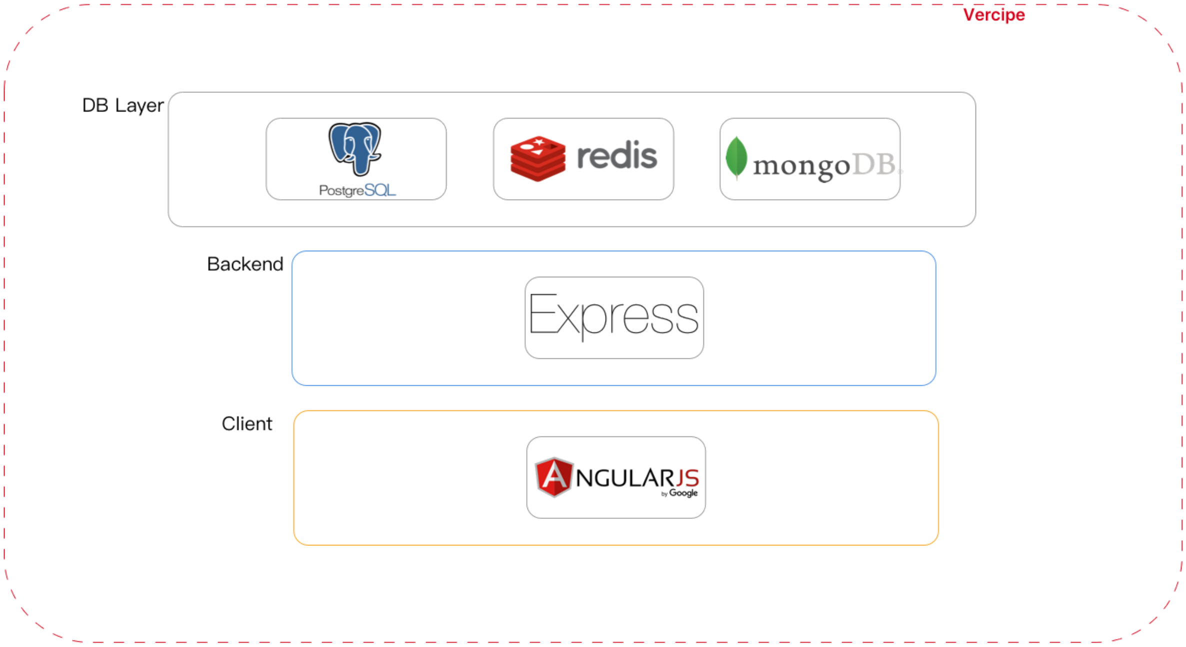
Requirement	Details	Priority
Home page is accessible from all pages	A home link should be on every different page to ease transition	Must
Webpage must load when the correct URL is used	If a user enters the correct URL in a browser, our project webpage should load	Must
Intuitive navigation	Users should clearly be able to determine what each page will do	Must
Database integrity is upheld	Databases should be able to lock in data and not become fragmented	Must

Client-side changes should translate to server-side changes	If the user changes a piece of data (like a password), then the server should properly reflect that	Must
Code is easily understandable by other users	If the code is passed to another team of developers, they should be able to easily pick up what is going on in the code	Must
Loads quickly	< 5 second load time	Should
Minimal navigation	Very few pages (perhaps a single page project)	Should
Many options for each part of sandwich	The more options the better for users and analytical purposes	Should

Project Schedule

We use the tool called Zenhub implemented inside github to schedule our development plan. We choose to do so because Zenhub can help us practice agile development methodology.

Wireframe Examples


Vercipe
DB Layer
PostgreSQL
redis
mongoDB
Backend
Express
Client
ANGULARJS
by Google