|  |
| --- |
| Coffee Mate |
| Web Application |
| John Hodmon  Student Number 98163813 |

Contents

[Overview 2](#_Toc447365135)

[User Authentication 2](#_Toc447365136)

[Add a Coffee 2](#_Toc447365137)

[Editing Coffees 3](#_Toc447365138)

[View All Coffees 4](#_Toc447365139)

[View Favourite List 4](#_Toc447365140)

[Architecture 5](#_Toc447365141)

[Authentication 6](#_Toc447365142)

[References 7](#_Toc447365143)

[Figure 1: google sign in 2](#_Toc447365144)

[Figure 2: user signed in 2](#_Toc447365145)

[Figure 3: adding a coffee 3](#_Toc447365146)

[Figure 4: editing coffees 4](#_Toc447365147)

[Figure 5: all coffees 4](#_Toc447365148)

[Figure 6: favourites list 5](#_Toc447365149)

[Figure 7: architecture 6](#_Toc447365150)

# Overview

The application is hosted on Heroku at the following address; <http://rhubarb-cobbler-99765.herokuapp.com/>

## User Authentication

Authentication is achieved by using Google sign-in (Google.com, 2016). Users can log in with their Google account. Users name and Google profile picture then appear beside each coffee they add to the application.

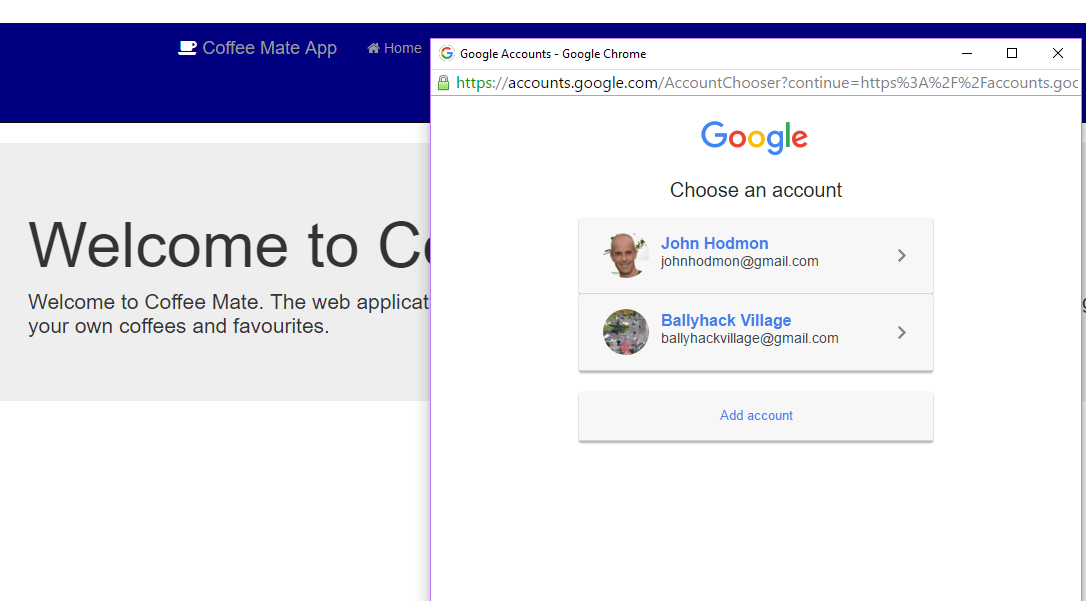


Figure 1: google sign in

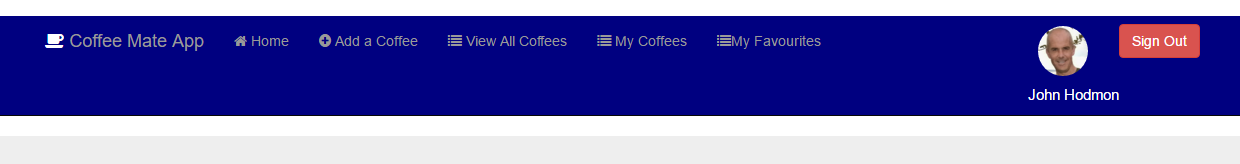


Figure 2: user signed in

## Add a Coffee

Once signed in users can add a coffee that they have tasted. The coffee details include name, the name of the coffee shop where it was bought and the price. When they add a coffee they are taken to a list of the coffees where they can edit or delete their coffees, see below. Coffees added are seen by all users of the application but other a user can only edit coffees added by him or her.

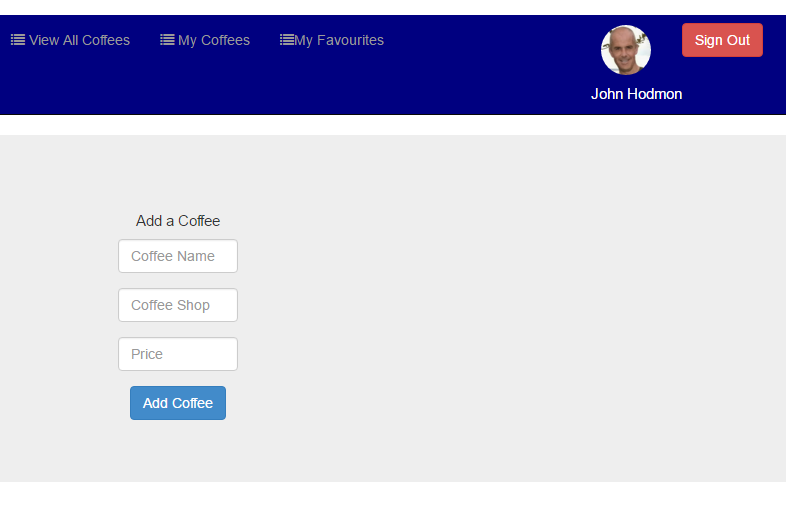


Figure 3: adding a coffee

## Editing Coffees

Users can view a list containing their own coffees by selecting ‘My Coffees’ in the navigation bar. Users can change the rating they give to a coffee by selecting the appropriate number of stars beside that coffee in the list. To edit name, coffee shop name or price users select the edit button and are brought to a form prefilled with the current values of the coffee. To delete a coffee users can select the delete button beside that coffee in the list. Users have the option to edit and delete only the coffees they have added, not other users coffees.

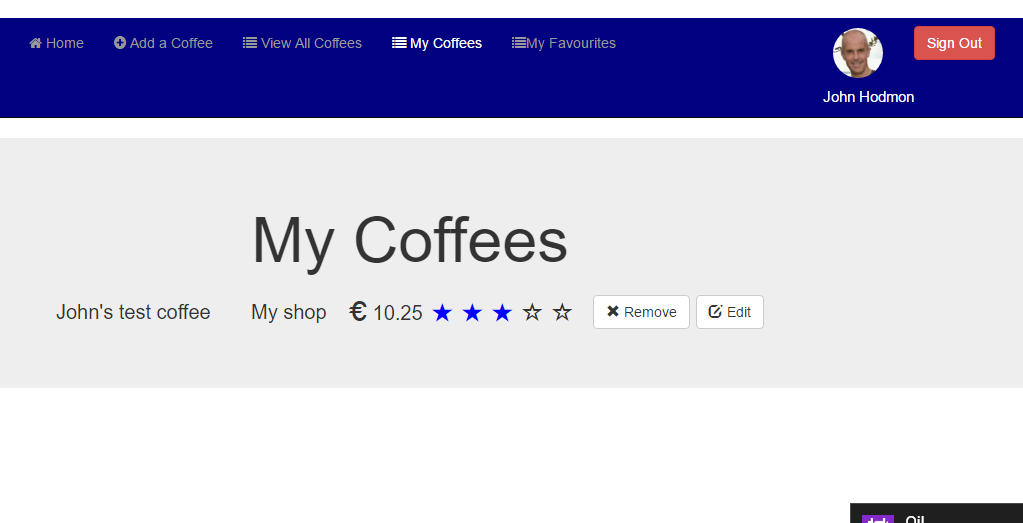


Figure 4: editing coffees

## View All Coffees

Users can see all coffees added by all users by selecting “View All Coffees” in the navigation bar. The coffee details along with the users’ name and profile picture appear in the list. On this screen users can ‘bookmark’ any coffee by selecting the star to the left of that coffee. When a coffee is bookmarked the star colour turns gold and the coffee is added to the signed in user’s favourite list (see below). To remove a coffee from his or her favourite lists the gold star is elected again, the gold colour is removed and the coffee is removed from the favourite list.

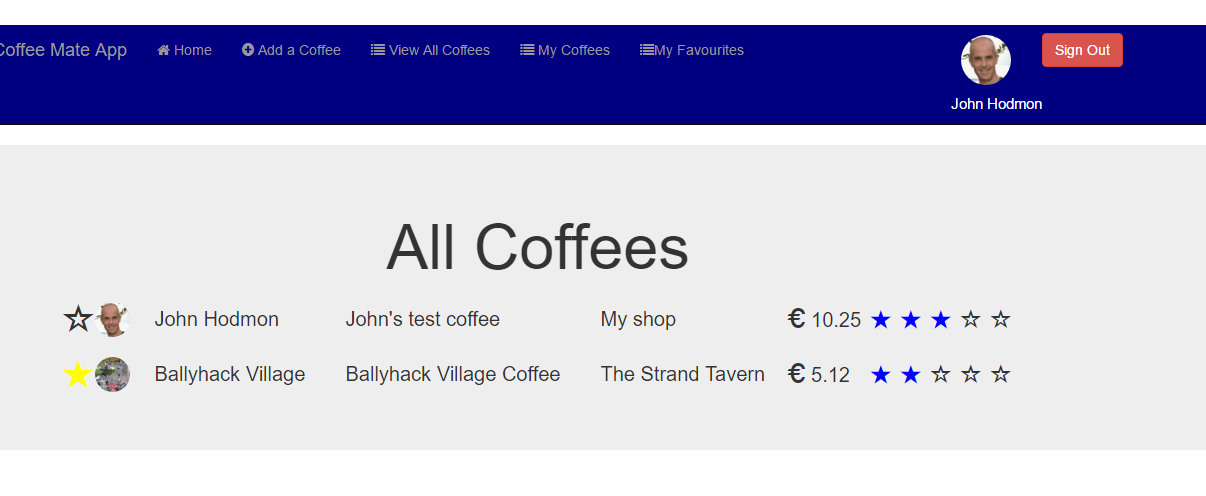


Figure 5: all coffees

## View Favourite List

Users can see their list of favourites by selecting “My Favourites” from the navigation bar. Favourites added as explained above are seen on this screen. Obviously if a coffee is removed from the database by the user who added it that coffee is also removed from the favourite lists of all users.

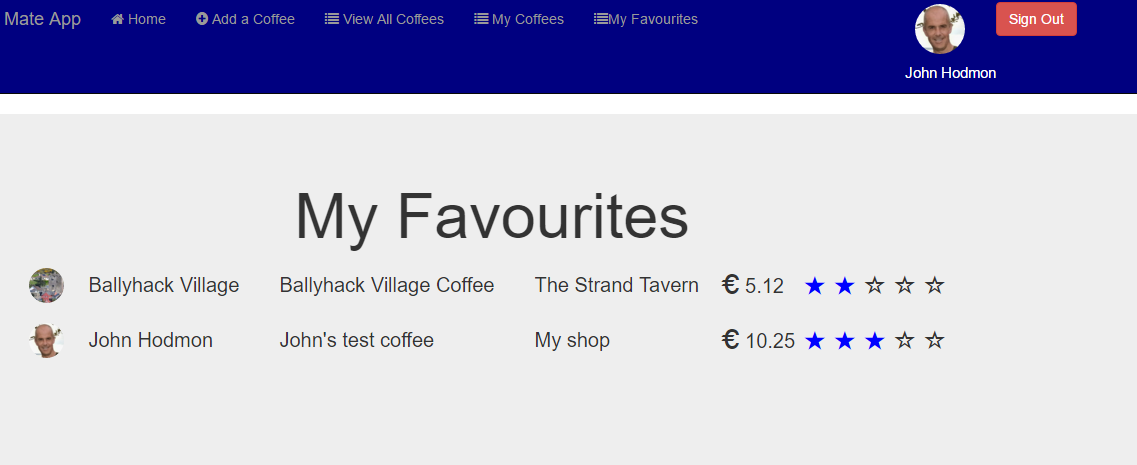


Figure 6: favourites list

# Architecture

Coffee Mate is a node.js application built on the Express framework. The data is stored in a MongoDB database, operations on the data are by way of the Node package Mongoose.

Two mongoose models that map to MongoDB collections are set up, one for coffees and one to store all users’ favourites. The ‘Favourites’ model simply stores the email address of the user who 'owns' the favourite and the coffee itself. The coffee model manages coffee name, coffee shop name, price and rating for each coffee.

Angular.js is combined with HTML and CSS to provide the browser views. Angular is used to dynamically display the lists of coffees, deleting them and adding them appropriately as the user interacts with the application. The update of the display when the user “bookmarks” a favourite coffee or increases the rating by clicking on the stars are also handled by Angular controller classes

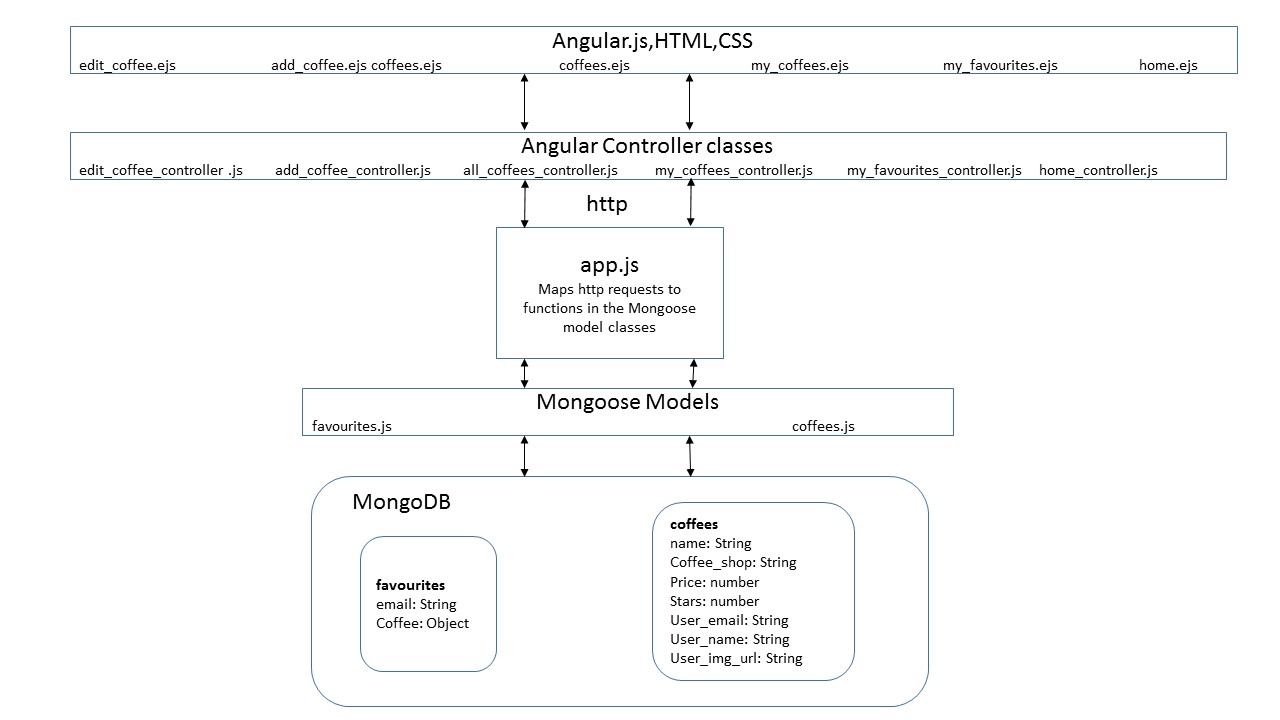


Figure : architecture

# Authentication

Authentication is handled by Google Identity Platform (Google, 2016). A client id for the Coffee Mate application was retrieved from the Google developer console. For testing the local host URL was registered with Google. When the application was deployed to Heroku the Heroku URL was registered also. The combination of client id and registered URL allows the application make authentication requests to the Google servers.

The client ID is specified in using metadata in the head of the applications index page. A sign-in button was created in the navigation bar using Googles styling guidelines. The authentication itself is handled in the Angular controller class “home\_view\_controller.js”. When authentication successful the user’s Google profile information (email, name and profile picture url) are retrieved and stored in an angular $scope object.

Angular is used to dynamically hide or show the sign-in button, sign-out button, and user profile information in the navigation bar depending in sign-in status. Angular is also used to control the visibility of the pages depending on sign-in status

# References

Google.com (2016), ‘Google Sign-In for Websites’, available: https://developers.google.com/identity/sign-in/web/ [accessed 17 Mar 2016]