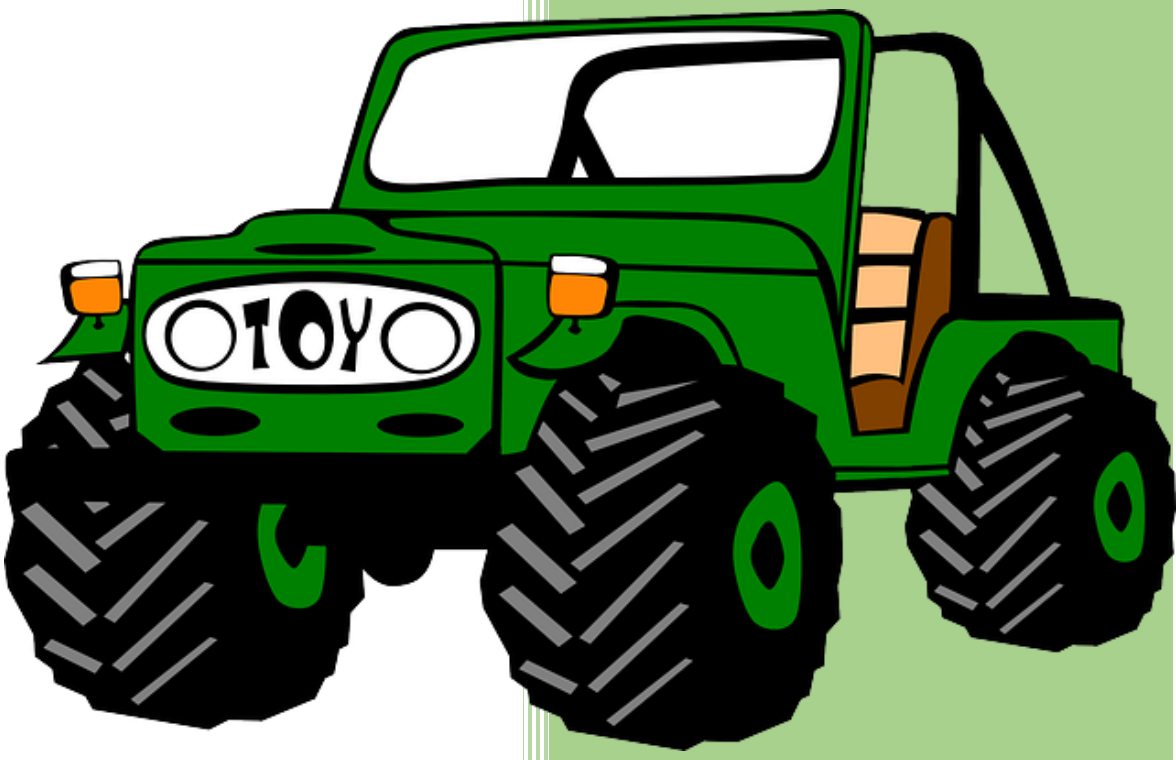
2016

WebCars



Mulkern & Goncalves

COMP 4620

2/1/2016

Contents

Project Goals

WebCars is a web application that allows users to view and post car advertisements (ads) in a simplified and straightforward way. The goal of the project is to develop WebCars in such way that the users can focus on viewing the available cars without getting distracted by other advertisements usually located on the side of the page.

Feature Descriptions

Alpha Features

The core feature of Webcars is being able to search for a desired car and to post a car for sale. WebCars is designed to make searching and posting ads as simple and quick as possible. Figure 1 shows the workflow of Webcars and how it can be used.

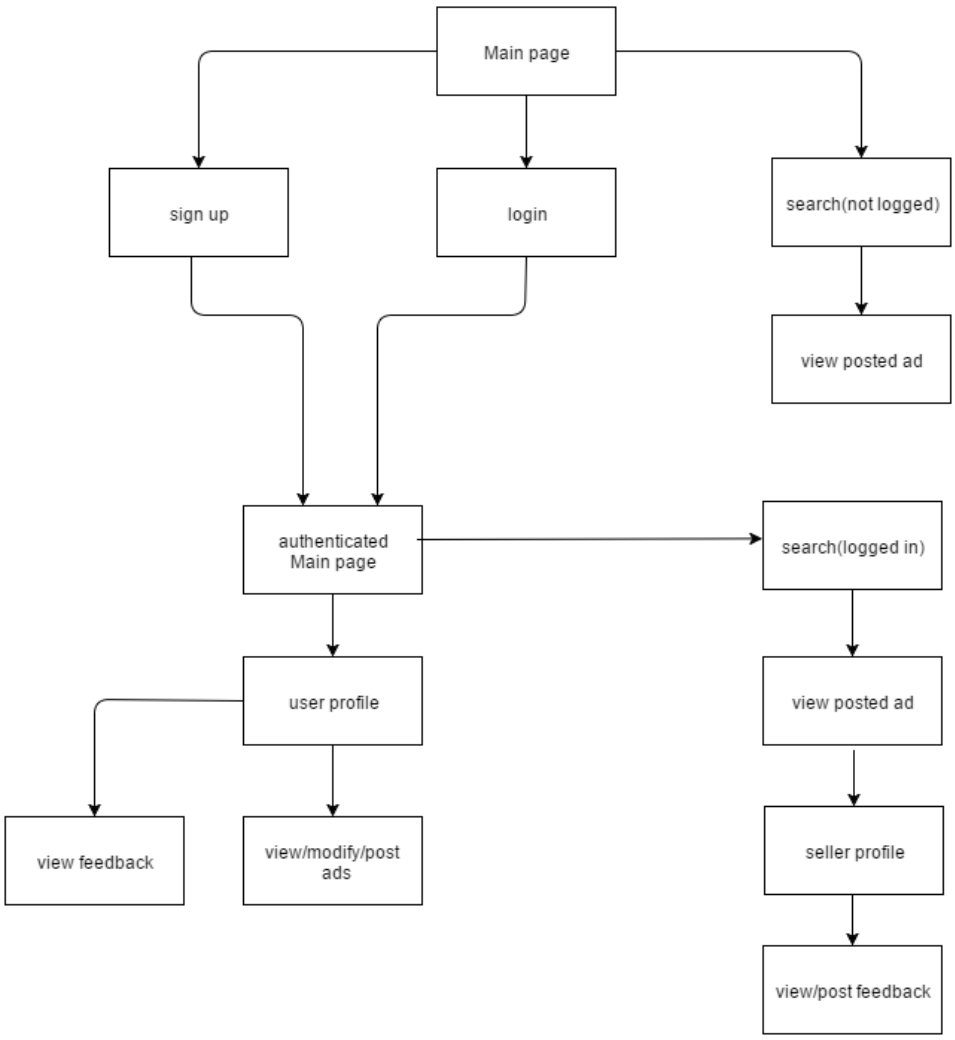


FIGURE 1 – FRONTEND WORKFLOW DIAGRAM.

To search for car ads, new users do not require to sign up for an account. However, to make a purchase, they will need an account to view the location of the car as well as the seller's contact information. An account is also required to post an ad to WebCars. Figure 2 is a sample of an account sign-up form.

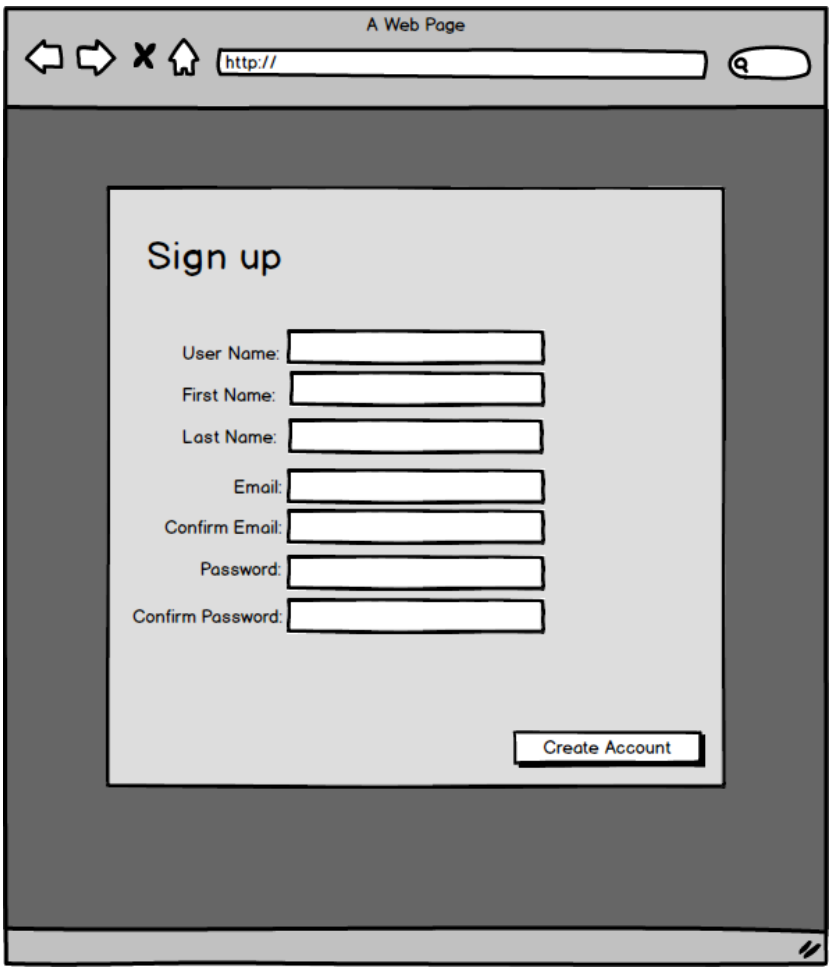


FIGURE 2 - SIGN UP FORM FOR NEW USERS

Once a user created an account, the user will be able to log into the account as shown in Figure 3. After logging in, users can view their profile and read any feedback left by other users as depicted in Figure 4. Upon logging in, the user will be redirected to the main page where the most recent car postings populating the page as seen in Figure 5. Users will also be able to see profile pages of other users and leave feedback for them if they have done business with them, as well as see other ads that the user has posted. From the main page the user will be able to view an ad to see more information about the car as seen in Figure 6.

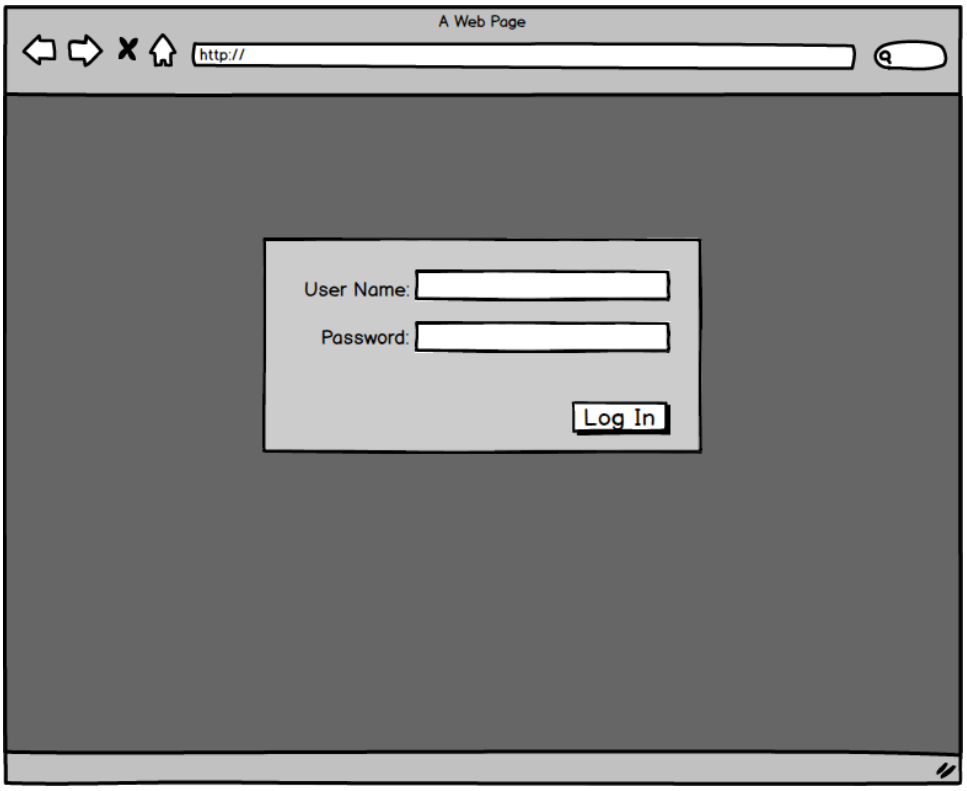


FIGURE 3 - USER LOG IN FOR ALREADY EXISTING USERS.

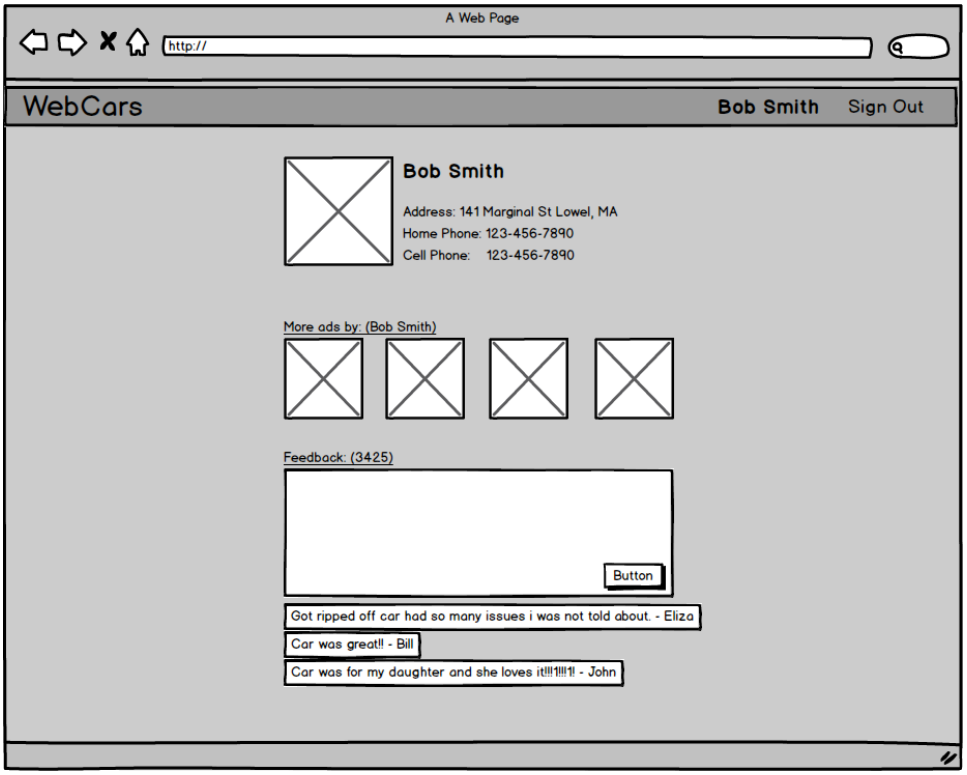


FIGURE 4 - USER PROFILE PAGE WITH CONTACT INFORMATION AND FEEDBACK.

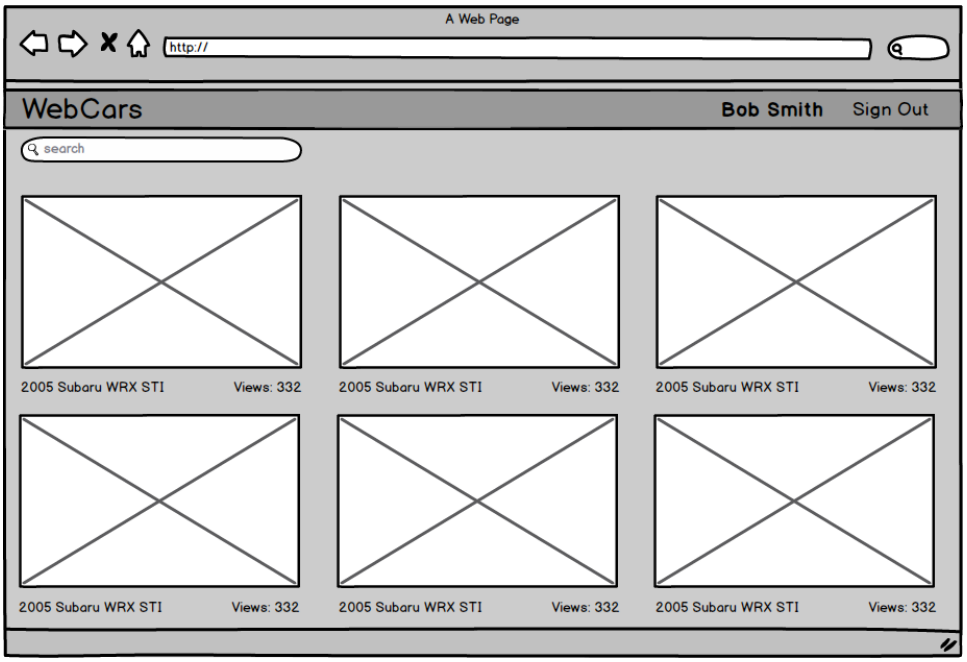


FIGURE 5 - MAIN PAGE LAYOUT UI WITH RECENT CARS ADDED POPULATING THE PAGE.

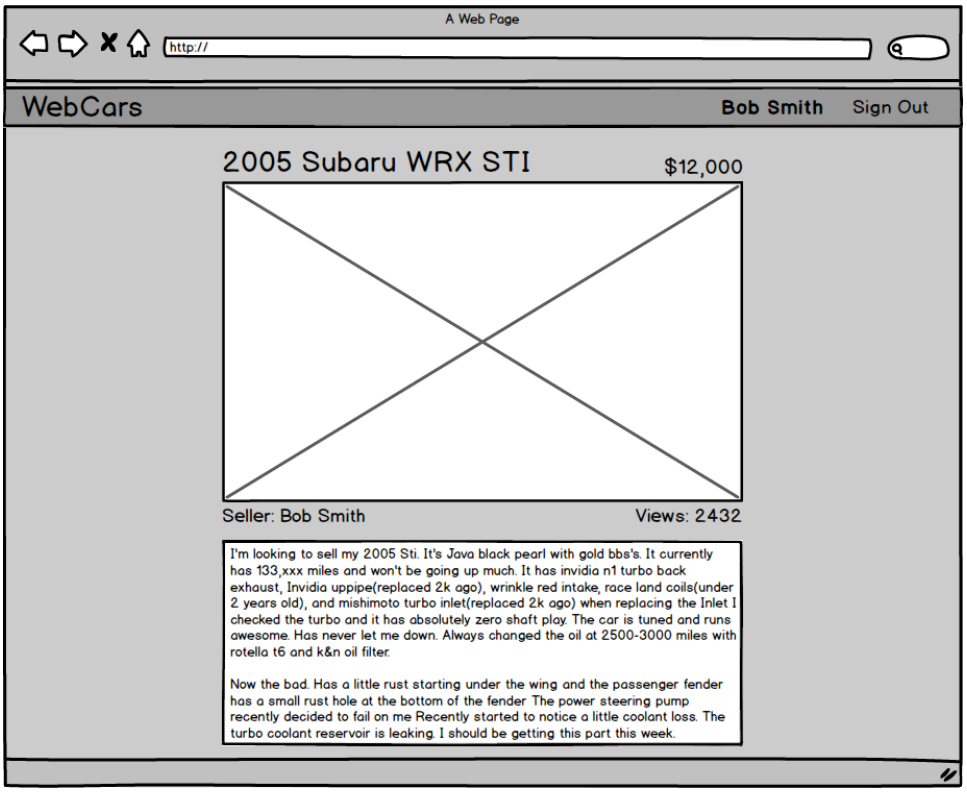


FIGURE 6 - CAR AD UI LAYOUT.

Beta Features

The features of WebCars are implemented in the following order if time permits:

- Category bar— allow users to filter data with a control bar.
- Auto-complete— the list of possible candidates that will come from the MongoDB.
- Message communication—a registered user can communicate with other registered users regarding the ads.
- Report System— allow users to report a scam or inappropriate content.
- CSS Mobile version—the application will choose a mobile or desktop CSS file.

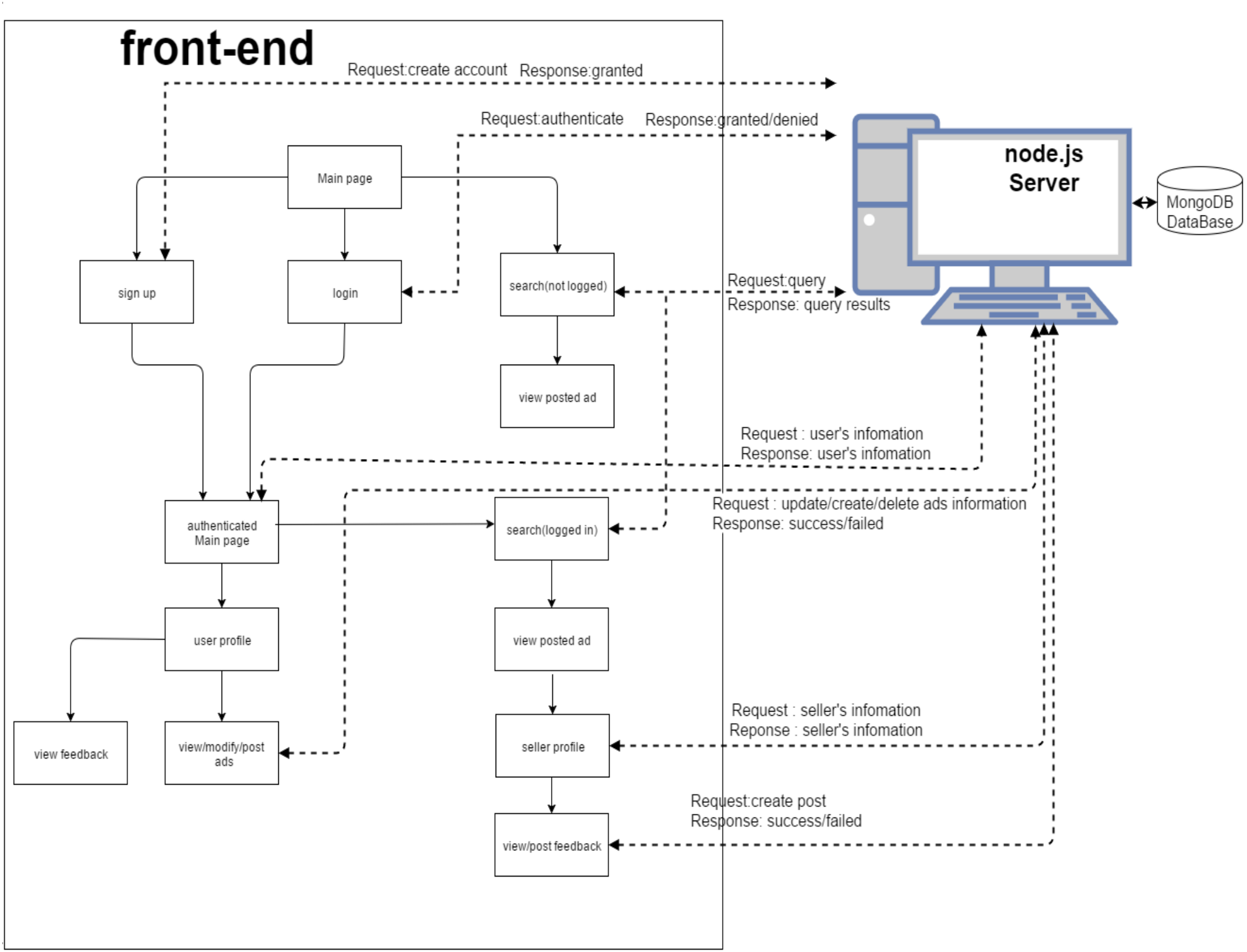


FIGURE 7 – THE CORE DESIGN OF THE APPLICATION SHOWING THE INTEGRATION OF THE UI, SERVER AND DATABASE.

Component Details

UI

The UI will be implemented using Angular.js, HTML5, JavaScript and CSS3 with the Bootstrap3 framework. The UI will communicate with the Node.js and Express server application via HTTP CRUD Methods. The CRUD Methods will be exchanging JSON objects with the server and client. The UI pages will be manipulated by the JSON objects received from the server with Angular.js controllers.

DATABASE

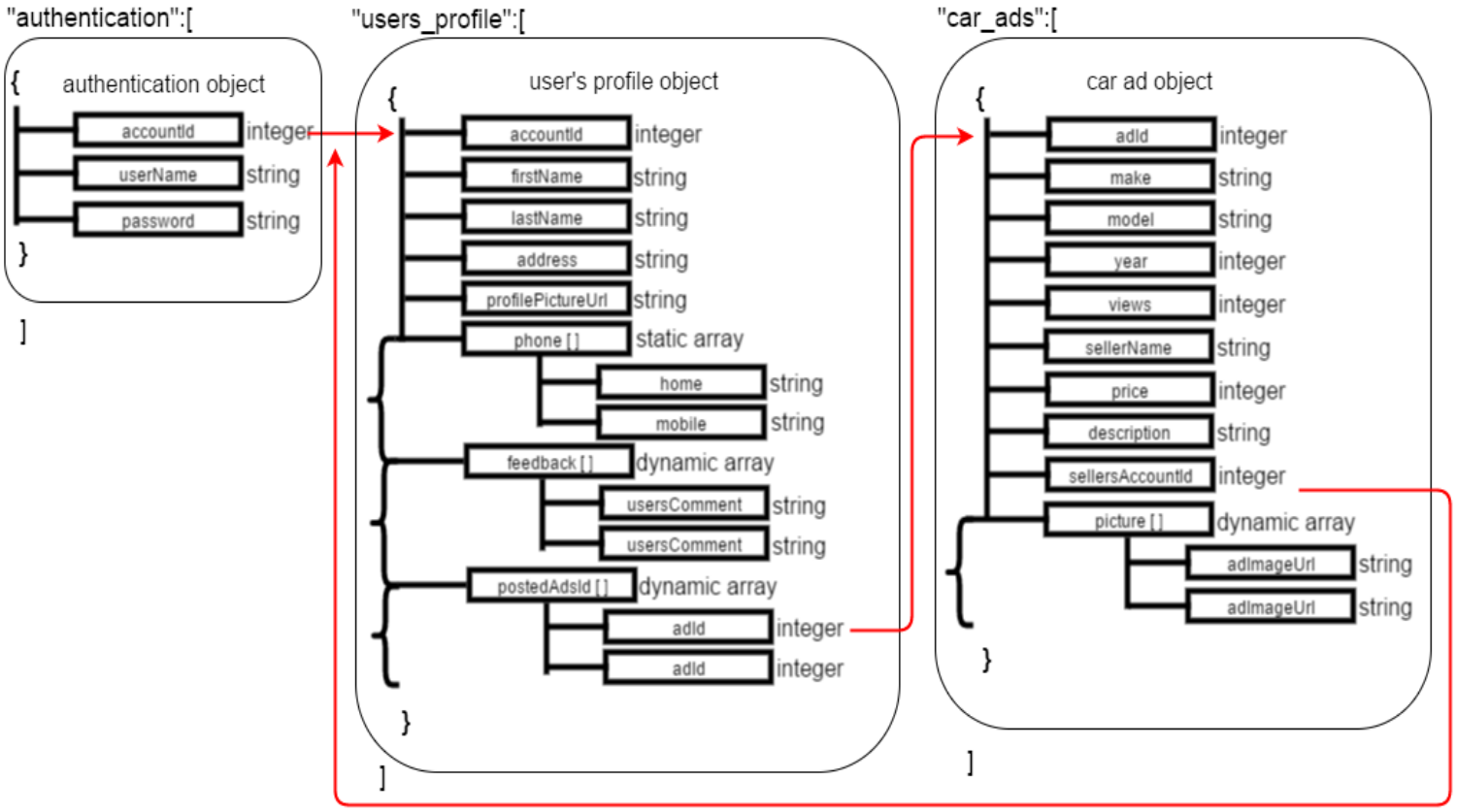


FIGURE 8 - AN INTENDED DESIGN OF THE MONGODB

MongoDB, the document-oriented database, will be organized to hold a collection of three objects as shown in Figure 8:

- Authentication Object Collection– Every element in this object holds an existing user's credentials information. The Login content will access the object with permission to read only. Create Account content will access the object with permissions to modify and read.

- User's Profile Object Collection–Every element in this object will hold the user's data in the application. The Profile content will access the object with permission to read only. Create Account content will access the object with permissions to read and modify.

- Car Ads Object Collection– Every element in this object holds posted car ad data. The Search content will access the object with permission read only. Create/Modify Ads content can access the object with permissions to modify and read. The Search content will be able to query this collection based on conditions.

SERVER

Node.js API will host the application, and JavaScript will be used as the server-side language. Express and JavaScript will process the client CRUD requests and manipulate the MongoDB to send a response back to the client. In Figure 7 the dashed lines represents some examples of the request and response of the communications between Express and Angular API. Figure 7 represents the total integration of the WebCars application using the MVC pattern. The front-end will be controlled by Angular.js making requests to the Express API. The Express API sends a response back to the Angular.js controllers. The Angular.js controller processes the template with the data response from Express API server.

Sources-

1 "JQuery." JQuery. N.p., n.d. Web. 1 Feb. 2016. <http://jquery.com/>
2 "Node.js." Node.js. N.p., n.d. Web. 1 Feb. 2016. <https://nodejs.org/en/>
3 "Bootstrap · The World's Most Popular Mobile-first and Responsive Front-end Framework." Bootstrap · The World's Most Popular Mobile-first and Responsive Front-end Framework. N.p., n.d. Web. 1 Feb. 2016. <http://getbootstrap.com/>
4 "MongoDB 3.2." MongoDB for GIANT Ideas. N.p., n.d. Web. 1 Feb. 2016. https://www.mongodb.org/

User Descriptions

Our target audience will be males and females between ages 16 and up who are looking to find a good deal on a car or looking to sell a car. Users in the lower end of this age group will probably be looking for a first car which would be easier to get used from another person rather than a dealership that marks up prices. Other users not in the lower age group may be looking for a car without all the paperwork and waiting that comes along with using a dealership. To help the users in getting a good deal on a car, the user feedback is a great source in determining how reliable the sellers are.

Discussion of Issues

We intend to make this application a Single Page Application (SPA). SPA definitely has its benefits, but the developers must always keep the current document managed with caution in order not to overload the browser memory.

The concept of a documented-oriented database is not our area of expertise. JavaScript as the server side and Angular.js is an upfront investment. If time allows we intend to use Google Maps API and Facebook Login API. These APIs sometimes can be tricky and time consuming. The fall back strategy if an API does not work as intended would be to look for an alternative or fake the same functionally. However, the proposed APIs in this application are stable and the web is filled with rich documentation.

Since we are not using a secured protocol network (HTTP), antivirus software such as AdBlocker monitors the clients HTTP Methods calls looking for malicious URLS paths to block. So disabling AdBlocker during the development stage is a must. As the application develops, choosing a templating engine that will be easy to debug and format is essential. Thus, EJS (Embedded JavaScript) was chosen for its similarity with raw HTML format.

Schedule

DATE	TASK	ASSIGNEE
2/4	Proposal	Mulkern, Goncalves
2/6	Setup Github/Heroku MongoDB and Node.js Express	Mulkern, Goncalves
2/13	Home/Search UI	Mulkern
2/13	Home/Search back-end integration	Goncalves
2/20	Login/Signup page UI	Mulkern
2/20	Login/Signup back-end integration	Goncalves
2/27	Profile page UI	Mulkern
2/27	Profile back-end	Goncalves
3/5	Car-ad page UI	Mulkern
3/5	Car-ad back-end integration	Goncalves
3/8	Alpha version	Mulkern, Goncalves
3/13	Feedback functionality	Mulkern, Goncalves
3/20	Post/Delete/Modify functionality	Mulkern, Goncalves
3/27	Search functionality	Mulkern, Goncalves
4/5	Beta version	Mulkern, Goncalves
4/5	Google Map API integration	Mulkern, Goncalves
4/12	Facebook Login API integration	Mulkern, Goncalves
4/19	Class presentation	Mulkern, Goncalves
4/28	Final submission	Mulkern, Goncalves

Acceptability Criteria

The application should allow users to:

- Create an account
- Log into an existing account
- Delete the account
- Search for car ads
- Post car ads
- View own profile
- View other user's profiles
- View feedback left by other users on own profile and other profiles