**College of Science and Engineering**

School of Computing Science

**Interactive Systems Report,**

**Assessed Exercise II**

Team 8:

Atanas Penchev - 2072742

Alex Chilikov – 2087690

Pavel Dikov - 2070616

Luke Doleman - 2084967

Tony Petrov – 2087829

Alexander Mladenov - 2074786

Date: 30th of November, 2015

**Contents**

1. *Summary of Design Requirements ...................... 3*
2. *How does the system stem from the paper prototype and requirements are met (System Description).............. 3*
3. *How are non-functional requirements met ......…..... 7*
4. *Testing and Evaluation ............................................... 8*

* *analytic evaluation .................................................... 8*
* *empirical evaluation .................................................. 9*
* *qualitative ................................................................... 9*

1. *Why this system is best for the job .............................. 9*
2. *System description (for marker) .................................. 9*
3. *Teammates contribution .............................................. 10*

Summary of Design Requirements:

The functional requirements of the system are separated into the following categories:

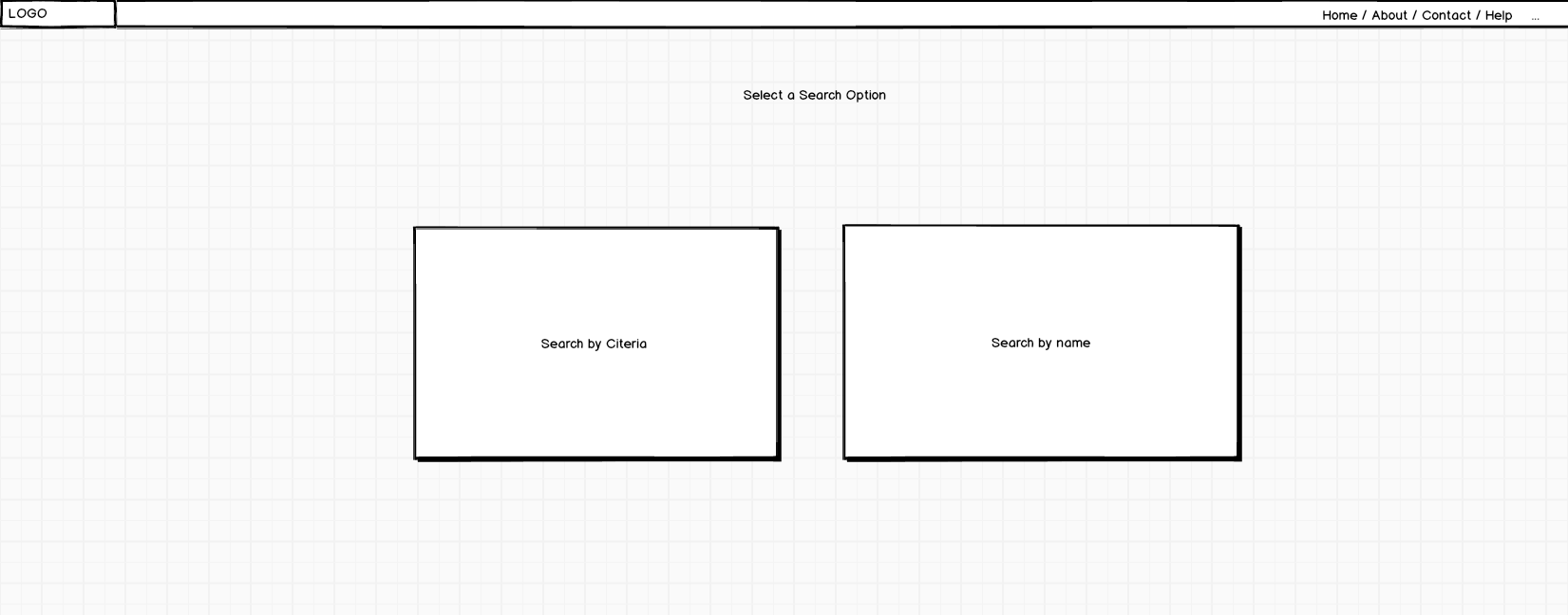
* search criteria - such as cuisine, location/distance, rating, dietary choices (vegan, diet, kosher/halal etc), price range
* restaurant info, such as contact details, address, restaurant category, web page
* map visualizing the result providing interaction for seeing a detailed view of the restaurant – contact details, address etc.
* line graph for visualizing how the restaurant’s rating has been changing over time
* search by name or phrase

Non-Functional Requirements respectively consist of the following:

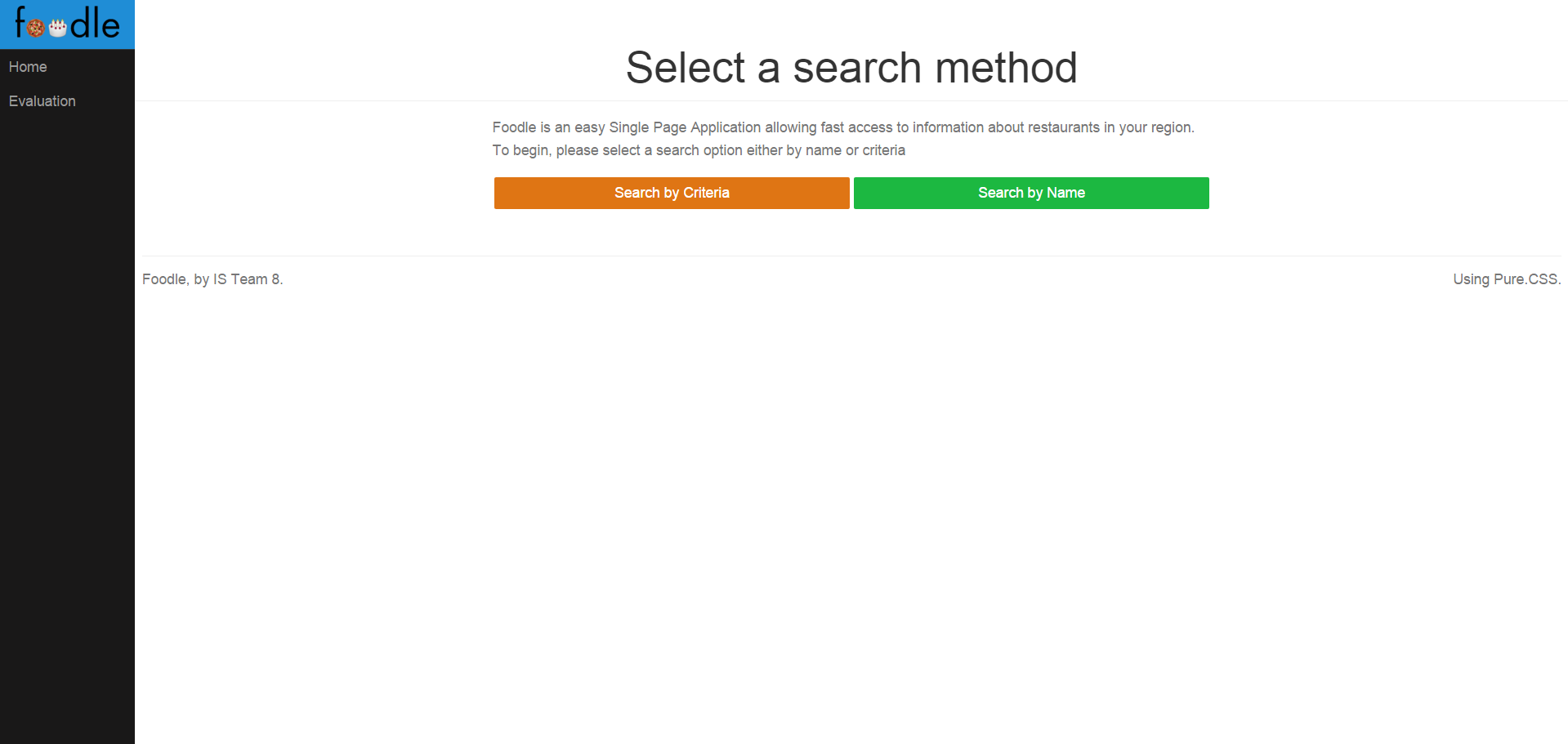
* performance (quick searches)
* geolocation using HTML5
* fetch reviews using APIs

How does the system stem from the paper prototype and requirements are met (System Description)

The initial page presented in the prototype which looks in the following way (meeting the search by name or criteria requirements):



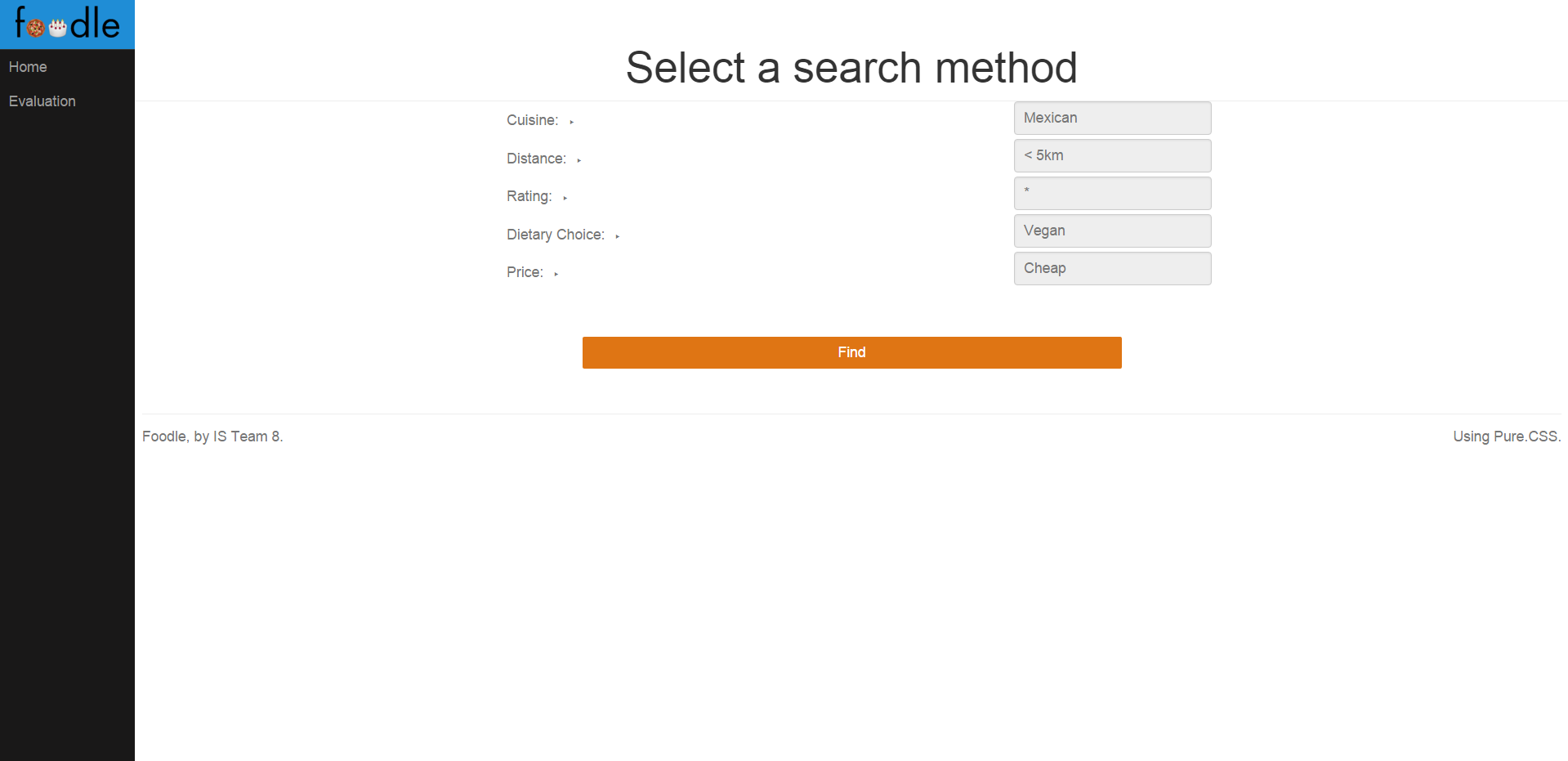
has been used as a basis for the home page. The actual home page presented below:



*Index Page*

shows how the index page looks in production. As you can see the logo and the two buttons search by criteria and search by name are still there. The select a search option is clarified with an extra paragraph providing more information about the two buttons. The navigation bar and the links on the left-hand side are slightly changed. As the help link is removed and an evaluation link is added to present the analysis of the user interaction.

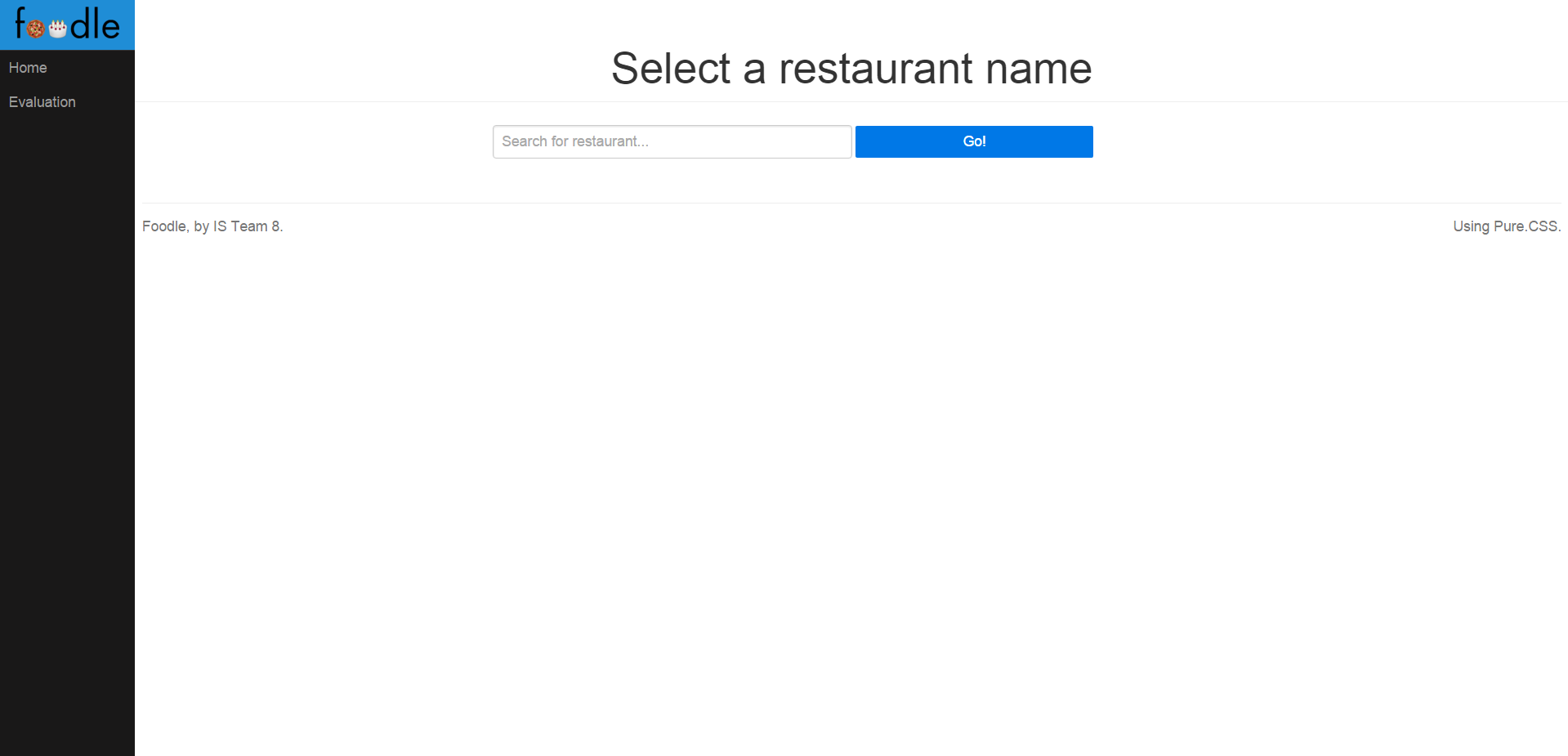
Then we can look at the search by criteria page below satisfying the search by criteria requirement:



*Search by Criteria*

As this page is supposed to be quite simple and basic it is not part of the prototype. It simply includes all of the categories and a search button.

The search by name page

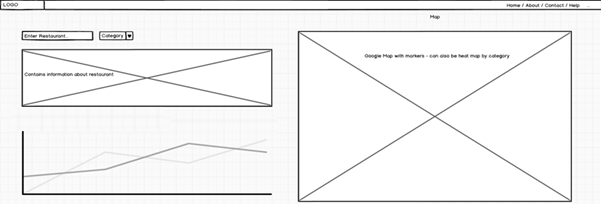


*Search by name*

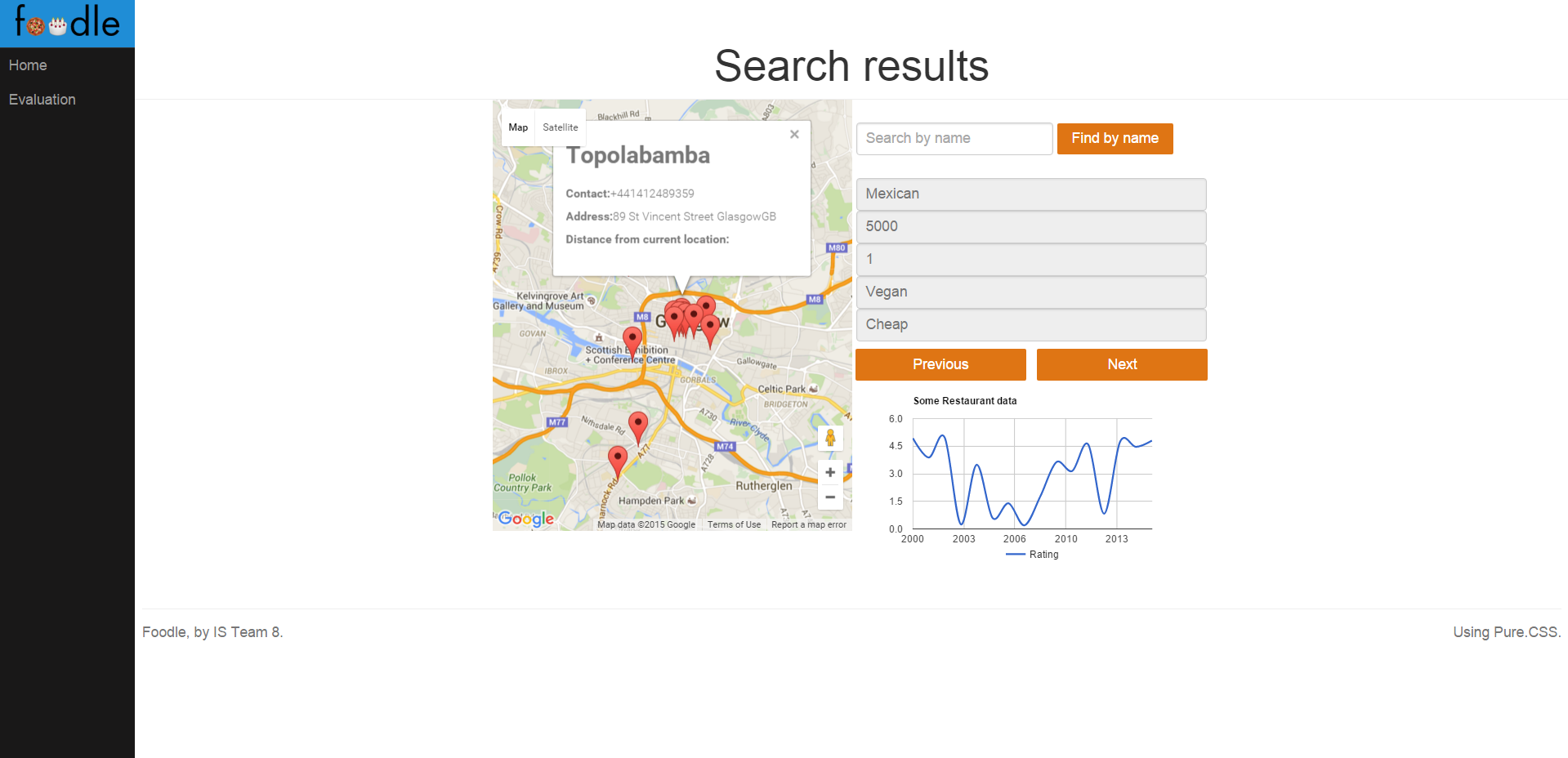
is similarly not included in the prototype as it is simple enough to be directly developed from scratch. It consists of just one input text field and a search button.

The flow of the application is split into these two paths for improved understandability, simplicity and effectiveness. The user either knows exactly where he or she wants to go in which case he/she searches by name, otherwise he/she wants to filter the data in some way to see only the results satisfying his/her desires.

The last page of the system showing the results were initially planned to look in the following way:



The final page:

*Results page*

is quite similar to the prototype, as the key features we are looking to provide here are the following:

* an interactive map /a requirement/ - showing the results making seeing the data in a much easier way boosting the decision making of the user. Using the map he/she can see where exactly each of the restaurants is located and get a detailed information about the place only if the location satisfies her/him instead of looking at a list of restaurant with plenty of information which is most likely not need or wanted.
* restaurant’s information /requirement/ – once a restaurant’s marker is clicked the restaurant’s information will appear.
* line graph /requirement/ – once a restaurant is selected clicking on the restaurant’s marker on the map a line graph in the bottom left-hand corner will show the dependence between the time (x axis) and restaurant’s rating (y axis) – how the rating has been changing over time.
* search by name and criteria – in order to make interaction faster and more enjoyable the search box and criteria dropdowns are also left on the page in case the user wants to search for something else, allowing direct access to the search form instead of going back to the index page and starting the process again. The reason why we do not have only one page (this last page) is that we want to allow the user to gradually familiarize himself/herself with the system going through either the search by name or search by criteria. Once the user is familiar with this options we can show more information on the last page knowing that the user has learned the previous steps and will not find it difficult to navigate on this page.

How are non-functional requirements met?

There are 3 main non-functional requirements satisfied as follows.

* First, the performance is improved by abstracting the search functionality in the API, in other words instead of processes the request and the data in our server we are sending the request as part of the HTML Restful GET request allowing the API server to compute the result for us. Similarly the line graph uses an optimized Google algorithm provided from the Google Charts API. The Google map is also visualized by gathering the data needed for the map ”locally” inside our server and then sending the Google map API for processing.
* Secondly, the geolocation is taken using the HTML5 functionality.

- Finally, the data for the restaurants is acquired from the <https://www.yelp.co.uk/developers/documentation/v2/search_api>

Testing and Evaluation

Analytic Evaluation (Cognitive Walkthrough)

The cognitive walkthrough is an example of an analytic evaluation in which an expert ‘walks through’ the system checking whether it conforms to psychological principles. At this experiment, we will pretend to be experts going through the system and evaluating it based on a set of predefined questions.

General questions:

* What impact will interaction have on user?
* Restaurant finder is a simple application allowing you to find the desired restaurant fast. The expected impact is a temporary satisfaction caused by finding the restaurant fast and meeting user’s requirements such as cuisine,rating,dietary choice.
* What cognitive processes are required?
* The web application is nicely split into smaller components restricting the cognitive effort required for achieving the desired result. The workflow goes from selecting a search option, entering the search information and receiving the result. This separation allows the user to concentrate his attention at completing small tasks at a time reducing errors. For the final page where the results are shown, there are several alternative mechanisms showing the results allowing the user to easily understand the page and find the right restaurant. Interactive map provides a visual summary of the results. On the right-hand side a table-like structure shows the search criteria not expecting the user to keep in mind his search. By clicking on a marker on the map the user can see a summary of the restaurant in a straightforward text format. On the other side, a line graph will dynamically be generated showing how the rating of the venue has been changing – giving a pattern simplifying user’s choice to choose this place or not.
* What learning problems may occur?
* Considering the simplicity of the application, major learning problems are not expected. Perhaps the interactive map and line graph visualizations can be categorized as an underlying problem for people with little computer experience as we do not see and interact often with maps and graph visualizations during our everyday use of computers. In order to resolve this problem a previous and next buttons are placed to offer an alternative.
* Does the design lead the user to generate the correct goals?
* Similarly to the answers from the above psychological principles, it can be detected that the workflow is quite simple giving an useful feedback each time we progress towards the desired goal. If we for example start from the index page and choose a search by criteria,the application will take us to the search by criteria page where we can choose criteria. This is clearly indicating the user that he/she is going in the right direction. Once the required fields are filled in, the user can click find which is expected to channel him to the results page.

Empirical Evaluation (Tests user experience metrics)

Usability metrics are metrics evaluating people, their behaviour and attitude. For our experiment we will use 5 volunteers ask them to use the application and record the usability metrics in the table below – all metrics are from 0 to 5, where 0 is negative and 5 is positive:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Subjects/Usability metrics | 1 | 2 | 3 | 4 | 5 |
| Task success | 5 | 4 | 5 | 5 | 4 |
| User satisfaction | 5 | 4 | 3 | 5 | 5 |
| Annoyance | 0 | 0 | 2 | 1 | 0 |
| Flexibility | 5 | 4 | 4 | 4 | 4 |
| Want to use again | 5 | 5 | 5 | 5 | 5 |

To sum up, it looks like most of the users have been satisfied interacting with the app. Most of them have managed to complete the task successfully (searching for a restaurant finding the right one). They look satisfied and most of them want to use the application again. The annoyance level is low but it looks like we can think a bit more about the flexibility of the application which is most likely related to better navigation and search functionality.

Qualitative Evaluation (Think Aloud)

Think aloud is an example of an empirical evaluation. It is quite often preferred for its simplicity, useful insight, and the fact it shows how the system is actually used. The disadvantages are that it is quite often very subjective and that the act of describing alters task performance. In this experiment, we will ask the subject several question while he/she is using the system and write down the responses.

* Tell me what you are thinking?
* I am thinking I would like to find a cheap restaurant nearby preferable for vegans. I do not mind to be lowly rated, I am simply looking for something cheap with Mexican cuisine.
* Tell me what you are trying to do?
* First, I see the home page, knowing what I want to do I am choosing the search by criteria, as I do not know a name of any restaurant nearby.
* Are you looking for something? What?
* At the moment I see the search by criteria page. I am looking to select the criteria I have in mind. Once, I go through the possible options, I choose the once I am interesting it. Now, I should click find.
* What did you expect to happen just now?
* I expected to see a list of restaurants meeting my requirements but I saw something a bit different.
* What do you mean by that?
* I did not expect to see a map but after thinking for a bit realizing that do not have a list of restaurants, I figured out that I should be somehow interacting with the map. The red markers attracted my attention helping me to understand that I should maybe try clicking on them. Once, I clicked I saw an information about the place and a graph with the rating which was quite useful to find out whether this venue has always been a good choice or it has just become popular. This extra visualization helped me choose the right restaurant.

Why this system is best for the job?

We believe this system is the best for the job as it both provides an extremely easy for use interface which is very interactive and a lot of different mechanisms for conceiving the results and searching for a restaurant. In our opinion, this unique combination of features makes our restaurant finder stands out. The interface and template uses a new dynamic approach following the current web app development trends for optimized user experience. The performance is extremely fast relying on external servers to compute the results. Looking at the evaluations performed it looks like both experts (Analytic Evaluation) and users (Empirical and Qualitative evaluation) are highly satisfied with the system for which reasons we believe our web app finder is currently the best restaurant finder. (in order to reduce repetitions some of the findings from the evaluations are not repeated at this stage but they provide an extra proves why the system is very professional)

System Description ( for the marker)

The system does not rely on any local libraries to be installed, all libraries used are acquired dynamically. For this reason, the only thing required for setting up the system is:

Please use Chrome (developed in chrome)

1. Extract it from the archive file.

2. Open index.html with a preferred browser and start using

3. Select a search option

4. Choose criteria for the search by criteria or enter a name of a restaurant for search by name

5. Click find

6. The results will appear

7. Either choose a marker from the map to see a particular restaurant or press previous/next to see restaurant information in turn.

8. Once a restaurant is selected a line graph on the left will be generated providing an additional visual data.

9. Then click the evaluation link from the navigation bar on the left-hand side.

10. Once clicked it will show you a workflow – how the mouse has been clicked from the user and in what order showing how a user navigate within the page. This can help for later refinement of the design.

Notes: to see the evaluation, you should go through select search by name or criteria and go to results page and look at the evaluation after this (as otherwise there will be nothing to be evaluated) (to clear the evaluation result you should clear the cache of the browser –as we do not have a back-end data is recorded in the browser’s cache)

Team Contribution

1. Alex Chilikov – personas and scenarios, prototype, Google maps, line graph, evaluation page, final report (including analytic, empirical, and qualitative evaluation)

2. Atanas Penchev – search by criteria + Yelp API request, search functionality on the results page, previous/next logic, binding yelp data with Google Maps and line graph

3. Pavel Dikov – requirements, front-end design (Pure.CSS), interaction semantics (search by criteria), evaluation (Fitts's law calculation & chart), video demonstration

4. Luke Doleman – requirements, template building (Pure.CSS), video demonstration

5. Tony Petrov – search by name page + Yelp API request

6. Alexander Mladenov – API

/team please add what you have been working on, I just wrote down what I managed to think of/

Imported libraries – no imported libraries, links to Google maps, Google graphs, jQuery, and D3. Pure.CSS used for the layout.