Professional Software Development (H)

Team I

Project Specification for Resdiary.com

***Outline:***

The current resdiary.com portal allows diners to search for restaurants and available tables for dining. There is a system in place on the site for users to restaurants in their area but ResDiary would like a recommendation engine to be added to recommend new restaurants based on a diner’s previous dining habits and their similarity to other diners. The goal of this project is to create a recommendation engine that accurately suggests places to eat based on previous restaurants and other people with similar preferences.

***Functional Requirements***

* The engine must accurately suggest places to eat based on the user’s own previously visited restaurants and other users with similar eating preferences.
* Recommendations should be in close proximity to where the user typically eats or is currently searching.
* The engine should re-recommend previously visited restaurants provided the user selects this option.

***Non-Functional Requirements***

* The engine should be written in C# for easy integration within existing Resdiary systems.
* The system should give a response within 1 second after receiving the request (provided data is stored locally). Alternatively due to the amount and nature of the data the system should perform nightly updates to the recommendations made.
* New users should be presented an optional quick quiz/questionnaire to gather initial data.
* User locations and restaurant recommendations should be interpreted using coordinates rather than city/town name as those could be of arbitrary precision.
* Have the ability to “fine tune” the recommendation engine by altering the weighting of the significance different components of the recommendation such as distance, price, reviews, etc.

***User Stories***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | TYPE | As a... | I want to... | So that I can... | Notes | Priority |
| 1 | Func. | Existing user | Be recommended new restaurants | Enjoy a new dining experience |  | High |
| 2 | Func. | New user | Be able to take a quiz | Get good initial recommendations |  | High |
| 3 | Func. | Existing user | Have recommendations based off my similarity to other users | Be recommended restaurants with similar tastes to my own | Client stated this should be key part of the engine. | High |
| 4 | Func. | Existing user | Be able to take a quiz | Refine recommendations to get more accurate ones |  | Med |
| 5 | Func | User | Have recommendations take my budget into account | I’m recommended restaurants within my price range. |  | Low |
| 6 | Func. | User | Be able to disable re-recommendations | Always be provided with a new experience. |  | Low |

***Split stories into tasks***

* Draw up a database schema for storing relevant user information
* Make a questionnaire for new users to generate initial data:
  + Create a basic front-end including simple yes/no questions, location fields, (preferred and non-preferred cuisine types -- again less emphasis on this was suggested by customer).
  + Set up the back-end to process and persist questionnaire data.
  + Customer stated the Netflix model as an example of how they imagine this would be incorporated for new users.
* Make a questionnaire / refinement section for existing users to refine recommendations:
  + Create a basic front-end including simple yes/no questions, location fields, (preferred and non-preferred cuisine types -- again less emphasis on this was suggested by customer).
  + Set up the back-end to process and persist questionnaire data.
  + Customer stated the Netflix model as an example of how they imagine this would be incorporated for users.
* Write the recommendations engine, taking the following parameters into account:
  + Similarity to user others -- customer stated this was key basis they wish recommendations to be made off
  + Other user reviews
  + Other user favourite restaurants
  + Current or typical location
  + Critic reviews
  + Restaurant availability

Customer stated that “Cuisine Type” and “Price” should not be as significant a weighting as we originally suggested at the project proposal meeting. While still a feature of the recommendation it should not be as significant as we anticipated.

* Create a stand-alone front-end prototype for displaying recommendations
* Integrate the engine into resdiary.com

***High-Level System Design***

UML.png

### ***Workflow***

1. ResDiary sends a User object to our Recommender System, which contains all their previous reservations and reviews.
2. Recommender System sends that object to all Recommenders in the system.
   1. Some of the Recommenders are collaborative: they find similar users and recommend restaurants that those users liked.
   2. Some are content-based: they look for similarities between restaurants and recommend similar restaurants based factors such as price range and cuisine.
   3. Some Recommenders can combine both approaches.
3. Each Recommender returns the top N recommendations to the Recommender System.
4. Recommender System combines them based on some predefined coefficients and outputs the final recommendations.

Evaluator class can be used to judge system performance. For example, the restaurant review data could be split based on timestamps. Earlier data can be provided to our Recommender System. The System recommendations can then be checked against later bookings and reviews. The Evaluator outputs the percentage of bookings that our system guessed right.