***02/11/2016***

Meeting began discussing implementation language. The possibility of using Python to build recommendation engine was discussed as well as the advantages and disadvantages of doing so. Main advantage being Python would be more suitable to the task and could make use of already existing libraries to improve the engine. Python also used commonly for similar task in industry. No definite decision was made on this subject though.

Tasks:

1. Investigate how easily it would be to call a Python script with parameters within a C# program.
2. Potentially build a Python prototype showcasing Python specific libraries (Edward selling Python to opposing team members / ResDiary)

Next topic was discussing when we should meet with the customer and what we should discuss with them. It was decided that a project specification / proposal be brought to the meeting including the following key items:

1. Overview, Functional & Non-Functional Requirements
2. User stories using Josh’s format template
3. Mock up UI designs to show how it might possibly be implemented into their website.
4. Questions to ask regarding implementation language and other aspects of the project.
5. Edward would like to bring a Python prototype showcasing why it is the language of choice over C#

This was decided as being the first major milestone of the project and tasks were set up and divided amongst ourselves.

Tasks:

1. Work on functional and nonfunctional requirements of the project.
2. Create user stories in Josh’s template format.
3. Make a sketch of front end implementation / UI prototype sketches.
4. Determine questions to ask customers at meeting.
5. Ensure meeting is set up and arranged for correct date.

Other discussions took place into different factors which will need to be considered when developing the recommendations. These will be noted more formally and discussed at a later date. Finally now that the project has began full on it was decided that it is important to document meetings in terms of what was discussed as well as planning, scheduling and other significant tasks. The team should track the outcome of these meetings in their associated project artifacts.

Tasks:

1. Write up meetings notes after each meeting.
2. Create milestones and tasks outlined above on Trac.
3. Get Trac wiki system in place to document meeting notes easily.

***09/11/2016***

Meeting began discussing (once everyone arrived) by going over what still needed to be done regarding the specification for meeting on the 16th with the customer. All the main pieces were determined to be in place - user stories written, UI sketch, etc. but it was determined that planning poker should be played at some point to get a general feeling throughout the team of what the most difficult aspects of the project will be. Created user stories were also discussed which lead to further feature discussion for the engine to occur also. These were all noted for write up later in the idea thread which serves as a thread of topics for further discussion at a later date.

Task:

1. Play planning poker on tasks
2. Add additional potential features and factors to the idea thread for further discussion at a later date.
3. Formalise idea thread on the wiki

Edward made his case for using Python once more and claimed to have been playing around with prototypes and libraries (though no code was seen) and further discussion took place around language choice. Edward also requested that we make an enquiry to get some data in order to test and continue his investigation further.

Task:

1. Send email to ResDiary requesting data to allow Python prototype to continue.
2. Take a look at any data / files received from ResDiary.

Finally, it was decided also that a high level system design of how we think the engine will be implemented should be created. It was decided to do this as a group activity at the end of the meeting as we each sketched individual prototypes of how we believe the engine would be implemented. These sketches varied in complexity and discussions took place around our initial ideas of how we might go about it. The final version was then added to the specification report with a description of how the system works. One concern which was brought up at this stage though was how the company ResDairy wants the recommendations to be passed back to them. This will need to be clarified with them either at the meeting or via communication with them at some point.

Task:

1. Create prototype sketch of the high level system design and add to specification. (Already completed)
2. Find out how ResDiary wants the recommendations to be passed back to them.

***16/11/2016 (Customer Meeting Recap)***

This was the first meeting with the customer since the initial requirements gathering meeting though we have kept in regular contact via email. We sent the proposed system specification to the customer a few days prior to the meeting and went over it again with them at the meeting. A link to this version of specification can be found here (link - ensure its iteration 1 on wiki)

A summary of comments / feedback made from the customers is as follows:

* Liked the idea to re-recommend restaurants to users. Would want the option to turn this feature off. Users specify if they want to be re-recommended locations or not. Possibly have default be if they haven’t eaten there if 12 months but have option to make it 6 months, 3 months, etc.
* Liked idea of a quiz for new users of the site. Referenced Netflix as an example of new user signing up to service and taking quiz.
* We initially thought that the price of a restaurant would be a higher emphasis of over similar users when making a recommendation. The customer however had a strong preference that the similar users be the emphasis of the recommendation and price point a secondary factor.
* Regarding our aim for a timely response within 1 second, the customer did not think this would be feasible given the amount of data they are working with. With approximately 10 million users there’s a lot of data to sift through. The nature of the data is that it’s also quite static (there’s only so many new recommendations which are going to be added in one day and it’s not going to significantly alter the recommendations made). As such they proposed that the system in place is a nightly build / offline basis as it is probably more suitable given the context here when considering the large dataset, the rate at which it changes and the effect that having a slight change in the data has on the recommendation made to the user.
* Regarding our high level design, they liked the idea to make the recommender configurable and suggested they would like the ability to have a weighting for each major category so that they could “fine tune”, tweak and refine the engine over time.
* Commented on the issue of personal information being given in data dump as users are identified as being unique by their email address. We do not believe this will be an issue though as having looked at the schema they sent previously (link to schema) the user’s have an ID which we can use. The email address is also not required in order to make recommendations and so can be omitted from any data sent to us.
* When asked how they imagined the system working, they proposed creating a basic front end database display to help assist with both showcasing the service from a “this is an assessed project and this will help show off the work and get marks” perspective and to help them better understand the system in place. They also suggested the potential of knocking together a basic API as well.
* When asked regarding the dummy users in the data some clarity was provided as to how this data occurs in the system. Essentially when restaurants take telephone bookings, the member of staff will just enter dummy user data such as “[a@a.com](mailto:a@a.com)”, “[asdf@zxc.com](mailto:asdf@zxc.com)”, etc. and as such this data is unpredictable. As this data should probably be avoided.

They did point out that for such data you often see reservations spikes where the number of reservations made by one user at a particular restaurant. A user isn’t going to make 12 bookings within one day for varying amounts of people.

Of note also from the above point is that there does not exist a unified database of users.

TASK: Investigate Netflix new user and similar quizzes.

TASK: Alter project spec to align with comments made at first meeting.

NOTE: Price of restaurants visited could be used as method of determining the similarity of users.

***Meeting Notes: 23/11/16***

*Eduard not present as he is at an intern interview.*

*Dom not present as he is attending a hackathon in Hong Kong to try and win a trip to London.*

The team discussed the weekly PSD task of setting up doctest for the project. The team agreed that we are not fans of this as it requires the tests be in the same file as code. We also have Python Unit Tests already set up which serves the same purpose.

We are still waiting on additional data from ResDiary to assist with work on the recommendation engine.

Paulius commented he is continuing researching data spikes. Graphs were shown with the initial investigation into this issue and it was decided that evidence of this research should be documented properly for marks.

Planning poker needs to be played.

After poker has been played the focus on getting a first working prototype together for the next meeting on the 7th in two weeks time.

The details of what we wish to achieve with this working prototype should be finalised next week when more / majority of the team are present..

***30/11/2016***

While waiting for the team to arrive the two members of the team (Joseph and Paulius) spoke with a lecturer, Dr Inah Omoronyia, about the initial planning for the project thus far. He commented on the following aspects of the specification which could be improved:

* Regarding the proposed user story of an existing user wanting to be “recommended new restaurants so that they can enjoy a new dining experience” may not be a suitable user story. This he suggested would be hard to break into tasks and tests to verify that it had been implemented correctly. This appeared to be more of a “soft goal” opposed to a user story case.
* Regarding the non-functional requirement of making a recommendation within 1s he laughed. (Dom I told you so). He suggested it would be very hard to meet that goal but did suggest that a timer based test could be setup to verify if the recommendation would be made in a specified timeframe. He also commented that such a test would likely never pass.
* Finally regarding the user stories he asked if we could see any dependencies between stories. Such dependencies should be eliminated from the user stories as they should not rely on one another.

Most importantly though he asked the following:

* How is user privacy and security being handled. He stated at the end presentation of the project that this would be the first question he would probably ask of the project. He mentioned also that it should be mentioned within the report and shown that the issue has been identified, discussed and handled in a suitable manner.

Edward gave reasons for not uploading the latest data set sooner. Dom determined to be the contact for future communication. Work on project has been slow but was anticipated due to assessed exercises and other commitments.

A summary of work done towards the demo so far is as follows:

* Filtering of data proof of concept - shows how invalid data can be identified from the data set.
* Basic similarity rating between users- shows how similar two users are based on visited locations and reviews made.

What should be presented at the demo next week was then discussed in detail. We arrive at the aim of showing the following:

* Filtering of user data to remove invalid data from dataset (already done)
* Front end which allows user to specify user ID - returns recommendations for that user
* Back end which makes basic recommendation to specified user

The plan for the demo is to create a basic front end and incorporate the basic similarity recommendations in as the recommendation engine. It will have a limited selection of users for which to make recommendations. The data will initially be loaded locally until Eduard sets up the online database which we then hope to transition so that the data is pulled from there.

The plan for the presentation itself being to show the data filtering and explain how we are filtering out the data based on the frequency of bookings. Then to show the front end which asks takes a given user ID and returns a list of recommendations for that user.

**02/12/16**

There are too many chiefs and not enough Indians.

A general plan for the demo development:

* Edward sets up database to house the data.
* Vladimir continues work on simple python recommendation and outputs some data.
* We ensure that the recommendations are in a format that the server / front end can use.
* Dom incorporates the recommendations into the front end for display.

In retrospect what happened was as follows:

* Vladimir did his job.
* Edward eventually got around to setting up the database and tying it with Vladimir's recommendation engine to produce some output.
* Dom and Paulius prototyped with an alternative solution for Edward's job. This system initially involved simply calling the Python engine from within the Nodejs server code to produce a CSV file which was then parsed on the server side to generate the recommendations. They then moved on to working / prototyping with an alternative database solution (! phpMyAdmin) to perform the same role as Edward. This was later discarded and Edward's solution used for the demo.
* Joseph did some R&D into topics such as nearest neighbour, CSS animations for the output and other such topics. He also helped out by implementing the calling of the Python script from within the Nodejs as this was something which had appeared in his R&D work for the project thus far.

Additional Notes:

* Edward attempted to set up AWS to house the data. The university network was found to be an issue as it blocked connections made to this. Alternatives were suggested / discussed, hissy fits were thrown.
* People and communication are not enjoyed by a portion of the team. All I say to those people in response is the [​following](https://www.youtube.com/watch?v=EHV0zs0kVGg).
* Paulius did maths.
* Josh looking into potential security / privacy / data protection act issues.
* Don't be a sheep, be a shark.