

## Outline

The following is the outline and plan for development of the first prototype which was presented to the customer on December 7th.

A brief note of some of the comments made during the first customer meeting which in turn had an effect on the development of the first prototype:

- Customer suggested building a lightweight front end for display purposes. This is primarily for demonstration purposes on our end as a way of showing of the built system at the end of the project to the assessors.
- Context was given on the dummy users within the day (restaurants using the same email address of multiple bookings) and so it was decided to investigate how to remove this data from the dataset. Furthermore if a user changes their booking it creates a new booking and the old one remains present also.
- The database for the application would preferably not be stored locally. For the prototype though a local setup is fine as a temporary alternative for prototyping purposes.

Remember prototypes are throwaway. With that in mind the team decided on the following aims for what the first prototyping session would achieve:

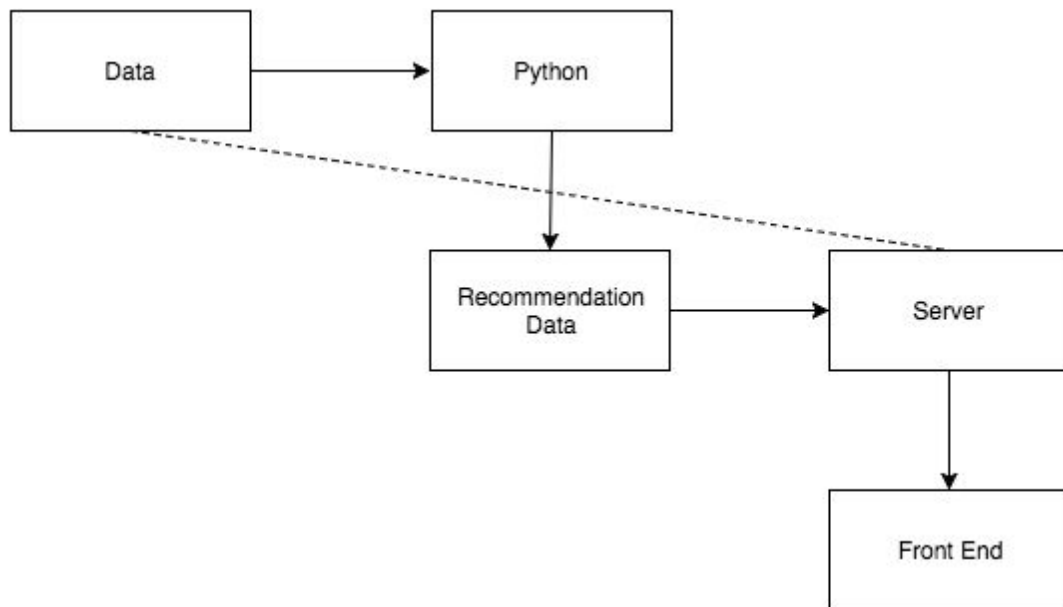
- Present the results of an initial investigation into identifying the dummy data. This will use some form of anomaly detection to try and identify abnormal amounts of bookings being made by a user. This will not involve tackling the issue that occurs when a user alters their booking.
- A database setup to store data and query and a simplistic, similarity script created using Python that generates recommendations for a user based off their similarity to other users using the system.
- A basic front end display to display recommendations for a preselected number of existing users.
- Have a selection of questions ready concerning the privacy, security and use of user data for the customer.

It should be noted that it was decided that the prototype would not achieve the following:

- Allow the existing user who recommendations are made for to be a user than the preselected number.
- Contain any new user functionality. Contain the possibility to refine the existing user functionality.

- Contain any other previously discussed feature that is not mentioned above.

In addition to this a quick diagram was sketched (later redrawn here) to provide a visual aid for the system and to help allocate roles within the team:



Initially the proposed development plan for the prototype was as follows:

- Joseph / Paulius would work on the anomaly detection.
- Joseph & Josh would assist on both aspects of the front and back end as needed or otherwise do research into future aspects of the system.
- Dom would build a lightweight front end for the application.
- Josh would research the privacy, security and use of user data issues.
- Vladimir would create a simple Python script to make the recommendations to the user.
- Eduard would set up an Amazon Redshift database to store the data.
- Eduard / Vlad would tie the database and script together and produce recommendations to be incorporated into the front end.
- Someone would incorporate the recommendations from the Redshift database into the front end.
- A local backup system would be created in the event problems with the online database setup occurred while Eduard set up the online equivalent.

The built system would then be presented to the customer alongside the dummy data detection. They would be walked through the plan for how we imagine the next iteration to go forward and the work done thus far. We also will take the opportunity to ask any questions which have arisen during this iteration such as the concerns with the privacy, security and use of user information.

In retrospective the development process for the prototype was more as follows:

- Vladimir created a Python script for the stated purpose.
- Paulius made a head start on the anomaly detection before Joseph got a chance to start work on it and the simple detector he created provided good results and so it was decided to leave it in this state for the first iteration and change development focus elsewhere.
- Josh did some research into the highlighted issues.
- Dom created a light front end interface to display recommendations for a preselected number of users using Nodejs.
- Eduard set up a Redshift database, query to it and incorporated the recommendations into the front end.
- Paulius and Joseph took Vladimir's script and made a locally running equivalent of the system whilst waiting for Eduard to setup the online version. Paulius handling the filtering of preselected users and Joseph calling the Python script from within Nodejs.
- Dom and Paulius experimented with setting up an alternative database (phpMyAdmin) whilst awaiting for Eduard to setup the Redshift database.
- Joseph did research into several aspects for future iterations of the system.

Outlier detection results:

