Decision Making

1. decide tutorial slot and team meeting time

According to the course schedule, we organised a doodle vote for possible time slots. Because we are a large team formed with 7 people, we can only count out one suitable time for everyone at first, but the course requires four. In this case, through active and timely communication and coordination, the team members finally decided to choose four time periods at the expense of other curriculum arrangements for minorities and registered the most inclined time periods by spokespeople. After the official schedule came out, we chose another alternative as the time for team meetings.

2. decide roles and responsibilities

In order to determine the roles, we first confirmed the requirements with our client. This semester he wants to change extractions and algorithms to make the website more Canberra focused. To satisfy Canberra users' habits, we also need to change the UI and conduct several user surveys. After confirming the goal, we divided the work into two groups according to each member's ability and interest: UI & user group consisting of three people, extraction and algorithm group consisting of four others.

3. decide the usage of trello

At the first group meeting, we confirmed that we will use trello as a process management tool. Subsequently, we had some discussions within the team on the details of how Trello will be used. Based on everyone's experience with trello, we finally decided on two options to choose.

First, only basic to do, doing and done are used to list items, and each member chooses his or her own tasks. The advantages of this method are easy operation, simple structure and time saving.

Second, besides todo, doing and done, we will have every week's list. Each member writes down what he actually did this week and marks the time in the list. The advantage of this method is that it can clearly see everyone's workload and facilitate the task assignment and burndown chart statistics. But it is obviously more time-consuming than the first way.

One of our team member has participated in tech launcher several times. He suggested that the first method would easily make tutor confused and not clearly reflect everyone's workload. Therefore, after a vote, we decided that the second method should be adopted despite the slight inconvenience.

4. decide goals and schedule for the whole semester

In order to determine the goals and schedule for this semester, we first confirmed client's requirements. After communicating with him, we divided into two working groups. One is the UI and user group, the other is the extraction and algorithm group. The client suggested that he would like to conduct 2-3 user surveys this semester. The results of user research can also be used to determine the direction of follow-up projects.

According to the program dates, we drafted a schedule and submitted it to the client. In this scheme, two user surveys will be conducted in the fifth and eighth weeks respectively, and the last user survey will be conducted in showcase. The aim is to ensure that we have enough time to take action on the results and that we can make major adjustments during the mid-term vacation.

5. decide to add user story map

Last semester, this project does not have any user story map. This proposal was put forward by several team members, they proposed that user story map can help us understand the product from the user's point of view. Previously, the scope of the product's users has been unclear, but this time the client clearly put forward the user group he expected: Canberra residents. Therefore, it is very important to understand the product from the user's point of view. After a team meeting, we decided to draw a user story map for the product.

6. decide which code from last semester to keep

Because we are continuous group, the previous team left a lot of code. We have two options for code retention.

First, join the repository of the previous team and make changes directly on their git repo. The advantage of doing so is convenience, but the disadvantage is that it may cause confusion in the document and inherit the bugs left behind.

Second, keep the useful code and construct a repository from scratch. The advantage of doing so is that the structure is clear, because we are sure that what remains is useful. The disadvantage is that it is troublesome to invite client, tutors, etc. again, and may cause dissatisfaction from previous contributors.

To solve this problem, we consulted course convener. Charles suggested to us at Piazza that repository should be built from scratch and a code review should be made to determine the useful structure and code to be left behind. Therefore, we left a useful part of the code on the previous repository and rebuilt the repository.