Foodback MIT
Individual Design Document Draft

Motivation

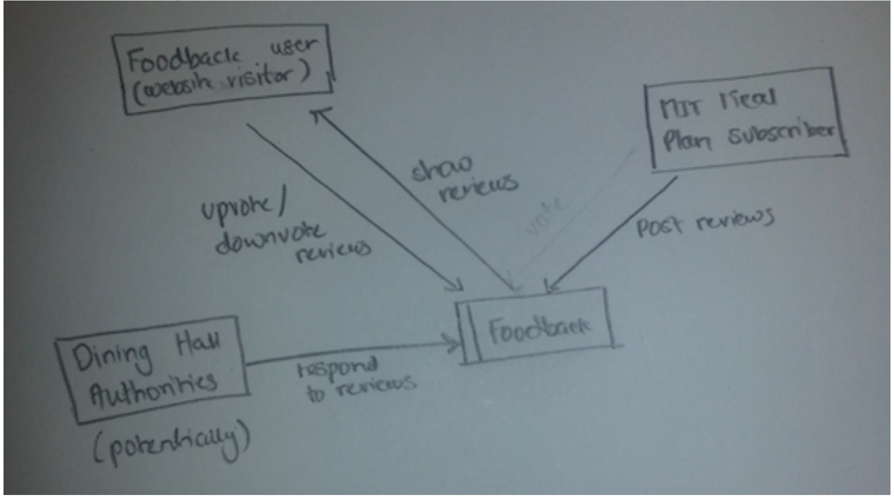
Foodback MIT is a web application that allows MIT students to review the dishes being proposed on a particular day. As feedback is gathered, people can make their dinner plans based on the feedback received so far about dinner at a particular location.

Potentially, Foodback may enable dining authorities to respond to a particular review in a transparent setting, where all users can view the response.

The purposes of Foodback are as follows:

1. Enable people subscribed to the MIT meal plan to decide where to eat.
   This is done on a rolling basis during meal times. As a subscriber has a meal, he submits a review about the meal at that particular location. This updates the rating of the food in real time, so other users get more information as more users get heir meals.
2. Enable dining authorities (individuals in charge of overseeing meals in each location) to receive feedback on the quality of the service they are providing.
3. Enable people not on a meal plan (for example guests) to decide where to eat. The reviews on Foodback are intended to be openly accessible, even though only MIT students are allowed to post reviews/vote on them.
4. Provide transparency in dealing with dining hall issues.
   Right now, the feedback system is done in one of two ways: physical cards that are put on bulletin boards in each dining location and an online feedback form. Both have limitations. In the former, reviews are restricted to a particular dining hall and only those who will walk up to the boards can see the reviews. In the later, others do not get to see the feedback/suggestions. Thus there is a general lack of transparency when it comes to seeing the measures that dining authorities are taking in order to address these issues.
5. (Potentially) Allowing dining authorities to respond to reviews and feedback.
   For this feature, we might have to divide users into one of two categories, diners and authorities. The idea is to keep all interaction transparent so subscribers know what issues are being worked on.

Context diagram



Concepts

1. Dining location:

This refers to the dining location for which the review is being submitted. In our implementation, we will use the dining hall names (Baker Dining, Next Dining, Mc Cormick Dining, Simmons Dining, Howard dining hall at Maseeh).

2. Feedback window:

The feedback window refers to the time of the day for which a review is being submitted (Breakfast, Brunch, Lunch, Dinner, Late Night). The menus vary by time of the day as well as location. Thus, depending on the meal periods during which each dining location operates, we will have a "name" (for example: Dinner at Simmons) so that all reviews for the meal "Dinner" at location "Simmons" can be processed together.

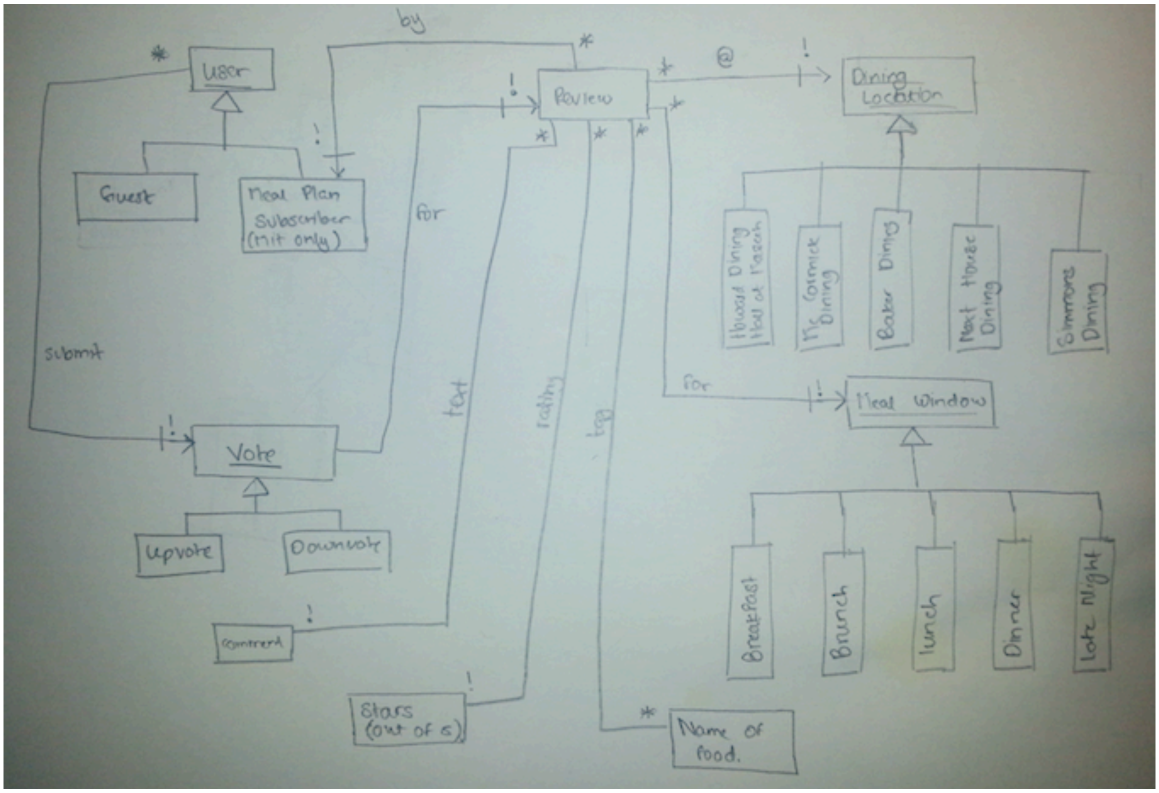
3. Review:

This is a piece of feedback submitted by a meal plan subscriber. Since we are using certificates to identify users, only MIT students can submit these reviews. So, for guests, the person guest-passing them will have to collect their reviews.

4. Upvote/Downvote:

This provides a way for the people reading the reviews to rate the usefulness of a particular review. Thus, the viewer will be able to see the most useful reviews as determined by the visitors of the website.

Data model



Design challenges

<u>1. Authenticating users:</u>

We need a way to prevent spamming on the web application. For example, user outside of MIT should not be allowed to post comments/spam the real users with spam reviews.

Proposed solutions:

- Allow users who have an MIT email to sign up for a Foodback account where they can submit/view feedback. However, with this approach, users can also create mailing lists and use the mailing list address to create a Foodback account (which will generate "dummy" accounts). However, this approach enables users to create account names that allow them to not be identified.
- Enabling users to use their MIT certificate to sign-in to the system. This allows users to sign-in without creating an account but we can also include the added feature of alowing users to also have a user-name. A viewer, can only see the reviewer's username (this is to allow a reviewer to speak his mind without being tracked by other users).

<u>2. Keeping the decision-making simple:</u>

A key idea is to keep it simple. A user should be able to quickly make a decision based on the reviews.

Proposed solutions:

- Enable users to comment on reviews so that users can address particular things they liked or disliked about a review. A disadvantage of this approach is that the threads can become very long and cumbersome, defeating the purpose of enabling users to decide quickly. However, it allows users to see exactly what other diners agreed on/challenged.
- Another approach is to allow users to upvote and downvote reviews based on how accurate and useful they are. Upvoting a review will push it further up in the list or reviews and downvoting, down in the list. An advantage of this approach is that the reviews are arranged by usefulness, while at the same time, keeping the posts easy to read (a user does not have to dive into the comments linked to a post).

3. Editing/Deleting reviews:

It is important to allow users to change their reviews and not deter them from submitting a review by setting it in stone, especially if their identity is linked to the post.

Proposed solution:

- Mask the identity of the reviewer (a user cannot tell who posted the review). Thus a reviewer and easily submit a post and not worry about repercussions. However, if the reviewer changes his/her mind about a meal, he cannot update his post to give other diners a more accurate picture.
- Enable the users to delete/update their posts. Since we will be rating the meal at a proposed location, it is important to also update the rating as a post is edited. This might make the implementation a little harder to code but will give users a more accurate picture and make them more comfortable to post a review.

4. Limiting the number of reviews:

Even after restricting the application to the MIT community, it is important to prevent other users from spamming other users with too many reviews to read for a particular meal.

Proposed solutions:

- We could also rely on our users for good behavior, that is, to post a reasonable number of reviews. However, this does not provide protection against malicious users who might want to spam the website. This might also cause issues with upvoting/downvoting where a malicious user would upvote least useful reviews, to confuse honest users.
- Since a user is only allowed to eat at one location at a particular time, we could restrict the number of reviews for a particular meal (for example, dinner on Monday) by a particular user, to one. Thus, a user can only post a single review for a period of time, and tag the particular food items he/she tried. Upvoting and downvoting should also be limited in the same way, a user must be authenticated, and can only submit one single vote on a particular review. This attempts to keep the voting/reviewing system closed within the MIT community and as accurate as possible.