# Foodback MIT

*Design Document - Phase 1*

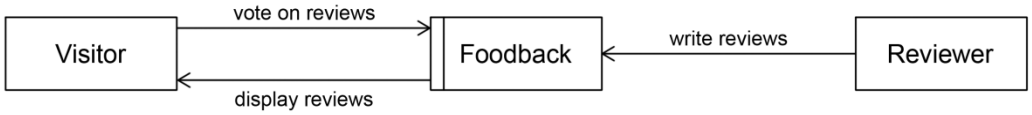
## Motivation

Foodback MIT (or just "Foodback" within the MIT community) is a web application that allows members of the MIT community to post and read reviews of the MIT dining halls. An MIT certificate is required to post a review, but the posted reviews are fully public.

The purposes of Foodback include:

1. **Help MIT meal plan subscribers decide where to eat.** By reading reviews from other members of the MIT community, meal plan subscribers can better evaluate their options and are better able to decide which dining hall they would most like to go to. Foodback will allow users to read reviews from only the current meal period so they can learn what people think about current offerings—such as the stir fry special at McCormick, which changes daily.
2. **Help guests to MIT decide where to eat.** Guests generally have little to no knowledge of the MIT dining halls. By reading the reviews for each dining hall before their visit, guests will be able to determine which dining hall best suits their tastes.
3. **Allow MIT dining hall staff to see how they're doing.** Since the Foodback website is open to the public, the staff of each dining hall can read their respective reviews and determine what they're doing well and what they can improve.
4. **Allow MIT students to voice their opinions on the dining halls.** Currently, students can provide feedback to the dining halls via paper comment cards or an online form. Foodback seeks to improve upon both of these methods. The website will allow students to leave feedback from any location, unlike the comment cards which are physically located in the dining halls. The website will also have a more user-friendly and intuitive UI than the official dining feedback form, and the submitted reviews will be public, rather than disappearing into the "black hole" of the current feedback submission box.
5. **(Potentially) allow dining hall staff to respond to reviews.** Foodback may also offer special staff accounts, through which dining hall staff can respond to their reviews.
6. **(Potentially) recommend a dining hall or specific dish.** The app may also serve as a way for users to laud certain dishes or services they encountered at dining halls. The app can display menus with dishes highlighted or dulled based on their popularity among diners.

## Context Diagram




## Concepts

**Dining Hall:** The specific MIT dining hall for which a review is submitted. A dining hall is associated with the reviews that are posted for it, and has a name that distinguishes it from other dining halls in the UI.

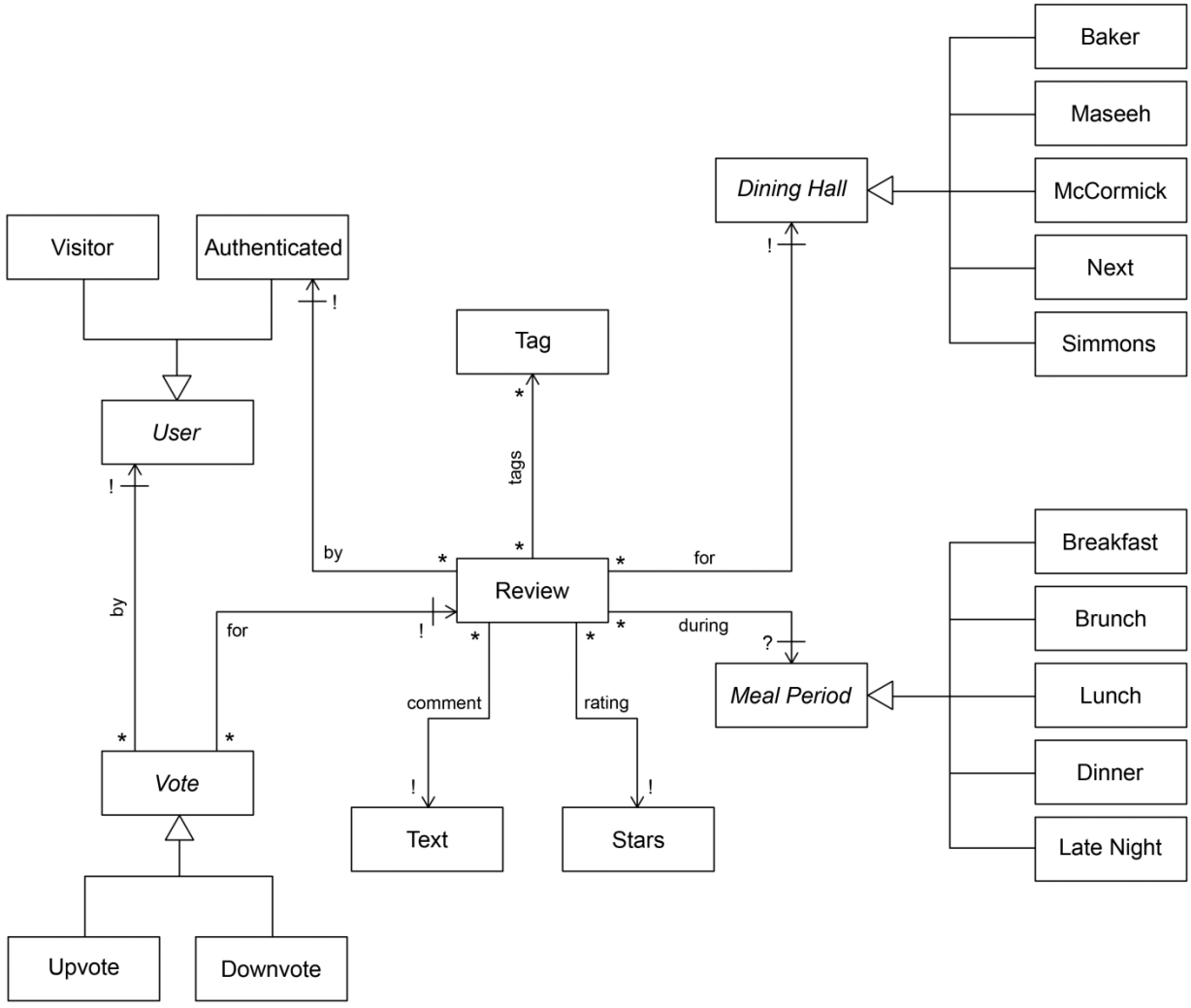
**Meal Period:** A period of time during which a dining hall offers a certain meal (breakfast, brunch, lunch, dinner, late night) and therefore has a certain menu. The available meal periods depend on the dining hall (only Maseeh offers lunch and only Simmons offers late night) and the timespan covered will also depend on the schedules of particular dining halls. Like dining halls, meal periods are identified by name and are associated with reviews.

**Review:** A piece of feedback submitted by a member of the MIT community. Aside from the dining hall, a review is associated with the user who posted it, a star rating, and some textual comment; and optionally, the meal period, or other customized tags specified by the author. Unlike the physical comment cards that are posted on the wall in dining halls, reviews are meant to be more substantial in their content.

**Votes (Upvote/Downvote):** Visitors to the Foodback website can vote on the helpfulness of reviews. This allows Foodback to display the most helpful reviews first. Naturally, votes are associated with specific reviews. They are closely related to the concept of **score**, which is the number of upvotes on a review minus the number of downvotes. Review scores allow for the highlighting of accurate portrayals of dining while weeding out inaccurate ones.

**Scoping:** Reviews are given within a *scope*, which is a way to set different levels of specificity in a review. The top level, which is required, is the dining hall. The next level is designating the meal hour (breakfast, brunch, lunch, dinner, late night). Next is the topic. Some are predefined – food, atmosphere, or service – while others can be user defined. Finally, the user can take another step to specify exactly what they wish to review. This is dependent on what the previous scope was. For example, if it is food, then the user can choose to highlight a specific dish. If service, then the user can comment on the hospitality of the staff. By using scoping, users can narrow what they read to specific subsets of reviews. It also helps to prevent incredibly vague reviews from cluttering the app. The level of scoping a review uses is up to the author.

## Data Model




## Design Challenges

**How do we authenticate users?** 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.

Potential 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 allowing 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).

**How do we help users make decisions quickly?** A key idea is to keep it simple, so that users are encouraged to use Foodback. A user should be able to quickly make a decision based on the reviews, and for this to be possible, the reviews must be easy to go through.

Potential 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).

**Can reviews be edited or deleted?** When a user leaves a comment card or submits to the online feedback form, that action cannot be taken back. However, it also (presumably) does not have permanent effects; comment cards are removed after a while, and feedback form submissions are never posted publicly in the first place. Foodback reviews should naturally be considered separately.

Potential Solutions:

*We will consider editing and deleting together, even though they are different operations, because they can be used to effectively replace each other. Deleting a review and then posting a new one in its place is almost the same as editing; editing a review to remove all of the text (or replace it with something like "DELETED" if empty reviews are not allowed) is almost the same as deleting.*

- Allow editing and deleting. This encourages users to post reviews freely, since they can edit or delete the review if they change their mind. As a result, it may increase participation.
- Do not allow editing and deleting. This may simplify the implementation of the application. It also encourages users to think twice about what they write.

**Should information about a reviewer be publicly displayed?** By authenticating reviewers with their MIT certificates, we have access to their names and Kerberos IDs. We may or may not want to display some of this information with the review.

Potential Solutions:

- Do not display any identifying information. This is the most effective for protecting user information. Users will feel more comfortable leaving honest reviews if they know that any offended parties cannot identify them.
- Display the user's first name publicly, and:

- No other information. Visitors will feel like the review is more personal and believable if there is a name next to it. A first name only can achieve this purpose fairly well. Users who submit reviews will also feel a little more accountable if their names are displayed.
- The user's last name or Kerberos ID when the reader is signed in. These should be considered together because, in most cases, a full name can be used to find a Kerberos ID, and a Kerberos ID can be used to find a full name. This maximizes accountability while also minimizing the exposure of user information to non-MIT visitors.

**How many reviews can a user submit?** If a particular user submits many reviews, it is possible for them to dominate the review system and make their opinion appear to be more prevalent than it really is.

Potential Solutions:

- Do not limit reviews. If a user submits many reviews, they are probably very passionate about the topic of MIT dining. We may actually want them to have more weight in the discussion because they might be better informed than other users. This eases the implementation, but would open the website up to spam.
- Limit reviews to one per user per meal period each day. A user may very well have something to say about the dining halls during each meal period. We should allow the posting of that number of reviews. By allowing no more than that many reviews, we can reduce spam and allow other users' voices to be heard.

**How are reviews organized?** For reviews to be useful to other users, there must be a way for users to find them in a structured, non-convoluted way. Using simple categories intuitively seemed the obvious way to go, but it became clear that reviews might focus on a whole dining hall or a very specific dish. That is, some categories would inevitably have a large share of reviews. Thus, using multiple categories seemed like the next best thing. However this was also troublesome. For example, most dining halls operate independently. They have different chefs, equipment, etc. Thus, a negative review on food without specifying which dining hall the meal was had in would negatively affect other dining halls, an unwanted side effect. Thus the idea of hierarchical categories, a.k.a. scoping, would solve the problem. Now a review on service would be forced to specify which meal (breakfast, brunch, etc.) the service was encountered, and which dining hall.

**How are reviews responded to?** There were two routes for this: comment thread or a voting system. Each had their pros and cons. A comment thread would allow the reader to get more descriptive perceptions on how the community feels about the review. Votes are purely quantitative, aside from the positive/negative connotation attributed to each vote. However, for a community the size of MIT dining, any one user could find themselves reading a long comment thread, yet be indecisive on the issue. Votes on the other hand are easily decisive. The user only has to take into account what the review says and the net reaction from the community (represented as a number) to make a decision, as opposed to sifting through a whole thread.