Slide 2:

This is the overview of what our application is meant to accomplish. The main premise is to present users with a custom menu that will cater to their dietary needs, as well as, providing users with easy access to restaurant information, and specials in a close proximity.

Slide 3:

This is a good example of a use case where the user does have dietary restrictions. Even though this application is geared towards people with dietary restrictions, it does offer complete functionality to the general population.

Slide 4:

- We created two native apps
- Used MVC
- Clients server running MySQL was used for back end

Slide 5:

In this slide we are showing how we would have liked to develop the application.

Slide 6:

The first challenge we encountered while developing was trying to convert all of our landscape oriented views to portrait. This proved to be more difficult in iOS than we originally imagined. We ended up having to redo a few of our views because our original layout did not scale to portrait well.

Another challenge we encountered was attempting to create consistency between existing designs and our designs.

Slide 7:

We did not have any true management problems. We all took care of the issues, we met often, and our client was easy to get a hold of.

Slide 8:

We did a lot of pair programming when we were attempting to learn something new. Meeting often helped keep the group organized and having a single client contact solved a lot of potential confusion with communication.

Slide 9:
Objective-C

- iOS development

MySQL Trigger syntax

- For example we used a trigger to update a restaurants average rating.
- Updating the likes for a review.

Proper software engineering techniques

- Reactive extensions

Proper Github source control

- 301 taught us the basics, but working on this project expanded our Git knowledge

Slide 10:

- Prateek made a custom drawer for debugging our application while we were testing.
- The debug drawer lets us inspect layouts, enable mock mode, enable logging etc.
- Dependency injection - dagger
- We were able to ship faster by leaning on open source projects. When needed, we came up with our own implementations, e.g. the

Slide 11:

Develop the views in both orientations as we went along rather than choosing a single orientation.
Develop both applications simultaneously.
Better separation of concerns with respect to the business logic.

Slide 12:

Describe our plan to centralize our business logic into web services. However, we did not have the means to do so since we did not have access to the client's web server.

Slide 13:

Describes the structure of the back-end architecture had we designed it.