Final presentation.

Project rundown.  
 Features:

-Solid backend api running on AWS cloud compute severs.  
 -API serves any number of businesses.  
 -

Development Process:  
 Github.

Throughout development, we used Github as a remote code repository and a version control system. Updates to the code were developed on independent branches, then merged to the shared development branch until we were confident the code was stable enough to push to the master branch.

Clickup.

Clickup was our main task management tool for the project, any work that needed to be completed was placed here. The Clickup boards were set up for an Agile-Sprint task flow, with team members taking ‘open’ tasks, assigning them to themselves, then placing them in ‘complete’ after their code had been reviewed and merged into the development branch.

Maven.

Used to automate the building of the java based backend.

Node PackageManager

Used to automate the frontend demo builds.

Discord.

Team communication, voice meetings and text chats.

CircleCI

Continuous integration and continuous deployment platform.

Retrospective.

What was interesting about the project.

What worked well?

- Product owner seemed satisfied by the end result.

- The team communicated well and worked hard to complete tasks

- Team members picked their own tasks well.

What could be improved?

- Time estimates could have been better.

- There are some documented bugs that could be fixed

What was a surprise?

- Surprised by hard hard some of the technolgy was to learn.

- Some of the backend entities needed to be changed a few times during development

What did we learn?

- React and that React sucks.

- Single page application development

- How to write code that integrates into other peoples code.

- How to use mockito

- Docker

Smells

- The way the Admin and Customer classes interact with the User class might cause issues if the project needs to be extended in the future.

- The frontend and backend both use a lot of loops, this may cause issues as the application handles large amounts of data.

- Possible code duplication between controllers, especially for error messages.  
-HSQLDB is a possible bottleneck for large scale deployments of the app. It would be worth increasing the abstraction between the backend and the database for larger deployments. Possibly dynamically starting new DB machines for different groups of businesses to reduce load on the current single machine system.