Team Aqua

Alac, Calvin, Robert, Tony, & Winnie
Deliverable 3

Agenda

1) Product Backlog
2) Sprint Backlog
3) Team Admin: Task Board, Burn Down Chart & Git Repo
4) System Designs
5) Demo of First Release

Product Backlog

Product Backlog

Main features/grouping in our backlog:

1. **Login & Sign Up**: Customers & restaurant personnel
2. **Connecting with Customers**: Discussion boards, etc
3. **Restaurant Profiles**: Restaurant info, menus, etc
4. **Placing Orders**
5. **Accepting Orders**
6. **Search Engine**
7. **Specials & Discounts**
8. **Restaurant Recipes**
9. **Restaurant Reviews & Favourite Restaurants**
10. **Platform Analytics**

Sprint Backlog

Sprint 1 Backlog (Jun 21-28)

Designing Sprint Backlogs

- **Point System**
  - Scale: 1(Easy) to 10 (Super hard)
  - Experience/Familiarity
  - Implementation length
- **Priority System**
  - Prioritized by client requirements
  - Also prioritized to create a MVP
- **Assignment of Tasks**
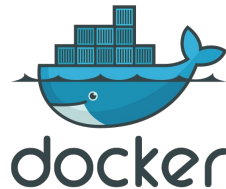 - Based off team member's ability and comfort
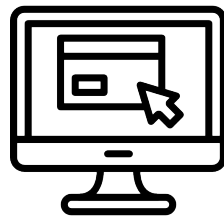
Features/Tasks worked On:

- **Back-end**
  - Docker for Angular & Django
  - System Designs
- **Front-end**
  - Home page UI
  - Customer login & signup
  - Payment UI
- **Deployment of Web Page**

Admin: Task Board

Sprint Task Board

Alac:
- Set up of MongoDB & login info

Calvin:
- UML/Component diagram & sign up info

Robert:
- Django + Docker & deploy webpage

Tony:
- Login, Signup & Payment UI

Winnie:
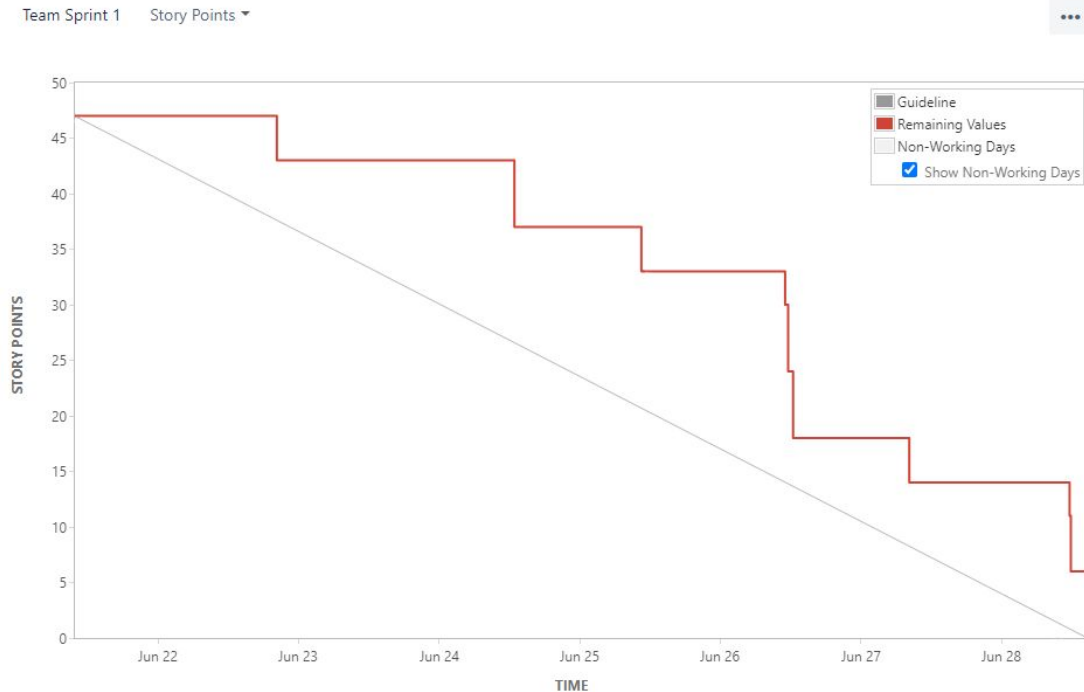- Angular & Docker & UI for home page

Task Reorganization

- Auth0 (authentication service) moved from Django to Angular
  - Resulted in Alac's task being reassigned to Winnie
- Auth0 covered sign up and log in
  - Resulted in Tony's task being reassigned to Winnie
  - Converted sign up UI experience to create payment UI
- Creating component diagram
  - Alac went to assist Calvin

Admin: Burn Down Chart
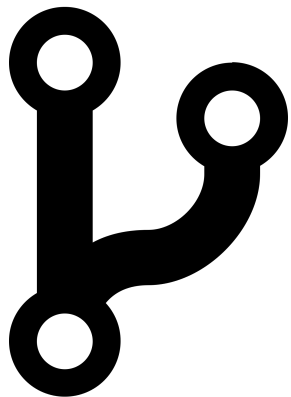
Sprint Burndown Chart

- Sprint 1 began June 21st @5pm and ended June 28th @4:59pm
- Total story points done in this sprint was 41

Admin: Git Repo

Git Repo - Branching Strategy

Master

- Contains working code for each deliverable

Development

- Contains features that have been approved by reviewed by at least 1 other team member

Feature Branches

- 1 branch per feature
- Branch name will be *TASK-ID/branch-name*
- Will be merged into development by pull request
- Will be deleted after merging into development

Git Repo - Commits

Types of File Committed
- Will use the auto generated .gitignore for Angular and Django to determine files to exclude

Commit Messages
- Messages should have the format of *TASK-ID: what was completed* going forward

Documentation
- We will be using Docusaurus for front-end and back-end documentation which will be stored in the repo

Git Repo - Commit Rules

Committing to the Master Branch
- Can only directly commit to master branch when committing deliverable materials (report & slides) or daily task board and burndown chart snapshots

All Other Commits
- Commit must be done on the feature branch corresponding to that feature
- The feature will be merged into the development or master branch via pull request
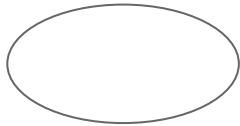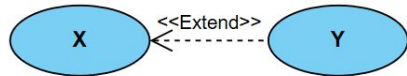
System Designs
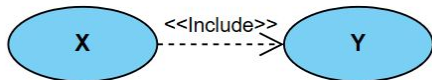
System Design - UML

Legend

- Actor
- Use Case
- Link
- Extends
- Generalization
- Includes

Link to UML Use Case Diagram:

https://github.com/CSCC01/team_08-project/blob/master/Deliverable%203/UML_Use_Case.pdf
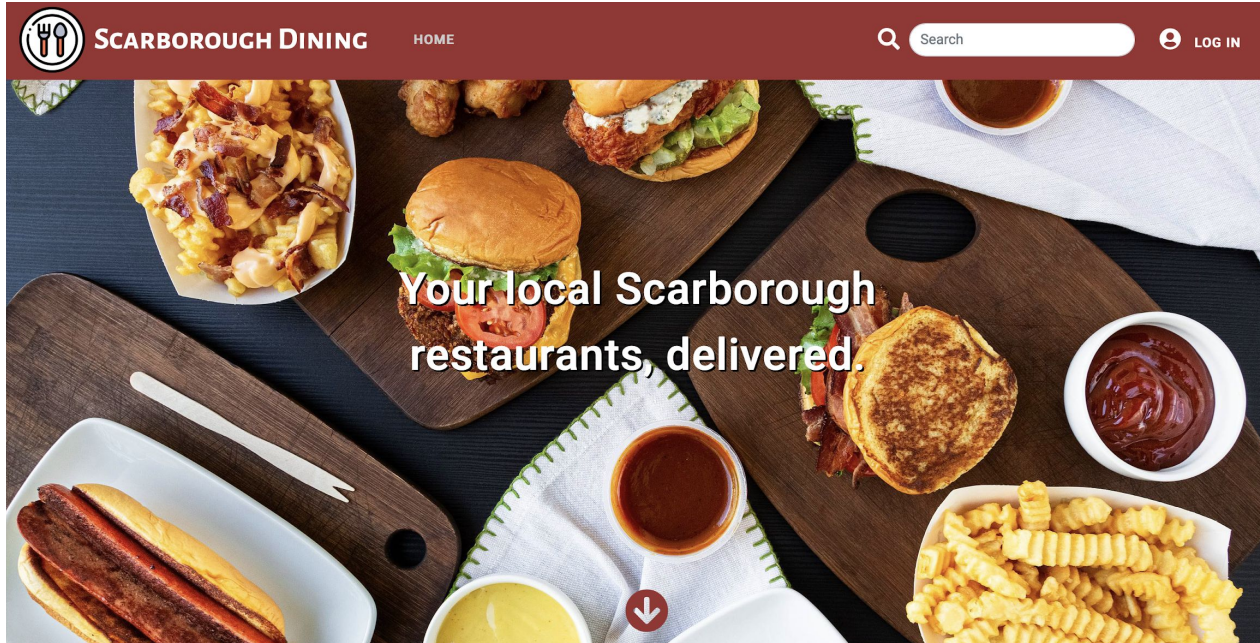
System Design - Components

Link to Component diagram:

https://github.com/CSCC01/team_08-project/blob/master/Deliverable%203/SCDining_Component_Diagram.pdf

Demo of First Release

Introducing Scarborough Dining Live

Live Webpage URL: https://3.18.84.86/

The End