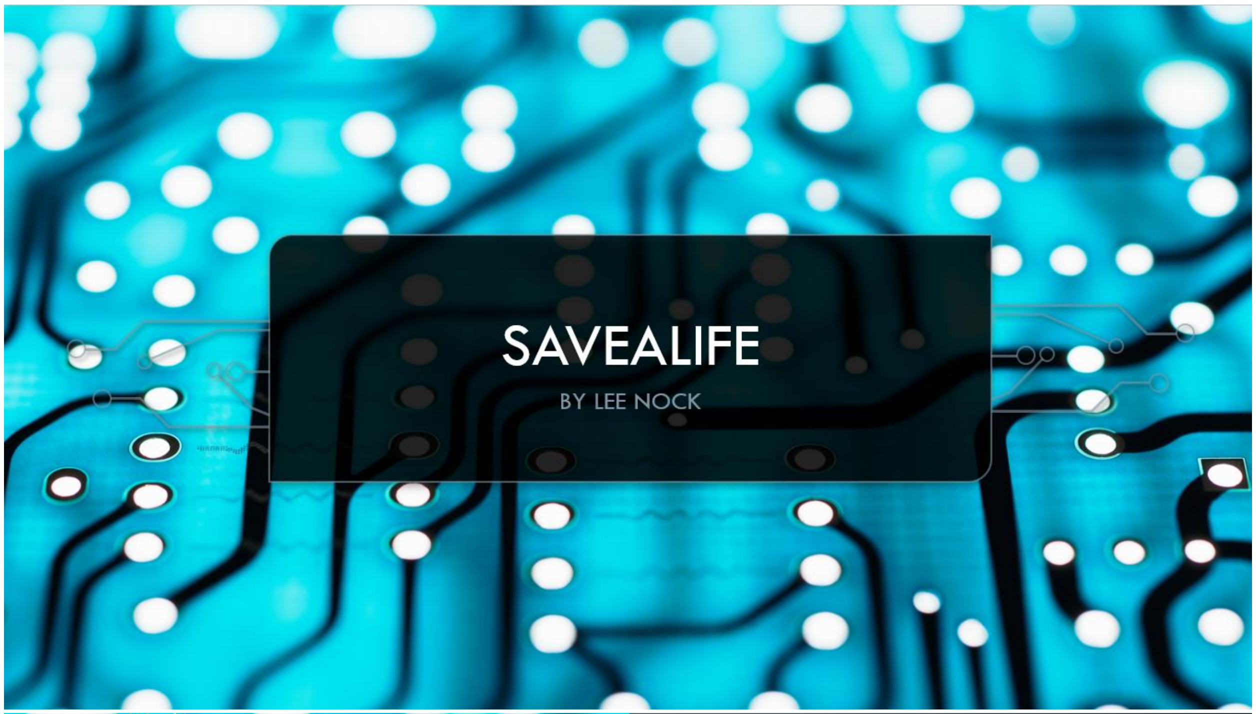
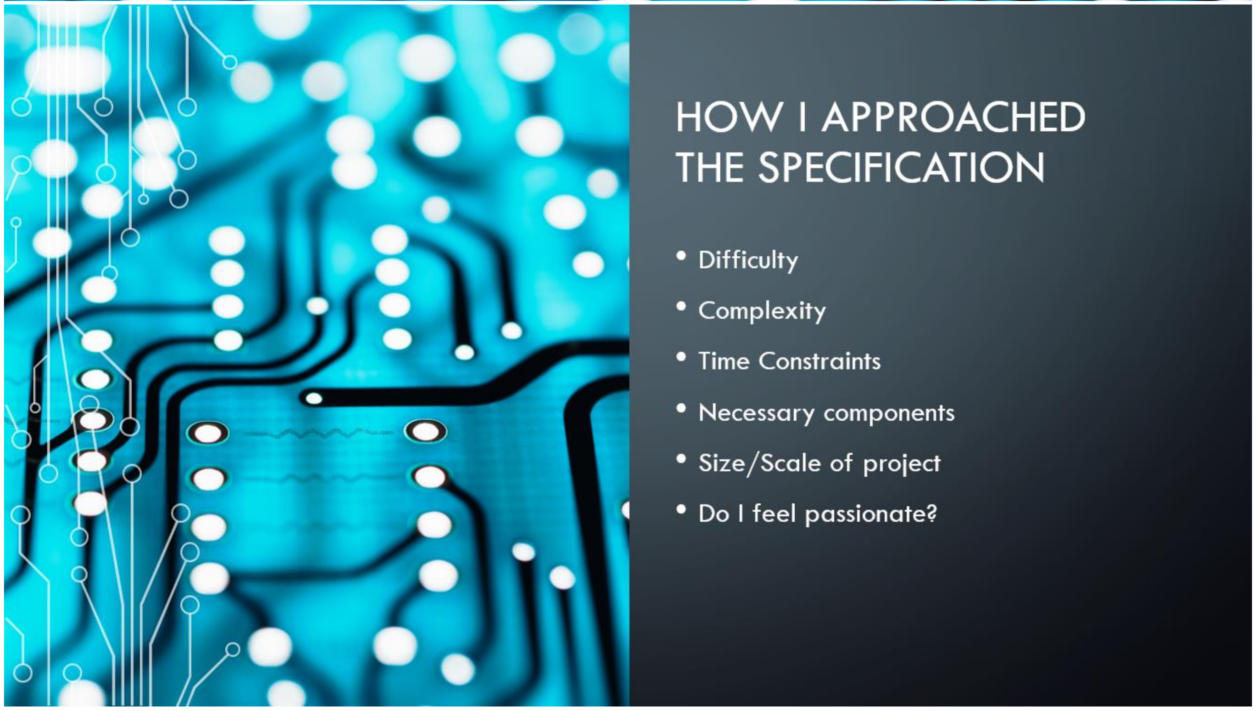
# SaveALife

By Lee Nock



SAVEALIFE

BY LEE NOCK

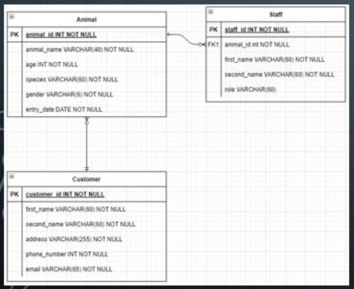


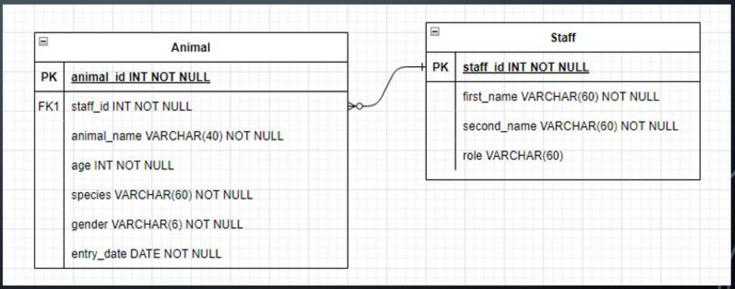
## HOW I APPROACHED THE SPECIFICATION

- Difficulty
- Complexity
- Time Constraints
- Necessary components
- Size/Scale of project
- Do I feel passionate?

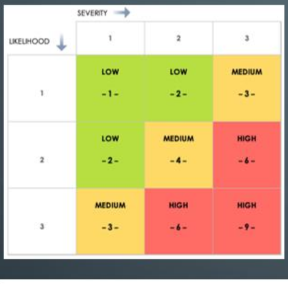
## CONCEPT

- Produced an idea (Animal Shelter)
- Considered alternatives
- Considered potential complications
- Considered risks (Shown on next slide)
- Created an ERD… Or two


Animal
PK animal_id INT NOT NULL
animal_name VARCHAR(40) NOT NULL
age INT NOT NULL
species VARCHAR(60) NOT NULL
gender VARCHAR(6) NOT NULL
entry_date DATE NOT NULL
Staff
PK staff_id INT NOT NULL
FK1 animal_id int NOT NULL
first_name VARCHAR(60) NOT NULL
second_name VARCHAR(60) NOT NULL
role VARCHAR(60)
Customer
PK customer_id INT NOT NULL
first_name VARCHAR(60) NOT NULL
second_name VARCHAR(60) NOT NULL
address VARCHAR(255) NOT NULL
phone_number INT NOT NULL
email VARCHAR(65) NOT NULL


Animal
PK animal_id INT NOT NULL
FK1 staff_id INT NOT NULL
animal_name VARCHAR(40) NOT NULL
age INT NOT NULL
species VARCHAR(60) NOT NULL
gender VARCHAR(6) NOT NULL
entry_date DATE NOT NULL
Staff
PK staff_id INT NOT NULL
first_name VARCHAR(60) NOT NULL
second_name VARCHAR(60) NOT NULL
role VARCHAR(60)

## RISK ASSESSMENT



| LIKELIHOOD ↓ \ SEVERITY → | 1 | 2 | 3 |
|---|---|---|---|
| 1 | LOW -1- | LOW -2- | MEDIUM -3- |
| 2 | LOW -2- | MEDIUM -4- | HIGH -6- |
| 3 | MEDIUM -3- | HIGH -6- | HIGH -9- |

**Risk Score: 6**
**Risk: Poor Productivity**
Whilst being lazy is unacceptable, poor productivity can also occur because of burnout. It is possible that overworking without regular breaks can create issues for the developer. Potentially, this could lower the quality of their code or prevent them from moving forward and hitting scheduling targets. To mitigate this, it is important that the developer sticks to a plan and considers their personal time carefully alongside this plan. Taking regular breaks allows a person to work more effectively.

**Risk Score: 6**
**Risk: Tight Scheduling**
To mitigate the problem, we must focus on highly effective planning. It will be important to ensure that the project hits the criteria of an MVP before concerning ourselves with additional features. Sometimes, things like bank holidays may effect the total hours of work or preparation for the project.

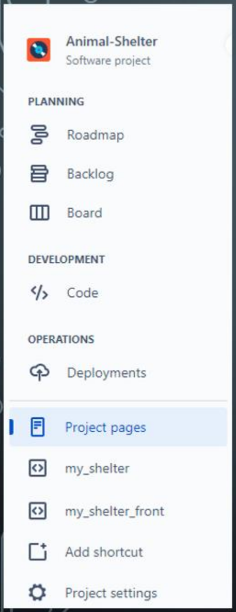
**Risk Score: 4**
**Risk: Technical Difficulties**
This problem could only be mitigated by taking the time to prepare back-up systems and methods. For example, if a computer isn't working, maybe we use a laptop. If there's a possibility our connection could fail, we need a back-up service in place for that.

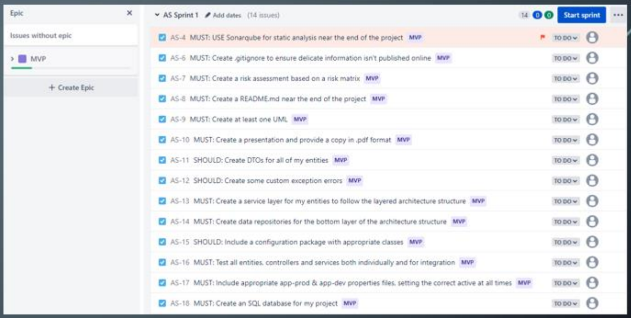
**Risk Score: 6**
**Risk: Inaccurate estimations**
This may fall slightly in the range of tight scheduling. However, it is important to note that if we do not correctly estimate the timing of our actions, we may introduce new issues. On the other hand, we may incorrectly assume the correct software requirements for a given task, taking us back to the planning stage or presenting poor solutions.

**Risk Score: 4**
**Risk: Poor Quality code**
It is particularly important to try to mitigate this issue on a project conducted by an individual, myself in this case. Having the input of only one developer can create a very tunnel visioned approach with a variation of issues. To mitigate the problem, I collaborated with some of my peers and considered their ideas.

## SPRINT PLAN

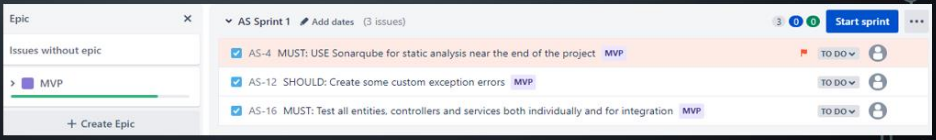
- Used Jira for backlogs/sprint board
- Used MoSCoW for prioritising tasks
- Connected Repo for issue updates





(14 issues)

- AS-4 MUST: USE Sonarqube for static analysis near the end of the project MVP TO DO
- AS-6 MUST: Create .gitignore to ensure delicate information isn't published online MVP TO DO
- AS-7 MUST: Create a risk assessment based on a risk matrix MVP TO DO
- AS-8 MUST: Create a README.md near the end of the project MVP TO DO
- AS-9 MUST: Create at least one UML MVP TO DO
- AS-10 MUST: Create a presentation and provide a copy in .pdf format MVP TO DO
- AS-11 SHOULD: Create DTOs for all of my entities MVP TO DO
- AS-12 SHOULD: Create some custom exception errors MVP TO DO
- AS-13 MUST: Create a service layer for my entities to follow the layered architecture structure MVP TO DO
- AS-14 MUST: Create data repositories for the bottom layer of the architecture structure MVP TO DO
- AS-15 SHOULD: Include a configuration package with appropriate classes MVP TO DO
- AS-16 MUST: Test all entities, controllers and services both individually and for integration MVP TO DO
- AS-17 MUST: Include appropriate app-prod & app-dev properties files, setting the correct active at all times MVP TO DO
- AS-18 MUST: Create an SQL database for my project MVP TO DO

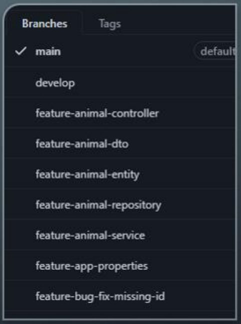


(3 issues)

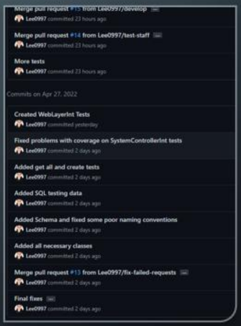
- AS-4 MUST: USE Sonarqube for static analysis near the end of the project MVP TO DO
- AS-12 SHOULD: Create some custom exception errors MVP TO DO
- AS-16 MUST: Test all entities, controllers and services both individually and for integration MVP TO DO

## CI - GIT

- Used git to create 'feature' branches and 'test' branches
- Used descriptions and regular commits
- Only pushed to main if the develop branch contained working code
- Only pushed to main from develop



- main
- develop
- feature-animal-controller
- feature-animal-dto
- feature-animal-entity
- feature-animal-repository
- feature-animal-service
- feature-app-properties
- feature-bug-fix-missing-id



Merge pull request #15 from Lee0997/develop
Lee0997 committed 23 hours ago

Merge pull request #14 from Lee0997/test-staff
Lee0997 committed 23 hours ago

More tests
Lee0997 committed 23 hours ago

Commits on Apr 27, 2022

Created WebLayerInt Tests
Lee0997 committed yesterday

Fixed problems with coverage on SystemControllerInt tests
Lee0997 committed 2 days ago

Added get all and create tests
Lee0997 committed 2 days ago

Added SQL testing data
Lee0997 committed 2 days ago

Added Schema and fixed some poor naming conventions
Lee0997 committed 2 days ago

Added all necessary classes
Lee0997 committed 2 days ago

Merge pull request #13 from Lee0997/fix-failed-requests
Lee0997 committed 2 days ago

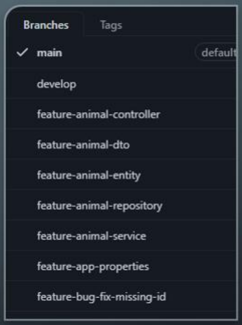
Final fixes
Lee0997 committed 2 days ago

## CONSULTANT JOURNEY

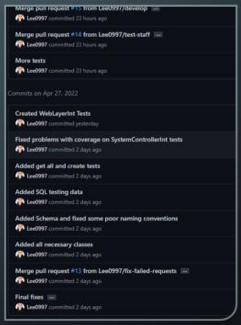
- Git / GitHub
- Jira
- MySQL
- Java
- Spring
- HTML, CSS, JavaScript
- Bootstrap
- Maven
- JUnit and Mockito

## CI - GIT

- Used git to create ‘feature’ branches and ‘test’ branches
- Used descriptions and regular commits
- Only pushed to main if the develop branch contained working code
- Only pushed to main from develop



main
develop
feature-animal-controller
feature-animal-dto
feature-animal-entity
feature-animal-repository
feature-animal-service
feature-app-properties
feature-bug-fix-missing-id

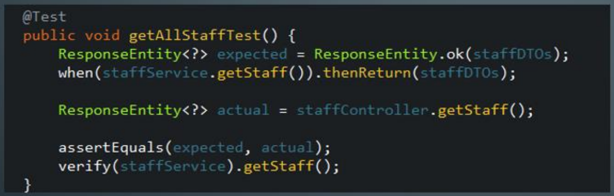


Merge pull request #15 from Lee0997/develop
Lee0997 committed 23 hours ago
Merge pull request #14 from Lee0997/test-staff
Lee0997 committed 23 hours ago
More tests
Lee0997 committed 23 hours ago

Commits on Apr 27, 2022

Created WebLayerInt Tests
Lee0997 committed yesterday
Fixed problems with coverage on SystemControllerInt tests
Lee0997 committed 2 days ago
Added get all and create tests
Lee0997 committed 2 days ago
Added SQL testing data
Lee0997 committed 2 days ago
Added Schema and fixed some poor naming conventions
Lee0997 committed 2 days ago
Added all necessary classes
Lee0997 committed 2 days ago
Merge pull request #13 from Lee0997/fix-failed-requests
Lee0997 committed 2 days ago
Final fixes
Lee0997 committed 2 days ago

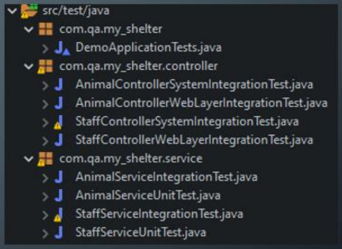
## TESTING



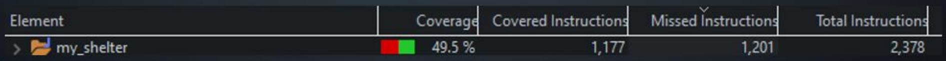
```java
@Test
public void getAllStaffTest() {
    ResponseEntity<?> expected = ResponseEntity.ok(staffDTOs);
    when(staffService.getStaff()).thenReturn(staffDTOs);

    ResponseEntity<?> actual = staffController.getStaff();

    assertEquals(expected, actual);
    verify(staffService).getStaff();
}
```

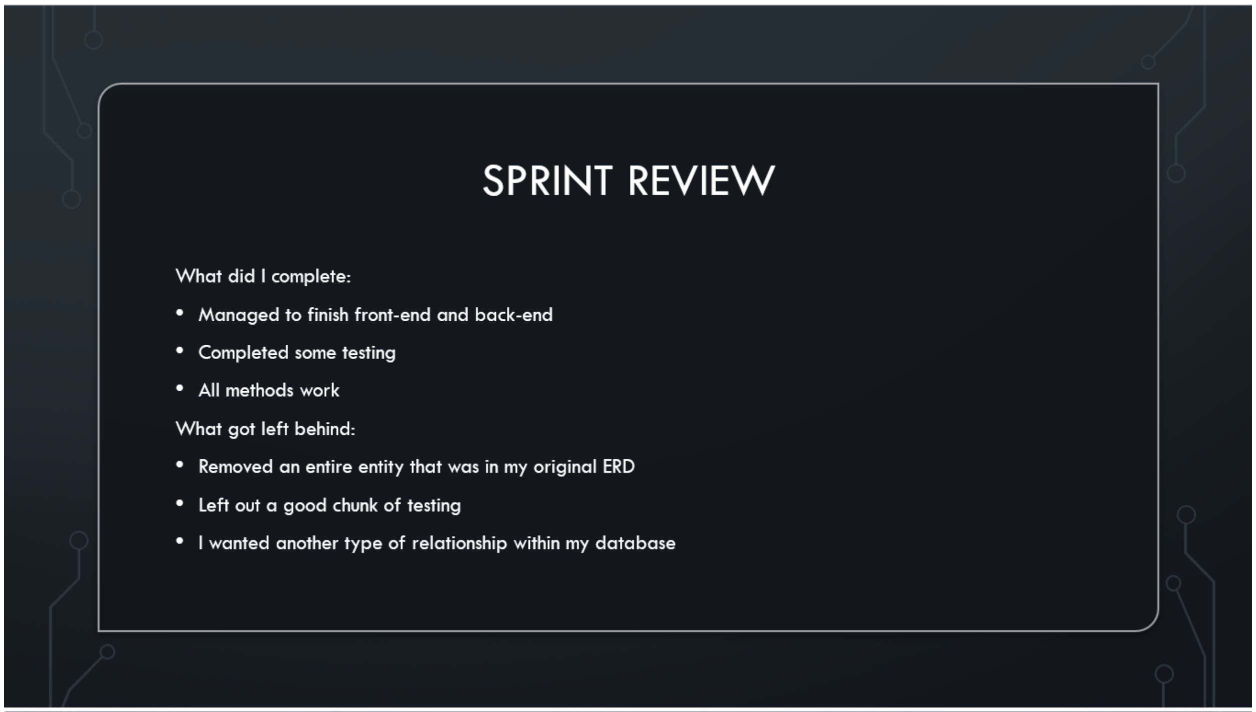


```
src/test/java
  com.qa.my_shelter
    DemoApplicationTests.java
  com.qa.my_shelter.controller
    AnimalControllerSystemIntegrationTest.java
    AnimalControllerWebLayerIntegrationTest.java
    StaffControllerSystemIntegrationTest.java
    StaffControllerWebLayerIntegrationTest.java
  com.qa.my_shelter.service
    AnimalServiceIntegrationTest.java
    AnimalServiceUnitTest.java
    StaffServiceIntegrationTest.java
    StaffServiceUnitTest.java
```



| Element | Coverage | Covered Instructions | Missed Instructions | Total Instructions |
|---|---|---|---|---|
| my_shelter | 🟥🟩 49.5 % | 1,177 | 1,201 | 2,378 |

## DEMO



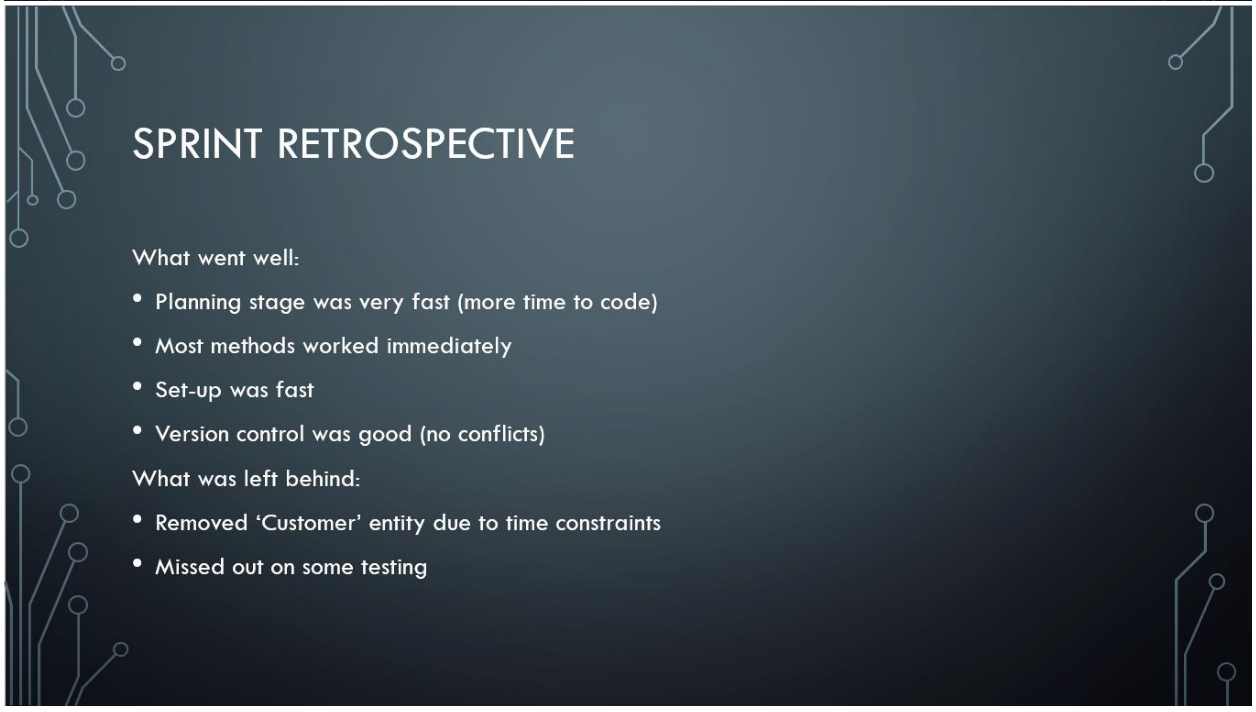
## SPRINT REVIEW

What did I complete:

- Managed to finish front-end and back-end
- Completed some testing
- All methods work

What got left behind:

- Removed an entire entity that was in my original ERD
- Left out a good chunk of testing
- I wanted another type of relationship within my database



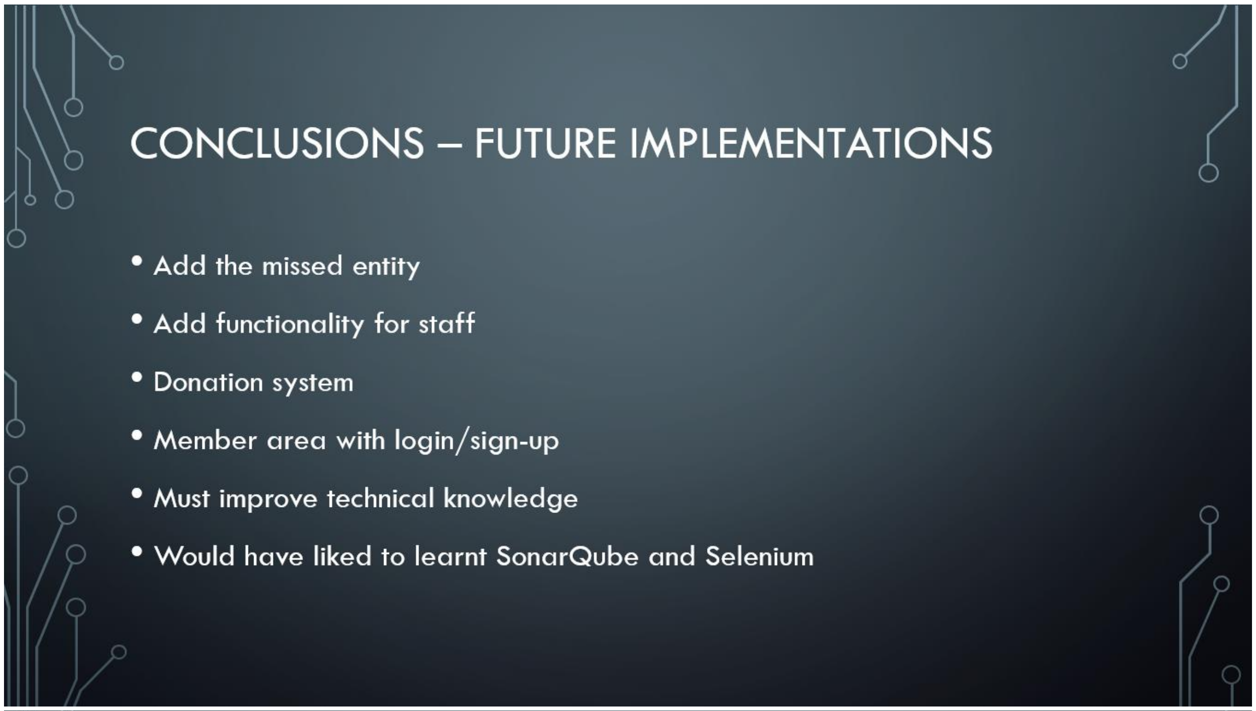
## SPRINT RETROSPECTIVE

What went well:

- Planning stage was very fast (more time to code)
- Most methods worked immediately
- Set-up was fast
- Version control was good (no conflicts)

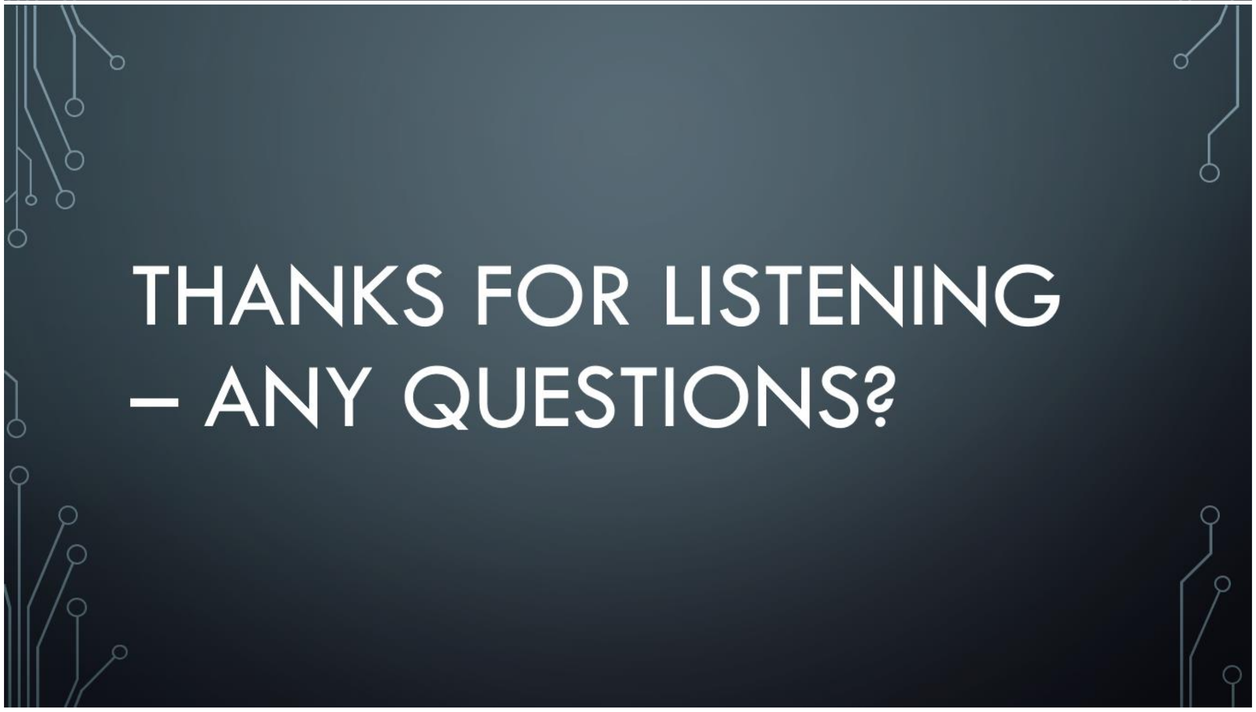
What was left behind:

- Removed 'Customer' entity due to time constraints
- Missed out on some testing



## CONCLUSIONS – FUTURE IMPLEMENTATIONS

- Add the missed entity
- Add functionality for staff
- Donation system
- Member area with login/sign-up
- Must improve technical knowledge
- Would have liked to learnt SonarQube and Selenium



## THANKS FOR LISTENING – ANY QUESTIONS?