Group 4 Comment Api

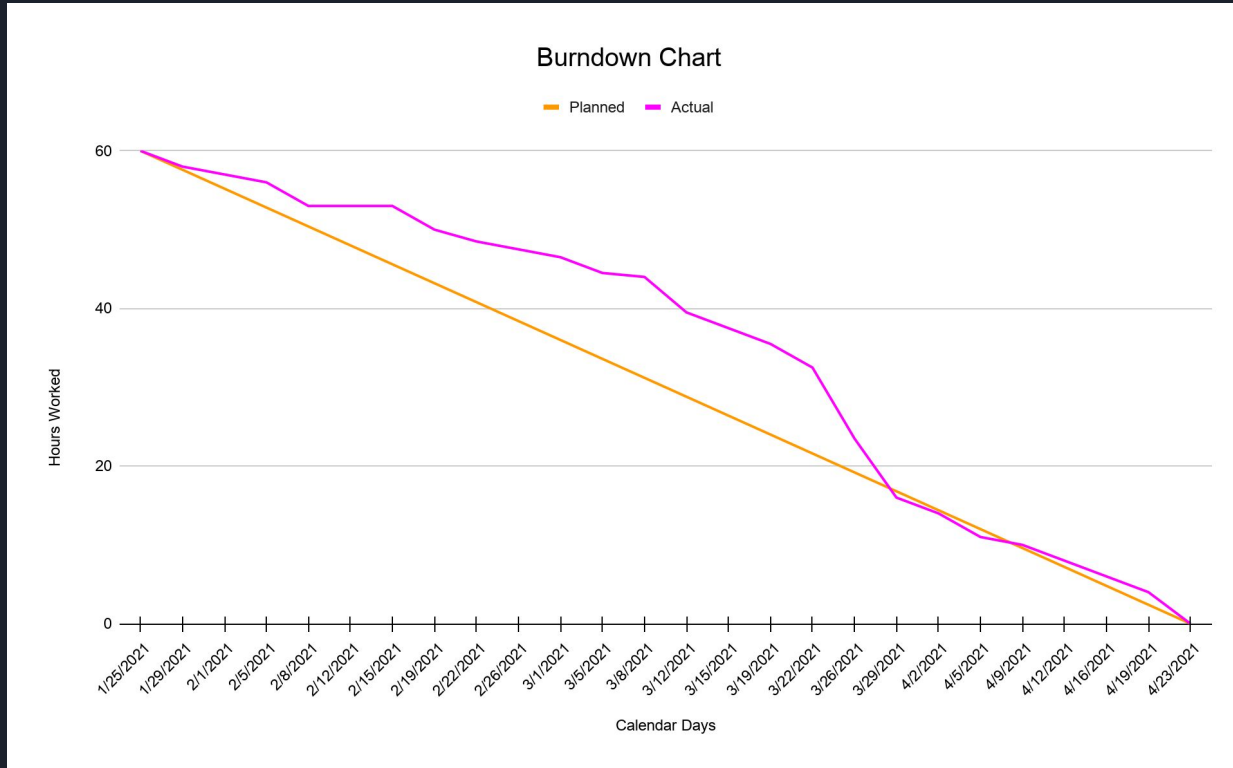Adam Saylor, Ryan Pallman, Alex Sutherland

Sprint 6 Review

What worked

- Polishing up features
- Address any remaining bugs
- Feedback from user testing

What did not work

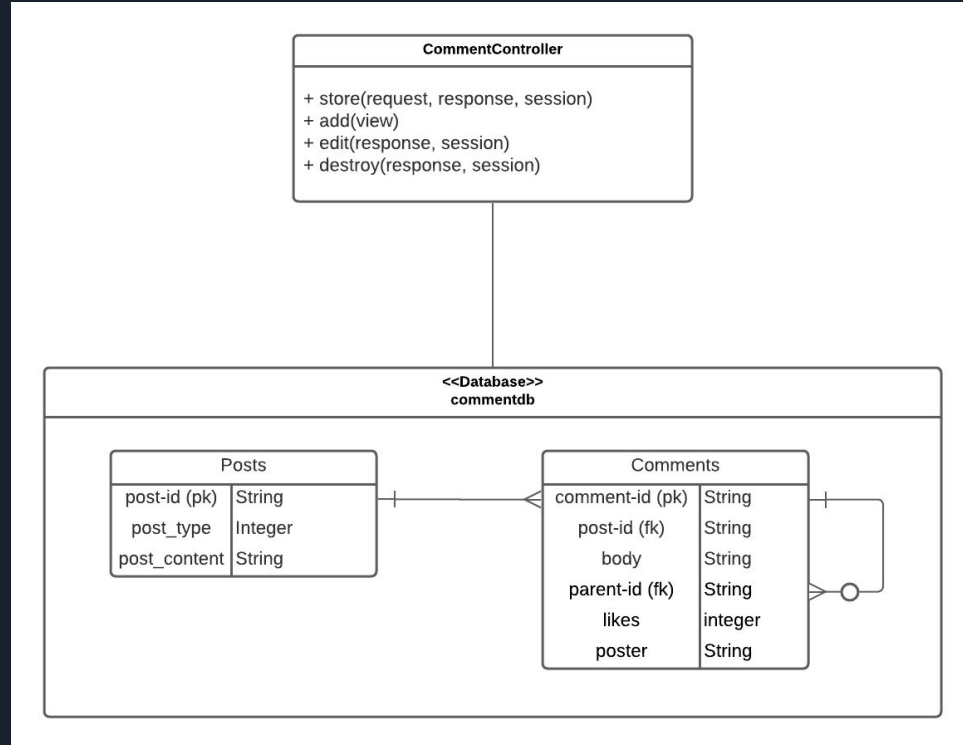- Hard to organize group time due to outside obligations
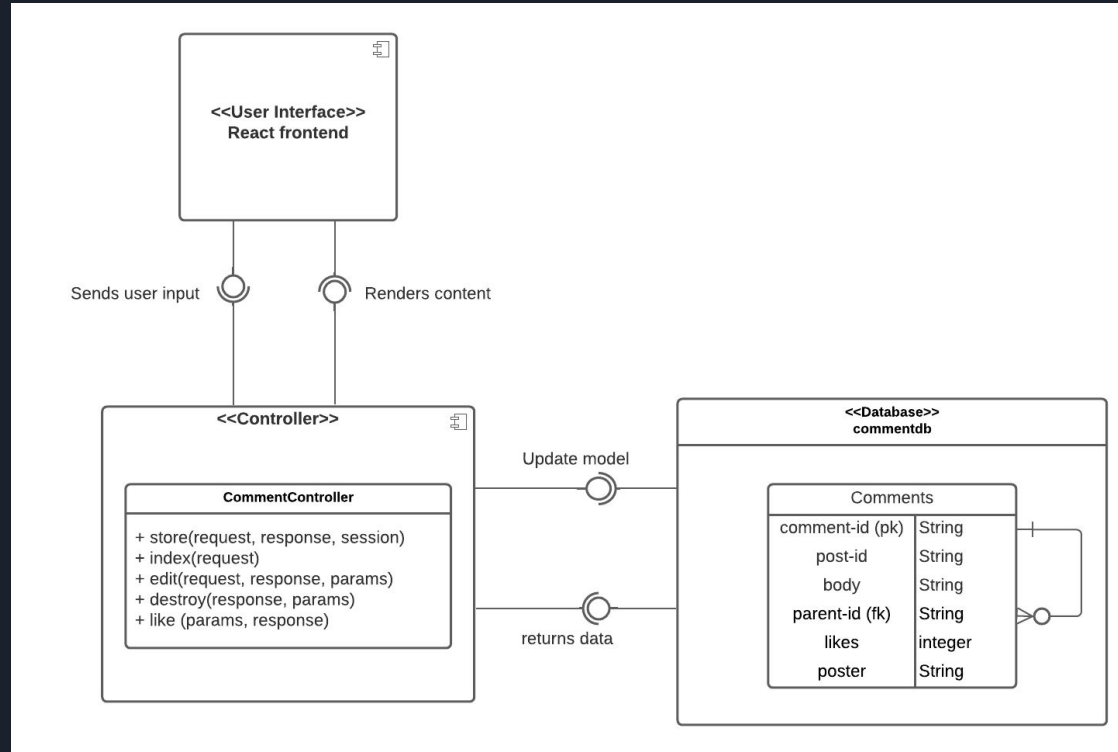
Burndown Chart

Backlog

Backlog

Sprint	Goal	Priority	Task Owner(s)	Item	Estimated Hours	Actual Hours	Completed
# 0 - Zero Feature Release	Setting up our dev environment	5	@team	Setup Github	1	0.5	☑
		5	@adam	Setup server to run at all times	1	1	☑
		5	@alex	Make user documentation	1	1	☑
		5	@ryan	Make Developer Documentation	1	1	☑
		5	@team	Set up database locally	3	1.5	☑
		5	@team	Setup routes and controller	3	3	☑
# 1 - Alpha	Basic functionality	4	@adam	Make a comment html box	2	2	☑
		3	@team	Implement database into github repo code	3	3	☑
		1	@alex @ryan	CI testing	2	2	☑
		2	@team	Set up postman	2	2	☑
		1	@team	Take a look into React	2	2	☑
		5	@team	Figure out how to store a comment in the db	4	4	☑
		2	@alex	UML Diagram	2	2	☑
# 2 - Beta	Expanded functionality	3	@alex	Pull comments by post id	2	2	☑
		5	@adam	Figure out how to pull all previous comments	2	2	☑
		2	@ryan	Set up bug tracking	2	2	☑
# 3 - ???	Polished functionality	3	@ryan	Swagger	0.33333	0.33333	☑
		2	@adam	React	2	2	☑
		5	@team	DB Finalization	2	2	☑
		3	@alex @ryan	Create JSON view	2	2	☑
# 4 - Feature Complete Release		3	@alex	Update portfolio	2	2	☑
		4	@adam	Add test users	2	1	☑
		3	@team	Reply to comment ability	6	5	☑
		5	@ryan @alex	Like Button	4	2	☑
		3	@adam	Delete button functionality	2	2	☑
		3	@team	Edit comment ability	5	5	☑
# 5 - Release Canidate	Polish project	5	@team	Address remaining bugs	3	3	☑
		4	@Adam	Clean up code	1	1	☑
		4	@ryan	Validate comment fields are filled out	2	1	☑
		3	@Alex	Add "Are you sure you want to delete this comment" - Sweet alert 2	3	3	☑
		2	@Alex	Add time and date comment was posted at	1	1	☑
		1	@Adam	Optimize for mobile	2	2	☑
# 6 - Full Release	Polish project	4	@Alex	Add set up instructions to dev docs	1		☑
		4	@Alex	Add swagger docs and update user docs	1		☑
		4	@team	Create presentation	3		☑
		2	@Adam	Polish CSS	2	2	☑

Initial UML

Final UML

Design Pattern: Model View Controller
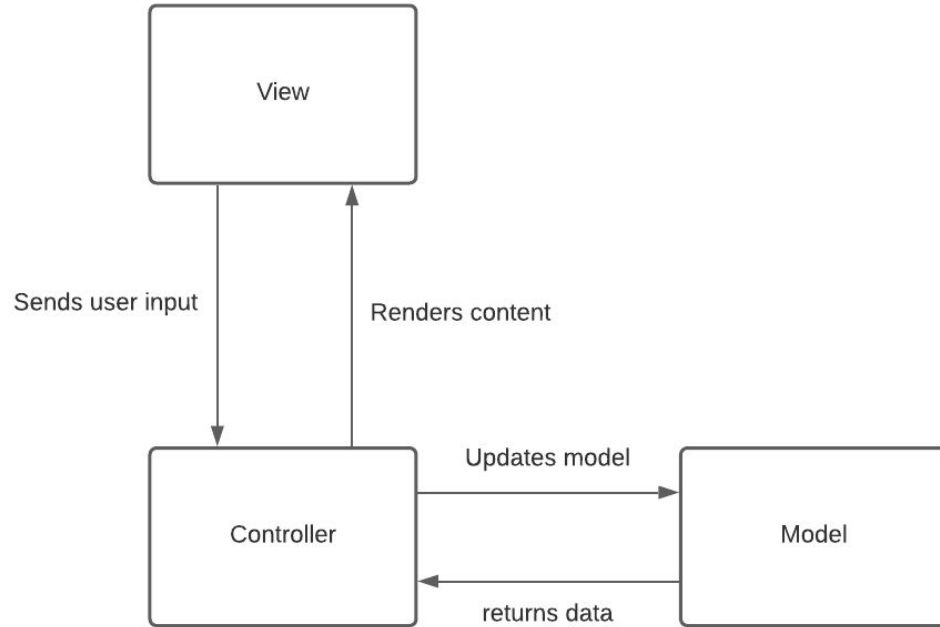
GitHub

- Utilized the Projects tab to delegate tasks and responsibilities.

- Utilized the Issues tab to track bugs and ensure they were addressed.

- Utilized the Pull Requests tab to track changes from our dev branch to main and merge them once reviewed.

March 25, 2021 – April 25, 2021   Period: 1 month ▾

Overview

29 Active Pull Requests   6 Active Issues

⌥ 29	⇅ 0	⊘ 6	⊘ 0
Merged Pull Requests	Open Pull Requests	Closed Issues	New Issues
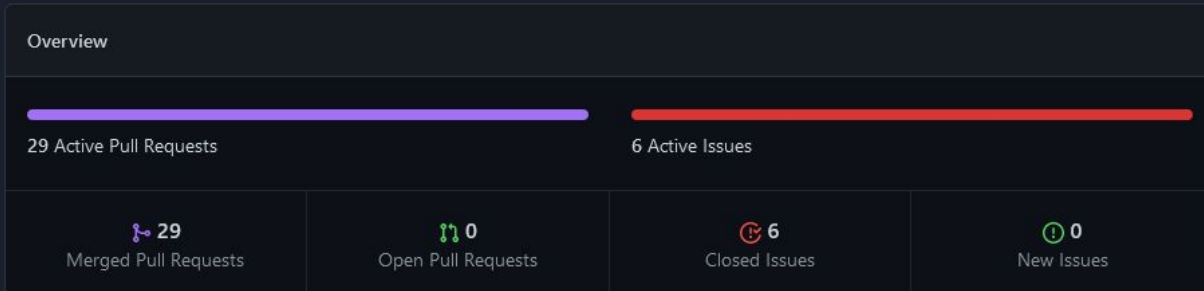
Design decisions - REACT

- Ensure there was a robust frontend to demonstrate our API

- Offers more frontend customization compared to Adonis

- Learning opportunity

React

Quality Assurance - Cypress + Cucumber

- Cypress works with DOM elements to provide frontend testing

- Set tests to run nightly to ensure features are always working

cypress

Quality Assurance - Cypress + Cucumber Cont.

- Cucumber for BDD (Behavior Driven Development)

  - Given the opportunity, would start our project writing these tests

- Demo!

```
Given( expression: 'I navigate to acommentapi',  implementation: () => {

    cy.intercept("GET", "/api/comments").as( alias: "getComments");
    cy.intercept("POST", "/api/comments").as( alias: "makeComment");

    cy.visit('https://acommentapi.com/');
    cy.wait("@getComments");
})

Then( expression: 'I want to add a comment',  implementation: () => {

    cy.get('button:contains(Add Comment)').click()
})

And( args: 'I type out a comment', () => {
```

```
Scenario: User adds a comment
    Given I navigate to acommentapi
    Then I want to add a comment
    And I type out a comment
    And I post the comment
    Then The comment posts
```

Demo!

https://acommentapi.com/