

# Sprint 2 Retrospective

Hagglers

**Dates of Sprint 1:** 01/22/2024 - 02/02/2024

**Sprint 1 Retrospective:** 02/05/2024 - 02/09/2024

## Team Roles

**Product Owner:** Stella Yang

**Scrum Master:** Jackson Stone

**Coding Monkey:** Justin Abraham, Victor Pan, Griffith Thomas, Nick Anaya, Emmie Teng

## Goals For This Sprint

Our goal for this sprint was to implement the core game mechanics allowing the user to progress through levels, which included creating goals for the user in the form of a shopping list and allowing the user to complete these goals by trading with NPCs. We also had a secondary goal of cleaning up some of the UI from sprint 1, including using dropdowns for item and character selection fields.

## Sprint Backlog

For Sprint 2, we had the following stories, chores, and bugs in our backlog.

#	Feature	Description	Points	Type
1	World 2 view	Make a second town view and display all of the characters that are there.	0	Chore
2	Add missing tests to models	Write rspec tests for all current models, testing validations and interactions	0	Chore
3	Display item value in world 2 and beyond	As a <b>player</b> So that I <b>can learn how monetary value influences trades</b> I want to <b>see the items' values in my own inventory and the non-player's inventory</b>	1	Story
4	+ Back button goes to previous page	<b>Current Behavior:</b> Clicking the back button returns to profile page for the player, which is not where they came from <b>Expected Behavior:</b> The page should go back to where the user came from prior to this page.	0	Bug

5	Select items to accept and offer from dropdown in new/edit nonplayers	As an <b>admin</b> So that I <b>can make a new nonplayer character without knowing the database id of the item</b> I want to <b>be able to select the item from a drop down</b>	1	Story
6	Select item and character from list in new/edit inventory	As an <b>admin</b> So that I <b>can make or edit a inventory entry without knowing the name of the item and character</b> I want to <b>be able to select the item and character from a drop down</b>	1	Story
7	+ Increase rubocop strictness	The rubocop.yml should not have as many exceptions for things that are not necessary	0	Chore
8	+ Default character image for new character	When a new character is created, the town page should still be available. We can fix this by having a default image for when a name does not match up	0	Chore
9	+ Default character image is not shown in production	<b>Current Behavior:</b> The characters are being rendered with an expectation that their name in lower .gif exists <b>Expected Behavior:</b> The characters without images should have a default image rendered	0	Bug
10	Shopping list	As a <b>player</b> So that I <b>have an incentive to get items</b> I want to <b>have a list of items to acquire as a game objective</b>	2	Story
11	Advance to next world button	As a <b>player</b> So that I <b>can progress to a new world</b> I want to <b>buy a ticket and see a new scene/town + have new interactions available to me</b>	2	Story
12	Fix player form	The player should not have a type field because it is determined by the class not by input. The player should not be allowed to have a negative level	0	Chore
13	Add code climate integration CI/CD	Add the code climate integration to our github CI pipeline to provide helpful information on our branches, keeping it clean as we go	0	Chore
14	+ Improve Rubycritic score	Refactor code to fix code smells that Rubycritic points out	0	Chore
15	+ Refactor item tests	The 'Item' table is missing some CRUD Cucumber tests need refactoring.	0	Chore

16	+ Refactor inventory tests	The 'Inventory' table is missing some CRUD Cucumber tests and some current tests need refactoring	0	Chore
17	+ Render world 2 sprites	Create and display sprites for world 2 characters and seed another character, Ron	0	Chore
18	Make a counter offer to an NPC	As a <b>player</b> So that <b>I can get a better deal</b> I want to <b>counter offer the npc with my own items and quantity</b>	3	Story
19	Player accept an offer from an NPC	As a <b>player</b> So that <b>I can work towards getting the items I want</b> I want to <b>be able to accept a trade offer from an npc</b>	2	Story

## Descriptions for added bugs and chores

- Back button goes to previous page
  - During dev testing work, we noticed that the back button would take the user to unexpected pages.
- Increase rubocop strictness
  - In order to stop developing code that is based on rubocop rules that are too generous, we chose to immediately address this issue and refactor as needed as opposed to having to refactor more later.
- Default character image for new characters
  - While testing our code for adding characters, we noticed that new characters would simply appear as text without an image like the rest of characters. We decided that we had enough bandwidth to bring this chore into the current sprint.
- Default character image not shown in production
  - After doing the development work for adding a default image, testing locally, and merging the chore, we realized that the image was not showing on the production deployment.
- Improve rubycritic score
  - While our PRs were getting merged, our rubycritic score was consistently decreasing. We decided to create a chore specifically to address this negative trend.
- Refactor item and inventory table tests
  - As we were implementing the cucumber tests, we noticed some issues causing the new tests to not be compatible with our design and some issues with the current tests. We refactored the tests to ensure good coverage and complexity.
- Render world 2 sprites
  - As we progressed on creating world 2, we decided to create new sprite art to distinguish the new NPCs that would be rendered on that page.

## Sprint Achievements

For this sprint both of our goals, creating game mechanics to progress and tweaking the UI to improve user experience, were completed. Additionally, we fixed many bugs and issues that were brought up during our last demo including an overly generous rubocop, and increased the resiliency of our CI pipeline by incorporating Code Climate.

### Completed Stories:

- Shopping list and level advancement
  - Gave the user a shopping list as motivation to trade
  - Implemented a button to “purchase” a ticket to the next world, only after the shopping list has been fulfilled
- Trading and bartering
  - Beyond level 1, item monetary values are visible to the user
  - Players can accept offers from NPCs
  - Players can make counter-offers to NPCs, which NPCs will accept based on the quality of the offer
- UI improvements
  - Use dropdowns instead of item IDs in the admin CRUD pages

## Team Contributions

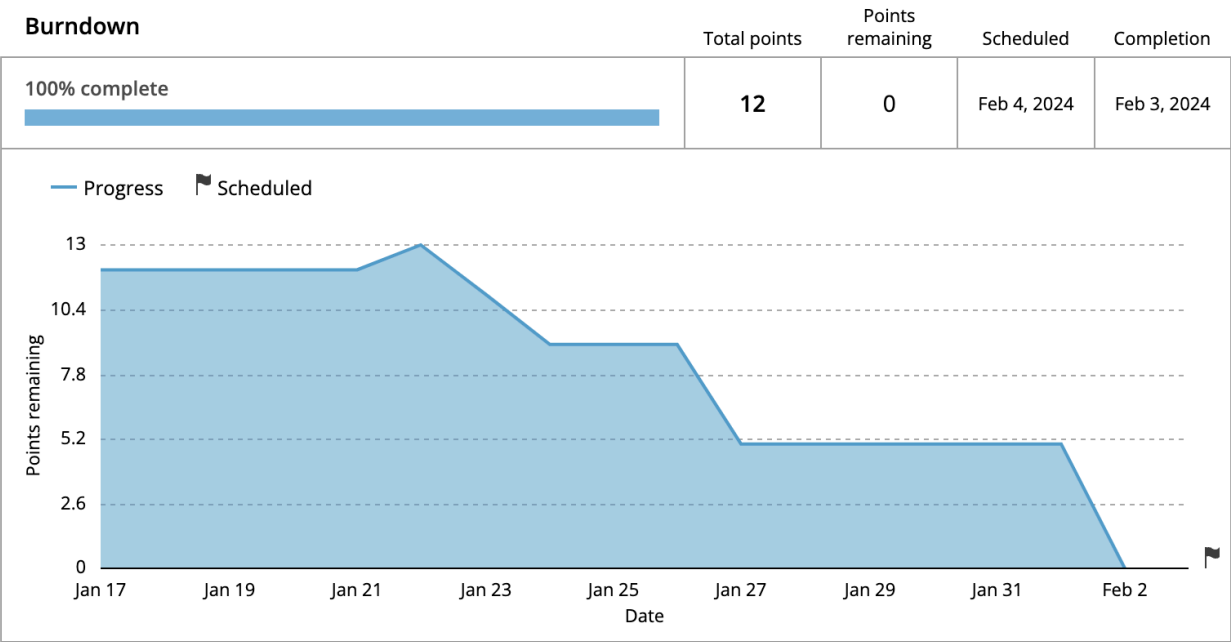
Name	Effort %	Work done
Nick	14%	<p>Wrote RSpec tests (5 hrs)</p> <p>Added counter offer methods in player model (1 hr)</p> <p>Modified NPC trade logic (0.5 hrs)</p> <p>Implemented inventory management system for counter offers (1.5 hrs)</p> <p>Integrated player and NPC models (0.5 hrs)</p> <p>Implemented/updated counter offers UI (0.5 hrs)</p> <p>Integrated frontend with backend (0.5 hrs)</p>
Emmie	15%	<p>Make World 1 only display characters from level 1 (0.5 hrs)</p> <p>Display all characters from level 2 (0.5 hrs)</p> <p>Write cucumber step definitions for item value (1hr)</p> <p>Detect player is in world 2 plus (0.5 hrs)</p> <p>Change the view to have conditional attribute display (0.5 hrs)</p> <p>Make back button go back to previous page (0.1 hrs)</p> <p>Made gif for default character (0.25 hrs)</p> <p>Advance next world rspec tests (0.5 hrs)</p> <p>Check player has enough item (1 hr)</p> <p>Change player's level in db (1.5 hrs)</p> <p>Add new routes for post request (0.25 hrs)</p> <p>Improve rubycritic score (3 hrs)</p> <p>Seed world 2 chars (0.5 hr)</p> <p>Create and render gif for world 2 chars (2hrs)</p>
Justin	14%	<p>Implement Cucumber tests (3hrs)</p> <p>Implement RSpec tests (5 hrs)</p> <p>Create trading logic (4.5 hrs)</p> <p>Integrate front end with backend (1 hr)</p> <p>Refactored structure of trade logic and reroute (2 hrs)</p>
Victor	14%	<p>Scaffold and seed ShoppingList (.5hr)</p> <p>Display shopping list on front-end (.5hr)</p> <p>Add check to see if Player has item (.5hr)</p> <p>Add indicator to front-end if Player has item (.5hr)</p> <p>Implement Cucumber tests (2.5 hrs)</p> <p>Write RSpec tests (1.5hrs)</p> <p>Add and fix missing Cucumber tests for Items (.2hr)</p> <p>Add and fix missing Cucumber tests for Inventory (.5hr)</p>
Griffith	15%	<p>Added missing tests to models (1 hr)</p> <p>Increased rubocop strictness (0.17 hrs)</p> <p>Initial work on figuring out how to use dropdowns dynamically from</p>

		the database (2 hrs) Worked on passing cucumber tests for character dropdowns (0.5 hrs) Added rspec tests for character dropdowns (0.5 hrs) Add character dropdowns on the edit page and fixed merge conflicts (0.5 hrs) Changed new and edit forms for nonplayer (0.5 hrs) Fixed the player form (removed type field and added validation) (0.5 hrs) Added code climate integration (1 hr)
Jackson	13%	Sprint Planning (3hrs) Daily Scrum Meetings (2 hrs) Sprint Retrospective document (2 hrs)
Stella	15%	Wrote cucumber tests for all of the stories (6 hr) Researched into bug causing assets not to be found on prod (1 hr) Peer programmed on launch button (2 hr) Peer programmed on shopping list (1.5 hr) Peer programmed on offer (.5 hr) Peer programmed on counter offer (3 hr) PR reviews and suggestions (1.5 hr)

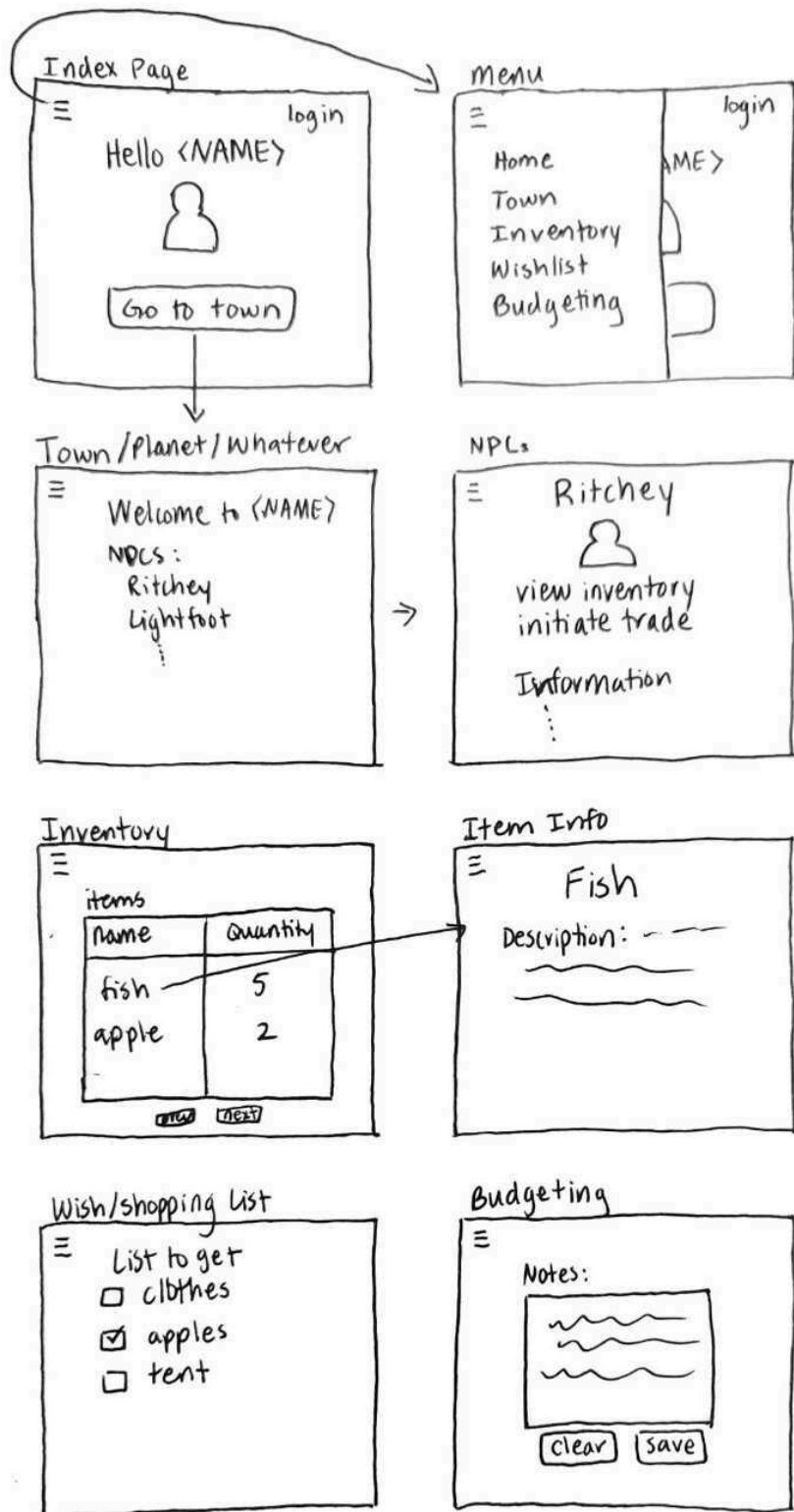
## Burn Down Chart

We expected to finish 12 story points, increasing our velocity from 8 due to the increase in development time in this sprint compared to sprint 1. We were able to fully complete all 12 story points. The brief peak at 13 is due to a chore being mistakenly added as a 1 point story before this was corrected.

Burndown

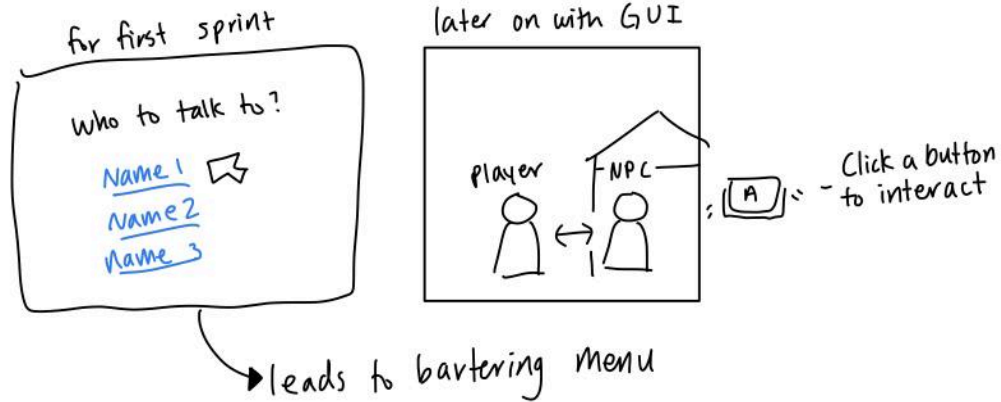


## Design Diagrams

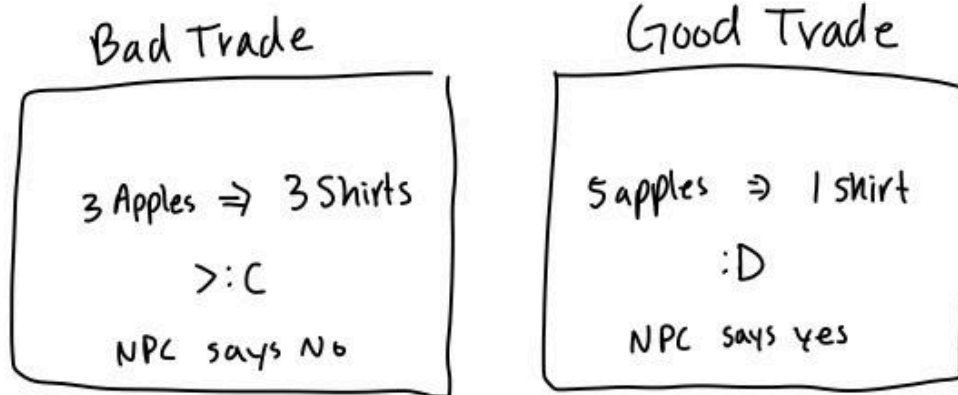




## Interaction System



## Item Value Checker



## Bartering

The game will be a 2d side scroller with the player moving along to different stalls to barter

Score: ~

Barter:

item  give

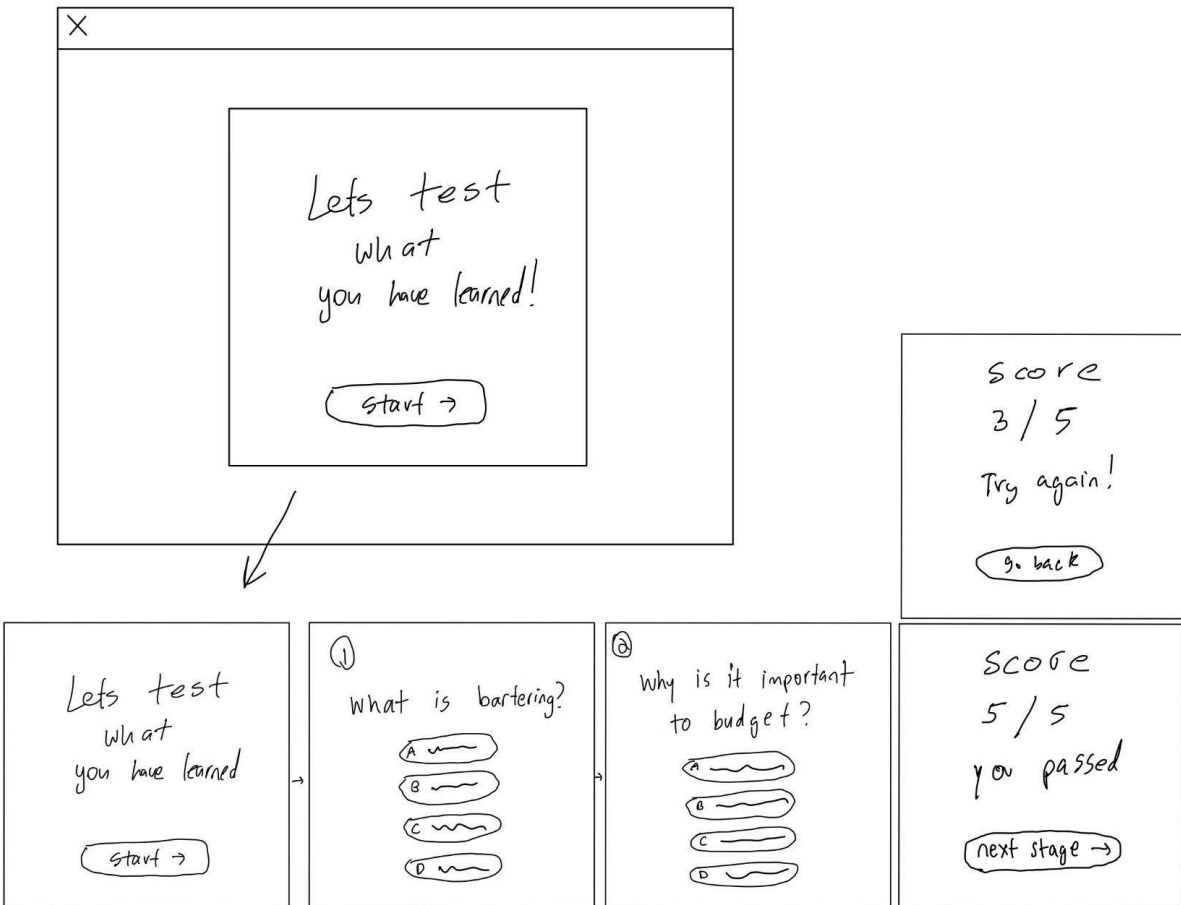
#

↓

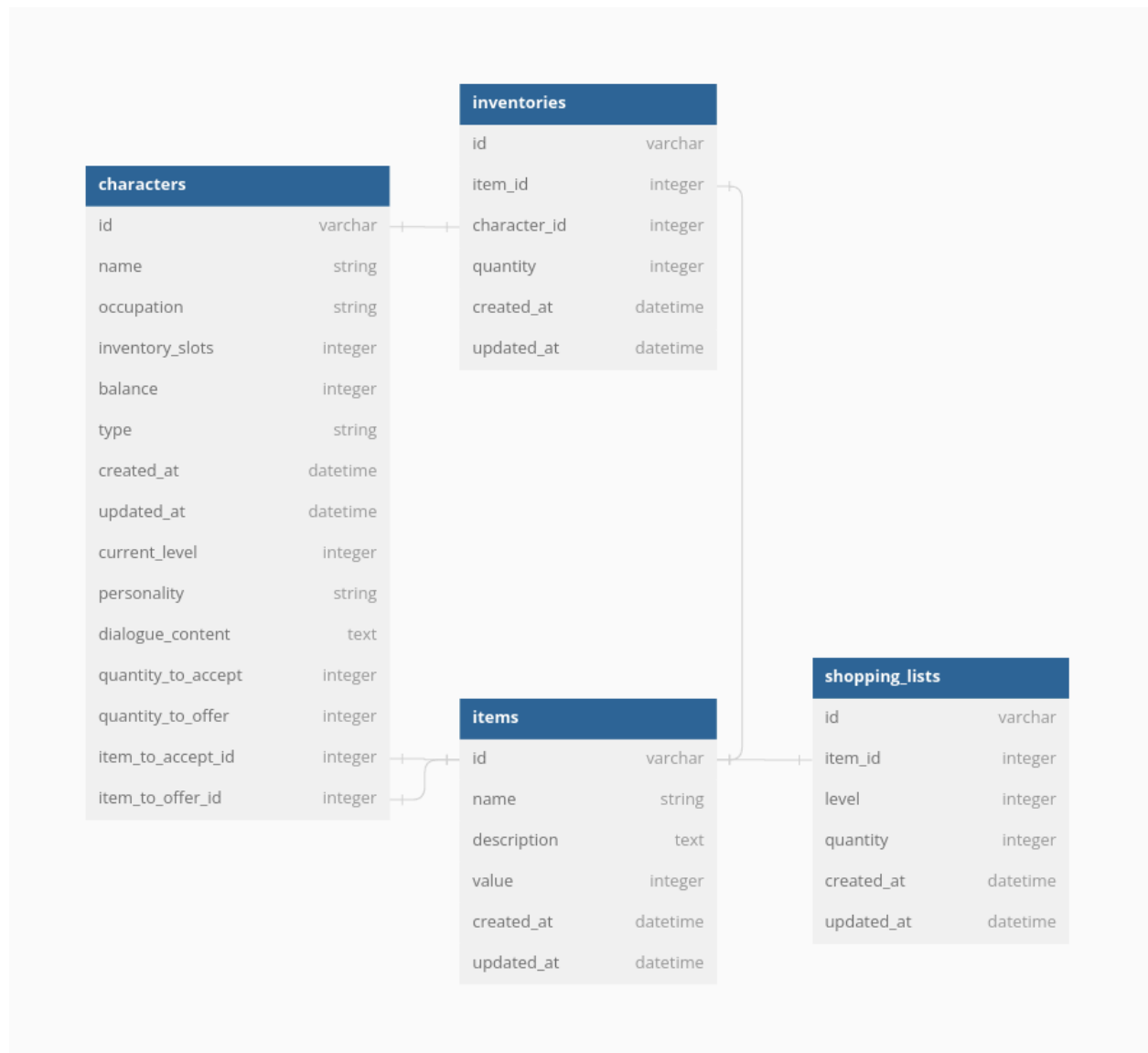
item  get

#

## End of Phase Quiz



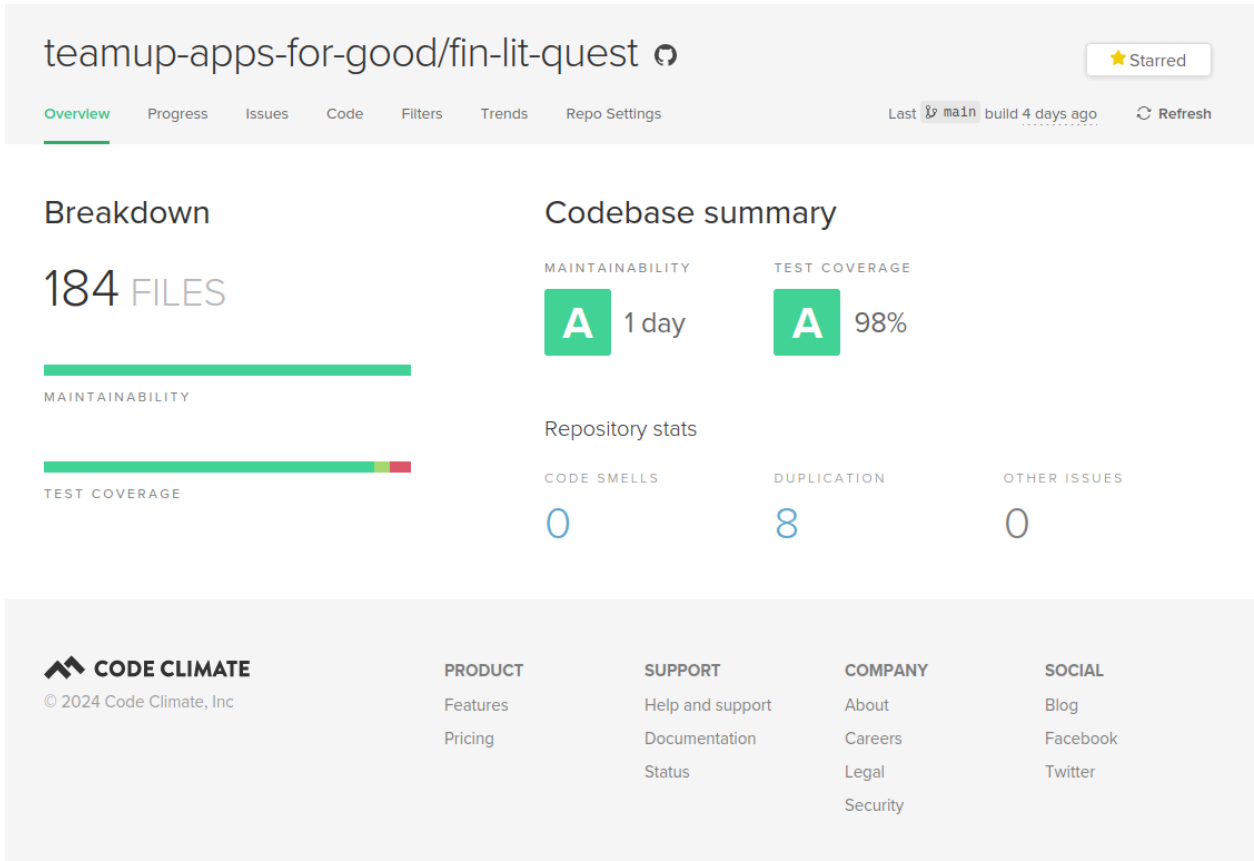
## UML Diagram for tables



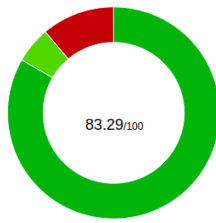
## Documentation of Changes

Throughout this sprint, the only element of our design that changed is our database design. We added a table to track shopping lists, storing an item, its quantity, and the level to which the shopping list applies. The shopping list has a many-to-one relationship with items, where each item may be on many shopping lists but each shopping list contains one item. The shopping lists for each level are compiled to produce the full shopping list for each level, and this table will be only modified by admins and should only be tweaked for gameplay improvements so will be relatively static. We continued our plan of making the game text-based first with images for sprites.

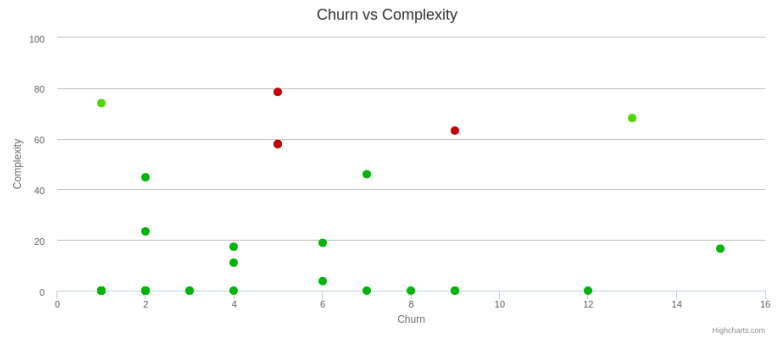
# Evaluation of code and Test Quality



# Overview



A B C D F highcharts.com



## Summary

A

30 121 33  
files churns smells

B

2 14 24  
files churns smells

C

0 0 0  
files churns smells

D

0 0 0  
files churns smells

F

4 24 52  
files churns smells

Filter: <input type="text" value="Filter by Name"/>						
Rating	Name	Churn	Complexity	Duplication	Smells	
F	InventoriesController	9	63.23	120	14	
F	ItemsController	5	57.92	120	11	
F	PlayersController	5	57.92	120	12	
F	ShoppingListsController	5	78.52	120	15	
B	NonplayersController	13	68.19	48	11	
B	TradeController	1	74.08	0	13	
A	Channel	1	0.0	0	1	
A	Connection	1	0.0	0	1	
A	ApplicationController	1	0.0	0	1	
A	CharactersController	15	16.6	0	3	

# Customer Meeting

**Date:** 02/01/2024

**Time:** 12:00 pm Central Time

**Location:** Google Meet

## Summary of Meeting:

During this meeting, we discussed the state of our sprint, demonstrating the stories which were complete and explaining the final two stories which would be completed later that day and the next day. The client was very happy with our progress, but also pointed out some core gameplay features which we should consider incorporating into the next sprint. This primarily consisted of adding more mechanics beyond the shopping list into the game such as expenses to motivate the player to continue to try and increase their wealth. We discussed how we would implement this, including using NPCs as teaching mechanics to the player, allowing the player to learn from them as examples and having direct dialogue teaching to the player. Furthermore, we discussed how OAuth would be a priority for us going forward, as during this sprint we started thinking about how we would incorporate authentication to store progress and decided it would be better to add this feature as soon as possible. We also discussed our direction of primarily creating a game engine first and then adding the content second, which the client agreed is a direction that is good and we should continue, citing his experience working with large game studios and his knowledge that this is how they generally produce new games. Furthermore, this allows the game to be much more extensible by other game designers and maintainers later on with the core functionality already in place. Finally, we discussed what we learned from this sprint, the main two points being: we should strive to maintain consistent progress instead of completing work in the first or latter half of the sprint while prioritizing blocking tasks, and us having finished most of the work of the sprint in the first half shows that we can increase our velocity going forward in an effort to maintain more consistent progress.

## Project Links

**Github:** <https://github.com/teamup-apps-for-good/fin-lit-quest>

**Project Tracker:** <https://www.pivotaltracker.com/n/projects/2687724>

**Slack:** <https://hagglerdev.slack.com>

**UML-Diagrams:** <https://dbdocs.io/stella/FinLit-Quest>