Ryan Pate
Nathan Raymon
5/2/2020

Project Step 3 Final Version

URL:http://flip2.engr.oregonstate.edu:6187/

Project Outline and Database Outline - Updated Version:

Overview

We are going to make a simple town simulator for use in a tabletop roleplaying game. Players of tabletop games often have to design their own towns and handle which NPCs have access to different goods and quests themselves. Our system will make this easy by simulating the town and the goods and services offered in it so that players can interact directly with the town. Our system will be designed to handle only a few interactions occasionally since it is designed to be used as a gameplay tool. Our system will help players navigate through the town by selecting shop locations to visit. At each shop the players will be able to interact with NPCs who will give them quests and shopkeepers which will allow them to purchase goods. Our system will NOT process orders or modify player inventory. It is expected that our tool will be used by a game master as a part of a larger game such as Dungeons & Dragons or Pathfinder. So, we expect the players and game master to handle adding items to the player inventories and adjusting their money after each purchase. Our system will only display the goods a player decides to purchase as well as the total cost of the sale after the purchase is complete. Orders will not be stored in the database or accessible at all once the player has left the sale screen.

Database Outline

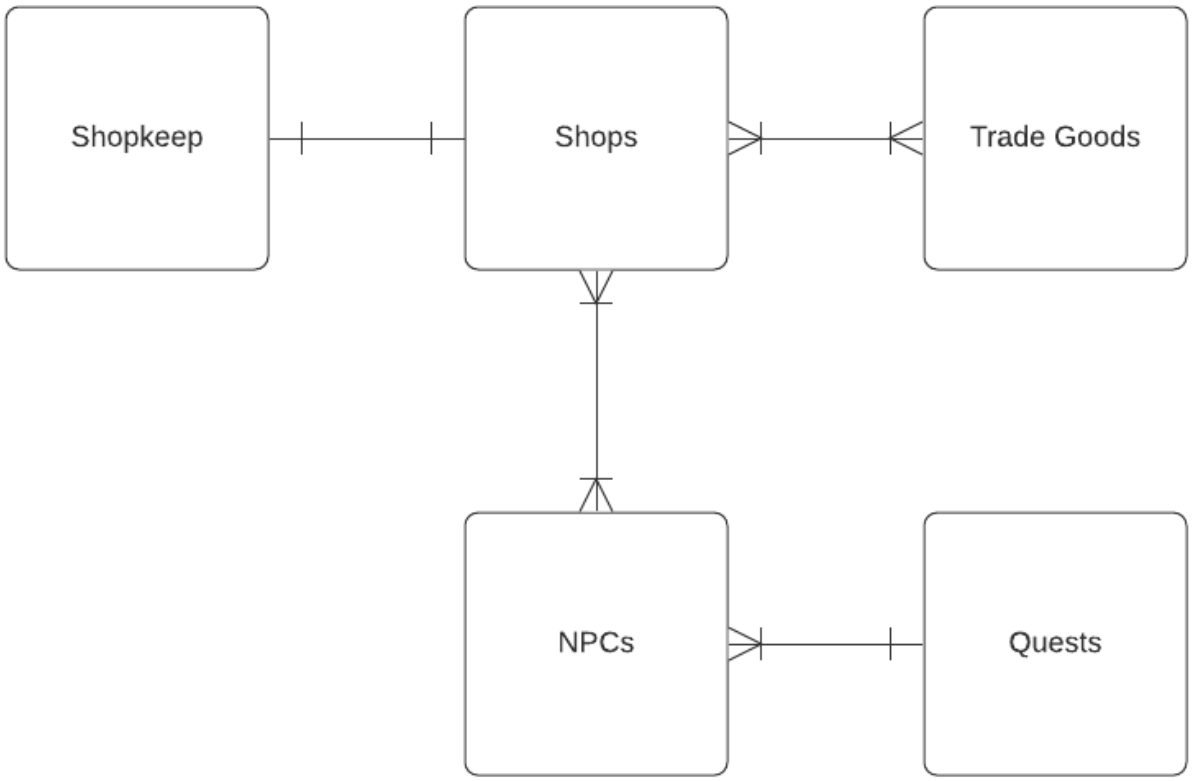
- Shops - Ryan
  Stores the shops that can be visited in the town as well as a name of the shop, a brief description, and a 1:1 relationship with the NPC that runs the shop called the "shopkeep".
  - id: int, auto_increment, unique, not null, primary key
  - shopkeep: varchar, not null
  - shopkeepId: int, not null
  - shopName: varchar, not null
  - shopDescription: varchar, not null
  - Relationships: M:M relationship with NPCs implemented in locations table, M:M relationship with trade goods implemented in Inventories table, 1:1 relationship with shopkeep NPC implemented as shop attribute shopkeepId.
- TradeGoods - Ryan
  Stores all the different trade goods that can be bought at the different shops in town. Each good has a name, value, quantity available, description, and weight.
  - id: int, auto_increment, unique, not null, primary key

    - name: varchar, not null
    - value: int, not null
    - quantity: int, not null
    - description: varchar, not null
    - weight: int, not null
    - Relationships: M:M relationship with Shops implemented in the Inventories table
- Locations: - Ryan
  Stores the relationships between non shopkeep NPCs and shops. Each NPC can be found in a variety of shops so this table stores those M:M relationships. This table will not store the shopkeeps relationship with a shop since that is a 1:1 relationship and will be stored in the Shop entity for convenience.
    - id: int, auto_increment, unique, not null, primary key
    - npcId: int, not null, foreign key specifying the id of the NPC in the NPCs table
    - shopId: int, not null, foreign key specifying the id of the shop in the Shops table
- Inventories - Ryan
  Stores the relationships between trade goods and shops. Each trade good can be bought in a variety of shops around town so this table stores those M:M relationships.
    - id: int, auto_increment, unique, not null, primary key
    - goodId: int, not null, foreign key specifying the id of the good in the TradeGoods table
    - shopId: int, not null, foreign key specifying the id of the shop in the Shops table
- NPCs - Nathan
  This table stores the name and greeting of the NPC when communicating with the user.
    - Many to many relationship with Shops
    - id: int, auto_increment, unique, not NULL
    - name: varchar, not NULL
    - greeting: varchar, not NULL
    - Relationships: a M:M relationship between NPCs and Shops implemented in the Locations table. 1:1 relationship with the shopkeep NPC implemented with shopkeep npcId as a FK inside of Shops under the shopkeepId attribute.
- Quests - Nathan
  This table will hold the name, what the quest will entail (description) and what they reward will be if the user completes the quest.
    - id: int, auto_increment, unique, not NULL
    - name: varchar, not NULL
    - description: varchar, not NULL
    - reward: varchar not, not NULL
    - Relationships: A 1:M relationship between Quests and NPCs is implemented with questId and npcId as a FK inside of npcQuests.
- NpcQuests - Nathan

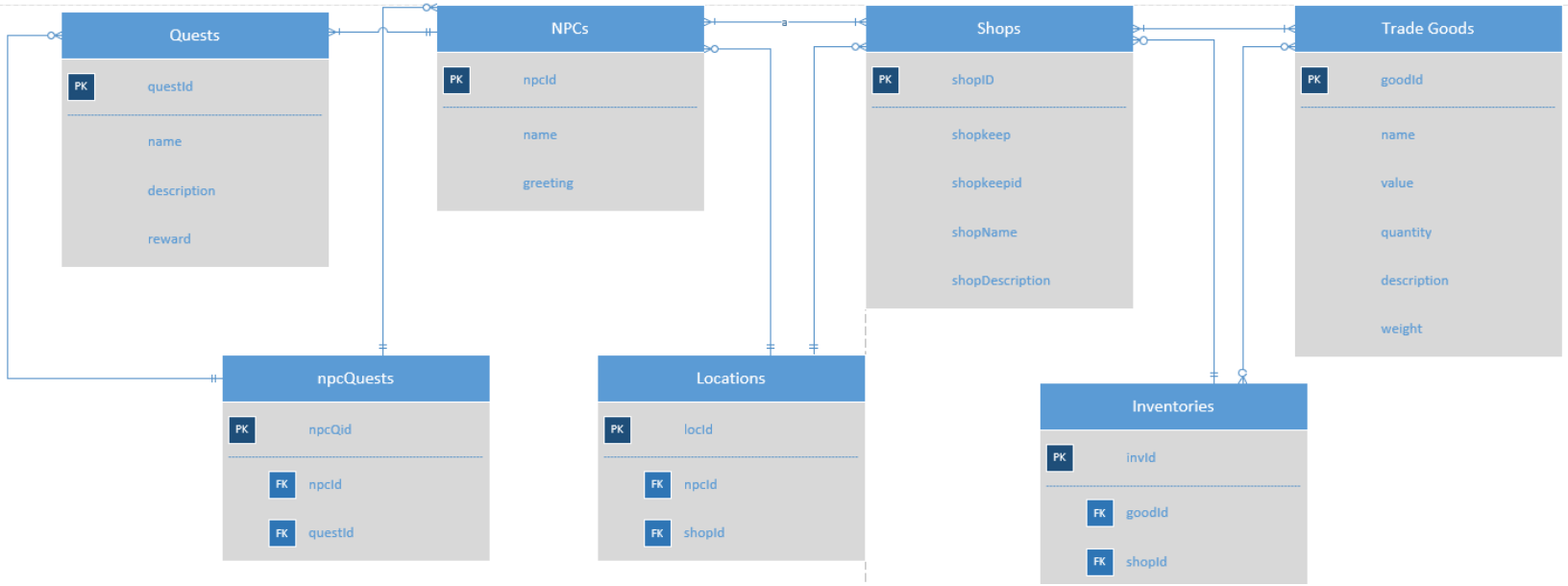
This table holds the ids of NPCs and Quests, correlating to which NPCs are holding which Quests.

- id: int, auto_increment, unique, not NULL
- npcId: int, not null, foreign key specifying the id of the NPC in the NPCs table.
- questId: int, not null, foreign key specifying the id of the quest in the Quest table.

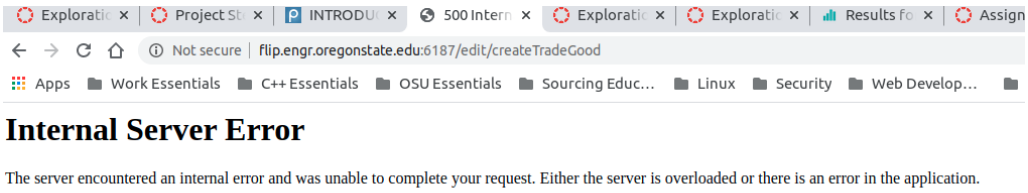
Entity-Relationship Diagram:



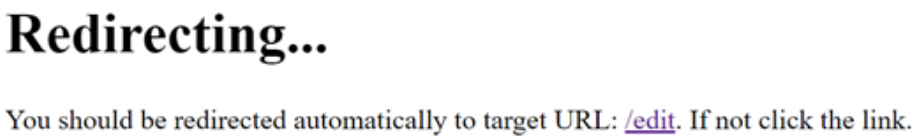
Schema



Michael Kochka - “Yes, everything that is listed appears to be working. However, when I tried adding a second item I got this error message :”



Jennifer Briere - “Shop and Edit pages are both showing data, which I presume is coming from the db and not static. I was able to add a trade good but after I submitted I got this page and it never redirected, I had to click the link:



- UI-wise I would suggest having some sort of navigation where the user can jump to any of the pages from whatever page they’re on, rather than just having a Back button.
- *Image on the home page is broken* – I may be wrong about this. My team isn’t doing our project with python/flask so I don’t know anything about it, but I think you should just be able to use <img src="/Images/runswick.jpg"> for the image instead of what you currently have”

Fixes based on feedback

When testing the feedback given to us by Michael, this error didn’t seem to happen. We were able to continuously add items without any problems. We are not too sure what might have caused this to happen. The feedback from Jennifer showed us that it was a problem and have since fixed the redirect to reload the page completely. We are also aware that the image is not working on the home page, but we were not too worried about fixing it for this week. We hope to have the webpage have a better interface when the entire project is done. However, when it comes to the navbar, we are still undecided on whether we do want one. We want it to feel like the players (users) are navigating through the town visiting places instead of just hopping around from page-to-page. It is still undecided we’ll see where we land with it.