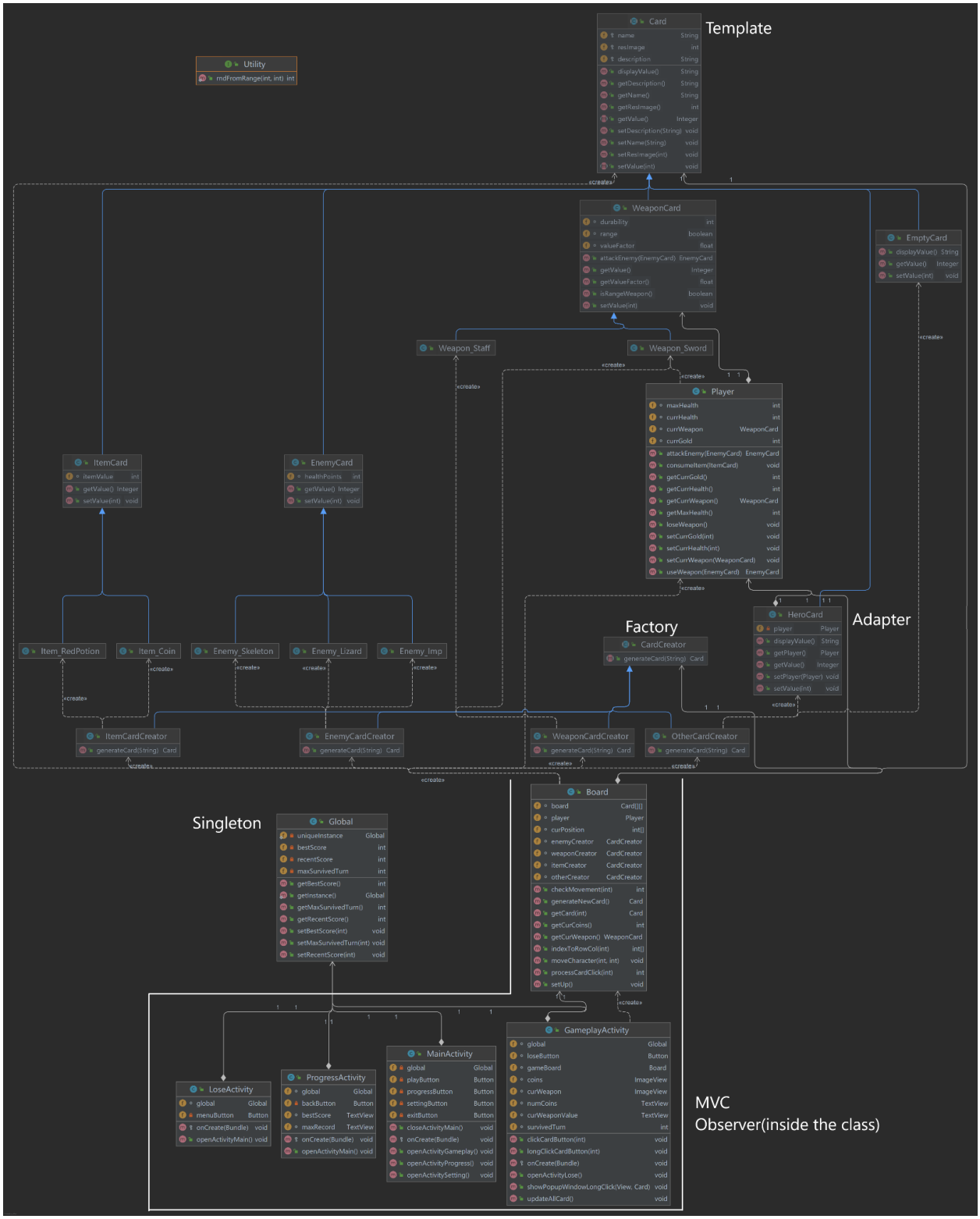
# Final Project Report

1. **Project Name: Dungeon Crawler**
    a. Team Members: Tiancheng Shao, Elijah Aldinger
    b. Note: Daniel Helfrich has not committed any code on the repo yet, but he plans to upload them later

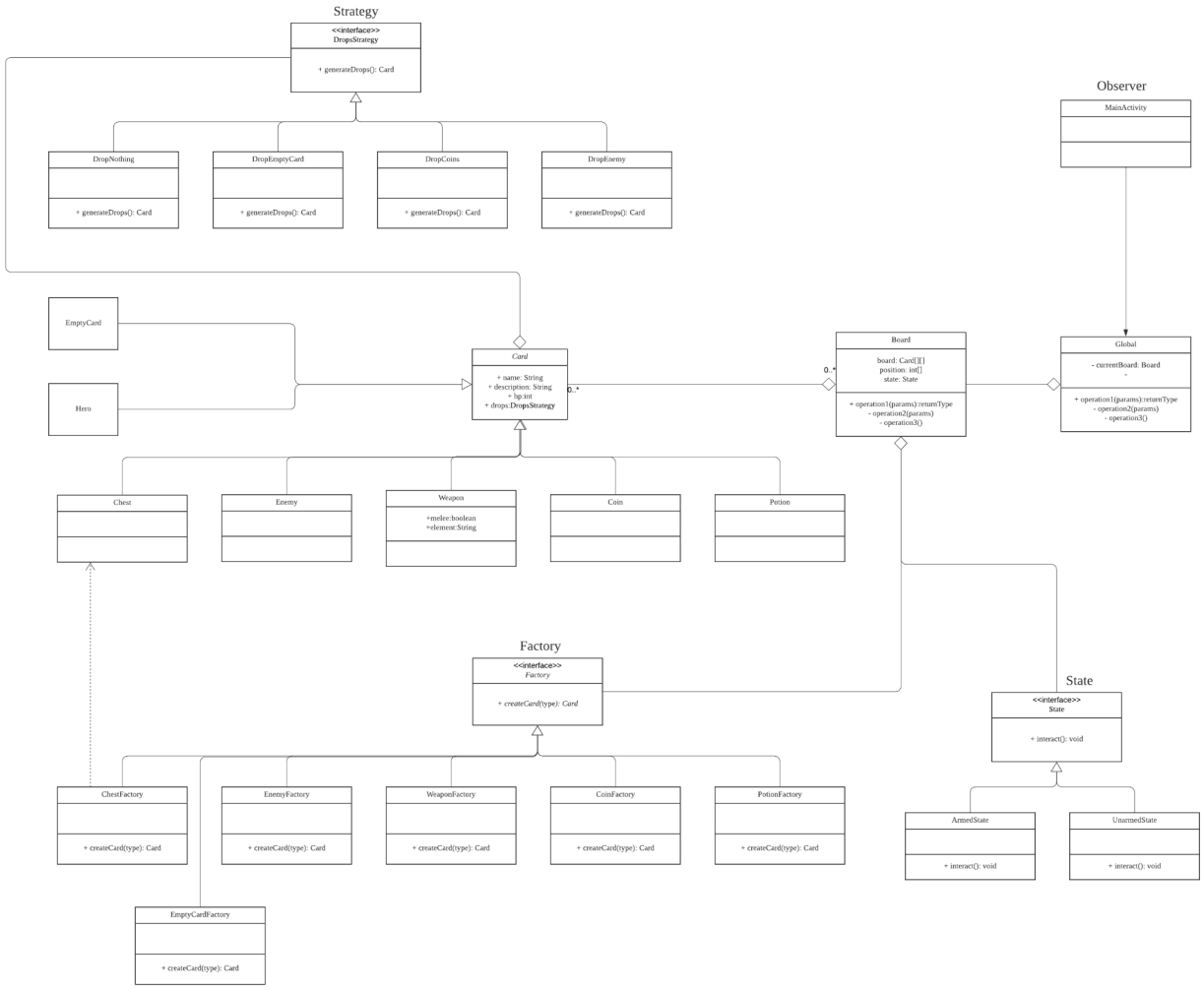
2. **Final State of System**
    a. The project is an Android-based app developed using IntelliJ/Android Studio. It will feature arcade gameplay based on getting as high of a score as possible before running out of health. The player moves around a 3x3 grid of cards by picking up one that is in an adjacent slot, causing the other side of the grid to spawn a new card, repeatedly. Different cards perform different actions when picked up (Enemies, Weapons, Items) requiring the player to be strategic about which cards to pick up when in order to survive.
    b. From project 6 we have implemented all game logic/interaction, the progress screen, and improved the quality of images. We implemented the long click functionality we introduced in project 5. We did not implement the data storage, difficulty variation, and some types of cards due to insufficient time.

3. **Final Class Diagram**
    a. Final Diagram:




Google Drive:
https://drive.google.com/file/d/1LWMM6Z6nFTroLvYc7MCX31eCOxSId4Dj/view?usp=sharing

b. Project 5 Diagram:




Google Drive:
https://drive.google.com/file/d/1jBYMc7_lkHw5-jPneec1Dp3MXc2hovoi/view?usp=sharing

c. Comparison Statement:
   i. We've made some changes since project 5 and 6. First, we didn't apply the state and strategy pattern in our program because we found those classes and patterns are unnecessary. We have more activity classes than we thought, and we have implemented some other design patterns during the development such as adapter and template.

4. Third-Party code
   a. We used the Android Java API Framework to build our project. We used some code from https://stackoverflow.com/, and they are commented in the code.

5. OOAD process
    a. Card Classes
        i. The card system in the game idea is what seemed interesting to implement with the OOAD process. Different types of cards with overlapping traits were implemented with inheritance, we used adapter/template to interpret the player character with stats to a card that can interact with other cards.
    b. Gameplay Activity Flow
        i. Our class GameplayActivity runs the flow of the game while playing, serving as a class that updates the GameplayView which contains the cards and images XML. It contains a reference to the Board class to retrieve the card information from. Our design process with this class was to encapsulate the Android/XML/graphics away from our Java classes.
    c. Board Class
        i. The Board class controls the collection of cards (the Card classes, GameplayActivity controls showing the cards with that info). It uses public methods to receive player movement and update the collection of cards accordingly. Our design process was to try and use patterns like Factory and Template to simplify the logic.