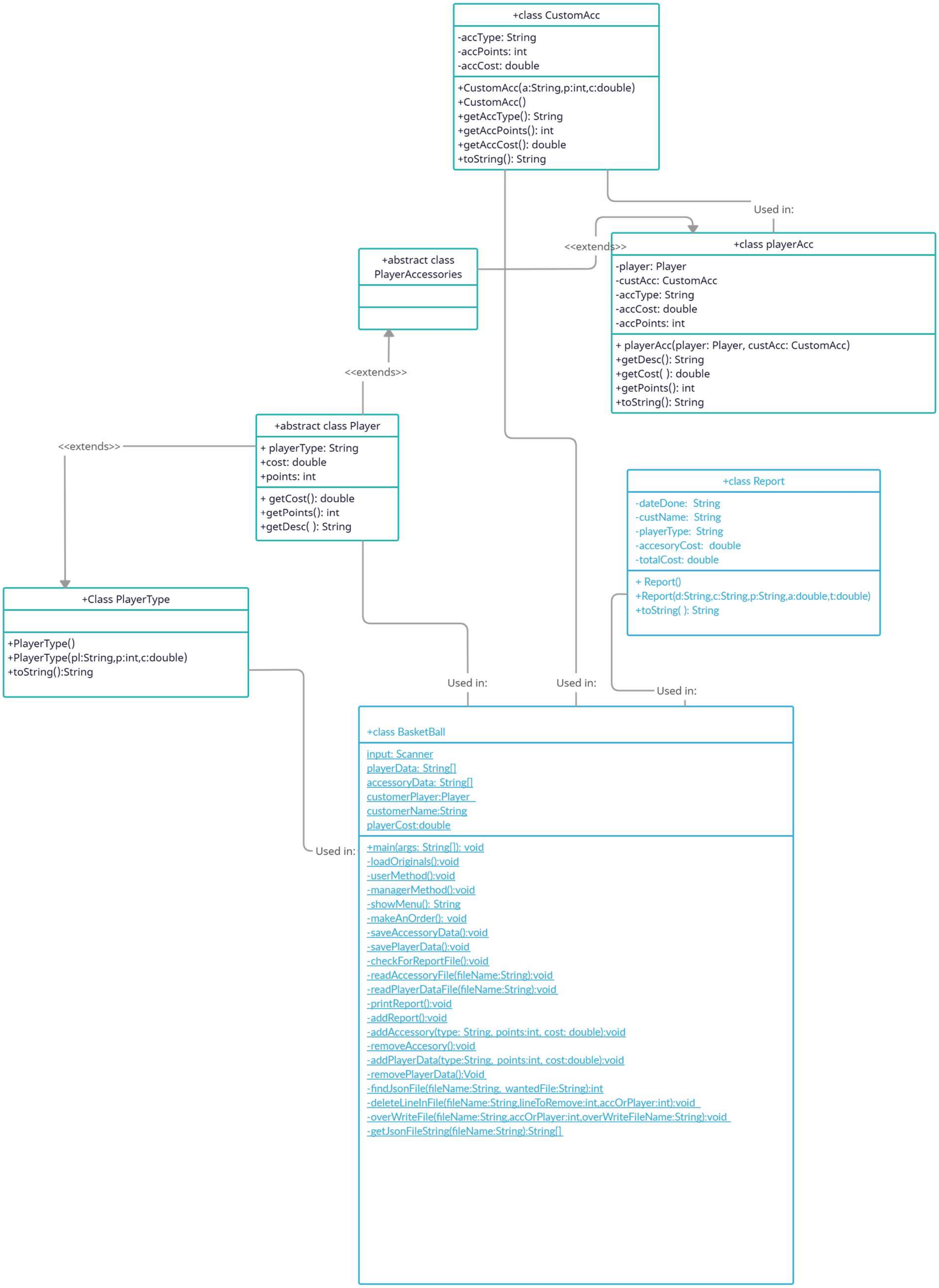
Class Diagram:



Pattern Utilized: Decorator Pattern

Design- file storage:

- Utilizes Json.
- Makes use of external library gson.
- Reports are stored in separate json file.
- Player data and accessory data are stored in separate json files.
- Saved data for player and accessory data are stored in separate files.
- ^ Only one "save slot"

Reports:

- Feature: date, customer name, player type(including accessories), accessory(s) cost, total cost.
- Invoked with addReport() private function.

Objects Map:

- Mapped through integer 0-n, n = size of data string array(acquired through json file)

User Interface:

- Users and Managers separate
- Users give name(customerName) once through program
- Users have 2 options, (1) view menu, (1)make an order
- Managers have 9 options, (1) view reports, (2) read in accessory data file(.json), (3) read in player data file(.json) ,(4) add accessory data ,(5) remove accessory data, (6) add player data, (7) remove player data, (8) save player data, (9) remove player data

Language chosen: Java

- Since all member of the group had more experience with Java than C#, also allowed us to use the external java library gson.

Design Constraints/Limitations

- Only one of each (player data and accessory data) save possible
- Possible to completely overwrite data
- Can read in accessory data to player data (resulting in null entries)

Future Improvements:

- Multiple save files
- Better UI
- Better file management