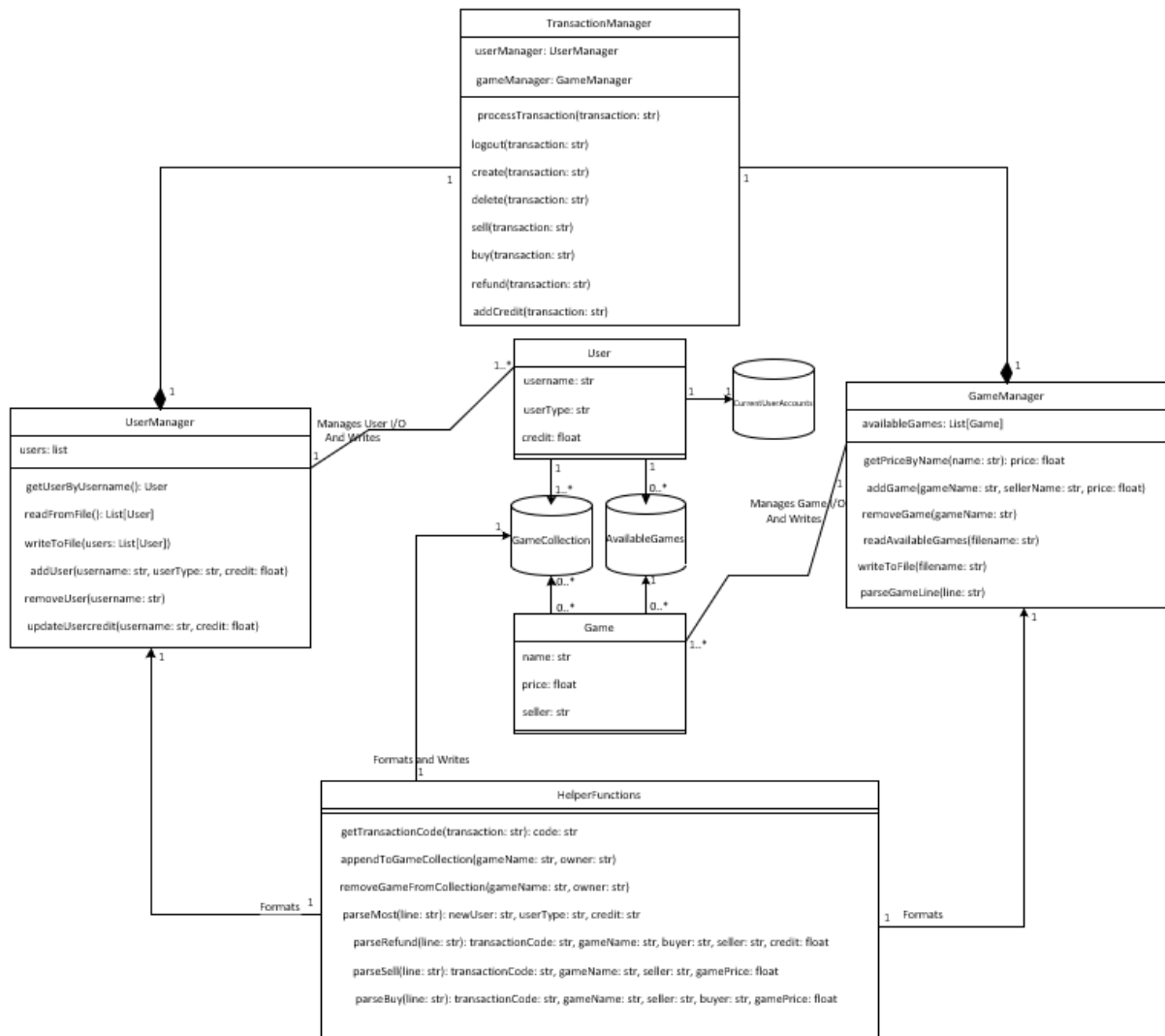


# Backend Design Document

This document illustrates the structure of the system's backend—its classes, methods, functions and their purpose. Refactoring of HelperFunctions into a new GameCollection class will be done during phase 5.

## Class Diagram



## Classes

### User

**Description:** This class is designed to represent the users in the system

**Located in:** UserManager.py

Methods	Description
User()	<b>Parameters:</b> none <b>Return type:</b> N/A  A default constructor to create a new User object with default attributes
User(string, string, string)	<b>Parameters:</b> string username, string userType, string credit <b>Return type:</b> N/A  A constructor creating a new User object with specified values

### UserFileManager

**Description:** The UserFileManager class is responsible for managing user-related operations and for file management for the CurrentUserAccounts.txt file

**Located in:** UserManager.py

Methods	Description
UserFileManager()	<b>Parameters:</b> none <b>Return type:</b> N/A  A default constructor to create a new UserFileManager object with the default attribute, an empty list of User objects
getUserByUsername(string)	<b>Parameters:</b> string username <b>Return type:</b> User  An accessor method that returns the User with the specified username

readFromFile(string)	<b>Parameters:</b> string filename <b>Return type:</b> List[User]  Reads user data from CurrentUserAccounts.txt and populates the users list with User objects created from the data
writeToFile(string)	<b>Parameters:</b> string filename <b>Return type:</b> void  Writes user data from the users list to the specified file. Each user's data is formatted and written as a separate line in the file
addUser(string, string, string)	<b>Parameters:</b> string username, string userType, string credit <b>Return type:</b> void  Creates new User object with specified User data and adds it to the user list
removeUser(string)	<b>Parameters:</b> string username <b>Return type:</b> void  Removes the User with the specified username from the user list
updateUsercredit(string, string)	<b>Parameters:</b> string username, string creditChange <b>Return type:</b> void  Updates the credit of the User with the specified username in the user list by adding the creditChange amount

## Game

**Description:** This class is designed to represent the games in the system

**Located in:** GameManager.py

Methods	Description
Game()	<b>Parameters:</b> none <b>Return type:</b> N/A  A default constructor to create a new Game object with default attributes
Game(string, string, string)	<b>Parameters:</b> string name, string seller, float price <b>Return type:</b> N/A

	A constructor creating a new Game object with the specified values
--	--

## AvailableGamesFileManager

**Description:** The AvailableGamesFileManager class is responsible for managing game-related operations and for file management for the AvailableGames.txt file

**Located in:** GameManager.py

Methods	Description
AvailableGamesFileManager()	<b>Parameters:</b> none <b>Return type:</b> N/A A default constructor to create a new AvailableGamesFileManager object with the default attribute, an empty list of Game objects
getPriceByName(string)	<b>Parameters:</b> string gameName <b>Return type:</b> float An accessor method that returns the price with the specified game name
getUsersGames(string)	<b>Parameters:</b> string username <b>Return type:</b> List[string] Returns a list of all the games being sold by the given user
readAvailableGames(string)	<b>Parameters:</b> string filename <b>Return type:</b> List[Game] Reads user data from CurrentUserAccounts.txt and populates the availableGames list with Game objects created from the data
writeToFile(string)	<b>Parameters:</b> string filename <b>Return type:</b> void Writes game data from the availableGames list to the specified file. Each game's data is formatted and written as a separate line in the file
addGame(string, string, float)	<b>Parameters:</b> string gameName, string sellerName, float price <b>Return type:</b> void

	Creates new Game object with specified Game data and adds it to the availableGames list
removeGame (string)	<b>Parameters:</b> string gameName <b>Return type:</b> N/A  Updates the availableGames attribute to not include the given game
parseGameLine(string)	<b>Parameters:</b> string line <b>Return type:</b> string, string, float  Parses a line from the AvailableGames.txt, extracting information such as the game name, owner, and price

## TransactionManager

**Description:** This class is designed to represent the transactions in the system

**Located in:** TransactionFunctions.py

Methods	Description
TransactionManager()	<b>Parameters:</b> none <b>Return type:</b> N/A  A default constructor to create a new TransactionManager object with default attributes
TransactionManager(UserFileManager userManager, AvailableGamesFileManager gameManager)	<b>Parameters:</b> UserFileManager userManager, AvailableGamesFileManager gameManager <b>Return type:</b> N/A  A constructor creating a new TransactionManager object with the specified values
processTransaction(string)	<b>Parameters:</b> string transaction <b>Return type:</b> void  Gets the transaction code indicating which type of transaction the line represents and calls the appropriate transaction function
logout(string)	<b>Parameters:</b> string transaction <b>Return type:</b> void  Does nothing because the logout transaction does not edit any of the

	files
create(string)	<b>Parameters:</b> string transaction <b>Return type:</b> void  Creates a new User object, appends it to the userManager List, and writes it to the UserAccounts file
delete(string)	<b>Parameters:</b> string transaction <b>Return type:</b> void  Deletes all references of a User from the userAccounts and availableGames files
sell(string)	<b>Parameters:</b> string transaction <b>Return type:</b> void  Creates a new Game object and writes it to the availableGames file
buy(string)	<b>Parameters:</b> string transaction <b>Return type:</b> void  Adds a game to a buying User's collection in the gameCollection file, Credits a seller, and deducts that amount from the buyer
refund(string)	<b>Parameters:</b> string transaction <b>Return type:</b> void  Removes an owned game from a User's collection in the gameCollection file, credits that buyer, and deducts that amount from the seller
addCredit(string)	<b>Parameters:</b> string transaction <b>Return type:</b> void  Adds the given credit to the user's balance and updates the CurrentUserAccounts file to reflect this change

## HelperFunctions.py

**Description:** A collection of functions for processing strings and file writing

**Located in:** HelperFunctions.py

Methods	Description
getTransactionCode(string)	<p><b>Parameters:</b> string transaction  <b>Return type:</b> string</p> <p>Returns the first 2 characters (the transaction code) of the line from the DailyTransaction.txt</p>
appendToGameCollection(string, string)	<p><b>Parameters:</b> string gameName, string owner  <b>Return type:</b> void</p> <p>Adds a new entry to the GameCollection.txt file</p>
removeGameFromCollection(string, string)	<p><b>Parameters:</b> string gameName, string owner  <b>Return type:</b> void</p> <p>Removes an entry from the GamesCollection.txt file</p>
parseMost(string)	<p><b>Parameters:</b> string line  <b>Return type:</b> string newUser , string userType, string credit</p> <p>Used for parsing values from a create, delete, add credit, or logout transaction string</p>
parseRefund(string)	<p><b>Parameters:</b> string line  <b>Return type:</b> string transactionCode, string gameName, string buyer, string seller, string credit</p> <p>Used for parsing values from a refund transaction string</p>
parseSell(string)	<p><b>Parameters:</b> string line  <b>Return type:</b> string transactionCode, string gameName, string seller, string gamePrice</p> <p>Used for parsing values from a sell transaction string</p>
parseBuy(string)	<p><b>Parameters:</b> string line  <b>Return type:</b> string transactionCode, string gameName, string seller, string buyer, string gamePrice</p> <p>Used for parsing values from a buy transaction string</p>