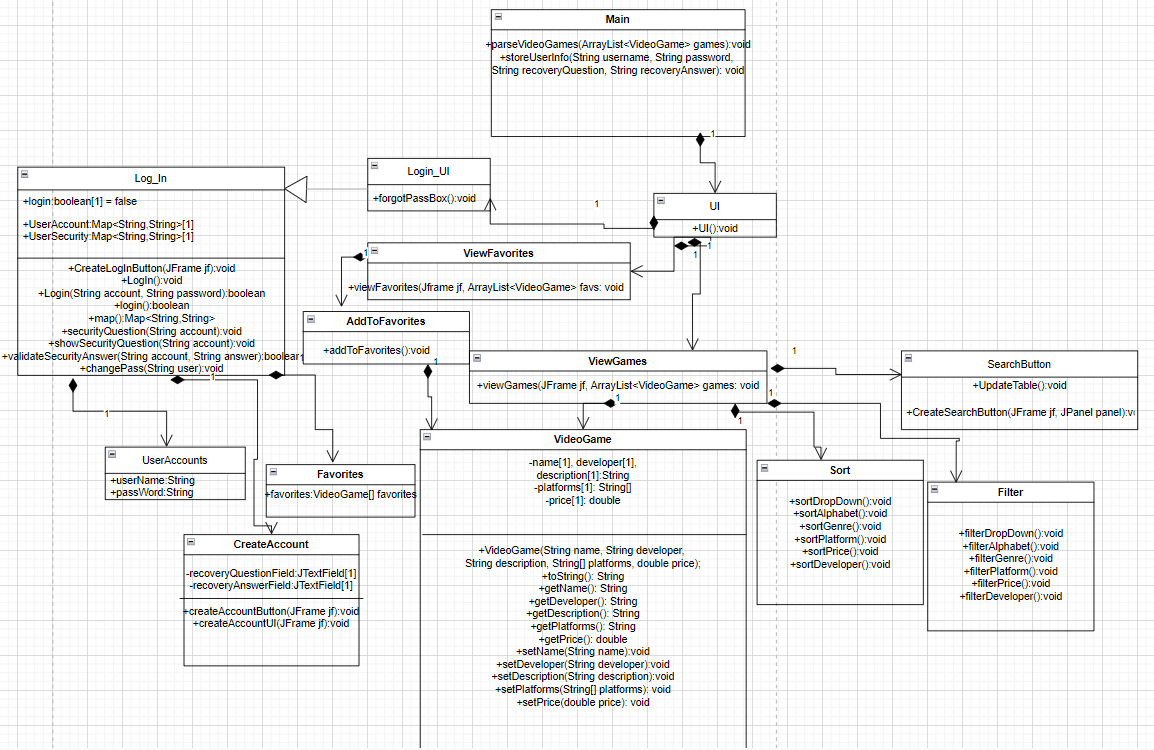
VideoGame Catalog

5Head Technologies

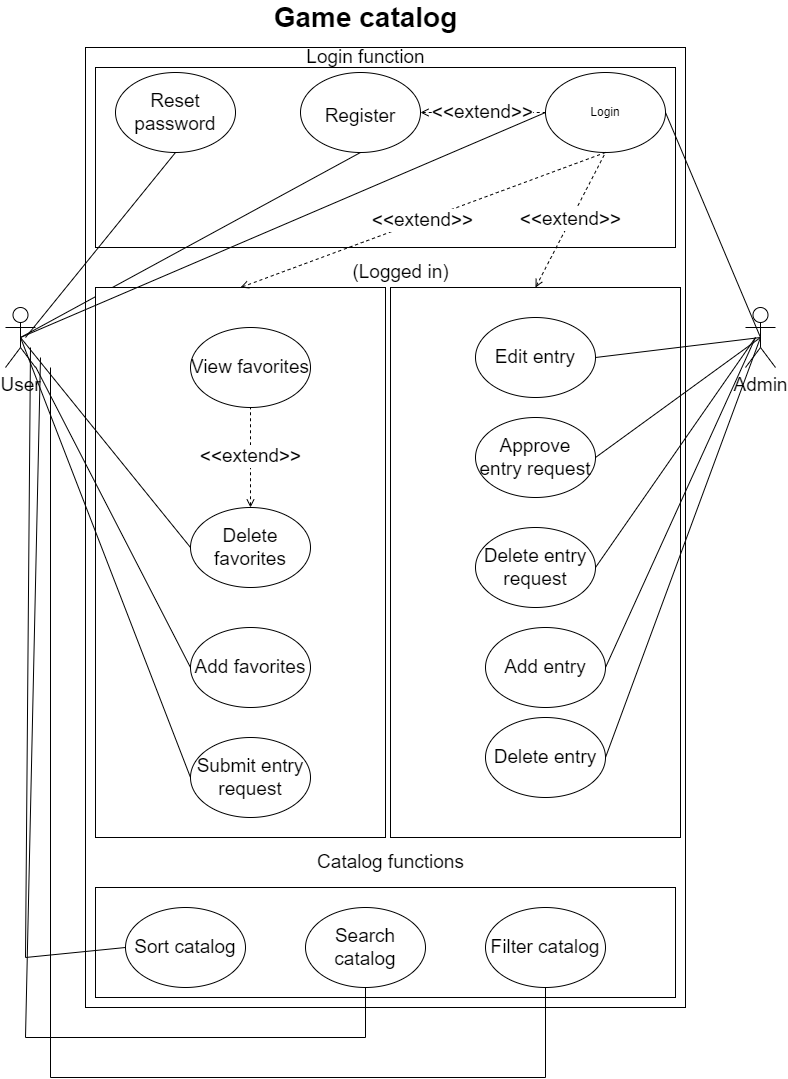
Group 3: Adam Seitz, Hong Chen, Jonny Chen, Dominique Hall

**Diagrams**

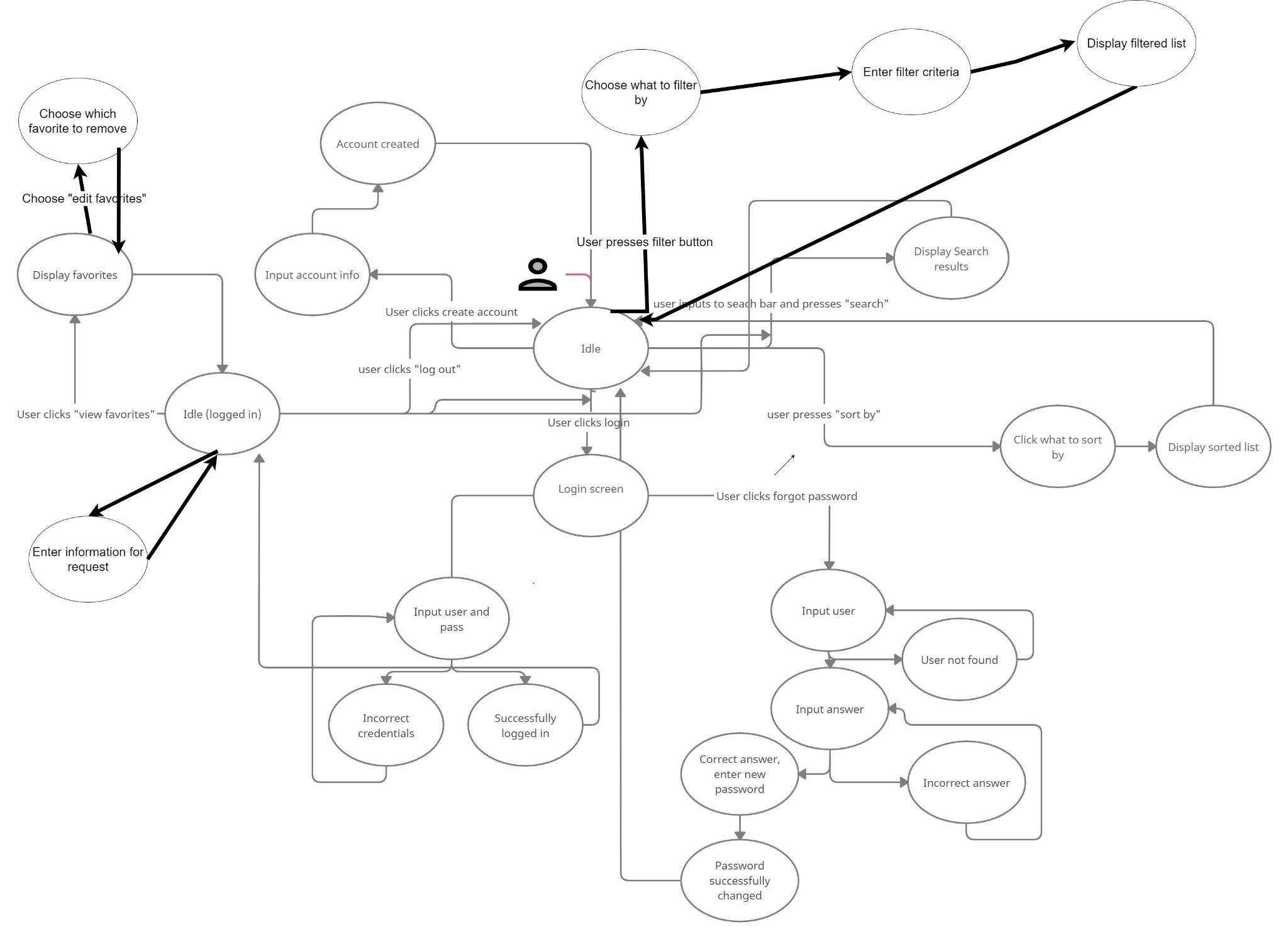
**Class Diagram**

****

**Use Case Diagram**

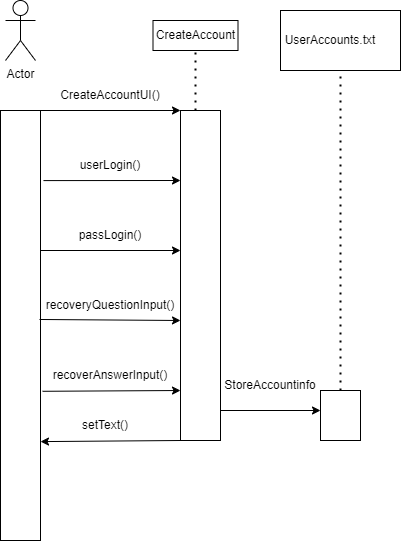
****

**State Diagram**

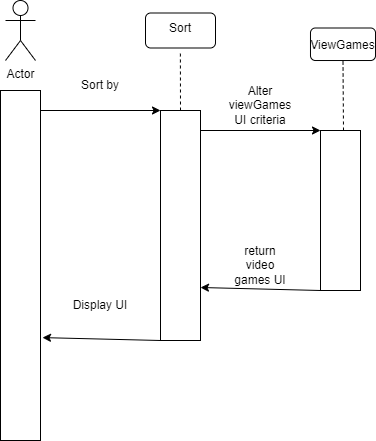
****

**Sequence Diagrams**

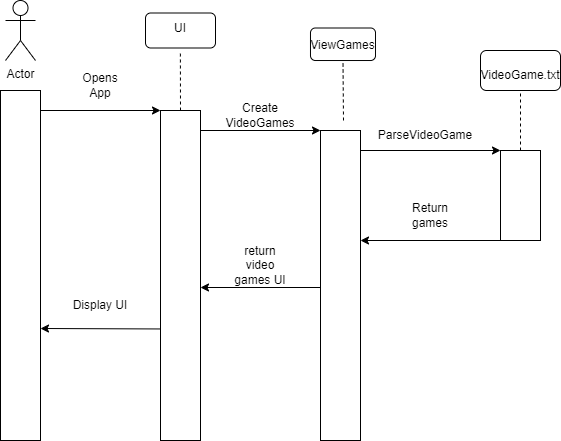
Use Case: The user creates an account.

****

Use Case: The user sorts the table based on the selected criteria.

****

Use Case: The user views the video game catalog when they open the application.

****

**How to Setup and Use**

1. **Loading the program:**

The zip file will need to be downloaded from <https://github.com/chenjlic/CSE201> using the green “Code” drop-down button, then clicking “Download ZIP”.

The zip file will need to be extracted by clicking “extract all” in your file explorer. This will create a new folder with the extracted files in the same location that the zip file was in, and with the same name.

Eclipse IDE must also be downloaded and installed from <https://www.eclipse.org/ide/> using the Download button in the top right corner.

Once both are downloaded, open Eclipse, click the File drop-down menu, and click “Open projects from file system”. Then choose “Directory” and find the project files for the program, and select them. The files will appear in the Project explorer window on the left side of the interface. Use the Project Explorer to find and open “Main.java” in the files folder and click the green arrow at the top to run it.

A window should appear with the catalog application, and Eclipse should show a little red box to indicate that the program is running. To end the program, click the X box at the top right of the window, or click on the red box in Eclipse.

If there is an error that has to do with FileNotFound, check to make sure that the two text files in the project folder are outside of the “default package” folder, but within the project folder.

1. **Filtering, searching and sorting**

You can filter, sort, and search the catalog using the buttons at the top right and the search bar.

To search, enter a search term into the bar and press “search”.

To sort, use the drop-down menu to choose what to sort by.

To filter, click the filter button and use the menu to choose what to filter by, then enter specific filtering criteria.

1. **Logging in**

In order to log in, you must first create an account. The button to create an account is in the top right. You will need a username, a password, and a recovery question.

After your account is made, use the login button at the top right to log in using your username and password. If you forgot your username or password, you can click the “Forgot password” button and use your security question to log in.

1. **Favorites**

To favorite a game, you must first be logged in. When you are logged in, a checkbox will appear below each game entry. Click this checkbox on whichever game(s) you want to favorite, then use the “add to favorite” button in the top right corner. Then, you can click “Show favorites” in the top left corner next to your username, and the catalog will show all of the games you have favorited. Click the “Show full catalog” button in the top right to return to the full game catalog.

1. **Submitting a game**

Use the “add an entry” button in the top left to add a game to the catalog. Enter the name, description, platforms, developer, and price of the game, then press enter and it will appear in the catalog.

**Technical Documentation**

UI.java

Creates the UI components for the application. Calls the following:

JFrame, JPanel, View Games UI, Login UI, Search Bar UI, Search Button UI, Sort menu UI, Filter UI, Add to favorite UI, View Favorite Games UI, Create Account Button UI.

CreateAccount.java

Creates the UI button for creating an account and functionality for checking whether the account information provided is valid.

ViewGames.java

Create a table containing information about video games.

VideoGame.java

Create data type VideoGame that includes information about a video game (name, developer, description, platforms, price).

The swap method is used to swap positions of two different columns of games.

UserData.java

Store and read the user’s account, password, security question, security answers and favorite game in a text file User Account.txt.

Sort.java

Create a sort menu and add events that sort the video game catalog when users use the sort menu.

SearchButton.java

Create the UI of the search button, and when the user clicks it, call the swap method in ViewGames to swap the game the user searched for to the most left of the table.

SearchBar.java

Create a text field and return the text user input to SearchButton class.

Login\_UI.java

Create the login window and forget password windows. It will receive account and password user input and validates using the UserAccount.txt file. Also, users can click forget password and enter the recovery answer they set, then reset their account data in UserAccount.txt.

Log\_In.java

Create a login button, and when the user clicks it, it will call Login\_UI to create a login window.

ViewFavorites.java

Read the data stored in AccountFavorites.txt and change the table to the table which contains the user’s favorite games.

AddToFavorite.java

Add a checkbox row in each video game entry so the user can click the checkbox to add their favorite game to AccountFavoritese.txt.

FilterDropDown.java

Create a filter that filters by developers, platforms, and price.

AccountFavorites.txt

A text file that holds the user’s username and corresponding favorite games.

UserAccounts.txt

A text file that holds the user’s account, password, security question, and security answers.

VideoGameList.txt

A text file that holds the video game’s data which includes name, price, platforms, and developer.

**Library Documentation**

java.util

The java.util package contains the collections framework, legacy collection classes, event model, data and time facilities, internationalization, and miscellaneous utility classes. For our program, we utilize the ArrayList a lot since we are storing the video games in a text file and then loading that information into an array list.

java.io

The java.io package provides for system input and output through data streams, serialization and the file system. For our program we utilize this a lot as we deal with system input and output through the file system many times throughout a run of our program. To start, we store all of the video games and the users favorite video games in separate text files. So to extract the data from the two text files we import File, FileNotFoundException, etc. and then place the input from that file into an array list provided from java.util.

java.awt

The java.awt package provides the classes to create user interfaces and to paint graphics and images. We use this heavily for our program since our entire catalog is on a graphical user interface. We use things such as actionListener, actionEvent, BorderLayout, etc. This helps us place what we use in the javax.swing package around in our interface and how we want them to work with data/each other to provide output for the user.

javax.swing

The javax.swing package provides a set of components that, to the max degree possible, work on all the same platforms. For our program, we use the javax.swing package for many things that we place inside of our GUI. Such as, a JButton, JComboBox, JFrame, JPanel, etc. We use the javax.swing package so our user can use the interface and choose what they would like to do with it. Another part of javax.swing we use is the JPasswordField and JTextField for when we want the user to login to an account. This helps us easily give the user a login interface to access the full functionality of the program.

**Goals and Team Description**

5Head Technologies consists of four team members that have limited technical skills in programming and software design, therefore the main focus of our project is to treat it as a learning experience while doing our best to meet TA expectations for our application. Our application is a catalog of video games that can be searched, filtered, and sorted based on user input. New games can be added through user requests, and users can create an account to add and view their favorite video games. We are using Java for functionality and Java swing for our UI. Data is stored in tab-delimited files. Over the course of this semester, we have learned to use technologies such as Trello to keep track of tasks for team members, GitHub to learn version control and host our code, and discord as a platform for communication. Additionally, we have learned that meeting TA expectations requires a lot of recurring meetings due to changing requirements and unexpected holdups; however, overall, if we were to restart the whole project knowing what we know now, we would function more productively because this project was a fantastic learning experience in software development.

Adam Seitz is the project manager for 5Head Technologies. His skills involve communication, programming, and task management. One of the responsibilities of being a project manager involves communicating with the client in a friendly, professional manner. Adam would greet the client during meetings and make sure the client feels comfortable, important, and heard. Adam would send weekly emails to the client regarding the agenda for the meetings, and if any issues arise or more design information is required, Adam makes sure to get feedback from the client. Additionally, Adam breaks down user stories into tasks and delegates tasks and deliverables to team members. Reminders and important announcements regarding responsibilities or new information are made by Adam on Discord. Lastly, Adam is skilled in programming and has made a lot of contributions in completing tasks using Java and Java swing and helping others with their programming.

Jonny Chen is the technical manager for 5Head Technologies. After working with the team in breaking down the application transcript into functional requirements,[Jonny Chen](mailto:chenjl@miamioh.edu)picked the technologies that best fit the software development process for this application. Jonny picked GitHub for hosting and version control, Discord for communication among team members, Trello for managing tasks, and Google Drive for managing documents and diagrams. Also, Jonny managed and analyzed the burn-down chart for every iteration making sure we have a good understanding of the current work completed and how much work needs to be done by the end of each iteration. Jonny also contributed to a lot of the programming by implementing the login feature as well as the sort feature, and he created the class diagram.

Hong Chen is the programming lead for 5Head Technologies. His skills in Java have helped the team implement features with his expertise in Java Swing. He helped create the table that is used when displaying the catalog of video games. He created the login user interface and made the search bar have functionality. His strong understanding of Java and object oriented-programming transferred to the design process as he knew what classes needed to be made in order to make the code more cohesive. Without Hong Chen, the diagrams would’ve been a lot more difficult to make. During iteration presentations, he understands the code better than anyone else and so he does a majority of the code explanation.

Dominique Hall is our data layer expert. During the designing process of the VideoGame catalog, he created the use case diagrams that gave both the client and the team a better understanding of how the application will function in the future. He also created the state diagram that helped explain the different states the system will be in as well as the different events that occur that alter the system state. Dominique Hall also helped others create the sequence diagrams and class diagrams, and he helped implement a lot of the UI and functionality such as the filter and add entry use cases.