Criterion B: Design­­­

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

# https://documents.lucidchart.com/documents/8cb7cbdf-31eb-42ff-9550-1991fc166ee4/pages/0_0?a=1742&x=-75&y=-24&w=1923&h=958&store=1&accept=image%2F*&auth=LCA%207fc490e91388bfc9e499a7fd8d61bbd7c6330bb9-ts%3D1519451289Flowchart

# Test Plan

|  |  |  |
| --- | --- | --- |
| Test Plan | Nature of Test | Example |
| This application uses Graphical User Interface | Will open trough IDE and a new window of the program should pop up | Open with NetBeans IDE and a new window should pop up with the program |
| Checks that the application allows the user to add a player (enter the players name, position, jersey number, and their statistics) | After clicking on add player, a new panel should pop up and it allows the user to input information in the text fields. | Enter “Dylan Bell” as the name, “Guard” as position, “4” as his jersey number, number of games played is “35”, total points is “594”, total assists is “317” and total rebounds is “218” |
| Checks that the application allows the user to sort it by jersey number, name, or position | In the start panel, click the view players button and a new panel will open up; click the sort dropdown menu which allows the user to sort in different ways | Click the sort button which is a dropdown menu which then allows the user to sort the player by their name, jersey number, or position |
| Checks that the program can successfully calculate the players seasonal average | In the view players panel, click the players name and then click the show statistics button and on the right side it will display the players information and stats | Create a player and enter 35 games played, 594 points, 317 assists, and 218 rebounds; after adding the player, view the player’s stats and the stats should be displayed as 17 PPG, 9.1 APG, and 6.2 RPG. |
| Checks that the user is able to view previous information | If the user exits out the program and reopens it, it will show previous players that have not been deleted | If the user exits the window and closes the program, they can re-run it and all the previous information will be saved |
| Checks that the user is able to edit previous information | In the view player panel, the user can click on the player and then click the edit player button which allows the user to update any information | After creating the player “Dylan Bell,” after clicking the view player button, the user will be able to go in and edit that players information |
| Checks that the user can update previous statistics of the player | After inputting all the updated information, the user must click the update info button and a message will appear that it was successfully updated | After inputting “Dylan Bells” information, click the updated info button which then update the info and then successfully updates the players statistics |
| Checks that the user is able to delete previous players | In the view players panel, the user can click the players name and then click the remove button which then deletes the player and their information | Click on the created player “Dylan Bell” and then click the remove button which then successfully deletes the player |
| Checks that the user can successfully exit out of the program and save all of the information  Test Plan Word Count: 571 | Once returning to the main panel, the user can click the exit button which will close the window, and if reopened all the previous information will be saved | The user creates a player and then exits the program, the user should be able to successfully run it again and the information should be saved and should be able to be viewed when reopened |

# GUI Design Panel Flowchart

# GUI Class Panel

**Start Menu Panel:**

* Clicking “Add Players” button open a new panel which allows the user to create a new player
* Clicking “View Players” opens a new panel and allows the user to view existing players
* Clicking “Exit” will exit the application

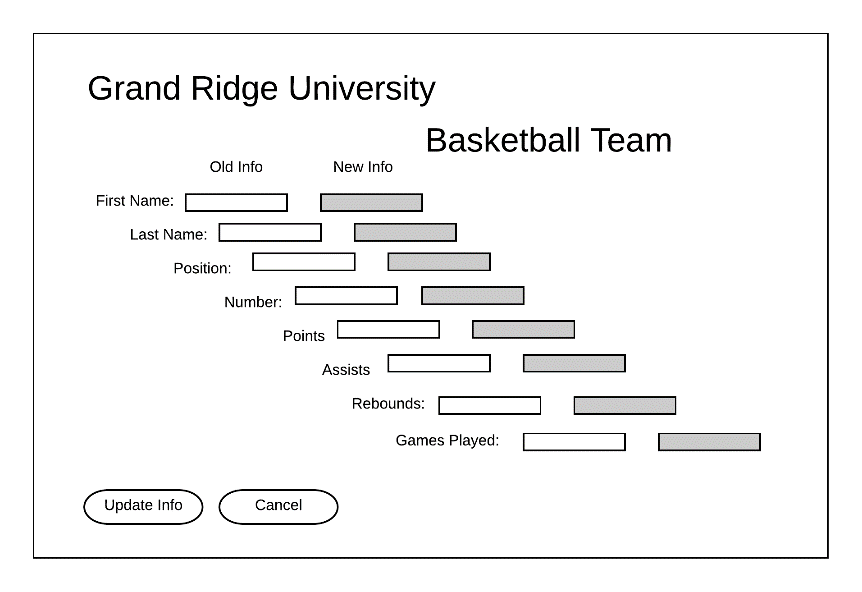
# 

**Add Players Panel:**

* Input all text fields that corresponds which the information given
* Clicking “Create Player” creates the player and shows a successful creation message
* Clicking “Cancel” will return to the start panel
* Error should be handled with an Error Message

**View Players Panel:**

* The sort dropdown menus will allow the user to sort it by name, jersey number, and position and then clicking “Sort” will sort by the option the user chose.
* Clicking “Remove” will remove a player
* Clicking a player’s name a the clicked “Show Stats” will view the players info
* Clicking “Edit Player” will open a new panel which allows the user to edit any previous player.
* Clicking “Cancel” will return to the Start Frame Panel.



**Update Info Panel:**

* Input new information about the player the user would like to edit
* Clicking “Update Info” will show a successful update message and will return to back to the View Players Panel
* Clicking “Cancel” will return back to the View Players Panel

# Players Class

|  |  |  |
| --- | --- | --- |
| Variable Name | Data Type | Description |
| fName | String | Input from the user; enter the first name of the player |
| LName | String | Input from the user; enter the last name of the user |
| jerseyNumber | Int | Input from the user; ienter the jersey number of the player |
| Position | Int | Input from the user; enter the position of the player |
| totalGames | Int | Input from the user; enter the total games played by the player |
| Points | Int | Input from the user; enter the total points scored by the player |
| assists | Int | Input from the user; enter the total assists by the player |
| rebounds | Int | Input from the user; enter the total rebounds by the player |

|  |  |  |  |
| --- | --- | --- | --- |
| Method Name | Parameters | Return Type | Description |
| Player | (String f, String l, int totalg, int jN, int re, int as, String pos, int pt) | String | Creates a player based on the input of the user and adds the player to another class |

# startFrame Class

|  |  |  |
| --- | --- | --- |
| Variable Name | Data Type | Description |
| pl | String | Total number players that were created |

|  |  |  |  |
| --- | --- | --- | --- |
| Method Name | Parameters | Return Type | Description |
| startFrame | none | void | Reads from the file |
| File | none | void | Creates a file when the user exits the program and it will output to a file and all information will be saved. |

# viewPlayers Class

|  |  |  |
| --- | --- | --- |
| Variable Name | Data Type | Description |
| seeing | String | Array of the number of players created |
| playerList | String | Array of the list of all the players that were created |

|  |  |  |  |
| --- | --- | --- | --- |
| Method Name | Parameters | Return Type | Description |
| ViewPlayers | (Player[] fat) | String | Checks if the players are created, if not found than it doesn’t show up |
| seeList | none | Void | Traverses through an array that hold that class and view all the created players |

# newPlayer Class

|  |  |  |
| --- | --- | --- |
| Variable Name | Data Type | Description |
| list | String | An array of the all the players and their statistics |

|  |  |  |  |
| --- | --- | --- | --- |
| Method Name | Parameters | Return Type | Description |
| newPlayer | None | String | A string in that class that shows if the player has been created |
| newPlayer | (Player[] pl) | String | To create a player to the application |

# updateInfo Class

|  |  |  |
| --- | --- | --- |
| Variable Name | Data Type | Description |
| Ppl | String | Returns true if all text fields were inputted |
| temp | int | Returns false with an error handle message if all text fields are not inputted |

|  |  |  |  |
| --- | --- | --- | --- |
| Method Name | Parameters | Return Type | Description |
| updateInfo | None | String | Takes existing players and then allows the user to edit them |
| updateInfo | (Player[] p, Player t) | string | Takes past players and updates the information |

Method Plan: 372 Words