Use Cases

Use Case: Add Boat Layout

Use Case: Edit Boat Layout

Use Case: Delete Boat layout

Use Case: View Boat Layout

Use Case: Search for layout

Use Case: Add Teammate

Use Case: Edit Teammate

Use Case: Delete Teammate

Use Case: Search for Teammate

Use Case: View Performance Analytics Chart

Use Case: Search for Performance Analytics Charts

Use Case: Delete Performance analytic

Use Case: Record GPS Data

Use Case: Create Performance Analytics Chart

**Diagram:**

|  |  |
| --- | --- |
| **Use Case Name** | Add Boat Layout |
| **Trigger** | The User selects to add a new boat layout to the database |
| **Precondition** | The User has accessed the Layouts screen |
| **Basic Path** | 1. The user clicks the “Add” button displayed on the layouts screen 2. The system presents a new screen where the user is prompted to give a layout name and layout size 3. The user decides if they want to use a predetermined size layout, a custom layout or a previously used layout 4. Upon selection the layout has been created and entered into the database |
| **Postcondition** | The database has been updated. |
| **Exception Paths** | The User may abandon the operation anytime |
| **Other** | The layout information includes the name of the layout and the number of seats to be included |

|  |  |
| --- | --- |
| **Use Case Name** | View Boat Layout |
| **Trigger** | The User selects a Layout on the Layouts screen |
| **Precondition** | The User has accessed the Layouts screen |
| **Basic Path** | 1. The user selects the desired layout on the Layout screen |

|  |  |
| --- | --- |
| **Use Case Name** | Edit Boat Layout |
| **Trigger** | The User selects to edit a selected Boat Layout |
| **Precondition** | The User has accessed the Layouts screen and has selected a Layout to view. |
| **Basic Path** | 1. The user selects the boat layout they wish to edit 2. The user can drag paddlers existing paddlers on the layout onto different positions 3. The user can open a sliding drawer menu with a repository of existing paddlers and can filter through by name and position 4. The user can drag selected paddlers onto the layout 5. The user saves the layout and the information is stored in the database |
| **Postcondition** | The database and layout have been updated. |
| **Alternative Paths** | In Step 2 the user can have a sorting algorithm automatically assign paddlers onto the boat depending on weight and optimal weight distribution onto the boat. This is only if Paddlers exist onto the boat  The user can also edit the name of a layout |
| **Exception Paths** | The User may abandon the operation at any time |

|  |  |
| --- | --- |
| **Use Case Name** | Delete Boat Layout |
| **Trigger** | The User selects to delete a selected Boat Layout |
| **Precondition** | The User has accessed the Layouts screen and has selected a Layout to view. |
| **Basic Path** | 1. The user selects the Layout they wish to delete 2. The user selects the delete option and is given a confirmation prompt 3. If selected yes, the selected layout is deleted |
| **Postcondition** | The layout has been deleted and the database has been updated |
| **Alternative Paths** | The user can also delete a layout from the Layouts screen. By holding a layout to enable the delete prompt in the Layouts screen and selecting a layout to be deleted |
| **Exception Paths** | The User may abandon the operation at any time |

|  |  |
| --- | --- |
| **Use Case Name** | Search for Boat Layout |
| **Trigger** | The User chooses to search for a Boat Layout in a search bar |
| **Precondition** | The User has accessed the Layouts screen |
| **Basic Path** | 1. The user selects the search navigation bar 2. The user can filter layouts by size 3. The user enters the name of the layout they wish to find |
| **Postcondition** | The user is given a filtered list of all layouts that match their criteria |
| **Exception Paths** | The User may abandon the operation at any time. If there are no existing layouts, the search option will return nothing. |

|  |  |
| --- | --- |
| **Use Case Name** | Add Teammate |
| **Trigger** | The User selects to add a new teammate to the database |
| **Precondition** | The User has accessed the Team screen |
| **Basic Path** | 1. The user selects the “Add” button displayed on the Team screen 2. The system presents a screen with empty text fields regarding teammate information 3. The user enters the appropriate information into the fields 4. The system checks if the name field is not blank and adds them to the database |
| **Postcondition** | A teammate has been added and the database has been updated |
| **Exception Paths** | The User may abandon the operation at any time |
| **Other** | Teammate information includes, name, email address, phone number, weight, height, age, if they’re a paddler, and paddling side preference, |

|  |  |
| --- | --- |
| **Use Case Name** | Edit Teammate |
| **Trigger** | The User selects to edit an existing teammates information |
| **Precondition** | The User has accessed the Team screen |
| **Basic Path** | 1. The user selects a teammate they wish to edit in the Team screen 2. The user views the teammates information in their details screen and selects the “Edit” option 3. The text fields become editable and the user edits the information they wish to modify 4. The user clicks the save option |
| **Postcondition** | The teammates information has been modified and the database has been updated. |
| **Exception Paths** | The User may abandon the operation at any time |
| **Other** | This use case can be used to add categories for an article, to correct typographical errors, or to remove a reviewer who has missed a deadline for returning a review. It may also be used to allow access to the named use case to enter an updated article or a review for an article. |

|  |  |
| --- | --- |
| **Use Case Name** | Delete Teammate |
| **Trigger** | The User selects to delete a teammate to the database |
| **Precondition** | The User has accessed the Team screen |
| **Basic Path** | 1. The user selects a teammate they wish to delete from the database 2. The user selects the edit option to change the accessible fields 3. The user selects the newly shown delete button 4. The user is given a prompt to confirm their choice and selects yes to confirm the delete operation |
| **Alternative Paths** | The user can also delete a teammate from the Team screen. By holding a teammate to enable the delete prompt in the Team screen and selecting the teammate to be deleted |
| **Postcondition** | The user has been deleted and the database has been updated. |
| **Exception Paths** | the User may abandon the operation at any time. |

|  |  |
| --- | --- |
| **Use Case Name** | Search for Teammate |
| **Trigger** | The User selects to search for a teammate in the Team screen |
| **Precondition** | The User has accessed the Team screen |
| **Basic Path** | 1. The user selects the search navigation bar 2. The user can filter teammates by name, paddling side preference, weight and height 3. The user enters the name of the teammate they wish to find |
| **Postcondition** | The user is given a filtered list of all teammates that match their criteria |
| **Exception Paths** | The User may abandon the operation at any time. Additionally, if there are no teammates in the database the search will retrieve nothing |

|  |  |
| --- | --- |
| **Use Case Name** | View Performance Analytics Chart |
| **Trigger** | The User selects to a performance chart to view |
| **Precondition** | The User has accessed the Analytics screen |
| **Basic Path** | 1. The user selects the desired chart in the Analytics screen |
| **Postcondition** | The user is routed to a screen displaying performance analytics about boat layouts for their selected chart |

|  |  |
| --- | --- |
| **Use Case Name** | Search for Performance Analytics Charts |
| **Trigger** | The User selects to add a new boat layout to the database |
| **Precondition** | The User has accessed the Team screen |
| **Basic Path** | 1. The user selects the search navigation bar on the Team screen 2. The user can filter charts by layouts, name, date and rating 3. The user can search by name of the chart |
| **Alternative Paths** | In Step 1 the user can be redirected from viewing a Boat Layout to searching for a performance chart. The search filter will be set to the information regarding the selected layout and information regarding that layout will be shown |
| **Postcondition** | A displayed filtered list of analytics charts depending on the search criteria |
| **Exception Paths** | The User may abandon the operation at any time. Additionally, if there are no charts in the database the search will retrieve nothing |

|  |  |
| --- | --- |
| **Use Case Name** | Delete Performance Analytic Chart |
| **Trigger** | The User selects to chart to delete from the database |
| **Precondition** | The User has accessed the Analytics screen |
| **Basic Path** | 1. The user selects the chart they wish to delete 2. The select the options menu in the top right and select the “delete” option 3. The user is given a prompt confirming if they wish to delete the chart |
| **Alternative Paths** | The user can also delete a chart from the Analytics screen. By holding a chart to enable the delete prompt in the Analytics screen and selecting the chart to be deleted |
| **Postcondition** | The chart has been deleted and the database has been updated. |
| **Exception Paths** | The User may abandon the operation at any time. |

|  |  |
| --- | --- |
| **Use Case Name** | Create Performance Analytics Chart |
| **Trigger** | The User given a prompt to save a chart |
| **Precondition** | The user has completed a route on the Ready screen and has GPS permissions enabled |
| **Basic Path** | 1. The user is given a prompt to save a performance analytics chart for the just completed boat route 2. The user can choose to save the route or not to save it 3. If the user saves, the system displays a text field to give the chart a name |
| **Postcondition** | The user saves a performance chart, the database will be updated with that information |
| **Exception Paths** | The user may abandon the operation at any time. If the user has not enabled GPS permissions, they cannot access this use case |

|  |  |
| --- | --- |
| **Use Case Name** | Record GPS Route |
| **Trigger** | The User selects the Go button on the Ready screen |
| **Precondition** | The User has accessed the Ready screen and has enabled GPS permissions in the application |
| **Basic Path** | 1. The user selects the layout they wish they use to record GPS data for 2. The user selects the start button to begin recording data 3. When the user decides to stop recording data they can press the stop button 4. Once the recording is stopped the user is given a chance to save a performance chart of the GPS data for that layout |
| **Alternative Paths** | In step 3 the user can press the pause button to optionally stop or restart recording data. And can further press stop again to stop for good.  In step 1, the user can choose not to use a layout and record a GPS route anonymously and save the chart as an anonymous layout |
| **Postcondition** | If the user chooses to save a performance chart, the database will be updated with that information |
| **Exception Paths** | The user may abandon the operation at any time. If the user has not enabled GPS permissions, they cannot access this use case |