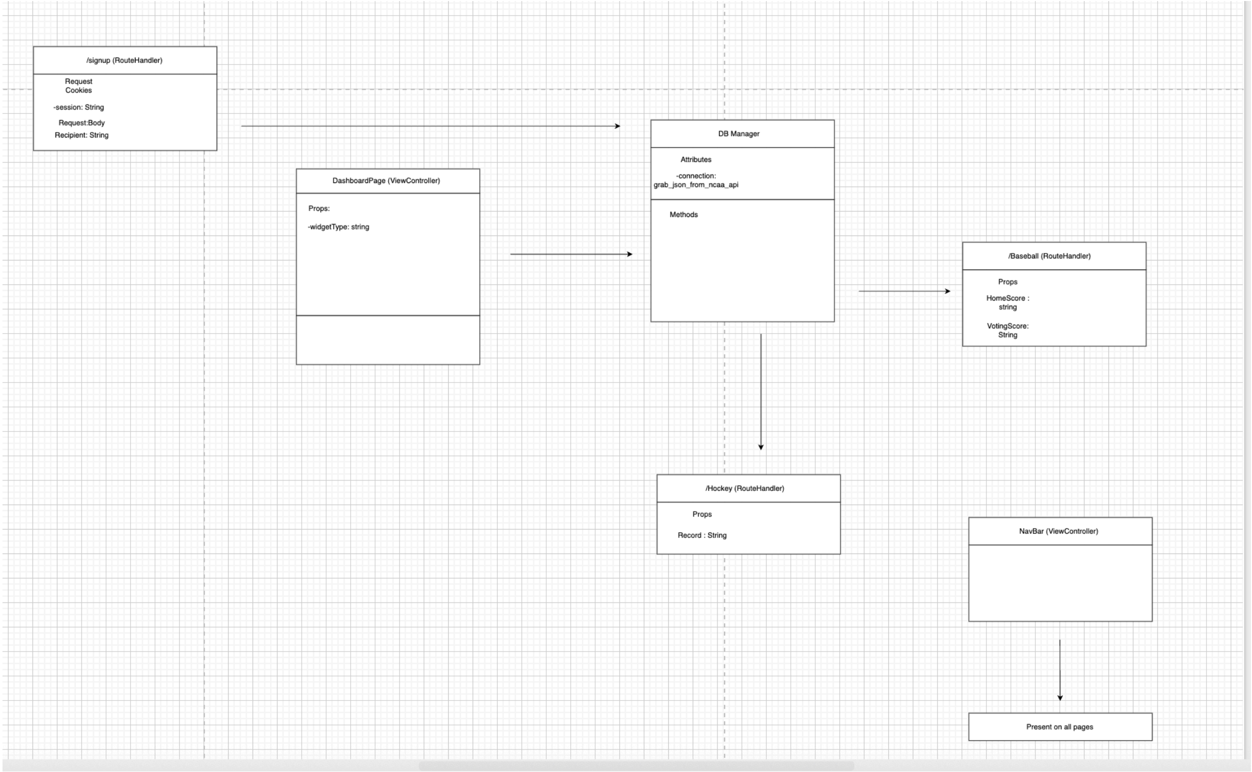
<For each part:>



(link to at least a commentable and readable version of the diagram) [*https://drive.google.com/file/d/1hkiTL69nkp8aT-k0RKeNknLqpmxhqkWj/view?usp=sharing*](https://drive.google.com/file/d/1hkiTL69nkp8aT-k0RKeNknLqpmxhqkWj/view?usp=sharing)

<narrative description of the part, for example:

"The diagram above shows the component hierarchy and their applications in our sports widget application to view information and scores. Solid arrows indicate Uses. As shown in the diagram, the widget component uses both Hockey and Baseball components through the strategy pattern - with more being added in future updates. This can be seen in the widgetType prop which determines which sport component to render and show. Without this strategic delegation, the Widget would need direct knowledge of each sport's implementation details. All data used by the sports components comes from either static JSON files (hockey) or live API. >

Design Patterns

Name of design pattern e.g. Facade

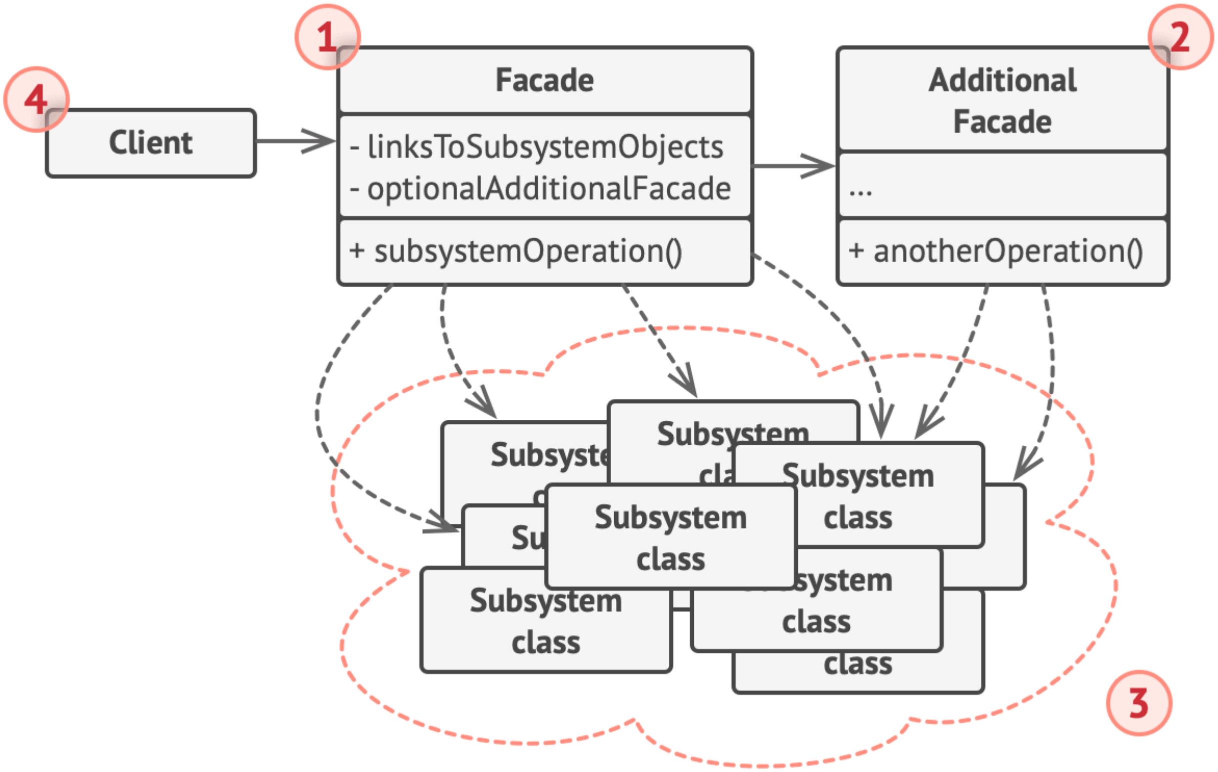
Design Pattern: Strategy
Used In: Widget Component System

The Black Bear Bulletin uses a Strategy pattern to manage the dynamic rendering of different sports components (ex. Hockey). The Widget component serves as the context that delegates to various specific sports based on the widgetType. This abstraction isolates the parent components from needing to know about the implementation details of individual sports widgets (Hockey and Baseball)

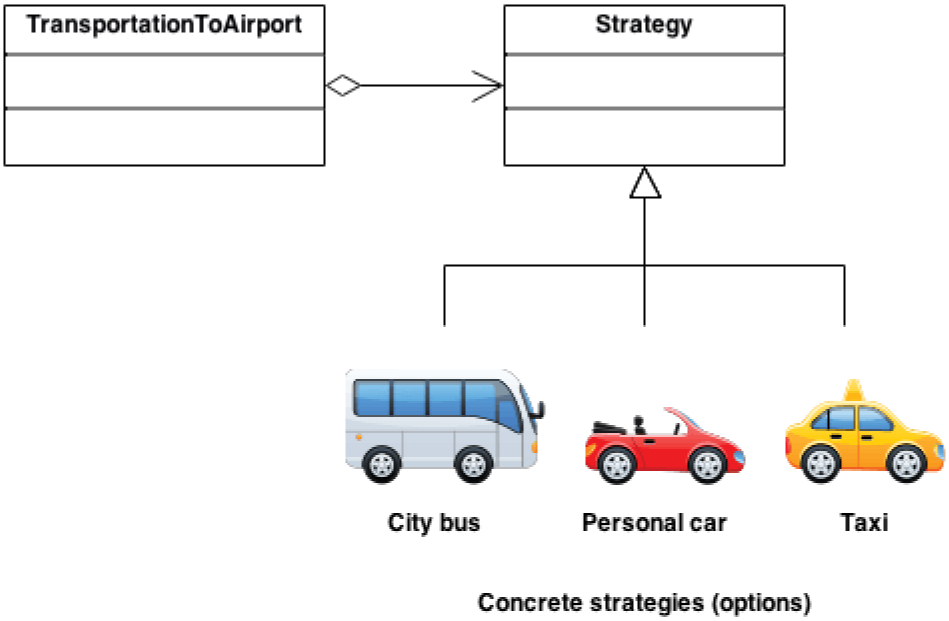
<Have a sub-diagram/excerpt of your design class diagram showing the design pattern.
(link to at least a commentable and readable version of the diagram)

(insert image here that can be viewed without following a link, e.g a screenshot of the diagram)>

<Have a the diagram that defines that design pattern from the lecture slides or from this website (example), and double check it's actually following the pattern!>



(If you used other design patterns, you may optionally include other design pattern examples)



Where city bus, personal car, and taxi could represent various different sports, and TransportationToAirport represents widgetType.