DESIGN AND ANALYSIS OF SOFTWARE SYSTEMS - S24

DRAWING EDITOR - REPORT

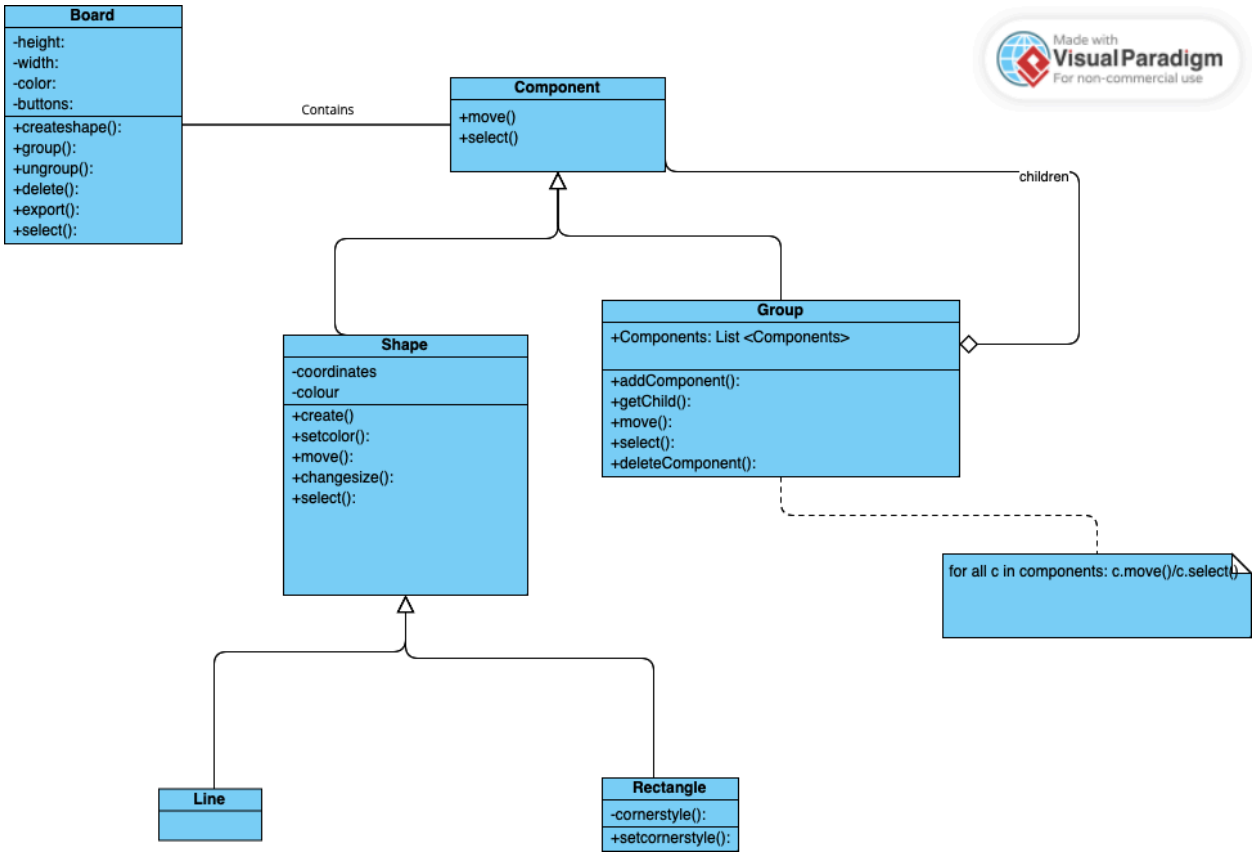
KUSHAL SHAH (2022101059),

TEJAS CAVALE (2022101087),

ADITYA MISHRA (2022101047),

PRITHVI KARTHIK (2022101020)

UML CLASS DIAGRAM:



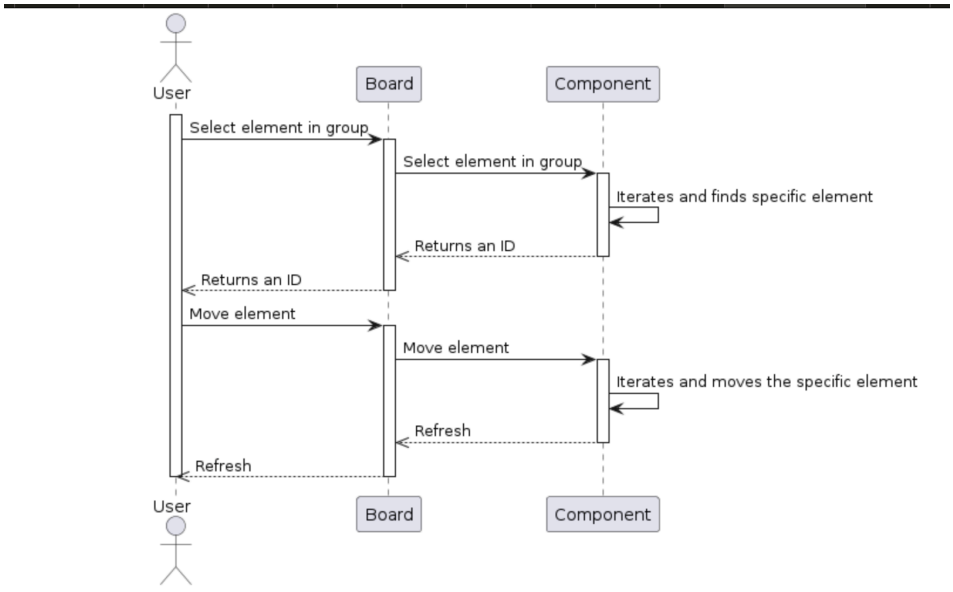
CLASS RESPONSIBILITIES:

Class Name	Responsibilities
1. Board	- Act as a front end to the client. - Selects needed parts in the "Component" to get needed segment. - Facilitates Saving / exporting files to necessary formats.
2. Component	- Facilitates the complex structure of groups and subgroups. - Has a composite pattern with groups and shapes to create a simple and understandable design.
3. Shape	- Contains the methods and data related to various properties such as size , colour , cornerstyle etc. - Can be created, edited and deleted. - Lines and Rectangles are the generalisations/subclasses of this superclass.
4. Group	- Contains a list of components as created when user groups specific items. - Can be moved , selected , ungrouped or deleted etc.

SEQUENCE DIAGRAMS:

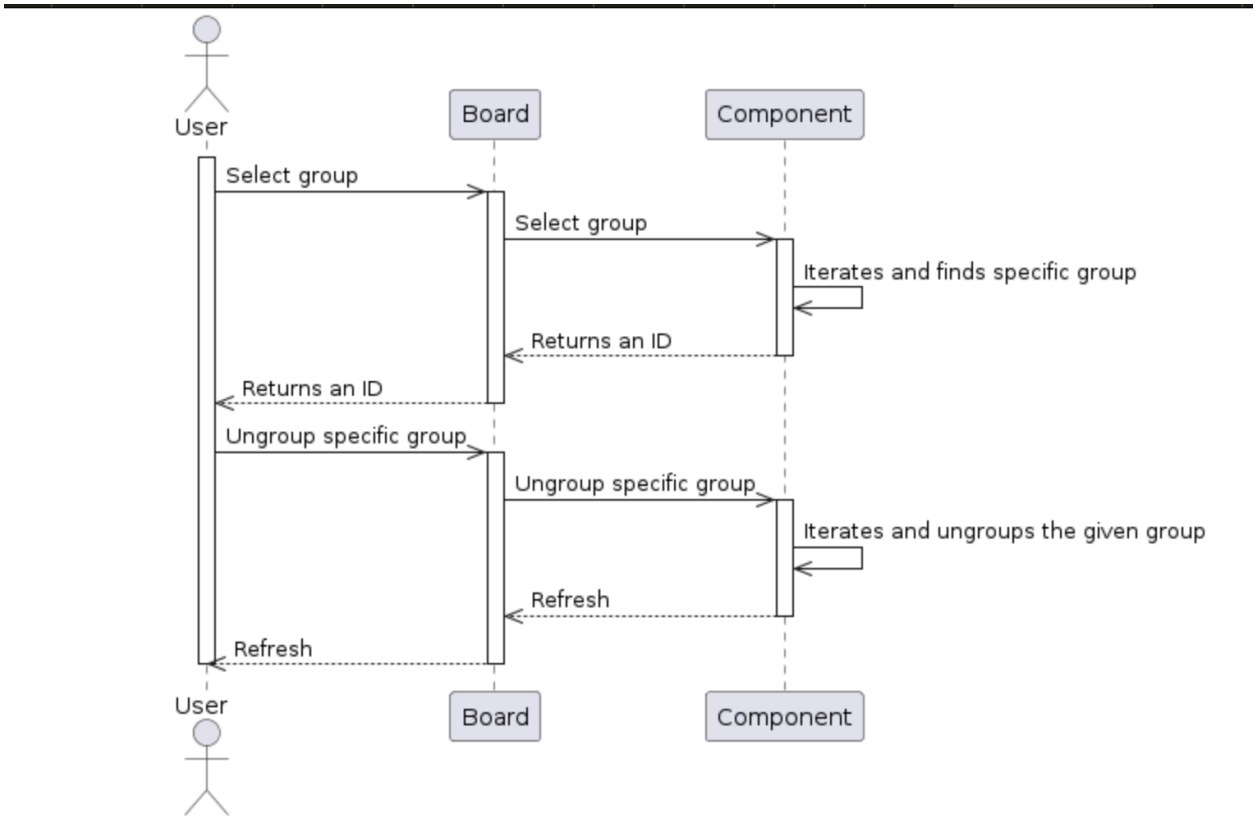
1) Moving an element:

USECASE: Mouse selection of an element that is in a group, drag to a new position, update of the model, refresh of the drawing.



2) Ungrouping:

USECASE: Mouse selection of an element that is in a group, ungrouping of the elements in the group



DESIGN PRINICPLES:

1. **Composite Pattern:** We have utilized the Composite pattern to manage the structure of grouping shapes and groups within our system. This pattern allows us to treat individual shapes and groups of shapes uniformly, enabling us to compose complex structures while keeping the interface consistent.
2. **Law of Demeter:** We adhere to the Law of Demeter by ensuring that the Board class only accesses the Component class, which in turn interacts with groups and shapes. This principle promotes loose coupling and encapsulation, as Board does not have direct knowledge of the internal structure of groups and shapes, reducing the risk of unintended dependencies and simplifying maintenance.
3. **Extensibility and Flexibility:** Our design is highly extensible, allowing for the seamless addition of new shapes, export formats, and operations on drawing objects. New shapes, such as ellipses, can be incorporated as subclasses of existing shape classes, ensuring compatibility with existing functionality. Additionally, new export formats and operations can be added without modifying existing code, promoting modularity and ease of maintenance.
4. **Reusability**: The user interface (UI) components of our system are designed for reusability across various diagramming purposes. This reusability enhances productivity by allowing us to leverage existing UI components for different types of diagrams, reducing development time and effort.
5. **Separation of Concerns:** We have achieved a clear separation of concerns by delineating responsibilities between the Board class and the Component class. The Board class is responsible for managing the frontend UI, while the Component class handles backend operations related to groups and shapes. This separation enhances maintainability and scalability by isolating changes to specific areas of the system and minimizing the ripple effects of modifications.