# Ontwerpdocument

**Stap 1**

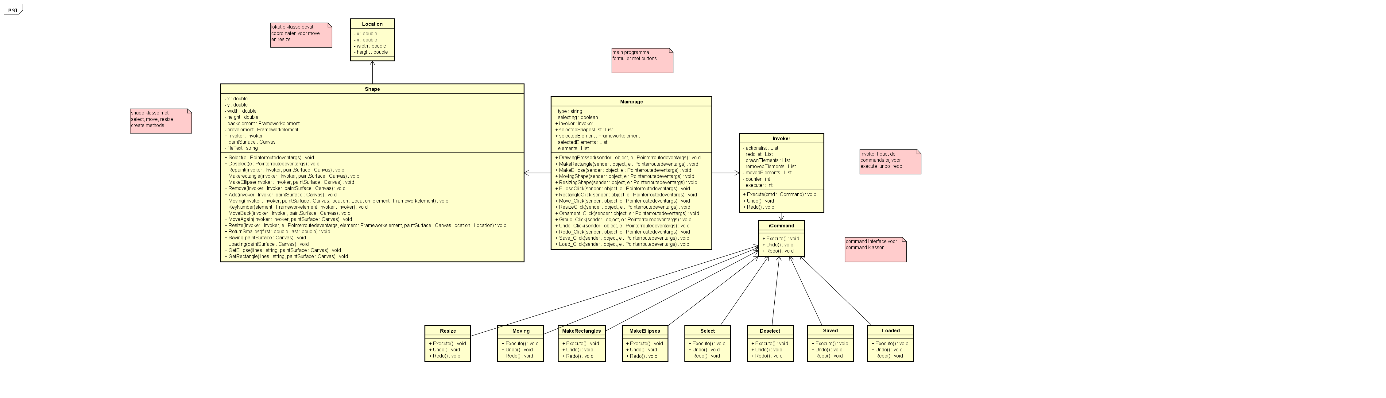
Dit heeft het main formulier met buttons. Het kan create, resize en move uitvoeren. Er moet eerst een rectangle of ellipse geselecteerd worden voordat een move of resize kan. Create kan door op de ellipse of rectangle button te klikken en vervolgens op een lege plek op het canvas.

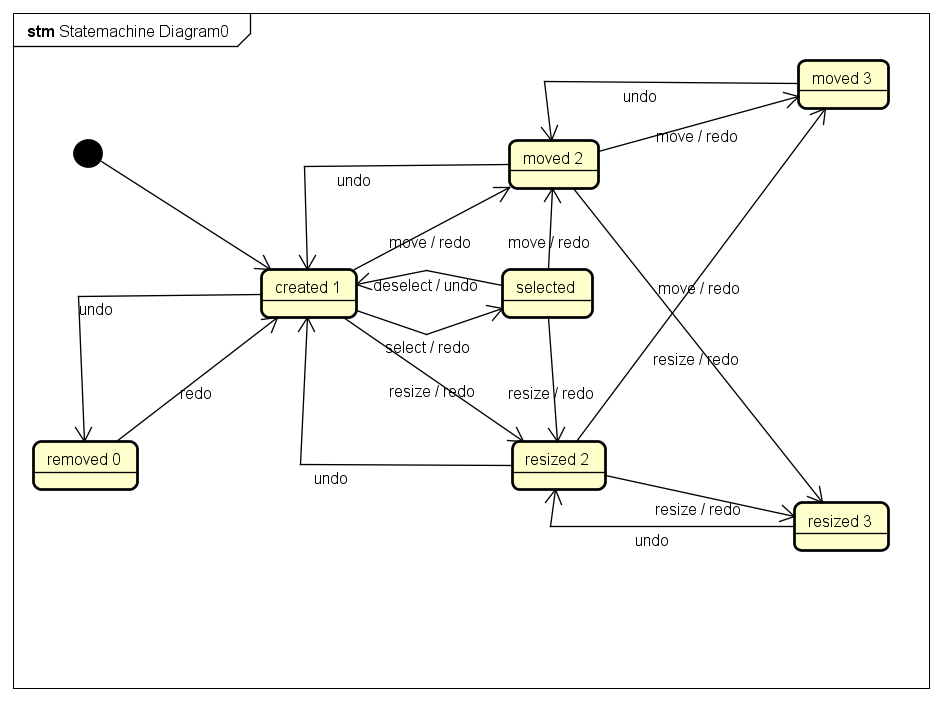
**Stap 2**

De shape bevat de methods voor create, move, resize en select.

De afzonderlijke commands klassen behoren bij de interface iCommand. Deze wordt in de invoker opgeslagen en bijgehouden voor de undo en redo.

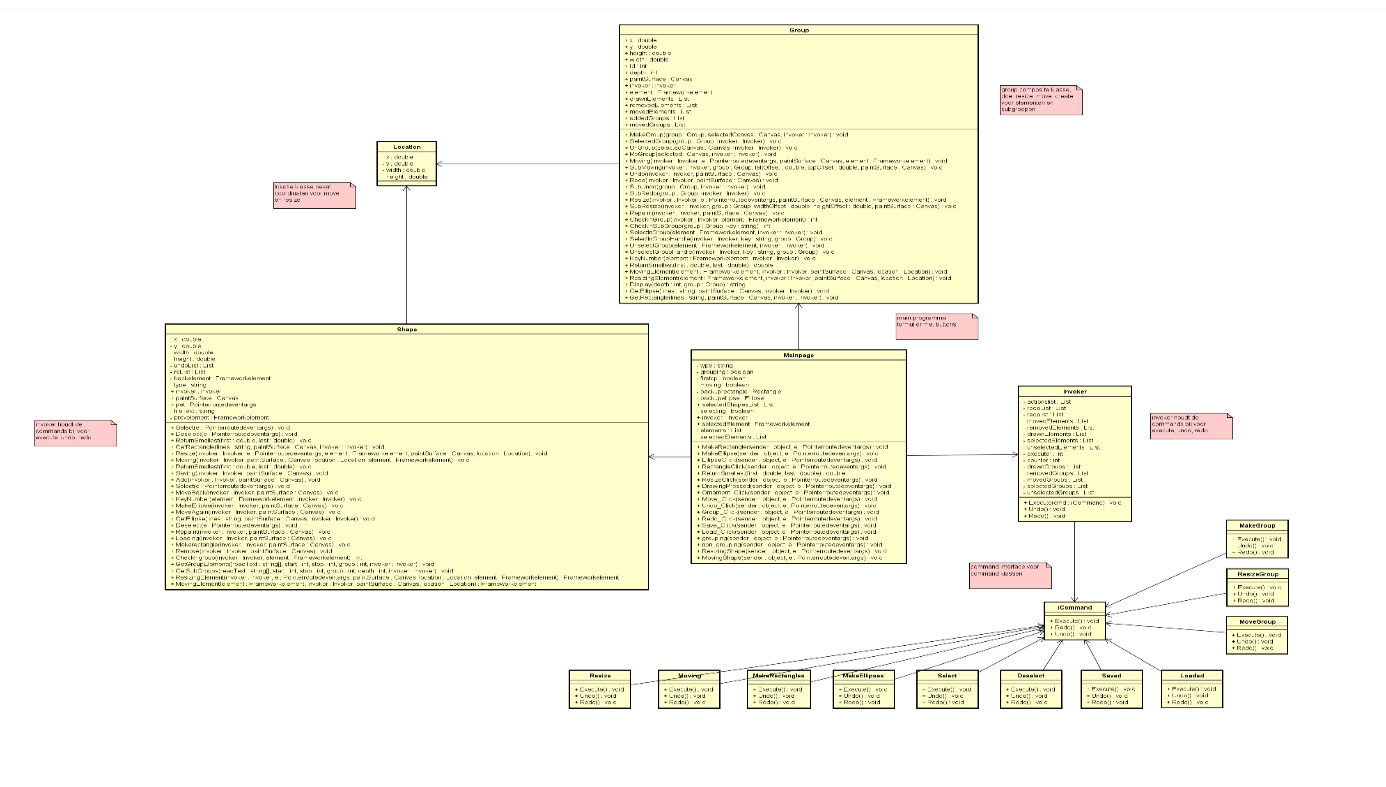
De location klasse bevat coördinaten voor de move en resize.



****

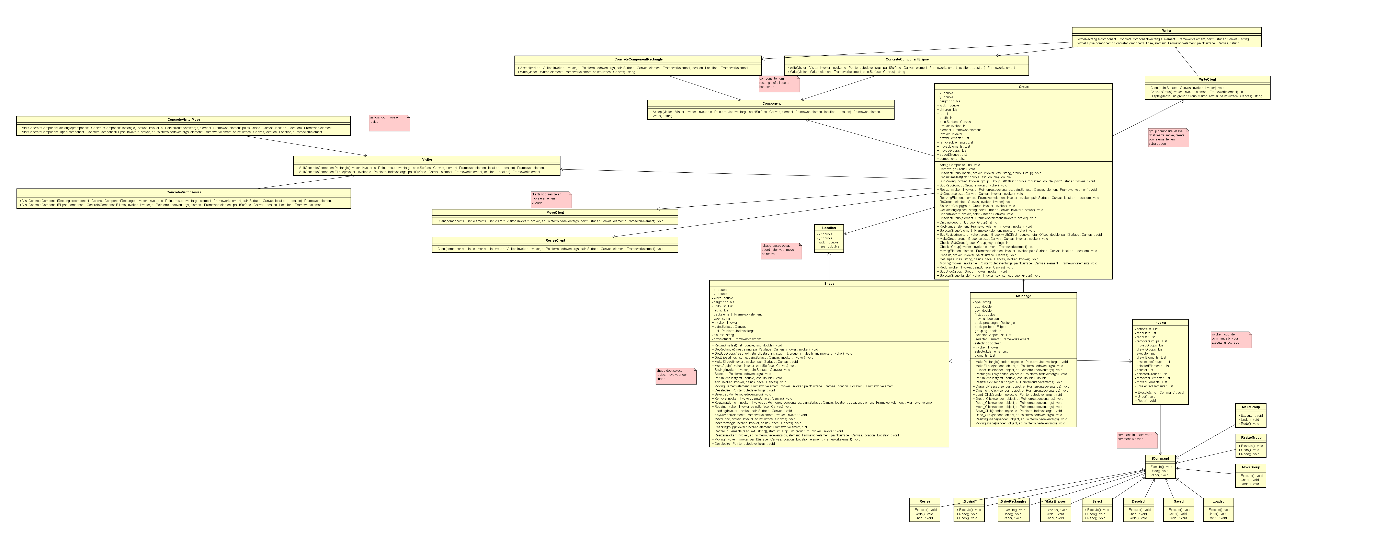
**Stap 3**

De group bevat de composite klasse. Deze voert uit voor de ellipse en rectangle in een groep en voor subgroepen, dat wilt zeggen groepen in een groep. De composite group heeft recursie.



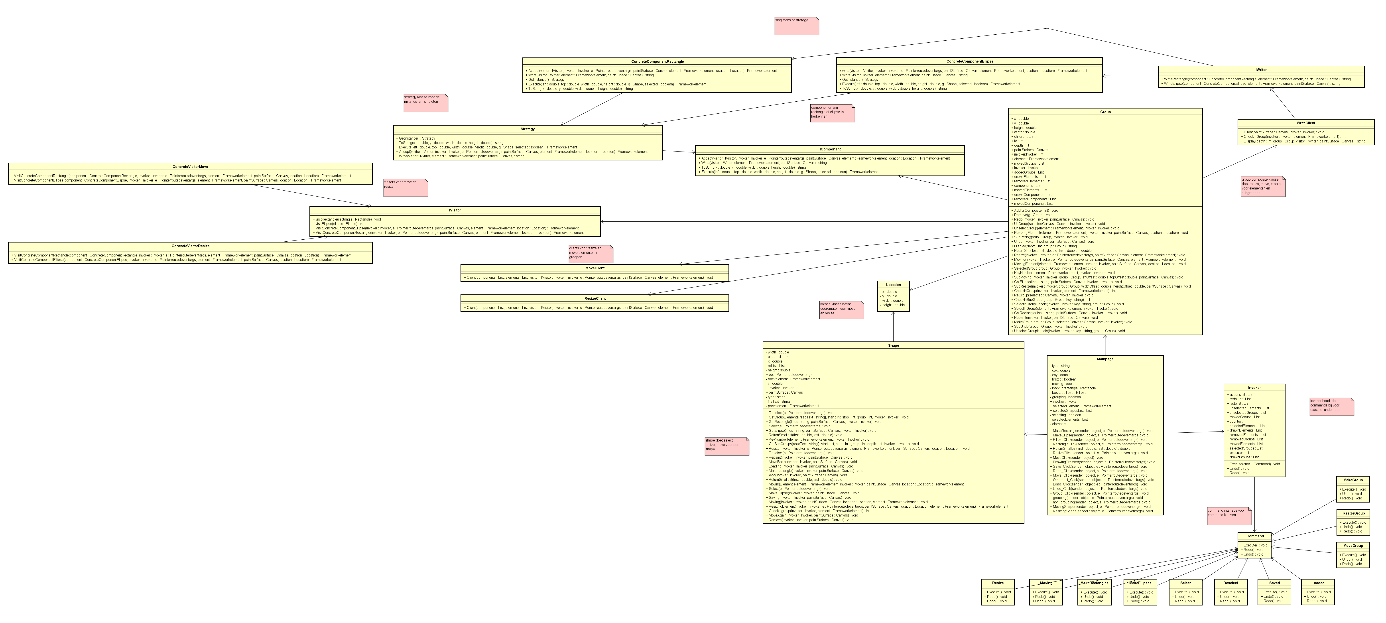
**Stap 4**

Writing naar file wordt met de WriteClient gedaan. Move en resize voor een composite groep met de MoveClient en Resizeclient. Er is een visitor voor de move dit is ConcreteVisitorMove en voor de resize is dit ConcreteVisitorResize. Er is voor ellipse en rectangle componenten klassen zoals de ConcreteComponentRectangle en ConcreteComponentEllipse. Deze helpen herkennen om wat voor component, lees rectangle of ellipse, het gaat.



**Stap 5**

Er is een Strategy klasse, die als basis is voor een shape en de ConcreteComponentEllipse en ConcreteComponentRectangle zijn de singleton ervan. Dat zijn de twee klassen die eerder al concrete componenten waren bij stap 4 voor de visitor pattern. Deze en de component interface zijn refactored.



**Stap 6**

Met de decorator klasse wordt een ornament toegevoegd aan een ornamentenlijst van een groep en rectangle of ellipse component. Om aan te geven dat de ornament bij een groep is wordt een element in een groep geselecteerd op het canvas. Met de tekstvak top, kan de positie aangegeven worden, zoals top, left, right, bottom en met tekstvak ornament de naam.

