

COS 214 Project

Project Summary and Reports



Julianne Venter, u20433752 (Team Leader)

Ronin Brookes, u19069686

Sameet Keshav, u21479373

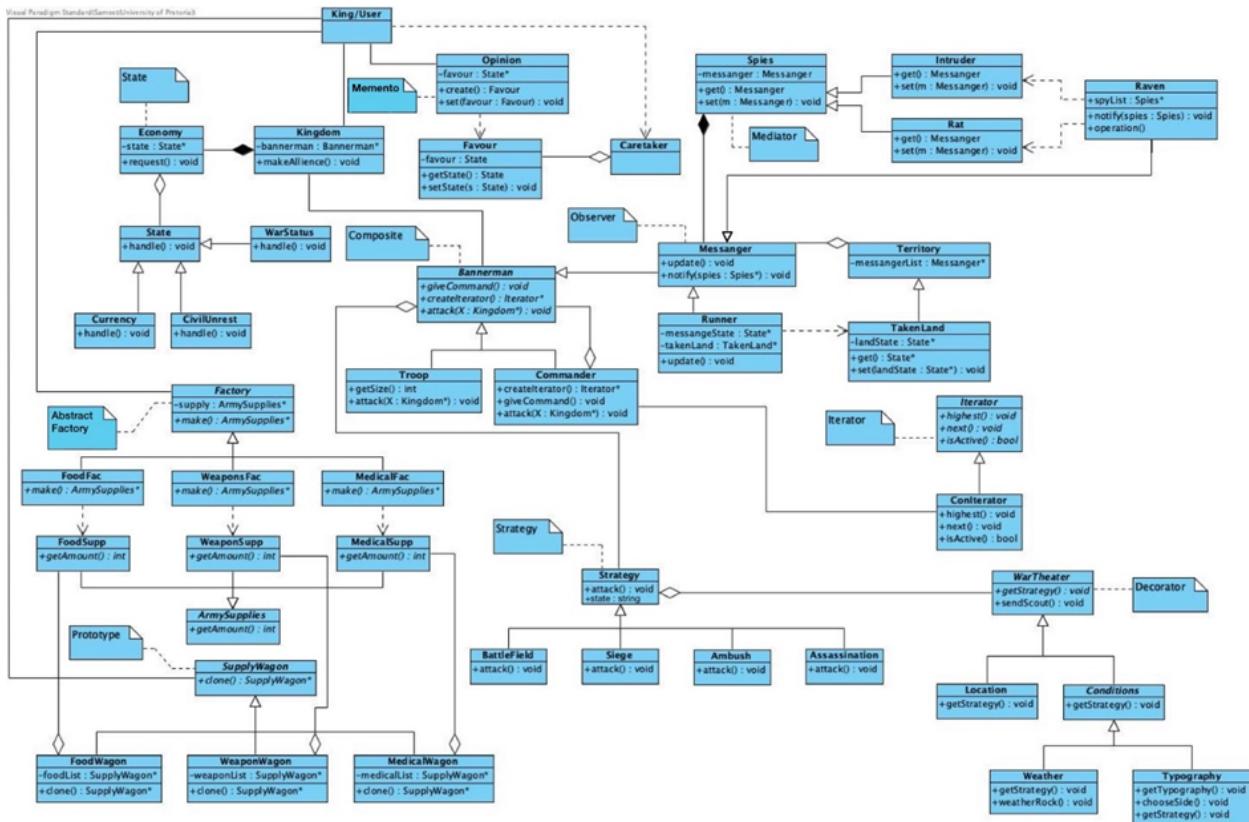
Keabetswe Mothapo, u21543462

Morgan Bentley, u18103007

Thapelo Thoka, u21499749

Group Name: PerfectStrangers

Task 1: Pre-initial design



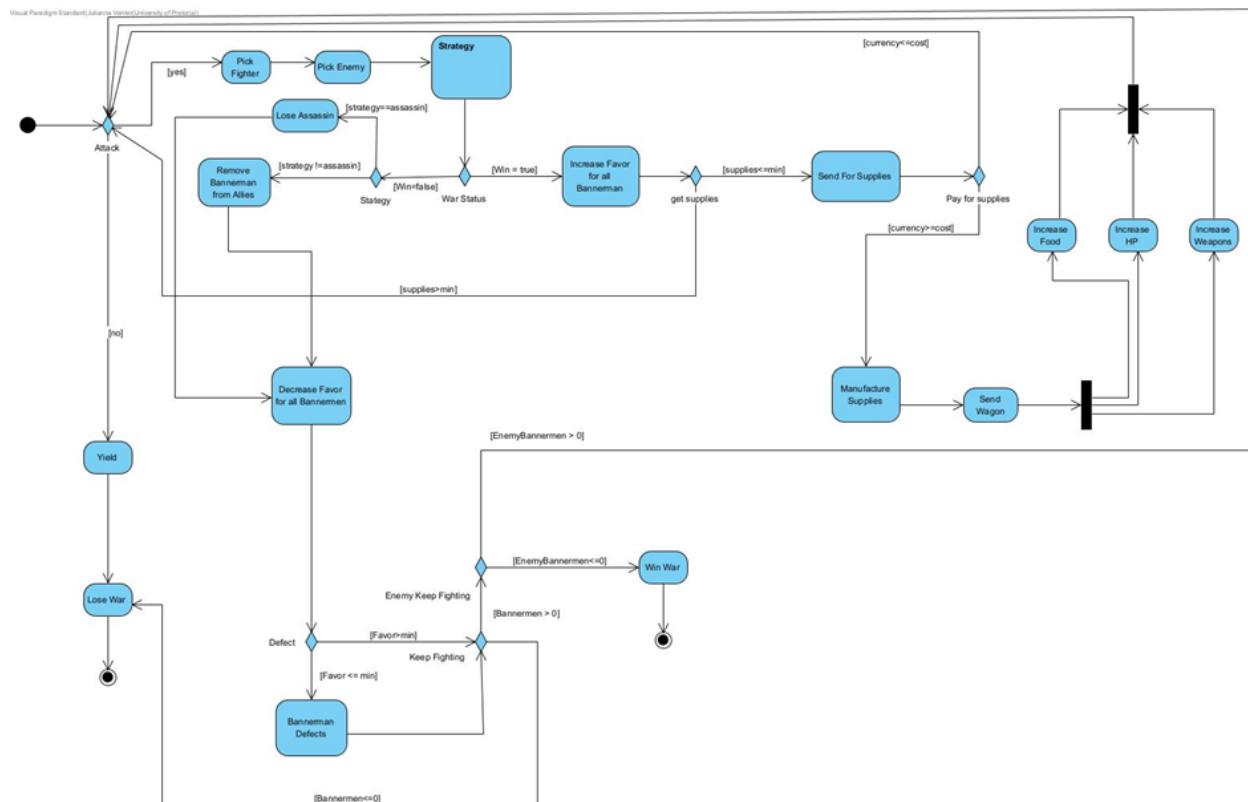
Task 2:

Functional Requirements:

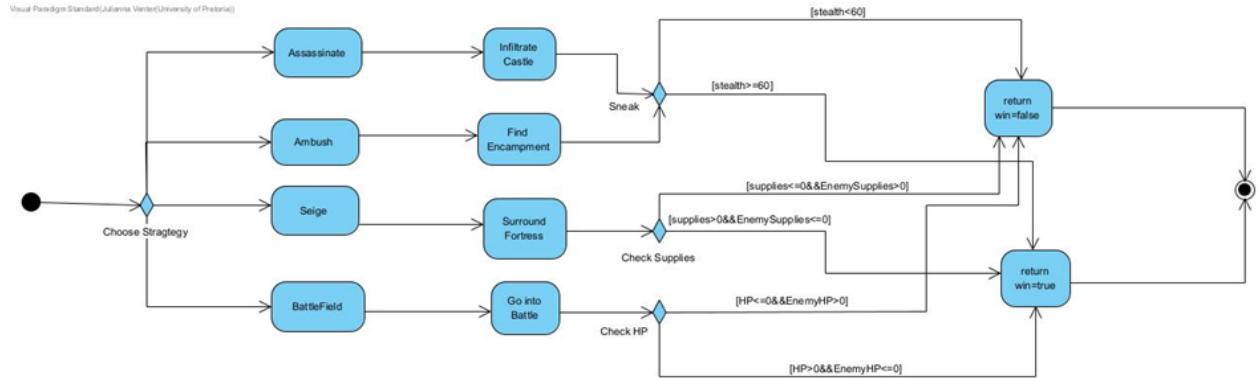
- Two bannermen lists must be created.
- A user must be able to choose to attack.
- If a user doesn't attack they are forced to yield and lose the war.
- If the user attacks they must be able to pick their attacking Bannerman.
- If a user attacks they must be able to pick who they are attacking.
- A user must be able to choose their attack strategy.
- A user must be able to save and restore the state of defected allies.
- The Kingdom must have its own economy in a specific state.
- Food, Medicine and Weaponry must be manufactured.

- Supplies must be transported.
- Health of the Economy must be overseen.
- A user will win or lose their war based on a mix of choices and random chance
- There must be a mechanism in place to check if supplies are needed, and to send wagons if they are.
- There must be a hierarchy system in place to show the dynamic between kingdoms, bannermen and troops.
- The system must be able to iterate through the troops.
- The system must be able to generate different war theaters which affect the outcome of battle.
- A user will win the battle if the enemy bannermen are all dead/defected, or the enemy kingdom has run out of money.
- A user will lose the battle if their bannermen are all dead/defected, or they run out of money.

Activity Diagrams:



Activity diagram for the main functionality from the user's perspective.



Activity diagram for the strategy classes.

Patterns:

1. Memento:

Saving the states of defected allies, and restoring the ones who return.

2. State:

State of the kingdom's economy and upkeep during the war.

3. Factory:

Manufacturing of food, weaponry, and medical supplies.

4. Prototype:

Supply wagons comprising food, weaponry and medical supplies. To be cloned and sent out when a successful troop returns from a battle.

5. Mediator:

Checks if the economy is healthy enough to support further warfare.

6. Observer:

Checks if supplies are needed for troops, sends for a supply wagon if needed.

7. Strategy:

Different types of battle to choose from when engaging the enemy.

8. Composite:

Comprises the Bannermen (allied kingdoms), the troops and commanders of these allies.

9. Iterator:

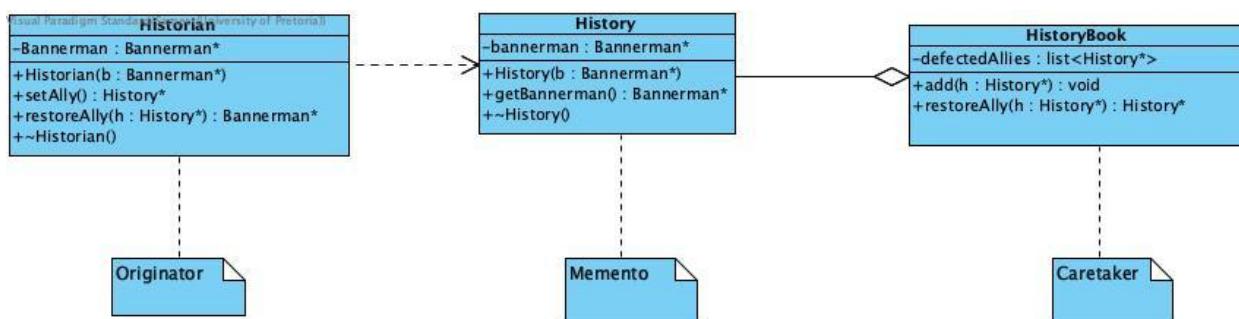
Controls line of command in troops.

10. Decorator:

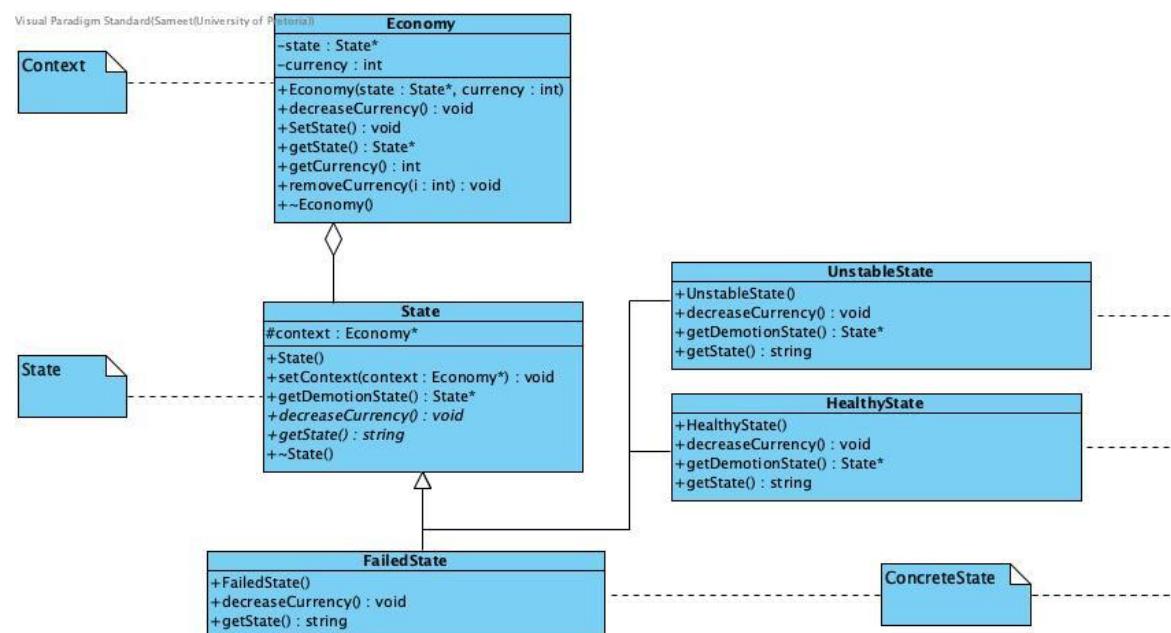
Controls and generates war theater, influencing outcome of battle.

Class diagrams:

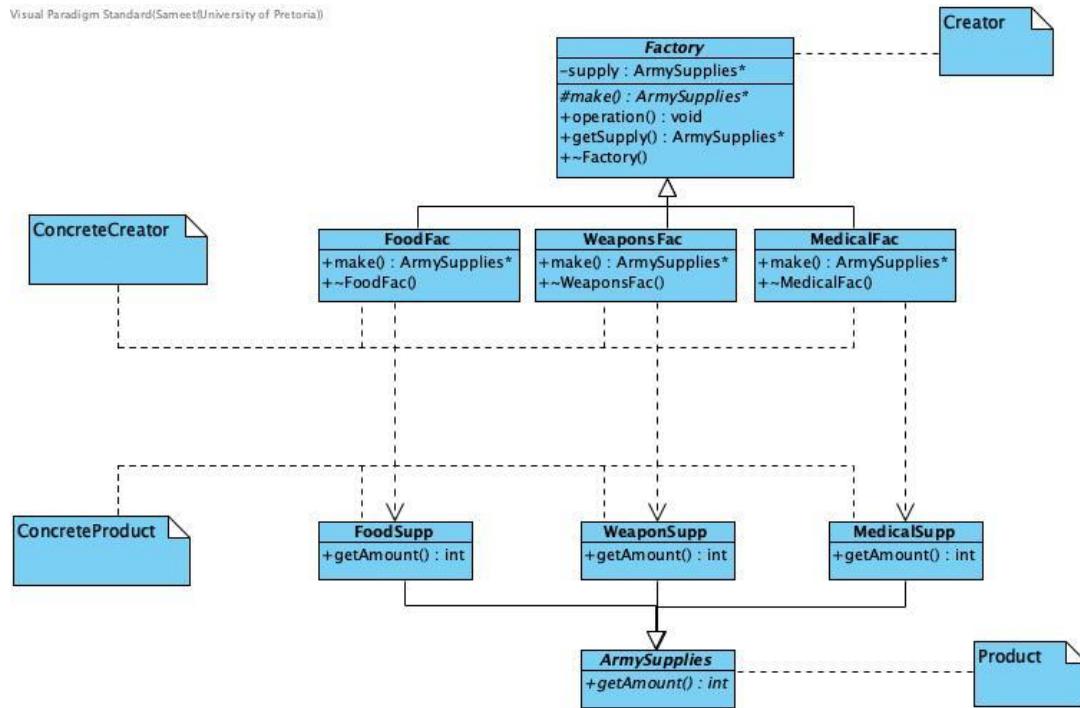
1. Memento:



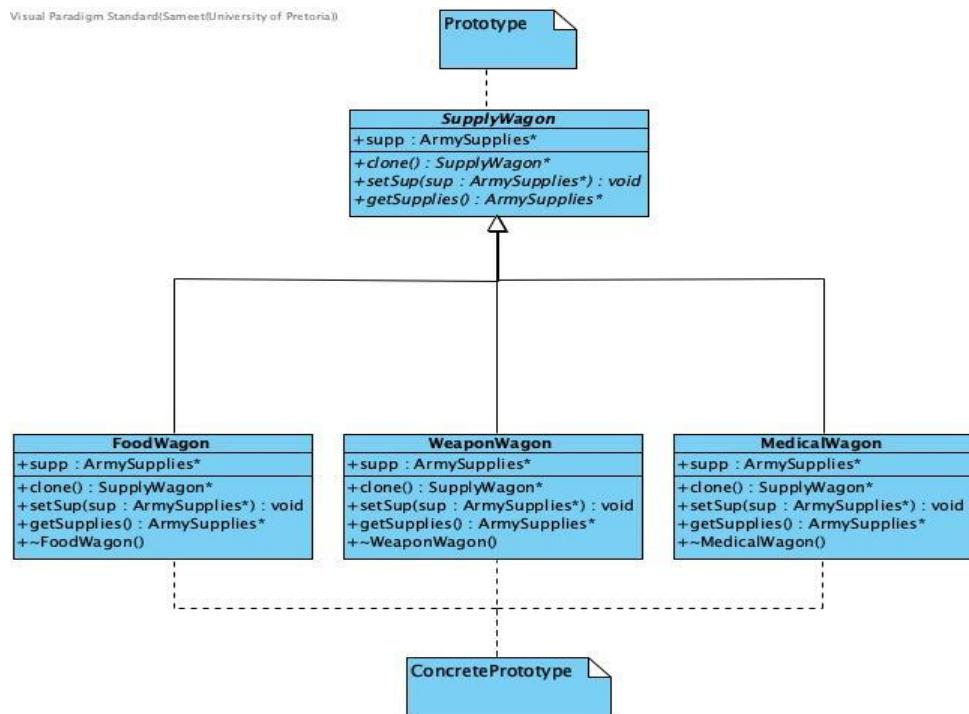
2. State:



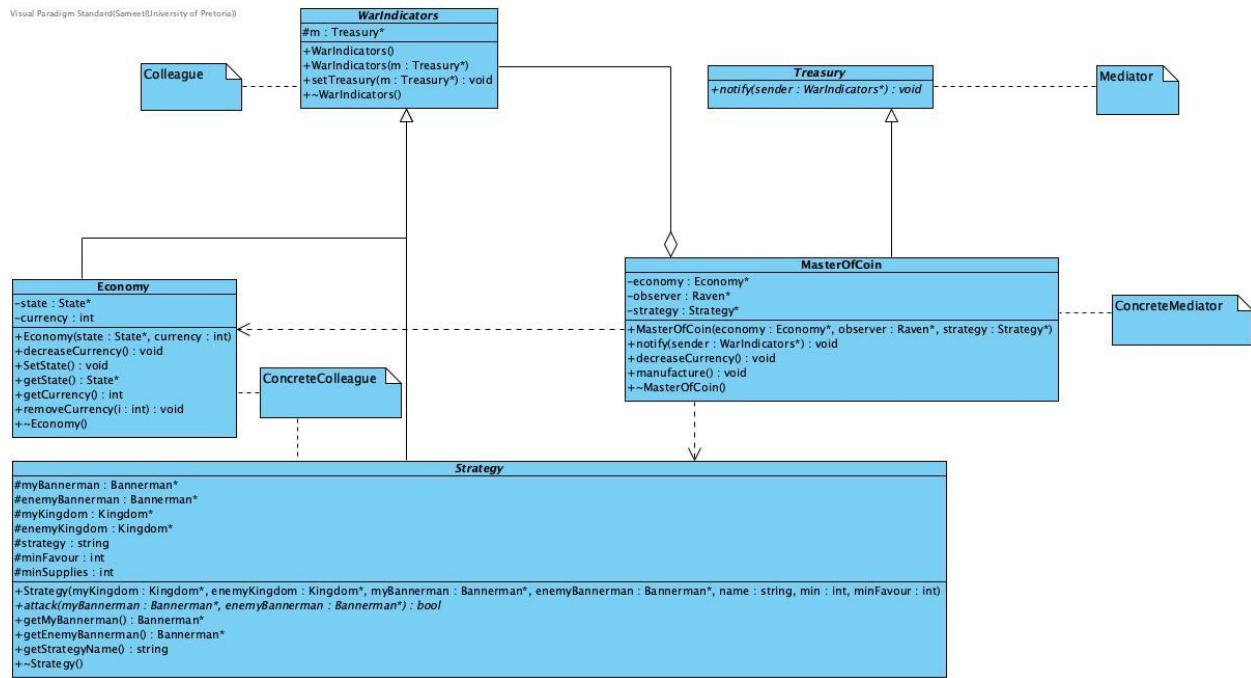
3. Factory:



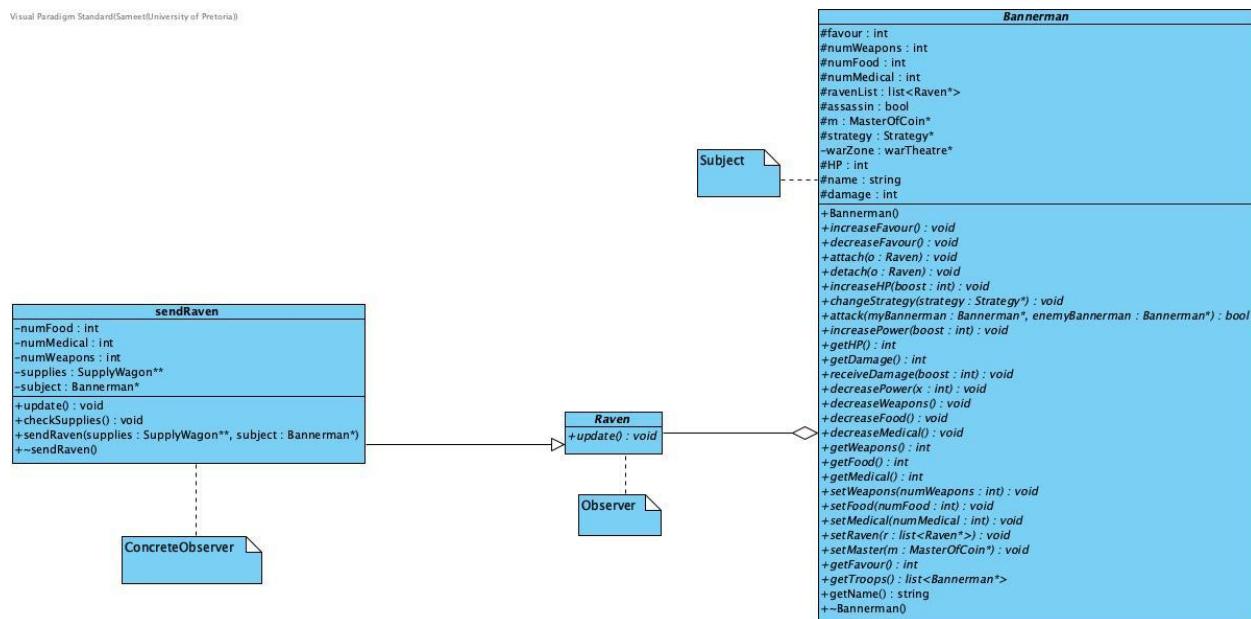
4. Prototype:



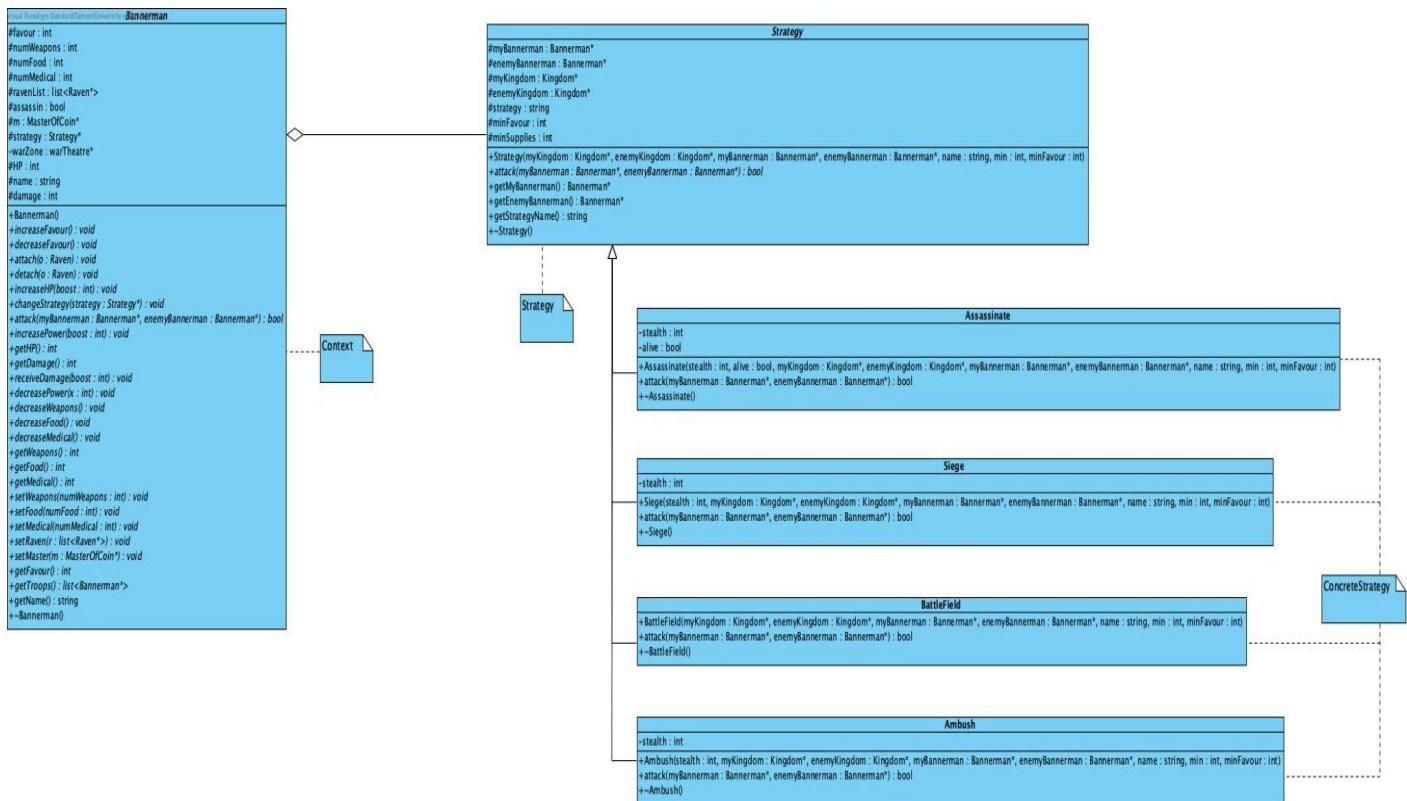
5. Mediator:



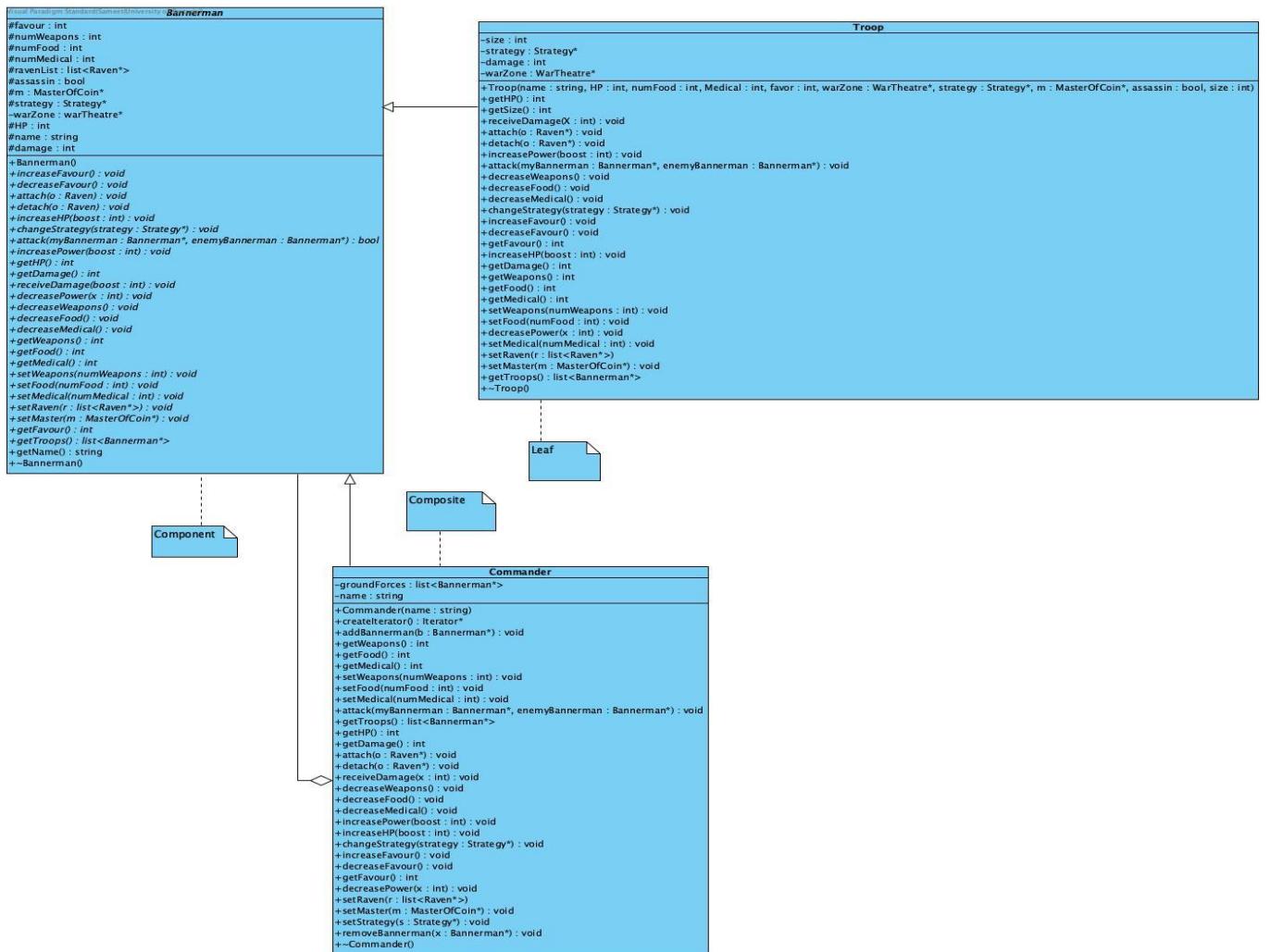
6. Observer:



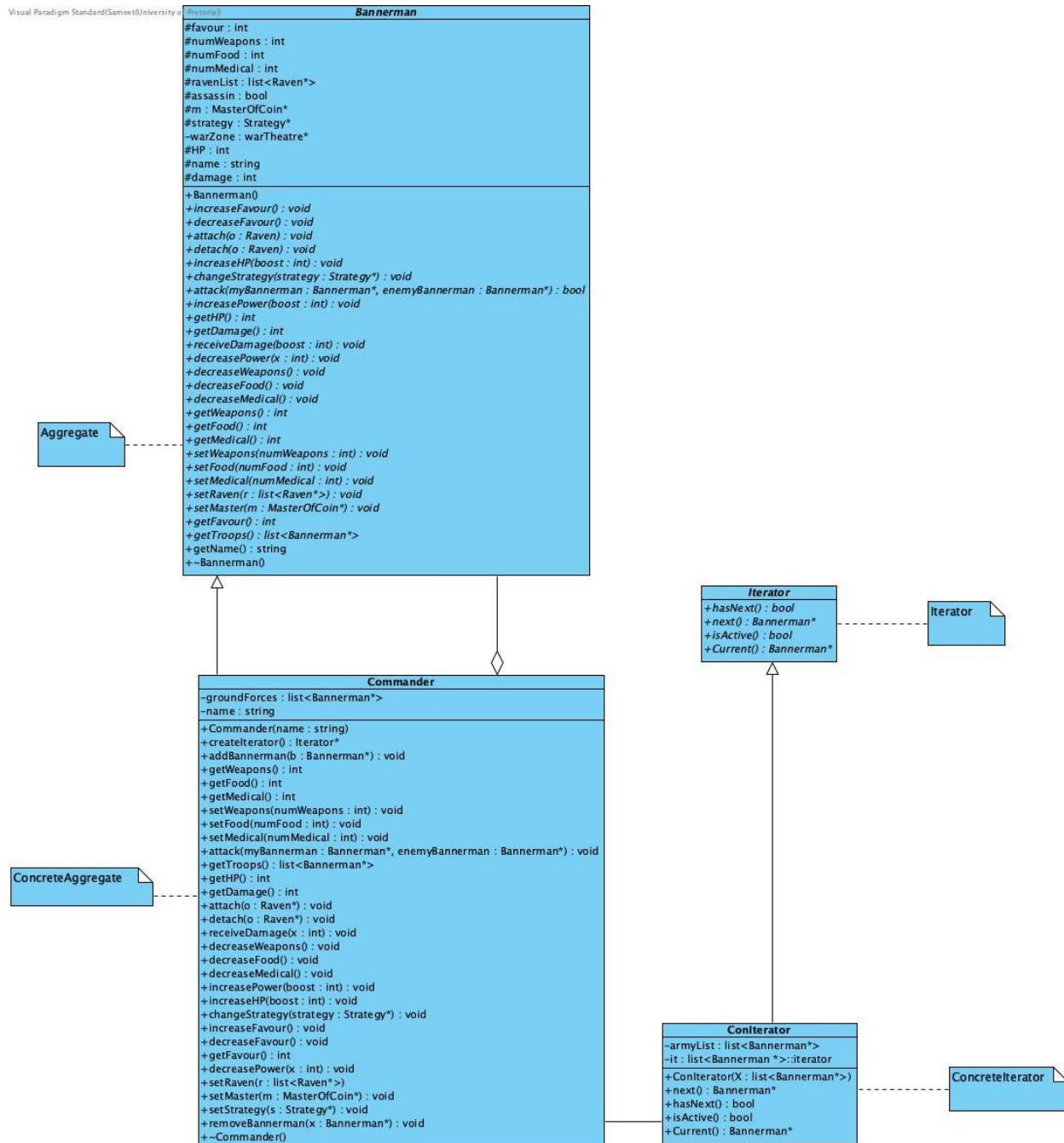
7. Strategy:



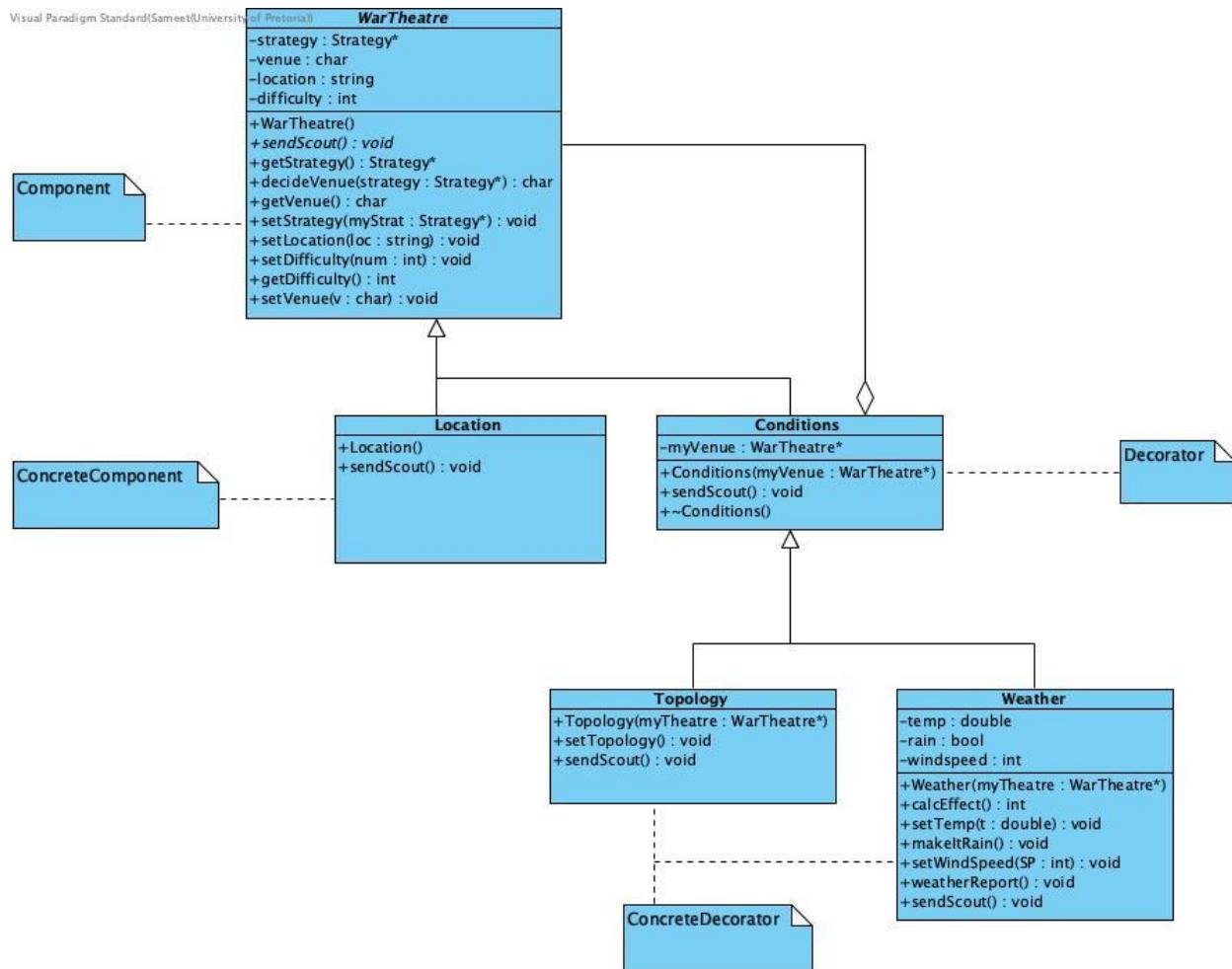
8. Composite:



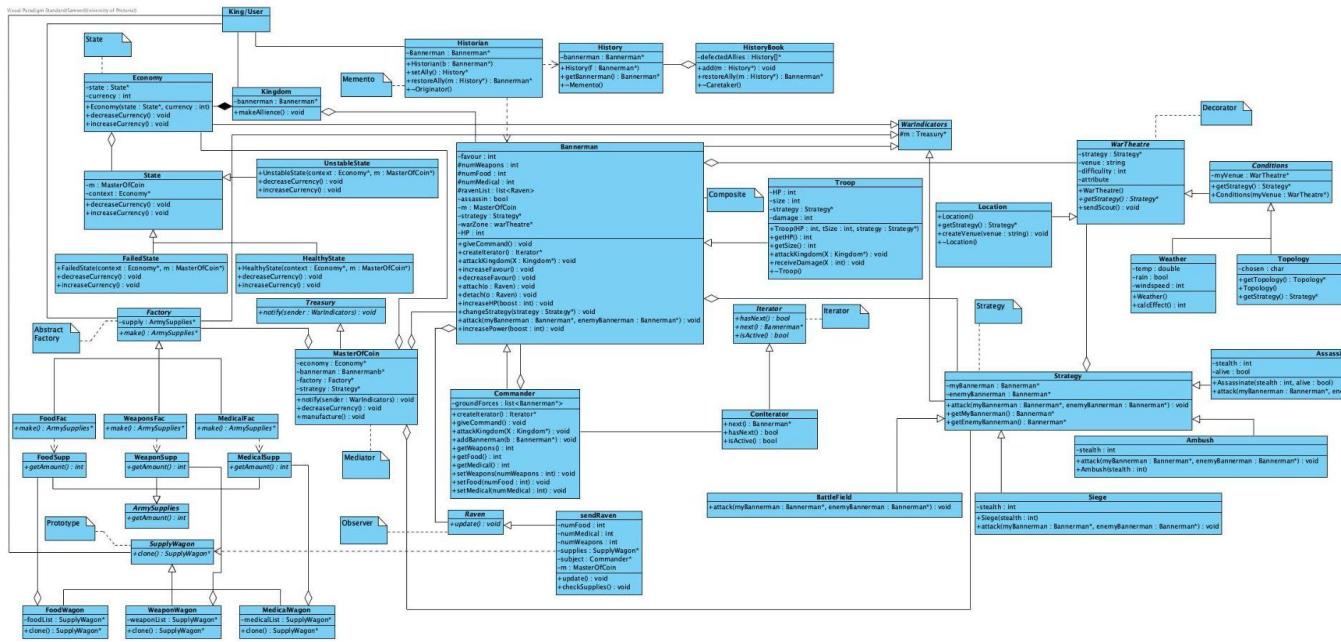
9. Iterator:



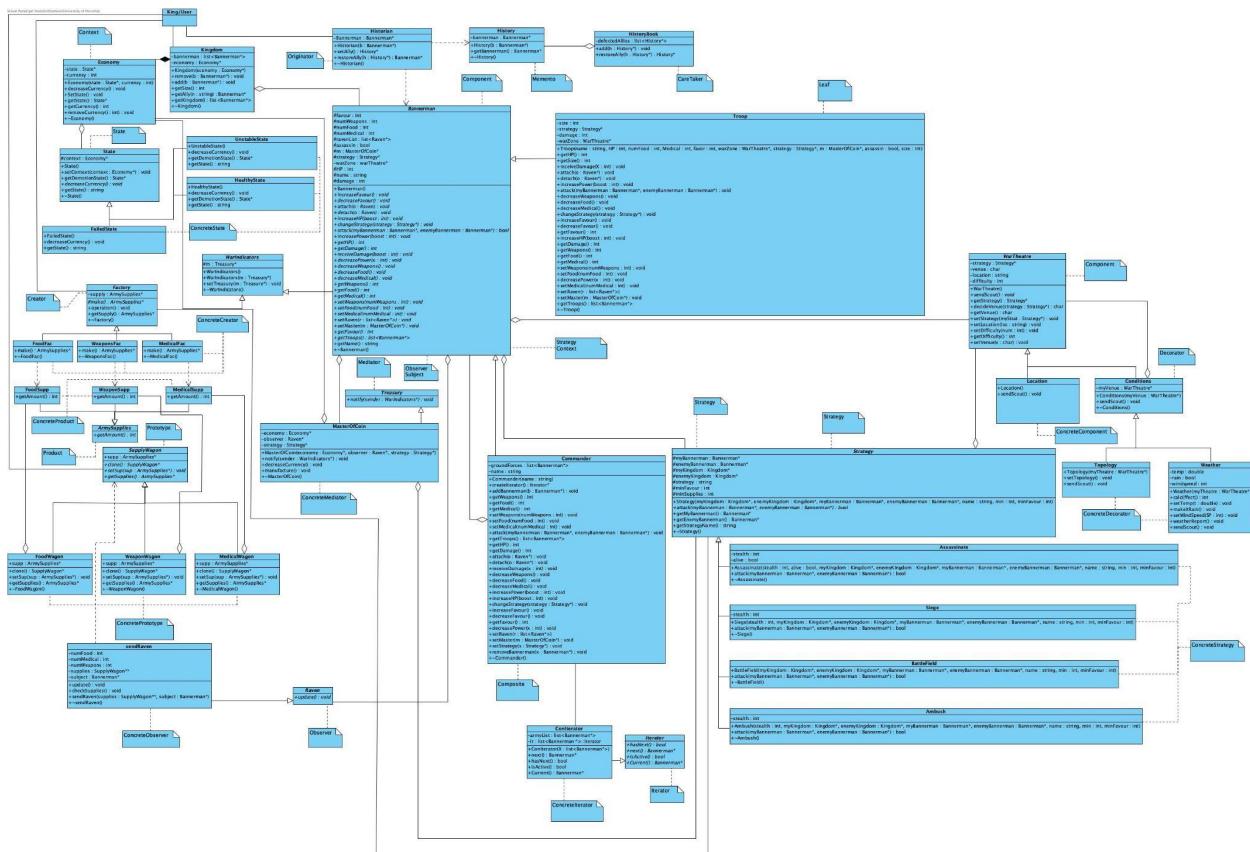
10. Decorator:



Initial Design:

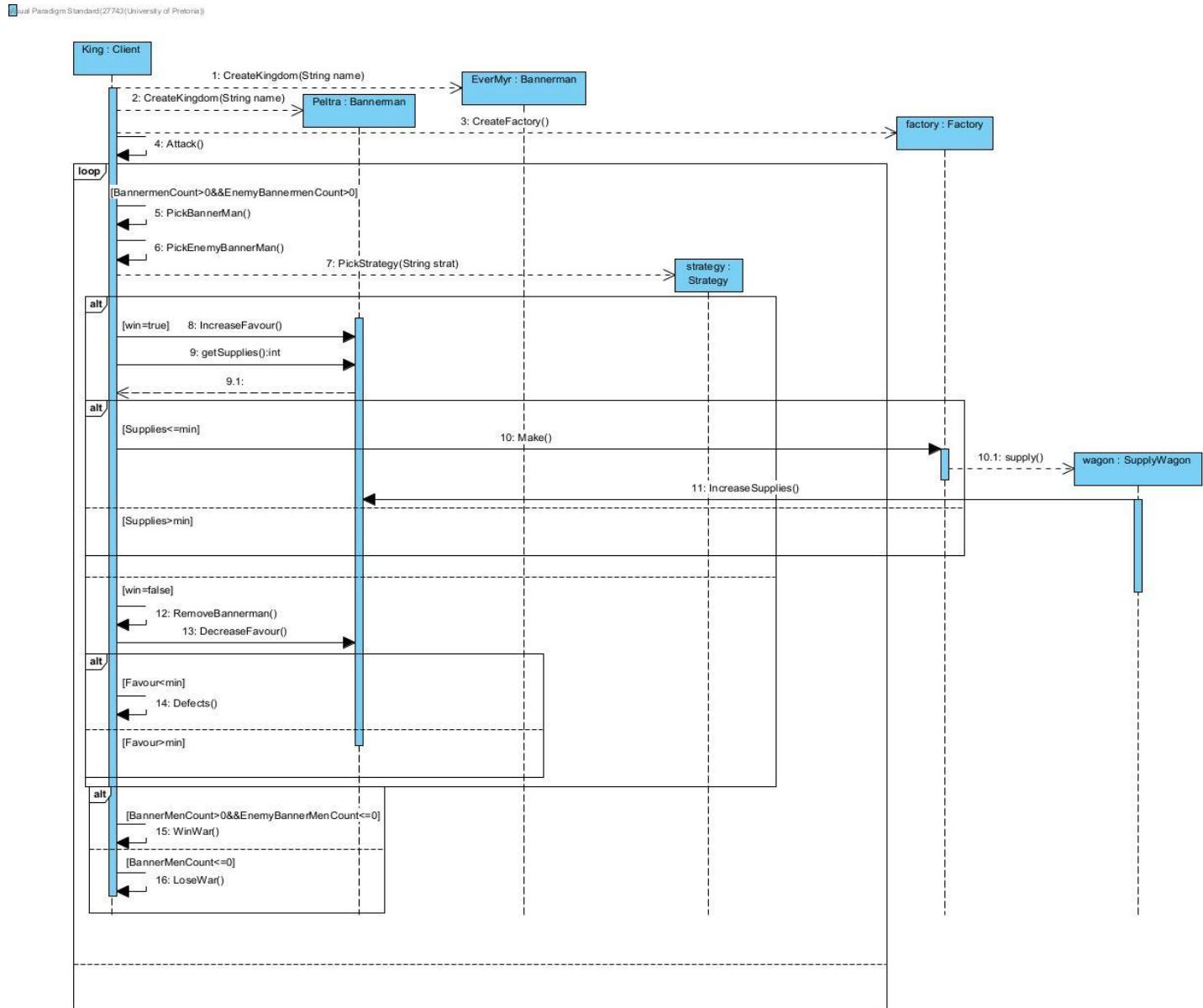


Final Design:

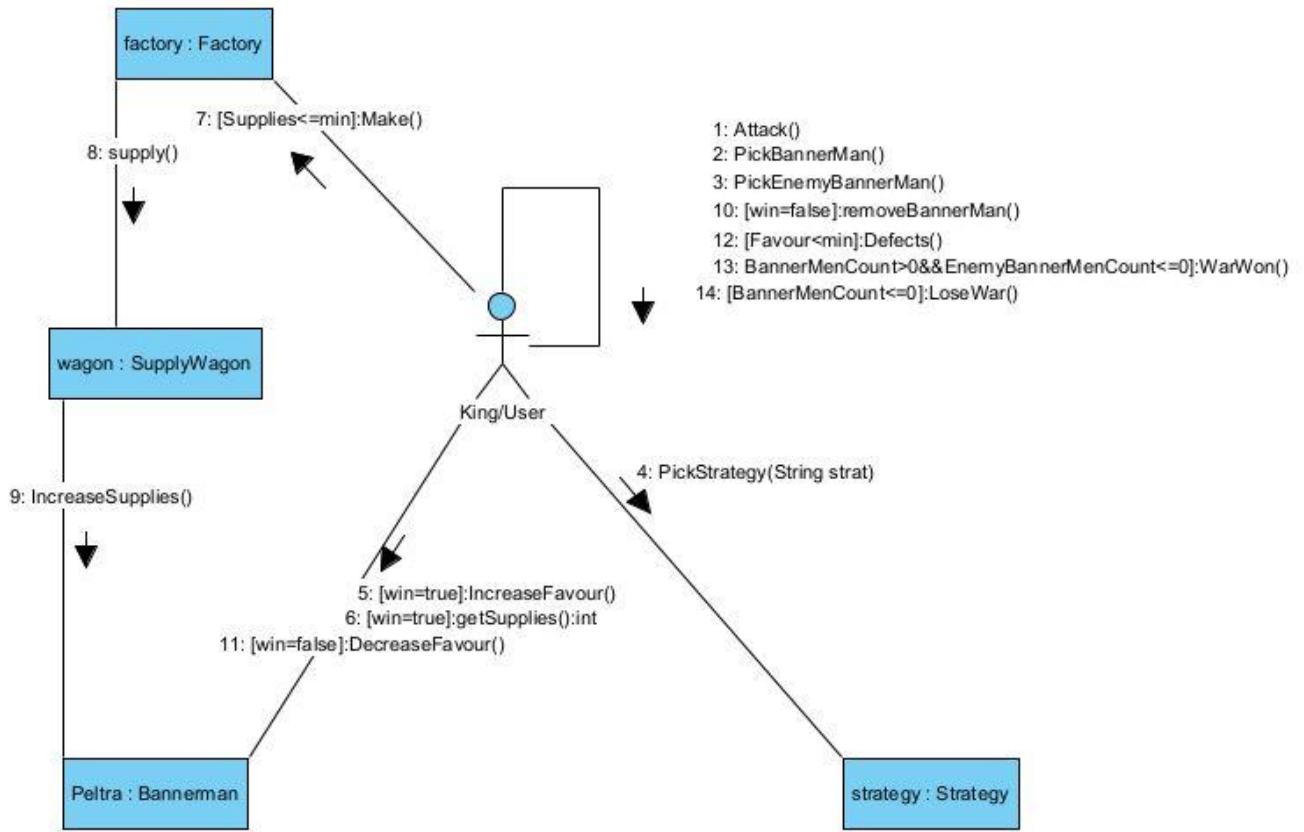


Sequence Diagram:

It should be noted that the sequence and corresponding communication diagrams below are a simplification/summary of what plays out in our main() function. An accurate sequence diagram would result in a diagram too wide and too complex to get a good overview of what our project does. Below you'll find a less detailed, oversimplified sequence diagram that still accurately describes the foundation of our program.

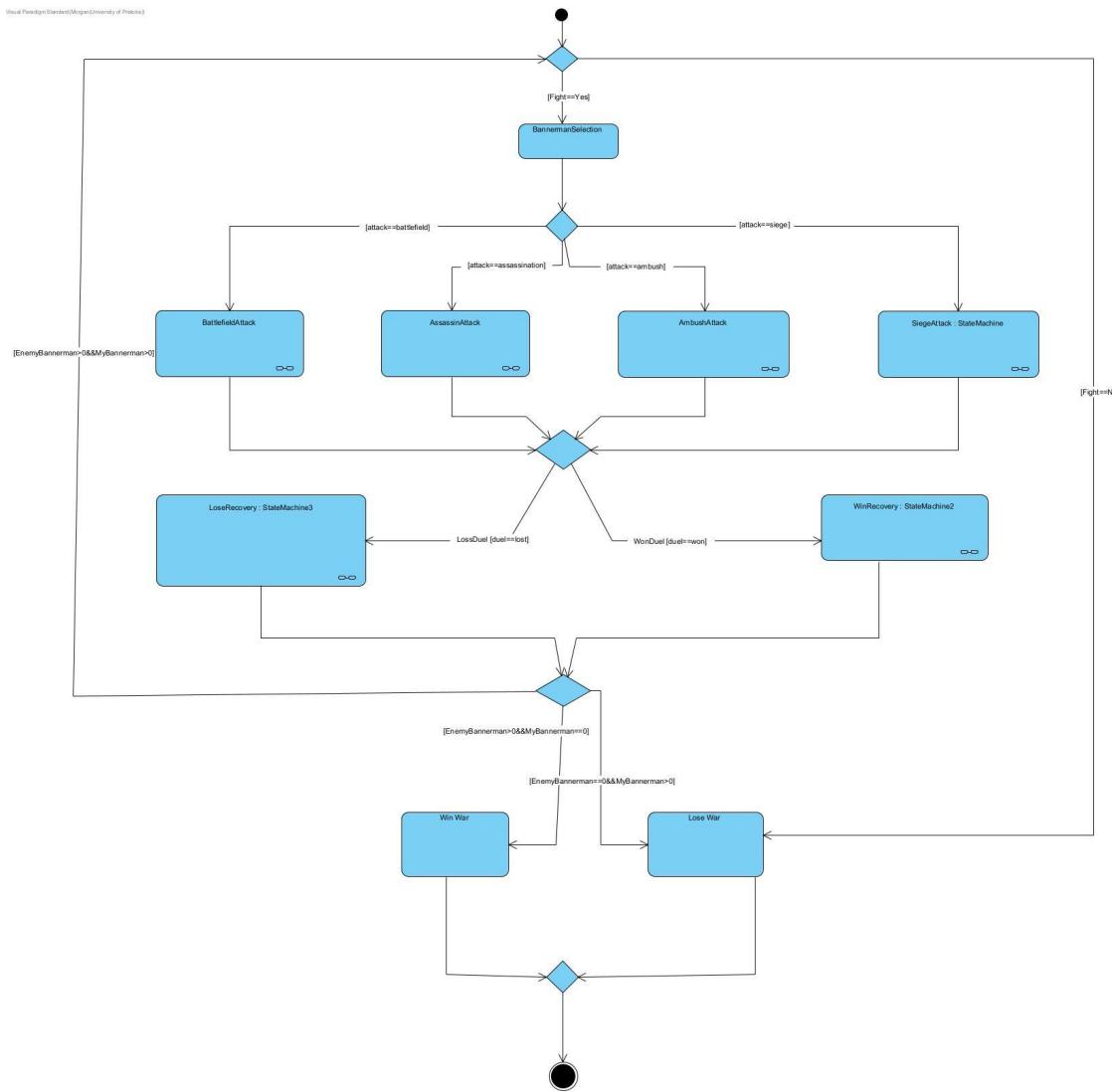


Communication Diagram:

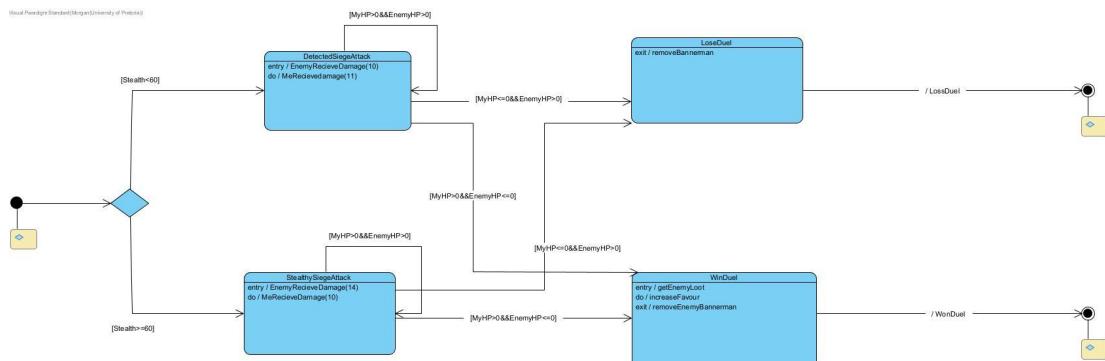


State Diagrams:

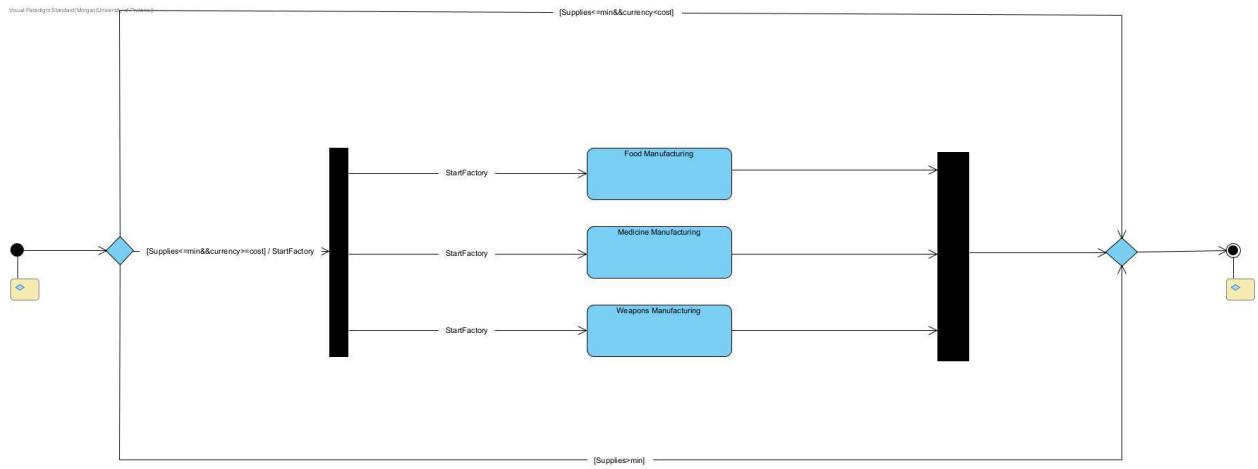
Main State Diagram:



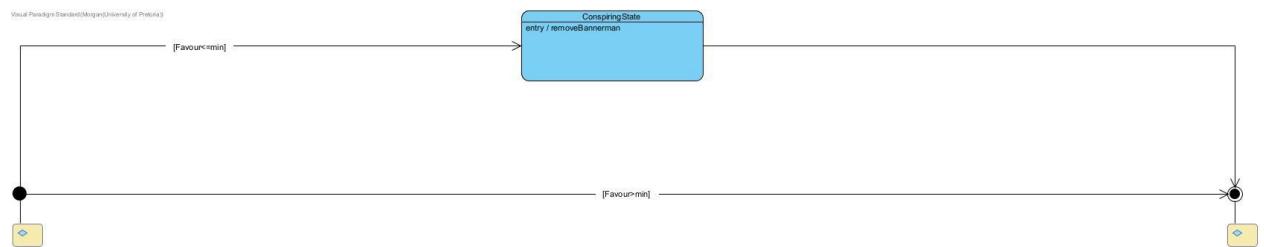
Siege State Diagram:



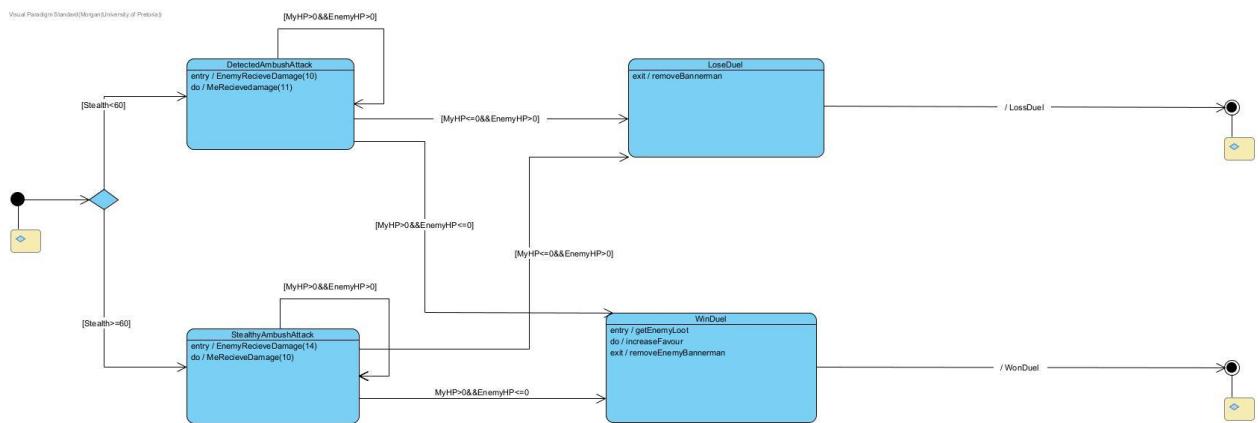
Factory State Diagram:



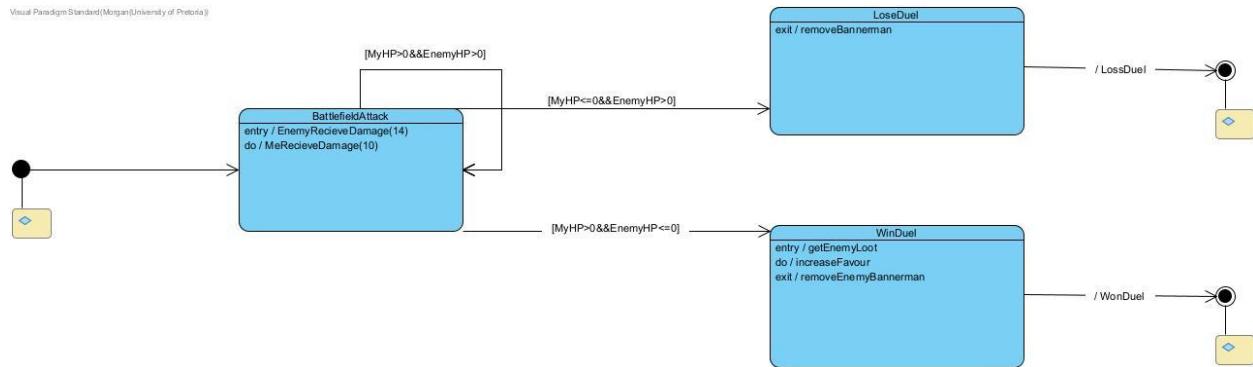
Lose Recovery State Diagram:



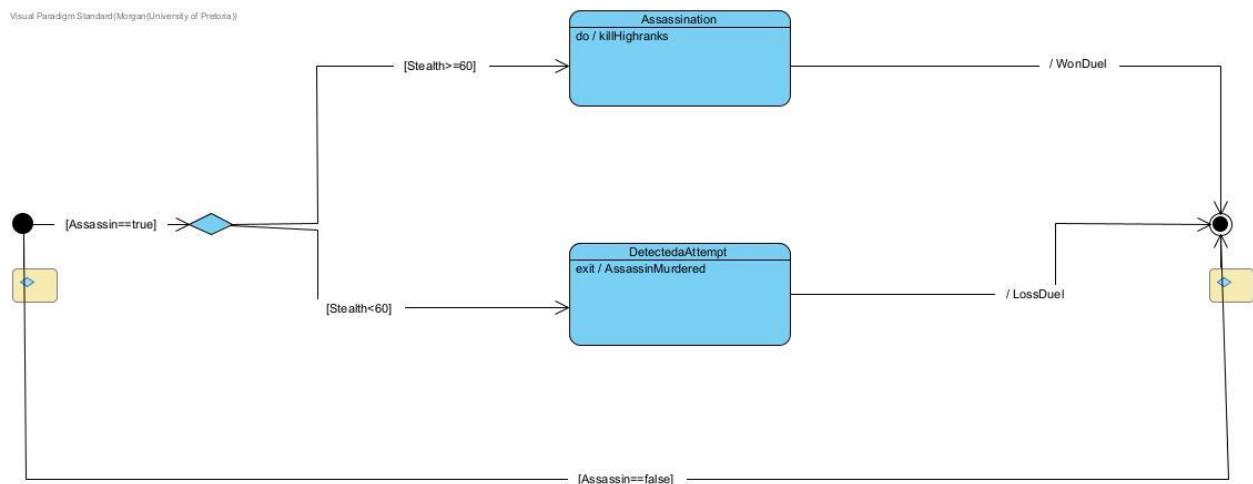
Ambush State Diagram:



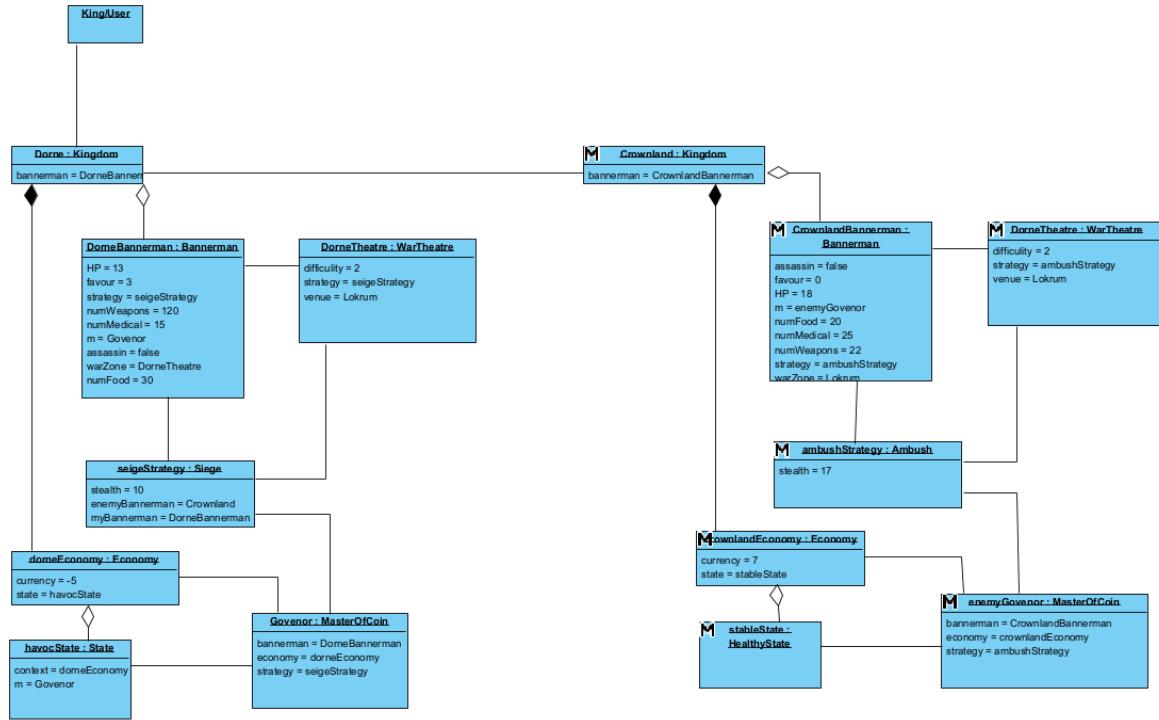
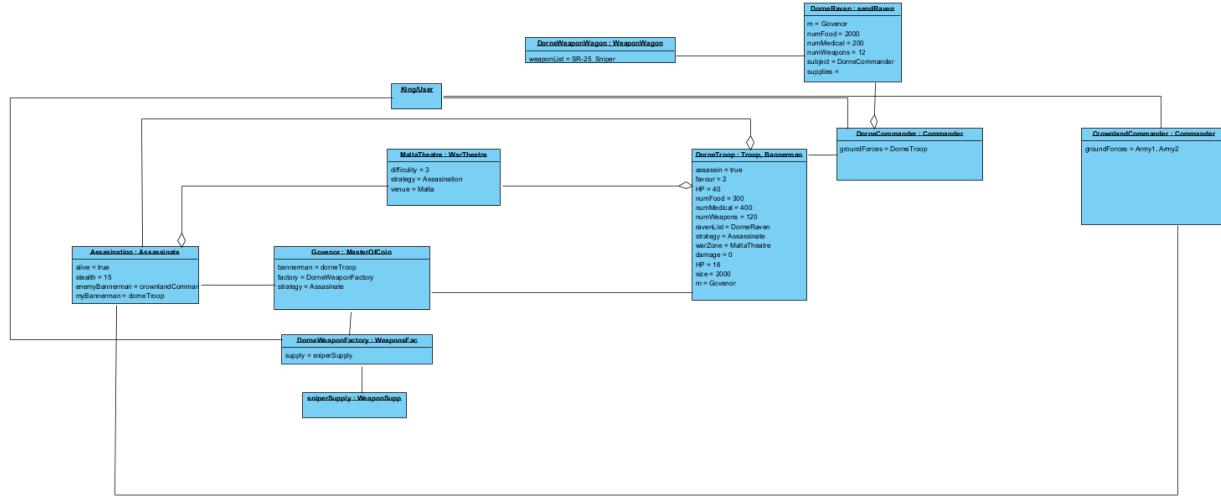
Battlefield State Diagram:



Assassination State Diagram:



Object Diagrams:



Task 3:

Implementation included in files.

Task 4:

Research Report:

The core theme of our war engine revolves around medieval warfare, as it provides more flexibility and creativity to apply the design patterns.

Our war simulation follows the monarch of the kingdom Dura (the user), who must make decisions throughout the war to defeat the United Kingdoms of Preadora. Our kingdom has five bannermen (Countries in Country groups allied to ours), who must individually fight five other bannermen of the enemy kingdom in order to take over their land.

Our simulation uses various elements to influence the outcome of the war. Each bannerman within the country is influenced by favor, which is increased or decreased depending on the outcome of a battle. This favor is then used to decide whether the bannerman defects from the allies and joins the enemies, or remains faithful to the allies. This bannerman may later decide to return to the alliance. The success of the war depends on how many bannermen remain standing and the kingdom's economy (money for supplies to support troops, the state of the civil population, etc).

In warfare, it is evident that hierarchy is crucial. Certain decisions are taken at different authority levels. We embodied this by implementing our own pyramid of command: the monarch (user) is in command of every bannerman in the kingdom, these bannermen each have troops that are assigned a commander.

The battles themselves are influenced by the war theaters, which have their own advantages and disadvantages. These include the landscape, climate & weather, time of day, length of the battle in that specific theater, and most importantly, the use of these facts to plan the attack or defense.

In many notable battles in history, weather has played a key role in the outcome of that battle. Conditions such as heavy snowfall easily stopped soldiers in their tracks (i.e. Napoleon and Hitler's attempt to invade Russia) or heavy winds working in the favor of others (Winds carrying flames over the Spanish Armada fleet) (*Thomas, D.*)

Our war simulator uses topology and weather to similarly determine if any advantages and disadvantages will be added to the battle.

“...the scale of war a thousand years ago appears small, but the share of resources that was consumed by war was enormous. This consumption included not only the expenses of

waging war but, almost invariably, the deliberate destruction of economic assets that accompanied invasions, and sometimes retreats as well.” - Jurgen Brauer & Hubert van Tuyll

War has a great impact on the economic state of a kingdom. There is great cost in keeping - not only the people of the kingdom - but the soldiers, fed and armed. Moreover, medical aid and housing supplies add to these expenses. (Brauer, J., et al.)

Our simulation takes this into consideration when planning attacks. War is quickly lost if no supplies are being sent out to troops and food runs scarce within the kingdom. This can make the bannermen lose favor, which will result in the loss of more and more allies until the war is lost. Similarly, if the costs have been well regulated and supplies sent out strategically, the victory is yours.

Beyond battles in fields, war is carried out in different events. Strategic decisions and actions are taken with the pure motivation of the war. These events include castle and fortress sieges, assassinations of influential parties and troop campsite ambushes.
(Swansea University)

Our simulation makes use of these strategies to give the user more flexibility in decision-making to gain the best outcome in the war. Different strategies have different effects on troops, supplies, likelihood of success and bannerman favor. For instance: sending an assassination spares your troops and supplies, but is a lot riskier since the chances of success are lower based on stealth.

Through these elements and entities our user has many factors to keep in mind throughout the stages of the war and, consequently, their discernment will determine the victorious kingdom.

References:

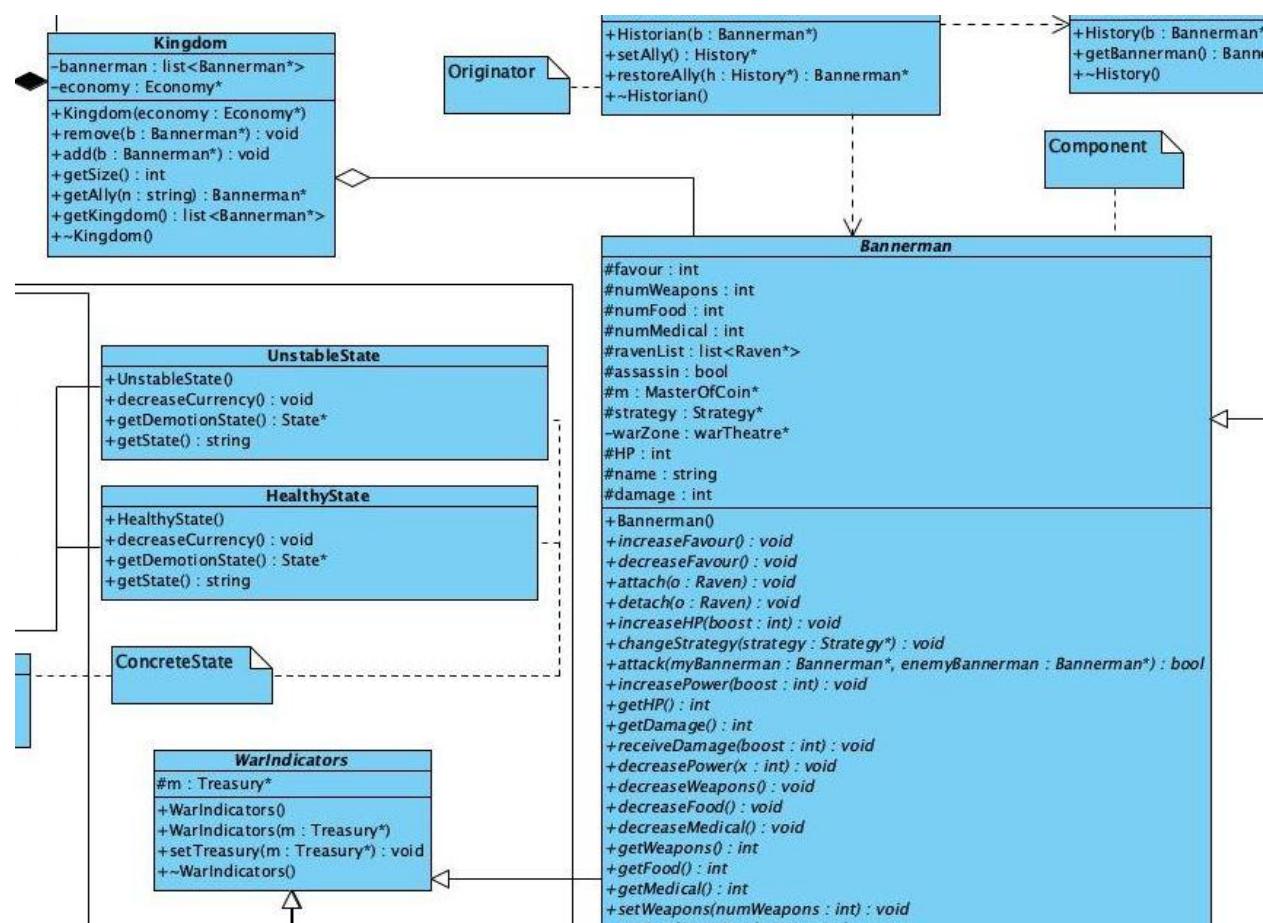
1. Brauer, J., van Tuyl, H.(2008) *Castles, Battles, and Bombs: How Economics Explains Military History*. Available at:<https://press.uchicago.edu/Misc/Chicago/071633.html> (Accessed: 14/10/2022)
2. Swansea University. *Warfare in Western Europe in the Central Middle Ages*. Available at:
<https://www.swansea.ac.uk/history/history-study-guides/warfare-in-western-europe-in-the-central-middle-ages/>. (Accessed: 14/10/2022)
3. Thomas, D. (2022). *7 Times Weather Affected War* . Available at:
<https://www.foxweather.com/learn/7-times-weather-affected-war>. (Accessed: 14/10/2022)

Design Report:

All functional requirements will be met as follows:

- Two bannermen lists must be created:

One list representing the bannermen on our side and the other list representing the enemy bannermen are to be initialized. These lists are contained in Kingdom objects and are initialized in the first phase of war, in the populateVectors() subroutine - which can be found in the main. The UML snippet below shows the relationship between each Kingdom and Their Bannermen.



- A user must be able to choose to attack:

A user must be able to inflict damage on an enemy kingdom and to decide on their method of attack. This is achieved by using the Strategy design pattern, which is explained in detail later.

- If a user doesn't attack they are forced to yield and lose the war:

If a kingdom yields, the enemy wins the war and colonizes the kingdom- this is implemented in the main.

- If the user attacks they must be able to pick their attacking Bannerman:

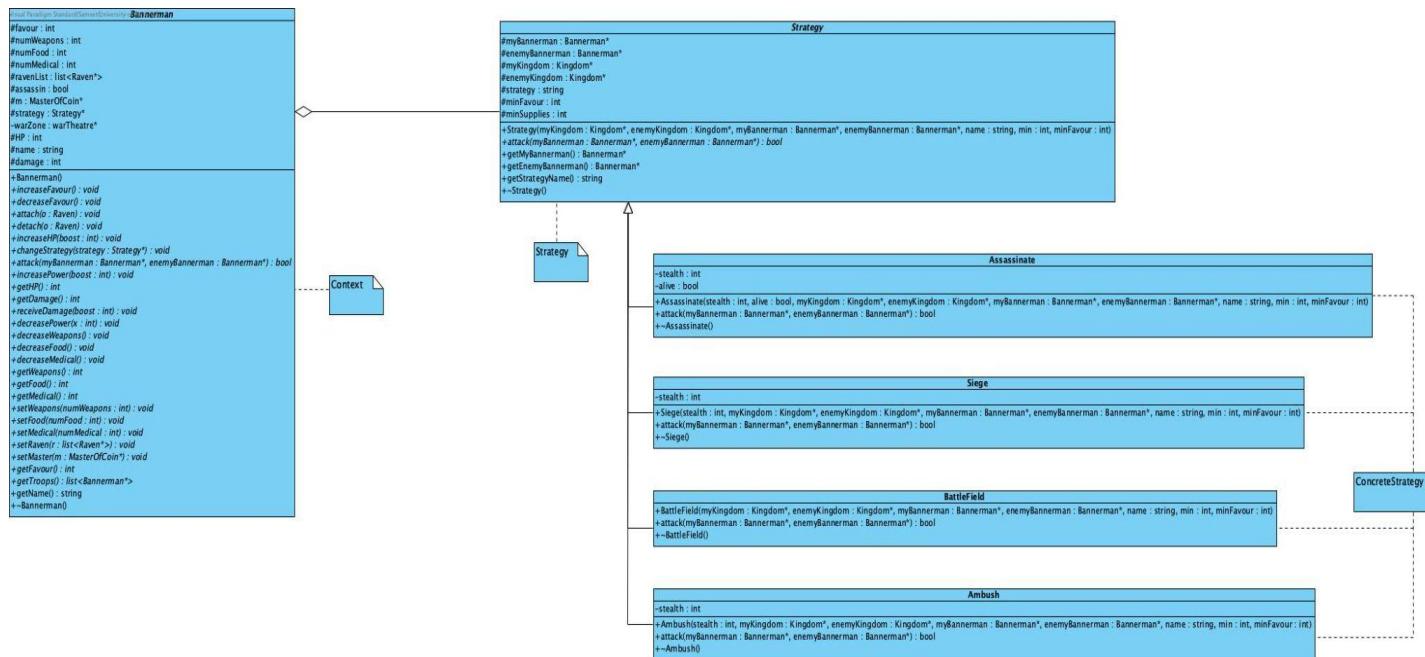
The user must be able to decide which one of their bannermen to use in a battle. The attacking bannerman is to be passed in as the third parameter in the war strategy constructor.

- If a user attacks they must be able to pick who they are attacking:

The user must be able to decide on the bannerman to attack within the enemy kingdom. The kingdom being attacked is to be passed in as the second parameter in the war strategy constructor, the bannerman belonging to this kingdom is to be passed in as the fourth parameter in the strategy constructor.

- A user must be able to choose their attack strategy:

After a User has selected which of the enemy kingdoms to attack. This option will be provided in the main. They are then prompted to select which strategy to use to attack. To accomplish this requirement we are making use of the Strategy Design Pattern as seen below:



Each option for what strategy to use will be within each of the derived classes; Battlefield

which represents fighting on an open field, Siege which represents our army attempting a siege on theirs, Ambush which implies our army ambushes theirs, Assassinate which implies we send one of our own assassins to kill the leader of the Enemy Bannerman. Each of these derived classes implements their individual strategy algorithm named ‘attack’ which takes in a pointer to both our bannerman and the enemy’s.

- A user will win or lose their war based on a mix of choices and random chance:

Combat between bannermen is an exchange of blows to the HP of each bannerman. Each bannerman has their own damage (implemented in the derived Troop Class) which is the default value to be deducted from the Enemy HP during combat. The first Bannerman to have HP less than or equal to 0 loses the battle, and the Kingdom loses that bannerman. The first Kingdom to run out of Bannermen loses. The medical and food supplies of each bannerman greatly affect their chances of winning. If your Medical and Food supplies are below a certain limit the damage to your HP will be double the default and the same rule goes for the Enemy Bannerman. Therefore, it is essential to have Medical and Food supplies on the ready. Note that the longer combat continues, the more supplies both sides lose.



The selection of which strategy to use against your opponent is crucial as it will affect

whether or not you win or lose a battle. The outcome of each battle is determined in each of the attack function implementations found in each of the derived classes. Each attack implementation goes as follows:

Assassinate: Each Bannerman has their own assassin which they can send out to kill the commander of the enemy bannerman. The success of his mission depends on his stealth variable. If successful, the commander is dead and the bannerman group will be dismantled instantly. No need for Combat.

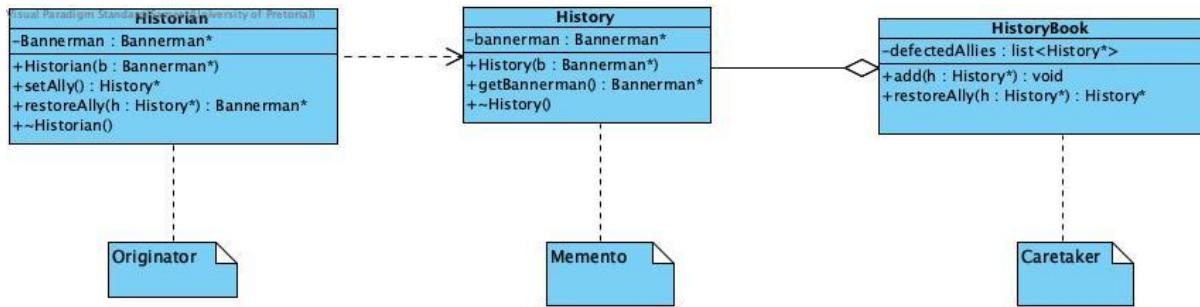
Siege: Our troops surround the enemy bannermans fortress and provided our bannermans stealth is high enough, their Food, Medical and Weapon supplies are greatly reduced. However if our stealth is not high enough then the siege is unsuccessful. Regardless of whether the siege was successful or not, combat still follows, followed by a victory or defeat.

Ambush: Depending on the stealth of your bannerman, you will have caught the other bannerman unprepared so your attacking power(damage) will be higher than normal and theirs will be lower than normal however if your stealth is too low your ambush is detected and will result in no power boost. Combat persists and is followed by a victory or defeat.

BattleField: With this strategy there is no advantage; it is just a typical battle on an open field where the only dependent aspect of the battle is attacking power and HP of each bannerman.

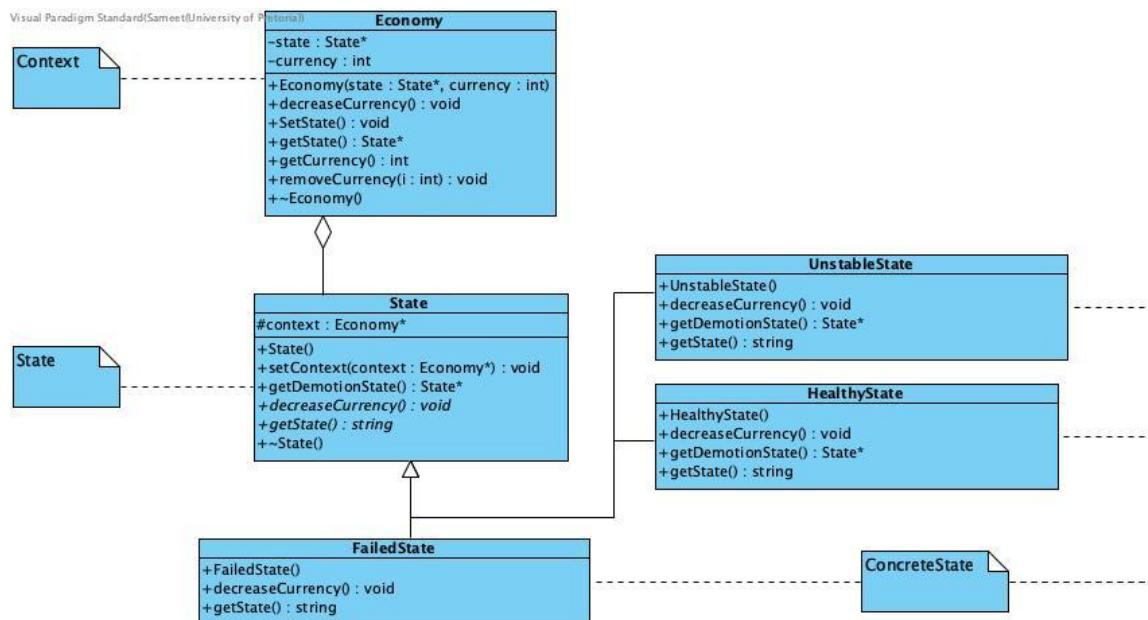
- A user must be able to save and restore the state of defected allies:

This is done via the Memento pattern. After each battle, If a bannerman's favor for the user is too low they defect, however later if their favor for the enemy is too low they can come back to your side, being restored. The **Memento Design Pattern** sees this accomplished. Saving the Bannerman and restoring him later. Below is the UML diagram:



- The Kingdom must have its own economy in a specific state:

Each Kingdom has a specific state. Failed State, Indicating we've run out of money, resulting in an instant loss of the game. Unstable state indicating we're on our way to losing all of our money if we don't get some in soon. Healthy State indicates we still have most of our money left for use. To acquire this feature we implement the **State Design Pattern**. To represent the state of our Kingdom's economy during the war. Each of the different concrete states implement a function which decreases our Kingdom's currency, which occurs whenever we stock up on supplies. Below is the UML diagram:

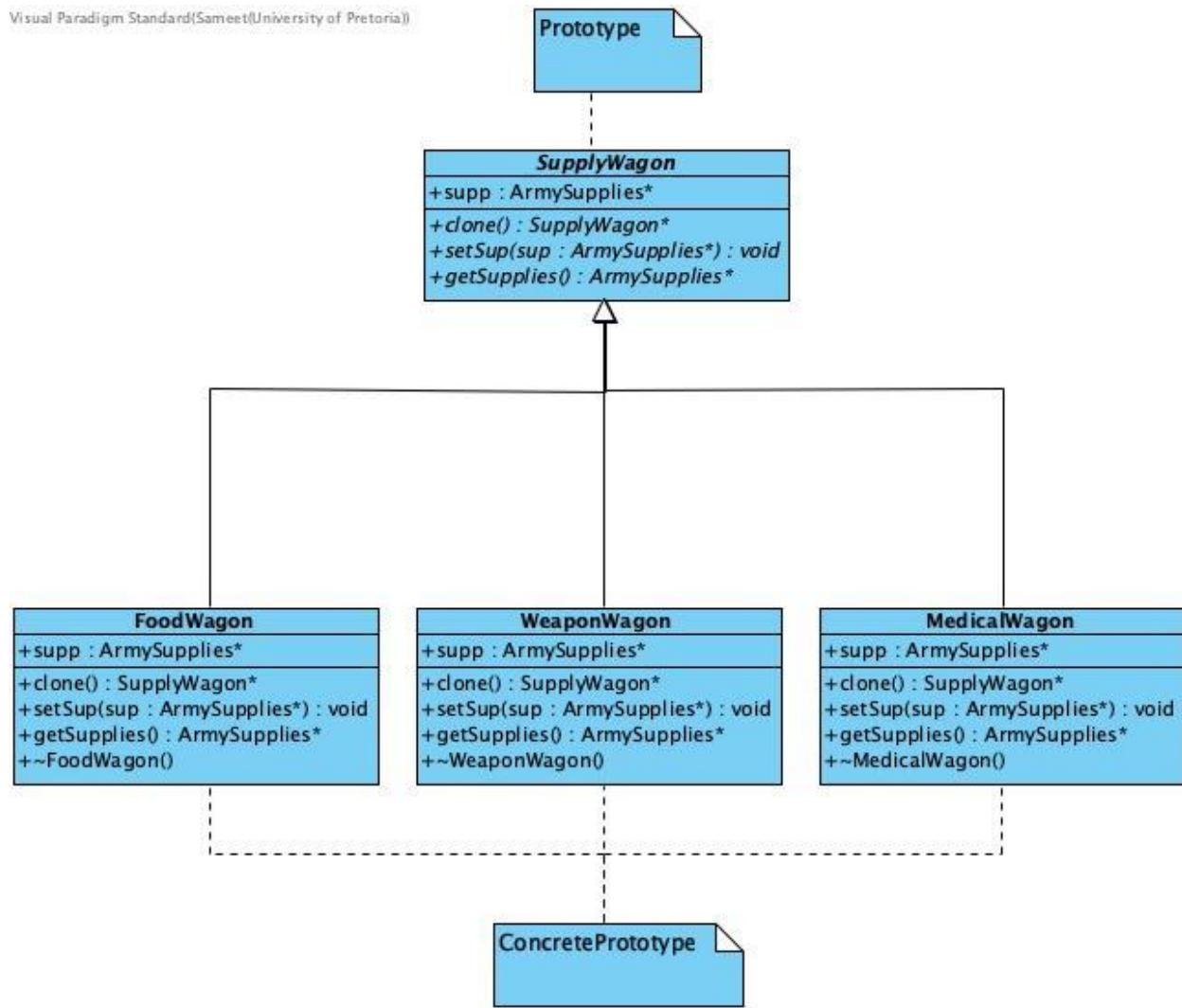


- **Food, Medicine and Weaponry must be manufactured:**

There must be a system in place to allow us to manufacture supplies for our troops. To do this we make use of the **Factory Design Pattern**. We have three different Factories to manufacture Weapons, Medicine and Food. These will be our concrete Factory classes and our Product will be supplies, with the concrete products being specific supplies in food, medicine, weapons. `Make()` is the typical factoryMethod. Below is the UML diagram:

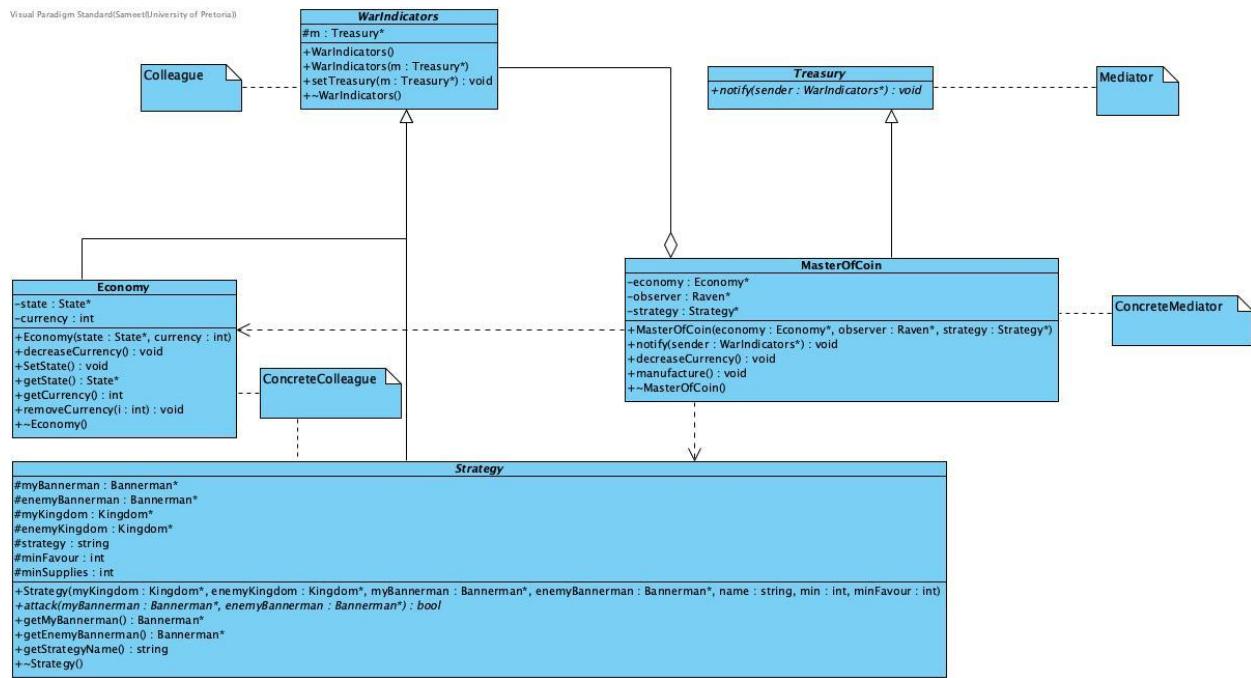
- **Supplies must be transported:**

We need a way to get our supplies to our troops so we use wagons. We have 3 distinct wagons for the three distinct types of supplies, weaponWagon, FoodWagon and MedicalWagon. Which serve as our concrete prototypes. We are using the **Prototype Design Pattern** for this feature to clone the wagons we need whenever we need them. Below is the UML diagram:



- **Management of the Economy funds be overseen:**

We need a way to make sure we are able to have communication occur between our economy, troops and factories. To do this we will make use of the **Mediator Design Pattern**. The Mediator participant is the Treasury. If notify() is called and the one calling it is our Strategy class, we decrease the currency of our Economy, if notify() is called by Economy, then we signal to the Observer class (Raven) to call for a manufacture of supplies. Below is the UML diagram:

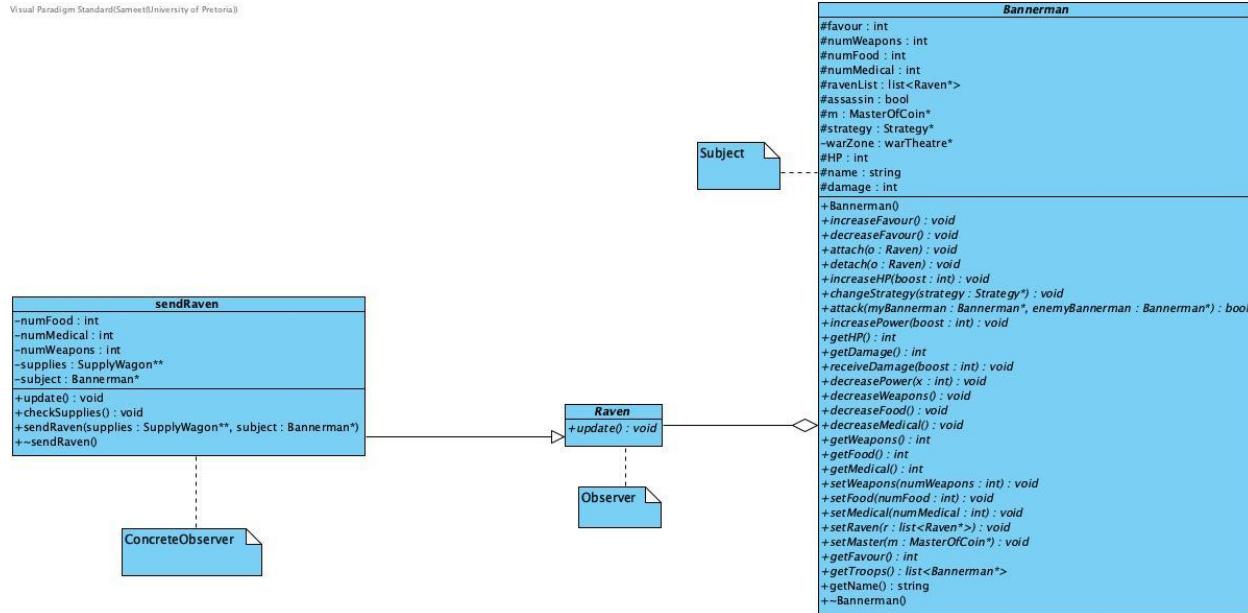


- There must be a mechanism in place to check if supplies are needed, and to send wagons if they are:

As mentioned earlier, in the Mediator, when notify() is called by Economy, then a signal is sent to Observer class to clone the appropriate wagons and send them to the troops.

We use the **Observer Design Pattern**.

Visual Paradigm Standard(Sameetl(University of Pretoria))



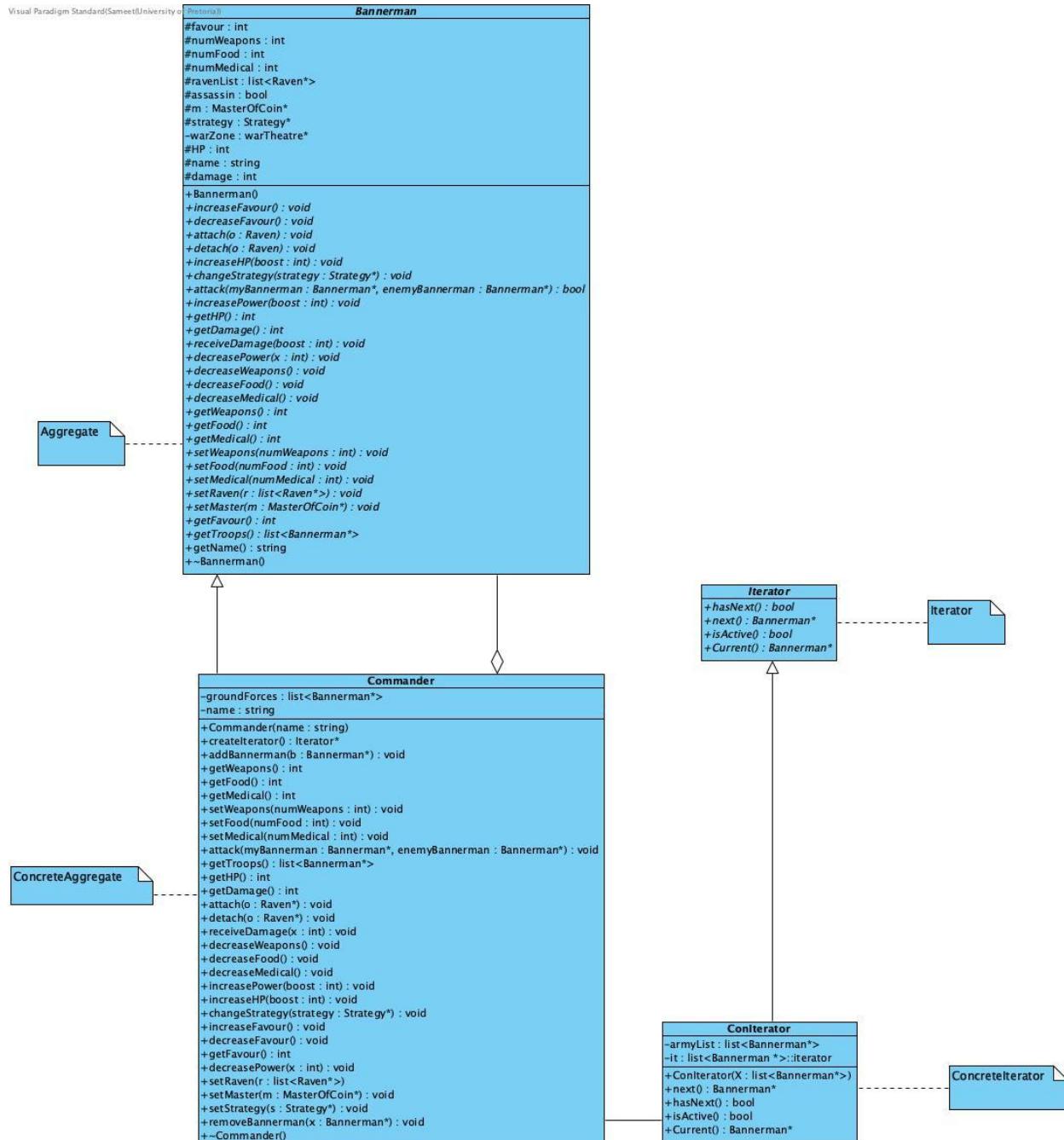
- There must be a hierarchy system in place to show the dynamic between kingdoms, bannermen and troops:

We want Troops and Commanders to both be derivatives of Bannerman and that Commanders are in control of several Troops. Therefore to accomplish this we make use of the **Composite Design Pattern**. The virtual functions in Bannerman are each implemented in the leaf or composite participant. This accurately represents the hierarchy in which one commander controls several troops but a single troop cannot order other troops or commanders.



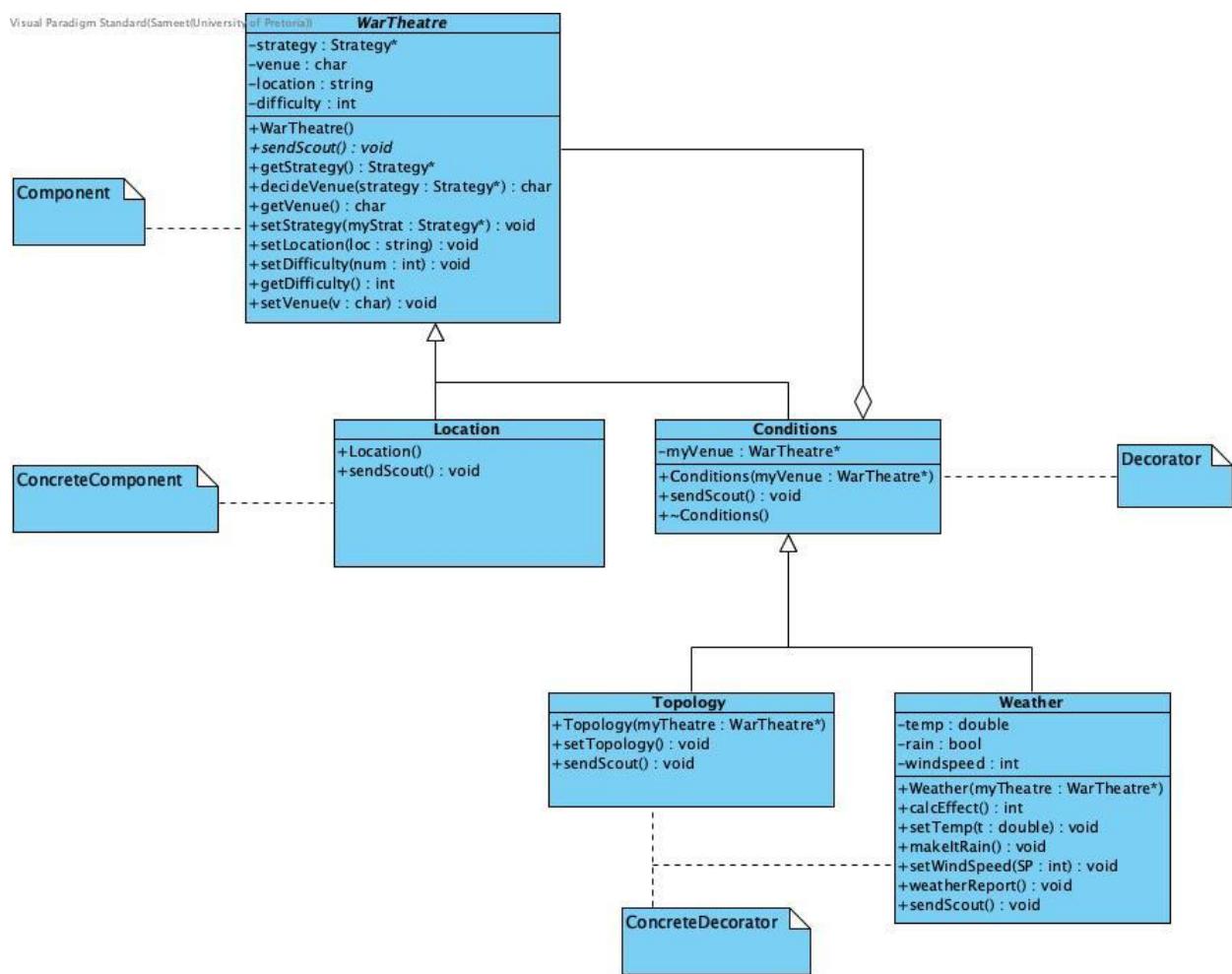
- The system must be able to iterate through the troops:

We use the **Iterator Design Pattern** to iterate through the Bannerman, Commander specifically such that you can get access to their HP. It Iterates through every troop to get their HP and adds it up to get the total HP for that specific bannerman.



- The system must be able to generate different war theaters which affect the outcome of battle:

For this we use the Decorator Pattern. Depending on the terrain, the more difficult the battle is, which will lessen the damage each bannerman can inflict upon the other. We use Decorator to add additional functionality to the war theaters such as topology and weather. These additional functionalities result in the battle being more or less difficult. The UML diagram is presented below:

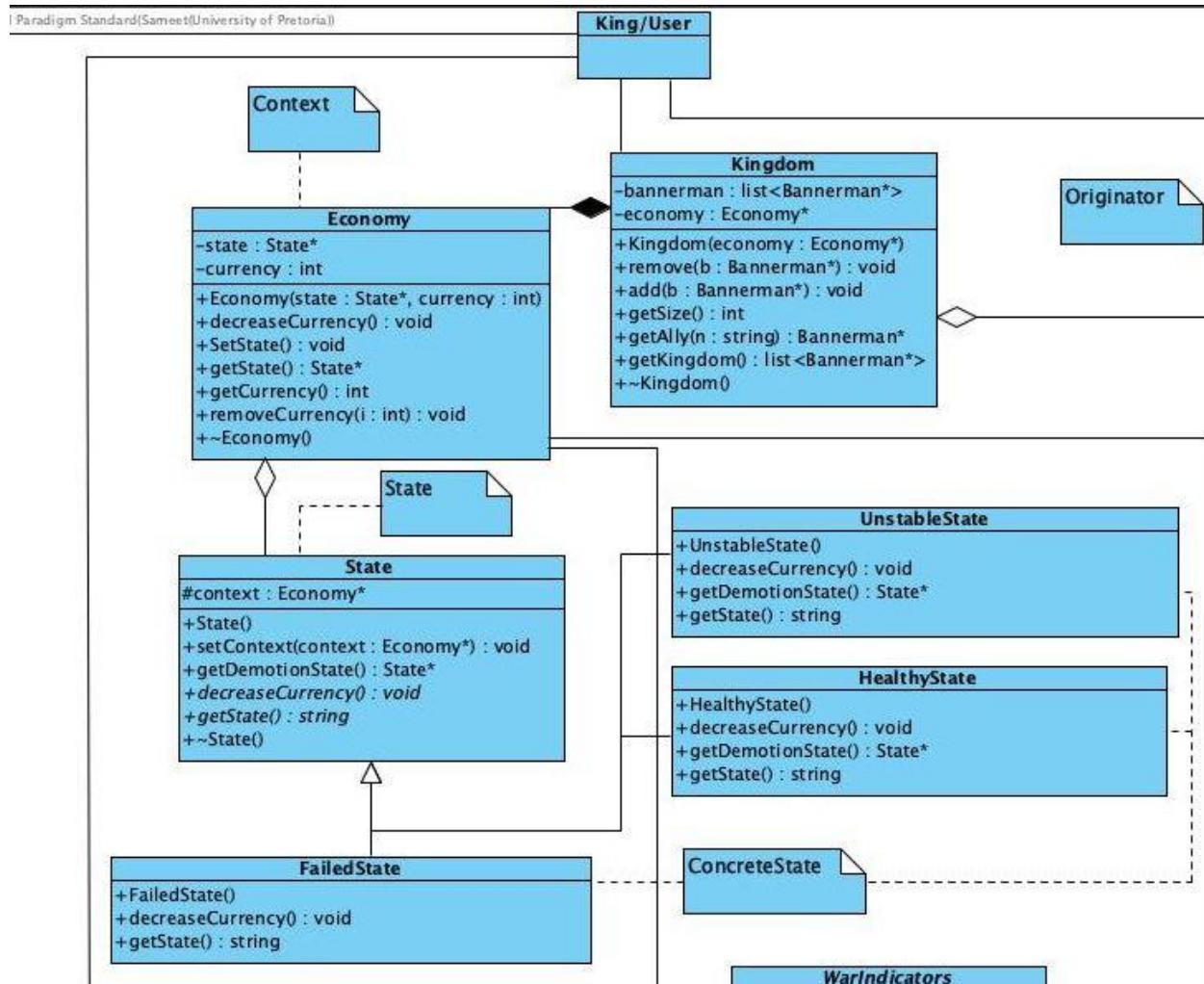


- A user will win the battle if the enemy bannermen are all dead:

At the end of our WarLoop function in our main we have an if-else statement to check whether or not our Enemy kingdom is out of money/bannermen, if so we win.

```
if (PreadoraEco->getState()->getState() == "has failed" || Preadora->getSize() <= 0){
    WarWon();
} else if (DuraEco->getState()->getState() == "has failed" || Dura->getSize() <= 0){
    WarLost();
}
```

- A user will lose the battle if their bannermen are all dead, or they run out of money:

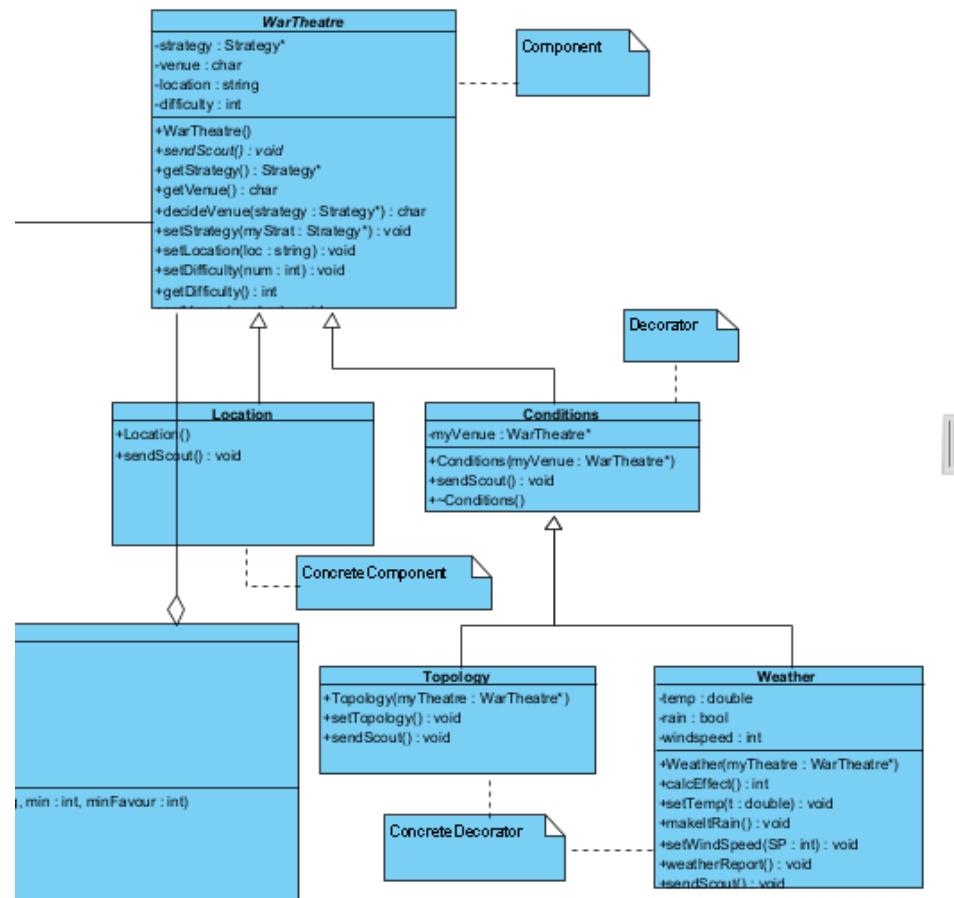


PredoraEco is a pointer to a State object which contains an Economy variable which holds currency, if this currency runs out we return a failed state indicating we've run out

of money. As the war goes on, it costs money to keep up supplies. This money can run out and if it does for your Kingdom's economy you lose the game.

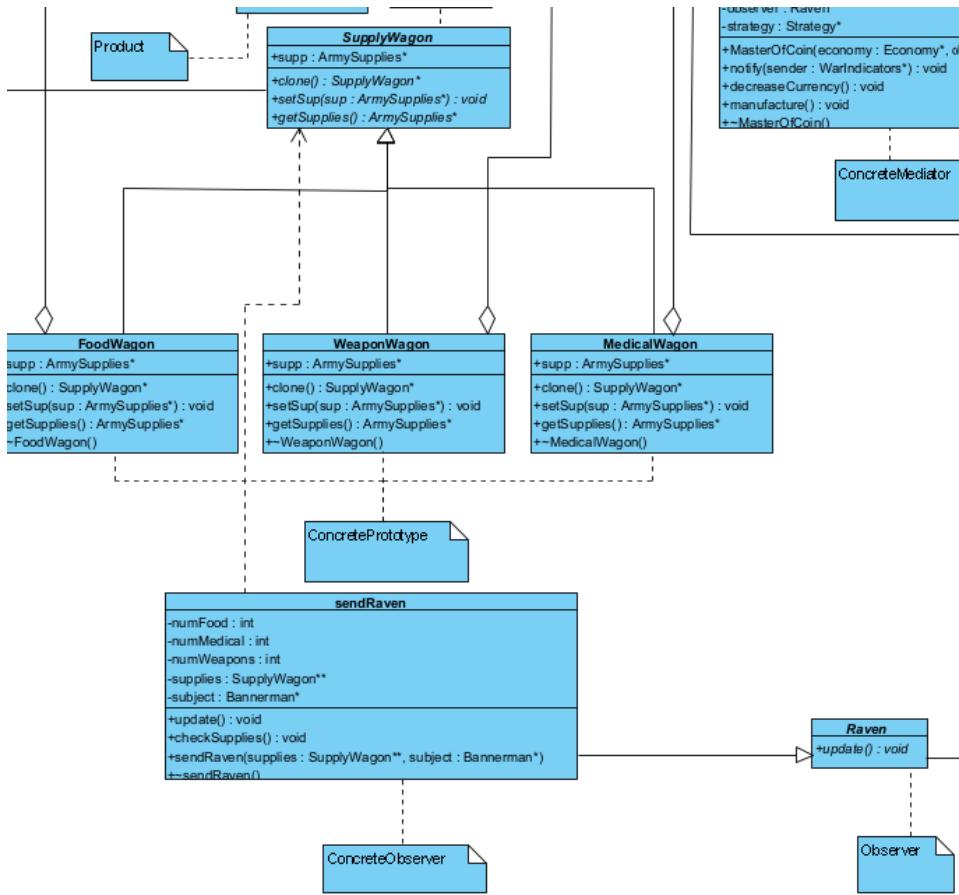
All Components of War to be Implemented as Follows:

1) War Theaters:



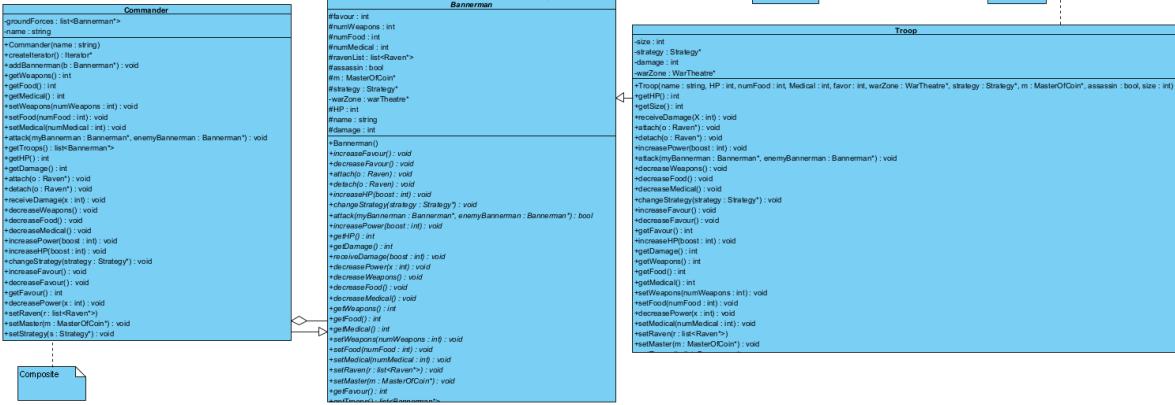
The decorator design pattern was used to address the war theater functionality. The Concrete decorators, Topology and weather determine the advantages and disadvantages that will be added to the battle which takes place at a Location (The Concrete Component). Location is the venue at which the battle between two kingdoms will take place, it is also the concrete component of the decorator design pattern in this implementation.

2) Transportation:



The prototype design pattern is used for the transportation wagons which transport food, medical supplies, and weapons to the Army through the Raven observer. The sendRaven Concrete observer creates a prototype and each time it requires a new object, the prototype is asked to clone itself.

3) Entities:



The composite design pattern is used in the creation and manipulation of entities which enter into warfare, namely Commander objects(The composite participant) and Troop objects (The leaf participant) which inherit from Bannerman(The Component participant). The commander contains children that are either Commander objects or Troop objects and provides a means to perform battle operations using all its children. The Troop class defines the primitive objects of the Commander and objects of the Troop class are the groups of soldiers in the battle.

4) Phases of War:

Phase 1. Gathering of Militia. This is illustrated in the populateVectors() subroutine in the main class, this is where the health of the kingdom's state is set. Army Commanders are also appointed in this phase of war for each Kingdom- subsequently, every Commander is also assigned Troops they are in charge of.

Phase 2. Mobilization. This is illustrated in the WarLoop() subroutine in the main class, this is where the king(user) can decide to instruct an attack on an enemy kingdom. If the king decides to surrender, he loses the battle. If the attacking kingdom has obliterated the enemy's ground forces, the attacking kingdom wins the war, whereas if the attacking kingdom's ground forces have been obliterated, the attacking kingdom will lose the war.

Phase 3. Occupy. The winning kingdom occupies the losing kingdom's land after conquering.

5) Changes to War Engine:

Bannermen can choose to turn against a kingdom's country group and join an enemy kingdom if loyalty favor is depleted.

When Bannermen are attacked, their HPs decrease- subsequently the armies die and their objects are destroyed. Once the kingdom has no Bannermen remaining, they lose the war and the enemy takes over their territory.

The user/king can choose a different or stronger attack strategy at any time they see fit, this is made possible by use of the prototype design pattern as described earlier.

The following are additional components:

6) Communication:

In war there has to be communication between all participating parties and thus we account for this with the Raven class in the Observer pattern which makes sure supplies are manufactured and sent out. Our Mediator also ensures communication with our Master of Coin, who keeps track of when money is spent for supplies.

7) Economy:

You need a strong economy to win a war and we have made use of the State Design pattern to ensure that we account for this. Our economy has several states all of which depend on the amount of money we have access to.

8) Politics:

What is a war without juicy politics? As mentioned earlier if your favor with your allied bannerman decreases, their allegiance to your cause grows thin. If their favor is too little they will betray you and join the enemy. It is important to make good choices to keep your allies happy. We make use of Memento to keep track of the defected bannermen.

9) Battle Tactics:

Battle tactics are what make you win at the end of the day. No success comes without a good strategy, As mentioned earlier we do in fact have a system in place in which you select what strategy you want to use to attack your enemy. This is done via the State Pattern.

Task 5:

- Git used and information included in submission readme.txt
- Doxygen used and information included
- Unit testing included

Task 6:

The following images represent instances of the simulator upon running the program:

```
A new dawn arrives and brings with it a red sky.  
You rule the great kingdom of Dura, which is in the midst of a great battle for the western  
isles, the United Kingdoms of Preadora.  
press Enter to continue...  
  
[A complex ASCII-art representation of a battlefield scene with soldiers, horses, and fortifications]  
  
press Enter to continue...  
=====  
  
Here is the latest news from your council:  
You have 3 Bannermen standing at your side.  
The state of your economy is healthy with the great banks valuing the kingdom's  
riches at 100 gold daras.  
Do you wish to launch an attack on Preadora?(y/n)  
y  
=====
```

Choose an enemy bannerman to attack:

```
Chose which enemy you would like to attack:  
  
0: Cirrane  
HP: 500  
HP of Squadron 1: 200  
HP of Squadron 2: 300  
Damage that this Bannerman can inflict: 35  
Allegiance that this bannerman has to the enemy: 7/10  
  
1: Bagger's Valley  
HP: 500  
HP of Squadron 1: 200  
HP of Squadron 2: 300  
Damage that this Bannerman can inflict: 35  
Allegiance that this bannerman has to the enemy: 7/10  
  
2: Marnmouth  
HP: 500  
HP of Squadron 1: 200  
HP of Squadron 2: 300  
Damage that this Bannerman can inflict: 35  
Allegiance that this bannerman has to the enemy: 7/10  
  
Name your chosen bannerman:0  
You chose to attack: Cirrane  
=====
```

Choose the bannerman fighting for you:

```
Chose which bannerman will fight for you:  
0: Stratham  
HP: 500  
HP of Squadron 1: 200  
HP of Squadron 2: 300  
Damage that this Bannerman can inflict: 29  
Allegiance that this bannerman has to you: 7/10  
  
1: Trudid  
HP: 500  
HP of Squadron 1: 200  
HP of Squadron 2: 300  
Damage that this Bannerman can inflict: 29  
Allegiance that this bannerman has to you: 7/10  
  
2: Mirefield  
HP: 500  
HP of Squadron 1: 200  
HP of Squadron 2: 300  
Damage that this Bannerman can inflict: 29  
Allegiance that this bannerman has to you: 7/10
```

Choose your strategy of attack:

1: Battle Field 2: Siege 3: Ambush 4: Send Assassin

1

BattleField

=====

A scout has been sent to the enemies location. We await news.

...

A raven from the scout has returned! Here is the news:

Gathering weather report...

The weather at the venue is as follows:

Temp: 29

Rain: True

WindSpeed: 39

The battle is at an open war field

=====

Battlefield attack with the user losing:

```
You face your foe on the battlefield!
You deal 34 damage to the enemy. Enemy HP:466
The enemy delivers their blow! The bannerman from Trudid takes 24 damage. Our HP: 476
You deal 28 damage to the enemy. Enemy HP:438
The enemy delivers their blow! The bannerman from Trudid takes 30 damage. Our HP: 446
You deal 34 damage to the enemy. Enemy HP:404
The enemy delivers their blow! The bannerman from Trudid takes 24 damage. Our HP: 422
You deal 34 damage to the enemy. Enemy HP:370
The enemy delivers their blow! The bannerman from Trudid takes 24 damage. Our HP: 398
You deal 35 damage to the enemy. Enemy HP:335
The enemy delivers their blow! The bannerman from Trudid takes 23 damage. Our HP: 375
You deal 25 damage to the enemy. Enemy HP:310
The enemy delivers their blow! The bannerman from Trudid takes 33 damage. Our HP: 342
You deal 26 damage to the enemy. Enemy HP:300
The enemy delivers their blow! The bannerman from Trudid takes 32 damage. Our HP: 310
You deal 22 damage to the enemy. Enemy HP:278
The enemy delivers their blow! The bannerman from Trudid takes 36 damage. Our HP: 300
You deal 26 damage to the enemy. Enemy HP:252
The enemy delivers their blow! The bannerman from Trudid takes 32 damage. Our HP: 268
You deal 25 damage to the enemy. Enemy HP:227
The enemy delivers their blow! The bannerman from Trudid takes 33 damage. Our HP: 235
You deal 25 damage to the enemy. Enemy HP:202
The enemy delivers their blow! The bannerman from Trudid takes 33 damage. Our HP: 202
You deal 26 damage to the enemy. Enemy HP:176
The enemy delivers their blow! The bannerman from Trudid takes 32 damage. Our HP: 170
You deal 29 damage to the enemy. Enemy HP:147
The enemy delivers their blow! The bannerman from Trudid takes 29 damage. Our HP: 141
You deal 27 damage to the enemy. Enemy HP:120
The enemy delivers their blow! The bannerman from Trudid takes 31 damage. Our HP: 110
You deal 30 damage to the enemy. Enemy HP:90
The enemy delivers their blow! The bannerman from Trudid takes 28 damage. Our HP: 82
You deal 23 damage to the enemy. Enemy HP:67
The enemy delivers their blow! The bannerman from Trudid takes 35 damage. Our HP: 47
You deal 29 damage to the enemy. Enemy HP:38
The enemy delivers their blow! The bannerman from Trudid takes 29 damage. Our HP: 18
You deal 30 damage to the enemy. Enemy HP:8
The enemy delivers their blow! The bannerman from Trudid takes 28 damage. Our HP: 0
Trudid fought bravely in the name of Dura. They were sadly lost in the battle.
=====
```

Assassination attempt:

```
Choose your strategy of attack:  
1: Battle Field      2: Siege      3: Ambush      4: Send Assassin  
4  
Assassinate  
=====  
  
A scout has been sent to the enemies location. We await news.  
...  
A raven from the scout has returned! Here is the news:  
Gathering weather report...  
-----  
The weather at the venue is as follows:  
Temp: 17  
Rain: True  
WindSpeed: 27  
-----  
The battle is at the enemy castle  
=====  
  
You sent an assassin into the chambers of the enemy commander under the cover of night.  
The assassin was spotted and immediately executed.  
=====
```

Only one bannerman left:

```
Chose which bannerman will fight for you:  
0: Mirefield  
HP: 500  
HP of Squadron 1: 200  
HP of Squadron 2: 300  
Damage that this Bannerman can inflict: 29  
Allegiance that this bannerman has to you: 5/10
```

```
Name your chosen bannerman:0  
You chose to have Mirefield as your fighter.  
=====
```

The bannerman defected as you led your last attack, resulting in losing the war.

```
Choose your strategy of attack:  
1: Battle Field      2: Siege      3: Ambush  
2  
Siege  
=====  
  
A scout has been sent to the enemies location. We await news.  
...  
A raven from the scout has returned! Here is the news:  
Gathering weather report...  
-----  
The weather at the venue is as follows:  
Temp: 46  
Rain: False  
WindSpeed: 50  
-----  
The battle is at the enemy's fortress  
=====  
  
The commanders are losing faith in your cause. Mirefield has defected to the other side.  
=====  
  
Here is the latest news from your council:  
You have 0 Bannermen standing at your side.  
The state of your economy is healthy with the great banks valuing the kingdom's  
riches at 100 gold daras.  
The long battle has ended.  
=====  
Preadora's bannermen have overwhelmed your forces and have been crowned victorious after these long battles. Dura is under her rule.  
=====
```

Successful assassination:

```
Choose your strategy of attack:  
1: Battle Field      2: Siege      3: Ambush      4: Send Assassin  
4  
Assassinate  
=====  
  
A scout has been sent to the enemies location. We await news.  
...  
A raven from the scout has returned! Here is the news:  
Gathering weather report...  
-----  
The weather at the venue is as follows:  
Temp: 23  
Rain: True  
WindSpeed: 33  
-----  
The battle is at the enemy castle  
=====  
  
You sent an assassin into the chambers of the enemy commander under the cover of night.  
The assassin was successful and disposed of the enemy's commander without being seen.  
With the enemy at their door, some of the Preadorean soldiers decided to join the Dura troops. The rest took their oath.  
  
The Preadora Kingdom has put a price on the Assassin's head, so he fled.  
=====
```

Battlefield fight, defeating the last enemy with sole bannerman standing, winning the war.

```
You deal 20 damage to the enemy. Enemy HP:50
The enemy's supplies are low! They take extra damage. Enemy HP:30
The enemy delivers their blow! The bannerman from Stratham takes 14 damage. Our HP: 286
Our supplies are low! We take extra damage. Enemy HP:272
You deal 18 damage to the enemy. Enemy HP:12
The enemy's supplies are low! They take extra damage. Enemy HP:0
The fight has ended. The commander took inventory of the supplies and sends for more.
Supplies have been received.
The enemy kingdom's bannerman has been defeated. Cirrane is now under your kingdom's rule.
=====
```

```
Here is the latest news from your council:
You have 1 Bannermen standing at your side.
The state of your economy is healthy with the great banks valuing the kingdom's
riches at 75 gold daras.
The long battle has ended.
=====
The enemy has been defeated! All the land under the Kingdoms of Preadora are now under your rule! Long live Dura!
=====
```

Surrender choice made:

```
Here is the latest news from your council:
You have 3 Bannermen standing at your side.
The state of your economy is healthy with the great banks valuing the kingdom's
riches at 100 gold daras.
Do you wish to launch an attack on Preadora?(y/n)
n
By choosing not to attack Preadora you will have to raise the white flag of surrender.
Do you agree?(y/n)
y
=====
You have raised the white flag. Preadora is victorious.
=====
```

Management

Basic outline of work distribution and internal deadlines.

Task 1:

Whole group contributed.

Deadline: 3 October - met

Task 2:

Functional requirements: Group contributions

Activity Diagrams: Julianna Venter

Patterns: group contributions

Class diagrams: Sameet Keshav

Sequence/Communications diagrams: Ronin Brookes

State Diagrams: Morgan Bentley

Object Diagrams: Thapelo Thoka

Deadline: 18 October - met

Task 3:

1. Memento: Julianna Venter
2. State: Morgan Bentley
3. Abstract Factory: Ronin Brookes
4. Prototype: Ronin Brookes
5. Mediator: Sameet Keshav

6. Observer: Sameet Keshav
7. Strategy: Morgan Bentley
8. Composite: Thapelo Thoka
9. Iterator: Thapelo Thoka
10. Decorator: Keabetswe Mothapo

Testing and updates: Each did own

Main functionality and simulation: Julianna Venter

Deadline: 31 October - met (final version 6 November - met)

Task 4:

Research report: Julianna Venter & Keabetswe Mothapo (**Done 13 - 24 October**)

Design report: Ronin Brookes & Thapelo Thoka

Deadline: 6 November - met

Task 5:

Github management: Sameet Keshav

Doxxygen management: Morgan Bentley

Unit Testing/Google Test: Sameet Keshav & Thapelo Thoka

Video Presentation: Keabetswe Mothapo

Deadline: 6 November - met