Daniel Azmanov – u20441135 (Group Leader)
Ruben Brits – u20463163
Reece Stephenson – u20431997
Kyle Pottinger – u20660652
Seyuran Pillay – u20471582

COS 214

PROJECT

CheemsChaps

COS 214 Project Final Report
CheemsChaps

Google Docs Report Link:

https://docs.google.com/document/d/1ujiAzFaYzfVJJ3HDp2y3NF8eIldMGeU7/edit?usp=sharing&ouid=116243338212191316961&rtpof=true&sd=true

GitHub repo link:
https://github.com/priyo5/CheemsChaps

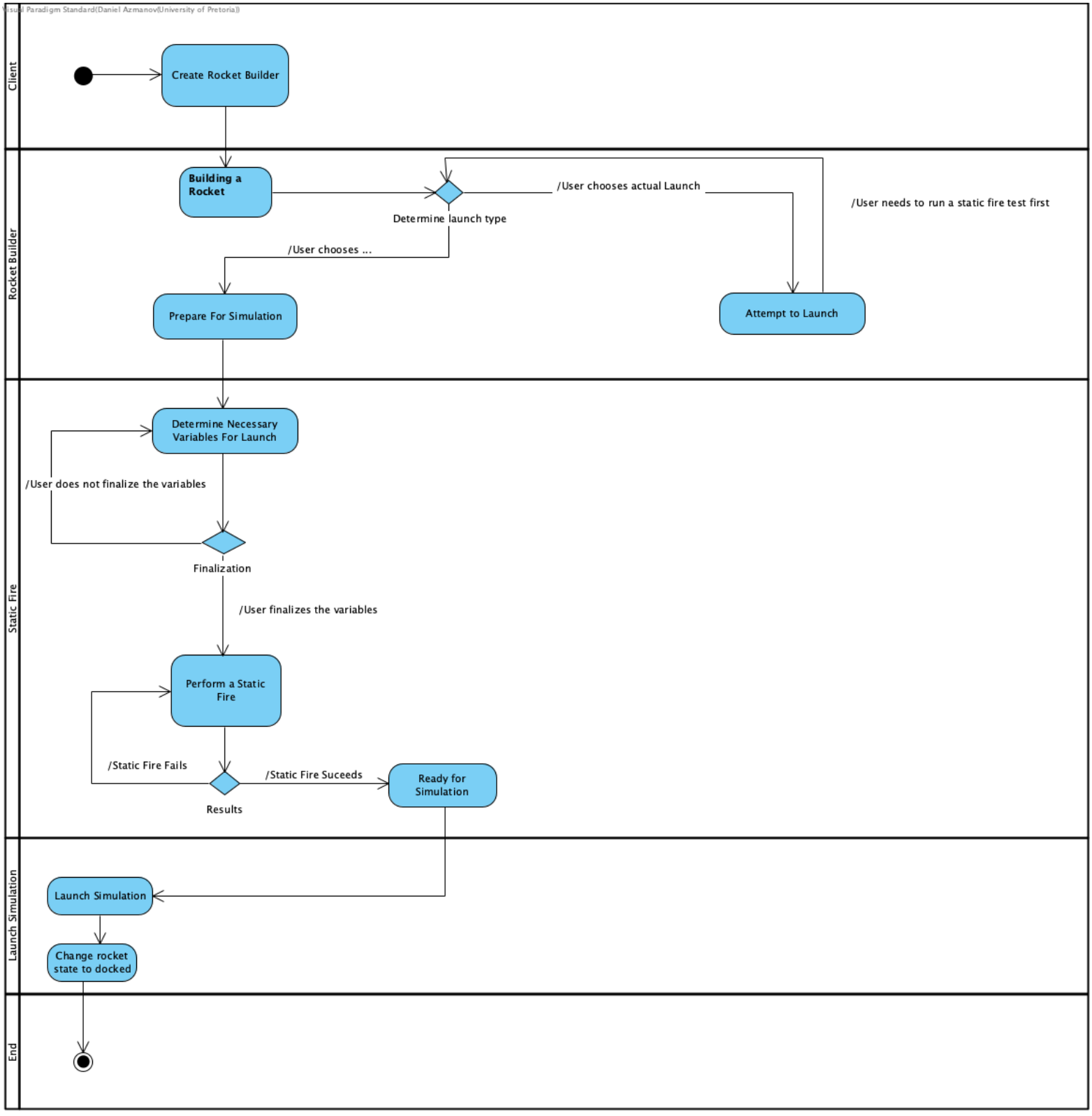
1.1 - Functional Requirements

- Build the rocket by allowing 3 rocket formats to be built:
    - PeopleRocketBuilder to build a rocket that carries humans
    - SatelliteRocketBuilder to build a rocket that carries satellites
    - StarlinkRocketBuilder to build a rocket that carries many Starlink satellites
- Choose Destination for the flight
    - Mars
    - Jupiter
    - The Moon
- Different spacecrafts (will be predetermined by one of the 3 rocket formats chosen).
    - Crew Dragon – Payload will be people
        - Set the number of people to be carried as specified by the user.
        - Fixed capacity of 20 people for Crew Dragon
    - Dragon Spacecraft – Payload will be Cargo
        - Cargo can be decorated with a Starlink Fleet (60 Satellites) or a single Satellite.
        - This cargo will have an observer attached to it.
- Do a static fire test on a built rocket, and test the following:
    - Will allow the user to modify contents of the rocket before and after static fire tests.
    - The rocket has a RocketType set.
    - The rocket has stage 1 and stage 2 added
    - The rocket has **a** spacecraft added.
    - Rocket has enough fuel to reach the destination (determined by a function).
- Stages – merely cout statements to tell the story of what's happening.
    - Stage 1:
        - Will deplete fuel and output the remaining fuel.
        - When the fuel is 0, stage 2 will fire.
    - Stage 2:
        - Will deplete fuel and output the remaining fuel.
        - When the fuel is 0, the Payload will arrive at it's desired orbit.
- Launch
    - The rocket object must be in a state LaunchReady()
    - The rocket object may not be modified once Launch sequence has begun.
    - Then will launch the simulation
    - cout story according to set variables about the rockets trip
    - Will print out the details of the rocket after the launch
        - With 0 fuel remaining and payload offloaded.

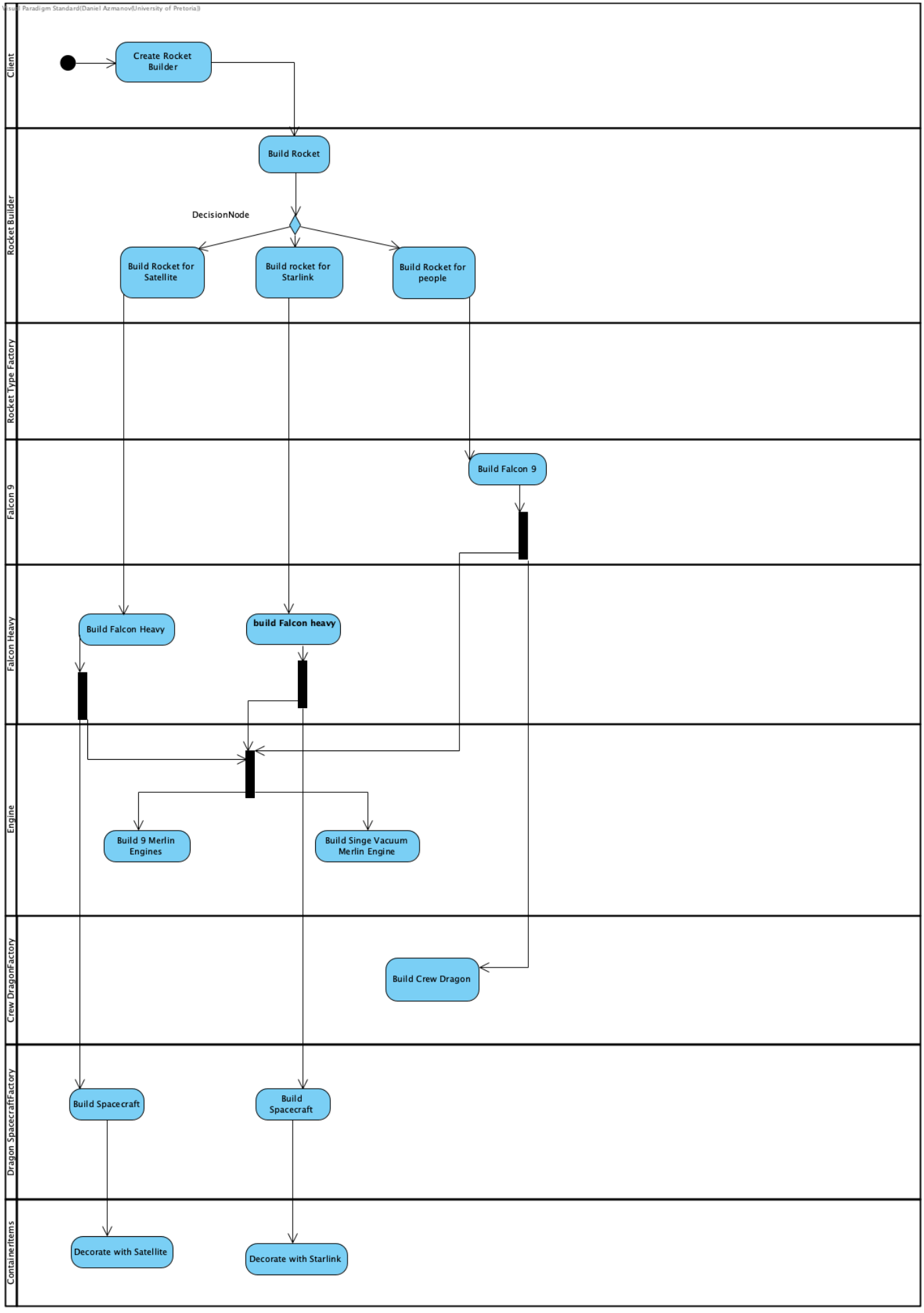
- Landing
    - Set the rocket object to a dockedReady() state.
    - Ask the user whether they want to restore the rocket to its original state before the launch or not.
    - Let the user decide to continue the simulation with the restored rocket or a new rocket.
    - Let the user end the simulation at the end of a launch sequence.
- Satellite communication
    - Use observer to detect, update and notify when the Starlink Fleet or a single satellite has arrived at its destination orbit.

1.2 - Activity Diagram:

Activity Diagram showing the process of a Rockets creation.



Activity Diagram showing the process of a Rockets creation.



1.3 - Design Patterns and their responsibilities:

- Factory:
    - Making the different rocket types
    - Making the different Spacecrafts
    - Making the different Engines

- Builder:
    - Builds the rocket with different components
    - The rockets built are between three popular models.
        - People Rocket: Falcon 9 Rocket, Crew Dragon Spacecraft with a stage 1 Merlin Engine and a stage 2 Vacuum
        - Satellite Rocket: Falcon Heavy Rocket, Dragon Spacecraft with satellite with a stage 1 Merlin Engine and a stage 2 Vacuum Merlin Engine
        - Starlink Rocket: Falcon Heavy Rocket, Dragon Spacecraft with satellite with a stage 1 Merlin Engine and a stage 2 Vacuum Merlin Engine
- Command:
    - Test the rocket (static test fire)
    - Launches the rocket (Actual launch)
- State:
    - The state if the rocket refers to the following:
        - StaticFireReady() – This means that the rocket hasn't passed the Static Fire test, and is not cleared for launch.
        - LaunchReady() – This means that the rocket has passes the Static Fire test and it is cleared for launch.
        - DockedReady() – This means that the rocket has reached it destination and it is ready to be docked at the Space station.

- Memento:
    - Restore the rocket in the following way:
        - The rocket after the launch must have it's variables set the rockets variables before the launch sequence.
        - Allows users to re-use a rocket in multiple simulations.

- Template Method:
    - Will allow engines to set their own fuel depletion rates.

- Observer:
    - Will be attached to the Cargo (either a satellite or a starlink)
    - Will notify when the cargo has arrived at its destination.

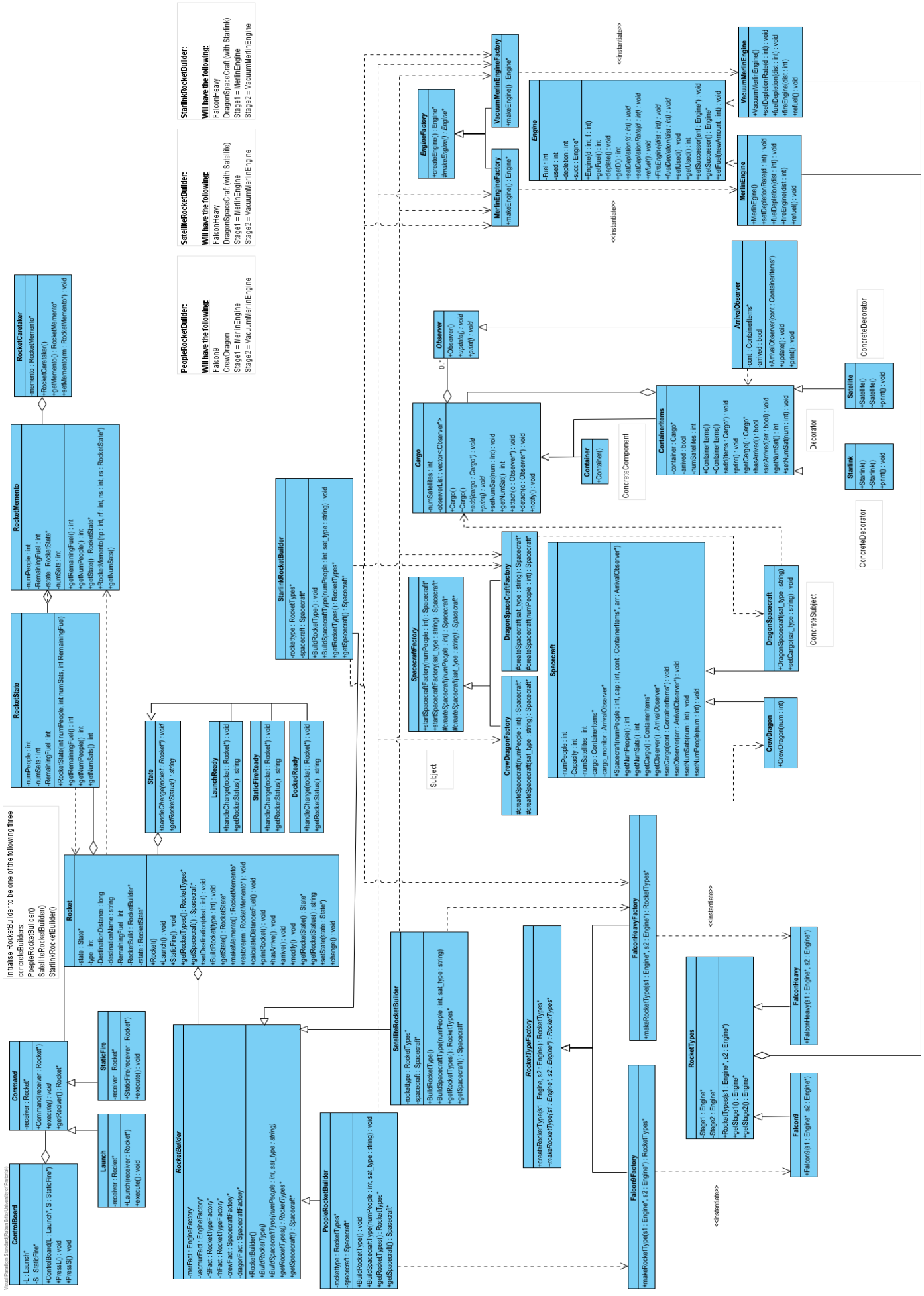
- Decorator:
    - Decorate the Cargo with specific satellites.
        - Starlink Fleet of 60 Satellites
        - A single Satellite.

- Composite:
    - Allow user to treat the Container and ContainerItems uniformly.

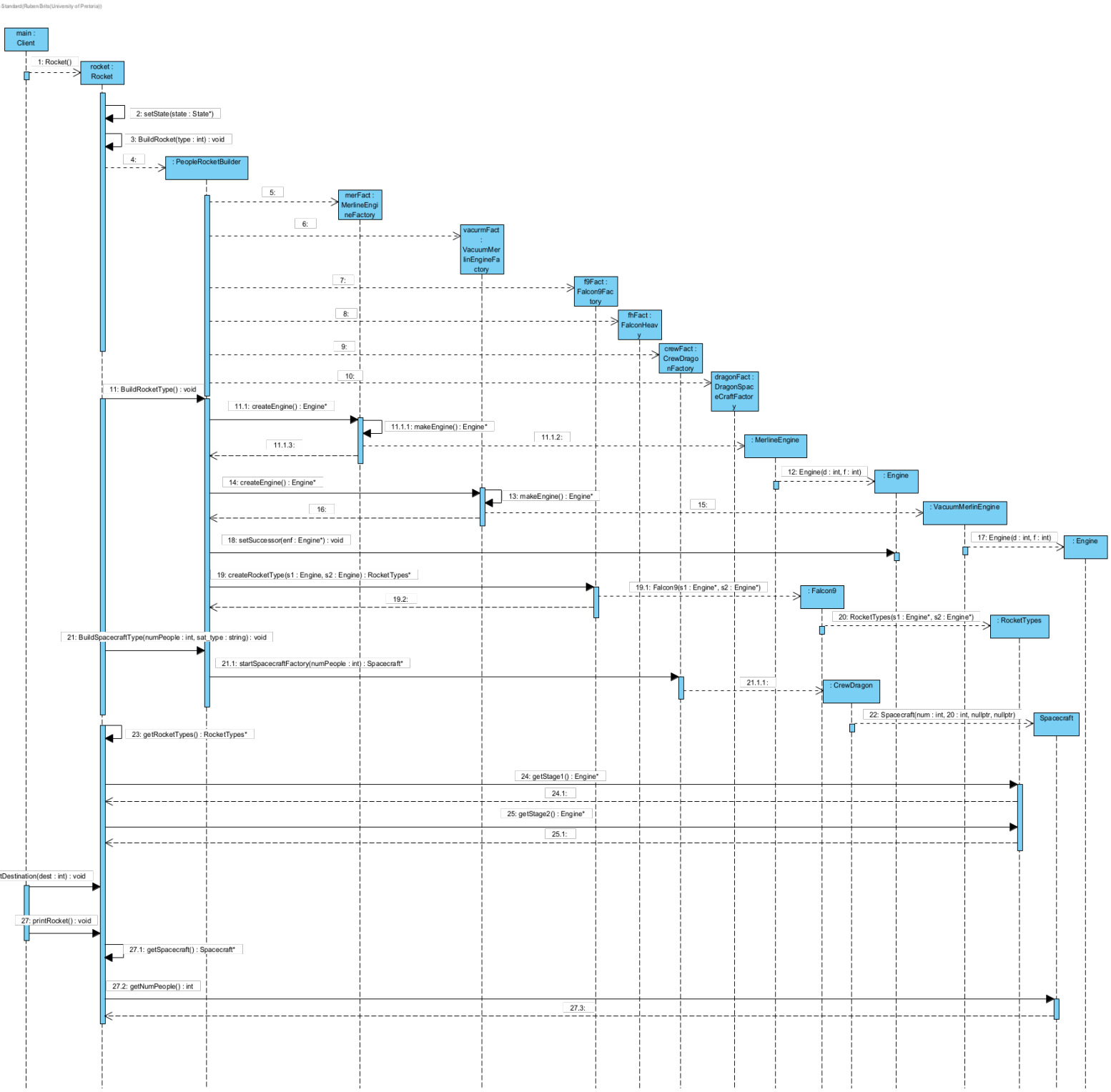
- Chain of Responsibility:

- Stage1 (Merlin Engine) has a successor Stage2 (Vacuum Merlin Engine).
- RocketBuilder defers the fireEngine() function from Stage1 to Stage2 to allow each stage to fire one at a time, and one after the other.

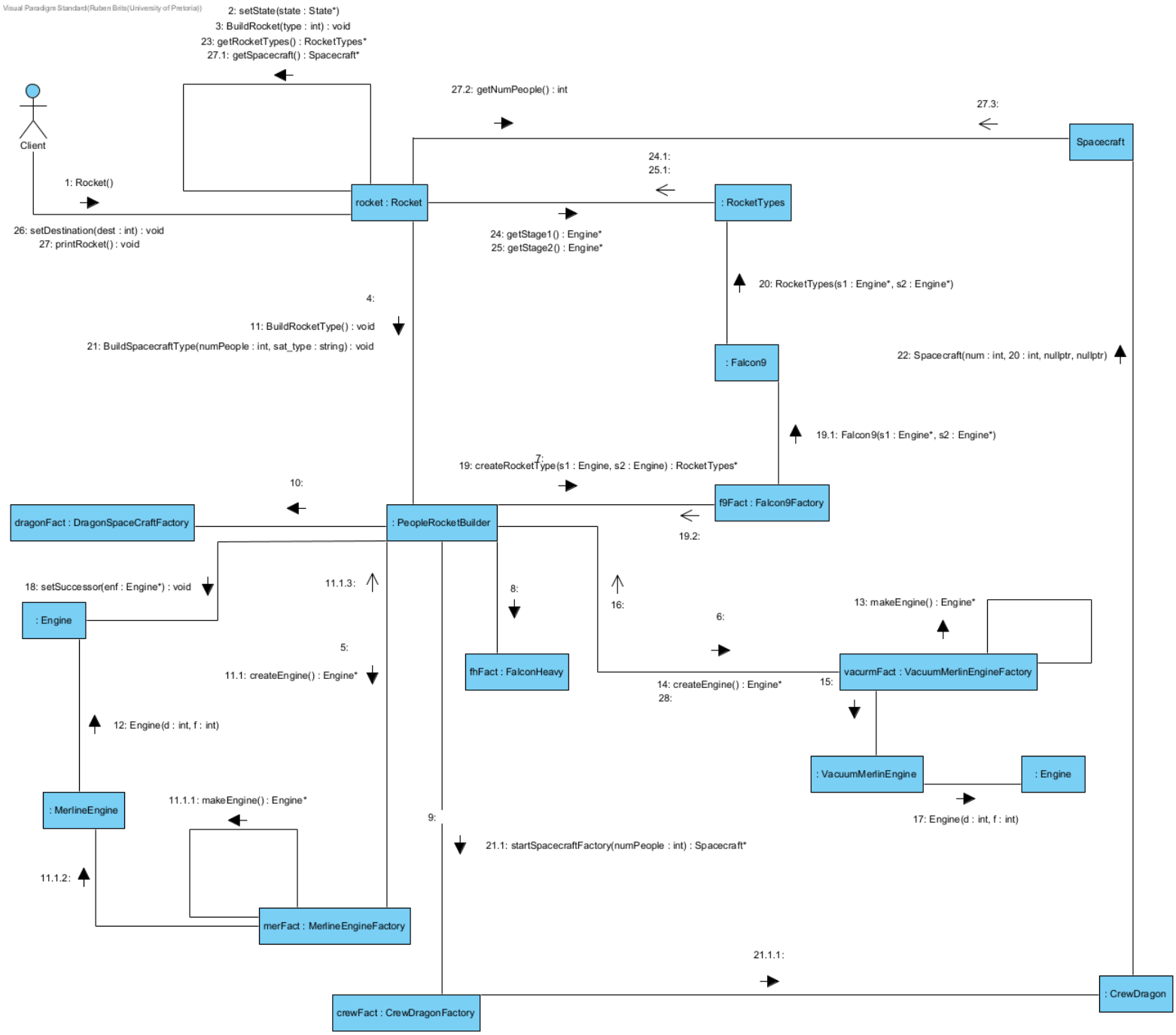
1.5 - Class Diagram:



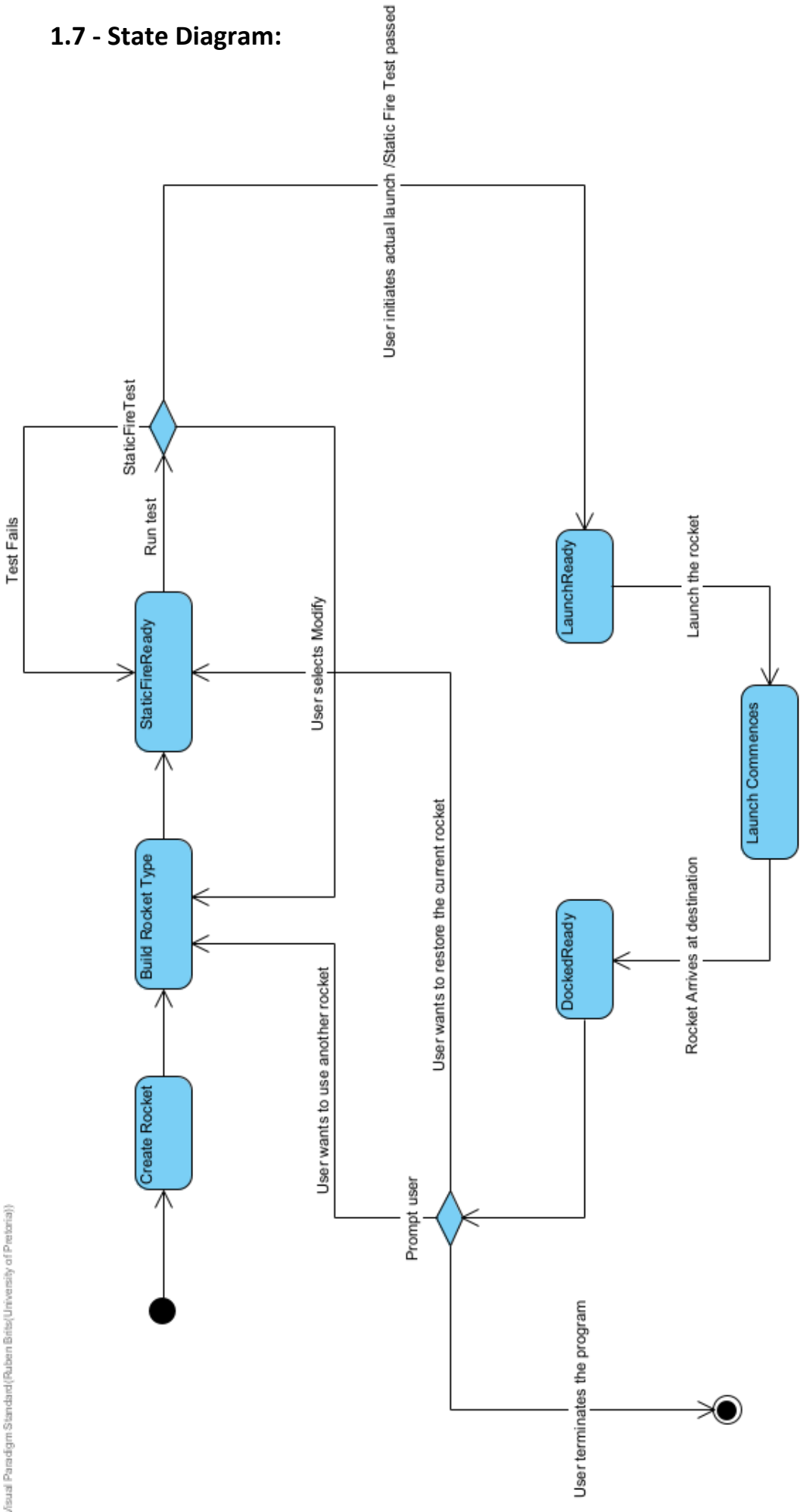
1.6 – Sequence Diagram:



1.6 – Communication Diagram:

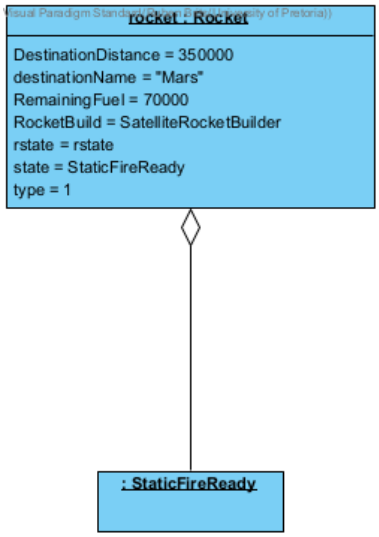


1.7 - State Diagram:

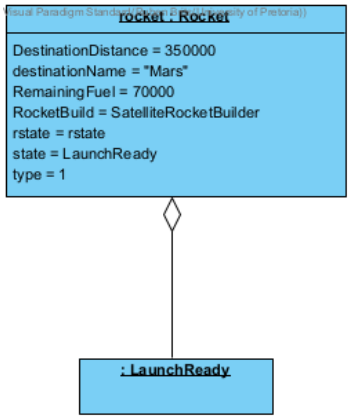


1.8 - Object Diagrams:

Object diagram showing a rocket after initialisation



Object diagram showing a rocket after it has passed the static fire test (ready for launch)



Object diagram showing a rocket after the rocket has launched docked at the space station