COS 214 Prac 6

Members:

Student Number	Name Surname
u21428574	Ms. Mmanape Manamela
u17209600	Mr. Daniel Lukyanov
u17267162	Ms. Saskia Steyn
u22498622	Mr. Stefan Müller
u22550055	Mr. Stefan Jansen van Rensburg
u22590732	Mr. Johan Jansen van Rensburg
u22526162	Mr. Amadeus Fidos

Task 2.1 (Functional Requirements):

Buildings:

The city contains various **building sections** like **Residential**, **Commercial**, **Industrial**, and **Landmarks**. Each section includes multiple buildings that have the following core attributes:

- `maxCitizens`: Maximum citizens that can occupy or work at the building.
- `currentNumberOfCitizens`: The current number of citizens working or living in the building.
- `sectionCoordinate`: The coordinate representing the location of the building in the city grid.

Some specialised sections have additional attributes:

- **Residential Buildings** (e.g., complexes) will have a `numberOfHouseholds` attribute that tracks the number of families living in the complex.

Functional Requirements:

1. Buildings will **track citizen capacity**.
2. The **Residential section** will calculate the number of families housed using `numberOfHouseholds`.
3. Buildings will interact with **utilities** (power, water, waste) based on the number of citizens and the building's function.
4. Buildings impact **citizen satisfaction**, resource consumption, and tax revenue generation.

Utilities:

Specialised **utility buildings** are required to serve the city's needs:

- **Power Plants**: Provide electricity.
- **Water Plants**: Supply water to the city.
- **Waste Management**: Handle garbage collection and recycling.

Each utility serves a specific number of citizens, and their operation impacts citizen satisfaction and building functionality.

Functional Requirements:

1. Utilities will supply services based on **citizen count** and **building demands**.
2. Shortages in utilities will **negatively affect citizen happiness** and building operations.
3. Utilities will consume **resources** (e.g., materials, budget) for expansion and maintenance

Transportation:

Each citizen has a primary **transport type** that determines how long it takes them to travel between buildings, influencing their happiness. The transportation system includes:

- **Vehicle Interface**: Specialised into `Car`, `Bus`, or `Taxi`, with each vehicle having a `passengerLimit`.
- **Train Stations**: Handle medium-distance travel and operate only at specific times of the day. Trains will travel between **two end destinations**.
- **Airports**: Handle long-distance travel and resource imports, also operating at specific times with **two end destinations**.

Functional Requirements:

1. Citizens will use **transportation** based on proximity and preferred mode (car, bus, taxi).
2. **Train stations** and **airports** operate on a schedule and can **only transport a set number of passengers** at a time.
3. Efficient transportation will **reduce travel time**, improving citizen happiness.

Citizens:

Each citizen has attributes that reflect their lifestyle and role in the city:

- **Happiness** is influenced by factors like tax rates, housing conditions, and distance to work or required services.
- **Home/Accommodation**: Each citizen has a home, which may be a complex or house in the residential section.
- **Employment Status**: Determines whether they have a job in a commercial, industrial, or landmark building.
- **Money**: Citizens accumulate income, which will later be taxed based on government policies.

Functional Requirements:

1. Citizens will **respond to changes** in housing, transportation, and utilities.
2. **Happiness** will adjust based on **commute times**, tax rates, and availability of services.
3. Citizens will interact with **buildings** based on their needs (housing, employment, entertainment).

Government:

The government system manages **tax policies** and **city budgeting**. It has the power to:

- Set **income taxes** and **business taxes**, which impact the budget and citizen happiness.
- Adjust **policies** to influence population growth or improve public services, even if these changes reduce citizen satisfaction.
- Control the **budget** based on taxes and allocate it toward services like utilities, public transportation, and construction.

Functional Requirements:

1. The government will **adjust tax rates** and collect taxes from citizens and businesses.
2. **Policies** will be introduced to promote growth, but may **negatively impact happiness**.
3. The **budget** will be used to maintain and upgrade buildings, utilities, transportation, and pay government workers' salaries.

Resources:

The city will manage critical **resources**:

- **Materials**: Used for constructing new buildings and expanding utilities.
- **Energy**: Distributed by power plants to buildings and citizens.
- **Water**: Provided by water plants and used by citizens and industries.
- **Budget**: Managed by the government, controlling the city's financial health.

Functional Requirements:

1. Buildings and utilities will consume **materials** for construction and maintenance.
2. **Energy and water** consumption will be tracked and managed to prevent shortages.
3. The city's **budget** will control infrastructure improvements and service quality.

Taxes:

Taxes are a core mechanic, set and managed by the government:

- **Tax Rates**: Income and business tax rates can be adjusted to control revenue.
- **Collection**: Taxes are collected from citizens and businesses based on their income or output.
- **Allocation**: Collected taxes are used to fund public services like transportation, utilities, and city expansion.
- **Impact on Citizens**: Higher taxes will negatively affect citizen happiness, while more public services funded by taxes can offset the discontent.

Functional Requirements:

1. **Income and business tax rates** will be adjustable and affect the city's budget and citizen happiness.
2. The government will **allocate the budget** toward services and growth.
3. Tax changes will **immediately impact happiness**.

City Growth:

City growth is a dynamic system influenced by:

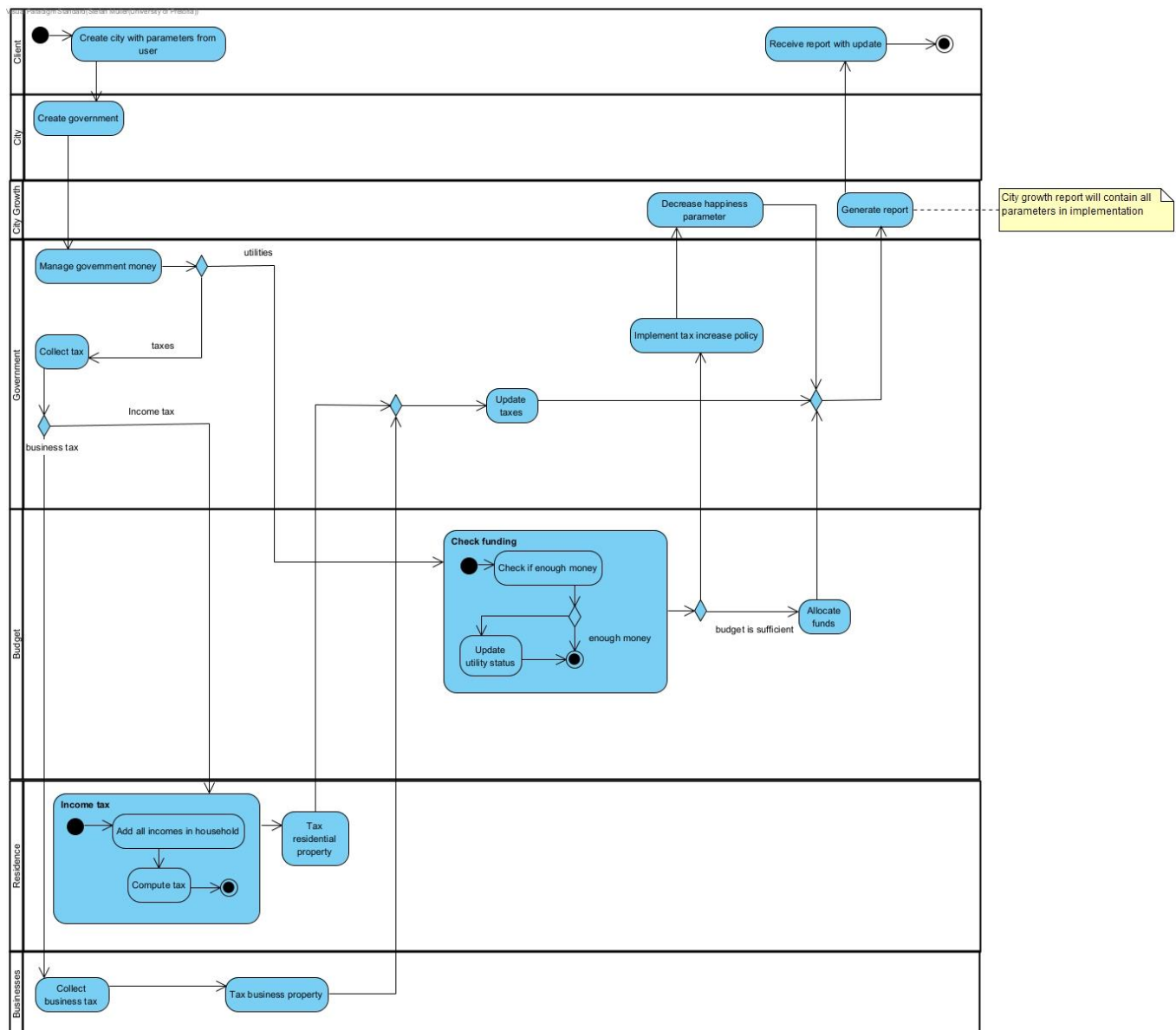- **Population Growth**: Citizens move in or out based on job availability, housing conditions, and overall satisfaction.
- **Housing Needs**: Residential expansion is necessary to accommodate population growth.
- **Economic Development**: The growth of commercial and industrial sectors provides jobs and revenue.
- **Infrastructure Expansion**: Utilities and transportation need to expand as the population grows.
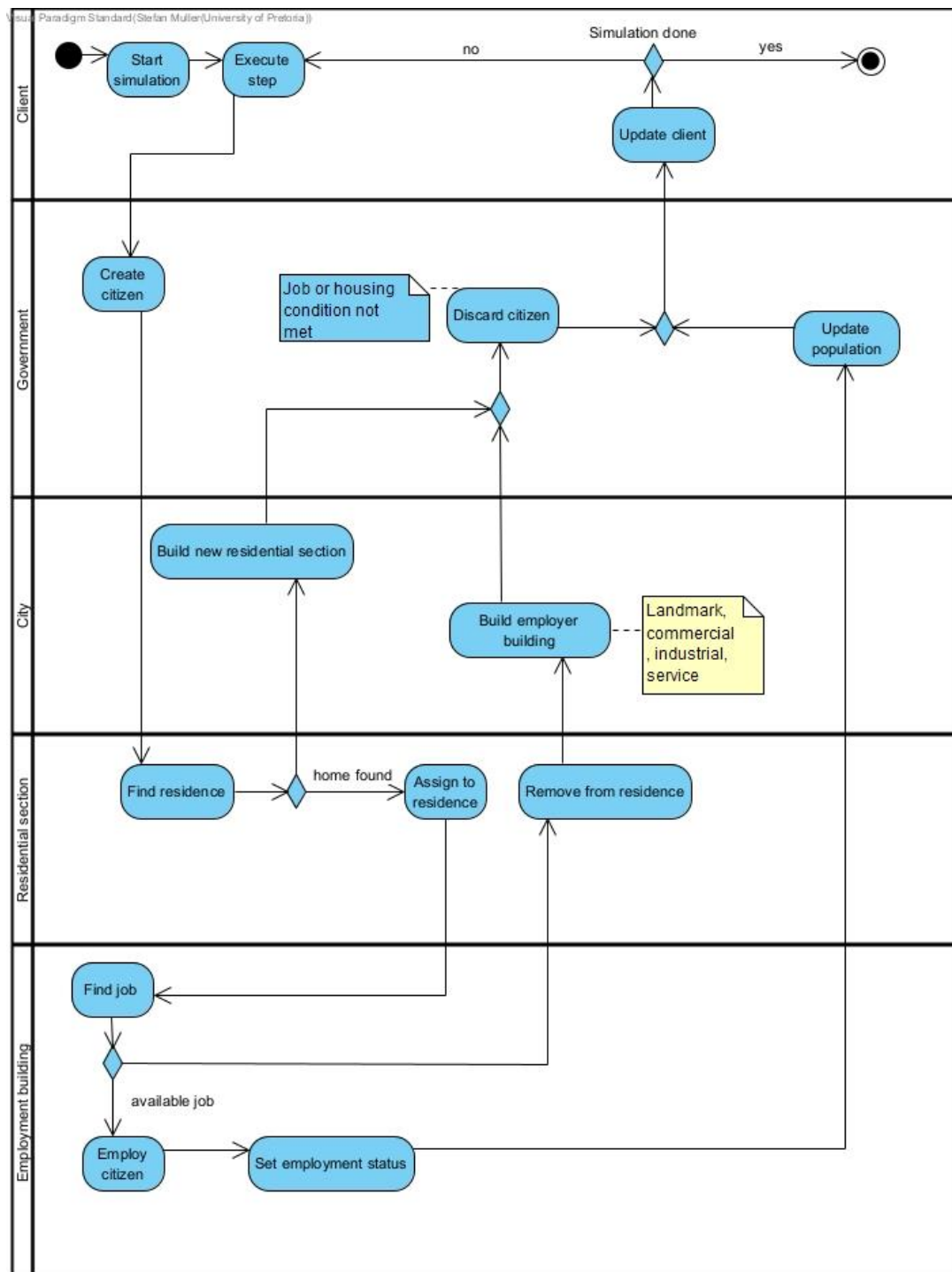
Functional Requirements:

1. The city will **update its parameters** based on population, economy, and infrastructure needs.
2. **Iterative simulations** will track outcomes like housing demand, resource usage, and transportation congestion.
3. Growth will lead to new challenges, such as the need for **more housing** or **expanded utilities**.

Task 2.2 (Activity Diagrams)

Taxation:

Construction:

Client
- Start simulation
- Execute step
- Simulation done → no / yes
- Update client

Government
- Create citizen
- Job or housing condition not met
- Discard citizen
- Update population

City
- Build new residential section
- Build employer building
- Landmark, commercial, industrial, service

Residential section
- Find residence
- home found → Assign to residence
- Remove from residence

Employment building
- Find job
- available job → Employ citizen
- Set employment status

Unemployed status not available (every citizen must have job at a building)

If residence or job buildings cannot accommodate new citizens then build new but discard new citizen

High-level:

Backend (client)

Step through simulation

Simulation complete — Yes

Report to user

Iteration complete

Government

Check population

Fund industry and service

Taxes

Growing?

yes — No action needed

no

tax businesses — tax industries

Update Population

Check num people

Limit reached?

yes — no

Implement population control

City

Add building sections that satisfy citizen needs

Population

Procreate

Pay tax

Sum of income in household

Multiply by tax rate — Pay tax

Work

Service

Industry

Business

Transport

Avialable transport?

no — No work today

yes

Travel to work

Travel home

Business

Make money for business — Get portion of money

Pay business taxes to government

Industrial

Produce resources — Get money for work

Pay business tax

Service

Get paid
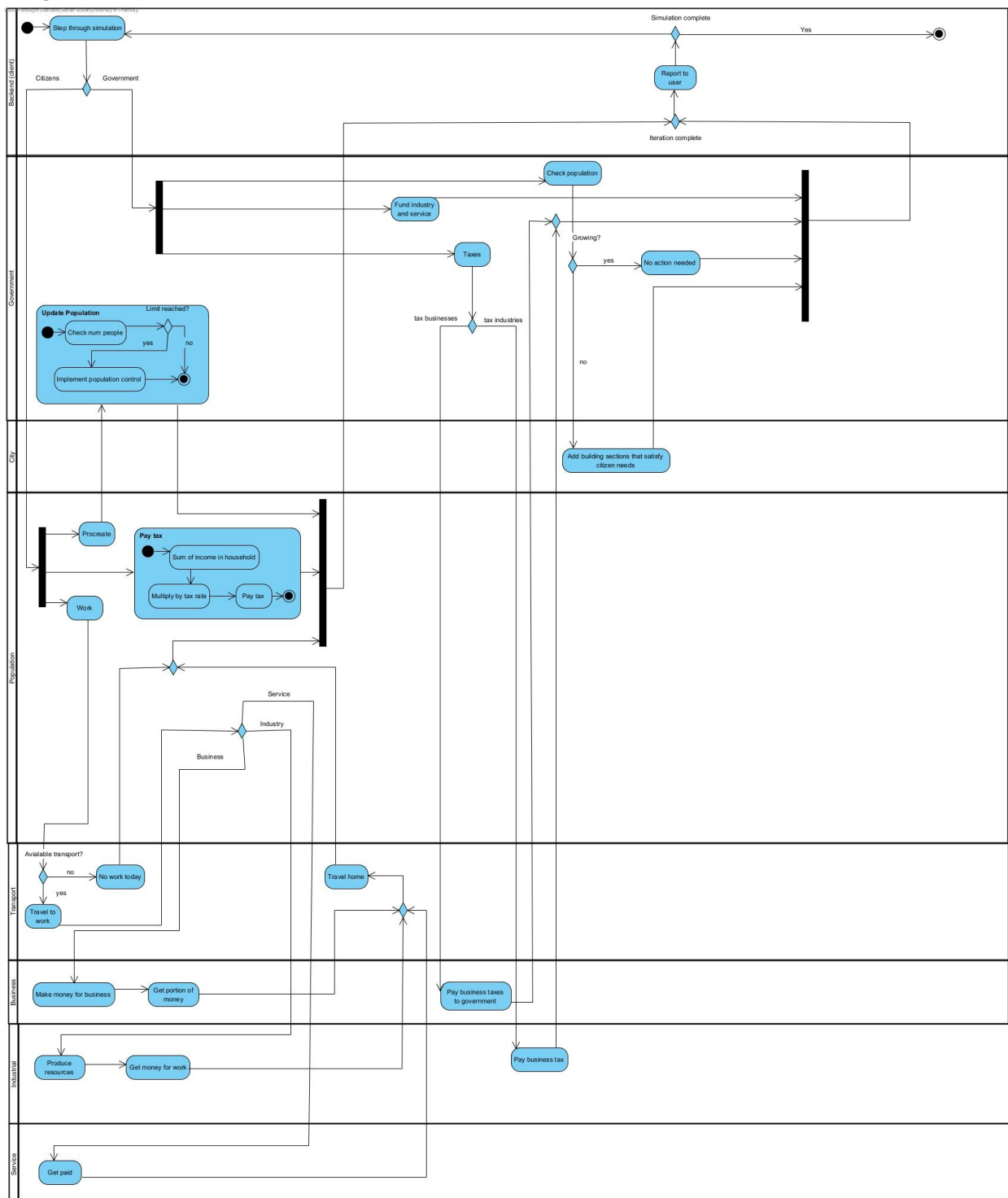
Services pay no tax

Task 2.3 (Design Patterns selected)

- **State Pattern**: The government assesses the overall condition of the city and adjusts its internal state (such as prosperous, growing, or declining) based on the current population, economy, and infrastructure. This state change reflects the city's situation and informs further actions.
- **Strategy Pattern**: The government adopts different strategies to improve the city's condition, switching between approaches like economic growth, infrastructure expansion, or citizen welfare, depending on the current city state and desired outcomes.
- **Command Pattern**: The government issues policies as commands, such as implementing new tax laws, zoning regulations, or infrastructure projects. Each policy is a discrete action that directly affects the city and its citizens.
- **Chain of Responsibility Pattern**: The government is structured into various departments, such as housing, transportation, and finance. Each department handles specific requests and functionality, passing on the responsibility if it's outside their domain, ensuring the appropriate department processes the request.
- **Memento Pattern**: The system allows the user to save and load the city's state at various points, enabling them to revisit previous scenarios and alter decisions or variables for a different outcome, essentially providing a rollback mechanism.
- **Mediator Pattern**: The communication between the frontend (user interface) and the backend (core city simulation logic) is managed by a mediator, ensuring that inputs from the user are processed and corresponding outputs are displayed efficiently and correctly.
- **Iterator Pattern**: The system allows traversal through collections of objects, such as iterating over citizens, buildings, or transportation vehicles, to perform operations like calculating happiness or resource consumption.
- **Composite Pattern**: Buildings are organized into a tree-like structure where each node represents a building section (e.g., residential, commercial). This allows entire sections of the city to be managed as a unified whole while maintaining individual control over specific buildings.
- **Abstract Factory Pattern**: The system uses abstract factories to create families of buildings with similar functionality, such as factories for different types of utilities (power plants, water plants) or residential complexes, ensuring that buildings are created according to their specific roles.

- **Singleton Pattern**: There is only one instance of the city, ensuring that all systems and operations refer to the same central city object, preventing inconsistencies and duplicated cities within the simulation.
- **Observer Pattern**: The system monitors the happiness of citizens by using observers. Any changes in factors like taxes, transport, or housing trigger updates that reflect changes in citizen satisfaction in real-time.
- **Prototype Pattern**: Citizens can replicate or "procreate" by creating new citizens based on an existing citizen's attributes. This allows the simulation to efficiently generate new citizens with similar properties while avoiding the overhead of creating them from scratch each time.

Task 2.4 (Classes)

Building + BuildingFactory:
    Residential + ResidentialFactory :
        -House
        -Apartment
        -Complex
    Commercial + CommercialFactory:
        -Shop
        -Office
        -Mall
    Industrial + IndustrialFactory:
        -Forestry
        -SteelFactory
        -ConcreteFactory
    Service + ServiceFactory:
        -Hospital
        -TownHall
        -School
        -TrainStation
        -Airport
        -PoliceStation
    Utilities
        -PowerPlant
        -WaterPlant
        -WasteManagement
    Landmark + LandmarkFactory:
        -Park
        -Museum

Vehicle:
- Car
- Taxi
- Bus (government)
- Plane
- Train
    - Freight
    - Passenger

Road:
- TarRoad
- RailRoad

Household

Government
- Taxes
- Resources
- State
    - OverPopulation
    - LowBudget
    - HighDisatisfaction
- Strategy
    - IncreaseTaxes
    - ExpandCity
- Department
    - Transport
    - Amenities
    - Labour
    - Finances

Tax
- PropertyTax
- BusinessTax
- IncomeTax

Population
- Citizen
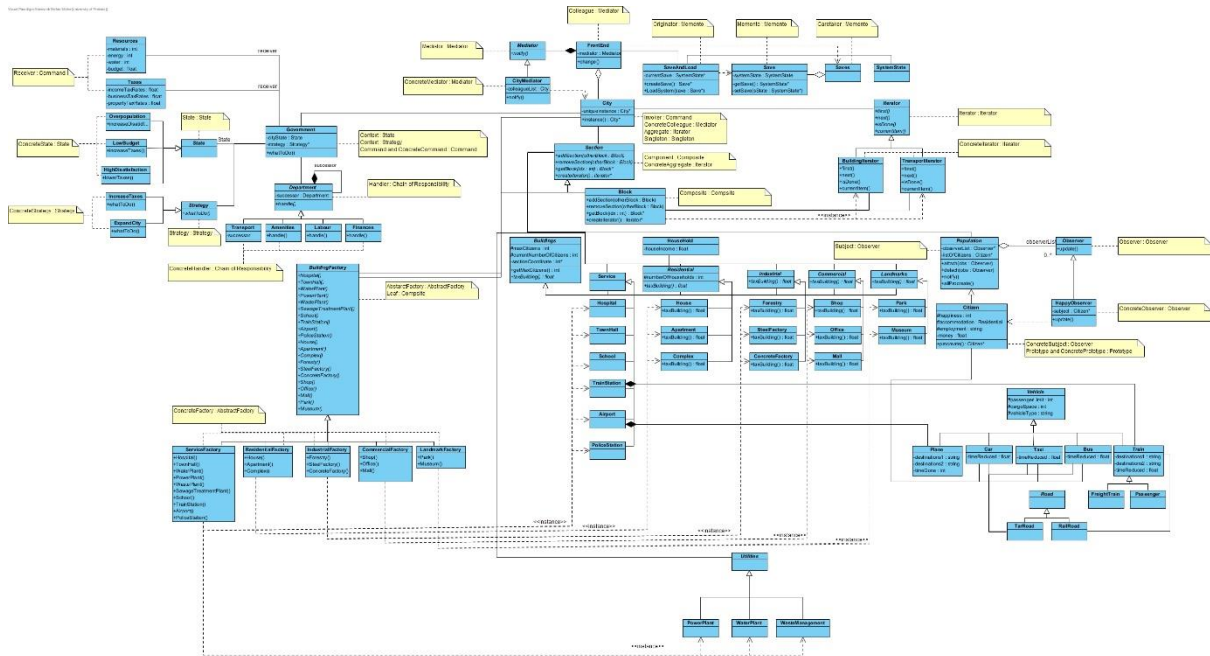
Observer
- HappyObserver

SaveAndLoad
- Save
- Saves
- SystemState

Frontend
        -Mediator
        -CityMediator
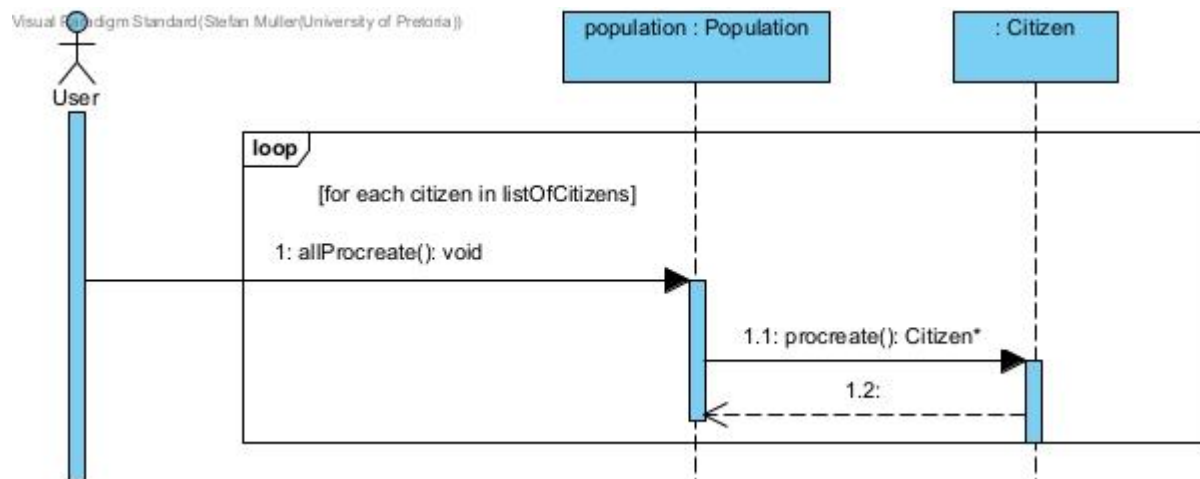   -City
                -Sections
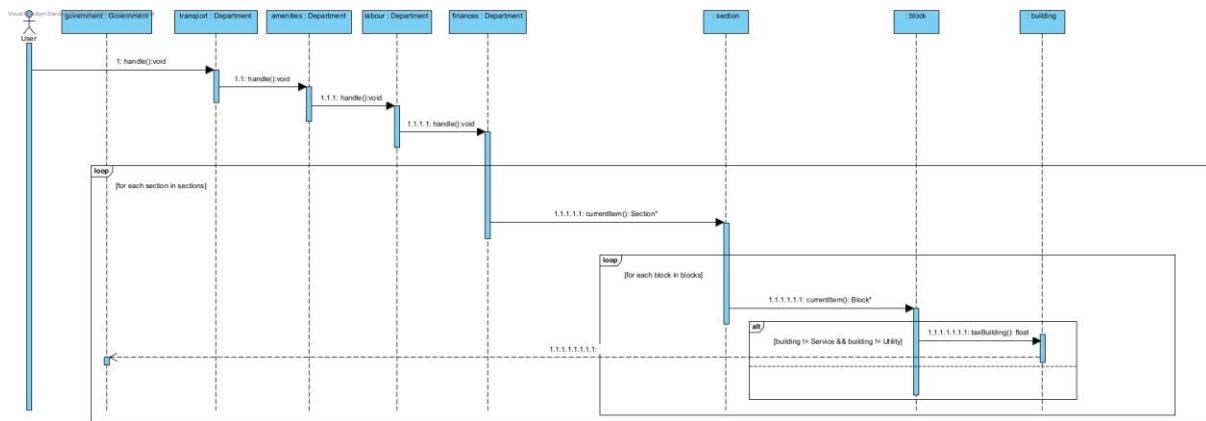   -Block

Task 2.5 (Class Diagram)

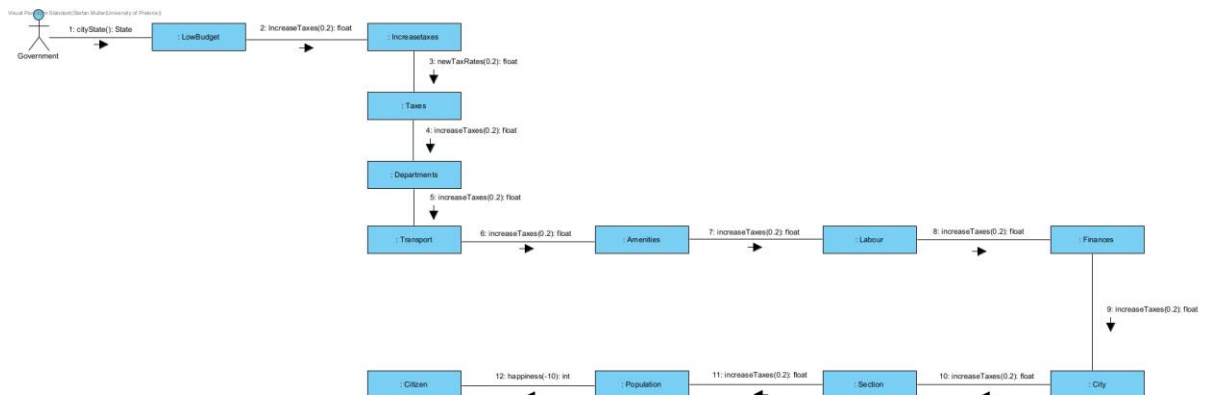# Task 2.6 (Sequence and Communication Diagrams)

Population Growth:

Tax Collection:

Government and Citizen:

Task 2.7 (State Diagram)

Citizen Happiness:

Citizen happiness

Can procreate

Happy

Happiness >= 75

Happiness >= 75

Happiness < 75

Unhappy

75 > Happiness >= 50

75 > Happiness >= 50

Angry

50 > Happiness >= 25

50 > Happiness >= 25

No longer recognizes authority

25 > Happiness
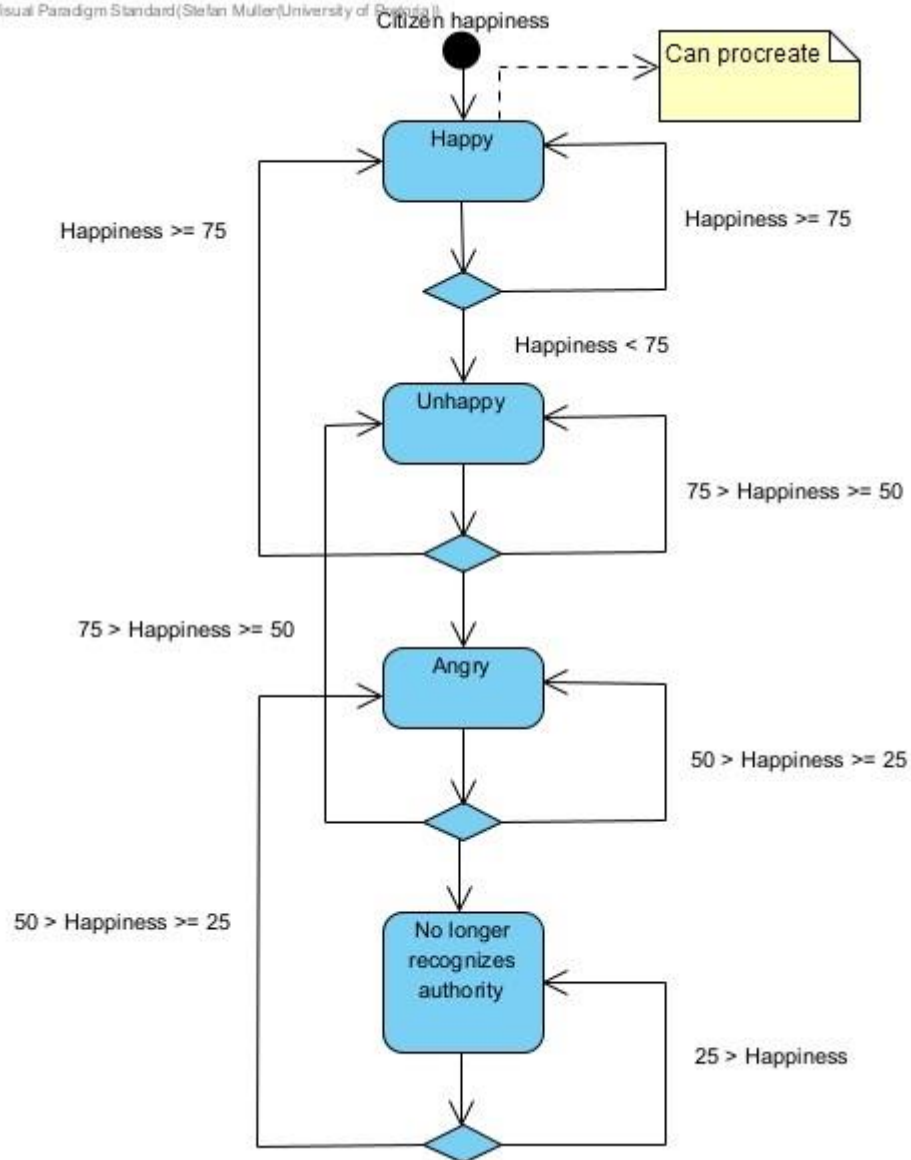
City Power:
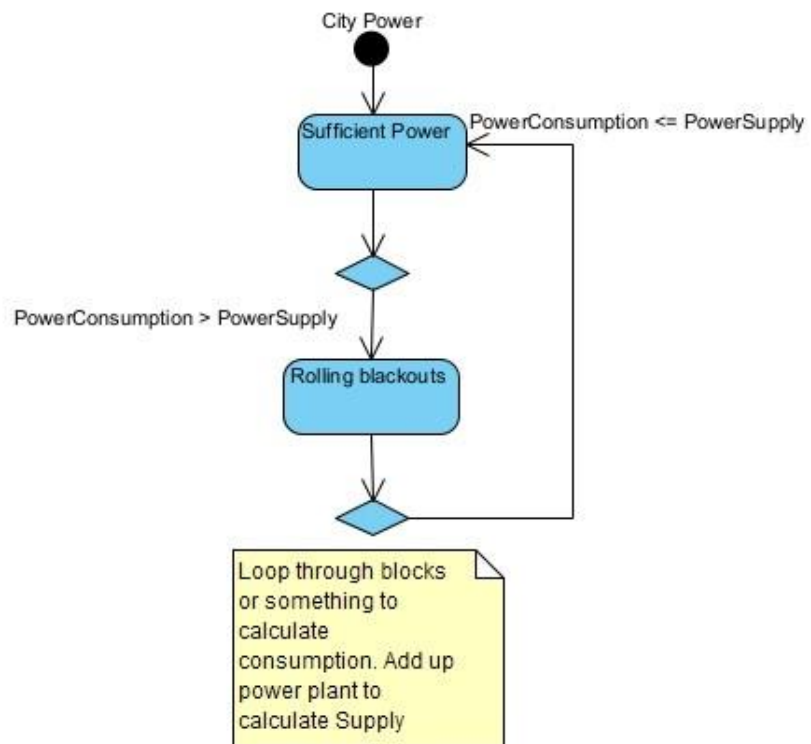
Population:
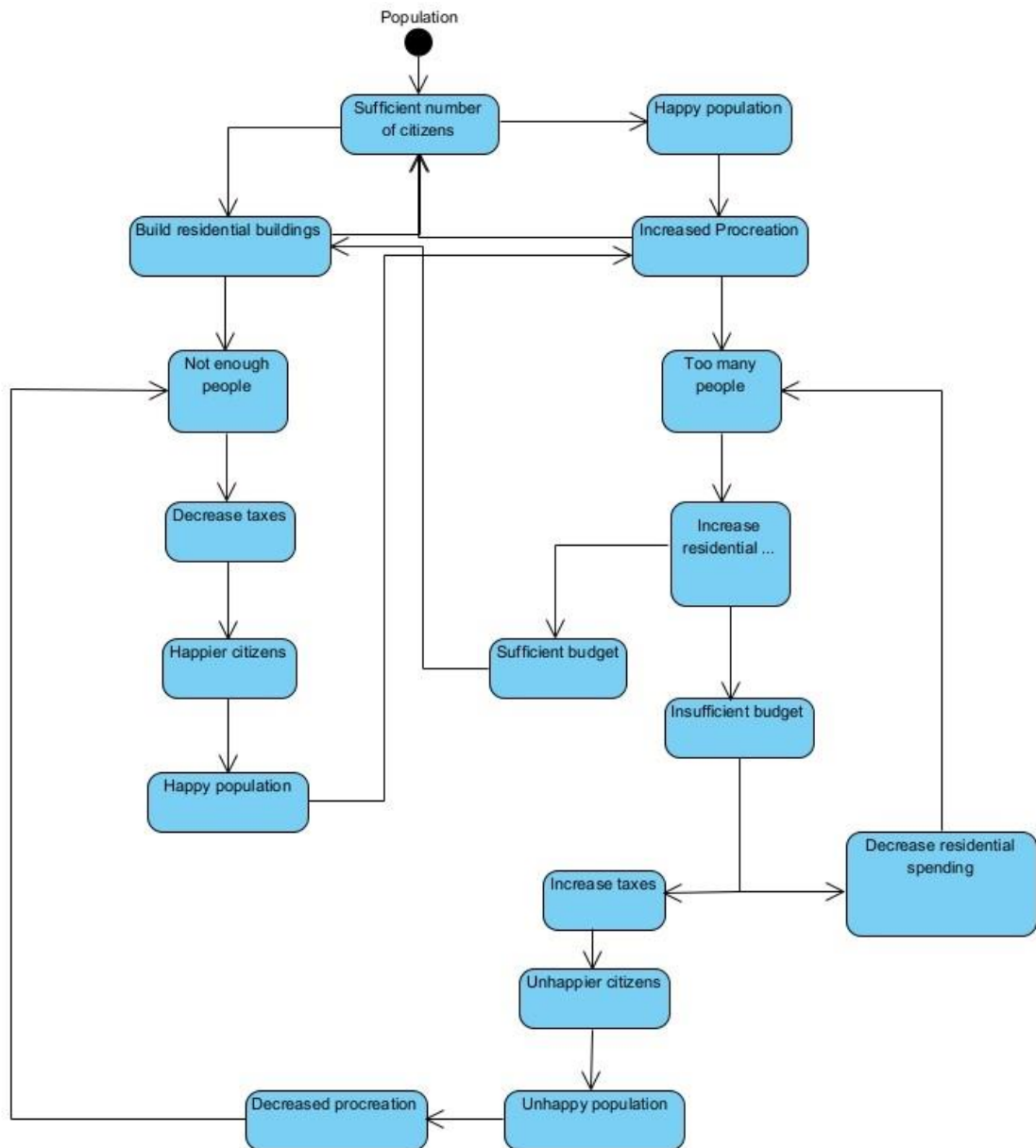
Task 2.8 (Object Diagram)

Chain of Responsibility:

cityGovernment : Government

strategy = increaseTaxes
cityState = lowBudget

brokeAFcity : LowBudget

cityResources : Resources

budget = 100000.00
energy = 50000
materials = 300000
water = 1000

increaseIncomeTax : IncreaseTaxes

incomeTax : Taxes

incomeTaxRates = 0.2
businessTaxRates = 0.3
propertyTaxRates = 0.05

governmentDepartments : Department

departmentOfTransport : Transport

successor = governmentDepartments

departmentOfAmenities : Amenities

successor = departmentOfTransport

departmentOfLabour : Labour

successor = departmentOfAmenities

departmentOfFinances : Finances

successor = departmentOfLabour

Observer:

steelFactoryWorkers : Population

listOfCitizens = steelWorkers[2]
observerList = HappyObserver

steelWorkers[2] : Citizen

steelWorkers[1] : Citizen

happiness = 80
accommodation = vacationHome
employment = SteelFactory Worker
money = 50000.00

moodDetector : HappyObserver

subject = steelWorkers[2]

vacationHome : House

numberOfHouseholds = 1

cutleryFactory : SteelFactory

mercedesBenz : Car

cargoSpace = 10
passengerLimit = 6
vehicleType = sedan