COS 214 PROJECT
City Builder Simulation

GitHub Repo:

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Functional Requirements:

The City Simulator will:

Provide a City Manager:

- Select Simulation Modes
- Initialize City Zones and layouts
- Control *Resource Allocation* to *Buildings*
- Let players make economic decisions
- Allow players to inspect *buildings* and *zones*

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Functional Requirements:

The City Simulator will:

Provide *Citizens* that:

- Have unique identifiers & demographics
- Can be employed in *buildings*
- *Pay taxes* based on income
- Have variable *satisfaction levels*
- Consume city resources
- Use transport networks

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Functional Requirements:

The City Simulator will:

Provide *Buildings* that:

- Can be constructed & have different types such as:
  - Commercial, Residential, Industrial, Public Service, Landmark
- Require maintenances and resources
- House citizens (residential)
- Provide employment (commercial/industrial)
- Generate tax revenue
- Have capacity limits

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Functional Requirements:

The City Simulator will:

Provide *Buildings* that:

- Affect citizen satisfaction based on their creation/change in state
- Form the city infrastructure.

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Functional Requirements:

The City Simulator will:

Provide a *Government* that:

- Collects taxes from citizens
- Manages city budget
- Implements city policies
- Responds to city needs
- Maintains infrastructure

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Functional Requirements:

The City Simulator will:

Provide *Resources* that:

- Supply essential utilities (Water, Power, Waste Management)
- Have production, consumption
- Require storage facilities
- Need distribution to *buildings*
- Affect building operations
- Impact citizen satisfaction

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Functional Requirements:

The City Simulator will:

Provide a Finance Department that:

- Manages tax collection
- Is used by the government to manage city finances/funds
- Handle fund allocation to different components

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Functional Requirements:

The City Simulator will:

Provide a Transport Manager that:

- Support different vehicle types
- Manage route systems
- Handle capacity allocation

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Functional Requirements:

The City Simulator will:

Provide a Utilities Manager that:

- Controls resource production
- Manages distribution of resources
- Monitors resource levels

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Assumptions & Design Decisions:

Assumptions made during the design process include:

- Citizens value proximity to essential services and employment.
- Improved transport networks that reduce congestion.
- A balanced mix of residential and commercial zones leads to economic vitality.

Understanding urban development and city management principles is crucial for creating sustainable and vibrant urban spaces. Each component plays a vital role in shaping the city's future.

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Design Patterns used (/functional requirement):

Buildings:

1. State Pattern (manages building states: UnderConstruction, Damaged, Operational)
2. Abstract Factory Pattern (for creating different building types: residential, commercial, industrial, landmarks)

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Design Patterns used (/functional requirement):

Utilities:

1. Factory Method Pattern (for creating power plants, water supply, waste management, sewage systems)
2. Mediator Pattern (coordinating resource distribution)
3. Command Pattern (for utility operations start/stop)
4. Prototype Pattern (for infrastructure expansion)

design wits
UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Design Patterns used (/functional requirement):

Transportation:

1. Strategy Pattern (for different transport methods: roads, public transit, trains, airports)
2. Iterator Pattern (for route network navigation)
3. Builder Pattern (for complex transport system creation)
4. Chain of Responsibility (for route management)

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Design Patterns used (/functional requirement):

Citizens:

1. Observer Pattern (for receiving tax notifications) Iterator Pattern (for route network navigation)
2. Decorator Pattern (for different citizen roles and capabilities)
3. Prototype Pattern (for population growth)

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Design Patterns used (/functional requirement):

Government:

1. Command Pattern (implementing policies)
2. Observer Pattern (as tax collection notifier)
3. Chain of Responsibility (processing requests)
4. Facade Pattern (managing different departments)

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Design Patterns used (/functional requirement):

Resources:

1. Mediator Pattern (coordinating resource distribution)
2. Factory Method Pattern (creating resource facilities)
3. Prototype Pattern (for resource facility expansion)

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Design Patterns used (/functional requirement):

Taxes:

1. Observer Pattern (tax collection notifications)
2. Facade Pattern (through FinanceDepartment)

design wits
UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Design Patterns used (/functional requirement):

City Growth:

1. Abstract Factory Pattern (for expanding different zones)
2. Builder Pattern (for complex infrastructure expansion)
3. Mediator Pattern (coordinating growth across systems)

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Design Patterns used (/design pattern):

State:

The State pattern enables a Building to modify its behavior through changeState() and handle() functions based on its current state (UnderConstruction, Damaged, or Operational), where each state class implements specific behavior for common operations, making the Building appear to change its class when transitioning between states.

Class	Participant
Building	Context
State	State
UnderConstruction , Operational , Damaged	Concrete State

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Design Patterns used (/design pattern):

Abstract Factory:

The Abstract Factory pattern is applied to manage building creation through a hierarchy of specialized factories. The Building (AbstractFactory) interface defines the common creation method (createBuilding), while concrete factories (ResidentialFactory, CommercialFactory, IndustrialFactory, LandmarkFactory, PublicServiceFactory) handle the creation of their specific building types. This pattern ensures that buildings are created consistently within their categories (e.g., houses through HouseFactory under ResidentialFactory), enforces zoning rules, and maintains clean separation between different building types while allowing easy addition of new building categories without modifying existing code.

Class	Participant
ResidentialFactory , IndustrialFactory , CommercialFactory , PublicServiceFactory , LandmarkFactory	AbstractFactory
HouseFactory , ApartmentFactory , TownhouseFactory , FactoriesFactory , PowerplantFactory , WarehouseFactory , ShopFactory , MallFactory , OfficeFactory , SchoolFactory , PoliceStationFactory , MedicalCenterFactory , ParkFactory , MonumentFactory , CulturalCenterFactory	ConcreteFactory
Residential , Industrial , Commercial , PublicService , Landmark	AbstractProduct
House , Apartment , Townhouse , Factories , Powerplant , Warehouse , Shop , Mall , Office , School , PoliceStation, MedicalCenter , Park , Monument , CulturalCenter	ConcreteProduct

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Design Patterns used (/design pattern):

Factory Method:

The Factory Method pattern has been implemented to manage two crucial aspects of city infrastructure. First, it handles basic utility creation through the UtilityFactory hierarchy, where specialized factories create specific infrastructure components - PowerPlantFactory for power generation facilities, WaterFactory for water supply systems, WasteFactory for waste management facilities, and SewageFactory for sewage treatment plants.

Second, it manages energy source diversity through the EnergyFactory hierarchy, with specialized factories (HydroFactory, WindFactory, NuclearFactory, and CoalFactory) creating different types of power generation facilities to meet the city's varying energy needs.

Class	Participant
UtilityFactory	Creator
PowerPlantFactory , WaterFactory , WasteFactory ,SewageFactory	ConcreteCreator
Utility	Product
UtilityPowerPlant , WaterSupply, WasteManagement ,SewageSystems	ConcreteProduct

Class	Participant
EnergyFactory	Creator
HydroFactory , WindFactory ,NuclearFactory , CoalFactory	ConcreteCreator
EnergySource	Product
HydroSource , WindSource ,NuclearSource , CoalSource	ConcreteProduct

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Design Patterns used (/design pattern):

Command:

The Command pattern functions within two vital control systems in the city simulation. The utility control system uses UtilityManager as an invoker to execute StartCommand and StopCommand operations, managing the activation and deactivation of various utility services through the Utility receiver

The government policy system employs the Government class as an invoker to implement PublicServicePolicies and EconomicPolicies, with these commands directly affecting the city's population through the CitizenInterface receiver. This dual implementation provides flexible control over both infrastructure operations and policy implementation.

Class	Participant
UtilityManager	Invoker
Command	Command
StartCommand , StopCommand	ConcreteCommand
Utility	Receiver

Class	Participant
Government	Invoker
Policies	Command
PublicServicePolicies , EconomicPolicies	ConcreteCommand
CitizenInterface	Receiver

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Design Patterns used (/design pattern):

Prototype:

The Prototype pattern serves the city's need for rapid infrastructure expansion and population growth. In the utility sector, it enables quick cloning of existing infrastructure including UtilityPowerPlant, WaterSupply, WasteManagement, and SewageSystems, allowing for efficient expansion of city services.
In the population management aspect, it facilitates the creation of new citizens by cloning existing Citizen and CitizenType templates, streamlining the process of population growth and demographic diversification.

Class	Participant
Utility	Prototype
UtilityPowerPlant , WaterSupply, WasteManagement ,SewageSystems	ConcretePrototype

Class	Participant
CitizenInterface	Prototype
Citizen , CitizenType	ConcretePrototype

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Design Patterns used (/design pattern):

Chain of Responsibility:

The Chain of Responsibility pattern manages two key processing chains within the city. The route management system uses RouteNode as a base handler with BestRouteNode and AccessibleRoute concrete handlers to optimize city navigation and ensure accessible pathways.
The government operations chain, built with Government as the base handler and specialized processors like UtilitiesSector, FinanceSection, and Main, efficiently routes citizen requests through appropriate departments, ensuring each concern is addressed by the most suitable authority.

Class	Participant
RouteNode	Handler
BestRouteNode , AccesibleRoute	ConcreteHandler

Class	Participant
Government	Handler
UtilitiesSector , FinanceSection , Main	ConcreteHandler
CitizenInterface	Client

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Design Patterns used (/design pattern):

Iterator:

The Iterator pattern has been specifically applied to the city's transportation network, using RouteNode as the aggregate interface with BestRouteNode and AccessibleRoute as concrete implementations. The MapIterator provides systematic navigation through the city's route network, enabling efficient pathfinding and transportation planning while keeping the underlying route structure encapsulated.

Class	Participant
RouteNode	Aggregate
BestRouteNode , AccesibleRoute	ConcreteAggregate
MapIterator	Iterator / ConcreteIterator

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Design Patterns used (/design pattern):

Strategy:

The Strategy pattern addresses the city's transportation needs through the TravelManager context, which coordinates different transport strategies including Train, Vehicle, and Plane. This implementation allows citizens to seamlessly switch between different transportation methods while maintaining consistent travel management interfaces throughout the city.

Class	Participant
TravelManager	Context
Transport	Strategy
Train ,Vehicle , Plane	ConcreteStrategy

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Design Patterns used (/design pattern):

Builder:

The Builder pattern manages complex transport system creation through the TravelManager director and TransportBuilder interface. Specialized builders (PlaneBuilder, TrainBuilder, VehicleBuilder) handle the intricate process of creating different transportation options, ensuring each transport type is constructed correctly and consistently while maintaining flexibility in the construction process.

Class	Participant
TravelManager	Director
TransportBuilder	Builder
PlaneBuilder , TrainBuilder , VehicleBuilder	ConcreteBuilder
Train , Vehicle , Plane	Product

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Design Patterns used (/design pattern):

Mediator:

The Mediator pattern centralizes resource management through Utility, which coordinates interactions between various city resources including power, water, revenue, waste management, and sewage systems. This central coordination point simplifies complex resource interdependencies and ensures efficient resource allocation throughout the city.

Class	Participant
Utility	Mediator / ConcreteMediator
Resources	Colleague
powerResource , waterResource , revenueResource , wasteManagementResource , sewageManagementResource	ConcreteColleague

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Design Patterns used (/design pattern):

Observer:

The Observer pattern has been specifically implemented to handle the city's tax collection notification system. The Government class acts as the subject (publisher) that monitors when tax collection periods begin, while the CitizenInterface serves as the observer interface. Each Citizen, as a concrete observer, is automatically notified by the Government when it's time to collect taxes. This pattern ensures efficient tax collection by automatically informing all registered citizens when they need to pay their taxes, eliminating the need for manual notification systems or periodic checks. The implementation streamlines the tax collection process by maintaining a direct communication channel between the Government and all citizens for tax-related matters.

Class	Participant
Government	Subject / ConcreteSubject
CitizenInterface	Observer
Citizen	ConcreteObserver

design wits
UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Design Patterns used (/design pattern):

Decorator:

The Decorator pattern enhances citizen functionality by using CitizenInterface as the base component and adding specialized behaviors through CitizenType decorators. The concrete decorators EmployedCitizen and PropertyOwner add specific responsibilities and privileges to citizens, allowing for dynamic modification of citizen capabilities based on their roles in the city.

Class	Participant
CitizenInterface	Component
Citizen	ConcreteComponent
CitizenType	Decorator
EmployedCitizen ,PropertyOwner	ConcreteDecorator

design wits
UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Design Patterns used (/design pattern):

Facade:

The Facade pattern simplifies access to complex subsystems through two main facades. The finance system uses FinanceDepartment to provide a unified interface to various taxation and budget systems.

The city control system employs CityController to coordinate CitizenController, BuildingController, and UtilitiesController. These facades reduce system complexity and provide clear access points for managing various city operations.

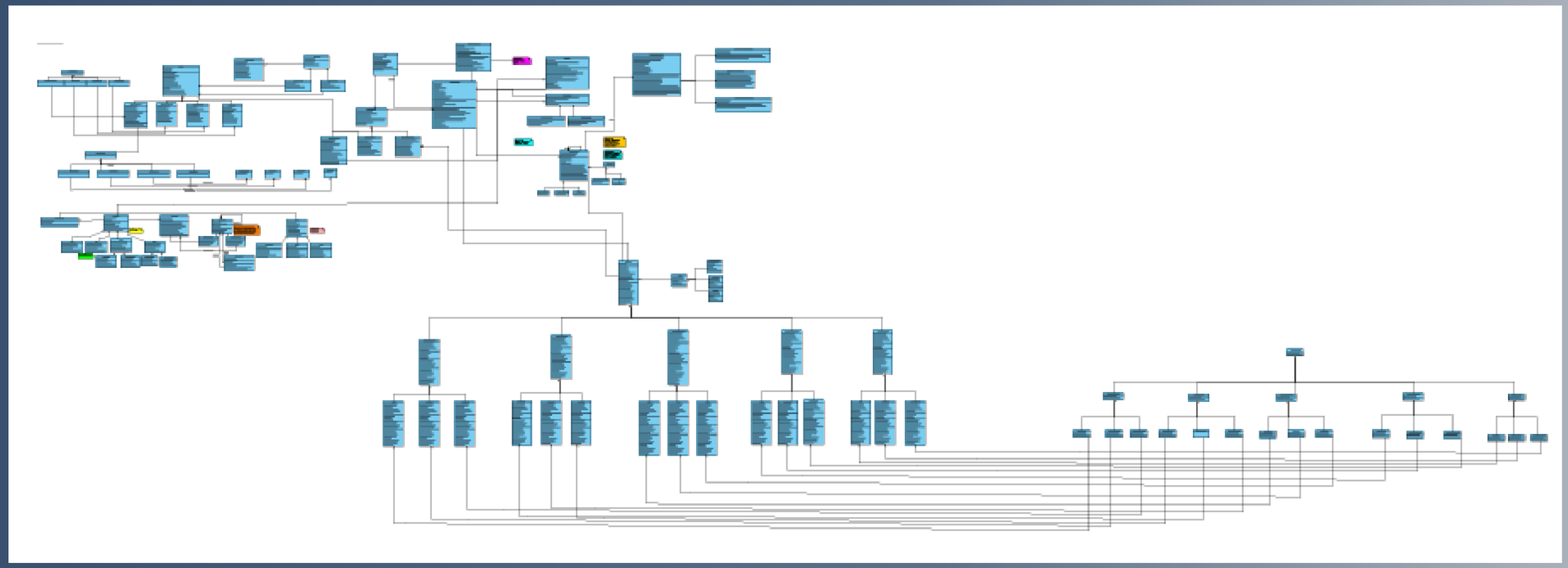
Class	Participant
Government	Client
FinanceDepartment	Façade
CommercialTaxationSystem , ResidentialTaxationSystem , BudgetAllocationSystem	Subsystem

Class	Participant
CityController	Façade
CitizenController , BuildingController , UtilitiesController	Subsystem

design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

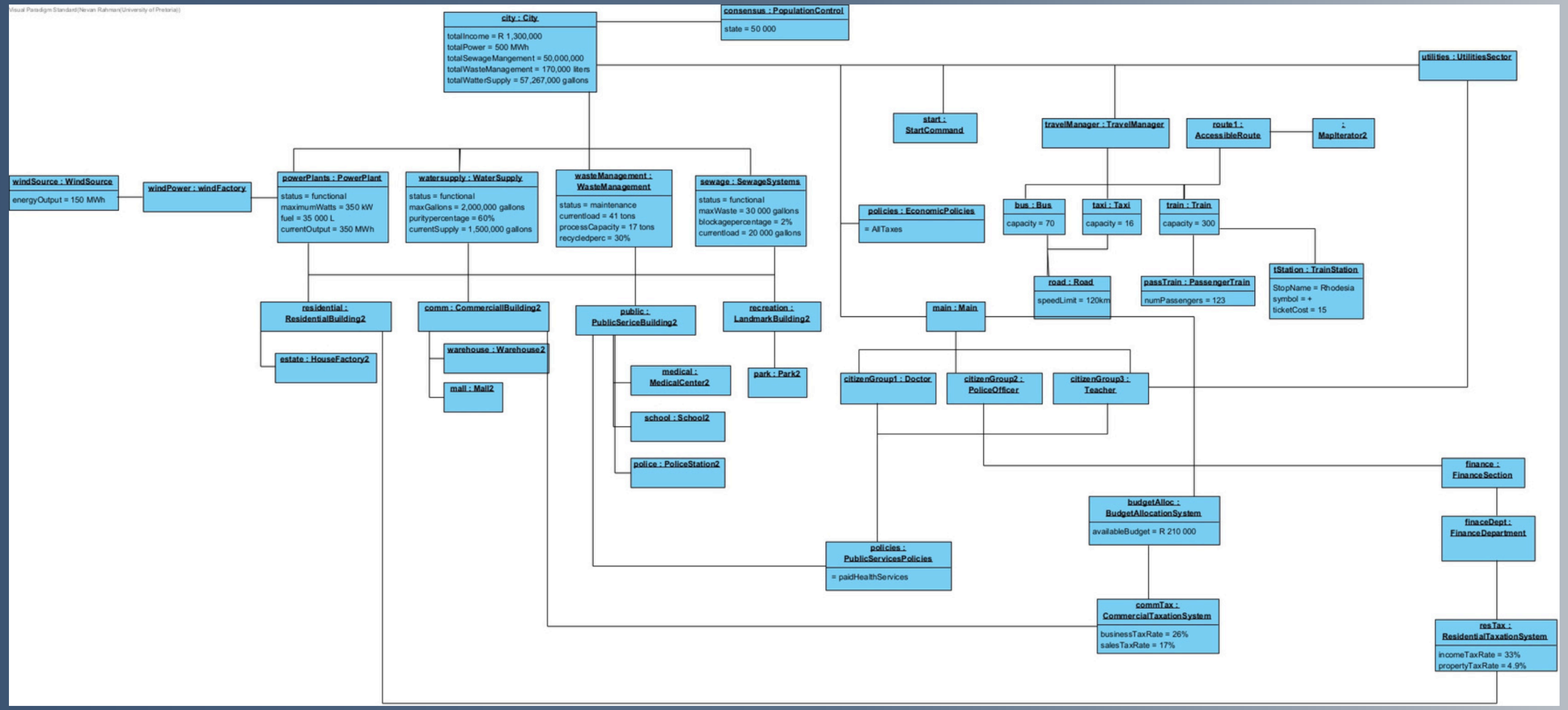
Class Diagram:



design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Object Diagram:

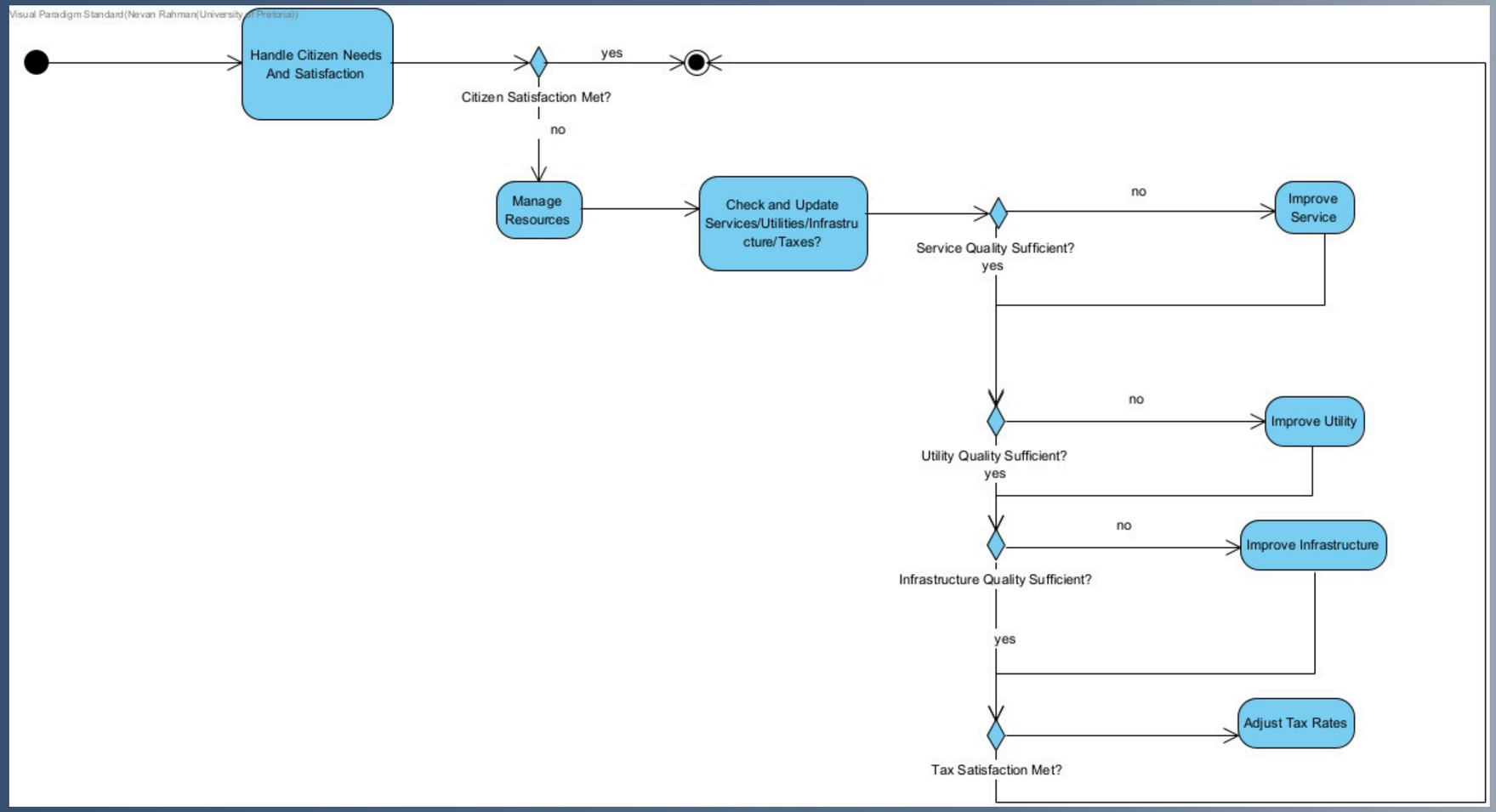


design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Activity Diagram:

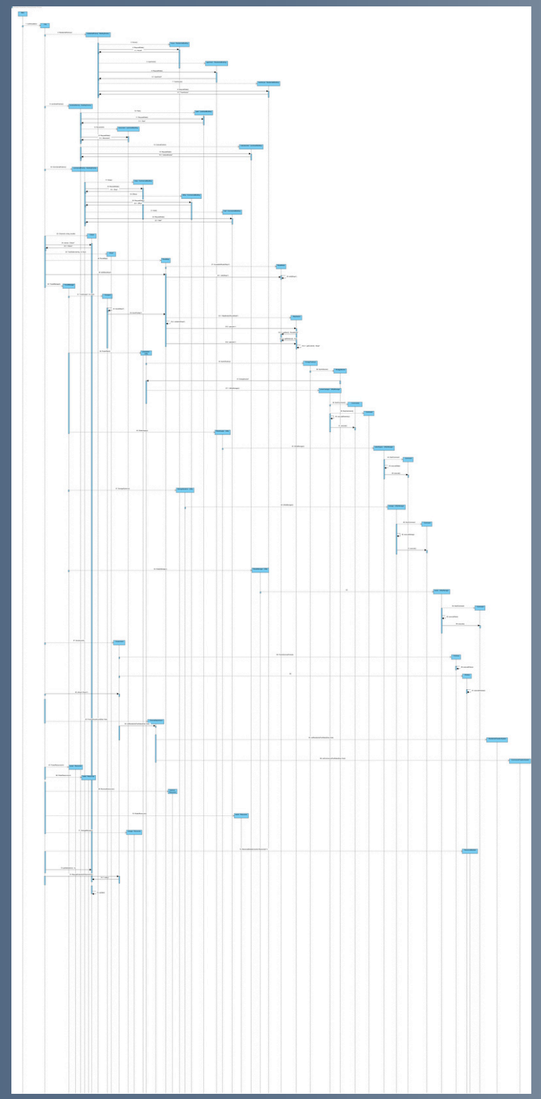
Handling Citizen Needs and Satisfaction



design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

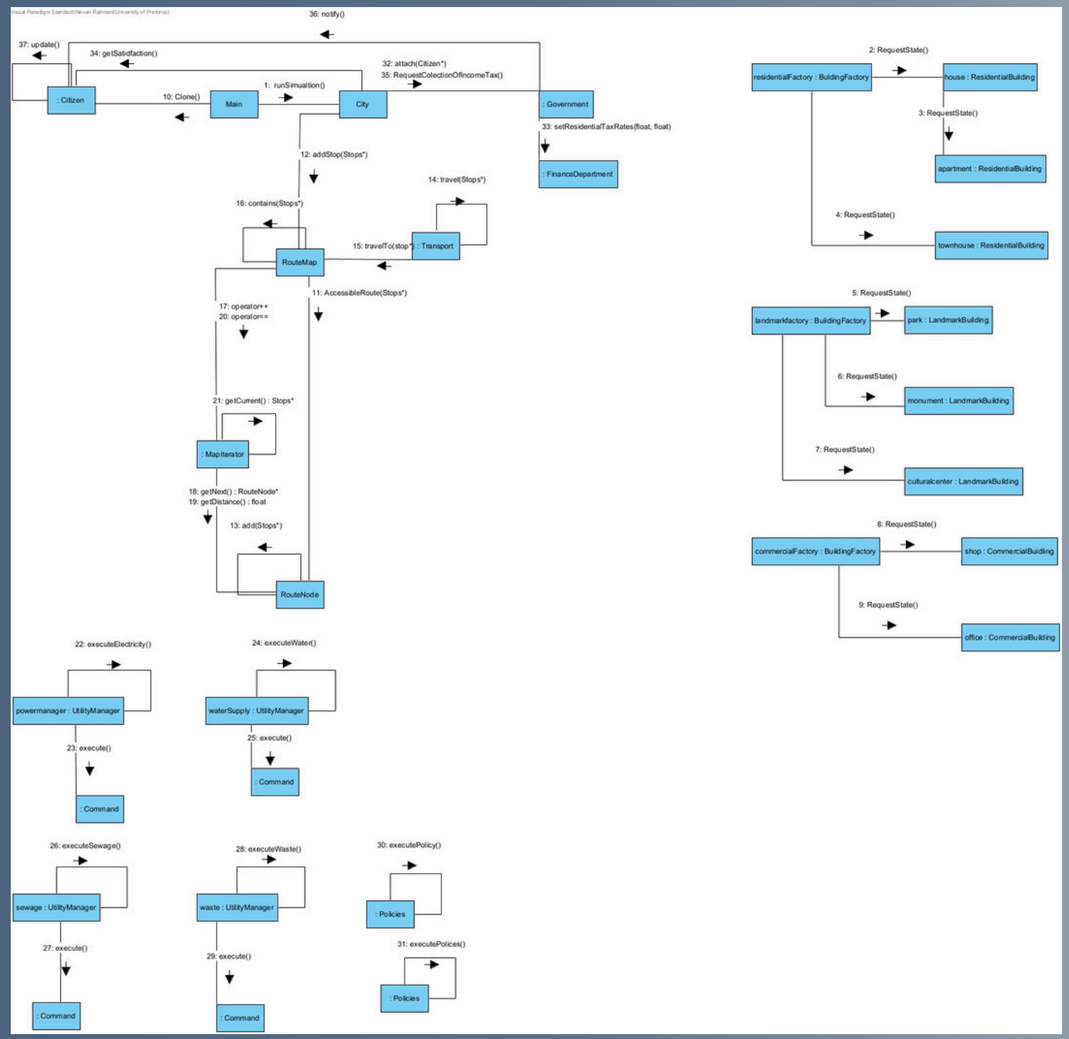
Sequence Diagram:



design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

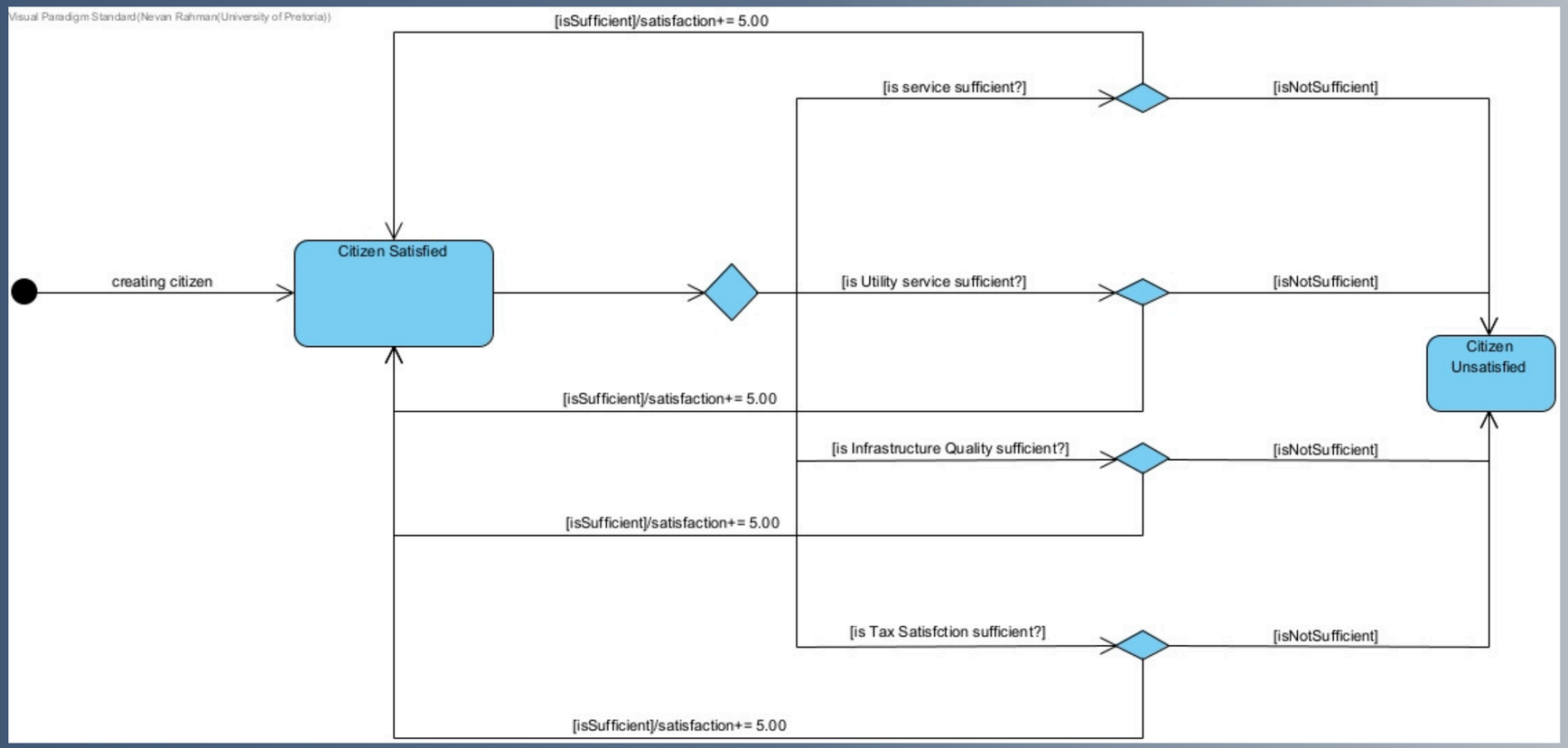
Communication Diagram:



design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

State Diagram:



design wits

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA