COS 214 REPORT

Name	**Student Number**	**Github**	**Role**
Mr. Shayden Naidoo	u23599236	@ShaydenNaidoo	Team lead and Taxes: Financial and Taxation Strategist
Miss. Zainab Abdulrasaq	u22566202	@abdulrasaqzainab	Buildings: Infrastructure and Building Manager
Mr. Navendran Naidoo	u21512494	@Naven13	Resources: Resource Management Specialist
Mr. Samvit Prakash	u23525119	@Sammy-2903	Transportation: Transportation Systems Coordinator
Mr. Arnaud Strydom	u23536013	@blizs	Citizens and game dev: Citizen Welfare and Demographics Analyst and front end Game developer
Mr. Neo Machaba	u23002167	@NeoMachabaUP	Utilities: Utilities and Service Manager
Mr. Rudolph Lamprecht	u20598425	@RudolphLamp	Government: City Governance Manager

System requirements:

Functional Requirements:

Core Functionality:

Building Management:

- Players can add,upgrade and remove buildings
- There are different types of buildings: Commercial,Residential,Industrial and Landmark buildings can be created using their respective factories

- Buildings can be interacted with by the player e.g when the player clicks on a building they can obtain information about the building and its residents or citizens that use the building. Ie. when a residential building is clicked on the the player can collect the taxes of its residents.
- Functional requirement: Need to manage the construction of a building , provide tracking for its construction state, resource consumption and the services and taxes it produces.

Utility Management:

- Utilities (e.g. water, electricity, waste management) must be provided to specified buildings.
- Utility consumption needs to be tracked based on the number of buildings and their types.
- Functional Requirement: Ensure that all buildings have sufficient utility services, with the ability to track shortages or surpluses.

Transportation Systems:

- The system must simulate transportation networks, such as roads, public transport, and other transit options like buses, taxis, and trains. These transportation systems should influence the efficiency of city mobility and economic activity.
- Road Infrastructure:
  The player must be able to build and upgrade roads to support vehicle movement between different parts of the city.
- Public Transit:
  The system should simulate public transportation options (e.g., buses, taxis, subways), which citizens can use to travel to workplaces, commercial centers, or homes.
- Traffic and Congestion Management:
  The system must track traffic flow and congestion in real-time, giving feedback to the player on where road upgrades or public transport improvements are needed.
- Impact on Citizen Satisfaction:
  Poor transportation systems should reduce citizen satisfaction and increase commute times, while well-developed systems should improve satisfaction and productivity.

- City Connectivity:
  Ensure the transportation network impacts the efficiency of economic activity and citizen mobility across residential, commercial, and industrial zones.

Citizen Simulation:

- The system should simulate citizens as dynamic entities that interact with the city's buildings, services, and policies. Citizens should respond to changes in employment, housing, services, and taxation, influencing city growth and development.
- Employment:
  Citizens should seek employment in available jobs across commercial and industrial buildings, and the system must track employment status (employed, unemployed).
- Satisfaction:
  Track citizen satisfaction based on access to services (e.g., healthcare, education), job availability, housing, and taxation. Higher satisfaction should lead to population growth, while dissatisfaction could lead to emigration.
- Population Growth:
  The system must simulate population growth based on birth rates, migration, and job availability. The player must be able to influence this through policy changes, building new residential areas, or improving city services.
- Tax Contribution:
  Citizens must pay taxes, and their willingness to pay should be tied to their satisfaction level. Refusal to pay taxes due to dissatisfaction could result in penalties for the city's budget.
- Residential Behavior:
  Citizens should interact with residential buildings, influencing demand for housing based on population growth or shrinkage. Overcrowding or housing shortages should reduce satisfaction.

Government System:

- The system must allow the player, acting as the government, to control and manage city policies, taxation, public services, and budget allocations. The government system should track the effectiveness of these policies on the city's economy, citizen satisfaction, and overall growth.
- Tax Policies:
  The player should set various tax rates (e.g., income tax, sales tax, property tax)

for citizens and businesses. The impact of tax policies should influence citizen satisfaction, economic activity, and the government's budget.

- Budget Allocation:
  The system must track the city's budget, allowing the player to allocate funds to various services like healthcare, education, law enforcement, and infrastructure development.
- City Services Management:
  The player must be able to manage public services such as healthcare, education, and security, with the quality of services impacting citizen satisfaction and city development.
- Policy Implementation:
  The player can implement policies that affect different city sectors (e.g., reducing pollution, encouraging green energy). The effects of these policies must be tracked and should influence the growth and satisfaction of citizens.
- Economic Monitoring:
  The system should provide economic reports on the city's financial health, including tax income, expenditures, deficits, and debts, allowing the player to adjust policies as necessary.
- Citizen Interaction:
  Citizens must respond to government decisions, particularly around taxation, public service quality, and city planning. High taxes with poor services should lead to dissatisfaction, while well-managed services can improve citizen loyalty and productivity.

Resource Management:

- The system must track and allocate the city's resources, such as building materials (wood, steel, concrete), water, electricity, and waste management.
- Waste management, including recycling, should contribute to resource allocation by reintroducing recycled materials into the resource pool.
- Resource consumption should be linked to city growth, building construction, and citizen needs.
- Track the supply and demand of resources for all buildings and infrastructure (e.g., water, electricity, wood, steel, concrete).
- Monitor shortages and surpluses in resource availability and provide reports to the player.
- Alert players when critical shortages occur and suggest corrective actions (e.g.,

build new water plants or increase recycling capacity).
- Track waste generation for each building (residential, commercial, industrial) based on usage.
- Recycle materials like plastic, metal, and glass from waste to convert them into usable resources (e.g., recycled steel or concrete).
- Reduce raw material consumption by incorporating recycled materials in new constructions and maintenance.
- Provide a Recycling Rate Report showing the percentage of successfully recycled materials and total recovered resources.
- Allocate resources dynamically as new buildings or infrastructures are constructed, factoring in available resources and waste-recovery contributions.

City Growth:

- The city grows dynamically based on factors like population increase,available services and building expansion.
- New land becomes available for expansion as the population increases.
- Functional requirement: model the dynamic expansion as the city population increases and the building need increases with it.

Taxation and Finance:

- The user represents the government and runs the city, so the user needs to be able to collect taxes and allocate those funds towards the construction of buildings ,transportation,utilities or public services like a normal government.
- A financial accounting just system to track tax income and expenditure needs to be implemented
- There are different types of tax that need to be charged to specific types of citizens e.g normal citizens pay income tax while business owners pay both income tax and sales tax.
- Impact of the tax rates on the citizens needs to be tracked and this satisfaction percentage must be shown to the user to allow the user to adjust the taxes accordingly to keep their citizens happy and avoid citizens refusing to pay tax due to dissatisfaction.
- Functional requirement: We need to implement a taxation system that allows players to adjust tax rates and manage the finances of the city. Depending on the

tax rates that the player sets can determine citizen satisfaction and how many citizens are happy to pay their taxes on time.

Interactions between components:

Government,Citizens and Taxes:

- The government sets policies and tax rates that influence citizen happiness, population growth and economic activity.
- Citizen satisfaction affects the overall growth and development of the city.
- Interaction: The government sets tax rates and the citizens are required to pay these taxes after a certain amount of time when they are ready to pay their taxes the government collects these taxes and allocates it towards funding various city projects this then increases the citizen satisfaction for paying their taxes and can increase the rate at which citizens produce tax money.

Subsystems to Split Functional Requirements

- Taxation Subsystem:
    - Manages the collection of taxes and adds them to the city fund
    - Manages the allocation of the city's budget to different services.
    - Ensures that taxes support city growth without negatively impacting citizen satisfaction.
    - Tracks citizen satisfaction regarding tax rates
    - Allows the government to set tax rates and policies for specific types of tax

Non-Functional requirements:

- The simulation needs to be scalable, maintainable, and flexible.
- Must be able to update the system and add new features to it easily.

Research Brief:

Introduction to Urban Development:

Urban development plays a crucial role in shaping sustainable, economically vibrant, and inclusive urban spaces. With a growing global population and increasing migration to cities, the demand for effective city management has intensified. Successful urban development hinges on the planning of residential, commercial, and industrial areas, alongside the provision of essential services and resources. By focusing on sustainability, economic development, and citizen satisfaction, cities can balance growth with quality of life.

City Management Principles:

Effective city management relies on principles that balance growth, resource allocation, and citizen satisfaction. Key management principles include:

- **Sustainable Growth:** This involves managing resources—such as energy, water, and waste—in a way that supports current needs without compromising future city needs.
- **Citizen-Centered Services:** Ensuring essential services, such as healthcare, education, and security, is vital to maintaining citizen satisfaction and quality of life.
- **Economic Development:** By creating opportunities for employment and business growth, cities support job availability and economic stability, which are crucial for both individual citizens and the overall urban environment.
- **Infrastructure Management:** Efficient infrastructure, particularly in transportation and utilities, is essential to ensure mobility and resource distribution.
- **Governance and Policy:** Governments implement policies on taxation, zoning, and public services that affect every aspect of city life, influencing economic growth and citizen satisfaction.

Role of City Components:

The simulation's components are designed to reflect the roles of various elements in a real city, ensuring a cohesive and dynamic city model:

- **Buildings:** Residential, commercial, and industrial buildings meet the housing, business, and manufacturing needs of the city. Landmarks enhance cultural and social satisfaction among citizens.
- **Utilities:** The availability of water, power, and waste management systems impacts the functioning of buildings and directly affects citizen satisfaction and overall city stability.
- **Transportation:** Roads, public transit, and other transit infrastructure support

citizen mobility, reduce traffic congestion, and improve access to different parts of the city.

- **Citizens:** Citizens are dynamic entities within the simulation, with needs and satisfaction levels influenced by housing, employment, taxation, and available services.
- **Government:** Governmental systems control budgeting, taxation, and public policies, which influence economic stability, service quality, and citizen satisfaction.
- **Resources:** Resource availability and management are essential for city expansion, and insufficient resources can slow down city growth and decrease satisfaction.

Influence on Project Design:

The above components guided the modular and scalable design of our simulation. By understanding the interaction between different city elements, we have implemented a system where buildings, utilities, citizens, and government subsystems work together dynamically. Design patterns like Observer, Factory, and Command are applied to support this structure, creating a simulation where user actions have cascading effects on city growth and citizen satisfaction. For example, the Observer Pattern enables real-time updates to citizens' satisfaction levels in response to government policies or resource allocation changes.

Assumptions and Design Decisions:

- **Assumptions:** A simplified model assumes that each building type has predictable impacts on the economy and citizen satisfaction. We also assume that resources like water and energy are high-level resources, with detailed consumption managed by the utility subsystem.
- **Design Decisions:** Each main component (buildings, citizens, resources, etc.) is modular, allowing for easy expansion. The system uses patterns such as Mediator to handle interactions between utilities and Factory to facilitate flexible building creation, maintaining separation of concerns.

Definitions and Explanations:

- **Urbanization:** The migration of populations from rural to urban areas, increasing the demand for urban infrastructure and services.
- **Citizen Satisfaction:** A metric influenced by service availability, job access,

housing quality, and tax policies, essential for measuring city growth and stability.
- **Design Patterns:** Reusable solutions to specific design challenges in software, applied here to enable scalability, maintainability, and dynamic simulation behavior.

Assumptions:

- Is a city builder simulator game
- All citizens are taxed
- 4 citizens live in medium sized 2 story residential houses 40 will live in a large 6 story flat
- The city is expandable
- Buildings need to be modular and can be moved around
- The player will start off with an empty lot of land
- Their are different types of citizens from government service citizens to business owner citizens who pay different taxes
- Business owner citizens need to pay personal income tax and sales tax for their business
- Normal citizens need to pay income tax and VAT
- Assume resources are limited and can be depleted by the city
- Resource usage and production needs to be tracked
- Buildings requires resources
- City expansion and increasing land costs money
- Building and construction cost money
- Starter pack city: player starts with an empty plot of land and R10 000 000 in the bank the player needs to first plot a 4 room house for R1 000 000, a government building for R1 000 000, a bank for R1 000 000, a water production system for R1 000 000,an electric plant for R1 000 000 , sewage disposal system for R1 000 000 and a shop for R1 000 000, to start growing their city.
- A residency of citizens provide the total happiness and satisfaction for those citizens
- If taxes are too high or policies are too harsh and strict then a citizen's happiness and satisfaction levels will decrease. If resources are scarce and basic needs are not being met then this will also decrease citizen happiness and satisfaction.

GIT Branching Strategy:

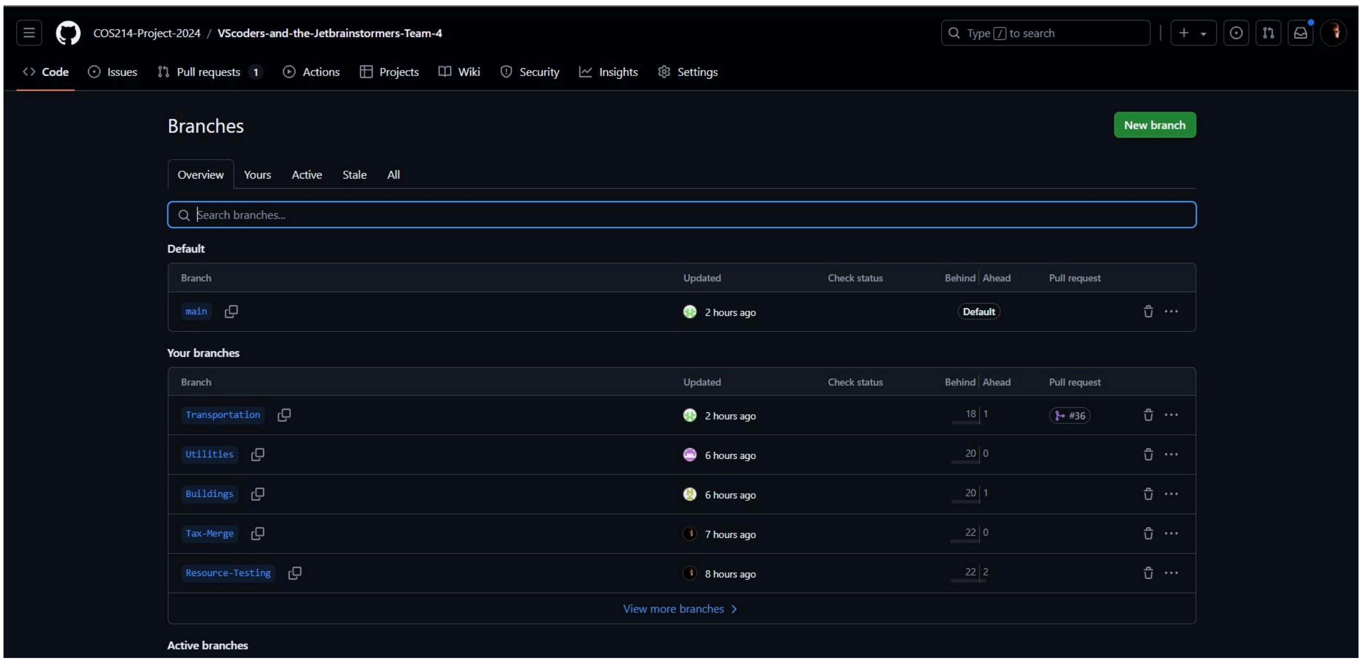
Our GitHub repository follows a structured branching strategy to manage the development of various city-building simulation components. This approach ensures that each team member can work independently on specific project components while maintaining a clean and stable main branch.

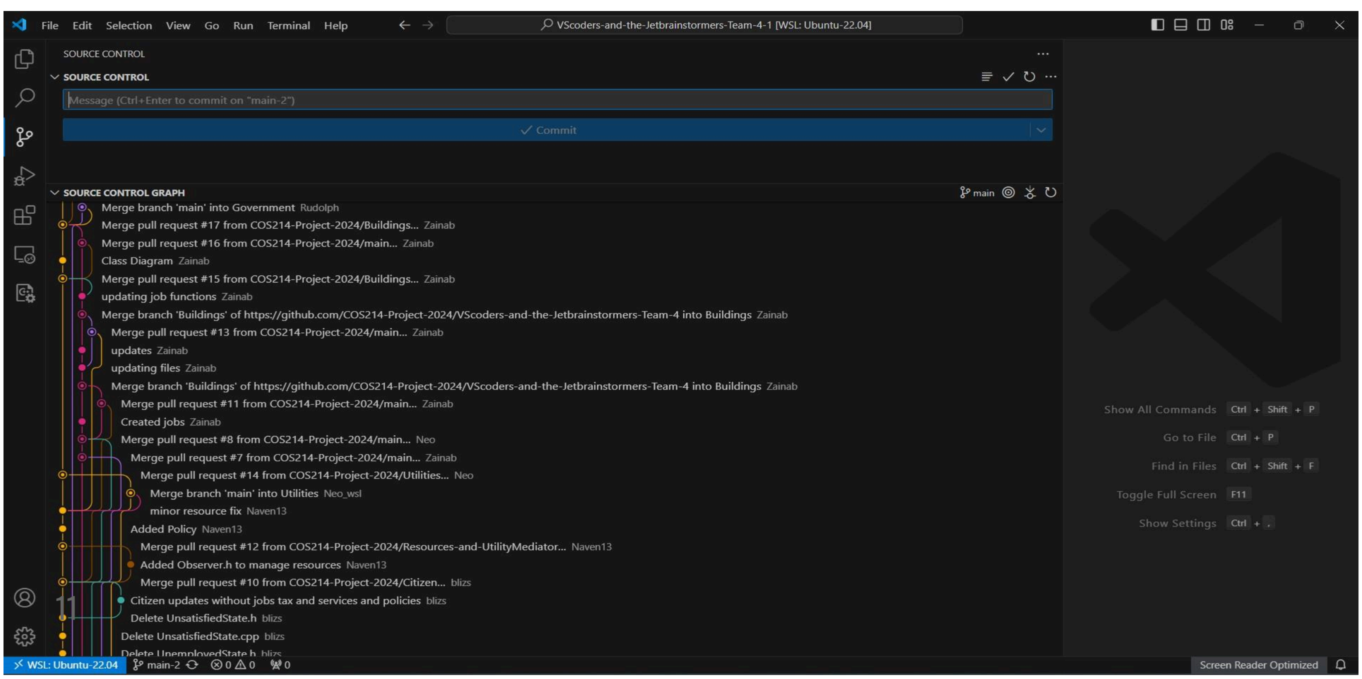
1. **Main Branch:**
    - The `main` branch serves as the primary integration branch. It contains the latest stable code, which includes fully tested and approved features.
    - Only completed, reviewed, and approved changes are merged into `main`, ensuring the stability and functionality of this branch at all times.
2. **Component Branches:**
    - Each major component of the project has its own dedicated branch. For example, if a team member is responsible for developing resources and utilities, they create branches named `resources` and `utilities` from the `main` branch.
    - These component branches act as isolated workspaces, allowing team members to develop and test their assigned features independently.
    - Naming conventions follow the component being developed (e.g., `resources`, `utilities`, `transportation`, `government`), making it clear which feature each branch is associated with.
3. **Workflow:**
    - When a team member begins working on a component, they create a new branch from `main` with a clear, descriptive name matching the component.
    - After completing development and initial testing on their component branch, the team member opens a pull request (PR) to merge changes into `main`.
    - All pull requests are reviewed by at least one other team member to ensure code quality, consistency, and integration readiness.
4. **Merging to Main:**
    - Once the pull request has been reviewed and approved, the branch is merged into `main`.
    - Merging is done only after all tests pass, and any potential conflicts are resolved, ensuring a smooth integration process.
    - After merging, the component branch may be deleted if no further work is

needed, keeping the branch structure clean and organized.

5. **Branch Protection:**
   - The `main` branch has protection rules enabled to prevent direct commits, ensuring that all changes go through a review process via pull requests.
   - This structure ensures quality control and prevents accidental changes to `main`.

By following this branching strategy, our team maintains a collaborative, well-organized workflow, enabling efficient development and seamless integration of components into the city simulation project.





Design patterns in the class diagram:

- [x] ~~Factory~~
- [x] ~~Observer~~
- [x] ~~Mediator~~
- [x] ~~Strategy~~
- [x] ~~State~~
- [x] ~~Chain of responsibility~~
- [x] ~~Iterator~~
- [x] ~~Builder~~
- [x] ~~Template~~
- [x] ~~Command~~
- [x] ~~Prototype~~
- [x] ~~Adapter~~

Identified Design Patterns:

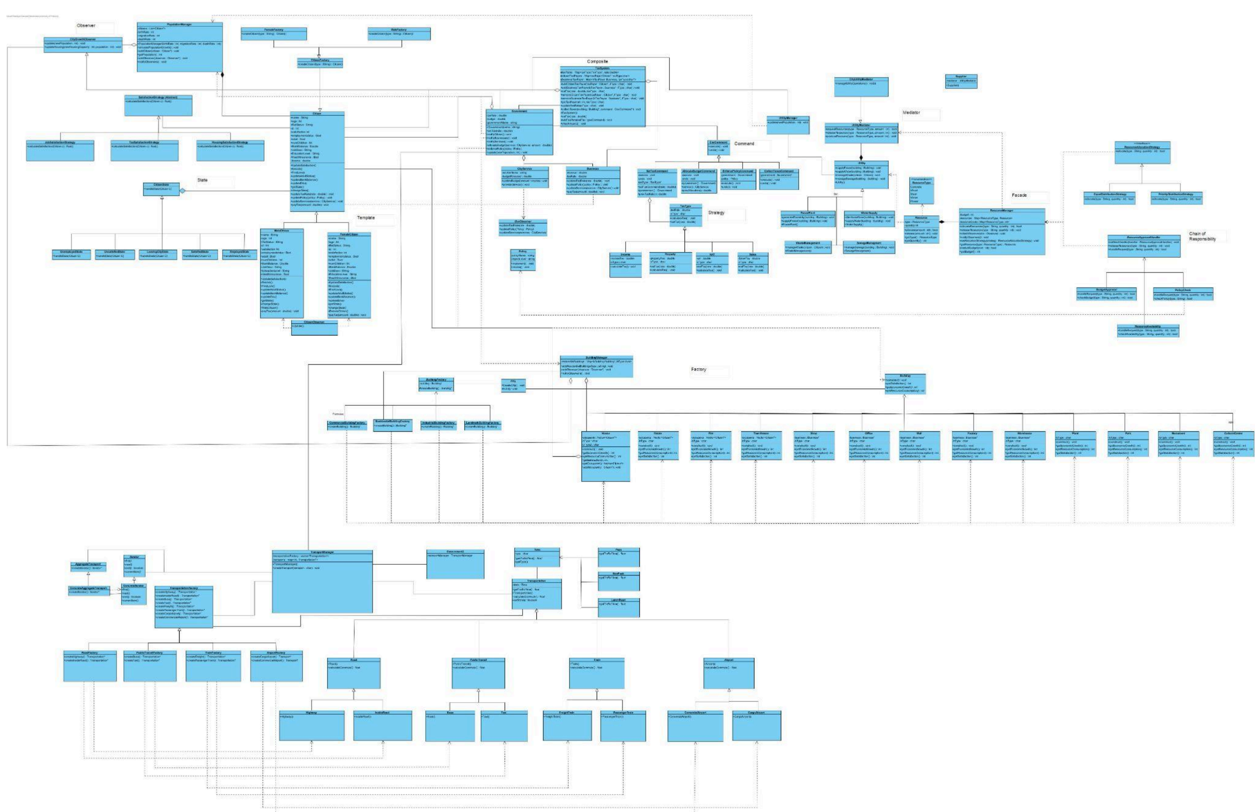
Design Pattern	Problem it Solves	Usage in Diagram
Factory Method	Allows the creation of different building types without specifying their concrete classes.	Factories like ResidentialBuildingBuilder, CommercialBuildingBuilder, IndustrialBuildingBuilder, and LandmarkBuildingBuilder create specific building types.
Strategy	Switches between different resource allocation strategies or tax types dynamically.	ResourceAllocationStrategy allows dynamic resource allocation strategies (e.g., PriorityDistributionStrategy, EqualDistributionStrategy). Also used for switching tax types.
Command	Encapsulates requests (e.g., tax setting, budget allocation) as objects for flexible execution and undo functionality.	GovCommand and its concrete implementations (SetTaxCommand, AllocateBudgetCommand, EnforcePolicyCommand) allow the government to execute and undo commands.
Observer	Automatically updates dependent objects when the state of a	CitizenObserver and CityGrowthObserver notify other entities about changes in population growth, resource usage, or

	subject changes.	citizen satisfaction.
Chain of Responsibility	Passes resource approval requests through a chain of handlers for validation (e.g., budget, policy, availability).	ResourceApprovalHandler and its subclasses (BudgetApprovalHandler, PolicyCheckHandler, ResourceAvailabilityHandler) process resource requests in a chain.
Mediator	Coordinates interactions between components (e.g., utilities, buildings, citizens) without direct interaction between them.	UtilityMediator and CityUtilityMediator mediate between utilities like PowerPlant, WaterSupply, and city infrastructure, ensuring coordinated resource distribution.
State	Changes behavior based on an internal state, such as citizen satisfaction or employment.	CitizenState transitions between states like EmployedState, UnemployedState, SatisfiedState, and UnsatisfiedState, affecting how citizens react to the city's economy and taxes.
Builder Pattern	Constructs complex objects (buildings) step-by-step, allowing for flexible creation.	BuildingBuilder and its subclasses (e.g., ResidentialBuildingBuilder, CommercialBuildingBuilder) build various types of buildings using flexible construction steps.
Prototype Pattern	Allows the cloning of existing building objects to create new instances efficiently.	BuildingPrototype clones buildings like ResidentialBuilding, CommercialBuilding, etc., for efficient replication in city expansion.

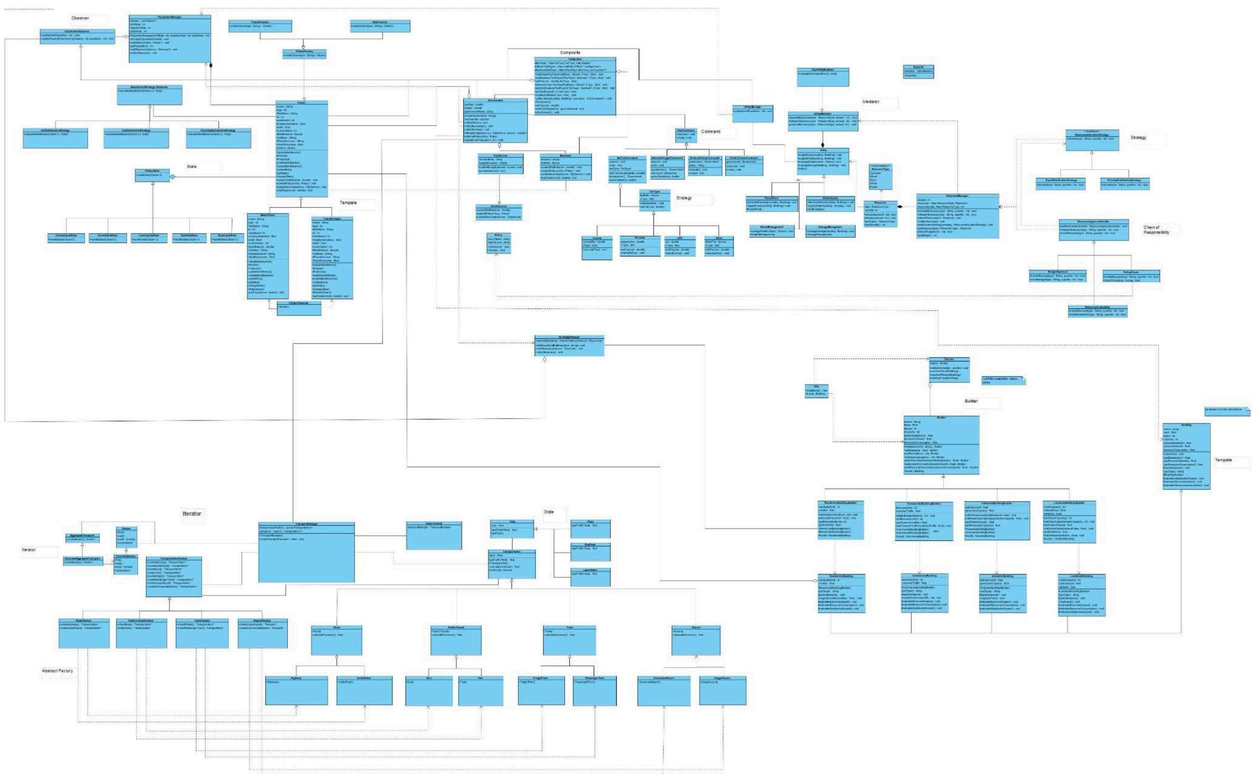
Adapter Pattern	Adapts incompatible interfaces (e.g., external utilities) to work with the internal system.	UtilityAPIAdapter adapts external utility services (e.g., water, electricity) to work with the city's ResourceManager for resource management.

UML Class Diagram:

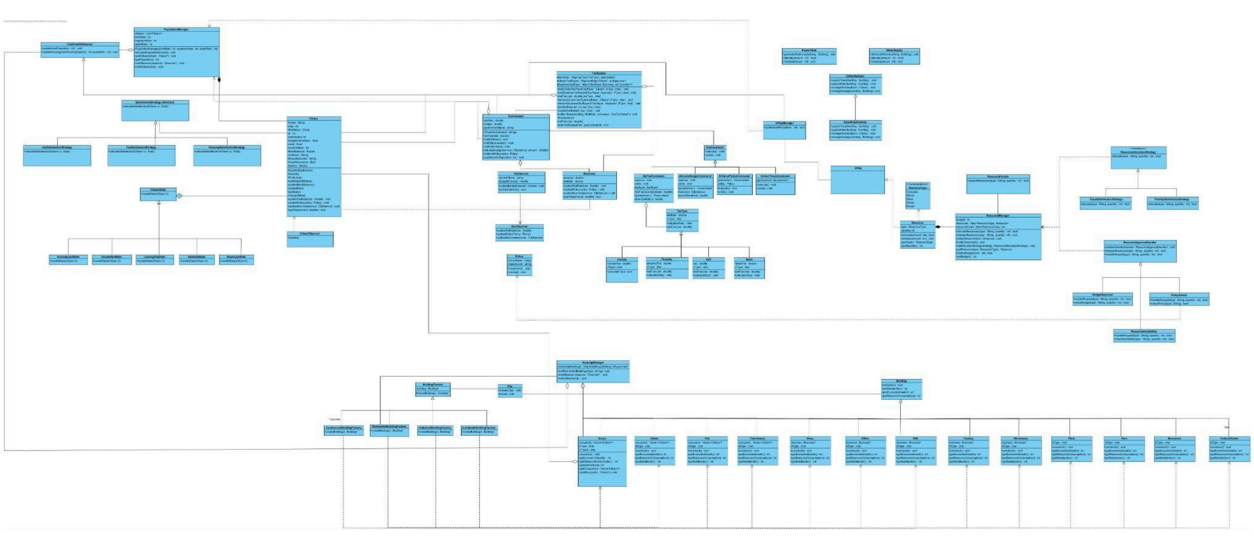
V1:



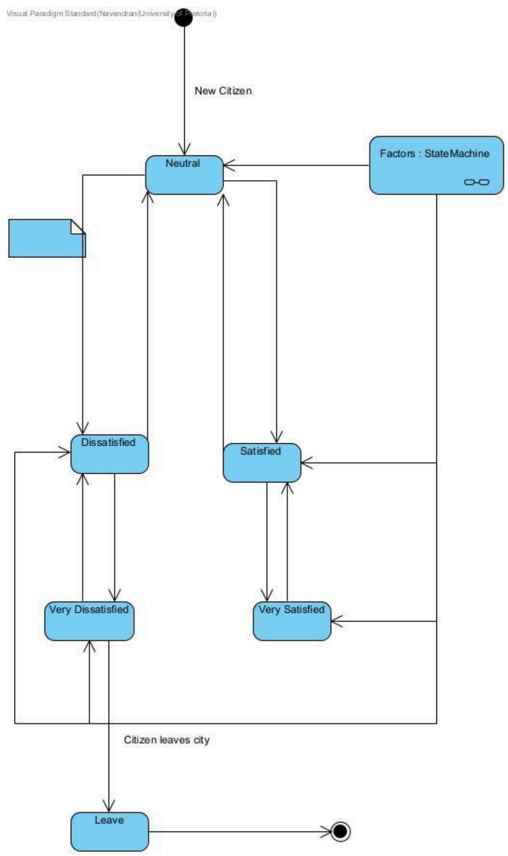
V2:


Observer
Composite
Mediator
Command
Strategy
State
Template
Strategy
Chain of Responsibility
Builder
Template
Iterator
Iterator
State
Abstract Factory

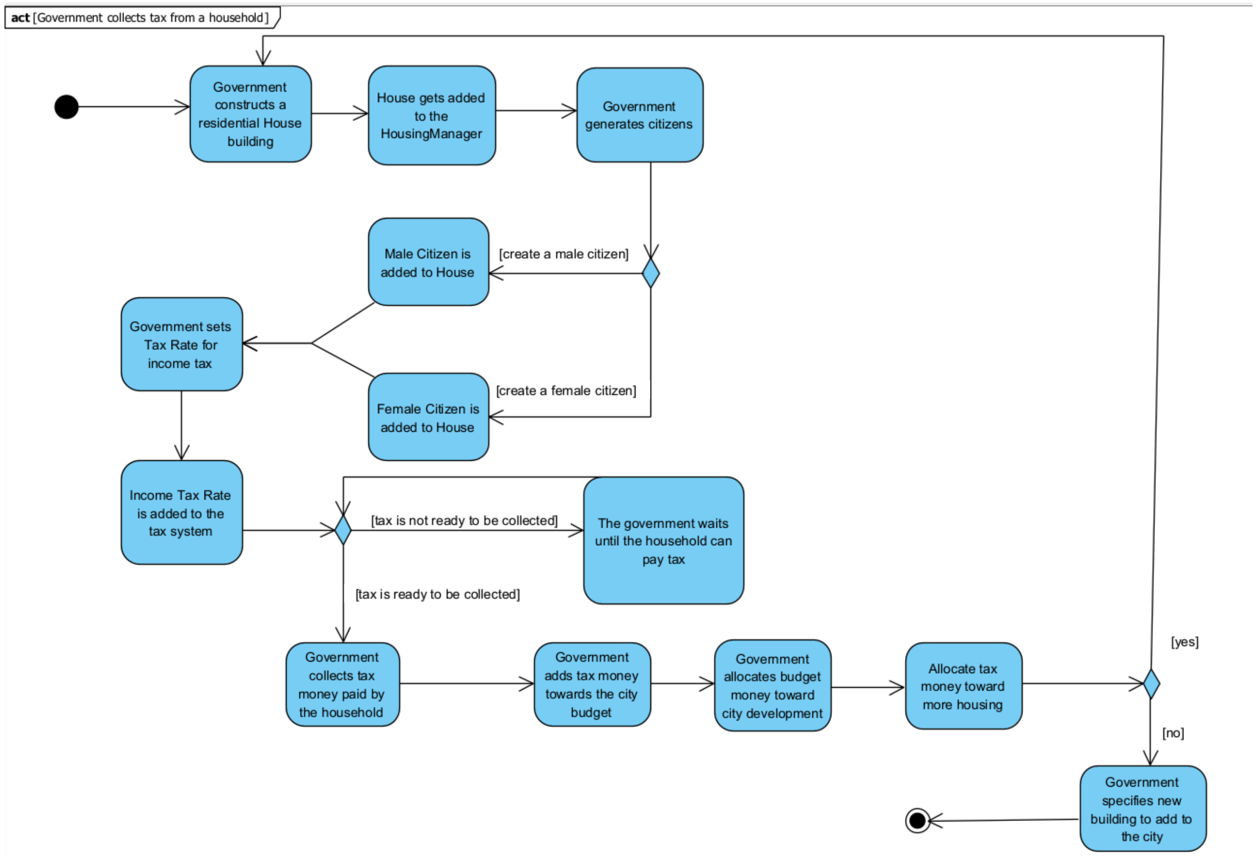
V3:

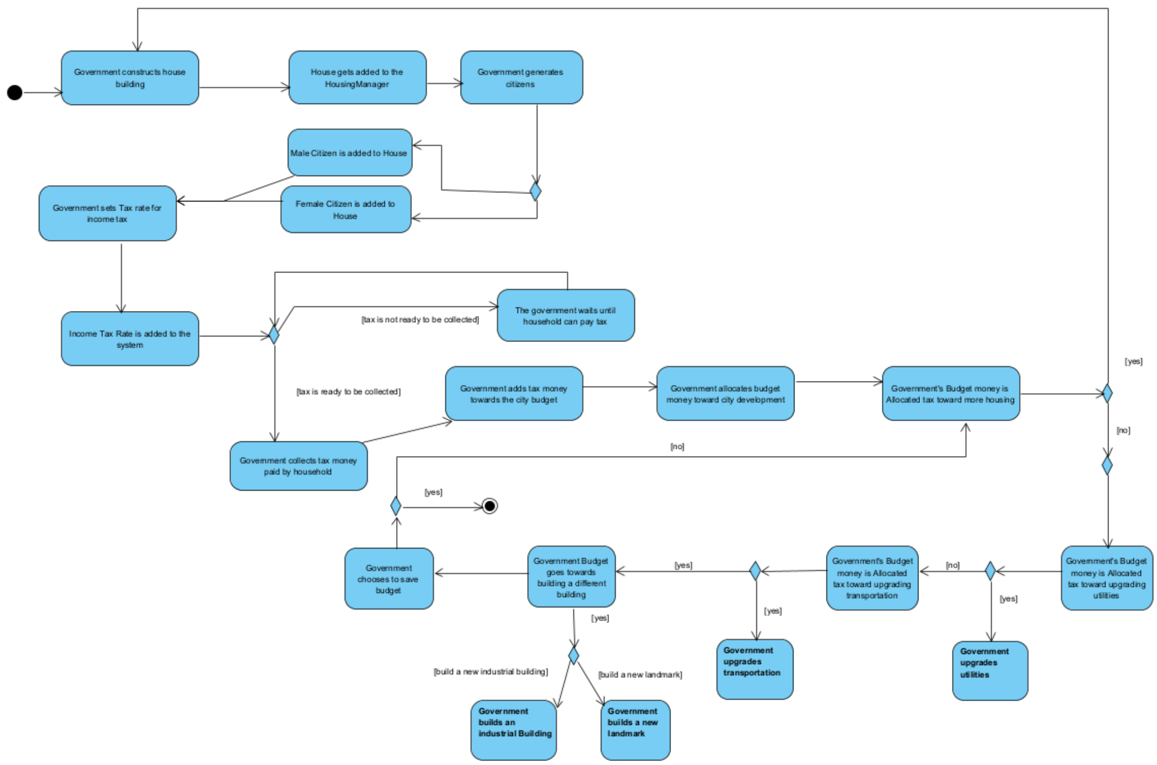


UML State Diagram(s):



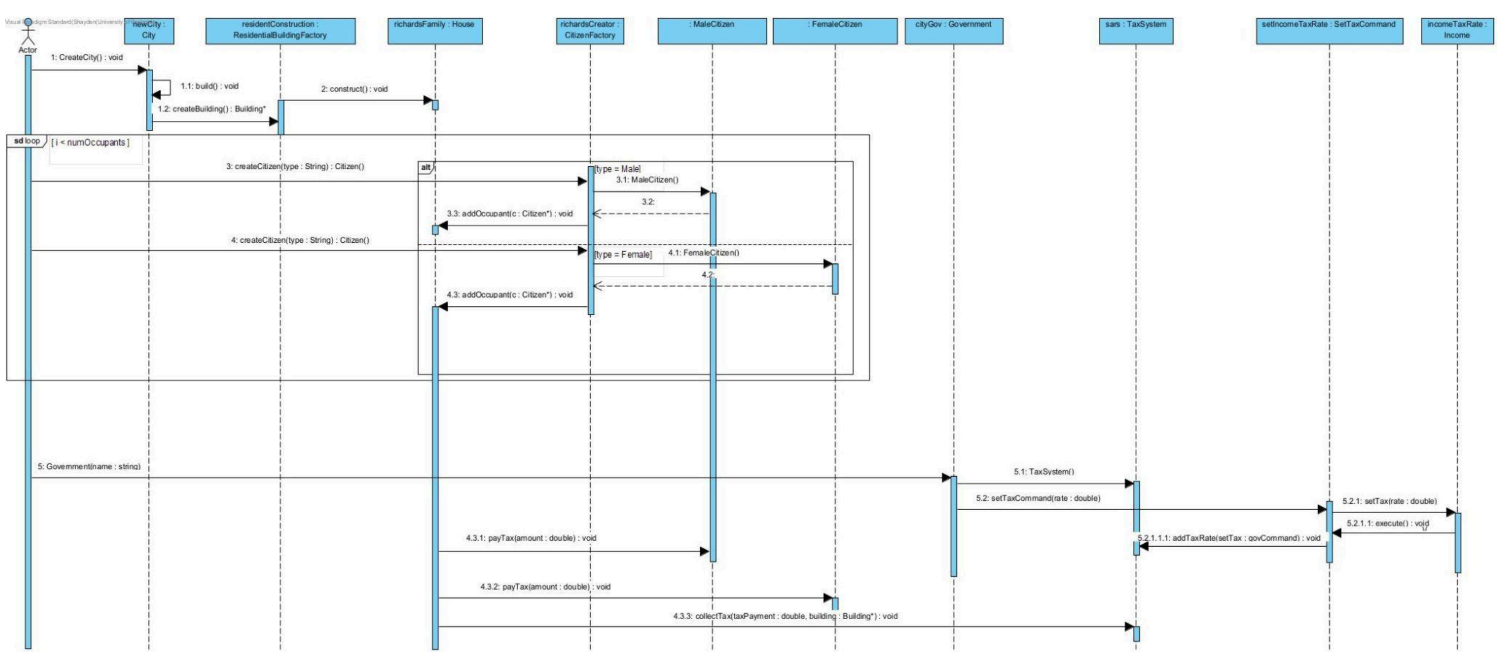
UML Activity Diagram(s):


act [Government collects tax from a household]
Government constructs a residential House building
House gets added to the HousingManager
Government generates citizens
[create a male citizen]
Male Citizen is added to House
[create a female citizen]
Female Citizen is added to House
Government sets Tax Rate for income tax
Income Tax Rate is added to the tax system
[tax is not ready to be collected]
The government waits until the household can pay tax
[tax is ready to be collected]
Government collects tax money paid by the household
Government adds tax money towards the city budget
Government allocates budget money toward city development
Allocate tax money toward more housing
[yes]
[no]
Government specifies new building to add to the city

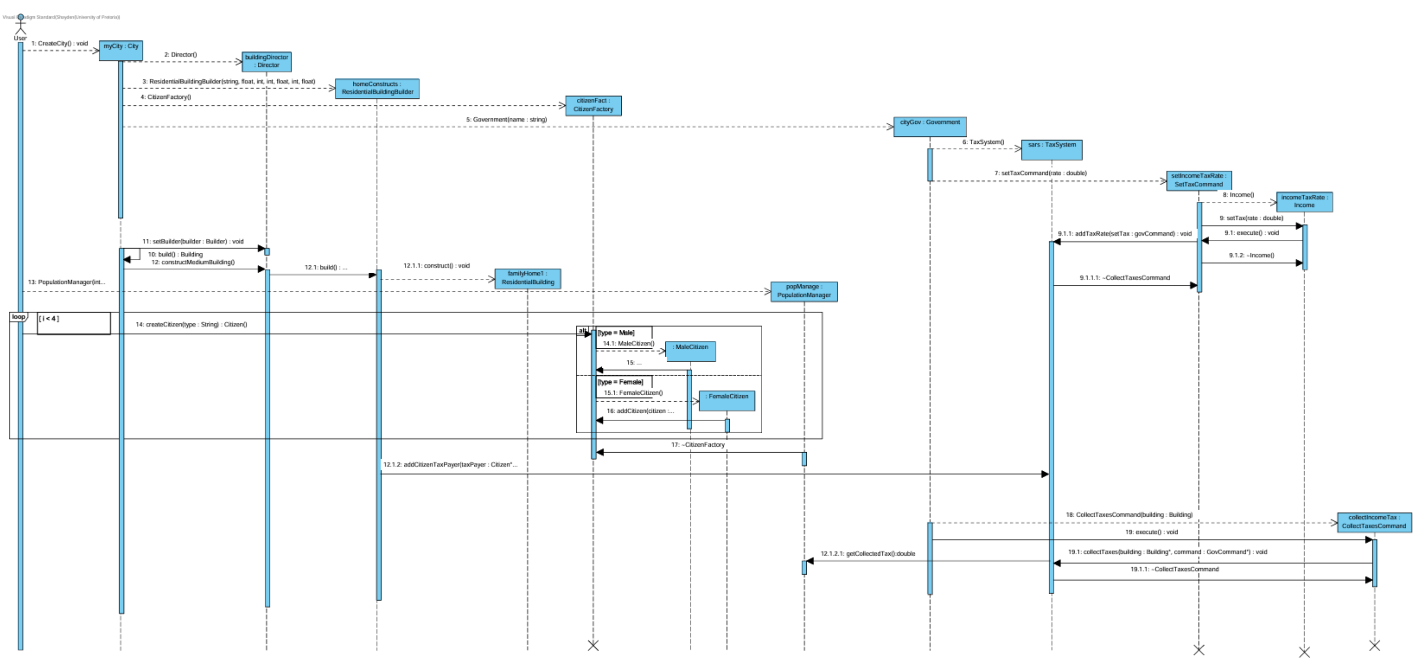


UML Sequence Diagram:

V1:

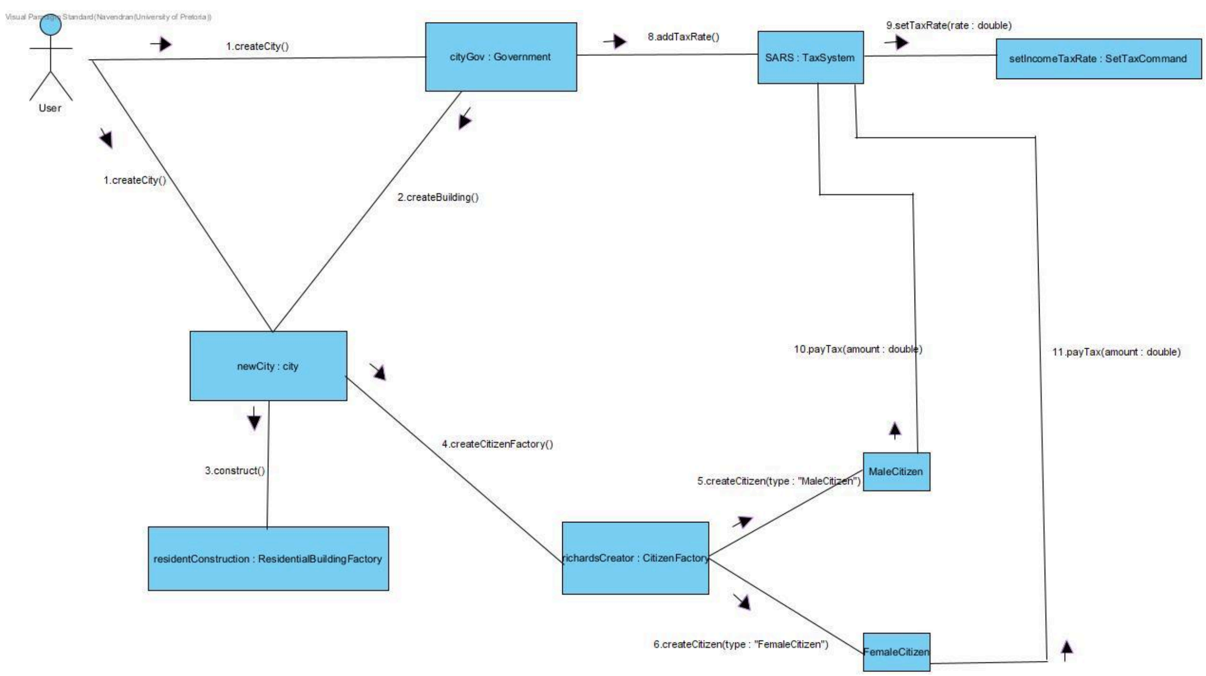


V2:

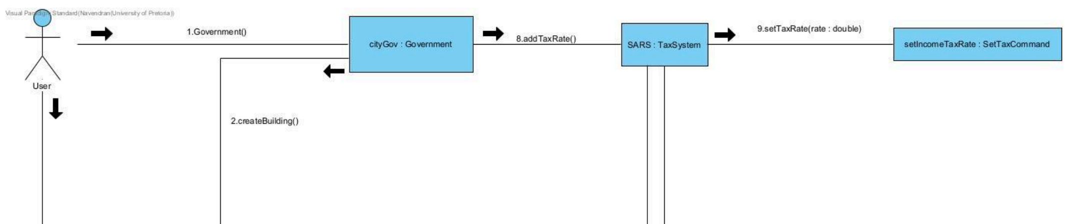

User
1: CreateCity() : void
myCity : City
2: Director()
buildingDirector : Director
3: ResidentialBuildingBuilder(string, float, int, int, float, int, float)
homeConstructs : ResidentialBuildingBuilder
4: CitizenFactory()
citizenFact : CitizenFactory
5: Government(name : string)
cityGov : Government
6: TaxSystem()
sars : TaxSystem
7: setTaxCommand(rate : double)
setIncomeTaxRate : SetTaxCommand
8: Income()
incomeTaxRate : Income
9: setTax(rate : double)
9.1: execute() : void
9.1.1: addTaxRate(setTax : govCommand) : void
9.1.2: ~Income()
9.1.1.1: ~CollectTaxesCommand
11: setBuilder(builder : Builder) : void
10: build() : Building
12: constructMediumBuilding()
12.1: build() : ...
12.1.1: construct() : void
familyHome1 : ResidentialBuilding
13: PopulationManager(int...
popManage : PopulationManager
loop
[i < 4]
14: createCitizen(type : String) : Citizen()
alt
[type = Male]
14.1: MaleCitizen()
: MaleCitizen
15: ...
[type = Female]
15.1: FemaleCitizen()
: FemaleCitizen
16: addCitizen(citizen :...
17: ~CitizenFactory
12.1.2: addCitizenTaxPayer(taxPayer : Citizen*...
18: CollectTaxesCommand(building : Building)
collectIncomeTax : CollectTaxesCommand
19: execute() : void
12.1.2.1: getCollectedTax():double
19.1: collectTaxes(building : Building*, command : GovCommand*) : void
19.1.1: ~CollectTaxesCommand

UML Communication Diagram(s):

V1:

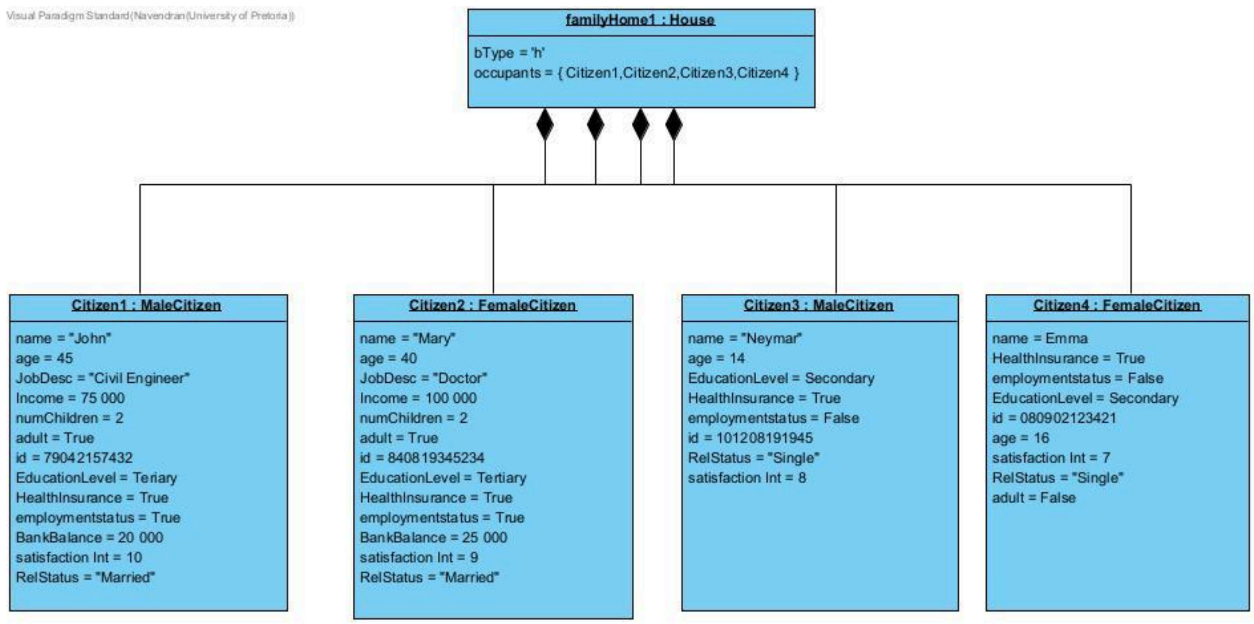


V2:

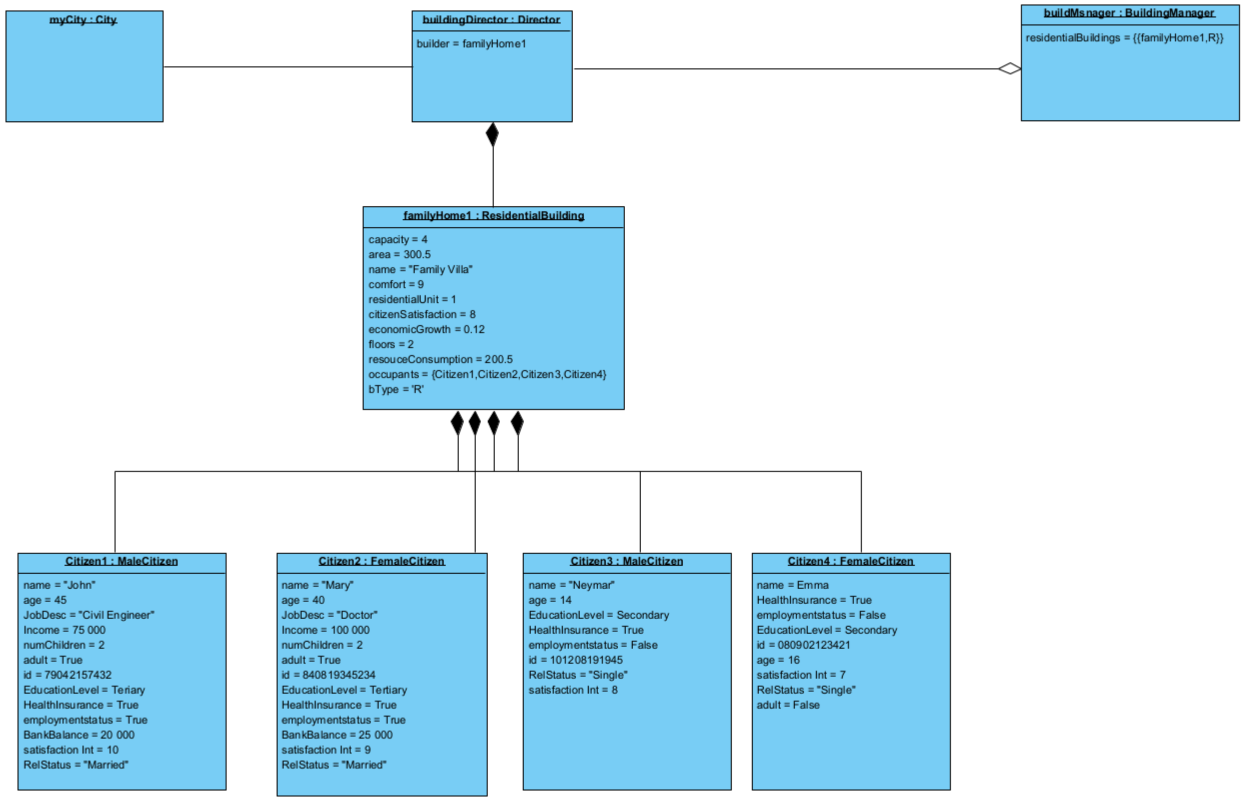


UML Object Diagram(s):

V1:



V2:



REFERENCES:

Marshall, L., 2016. **Tackling Design Patterns**. [online] Department of Computer Science, University of Pretoria. Available at: https://www.cs.up.ac.za/cs/lmarshall/TDP/TDP.html [Accessed 15 October 2024].