**SFE 4020A Group Work/project**

**Group Members**

1. **Collins Kinyanjui 664884**

**Website of choice: NyumbaVerse: Real Estate Platform**

**Live demo:** <https://kinyanjuic.github.io/nyumbaverse>

The Group work/ project consists of the design and implementation of the software architecture of a project of your choice. Implementation will take place using a tool of your choice.

**Task 1**

1. In groups of 3-5 brainstorm on a webApp or a mobileApp that you can develop as a team.
2. What area does your application Cover e.g. health, Finance, tourism etc.? Further explain why you chose that area for the application.

The NyumbaVerse application covers the real estate sector in East Africa. This is a strategic choice because:

* East Africa has a growing real estate market with increasing urbanization
* There's a need for a centralized platform to connect buyers, sellers, and renters
* The region has unique property types and market dynamics that need specialized handling
* There's potential for significant economic impact through property transactions

1. Who are the intended users of the application?

* Property buyers and renters
* Property sellers and landlords
* Real estate agents
* Property investors
* Property managers
* Home seekers across East Africa

1. What is the purpose of your application?

* To provide a comprehensive real estate platform for East African properties
* To connect property seekers with available properties
* To facilitate property management
* To provide market insights and property valuation services

1. What are the benefits of the Application?

* Centralized access to East African properties
* Market transparency and information access

1. Which tools are you likely to use for your implementation and why? Do you have the skillset required? If not, what how do you plan to bridge the knowledge gap?

Skillset Analysis:

* Required: Web development, database management, UI/UX design, API development

Knowledge Gap Bridging:

* Online courses (Coursera, Udemy)
* Documentation and tutorials
* Team collaboration and knowledge sharing
* Mentorship from experienced developers

**Task 2**

2. Describe the organization of catalog along with the following design patterns as given by the gang of four (GoF).

a. Creational Patterns.

* Abstract Factory: For creating property-related objects
* Builder: For constructing complex property objects
* Factory Method: For creating different types of property listings
* Singleton: For managing global configurations

b. Structural Patterns.

* Adapter: For integrating different payment systems
* Bridge: For separating property abstraction from implementation
* Composite: For managing property portfolios
* Decorator: For adding features to property listings
* Facade: For simplifying complex property search operations
* Proxy: For controlling access to property data

c. Behavioral Patterns.

* Chain of Responsibility: For handling property requests
* Command: For property management operations
* Iterator: For traversing property collections
* Mediator: For coordinating between different property services
* Observer: For property price updates
* Strategy: For different property search algorithms
* Visitor: For property valuation operations

3. Using a table show the classification of design patterns as introduced by the Gang of Four. The table should have the following subheadings.

|  |  |  |
| --- | --- | --- |
| Purpose | Design Patterns | Scope |
| Creational | |  |  | | --- | --- | | Abstract Factory, Builder, Factory Method, Prototype, Singleton |  | | Object |
| Structural | Adapter, Bridge, Composite, Decorator, Façade, Flightweight, Proxy | Class & Object |
| Behavioral | Chain of Responsibilty, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, strategy, Template, Method, Visitor | Class & Object |

**Task 3**

4. for your chosen area of application, Design;

1. Use cases

**1. Property Search**

**Actors**: Tenant, Buyer, Guest  
**Preconditions**: User accesses the search interface.  
**Postconditions**: System displays properties matching basic criteria.  
**Steps**:

1. User enters a location (e.g., "Nairobi").
2. User clicks "Search."
3. System fetches properties from the database matching the criteria.
4. System displays results in a grid/list view.  
   **Exceptions**:

* No properties found → Show "No results" message.

**2. User Management**

**Actors**: Guest  
**Preconditions**: Guest navigates to the registration page.  
**Postconditions**: New user account created.  
**Steps**:

1. User enters email, password, and phone number.
2. User clicks "Sign Up."
3. System validates uniqueness of email/phone.
4. System stores user data.
5. System displays verification.  
   **Exceptions**:

* Invalid email → Show " Please enter a valid email address" error.

**3.Property Management**

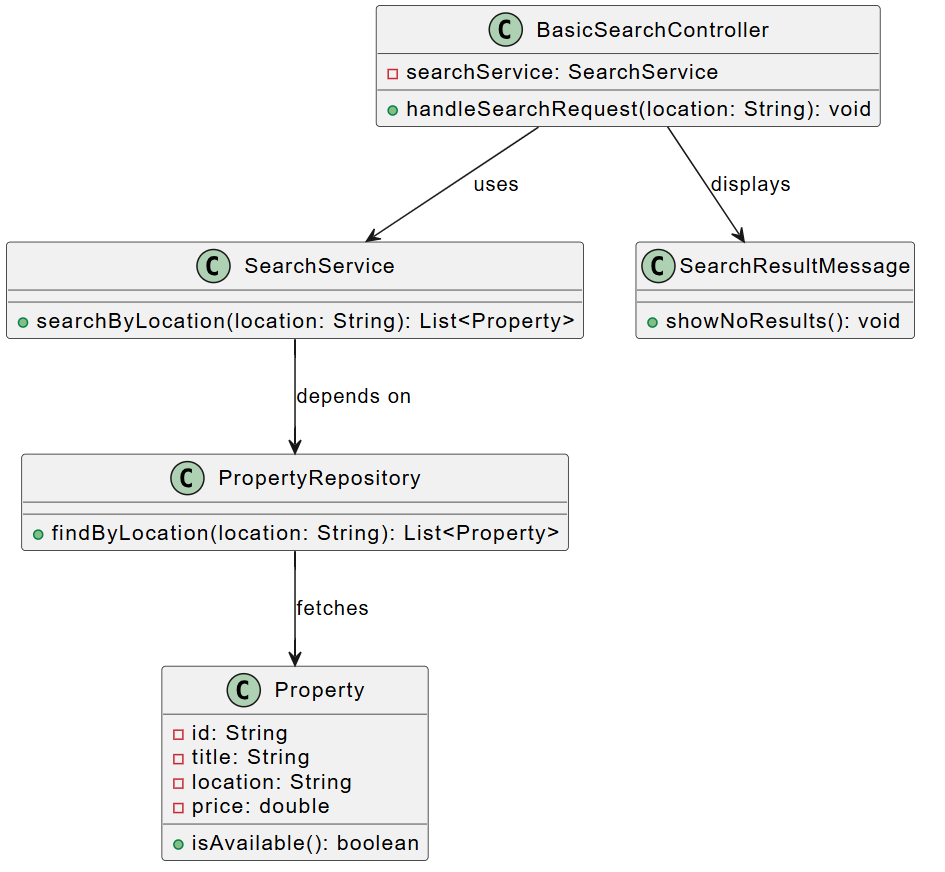
**3.1Add to Cart**

**Actors**: Registered User  
**Preconditions**: User views property details.  
**Postconditions**: Property added to the cart.  
**Steps**:

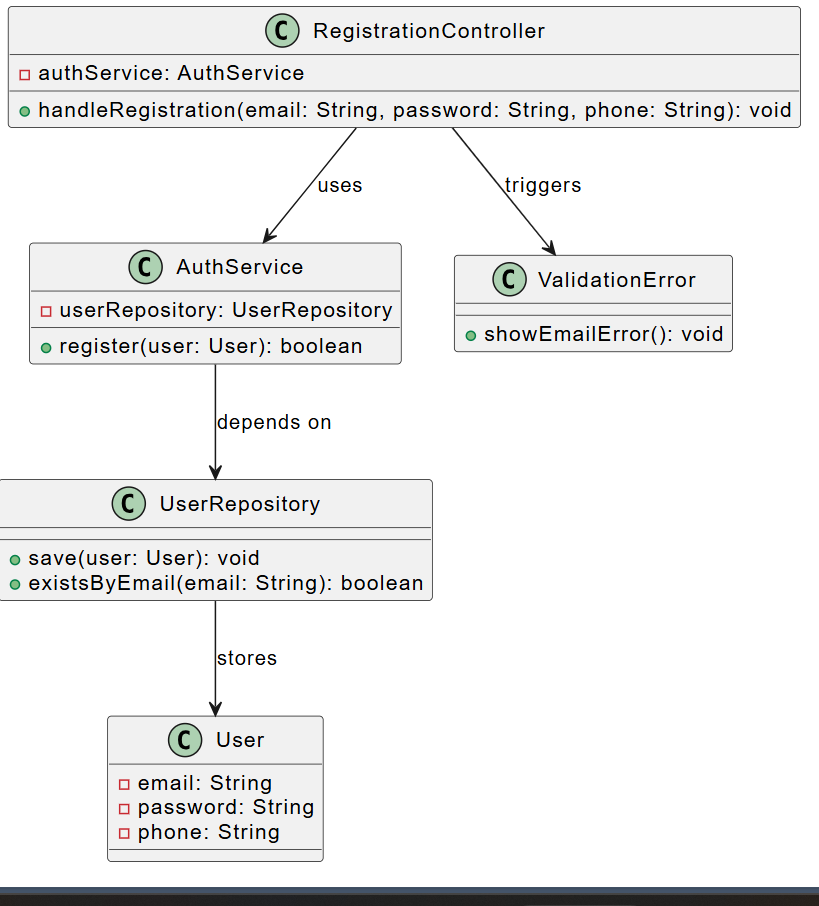
1. User clicks "Add to Cart."
2. System checks property availability.
3. System adds the property to the user’s cart.
4. Cart counter increments.  
   **Exceptions**:

* Property no longer available → Show error and remove listing.

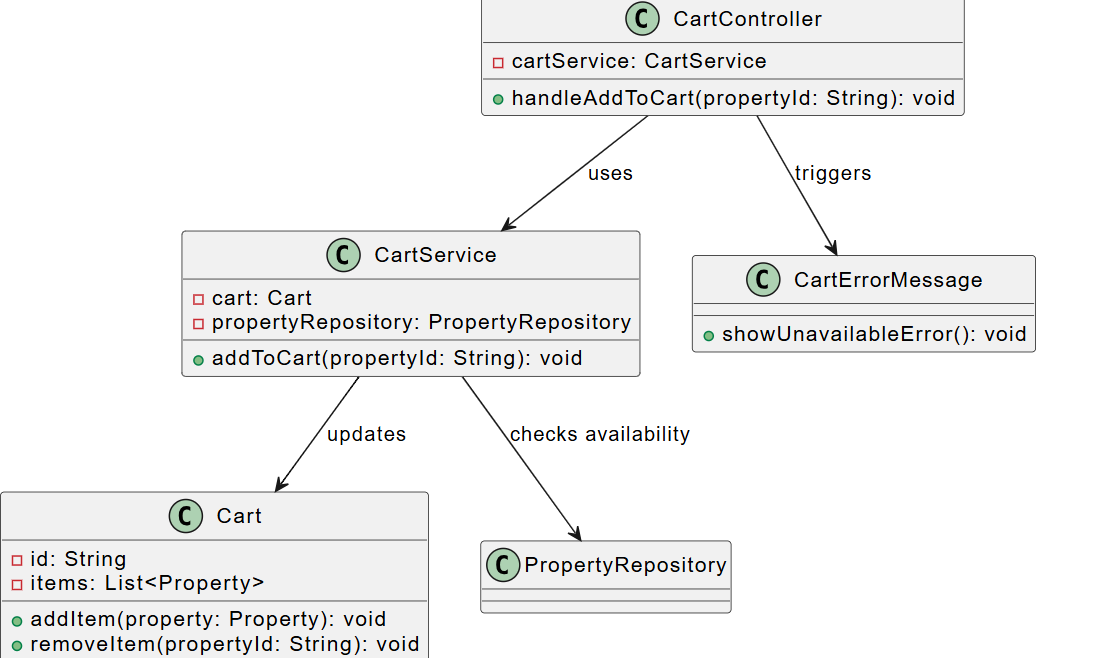
1. Class Diagrams
   * + 1. Basic Property search



* + - 1. User Registration

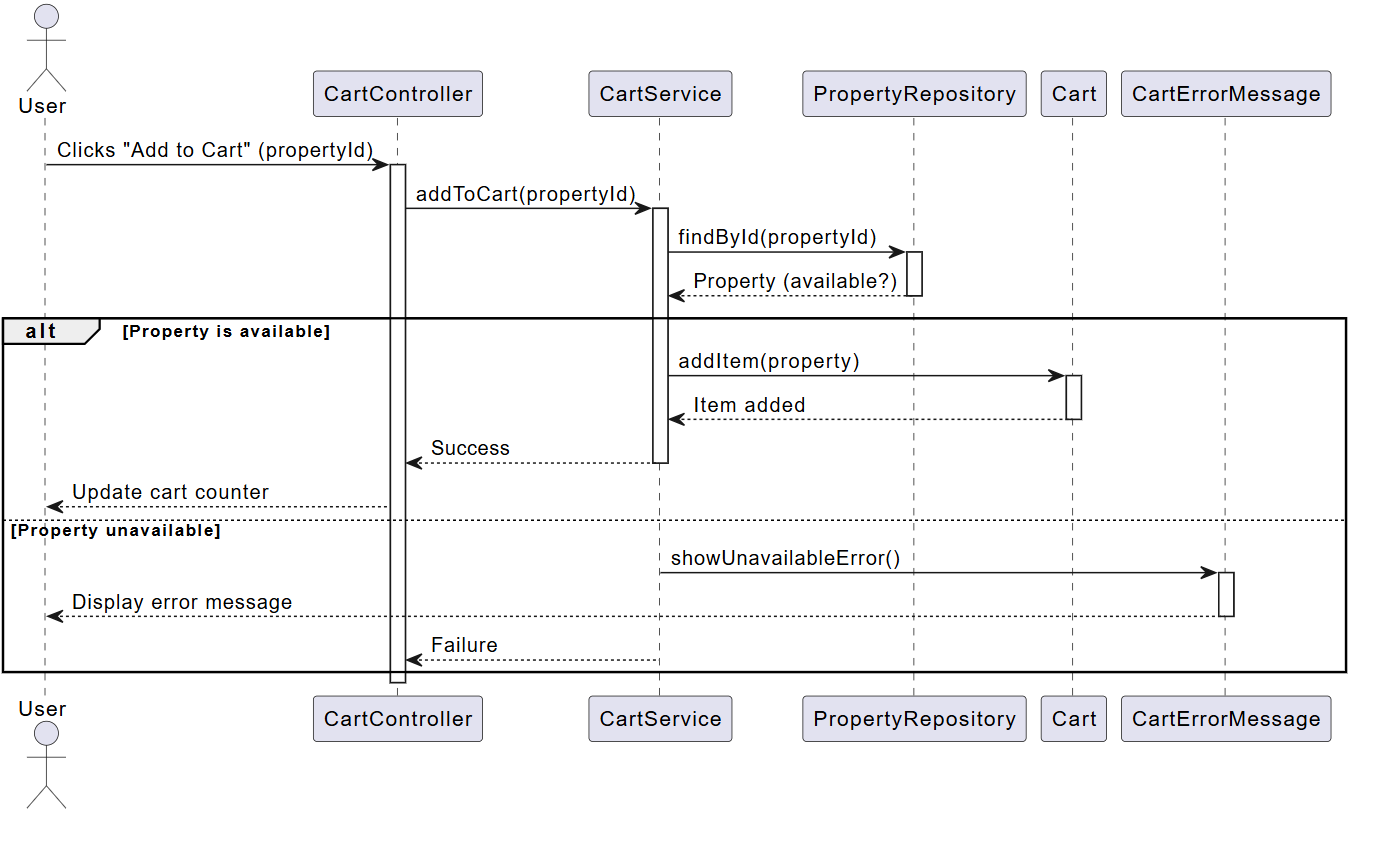


* + - 1. Add to Cart



1. Any other UML diagram of your choice.

* **sequence diagram** for the **Add to Cart** use case



5. Design Patterns;

Using UML design;

1. Abstract factory design pattern
2. Adapter-class Design pattern
3. Adapter-object Design pattern
4. Strategy Design pattern
5. Builder Design pattern
6. Bridge Design pattern
7. Decorator Design pattern
8. Flyweight Design pattern
9. Facade Design pattern
10. Iterator Design pattern
11. Mediator Design pattern
12. Proxy Design pattern
13. Visitor Design pattern

NB..chose any six. Do ensure that there is atleast one from each of the three major categories, i.e. Creational Patterns, Structural Patterns and Behavioral Patterns.

**Task 4**

6. Implementation.

Using a tool of your choice write a program to implement each of the design pattern you choose in question 5 above.