

Object-Oriented Programming Project

Angie Nathaly Cisneros Guerrero

20242020084

Luis Sebastián Correa Barreto

20242020085

Project: Replicate the functionality of the "El Espectador" website, which provides access to current news.

1.1 Conceptual Design Updates:

- **Objectives:**

- Keep the user informed through easily accessible news.
- Find news by classification to make your search easier.
- Offer comprehensive and detailed reports on relevant topics.

- **Functional requirements:**

- View the main news on the homepage.
- View top news.
- See the different news sections.
- View latest news (last 9 published).
- Access full details of a news article.

- **Non-Functional requirements:**

- Scalability for increased news content.
- Intuitive and accessible navigation (for demographics).
- Modular and maintainable codebase.
- Adequate documentation of code and architecture.

- **User stories:**

Title: Home Page Reader	Priority: High	Estimate: 6 weeks
User Story: <i>As a visitor</i> , I want to see the main and top news clearly on the homepage so I can quickly get informed about the most relevant events.		
Acceptance Criteria: -Main news article displayed prominently. - Section for "Top News" visible under the main article.		
Title: Latest News Reader	Priority: Medium	Estimate: 4 weeks
User Story: <i>-As a</i> frequent visitor, I want to view the latest news quickly to stay updated.		
Acceptance Criteria: - "Latest News" section that displays the 9 most recently published articles. - Articles ordered from newest to oldest.		

Title: International News	Priority: Low	Estimate: 4 weeks
User Story: <i>As an</i> international affairs enthusiast, I want a dedicated section for international news so I can stay informed globally.		

Acceptance Criteria:

- International" section easily accessible.
- Displays at least 9 articles from different countries.

Title:

News

Priority:

High

Estimate:

1 week

User Story:

As an app developer I want to create news whit title, content and date so the readers can see the info sorted and updated.

Acceptance Criteria:

- “News” this is the most important thing in the project because it’s the base for all sections.

Title:

Different types of sections

Priority:

Medium

Estimate:

4 weeks

User Story:

-As a frequent visitor, I want to view a section of latest news and another I don’t mind whatever it is.

Acceptance Criteria:

- This is acceptable and good because we can create a single class called “Section” to spare different kind of news

- **CRC Cards:**

Class Name: New

Responsibilities:

Collaborators:

-Manage article attributes (Title, content, date, etc.)	-Section.
Class Name: Section	
Responsibilities:	Collaborators:
-Group news by category (Latest news, international news).	-News.
Class Name: Homepage	
Responsibilities:	Collaborators:
-Display last news	-News -Section
Class Name: LatestNewsSection	
Responsibilities:	Collaborators:
-Group news by category (Latest news, international news).	-News. -Section

1.2 Documentation of changes respect to workshop-1:

- ❖ Decisions emerged from further planning and discussions.

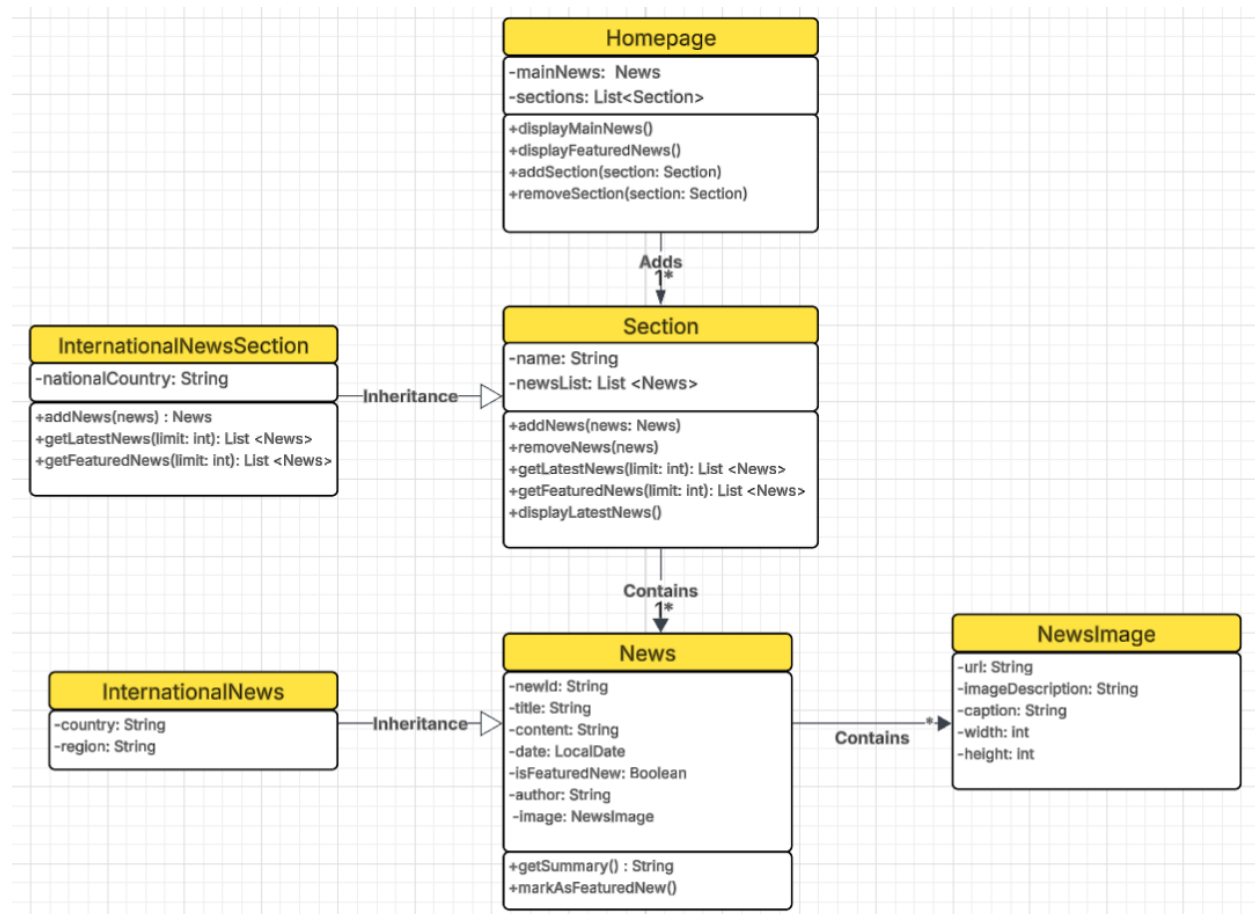
Workshop 1 → Workshop 2: Evolution of UML & OOP Implementation:

-Objective: Translate the initial conceptual design into a technical blueprint using OOP principles.

- Refined Class Diagram
- Added OOP Pillars
- New Components
- User Story Alignment
- CRC card classes (Section, News, Homepage, International news).
- Enhanced responsibilities

2. Technical design:

- UML diagram; there's hierarchies, class members (attributes, methods), and relationships:



2.1 How classes implement or override methods to fulfill user requirements and how they integrate the OOP principles:

News (Base Class)

User Requirements:

- Provides the core structure for news articles (title, content, date) to satisfy the need for displaying news.
- Generates summaries (combining title, author, and content snippets) for quick previews.

OOP Integration:

- **Encapsulation:** Hides internal data (title, content) behind private fields and exposes them via controlled methods (getters/setters with validation). For example, it rejects empty titles.
- **Reusability:** Serves as the parent class for specialized news types (like international news), avoiding code duplication.

NewsImage (Base Class)

User Requirements:

- Provides a structured way to represent images associated with news articles (url, caption, image description) to fulfill the need for displaying visual content and improving accessibility.
- Potentially includes metadata (caption, dimensions) for richer image representation and layout management.

OOP Integration:

Encapsulation: Conceals internal data (url, image description, caption) using private fields and offers controlled access through methods (getters/setters with validation). For example, it might reject empty url's or enforce maximum lengths for description text.

Reusability: Serves as a component of the News class (through composition), allowing different types of news to easily include image information without repeating image-related attributes in each specialized news class. This promotes a cleaner and more maintainable News hierarchy.

InternationalNews (Derived Class)

User Requirements:

- Extends news with geographic data (country/region) to support the "International News" section.
- Filters out national news (non-Colombian news) to meet the demand for global content.

OOP Integration:

- **Inheritance:** Inherits all attributes/methods from News (getSummary) while adding location-specific features.
- **Polymorphism:** Designed to override methods like getSummary (though not yet implemented) to include country info, enabling dynamic behavior.

Section (Composition Class)

User Requirements:

- Groups news by categories ("Latest News," "International") to organize content.
- Retrieves recent news (last 9 articles) for the "Latest News" section.

OOP Integration:

- **Composition:** Manages a list of News objects instead of inheriting from News, adhering to "has-a" relationships.
- **Single Responsibility:** Focuses solely on news organization (sorting by date) without handling rendering or storage.

Homepage (Aggregator Class)

User Requirements:

- Displays main and featured news on the homepage.
- Delegates news listing to Section classes ("Latest News" section).

OOP Integration:

- **Delegation:** Relies on Section to fetch news (getLatestNews), promoting modularity.
- **Dependency Injection:** Accepts News and Section objects via constructor, decoupling dependencies.

InternationalNewsSection (Specialized Logic)

User Requirements:

- Customizes the standard Section to exclude national news, ensuring only international content appears.

OOP Integration:

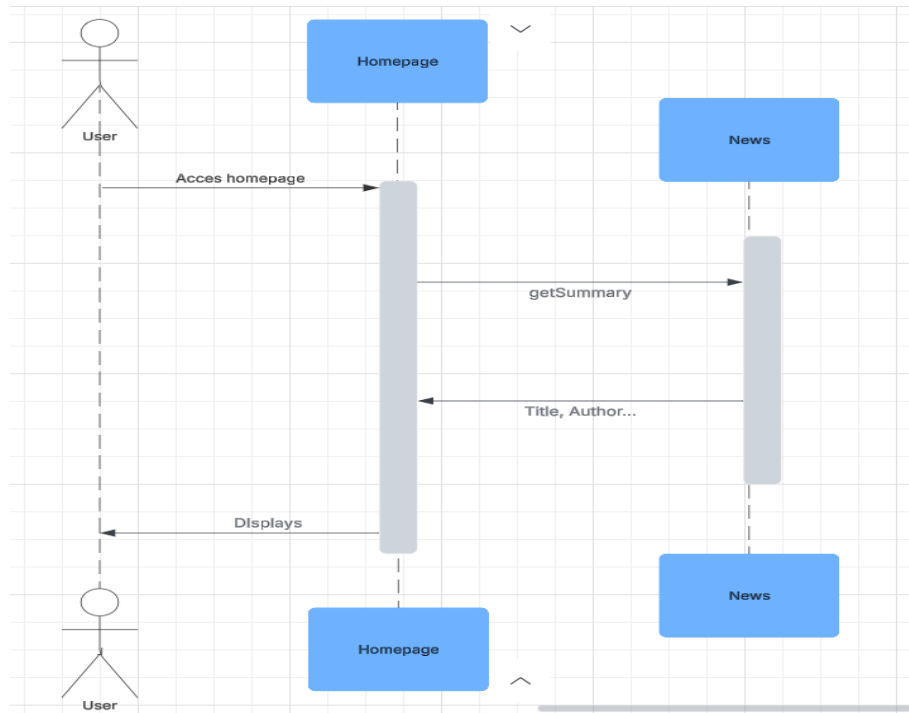
- **Polymorphism:** Introduces unique filtering logic (isInternationalNews) without modifying the base Section class.

OOP Principles in Action

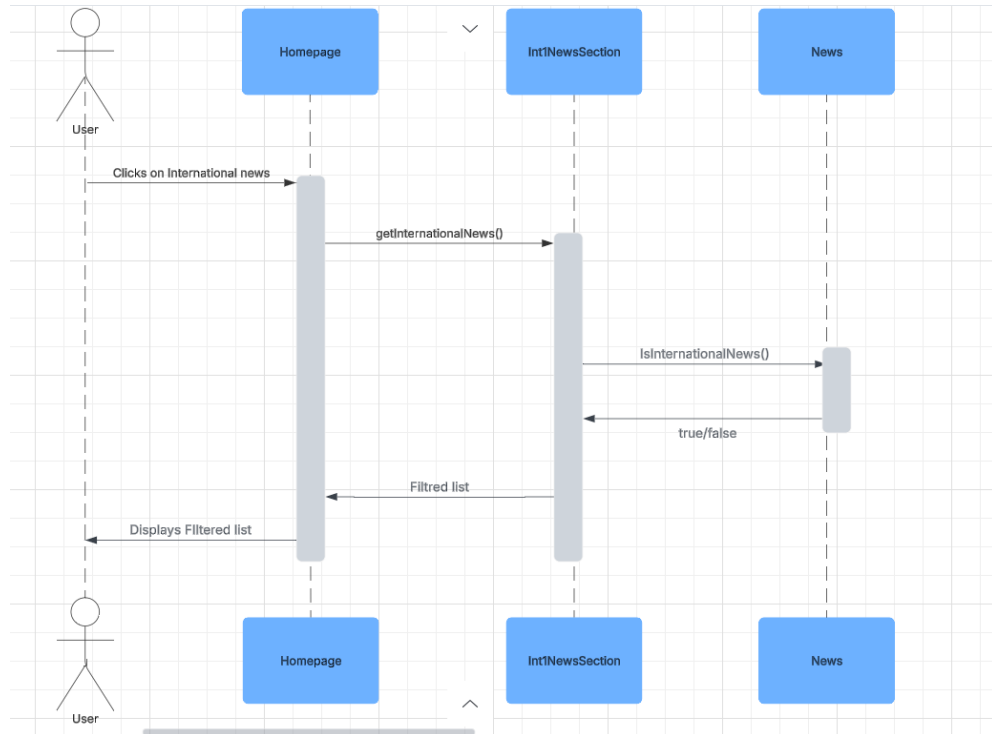
- **Encapsulation:** Safeguards data integrity (News validates dates).
- **Inheritance:** InternationalNews builds on News without rewriting common features.
- **Polymorphism:** Methods like getSummary can adapt to subclasses (e.g., appending country names).
- **Composition:** Homepage and Section collaborate through object relationships, not inheritance.

2.3 Sequences diagrams; these illustrate interactions between classes and the flow of data:

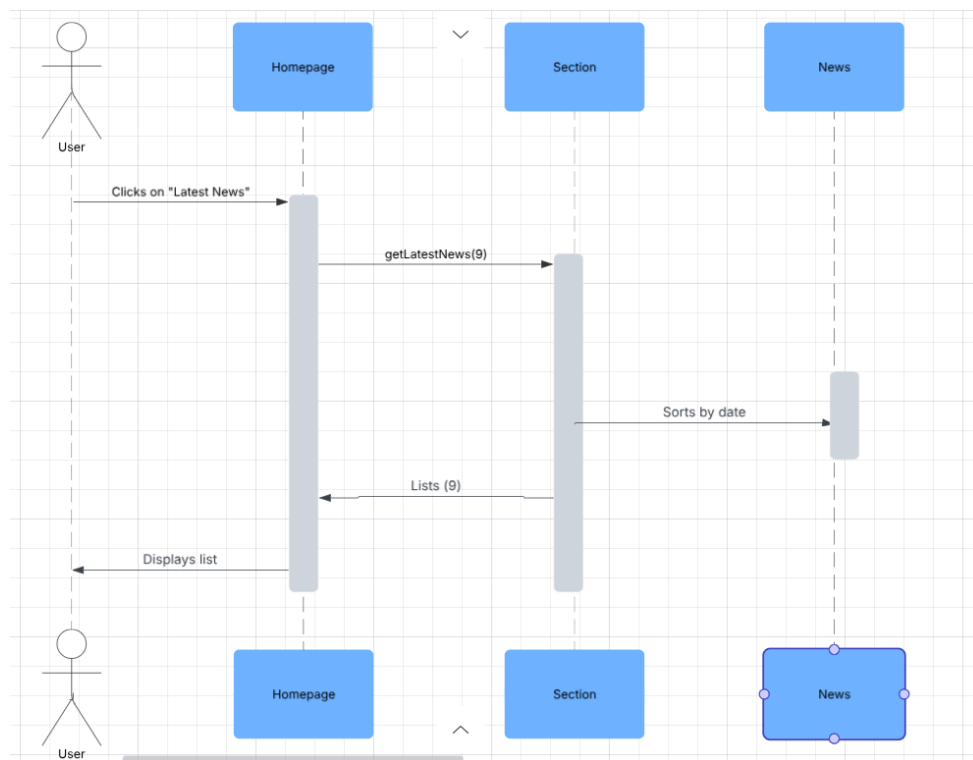
- **Diagram “Homepage”:**



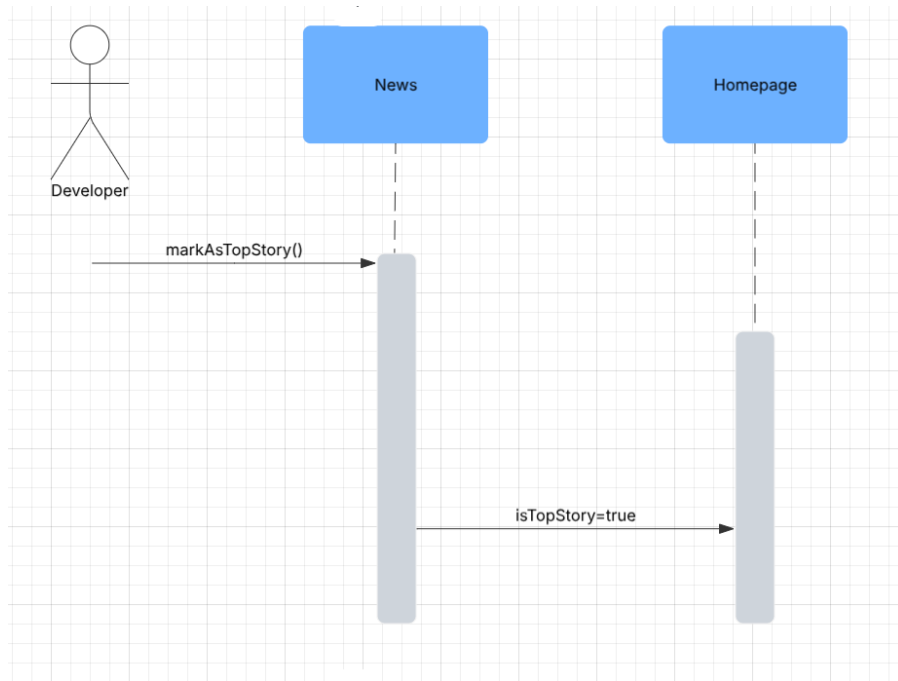
- **Diagram “Access to international news”:**



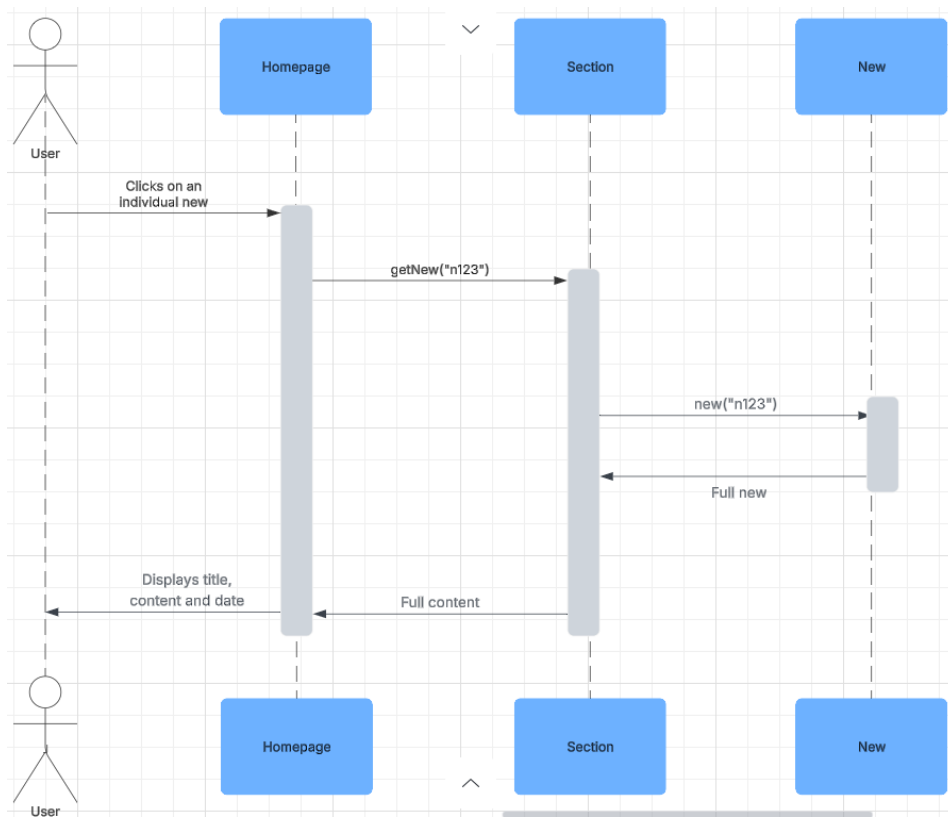
○ **Diagram display Latest News:**



○ **Diagram Mark as Top Story:**



○ **Diagram display individual new:**



3. Implementation Plan for OOP Concepts:

Summary of how our code will realize encapsulation, inheritance and polymorphism:

- **Encapsulation:** Protecting Data
 - Each class controls access to its internal attributes through private methods. For example, the News class keeps its title and content as private properties, only allowing modifications through methods that include validations (like ensuring the title isn't empty).
- **Inheritance:** Specializing Behavior
 - The InternationalNews class inherits from the News, leveraging all its base attributes and methods while adding specific properties like country of origin, demonstrating the principle of extending without modifying.
- **Polymorphism:** Flexible Behavior
 - When the system calls the "getSummary" method, the InternationalNews class responds differently than the base News class by automatically including country information. This allows processing different news types uniformly.
 - Also, when accessing sections from the Homepage and calling getLatestNews(), the InternationalNewsSection will execute its own logic to retrieve the latest news (excluding those from Colombia), while other sections, such as the general section, use different logic and simply list the most recent news by date without considering other aspects.
- **Composition:** Building Relationships
 - The Section class contains and manages collections of News objects, coordinating how news is grouped and retrieved. The Homepage in turn uses these sections to present information to users.
- **System Flow:**
 - Creation: News instances are created with their basic data.
 - Organization: News is assigned to thematic sections.
 - Presentation: The Homepage requests news from sections and displays it.
 - Interaction: Users can navigate through the app without any kind of log in.

➤ **Advantages of This Design:**

- Simplicity: Working entirely in-memory avoids infrastructure complexity.
- Flexibility: Easy to extend with new news types.
- Maintainability: Each class has clear, well-defined responsibilities.
- Scalability: The structure easily accommodates new features.

➤ **Logical Structure:**

- Models: News (base), InternationalNews (specialized).
- Managers: Section (organizes news).
- Interface: Homepage (presents information).

- This approach leverages OOP principles to create a cohesive system where components collaborate while maintaining independence, ideal for projects requiring clarity and evolution capability. The complete in-memory implementation ensures lightweight operation.

3.1 Preliminary directory structure:

```
├── src/
│   ├── Homepage.java    # Manages featured news and sections (composition)
│   ├── News.java        # Base class (encapsulation)
│   ├── NewsImage.java   # Represents the image associated with a news (encapsulating)
│   ├── InternationalNews.java # Derived class (inheritance)
│   ├── Section.java     # Groups news (composition)
│   └── InternationalNewsSection.java # Specialized section (polymorphism)
├── Workshop-2.pdf       # Design documentation
└── README.md           # Project overview
```

4. Initial code snippets illustrating class definitions, placeholder methods, or abstract classes:

- These are contained in src folder.

4.1 UML-to-Code Correspondence:

Encapsulation (Data Protection)

- **UML Shows:** Private attributes (-country, -title) and public methods (+getSummary()).
- **Code Implementation:**
 - News class uses private String title with public setTitle() (validates non-null).
 - InternationalNews hides country/region but exposes them via validated setters (setCountry() checks length ≤ 44 chars).

Inheritance (Specialization)

- **UML Shows:** InternationalNews inherits from News (attributes/methods under "Inheritance" box).
- **Code Implementation:**
 - InternationalNews extends News to reuse title, content, etc. while adding geographic properties.
 - **Design Rationale:** Avoids duplicating shared attributes like date or author.

Polymorphism (Flexible Behavior)

- **UML Suggests:** Potential method overriding (getLatestNews() differs in InternationalNewsSection).
- **Code Implementation:**

- InternationalNewsSection overrides getLatestNews() to exclude Colombian news via isInternationalNews().
- **Design Rationale:** Enables sections to customize filtering while maintaining a consistent interface.

Composition (Relationships)

- **UML Shows:** Section contains List<News> (aggregation).
 - **Code Implementation:**
 - Section manages newsList to group news thematically.
 - **Design Rationale:** Decouples news storage (Section) from news logic (News).

Design Rationale Explained

System Flow Alignment

1. **Creation:** Matches UML's News/InternationalNews initialization.
2. **Organization:** Section subclasses (like InternationalNewsSection) implement UML's getLatestNews() for their specific logic.
3. **Presentation:** Homepage (implied in UML) queries sections to display results.

Mockup:

- [Mockup](#).

References:

- [Viasure](#).

- [Leanmind](#).