UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Universidad Distrital Francisco José de Caldas

Departament of Computer Science

Systems engineering

Technical Report: Notizo
A Modular Digital News Plataform using OOP

Angie Nathaly Cisneros Guerrero 20242020084

Luis Sebastian Correa Barreto 20242020085

July 11, 2025

Declaration

Declaration of Originality and Authorship

We, **Angie Nathaly Cisneros Guerrero**, student ID **20242020084**, and **Luis Sebastián Correa Barreto**, student ID **20242020085**, hereby declare that:

- This technical report titled **"A Modular Digital News Platform using Object-Oriented Programming"** represents our own original work.
- All contributions from other authors, researchers, and sources have been properly acknowledged through citations and references.
- The content of this report has not been previously submitted for academic credit at this or any other institution.
- All experimental results, data analyses, and software implementations were conducted by the authors unless otherwise specified.
- We understand the consequences of academic dishonesty and confirm that this work complies with the university's academic integrity policies.

Signature: _____ **Date:** _____
Angie Nathaly Cisneros Guerrero July 11, 2025

Signature: _____ **Date:** _____
Luis Sebastián Correa Barreto July 11, 2025

Submitted in partial fulfillment of the requirements for the
Systems Engineering Program
at
Universidad Distrital Francisco José de Caldas

Abstract

This report presents the development of NOTIZO, a digital news platform designed using Object-Oriented Programming (OOP) principles. The main purpose is to demonstrate how OOP can effectively address challenges in digital journalism such as scalability, maintainability, and data protection by creating a modular and adaptable system. The project also serves as a technical reference for developers working on similar applications and explores the transition from traditional print to digital media formats.

The methodology applies key OOP concepts—encapsulation, inheritance, polymorphism, and composition—to organize content, manage data efficiently, and enable modular, flexible system architecture. Features include region-specific filtering (excluding national news in international sections), and a simple, intuitive interface developed with Java Swing. The system is tailored for educational use and includes minimal but practical documentation, making it a suitable model for academic and training environments.

As a result, NOTIZO presents a lightweight in-memory academic prototype that effectively demonstrates key software design strategies. Although it has limitations, such as the absence of persistent storage and limited support for multimedia content, it serves a dual function: it acts as a functional news platform and as a reference guide for developers working on similar web applications. Its architecture is designed to scale with content growth and provide an intuitive browsing experience for users. The accompanying technical report lays out the system design in detail, describing how object-oriented programming principles were applied to address both functional requirements and non-functional needs, including maintenance, security, and scalability. In addition, the evolution of the project from initial conception to final implementation is documented, highlighting the design refinements and decision-making processes that guided its development.

Keywords: object-oriented design, news platform, software architecture, Java implementation.

Contents

List of Figures

List of Tables

Glossary

- **OOP (Object Oriented Programming):** A programming paradigm that structures software development around objects, which encapsulate both data (attributes) and related behaviors (methods). This approach promotes a more natural and modular representation of real-world entities, facilitating code reuse, extensibility and long-term system maintenance.
- **SOLID Principles:** A set of five software design principles (SRP, OCP, LSP, ISP, DIP) that seek to improve the robustness, flexibility, maintainability and scalability of object-oriented software systems.
  - **SRP (Single Responsibility Principle):** Each class has a unique and well-defined responsibility. For example, the News class is exclusively responsible for the data and behaviors of an individual news item, which minimizes the impact of changes.
  - **OCP (Open/Closed Principle):** Software entities should be open for extension (allowing new functionality to be added) but closed for modification (without altering the existing code base). This is achieved through inheritance and abstraction
  - **LSP (Liskov Substitution Principle):** Objects in a program must be able to be replaced by instances of their subtypes without altering the correctness of the program. The abstract class Section and its subclasses such as InternationalNewsSection are an example of this.
  - **DIP (Dependency Inversion Principle):** High-level modules should not depend on low-level modules; both should depend on abstractions. The Homepage depends on the abstraction Section instead of concrete implementations.
- **UML (Unified Modeling Language):** Standard graphical language for visualizing, specifying, constructing and documenting the artifacts of a software system. Used for class and sequence diagrams in the project.
- **CRC Cards (Class-Responsibility-Collaborator Cards):** Design tool used in the conceptual stage to identify the classes of a system, their responsibilities and how they collaborate with each other.
- **CMS (Content Management System):** Software application that allows users to create, edit, organize and publish digital content without the need for technical knowledge. Its absence is a recognized limitation of the current project
- **GUI (Graphical User Interface):** Visual component of the system that allows user interaction through graphical elements. The NOTIZO prototype was implemented using Java Swing

- **Volatile Memory Storage:** Type of data storage where the information is stored only while the program is running and is lost at the end of the program. It is a conscious limitation of the project to simplify the initial development.

Chapter 1

Introduction

In the Internet era, the traditional media (press, radio and television) have been fully introduced into a highly competitive and dynamic online environment, which has generated the so-called digital journalism, which, as its name suggests, is a journalistic practice that is developed exclusively in the digital environment. It uses the Internet as its main distribution platform, which allows it to reach a global audience instantly.

To develop a good digital environment, it is relevant to apply the paradigm of object-oriented programming, since for a project such as the digital newspaper it is necessary to meet demands such as scalability and data protection, which with concepts such as encapsulation and modularity of the code would be solved and helps us to detect errors in the code and its maintenance.

Taking into account the problems that the project seeks to solve, keeping people informed in an easy, accessible and international way, the fact that the news can be classified saving search time compared to traditional newspapers, the fact that it is environmentally more sustainable instead of mass producing newspapers to meet the requirement of scalability, wasting resources such as ink and paper, also that in a digital environment is much more flexible in terms of the type of files that can be included and the ability to update or correct files. For the problems that the project seeks to solve it is convenient to use the concepts of poo.

With this technical report we hope to document the design and implementation of the object-oriented programming paradigm in our digital newspaper project, explaining the system architecture and design decisions, serving as a guide for other developers. Detailing aspects such as project objectives, background, limitations, methodology and results of the project.

1.1 Background

This new digital environment offers distinctive features such as the integration of text, audio, video, graphics, etc., navigation through links, audience participation, and especially the immediacy and constant updating of news, allowing access to information from anywhere and instantaneously. In addition, digital journalism has become a more environmentally sustainable alternative by dispensing with paper and ink, offering greater flexibility in terms of file

types and ease of updating or correcting information.

The transition to digital also presents challenges. To effectively replicate the functionality of an online newspaper and meet the expectations of users and the medium itself, robust and efficient digital platforms are required. Specifically, imperative needs arise such as scalability for an increasing volume of content, intuitive and accessible navigation, a modular and maintainable code base, and adequate data protection. Efficient organization of content, enabling news sorting and search, is equally crucial to enhance the user experience.

To address these challenges and needs in software development, especially in the context of complex web applications such as a digital newspaper, Object Oriented Programming (OOP) emerges as a fundamental and relevant paradigm.

In the context of a digital newspaper, the application of OOP facilitates the management of different types of news and sections, the structured representation of elements such as images, and the organization of content on the home page, directly helps to solve scalability and data protection demands, facilitating error detection and code maintenance and taking into account the specific problem addressed by our project, which is the need to create a digital platform that replicates the functionality of an online newspaper, similar to "El Espectador", with the objective of keeping the user informed in an easy, accessible and international way. This implies the ability to visualize main and highlighted news, access different sections such as breaking news or international news, and see the complete detail of the articles.

The need to be able to sort news to save search time and the convenience of a more flexible and sustainable digital environment reinforce the relevance of the project. To address these functionalities and non-functional requirements such as scalability and maintainability, the Object Oriented Programming approach has been adopted and documented, as detailed in this report.

1.2 Objectives

General objective: objective To comprehensively document the technical design and implementation of the Object Oriented Programming paradigm in the digital newspaper project, detailing the system architecture and design decisions adopted to serve as a guide and reference for other general developers.

Specific objectives:

- Detail how the fundamental principles of OOP (Encapsulation, Inheritance, Polymorphism and Composition) have been applied in the design and structuring of the project classes.
- Explain the rationale behind key design decisions, demonstrating how the application of OOP concepts contributes to meeting the functional and non-functional requirements of the project, such as scalability, maintainability and data protection.

- Documenting the evolution of the project design, highlighting the changes and refinements made from the initial conceptual stages to the current technical blueprint defined with OOP.
- Provide a clear and detailed reference document that facilitates the understanding of the code base and system architecture to other developers, enabling their effective contribution and maintenance.

Chapter 2

Literature Review

2.1 Foundations of Object-Oriented News Systems

The design of news platforms using Object-Oriented Programming (OOP) principles builds upon established software engineering paradigms. Our implementation draws from three key theoretical foundations:

- **Encapsulation:** Following Meyer's Meyer (1997) principle of information hiding, our `News` class protects attributes while exposing controlled interfaces
- **Class Hierarchy:** The specialization of `InternationalNews` from the base `News` class exemplifies Liskov's Substitution Principle Liskov and Wing (1994)
- **Component Design:** Our section-based organization aligns with Szyperski's component software theory Szyperski (2002)

2.2 Related Work in News Platform Architectures

2.2.1 Content Organization Patterns

As shown in Table 2.1, our design compares favorably with existing approaches:

2.2.2 User Experience Considerations

Our user stories reflect established patterns in news platform design:

1. **Homepage Layout:** Confirms Nielsen's Nielsen and Budiu (2011) findings about prominent article placement
2. **Section Navigation:** Implements Card Sorting techniques described by Spencer Spencer (2009)
3. **Content Filtering:** Extends international news handling beyond basic implementations

2.3 Technical Implementation Approaches

2.3.1 Class Design Methodology

Our CRC (Class-Responsibility-Collaborator) cards follow Wirfs-Brock's Wirfs-Brock and McKean (2003) methodology for:

Table 2.1: Architectural Comparison of News Platforms

Approach	Advantages	Limitations
Monolithic (Our implementation)	▪ Simplified in-memory operation ▪ Clear class responsibilities	▪ Limited horizontal scalability
Microservice (Industry standard)	▪ Independent scaling ▪ Polyglot persistence	▪ Increased complexity

- Defining clear class boundaries
- Establishing well-defined collaborations
- Maintaining single responsibility principle

2.3.2 Sequence Diagram Conventions

The interaction patterns shown in our diagrams adhere to:

- UML 2.0 standards *Unified Modeling Language (UML) Version 2.0* (2005)
- Fowler's Fowler (2004) distilled notation practices
- Message sequencing conventions from Rosenberg Rosenberg and Stephens (2007)

2.4 Research Gaps and Contributions

While existing literature covers general OOP patterns, our project makes specific contributions:

- **Lightweight Implementation:** Documents an educational-grade in-memory architecture not covered in production-focused literature
- **Regional Adaptation:** Addresses the Colombian context through specialized filtering in `InternationalNewsSection`
- **Minimalist Documentation:** Provides a template for concise technical reporting in academic settings

Chapter 3

Scope

3.1 Boundaries of the Research

This technical report focuses on the design and implementation of an object-oriented news platform replicating core functionality of "El Espectador" website. The scope encompasses:

3.1.1 Included Elements

- **Core Functionality:**
  - Homepage news display (main and featured articles)
  - Section-based news organization (latest, international)
  - Article viewing with full details
- **Technical Implementation:**
  - Class hierarchy following OOP principles
  - UML design and sequence diagrams
  - In-memory data management
- **Quality Attributes:**
  - Code maintainability and modularity
  - User interface accessibility
  - Documentation standards

3.1.2 Excluded Elements

- **Extended Features:**
  - User authentication/authorization
  - Commenting system
  - Search functionality
- **Infrastructure:**
  - Database persistence
  - Cloud deployment

  - Load balancing

- **Non-Functional Aspects:**
  - Performance benchmarking
  - Security auditing
  - Cross-browser compatibility

3.2 Limitations

The current implementation has several deliberate limitations that define the project scope:

Table 3.1: Implementation Boundaries

Area	Limitation
Data Persistence	Volatile in-memory storage only
Scalability	Single-node implementation
Internationalization	Basic country filtering without multilingual support
Content Types	Text and images only (no video/audio)
Administration	No CMS interface for content management

3.3 Rationale for Scope Decisions

The boundaries were established based on:

1. **Educational Focus:** Prioritizing OOP principles demonstration over production-ready features
2. **Time Constraints:** 6-week development cycle per user story estimates
3. **Resource Availability:** Limiting to Java Standard Edition capabilities
4. **Project Objectives:** Aligning with the core requirement of replicating basic news portal functionality

This scoping ensures the project remains focused on its primary goal: demonstrating object-oriented design patterns in a news platform context while providing a foundation for potential future extensions.

Chapter 4

Assumptions

4.1 About Users

- **User needs:** It is assumed that users are looking for up-to-date information, organized by sections or categories, with intuitive and accessible navigation. This includes the need to display main and featured news on the home page, access different sections (such as breaking news or international news), and view the full detail of articles. It is also assumed that news classification is necessary to save search time. The accessibility of information at any time and place with an Internet connection is also a key assumption related to user needs in the digital environment.
- **User behavior:** It is assumed that users have a certain level of familiarity with navigating websites and interacting with digital content. In addition, it is expected that they will want to actively participate, taking advantage of the interactivity offered by digital journalism, by sharing content.
- **Interest in content:** It is assumed that a significant portion of the audience has an interest in consuming news, especially in the digital format that allows global and instantaneous reach. Particularly in the Colombian context, it is assumed that there is considerable interest in accessing both local and international news. The project is based on the assumption that there is an audience that seeks to stay informed in an easy, accessible and international way.

4.2 About Design and Implementation (OOP)

- **Adequacy of the OOP paradigm:** It is assumed that the Object Oriented Programming paradigm is relevant and adequate for the development of a digital newspaper. It is assumed that OOP facilitates content organization, code reuse and system maintenance. It is also assumed that this paradigm is effective in meeting key non-functional demands such as scalability and data protection, as well as aiding in error detection.
- **Correct application of OOP principles:** It is assumed that the fundamental principles of OOP, including encapsulation, inheritance, polymorphism and composition, have been correctly applied in the design and structuring of the project classes.

  It is assumed that the class of the system (News, Section, Homepage, InternationalNews, NewsImage, InternationalNewsSection) and their relationships (such as the composition between Section and News, or the inheritance between InternationalNews and News) have been clearly identified.

It is assumed that the structure of the code has been designed in a modular fashion, allowing different components to function with some independence and that modifications in one area do not significantly impact other parts of the system.

It is assumed that the implemented design, based on OOP, facilitates the scalability of the system, allowing the management of a greater volume of content and users, as well as the addition of new functionalities.

4.3 About the Development Methodology

- **Clear Understanding of Requirements:** It is assumed that there has been a clear and accurate understanding of the functional and non-functional requirements of the digital newspaper project. This understanding is aligned with the objectives defined for the project and considers the context of digital news consumption. Adherence to the specific user stories defined for the project is assumed to validate this understanding.
- **Validity of the Tools:** It is assumed that the tools and technologies selected for the development of the project, such as the Java implementation, are adequate and will work as expected to build the digital platform. The decision to use in-memory data storage in the initial phase is based on the assumption that it is sufficient for the purposes of the initial project and minimizes infrastructure complexity in the short term, although avenues for future migration to persistent storage are contemplated.

Chapter 5

Methodology

5.1 Development Process

The project followed an iterative object-oriented methodology with three key phases:

1. **Conceptual Design** (Workshop 1)
    - CRC card modeling
    - Initial user stories
    - Functional requirements specification
2. **Technical Design** (Workshop 2)
    - UML diagram refinement
    - Class responsibility updates
    - Sequence diagram development
3. **Implementation**
    - Java code generation
    - In-memory data modeling
    - Validation against acceptance criteria

5.2 Modeling Techniques

5.2.1 UML Diagrams

The core design artifacts included the following UML diagrams:
Key diagram elements:

- Class diagram demonstrates:
    - Implementation (`InternationalNews` → `News`)
    - Agregation (`Section` → `News`)
    - Encapsulation (private fields with public methods)
- Sequence diagrams illustrate:
    - Homepage initialization
    - News retrieval process
    - International news filtering

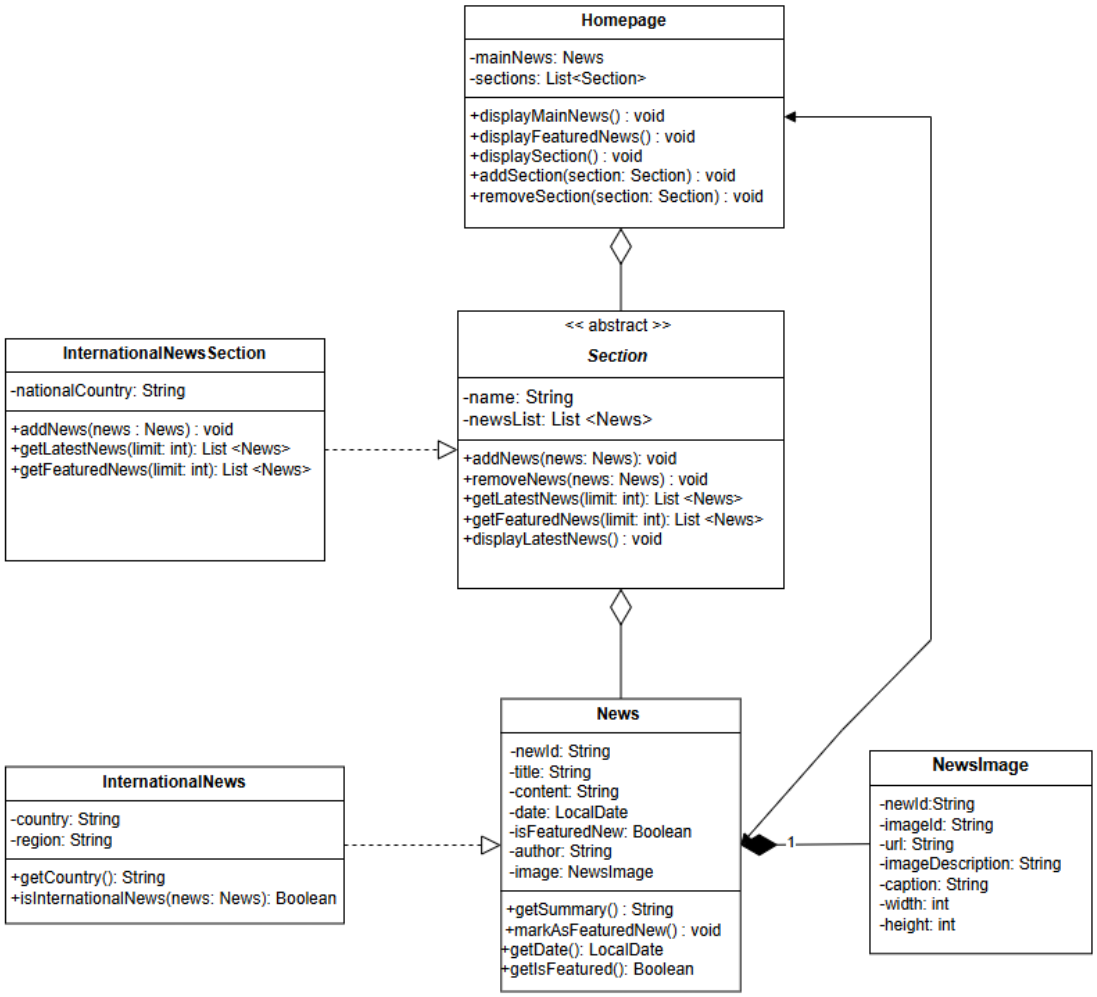

Figure 5.1: Class diagram showing inheritance hierarchy and composition relationships

5.3 Implementation Methods

5.3.1 Class Development

Each class was implemented following this protocol:

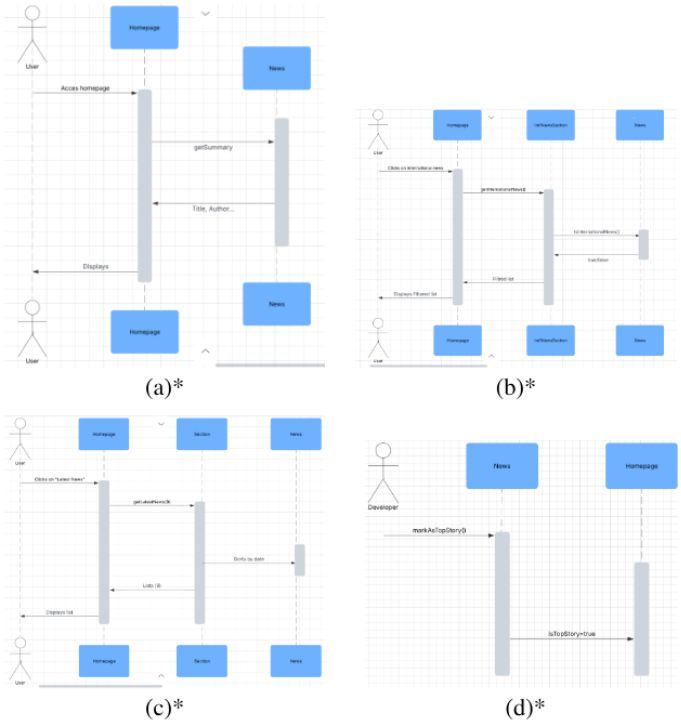

Figure 5.2: Sequence diagram for homepage news display workflow

Table 5.1: Class Implementation Methodology

Step	Activities
1. Stub Creation	- Define class signature - Declare private fields
2. Method Implementation	- Getters/setters with validation - Core functionality methods
3. Relationship Establishment	- Inheritance (extends) - Composition (instance fields)
4. Testing	- Unit test each method - Verify collaborations

5.4 Validation Approach

The system was validated through:

- **User Story Verification:**
  - Manual testing against acceptance criteria
  - Checklist-based validation
- **OOP Principle Adherence:**

  - Encapsulation audits (field visibility checks)
  - Polymorphism verification (method overriding tests)
  - Composition validation (object graph inspection)

- **Code Quality Metrics:**
  - Class cohesion analysis
  - Coupling measurement
  - Documentation completeness

This methodology provides a reproducible blueprint for similar OOP projects, with UML diagrams serving as the central design artifacts that bridge requirements and implementation.

5.5 Java Swing for the user interface

For the implementation of the user interface, Java Swing was chosen, with the objective of building a graphical prototype focused on simplicity and functionality, facilitating an intuitive and accessible navigation. The main structure was articulated around a JFrame as a top-level container, housing the essential components of the user interface. A JTextArea was used as the main display area for the news content, configured in read-only mode as no direct user input was required. Navigation was managed through a JMenuBar that included a "Sections" menu and a JButton for actions such as "Change Section", which triggered a data reload. The interface design took advantage of BorderLayout for a clear organization of the components, placing the central text area and the refresh button at the bottom. Components such as CenterPanel managed the different views (StartPanel, NewsView) via CardLayout, allowing smooth transitions between news sections. In addition, the handling of pre-saved images in a folder, which the application searched for and displayed according to the name, was contemplated.

The connection of this graphical interface with the business logic of the system was designed under a principle of strict separation of responsibilities, completely delegating the business logic to the controller layer. This implies that the view, when requiring updated information, simply requests the data from the controller, instead of directly accessing the model classes. The controller acts as a key intermediary, receiving events generated by the user interface and translating them into operations for the model, then returning formatted data ready for display. This architecture ensures effective decoupling between the user interface, business logic and persistence layers, which simplifies future modifications or extensions and maintains the layered structure of the system. The conscious application of SOLID principles, especially the Single Responsibility Principle (SRP) and the Dependency Inversion Principle (DIP), reinforced this modularity, allowing high-level components such as Homepage (which belongs to the presentation layer) to orchestrate the visualization without relying directly on the concrete implementations of the sections, but rather on their abstractions.

Chapter 6

Results

Along with the digitization of media and news, there is a new need to create robust and efficient platforms, which also support the high traffic of people (scalable), with intuitive and accessible navigation, and for this a modular and maintainable code. This transition from traditional to digital media has transformed journalism into a competitive environment that uses the Internet as its main platform, demanding solutions that handle the growing volume of content and allow efficient organization for sorting and searching news, reducing search time compared to traditional newspapers. In addition, the project addresses the environmental unsustainability of the mass production of traditional newspapers, offering a more flexible and sustainable digital alternative that dispenses with paper and ink.

With that in mind, the NOTIZO digital platform is proposed, a modular digital news platform that applies the principles of the object-oriented programming (OOP) paradigm. The goal is to keep the user informed with accessible news, with specific news sections and featured news that are tolerant to updates, prioritizing relevant information. Users can navigate through different news categories via a menu, including a section dedicated to the 9 most recent news items ("Latest News") sorted by descending date. There is also a specialized section for International News, which filters and displays only global content, excluding national news. Clicking on a headline provides access to the full details of the article, and the system generates concise summaries for a quick overview of the content.

Figure 6.1: Final Notizo Interface

6.1 intuitive user interface

Java Swing was used for the development of the intuitive user interface. A prototype graphical user interface (GUI) was developed, focusing on simplicity and functionality, allowing easy and accessible navigation without requiring validations or user input parameters. The interface is based on a JFrame structure as the main container, a JTextArea as the primary display area for the news content (configured in read-only mode), a JMenuBar for the navigation structure (with a "Sections" menu) and a JButton for reloading data, located in the bottom panel. The GUI design follows a strict separation of responsibilities, delegating all business logic to the controller layer, which ensures that the view simply requests updated data without directly accessing the model classes.

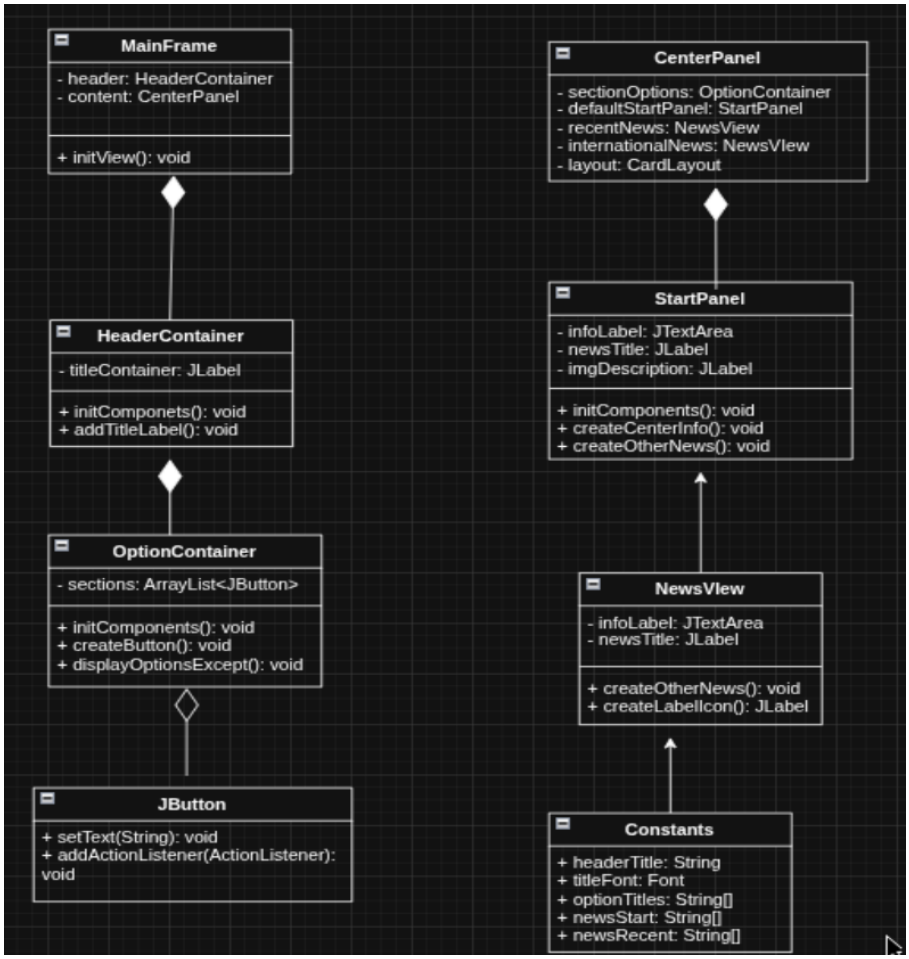

Figure 6.2: Swing-based GUI Prototype

6.2 adherence to OOP principles

The NOTIZO platform demonstrates a rigorous application of Object Oriented Programming (OOP) principles, resulting in a robust, efficient and adaptable system. This is evidenced by the implementation of SOLID principles. Encapsulation is ensured by protecting the internal attributes of News and NewsImage with private fields and exposing them through controlled methods, which includes thorough validations to safeguard data integrity and prevent corruption. This practice also reinforces the Single Responsibility Principle (SRP), as News and

NewsImage focus solely on managing their own data and inherent behaviors.

Inheritance is effectively implemented with InternationalNews extending News, allowing reuse of common attributes and methods such as getSummary(), while adding specific properties such as country/region. This avoids code duplication and allows specialization without modifying the base, exemplifying the Open/Closed Principle (OCP), making it easy to add new functionality through extension. Polymorphism is key to flexibility, allowing methods such as getSummary() to adapt to subclasses and InternationalNewsSection to overwrite news filtering logics, ensuring the Liskov Substitution Principle (LSP) by allowing subclasses to be substituted without altering the correctness of the program. Finally, the Composition is observed in Sect.

Table 6.1: Functional Requirements Verification

Functional Requirement / User Story	**Implementation Status**
Display of main news on the home page	Implemented
Display of featured news on the home page	Implemented
Organization of news by sections	Implemented
Access and display of specialized sections (e.g. International)	Implemented
Display of the complete detail of an article when clicking	To be implemented
Generation of concise article summaries	To be implemented

Chapter 7

Discussion

The choice of Object Oriented Programming (OOP) for the development of the NOTIZO platform is confirmed as the right and fundamental architectural decision to solve the challenges inherent to digital journalism, such as scalability, maintainability, data protection, support for multimedia content and instant updates. Adherence to SOLID principles is the pillar of this solution, transforming the operation flows from monolithic to modular and adaptable, evidenced by Encapsulation, Inheritance and Polymorphism, ensuring high maintainability and flexibility by allowing the extension of functionalities without modifying the existing code base, which is crucial for scalability in the face of a growing volume of content.
The project serves to demonstrate how OOP theory translates into functional implementations that meet real user needs, improving the clarity, consistency, flexibility and robustness of data flows and operations, while facilitating the extension and adaptation of the system to new requirements and meeting the demands of modern digital journalism.

Compared to other proposals focused on news platform architecture, NOTIZO's approach stands out by prioritizing simplicity and in-memory operations, which represents a clear advantage in educational contexts. This lightweight architecture, focused on simplified operation and well-defined class responsibilities, facilitates both understanding and implementation of the system. The project effectively addresses common challenges such as content organization and the separation between presentation and business logic through strict encapsulation in the News class and functional delegation to Homepage. Unlike more complex production-oriented solutions, NOTIZO offers a practical and accessible alternative that effectively illustrates the fundamentals of software design.

This structural clarity is reflected in its strong adherence to SOLID principles, which turn monolithic flows into modular, adaptable and easy-to-maintain systems. The Single Responsibility Principle (SRP) ensures that each class, such as News or NewsImage, has a well-defined role, which reduces the impact of changes and simplifies debugging. In addition, principles such as Open/Closed (OCP) and Liskov Substitution (LSP) promote extensibility: by defining Section as an abstract class, the system allows new sections to be incorporated without modifying existing code. The use of polymorphism, as in InternationalNewsSection, guarantees the incorporation of specialized logics without altering the base interface. Finally, the Dependency Inversion Principle (DIP) promotes decoupling, allowing Homepage to interact with different sections without knowing their internal implementation. These practices were reinforced with encapsulation audits, polymorphism validation, composition analysis and quality metrics such as cohesion and coupling, confirming a robust, extensible design aligned with good software engineering practices.

Chapter 8

Conclutions

The Object Oriented Programming (OOP) approach in the NOTIZO project effectively addresses the challenges of digital journalism, enabling highly efficient and adaptive content organization and classification. Thanks to a modular structuring, the system is able to accurately classify news, which significantly reduces search time for the user. In addition, its efficient design, which dispenses with physical resources, contributes directly to environmental sustainability by replacing the production of printed newspapers with a fully digital platform.

NOTIZO meets its objectives by developing a functional digital news system, simulating the experience of a modern portal, and documenting the application of design principles. The project findings confirm that rigorous implementation of SOLID principles is key to building a robust, efficient and adaptable system. In this context, the platform overcomes typical challenges in the development of information solutions, such as content organization, separation between presentation logic and business logic, and structural rigidity, thus demonstrating how object-oriented design can offer real and sustainable solutions for digital journalism.

The project also offers minimalist but effective documentation, serving as a concise template for technical reports in academic environments. This combination of adherence to SOLID principles and practical contributions demonstrates how object-oriented design can solve real-world problems and offer an adaptable and efficient model for digital journalism but also presents inherent limitations to its initial scope and academic focus therefore for future research and extensions, data persistence is a critical area, migration to a persistent database (SQL or NoSQL) is indispensable for a real production application. This extension should be designed while maintaining the Dependency Inversion Principle (DIP), Other avenues for future development include expanding support for various types of multimedia content, developing extended user functionality such as authentication, commenting systems and advanced search, and exploring cloud deployment strategies and horizontal scalability to handle increasing traffic volumes.

References

Fowler, M. (2004), *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, 3rd edn, Addison-Wesley.

Liskov, B. and Wing, J. M. (1994), 'Behavioral subtyping using invariants and constraints', *Formal Methods in System Design* **4**(2), 93–122.

Meyer, B. (1997), *Object-Oriented Software Construction*, 2nd edn, Prentice Hall.

Nielsen, J. and Budiu, R. (2011), *Mobile Usability*, New Riders.

Rosenberg, D. and Stephens, M. (2007), *Use Case Driven Object Modeling with UML: Theory and Practice*, Apress.

Spencer, D. (2009), *Card Sorting: Designing Usable Categories*, Rosenfeld Media.

Szyperski, C. (2002), *Component Software: Beyond Object-Oriented Programming*, Addison-Wesley.

Unified Modeling Language (UML) Version 2.0 (2005).

Wirfs-Brock, R. and McKean, A. (2003), *Object Design: Roles, Responsibilities, and Collaborations*, Addison-Wesley.

Appendix A

An Appendix Chapter

A.1 CRC Cards

Class Name: Homepage	
Responsibilities:	**Collaborators**
Display last news	-News -Section

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: InternationalNews	
Responsibilities:	**Collaborators**
Representing and managing international news data	News

Class Name: InternationalNewsSection	
Responsibilities:	**Collaborators**
Manage and filter specifically international news	News

Class Name: NewsImage	
Responsibilities:	**Collaborators**
Represent and manage the data of a news image	News

Figure A.1: CRC Cards

A.2 Activity diagrams

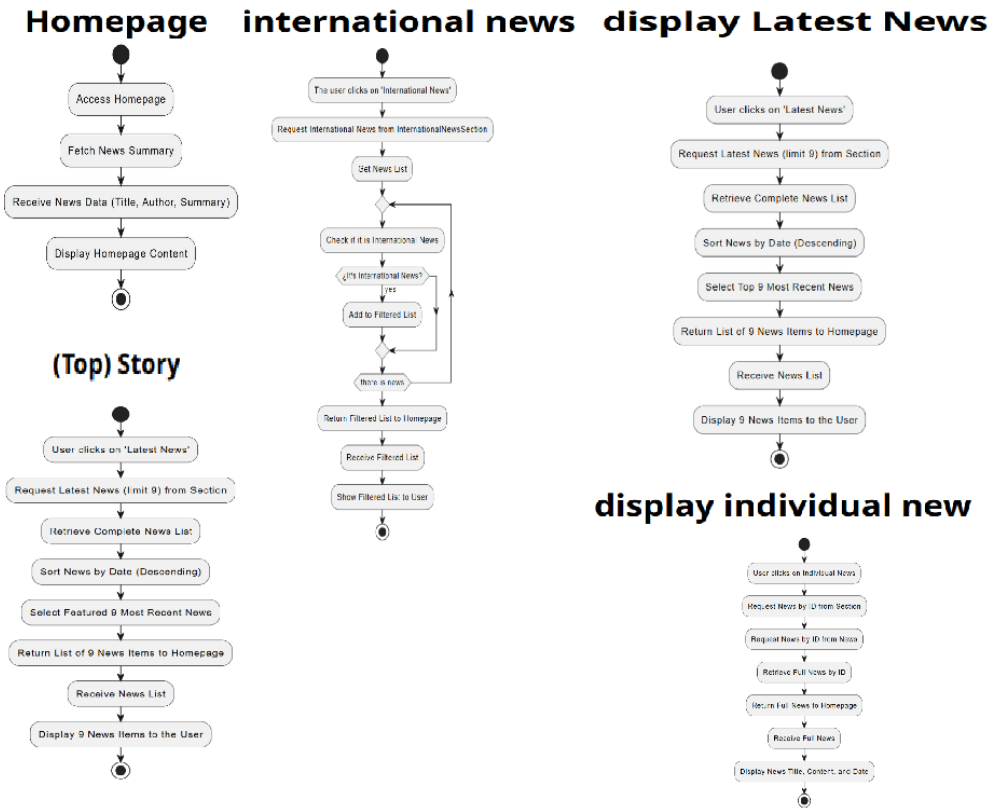

Figure A.2: Activity Diagrams