**Case Study: Enhancing a Flutter News App**

**Background**

Company X is an emerging software development firm specializing in mobile applications. Recently, the company took on a project to develop a news application using Flutter. However, the current implementation has several logical errors affecting its performance and functionality. The goal is to identify these errors, propose solutions, and establish a robust domain model and sequence diagram for the corrected version.

**Domain Model**

The domain model defines the core entities and their relationships:

**Entities & Attributes**

* **News**
  + title (String) → The headline of the news article.
  + publishedDate (DateTime) → The date and time the article was published.
* **Author**
  + name (String) → The full name of the author.
  + email (String) → The contact email of the author.
* **Source**
  + name (String) → The name of the news source (e.g., BBC, CNN).
  + url (String) → The official website or reference link of the source.
* **NewsCategory**
  + categoryName (String) → The name of the category (e.g., Politics, Sports).
  + description (String) → A brief summary of what the category covers.
* **User**
  + username (String) → The unique identifier for a user.
  + email (String) → The user's contact email.

**Cardenality:**

* *A NewsCategory* ***can have*** *multiple News articles, but each News article* ***belongs to only one*** *NewsCategory.*
* *Each News article* ***has*** *exactly one Source, and each Source* ***is associated with*** *only one News article.*
* *An Author* ***can write*** *multiple News articles, but each News article* ***is written by*** *only one Author.*
* *A User* ***can bookmark*** *multiple News articles, and each News article* ***can be bookmarked by*** *multiple Users.*
* *A User* ***can follow*** *multiple Authors, and each Author* ***can have*** *multiple Users following them.*

**Relationships:**

* A NewsCategory contains multiple News articles, where each News article can belong to a category but can also exist independently of it.
* Each News article has exactly one Source, which provides the origin of the news. If a News article is deleted, its Source reference is also removed, as the source is tightly bound to the news article.
* An Author is associated with multiple News articles, meaning an Author can write many articles, and each News article must have an Author who created it.
* A User can bookmark multiple News articles, and each News article can be bookmarked by multiple Users, forming a many-to-many association.

**Sequence Diagram**

This sequence diagram illustrates how the app retrieves and displays news articles.

**Actors & Objects:**

1. **User (initiates the process)**
2. **App (Main Controller) (handles logic)**
3. **Local Storage (checks for cached articles)**
4. **News API (fetches news if needed)**
5. **UI (displays articles)**

**Steps:**

1. **User** opens the app.
2. **App** checks **Local Storage** for cached articles.
   1. **Condition:** If cached articles exist → **App retrieves them** and **passes them to the UI**.
   2. **Else:**  
      3. **App** requests news from the **News API**.  
      4. **API** returns a list of news articles.  
      5. **App** saves articles in **Local Storage** for future use.
3. **App updates UI** with the retrieved articles.
4. **UI renders the articles** for the **User**