

# News Management System



## Faculty of Computer and Information Science Ain Shams University Gen\_35

2022170175	زيد هاني محمد صلاح الدين
2022170618	محمد محمود عبد التواب عبد القادر جاد
2022170137	خالد سيد عبد العزيز محمود
2022170152	رضا محمد رضا محمد شويل
2022170256	على أشرف ابراهيم سيد
2022170379	محمد عبد العليم عبد الحميد مكتي
2022170130	حسام احمد محمد تاج الدين

---

# News Class

---

## Overview

This class is designed to store each article's title, description, rate and more, it also allows the user functionalities as to bookmark, rate the articles and much more.

The admin manipulates the data in the articles by updating it as well as adding new articles.

## Functioning

### Public Methods

- **void rateNews(string username):**
  - allows the user to rate a specific article.
  - **Parameters:** **string username** - The username.
  - **Returns:** **void.**
- **void updateMenu():**
  - allows the admin to update a specific article's title, description, category or date.
  - **Parameters:** **void.**
  - **Returns:** **void.**

- **void displayNewsByCategoryName() :**
  - allows the user to display articles of a specific category.
  - **Parameters:** **void.**
  - **Returns:** **void.**
- **void displayPost() :**
  - Displays all the article's details.
  - **Parameters:** **void.**
  - **Returns:** **void.**
- **void displayCategories() :**
  - display all the categories available.
  - **Parameters:** **void.**
  - **Returns:** **void.**
- **void calculateAverageRate() :**
  - updates the rating of a specific article.
  - **Parameters:** **void.**
  - **Returns:** **void.**
- **vector<News> serachNews(string key) :**
  - allows the user to search for articles by title, description and date.
  - **Parameters:** **the user's input**
  - **Returns:** **a container containing all the matched results.**

- **void displayAllNews(string sort, int type, string detailed, int size = 10):**
  - displays news based either on the rating or date and for whom as well as if it's detailed or not and the size of the displayed articles.
  - **Parameters:** type of sort, user's type, detailed or not and size of displayed articles.
  - **Returns:** void.
- **void displayNewsDetails():**
  - displays everything regarding a specific article from its details to its comments.
  - **Parameters:** void.
  - **Returns:** void.

## Features

### Getters

- **float getRate():** Returns the rate of an article.
- **string getTitle():** Returns the title of an article.
- **string getDescription():** Returns the description of an article.
- **string getCategory():** Returns the category of an article.

- **Date getDate()** : Returns the current date as a string, using the provided separator character.
- **int getSpamCount()** : Returns the number of users that marked that article as spam.

## Setters

- **void updateNewsTitle(string new\_title)** : Sets the title for an article.
- **void updateNewsDescription(string new\_description)** : Sets the description for an article.
- **void updateNewsDate(Date new\_date)** : Sets the date for an article.
- **void updateNewsCategory()** : Sets the category for an article.
- **void updateSpamCount(int)** : Sets the spam count for an article.

## Member Variables

- **string title**: Stores the title of the article.
- **string description** : Stores the description of the article.

- **string category**: Stores the category of the article.
- **Date date**: Stores the date of the article.
- **int spamCount**: Stores the number of people who marked the article as spam.

## Data Structures

- **vector<Comment> comments**: Stores all the comments the article.
- **map<string, int> allRate**: Stores each user's rating and his rate as a key and value for all users.
- **static vector<News> news**: Stores all the articles.
- **static vector<string> categories**: Stores all the categories.
- **static vector<News> valid**: Stores all the valid news for the user in runtime.

## Constructors

- **News(string title, string description, string category, float rate, Date date, int spamcount)**: Constructs a **News** object from a title, a description, a category, a rate of 0, a date and a spam count of 0.

- **News(string title, string description, string category, float rate):** Constructs a **News** object from a title, a description, a category and a rate of 0.
- **News(string title, string description, string category):** Constructs a **News** object from a title, a description and a category.
- **News(string title, string description):**  
Constructs a **News** object from a title and a description.

## Others

### Comparators

- **bool sortNewsByRating(News news1, News news2):**
  - comparator that sorts all the articles descending by rate.
  - **Parameters:** 2 news
  - **Returns:** true if the first has a rating higher than the second and false otherwise.

- `bool sortNewsByDate(News news1, News news2):`
  - comparator that sorts all the articles descending by date.
  - **Parameters:** 2 news
  - **Returns:** true if the first has a date more recent than the second and false otherwise.

---

# Comment Class

---

## Overview

This class is a subclass in the **News** class has a comment body and the username who added that comment.

## Functioning

### Static Functions

- `void commentMenu():`
  - displays a menu for the user to choose to either to add a comment or remove a comment or to display comments for a specific article.



- Parameters: `void`.
  - Returns: `void`.
- 
- `void displayComments():`
    - displays comments for a specific article.
    - Parameters: `void`.
    - Returns: `void`.
- 
- `void addComment():`
    - Adds comment for a specific article.
    - Parameters: `void`.
    - Returns: `void`.
- 
- `void removeComment(int index):`
    - Adds comment for a specific article.
    - Parameters: `the index of the news`.
    - Returns: `void`.

## Member Variables

- **String commentUserName**: Stores the username of the comment.
- **String commentBody**: Stores the body of the comment.

## Features

### Constructors

- **Comment(const string user, const string body)**: Constructs a **Comment** object from a comment body and a username.

### Getters

- **getUserName()**: gets the username of the comment.
- **getBody()**: gets the body of the comment.

### Setters

- **setUserName()**: sets the username of the comment.
- **setCommentBody()**: sets the body of the comment.

---

# User Class

---

## Overview

This class has all data structures and functions related to the User.

## Functioning

### Static Methods

- **void adminMenu():**
  - shows the options for the admin on the command line.
  - **Parameters:** **void**.
  - **Returns:** **void**.
- **void userMenu():**
  - shows the options for the user on the command line.
  - **Parameters:** **void**.
  - **Returns:** **void**.
- **void is\_password\_valid(string& password):**
  - checks if the entered password is valid using regex, it must contain at least 8 characters, have a capital letter, have a small letter and have a number in it.

- **Parameters:** entered password.
  - **Returns:** void.
- 
- **void is\_email\_valid(string& email):**
    - checks if the entered email is valid using regex, it must contain an “@” symbol and a dot.
    - **Parameters:** entered email.
    - **Returns:** void.
- 
- **void ForgetPassword(string username):**
    - Function that sends a code to the user’s gmail to reset their password.
    - **Parameters:** the username.
    - **Returns:** void.
- 
- **void GenRandomPassword():**
    - Function that generates a random code of size 6 that consists of lowercase letters, uppercase letters and digits.
    - **Parameters:** void.
    - **Returns:** void.
- 
- **void SpamNewsFunc():**
    - Function that adds news to spam.
    - **Parameters:** void.
    - **Returns:** void.

- **void spamNewsMenu():**
  - Function that shows the spam menu options for the user on the command line.
  - **Parameters:** void.
  - **Returns:** void.
- **void removeSpamNews():**
  - Function that removes news from spam to normal news.
  - **Parameters:** void.
  - **Returns:** void.
- **void PrintSpam():**
  - Function that prints all the spam news that the user has.
  - **Parameters:** void.
  - **Returns:** void.
- **void PostNews():**
  - Admin function that adds new news.
  - **Parameters:** void.
  - **Returns:** void.
- **void RemoveNews():**
  - Admin function that removes news.
  - **Parameters:** void.
  - **Returns:** void.

- **void BookMarkingMenu():**
  - Function that shows the bookmarking menu options to the user.
  - **Parameters:** void.
  - **Returns:** void.
- **void AddToBookmarks():**
  - Function that bookmarks news.
  - **Parameters:** void.
  - **Returns:** void.
- **void IsInBookmarks(string title):**
  - Function that checks if the news is already bookmarked or not.
  - **Parameters:** title of the article.
  - **Returns:** void.
- **void RemoveFromBookmarks():**
  - Function that removes bookmarked news.
  - **Parameters:** void.
  - **Returns:** void.
- **void PrintBookmarks():**
  - Function that prints all the bookmarks to the user.
  - **Parameters:** void.
  - **Returns:** void.

- **void InterestedCategoriesMenu():**
  - Function that shows the categories menu options to the menu.
  - **Parameters:** **void.**
  - **Returns:** **void.**
- **void AddCategoryToInterested():**
  - Function that adds category to the interested categories.
  - **Parameters:** **void.**
  - **Returns:** **void.**
- **void RemoveCategoryFromInterested():**
  - Function that removes category from the interested categories.
  - **Parameters:** **void.**
  - **Returns:** **void.**
- **void DisplayInterestedCategories()**
  - Function that prints all the interested categories list to the command line.
  - **Parameters:** **void.**
  - **Returns:** **void.**
- **void emailInterestedUsers(string category):**
  - Function that emails the user when a new article is added to the user's interested categories.

- Parameters: **the category**.
  - Returns: **void**.
- 
- **void getAverageRateByTitle():**
    - Admin function that gets the average rating for articles from all users.
    - Parameters: **void**.
    - Returns: **void**.
- 
- **void search():**
    - Function that searches the news by date or keyword in either the description or title.
    - Parameters: **void**.
    - Returns: **void**.
- 
- **void AddCategory():**
    - Admin function that adds new categories.
    - Parameters: **void**.
    - Returns: **void**.
- 
- **void addCategoryAuto(string category):**
    - Admin function that adds new category if when adding news the category does not exist.
    - Parameters: **category to be added**.
    - Returns: **void**.



- **void Register():**
  - Function that makes an account for new users.
  - **Parameters:** **void**.
  - **Returns:** **void**.
- **void Login():**
  - Function that logs in users into their accounts if they exist.
  - **Parameters:** **void**.
  - **Returns:** **void**.

## Features

### Constructors

- **User(string username, string password, string email):** Constructs a **User** object from a username, a password and an email.
- **User():** Constructs a default **User** object.

### Getters

- **getUserName():** gets the username of the user.
- **getPassword():** gets the password of the user.
- **getEmail():** gets the email of the user.

## Setters

- `setUserName()`: sets the username of the user.
- `setPassword()`: sets the password of the user.
- `setEmail()`: sets the email of the user.

## Member Variables

- `string Username`: Stores the username of the user.
- `string Password`: Stores the password of the user.
- `string Email`: Stores the email of the user.
- `int LoginAttempts`: Stores the number of available login attempts for the user.
- `static string currentUsername`: Stores the current logged in user's username.
- `static string currentPassword`: Stores the current logged in user's password.

## Data Structures

- `static map<string, User> users`: Stores all the users.

- **static <string, int> allRate:** Stores each user's rating and his rate as a key and value for all users.
- **static vector<News> news:** Stores all the articles.
- **static vector<string> categories:** Stores all the categories.
- **static vector<News> valid:** Stores all the valid news for the user in runtime.
- **static unordered\_map<string, unordered\_set<string>> bookmarks:** Stores each users' bookmarks.
- **static unordered\_map<string, unordered\_set<string>> interestedCategories:** Stores each users' interested categories.
- **static unordered\_map<string, unordered\_set<string>> spamNews:** Stores each users's spam news.

---

# Database Class

---

## Overview

Saving all data from users and news and store them into files and backup when program run again.

## Functioning

- **void write()** : call the function at the end of the program to save data.
- **void read()** : call the function at the beginning of the program to load the data from file.

---

# Date Class

---

## Overview

The Date class is designed to represent and manipulate dates in a simple and intuitive manner. This class allows for the creation, comparison, and formatting of dates. It provides both constructors and various methods for setting and getting date components, as well as utility functions for string conversion and date formatting.

## Functioning

### Private Methods

- **`void constructDate(string str_date)`** : Parses a string representing a date and sets the internal day, month, and year accordingly.
- **`vector<int> getDayMonthYear(string date)`** : Parses a date string and returns a vector containing the day, month, and year as integers.

### Member Variables

- **`int mday`** : Stores the day of the month.
- **`int month`** : Stores the month of the year.
- **`int year`** : Stores the year.

# Features

## Constructors

- **Date(string date\_str)** : Constructs a **Date** object from a string representation of a date.
- **Date()** : Default constructor that initializes the date to a default value (usually the current date or a specific default date).

## Getters

- **int getDay()** : Returns the day of the month.
- **int getMonth()** : Returns the month of the year.
- **int getYear()** : Returns the year.
- **string fullDate(char sep)** : Returns the full date as a string, using the provided separator character between day, month, and year.
- **static string getCurrentDate(char sep)** : Returns the current date as a string, using the provided separator character.

## Setters

- **void setDay(int \_d)** : Sets the day of the month.

- `void setMonth(int _m)` : Sets the month of the year.
- `void setYear(int _y)` : Sets the year.

## Others

### Comparison Operators

- `bool operator>(Date& date)` : Compares two dates and returns **true** if the current date is greater than the provided date.
- `bool operator==(Date& date)` : Compares two dates and returns **true** if the dates are equal.
- `bool operator<(Date& date)` : Compares two dates and returns **true** if the current date is less than the provided date.

### Friend Functions

- `friend std::ostream& operator << (std::ostream& os, const Date& d)` : Outputs the date in the format 'dd/mm/yyyy'.

### Utility Methods

- `std::string toString() const` : Converts the date to a string in the format 'yyyy/mm/dd'.
- `static Date fromString(const std::string& dateStr)` : Creates a **Date** object from a string in the format 'dd-mm-yyyy'.

---

# Utility Class

---

## Overview

The **Utility** class is a collection of static utility functions that provide various operations such as date format validation, string manipulation, conversion between data types, and debug message output. This class does not require instantiation and all methods can be accessed statically.

## Functioning

### Member Variables

- The **Utility** class does not contain any member variables.

## Features

### Static Methods

- **static bool isDigitChar(char c):**
  - Checks if the given character is a digit between '1' and '9'.
  - **Parameters:** **char c** - The character to check.
  - **Returns:** **true** if the character is a digit, **false** otherwise.
- **static bool checkDateFormat(string date) :**
  - Validates if the date string is in the format '**dd/mm/yyyy**'.
  - **Parameters:** **string date** - The date string to validate.



- **Returns:** **true** if the format is correct, **false** otherwise.
- **static vector<string> getKeywords(string key) :**
  - Splits the given string into keywords based on some criteria (e.g., spaces, commas).
  - **Parameters:** **string key** - The string to split into keywords.
  - **Returns:** A vector of strings containing the keywords.
- **static string toLower(string word) :**
  - Converts the given string to lowercase.
  - **Parameters:** **string word** - The string to convert.
  - **Returns:** A new string in lowercase.
- **static string toString(int num) :**
  - Converts an integer to its string representation.
  - **Parameters:** **int num** - The integer to convert.
  - **Returns:** The string representation of the integer.
- **static int toInt(string str) :**
  - Converts a string to its integer representation.
  - **Parameters:** **string str** - The string to convert.
  - **Returns:** The integer representation of the string.
- **static void dateHandler(string& date) :**
  - Handles and processes the date string in a specific manner (e.g., formatting or validation).
  - **Parameters:** **string& date** - The date string to process.
  - **Returns:** **void**
- **static void DM(string source, string message) :**
  - Outputs a debug message to the console.

- **Parameters:**
  - `string source` - The source of the message.
  - `string message` - The debug message to output.
- **Returns:** `void`
- `static void getDateOrder(string date, int& year, int& month, int& day)` : Extracts the year, month, and day from a date string.
  - **Parameters:**
    - `string date` - The date string in the format 'dd/mm/yyyy'.
    - `int& year` - Reference to an integer to store the extracted year.
    - `int& month` - Reference to an integer to store the extracted month.
    - `int& day` - Reference to an integer to store the extracted day.
  - **Returns:** `void`

---

# Style Class

---

## Overview

The **Style** class is a class that contains functions that set up the console at the runtime as well as style texts.

## Functioning

### Static Methods

- **void setupConsole() :**
  - Sets up the console by adding a background to the window and sets up a fixed width as well as a height.
  - **Parameters:** **void**.
  - **Returns:** **void**.
- **void styleText(string text) :**
  - Styles a given text by center aligning the text as well as adding a background color to it.
  - **Parameters:** **the text to style**.
  - **Returns:** **void**.