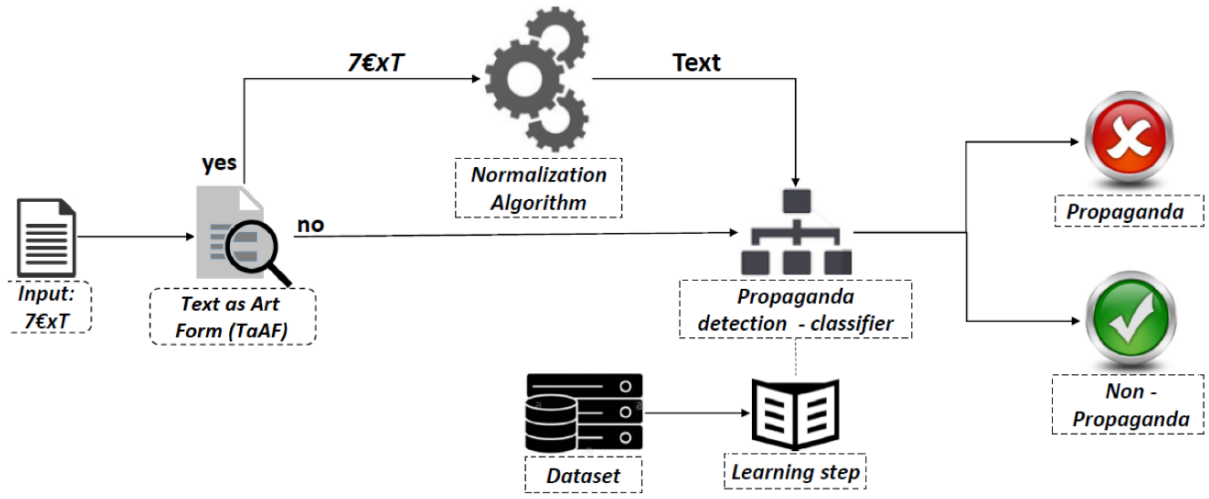
Propaganda Detection Application

	Name	ID
1	Aly Farrag Elmandouh	20100297
2	Amira Ali Mohamed	21100789
3	Aya Adel Saied	21100786

Architecture:



PROJECT DESCRIPTION:

As part of Alamein International University's commitment to promoting critical media literacy and combating misinformation, the Propaganda Detection Application is designed to identify and classify propaganda techniques within text inputs. The system is essential for researchers, educators, and analysts who seek to understand and mitigate the impact of propaganda in various forms of communication.

The focus of this project is to develop a robust system capable of detecting and classifying propaganda within text inputs using advanced machine learning models. By leveraging Transformer architectures, CNN, and LSTM layers, the system is designed to address both binary and multi-label classification tasks, making it a versatile tool for researchers and analysts.

In an era where information is a strategic resource, the Propaganda Detection Application empowers users with sophisticated tools for real-time analysis, helping them make informed decisions and contribute to the broader effort to counter misinformation and propaganda.

REQUIREMENTS:

Functional Requirements

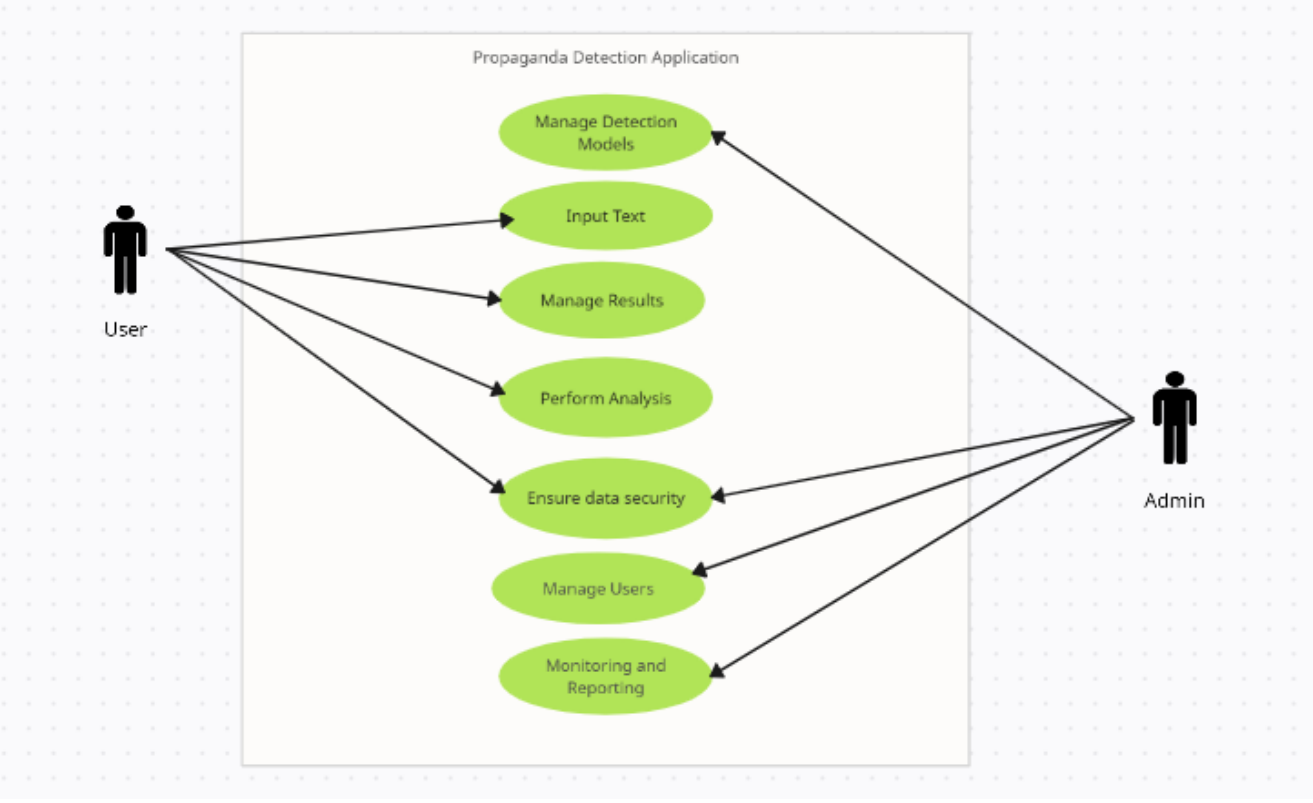
1. **Binary Propaganda Detection**
    - The application must classify input text as either "Propaganda" or "Not Propaganda" using a pre-trained transformer model.
    - The binary detection model must load and run efficiently on a GPU.
2. **Multilabel Persuasion Technique Detection**
    - The application must detect multiple persuasion techniques from input text using a multilabel classification model.
    - The detected techniques should be displayed clearly in the UI.
3. **Text Input and File Upload**
    - Users must be able to input text directly into a textbox for analysis.
    - The application must allow users to upload a .txt file for batch analysis, with results displayed in the UI.
4. **Error Handling**
    - The application must validate input to ensure that text is in Arabic and not empty. If the input is invalid, an error message should be displayed to the user.

- The application must handle exceptions during model loading, predictions, and file operations, displaying appropriate error messages.

Non-Functional Requirements

1. **Performance**
    - The application should load models and make predictions within a reasonable time frame, ideally under 5 seconds for text inputs of up to 500 words.
    - The application should handle large files (up to 5MB) without significant performance degradation.
2. **Usability**
    - The GUI must be intuitive, with clear labels, buttons, and input fields.
    - The application should provide feedback to the user during processing, such as loading indicators or progress bars for file analysis.
3. **Scalability**
    - The application should be designed to easily incorporate additional models or classifiers in the future without requiring major architectural changes.

USE CASE DIAGRAM:



USE CASE DESCRIPTION AND INTERACTION SCENARIOS:

Use Case: Determine Propaganda Presence

Actors:

user

Description:

The user logs into the application and navigates to the Text Input section.

The system provides a text box where the user can input an article for analysis.

The user enters or pastes the text of the article.

The system processes the input text to determine whether it contains propaganda.

Once the analysis is complete, the system displays the result, indicating "Propaganda" or "Not Propaganda."

The user can choose to save the result or export it for further use.

Alternative Courses:

If the input text is empty, the system displays an error message prompting the user to enter text.

If the input text is not in Arabic, the system displays an error message informing the journalist to provide Arabic text.

Use Case: Analyze Persuasion Techniques in Multiple Articles

Actors:

User

Description:

The User logs into the application and navigates to the File Upload section.

The system provides an option to upload a .txt file containing multiple articles.

The User selects the file and uploads it for analysis.

The system validates the file format and begins processing the articles, displaying a progress bar.

Once the analysis is complete, the system displays the detected persuasion techniques for each article.

The User can choose to save the results or export them in a preferred format.

Alternative Courses:

If the file format is not supported, the system displays an error message and prompts the researcher to upload a valid file type.

If the file is empty or contains non-Arabic text, the system displays an appropriate error message**.**

Use Case: Upload and Process Text File

Actors:

User

Description:

The user logs in and navigates to the File Upload section.

The system provides an option to upload a text file for analysis.

The user selects and uploads the file.

The system validates the file format and begins processing the text, displaying a progress bar.

Once the analysis is complete, the system displays the results, including any detected propaganda techniques and their context within the text.

The user can choose to save the results or export them in a preferred format.

Alternative Courses:

If the file format is not supported, the system displays an error message and prompts the user to upload a valid file type.

If the file is empty or contains non-Arabic text, the system displays an appropriate error message.4. Use Case: Manage Results

SUBSYSTEMS

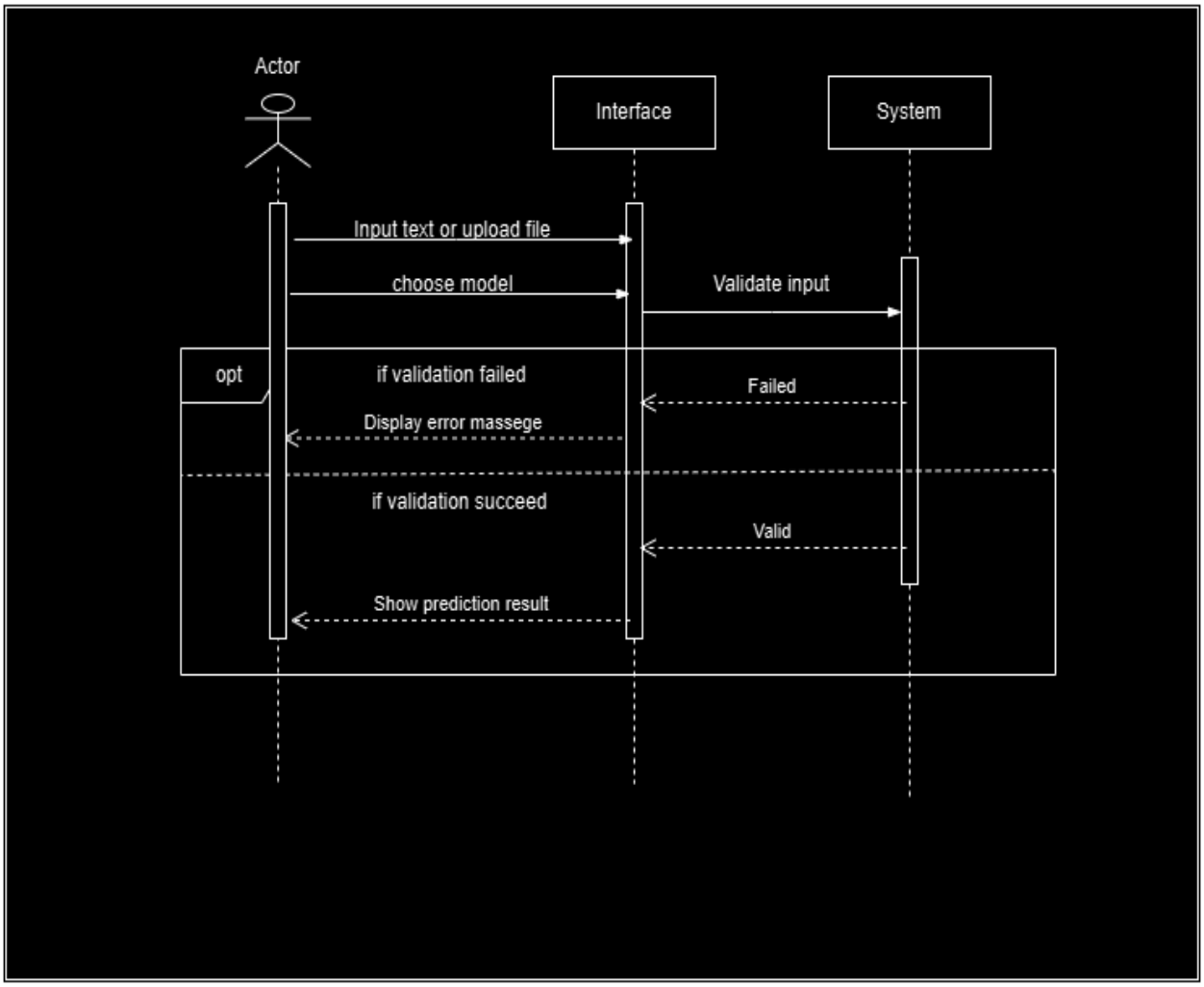
Subsystem Name	Subsystem Function	Subsystem Interface
Model Management	Manage and configure propaganda detection models	public function index(), public function create(), public function store(Request $request), public function update(string $id), public function destroy(string $id)
Text Input Management	Handle text input processing, including manual entry and document uploads, with real-time validation and feedback.	public function upload(), public function validate(), public function process(), public function showresults(string $id)
Classification and Analysis	Implement machine learning models for binary and multi-label classification, providing detailed analysis reports.	public function classify(string $text), public function detect(), public function report(string $id)
Results Management	export analysis results	public function index(), public function filter(), public function export(string $format), public function visualize(string $id)
Security and Privacy	Ensure data security and compliance with regulations, including	public function authenticate(), public

	encryption and user authentication.	function authorize(), public function audit(string $id)
User Experience	Provide a responsive and user-friendly interface, including customizable dashboards and accessibility features.	public function dashboard(), public function customize(), public function help()

TRACEABILITY MATRIX

Feature	Model Management	Text Input Processing	Classification and Analysis	Results Management	Security and Privacy	User Experience
CRUD Operations	✓	✓	✓	✓	✓	
Version Control	✓					
File Upload		✓				
Real-Time Feedback		✓	✓			
Classification			✓			
Visualization			✓	✓		
Export Data				✓		
Encryption					✓	
User Authentication					✓	
Customization						✓
Accessibility						✓

Sequence Diagram:



TEST REQUIREMENTS:

1. **Text Input and Validation:**

TR1: Validate that the system correctly identifies and processes Arabic text input.

TR2: Ensure that the system displays an error message if the input text is not in Arabic.

TR3: Confirm that the system prompts the user with an error if an attempt is made to analyze empty text input.

2. **File Upload and Processing:**

TR4: Validate that the system accepts and processes .txt files containing Arabic text.

TR5: Ensure that the system displays an error message if the uploaded file format is unsupported.

TR6: Verify that the system displays an error message if the uploaded file is empty or contains non-Arabic text.

TR7: Confirm that the system displays a progress bar during file processing and provides results once the analysis is complete.

3. **Propaganda and Persuasion Technique Detection:**

TR8: Validate that the system accurately detects and displays whether a text contains propaganda.

TR9: Ensure that the system correctly identifies and lists all relevant persuasion techniques present in the text.

TR10: Confirm that the "no technique" label is displayed only when no other persuasion techniques are detected.

4. **User Interface and Usability:**

TR11: Validate that the dropdown for model selection functions correctly, allowing users to switch between binary and multilabel detection models.

TR12: Ensure that the text input box and file upload controls are intuitive and accessible.

TR13: Confirm that the system allows users to save or export the analysis results in a preferred format.

TR14: Verify that the GUI layout is visually appealing and user-friendly, with clear labels and buttons.

5. **Error Handling:**

TR15: Validate that appropriate error messages are displayed when the user inputs non-Arabic text or leaves the input field empty.

TR16: Ensure that error messages are clear and guide the user on how to correct the input.

TR17: Confirm that the system handles unexpected errors gracefully without crashing or losing data.

TEST Cases:

ID	Test Case	Input	Expected Result	Status
TC1	Propaganda techniques detection	"Text containing Loaded Language"	"Propaganda detected: Loaded Language"	Pass
TC2	predicts no propaganda	"Text without propaganda"	"No propaganda techniques detected."	Pass
TC3	handling non-Arabic input.	“English text."	"Error: Input text must be in Arabic."	Pass
TC4	input text is empty.	(Empty text)	"Error: Input cannot be empty."	Pass
TC5	the input text exceeds the maximum token length.	"Very long text exceeding max tokens"	"Error: Text exceeds maximum length."	Fail
TC6	An unsupported file format.	"Upload .png file"	"Error: Unsupported file format."	Pass
TC7	model is not loaded correctly.	"Any text input"	"Error: Model not loaded."	Fail
TC10	multiple propaganda techniques are correctly detected	"Text with multiple techniques"	"Propaganda detected: Loaded Language, Name Calling"	Pass