**SOFTWARE REQUIREMENT SPECIFICATION**

VERSION 2.0

October 26, 2024

**UP-TO-DATE (UTD)**

(News Aggregator Web Application)

**SUBMITTED TO**

Professor Gananand Kini

**SUBMITTED BY**

Sumiran Jaiswal

UMD Directory ID: sumi0309

UMD Email ID: [sumi0309@umd.edu](mailto:sumi0309@umd.edu)

Submitted in partial fulfillment of the requirements of ENPM680 (Introduction to Secure coding for Software Engineering)

**Table of Contents**

Table of Contents…………………………………………………...……....i

List of Figures…………………………………………………...……….…ii

1. Introduction……………………………………………………….......1

1.1. Purpose of the Document…………………………………………….......1

1.2. Brief Description…………………………………………………………1

1.3. Glossary…………………….……………………………………………2

1.4. References……………………………………………....……………......2

2.0 Detail Description……………….……………………………………3

2.1 System Working model…………………………………………………...3

2.2 Requirement Specification………………………………………………..4

2.2.1 Use Cases………………………….……….…….…………….…...5

2.2.2 Misuse Cases………………………………………….…………...13

2.3 Functional and Non-Functional Requirements…………………………..19

2.3.1 Functional Requirement……………………………………...19

2.3.2 Non-Functional Requirement………………………………...24

2.3.3 Security Requirements……………………………………….25

2.4 Logical flow of processes………………………………………………..26

2.5 User Characteristics………………………………………………….......26

3.0 Index….……..…………………...……………………………………... 28

**List of Figures**

Figure 1 – System Environment………………………..3

Figure 2 – Abuse Case – A……………...…………….15

Figure 3 – Abuse Case - B……………………………16

Figure 4 – Abuse Case - C………………………...….18

Figure 5 – Threat Model……………...………………18

Figure 6 – Threat Analysis Report…...………………12

Figure 7 - Logical Structure of System………………26

1. **Introduction**
   1. *Purpose of the document*

This document aims to provide a comprehensive view of the News aggregator web application named Up-To-Date. It will describe what the system is for, what it can do, and how it works. This document is meant for both the people interested in the system and the developers working on it.

* 1. *Brief description*

This software system, named Up-To-Date, is a news aggregator web application designed to keep users informed about local events and news in their neighborhood. The application will cater to various user roles, allowing individuals to sign in, submit news articles with timestamps and locations, search for news using relevant keywords, delete news items, and suggest modifications or contribute additional information.

The primary goal of Up-To-Date is to provide a platform where users can stay current with the latest happenings in their area, enhancing community engagement and information sharing. The system will include features to ensure ease of use while effectively managing news content and user interactions, making it a valuable tool for local news aggregation and dissemination.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Author | The person submitting a latest news. |
| Contributor | Anyone who wants to contribute to an already posted news. |
| Database | Collection of all the information monitored by the system. |
| Reader | Anyone visiting the site to read news. |
| Software Requirement Specification | A document that completely describes all of the functions of a proposed system. As an instance, this document. |
| Stakeholder | Any person with an interest in the project who is not a developer. |
| User | Reader or Author. |
| UTD | Up To Date (the news aggregator web application discussed in this document) |

* 1. *Glossary*
  2. *References*

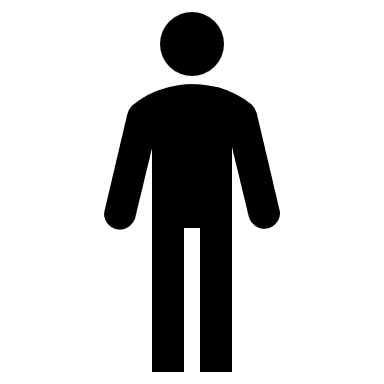
1. **IEEE Std 830-1984** IEEE Guide to Software Requirement Specification.
2. **Perforce Blog.** "How to Write a Software Requirements Specification (SRS) Document." [Online]. Available: <https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document>.
3. **Sample SRS document**, SRSExample-webapp.doc by Joan, Paul Adams, Bobbie Baker and Charles Charlie.
4. **Lucid Chart web application,** used to create the various Use case and Abuse case diagrams.

**2.0 Detailed description**

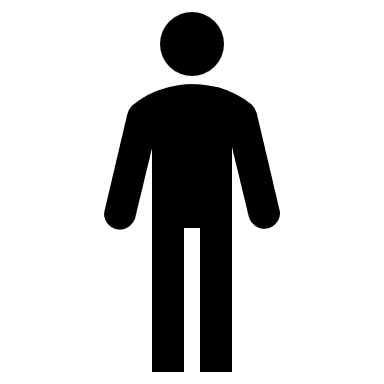
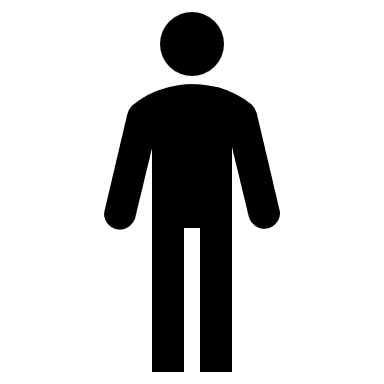
*2.1. System Working Model*

NEWS

FEED

**

**Author**

**

**Reader**

**Contributor**

UP TO DATE

**Fig. 1 System Environment**

UTD is a web application designed to keep users informed and up to date with the latest happenings in their region. The system allows users to interact with the news feed through three different personas, each requiring a sign-in with their respective accounts:

**Author**: Responsible for writing and posting the latest news on the feed. Authors can add news with a headline, date, time, and detailed information. Once the news is posted, it becomes available for others to view.

**Reader**: Focused on consuming content, the Reader can browse through headlines and click on a "Details" button to view the full content of any news article as written by the Author. Additionally, Readers have access to a Search News functionality, where they can type in any keyword related to the news, and the most relevant news articles will be presented to them.

**Contributor**: Contributors play a vital role in maintaining the accuracy and relevance of the news. They can edit or suggest changes to any news article to improve its content or update information.

Each user must sign in with their account to access their respective functionalities. The UTD application is designed to ensure that users always have access to the most current and accurate news in their area, with enhanced search capabilities for readers.

*2.2. Requirement Specification*

This section highlights the use cases for each of the active user, the Author, Reader, and the Contributor along with the detailed use case of each persona.

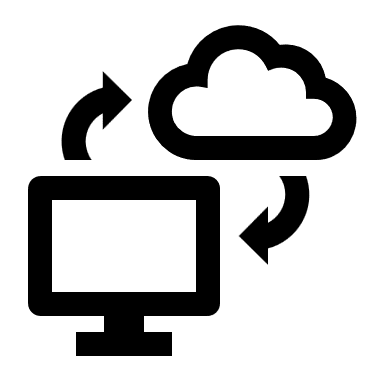
**2.2.1 Use Cases**

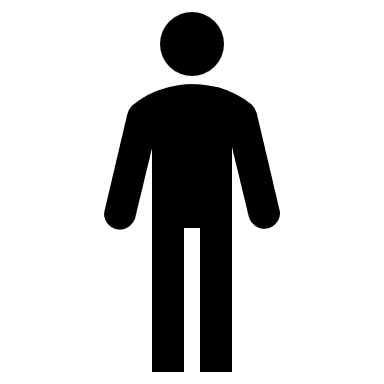
**Author Use Case**

Author is the user responsible for creating and posting news on the feed.

**Use case: Submit the news**

**Diagram**:





**Feed**

**Adding News**

**Author**

**Actors**: Author

**Precondition**: Author must be logged into the system.

**Flow of Events**:

1. The Author accesses the "Create News" interface.
2. The Author enters the following details for the news article:
   * **Heading**: The main title of the news.
   * **Detail Description**: A detailed explanation of the news.
   * **Time**: The date and time associated with the news event.
   * **Place**: The location where the news event occurred.
3. The Author submits the news article for publishing.

**Postcondition**: The news article is added to the feed and is available for Readers and Contributors to view and interact with.

**Alternate Flow:**

* **A1:** The Author decides to save the article as a draft instead of submitting it for publication.
  + The Author selects the "Save as Draft" option.
  + The news article is saved in the drafts section for future editing.

**Error Flow:**

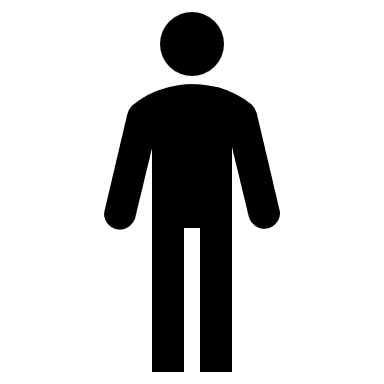
* **E1:** The Author fails to fill in required fields (e.g., heading or detailed description).
  + The system prompts the Author to complete the missing fields and does not submit the article.
* **E2:** The Author encounters a system error during submission.
  + The system displays an error message, and the Author is advised to try submitting again later.

**Reader Use Case**

The Reader is a user who browses, searches, and reads news posted on the feed. Reader can also post some suggestions regarding any particular news along with uploading some articles to the website.

**Use Case: Browse through the feed**

**Diagram**:



**Feed**

**Browsing**

**Reader**

**Actors**: Reader

**Precondition**: Actors must be logged into the system.

**Flow of Events**:

1. The Reader accesses the news feed and views the headlines of various news articles.
2. The Reader can click on a headline to see the full article, which includes the detailed description, time, and place provided by the Author.
3. The Reader can use the Search News functionality to find specific news articles by typing in keywords. The system will display the most relevant articles based on the search query.

**Postcondition**: The Reader successfully views the full details of the news articles they are interested in, either by browsing the feed or using the search feature.

**Alternate Flow:**

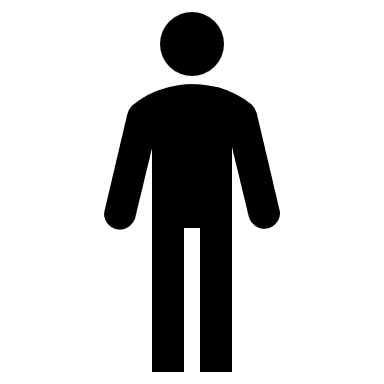
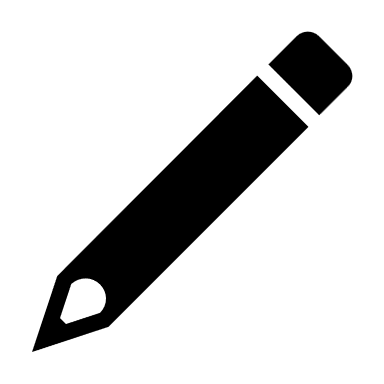
* **A1:** The Reader uses filters to narrow down news articles by category or date.
  + The system displays the filtered list of articles based on the selected criteria.

**Error Flow:**

* **E1:** The Reader’s search query yields no results.
  + The system displays a message indicating that no articles match the search criteria.
* **E2:** The system experiences a loading issue.
  + The Reader sees a loading error message and is prompted to refresh the page.

**Use Case: Post suggestions**

**Diagram**:



**Suggest a change for a news**

**Feed**

**Reader**

**Actors**: Reader

**Precondition**: Actors must be logged into the system.

**Flow of Events**:

1. The Reader accesses the news feed and views a particular news article.
2. The Reader can click on a headline to see the full article, which includes the detailed description, time, and place provided by the Author.
3. The Reader can go to the suggestion page to provide a suggestion regarding the news with its Id.

**Postcondition**: The Reader successfully submits the suggestion and it is visible to them on the webpage.

**Alternate Flow:**

* **A1:** The Reader decides to edit their suggestion before submission.
  + The Reader modifies the suggestion and then submits it.

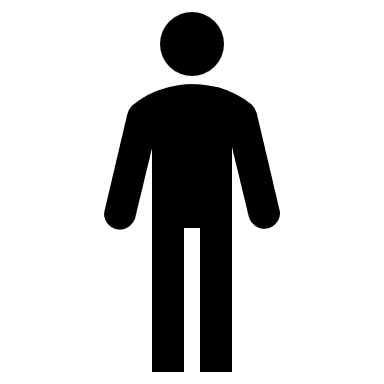
**Error Flow:**

* **E1:** The Reader fails to enter a valid suggestion (e.g., too short).
  + The system prompts the Reader to provide a more detailed suggestion before allowing submission.
* **E2:** The submission encounters a system error.
  + The system displays an error message, and the Reader is advised to try submitting again later.

**Use Case: Read and Post Article**

**Diagram**:





**Add an article**

**Web page**

**Reader**

**Actors**: Reader

**Precondition**: Actors must be logged into the system.

**Flow of Events**:

1. The Reader accesses the Articles page.
2. The Reader reads article of their interest.
3. The Reader can click on a download button to view and download the full article.
4. Reader clicks on create article button and submits their article’s file along with it.

**Postcondition**: The Reader successfully submits the article and it is visible to them on the webpage.

**Alternate Flow:**

* **A1:** The Reader decides to edit their article before final submission.
  + The Reader modifies the article file and resubmits it.

**Error Flow:**

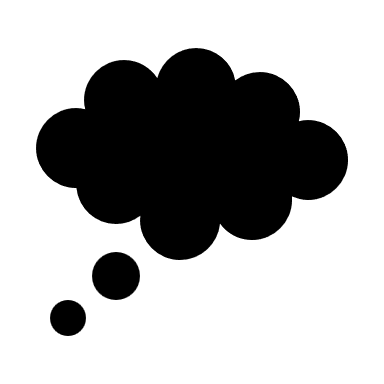
* **E1:** The Reader tries to upload a file in an unsupported format.
  + The system displays an error message indicating the accepted file formats.
* **E2:** The upload fails due to a network issue.
  + The Reader receives an error message and is encouraged to try uploading again.

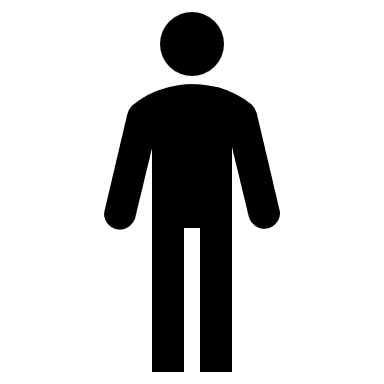
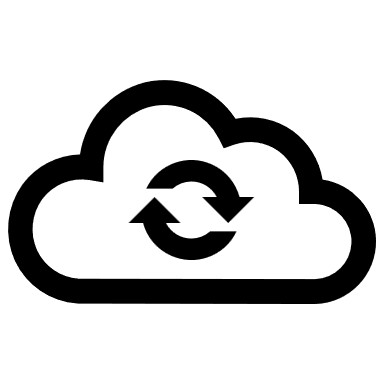
**Contributor Use Case**

The Contributor is a user who can edit and suggest changes to existing news articles on the feed to improve accuracy and relevance.

Use case: Edit or suggest any change

**Diagram**:





**Suggestions**

**Feed**

**Contributor**

**Actors**: Contributor

**Precondition**: Contributor must be logged into the system

**Flow of Events**:

1. The Contributor accesses the news feed and selects a news article that needs editing or updates.
2. The Contributor reviews the article's content, including the heading, detailed description, time, and place.
3. The Contributor makes edits or suggests changes to the article to improve its content or correct any inaccuracies.
4. The Contributor submits the changes.

**Postcondition**: The Contributor's changes are reflected in the news article, ensuring the information is accurate and up to date for all users.

**Alternate Flow:**

* **A1:** The Contributor decides to save their suggested changes as a draft.
  + The changes are saved in a draft status for future editing.

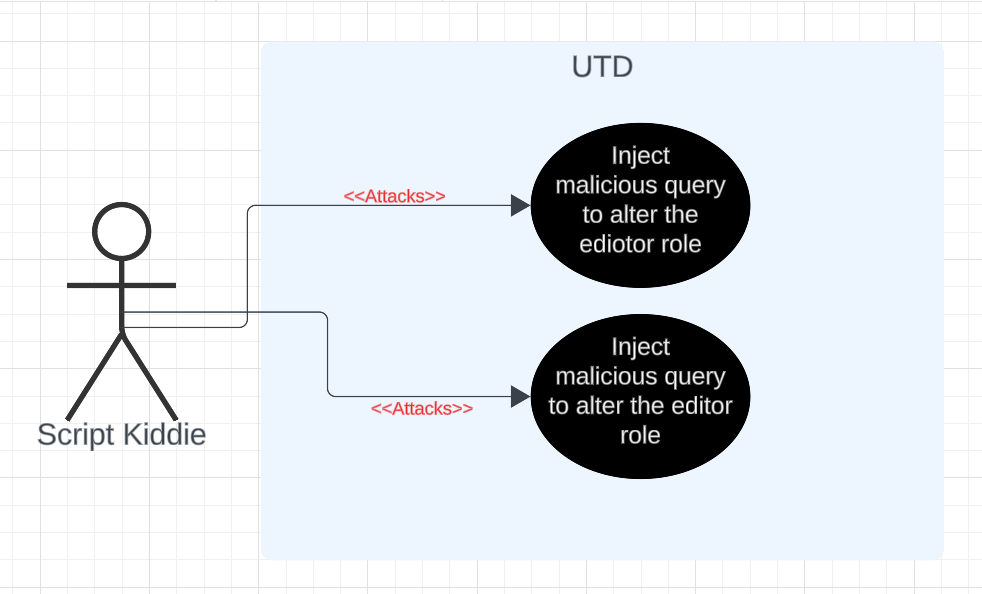
**Error Flow:**

* **E1:** The Contributor fails to make any changes before submitting.
  + The system prompts the Contributor to make changes before allowing submission.
* **E2:** The submission encounters a system error.
  + The system displays an error message, and the Contributor is advised to try submitting again later.

**2.2.2 Misuse Cases**

1. Changing the role of Author/Editor using SQL Injection.

Abuse case diagram:



**Fig. 2 Abuse Case - A**

Name: SQL Injection

Actor: Script Kiddie

Trigger: Script Kiddie Injects the malicious query in the suggestion field.

Preconditions: Script Kiddie has a registered account on the application to post a suggestion.

Postcondition:

* Success postcondition: Script Kiddie fails to Inject the query into the system.
* Failure postcondition: Script Kiddie successfully Injects the malicious query.

Basic Flow:

1. Script Kiddie logs in using the credentials.
2. He/she navigates to suggestions page.
3. He/she puts the malicious query into the input field and clicks on submit button.
4. He/she logs out of the application.

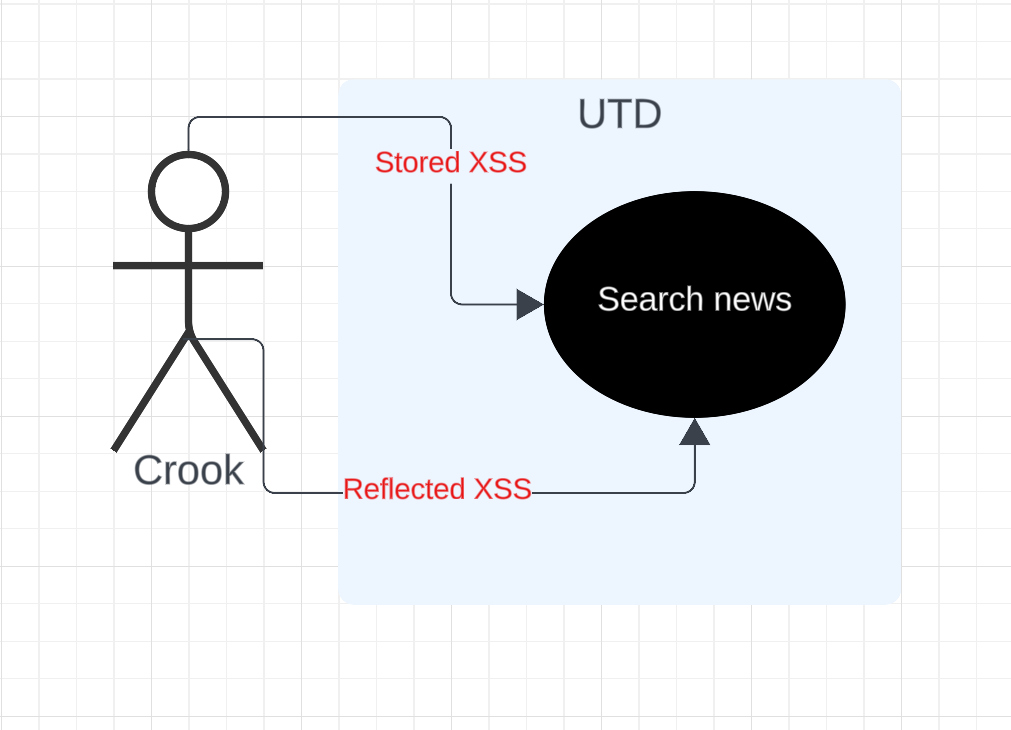
Alternative flow:

1. Script Kiddie tries to first investigate application’s functionality by submitting random suggestions.

Exception flow:

1. System’s response fails to mitigate the attack; and the roles for author/editor get changed.
2. Disrupting the search functionality by a Cross Site Scripting attack

Abuse case diagram:



**Fig. 3 Abuse Case - B**

Name: Cross site Scripting attack

Actor: Crook

Trigger: Crook Injects the malicious script in the suggestion field.

Preconditions: Crook has a registered account on the application to post a suggestion.

Postcondition:

* Success postcondition: Crook fails to Inject the script into the system.
* Failure postcondition: Crook successfully injects the malicious script.

Basic Flow:

1. Crook logs into the application using the credentials.
2. He/she navigates to suggestions page.
3. He/she with intent to disrupt the search page, puts the malicious script into the input field and clicks on submit button.
4. He/she logs out of the application.

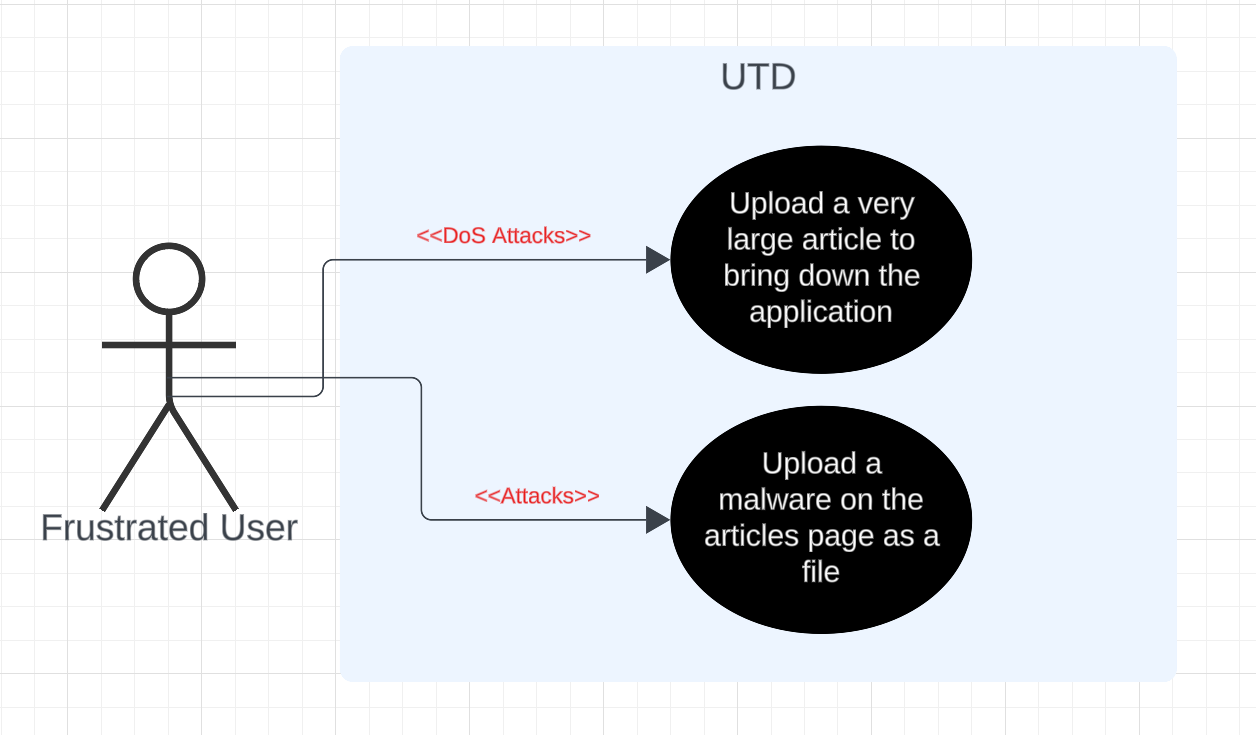
Alternative flow:

1. Crook finds other places on the application to inject the script.

Exception flow:

1. System’s response fails to mitigate the attack; and search functionality get disrupted.
2. Denial of service attack by uploading a malicious file

Abuse case diagram:



**Fig. 4 Abuse Case - C**

Name: Upload of malicious file on articles page

Actor: Frustrated User

Trigger: A Frustrated User uploads a malicious file on the articles page.

Preconditions: User has a registered account on the application to upload the file.

Postcondition:

* Success postcondition: The attack fails as the system rejects the malicious file upload.
* Failure postcondition: User succeeds to upload the malicious file as an article.

Basic Flow:

1. User logs into the application using their credentials.
2. He/she navigates to articles page.
3. With intent to disrupt the search page, he/she uploads the malicious file using choose file button on the create article interface.
4. He/she logs out of the application.

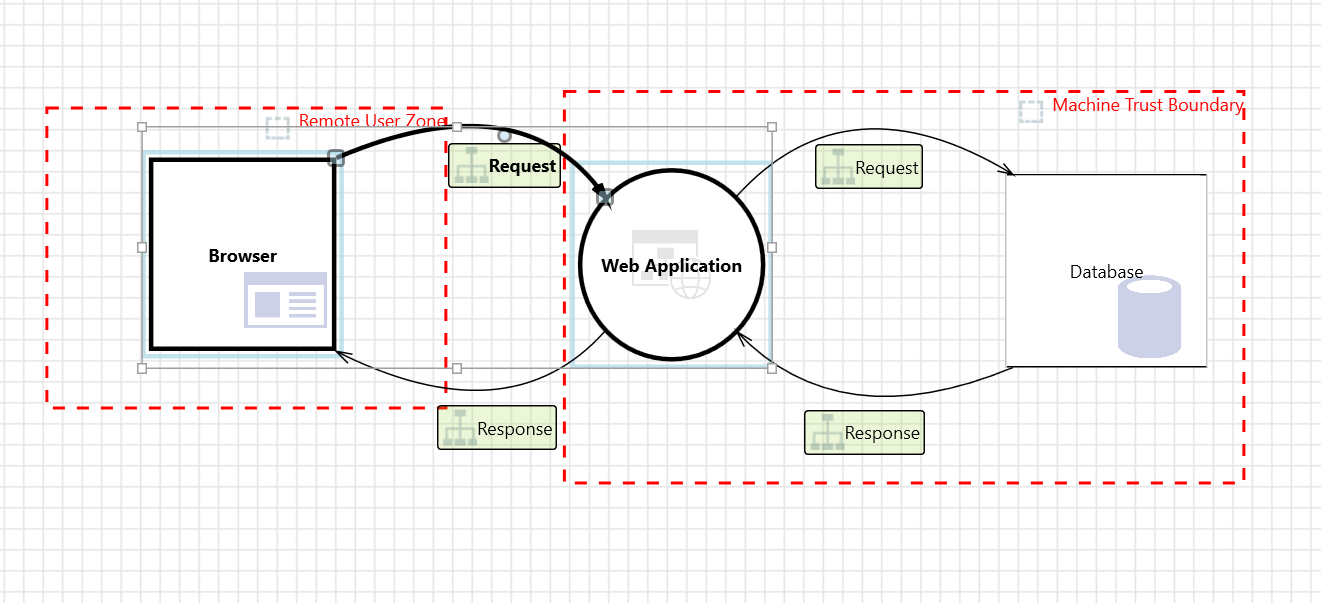
Alternative flow:

1. User tries to upload different file first to test the functioning.

Exception flow:

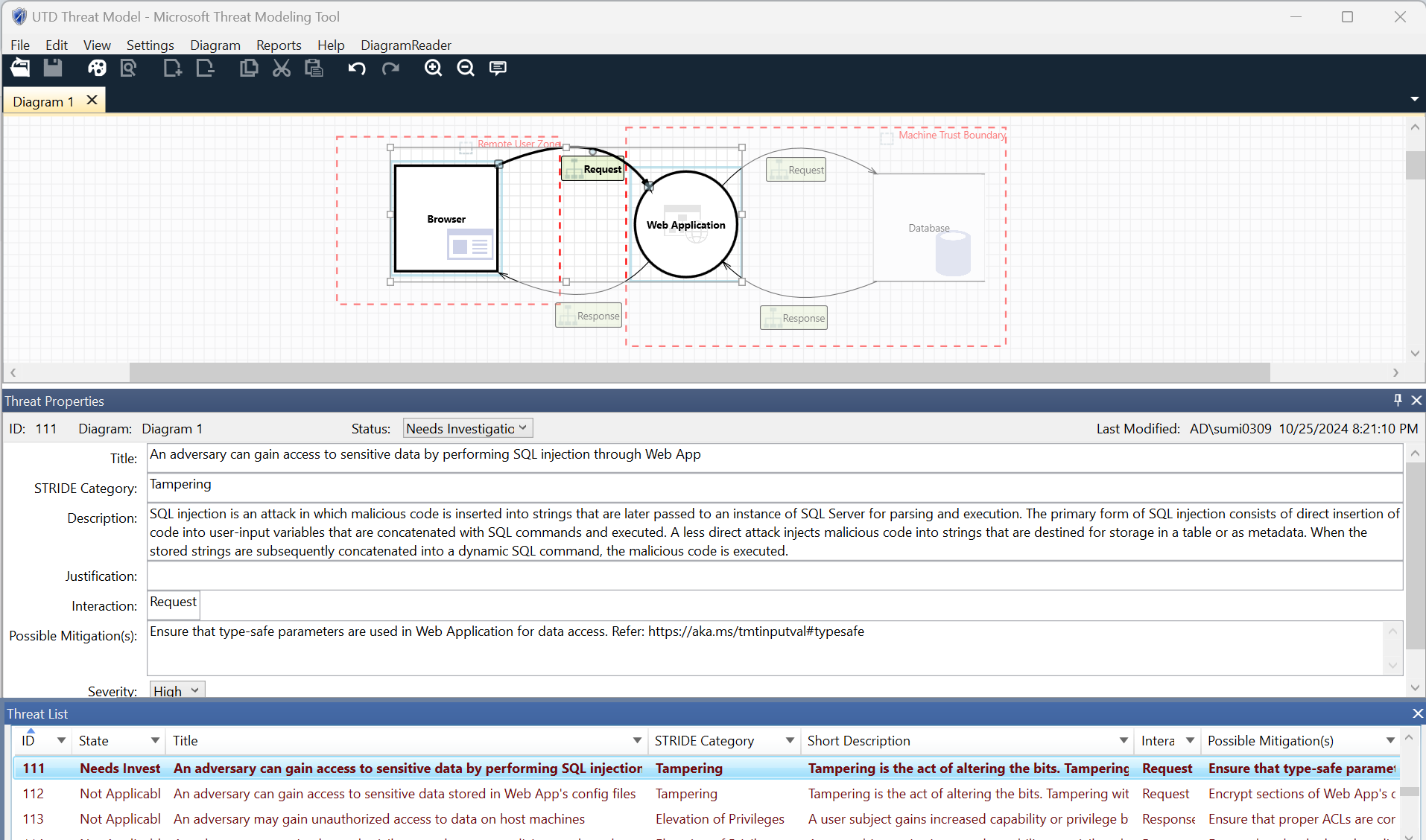
1. System’s response fails to mitigate the attack; and the application experiences DoS attack/Malware gets into the database.

The designed threat model using the Threat Modeling Tool:



**Fig. 5 Threat Model**

Some of the threats identified by the tool:



**Fig. 6 Threat Analysis Report**

The full report file of the Threat Model can be found at the following link:

<https://drive.google.com/file/d/1nbu6VCX9_5N6eptZ8nG5-SeQlWxEbfSO/view?usp=sharing>

*2.3 Functional and Non-Functional Requirements*

2.3.1 Functional Requirements

The following functional requirements have been derived from the Author, Reader, and Contributor use cases in the SRS:

**Author Use Case Requirements:**

* FR1: The system must allow an Author to log in and access the "Create News" interface.
  + **Precondition**: The Author must be registered and have valid login credentials.
  + **Success** **Flow**: The Author enters correct credentials and is redirected to the News Feed with access to the "Create News" interface.
  + **Alternative** **Flow**: If the Author is not registered, they must register before accessing the system.
  + Postcondition: The Author is successfully logged in and can create news articles.
  + **Error** **Flow**: If invalid credentials are entered, an error message is displayed, and the login fails.
* FR2: The system must enable an Author to add a news article with a headline, detailed description, time, and location.
  + **Precondition**: The Author must be logged in and must have accessed the "Create News" interface.
  + **Success** **Flow**: The Author fills out all required fields (headline, description, time, location) and clicks "Publish."
  + **Postcondition**: The news article is added to the system and available for others to see.
  + **Error** **Flow**: If there is a system error while saving the news article, the operation is aborted, and an error message is shown.
  + **Trigger**: Create News button on the News Feed page.
* FR3: The system must allow the Author to submit the news article for publication, making it visible to all users on the news feed.
  + **Trigger**: The Publish button on the Add News page.
  + **Precondition**: The Author has successfully added a news article and is ready to submit.
  + **Success** **Flow**: The Author clicks the "Publish" button, and the news article is visible to Readers and Contributors on the feed.
  + **Postcondition**: The news article is live and accessible to all users.
  + **Error** **Flow**: If the system fails to publish the article, an error message is displayed, and the article remains unpublished.

**Reader Use Case Requirements:**

* FR4: The system must allow a Reader to log in and access the news feed.
  + **Trigger**: The Login Button on the Login page.
  + **Precondition**: The Reader must be registered and have valid login credentials.
  + **Success** **Flow**: The Reader enters valid credentials and is redirected to the News Feed.
  + **Alternative** **Flow**: If the Reader is not registered, they must sign up before logging in.
  + **Postcondition**: The Reader is successfully logged in and can browse the news feed.
  + **Error** **Flow**: If the login fails due to incorrect credentials, the system displays an error message, and login is prevented.
* FR5: The system must enable a Reader to browse news articles and view details like the headline, description, time, and location.
  + **Precondition**: The Reader must be logged in.
  + **Success** **Flow**: The Reader selects a news article from the feed and views the full details.
  + **Alternative** **Flow**: If there are no articles available, the system does not display anything.
  + **Postcondition**: The Reader successfully views the article details.
  + **Error** **Flow**: If the system fails to retrieve the article details, an error message is displayed.
* FR6: The system must provide a search functionality for Readers to search for news articles using keywords.
  + **Trigger**: The search button on the search page.
  + **Precondition**: The Reader must be logged in and have access to the search page.
  + **Success** **Flow**: The Reader enters a keyword, and the system displays relevant news articles.
  + **Alternative** **Flow**: If no matching articles are found, nothing is displayed.
  + **Postcondition**: The search results are displayed to the Reader.
  + **Error** **Flow**: If the search function fails, an error message is shown.

**Contributor Use Case Requirements:**

* FR7: The system must allow a Contributor to log in and access the news feed.
  + **Trigger**: Login button on the login page.
  + **Precondition**: The Contributor must be registered and have valid login credentials.
  + **Success** **Flow**: The Contributor logs in and is redirected to the news feed.
  + **Alternative** **Flow**: If the Contributor is not registered, they must register before accessing the system.
  + **Postcondition**: The Contributor is successfully logged in and can access articles to edit.
  + **Error** **Flow**: If invalid credentials are entered, an error message is displayed, and the login fails.
* FR8: The system must enable a Contributor to select and edit existing news articles.
  + **Trigger**: Edit button on the News Feed page.
  + **Precondition**: The Contributor must be logged in and have access to the news feed.
  + **Success** **Flow**: The Contributor selects a news article and makes the necessary edits.
  + **Postcondition**: The edited article is published and visible on the News Feed.
  + **Error** **Flow**: If the edit operation fails, an error message is shown.
* FR9: The system must allow a Contributor to submit updates or changes to an article for review.
  + **Trigger**: Publish button on the Add News page.
  + **Precondition**: The Contributor must have edited a news article.
  + **Success** **Flow**: The Contributor submits the edited article, and the changes are reflected on the news feed.
  + **Postcondition**: The article is updated and visible on the feed with the changes.
  + **Error** **Flow**: If the submission fails, an error message is displayed.
* FR10: The system must allow a Reader to submit suggestions for improving an article.
* **Trigger**: Submit Suggestion button on the suggestions page.
* **Precondition**: The Reader must have accessed an article they wish to suggest changes for.
* **Success** **Flow**: The Reader submits the suggestion, and the suggestion get posted on the suggestion page.
* **Postcondition**: The suggestion is recorded in the system and made available for the Editor to review and consider for implementation.
* **Error** **Flow**: If the submission fails, an error message is displayed, prompting the Reader to try again.
* FR11: The system must allow a Reader to upload a new article to the Articles page, including a title and description.

Trigger: Upload Article button on the Articles page.

Precondition: The Reader must have the content of the article ready for submission, including a title and description.

Success Flow: The Reader uploads the article, and the article becomes available to view.

Postcondition: The uploaded article is recorded in the system and is published on the Articles page.

Error Flow: If the upload fails (e.g., due to missing fields or file size limits), an error message is displayed, prompting the Reader to correct the issues.

*2.3.2 Non-Functional Requirement*

* NFR1: The system must load the news feed within 3 seconds for up to 95% of the user requests.
* NFR2: The system should be intuitive and user-friendly, with a clear interface for Authors, Readers, and Contributors.
* NFR3: The application should maintain 99.9% availability over any 30-day period.
* NFR4: The system must be responsive, working efficiently on various screen sizes (e.g., desktop, tablet, and mobile).
  + 1. *Security requirements*

Following are the security requirements as identified after analysing the misuse cases:

* SR1: Implementing measures to prevent SQL injection attacks.
* SR2: Establishing safeguards against cross-site scripting (XSS) attacks.
* SR3: Enforcing restrictions on file uploads on the articles page to prevent malicious files and limit file size.
* SR4: Utilizing a strong hashing algorithm for secure password storage and data integrity.
* SR5: Ensuring secure connection streams to protect against data traffic sniffing.
* SR6: Providing functionality to enhance user privacy and security.
* SR7: Integrate mechanisms for security events to facilitate traceability and incident response.
* SR8: Ensuring sensitive information is not transmitted in clear text.

These security requirements account for all the misuse cases identified in the previous section.

*2.4. Logical flow of the processes*

**AUTHOR**

**WRITES**

READER

NO CHANGES

CONTRIBUTOR

SUGGESTIONS

**NEWS**

**Fig. 7 Logical Structure of System**

*2.5. User Characteristics*

1. **Authors**: Authors are users who write and post news articles. They should know how to use web applications and be good at writing clearly.
2. **Readers**: Readers are users who come to the application to read news. They might have different levels of experience with technology, but their main focus is on finding and reading news articles.
3. **Contributors**: Contributors are users who help improve the quality of news articles. They might have experience in writing, editing, or expertise in the topic of the news.

**Index**

Article, 1, 4, 5, 6, 7, 8, 9

Author, 2, 3, 4, 5, 6, 8

Contributor, 4, 6, 7, 8, 9

Database, 2

Description, 1, 3, 5, 6, 7

Edit, 4, 7

Environment, 2, 3

Information, 1, 2, 3, 4, 8

Reader, 1, 2, 3, 4, 5, 6, 7, 8, 9

Requirement, 1, 2, 4

Review, 7

Specification, 1, 2, 4

System, 1, 2, 3, 5, 6, 7, 8

Update, 4, 7

User, 1, 2, 3, 4, 6, 7, 8, 9

Web, 1, 2, 3, 8