

**Group Report**

*COMP1640 - Enterprise Web Software Development*

**Group Name :** Group 4 – TCH2404

**Member 1 :** Le Minh Vu –Greenwich ID

**Member 2 :** Tran Minh Hieu –Greenwich ID

**Member 3 :** Nguyen Van Hiep –Greenwich ID

**Member 4 :** Nguyen Van Hue –Greenwich ID

**Member 5 :** Bui Nhat Minh –Greenwich ID

**Instructor :** MSc. Pham Duc Tho

Table of Contents

[❖ **General Information** 4](#_Toc162613344)

[**1.** **Introduction** 5](#_Toc162613345)

[1.1. Introduction about the project 5](#_Toc162613346)

[1.2. Purpose of the Report 5](#_Toc162613347)

[1.3. Overview of Main Sections 6](#_Toc162613348)

[**2.** **Requirement Analysis** 6](#_Toc162613349)

[2.1. Functional and non-functional requirements 6](#_Toc162613350)

[2.2. Use-case Diagram & Specification 8](#_Toc162613351)

[2.2.1. Diagram 9](#_Toc162613352)

[2.2.2. Specification 11](#_Toc162613353)

[**3.** **Database** 24](#_Toc162613354)

[3.1. Database normalization 24](#_Toc162613355)

[3.2. Data security 24](#_Toc162613356)

[3.3. Data validation 24](#_Toc162613357)

[3.4. ERD 24](#_Toc162613358)

[3.5. Class diagram 24](#_Toc162613359)

[3.6. Database diagram 24](#_Toc162613360)

[**4.** **Design** 24](#_Toc162613361)

[**5.** **Implementation** 24](#_Toc162613362)

[**6.** **Testing** 24](#_Toc162613363)

[**7.** **Methodology** 24](#_Toc162613364)

[**8.** **Conclusion** 25](#_Toc162613365)

[❖ **Bibliography** 25](#_Toc162613366)

[Bibliography 25](#_Toc162613367)

**Table of Figures**

[Figure 1 – Student Actor 9](#_Toc162610971)

[Figure 2 - Guest Actor 9](#_Toc162610972)

[Figure 3 - Marketing Coordinator Actor 10](#_Toc162610973)

[Figure 4 - Marketing Manager Actor 10](#_Toc162610974)

[Figure 5 - Admin Actor 11](#_Toc162610975)

**List of Tables**

Table 1 - Functional and Non-functional Requirements 7

Table 2 – Guest Use-case Specification 12

Table 3 – Student Use-case Specification 13

Table 4 - Marketing Coordinator Use-case Specification No.1 14

Table 5 - Marketing Coordinator Use-case Specification No.2 15

Table 6 - Marketing Coordinator Use-case Specification No.3 16

Table 7 - Marketing Manager Use-case Specification No.1 17

Table 8 - Marketing Manager Use-case Specification No.2 18

Table 9 - Admin Use-case Specification No.1 20

Table 10 - Admin Use-case Specification No.2 21

Table 11 - Admin Use-case Specification 22

# **General Information**

|  |  |  |  |
| --- | --- | --- | --- |
| Group name | Group 4 | | |
| Repository URL | *GitHub/Gitlab link, etc.* | | |
| Group Board URL | *Not yet* | | |
| Web Hosting URL | *https://marketingeventgroup4.azurewebsites.net/swagger/* | | |
| Test accounts | *Role* | *Username* | *Password* |
| admin | admin | 123456 |
| manager | manager | 123456 |
| *…* | *…* | *…* |
| Screen cast URL | *YouTube link* | | |
| Backlog URL | *Google Docs/Sheet link* | | |
| Test plan URL | *Google Docs/Sheet link* | | |
| Front-End Technology | *JavaScript, HTML, CSS* | | |
| Back-End Technology | *.NET Core, NodeJS, C#* | | |
| Database | *SQL Server Azure* | | |

*Table 1 - Project Information*

|  |  |
| --- | --- |
| Full Name | Role |
| Le Minh Vu | Scrum Master, Business Analysis |
| Nguyen Van Hue | Front-end Developer |
| Tran Minh Hieu | Back-end Developer |
| Bui Nhat Minh | Back-End Developer |
| Nguyen Van Hiep | UI/UX Designer |

*Table 2 – Member Role*

# **Introduction**

## Introduction about the project

This report gives a thorough rundown of our web development project and describes how our team worked together to build a safe, role-based system to gather student submissions for the university magazine every year. The project's justification stems from the requirement for an organized and consistent method of collecting contributions from different university faculties while strictly adhering to privacy and data security regulations.

Our team has committed to creating an intuitive interface that is compatible with a variety of devices, with understanding being made on the distinct requirements of both educators and students. With the help of the system we've created, everyone involved with the completed project/product of our - from students to the university marketing manager - can interact with the platform in a way that streamlines their workflow and helps the editorial process.

## Purpose of the Report

Clarification of what this report aims to accomplish will be made throughout each section, which serves as an explanation on a comprehensive account of our team's joint work and the agile scrum techniques used during the project's duration. Additionally, the methods that were used by our team to guarantee the system's effective development and implementation will also be described in this paper.

## Overview of Main Sections

Here, we offer a succinct overview of each of the report's major sections. These sections will address:

* Requirement Analysis: This section describes how to collect and analyze the system's functional and non-functional needs. It explains how we arrived at our conclusions about the goals and functionality of the system.
* Database: In this section, we go over the architecture and configuration of the database that underpins the system, with a particular emphasis on how it safely saves and retrieves information relevant to user interactions and submissions.
* Design: The information architecture and user interface design that guarantee accessible and easy-to-use navigation across a range of devices are covered in this part along with the overall design approach.
* Implementation: This section explains the coding and development techniques our programmers used to convert the design into a working system.
* Testing: We examine the methods used for evaluating the system's resilience and dependability, as well as the various tests carried out and their results.
* Methodology: This section describes our team structure, the agile scrum framework we used, and the iterative cycle management we used for the project and development activity.
* Conclusion: The report's conclusion summarizes the data from the earlier sections and offers a reflection on the system's performance, our accomplishments, the difficulties we encountered, and the lessons we gained during the project.

# **Requirement Analysis**

## 2.1. Functional and non-functional requirements

Two essential components that specify what a system should accomplish and how it should perform are functional and non-functional requirements. The following should be explanations provided for each of them:

* Functional Requirements: These are the requirements outlining the different tasks that the system has to be able to complete. They specify how the system will behave while interacting with people and other systems. The common formats for capturing functional requirements include scenarios, use cases, and user stories.
* Non-functional requirements address the manner in which the system carries out its operations, as opposed to functional requirements, which concentrate on what the system accomplishes. They provide an overview of the system's functionalities and quality features, including usability, scalability, security, performance, and maintainability. The effectiveness of the system as a whole and the user experience in general depend on these needs.

It is important to consider what are the differences between these 2 types of requirements, the following table, which is derived from an article published on GeeksforGeeks (2024):

|  |  |
| --- | --- |
| Functional Requirements | Non-functional Requirements |
| A system or one of its components is defined by a functional requirement. | A software system's quality attribute is specified by a non-functional requirement. |
| "What should the software system do?" is clarified. | "How should the software system fulfill the functional requirements?" is constrained by it. |
| The user specifies the functional requirements. | Technical professionals, such as architects, technical leaders, and software developers, specify non-functional requirements. |
| It is mandatory. | It is not mandatory. |
| It's included in the use case. | It is recorded as a characteristic of quality |
| Specified on a component basis. | Applied to the entire system. |
| Assists you in confirming the software's functionality. | aids in the verification of the software's performance. |
| Functional testing is carried out, including system, integration, end-to-end, and API testing, etc. | Testing for non-functional aspects such as performance, stress, usability, security, etc., is carried out. |
| Usually simple to describe. | Typically harder to describe. |

Table 1 - Functional and Non-functional Requirements

Based on the previously mentioned project scenario, the following functional and non-functional are needed in the context of the university magazine web-based system:

* Functional Requirements:
* Users should be able to create accounts with roles such as student, Marketing Coordinator, University Marketing Manager, and Administrator.
* Students must be able to submit articles and high-quality images.
* Submissions must be automatically closed after a specified closure date.
* There should be a Terms and Conditions acceptance feature before article submission.
* The system must send email notifications to the respective Faculty’s Marketing Coordinator upon student submission.
* Marketing Coordinators can only access and edit contributions from students in their faculty.
* The University Marketing Manager should be able to view and download selected contributions in a ZIP file.
* An Administrator should manage system data, like setting the closure dates.
* The system must generate reports and statistical analyses as required.
* Non-functional Requirements:
* The system should provide a high level of security to ensure that all user data and submitted content are protected.
* It should be reliable, maintaining uptime and data integrity.
* The system needs to perform efficiently, handling the expected load of student submissions without lag.
* It requires availability on different devices and screen sizes, with responsive design for mobile phones, tablets, and desktop computers.
* Maintainability is important for future updates and improvements to the system. The interface should be intuitive and user-friendly.

## 2.2. Use-case Diagram & Specification

As mentioned in an article written by Kate Brush (2022), a use case is a way for organizing, defining, and identifying system needs in system analysis. A use case is a collection of potential interactions between people and systems in a certain setting that are all aimed at achieving a specific objective. The process generates a document that lists every action a user takes to finish a task.

Business analysts are engaged at several stages of software development, including establishing system requirements, validating design, testing software, and generating an overview for online help and user guides. They are generally in charge of writing use cases. The development team may find and comprehend potential error locations during a transaction with the use of a use case document, allowing for issue resolution.

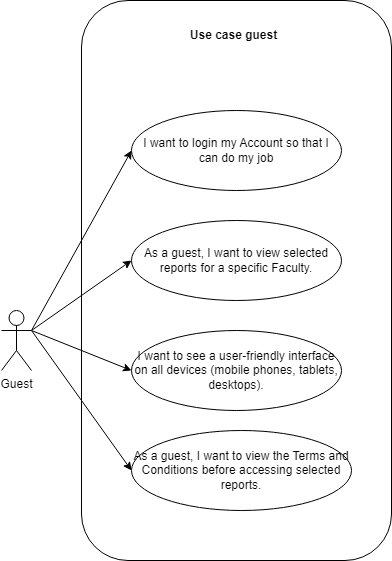
Three components are present in every use case:

* The actor: The individual or group of persons utilizing the system to interact with the process is known as the system user.
* The goal: the successful conclusion that brings the process to a close.
* The system: The procedures followed in order to accomplish the final result, together with the functional requirements that must be met and the expected behaviors of those requirements.

That said, the following sections shall provide more information on each type of the use case, namely diagram and specification, with the former describes briefly on what can each actor do, and the former for demonstrating further into their specifications.

### 2.2.1. Diagram

This section is self-explanatory in which each actor of the system will be depicted on what can they interact with the system (based on the initial requirements). Therefore, please refer to the following use-case diagrams for more details:



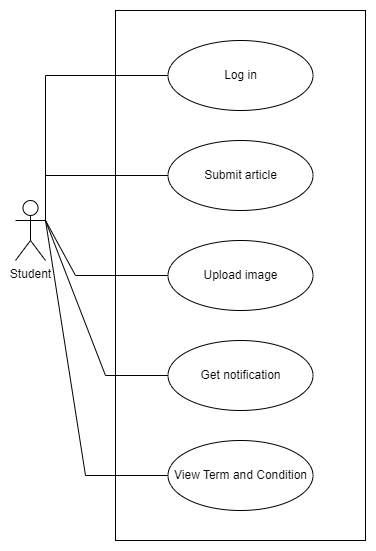


Figure 1 – Student Actor

Figure 2 - Guest Actor

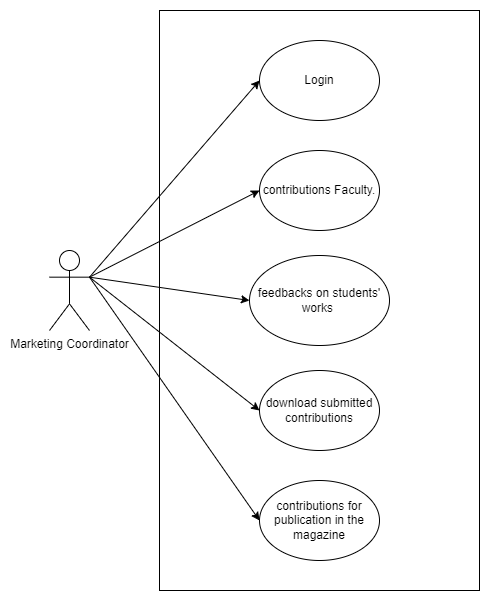
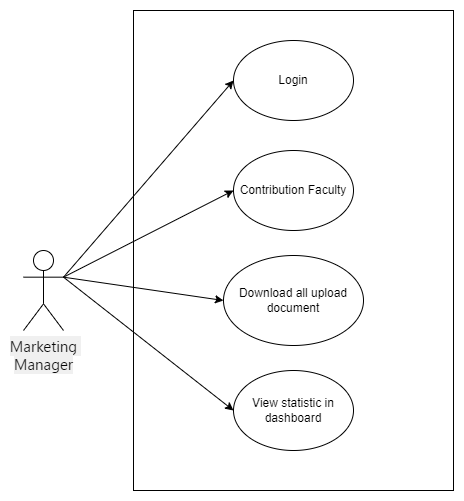


Figure 3 - Marketing Coordinator Actor

Figure 4 - Marketing Manager Actor

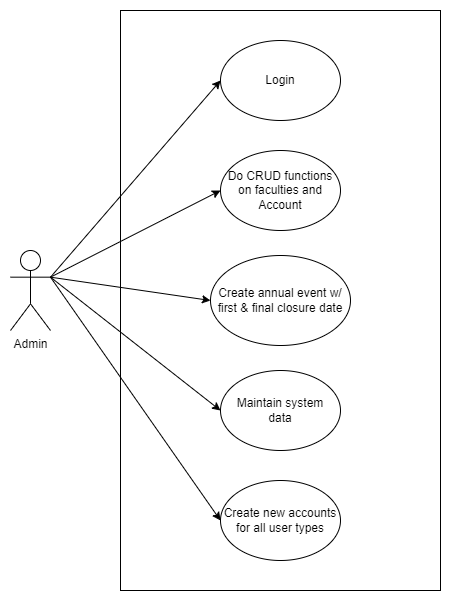


Figure 5 - Admin Actor

### 2.2.2. Specification

For specification, we will dive further into the details for each of the action that each type of user can interact with the system, this section will be further divided into 5 different parts for 5 different roles. Any of the specification that has overlap requirements, such as login, will be seperated into the 6th part, which is the common specification for all roles.

#### 2.2.2.1. Guest

|  |  |
| --- | --- |
| Use-Case Name | View Terms and Conditions |
| Description | This use-case describes the process by which a guest views the Terms and Conditions on the system before being granted access to view selected reports for a specific Faculty. |
| Actors | Guest |
| Pre-conditions | * The guest must be logged into their account with guest access privileges. * Terms and Conditions must be up-to-date and available in the system. * Selected reports for viewing are already determined by the system based on guest privileges. |
| Basic Flow | * A guest logs in to the system and navigates to the section where selected reports can be accessed. * The system detects that the Terms and Conditions have not been agreed to by the user during this session. * The system prompts the guest with a message to review the Terms and Conditions before proceeding. * The guest clicks on the provided link or button to view the Terms and Conditions. The system displays the complete Terms and Conditions document. * The guest reviews the document and indicates their agreement by checking a checkbox or clicking a 'Agree' button. * The system records the guest's agreement and permits access to the selected reports. * The guest is then able to view the selected reports for a specific Faculty. |
| Alternative Flows | 1. If the guest does not agree to the Terms and Conditions:   1. The system does not grant access to the selected reports.  2. The guest is informed that they cannot proceed without accepting the Terms and Conditions. |
| Post-conditions | * The system has recorded the guest's agreement to the Terms and Conditions. * The guest is granted access to the selected reports. |
| Exceptions | If the Terms and Conditions cannot be displayed due to a system error, the system will inform the user of the issue and suggest trying again later. |

Table 2 – Guest Use-case Specification

#### 2.2.2.2. Student

|  |  |
| --- | --- |
| Use-Case Name | Submit Article, upload image, review Terms & Condtions and receive notifications |
| Description | This use-case outlines the steps a student follows to submit an article and accompanying high-quality images for inclusion in the university magazine as well as a notification. |
| Actors | Student |
| Pre-conditions | * The student is registered and logged into their account. * The system is before the initial closure date for new submissions. |
| Basic Flow | * The student navigates to the "Submit Article" area within the system. * The system presents a submission form requiring details of the article and an option to upload files. * The student fills in the fields for the title and abstract of the article. * The student uploads the main article document and selects any related high-quality images for submission. * The system prompts the student to review the Terms and Conditions. * The student reads and accepts the Terms and Conditions by clicking the acceptance button or checking a checkbox. * The system checks the file types and sizes against the permitted formats and limits. * The student submits the article and images by clicking the "Submit" button. * The system validates and confirms the submission, recording it in the database. * The student receives a pop-up notification confirming the successful submission of their work. |
| Alternative Flows | 1. If the student declines the Terms and Conditions:   1. The system prevents the submission from being completed.  2. The student is informed that acceptance of the Terms and Conditions is required to proceed with the submission. |
| Post-conditions | * The student’s article and images are successfully submitted to the system. * The Marketing Coordinator is notified of the new submission for their Faculty. |
| Exceptions | In case of a network or system failure during submission:   * The system alerts the student to the failure. * The student is advised to save their work and attempt to resubmit at a later time. |

Table 3 – Student Use-case Specification

#### 2.2.2.3. Marketing Coordinator

|  |  |
| --- | --- |
| Use-Case Name | Review and Provide Feedback on Contributions |
| Description | This use-case outlines the process by which a Marketing Coordinator reviews student contributions from within their faculty and provides feedback. |
| Actors | Marketing Coordinator (Primary Actor) & Student (Secondary Actor – Receives feedback) |
| Pre-conditions | * The Marketing Coordinator must be logged into the system with the appropriate access rights. * Submissions from students within the Marketing Coordinator's faculty must be present for review. * The system must allow for feedback to be submitted on each individual student contribution. |
| Basic Flow | * The Marketing Coordinator logs into the system and navigates to the "Contributions" section. * The system displays a list of submissions from students within the Marketing Coordinator's faculty. * The Marketing Coordinator selects an individual contribution to review. * The system retrieves and displays the selected article and any associated images. * The Marketing Coordinator reads the article and reviews the images, if any. * The Marketing Coordinator fills in the feedback form provided by the system, detailing suggestions for edits or approval. * The system saves the feedback and marks the reviewed submission with the coordinator's notes. * An automatic notification is sent to the student regarding the new feedback on their contribution. |
| Alternative Flows | If no up-to-date contributions are present for review: The system informs the Marketing Coordinator there are no current submissions to review. |
| Post-conditions | * Each reviewed student contribution has feedback associated with it in the system. * Students can view the feedback and make necessary revisions to their contributions. |
| Exceptions | In the event of a system error preventing feedback submission:   * The Marketing Coordinator is prompted with an error message. * The system logs the error and advises the coordinator to try again later. |

Table 4 - Marketing Coordinator Use-case Specification No.1

|  |  |
| --- | --- |
| Use-Case Name | Download Submitted Contributions |
| Description | This use-case ensures the Marketing Coordinator can download submitted student contributions within their faculty for offline review and consideration for publication. |
| Actors | Marketing Coordinator |
| Pre-conditions | * The Marketing Coordinator is logged into the system with appropriate credentials. * There are submitted student contributions available for download. |
| Basic Flow | * The Marketing Coordinator accesses the system and navigates to the relevant section containing student contributions. * The system displays the list of available submissions from students within the Marketing Coordinator's faculty. * The Marketing Coordinator selects either individual contributions or opts to bulk download multiple submissions. * The system prepares the selected files, which may include articles and images, for download. * The Marketing Coordinator initiates the download process. * The system compresses the files into a downloadable format, such as a ZIP file, and provides a download link. * The Marketing Coordinator clicks the link and the download begins. * Once downloaded, the Marketing Coordinator is able to review the contributions offline. |
| Alternative Flows | If the selected contributions are unable to be compressed or downloaded:   1. The system displays an error message explaining the issue. 2. The Marketing Coordinator is advised to attempt the download again or contact technical support if the issue persists. |
| Post-conditions | If network issues disrupt the download process, the system advises the Marketing Coordinator to check their internet connection and try again. |
| Exceptions | In the event of a system error preventing feedback submission:   * The Marketing Coordinator is prompted with an error message. * The system logs the error and advises the coordinator to try again later. |

Table 5 - Marketing Coordinator Use-case Specification No.2

|  |  |
| --- | --- |
| Use-Case Name | Select Contributions for Publication |
| Description | This use-case describes the process by which the Marketing Coordinator evaluates submitted contributions and selects certain pieces for inclusion in the university magazine. |
| Actors | Marketing Coordinator |
| Pre-conditions | * The Marketing Coordinator has successfully logged into the system. * Student contributions are available for review and selection. * The Marketing Coordinator has access rights to approve contributions for publication. |
| Basic Flow | * The Marketing Coordinator logs into the system and navigates to the submission review section. * The system presents a list of submitted contributions from students within the Marketing Coordinator's faculty. * The Marketing Coordinator reviews the submissions and decides which contributions are suitable for publication. * For each selected contribution, the Marketing Coordinator marks it as ‘selected for publication’ using the system interface. * The system updates the status of each marked contribution to reflect its selection for publication. * The list of chosen articles is compiled and stored within the system for final review and confirmation. * The Marketing Coordinator confirms the final selection. * The system sends notifications to the respective students informing them of their contribution’s selection for publication. |
| Alternative Flows | If the Marketing Coordinator changes their mind about a selected piece:   1. The Marketing Coordinator can deselect the contribution before finalizing their choices. 2. The system updates accordingly and removes the article from the publication list. |
| Post-conditions | * Selected contributions are marked and confirmed for publication. * Students are notified of their contributions being selected. |
| Exceptions | If there is a system failure or error during the selection process:   * The system records the error and notifies the Marketing Coordinator. * The system recommends that the Marketing Coordinator attempts the process again once the issue is resolved. |

Table 6 - Marketing Coordinator Use-case Specification No.3

#### 2.2.2.4. Marketing Manager

|  |  |
| --- | --- |
| Use-Case Name | View and Download All Faculty Contributions |
| Description | This use-case allows the Marketing Manager to view all student contributions across different Faculties and download all contributions for final review or archiving purposes. |
| Actors | Marketing Manager |
| Pre-conditions | * The Marketing Manager must be logged into the system with high-level access rights. * There are contributions from various Faculties that have been submitted and are available for viewing and download. |
| Basic Flow | * The Marketing Manager logs into the system and navigates to the section dedicated to viewing all Faculty contributions. * The system displays a comprehensive list or dashboard of student contributions grouped by Faculty. * The Marketing Manager reviews the contributions from different Faculties as needed. * To download contributions, the Marketing Manager selects the option to "Download All" or selects specific contributions for download. * The system compiles the selected articles and images into a downloadable file format, such as ZIP. * The system provides a download link to the Marketing Manager. * The Marketing Manager clicks the download link, and the downloading process begins. * After the download is complete, the Marketing Manager has offline access to all the contributions selected. |
| Alternative Flows | If the Marketing Manager wants to download contributions from specific Faculties only:   1. The Marketing Manager selects the particular Faculty or Faculties. 2. The Marketing Manager initiates the download process for the selected subset. |
| Post-conditions | * The Marketing Manager can view all the contributions from the entire university. * The Marketing Manager successfully downloads the required files for offline access and review. |
| Exceptions | If there is a system error or failure during the download:   * The system generates an error message and logs the incident. * The Marketing Manager is prompted to try the download again or to seek assistance from technical support. |

Table 7 - Marketing Manager Use-case Specification No.1

|  |  |
| --- | --- |
| Use-Case Name | Access Statistics Dashboard |
| Description | This use-case enables the Marketing Manager to access and view various statistical data regarding student contributions through a dashboard interface. |
| Actors | Marketing Manager |
| Pre-conditions | * The Marketing Manager is logged into the system with appropriate access rights. * Statistical data related to student contributions is collected and available in the system. * The dashboard functionality is operational and up to date. |
| Basic Flow | * The Marketing Manager logs into the system with their credentials. * The Marketing Manager navigates to the dashboard feature within the system. * The system displays the dashboard with various statistical data and visualizations. * The Marketing Manager selects specific data points or criteria (e.g., academic year, a particular Faculty) to tailor the dashboard view. * The system updates the dashboard to reflect the Marketing Manager's selections, showing relevant statistics such as the number of submissions, percentage of contributions by Faculty, number of approved contributions, and any other pertinent metrics. * The Marketing Manager reviews the displayed statistics to gain insights into the contributions and trends across Faculties. |
| Alternative Flows | If the dashboard fails to load or display updated statistics:   1. The system shows an error message detailing the issue. 2. The Marketing Manager is advised to refresh the dashboard or contact technical support if the problem persists. |
| Post-conditions | * The Marketing Manager successfully accesses and views the desired statistical data. * The Marketing Manager has a clearer understanding of the contribution landscape across the university. |
| Exceptions | In the event of connectivity issues or system errors that prevent dashboard access:   * The system informs the Marketing Manager of the connectivity or technical error. * The Marketing Manager is directed to retry accessing the dashboard or report the issue to the IT department. |

Table 8 - Marketing Manager Use-case Specification No.2

#### 2.2.2.5. Admin

|  |  |
| --- | --- |
| Use-Case Name | Manage Faculties (CRUD Operations) |
| Description | This use-case outlines the administrative functions that enable the system's Admin to create, read, update, and delete faculty records within the system. |
| Actors | Administrator |
| Pre-conditions | * The Administrator is logged into the system with full administrative access rights. * The Admin has access to the section of the system where faculty data is managed. |
| Basic Flow | * The Admin logs into the system and navigates to the faculty management interface. * For creating a new faculty record. * The Admin selects the option to create a new faculty. * The Admin fills in the necessary details for the new faculty record and saves the information. * The system validates the information and creates a new faculty entry. * For reading (viewing) faculty records: * The Admin browses through a list of existing faculties. * The Admin selects a faculty to view more detailed information. * The system displays the complete record for the selected faculty. * For updating an existing faculty record: * The Admin selects an existing faculty and chooses to edit its details. * The Admin makes the required changes in the faculty information form. * The system validates and updates the faculty record with the new information. * For deleting a faculty record: * The Admin selects a faculty and initiates the delete action. * The system asks the Admin to confirm the deletion to avoid accidental loss of data. * Upon confirmation, the system permanently removes the faculty record. |
| Alternative Flows | 1. If the Admin attempts to create a duplicate faculty record: 2. The system displays a warning message indicating that the faculty already exists. 3. The Admin is prompted to either update the existing record or cancel the creation process. 4. If the Admin inputs invalid data during the create or update actions: 5. The system rejects the submission and highlights the errors to the Admin. 6. The Admin corrects the mistakes and resubmits the form. |
| Post-conditions | * The Admin has successfully performed the required CRUD operations on faculty records. * The system reflects the changes accurately and maintains data integrity. |
| Exceptions | In case of system malfunction or failure during any CRUD operation:   * The system records the error and notifies the Admin. * The Admin is prompted to retry the operation or contact IT support if the issue is not resolved. |

Table 9 - Admin Use-case Specification No.1

|  |  |
| --- | --- |
| Use-Case Name | Manage Accounts (CRUD Operations) |
| Description | This use-case details the process for the Administrator to perform create, read, update, and delete operations on user accounts within the system. |
| Actors | Administrator |
| Pre-conditions | * The Administrator must be logged in with the necessary permissions to manage user accounts. * The Admin interface for account management is operational. |
| Basic Flow | * The Administrator logs into the system and accesses the account management section. * For creating a new account: * The Admin selects the option to create a new user account. * The Admin fills out the account creation form with the user's details and role. * The system validates the new account information and adds the account to the system. * For reading (viewing) account details: * The Admin selects an option to view existing user accounts. * The system displays a list of accounts. * The Admin can select any account to view its full details. * For updating an existing account: * The Admin searches for and selects the user account that needs editing. * The Admin makes changes to the account information. * The system validates the changes and updates the account details. * For deleting an account: * The Admin selects the user account to be deleted. * The system requests confirmation to ensure the action is not accidental. * Upon confirmation, the system deletes the user account. |
| Alternative Flows | 1. In the event an Admin tries to create an account that already exists: 2. The system flags the duplication and prevents the creation of the account. 3. The Admin can opt to update the existing account or create a different account. 4. If an Admin enters invalid details while creating or updating an account: 5. The system displays an error message detailing the validation failure. 6. The Admin corrects the errors and retries the operation. |
| Post-conditions | * The Administrator successfully executed the required CRUD operations. * The system's user accounts reflect accurate, up-to-date information. |
| Exceptions | If a technical issue occurs during any CRUD operation:   * The system logs the error and alerts the Administrator. * The Admin is advised to retry the action, and if the issue persists, to escalate the problem to the technical support team. |

Table 10 - Admin Use-case Specification No.2

|  |  |
| --- | --- |
| Use-Case Name | Maintain System Data |
| Description | This use-case outlines the administrative responsibility for maintaining, updating, and ensuring the integrity of the system data, including settings, configurations, and key information related to the operation of the university magazine system. |
| Actors | Administrator |
| Pre-conditions | * The Administrator is logged into the system with administrative privileges that grant access to system data maintenance functionalities. * The data that needs to be maintained is identified and accessible within the admin interface. |
| Basic Flow | * The Administrator logs into the system and navigates to a centralized dashboard or section dedicated to system data maintenance. * The system presents the Administrator with a range of system data categories, such as closure dates, terms and conditions, account role permissions, and other configurable settings. * To update or modify system data: * The Admin selects the category of system data to be updated. * The Admin makes the necessary changes, such as updating closure dates or editing terms and conditions content. * The system validates the changes and, upon approval, saves the updated information. * To verify system data integrity: * The Admin reviews logs, data usage reports, and performs system health checks. * The system provides real-time diagnostics and highlights any discrepancies or areas that require attention. * The Admin addresses any highlighted issues and ensures the system data integrity is maintained. * The Administrator can perform periodic data clean-up: * The Admin assesses data for relevance, accuracy, and redundancy. * The system facilitates the secure removal or archiving of outdated or unnecessary data as directed by the Admin. * The system completes the data clean-up process and updates the information logs accordingly. |
| Alternative Flows | If the system data cannot be updated due to restricted access or validation failure:   1. The system displays an error message specifying the reason for the failure. 2. The Administrator makes the necessary adjustments or seeks higher-level authorization and retries the update.. |
| Post-conditions | * The system data has been successfully maintained, updated, or cleaned up by the Administrator. * The system is up-to-date with accurate settings and configurations that reflect the operational needs of the university magazine system. |
| Exceptions | In the event of a system malfunction or error during maintenance:   * The system records the error and notifies the Administrator. * The Administrator investigates the cause of the error and can choose to resolve it directly or escalate it to the technical support team. |

Table 11 - Admin Use-case Specification

#### 2.2.2.6. Common Use-case Specification

|  |  |
| --- | --- |
| Use-Case Name | User Login |
| Description | This use-case describes the process by which users of different types (guest, student, Marketing Coordinator, Marketing Manager, administrator) access their accounts on the university magazine system. |
| Actors | User - This actor represents any user type with the intent to log into the system. |
| Pre-conditions | * The user possesses a valid account with username and password credentials. * The system's login functionality is operational and accessible. |
| Basic Flow | * The user navigates to the login page of the university magazine system. * The system presents fields for entering a username and password. * The user enters their username and password into the respective fields. * The user submits the login credentials by clicking the 'Login' button or hitting 'Enter'. * The system validates the provided credentials against stored user data. * Upon successful authentication, the system grants access to the user interface based on user type and associated permissions. * The user is redirected to their respective dashboard or home page within the system. |
| Alternative Flows | 1. If the user enters incorrect login credentials: 2. The system displays a message indicating that the login attempt was unsuccessful. 3. The user can attempt to login again or use a password recovery feature if available. 4. If the user has forgotten their password: 5. The user selects a "Forgot Password?" link or button 6. The system guides the user through a secure password reset process. |
| Post-conditions | The user is successfully logged in and has access to the system functionalities applicable to their role. |
| Exceptions | In case of a system error preventing the user from logging in:   * The user is presented with a general error message and the option to try again. * If the issue persists, the system may suggest contacting technical support for assistance. |

Table 12 - Common Use-case Specification

# **Database**

As stated and described in an article posted by Oracle (Oracle, n.d. ), an electronic collection of structured data, usually kept in an ordered manner within a computer system, is called a database, there is also a database management system (DBMS) that typically has control over a database. The term "database system" in this case, which is frequently abbreviated as "database," refers to the combination of the data, the DBMS, and the related applications. To facilitate efficient processing and data querying, data in the most widely used database formats now in use is usually organized into a sequence of tables with rows and columns. After then, it will be simple to manage, control, update, modify, and arrange the data. Structured query language, or SQL, is what most databases use to write and query data. For this specific project, SQL is utilized for accurate data normalzation and organization, additionally, stable online deployment of database is made possible thanks to Azure subscription.

## 3.1. Database diagram

## 3.2. Database normalization

Database normalization is a database design principle for organizing data in an organized and consistent way in which it helps avoiding redundancy and maintain the integrity of the database. Additionally, it assists the user to eliminate undesirable characteristics associated with insertion, deletion, and updating (Chris, 2022). Database administrators can use composite, foreign, and primary keys to make the data separated into multiple tables and connected by relationships throughout the normalization process to accomplish these relationships. Normalization also has a unique trait of different types, the first three types of database normalization are 1NF, 2NF, and 3NF, in which first, second, and third normal forms are what they stand for, respectively. Every kind of database normalization is cumulative, meaning that each one builds upon the types that come before it. Thus, every concept in 1NF is also applicable in 2NF, and so forth. The following sub-section will demonstrate more into how each type of normaliztion is achieved in this database

### 3.2.1. 1NF

3.2.2. 2NF

3.2.3. 3NF

## 3.3. Data security

## 3.4. Data validation

## 3.5. ERD

## 3.6. Class diagram

database normalization, data security, data validation, ERD, class diagram, database diagram

# **Design**

sitemap, wireframes,  flowcharts, mockup (Figma), activity diagram, sequence diagrams

# **Implementation**

overview about selected techniques (FE+BE+DB), system structures , features introduction & explanation for outstanding/important code snippets, final web screenshots

# **Testing**

Test overview, Test plan, test scenarios, test cases, test logs, test evidences, test result evaluation

# **Methodology**

user stories, product backlog, sprint backlogs, burndown chart, meeting minutes, team challenges

# **Conclusion**

summarize have been presented in this report/what have done in this project, overall evaluation for web application/Agile methodology, future improvements

# **Bibliography**

# Bibliography

Brush, K. (2022, November n.d. ). *Definition of Use Case*. Retrieved from TechTarget: https://www.techtarget.com/searchsoftwarequality/definition/use-case#:~:text=A%20system%20use%20case%20diagram,the%20boundaries%20of%20the%20processes.

Chris, K. (2022, December 21). *Database Normalization – Normal Forms 1nf 2nf 3nf Table Examples*. Retrieved from freeCodeCamp: https://www.freecodecamp.org/news/database-normalization-1nf-2nf-3nf-table-examples/

GeeksForGeeks. (2024, Jan 17). *Functional vs Non Functional Requirements*. Retrieved from GeeksForGeeks: https://www.geeksforgeeks.org/functional-vs-non-functional-requirements/

Oracle. (n.d. , n.d. n.d.). *What is a Database?* Retrieved from Oracle: https://www.oracle.com/database/what-is-database/