

**UNIVERSITY OF GREENWICH**  
COMP1640 – Enterprise Web Software Development

Group Coursework

|  |  |
| --- | --- |
| Student name | Luong Gia Luan |
| ID number (00xxxxxxx) | 001353344 |
| Lecturer/Tutor name | Doan Dinh Ho |
| Student submission date |  |

**Team Name**: Group 5

|  |  |  |
| --- | --- | --- |
| **Member** | **ID** | **Role** |
| Luong Gia Luan | 001353344 | Scrum Master, Product Owner, & Programmer |
| Nguyen Nhat Chinh | 001353253 | Database Designer & Programmer |
| Luu Pham Anh Kiet | 001353326 | Tester & Programmer |
| Huynh Quoc Tuan | 001353655 | Tester & Web Designer |
| Truong Quan Minh | 001353358 | Information Architect & Programmer |
| Nguyen Vu Anh | 001353224 | Web Designer & Programmer |

Table of Contents

[**1.** **Introduction** 5](#_Toc162306916)

[**2.** **Database** 5](#_Toc162306917)

[**2.1.** **Security** 6](#_Toc162306918)

[**2.2.** **Appropriate Data Types and Validation** 6](#_Toc162306919)

[**2.3.** **ERD** 7](#_Toc162306920)

[**2.4.** **Referential Integrity** 9](#_Toc162306921)

[**2.5.** **Role Implementation** 9](#_Toc162306922)

[**3.** **Site Design** 10](#_Toc162306923)

[**3.1.** **Responsive Design** 11](#_Toc162306924)

[**3.2.** **Information Architecture** 11](#_Toc162306925)

[**3.3.** **Usability Heuristics** 11](#_Toc162306926)

[**3.4.** **Usability** 12](#_Toc162306927)

[**3.5.** **Accessibility** 12](#_Toc162306928)

[**4.** **Functionality** 14](#_Toc162306929)

[**4.1.** **Role Based Security** 14](#_Toc162306930)

[**4.2.** **Submission of Report** 14](#_Toc162306931)

[**4.3.** **Email Notification** 14](#_Toc162306932)

[**4.4.** **Summary and Exception Reports** 14](#_Toc162306933)

[**4.5.** **UML diagram** 15](#_Toc162306934)

[**5.** **Testing** 15](#_Toc162306935)

[**5.1.** **Test Plan** 15](#_Toc162306936)

[**5.1.1.** **Test Case Sprint 1** 1](#_Toc162306937)

[**5.1.2.** **Test Case Sprint 2** 1](#_Toc162306938)

[**5.1.3.** **Test Case Sprint 3** 2](#_Toc162306939)

[**5.1.4.** **Test Case Sprint 4** 2](#_Toc162306940)

[**5.1.5.** **Test Case Sprint 5** 2](#_Toc162306941)

[**5.1.6.** **Test Case Sprint 6** 2](#_Toc162306942)

[**5.2.** **Test Log** 2](#_Toc162306943)

[**5.3.** **Sufficient Data to Fully Test** 2](#_Toc162306944)

[**6.** **Agile Methods Followed** 1](#_Toc162306945)

[**6.1.** **Burn Down Charts** 1](#_Toc162306946)

[**6.2.** **Minutes of Meetings** 1](#_Toc162306947)

[**6.3.** **User Stories** 5](#_Toc162306948)

[**6.4.** **Sprints** 7](#_Toc162306949)

[**6.4.1.** **Sprint 1** 7](#_Toc162306950)

[**6.4.2.** **Sprint 2** 7](#_Toc162306951)

[**6.4.3.** **Sprint 3** 7](#_Toc162306952)

[**6.4.4.** **Sprin 4** 7](#_Toc162306953)

[**6.4.5.** **Sprint 5** 7](#_Toc162306954)

[**6.4.6.** **Sprin 6** 7](#_Toc162306955)

[**6.5.** **Product Backlogs** 8](#_Toc162306956)

[**7.** **Screencast and Presentation** 1](#_Toc162306957)

[**8.** **Conclusion** 1](#_Toc162306958)

1. **Introduction**

In today's constantly evolving world of higher education, institutions need to adopt advanced technological solutions to streamline administrative processes and enhance student engagement. This report documents the creation of a secure web-based system that is customized to collect student contributions. The report provides a detailed account of the entire process, including conceptualization, design, and implementation. It covers various aspects such as database architecture, site development, functionality integration, testing methodologies, Agile practices, and final presentation. The report highlights the collaborative efforts, technical proficiency, and unwavering commitment to excellence that went into the project. Starting with foundational database design, the report then goes on to describe the site design, functionality implementation, and rigorous testing. It also delves into Agile project management methodologies and concludes with a professional presentation and screencast showcasing the system's capabilities. This comprehensive narrative underscores the transformative impact of the project and the dedication of all involved toward achieving operational excellence.

1. **Database**

The selection of MySQL as the database management system for the project was based on its robust features, reliability, and widespread adoption in web-based applications. The system's proven track record in handling large volumes of data efficiently and securely makes it an exceptional choice for managing student contributions to the university magazine.

Moreover, the advanced security features, such as role-based access control and encryption, offered by MySQL align perfectly with the project's requirements for safeguarding sensitive student data. Its flexibility in handling various data types and scalability enables it to accommodate diverse content submissions expected for the magazine.

In addition, MySQL's compatibility with popular programming languages and frameworks commonly used in web development simplifies the integration process with the rest of the system components. This interoperability facilitates smooth data interactions between the web-based application and the database backend, thus streamlining the development process.

Overall, MySQL's robustness, security, and flexibility make it a suitable choice for the project, and we are confident that its adoption will enhance the reliability and efficiency of the system.

Oracle. (n.d.). MySQL. Retrieved from <https://www.mysql.com/>

MySQL Documentation. (n.d.). Retrieved from <https://dev.mysql.com/doc/>

* 1. **Security**

MySQL is a widely used relational database management system that offers robust security features to safeguard sensitive data. These security measures encompass various aspects such as authentication, authorization, encryption, access control, and auditing.

Security was of utmost importance in handling user credentials within the database. A robust approach was adopted to ensure the confidentiality and integrity of sensitive information. The MySQL database used for storing user details underwent meticulous security enhancements. To avoid storing plaintext passwords within the database, the application used PHP's built-in password hashing functions, specifically the password\_hash() method.

Instead of storing passwords directly, the application generated salted and hashed representations using the password\_hash() function before storing them in the database. This cryptographic hashing technique transformed passwords into irreversible, encrypted representations, enhancing security. During the authentication process, the hashed password stored in the database was compared with the hashed version of the user's input, verifying their identity without exposing sensitive information.

By employing this approach, user accounts remained safeguarded even in the unlikely event of a security breach. The use of robust hashing algorithms and salting mechanisms provided a resilient defence against unauthorized access and data breaches, ensuring the confidentiality and integrity of user credentials within the MySQL database.

* 1. **Appropriate Data Types and Validation**

Proper selection of data types and implementation of validation mechanisms are crucial elements in designing a database, especially when managing student contributions for the university magazine project. The database serves as the central repository for storing and organizing different types of information, from student details to article submissions.

When it comes to data types, it is important to choose the appropriate data type that accurately represents the nature of the stored data while optimizing storage efficiency. For example, VARCHAR data types are ideal for variable-length strings such as names and article titles, INT data types for numerical identifiers like student IDs, and DATE/DATETIME data types for temporal information such as submission deadlines or comments.

Validation mechanisms act as safeguards against inconsistent and erroneous data entry, bolstering data integrity within the database. Through rigorous validation, the system ensures that only valid and consistent data is accepted, reducing the risk of errors and maintaining the reliability of the database. These mechanisms include a variety of techniques, such as data format validation, range validation, presence validation, referential integrity constraints, and unique constraints.

For the project, data format validation ensures that inputs such as email addresses adhere to standard formats, while range validation verifies that numerical values fall within acceptable boundaries. Presence validation enforces the completion of mandatory fields, preventing the entry of empty or null values. Additionally, referential integrity constraints establish and maintain relationships between different entities within the database, such as linking articles to their respective authors (students). Meanwhile, unique constraints prevent the duplication of essential information, such as ensuring each student is assigned a unique identifier (student ID).

By carefully selecting appropriate data types and implementing robust validation mechanisms, the database accurately represents the information relevant to student contributions while ensuring its integrity and consistency. This foundational aspect of database design underpins the effective management and organization of data, facilitating seamless operations and enhancing the overall functionality of the university magazine project.

* 1. **ERD**

To guarantee the successful implementation of crucial features, an Entity Relationship Diagram (ERD), as Figure 1 presents, was created for the magazine management system's database. This diagram aids the frontend application in achieving the necessary functionality, particularly regarding role implementation and accommodating multiple articles for a contribution. Seven tables have been thoughtfully implemented, each containing suitable constraints, which enables the formation of links between them.

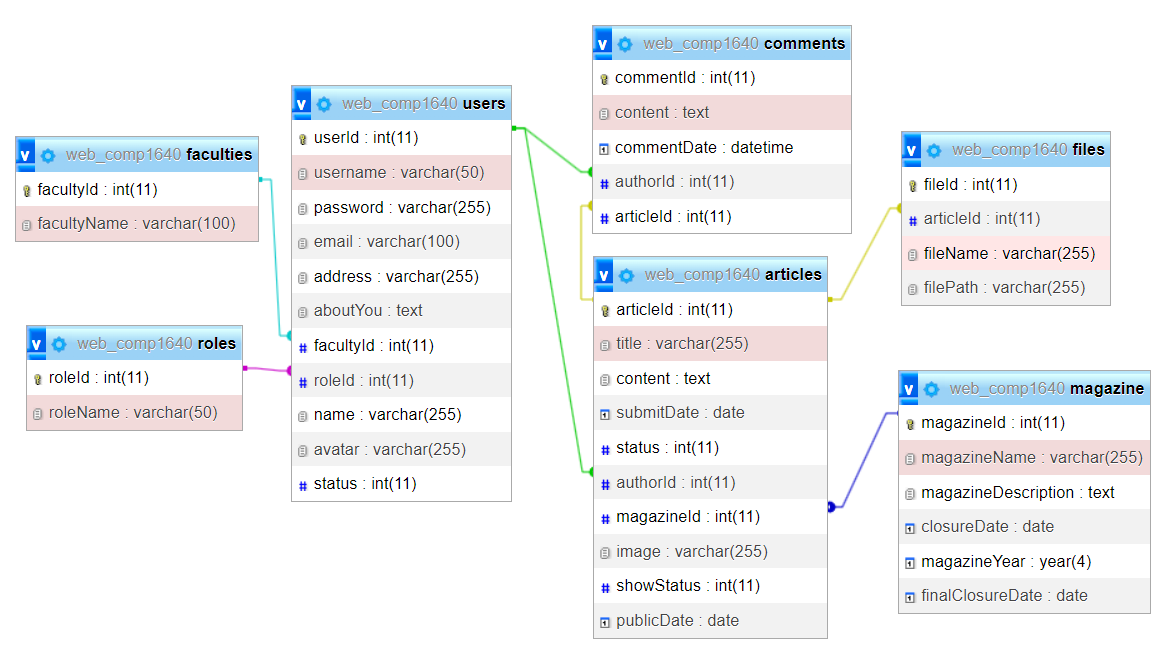


Figure 1. Entity relationship diagram

The following text explains the Entity-Relationship Diagram (ERD) Figure 1, which represents the system of a university magazine project that manages student contributions. The ERD diagram shows all entities in the project database, consisting of 7 tables and their relationships. In this project, each entity plays an important role and will be discussed in detail below.

The users entity is a fundamental element in the student magazine system's ERD. It manages user account for various individual interecting with the system and provides a mechanism for user authentication and authorization. The attributes of uses entity including userId (Primary Key), username, password, email, address, aboutYou, facultyID (for faculty users), roleId (for role of users), name, avatar, and status (for active or inactive account).

The roles entity plays a vital role in the student magazine system by establing a permission structure for different user types. It defines various user roles within the system and specific permissions ensuring a controlled and secure eviroment. The attributes of roles entity including roleId (Primary Key) and roleName.

The faculties entity plays an optional role in the student magazine system, depending on the specific functionalities designed for faculty involvement. It stores information about faculties that each student must be belong to a faculty. The attributes of faculties including facultyId (Primary Key) and facultyName.

The articles entity plays a central role in the student magazine system, serving as the repository for student contribution. It stores information about submitted articles for the university magazine and provide a central location for managing and organizing student contributions. The attributes of articles entity including articleId (Primary Key), title, content, submitDate, publicDate, status (for articles are pending, approved or denied), authorId (for user submitted the article), magazineId (for magazine that article submit to), image, and showStatus (for approved articles are public or not).

The comments entity plays a crucial role in facilitating interaction and discussion around submitted articles. It stores comments left by users on submitted articles. The attributes of comments entity including commentId (Primary Key), content, commentDate, authorId (for the user comment), and articleId (for the article user comment).

The magazine entity plays an optional tole in the student magazine system, depending on how the system manages and organizes student contributions. It stores information about magazine issues and provide a way to categories and present articles within specific magazine issues. The attributes of magazine entity including magazineId (Primary Key), magazineName, magazineDescription, closureDate, finalClosureDate, and magazineYear.

The files entity plays a role in the student magazine system, depending on student upload one or more files alongside submitted articles. It stores information about uploaded files associated with articles and enables the system to manage and reference uploaded content such as images, files, and allow to download them.

* 1. **Referential Integrity**

Referential integrity is a fundamental principle in database design that maintains the reliability and consistency of relationships between tables. It enforces predefined rules on data integrity to ensure the validity of connections between related tables, preventing orphaned records or inconsistencies that could undermine the database's reliability.

For the project of overseeing student contributions for the university magazine, referential integrity holds significant importance in managing interrelated entities. By upholding referential integrity, the database guarantees that every reference from one table to another remains valid and that changes to related data are appropriately coordinated.

Foreign key constraints are the primary mechanism for enforcing referential integrity. These constraints establish relationships between tables, such as linking submissions to their respective authors (students). For example, the article table could include a foreign key referencing the student table, ensuring that each submission article is associated with a valid student record.

Additionally, cascade actions can be configured to dictate the behaviour of related records when a referenced record is modified or deleted. For instance, cascade deletion can automatically remove related submissions when a student record is deleted. However, it's crucial to be careful to prevent unintended data loss, and cascade actions should be used judiciously based on project requirements.

By adhering to referential integrity principles, the database maintains data consistency and integrity, averting orphaned records, inconsistent data, and other anomalies that may arise from incomplete or improper relationships. In our project's context, applying referential integrity ensures the coherence of student contributions data, thereby enhancing the effectiveness and trustworthiness of the university magazine's database.

In conclusion, referential integrity is a cornerstone of database design that bolsters data reliability and accuracy by enforcing consistent relationships between tables. Its application in our project's database schema ensures the integrity and coherence of student contributions data, facilitating seamless management and organization within the university magazine project.

* 1. **Role Implementation**

Role implementation is a critical part of the project, as it determines how users interact with the system based on their roles and responsibilities. The project requirements call for five different roles, including Administrator, University Marketing Manager, Faculty's Marketing Coordinator, Student, and Guest. Implementing these roles ensures efficient management of user access and system functionalities.

Each role within the system has unique permissions and responsibilities, defining the scope of actions users can perform. The Administrator role has the highest level of authority, granting almost full access to system functionalities such as user management and maintenance of data such as magazines and faculties. On the other hand, the Guest role is granted limited permissions, allowing them to browse published content without engaging in interactive features.

Several key steps are involved in implementing roles within the project. First, the database schema must be designed to accommodate user roles. As shown in Figure 1 of the ERD, the users table has an attribute role, ensuring scalability and flexibility for future enhancements. Authentication mechanisms are then implemented to verify user identities, while authorization logic enforces role-based access control, dictating which functionalities are accessible to users based on their roles.

Role-based access control assigns specific permissions to each role, mapping out the functionalities accessible to users. This ensures that users are granted only the permissions necessary to perform their tasks, adhering to the principle of least privilege and minimizing the risk of unauthorized access. Furthermore, customization of the user interface is necessary to provide a tailored experience for users based on their roles. By displaying relevant features and options specific to each role, the system ensures an intuitive and streamlined user experience. Error-handling mechanisms are also implemented to manage unauthorized access attempts gracefully and provide informative feedback to users, enhancing usability and security.

In summary, effective role implementation within the project's database schema and application logic strengthens access control, enhances data security, and preserves the integrity of the system. By defining roles and assigning appropriate permissions, the system ensures that users can carry out their tasks efficiently while maintaining a secure environment for managing student contributions to the university magazine.

1. **Site Design**

Designing the website for the project is of utmost importance to ensure a smooth user experience and reflect the project's professionalism. Key factors that need to be considered include responsive design that can adjust to various devices, clear and easy-to-follow information architecture for seamless navigation, visually appealing design consistent with the branding, usability and accessibility features, efficient presentation of content, and compatibility with multiple web browsers. By prioritizing these aspects, we can create a user-friendly interface for managing student contributions to the university magazine.

* 1. **Responsive Design**
  2. **Information Architecture**
  3. **Usability Heuristics**

Usability heuristics are a set of principles proposed by Jakob Nielsen in 1995 to evaluate and enhance the usability of a user interface. By applying these heuristics to a project, one can identify usability issues and guide the design and development process towards creating a more user-friendly system. Let's briefly discuss each of Nielsen's heuristics and explore how they can be applied to the project.

First and foremost is the principle of visibility of system status. Users should be kept informed of system actions, feedback, and progress at all times. For instance, when a user submits content or performs an action, clear feedback should be provided to indicate the status of their request, ensuring they remain aware of the system's response.

Aligning the system with real-world concepts and tasks is another crucial heuristic. The language, layout, and functionality of the project should resonate with users' expectations and experiences. Using familiar terminology and intuitive navigation paths, such as labelling buttons with clear action verbs like "Submit Article" or "View Submissions", helps users navigate the system effortlessly and reinforces their mental model of the task at hand.

Offering users control and freedom within the system is essential for a positive user experience. Users should feel empowered to explore the system without fear of making irreversible errors. Providing clear options for undoing actions, navigating back, or cancelling processes ensures users have the freedom to interact with the system confidently, without worrying about unintended consequences.

Consistency and adherence to standards play a significant role in maintaining usability. By following established UI design patterns and maintaining consistency in design elements and interactions throughout the system, users can predictably navigate and interact with the interface. Consistent button styles, placement of navigation elements, and colour schemes contribute to a seamless user experience and reduce cognitive load.

Error prevention is paramount in ensuring smooth user interactions. By anticipating and preventing errors before they occur, the system can guide users toward correct actions and minimize frustration. Implementing validation checks, clear instructions and constraints throughout the user journey helps users avoid potential pitfalls and ensures a smooth workflow.

Recognition rather than recall is another key heuristic to consider. Users should not be required to remember information from one part of the system to another. Instead, information, options, and actions should be presented in a format that is readily recognizable and easily accessible. Descriptive labels, tooltips, and contextual help can aid users in navigating complex tasks without relying on memory.

Flexibility and efficiency of use are important considerations, particularly for accommodating users with varying needs and skill levels. Providing flexible interaction options, such as keyboard shortcuts or advanced features for experienced users, alongside simpler interfaces for novices, ensures a tailored experience for all users. This approach allows users to interact with the system in a way that best suits their preferences and proficiency levels.

Aesthetic and minimalist design principles contribute to a visually appealing and user-friendly interface. By focusing on essential elements and minimizing distractions, the system can create a clear and intuitive user experience. Utilizing whitespace, clear typography, and visual hierarchy helps prioritize important content and guide users' attention effectively.

Helping users recognize, diagnose, and recover from errors is essential for maintaining usability. Clear and actionable error messages, along with guidance and recovery options, assist users in resolving issues and completing tasks successfully. Accessible help resources and documentation further support users who require additional assistance, ensuring they can easily find answers to their questions and troubleshoot problems effectively.

Finally, providing help and documentation within the system helps users access additional information and support as needed. Contextual help, tooltips, FAQs, and user guides offer valuable resources for users seeking assistance with specific tasks or features. By ensuring help resources are easily accessible and tailored to users' needs, the system can empower users to overcome challenges and make the most of its functionality.

By applying Nielsen's heuristics to the project's user interface design, designers and developers can identify usability issues, streamline interactions, and create a more intuitive and user-friendly system for managing student contributions to the university magazine.

[10 Usability Heuristics for User Interface Design (nngroup.com)](https://www.nngroup.com/articles/ten-usability-heuristics/)

* 1. **Usability**
  2. **Accessibility**

Ensuring that a project meets accessibility criteria is of utmost importance to provide an inclusive user experience. Compliance with accessibility standards, such as the Web Content Accessibility Guidelines (WCAG), is crucial. Adhering to WCAG guidelines ensures that the system is perceivable, operable, understandable, and robust for all users, regardless of their abilities or disabilities.

One vital aspect of accessibility is providing alternative text for images. This involves including descriptive alternative text (alt text) for all images within the system. Alt text conveys the content and purpose of the image to users who may be using screen readers or have images disabled, ensuring that everyone can access and understand the information conveyed by images.

Keyboard navigation support is essential to accommodate users who rely on keyboard input for navigation. All interactive elements such as links, buttons, and form fields should be operable using only a keyboard. Implementing keyboard navigation support enables users with mobility impairments or dexterity limitations to access all features and functionalities of the system effectively.

Visible focus indication is another crucial aspect of accessibility. Users who navigate using a keyboard rely on visible focus indicators to understand which element is currently in focus. Providing clear focus indication ensures that users can navigate through the interface with ease, enhancing usability for those with visual impairments or mobility limitations.

Colour contrast and visual design also play a significant role in accessibility. Adequate color contrast ratios ensure that text and interactive elements are easily readable for users with low vision or color blindness. Clear typography and sufficient contrast between text and background colors enhance readability and ensure that information is accessible to all users, regardless of their visual abilities.

Accessible forms and controls are essential for users with disabilities. Forms should be designed and labeled in a way that is accessible to users using screen readers or assistive technologies. Providing descriptive labels, proper field grouping, and clear instructions improves the usability of forms for all users, enhancing accessibility and user experience.

Responsive design is crucial for ensuring accessibility across various devices. The system should adapt seamlessly to different screen sizes and devices, allowing users to access it from desktops, laptops, tablets, and smartphones. Responsive design principles ensure a consistent and accessible user experience across all devices, catering to users' diverse needs and preferences.

Conducting thorough testing with accessibility evaluation tools and assistive technologies is essential to identify and address accessibility barriers. Testing the system with screen readers, magnifiers, and other assistive technologies helps ensure that it is usable and accessible to users with diverse needs, meeting the accessibility criteria and providing an inclusive user experience for all.

1. **Functionality**
   1. **Role Based Security**

In the project, implementing role-based security involves defining five roles: administrator, university marketing manager, faculty's marketing coordinator, student, and guest. Each role is assigned specific permissions based on their responsibilities and privileges within the system.

For users with the role of student or guest, access to features reserved for administrators, managers, and coordinators is restricted. When a student attempts to access such features, they are redirected to another page or receive an "access denied" message. This ensures that students only have access to functionalities relevant to their role, such as submitting contributions or viewing their own submissions.

The role-based access control mechanism ensures that each user is granted access only to the functionalities and data necessary for their role, enhancing security and maintaining data privacy and confidentiality within the system. Additionally, administrators have the ability to manage user roles and permissions, ensuring that access control is effectively enforced and aligned with the project's security requirements.

* 1. **Submission of Report**
  2. **Email Notification**
  3. **Summary and Exception Reports**
  4. **UML diagram**

1. **Testing**
   1. **Test Plan**

**Objective**: To ensure the functionality, security, and usability of the web-based secure role-based system for managing student contributions to the university magazine.

**Scope**: The testing will cover all system functionalities, including user authentication, role-based access control, submission of reports, email notifications, database integrity, site design, responsiveness, and overall user experience.

**Testing Approach**: Manual testing will be conducted by designated testers using predefined test cases. Automated testing tools may also be employed for certain functionalities, such as database integrity checks and security vulnerabilities scanning.

**Test Environment**: The system will be tested on various browsers (Google Chrome, Microsoft Edge, Firefox, Safari) and devices (desktop, laptop, tablet, smartphone) to ensure compatibility and responsiveness across different platforms.

**Testing Time**: 26/02/2024 – 07/04/2024

* + 1. **Test Case Sprint 1**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **User’s Story** | **Test case** | **Function** | **Testing Data** | **Expected output** | **Actual Output** | **Result** |
| 1 | As a new user, I want to be able to register for an account as Student account |  | Register for Student |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

* + 1. **Test Case Sprint 2**
    2. **Test Case Sprint 3**
    3. **Test Case Sprint 4**
    4. **Test Case Sprint 5**
    5. **Test Case Sprint 6**
  1. **Test Log**
  2. **Sufficient Data to Fully Test**

1. **Agile Methods Followed**
   1. **Burn Down Charts**
   2. **Minutes of Meetings**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Meeting No** | **Date** | **Minutes of meeting** | **What was discussed** | **Tasks Achieved:** | **Member present:** |
| 1 | 20/02/2024 | 60 | 1. Discussed technologies to use 2.Created User Stories | 1. Choose PHP, Mysql, boostrap 2. Finish all user stories | Luan, Chinh, Kiet, Tuan, Minh, Anh |
| 2 | 22/02/2024 | 60 | 1. Broke down user stories to tasks 2. Prioritsed tasks in backlog 3. Assigned tasks to team members | 1. Finish product backlog with priority 2. Plan all 6 sprints 3. Assigned tasks to team members | Luan, Chinh, Kiet, Tuan, Minh, Anh |
| 3 | 23/02/2024 | 60 | 1. Discussed database 2. Discussed Usecase | 1. Create ERD diagram 2. Create Usecase diagram | Luan, Chinh, Minh |
| 4 | 26/02/2024 | 60 | 1. Review user stories in Sprint 1 2. Task Breakdown | 1. Discuss the user stories planned for the Sprint 1 2. Break down user stories into smaller tasks and assign them to team members. | Luan |
| 5 | 2/3/2024 | 120 | 1. Discussed complete and incomplete tasks in Sprint 1 2. Demo of completed work | 1. Complete tasks in Sprint 1 2. Each team member demonstrated the tasks they completed during the sprint 3. The results of the demo work match with the user stories. | Luan, Chinh, Kiet, Tuan, Minh, Anh |
| 6 | 3/3/2024 | 60 | 1. Reflect on what went well during the Sprint 1 and areas for improvement. 2. What tasks worked in Sprint 1? 3. What tasks did not work in Sprint 1? | 1. All tasks in Sprint 1 are worked. 2. Project are use PHP and MySQL. 3. Complete register, login, logout, and manage user accounts, roles and permission functionalities well. | Luan, Chinh, Kiet, Tuan, Minh, Anh |
| 7 | 4/3/2024 | 90 | 1. Review user stories in Sprint 2 2. Task Breakdown | 1. Discuss the user stories planned for the Sprint 2 2. Break down user stories into smaller tasks and assign them to team members. | Luan |
| 8 | 9/3/2024 | 120 | 1. Discussed complete and incomplete tasks in Sprint 2 2. Demo of completed work | 1. Complete tasks in Sprint 2 2. Each team member demonstrated the tasks they completed during the sprint 3. The results of the demo work match with the user stories. | Luan, Chinh, Kiet, Tuan, Minh, Anh |
| 9 | 10/3/2024 | 60 | 1. Reflect on what went well during the Sprint 2 and areas for improvement. 2. What tasks worked in Sprint 2? 3. What tasks did not work in Sprint 2? | 1. All tasks in Sprint 2 are worked. 2. Complete submissions articles and send email functionalities. | Luan, Chinh, Kiet, Tuan, Minh, Anh |
| 10 | 11/3/2024 | 90 | 1. Review user stories in Sprint 3 2. Task Breakdown | 1. Discuss the user stories planned for the Sprint 3 2. Break down user stories into smaller tasks and assign them to team members. | Luan |
| 11 | 16/03/2024 | 120 | 1. Discussed complete and incomplete tasks in Sprint 3 2. Demo of completed work | 1. Almost tasks complete tasks in Sprint 3 2. Each team member demonstrated the tasks they completed during the sprint 3. Error logic occur that student can submit articles even expired closure date and final closure date. 4. Error logic occur that coordinator in a faculty still view all articles from another faculty. | Luan, Chinh, Kiet, Tuan, Minh, Anh |
| 12 | 17/03/2024 | 60 | 1. Reflect on what went well during the Sprint 3 and areas for improvement. 2. What tasks worked in Sprint 3? 3. What tasks did not work in Sprint 3? | 1. Verify two error logic occured and found solutions to fix them. 2. Process to fix errors through Sprint 4. 3. Complete view and give feedback/comment functionalities | Luan, Chinh, Kiet, Tuan, Minh, Anh |
| 13 | 18/03/2024 | 90 | 1. Review user stories in Sprint 4 2. Task Breakdown | 1. Discuss the user stories planned for the Sprint 4 2. Break down user stories into smaller tasks and assign them to team members. | Luan |
| 14 | 23/03/2024 | 120 | 1. Discussed complete and incomplete tasks in Sprint 4 2. Demo of completed work | 1. Complete tasks in Sprint 4 2. Each team member demonstrated the tasks they completed during the sprint 3. The results of the demo work match with the user stories. 4. Complete fix errors existed in Sprint 3. | Luan, Chinh, Kiet, Tuan, Minh, Anh |
| 15 | 24/03/2024 | 60 | 1. Reflect on what went well during the Sprint 4 and areas for improvement. 2. What tasks worked in Sprint 4? 3. What tasks did not work in Sprint 4? | 1. All tasks in Sprint 4 are worked. 2. Complete download files and manage status of articles functionalities. | Luan, Chinh, Kiet, Tuan, Minh, Anh |
| 16 | 25/03/2024 | 90 | 1. Review user stories in Sprint 5 2. Task Breakdown | 1. Discuss the user stories planned for the Sprint 5 2. Break down user stories into smaller tasks and assign them to team members. | Luan |
| 17 | 30/03/2024 | 120 | 1. Discussed complete and incomplete tasks in Sprint 5 2. Demo of completed work | 1. Complete tasks in Sprint 5. 2. Each team member demonstrated the tasks they completed during the sprint 3. The results of the demo work match with the user stories. | Luan, Chinh, Kiet, Tuan, Minh, Anh |
| 18 | 31/2024 | 60 | 1. Reflect on what went well during the Sprint 5 and areas for improvement. 2. What tasks worked in Sprint 5? 3. What tasks did not work in Sprint 5? | 1. All tasks in Sprint 5 are worked. 2. Complete manage profile functionalities and non-functional requirements | Luan, Chinh, Kiet, Tuan, Minh, Anh |
| 19 | 1/4/2024 | 90 | 1. Review user stories in Sprint 6 2. Task Breakdown | 1. Discuss the user stories planned for the Sprint 6 2. Break down user stories into smaller tasks and assign them to team members. | Luan |
| 20 | 6/4/2024 | 120 | 1. Discussed complete and incomplete tasks in Sprint 6 2. Demo of completed work | 1. Complete tasks in Sprint 6. 2. Each team member demonstrated the tasks they completed during the sprint 3. The results of the demo work match with the user stories. | Luan, Chinh, Kiet, Tuan, Minh, Anh |
| 21 | 7/4/2024 | 60 | 1. Reflect on what went well during the Sprint 6 and areas for improvement. 2. What tasks worked in Sprint 6? 3. What tasks did not work in Sprint 6? | 1. All tasks in Sprint 5 are worked. 2. Complete statistics functionalities and make website responsive. | Luan, Chinh, Kiet, Tuan, Minh, Anh |

* 2. **User Stories**

As a new user, I want to be able to register for an account as Student account

As a registered user, I want to be able to log in and log out of my account

As a system administrator, I want to be able to manage user accounts, roles and permissions to control access to system features.

As a registered user, I want to be able to view and edit my profile information to keep my information up to date.

As a registered user, I want to have the option to change my password for security reasons.

As a student, I want to agree to the Terms and Conditions before submitting my work.

As a student, I want to submit articles and upload images for consideration in the university magazine.

As a administrator, I want to manage the magazine for each academic year and update closure dates until a final closure date expires.

As a faculty coordinator, I want to receive an email notification upon receiving a new submission from a student.

As a administrator, I want to manage information of faculties.

As a faculty coordinator, I want access to submissions from students within my faculty for review and approve.

As a faculty coordinator, I want to provide feedback/comments on submitted articles for students to improve their work.

As a student, I want to receive email notifications regarding the status of my submissions and any feedback provided.

As a student, I want to view feedback/comments on my submissions to understand areas for improvement.

As a student, I want to update my submissions the article to faculty's coordinator approve it.

As the University Marketing Manager, I want to view all selected contributions but not be able to edit them.

As the University Marketing Manager, I want to download all selected contributions in a ZIP file after the final closure date for transfer out of the system

As the University Marketing Manager, I want to manage approved articles of all faculties and then select articles to be public on the system so that the guests can view them.

As a student, I want to be able to download documents related to my submissions

As a guest user, I want to view selected reports from each faculty.

As a System Administrator, I want to generate statistical reports on the number of contributions per Faculty for each academic year.

As a Marketing Coordinator, I want to access reports on the status of submissions within my Faculty

As a University Marketing Manager, I want to view reports on the overall progress of the submission process

As a system administrator, I want the password of each account should be at least 8 characters, inclusion of uppercase letters, lowercase letters, numbers, and special characters.

As a user, I want the interface to be responsive and accessible on all devices for ease of use.

As a user, I want the system to have role-based access control to ensure data security and privacy.

As a user, I want data transmission and storage to be secure to protect my submissions and information.

As a registered user, I want the application to validate and sanitize my inputs to prevent attacks like SQL injection and cross-site scripting, ensuring that my personal information remains secure.

As a Product owner, I want to use php and mysql technology

* 2. **Sprints**
     1. **Sprint 1**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint goal:** | Make the application run on Php, Mysql. Focus on user management functionalities. | | | | | | | | | |
| **Start date:** | 26/02/2024 | | | | | | | | | |
| **End date:** | 1/3/2024 |  |  |  | **Day 1** | **Day 2** | **Day 3** | **Day 4** | **Day 5** |  |
| **Backlog ID** | **Backlog Item** | **Sprint task** | **Volunteer** | **Estimate effort hours** | **Mon** | **Tues** | **Wed** | **Thur** | **Fri** | **Total** |
| **Total estimate effort hours** | | | | **68** | **18** | **16** | **14** | **10** | **10** | **68** |
| 1 | As a new user, I want to be able to register for an account as Student account | Design registration page layout | Tuan | 4 | 4 |  |  |  |  | **4** |
| Implement functionality to handle user registration. | Minh | 2 |  |  | 2 |  |  | **2** |
| Test registration process and fix any bugs. | Kiet | 4 |  |  |  | 4 |  | **4** |
| 2 | As a registered user, I want to be able to log in and log out of my account | Design login page layout | Tuan | 4 |  | 4 |  |  |  | **4** |
| Implement functionality to handle user authentication | Chinh | 4 |  |  | 4 |  |  | **4** |
| Create session management for logged-in users. | Minh | 1 |  |  | 1 |  |  | **1** |
| Implement logout functionality. | Anh | 1 |  |  | 1 |  |  | **1** |
| Test login and logout processes and fix any bugs. | Kiet | 4 |  |  |  |  | 4 | **4** |
| 3 | As a system administrator, I want to be able to manage user accounts, roles and permissions to control access to system features. | Create UI for system administrator to view and modify user roles and permissions. | Anh | 6 | 6 |  |  |  |  | **6** |
| Implement functionality to manage user, accounts, roles and permissions. | Chinh, Luan | 12 |  | 12 |  |  |  | **12** |
| Test role and permission management functionality. | Kiet | 6 |  |  |  | 6 |  | **6** |
| 24 | As a system administrator, I want the password of each account should be at least 8 characters, inclusion of uppercase letters, lowercase letters, numbers, and special characters. | Implement password validation logic to ensure it contains at least 8 characters, one uppercase letter, one lowercase letter, one number, and one special character. | Chinh | 2 |  |  | 2 |  |  | **2** |
| Handle validation errors and provide appropriate error messages to users. | Minh | 4 |  |  | 4 |  |  | **4** |
| Test password validation functionality with various password combinations. | Tuan | 6 |  |  |  |  | 6 | **6** |
| 29 | As a Product owner, I want to use php and mysql technology | Set up PHP development environment and MySQL database. | Minh | 4 | 4 |  |  |  |  | **4** |
| Design and modify database | Chinh | 4 | 4 |  |  |  |  | **4** |

* + 1. **Sprint 2**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint goal:** | Implement submission, and email notifications functionalities. | | | | | | | | | |
| **Start date:** | 4/3/2024 | | | | | | | | | |
| **End date:** | 8/3/2024 |  |  |  | **Day 1** | **Day 2** | **Day 3** | **Day 4** | **Day 5** |  |
| **Backlog ID** | **Backlog Item** | **Sprint task** | **Volunteer** | **Estimate effort hours** | **Mon** | **Tues** | **Wed** | **Thur** | **Fri** | **Total** |
| **Total estimate effort hours** | | | | **62** | **12** | **17** | **13** | **12** | **8** | **62** |
| 6 | As a student, I want to agree to the Terms and Conditions before submitting my work. | Design layout for agreement checkbox and displaying Terms and Conditions. | Tuan | 4 | 4 |  |  |  |  | **4** |
| Implement functionality to require agreement before submission. | Minh | 1 |  | 1 |  |  |  | **1** |
| Test agreement process. | Kiet | 1 |  |  | 1 |  |  | **1** |
| 7 | As a student, I want to submit articles and upload images for consideration in the university magazine. | Design layout for article submission form. | Anh | 4 | 4 |  |  |  |  | **4** |
| Implement functionality to handle article submission. | Chinh | 6 |  | 6 |  |  |  | **6** |
| Test submission process. | Tuan | 4 |  |  | 4 |  |  | **4** |
| 9 | As a faculty coordinator, I want to receive an email notification upon receiving a new submission from a student. | Set up email notification system. | Luan | 2 | 2 |  |  |  |  | **2** |
| Design email template for submission notifications. | Anh | 2 |  | 2 |  |  |  | **2** |
| Implement functionality to trigger email notifications upon new submissions. | Luan | 4 |  | 4 |  |  |  | **4** |
| Test email notification functionality. | Kiet | 4 |  |  |  | 4 |  | **4** |
| 11 | As a faculty coordinator, I want access to submissions from students within my faculty for review and approve. | Design UI for accessing student submissions. | Tuan | 4 |  | 4 |  |  |  | **4** |
| Implement functionality to retrieve submissions from the database. | Chinh | 2 |  |  | 2 |  |  | **2** |
| Create functionality to filter submissions by faculty. | Minh | 2 |  |  | 2 |  |  | **2** |
| Implement functionality to approve submissions. | Minh | 4 |  |  |  | 4 |  | **4** |
| Test submission access and approve functionality. | Tuan | 4 |  |  |  |  | 4 | **4** |
| 15 | As a student, I want to update my submissions the article to faculty's coordinator approve it. | Design UI for updating submissions. | Anh | 2 | 2 |  |  |  |  | **2** |
| Implement functionality to retrieve the student's submissions from the database and update the submissions based on the student's changes | Chinh | 4 |  |  | 4 |  |  | **4** |
| Send a email notification to the faculty coordinator about the updated submission | Luan | 4 |  |  |  | 4 |  | **4** |
| Test the submission update functionality to ensure it works as expected | Kiet | 4 |  |  |  |  | 4 | **4** |

* + 1. **Sprint 3**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint goal:** | Implement magazine, faculties, feedback/comments managements, and email notifications managements. | | | | | | | | | |
| **Start date:** | 11/3/2024 | | | | | | | | | |
| **End date:** | 15/03/2024 |  |  |  | **Day 1** | **Day 2** | **Day 3** | **Day 4** | **Day 5** |  |
| **Backlog ID** | **Backlog Item** | **Sprint task** | **Volunteer** | **Estimate effort hours** | **Mon** | **Tues** | **Wed** | **Thur** | **Fri** | **Total** |
| **Total estimate effort hours** | | | | **56** | **12** | **12** | **16** | **8** | **8** | **56** |
| 8 | As a administrator, I want to manage the magazine for each academic year and update closure dates until a final closure date expires. | Design UI for managing magazines. | Anh | 4 | 4 |  |  |  |  | **4** |
| Implement functionality to store and retrieve information from magazines. | Minh | 2 |  | 2 |  |  |  | **2** |
| Create functionality to set initial closure date and final closure date for new submissions. | Minh | 2 |  | 2 |  |  |  | **2** |
| Implement functionality to allow updates closure date until final closure date. | Chinh | 2 |  | 2 |  |  |  | **2** |
| Test magazine and closure date management functionality. | Tuan | 4 |  |  | 4 |  |  | **4** |
| 10 | As a administrator, I want to manage information of faculties. | Design UI for managing faculties. | Tuan | 4 | 4 |  |  |  |  | **4** |
| Implement functionality to manage to store and restrieve information from faculties. | Luan | 2 |  | 2 |  |  |  | **2** |
| Test faculties managemnt functionality | Kiet | 4 |  |  | 4 |  |  | **4** |
| 12 | As a faculty coordinator, I want to provide feedback/comments on submitted articles for students to improve their work. | Design UI for providing feedback/comments. | Tuan | 2 | 2 |  |  |  |  | **2** |
| Create functionality to allow faculty coordinators to submit feedback/comments. | Minh | 2 |  |  | 2 |  |  | **2** |
| Implement functionality to store and retrieve feedback/comments for submitted articles. | Chinh | 4 |  |  | 4 |  |  | **4** |
| Test feedback/comment submission functionality. | Kiet | 4 |  |  |  | 4 |  | **4** |
| 13 | As a student, I want to receive email notifications regarding the status of my submissions and any feedback provided. | Implement functionality to generate and send notifications. | Luan | 4 |  | 4 |  |  |  | **4** |
| Create functionality to notify students of submission status changes. | Chinh | 2 |  |  | 2 |  |  | **2** |
| Test email notification functionality. | Kiet | 4 |  |  |  |  | 4 | **4** |
| 14 | As a student, I want to view feedback/comments on my submissions to understand areas for improvement. | Design UI for displaying feedback/comments. | Anh | 2 | 2 |  |  |  |  | **2** |
| Implement functionality to retrieve feedback/comments for submitted articles. | Anh | 2 |  |  |  | 2 |  | **2** |
| Create functionality to display feedback/comments to students. | Luan | 2 |  |  |  | 2 |  | **2** |
| Test viewing functionality for feedback/comments. | Tuan | 4 |  |  |  |  | 4 | **4** |

* + 1. **Sprin 4**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint goal:** | Implement download files, and public articles management. | | | | | | | | | |
| **Start date:** | 18/03/2024 | | | | | | | | | |
| **End date:** | 22/03/2024 |  |  |  | **Day 1** | **Day 2** | **Day 3** | **Day 4** | **Day 5** |  |
| **Backlog ID** | **Backlog Item** | **Sprint task** | **Volunteer** | **Estimate effort hours** | **Mon** | **Tues** | **Wed** | **Thur** | **Fri** | **Total** |
| **Total estimate effort hours** | | | | **52** | **14** | **12** | **14** | **6** | **6** | **52** |
| 16 | As the University Marketing Manager, I want to view all selected contributions but not be able to edit them. | Design UI for displaying selected contributions. | Anh | 4 | 4 |  |  |  |  | **4** |
| Implement functionality to retrieve selected contributions from the database. | Minh | 3 |  | 3 |  |  |  | **3** |
| Create functionality to filter contributions by selection status. | Minh | 1 |  | 1 |  |  |  | **1** |
| Test viewing functionality for selected contributions. | Tuan | 4 |  |  | 4 |  |  | **4** |
| 17 | As a University Marketing Manager, I want to download all selected contributions in a ZIP file after the final closure date for transfer out of the system | Implement functionality to gather all selected contributions. | Chinh | 2 | 2 |  |  |  |  | **2** |
| Create functionality to generate a ZIP file containing all selected contributions. | Luan | 6 | 6 |  |  |  |  | **6** |
| Test download functionality for the ZIP file. | Tuan | 4 |  | 4 |  |  |  | **4** |
| 18 | As the University Marketing Manager, I want to select approved articles to be public on the system so that the guests can view them. | Implement functionality to retrieve approved articles from all faculties. | Minh | 2 |  |  | 2 |  |  | **2** |
| Create functionality to mark articles as selected for public display and handle for public the article. | Anh | 2 |  |  |  | 2 |  | **2** |
| Test the article management functionality to ensure it works as expected | Kiet | 4 |  |  |  |  | 4 | **4** |
| 19 | As a student, I want to be able to download documents related to my submissions | Design UI for displaying submission documents. | Anh | 4 |  |  | 4 |  |  | **4** |
| Implement functionality to retrieve submission documents from the database. | Minh | 2 |  |  |  | 2 |  | **2** |
| Create functionality to generate download links for submission documents. | Luan | 2 |  |  |  | 2 |  | **2** |
| Test document download functionality for registered users. | Tuan | 2 |  |  |  |  | 2 | **2** |
| 20 | As a guest user, I want to view selected reports from each faculty. | Design UI for displaying selected reports. | Tuan | 2 | 2 |  |  |  |  | **2** |
| Implement functionality to retrieve selected reports from the database. | Minh | 2 |  | 2 |  |  |  | **2** |
| Create functionality to filter reports by faculty. | Chinh | 2 |  | 2 |  |  |  | **2** |
| Test viewing functionality for selected reports. | Kiet | 4 |  |  | 4 |  |  | **4** |

* + 1. **Sprint 5**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint goal:** | Implement manage profile, and non-requirement functionalities. | | | | | | | | | |
| **Start date:** | 25/03/2024 | | | | | | | | | |
| **End date:** | 29/03/2024 |  |  |  | **Day 1** | **Day 2** | **Day 3** | **Day 4** | **Day 5** |  |
| **Backlog ID** | **Backlog Item** | **Sprint task** | **Volunteer** | **Estimate effort hours** | **Mon** | **Tues** | **Wed** | **Thur** | **Fri** | **Total** |
| **Total estimate effort hours** | | | | **57** | **14** | **15** | **12** | **8** | **8** | **57** |
| 4 | As a registered user, I want to be able to view and edit my profile information to keep my information up to date. | Design profile page layout | Anh | 6 | 6 |  |  |  |  | **6** |
| Implement functionality to retrieve and display user profile information. | Chinh | 4 |  | 4 |  |  |  | **4** |
| Implement functionality to allow users to edit their profile information. | Kiet | 3 |  | 3 |  |  |  | **3** |
| Test profile viewing and editing processes and fix any bugs. | Tuan | 4 |  |  | 4 |  |  | **4** |
| 5 | As a registered user, I want to have the option to change my password for security reasons. | Design UI for password change layout | Anh | 4 | 4 |  |  |  |  | **4** |
| Implement functionality to handle password change requests. | Chinh | 4 |  | 4 |  |  |  | **4** |
| Test password change functionality and fix any bugs. | Tuan | 4 |  |  | 4 |  |  | **4** |
| 26 | As a user, I want the system to have role-based access control to ensure data security and privacy. | Implement functionality to encrypt sensitive data before storage. | Minh | 4 | 4 |  |  |  |  | **4** |
| Test data transmission and storage security. | Tuan | 4 |  | 4 |  |  |  | **4** |
| 27 | As a user, I want data transmission and storage to be secure to protect my submissions and information. | Implement functionality to encrypt sensitive data before storage. | Minh | 4 |  |  | 4 |  |  | **4** |
| Test data transmission and storage security. | Kiet | 4 |  |  |  |  | 4 | **4** |
| 28 | As a registered user, I want the application to validate and sanitize my inputs to prevent attacks like SQL injection and cross-site scripting, ensuring that my personal information remains secure. | Implement input validation for user input fields. | Kiet | 4 |  |  |  | 4 |  | **4** |
| Implement sanitization to prevent SQL injection attacks. | Minh | 2 |  |  |  | 2 |  | **2** |
| Implement sanitization to prevent cross-site scripting attacks. | Chinh | 2 |  |  |  | 2 |  | **2** |
| Test input validation and sanitization functionality. | Tuan | 4 |  |  |  |  | 4 | **4** |

* + 1. **Sprin 6**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint goal:** | Make website responsive and implement statistics functionalities. | | | | | | | | | |
| **Start date:** | 1/4/2024 | | | | | | | | | |
| **End date:** | 5/4/2024 |  |  |  | **Day 1** | **Day 2** | **Day 3** | **Day 4** | **Day 5** |  |
| **Backlog ID** | **Backlog Item** | **Sprint task** | **Volunteer** | **Estimate effort hours** | **Mon** | **Tues** | **Wed** | **Thur** | **Fri** | **Total** |
| **Total estimate effort hours** | | | | **57** | **8** | **14** | **14** | **9** | **12** | **57** |
| 21 | As a System Administrator, I want to generate statistical reports on the number of contributions per Faculty for each academic year. | Design UI for generating statistical reports. | Anh | 4 | 4 |  |  |  |  | **4** |
| Implement functionality to retrieve contribution data from the database. | Minh | 4 |  | 4 |  |  |  | **4** |
| Create functionality to generate reports based on contribution data. | Chinh | 6 |  | 6 |  |  |  | **6** |
| Test report generation functionality. | Tuan | 4 |  |  | 4 |  |  | **4** |
| 22 | As a Marketing Coordinator, I want to access reports on the status of submissions within my Faculty | Design UI for accessing submission status reports. | Tuan | 4 | 4 |  |  |  |  | **4** |
| Implement functionality to retrieve submission status data for the faculty. | Kiet | 4 |  |  | 4 |  |  | **4** |
| Create functionality to display submission status reports. | Anh | 3 |  |  |  | 3 |  | **3** |
| Test access to submission status reports. | Kiet | 4 |  |  |  |  | 4 | **4** |
| 23 | As a University Marketing Manager, I want to view reports on the overall progress of the submission process | Design UI for viewing overall progress reports. | Anh | 4 |  | 4 |  |  |  | **4** |
| Implement functionality to calculate overall submission progress. | Chinh | 3 |  |  |  | 3 |  | **3** |
| Create functionality to display overall progress reports. | Minh | 3 |  |  |  | 3 |  | **3** |
| Test access to overall progress reports. | Tuan | 4 |  |  |  |  | 4 | **4** |
| 25 | As a user, I want the interface to be responsive and accessible on all devices for ease of use. | Implement responsive design using CSS frameworks | Chinh | 6 |  |  | 6 |  |  | **6** |
| Test interface responsiveness on various devices and screen sizes. | Kiet | 4 |  |  |  |  | 4 | **4** |

* 1. **Product Backlogs**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Item ID** | **Items** | **Priority** | **Sprint (1-6)** | **Start date** | **Finish date** | **Status** | **Story point** | **Assigned to sprint (Y/N)** |
| **Functional Requirements** | | | | | | | | |
| 1 | As a new user, I want to be able to register for an account as Student account | High | 1 | **26/02/2024** | **29/02/2024** |  | 5 |  |
| 2 | As a registered user, I want to be able to log in and log out of my account | High | 1 | **27/02/2024** | **1/3/2024** |  | 5 |  |
| 3 | As a system administrator, I want to be able to manage user accounts, roles and permissions to control access to system features. | High | 1 | **26/02/2024** | **29/02/2024** |  | 8 |  |
| 4 | As a registered user, I want to be able to view and edit my profile information to keep my information up to date. | Medium | 5 | **25/03/2024** | **27/03/2024** |  | 3 |  |
| 5 | As a registered user, I want to have the option to change my password for security reasons. | Medium | 5 | **25/03/2024** | **27/03/2024** |  | 3 |  |
| 6 | As a student, I want to agree to the Terms and Conditions before submitting my work. | High | 2 | **4/3/2024** | **6/3/2024** |  | 2 |  |
| 7 | As a student, I want to submit articles and upload images for consideration in the university magazine. | High | 2 | **4/3/2024** | **6/3/2024** |  | 8 |  |
| 8 | As a administrator, I want to manage the magazine for each academic year and update closure dates until a final closure date expires. | High | 3 | **11/3/2024** | **13/03/2024** |  | 5 |  |
| 9 | As a faculty coordinator, I want to receive an email notification upon receiving a new submission from a student. | High | 2 | **4/3/2024** | **7/3/2024** |  | 5 |  |
| 10 | As a administrator, I want to manage information of faculties. | Medium | 3 | **11/3/2024** | **13/03/2024** |  | 5 |  |
| 11 | As a faculty coordinator, I want access to submissions from students within my faculty for review and approve. | High | 2 | **4/3/2024** | **8/3/2024** |  | 8 |  |
| 12 | As a faculty coordinator, I want to provide feedback/comments on submitted articles for students to improve their work. | High | 3 | **11/3/2024** | **14/03/2024** |  | 8 |  |
| 13 | As a student, I want to receive email notifications regarding the status of my submissions and any feedback provided. | High | 3 | **12/3/2024** | **15/03/2024** |  | 8 |  |
| 14 | As a student, I want to view feedback/comments on my submissions to understand areas for improvement. | High | 3 | **11/3/2024** | **15/03/2024** |  | 5 |  |
| 15 | As a student, I want to update my submissions the article to faculty's coordinator approve it. | High | 2 | **6/3/2024** | **8/3/2024** |  | 8 |  |
| 16 | As the University Marketing Manager, I want to view all selected contributions but not be able to edit them. | High | 4 | **18/03/2024** | **20/03/2024** |  | 8 |  |
| 17 | As the University Marketing Manager, I want to download all selected contributions in a ZIP file after the final closure date for transfer out of the system | High | 4 | **18/03/2024** | **19/03/2024** |  | 13 |  |
| 18 | As the University Marketing Manager, I want to manage approved articles of all faculties and then select articles to be public on the system so that the guests can view them. | High | 4 | **20/03/2024** | **22/03/2024** |  | 8 |  |
| 19 | As a student, I want to be able to download documents related to my submissions | Medium | 4 | **18/03/2024** | **22/03/2024** |  | 5 |  |
| 20 | As a guest user, I want to view selected reports from each faculty. | Medium | 4 | **18/03/2024** | **20/03/2024** |  | 3 |  |
| **Reporting** | | | | | | | | |
| 21 | As a System Administrator, I want to generate statistical reports on the number of contributions per Faculty for each academic year. | Medium | 6 | **1/4/2024** | **3/4/2024** |  | 13 |  |
| 22 | As a Marketing Coordinator, I want to access reports on the status of submissions within my Faculty | Medium | 6 | **1/4/2024** | **5/4/2024** |  | 8 |  |
| 23 | As a University Marketing Manager, I want to view reports on the overall progress of the submission process | Medium | 6 | **2/4/2024** | **5/4/2024** |  | 8 |  |
| **Non-Functional Requirements** | | | | | | | | |
| 24 | As a system administrator, I want the password of each account should be at least 8 characters, inclusion of uppercase letters, lowercase letters, numbers, and special characters. | High | 1 | **28/02/2024** | **1/3/2024** |  | 5 |  |
| 25 | As a user, I want the interface to be responsive and accessible on all devices for ease of use. | Medium | 6 | **3/4/2024** | **5/4/2024** |  | 8 |  |
| 26 | As a user, I want the system to have role-based access control to ensure data security and privacy. | High | 5 | **25/03/2024** | **26/03/2024** |  | 13 |  |
| 27 | As a user, I want data transmission and storage to be secure to protect my submissions and information. | High | 5 | **27/03/2024** | **29/03/2024** |  | 13 |  |
| 28 | As a registered user, I want the application to validate and sanitize my inputs to prevent attacks like SQL injection and cross-site scripting, ensuring that my personal information remains secure. | High | 5 | **28/03/2024** | **29/03/2024** |  | 13 |  |
| 29 | As a Product owner, I want to use php and mysql technology | High | 1 | **26/02/2024** | **26/02/2024** |  | 3 |  |

1. **Screencast and Presentation**

Presentation

Screencast

1. **Conclusion**