

**Software Engineering**

MOD003263

Faculty: Science and Technology

Department: Computing and Technology

Student ID: **2276245**

Submitted on: December 2024

Academic Year: 2024/25

Table of Contents  
1. Introduction............................................................................................................................3  
2. Problem Definition Document................................................................................................4  
3. Software Design......................................................................................................................5  
3.1. Wireframe.....................................................................................................................5  
3.2. Database design............................................................................................................5  
3.3. Software design. ...........................................................................................................5  
3.3.1. Use Case Diagram(s). ................................................................................................5  
3.3.2. Class Diagram............................................................................................................5  
3.3.3. Sequence Diagrams...................................................................................................6  
4. Software Development Documentation.................................................................................7  
4.1. User story 1...................................................................................................................7  
4.2. User story 2...................................................................................................................7  
5. Testing ....................................................................................................................................9  
6. Plan of the Software Maintenance .......................................................................................10  
7. Conclusions...........................................................................................................................11  
References (example!!!) .......................................................................................................................12

**Introduction**

This report documents the development of a software application designed to address a specific problem mentioned in the provided case study. The aim is to deliver a robust, functional, and a usable system that meets the user needs while providing to best practises in software engineering.

In this project, we formed a 4 student group to focus on understanding the current limitations on the existing system and creating a solution tailored to the problem at hand. Through our careful analysis, we identified key requirements and implemented them. Our approach integrates software design principles, modern development tools, and a clear way to ensure a well-structured and maintainable application.

This report serves as a guide to the project, covering all aspects of our development, including problem definition, design, implementation, testing, and maintenance planning. Each section provides insights into the methods used, challenges faced, and the outcomes achieved, offering a clear picture of the project's scope and impact.

**Problem Definition Document**

The primary aim of this project is to design and develop a software application that addresses the challenges outlined in the case study. The project focuses on delivering a solution that is functional, scalable, and user-friendly. Specifically, the objectives include:

* To analyze and understand the requirements of the current system.
* To design a well-structured solution with clear functionality and a responsible user interface.
* To implement a reliable and efficient software application using modern development tools and practises.
* To ensure the solution is modular, maintainable, and capable of future enhancements.
* To test the system thoroughly to minimize errors and ensure robustness and continued usability.

Resources Required to Complete the Project:

The successful completion of the project relies on the following team members, and tools.

Team Description:

* Alif Sathar- Responsible for creating Event page and Digital Content module. Few other tasks that are common to all group Members.
* Jakes Jacob Mathew-
* Muhammed Aseem-
* Niroshan Kekulawala-

Software Tools Used:

* Programming Language- C#
* Development Environment- Microsoft Visual Studio
* Database Management System- MySQL
* Design Tools- Lucidchard, Visual Paradigm, MermaidChart, Sequence diagram.org

List of User Stories:

* As a member, I want to browse and navigate through available events so that I can find and book events that interest me.
* As a member I want to view my membership details, including my interests and status, so that I can stay informed about my engagement with the community.
* As a member I want to see the events I have previously booked so that I can keep track of my participation in community activities.
* As a member I want to view the benefits so that I can make the most of my membership.
* As a member, I want to view all upcoming events so that I can plan and decide which ones to attend.
* As a member, I want to book an event online so that I can confirm my attendance and secure a spot.
* As an administrator, I want to search for specific events or time periods so that I can see attendance trends and event performance.
* As an administrator, I want to view members who have booked specific events so that I can manage attendance records efficiently.
* Digital Content Module
* As a member, I want to view the full list of available digital content modules so that I can choose what interests me.
* As a member, I want to enroll in digital content modules so that I can access and engage with the content.
* As a member, I want to see a dashboard showing utilized and unused benefits so that I can track my engagement and make the most of my membership.
* As a member, I want to see suggestions for content or events based on my past activities so that I can discover relevant and interesting opportunities.
* These user stories align with the initial four points of the requirements, ensuring coverage of viewing, booking, tracking, and accessing both events and digital content modules.
* As an administrator, I want to add new members and indicate their interests (Caring, sharing, creating, experiencing, working) when they sign up so that we can track how their interests shift over time in the community.
* As an administrator, I want to differentiate between membership types (Community Members, Key Access Members, Creative Workspace Members) so that I can manage different levels of access and benefits.
* As an administrator, I want to search for individual members and non-members to see all their events and visits to Together Culture so that I can track engagement and participation.

**Software Design**

This includes the design of the database and software:

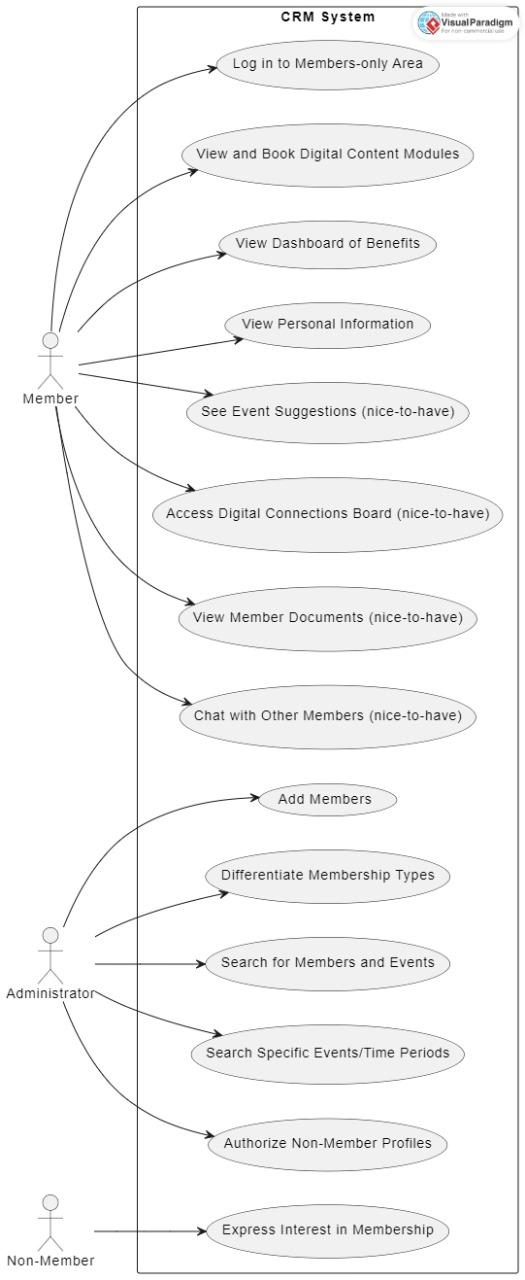
Wireframe: A screenshot of a calendar

Description automatically generated

Database Design: A close-up of a diagram

Description automatically generated

Use Case Diagram:



The provided use case diagram shows the functional requirements of a **Customer Relationship Management (CRM) System** by showcasing the interactions between three types of users: **Member**, **Administrator**, and **Non-Member**. The use cases represent various activities each role can perform within the system.

This diagram effectively shows the interactions and permissions across various user types in the CRM system, ensuring the system's functionalities are well-defined for each role.

Class Diagram:

A diagram of a company

Description automatically generated

The class diagram illustrates the **CRM System's structure**, showing key classes, attributes, methods, and their relationships. The system comprises several classes, including **Administrator**, **Member**, **LoginManager, and other supporting classes like Event, Module, Benefit, ActivityLog, and Tag. Each class contains attributes and methods that define its functionality.**

Sequence Diagrams:

**A screenshot of a computer

Description automatically generated**

**Software Development Documentation**

User Story 1:

* As a member, I want to browse and navigate through available events so that I can find and book events that interest me

Class Diagram:

A screenshot of a computer screen

Description automatically generated

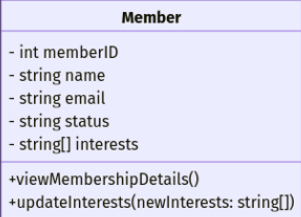
Sequence Diagram:

A screenshot of a computer

Description automatically generated

* As a member I want to view my membership details, including my interests and status, so that I can stay informed about my engagement with the community.

Class Diagram:



Sequence Diagram:

A screenshot of a black background

Description automatically generated

* As a member I want to see the events I have previously booked so that I can keep track of my participation in community activities

Class Diagram:

A screenshot of a computer screen

Description automatically generated

Sequence Diagram:

A diagram of a company

Description automatically generated

* As a member I want to view the benefits so that I can make the most of my membership these are my user stories

Class Diagram:

A diagram of a program

Description automatically generated

Sequence Diagram:

A diagram of a membership system

Description automatically generated

Our online Github Repository:  
https://github.com/JAKE-345/software-engineering

A screenshot of a computer

Description automatically generated

My contribution sneak peek

A screenshot of a website

Description automatically generated

**Testing**

Summarising the testing:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testing no. | Test Description | Expected Result | Actual Result | Pass/Fail |
| 1 | Testing whether it runs | Opens a window showing it worked with the contents displayed | It did open | Pass |
| 2 | To see if there is any bug | There is no bug | There was no bug | Pass |
| 3 | find out whether the buttons work | It will open a new form | It did open a new form.cs but got a slight glitch | Pass |
| 4 | To see if the page is stable | Would be working perfectly | It did work perfectly | Pass |

**Plan of the Software Maintenance**

Maintaining the software system is critical to ensure that it remains functional, secure, and aligned with the organization’s needs. This section outlines the maintenance approach for the Customer Relationship Management (CRM) solution developed for Together Culture.

Bug Fixes and Error Resolution:

Over time, unexpected bugs or issues may arise that affect the system’s functionality.

* Issue Tracking: All reported issues will be logged in a shared tracking system (e.g., on GitHub Issues).
* Priority Handling: Bugs will be prioritized based on their severity. Critical issues causing bugs will be resolved within 24–48 hours.
* Testing: Fixes will undergo testing to ensure they do not disrupt other parts of the system.

Adapting to Changes:

As the needs of Together Culture evolve or new technologies emerge, the system will require updates to remain relevant.

* Feature Enhancements: Additional features, such as personalized event suggestions or an integrated chat system like chatbot, may be introduced based on feedback.
* Environment Updates: The CRM will be reviewed for compatibility with updated browsers, operating systems, and allowing third-party tools.

Performance Optimization:

Improving the CRM’s performance and usability is a key part of long-term maintenance.

* User Feedback: Feedback from administrators and members will be collected to identify problems.
* System Speed: Database queries and search functionalities will be optimized to reduce response times.
* User Experience (UX): Enhancements, such as a more intuitive dashboard or streamlined booking process, will be considered to improve usability.

Preventive Measures:

To reduce the risk of system failure and ensure smooth operation:

* Regular Backups: Weekly automated backups of the database and application will be performed.
* Security Checks: The system will undergo quarterly security audits to identify and fix vulnerabilities.

Team Responsibilities:

* Developers: Resolve bugs, implement changes, and optimize system performance.
* System Administrator: Monitor the CRM, report issues, and manage backups.
* Project Manager: Oversee the maintenance schedule and coordinate tasks

CONCLUSION

This project demonstrates a very ell structured approach to developing a robust software solution for managing members' access to digital content modules and event recommendations. By systematically addressing the problem definition, software design, implementation, testing, and maintenance, the project ensures the delivery of a scalable and user-centric application.

I helped focusing my teammate on identifying user needs and converting them into clear requirements. This involved creating user stories like enrolling in digital content modules, getting personalized recommendations, and tracking engagement. These ensured the software addressed both functional and non-functional needs effectively.

I also contributed to the design phase by creating wireframes, the ERD, and UML diagrams. These designs helped visualize the system’s structure and behaviour.

I also helped in developing and documented test cases to ensure the software’s robustness. These tests validated functionality, performance, and reliability, ensuring the system met the defined requirements.

I also helped in creating a maintenance plan to address future updates, bug fixes, and enhancements. This ensures the software remains sustainable and adaptable to user needs over time.

This complicated and well-documented project demonstrates a strong grasp of software engineering principles. By combining a detailed problem definition, thoughtful design, efficient implementation, rigorous testing, and a clear maintenance strategy, it delivers a user-focused solution that meets the outlined objectives.