Table

Description automatically generated

Contents

[Acknowledgement 3](#_Toc105175649)

[Abstract 4](#_Toc105175650)

[Workload Matrix 5](#_Toc105175651)

[Introduction 6](#_Toc105175652)

[Project Plan 8](#_Toc105175653)

[System Development Methodology 8](#_Toc105175654)

[Project Gantt Chart 10](#_Toc105175655)

[System Hierarchy Chart 12](#_Toc105175656)

[Context Diagram 13](#_Toc105175657)

[Data Flow Diagram Level 0 14](#_Toc105175658)

[Data Flow Diagram Level 1 – Individual 15](#_Toc105175659)

[Data Dictionary – Individual 20](#_Toc105175660)

[Entity Relationship Diagram 33](#_Toc105175661)

[Pseudocode or Flowcharts – Individual 34](#_Toc105175662)

[Screens Design and User Manual 47](#_Toc105175663)

[Report Designs and User Manual 77](#_Toc105175664)

[Test Plan 78](#_Toc105175665)

[Unit Testing 78](#_Toc105175666)

[User Acceptance Testing 94](#_Toc105175667)

[Significant Source Codes 106](#_Toc105175668)

[Conclusion 107](#_Toc105175669)

[References 108](#_Toc105175670)

[Appendix 109](#_Toc105175671)

Acknowledgement

We would like to express our deepest appreciation to those who have provided us the possibility to complete this assignment. A special gratitude to our final semester project lecturer, Ms. Salasiah Sulaiman, who’s contributed to providing us suggestion and encouragement throughout the completion of this assignment and report.

Furthermore, we would also like to acknowledge with much appreciation the crucial role of the staff of Mr. Reshiwaran, who helped us do some system testing to complete our requirement for the test plan task. We also appreciate the guidance given by other supervisor as well as the panels especially in our project report that has improved our report writing skills thanks to their comment and advice. Our thanks and appreciations also go to our classmates in developing the project and people who have willingly helped us out with their abilities.

Abstract

We are creating a management system for our club and society department called Radiant club and society. This management system is to aid our club and society department to manage their clubs easier and help students who are interested to join, to have an easier time researching and participating in the clubs that we have. In this management system, we have various roles such as student, event organizer, manager and admin which will have their responsibilities. This management system is created due to problems that arise for our club and society department. Those problems include not being well known due to the COVID-19 pandemic, information about our clubs and society being insufficient and unclear to students and it is difficult for students to join our club and society because it requires them to come physically to register with us which students do not feel safe to do so during the pandemic. However, with the management system developed for our club and society department, it solves the problems that we have. With the management system developed, more exposure can be given to the students that are unaware of our club and society department. More information can be found on our website which makes student life easier to research the clubs and societies in which they are interested. Lastly, students can register with us through our website which is convenient for the students. When developing this management system, a system development methodology is written for our project plan where we will identify the requirements and system design for our system and then we go to the implementation phase where we will start to code the functions according to how we want our website to work. The purpose of this report is that we have highlighted the problems that we have as a physical department and having this management system developed for our department will solve those problems that we have which will improve the student’s quality of life.

Workload Matrix

Graphical user interface, application, table

Description automatically generated

# Introduction

**Project Background**

The system we are creating is a management system of the club and society, we named our system as Radiant club and society management system. This system allows students to view and participate the club in which they are interested. Next, there will be a few roles such as student, event organizer, manager and admin. Student are the ones who will view and participate the bodies while event organizer is responsible for the upcoming events or even announcement. Then, manager and admin have access to managerial actions such as manage users, modify information and many more. Furthermore, before student apply for the club or society they like, student would have to first log in to the website to have more access.

**Problem Context**

We have created this system to manage the few clubs and society in our system and to solve problems that the club has. One of the problems is that the clubs and society are not well-known by many students due to the recent pandemic that arises. With this system developed, we hope that it will bring more audience and participation. Furthermore, many of the clubs and society provides less information or unclear statement regarding what services they provides and hence with this system, we provide sufficient information to let guest understand and view clearly about what the club and society provides. Finally, students who are interested in joining the club are not able to register due to their inability to come physically, which is a problem affected by the pandemic. With the system developed, it can ease student to join the club by simply clicking a few buttons to register online.

**Proposed Solution**

Having a system developed for the Radiant club and society can give more exposure to the people who are interested in the club. On the website, there will be information about the clubs and society such as the history, structure, foundation, partners and more. After learning more about the clubs and society through the website, users can also view the programs and services that the club and society provides before joining the club to ensure users have the knowledge of what to expect beforehand if they are interested in joining. Lastly, after having a full preview of the chosen club or society, if the user decided to join, they can do so through the website by filling in their information into a form that is required to join the club and will soon be notified by the chosen club regarding their registration form.

**Project Objectives**

In this project, users can more easily join the club they want to join. This provide convenience for the students that want to join the club this is because they can easily know about the various clubs on our website. If they decide to join one of the clubs, they can fill out and submit the application form directly on the web page, which will greatly reduce the time for the users to go to the university to join the club. Using our system enables club managers to review, approve or reject user applications more effortless. Lastly, another advantage using our system is that you can get support from our staff in a brief time which will benefit students where they do not need to wait a long time to get support.

**Project Scope**

Our Radiant club and society management system is a website which require internet to access it, without internet, the system is unable to be access or load hence any new update or feature for the system is not able to be process. Next, our system is best view on desktop or any other devices with a large display since our website have a numerous clubs and society for users to view and chose from hence navigating the website is easier with a large display, mouse, and keyboard combination. Furthermore, our system require user to log in to access all the main function such as registration for a club or society, providing feedback and many more other functions. To add on, only admin can have access to managerial functions of all users while club and society manager is able to manage their members. Moreover, our system supports all type of web browser platform such as google chrome, safari, internet explorer and many other similar platforms.

# Project Plan

## System Development Methodology

**Waterfall model**

In this project, the suitable system development models for our system are waterfall model. Compared to other system development model, waterfall models are simpler constructed, easier to understand and operate. In waterfall models, it uses linear sequence flow to show the development process of software. Meaning that, each phase only begins when the previous phase is completed, and phases do not overlap. It helps the developer to focus on one phase until the phase is completed before proceeding to the next phase of the project, this can prevent any error from occurring. Waterfall model also referred to as Liner-Sequential Life Cycle Model. There are 6 phases in the waterfall models such as Requirement, System Design, Implementation, Integration, and Testing, Deployment of System, and Maintenance (Sharma, 2021).

**Requirements**

In the first stage of the waterfall model, we need to understand what needs to be designed on our website and what its function and purpose are, this will give us an idea of how it will work and look after the website is finished. These will be included in the specification document, and a feasibility study will be conducted to assess their effectiveness. Additionally, at this phase, it is important to consider any constraints or limitations that may delay website development, such as development budgets and time frames (Sharma, 2021).

**System design**

When it comes to the second phase, we examine the needs specification from the first phase and prepare the system design for our website. System design assists in defining the overall system architecture as well as specifying hardware and system requirements (Sharma, 2021). System design is further divided into high-level design stage and low-level design stage. It is crucial to breaking it down into individual components during the high-level design phase, while the low-level design takes that documentation and interprets each component, allowing programmers to build code directly from the documentation (ZULQADAR, 2019).

**Implementation**

During the Implementation phase, the source code is written according to the specifications during this phase, it is then converted into a functional code, which is initially developed in a small program called units and then integrated into the unit testing phase. (Sharma, 2021)

**Integration and Testing**

After the implementation phase, all unit-developed software is handed over to the testers to ensure that all required features work flawlessly. They will also perform software testing to determine whether there are any errors or bugs. If the tester found out any errors or bugs, the tester will report to the programmer to debug the errors or bugs so that the customer does not experience any technical issues while using the website (ZULQADAR, 2019).

**Deployment**

After all software testing is done in the integration and unit testing phase. At this phase, the website will be deployed to the Internet to check its performance. Once the website is stabilized, end users can use it (ZULQADAR, 2019).

**Maintenance**

The last phase is maintenance, which is the phase that occurs after the deployment phase. During this phase, we will collect all feedback from end-users, provide support and maintenance to the software to ensure that all functions work flawlessly. If the end-user encounters any errors or bugs, our programmers will fix them to ensure that the website is running smoothly (ZULQADAR, 2019).

## Project Gantt Chart

Table

Description automatically generated with medium confidence Table

Description automatically generated

A screenshot of a computer

Description automatically generated  
A picture containing table

Description automatically generated  
Table

Description automatically generated

# System Hierarchy Chart

Diagram, schematic

Description automatically generated

# Context Diagram

Diagram, schematic

Description automatically generated

# Data Flow Diagram Level 0

Diagram

Description automatically generated

# Data Flow Diagram Level 1 – Individual

**Lee Jia Qian TP061863**

Diagram

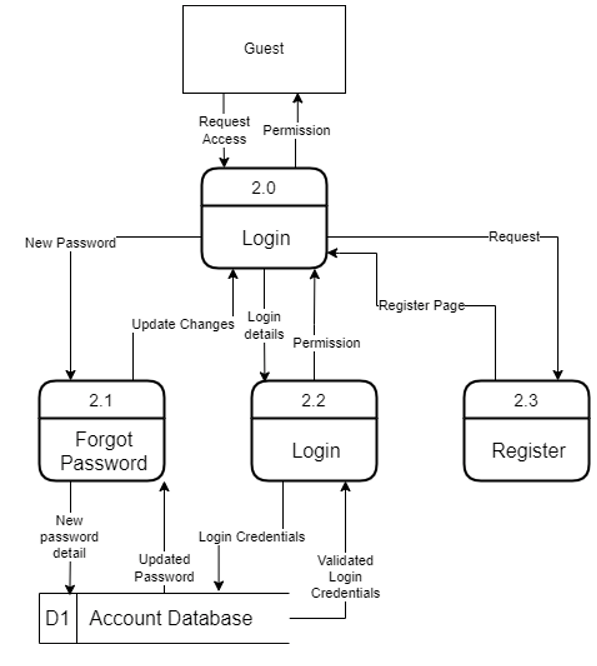
Description automatically generated

**Wong Yew Wei TP060747**

Diagram

Description automatically generated

**Tee Chor Yang TP061339**



**Thaddeus Then Ze Qin TP060326**

Diagram

Description automatically generated

**Teoh Yee Chin TP061586**

Diagram

Description automatically generated

# Data Dictionary – Individual

**Lee Jia Qian TP061863**

Data Dictionary – Individual – Log Out / Admin Page

**External Entity**

|  |  |
| --- | --- |
| **Name:** | Guest |
| **Description:** | User who is logged out or have not log in. |
| **Input Data Flows** | Guest home page |
| **Output Data Flows** | Login detail, register details |

|  |  |
| --- | --- |
| **Name:** | Admin |
| **Description:** | Gain access to admin page |
| **Input Data Flows** | Status |
| **Output Data Flows** | - |

**Process**

|  |  |
| --- | --- |
| **Name:** | 3.0 Log Out |
| **Description:** | Log out user from the system by ending session. |
| **Input Data Flows** | Log out |
| **Output Data Flows** | Guest home page |
| **Process:** | PROGRAM Log out:  Start session.  If log out button is clicked, clear the session of the user, otherwise user stays login.  END. |

|  |  |
| --- | --- |
| **Name:** | 6.0 Admin Page |
| **Description:** | Gain access to admin feature after validating user details. |
| **Input Data Flows** | Access |
| **Output Data Flows** | Permission |
| **Process:** | PROGRAM adminpage:  Start session;  If adminpage button is clicked, check database if user’s role equal to admin, if user’s role equal to admin, then the user is admin, otherwise the user stays student.  END |

**Data Flow**

|  |  |
| --- | --- |
| **Name:** | Log Out |
| **Description:** | Allow user to log out of the system |
| **Origin/Source:** | Log Out Process |
| **Destination/Sink:** | Guest external entity |
| **Data Structure:** | username |

|  |  |
| --- | --- |
| **Name:** | Status |
| **Description:** | Gain admin access after validation |
| **Origin/Source:** | Admin page process |
| **Destination/Sink:** | Admin external entity |
| **Data Structure:** | username |

**Data Store**

|  |  |
| --- | --- |
| **Name:** | Account Database |
| **Description:** | Stores registration, login details and various user details to help in validating roles while also provides updated users details. |
| **Input Data Flows:** | Validate access |
| **Output Data Flows:** | Validate access |
| **Data Structure:** | Sid, Student ID, Username, Password, First Name, Last Name, Intake, Contact Info, Email, IC/Passport, Gender, Birthday, Country, Club ID and Role |

**Wong Yew Wei TP060747**

Data Dictionary (Individual) – Edit Profile

**External Entity**

|  |  |
| --- | --- |
| Name | Student |
| Description | Student able to edit their profile |
| Input Data Flows | Updated Information |
| Output Data Flows | Profile Details |

**Process**

|  |  |
| --- | --- |
| Name | 4.0 Edit Profile |
| Description | This process allow student to edit their profile |
| Input Data Flows | Profile Details |
| Output Data Flows | Updated Information |
| Process | PROGRAM EditProfile:  IF EditProfile button is clicked, get the specific student ID, display the student profile details.  ELSE IF Submit button is clicked, get the specific student ID.  Update Password.  END. |

**Data Flows**

|  |  |
| --- | --- |
| Name | Profile Details |
| Description | Student's profile details will be sent to Process 4.0 Edit Profile to make modification |
| Origin/Source | Student |
| Destination/Sink | Process 4.0 Edit Profile |
| Data Structure | Student ID, Username, Password, First Name, Last Name, Intake, Contact Info, Email, IC/Passport, Gender, Birthday, Country, Club ID and Role |

|  |  |
| --- | --- |
| Name | Updated Information |
| Description | Student's profile details will be updated |
| Origin/Source | Process 4.0 Edit Profile |
| Destination/Sink | Student |
| Data Structure | SID, Student ID, Username, Password, First Name, Last Name, Intake, Contact Info, Email, IC/Passport, Gender, Birthday, Country, Club ID and Role |

Data Dictionary (Individual) – Committee

**External Entity**

|  |  |
| --- | --- |
| Name | Student |
| Description | Student who has a role committee able to access into committee features |
| Input Data Flows | Permission |
| Output Data Flows | Access |

|  |  |
| --- | --- |
| Name | Club Manager |
| Description | Committee able to view, approve application and modify their own club |
| Input Data Flows | Status |
| Output Data Flows | - |

**Process**

|  |  |
| --- | --- |
| Name | 7.0 Committee |
| Description | This process allow committee to view, approve application and modify their own club |
| Input Data Flows | Access |
| Output Data Flows | Permission |
| Process | PROGRAM ManageClub:  IF Club button is clicked, get the specific club ID, display the club details.  ELSE IF EditClub button is clicked, get the specific club ID, edit the club details.  ELSE IF ClubMembers button is clicked, get the specific club ID, display all the students’ details.  ELSE IF EditStudent button is clicked, get the specific student ID, edit the student detail.  ELSE IF DeleteStudent button is clicked, get the specific student ID, delete the student detail.  ELSE IF JoinClubApplication button is clicked, get the specific application ID, display all the applicant’ details.  ELSE IF ApproveStudent button is clicked, get the specific application ID, approve student become our club member.  ELSE IF DeclineStudent button is clicked, get the specific application ID, decline student to join our club member.  ELSE IF ClubEvent button is clicked, get the specific event ID, display all the events’ details.  ELSE IF PostEvent button is clicked, display the post event form.  ELSE IF EditEvent button is clicked, get the specific event ID, edit the event detail.  ELSE IF DeleteEvent button is clicked, get the specific event ID, delete the event detail.  END. |

**Data Flows**

|  |  |
| --- | --- |
| Name | Access |
| Description | Student who has a role committee can access into committee features |
| Origin/Source | Student |
| Destination/Sink | Process 7.0 Committee |
| Data Structure | SID, Student ID, Username, Password, First Name, Last Name, Intake, Contact Info, Email, IC/Passport, Gender, Birthday, Country, Club ID and Role |

|  |  |
| --- | --- |
| Name | Permission |
| Description | Student with committee role has the permission to access committee features |
| Origin/Source | Process 7.0 Committee |
| Destination/Sink | Student |
| Data Structure | Student ID, Username, Password, First Name, Last Name, Intake, Contact Info, Email, IC/Passport, Gender, Birthday, Country, Club ID and Role |

|  |  |
| --- | --- |
| Name | Validate Access |
| Description | Validate the student who has the committee role |
| Origin/Source | D1 Account database |
| Destination/Sink | Process 7.0 Committee |
| Data Structure | SID, Student ID, Username, Password, First Name, Last Name, Intake, Contact Info, Email, IC/Passport, Gender, Birthday, Country, Club ID and Role |

|  |  |
| --- | --- |
| Name | Status |
| Description | Display all data that related to their own club |
| Origin/Source | Process 7.0 Committee |
| Destination/Sink | Club Manager |
| Data Structure | SID, Student ID, Username, Password, First Name, Last Name, Intake, Contact Info, Email, IC/Passport, Gender, Birthday, Country, Club ID and Role |

**Data Store**

|  |  |
| --- | --- |
| Name: | D1 Account Database |
| Description: | Stores all student details, it also verifies roles while providing updated student details. |
| Input Data Flows: | Validate access |
| Output Data Flows: | Validate access |
| Data Structure: | SID, Student ID, Username, Password, First Name, Last Name, Intake, Contact Info, Email, IC/Passport, Gender, Birthday, Country, Club ID and Role |

**Tee Chor Yang TP061339**

Data Dictionary (Individual) – Register

**External Entity**

|  |  |
| --- | --- |
| Name | Guest |
| Description | Guest register as a student by filling up the information in the form. |
| Input Data Flows | Register form notice |
| Output Data Flows | Register Details |

**Process**

|  |  |
| --- | --- |
| Name | 1.0 Register |
| Description | This process require guest to enter their information. |
| Input Data Flows | Register Details |
| Output Data Flows | Register form notice |
| Process | PROGRAM Register:  IF Register button is clicked, display the registration from.  ELSE IF Login button is clicked. Redirect to Login page.  ELSE IF Submit button is clicked, get user input, user did not enter all the required information. Display instruction for user to fill out specifies input field.  ELSE IF Submit button is clicked, get user input, password do not match with confirm password. Display instruction for user to try again.  ELSE IF Submit button is clicked, get user input, username/ email/ student id exists in database. Display instruction for user to try again.  ELSE IF Submit button is clicked, get user input, user enter all required input field, password match with confirm password, username/ email/ student id does no exists in database. Redirect to Sign Up Success page.  END. |

**Data Flows**

|  |  |
| --- | --- |
| Name | Register Details |
| Description | Users need to enter all the required information in the form, in order to create an account. |
| Origin/Source | Guest |
| Destination/Sink | Process 1.0 Register |
| Data Structure | Username, Password, Confirm Password, First Name, Last Name, Student ID, Intake, Contact Info, Email, IC/Passport, Gender, Birthday, Country, Club ID and Role |

|  |  |
| --- | --- |
| Name | Register form notice |
| Description | After user complete the registration, system will display register successfully. |
| Origin/Source | Process 1.0 Register |
| Destination/Sink | Guest |
| Data Structure | Username, Password, Confirm Password, First Name, Last Name, Student ID, Intake, Contact Info, Email, IC/Passport, Gender, Birthday, Country, Club ID and Role |

|  |  |
| --- | --- |
| Name | Send register details |
| Description | Send user information to database, in order to store it. |
| Origin/Source | Process 1.0 Register |
| Destination/Sink | D1 Account database |
| Data Structure | Username, Password, Confirm Password, First Name, Last Name, Student ID, Intake, Contact Info, Email, IC/Passport, Gender, Birthday, Country, Club ID and Role |

**Data Store**

|  |  |
| --- | --- |
| Name: | D1 Account Database |
| Description: | Stores all user details, after user successfully register an account. |
| Input Data Flows: | Send Register Details |
| Output Data Flows: | - |
| Data Structure: | Username, Password, First Name, Last Name, Student ID, Intake, Contact Info, Email, IC/Passport, Gender, Birthday, Country, Club ID and Role |

**Teoh Yee Chin TP061586**

Data Dictionary (Individual) – Login

**External Entity**

|  |  |
| --- | --- |
| **Name:** | Guest |
| **Description:** | Guest is able to login to become a student account |
| **Input Data Flows:** | Guest home page |
| **Output Data Flows:** | Login Detail |

|  |  |
| --- | --- |
| **Name:** | Student |
| **Description:** | Student can update status to database |
| **Input Data Flows:** | Login Successful |
| **Output Data Flows:** | Update Status |

**Process**

|  |  |
| --- | --- |
| **Name:** | 2.0 Login |
| **Description:** | Guest login becomes a student account to access more feature |
| **Input Data Flows:** | Update Status |
| **Output Data Flows:** | Login Success |
| **Process:** | PROGRAM Login:  IF Login button is Clicked.  THEN Login to student account.  ELSE Return to Login page.  ENDIF;  END |

**Data Flow**

|  |  |
| --- | --- |
| **Name:** | Login Success |
| **Description:** | Allow student access more feature |
| **Origin/Source:** | Process 2.0 Login |
| **Destination/Sink:** | Student |
| **Data Structure:** | Username, Password |

|  |  |
| --- | --- |
| **Name:** | Update Status |
| **Description:** | Check or update validation with database |
| **Origin/Source:** | Student |
| **Destination/Sink:** | Process 2.0 Login |
| **Data Structure:** | Username, Password |

|  |  |
| --- | --- |
| **Name:** | Login Details |
| **Description:** | Guest can input the login detail to become as student |
| **Origin/Source:** | Guest |
| **Destination/Sink:** | Process 2.0 Login |
| **Data Structure:** | Username, Password |

|  |  |
| --- | --- |
| **Name:** | Login Access |
| **Description:** | Those who have an account can access more features |
| **Origin/Source:** | Process 2.0 Login |
| **Destination/Sink:** | Guest |
| **Data Structure:** | Username, Password |

|  |  |
| --- | --- |
| **Name:** | Send Login Details |
| **Description:** | Send user detail to database |
| **Origin/Source:** | Process 2.0 Login |
| **Destination/Sink:** | D1 Account database |
| **Data Structure:** | Username, Password |

|  |  |
| --- | --- |
| **Name:** | Validate Login Details |
| **Description:** | Validate user details who has an account |
| **Origin/Source:** | D1 Account database |
| **Destination/Sink:** | Process 2.0 Login |
| **Data Structure:** | Username, Password |

**Data Store**

|  |  |
| --- | --- |
| **Name:** | Account Database |
| **Description:** | Check login details and various user details to help in validating roles |
| **Input Data Flows:** | Validate Login Details |
| **Output Data Flows:** | Send Login Details |
| **Data Structure:** | Username, Password |

**Thaddeus Then Ze Qin (TP060326)**

Data Dictionary (Individual) – Organizer

**External Entity**

|  |  |
| --- | --- |
| Name | Student |
| Description | The student will need to get authorization to access the organizer role. |
| Input data flows | Permission |
| Output data flows | Access |

**Process**

|  |  |
| --- | --- |
| Name | 8.0 Organizer |
| Description | To get authorization as a organizer |
| Input data flows | Access |
| Output data flows | Permission |
| Process | PROGRAM Organizer:  Connect to student database.  Read student database.  If the username read from the database matched the fullname  entered by the user, then it will print out the organizer button.  END. |

**Data Flow**

|  |  |
| --- | --- |
| Name | Permission |
| Description | To give permission to the student to access the organizer role. |
| Origin / source | Organizer process |
| Destination / Sink | Student external entity |
| Data structure | Username, role |

**Data Store**

|  |  |
| --- | --- |
| Name | Account database |
| Description | Stores existing account details and to validate which role they belong to. |
| Input data flows | Validate access |
| Output data flows | Validate access |
| Data structure | Student\_ID, username, email, password, First name, Last name, intake, mobile number, gender, birth date, IC/Passport, country, role and club. |

Data Dictionary (Individual) – Browse

**External Entity**

|  |  |
| --- | --- |
| Name | Student |
| Description | Student would like to browse the website. |
| Input data flows | Navigate |
| Output data flows | Navigate |

**Process**

|  |  |
| --- | --- |
| Name | 5.0 Browse |
| Description | The student can see what can be browse in the website. |
| Input data flows | Navigate |
| Output data flows | Navigate |
| Process | PROGRAM Menu:  Display home button.  If the user clicks the home button, the program will call  index.php.  Display home button.  If the user clicks about us, the program will call about.php.  Display club and society button.  If the user clicks club and society, the program will call  club.php.  Display events button  If the user clicks the home button, the program will call  events.php.  Display contact us button  If the user clicks the home button, the program will call  contact.php.  END. |

**Data Flow**

|  |  |
| --- | --- |
| Name | Navigate |
| Description | To display the web page that the student navigates to. |
| Origin / source | Browse process |
| Destination / Sink | Student external entity |
| Data structure | - |

**Data Store**

|  |  |
| --- | --- |
| Name | - |
| Description | - |
| Input data flows | - |
| Output data flows | - |
| Data structure | - |

# Entity Relationship Diagram

Map

Description automatically generated

# Pseudocode or Flowcharts – Individual

**Lee Jia Qian TP061863**

**Administrator Modules/Page**

Diagram

Description automatically generated**Manage user (admin)**

Diagram

Description automatically generated

**Manage Club (admin)**

Diagram

Description automatically generated

**Manage Message (admin)**

Diagram

Description automatically generated

**Manage Report (admin)**

Diagram

Description automatically generated

**Wong Yew Wei TP060747**

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

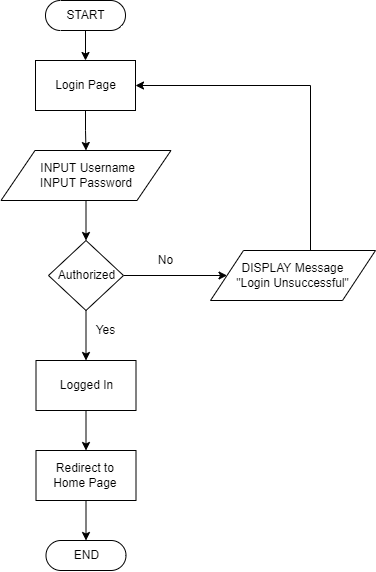
Description automatically generated

Diagram

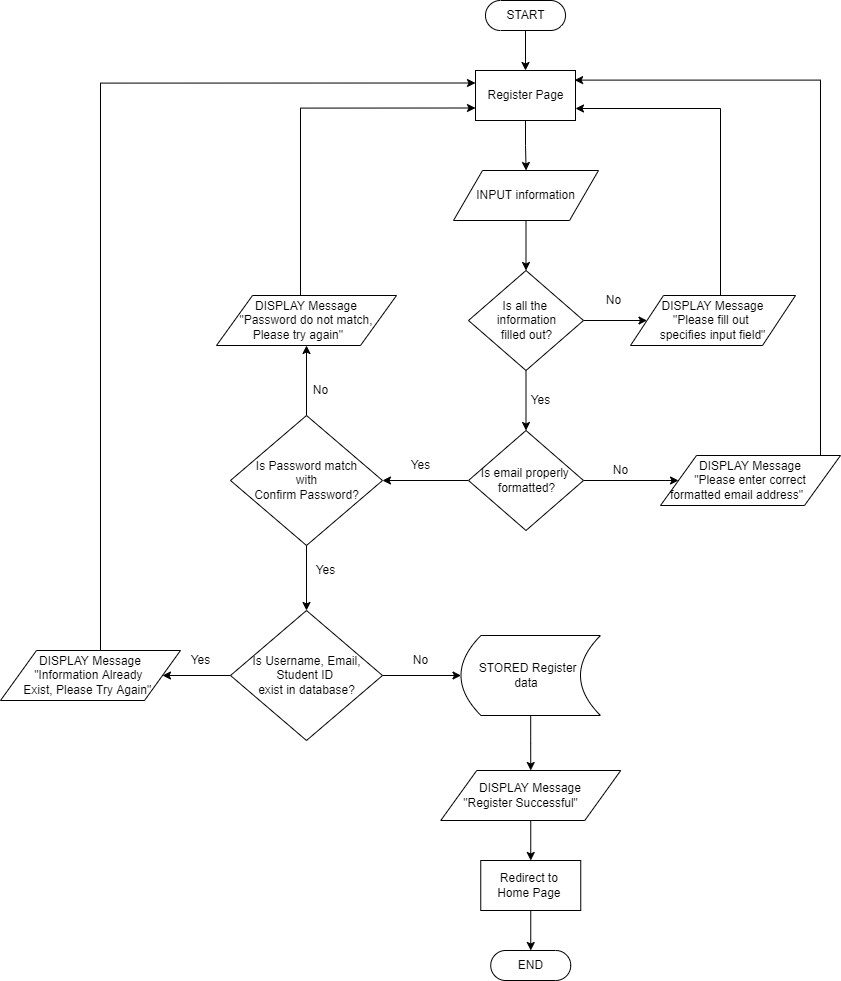
Description automatically generated

**Tee Chor Yang TP061339**

**Login**



**Register**



**Teoh Yee Chin TP061586**

|  |  |
| --- | --- |
| Student (Logout) | Student (Apply to Join Club) |
| Diagram  Description automatically generated |  |

|  |  |
| --- | --- |
| **Club & Society Organizer**  **(Make Announcement)** | **Club & Society Organizer**  **(Post Event)** |
|  |  |

|  |
| --- |
| **Club & Society Organizer**  **(Modify Event Information)** |
| Diagram  Description automatically generated |

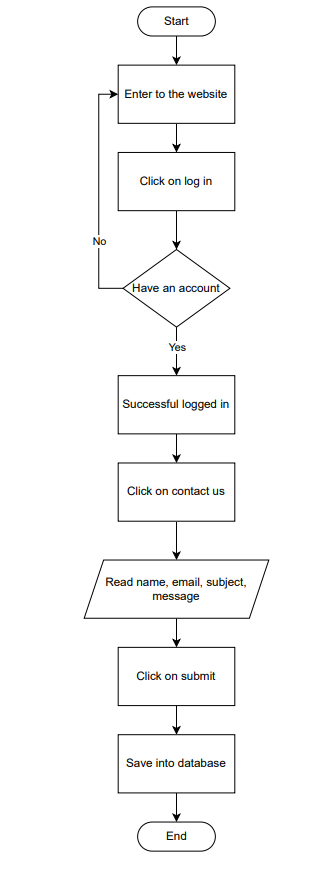
**Thaddeus Then Ze Qin TP060326**

**View personal profile**

**Diagram

Description automatically generated**

**Customer service**



**Browse**

**Diagram

Description automatically generated**

Screens Design and User Manual

**Home page**

When users visit our website, the users will be greeted by a home page where they are able to learn what our website offers. At the homepage, users are able to navigate around to explore the clubs and society that we have, learn more about our website, the events that is ongoing, and a help desk where they can contact us. However, there are also buttons where the users can click if they don’t want to go through that hassle to go to a specific place.

Graphical user interface

Description automatically generated (Figure 1: Home page)

**About us**

The about us page provides all the information that user needs to know about our Radiant club and society website for example what our website provide and what our goals are.

Graphical user interface, text, website

Description automatically generated

(Figure 2: About us)

**Club and society**

At the club and society page, we show all the clubs and society that currently have. There will be a “learn more” button for each club and society which will give more information. If the user wish to search for a specific club or society, user just have to hover to the club and society tab and a drop down list will appear, show all the specific category for each club and society.

Graphical user interface, text, application

Description automatically generated

(Figure 3: Club and society)

**Badminton club**

Once the user clicks on the “learn more” button for example for the badminton club ones, then the website will load into to badminton club page where it will give more information about the club or society. If the user found the club interesting, they can click on the “join us” button to join the club. In this page, it also shows the event that the club or society has and where they can contact us if they got any enquiry about the club.

Graphical user interface, website

Description automatically generated

(Figure 4: Badminton club)

**Application form**

If the user wants to join a club or society, they have to log in first. Once the user clicks the “join us’ button at the club or society page, the user will be brought to an application form. At the application form all the personal information is keyed in automatically because it is already saved in the database. However, if the user wishes to add more information about themselves, At the end of the page there is a personal statement box where they can fill in. In this page, it also shows the club information to make sure the user knows which club they are signing up for.

Graphical user interface

Description automatically generated

(Figure 5: Application form)

**Event**

The events page shows all the events that we have. If the user wishes to know more about the event, the user can click on the “learn more” button.

Graphical user interface, application

Description automatically generated

(Figure 6: Event)

**Learn more: event**

When the user clicks on the “learn more” button for one of the event, the website will load into a page where it shows more information and announcement about the event.

Graphical user interface, text, website

Description automatically generated

(Figure 7: BAT X APU WEEK: Ideathon event)

**Contact us**

At the contact us page, it shows various ways to get in touch with us. If the user has more question that would like to ask us, they can send us a message at this page.

Graphical user interface, text, email

Description automatically generated

(Figure 8: contact us)

**Edit profile**

For the users to edit their profile, they need to click their own name at the top right. After clicking their name, there will be an edit profile option. Clicking edit profile will bring the user to the page shown at Figure 9 where they can change their password. If they wish to change their personal information, they would need to contact us to change it for them.

Graphical user interface, table

Description automatically generated

(Figure 9: Edit profile)

**Log in**

Users have to click the “log in” button at the home page to log into their account. At the log in page, the users are asked to fill in their username and password. Once the user fills in the required information and click “log in”, the website will check the database if there is an existing user. If the username and password does not match any of the account in our database, the user is not allowed to log in. At the log in page, user who forgot their password are able to do so. User who are new to the website, they is an option for them to sign up.

Graphical user interface, application, Word

Description automatically generated

(Figure 10: Log in)

**Forgot password**

At the forgot password page, the user is required to enter the email that they wish to receive an email from us to reset their password.

Graphical user interface, text, application, email

Description automatically generated

(Figure 11: Forgot password)

**Sign up**

At the sign-up page, users are required to fill in all their personal information that is required to create their account. However, if the user already has an existing account, there is a “log in” button where the user can click on to log in.

Graphical user interface, application

Description automatically generated

(Figure 12: Sign up)

**Organizer: event**

The organizer event page only can be access by the organizer where he or she is able to post event, post announcement, edit and delete the event. To get to the organizer page, the user has to log in an account where the role is assigned as an organizer. After logging in as an organizer, at the home page the user has to click their name at the top right and click at the organizer option.

A screenshot of a computer

Description automatically generated

(Figure 13: Organizer event page)

**Organizer: Post event**

To post event, the organizer must fill in all the event details. The club information is also shown in this page. At the right side, the of the page, there are tips that the organizer can use when filling up the event details.

Graphical user interface, application

Description automatically generated

(Figure 14: Post event)

**Organizer: Post announcement**

At the post announcement page, it will show the information about the event and the organizer is able to enter the announcement information at the announcement box.

Graphical user interface, application

Description automatically generated

(Figure 15: Post announcement)

**Organizer: Edit event**

When the organizer clicks on the “edit” button, he or she will be brought to this page. The organizer is only allowed to edit the announcement, event title and description of the event.

Graphical user interface, text, application

Description automatically generated

(Figure 16: event edit page)

**Organizer: Delete**

By clicking the delete button, the event will be deleted from the database and the even will be remove from the event page.

Text

Description automatically generated

(Figure 17: Delete)

**Committee page**

The committee page only can be access by the committee where he or she is able to view the club, club member, join request and club events. To get to the committee page, the user has to log in an account where the role is assigned as a committee. After logging in as a committee, at the home page the user has to click their name at the top right and click at the committee option.

A screenshot of a computer

Description automatically generated

(Figure 18: Committee)

**Committee: Club**

At the club page, the committee is able to see all the information about the club that is retrieve from the database and is also able to edit the information of the club.

A screenshot of a computer

Description automatically generated

(Figure 19: Club)

**Committee: Edit club**

At the edit club page, the committee is able to change the club name, category, description, link, email, venue and location.

Graphical user interface, text, application

Description automatically generated

(Figure 20: Edit club)

**Committee: Club member**

At the club member page, the committee can see all the club members with the ID 10. The committee can also edit or remove the club members.

A screenshot of a computer

Description automatically generated

(Figure 21: Club member)

**Committee: Edit club member**

At the edit club member page, the committee is only allowed to change the member role. Other information, the committee will not be able to edit.

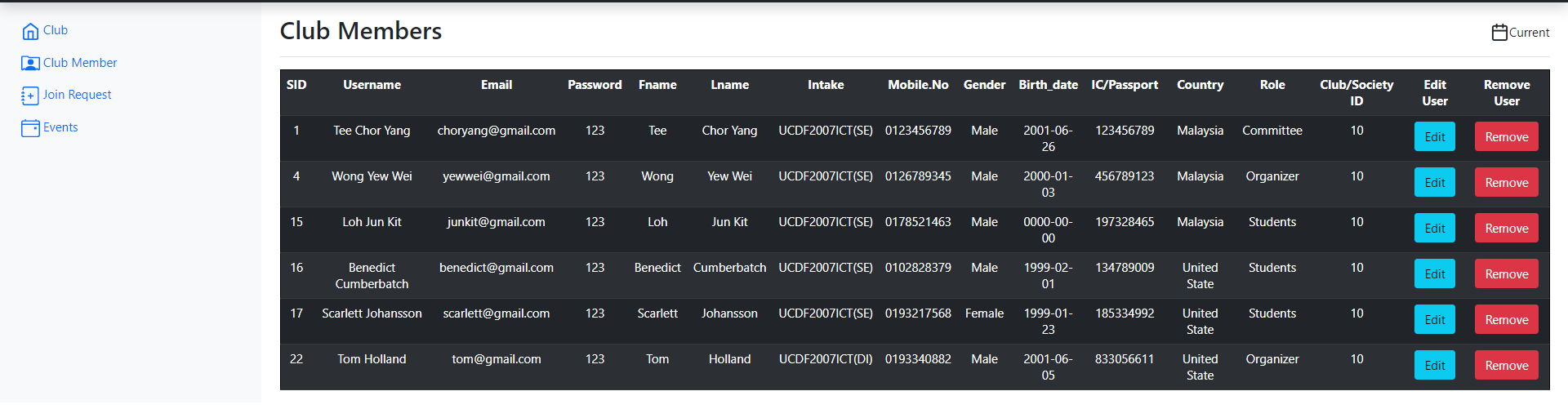
Graphical user interface, table

Description automatically generated

(Figure 22: Edit club member)

**Committee: Remove club member**

At the club member category, the committee is allowed to delete a club member. Once the club member is removed, the data of the club member is erased from the database.



(Figure 23: Delete club member)

**Committee: Join request**

At the join request page, the committee is able to see all the application form that is requested from the user that would like to join the club where the committee can approve or decline the request.

A screenshot of a computer

Description automatically generated

(Figure 24: Join request)

**Committee: Events**

At the event category, similar to the organizer, the committee can see all the club event that is retrieve from the database, post event, post announcement for the specific event, edit event and delete event.

A screenshot of a computer

Description automatically generated

(Figure 25: committee event)

**Admin: Dashboard**

Similar to how previously the organizer and committee got into their respective page, the admin has to log in with the correct credentials and click the admin option to get to the admin exclusive page. Once the admin reaches the admin page, he or she will be greeted with the dashboard where it will show the total student, student with club, student without club, club pending request, total club, total ongoing event, total announcement and total feedback of the website.

A screenshot of a computer

Description automatically generated

(Figure 26: Dashboard)

**Admin: User**

For the admin, the admin has full authorization to add new user, edit the user information and delete the user from the database.

A screenshot of a computer

Description automatically generated

(Figure 27: admin’s user page)

**Admin: Add new user**

At the add new user page, the admin is able to fill in all the information that is required to sign up for the new user.

Graphical user interface, application

Description automatically generated

(Figure 28: Add new user)

**Admin: Edit user**

For the edit user page, the admin is able to change the first name, student ID, contact information, password, last name, intake, email, gender and role of the user.

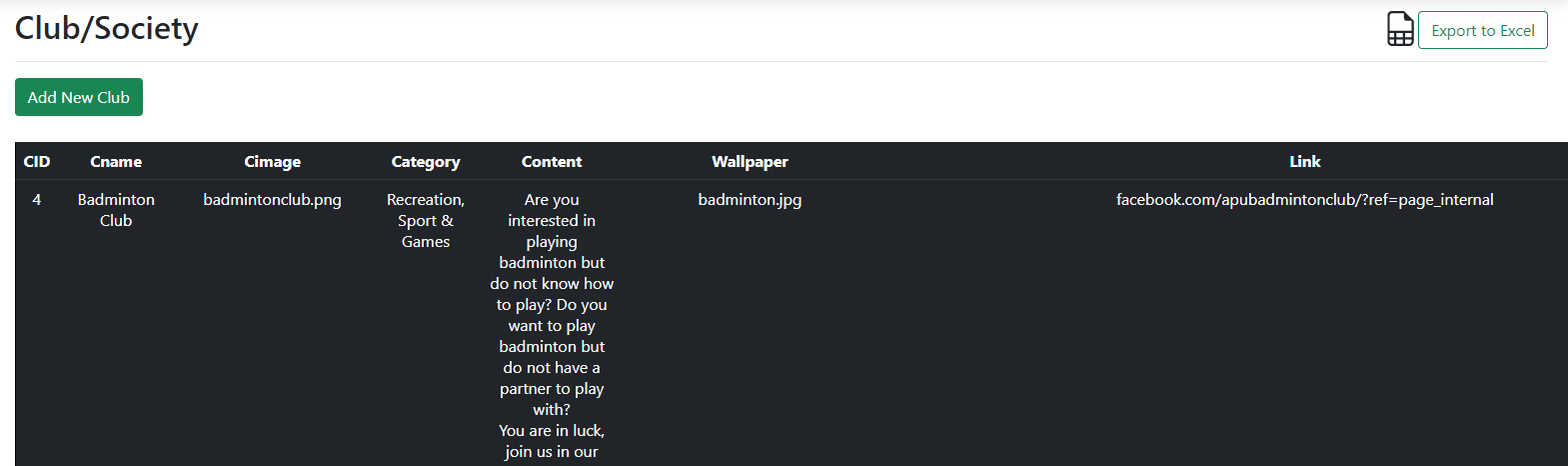
Graphical user interface, table

Description automatically generated

(Figure 29: Edit user)

**Admin: Club and society**

At this page, the admin is able to see all the club and society that has been registered into the website. The admin also can add new club and society, edit or delete them.



(Figure 30: Club and society)

**Admin: Add new club or society**

At this page, the admin has the ability to add new club or society into the website. The admin has to insert all the required information such as the club name, club logo, category and so on in order for the new club or society to be posted at the website.

Graphical user interface, application

Description automatically generated

(Figure 31: Add new club or society)

**Admin: Edit club or society**

At this page, the admin is able to edit existing club or society information except for the club or society ID which has been locked.

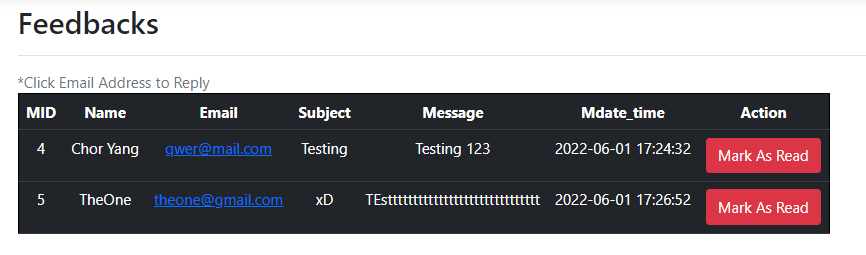
Graphical user interface

Description automatically generated

(Figure 32: Edit club or society)

**Admin: Feedbacks**

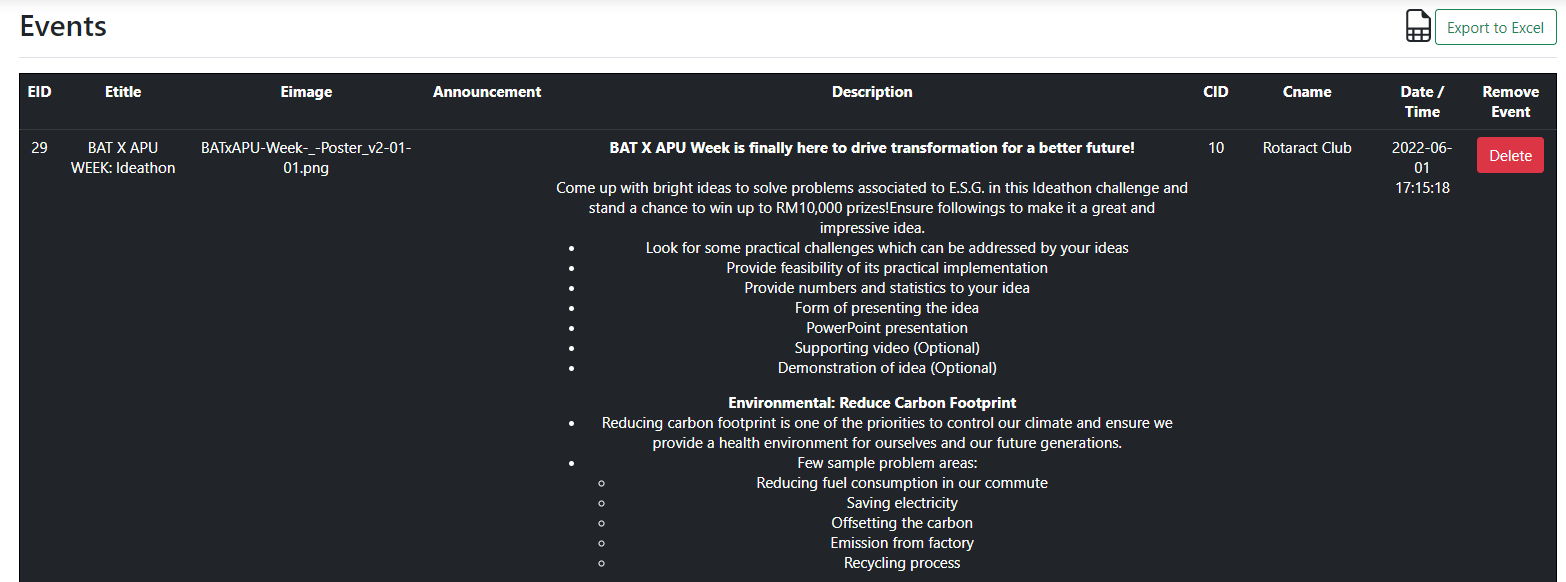
At the feedback page, the admin can view all the feedback that the user has given. After reading the message, the admin can mark as read and the feedback will be removed from the feedback. If the admin reply to any of the messages, then the user have to check their email for the respond.



(Figure 33: Feedbacks)

**Admin: Events**

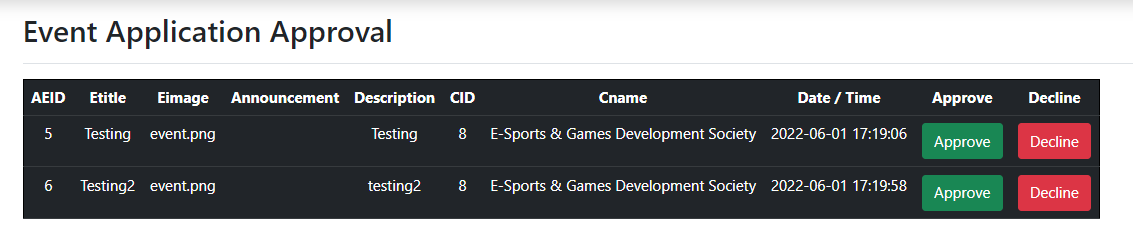
At the event category for the admin page, the admin can view all the existing event information that is posted at the website. Other than that, the admin can also delete existing events that has been posted on the website.



(Figure 34: Admin’s event)

**Admin: Approve events**

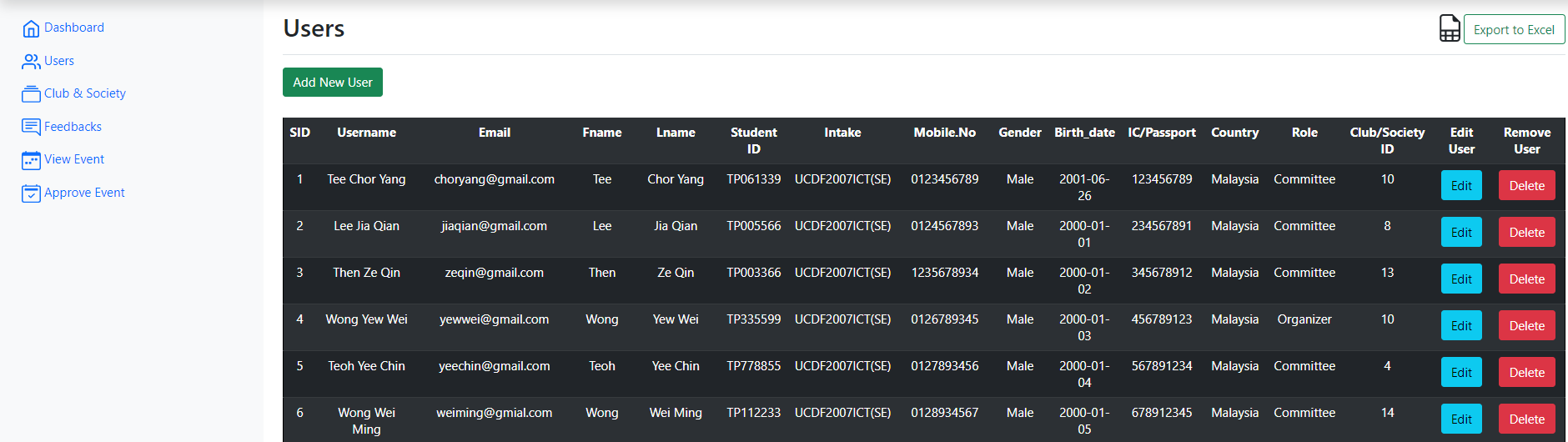
At the approve event category, the admin is able to view all the proposed event application and decide whether he or she want to approve or decline the request.



(Figure 35: Approve events)

# Report Designs and User Manual

Only the admin has this function where he or she can export the data as shown in figure 36, by clicking the “export to excel” button at the top right corner of the page. The “export to excel” also applies the same to the club and society, feedbacks, view events, and approve events.



(Figure 36: user’s data export to excel)

After exporting the data into excel, we can create a graph to show the total number of users that has the role of committee, organizer, student, and admin that is currently existing in Radiant clubs and society database. Having the chart helps the admin analyze easier in the future so that they can keep track.

(Figure 37: User role)

# Test Plan

## Unit Testing

**Students**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Test Unit** | **Test Procedures** | **Test Data** | **Expected result** | **Actual Result** | **Result** | **Re-**  **mark** |
| 1 | User Login | Insert Username and Password.  Click “Login” button. | Correct Input Data  Username: Teoh Yee Chin  Password:  123 | Successfully Login into system. | Login Successful. | Success | - |
|  | User Login | Insert Username and Password.  Click “Login” button. | Incorrect Input Data  Username:  Teoh Yee Chin  Password:  1234 | Unable Login into system and show an error message “User or Password Incorrect, Login Unsuccessful, Pls Try Again” | Login Failed and error message will display. | Success | - |
| 2 | User view personal profile | Click on Username  Click Edit Profile | System Show:  Username  Email  Mobile Number  ID  First Name  Last Name  Student ID  Intake  IC/Passport  Gender  Birthday  Country  Club  Committee | System shows all detail of the user | Display details of the user | Success | - |
| 3 | Users modify personal profile | Click on Username and Edit Profile  Change Password  Click “Update Password” | System Show:  Current Password:  123  New Password:  1234  Confirm Password:  1234 | System will update the password and show a message “Password Update Successfully” | Update Successful | Success | - |
| 4 | Send message to Admin | Click “Contact Us”  Fill up the form and click “Submit” | System Show:  Name:  Mc Uncle  Email:  mcd@gmail.com  Subject:  testing  Message:  testing | System will save it and show a message “Submitted Successfully!  We will respond you through email, it might will take 1-3 working days.” | Submit Successful. | Success | - |
| 5 | Apply to join a club | Double Click” Club & Society”  Choose a club user want to join and click “Learn More”  Click “Join Us”  Write a personal statement and click “Submit” | System Show:  Username  Email  Contact Number  First Name  Last Name  Student ID  Intake  Gender  Birthday  Country  Club Name  Club Type  Club Email  Personal Statement | System will save it and show a message “Application Form Submitted Successfully!” | Submit Successful. | Success | - |

**Guest**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Test Unit** | **Test Procedures** | **Test Data** | **Expected result** | **Actual Result** | **Result** | **Re-**  **mark** |
| 1 | User Sign Up | Click “Login” and “Sign Up”  Fill up the form  Click “Sign Up” button | Correct Input Data  Username:  Chen Chick  Email: chen@gmail.com  Password: 123  Confirm Password: 123  First Name: Chen  Last Name: Chick  Identity or Passport Number: 123456789  Student ID: TP012345  Intake: UCDF2007ICT(SE)  Mobile: Number:0123456789  Gender: Other  Birth Date:01/01/1971  Country: South Africa | Successfully Sign Up a new account. | Sign Up Successfully | Success | - |
|  | User Sign Up | Click “Login” and “Sign Up”  Fill up the form  Click “Sign Up” button | Incorrect Input Data  Username:  Chen Chick  Email: chen.gmail.com  Password: 123  Confirm Password: 123  First Name: Chen  Last Name: Chick  Identity or Passport Number: 123456789  Student ID: TP012345  Intake: UCDF2007ICT(SE)  Mobile: Number:0123456789  Gender: Other  Birth Date:  01/01/1971  Country: South Africa | Unable to Sign Up a account and show a message at email column  “Please include an ‘@’ in the email address, ‘chen.gmail.com’ is missing an ‘@’ ”. | Sign Up Failed and error message will display. | Success | - |
| 2 | Send message to Admin | Click “Contact Us”  Fill up the form and click “Submit” | System Show:  Name:  Mc Uncle  Email:  mcd@gmail.com  Subject:  testing  Message:  testing | System will save it and show a message “Submitted Successfully!  We will respond you through email, it might will take 1-3 working days.” | Submit Successful. | Success | - |
| 3 | Login as students after signup an account | Click “Login”  Fill up the column  Click “Login” | System Show:  Username  Password | Log in as a student account | Login Successful | Success | - |
| 4 | Browse website as guest | Click Home, About Us, Club &Society, Event page | Explore more information as a guest but can’t join any club before becoming a student | When the guest clicks the “Join Us” in the “Club &Society” page it will show a message “Sorry You Are Not Allowed to Access This Page. To access this page, you must be logged in. Please try again after you LOG IN” | The system will take the guest to login page. | Success | - |

**Organizer**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Test Unit** | **Test Procedures** | **Test Data** | **Expected result** | **Actual Result** | **Result** | **Re-**  **mark** |
| 1 | View Event Information | Click username and choose “organizer” | System will show all the event detail about that club.  (For example: badminton club) | Show all event detail about badminton club | Display all event detail about badminton club | Success | - |
| 2 | Edit Event Detail | Click username and choose “organizer”  Click “Edit” and change the detail about the event  Click “Submit” | System Show:  Event ID  Event Title  Announcement  Description  Club ID  Club Name  Date / Time | The event detail will be update successfully. | Update successfully | Success | - |
| 3 | Delete Event | Click username and choose “organizer”  Click “Delete” |  | System will directly delete all the event detail | Delete successfully | Success | - |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | Post Announcement | Click username and choose “organizer”  Click “Post”  Fill up the form  Click “Post” | System Show:  Event ID  Event Title  Current Announcement  Announcement | Announcement posted successful and system will show a message “Announcement Posted” | Posted Successful | Success | - |
| 5 | Post Event | Click username and choose “Organizer”  Click “Post Event”  Fill up the form  Click “Post Event” | System Show:  Event Title  Event Image  Event Description  Announcement | Event posted successful and system will show a message “Event Posted Successfully”  When the admin approves the event will be post | Wait for the admin approves the event and post it out | Success | - |

**Committee**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Test Unit** | **Test Procedures** | **Test Data** | **Expected result** | **Actual Result** | **Result** | **Re-**  **mark** |
| 1 | View Club Information | Click username  Choose “committee” | System will show all the club detail. | Display every club detail. | Display every club detail. | Success | - |
| 2 | Modify Club Detail | Click username  Choose “committee”  Click “Club” and “Edit”  After change information click “Submit” | System Show:  Club ID  Club Name  Description  Link  Email  Venue  Location | System updates successful the club detail. | Update Successfully | Success | - |
| 3 | View Club Member | Click username  Choose “committee”  Click “Club Member” | System will show all the club member and member detail. | Display every club member and member detail. | Display every club member and member detail. | Success | - |
| 4 | Modify Club Member | Click username  Choose “committee”  Click “Club Member” and click “Edit”  After change role click “Submit” | System Show:  User ID  Username  Password  First Name  Last Name  Student ID  Intake  Contact Info  Email  IC/Passport  Gender  Birthday  Country  Club ID  Role | Committee change the member role and update successful | Update Successfully | Success | - |
| 5 | Remove Club Member | Click username  Choose “committee”  Click “Club Member” and click “Remove” |  | System will directly remove the member, the user does not belong to any club member, and the user is just a normal student. | Remove successful | Success | - |
| 6 | Approve user become a member | Click username  Choose “committee”  Click “Join Request” and click “Approve” |  | The students became a club member | Student became a club member | Success | - |
| 7 | Decline user become a member | Click username  Choose “committee”  Click “Join Request” and click “Decline” |  | System will directly remove and delete the request; the user is still a normal student. | Decline successful | Success | - |
| 8 | Edit Event Detail | Click username and choose “committee”  Click “Edit” and change the detail about the event  Click “Submit” | System Show:  Event ID  Event Title  Announcement  Description  Club ID  Club Name  Date / Time | The event detail will be update successfully. | Update successfully | Success | - |
| 9 | Delete Event | Click username and choose “committee”  Click “Delete” |  | System will directly delete all the event detail | Delete successfully | Success | - |
| 10 | Post Announcement | Click username and choose “committee”  Click “Post”  Fill up the form  Click “Post” | System Show:  Event ID  Event Title  Current Announcement  Announcement | Announcement posted successful and system will show a message “Announcement Posted” | Posted Successful | Success | - |
| 11 | Post Event | Click username and choose “committee”  Click “Post Event”  Fill up the form  Click “Post Event” | System Show:  Event Title  Event Image  Event Description  Announcement | Event posted successful and system will show a message “Event Posted Successfully”  When the admin approves the event will be post | Wait for the admin approves the event and post it out | Success | - |

**Admin**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Test Unit** | **Test Procedures** | **Test Data** | **Expected result** | **Actual Result** | **Result** | **Re-**  **mark** |
| 1 | View all user detail | Click username  Choose “Admin”  Click “User” | System will show all the user detail. | Display every user detail. | Display every user detail. | Success | - |
| 2 | Modify User Detail | Click username  Choose “Admin”  Click “User” and click “Edit”  Click “Submit” | System Show:  User ID  Username  Password  First Name  Last Name  Student ID  Intake  Contact Info  Email  IC/Passport  Gender  Birthday  Country  Club ID  Role | System updates the user detail successfully | System updates successful | Success | - |
| 3 | Delete User | Click username  Choose “Admin”  Click “User” and click “Delete” |  | System will delete all the user detail or information. | Delete successful | Success | - |
| 4 | View Club & Society Detail | Click username  Choose “Admin”  Click “Club & Society” | System will show all the Club & Society detail. | Display every Club & Society detail. | System displays every Club & Society detail. | Success | - |
| 5 | Modify Club & Society Detail | Click username  Choose “Admin”  Click “Club & Society” and click “Edit”  Click “Submit” | System Show:  ID  Club Name  Category  Content  Link  Email  Venue  Location | System updates the Club & Society detail successfully | System updates successful | Success | - |
| 6 | Delete Club & Society | Click username  Choose “Admin”  Click “Club & Society” and click “Delete” |  | System will delete all the Club & Society detail or information. | Delete successful | Success | - |
| 7 | Add Club | Click username  Choose “Admin”  Click “Club & Society” and click “Add Club”  Fill up the Form and click “Submit” | System Show:  Club Name  Club Logo  Category  Club Description  Club Wallpaper  Club Website Link  Club Email  Club Venue  Club Location | System will create the club or society in the list. | Create Successful | Success | - |
| 8 | View Event | Click username  Choose “Admin”  Click “View Event” | System will show all the event detail. | Display every event detail. | System displays every event detail. | Success | - |
| 9 | Delete Event | Click username and choose “Admin”  Choose “View Event” and click “Delete” |  | System will directly delete all the event detail | Delete successfully | Success | - |
| 10 | View User Feedback | Click username and choose “Admin”  Choose “Feedback” | System will show all the user feedback. | Display every user feedback. | Display every user feedback. | Success | - |
| 11 | Delete the Feedback | Click username  Choose “Admin”  Click “Feedbacks” |  | System should be deleting the feedback | Delete Successful | Success | - |
| 12 | View the webpage report | Click username and choose “Admin”  Choose “Dashboard” | System will show all the data of the webpage. | Display every data of the webpage. | Display every data of the webpage. | Success | - |
| 13 | Approve Event Application | Click username and choose “Admin”  Click “Approve Event” and click “Approve” |  | The event will be posted out | System will  posted out the event | Success | - |
| 14 | Decline user become a member | Click username and choose “Admin”  Click “Approve Event” and click “Decline” |  | The event will be rejected | System will  Reject and delete the event application | Success | - |

## User Acceptance Testing

**Tester 1**

|  |  |
| --- | --- |
| **Tester Profile** | |
| Name: | Reshiwaran Jegatheswaran |
| Position: | Tutor/Lecturer |
| Email: | [TP038338@mail.apu.edu.my](mailto:TP038338@mail.apu.edu.my) |
| Date | 2 June 2022 |
| Time | 4:30 PM |

Thank you for taking part in the testing phase of the club and society's management system. This User Acceptance Testing (UAT) tester is required to go through the testing cases specified below that are within the club and society's management system. If you have any questions while testing, please contact the developer team. We appreciate your cooperation!

The following scale standardizes the rating for the testing phase below:

* 1 – Poor
* 2 – Moderate
* 3 – Satisfied
* 4 – Good
* 5 – Outstanding

|  |  |  |
| --- | --- | --- |
| **No** | **Testing Case** | **Rate 1 – 5 (From Poor to Outstanding)** |
| **Guest** | | |
| 1 | Register an Account | 1 2 3 4 5 |
| 2 | Login | 1 2 3 4 5 |
| 3 | Send Feedback or Message to Admin | 1 2 3 4 5 |
| 4 | Browse Home, About Us, Club &Society, Event page | 1 2 3 4 5 |
| 5 | Deny guest when try to access ‘Join Club’ | 1 2 3 4 5 |

|  |  |  |
| --- | --- | --- |
| **No** | **Testing Case** | **Rate 1 – 5 (From Poor to Outstanding)** |
| **Students** | | |
| 1 | View Personal Profile | 1 2 3 4 5 |
| 2 | Change Password | 1 2 3 4 5 |
| 3 | Send Feedback or Message to Admin | 1 2 3 4 5 |
| 4 | Browse Home, About Us, Club &Society, Event page | 1 2 3 4 5 |
| 5 | Apply to join a Club | 1 2 3 4 5 |
| 6 | Deny access to ‘Join Club’ when students already have a club | 1 2 3 4 5 |
| 7 | Log Out | 1 2 3 4 5 |

|  |  |  |
| --- | --- | --- |
| **No** | **Testing Case** | **Rate 1 – 5 (From Poor to Outstanding)** |
| **Organizer** | | |
| 1 | View Event Information | 1 2 3 4 5 |
| 2 | Modify Event Information | 1 2 3 4 5 |
| 3 | Post Announcement | 1 2 3 4 5 |
| 4 | Delete Event | 1 2 3 4 5 |
| 5 | Post Event | 1 2 3 4 5 |

|  |  |  |
| --- | --- | --- |
| **No** | **Testing Case** | **Rate 1 – 5 (From Poor to Outstanding)** |
| **Committee** | | |
| 1 | View Club Information | 1 2 3 4 5 |
| 2 | Modify Club Information | 1 2 3 4 5 |
| 3 | View Club Member | 1 2 3 4 5 |
| 4 | Assign Role to Club Member | 1 2 3 4 5 |
| 5 | Remove Club Member | 1 2 3 4 5 |
| 6 | Approve or Reject Join Club Application | 1 2 3 4 5 |
| 7 | Modify Event Information | 1 2 3 4 5 |
| 8 | Post Announcement | 1 2 3 4 5 |
| 9 | Delete Event | 1 2 3 4 5 |
| 10 | Post Event | 1 2 3 4 5 |

|  |  |  |
| --- | --- | --- |
| **No** | **Testing Case** | **Rate 1 – 5 (From Poor to Outstanding)** |
| **Admin** | | |
| 1 | View User Information | 1 2 3 4 5 |
| 2 | Modify User Information | 1 2 3 4 5 |
| 3 | Delete User | 1 2 3 4 5 |
| 4 | View Club & Society Information | 1 2 3 4 5 |
| 5 | Modify Club & Society Information | 1 2 3 4 5 |
| 6 | Delete Club & Society | 1 2 3 4 5 |
| 7 | Add Club | 1 2 3 4 5 |
| 8 | View Event | 1 2 3 4 5 |
| 9 | Delete Event | 1 2 3 4 5 |
| 10 | Check User Feedback | 1 2 3 4 5 |
| 11 | Delete Feedback | 1 2 3 4 5 |
| 12 | View the Report | 1 2 3 4 5 |
| 13 | Approval Event Application | 1 2 3 4 5 |
| 14 | Decline Event Application | 1 2 3 4 5 |

|  |  |
| --- | --- |
| **Tester** | **Comment** |
| Reshiwaran Jegatheswaran | All good! All function works as intended. The feels and looks of the page are responsive yet simple looking. |
| **Developer** | **Action** |
| Tee Chor Yang | We will continue to improve so that users can use the website smoothly and effectively. We will continue to strive for perfection! |

**Tester 2**

|  |  |
| --- | --- |
| **Tester Profile** | |
| Name: | Lee Bing Hui |
| Position: | Lee Jia Qian’s Family Member |
| Email: | leebinghui12@gmail.com |
| Date | 31 May 2022 |
| Time | 10:30 PM |

Thank you for taking part in the testing phase of the club and society's management system. This User Acceptance Testing (UAT) tester is required to go through the testing cases specified below that are within the club and society's management system. If you have any questions while testing, please contact the developer team. We appreciate your cooperation!

The following scale standardizes the rating for the testing phase below:

* 1 – Poor
* 2 – Moderate
* 3 – Satisfied
* 4 – Good
* 5 – Outstanding

|  |  |  |
| --- | --- | --- |
| **No** | **Testing Case** | **Rate 1 – 5 (From Poor to Outstanding)** |
| **Guest** | | |
| 1 | Register an Account | 1 2 3 4 5 |
| 2 | Login | 1 2 3 4 5 |
| 3 | Send Feedback or Message to Admin | 1 2 3 4 5 |
| 4 | Browse Home, About Us, Club &Society, Event page | 1 2 3 4 5 |
| 5 | Deny guest when try to access ‘Join Club’ | 1 2 3 4 5 |

|  |  |  |
| --- | --- | --- |
| **No** | **Testing Case** | **Rate 1 – 5 (From Poor to Outstanding)** |
| **Students** | | |
| 1 | View Personal Profile | 1 2 3 4 5 |
| 2 | Change Password | 1 2 3 4 5 |
| 3 | Send Feedback or Message to Admin | 1 2 3 4 5 |
| 4 | Browse Home, About Us, Club &Society, Event page | 1 2 3 4 5 |
| 5 | Apply to join a Club | 1 2 3 4 5 |
| 6 | Deny access to ‘Join Club’ when students already have a club | 1 2 3 4 5 |
| 7 | Log Out | 1 2 3 4 5 |

|  |  |  |
| --- | --- | --- |
| **No** | **Testing Case** | **Rate 1 – 5 (From Poor to Outstanding)** |
| **Organizer** | | |
| 1 | View Event Information | 1 2 3 4 5 |
| 2 | Modify Event Information | 1 2 3 4 5 |
| 3 | Post Announcement | 1 2 3 4 5 |
| 4 | Delete Event | 1 2 3 4 5 |
| 5 | Post Event | 1 2 3 4 5 |

|  |  |  |
| --- | --- | --- |
| **No** | **Testing Case** | **Rate 1 – 5 (From Poor to Outstanding)** |
| **Committee** | | |
| 1 | View Club Information | 1 2 3 4 5 |
| 2 | Modify Club Information | 1 2 3 4 5 |
| 3 | View Club Member | 1 2 3 4 5 |
| 4 | Assign Role to Club Member | 1 2 3 4 5 |
| 5 | Remove Club Member | 1 2 3 4 5 |
| 6 | Approve or Reject Join Club Application | 1 2 3 4 5 |
| 7 | Modify Event Information | 1 2 3 4 5 |
| 8 | Post Announcement | 1 2 3 4 5 |
| 9 | Delete Event | 1 2 3 4 5 |
| 10 | Post Event | 1 2 3 4 5 |

|  |  |  |
| --- | --- | --- |
| **No** | **Testing Case** | **Rate 1 – 5 (From Poor to Outstanding)** |
| **Admin** | | |
| 1 | View User Information | 1 2 3 4 5 |
| 2 | Modify User Information | 1 2 3 4 5 |
| 3 | Delete User | 1 2 3 4 5 |
| 4 | View Club & Society Information | 1 2 3 4 5 |
| 5 | Modify Club & Society Information | 1 2 3 4 5 |
| 6 | Delete Club & Society | 1 2 3 4 5 |
| 7 | Add Club | 1 2 3 4 5 |
| 8 | View Event | 1 2 3 4 5 |
| 9 | Delete Event | 1 2 3 4 5 |
| 10 | Check User Feedback | 1 2 3 4 5 |
| 11 | Delete Feedback | 1 2 3 4 5 |
| 12 | View the Report | 1 2 3 4 5 |
| 13 | Approval Event Application | 1 2 3 4 5 |
| 14 | Decline Event Application | 1 2 3 4 5 |

|  |  |
| --- | --- |
| **Tester** | **Comment** |
| Lee Bing Hui | Could add a little more creative looks in the managerial positions. Besides that, everything is fine. |
| **Developer** | **Action** |
| Lee Jia Qian | We will continue to improve on the looks of the overall managerial role interfaces to make it more attractive for the user while completing or navigating through the processes. |

**Tester 3**

|  |  |
| --- | --- |
| **Tester Profile** | |
| Name: | Yeap Xin Wei |
| Position: | APU Degree Student |
| Email: | TP067154@mail.apu.edu.my |
| Date | 1 June 2022 |
| Time | 1:00 PM |

Thank you for taking part in the testing phase of the club and society's management system. This User Acceptance Testing (UAT) tester is required to go through the testing cases specified below that are within the club and society's management system. If you have any questions while testing, please contact the developer team. We appreciate your cooperation!

The following scale standardizes the rating for the testing phase below:

* 1 – Poor
* 2 – Moderate
* 3 – Satisfied
* 4 – Good
* 5 – Outstanding

|  |  |  |
| --- | --- | --- |
| **No** | **Testing Case** | **Rate 1 – 5 (From Poor to Outstanding)** |
| **Guest** | | |
| 1 | Register an Account | 1 2 3 4 5 |
| 2 | Login | 1 2 3 4 5 |
| 3 | Send Feedback or Message to Admin | 1 2 3 4 5 |
| 4 | Browse Home, About Us, Club &Society, Event page | 1 2 3 4 5 |
| 5 | Deny guest when try to access ‘Join Club’ | 1 2 3 4 5 |

|  |  |  |
| --- | --- | --- |
| **No** | **Testing Case** | **Rate 1 – 5 (From Poor to Outstanding)** |
| **Students** | | |
| 1 | View Personal Profile | 1 2 3 4 5 |
| 2 | Change Password | 1 2 3 4 5 |
| 3 | Send Feedback or Message to Admin | 1 2 3 4 5 |
| 4 | Browse Home, About Us, Club &Society, Event page | 1 2 3 4 5 |
| 5 | Apply to join a Club | 1 2 3 4 5 |
| 6 | Deny access to ‘Join Club’ when students already have a club | 1 2 3 4 5 |
| 7 | Log Out | 1 2 3 4 5 |

|  |  |  |
| --- | --- | --- |
| **No** | **Testing Case** | **Rate 1 – 5 (From Poor to Outstanding)** |
| **Organizer** | | |
| 1 | View Event Information | 1 2 3 4 5 |
| 2 | Modify Event Information | 1 2 3 4 5 |
| 3 | Post Announcement | 1 2 3 4 5 |
| 4 | Delete Event | 1 2 3 4 5 |
| 5 | Post Event | 1 2 3 4 5 |

|  |  |  |
| --- | --- | --- |
| **No** | **Testing Case** | **Rate 1 – 5 (From Poor to Outstanding)** |
| **Committee** | | |
| 1 | View Club Information | 1 2 3 4 5 |
| 2 | Modify Club Information | 1 2 3 4 5 |
| 3 | View Club Member | 1 2 3 4 5 |
| 4 | Assign Role to Club Member | 1 2 3 4 5 |
| 5 | Remove Club Member | 1 2 3 4 5 |
| 6 | Approve or Reject Join Club Application | 1 2 3 4 5 |
| 7 | Modify Event Information | 1 2 3 4 5 |
| 8 | Post Announcement | 1 2 3 4 5 |
| 9 | Delete Event | 1 2 3 4 5 |
| 10 | Post Event | 1 2 3 4 5 |

|  |  |  |
| --- | --- | --- |
| **No** | **Testing Case** | **Rate 1 – 5 (From Poor to Outstanding)** |
| **Admin** | | |
| 1 | View User Information | 1 2 3 4 5 |
| 2 | Modify User Information | 1 2 3 4 5 |
| 3 | Delete User | 1 2 3 4 5 |
| 4 | View Club & Society Information | 1 2 3 4 5 |
| 5 | Modify Club & Society Information | 1 2 3 4 5 |
| 6 | Delete Club & Society | 1 2 3 4 5 |
| 7 | Add Club | 1 2 3 4 5 |
| 8 | View Event | 1 2 3 4 5 |
| 9 | Delete Event | 1 2 3 4 5 |
| 10 | Check User Feedback | 1 2 3 4 5 |
| 11 | Delete Feedback | 1 2 3 4 5 |
| 12 | View the Report | 1 2 3 4 5 |
| 13 | Approval Event Application | 1 2 3 4 5 |
| 14 | Decline Event Application | 1 2 3 4 5 |

|  |  |
| --- | --- |
| **Tester** | **Comment** |
| Yeap Xin Wei | All functions work as expected without bugs or unexpected error. User interface feels responsive and direct. Overall, the system functions smoothly with a minimalistic look. |
| **Developer** | **Action** |
| Lee Jia Qian | We will continue to make sure our code works as intended without any bugs and recheck to see if the codes have any redundant lines to prevent unexpected errors while running. |

# Significant Source Codes

1. **Session Start**



As the name suggest, it’s a code which creates a session or resumes the current one based on a session identifier via a GET or POST request or passed via a cookie (function.session-start.php, 2011).

**Why is this code important?**

This code is important to ensure the user stays logged in while navigating to other page after completing Log In. If this code is not present, the user is unable to keep its access and use features in the website due to the user not able to keep its access in the system.

1. **Database Connection**

These lines of code are to make connection to the database server.

Text

Description automatically generated

**Why is this code important?**

These lines of code are to ensure our system is able to connect to the database and retrieve data from it. Without these lines of code, the process of getting data is unable to proceed because there is no database to access to hence these lines of code let us to have access to the database and get the data we need to process certain functions.

# Conclusion

To conclude the report, our Radiant club and society need a management system to solve some existing problems that we have as a physical department. We have discussed the problems that we have above, and we have proposed some solutions that the management system can help us solve. In addition, the project objectives and scope play a huge factor when developing this system. It shows the importance of this management system for our Radiant club and society where it can help tremendously for the students. Before developing or implementing working codes for our management system, we have a project plan where we write our system development methodology which contains the model that we choose to work with doing this project, the requirements, system design, implementation, integration and testing, deployment and maintenance. Moreover, a project Gantt chart is also produced so that we can meet all our due dates for our tasks on time. The report also consists of a system hierarchy chart to show roughly how we want our management system to work at the end of the project. After the hierarchy chart, a context diagram is also produced where it shows how the program flow will look and from the context diagram, we can construct data flow diagram levels 0 and 1 to show more in the depth of the program. A data dictionary is then documented to specify data phrases to gather and coordinate which will be easier for other people or organizations to understand the system. Furthermore, an entity-relationship diagram otherwise known as ERD is constructed to specify a set of interconnected items of interest in a given field of expertise. Pseudocode and flowcharts are also shown based on the hierarchy chart. Screen design and user manual, as well as report design and user manual, are also shown in this report to show the physical design used by the system. Additionally, a test plan for both unit testing and user acceptance testing is carried out which will outline whether the system is usable at the end of development and feedback is given by the testers at the end. Lastly, screenshots of significant source codes are shown at the end of this document.

# References

*function.session-start.php*. (2011). Retrieved from php.net: https://www.php.net/manual/en/function.session-start.php

# Appendix

**Minutes of Meetings**

|  |
| --- |
| 1. **Project Introduction**   **Opening**  On March 18 Friday 2:10 PM, meeting is called by Lee Jia Qian and the meeting happened in Microsoft Teams.  **Attendant –** All members (Full Attendant)  **Type of Meeting –** Project Introduction  **Agenda –** Introduction to assignment and task allocation  **Meeting Description**  Group leader Lee Jia Qian brief about the assignment and the group discuss about what type of system to proceed with. Then Lee Jia Qian allocate task regarding the proposal form to member. |

|  |
| --- |
| 1. **Progress Meeting**   **Opening**  On March 23 Wednesday 12:00 PM, meeting is called by Lee Jia Qian and the meeting happened in Discord.  **Attendant** – All members (Full Attendant)  **Type of Meeting** – Progress Check  **Agenda** – Check all members task and progress while correcting unsure parts.  **Meeting Description**  Every member shows their progress and next discuss about the issues and unsure parts of their task. All member work together to clear out any misunderstanding and issue that is found. All member states out each other problems and corrects it. Lastly before the meeting ends, Lee Jia Qian announces about a meeting schedule before the proposal presentation which is March 27 Sunday to make a final check up and ready for the upcoming presentation. |
| 1. **Final Proposal Meeting**   **Opening**  On March 27 Sunday 10:00 PM, meeting is called by Tee Chor Yang and the meeting happened in Microsoft Teams.  **Attendant** – All members ( Full Attendant )  **Type of Meeting** – Final Proposal Meeting / Presentation Preparation  **Agenda** – Completion of proposal and prepare for presentation  **Meeting Description**  All members show their completed work while also correcting and spots everybody errors, after the work is check and agree upon, all members then make preparation towards next day proposal presentation. |

|  |
| --- |
| 1. **Logical Diagram**   **Opening**  On April 1 Friday 2:00 PM, meeting is called by Lee Jia Qian and the meeting happened in Discord.  **Attendant** – All members ( Full Attendant )  **Type of Meeting** – Briefing about logical diagram / task allocation  **Agenda** – Start doing logical diagram  **Meeting Description**  Proceed with assignment on logical diagram such as context diagram, level 0 and ERD. Plans on logical diagram and how to construct them discussed .Next, task is then allocated to everybody evenly to complete each logical diagrams. |
| 1. **Logical Diagram Progress**   **Opening**  On April 11 Monday 2:00 PM, meeting is called by Thaddeus Then Ze Qin and the meeting happened in Discord.  **Attendant** – All members ( Full Attendant )  **Type of Meeting** – Progress check  **Agenda** – Check logical diagram progression  **Meeting Description**  Everybody shows their work of progress and minor tweak is done to keep everybody at the same pace and path towards the final diagram. |

|  |
| --- |
| 1. **System Development**   **Opening**  On April 18 Monday 2:00 PM, meeting is called by Lee Jia Qian and the meeting happened in Discord.  **Attendant** – All members ( Full Attendant )  **Type of Meeting** – Website design  **Agenda** – Start to develop website  **Meeting Description**  All member starts with the planning phase of the system such as looks, functions, and how to code it. Next, all idea is discuss and tools to start implement the system is downloaded. Then, task is allocated to everybody evenly to code each part of the system. |
| 1. **System Development Progress**   **Opening**  On April 26 Tuesday 2:00 PM, meeting is called by Lee Jia Qian and the meeting happened in Discord.  **Attendant** – All members ( Full Attendant )  **Type of Meeting** – Progress check on coding  **Agenda** – Check all member task progress  **Meeting Description**  All member shows their progress regarding the system. Function of the system is prioritised as for this meeting to ensure all function works as intended. All member have yet to finish all functioning part as all part is still in development. |

|  |
| --- |
| 1. **System Development Progress**   **Opening**  On May 9 Monday 2:00 PM, meeting is called by Teoh Yee Chin and the meeting happened in Discord.  **Attendant** – All members ( Full Attendant )  **Type of Meeting** – Progress check on coding  **Agenda** – Check all member task progress  **Meeting Description**  All member shows their progress regarding the system. Resuming the last meeting, functions of the system is check and ensure that every member deliver the function successfully. All member show part of functioning code and other finished part such as design. As till this meeting, design of the system is not done around 40% of the functions is done. |
| 1. **System Development Progress**   **Opening**  On May 16 Mon 2:00 PM, meeting is called by Lee Jia Qian and the meeting happened in Discord.  **Attendant** – All members ( Full Attendant )  **Type of Meeting** – Progress check on coding  **Agenda** – Check all member task progress  **Meeting Description**  All member shows their progress regarding the system. As similar to the last meeting, all member shows their part and focuses the function of the system such as buttons, reading inputs, creates, delete and many more. Around 70% of the functions are made and errors are starting to appear during implementation of code. Errors are discuss and all member assist each other to solve the problem occurred. |

|  |
| --- |
| 1. **System Development Progress**   **Opening**  On May 23 Monday 2:00 PM, meeting is called by Lee Jia Qian and the meeting happened in Discord.  **Attendant** – All members ( Full Attendant )  **Type of Meeting** – Progress check on coding  **Agenda** – Check all member task progress  **Meeting Description**  All member shows their progress regarding the system. With this meeting, majority of the function is presented by all member and implementation of website design starts. All members would now start doing their system part as for design and also resume their completion towards all function. |
| 1. **System Development Progress**   **Opening**  On May 27 Friday 2:00 PM, meeting is called by Wong Yew Wei and the meeting happened in Discord.  **Attendant** – All members ( Full Attendant )  **Type of Meeting** – Progress check on coding  **Agenda** – Check all member task progress  **Meeting Description**  All member shows their progress regarding the system. Design of the system is completed and all functions are create but some function are still yet to work. Error in the code appeared commonly and during this meeting, all member priorities their task on debugging their code and keep the codes neatly. |

|  |
| --- |
| 1. **Progress Meeting To Lecturer**   **Opening**  On May 29 Sunday 11:00 PM, meeting is called by Lee Jia Qian and the meeting happened in Discord.  **Attendant** – All members ( Full Attendant )  **Type of Meeting** – Preparation.  **Agenda** – prepare for progress meeting to lecturer  **Meeting Description**  All members finish up codes that are close to completion and set aside other task which still need some time to ensure we are able to present as much for a progress to lecturer. All completion, all members do a check and update everybody code to ensure everybody stayed up to date to the latest system code. |
| 1. **System Development Completion**   **Opening**  On May 31 Mon 2:00 PM, meeting is called by Tee Chor Yang and the meeting happened in Discord.  **Attendant** – All members ( Full Attendant )  **Type of Meeting** – Progress completion  **Agenda** – Finish up the system  **Meeting Description**  All members ensure all parts of theirs and each other task is completed, and a full system test is done to ensure all part meets the requirement and is working as intended. Lee Jia Qian then starts to allocate task for the completion of final documentation. |

|  |
| --- |
| 1. **Individual Part Progress/task allocation**   **Opening**  On June 1 Wednesday 2:00 PM, meeting is called by Lee Jia Qian and the meeting happened in Discord.  **Attendant** – All members ( Full Attendant )  **Type of Meeting** – Progress check  **Agenda** – check on all member’s individual part  **Meeting Description**  All members state out their progression for individual part and insert into the assignment file for part that is completed. Meantime, task is allocated to all member regarding the final documentation. |
| 1. **Final Documentation Progress**   **Opening**  On June 3 Thursday 2:00 PM, meeting is called by Lee Jia Qian and the meeting happened in Discord.  **Attendant** – All members ( Full Attendant )  **Type of Meeting** – Progression  **Agenda** – check on final documentation progress  **Meeting Description**  All member shows their completed part and paste in the assignment word file and checks on each other part to ensure no careless mistake are made. By this meeting, around 70% of the documentation is done, part such as screen, report design and test plan are still unfinished. |

|  |
| --- |
| 1. **Final Documentation Progress**   **Opening**  On June 5 Sunday 2:00 PM, meeting is called by Lee Jia Qian and the meeting happened in Discord.  **Attendant** – All members ( Full Attendant )  **Type of Meeting** – Progress check  **Agenda** – check progress on final documentation  **Meeting Description**  All member shows their completed part and paste in the assignment word file and checks on each other part to ensure no careless mistake are made. By this meeting, around 85% of the documentation is done, part such as test plan and appendix are still unfinished. |
| 1. **Documentation Check**   **Opening**  On June 7 Tuesday 2:00 PM, meeting is called by Lee Jia Qian and the meeting happened in Discord.  **Attendant** – All members ( Full Attendant )  **Type of Meeting** – Assignment check up  **Agenda** – Ensure all part of the assignment is done  **Meeting Description**  All members ensure all part of the assignment is done and do some minor tweaks to keep the documentation neat and presentable. Any information to be added is also added by members who thinks some part needed extra details. |

|  |
| --- |
| 1. **Final Check**   **Opening**  On June 8 Wednesday 2:00 PM, meeting is called by Lee Jia Qian and the meeting happened in Discord.  **Attendant** – All members ( Full Attendant )  **Type of Meeting** – Revise  **Agenda** – Do a final check up on the assignment  **Meeting Description**  All members do a final check-up on the assignment to ensure all part is completed. Next, the assignment file is submitted by Lee Jia Qian. |

**Installation Guide**

1. Download XAMPP (<https://www.apachefriends.org/index.html>)
2. Download our MySQL data file ( sdp.sql in the zip file )
3. Start-up XAMPP and click on start for APACHE and MYSQL

A picture containing shape

Description automatically generated

1. Click on admin on MySQL to open up database in your browser



1. Drag sdp.sql database file into phpMyAdmin browser page to download the database.

Graphical user interface, text, application

Description automatically generated

1. A notification of your action will pop out at your bottom right corner regarding if database is successfully created.
2. Next, u will see a new database call sdp at the left side of the menu bar. Click on it to view information regarding the database.

Graphical user interface, text, application

Description automatically generated

1. Download any code editor ( recommended is Visual Studio Code as we used it )

<https://code.visualstudio.com/>

1. Download our code file in the zip or go to GitHub and download it manually. ( <https://github.com/TheOne0626/RADIANT-Club-Soicety> )

Note if downloading from GitHub – Click on the green button called Code and click on download ZIP

Graphical user interface, text, application, chat or text message

Description automatically generated

1. Place the code file into XAMPP htdocs folder, this is too able to open our code in browser.



1. Open the file using your code editor.
2. Navigate to index.php and open in browser.



Graphical user interface, text

Description automatically generated

1. You are now able to use our website.

**Extra notes**

* **Reminder to turn off XAMPP**
* **When terminating XAMPP application, an error always pops out (not sure why)**