

Software Requirements CSE 6224

Group member:

TEOW WEI TING 1231302923

ENG ZI YING 1211112187

SEE JIE SHENG 1211110469

SUA WEI KHONG 1221101288

**Part 4: Elicitation Execution**

# 

To gather a robust set of requirements for our Student Club Management System, we executed three distinct elicitation techniques as planned: Brainstorming, Questionnaire, and Perspective-based Reading. The findings from each session are documented, categorized using the Kano Model.

# 

# **1. Technique 1: Brainstorming**

## **1.1 Execution**

**Session Overview**

We conducted three separate brainstorming sessions, each targeting different modules of the Student Club Management System. The sessions followed a structured format with a rotating facilitator and documented ideas collaboratively using Google Docs. Each session focused on a specific functional area to ensure depth and relevance.

**Tools Used:** Google Docs, WhatsApp Group Chat, Teams meetings

**Key Outcomes Categorized by Kano Model:**

* **Dissatisfiers:**
  + User Authentication
  + Event Creation Interface
  + Basic Membership Management
  + Basic Budget Tracking and Report Download
  + Conflict Detection in Venue Booking
* **Satisfiers:**
  + Real-Time Venue Availability
  + Budget Submission & Approval Flow
  + Role Management
  + Notification System
  + Calendar Integration with Personal Accounts
* **Delighters:**
  + AI-powered Event Suggestions
  + Mobile App Access
  + Automatic Event Promotion via University Channels

## 

## 

## 

## 

## 

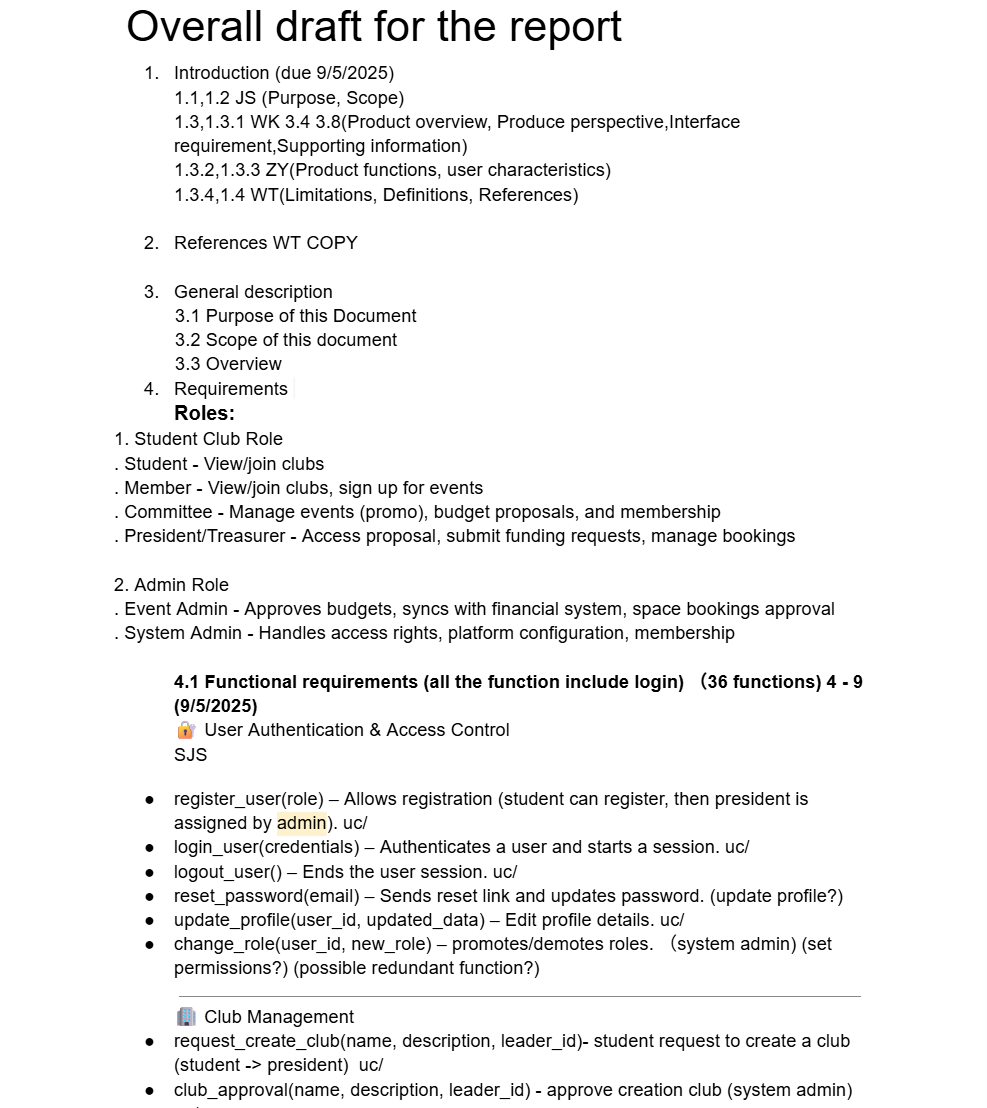
## 

## 

## 

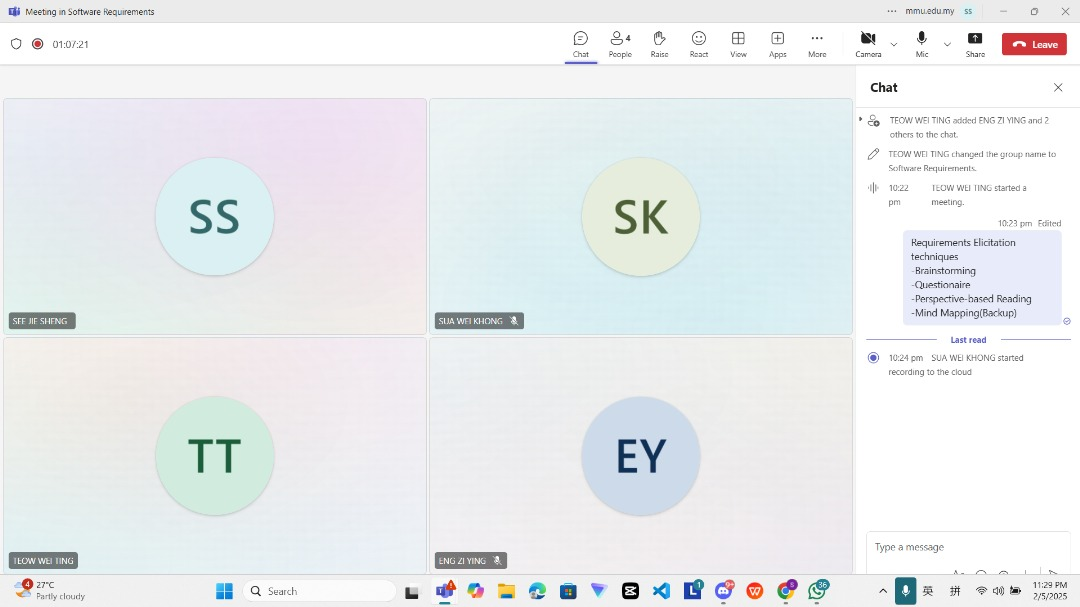
## **1.2 Proof**

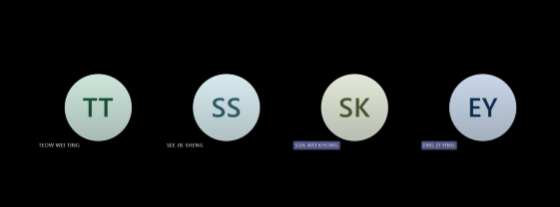
Shared Google Docs for idea collection:



Team meeting in using Microsoft Teams for brainstorming session discussion







# 

# 

# 

# 

# **2. Technique 2: Questionnaire Distribution**

## **2.1 Execution**

**Execution Overview:**We designed and distributed a questionnaire via Microsoft Forms and shared it through social media and WhatsApp groups. We received responses from 30 individuals across different stakeholder categories.

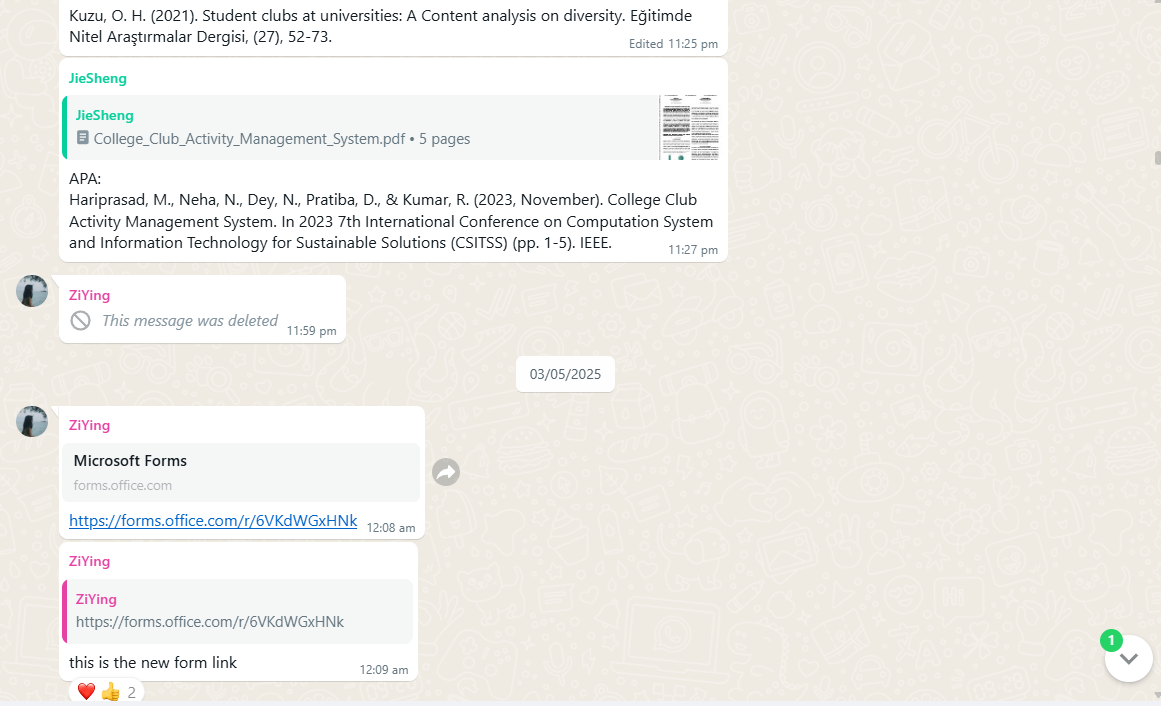
**Target Audience:** Club Members, Club Presidents, Committee, Admin  
**Total Valid Responses:** 30  
**Tools Used:** Microsoft Forms, Excel for analysis

**Key Outcomes Categorized by Kano Model:**

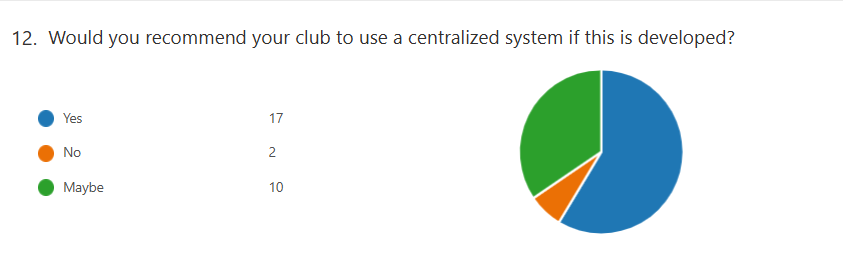
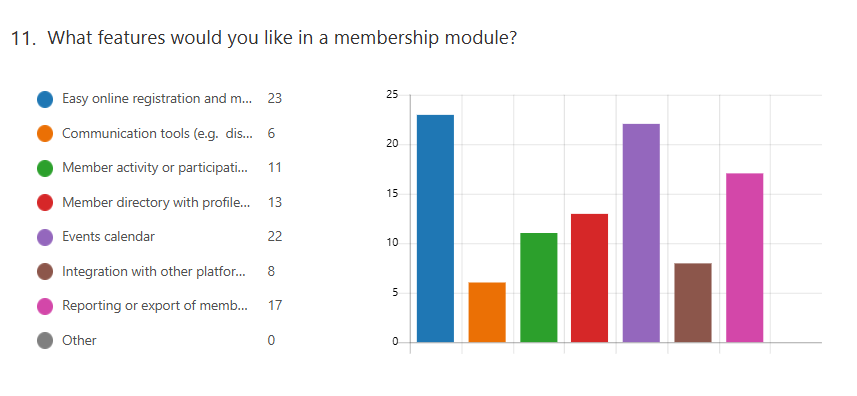
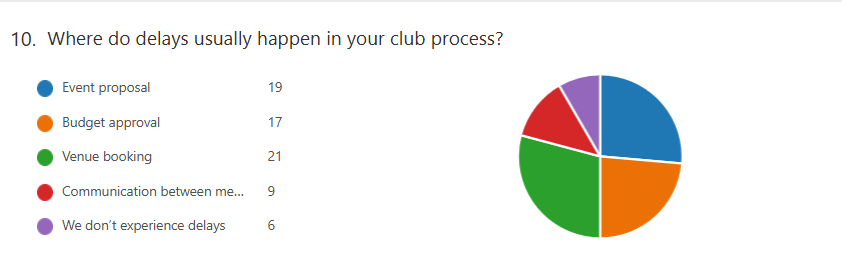
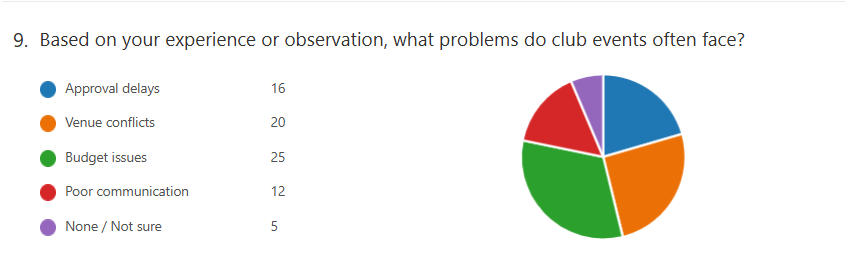
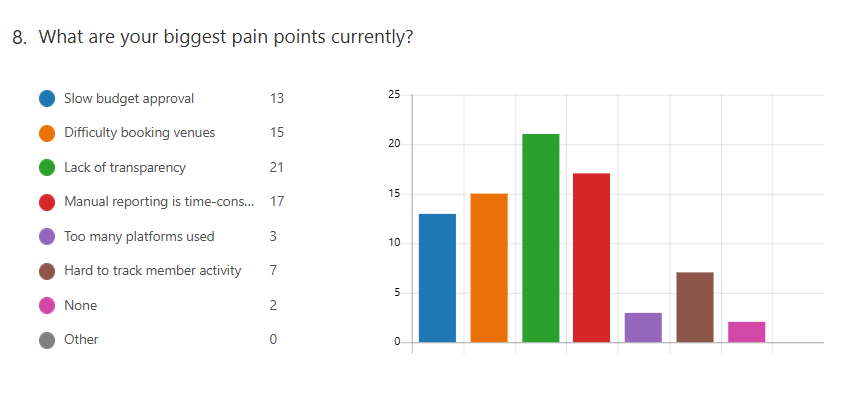
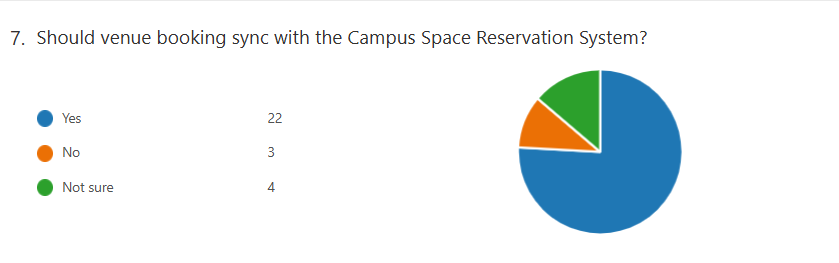
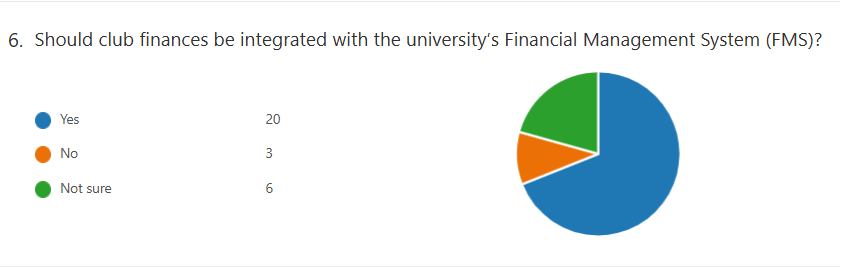
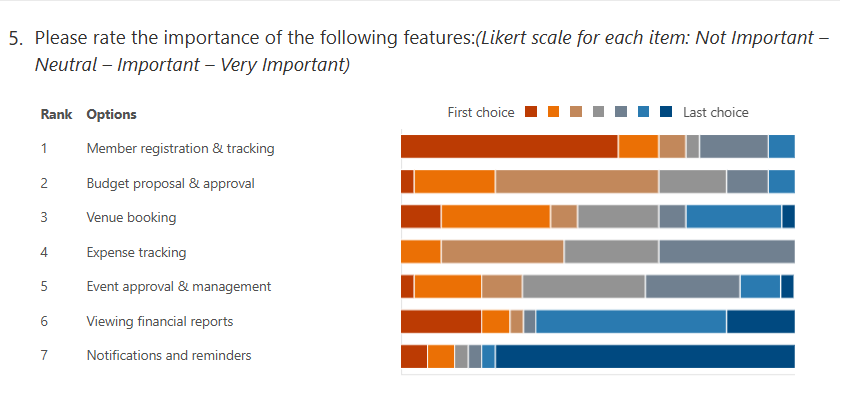
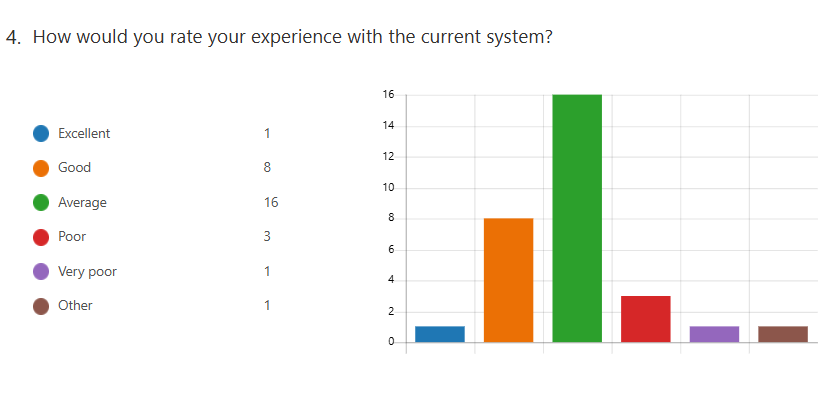
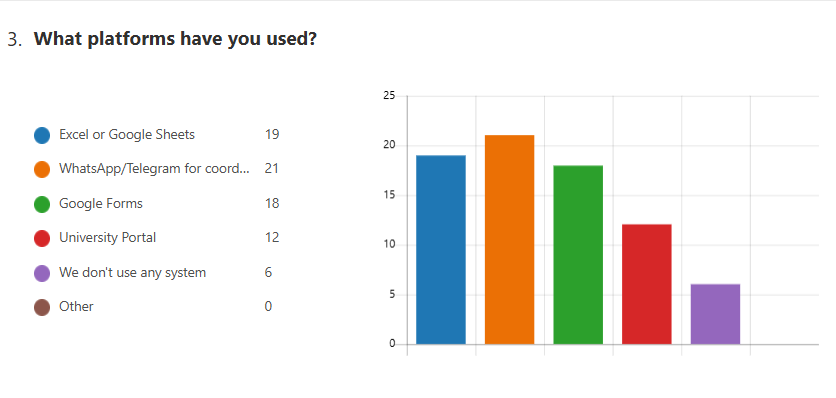
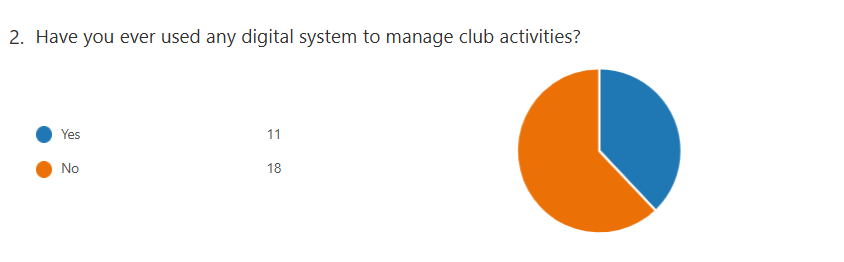
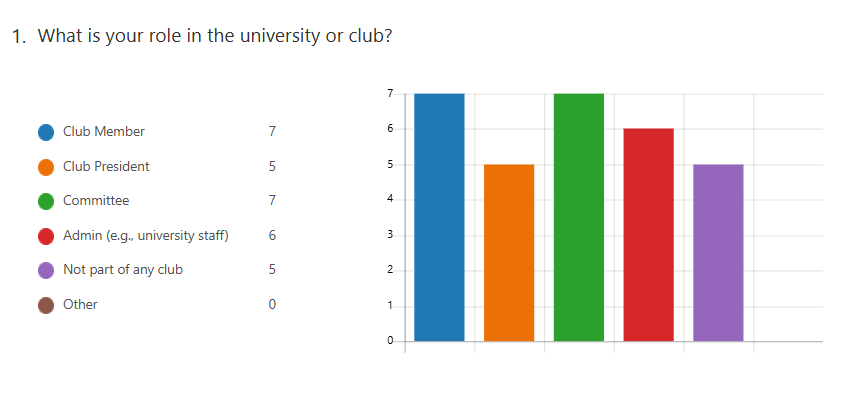
* **Dissatisfiers:**
  + **User Authentication**: Basic login/signup functionality for all user roles (e.g., admins, club officers, members).
  + **Membership Management**: Ability to register, manage, and track student memberships in clubs.
  + **Event Creation**: Functionality to create and manage events (title, description, date, etc.).
  + **Budget Management**: View current budgets and track expenses, integrated with the university's financial system.
  + **Venue Reservation Integration**: Basic integration with the university’s venue reservation system to book spaces.
* **Satisfiers:**
  + **Budget Submission and Approval System**: Submit budget proposals and track their approval status.
  + **Expense Tracking & Monitoring**: Real-time tracking and categorization of club expenses.
  + **Membership Renewal**: Online renewal process for club memberships.
  + **Communication Tools**: Integrated tools for announcements and event communications within the platform.
* **Delighters:**
  + **Automated Budget Optimization**: Suggest ways to optimize event budgets based on spending patterns.
  + **Real-Time Budget Alerts**: Notifications when budgets are nearing limits or discrepancies are detected.
  + **Peer Reviews of Events**: Allow members to rate events, providing insights into their success and engagement.
  + **Advanced Reporting Tools**: Visual analytics for tracking fund usage across categories or time periods.

## **2.2 Proof**

Created Microsoft Forms and shared in group



Gathered data from the questionnaire:



# 

# 

# 

# 

# 

# 

# 

# **3. Technique 3: Perspective-based Reading**

## **3.1 Execution**

We have found the related journals, and do the research for every journal. Besides, we have listed out the session overview and the parts which are related to our topic. So, we can have a better understanding on our topic which can make the report as comprehensive as we can.

**Session Overview**

To support the elicitation of functional and non-functional requirements for the Student Club Management System, the team analyzed five journal articles and case studies that explored similar platforms. Each member reviewed these resources from the perspective of a specific actor (Student, Club Member, Committee, President, Admin), focusing on identifying relevant system features, role-based workflows, and pain points.

1. **Campus Club Management System Application**- Mukthashree B\* , Manasi N S\* , Chinmayee K G\* , Amrutha G\* , Manjuprasad B# Student\*, Associate Professor#

**Summarize By: Teow Wei Ting**

**Overview:**

A web-based system designed to streamline the planning, feedback, and certificate generation of student club events is presented in the journal paper "Campus Club Management System Application." The project, which was constructed with PHP, MySQL, HTML, CSS, and JavaScript, makes possible features like certificate creation, event registration, feedback gathering, and admin-based control for resource uploads and participant tracking.

**Similarities to our project:**

* **Centralized Club Operations:** Both projects aim to provide a single digital platform where club-related tasks—like event registration, user roles, and activity tracking—can be managed efficiently.
* **Role-Based Access:** Just like your system, the journal describes different user roles (e.g., admin, student), each with specific functionalities and access rights.
* **Student-Centered Design:** The primary goal in both cases is to streamline the club experience for students and encourage participation by making the process simple and transparent.
* **Event Workflow:** Both systems enable end-to-end event workflows, from proposal or initiation to registration and post-event feedback.

1. **An Enhanced Django Framework for effective Club Connect Management System**- Chenjeri Aditya Sai\* , T. Satya\* , K. Santhan Mourya\* , Dr. Murali Krishna Muddada#

**Summarized by: Eng Zi Ying**

**Overview:**

The creation and implementation of the Club Connect Management System, a web-based platform designed to simplify college student clubs' communication, coordination, and event planning procedures, are covered in the magazine. It was created with Django, HTML, CSS, and Bootstrap and provides features for viewing and filtering student data, adding and removing members, and centralizing student club information for simpler event management. By allowing instructors and staff to select students according to their responsibilities and abilities for certain events, it facilitates focused student engagement. The approach seeks to lower event planning expenses, increase efficiency, and promote teamwork.

**Similarities to our project:**

1. **Centralized Platform** Both systems aim to provide a centralized digital hub for managing student club operations.
2. **Membership Management** Both support adding, removing, viewing, and filtering student member information.
3. **Role-based User Access** Users are categorized based on roles (e.g., student, faculty, admin) to control access and functionality.
4. **Event Management** Both systems allow for creating and managing club events, with support for organizing and tracking student involvement.
5. **Communication Facilitation** Both include features for improving communication among members and faculty (e.g., announcements, outreach based on skills).
6. **Web-based Interface** Both are implemented as responsive web applications with user-friendly interfaces, designed using Django as the backend framework.
7. **Database Integration** Use of Django’s ORM and relational databases (SQLite) for managing structured data like student records and club details.
8. **Focus on Efficiency** Both projects aim to reduce manual effort in coordinating events and managing clubs through automation and digitization.
9. **Scalability Considerations** Both mention the need for the system to scale effectively with the number of clubs or users over time.
10. **Target Users** Faculty, staff, and student club members are primary users in both systems, with faculty involved in oversight and student selection.
11. **College Club activity management system**- Malavika Hariprasad, Neha N, Nimisha Dey, Dr. Pratiba D, Dr Ramakanth Kumar P

**Summarized by: See Jie Sheng**

**Overview:**

A web-based platform called the College Club Activity Management System was created to make managing student organizations in educational institutions easier. Features like role management, permissions, communication, event scheduling, membership registration, and feedback analysis are all centralized. The system integrates both SQL (SQLite) and NoSQL (MongoDB) databases, and its backend is built with Django. Its integration with a Natural Language Processing (NLP) model for sentiment analysis of event feedback is a noteworthy feature that aids clubs in enhancing their performance through insights derived from data. To facilitate future expansion, the platform places a strong emphasis on modular design, secure role-based access, and user-friendliness.

**Similarities to our project:**

1. **Centralized Platform**:  
    Both systems serve as a unified web application to manage all student clubs and their activities.
2. **Membership Management**:  
    Support for member registration, role assignment (e.g., president, secretary), join requests, and approval workflows.
3. **Role-Based Access Control**:  
    Permissions and functionalities are restricted based on user roles such as admin, core team member, or club member.
4. **Event Scheduling and Management**:  
    Events can be created, scheduled, and viewed in an integrated calendar interface.
5. **Communication Features**:  
    Both include functionality to send notices/announcements to club members for coordination and updates.
6. **Feedback Collection**:  
    Collection of event feedback from members for analysis and reporting.
7. **Use of Django Framework**:  
    Backend developed using Django, supporting modularity, scalability, and quick development.
8. **Responsive Web Design**:  
    Frontend is implemented using HTML, Tailwind CSS, and JavaScript for mobile and desktop compatibility.
9. **Automation of Administrative Tasks**:  
    Both systems aim to automate routine workflows (e.g., membership handling, event scheduling).
10. **Analytics and Reporting**:  
     Generate event or feedback reports; the journal uses NLP sentiment analysis while your system uses structured metrics.
11. **Modular System Architecture**:  
     Features are organized in modules to allow easier maintenance and future feature expansion.
12. **Database Support**:  
     Both use SQL databases (SQLite or PostgreSQL) to store persistent data such as users, events, and permissions.

**Overview:**

The design of an Integrated Student Activities Management System (iSAMS), created for Universiti Teknologi MARA to simplify the entire co-curricular activity lifecycle for students, is presented in the journal. The entire workflow is digitalized using iSAMS, a web-based platform that handles everything from budget approval and proposal submission to event planning, feedback gathering, and reporting. The solution intends to improve management skills for more than 100 student clubs operating across several campuses and eliminate inefficiencies caused by handwritten documentation. Notifications, role-based access, attendance monitoring, document uploads, and report generation are among its capabilities, which are all part of its MVC design. Stakeholders' actual user and system requirements serve as the foundation for the methodology's structured Waterfall paradigm.

**Similarities to our project:**

1. **Centralized Platform** Both systems are designed to be a unified hub for managing all student club activities and data.
2. **Lifecycle Management** Handle full club activity lifecycle: proposal submission, approvals, execution, closure reporting.
3. **Budget Management** Each system includes features to submit and review event budgets, upload receipts, and track usage.
4. **Event Proposal and Approval** Clubs can submit proposals, and admins can approve or reject them through the system.
5. **Role-Based Access** Users such as students, organizers, and admins have different roles and permissions.
6. **Notification and Reminders** Systems notify users of new events, approvals, deadlines, and low budget warnings.
7. **Feedback Collection** Event participants can provide feedback, which can be used for reporting or post-mortems.
8. **Web-Based Interface** Both are accessible through secure web portals and use modern UI frameworks.
9. **Secure Document Storage** Upload and manage supporting documents (receipts, photos, reports) related to activities.
10. **Report Generation** Generate activity summaries, attendance reports, and financial overviews in specific formats.
11. **Designed for Scalability** Systems are intended to support large numbers of clubs and high event frequency.
12. **Structured Methodology** Both projects rely on formal design approaches (e.g., Waterfall, MVC) with clear architectural planning.

**4. Design Architecture Of An Integrated Student Activities Management System For Higher Education**- Azlan Abdul Aziz1 , Jamaluddin Jasmis2 , Syamsul Arifin Yahya 3 , AlyaGeogiana Buja4 , Noor Afni Deraman5 , MohdNorHajarHasrol Jono6

**Summarized by: Sua Wei Khong**

**Overview:**

The design of an Integrated Student Activities Management System (iSAMS), created for Universiti Teknologi MARA to simplify the entire co-curricular activity lifecycle for students, is presented in the journal. The entire workflow is digitalized using iSAMS, a web-based platform that handles everything from budget approval and proposal submission to event planning, feedback gathering, and reporting. The solution intends to improve management skills for more than 100 student clubs operating across several campuses and eliminate inefficiencies caused by handwritten documentation. Notifications, role-based access, attendance monitoring, document uploads, and report generation are among its capabilities, which are all part of its MVC design. Stakeholders' actual user and system requirements serve as the foundation for the methodology's structured Waterfall paradigm.

**Similarities to our project:**

1. **Centralized Platform** Both systems are designed to be a unified hub for managing all student club activities and data.
2. **Lifecycle Management** Handle full club activity lifecycle: proposal submission, approvals, execution, closure reporting.
3. **Budget Management** Each system includes features to submit and review event budgets, upload receipts, and track usage.
4. **Event Proposal and Approval** Clubs can submit proposals, and admins can approve or reject them through the system.
5. **Role-Based Access** Users such as students, organizers, and admins have different roles and permissions.
6. **Notification and Reminders** Systems notify users of new events, approvals, deadlines, and low budget warnings.
7. **Feedback Collection** Event participants can provide feedback, which can be used for reporting or post-mortems.
8. **Web-Based Interface** Both are accessible through secure web portals and use modern UI frameworks.
9. **Secure Document Storage** Upload and manage supporting documents (receipts, photos, reports) related to activities.
10. **Report Generation** Generate activity summaries, attendance reports, and financial overviews in specific formats.
11. **Designed for Scalability** Systems are intended to support large numbers of clubs and high event frequency.
12. **Structured Methodology** Both projects rely on formal design approaches (e.g., Waterfall, MVC) with clear architectural planning.

**5. Enhancing Campus Engagement with A Secure and Comprehensive Platform for Club Management and Student Participation**- Ishika Amit Khatr, Manthan Ghonge, Riddhi Mirajkar, Suhani Shinde, Ritesh Hon, Anuradha Yenkikar

**Summarized by: See Jie Sheng**

**Overview:**

The article presents a mobile application designed to improve student involvement and simplify club administration in educational settings. The tool, which was created specifically for use at Vishwakarma Institute of Information Technology, gives students the ability to find and join organizations, sign up for events, and get updates in one digital location. Club presidents and faculty advisers have the authority to publish event information, maintain club profiles, and monitor member participation. Additionally, the platform offers a chatbot driven by AI to provide prompt answers to student questions, encouraging informed engagement. By accepting club registrations, event plans, and student inquiries, administrators play a crucial part in maintaining the system. The system, which was developed with Firebase and Android technologies, facilitates real-time communication and has a scalable architecture designed to foster a lively and connected campus community.

**Similarities to our project:**

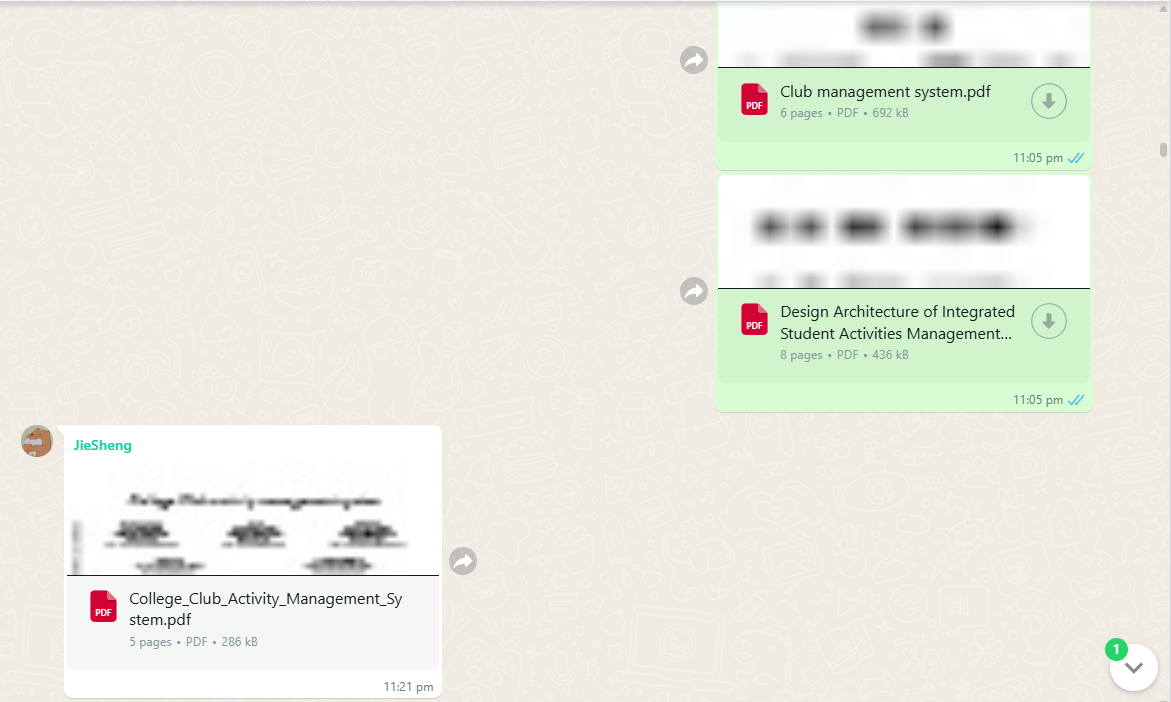
1. **Centralized Platform** Both platforms serve as unified systems for managing student clubs, events, and user data.
2. **Lifecycle Management** Cover the entire process from club creation and event planning to approvals and execution.
3. **Budget Management** Enable submission and approval of budget proposals, with oversight from administrators.
4. **Event Proposal and Approval** Clubs can submit events for review, and designated admins can approve or reject proposals.
5. **Role-Based Access** Defined roles (e.g., student, president, admin) determine access and control over specific features.
6. **Notification and Reminders** Include automated alerts for upcoming events, submission deadlines, and approvals.
7. **Feedback Collection** Support feedback mechanisms to gather student input after events.
8. **Mobile or Web-Based Interface** Campus Connect uses a mobile-first approach, while our project uses a responsive web portal.
9. **Secure Document Storage** Allow uploading of supporting materials like event flyers, receipts, and attendance logs.
10. **Report Generation** Generate downloadable reports for activities, financial tracking, and event statistics.
11. **Designed for Scalability** Both are designed to handle multiple clubs and events across large institutions.
12. **Structured Methodology** Each system is based on a structured development approach (Campus Connect uses a systematic phase-by-phase methodology; ours uses MVC and Django).

**Reviewed Articles Summary**

1. **Campus Club Management System Application**
   * Focuses on event planning, feedback, and certificate generation using a web-based system.
   * Supports role-based access for admins and students.
   * Key Features: Event registration, resource uploads, participant tracking.
   * Relevance: Emphasizes usability, end-to-end event workflows, and centralized access similar to our system.
2. **An Enhanced Django Framework for Effective Club Connect Management**
   * A Django-based system that centralizes communication and member/event management.
   * Includes role filtering, data-driven student engagement, and administrative tools.
   * Key Features: Add/remove members, filter by skills, simplified event workflows.
   * Relevance: Highlights efficiency, targeted engagement, and Django ORM integration.
3. **College Club Activity Management System**
   * A modular, Django-based platform integrating SQL and NoSQL databases.
   * Features include feedback analytics via NLP, secure role-based access, and event scheduling.
   * Key Features: Role assignment, event feedback analysis, announcements.
   * Relevance: Aligns with our goals of feedback-driven improvement and scalable modular design.
4. **Integrated Student Activities Management System (iSAMS)**
   * A structured, Waterfall-model-based system for end-to-end club activity lifecycle management.
   * Includes proposal and budget workflows, document uploads, and automated reminders.
   * Key Features: Attendance monitoring, budget tracking, scalable MVC architecture.
   * Relevance: Emphasizes structured lifecycle and administrative oversight, useful for President and Admin roles.
5. **Campus Engagement Platform for Club Management**
   * A mobile-first system with chatbot integration for real-time student support.
   * Enables event registration, club joining, and notification updates.
   * Key Features: Real-time communication, AI-driven chatbot, faculty-advised club oversight.
   * Relevance: Highlights mobile engagement, automation, and scalable design—especially valuable from the student and committee perspective.

## **3.2 Proof**

Share found research paper in team group chat



Summary for each article found from each member:  
1) **Campus Club Management System Application**- Mukthashree B\* , Manasi N S\* , Chinmayee K G\* , Amrutha G\* , Manjuprasad B# Student\*, Associate Professor#

**Summarise by: Teow Wei Ting**

**Similarities to our project:**

* **Centralized Club Operations:** Both projects aim to provide a single digital platform where club-related tasks—like event registration, user roles, and activity tracking—can be managed efficiently.
* **Role-Based Access:** Just like your system, the journal describes different user roles (e.g., admin, student), each with specific functionalities and access rights.
* **Student-Centered Design:** The primary goal in both cases is to streamline the club experience for students and encourage participation by making the process simple and transparent.
* **Event Workflow:** Both systems enable end-to-end event workflows, from proposal or initiation to registration and post-event feedback.

**2) An Enhanced Django Framework for effective Club Connect Management System**- Chenjeri Aditya Sai\* , T. Satya\* , K. Santhan Mourya\* , Dr. Murali Krishna Muddada#

**Summarise by: Eng Zi Ying**

**Similarities to our project:**

1. **Centralized Platform** Both systems aim to provide a centralized digital hub for managing student club operations.
2. **Membership Management** Both support adding, removing, viewing, and filtering student member information.
3. **Role-based User Access** Users are categorized based on roles (e.g., student, faculty, admin) to control access and functionality.
4. **Event Management** Both systems allow for creating and managing club events, with support for organizing and tracking student involvement.
5. **Communication Facilitation** Both include features for improving communication among members and faculty (e.g., announcements, outreach based on skills).
6. **Web-based Interface** Both are implemented as responsive web applications with user-friendly interfaces, designed using Django as the backend framework.
7. **Database Integration** Use of Django’s ORM and relational databases (SQLite) for managing structured data like student records and club details.
8. **Focus on Efficiency** Both projects aim to reduce manual effort in coordinating events and managing clubs through automation and digitization.
9. **Scalability Considerations** Both mention the need for the system to scale effectively with the number of clubs or users over time.
10. **Target Users** Faculty, staff, and student club members are primary users in both systems, with faculty involved in oversight and student selection.

**3) College Club activity management system**- Malavika Hariprasad, Neha N, Nimisha Dey, Dr. Pratiba D, Dr Ramakanth Kumar P

**Summarize by: See Jie Sheng**

**Similarities to our project:**

1. **Centralized Platform**:  
    Both systems serve as a unified web application to manage all student clubs and their activities.
2. **Membership Management**:  
    Support for member registration, role assignment (e.g., president, secretary), join requests, and approval workflows.
3. **Role-Based Access Control**:  
    Permissions and functionalities are restricted based on user roles such as admin, core team member, or club member.
4. **Event Scheduling and Management**:  
    Events can be created, scheduled, and viewed in an integrated calendar interface.
5. **Communication Features**:  
    Both include functionality to send notices/announcements to club members for coordination and updates.
6. **Feedback Collection**:  
    Collection of event feedback from members for analysis and reporting.

**4) Design Architecture Of An Integrated Student Activities Management System For Higher Education**- Azlan Abdul Aziz1 , Jamaluddin Jasmis2 , SyamsulAriffin Yahaya3 , AlyaGeogiana Buja4 , Noor Afni Deraman5 , MohdNorHajarHasrol Jono6

**Summarize by: Sua Wei Khong**

**Similarities to our project:**

1. **Centralized Platform** Both systems are designed to be a unified hub for managing all student club activities and data.
2. **Lifecycle Management** Handle full club activity lifecycle: proposal submission, approvals, execution, closure reporting.
3. **Budget Management** Each system includes features to submit and review event budgets, upload receipts, and track usage.
4. **Event Proposal and Approval** Clubs can submit proposals, and admins can approve or reject them through the system.
5. **Role-Based Access** Users such as students, organizers, and admins have different roles and permissions.
6. **Notification and Reminders** Systems notify users of new events, approvals, deadlines, and low budget warnings.
7. **Secure Document Storage** Upload and manage supporting documents (receipts, photos, reports) related to activities.
8. **Report Generation** Generate activity summaries, attendance reports, and financial overviews in specific formats.

# **4. Technique 4: Interview**

## **4.1 Execution**

To gain deeper insights into user expectations and ensure the system is both user-friendly and comprehensive, we conducted an interview with Wong Jun Jie, an experienced student club leader actively involved in organizing campus events. The objective of this interview was to understand the practical needs, frustrations, and desired features from a stakeholder’s perspective. By applying the Kano Model during the session, we were able to classify system features into must-have, performance, and delighter categories, which will guide our design decisions and help prioritize functionalities that enhance overall user satisfaction.

**1. Functional vs. Dysfunctional Questions**

* Ask: "How do you feel if this feature is there?"
* Ask: "How do you feel if this feature is not there?"
* Use answers like:
  + I like it
  + I expect it
  + I’m neutral
  + I can accept it
  + I don’t like it

**2. Probing Questions**

* Ask follow-ups like:
  + “Why is this useful to you?”
  + “What if it’s missing?”

**3. Stakeholder Role Questions**

* Ask based on their role:
  + “What helps you manage events or budgets?”
  + “What do you find frustrating?”

**4. Feature Ranking**

* Ask: “Which feature is most important?”
* Helps to group features as must-have, useful, or bonus.

**Participants:**

* **Interviewer (INT)** – Project team member
* **Wong Jun Jie (WJJ)** – Stakeholder and experienced student club president

## **4.2 Proof**

**Transcript:**

**INT:** Good afternoon, Wong Jun Jie. Thanks for taking the time to speak with us today about your experience and expectations for a club management system.

**WJJ:** No problem at all, happy to help.

**INT:** Let's start simple. What features would you expect any student club management platform to have?

**WJJ:** Well, basic stuff first—logins for members, a dashboard where we can post events, and some kind of attendance tracking or signup.

**INT:** Great. Now, for each feature, I’d like to ask how you'd feel if the system *had* it and how you'd feel if it *didn’t*. First: If the system allows online event creation and management—how would you feel?

**WJJ:** I’d say I expect it. It’s kind of standard now.

**INT:** And if the system *didn’t* allow you to create or manage events online?

**WJJ:** That would be frustrating. We’d go back to Google Forms or WhatsApp. Very inefficient.

**INT:** So it’s a **must-be** feature. What about real-time venue booking—being able to see available slots and book instantly?

**WJJ:** Oh, I’d love that. That would save so much time! Right now, we send emails and wait for replies.

**INT:** And without it?

**WJJ:** I can tolerate that, but I’d be disappointed. I think it adds value—maybe a **satisfier**?

**INT:** Noted. Next—suppose the system not only tracks expenses but also shows alerts when you’re reaching budget limits. How does that sound?

**WJJ:** That would be fantastic. We always go over budget and only realize after the event.

**INT:** And if the system didn’t offer budget alerts?

**WJJ:** It wouldn’t be the end of the world, but it would definitely reduce the system’s usefulness.

**INT:** Sounds like a **delighter** feature. Now let’s talk about notifications—if the system reminds you about pending approvals or event deadlines?

**WJJ:** I like it that way. Keeps things on track.

**INT:** Without it?

**WJJ:** We’d miss things, so again, not ideal. I’d say this is a strong **satisfier**.

**INT:** Understood. What about something less expected—like an AI tool that suggests the best event timing or budgeting strategy based on past events?

**WJJ:** Wow, that’s interesting. Never thought of that. I’d be really impressed if that worked well.

**INT:** And if it wasn’t included?

**WJJ:** I wouldn’t expect it, so I wouldn’t miss it.

**INT:** Definitely a **delighter**, then. Last one: How do you feel about in-system peer reviews where members can rate events?

**WJJ:** That would be cool. It could help us improve future events. But again, not essential.

**INT:** Thanks, Jun Jie. That helps us classify which features are essential, expected, or exciting. One last question—do you have any features in mind that you feel are often overlooked?

**WJJ:** Hmm. I think renewal of memberships is often missed. Sometimes students don’t know their membership expired.

**INT:** Good point. So an auto-reminder or renewal system?

**WJJ:** Yes, that would be helpful.

**INT:** Excellent. Thanks again for your input—it’s been very insightful.

**WJJ:** You’re welcome! Looking forward to seeing the system in action.