APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE
e-ISSN: 2773-5141

AITCS

Vol. 7 No. 1 (2026) 488-507
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

SITARS: Leveraging Tagging-Based and Collaborative Filtering Recommendations in Student Interest Prediction

Piong Wei Xun[1], Rosmamalmi Mat Nawi[1]*

[1] *Faculty of Computer Science and Information Technology,*
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: rosmamalmi@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2026.07.01.026

Article Info

Received: 13 June 2025
Accepted: 18 June 2026
Available online: 30 June 2026

Keywords

Collaborative Filtering, Tagging-Based, Recommender System, Student Interest Tendencies, Education.

Abstract

In contemporary higher education, universities often struggle to engage students due to a lack of understanding of their diverse interests, leading to poor attendance at academic and extracurricular events. The Student Interest Tendencies and Recommendation System (SITARS) addresses this challenge through a web-based, data-driven framework that predicts student interests and enhances engagement. SITARS employs a hybrid recommendation approach combining Tagging-Based Recommendation, Item-Based Collaborative Filtering (IBCF), and User-Based Collaborative Filtering (UBCF), supported by Cosine Similarity and K-Nearest Neighbors (KNN). IBCF recommends events to students based on their preferences, while UBCF helps organizers identify students likely to be interested in specific events. By aligning event offerings with student interests across various domains, SITARS improves event relevance, planning efficiency, and resource allocation. Additionally, it provides valuable analytics for university decision-makers. Overall, SITARS promotes a more personalized and strategic approach to student engagement, ultimately fostering a more student-centered academic environment.

1. Introduction

The Student Interest Tendencies and Recommendation System (SITARS) is a web application system specifically developed for the Faculty of Computer Science and Information Technology (FSKTM) at Universiti Tun Hussein Onn Malaysia (UTHM). In today's rapidly evolving educational environment, universities face significant challenges in understanding and catering to the diverse interest tendencies of students, which often leads to low engagement and poor attendance in events and programs. Without a reliable and effective system for analyzing student preferences, event planning becomes inefficient, with universities resorting to generic, one-size-fits-all approaches that fail to resonate with students' interests. As a result, students are often overwhelmed with irrelevant information about various events, leading to disengagement and low participation. Student engagement is a critical determinant of academic success and overall educational experience in higher education institutions. According to student involvement theory for higher education, the learning and personal growth of students in an educational program increase as the quality and quantity of students' involvement increase [1]. Active participation in both academic and extracurricular activities has been shown to foster better employability and support personal development among university students [2]. However, universities continue to face significant challenges in effectively engaging their diverse student populations. Many institutional event planning efforts rely on generalized approaches that do not account for the varying interests, preferences, and behaviors of students. As a result, participation rates in organized activities often remain low, leading to underutilized resources and diminished student satisfaction.

SITARS is designed to address these challenges by leveraging data analytics algorithms to analyze collected data such as demographics, academic performance, historical event participation, and student feedback. By analyzing these data, SITARS can identify and predict student interest patterns, hence offering personalized event recommendations tailored to individual students, helping universities better understand and cater to their diverse interests. SITARS aims to optimize event planning, enhance resource allocation, and improve student engagement. By aligning events with students' evolving preferences, the system can increase attendance rates and reduce inefficiencies in event promotion. Ultimately, SITARS fosters a more dynamic, inclusive learning environment that enhances student satisfaction and retention by ensuring that future events are more accurately aligned with students' interests.

One of the aims of the proposed system, SITARS, is to design and develop a web application that predicts the interest tendencies of FSKTM students regarding events and programs. This system aims to enhance student participation by matching their preferences with tailored events and programs, ensuring greater engagement. Additionally, the project targets to evaluate the proposed system's functionality and usability to ensure its effectiveness in achieving stakeholders' requirements. The key stakeholders of this proposed system are FSKTM students, event organizers, and administrators. SITARS will incorporate advanced technologies like Tagging-Based Recommendation and Item-Based Collaborative Filtering (IB-CF) to recommend academic and extracurricular events tailored to students' preferences. It will include features for user management, data visualization, and reporting to support decision-making while ensuring responsiveness and operability across both mobile and web platforms.

SITARS is expected to yield significant improvements in student engagement by tailoring educational experiences to individual interests, fostering a deeper connection between students and their academic and extracurricular events. By providing accurate, data-driven recommendations that align with students' interests, learning goals, and professional aspirations, the system is expected to boost participation and enthusiasm in university programs. Personalized suggestions will encourage students to develop their skills confidently, fostering a proactive and enthusiastic approach to their educational journey.

2. Literature Review

This section discusses the literature review conducted for the proposed system. The purpose of this section is to provide a comprehensive overview of the system's development, beginning with a discussion of student interest tendencies in both academic and extracurricular activities. The section will then discuss the concept of the recommender system and the main technologies implemented. A critical analysis of existing related systems will also be included, resulting in a comparative evaluation between the existing solutions and the proposed system.

2.1 Domain Background: Student Interest Tendencies in Academic and Extracurricular

According to Munir and Zaheer [2], extracurricular activities are activities that students perform other than earning a degree during their education at a particular institute. There is no debate about the importance of extracurricular activities in a student's life. Understanding student interests in both academic and extracurricular activities is crucial for fostering an engaging educational environment. Engaged students are more likely to perform better and also develop enthusiasm for learning. However, current one-size-fits-all approaches to student recommendations often lead to irrelevant, inconsistent, and overwhelming suggestions, making it difficult to effectively guide students toward their interests. To address this, there is a need for a system that collects, integrates, and analyses student data in a standardized manner. This system would provide personalized, relevant, and timely recommendations, helping students explore new areas of interest, discover potential career paths, and enhance their skills. Furthermore, such a system would enable universities to track evolving student interests and make data-driven decisions to customize learning experiences. By aligning students with activities that match their strengths and aspirations, the system would contribute to a more personalized, engaging educational experience, promoting academic success, personal growth, and long-term career readiness.

2.2 Recommender System

A recommender system (RS) is a type of software or algorithm primarily powered by the technologies of artificial intelligence (AI) and machine learning (ML). It is specifically designed to suggest items or content to users based on their preferences, behaviors, historical interactions, cultural context, geolocation, or other factors. Basically, methods in RS can be generally categorized into three types: the Collaborative Filtering method (CF), the Content-Based method (CB), and the Hybrid method [3]. As defined by Milano et al. [4], RS is a class of algorithms designed to solve the recommendation problem by leveraging either of the RS methods to collect, curate, and process large amounts of personal information data to enhance individual experiences and social interactions. In RS, user and item profiles are established by analyzing user engagements with items in the system. This analysis examines the attributes of both the user and the item to identify the user's preferences [5].

The RS becomes an indispensable tool across a wide range of online applications by providing personalized content and services to overcome the challenges associated with information overload [6]. Nowadays, RS are widely used across various domains like e-commerce [7]–[9], social media [10], [11], tourism [12]–[14], and health applications [15], [16], providing tailored suggestions to help users navigate information overload and improve decision-making.

Figure 1 below briefly shows the common workflow of the recommendation system, explaining how the recommendation system works to offer users consistent, personalized and relevant suggestions. The workflow of a recommendation system involves collecting user data (preferences, ratings, demographics), item data (descriptions, categories, pricing), and interaction data. This data is cleaned, stored, and analyzed by predictive algorithms to identify patterns in user preferences. Once processed, the system generates personalized suggestions based on relevance and ratings, presenting them to the user. This helps optimize the user experience by offering consistent and relevant recommendations and enhancing engagement with digital platforms.

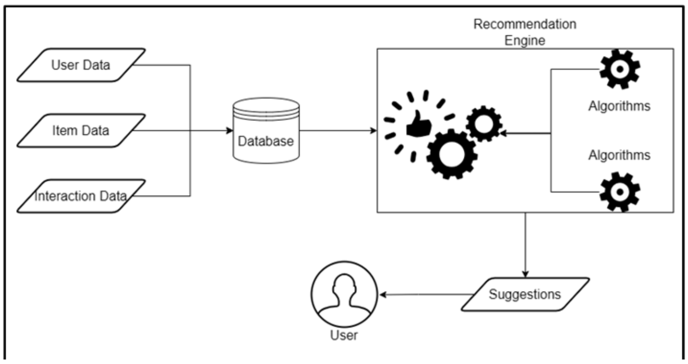


Fig. 1 Common Workflow of Recommender System

2.3 Main Technologies Implemented

The main technologies implemented are Tagging-Based Recommendation, and CF. The tagging-based recommendation generates personalized suggestions by associating content with relevant tags or keywords. According to Tang et al. [17], they identify three key factors that impact tag recommendation accuracy: sequential text modelling, tag correlation, and content-tag overlap. Meanwhile, the study of Zuo et al. [18] propose two-stage tag-aware multi-objective framework in deep learning for providing accurate and diversity recommendation. In parallel, C. Wang et al. [19] applied sentiment classification to tackle the issue of weak tagging data. Zhang et al. [20], on the other hand, leverage cross-domain recommendation with semantic correlation in tagging system. In the context of SITARS, the tagging-based recommendation module plays a critical role in aligning student preferences with relevant campus events. By matching the tags associated with upcoming events to those within a student's profile, SITARS can generate highly relevant recommendations that are more likely to resonate with individual interests.

On the other hand, CF relies on leveraging either user community opinions or past user behavior to predict items that would capture a user's interest [21]. It evaluates the degree of similarity in preferences between two users by examining their respective rating histories. Generally, CF recommendations algorithm can be subdivided into two primary classes: memory-based CF and model-based CF [22]. The IB-CF and user-based CF (UB-CF) are included in memory-based CF. According to Shetty et al. [23], IB-CF is an advanced recommendation method based on calculating item similarity scores from consumer interactions. Unlike UB-CF, which identifies users with similar preferences, IB-CF finds items similar to those a user has interacted with. It uses metrics like Cosine Similarity or Pearson Correlation to calculate item similarity. In this project, both IB-CF and UB-CF were chosen and assisted using Cosine Similarity and K-Nearest Neighbors (KNN) to provide accurate recommendations, even with limited user-event interaction data.

2.4 Reviewing of Existing System

In this section, the features and functionalities of the existing systems are thoroughly examined and compared with the proposed system to highlight similarities, differences, and areas for improvement. The similar existing systems that have been chosen to be reviewed due to their relevance in the realm of academic and extracurricular activities recommendation are Coursera [24], Swayam Central [25], and Eventbrite [26].

Coursera is an online learning platform offering diverse courses, certifications, and degrees in partnership with leading universities and institutions. It uses content-based and collaborative filtering algorithms to personalize learning based on user preferences, behavior, and contextual factors like time, pace, and regional trends. Peer reviews and social feedback further refine its adaptive recommendations, aligning with users' academic and career goals.

SWAYAM (Study Webs of Active Learning for Young Aspiring Minds) is India's national online education platform, offering free courses and workshops in collaboration with universities and educational institutions. Unlike Coursera's advanced personalization, SWAYAM's basic recommender system relies on enrolment data and search history, often using rule-based or general categorization, resulting in less tailored guidance.

Eventbrite is an event management and ticketing platform that recommends events based on user interactions, location, and preferences. While not focused on education, it offers contextual recommendations using trends and demographics. Unlike SWAYAM, Eventbrite allows users to create events without administrative approval, providing greater flexibility.

Table 1 compares Coursera, SWAYAM Central, Eventbrite, and the proposed SITARS system across features like recommendation types, personalization, AI/ML usage, platforms, user management, and security. While all offer academic recommendations, only Eventbrite and SITARS include extracurricular suggestions. Coursera and SITARS provide advanced personalization, whereas SWAYAM and Eventbrite offer basic recommendations. AI/ML is used across all systems, though limited in SWAYAM. SITARS is web-based, while others also support Android and iOS. All systems support user and admin management, search, analytics, and notifications; Coursera and SITARS also integrate feedback into recommendations. Coursera and Eventbrite have strong security due to payment features, while SWAYAM and SITARS offer moderate security. Gamification is present in all except Eventbrite to enhance user engagement.

Table 1 Comparison Between Existing Systems and Proposed System

Features/System	Coursera	Swayam Central	Eventbrite	Proposed System
Recommendation for academic activities	Yes	Yes	Yes	Yes
Recommendation for extracurricular activities	No	No	Yes	Yes
Personalization	High	Basic	Basic	High
AI/ML technology	Yes	Limited	Yes	Yes
Web-based or mobile application	Both	Both	Both	Web-based
User Management	Yes	Yes	Yes	Yes
Search and filter	Yes	Yes	Yes	Yes
Report and analytics	Yes	Yes	Yes	Yes
Notification	Yes	Yes	Yes	Yes
Feedback and evaluation	Yes	No	No	Yes
Security and privacy	High	Moderate	High	Moderate
Gamification	Yes	Yes	No	Yes

3. Methodology

This section outlines the methodology used to develop SITARS. According to Dennis et al. [27], the System Development Life Cycle (SDLC) describes how an Information System (IS) can meet business needs, be designed, developed, and handed over to users. To achieve the project's goals, the Prototyping Model was chosen for its clear phases, which enhance efficiency.

3.1 Prototype Model

The Prototyping Model is a widely used SDLC approach that involves designing, developing, testing, and refining an initial system prototype through repeated iterations based on stakeholder feedback and evolving requirements. This model emphasizes building a quick prototype, even if it lacks full functionality, and relies on stakeholder engagement to enhance the system and ensure it aligns with their needs. It is particularly suitable for projects with unclear, changing, or evolving requirements. By fostering strong stakeholder involvement, the Prototyping Model clarifies project requirements and helps create a user-friendly product.

The Prototyping Model, as described by Dennis et al. [27], begins with planning and analysis to gather requirements and define objectives. A quick prototype is then developed and iteratively refined based on stakeholder feedback. The planning phase sets goals and timelines, while the analysis phase captures functional and non-functional requirements and evaluates the current system. The design phase produces wireframes and a database schema, followed by implementation using a web-based technology stack comprising front-end, back-end, and database components, further incorporating tagging-based and CF recommendation algorithms. After iterative refinements, it undergoes Functional Testing and User Acceptance Testing (UAT) before deployment.

3.2 SITARS Framework

The SITARS framework is structured into five distinct yet interconnected phases to ensure personalized and accurate event recommendation.

In the initial phase that serves as the foundational stage, core data which is 'Student Data', 'Event Data', and 'Interaction History' will be collected from institutional databases and surveys. This includes demographic information, academic records, historical attendance data, and feedback from past events. Additional data such as students declared interests or participation in clubs and online engagement (e.g., rating) may also be gathered to enrich the dataset. The collected data will undergo preprocessing, including cleaning, normalization, and transformation, to ensure consistency and readiness for analysis.

The second phase involves designing the architecture of the SITARS platform. The process of extracting and mapping the data will be done in this phase wherein relevant tags and categories are extracted from the collected data. The third phase calculates the Cosine Similarity scores with KNN-assisted filtering based on user-item interaction patterns, identifying similar users (UB-CF) and events (IB-CF) to generate relevant personalized recommendations.

Event Recommendation (Phase 4) integrates the outputs from both Tagging-Based and CF approaches. The system will analyze student interest patterns by tagging event-related data and matching it with student profiles. The IB-CF will identify similarities between events based on student interaction patterns and recommend events similar to those the student has previously shown interest in. Meanwhile, the UB-CF will identify students with similar interests or behavior patterns and recommend event organizers with a list of potential students, thereby ensuring double side recommendations for students and also event organizers. In this context, the system will generate personalized event recommendations for the student, and their engagement and interaction will be tracked. Feedback will be collected through surveys and system usage logs to assess the accuracy, usability, and relevance of the recommendations.

Finally, Phase 5: Accuracy Enhancement focuses on iterative improvement. It captures student feedback and interaction data post-recommendation, which is then utilized to refine the recommendation algorithms, ensuring the system evolves in response to user behaviors and maintains high relevance over time. Figure 2 below represents the research framework that will be utilized in this study.

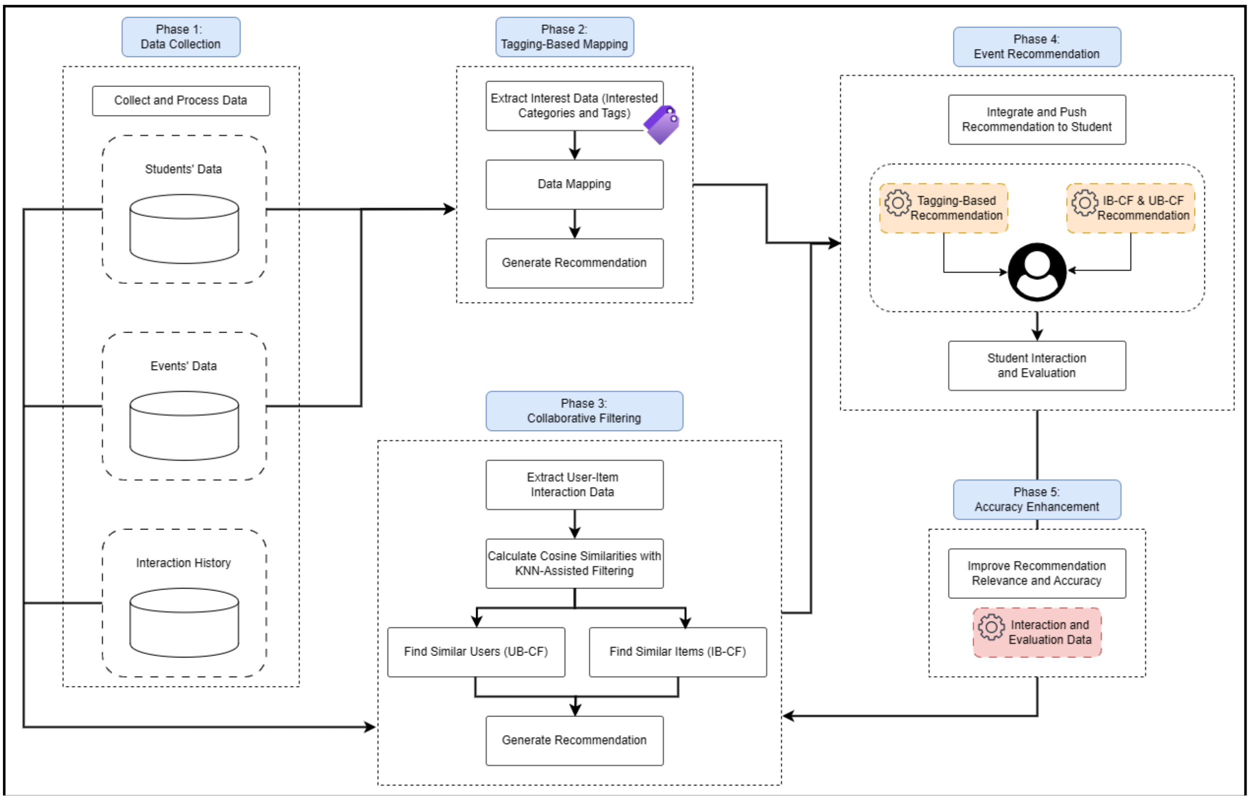


Fig. 2 Framework of the SITARS

4. System Analysis and Design

This section presents the system analysis and design of SITARS, including both functional and non-functional requirements. System features are illustrated using Data Flow Diagrams (DFD), and an Entity Relationship Diagram (ERD). An algorithm is also provided to describe the key steps of the predictive analytics and recommendation module.

4.1 System Requirements Analysis

System Requirement Analysis is a crucial process of the SDLC analysis phase, focusing on identifying and documenting functional and non-functional requirements. This ensures a shared understanding of project needs between stakeholders and the development team. Tables 2 and 3 outline these requirements, detailing what the system should do and how well it should perform.

Table 2 Functional Requirements of the Proposed System

Modules	Description
Data Collection and Integration Module	• The system should provide a collection of forms implemented at different parts of the system to collect various types of student-related data. • The system should consist of backend logic to accurately handle, standardize, and store these data in the database based on user input and predefined data format.
Predictive Analytics and Recommendation Module	• The system should be able to analyze interest tendencies and predict future preference patterns according to the data collected. • The system must be able to provide precise personalized recommendations for both academic and extracurricular events.
Feedback and Evaluation Module	• The system should enable users (students and event organizers) to provide feedback regarding the academic and extracurricular events participated in or organized. • The system should enable users to evaluate the recommendation provided to enhance the accuracy of predictive algorithms.
Data Visualization and Reporting Module	• The system should visualize the data in terms of charts, tables, and reports, and enable exporting based on filtration
User Management Module	• The system should enable users to log in with valid credentials. • The system should enable users and higher-permission-level admins to manage their profiles. • The system should enable admins to import user lists for multiple account creation.
Event and Program Management Module	• The system should enable event organizers and admins to manage academic and extracurricular events. • The system should enable event categorization and tagging.
Notification Module	• The system should send personalized notifications via email and the system portal regarding the event suggestion. • The system should allow admins and event organizers to send notifications for event invitations, confirmation, approval and rejection.
Security and Privacy Module	• The system should apply role-based access controls to differentiate normal users, event organizers and admins. • The system should apply hashing and encryption technology to ensure data security for certain sensitive information.

Table 3 Non-Functional Requirements of the Proposed System

Requirements	Description
Usability	• The system should have a user-friendly and intuitive UI. • The system should be responsive across different operating systems, devices, browsers, and screen sizes.
Performance	• The system should be available to be used at any time except during scheduled maintenance periods. • The system should be able to provide responses to users' requests within a few seconds. • The system should handle peak usage times without significant degradation in performance.

Table 3 (Cont.)

Requirements	Description
Operational	• The system should routinely back up data to prevent data loss in the event of failure. • The system should log clearly in detail for further troubleshooting, auditing, and improving operational performance.
Security	• The system should implement strong authorization and authentication mechanisms. • The system should have role-based access controls to ensure users have access only to the information they need. • The system should hash or encrypt sensitive information.

4.2 Context Diagram

A Context Diagram (CD) is a simplified Data Flow Diagram (DFD) that represents the entire system as a single process, focusing on its interactions with external entities. It outlines data flows between the system and its environment, clearly defining system boundaries without detailing internal processes. Appendix A illustrates the CD for SITARS, showing its interactions with external entities such as FSKTM students, event organizers, and administrators.

4.3 Data Flow Diagram Level 0

A Data Flow Diagram (DFD) is a type of graphical representation of the flow of data within a system application, demonstrating the transformation of input into output via processes, data stores, and interactions with external entities. It helps stakeholders and developers better understand, analyze, and design system functionality. In this context, the DFD Level 0 offers a high-level overview of the system's workflow, showing how data flows from three external entities through twelve core processes and into seven data stores. The complete DFD Level 0 is provided in Appendix B.

4.4 Entity Relationship Diagram

The ERD of the proposed system visualizes the relationships between entity sets stored in the database, playing a key role in efficient database design and meeting the system's requirements. Appendix C shows the entities involved, including seven tables that outline the attributes and relationships of each entity.

4.5 Algorithm

The algorithm as in Figure 3 outlines the main steps and functionalities of the 'Predictive Analytics and Recommendation' module.

Algorithm

```
Input: Students' Data, Events' Data, Interaction History
Output: Personalised Recommendations
START
    1. Collect and preprocess Students' Data, Events' Data, and
       Interaction History.
    2. Store the processed data in a centralised database for further
       analysis.
    3. Extract student interests (categories and tags) from the
       collected data.
    4. Use JSON mapping to map student-event data based on the identified
       categories and tags.
    5. Generate preliminary recommendations using the tagging-based
       mapping approach.
    6. Extract student-event (user-item) interaction data (e.g.,
       historical attendance record, rating, feedback) from different
       tables into an empty dataset.
    7. Calculate the cosine similarities for students (UB-CF) and events
       (IB-CF) using K-Nearest Neighbors assisted filtering.
    8. Generate additional recommendations using the collaborative
       filtering (CF) approaches (e.g., IB-CF for event recommendation
       to students and UB-CF for student recommendation to event
       organizer).
    9. Integrate recommendations from both methods and push to students
       and event organizers.
    10. Collect student interaction and evaluation data to refine the
        recommendation model.
    11. Update the algorithms to improve the relevance and accuracy of
        recommendations.
END
```

Fig. 3 Algorithm of Predictive Analytics and Recommendation Module

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 488-507

The algorithm used in the Predictive Analytics and Recommendation module is designed to deliver personalized event recommendations to students by combining tagging-based filtering and CF techniques. It begins with the collection and preprocessing of three types of data: students' information, events' information, and student-event interaction data such as attendance and ratings. These data are cleaned and structured before being stored in the database to enable consistent and scalable access. Next, the system extracts students' interests from interest categories and tags. These extracted interest data are then mapped to relevant event data using a JSON-based structure, allowing for the generation of initial recommendations through a tagging-based matching process. A basic score-based ranking mechanism is implemented to ensure the relevance between student interest and event data. This early-stage recommendation serves as a solution to the cold-start problem, ensuring that new users can still receive relevant suggestions based on their stated interests.

To provide double-sided recommendations, the system also extracts student-event interaction data and calculates item similarities using Cosine Similarity. This enables the use of CF, particularly IB-CF to suggest similar events, and UB-CF to indicate a list of potential students with similar preferences. Recommendations generated by both the tagging and CF methods are integrated to provide more accurate and diverse suggestions. Student interactions and evaluations of the recommended events are continuously collected to further refine the model. Over time, the system updates the recommendation algorithms based on this feedback to improve the accuracy, relevance, and responsiveness of future recommendations. This hybrid design was chosen because it balances the strengths of both content-based and behavior-based systems, thereby addressing limitations such as data sparsity and lack of historical user behavior.

5. Results and Discussion

SITARS combines modern web technologies and machine learning to deliver an intelligent, user-friendly platform. The web interface, built with Bootstrap 5, ensures responsiveness and a clean design, while DOM (Document Object Model) manipulation enhances interactivity. On the back end, plain PHP handles core logic for simplicity and maintainability. Machine learning functionalities are integrated through Rubix ML, a PHP-based library that enables advanced algorithms such as IB-CF and UB-CF via the calculation of Cosine Similarities using KNN-assisted filtering, allowing for accurate and personalized event suggestions. MySQL serves as the relational database for managing student data, event details, and recommendations. This technology stack creates an efficient and scalable system that drives personalized student engagement at UTHM.

5.1 System Implementation

This section provides a detailed overview of the most important pages and features of the SITARS system, with a focus on the core components that drive its functionality. It will showcase the essential pages that users interact with, offering insight into the system's design and how these elements work together to support its objectives. The intention is to highlight the critical aspects of the system's implementation, ensuring that key functionalities are clearly understood without delving into less impactful details.

Figure 4 below shows the "Discover Events" page of the SITARS system. This page allows users, especially students, to explore various events, categorized and tagged for easy navigation. Each event card includes key details like categories, tags, pax number, fee, registration deadline, and event duration. There is a search bar with advanced filters for users to filter events based on the attributes. Meanwhile, users can decide to click on 'View' to check the event details and 'Join' to join that particular event.

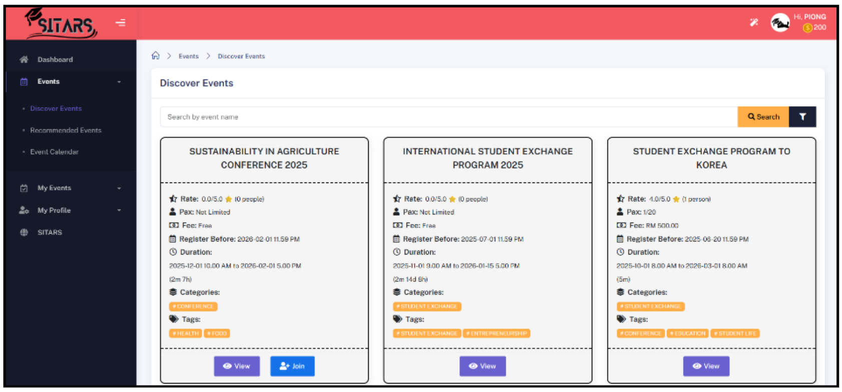
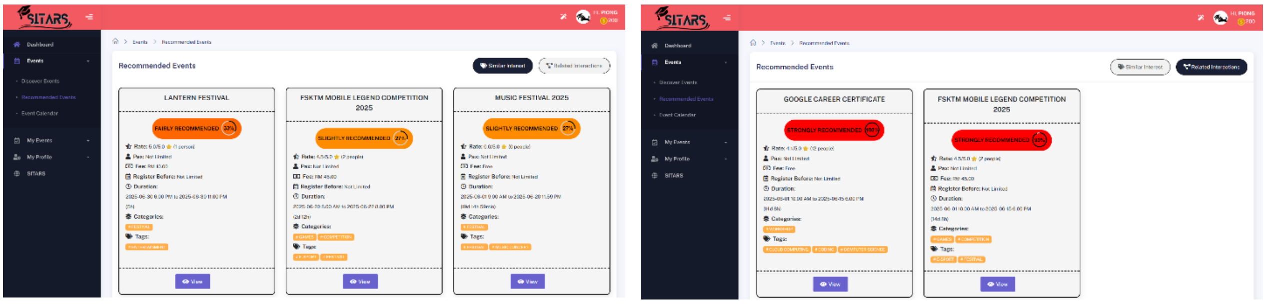


Fig. 4 User Interface of Discover Events

Figure 5 below shows the "Recommended Events" page of the SITARS system. This page allows students to view a personalized list of events that are recommended based on their interests, preferences, and past activities.

The system analyzes user data and hence implements tagging-based and IB-CF recommendation approaches to suggest relevant events, ensuring that students have easy access to opportunities that align with their academic and extracurricular goals. By providing these tailored recommendations, the "Recommended Events" page helps students stay informed about upcoming activities that enhance their learning and campus experience. While in Figure 6, the diagram shows a segment of the “Event Details” page, which this segment displays a list of similar events and potentially liked events relevant to the event selected.



(a) (b)

Fig. 5 User Interface of Recommended Events

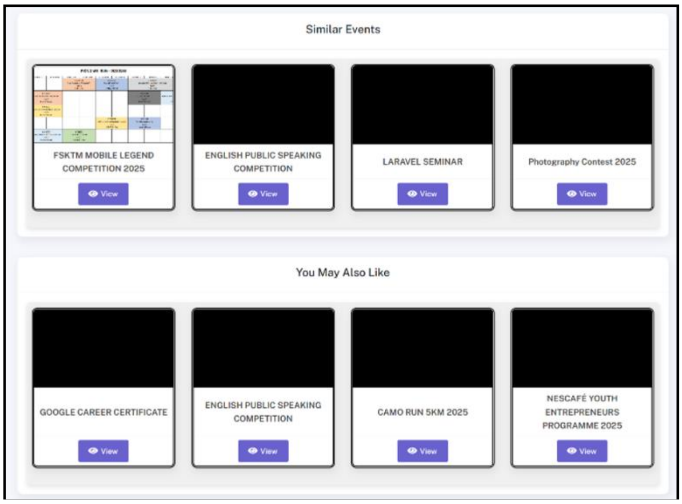


Fig. 6 User Interface Segment of Event Details

The algorithm illustrated in Figure 7 outlines a hybrid approach to generating personalized event recommendations for students by utilizing their data, event details, and student-event interaction data. It begins by loading relevant datasets and extracting student interest data from interest categories and tags. These interest data are then mapped to event tags using a structured JSON mapping method, prioritizing the categories based on relevance. This process enables the system to generate Recommendation 1 through a tagging-based mapping approach, which is particularly effective in addressing the cold-start problem by providing relevant suggestions based on explicit user interests.

To complement this, the algorithm also analyzes student-event interaction data to capture behavioral preferences. A user-item interaction matrix is constructed, and Cosine Similarity is used to calculate how closely related events are, based on past student engagement. The similarity matrix is then processed using a KNN-assisted IB-CF technique to generate Recommendation 2, which recommends events similar to those the student has already interacted with. This method allows the system to identify patterns and make informed suggestions even when explicit interest tags are insufficient.

Both recommendations are presented to students using navigation pills in the user interface, as shown in Figure 5, allowing users to easily toggle between interest-based and behavior-based suggestions. Additionally, the tagging-based recommendation strategy is used in Figure 6 to generate events similar in interest categories and tags, while the IB-CF algorithm is applied to discover events a student might like based on similar user behavior. This hybrid recommendation model was chosen to ensure higher accuracy, flexibility, and personalization by balancing explicit preferences with inferred behavior, ultimately enhancing the user experience.

```
Algorithm
Input: Students' Data, Events' Data, Interaction History
Output: Personalised Event Recommendations for Student
START
    1. Load the students' data, events' data, and student-event
       interaction data.
    2. Extract student interests (categories and tags) from the loaded
       data.
    3. Use JSON mapping to map student interest data (the identified
       categories and tags) to events' categories and tags, prioritising
       the categories and tags according to their order.
    4. Generate Recommendation 1 using the tagging-based mapping
       approach.
    5. Extract student-event (user-item) interaction data and compute a
       similarity score matrix using the Cosine Similarity measure.
    6. Process the similarity matrix using KNN-assisted filtering to
       generate Recommendation 2 based on the IB-CF concept.
    7. Push both event recommendations to students, separating them
       using navigation pills.
END
```

Fig. 7 Algorithm to Generate Personalized Event Recommendations

Figure 8 shows a segment of the "Event Participants List" page in the "Event Management" section, which is exclusively accessible to administrators and event organizers but not students. This segment provides a comprehensive list of potential students who may be interested in attending the event. Event organizers can review the list, filter potential participants based on tagging-based and UB-CF recommendations, and then send personalized invitations to selected students. By offering a tailored selection of participants, this feature enables organizers to manage attendance more effectively and ensure that the event reaches the right audience.

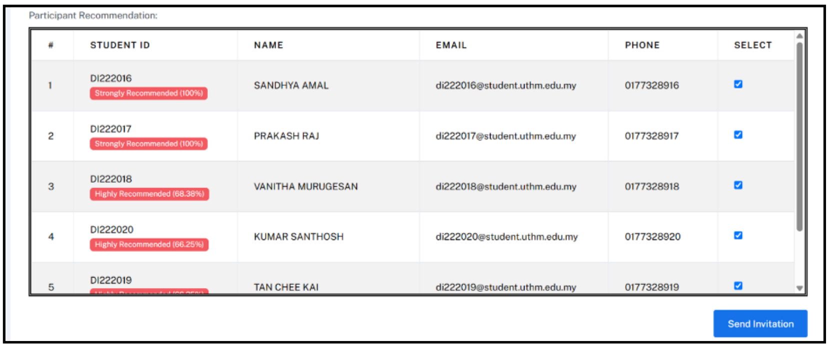


Fig. 8 User Interface Segment of Event Participants List

The algorithm depicted in Figure 9 is tailored to support event organizers by providing personalized student recommendations for their events. This reverse recommendation process begins by loading essential datasets, including student profiles, event details, and interaction histories between students and events. The system then extracts student interest information, which serves as a basis for matching students to relevant events. Through a structured JSON mapping approach, these interests are aligned with event categories and tags, prioritizing them according to their relevance or order. This step enables the generation of Recommendation 1 using a tagging-based method, which efficiently identifies students whose stated interests align with specific event themes or topics.

To further refine these matches and capture implicit preferences, the algorithm proceeds to analyze the user-item interaction data. It constructs a similarity matrix using Cosine Similarity, which measures the closeness between students based on shared interaction patterns. Leveraging this matrix, the system applies a KNN-assisted UB-CF technique to produce Recommendation 2. This step identifies students with similar historical behavior to those who have shown interest in the event, enabling more dynamic and behavior-driven targeting. The inclusion of UB-CF ensures the system goes beyond static interest data to capture student preferences based on actual interactions.

The final stage of the algorithm involves compiling a list of potential student candidates generated from both the tagging-based and UB-CF approaches. These two sets of recommendations are originally planned to be visually separated using navigation pills in the user interface, making it easier for event organizers to explore and act upon both interest-based and behavior-based student suggestions. However, after evaluating the real-case effectiveness, the UI had changed to combine the results in one table, as shown in Figure 9. This hybrid approach was selected to balance explicit user interests with behavioral insights, resulting in more accurate and personalized outreach for event promotion. By combining both content and collaborative filtering techniques, the algorithm increases the likelihood of student engagement and ensures that event organizers can effectively reach the right audience.

Algorithm

```
Input: Students' Data, Events' Data, Interaction History
Output: Personalised Student Recommendations for Event
START
   1. Load the students' data, events' data, and student-event
      interaction data.
   2. Extract student interests (categories and tags) from the loaded
      data.
   3. Use JSON mapping to map student interest data (the identified
      categories and tags) to events' categories and tags, prioritising
      the categories and tags according to their order.
   4. Generate Recommendation 1 using the tagging-based mapping
      approach.
   5. Extract student-event (user-item) interaction data and compute a
      similarity score matrix using the Cosine Similarity measure.
   6. Process the similarity matrix using KNN-assisted filtering to
      generate Recommendation 2 based on the UB-CF concept.
   7. Generate a list of potential students based on both student
      recommendations to event organizers, separating them using
      navigation pills.
END
```

Fig. 9 Algorithm to Generate Personalized Student Recommendations

Figure 10, 11, 12 shows the segments of the "User Profile" page. These segments play critical roles in SITARS, as they recorded the students' interested categories and tags. By recording these details, the system builds a foundational database that feeds into the recommendation module. This allows SITARS to suggest events, activities, and resources that are tailored to each student's unique profile. The information stored on the "User Profile" page not only helps to personalize the user experience but also ensures that the recommendations provided are relevant and aligned with individual preferences, thereby enhancing student engagement and participation.

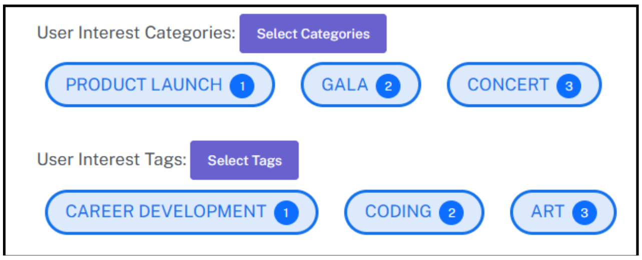


Fig. 10 User Interface Segment of User Profile (1)

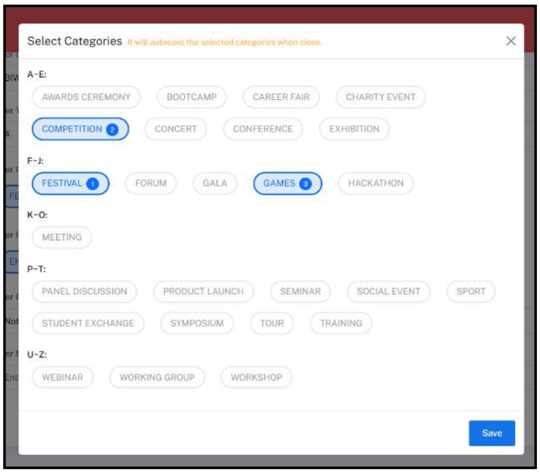


Fig. 11 User Interface Segment of User Profile (2)

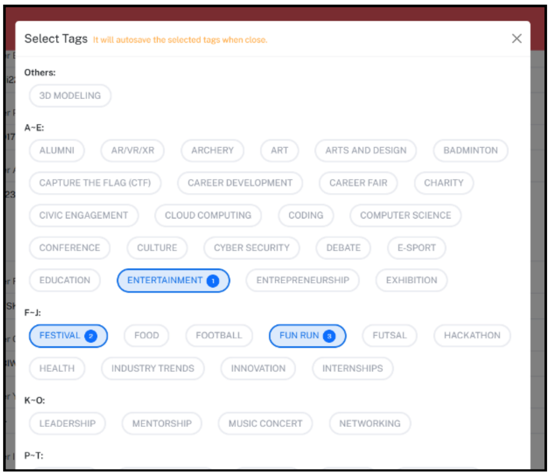


Fig. 12 User Interface Segment of User Profile (3)

Figure 13 illustrates the "Event List" page, designed exclusively for admin users. This page provides an organized overview of all events within the system, displayed in a tabular format. It includes key details such as event names, categories, dates, and statuses, along with action buttons for managing events (e.g., editing, deleting, or viewing detailed information). The interface is streamlined to support efficient event management, ensuring admins can oversee and maintain event data effectively.

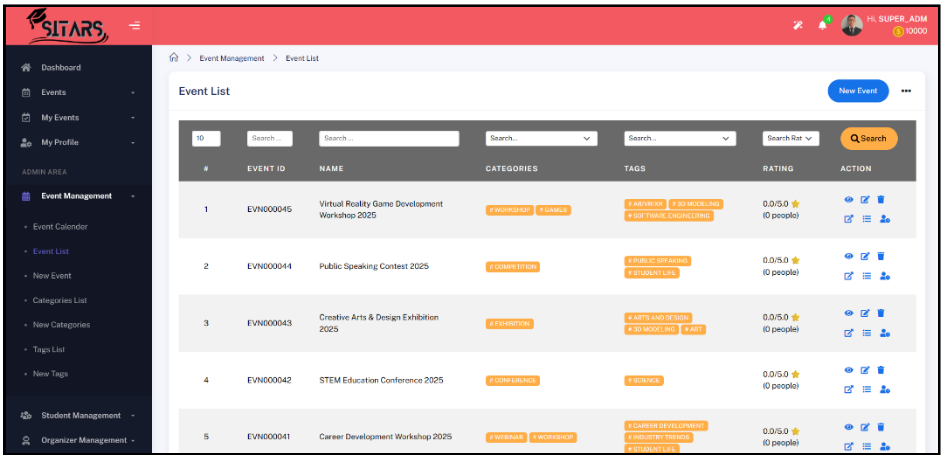


Fig. 13 User Interface of Event List

Figure 14 displays the Event Attendance page, where organizers and administrators can mark attendance, update participant statuses, and manage attendance history for each event, helping ensure accurate tracking of participation. Figure 15 shows a QR code that students can scan upon arrival for a quick and efficient check-in. This QR-based system automates the attendance process, reducing manual effort and improving overall event management.

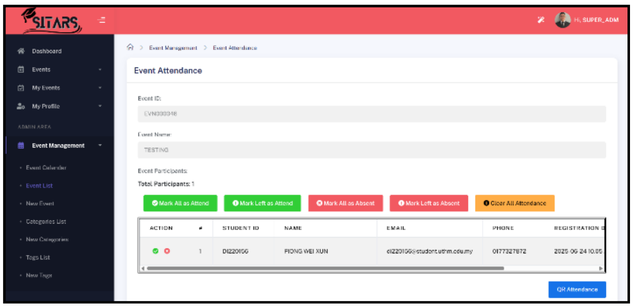


Fig. 14 User Interface Segment of Categories List

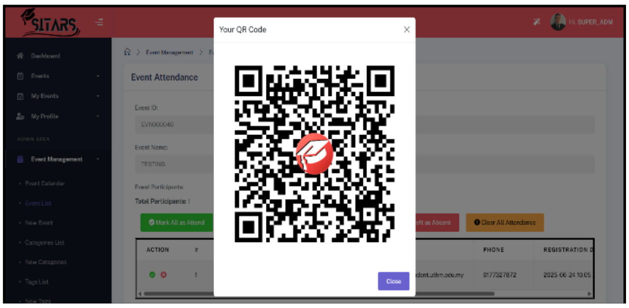


Fig. 15 User Interface Segment of Tags List

Figures 16 and 17 illustrate the segments of " Categories List" and "Tags List" pages, respectively, which are designed to help administrators efficiently organize and maintain the metadata used throughout the system. The "Categories List" page allows admins to create, update, or remove categories, providing a structured classification system for events. Similarly, the "Tags List" page enables admins to add, edit, delete, or view tags, ensuring events are accurately labelled for better searchability and relevance to users. Together, these pages streamline the management of tags and categories, enhancing event organization and user experience.

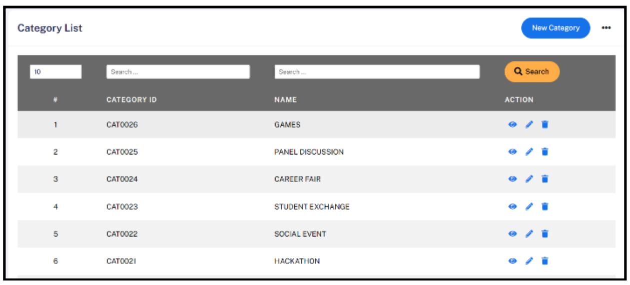


Fig. 16 User Interface Segment of Categories List

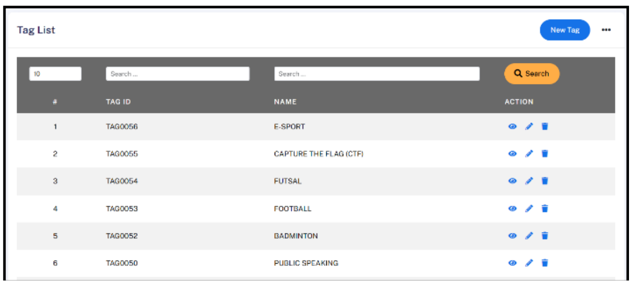


Fig. 17 User Interface Segment of Tags List

5.2 Testing

The system's functional modules are tested using a structured test plan, as detailed in the tables below, to ensure that all features operate correctly and align with the specified requirements. Each module of SITARS is evaluated through functional and integration testing, covering various input scenarios, expected outputs, and system behaviors. The test plan includes specific test cases that assess data collection accuracy, recommendation effectiveness, user feedback handling, data visualization, user and event management, notification delivery, and system security. This comprehensive testing process ensures the system's reliability, accuracy, and readiness for real-world deployment. Table 4 below shows the test case result of SITARS.

Table 4 Test Case Result of SITARS

Test Case	Expected Outcome	Actual Outcome	Status
Data Collection and Integration Module			
Validate student data form submission	Data saved to database in correct format	As expected outcome	Pass
Handle incomplete form submissions	Show validation errors, no data saved	As expected outcome	Pass
Validate error handling for invalid input data formats	Error message displayed, data not submitted	As expected outcome	Pass
Backend data standardization	Data normalized and stored correctly	As expected outcome	Pass
Data integrity check	No duplication or data loss	As expected outcome	Pass
Predictive Analytics and Recommendation Module			
Generate personalized recommendations	Relevant academic or extracurricular suggestions displayed	As expected outcome	Pass
Generate potential student list for events	A list of potential students displayed for event organizer	As expected outcome	Pass
Update event recommendations based on new data	Event recommendations adjusted accordingly to the new data	As expected outcome	Pass
Feedback and Evaluation Module			
Submit event rating and comment	Feedback stored and linked to event	As expected outcome	Pass
Evaluate the accuracy of recommendation	Evaluation stored for analytics	As expected outcome	Pass
Data Visualization and Reporting Module			
Generate chart view of participation data	Correct chart displayed	As expected outcome	Pass
Export reports in Excel/PDF	Report downloaded successfully	As expected outcome	Pass
User Management Module			
Login with valid credentials	Redirect to dashboard	As expected outcome	Pass
Edit user profile	Changes saved and reflected	As expected outcome	Pass
Student, organizer and admin account management	Manage account successfully	As expected outcome	Pass
Event and Program Management Module			
Create new event	Event listed and stored	As expected outcome	Pass
Tag and categorize event	Tags and categories applied and used in filtering	As expected outcome	Pass
Edit or delete event	Event updated or removed	As expected outcome	Pass
Notification Module			
Push notification to users	Notification email able to be sent to users	As expected outcome	Pass
Manual invitation emails	Invitations received successfully	As expected outcome	Pass

Table 4 (Cont.)

Test Case	Expected Outcome	Actual Outcome	Status
Security and Privacy Module			
Role-based access control	Appropriate access level enforced	As expected outcome	Pass
Hash sensitive data	Data stored as hashed value	As expected outcome	Pass
Prevent unauthorized data access	Access denied	As expected outcome	Pass

5.3 User Acceptance Testing

UAT is the final phase of testing where the end users or stakeholders validate that the system meets their requirements and is ready for production. During UAT, users perform real-world scenarios to ensure the application functions as expected in the business context. This phase helps confirm that the software is intuitive, user-friendly, and fulfills the intended goals before being officially deployed. Successful UAT indicates that the system is ready for release, with any critical issues addressed.

The bar chart as in Figure 18 illustrates the results of functionality testing, showing a strong positive reception across all tested aspects. For each of the five functionality statements, a significant majority of respondents either "Agree" or "Strongly Agree," with "Strongly Agree" consistently being the most chosen option, indicating high satisfaction with the system's core functions. Only a small number of users selected "Neutral," and virtually none chose "Disagree" or "Strongly Disagree.

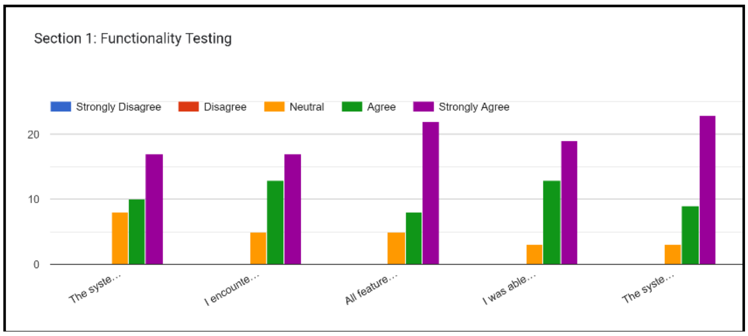


Fig. 18 User Satisfaction on System Functionality

The bar chart as in Figure 19 presents the feedback on the system's usability and user interface. The results clearly indicate high user satisfaction, with the overwhelming majority of responses falling into the "Agree" and "Strongly Agree" categories for statements regarding ease of use, layout, satisfaction, button functionality, and overall system usability. "Strongly Agree" is the dominant response for most questions, suggesting a very positive user experience.

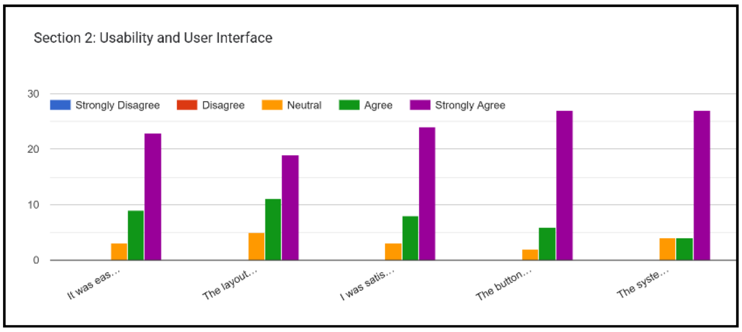


Fig. 19 User Satisfaction on System Usability and User Interface

The bar chart as in Figure 20 summarizes user feedback on the system's performance and feedback mechanisms. Across all five statements, the responses are predominantly positive, with a large proportion of users selecting "Agree" or "Strongly Agree." While "Strongly Agree" is a leading choice for the overall system and system feedback, there's a slightly higher presence of "Neutral" and "Agree" for some questions, indicating good but perhaps not universally outstanding performance or feedback aspects.

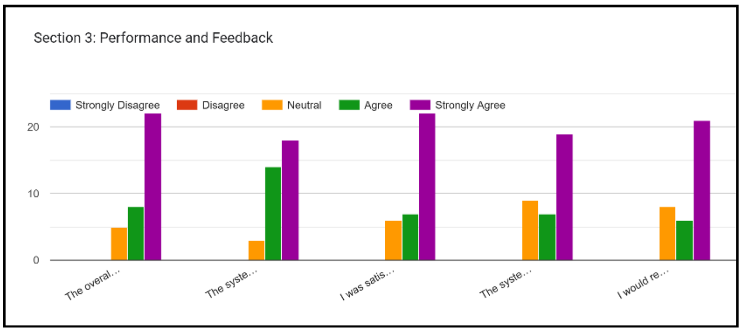


Fig. 20 User Satisfaction on System Performance and Overall Feedback

The pie chart in Figure 21 displays the overall approval for the system's release based on 35 responses. A substantial 80% of respondents approve of the system for release, while 20% indicated "Maybe," suggesting some areas that might warrant further consideration or minor adjustments before full deployment. No respondents selected "No.".

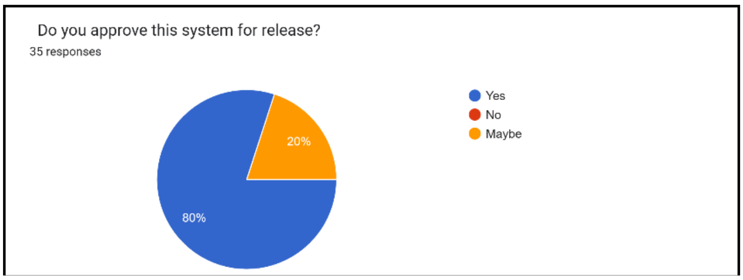
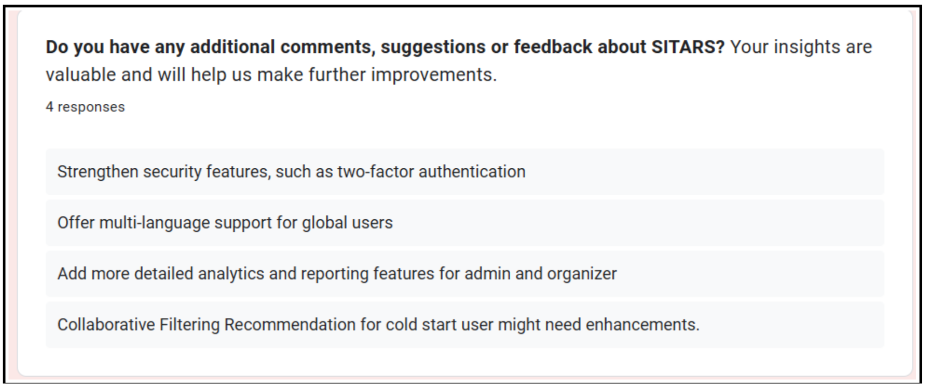


Fig. 21 User Approval for System's Release

Figure 22 presents four qualitative responses providing additional comments and suggestions for SITARS. The feedback highlights specific areas for improvement, including strengthening security features (e.g., two-factor authentication), offering multi-language support, adding more detailed analytics and reporting features for administrators and organizers, and enhancing collaborative filtering recommendations for cold start users.


Do you have any additional comments, suggestions or feedback about SITARS? Your insights are valuable and will help us make further improvements.

4 responses

Strengthen security features, such as two-factor authentication

Offer multi-language support for global users

Add more detailed analytics and reporting features for admin and organizer

Collaborative Filtering Recommendation for cold start user might need enhancements.

Fig. 22 User Additional Comments, Suggestions, and Feedback

The UAT for the SITARS demonstrates its strong effectiveness, with results indicating high user satisfaction across functionality, usability, and performance. Users overwhelmingly approved the system's core functions, found its interface intuitive and visually appealing, and were satisfied with its speed and responsiveness. While the majority of users approved SITARS for release, some provided valuable qualitative feedback, suggesting enhancements in particular areas, all of which would be taken into consideration to further refine the system and cater to a broader range of user needs before its official launch.

6. Conclusion

The Student Interest Tendencies and Recommendation System (SITARS) represents a significant advancement in personalized student engagement within academic environments. Designed as a data-driven platform, SITARS effectively analyzes student preferences, behavioral patterns, and historical participation to generate tailored academic and extracurricular event recommendations. By leveraging advanced recommendation algorithms such as Tagging-Based Recommendations, IB-CF, and UB-CF, the system delivers highly accurate and relevant suggestions that enrich the student experience and promote greater involvement in campus activities.

Beyond its role as a recommendation engine, SITARS serves as a powerful administrative and analytical tool for institutions like the Faculty of Computer Science and Information Technology (FSKTM) at Universiti Tun Hussein Onn Malaysia (UTHM). It provides event organizers and administrators with actionable insights, streamlines the event management processes, and supports strategic planning through data-informed decision-making. The system's dual-sided recommendation model benefits both students and organizers, fostering meaningful connections and improving event outcomes.

Despite its strengths, SITARS has certain limitations, including reliance on sufficient user data for effective recommendations, computational demands as the system scales, and the need for greater flexibility to accommodate diverse institutional needs. These challenges highlight areas for ongoing refinement and innovation.

Looking ahead, future enhancements for SITARS could include the implementation of stronger security measures like two-factor authentication, the introduction of multi-language support, and the development of a mobile application to improve accessibility. Expanding the analytics capabilities and improving recommendations for cold-start users through hybrid or content-based filtering approaches would further increase the system's robustness and appeal. Additionally, integration with academic advising and career development platforms could extend SITARS's impact, aligning students' extracurricular engagement with their academic and professional goals.

In summary, SITARS has demonstrated its potential as a comprehensive platform that not only personalizes student experiences but also enhances institutional effectiveness. With continuous development and user-focused improvements, it is well-positioned to evolve into a holistic student engagement system that significantly contributes to a more dynamic, inclusive, and data-driven educational ecosystem at UTHM.

Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: **study conception and design:** Piong Wei Xun, Rosmamalmi Mat Nawi; **data collection:** Piong Wei Xun; **analysis and interpretation of results:** Piong Wei Xun, Rosmamalmi Mat Nawi; **draft manuscript preparation:** Piong Wei Xun, Rosmamalmi Mat Nawi. All authors reviewed the results and approved the final version of the manuscript.

References

[1] A. W. Astin, "Student Involvement: A Development Theory for Higher Education," *Journal of College Student Personnel*, vol. 25, no. 4, pp. 297–308, 1984. [Online]. Available: https://www.researchgate.net/publication/220017441

[2] S. Munir and M. Zaheer, "The role of extra-curricular activities in increasing student engagement," *Asian Assoc. Open Univ. J.*, vol. 16, no. 3, pp. 241–254, 2021. [Online]. Available: https://doi.org/10.1108/AAOUJ-08-2021-0080

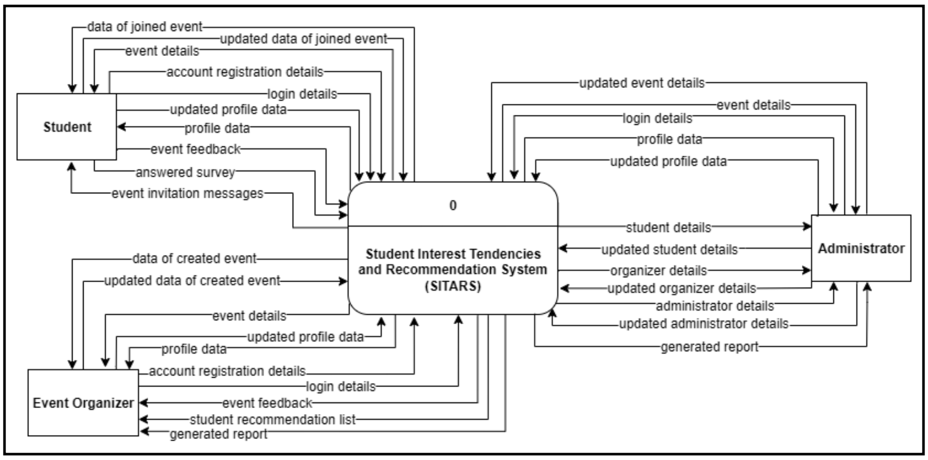
[3] R. M. Nawi, S. A. M. Noah, and L. Q. Zakaria, "Integration of Linked Open Data in Collaborative Group Recommender Systems," *IEEE Access*, vol. 9, pp. 150753–150767, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3124939

[4] S. Milano, M. Taddeo, and L. Floridi, "Recommender systems and their ethical challenges," *AI and Society*, vol. 35, no. 4, pp. 957–967, 2020. [Online]. Available: https://doi.org/10.1007/s00146-020-00950-y

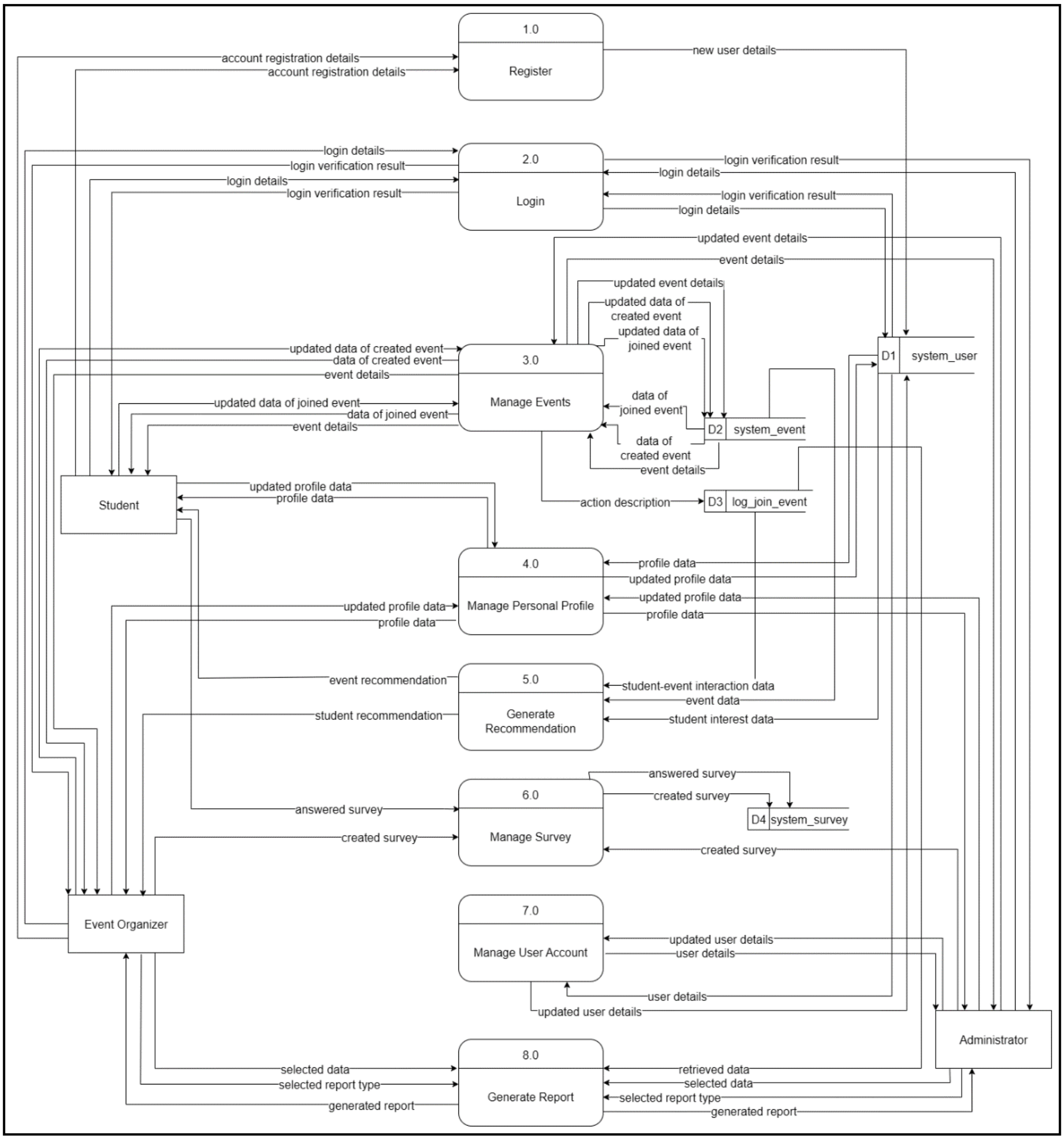
[5] T. M. A. U. Gunathilaka et al., "Addressing sparse data challenges in recommendation systems: A systematic review of rating estimation using sparse rating data and profile enrichment techniques," *Intell. Syst. with Appl.*, vol. 25, 2025. [Online]. Available: https://doi.org/10.1016/j.iswa.2024.200474

[6] Z. Zhao et al., "Recommender Systems in the Era of Large Language Models (LLMs)," *IEEE Trans. Knowl. Data Eng.*, 2024. [Online]. Available: https://doi.org/10.1109/TKDE.2024.3392335
[7] P. Chu, Y. Mao, S. Lee, and C. Hou, "Leveraging User Comments for Recommendation in E-Commerce," *Applied Sciences*, vol. 10, p. 2540, 2020. [Online]. Available: https://doi.org/10.3390/app10072540
[8] D. N. Nguyen, V. H. Nguyen, T. Trinh, T. Ho, and H. S. Le, "A personalized product recommendation model in e-commerce based on retrieval strategy," *J. Open Innov. Technol. Mark. Complex.*, vol. 10, no. 2, 2024. [Online]. Available: https://doi.org/10.1016/j.joitmc.2024.100303
[9] Z. Wang et al., "An Industrial Framework for Personalized Serendipitous Recommendation in E-commerce," in *Proc. 17th ACM Conf. Recommender Syst. (RecSys)*, 2023, pp. 1015–1018. [Online]. Available: https://doi.org/10.1145/3604915.3610234
[10] J. K. Aujero et al., "Cities Talking – A Social Media Platform: Crowdsourcing with Sentiment Analysis and Recommender System," in *Proc. 11th Int. Workshop Comput. Sci. Eng. (WCSE)*, 2021, pp. 978–981. [Online]. Available: https://doi.org/10.18178/wcse.2021.06.028
[11] A. Majid et al., "A Context-Aware Personalized Travel Recommendation System Based on Geotagged Social Media Data Mining," *Int. J. Geogr. Inf. Sci.*, vol. 27, no. 4, pp. 662–684, 2013. [Online]. Available: https://doi.org/10.1080/13658816.2012.696649
[12] Z. Abbasi-Moud, H. Vahdat-Nejad, and J. Sadri, "Tourism recommendation system based on semantic clustering and sentiment analysis," *Expert Systems with Applications*, vol. 167, 2021. [Online]. Available: https://doi.org/10.1016/j.eswa.2020.114324
[13] R. K. Mishra, J. A. A. Jothi, S. Urolagin, and K. Irani, "Knowledge based topic retrieval for recommendations and tourism promotions," *Int. J. Inf. Manag. Data Insights*, vol. 3, no. 1, 2023. [Online]. Available: https://doi.org/10.1016/j.jjimei.2022.100145
[14] S. Rani, D. M. Arifin, S. N. Huda, and D. H. Fudholi, "Case-based Mobile Tourism Attractions Recommender System," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1077, no. 1, p. 012009, 2021. [Online]. Available: https://doi.org/10.1088/1757-899x/1077/1/012009
[15] M. Stratigi, H. Kondylakis, and K. Stefanidis, "Multidimensional Group Recommendations in the Health Domain," *Algorithms*, vol. 13, p. 54, 2020. [Online]. Available: https://doi.org/10.3390/a13030054
[16] J. Wang et al., "Improving healthy food recommender systems through heterogeneous hypergraph learning," *Egypt. Inform. J.*, vol. 28, 2024. [Online]. Available: https://doi.org/10.1016/j.eij.2024.100570
[17] S. Tang et al., "An Integral Tag Recommendation Model for Textual Content," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 5109–5116. [Online]. Available: www.aaai.org
[18] Y. Zuo, Y. Zhou, S. Liu, and Y. Liu, "A Tag-aware Recommendation Algorithm Based on Deep Learning and Multi-objective Optimization," in *Proc. Int. Conf. Pattern Recognit., Mach. Vis. Intell. Algorithms (PRMVIA)*, 2023, pp. 42–46. [Online]. Available: https://doi.org/10.1109/PRMVIA58252.2023.00013
[19] C. Wang, X. Yang, and L. Ding, "Deep Learning Sentiment Classification Based on Weak Tagging Information," *IEEE Access*, vol. 9, pp. 66509–66518, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3077059
[20] Q. Zhang, P. Hao, J. Lu, and G. Zhang, "Cross-domain Recommendation with Semantic Correlation in Tagging Systems," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2019, pp. 1–8. [Online]. Available: https://doi.org/10.1109/IJCNN.2019.8852049
[21] L. El Youbi El Idrissi, I. Akharraz, A. El Ouaazizi, and A. Ahaitouf, "Enhanced Collaborative Filtering: Combining Autoencoder and Opposite User Inference to Solve Sparsity and Gray Sheep Issues," *Computers*, vol. 13, no. 11, 2024. [Online]. Available: https://doi.org/10.3390/computers13110275
[22] M. Altulyan, L. Yao, X. Wang, C. Huang, S. S. Kanhere, and Q. Z. Sheng, "A Survey on Recommender Systems for Internet of Things: Techniques, Applications and Future Directions," *The Computer Journal*, vol. 65, no. 8, pp. 2098–2132, 2022. [Online]. Available: https://doi.org/10.1093/comjnl/bxab049
[23] A. Shetty, A. Shetye, P. Shukla, A. Singh, and S. Vhatkar, "A Collaborative Filtering-Based Recommender Systems approach for Multifarious applications," *J. Electr. Syst.*, vol. 20, no. 4, 2024.
[24] Coursera. [Online]. Available: https://www.coursera.org/
[25] SWAYAM. [Online]. Available: https://swayam.gov.in/
[26] Eventbrite. [Online]. Available: https://www.eventbrite.com/
[27] A. Dennis, B. H. Wixom, and R. M. Roth, "Systems Analysis and Design Seventh Edition", 2018.

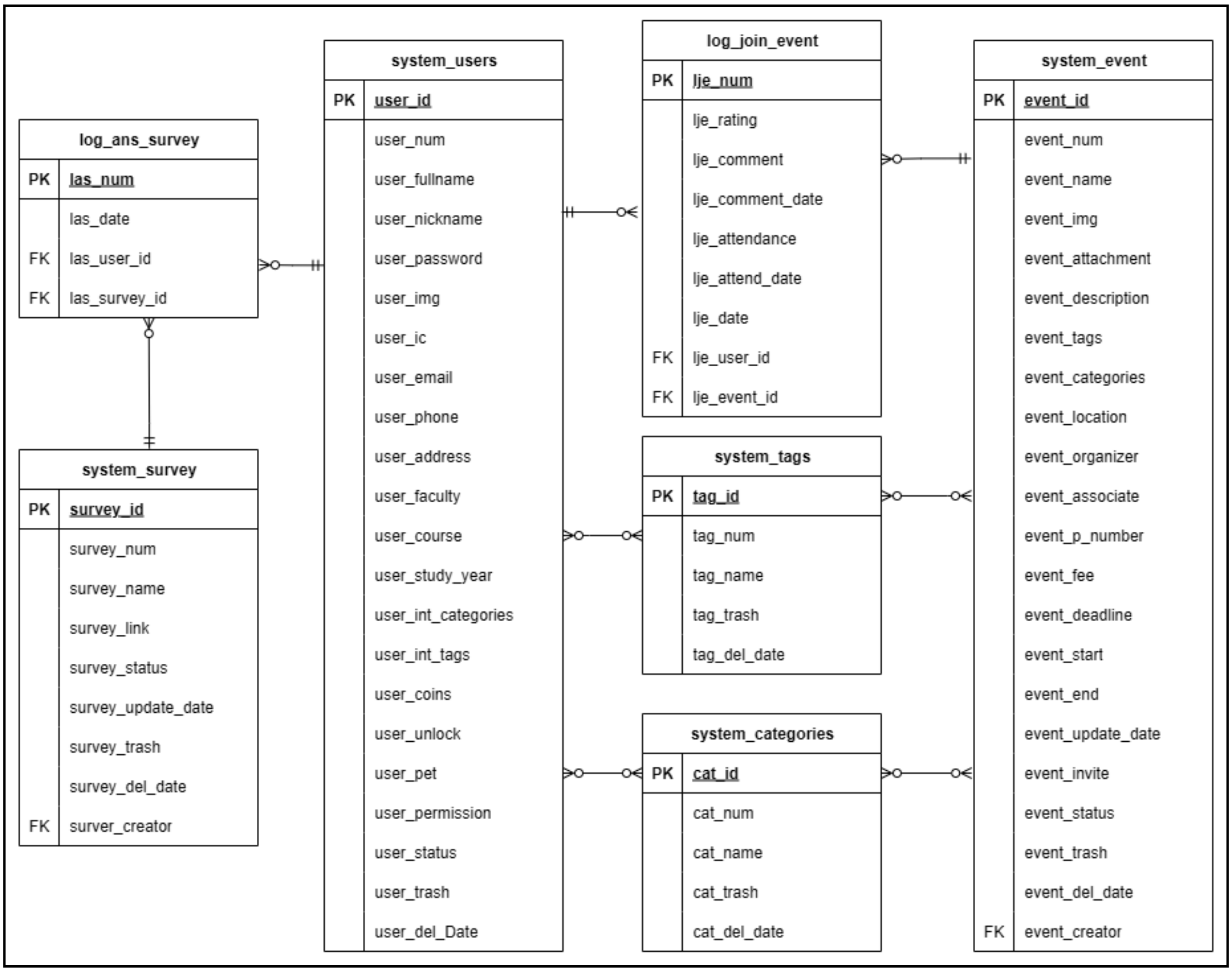
Appendix A: Context Diagram



Appendix B: DFD Level 0



Appendix C: ERD



Appendix D: UAT

User Acceptance Testing (UAT): Evaluating the Effectiveness of the ***Student Interest Tendencies and Recommendation System (SITARS)***

Welcome to the User Acceptance Testing (UAT) form for the ***Student Interest Tendencies and Recommendation System (SITARS).*** This form is designed to gather your feedback on the functionality, usability, and overall performance of the system. Your input will help us ensure that SITARS meets the expectations and needs of students in terms of accurate interest prediction and personalized recommendations.

Please rate each aspect of the system based on your experience, and feel free to provide any additional comments or suggestions. Your feedback is crucial for refining the system before its official release.

Thank you for participating!

Sign in to Google to save your progress. Learn more

* Indicates required question

Section 1: Functionality Testing *

In this section, we assess whether the system meets the functional requirements.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The system worked as expected for the tasks I performed.	○	○	○	○	○
I encountered no unexpected errors or issues during my use of the system.	○	○	○	○	○
All features of the system (e.g., form submission, data processing) worked correctly.	○	○	○	○	○
I was able to complete my tasks without needing external assistance.	○	○	○	○	○
The system did not crash or encounter any bugs during my use.	○	○	○	○	○

Section 2: Usability and User Interface *

This section assesses how user-friendly and intuitive the system is.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
It was easy to navigate the system and find the information I needed.	○	○	○	○	○
The layout of the system was intuitive and easy to understand.	○	○	○	○	○
I was satisfied with the visual design of the system (color scheme, fonts, etc.).	○	○	○	○	○
The buttons and actions were easy to find and understand.	○	○	○	○	○
The system's instructions and messages (e.g., error messages, prompts) were clear.	○	○	○	○	○

Section 3: Performance and Feedback *

This section focuses on system performance and overall feedback.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The overall performance of the system (speed, responsiveness) was satisfactory.	○	○	○	○	○
The system performed well under normal usage load.	○	○	○	○	○
I was satisfied with the time it took for the system to load and respond to my actions.	○	○	○	○	○
The system met my needs and expectations for the tasks I was testing.	○	○	○	○	○
I would recommend this system to other users.	○	○	○	○	○

Do you approve this system for release? *

- ○ Yes
- ○ No
- ○ Maybe

Do you have any additional comments, suggestions or feedback about SITARS? Your insights are valuable and will help us make further improvements.

Your answer

Thank you for participating in the User Acceptance Testing (UAT) for the ***Student Interest Tendencies and Recommendation System (SITARS)*** ! Your feedback is invaluable and will play a crucial role in improving the system before its official release. We appreciate your time and effort in helping us enhance the user experience and functionality of SITARS.

If you have any further comments or suggestions, feel free to reach out. We are committed to making this system as effective and user-friendly as possible.

Thank you again for your valuable input!

Submit Clear form

Never submit passwords through Google Forms.

This form was created inside UTHM Siswa.
Does this form look suspicious? Report

Google Forms