RIOT.ID: Revolutionizing Running Community Management with Next.js and Gamification

Peter Shaan[1], Jay Idoan Sihotang*[2]
Faculty of Information Technology, Universitas Advent Indonesia
e-mail: [1]2181002@unai.edu, [2]jay.sihotang@unai.edu

Abstract

The RIOT Indonesia running community has been actively engaged since 2018, continuing to expand its reach. However, despite this significant growth, the community still relies on manual systems for managing membership and recording activities. Information technology, recognized as a catalyst for transformation across various sectors, including sports, presents an opportunity to enhance the efficiency of community management. This research focuses on the design of the Web RIOT.ID application, a web-based solution that integrates QR Code technology and gamification principles through an Experience Points (XP) system to motivate member participation. The user interface design of the application employs the Design Thinking methodology, ensuring that the solutions developed are tailored to meet user needs. Developed using a Rapid Application Development (RAD) approach, the application leverages Next.js and MongoDB. Evaluation is conducted through black box testing to confirm the application's functionality and its alignment with the established objectives. The RIOT.ID application is expected to serve as a model for other communities aiming to harness information technology to enhance organizational management and member engagement.

Keywords— Information technology, Website, Next.js, Gamification, RIOT.ID

1. INTRODUCTION

Information technology has become a driving force behind transformative changes across various sectors [1], including the sports industry. This technological advancement compels sectors to adapt and leverage technology to enhance the efficiency and effectiveness of their activities. One significant development in this area is the breakthrough in data collection and analysis [2], which has led to the creation of tools and platforms that not only support but also reshape how sports activities are managed and executed. In this context, the application of information technology has opened new opportunities to improve the convenience and experiences of individuals.

RIOT Indonesia is a community dedicated to individuals passionate about sports and health. Established in 2018, this organization has been promoting healthy lifestyles through running activities, and expanding its presence on social media, particularly Instagram, to 15 cities across Indonesia. Despite its growth and recognition of the rapid advancements in technology, RIOT's membership administration and activity tracking still rely on manual systems. This manual approach poses several challenges, including inefficiencies in data management, difficulties in tracking members, and a lack of motivation for users to engage actively in activities.

To address these issues, the utilization of information technology in membership and attendance recording can significantly improve the efficiency and accuracy of community management. A recent study by [3] reviewed advancements in automated attendance systems, highlighting their efficiency and effectiveness in streamlining attendance processes and reducing time wastage associated with manual methods. This is further supported by prior research on QR code-based attendance systems, that the benefits of QR code-based attendance systems, offering a contactless, efficient, and cost-effective alternative to traditional biometric methods also helps employees avoid waiting in lines and start their tasks promptly [4], [5]. This study is expected to

contribute to the implementation of a unique attendance scenario, where each user is assigned a distinct QR code for each event, and each event carries a specific point value. This contribution is anticipated to motivate users to participate more actively in physical activities within the community. As mentioned before, this study incorporates the concept of gamification, a strategy that has become a prominent tool in marketing and customer engagement [6], Even the education sector has begun to adopt gamification, and research has shown its positive impact on learning motivation [7]. By utilizing gamification, the designed application is expected to enhance the motivation of RIOT Indonesia members to participate in sports activities.

Building upon these observations, this research aims to design a web-based application for the RIOT community that integrates key features such as QR code technology and gamification through an experience points (XP) system. The application will be developed using the Rapid Application Development (RAD) approach, which has been widely adopted in research [8]. Technologies employed include Next.js for the front end and MongoDB as the database. The goal of this research is not only to develop a technological solution but also to evaluate its impact on operational efficiency and member participation.

Through black-box testing and user surveys, this study aims to assess the effectiveness of the application in meeting its objectives. The findings are expected to serve as a model for other sports communities looking to leverage technology to improve organizational management and member engagement, while also contributing to the literature on the application of information technology and gamification within sports and health-oriented communities.

2. RESEARCH METHODS

2.1. Rapid Application Development

The method used in this research is the Rapid Application Development (RAD) methodology, first introduced by James Martin in his 1991 book, Rapid Application Development [9]. RAD is known for its advantages in flexibility, its quick adaptability to changes, and relatively low development costs [10]. In a study conducted by Deni and Henri, RAD proved to be more efficient than the Waterfall method in terms of speed and its ability to accommodate changing user requirements [11]. Additionally, the rapid development cycle of RAD is highly adaptive to shifts in project scope and user needs [12]. However, while the RAD approach allows for fast-paced development, it also introduces higher risks due to the speed of execution, necessitating an experienced development team capable of managing the complexities involved [13]. The process patterns employed in the RAD methodology for this study are illustrated in Figure 1.

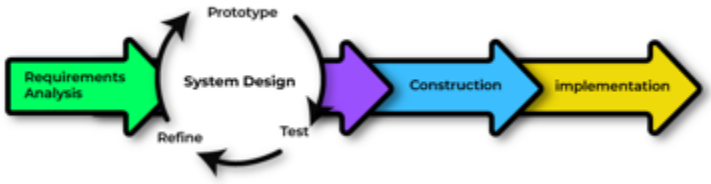

Figure 1. RAD Development Process

In Figure 1, the sequence of activities involved in RAD development is visually illustrated. This process emphasizes an iterative approach, where users are continuously engaged in testing and refining the system, leading to a final product that better aligns with user needs and expectations. The RAD methodology not only accelerates development time but also enhances the quality and relevance of the resulting system.

1. Requirements Analysis: Requirements analysis is the foundational stage of system development. This phase is crucial as it establishes the groundwork for developing an application that is both accurate and effective. In this study, the analysis includes identifying the essential features that the application must have to meet the goals of the RIOT Indonesia community. Lead research [14] using the KANO Model defines the need

and importance of each feature from RIOT members. Additionally, decisions on application methods and the database were also made during this phase. The application was developed using Node.js, with the React library in the Next.js framework, and MongoDB as the database.

2. System Design: Once the requirements analysis is complete, the system design process begins. This phase is iterative, allowing for adjustments and refinements if discrepancies are identified in the earlier stages. This iterative aspect of RAD is what makes it particularly effective in meeting user needs and enhancing user satisfaction. Early prototypes, covering key features identified during the requirements analysis, are created, often in the form of documentation like Unified Modeling Language (UML) diagrams.
3. Construction: In this phase, the actual development of the application begins, based on the system design. Due to the rapid nature of RAD, the construction stage produces beta versions leading to the final version of the application. It is also possible to return to the system design phase during development if the outcome does not yet meet user requirements.
4. Implementation: The final phase, also known as cutover, is where the system is fully implemented and introduced to users. At this stage, the system is ready for production and can be packaged for deployment.

2.2. Gamification Concept

Gamification is an approach that involves the application of game-like elements in non-game contexts to solve real-world problems [15], [16]. Gamification introduces game-like experiences into non-game services to drive behavioral outcomes and enhance user engagement. In today's digital world, gamification is increasingly applied to boost user interaction by incorporating challenges, rewards, or rankings similar to those in RPG games [16]. The core idea is to create an engaging experience that motivates users to stay involved, much like how players are incentivized to level up or achieve high scores in a game.

In the system developed in this study, and aligning with general strategies observed in recent years, gamification is implemented through an XP (Experience Points) system for members. Each time a member of the RIOT Indonesia community participates in an activity, they earn XP, which helps them elevate their level or status within the community. Users with the highest XP will appear at the top of the ranking system. In this application, "Elite" represents the highest rank, while "Jogger" is the lowest. In this way, the application functions not only as an administrative tool but also as a motivational platform, encouraging members to actively engage in various community activities and prioritize their health.

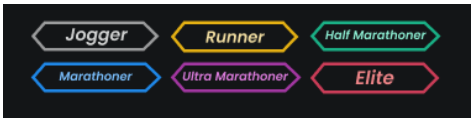

Figure 2. Six Levels of Gamification on RIOT.ID App from lowest to highest

2.3. Next Js

Next.js is a React framework built on the JavaScript programming language [17]. It is a flexible framework widely used for developing full-stack applications. This framework is chosen primarily for its server-side rendering capabilities, which allow web pages to be rendered on the server before being sent to the client, leading to faster load times for users in web browsers.

2.4. MongoDB

MongoDB is a document-based NoSQL database that has emerged as a solution for handling large-scale data, which is increasingly essential in modern applications [18]. Unlike traditional relational databases, MongoDB does not use tables, columns, or rows. Instead, data is stored as BSON (Binary JSON), a format derived from JSON. Each document in MongoDB uses

a key-value structure, where the document's elements are identified by unique keys. In this research, MongoDB provides fast access and development capabilities for the RIOT application system.

2.5. Unified Modeling Language (UML)

Unified Modeling Language (UML) is a standard language used for documenting, designing, and developing software systems [19]. UML offers various diagrams that allow for clear and structured visualization of both the system's structure and behavior. In this study, UML is employed to design the RIOT.ID application for the RIOT Indonesia community. The system supports three main user types: User, Officer, and Admin. Users can register for and participate in events, each of which awards points upon successful participation. Officers and Admins have privileges for creating events, managing participant attendance, and generating reports, though only Admins can manage user and officer accounts.

2.6. Use Case Diagram

A Use Case Diagram is a UML diagram that illustrates interactions between actors (users, officers, and admins) and the system functions being developed [20], [21]. In this study, Figure 3 presents the use case diagram, which highlights seven key functions accessible to users, officers, and admins. Users can browse events, view rankings, and search for profiles without logging in, but they must register to manage personal profiles and register for events. After successful registration, users receive email notifications and tickets that award points upon attendance verification by officers. Officers manage event participants, while the Admin, as a generalization of the officer role, has exclusive rights to manage both users and officers.

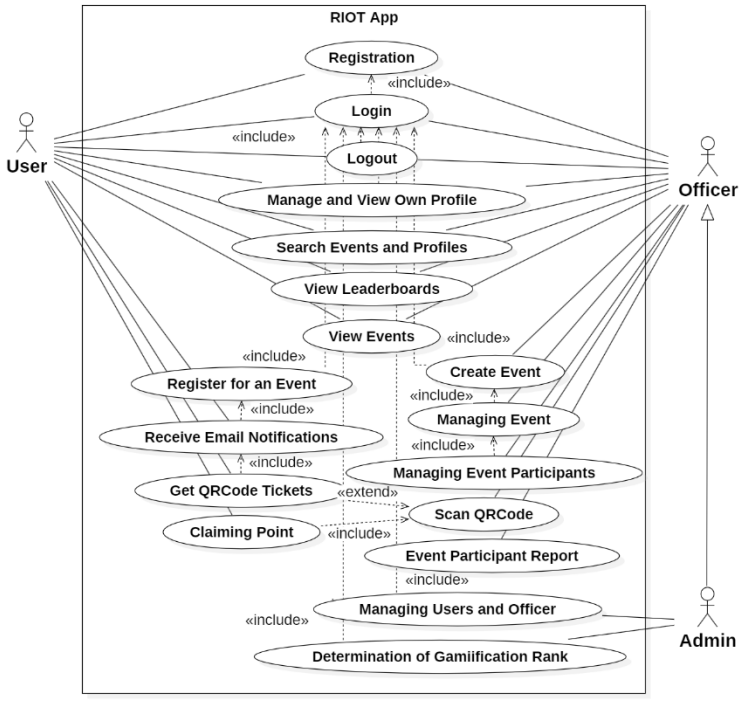

Figure 3. Use Case Diagram RIOT.ID

2.7. Activity Diagram

An Activity Diagram is a more detailed UML diagram compared to a Use Case Diagram. Both are part of UML, which is the standard for documenting, designing, and developing software systems [22]. An Activity Diagram illustrates the procedures of an actor's activities, whether logically or as part of the business processes within a system or application.

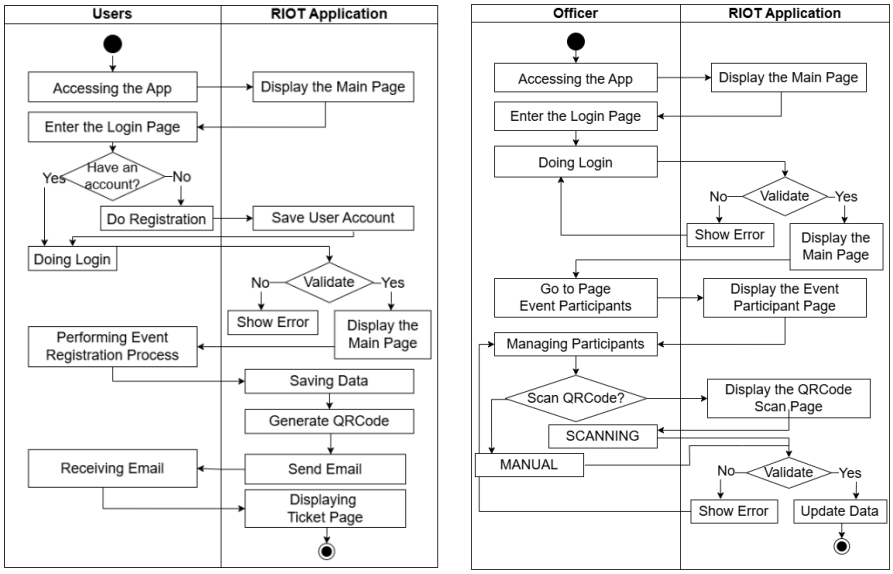

Figure 4. Activity Diagram of Event Registration and Management Attendance

From Figures 4 and 5, the flow of the activity diagrams for Users and Managers is depicted. The User can perform Login and Registration, and similarly, the Manager can execute these actions as well. The registration process can only take place after the login credentials are successfully obtained. During this process, a QR Code will be generated by the application, which will subsequently be displayed on the ticket page along with notifications for the user. On the other hand, the Manager can manage participants through the event participant page, which will include a QR Code scanning interface used for user attendance, although manual attendance is also an option.

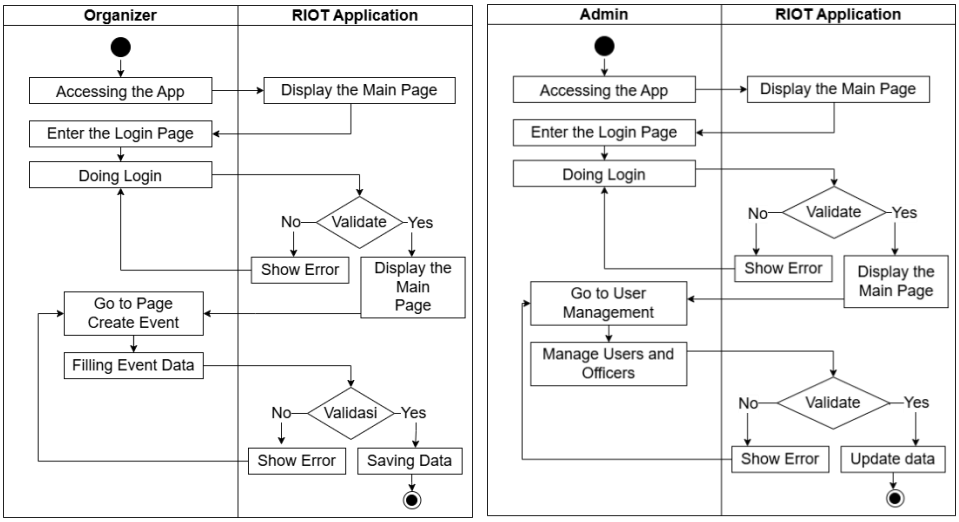

Figure 5. Activity Diagram of Event Creation and User/Manager Management

The subsequent flow pertains to the activity diagrams for event creation and the management of Users/Managers. The event creation process is conducted by the Manager, with a data validation check in place to ensure that the event input data aligns with the system requirements. A similar validation process is applicable for User or Manager management by the Admin actor, who has the authority to designate Users or elevate their status to Manager. Validation will also occur if the Admin provides data that does not conform to the system specifications.

2.8. Class Diagram

The Class Diagram is a crucial component of the Unified Modeling Language (UML) that visualizes the static structure of the system by displaying classes, attributes, methods, and the relationships between these classes. Based on this research, the Class Diagram is illustrated in Figure 5, showcasing five primary classes: Events, User, Points, Attendance, and Ticket, along with an additional Authentication class. Each class has a specific role in supporting the application's functionalities, such as event management, user profile handling, attendance recording, and QR Code processing within tickets.

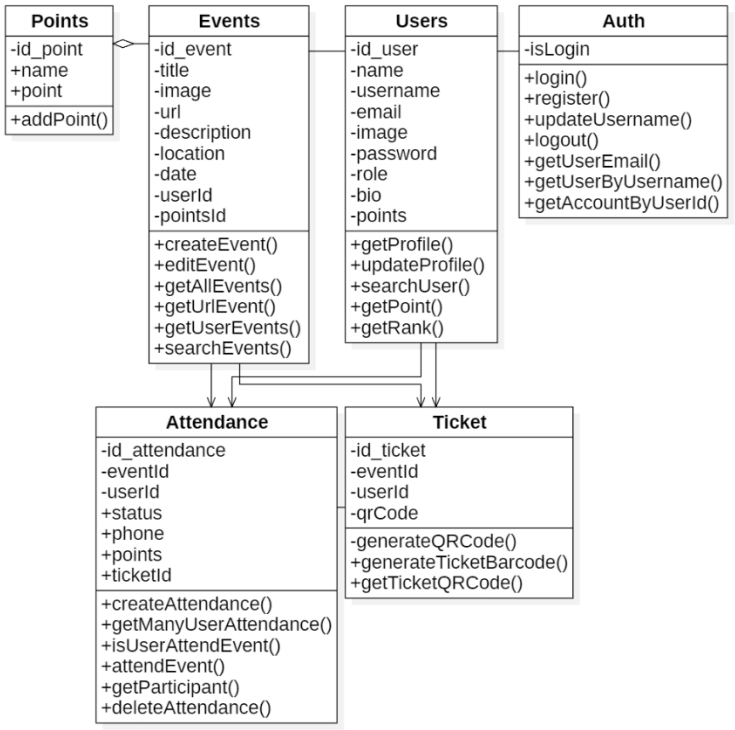

Figure 6. Class Diagram RIOT.ID

In Figure 6, the relationships between the classes illustrate how entities like Users and Events interact; for instance, Users can view and participate in Events, while their attendance is recorded through Tickets managed by the system. Additionally, the Points class has an aggregation relationship, indicating that Points can exist independently of Events. This Class Diagram not only aids in the design and development of the system but also facilitates a deeper understanding of the architecture of the proposed application in this research.

2.9. Database Design with MongoDB Atlas

MongoDB Atlas is a service from MongoDB that provides an easy-to-manage infrastructure. Atlas allows for the deployment of MongoDB databases in the cloud without the need to handle server configuration directly, thereby accelerating development time and reducing administrative complexity [22]. The MongoDB Atlas service in this study offers various features that support efficient database design. Advantages include automatic scaling that can adjust capacity according to application needs, stringent security with data encryption at rest and in transit, as well as support for various programming languages and frameworks. The researcher defined five main collections for the RIOT.ID application: user, event, points, attendance, and ticket.

User Collection	
Field	Tipe Data
id	String
name	String
username	String
email	String
image	String
password	String
role	String
bio	String
points	Int
createdAt	Datetime
updatedAt	Datetime

Attendance Collection	
Field	Tipe Data
id	String
eventId	String
userId	String
status	Attendance Status
phone	String
points	Int

Event Collection	
Field	Tipe Data
id	String
title	String
image	String
url	String
description	String
buildingName	String
location	String
date	DateTime
userId	String
pointsId	String

Point Collection	
Field	Tipe Data
id	String
name	String
points	Int

Ticket Collection	
Field	Tipe Data
id	String
eventId	String
userId	String
qrcode	String

Figure 7. Collection Database MongoDB

2.10. Design Interface

After analyzing the system used in the research, the researcher designed the interface of the RIOT.ID application as a reference to meet user needs. The researcher utilized Balsamiq Mockups 3 software. In this design process, the researcher applied the Design Thinking concept, which places humans at the center of development, also known as Human-Centered [23]. This process is reinforced by previous research [14] that used the KANO method to assess the importance of each function and user needs, thereby creating a more effective design. Design Thinking encompasses five principles, known as the five-step thinking model by Stanford School.

1. Empathize: The researcher engages in an empathy process to understand issues related to the ineffectiveness of manual attendance recording and individual boredom in exercising within the RIOT.ID community through observation and interviews.
2. Define: This stage involves a detailed analysis of the existing problems. The researcher defines the primary objective of the application as improving administrative efficiency and increasing member participation through a gamification approach.
3. Ideate: After defining the problems, the researcher explores various solutions, such as creating a QR Code page for attendance, a ranking page within the XP (Experience Points) system to motivate members, and pages for tickets and emails.
4. Prototype: The researcher builds an application prototype that includes key features such as a QR Code-based attendance system, an XP system, and a user interface designed for ease of use.
5. Test: The prototype is tested by members of the RIOT Indonesia community to identify issues, understand user interactions, and gather feedback. The main pages of the application include the home page, log-in, and registration, about and ranking, profile and edit profile, event creator, email and tickets, participant and scan participant, as well as the admin page.

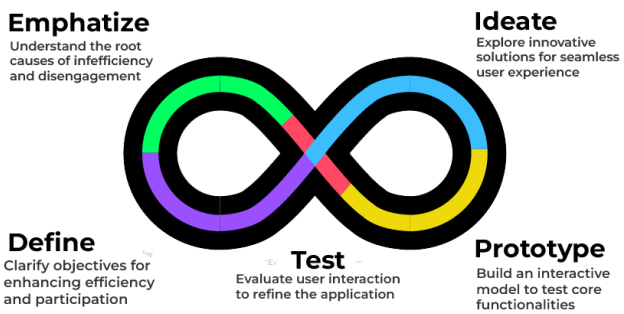

Figure 8. Design Thinking

The results of the application design using the Design Thinking method begin with Figure 9. The Home Page, as shown in Figure 9, is a central component of the RIOT.ID interface. Additionally, there is an Event page, which serves as the primary feature for browsing and registering for events. It also includes event descriptions and a map. Figure 10 presents the Email & Ticket page, which users receive upon attending an event.

Figure 9. Design of the Home and Event Pages

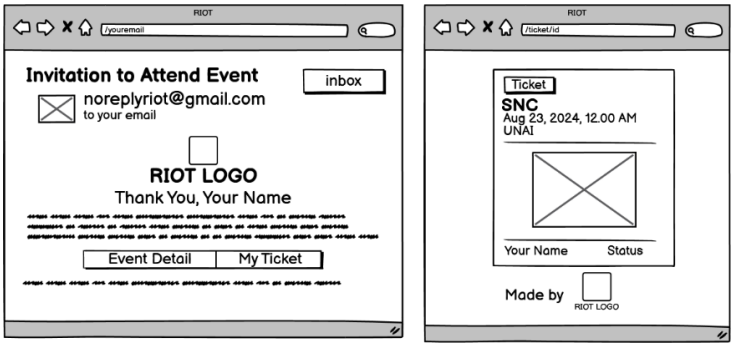

Figure 10. Design of the Email and Ticket Page

Figures 11 depict design pages accessible by Officers or Admins, allowing them to manage attendance, scan participants, and perform account management tasks.

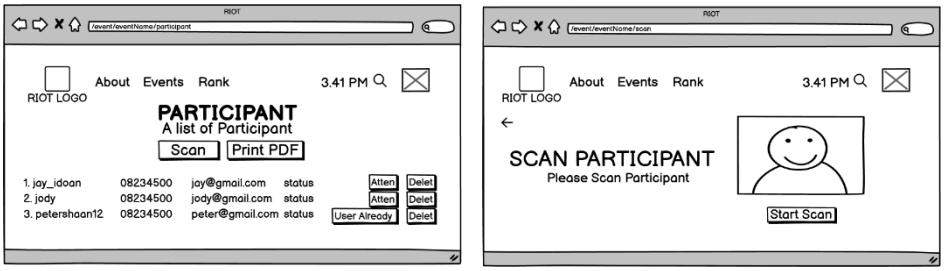

Figure 11. Design of the Participant and Scan Participant Pages

3. RESULT AND DISCUSSION

Based on the design and needs analysis conducted in the previous sections, the RIOT.ID application has been completed and is now live with the following interface. The implemented design is accessible via the website ***https://riot-id.vercel.app/***. Due to space limitations, not all sections of the fully functioning application can be displayed in this document.

Figure 12 shows the initial display that visitors encounter when accessing the website and event page of the RIOT community. Visitors can explore these pages without authentication, allowing them to learn more about RIOT Indonesia.

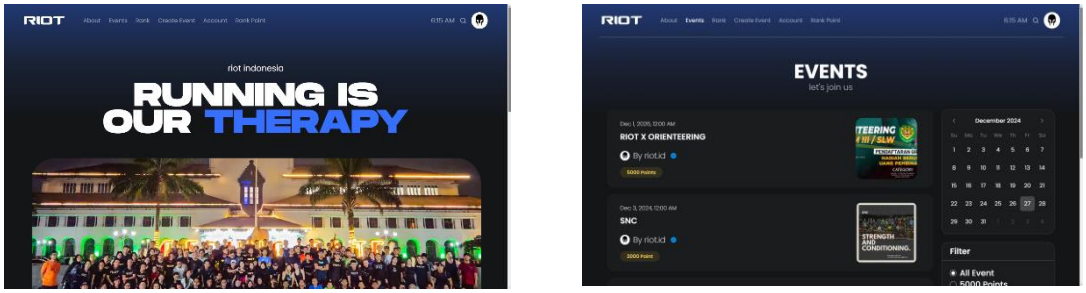

Figure 12. Front Page and Events Page

Figure 13 displays the design of the user login page in the RIOT application. All users must undergo the authentication process before they can register for an event. Additionally, the user profile is included, featuring their rank status and gamification details.

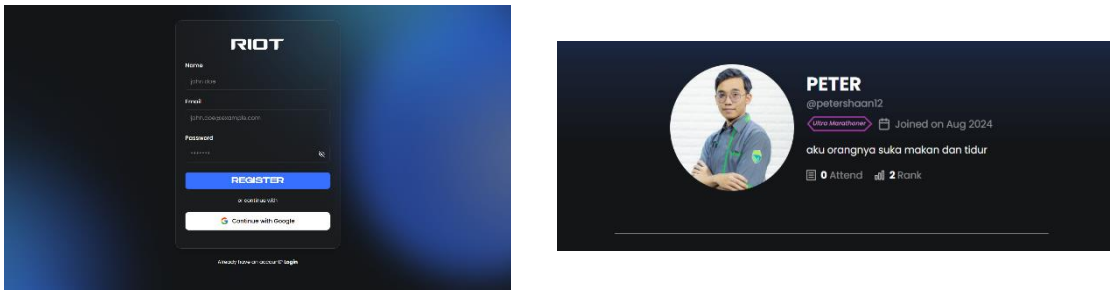

Figure 13. Login Page and Profile Page

Figures 14-15 display the event details, including a map and description where users can register for the event, It also shows information about the owner, and event date, and provides a point the user will get also the option to copy the event link for easy sharing. As well as the ticket page and email notification. Users will receive an email upon registering for an event and have a ticket.

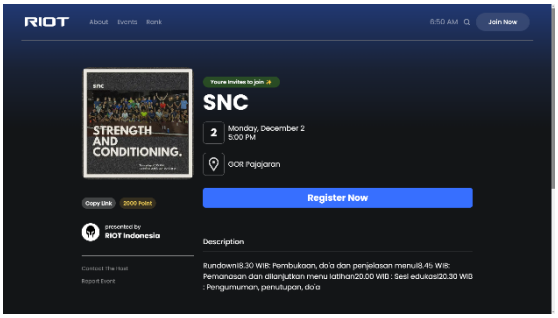

Figure 14. Login and User Account Page

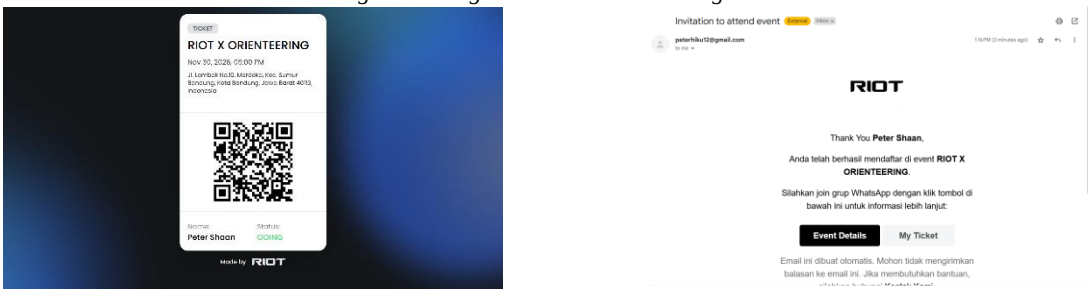

Figure 15. Ticket Page and Email Notification by System

Figure 16-17 shows the participant page for the event, which facilitates attendance tracking and reporting. It includes a scan page. When a member scans their ticket, their data is

recorded, and points are awarded. It also includes a PDF feature that allows event organizers to print a PDF of the event as a record of the event outcomes and member attendance.

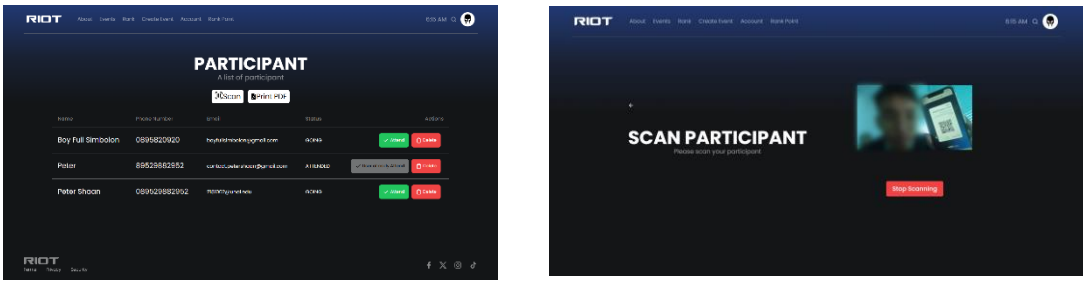

Figure 16. Display of Participant Page and Scan Participant from Management

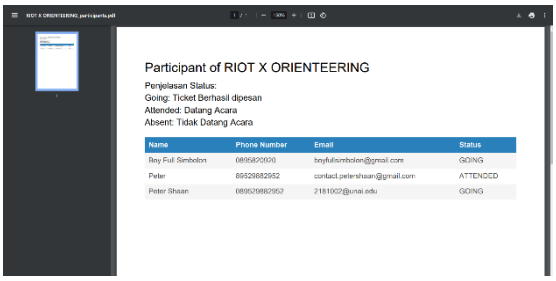

Figure 17. Display when the officer prints the event member report in PDF

Figure 18 displays the admin interface for managing users and administrators, as well as the management of gamification ranks for users. The admins were able to create a rule of how many XP per event as well defining the validity (expiration time) of the points. And defining the roles of each user (i.e.: Chapter Captain, Officers, etc.).

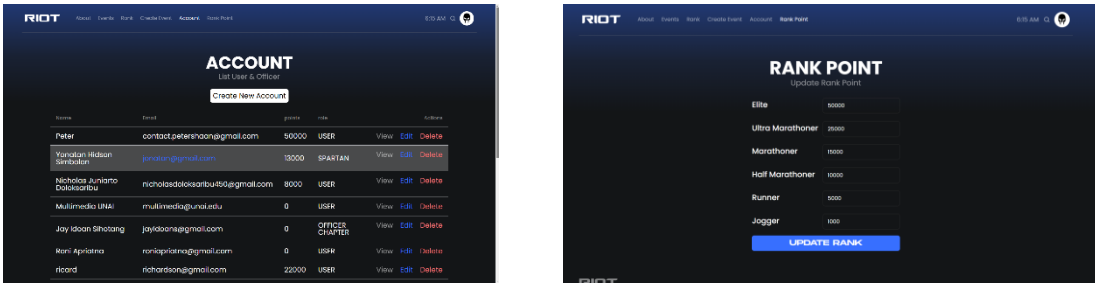

Figure 18. Account and Rank Point Page View from the Admin Side

3.1. Testing Table (Black Box Testing)

The purpose of conducting the Testing Table for the system is to ensure that the developed application meets the specifications and fulfills user requirements. Testing is performed on various aspects of the application, including functionality, performance, security, and usability. For this testing phase, a qualified software tester at the IT Department, FTI UNAI lecturer, and a RIOT user were testing this application.

Table 1. Final Acceptance Testing

Checked System	**Expected output**	**Result**
Authentication	Successful login with correct credentials.	[✓] Valid [ ] Invalid
Authentication	Error message appears for incorrect credentials.	[✓] Valid [ ] Invalid
Authentication	New user registration is successful with proper validation.	[✓] Valid [ ] Invalid
Authentication	The login option with Google works properly.	[✓] Valid [ ] Invalid

Home Page	The navigation menu and search features function properly.	[✓] Valid [ ] Invalid
Events Page	Event registration works with the required authentication.	[✓] Valid [ ] Invalid
Ticket Page	An event ticket is generated after successful registration.	[✓] Valid [ ] Invalid
Ticket Page	The QR Code on the ticket can be scanned and verified.	[✓] Valid [ ] Invalid
Ticket Page	Attendance status is displayed correctly.	[✓] Valid [ ] Invalid
Profile Page	User profile information is displayed correctly.	[✓] Valid [ ] Invalid
Profile Page	Users can edit their profile information.	[✓] Valid [ ] Invalid
Profile Page	Changes to the profile are successfully saved and updated.	[✓] Valid [ ] Invalid
Create Event Page	Pages can only be accessed by Manager or admin accounts.	[✓] Valid [ ] Invalid
Create Event Page	The event creation form can be filled out and submitted.	[✓] Valid [ ] Invalid
Create Event Page	The created event is displayed on the event page.	[✓] Valid [ ] Invalid
Leaderboards Page	Member rank by XP is displayed correctly.	[✓] Valid [ ] Invalid
Leaderboards Page	Leaderboards show accurate rankings.	[✓] Valid [ ] Invalid
Email	Email notifications are sent after event registration.	[✓] Valid [ ] Invalid
Attendance and QR Code Scan	QR Codes can be scanned and verified for attendance.	[✓] Valid [ ] Invalid
Attendance and QR Code Scan	Attendance status is updated after QR Code scanning.	[✓] Valid [ ] Invalid
Attendance and QR Code Scan	Print PDF can be done and provides a list of members correctly.	[✓] Valid [ ] Invalid
Admin Page	Admin can manage users and administrators.	[✓] Valid [ ] Invalid
Admin Page	Admin can delete accounts or change user status.	[✓] Valid [ ] Invalid
Security Testing	User data is protected with encryption.	[✓] Valid [ ] Invalid
Security Testing	User access rights are in accordance with their role.	[✓] Valid [ ] Invalid
Ease of Use Testing	The application interface is easy to use and understand.	[✓] Valid [ ] Invalid

The results of testing conducted by several users indicate that the application operates effectively following the planned requirements. Key functionalities, such as attendance tracking using a unique QR code for each user per activity, are functioning as intended, and the gamification feature, including point calculations, operates correctly. Functionally, the application meets user expectations based on the requirements outlined in the earlier stages of development.

4. CONCLUSION

From the results of the research and discussion, it can be concluded that the RIOT.ID Application, which was built using QR Code technology and gamification with the Experience Points (XP) system, has succeeded in providing an effective solution to improve the efficiency of managing the RIOT Indonesia community. Through the Rapid Application Development (RAD) approach and interface design based on the Design Thinking method, this application is relevant to user needs and hopefully able to increase member motivation to participate. Blackbox testing and user surveys show that the application works well, and its potential to be adapted by other sports communities is also wide open.

5. SUGGESTION

Some development suggestions that can be considered include expanding the gamification system by adding a points (XP) redemption feature for community merchandise, such as clothing, sports accessories, or other gifts that can increase member motivation. In addition, integration with the Strava app or other sports communication tools could enrich the user experience, allowing members to share achievements and interact more effectively within the community. For further research, a follow-up questionnaire or research was needed to justify the needs of the users, especially in Service Quality, Application Security, Future Development, etc. And also, the use of another Stack or developing a native mobile app to enrich the user experience based on convenience. Integrating the application into other sports services such as Strava or Garmin Connect to enable validation of event completion such as training calendar could enhance the engagement of the app.

6. ACKNOWLEDGEMENTS

This research was conducted with the invaluable support and assistance of Universitas Advent Indonesia and RIOT Indonesia (Chapter Bandung). The authors, as both students and lecturers of the institution, are grateful for the resources and opportunities provided that made this study possible.

8. REFERENCES

[1] M. A. Mohamed Hashim, I. Tlemsani, and R. Matthews, "Higher education strategy in digital transformation," *Educ. Inf. Technol.*, vol. 27, no. 3, pp. 3171–3195, Apr. 2022, doi: 10.1007/s10639-021-10739-1.

[2] A. Nurdiassa, A. Zulfikar, F. Rasyid, and A. T. Wulandari, "Implementasi Kebijakan Smart City dalam Mewujudkan Makassar Kota Dunia," *J. Public Adm. Gov.*, vol. 3, no. 1, pp. 37–46, Apr. 2021, doi: 10.22487/jpag.v3i1.114.

[3] S. Yelve, D. Patil, R. Singh, and J. Sangoi, "A review on the Advancement of Automated Attendance System.pdf," *VIVA-Tech Int. J. Res. Innov.*, vol. 1, no. 5, pp. 1–5, 2023, doi: 10.6084/M9.FIGSHARE.22270309.V1.

[4] A. Y. Felix, S. Vignesh, and M. J. Andrew, "Mobile application and QR code for attendance system," *AIP Conf. Proc.*, vol. 3075, no. 1, p. 020236, Jul. 2024, doi: 10.1063/5.0217048.

[5] M. Ichsan and H. Hutrianto, "Implementasi Algoritma Vignere Chiper Berbasis QRCode Untuk Absensi Pegawai PT. Delameta Bilano Cabang Ruas Tol Palembang - Inderalaya," *J. Softw. Eng. Ampera*, vol. 3, no. 3, pp. 170–185, Oct. 2022, doi: 10.51519/journalsea.v3i3.267.

[6] S. Deterding, D. Dixon, R. Khaled, and L. E. Nacke, "Gamification: Toward a Definition," *CHI*, May 2011.

[7] S. Srimuliyani, "Menggunakan Teknik Gamifikasi untuk Meningkatkan Pembelajaran dan Keterlibatan Siswa di Kelas," *Educ. J. Pendidik. Dan Kesehat.*, vol. 1, no. 1, pp. 29–35, Nov. 2023, doi: 10.31004/jedu.v1i1.2.

[8] A. Rachmat, H. Santoso, A. Wibowo, and R. Delima, "Developing Agriculture Land Mapping using Rapid Application Development (RAD): A Case Study from Indonesia," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, pp. 232–241, Nov. 2019, doi: 10.14569/IJACSA.2019.0101033.

[9] J. Martin, *Rapid Application Development*. Macmillan Publishing Company, 1991.

[10] R. Naz and M. N. A. Khan, "Rapid Applications Development Techniques: A Critical Review," *Int. J. Softw. Eng. Its Appl.*, vol. 9, no. 11, pp. 163–176, Nov. 2015, doi: 10.14257/ijseia.2015.9.11.15.

[11] D. Murdiani and H. Hermawan, "Perbandingan Metode Waterfall dan RAD (Rapid Application Development) Pada Pengembangan Sistem Informasi," *JurTI J. Teknol. Inf.*, vol. 6, no. 1, pp. 14–23, Jun. 2022, doi: 10.36294/jurti.v6i1.2544.

[12] R. Delima, H. B. Santosa, and J. Purwadi, "Development of Dutatani Website Using Rapid Application Development," *IJITEE Int. J. Inf. Technol. Electr. Eng.*, vol. 1, no. 2, pp. 36–44, Sep. 2017, doi: 10.22146/ijitee.28362.

[13] A. Munawir and N. Nugroho, "Penerapan Metode Rapid Application Development Pada Sistem Informasi Monitoring Pelanggaran Siswa," *J. Ilm. Sist. Inf. Akunt.*, vol. 3, no. 1, pp. 69–78, Jun. 2023, doi: 10.33365/jimasia.v3i1.3697.

[14] J. I. Sihotang, "Pemanfaatan Model Kano Dalam Mengidentifikasi Kebutuhan Utama Pengguna pada Aplikasi RIOT:ID," *TeIKa*, vol. 14, no. 2, pp. 107–116, Nov. 2024, doi: 10.36342/teika.v14i2.3728.

[15] C. P. A. Sukran and I. Huda, "Pengembangan Aplikasi Pembelajaran Online dengan Metode Gamifikasi Berbasis Web," *J. Internet Softw. Eng.*, vol. 4, no. 1, pp. 6–11, May 2023, doi: 10.22146/jise.v4i1.2456.

[16] A. R. Ammalia and S. Sumar, "PERAN NILAI-NILAI GAMIFIKASI DAN SOSIAL TERHADAP ADOPSI E-COMMERCE," *J. Bisnis Dan Manaj. JBM*, pp. 29–39, Jan. 2024, doi: 10.23960/jbm.v20i1.2041.

[17] S. Srivastava, H. Shukla, N. Landge, A. Srivastava, and D. Jindal, "A Comprehensive Review of Next.js Technology: Advancements, Features, and Applications," *SSRN Electron. J.*, p. 4, 2024, doi: 10.2139/ssrn.4831070.

[18] M. Rathore and S. S. Bagui, "MongoDB: Meeting the Dynamic Needs of Modern Applications," *Encyclopedia*, vol. 4, no. 4, pp. 1433–1453, Dec. 2024, doi: 10.3390/encyclopedia4040093.

[19] J. Panjaitan and A. F. Pakpahan, "Perancangan Sistem E-Reporting Menggunakan ReactJS dan Firebase," *J. Tek. Inform. Dan Sist. Inf.*, vol. 7, no. 1, pp. 20–34, Apr. 2021, doi: 10.28932/jutisi.v7i1.3098.

[20] L. Nurhidayah, A. Salsabillah, and F. R. Yanti, "Perancangan Aplikasi Pemesanan Tiket Bioskop di Kota Medan Berbasis Android," *J. Komput. Teknol. Inf. Dan Sist. Inf. JUKTISI*, vol. 2, no. 2, pp. 305–314, Sep. 2023, doi: 10.62712/juktisi.v2i2.76.

[21] J. Margaretha and A. Voutama, "Perancangan Sistem Informasi Pemesanan Tiket Konser Musik Berbasis Web Menggunakan Unified Modeling Language (UML)," *JOINS J. Inf. Syst.*, vol. 8, no. 1, pp. 20–31, Jun. 2023, doi: 10.33633/joins.v8i1.7107.

[22] F. Uz Zaman, M. Khuhro, K. Kumar, N. Mirbahar, M. Z. Khan, and A. Kalhoro, "Comparative Case Study Difference Between Azure Cloud SQL and Atlas MongoDB NoSQL Database," *Int. J. Emerg. Trends Technol. Comput. Sci.*, vol. 9, pp. 999–1002, Jul. 2021, doi: 10.30534/ijeter/2021/26972021.

[23] Sherlinda Fadhillah Tanjung, "Implementasi Design Thinking Dalam Membangun Rancangan Ide Inovasi Model Bisnis Borneo Box: Aplikasi Distribusi Produk Pertanian Khusus Daerah Kalimantan Timur," *Inisiat. J. Ekon. Akunt. Dan Manaj.*, vol. 3, no. 2, pp. 286–294, Mar. 2024, doi: 10.30640/inisiatif.v3i2.2332.