



OPTIMIZING BEAUTY BRAND COMMUNITY MANAGEMENT THROUGH A LARAVEL-BASED WEB SYSTEM

Hanna Mariana Angelika¹⁾, Bhustomy Hakim²⁾

^{1,2} Information Systems, Universitas Bunda Mulia

^{1,2} Jl. Jalur Sutera Barat No. Kav. 7-9, Tangerang, Banten, 15143

Email: ¹hnnmariana@gmail.com, ²bhustomy.hakim@gmail.com

Abstract

Globalization has accelerated advancements in technology, communication, and information systems, intensifying competition across industries. In response, companies are adopting marketing communication strategies to build stronger relationships with consumers. One such strategy involves establishing brand communities, particularly within the beauty industry. In Indonesia, several beauty brands have established brand communities to enhance customer interaction and strengthen brand loyalty. One such brand, referred to here as DRS, specifically targets individuals aged 17 to 30. These communities typically offer initiatives such as point rewards, educational webinars, interactive challenges, product sampling, and early access to new products. However, DRS continues to manage its community manually through messaging platforms and basic data records. This manual approach results in several operational inefficiencies, including disorganized communication, data entry errors, and delays in program validation, all of which undermine administrative performance and reduce overall member satisfaction and engagement. To address these issues, this study proposes a web-based brand community management system with two main components. The first is a management interface used by admins and superadmins. The second is a user platform accessed by account users and members. Key features include program management, discussion forums, point redemption, leaderboards, and reporting functionalities. The system also incorporates principles from learning management systems to support continuous interaction, progress tracking, and sustained member engagement. It is developed using the Waterfall model of the software development life cycle, which supports structured planning and documentation. Laravel is selected as the development framework because of its efficiency, scalability, and model-view-controller architecture. The system is tested using the Black Box Testing method, and the results show that all functionalities operate according to the defined specifications. This system is expected to improve operational efficiency, enhance member experience and engagement, strengthen brand loyalty, enhance brand awareness and brand image, and support sustainable growth for beauty brands in Indonesia, particularly for brand DRS.

Keyword: Brand Community, Laravel, Beauty Industry, Waterfall, Web.

1. INTRODUCING

Globalization has marked significant progress in technology, communication, and information systems [1], profoundly influencing various aspects of life, particularly the economy, by intensifying competition among businesses [2]. In response to this growing competition, companies are increasingly adopting marketing communication as a strategic approach to foster closer relationships with consumers. Marketing communication functions as a tool to inform, persuade, and remind consumers about products or brands through both direct and indirect channels [3]. When executed effectively, it ensures that messages are clearly delivered, encourages purchasing decisions, and nurtures long-term consumer loyalty.

Each organisation develops a distinct communication strategy to achieve its business objectives, with some choosing to leverage brand communities as strategic communicators for promoting their products [4]. A brand community is defined as a group of individuals who share a strong emotional connection with a particular brand, regardless of their geographic location. These communities are built upon shared values, behaviours, and structured social interactions among members [5]. The rapid development of internet technology has accelerated the growth of online brand communities, enabling individuals with similar interests to connect and engage without spatial limitations [6].

In Indonesia, several beauty brands have established beauty brand communities to enhance customer engagement and strengthen brand loyalty. One such example is a brand referred to here as DRS, which primarily targets youth aged 17 to 30. Programs typically include point rewards, educational classes, interactive challenges, product reviews, and early



trials. However, DRS is still managed manually through messaging platforms and basic data records. This manual approach results in various operational inefficiencies, such as disorganized information, data entry errors, and delays in program validation, all of which affect administrative quality and user satisfaction. Key information, such as program schedules and guidelines, is shared through group chats, where important messages are often buried beneath newer ones, making it difficult for new members to trace earlier communications. Participation evidence is submitted individually via uploads to cloud storage and must be manually reviewed by admins before being recorded in spreadsheets. This process is time-consuming and prone to errors, duplication, or loss of data. The point redemption system is also not yet structured, requiring members to contact admins directly to check reward availability and await manual confirmation and point deduction. The absence of real-time visibility into point balances and reward stock further reduces operational efficiency and weakens the user experience. Additionally, the content submitted by members in response to challenges frequently fails to meet the expected standards and brief, potentially damaging the brand's image and influencing consumer purchasing decisions [7].

Given the increasing advancement of technology, there is a need for companies to adopt digital solutions that can streamline operations and enhance the overall experience of community members. This study aims to develop a web-based community management system by employing the Waterfall model of the Software Development Life Cycle. The Waterfall model is selected for its clear documentation and structured approach, which facilitates project tracking and supports systematic planning [8]. The system is developed using the Laravel framework, which is one of the highest-rated frameworks on GitHub, supported by more than 56,000 developers worldwide. Laravel adopts the Model-View-Controller architecture, which ensures separation between data processing, user interface, and application control [9]. It also accelerates development through the availability of preloaded libraries and tools [10].

The structure of this community management system resembles that of a Learning Management System. An LMS provides an environment that supports user engagement and learning progress. It allows users to register for classes, monitor their progress, and receive updates and announcements regarding course activities [11]. This is comparable to typical community programs such as webinars and interactive learning sessions, as well as point-based challenges that promote member participation. Moreover, the system's online accessibility ensures that users can engage with the platform anytime and with greater flexibility [12].

The objective of this research is to design and implement a web-based brand community management system that enhances the effectiveness and efficiency of community operations for beauty brands. Specifically, the system aims to support admins and community members by improving response time, minimising the risk of operational errors, and facilitating more informed organisational decision-making [13]. The system integrates key features such as program management covering educational sessions, interactive challenges, discussion forums, point redemption mechanisms, and a leaderboard to encourage member participation. Furthermore, it provides real-time activity reporting to support ongoing evaluation and member development monitoring. Through these capabilities, the system is intended to optimise operational workflows while enriching the user experience, thereby strengthening brand loyalty, increasing brand awareness and image, and ultimately contributing to sustainable business growth.

2. RESEARCH METHODS

This research adopts the Software Development Life Cycle (SDLC) methodology using the Waterfall model. Chosen for its structured process and thorough documentation, this approach supports clear project tracking and maintains clarity at every stage of development [8]. The standard stages within this model include requirements analysis, system design, implementation, testing, deployment, and maintenance. However, the scope of this research is limited to the testing phase. The following section outlines the phases conducted in this research, as illustrated in Figure 1.

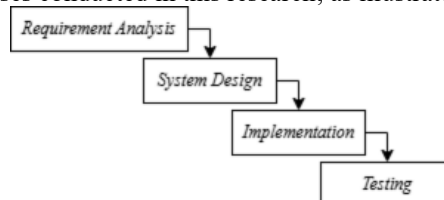


Figure 1. Waterfall Method

2.1 Requirement Analysis

The initial phase of system development is essential, requiring a clear understanding of user needs and expectations to ensure the software meets its intended purpose [14]. This phase involved data collection through observation and interviews. Observation allowed direct engagement in community activities to explore user behaviour, interaction patterns, and operational workflows. Meanwhile, interviews with community managers provided insights into the



community’s goals, strategic value, member characteristics, program types, activity structures, operational challenges, and essential system features to support more effective management.

2.2 System Design

After analysing the system requirements, the next phase involves designing the system and defining its core abstractions [14]. This stage outlines the main functionalities and interface design, specifies hardware and system needs, and establishes the overall system architecture [15]. In this research, the system design was carried out by developing UML diagrams and creating user interface designs, all of which were based on the previously formulated detailed specifications.

2.3 Implementation

Once the system design was completed, the process moved to the implementation phase, where the design was translated into functional code using suitable technologies [11]. Back-end development focused on system logic and data processing, while front-end development aimed to create an interactive and user-friendly interface based on user needs. Laravel was selected for its support of the Model View Controller (MVC) architecture, which ensures separation of concerns and facilitates easier system maintenance [9].

2.4 Testing

The final stage of this research was testing, a critical part of the system development process to ensure each feature operates as intended. The system was thoroughly evaluated to identify potential issues, such as logical errors or functional failures [14], ensuring that all components produce accurate and consistent results under various conditions [16]. Testing was conducted using the Black Box Testing method to ensure that all features performed correctly and supported effective community management. This approach, also known as behavioural testing, evaluates how the system responds to various inputs by comparing the results to the expected outcomes, without inspecting the internal structure or source code [17].

3. RESULT AND DISCUSSIONS

3.1 System Design

The following provides an overview of the use case diagram, which illustrates the system functions and user interactions according to their assigned roles [18].

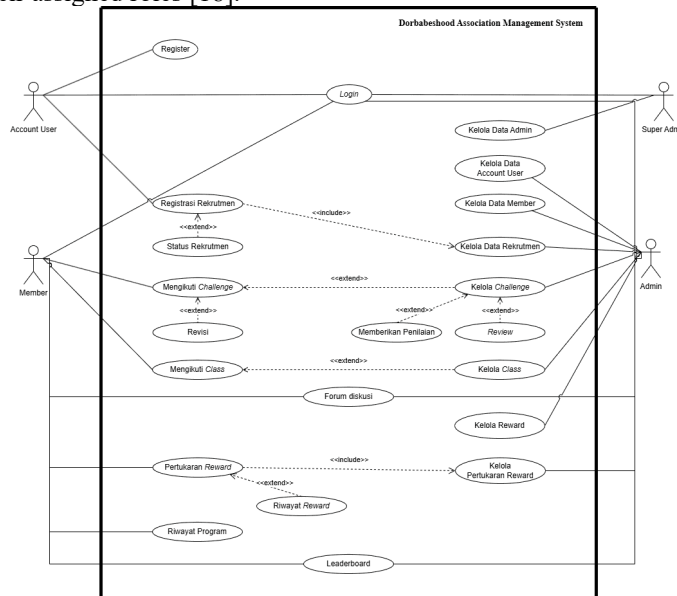


Figure 2. Use Case Diagram

Figure 2 illustrates four main actors within the system, namely Account User, Member, Admin, and Superadmin. Account Users refer to individuals who have registered but have not yet joined the community, while Members are users who have formally become part of it. Admins are responsible for managing operational tasks, whereas Superadmins oversee system configuration and overall control. Each actor is associated with specific features according to their access



level and assigned responsibilities. The diagram defines how each user interacts with the system and serves as a foundation for understanding its overall functional structure.

3.2 Implementation

Based on the results of the system analysis and design, the following presents the implementation outcome of the web-based brand community management system.

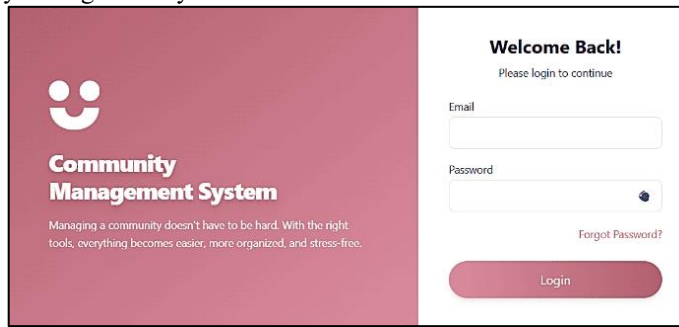


Figure 3. Login Page UI

Figure 3 shows the login page, which manages access for account users, members, admins, and superadmins. It features input fields for email and password, a toggle for password visibility, and a link for password recovery. System access is restricted to admins with active status, while those marked as inactive are denied login. Upon successful authentication, users are redirected to their respective dashboard or homepage according to their role.

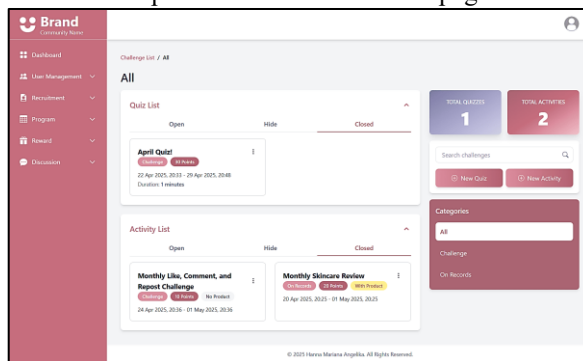


Figure 4. Challenge List Page UI (Superadmin and Admin Side)

Figure 4 manages two types of challenges, which are quizzes and activities. Each has its own section with tabs for open, hide, and closed status. Cards display challenge details with edit and delete options. The page also includes search statistics and category filters. The class list page uses the same structure.

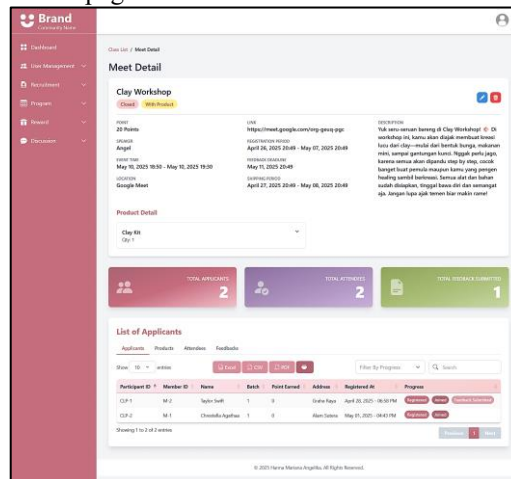


Figure 5. Class Detail Page UI (Superadmin and Admin Side)



Figure 5 provides an overview of class information, including total participants, attendance, and feedback. It includes tabs for applicants, attendees, and feedback, all presented in table format with search and export features. A products tab displays shipping data in card format, showing delivery status and related details. The same layout is applied to the quiz detail page, which provides an overview of quiz information with general details at the top and summary cards in the centre. It includes four tabs, namely questions, summary of answers, validation of answers and grading. The questions tab lists each item along with its type, score, and available options. Answer summaries are visualised using doughnut charts for multiple-choice questions and text listings for essay responses. Validation and grading tabs present participant data, response status, and scores in structured tables. The activity detail page also adopts a similar layout, providing an overview of activity information with general details and summary cards. There are three tabs available, namely participants, products, and submissions. The products tab functions the same as in the class page, displaying shipping data in card format, including delivery status and shipping details. The submissions tab displays uploaded files, whether draft or final, along with timestamps, statuses, and comments. Final submissions allow admins to input points and assign scores directly. This consistent structure across class, quiz, and activity pages ensures a user-friendly and efficient experience for managing various types of program content and evaluations.

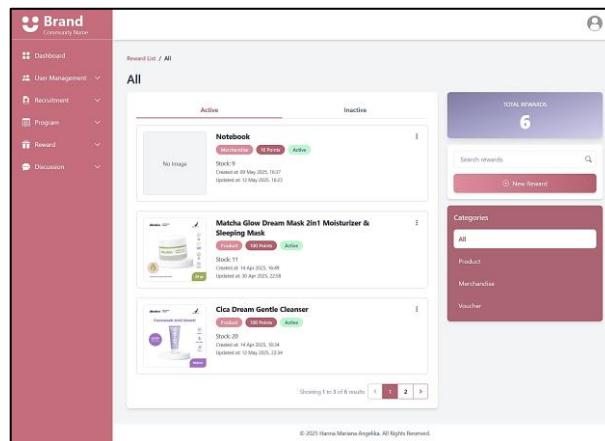


Figure 6. Reward Management Page UI (Superadmin and Admin Side)

Figure 6 shows the reward list in card format with filters for active and inactive status. Each card includes reward details, edit and delete options, total statistics, and a search bar.

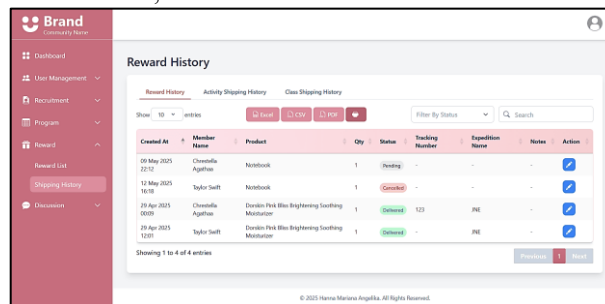


Figure 7. Reward History Management Page UI (Superadmin and Admin Side)

Figure 7 shows the shipping history with three tabs, which are reward history, activity shipping history, and class shipping history. Each tab presents shipment records in table format with search status filters and export functions.

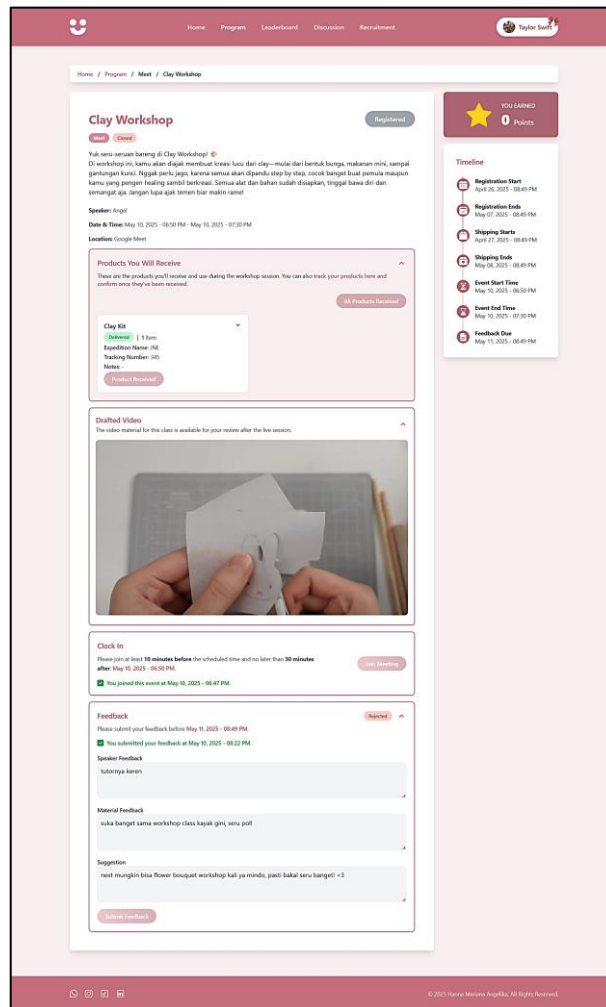


Figure 8. Class Detail Page UI (Member Side)

Figure 8 shows class information, including a register button and a complete description. The products section displays shipping status. A drafted video section appears if required. The clock in section provides a join meeting button, and the feedback section allows users to submit their input. A points card and program timeline are displayed on the right side of the page. The quiz detail page follows a similar layout. It displays the quiz title, category, status, grading status, description, and duration. Once the quiz begins, the page presents the questions along with a timer, question type, point value, and answer status. A points card, timeline, and answer statistics are also shown on the right. The activity detail page also uses the same layout. It presents activity information, including a registration button and timing indicators. The products section shows delivery status along with a product received button. Both draft and final submissions are displayed with file uploads, status, and comments. A points card and timeline appear on the right, providing a consistent and user-friendly experience across different types of program detail pages.

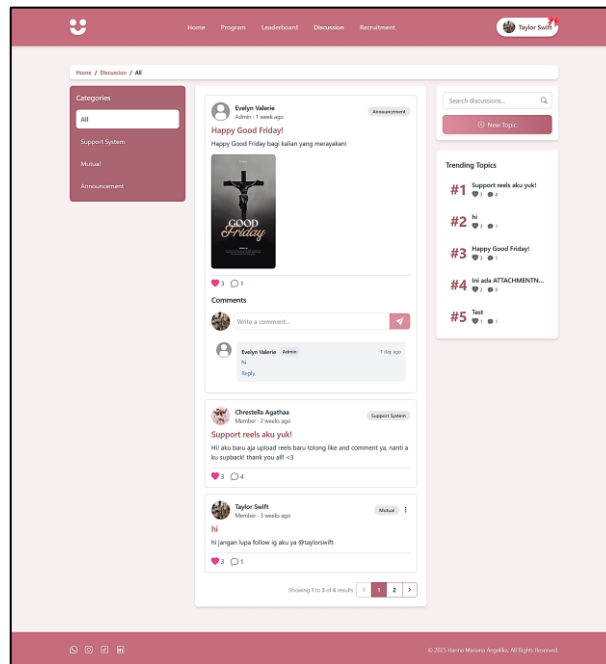


Figure 9. Discussion Page UI (Member Side)

Figure 9 shows a discussion list with a category filter on the left. Each post displays the creator timestamp, category, content, attachments, and buttons for like and comment. Users can edit or delete their own posts. A search bar and a new topic button appear at the top right, with trending topics shown at the bottom. The admin view follows the same structure.

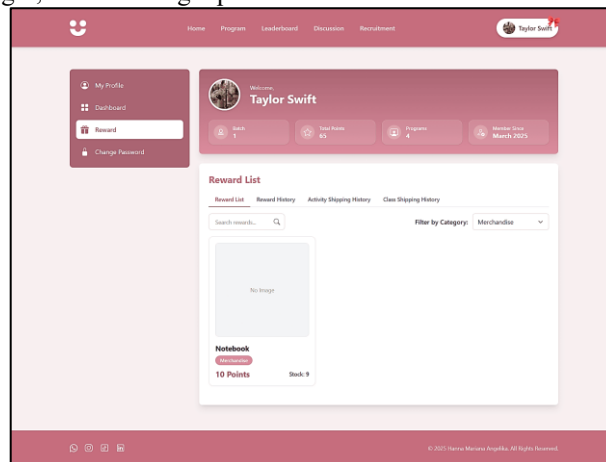


Figure 10. Reward List Page UI (Member Side)

Figure 10 functions as the point redemption page. A profile summary is shown at the top right, and below are four tabs, which are reward list, reward history, activity shipping history, and class shipping history. The reward list displays product cards with category filters and a search bar. Selecting a card opens a detailed modal.

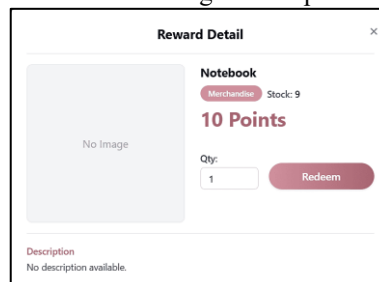


Figure 11. Reward Detail Modal UI (Member Side)



Figure 11 shows a modal that appears when a user selects a reward card. It displays product image, name, category, stock required points, and a brief description. Users can enter the quantity to redeem from the available stock. If the points are insufficient, the redeem button remains inactive. Once the points are sufficient, the user can continue with the redemption process.

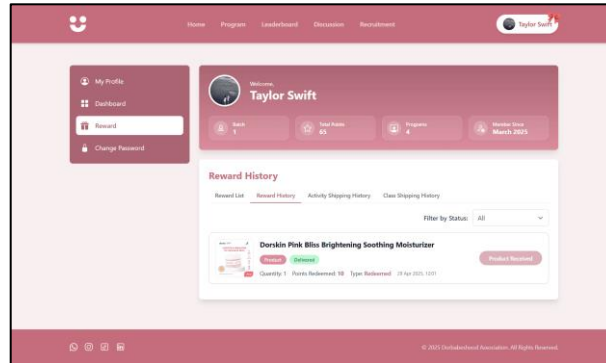


Figure 12. Reward History Page UI (Member Side)

Figure 12 shows the reward history tab, which follows the same layout as activity and class shipping history. It features status filters and product cards showing image name, category, delivery status, quantity points used, transaction type, and date. Cancelled and rejected statuses are excluded from the activity and class views. The cancel button only appears in reward history when the status is still pending, and within one day after which it changes to product received.

3.3 Black Box Testing

This study adopted the Black Box Testing method to evaluate the functionality and reliability of the developed web-based brand community management system. Table 1 presents the detailed results of the Black Box Testing process, which consists of six components. These include the system module being tested, a description of the test case, the test steps performed, the expected results based on the functional requirements, the actual results observed during testing, and the final status indicating whether each test was passed or failed.

Table 1. Black Box Testing

Module	Test Case	Test Steps	Expected Results	Actual Results	Status
Login	Login using a valid email and password	<ol style="list-style-type: none"> 1. Enter a valid email and password 2. Click the Login button 	The user is redirected to the dashboard/home page	The user is redirected to the dashboard/home page	Passed
Program management	Add class detail data	<ol style="list-style-type: none"> 1. Check the With Product checkbox 2. Fill in 14 class input fields 3. Fill in 6 product-related input fields 4. Click the Save 	Both class detail and class product data are added to the database	Both class detail and class product data are added to the database	Passed
	Delete class detail data	<ol style="list-style-type: none"> 1. Click the three-dot icon on the class card 2. Click the Delete icon 3. Confirm deletion by clicking Yes, delete it 	The class detail is removed from the database	The class detail is removed from the database	Passed
	Edit class detail data	<ol style="list-style-type: none"> 1. Click the three-dot icon on the class card 2. Click the Edit icon 3. Modify the desired data 4. Click the Save button 	The class detail is updated	The class detail is updated	Passed
	Edit the class shipping history data	<ol style="list-style-type: none"> 1. Click the Edit icon 2. Enter expedition detail, status, and notes in the class product list section 3. Click the Save button 	The shipping history data is updated	The shipping history data is updated	Passed



Module	Test Case	Test Steps	Expected Results	Actual Results	Status
	Change the class feedback answer status to Accepted	<ol style="list-style-type: none"> 1. Click the status dropdown 2. Select the Accepted option 	The status is updated to Accepted, and the member's points increase by the class point value	A success message is displayed, the status is updated to Accepted, and the member's points increase by the class point value	Passed
	Add activity detail data	<ol style="list-style-type: none"> 1. Click the New Class button 2. Check the With Product checkbox 3. Fill in 9 activity input fields 4. Fill in 5 product-related input fields 5. Click the Save button 	Both the activity detail and activity reward data are added to the database	Both the activity detail and activity reward data are added to the database	Passed
	Delete activity detail data	<ol style="list-style-type: none"> 1. Click the three-dot icon on the activity card 2. Click the Delete icon 3. Confirm deletion by clicking Yes, delete it 	The activity detail is removed	The activity detail is removed	Passed
	Edit activity detail data	<ol style="list-style-type: none"> 1. Click the three-dot icon on the activity card 2. Click the Edit icon 3. Modify the desired data 4. Click the Save button 	The activity detail is updated	The activity detail is updated	Passed
	Edit the activity shipping history data	<ol style="list-style-type: none"> 1. Click the Edit icon 2. Enter expedition name, tracking number, status, and notes in the activity product list section 3. Click the Save button 	The shipping history data is updated	The shipping history data is updated	Passed
	Update the point for activity submission with the Final type	<ol style="list-style-type: none"> 1. Enter the point value 2. Click the +Point button 	The Final submission point is updated, and the point is added to the member's total points	The Final submission point is updated, and the point is added to the member's total points	Passed
	Add quiz detail data	<ol style="list-style-type: none"> 1. Click the New Quiz button 2. Fill in 13 input fields, including quiz details and questions 3. Click the Save Quiz & Question button 	The quiz detail is added to the database	The quiz detail is added to the database	Passed
	Delete quiz detail data	<ol style="list-style-type: none"> 1. Click the three-dot icon on the quiz card 2. Click the Delete icon 3. Confirm deletion by clicking Yes, delete it 	The quiz detail is deleted	The quiz detail is deleted	Passed
	Edit quiz detail data	<ol style="list-style-type: none"> 1. Click the three-dot icon on the quiz card 2. Click the Edit icon 3. Modify the desired data 4. Click the Save button 	The quiz detail is updated	The quiz detail is updated	Passed
	Validate the correct answer	Click the checkbox under the Correct column	The quiz answer status changes to Correct, and the question point is added to the quiz	The quiz answer status changes to Correct, and the question point is added to the quiz	Passed



Module	Test Case	Test Steps	Expected Results	Actual Results	Status
			grading once all answers are validated	grading once all answers are validated	
	Validate the incorrect answer	Click the checkbox under the Incorrect column	The quiz answer status changes to Incorrect	The quiz answer status changes to Incorrect	Passed
Reward list and reward history management	Add reward data	<ol style="list-style-type: none"> Click the New Reward button Fill in 8 input fields Click the Save button 	The reward data is added to the database	The reward data is added to the database	Passed
	Delete reward data	<ol style="list-style-type: none"> Click the three-dot icon on the reward card Click the Delete icon Confirm deletion by clicking Yes, delete it 	The reward data is deleted	The reward data is deleted	Passed
	Edit reward data	<ol style="list-style-type: none"> Click the three-dot icon on the reward card Click the Edit icon Modify the desired data Click the Save button 	The reward data is updated	The reward data is updated	Passed
	Delete reward history data	<ol style="list-style-type: none"> Click the Delete icon on the selected reward history Confirm deletion by clicking Yes, delete it 	The reward history data is deleted	The reward history data is deleted	Passed
	Edit reward history data	<ol style="list-style-type: none"> Click the Edit icon on the selected reward history Modify the desired data Click the Save button 	The reward history data is updated	The reward history data is updated	Passed
	Change reward history status to Rejected	<ol style="list-style-type: none"> Click the Edit icon on the selected reward history Open the status dropdown Select the Rejected option Click the Save button 	The status is updated to Rejected, and the member's points and reward stock are restored	The status is updated to Rejected, and the member's points and reward stock are restored	Passed
	Discussion	Add a discussion with an attachment	<ol style="list-style-type: none"> Click the New Topic button Enter category, title, content, and upload an attachment Click the Post button 	The discussion is added to the database	The discussion is added to the database
Delete discussion		<ol style="list-style-type: none"> Click the three-dot icon on the discussion card Click the Delete icon Confirm deletion by clicking Yes, delete it 	The discussion is deleted	The discussion is deleted	Passed
Edit discussion		<ol style="list-style-type: none"> Click the three-dot icon on the discussion card Click the Edit icon Modify the content Click the Save button 	The discussion is updated	The discussion is updated	Passed
Like discussion		Click the heart icon with a grey border on the discussion card	The heart icon turns solid red, and a like is recorded in the database.	The heart icon turns solid red, and a like is recorded in the database.	Passed



Module	Test Case	Test Steps	Expected Results	Actual Results	Status
	Add a comment to the discussion	<ol style="list-style-type: none"> 1. Enter a comment in the input field 2. Click the Send icon 	The comment is saved in the database and appears in the comment list	The comment is saved in the database and appears in the comment list	Passed
Program detail	Register for an Open class	Click the Register Now button	The member is registered, data is added to the class participant database, and the Clock-In and Feedback sections appear	The member is registered, data is added to the class participant database, and the Clock-In and Feedback sections appear	Passed
	Confirm product received	Click the Product Received button on the product list card	Product shipping status for the member is updated to Delivered in the class shipping history	Product shipping status for the member is updated to Delivered in the class shipping history	Passed
	Attend a class (Online or Onsite)	Click the Join Meeting button	Redirects to the meeting link for online classes or logs check-in data for onsite classes.	Redirects to the meeting link for online classes or logs check-in data for onsite classes.	Passed
	Submit class feedback	<ol style="list-style-type: none"> 1. Fill in speaker feedback, material feedback, and suggestions 2. Click the Submit Feedback button 	Feedback data is saved to the class feedback answer database	Feedback data is saved to the class feedback answer database	Passed
	Register for an Open activity	Click the Register Now button	The member is registered, data is saved, and the Draft and Final Submission sections are displayed.	The member is registered, data is saved, and the Draft and Final Submission sections are displayed.	Passed
	Confirm product received	Click the Product Received button on the product list card	Product shipping status in the reward history is updated to Delivered	Product shipping status in the reward history is updated to Delivered	Passed
	Submit a draft submission	<ol style="list-style-type: none"> 1. Select submission option (URL or attachment) 2. Fill in the URL or upload an attachment 3. Click the Submit Draft button 	Draft submission data is added to the database and appears in the Draft Submission section	Draft submission data is added to the database and appears in the Draft Submission section	Passed
	Submit the final submission	<ol style="list-style-type: none"> 1. Select submission option (URL or attachment) 2. Fill in the URL or upload an attachment 3. Click the Submit Final button 	Final submission data is added to the database and appears in the Final Submission list	Final submission data is added to the database and appears in the Final Submission list	Passed
	Start quiz	Click the Start Quiz button	Quiz questions and a timer are displayed, and data is added to the quiz grading database	Quiz questions and a timer are displayed, and data is added to the quiz grading database	Passed
	Submit quiz answers	<ol style="list-style-type: none"> 1. Fill in all quiz questions 2. Click the Submit button 	Answers are submitted and saved to the quiz answer database	Answers are submitted and saved to the quiz answer database	Passed
Reward list and reward history	Redeem a reward	<ol style="list-style-type: none"> 1. Click the reward card to view details 2. Enter quantity 3. Click the Redeem button 	The reward history is saved, and member points and stock are reduced	The reward history is saved, and member points and stock are reduced	Passed



Module	Test Case	Test Steps	Expected Results	Actual Results	Status
		Confirm by clicking Yes, redeem it			
	Cancel reward redemption	1. Click the Cancel button 4. Confirm cancellation by clicking Yes, cancel it	The reward history status is updated to Cancelled, and member points and reward stock are restored	The reward history status is updated to Cancelled, and member points and reward stock are restored	Passed
	Confirm reward received	4. Click the Product Received button	The reward history status is updated to Delivered	The reward history status is updated to Delivered	Passed

Based on the results of Black Box Testing, all test cases were executed successfully and produced outcomes that aligned with the predefined expectations. A total of 40 test cases were conducted across 6 system modules, and all of them passed, resulting in a 100% success rate. As shown in Table 1, these results indicate that the system performs reliably across all tested functionalities and effectively supports its intended operational objectives.

4. CONCLUSION

This study produced a web-based brand community management system designed to improve the efficiency of community operations for beauty brands in Indonesia. The system comprises two primary components which are a community management interface for admins and superadmins, and a community platform designed for account users and members.

The system addresses several key issues currently experienced by brand DRS in the manual management of its brand community. First, program schedules and guidelines are communicated through group chats, where important messages are often buried beneath newer ones. This makes it difficult for members, especially those who are new, to access essential information. The proposed system offers a solution by delivering programs through the platform and providing a structured discussion feature with category filters to improve message organisation and accessibility. Second, participation evidence is submitted individually via uploads to cloud storage and must be manually reviewed by admins before being recorded in spreadsheets. The system responds to this by allowing participation data to be stored directly in a database and displayed within the system interface, where admins only need to validate the entries for automatic point allocation.

Third, the point redemption process also remains unstructured, requiring members to contact admins directly to check reward availability and confirm point deductions. The proposed system addresses this by enabling members to redeem rewards directly based on their current point balance, with real-time visibility into available rewards and a transparent reward history. Another ongoing issue involves the quality of content submitted for challenges, which often does not meet established expectations or brief requirements. To help address this, the system includes a draft submission feature that allows members to submit initial versions of their content, receive feedback, and revise their submissions before final approval. This encourages content that is more aligned with brand standards and helps maintain a positive brand image.

The system was tested using the Black Box Testing method, and all functionalities operated as intended. Overall, this system aims to optimise community management while enhancing member experience and engagement, which in turn strengthens loyalty, improves brand awareness and brand image, and supports sustainable business growth.

ACKNOWLEDGEMENT

The author gratefully acknowledges the support received from all individuals and institutions who contributed, whether directly or indirectly, to the completion of this research. Appreciation is also extended to Universitas Bunda Mulia and to the lecturers who have provided valuable insights, guidance, and constructive feedback throughout the research process. The author further expresses thanks to the brand referred to as DRS for the opportunity, resources, and time that made the successful implementation of this study possible.

REFERENCES

- [1] D. U. Sari, "Strategi Komunikasi Pemasaran PT POS Indonesia Sebagai Jasa Pengiriman Pada Era Globalisasi," *Journal of Media and Communication Science*, pp. 97-104, 2024.
- [2] C. D. Maulidasari and D. , "Dampak Promosi Produk pada Pemasaran Online," *Jurnal Bisnis dan Kajian Strategi Manajemen*,



- pp. 137-142, 2021.
- [3] A. Hermawati, "Pengaruh Penanganan Keluhan, Komunikasi Pemasaran dan Kepercayaan terhadap Loyalitas (Literature Review Manajemen Pemasaran)," *Jurnal Manajemen dan Pemasaran Digital (JMPD)*, pp. 14-28, 2023.
 - [4] F. F. F. H. Rambe, "Pengaruh Brand Community Emina Girl Gang Ambassador (EGGA) Terhadap Brand Loyalty di Kalangan Konsumen Emina Cosmetics," 2024.
 - [5] D. Mardhiyah, A. I. Futuwwah, M. F. Hilmi, D. O. Rosidi and N. H. Zain, "The Relationship of Online Brand Community, Value Creation Practices, Brand Trust, and Brand Loyalty," *Journal of Theoretical and Applied Management*, pp. 473-490, 2023.
 - [6] D. D. Damayanti, "Literature Review Brand Loyalty Melalui Brand Community Online," *JIMBis : Jurnal Ilmiah Manajemen dan Bisnis*, pp. 202-213, 2023.
 - [7] N. K. S. R. Indriyani, N. L. K. A. S. Sucandrawati and N. P. A. D. Laksmi, "Kualitas Konten Digital Marketing Dan Pengaruhnya Terhadap Brand Image Dan Keputusan Pembelian Konsumen Di Kota Denpasar," *KOLONI: Jurnal Multidisiplin Ilmu*, pp. 816-822, 2022.
 - [8] S. Pargaonkar, "A Comprehensive Research Analysis of Software Development Life Cycle (SDLC) Agile & Waterfall Model Advantages, Disadvantages, and Application Suitability in Software Quality Engineering," *International Journal of Scientific and Research Publications*, pp. 120-124, 2023.
 - [9] R. Y. Endra, Y. Aprilinda, Y. Y. Dharmawan and W. Ramadhan, "Analisa Perbandingan Bahasa Pemrograman PHP Laravel dengan PHP Native pada Pengembangan Website," *EXPERT: Jurnal Manajemen Sistem Informasi dan Teknologi*, pp. 48-55, 2021.
 - [10] S. M. P. Kharisma, M. A. Assuja and G. Pramitha, "Pelatihan Pemrograman Web Menggunakan Framework Laravel di SMKN 1 Bandar Lampung," *Journal of Engineering and Information Technology for Community Service (JEIT-CS)*, vol. 1, no. 2, pp. 87-94, 2022.
 - [11] V. M. Bradley, "Learning Management System (LMS) Use with Online Instruction," *International Journal of Technology in Education*, vol. 4, no. 1, pp. 68-92, 2021.
 - [12] Z. Alawi and A. R. Zayn, "Perencanaan Manajemen Proyek dalam Pengembangan Learning Management System Menggunakan Trello," *Teknologi: Jurnal Ilmiah Sistem Informasi*, vol. 12, no. 2, pp. 1-7, 2022.
 - [13] J. Santoso, A. Nugroho and I. W. K. Utama, "Perancangan Sistem Informasi Komunitas Anggota Paguyuban (SIKAP) Berbasis Web pada Paguyuban Ngeksigondo," *Jurnal Teknologi Informasi Dan Komputer*, vol. 7, no. 3, pp. 228-234, 2021.
 - [14] W. Harjono and K. J. Tute, "Perancangan Sistem Informasi Perpustakaan Berbasis Web Menggunakan Metode Waterfall," *Jurnal Sains Teknologi dan Sistem Informasi*, vol. 2, no. 1, pp. 47-51, 2022.
 - [15] V. Olindo and A. Syaripudin, "Perancangan Sistem Informasi Absensi Pegawai Berbasis Web Dengan Metode Waterfall (Studi Kasus : Kantor DBPR Tangerang Selatan)," *OKTAL: Jurnal Ilmu Komputer dan Science*, vol. 1, no. 1, pp. 17-26, 2022.
 - [16] K. and M. Badrul, "Penerapan Metode Waterfall untuk Perancangan Sistem Informasi Inventory pada Toko Keramik Bintang Terang," *Jurnal Pengembangan Riset dan Observasi Sistem Komputer (PROSISKO)*, vol. 8, no. 2, pp. 47-52, 2021.
 - [17] S. J. Putri, D. G. P. Putri and W. H. N. Putra, "Analisis Komparasi pada Teknik Black Box Testing (Studi Kasus: WebsiteLars)," *Journal of Internet and Software Engineering (JISE)*, vol. 5, no. 1, pp. 23-28, 2024.
 - [18] H. Pratiwi and I. A. Mastan, "Perancangan Enterprise Architecture Sistem Informasi Perusahaan Gadai Menggunakan The Open Group Architecture Framework (TOGAF)," *Journal of Business and Audit Information Systems*, pp. 41-51, 2022.