Journal of Computing and Smart Ecosystem
Universitas Muhammadiyah Semarang
Vol. 2 No.1, 2025: 1-16

J-CaSE
e-ISSN 3110-5777

Moodymap: An Interactive Map Application For Saving User Stories And Moods Based On Location

Naimatul Husna[1], Basirudin Ansor
[1] Department of Information Technology, Faculty of Computer Science and Digital Technology, Muhammadiyah University of Semarang, Indonesia

***Corresponding author:** naimaatulhusna@gmail.com

Article Info:
Received: Augst 6, 2025
Accepted: January 27, 2026
Available Online: June 30, 2026

Keywords:
Interactive Map; Story; Mood; MoodyMap; Leaflet.js
.

Abstract: MoodyMap is an interactive map application designed to allow users to record and revisit personal stories and moods based on geographical locations. The application was developed using the Software Development Life Cycle (SDLC) with an Agile approach, utilizing Laravel and Bootstrap as development frameworks and Leaflet.js for interactive map visualization. MoodyMap provides several main features, including user registration and authentication, adding location-based memories with automatic coordinates, mood representation using emojis, photo uploads, memory management, memory history, profile management, and password reset via email. Functional testing using black-box testing showed that the main features operated according to the expected results. The developed application demonstrates that integrating geographical information, personal narratives, and mood representation can provide a more contextual way for users to document and revisit their location-based memories.

1. INTRODUCTION

In the current digital age, storing memories is no longer limited to text and images but can also be integrated with a geographical context. Using interactive maps to store user stories provides a deeper and more contextual experience. While popular applications like Google Maps include location history features, they don't specifically accommodate recording mood or rich, personal stories.

The development of MoodyMap was motivated by the need for a personal and intuitive platform that enables users to document memories together with their geographical and emotional contexts. Unlike conventional memory-recording applications that mainly store text or images, MoodyMap allows users to associate each memory with a specific location, date, story, photo, and mood represented by an emoji. This combination enables users to revisit not only what happened but also where the experience occurred and how they felt at that moment.

The novelty of MoodyMap lies in the integration of interactive geographical visualization with personal narratives and explicit mood representation. While existing mapping applications primarily focus on navigation, places, or location history, MoodyMap

[DOI: 10.14710/JCASE. vol2.iss1.791]
OPEN ACCESS
CC BY SA

uses geographical coordinates as part of a personal memory-recording mechanism. Therefore, the research addresses a gap in conventional memory-logging applications by combining the spatial, narrative, and emotional dimensions of personal experiences within a single web-based application.This research's novelty lies in integrating an interactive map feature for spatially visualizing memories, combined with the ability to reflect mood through emojis. This feature distinguishes MoodyMap from conventional memory-logging applications, which often lack a geographical dimension or explicit emotional representation. The research gap that MoodyMap addresses is its focus on not just recording "what" happened, but also "where" and "how the user felt" at that moment.

[DOI: 10.14710/JCASE. vol2.iss1.791]

2. METHODOLOGY

Commented [A1]: The section is comprehensive, but the subheadings (e.g., "2.1 Requirements Definition") would be better aligned as level-2 or level-3 headings, not treated like paragraphs.

The MoodyMap application was developed using the Software Development Life Cycle (SDLC) with an Agile approach. Agile emphasizes iterative and collaborative software development through a series of short development cycles. This approach enables requirements, designs, and application features to be refined continuously based on evaluation and user feedback during the development process. The Agile approach was selected because MoodyMap requires flexibility in developing interactive features and user interfaces while allowing improvements to be implemented in each iteration. The development stages applied in this study include requirements definition, system and software design, implementation and unit testing, and operation and maintenance. The Agile development cycle used in this study is illustrated in Image 1.

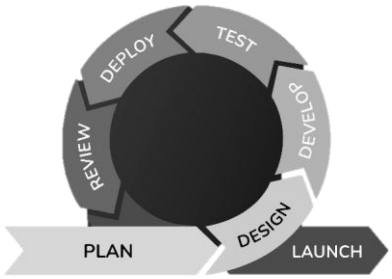


Image 1. *Agile* Methods

2.1. Requirements Definition

Commented [A2]: The requirement analysis is detailed. However, listing the sources of user input (interviews, observations) could be more structured. Consider separating user research methods and UI/UX considerations into separate paragraphs.

This stage aims to identify the functional and non-functional requirements required for the development of the MoodyMap application. The requirement-gathering process was conducted by observing user behavior, conducting informal interviews with potential users, and studying similar web-based applications related to interactive maps, journeys, and location-based memories. Applications such as Google Maps, StoryMapJS, and location-based social media platforms were also reviewed to understand common interaction patterns and features that could support the development of MoodyMap.

In addition to identifying user requirements, UI/UX considerations were incorporated into the requirement analysis. These considerations included interface responsiveness, simple navigation, light visual interactions, and the use of Leaflet.js to provide an interactive map experience. The interface was designed to make the process of adding, viewing, and managing location-based memories easy to understand and accessible across different screen sizes.

The results of the requirement analysis were then classified into functional and non-functional requirements. Functional requirements define the main activities that users can perform within the application, while non-functional requirements describe the technical and operational conditions needed to support the system. These requirements serve as the basis for the subsequent system design and implementation stages.

Table 1. Functional and Non-Functional Requirements Table

Functional Requirements	Non-Functional Requirements
▪ User account registration and login ▪ Reset password via email ▪ View location map and memories	▪ Minimum 2 GB RAM device ▪ Stable internet connection

[DOI: 10.14710/JCASE. vol2.iss1.791]

- Add new memories (title, description, mood, date, photo, location)
- View memory detail
- Edit and delete memories
- View personal memory history
- View and edit profile
- Logout from account

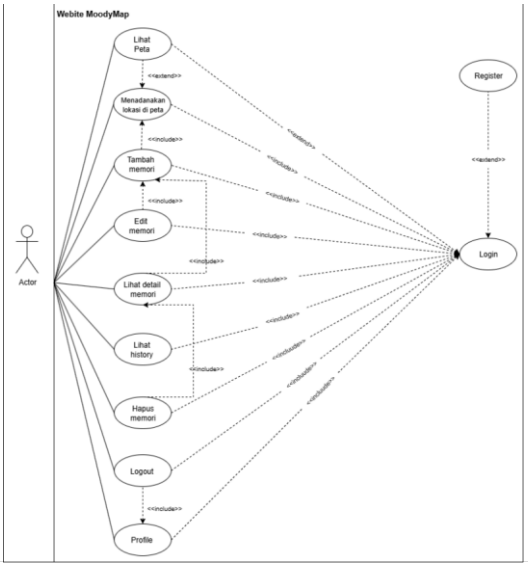
- Modern web browser (Chrome, Firefox, Edge, etc)
- Responsive on various screen size
- Data stored in a database (MySQL)

2.2. System and Software Design

This stage involved creating the initial design for the MoodyMap application based on the requirements analysis. The design is presented in the form of a *Use Case* Diagram, a *Class Diagram*, and user interface designs for each main application page. The purpose of this stage is to translate software requirements into a design representation that can be properly implemented during the development process. This design was also created to minimize the risk of misunderstanding by the development team or other parties regarding the system's flow and procedures, and to serve as a form of visual planning documentation that clarifies the overall system requirements.

2.2.1 *Use Case* Diagram

A *use case* diagram is a graphic representation of several actors who play the role of users in a system and what users can do. A *use case* diagram also shows the relationship between actors and the system (Rumbaugh et al., 2004). The web plays a fundamental role as an institutional communication tool and is key in representing an organization as an entity (Daries, Cristobal-Fransi, & Ferrer-Rosell, 2021). Image 2 shows a *use case* diagram.



[DOI: 10.14710/JCASE. vol2.iss1.791]

Image 2. *Use Case* Diagram

In the MoodyMap system, users can access the main application functionalities after completing the registration or login process. Authenticated users can view the interactive map, select a geographical location, and create a memory associated with the selected coordinates. Each stored memory can subsequently be viewed in detail, edited, or deleted. Users can also access the memory history page to review previously recorded memories and use the profile page to view or update their account information. The system also includes logout functionality as part of session management. For new users, the registration functionality is provided to create an account before accessing the application's authenticated features.

2.2.2 *Class* Diagram

A *Class* Diagram is a type of *Unified Modeling Language* (UML) diagram used to model a system's static structure. This diagram illustrates the classes in the system, their attributes, methods/functions, and the relationships between classes, such as association, inheritance, and dependency. *Class* diagrams are crucial in the design phase because they help developers understand how data structures and object logic will be organized in the program's implementation.

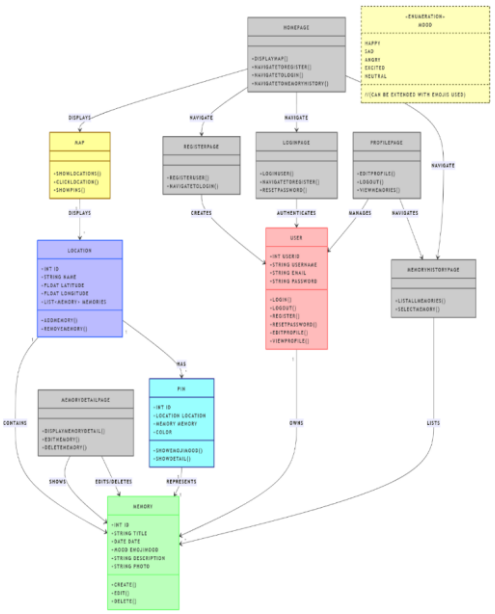


Image 3. *Class* Diagram

2.2.3 Application Interface Design

a. Home Page

This interface is the main display of the website. It contains an interactive world map implemented using the Leaflet.js library. The map can be zoomed, panned, and clicked to select a location.

[DOI: 10.14710/JCASE. vol2.iss1.791]

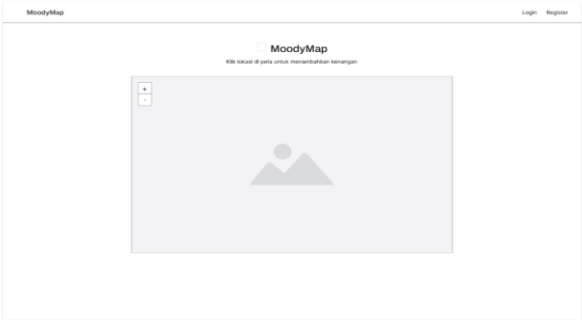


Image 4. Home Page

b. Profile Page

The profile page is designed to contain a user's profile photo at the top and user data below.

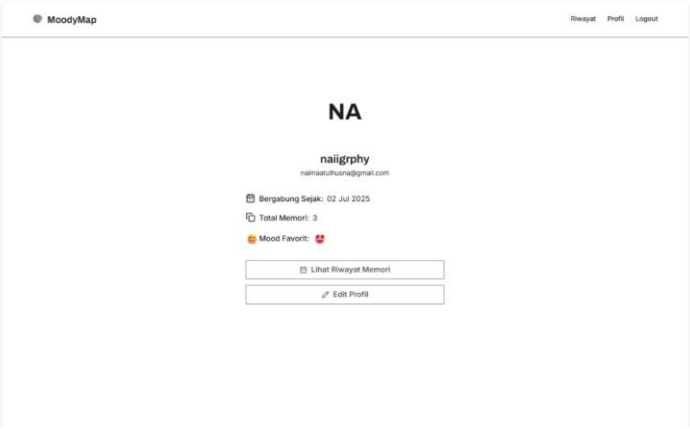


Image 5. Profile Page

c. Login Page

The login page is used for authentication into the system. Users are asked to enter their email and password from the data they have registered.

[DOI: 10.14710/JCASE. vol2.iss1.791]

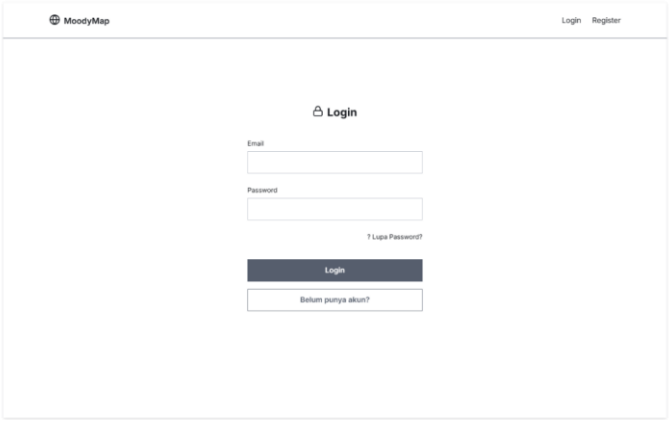


Image 6. Login Page

2.3 Implementation and Unit Testing

At this stage, the system design was translated into a functional web application through the implementation of program code. MoodyMap was developed using PHP, JavaScript, HTML, CSS, and MySQL, with Laravel used as the main web application framework. Leaflet.js was implemented to provide interactive map functionality, while Bootstrap was used to support responsive user interface development.

Visual Studio Code was used as the primary code editor during the implementation process, while XAMPP provided the local development environment and database services required for system development and testing. The implementation process focused on translating the previously defined requirements and system designs into functional application modules.

After implementation, each major functionality was examined to ensure that the application behaved according to the defined requirements. Functional verification was subsequently performed through black-box testing, including registration, login, password reset, memory creation and management, profile management, map interaction, and navigation functions.

2.4 Operation and Maintenance

The operation and maintenance stage is conducted after the application has been implemented and evaluated. This stage aims to ensure that MoodyMap continues to operate properly and remains consistent with user requirements. Maintenance activities include identifying and fixing application errors, monitoring system functionality, performing database backups, maintaining stored user and memory data, and updating application features when improvements are required.

Maintenance may also include adjustments to the user interface, compatibility updates, security improvements, and modifications to existing functionality based on evaluation results or future user requirements. These activities are important because software maintenance is

[DOI: 10.14710/JCASE. vol2.iss1.791]

required to correct errors and adapt the application to changing operational conditions after implementation (Mujab, Satoto, & Martono, 2014)..

3. RESULT AND DISCUSSION

3.1 Home Page

This page contains an interactive world map implemented using the Leaflet.js library. The map can be zoomed, panned, and clicked to select a location. If a user does not have an account or is not logged in, the map is still displayed, but it does not show any memory location pins. A user who clicks on a location on the map while not logged in will be redirected to the registration page. After logging in, the home page will display blue pins that show the locations of the user's memories. Each pin also shows an emoji corresponding to the mood selected when the memory was added.

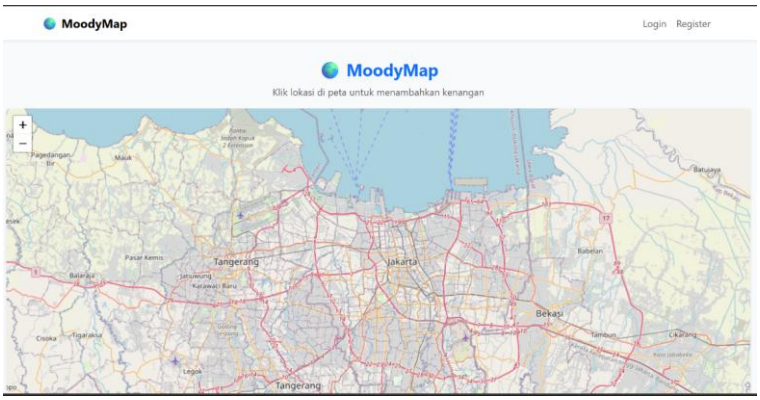


Image 7. Home Page

3.2 Register Page

This interface is used to register a new account. Users are asked to fill in their full name, email, and create a password and password confirmation. After a successful registration process, the user will be automatically logged in and redirected to the home page.

[DOI: 10.14710/JCASE. vol2.iss1.791]

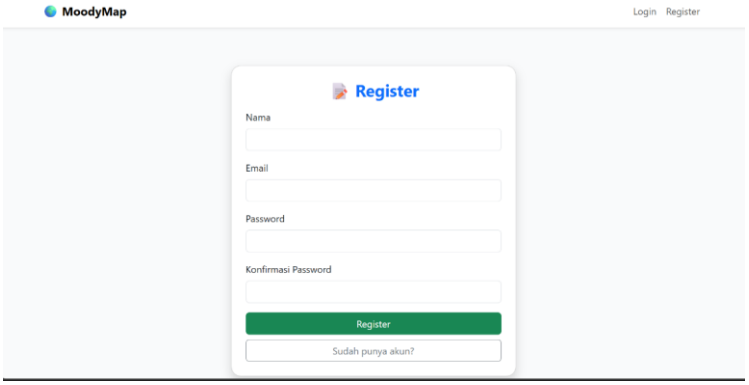


Image 8. Register Page

3.3 Login Page

This page is for authenticating into the system. Users are asked to enter their email and password according to the data they have registered. If the data is correct, the user will be redirected to the home page and can see all their memories. If the user does not have an account, there is a link to the registration page. Additionally, a link is available to reset the password if the user forgets it.

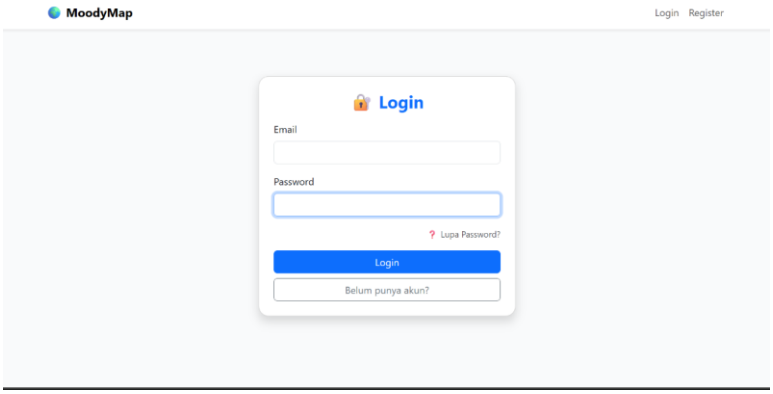


Image 9. Login Page

3.4 Add Memory Page

This page is used to add a new memory based on a location selected from the map. After a user clicks on a location on the map, this page will open with the location coordinates (latitude and longitude) automatically filled in. On this page, users are asked

[DOI: 10.14710/JCASE. vol2.iss1.791]

to fill in a memory title, the date of the event, select an emoji to represent the mood, write a story description, and can upload a photo as documentation. After all the data is filled in and saved, the user will be redirected back to the home page, and the memory pin will be automatically added to the map.

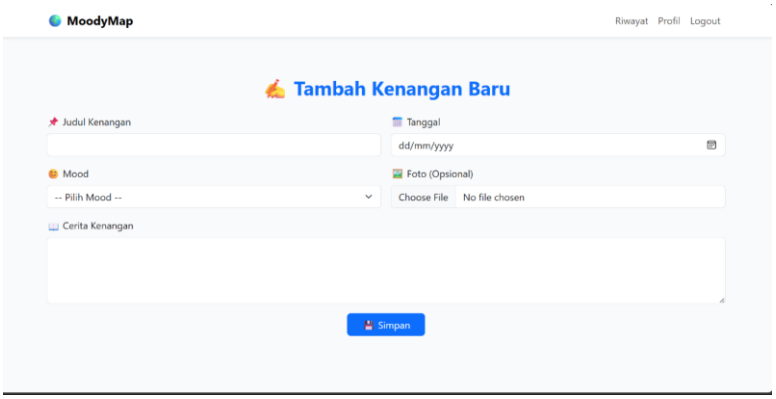


Image 10. Add Memory Page

3.5 Map Pin Page

On the Home page, in addition to displaying the interactive world map and location markers, the system also embeds a popup feature on each pin that has been added by the user. Each pin or location marker represents a single memory that was previously created. When the user hovers over and clicks on a pin on the map, a small popup will automatically appear above it. This popup contains a brief summary of the stored memory, such as the memory title, the chosen mood emoji, and the date the memory was created. The popup also contains a link that says "View Detail," which, when clicked, will direct the user to the related memory detail page. This popup feature helps users access information quickly and contextually without having to navigate to a new page.

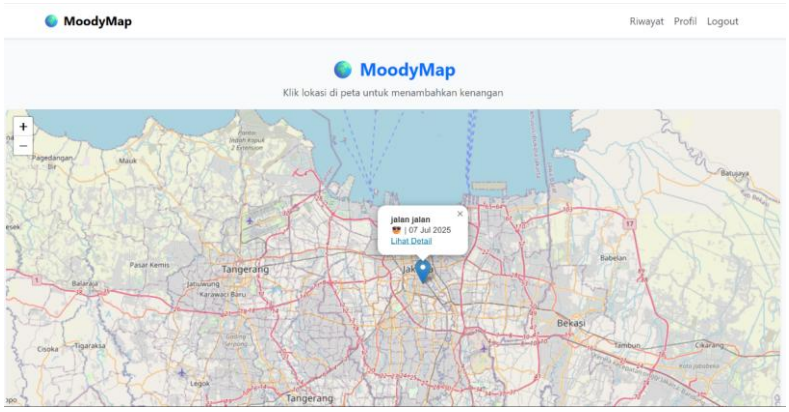


Image 11. Map Pin Page

[DOI: 10.14710/JCASE. vol2.iss1.791]

3.6 Memory Detail Page

The Memory Detail page functions to display the complete information of a memory that has been added. This information includes a memory photo (if uploaded), title, date, mood, and a full description of the memory. At the bottom of the page, there are two buttons: one to edit the memory and one to delete it. If the user chooses to edit a memory, they will be taken to a page similar to the add memory page and can change any data they want, then save the information. If the user chooses to delete, the memory will be removed from the system, and the pin on the map will also be automatically removed.

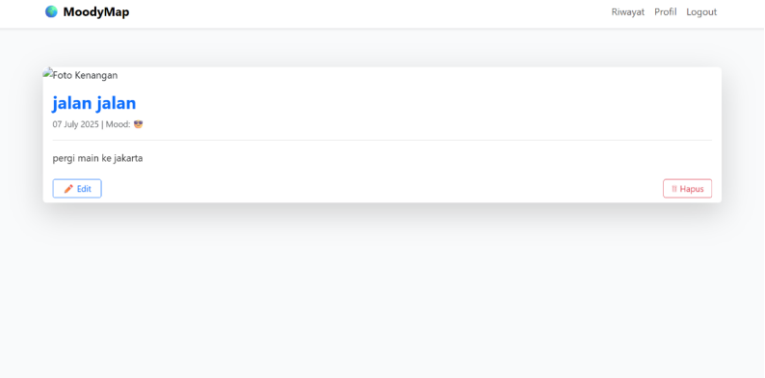


Image 12. Memory Detail Page

3.7 Memory History Page

The Memory History page displays a list of all memories that have been added by the user. The data is displayed in a card format containing a photo, title, mood, date, and a short excerpt from the memory description. This page is also equipped with a search feature based on the title and a filter based on mood. A "View Detail" button is available on each card to access the detail page of the corresponding memory.

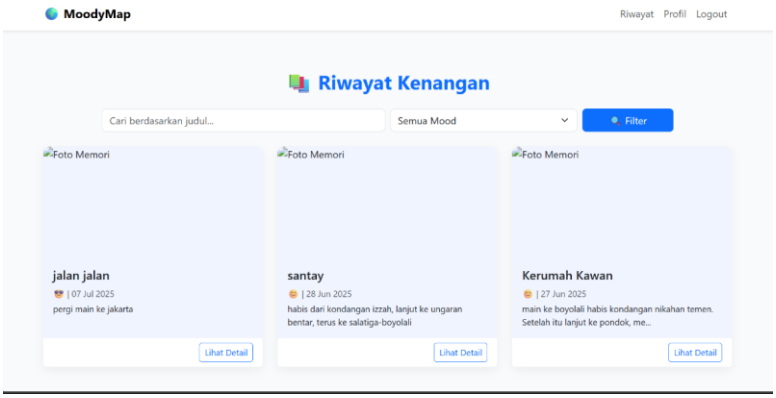


Image 13. Memory History Page

[DOI: 10.14710/JCASE. vol2.iss1.791]

4. *Blackbox Testing*

Black-box testing was conducted to evaluate whether the main functionalities of the MoodyMap application operated according to the predefined requirements. The testing focused on system inputs and outputs without examining the application's internal source code. Test scenarios covered user authentication, password reset, memory management, map interactions, profile management, navigation, validation processes, and other major application functions. The results of the functional testing are presented in Table 2.

Table 2. Black Box Testing Results

No	Tes	Tes Case	Expected Result	Test Result
1.	Register	User fills in name, email, password, and password confirmation	Account created successfully and redirected to the home page	Valid
2.	Register	Email is already registered	An error message appears: "the email has already been taken"	Valid
3.	Register	Password and password confirmation do not match	A validation error appears: "the password field confirmation does not match"	Valid
4.	Register	Password is less than 8 characters	An error message appears: "The password field must be at least 8 characters"	Valid
5.	Register	Name is empty	Validation fails: a message "please fill out this field" appears	Valid
6.	Login	Correct email and password	Successfully logged in and entered the home/map page	Valid
7.	Login	Wrong or unregistered email	Login failed, a message appears "These credentials do not match our record"	Valid
8.	Reset password	Enter a registered email	A password reset email is sent to the user's inbox	Valid
9.	Reset password	Enter an unregistered email	A message appears: "we can't find user with that email address"	Valid
10.	Add Memory	All fields (title, date, mood, location, description, photo) are filled in correctly	Memory is successfully saved, a pin appears on the map	Valid
11.	Add Memory	Title or date is empty	Validation fails: A message appears "Please fill out this field"	Valid
12.	Add Memory	No mood is selected	Validation fails: A message appears "please select an item in the list"	Valid

Commented [A3]: The table is detailed and well-structured. However, the word "Tes" in the second column should be "Test" for consistency.

Add a short paragraph discussing the significance of the successful results or any limitations encountered during testing.

[DOI: 10.14710/JCASE. vol2.iss1.791]

13.	Add Memory	Photo is not uploaded	Memory is still saved without a photo	Valid
14.	Edit Memory	Change description and save	Memory successfully updated with new data	Valid
15.	Edit Memory	Do not change any information in the form	Memory is still successfully saved with the old data	Valid
16.	Delete Memory	Click the delete button and confirm	Memory is deleted from the database and the pin disappears from the map	Valid
17.	Detail Memory	Click the "Detail" button on the pin popup	The memory detail page is displayed with all stored information	Valid
18.	Memory History	Enter the history page without a filter	All memories belonging to the user are displayed	Valid
19.	Memory History	Filter for a specific mood	Displays the selected filter with the highest priority	Valid
20.	Memory History	Fill in the title search form	Memories containing the search keyword in the title are displayed	Valid
21.	Profile	Click the "Profile" page on the home page	Data such as name, email, join date, number of memories, and favorite mood are displayed	Valid
22.	Edit Profile	Change name and email, click save	Profile data is successfully updated and a notification alert appears	Valid
23.	Edit Profile	Leave name empty and click save	Validation fails: A message appears "please fill out this field"	Valid
24.	Logout	Click the logout button	Logout successful, redirected to the initial map page as a guest	Valid
25.	Click Map (Guest)	A user who is not logged in clicks the map	Redirected to the registration page	Valid
26.	Navbar Navigation	Click "Home", "Profile", "History", "Logout" menu	The menu functions according to the route	Valid
27.	Pin Popup	Click a pin on the map	A popup appears with the mood emoji, date, title, and a "Detail" button	Valid
28.	Pin Popup	Click the "Detail" button in the popup	Redirected to the Memory Detail page	Valid
29.	Reset password without input	Click the send link button without filling in the email	Validation fails: A message appears "please fill out this field"	Valid

[DOI: 10.14710/JCASE. vol2.iss1.791]

30.	Photo not displayed	The photo has been uploaded but does not appear	Check the storage/photos path	Valid

Based on the black-box testing results, all 30 test scenarios produced outputs that were consistent with the expected results and were therefore classified as valid. These results indicate that the main functionalities of MoodyMap, including user authentication, memory management, map interaction, profile management, navigation, and input validation, operated correctly under the tested conditions. The successful execution of these scenarios also demonstrates that the integration between the application interface, database, and interactive map functions was able to support the primary requirements of the system.

However, the testing conducted in this study was limited to functional black-box testing and primarily focused on verifying application behavior under predefined scenarios. The study did not specifically evaluate system performance under a large number of simultaneous users, usability through standardized user-experience measurements, or security through penetration testing. Therefore, future evaluation may include performance testing, usability testing involving a broader group of users, and security testing to provide a more comprehensive assessment of MoodyMap.

5. CONCLUSION

MoodyMap provides a web-based platform for recording and revisiting personal memories based on geographical locations. By integrating an interactive map, personal narratives, photographs, and mood representation using emojis, the application enables users to document not only what happened but also where an experience occurred and how they felt at that moment. The implementation of Laravel, Bootstrap, MySQL, and Leaflet.js successfully supported the development of the application's main functionalities.

Black-box testing involving 30 functional test scenarios showed that all tested functions produced results consistent with the expected outcomes. This indicates that the main features of MoodyMap, including authentication, memory management, profile management, map interaction, navigation, and input validation, functioned properly under the tested conditions. Future development can focus on social networking features, memory notifications, location radius-based search, performance evaluation, usability testing, and additional security assessment.

ACKNOWLEDGMENT

The author would like to thank Supervising Lecturer Mr. Basirudin Ansor, M.Kom for his guidance, as well as all parties in the Information Technology Study Program at Muhammadiyah University of Semarang who have supported the smooth completion of this project.

REFERENCES

[1] **Beck, K., et al.** (2001). *Manifesto for Agile Software Development*. Agile Alliance. https://agilemanifesto.org
[2] **Highsmith, J.** (2009). *Agile Project Management: Creating Innovative Products*. Addison-Wesley Professional.
[3] **Pressman, R. S., & Maxim, B. R.** (2014). *Software Engineering: A Practitioner's Approach* (8th ed.). McGraw-Hill Education.
[4] **Kramer, J.** (2018). The Nature of Software Engineering. *Communications of the ACM*, 50(4), 33–38. https://doi.org/10.1145/1232743.1232762
[5] **Sommerville, I.** (2015). *Software Engineering* (10th ed.). Pearson Education Limited.
[6] **Yassine, S. A., & Basha, H. S.** (2021). A Comparative Study of Web Development Frameworks: Laravel vs. Django. *International Journal of Scientific & Technology Research*, 10(1), 101–106.
[7] **Indriyani, R., & Helling, I.** (2018). Analisis UX pada Aplikasi E-commerce Terbesar di Indonesia. *Jurnal Teknologi dan Sistem Komputer*, 6(3), 121–128. https://doi.org/10.14710/jtsiskom.6.3.121-128
[8] **Agung, P. P., Kridalukmana, A., & Windasari, N. A.** (2016). Analisis Kebutuhan Pengguna untuk Sistem Informasi. *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*, 4(1), 45–50.
[9] **Firdausy, R. A., Yudhana, D., & Ahmad, A.** (2008). Perancangan Prosedur Sistem Informasi Akademik. *Jurnal Teknik Informatika*, 2(2), 99–104.
[10] **Handayani, D.** (2018). Desain Antarmuka Sistem Informasi Akademik. *Jurnal Sistem Informasi*, 14(1), 33–40.
[11] **Siregar, R.** (2021). *Penggunaan Leaflet.js untuk Visualisasi Peta Interaktif Berbasis WebGIS*. *Jurnal Teknik Informatika dan Sistem Informasi*, 7(2), 123–132.
[12] **Marwan, D. & Cahyono, H.** (2020). Pemanfaatan Laravel Framework untuk Pengembangan Aplikasi Web. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIIK)*, 4(6), 1234–1241.
[13] **Mujab, U., Satoto, B. D., & Martono, K. T.** (2014). Pemeliharaan Sistem Informasi Akademik Menggunakan Software Maintenance. *Jurnal Komputasi*, 2(2), 43–49.
[14] **Maulidina, A. R., & Yuliana, L. A.** (2022). Penerapan Leaflet.js dan OpenStreetMap untuk Visualisasi Geospasial. *Jurnal Teknologi dan Sistem Komputer*, 10(1), 77–84.
[15] **Puspitasari, D., & Lestari, R.** (2021). Pengaruh User Interface terhadap Kepuasan Pengguna pada Aplikasi Web. *Jurnal Rekayasa Sistem dan Teknologi Informasi*, 4(1), 51–58.