

Article

Not peer-reviewed version

Eureka: Question Answer Course Discussion using Agile Methods

[Wisnu Uriawan](#)*, [Danny Suggi Saputra](#), Andika Nuralamsyah, Arief Rahman Mubarak, Dailfhiana Fujianti, Denis Firmansyah, Diman Fathurohman

Posted Date: 28 June 2024

doi: 10.20944/preprints202406.1951.v1

Keywords: agile methodology; collaborative learning; educational technology; iterative development; online Q&A platform



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Eureka: Question Answer Course Discussion using Agile Methods

Wisnu Uriawan *, Danny Suggi Saputra, Andika Nuralamsyah, Arief Rahman Mubarak, Dailfhiana Fujianti, Denis Firmansyah and Diman Fathurohman

Informatics Department, UIN Sunan Gunung Djati Bandung, Jawa Barat, Indonesia; danny.saputra3003@gmail.com (D.S.S.); andika.nuralamsyah@gmail.com (A.N.); ariefmubarak25@gmail.com (A.R.M.); dailfhianafujianti@gmail.com (D.F.); denisfirmansyah55@gmail.com (D.F.); dimanfn24@gmail.com (D.F.)

* Correspondence: wisnu_u@uinsgd.ac.id

Abstract: "Eureka" is a transformative platform designed to address the academic challenges faced by students by fostering an inclusive and collaborative learning environment. Through an online Q&A discussion format, it facilitates knowledge sharing, coursework discussions, and peer support, thereby enhancing collaborative learning. Key features such as Q&A, Answer Validation, and Leaderboard ensure efficient and reliable information exchange, motivating active student participation and mutual support. The platform's development using Agile methodologies has proven effective, emphasizing iterative development, constant user feedback, and collaborative efforts. This approach has significantly boosted student engagement and improved learning outcomes. "Eureka" not only aids in students' understanding of course material but also contributes to overall academic performance, serving as a powerful tool in creating a supportive academic environment. Its Agile-based development ensures ongoing responsiveness to student needs, enabling continuous improvement and a sustained positive impact on learning experiences.

Keywords: agile methodology; collaborative learning; educational technology; iterative development; online Q&A platform

1. Introduction

Students often face significant challenges when trying to find answers to their questions. Although online resources are available in abundance, searching for specific information often results in frustration due to incomplete or irrelevant results. Traditional classroom discussions often fail to accommodate diverse learning speeds and styles, leaving many students without the support they need. To overcome this problem, "Eureka" provides an online question and answer discussion platform designed to facilitate interaction and collaboration between students in a faculty. The platform aims to help students share knowledge, discuss coursework, and provide and receive academic assistance from fellow students, thereby creating a more inclusive and supportive learning environment.

By utilizing the "Eureka" platform, students can easily find discussion partners who have a better understanding of a particular topic, seek advice on how to approach assignments, and share learning resources. The platform's features allow students to conduct discussion sessions and study together online. This fosters a sense of community and collaboration, helping students overcome academic challenges more effectively.

This platform is equipped with several important features, namely Q&A, Answer Validation, and Leaderboard. The Q&A feature allows students to ask questions related to assignments or course material, which can then be answered by fellow students, facilitating the exchange of information quickly and efficiently. To ensure that the answers provided are accurate and reliable, this platform provides an Answer Validation mechanism, where the answers provided will be reviewed by students who have more authority or experience in the subject, or by assigned teaching assistants.

In addition, the Leaderboard feature rewards students who actively participate and provide valid answers, using a points system to assess their contributions. Students with the highest points will be placed at the top of the leaderboard, which not only motivates students to be more active in helping each other, but also creates a competitive and collaborative learning environment.

Agile methodologies, known for their adaptability and focus on user feedback [1], offer a promising approach to developing such a dynamic and user-centric platform. By employing iterative development processes, constant user feedback, and collaborative efforts among the development team and end-users, "Eureka" can continually evolve to better meet the needs of its users. This paper examines the application of Agile methods in the creation of "Eureka" and evaluates its impact on student engagement and learning outcomes. Initial findings suggest that by promoting cooperation and constructive interaction, "Eureka" enhances students understanding of course material and contributes to better academic performance.

2. Related Work

The research article titled "Towards Understanding Online Question & Answer Interactions and their Effects on Student Performance in Large-Scale STEM Classes" aims to quantify the impact of online Q&A interactions on student performance in STEM education. The study involved 218 computer science students from a large university in the southeastern United States, focusing on their participation in online Q&A activities, including asking, answering, and viewing questions. The findings revealed that viewing questions and answers had the most significant positive effect on student performance, while direct interactions with instructors had minimal impact. The study concludes that online Q&A interactions, particularly passive engagement like viewing, can enhance student learning outcomes in large-scale STEM classes. This research fills a gap in the literature by specifically examining the effects of online Q&A interactions in STEM education, providing valuable insights for the effective implementation of these tools in large classroom settings [2].

The research article titled "Agile Manifesto for Teaching and Learning" by Timothy C. Krehbiel et al., published in *The Journal of Effective Teaching*, explores the application of Agile methodologies, originally developed for software development, in higher education. The study involved faculty members from a public university who adapted Agile principles such as Scrum, daily stand-ups, and retrospectives in their classrooms. The results showed that these methods increased student engagement, responsibility, and collaboration, leading to higher quality outcomes. The authors proposed an Agile Manifesto for Teaching and Learning, emphasizing values like adaptability, collaboration, and continuous improvement over traditional teaching methods. The study concluded that Agile methodologies can effectively enhance teaching and learning experiences across various disciplines [3].

Agile methods have a significant impact on software development. This software development is highly needed, particularly in business performance. A study titled "Agile software development projects—Unveiling the human-related critical success factors" states that the research was conducted to understand the success of software development projects using agile methods, influenced by human-related critical factors. Four explanatory factors are included in the study: team autonomy, psychological safety, customer involvement, and team capacity. The success of agile software development projects is another dependent variable. A survey using a questionnaire was used in this investigation, and 177 valid replies were received. After that, a partial least squares analysis was done to evaluate the theoretical framework. The findings show that the two factors that have the biggest impacts on the success of agile software development projects are team capacity and client involvement. In addition, psychological safety is a key component of indirect success. However, there is a competitive dynamic between team autonomy and psychological safety regarding the other two elements [4].

Agile methods are one of the approaches that empirically influence the software development process. This method can bring about changes to improve quality, economy, and simplicity. The study titled "Agile software development and UX design: A case study of integration by mutual adjustment" explains how a Danish software company develops software using Agile methods and integrates UX design. In this paper, a case study is presented that includes two phases: (a) a single round of personal interviews with ten staff members (four user experience designers, three software engineers, two project managers, and one solution architect); and (b) a second round of workshops with six staff members (two project managers, three user experience designers, and two solution

architects) conducted two years later. The research examines the organization's integration strategy, which involves "front-loaded design" and "working in parallel" and involves collaboration as opposed to assimilation or the division of software development and UX design [5].

According to the study, UX designers and software engineers have distinct contributions to make in terms of attaining agility when it comes to integration through mutual adjustment. They employ completely different labor procedures, yet they are nevertheless able to adapt to change and improve simplicity, economy, or quality. Nonetheless, these procedures continue to be susceptible to integration difficulties over time, as demonstrated by the follow-up workshops held two years following the first interviews. In light of this, it can be said that integration based on collaboration may distinguish but also enhance the agility of software engineers and UX designers. In contrast to assimilation, which could result in their agility being identical, and separation, which results in their agility becoming distinct and competitive, this integration [5].

3. Methodology

In developing this software we used the Agile method, which is an approach to software development that is iterative and incremental, where development is carried out collaboratively between the development team and stakeholders. One of the special points in agile is that the process is divided into small, iterative processes. The 2013 IT Project Success Rate Survey results prove the superiority of Agile methods in the context of effectiveness and success compared to traditional methods [6]. Agile has a more flexible iterative flow unlike waterfall, so we can make adjustments, cut, or change something that has been determined without having to wait for the project to be completed first [7].

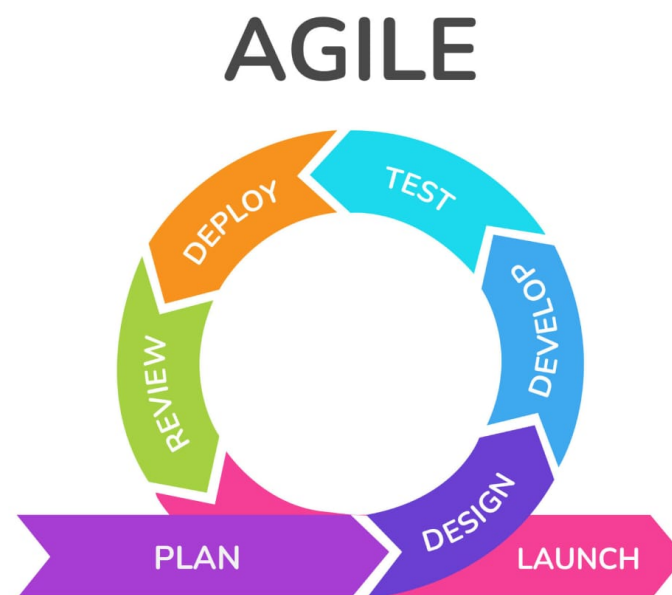


Figure 1. Agile Method

Unlike the traditional product-focused process of defining the experience, which centers around the user, this methodology places a strong emphasis on innovation, which creates greater value in products and services. Additionally, because of its application, it generates solutions that might not have been implemented otherwise, allowing developers to "put themselves in the shoes" of their users [8].

Research Stages Using the Agile Method which has the following general stages:

3.1. Initial Phase

Initial Phase or Initialization Phase is a phase that aims to establish the vision and initial scope of the project, identify stakeholders, and collect requirements that will be included in the product backlog. Software development planning is carried out in this phase where all the resources needed in the future development steps are collected. The processes included in this phase are the planning and analysis processes.

1. Planning

In the planning process we determine the vision, goals, and scope of our software [9]. Creating a list of software backlogs that we will develop and identifying stakeholders on this project. And we found that students are one of the biggest references in the development of this software because they are the main target of the problems we raise.

2. Analysis

In the analysis stage we try to get opinions from students using user personas, in this way we get the important things needed in the development of our software. In addition, we also update the backlog that was previously designed to be redesigned according to the user persona data we received for the effectiveness of the software.

3.2. Development Phase

The Development Phase is the core of Agile development, where software is developed through several development processes before finally being launched [10]. In this phase there are several important processes that must be carried out, namely the design process, the develop process, and the test process. Development in this phase must of course be in accordance and in line with what has been determined in the initialization phase.

1. Design

The design stage uses Figma software as an assistance tool, here we design the UI / UX design of each interrelated software page. In developing this software we use several programming languages such as PHP and JavaScript and use various frameworks such as React and Laravel.

2. Develop

In the next stage we started developing the software. Realize the software that has been designed and implement the backlog into program code.

3. Test

Furthermore, at the test stage, two types of testing are carried out, namely Black Box Testing and User Acceptance Test [11]. Black Box Testing is used to test the functionality of the software, while User Acceptance Test is used to test the suitability of the development results with user needs.

3.3. Closure Phase

Closure Phase is a phase in the agile method that enters the final stage. In this phase, everything that has been developed will be deployed and then a review related to the software that has been made before later entering the launch stage. There are three processes that go into this phase, namely the deploy process, the review process, and the launching process.

4. Result and Discussion

4.1. Planning

In the planning process, we established the foundational elements required for our software development project. We decided to use the PHP programming language with the Laravel framework for the server-side and JavaScript along with the Tailwind CSS framework for the client-side. This technical stack was chosen based on its robustness and compatibility with the project requirements.

Additionally, we designed the initial backlog and identified key stakeholders. The backlog included a prioritized list of features and tasks to be developed, ensuring a clear roadmap for the project. Stakeholders, primarily students, were identified as the main users of the software, given that they are the primary focus of the problems we aim to address.

4.2. Analysis

In the analysis stage, we focused on gathering detailed insights to inform the redesign of our project. This involved creating user personas, use case diagrams, and an entity-relationship diagram (ERD). These tools were essential for understanding user needs, defining system requirements, and ensuring that the design and functionality of our software would be aligned with user expectations.

Table 1. User Persona.

User Persona 1	
Name	Bintang Tsany
Major	Electrical Engineering
Semester	6
Needs	Find student discussion applications to answer assignments and problems related to their lectures.
Solution	Create a credible discussion platform
User Persona 2	
Name	Serevina Sherly Maulida
Major	Informatics Engineering
Semester	4
Needs	Requires a question and answer platform that can provide answers as quickly as possible and with varied answers.
Solution	Provides a question and answer platform that can solve student problems quickly and be answered by students between departments

1. User Persona

User personas are fictional representations of our primary user groups [12]. They were created based on data collected from students, who are the main target users of our software. These personas included detailed profiles that described the users’ demographics, goals, behaviors, and pain points. By developing user personas, we were able to empathize with our users and ensure that our design decisions addressed their specific needs and preferences.

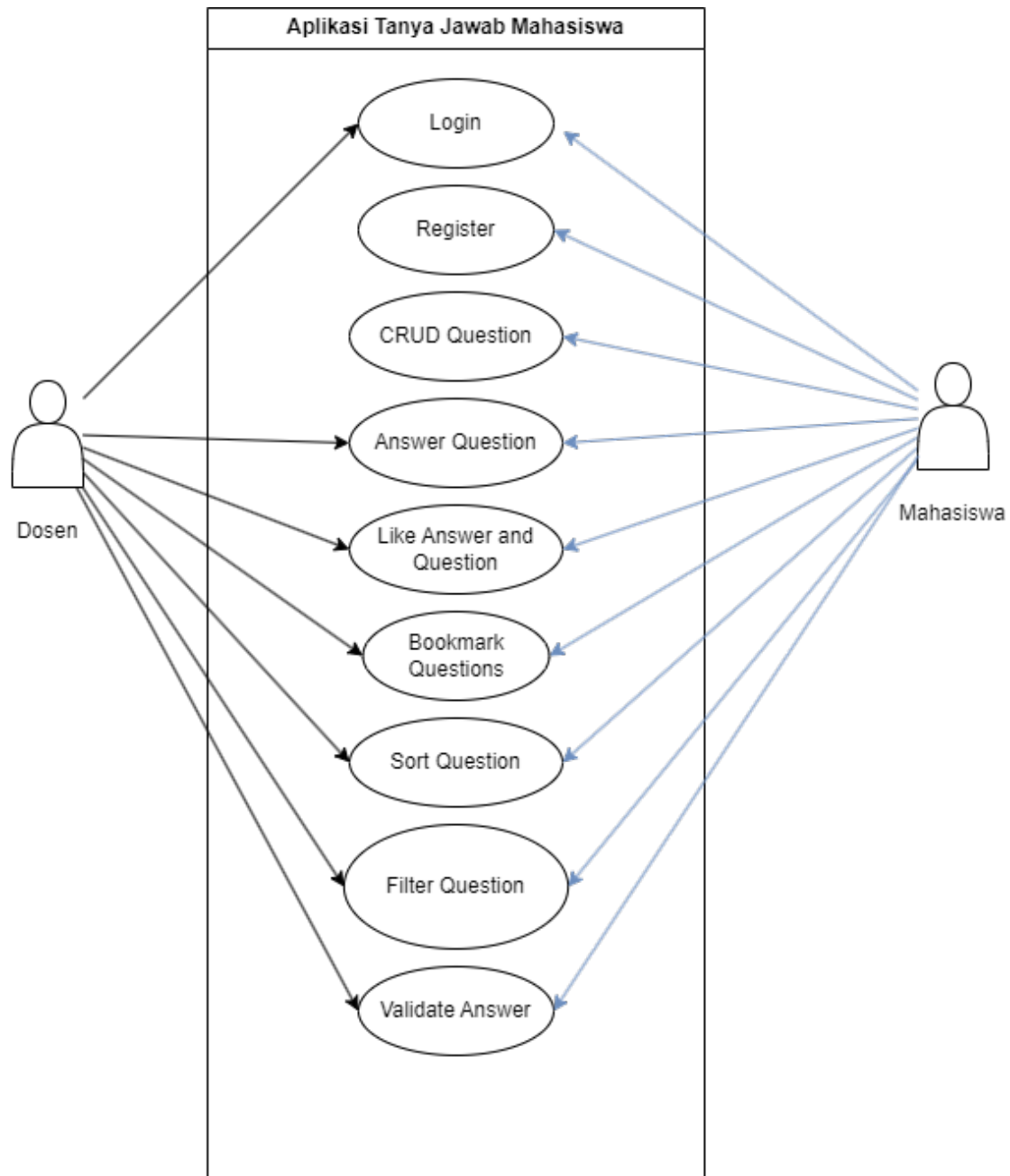


Figure 2. Use Case Diagram.

2. Use Case Diagram

The use case diagram was developed to visualize the interactions between users and the system [13]. This diagram helped in identifying and defining the different functionalities that the software needed to support. Each use case represented a specific user interaction with the system, ensuring that all possible actions a user might take were considered in the design.

3. Entity-Relationship Diagram (ERD)

The ERD was created to model the data structure of the system [14]. It provided a detailed view of the entities, their attributes, and the relationships between them. This diagram was crucial for designing the database schema, ensuring that all necessary data would be accurately captured and efficiently stored.

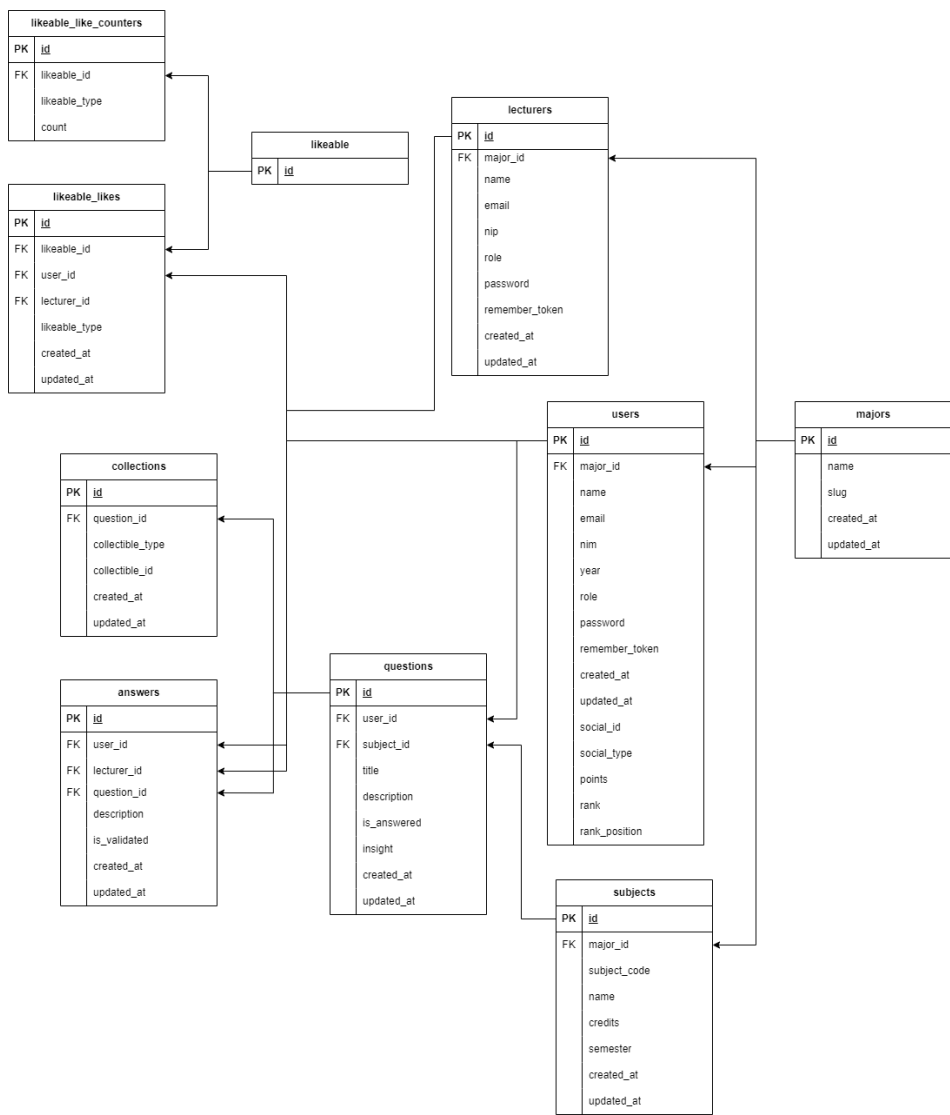


Figure 3. Entity Relationship Diagram.

The insights gained from the user personas, use case diagram, and ERD were instrumental in redesigning the project. The feedback from user personas highlighted specific user needs and preferences, which were then incorporated into the updated backlog. The use case diagram ensured that all necessary functionalities were included, and the ERD provided a robust framework for the database design.

This comprehensive analysis and subsequent redesign played a critical role in increasing the effectiveness of our product to meet user needs. The iterative process of the backlog and ensuring that the final product was gathering data, analyzing it, and refining the design embodies the Agile methodology’s focus on continuous improvement and user-centric development.

4.3. Design

Before we move on to a larger stage, we previously created a low-fidelity frame containing a wireframe to serve as a framework for creating the original design. Here we have created some wireframes that can help in creating the initial design.



Figure 4. Low Fidelity Design.

After obtaining several designs, we have created a design that can be implemented into a coding language. The design we create covers all the functions that we will develop. The wireframe that we have created is then implemented into a complete design by adding appropriate colors and text. Sometimes there are several wireframes that are used to create more than one complete design, such as a sidebar in one wireframe. One of the entire wireframe parts is used to develop other designs.



Figure 5. High Fidelity Design.

4.4. Develop

The design stage is carried out using various tools such as Figma, the design is adjusted to the design which refers to the analysis stage. To maximize design results, we use low fidelity to evaluate every concept and design created. For the programming process, we also create components and layouts that will be made so that they are easy to implement in future programs.

After some planning, we have created a design for the "Eureka" website. There are several displays that we have designed and can be seen as follows.

1. Landing Page

On the landing page we have provided an initial display for users. Where on the navigation bar there are questions, register and login buttons. Continued at the bottom there is a search column for relevant questions. Then scroll further down there is also a list of majors which can be used as a question format so you can filter questions based on majors. Scroll down again there is a description of the website name "Eureka". Then in the footer there is a slogan that we use to motivate users.



Figure 6. Landing Page.

2. Login and Register Page

First, starting from the login page, here we provide two columns to enter the email address and user account password. Below that there is a login button to verify the email address and password that was entered previously. And next to it there is a forget password button if the user forgets the password for their account. Apart from that, we also provide a login button with Google as a function to log in using a Google account so you don't have to retype your email address and password.

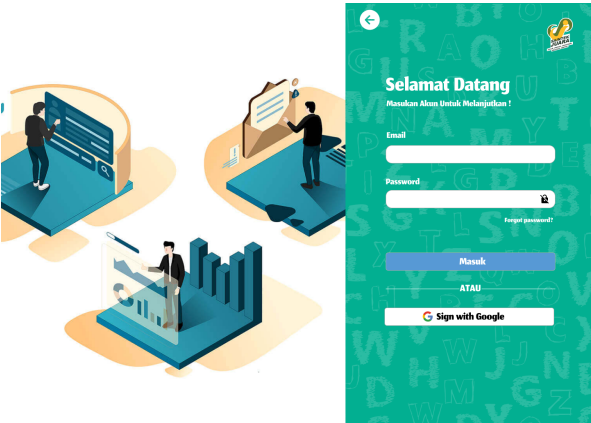


Figure 7. Login Page.

For the registration form, we provide some personal data that users must fill in to register their account. Apart from creating an account using email, users can also register using a Google account.



Figure 8. Register Page.

3. Question Page

After logging in using a registered account, the user will be directed to the home page or inquiry page. Where there are several questions displayed randomly. Users can also sort questions by newest, recommended, and frequently viewed options.

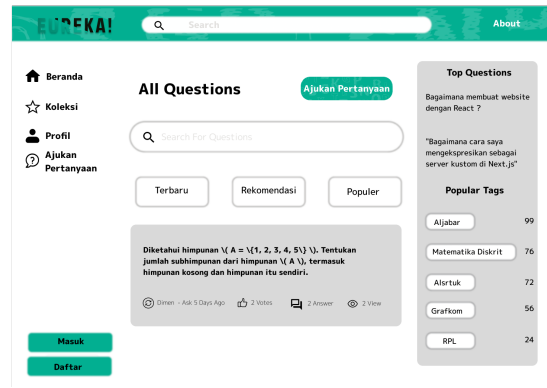


Figure 9. Question Page.

4. Detail Question Page

When the user presses the question box, the user will be directed to a page that displays details regarding the question. Users can comment or answer questions. Users can also like answers given by other users, and comment on the answers.

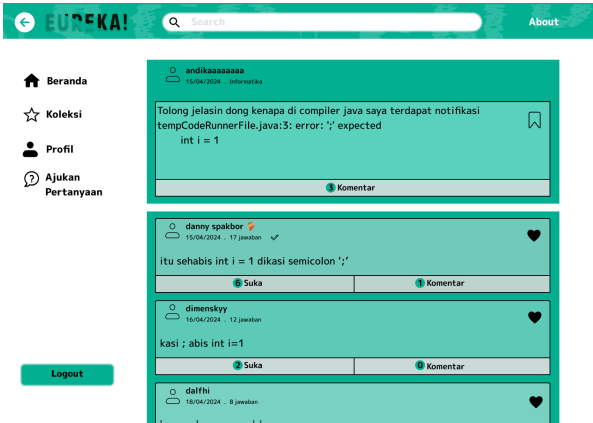


Figure 10. Detail Question Page.

5. Ask Question Page

This leaderboard page can be accessed from the profile page by pressing the sequence number button. This page provides information about the achievements achieved by users by getting badges in the form of answers verified by lecturers regarding their department. Users can use this as a challenge to get the top position. That way, the user who gets the most answer verifications will have their name displayed first.

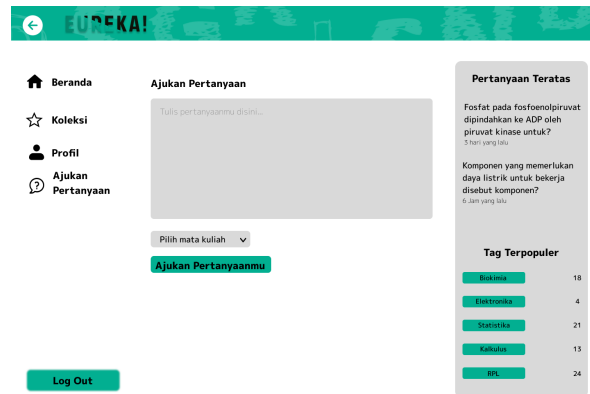


Figure 11. Ask Question Page.

6. Collection Page

In the collection page which can be accessed in the left side bar, there are several questions that have been bookmarked by users. There, users can review their favorite questions if they are still interested in discussing these questions.

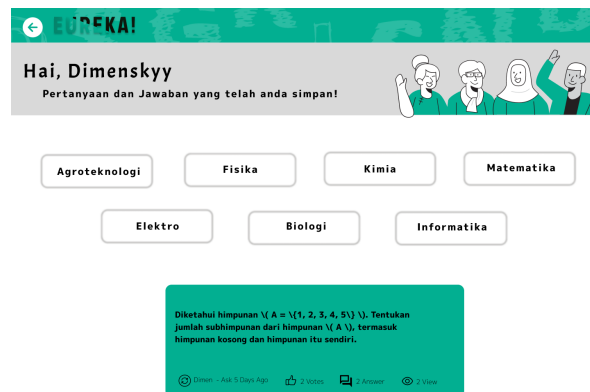


Figure 12. Collection Page.

7. Profile Page

On the profile page, users can see all their personal data that has been entered during the registration process.

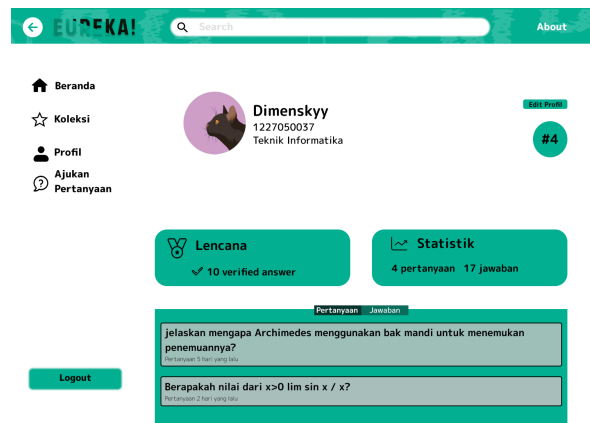


Figure 13. Profile Page.

Apart from that, by pressing the edit profile button, users can also edit their personal data according to what they want to change

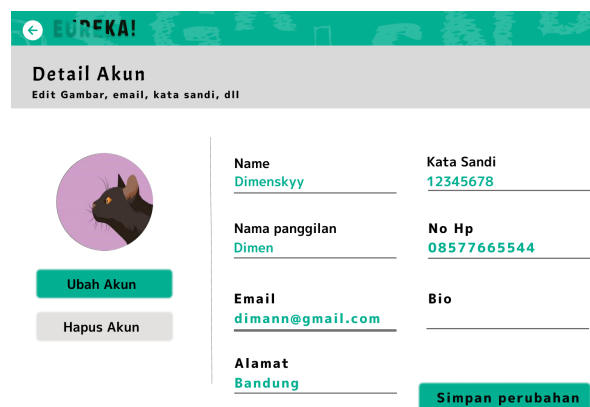


Figure 14. Edit Profile Page.

8. Leaderboard Page

This leaderboard page provides information about the achievements achieved by users by getting badges in the form of answers verified by lecturers regarding their department. Users can use this as a challenge to get the top position. That way, the user who gets the most answer verifications will have their name displayed first.



Figure 15. Leaderboard Page.

9. Reset Password Page

Users will be directed to this page if they request a forgotten password. Users will be directed to fill in the email connected to their Eureka account.

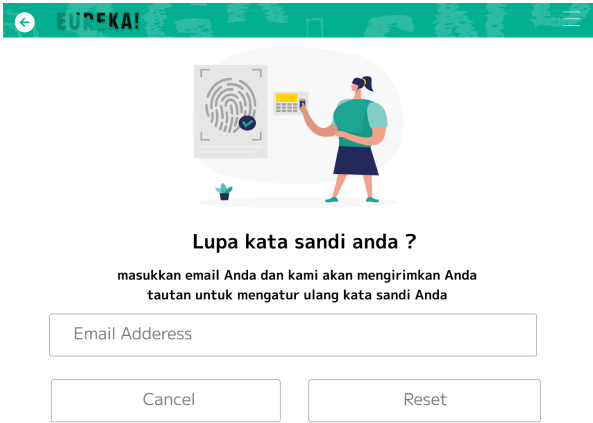


Figure 16. Forgot Password Page.

Next, a verification link will be sent to the email if the email entered matches the one on the Eureka website.

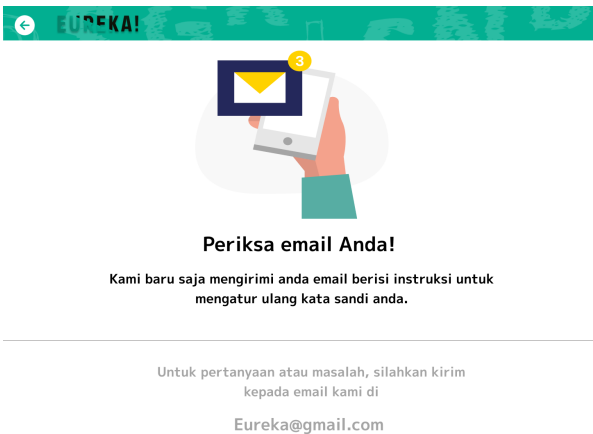


Figure 17. Email Confirmation Page

After that, the user will be asked to change the password in two steps to change the password.



Figure 18. Reset Password Page.

4.5. Testing

The software testing conducted in the table above uses the functional testing method to ensure that the various features in the application function according to the expected specifications. This test covers various use cases ranging from the login process, registration, CRUD (Create, Read, Update, Delete) for questions, to interaction with answers and additional features such as liking, bookmarking, sorting, filtering, answer validation, and profile editing.

The testing method used is black-box testing, where the tester does not need to know the internal details or source code of the application being tested [15]. The main focus of this test is to check whether each feature in the application provides the expected results when given certain inputs.

In the login test, the user is tested to be able to log in with valid credentials, and the results show that the user is successfully logged in and redirected to the main page. In addition, the test also ensured that the user could not login with invalid credentials, with the result being an error message displayed, indicating a failed login. Both scenarios were successful, indicating that the authentication mechanism works well.

Registration testing is also conducted to ensure that users can register with valid data, the result of which is that the user successfully registers and is redirected to the login page. Conversely, for invalid data, the system displays an error message and registration fails, ensuring that data validation in the registration process works correctly.

The CRUD functionality on questions was tested by ensuring that users could create, read, update and delete questions. Each of these scenarios succeeded with the expected results: new questions appeared in the list, question details could be read, changes to questions were saved and displayed, and questions could be deleted from the list.

Testing interaction with answers included the ability for users to answer questions, like questions and answers, and bookmark questions. All of these functions were tested and the results were successful, indicating that the application allows the expected interactions from the user.

Additional features such as question sorting and filtering were also tested to ensure users could sort and filter questions according to their needs. The results of this test show that both features work well.

Answer validation by the inquiring user was also tested, with the result that answers could be validated, and profile editing testing ensured that changes to the user's profile were saved and displayed correctly.

Overall, the test results showed that all tested features functioned according to the expected specifications, indicating that the application has been well developed and meets the identified user needs.

4.6. Discussion

We have created Eureka to help students experience their assignments so they can discuss them with others. Apart from our success in creating Eureka, we have also fulfilled all aspects that have been proposed by several participants who filled in the form we provided.

Apart from what Eureka offers, namely ease in carrying out assignments, Eureka also actually gives us the experience of helping other people in carrying out their assignments. That is another important thing why Eureka was formed, namely to further develop our intelligence which answers the question.

But that's not all, users who see questions and the answers to which are available can also gain knowledge about the problems and solutions being asked. This can be used as a lesson for us to find out knowledge that we might encounter someday.

5. Conclusion

The "Eureka" platform represents a significant step forward in addressing the challenges students face when seeking academic assistance and fostering a more inclusive learning environment. By leveraging an online Q&A discussion format, it empowers students to share knowledge, discuss coursework, and provide and receive support from peers, thereby enhancing collaborative learning. Key features such as Q&A, Answer Validation, and Leaderboard ensure that information exchange is both efficient and reliable, motivating students to actively participate and support each other's academic endeavors.

The implementation of Agile methodologies in developing "Eureka" has proven to be particularly effective. By focusing on iterative development, constant user feedback, and collaborative efforts, the platform continually evolves to better meet the needs of its users. Initial findings indicate that this approach significantly boosts student engagement and improves learning outcomes. By promoting cooperation and constructive interaction, "Eureka" not only aids in students' understanding of course material but also contributes to their overall academic performance.

In conclusion, "Eureka" serves as a powerful tool in creating a supportive and collaborative academic environment, helping students to overcome challenges more effectively and achieve better academic results. Its development through Agile methods ensures that it remains responsive to student needs, paving the way for continuous improvement and sustained impact on student learning experiences.

Acknowledgments: With our success in creating this website, we are grateful for the support provided by our campus, the Informatics Department UIN Sunan Gunung Djati Bandung. Without their support we might not have been able to make this project.

References

1. Caballero, L.; Moreno, A.M.; Seffah, A. How agile developers integrate user-centered design into their processes: a literature review. *International Journal of Software Engineering and Knowledge Engineering* **2016**, *26*, 1175–1201.
2. Smith IV, D.H.; Hao, Q.; Dennen, V.; Tsikerdekis, M.; Barnes, B.; Martin, L.; Tresham, N. Towards Understanding Online Question & Answer Interactions and their effects on student performance in large-scale STEM classes. *International Journal of Educational Technology in Higher Education* **2020**, *17*, 1–15.
3. Krehbiel, T.C.; Salzarulo, P.A.; Cosmah, M.L.; Forren, J.; Gannod, G.; Havelka, D.; Hulshult, A.R.; Merhout, J. Agile Manifesto for Teaching and Learning. *Journal of Effective Teaching* **2017**, *17*, 90–111.
4. Barros, L.; Tam, C.; Varajão, J. Agile software development projects—Unveiling the human-related critical success factors. *Information and Software Technology* **2024**, *170*, 107432. doi:<https://doi.org/10.1016/j.infsof.2024.107432>.

5. Persson, J.S.; Bruun, A.; Lárusdóttir, M.K.; Nielsen, P.A. Agile software development and UX design: A case study of integration by mutual adjustment. *Information and Software Technology* **2022**, *152*, 107059. doi:https://doi.org/10.1016/j.infsof.2022.107059.
6. Alqudah, M.; Razali, R. A review of scaling agile methods in large software development. *International Journal on Advanced Science, Engineering and Information Technology* **2016**, *6*, 828–837.
7. VanderLeest, S.H.; Buter, A. Escape the waterfall: Agile for aerospace. 2009 IEEE/AIAA 28th Digital Avionics Systems Conference. IEEE, 2009, pp. 6–D.
8. Romero, C.A.T.; Ortiz, J.H.; Khalaf, O.I.; Ortega, W.M. Software architecture for planning educational scenarios by applying an agile methodology. *International Journal of Emerging Technologies in Learning (Online)* **2021**, *16*, 132.
9. Blomkvist, S. Towards a model for bridging agile development and user-centered design. In *Human-centered software engineering—integrating usability in the software development lifecycle*; Springer, 2005; pp. 219–244.
10. Seikola, M.; others. The Scrum Product Backlog as a Tool for Steering the Product Development in a Large-Scale Organization. Master's thesis, Aalto University, 2010.
11. Kusuma, A.B.; Hadinata, N. The Implementation of the Black Box Method for Testing Smart Hajj Application Ministry of Religion. *Journal of Information Systems and Informatics* **2022**, *4*, 673–686.
12. Nielsen, L. *Personas-user focused design*; Vol. 15, Springer, 2013.
13. Beimel, D.; Kedmi-Shahar, E. Improving the identification of functional system requirements when novice analysts create use case diagrams: the benefits of applying conceptual mental models. *Requirements Engineering* **2019**, *24*, 483–502.
14. Gocheva, D.; Batchkova, I.; Popov, G. Ontology-Based Data Access and Model Transformations for Enterprise Interoperability. *Machines. Technologies. Materials.* **2016**, *10*, 16–19.
15. Henard, C.; Papadakis, M.; Harman, M.; Jia, Y.; Le Traon, Y. Comparing white-box and black-box test prioritization. Proceedings of the 38th International Conference on Software Engineering, 2016, pp. 523–534.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.