DESIGN OF CLOUD-BASED CHATBOT APPLICATION AT PT. TRAVELOKA SINGAPORE USING THE AGILE METHOD

Yuda Adi Pratama¹, Titin Kristiana²

Information System
Nusa Mandiri University
11211134@nusamandiri.ac.id, titin.tka@nusamandiri.ac.id
yudaa0110@gmail.com, titin.tka@nusamandiri.ac.id
(¹) Corresponding Author

Abstract— The role of customer service will be good if it can serve all obstacles or difficulties users face directly and in real time. However, there are times when the number of customer service is not proportional to the number of users who make complaints to customer service, and often, users ask repetitive questions generally available on Frequent Ask Questions (FAQ), so there are limitations and are fewer responsive in serving user complaints. By developing a cloud computing-based Chatbot application, it is hoped that it will make it easier for customer service to handle recurring questions and increase response time to users in real-time. The development of this chatbot application uses the agile method with the scrum framework. Where in the development process carried out is divided into several phases called sprints. The development of this application was carried out in 3 sprints from the time the project was announced to completion.

Keywords: Chatbot, Cloud Computing, Agile.

INTRODUCTION

Traveloka Singapore is a leading travel and lifestyle platform that offers various services to help users plan their trips and book flights, hotels, activities, and other travel-related services. With the increasing number of users and features on the Traveloka Singapore application, providing quality customer service has become crucial to enhancing user experience and satisfaction.

Effective customer service plays a significant role in building solid customer relationships and increasing loyalty toward Traveloka Singapore. Users expect fast and appropriate solutions to their queries, complaints, or problems. However, sometimes, the number of customer service personnel may not be sufficient to cater to the growing user base, resulting in delayed responses and unsatisfied customers (Çalı and Çalı 2022; Taylor et al. 2020). In addition, users may also ask repetitive questions already available in the Frequently Asked Questions (FAQ) section, leading to inefficiency in resolving customer queries.

Building a cloud computing-based chatbot application aims to make it easier for customer services to handle recurring user questions and complaints in real-time (Li and Zhang 2023). One of the objectives is to facilitate users submitting complaints to customer services by providing a chatbot that can respond quickly and efficiently to their concerns. Additionally, the chatbot will be able to handle repetitive questions and complaints, improving response time to users. As a result, users will not have to wait for long periods before their issues are resolved. Finally, the chatbot will help to reduce the operational burden of customer services. This is because the chatbot can handle many tasks typically performed by human customer service representatives, such as responding to frequently asked questions and resolving common issues.

Studies by (Ananda Dwi et al. 2018) and (Nugroho, Adi, and Gumelar 2020) have highlighted the need for innovative solutions to address these challenges. One such solution is integrating chatbot technology, powered by cloud computing, to facilitate customer service operations. Chatbots are
intelligent conversational agents that can automate customer service processes by answering frequently asked questions, providing real-time assistance, and even resolving simple issues without human intervention (Enterprise 2018).

The previous research from S. Sudaryono, N. Lestari, and K. Gunawan (Sudaryono, Lestari, and Gunawan 2020) shows that building a Virtual Assistant significantly impacts human work and provides quick and easy access to information. The Agile concept with the Scrum method used in the development of Virtual Assistants makes it fast and efficient, which enables the application to provide excellent service to all its users. Meanwhile (Chandra and Kosdiana 2019) indicates that building the Chatbot Line application can improve the efficiency and speed of responding to user communications and providing information.

Based on the journal of (Ananda Dwi et al. 2018) concludes that the chatbot application built can help SME Minsu's customer service by answering customer questions and placing orders for products. The chatbot can serve the role of human customer service efficiently and effectively. Another journal from (Astuti and Fatchan 2019) shows that building a Chatbot application called RiChat simplifies customer service operations in answering user questions, making it more efficient, effective, and faster since the chatbot can work 24 hours a day.

Although no GAP analysis was conducted in this study, the author attempts to explore the topic by conducting a comprehensive literature review and in-depth analysis of the problem to be solved. The author found several studies related to the same topic in the literature review but with different approaches and methodologies. Therefore, the author strives to adopt the appropriate approach and methodology to address the problem.

Using cloud computing technology, Traveloka Singapore can optimize its customer service operations by deploying a chatbot application that can address frequently asked questions and resolve simple issues in real time. This approach can reduce the workload of customer service personnel and enable them to focus on complex queries and complaints that require human intervention (Larasati Amalia and Wahyu Wibowo 2019). The chatbot application can also provide 24/7 customer support, ensuring users receive prompt assistance whenever required.

To develop and deploy a cloud-based chatbot application, Traveloka Singapore can leverage the benefits of cloud computing technology (Wijaya and Wjiaya 2018). Cloud computing provides scalable and flexible infrastructure, enabling the chatbot application to run optimally and handle varying levels of user traffic. Cloud computing also ensures high availability and reliability of the chatbot application, minimizing downtime and enhancing user experience.

In conclusion, integrating cloud-based chatbot technology in customer service operations can enhance the quality of customer service Travelok Singapore provides. This innovative solution can reduce response time, automate processes, and improve user satisfaction, contributing to the platform’s overall success.

MATERIALS AND METHODS

1. Data Collection Technique
   a. Observation
      To get datasets related to this chatbot, developers look for datasets on many dataset provider websites such as Kaggle. The developer also uses frequently asked questions on the Traveloka Singapore website as an additional dataset for the chatbot application.
   b. Interview
      The developer conducted an online interview with Mr. Juan Kanggrawan as PIC of Traveloka Singapore via WhatsApp message by asking several things related to the design of the program to be developed.
   c. Literature Review
      The developer conducted a literature study by looking for books and journals related to the design of this chatbot application.

2. System Development Model
   The developer designs this chatbot application using the agile method with the Scrum framework. According to Schwaber & Sutherland (Hadji, Taufik, and Mulyono 2019), Scrum is a framework that can address complex and ever-changing problems and provide suitable product quality according to user needs creatively and productively. The scrum method has several steps that need to be done, such as
   a. Product Backlog
      The product backlog in developing this chatbot application is the development of machine learning models, deploying machine learning models using APIs, and displaying them in the Android mobile application.
   b. Sprint Backlog
      The sprint backlog in the development of this application will later divide into several more detailed phases of the product backlog that have been defined in the product backlog process.
   c. Sprint Planning
      Sprint planning in developing this application will perform system design and analysis using UML diagrams.
d. Sprint
Sprint itself is the heart of the Scrum method, which consists of several work plans that have been previously divided. Sprints are defined in the backlog that must be completed within a pre-agreed time.

e. Sprint Review dan Sprint Retrospective
After the sprint, the application must be reviewed to inspect and adapt the product backlog as needed. Next, make a retrospective sprint so that the application that has been developed gets input, whether it is by the requirements or not. It will be readjusted and added to the backlog if it does not match. If appropriate, then the application will be tested.

RESULT AND DISCUSSION

1. Product Backlog
The product backlog in developing this chatbot application is the development of machine learning models, deploying machine learning models using APIs, and displaying them in the Android mobile application.

<table>
<thead>
<tr>
<th>No</th>
<th>Backlog Name</th>
<th>Priority</th>
<th>Task</th>
<th>Estimated (Hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dataset gathering</td>
<td>100</td>
<td>Defines the dataset to be used</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>Model machine learning development</td>
<td>100</td>
<td>Searching for datasets on Kaggle or other platforms</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Training dan evaluation model</td>
<td>100</td>
<td>Develop models from existing datasets</td>
<td>40</td>
</tr>
<tr>
<td>4</td>
<td>Export model machine learning</td>
<td>100</td>
<td>Conduct training and evaluation of machine learning models</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>API endpoint development</td>
<td>100</td>
<td>Export the finished model</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>API testing development</td>
<td>100</td>
<td>Looking for references for developing APIs with Flask API</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>Prepare infrastructure server</td>
<td>100</td>
<td>Doing API testing after development</td>
<td>40</td>
</tr>
<tr>
<td>8</td>
<td>Deploy API endpoints on the server</td>
<td>100</td>
<td>API integration</td>
<td>16</td>
</tr>
<tr>
<td>9</td>
<td>Perform API testing after deployment</td>
<td>100</td>
<td>Perform API endpoint integration within the application</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>Create Android application design</td>
<td>100</td>
<td>Perform functional testing of applications after integration</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>Creating Android application pages</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Performing testing the application</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>a. Diagram Activity</td>
<td>100</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The table above shows the division of tasks and responsibilities to be carried out by each member involved, along with the allocation of time required to perform those tasks and responsibilities.

3. Sprint Planning
Sprint planning in developing this application will perform system design and analysis using UML diagrams.
The activity diagram above will occur when the user uses the application.

b. Apps Infrastructure

The developed application is Android-based and connected to an API deployed using the Google Cloud Platform through VM Instances. The machine learning model is located in Cloud Storage, and the firewall is configured to enable the API to be accessed by the application.

4. Sprint

Sprint itself is the heart of the Scrum method which consists of several work plans that have been previously divided. Sprints are defined in the backlog that must be completed within a pre-agreed time. In developing this chatbot application, it is divided into 3 sprints consisting of several tasks for each team.

5. Sprint Review dan Retrospective

a. Sprint Review

After the sprint, the application must be reviewed to inspect and adapt the product backlog as needed.

1) Sprint 1 (9 May – 21 May)

a) Dataset Gathering

Figure 4. Dataset Sample

The above is an example of a dataset the machine learning team obtained.

b) Model machine learning development

Figure 5. Sample Model Machine Learning

The above is a code snippet used in developing the machine learning model.

c) Training and evaluation model machine learning
The above Figure 6 shows the results of the training and evaluation.

d) Design UI/UX Apps

In application development, it is incomplete without any design to be developed. The above is an example of a design to be developed.

2) Sprint 2 (23 May - 4 June)

a) Export model machine learning

The machine learning model that is ready to use will then be exported to Google Cloud Storage. In Figure 8 above, the model was first uploaded to Google Drive.

b) API Endpoint Development

Figure 9 shows a snippet of the API code being developed so that the machine-learning model can be used in the future.

c) API Testing Development

After developing the previously conducted API, it is necessary to conduct testing to ensure that the API and machine learning model can function correctly.

d) Implement and Testing UI/UX Apps

Figure 11. Implement and Testing UI/UX Apps
From the design presented in Figure 7 above, it will be developed into an interactive application, as shown in Figure 11.

3) Sprint 3 (6 June – 16 June)
   a) Final Model Machine Learning

![Figure 12. Final Model Machine Learning](image)

In system development, there will inevitably be trial and error. Figure 12 shows the final machine learning model the Machine Learning team has improved.

b) Prepare Infrastructure Server

![Figure 13. Production Server with Compute Engine](image)

The Cloud Computing team will prepare a server that will be used for deployment, as shown in Figure 13, using Compute Engine.

c) API Deployment

![Figure 14. Deployed API Server](image)

Figure 14 shows that the server is ready to use and deployed successfully. The API server will remain active using Process Manager (PM2).

d) API Testing Production

![Figure 15. API Testing Production](image)

Before the Mobile Development team uses the API for integration with the application, the API must be tested first so that the results obtained can meet expectations.

e) API Integration dan Testing Production

![Figure 16. API Integration and Testing Production](image)

The Mobile Development team will integrate the deployed API into the application and perform retesting to ensure the results match the desired outcome.

b. Sprint Retrospective

Next, make a retrospective sprint so that the application that has been developed gets input, whether it is by the requirements or not. It will be readjusted and added to the backlog if it does not match. If appropriate, then the application will be tested.

**CONCLUSION**

Based on the above discussion regarding the development of cloud computing-based chatbot applications, the author can conclude that this chatbot application was developed based on cloud computing, which means the entire process of machine learning models and APIs was developed entirely using cloud technology. With this application, it is hoped that it will make it easier for users to make complaints and can reduce repeated questions that users often ask. The development of this chatbot application uses the agile method with the scrum framework. Where in the development process carried out is divided into several phases...
called sprints. The development of this application is carried out in 3 sprints starting from the first time this project was announced to completion.

**REFERENCES**


