Journal of Information Systems and Informatics
Vol. 6, No. 1, March 2024 e-ISSN: 2656-4882 p-ISSN: 2656-5935
DOI: 10.51519/journalisi.v6i1.650 Published By DRPM-UBD

Exploration of Modernity: Worship Reservation System at Rose of Sharon Church Salatiga Utilizing Flutter Framework

Davin Kurnia Hiuredhy[1], Henoch Juli Christanto[2], Christine Dewi[3], Stephen Aprius Sutresno[4]

[1,3]Department of Information Technology, Satya Wacana Christian University, Salatiga, Indonesia
[2,4]Information System Department, Atma Jaya Catholic University of Indonesia, Jakarta, Indonesia
Email: [1]davinkurnia@outlook.com, [2]henoch.christanto@atmajaya.ac.id, [3]christine.dewi@uksw.edu, [4]stephen.sutresno@atmajaya.ac.id

Abstract

Rose of Sharon Church (ROSC) Salatiga, as a rapidly growing spiritual community, faces challenges in managing the increasing number of congregants, especially during events and compliance with health protocols. This significant growth necessitates more effective attendance management to avoid overcrowding. This research addresses this need by developing the ROSC Worship Reservation application based on Progressive Web Apps (PWA) using the Flutter Framework. The application aims to simplify congregants' worship reservations, reduce crowds, and support church administrative tasks. The use of Flutter as the primary framework provides advantages in rapid development and a user-friendly interface. The results of the User Acceptance Test (UAT) show a high satisfaction rate, reaching 92%, regarding the application's functionality and user interface. Additionally, 84% of respondents state that the seating layout displayed by the application significantly aids in effectively choosing seats. It is hoped that the application will continue to provide benefits in managing congregational attendance systematically, efficiently, and in compliance with health protocols. The conclusion of this research emphasizes that the ROSC Salatiga Worship Reservation has successfully created an innovative solution to support church management amid the dynamic growth of congregational attendance.

Keywords: Worship Reservation System, Rose of Sharon Church, Flutter Framework, Progressive Web Apps, Mobile Application

1. INTRODUCTION

Rose of Sharon Church (ROSC) Salatiga, as a rapidly growing spiritual community, currently boasts more than 300 congregants actively registered through the My ROSC application provided by the central ROSC in West Surabaya. With this significant growth, there arises a need to design a more effective system to manage the attendance of congregants, especially in anticipation of queues that may occur when the church's capacity is limited [1].

The importance of attendance management becomes increasingly apparent, especially when the church plans special events or worship activities. In such situations, there is concern that some congregants may not secure the seats they need due to limited seating capacity. Recognizing this limitation, the church needs to find adequate solutions to ensure each congregant can experience meaningful participation in every church activity [2]. However, challenges arise when congregant data collection is conducted during worship services. This process consumes valuable time and energy, resources that could otherwise be allocated to spiritual services and other activities. Amid the pandemic, some church personnel have had to be redirected to serve as health protocol officers, intensifying the urgency for efficiency in administration [3].

Hence, there is an urgent need to develop an application that can assist in seat reservations when congregants wish to attend worship at ROSC Salatiga. This application aims not only to address queuing issues and manage church capacity but also to significantly contribute to recording the number of congregants present during each worship service. By leveraging Progressive Web Apps (PWA) technology and utilizing the Flutter Framework, "Worship Reservation at Rose of Sharon Church Salatiga" is expected to be an innovative and easily accessible solution. This application not only facilitates seamless seat reservations for congregants but also aids the church in efficiently managing and monitoring congregant attendance, especially in the face of ongoing pandemic challenges.

Through this research, it is anticipated that a deeper understanding will emerge regarding how congregants at ROSC Salatiga can easily make worship seat reservations. As a result, the Worship Reservation at ROSC Salatiga is expected to be a significant contributor in supporting the efficiency of church management, providing a better worship experience for congregants, and creating a more organized and secure spiritual atmosphere.

The previous research describes an information system designed to assist users in booking concert tickets and selecting seats online. Users of this application can reserve specific seats according to their preferences, reducing queues during ticket exchange on the day of the event [4]. The implementation of the online reservation system based on user preferences greatly facilitates users as they can select seats according to their preferences. Additionally, the application's display, in the form of a seating layout arranged according to the original room layout, also makes it easier for users to choose seats, similar to being there in person. The results of the research provide support for the conducted research, which also implements seat reservations based on user preferences for worship events at ROSC Salatiga. However, the differences lie in the research objectives, platforms, and also the case studies of the built applications. The previous research aimed to develop an existing application on the web platform for the motikdong.com website, while

this research aims to build a new system to replace the conventional process in the form of an application on the mobile platform for ROSC Salatiga.

In another research titled explains an information system for reserving meeting rooms and avoiding schedule conflicts during busy hours. Flutter was chosen as the framework due to its rapid development and user-friendly interface, facilitating users [5]. The utilization of the Flutter framework in the research proved to be highly beneficial for application developers, as it eliminates the need for hard coding and provides a well-structured code base, making it easy to use with ample documentation to support code implementation. Several advantages of the Flutter framework are suitable for implementing this research in constructing the Worship Reservation application for ROSC Salatiga, as this application will be extensively used and undergo continuous development. Therefore, with a well-structured code base, application developers can seamlessly continue coding to add various features in the future. A clear difference observed from previous research lies in the reservation object and the research case study. In the previous research, an application was developed for reserving a single conference room without room layouts, with a general research case study. In contrast, this research allows for personalized seat selection in the case study at ROSC Salatiga.

The research titled discusses the use of Firebase to provide information about dining places and expedite the food ordering process. Firebase offers libraries that allow integration with various platforms, including Android, and operates on a real-time concept [6, 7]. This helps maintain database performance when multiple users make reservations simultaneously. Based on the previous research, it can be concluded that the combination of Flutter and Firebase technologies in the Worship Reservation application for ROSC can be an effective and efficient solution in application development. This development is expected to assist the management of ROSC Salatiga in organizing attendance data systematically while adhering to existing health protocols.

2. METHODS

2.1. Research Methods

In this research, the research method utilized encompasses a series of structured stages to ensure the effective achievement of research objectives as depicted in Figure 1. The first stage involves gathering relevant data and information pertaining to the research subject. Subsequently, planning is conducted to organize the subsequent steps in the product development process. The next stage is the development of the initial form of the product, which is then preliminarily field-tested to obtain initial feedback. Following this, operational field testing is conducted to evaluate the

product's performance more deeply. Based on the results of these tests, the product undergoes operational revisions before proceeding to the main field-testing stage. Following the main testing stage, the main product is revised according to the obtained results before entering the final product revision stage. The final stage involves the dissemination and implementation of the final product to relevant stakeholders. Thus, this research method provides a systematic approach to ensure the quality and success of the final product according to the needs of the users.

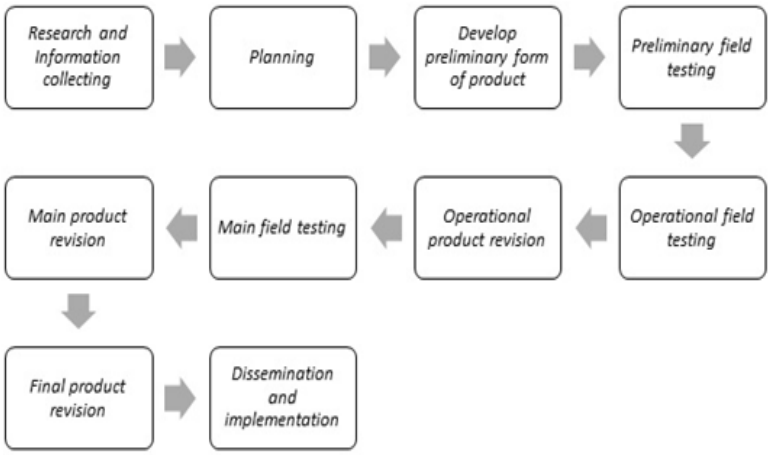


Figure 1. Research Methods

The challenge at hand is figuring out an innovative way to simplify the reservation process for congregants attending church events, while ensuring strict adherence to health protocols during worship at ROSC in Salatiga. Understanding this predicament, a decision was made to embark on the development project for the ROSC Salatiga worship reservation application.

After mapping out potentials and challenges, the next step involves gathering various information and literature studies. This data serves as a crucial foundation for strategic planning to overcome potential hurdles. Based on the compiled knowledge, the planning then leads to the selection of Flutter as the framework for developing the worship reservation application. This decision is made considering the rapid development speed associated with Flutter, surpassing other framework alternatives.

Flutter is an open-source framework developed by Google to build consistent and beautiful user interfaces (UI). Flutter has several advantages, such as a single codebase that can be used for multiple platforms, easy UI creation due to various

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

provided UI libraries like Material UI, and it possesses a rather complex architecture as illustrated in Figure 2 [8].

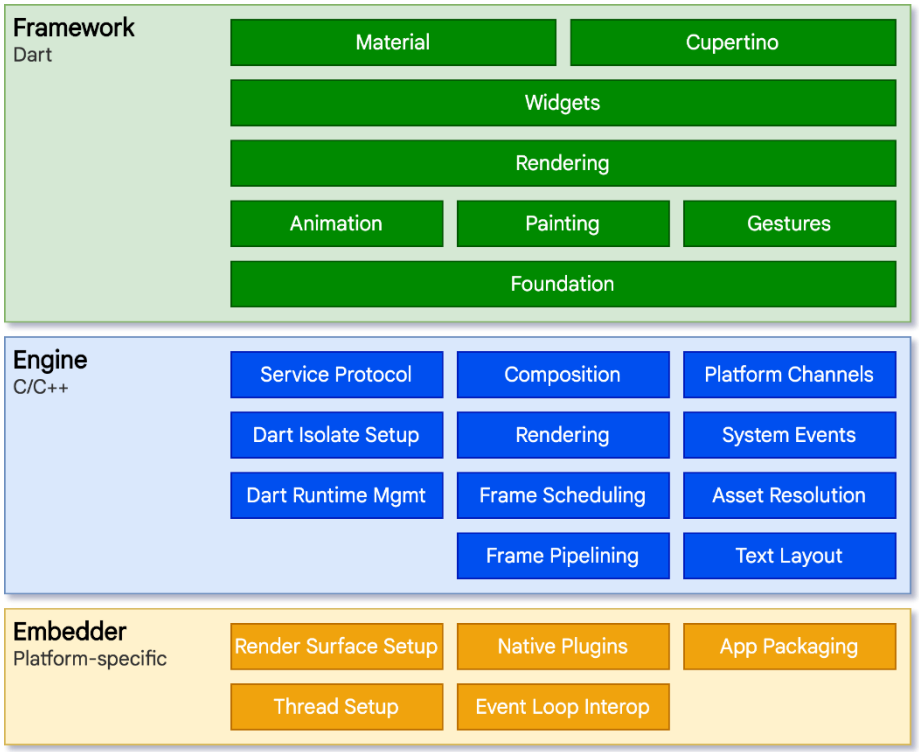


Figure 2. Flutter Framework Architecture

In Figure 2, it can be observed that the Flutter framework has a C/C++ base engine, which makes it easier and lighter to run Flutter. Additionally, Flutter also boasts high performance due to its use of its own rendering engine called Skia, and its development is facilitated by features like hot reload, allowing changes to the code to be immediately reflected in the application's interface when run on an emulator or physical device. Overall, the implementation of the Flutter framework facilitates research, especially in the development phase.

In the development phase, the primary focus is on creative aspects of application design [9]. This plan is then presented to the church for feedback, concurrently designing and projecting the needs of the web application before initiating the implementation phase with the most suitable programming language [10]. After completing the web application design phase, the church actively participates in validation, providing feedback, and adding additional features deemed essential.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

The subsequent stages involve usage testing, where the application undergoes scrutiny by the church to ensure optimal functionality. Revisions to the application will be concentrated on enhancing database recording performance, improving user interface, and delivering a more satisfying user experience [11, 12].

Once the revision process is complete, the mass production phase will be initiated. The matured web application will be made available for widespread use by the entire church congregation. Thus, this research journey involves an engaging series of stages, ranging from deep understanding, strategic planning, creative development, intensive testing, to achieving mass production of the ROSC Salatiga worship reservation application.

2.2. Use Case Diagram

Use Case Diagram is a modeling technique created to observe and design the behavior and functional requirements of the system to be developed [13]. The Use Case Diagram for the Worship Reservation Application ROSC Salatiga is illustrated in Figure 3.

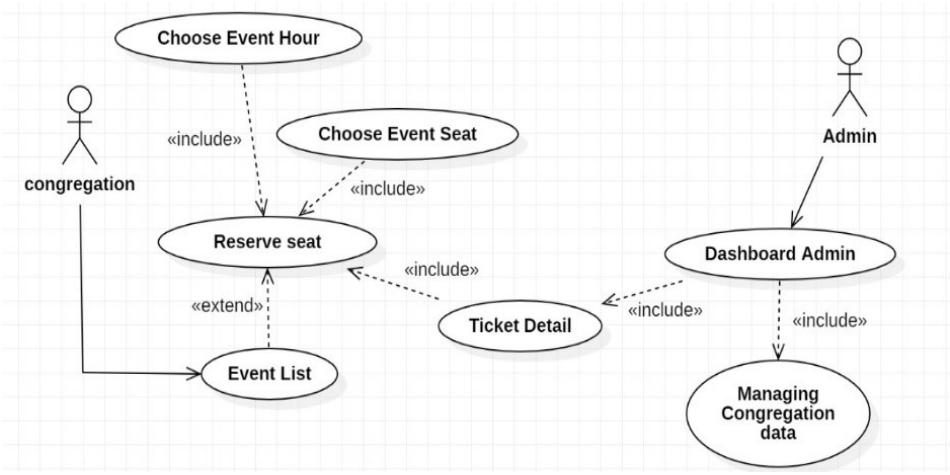


Figure 3. Use Case Diagram

Figure 3 illustrates the use case diagram that will be utilized in the system. The use case involves two actors: the congregant and the admin. The admin has access to open and manage the database, which includes congregant data and worship reservation information. On the congregant side, they can control their personal information, make and cancel reservations for worship sessions on their end.

Journal of Information Systems and Informatics
Vol. 6, No. 1, March 2024
p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

2.3. Activity Diagram

Activity Diagram provides an illustration or model of the workflow of the system [14, 15]. The Activity Diagram for the Worship Reservation Application ROSC Salatiga can be observed in Figure 4.

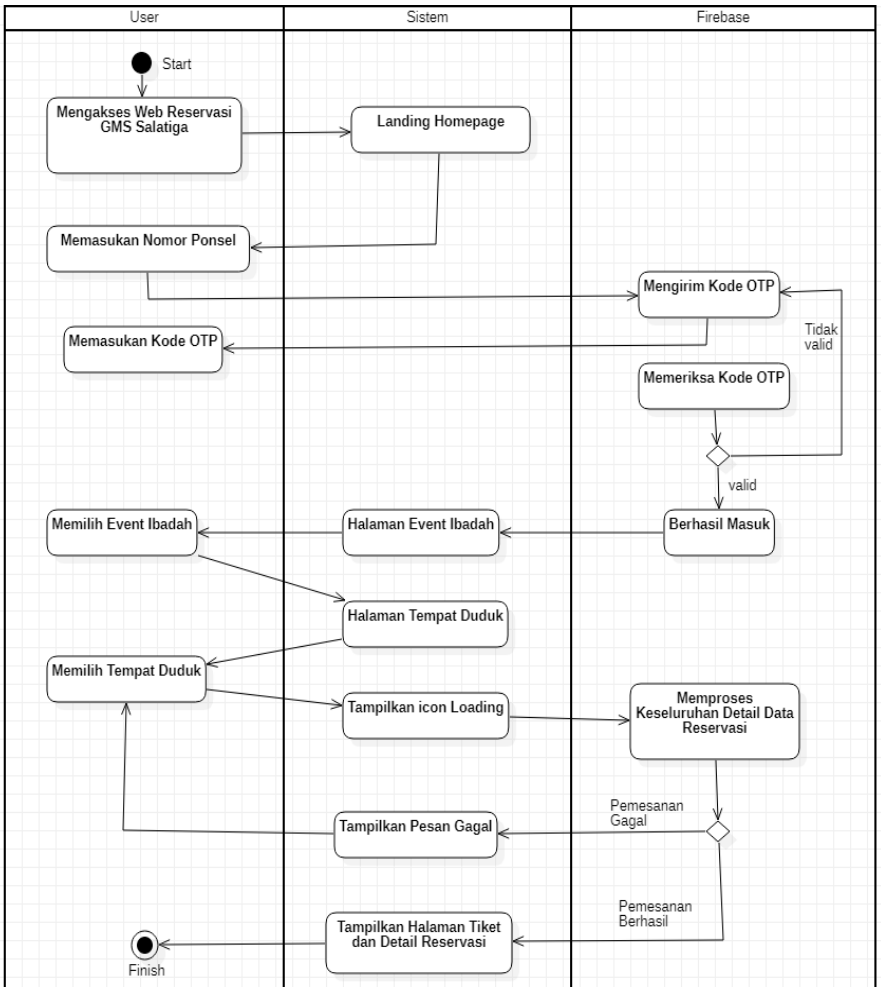


Figure 4. Activity Diagram

The explanation of Figure 4 is that the user or church member enters their name and phone number, and selects the pastoral care option in the login menu. After that, the system will check the database to see if the phone number of that church member has been registered before. If it has been registered, the system will display a warning that the phone number has already been registered and return to the login menu. If the phone number has not been registered before, the system will

display the available worship schedules. After the user selects a worship schedule, the system will check the available seat codes from the database and then display the layout of the worship seats. Next, the user selects an available seat code and presses the submit button. The system will display a loading message and check the database to see if the seat code is available or has already been reserved by another user who pressed the submit button earlier. If the seat code has already been reserved, the system will display the seating layout again and provide a warning that the seat has already been reserved by another user. If the seat code has not been reserved, the system will display the worship schedule data and the seat code for that user, which can be shown when they come to the church.

2.4. Class Diagram

Class Diagram serves as a general representation model of the structure of an application [16]. The Class Diagram for the Worship Reservation Application ROSC Salatiga is depicted in Figure 5.

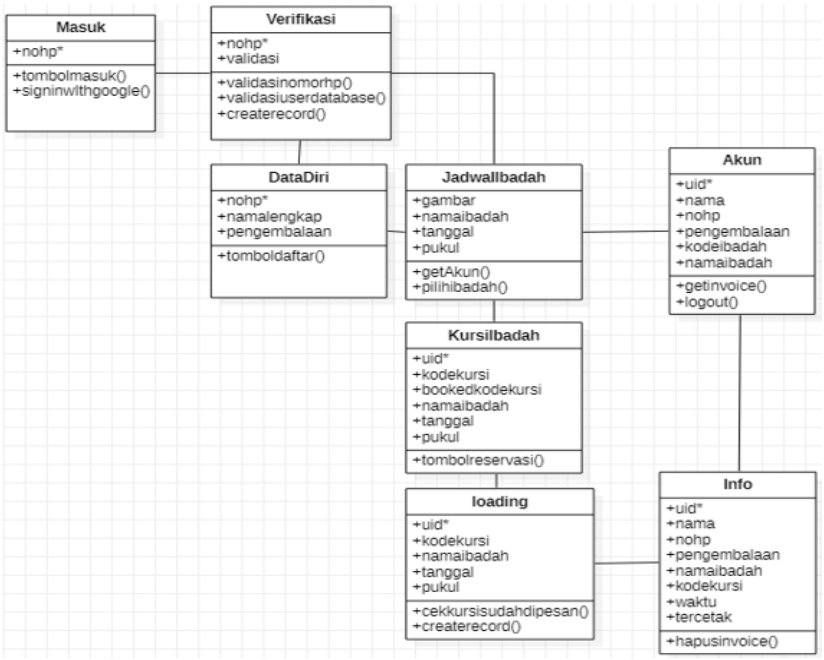


Figure 5. Class Diagram

In Figure 5, there are 8 classes employed in the creation of the Worship Reservation Application ROSC Salatiga. These classes encompass logging in using a phone number and Google account, selecting worship schedules, worship seats, and worship tickets.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

3. RESULTS AND DISCUSSION

The result of this research is the development and implementation of the ROSC Salatiga Worship Reservation web application using the Flutter framework. Several application views have been generated, accessible to users, including the home page, reservation page, ticket page, and several other pages.

3.1. Home Page

Users or congregation members must first register on the registration page, after which they can log in to the application. Upon successful login, the main page of this ROSC application will be displayed.

Figure 6. Home Page

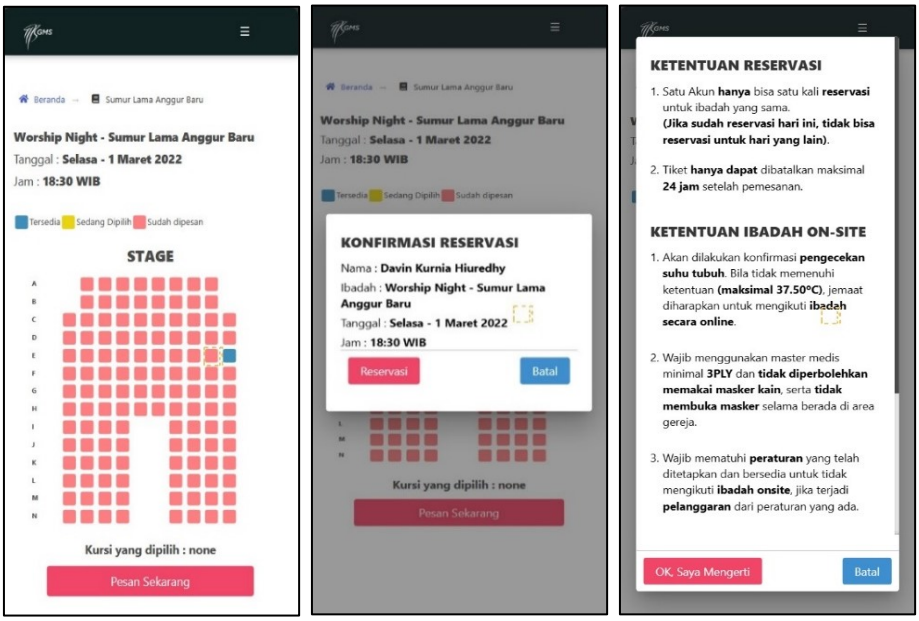
Figure 6 depicts the page displaying the homepage of the ROSC Salatiga worship reservation mobile application. On this page, the available worship schedule is shown, along with a button leading to the profile page in the top right corner of application.

3.2. Reservation Page

When the congregation has selected a worship schedule, they will be directed to a page, as shown in Figure 7, displaying detailed activity times and the layout of the

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

worship space along with seat codes that the user can choose and reserve for the worship service.



(a) (b) (c)

Figure 7. Reservation Page (a) Choose a Seat (b) Confirmation (c) Terms and Conditions

The seats highlighted in red indicate that they are already reserved, and thus, other users cannot book those seats. When a user selects one of the seats, it will turn orange to signify that the seat is currently being chosen. If a user hasn't selected a seat and tries to proceed with registration, a warning will appear stating that no seat has been selected, preventing them from advancing to the next page. Additionally, users cannot reserve seats again if they have already made a reservation for the same worship service and time. A warning will appear indicating that the user has already registered for that worship service.

3.3. Ticket Page

For the validation process of users who have reserved worship seats, a ticket and QR Code will be provided, as shown on the Ticket Page. Later, these will be confirmed by ushers present in the church.

Journal of Information Systems and Informatics
Vol. 6, No. 1, March 2024
p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

Figure 8. Ticket Page

3.4. Source Code

```
1.  Expanded(
2.   child: StreamBuilder<QuerySnapshot>(
3.     stream: FirebaseFirestore.instance
           .collection("Event")
           .snapshots(),
4.     builder: (BuildContext context,
           AsyncSnapshot<QuerySnapshot> querySnapshot) {
5.       if (querySnapshot.hasError) {
6.         return Text(
7.             "Terjadi kesalahan, silahkan coba lagi nanti");
8.       }
9.       if (querySnapshot.connectionState ==
             ConnectionState.waiting) {
10.        return Text("Sedang mengambil Data");
11.      }
12.      if (!querySnapshot.hasData) {
13.        return Text("Belum Ada Event ");
14.      } else {
15.        final list = querySnapshot.data.docs;
16.        return ListView.builder(
17.          itemBuilder: (context, index) {
18.            return Container(
19.              padding: EdgeInsets.only(left: 20, right: 20),
20.              child: Card(
21.                clipBehavior: Clip.antiAlias,
```

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

```
22.                 child: Column(
23.                   children: [
24.                     Image.network(
25.                       list[index]["Gambar"],
26.                       width: 600.0,  fit: BoxFit.cover,
27.                     ),
28.                     Container(
29.                       padding: EdgeInsets.only(top: 15),
30.                       child: ListTile(
31.                         title: Text(
32.                           list[index]["Nama-Ibadah"],
33.                           style: TextStyle(
34.                               fontSize: 25,
35.                               color: Colors.black),
                          ),
36.
```

Code 1. One of the Firebase CRUD Functions

Source code 1 displays one of the source code to fetch data from Firebase. The data is retrieved using the StreamBuilder widget, as shown in line 2, to ensure that the web application continues to run even if the data from Firebase has not been fetched or is currently being fetched. The code for fetching data is demonstrated in line 3, where to retrieve data from Firebase, it starts by entering the Collection, then goes to a specific Document, and inside the Document, there is the data that needs to be fetched. After that, the retrieved data is placed within the Card widget, which contains an image and a Container widget to hold text.

```
1.  Container(
2.   height: MediaQuery.of(context).size.height / 15,
3.   width: MediaQuery.of(context).size.width / 6,
4.   padding: EdgeInsets.only(
5.       left: 5, right: 5, bottom: 5),
6.   child: FlatButton(
7.     color: isC10Booked ? Colors.red : C10 ? Colors.amber :
Colors.grey, ),
```

Code 2. Display and Function of Buttons

In Source Code 2, there is a button display wrapped in a container to allow customization of the button's dimensions, as well as padding, margin, and position. The seat button in line 7 has conditions: if the user presses the seat button, it will change to a yellow color. If the seat button has already been reserved by another user, it will be red.

3.5. System Testing

System testing is conducted to verify whether the functions of the application are working properly or not [17, 18]. This testing consists of Black Box Testing and

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

User Acceptance Test. Black Box Testing is performed to determine if the functions of the ROSC Salatiga worship reservation application are operating correctly.

Tabel 1. Black Box Testing

Tested Function	Condition	Expected Output	Actual Output	Validity
Input validation if there are unfilled forms	When the register button is clicked	Successfully displays a warning message	Successfully displays a warning message	Valid
Phone number validation if already registered	When the register button is clicked	Successfully displays a warning message	Successfully displays a warning message	Valid
Load Worship Schedule	Displays a list of worship schedules	Successfully displays the available worship schedules	Successfully displays the available worship schedules	Valid
Load seat codes	When the user has selected a worship schedule	Successfully displays the available worship schedules for that worship time	Successfully displays the available worship schedules for that worship time	Valid
Validation of seat code if already reserved by another user	When the user has selected a seat code and clicks the submit button	Successfully displays a warning message	Successfully displays a warning message	Valid

The comprehensive results of the ROSC Salatiga worship reservation application's system testing, as outlined in Table 1, reveal that each functional test, including input validation, phone number validation, loading worship schedules, and seat code validation, produced valid outcomes. Based on the results of Black Box Testing, it can be observed that the application's functionalities operate successfully, aligning with expectations. Consequently, it can be concluded that the application is functioning well and correctly, meeting the criteria set forth during the testing process.

Meanwhile, User Acceptance Test is carried out to assess whether the application is deemed suitable for use by the relevant users [19, 20]. The testing evaluation scale is determined on a scale of 1-5, where 'Strongly Disagree' is rated 1, 'Disagree' is rated 2, 'Neutral' is rated 3, 'Agree' is rated 4, and 'Strongly Agree' is rated 5.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

Tabel 2. The Total Number of Responses for the User Acceptance Test (UAT) of the ROSC Salatiga Worship Reservation Application

No.	Statement	Number of Responses				
		1	2	3	4	5
1	This application helps me in making reservation orders at ROSC Salatiga.	0	0	1	2	7
2	The layout on the seating map page is helpful for me in choosing seats effectively.	0	0	2	4	4
3	Information about worship services and events at ROSC Salatiga is displayed well.	0	1	2	3	4

Table 2 contains the response data from 10 churchgoer respondents of the ROSC in Salatiga who have filled out a questionnaire using the Likert scale. This methodology allows for a quantitative analysis of user acceptance, providing insights into the effectiveness and favorability of the ROSC Salatiga worship reservation application based on user feedback. The Likert scale is used to calculate the percentage for each question. The following is the formula (1) for the Likert scale [21]:

$$\rho = \frac{f}{n} x100\% \quad (1)$$

where:

ρ = percentage
f = frequency of questionnaire responses
n = total ideal score

Table 3 displays the outcomes of the User Acceptance Test (UAT) presentation for the ROSC Salatiga Worship Reservation Application. The ratings, ranging from 1 to 5, represent the frequency of data occurrence for each statement, and the Total and IKK (Percentage of Agreement) are calculated accordingly.

Tabel 3. The Results of the User Acceptance Test (UAT) Presentation for the ROSC Salatiga Worship Reservation Application

No. Statement	Rating Scale * Frequency of Data Occurrence					Total	IKK (%)
	1	2	3	4	5		
1	0	0	3	8	35	46	92%
2	0	0	6	16	20	42	84%
3	0	2	6	12	20	40	80%

For the UAT rating interval from 0% to 100%, it is defined as follows: 0% to 15% is declared as strongly disagree, 16% to 30% is considered disagree, 31% to 45% is regarded as neutral, 46% to 60% is considered agree, and 61% to 100% is declared as strongly agree.

Analyzing the results from Table 3, it is evident that 92% of respondents affirmed that the ROSC Salatiga Worship Reservation Application has significantly aided them in making worship reservations. Additionally, 84% of respondents acknowledged that the seating layout page has been instrumental in their seat selection process. Furthermore, 80% of respondents expressed satisfaction with the well-displayed schedule information for worship and events at ROSC Salatiga. These percentages indicate a high level of agreement among users regarding the application's effectiveness and user-friendliness.

The user response indicates that the application has effectively addressed the problem and objective of this research, which is to facilitate users without the need to physically attend church to choose their seats for worship events. The administrative aspect also does not need to worry about recording reservation data, as all data is automatically recorded through the application. Furthermore, the application's interface is user-friendly, making it easy for users, especially during seat selection. These results are supported by user responses, which tend to lean towards selecting 'Agree' or 'Strongly Agree' responses in that regard.

4. CONCLUSION

It can be concluded that the development of the ROSC Salatiga Worship Reservation application has had a positive impact on administrative systems and enhanced the congregation's experience in making online worship seat reservations. This can be seen in the UAT results, which indicate that 92% of respondents agree that the application assists them in making worship reservations, and 84% of respondents agree that the seating layout selection displayed by the application is very helpful in effectively choosing seats. Thus, the results from this built application can replace conventional processes and aid in administrative tasks. Additionally, the implementation of the Flutter Framework helps simplify and accelerate application development, particularly due to the various features provided by the framework. Therefore, this application is expected to continue providing positive benefits and improving the management of structured and efficient congregation attendance in the future. As a development suggestion, various features could be added, such as online offering transfers using QR codes or other methods, reminder notifications that can be synchronized with Google Calendar or smartphone systems, and various other features could be incorporated.

REFERENCES

[1] H. J. Christanto, S. A. Sutresno, J. V. M. Bata, D. J. C. Sihombing, P. K. Prihanto and D. V. T. Linestyo, "Pelatihan Penggunaan Sistem Informasi

Gereja Berbasis Android Pada Gia Purwodadi Kabupaten Grobogan," *RESWARA: Jurnal Pengabdian Kepada Masyarakat*, vol. 5, no. 1, pp. 119-127, 2024.

[2] H. J. Christanto, S. O. B. Widiarto, S. A. Sutresno and D. V. T. Linestyo, "Pengembangan Aplikasi Sistem Informasi Gereja Isa Almasih Jemaat Purwodadi Pada Platform Android," *Jurnal Teknologi Informasi dan Komunikasi*, vol. 8, no. 1, 2024.

[3] C. Dewi, A. P. S. Chen and H. J. Christanto, "Deep Learning for Highly Accurate Hand Recognition Based on Yolov7 Model," *Big Data and Cognitive Computing*, p. 53, 2023.

[4] Wahyudi, "Pengembangan Aplikasi Sistem E-Ticketing Konser Musik dengan Seating Number pada Website motikdong.com," *Akrab Juara : Jurnal Ilmu-ilmu Sosial*, vol. 5, no. 1, pp. 180-187, 2020.

[5] A. Praveen, K. Nanda, N. Rajith, N. Giriraj, R. Radhika, N. Mahesh, K. Vishnu, T. Anjali and S. Sarath, "Conference Room Booking Application using Flutter," *in International Conference on Communication and Signal Processing (ICCSP)*, Chennai, India, 2020.

[6] G. R. Payara and R. Tanone, "Penerapan Firebase Realtime Database Pada Prototype Aplikasi Pemesanan Makanan Berbasis Android," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 4, no. 3, pp. 397-406, 2018.

[7] F. Afira and J. W. Simatupang, "Real-Time Web-based Dashboard using Firebase for Automated Object Detection Applied on Conveyor," Green Intelligent Systems and Applications, vol. 3, no. 1, pp. 35-47, 2023.

[8] P. N. Ladke, P. D. Vedpathak and R. R. Saraogi, "Flutter," *International Journal of Creative Research Thoughts (IJCRT)*, vol. 10, no. 5, pp. 39-44, 2022.

[9] H. J. Christanto and Y. A. Singgalen, "Analysis and Design of Student Guidance Information System through Software Development Life Cycle (SDLC) and Waterfall Model," *Journal of Information Systems and Informatics*, vol. 5, no. 1, pp. 259-270, 2023.

[10] S. A. Sutresno, H. J. Christanto, Y. A. Singgalen, E. Saputra, B. D. Farabi, B. A. Rig, R. Rahmananta and O. Hutapea, "User interface and user experience design for the TB. Berkat rezeki website," *Jurnal Mantik*, vol. 7, no. 4, pp. 2962-2973, 2024.

[11] S. A. Sutresno, H. J. Christanto, Y. A. Singgalen, D. Denis, J. Kevin and S. Mavish, "Design and Development of a Mobile Application for Social Assistance," *Journal of Information Systems and Informatics*, vol. 5, no. 4, pp. 1257-1273, 2023.

[12] B. Suratno and J. Shafira, "Development of User Interface/User Experience using Design Thinking Approach for GMS Service Company," *Journal of Information Systems and Informatics*, vol. 4, no. 2, pp. 469-494, 2022.

[13] F. Sonata and V. W. Sari, "Pemanfaatan UML (Unified Modeling Language) Dalam Perancangan Sistem Informasi E-Commerce Jenis Customer-To-

Customer," *Jurnal Komunikasi, Media dan Informatika*, vol. 8, no. 1, pp. 22-31, 2019.

[14] T. Ahmad, J. Iqbal, A. Ashraf, D. Truscan and I. Porres, "Model-Based Testing Using UML Activity Diagrams: A Systematic Mapping Study," *Computer Science Review*, vol. 33, pp. 98-112, 2019.

[15] M. Abbas, R. Rioboo, C.-B. Ben-Yelles and C. F. Snook, "Formal modeling and verification of UML Activity Diagrams (UAD) with FoCaLiZe," *Journal of Systems Architecture*, vol. 114, p. 101911, 2021.

[16] E. R. Subhiyakto and Y. P. Astuti, "Aplikasi Pembelajaran Class Diagram Berbasis Web untuk Pendidikan Rekayasa Perangkat Lunak," *Simetris Jurnal Teknik Mesin Elektro dan Ilmu Komputer*, vol. 11, no. 1, pp. 143-150, 2020.

[17] H. J. Christanto and E. Sediyono, "Analisa Tingkat Usability Berdasarkan Human Computer Interaction (HCI) Untuk Sistem Pemesanan Tiket Online Kereta Api," *Jurnal Sistem Informasi Bisnis*, vol. 10, no. 2, pp. 163-172, 2020.

[18] H. J. Christanto, S. A. Sutresno, A. Denny and C. Dewi, "Usability analysis of human computer interaction in google classroom and microsoft teams," *Journal of Theoretical and Applied Information Technology*, vol. 101, no. 16, pp. 6425-6436, 2023.

[19] Q. Xianjun, C. Minghong and L. Xiaoli, "User Acceptance Model of Government Microblog and Its Empirical Study," *Procedia Computer Science*, vol. 162, pp. 940-945, 2019.

[20] I. Masudin, D. P. Restuputri and D. B. Syahputra, "Analysis of Financial Technology User Acceptance Using the Unified Theory of Acceptance and Use of Technology Method," *Procedia Computer Science*, vol. 227, pp. 563-572, 2023.

[21] K. Memmedova and B. Ertuna, "Development of a Fuzzy Likert Scales to Measure Variables in Social Sciences," *Information Sciences*, vol. 654, p. 119792, 2024.