A-Qyu General Purpose Cloud-based Queue Management

by Erico Darmawan Handoyo, Sulaeman Santoso, Daniel Jahja Surjawan

2023 1st IEEE International Conference on Smart Technology (ICE-SMARTec)
Bandung, Indonesia, 17-19 July 2023

A-Qyu General Purpose Cloud-based Queue Management

Erico Darmawan Handoyo
Faculty of Information Technology
Maranatha Christian University
Bandung, Indonesia
erico.dh@maranatha.ac.id

Sulaeman Santoso
Faculty of Information Technology
Maranatha Christian University
Bandung, Indonesia
sulaeman.santoso@maranatha.ac.id

Daniel Jahja Surjawan
Faculty of Information Technology
Maranatha Christian University
Bandung, Indonesia
daniel.js@maranatha.ac.id

2023 1st IEEE International Conference on Smart Technology (ICE-SMARTec) | 979-8-3503-0615-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICE-SMARTECH59237.2023.10461967

***Abstract*—Queueing in a venue is a part of our everyday lives. Digital queues have been implemented with great success at multiple venues. However, there are several disadvantages to traditional queue management. As an example, the traditional queue management system (QMS) requires the participant to be present at the venue. Traditional QMS also usually uses specific hardware setup, a computer or an IOT device. This research analyzes general uses of QMS from several specific QMS with its respective strengths and weakness and then designs a general-purpose solution to QMS using cloud technology for storage that provides real time data. To evaluate the performance of the QMS, standard application testing is conducted with additional survey to assess system usability when faced with a general queue situation. The survey is done within a university setting with two classes of students. Survey results found that the proposed solution does provide better queue management for general purpose use.**

Keywords—cloud application, mobile application, queue management

I. Introduction

Ever since the world became largely populated, there is a need for queue management in many places. Places such as hospitals, concert halls, restaurants, and places that serve a high number of people would generally require some form of queue management. Before the digital age, queue management was done manually. Writing names on books or using a numbered card to show the position of the individual within the queue is one of the traditional yet effective approaches. There is of course a limitation to this manual approach. The first limitation is that you need to provide a pin, a card, or some type of material to indicate a position on the queue. Since the material is physical it is highly subjected to damage, tampering, or simply loss. Queue management using a written queue also has a limitation of not being flexible with its queue. In a traditional queue it is difficult to query a position within the queue, or to estimate when a person would be called. This holds true especially when the number of people in a queue increases. There is also a limitation in space for the queue. For small rooms, if the number of queues exceeds the room capacity, some of the people in the queue will have to wait outside of the premises. For a large queue and large room, the queue number would be hard to see and to call a person would be hard. Not to mention that the person queueing will have no knowledge or estimation of how long they are going to wait. These limitations could be reduced or removed using modern technology.

Throughout the years, inventions and technology have upgraded traditional queue approaches to better deal with different environments. To deal with large rooms for example, queue numbers are not being shouted out by a person, but rather shown on a screen or broadcast using a speaker. Other devices such as IOT tokens and large screens are used to call a person in queue. In 2015, Mohhamed Gazal et.al proposed an IOT smart queue management with the use Arduino board and a mobile phone app that uses NFC, this system however requires the user to scan or uses NFC to update queue position and thus needs to be near the premise to do that [1]. In 2016, Jidin et.al [2] developed a queue management system by using an arduino board that will take a customer ticket via short message and call them using another short message. This system also requires a desktop application to show its GUI. Siti Nur Huda [3] in 2019 also develops queue management using arduino. This application requires the customer to scan QR code in order to queue. Chi-Yi Lin et.al [4] developed a Bluetooth beacon that will inform customers when their number is being called. This device helps people to observe the queue and estimate when they are going to be called. Rukiton smart queues use a combination of offline and online techniques using IoT and web applications to accommodate its user [5]. In 2021, Mai Abusair et.al uses priority queue algorithm and a mobile phone app to create a QMS that will prioritize queue according to critical needs, however this approach is highly specialized because of the priority system and will only work in the hospital setting where it is proposed at and not suitable for general purpose use [6].

These approaches are generally targeted toward specific instances of use, with a specific setting and hardware requirements. For example, queue management system developed by Sumit Soman et al [7], Rukiton and TiQS by Amirul Syafiq [8] was specifically designed for hospitals settings, *Queuewe* [9] was intended to be in a restaurant setting, and other QMS with their own settings. Therefore, this approach lacks the flexibility to be implemented in different general cases. For example, a hospital queue will only need number information while a restaurant queue needs order information, and other types of queues need other types of information. Different types of settings would also impact the timing requirement of queues, this is the importance of queue, the way to deal with absent queues, and other types of queue modification. And of course, different settings will also affect different types of hardware that are available.

There are cases where queue management is required to be flexible in its uses, and hardware. Take for example, a distribution food line for refugee/disaster victims, or an impromptu vaccine distribution in a pandemic area. These cases would require queue management, but a queue management system with specific hardware installation will not be suitable or will cost too much time and energy to be deployed (when for instance the queue will only last a couple of days). Of course, a manual queue management system is applicable in these cases, but then the problem with manual queue management will also be present. This research proposes a queue management system that is designed for general purpose queues. This system uses mobile devices to queue and to set up the queue without the need for any other external hardware. This will make the system available at very little preparation time with hardware that is common to its user.

This research will also address some of the limitations of today's digital queue management system. One of which is to expand the queue management system by using cloud stream technology. This technology will allow a flexible queue management system. Most queue management systems, traditional or modern, require that the user should stay within or in proximity with its premise. With additional hardware like a Bluetooth beacon for instance, user can leave the premise but still stay within the range of the devices. With cloud stream technology this research proposed queue management that will allow the user to leave the premises and return without missing the queue. This will leave the user with a lot more freedom to do activities that would otherwise be spent waiting in line. In addition, by being a cloud database, users can queue anywhere there is an internet connection.

II. General Purpose QMS

Several attempts at general purpose QMS have been recorded. A "walk-away" QMS by Aizan et al [10] uses mobile applications and relational databases to store queues. Short message service or telegram app messages are then used to inform the user when their number is being called. But since the notification only happens during events, a user cannot query the queue at an arbitrary moment to know or to estimate when they are being called. Not to mention that this system requires a dedicated server to store the queue which makes it less portable. QMS from Manoon et. al [11] manage the hospital queue online where customers, patients and stakeholder can access theirs queues remotely over the Internet through a web application. While it can inform users of the current queue on demand, this system still lacks the flexibility for general purpose queues.

To develop a general purpose QMS, one should be aware of the requirements needed. By using several common queues scenarios, a generalization is then constructed. Table I shows a couple of common scenarios for queues. As seen, some scenarios have fixed location, and some may have to move places. Some will have little to moderate budgets while some can spare a moderate – high budget. This budget determines what number or kind of devices can be used to accommodate queue scenarios. Queue scenarios also differ in the amount of added information needed. Some scenarios don't require any information than a person's queue number, while in the example of food truck queues, the order information is required.

Analyzing several different queue scenarios, this research proposed the requirement for general purpose QMS as the following:

- System requires as little as possible additional hardware. Additional hardware would mean additional cost and time to set up. This will also mean the system would not be readily available on quick notice.
- System requires little time to set up and use. A general QMS should be simple and intuitive to use so that the customer would not need additional training.
- Information could be added alongside the queue number to enable different scenarios of queueing. Since the information requirement is varied, a common type of information should be use.
- Users should be able to query queue state at any given time or should be given an update on the status of their queue in real time.
- System should be able to anticipate user behavior such as leaving the queue or advancing multiple queue numbers at the same time.

TABLE I. Sample Queue Needs

No	Queue Scenario	Location	Budgets	Time req	Added Info
1	Class presentation queue	Anywhere (variable)	Little-moderate	limited	No
2	Hospital /Medical Queue	Hospital/ Medical Facility (fixed)	Moderate -high	flexible	No / User Id
3	Food Truck Queue	Anywhere (variable)	Low-moderate	flexible	Order info
4	Distribution Queue	Anywhere (variable)	Low-moderate	limited	No / User Id

Requirement 1 could be met by using devices that are already normally present in general situations, namely the mobile phone. By using mobile apps as queue provider and as queue user, the need for additional hardware such as a beacon or an IOT device could be removed, and of course this will save cost for the device. Requirement 2 could be met by designing a QMS experience that mimics the traditional queue experience. This will enable users to understand the flow of the application better. And the rest of the requirements can be met within the application design and implementation.

III. Methodology

A-Qyu Digital QMS is an application that is designed to fulfill all the requirements of a general purpose QMS. When dealing with a queue scenario there are generally two types of roles, the creator, and the user. The creator is the party that creates the queue and manages it (create, update, delete, and more). In a real-life scenario this would be the hospital or the restaurant that needed to have a QMS. The user is then the person inside the queue. The user is then able to queue up or leave the queue when needed. Seeing that the functionality of both the creator and the user are quite different, A-Qyu system is divided into 2 separate applications, the A-Qyu creator and the A-Qyu user.

Both applications will then need to communicate with one another. Instead of peer-to-peer communication, it is decided to instead use a cloud stream database that would then be accessible to both applications. Some of the research mentioned above uses a server to store data in the form of a database. But since the general purpose of a queue requires as little hardware addition as possible, a cloud server to store data would be a more fitting approach. Cloud databases such as google firestore provide storage that can be accessed in real time [12]. This provider also allows data streaming. Data streaming means that if the data on the server is changed. The changes will then be pushed to all the devices that are subscribed to that stream of data. This technology is perfect for the functionality of a QMS since then whenever a queue is updated, the changes within the queue will be pushed to all devices subscribing to that queue. And since by nature cloud databases are available wherever an internet connection is available, then a QMS based on cloud will be accessible without the need to invest in an on-premises server or other similar hardware.

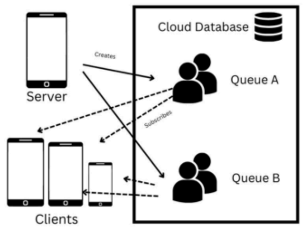

Fig. 1. QMS data flow

Fig. 1 shows the connection between A-Qyu creator and A-Qyu user and its functionality. As shown in the picture the communication between applications is done via data in the cloud server. And since the data in each application is the most recent one, then users could at any given time query their position in the queue.

To facilitate general purpose QMS requirements, A-Qyu provides a queue number with a text field. This text field value could then be adjusted by the creator to whatever information needed. For example, for a restaurant, this value could be the number of menus ordered. While for a vaccine queue this can be the person's personal id. By providing the user with a large container for information, A-Qyu can accommodate different types of information requirement.

To create a queue, the creator must log into the creator application. The creator can then create a queue on the main page. When a creator creates a queue, the creator is then asked the name of the queue and if additional information is needed. For a simple queue, this option would not be necessary while for other types of queues this option can be used accordingly.

When a queue is created, a collection is stored on the cloud server. The structure of the data is as follows: a user data will have two sections: the creator part and the user part. This is to ensure a creator can also queue using the same account. The creator part will consist of queue data while the user part will consist of the data needed for the queue. The data on the user part consists of two fields: the username and the active ticket. The active ticket path refers to a queue document in the cloud database.

The queue data structure consists of several single fields and a collection of tickets. This collection of tickets refers to each user in a queue. While the fields store the name and the title for the information part of the queue. By providing title field for the information, the field is then malleable to use on different queue scenarios according to their needs.

Every ticket in the queue will have information regarding the user. This contains information on the user's email, name, notes for queue, queue number, and a field called IsCalled. The field isCalled is set to true when the user's number is being called. This change will be pushed to the device by the cloud database and provide an updated queue state for the user. The note field provides additional information for each individual user. The number field provides the user with a unique identifier for the queue.

One of the important parts of a general purpose QMS is that the system should be flexible. There are many scenarios that could happen within a queue that will affect the current queue. These scenarios should be handled so as not to disturb the user or the queue itself. These are some of the scenarios that are dealt in A-Qyu:

- User spamming to join the queue. In a lot of digital QMS there is no limitation on queue and such a person can join the queue multiple times and deny service to other users. This scenario is dealt with within A-Qyu by limiting users to join A-Qyu only a single time.
- User leaving the queue. When a user leaves A-Qyu or forfeits their position in the queue, the queue should still move up without problems. When a user leaves the queue in A-Qyu, their ticket number is erased, and the next number will then be available automatically.
- Moving the queue. There are a lot of situations that would require a queue to move one by one. But in other situations a queue could move up in batches. A-Qyu provides the ability to call several tickets at the same time and thus removing this problem.
- Idle user, one of the problems with digital QMS is that sometimes users will not be aware that their number has already been called. A-Qyu solves this problem by letting the creator choose whether to remove the tickets or leave it for the user to come at a later time.

The common scenario for A-Qyu QMS would be as follows. The creator would first need to create an A-Qyu queue by inputting the name of the queue and its information label. When the queue is created, a QR code will be generated. This QR code functions as a means for the user to join the queue. The creator can then use the QR code in various ways, it can be printed on a paper, distributed via other media, or just used within the application.

The user app will then need to scan the QR code to join a queue. When the user joins a queue, the user will receive information regarding their current number and the state of the queue. By viewing the current queue number, the user may estimate whether their turn will come sooner or later thus enabling them to do other work. Fig. 2 shows the user main page with the user's number shown in blue while the current queue is shown in a smaller number below. If the queue creator sets additional information, then the main page will also show the information as a label within the same page.

When viewed from the A-Qyu creator, the queue information will show every user in the queue with its respective information sorted ascending by their id. Fig. 3 showed the creator queue page. The uncalled queue is shown with a colored card while the called queue is shown with a white card. The creator may choose to process the next queue multiple times to accommodate batch processes. The creator may also reset the queue number when it is needed.

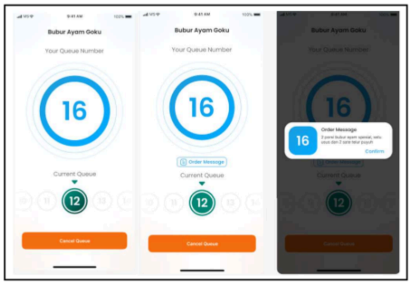

Fig. 2. User main page

When the creator calls a queue number, that user will then receive an update on its main page showing that their number is being called. This update is done automatically via cloud server. When the ticket data changes on the server, the server will push the update on each stream that are connected to that data. When the queue is being called the user may not cancel the queue, this is done to protect the integrity of the queue in several scenarios such as an order at a restaurant. If it is truly necessary when the creator reset the queue number, the queue will be reset along with all the tickets within it.

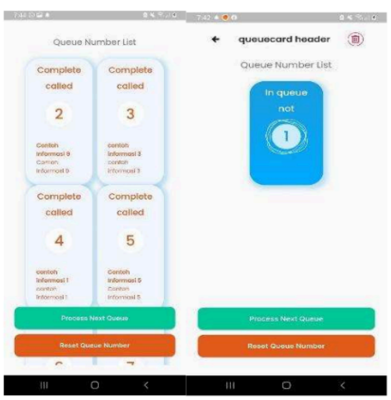

Fig. 3. A-Qyu creator queue page

To accommodate rapid changes in development, A-Qyu QMS was developed using a clean architecture approach where the application was separated into dependency and concerns [13]. To accommodate separation of concerns, A-Qyu QMS is developed into three projects, the A-Qyu creator, A-Qyu user, and A-Qyu package. The A-qyu package consists of two main parts, data, and domain. Data consists of repositories and services. This part of package handles the connection to cloud storage. By keeping the repositories separate from the services, different cloud providers could be supported by merely swapping the repository. Domain consists of entities and usecases. Every functionality in both A-Qyu user and A-Qyu creator is provided within this domain package. By doing so, A-Qyu user and A-Qyu creator project can focus on high level abstraction and user interfaces. This is another aspect of clean architecture which is the dependency rule where high level component should depend on low level component and not the other way around. Fig.4 shows the structure of A-Qyu QMS's clean architecture.

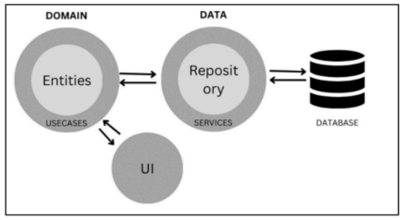

Fig. 4. Clean architecture structure of a-qyu QMS

IV. EVALUATION

Evaluation process for A-Qyu QMS was conducted in two stages. The first stage was a software evaluation to test the functionality of the software while the second stage of the evaluation was a survey to test usability and design of the software. A-Qyu QMS consists of several functionalities, to evaluate each core functionality a blackbox test is conducted by running the software on the device multiple time on different scenarios. Table II shows the evaluated feature along with its test results.

A survey is conducted on two classes of IT university students (a number of 25 students) acting as subscribers of a queue. The students are chosen as survey sample for several reasons. The chosen students are familiar with different types of queues and are familiar with different digital QMS. The IT students are also familiar with software engineering and different types of application so that they are qualified to evaluate digital solutions better.

To simulate general purpose uses, a queue scenario is devised on the spot without additional device for several uses. One of the cases was for a test scenario, the student needs to queue to present their homework using the QMS. The student plays a part as the subscriber of the queue using their personal mobile phones. The student is also asked to conduct standard queue behavior mentioned before such as leaving queue or spamming join to a queue. Students are then asked to fill in a survey evaluating the efficiency and uses of the application. The survey consists of 5 questions. Each question can be answered with a number from 1 to 5, with 1 completely disagree with the statement and 5 being totally agreeing with the statement. After the 5 questions respondents are asked to give their input on the system. Table III lists the questions as follows.

For Q1, 64.7% of users found the application to help create queues and to join them. 35.3% of respondents give a value of

4 on the questions, and no respondent gives a value below 4 for this statement. For Q2, it is shown that more than half of the respondents give a value of 5 that the application creates a better comfortable queue experience. Only 1 respondent is neutral on the statement. Q3 also shown that only 2 respondents give neutral value on the ease of use of the application. Relating the value with the written respondent input. The value is given due to login error when using google services and not on the core functionality of the system. Another feature that is not yet implemented is push notification on the application that is deemed important to some users.

TABLE II. SAMPLE QUEUE NEEDS BLACKBOX TESTING OF FUNCTIONALITY

No	Functionality	Expected Result	Test Result
AU1	User login	On success: user log in into account showing Active Queue Ticket or Active Queue On fail: user shown an error message	User Login test completed successfully on both scenarios.
A2	Create Queue	On success: creator is given a QR Code for newly active queue, Cloud server creates a data entry for aforementioned queue. On fail: creator is given an error message	Create queue test completed successfully on both scenarios.
A2	Call Ticket	On success: first non-active ticket will be set to active, next queue number will jump to the next non active ticket. User will be informed of queue changes On fail: creator is given an error message	Call ticket test completed successfully on both scenarios.
A3	Reset Queue	On success: all tickets on the queue will be deleted and the queue number is returned to one. On fail: creator is given an error message	Reset queue test completed successfully on both scenarios.
U2	Join queue	On success: user is given a queue number and a queue path in cloud storage. On fail: user is given an error warning	Join Queue test completed successfully on both scenarios.
U3	Leave queue	On success: user ticket is erased; queue position is adjusted. On fail: user is given an error message	Leave queue test completed successfully on both scenarios.

TABLE III. SURVEY QUESTIONS

No	Question
Q1	This Application helps in creating a queue and joining one.
Q2	This Applications create a more comfortable queue experience
Q3	This application is easy to use
Q4	Viewing the current queue helps the user in a queue
Q5	This application is aesthetically pleasing

Q4 shows a 100% respondent gives positive value on the current queue feature being helpful for a user in queue. This strengthened the idea that the proposed application design is beneficial for the user in a general-purpose situation. Q5 evaluates the aesthetic of the application. While the aesthetic aspect of an application is most often subjective. Most respondents give a positive value on the design of the application. Fig. 5 shows the survey results in the form of a bar chart.

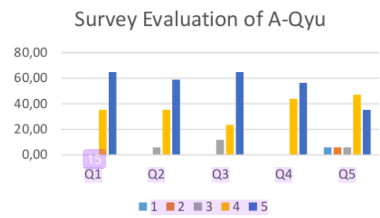

Fig. 5. Survey evaluation of A-Qyu

The survey has also undergone Cronbach's alpha coefficient measurement to indicate reliability of the survey. Cronbach's alpha ranges from 0 to 1 with 0 being that questions on the survey have no correlation with each other while 1 being perfectly correlated. Cronbach's alpha value is calculated with Equation (1).

$$\propto = \frac{N*C}{V+(N-1)*C} \quad (1)$$

With N is the number of questions on the survey, C the mean covariance between items and the item variance. To calculate this each question on the survey should be calculated for its variance and average as shown on Table IV

TABLE IV. SURVEY QUESTIONS VARIANCE

Question	Average	Variance
Q1	4,647058824	0,242647059
Q2	4,529411765	0,389705882
Q3	4,529411765	0,514705882
Q4	4,588235294	0,257352941
Q5	4	1,25

Variance analysis of the survey is shown in Table V. Cronbach's alpha value is then calculated to be of 0.7.6 for the survey which is rated reliable.

Source Variation	Sum of Square	Degree of freedom	Mean of Sum	F
Rows	21,9058823	16	1,369117647	4,26087
Columns	4,63529411	4	1,158823529	3,60641
Error	20,5647058	64	0,321323529	
Total	47,105882	84		

By considering that user acceptance (survey result with a value of 4 and 5) is above 70% on each survey question with its written response, it is then concluded that A-Qyu QMS may help users to create and join a queue within a general-purpose situation. Further development of this QMS could be to add the ability to stream queue lists to other devices. And to analyze queue for improvement. Different Queueing method could also be explored to create better general purpose QMS.

V. Conclusion

A-Qyu QMS provides general purpose uses with added advantages for users that they may choose to leave and return without missing the queue. It gives users more freedom instead of always waiting in line. Survey results also showed that the application meets user needs such as being a comfortable and easy-to-use queuing experience.

Although A-Qyu is applicable for almost every general queue situation, being a mobile solution, the system is unusable without a mobile phone. Should such a need arise the user may still revert to traditional approaches. Still this approach minimizes the use of additional hardware to queue.

Acknowledgment

We would like to thank Universitas Kristen Maranatha which has contributed to this research and to the institution that has sponsored this research. The author gratefully acknowledges Universitas Kristen Maranatha for providing financial support for this study.

References

[1] M. Ghazal, R. Hamouda, and S. Ali, "An IoT smart queue management system with real-time queue tracking, "*Proceedings - 2015 5th International Conference on e-Learning, ECONF 2015*, pp. 257–262, 2016.

[2] A. Z. Jidin, N. M. Yusof, and T. Sutikno, "Arduino Based Paperless Queue Management System," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 14, no. 3, pp. 839–845, 2016.

[3] S. N. H. Binti MD Yusof, "Development of Queue Line System for Washing Machine Using Android," Universiti Teknikal Malaysia Melaka (UTeM), 2019.

[4] C.-Y. Lin, G.-Y. Jhu, and S. Siao, "An Innovative Queue Management System Based on Bluetooth Beacon Technology," in *2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, 2019, pp. 427–430.

[5] F. A. Maulana, H. P. B. A. Nugroho, and H. Santoso, "RUKITON — Smart queue reservation system for medical center," in *2017 IEEE 8th Control and System Graduate Research Colloquium (ICSGRC)*, 2017, pp. 159–163.

[6] M. Abusair, M. Sharaf, T. Hamad, R. Dahman, and S. Abuodeh, "An Approach for Queue Management Systems of Non-Critical Services," *2021 7th International Conference on Information Management, ICIM 2021*, pp. 167–171, 2021.

[7] S. Soman, S. Rai, P. Ranjan, A. S. Cheema, and P. K. Srivastava, "Mobile-Augmented Smart Queue Management System for Hospitals," in *2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS)*, 2020, pp. 421–426.

[8] A. S. bin Ahmad Zulfikri and S. O. bin Abd Rahman, "Ticketless Queue System (TiQS) for Pusat Kesihatan UTM on Android Mobile Application," in *Innovations in Computing Technology and Applications*, 2018, pp. 1–4.

[9] G. S. Aquino *et al.*, "Queuewe: An IoT-based solution for queue monitoring," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, 2017, pp. 232–246.

[10] A. L. Aizan, A. Z. Mukhtar, K. A. A. Bashah, N. L. Ahmad, and M. K. A. M. Ali, "'Walk-Away' Queue Management System Using MySQL and Secure Mobile Application," *Journal of Electrical Power and Electronic Systems*, vol. 1, no. 1, pp. 1–4, 2019.

[11] M. Ngorsed and P. Suesaowaluk, "Hospital service queue management system with wireless approach," *Lecture Notes in Electrical Engineering*, vol. 375, pp. 627–637, 2016.

[12] "Firebase Documentation." https://firebase.google.com/docs/ (accessed Apr. 03, 2023).

[13] S. Boukhary and E. Colmenares, "A Clean Approach to Flutter Development through the Flutter Clean Architecture Package," in *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2019, pp. 1115–1120.

A-Qyu General Purpose Cloud-based Queue Management

ORIGINALITY REPORT

12%	11%	10%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Telkom University Student Paper	2%
2	www.manuelgarcia.designrshub.com Internet Source	1%
3	Sumit Soman, Sudeep Rai, Priyesh Ranjan, Amarjeet Singh Cheema, Praveen K Srivastava. "Mobile-Augmented Smart Queue Management System for Hospitals", 2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS), 2020 Publication	1%
4	Tayyaba Naseer Khan, Nazir Ahmad Zafar. "Blockchain Based Formal Modelling of Patient Management in Hospital Emergency System", 2021 International Conference on Digital Futures and Transformative Technologies (ICoDT2), 2021 Publication	1%
5	asianssr.org Internet Source	1%
6	coek.info Internet Source	1%
7	www.jatit.org Internet Source	1%
8	repository.ittelkom-pwt.ac.id Internet Source	1%
9	www.mecs-press.org Internet Source	1%

10	Submitted to Politeknik Kesehatan Kemenkes Surabaya Student Paper	<1%
11	export.arxiv.org Internet Source	<1%
12	ph01.tci-thaijo.org Internet Source	<1%
13	cris.vtt.fi Internet Source	<1%
14	www.researchgate.net Internet Source	<1%
15	www.coursehero.com Internet Source	<1%
16	Submitted to Universiti Teknologi Petronas Student Paper	<1%
17	"Proceedings of the 1st International Conference on Electronics, Biomedical Engineering, and Health Informatics", Springer Science and Business Media LLC, 2021 Publication	<1%
18	e-journal.usd.ac.id Internet Source	<1%
19	Hapnes Toba, Mewati Ayub, Maresha Caroline Wijanto, Roy Parsaoran, Ariyanto Sani. "Groups Allocation Based on Sentiment-Epistemic Analysis in Online Learning Environment", 2021 International Conference on Data and Software Engineering (ICoDSE), 2021 Publication	<1%
20	digitalcollection.utem.edu.my Internet Source	<1%
	scholarworks.waldenu.edu	

21	Internet Source	<1%
22	Dinithi Mudalige, Dylan X. Guan, Clive Ballard, Byron Creese et al. "Early life factors and dementia risk: A study of adverse childhood experiences and later-life cognition and behaviour", Public Health, 2025 Publication	<1%
23	cpee2021.zcu.cz Internet Source	<1%
24	"Frontier Computing", Springer Science and Business Media LLC, 2016 Publication	<1%
25	Edward S. Blurock. "JThermodynamicsCloud: Cloud-based tool for Temperature Dependent Thermodynamic Calculations using Group Additivity Rules and Data Management", Case Studies in Chemical and Environmental Engineering, 2024 Publication	<1%
26	online-journals.org Internet Source	<1%
27	utpedia.utp.edu.my Internet Source	<1%
28	"Inventive Computation and Information Technologies", Springer Science and Business Media LLC, 2022 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

A-Qyu General Purpose Cloud-based Queue Management

GRADEMARK REPORT

FINAL GRADE	GENERAL COMMENTS
/0	

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6