



Fair Queue-Based Resource Allocation for Community Sharing Platforms

Santhiya R¹, DR .M . Hemalatha²

¹ Pg & Research Department of Computer Science, Sri Ramakrishna College Of Arts & Science

² Associate Professor, Pg & Research Department of Computer Science, Sri Ramakrishna College Of Arts & Science, Coimbatore, India.

Article Info

Article History:

Published: 28 Feb 2026

Publication Issue:

Volume 3, Issue 2
February-2026

Page Number:

494-498

Corresponding Author:

Santhiya R

Abstract:

Community sharing platforms are common for redistributing excess resources like food, clothes, and home items to people within a given locality. While the platforms are useful in waste reduction and resource accessibility, some platforms encounter challenges when several people within the community request a particular resource at the same time. In such cases, the donors have to select the recipients manually, which can be associated with confusion and injustices. In an unstructured environment, the donor can rely on the order of messages or their own discretion when selecting a receiver. This can lead to a situation where a few people receive resources regularly while others hardly get any. Eventually, people will lack confidence in the platform, and the number of users will go down. As such, fairness becomes a critical component in maintaining a resource-sharing application. To address this problem, a queue-based allocation system is therefore proposed. The system keeps track of the incoming requests in a sequential manner and displays them to the donor in an organized manner. After the donor approves the receiver, the system automatically changes the status and does not allow other people to request the same item. The proposed system is developed through the use of a mobile app that is linked to a real-time database. The results have shown that organized allocation helps in reducing conflicts and improving transparency. To resolve this problem, a resource allocation system using a queue mechanism is proposed. The system keeps track of the requests in a sequential manner and displays them to the donors in an organized list. Once a receiver is accepted, the system automatically changes the status and does not allow repeated claims for the resource, which will be allocated to only one user.

Keywords: Resource Allocation, Queue Mechanism, Community Sharing, Fair Distribution, Mobile Application

1. INTRODUCTION

In recent years, community-based digital platforms for sharing resources have been found to be an efficient means of dealing with the wastage of useful resources. Many families tend to dispose of excess food, clothing, books, and household items that are still in good condition. This is often done not because the items are not useful, but because there is no proper means of distributing them. At the same time, many people face difficulties in obtaining necessary resources due to financial constraints. This creates a gap between the availability of resources and the lack of access to them. Mobile applications have made it easier for the interaction between the donors and the receivers to take place through instant communication and accessibility. With the use of digital platforms, users are able to upload information about the items, search for available resources, and send requests. Despite the advancement brought about by technology, many sharing systems continue to face challenges in terms of fairness

when it comes to multiple users requesting the same item. In most instances, the donors receive multiple requests and have to manually compare the times of the requests to determine whom to choose. This has led to confusion and perceived unfairness, especially when the order of the requests is not visible.

Lack of structure in handling requests may lead to dissatisfaction among users who are unable to access resources on multiple occasions. If some users are consistently allocated items while others succeed only occasionally, trust in the system is gradually lost. Users may withdraw from the system, and donors may also be reluctant to continue donating items.

The concept of queue-based processing is well-recognized in computer systems involving a series of operations. The application of this concept to community sharing platforms can ensure that each request is documented and processed in a systematic manner. The use of a queue system can enable donors to view and choose the receivers of their donations in a clear manner. This research aims to develop a fair queue-based allocation system that enhances clarity and minimizes conflicts in digital sharing platforms.

2. Objective of the Study

The main aim of this project is to develop a fair allocation system that can process multiple requests for the same resource in an organized way. In most community sharing systems, users compete for limited resources, and the lack of an organized selection process causes confusion and dissatisfaction among users. The system will thus keep a proper record of all requests and display them in a clear way that makes it easy for users to decide. The system also ensures that all users are given an equal chance through the use of a queue. Another significant objective of this study is to avoid the distribution of the same item to multiple users, thus creating unnecessary conflicts. If multiple users request the same item, only one user should be able to take the item after approval, while others are automatically notified of the outcome. This will avoid repeated communication and avoid conflicts between item donors and receivers. The system will also maintain a record of transactions so that the history of item allocation can be traced if necessary. The study also aims to enhance transparency and trust among the community platform users. Users should be able to know the status of their requests without having to wait for confirmation messages. This will encourage users to continue contributing to the platform since they will feel that the system is fair. Transparency will also encourage donors to donate more items since they will be able to see the organized distribution of items among the users. Finally, the objective includes the usability of the application. A predictable and simple process enables users of different age groups and technical knowledge to use the platform comfortably. If the allocation process is easy to understand, the platform will be reliable for everyday community use and will enable the sustainability of resource sharing activities.

3. Methodology

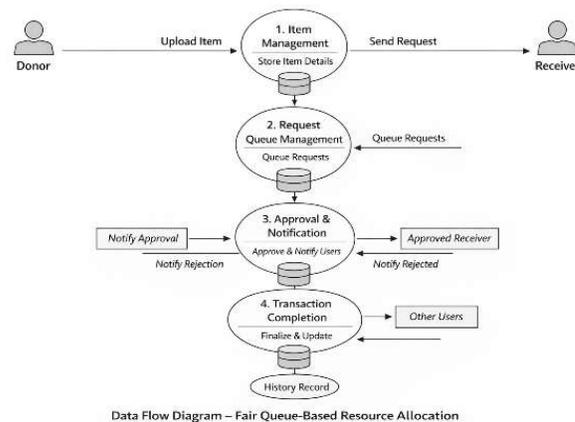
The process starts when the donor uploads an item into the application along with basic information such as description and category. The system gives a unique identifier to the item so that all related requests can be handled correctly. Once the item is visible to other users, interested users can make a request using the application

interface. All requests are registered immediately along with the time of registration and user details. After receiving several requests, the system automatically arranges them in an ordered form to create a queue. This helps the donor view all interested participants in an equal and unbiased way. Rather than comparing messages, the donor only looks at the list and chooses suitable participant. The queue system thus serves as a decision-support system that makes the allocation process easier. Once the receiver is approved, the system will update the database and automatically close other pending requests for the same item. All users will get a real-time notification about the status of their request, eliminating any ambiguity and the need to make repeated inquiries. The approved user will then arrange for the item pickup, while the other users are aware of the response without any ambiguity. Once the transaction is completed, the status of the item will be updated to completed, and the transaction will be recorded for future reference. The database will ensure that all devices are in sync, and all users will be able to view the same updated information.

4. System Design and How It Works

The application works through a client-server model where the mobile interface communicates with a central database. The client part of the application enables the user to upload products and make requests, while the server part handles storage and synchronization of the data. This model ensures that all users are updated with the latest information at the same time, irrespective of their location in terms of devices. The central database is always tracking changes and instantly updating the status of the request to all connected devices. This instant synchronization eliminates delays and ensures that all users access information on the availability of products at all times. When a product is donated, it is reflected in the list of available products for the community. Users who are interested in the product make requests that are instantly added to the list. The donor is able to see the list along with the order of the requests and chooses the recipient based on this. Since the information is organized, the donor does not have to depend on communication outside the application to establish the order of the requests. Once the receiver is selected, the system changes the status and notifies all participants. The accepted receiver communicates with the donor through the platform to set up collection information. The other users are notified that the item is no longer available, preventing confusion and unnecessary messages. This automatic update system eliminates confusion and ensures all participants are aware of the current status of the transaction. It also boosts user confidence since the status of all decisions is visible in the application. After the collection is confirmed, the item is removed from the list and marked as distributed in the database. This system eliminates the possibility of double distribution and ensures accuracy in records. The system thus promotes transparency, reliability, and ease of use in the community sharing system. The system also helps in monitoring distribution trends and dispute resolution in case a dispute arises. The organized system ensures smooth operation even when several users interact at the same time.

5. Data Flow Diagram



6. Testing and Results

The system was also tested by creating a situation where several users requested the same resource at the same time. The queue maintained the order of requests and provided the same information to all users. This proved that the synchronization process was functioning properly. Even when users were accessing the application simultaneously, the database maintained the correct order without any issues. The smooth operation of the system proved that it was ready for use in a real-life setting. During the approval process, the system updated the status immediately and closed other requests automatically. Users were notified instantly so that they could understand the status clearly. This reduced communication between users and also prevented misunderstandings regarding the availability of resources. The system also ensured that users did not wait unnecessarily for a response. This made the interaction between the donor and the receivers more efficient and organized. The donors found it easier to select the receivers since the requests were arranged in a logical manner. The donors did not have to compare the timestamps of the messages; instead, they could select from the list of messages. The donors also felt more comfortable sharing their items since they believed that the system was fair. This shows that usability has a direct effect on the levels of participation. The results showed that the queue-based system was able to eliminate duplicate distribution and that the overall user satisfaction was improved. The users felt that they received fairer allocation than when the system was used manually. This shows that the system is able to improve the usability of the community sharing system. The system is also able to eliminate unnecessary messaging traffic, making communication clearer and faster.

7. Conclusion

The fair queue-based allocation mechanism is an effective way of dealing with competing requests in community sharing applications. The mechanism ensures that the requests are visible, thus avoiding confusion. The mechanism also makes it easy to choose the receivers. This makes it efficient in the allocation of resources. The mechanism also ensures that the interaction between the donors and the receivers is smooth, even when there are

many requests. The automatic status updates enhance transparency. This is because the participants are able to understand the allocation process clearly. When users trust the system, they are more likely to donate and request resources. This enhances the sense of cooperation in the community. The transparency also reduces conflicts, as all the processes are clearly documented. The mechanism also provides accountability through the retention of transaction records and the avoidance of double claims. This organized form of management allows the platforms to run efficiently even with growing levels of user engagement. The mechanism thus plays a role in ensuring long-term reliability and consistency in usage. It also promotes prudent use of resources in the communities. Future enhancements may involve priority-based selection and recommendation algorithms to further improve the efficiency of allocation. The proposed method can also be used in other collaborative spaces where fair allocation is necessary. Further development of the system with new features can also improve community engagement and resource sharing.

References

1. Google Developers, Flutter Documentation Accessible: <https://flutter.dev/docs>
2. Google Developers, Firebase Documentation – Authentication, Firestore and Cloud Messaging. Accessible: <https://firebase.google.com/docs>
3. Android Developers, Android Application Development Guide. Accessible: <https://developer.android.com/guide>
4. I. Sommerville, Software Engineering, 10th ed., Pearson Education, 2016.
5. R. S. Pressman and B. Maxim, Software Engineering: A Practitioner’s Approach, 8th ed., McGraw-Hill Education, 2015.
6. M. Weiser, “The Computer for the 21st Century,” Scientific American, vol. 265, no. 3, pp. 94-104, 1991.
7. A. Pathak and P. Shah, “Mobile Applications for Social Good: Resource Sharing Platforms,” International Journal of Computer Applications, vol. 182, no. 31, pp. 20-25, 2018.