

Qaring : A Web-Based Charity System for Orphanage Donation

Muhammad Arif Mu'az Mohammad Nor Zamri¹, Firkhan Ali Hamid Ali^{1*}

¹ Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: firkhan1977@yahoo.com

DOI: <https://doi.org/10.30880/aitcs.2026.07.01.018>

Article Info

Received: 12 June 2025

Accepted: 18 June 2026

Available online: 30 June 2026

Keywords

Donation, Track Donation ,Agile,
Donation Tracking , Web-Based
Application, User Role Management

Abstract

Currently, donations to orphanages are often managed manually or through fragmented platforms, leading to inefficiencies, mismatched aid, and limited donor-orphanage engagement. The main objective of this project is to develop QARING, a web-based charity system designed to streamline and manage donations to orphanages efficiently. The system enables donors to contribute monetary or item-based donations while allowing orphanages to post their needs transparently. The proposed system adopts the Waterfall model, comprising five structured phases: planning, analysis, design, implementation, and testing. Key features include user role management (Admin, Donor, Orphanage), donation tracking, orphan and needs management, and location-based matching of donors to nearby orphanages. The findings suggest that the system can enhance transparency, increase successful donation matches, and reduce the burden on orphanage staff. The structured data collected through this platform will be integrated into its modules to further improve donation efficiency and strengthen community support.

1. Introduction

Orphans and vulnerable children face significant challenges due to lack of parental care and resource constraints [1]. Globally, there are an estimated 153 million orphans (children who have lost one or both parents) according to UNICEF [2]. Many of these children are placed in institutional care; a 2017 UNICEF report estimated at least 2.7 million children living in orphanages and residential care worldwide [3].

Donors who wish to contribute often lack a direct platform to connect with orphanages, resulting in untapped goodwill and unmet needs. Project QARING (Quality Orphanage Donor-Caring System) was conceived to leverage technology in easing orphanage management and facilitating public donations. The system aims to streamline contributions of funds and supplies, inspired by previous charity management systems that demonstrated digital platforms can significantly improve record-keeping and donor engagement [3].

The primary challenges addressed by QARING include inefficient manual processes, limited donor reach, and a lack of an integrated system combining donor management, orphanage management, and donation tracking. QARING addresses these issues by digitalizing records and creating an online marketplace for charity, bridging donors and orphanages in real-time. The scope of QARING encompasses three user roles Donor, Orphanage, and Admin, with modules including Donor Management, Orphanage Management, Donation Tracking, Administration, Notifications, and Reports. The implementation is web-based without native mobile app development or financial payment gateway integration.

Table 1 gives an overview of the Qaring project, including the main features that are being worked on (in-scope) and those that are not (out-of-scope).

Table 1 *Project Scope of Qaring*

User	Module	Function
Donor	Authentication	Register and log in securely to the system
	Dashboard & Profile	View donation history, update personal information
	Orphanage Browsing	Search and view orphanages, filter by needs or location
	Donation Management	Make donations (monetary/in-kind), track donation status
Orphanage	Authentication	Register orphanage account, login securely
	Orphan Management	Add, edit, and manage orphan profiles
	Needs Posting	Post and update donation requests
	Donation Tracking	View and confirm received donations
Administrator	User Management	Approve orphanage accounts, manage all users
	System Configuration	Monitor system usage, manage categories or need types
	Reporting Module	Generate system usage, donation, and activity reports
All Users	Notification Module	Receive real-time alerts for donations, approvals, status changes
	Search & Filter	Search needs or users, filter by categories and regions

2. Related Work

2.1 Technology Used

QARING is developed using PHP, MySQL, HTML, JavaScript, and Bootstrap CSS. PHP, a widely used server-side scripting language, powers approximately 77% of websites globally [8], offering robust database integration capabilities and rapid application development features. MySQL, a reliable and widely adopted relational database management system, ensures efficient data storage and retrieval, essential for managing donor information, orphanage details, and donation records. HTML and JavaScript provide the foundational structure and interactive capabilities necessary for user interfaces, while Bootstrap CSS, a leading framework, ensures responsive and visually consistent design across various devices, thereby enhancing user experience and accessibility [9]. The selected technology stack aligns with those employed in similar successful charity management systems [3].

2.2 Study Of Existing System

Several systems exist to support orphanage management and donations. The Digitalized Orphanage Home Management System developed by Ewald et al. focuses primarily on managing orphan data digitally, replacing paper-based methods with computer systems to ensure better data integrity and easier data management [4]. However, this system primarily addresses internal record-keeping without external donor interaction. The Charity Connecting System developed by Archana and Mouthami, enables donors to contribute to orphanages but uses a centralized donation distribution approach without specific orphan or orphanage visibility for donors [5]. Additionally, Orphanage Finder Applications developed by Muthuselvan et al. [6] and Indhu et al. [7] leverage location-based technologies to help users find nearby orphanages but do not provide extensive donation management features or detailed orphan profiles. These systems individually address important aspects of orphanage management but fail to offer a comprehensive, integrated platform that combines orphan profile management, direct donor-to-orphanage interactions, and detailed donation tracking.

Table 2 *Comparison of Existing Solutions and QARING*

System	Digitalized Orphanage System	Charity Connecting System	Orphanage Finder App	QARING
Module				
Orphan Profile Management	Yes	Limited	No	Yes
Donor Registration and Login	No	Yes	No	Yes

Table 2 <i>Comparison of Existing Solutions and QARING(cont.)</i>				
Monetary Donation Handling	No	Yes	No	Yes

In-kind Donation Tracking	No	Possible	No	Yes
Orphanage Account/Portal	Yes	Yes	No	Yes
Reporting and Analysis	Basic	Limited	No	Yes

3. Methodology

In this project, the Agile methodology was adopted for the development of the QARING system, a web-based platform designed to streamline the donation process between donors and orphanages. Agile was selected for its flexibility, iterative workflow, and capacity to adapt to changing requirements based on user feedback. This methodology promotes continuous improvement, collaborative development, and incremental delivery. It's ideal for a system like QARING, which needed to evolve based on real-time input from orphanage staff and administrative users. The Agile process was organized into a series of short development cycles or sprints, with each sprint focused on delivering a functional module such as user registration, donation tracking, or need posting. This iterative approach allowed the team to gather feedback early and often, ensuring that the system remained aligned with stakeholder needs and expectations throughout the development lifecycle. Key phases in the Agile process included requirements gathering, design, development, testing, deployment, and review which each contributing to the delivery of a user-centered and adaptable solution.

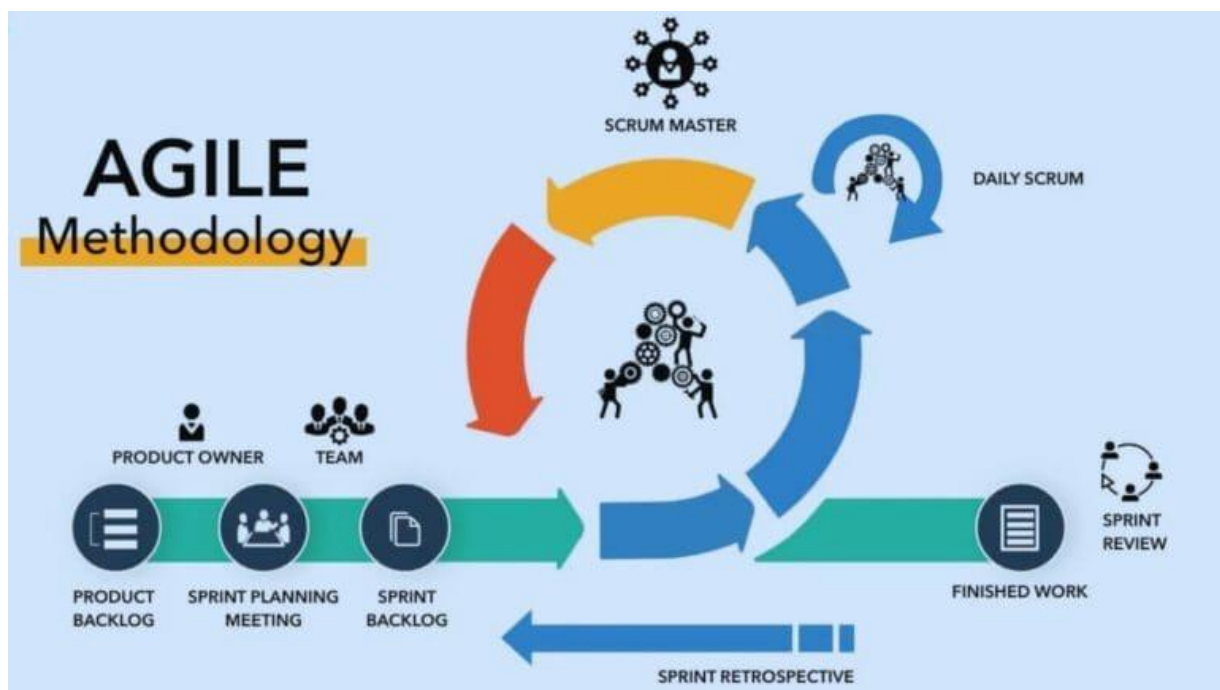


Fig 1 Agile Model[11]

The development of QARING began with a planning phase focused on gathering requirements in collaboration with key stakeholders, particularly orphanage staff. Through direct discussions and interviews, the team identified the core needs of three primary user roles which are Donor, Orphanage, and Admin. Based on these insights, user stories were created, which served as the foundation for the product backlog. This backlog included high-priority features such as donor registration, need posting, donation tracking, and administrative reporting tools.

Once the requirements were established, the design phase commenced at the start of each sprint. User interfaces and system workflows were visualized through wireframes and flowcharts, created using Canva and Bootstrap. The design process emphasized a clean and responsive layout to ensure accessibility across different devices. Early prototypes were shared with stakeholders to gather quick feedback, allowing for iterative improvements before the development phase began.

Development was carried out in modular sprints, each targeting a specific functional area. Sprint 1 focused on user registration and login while sprint 2 involved building the orphan profile management module and lastly sprint 3 concentrated on donation management and notification systems. The development team used PHP for server-side logic and MySQL for database management, while version control was maintained using GitHub. This modular and collaborative approach allowed for parallel progress and efficient issue tracking.

Testing was an integral part of each sprint, conducted immediately after development tasks were completed. The team implemented unit testing, integration testing, and form validation checks to ensure system stability. Orphanage staff participated in user acceptance testing (UAT), performing real-life actions like adding orphan records and submitting donations. Manual testing in different browsers and devices helped confirm responsiveness and usability across platforms.

Upon passing internal testing, the system was deployed to a staging environment hosted on a test server using services like Hostinger. This allowed real users to interact with the system in a near-final setting. The deployed version included a public donor view, orphanage dashboard, and admin panel. This stage ensured that all components were integrated correctly and ready for user-facing feedback.

The review process was crucial to refining the system. Feedback was collected through Google Forms and follow-up interviews with orphanage staff. Based on this input, the development team made several enhancements, such as improving orphanage search functionality, adding confirmation prompts, and introducing summary dashboards. Each sprint concluded with a retrospective session to identify what went well and what could be improved, ensuring that future sprints were even more aligned with user expectations, as shown in the Agile process [11].

3.1 System Development Process

QARING was developed using the Agile methodology to promote flexibility and iterative progress. Agile emphasizes collaboration, rapid prototyping, and continuous feedback from users. The development process was broken into sprints, each focusing on delivering a specific module such as user registration, donation tracking, or orphanage management. At the end of each sprint, a review session was conducted with stakeholders, including potential users like donors and orphanage staff, to gather feedback and plan the next sprint. This iterative approach ensured that the system was developed incrementally, allowing refinements to be made based on user experience and actual needs. Key Agile ceremonies such as daily stand-ups, sprint planning, and retrospectives were practiced informally among the development team to maintain progress and adaptability. The flexibility of Agile enabled the incorporation of real-time feedback, improving usability and relevance of features such as donation notifications, orphan profile updates, and administrative reports.

Table 3 *Agile Development Phases and Activities for QARING.*

Sprint/Phase (Duration)	Key Activities	Deliverables
Sprint 1: Planning and Setup (2 weeks)	Identify stakeholders, define requirements, build product backlog	User stories, prioritized backlog, scope plan
Sprint 2: Analysis and UI Design (2 weeks)	Develop wireframes, UI mockups, system diagrams (ERD, DFD, context diagram)	UI prototypes, ERD, DFD, context diagram
Sprint 3: Backend & Database Setup (2 weeks)	Create MySQL schema, configure database connections, build core backend logic	Functional database, basic backend structure
Sprint 4: Frontend & Integration (2 weeks)	Develop front-end using Bootstrap, integrate frontend and backend modules	Responsive web pages, working core functions
Sprint 5: Testing & Refinement (2 weeks)	Conduct unit testing, fix bugs, run full system tests	Test cases, test results, debugged system
Sprint 6: Deployment & UAT (1 week)	Deploy system on test server, collect user feedback via UAT surveys	Deployed demo, UAT response data
Future Sprint: Maintenance & Scaling (ongoing)	Post-launch enhancements, bug fixes, possible mobile version, volunteer module	Maintenance plan, feature backlog

3.2 Website Development Workflow

This section talks about how the development tasks were done, including the tools and workflow. The last section talked about what was done in each step. The way QARING was made used both normal website development methods and cloud-based cooperation.

Table 4 *Development Workflow and Tools for QARING.*

Development Stage	Workflow and Tools Used
1)UI/UX Design	Used Canva to design wireframes and interface mockups. Layouts were based on Bootstrap components for responsive design. Shared drafts with users for feedback.
2)Database Design	Designed using MySQL Workbench and managed through phpMyAdmin. Created ERD and normalized tables up to 3NF to ensure relational integrity and scalability.
3)Role-Based Access Control	Implemented session-based access in PHP using \$_SESSION. Defined user permissions for Donor, Orphanage, and Admin roles across all pages.
4)Coding & Implementation	Used Visual Studio Code for development of PHP backend and Bootstrap frontend. Version control managed through GitHub using branches for each module.
5)Form Validation & Security	Client-side validation using JavaScript and Bootstrap classes. Backend validation done using prepared statements in PHP. Passwords hashed with password_hash().
6)Testing	Conducted testing on XAMPP (local server). Manually ran unit, form, and integration tests with test user data. Debugged modules iteratively.
7)Deployment	Deployed using Hostinger shared hosting. Used File Manager and FileZilla FTP to upload files. phpMyAdmin used to import the database. Configured .htaccess for clean URLs.
8)UAT & Feedback Loop	Created Google Forms for Donor, Orphanage, and Admin users. Collected feedback on functionality, usability, and UI. Refined interface and fixed UX issues accordingly.

4. System Analysis

4.1 System Requirement Analysis

During the planning stage, detailed functional requirements were set to make sure that QARING could meet the needs of its users. These Functional Requirements spell out what the system should do and how it should act. Table 5 shows the main functional requirements for QARING. Each one is linked to one or more features or use cases on the website.

Table 5 *Functional Requirements of QARING*

Function Name	Functional Requirement Description
1) User Registration & Login	Allow secure creation of donor, orphanage, and admin accounts
2) View & Search Orphanage Needs	Donors browse orphanages, search needs by keyword or location
3) Donation Processing	Enable donors to submit donation pledges to orphanages
4) Donation Record Management	System records and tracks donations, notifies recipients
5) Orphanage Profile Management	Orphanage staff can manage detailed profiles of orphans
6) Posting Donation Needs	Orphanage users post specific donation requirements

Table 5 *Functional Requirements of QARING (cont.).*

7) Administrative Oversight	Admin manages users, approves orphanages, generates reports
8) Notification System	Notify users upon relevant actions (donations, registrations)
9) Access Control	Enforce secure access based on user roles and permissions

4.2 Non-Functional Requirements

Table 6 *Functional Requirements of QARING.*

Attribute	Non-Functional Requirement Description
Usability	The system should have a clean, intuitive user interface designed for users with minimal technical background. Navigation should be simple, and feedback messages should be clear.
Performance	Core pages (e.g., dashboards, orphanage list, donation form) must load within 3 seconds under normal network conditions. The system should support up to 50 simultaneous users.
Security	All sensitive data must be protected. Passwords are hashed using <code>password_hash()</code> in PHP. Inputs must be validated to prevent SQL injection, XSS, and session hijacking.
Compatibility	The system must work consistently across major web browsers (Chrome, Firefox, Edge, Safari). Bootstrap ensures responsive layout across desktops, tablets, and mobile devices.
Reliability	The system should operate with minimal downtime. Data backups should be scheduled daily to avoid data loss. All critical operations (e.g., donations) must be transaction safe.
Maintainability	The codebase should follow modular design principles. PHP functions should be organized by responsibility (e.g., user, donation, reporting). GitHub used for version control and tracking.
Scalability	The system should be designed to scale horizontally if needed. Additional orphanages, donations, and user roles can be added without restructuring the core database or system logic.

4.3 Context Diagram

The context diagram for the QARING system illustrates the interactions between the external users (Donor, Orphanage, and Admin) and the centralized donation management platform. Donors interact with the system to register, view orphanage needs and submit donations. Orphanages use the system to register, post their needs, manage orphans, and track donations received. Admins oversee the entire platform by managing users and monitoring all system activities, including donation records and user actions. All user interactions are directed toward a single centralized system that handles data storage, validation, and user-role-based access to ensure organized and efficient donation management. The context diagram is provided in Appendix A.

4.4 Data Flow Diagram (DFD)

The Data Flow Diagram (DFD) illustrates how data moves between key processes within the QARING system, including user management, donation tracking, and orphanage operations. Each user role interacts with specific modules and data stores, and these interactions are modeled in a structured, logical flow. The Level 1 DFD is provided in Appendix B.

4.5 Entity Relation Diagram (ERD)

The Entity Relationship Diagram (ERD) represents the underlying database design of the QARING system. It identifies the primary entities, such as User, Orphanage, Orphan, Need, and Donation, and the relationships that ensure data consistency and normalization. The full ERD diagram is included in Appendix C.

4.6 User Interface Design

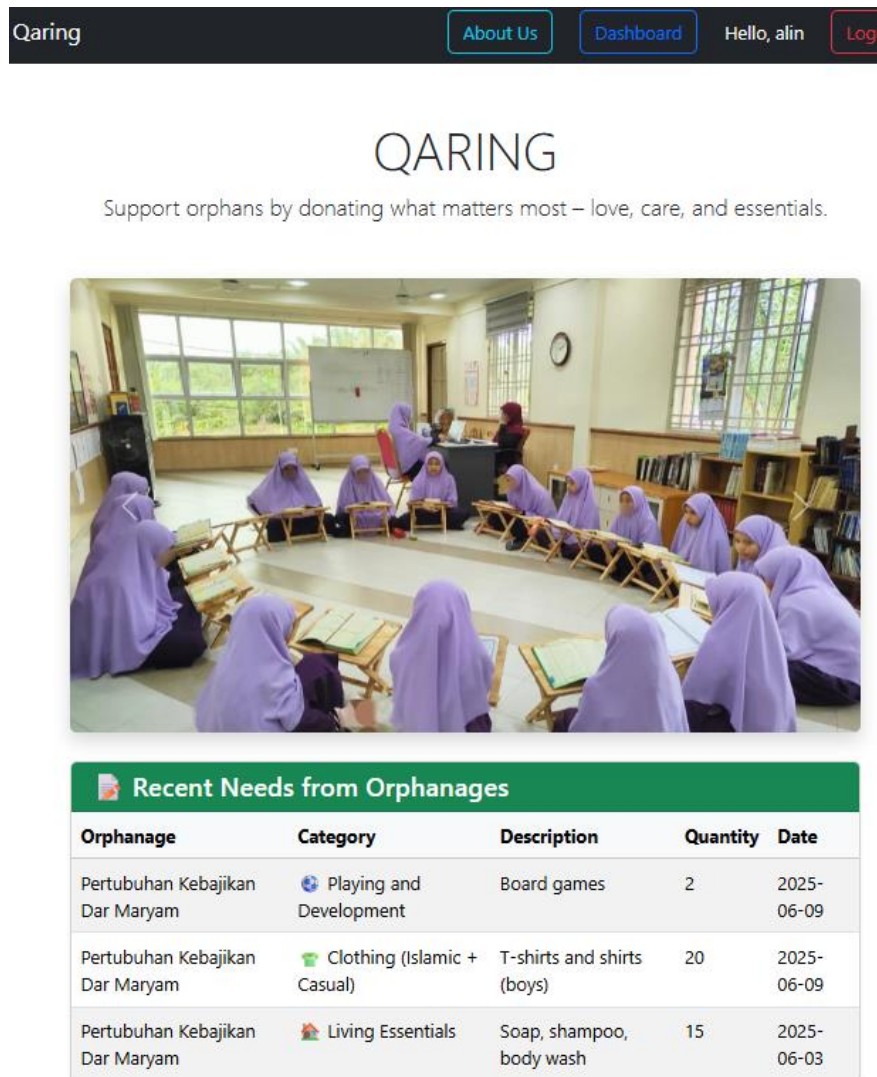


Fig 2 Donor View submitted needs by orphanage

Qaring [About Us](#) [Dashboard](#) Hello, Pertubuhan Kebajikan Dar Maryam [Logout](#)

Add Orphanage Needs

Category

Item

Quantity

[Add Need](#)

© 2025 Qaring. All rights reserved.

(a)

Qaring [About Us](#) [Dashboard](#) Hello, Pertubuhan Kebajikan Dar Maryam [Logout](#)

Your Submitted Needs

You

Playing and Development		
Item Description	Quantity	Requested On
Board games	2	2025-06-09

Clothing (Islamic + Casual)		
Item Description	Quantity	Requested On
T-shirts and shirts (boys)	20	2025-06-09

Living Essentials		
Item Description	Quantity	Requested On
Soap, shampoo, body wash	15	2025-06-03
Towels and face cloths	23	2025-06-12

© 2025 Qaring. All rights reserved.

(b)

From Fig 3 (a) Add needs (b) Orphanage view Submitted needs

5. Implementation and Testing

5.1 System Implementation

The Qaring website was implemented using a combination of Visual Studio Code for the backend development , Hostinger for the hosting.

Qaring

About Us

Login

Register

Login

Email

Password

Login

© 2025 Qaring. All rights reserved.

(a)

Qaring

About Us

Login

Register

Register

Name

Email

Password

Role

-- Select Role --

Register

© 2025 Qaring. All rights reserved.

(b)

Qaring

About Us

Dashboard

Hello, Pertubuhan Kebajikan Dar Maryam

Logout

Add Orphanage Needs

Category

Living Essentials

Item

Towels and face cloths

Quantity

23

Add Need

© 2025 Qaring. All rights reserved.

(c)

Recent Needs from Orphanages				
Orphanage	Category	Description	Quantity	Date
Pertubuhan Kebajikan Dar Maryam	Playing and Development	Board games	2	2025-06-09
Pertubuhan Kebajikan Dar Maryam	Clothing (Islamic + Casual)	T-shirts and shirts (boys)	20	2025-06-09
Pertubuhan Kebajikan Dar Maryam	Living Essentials	Soap, shampoo, body wash	15	2025-06-03

(d)

From Fig 4 (a) login, (b) Multi-role Registration, (c) add orphanage needs(orphanage) (d)view orphange needs(donor)

5.2 System Functionality

The final QARING system includes all planned core features for each user role which are Donor, Orphanage, and Administrator. We validated the system against user stories to ensure each role could complete key tasks smoothly. A donor registered, browsed needs, donated based on an orphanage posted needs, received donations, and updated statuses, and the admin reviewed user activity, approved accounts, and generated reports.

All primary interactions from registration to donation fulfillment, were tested through complete user cycles to ensure data integrity and role-based functionality. While features like the dashboard summaries and notification alerts are not core donation mechanisms, they significantly improve user experience and were confirmed to be intuitive and effective during testing.

QARING successfully connects donors and orphanages, simplifies donation workflows, and ensures transparency through structured access control and confirmation mechanisms. The next section details the testing performed to ensure all modules work together reliably to fulfill project objectives.

5.3 Functional Testing

Each use case and system requirement were translated into test cases. Table 7 shows a sample of functional test cases and their outcomes. The testing covered scenarios such as user registration and login (ensuring that validation works and accounts are created), orphanage profile creation, posting a new need, donor browsing and filtering needs, donating pledge, updating the need status when a donation is fulfilled, and admin actions like approving orphanage registrations or generating reports. All critical paths were tested. For example, a test case for the donation flow verified that when a donor submits a donation form for a specific need, the system creates a donation record linked to that donor and need, sends a confirmation notification to the orphanage, and marks the need as “Fulfilled” if the donation covers it completely. All identified functional test cases passed, indicating that the system meets its functional requirements. Minor bugs encountered during testing (such as an initial misassignment of donation IDs and a formatting issue in date display) were fixed and re-tested.

Table 7 *Test Cases*

Module	Test Case	Expected Result	Status
Authentication Module	Account creation and login	New user account is created; login successful	Pass
Donation Module	Donation form submission	Donation is recorded and notification sent	Pass
Orphan Management	Add orphan profile	Orphan data is saved and displayed properly	Pass
User Management	Approve orphanage registration	Orphanage is approved and visible to donors	Pass
Access Control Module	Role-based access enforcement	Unauthorized access is blocked or redirected	Pass

5.4 User Acceptance Testing

After internal testing, we conducted User Acceptance Testing (UAT) with sample users simulating real end-users: donors, orphanage staff, and an admin. We prepared separate UAT survey forms for each role to gather structured feedback on functionality, usability, and interface design. Each survey had three sections (Functionality, Usability, Interface Design), each with 5 statements to be rated on a Likert scale (1-Strongly Disagree to 5-Strongly Agree). Figure 5, Figure 6, and Figure 7 outlines the example questions from the UAT questionnaire for each user type.

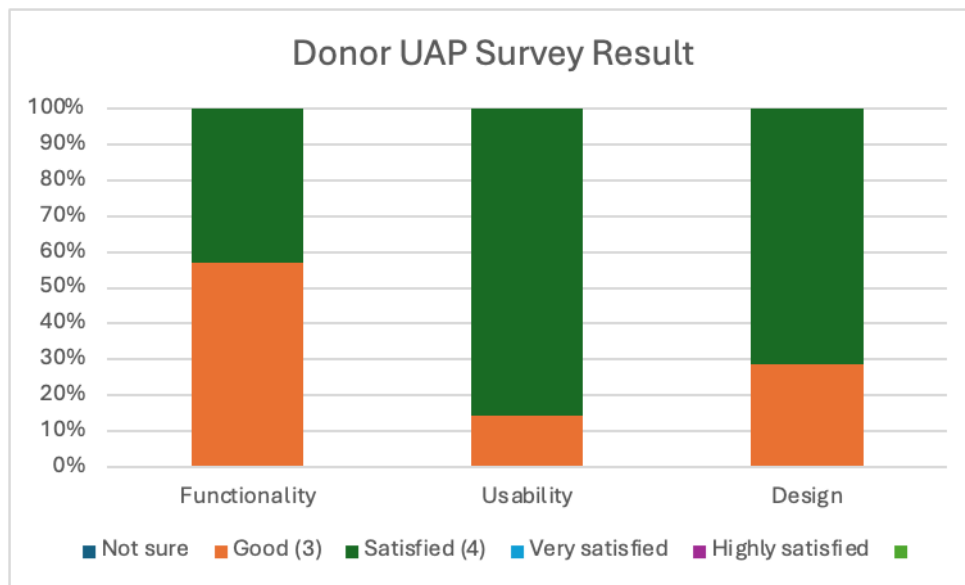


Fig 5 100% stacked column chart for Donor UAT

Donor feedback presents a slightly more varied picture. For Functionality, donors mostly rated the system as “Good,” with a significant portion indicating they were “Not sure.” This suggests that while the donation process and tracking system are generally acceptable, some donors may not have fully explored or understood all available features, or found the donation-matching system less direct. In terms of Usability, the results are highly positive—most donors found it easy to browse needs, donate items, and update their profiles, indicating a well-structured donation flow. When it comes to UI/UX Design, most donors found the website visually appealing and easy to navigate. However, a portion of users selected “Good,” suggesting that while the interface is functional and donor-friendly, there may be opportunities to make it even more engaging or emotionally resonant for the donation experience.

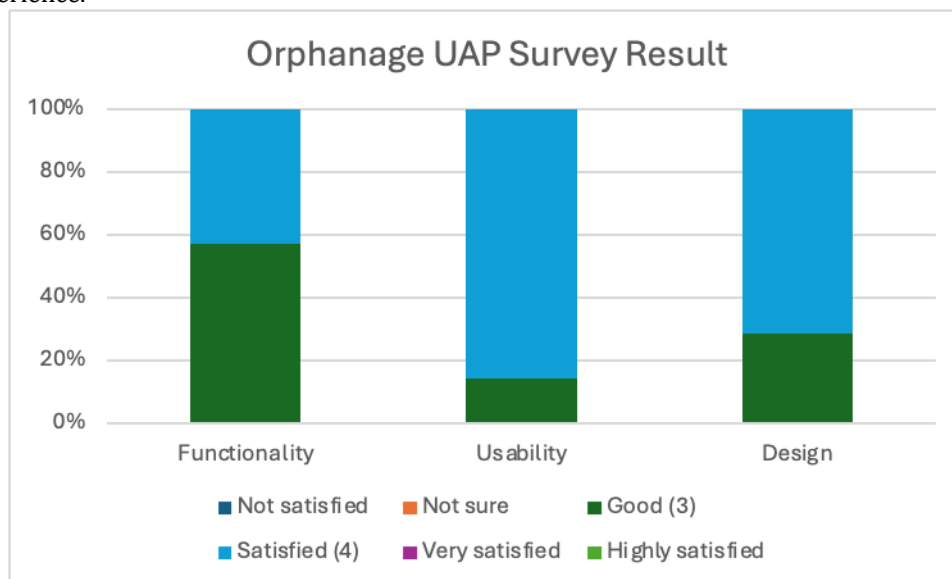


Fig. 6 100% stacked column chart for Mechanic UAT

The Orphanage user group provided consistently strong feedback across all three areas. For Functionality, most responses indicate “Satisfied,” followed by “Good,” showing that orphanage users find it effective to add orphan details, manage donation needs, and track updates. The donation-matching features seem reliable, though some users may see room for further improvements. Usability received overwhelmingly positive ratings, with nearly all respondents selecting “Satisfied.” This reflects a smooth user experience in navigating the site, understanding instructions, and managing orphanage accounts. For UI/UX Design, the results were similarly favorable, with most users finding the layout appealing and supportive of their tasks. However, the mix of “Good” and “Satisfied” indicates potential to further enhance the visual consistency and task-oriented design of the orphanage portal.

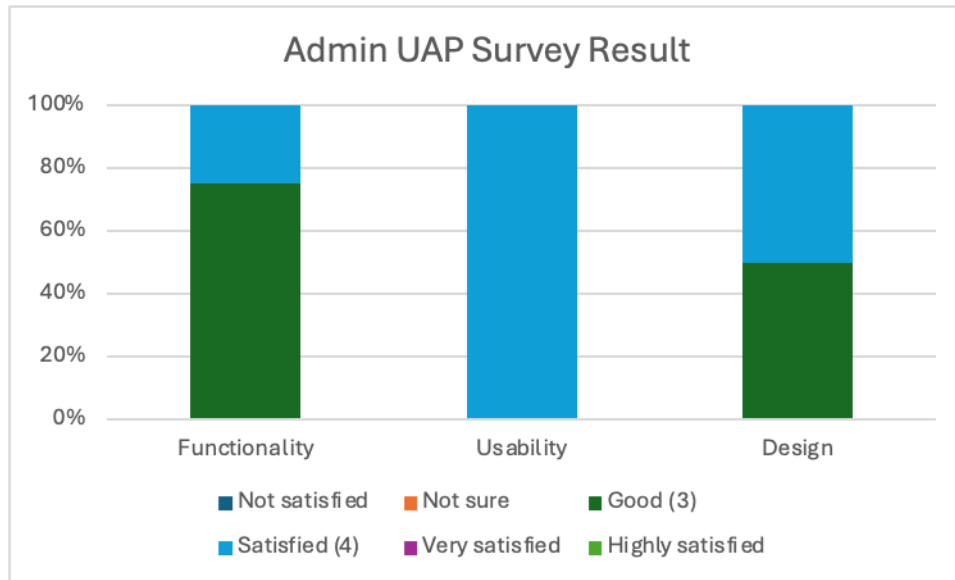


Fig. 7 100% stacked column chart for Admin UAT

The Admin UAT survey results indicate that administrators are generally satisfied with the system's core functionalities. Most responses for Functionality fall within the "Good" and "Satisfied" categories, showing that the admin users find features like user management, dashboard insights, and report handling reliable and effective, though not exceptionally advanced. For Usability, all responses reflect a high level of satisfaction, suggesting that tasks such as approving or deactivating users and performing maintenance are straightforward and intuitive. This reflects strong administrative design thinking in the platform. Regarding UI/UX Design, the responses are again mostly positive, with many users selecting "Good" or "Satisfied," implying that the interface is professional and consistent across pages, but may still benefit from visual improvements to enhance decision-making speed and overall aesthetic appeal.

6. Conclusion

This study successfully achieved its three primary objectives. First, to design the system architecture, user interface, and key functional modules of QARING, a web-based platform aimed at facilitating transparent and organized connections between donors and orphanages; second, to develop a fully responsive, role-based website using PHP, MySQL, and Bootstrap enabling donors to view and contribute to orphanage needs, allowing orphanage staff to manage orphan and donation records, and providing administrators with oversight and system integrity controls; and third, to evaluate the QARING website's performance, usability, and functionality through systematic testing and user acceptance assessments involving actual users from each stakeholder group (Donor, Orphanage, Admin).

Despite achieving its core goals, this initial implementation of QARING has several limitations, it currently lacks mobile-native support beyond responsive web design, there is no volunteer coordination or advanced search functionality, and user-to-user messaging is not supported. Additionally, the hosting on shared plans (e.g., Hostinger) may limit scalability, and user acceptance testing was limited in scope to small sample sizes (10 donors, 5 orphanages, 1 admin), which might not represent the broader population well.

To address these shortcomings, future enhancements will aim to introduce a dedicated mobile app with push notifications, incorporate volunteer coordination and advanced search filters, and implement a secure messaging module between donors and orphanages. Plans also include migrating to a cloud-based, scalable architecture to support user growth, and conducting larger-scale UAT with diverse participants to enhance generalizability and reliability of the evaluation results.

In summary, the QARING project successfully delivers a functional, tested, and user-approved web-based system that streamlines the donation process for orphanages. While some limitations remain, the foundational design, development, and evaluation provide a strong base for future expansion. With strategic upgrades—such as mobile support, additional modules, and broader testing—QARING has the potential to greatly enhance donation transparency and impact within the charitable ecosystem.

Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

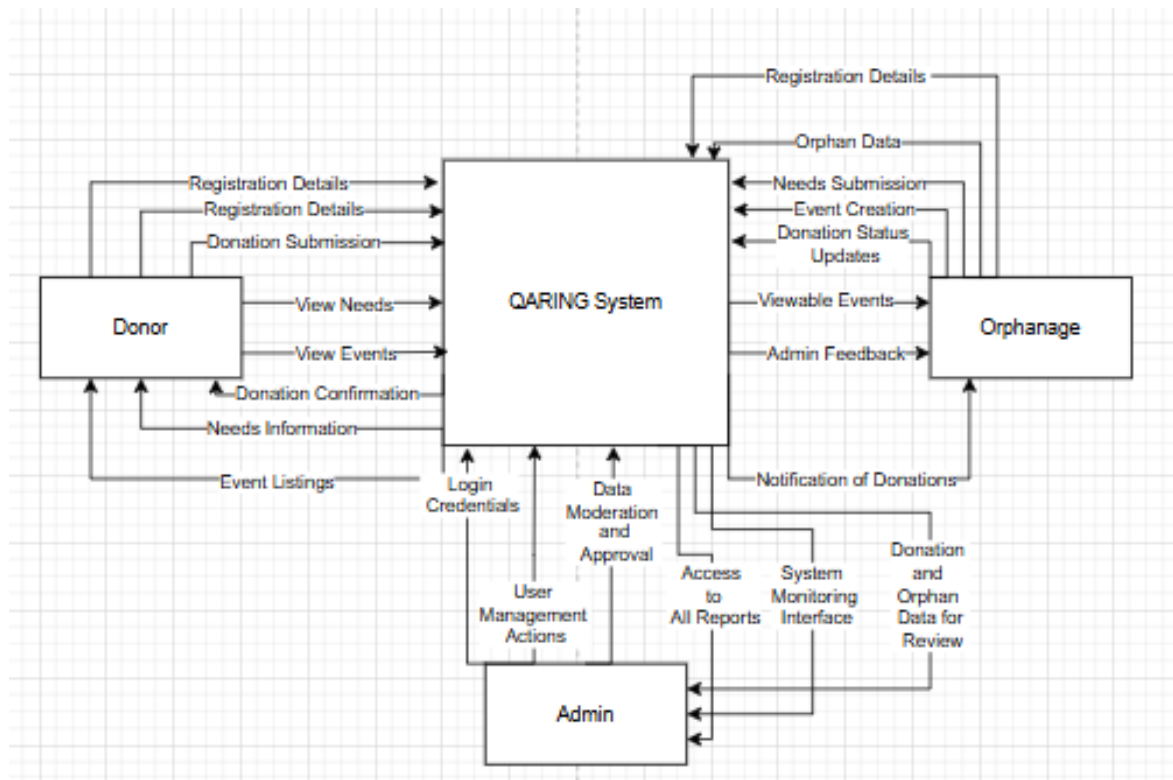
Author Contribution

The authors confirm their contributions to the paper as follows: study conception and design were carried out by Muhammad Arif Mu'az Mohammad Nor Zamri and Firkhan Ali Hamid Ali. Data collection, as well as analysis and interpretation of results, were collaboratively undertaken by Muhammad Arif Mu'az Mohammad Nor Zamri and Firkhan Ali Hamid Ali. Draft manuscript preparation was also a joint effort by Muhammad Arif Mu'az Mohammad Nor Zamri and Firkhan Ali Hamid Ali. Both authors reviewed and approved the final version of the manuscript and take full responsibility for its content.

References

- [1] K. E. R. Ewald et al., "Digitalized Orphanage Home Management System Consisting of Mass Data Entries," *Int. J. Industrial Electronics and Electrical Engineering*, vol. 8, no. 6, 2020.
- [2] SOS Children's Villages USA, "Worldwide Orphan Statistics," SOS USA, 2025. [Online]. Available: <https://www.sos-usa.org/>
- [3] UNICEF, "Data gaps on children in residential care leave the most vulnerable unaccounted for," UN Vietnam, 2017. [Online]. Available: <https://vietnam.un.org/en/7565>
- [4] R. Kaladevi et al., "Orphan Adoption Management System using Machine Learning Approach," *Int. J. Education and Management Engineering*, vol. 12, no. 5, pp. 18–23, 2022.
- [5] M. Archana and K. Mouthami, "Charity Connecting System," *IJLTEMAS*, vol. 3, no. 7, pp. 137–142, 2014.
- [6] S. Muthuselvan et al., "Location Based Orphanage Finder Application for Google Android Phones," *Int. J. Pure and Applied Mathematics*, vol. 119, no. 16, pp. 2009–2015, 2018.
- [7] Indhu K. et al., "Orphanage Finder Application," *Int. J. Computer Science Trends and Technology (IJCST)*, vol. 10, no. 1, pp. 99–103, 2022.
- [8] W3Techs, "Usage statistics of PHP for websites," W3Techs Survey, June 2025. [Online]. Available: <https://w3techs.com/technologies/details/pl-php>
- [9] Bootstrap, "The most popular HTML, CSS, and JS library in the world," GetBootstrap, ver. 5.3, 2025. [Online]. Available: <https://getbootstrap.com>
- [10] A. S. Gillis, "What is user acceptance testing (UAT)?", TechTarget, Mar. 14, 2022. [Online]. Available: <https://www.techtarget.com>
- [11] ROSEMET LLC, "Agile Methodology Definitions: Essential Guide For Teams," Apr. 08, 2025. [Online]. Available: <https://www.rosemet.com>

Appendix A: Context Diagram



Appendix B: Data Flow Diagram

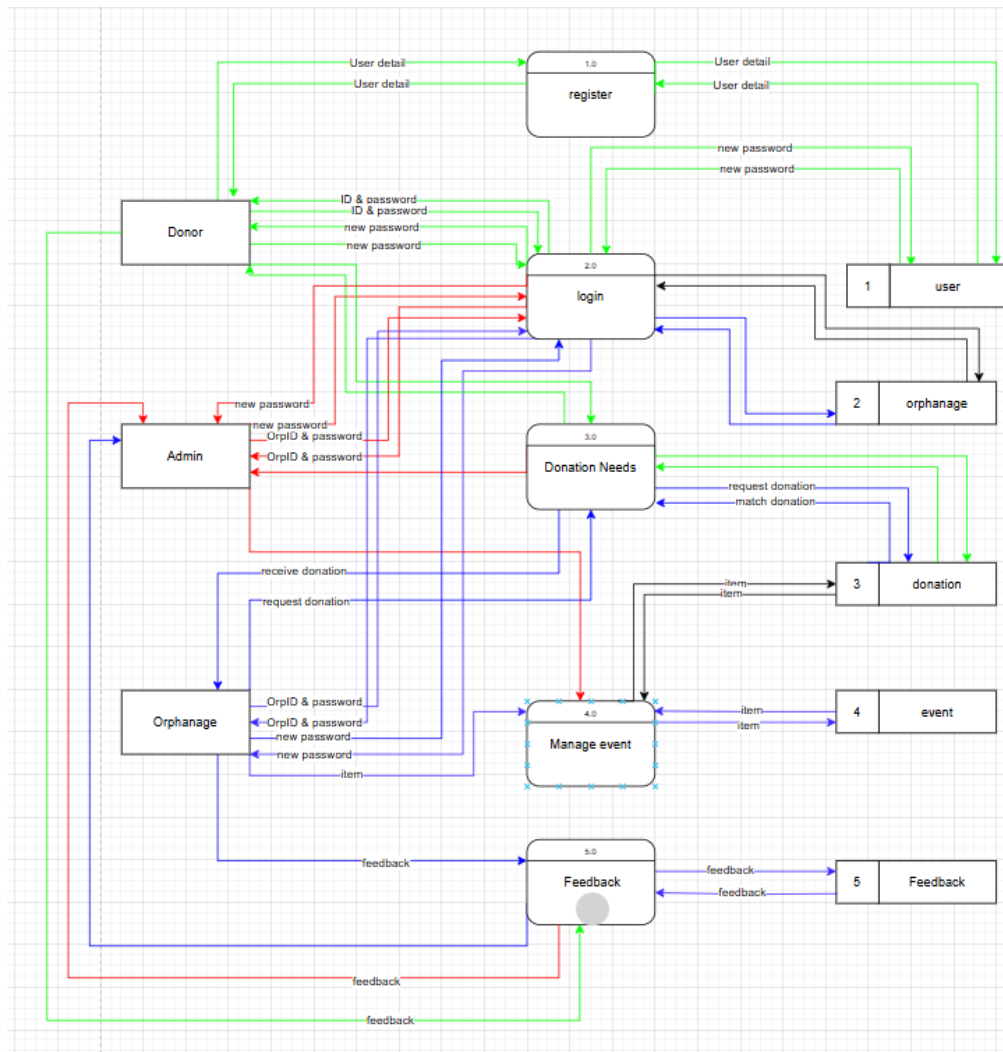


Fig B1 Data Flow Diagram (DFD Level 0) of QARING System

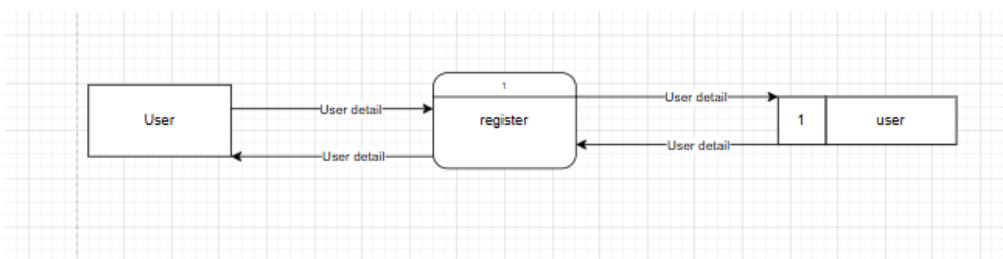


Fig B2 The registration process in the QARING system

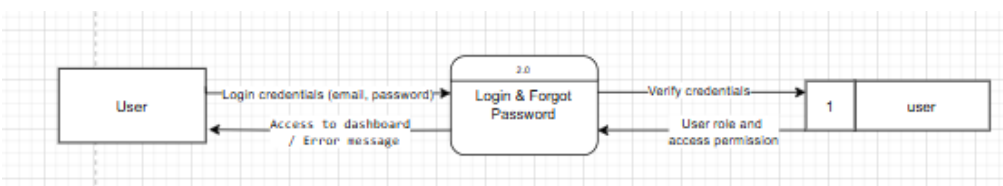


Fig B3 the login and password recovery process

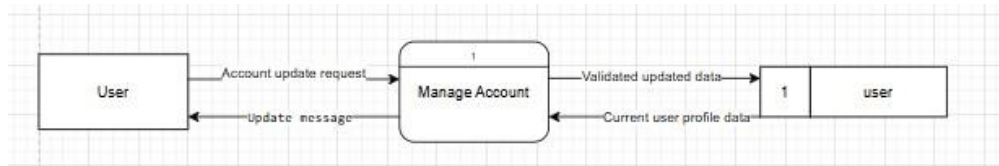


Fig B4 The process of managing user profiles

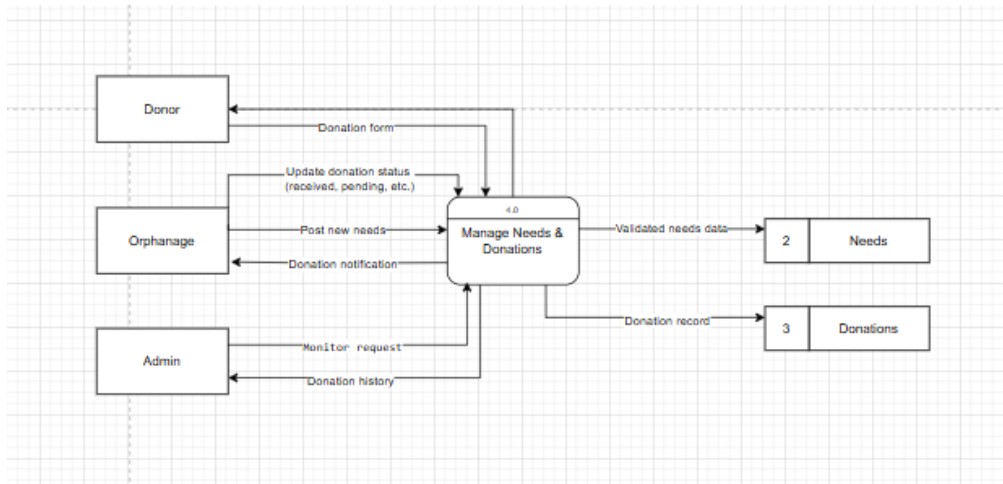


Fig B5 The process for handling item-based and monetary donation requests

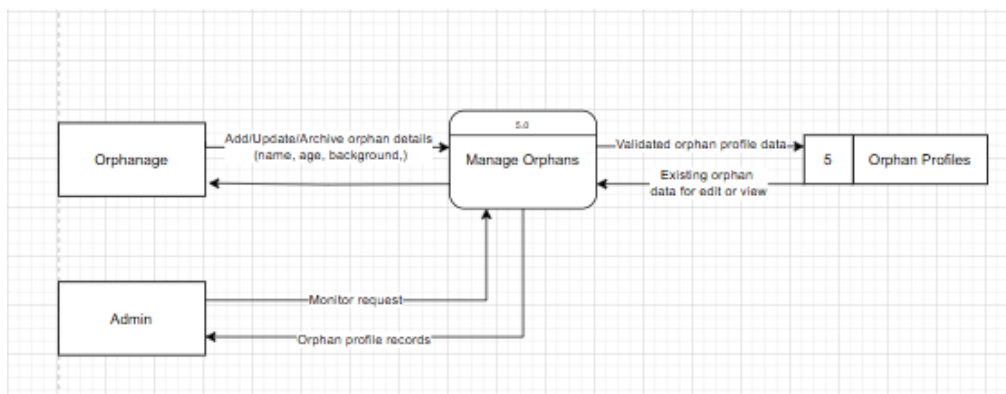


Fig B6 The process of managing orphan profiles

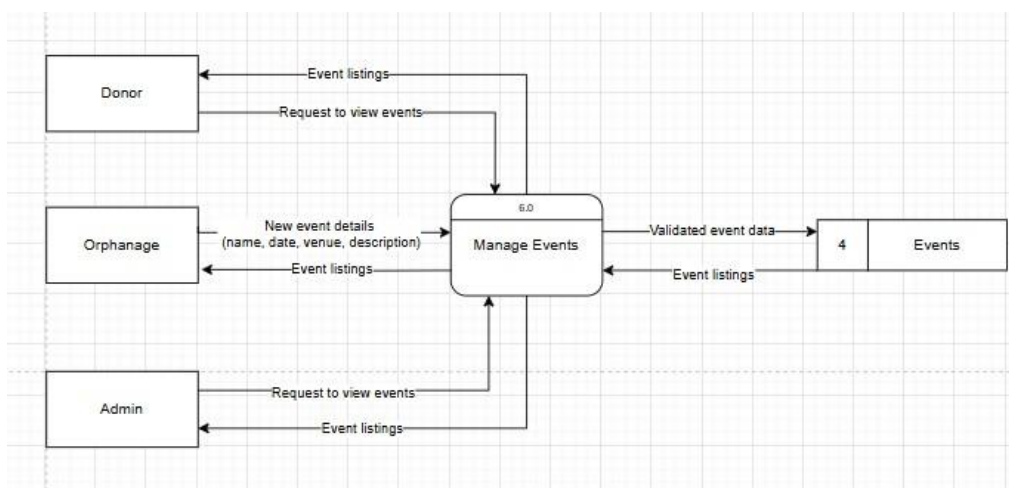
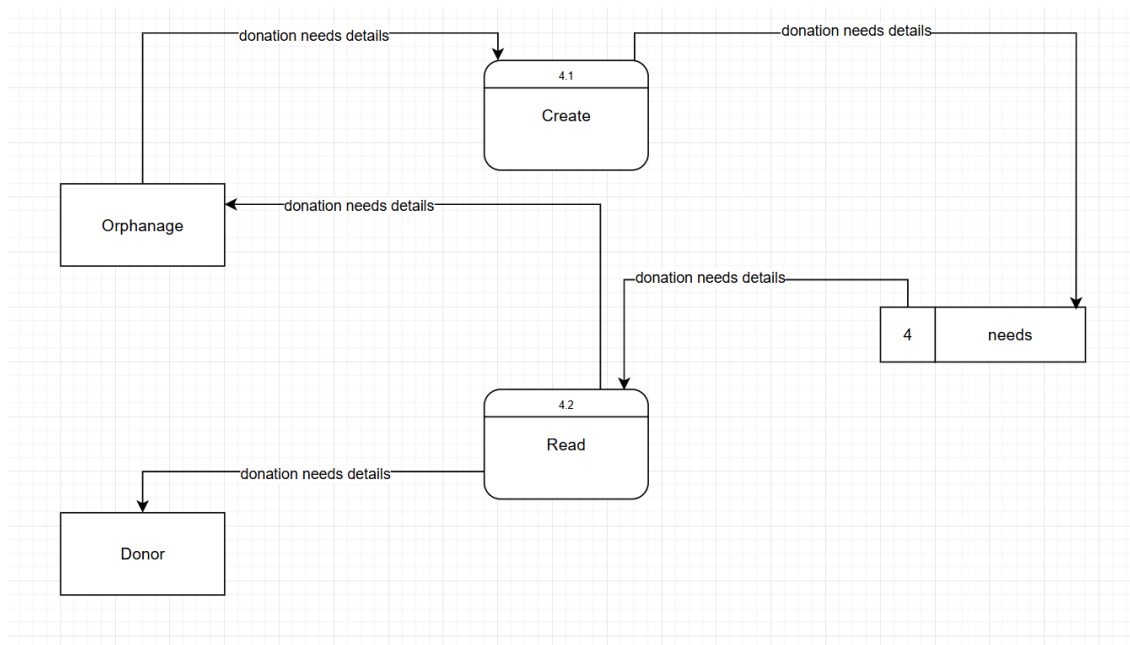


Fig B7 The event management process

**Fig B8** Manage donation

Appendix C: Entity Relationships Diagram

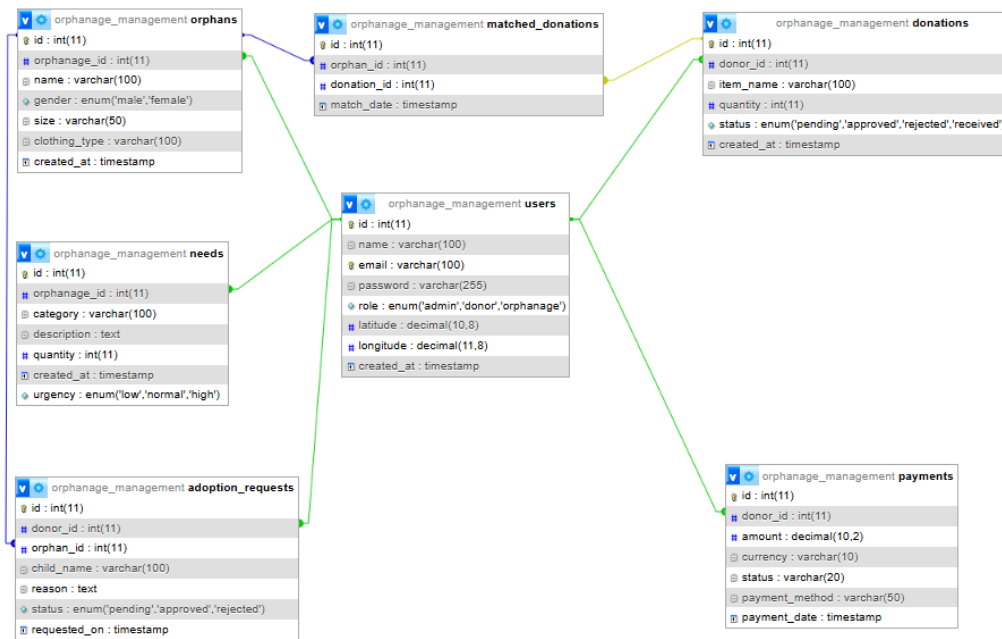


Fig C1 ERD from phpmyadmin

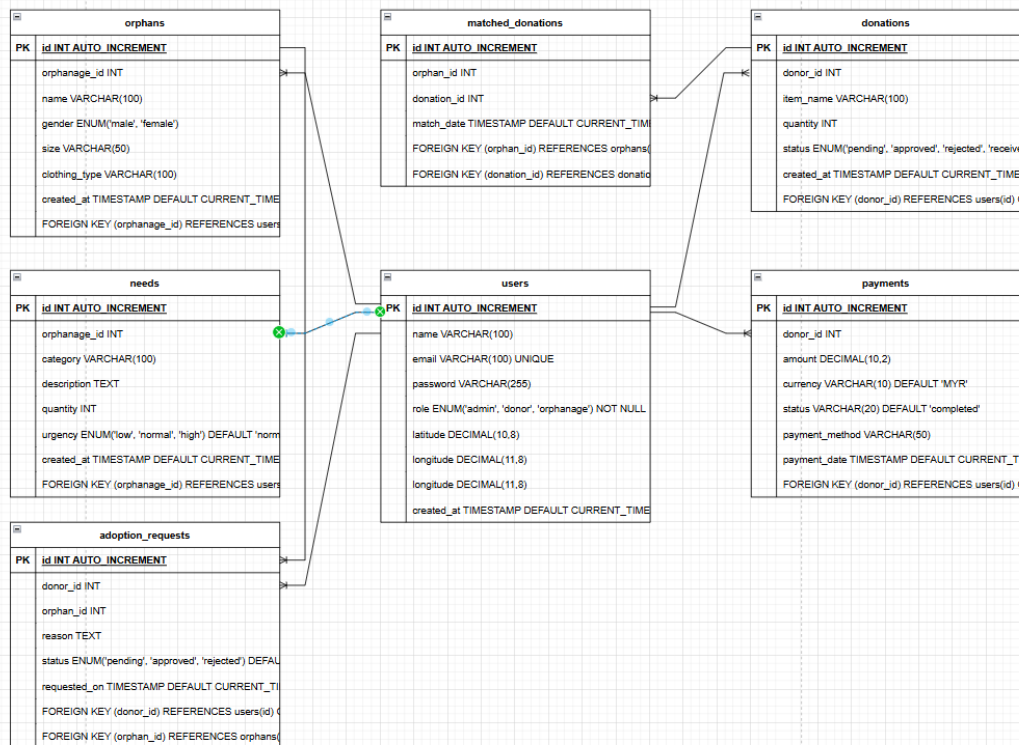


Fig C2 ERD