

Data Request and Record Management System

Nisen Segar¹, Noryusliza Abdullah^{1*}

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

*Corresponding Author: yusliza@uthm.edu.my

DOI: <https://doi.org/10.30880/aitcs.2026.07.01.030>

Article Info

Received: 12 June 2025

Accepted: 18 June 2025

Available online: 30 June 2025

Keywords

Data Management, Decision-Making,
Data Request, Record Management
System, Waterfall Model, Digital
Transformation

Abstract

Effective data management is important for informed decision-making in organizations. Without a proper data management strategy, organizations may make poor decisions that can hinder their competitiveness. The Academic Administration Office (PPA) at University Tun Hussein Onn Malaysia (UTHM) faces challenges such as processing delays, human errors, and inefficient data request handling. This study aims to address these issues by developing a user-friendly Data Request and Record Management System. The system was developed using the waterfall model by following detailed sequential phases. It streamlines data request management, reduces errors, enhances decision-making, and supports digital transformation, ultimately leading to effective data management practices.

1. Introduction

Data is present at every step of the process and managing it is one of the most important aspects. Data management makes sure that data is properly handled so that organizations can turn it into insightful knowledge that informs wise decisions. Data must be timely, accurate, clean, unbiased, and most importantly trustworthy [1]. Data management is a process of creating, gathering, storing, maintaining, and securing data that is created and collected throughout the initial stage to the end. To successfully manage huge volumes of data and allow businesses to execute well-informed decisions, it requires techniques and tools. An organization cannot compete in the market if it does not have sufficient technique for handling data because decisions based on the data are of poor quality.

This project proposes to solve the issues faced by the Academic Administration Office (PPA) at University Tun Hussein Onn Malaysia (UTHM), as well as the challenges faced by the undergraduates and responsibility centre (PTj). This system provides users with an integrated service that includes features like requesting data, determining the status of data, converting conventionally created stuff into digital formats, and more. It increase operational efficiency and reduce the possibility of errors by streamlining the data request and record management. Based on previous research, it has been pointed out that there was no exact system that provided any kind of data request and record management system in the PPA department. Using traditional methods is not that very efficient, which can affect the effectiveness of data management. This system is much needed since implementing such a system will surely lead to future success for UTHM.

This system is a web-based application, and it will be accessed and operated on an internet browser. The targeted user of this system is the PPA department and the data requestor, such as student and responsibility centers (PTj). Detailed information for the PPA department will be provided about who requested the data and what was requested by the requestor. The proposed project's expected result is the successful development and implementation of the Data Request and Record Management System, which will streamline the PPA department's record management and data request operations. By developing this system, all the manual

operations done by the PPA staff will automate, so it will lead to proper data management and eliminate the possibility of human errors happening.

2. Related Work

This section discusses the literature review that has been conducted for the PPA Data Request and Record Management System. The review covers the discussion on the concept and issues on data requests, information system, record management system, and existing system which are used for comparison with the proposed system.

2.1 Data Request

Data is gathered and kept for a variety of uses, and this important data is utilized whenever it is needed for an action. Whether in the government or private sector, there is always a system for managing data and managed by a professional. Traditional data request processes involve delays in submission and handling, often requiring in-person visits to track the status of requests. During the COVID-19 pandemics, most of the firms switched to modern data request systems, such as using Microsoft Forms and Google Forms [2]. These streamline the process, eliminating the need for constant human follow-up, increasing productivity, and speeding up the data request process. An example of such a system is Malaysia's MySejahtera app, launched in April 2020, which allowed users to submit health data for tracking COVID-19. This system helped identify over 9,000 cases, demonstrating the importance of efficient centralized data request platforms for improving decision-making and response times during crises.

2.2 Information System

Information System (IS) is a component that collects, stores, processes, and share information for good decision making. They are integral to daily operations and are present in many forms. In this digital world, every business organization needs an information system to support everything starting from the business planning until to the final output of product delivery [3]. It assists in timely decision-making by automating tasks and making more informed decisions to allow focus on a more critical path. Choosing the right type of information system is crucial to effectively address the challenges and achieving business goals. Implementing the right information system selection can help the company become more flexible.

2.3 Record Management System

Records are recorded information that requires proper management from creation to disposal. An organization depends on the records and data it generates for the analysis, mitigation, and elimination of business risks [4]. It serves as a roadmap for decision-making in the future. A document which does not properly managed can be easily falsified by anyone [5]. Different organizations have their own way on managing their records, whether traditional approach or modern technique. Traditional record management systems are paper-based, manual, and vulnerable to physical damage, mismanagement, and retrieving specific records is time-consuming. Modern record management systems offer faster access, better security, and improved efficiency. Transitioning to modern record management systems is essential to ensure reliable data handling in an information society.

2.4 Review of Existing System

This section discusses the analysis of existing systems, which was conducted to review the systems currently available. This is done to gain a deeper understanding of how these systems operate, identify their advantages and disadvantages, and develop a more effective proposed system.

2.4.1 Freshdesk

Freshdesk is a cloud-based platform that offers efficient handling of requests, ticket management, and record maintenance. Suitable for businesses of all sizes, offering scalable solutions for quick issue resolution. It streamlines tracking and collaboration by allowing teammates across departments to share updates, progress, and visibility on tickets. With support over 40 languages, it facilitates global operations. Freshdesk also integrates with Microsoft Teams, enabling users to create tickets directly within the platform. Issues are logged as tickets, assigned to team members, and updated after resolution. Resolved tickets are saved in logs for future reference.

2.4.2 eAduan KPDN System

The eAduan KPDN system, developed by the Ministry of Domestic Trade and Consumer Affairs, Malaysia, to manage consumer complaints. This system allows users to file complaints online without the need for physical

visits. The complaint can be made easily with necessary information, which is details of the complaint, complaint report, location of the complaint with picture as evidence. Each complaint is logged into a ticketing system for efficient resolution. Complaints are automatically assigned to relevant departments or officers, accompanied by real-time notifications to ensure timely action been taken. Users can track the progress of their complaints in real-time at every stage. The resolved complaints are stored for future analysis.

2.4.3 WeCare UTHM

The UTHM Complaint & Feedback Management System (WeCare UTHM) is designed for UTHM citizens to provide feedback and place complaints. This system is accessible to all users with minimal technical knowledge. Its functionality is like feedback systems used in the government sector with greater efficiency. A more detailed and clearer complaint form is included in this system to enable the user to provide accurate and relevant information. Complaints can be monitored by inputting a ticket number or IC number. The system helps UTHM evaluate and enhance their service quality based on user feedback.

2.5 Comparison of Reviewed System and the Proposed System

A comparison was made between the reviewed applications and the proposed system to understand their similarities and differences. This analysis focused on key features and modules of each system. Table 1 displays the comparison between the existing systems and the proposed application.

Table 1 Comparison table for the reviewed system and the proposed system

Features	Freshdesk	eAduan KPDN	WeCare UTHM	Proposed System
Ticket Management	✓	✓	✓	✓
Notification	✓	✓	✓	✓
Search Engine	✓	✓	✓	✓
Announcement Dashboard	✓	✗	✗	✓
Language Support	✓	✓	✓	✓
Report and Analytics	✓	✓	✓	✓
Role-Based Access Control	✓	✓	✓	✓
Web-Based	✓	✓	✓	✓
Mobile Application	✓	✓	✓	✗
Data Recovery	✓	✓	✓	✓

3. Methodology

The Waterfall Model was chosen to develop the Data Request and Record Management System. It is referred to as a linear-sequential life cycle model, which means that the next phase can only begin once the previous phase is completed [6]. This model is ideal for projects with well-defined requirements and minimal expected changes. The advantage of using the Waterfall Model is that system requirements are collected early, allowing for accurate time and budget estimations. Testing is performed before deployment, identifying and resolving system defects for better functionality. However, unexpected changes or defects may cause delays since testing is done at the end. The waterfall model consists of five stages as shown in Fig 1.

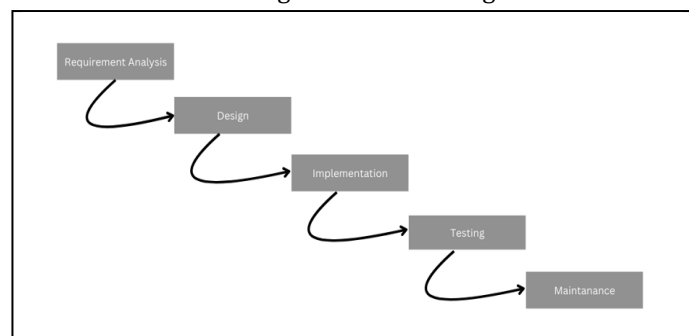


Fig. 1 Waterfall Model

3.1 System Development Workflow

According to the Waterfall Model, there are five phases involved in the SDLC, which serve as guidelines for developing the proposed system. Table 2 shows the tasks conducted in each phase and their output deliverable for the entire project.

Table 2 *System Development Workflow*

Phases	Activities	Outputs
Requirement Analysis	- Conducted interview session with PPA department to understand the existing process. - Journal papers were reviewed to enhance knowledge and fully understand the project's characteristics. - Existing workflows were analyzed to ensure the proposed system effectively includes key features and functions.	- Defined project scope, problem statement, and objectives. - Finalized system requirement. - Proposal for the system - Developed Gantt chart.
Design	- Transformed gathered requirement into system design and workflows.	- Flowchart. - Context Diagram. - Data Flow Diagram (DFD). - Entity Relationship Diagram (ERD) - User Interface (UI) design.
Implementation	- Developed the system using PHP for backend, HTML for structure, CSS for styling, JavaScript for interactivity, and phpMyAdmin for database management - Developed database.	- System prototype. - Database setup.
Testing	- Conducted testing. - Evaluated the prototype. - Identified and fixed system defects.	- Test Case - UAT - Enhance functionality
Maintenance	- Collected errors. - Resolved performance issues after deployment.	- Maintained system updates.

4. Analysis and Design

This section explains the analysis and design of the proposed system.

4.1 System Requirements Analysis

The system requirement analysis was conducted early in the system development through interviews with the PPA department to ensure the system meets stakeholder needs and aligns with the project scope. This section discusses the identified system's functional and non-functional requirements.

4.1.1 Functional Requirements

Functional requirements describe the features and capabilities that a system should have to fulfil the project objectives and goals. The statement of functional requirements is crucial as they outline the system's CRUD operations [7]. Functional requirements act as the foundation for development and act as a guide for designing and implementing the system effectively. Lack of clear functional requirements can result in missing key features, potentially causing the system to fail in meeting stakeholder needs. Table 3 shows the functional requirements of the proposed system.

Table 3 *Functional requirements of the proposed system*

Modules	Activities
Login and Registration Module	- The system should enable users to log in with valid credentials. - The system should support different role-based access controls.
Data Request Module	- The system should enable requestors to submit data requests. - The system should enable users to upload relevant files as supporting details for their request.
Request Tracking Module	- Send real-time notification or emails to users about updates on their request status. - Provide users with view the status of their submitted requests.
Record Management Module	- The system should allow PPA staff to view, approve, or reject the data requests. - All requested data details should be saved in the database for future reference.
Notification Module	- The system should send personalized notification when there are any updates on the status of data request. - Notify PPA staff of pending requests.
Reporting Module	- The system should allow admin to export data in specific format types, such as excel and PDF file formats.

4.1.2 Non-Functional Requirements

The system's nonfunctional requirements are derived from the functional requirements [8]. These requirements focus on the system's performance and usability. Non-functional requirements are very important because they impact the user experience. Even if the system provides all the required functionalities, poor performance or any quality issues can lead to user frustration. This may result in users leaving the system, especially when better responsiveness systems are available. Table 4 shows the non-functional requirements of the proposed system.

Table 4 *Non-functional requirements of the proposed system*

Requirements	Description
Usability	- The system should have a user-friendly and intuitive UI. - The system should be responsive across different operating systems, devices, browsers, and screen sizes.
Performance	- The system should be available to be used at any time except during scheduled maintenance periods. - The system should be able to provide responses to users' requests within a few seconds. - The system should handle peak usage times without significant degradation in performance.
Data Integrity	- The system should make sure that all data is protected from being lost or damaged during use.
Security	- The system should implement strong authorization and authentication mechanisms. - The system should have role-based access controls to ensure users have access only to the information they need.

4.2 Context Diagram

Context Diagram is a high-level diagram that outlines the interaction between external entities and the internal system. It shows the input and output data flow between entities and system. Fig 2 illustrates the data flow between the system and its external entities, which include the user and the admin of the proposed system.

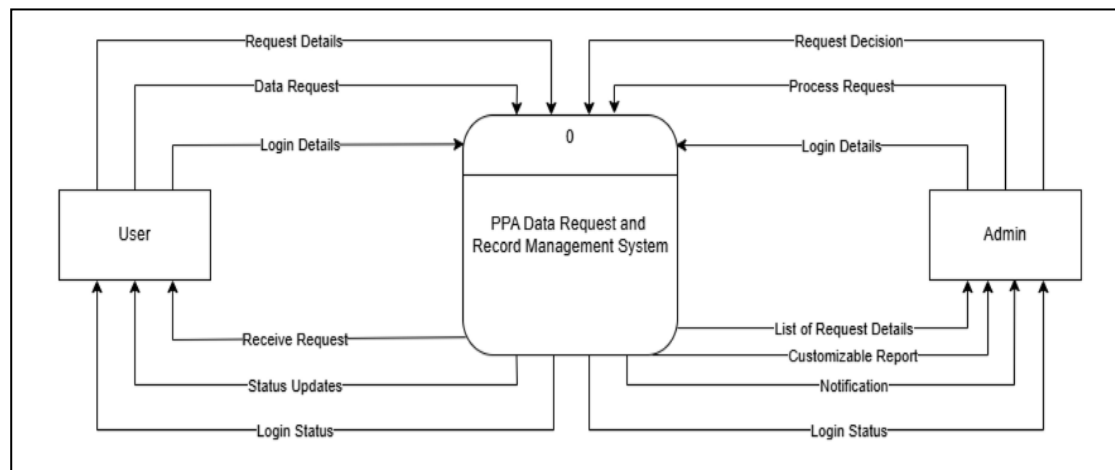


Fig. 2 Context Diagram

4.3 Data Flow Diagram Level 0

Data Flow Diagram (DFD) is a diagram that uses notation to illustrate the workflow of a system, which helps in understanding the system logically and clearly [9]. This graphical representation showcases the interaction between the system, external entities, and data storage. It highlights the main processes that occur in the PPA Data Request and Record Management System. Six important processes are included, with eight data stores integrated. All these processes begin with two external entities, which are admin and user. The Level 0 DFD for this project can be found in Appendix A.

4.4 Entity-Relationship Diagram

The entity-relationship diagram (ERD) is a model that represents the relationships between the entities within the system which consists of entities, attributes, and relationships. It is essential for efficient database design and ensures alignment with the system requirements. The ERD of the developed system can be found in Appendix B.

4.5 Flowchart

A flowchart is a simple graphical representation of how a system functions and operates. This flowchart represents the workflow of the Data Request and Record Management System starting from the login to the end. To enhance clarity, the flowchart has been separated by role to get a better understanding of system's functionality and system flow. The admin role flowchart is shown in Appendix C, while the user role flowchart is presented in Appendix D.

5. Result and Discussion

This section showcases the achievements and outputs derived from this undergraduate project. First, the requirements were gathered at the early stages of system development through an interview session with the PPA department. This was followed by transforming the collected analysis into visual representations that outline the system's functionality and structure. All modules of the proposed system were developed. The process included gathering requirements through interviews, designing the system, and implementing the design in the user interface through coding.

5.1 System Implementation

The PPA Data Request and Record Management System was fully implemented using Visual Studio Code. All the system requirements and functionalities were developed using a combination of programming languages with a specific purpose. For instance, HTML was used to create forms, CSS was applied for designing interactive and user-friendly interfaces, JavaScript was used to enhance client-side interactivity and form validation, and PHP was utilized for backend development to manage data and handle the database connection.

5.1.1 Login and Registration

Fig 3 shows the registration page, where users who wish to submit data requests can register to create an account and gain access to the system. To create an account, users must provide their full name, matric number, password, faculty, and student email address. Fig 4 shows the login page, where both users and the admin will enter their username and password to gain access to the system dashboard.

Fig 3 Registration Page

Fig 4 Login Page

5.1.2 Admin Role

The admin represents the PPA department and plays a crucial role in managing and handling data requests submitted by users, either by solving or rejecting them with a reason. In addition, the admin can add new data request types, generate reports, and perform other administrative tasks for maintaining and monitoring the system.

5.1.2.1 Admin Dashboard Page

Fig 5 shows the Admin Dashboard page. It provides an overview of important information, including the number of solved, pending, and rejected requests. The calendar and upcoming deadlines section highlight request deadlines to help the admin stay informed. Additionally, a pie chart is included to visualize system statistics.

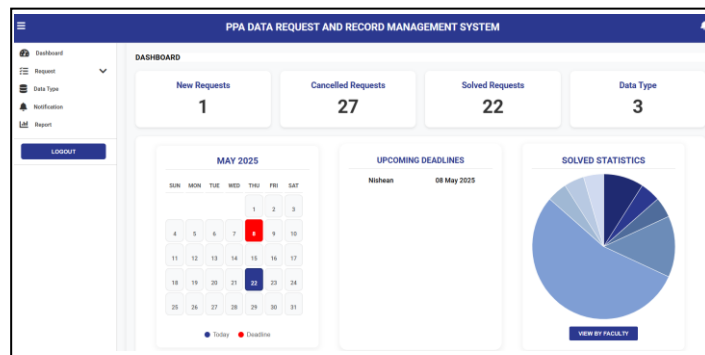


Fig 5 Admin Dashboard Page

5.1.2.2 Request List Page

Fig 6 shows the Request List Page. This page displays all data requests submitted by users. The admin can view the full list and take appropriate actions, either by rejecting a request with a reason or solving it by uploading the required document according to the request type. Fig 7 shows the Upload Document modal, which appears when the admin clicks the upload icon to submit the requested file for a specific request.

The Request List Page displays a table of data requests with the following columns: REQUEST DETAILS, REQUEST TYPE, REQUESTER NAME, USERNAME, FACULTY, CREATED DATE, and ACTION. The table contains one row of data:

REQUEST DETAILS	REQUEST TYPE	REQUESTER NAME	USERNAME	FACULTY	CREATED DATE	ACTION
Request for course feedback	Course	Nisihan	D220140	FSKTM	29 May 2025	Reject

Fig 6 Request List Page

The Admin Upload Document Modal Pop-up shows a form to upload a document for a specific request. The form includes a 'Choose File' button and a 'Name' field. The modal is displayed over the Request List Page.

Fig 7 Admin Upload Document Modal Pop-up

5.1.2.3 Admin Request Rejection Message Page

Fig 8 shows the Admin Request Rejection Message Page. This page allows the admin to compose and send a rejection message to the user after reviewing a data request. The text editor provides a pre-formatted default rejection message, which can be customized to include specific reasons for the rejection.

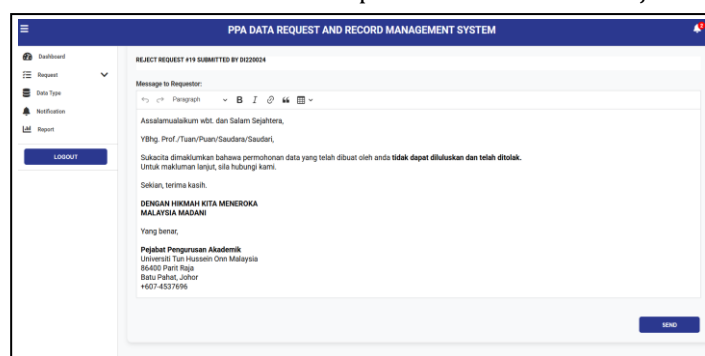


Fig 8 Admin Request Rejection Message Page

5.1.2.4 Solved Request List Page

Fig 9 displays all solved data requests by the admin. Each entry includes relevant request details. Admins can view the submitted data files to ensure transparency and track the history of resolved data requests.

REQUEST DETAILS	REQUEST TYPE	REQUESTER NAME	USERNAME	FACULTY	REQUESTED DATE	SOLVED BY	UPLOADED DATE	ACTION
Request for conference registration data	Conference	Vaighish	D220024	FSKTM	19 Apr 2025	Admin	30 Apr 2025	View
Request for student health records	Health	Vaighish	D220024	FSKTM	18 Apr 2025	Admin	30 Apr 2025	View
Request for departmental meeting minutes	Administrative	Vaighish	D220024	FSKTM	17 Apr 2025	Admin	30 Apr 2025	View
Request for accreditation documents	Accreditation	Vaighish	D220024	FSKTM	14 Apr 2025	Admin	01 May 2025	View
Scholarship	B40 list	Nuhsan	D220140	FSKTM	08 May 2025	Admin	08 May 2025	View

Fig 9 Solved Request List Page

5.1.2.5 Rejected Request List Page

Fig 10 display all rejected data requests by the admin. Admins can view each request's details and the corresponding rejection message. This helps maintain a clear record of the reasons behind each rejection.

REQUEST DETAILS	REQUEST TYPE	REQUESTER NAME	USERNAME	FACULTY	REQUESTED DATE	SOLVED BY	UPLOADED DATE	ACTION
Request for conference registration data	Conference	Vaighish	D220024	FSKTM	19 Apr 2025	Admin	30 Apr 2025	View
Request for student health records	Health	Vaighish	D220024	FSKTM	18 Apr 2025	Admin	30 Apr 2025	View
Request for departmental meeting minutes	Administrative	Vaighish	D220024	FSKTM	17 Apr 2025	Admin	30 Apr 2025	View
Request for accreditation documents	Accreditation	Vaighish	D220024	FSKTM	14 Apr 2025	Admin	01 May 2025	View
Scholarship	B40 list	Nuhsan	D220140	FSKTM	08 May 2025	Admin	08 May 2025	View

Fig 10 Rejected Request List Page

5.1.2.6 Data Type Management Page

Fig 11 shows the Data Type Management Page. This page allows the admin to manage data request types within the system. It functions as a simple Content Management System (CMS) where admins can add new data request types, which will automatically appear in the user data request form. Admins can also delete existing types directly from this interface without needing to modify the database or system code manually.

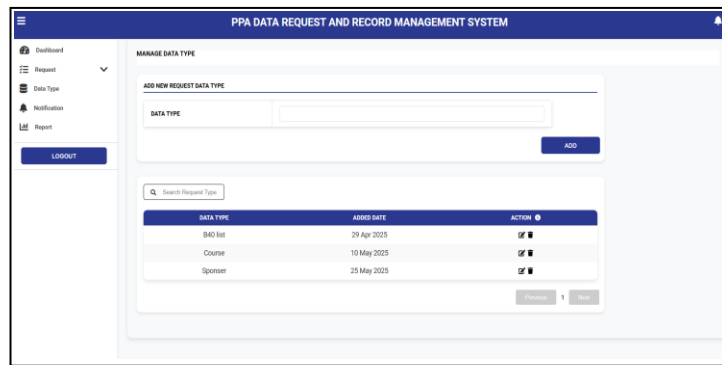


Fig 11 Data Type Management Page

5.1.2.7 Notification Page

Fig 12 shows the Notification Page where all system-generated notifications for the admin are displayed. These notifications appear when a data request submitted by a user remains unresolved for more than three days. This is to ensure no requests are overlooked.

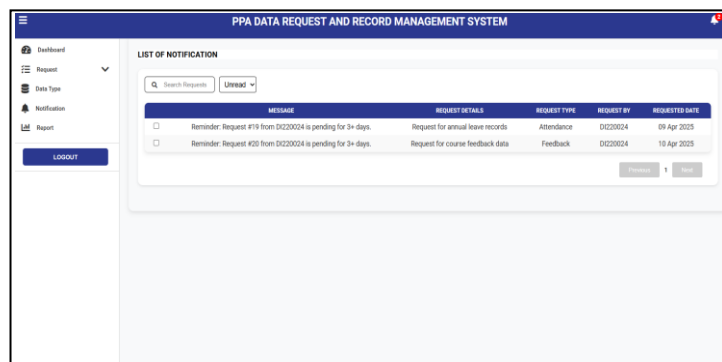


Fig 12 Notification Page

5.1.2.8 Report Page

Fig 13 shows the Report Page. This page allows the admin to generate and view previously generated reports. Reports can be filtered by request type (solved or cancelled), and by specific dates (day, month, or year). Admins can export these filtered reports to PDF or Excel formats depending on their needs.

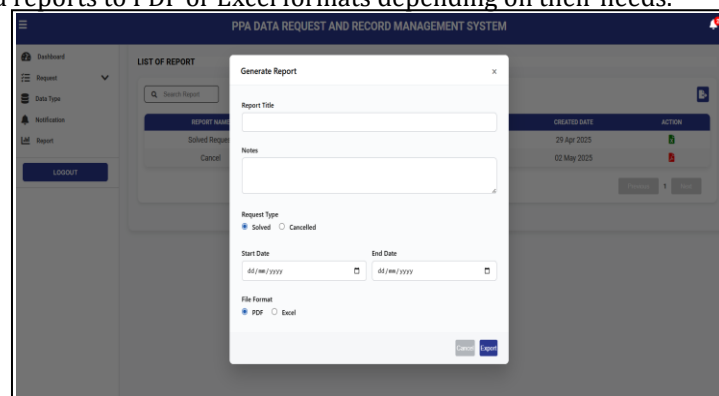


Fig 13 Report Page

5.1.3 User Role

The user in the PPA Data Request and Record Management System is the requestor, who submits data requests to the PPA department. This section showcases a user-friendly interface that enables users to submit their data requests and track the progress once submitted.

5.1.3.1 User Dashboard Page

Fig 14 shows the User Dashboard Page. This page displays all data requests submitted by the logged-in user that are still in the pending status. The user still can edit or delete their own pending requests from this page.

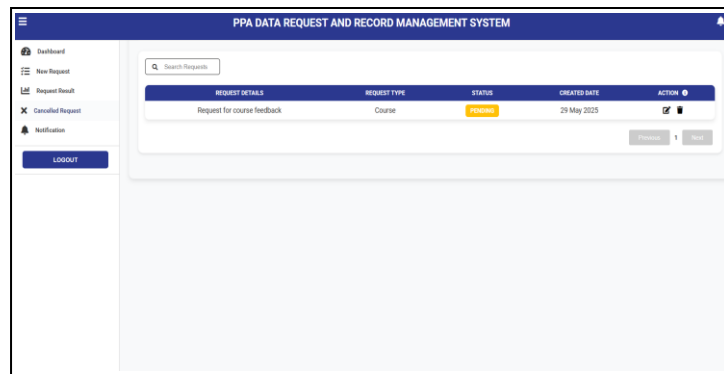


Fig 14 User Dashboard Page

5.1.3.2 New Data Request Page

Fig 15 shows the New Data Request Page. This page displays a form where the user is required to provide request details, upload a supporting document to validate the request, and select the appropriate data request type. The user must check the checkbox to acknowledge the terms before submitting the data request.

Fig 15 New Data Request Page

5.1.3.3 User Solved Request Page

Fig 16 illustrates the User Solved Request Page, where the logged-in user can view all the data requests that have been successfully solved by the admin. This page displays requests made by the particular logged-in user and allows them to access the files uploaded by the admin in response to their request. Users can also download these files directly from the page itself.

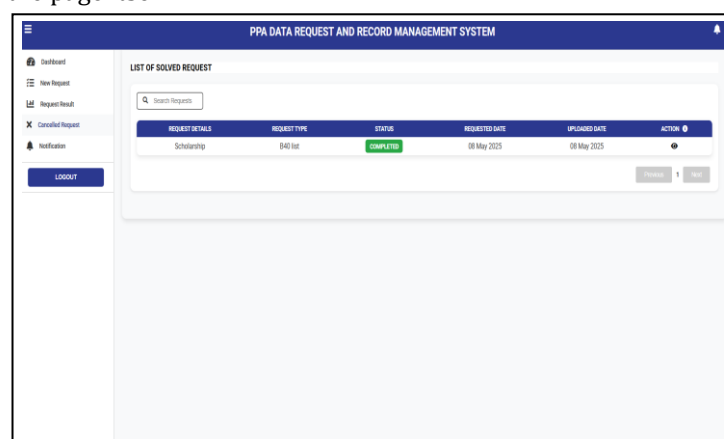


Fig 16 User Solved Request Page

5.1.3.4 User Request Rejection Page

Fig 17 illustrates the User Request Rejection Page, where the logged-in user can view only their own data requests that have been rejected by the admin. For each rejected request, the user can access the supporting documents they submitted and view the corresponding rejection message from the admin.



Fig 17 User Request Rejection Page

5.1.3.5 User Notification Page

Fig 18 shows the User Notification Page, where the logged-in user can receive and view system-generated notifications. These notifications are primarily triggered for two reasons: when a request submitted by the user has been successfully completed, or when it has been rejected.



Fig 18 User Notification Page

5.2 System Testing

System testing is a type of testing where an entire newly developed system is tested against its specifications [10]. It plays an important role in ensuring the system has been developed according to what was planned in the previous chapter without any errors. Both test cases and User Acceptance Testing (UAT) were conducted for system testing to validate the functionality, reliability, and usability of the system.

5.2.1 Test Cases

Test cases play an important role in the testing process, since it verifies the implementation of the system whether meets the required specifications and functions as expected [11]. During the testing phase, test cases were executed to ensure that each module functions correctly. Table 5 shows the results of the test cases conducted for the proposed system.

Table 5 Test case result of the proposed system

Login and Registration Module			
Function	Test Cases	Expected Outcome	Actual Output
User registration	Fill in the registration form and submit.	New user accounts are created and redirected to login page.	As expected outcome
Role-based login authentication	Enter valid credentials as user or admin.	Redirected to the appropriate dashboard based on the role.	As expected outcome
	Enter incorrect credentials.	Error message: "Invalid".	As expected outcome

Table 5 *Test case result of the proposed system(continue)*

Data Request Module			
Function	Test Cases	Expected Outcome	Actual Output
Submit data request	User submits requests with details and supporting file.	Request saved successfully, confirmation message shown.	As expected outcome
Manage submitted requests	Users can view, update, or delete their requests.	CRUD operations are available for user requests.	As expected outcome
Request Tracking Module			
Function	Test Cases	Expected Outcome	Actual Output
View submitted requests	User views request in Dashboard Page.	The list of submitted requests is shown with current statuses.	As expected outcome
Track request status	Admin updates status to "Completed" or "Rejected"	User sees updated status and receives notification and email.	As expected outcome
Record Management Module			
Function	Test Cases	Expected Outcome	Actual Output
Manage Request	Admin clicks rejects, adds message, and submits.	Status changes to "Rejected", and the rejection message is sent via email and system notification.	As expected outcome
	Admin uploads requested data file for a pending request.	Status changes to "Completed", and the user is notified via email and system notification.	As expected outcome
Notification Module			
Function	Test Cases	Expected Outcome	Actual Output
Admin Notification	Admin receives an email for new requests, and system notification for pending requests older than 3 days.	Admin is notified via email for new requests, and via system for pending requests older than 3 days.	As expected outcome
User notifications	User receives email and system notification when their request is submitted or rejected.	Users are notified via email and system when their request is submitted or rejected.	As expected outcome
Report Module			
Function	Test Cases	Expected Outcome	Actual Output
Manage report	Admin generates reports for solved or cancelled request.	Report is generated successfully in the selected format.	As expected outcome
	Admin view and downloads the generated report.	Report is downloaded successfully in the selected format.	As expected outcome

5.2.2 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) is the process of verifying with actual users that the developed system meets their requirements and expectations. It is typically the final phase of testing before the system goes live, where the system is assessed not as individual features but as a complete and functioning entity. In this project, UAT was conducted with end users of the PPA Data Request and Record Management System to identify any remaining system errors and validate that each functions performed correctly. Table 6 shows the result of UAT.

Table 6 *User Acceptance Testing (UAT)*

Scenario	Test Steps	Expected Result	Pass/Fail
UAT-001 – User/Admin Login	1. Go to login page 2. Enter valid credentials 3. Click "Login"	User/Admin is authenticated and redirected to an appropriate dashboard.	Pass

Table 6 User Acceptance Testing (UAT)

UAT-002 – User Registration	1. Go to registration page 2. Fill in user details 3. Click “Register”	User account is created and can log in successfully	Pass
UAT-003 – Submit new request	1. Login as user 2. Navigate to request form 3. Fill required fields 4. Upload document 5. Submit	Request is saved in database and admin receives notification.	Pass
UAT-004 – Admin rejects a request	1. Login as admin 2. Go to pending requests 3. Click “Cancel Symbol” 4. Type rejection reason and submit	Request moves to cancelled requests, and users receive rejection email and notification.	Pass
UAT-005 – Admin solves a request	1. Login as admin 2. Go to pending requests 3. Upload requested data file 4. Submit	Request moves to data records, and users receive solved request notification and email.	Pass
UAT-006 – View rejection message	1. User logs in 2. Go to rejected requests 3. Click “Cancel Symbol”	Rejection message shown in a modal	Pass
UAT-007 – View Solved Request	1. Login as user 2. Go to “Solved Requests” page 3. Click eye icon button	Requested data file view in new tab.	Pass

6. Conclusion

The development of this system has been carried out while fulfilling the system objectives and purposes. This development will undoubtedly benefit the PPA department and improve overall data management workflow and surely will resolve all the existing challenges faced by the PPA department, such as last-minute data request management issues and inefficiencies in handling data requests. There are still some areas that need improvement, but with future upgrades and development, the system will surely become more systematic, and user-friendly. The system is expected to be highly beneficial to the PPA department and other related users, as it effectively manages both data requests and records.

Acknowledgement

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

Conflict of Interest

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

Author Contribution

This journal requires that all authors take public responsibility for the content of the work submitted for review. The contributions of all authors must be described in the following manner:

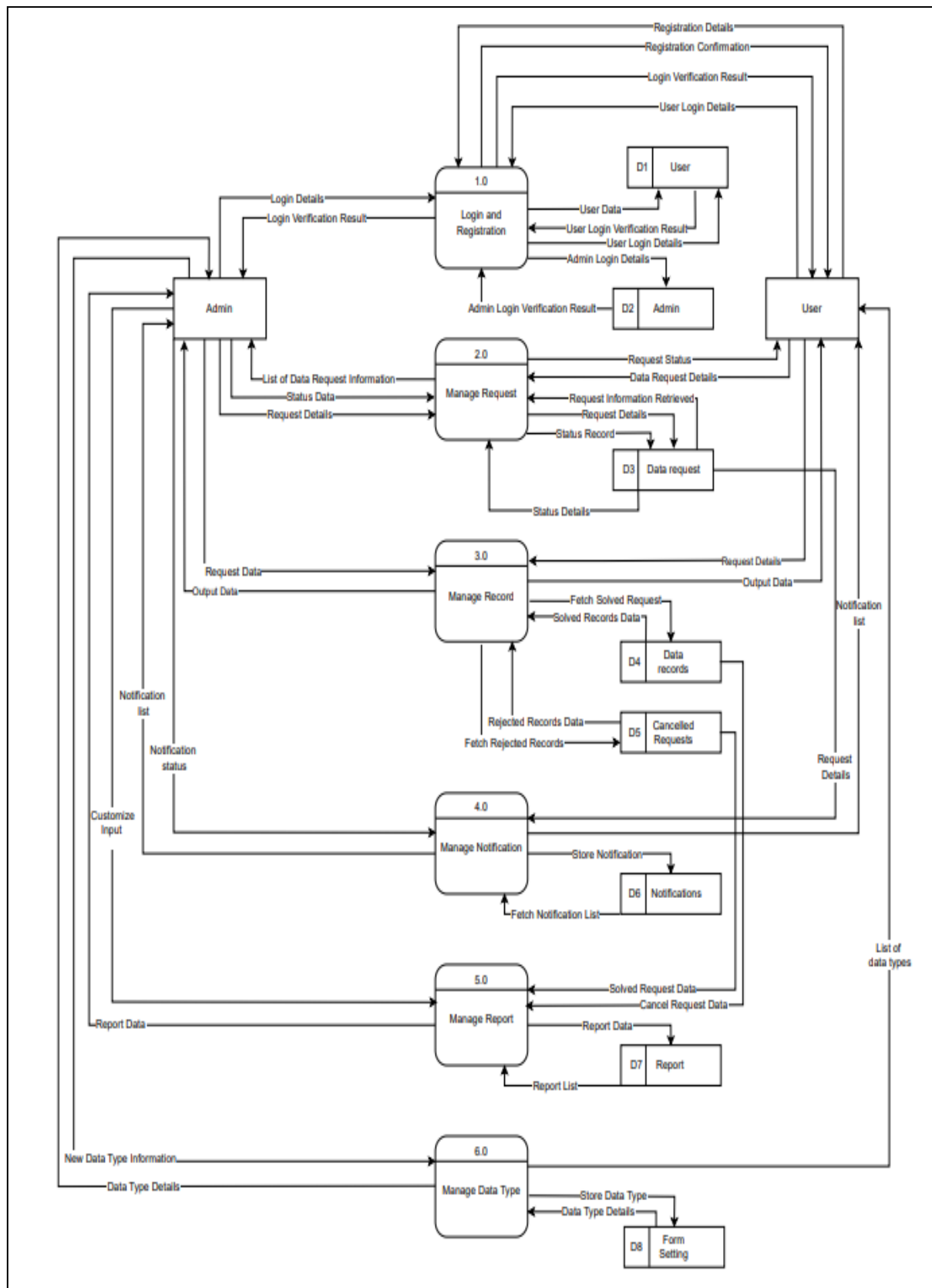
*The authors confirm contribution to the paper as follows: **study conception and design:** Nisen Segar, Noryusliza Abdullah; **data collection:** Nisen Segar; **analysis and interpretation of results:** Nisen Segar, Noryusliza Abdullah; **draft manuscript preparation:** Nisen Segar, Noryusliza Abdullah. All authors reviewed the results and approved the final version of the manuscript.*

References

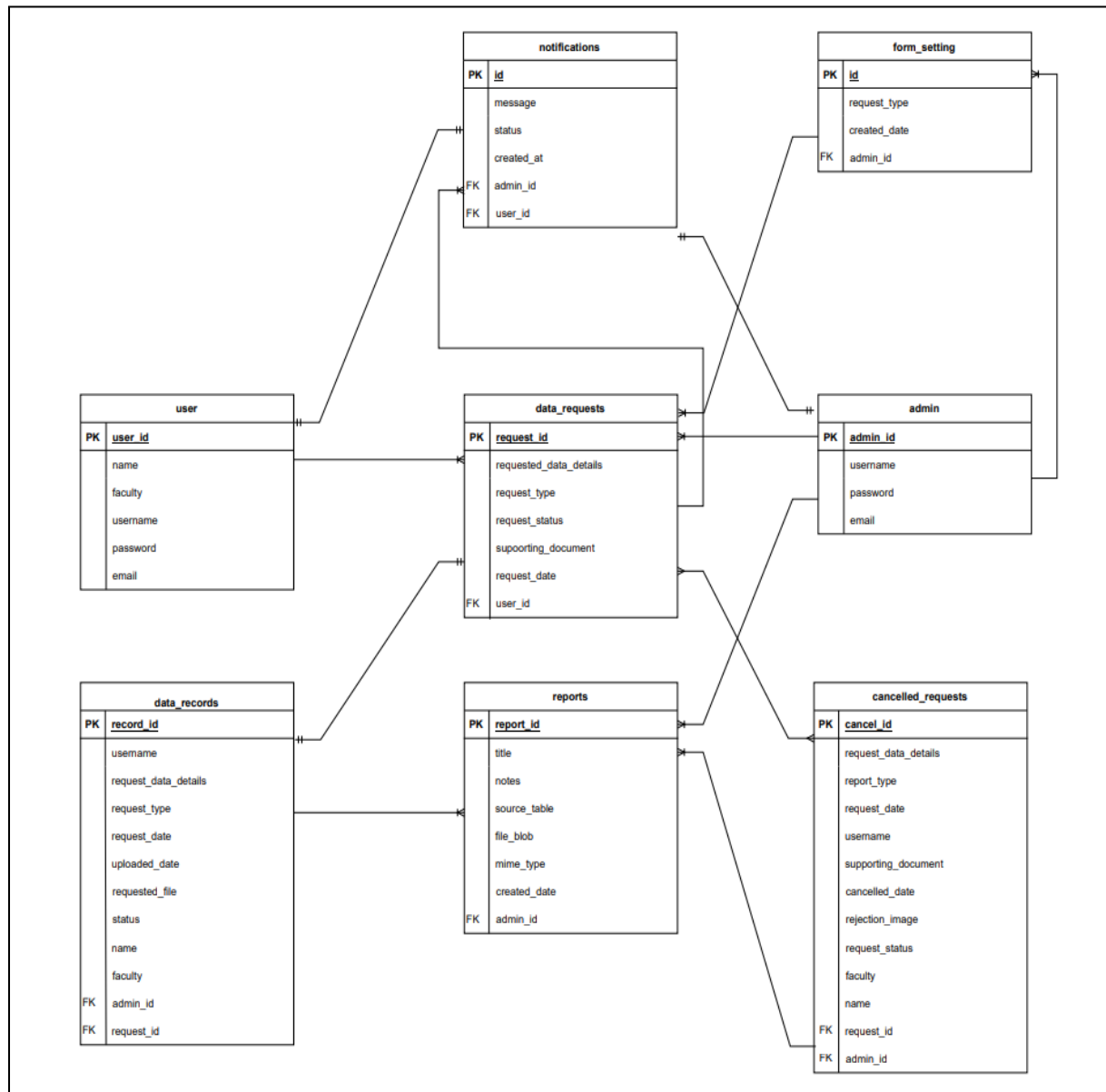
- [1] Anderson, C. (2015). Creating a Data-Driven Organization (First). https://books.google.com.my/books?hl=en&lr=&id=MVpDCgAAQBAJ&oi=fnd&pg=PR2&dq=why+data+is+important+in+organization&ots=Ik314i5TAV&sig=HH0pk-H5Wpo7mYQIBiGWdGUDv8M&redir_esc=y#v=onepage&q&f=false

- [2] Rassameeroj, I., Jomkhamsri, P., & Thaithaweewattana, N. (2022). Student Request System Prototype Using Low-Code Development Platform. Proceedings - 2022 International Conference on Algorithms, Data Mining, and Information Technology, ADMIT 2022, 190–195.
<https://doi.org/10.1109/ADMIT57209.2022.00038>
- [3] Nowduri, S., & DBA, S. A.-D. (2012). Management Information Systems and Its Support to Sustainable Small and Medium Enterprises. International Journal of Business and Management, 7(19).
<https://doi.org/10.5539/ijbm.v7n19p125>
- [4] Mohamed, F. E., & Bunawan, A.-A. (2022). Effective Records Management Program in Organization. International Journal of Academic Research in Business and Social Sciences, 12(4).
<https://doi.org/10.6007/ijarbss/v12-i4/12998>
- [5] Nyampong, S. A. (2015). Electronic Records Management in National Development: A Case Study in Ghana Immigration Service. In European Journal of Business and Management www.iiste.org ISSN (Vol. 7, Issue 10). Online. <https://www.iiste.org/Journals/index.php/EJBM/article/view/21337/22385>
- [6] Er Parag Verma. (2015). Waterfall Model : Design Phase, advantages disadvantages and applications.
<https://er.yuvayana.org/waterfall-model-design-phase-advantages-disadvantages-and-applications/>
- [7] Becker, P., Tebes, G., Peppino, D., & Olsina, L. (n.d.). Applying an Improving Strategy that embeds Functional and Non-Functional Requirements Concepts Aplicando una Estrategia de Mejora que incluye Conceptos de Requisitos Funcionales y No Funcionales. Retrieved December 30, 2024, from
https://sedici.unlp.edu.ar/bitstream/handle/10915/87772/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y
- [8] Summers, J. D., & Morkos, B. (2014). Requirements Evolution: Impact of Functional and Non-Functional Change on Project Success. <https://www.researchgate.net/publication/255568492>
- [9] Surya HendraPutra SE, M. K., & Yulia Endah Sari. (2019). casc.
<https://jurnal.polgan.ac.id/index.php/remik/article/view/10113/230>
- [10] Briand, L., & Labiche, Y. (2002). *A UML-Based Approach to System Testing*.
[https://academia.edu/4295048/A UML Based Approach to System Testing](https://academia.edu/4295048/A_UML_Based_Approach_to_System_Testing)
- [11] Jim. (2001). *Generating Test Cases From Use Cases*.
http://www.therationaledge.com/content/jun_01/m_cases_jh.html

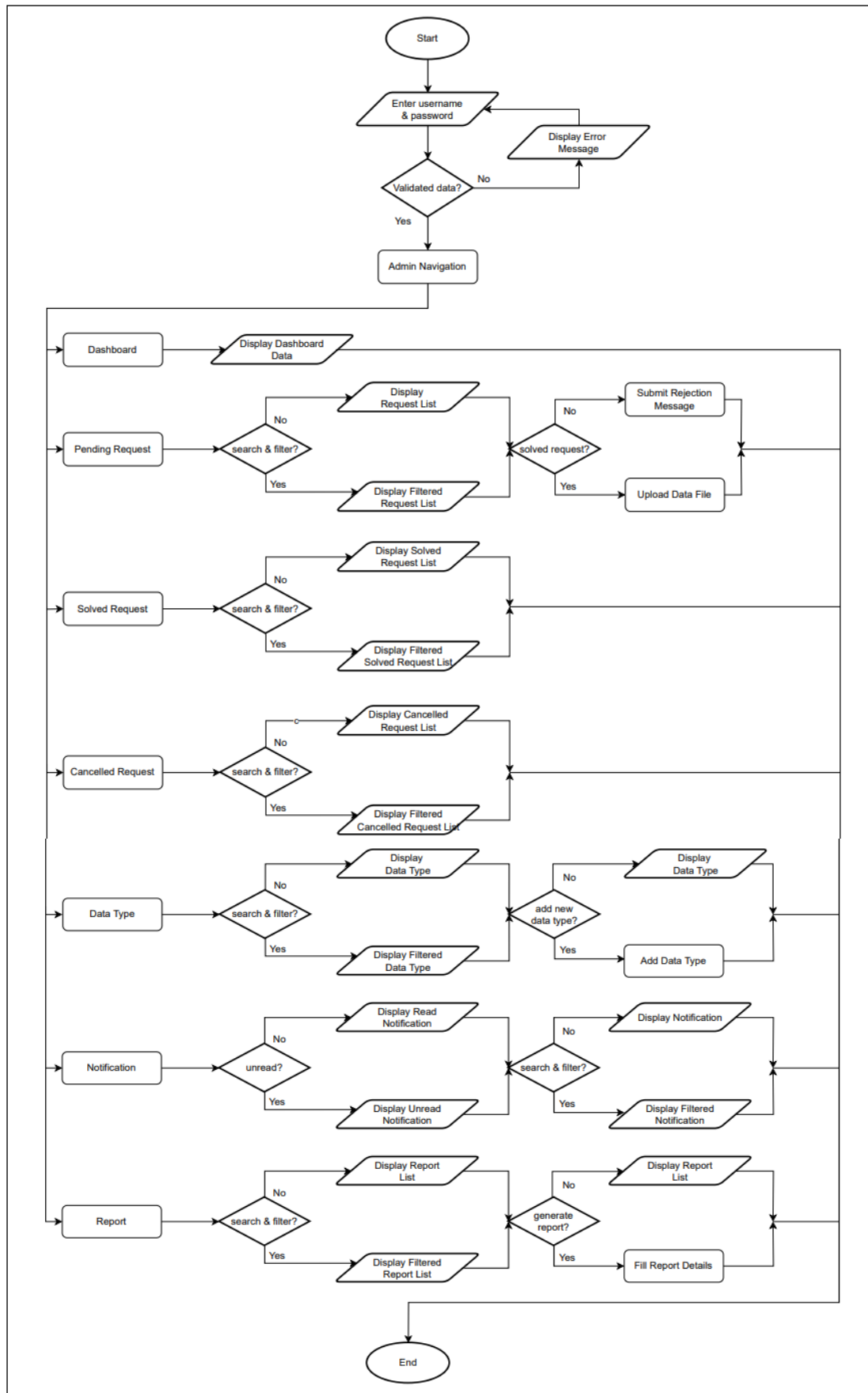
Appendix A: Data Flow Diagram Level 0



Appendix B: Entity-Relationship Diagram



Appendix C: Admin Flowchart



Appendix D: User Flowchart

