APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE
e-ISSN: 2773-5141

AITCS

Vol. 5 No. 2 (2024) 1180-1199
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

AstroWealth: Personal Finance Management Web Application

Putri Khaireen Jasmin Ahmad Khairi[1], Suhaila Mohd. Yasin[1]*

[1] *Fakulti Sains Komputer dan Teknologi Maklumat,*
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: ysuhaila@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2024.05.02.062

Article Info

Received: 14 July 2024
Accepted: 30 October 2024
Available online: 15 December 2024

Keywords

Financial Management, Personal Finance, Web Application, Object-Oriented Approach.

Abstract

AstroWealth is a comprehensive personal finance management web application specifically designed to assist university students in building and managing their wealth as they transition into adulthood, all while receiving support from their parents. Many students today continue to rely on outdated and traditional methods for financial management, which have often proven to be inefficient and inadequate in addressing the unique challenges they face. Recognizing these limitations, AstroWealth offers a collaborative and modern approach that underscores the significance of parental involvement and thorough personal finance management, particularly crucial in an era where educational expenses are increasing. AstroWealth comprises eight distinct modules, each serving a specific purpose. The application is developed using an object-oriented approach which emphasizes objects to model real-world financial entities and their interactions using UML diagrams and PHP, a widely-used server-side scripting language. Focusing on collaborative financial management, AstroWealth stands out as a vital tool in today's educational landscape.

1. Introduction

As university students embark on their independent journeys, managing personal finances becomes a pressing reality. However, the lack of adequate knowledge and accessible tools leaves them navigating this financial landscape without a compass. Many find themselves at a crossroad, facing the intricacies of budgeting, expenses, and financial planning without the necessary knowledge or digital collaborative approach with their parents. Many students juggle part-time jobs, scholarships, loans, and parental support to make ends meet. They often struggle to track and manage their finances effectively, leading to financial stress, uncertainty, and, in some cases, debt accumulation. Personal finance, which focuses on how people spend, save, protect, and invest their money to achieve financial success, is the study of individual and family resources that are thought to be significant [1]. The findings of the research show that even though students have easy access to financial services like student loans and credit, they lack the expertise in money management that could prevent them from managing their finances properly and cause financial issues [2] - [5].

Students predominantly utilize traditional double-entry methods, employing either digital or physical notebooks to manually record their financial transactions. However, this conventional approach presents challenges such as error-prone manual calculations, potential inaccuracies in monthly financial figures, and the vulnerability of paper records to wear and tear. The extensive list of records and the tedious process make it difficult for students to efficiently track expenses on specific dates, leading to disengagement and reduced motivation amid the demands of daily academic commitments. To address these challenges comprehensively, the proposed AstroWealth will provide an accessible solution which simplifies expense tracking, addressing the

inefficiencies of the current method, eliminating the need for extensive searching, and accommodating parental involvement in financial management.

This project follows an object-oriented approach to ensure a systematic and structured development process. AstroWealth consists of eight distinct modules, each tailored to cater to specific needs of student financial management and offers comprehensive functionalities for effective personal finance management, allowing students and parents to seamlessly add, edit, and delete expenses and income sources, create budgets for different expense categories, and set savings targets. The manage students module enable parent to add or delete a student to their account in order to have access to their financial transactions. Additionally, it enables students to generate detailed financial reports, offering insights into monthly spending. This project holds significance in tackling the crucial challenges that university students encounter in handling their personal finances. It aims to revolutionize traditional methods and minimizing the risk of financial mismanagement

There are five sections in this paper. Section 1 describes the project's background, setting the context for the study. Section 2 explores related works, offering insights into existing research and projects in the field. In Section 3, the methodology is detailed, outlining all necessary information for obtaining study results. Section 4 presents the results and discussion. Finally, Section 5 summarizes the main findings of the study in the conclusion.

2. Related Work

In this section, a comprehensive exploration of pertinent areas is undertaken. This encompasses delving into the technologies, examination of related systems that provided valuable insights into existing frameworks and features relevant to personal finance management.

2.1 Work Approaches

AstroWealth, an online financial management system uses personal finance management methodologies as its foundation, focusing on essential skills like budgeting, saving, and expense tracking to help students manage their finances effectively. As a web-based application, AstroWealth leverages the accessibility and efficiency of online platforms, providing dynamic financial tools and information across multiple devices without the need for specific hardware or software installations. PHP, created in 1994, is a core programming language for dynamic web pages, requiring developers to write code scripts without relying on libraries [6]. When combined with MySQL, PHP is highly effective for dynamic, database-driven web design and remains popular due to its open-source nature. The addition of JavaScript and CSS enables the creation of interactive web applications. Despite not being inherently object-oriented, PHP includes features that facilitate web-based application development, making it suitable for projects like AstroWealth.

2.2 Study of Existing Systems

This project delves into a comparative analysis of three leading budgeting and personal finance systems: Mobillis, Goodbudget, and EveryDollar. By examining their features, user interfaces, and overall effectiveness, this study aims to uncover insights into the diverse approaches these platforms take to assist users in managing their finances. The first system to be reviewed is called Mobills [7]. It is a personal finance management app designed to help users track and manage their expenses, create budgets and gain insights into their spending habits. Users can categorize their expenses, keep budget plans, and receive notifications to stay on top of their financial activities. It features a user-friendly interface, an intuitive data entry mechanism, and basic income and expenses control functions

The second system to be reviewed is called Goodbudget [8]. Goodbudget is a personal finance and budgeting application that uses the envelope budgeting method. It is designed to help users manage their money by allocating funds into virtual envelopes for different spending categories. Users can set budget limits for each category and track their spending against those limits. Goodbudget allows for manual entry of transactions, and it provides a visual representation of the budgeting process. The virtual envelopes are divided into two sections: monthly expenses and annual expenses. It features a comprehensive list of transactions, the ability to generate various types of reports, and a virtual envelope system.

The third system to be reviewed is called EveryDollar [9]. EveryDollar is a budgeting application developed by Ramsey Solutions, associated with financial expert Dave Ramsey. EveryDollar is designed to assist users in creating and managing a monthly budget. Users can track their income, expenses, and savings goals, categorizing transactions to get a clear picture of their financial situation. The app follows a zero-based budgeting approach, where every dollar has a designated purpose. It features essential basic tasks including adding, deleting, and editing income and expense records, and a budget management system divided into three categories: planned, spent, and remaining.

AstroWealth is a web-based application designed for recording and tracking expenses and income with predefined and custom categories. It facilitates budget creation, allocates spending limits, issues real-time alerts for budget exceedance, parental involvement in monitoring transactions, and generates financial reports. A comparative analysis in **Table 1** assesses three existing systems and AstroWealth, aiming to provide valuable insights for selecting an optimal student-centric personal finance management solution

Table 1 *Systems' comparison*

Features \ System	Mobills	Goodbudget	EveryDollar	AstroWealth
Manage Transactions	√	√	√	√
Manage Budgets	X	√	√	√
Manage Goals	√	X	X	√
Manage Students	X	X	X	
Generate Report	√	√	√	√
Alerts of Budgets' Exceedance	X	X	X	√
Category Customization	X	√	√	√

According to the results in **Table 1**, it was observed that the three existing systems lack certain features under consideration. Mobills, for instance, only meets the manage transactions, goals, and report generation. In contrast, Goodbudget and EveryDollar encompass four fundamental aspects of personal finance management, including alerts and category customization. Most importantly, none of these systems support parental progress monitoring which is the manage students module. Recognizing these limitations, the proposed system aims to incorporate all these features to build a comprehensive personal finance management system.

3. Methodology

Central to this approach is the use of an object-oriented approach. The Object-Oriented Approach in software development have developed to assist developers in leveraging the expressive capabilities of object-based and object-oriented programming languages, utilizing the class and object as basic building blocks [10]. The object-oriented approach was chosen because it includes the use of Unified Modelling Language (UML) diagrams. UML diagrams provide a graphical representation of the system's architecture, fostering clearer communication and design visualisation. The object-oriented approach involves several phases, similar to the traditional software development life cycle. As the specific stages are explored, including requirements analysis during initiation and planning, analysis and design, programming (implementation), and testing, these principles guide the systematic creation and organization of the proposed system. **Fig. 1** shows the life cycle of object-oriented approach.

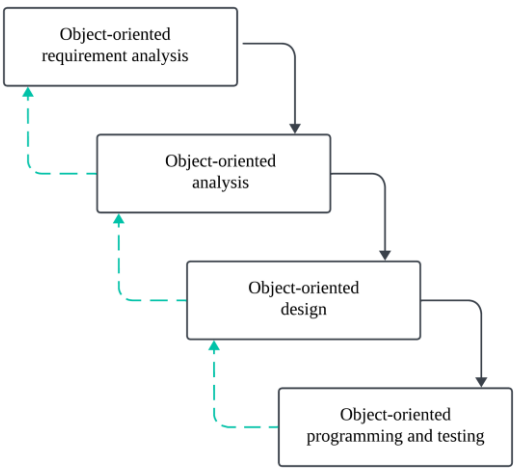

Fig. 1 Object-oriented life cycle model [11]

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 1180-1199

3.1 Object-Oriented Requirement Analysis Phase

During the planning phase of developing a web application for student personal finance management, the utilization of object-oriented requirements analysis plays a pivotal role. This approach establishes a comprehensive understanding of the web application, laying a strong foundation for subsequent development stages before delving into detailed analysis or design. It aids in defining project objectives, scopes, and identifying stakeholders, addressing specific challenges faced by students in managing their finances. Thorough requirements analysis ensures a clear development roadmap, aligning the application precisely with user needs for effective financial management. Additionally, the exploration of expected results and project significance clarifies the overall use of the proposed system. Setting development tools is a crucial sub-activity, determining the tools to be used in each development phase. Hardware and software requirements specifications for AstroWealth are detailed in **Table 2** and **Table 3**, respectively. The Gantt chart is available in **Appendix A**.

Table 2 *Hardware requirements specification*

No.	Hardware	Specification
1.	Central Processing Unit (CPU)	Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz
2.	Random Access Memory (RAM)	8 Gigabyte (GB)
3.	Storage	256 Gigabyte (GB) Solid State Drive (SSD)

Table 3 *Software requirements specification*

No.	Type	Software	Functionality
1.	Integrated Development Environment (IDE)	Visual Studio Code (VSCode)	Write, debug, and manage codes efficiently
2.	Development Environment	Hypertext Preprocessor (PHP), Hypertext Markup Language(HTML5), Cascading Style Sheets (CSS3), JavaScript	Used to develop the backend logic and frontend of the web application
3.	Design and Management Tool	WordPress, Figma	Design and edit graphics
4.	Database	MySQL	Storing user account information, transaction history, and other relevant financial data securely.
5.	Operating System	Microsoft Windows 10	To manage files, run development tools, and testing the web application.

3.2 Object-Oriented Analysis Phase

The project involves gathering various types of requirements—user, functional, and non-functional—to shape the design and development of the proposed system, AstroWealth. Apart from collecting input from stakeholders, the study examines existing similar applications as a primary guide for enhancing the proposed system. AstroWealth aims to meet the needs of university students by comparing it with three other personal finance web apps: Mobills, Goodbudget, and EveryDollar. The study reveals important details like features' limitations, design ideas, and advantages from existing systems. Following this, the research explores an appropriate Software Development Lifecycle Model for the project, opting for an object-oriented approach. The development stage uses Unified Modelling Language (UML) for a high-level design of user interaction and includes the initial design of the database schema and user interface.

3.3 Object-Oriented Design Phase

In this phase, the creation of the proposed system's entire architecture is undertaken, which includes the development of a class diagram. A pivotal sub-activity involves the design of the database schema, entailing the structuring and organization of the data storage system in alignment with the identified objects and relationships from the analysis phase. This encompasses the definition of tables, fields, and relationships between different entities to ensure the efficient capture and management of data within the proposed system.

To ensure correct storage and retrieval of all pertinent data, the data dictionary describes the metadata of each entity, including data type, data size, and primary key. Concurrently, the development of the user interface design for the front-end is carried out, covering aspects such as layout, navigation, and visual elements that harmonize with the identified objects and behaviours to enhance overall usability.

3.4 Object-Oriented Programming Phase

In the development phase, the front-end of the web application is crafted using HTML5 for structuring, CSS3 for styling, and JavaScript for dynamic interactivity. Visual Studio Code (VSCode) serves as the integrated development environment (IDE), providing a user-friendly platform for code editing, debugging, and version control. It assists in organizing HTML templates for the user interface, applies styles with CSS, and implements client-side functionality using JavaScript. This process involves designing the visual layout, ensuring responsive design, and incorporating user interactions to enhance the overall user experience. Simultaneously, the back-end development is carried out using PHP including the designs and implementation of server-side logic, handling tasks such as user authentication, data processing, and business logic execution. Additionally, MySQL is employed to set up the database, and data access functionality is implemented to enable seamless interaction between the PHP back end and the MySQL database. The integration phase involves connecting the front-end and back-end components, ensuring that data flows accurately between the two layers.

3.5 Object-Oriented Testing Phase

The purpose of object-oriented testing is to verify and validate the functionality and behavior of an object-oriented system, ensuring that the objects, classes, and their interactions function as intended and meet the specified requirements. To make sure the proposed system functions as intended, alpha testing, also known as functional testing, is carried out. The application's modules are separately tested in accordance with the test cases. Iterations between the design, programming, and testing phases occur if deficiencies are discovered during alpha testing and continue until all faults are corrected. When employing object-oriented testing in the development of the proposed system, a requirement traceability matrix (RTM) is an essential tool. It will be developed to establish a link between the test cases used in object-oriented testing and the specific requirements aimed to verify, ensuring comprehensive coverage. By referencing the RTM, the testing process will become more systematic and transparent, enabling a thorough examination of how well AstroWealth aligns with its defined objectives.

3.6 System Development Workflow

There are total of five phases from the object-oriented approach. As shown in **Table 4**, each phase has its own assignment and output that need to produce during the entire project development. Besides that, the output had been completed within the specific days that have been given.

Table 4 *Software development activities and tasks*

Module	Task	Functionalities
Object-oriented requirement analysis (planning)	• Define objectives, problem statements, and project scopes • Identify stakeholders • Identify the hardware and software requirements specifications • Develop project timeline for each period in the development process	• Objectives • Problem statements • Hardware and software requirements specifications • Gantt chart • Software Requirement Specification (SRS) document • Requirements Traceability Matrix (RTM)
Object-oriented analysis	• Identify the requirements • Study the existing systems by determining the limitations, advantages and design ideas used of the existing systems • Compare the existing systems with the proposed system in terms of functionality and design • Identify the methodology • High-level representation of proposed system using UML	• Literature review • Comparison analysis table • User requirements • Functional requirements • Non-functional requirements • Use case diagram • Use case specifications • Sequence diagrams • As-is diagram • Activity diagram

Table 4 *(cont)*

Module	Task	Functionalities
Object-oriented design	• Improve and enhance UML diagrams • Design database • Design and determine the UI/UX	• Database information • User interface design • Class diagram
Object-oriented programming (Implementation)	• Use the software to develop AstroWealth web application • Develop user interfaces and database • Coding implementation using object-oriented programming language such as PHP	• Completion of AstroWealth web application • Front-end and back-end source codes
Object-oriented testing	• Perform Alpha testing • Defects identification and requirements fulfilment checking	• Alpha test results • Test plans and test cases • A complete proposed systems performance validation document

4. Result and Discussion

In this section, the requirement analysis comprehensively covers functional requirements as shown in **Table 5**. This is followed by an in-depth analysis of the UML diagrams and interface design. This analysis provides insights into the specific user interactions, visual components, and navigational pathways.

Table 5 *Functional requirements of AstroWealth*

Module	Functionalities
Register	• The system must provide a secure user registration process allowing users to sign up with first name, last name, email password and their role. • The system must provide options for student or parent as the role. • The system shall not accept duplicated email into the database.
Login	• Users must be able to securely log in to their accounts using valid email and password. • The system should support a password reset feature to allow users to regain access to their accounts.
Manage Profile	• Users should have the ability to view and edit their personal information. • Users should have the option to change password.
Manage Transactions	• Users must be able to add, edit, and delete their financial transactions, specifying details such as amount, date, description and category. • The system should allow users to create custom categories based on their specific financial activities. • Transactions type should be categorized as expense or income. • Users must be able to view their child's or a student's transaction records.
Manage Budgets	• Users should be able to add, edit, and delete budgets for various expense categories • Users must receive notifications or alerts when approaching or exceeding budget limits.
Manage Goals	• Users must be able to add, edit and delete savings goals, specifying the target amount and deadline. • Users must receive notifications or alerts when target amount is reached. • Users must be able to add and delete contributions of a goal.

Table 5 *(cont)*

Module	Functionalities
Manage Students	• Users must be able to add, and delete a student to their account.
Generate Report	• Users should have the ability to view monthly expense reports. • Users should be able to choose the desired month and year. • The system shall display the breakdown of expense categories.

4.1 Unified Modelling Language (UML) Diagram

UML diagrams used in the proposed solution provide a visual representation of the system's structure and behavior. In the context of AstroWealth, UML diagrams, including use case, sequence diagrams and activity diagrams, are employed to illustrate the interaction between the user and the application, as well as the flow of financial processes. A use-case scenario outlines a series of interactions between actors and the system, serving as an abstract template for various scenarios [12]. **Fig. 2** depicts the use case of the web-based application.

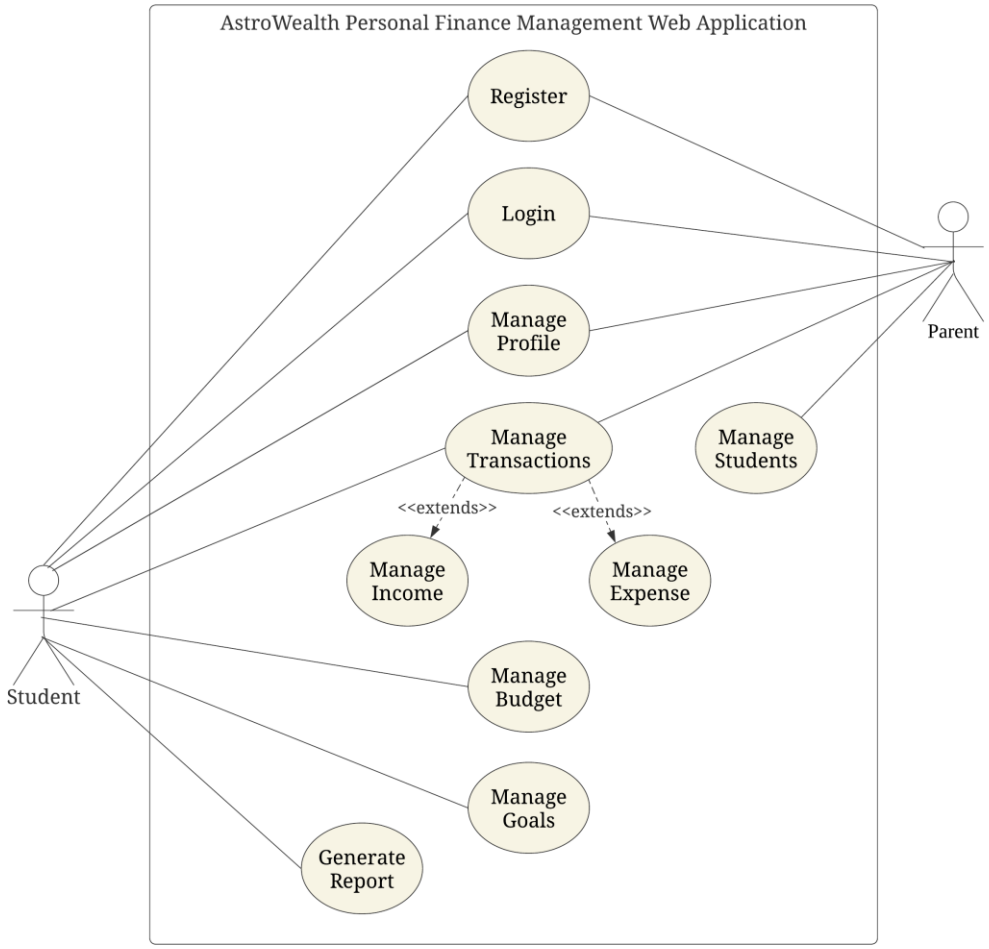

Fig. 2 *Use case diagram of AstroWealth*

The detailed sequence diagrams for each use case depicted in the use case diagram are provided in **Appendix B** for reference and in-depth analysis. Next, the class diagram for AstroWealth will model the relationships and interactions between various classes within the application. This visual representation assists in grasping the structural organization of the proposed solution, aiding in the efficient implementation of AstroWealth's features. In the class diagram, each class which are users, transactions, budgets, goals, and contributions, plays a distinct role in the AstroWealth system, and they are intricately connected. These classes are uniquely identified by the primary keys. It depicts several key relationships that structure the interactions between entities within the system. The users class is central, connecting to multiple other classes through one-to-many relationships. Each user can manage multiple financial transactions, as shown by the link between the users and transactions classes. Similarly, a user can oversee multiple budgets, indicated by the connection to the

budgets class. Financial goals set by users are represented by the goals class, again highlighting a one-to-many relationship, where each user can have several goals. Additionally, each goal can receive multiple contributions, establishing a one-to-many relationship between the goals and contributions classes. This design allows for comprehensive financial management, where users can track and control their transactions, budgets, and goals, contributing towards their financial targets incrementally. **Fig. 3** depicts class diagram of AstroWealth.

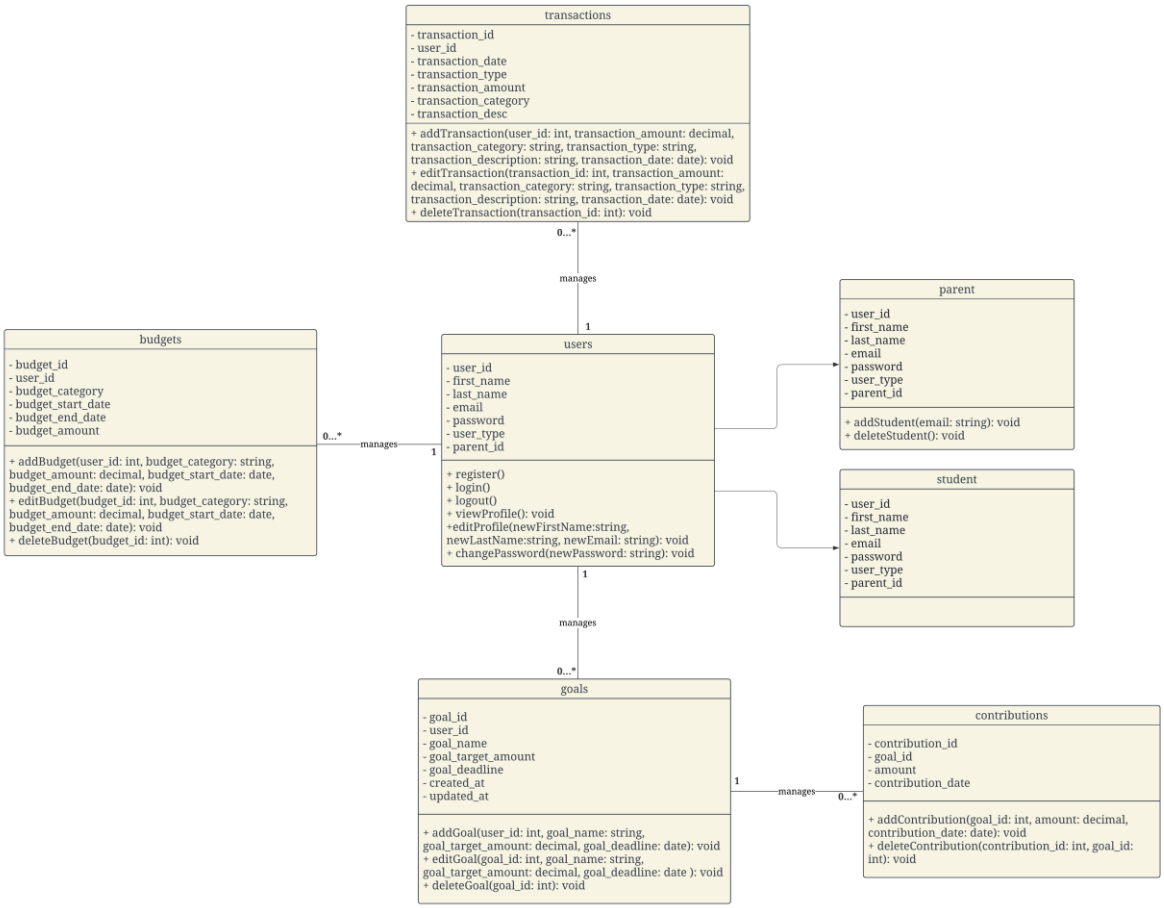

Fig. 3 *Class diagram of AstroWealth*

4.2 Interface Design

The examination delves into the initial draft of the interface design, marking a pivotal step in the progression toward the final design. The interface design process begins with the creation of wireframes, which serve as the initial conceptual sketches or blueprints for the user interface. These wireframes outline the basic structure, layout, and functionality of the interface. This phased approach, starting from wireframes and progressing to a prototype before the development of the complete web application. Due to the paper limitation, only some modules of their visual representations of the interface design can be found in **Appendix C.**

5. Implementation and Testing

The primary goal of the implementation phase is to confirm that the web application is developed in accordance with the analysis and design phase. AstroWealth is created using PHP for logic processing and JavaScript for a dynamic interface, all within Visual Studio Code. Additionally, a MySQL database is utilized to store all data, and XAMPP serves as the local server to access the database.

5.1 System Implementation

Only certain modules are shown due to the limitation of this paper. **Fig. 4** shows the manage transactions module where students can view, add, edit and delete their transactions. Additionally, there are filtering options that can be applied to the transaction table display, allowing users to sort transactions by type (income or expense) and by date (month and year).

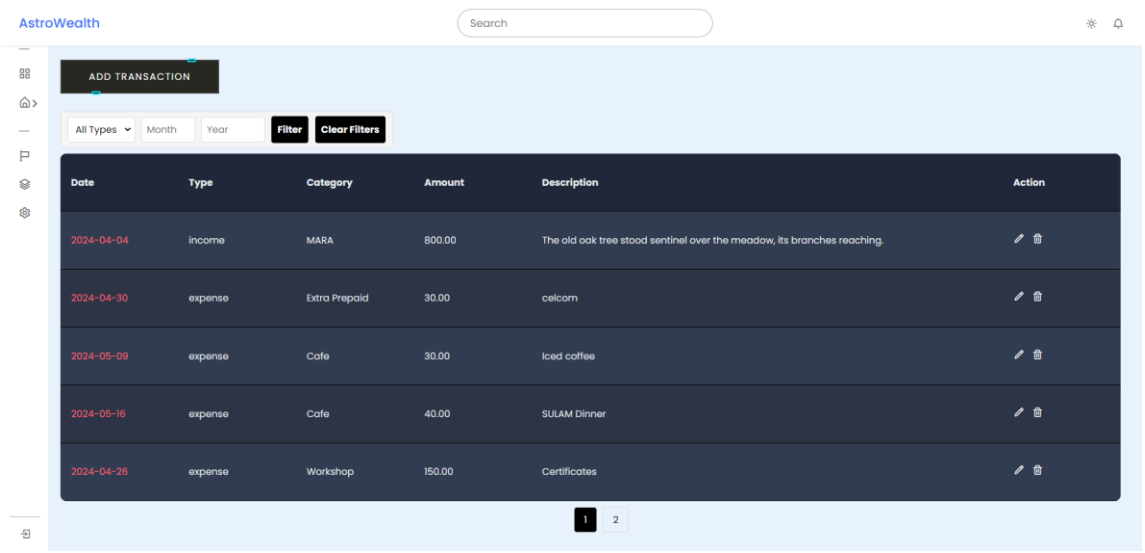

Fig. 4 *Manage transactions interface*

Apart from that, parents can manage transactions, which includes viewing and adding income for their children, directly from the manage students interface. Additionally, from the students table, clicking the view icon displays the student's transaction records. **Fig. 5** and **Fig. 6** shows the add income form and records.

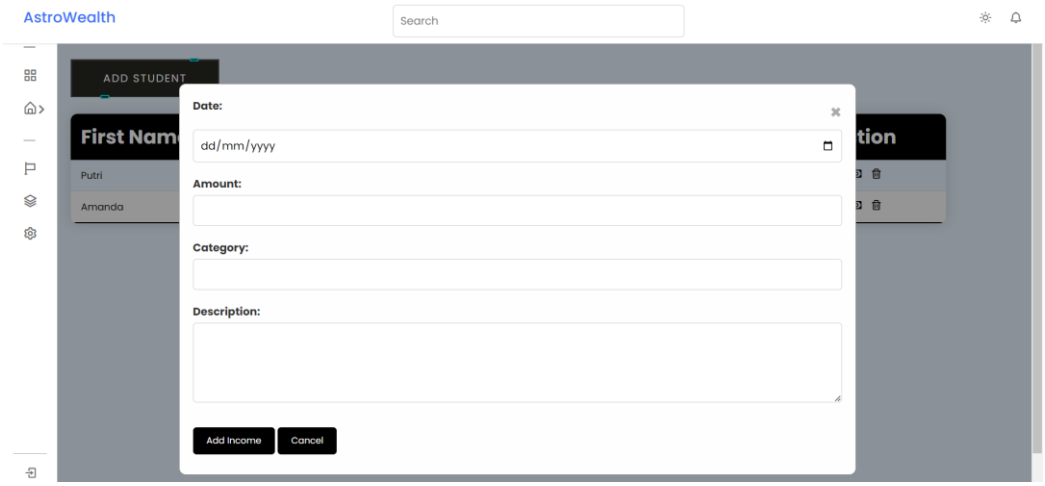

Fig. 5 *Add income form*

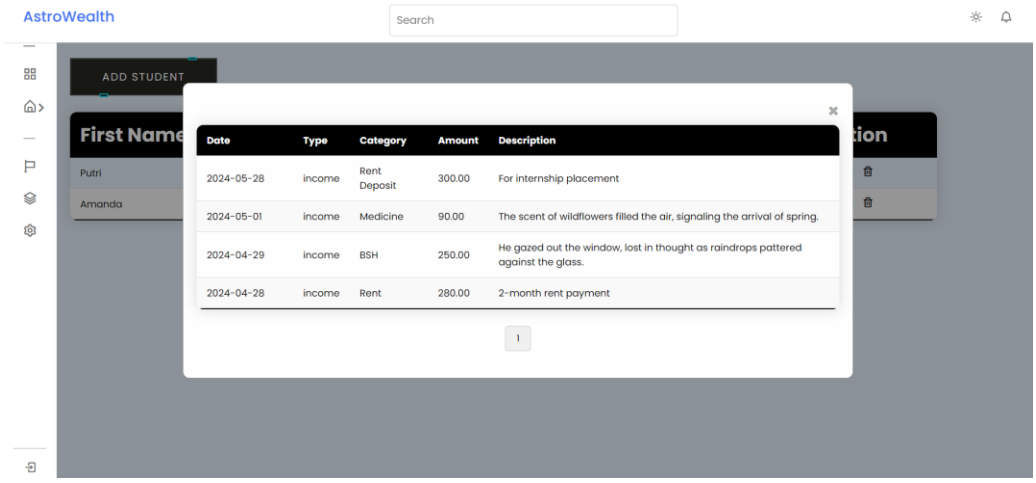

Fig. 6 *Student's transaction records*

Fig. 7 shows the manage budgets interface. Students can add a new budget and it will be displayed horizontally in a box. Each box also features a progress bar to visually represent spending against the budget,

along with action icons for deleting or editing the budget. Additionally, if any budget reaches or exceeds 80% of its allocated amount, a 5-second alert is prominently displayed at the top right corner.

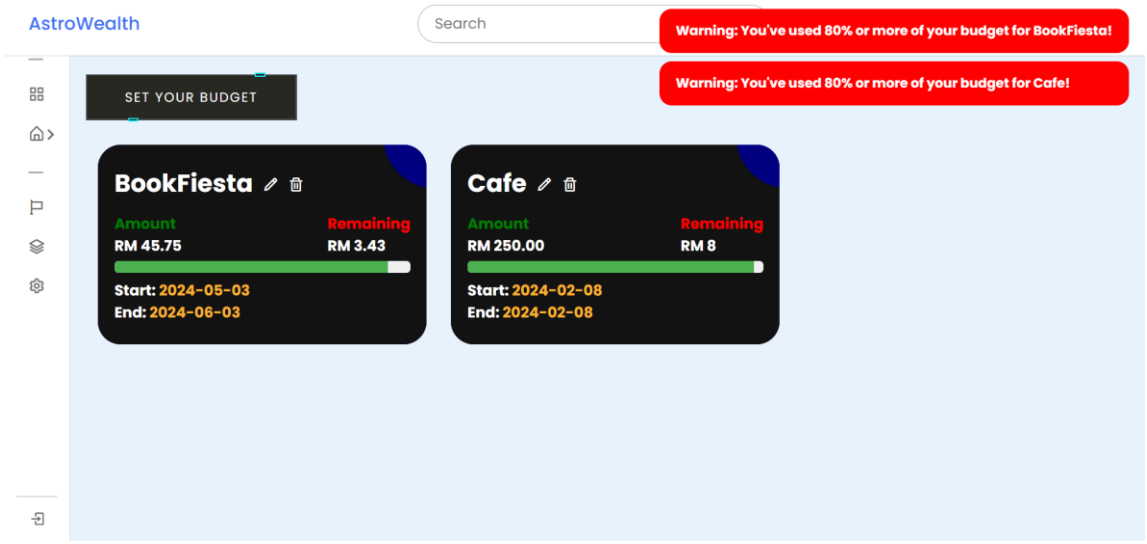

Fig. 7 *Manage budgets interface*

The layout from manage budgets are applied in the manage goals modules to ensure consistency. Additionally, each goal box includes an add icon at the bottom left corner, enabling users to make contributions towards their goals. **Fig. 8 and Fig. 9** shows the manage goals interface and contributions table.

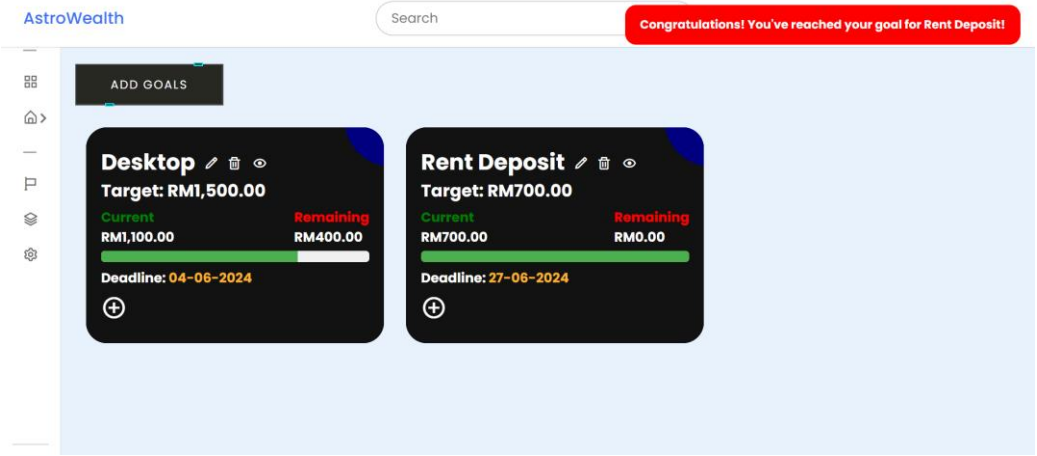

Fig. 8 *Manage goals interface*

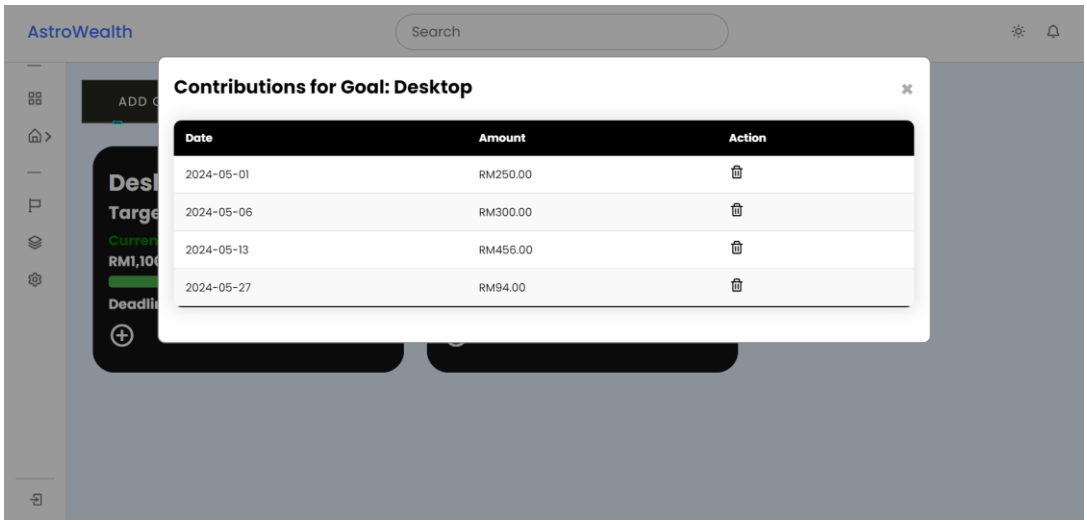

Fig. 9 *List of contributions table*

The manage students interface is tailored to help parents oversee their children's financial activities. Parents add a student by entering the student's email. Once added, students are displayed in a table format, showcasing key information. Each row in the table includes action icons for various tasks: viewing the student's transactions to monitor their spending, adding income to the student's account, and deletion. The deletion function allows parents to unlink or remove the student, thereby revoking their access to the student's financial records. **Fig. 10** and **Fig. 11** shows add student form and manage students interface.

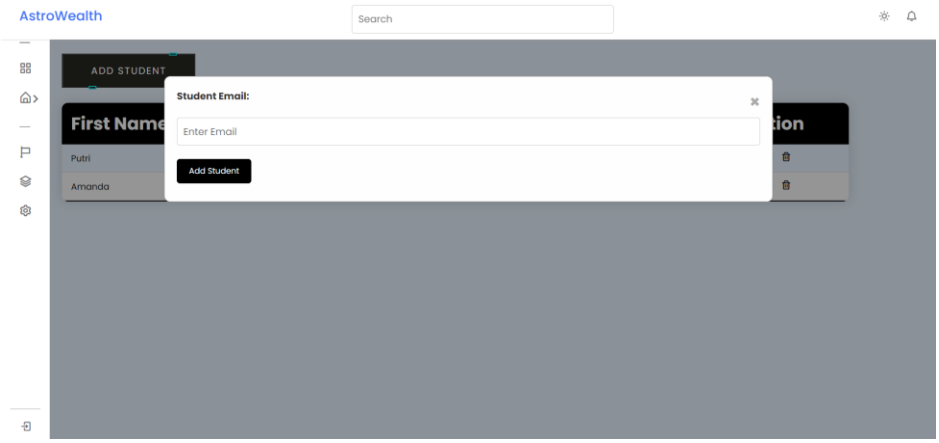

Fig. 10 *Add student form*

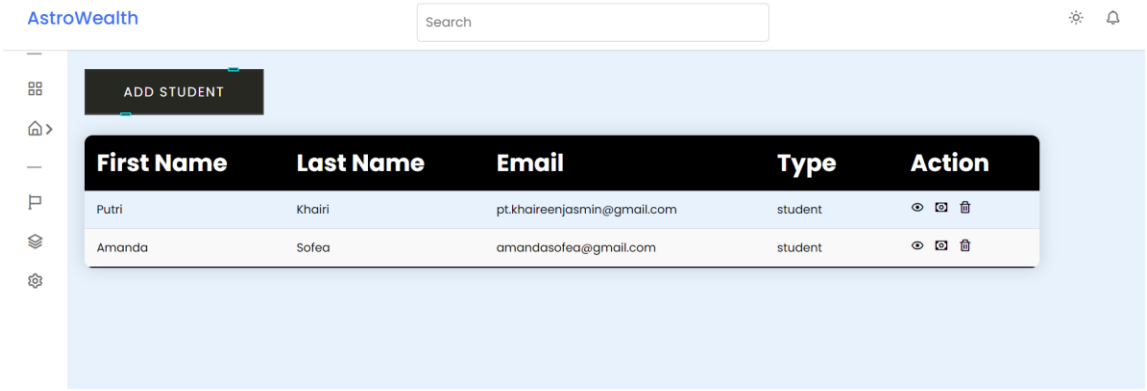

Fig. 11 *Manage students interface*

In generate report module, students can generate the pie chart by selecting the desired month and year. **Fig. 12** shows the generate report interface.

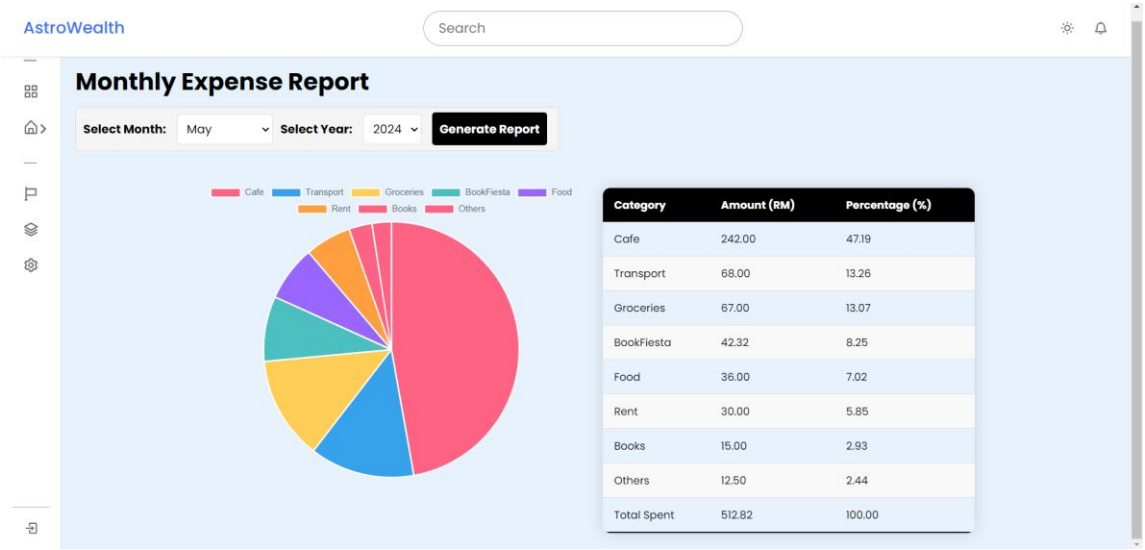

Fig. 12 *Generate report interface*

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 1180-1199

5.2 System Testing

After the web application was implemented, alpha testing was conducted to identify any bugs and errors. Once detected, these issues were addressed following the testing phase. The type of alpha testing employed is functional testing, which focuses on evaluating the core functionalities of the application. This testing approach aims to ensure that the software performs as expected and meets the specified requirements by testing its essential functions and features. **Table 6** displays the test plans and its outcomes.

Table 6 *Test plans*

No.	Test Cases	Expected Result	Actual Result	Status
		Register Module		
1.	Enter valid email address, role, and password	Account is created successfully, and a confirmation message is displayed	As expected	Pass
2.	Enter an email address that is already registered.	Error message is displayed and registration is rejected	As expected	Pass
3.	Incomplete data input	Error message is showed and prompt user to fill in the blank	As expected	Pass
		Login Module		
1.	Enter valid username and password	User is logged in successfully and redirected to the dashboard	As expected	Pass
2.	Enter an incorrect username or password	Error message is displayed and login request is rejected	As expected	Pass
3.	Leave username or password field empty	Error message indicating the field cannot be empty is displayed.	As expected	Pass
4.	Forgot password	Password reset instructions are sent to the registered email address.	Password reset instructions are sent to the registered email address.	Fail
		Manage Profile		
1.	Change personal information	Successful message is displayed, and user information is updated	As expected	Pass
2.	Enter old and new password	Successful message is displayed, and user password is changed	As expected	Pass
3.	Enter wrong old password	Error message is displayed, change password request is rejected	As expected	Pass
		Manage Transactions		
1.	Student navigate to the transactions interface	Transaction interface is displayed	As expected	Pass
2.	Parent navigate to the add students interface	Add student interface is displayed	As expected	Pass
3.	Student click "Add Transaction", fill in transaction details (amount, date, category, description)	Transaction is added and saved successfully	As expected	Pass
4.	Parent click the money icon to add income for a specific student	Income is added and saved successfully	As expected	Pass

Table 6 *(cont)*

No.	Test Cases	Expected Result	Actual Result	Status
5.	Select an existing transaction, edit details	Transaction details are updated successfully	As expected	Pass
6.	Select and delete an existing transaction	Transaction is deleted successfully	As expected	Pass
		Manage Budgets		
1.	Navigate to the budget interface	Budget interface is displayed	As expected	Pass
2.	Click "Add Budget", fill in budget details (amount, start and end date, category)	Budget is added and saved successfully	As expected	Pass
3.	Select an existing budget, edit details	Budget details are updated successfully	As expected	Pass
4.	Select and delete an existing budget	Budget is deleted successfully	As expected	Pass
5.	Enter an expense of a budget category	Budget progress bar and remaining amount are updated	As expected	Pass
6.	Enter the maximum amount of a budget category	System displayed exceeding budget limit alert	As expected	Pass
		Manage Goals		
1.	Navigate to the goal interface	Goal interface is displayed	As expected	Pass
2.	Click "Add A Goal", fill in goal details (amount, target amount, deadline, category)	Goal is added and saved successfully	As expected	Pass
3.	Select an existing goal, edit details	Goal details are updated successfully	As expected	Pass
4.	Select and delete an existing goal	Goal is deleted successfully	As expected	Pass
5.	Enter a contribution of a goal	Goal progress bar and current amount are updated	As expected	Pass
6.	Delete an existing contribution of a goal	Goal progress bar and current amount are updated. Successful deletion message is displayed	As expected	Pass
7.	Enter the maximum contributions of a goal	System displayed target goal reached alert	As expected	Pass
		Manage Students		
1.	Navigate to the add student interface	Add student interface is displayed. List of students added are displayed	As expected	Pass
2.	Click "Add Student", enter student's email	Student is successfully added and linked to the user account. Successfully message is displayed	As expected	Pass
3.	Enter invalid or nonexistent student's email	Error message is displayed, add student process stopped	As expected	Pass
4.	Select and delete an existing student	Successful deletion message is displayed, and list of added students are updated	As expected	Pass

Table 6 *(cont)*

No.	Test Cases	Expected Result	Actual Result	Status
		Generate Report		
1.	Navigate to the report interface	Report interface is displayed with no generated report	As expected	Pass
2.	Choose the month and year	Pie chart is generated accordingly. Breakdown of categories is displayed in a table	As expected	Pass
3.	Choose the month and year with no transaction records	No report is generated	As expected	Pass

5.3 Overall Result

Based on the eight test plans, the overall testing result is largely successful, with the vast majority of test cases passing as expected. The passing percentage is 97.22%. However, there was one failed test case in the login test plan, specifically in forgot password functionality, where the password reset instructions were not received. Despite this single failure, which accounts for approximately 2.78% of the total test cases, the application demonstrates core functionalities and reliability across various aspects, including user registration, login, personal information management, transaction handling, budget management, goal setting, student management, and report generation. These successful tests validate the application's ability to perform essential tasks accurately and efficiently, ensuring a seamless user experience.

6. Conclusion

In conclusion, the AstroWealth presents a personal finance management web application that enables students to efficiently track their expenses and manage their budgets. Furthermore, it allows parents to monitor their children's spending and add income on their behalf, fostering a collaborative approach to financial management. This tool manage to promote financial literacy and responsible spending habits among university students while providing a proactive role for parents in their children's financial education. Furthermore, all of those objectives that were originally mentioned in Chapter 1 of this project were accomplished.

However, despite its numerous advantages, the proposed system also has several limitations that need to be addressed. These disadvantages include limited report export options, lack of report granularity, and limited parental engagement features. Future iterations of the system should prioritize the implementation of additional report export options, such as Portable Document Format (PDF), Excel, or image formats. Next, focus on enhancing report granularity. This may include introducing weekly or yearly breakdowns in addition to monthly reports. Additionally, in order to foster greater parental involvement in their children's financial management, future iterations of the system should aim to expand parental engagement features. This may involve incorporating options for parents to set spending limits, allocate allowances, or receive notifications for significant transactions.

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 contribution to the paper as follows: ***study conception and design:*** *Putri Khaireen Jasmin Ahmad Khairi, Suhaila Mohd. Yasin;* ***data collection:*** *Putri Khaireen Jasmin Ahmad Khairi;* ***analysis and interpretation of results:*** *Putri Khaireen Jasmin Ahmad Khairi, Suhaila Mohd. Yasin****; draft manuscript preparation:*** *Putri Khaireen Jasmin Ahmad Khairi, Suhaila Mohd. Yasin. All authors reviewed the results and approved the final version of the manuscript.*

Appendix A: Gantt Chart

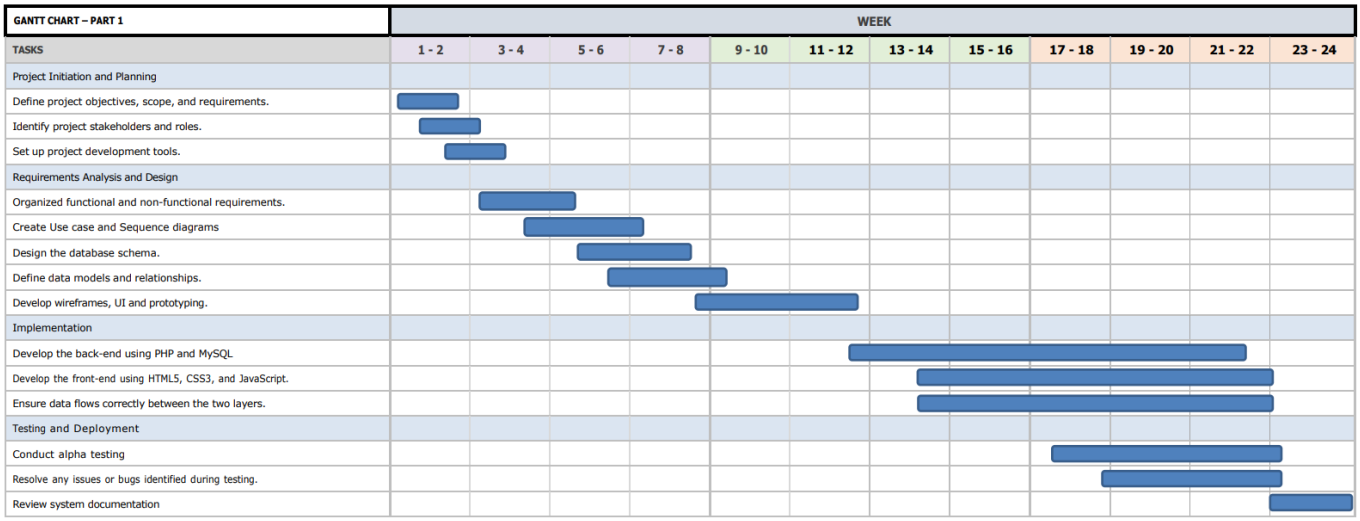

Fig. 13 Gantt chart

Appendix B: Sequence Diagram

The following figures illustrates the sequence diagram of each use cases in the use case diagram.

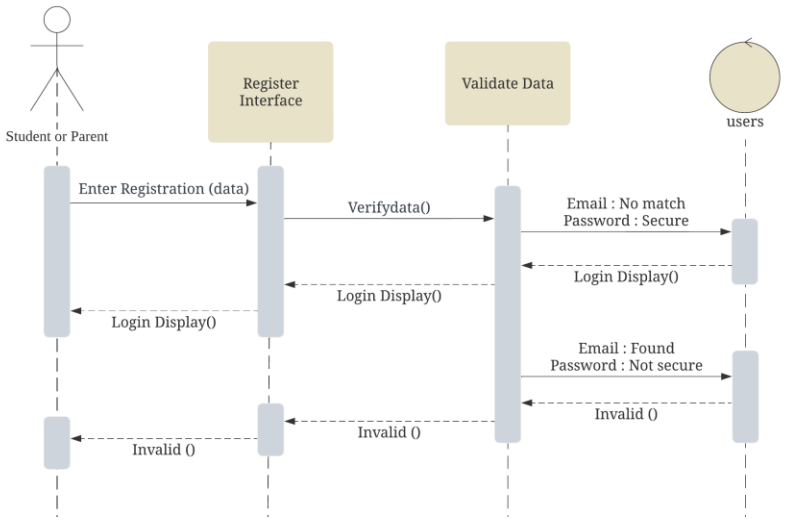

Fig. 14 Sequence diagram of Login

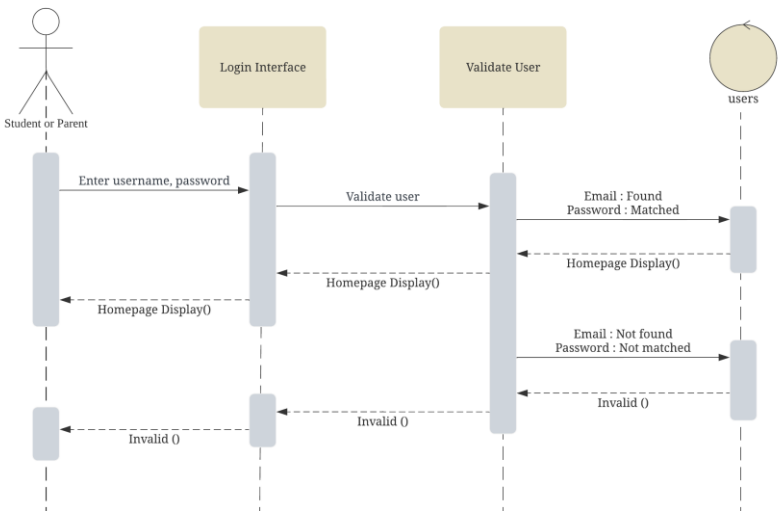

Fig. 15 Sequence diagram of Login

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 1180-1199

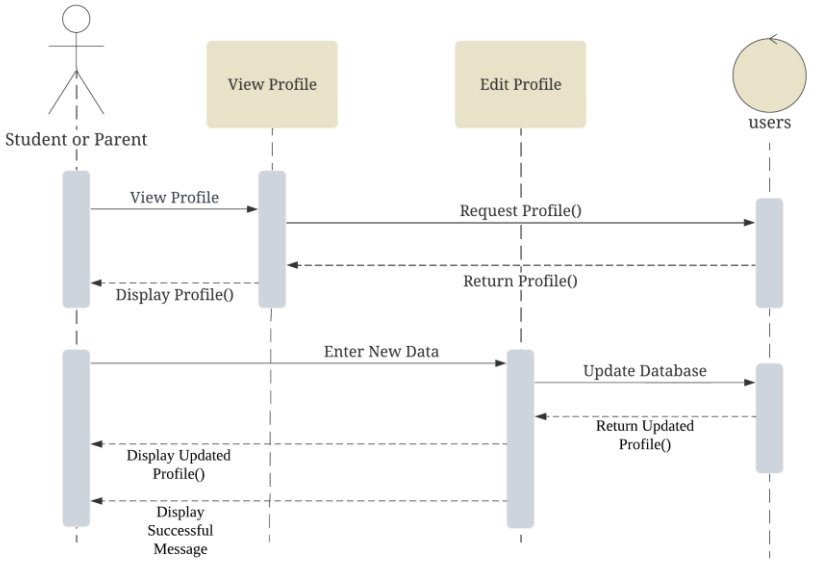

Fig. 16 Sequence diagram of Manage Profile

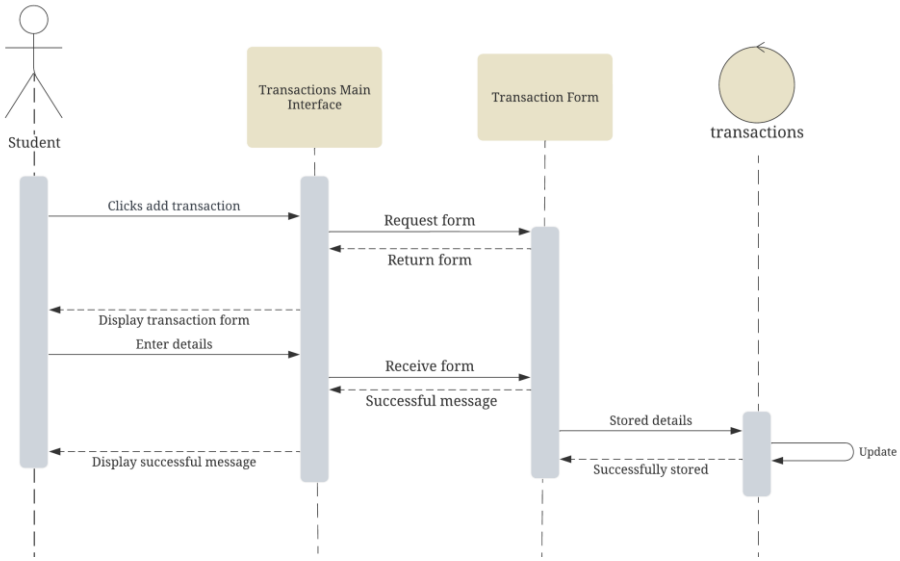

Fig. 17 Sequence diagram of Manage Transactions

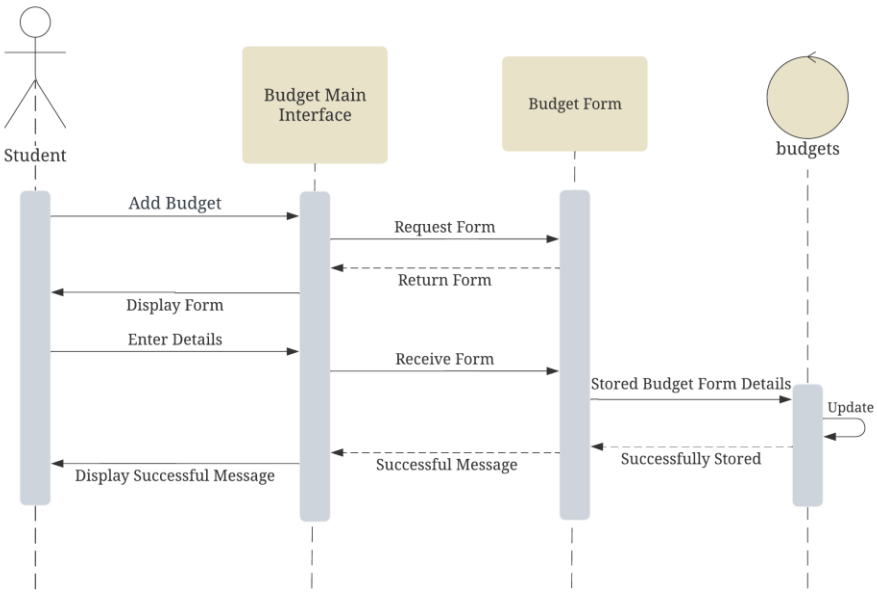

Fig. 18 Sequence diagram of Manage Budget

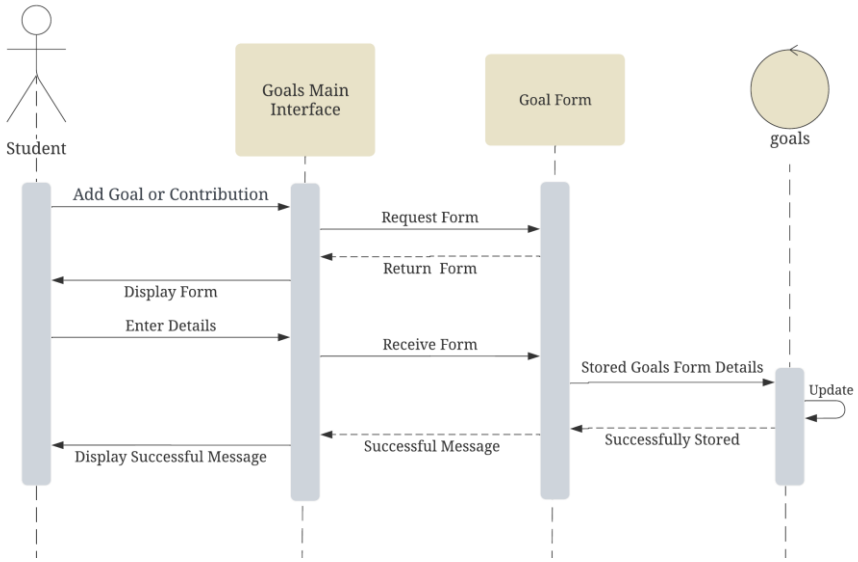

Fig. 19 Sequence diagram of Manage Goals

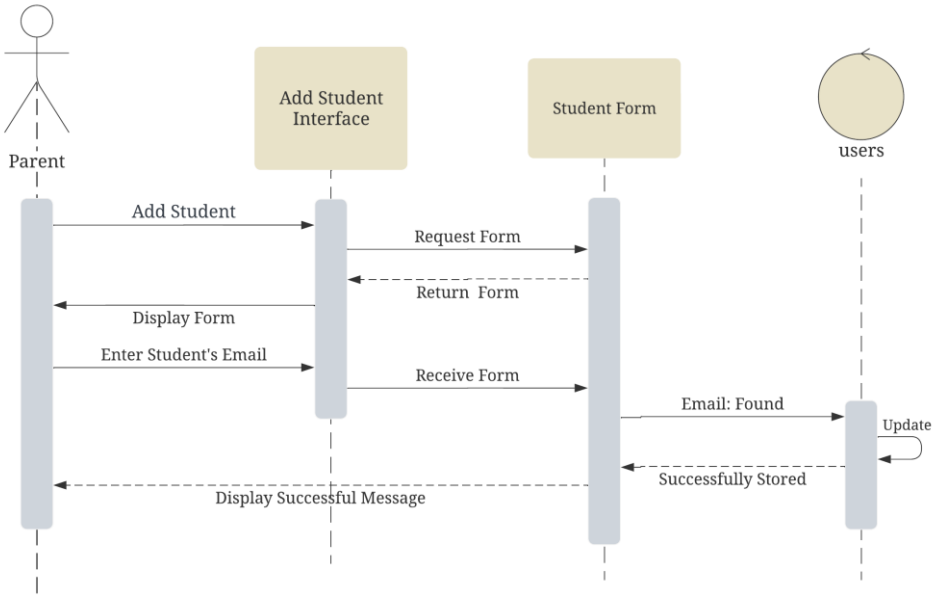

Fig. 20 Sequence diagram of Manage Students

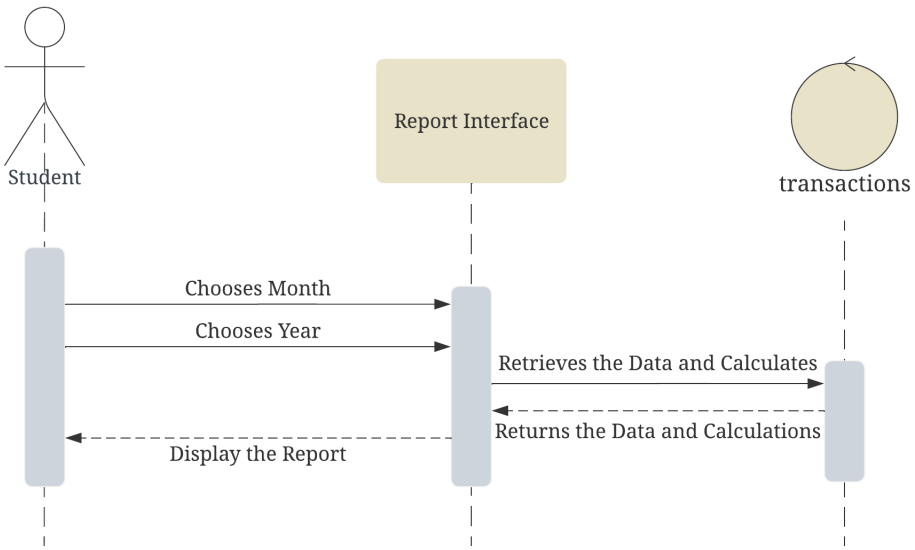

Fig. 21 Sequence diagram of Generate Report

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 1180-1199

Appendix C: Interface Design

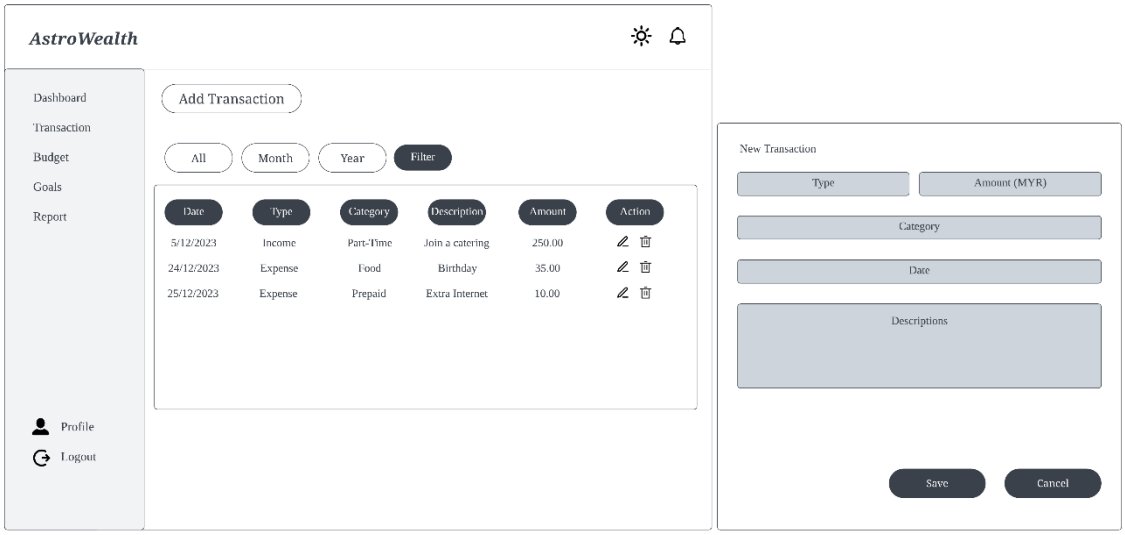

Fig. 22 Manage transactions interface of AstroWealth

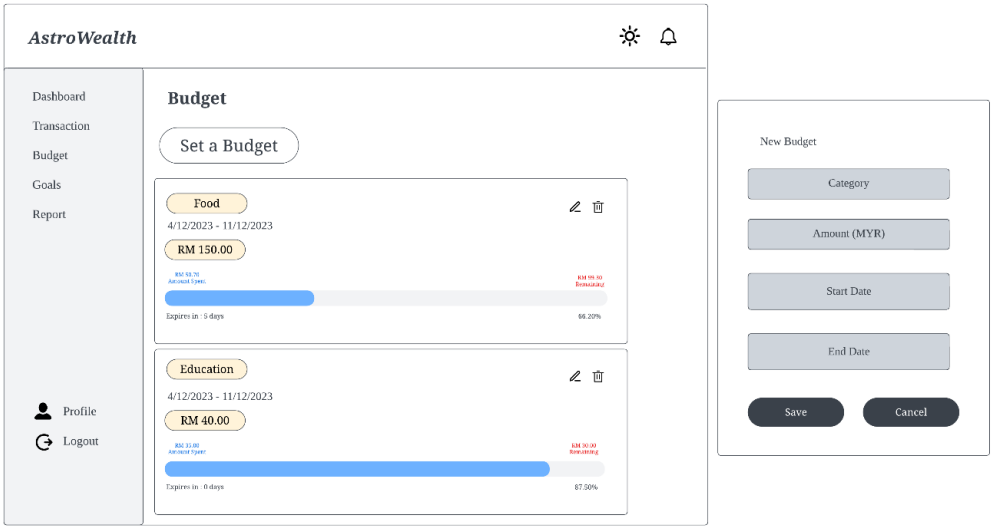

Fig. 23 Manage budget interface of AstroWealth

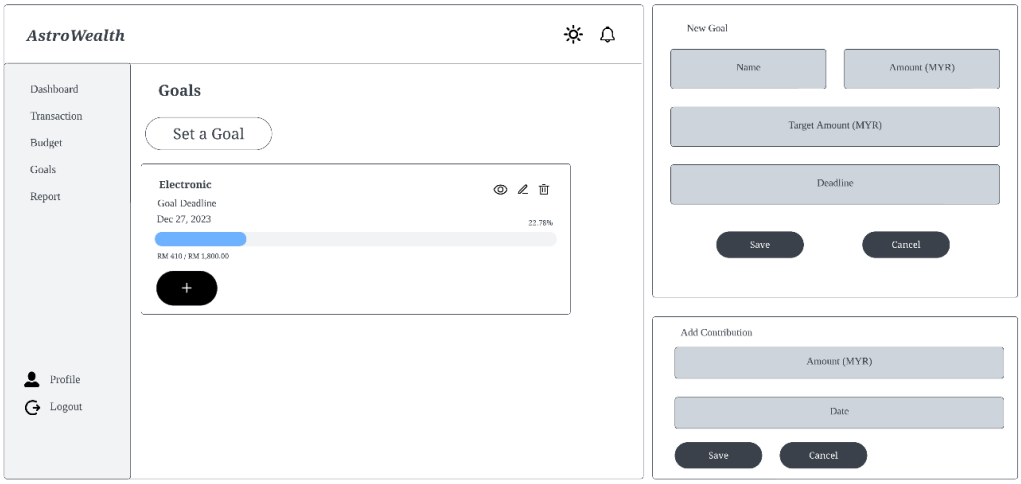

Fig. 24 Manage goals interface of AstroWealth

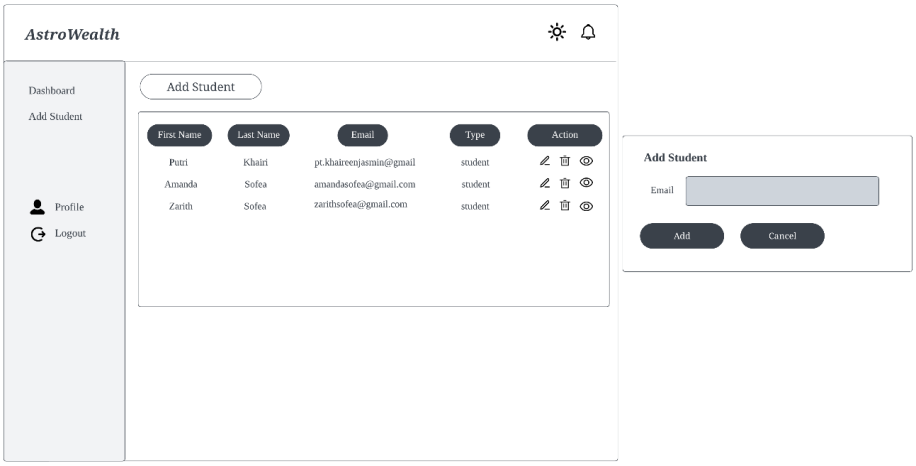

Fig. 25 Manage students interface of AstroWealth

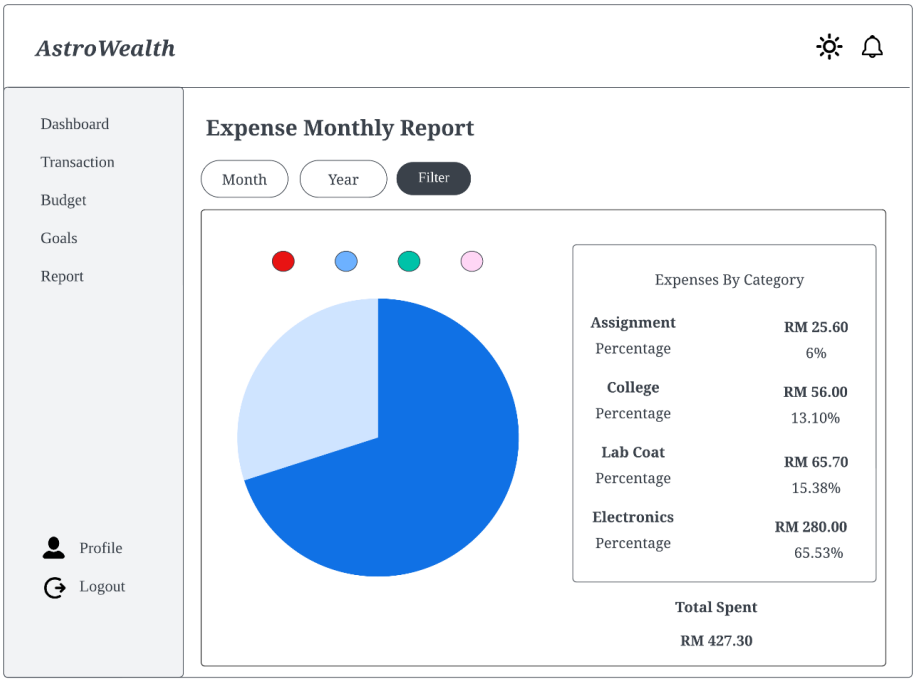

Fig. 26 Generate report interface of AstroWealth

References

[1] E. T. Garman and R. E. Forgue, *Personal Finance,* 12th ed. Boston, MA: Cengage Learning, 2015.
[2] R. E. Goldsmith and E. B. Goldsmith, "The effects of investment education on gender differences in financial knowledge," *Journal of Personal Finance*, vol. 5, no. 2, pp. 55-69, 2006.
[3] B. Kidwell and R. Turrisi, "An examination of college student money management tendencies," *Journal of Economic Psychology*, vol. 25, no. 5, pp. 601-616, 2004.
[4] J. Masud, A. R. Husniyah, P. Laily, and S. Britt, "Financial Behaviour and Problems Among University Students: Needs for Financial Education, *Journal of Personal Finance*, vol. 3, no. 1, pp. 82-96, 2004.
[5] J. M. Norvilitis, M. M. Merwin, T. M. Osberg, P. V. Roehling, P. Young, and M. M. Kamas, Personality factors, money attitudes, financial knowledge, and credit-card debt in college students 1, *Journal of Applied Social Psychology*, vol. 36, no. 6, pp. 1395-1413, 2006.
[6] N. A. Haris and N. Hasim, "PHP Frameworks Usability in Web Application Development," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 3, pp. 109-110, 2019. doi: 10.35940/ijrte.C1020.1083S19.
[7] C. A. T. Batista and D. Batista, Mobills. Retrieved Nov. 12, 2023, from https://www.mobillsapp.com/.
[8] M. C. Chi, Goodbudget. Retrieved Nov. 12, 2023, from https://goodbudget.com/.
[9] D. Ramsey, EveryDollar. Retrieved Nov. 12, 2023, from https://www.ramseysolutions.com/ramseyplus/everydollar.

[10] G. Booch, *Object-oriented Analysis and Design with Applications.* Addison-Wesley Professional, 1994.
[11] Y. Singh and R. Malhotra, *Object-Oriented Software Engineering.* PHI Learning Pvt. Ltd., 2012.
[12] X. Li, Z. Liu, and H. Jifeng, "A formal semantics of UML sequence diagram," *2004 Australian Software Engineering Conference. Proceedings.,* Melbourne, Victoria, Australia, 2004, pp. 168-177, doi: 10.1109/ASWEC.2004.1290469.