

Enterprise Budget and Financial Workflow Management System for the People's Bank

**W.M.L.P Lakmali
2021**



Enterprise Budget and Financial Workflow Management System for the People's Bank

**A dissertation submitted for the Degree of Master
of Information Technology**

**W.M.L.P Lakmali
University of Colombo School of Computing
2021**



Declaration


The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: W.M.L.P Lakmali

Registration Number: 2018/MIT/045

Index Number: 18550451



Signature:

Date: 28.11.2021

This is to certify that this thesis is based on the work of

Ms. W.M.L.P Lakmali

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Mr. Dasun Bamunuarachchi



Signature:

Date: 28.11.2021

Abstract

The Enterprise Budget and Financial Workflow Management System is a solution to the problem of manual processes in People's Bank when achieving internal financial operations. The bank follows manual procedures for budgeting, payments, data storing, and payment approval flows. Therefore, this system aims to turn off manual systems and move to an automated workflow system as possible to make internal processes as effective and efficient. Apart from business requirements, the system provides a user-friendly, flexible environment and better performance to achieve the purpose of the system.

This report included the problem areas and scope of the project by analyzing the requirements and identifying similar systems. This report highlights the current state of manual systems and the appropriate technologies and methods that can be applied to solve problems by automating the process. The report describes the system design represented by including the UML diagram, implementation environment architecture, and design strategy. It also covers development tools and technologies, database schema details, implementation modules, and client and server environmental requirements. The testing approach and evaluation of the developed system are described in this report. The test cases were prepared and run for all the key business scenarios identified. At the end of the report includes a general description of the implemented project and future work that can be done on the solution.

Acknowledgement

I would like to express my heartfelt gratitude to all who have supported me through the project survey. First of all, my supervisor, Mr. Dasun Bamunuarachchi, a lecturer at the University of Colombo School of Computing, sincerely appreciates the enthusiasm, insightful comments, feedback, useful information, practical advice, and valuable ideas that have greatly helped me always in my final project and writing this thesis. Without his support and guidance, this project would not have been possible.

I would also like to thank MIT Project Coordinator Ms. HND Thilini for providing the project schedule, guidance, and her kind support throughout the project. In addition, I would like to thank all the other UCSC lectures for sharing knowledge and guidance with us. They all gave me valuable experience to learn from them. I would like to thank all the facilities and support of the University of Colombo Master of Information Technology Degree Program.

I would like to thank the staff at People's Bank head office for providing the information and guidance needed to complete the project. Special thanks and gratitude to my friends for their utmost support in writing, inspiring me to work towards my goals.

Finally, I would like to especially thank my parents and my siblings for all the sacrifices you made on my behalf. Your prayers for me have supported me so far. Without the patience of their love and encouragement, I would not have been able to complete my project.

Table of Contents

| | |
|---|-----|
| Declaration..... | iii |
| Abstract..... | iv |
| Acknowledgement | v |
| List of Figures | ix |
| List of Tables | x |
| List of Acronyms | xi |
| Chapter 1 - Introduction..... | 1 |
| 1.1 Project Overview..... | 1 |
| 1.2 Motivation..... | 1 |
| 1.3 Statement of the Problem..... | 2 |
| 1.4 Aims and Objectives | 2 |
| 1.5 Background of the Study | 3 |
| 1.6 Scope of the Study | 4 |
| 1.7 Structure of the Thesis | 5 |
| Chapter 2 - Background..... | 6 |
| 2.1 Introduction..... | 6 |
| 2.2 Requirement Analysis..... | 8 |
| 2.2.1 Functional Requirement..... | 8 |
| 2.2.2 Non-functional requirements | 10 |
| 2.3 Review of Similar Systems..... | 10 |
| 2.3.1 SAP – ERP System | 10 |
| 2.3.2 QuickBooks – Finance/Accounting Software..... | 11 |
| 2.4 Related Technologies..... | 12 |
| 2.5 Related Design Strategies | 13 |
| 2.6 Quality of the Solution..... | 13 |
| 2.7 Feasibility Study | 14 |
| 2.8 Summary..... | 15 |
| Chapter 3 - Design Architecture | 16 |
| 3.1 Introduction..... | 16 |
| 3.2 System Architecture..... | 16 |
| 3.2.1 Client Server Architecture | 16 |

| | |
|--|----|
| 3.2.2 Software Design Pattern: Model-View-Controller (MVC) | 18 |
| 3.3 UML Diagrams | 19 |
| 3.3.1 Use Case Diagram..... | 19 |
| 3.3.2 Activity Diagram | 20 |
| 3.3.3 Class Diagram..... | 23 |
| 3.4 Summary | 24 |
| Chapter 4 - Implementation | 25 |
| 4.1 Introduction..... | 25 |
| 4.2 Implementation Environment | 25 |
| 4.2.1 Server Side Environment | 25 |
| 4.2.2 Client Side Environment..... | 25 |
| 4.3 Technology and Development Tools of the Solution | 26 |
| 4.3.1 Technologies | 26 |
| 4.3.2 Development Tools..... | 29 |
| 4.4 Data Flow Diagram..... | 29 |
| 4.5 Database Design..... | 30 |
| 4.5.1 EER Diagram | 30 |
| 4.5.2 Relational Schema | 35 |
| 4.6 Implementation of Modules..... | 36 |
| 4.6.1 Implementation of Reference module..... | 37 |
| 4.6.2 Implementation of Budget module | 37 |
| 4.6.3 Implementation of Transaction module | 38 |
| 4.6.4 Dashboard and Notification | 39 |
| 4.6.5 Reports and Charts Module | 39 |
| 4.6.6 Main User Interfaces..... | 39 |
| 4.7 Summary | 51 |
| Chapter 05 - Testing and Evaluation | 52 |
| 5.1 Introduction..... | 52 |
| 5.2 Test Plan..... | 52 |
| 5.3 Test Scope..... | 52 |
| 5.4 Testing Strategy | 53 |
| 5.4.1 Unit Testing | 53 |
| 5.4.2 System and Integration Testing | 53 |

| | |
|--|----|
| 5.5 Test Cases | 53 |
| 5.6 Evaluation of the System | 65 |
| 5.7 Summary | 66 |
| Chapter 06 - Conclusion | 67 |
| 6.1 Introduction..... | 67 |
| 6.2 Conclusion | 67 |
| 6.3 Limitations and Future Enhancement | 67 |
| 6.4 Summary | 68 |
| References..... | 69 |
| Appendix A - Code Segments..... | 71 |
| A.1 Introduction..... | 71 |
| A.2 Folder Structure..... | 71 |
| A.3 Java Approval Flow Implementation | 72 |
| A.4 JQuery Ajax Implementation | 73 |
| A.5 Method Reuse | 73 |
| A.6 Session Validate | 75 |
| A.7 Maintain Key-Value Pairs..... | 75 |
| Appendix B - Database Table Structure (MySQL)..... | 76 |
| Appendix C - Web Application UI | 77 |
| C.1 Introduction | 77 |
| C.2 URL of the System | 77 |
| C.3 System Messages..... | 77 |
| C.4 Action Code UI | 78 |
| C.5 Budget Code UI..... | 79 |
| Appendix D – Implementation of MIS Reports..... | 81 |
| D.1 Introduction..... | 81 |
| D.2 Payment Summary Report | 81 |
| D.3 Budget Summary Report..... | 83 |
| D.4 Daily Reports | 85 |

List of Figures

| | |
|---|----|
| Figure 2. 1: Existing Workflow Diagram | 7 |
| Figure 3. 1: Web Request - Response Cycle..... | 17 |
| Figure 3. 2: MVC Design Pattern | 18 |
| Figure 3. 3: MVC Implementation..... | 19 |
| Figure 3. 4: Use Case Diagram | 20 |
| Figure 3. 5: Activity Diagram for Budget & Financial Operational Process Flow | 21 |
| Figure 3. 6: Activity Diagram for Financial Department Approval Process Flow | 22 |
| Figure 3. 7: Class Diagram | 23 |
| Figure 4. 1: Data Flow Diagram | 30 |
| Figure 4. 2: EER Diagram-Part 1..... | 31 |
| Figure 4. 3: EER Diagram-Part 2..... | 32 |
| Figure 4. 4: EER Diagram-Part 3..... | 33 |
| Figure 4. 5: EER Diagram-Part 4..... | 34 |
| Figure 4. 6: Login Page..... | 39 |
| Figure 4. 7: Dashboard..... | 40 |
| Figure 4. 8: Notification..... | 40 |
| Figure 4. 9: Dashboard menu - part 1 | 41 |
| Figure 4. 10: Dashboard menu – part 2..... | 41 |
| Figure 4. 11: Supplier/Beneficiary Detail | 42 |
| Figure 4. 12: Supplier Add UI with Validation | 42 |
| Figure 4. 13: Ledger A/C List and Edit UI | 43 |
| Figure 4. 14: Approval Document View/Edit Screen..... | 44 |
| Figure 4. 15: Budget Provision Details..... | 45 |
| Figure 4. 16: Budget Provisioning for a Year..... | 45 |
| Figure 4. 17: Source Document UI – part 01 | 46 |
| Figure 4. 18: Source Document UI– part 02..... | 47 |
| Figure 4. 19: Payment - General Info | 48 |
| Figure 4. 20: Payment - Beneficiary Selection | 48 |
| Figure 4. 21: Payment - Payable/Deduction | 49 |
| Figure 4. 22: Payment - GL entries..... | 49 |
| Figure 4. 23: Payment Explorer | 50 |
| Figure 4. 24: Approval Workflow Table | 50 |

| | |
|---|----|
| Figure 4. 25: Chart - Monthly Budget Utilization | 51 |
| Figure 5. 1: Feedback Summary | 66 |
| Figure A. 1: Project Structure | 71 |
| Figure A. 2: Generate Approval Flow | 72 |
| Figure A. 3: JQuery Ajax Implementation | 73 |
| Figure A. 4: Save multiple records | 74 |
| Figure A. 5: Delete multiple records..... | 74 |
| Figure A. 6: Session Validate | 75 |
| Figure A. 7: Property File | 75 |
| Figure B. 1: Database Tables | 76 |
| Figure C. 1: System URL..... | 77 |
| Figure C. 2: System Messages | 77 |
| Figure C. 3: Budget provisioning validation for duplicate entry | 78 |
| Figure C. 4: Action Code List and Edit Screen | 79 |
| Figure C. 5: Budget Code List and adding UI | 80 |
| Figure D. 1: Payment Summary Filter Criteria..... | 82 |
| Figure D. 2: Payment Summary Report..... | 83 |
| Figure D. 3: Budget Summary Filter criteria | 83 |
| Figure D. 4: Budget Summary Report | 85 |
| Figure D. 5: Daily Report Screen | 85 |

List of Tables

| | |
|--|----|
| Table 5. 1: Test Cases for Manage Approval Document..... | 54 |
| Table 5. 2: Test Cases for Budget function..... | 56 |
| Table 5. 3: Test Cases for Manage Source Document..... | 59 |
| Table 5. 4: Test Cases for Payment Request..... | 61 |
| Table 5. 5: User Evaluation Form..... | 65 |

List of Acronyms

| | | |
|------|---|---|
| API | - | Application Programming Interface |
| CRM | - | Customer Relationship Management |
| DGM | - | Deputy General Manager |
| EER | - | Enhanced entity-relationship |
| EOC | - | Executive Operation Committee |
| GL | - | General Ledger |
| GRN | - | Good Receive Note |
| HOD | - | Head of Department |
| MVC | - | Model-View-Controller |
| PO | - | Purchase Order |
| SAP | - | Systems Applications and Product |
| SDGM | - | Senior Deputy General Manager |
| SIBS | - | Silverlake Axis Integrated Banking Solution |
| UX | - | User Experience |

Chapter 1 - Introduction

1.1 Project Overview

People's bank is a state-owned, major commercial bank in Sri Lanka. The bank offers both retail and corporate banking services, with a network of 739 local branches, service centers, and over 14 million customers. There are more than 30 departments in the People's Bank head office. The finance department is responsible for managing funds within the organization and manages the expenditure of funds on various assets. It ensures efficient financial management and financial control necessary to support all banking business activities. Each financial year starts with by allocation of budget. The project ensures managing budget, manage funds within the organization's departments and automate, payment approval workflow of the bank.

1.2 Motivation

At the beginning of each financial year, budget managers allocate budgets for every department. Any outside expenses in departments should manage within the budget limit. The Department of finance daily receives a considerable amount of payment requests from various departments of People's Bank. These payments include any expenses of departments such as payments for suppliers/vendors, consultant fees, advertising/publicity, visa fee for an official conference, building rentals, or any external payments.

All the departments of the People's Bank engaging in number of financial transactions to achieve the objective of the bank. On such occasions, they need to keep all contracts records with a large number of vendors, suppliers, and beneficiaries. Finally, it will come up with a large number of payments. In the marketing department, production and advertising are major invoices that had to be paid. Since the manual paperwork is going on, there are high time-consuming in the event of proceeding payments. It involves various types of scrutiny work. Presently marketing department engaging with 31 plus media channels and almost all other press institutions. Not only marketing other departments also involve with a huge number of contracts and payments with outsiders. Since bank should precede all the payments on time to receive a better output of them.

Each payment is based on approval document/s which granted the payment. These approvals can be board approvals, EOC (Executive Operation Committee) papers, letters, tenders, and higher management approval such as SDGM (Senior Deputy General Manager) approval, DGM (Deputy General Manager) approvals, HOD (head of department) approvals. These approvals defined two types of payments: one-time full payment or part payments. As a state bank, all transactions should be able to provide any evidence documents such as invoices, receipts, purchase orders (PO), good receive note (GRN), or letters. Since there can be misplaced or can be corrections to such documents, a manual approach can be doubtful about the accuracy of payments.

To get the cash or pay order, each payment request should get approval through higher authorized people. This financial workflow has a bank-specific hierarchy. This is common for all the departments in head office. Each payment should get acknowledge by authorized department's officers and should reach the financial department at the end. Final approval is granted only from the financial department. When the final approval is granted pay order will be delivered or money will be transfer to a customer account. Currently, this workflow is managed manually.

1.3 Statement of the Problem

Usually, a large number of transactions are happening daily in the bank. Hence there should be a way to analyze budget variation throughout the year and also an efficient way to manage all the approvals, payment requests, and payments. There are huge amount of payments are handle within each day. Currently, most of the departments separately maintain excel sheets to manage an internal budget. Further, in order to reduce the time, paperwork, and staff allocation for maintaining manual budget and other financial data recordings and payments approval process, departments of the People's Bank require an automated finance and workflow management system.

1.4 Aims and Objectives

This project aims to automate the finance and workflow management process of People's Bank. The proposed system automates the process of budget initialization to approval flow of all the payments and through that gives a proper solution to the current manual system.

The automation process includes budget defining and allocation, ledger account handling, maintaining approvals granted documents and source documents (invoices/receipts) of payments, documents repository, and approval flow (dynamic) and report module. Each module access is control by the system and maintains security levels. Each department has its own report requirements. So reports will be generating according to budget and finance departments' requirements.

Main objectives of the project

1. To automate payment approval flow. The approval level should be dynamically changed.
2. To Store, all information regarding approval documents and other evidence, support documents of expenses to handle part payment, and also prevent misplacing and damaging of files.
3. To increase the transparency of all expenses, payments, and tracks all the changes during process workflow.
4. Maintaining access levels to protect confidential information
5. To provide an up-to-date status of bank expenses, department budget, pending payment, and unallocated payments
6. Minimize the waiting time and speed up the financial and approval flow.

1.5 Background of the Study

There is no budget or payment handling system in the People's Bank finance department. They use excel sheets to manage budget allocation and utilization. The marketing department of People's bank internally uses a small application to manage the budget for their internal products. In here they mark received invoices on to related budget code. But this process is not centrally visible. Each other departments separately use excel sheets to explore their budget allocations. Otherwise, they can't know the remaining budget for their future payments.

Payment approval flow doesn't have any system and it happened manually (People's Bank, 2020). Each payment requested through a paper document. It contains standard format and prepares it using MS word file for each payment (People's Bank, 2019). Then the request form sends to approval officers in the department and finally should reach to finance

department. Each evidence or related documents should be sent along with the payment request form.

There are some similar systems in the world related to the problem area. Chapter 2 will be illustrating a review of them.

1.6 Scope of the Study

The system will be fulfilling the requirement of the finance department, budget, and other departments. System scope related to the business person is categorized as follows.

- Finance department user:-
 - Ledger Accounts Handling
 - Report generate for analyzing budget and payments
 - Approve or rejects payments which received any departments
- Budget and Budgetary unit user:-
 - Budget Defining
 - Budget Allocation
- Common:-
 - Supplier/Stakeholder Management
 - Approval document and part payment handling
 - Invoices/Source document handling
 - Payment Request
 - Approve or rejects payments in department level
 - Explore payments

Functions within the scope of the system

- Maintaining ledger accounts of the bank. All the double entries should be based on these ledger accounts
- Budget should be defined and assigned for each financial year, before process any payment. The budget should update in real-time when payment is made.
- System should maintain People's Bank supplier base.

- There are approval granted documents are available to make a payment. Such as board approvals, top management approvals, EOC papers, letters, etc. These documents should reflect in the system with relevant to payments.
- Each expense have evidence or source documents such as invoices, receipts, purchase order, good receive note (GRN). These documents should be reflected in the system.
- The finance department receives payment requests from every department. Each outside payment should consist of required details with evidence, approvals, or supportive documents. Each payment request should go through approval flow according to bank regulations. This process should automate and approval flow should be dynamic. According to payment mode, after getting final approval for payment a cheque will be printed (manual process) or/and update GL. This cheque printing and GL update are not implemented in this system scope. Because core integration can't be implemented in this project. However double entries for each payment are stored in the system
- Enable upload facility to scanned copies of any supportive documents.
- Budget summary report and payment summary reports should be produced by the system.
- System notification should be received for pending approvals
- Other features: - User dashboard with graphical representation.

1.7 Structure of the Thesis

This report describes the implementation of implementation-based project development. Chapter 01 contains an introduction and overview of the project. Chapter 02 provides background, requirements analysis, and similar system reviews based on this topic. Chapter 03 described the system design architecture and event flow using UML diagrams graphically. Chapter 04 details the implementation of the system and the methodologies, strategies, and techniques used to develop the system. It includes a choice of implementation environment, hardware infrastructure, database, and application development architecture. The testing and evaluation process is described in Chapter 05. Finally, Chapter 06 described the project conclusions and future work related to the solution.

Chapter 2 - Background

2.1 Introduction

This chapter illustrates a review of similar systems related to the business area. There are some systems developed by international companies which include some functions related to the proposed system. Although we can't match exact requirements, by studying them can be got improvement ideas of system features and design strategies. In the People's Bank, there is no system for budgeting or payment. Currently, they use excel sheets to manage budget allocation. Only the marketing department internally uses a small system called "ADS System" to manage the budget for their internal products. Approval flow doesn't have any system and it happened manually by transferring payment documents.

There are several users engaged in the current manual system. They are,

1. A user for store all manual data and filed the approvals and other supportive documents. Each department has at least one user responsible for these tasks.
2. Manager level users as Officer A and Officer B from each department.
3. Budget managers from the budgetary unit of the finance department
4. Higher-level user (above AGM) for approving higher amounts.

Current manual process and workflow are depicted in the following Figure 2. 1: Existing Workflow Diagram.

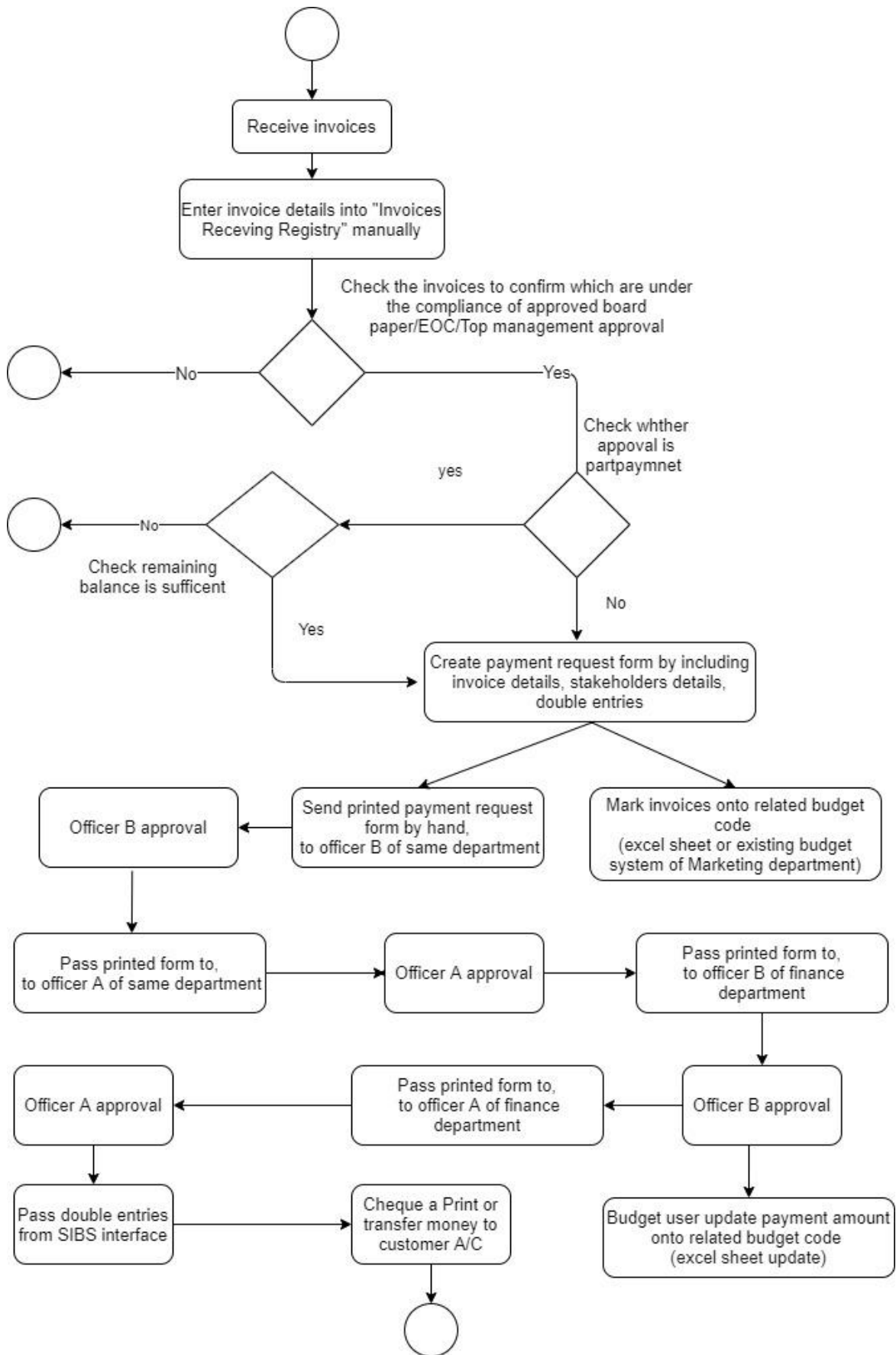


Figure 2. 1: Existing Workflow Diagram

2.2 Requirement Analysis

The proposed system is a solution to the shortcomings identified in the current manual procedure which is shown in Figure 2. 1: Existing Workflow Diagram, explaining that the following functional requirements are for the required system behavior and what the system is supposed to perform. The main functional and non-functional requirements of the developed system are listed below. Requirements are collected from payment managers in finance, marketing, and IT departments.

2.2.1 Functional Requirement

1. Maintain ledger accounts that are used in payments:-

Ledger accounts should be added to the system by financial department users before process any payment from the system. Because double entries should maintain in payments.

2. Budget creation and allocation to each financial year:-

Budget department users should be able to categorize the budget according to each department and allocate the budget for the next financial year. The budget can be updated by an authorized user from the budgetary unit. The available budget should be updated in real-time when payment is made. So that system allows tracking all the budget variations throughout the year.

3. Manage supplier detail:-

The system should maintain the People's Bank supplier base. Users should be able to add any new supplier regardless of whether that particular supplier has an account in the People's Bank or not.

4. Maintains payment approval granted documents and part payment:-

Most payments have prior approvals. All reference detail of approvals should be maintained in the system. Part payment should handle according to the definition of the approval. These approval documents are,

- Board approvals
- Top Management approvals
- SDGM/DGM/HOD approvals

- EOC approvals
- Letters
- Tenders

5. Maintain source documents of payments:-

Each expense has evidence or source documents such as invoices, receipts, purchase order, GRN. Required financial information can be feed to the system such as amount details with VAT, customer details, relevant approval documents, etc.

6. Request payment:-

Any department should be able to request payment from the finance department. Each request should have a request number, particular budget, beneficiary, source documents, amount details with taxes, and double entries. Two payment modes should be handled by the proposed system.

- 1). Pay Order: - Entries will be passing to bank ledger accounts and “Pay Order” will be generated for a supplier.
- 2). Customer Account: - If mode of payment is 'Customer Account' user should enter the customer account number into the system and payment will be credit to the customer account. All expenses are LKR only.

Entries passing to the core system and cheque printing not handle by this system. But payment mode and entries details are stored in the system.

7. Document Upload:-

Document uploading facility should enable for supportive documents. Users can upload the scan copies to the system

8. Dynamic payment approval flow:-

Any payment should have approval flow and authorized users can approve or rejects according to the approval sequence. There are 2 main approval levels maintain within the department called Officer “A” and officer “B”. Each request should reach to finance department for approval. After getting final approval from the finance department, the cheque will be printed (a manual process) and update GL. This cheque printing and GL update do not

implement in the proposed system scope. Because core integration can't implement in this project. This cheque printing happens only for the "Pay Order" payment.

9. Report generate:-

All required reports should be available in the system with the capability of view as an Excel file. Daily reports, budget summary, and payment summary reports should be provided.

10. Generate system notification:-

System notification required for pending approval of payments.

2.2.2 Non-functional requirements

Non-functional requirements are not directly affected by system functionality. However, it can be used to determine the behavior of the system. Non-functional requirements affect the quality attributes of software systems. The non-functional requirements of the developed system are shown below.

1. The system should provide user-friendly and flexible interfaces that are not difficult to learn and use.
2. All data should be accurate since most data depict financial conditions in the bank and the system should maintain consistency.
3. Privacy of information: Only the finance department can view all payment requests. Others can view only their department.
4. The system should have a possibility to adapt change requests.
5. The system should be flexible and portable. So moving from one OS to another does not create any problem.

2.3 Review of Similar Systems

2.3.1 SAP – ERP System

SAP software is a solution is mainly targeting of managing business operations and customer relationships. Developed by a German company. It consists of a number of fully integrated modules that cover all aspects of business management. The purpose is to integrate all the different kinds of modules together. SAP has a wide range of options for customization. When changes are made to one module, the other modules are updated immediately. The

SAP system provides the ability to manage finance, manage assets, and costing, production operations and materials, personnel, plants, and archived documents. SAP produces many standard project reports which used to manage projects. The capabilities of analytics, flexibility, and business integration make the report more versatile than it can meet the needs of both project managers and accounting people. SAP and its support partners have crossed the boundaries of traditional business intelligence (BI) with a wave of new business analytics solutions that revolutionize information (User, 2020).

“SAP includes several modules such as sales management, material management, production planning, financial accounting, and human resources” (User, 2020). However, the SAP ERP solution contains two main types of modules (functional modules and technical modules) to execute transactions that help execute important business processes. However, SAP is primarily used by leading companies that can afford the cost of implementing such high-profile software systems. The complexity is very high. Most companies implement one function of software at a time, allowing employees to become familiar with the system before moving on. The complete implementation process can take several years.

2.3.2 QuickBooks – Finance/Accounting Software

QuickBooks is an accounting software package, developed and marketed by “Intuit”. QuickBooks provides a facility to manage income and expenses and keep track of the financial health of the businesses. Its primary market is small businesses and non-profit organizations. QuickBooks is so simple to use, most tasks can be completed without an in-depth knowledge of accounting procedures (Hammack, 2021). It can use to invoice customers, pay bills, track inventory, generate reports, and prepare for taxes. QuickBook provides up-to-date and accurate information to enable users to make the right commercial decisions. With QuickBooks, can break down the results by day, week, and month for products or customers and also compare with the previous years or with the budget. The product is user-friendly simple interfaces and simple menu structures.

However, there are some concerns regarding QuickBooks. To keep the software up-to-date, there are a number of upgrades that need to purchase. These updates are required once a year. QuickBooks online service offers fewer features than the desktop version of the software. According to clients who use the system, another potential problem is that QuickBooks doesn't have a fixed-asset section. However, Intuit, realizing it and now provides a separate report for listing a company's fixed assets (Hammack, 2021).

2.4 Related Technologies

Over the years technology has been growing fast. It's very important to study and adopt the latest technologies in our application to challenge the current competent world. Below mentioned study of technology area in "SAP" ERP system, "QuickBooks" finance system, and technologies what I should adopt in my project.

SAP ERP system is a very large-scale implementation project. The final price is completed with a total amount of full modules implementation. Which will take up to few years to complete. The cost of deploying and maintaining an SAP solution varies from organization to organization. The SAP system does not require a traditional system upgrade. The technical installation of business features does not affect the behavior of the current system. This means that even after installing the new business requirements, there are no changes to existing features before activating the new features. Installation of new features is decoupled from its activation, allowing organizations to choose which business features to activate (SAP ERP - Wikipedia, 2021). This is a very important feature in modern large systems. Many more new frameworks have been introduced to support developers and architectures.

"The Intuit developers who developed the QuickBooks are following development automation as the trend of technology" (Intuit Developer Community Blog, 2021). Development automation is the automation of manual tasks that are repeated during the software development process and requires less skill and patience. By automating these tasks but performing time-consuming tasks, developers can build, test, and deploy their products efficiently and safely. It's like AI technology. In addition to its on-premises accounting application, QuickBooks provides a cloud-based version that accepts business payments, manages and pays invoices, and performs payroll functions (M. Salter, 2019, pp.6-11).

Intuit go for a Low-code development platform (LCDP) which provides an easy development environment used to create applications through graphical interfaces and configuration (Intuit Developer Community Blog, 2021). So it reduces traditional hand code programming and no need to invest time to be proficient in programming languages.

Frameworks are created to be reused that developers can program applications with less programming. There are many java frameworks. The proposed system will be used java frameworks. Java frameworks include reusable pre-written codes acting as basic templates.

Developers can use them to develop applications by less coding and also customize the reusable components as needed.

2.5 Related Design Strategies

It is good to research how to work popular systems or what are design strategies in the current competing world. Following I have described the design strategies of similar systems in my application area and also what are the design strategies I'm following in my project.

SAP ERP system is a combination of several business components. Such as Business Intelligence (BI), Supply Chain Management (SCM), and Customer Relationship Management (CRM). SAP ERP software component is the primary component and provides capabilities for integrating business data from other application components. SAP ERP system is structured as three-tier client/server architecture. Because it enables high scalability and flexibility. But it provides one tier and two tier deployment also. The heart of the SAP system is in the application tier (application server). The application server gets the input from and displays output to the presentation tier (SAP GUI). Furthermore, it stores data in the database tier (ERProof, 2021).

QuickBooks follows an agile approach to funding system development based on a small development team (Mindfire Solutions, 2021). They always build the application that best suits the customer's budget and interesting business area. The QuickBooks team develops highly dynamic systems with responsive applications that sync with other QuickBooks databases and third-party CRM applications.

2.6 Quality of the Solution

Every software product should consist of some level of custom functionality. It will be easier for the system to maintain and also change request of customers. The "Enterprise Budget and Financial Workflow Management System" is developed as layered architectures using MVC pattern. The application contains several different modules, each module providing its own individual business operation. Modules facilitate development by allowing programmers to focus on only one area of software application functionality. Modules are usually incorporated into the program through an interface.

The system is used by many employees in several departments. Each function should be as easy to use as possible. Every user can access requires modules only. There are more

advantages of this project in contrast to off-the-shelf products which I mentioned above. The project was designed and developed for the People's Bank. So no changes were introduced into the workflow. The bank won't have to change its operational processes. The developer can get a better understanding of corporate culture and business areas. The cost is very high to buy this type of large complex system like ERP. But this project provides what is exactly required by the customer. If the project is built internally, it is easy to provide ongoing support for business people and customization for the implemented system.

2.7 Feasibility Study

➤ **Technical Feasibility:**

The Enterprise Budget and Financial Workflow Management System is a complete web application. The main technologies, techniques, and tools that are associated with the system are,

- Java,
- JavaScript
- JSON
- Ajax
- jQuery
- Bootstrap
- Highcharts
- MySQL database
- Spring, Hibernate and Struts frameworks
- Eclipse IDE
- Apache Tomcat

These technologies and tools are freely available and required technical skills are manageable. Java is the most suitable programming technology for financial systems as it offers greater security and is good for deal with a huge amount of data. Bootstrap responsive structures and styles provide greater support for user interface.

➤ **Resource and Time Feasibility:**

Resources that are required for the developments are a laptop with required software tools. So its clear development resources are available. The deployment required tomcat server and MySQL server available in the People's Bank. Project scope was defined as to complete within 2 semesters.

2.8 Summary

The review of similar systems was worth finding out the latest technologies, new features and learn new ideas which could support the proposed system. The reviewing also could expose the current issues and drawbacks in the related business area. By analyzing the user requirement and find out specific users of the system, their job role in the business area, and required system behavior.

Chapter 3 - Design Architecture

3.1 Introduction

This chapter presents the proposed system design architecture and uses UML diagrams to graphically describe system behavior and event flow. These diagrams helped to understand the exact requirements of the client and support for system implementation. The use case diagram shows the functionality provided by the system and the relationship between stakeholders and the system. The activity diagram shows the operational flow of the system. A class diagram is a static structure diagram that describes the structure of a system by showing the classes, attributes, operations, and relationships between objects in the system. This section details the architecture and design strategy of implementation.

3.2 System Architecture

3.2.1 Client Server Architecture

Client Server Architecture is also known as a networking computing model in which the server hosts, delivers, and manages resources and services to be consumed by the client. The system architecture works on a request-response model. The client of a system sends the request to the server for required function or information & the server responds with it. Clients are situated at workstations or on personal computers, while servers are located elsewhere on the network, usually on more powerful machines. This computing model is especially effective when clients and the server each have distinct tasks that they routinely perform. Many clients can access information on the server at the same time, while the client computer can perform other tasks such as sending e-mail. This is because both the client computer and the server computer are considered intelligent devices (Cio-wiki.org, 2021).

The client server system is more stable and can be scaled as needed. However, in a peer-to-peer network, performance suffers as the number of nodes increases. A client-server system, a dedicated file server, provides a level of access to clients while providing greater security than peer-to-peer systems where security is handled by the system end-user (Cio-wiki.org, 2021).

The architecture of a web application is a mechanism that determines how application components work and communicate with each other. In simply the way the client and the server connection is established by web application architecture. The diagram of client server architecture is illustrated in Figure 3. 1: Web Request - Response Cycle.

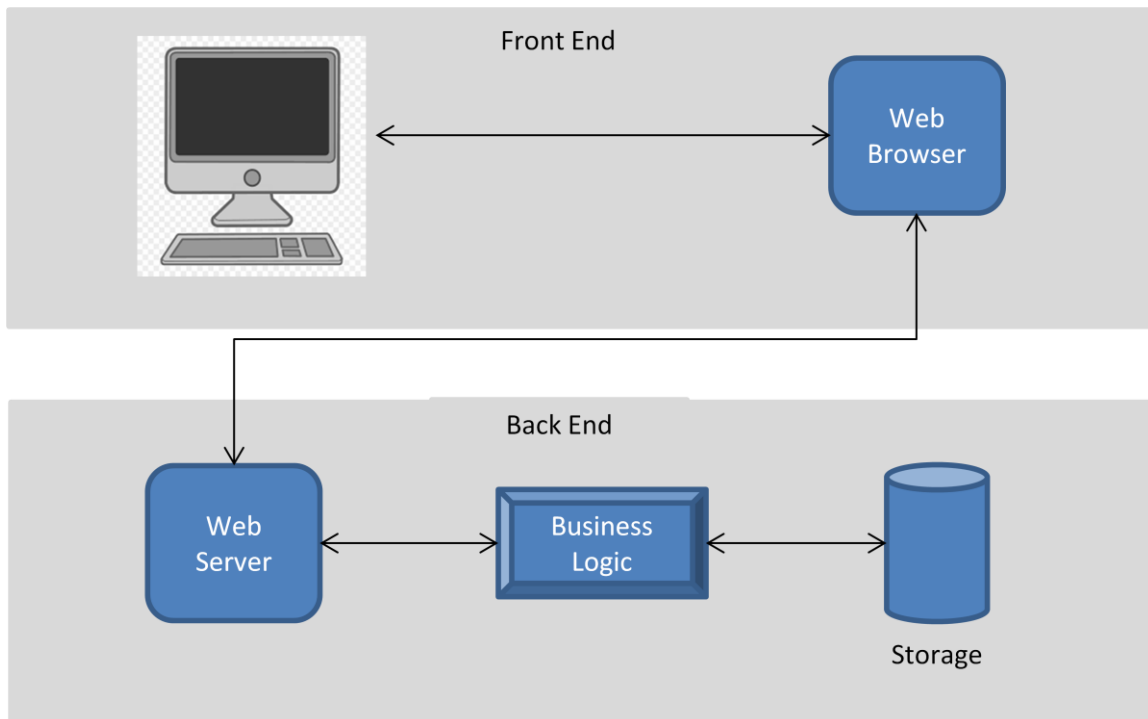


Figure 3. 1: Web Request - Response Cycle

- (1) When the user types in the URL and presses Enter, the browser needs to know the address of the server where the page is located, so the browser prepares to recognize this URL. Therefore, it sends a user request to the Domain Name Server (DNS), which is a repository of domain names and their IP addresses. If you previously accessed the URL from the same browser, it will get the address from the cache. The browser then uses the HTTPS protocol to send a request to the IP address it finds.
- (2) **Server process:** The web server where the URL is located catches the request and sends it to the storage area to retrieve the page and all subsequent data. Its routes are maintained through business logic, managing how each data is accessed and determining this workflow for each application. When the business logic processes the request, the request is sent to storage to find the data it needs (AltexSoft, 2019).

(3) **User receives data:** The user's response returns to the user and the content of the web page can be displayed on the display. The graphical interface that the user can view is called the application front end. All UX and UI components are shown to give users access to the information and features they are looking for.

3.2.2 Software Design Pattern: Model-View-Controller (MVC)

The system development use Model View Controller (MVC) software design pattern. Design patterns state how to write code effectively, easy maintenance, high re-usability, and readability. In MVC the application logic is divided into three components on the basis of functionality. They are,

- **Model:** Represents the business layer of the application
- **View:** Defines the visualization of the application
- **Controller:** Manages application flow. It controls the data flow into model and updates the view according to data changes. It keeps view and model separate.

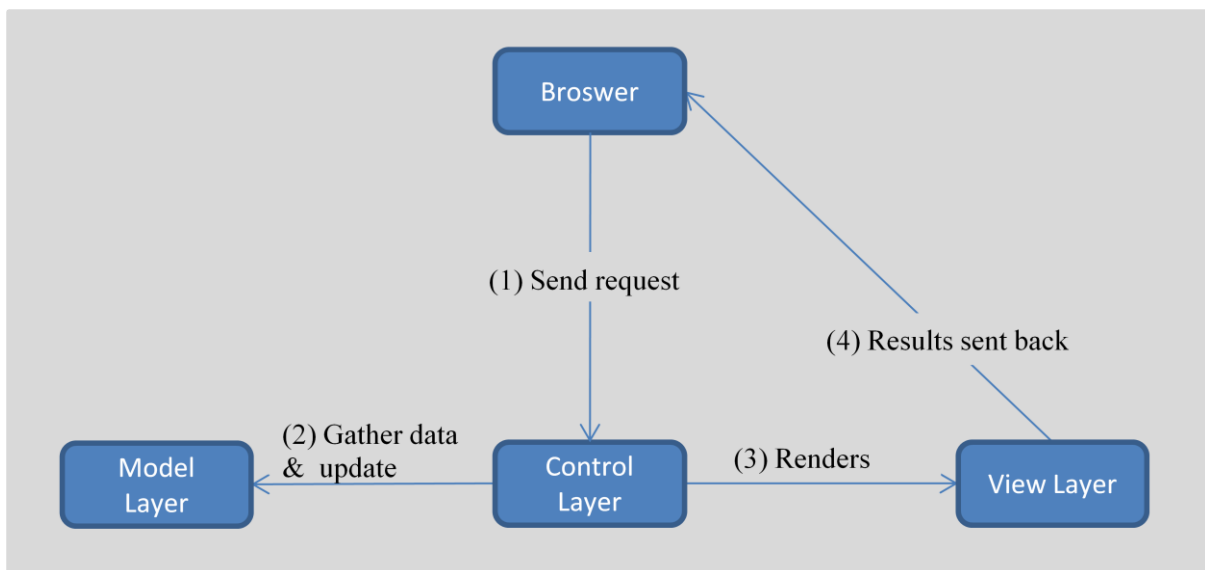


Figure 3. 2: MVC Design Pattern

Hence the system development is based on Java programming and Java EE frameworks context, the Model consists of general Java classes, the View used to display the data and the Controller consists of servlets. There are more Java EE frameworks which support MVC design such as Spring, Struts and Hibernate frameworks. Each user request will be processed as shown in Figure 3. 2: MVC Design Pattern and below Figure 3. 3: MVC Implementation diagram describes how the proposed system implementing it.

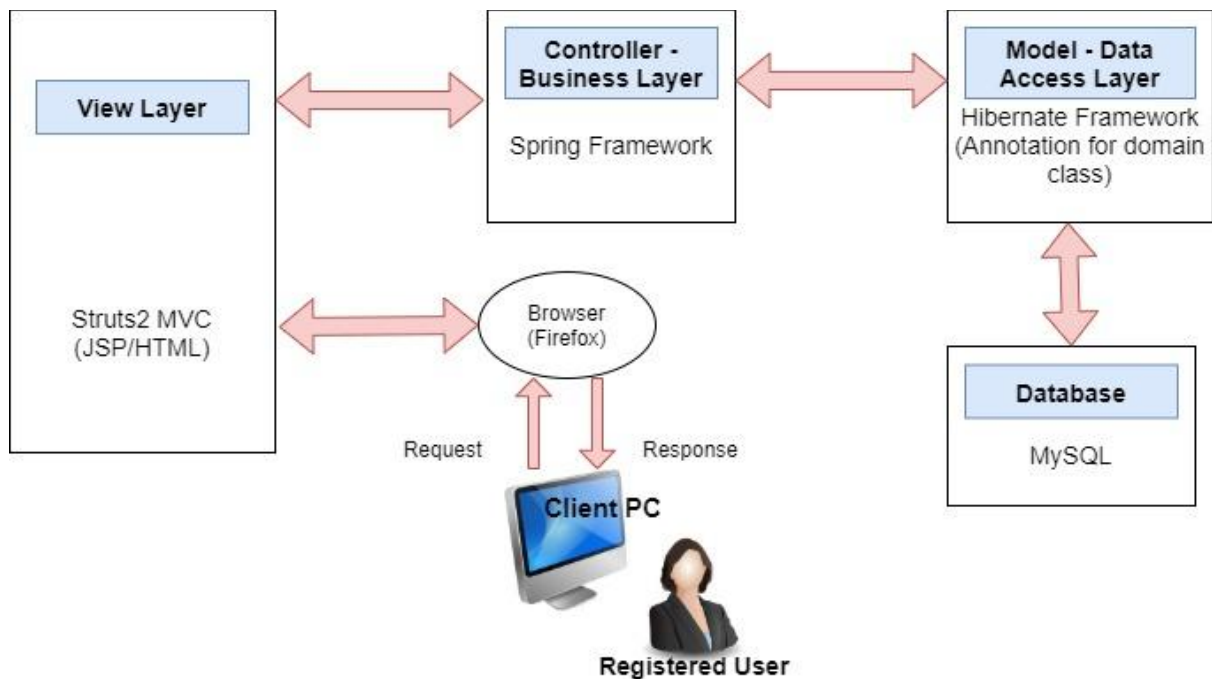


Figure 3. 3: MVC Implementation

The steps of how user request are processed described in below.

1. View forwarding the request to the business layer (controller) which does business processing and displays it to the system user.
2. The controller performs the action by invoking the model and then collects the required data to respond to the request.
3. The controller then passes the retrieved data to the view
4. The view is rendered gathered data and sent back to the client for the browser to display

3.3 UML Diagrams

3.3.1 Use Case Diagram

Use case diagrams provide the main functionalities of a system using actors and use cases. The use case diagram of the system is illustrated in Figure 3. 4: Use Case Diagram.

(1) Activity diagram for budget, approvals, sources (invoices, PO, etc.), payment process, and approval flow within departments

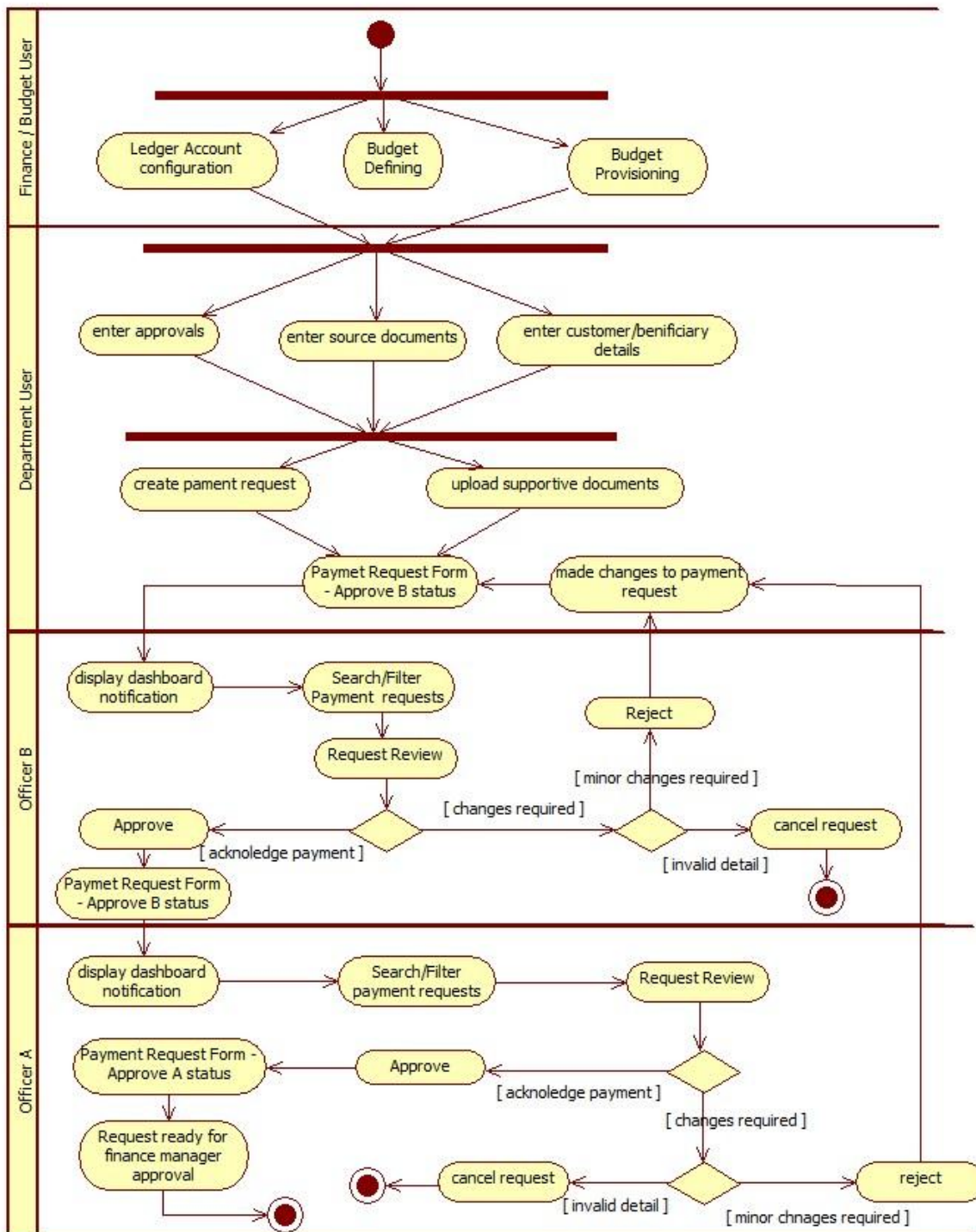


Figure 3. 5: Activity Diagram for Budget & Financial Operational Process Flow

(2) Activity diagram for financial department approval process flow

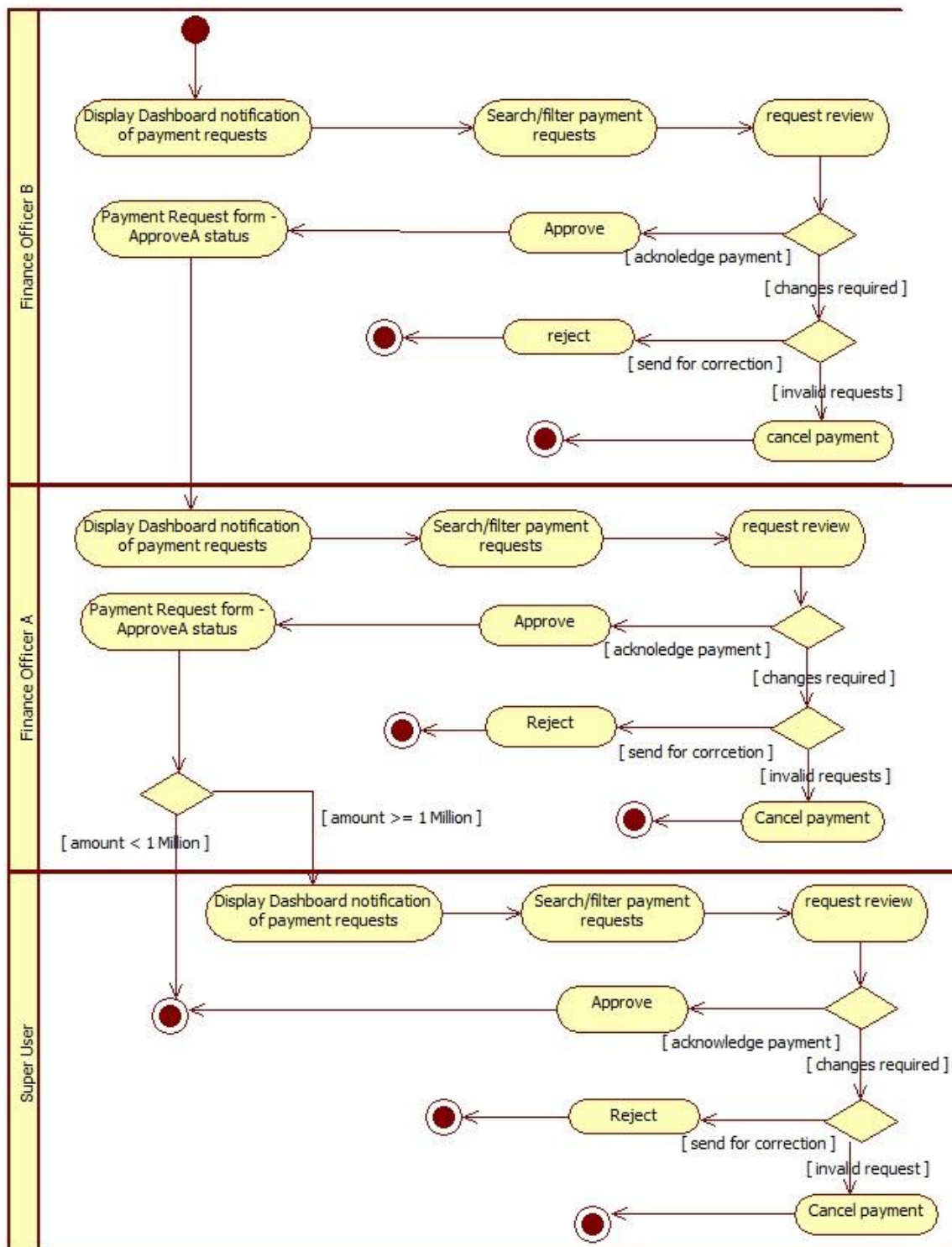


Figure 3. 6: Activity Diagram for Financial Department Approval Process Flow

3.3.3 Class Diagram

“A UML class diagram is a type of static structural diagram that describes the structure of a system by showing the classes of the system, their attributes, operations (or methods), and the relationships between objects” (Visual-paradigm.com, 2021). The class diagram of the system is illustrated in the following Figure 3. 7: Class Diagram.

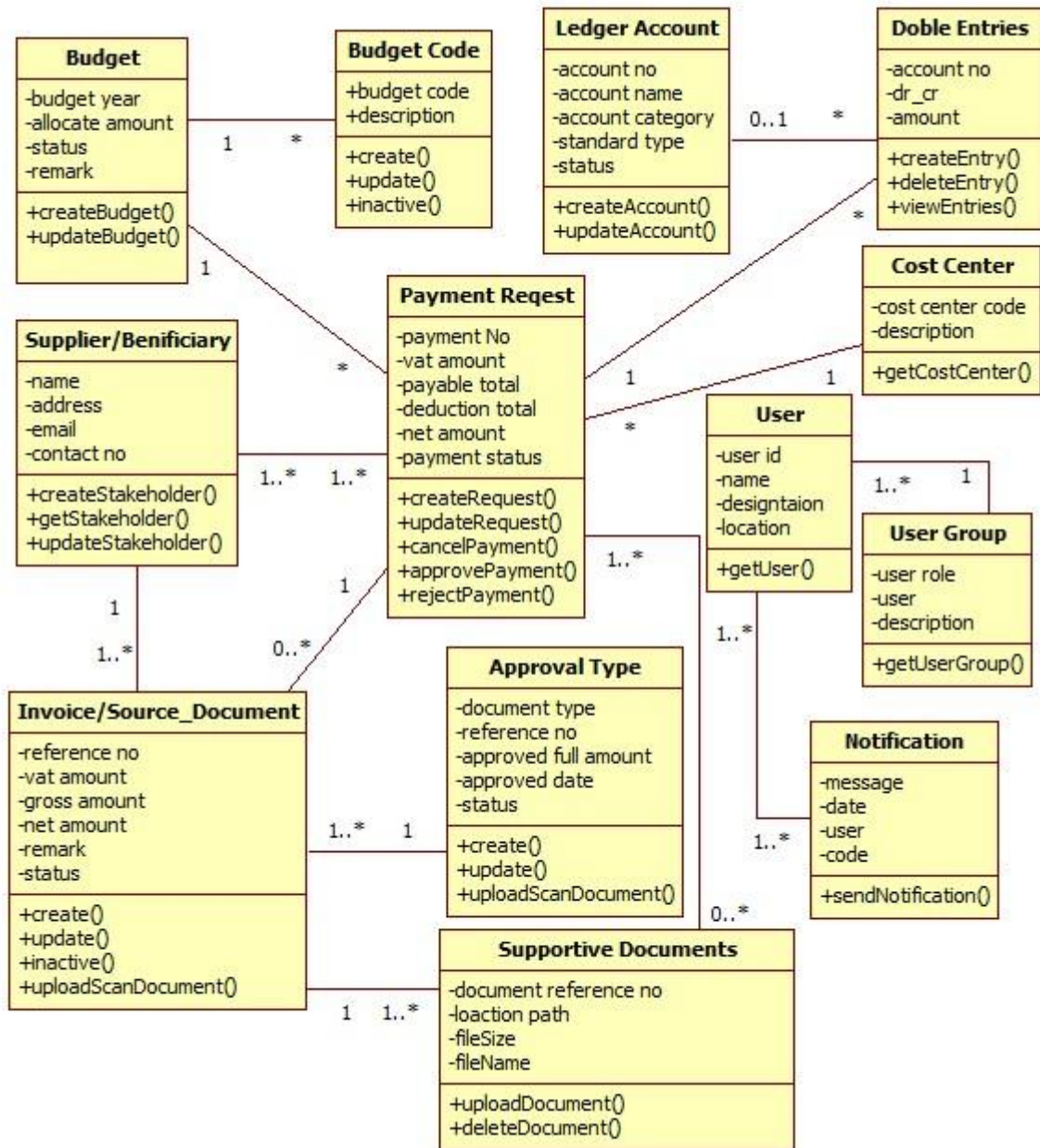


Figure 3. 7: Class Diagram

3.4 Summary

This chapter deals with the design aspects of the project using UML diagrams. These diagrams are used full to identify system events, flows, relationships, and overall behavior. The network architecture of the software model and development design pattern used to implement the system are described in detail.

Chapter 4 - Implementation

4.1 Introduction

This chapter contains system implementation details and describes the methodologies and techniques used to develop the system. Different methods, tools, and technologies were used to account for complexity, the importance of business requirements, data capacity, improved system quality, attractiveness, and efficiency. This section details the implementation environment, hardware infrastructure, development strategy, development tools, database, and application development architecture.

4.2 Implementation Environment

This web-based system has been developed as a solution to the inefficiencies and problems of manual processes related to budget and payment process and approval flow issues. Server environment and client environment are the main categories in hardware infrastructure.

4.2.1 Server Side Environment

When this solution is developed as client-server architecture-based centralized web application, following are the minimum hardware and software requirements recommend for hosting server-side applications hardware and software requirements.

- Server - xeon 4 core 2.8 MHz processor ,Windows 2012/2016
- RAM 64GB
- Hard Disk –500 SSD Sata for log and backup 1 TB
- 1Gbps network two interfaces
- Java
- MySQL

4.2.2 Client Side Environment

Following are the list of hardware and software configuration for client-side environment,

- Intel Core I5 Processor
- OS Windows 7/10

- 4GB RAM
- 250MB HDD
- Network or WiFi
- The latest version of Firefox

4.3 Technology and Development Tools of the Solution

To develop a web-based solution for the system following technologies and tools were used.

4.3.1 Technologies

4.3.1.1 Object Oriented System Development

In an object-oriented approach, a system is defined as a collection of objects that work together to perform a task. Here, the objects take action when requested, and each object maintains its own specific data. There are many object-oriented languages, but few major commercial languages, so Java was used as the programming language.

Object-oriented design improves software maintenance, reusability, and changeability. Some object-oriented concepts are encapsulation, inheritance, and polymorphism. Objects that represent in an object-oriented system fall into main three categories of classes. They are the problem domain class, the GUI class, and the data access class.

4.3.1.2 Web Technology with Java EE

By using JAVA language “The Enterprise Budget and Financial Workflow Management System” is developed. Java 11 is used for developing the application. Java is a widely used programming language and also a computing platform. A platform is the hardware or software environment in which programs run. Java provides the capability to write standalone applications as well as web application. Most of the current web applications are now getting implemented in Java. There are several Java platforms. Any Java platforms consist of a Java Virtual Machine (VM) and an application programming interface (API).

Java SE is the standard version and contains all the base classes needed to create single-user applications when the Enterprise Edition is set up for multi-tier enterprise applications. A Java web application is a collection of dynamic resources such as Servlets, Java Server Pages,

Java classes and jars, and static resources (HTML pages and images). Java web applications can be deployed as ".war" files. A ".war" file is a zip file that contains the complete content of the corresponding web application.

Why Java EE?

Java EE is portable across different platforms such as Windows, Linux, and Mac. JavaEE simplifies enterprise applications by defining and specifying a complete set of common standard services such as naming, concurrency, transaction management, security processing, and database access (IBM Docs, 2016). There are numerous java EE support frameworks and tools available, including the Struts framework, Spring framework, Hibernate framework, Java Server Faces (JSF), Google Web Kit framework, NetBeans, Eclipse, and more. Another advantage is that user doesn't have to worry about paying licenses because they are all open source.

- **Hibernate framework**

Hibernate is an object-relational mapping library for java. It is designed to map objects to a relational database and to implement the object-oriented programming concepts in a relational database. It does simplify the database interaction of the application.

- **Spring framework**

The Spring framework provides a comprehensive programming and configuration model for modern Java-based enterprise applications - on any kind of deployment platform (Spring.io, 2021). Spring framework can use to create high-performing, easily testable, and reusable code.

- **Struts framework (Apache Struts 2)**

Apache Struts is an open-source framework that is used to create elegant modern Java web applications. The Struts framework is a web tier type application framework. Struts applications are hosted by web containers, so need to use the services provided by the container, such as handling requests over HTTP and HTTPS protocols. For web applications that have a large scope and run on multiple JSPs, so a large number of requests and responses are received or sent from the server. That makes the controller (servlet) very complicated. In

that case, Struts is used. Struts has built-in Jakarta Servlets (action Servlets) that process these and control the navigation flow, reducing overhead and simplifying configuration.

- **JXLS library**

This library provides the possibility of generating excel templates by including some special logic. Not like other static report templates, this is simpler to adopt changes when a user requested.

4.3.1.3 Client-Side Technologies

- **Ajax**

AJAX is a very popular technique for creating better, faster, and more interactive web applications. With Ajax, web applications can send and retrieve data from a server asynchronously without reloading the existing page.

- **JavaScript**

JavaScript is a lightweight scripting technology that helps to make websites more interactive. JavaScript implementations are used as client-side scripts to interact with the user control the browser and alter the document content that is displayed.

- **Bootstrap**

Bootstrap is a free and open-source, surprisingly powerful, and effective front-end framework for building responsive websites and web applications. It is a consistent framework that supports all major browsers and CSS compatibility fixes. Customized Bootstrap styles are used to build user interfaces in this project.

- **Highcharts**

Highcharts is a JavaScript software library used for visual representation of business information.

4.3.2 Development Tools

- **Eclipse**

The Eclipse IDE is famous for Java Integrated Development Environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. So I used Eclipse IDE (2021-03 (4.19.0)) for development.

- **Apache Tomcat**

Apache Tomcat is used to deploy Java Servlets and Java sever pages. So in Java project can build WAR file, and drop it in the deploy directory in Tomcat.

- **MySQL**

MySQL was used as a database management system for developed the system. MySQL is a popular choice of database for use in web applications. It is very easy to install, and setting up implementation is a relatively simple task. Furthermore, it's an easy database to work with and compatible with virtually every operating system.

- **HeidiSQL**

HeidiSQL is used as an administration tool for MySQL. HeidiSQL provides a graphical interface for managing MySQL databases. It is used for query and edit data, create and update tables, views, procedures, triggers, and scheduled events.

4.4 Data Flow Diagram

A Data Flow Diagram shows a visual representation of the information flows within a system. The data flow between processes and integration of separate modules are depicted in the following diagram (1-level DFD) Figure 4. 1: Data Flow Diagram.

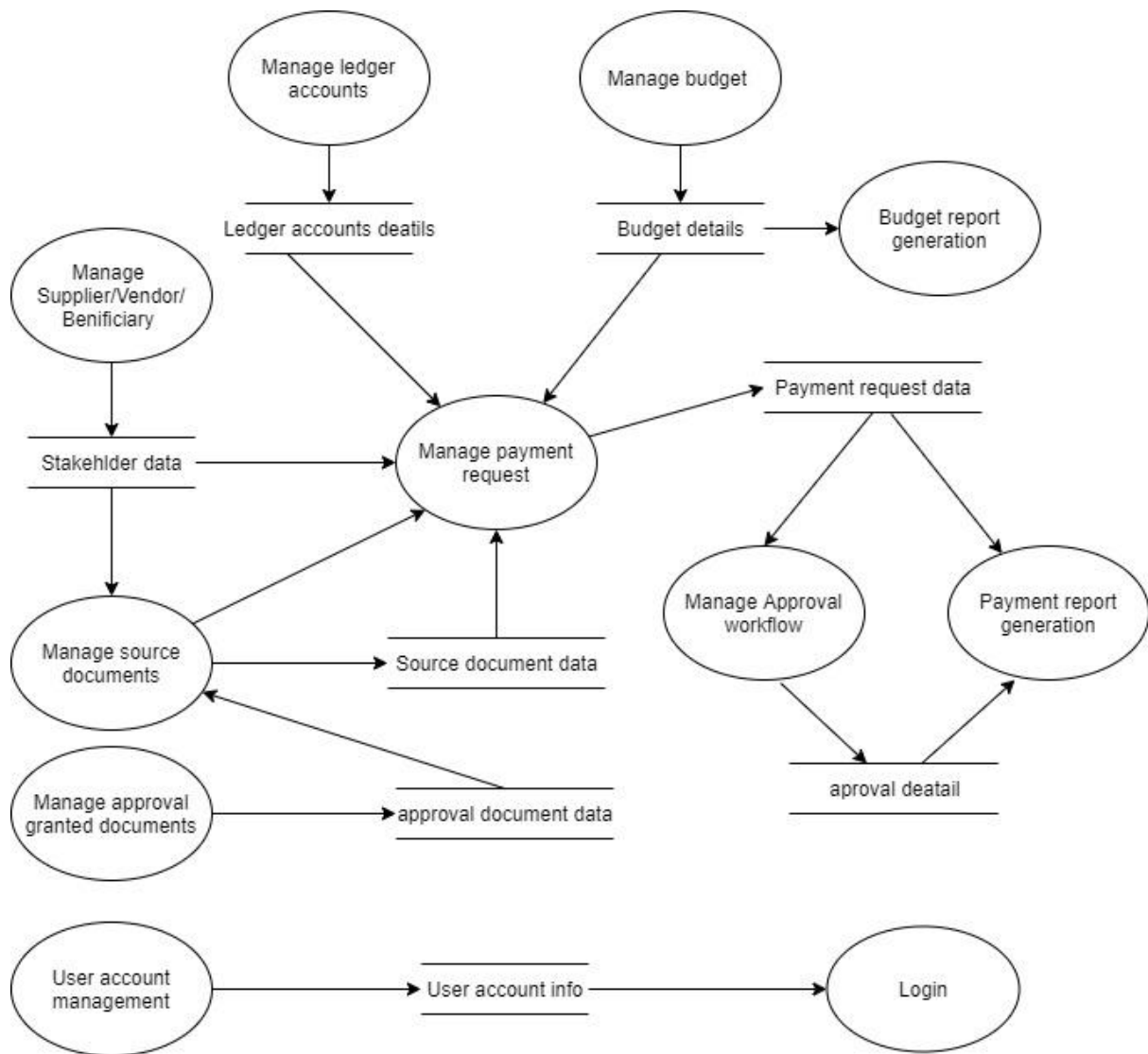


Figure 4. 1: Data Flow Diagram

4.5 Database Design

4.5.1 EER Diagram

A good database design process is based on specific design rules. A good database design is important in ensuring consistency of data, elimination of data redundancy, the efficiency of query execution, and high-performance application. Taking the time to design a database saves time and frustration during development, and a well-designed database ensures ease of access and retrieval of information (Writer, 2020). A good database design can use simpler queries and fast execution. The Overall performance of a database is dependent on the design. A good database is designed in a way that allows for easy maintenance.

The following Entity-Relationship diagrams in Figure 4. 2: EER Diagram-Part 1, Figure 4. 3: EER Diagram-Part 2, Figure 4. 4: EER Diagram-Part 3, and Figure 4. 5: EER Diagram-Part 4 describes the structure of the relational database model.

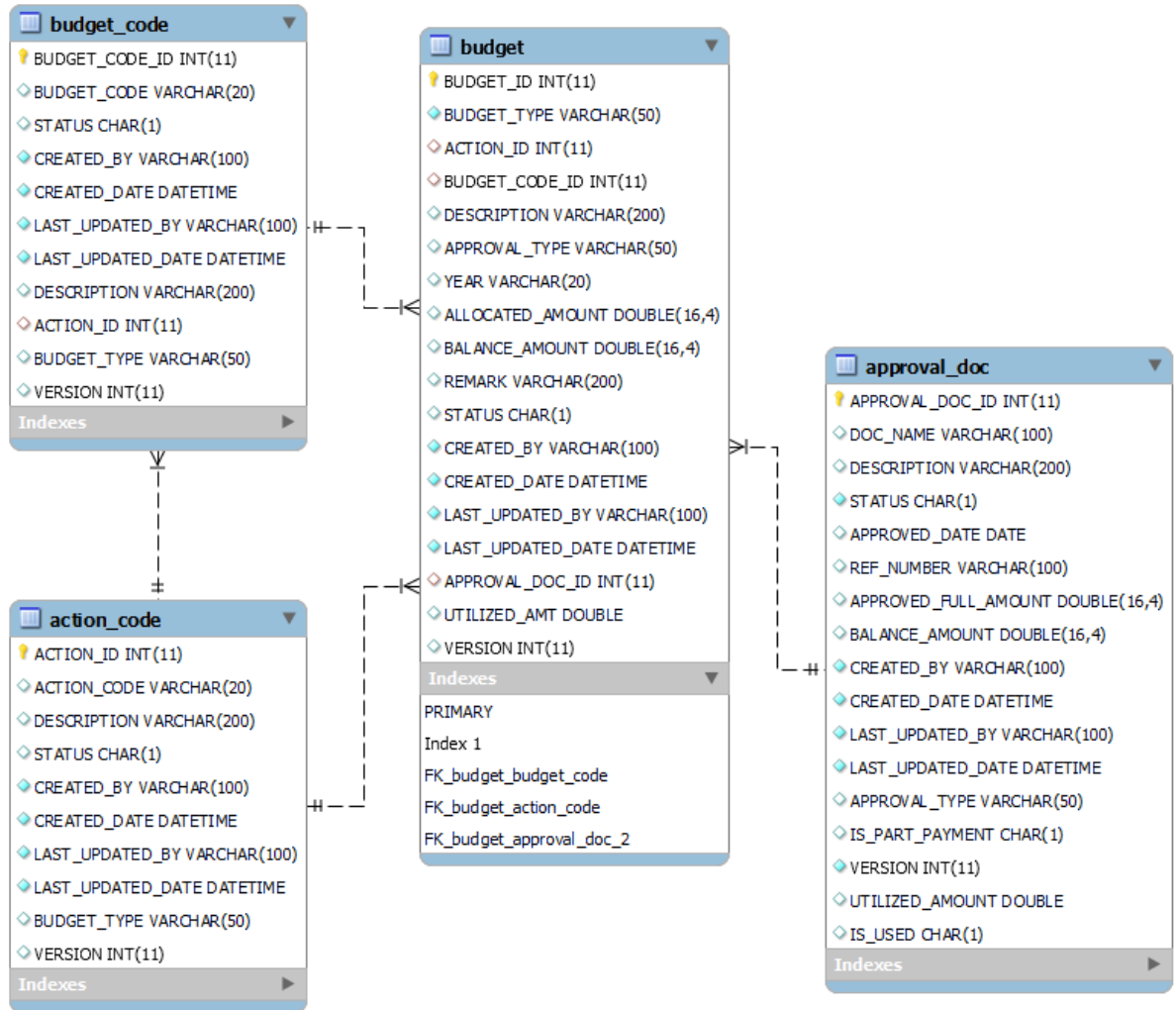


Figure 4. 2: EER Diagram-Part 1

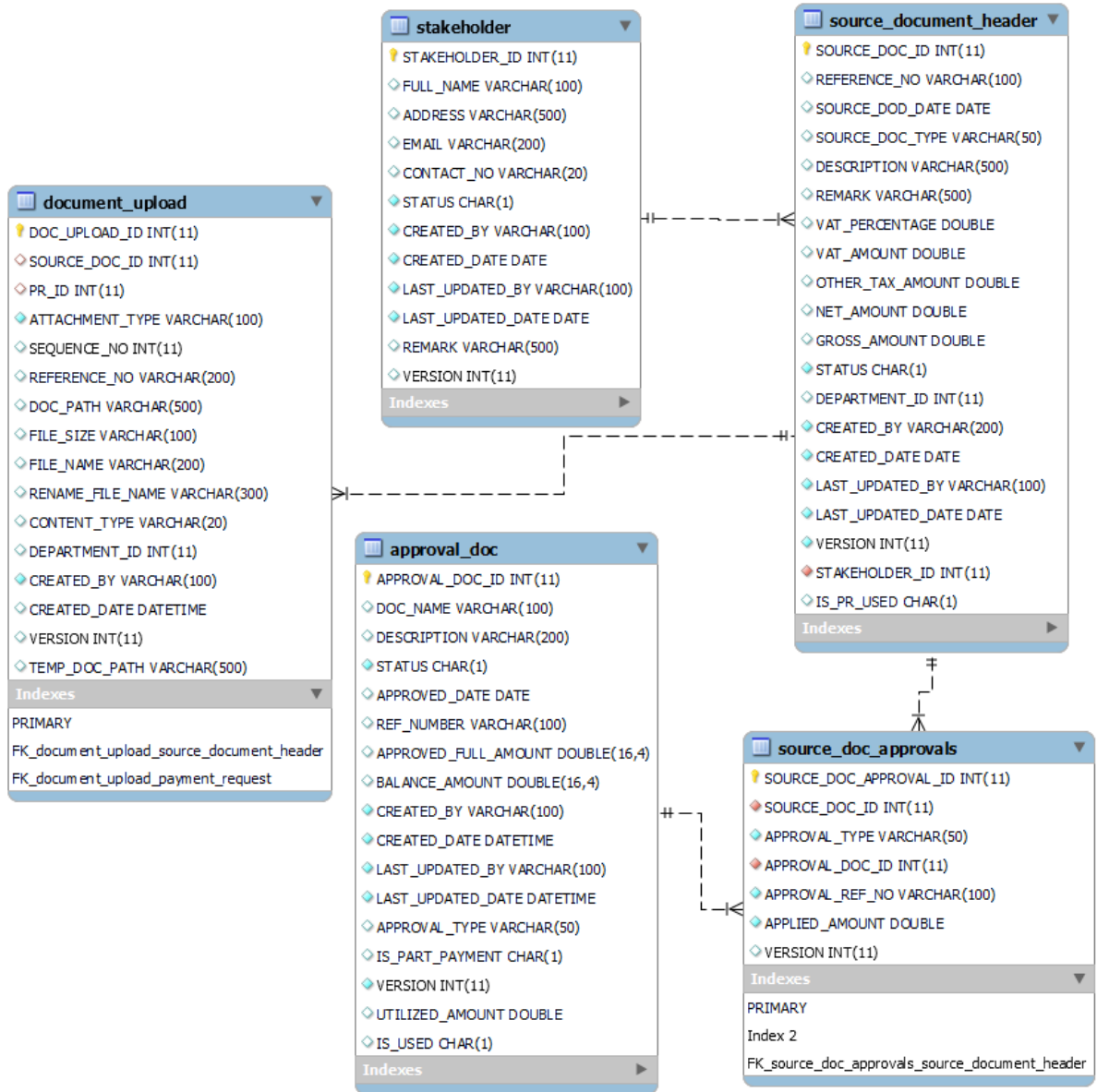


Figure 4. 3: EER Diagram-Part 2



Figure 4. 4: EER Diagram-Part 3

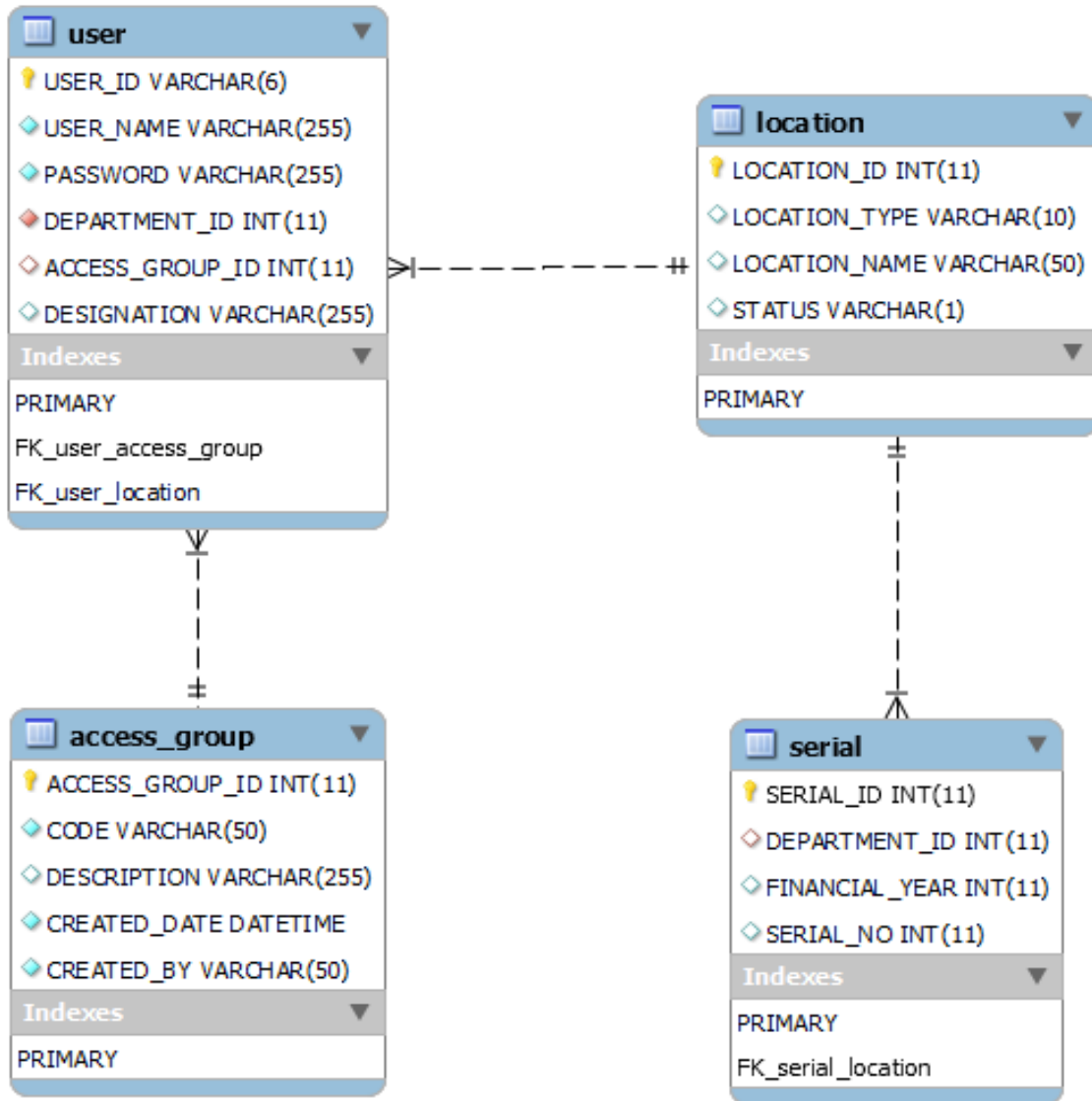


Figure 4. 5: EER Diagram-Part 4

Enhanced Entity-Relationship (EER) diagrams provide a visual representation of the relationships among the tables in the system. The current version of the EER diagram represents 18 entities with its attributes. For better representation, the EER diagram is divided into 4 parts according to functional behavior. All the relationships, primary keys, and foreign keys are displayed in the diagram. These entities are relevant to all the main functions of the system.

4.5.2 Relational Schema

budget_code (budget_code_id, budget_code, status, created_by, created_date, last_updated_by, last_updated_date, description, action_id, budget_type, version)

action_code (action_id, action_code, description, status, created_by, created_date, last_updated_by, last_updated_date, budget_type, version)

budget (budget_id, action_id, budget_code_id, description, approval_type, year, allocated_amount, balance_amount, remark, status, created_by, created_date, last_updated_by, last_updated_date, approval_doc_id, utilized_amount, version)

approval_doc (approval_doc_id, doc_name, description, status, approved_date, ref_number, approved_full_amount, balance_amount, utilized_amount, created_by, created_date, last_updated_by, last_updated_date, approval_type, is_part_payment, is_used_version)

stakeholder (stakeholder_id, full_name, address, email, contact_no, status, created_by, created_date, last_updated_by, last_updated_date, remark, version)

source_document_header (source_doc_id, reference_no, source_doc_date, source_doc_type, description, remark, vat_percentage, vat_amount, other_tax_amount, net_amount, gross_amount, status, department_id, created_by, created_date, last_updated_by, last_updated_date, stakeholder_id, is_pr_used, version)

document_upload (doc_upload_id, source_doc_id, pr_id, attachment_type, sequence_no, reference_number, doc_path, file_size, file_name, rename_file_name, content_type, created_by, created_date, department_id, temp_doc_path, version)

source_doc_approval (source_doc_approval_id, source_doc_id, approval_type, approval_doc_id, approval_ref_no, applied_amount, version)

payment_request (pr_id, department_id, subject, cost_center, pr_date, budget_type, action_code_id, budget_code_id, budget_id, payable_amount, vat_amount, nbt_amount, payable_other_amount, payable_total, withlodging_tax, retained_amount, deduction_other, deduction_total, net_amount, currency_type, created_by, created_date, last_updated_by, last_updated_date, pr_status, version, pr_number, penalty_amount, budget_year, stakeholder_id, pay_mode)

ledger_account (ledger_account_id, account_no, account_name, account_category, standard_type, remark, created_by, created_date, last_updated_by, last_updated_date, status)

pr_entries (pr_entries_id, pr_id, stakeholder_id, dr_cr, ledger_account_id, stk_account_no, amount, created_by, created_date, department_id)

pr_status (pr_status_id, pr_id, status, reject_reason, department_id, created_by, created_date, user_level)

pr_stk_source_doc (pr_stk_src_doc_id, pr_id, stakeholder_id, source_doc_id, created_by, created_date, department_id)

approval_workflow (workflow_id, pr_id, sequence_no, user_type, department_id, is_complete, is_cancelled, is_discard, created_date, created_user, updated_date, updated_user)

access_group (access_group_id, code, description, created_date, created_by)

user (user_id, user_name, password, department_id, access_group_id, designation)

location (location_id, location_type, location Name, status)

serial (serial_id, department_id, financial_year, serial_no)

4.6 Implementation of Modules

In development, the system decomposed into smaller modules. According to business function identified a number of possible subsystem areas for development, such as modules to deal with budget functions, reference data, business operations, approvals and acknowledgement, report generations. Although these areas perform as independent, there are connections between them. System user login handle through backend tables. Since the bank employee details are centrally maintained and login through “active directory (AD)”. Therefore backend table used for represent user login detail.

Each function available for authorized person only. This will manages through predefined access levels. All transactions managed concurrency by maintain version for record. Data integrity will be checked for delete function.

Below listed user levels are maintained in the system.

- System Administrator
- System Operator
- Officer A
- Officer B
- Super User
- Budget User

Main modules and features were developed in the system namely,

- Reference Data Module
- Budget Module
- Transaction Module
- Approval flow
- Report and Charts
- User Dashboard
- Notification for pending approvals

4.6.1 Implementation of Reference module

The “Reference module” controls all the master data which need to precede the system. The main functionalities of the Reference module are as follows.

- **Ledger Account** – All the ledger account should feed to system to conduct financial transaction.
- **Stakeholder** - System provide maintains facility for People’s Bank suppliers, contractors, beneficiary or any stakeholder.
- **Approval Documents** – Manage board paper approvals, top management approvals, letters, tenders, DGM approval or any approval documents for which granted payments. These approvals define whether part payment allows or not.

4.6.2 Implementation of Budget module

Define the budget codes, budget allocation for upcoming years, and maintaining the budget are done by this module. Following interfaces are available for budget managers.

- **Action Code Defining** – Budget categorize into main action areas and each category is identified by an action code. These action codes are related to pre-defined budget types.
- **Budget Code Defining**– Users can create any number of budget codes. These budget codes are under above mentioning action code. Budget codes use for the allocation of budget. These codes can change or inactivate through the system.
- **Manage Budget** –Users can allocate budget for upcoming budget year/s and also can monitor available budget balances for the already created budgets. If a user wants to stop the particular budget, the user can use Inactivate facility.

4.6.3 Implementation of Transaction module

In the transaction module, the following functions are facilitated.

- **Manage Source Document/Invoices** - This interface provides a facility to feed details of evidence or source documents such as invoices, receipts, purchase order, GRN, etc. Softcopy of these documents can upload to the system. These details should be able to retrieve throughout the payment approval flow to reflect the accuracy and integrity of the transaction
- **Request Payments** – This screen lets any department user request a payment for their invoices from the finance department. All the standard information should provide including suppliers, approvals evidence, source evidence, budget, double entries, payment mode, amounts with payable, and deduction separately.
- **Payment Workflow** – When request a payment it automatically create an approval workflow. This approval flow will be generated dynamically according to the requested amount. Each request should get department-level managers' approvals then approval will be pending to finance managers. For the higher payments, super user approval is needed. If anyone rejects during the approval flow, the approval process should be stopped and send back to changes.
- **Explore payments** – This provides the facility to find out pending approvals and payment status of the invoices and whether if it was paid or not. Several criteria will be provided as a filter option. According to user permission filter will be applied and approval function will be available

4.6.4 Dashboard and Notification

The user dashboard represents budget and payment status via charts. Notification alert display for pending approvals. This is only visible if the user level exists pending approvals.

4.6.5 Reports and Charts Module

The system produces daily transactional reports and summary reports for management. These summary reports are budget summary, payment summary, and chart for month budget utilization

4.6.6 Main User Interfaces

This section includes a few user interfaces for the main modules of the application that shows the system interface structure and system flow. All menu programs in the reference module and budget module include view, add, update, and delete functions. All add screens with a clear function and edit screens with reset function. For the delete function, data integrity has been checked. Further, these UI generate PDF output for existing records.

Login Page

The user login to the system by entering the employee user id and password to Figure 4. 6: Login Page.

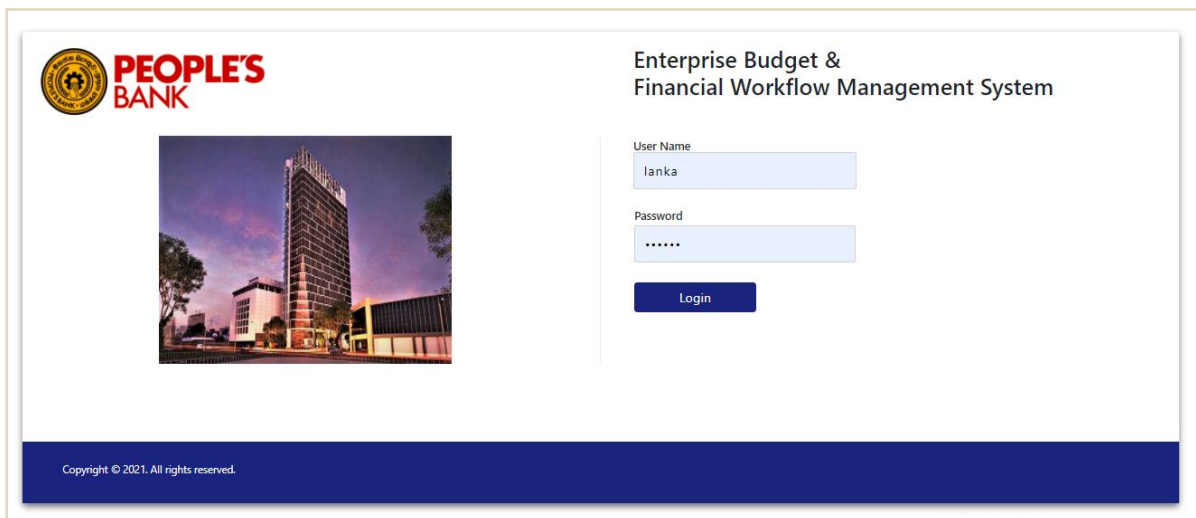


Figure 4. 6: Login Page

Dashboard

After successfully login into the system, the user can view the system dashboard as Figure 4. 7: Dashboard. This includes a pie chart to represent budget utilization of the current year and a column chart to denote the max number of payments requested by departments. Further, it shows the status of payments and notifications indicating pending payments as Figure 4. 8: Notification.

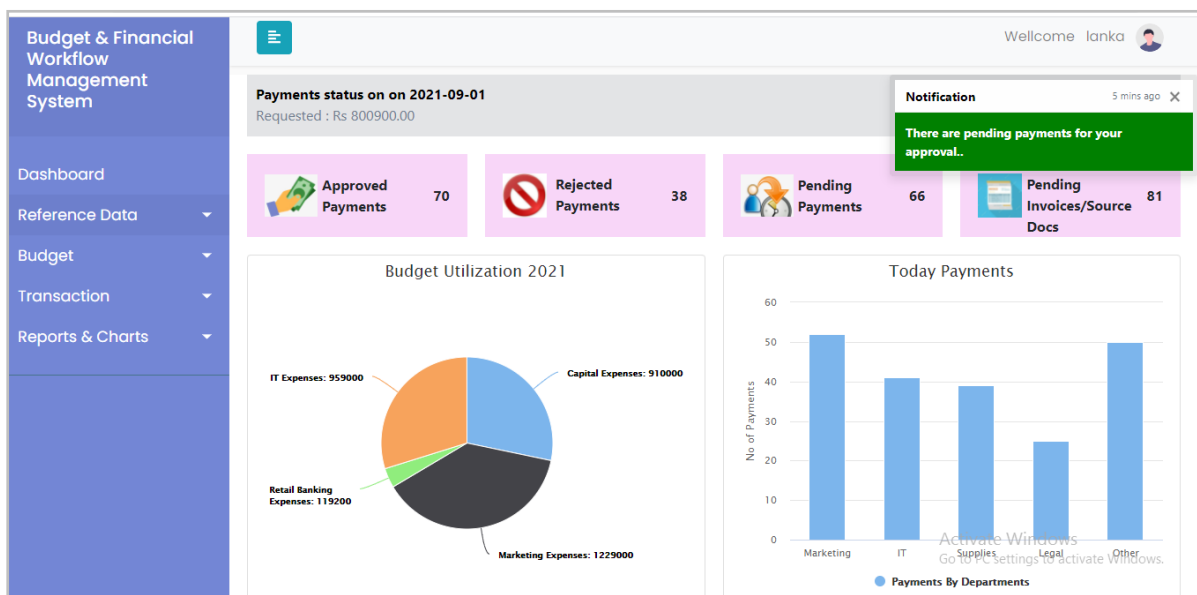


Figure 4. 7: Dashboard

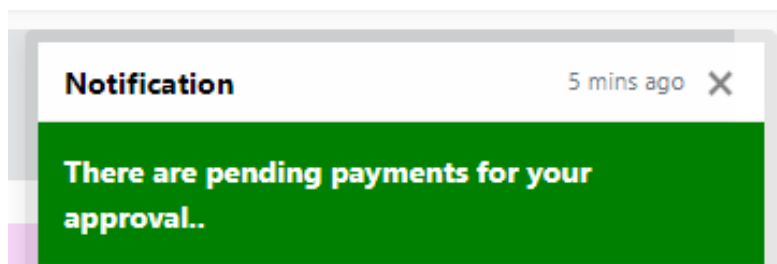


Figure 4. 8: Notification

Program Menu

After successfully login into the system, the user can access the system menu as Figure 4. 9: Dashboard menu - part 1. The expanded view of “Transaction” and “Reports” is displayed in Figure 4. 10: Dashboard menu – part 2. These all dashboard menu items are not visible to all users. Accessibility to menu items depends on user access level.

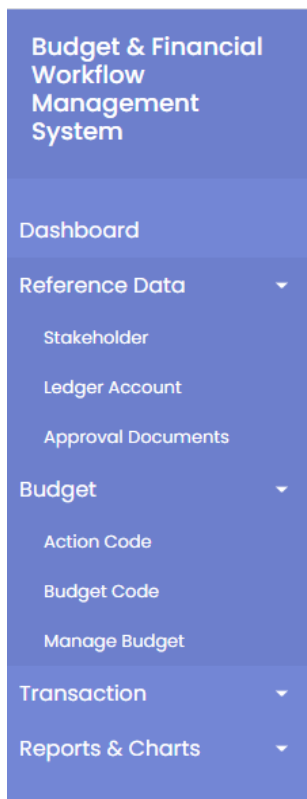


Figure 4. 9: Dashboard menu - part 1

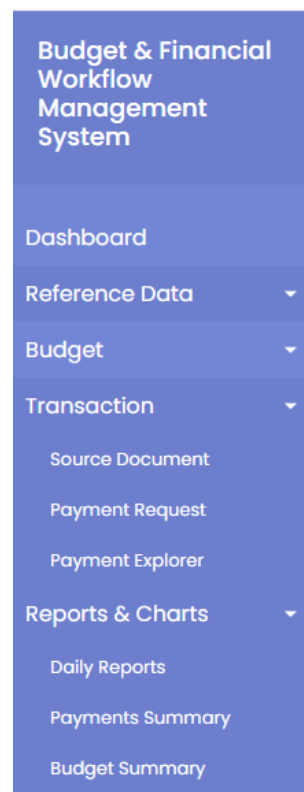




Figure 4. 10: Dashboard menu – part 2

Supplier/Stakeholder UI

Supplier, contractor, or any stakeholder can manage through this screen. The main screen of stakeholder UI is displayed in Figure 4. 11: Supplier/Beneficiary Detail. Using this, the user can go, add new stakeholder screen in Figure 4. 12: Supplier Add UI with Validation. Mandatory field validation messages are also displayed in Figure 4. 12: Supplier Add UI with Validation.

 **Supplier/Beneficiary**  Add New



16 items found, displaying 1 to 10. [\[First/Prev\]](#) 1, 2 [\[Next/Last\]](#)


| | Full Name | Address | Contact No | E-mail | Status | |
|----|--------------------------------------|-----------------------------------|----------------------|----------------------|----------|---|
| 1 | Saman Publication | No 34, Colombo 01 | <input type="text"/> | <input type="text"/> | Active | Edit Delete |
| 2 | Perera and Perera | No 12 Kandy Rd, Kadawatha | <input type="text"/> | <input type="text"/> | Active | Edit Delete |
| 3 | ABC Hardware | No 255/b, Colombo 02 | <input type="text"/> | | Active | Edit Delete |
| 4 | Tiddy Vedio | No 12, Kandy Rd, Kadawatha | <input type="text"/> | <input type="text"/> | Active | Edit Delete |
| 5 | Success Trading | Battaramulla, Colombo, Sri Lanka | <input type="text"/> | <input type="text"/> | Active | Edit Delete |
| 6 | Nano Foam Technology Private Limited | Colombo 02, Colombo 02, Sri Lanka | <input type="text"/> | | Active | Edit Delete |
| 7 | Magna Enterprises PVT (LTD) | Kegalla, Sri Lanka | <input type="text"/> | | Active | Edit Delete |
| 8 | Mewna Engineering Pvt Ltd | No 45/A, Colombo 04 | <input type="text"/> | <input type="text"/> | Active | Edit Delete |
| 9 | SMS Marketing Sri Lanka | No 11,2 C, Peliyagoda | <input type="text"/> | | Active | Edit Delete |
| 10 | Revo - Revolution | No 25, Kandy rd , Kiribathgoda | <input type="text"/> | <input type="text"/> | Inactive | Edit Delete |

Activate Windows
Go to PC settings to activate Windows.

Export options: [PDF](#)

Figure 4. 11: Supplier/Beneficiary Detail

 Wellcome lanka 

 **Add Supplier/Beneficiary**

Full Name / Company Name * Full Name / Company Name is required.

Address * Address is required.

Remark

Contact No * Contact No is required.

Email

Go Back

Submit
Clear

Figure 4. 12: Supplier Add UI with Validation

Ledger Account UI

A general ledger account list contains the names and numbers of all the bookkeeping accounts. This is called the chart of accounts. Bank chart of accounts can manage using these interfaces. The ledger account list and edit screens are displayed in Figure 4. 13: Ledger A/C List and Edit UI.

The screenshot displays two overlapping user interface screens. The top screen, titled 'Ledger Account', shows a table with 7 items. The bottom screen, titled 'Edit Ledger Account', shows a form for editing an account.

| | Account No | Account Name | Account Category | Standard Type | |
|---|------------|--------------------------------|------------------|---------------|-------------|
| 1 | 51101 | Technical Instruction Supplies | EXPENSE | DR | Edit Delete |
| 2 | 51001 | General Supplies and Materials | EXPENSE | DR | Edit Delete |
| 3 | 51203 | Copying, Printing, Publishing | EXPENSE | DR | Edit Delete |
| 4 | 51305 | General Expenses | EXPENSE | DR | Edit Delete |
| 5 | 51404 | | | | |
| 6 | 55017 | | | | |
| 7 | 51904 | | | | |


The 'Edit Ledger Account' form contains the following fields:

- Account No: 51305
- Account Name: General Expenses
- Account Category: Expense
- Standard Type: Debit
- Remark: Postage and Shipping
- Status: Active Inactive

Figure 4. 13: Ledger A/C List and Edit UI

Approval granted documents

One of the major screens in the project. Each payment is based on approvals. These approvals should enter into the system before making a payment. These documents detail very importantly to handle part payments. Approval document list and edit screen of the existing record is displayed in Figure 4. 14: Approval Document View/Edit Screen.

Wellcome lanka 

Approval Document + Add New

11 items found, displaying 1 to 10. [\[First/Prev\]](#) 1, 2 [\[Next/Last\]](#)

| | Approval Type | Reference No | Approved Date | Description | Full Amount | Balance Amount | Status | |
|----|---------------|--------------|---------------|--------------------------|-------------|----------------|--------|---|
| 1 | EOC | E01256 | 2021-09-03 | Covid 19 transport staff | 800000.0 | 770000.0 | Active | Edit Delete |
| 2 | Board Paper | B148-2021 | 2021-09-03 | IT infrastructure | 108000.0 | 88000.0 | Active | Edit Delete |
| 3 | HOD Approval | 2021-9893 | 2021-09-02 | Staff travelling | 250000.0 | 250000.0 | Active | Edit Delete |
| 4 | Board Paper | B-56963 | | | | | | |
| 5 | EOC | E108-20 | | | | | | |
| 6 | EOC | E256-20 | | | | | | |
| 7 | Board Paper | B1259 | | | | | | |
| 8 | SDGM Approval | A14-202 | | | | | | |
| 9 | Board Paper | B12356 | | | | | | |
| 10 | EOC | E 14562 | | | | | | |

Export options: [PDF](#)

Edit Approval Document

Approval Type *

Reference No *

Description

Full Amount *

Part Payment

Utilized Amount

Approval Date


Status * Active Inactive

[Go Back](#) [Submit](#) [Reset](#)

Figure 4. 14: Approval Document View/Edit Screen

Budget UI

Budget UI is available for budget users only. Budget modules have three main functions, “Action Codes”, “Budget Code” and “Budget Provisioning”. Action code and Budget Code are displayed in Appendix C. Budget provisioning each year with respect to budget codes are shown in Figure 4. 15: Budget Provision Details and Figure 4. 16: Budget Provisioning for a Year.

Wellcome lanka 

Budget Allocation + Add New

13 items found, displaying 1 to 10. [First/Prev] 1, 2 [Next/Last]

| | Budget Type | Action Code | Budget Code | Description | Year | Allocated Amount | Balance Amount | |
|----|--------------------|-------------------------|---|---------------------|------|------------------|----------------|---|
| 1 | Marketing Expenses | Product & Production | Isuru Udana | Isuru udana 2021 | 2021 | 300000.0 | 300000.0 | Edit Delete |
| 2 | Marketing Expenses | Product & Production | Vanitha Vasana | Vanitha vasana 2021 | 2021 | 100000.0 | 100000.0 | Edit Delete |
| 3 | Marketing Expenses | Product & Production | YES | Yes account 2021 | 2021 | 600000.0 | 511440.0 | Edit Delete |
| 4 | Capital Expenses | Information Technology | ATM / SBU Construction Cost / New Digital Branches | IT capital 2021 | 2021 | 1800000.0 | 1800000.0 | Edit Delete |
| 5 | IT Expenses | IT SYSTEMS | Infrastructure (Enhance / Upgrade) Existing Systems | IT infrastructure | 2021 | 750000.0 | 750000.0 | Edit Delete |
| 6 | IT Expenses | BUSINESS BENEFITS | New Solutions | New projects | 2022 | 950000.0 | 950000.0 | Edit Delete |
| 7 | IT Expenses | IT SYSTEMS | Infrastructure | IT infrastructure | 2022 | 1600000.0 | 1600000.0 | Edit Delete |
| 8 | Marketing Expenses | Product & Production | YES | Yes promoting 2022 | 2022 | 400000.0 | 400000.0 | Edit Delete |
| 9 | Capital Expenses | Information Technology | ATM / SBU Construction Cost / New Digital Branches | New unit | 2022 | 325000.0 | 325000.0 | Edit Delete |
| 10 | Capital Expenses | Building & Construction | Month Building | Building Rent | 2022 | 1800000.0 | 1800000.0 | Edit Delete |

Export options: PDF Activate Windows
Go to PC settings to activate Windows.

Figure 4. 15: Budget Provision Details

Edit Budget

Budget Type *

Action Code *

Budget Code *

Year *

Approval Type *

Approval Reference No *

Allocated Amount *

Utilized Amount

Description *

Remark

Status * Active Inactive

[Go Back](#)

Figure 4. 16: Budget Provisioning for a Year

Manage Invoice/Source Document UI

Each department system operator level user should feed source detail to the system. This screen provides the following facilities.

- Adding document general information
- Amount details, vat and other taxes if available
- Upload scan copy of the document
- Select approval details for the payment. There can be several approvals for one invoice.

Required information for invoice/ sources are displayed in Figure 4. 17: Source Document UI – part 01 and Figure 4. 18: Source Document UI– part 02. Because of complexity, here the screen displays as two figures.

Edit Source Document

Document Type * Invoice

Reference No * INV 2021-2836

Document Date * 03-Sep-2021

Supplier/Beneficiary * Revo - Revolution

Description * Office IIT Equipment

Remark

Status * Active Inactive

Net Amount * 50,000.00 Gross Amount 54,000.00

VAT Amount 4,000.00 VAT Percentage 8.0 Other Tax Amount 0.00

Figure 4. 17: Source Document UI – part 01

| | | | |
|--------------|-----------|------------------|-----------|
| Net Amount * | 50,000.00 | Gross Amount | 54,000.00 |
| VAT Amount | 4,000.00 | VAT Percentage | 8.0 |
| | | Other Tax Amount | 0.00 |

| | | |
|------------------|---|---|
| Edit Upload File | <input type="button" value="Choose File"/> No file chosen | Download file : INV 2021-2836.png |
|------------------|---|---|

| | | | |
|--------------------------|----------------------|-----------------------|---|
| Approval Detail * | | | |
| Approval Type | Select Approval Type | Approval Reference No | Select Approval No |
| | | | Full Amount |
| | | | 0.0 |
| Balance Amount | 0.0 | Applied Amount | 0.0 |
| | | | <input type="button" value="Add Approval"/> |


| | Approval Type | Approval No | Applied Amount | Delete |
|---|---------------|-------------|----------------|--------------------------|
| 1 | BOARD_PAPER | B148-2021 | 20000.0 | <input type="checkbox"/> |
| 2 | EOC | E01256 | 30000.0 | <input type="checkbox"/> |

| | | |
|--|---------------------------------------|--------------------------------------|
| <input type="button" value="Go Back"/> | <input type="button" value="Submit"/> | <input type="button" value="Reset"/> |
|--|---------------------------------------|--------------------------------------|

Figure 4. 18: Source Document UI– part 02

Payment Request

Main transaction of the system. Each payment should request through this UI. Payment reference numbers will generate for each request. Approve and rejection functions are available according to user level only. This UI is divided into four subsections by considering the complexity. However this all steps should complete making a payment. Existing payment details of approval pending record has appeared in Figure 4. 19: Payment - General Info, Figure 4. 20: Payment - Beneficiary Selection, Figure 4. 21: Payment - Payable/Deduction, and Figure 4. 22: Payment - GL entries. This view enables “Officer B” users and above.

Wellcome dilshan 

Payment No: 2021-611-000035
Status :PENDING

[Approve](#) [Reject](#) [Cancel](#)

General Info

Department: Information Technology Cost Center: 611 Provision Year *: 2021

Budget Type *: Marketing Expenses Action Code *: Product & Productic Budget Code *: YES

Budget Description: Yes account 2021

Allocated Amount: 600000.0

Utilized Amount: 88560.0

Balance Amount: 511440.0

Subject *: Account Campaign

Figure 4. 19: Payment - General Info

Supplier/Beneficiary Selection


Supplier/Beneficiary *: Revo - Revolution Payment Mode *: Pay Order [Search](#)

| | Document Type | Reference No | Date | Net Amount | VatAmount | Gross Amount | Select |
|---|---------------|---------------|------------|------------|-----------|--------------|-------------------------------------|
| 1 | Invoice | INV 2021-2836 | 2021-09-03 | 50000 | 4000 | 54000 | <input checked="" type="checkbox"/> |
| 2 | Invoice | In 202996 | 2021-09-04 | 32000 | 2560 | 34560 | <input checked="" type="checkbox"/> |

[Go Back](#) [Next](#)

Activate Windows
Go to PC settings to activate Windows.

Figure 4. 20: Payment - Beneficiary Selection

Wellcome dilshan 

Payment No: 2021-611-000035
Status : PENDING

Payable/Deduction

| | |
|--------------------------------------|----------------------------------|
| Payable Amount * | Withholding Tax |
| <input type="text" value="82000.0"/> | <input type="text" value="0.0"/> |
| VAT Amount | Retained Amount |
| <input type="text" value="6560.0"/> | <input type="text" value="0.0"/> |
| Payable Other | Deduction Other |
| <input type="text" value="0.0"/> | <input type="text" value="0.0"/> |
| Payable Total | Deduction Total |
| <input type="text" value="88560.0"/> | <input type="text" value=""/> |

Figure 4. 21: Payment - Payable/Deduction

Ledger Entries

Dr/Cr * Ledger Account * Customer Account *

Amount *

| | Dr/Cr | Ledger Account No | Customer Account | Amount | |
|---|-------|-------------------|------------------|---------|------------------------|
| 1 | CR | 51101 | | 88560.0 | Delete |
| 2 | DR | 51001 | | 88560.0 | Delete |

Activate Windows
 Go to PC settings to activate Windows.

Figure 4. 22: Payment - GL entries

Payment Explorer

This UI is used to filter any payment. According to user permission, the filter will be applied and an approval function will be available. Available filter criteria are displayed in Figure 4. 23: Payment Explorer.

The screenshot shows the 'Payment Explorer' interface. At the top right, it says 'Wellcome lanka' with a user profile icon. The main area contains several input fields: 'Payment No' with a placeholder 'Enter No (1-999999)', 'Payment Status' set to 'Pending', 'Supplier/Beneficiary' set to 'Saman Publication', 'From Date' set to '01-Sep-202', and 'To Date' set to '09-Sep-202'. Below these is a 'Pending Approval' section with 'Approval User' set to 'Officer B' and 'Department' set to 'Select Department'. At the bottom right are 'Search' and 'Reset' buttons.

Figure 4. 23: Payment Explorer

Approval Flow

Approval flow is one of the main objectives of the system. When the officer logged into the system, payment approval will enable according to workflow. This is a backend process. Following Figure 4. 24: Approval Workflow Table represented the database table which generated a record for each request.

| # | Name | Datatype | Length/Set | Unsign... | Allow N... | Zerofill | Default |
|----|---------------|----------|------------|--------------------------|-------------------------------------|--------------------------|----------------|
| 1 | WORKFLOW_ID | INT | 11 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | AUTO_INCREMENT |
| 2 | PR_ID | INT | 11 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | No default |
| 3 | SEQUENCE_NO | INT | 11 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | No default |
| 4 | USER_TYPE | VARCHAR | 100 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | " |
| 5 | DEPARTMENT_ID | INT | 11 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | '0' |
| 6 | IS_COMPLETE | VARCHAR | 50 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | 'N' |
| 7 | IS_CANCELLED | VARCHAR | 50 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | 'N' |
| 8 | IS_DISCARD | VARCHAR | 50 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | 'N' |
| 9 | CREATED_DATE | DATETIME | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | No default |
| 10 | CREATED_USER | VARCHAR | 100 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | No default |
| 11 | UPDATED_DATE | DATETIME | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | NULL |
| 12 | UPDATED_USER | VARCHAR | 100 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | NULL |

Figure 4. 24: Approval Workflow Table

Reports and Charts

Budget utilization throughout the year is represented as a column chart in Figure 4. 25: Chart - Monthly Budget Utilization. This graphical representation helps management to analyze budget variation and which budget category makes higher consumption. Management can decide on future budget allocations.

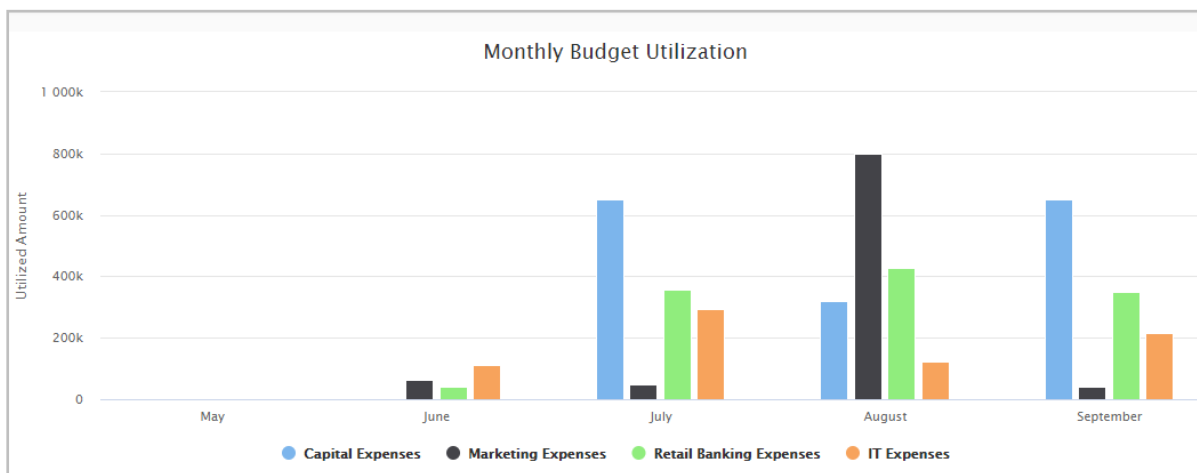


Figure 4. 25: Chart - Monthly Budget Utilization

Refer to “**Appendix D**” for transaction and management reports of the system.

4.7 Summary

This chapter illustrated the implementation phase of the system. Implementation environment, development tools and technologies, diagram for system data flows, database tables, and implemented system modules with main UI parts are discussed.

Chapter 05 - Testing and Evaluation

5.1 Introduction

Testing and evaluation is the process of comparing a system to requirements and specifications through testing. Testing is a mechanism for ensuring the quality of a system. Software testing consists of validation and verification. This chapter focuses on the entire testing and evaluation process for Enterprise Budget and Financial Workflow Management System.

5.2 Test Plan

Creating a test plan is one of the most foundational concepts in software testing. A test plan that describes the test strategy, objectives, resources, environment, schedule of the testing process. For an effective test plan, it is good to keep the test plan short and simple to understand and keep the test plan up-to-date. Once the test plan is well prepared, then write test scenarios and test cases based on the test plan document (Rajkumar, 2020). The objective of the testing activities is to check the functions and features of the developed system elaborated for web browsers.

5.3 Test Scope

This scope describes what is being tested, such as all the functions of the system, its interfaces behavior, and the integration of all functions. The following components and functions have been tested in high priority.

1. Adding approval documents and maintaining approval granted amount in transaction
2. Budget allocation and manage budget limit in a transaction
3. Manage Source Document function
4. Payment Request
5. Payment Approval Workflow
6. Payment Request Rejection and re-approval
7. Budget and Payment summary report

5.4 Testing Strategy

Describe the overall approach to testing. How testing will be performed, such as methodology, testing types, test techniques, etc. For each major function, integrations specify the approach which will ensure the behavior is adequately tested. Following approaches are used to test the system.

5.4.1 Unit Testing

Unit testing examines the entire system on a unit-by-unit basis to see if the software code works as expected. A validation and verification process that focuses on testing the smallest components or modules in the system, such as code fragments. Unit tests fall into the white-box test class. This test helps ensure that individual units or components of software source code are good enough for the user and that the code meets its design and works as intended. Unit testing is done by the programmer to test that the unit implemented by the programmer produces the expected output. Each program is considered a unit and termination of development for each program feature, and defects are tested, identified, and fixed. There are several unit testing techniques available. In the black box test, we tested the screen using input and output data. In transactional forms, each program has some key features including several main functions to test.

5.4.2 System and Integration Testing

Integration testing is the testing of a system by treating the entire system as a unit and seeing if the system can achieve its goals. After unit testing, can be integrated them to build a subsystem and then build a full system. Then should be tested the system again to find and remove any errors or flaws caused by the integration.

5.5 Test Cases

A test case is a specification of inputs, sets of conditions, test procedures, and expected results that define a single test performed to achieve a particular test objective. In the test case after applying a series of input values, the system gives particularly expected results. Test cases can be applied to all features in the system. Below are test cases for the key business scenarios used for system testing.

Test cases and results for approval document handling are in Table 5. 1: Test Cases for Manage Approval Document.

Table 5. 1: Test Cases for Manage Approval Document

| | | | |
|----------------------------------|--|--|---------------|
| Test Scenario: | | Manage approval documents | |
| Test Scenario Description | | View, add, edit and delete approval granted documents | |
| Pre-Requisites | | Log into a valid 'System Operator' profile | |
| No | Test Step | Expected Result | Status |
| 01 | 1. Click Approval Document from Reference Data menu 2. Click Add New button 3. Enter only the optional fields (Skip any mandatory field). 4. Submit the entered data. | Approval document should not be created. An error message should pop up saying that the mandatory fields need to be filled. | Pass |
| 02 | 1. Click Approval Document from Reference Data menu 2. Click Add New button 3. Enter all mandatory fields. 4. Submit the entered data. | Approval document should be successfully created. Success message should display | Pass |
| 03 | Click Approval Document from Reference Data menu | Should display the already entered approvals in a table with below fields. • Approval Type • Reference No • Approved Date • Description • Full Amount • Balance amount • Status • Edit and Delete button | Pass |
| 04 | 1. Click Approval Document from Reference Data menu 2. Select record from data table 3. Click Edit button | Edit source document page should load and relevant fields should be filled | Pass |

| | | | |
|----|--|---|------|
| 05 | <ol style="list-style-type: none"> 1. Click Approval Document from Reference Data menu 2. Select record from data table where part payment is yes 3. Click Edit button 4. Change full amount value as less than utilized amount 5. Click submit button | Error message should display as “Approved amount can't be less than utilized amount” | Pass |
| 06 | <ol style="list-style-type: none"> 1. Click Approval Document from Reference Data menu 2. Select record from data table where balance amount equal to zero and part payment is No 3. Click Edit button 4. Change approval amount 5. Click submit button | Record should not update. Error message should display. | Pass |
| 07 | <ol style="list-style-type: none"> 1. Click Approval Document from Reference Data menu 2. Select record from data table where part payment is yes 3. Click Edit button 4. Increase full amount 5. Change description 6. Click submit button | Approval document should update and display successfully update message | Pass |
| 08 | <ol style="list-style-type: none"> 1. Click Approval Document from Reference Data menu 2. Select active record from data table where utilized amount less than full amount 3. Click Edit button 4. Change status as Inactive 5. Click submit button | Approval document should updated as Inactive This document can't used for source document or budget allocation | Pass |
| 09 | <ol style="list-style-type: none"> 1. Click Approval Document from Reference Data menu 2. Select record from data table 3. Click delete button | Display confirmation message to delete | Pass |

| | | | |
|----|---|--|------|
| 10 | 1. Click Approval Document from Reference Data menu 2. Select record from data table where utilized amount greater than zero 3. Click delete button 4. Ok to confirm | Record can't delete. Display error message by mentioning it has related record. | Pass |
| 11 | 1. Click Approval Document from Reference Data menu 2. Select record from data table where utilized amount is zero 3. Click delete button 4. Ok to confirm | Record should delete from data table. Successfully deleted message should display | Pass |

Manage budget function include allocate budget for each financial year. Test cases and results for main budget business scenarios are in Table 5. 2: Test Cases for Budget function.

Table 5. 2: Test Cases for Budget function

| | | | |
|----------------------------------|--|---|---------------|
| Test Scenario | | Manage Budget | |
| Test Scenario Description | | View Budget , budget allocation, update and delete budget | |
| Pre-Requisites | | Log into a valid “Budget User” profile | |
| No | Test Step | Expected Result | Status |
| 01 | 1. Click Manage Budget from Budget menu 2. Click Add New button | System should load followings, Budget Type – dropdown list Possible values: <ul style="list-style-type: none"> • Capital Expenses • Marketing Expenses • Retail Banking Expenses • IT Expenses • Other | Pass |
| 02 | | Action Code – blank, dropdown list | Pass |
| 03 | | Budget Code- blank ,dropdown list | Pass |

| | | | |
|----|--|---|------|
| 04 | | Year- dropdown list Possible values :- minimum year = current year-2 maximum year = current year + 3 | Pass |
| 05 | | Approval Type – dropdown list Possible values: <ul style="list-style-type: none"> • EOC • Board Paper • All higher approvals | Pass |
| 06 | | Approval Reference Number – blank, dropdown list | Pass |
| 07 | | Allocated Amount – text field | Pass |
| 08 | | Utilized Amount – read only field | Pass |
| 09 | | Description - text field | Pass |
| 10 | | Remark - text field | Pass |
| 11 | | Submit, Clear, Go Back buttons | Pass |
| 12 | 1. Click Manage Budget from Budget menu 2. Click Add New button 3. Select any Budget Type | Action codes should be loaded to the action code dropdown according to the selected budget type. | Pass |
| 13 | 1. Click Manage Budget from Budget menu 2. Click Add New button 3. Select any Budget Type 4. Select any Action Code | Budget codes should be loaded to the action code dropdown according to the selected budget type. | Pass |
| 14 | 1. Click Manage Budget from Budget menu 2. Click Add New button 3. Select any Approval Type | Approval numbers should be loaded to the approval ref. no dropdown according to the selected approval type. | Pass |
| 15 | 1. Click Manage Budget from Budget menu 2. Click Add New button 3. Enter only the optional fields (Skip any mandatory field). 4. Submit the entered data. | Budget should not be created. An error message should pop up saying that the mandatory fields need to be filled. | Pass |
| 16 | 1. Click Manage Budget from Budget menu 2. Click Add New button | Budget should be successfully created. Success message should display | Pass |

| | | | |
|----|---|---|------|
| | 3. Fill all mandatory data 4. Submit the entered data. | | |
| 17 | Click Manage Budget from Budget menu | Should display the already entered budget data in a table with below columns. <ul style="list-style-type: none"> • Budget Type • Action Code • Budget Code • Description • Year • Allocated Amount • Balance Amount • Edit and Delete button | Pass |
| 18 | 1. Click Manage Budget from Budget menu 2. Select a record 3. Click Edit button | Edit budget page should load and relevant fields should be filled | Pass |
| 19 | 1. Click Manage Budget from Budget menu 2. Select a record 3. Click Edit button 4. Change allocated amount as less than utilized amount 5. Click submit | Error message should display as “Allocated amount can't be less than utilized amount” | Pass |
| 20 | 1. Click Manage Budget from Budget menu 2. Select a record where utilized amount greater than zero 3. Click Edit button 4. Change budget type 5. Click submit | Record should not update. Error message should display. | Pass |
| 21 | 1. Click Manage Budget from Budget menu 2. Select a record where utilized amount is zero 3. Click Edit button 4. Change filed values 5. Click submit | Record should be updated. Success message should display. | Pass |
| 22 | 1. Click Manage Budget from | Confirmation message should popup | Pass |

| | | | |
|----|--|---|------|
| | Budget menu 2. Select a record 3. Click Delete button | | |
| 23 | 1. Click Manage Budget from Budget menu 2. Select a record where utilized amount greater than zero 3. Click Delete button 4. Confirm ok to delete | Record should not be deleted and error message should display | Pass |
| 24 | 1. Click Manage Budget from Budget menu 2. Select a record where utilized amount is zero 3. Click Delete button 4. Confirm ok to delete | Record should be deleted and display successfully delete message. | Pass |

Test cases and results for Invoice/Source document handling are listed in Table 5. 3: Test Cases for Manage Source Document.

Table 5. 3: Test Cases for Manage Source Document

| | | | |
|----------------------------------|--|---|---------------|
| Test Scenario: | | Source Document Management | |
| Test Scenario Description | | View, Add, Edit and Delete source documents (Invoices, receipts, pay orders, and other) and integration with payments was tested here. | |
| Pre-Requisites | | Log into a valid 'System Operator' profile | |
| No | Test Step | Expected Result | Status |
| 01 | 1. Click Manage Source Document 2. Click Add New button 3. Enter only the optional fields (Skip any mandatory field). 4. Submit the entered data. | Particular source document should not be created. An error message should pop up saying that the mandatory fields need to be filled. | Pass |

| | | | |
|----|--|--|------|
| 02 | <ol style="list-style-type: none"> 1. Click Manage Source Document 2. Click Add New button 3. Enter all the mandatory details. 4. Save the entered data. | Particular source document should be successfully created. | Pass |
| 03 | <ol style="list-style-type: none"> 1. Click Manage Source Document | <p>Should display the already entered Source documents in a table with below fields.</p> <ul style="list-style-type: none"> • Reference No • Document date • Stakeholder Name • Description • Net amount • Gross amount • Is paid • Status • Edit and Delete button | Pass |
| 04 | <ol style="list-style-type: none"> 1. Click Manage Source Document 2. Select record from data table 3. Click Edit button | Edit source document page should load and relevant fields should be filled | Pass |
| 05 | <ol style="list-style-type: none"> 1. Click Manage Source Document 2. Select record from data table where “Is payment used” value is YES. 3. Click Edit button | Should not allow to update the source document content. | Pass |
| 06 | <ol style="list-style-type: none"> 1. Click Manage Source Document 2. Select record from data table where “Is payment used” value is No. 3. Click Edit button 4. Changed the field values and submit with mandatory field values | <p>Should allow to edit particular source document.</p> <p>Particular changes should be successfully made.</p> | Pass |

| | | | |
|----|--|--|------|
| 07 | <ol style="list-style-type: none"> 1. Click Manage Source Document 2. Select record from data table 3. Click Edit button 4. Click scan copy download button | Uploaded softcopy should be download to users machine | Pass |
| 08 | <ol style="list-style-type: none"> 1. Click Manage Source Document 2. Select active and is payment used ="no" record from data table 3. Click Edit button 4. Change status as "Inactive" 5. Click submit button | <p>Source document status changed as Inactive.</p> <p>This document can't use for make payment request.(The document should not available in payment request screen)</p> | Pass |

Request payment is one of the major functions of the project. Test cases were executed for major business scenarios and they are in below Table 5. 4: Test Cases for Payment Request.

Table 5. 4: Test Cases for Payment Request

| | | | |
|----------------------------------|-------------------------------------|---|---------------|
| Test Scenario: | | Payment Request | |
| Test Scenario Description | | Request payment from finance deaprtment and relevant condition and controls are tested here. | |
| Pre-Requites | | Log into a valid 'System Operator' profile | |
| No | Test Step | Expected Result | Status |
| 01 | Click Manage Payment Request screen | System should load followings, User Department – read only text field | Pass |
| 02 | | Cost Center No - read only text field | Pass |
| 03 | | Provision Year – Dropdown list. Should load current year and previous 3 years. | Pass |
| 04 | | Subject – Text field | Pass |
| 05 | | Type of Budget – load data to dropdown list | Pass |
| 06 | | Action Code – load as disable field. When user changes "type of budget", relevant action codes should be loaded to the action code dropdown according | Pass |

| | | | |
|----|----------------------|---|------|
| | | to the selected budget type. | |
| 07 | | Budget Code – load as disable field. When user change “action code” relevant budget codes should be loaded to the budget code dropdown according to the selected action code. | Pass |
| 08 | | Budget Description – read only field. Data will fill according to the selected budget code | Pass |
| 09 | | Allocated Amount - Read only field. Data will fill according to the selected budget code | Pass |
| 10 | | Balance Amount - Read only field. Data will fill according to the selected budget code | Pass |
| 11 | | Supplier/Beneficiary – dropdown list. All active stakeholders should load | Pass |
| 12 | | Payment Mode– dropdown list. Possible values : Pay Order, Customer Account | Pass |
| 13 | | List of Sources – Blank data table | Pass |
| 14 | | Amount Payable Header - Label | Pass |
| 15 | | Amount – Text field | Pass |
| 16 | | VAT – Text field | Pass |
| 17 | | Any Other – Text field | Pass |
| 18 | | Payable Total – read only field | Pass |
| 19 | | Deduction - Lable | Pass |
| 20 | | Withholding Tax – Text field | Pass |
| 21 | | Retained Amount – Text field | Pass |
| 22 | | Any Other – Text Field | Pass |
| 23 | | Deduction Total - read only field | Pass |
| 24 | | Entries - Label. | Pass |
| 25 | Entries input fields | DR/CR - dropdown list. Possible values : DR, CR | Pass |
| 26 | | Ledger Account - dropdown list. All active ledger accounts should be load. | Pass |

| | | | |
|----|---|--|------|
| 27 | | Customer Account – text field | Pass |
| 28 | | Add Button | Pass |
| 29 | | Entries table - Blank | Pass |
| 30 | | Document Upload Label | Pass |
| 31 | Document Upload input field | Document Type - dropdown | Pass |
| 32 | | Reference No – text box | Pass |
| 33 | | Browse button | Pass |
| 34 | | Add button | Pass |
| 35 | | Upload document data table - blank | Pass |
| 36 | Footer Row fields | Submit , Clear buttons | Pass |
| 37 | 1. Click Manage Payment Request screen 2. Select Supplier/any stakeholder | All active , not used source document should load to list of sources data table | Pass |
| 38 | 1. Click Manage Payment Request screen 2. Select Supplier/any stakeholder 3. select a record from list of sources data table 4. Click Next button | Payable and deduction columns should fill according to selected row. <ul style="list-style-type: none"> • Amount • VAT • Any other • Payable total • Withholding tax • Retained amount • Other deduction • Deduction Total | Pass |
| 39 | 1. Click Manage Payment Request screen 2. Select Supplier/any stakeholder 3. select a record from list of sources data table 4. unselect a selected record from list of sources data table | Payable and deduction columns should update according to unselected row.(amount should be deduct accordingly) | Pass |

| | | | |
|----|--|--|------|
| 40 | <ol style="list-style-type: none"> 1. Click Manage Payment Request screen 2. Select Supplier/any stakeholder 3. Select Dr or Cr 4. Select Ledger Account 5. Select Customer Account 6. Enter amount 7. Click add button | Record should not add to data table Error message should display as both ledger account and customer account can't add together | Pass |
| 41 | <ol style="list-style-type: none"> 1. Click Manage Payment Request screen 2. Select Supplier/any stakeholder 3. Select Dr or Cr 4. Select Ledger Account or Customer Account 5. Enter amount 6. Click Add button | Record should added to data table | Pass |
| 42 | <ol style="list-style-type: none"> 1. Click Manage Payment Request screen 2. Add data to entries data table 3. Click delete button from entries data table | Record should removed from entries table | Pass |
| 43 | <ol style="list-style-type: none"> 1. Click Manage Payment Request screen 2. Select and enter all required field 3. Enter double entries where CR and DR total not equal 3. Click submit button | Error message should be displayed by indicating entries are not tally. | Pass |
| 44 | <ol style="list-style-type: none"> 1. Click Manage Payment Request screen 2. Select and enter all required field 3. Click submit button | Record should be saved Payment Request No should be generated. The payment status should display as "Draft" | Pass |

5.6 Evaluation of the System

User evaluation was conducted with bank employees using the following evaluation form in Table 5. 5: User Evaluation Form. Overall feedback summary represents in Figure 5. 1: Feedback Summary.

Table 5. 5: User Evaluation Form

| User Evaluation Form Enterprise Budget and Financial Workflow Management System People's Bank | | | | | | |
|--|--|------------------------------|---------------------|--------------------|------------------|---------------------------|
| Date | | | | | | |
| Employee Name | | | | | | |
| Please select answers to the below stated questions. Mark "X" in the appropriate column | | | | | | |
| No | Question | Strongly Disagree (1) | Disagree (2) | Neutral (3) | Agree (4) | Strongly Agree (5) |
| 01 | This web base system is essential to the bank | | | | | |
| 02 | Is user screens are easy to understand? | | | | | |
| 03 | Informations and calculations are appearing correctly. | | | | | |
| 04 | Drawbacks of manual process are solved | | | | | |
| 05 | Is this system help and easy for my job | | | | | |
| 06 | Is this system is better than manual process? | | | | | |
| 07 | I am satisfied with the content of this system | | | | | |
| 08 | Is automated approval workflow is efficient? | | | | | |
| 09 | Is system generated reports are useful | | | | | |
| 10 | The system provides better information for the decision-making process | | | | | |
| 11 | The interfaces of this system pleasant | | | | | |
| 12 | Do you think the system improve transparency of payments and approvals | | | | | |

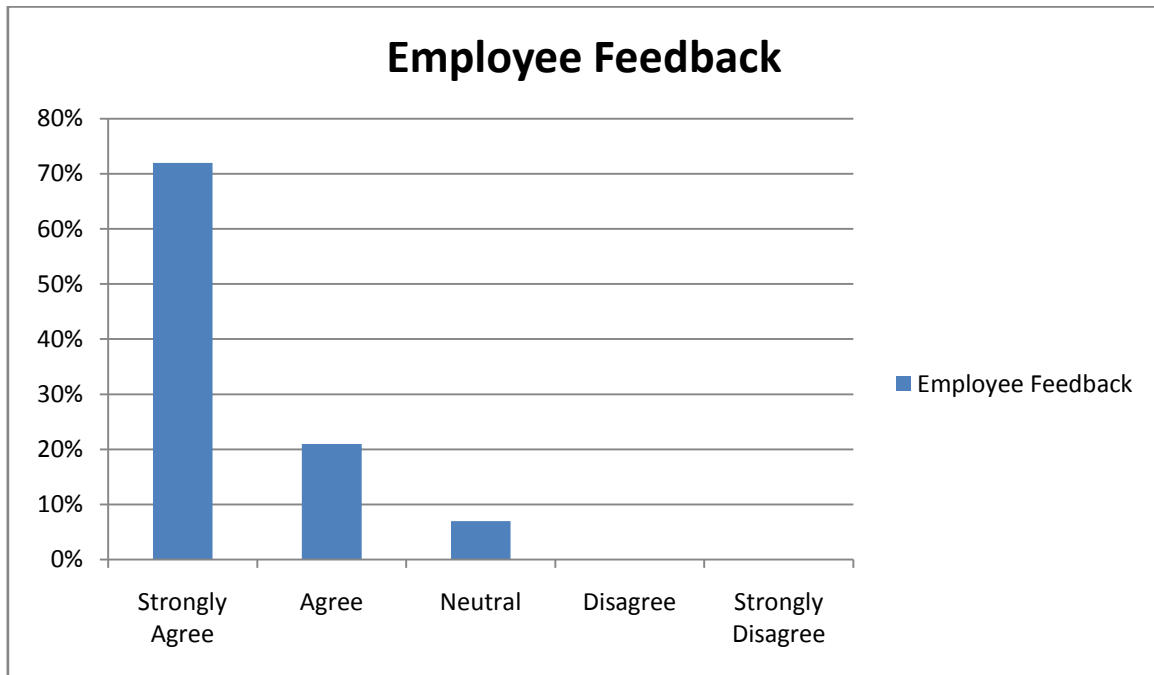


Figure 5. 1: Feedback Summary

Evaluation done by six employees of the bank after goes through main functions of the system. The result was collected through Table 5. 5: User Evaluation Form. The feedback summary illustrated in Figure 5. 1: Feedback Summary, is generated by collecting a total of answers (user count) for each category concerning each question.

Total user count for 12 question = 72

Summary: Strongly agree = $52/72 = 72\%$

Agree = $15/72 = 21\%$

Neutral = $5/72 = 7\%$

5.7 Summary

This chapter describes project testing and evaluation. This chapter explained how to run tests, define test cases, and test results. After completing the user test, when reviewing the execution results, the evaluation process either meets the functional requirements provided by the client or the user performs normal tasks effectively and efficiently through the new system. According to the test results obtained in the test cases, most of the expected functions of the system have been successfully achieved. Limitations and future development are discussed in the last chapter.

Chapter 06 - Conclusion

6.1 Introduction

This chapter describes the overall outcome of the project objectives, the problems faced during the project, some limitations of the solution, and other tasks that can be performed on the system after the final product is evaluated.

6.2 Conclusion

People's bank is one of the top state-owned commercial bank in Sri Lanka. There are huge amounts of outside payments handle daily, to manage day-to-day business operations smoothly. Finance Department is responsible for managing funds within the organization and manages expenditure of funds on various assets. It is essential to efficient financial management and financial control to support all banking business activities. The aim of this project is to automate the finance and workflow management system for People's Bank. The implemented system automates the process of budget allocation to payment approval workflow of all the payments and through that give proper solution to current manual system drawbacks. The automation process includes budget allocation, ledger account handling, approvals maintains, documents repository, and dynamic approval workflow. The system generates daily reports and summary reports of budget and payments. These reports will be used for analyzing budget variation, control payments, and also funding decisions. A chart available for analyzing monthly utilization of budget.

6.3 Limitations and Future Enhancement

Payment approval flow ends with getting approval from "Office A" or "Super User" of the financial department. This depends on the payment amount. For higher amounts, super approval is needed. The next step will be pay order printing and double-entry passing to the bank core system. These steps do not integrate with this project. Bank already has an interface to connect with the core system and at this level, we can't connect it. But relevant financial information can be provided via this system.

In customer account payment the user should enter customer/supplier account number into the system. Because of the unavailability of core integration in this project, the account

number validation can't process. As a next step, the system will be integrated with core and get account information in real-time. Further, at the end of the payment, the system can generate an SMS alert to a customer by indicating money transferring. Customer contact information already manages by the developed system.

For the pending approval system notification already implemented in this project. Because of the criticality of payment, this notification can enhance mobile messages as a further improvement.

6.4 Summary

The Conclusions chapter completes the developed web application and describes further development areas of the project. References and appendices can be found in the next section.

References

AltexSoft. 2019. *Web Application Architecture: How the Web Works*. [online] Available at: <<https://www.altexsoft.com/blog/engineering/web-application-architecture-how-the-web-works>> [Accessed 28 March 2021].

Cio-wiki.org. 2021. *Client Server Architecture - CIO Wiki*. [online] Available at: <https://cio-wiki.org/wiki/Client_Server_Architecture> [Accessed 28 March 2021].

En.wikipedia.org. 2021. *SAP ERP - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/SAP_ERP> [Accessed 6 February 2021].

ERProof. 2021. *How Does SAP Work? - Easy Explanation with An Example*. [online] Available at: <<https://erproof.com/how-does-sap-work>> [Accessed 3 February 2021].

Hammack, H., 2021. *The Disadvantages of Quickbooks*. [online] Available at: <<https://www.techwalla.com/articles/the-disadvantages-of-quickbooks>> [Accessed 22 November 2020].

Ibm.com. 2016. *IBM Docs*. [online] Available at: <<https://www.ibm.com/docs/en/odm/8.8.1?topic=application-java-se-java-ee-applications>> [Accessed 23 May 2021].

Infoentrepreneurs.org. 2021. *Budgeting and business planning*. [online] Available at: <<https://www.infoentrepreneurs.org/en/guides/budgeting-and-business-planning>> [Accessed 3 September 2020].

Intuit Developer Community Blog. 2021. *What do developers want to see more of in 2021? - Intuit Developer Community Blog*. [online] Available at: <<https://blogs.intuit.com/blog/2021/01/28/what-do-developers-want-to-see-more-of-in-2021>> [Accessed 3 February 2021].

M. Salter, R., 2019. *QuickBooks Online*. [ebook] Intuit Limited. Available at: <<https://quickbooks.intuit.com/content/dam/intuit/quickbooks/msm/sites/en/global/Global%20QuickBooks%20Online%20User%20Guide%202019.pdf>> [Accessed 4 February 2021].

Mindfire Solutions. 2021. *Custom Software Development Company in India - Mindfire Solutions*. [online] Available at: <<https://www.mindfiresolutions.com/Offshore-QuickBooks-Development-India.htm>> [Accessed 6 February 2021].

People's Bank. 2020. *Payment Work Flow Diagram*, Colombo: People's Bank.

People's Bank. 2019. *Payment Request Form*, Colombo: People's Bank.

Rajkumar, 2020. *Software Test Plan Template with Detailed Explanation [Sample Test Plan Document]*. [online] Software Testing Material. Available at: <<https://www.softwaretestingmaterial.com/test-plan-template>> [Accessed 2 July 2021].

Spring.io. 2021. *Spring Framework*. [online] Available at: <<https://spring.io/projects/spring-framework>> [Accessed 23 May 2021].

Sswm.info. 2021. *Budget Allocation and Resource Planning | SSWM - Find tools for sustainable sanitation and water management!*. [online] Available at: <<https://sswm.info/humanitarian-crisis/prolonged-encampments/planning-process-tools/implementation-tools/budget-allocation-and-resource-planning>> [Accessed 3 September 2020].

User, S., 2020. *TechnoFunc - How Sap Works*. [online] Technofunc.com. Available at: <<https://www.technofunc.com/index.php/erp/145-how-sap-works>> [Accessed 22 November 2020].

Visual-paradigm.com. 2021. *What is Class Diagram?*. [online] Available at: <<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>> [Accessed 17 July 2021].

Writer, S. 2020. *Why Is Database Design so Important?*. [online] Reference.com. Available at: <<https://www.reference.com/world-view/database-design-important-42dda832528ce266>> [Accessed 24 May 2021].

Appendix A - Code Segments

A.1 Introduction

This section describes some important areas of system development. The implemented system used MVC design pattern.

A.2 Folder Structure

Eclipse IDE is used for the development environment. The project folder structure in eclipse IDE is displayed in Figure A. 1: Project Structure.

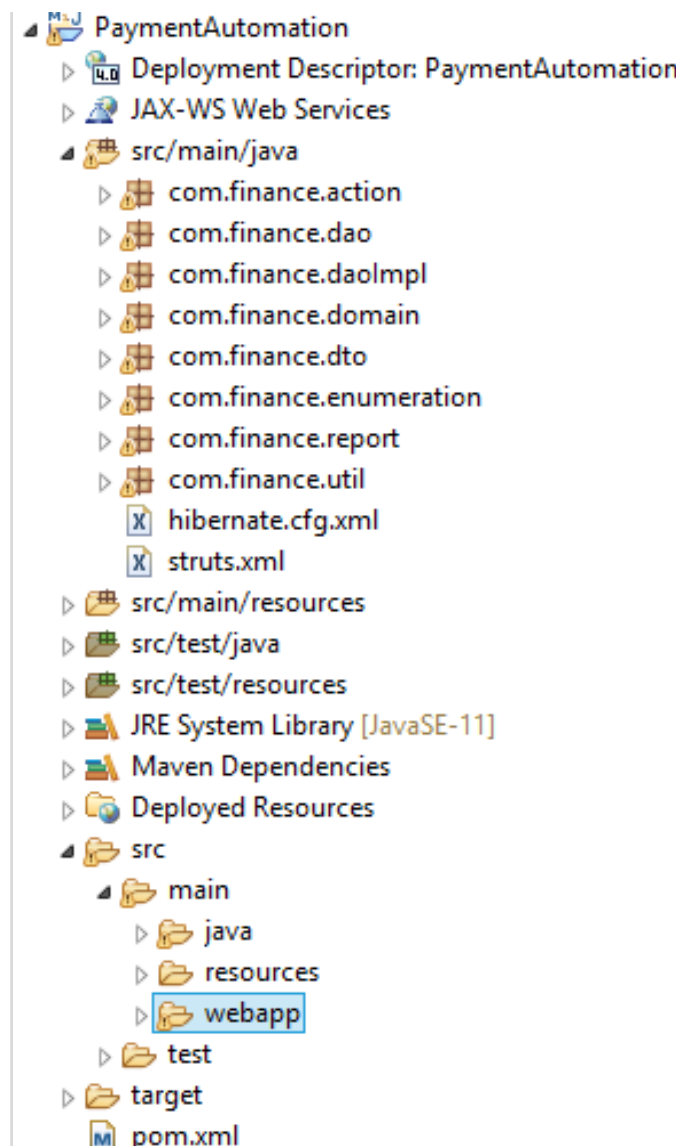


Figure A. 1: Project Structure

A.3 Java Approval Flow Implementation

When payment is requested through the system, the method in Figure A. 2: Generate Approval Flow is activated. It generates suitable approval flow by indicating sequential order of approval users. Each payment should follow an approval flow.

```
public List<ApprovalWorkflow> createApprovalWorkflow(PaymentRequest paymentRequest, UserSession userSession) {
    List<Map<AccessLevelEnum, Integer>> list = new ArrayList<Map<AccessLevelEnum, Integer>>();

    Map<AccessLevelEnum, Integer> map = new HashMap<AccessLevelEnum, Integer>();
    map.put(ApprovalFlowEnum.BASIC_DEPARTMENT.getLevel1(), paymentRequest.getDepartmentId());
    list.add(map);

    map = new HashMap<AccessLevelEnum, Integer>();
    map.put(ApprovalFlowEnum.BASIC_DEPARTMENT.getLevel2(), paymentRequest.getDepartmentId());
    list.add(map);

    if(paymentRequest.getNetAmount()>=1000000) {
        map = new HashMap<AccessLevelEnum, Integer>();
        map.put(ApprovalFlowEnum.SUPER_FINANCE.getLevel1(), ParameterEnum.FINANCE_DEPARTMENT.getDepartmentId());
        list.add(map);

        map = new HashMap<AccessLevelEnum, Integer>();
        map.put(ApprovalFlowEnum.SUPER_FINANCE.getLevel2(), ParameterEnum.FINANCE_DEPARTMENT.getDepartmentId());
        list.add(map);

        map = new HashMap<AccessLevelEnum, Integer>();
        map.put(ApprovalFlowEnum.SUPER_FINANCE.getLevel3(), ParameterEnum.FINANCE_DEPARTMENT.getDepartmentId());
        list.add(map);
    } else {
        map = new HashMap<AccessLevelEnum, Integer>();
        map.put(ApprovalFlowEnum.BASIC_FINANCE.getLevel1(), ParameterEnum.FINANCE_DEPARTMENT.getDepartmentId());
        list.add(map);

        map = new HashMap<AccessLevelEnum, Integer>();
        map.put(ApprovalFlowEnum.BASIC_FINANCE.getLevel2(), ParameterEnum.FINANCE_DEPARTMENT.getDepartmentId());
        list.add(map);
    }

    int sequence=0;
    List<ApprovalWorkflow> approvalList = new ArrayList<ApprovalWorkflow>();
    for (Map<AccessLevelEnum, Integer> m : list) {
        for (Map.Entry<AccessLevelEnum, Integer> entry : m.entrySet()) {
            ApprovalWorkflow approvalWorkflow = new ApprovalWorkflow();
            approvalWorkflow.setPrId(paymentRequest.getPaymentRequestId());
            approvalWorkflow.setSequenceNo(++sequence);
            approvalWorkflow.setUserType(entry.getKey().getCode());
            approvalWorkflow.setDepartmentId(entry.getValue()); //department id
            approvalWorkflow.setIsComplete("N");
            approvalWorkflow.setIsCancelled("N");
            approvalWorkflow.setIsDiscard("N");
            approvalWorkflow.setCreatedDate(new Date());
            approvalWorkflow.setCreatedUser(userSession.getUserName());
            approvalWorkflow.setUpdatedDate(null);
            approvalWorkflow.setUpdatedUser(null);
            approvalList.add(approvalWorkflow);
        }
    }
    return approvalList;
}
```

Figure A. 2: Generate Approval Flow

A.4 JQuery Ajax Implementation

Following code segment in Figure A. 3: JQuery Ajax Implementation used to filter action code dropdown list according to user selected budget type.

```
<script type="text/javascript">

    $(document).ready(function() {
        $('#budgetTypes').change(function(event) {
            var budgetTypes = $("select#budgetTypes").val();
            var actionCodeMap = "";
            var dummyMsg="";
            $.getJSON('ajaxAction?method=getActionCodes', {
                budgetType : budgetTypes
            }, function(jsonResponse) {
                var select = $('#actionCodes');
                select.find('option').remove();
                $.each(jsonResponse.actionCodeMap, function(key, value) {
                    $('<option>').val(key).text(value).appendTo(select);
                });
            });
        });
    });

</script>
```

Figure A. 3: JQuery Ajax Implementation

A.5 Method Reuse

Reusable methods have been implemented in a separate class to CRUD functions with exception handling and rollback features. A rollback function is required when saving and deleting a list of records if there is an error occurs. Code segments for these functions are displayed in Figure A. 4: Save multiple records and Figure A. 5: Delete multiple records.

```

public void saveAll(List<Object> objects) throws BaseException{
    Session session = openSession();
    Transaction tn = null;

    try{
        tn = (Transaction) session.beginTransaction();
        for(Object object : objects) {
            session.save(object);
        }
        tn.commit();
    }
    catch (Exception ex){
        rollbackTransaction(tn);
        if (ex instanceof ConstraintViolationException){
            throw new BaseException("ERROR_UNIQUE_CONSTRAINT");
        }
        if (ex instanceof JDBCException){
            throw new BaseException(((JDBCException)ex).getSQLException());
        }
        throw new BaseException(ex);
    }
    finally{
        closeSession(session);
    }
}

```

Figure A. 4: Save multiple records

```

public void deleteAll(List<Object> list) throws BaseException,ConstraintException {
    Session session = openSession();
    Transaction tn = null;

    try {
        tn = session.beginTransaction();
        for(Object object : list) {
            session.delete(object);
        }
        tn.commit();
    }
    catch (HibernateException e) {
        rollbackTransaction(tn);

        if (e instanceof ConstraintViolationException) {
            throw new ConstraintException(e.getMessage());
        }
        throw new BaseException(e);
    } finally {
        closeSession(session);
    }
}

```

Figure A. 5: Delete multiple records

A.6 Session Validate

Each function validates the user session and if it is expired automatically terminates user login. Code segment of session validate is appeared in Figure A. 6: Session Validate.

```
@Override
public String execute() throws Exception {
    try {
        if (!SessionUtil.validateSession(getHttpServletRequest())){
            addActionError(getText("error.sessionExpired"));
            return ERROR;
        }
        clearErrorsAndMessages();
        userSession = (UserSession) getHttpServletRequest().getSession().getAttribute("LOGIN_USER");
        stakeholder = new Stakeholder();
        list = getAllStakeholders();
        return SUCCESS;
    } catch (BaseException e) {
        addActionMessage(e.getMessage());
    }
    return SUCCESS;
}
```

Figure A. 6: Session Validate

A.7 Maintain Key-Value Pairs

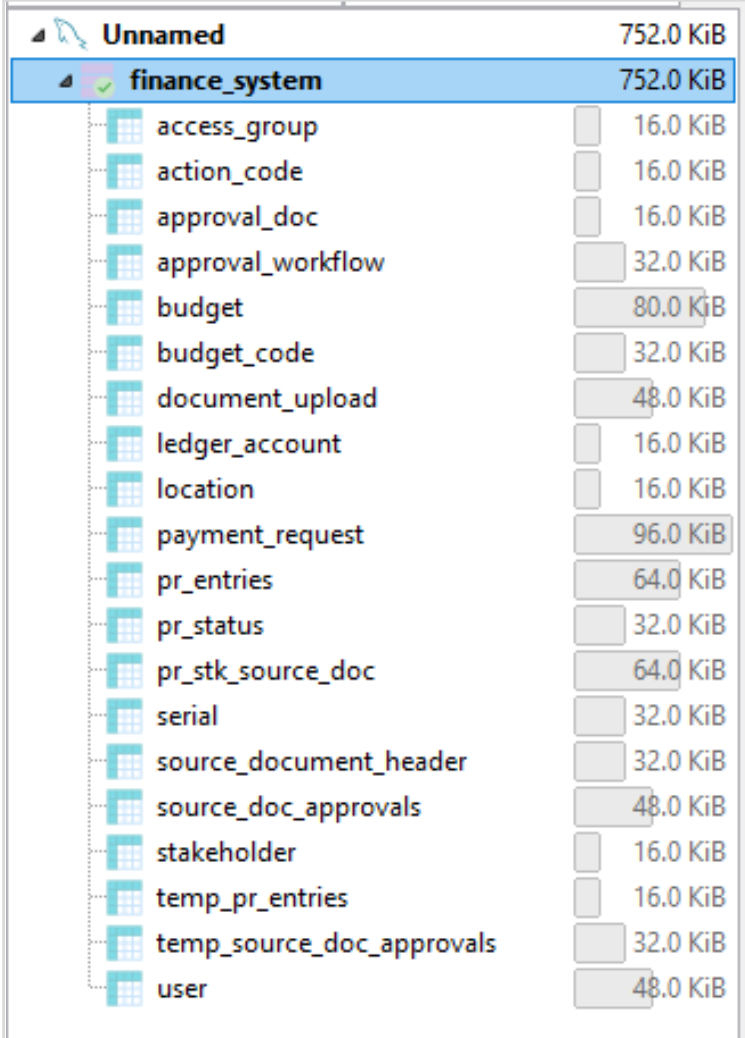
Property files were used to represent system labels, message contents, and input validations in this project. Some values of the property file content are represented in Figure A. 7: Property File

```
46 label.subject=Subject
47 label.prNo=Payment No
48 label.fromDate=From Date
49 label.toDate=To Date
50 label.paymentStatus=Payment Status
51 label.approvalLevel = Approval User
52
53 button.submit=Submit
54 button.clear=Clear
55 button.reset=Reset
56 button.next=Next
57 button.add=Add
58 button.back=Back
59 button.update=Update
60
61 message.addSuccess=SUCCESS : Record Successfully Created.
62 message.updateSuccess=SUCCESS : Record Successfully Updated.
63 message.deleteSuccess=SUCCESS : Record Successfully Deleted.
64 message.searchResultNotFound = Matching Results not found.
65
66 errors.required={0} is required.
67 errors.double={0} can't be zero.
68 errors.approvalsRequired=Approval details are required.
69 errors.twoFieldRequired=Please select/enter at least 2 criteria for report.
70 error.invalidUserPassword= Invalid user name and password.
71 error.loginError=Login Error.
72 error.approvalDocDeletion=ERROR : Approval document must be removed from all other related entities before delete
73 error.actionCodeDeletion=ERROR : Action Code can't delete. It have related entities.
74 error.budgetCodeDeletion=ERROR : Budget Code can't delete. It have related entities.
```

Figure A. 7: Property File

Appendix B - Database Table Structure (MySQL)

MySQL was used as a database management system in this project. There are 20 tables maintain in the system. Existing tables are illustrated in Figure B. 1: Database Tables



| Database | Size |
|---------------------------|-----------|
| Unnamed | 752.0 KiB |
| finance_system | 752.0 KiB |
| access_group | 16.0 KiB |
| action_code | 16.0 KiB |
| approval_doc | 16.0 KiB |
| approval_workflow | 32.0 KiB |
| budget | 80.0 KiB |
| budget_code | 32.0 KiB |
| document_upload | 48.0 KiB |
| ledger_account | 16.0 KiB |
| location | 16.0 KiB |
| payment_request | 96.0 KiB |
| pr_entries | 64.0 KiB |
| pr_status | 32.0 KiB |
| pr_stk_source_doc | 64.0 KiB |
| serial | 32.0 KiB |
| source_document_header | 32.0 KiB |
| source_doc_approvals | 48.0 KiB |
| stakeholder | 16.0 KiB |
| temp_pr_entries | 16.0 KiB |
| temp_source_doc_approvals | 32.0 KiB |
| user | 48.0 KiB |

Figure B. 1: Database Tables

Appendix C - Web Application UI

C.1 Introduction

This section represents some important system UI which not included in the implementation chapter.

C.2 URL of the System

Development URL displayed in Figure C. 1: System URL.

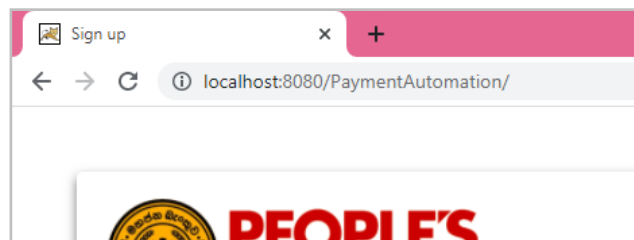


Figure C. 1: System URL

C.3 System Messages

System display several message types for success, error, info, and warnings. Those are represented in Figure C. 2: System Messages.

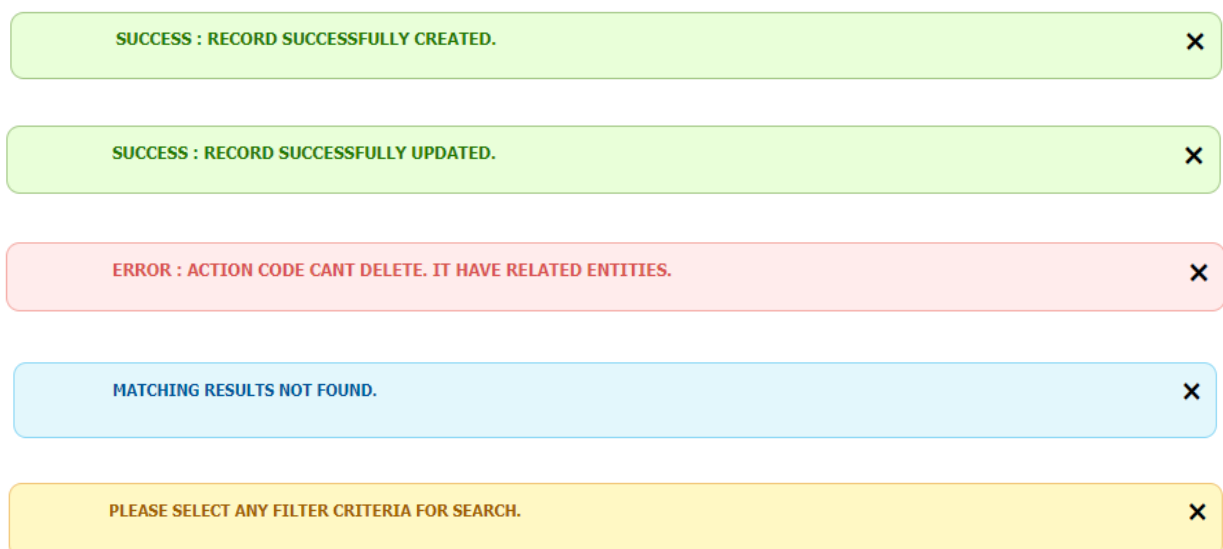
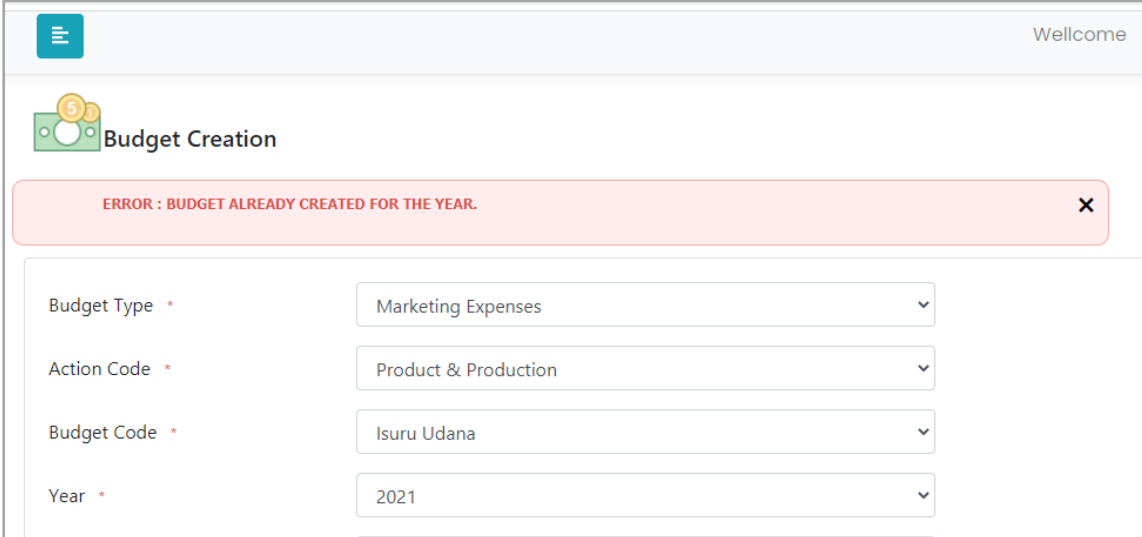


Figure C. 2: System Messages

All user functions validate for mandatory values, duplicates, and data integrity. Duplicates entry prohibited in budget function is displayed in Figure C. 3: Budget provisioning validation for duplicate entry.



The screenshot displays a web interface for 'Budget Creation'. At the top right, there is a 'Wellcome' greeting. Below the header, the page title is 'Budget Creation' with a green icon. A prominent red error message box states: 'ERROR : BUDGET ALREADY CREATED FOR THE YEAR.' Below this, the form contains four dropdown menus: 'Budget Type' (selected: Marketing Expenses), 'Action Code' (selected: Product & Production), 'Budget Code' (selected: Isuru Udana), and 'Year' (selected: 2021). Each dropdown menu has a small asterisk next to its label.

Figure C. 3: Budget provisioning validation for duplicate entry

C.4 Action Code UI

To simplify the budget defining budget codes are divided into action types. These action types can define through this UI. This is a function of the budget user. Existing action code list and edit screen of a record displayed in Figure C. 4: Action Code List and Edit Screen.

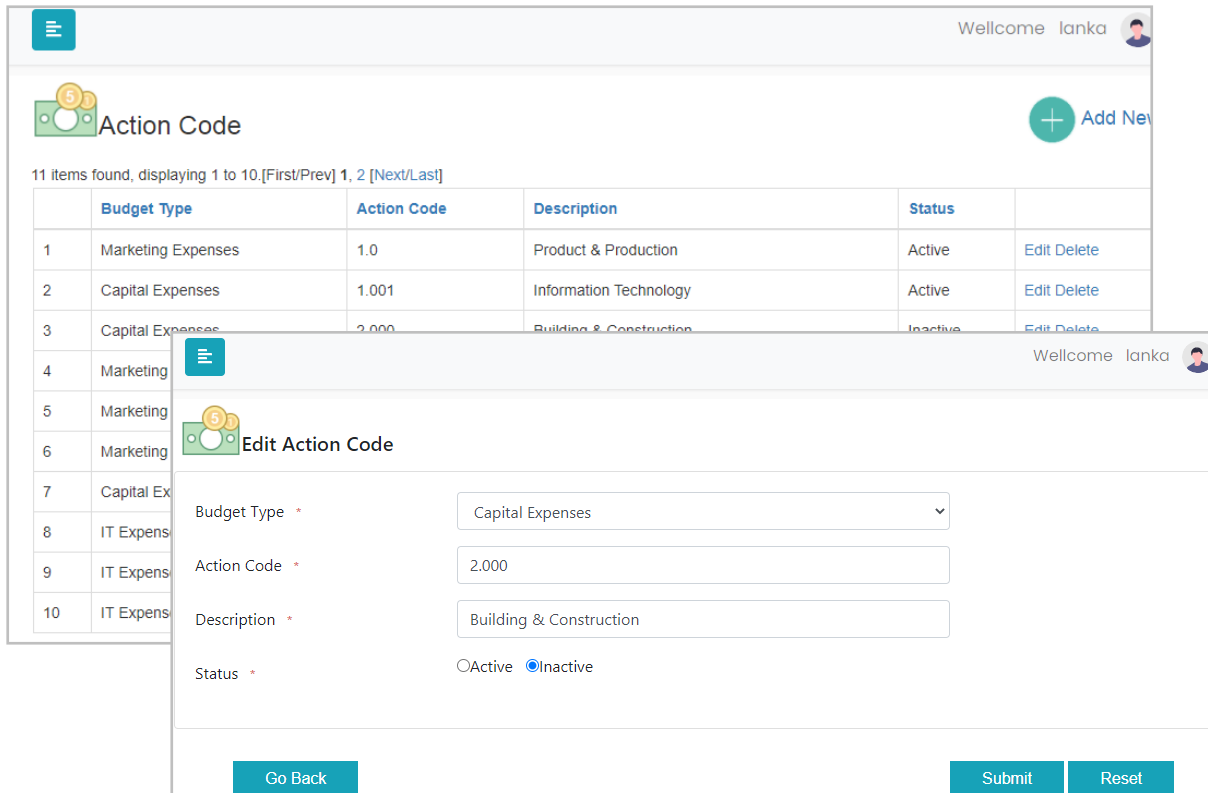


Figure C. 4: Action Code List and Edit Screen

C.5 Budget Code UI

Budget codes are used to define the budget. Each year budget values are allocating for these codes. So these budget codes are remaining each year, only provision amounts are changed. Budget code list and adding screen with validation shown in Figure C. 5: Budget Code List and adding UI.

Budget Code

+ Add New

11 items found, displaying 1 to 10. [First/Prev] 1, 2 [Next/Last]

| | Budget Type | Action Code | Budget Code | Description | Status | |
|----|--------------------|----------------------|-------------|------------------|--------|---|
| 1 | Marketing Expenses | Product & Production | 1.001 | Isuru Udana | Active | Edit Delete |
| 2 | IT Expenses | BUSINESS BENEFITS | 2.00 | Ongoing Projects | Active | Edit Delete |
| 3 | Marketing Expenses | Product & Production | 2.001 | Vanitha Vasana | Active | Edit Delete |
| 4 | Marketing Expenses | Product & Production | 3.00 | YES | Active | Edit Delete |
| 5 | IT Expenses | BUSINESS BENEFITS | 1.001 | New Solutions | Active | Edit Delete |
| 6 | IT Exp | | | | | |
| 7 | IT Exp | | | | | |
| 8 | Capita | | | | | |
| 9 | IT Exp | | | | | |
| 10 | Market | | | | | |

Export options

Wellcome lanka

Add Budget Code

Budget Type *

Action Code *

Budget Code *

Description *

Budget Type is required.

Action Code is required.

Budget Code is required.

Description is required.

Go Back

Submit

Clear

Figure C. 5: Budget Code List and adding UI

Appendix D – Implementation of MIS Reports

D.1 Introduction

The system allows the manager level users to access summary reports and to generate well-documented excel reports with a filtering facility. Manager level users are “Officer A”, “Officer B”, and “Super User” of the system. Access granted users can access report and chart module from the dashboard menu. There are four major report categories handled by the system. Those are,

1. Payment Summary Report
2. Budget Summary Report
3. Daily Reports
4. Charts – Monthly Budget Utilization (refer Figure 4. 25: Chart - Monthly Budget Utilization)

D.2 Payment Summary Report

The user should be able to direct to the payment summary report screen by clicking on the ‘Payment Summary Report’ from sub-menu of the reports menu. This report can generate any time with given filtering criteria. Filter criteria for payment summary report shown in Figure D. 1: Payment Summary Filter Criteria.

Payment Summary

Payment Request No

Cost Center

Status

Supplier/Beneficiary

Minimum Payment Amount

From Date

To Date

[Generate Excel](#)

Figure D. 1: Payment Summary Filter Criteria

This report has the following filter criteria.

- Payment Request No – Each request denotes a unique payment number.
Number format: department code + financial year + 6 digit serial number
This field expects the last sequence number only.
- Cost center - According to user location, this will be filtered. If the user location is finance department he/she can generate a report for entire departments or selected departments. Other users this will be automatically filled with respective to their Location
- Status - This provides the facility to generate a status-wise report. Possible payment status are draft, pending, approved, rejected, and canceled. This indicates whether the payments were paid, pending, or canceled.
- Supplier/Beneficiary – All stakeholders are listed here. Users can generate total payments made separately to the supplier/vendor wise.
- Minimum Payment Amount – Support for larger amount payment inquiries.
- From Date and To Date – The user can generate payments between these dates.

This report provides a facility to generate all requested payments and approved payments by the financial department. The above criteria are defined according to customer requirements. There is a huge number of daily payments handled by the finance department. So this report provides the best method to analyze bank outside payments.

System generated “payment summary” excel report is shown in Figure D. 2: Payment Summary Report.

| | | | | | | | | | |
|----|-------------------------------|------------------|--------------------|-----------------------------|---------------------|-------------------|-------------------|--------------------|---------------|
| 1 | Payment Summary Report | | | | | | | | |
| 2 | Payment No : | 2021-09-04 | Report Date : | 2021-09-04 | | | | | |
| 3 | Status : | Pending | User: | lanka | | | | | |
| 4 | Supplier/Beneficiary : | All | Cost Center: | 611 | | | | | |
| 5 | From Date | All | To Date: | All | | | | | |
| 6 | | | | | | | | | |
| 7 | Payment No | Subject | Budget Code | Supplier/Beneficiary | Payment Mode | Amount | Invoice No | Approval No | Amount |
| 8 | 2021-611-000035 | Account Campaign | YES | Revo - Revolution | Pay Order | 88,560.00 | INV 2021-2836 | B148-2021 | 20,000.00 |
| 9 | | | | | | | | E01256 | 30,000.00 |
| 10 | | | | | | | In 202996 | BD - 2021-2 | 32,000.00 |
| 11 | 2021-611-000036 | New opening | CORPORATE BANKING | Kalana Chathuranga | Pay Order | 50,500.00 | IN 2563 | E120-2021 | 50,500.00 |
| 12 | 2021-611-000037 | staff travelling | CORPORATE BANKING | WA Silva | Pay Order | 21,600.00 | kl2569 | E108-2021 | 20,000.00 |
| 13 | | | | | | 160,660.00 | | | |
| 14 | | | | | | | | | |

Figure D. 2: Payment Summary Report

D.3 Budget Summary Report

The user should be able to direct to the budget summary report screen by clicking on the ‘Budget Summary Report’ from the sub-menu of the reports menu. This can generate any time with given filtering criteria. Filter criteria for budget summary report shown in Figure D. 3: Budget Summary Filter criteria.

Budget Summary

Budget Type * ▼
Marketing Expenses

Action Code ▼
Product & Production

Budget Code ▼
Isuru Udana

Year ▼
2021

Generate Excel

Figure D. 3: Budget Summary Filter criteria

This report has following filter criteria.

- Budget Type – There are four major budget types are defined by the budget department. Those are,
 - Retail Banking Expenses
 - Capital Expenses
 - IT Expenses
 - Marketing Expenses

All the budget categories maintain under the above types. The system will generate summary reports for expenditures according to the above budget types for the system date or selected year.

- Action Code – There is a huge number of budget categories for each budget type. The system simplifies it by dividing it into action codes. These codes can define through the system by budget department users.
- Budget Code – Every payment is made under budget codes. The system provides the facility to defined budget codes. This report includes total payments made under each budget code.
- Year – Each budget is defined for a particular financial year. This enables year-wise filtering for budget summary

This report provides a facility to generate all the payments made under each budget. Users can identify total budget utilization for a year, remaining balances for previous years, budget variation throughout the year. This information will support the funding planning of the bank. It derives the amount of cash that will be needed to support operations. The budgeting information helps management to think about the business environment. Budgeting provides a means of controlling the income and expenditure of a business. This report gives a plan for spending and the budget can be revised if necessary.

Generated “Budget Summary” excel report is displayed in Figure D. 4: Budget Summary Report.

| Budget Summary Report | | | | | | | |
|-----------------------|--------------------|------|-------------|---------------------|-------------------|---------------------|----------------|
| Budget Type : | MARKETING_EXPENSES | | | Report Date : | 2021-09-04 | | |
| Action Code : | All | | | User: | Lanka | | |
| Budget Code : | All | | | Cost Center | 611 | | |
| Year : | All | | | | | | |
| Action Code | Budget Code | Year | Approval No | Allocated Amount | Utilized Amount | Balance Amount | Current Status |
| Product & Production | Isuru Udana | 2021 | BD - 2021 | 300,000.00 | 0.00 | 300,000.00 | Active |
| Product & Production | Isuru Udana | 2022 | E01256 | 285,000.00 | 0.00 | 285,000.00 | Active |
| Product & Production | Vanitha Vasana | 2021 | BD - 2021 | 100,000.00 | 0.00 | 100,000.00 | Active |
| Product & Production | YES | 2021 | BD - 2021 | 600,000.00 | 88,560.00 | 511,440.00 | Active |
| Product & Production | YES | 2022 | BD - 2021 | 400,000.00 | 0.00 | 400,000.00 | Active |
| SPONSORSHIPS & OTHERS | CORPORATE BANKING | 2021 | B148-2021 | 450,000.00 | 72,100.00 | 377,900.00 | Active |
| SPONSORSHIPS & OTHERS | CORPORATE BANKING | 2022 | A14-2020 | 66,000.00 | 0.00 | 66,000.00 | Active |
| | | | | 2,201,000.00 | 160,660.00 | 2,040,340.00 | |

Figure D. 4: Budget Summary Report

D.4 Daily Reports

The user should be able to direct to the daily transaction reports screen by clicking on the ‘Daily Reports’ from the submenu of the reports menu. Users should be able to generate daily reports on a daily basis. The user can generate daily reports by following Figure D. 5: Daily Report Screen.

The screenshot shows a web application interface for generating daily reports. At the top right, it says 'Wellcome lanka' next to a user profile icon. The main section is titled 'Daily Reports' and contains a form with two input fields: 'Report Name' with a dropdown menu currently showing 'Ledger Entries', and 'Cost Center' with an empty text box. Below these fields is a blue button labeled 'Generate Excel'.

Figure D. 5: Daily Report Screen

Filter options are “Report Name” and “Cost Center”. Those are described below.

- Report Name – There are 2 daily reports generate by the system.

1. Payment Report -

This report includes all payments generated within the system date. Each department officers can get information about their daily payments request and finance department officers can get all payment request detail regardless of the cost center/department. The following informations are represented in the report.

- Payment Request No
- Budget Code
- Request Status – This denote current status of payment requests.
- Payment Amount
- Requested User

2. Ledger Entries Report –

This report includes all ledger entries which are passed to the bank core system (SIBS) each day. These entries are passed when the approval workflow is completed. Although SIBS integration did not handle by the system, related double-entry details are maintained in the system. Each department officers can get this report respective to their department. Finance department officers can get the report for all departments. The following information are represented in the report.

- Account Name
- Account Type – Customer Account or Ledger Account
- Amount
- Debit/Credit
- Payment mode

- Cost Center/Department No – According to user location, this will be filtered. If the user location is finance department he/she can generate a report for all departments or a selected department. For other users, this will be automatically filled according to their location.