

DESIGN AND CONSTRUCTION OF WEB-BASED BUDGETING SUBMISSION INFORMATION SYSTEM AT CV EXPRESSA

Muhammad Hashfi Rafid Muttaqin
information Systems
Universitas Duta Bangsa
Surakarta, Indonesia
muhammadhasfi@gmail.com

Hanifah Permatasari
information Systems
Universitas Duta Bangsa
Surakarta, Indonesia
hanifah_permatasari@udb.ac.id

Dwi Hartanti
information Systems
Universitas Duta Bangsa
Surakarta, Indonesia
dwhartanti@udb.ac.id

Abstract— *CV Expressa faces problems in budget management due to a slow manual process, less transparent, and vulnerable to errors. This study aims to design a web-based budgeting information system to improve efficiency and accuracy. System development uses the SDLC Waterfall method (Analysis of Needs, Design, Implementation, Testing, Maintenance) with Laravel Technology (PHP) and PostgreSQL Database. Features include budget submission, tiered approval, reporting, and audit trail. Blackbox testing test results show all functions run optimally, supported by automatic notifications and printed reports. This system is proven to speed up the process, reduce errors, and increase budget transparency at CV Expressa.*

Keywords—*Information Systems, Budget Submission, SDLC Waterfall, Laravel, Efficiency.*

I. INTRODUCTION

Budget management or budgeting is the process of planning, allocating and controlling financial resources which aims to ensure the operational and financial stability of the company. Good budget management enables organizations or individuals to manage financial resources efficiently and effectively, estimate future financial needs, and consider risks and uncertainties in financial decision-making[1]. This process includes budget planning, approval, monitoring, and reporting. This stage requires accuracy and transparency for the allocation of funds according to business needs and priorities.

In its operational practices, CV Expressa faces challenges in budget management, especially the arrangement, monitoring and reporting of funds. The application process is still manual, resulting in slow approval, less transparent, and prone to error. This hampers decisions, reduces the efficiency of resource allocation, and has the potential to impact the company's performance and financial stability.

To overcome these problems, this study aims to design a Web-Based Budget Submission Information System that can accelerate the submission process, improve data accuracy, and facilitate monitoring and reporting. The method used is the Software Development Life Cycle (SDLC) with a Waterfall approach, which includes analysis, design, implementation, testing, and maintenance.

The implementation of this system is expected to improve the efficiency of budget management at CV Expressa, support productivity, and reduce errors. Furthermore, effective application of information technology can minimize unexpected failures and enhance system flexibility[2].

Unlike previous studies that primarily focus on budget input and report generation, this system introduces integrated multi-level approval workflows and real-time status tracking tailored to CV Expressa organizational structure, which constitutes the novelty of this research.

II. METHODS

Various methods were used in this research to support success, including:

A. Data Collection Methods

Data collection methods are the methods used by researchers to obtain relevant data, using several techniques, namely:

1. Primary data is data that comes directly from the research subject, the author collected data through direct observation at CV Expressa.
2. Secondary data is data obtained indirectly through public sources such as journals, research reports, articles, or other relevant documents.

B. Data collection methods

1. Observation: In this method, the author makes observations to obtain general data by looking, observing, and recording directly at CV Expressa.
2. Interview: This method is carried out by means of direct question and answer with Mr. Aditya Erda Syahrial, where the object of the research is to clarify the problems that exist when conducting observations.
3. Literature Study: Data collection was carried out through a review of books, journals and research results related to budgeting information systems and web-based system development.

C. System Development Methods

The system development approach adopted in this study is the System Development Life Cycle (SDLC), which consists of several phases to guide the system development process[3], namely:

1. Requirement

At this stage, the budget management system needs analysis is carried out using the PIECES approach (Performance, Information, Economy, Control, Efficiency, Services) to identify the weaknesses of the current manual system and detail the functional and non-functional needs of the system to be developed.

2. Design

This stage includes system design using UML diagrams such as Use Case and Activity Diagram to describe the submission, validation, and reporting flow. Class diagram is used to design the relationship between budget data tables, users, and approval status. Interface design is also carried out to facilitate input, access, and monitoring of the submission process.

3. Implementation

This stage includes system development using Laravel and PHP, with PostgreSQL as the database to store budget data, users, and approval status. Visual Studio Code is used as a text editor and code development tool.

4. Integration & Testing

System testing is carried out using the Blackbox Testing method to ensure that all core features including budget submission, validation, approval, and reporting function as intended. This testing focuses on verifying outputs based on given inputs without examining internal code structure, and aims to identify any functional errors or issues that could disrupt the application's performance and usability.

5. operation & maintenance

This is the stage where the software starts to be run by users in an operational environment. During this stage, maintenance is carried out periodically to ensure the system continues to run properly, including bug fixes, performance improvements, and the addition of new features according to user needs or system developments.

While the Waterfall model offers a structured and sequential approach to system development, it presents challenges in adapting to evolving user requirements. In contrast, Agile methodologies are inherently more flexible and iterative. However, considering the limited development timeframe and the absence of an agile-ready team environment during the internship period, the Waterfall model was deemed the most practical and manageable approach for this study.

III. RESULT AND DISCUSSION

A. Requirement

System weakness analysis is conducted to identify deficiencies in the current system and provide solutions through a new system. The method used is PIECES, which includes aspects of Performance, Information, Economy, Control, Efficiency, and Service.

Table 1. Pieces

Parameters	Old System	New System
Performance	Slow process due to manual (paper & email).	Faster process with web-based automated system.
Information	Incomplete data, difficult to track.	Structured, complete, and easily accessible information.
Economy	High costs due to paper usage and manual communication.	Reduced costs with digitization and faster processes.
Control	Weak control, frequent errors/data duplication.	Automated control with validation and audit trail.
Efficiency	Wasted time and effort on manual processes.	More efficient with automated input to approval.
Service	Slow service, frequent delays.	Improved service with notifications and process transparency.

B. Design

1. Usecase

A use case diagram illustrates the interactions between users (actors) and the system. In this study, the use case describes how users interact with the system during the budget management process[4]. The following is an explanation of several main use cases in the system.

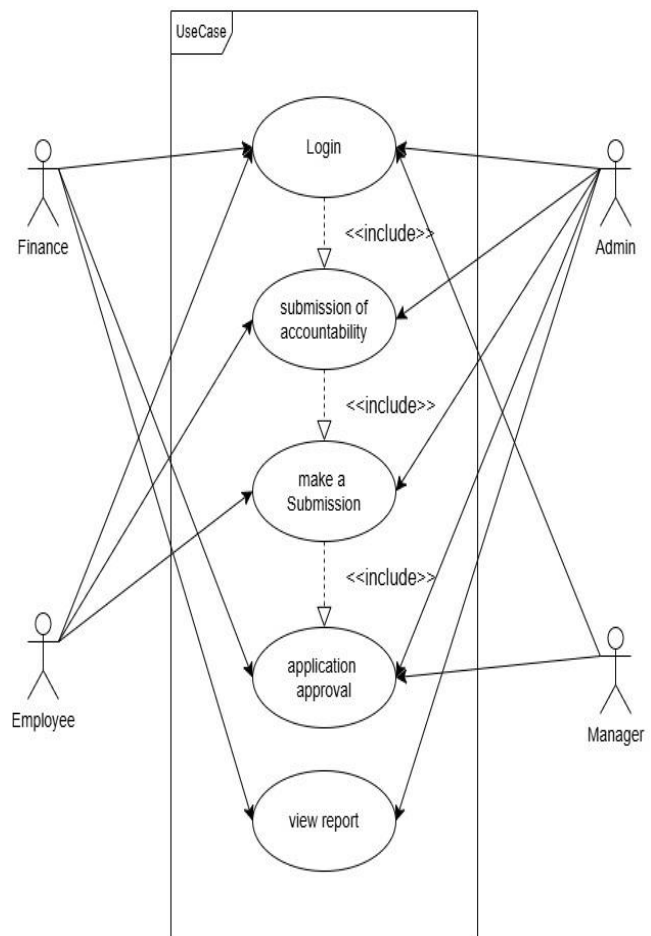


Fig 1. Usecase

2. Activity Diagram

The activity diagram describes the process flow and system activities available in the software[5]. This diagram specifically shows the actions or roles carried out by actors in the system. This explanation is structured to facilitate understanding of how the system works from the perspective of the user or actor involved.

a. Submission Activity

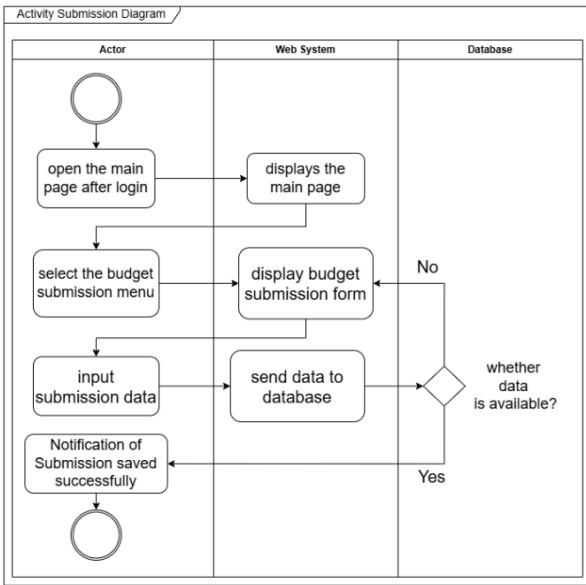


Fig 2. Submission Activity

Submission is done by Admin and Employees who have logged into the system. The actor selects the submission menu, fills out the form, and saves it. The system then verifies, saves the data to the database, and provides a notification that the submission was successfully saved.

b. Approval activity

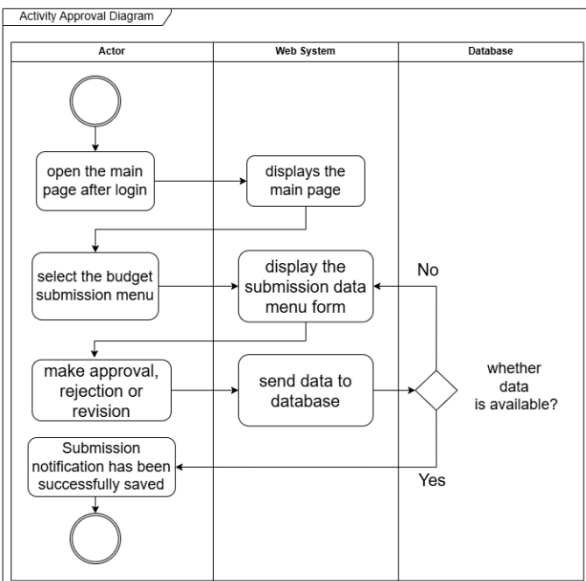


Fig 3. Approval activity

Submission approval is done by Admin, Manager, and Finance who have logged into the system. The actor selects the submission menu, makes a decision (agree, reject, or revise), then saves the data. The system saves the decision to the database and displays a notification as confirmation.

c. Accountability Activity

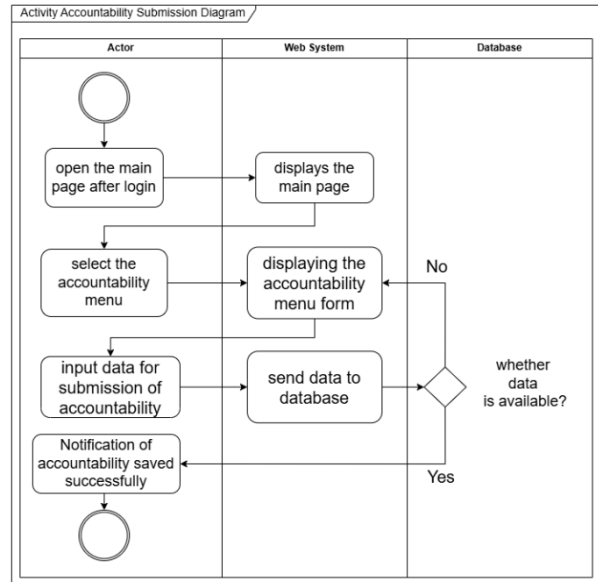


Fig 4. Accountability activity

Accountability submission is done by Admin and Employees who have logged in and have access. The actor fills in the required data, then saves it. The system verifies, saves the data to the database, and provides notification that the submission has been successfully saved.

d. Report Activity

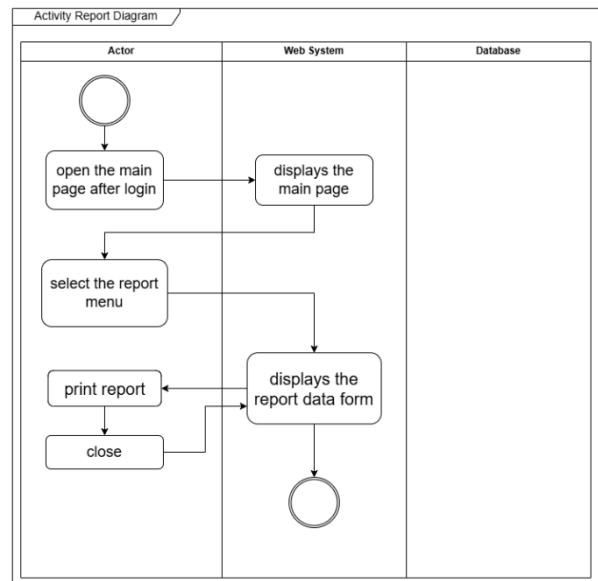


Fig 5. Report Activity

Reports are done by Admin and Finance who have logged into the system. The actor selects the report menu

to view and print the available data, then exits the system after completion.

3. Class Diagram

A class diagram is a schematic representation of the classes that will be constructed to build the system[6]. The following is a class diagram for the application, which can be seen in the image.

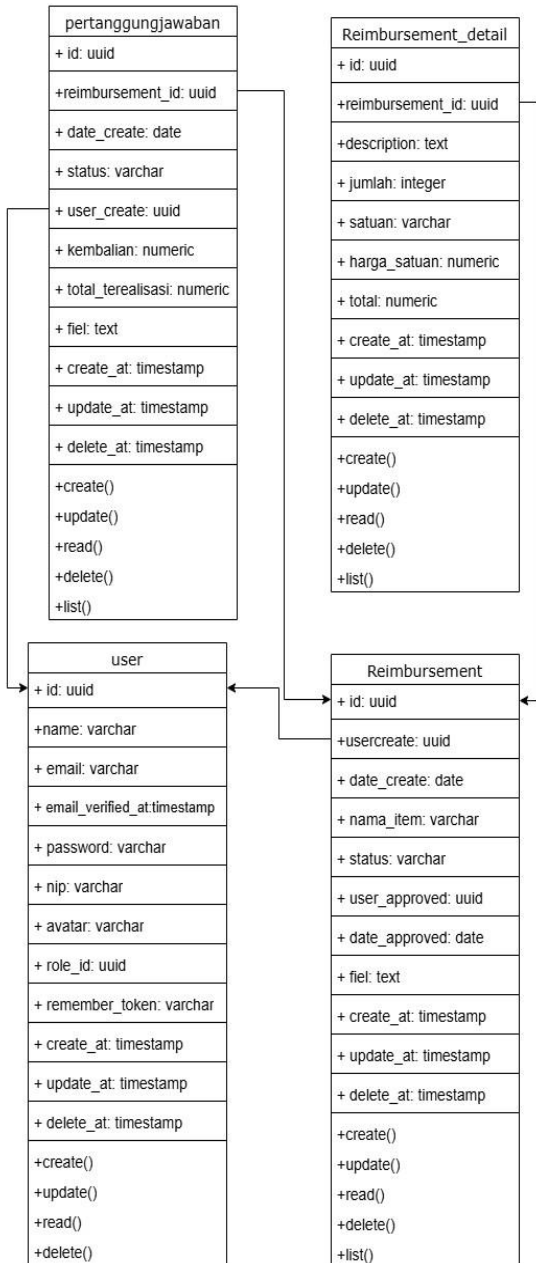


Fig 6. Class Diagram

C. Implementation

1. Login page

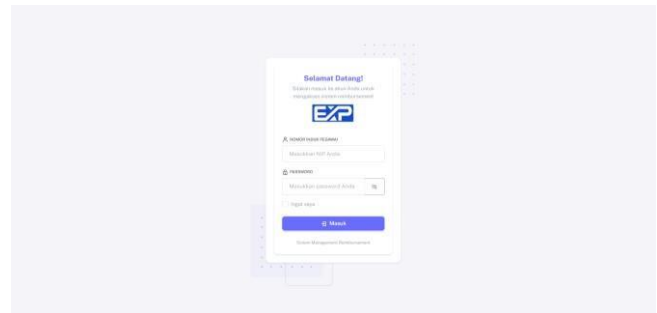


Fig 7. Login Page

Before users can access the system. Users are asked to enter a username and password according to their respective access rights.

2. Form for Submission

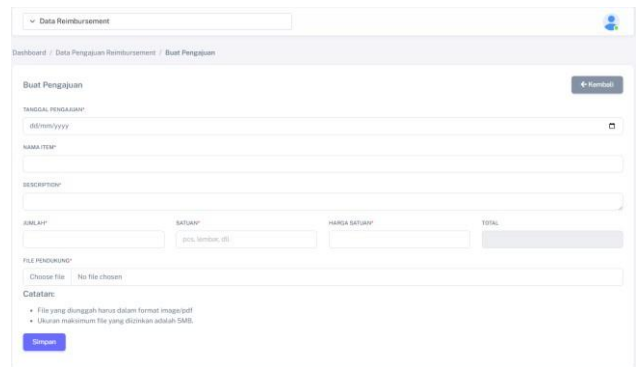


Fig 8. Form for Submission

The Create Application page is used to submit a reimbursement by filling in the application date, application name, costs and a complete description.

3. Submission Page

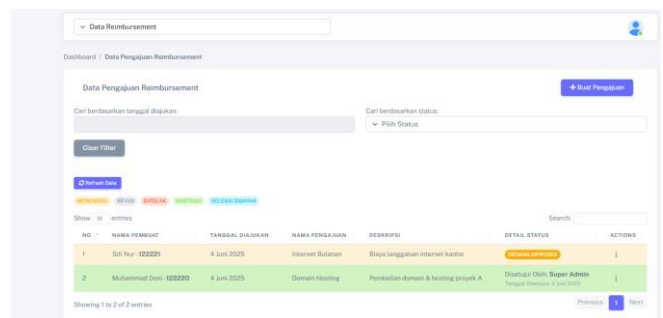


Fig 9. Submission Page

The Submissions page displays a list of submissions with complete information, a search feature, filters by date or status, and a button to create a new submission. Submission statuses are color-coded for easy identification.

4. Accountability Form

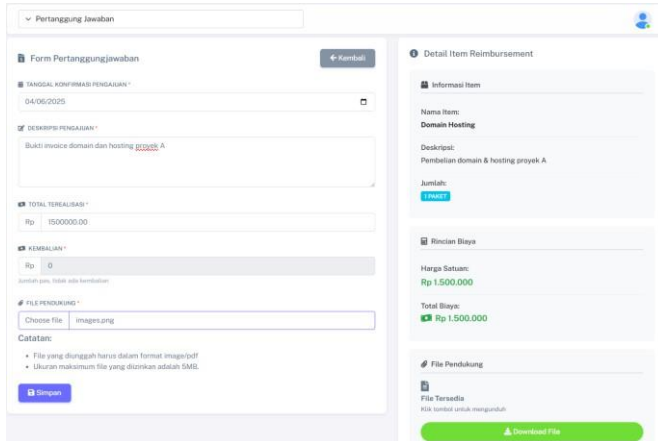


Fig 10. Accountability Form

This form is used to create a realization of funds from a reimbursement application, including date, description, total funds, returns, and supporting files. The side panel displays item details and cost details.

5. Accountability Page

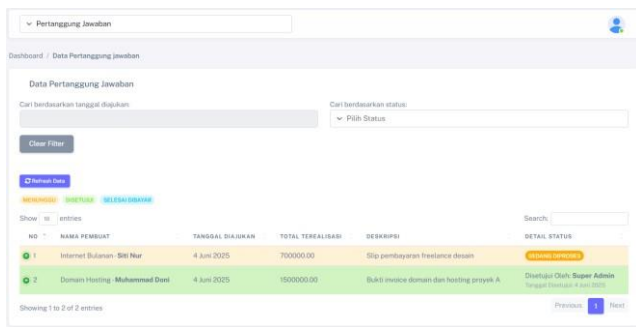


Fig 11. Accountability Page

This page displays a complete list of user responsibilities with basic information and process status. Equipped with search and filter features to facilitate data monitoring and verification.

6. Report Page

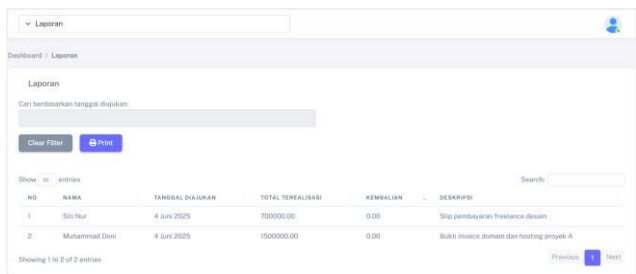


Fig 12. Report Page

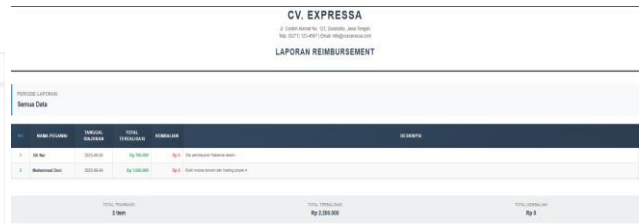


Fig 13. Report

On the report page, the completed submission data is displayed. There is a table that displays submission information, such as number, name, date submitted, submission name, costs, and description and there is a print button to print the report.

D. Testing

1. Blackbox Testing

System testing is a crucial phase in the software development process, aimed at evaluating whether the system operates as expected, in terms of both performance and user interface[7].

Table 2. Login Testing

No	Test Case	Input	Expected Output	Status
1	Successful login	Valid username and valid password	User is redirected to dashboard	success
2	Invalid credentials	Incorrect username or password	Invalid username or password	success
3	Empty field	Username and/or password left empty	"Username & password required"	success
4	Invalid special characters	Username not allowed	"Input invalid or blocked"	success

Table 3. Form for Submission Tetsing

No	Test Case	Input	Expected Output	Status
1	Submit valid data	All fields filled correctly	"Data saved successfully"	success
2	Submit empty form	All fields empty	Required field errors	success
3	"Item Name" missing	All filled except item name	"Item Name is required"	success
4	Letters in "Quantity"	Quantity = "abc"	"Quantity must be a number"	success
5	Unsupported file type	File = .exe / .docx	"File must be in image/pdf format"	success
6	Auto-calculate total	Quantity = 2, Price = 100000	Total auto-calculated = 200000	success
7	No file uploaded	All filled, no file attached	Error shown if file is required	success

Table 4. Submission Page Testing

No	Test Case	Input	Expected Output	Status
1	View submissions	Open submission page	List shown with info & status color	Success

2	Search	Enter keyword	Only matching items shown	Success
3	Filter	Choose status/date range	Submissions filtered accordingly	Success
4	New submission	Click New Submission	Form displayed	Success

Table 5. Accountability Form Testing

No	Test Case	Input	Expected Output	Status
1	Submit form	Enter data, click Save	Form submitted & saved	success
2	Required field check	Leave fields empty	Validation warning shown	success
3	View item details	Open reimbursement form	Side panel shows name, description, total	success
4	Download file	Click "Download File"	File downloaded successfully	success
5	Download uploaded file	Download from side panel	File downloads successfully	success

Table 6. Accountability Page Testing

No	Test Case	Input	Expected Output	Status
1	View records	Open page	User data table displayed	success
2	Status color	Status = "IN PROCESS"	Yellow label displayed	success
3	Filter by status	Select from dropdown	Table shows matching records	success
4	Keyword search	Enter name/description	Table filters by keyword	success
5	View details	Click "Detail"	Viewable if available	success

Table 7. Report Page Testing

No	Test Case	Input	Expected Output	Status
1	View completed data	Open report page	Table shows completed reimbursements	success
2	Search reports	Enter keyword	Matching results displayed	success
3	Print report	Click "Print"	Printable view shown / printed	success
4	View table columns	Check table rows	Columns: No, Name, Date, Amount, etc.	success
5	Pagination	Click "Next" (>10 rows)	Next page displayed	success

2. System Impact Evaluation

To assess the system's effectiveness, a comparison was conducted between the conditions before and after implementation. The evaluation focuses on processing time, error rate, and workflow efficiency.

Table 8. evaluation table

No	Aspect	Before System	After System	Efficiency
1	Budget submission duration	2-3 days	4-6 hours	~75% faster
2	Input error frequency	3-5 times/month	0-1 times/month	Significantly reduced

3	Approval process	Manual (email)	Automated in system	More efficient
4	Status tracking	Not available	Real-time	Transparent
5	Document printing	Required	Not needed	Paperless

To further assess the practical value of the developed system, a usability validation was conducted involving five end users from various departments within CV Expressa. Participants consisted of administrative personnel, finance staff, and a managerial-level employee, all of whom interact directly with the budgeting workflow.

Each user was asked to complete a full budgeting submission scenario using the web-based system, including inputting data, reviewing notifications, and following the approval status. Observations and short interviews were used to gather feedback regarding ease of use, accuracy, and overall experience.

The majority of users expressed satisfaction with the interface and system responsiveness. Four out of five respondents stated that the system reduced submission time and improved their visibility into approval stages. One respondent suggested adding color indicators to better differentiate status levels. This validation indicates that the system meets user expectations and lays a solid foundation for broader implementation within the organization.

E. Maintenance

Information system maintenance plays an essential role in ensuring that the developed applications continue to function optimally, meet evolving user needs, and adapt to changes in technological and operational environments[8].

1. Hardware Maintenance

Routine maintenance of hardware such as servers, client computers, and network devices is required to maintain system stability. This activity includes physical cleaning, checking network connections, replacing damaged components, and ensuring that the cooling system is working properly so that the device temperature remains stable.

2. Software Maintenance (System Applications)

Web-based applications that have been developed using Laravel and PHP require regular maintenance. This includes:

- Bug fixes: if any logical errors or inappropriate displays are found, they must be fixed immediately.
- System updates: updates are made to ensure the system remains compatible with the latest versions of PHP, Laravel, and other supporting libraries.
- Addition of features: if there are new needs from users or changes in business processes, then additional development is carried out in a structured manner.

3. Database Maintenance

The PostgreSQL database used as the primary storage for submission, user, and report data needs to be maintained properly:

- a. Routine backup: done every day/week to prevent data loss when a system failure occurs.
- b. Query and index optimization: ensuring that every data request is fast and efficient.
- c. Cleaning of invalid or duplicate data: to maintain consistency and accuracy of information in the system.

IV. CONCLUSION

This research successfully developed a web-based Budgeting Submission Information System using the SDLC Waterfall method to address inefficiencies in CV Expressa's manual budgeting process. The system significantly improved the efficiency of budget submissions, reducing processing time by approximately 75% and minimizing manual errors. Blackbox Testing confirmed that all key features submission, approval, and reporting functioned optimally. Positive user feedback also supports the system's usability and practicality across roles. Beyond these improvements, the system introduces a novel integration of multi-level approval workflows and real-time status tracking tailored to the company's organizational structure, distinguishing it from conventional budgeting systems.

REFERENCES

- [1] Y. A. P. Nst, and Nurlailah, 'Efektivitas Pengelolaan Anggaran Badan Penyelenggara Jaminan Sosial (Bpjs) Kesehatan Kota Kisaran', *SEIKO: J. Manag. Bus.*, vol. 6, no. 2, 2023.
- [2] N. S. D. Hapsari, 'Penerapan Teknologi Enterprise Resource Planning Menggunakan Modul Inventory Management dengan Metode FIFO Berbasis Website Pada CV Pandan Mas Sampit', *Terapan Inform. Nusant.*, vol. 2, no. 6, 2021.
- [3] I. Mulyana, D. Cahyadi, and S. N. Faujiah, 'Implementasi Sistem Informasi Pembayaran SPP Berbasis Web dengan Metode SDLC Pada SMA Al Mahfirah', *J. Al Muslim Inf. Syst.*, vol. 3, no. 1, 2024.
- [4] Roby Setiawan, Agus Nugroho, and B. Redhy Putra, 'Development of Platformer Game "Adventure Drake" Using Finite State Machine Method', *Media J. Gen. Comput. Sci.*, vol. 1, no. 1, pp. 29–32, Dec. 2023, doi: 10.62205/mjgcs.v1i1.10.
- [5] N. Harahap, M. D. Irawan, and Y. Hanifiah, 'SIPENMA: Pemodelan UML dan Uji BlackBox pada Perancangan Sistem Pendaftaran Magang', *J. Sist. Inf.*, vol. 6, no. 2, 2022.
- [6] R. E. Pambudi, Hartono, and Faris, 'Optimalisasi Proses Pemesanan Jasa Servis Komputer dan Laptop melalui Aplikasi Web Menggunakan Framework Codeigniter', *Sienna*, vol. 5, no. 2, pp. 122–141, Dec. 2024, doi: 10.47637/sienna.v5i2.1638.
- [7] A. C. Praniffa, A. Syahri, F. Sandes, U. Fariha, and Q. A. Giansyah, 'Pengujian Black Box Dan White Box Sistem Informasi Parkir Berbasis Web', *J. Test. Implementasi Sist. Inf.*, vol. 1, no. 1, 2023.
- [8] S. Burhani, S. M. Amir, and I. Gaffar, 'Perancangan Sistem Informasi Pemeliharaan Fasilitas Pelabuhan Perikanan dengan Pendekatan Preventif Maintenance', *Bul. Sist. Inf. Dan Teknol. Islam*, vol. 5, no. 4, pp. 293–303, Dec. 2024, doi: 10.33096/busiti.v5i4.2472.