

Cek EBMTJ 182

Fecha de entrega: 28-jul-2026 05:06a. m. (UTC-0500)

Identificador de la entrega: 3004647917

Nombre del archivo: Cek_Ebmtj_182.docx (992.42K)

Total de palabras: 2391

Total de caracteres: 14939

Development Of a Website-Based Inventory Loan Management Information System Using the First Come First Served (FCFS) Algorithm in An IT Service Provider Company

ABSTRACT

An IT service provider encountered difficulties handling inventory borrowing, which was still conducted traditionally via paper forms and spreadsheets. This traditional system resulted in documentation inaccuracies, data inconsistencies, protracted approval procedures, and borrowing disputes when numerous employees need the same things concurrently. This study sought to create a web-based inventory borrowing management information system utilizing the First Come First Served (FCFS) algorithm to ensure equitable and transparent management of borrowing priorities. The research methodology employed descriptive analysis, with observations and interviews, to ascertain system requirements. The system development utilized Agile Scrum methodology with sprint divisions to ensure outcomes aligned with user expectations. The system is structured as a web application utilizing the Laravel framework and MySQL database, with user identification, real-time dashboards, digital borrowing forms, a first-come, first-served queue system, automated notifications, and transaction reporting. The execution of the FCFS algorithm enables the system to log timestamps for each request and to process the queue in the order of submission. The research findings indicate that the system can resolve manual recording problems, deliver real-time item availability data, impartially manage borrowing priorities, and improve the company's operating efficiency. This system facilitates the digital transformation of inventory management by providing an accessible, precise, and reliable solution that enhances operational efficiency.

Keywords: Management Information Systems, Inventory Lending, FCFS Algorithm, Website, Agilesrum, Digital Transformation.

INTRODUCTION

The global ascent of digital transformation has profoundly altered the management of assets and resources by enterprises. A Gartner survey (2024) reveals that more than 70% of worldwide firms are transitioning to digital management systems to improve efficiency, transparency, and operational precision. This transition is integral to the Industry 4.0 epoch, which prioritizes the amalgamation of information technology, automation, and real-time data within company operations.

An IT service company is having difficulty maintaining its internal inventory. Notwithstanding the emphasis on digitalization, the inventory borrowing procedure is still executed manually using physical forms and spreadsheets. This scenario leads to

documentation inaccuracies, delays in lending procedures, and disputes among staff regarding simultaneous requests for identical things. ³ The First Come First Served (FCFS) algorithm offers a viable resolution for managing borrowing priority concerns. This algorithm operates on the idea of processing each request in the sequence of its arrival, without bias. (Atencia-Mckillop et al., 2024; Prakash et al., 2018) demonstrate that FCFS can diminish conflict levels and enhance the equity of resource allocation in industrial queue systems.

Fundamental Concept of a System, a system is an assemblage of pieces or components that interact to accomplish a particular objective. Gumanti, (Setiyawan & Sudarmilah, 2023) define a system as a collection of components that collaborate inside a specific framework to address a given problem or attain an objective.

Fundamental Concept of Information, information is data that has been processed to convey meaning and significance in decision-making. (Nazri et al., 2024) asserts that the data generated by a management information system is essential for facilitating managerial operations, since it underpins the formulation of precise judgments.

⁸ Management Information System, a Management Information System (MIS) is a system intended to manage, process, and present information to facilitate decision-making inside an organization" (Rahayu & Voutama, 2024). Management Information Systems amalgamate information technology with business processes to improve operational efficiency and data precision. According to (Christian & Voutama, 2024), Management Information Systems (MIS) comprise components including hardware, software, databases, procedures, and human resources that collaborate to fulfill corporate objectives.

Concept of Inventory and Inventory Management, research by (Fortuna Alfarisi et al., 2023) demonstrates that a web-based borrowing system is more efficient than manual recording because it can incorporate automatic notifications, data validation, and item status tracking. This system is accessible from any location, enabling users to submit borrowing requests. ³ First-Come, First-Served (FCFS) Algorithm. The First Come First Served (FCFS) algorithm is a scheduling technique that prioritizes processes or requests according to their arrival time (Nadhia & Suwarno, 2023). This algorithm prioritizes users who make requests first in the context of inventory borrowing.

⁵ Unified Modeling Language (UML), the Unified Modeling Language (UML) is a standardized visual modeling language used for the design and documentation of software systems, especially object-oriented systems. The document encompasses various diagram types: the Use Case Diagram illustrates key elements, including actors, use cases, and their interrelations, facilitating the analysis and design of system requirements; the Activity Diagram delineates the progression of processes or activities within the system, from initiation through decision-making to conclusion; the Sequence Diagram portrays the temporal sequence of messages exchanged between use cases; and the Class Diagram represents the system's architecture in terms of classes, attributes, methods, and inter-class relationships.

¹ Black-box Evaluation, black-box testing is a methodology for assessing software functionality without knowledge of its internal architecture. Testing is performed from the user's viewpoint, concentrating exclusively on the system's input and output. This method

does not necessitate access to or familiarity with the program's code, hence simplifying execution and eliminating the need for substantial technical preparation. It is especially appropriate for evaluating if the system functionalities adhere to standards (Arianti & Maulana, 2024).

White-box Testing, white-box testing is a software testing technique that examines the internal architecture and logic of a program's code. This method allows the tester to possess a comprehensive understanding of the code structure, enabling the examination of every logical path, branch, and iteration within the program. Testing is performed by transforming activity diagrams into flowgraphs, computing cyclomatic complexity, and creating test scenarios derived from separate paths. White-box testing is exceptionally proficient at identifying concealed problems within the program's architecture, including those not apparent through the user interface (Yusuf & Dermawan, 2025).

Cyclomatic complexity calculation using the formula: $V(G) = E - N + 2$
Where: $V(G)$ = Cyclomatic Complexity, E = Number of Edges (connecting lines), N = Number of Nodes (process points)

Laravel Framework, laravel exhibits a more refined and articulate design. It provides a multitude of capabilities that facilitate PHP developers in efficiently and swiftly constructing web applications with robust tools and architecture. The Model View Controller (MVC) is a software development architectural pattern that divides programs into three components: Model, View, and Controller. This methodology organizes the application to enhance code reutilization and facilitate application upkeep. In the MVC architecture, application components are categorized into three segments: The Model, which embodies the data architecture and business logic of the application. The View is the output representation of a model or the user interface perceived by users. The Controller is the component that receives user input and converts it into commands for the model and/or view.

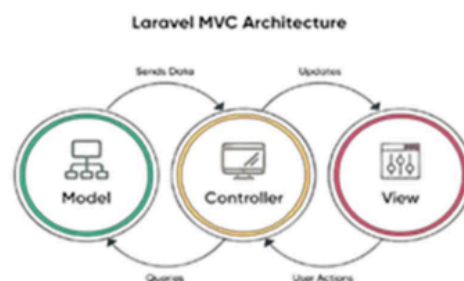


Figure 1. MVC Workflow

This project seeks to create a web-based inventory borrowing management information system that uses the FCFS algorithm to equitably and openly manage borrowing priority at an IT service provider.

METHOD

The employed research method is descriptive analysis, offering a systematic overview of customer requirements and the operational business processes within the IT service provider

organization. The process includes a literature review in which the team examines theories and prior research relevant to management information systems, inventory management, and the Agile Scrum methodology. Observations of the inventory management processes at the IT service provider are conducted, and interviews with relevant stakeholders are conducted to gather information on system requirements. Ultimately, data analysis elucidates the findings from both observations and interviews to formulate a solution that satisfies user requirements.

Development Method, the system development employs the Agile methodology, using the Scrum framework, selected for its adaptability, emphasis on team collaboration, and ability to adapt to evolving requirements incrementally. The Scrum process comprises several stages: the Product Backlog, which comprises a compilation of system features and requirements; Sprint Planning, during which the team selects the backlog items to be tackled in a particular sprint; and Sprint Execution, during which features are developed in accordance with the plan. Furthermore, the Daily Scrum functions as a concise meeting to assess work progress, whereas the Sprint Review concentrates on evaluating outcomes with stakeholders. The Sprint Retrospective facilitates a reflective process for enhancements in the subsequent sprint.

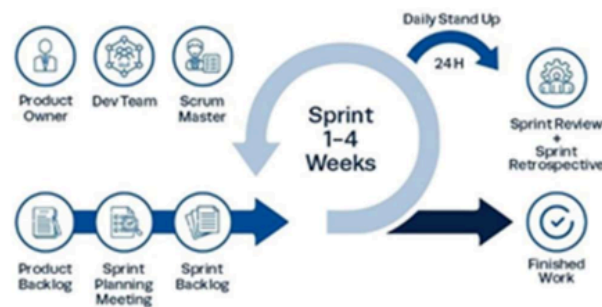


Figure 2. Agile Scrum Method.

RESULTS AND DISCUSSION

Observations and interviews conducted at an IT service provider on May 20, 2025, revealed multiple difficulties inside the manual inventory borrowing system. Employees currently complete physical forms or email the administration to request products, resulting in delayed response times because the admin must manually verify requests. Furthermore, inventory data is maintained in Excel files on the administrator's local computer, rendering this method susceptible to human error, including erroneous data entry, duplicate entries, or inadvertent deletions that are challenging to detect. The administrator must conduct physical inspections of the warehouse to assess the condition of objects, leading to inconsistencies between administrative records and actual conditions; frequently, returned items are documented as borrowed. Moreover, in the absence of a definitive priority mechanism for simultaneous employee requests for the same item, the administrator struggles to render equitable judgments, thereby fostering workplace friction and hindering productivity.

Design, a Use Case is a diagram in Unified Modeling Language (UML) that illustrates the interactions between actors (users or external systems) and the system under development. A Use Case models the system's functional requirements from the user's viewpoint. The following is the use case diagram developed for the Inventory Borrowing System at the IT service provider organization:

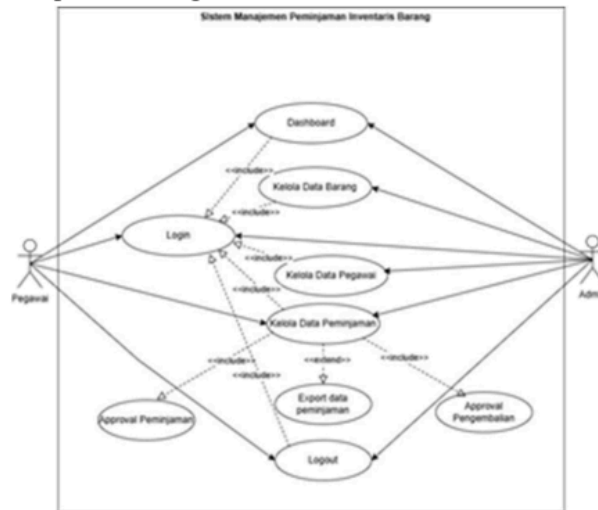


Figure 3. Use Case Diagram for Inventory Loaning

Activity Diagram, the following activity diagram depicts the workflow of a menu system within the sales information system application:

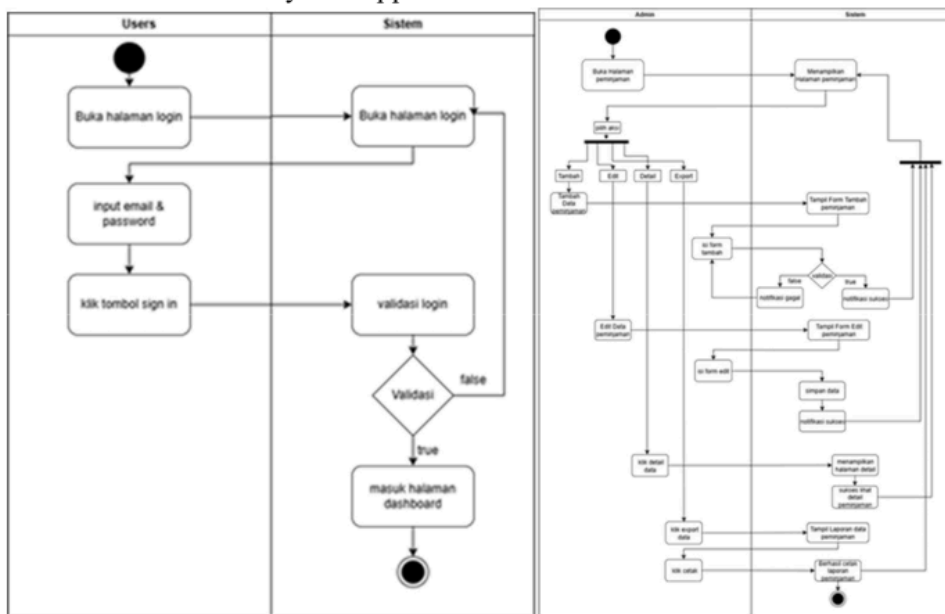


Figure 4. Activity Diagram Login User and Loan Activity Diagram (Admin)

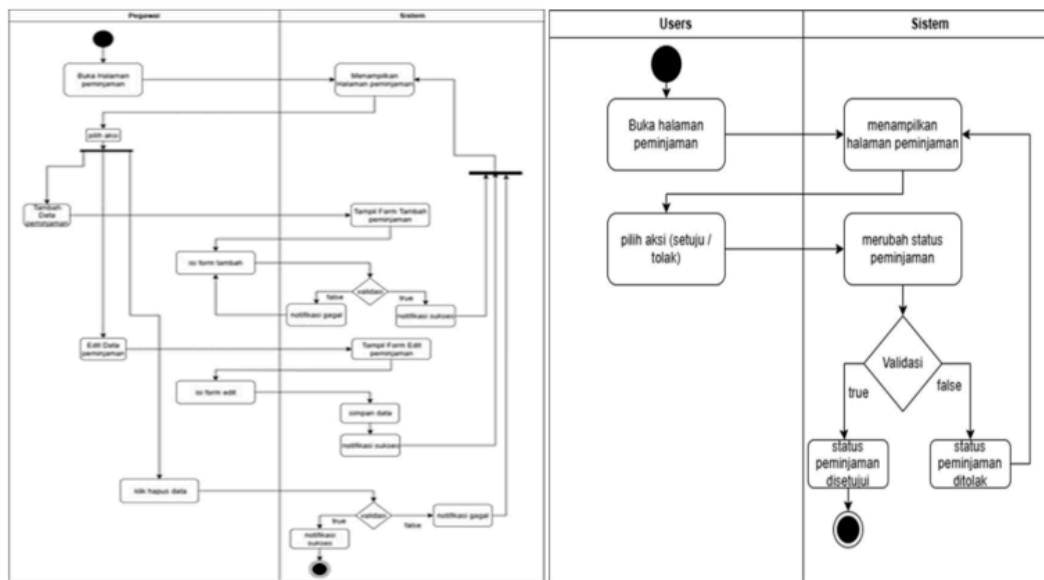


Figure 5. Loan Activity Diagram (Employees) and Loan Approval Activity Diagram

2

Sequence Diagram, a Sequence Diagram is a diagram within the Unified Modeling Language (UML) that illustrates the sequential flow of interactions among objects in a system over time. The vertical lines in the diagram denote the progression of time from top to bottom, and the horizontal arrows illustrate the communication transferred between actors or objects. Consequently, the sequence diagram elucidates the operation of a feature or use case from inception to completion. The following is the sequence diagram generated for this system:

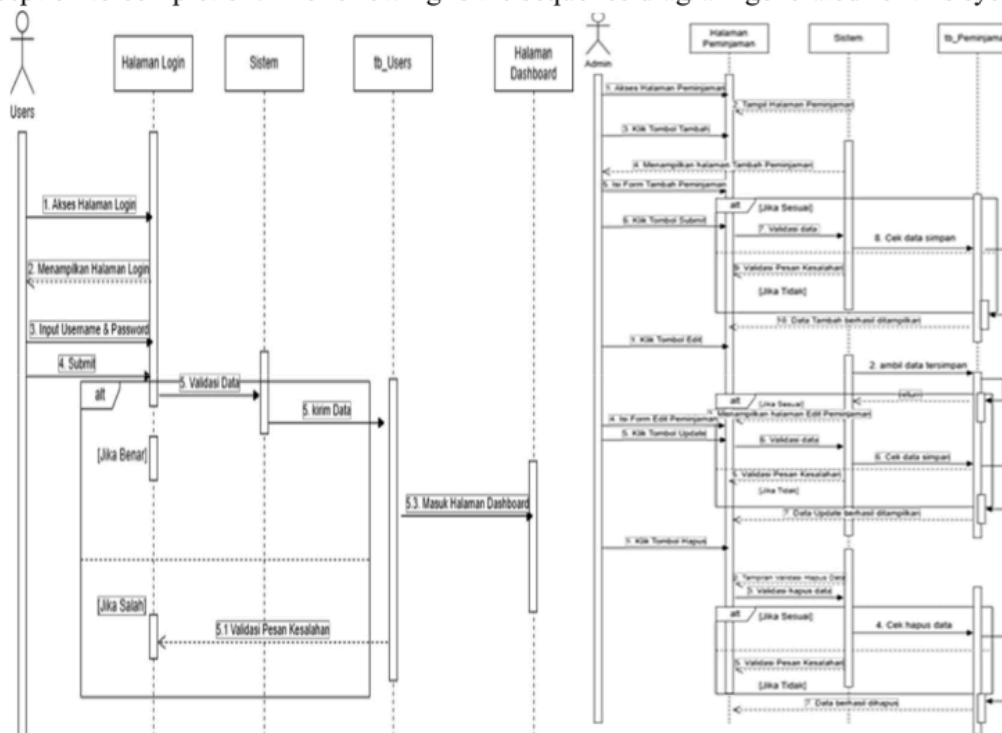


Figure 6. User Login Sequence Diagram and Sequence Diagram of Loan (Admin)

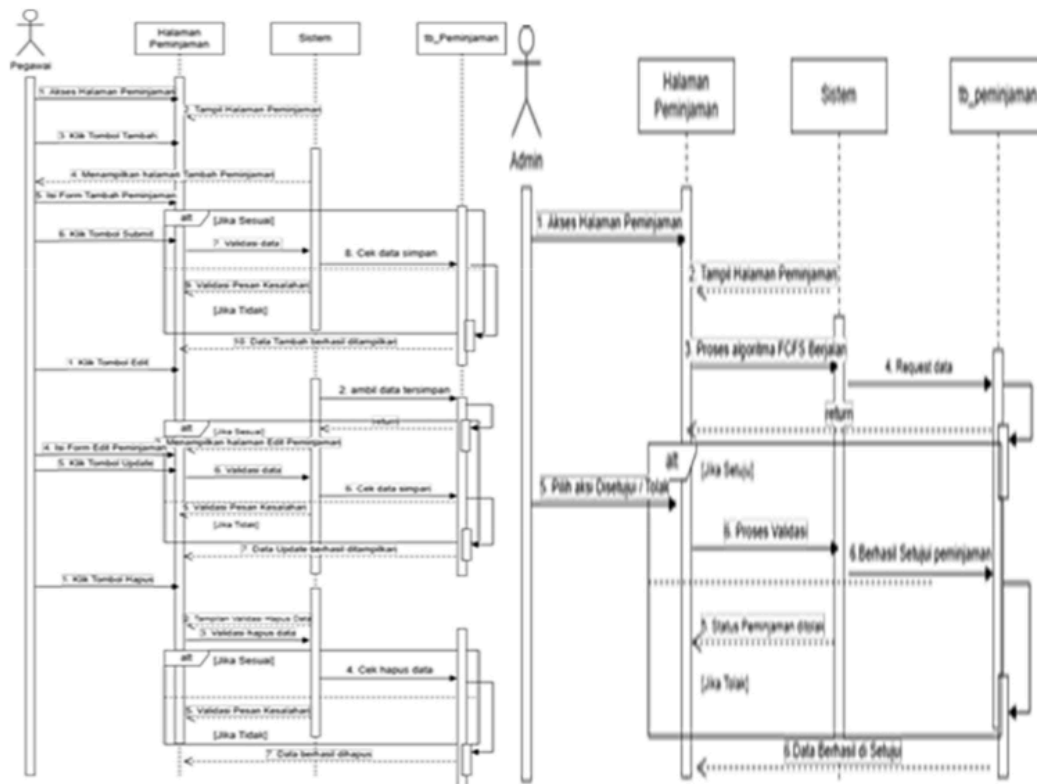


Figure 7. Sequence Diagram of Loans (Employees) and Loan Approval Sequence Diagram

Class Diagram, a Class Diagram is a UML representation that illustrates the static architecture of a system. The diagram illustrates the system's classes, including their attributes, methods, and interrelationships. A class diagram illustrates the interconnections among the system's components. Sommerville (2020) asserts that class diagrams facilitate clearer comprehension of object-oriented system architecture for developers, thereby simplifying implementation in program code.

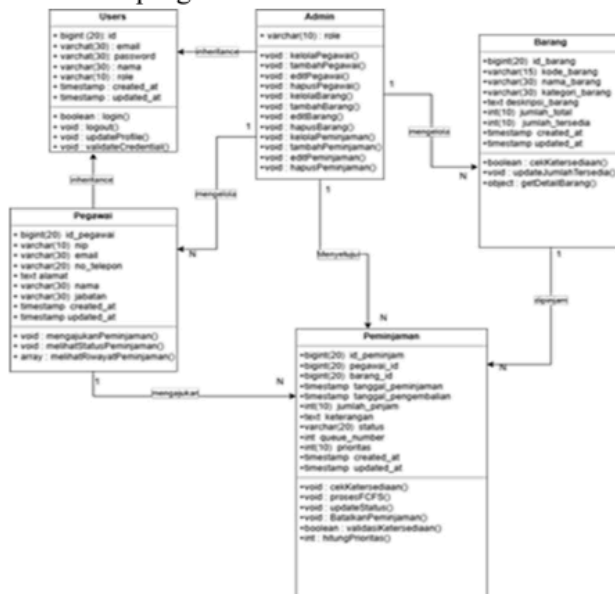


Figure 8. Class Diagram

Database Implementation, the discourse on the table structure in the database outlines the comprehensive configuration of the tables in the phpMyAdmin database, informed by the design outcomes from the preceding chapter. This program comprises a single database that encompasses multiple tables, including:

Figure 9. Users Table and Item Table

Figure 10. Employee Table and Loan Table

Interface Implementation, the system implementation phase involves applying the design outcomes in a practical setting, enabling users to utilize the system in accordance with the defined objectives. Pressman (2021) defines system implementation as the process of converting conceptual designs into fully operational systems that are prepared to facilitate organizational actions. The following are the conclusive display outcomes of the system:

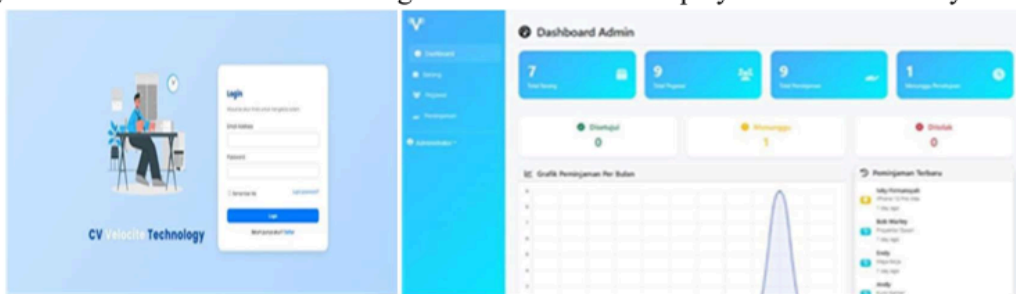


Figure 11. Login Page and Dashboard page

Figure 12. Item Page, Employee Page and Loan Page

System Testing, system testing represents the concluding phase of the software development process, designed to verify that all system components function cohesively and yield outputs that satisfy user requirements and specifications. This research on system information development identifies two prevalent methods of testing:

Blackbox Testing

Table 1. Login Testcase

No	Scenario	Test Case	Input	Output	Status
1	Login as admin with valid data	The user must enter the correct admin username and password.	Username: admin Password: admin123	Login successful, redirect to admin dashboard	Passed / Not Passed
2	Employee login with valid data	The user enters the correct employee username and password	Username: employee01 Password: employee123	Login successful, redirect to employee dashboard	Passed / Not Passed
3	Logout	User clicks the logout button	-	Logout successful, redirect to login page, session deleted	Passed / Not Passed

Table 2. Borrowing Testcase

No	Scenario	Test Case	Input	Output	Status
1	View loan data	The admin opens the loan data page.	-	A table displays a list of all loans (pending, approved, rejected, returned)	Passed
2	Filter loans by status	Admin filter status "waiting"	Status: waiting	Only loans with pending status are displayed.	Passed
3	Approve the loan	The admin clicks the approve button on the loan application.	-	The status changes to "Approved," the stock decreases, and a success notification appears.	Passed
4	Reject the loan	Admin clicks the reject button on the loan pending.	-	Status changes to "Rejected", stock of goods does not change, notification appears.	Passed
5.	Return process	Admin clicks the return button on approved loans	-	Status changes to 'Returned', stock of goods increases again, success notification appears	Pased

No	Scenario	Test Case	Input	Output	Status
6.	View loan details	Admin clicks - the details button		Display complete loan information (employee, item, date, status)	Pased
7.	Delete borrowing data	Admin deletes - completed loan data		Confirmation appears, data was successfully deleted.	Pased

Whitebox Testing

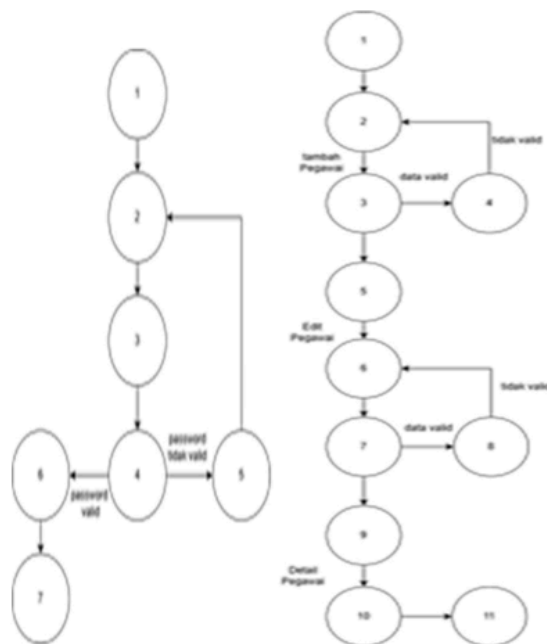


Figure 13. Login Testing Flowgraph and Borrowing Testing Flowgraph

CONCLUSION AND RECOMMENDATION

This study creates a web-based inventory borrowing management information system utilizing the First Come First Served (FCFS) algorithm for an IT service provider. From the outcomes of the implementation and system analysis, the subsequent conclusions can be derived: The solution effectively improves the precision of inventory documentation by integrating a MySQL database and implementing automatic validation, hence removing human errors associated with manual recording. The real-time monitoring tool provides immediate access to information regarding item availability, hence minimizing overlaps in borrowing and conflicts in asset utilization. The application of the FCFS algorithm has demonstrated efficacy in establishing an objective and transparent queuing system predicated on submission timestamps, thereby resolving conflicts among employees concerning borrowing priorities. The system ensures synchronization between administrative data and

actual item conditions via automatic status updates and thorough history monitoring, hence enhancing responsibility in inventory management. The designed system offers an efficient resolution to the challenges of traditional inventory management, improving operational efficacy and transparency in asset management.

REFERENCES

Cek EBMTJ 182

INFORME DE ORIGINALIDAD

8%

INDICE DE SIMILITUD

7%

FUENTES DE INTERNET

3%

PUBLICACIONES

5%

TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1

www.coursehero.com

Fuente de Internet

2%

2

dspace.enset-skikda.dz

Fuente de Internet

1%

3

ir.bdu.edu.et

Fuente de Internet

1%

4

www.numberanalytics.com

Fuente de Internet

1%

5

www.softguide.com

Fuente de Internet

1%

6

Submitted to University of Wales Institute, Cardiff

Trabajo del estudiante

1%

7

www.tutorialspoint.com

Fuente de Internet

<1%

8

brokenwings021.blogspot.com

Fuente de Internet

<1%

9

Aharon Yadin. "Computer Systems Architecture", Chapman and Hall/CRC, 2016

Publicación

<1%

10

"Model-view-controller(MVC)", Salem Press Encyclopedia of Science, 2017

Publicación

<1%

Excluir citas

Excluir bibliografía

Apagado

Activo

Excluir coincidencias

Apagado