# SwiftSell: A Holistic, Agile-Driven Point-of-Sale System for Enhancing Retail Efficiency and Customer Engagement

Addy Arif bin Mahathir * , Sivamuganathan Mohana Dass , Sai Rama Mahalingam , Dylan Chan Kit Lun , Ahmed Abdullah Faisal Ghaleb , Ahmed Abdullah Alsukhailah , Noor Ul Amin

Posted Date: 24 April 2025

doi: 10.20944/preprints202504.1899.v1

Keywords: cybersecurity; POS system; AI; customer interaction

 doi:10.20944/preprints202504.1899.v1



*Article*

# SwiftSell: A Holistic, Agile-Driven Point-of-Sale System for Enhancing Retail Efficiency and Customer Engagement

**Addy Arif bin Mahathir *, Sivamuganathan Mohana Dass, Sai Rama Mahalingam, Dylan Chan Kit Lun, Ahmed Abdullah Faisal Ghaleb, Ahmed Abdullah Alsukhailah and Noor Ul Amin**

Taylors University

* Correspondence: addyarif1910@gmail.com

**Abstract:** The SwiftSell project brings a cutting-edge Point-of-Sale (POS) system to modern retail and service environments as a tool to optimize operations and customer interaction, as well as the data-driven making of decisions within the business. Being different from all the conventional POS systems out there, SwiftSell has been built on an integrated software architecture, converging a stable backend with a very dynamic frontend and enriched analytics. It is a multi-platform user access system for the customer, cashier, and supervisor and includes real-time inventories, attendance logging, placing orders through mobile devices, and automated feedback loops to customers. The development of SwiftSell was informed by a hybrid project management methodology that combines Waterfall and Agile approaches. This enabled the structured planned activity to abide with the flexibility needed due to user feedback and market shifts. Besides, there were perfected testing strategies, such as unit and integration tests, system tests, and user acceptance tests, contributing to a strong, secure, and friendly-to-user solution throughout the software development life cycle. Risks involved cybersecurity, customer satisfaction, and market competition. SwiftSell's modular structure and mechanisms for continuous improvements render the system scalable, sustainable, and ready for the inevitable innovations in POS systems and their future presence in AI.

**Keywords:** cybersecurity; POS system; AI; customer interaction

## 1. Introduction

In the context of today's fast-paced life of service and retail businesses, businesses don't merely require antique transaction interfaces—they require industries-compatible, flexible, secure, and easy-to-operate systems capable of efficiently streamlining day-to-day work flows without affecting shopping. Considering the requirements, the innovation concept SwiftSell emerged to modernize current retail management with its groundbreaking POS integrating management centralization, employee tracking, customer relation management, and analytical reporting under a productive one-digit computing technology [1,2].

SwiftSell isn't a mere transactional solution—it's a comprehensive platform empowering business by virtue of its module-based architecture, cross-platform availability, and interface-friendly nature. The solution accommodates multiple interaction modes for lead user personas like customers (through mobile app), employees (through tablet), and supervisors (through web console) each catering to specific functionality like facial recognition-enabled attendance, inventory status in real-time, reporting based on data, and tailored customer interaction.

The development process from ideation to deployment was a hybrid project management approach. The project was initially structured under the Waterfall framework to create absolute developmental milestones, after which it adopted Agile principles to enable iterative feedback and dynamic scope development. This enabled a balance between comprehensive documentation and adaptive, real-time problem-solving [3].

 doi:10.20944/preprints202504.1899.v1



To ensure quality and security at all levels, SwiftSell's development was underpinned by a rigorous multi-level software testing regime, ranging from unit tests of individual modules through to user acceptance tests at the system-wide level. Particular attention was given to the employee checkout interface due to its direct effect on customer satisfaction and operational throughput.

Amidst emerging cyber threats and competitive marketplace dynamics, SwiftSell's architecture further prioritizes sweeping security protocols, proactive risk mitigation procedures, and a scalable infrastructure ready for future innovations—including AI-powered capabilities and strategic business partnerships. Overall, SwiftSell is a visionary solution in POS technology, engineered to adapt, acclimate, and deliver long-term value in changing business environments [4].

## 2. Literature Review

SwiftSell as a complete Point-of-Sale (POS) solution must be grounded in thorough literature regarding the changes ushered in by the current POS systems that boost efficiency, take on the complexities of managing commodity ranges, develop a better customer experience, and support effective decisions based on data. Today, POS systems have progressed from simple cash registers, which worked merely as points of sale, into information management platforms capable of accepting a number of other functions, such as inventory and employee reports, as well as customer extension [5,6] Therefore, with increasing flexibility demand while minimizing infrastructure costs, POS systems with cloud and mobile compatibility have become a norm in today's environments of modern retail [7].

Indeed, successful implementation of such systems as SwiftSell would definitely require a well-thought, intuitive system design to serve the varying needs of users-customer, employees, or managers. According to the author [8], set simple and visible system status as error prevention as the fundamental for a user-friendly interface. In support of this design philosophy, UML principle applications include activity and state diagrams that provide strong visual modeling frameworks for the representation of complex system interactions [9,10].

SwiftSell's project methodology features software engineering best practices. The Waterfall model was first utilized for systematic planning, but Agile brought about the most effective user feedback and requirement change accommodation. Literature backs up this type of hybrid Agile-Waterfall approach as this combines the comprehensive initial planning needed and the flexibility of iteration which is suitable for large-scale flexible and dynamic development environment [11,12].

An All-Important Part is a Test Framework in Total. The stress laid by SwiftSell on different test types namely, unit test, integration tests, system tests, and user acceptance tests is match with international standards such as the ones set by the ISTQB. As per [13,14], rigorous application of such test procedures makes it easier to identify and resolve defects from a software view early in development and improve their reliability in the long term. According to industry practitioners like [15], there are methodologies and toolkits that have been used resoundingly by many in test automation and validation processes, in turn, giving further robustness to SwiftSell.

As environments become more connected and online, Point-of-sale (POS) systems will require more emphasis on cybersecurity and data protection. Major threats that have been highlighted by [16,17], include phishing, malware, and denial-of-service attacks, which can compromise system integrity as well as customer data. This implies that there will be a need for multilayer security such as firewalls, encryption, authentication, and Data Loss Prevention (DLP) policies in order to mitigate against such risks.

Lastly, customer satisfaction and competitive strategy have become very important for the sustenance of business success in the long term. It can be argued, as Lemon and Verhoef (2016) do, that a personalized and seamless customer journey significantly increases retention and loyalty. The features on offer by SwiftSell such as loyalty programs, real-time feedback devices, and personalized communication are good examples of such expectations. Also, Five Forces model, such innovative practices fused with strategic positioning ensure that SwiftSell remain competitive to keep it relevant and profitable in a fast-changing digital marketplace.

 doi:10.20944/preprints202504.1899.v1



**Sequence Diagram**

Sequence diagrams illustrate how the different elements in the SwiftSell system communicate at different points in time. Sequence diagrams assist in demystifying hard-to-grasp user-system interactions by breaking them down into individual use cases.

This Figure 1 offers a high-level overview of the system's operations. With all these features—customer ordering, employee attendance, ordering inventory, collecting feedback, producing reports, updating products, and employee authorization—planning required extensive modeling. Therefore, the team created several detailed sequence diagrams to accurately represent each of these individual flows.

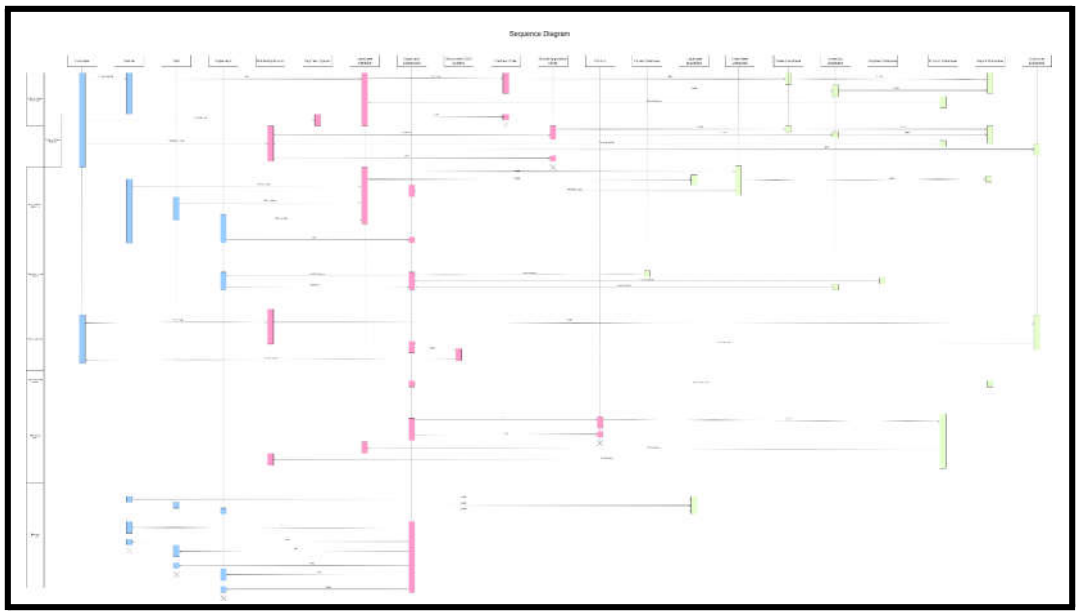


**Figure 1.** Overall Sequence Diagram.

The Customer Ordering Sequence Diagram (Figure 2) depicts two key methods of placing orders in SwiftSell POS. Firstly, there is cashier-assisted, whereby the cashier directly interfaces with the customer via the employee interface, selecting goods stored in the product database. Once confirmed, the order is updated simultaneously in the sales, inventory, and report databases. Customers can request order alterations prior to final confirmation. The second is through the mobile application, whereby customers register or continue as visitors, explore the product menu, and manage the cart. At the point of confirmed order placement, the same databases are similarly updated, plus the customer database, while securely making payments inside the app environment.

In the Employee Attendance Sequence Diagram (Figure 3), employees like supervisors and cashiers punch in via facial recognition on the employee interface [19]. This activity refreshes the employee and timesheet databases, and data is also saved to the report database. Supervisors are able to readily see attendance records in real-time via the supervisor dashboard in order to enable effective employee tracking.

The Ordering Inventory Sequence Diagram (Figure 4) explains how supervisors track inventory levels through their dashboard. They can access the supplier and inventory databases completely so that they can analyze customer order trends and maintain the appropriate levels of inventory, thereby enabling demand-driven restocking and supply chain management.

According to the Feedback Sequence Diagram (Figure 5), customers are asked to give feedback after they have confirmed their order using the mobile application. After the feedback is given, it is automatically stored in the customer database and made available in the supervisor dashboard. The

 doi:10.20944/preprints202504.1899.v1



system is now able to send automatic acknowledgement or appreciation messages, hence further increasing customer interaction and satisfaction.

The Analysis Reports Sequence Diagram (Figure 6) is dedicated to performance tracking for managers. Sales, attendance, and inventory reports are collected and offered from the dashboard, fetched from back-end databases that update in real time. The centralized access to analytical data supports efficient managerial decision-making.

In the Update Products Sequence Diagram (Figure 7), supervisors can update product data from the dashboard. They can introduce new products or remove old ones, which updates the product database automatically. The updates are reflected in the mobile app as well as the employee interface so that everybody is working with updated product data.

The Employee Authorization Sequence Diagram (Figure 8) addresses internal user management. Supervisors can add or remove employees like other supervisors, cashiers, and employees. The changes are securely logged into the employee database, with proper access control and ensuring that only authorized staff interact with specific system components [20–22].

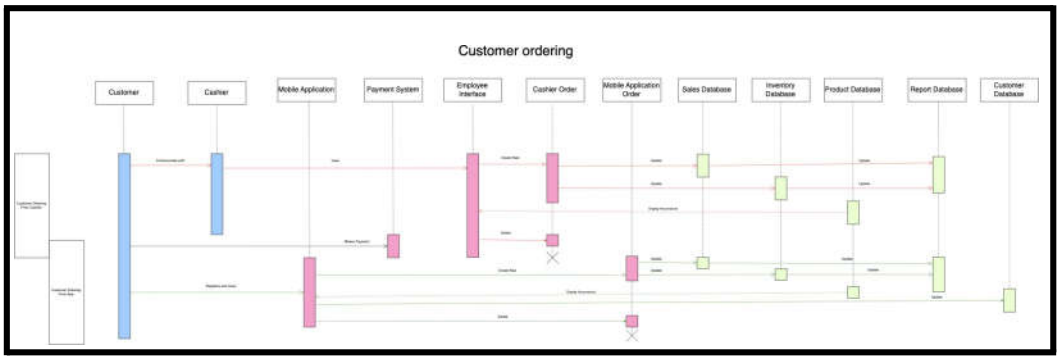



**Figure 2.** (Customer Ordering Sequence Diagram).

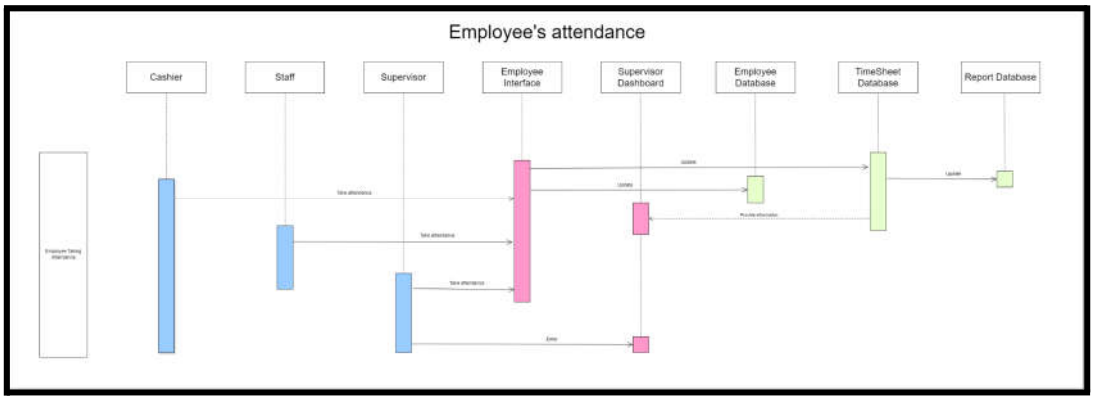



**Figure 3.** (Employee's Attendance Sequence Diagram).

 doi:10.20944/preprints202504.1899.v1



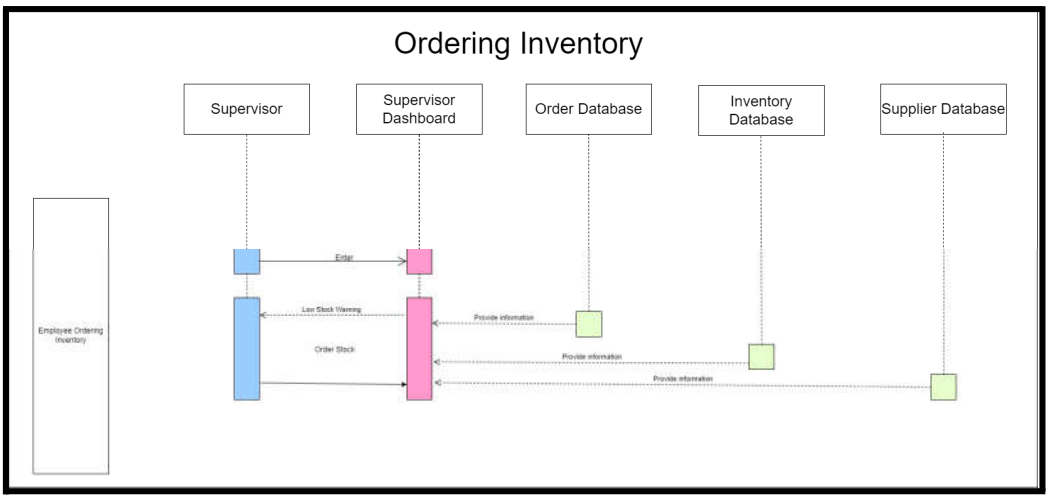



**Figure 4.** (Ordering Inventory Sequence Diagram).

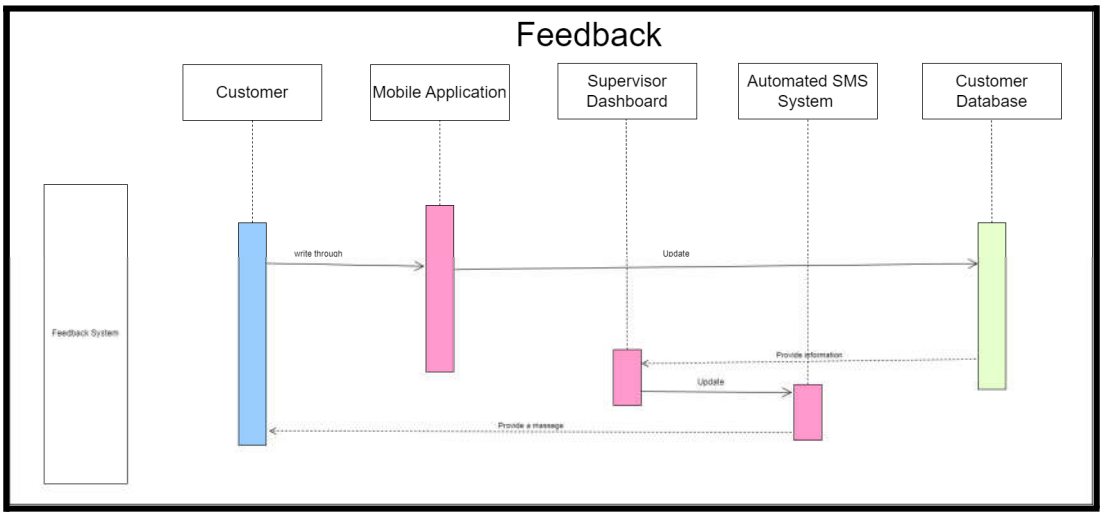



**Figure 5.** (Feedback Sequence Diagram).

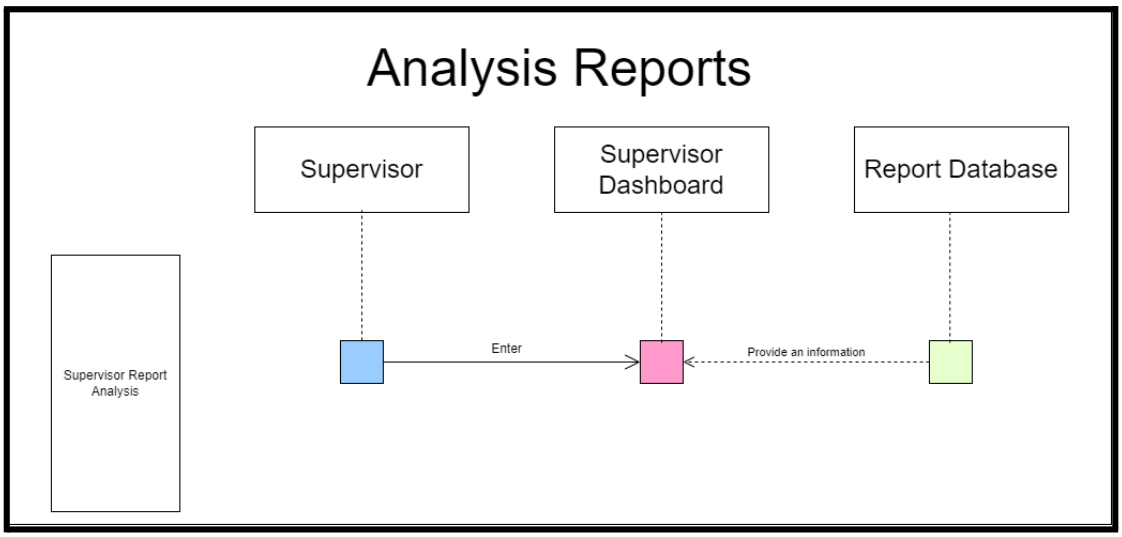



**Figure 6.** (Analysis Reports Sequence Diagram).

 doi:10.20944/preprints202504.1899.v1



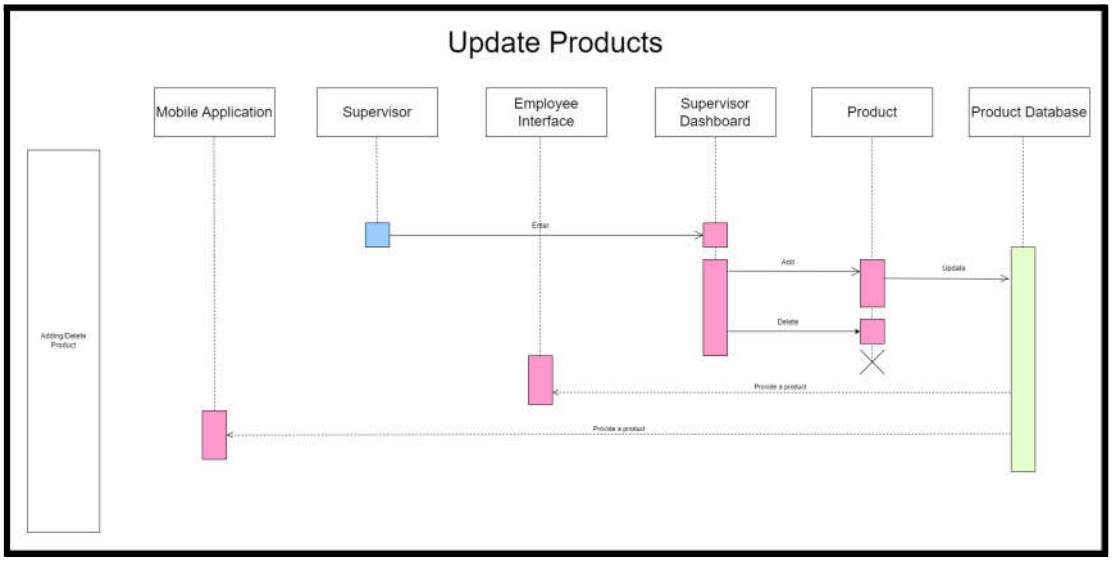



**Figure 7.** (Update Products Sequence Diagram).

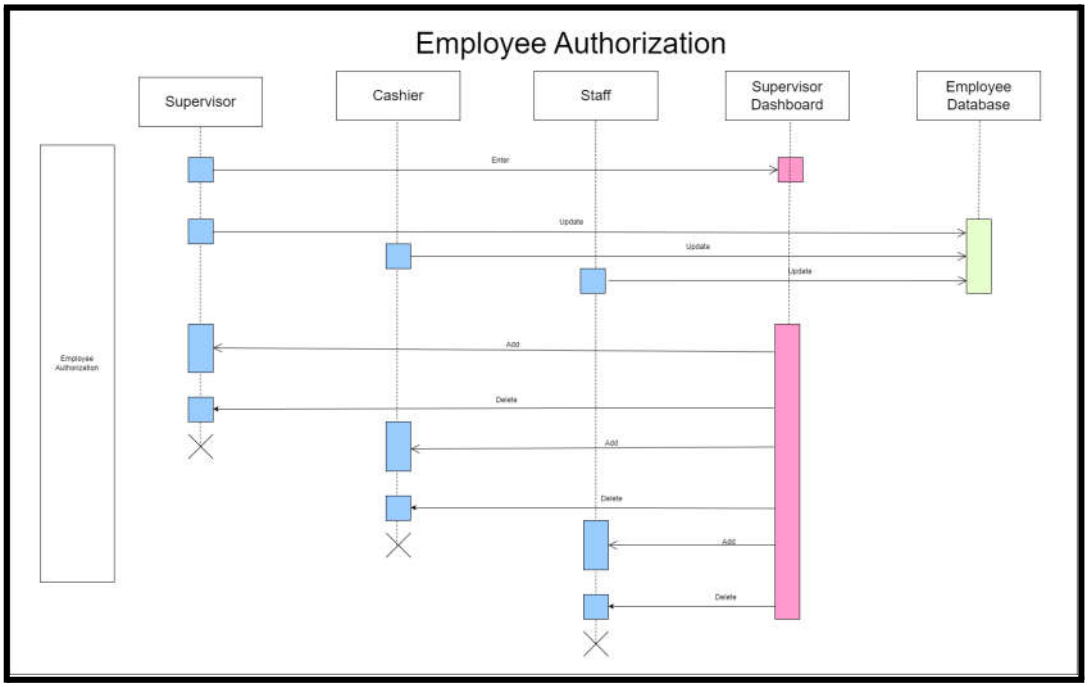



**Figure 8.** (Employee Authorization Sequence Diagram).

**Activity Diagram**

In designing a program, it is important to craft the activity diagram in order to understand the relationship between the users and the systems. For this system, 3 diagrams were created, each focusing on different types of users: customer, cashier and supervisor.

 doi:10.20944/preprints202504.1899.v1



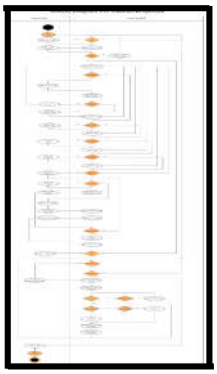


**Figure 9.** (Activity Diagram for Cashier/Employee) - Appendix 1.

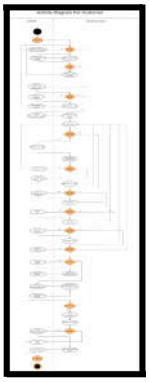


**Figure 10.** (Activity Diagram for Customer) - Appendix 2.

**Figure 11.** (Activity Diagram for Supervisor) - Appendix 3.

 doi:10.20944/preprints202504.1899.v1



## State Chart Diagram

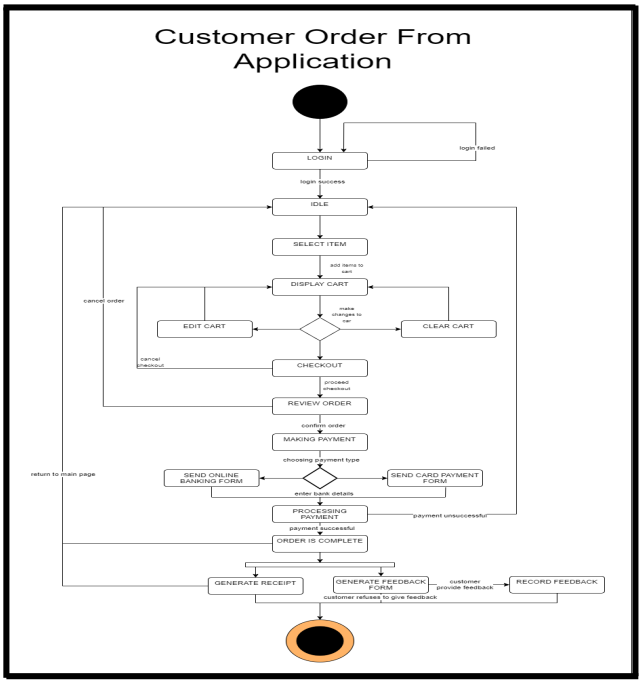



**Figure 12.** (State Chart Diagram for Customer Application).

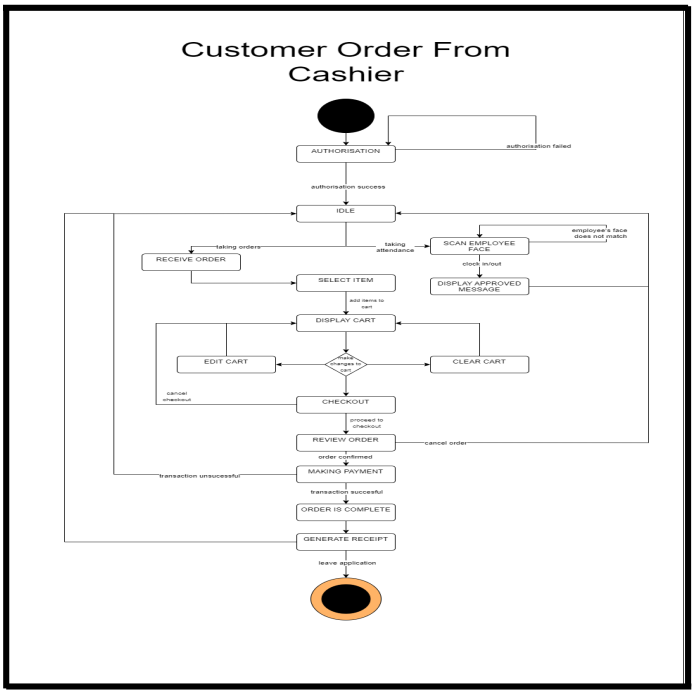



**Figure 13.** (State Chart Diagram for Cashier Application).

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 April 2025 doi:10.20944/preprints202504.1899.v1



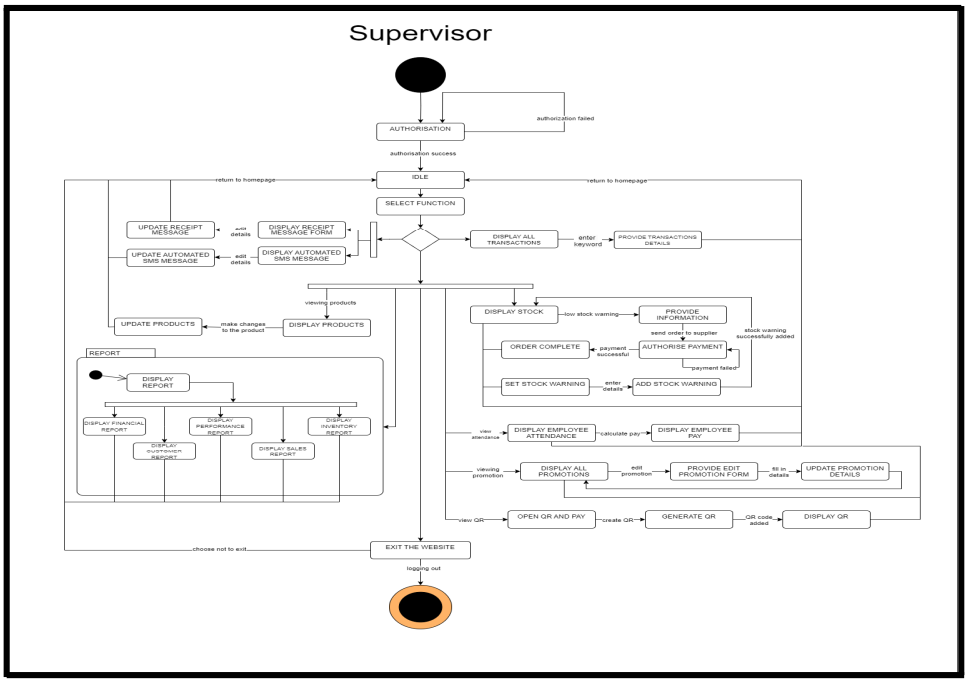



**Figure 14.** (State Chart Diagram for Supervisor).

## 2. Frontend System Design Rationale

### *2.1. User Interface Design*

The frontend of the SwiftSell POS system was created with a strong focus on usability so that all types of users—customers, employees, and supervisors—experience a natural, easy interaction. Among the many user interactions created, five were determined to be the most critical. For the customers and employees, the most critical interactions are to place items in the cart and checkout. For supervisors, the key functions are product management and setting up automated SMS communication for improved business productivity and customer engagement.

The Customer Mobile Application Interface (Figure 15) presents an easy-to-use experience from malleable access points. Customers login, sign up, or proceed as a guest based on individual choice. Within the app, users can click on every item to view detail and add it to the cart. Once they have done their selection, they proceed to checkout via their selected mode of payment. Users can also store a new credit card on their account for ease of use. After the completion of their orders, they are also encouraged to provide feedback in the form of a review of the product or service. The users can also return to the home menu at any time through the "Back to Home" feature for easy navigation throughout the app.

The Employee Tablet Interface (Figure 16) facilitates main functions to aid the operation of employees. Staff members check in and out using facial recognition, which also drives the device's camera. They can log out of the session for security reasons. When dealing with orders, the employee can leave comments on items before placing them in the cart, apply promotions by entering a code, and cancel orders to clear the cart when needed. Once the order is placed, they can proceed to checkout, which triggers the system to pass on the order to the associated backend services [23].

The Supervisor Website Interface, as illustrated in Figure 17, provides an all-encompassing management dashboard for supervisors with several key ingredients. In the login page, supervisors enter their credentials to access the system or register for an account via a dedicated registration form. On logging in, supervisors are greeted with the dashboard page, which summarizes vital business metrics such as total expense, deposit, expected income, and graphical representations tracking monthly expenditure patterns. Supervisors can interact with these graphs to analyze specific periods in detail.

 doi:10.20944/preprints202504.1899.v1



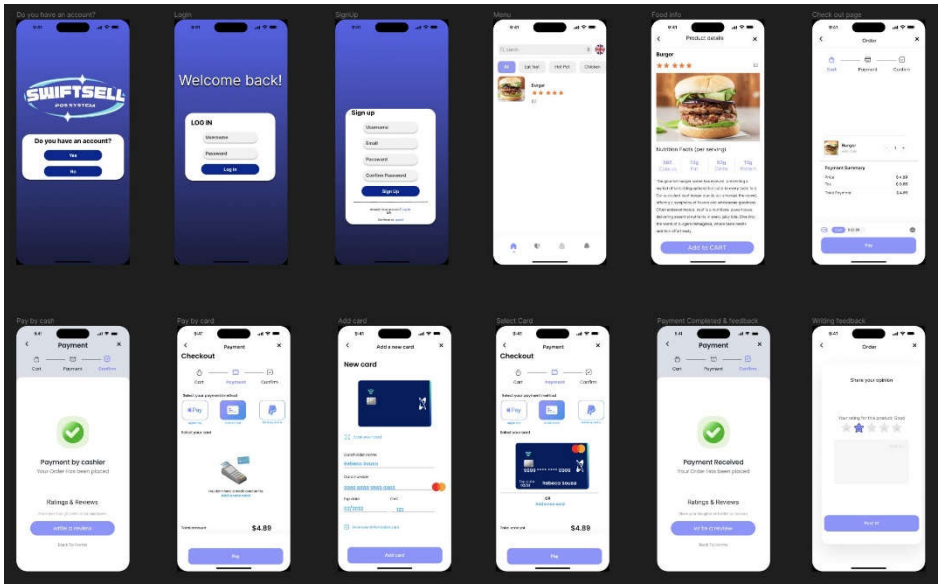



**Figure 15.** (Customer Mobile Application Interface).

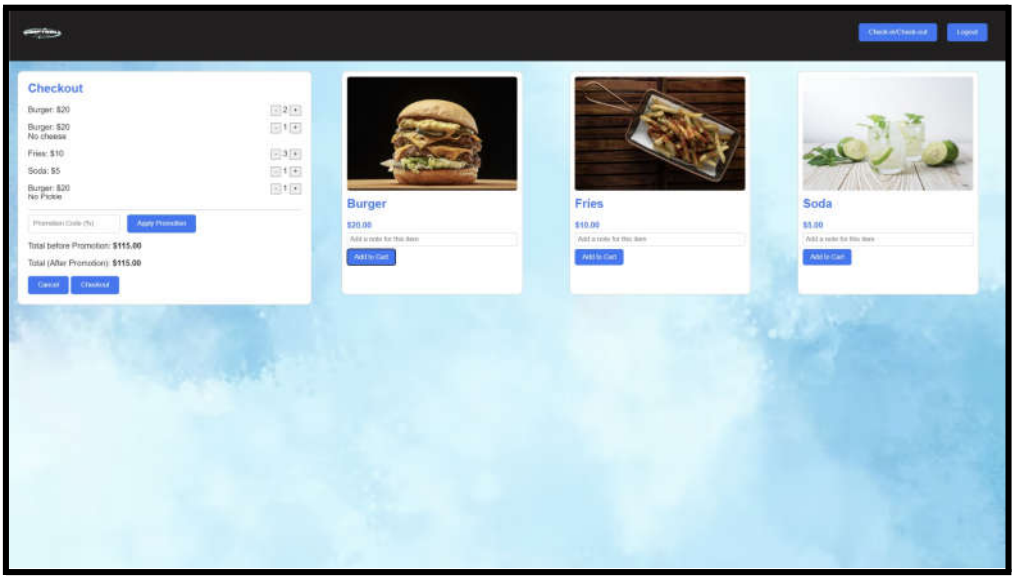



**Figure 16.** (Employee Tablet Interface).

 doi:10.20944/preprints202504.1899.v1



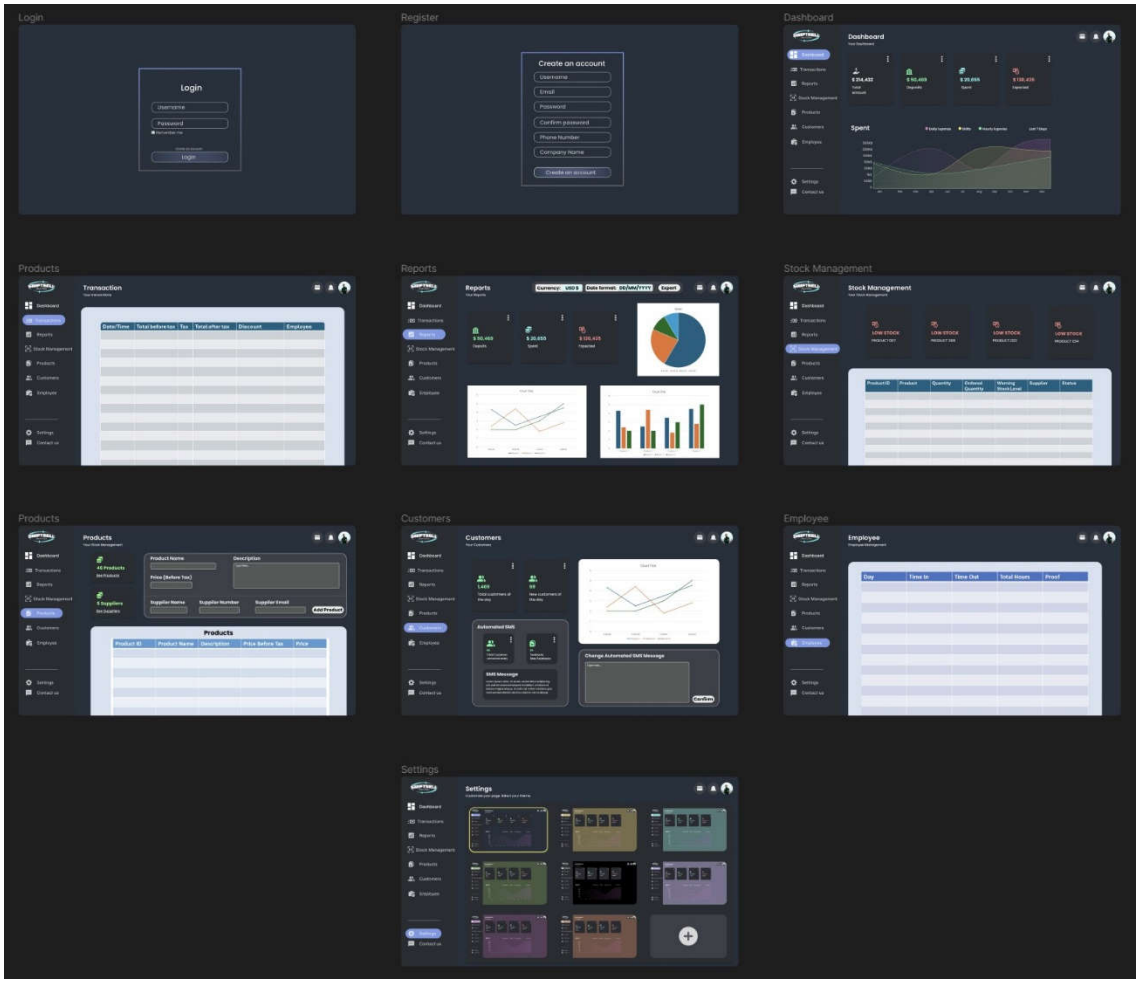



**Figure 17.** (Supervisor Website Interface).

On the transaction page, supervisors can view tables of transaction history with details such as date, time, items purchased, and total amounts, and customer information. The report page generates financial reports for selected date ranges and preferred formats in currency. Further filtering options are available for filtering and exporting these reports for analysis or record purposes.

From the stock management page, supervisors can view the current stock level and receive alerts for low stock via visual indicators, which guide them to maintain efficient inventory control. On the products page, supervisors view a list of all items for sale and take necessary actions like adding new products by providing relevant information, editing product details, and deleting obsolete ones with safety pop-ups to prevent accidental deletion.

The customer page provides access to a comprehensive list of customers, contact information, purchasing history, and user reviews. Supervisors may also view extensive customer reviews and manage communication through sending automated SMS messages. These messages can be tailored to numerous marketing campaigns or service notifications by using the edit SMS message tool.

Finally, on the employees page, the supervisors can check attendance details of each employee such as clock in and clock out time and number of hours worked. They also can modify employees' details like job titles or contacts to update the system correctly and in a timely manner. Generally, frontend development of SwiftSell is engineered to make tasks easy, be easier to use, and support smooth interaction of all users taking part in the system.

## 3. Methodology

This hybridized project management approach was quite helpful for the development of the SwiftSell POS System as it combined both Waterfall and Agile methodologies in order to incorporate

 doi:10.20944/preprints202504.1899.v1



well-structured planning with the necessary adaptive responsiveness into it. As such, the Waterfall methodology was actively adopted and applied by the team. The traditional linear approach adopted by Waterfall made it easy for the separation of phases and documentation of development processes. This was great, especially at the beginning stages of the project where there was a need to set solid objectives, define detailed technical specifications, and prepare complete design documents. The Waterfall model owned a very clear transition through the stages of requirement gathering, design, development, and initial testing [24,25].

As the development was progressing, it was becoming apparent that the Waterfall model was an inflexible approach, with the inability of the project team to accommodate user's and other stakeholders' continuing feedback. The use cases arising out of reality and emerging customer requisites pointed toward a need to bend, loop, and respond quickly. Hence, the approach the team shifted to was integrating Agile methodology into their workflows. The iterative development cycles through sprint offered increased eigenvalue flexibility while it focused more on user feedback, continuous testing, and incremental delivery, all of which are very much essential in preparing market-responsive and user-centered POS solutions [26].

The hybridization therefore borrowed the best from both forms. Waterfall gave clarity, direction, and a base structure for planning. In contrast, Agile provided the program with flexibility, iteration along the life of the project, and interaction with interested parties. Thus, it could create a very dynamic development situation where all planning and execution of work, feedback, and revision were done in a continuous loop and in a collaborative-yet-independent manner. This dual-model strategy enabled the team to remain grounded in initial project objectives while remaining nimble enough to incorporate new insights without major disruption to the overall timeline or system architecture [27,28].

### *3.1. Project Monitoring*

Gantt charts were the chief instruments for project monitoring for the team during development because they could visualize the timeline of the project by clearly specifying all the task start and finish dates, duration and dependencies of every phase in development. Thus, this visual framework would ensure the improvement of all planning accuracy as well as coordination of tasks much better communication of team members or stakeholders.

Additionally, in terms of Waterfall methodology, Gantt charts were very useful for documenting the linear stages in a project, such as requirement specification, system design, coding, testing and deployment. However, with the project's preference for Agile adoption, Gantt chart formats were modified to illustrate the cycles of sprints, milestones, and goals for iterative development. This flexible usage of Gantt chart facilitated the overall control and management of the project without precluding the requisite dynamism needed to adapt to changes occurring in the project.

It is actually a Gantt chart that decides the bridge between a form of planning of Waterfall and the Gantt chart, which is more flexibly used to portray the way into the world's most modern, iterative methodologies like Agile. It can keep the time in deriving deadlines while judging risk and giving the opportunity to exercise informed options about resource allocation or schedule revisions-in other words, keeping the project on track while enhancing aspect variations with iterative feedback and continuous improvement.

## 4. Software Testing

Software testing actually forms the basis for a strong quality assurance system in SwiftSell POS development. A structured multistage testing approach formed the foundation for testing functionality, reliability and usability of one component and the system at large. The structure consists of unit tests, integration tests, system tests, and end-user acceptance tests, along with a collaborative approach to resolving defects found.

 doi:10.20944/preprints202504.1899.v1



### *4.1. Unit Testing*

Unit testing had performed in the initial stages of development for the purpose of testing individual components in isolation. Each function was rigorously tested by developers to find bugs before placing the function out within the system. As a first layer of defense against cumulative errors later in the development process, this allows the basic building blocks of the system to operate reliably. Unit testing is the process of writing specific code segments that will test on a single function. By doing this, unnecessary dependencies are revealed that exist between modules, with the opportunity then being presented to isolate and eliminate the dependencies. To automate these tests, the development team employed popular frameworks such as Unit Test, along with other open-source tools like JUnit, NUnit, JMockit, EMMA, and Punit.

### *4.2. Integration Testing*

Post unit testing and integration testing was performed to evaluate the working relationship between software components. Herein lies an important activity wherein data and functionality exchange seamlessly between the connected modules. Integration testing was done according to a comprehensive test plan that catered to incremental and big-bang testing approaches. Testers developed scenarios, test cases, and scripts to emulate actual integration business workflows. Errors discovered during testing were logged, fixed, and re-tested iteratively until the modules behaved as expected. Testing was administered through open-source tools, including Jasmine, Protractor, and Selenium, to automate and simplify the testing process to assist developers in assuring compatibility across the entire system.

### *4.3. System Testing*

System testing, which was led by the Quality Assurance (QA) team, was the final technical validation phase prior to deployment. This integrated testing phase verified that all the components that were integrated functioned together in harmony and met the system's laid-down requirements. The system test process began with the setup of a test environment to replicate production. Test cases were devised to test specific features and functionalities, and test data was then developed to simulate a variety of user scenarios. Defects, if any, were discovered, documented, and fixed while tests were being executed. Regression tests were conducted after fixes were made to ascertain that the new code added did not disrupt existing features. End retesting ensured system stability and readiness for deployment. This stage was supported by a range of open-source and commercial tools like Squish and Spira Test for end-to-end automation and management, and Robotium and SoapUI for functional and API testing.

### *4.4. User Acceptance Testing (UAT)*

The final validation phase involved User Acceptance Testing (UAT) performed by end-users and client representatives. Such known stakeholders, being aware of business needs and anticipated business flows, played a crucial role in simulating actual usage scenarios. They played a major role in the evaluation of usability and business goal consistency of the software. Through systematic testing and candid feedback, they verified if the system met their requirements or not. Validation by this team ensured that the product was operational and deployable in a live environment (Testsigma, 2023).

### *4.5. Bug Fixing Methodology*

In the testing and development phases, any issues or bugs that arose were addressed by an open bug-fixing process. As soon as a bug was found, the team embarked on fervent internal debate to analyze its root cause. Brainstorming was employed to enable open sharing of various solutions and ideas. After the best solution method had been decided upon, the team implemented the solution and then performed further testing to ensure that the issue was resolved. This open and transparent

 doi:10.20944/preprints202504.1899.v1



process for resolving a problem not only accelerated resolution but also enhanced team cohesion and code quality.

## 5. Test Plan

Given the SwiftSell POS system project's broad breadth and complexity, we must focus our time and resources on crucial components to ensure complete testing and quality assurance. As a result, in this report, we will focus on the testing plan regarding the Employee Checkout Interface. This interface is a critical component of the POS system, influencing the efficiency and effectiveness of the checkout process. By focusing on this component, we can deliver a detailed and specialised test strategy that covers the intricacies and unique requirements of the checkout process. This method assures that we can painstakingly assess and improve the functionality, dependability, and usability of the Employee Checkout Interface, adding considerably to the overall success and effectiveness of the Swiftsell POS system.

| | |
|---|---|
| Module Name: | Employee Ordering System |
| Test Case ID: | SwiftSell_0001 |
| Author Name: | DYLAN CHAN KIT LUN |
| Tester Name: | DYLAN CHAN KIT LUN |
| Location: | Taylors University |
| Test Case Description: | To check the functionality of the ordering system in the employee's point of view. |
| Prerequisites: | 1. Stable internet connection.<br>2. SwiftSell App |
| Environmental Information: | 1. OS: IOS/Android<br>2. System: Tablet |
| Test Scenario | Checking that after selecting an item to add to cart, the item is added and when selecting delete, the item is deleted. |

| Priority | Test Case ID | Test Steps | Test Input | Expected Results | Actual Results | Status | Comments | Date of Execution |
|---|---|---|---|---|---|---|---|---|
| Very Important | 1. | 1. Add Note to Item<br>2. Add Item to Cart<br>3. Check Cart | Enter note in text box<br>Click on "Add to Cart" | The item should be in the cart with its price.<br>The total should be calculated at the footer of the cart. | The item is in the cart with its price.<br>The total is correctly calculated. | Pass | Feature worked without any lag. | 11/3/2024 |
| Very Important | 2. | 1. Add Item to Cart<br>2. Delete Item<br>3. Check Cart | Click on "Add to Cart"<br>Click on "-" | The item should not be in the cart and the total should be zero. | No items in cart and total is zero. | Pass | Cart was empty as required.<br>Note: future enhancements to keep the cart height the same throughout. | 11/3/2024 |
| Very Important | 3. | 1. Add Item to Cart<br>2. Add more of the Item<br>3. Check Cart | Click on "Add to Cart"<br>Click on "+" | The quantity of the item (2) should be in the cart with its price.<br>The total should be calculated at the footer of the cart. | The quantity of the item is displayed as 2 with the price.<br>The total is correctly calculated. | Pass | Addition of more items worked.<br>Tester also tested the note option and it added a note to the order. | 11/3/2024 |

**Figure 19.** (Test Plan for Employee Checkout Interface).

### *5.2. Risk Management Plan*

Safety of our website and its data are one of the greatest threats in our system. Our website can be invaded by hackers, malware, phishing, and denial-of-service attacks and reveal private details of the stakeholders. Strong security measures like encryption, firewalls, authentication, backup, and recovery must be adopted in order to avoid and manage security threats. Additionally, stakeholders must inform their clients and employees about internet safety and abide by all applicable laws and regulations, such as the Payment Card Data Security Standard.

### *5.3. Client Satisfaction Risk*

Delivering quality goods and services, shipping and delivery on time, making website operation smooth and simple, and hearing client feedback are necessary to improve and sustain customer satisfaction.

### *5.4. Competition Risk*

SwiftSell operates in a very competitive sector. Its profitability, market share, and expansion are all at risk from prospective competitors, substitutes, and entrants. More affordable prices, improved features, increased selection, or improved advertising can all lead to competition dangers. Competition dangers can lead to reduced return on investment which can lead to a loss of investors and potential buyers.

 doi:10.20944/preprints202504.1899.v1



*5.5. Risk Mitigation*

The team has identified the target market and analyzed other players' strategies and performances. By continuously analyzing the market and performing research, taking advantage of the unique selling proposition, and monitoring competition threats, SwiftSell can manage and produce the system with less risk. In addition, the 20% contingency costs that the team has calculated can cover any problems that may occur in the production and rollout of the product.

*5.6. Maintenance and Evolution Plan*

Maintenance and evolution of software have to be assured to ensure that softwares is kept updated and functional. They include repairing faults, adding new features, and conforming to evolving specifications.

*5.7. Security*

In the first instance, necessary measures have to be taken to ensure proper maintenance and development of the online platform. Security of the website is the top priority because the website software must be updated and patched on a regular basis in an attempt to fix the security holes quickly. Secure coding practices and regular security scans will build a solid system against any attacks. Moreover, the utilization of web application firewalls provides the capability to monitor and filter incoming traffic and enhances the defence mechanism of the system.

*5.8. Data Security*

Data security is yet another important aspect that needs to be strictly regulated in the context of protecting sensitive information. Data encryption while data transfer and data storage needs to be conducted with the help of high-level cryptographic methods to maintain information security at all times. Authentication procedures and access control systems are utilized to add more layers of security. This is necessary so that unauthorised access to the system can be prevented. Moreover, backing up data regularly in off-site storage is vital. Off-site storage is also necessary to enable the recovery of data during a data breach, such that business continuity and reliability are attained.

*5.9. Customer Satisfaction*

Setting up the customer satisfaction and product management systems are required for maintaining the optimum quality of the service and satisfaction of the customers. This begins from providing the highest quality of product that adheres to stiff quality checks. A robust website performance is paramount while securing and providing safe and trouble-free payment processing solutions. By constantly monitoring customers' grievances and apologizing for their issues presented in person or on social media, issues will be resolved, and utmost customer satisfaction will be obtained. Besides that, generating brand loyalty by using personalized experiences, customer loyalty programs, and top-notch customer service, that contribute positively towards customer retention as well.

*5.10. Additional Innovation*

For being competitive and profitable in the long run, a company must pursue a preemptive and distinct approach that separates them from the others and will keep their customers for themselves. It is a matter of making out-of-the-way features, advantages or services which will only appear in such goods and set them on the forefront of the marketplace. Other options such as integration with Artificial Intelligence would create a more uniform system that would be capable of functioning independently and thus reduce the likelihood of human error.

 doi:10.20944/preprints202504.1899.v1



*5.11. Forming Alliances*

By forming strategic alliances or partnerships with people of similar minds, it is possible to travel far and broad and become more competitive by providing access to new people, resources and markets. Eventually, the hard work pays off in the form of good word-of-mouth and the brand is well established as a trustworthy industry which, as a consequence, ensures the successful existence of the brand in an uncertain business world.

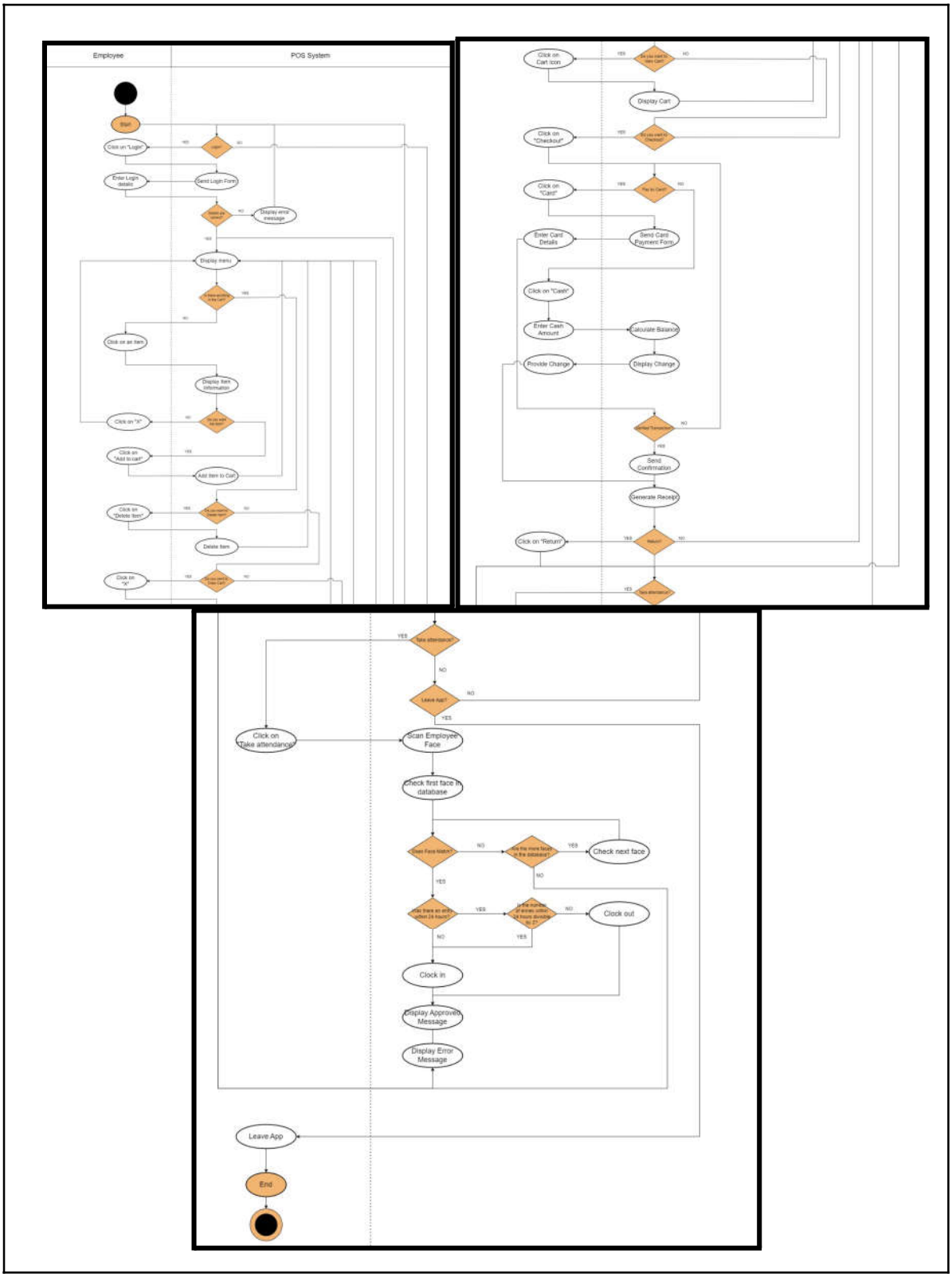
## Conclusion

The development of SwiftSell is a giant step towards the evolution of point-of-sale systems because it strikes a perfect balance between functionality, usability, and innovation. Starting from conceptualization to actualization, every aspect of the system—from backend logic and frontend beauty to extensive testing and risk management—has been meticulously designed to develop a solution that exceeds traditional POS systems. SwiftSell is not just a transaction processing software but an end-to-end business intelligence solution that empowers stakeholders with real-time data analytics, efficient inventory management, user-friendly interfaces, and more engaging customer interaction. With the adoption of a hybrid project management style combining the structural predictability of Waterfall and the adaptability of Agile, the team has been able to remain on schedule while being open to evolving requirements. The implementation of rigorous testing methodologies and an agile risk management strategy ensures the system's strength, security, and scalability. Moreover, SwiftSell's commitment to continuous enhancement, customer delight, and forward-thinking innovation—such as the potential for AI integration—establishes it as a revolutionary force in the retail and services industries. Last but not least, SwiftSell is not software but a strategic tool for businesses seeking operational excellence and long-term survival in a digital market that's becoming more dynamic by the day.

 doi:10.20944/preprints202504.1899.v1

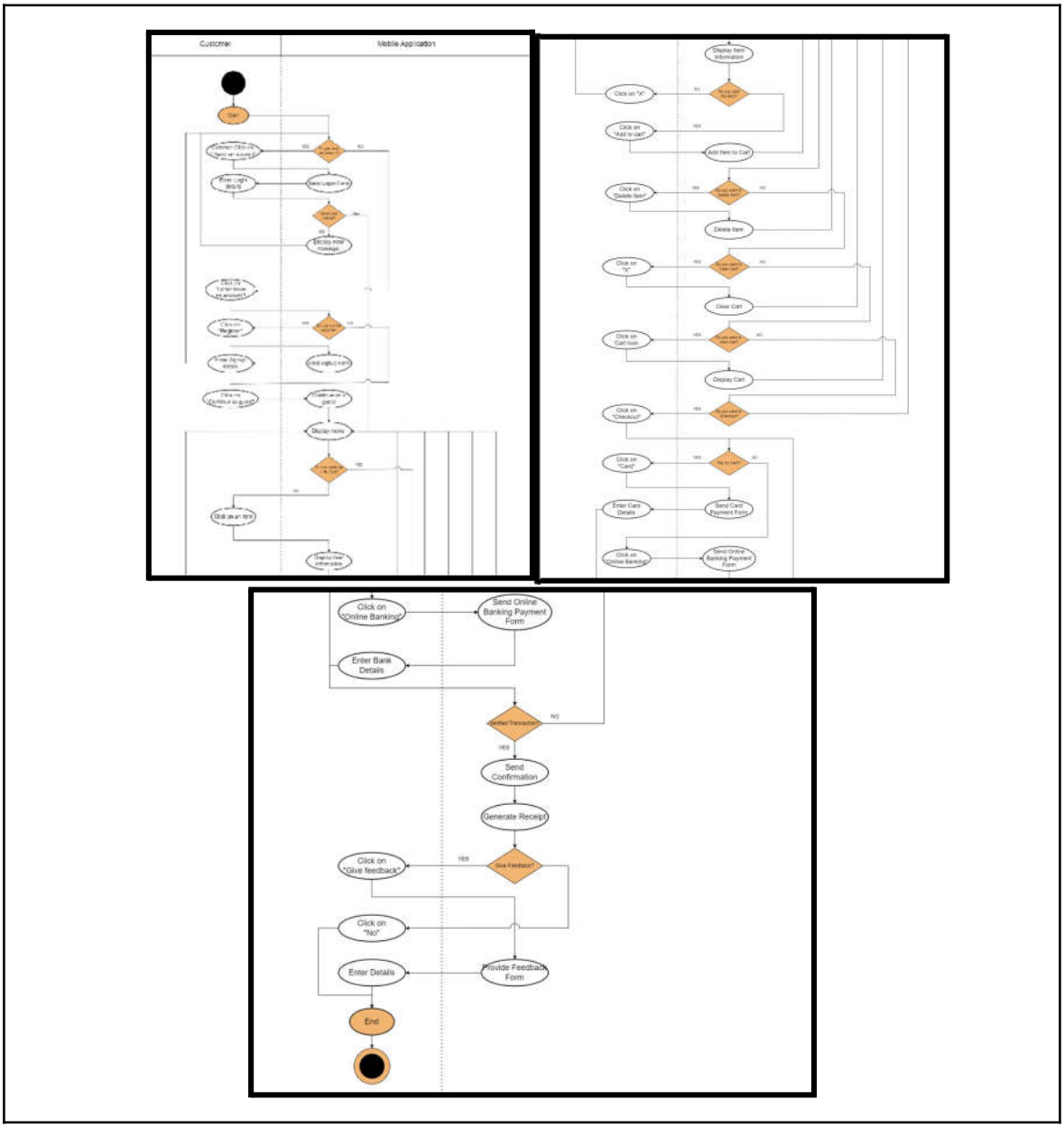


# Appendix A



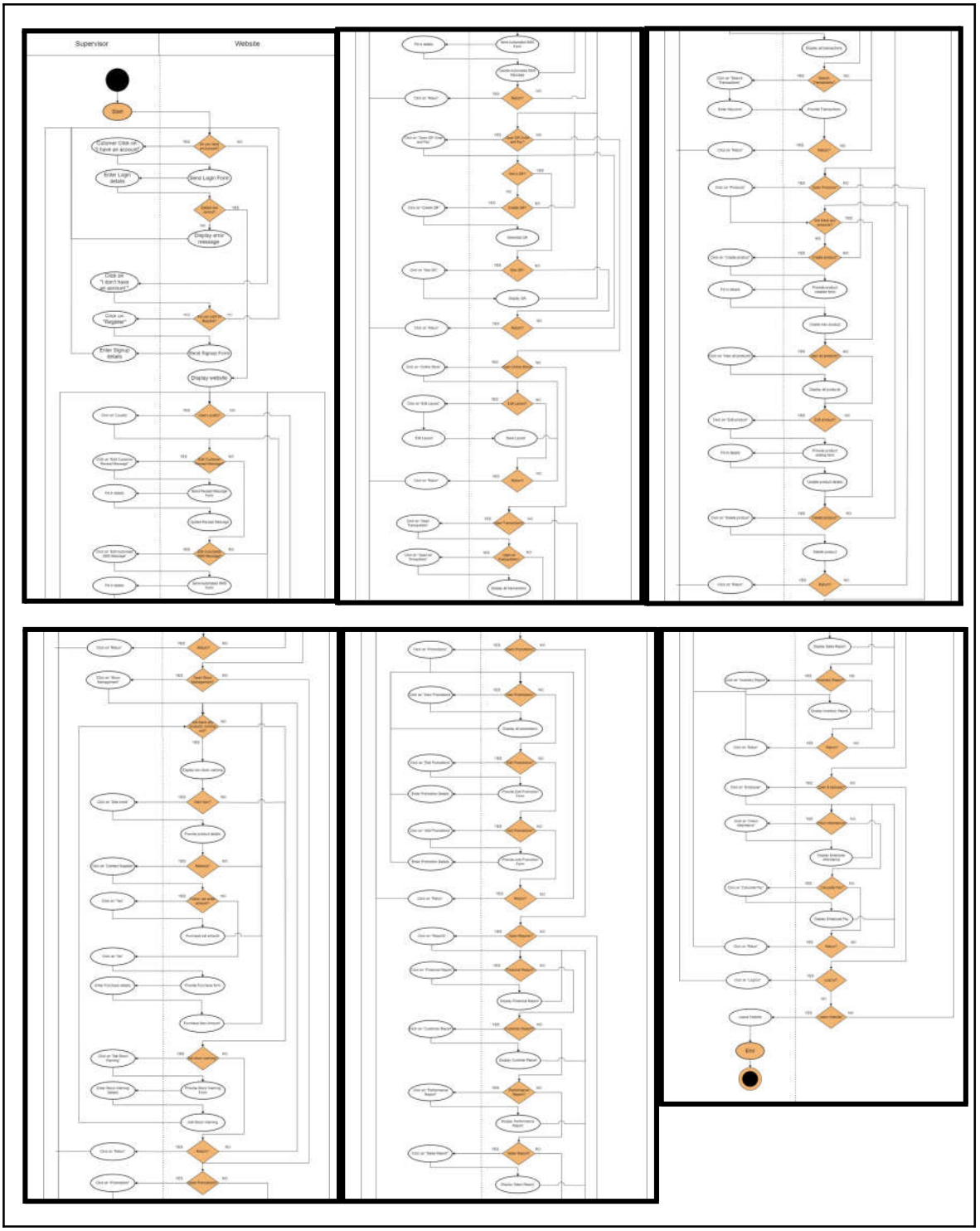
 doi:10.20944/preprints202504.1899.v1





 doi:10.20944/preprints202504.1899.v1






## References


1. Patel, M., Hirpara, P., Bhimani, P., Upadhyay, S. K., & Dubey, A. (2025). *Enhancing Restaurant Operations through a Modern POS System*. International Journal of Innovative Research in Technology, 11(10)
2. MobiDev. (2025). *Top Point-of-Sale (POS) Technology Trends & Innovations in 2025*
3. Esra, A. A., Abdelsalam, M. M., Tawfig, A., & Salwa, E. (2020). Generating UML Class Diagram using NLP Techniques and Heuristic Rules.
4. Güncan, D., & Onay Durdu, P. (2021). A user-centered behavioral software development model. *Journal of Software: Evolution and Process*, *33*(2), e2274.

 doi:10.20944/preprints202504.1899.v1




5. Natarajan, T., & Pichai, S. (2024). Transition from Waterfall to Agile Methodology-An Action Research Study. *IEEE Access*.
6. Yahya, N., & Maidin, S. S. (2022, September). The Waterfall Model with Agile Scrum as the Hybrid Agile Model for the Software Engineering Team. In *2022 10th International Conference on Cyber and IT Service Management (CITSM)* (pp. 1-5). IEEE.
7. Najihi, S., Elhadi, S., Ait Abdelouahid, R., & Marzak, A. (2022). Software Testing from an Agile and Traditional view. *Procedia Computer Science*, *203*, 775-782
8. Alferidah, D. K., & Jhanjhi, N. Z. (2020, October). *Cybersecurity impact over big data and IoT growth*. In 2020 International Conference on Computational Intelligence (ICCI) (pp. 103–108). IEEE. https://doi.org/10.1109/ICCI51257.2020.9247742 *(Add DOI if available)*
9. Jena, K. K., Bhoi, S. K., Malik, T. K., Sahoo, K. S., Jhanjhi, N. Z., Bhatia, S., & Amsaad, F. (2022). E-learning course recommender system using collaborative filtering models. *Electronics, 12*(1), 157. https://doi.org/10.3390/electronics12010157
10. Aherwadi, N., Mittal, U., Singla, J., Jhanjhi, N. Z., Yassine, A., & Hossain, M. S. (2022). Prediction of fruit maturity, quality, and its life using deep learning algorithms. *Electronics, 11*(24), 4100. https://doi.org/10.3390/electronics11244100
11. Kumar, M. S., Vimal, S., Jhanjhi, N. Z., Dhanabalan, S. S., & Alhumyani, H. A. (2021). Blockchain-based peer-to-peer communication in autonomous drone operation. *Energy Reports, 7*, 7925–7939. https://doi.org/10.1016/j.egyr.2021.09.016
12. Jhanjhi, N. Z., Humayun, M., & Almuayqil, S. N. (2021). Cybersecurity and privacy issues in industrial Internet of Things. *Computer Systems Science & Engineering, 37*(3), 357–368. https://doi.org/10.32604/csse.2021.015487
13. Lee, S., Abdullah, A., & Jhanjhi, N. Z. (2020). A review on honeypot-based botnet detection models for smart factory. *International Journal of Advanced Computer Science and Applications, 11*(6), 319–326. https://doi.org/10.14569/IJACSA.2020.0110641
14. Ngoc, N. M., Hieu, V. M., & Tien, N. H. (2023). Impact of accreditation policy on quality assurance activities of public and private universities in Vietnam. *International journal of public sector performance management*, *10*, 1-15.
15. Balkrisna, P. J. (2021). *The Development of an Authentic Indian Restaurant Concept in Portugal-Idea Implementation* (Master's thesis, Universidade NOVA de Lisboa (Portugal)).
16. Wijesooriya, C., & Basnayake, R. (2024). Digital transformation in redefining the role of the finance and audit professional of the future. In *Digital Transformation in Accounting and Auditing: Navigating Technological Advances for the Future* (pp. 61-104). Cham: Springer International Publishing.
17. Zulvia, P., & Haryanto, N. D. (2021). Increasing Customer Statisfaction Through Improving Service Quality at the Purworejo Post Office 54100. *Jurnal Ilmu Administrasi: Media Pengembangan Ilmu Dan Praktek Administrasi*, *18*(2), 195-211.
18. Ashfaq, F., Jhanjhi, N. Z., & Khan, N. A. (2023, April). Badminton player's shot prediction using deep learning. In *Innovation and Technology in Sports: Proceedings of the International Conference on Innovation and Technology in Sports,(ICITS) 2022, Malaysia* (pp. 233-243). Singapore: Springer Nature Singapore.
19. Ashfaq, F., Jhanjhi, N. Z., Khan, N. A., & Das, S. R. (2023, February). Synthetic crime scene generation using deep generative networks. In *International Conference on Mathematical Modeling and Computational Science* (pp. 513-523). Singapore: Springer Nature Singapore.
20. Alourani, A., Ashfaq, F., Jhanjhi, N. Z., & Ali Khan, N. (2023). BiLSTM-and GNN-Based Spatiotemporal Traffic Flow Forecasting with Correlated Weather Data. *Journal of Advanced Transportation*, *2023*(1), 8962283.

 doi:10.20944/preprints202504.1899.v1




21. Saeed, S., Abdullah, A., Jhanjhi, N. Z., Naqvi, M., & Humayun, M. (2020). Statistical analysis of the pre-and post-surgery in the healthcare sector using high dimension segmentation. In *Machine Learning for Healthcare* (pp. 159-174). Chapman and Hall/CRC.
22. Saleh, M., Jhanjhi, N., & Abdullah, A. (2020, February). Fatima-tuz-Zahra,''Proposing a privacy protection model in case of civilian drone,''. In *Proc. 22nd Int. Conf. Adv. Commun. Technol.(ICACT)* (pp. 596-602).
23. Kaur, N., Verma, S., Jhanjhi, N. Z., Singh, S., Ghoniem, R. M., & Ray, S. K. (2023). Enhanced QoS-aware routing protocol for delay sensitive data in Wireless Body Area Networks. *IEEE Access*, *11*, 106000-106012.
24. Singh, T., Solanki, A., Sharma, S. K., Jhanjhi, N. Z., & Ghoniem, R. M. (2023). Grey Wolf Optimization-Based CNN-LSTM Network for the Prediction of Energy Consumption in Smart Home Environment. *IEEE Access*, *11*, 114917-114935.
25. Saeed, S., Jhanjhi, N. Z., Naqvi, S. M. R., & Khan, A. (2022). Analytical Approach for Security of Sensitive Business Cloud. *Deep Learning in Data Analytics: Recent Techniques, Practices and Applications*, 257-266.
26. Srinivasan, K., Garg, L., Chen, B. Y., Alaboudi, A. A., Jhanjhi, N. Z., Chang, C. T., ... & Deepa, N. (2021). Expert System for Stable Power Generation Prediction in Microbial Fuel Cell. *Intelligent Automation & Soft Computing*, *30*(1).
27. Javed, D., Jhanjhi, N. Z., Khan, N. A., Ray, S. K., Al Mazroa, A., Ashfaq, F., & Das, S. R. (2024). Towards the future of bot detection: A comprehensive taxonomical review and challenges on Twitter/X. Computer Networks, 254, 110808.
28. Aldughayfiq, B., Ashfaq, F., Jhanjhi, N. Z., & Humayun, M. (2023, April). Yolo-based deep learning model for pressure ulcer detection and classification. In *Healthcare* (Vol. 11, No. 9, p. 1222). MDPI.