CEBU TECHNOLOGICAL UNIVERSITY-GINATILAN EXTENSION CAMPUS
San Roque , Ginatilan , Cebu

CEBU TECHNOLOGICAL UNIVERSITY - ARGAO CAMPUS
SUPPLY OFFICE REQUEST MONITORING SYSTEM

A Capstone Project Submitted to the Cebu Technological University Ginatilan Extension - Argao Campus Information Technology Department

In Compliance with the Requirements for the Degree of Bachelor of Science in Information Technology

By

Ralph Balansag Tuazon
Jhake Pasilan Dugaduga
Jeykie Rose Rubio Duhig
Jane Carmel Barro Duhig
Danica Mae Dinawanao Maceda

January 2025

APPROVAL SHEET

CEBU TECHNOLOGICAL UNIVERSITY - ARGAO CAMPUS SUPPLY OFFICE REQUEST MONITORING SYSTEM

By

RALPH BALANSAG TUAZON
JHAKE PASILAN DUGADUGA
JEYKIE ROSE DUHIG
JANE CARMEL BARRO DUHIG
DANICA MAE DINAWANAO MACEDA

Has been approved.

2025

Approving Committee:

Names	Role	Signature
Mr. Glicerio A. Baguia	Adviser	
Mr. Rochell B. Belvestre	Panel Member	
Mr. Aga Peter Duran	Panel Member	

Accepted by:

Mr. Glicerio A. Baguia
OIC-Chair, BSIT & CIT-Comp Tech
CTU-Ginatilan Extension Campus

ACKNOWLEDGEMENT

We would like to express our deepest gratitude to Professor Glicerio A. Baguia , University Chairperson and Advisor for his unwavering support, invaluable guidance and constructive feedback throughout the development of this research . His expertise and encouragement have been instrumental in shipping this study.

We also extend my heartfelt thanks to the faculty and staff of Cebu Technological University - Argao Campus , particularly the Supply Office department, for their cooperation and for providing the necessary information that was crucial to the research.

To my colleagues and classmates who provided moral support and offered constructive critiques, I am truly grateful for your collaboration and encouragement.

Lastly, We're thankful to all the individuals and groups who participated in the testing phase of the proposed Supply Monitoring Request System. Your feedback and suggestions have been invaluable in ensuring the system's usability and functionality.

TABLE OF CONTENTS

Page

TABLE OF FIGURES

ABSTRACT

Effective supply and request monitoring is significant in ensuring perfect operation of educational institutions. Managing and monitoring supply requests at the Cebu Technological University-Argao Campus is currently difficult, which causes delays, inefficiencies, and a lack of transparency. The creation of a web-based supply request monitoring system tailored to campus requirements is suggested by this study.
. The proposed system aims to automate the tracking and approval process for the supply request, improve the record-keeping and make better communication between the requester and the supply officers. It will be designed using modern technologies , including HTML , CSS , JAVASCRIPT , and a robust database system. The system will feature a user-friendly interface , real time status updated, and secure access for multiple user roles, including faculty , supply officers, and administrator .

Data will be collected and analyzed using a mixed-methods methodology. In order to determine current issues and collect system requirements, this will involve conducting surveys and interviews with important stakeholders. To assess its efficacy and make sure the system meets user expectations, prototypes will also go through usability testing.

An effective and dependable supply monitoring system reduces processing time, lowers errors, and boosts supply management transparency is the anticipated result. In addition to resolving the present problems, this technology will offer a scalable solution that may be used to other Cebu Technological University campuses.

INTRODUCTION

Cebu Technological University-Argao Campus, in collaboration with Ginatilan Extension Campus, has introduced the Inventory Supply System website project to improve inventory management and communication between requesters and the Bids and Awards Committee (BAC), Campus Director (CD), Budget Officer (BO) offices. Inspired by this initiative, the CTU-AC Supply Office wants to create a Request Monitoring System to solve present disorganization in tracking supply requests, managing inventories, and coordinating their requests. The campus wants to make its internal procedures more effective and advantageous in order to fulfill its objective including the supervision and administration of supply. The current manual tracking method used by CTU-Argao for supply requests causes delays, miscommunications, and difficulties with inventory control. These inefficiencies have a big influence on how the campus runs regularly and how well departments perform generally, especially when it comes to ensuring that they get the resources they require when they need them.

It gets more difficult to monitor inventory and request status when the demand for supplies in various departments increases. When there isn't a single system to handle everything, it is not easy for them to request an item that is needed on campus , there will be possible delays and misdivocation of the intended item. An automated supply request monitoring system is very useful for the problem that is facing in the CTU- Argao Campus. This system is simple to use and capable of modernizing the entire process, and is clearly needed in light of this research. from making a request to sending supplies, all the while giving everyone engaged real-time information.

This research aims to develop a web-based Supply Request Monitoring System for CTU-Argao that will automate the submissions , approval and tracking of supply requests. The purpose of this system is to organize the platform where users can submit and monitor their request , while the administrator can manage the inventory , approve requests and ensure timely delivery. The CTU - Argao Supply Monitoring System.

THEORETICAL BACKGROUND

The important management of supplies is intense to the operational success of any institution . This has led to development of various Inventory Management Systems , which aim to ensure that the proper control of tracking of allocation of supplies is well organized . Inventory management theory , as discussed by authors like Heizer , Render , and Munson (2017), highlights the importance of maintaining the right balance of inventory - neither too much, which moves up resources , nor too little , which leads stockouts and delays. In educational associations like Cebu Technological University Argao Campus , maintaining such a balance becomes particularly challenging as the demand for supply changes and need for timely delivery increases.

To deal with these challenges , modern institutions have turned to automated systems to modernize and improve the request process within the requester. Using technology to carry out tasks that would otherwise require human intervention is the main idea of automation theory. Automating administrative processes, such supplier requests and approvals, can greatly increase operational efficiency by lowering delays, human error, and administrative overheads, as suggested by Davenport and Short (1990) in their work on business process reengineering. By moving from manual to automated processes, organizations can ensure that the requests are processed quickly, stock levels are accurately tracked , and supplies are delivered on time .

In addition , an essential basis for understanding how information technologies helps to the enhancement of organizational workflow efficiency is provided by the Communication and Information Sharing Theory.According to the hypothesis, which is covered in Laudon and Laudon's (2016) publications, a system that centralizes data and provides real-time access to it can greatly improve communication and decision-making across various organizational units , The Cebu Technological University Argao Supply Request Monitoring System's web based platform will facilitate improved communications and the sharing of current information about the supply request with the department and administrative bodies , such as the Bids and Awards Committee (BAC) and Budget Officers(BO) , to communicate more effectively sharing up to date information regarding supply requests.

In conclusion , efficient supply management is essential to the smooth operation of establishments such as Cebu Technological University Argao Campus. It can be difficult to maintain inventory balance while demand is changing in order to prevent both shortages and excess . These obstacles can be solved by information technology and automation , which can minimize errors and delays of supply delivery . Monitoring supply demands via a web-based platform will improve departmental communication and real-time data exchange, which will improve operational flow and decision-making. This system's ultimate goal is to provide prompt, effective supply management.

CONCEPTUAL FRAMEWORK OF THE STUDY

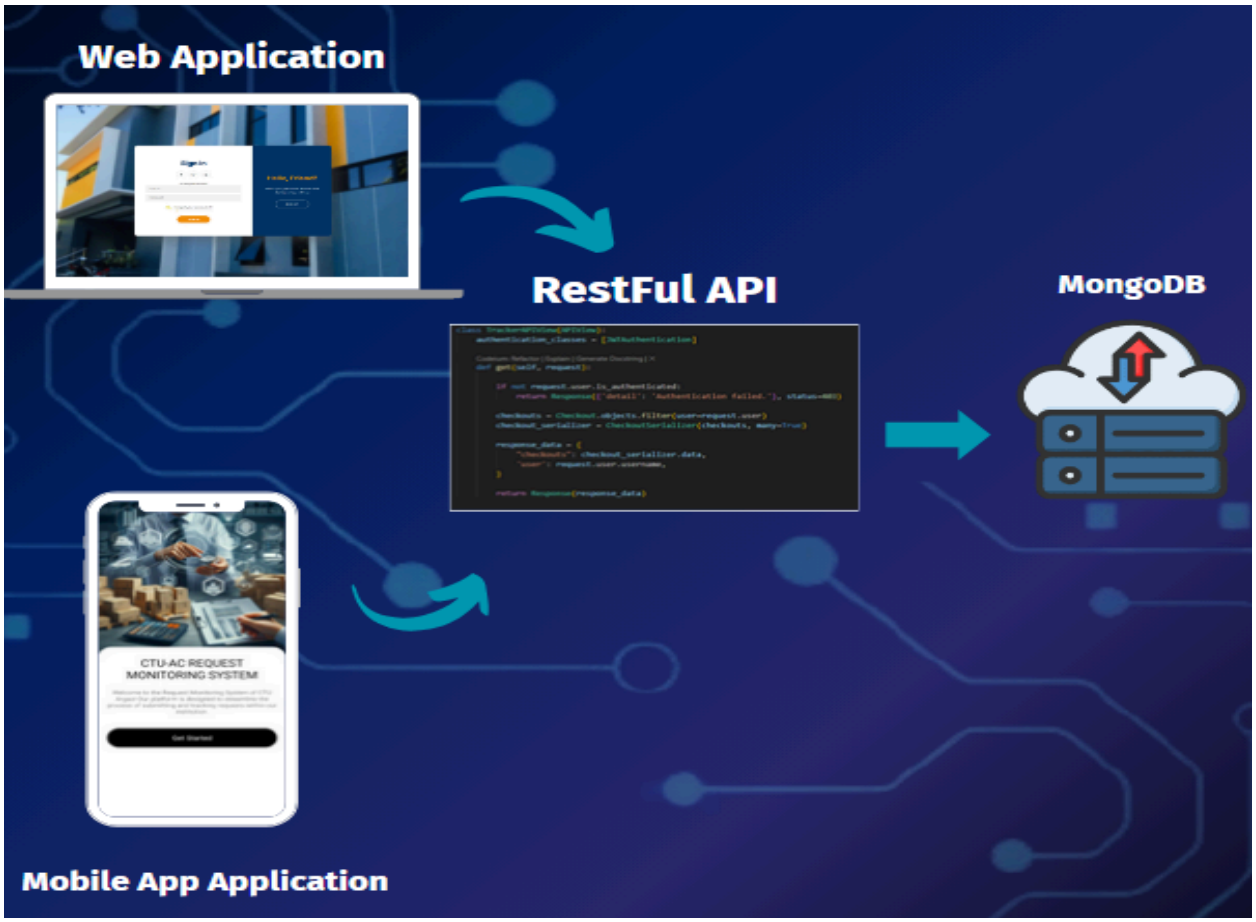


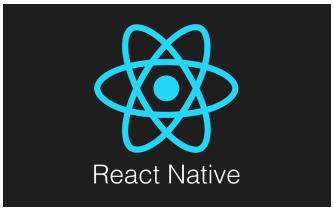
Figure 1 : Framework of the Mobile Application and Website using Restful API

This framework illustrates a system where a **Web Application** and **Mobile Application** serve as front-end interfaces for users to interact with the system, sending requests to a **RESTful API**. The RESTful API acts as the backend, processing user requests, enforcing authentication/authorization, and interacting with a **MongoDB** database to store or retrieve data. The data flow starts with user input on the web or mobile app, which is sent to the API, processed, and then forwarded to the database. The database's response is returned via the API to the requesting application, ensuring seamless communication and secure data management across platforms.

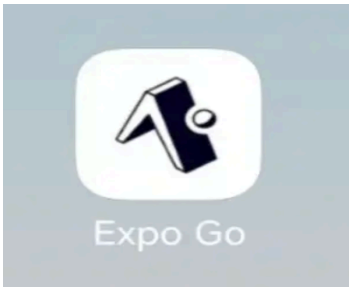
Django Framework manages backend and data
storage to ensure smooth, scalable, and reliable performance that efficiently handles interactions, data processing, and synchronization.

Rest Framework enhances productivity,
provides security features, and simplifies the overall process of building and maintaining RESTful APIs between different parts of an app.

React Native facilitates effective cross-platform development and guarantees seamless data
synchronization between the app and the backend.



Expo enhances real-time updates, streamline data synchronization, and guarantees flawless interaction
between the application and backend.



MongoDB provides a reliable and scalable database solution for managing and storing data and also provides built-in replication,
which ensures data is copied across multiple servers.

.

COMPETITOR ANALYSIS

The success of the Cebu Technological University Argao Supply Request Monitoring System can be advanced by analyzing a similar system currently in use . By identifying their strengths and weaknesses, the proposed system can improve the different existing problems facing CTU - Argao Campus in tracking items that the user has requested.

Table 1 : Comparison Between Mobile App Application and Web Application System

System	Strengths	Weaknesses
CTU ARGAO CAMPUS SUPPLY OFFICE REQUEST MONITORING SYSTEM (Web Application)	• Easier to maintain and update from a centralized server. • Broader compatibility across devices and operating systems. • Better suited for detailed workflows and reporting.	• relies on a stable internet connection. • Lack of real-time notification functionality compared to mobile apps. • Less portable compared to mobile devices.
CTU ARGAO CAMPUS SUPPLY OFFICE REQUEST MONITORING SYSTEM (Mobile App Application)	• Accessible anytime , anywhere . • High portability and convenience. • Real-time notifications via push technology.	• Requires frequent updates and compatibility testing. • Limited screen space for displaying detailed information. • Device-dependent performance.

The possibility of combining the advantages of both systems into a hybrid solution is another crucial factor to take into account when doing a competition analysis. Although the mobile application provides unmatched portability and real-time notifications, the online application's wider compatibility and adaptability for intricate processes can offset the mobile app's screen size and device dependency constraints. Similar to this, using the offline capabilities of the mobile app for specific functionalities can help offset the online application's dependence on consistent internet connectivity. By merging the advantages of both systems, the CTU Argao Supply Request Monitoring System can offer a more complete and reliable solution that meets the numerous demands of its users while guaranteeing effectiveness and accessibility in a range of situations.

Statement of the Problem

This research addresses the current manual system used by Cebu Technological University-Argao Campus for tracking and managing supply requests has proven to be disorganized and problematic. The lack of an automated platform results in several operational challenges that hinder the smooth execution of requests and supply processes. The research aims to :

1. Develop a web-based Supply Request Monitoring System for CTU-Argao that will automate the submissions , approval and tracking of supply requests.

2. Make it easier for requesters, the Bids and Awards Committee (BAC), the Campus Director (CD), and the Budget Officer (BO) to communicate and work together smoothly, ensuring that requests are processed quickly and efficiently.

3. Provide stakeholders with real-time access, reduce delays and errors to the status of supply requests and inventory, ensuring transparency and accountability throughout the process.

Scope and Limitations of the Study

The research project focused on developing and implementing an Cebu Technological Argao Campus Request Monitoring System for improving the efficiency and transparency of supply requests. The development focused on automating the process of creating purchase requests, getting them approved, and tracking supply requests. It also allows administrators to easily manage and oversee the requests made by users. The study also included a user-friendly interface to ensure ease of use for all staff levels and it is also accessible via desktop and mobile devices within the campus network. Additionally , the system also incorporates user authentication to secure access and protect sensitive data, allowing only authorized personnel to manage and monitor requests. Furthermore, the project aims to address the current needs of the campus while being future-proofed to accommodate additional functionalities and scalability. The campus can expect improved management of requests, better monitoring of supply chains, and increased operational efficiency across its departments.

Several constraints might have impacted the extent and applicability of the study's findings. This includes geographic scope, the project relies on a campus network and the performance is limited to the reliability of the network infrastructure. It also required internet connection on both desktop and mobile devices for real-time functionality and synchronization of the process. Additionally the system has been designed to accommodate future enhancements, there could be constraints in scalability if user demand grows substantially beyond the current projections without significant upgrades. Integration Challenges, the new system needs to integrate with existing administrative software, and full integration might not be achieved due to compatibility issues. This can lead to inefficiencies such as manual data entry, affecting the efficient management of resources. Finally, the project must adhere to regulatory or institutional guidelines, which might restrict certain functionalities or require modifications to ensure compliance.

Methodology

The procurement process involves multiple stakeholders, including requesters and administrators, working together to ensure the efficient acquisition of goods or services.This methodology outlines the steps involved in the procurement process from both perspectives, aiming to provide a clear, structured approach to manage and track purchase requests.

The requester's process begins with registration and login to ensure only authorized users can create purchase requests, followed by filling out a purchase request form with necessary details, and then submitting and tracking the request through various approval stages. The administrator's process also starts with registration and login, followed by viewing and reviewing all submitted purchase requests, and updating the request status to keep the procurement process transparent and up-to-date. Detailed steps for requesters include navigating to the registration page, filling out personal and professional details, logging in, creating a purchase request with specific item details, andtracking the request status.

Administrators ensure their accounts are set up, log in to the system, view and filter submitted requests, and update the status to reflect the current procurement stage. Requesters are responsible for creating and tracking requests, ensuring accuracy, while administrators review, update, and manage request statuses to ensure transparency and accuracy in the process. Tools and technologies supporting this methodology include a user interface for registration, login, and request forms, a database for storing information, and a notification system for alerts and updates. This structured approach ensures transparency, accountability, and timely acquisition of goods and services, with regular reviews and updates for continuous improvement and to address emerging needs or issues.

This methodology provides a structured approach to manage the procurement process efficiently. By clearly defining the steps and responsibilities of both requesters and administrators, the process ensures transparency, accountability, and timely acquisition of goods and services. Regular reviews and updates to the system should be conducted to ensure continuous improvement and address any emerging needs or issues.

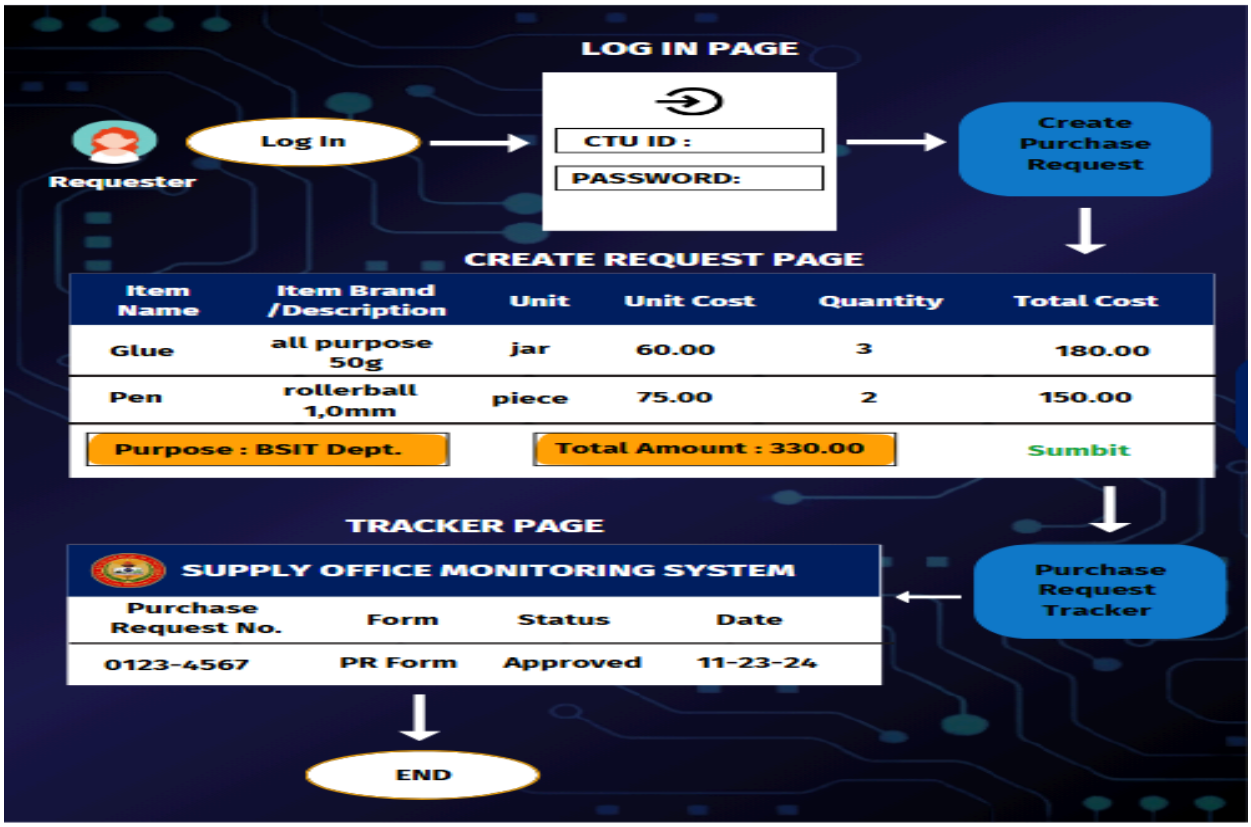


Figure 2: Requester's Dashboard Procurement Process

The diagram illustrates the procurement process on the requester's page. To initiate a purchase request, the requester must be registered and logged into the system. Once logged in, the requester can create a purchase request by filling out the provided form. Upon submission, the purchase request will be reflected on the tracker page, allowing the requester to monitor its progress from the budget officer approval to item receipt.

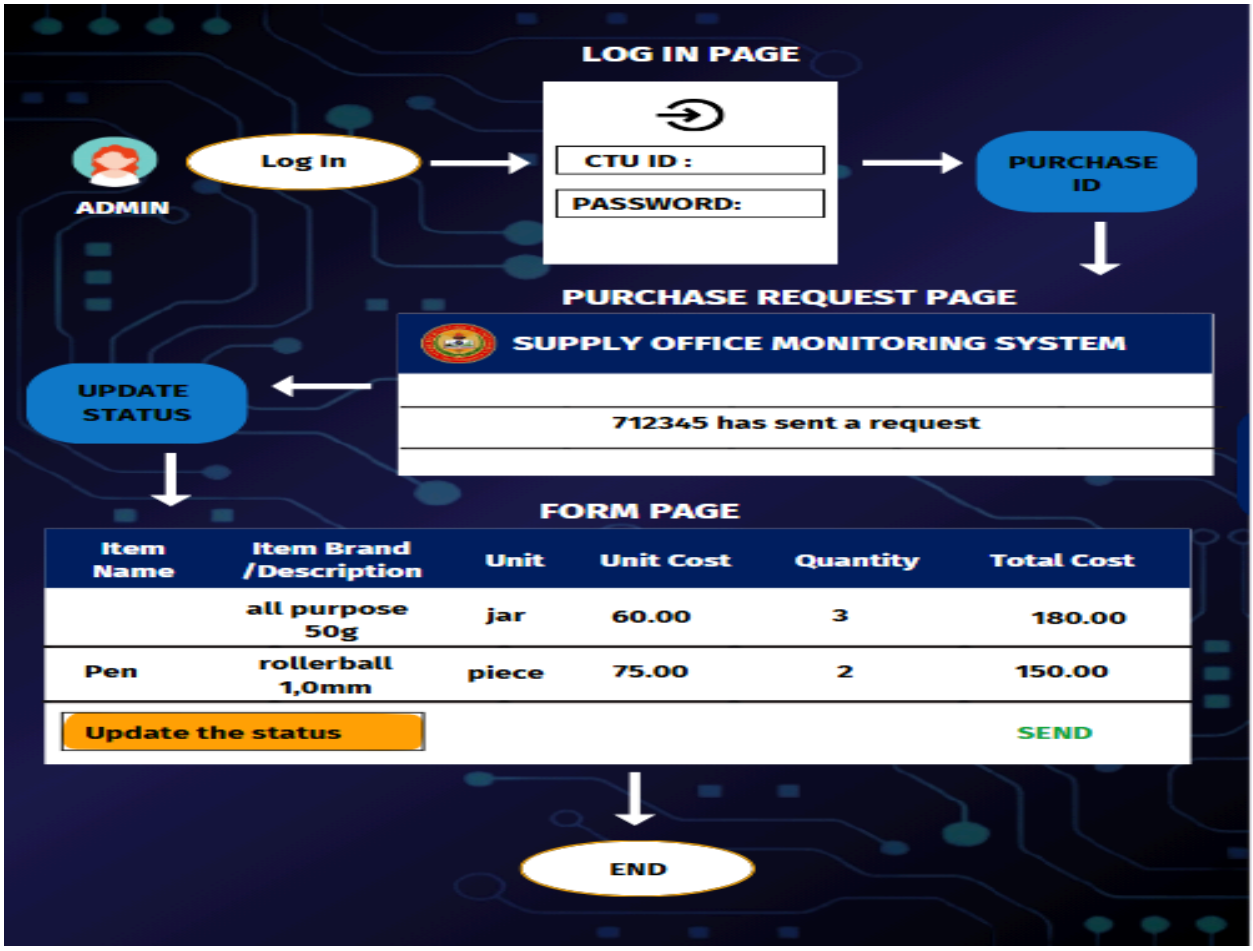


Figure 3: Administrator's Dashboard Procurement Process

This diagram outlines the procurement process on the administrator's dashboard. Assuming the administrator is registered and logged in, they can view purchase requests submitted by requesters.The administrator is responsible for updating the request status, tracking its progress from the budget officer approval to item receipt.

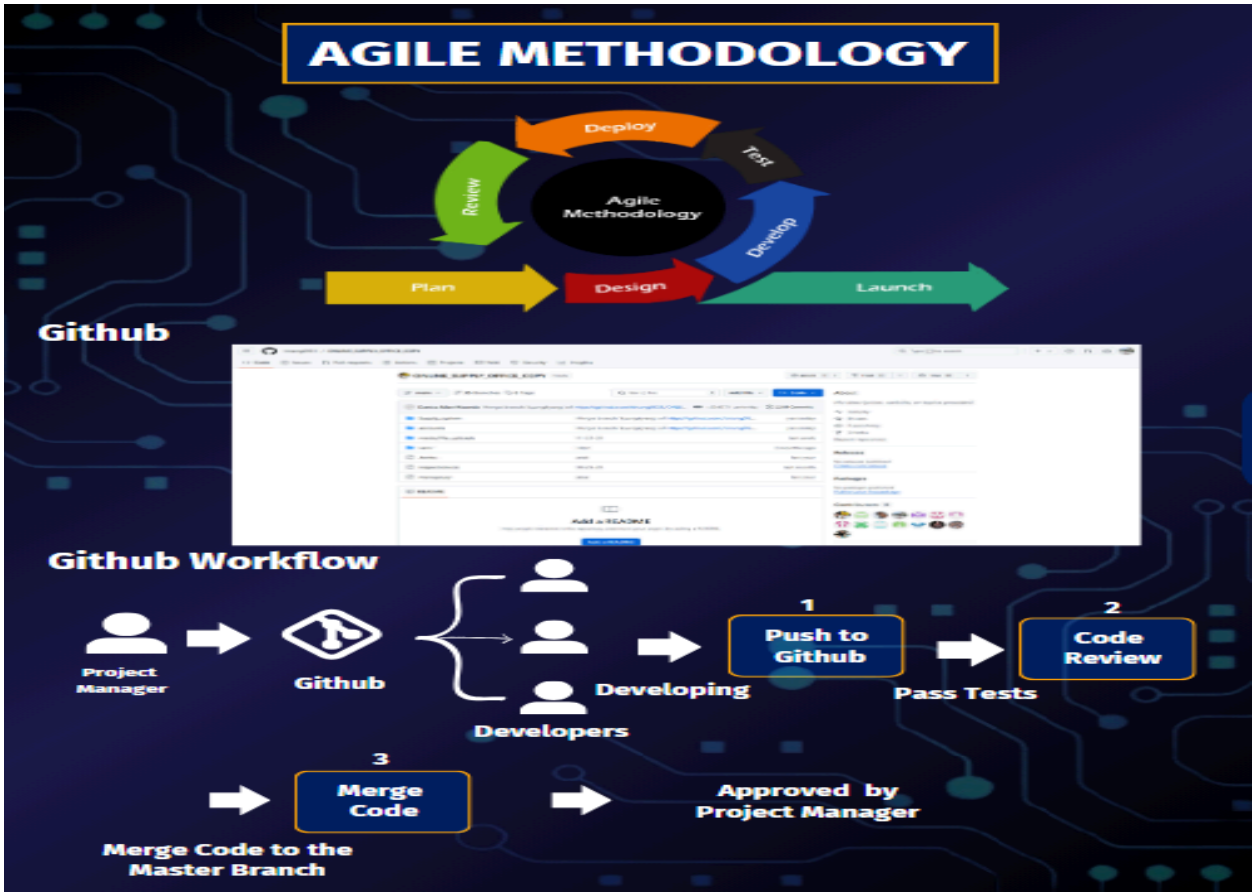


Figure 4: Agile Methodology

This methodology is best for projects that require flexibility and continuous improvement, and are well-suited for small to medium-sized teams. Agile focuses on delivering individual pieces of software, rather than the entire application. One of the examples of the agile methodology that we are currently using in our web application is the github, github is a web-based platform for version and collaboration that allows developers to host, manage, and track changes in their code projects.

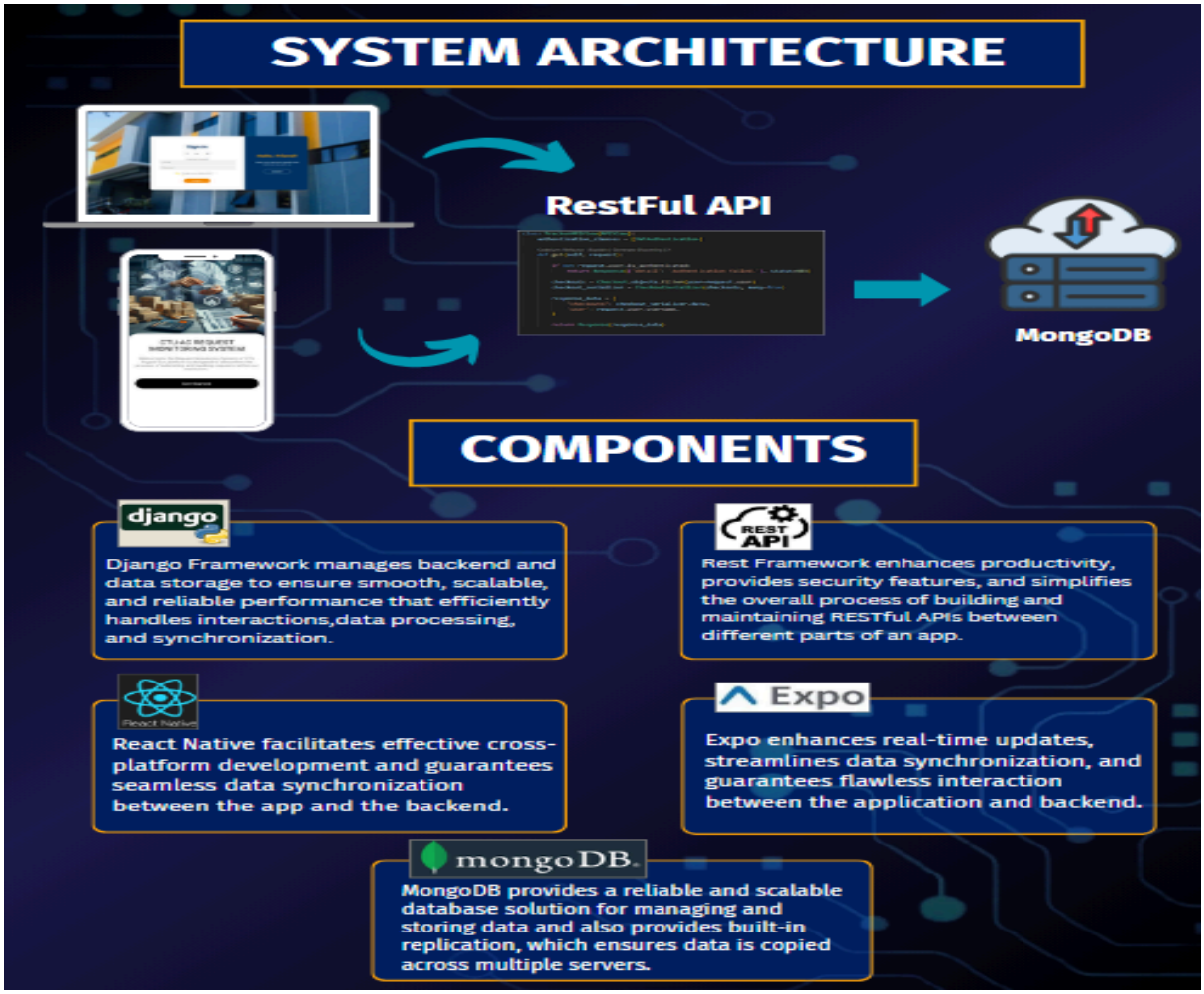


Figure 5: System Architecture

It combines multiple powerful technologies to create a strong, scalable, and efficient application ecosystem. The Django Framework's fundamental function is to manage backend processes, assuring seamless, reliable performance and efficient data processing, synchronization, and user interactions. In addition, the Django REST Framework (DRF) makes it easier to create and manage RESTful APIs, allowing for seamless communication between the frontend and backend. On the frontend, React Native makes cross-platform app development easier, providing a consistent experience for iOS and Android consumers, while Expo improves real-time updates and streamlines interactions between the app and backend. The architect's data layer is powered by MongoDB, a scalable and dependable database system that handles data storage and replication while maintaining high availability and integrity. These components work together to produce a coherent system in which the mobile app connects with the backend via RESTful

APIs, the backend processes requests and communicates with the database, and the database securely stores and organizes data, resulting in a seamless user experience.

NoSQL databases are a type of database that is designed to store and manage large amounts of unstructured data. The databases' features are scalable, flexible, and performance,which makes them well-suited for the needs of online supply websites. This makes a good choice for online supply websites, which typically have a lot of product data, customer data, and order data to store. With this, it allows efficiency and lesser costs.

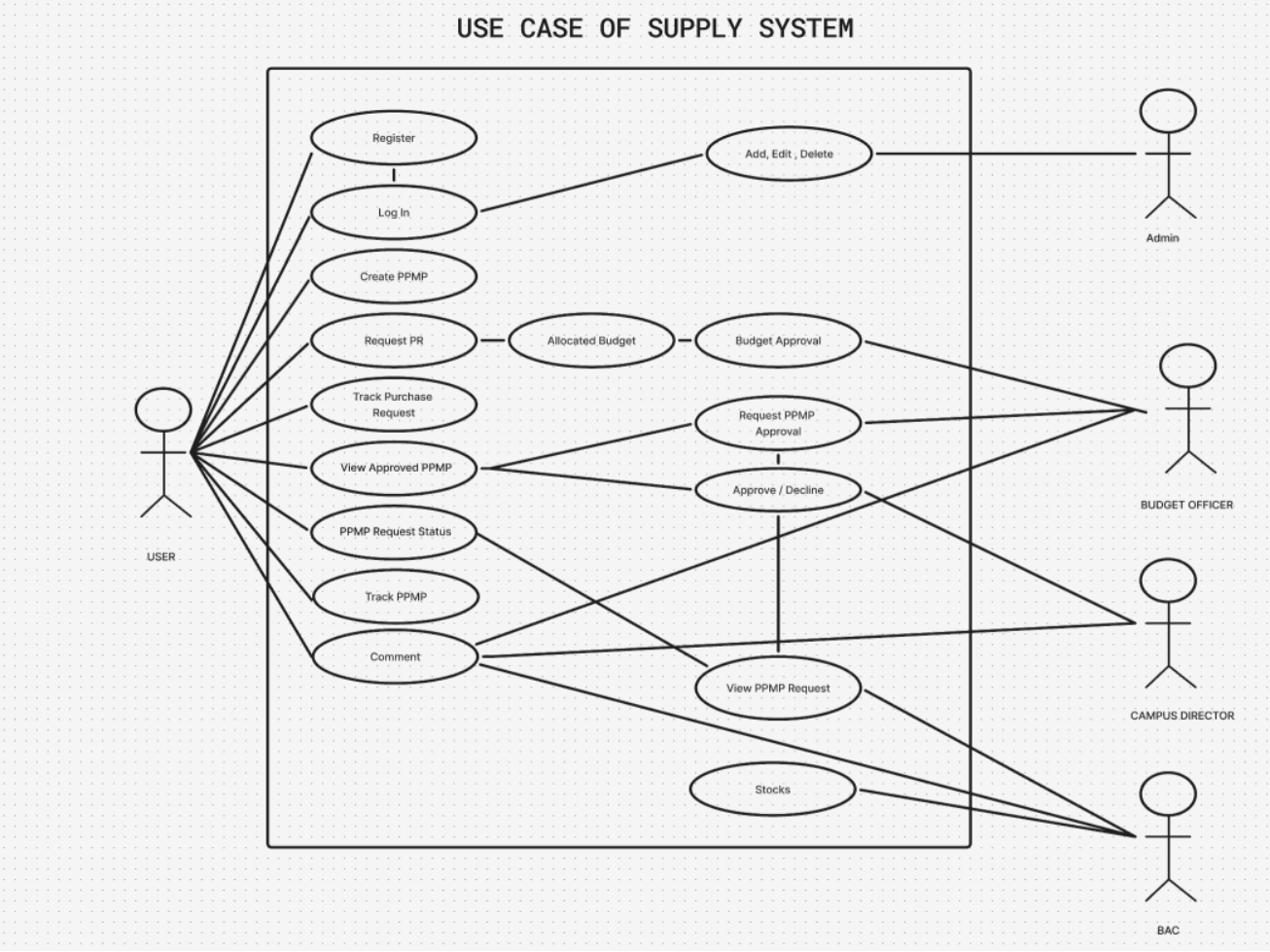


Figure 6: Use Case of Supply System

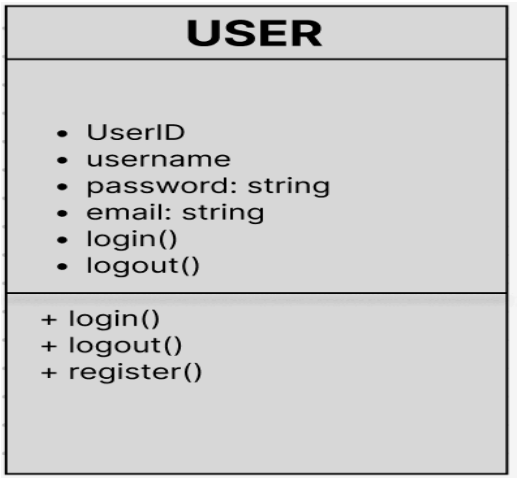


Figure 7: Class Diagram User

The class diagram represents a `USER` class in a Unified Modeling Language (UML) format. It defines a user entity with attributes and methods essential for managing user-related functionalities in a system. The class is divided into three sections: the class name, attributes, and methods.

The **attributes** section lists the key properties associated with a user. These include `UserID`, a unique identifier for each user; `username`, which stores the user's chosen name for identification; `password` and `email`, both represented as strings, to store the user's login credentials and contact details. These attributes form the core data needed to manage and identify users.

The **methods** section defines both private and public functions associated with the `USER` class. The private methods (`login()` and `logout()`) are internal to the class and likely handle the core processes of logging in and out. In contrast, the shaded section lists public methods (`+ login()`, `+ logout()`, and `+ register()`), accessible by external components of the system. The `login()` and `logout()` methods allow users to securely access and exit their accounts, while the `register()` method enables the creation of new user accounts.

The primary purpose of this class is to encapsulate user-related data and operations, ensuring secure management of user authentication and registration. It adheres to the principles of object-oriented programming by combining data (attributes) and behavior (methods) into a single, reusable structure. This design promotes modularity, making the `USER` class an essential component of systems that require user management, such as web applications, online platforms, or enterprise systems.

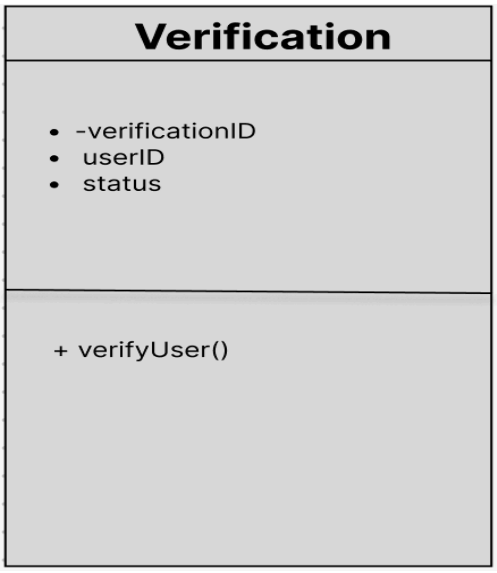


Figure 8: Class Diagram of Verification

The `Verification` class in UML is designed to manage user verification processes within a system. It includes three attributes: `verificationID`, a private unique identifier for the verification process; `userID`, which associates the verification with a specific user; and `status`, which tracks the current state of the verification (e.g., pending, approved, or rejected). The class also includes a public method, `verifyUser()`, which handles the process of verifying a user by interacting with the attributes, such as updating the `status` based on specific conditions. The primary purpose of this class is to ensure user identities are validated, enhancing system security and maintaining integrity in access control processes.

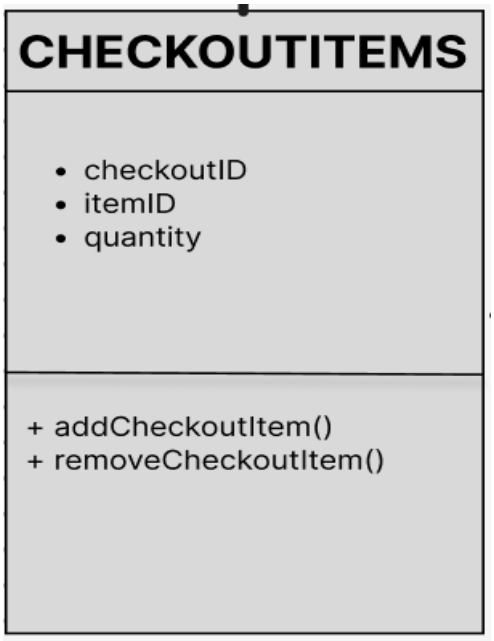


Figure 9: Class Diagram of Checkout Item

The `CHECKOUT ITEMS` class represents a structure for handling items in a checkout system, commonly used in e-commerce or point-of-sale applications. It contains three attributes: `checkoutID`, which acts as a unique identifier for the checkout session; `itemID`, which identifies the specific item being purchased; and `quantity`, which denotes how many units of the item are being checked out.

The class also features two key methods: `addCheckoutItem()` and `removeCheckoutItem()`. The `addCheckoutItem()` method is responsible for adding an item to the checkout list, potentially increasing the quantity if the item already exists. Conversely, `removeCheckoutItem()` allows for the removal of an item from the checkout, either completely or by decreasing the quantity. Together, these methods facilitate the management of items during the checkout process, enhancing user experience and ensuring accurate transaction handling.

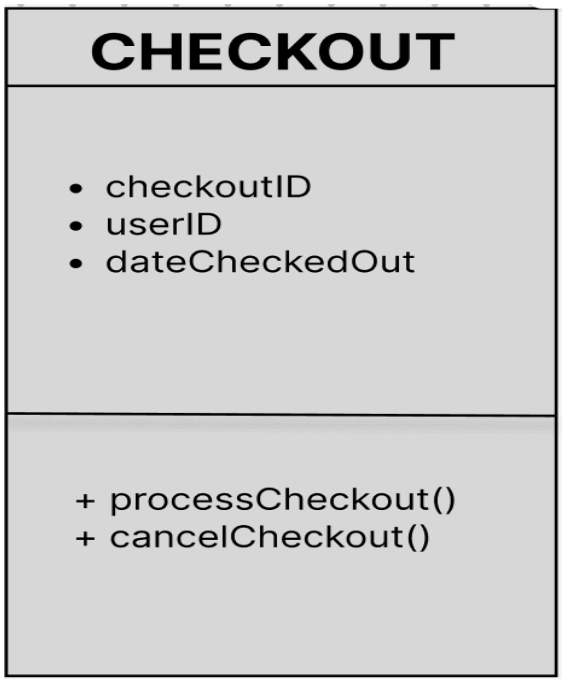


Figure 10: Class Diagram of Checkouts

The `CHECKOUT` class in UML represents the process of managing item checkouts in a system. It includes three attributes: `checkoutID`, which uniquely identifies the checkout transaction; `userID`, which associates the transaction with a specific user; and `dateCheckedOut`, which records the date the checkout occurred.

The class also defines two public methods. The `processCheckout()` method handles the logic for completing the checkout process, such as updating records or confirming the transaction. The `cancelCheckout()` method provides functionality to cancel an existing checkout, possibly reversing any changes made during processing.

The purpose of the `CHECKOUT` class is to encapsulate data and operations related to checkout transactions, ensuring accurate and efficient handling of this process in a system. It plays a crucial role in systems like e-commerce platforms, library management systems, or inventory tracking applications.

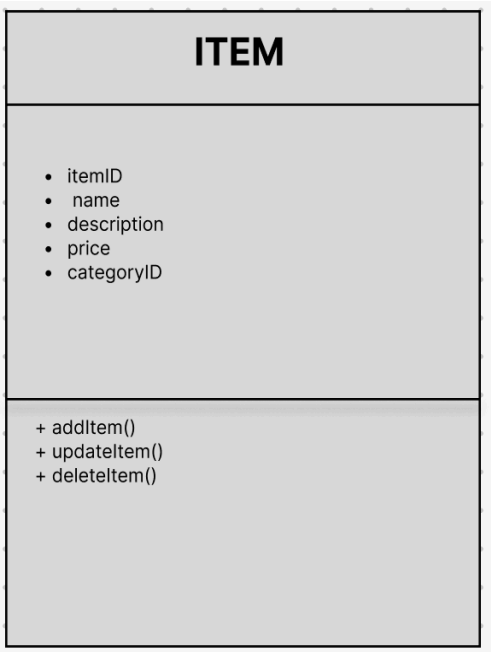


Figure 11: Class Diagram of the Item

The class diagram for the `ITEM` class consists of three sections: class name, attributes, and methods. The class name, `ITEM`, identifies the class. The attributes listed in the middle section include `itemID`, `name`, `description`, `price`, and `categoryID`, defining the properties of the `ITEM` class. The methods listed in the bottom section are `addItem()`, `updateItem()`, and `deleteItem()`, which facilitate adding, updating, and deleting items in the system. This diagram serves as a blueprint for developers, guiding the creation, manipulation, and management of items in the software application.

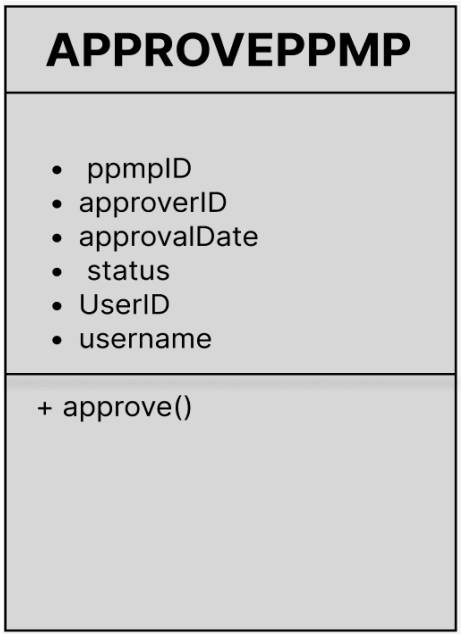


Figure 12 : Class Diagram of Approved PPMP

The class diagram presents an `APPROVED PMP` class with six attributes and one method. The attributes are `ppmpID`, `approverID`, `approvalDate`, `status`, `UserID`, and `username`, which define the properties of the class. The class also includes one method, `approve()`, which outlines the behavior of the class for approving an item. The purpose of this class diagram is to provide a clear, structured representation of the `APPROVED PMP` class, guiding developers in understanding how to implement approval processes within a software application.

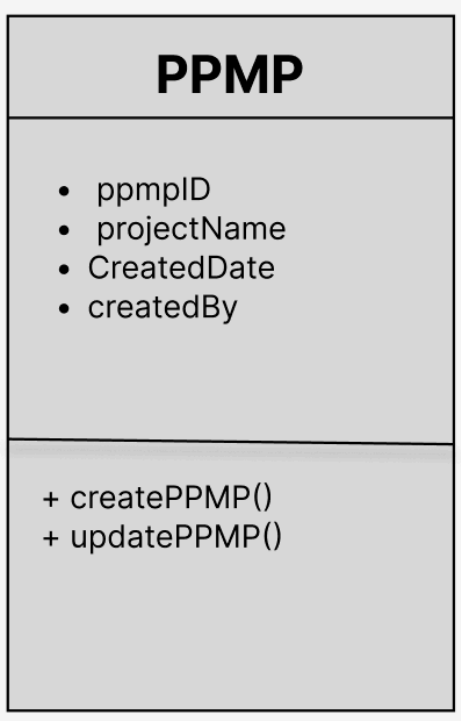


Figure 13 : Class Diagram of PPMP

The class diagram for the `PPMP` class consists of attributes and methods that define its properties and behavior. The attributes include `ppmpID`, `projectName`, `createdDate`, and `createdBy`, which outline the key properties of the `PPMP` class. The methods, `createPPMP()` and `updatePPMP()`, define the actions that can be performed on instances of this class, allowing the creation and updating of `PPMP` objects. The purpose of this class diagram is to provide a structured and clear representation of the `PPMP` class, aiding developers in understanding how to implement and manage PPMP objects within a software application.

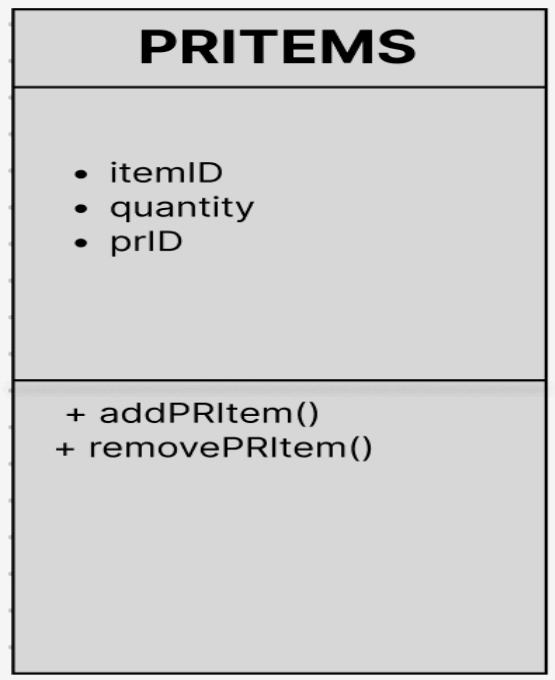


Figure 14 : Class Diagram of PR Items

The class diagram for `PRITEMS` showcases three distinct sections. The top section identifies the class with the name `PRITEMS`. In the middle section, the attributes listed include `itemID`, `quantity`, and `prID`, defining the properties that instances of the `PRITEMS` class will possess. The bottom section outlines the methods `addPRItem()` and `removePRItem()`, which detail the actions that can be performed on instances of this class. The purpose of this class diagram is to provide a clear and structured representation of the `PRITEMS` class, aiding developers in understanding how to implement, add, and remove purchase request items within a software application.

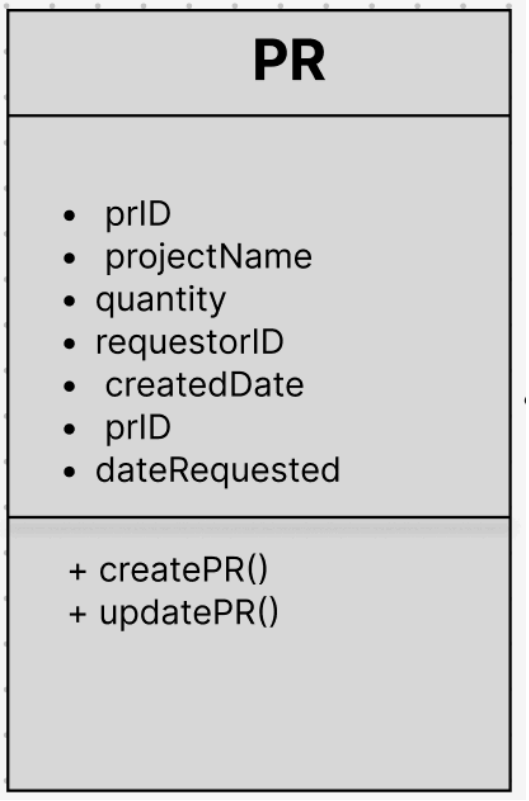


Figure 15 : Class Diagram of PR

The class diagram for the `PR` class includes various attributes and methods that define its structure and behavior. The attributes listed are `prID`, `projectName`, `quantity`, `requestorID`, `createdDate`, a duplicate `prID`, and `dateRequested`, which specify the key properties of the `PR` class. The methods `createPR()` and `updatePR()` outline the actions that can be performed on instances of this class, allowing the creation and updating of `PR` objects. The purpose of this class diagram is to provide a clear and structured representation of the `PR` class, guiding developers in understanding how to implement and manage `PR` objects within a software application.

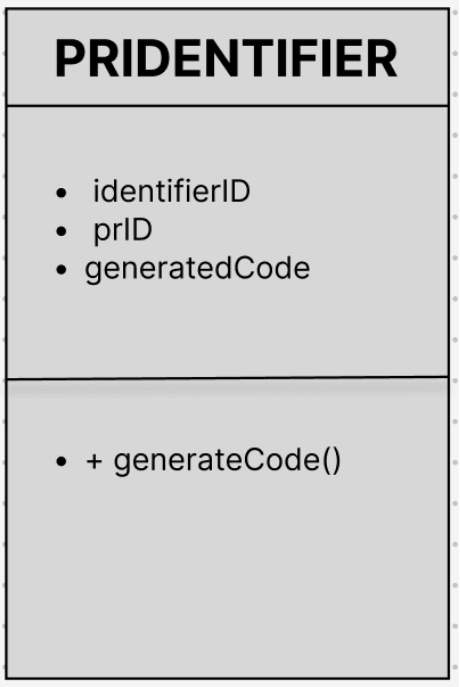


Figure 16 : Class Diagram of Pr Identifier

The class diagram of the Purchase Request Identifier is designed to keep track of important information related to a purchase request. It includes three main parts: the Identifier ID, which gives each request a unique number for easy tracking; the PR ID (Purchase Request ID), which connects the identifier to a specific purchase request; and the Produced Code, which is created automatically to clearly label and organize the request. Together, these elements help manage and reference purchase requests in the system.

Figure 17 : Class Diagram of History

The class diagram for the `History` class includes several key components. It has four attributes: `historyID`, `userID`, `action`, and `timestamp`, which define the properties of the class. The class also includes two methods: `addHistory()` and `getHistory()`. The `addHistory()` method allows for the addition of new history records, while the `getHistory()` method retrieves the history records. The purpose of this class diagram is to provide a clear and structured representation of the `History` class, guiding developers in understanding how to implement and manage historical actions within a software application.

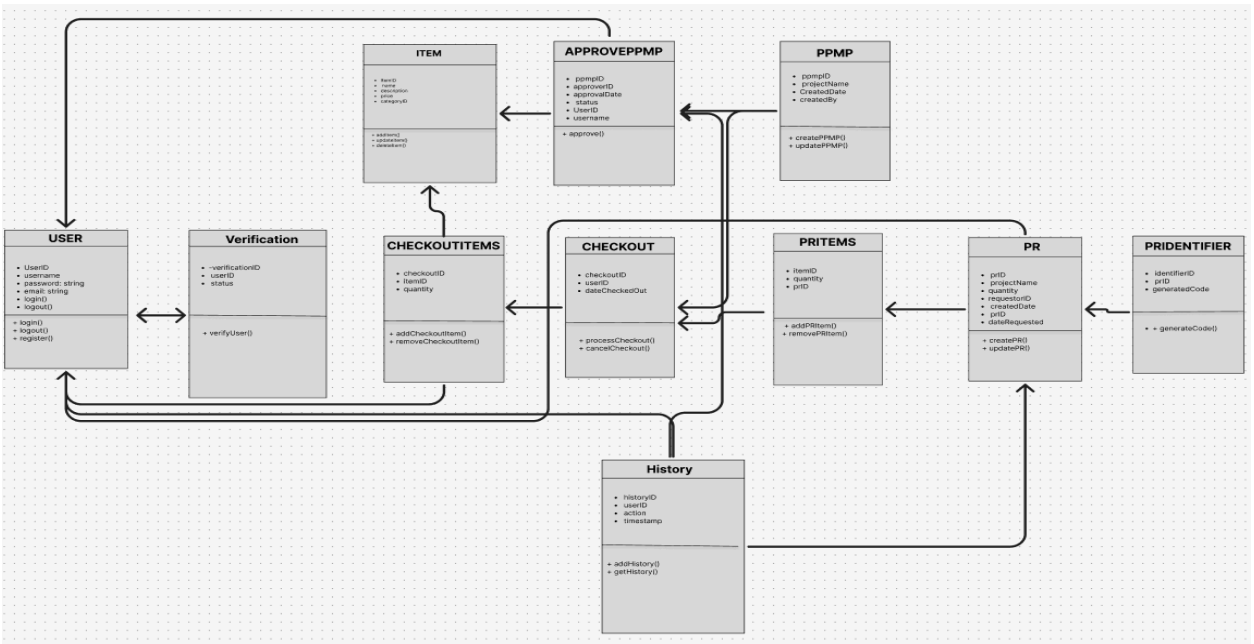


Figure 18: The Combination of the Class Name Diagram

Figure 18 shows the combination of class name diagrams that represent the structure of a system by illustrating the relationships between various classes. It helps to understand how the different components of the system interact with each other within the system. The diagram includes classes, which represent key components or entities like user roles, system features, or business logic elements. Each class contains attributes, data or properties and methods, functions or behaviors that define its characteristics and actions. This reflects how the system processes user information or responds to user inputs. Additionally, the diagram shows the relationships between classes, such as inheritance where one class inherits attributes and methods from another , associations indicating how classes interact or depend on each other, and aggregation or composition where one class is made up of objects from other classes. For instance, it may show how classes involved in user authentication interact with those managing user profiles, or how transaction management classes relate to activity tracking classes. In essence, this diagram serves to highlight both the technical structure of the system and how its design influences the user experience, offering insights into how the different parts of the system work together to support the overall use of the system.

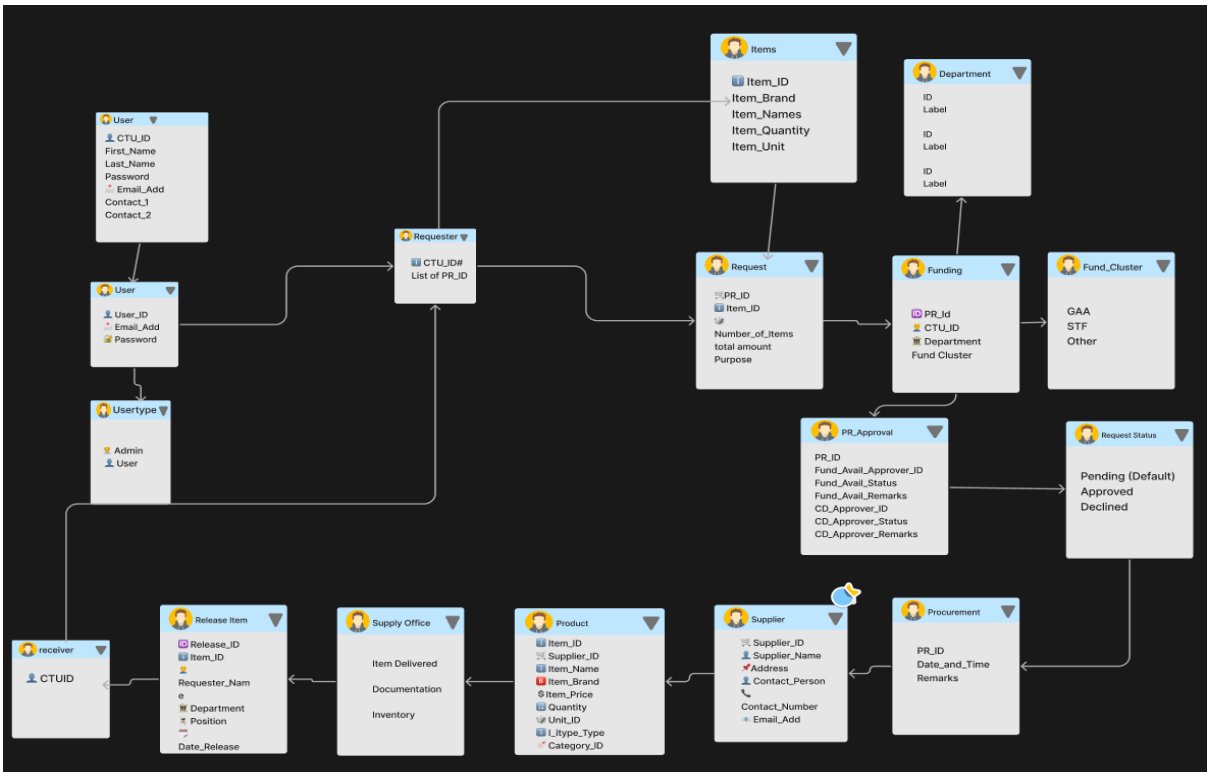


Figure 19: Schematic Diagram of the Supply Monitoring System (Database Schema)

Figure 19 illustrates the schematic diagram of the Supply Monitoring System through a procurement procedure that involves several parties and their interactions. In the core of the system, users with designated roles—such as requesters—submit purchase requests that specify the goods required, their quantities, and their intended uses. The relevant funds are distributed from several fund clusters, and these requests are connected to the corresponding departments. Following a thorough review process, the requests are approved or denied by a number of approvers who confirm the funds' availability. Prior to the completion of any procurement, this approval procedure guarantees that the required checks and balances are in place.

In order to ensure correct paperwork and delivery, the supply office oversees the transfer of commodities to recipients after a request has been granted. Throughout the process, careful attention is paid to maintaining comprehensive product information, including supplier information, item brands, prices, and quantities. The procurement process is made more efficient and transparent by this methodical approach.

RESULTS AND DISCUSSION

NoSQL databases are great for dynamic projects since they can handle a wide variety of data formats. MongoDB is a NoSQL database that was utilized in the Supply Office Request Monitoring System, which officers flexible data storage and efficient handling of unstructured data. Through the use of sharding, a technique that improves data processing security by splitting the database into independent sections, the system was able to scale horizontally. In contrast, typical SQL databases rely on vertical scalability and frequently struggle with unstructured data.

The CTU-Argao Campus Supply Office Request Monitoring System usage of NoSQL databases highlights scalability, flexibility, and the capacity to handle a large amount of data, all of which may be advantageous for future online supply websites. This project is beneficial for CTU-Argao Campus and other institutions by creating opportunities for economic development, reaching a wider audience for education, and raising growth in business.

Table 2: Experimental Results of Various Web Application Features

Experiment	Training	Results
Experiment 1	Request Tracking and Management	The system attempted to automate supply request tracking, but issues were encountered in reducing manual processing delays and providing real-time status updates. Example: Requesters faced challenges in tracking the progress of their requests, from submission to approval.

Experiment 2	Communication between Stakeholders	The notification system was integrated to ensure seamless communication between requesters and administrators, enhancing transparency and reducing confusion by providing updates on request approvals, denials, and additional requirements. **Data:** 90% of users reported receiving timely updates regarding their supply requests.
Experiment 3	Increased Operational Efficiency	The system's inventory management module increased supply monitoring by reducing misallocation and stockouts while also assuring prompt identification and distribution of requested items. **Performance Metrics:** Request processing time decreased by 40% compared to the manual system.
Experiment 4	User Satisfaction	Both requesters and administrators expressed high satisfaction with the system’s ease of use, intuitive design, and ability to provide accurate, real-time data. **Survey Results:** 85% of users rated the system as "very effective" in addressing previous inefficiencies.

Experiment 5	Database	The database was optimized for handling high transaction volumes, resulting in a 30% reduction in query execution time and 80% uptime during testing.

The main problem with CTU Argao is that they manually track the requests from various departments, which is time-consuming and this system was implemented to streamline operations, increase productivity, and focus on higher-priority tasks at CTU-Argao, reducing errors and inefficiencies. The Supply Request Monitoring System at the CTU-Argao Campus has enhanced supply management operations by lowering errors and increasing transparency. The system streamlines the submission process by allowing authorized users to create and manage requests, and it offers real-time updates to both requesters and administrators. It also improves inventory management, prevents over-requests, and reduces resource misallocation. The system bridges communication gaps among stakeholders, ensuring timely approvals and reviews. Regular training and hybrid solutions can help to address system accessibility difficulties. The system is a transformative instrument for increasing efficiency and resource usage.

```
@authenticated_user
def addItem(request):
    if request.method == 'POST':
        item_data = request.POST.get('item')
        item_brand_description = request.POST.get('item_Brand_Description')
        unit = request.POST.get('unit')
        unit_cost = request.POST.get('unit_Cost')



        user = request.user
        Item.objects.create(
            user=user,
            item=item_data,
            item_brand_description=item_brand_description,
            unit=unit,
```

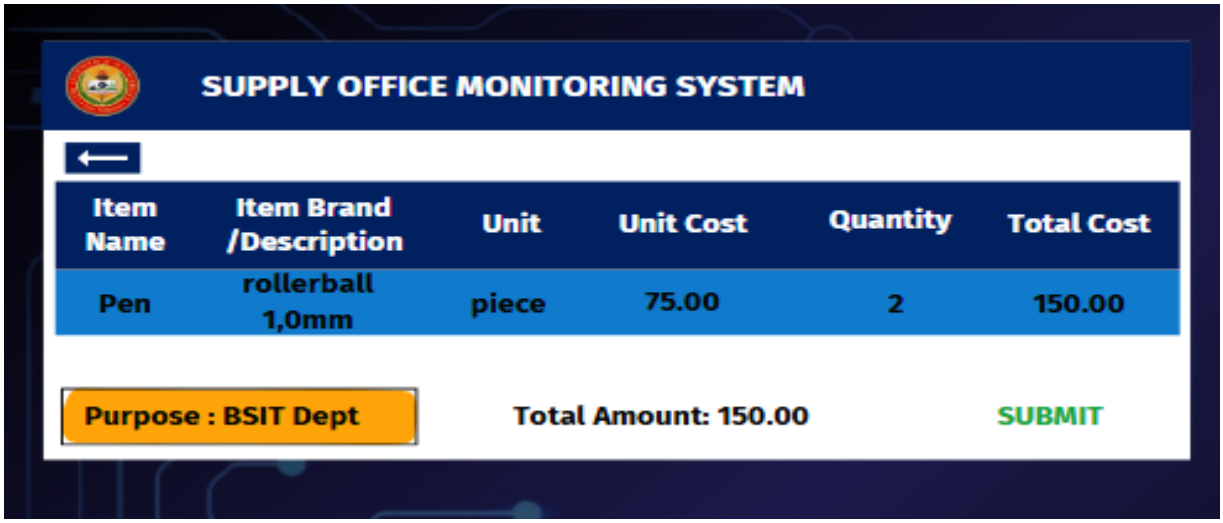
```
        unit_cost=unit_cost,
    )

    return redirect('ppmp')

return render(request, 'accounts/User/ppmp.html')
```

Visual Studio Source Code

This source code represents the ‘add item’ function responsible for containing new items in the system. In that case, item data, including the related modal, is extracted from the request body. Then the extracted data is utilized to generate a new item object, which is stored in the MongoDB database.



Purchase Request

This image illustrates the user interface where users create requests for specific items or products. Users also have the ability to edit their requests if they need to make any changes. During this process, users can view the details of their requested items, obtaining information about the status of each request.

CONCLUSION AND FUTURE RECOMMENDATIONS

In this project, the development and implementation of CTU AC Supply Office Request Monitoring System successfully addressed the challenges related to manual tracking , processing, and monitoring of supply requests within the office. Furthermore, these projects have proven to be valuable for monitoring, requesting tracking of supply, and approving requests. The system's user-friendly and real-time updates will help both requesters and administrators stay informed and engaged at each step of the process. The project successfully automated and streamlined the manual processes of tracking, processing, and monitoring supply requests, enabling the supply office to manage them more efficiently and accurately.

Although this project demonstrates a possibility for success in the future , there are areas for improvement and expansion. First, consider the regular training sessions conducted for both requesters and administrators to ensure that all users are fully aware of the features and functionalities of the Supply Request Monitoring System. Providing assistance for any potential system-related problems would also fall under this category. Then, to make sure it operates effectively, regular performance monitoring must be improved. Regular updates are made to the system to fix any bugs, security flaws, or add new features.

While the current focus is on managing purchasing requests and tracking supplies and approvals using the website and the mobile application. Getting the system to interface with other supplementary office tools such as inventory management and financial tracking systems would provide an added perspective and help in optimizing operational efficiency; scalability should ensure it can hold more requests and users as the organization grows.

Additionally, improving the functionality of the mobile app and creating an interface that is more user-friendly would significantly increase the overall experience of the users. Automated alerts and notifications all along the supply request process would keep people informed and provide transparency so nothing is missed in the process. Expanding the automation in the workflows may reduce the number of manual tasks required to be performed. Access to the system could be had from any location if the system is moved into a cloud-based architecture. There would be a

lot of room for scalability on demand. This, in return, would provide much more effective and efficient operation within the organization.

REFERENCES

[1] Alfian, Ganjar, Muhammad Syafrudin, and Jongtae Rhee. 2017. "Real-Time Monitoring System Using Smartphone-Based and NoSQL Database for Perishable Supply Chain" Sustainability 9, no. 11: 2073
Real-Time Monitoring System Using Smartphone-Based

[2] Sassoon, David M. "Monitoring the Procurement Process Vol. 12, Iss. 2, (Jun 1, 1975): 11.
https://www.proquest.com/openview/a0a4518a34b5cca2b097ef0941845315/1?pq-origsite=gscholar&cbl=1819673

[3] Santoso, M., & Noche, B. (2015). Mobile App Development for Tracking System and Supply Chain Monitoring . *Operations and Supply Chain Management: An International Journal*, 8(1), 48-56.
https://journal.oscm-forum.org/publication/article/mobile-apps-development-for-tracking-system-and-supply-chain-monitoring-

[4] Jang, S., & Lee, E. (2009). Reliable mobile application modeling based on open API. In *Communications in computer and information science* (pp. 168–175).
https://doi.org/10.1007/978-3-642-10619-4_21

[5] Ziegel, E. R., & Winston, W. (1988). Operations Research: Applications and Algorithms. *Technometrics*, *30*(3), 361.
https://doi.org/10.2307/1270107

[6] De Araújo, M. C. B., Alencar, L. H., & De Miranda Mota, C. M. (2017). Project procurement management:*International Journal of Project Management*, *35*(3), 353–377.
https://www.sciencedirect.com/science/article/abs/pii/S0263786317300595

[7] Kuitunen, M. (2019). *Cross-Platform Mobile Application Development with React Native*.
https://trepo.tuni.fi/handle/123456789/27139

[8] L. Sani, D. K. Basuki and M. M. Bahctiar, "Supply Delivery Monitoring System Using Mobile Devices Based on Internet of Things Case Study (Pt. Semen Gresik Persero Tbk.)," *2018 International Conference on Applied Engineering (ICAE)*, Batam, Indonesia, 2018, pp. 1-5, doi: 10.1109/INCAE.2018.8579392.
https://ieeexplore.ieee.org/document/8579392

[9] Suzanti, I. O., Fitriani, N., Jauhari, A., & Khozaimi, A. (2020). REST API implementation on Android based monitoring application. *Journal of Physics Conference Series*, *1569*(2), 022088.
https://iopscience.iop.org/article/10.1088/1742-6596/1569/2/022088/meta

[10] Tungcul, M. B., & Kummer, M. G. C. (2021). Supplies and Equipment Inventory, Monitoring and Tracking Management System using Data Mining Techniques.
https://www.ijrte.org/wp-content/papers/v10i2/B61740710221

[11] Kros, J. F., Falasca, M., & Nadler, S. S. (2006). Impact of Just-in-Time Inventory Systems on OEM Suppliers. Industrial Management & Data Systems, 106(2), 224-241.
https://www.emerald.com/insight/content/doi/10.1108/02635570610649871/full/

APPENDICES

Appendix A : Outputs

Figure 20: Landing Page of the Website

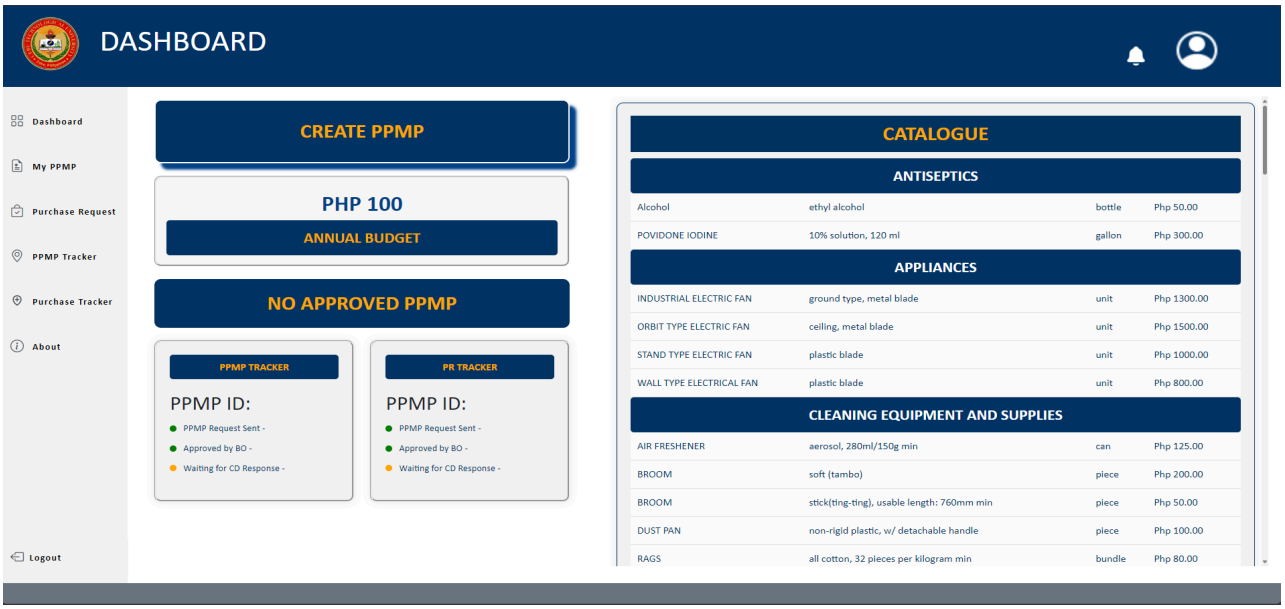


Figure 21: Dashboard Page of the Website

Figure 22: Landing Page of the Mobile Application

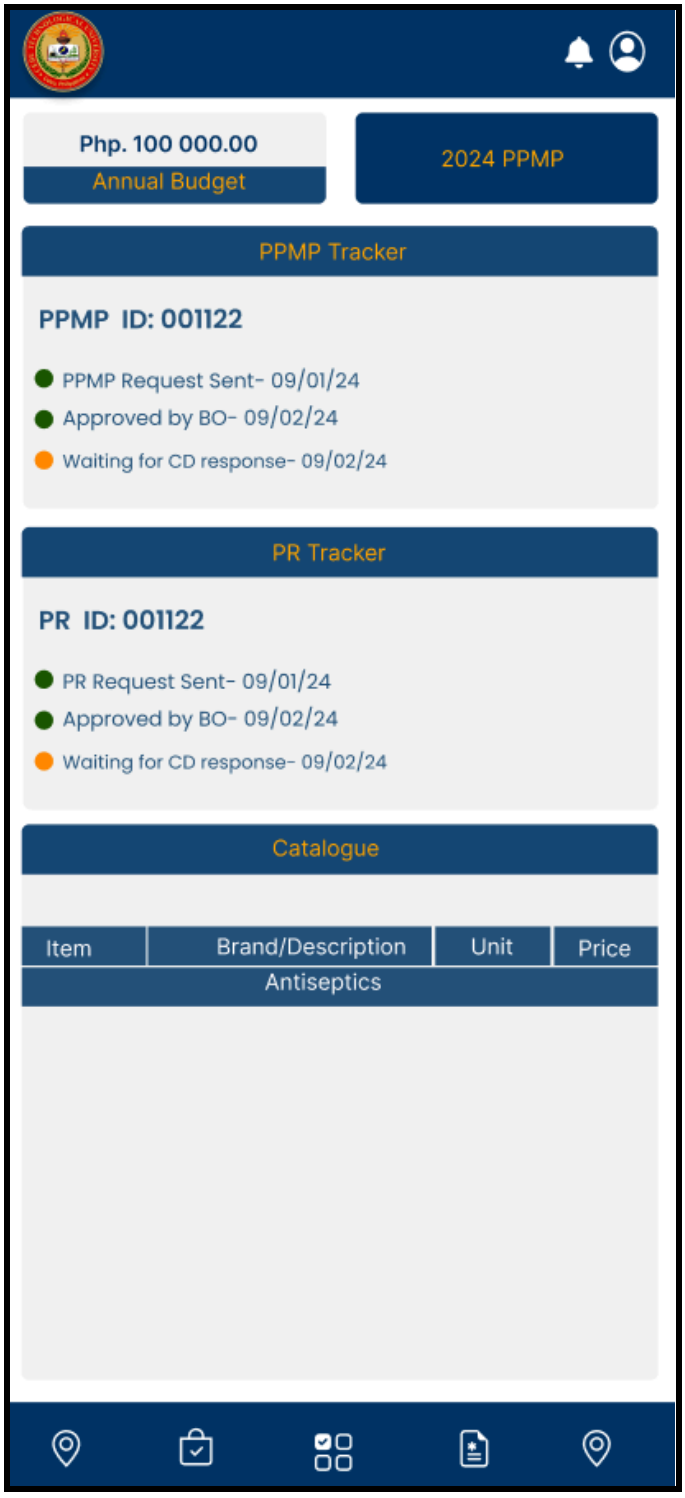


Figure 23: Dashboard of the Mobile Application

Appendix B : Cebu Technological University - Argao Campus Supply Office Request Monitoring System Source Code in Website and Mobile Application

```
from django.contrib.sites.shortcuts import get_current_site
from django.contrib.auth.models import User
from django.http import HttpResponse
from django.shortcuts import render, redirect
from django.utils.http import urlsafe_base64_encode, urlsafe_base64_decode
from django.core.mail import EmailMessage
from django.contrib.auth import get_user_model
from .tokens import account_activation_token
from django.contrib.auth import update_session_auth_hash
from django.core.mail import send_mail
from django.contrib import messages
from django.utils.crypto import get_random_string
from .models import VerificationCode
from django.views.decorators.http import require_POST
from django.shortcuts import render, get_object_or_404, redirect
from django.http import JsonResponse
from django.views.decorators.csrf import csrf_exempt
import random
from itertools import groupby
from django.core.files.base import ContentFile
from .models import *
from django.shortcuts import render, redirect
from django.views import View
from django.contrib.auth.models import User
from django.contrib.auth.tokens import default_token_generator
from django.utils.encoding import force_str
from django.utils.http import urlsafe_base64_decode
from django.contrib import messages
from django.core.mail import send_mail
from django.conf import settings
```

Figure 24: Required Libraries

```
DATABASES = {
   'default': {
      'ENGINE': 'djongo',
      'NAME': 'inventory',
      'ENFORCE_SCHEMA': False,
```

```
        'CLIENT': {
            'host':
'mongodb://duhig:duhig123@192.168.0.160:27017/inventory?authSource=inventory',

            'authMechanism': 'SCRAM-SHA-1'
        }
    }
}
```

Figure 25: Database Connection

```
class User(AbstractUser):
    id = models.BigAutoField(primary_key=True)
    username = models.CharField(max_length=12, unique=True)
    first_name = models.CharField(max_length=12)
    last_name = models.CharField(max_length=12)
    contact1 = models.PositiveIntegerField()
    email = models.EmailField(unique=True)
    budget = models.DecimalField(max_digits=10, decimal_places=2, null=True, blank=True)

    USER_TYPES = [
        ('admin', 'Admin'),
        ('regular', 'Regular User'),
        ('cd', 'Campus Director'),
        ('budget', 'Budget Officer'),
        ('bac', 'BAC'),
    ]

    user_type = models.CharField(max_length=15, choices=USER_TYPES)

    is_admin = models.BooleanField(default=False)
    is_regular = models.BooleanField(default=False)
    is_cd = models.BooleanField(default=False)
    is_budget = models.BooleanField(default=False)
    is_bac = models.BooleanField(default=False)
    is_approved = models.BooleanField(default=False)



    @property
    def get_user_type_display(self):
```

```
        return dict(self.USER_TYPES).get(self.user_type, 'Unknown')

    def save(self, *args, **kwargs):

        self.is_admin = self.user_type == 'admin'
        self.is_regular = self.user_type == 'regular'
        self.is_cd = self.user_type == 'cd'
        self.is_budget = self.user_type == 'budget'
        self.is_bac = self.user_type == 'bac'

        super().save(*args, **kwargs)

    def __str__(self):
        return self.username
```

Figure 26: MongoDB Setup

```
const handleLogin = () => {

            const username = document.getElementById('username').value;

            const password = document.getElementById('password').value;

            console.log('Username:', username, 'Password:', password); // Debugging step

            const formData = new FormData();

            formData.append('username', username);

            formData.append('password', password);


            fetch('http://127.0.0.1:8000/api/token/', {

               method: 'POST',

               body: formData,

            })
```

```
        .then(response => response.json())
        .then(data => {
            console.log('Login response:', data); // Debugging step
            if (data.access && data.refresh) {
                localStorage.setItem('accessToken', data.access);
                const tokenPayload = JSON.parse(atob(data.access.split('.')[1]));
                const userType = tokenPayload.user_type;

                switch (userType) {
                    case 'regular':
                        window.location.href = '/ppmp101';
                        break;
                    case 'admin':
                        window.location.href = '/admin';
                        break;
                    case 'bac':
                        window.location.href = '/baclanding';
                        break;
                    case 'budget':
                        window.location.href = '/bo_landing';
                        break;
                    case 'cd':
                        window.location.href = '/cdlanding';
                        break;
```

```
                    default:
                        alert('Unknown user type. Please contact support.');
                }
            } else {
                alert('Login failed: ' + (data.detail || 'Unknown error'));
            }
        })
        .catch(error => {
            console.error('Error:', error);
        });
    };
```

Figure 27: Application Programming Interface (API) Setup

```
export default function SignIn() {
  const router = useRouter();
  const [username, setUsername] = useState('');
  const [password, setPassword] = useState('');
  const [loading, setLoading] = useState(false);
  const handleSignIn = async () => {
    setLoading(true);
```

```
try {
    const response = await fetch('http://10.0.2.2:8000/login', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
        },
        body: JSON.stringify({ username, password }),
    });
     if (!response.ok) {
        throw new Error(`HTTP status ${response.status}`);
    }
    const result = await response.json();
    console.log('Response:', result);
```

Figure 28: Mobile Initialization

Appendix C : Test Case by the Quality Assurance

Project Name: CTU-AC Supply Office Request Monotoring System
Module Name: Register Page

CTU-AC SUPPLY OFFICE REQUEST MONITORING SYSTEM - Manual Testing Sheet

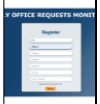
Section	Title	Test Case ID	Test Case Description	Precondition	Steps	Excpected Output	Actual Output	Status	Priority	Defects	Type	Screenshot
Register Page		C20		Enter invalid Email	Navigate to Register Page. Enter Email: a@com. Enter Email: a@a. Enter Email: a@com. Enter Email: 123@com	The entered email is invalid or not in proper format	The email input field accepts invalid email or improper format	Fail	Critical	Email input accepts invalid or improper formats	Functional	
Register Page		C21	There should be a place to add a suffix name, as some users have a suffix that needs to be included	Enter a suffix name	Navigate to Register page	There should be a suffix name.	There is no suffix name	Fail	Critical		Functional	
Register Page		C22		Enter correct inputs on CTU ID, Email, First Name, Last Name, Contact No., Password, Confirm password	Navigate to register page and fill the required inputs correctly	The user will be registered and can freely log in	A prompt stated that a user is now registered and waiting for email confirmation. However, the email message failed	Fail	Critical	Email message confirmation failed	Functional	
Register Page		C23		Testing password security	[Enter password: mypassword] [Enter password: 1234]	The input is invalid and it will only allow password with a mix of uppercase, lowercase, numbers and symbols.	The password input allow the user to input low level of security password.	Fail	Critical	Password requirements	Security	

Figure 29: Register Page

Project Name: CTU-AC Supply Office Request Monotoring System
Module Name: Login Page

CTU-AC SUPPLY OFFICE REQUEST MONITORING SYSTEM - Manual Testing Sheet

Section	Title	Test Case ID	Test Case Description	Precondition	Steps	Excpected Output	Actual Output	Status	Priority	Defects	Type	Screenshot
Log In Page	Regular User	C1		Enter correct CTU id & Enter correct password	Navigate to the log in page. Enter Ctu id: 111111. Enter password: 123	The user will be prompt to the homepage or access will be granted as a regular user credential.	Regular user credential accepted	Pass	Low		Functional	
Log In Page	Regular User	C2		Enter wrong CTU id and Enter correct password	[Enter CTU id: 11111] & [Enter password:123] [Enter CTU id: -1111] & [Enter password:123] [Enter CTU id: 010101 & [Enter password:123] [Enter CTU id: 000000 & [Enter password:123] [Enter CTU id: 666666 & [Enter password:123] [Enter CTU id: 1111111 & [Enter password:123]	Prompt invalid log in credentials	Prompt invalid log in credentials and let the user to try again	Pass	Low		Functional	
Log In Page	Regular User CD BO BAC	C3		When I log in to the CTU ID login page, I will use numbers and characters for the login.	Enter characters in CTU ID login page.	The expected result is that only numbers should be typed, and characters should not be included.	When I typed the CTU ID in Login Page, I tried all the numbers, and they were successful. But, when I tried typing the characters, the characters couldn't be typed, except for one character, which was the letter "e".	Fail	Critical	CTU ID received character	Functional	
Log In Page	Regular User	C4		Enter correct CTU id and Enter wrong password	[Enter CTU id: 111111] & [Enter password:1234] [Enter CTU id: 111111] & [Enter password:0123] [Enter CTU id: 111111 & [Enter password:321] [Enter CTU id: 111111 & [Enter password:246] [Enter CTU id: 111111 & [Enter password:12345] [Enter CTU id: 1111111 &	Prompt invalid log in credentials	Prompt invalid log in credentials and let the user to try again	Pass	Low		Functional	

Figure 31: Login Page

Project Name: CTU-AC Supply Office Request Monotoring System
Module Name: Regular User

CTU-AC SUPPLY OFFICE REQUEST MONITORING SYSTEM - Manual Testing Sheet

Section	Title	Test Case ID	Test Case Description	Precondition	Steps	Excpected Output	Actual Output	Status	Priority	Defects	Type	Screenshot
Reg_Homepage	Regul...	C25	PHP money should definitely have a comma sign.	Enter Regular user account.	Regular user dashboard of the annual budget, there are no comma sign in PHP money.	The PHP money should have a comma sign.	The PHP money has no comma sign.	Fail	Medium		Usability	
Reg_Homepage	Regul...	C26		Enter Regular user account.	Regular user dashboard clicking the button icon.	The button icons should be consistent	The button icons are inconsistent. When I click on the purchase request, the dashboard icon disappear, and when I click on the PPMP tracker, the dashboard icon reappear.	Fail	High	When I click on the purchase request icon, the dashboard icon disappear.	Accessibility	

Figure 32: Regular User

Project Name: CTU-AC Supply Office Request Monotoring System
Module Name: Campus Director

CTU-AC SUPPLY OFFICE REQUEST MONITORING SYSTEM - Manual Testing Sheet

Section	Title	Test Case ID	Test Case Description	Precondition	Steps	Excpected Output	Actual Output	Status	Priority	Defects	Type	Screenshot
CD_Homepage	CD	C27		Testing the homepage of CD by zooming in, zooming out, and scrolling adaptability. Also, check the spacing.	Zoom in and zoom out on all hompage options.	The display text and icons should remain clear and undistorted, ensuring adaptability to different screen sizes and flexibility for scrolling.	The icons and texts are deformed during the zoom in and zoom out action. Scrolling during the zoom in is not feasible.	Fail	High	Text and icons should be adaptable to user visibility preferences and must be scrollable during zooming in or out.	Accessibility	

Figure 33: Campus Director page

Project Name: CTU-AC Supply Office Request Monotoring System
Module Name: Budget Officer

CTU-AC SUPPLY OFFICE REQUEST MONITORING SYSTEM - Manual Testing Sheet

Section	Title	Test Case ID	Test Case Description	Precondition	Steps	Excpected Output	Actual Output	Status	Priority	Defects	Type	Screenshot
BO_Homepage	BO	C28		Testing the homepage of CD by zooming in, zooming out, and scrolling adaptability. Also, check the spacing.	Zoom in and zoom out on all hompage options.	The display text and icons should remain clear and undistorted, ensuring adaptability to different screen sizes and flexibility for scrolling.	The icons and texts are deformed during the zoom in and zoom out action. Scrolling during the zoom in is not feasible.	Fail	High	Text and icons should be adaptable to user visibility preferences and must be scrollable during zooming in or out.	Accessibility	
NUM_INPUT	BO	C31		Enter characters on number input only.	Input characters in the Enter budget field.	Character input should be restricted or must not be an input for a number required input field.	The character "e" passed the Enter budget input.	Fail	High	The letter "e" must be prohibited as number input.	Functional	

Figure 34: Budget Officer Page

Project Name: CTU-AC Supply Office Request Monotoring System
Module Name: BAC User

CTU-AC SUPPLY OFFICE REQUEST MONITORING SYSTEM - Manual Testing Sheet

Section	Title	Test Case ID	Test Case Description	Precondition	Steps	Excpected Output	Actual Output	Status	Priority	Defects	Type	Screenshot
BAC_Hompage	BAC	C29		Testing the homepage of CD by zooming in, zooming out, and scrolling adaptability. Also, check the spacing.	Zoom in and zoom out on all hompage options.	The display text and icons should remain clear and undistorted, ensuring adaptability to different screen sizes and flexibility for scrolling.	The icons and texts are deformed during the zoom in and zoom out action. Scrolling during the zoom in is not feasible.	Fail	High	Text and icons should be adaptable to user visibility preferences and must be scrollable during zooming in or out.	Usability	
BAC_SPELL	BAC	C30		Checking texts spelling	Inspect every text content of BAC users.	All texts content are in correct spelling.	There is incorrect spelling in the "AVIALABLE ITEMS" text..	Fail	Medium	The correct text is "AVAILABLE ITEMS".	Usability	

Figure 35: BAC Secretariat

CURRICULUM VITAE

DANICA MAE MACEDA

danicamaedinawanaomaceda@gmail.com

Mobile No: 09914484596

Present/Permanent Home Address:

Lutak , Guiwanon , Ginatilan Cebu 6028

Birthdate and Birthplace:

February 21 , 2003 - Negros , Tanjay Oriental

Religion:

Roman Catholic

Civil Status:

Single

Education

College

BS Information Technology

Cebu Technological University - Ginatilan Extension

Barangay San Roque, Ginatilan, Cebu

Secondary

Guiwanon National High School

Barangay Guiwanon, Ginatilan Cebu

Graduated July 2021

Elementary

Guiwanon Elementary School

Barangay Guiwanon , Ginatilan Cebu

Graduate March 2014

Technical Skills

- Front-End Developer
- Web Development

Projects Developed

Wifi - Based Bulb Switches
Software Developer

Cebu Technological Argao Campus Supply Office Monitoring System
Frontend Developer

Seminars Attended

Pre-Employment Advocacies
Malabuyoc , Cebu
2024

Cyber Resilience: Understanding the Evolving Landscape of Data Leaks
Via Online (Zoom)
2024

Navigating the Professional World : Work Ethics and Transitioning from Student to Professional
Cebu Technological University - Ginatilan Extension
2023

Character References

Mr. Glicerio A. Baguia
Chair - BSIT & BIT-CT Programs
Cebu Technological University - Ginatilan Extension Campus
Email: glicerio.baguia@ctu.edu.ph

Mr. Rochelle Belvestre
Non-resident faculty
Cebu Technological University - Ginatilan Extension Campus
Email: rochell.belvestre@ctu.edu.ph

RALPH B. TUAZON

ralphtuazon@gmail.com

Mobile No: 09229047016

Present/Permanent Home Address:

Salamanca, Ginatilan Cebu 6028

Birthdate and Birthplace:

August 10, 2001

Ginatilan, Cebu

Religion:

Roman Catholic

Civil Status:

Single

Education

College

BS Information Technology

Cebu Technological University - Ginatilan Extension

Barangay San Roque, Ginatilan, Cebu

2021 - Present

Secondary

Pungtod National High School

Barangay Pungtod, Oslob, Cebu

Graduated March 2019

Elementary

Salamanca Elementary School

Barangay San Roque, Ginatilan Cebu

Graduated March 2014

Technical Skills

- Database Management
- Full-Stack Developer
- Mobile Developer

Projects Developed

Wifi - Based Bulb Switches
Mobile Developer

Cebu Technological Argao Campus Supply Office Monitoring System
Full Stack Developer

Seminars Attended

Pre-Employment Advocacies
Malabuyoc , Cebu
2024

Cyber Resilience: Understanding the Evolving Landscape of Data Leaks
Via Online (Zoom)
2024

Navigating the Professional World : Work Ethics and Transitioning from Student to Professional
Cebu Technological University - Ginatilan Extension
2023

Character References

Mr. Glicerio A. Baguia
Chair - BSIT & BIT-CT Programs
Cebu Technological University - Ginatilan Extension Campus
Email: glicerio.baguia@ctu.edu.ph

Mr. Rochelle Belvestre
Non-resident faculty
Cebu Technological University - Ginatilan Extension Campus
Email: rochell.belvestre@ctu.edu.ph

JANE CARMEL B. DUHIG

janecarmelbarroduhig@gmail.com

Mobile No: 09229047016

Present/Permanent Home Address:

Palanas, Ginatilan Cebu 6028

Birthdate and Birthplace:

September 01, 2002 - Ginatilan, Cebu

Religion:

Roman Catholic

Civil Status:

Single

Education

College

BS Information Technology

Cebu Technological University - Ginatilan Extension

Barangay San Roque, Ginatilan, Cebu

2021 - Present

Secondary

Samboan National High School

Barangay Poblacion , Samboan Cebu

Graduated March 2021

Elementary

Ginatilan Central School

Barangay San Roque, Ginatilan Cebu

Graduated March 2015

Technical Skills

- Front-End Developer
- Web Development

Projects Developed

Wifi - Based Bulb Switches
Software Developer

Cebu Technological Argao Campus Supply Office Monitoring System
Frontend Developer

Seminars Attended

Pre-Employment Advocacies
Malabuyoc , Cebu
2024

Cyber Resilience: Understanding the Evolving Landscape of Data Leaks
Via Online (Zoom)
2024

Navigating the Professional World : Work Ethics and Transitioning from Student to Professional
Cebu Technological University - Ginatilan Extension
2023

Character References

Mr. Glicerio A. Baguia
Chair - BSIT & BIT-CT Programs
Cebu Technological University - Ginatilan Extension Campus
Email: glicerio.baguia@ctu.edu.ph

Mr. Rochelle Belvestre
Non-resident faculty
Cebu Technological University - Ginatilan Extension Campus
Email: rochell.belvestre@ctu.edu.ph

JEYKIE ROSE R. DUHIG

duhigjeykierose@gmail.com

Mobile No: 09267576540

Present/Permanent Home Address:

Mangaco, Ginatilan, Cebu 6028

Birthdate and Birthplace:

January 04, 2002 - Ginatilan, Cebu

Religion:

Roman Catholic

Civil Status:

Single

Education

College

BS Information Technology

Cebu Technological University - Ginatilan Extension

Barangay San Roque, Ginatilan, Cebu

2021- present

Secondary

Holy Trinity College

Barangay San Roque, Ginatilan, Cebu

Graduated March 2021

Elementary

Mangaco Integrated School

Mangaco, Ginatilan, Cebu

Graduated March 2014

Technical Skills :

- Front-End Developer
- Web Development

Project Highlights:

Wifi - Based Bulb Switches

Software Developer

Cebu Technological Argao Campus Supply Office Monitoring System
Frontend Developer

Seminars Attended:

Pre - Employment Advocacies
Malabuyoc, Cebu
2024

Navigating the Professional World
San Roque, Ginatilan, Cebu
2023 - 2024

Cyber Resilience: Understanding the Evolving Landscape of Data Leaks
Philippines via Zoom
2023-2024

Character References:

Mr. Glicerio A. Baguia
OIC, Chair BSIT & BIT Programs Cebu Technological
University - Ginatilan Extension Campus
Email: glicerio.baguia@ctu.edu.ph

Mr. Rochell Belvestre
Non- resident Faculty
Cebu Technological University - Ginatilan Extension Campus
Email: rochell.belvestre@ctu.edu.ph

JHAKE P. DUGADUGA

jhakedugaduga@gmail.com

Mobile No: 0952 629 3482

Present/Permanent Home Address:

Calabawan, Ginatilan, Cebu

Birthdate and Birthplace:

July 01 2002 -Calabawan,Ginatilan,Cebu

Religion:

Roman Catholic

Civil Status:

Single

Education

College

BS Information Technology

Cebu Technological University - Ginatilan Extension

Barangay San Roque, Ginatilan, Cebu

Secondary

Guiwanon,National,HighSchool

Graduated March 2021

Elementary

Cabatuan, Elementary,School

Graduated March 2014

Projects Developed

Wifi - Based Bulb Switches

Software Developer

Cebu Technological Argao Campus Supply Office Monitoring System

Frontend Developer

Technical Skills

- Front-End Developer
- Web Development

Seminars Attended

Pre-Employment Advocacies
Malabuyoc , Cebu
2024

Cyber Resilience: Understanding the Evolving Landscape of Data Leaks
Via Online (Zoom)
2024

Navigating the Professional World : Work Ethics and Transitioning from Student to Professional
Cebu Technological University - Ginatilan Extension
2023

Character References

Mr. Glicerio A. Baguia
Chair - BSIT & BIT-CT Programs
Cebu Technological University - Ginatilan Extension Campus
Email: glicerio.baguia@ctu.edu.ph

Mr. Rochelle Belvestre
Non-resident faculty
Cebu Technological University - Ginatilan Extension Campus
Email: rochell.belvestre@ctu.edu.ph