UCCD2213 SOFTWARE ENGINEERING PRINCIPLES

Assignment Title: Designing a Business Information System for Supply Chain with AI: Inventory Management and Demand Forecasting

No	Student Name	Student ID	Total Mark
1	Richard Ting Li Zeng	2301973	
2	Lee Kean Yang	2203125	
3	Ivan Ng Yong Zhe	2202313	
4	Low Kai Hang	2202440	

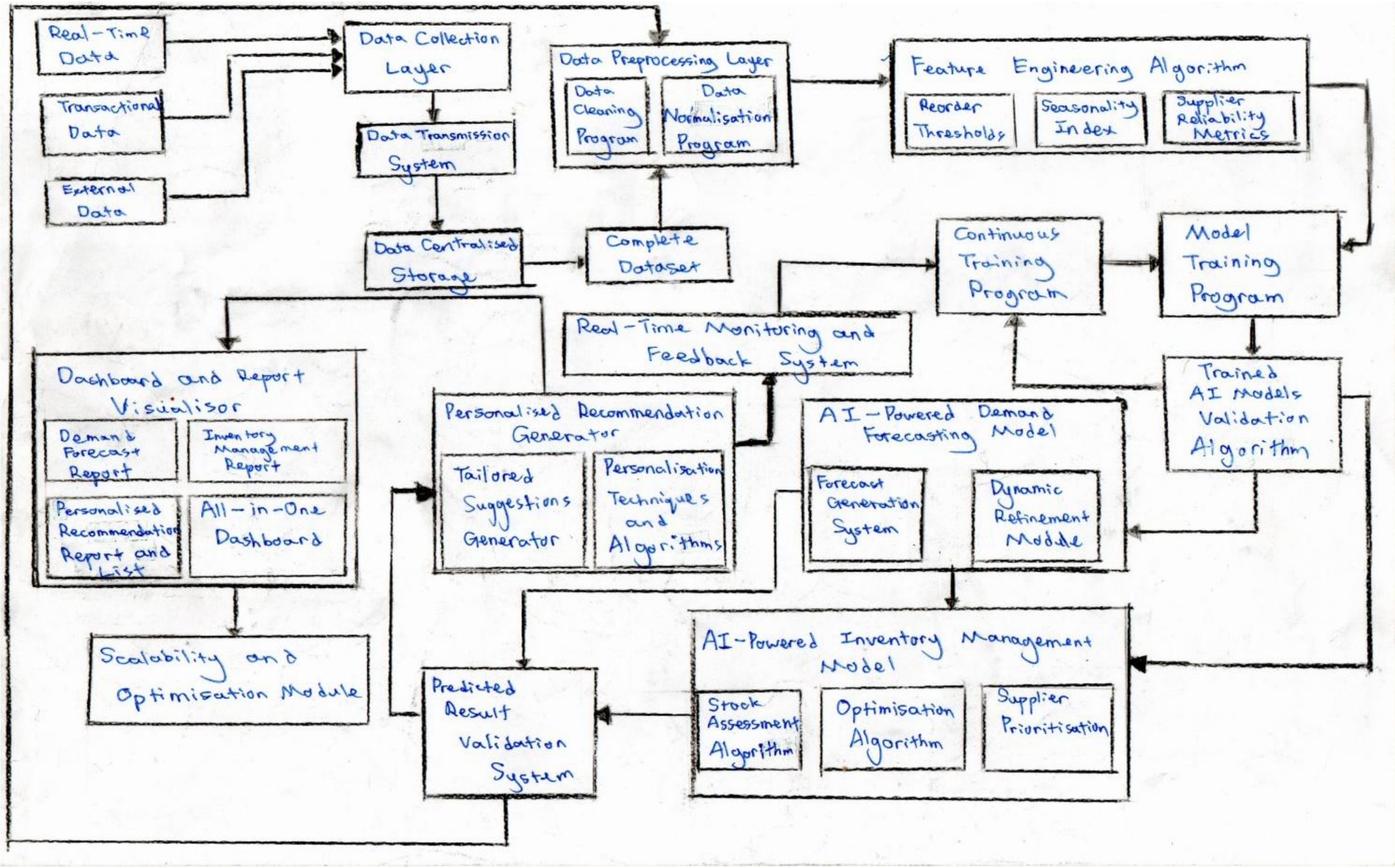
No	Assessment Components	Assessment Marks		Students’ Marks				
		Group	Individual	Group	Individual			
					1	2	3	4
1.0	**PART 1 (20 marks)**							
1.1	Design an architectural design for the system.	**-**	**20**	-				
2.0	**PART 2 (80 marks)**							
2.1	Q1: Requirement Analysis and Design Principles	**20**	**-**		-	-	-	-
2.2	Q2: Pattern Selection and Justification	**20**	**-**		-	-	-	-
2.3	Q3: Component and Architecture Design	**25**	**-**		-	-	-	-
2.4	Q4: Reflection on Design Choices		**15**	-				
	TOTAL	**100**						

INDEX CONTENTS	**PAGE**
COVER PAGE	**01**
INDEX	**02**
1.0 PART I: SYSTEM DEGSIGNING WITH SIMPLE BLOCK DIAGRAM (INDIVIDUAL)	
1.1 Personalized Recommendation Module (Richard Ting Li Zeng)	**04**
1.2 Data Integration Module (Lee Kean Yang)	**07**
1.3 Supplier and Procurement Management Module (Ivan Ng Yong Zhe)	**10**
1.4 Data Privacy and Security Module (Low Kai Hang)	**13**
2.0 PART II: ADVANCED SYSTEM DESIGNING AND DEVELOPMEMT PLANNING (GROUP & INDIVIDUAL)	
2.1 Q1 - Requirement Analysis and Design Principles	**16**
2.2 Q2 – Pattern Selection and Justification	**18**
2.3 Q3-Component and Architecture Design	
2.3.1 Component Diagram	**20**
2.3.2 Class Diagram	**24**
2.3.3 Deployment Diagram	**28**
2.4 Q4 - Reflection on Design Choices	
2.4.1 Reflection on Design Choices (Richard Ting Li Zeng)	**31**
2.4.2 Reflection on Design Choices (Lee Kean Yang)	**33**
2.4.3 Reflection on Design Choices (Ivan Ng Yong Zhe)	**34**
2.4.4 Reflection on Design Choices (Low Kai Hang)	**35**

1.0 Part I: System Designing with Simple Diagram Drawing (Individual Part)

Based on the title, we would like to have a business information system for the supply chain domain with AI-powered models and features. Demand forecasting and inventory management will be the focus of this business information system as these are crucial functionalities for the logistics company to achieve a better supply chain operation. For this report, we will focus on four of the main functionalities or modules for this business information system: a personalized recommendation module, a data integration module, a supplier and procurement management module, and a data privacy and security module.

1.1 Personalized Recommendation Module (Richard Ting Li Zeng)



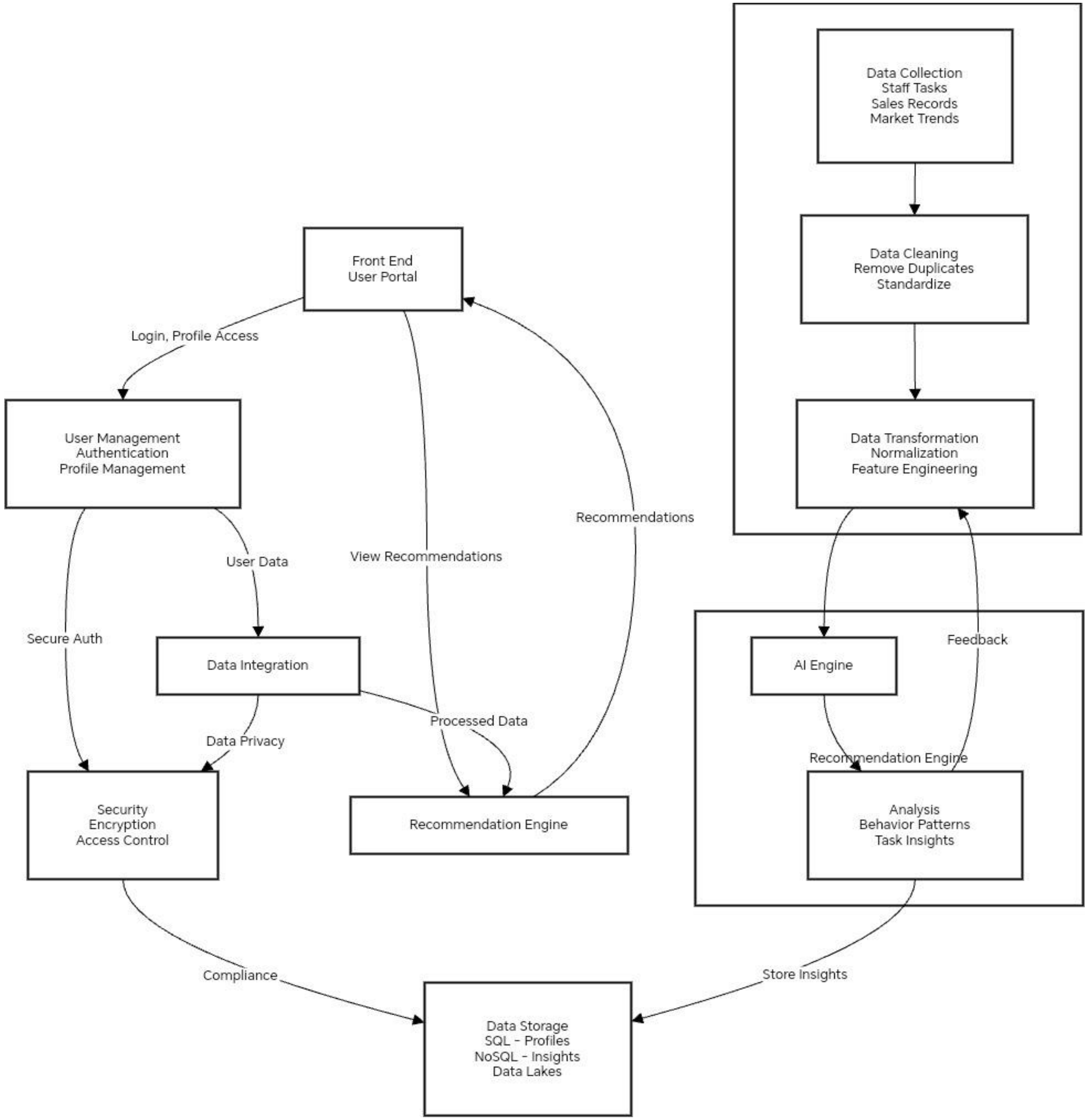
This is the simple block diagram for the personalized recommendation module. It comes with three main features: two models for the focusing terms, demand forecasting, and inventory management, which are powered by artificial intelligence and have been trained by certain datasets and programs, and also a personalized recommendation generator. For this module, we aim to generate recommendations that have been tailored for certain weather conditions based on the predicted result of the AI models, with the real-time demand and inventory information. The recommendation would help the companies to have a better way to provide the services and solve the problem. This is because the companies would be able to avoid the obstacles or the uncertainties that would affect the services provided, especially for transportation arrangements, and order deliveries based on the recommendations given by the system. Companies are able to have a better understanding of the actions and strategies that they should take and use in a certain predicted period for certain conditions and situations.

The process begins with the collection of raw data, sourced from real-time systems, transactions, and external environments. This data flows into the data collection layer, then passes through a data preprocessing layer with undergoing on data cleaning and normalization process. During this stage, key features such as reorder thresholds, seasonality indexes, and supplier reliability metrics are derived through feature engineering. The processed and feature-enriched dataset becomes the foundation for the AI model training. The model training program uses this refined dataset to train the AI-powered demand forecasting and inventory management models. Once trained, these models undergo validation using a dedicated trained AI model validation algorithm to ensure the reliability and accuracy of outputs. The validated models are integrated into the system and continuously improved through a continuous training program based on the feedback from the real-time monitoring and feedback system and the validation errors after the model and predicted result validation process.

After the validation for the model, both AI-powered models, which are the demand forecasting model and the inventory management module, will be used for the demand forecasting and inventory management functionality with some features such as a forecast generation system and dynamic refinement module for demand forecasting model, and some algorithms in the inventory model. The output of both AI-powered models would be validated by the predicted result validation system before the outputs can be used for personalized recommendation generation. The invalid predicted results would be used as the example, and the errors would be sent to the continuous learning program for refinement. With up-to-date and validated outputs from both demand forecasting and inventory management AI models, the personalised recommendation generator applies personalization techniques and algorithms to create highly specific recommendations. These would be visualized in the dashboard and report visualiser as four crucial features: personalised recommendation reports and lists, demand forecasting reports, inventory management reports, and an all-in-one dashboard. The recommendations are also routed to a scalability and optimisation module to ensure the effectiveness of the system.

Ultimately, this module empowers companies with data-driven insights to understand what actions and strategies to implement in specific periods and under specific conditions. It enhances operational efficiency and decision-making, especially in environments affected by fluctuating demand or external disruptions.

1.2 Data Integration Module (Lee Kean Yang)



Referring to the above simple block diagram, the process begins with Data Collection component which gathers raw data. The data serves as the foundation for the entire recommendation module to ensure AI models have access to comprehensive and relevant information. Then, collected data is passed to Data Cleaning stage where processes undergo to remove duplicates and standardize formats. This step is crucial to eliminate inconsistencies and errors that could skew AI models' predictions to ensure data is reliable for further processing.

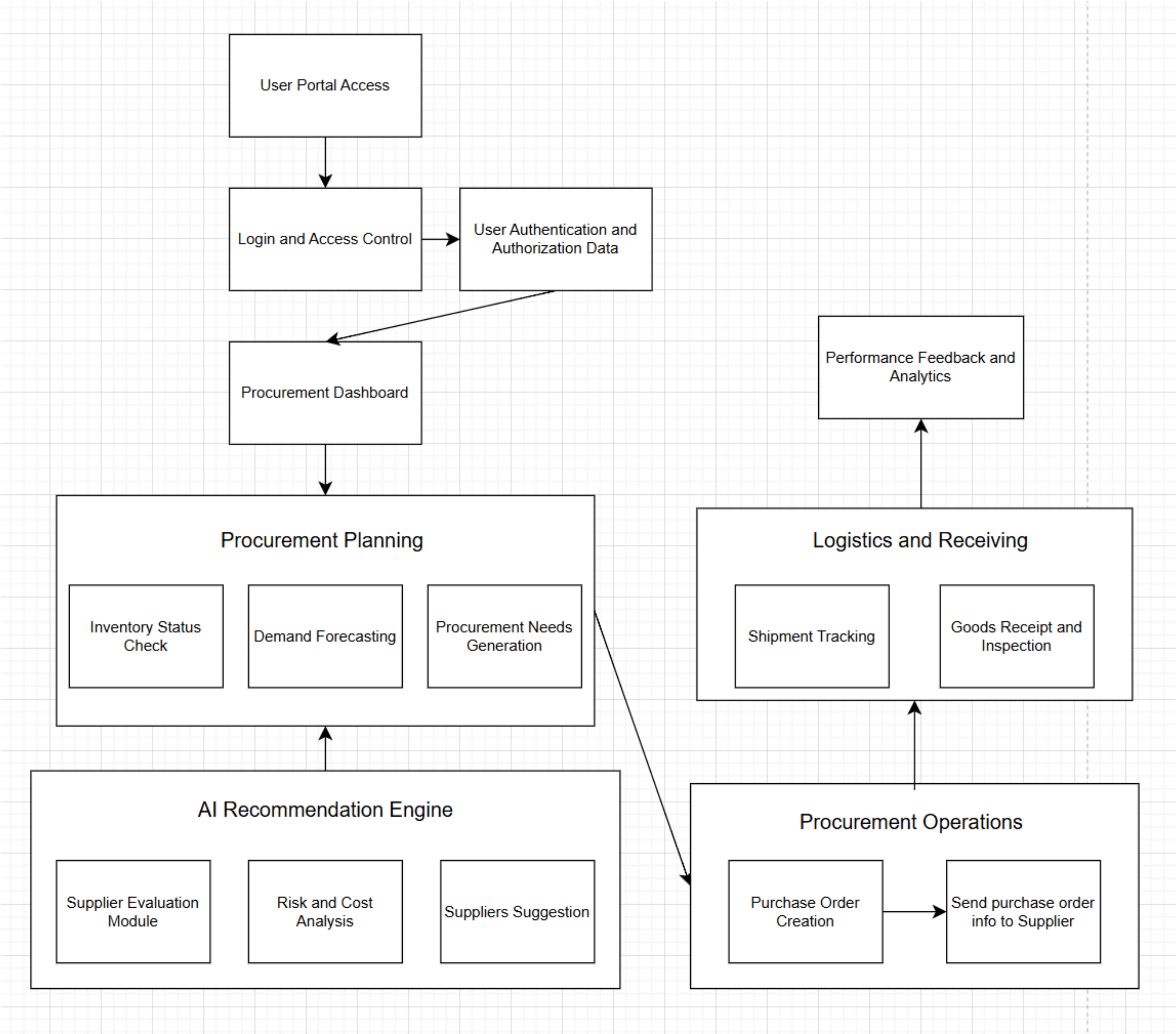
Once data is cleaned and moves to Data Transformation, Normalization, and Feature Engineering component. The system normalizes data to a consistent scale and extracts meaningful features. The features are specifically engineered to enhance AI models' ability to identify patterns and trends relevant to demand forecast and inventory management. Then, transformed data is fed into AI Engine which two AI models focused on demand forecast and inventory management. The models are trained on preprocessed datasets. The advanced algorithms predict future demand and optimize inventory levels under vary conditions. The AI Engine continuously refines predictions by incorporating feedback loops to ensure models adapt to new data and improve over time.

Then, output from AI Engine is utilized by Recommendation Engine which generates personalized recommendations for specific scenarios. As shown in diagram, the engine integrates AI predictions with real-time demand and inventory data to produce actionable insights during adverse weather conditions. Recommendation Engine employs Analysis Behavior Patterns and Task Insights to further customize suggestions, ensure align with company’s operational goals and staff workflows. The recommendations are delivered to Front End where users can view after logging in and accessing profiles via User Profile Management component. The front-end interface allows staff to interact with the system seamlessly, view recommendations help making informed decisions about transportation arrangements and order deliveries, ultimately reducing uncertainties and improving service efficiency.

To ensure the system operates securely and complies with regulations, the Security, Encryption, and Access Control component works in tandem with Data Privacy module. The components encrypt sensitive data, enforce access controls, and ensure compliance with data protection laws as indicated by Compliance flow in diagram. The Data Integration component plays a vital role in aggregating user data and processed data from various sources, ensuring Recommendation Engine has access to unified dataset. Lastly, Data Storage components which includes SQL profiles, NoSQL insights, and data lakes, securely stores all processed data and insights, available for future analysis and model retraining. The Store Insights loop back to Recommendation Engine ensures historical recommendations and outcomes are used to refine future suggestions, create continuous improvement cycle.

The personalized recommendation module not only enhances decision-making for supply chain operations but also drives business growth by enabling companies to proactively face challenges. By providing recommendations based on real-time data and predictive analytics, the system empowers staff to focus on strategic tasks, improve productivity and operational efficiency. The integration of security and privacy measures ensures the system remains trustworthy and compliant while feedback mechanisms evolve with the company’s needs solution for modern supply chain management.

1.3 Supplier and Procurement Management Module (Ivan Ng Yong Zhe)



This is the simple block diagram of the **Supplier and Procurement Management Module** which is designed to streamline and optimize the procurement lifecycle within an organization. The flow of the whole processes starts from **User Portal Access**, **Procurement Planning**, **AI Recommendation Engine**, **Procurement Operations**, **Logistics and Receiving** and lastly **Performance Feedback and Analytics**.

The whole process will begin when a user accesses the system from the **User Portal Access**. First, the user will need to go through the **Login and Access Control** layer to ensure secure interaction with the module, where authentication and authorization will verify the user and define the access rights. After the user has been verified, the user will be directed to the **Procurement Dashboard**, a user interface where user can manage the procurement activities.

In the **Procurement Dashboard**, the user will engage with the **Procurement Planning** which manages the main preparation processes. The processes are **Inventory Status Check**, **Demand Forecasting** which will be powered by the **AI Recommendation Engine** and **Procurement Needs Generation**. These components will work together to identify what resources are needed, when they are needed and in what quantities. Accurate planning is important to avoid stockouts and excess inventory.

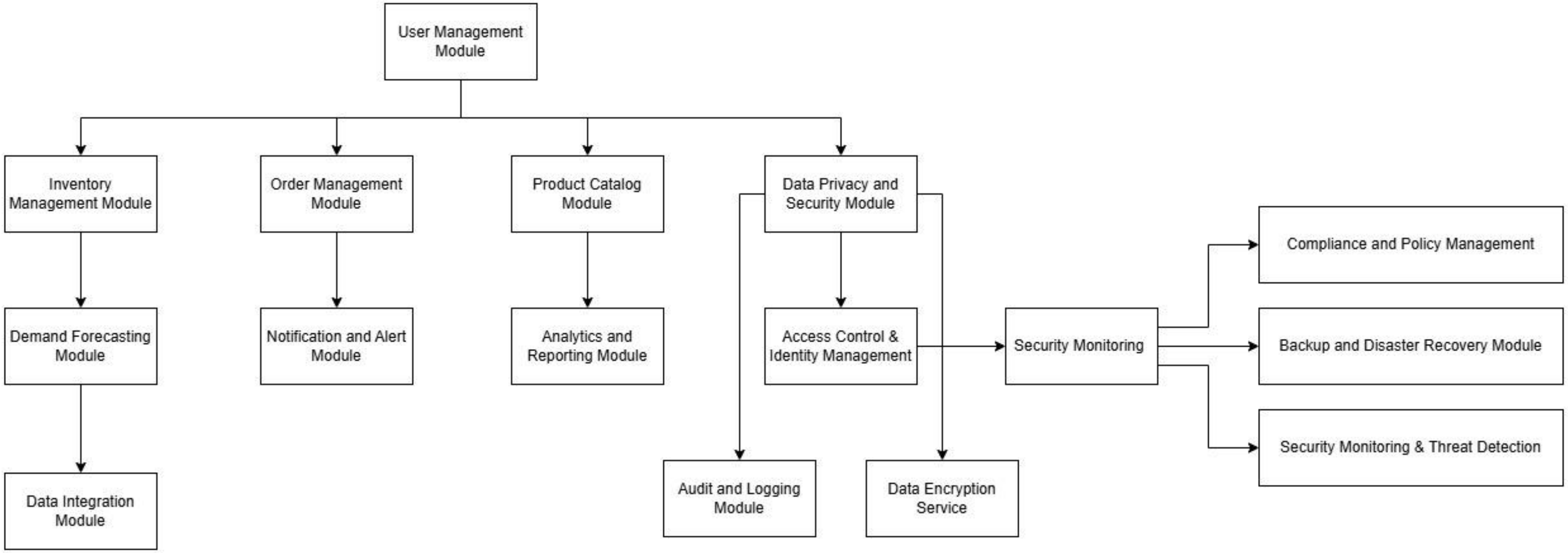
To have better decision-making, the **AI Recommendation Engine** is used to support the planning process. It has three features, **Supplier Evaluation Module**, **Risk and Coast Analysis** and **Suppliers Suggestion**. These tools will analyze historical performance, financial implications and supplier reliability to provide some smart recommendations. This will help the procurement actions to be optimized for quality and cost-effectiveness.

After the planning and recommendations, the process will be continued to **Procurement Operations**. First, the **Purchase Order Creation** takes place and then the relevant order details are communicated to suppliers through the **Send Purchase Order Info to Supplier step**. This process marks the execution of procurement, turning insights and decisions into actual orders. After that, the **Logistics and Receiving** stage will be follow up. In this process, **Shipment Tracking** and **Goods Receipt and Inspection** will be carried out. This ensures transparency in the delivery process and the items will be verified for compliance and quality.

Lastly, the last process is **Performance Feedback and Analytics**. In this process, data across all the previous processes are gathered and will be transformed into valuable insights. Important information like supplier performance, delivery timelines and procurement efficiency are analyzed to inform future strategies and decisions. The feedback will also help improve the AI model and planning methods.

In conclusion, this **Supplier and Procurement Management Module** offers a comprehensive and intelligent framework for managing procurement from the beginning to end. By integrating AI-driven recommendations with traditional planning and execution workflows, it is not only improves accuracy and efficiency but also builds a sustainable, data-informed procurement ecosystem for organizations.

1.4 Data Privacy and Security Module (Low Kai Hang)



This block diagram focuses on delivering intelligent, secure, and scalable business information systems for supply chain management, which aiming to enhance business growth, improve staff productivity and most importantly ensuring data privacy and security. The system provides an intelligent, integrated enviroment where supply chain staff can access real-time inventory updates, receive alerts, automate order processing, and benefit from predictive analytics which are all powered by AI. Additionally, user data and transactional information are all protected through advanced privacy measures and regulatory compliance, with the solution of automating critical procedures and providing timely information, this has enhace the productivity of staff and managers.

The core components are the User Management Module, which controls access to the system by authenticating users' roles, such as Admin, Manager, or Staff, and ensures they can only access functions that are relevant to their responsibilities. After the authentication process, user can track stock levels across warehouse using the Inventory Management Module, real time updates as products are received or dispatched are visualable. Once the stock drop below defined thresholds, it will automatically trigger alerts. Using historical sales data, seasonal trends, and external market factors, the Demand Forecasting Module projects future inventory needs. It provides useful information and enables more intelligent purchase planning with the use of AI algorithms for businesses use. The Data Integration Module provides smooth interface with supplier APIs, ERP systems, and logistical services to enable cooperation with other organizations. It ensures effective data flow and enables businesses to respond quickly to changes in demand and supply.

For the Product Catalog Module, it uses for update product information which include SKUs, categories and pricing. The Analytics and Reporting Module will transform raw data into visual dashboard and reports. Managers can make data-driven choices by seeing sales trends, performance indicators, and prediction accuracy. Order Mangement Module handles both incoming and outgoing orders by using the information from the product database. The use of Notification and Alert Module is the system will notify the employees or clients of order status, low stock, or anomalies once an order is confirmed, and inventory levels are updated.

Given the increasing importance of data security, the system has a complete Data Privacy and Security Module. This module encrypts sensitive data while it's in transit and at rest. It also makes regulatory compliance, notably GDPR, simpler by enforcing data access restrictions and retention requirements. By the Audit and Logging Module, it records all user activities, transactions and system events. This logging system is crucial for security audits, transparency, and abuse monitoring. The Access Control and Identity Management component, which also enforces advanced authentication techniques like multi-factor authentication (MFA), offers role-based access to important services. The Backup and Disaster Recovery Module regularly creates backups and guarantees that essential data may be swiftly retrieved in the event of a breakdown, to maintain system availability and resilience. The Security Monitoring and Threat Detection Module supports this by applying AI to instantly discover cybersecurity concerns or dubious activity.

In conclusion, this system architecture provides a strong answer to today's supply chain problems. Real-time inventory management, AI-powered forecasting, and robust security measures are combined to create a productive workplace for staff members and protect the preservation of important business data. The system's modular architecture and use of design patterns guarantee its scalability, adaptability, and maintainability while enabling it to adjust to evolving business requirements. In the end, intelligent automation and secure information management increase operational efficiency while promoting long-term organizational expansion.

2.0 PART II: Advanced System Designing and Development Planning

2.1 Q1 - Requirement Analysis and Design Principles

The design of the Supply Chain AI System for inventory management and demand forecasting requires analysis of both functional and non-functional requirements to ensure satisfying the requirements of the business while adhering to design principles. The system support user profile management, enable secure authentication and role-based access for staff as in the interaction between Front End and User Profile Management components in previous diagram as well as needs to handle data integration, aggregating real-time and transactional data from internal and external sources as indicated by Real-Time Data, Transactional Data, and External Data inputs feeding into Data Collection Layer. The system provides personalized recommendations which are achieved through Personalised Recommendation Generator. AI models for demand forecast and inventory management produce reports like Personalised Recommendation Report and List. In addition, the system requires data privacy and security, ensures compliance with regulations which is implied through need for secure data handle in Data Centralised Storage and Data Preprocessing Layer. Lastly scalability is one of the requirements as the system supports a growing user base and evolved features, reflected in Scalability and Optimisation Module in diagram.

Furthermore, non-functionally system prioritize performance, ensure AI-Powered Demand Forecast Model and AI-Powered Inventory Management Model can process large datasets efficiently to deliver real-time insights as shown by Real-Time Monitoring and Feedback System. Reliability is critical as the system provides accurate predictions and recommendations which is supported by Trained AI Models Validation Algorithm and Continuous Training Program to refine model accuracy. Usability is another focus with All-in-One Dashboard provide centralized interface for staff to access demand forecasts, inventory reports, and recommendations, enhance productivity. Security is non-negotiable and requires encryption and access controls to protect sensitive data throughout pipeline from Data Collection Layer to Data Centralised Storage. Finally, maintainability ensures system can evolve with Dynamic Replacement Module allowing for updates to algorithms and techniques in AI models without disrupting operations.

Design principles such as modularity, scalability, maintainability, and security guide architectural solutions to the requirements effectively. Modularity is evident in layered architecture of system as shown in diagram. Data Collection Layer, Data Preprocessing Layer, Model Training, and Personalised Recommendation Generator are different components with specific responsibilities to allow for independent development, testing, and updates. For instance, Data Preprocessing Layer handles data cleaning, normalization, and feature engineering, ensure AI models receive high-quality inputs without being tightly coupled to data collection process. This separation enhances flexibility as new data sources can be integrated by updating the Data Collection Layer without affecting downstream components.

Scalability is through the Scalability and Optimisation Module which ensures the system can handle increased data volumes and user demands. The diagram indicates AI-Powered Inventory Management Model includes Stock Assessment Algorithm and Optimisation Algorithm which can be scaled horizontally by adding more computational resources to process larger datasets or support more users. Real-Time Monitoring and Feedback System further supports scalability by enabling systems to adapt dynamically to changing conditions like sudden spikes in demand, ensure consistent performance as business grows. Maintainability is facilitated by Dynamic Replacement Module and Continuous Training Program which allow for seamless updates to AI models and algorithms. For example, if a more advanced demand forecast algorithm becomes available, Trained AI Models Validation Algorithm can validate, and Dynamic Replacement Module can integrate without downtime to ensure the system remains updated and easy to maintain.

Security is a foundational principle across systems to protect sensitive data and ensure compliance with regulations. While not shown in the current diagram, Data Centralised Storage implies need for encryption and access controls as sensitive data like sales records and supplier metrics are stored. Data Preprocessing Layer plays a vital role by anonymizing data during feature engineering to reduce risk of privacy breaches when training AI models. In addition, Real-Time Monitoring and Feedback System can include audit logging to track data access, ensure compliance and enhance trust. By embedding the design principles, architectural solutions ensure adaptable, and secure effectively satisfy both functional and non-functional requirements while supporting business growth and staff productivity in supply chain.

2.2 Q2 – Pattern Selection and Justification

To address the Supply Chain AI System's requirements, we have selected four design patterns that we think are appropriate to the system. These four design patterns are Observer from behavioural pattern, Mediator from behavioural pattern, Factory Abstract Method from creational pattern and Singleton from creational pattern. Each of these patterns contributes to solve design challenges in the system.

First, the Observer Pattern is used to manage real-time updates between the AI-powered demand forecasting and inventory management models and the modules that relying on their outputs, such as the personalized recommendation and visualization dashboard. When new predictions are made by the AI models or the AI models are refined by the continuous learning process, all the modules are informed and updated automatically. This ensures data consistency, reduces tight coupling between modules, and allows for easy synchronization of real-time intelligence across the system.

Next, the Mediator Pattern helps simplify the communication between multiple interdependent modules, such as the data preprocessing unit, model training program, validation algorithm and recommendation generator. Instead of the modules directly referencing each other, they communicate through a central mediator. This reduces the complexity of interactions and prevents tightly coupled logic, which improves maintainability and scalability. For example, the personalized recommendation generator does not need to know the internal details of both the AI models or the validation system; the mediator coordinates these interactions in a streamlined manner.

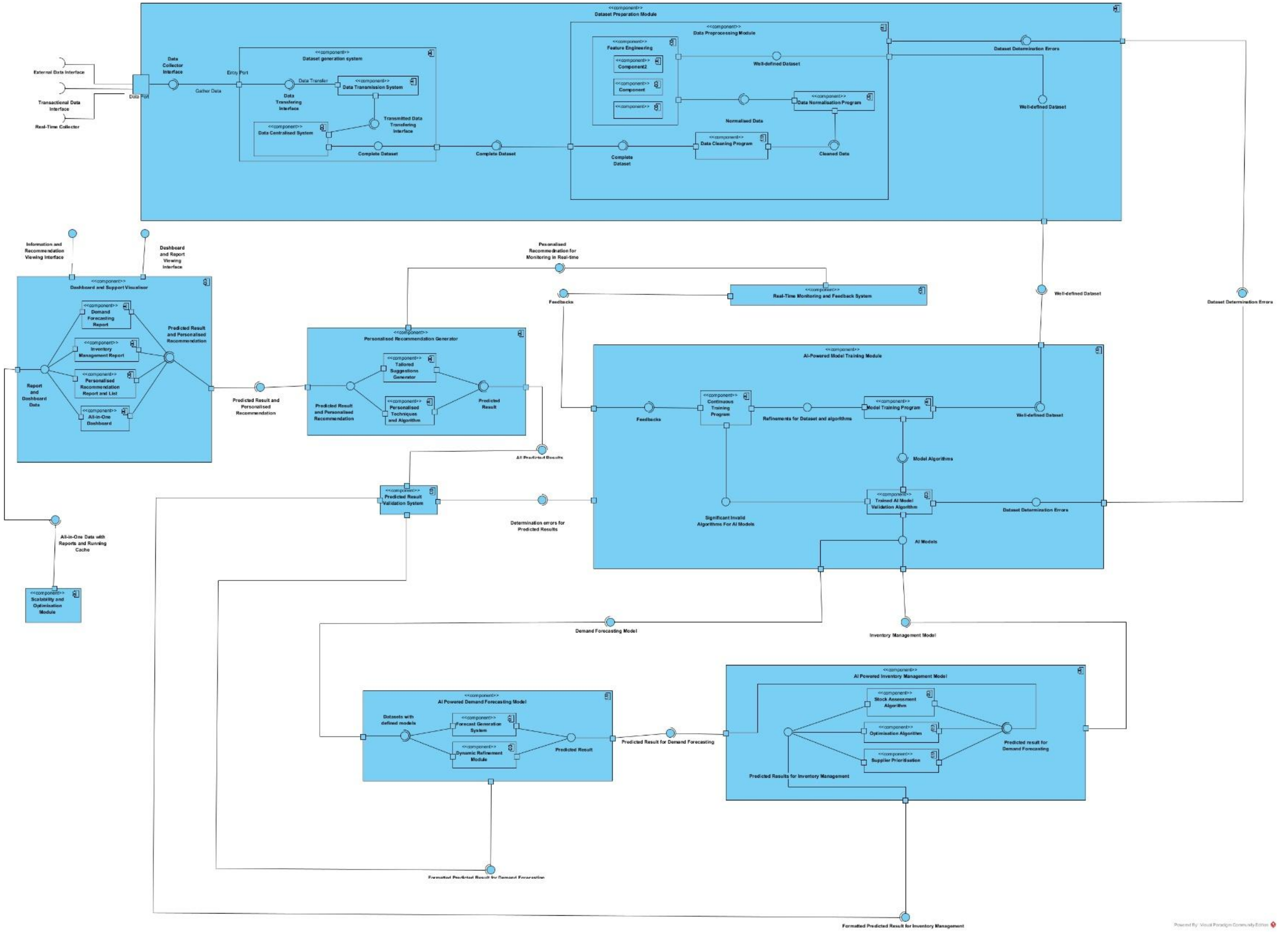
The Factory Abstract Method Pattern is applied to support the creation of different variants of AI models and recommendation strategies depending on operational needs. For example, the system might require different forecasting models for weather-sensitive products or peak-demand seasons. Using the Factory Abstract Method allows new model types to be introduced without altering the core logic and promoting flexibility. The pattern also supports the instantiation of modular components such as inventory optimization algorithms or report generators with minimal changes.

The Singleton Pattern is used for important components that must have a single, shared instance across the system, such as the model validation system and the real-time monitoring and feedback module. These components must maintain a consistent state and coordinate actions across the system, such as verifying model outputs or gathering performance feedback. It can help to ensure only one instance exists and prevents conflicts, conserves resources and enhances control over the whole system behavior.

In conclusion, the four design patterns which are Observer pattern, Mediator pattern, Factory Abstract Method pattern and Singleton pattern help to address the system’s design challenges like modularity, scalability, real-time responsiveness and system consistency. By using these patterns, the system is become more maintainable, scalable and adaptable for the future. This structured approach ensures efficient coordination between components, promotes clean architecture, and supports the overall reliability, effectiveness and performance of the Supply Chain AI System.

2.3 Q3-Component and Architecture Design

2.3.1 Component Diagram



This flowchart illustrates a sophisticated and well-considered strategy for data management, forecasting, and supplier monitoring. By effectively handling the many phases of data gathering, processing, predictive models, and inventory control, the system employs a complete approach that aids firms in making better decisions.

The procedure begins with data collecting and rudimentary processing. This entails gathering data from several internal and external sources. Verifying that the system can connect to several huge data sources is a crucial initial step. Following data collection, the initial processing of the data takes place. This entails ensuring that the raw data is accurate, cleaning it up, and organising it logically for usage. To ensure that the data is accurate and prepared for the following stage, the system verifies its consistency. This stage ensures that the estimations and models that follow are founded on accurate and trustworthy data.

The feature development phase of data preparation begins once the data has been cleaned up and verified. Preparing the data for machine learning models is the aim of this stage. The process of identifying the most crucial features in a dataset and eliminating them to improve the model's performance is known as feature engineering. The system is better at identifying patterns by concentrating on the most significant portions of the data, which is crucial for creating precise prediction models. At this stage, you can modify current data points or add new characteristics to ensure the updated data satisfies the model's requirements.

Predictive analytics and forecasting models are essential components of the AI-driven forecasting system. Machine learning techniques are useful in this situation because they can forecast future events based on historical data. The machine uses a variety of methods to analyse data and provide predictions. Important company metrics like inventory levels, sales performance, and product demand may be linked to these records. Predictive analytics is a technology that helps organisations make better decisions by predicting future events. As fresh data becomes available, the training tool adds it to maintain these models stronger. The AI-powered system improves its ability to forecast future events over time, producing better outcomes in the real world.

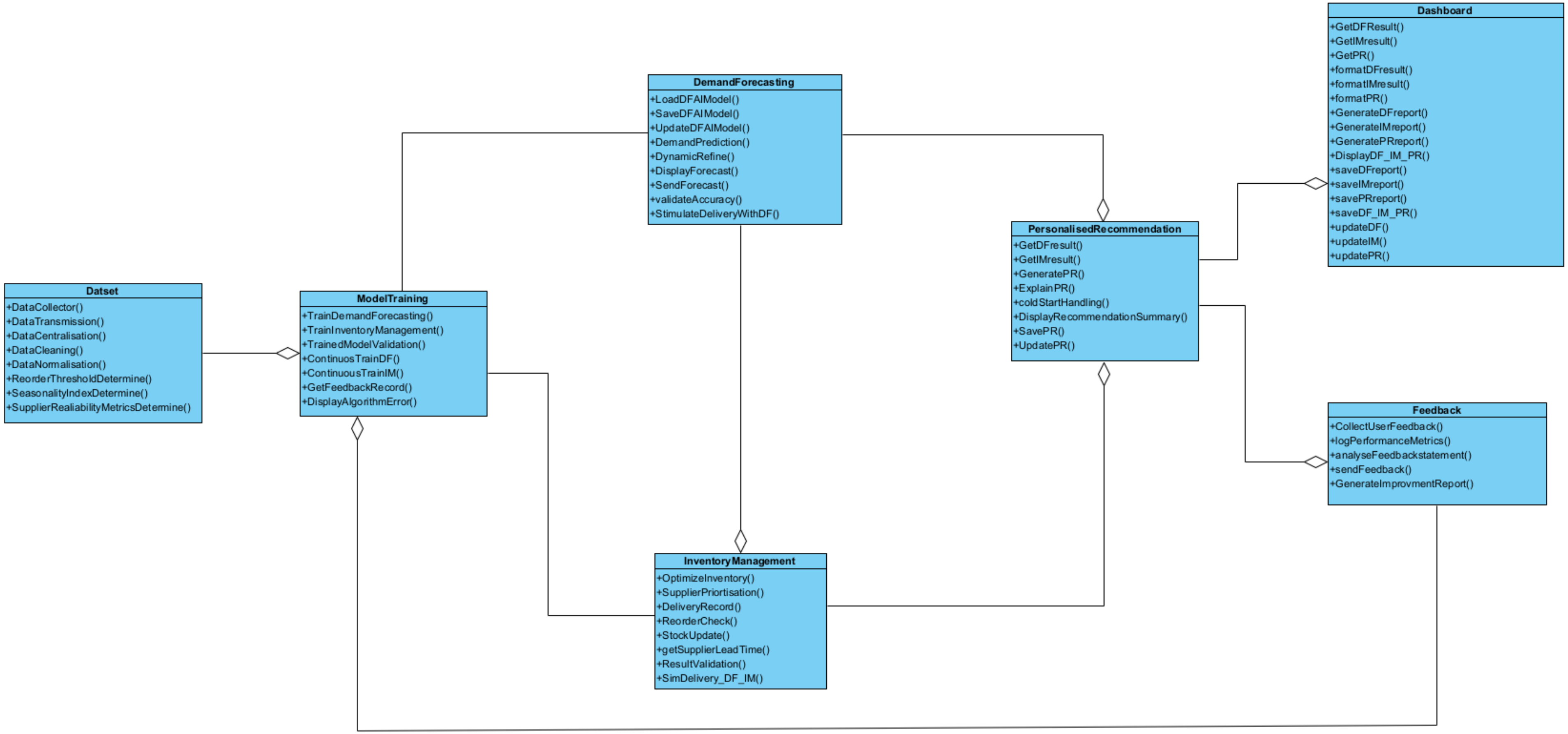
The importance of monitoring supplies and items inside the system is demonstrated by the inventory and stock management graph. The "Inventory Management Model" increases the quantity of items by using AI-generated forecasts. This entails forecasting the amount of material required, monitoring the flow of commodities, and ensuring that there are neither surpluses nor gaps. By integrating predictive data with inventory management, businesses may be able to save expenses, increase productivity, reduce losses, and maintain the proper quantity of items on hand. The retail system may begin replenishing depending on consumer preferences because the purchase process is automated. Because it reduces the amount of time workers must spend performing tasks by hand, this technology improves the supply chain.

Because they continuously verify and enhance the model's predictions and outputs, feedback loops are a crucial component of the system. This approach will continue to improve the system. We should examine the discrepancies between what really occurred and our predictions in order to improve the model's accuracy. Clients may view real-time dashboards and performance figures that display information such as inventory levels, sales trends, and the general effectiveness of the system since the system can manage data. These reports assist decision-makers in monitoring key performance indicators (KPIs) and determining the impact of the system on company operations.

The final phase involves compiling and reporting reliable data. After the procedure is complete, the system generates comprehensive reports that may contain data on the effectiveness of material usage, forecast accuracy, and other crucial business metrics. These products are designed with end users in mind and provide crucial information to decision-makers. To ensure that everyone in the organisation receives the most recent information, the data might be incorporated to various business tools. Predicted insights from AI models are used in this collaboration to improve strategy planning and empower the business to make more informed decisions.

Having said that, this graph illustrates a comprehensive system that makes use of artificial intelligence (AI) and sophisticated data processing to improve corporate operations, particularly in the areas of planning and supply monitoring. Businesses may maximise the value of their products, save expenses, and improve decision-making by making more accurate forecasts thanks to the system's precise data collecting, processing, and analysis capabilities. Businesses may be flexible and fast to adjust to changes in the market thanks to the technology, which continuously improves its models based on input and incorporates the findings into everyday operations.

2.3.2 Class Diagram



This picture offers a thorough examination of a system intended to oversee and improve operations via feedback loops that show continuous development. The system can estimate demand, manage supply, and provide customized suggestions, among other things. The system's several parts work together to produce accurate and useful results. Each part has a specific function. The dataset, model training, demand forecasting, personalised recommendation, dashboard, feedback, and inventory management classes depicted in the figure work together to provide a system that efficiently provides insightful information and improves business procedures.

A key component of the system is the Dataset class, which is responsible for handling the raw data used in later stages. Data collection, transmission, storage, cleaning, correct formatting, and the identification of important indications like source dependability and reorder thresholds are all covered in this session. Accuracy, consistency, and processing readiness of the data given to the system are guaranteed by the methods ‘DataCollector()’, ‘DataTransmission()’, and ‘DataCleaning()’. Because it includes vital information that guarantees the system operates efficiently, this class is necessary.

The purpose of the ModelTraining class is to train various models that support activities like supply management and demand forecasting. This class contains methods that let you build and test machine learning models, such ‘TrainDemandForecasting()’, ‘TrainInventoryManagement()’, and ‘ContinuousTrainDF()’. With the help of its continuous training feature, the system can adapt and perform better as it gets fresh data. In order to guarantee that training models operate efficiently and stay up to date, this session offers techniques for detecting flaws and pinpointing their locations.

The DemandForecasting class uses its knowledge to forecast future demand after the model has been trained. ‘DemandPrediction()’, ‘DynamicRefine()’, and ‘StimulateDeliveryWithDF()’ are some of the methods in this class that use real-time data to anticipate demand, modify dynamic forecasts, and improve supply chain choices in the future. By keeping the predictive models up to date and maintained, this class makes sure that forecasts are based on the most recent data.

Demand projections and other data are used by the PersonalizedRecommendation class to offer customized recommendations for supply chain and product choices. This class is crucial for making sure the system can handle situations like a "cold start," when it needs to make recommendations with little knowledge, and yet meet each customer's unique demands. ‘DisplayRecommendationSummary()’ and ‘GeneratePR()’ both help with the creation and provision of customized recommendations. ‘UpdatePR()’ guarantees that the recommendations stay pertinent.

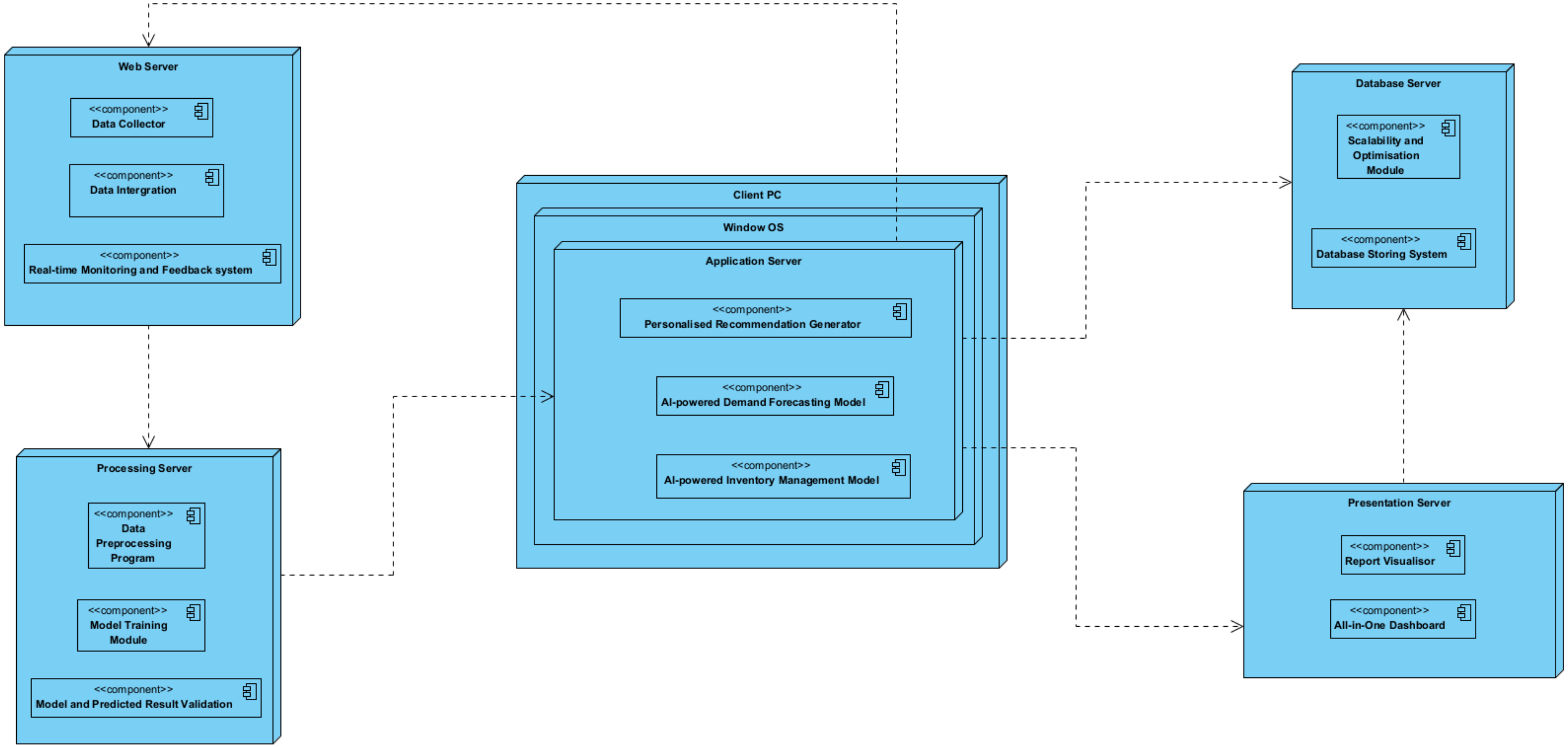
The results of the system's operations are shown via the Dashboard class. It produces reports with demand projections and personalised suggestions, enabling all parties to comprehend the circumstances and make wise decisions. Complete reports and comprehensive highlights from functions like ‘DisplayDF_IM_PR()’ and ‘GenerateDFReport()’ are available to users. Because it gives them real-time information on the system's performance and projections, the screen is essential for decision-makers.

Loops of feedback are essential to this mechanism. The purpose of the Feedback class is to collect user feedback and assess the system's efficacy. This course teaches how to monitor performance statistics, provide reports on improvements, and get user input. With the use of functions like ‘CollectUserFeedback()’ and ‘GenerateImprovementReport()’, the system may adjust and improve itself in response to user feedback and performance assessments. The system's constant alignment with user requirements and corporate goals is guaranteed by this iterative process.

The InventoryManagement class is in charge of setting priorities for sources, managing stock changes precisely, and making sure that inventory levels are just right. You may successfully manage your inventory by using functions like ‘OptimizeInventory()’, ‘SupplierPrioritization()’, and ‘StockUpdate()’, which make sure that supply levels match demand. This closes gaps and cuts down on waste. The supply chain may be improved by anticipating demand, managing inventories, and making customized suggestions.

The system tracks inventory, forecasts demand and provides customized suggestions in a dependable and adaptable manner. The system's performance and efficiency may be improved by grouping it into adaptable classes, each of which would be in charge of a particular step in the procedure. Through a variety of techniques, including reporting, providing recommendations, forecasting, training models, and efficiently handling data, these classes can interact with one another. The system can adjust and perhaps get better over time thanks to the constant feedback technique. For companies trying to improve their operations and give their clients better service, this is a great resource.

2.3.3 Deployment Diagram



The model shown is a complex system design meant for an AI-driven, data-centric solution, possibly in the field of analytics for inventory management, e-commerce, or a similar area. The system is made up of many computers, or servers. Each server has different parts that work together to collect, handle, examine, and show data. These parts are meant to let decisions be made in real time, provide personalised services, and make sure that operations run as smoothly as possible.

The Web Server is the primary point of entry into the system and manages interactions with the external environment. There are three major components that make up the Web server. For the system to function, the Data Collector compiles information from several internal and external sources. Before proceeding to the following stages of processing, the data integration section examines and aggregates this data to ensure consistency. You may immediately provide comments and points of view and keep track of happenings using the Real-time Monitoring and Comments System. This is particularly useful when things are changing rapidly and a decision needs to be taken.

Using AI models to add value is the focus of three key components of the Application Server. By employing machine learning algorithms to recommend products or services that are more likely to pique customers' interest, the Personalised Recommendation Generator enhances their user experiences. Businesses may plan their product quantities and satisfy client demands by using the AI-driven Demand Forecasting Model, which forecasts what customers will desire in the future. In a similar vein, the AI-driven Goods Management Model lowers operating expenses while assisting companies in maintaining the proper stock levels to prevent both running out and having too much.

The database server maintains the system's data and ensures its management. As the volume of data increases, the Scalability and Optimisation Module ensures that the database may expand swiftly. As a result, system performance remains consistent even as demand increases. All required data, including user-provided raw data and processed data utilised in reports and model projections, are tracked by the database storage system. The entire system is based on this data, which enables data analysis and informed decision-making.

The purpose of the Presentation Server is to facilitate end users' viewing and comprehension of information. The Report Visualiser simplifies the process of examining data and drawing conclusions by converting reports into graphs. Users may stay informed and make informed decisions based on real-time data by using the All-in-One Dashboard, which displays critical system operations and performance metrics simultaneously.

The end user interacts with the system through the client PC, which is running Windows OS. To access services like monitoring supply, forecasting demand, and providing tailored recommendations, it communicates with the Application Server. People currently utilise the system to make decisions based on the data and models produced by the core design.

The way the data moves through the architecture determines how well the system functions. Indirect linkages provide information and data across computers and systems so that they can perform their functions, as indicated by the dashed lines joining the components. Every component and server has a distinct function, yet for the system to function well, they must all cooperate. The Web server gathers the data, the Application server applies the taught models, and the Database server stores the data. Finally, this information is shown to consumers in an understandable manner using the Presentation Server.

In conclusion, building a complex, data-driven system is made easier with this architecture. The technology helps organizations function more effectively, make better choices, and offer more individualized services by integrating real-time data collecting, machine learning, AI-driven forecasting, and effective data display. All of the components and computers are essential to ensuring that the system functions properly and can expand to accommodate corporate demands. This is a must if want to increase consumer satisfaction and business efficiency in areas like e-commerce or inventory management.

2.4 Q4 - Reflection on Design Choices

2.4.1 Reflection on Design Choices (Richard Ting Li Zeng)

As a designer for personalized recommendation module, applying a pattern-based technique, which is the simple block diagram, is easier to have the visualization for the module no matter to stakeholders and developers like us. We are able to see that actually the module includes the core functionalities, demand forecasting and inventory management, which have been mentioned in the title, and both of them have been developed with artificial intelligence algorithms to have the automations of the system. This characteristic has become one of the reasons that I have made this module as my first and prior choice for this part. Another reason worth noting is that the simple block diagram is designed with the completion of the module, which means that the module has been designed with complete functionality. This module includes the AI models with demand forecasting and inventory management, and also the personalised recommendation generators for generating the tailored recommendation based on the predicted result from AI-powered demand forecasting and inventory management models with the real-time situations, such as the weather, stock level, delivery transportation availability, and so forth.

The application of the pattern-based technique is able to bring out the clarification for the module or the system, which we have mentioned above, and this situation is able to let the stakeholders and developers know the concept of the design and development of the system. In addition, with the visualization of the simple block diagram, we are able to look on the potential of the module and the importance that it would be able to bring out for the crucial decision making on facing the challenges with correct solutions and strategy. Hence, we are able to find that the simple block diagram is not just able to have a visualization for the module, especially for module design and development, and also it can give the direction for the designers to have better ways and directions for improvements without any misleading by other factors so that the system would be more complete with all of the main functionalities since the simple block diagram is the fundamental and simplest diagram for showing the process and flow of a module or system. Moreover, the simple block diagram can also be used repeatedly since it is the foundation for all of the related systems with the required functionalities. For example, the simple block diagram for the personalised recommendations must have the module or model for the analysis of requirements and recommendations generator with the related algorithm, just need to add on with different usage, such as different AI models for different problems or challenges that have been faced.

In this assignment, I have learned about the usage of design patterns. The design patterns not only improve the program and code structure but also enhance system scalability, maintainability, and modularity. The problems of the complex integration and communication challenges within our personalised recommendations module are able to be solved for incorporating multiple design patterns. Based on our lecture note for the patterns design learning, we have learned about seven patterns, and I would have four of them, Observer patterns, Mediator patterns, Factory Abstract patterns, and Singleton patterns for this assignment since we have used this four for this assignment.

First and foremost, I have learned from the Observer Pattern on decoupling systems that need to stay synchronized. The synchronisation of the module is important, especially for real-time updates across interconnected modules like demand forecasting, inventory management, and personalized recommendations. This would make it easier to scale or modify components independently without coupling. In addition, the Mediator Pattern helped reduce the complexity of communication among highly interdependent modules. We are able to minimise direct dependencies and maintain a clean, modular architecture that enhances future maintainability by centralizing the interactions through a mediator.

Moreover, the Factory Abstract Method Pattern highlighted that the extensibility of the module can be achieved with the separation based on object creation logic from implementation. This pattern allowed us to dynamically create different AI model types and recommendation strategies based on specific use cases. All of the creation would be used with high adaptability since it is easy to adapt in different situations, and the core system logic would not be changed at the same time. Last but not least, the Singleton Pattern is able to emphasize the need for consistent shared instances in components like model validation and feedback systems. It helped ensure a single, global access point for managing critical resources. This is able to avoid conflicts and promote reliable coordination across modules. Overall, these design patterns provided valuable insights into building robust software architecture for complex systems.

2.4.2 Reflection on Design Choices (Lee Kean Yang)

Applying pattern-based techniques to architectural design of Supply Chain AI System significantly influenced my decision-making process, structured approach to system's requirements while ensuring flexibility, scalability, and maintainability. I incorporated patterns like Singleton Pattern for user profile management to ensure single point of authentication and Adapter Pattern for data integration to handle heterogeneous data sources. These patterns provided blueprints for tackling specific challenges like ensuring secure user access, integrating diverse data, and enabling systems to evolve with new AI models.

Reflecting on the process, one lesson I learned was the importance of balance complexity and simplicity when applying design patterns. Initially, I was tempted to overuse patterns to automatically improve the design. However, I realized each pattern introduces additional complexity, such as increased code overhead or the need for careful configuration. This taught me to critically evaluate whether a pattern is truly a specific problem or if simpler solution might suffice, ensure design remains practical and efficient. Another lesson was the value of patterns in promoting modularity and separation of concerns. The Adapter Pattern used in Data Integration component allowed me to decouple data source integration from core system logic, easier to add new data sources without affecting AI Engine or Recommendation Engine. This modularity reflected in layered architecture of the diagram made system easier to test, debug, and scale which was significant for meeting scalability requirement.

Another significant takeaway was the vital role of patterns in future-proofing systems. However, I learned patterns are not one-size-fits-all solution. For example, while Singleton Pattern was appropriate for authentication it might not have been suitable for other components where multiple instances could improve performance like in Model Training process. This taught me to carefully consider and trade-offs of each pattern to ensure my design choices align with the system's specific goals and constraints.

To sum up, using design patterns in this assignment deepened my understanding of ways to structure complex systems systematically. The provided language to communicate design intent clearly as in Personalised Recommendation Generator integrates with All-in-One Dashboard to deliver actionable insights. However, the process underscored the need for pragmatism patterns to serve the system's needs, not dictate design. I would focus on documenting rationale for each pattern more thoroughly to ensure future maintainers understand design decisions, especially in collaborative setting where group members might need to build on the work. This reflection has reinforced my appreciation for pattern-based design as a powerful tool for creating scalable systems while the importance of applying to achieve balance between functionality, performance, and simplicity.

2.4.3 Reflection on Design Choices (Ivan Ng Yong Zhe)

The use of pattern-based techniques greatly influenced my architectural decisions in designing the Supply Chain AI System. These design patterns helped ensure that the system achieves modularity, scalability and maintainability. This is important because of its complexity and real-time data dependencies. The architecture pattern that is used is Client-Server Architecture pattern, which it separates the user interface and visualization components from backend processes. For example, users interact with front-end like the Dashboard and Report Visualiser, while back-end such as data preprocessing, feature engineering, and AI model execution are handled independently. This separation supports scalability and makes it easier to maintain or upgrade specific modules without affecting others.

Besides that, the Observer Pattern is applied in the Real-Time Monitoring and Feedback System, which tracks the performance and outcomes from the AI-Powered Demand Forecasting Model and AI-Powered Inventory Management Model. When changes in model accuracy or prediction trends occur, the observer notifies the Continuous Training Program to trigger retraining or adjustments. This dynamic update mechanism enables continuous improvement without manual reconfiguration and ensures the models remain accurate and responsive to new data.

Other than that, The Mediator Pattern is used for managing the interactions between the modules such as the Personalised Recommendation Generator, Predicted Result Validation System, and AI components. Rather than having these components directly depending on each other, the mediator manages the communication, reducing tight coupling. This makes it easier to update individual modules, such as replacing the recommendation logic or updating the validation without disrupting other parts of the system. The Singleton Pattern is also used for the Data Centralised Storage. Since multiple modules such as the data preprocessing layer, AI training modules, and dashboard are all relying on a shared dataset. It is important to ensure that only a single instance of the storage exists. This will help to maintain consistency across the system and avoid issues like data duplication.

By using the design patterns in this assignment, I learned that how these patterns can greatly improve the structure and clarity of a system's architecture. By applying patterns like Client-Server, Observer, Mediator and Singleton, I was able to create a system that is easier to understand and flexible for future changes. These patterns show how different components will interact, making it easier to identify each components' responsibilities to reduce coupling between modules. These patterns also help future proof, such as when need add new features or adapting new requirements. There will be no need to rewrite existing components. Overall, design patterns taught me the importance of planning for scalability and adaptability which leads to more maintainable and robust system.

2.4.4 Reflection on Design Choices (Low Kai Hang)

Building the personalised suggestion module really highlighted the value of design patterns, demonstrating their usefulness not only in programming but also in establishing the overall framework of a system. I aimed to prioritise organisation, flow, and maintainability before focussing on constructing things in a specific order. One of the aims was to create a system that is intelligent, adaptable, and proficient in managing real-time data, all while providing valuable outcomes for end users.

A significant element was the Layered Architecture Pattern. This process breaks down the system into various stages, including data collection, preparation, model training, and generating personalised suggestions. This really clarified how the tasks were divided and showed me ways to enhance or modify the system moving forward, all without needing to begin anew.

The Observer Pattern significantly influenced my design of the Real-Time Monitoring and Feedback System. This way of thinking, although not currently in use, highlights the significance of certain components in "monitoring" system performance and suggesting improvements, particularly in relation to AI.

The personalised recommendation generator is designed for the strategy pattern. Since recommendations might vary based on factors such as stock availability and weather conditions, it was wise to design the system with adaptable and interchangeable formulas. This approach allows for various methods to be tested or enhanced without jeopardising the entire system design.

One key takeaway I've discovered is that design patterns simplify the process of creating something functional and scalable, even during the planning or sketching phase. I wasn't just focused on connecting the boxes; I was really considering the roles, how data flows, and what will be needed for future maintenance.

After applying these design principles, I found myself engaging with the system as a builder rather than merely a student of diagrammatic representation. The design became more robust, instilling confidence that the structure could adjust to growth and practical requirements if future changes are necessary.