# DOCUMENTATION

# Software process models

## Waterfall process model

* The waterfall software process model can be considered suitable for developing a shoe management system. Here are some reasons why the waterfall model may be ideal for this case:

1. Stable and Well-Defined Requirements-If the requirements for the shoe management system are well-understood and unlikely to change significantly during the development process, the waterfall model can be effective. Since the model follows a sequential approach, it requires clear and stable requirements upfront to proceed with subsequent phases.
2. Linear and Predictable Timeline-The waterfall model follows a linear progression from one phase to another. This makes the timeline and deliverables for developing the shoe management system more predictable.
3. Emphasis on Documentation-The waterfall model emphasizes comprehensive documentation at each stage of the development process. This documentation can be valuable for regulatory compliance, future maintenance, and knowledge transfer.
4. Sequential and Structured Approach-The waterfall model provides a structured approach with distinct phases, ensuring that each phase is completed before moving to the next. This can be advantageous when developing the shoe management system as it allows for careful planning and execution of each stage.
5. Well-Defined Project Scope-The scope of the shoe management system project is well-defined and there is a clear understanding of the desired functionalities and deliverables hence the waterfall model is suitable.

* Developing a shoe management system using the waterfall software process model will involve following a sequential, linear approach with distinct phases. Here's a detailed explanation of how to go through each phase:

1. Requirements Gathering- This involves understanding the specific needs of the shoe business, such as inventory management, order processing, reporting, and integration with external systems.
2. System Design-Once the requirements are gathered, we proceed with the system design phase. Here, we would create a high-level design that outlines the overall architecture and components of the shoe management system. This includes designing the database schema, defining the user interface, and planning the integration with external services. The design should align with the identified requirements and ensure scalability, performance, and security.
3. Implementation-After completing the system design, the implementation phase begins. This involves coding and developing the shoe management system based on the design specifications. We would start by implementing the core functionalities, such as product management, inventory tracking, order processing, and customer management. Programming languages, frameworks, and technologies that best suit requirements will be used.
4. Testing and Quality Assurance-Once the implementation phase is complete, the shoe management system needs to undergo rigorous testing. This includes functional testing, where we verify that each component and feature of the system performs as expected. Conduct integration testing to ensure that different modules work together seamlessly. Additionally, perform performance testing, security testing, and user acceptance testing to validate the system's quality, reliability, and usability.
5. Deployment-After successful testing and quality assurance, we are ready to deploy the shoe management system. This involves setting up the necessary infrastructure, configuring servers, and ensuring compatibility with the target environment. Prepare the system for production use, including database migration, data population, and configuration of external integrations.
6. Operation and Maintenance-Once the shoe management system is deployed, it enters the operation and maintenance phase. Regularly monitor the system's performance, security, and stability. Provide ongoing support to address any issues or bugs that may arise. Periodic maintenance activities, such as software updates, database optimizations, and bug fixes will be performed.

# Cart\_Craze shoe Inventory Management System Case Study

* The case study revolves around the implementation of Cart\_Craze’s shoe Inventory management system for a successful shoe business.
* It aims to highlight the reasons for developing the Inventory Management System, its intended use, overall goals, key features, and advantages compared to other systems.
* Cart\_Craze Inventory Management System was developed to address the challenges faced by the shoe business. The business struggled with manual inventory management, leading to inefficiencies, stock outs, and missed sales opportunities.
* This Inventory Management System was created to automate and streamline inventory processes, providing accurate stock information, and optimizing the overall shopping experience for customers.

## Overall Goals of the System

1. Automate inventory management: Cart\_Craze Inventory Management System aims to replace time-consuming manual inventory tracking with an automated system, ensuring real-time stock visibility and accuracy.
2. Improve customer satisfaction: By having accurate stock information readily available, this Inventory Management System enables its staff to provide accurate product availability to customers, enhancing their shopping experience.
3. Optimize stocking decisions: Cart\_Craze’s Inventory Management System data-driven insights assists in making informed decisions regarding restocking, identifying popular products, and forecasting demand to minimize stock outs and overstock situations.
4. Increase operational efficiency: Cart\_Craze Inventory Management System streamlines inventory-related tasks, freeing up staff time to focus on customer service, visual merchandising, and other value-added activities.

## Key Features

1. Real-time stock tracking: The system provides a live inventory dashboard that displays the current stock levels for each shoe model, including sizes, colours, and quantities.
2. Automated alerts and notifications: The system sends real-time notifications to the staff when stock levels reach predefined thresholds, ensuring timely reordering or restocking.
3. Sales integration: The system integrates with its point-of-sale system, allowing automatic inventory updates whenever a purchase is made, reducing the risk of selling out-of-stock items.
4. Demand forecasting: It employs advanced analytics to analyse historical sales data, seasonality trends, and customer preferences, helping the business predict future demand accurately.
5. Reporting and analytics: The system generates comprehensive reports on sales performance, inventory turnover, and popular shoe models, enabling the business to make data-driven business decisions.

## Advantages of Cart\_Craze Inventory Management System Compared to Other Systems

1. Tailored for shoe businesses: It is specifically designed to meet the unique inventory management needs of shoe businesses, considering factors such as multiple sizes, colours, and styles.
2. Seamless integration: It seamlessly integrates with Cart\_Craze’s existing point-of-sale system, minimizing disruptions to their operations and ensuring accurate inventory synchronization.
3. Real-time updates: It provides instant stock updates, ensuring that Cart\_Craze’s staff and customers have up-to-date information on product availability.
4. Data-driven insights: Its analytics and reporting features empower Cart\_Craze with valuable insights into sales patterns, popular shoe models, and inventory performance, helping them make informed decisions.

## Laws and regulations to be considered when developing the system

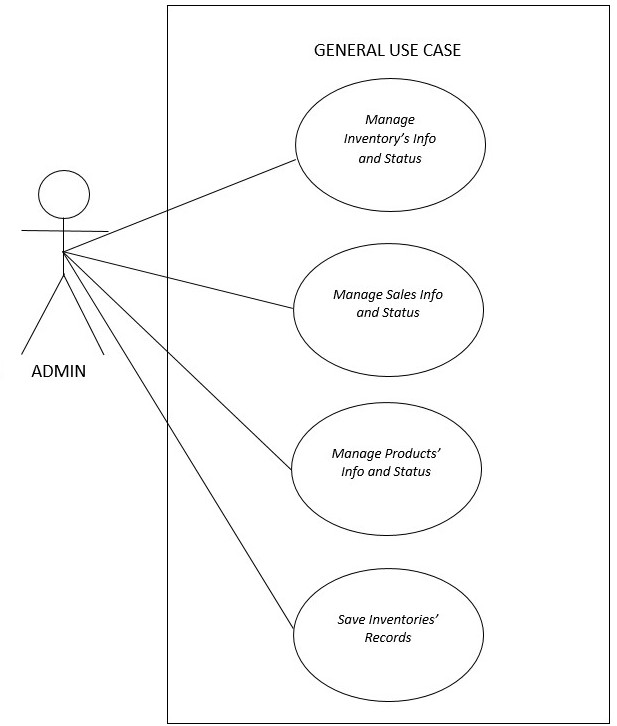
1. E-Commerce Laws-Consumer Protection Laws: Ensure compliance with consumer protection laws, which vary by jurisdiction, to protect consumers from unfair practices, ensure transparent pricing, and provide clear terms and conditions for purchases.
2. Intellectual Property Laws-Trademark and Copyright Laws: Respect trademarks and copyrights of brands and designers when showcasing their products on your website, and implement measures to prevent intellectual property infringement by users.
3. Payment Card Industry Data Security Standard (PCI DSS)-If you accept credit card payments on your website, you must comply with the PCI DSS, which sets standards for securely handling, processing, and transmitting credit card information.
4. Advertising and Marketing Laws-Truth in Advertising: Ensure that your shoe business's advertising and marketing materials are truthful, not misleading, and comply with advertising standards and regulations.
5. International Trade and Customs Laws-Understand and comply with international trade laws and customs regulations when shipping shoes across borders, including import/export restrictions, duties, and taxes.

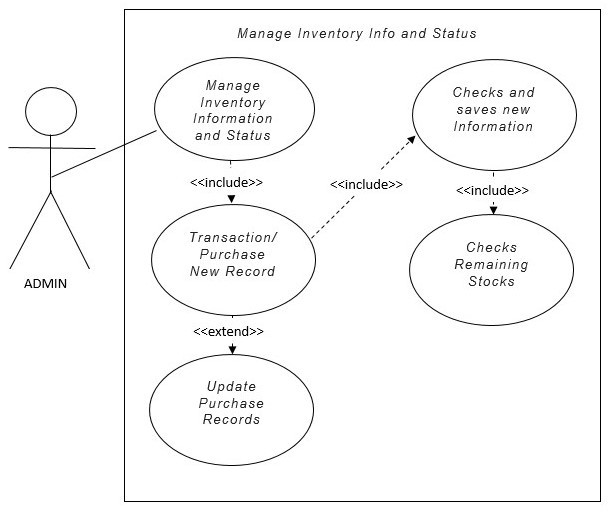
## Users of the System

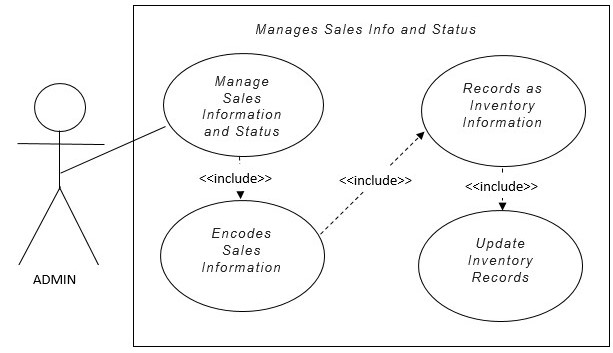
They include:

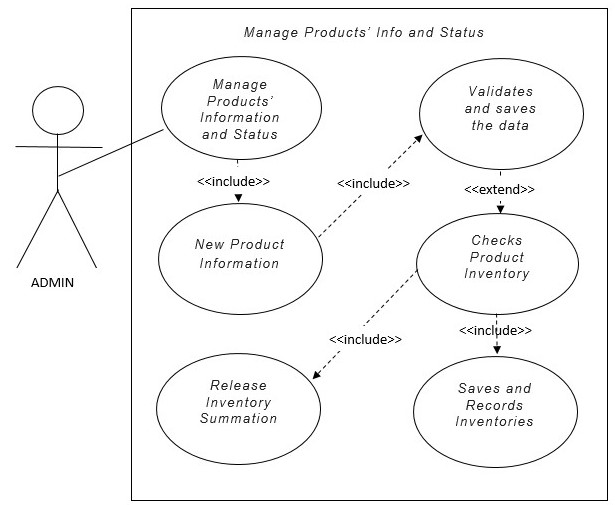
1. Administrators: System administrators have privileged access and are responsible for managing and maintaining the system. They handle tasks such as user management, system configuration, and security management.
2. Sales and Customer Support Staff: These users utilize the system to process orders, manage customer inquiries, track shipments, and provide support to customers.
3. Inventory Managers: These users are responsible for managing and updating the product inventory within the system, including stock levels, adding new products, and handling inventory-related tasks.

# USE CASE DIAGRAMS









# System modelling

* System modelling is the process of creating abstract representations of a system to better understand its structure, behaviour, and interactions.
* It involves using various modelling techniques and tools to visually depict different aspects of the system and its components.
* System modelling plays a crucial role in system analysis, design, and development by providing a clear and systematic way to represent and communicate complex system concepts.

## Types of system modelling techniques used.

1. Context Models-Context models help in understanding the external entities and their interactions with the system. For the shoe management system, a context model would identify the external entities that interact with the system, such as customers, inventory suppliers, payment gateways, and shipping providers. It would illustrate the relationships and interactions between these entities and the system. This model helps in visualizing the system's boundary and its connections to the external environment.
2. Interaction Models-Interaction models focus on the flow of information and actions within the system. In this case, an interaction model would capture the sequence of interactions between different components and actors within the system. It would depict how users interact with the system's user interface, how their actions trigger processes such as product search, order placement, and payment processing, and how the system responds to those actions. Interaction models can use techniques such as use case diagrams, sequence diagrams, and activity diagrams to visualize these interactions.
3. Structural Models-Structural models represent the static structure and organization of the system's components and their relationships. In this case, structural models would illustrate the various components and their dependencies, such as the user interface, database, inventory management module, order processing module, and external service integrations. This can be represented using component diagrams, class diagrams, or entity-relationship diagrams.
4. Behavioural Models-Behavioural models capture the dynamic behaviour of the system, including how it responds to different events and inputs. In this case, behavioural models would show the system's response to user actions, such as the process flow for adding items to the cart, updating inventory levels, or generating order confirmations. Behavioural models help in understanding the system's functionality, process flows, and event-driven behaviour.
5. Model Driven Engineering-Model-driven engineering (MDE) is an approach where models play a central role in the development process. MDE involves using models to specify, design, and generate the system's code and artefacts. In the case of the shoe management system, MDE can be applied by creating high-level models that capture the system's requirements, design, and behaviour. These models can then be used to automatically generate the underlying code, database schema, user interfaces, and other system artefacts. MDE promotes consistency, reusability, and maintainability throughout the development process.