# Software Engineering Analysis and Design



Group members:

Klei Jahaj

Xhesjano Halla

Angjelos Goga

Department: Computer Engineering

Course: Software Engineering Analysis and Design

Project: Inventory system

Accepted by: Dr. Igli Hakrama

Table of Contents

1 SUMMARY OF THE PROJECT1

1.1 Project Overview2

1.2 Purpose and scope of this specification 3

2 PRODUCT AND PRODUCT DESCRIPTION3

2.1 User characteristics3

2.2 Assumption5

2.3 Constraints………………………………………………………………………………………………………………………………6

2.4 Dependencies………………………………………………………………………………………………………..6

**3.1 FUNCTIONAL REQUIREMENTS……….……………………..………8**

**3.2 NON-FUNCTIONAL REQUIREMENTS……………………………..11**

3.2.1 User interface requirements………………………………………………………………………………..11

3.2.2 Usability………………………………………………………………………………………………………….13

3.2.3 Performance……………………………………………………………………………………………………..16

3.2.4 Availability………………………………………………………………………………………………………18

3.2.5 Latency……………………………………………………………………………………………………………20

**3.3 MANAGEABILITY/MAINTAINING...……………………………...21**

3.3.1 Monitoring………………………………………………………………………………………………….……21

3.3.2 Maintenance….………………………………………………………………………………………………..22

3.3.3 Operations……………………………………………………………………………………………………….22

**3.4 ORGANIZATIONAL REQUIREMENTS…………………………….23**

3.4.1 Environmental requirements………………………………………………………………………………23

3.4.2 Operational requirements………………………………………………………………………………….24

**3.5 EXTERNAL REQUIREMENTS…………………………………………25**

3.5.1 Ethical requirements………………………………………………………………………………………….25

3.5.2 Legislative requirements…………………………………………………………………………………….26

3.5.3 Security requirements………………………………………………………………………………………..27

**3.6 DOMAIN REQUIREMENTS……………………………………………29**

3.7 BPMN

3.8 Use Case

3.9 Activity Diagram

3.10 State Diagram

1. **Summary of the project**

**1.1 Project overview**

To keep track of all the stocks of the school throughout the whole supply chain that the university has can be rather challenging and tiring if done by hand. Noting everything in a piece of paper can also bring up many mistakes that can lead to many situations that the university does not want to be in. In order to keep track of every good and possession that the university has it is very useful to have a system that sorts everything in classes, offices of the professors and in the storage rooms.

The Inventory System web application is the best solution to sort all these problems. The information will be stored in a protected server which would be very flexible in the way of modifying it.

* 1. **Purpose and scope of this specifications**

The main focus of our program will be **STOCKS**. Everything will be related to the stocks of the university.

Stocks will be divided in **Classes, Offices, Storages** and **Special Classes** (laboratories, etc.)

This program will help to keep track of the stock of the university via the web-application which is easier to manage rather than doing everything by hand. This web-application will save time, will make the inventory system more efficient and the departments will be informed of the stocks the university has and what is in need of.

**2. Product and product description**

**2.1 User characteristics**

This program will be used by these users:

* + HRS
  + Finance
  + IT
  + Professors
  + Stock providers

1. HRS

Another important administrator is the HRS which will decide which stock will be removed, where the stock will be placed and in which storeroom it will be held.

This administrator will manage the inventory system, also will keep track of inventory status. The HRS department will also take care of the user privileges. They can also accept or decline offers from stock providers.

1. Finance

The Finance department can get a report from the web-application of what is missing and they can add what they will provide for the university and whether the objects will be held in the Storage or be used in one of the above mentioned places. This user will also see the status of a product and the web application can predict what the university might need in the future.

1. IT

The IT department will get requests from different faculties and will be aware if any stock is needed that is related to the IT.

This administrator will decide where the stock will be placed and it will also be updated in the inventory web-application system. If the IT department wants to add a device or is in need of a device, it will make a request which will go to the HRS department.

1. Professors

The person that will report in the application about offices will be the professor that utilizes the office. The reports on the special classes will be made by the professor that is in charge of the special class

(for instance for the Physics lab the Physics professor will make the report). Classes have no one in charge, but the professor that detects the problem can submit the report via the web-application to the HRS or the IT department based on the problem.

1. Stock provider

Stock providers will be able to see what does the university need and they can submit an offer via the web-application and they will get notified if their offer is accepted or rejected.

**2.2 Assumptions**

* It is assumed that the data generated and registered will be fully confidential and it will be available only for the specific departments of the university.
* It is assumed that every user has the appropriate equipment (computer or mobile device, internet access) to use the program.
* It is assumed that the users of this program have general knowledge on how to use the internet and their respective device effectively.
* It is assumed that every change that happens should be updated ion the program for better organization and better usage of this program.
* It is assumed that every user of the program has his/her university email activated and working properly.
* It is assumed that the stock is already registered by the service department.
* It is assumed that the users will update the inventory for every object that is missing, damaged or broken.
* It is assumed that we have all the barcodes for each product registered.

**2.3 Constraints**

The system may be constrained by:

* Having every user to understand how this system works and making sure they do

not do any mistakes.

* Problems with access management or security.
* The system does not allow the user to make customizations regarding the system name components as it make cause anomalies.
* Other constraints can be found during the way.

**2.4 Dependencies**

List of dependencies that affect the requirements:

* The user should always have internet access.
* The user should have an electronic device.
* The need of at least moderate internet speed.
* There should be a user from HRS department to make changes such as adding or deleting a user, adding or deleting stocks etc.
* The response time is depended on the volume of requests and the number of users that are currently using the system.

**3.1.1 Functional Requirements**

**1. Item Management**: The system should allow users to add, modify, and delete items in the inventory database. This includes the ability to enter item descriptions, unit of measure (such as pieces, etc.), and stock levels. The item management feature should also allow users to categorize items into groups, such as product type or supplier.

**2. Stock Management:** The inventory management system should track the current stock levels of each item and provide notifications for low stock levels. This helps ensure that there is enough inventory to fulfill the demands and to prevent stockouts. The system should also allow users to set reorder points for each item, which triggers a notification when the stock level drops below a certain threshold.

**3. Order Management:** The system should provide a way to manage and track purchase orders. This includes the ability to create, approve, and fulfill purchase orders. The order management feature should also allow users to view order history and track the status of open orders.

**4. Reporting:** The inventory management system should provide a variety of reports that allow users to view information about their inventory, sales, and purchasing activities. This may include reports on stock levels, sales history, purchase history, and supplier performance. The system should also allow users to customize reports and export data to other formats, such as Excel or PDF.

**5. Barcode and Scanning:** The system should provide support for barcode scanning and QR code scanning. This makes it easy for users to scan items for accurate and efficient inventory management. The barcode and scanning feature should also allow users to print barcodes for items that do not already have them.

**6. Multi-Location Support:** The system should allow users to manage inventory across multiple locations. This may include the ability to transfer inventory between locations and to view inventory levels for each location.

**7. User Management:** The system should allow administrators to manage user accounts, assign roles and permissions, and control access to the system. This helps ensure that only authorized users have access to sensitive information, such as financial data.

**8. Mobile Accessibility:** The system should be accessible from mobile devices, allowing users to manage or view their inventory, make requests on the go. This includes the ability to access inventory levels, add or modify items and view reports.

**9. Integration:** The inventory management system should be able to integrate with other systems, such as accounting. This helps streamline business processes and reduces the need for manual data entry.

**10. Data Security:** The system should provide robust data security to protect sensitive information, such as financial data, customer information, and inventory details. This includes secure storage of data, secure transmission of data, and access controls to prevent unauthorized access. The system should also have backup and recovery features to ensure that data can be restored in the event of a disaster.

**11. Real-time updates:** The system should provide real-time updates on inventory levels, sales, and purchasing activities. This helps users stay up-to-date with the latest information and make informed decisions.

**12. Invoice management**: The system should provide a way to manage and track invoices, including the ability to generate invoices, track payments, and view invoice history.

**13. Accounting integration:** The system should integrate with accounting software, such as QuickBooks or Xero, to streamline financial reporting and reduce manual data entry.

**14. Stock valuation:** The system should provide a way to value inventory, such as using first-in, first-out (FIFO) or last-in, first-out (LIFO) methods. This helps users determine the value of their inventory for financial reporting purposes.

**15. Multi-currency support:** The system should support multiple currencies, allowing users to manage inventory in their local currency.

**16. Vendor management:** The system should provide a way to manage and track vendors, including the ability to view vendor history, track payments, and generate vendor reports.

**17. Customizable dashboards:** The system should allow users to create custom dashboards that display real-time information about their inventory, and purchasing activities.

**18. Data import/export:** The system should allow users to import and export data from other systems, such as spreadsheets or databases, to help streamline data entry and reduce manual data entry.

**19. Automated alerts:** The system should provide automated alerts for low stock levels, and other events, helping users stay informed and take timely action.

**20. Stock adjustment:** The system should allow users to make adjustments to stock levels, such as to correct discrepancies or to account for damaged or expired items.

**21. Warehouse management:** The system should provide a way to manage and track warehouse activities, such as receiving, storing. This may include features such as barcode scanning, pick and pack functionality.

**22. Inventory forecasting:** The system should provide a way to forecast inventory levels based on purchasing data. This helps users plan for future inventory needs and avoid stockouts.

**23. Customizable workflows:** The system should allow users to customize workflows, such as purchase order approval processes, to meet the unique needs.

**24. Analytics and business intelligence:** The system should provide robust analytics and business intelligence tools to help users gain insights into their inventory, and purchasing activities. This may include data visualization, trend analysis, and predictive analytics.

**25. Purchase order approval:** The system should provide a way for users to approve purchase orders, helping ensure that only authorized purchases are made.

**26. Product management:** The system should provide a way to manage and track product information, such as product descriptions, pricing, and images. This helps users keep their product information up-to-date and accurate.

**27.** The requirements for an "on-click login" feature using a Microsoft account could include the following:

* **User registration:** The ability for users to create a Microsoft account if they do not already have one.
* **Login button:** A button or link that allows users to initiate the login process with a single click.
* **Microsoft account authentication:** The ability to authenticate a user's Microsoft account credentials and retrieve their profile information.
* **User authorization:** The ability to grant the user's authorization for the application to access their Microsoft account information.
* **Token management:** The ability to securely store and manage the access token received from Microsoft.
* **Single sign-on (SSO):** The ability to recognize a user who has already logged in with their Microsoft account and automatically log them in without requiring additional authentication.
* **User profile information:** The ability to access and display the user's Microsoft account information, such as their name, email, and profile picture.
* **Logout:** The ability for users to log out of their Microsoft account and end the session.
* **Error handling:** The ability to handle and display error messages in the event of authentication or authorization failure.
* **Security:** The ability to securely store and transmit sensitive user information, such as passwords and access tokens, in accordance with industry standards.

**3.2 Non-Functional Requirements**

**3.2.1 User Interface Requirements**

* The interfaces must be user-friendly and should be easy to learn.
* Each role in the system has its own graphical user interface.
* The interface for the other users should be protected and only accessed if the correct email and password are taken as input.
* If the login is successful, the system will direct the user only to the page that they have access on.
* Management page, where the users that have the permission can manage other lower-ranking users.
* Dynamic screen size for any device like mobile phone, tablet, laptop or computer.
* Real time notification for HRS and It department.

But the User Interface Requirements also consist of:

1. **Usability:** The system should have an intuitive and user-friendly interface, making it easy for users to navigate and perform tasks.
2. **Consistency:** The system should provide a consistent user experience, with a consistent look and feel, to minimize confusion and maximize efficiency.
3. **Responsiveness:** The system should provide a responsive interface, adapting to different screen sizes and orientations, to ensure that the system is usable on a variety of devices.
4. **Navigation:** The system should provide clear and intuitive navigation, allowing users to easily access and navigate to different sections of the system.
5. **Feedback:** The system should provide clear and timely feedback, such as through notifications or status messages, to inform users of the results of their actions and provide context for their actions.
6. **Error handling:** The system should provide effective error handling, with clear and detailed error messages, to help users troubleshoot and resolve issues quickly.
7. **Search functionality:** The system should provide robust search functionality, allowing users to search for specific items or information within the system.

**3.2.2 Usability**

**1. User-friendly interface:** The system should have a user-friendly interface that is intuitive and easy to use, with clear navigation and understandable graphics.

**2. Responsive design:** The system should be designed with a responsive layout that adjusts to different screen sizes, allowing users to access the

system from desktop computers, tablets, or smartphones.

**3. Customizable settings**: The system should allow users to customize settings, such as default settings for new users, to meet the unique needs of their business.

**4. Role-based access control:** The system should provide role-based access control, allowing users to restrict access to sensitive information and functionality based on user role.

**5. Context-sensitive help:** The system should provide context-sensitive help, such as tooltips or inline help, to assist users in understanding the system's features and functionality.

**6. Performance:** The system should have fast and reliable performance, with minimal load times and a responsive user interface.

**7. Error handling:** The system should have robust error handling, providing clear and concise error messages when problems occur and allowing users to quickly resolve errors.

**8. Ease of setup:** The system should be easy to set up and configure, with clear instructions and a streamlined setup process.

**9. Data migration:** The system should provide a way to migrate data from existing systems, such as spreadsheets or databases, to help users transition to the new system smoothly.

**10. Search functionality:** The system should have a robust search functionality, allowing users to quickly find the information they need, such as specific products or suppliers.

**11. Sorting and filtering:** The system should provide the ability to sort and filter data, such as by product category or supplier name, to help users quickly find the information they need.

**12. Customizable dashboards:** The system should provide customizable dashboards, allowing users to create and save custom views of inventory data and other information that is important to their business.

**13. Notifications and alerts:** The system should provide notifications and alerts, such as email notifications or system-generated messages, to help users stay informed about important events, such as low stock levels or upcoming expiration dates.

**14. Integration with external systems:** The system should integrate with external systems, such as accounting or ERP systems, to provide a seamless and integrated user experience.

**15. User-defined fields:** The system should provide the ability to add user-defined fields to product records and other data, allowing some users to store and manage information that is specific to their business.

**16. Keyboard shortcuts:** The system should provide keyboard shortcuts, allowing users to perform common tasks quickly and efficiently, without having to navigate the system's user interface.

**17. Data import and export:** The system should provide the ability to import and export data, such as product and supplier information, to and from external systems and files.

**18. Multi-language support:** The system should provide multi-language support, allowing users to view the system in their preferred language, to help increase accessibility and usability for a global audience.

**19. User authentication:** The system should provide a secure authentication mechanism, such as password-based authentication or two-factor authentication, to protect sensitive data and ensure that only authorized users can access the system.

**20. Data backup and recovery**: The system should provide a way to backup and recover data, such as through regular backups or snapshots, to protect against data loss in the event of a system failure or disaster.

**21. Visual reporting:** The system should provide visual reporting and analytics, such as graphs and charts, to help users quickly understand trends and patterns in their inventory data.

**22. Scalability**: The system should be designed for scalability, allowing the system to accommodate growing amounts of data and increasing numbers of users without sacrificing performance.

**23. Compliance with industry standards:** The system should comply with industry standards, such as security and data privacy standards, to ensure that sensitive data is protected and secure.

**24. Technical support:** The system should provide technical support, such as through a support portal or helpdesk, to help users resolve technical issues and receive assistance with the system.

**25. Data security:** The system should provide a secure data storage and transmission mechanism, such as encryption, to protect sensitive data from unauthorized access and theft.

**3.2.3 Performance**

This particular application will be a web-application and will be stored in a web server. The performance of this application will depend on:

* The internet connection
* The efficiency of retrieving data from the database
* The number of active users accessing or using the application

Performance also depends on:

1. **Response time:** The system should have a fast response time, providing immediate feedback to users to ensure a smooth and efficient user experience.

2. **Throughput:** The system should be capable of handling high volumes of transactions and data, with fast and efficient processing of requests to ensure high throughput.

3. **Scalability:** The system should be designed for scalability, with the ability to handle increasing amounts of data and traffic, to ensure that performance remains fast and efficient even as the system grows.

4. **Performance optimization:** The system should be optimized for performance, with efficient algorithms and data structures, to minimize latency and improve response times.

5. **Load testing:** The system should undergo load testing, to evaluate the system's performance under different load conditions and to identify performance bottlenecks.

6. **Monitoring and analysis:** The system should provide monitoring and analysis tools, such as real-time monitoring and logging, to track and diagnose performance issues and improve response times.

7. **Caching:** The system should implement caching, such as storing frequently accessed data in memory, to improve response times and reduce latency.

8. **Distributed systems:** If the system is deployed on multiple servers, it should use distributed systems techniques, such as load balancing and data replication, to improve performance and ensure fast and responsive performance.

9. **Network optimization:** The system should be optimized for network performance, with efficient use of network bandwidth and low network latency, to ensure fast and responsive performance.

10. **Capacity planning:** The system should include capacity planning, to ensure that the system has sufficient resources to meet the needs of the organization over time, and to minimize the risk of performance degradation or downtime.

**3.2.4 Availability**

The availabilities of this application are:

* This application will be accessible any time during the day and night which means that it will be accessible 24/7.
* If something goes wrong or if an action gives an error, messages related to that action will be displayed.
* Different type of users will have different availability and access.

Availability also depends on:

1. **Uptime:** The system should have a high uptime, with minimal downtime, to ensure that users have continuous access to the system and its data.
2. **Redundancy:** The system should provide redundancy, such as multiple servers and backup systems, to ensure that the system remains available even in the event of failures or outages.
3. **Load balancing:** The system should provide load balancing, such as through a load balancer, to distribute traffic across multiple servers and ensure that the system remains available even under heavy load.
4. **Failover:** The system should provide failover, such as automatic failover to a secondary server in the event of a failure, to ensure that the system remains available even in the event of failures or outages.
5. **Disaster recovery:** The system should provide a disaster recovery mechanism, such as a secondary backup server, to ensure that the system can recover from failures and continue functioning even in the event of a disaster.
6. **Service level agreements (SLAs):** The system should have clearly defined service level agreements, such as guaranteed uptime and response times, to ensure that users have consistent and reliable access to the system and its data.
7. **Maintenance windows:** The system should have regularly scheduled maintenance windows, during which maintenance and updates can be performed, to ensure that the system remains up-to-date and secure.
8. **Monitoring and alerting:** The system should provide monitoring and alerting, such as through logs and notifications, to detect and respond to failures and outages in real-time.
9. **Backups:** The system should provide regular backups, such as daily or weekly backups, to ensure that data can be recovered in the event of failures or outages.
10. **Scalability:** The system should be designed for scalability, allowing the system to accommodate growing amounts of data and increasing numbers of users without sacrificing performance or availability.
11. **This Software:** This software is supposed to cover everything physical that the university possesses.

**3.2.5 Latency**

The latency of this application will depend on:

* The size of the database
* The internet connection data transfer rate
* The speed of which data is retrieved from the specific database
* How many users are operating in the website at that specific time

Latency also depends on:

1. **Response time:** The system should have a fast response time, providing immediate feedback to users to ensure a smooth and efficient user experience.
2. **Latency targets:** The system should meet specific latency targets, such as response times within a certain number of milliseconds, to ensure that users have a fast and responsive experience.
3. **Real-time updates:** The system should provide real-time updates, allowing users to see the latest information in real-time, without delays.
4. **Performance optimization**: The system should be optimized for performance, with efficient algorithms and data structures, to minimize latency and improve response times.
5. **Load testing:** The system should undergo load testing, to evaluate the system's response times under different load conditions and to identify performance bottlenecks that may impact latency.
6. **Monitoring and analysis**: The system should provide monitoring and analysis tools, such as real-time monitoring and logging, to track and diagnose performance issues and improve response times.
7. **Caching:** The system should implement caching, such as storing frequently accessed data in memory, to improve response times and reduce latency.
8. **Distributed systems:** If the system is deployed on multiple servers, it should use distributed systems techniques, such as load balancing and data replication, to minimize latency and ensure fast and responsive performance.
9. **Network optimization:** The system should be optimized for network performance, with efficient use of network bandwidth and low network latency, to ensure fast and responsive performance.
10. **Scalability:** The system should be designed for scalability, with the ability to handle increasing amounts of data and traffic, to ensure that response times remain fast and responsive even as the system grows.

**3.3 Manageability/Maintainability**

**3.3.1 Monitoring**

* This application will be built to be secure, and trustworthy
* For every error that occurs a message will pop up and the user will be notified about the problem and its possible causes
* The user should put the correct and corresponding email and password in order to proceed and use this application, and if the login credentials are not valid an error message will pop up and the user will get notified about that
* This application will also be built to be reliable and all information entered by the users will be validated
* A hierarchy will be set among users as we mentioned earlier by the HRS department

**3.3.2 Maintenance**

In case this application crashes the system will restart and redirect the user to the homepage. Everything that the user has done in the application will be saved. If the error occurs it’s safe to restart the web-page. If the application crashes again a maintenance break will occur and the users will get notified for this.

**3.3.3 Operations**

* The software will be online 24/7
* All data of the users will be validated
* All operations based on the user will be available
* In case of something uninspected happens, it will be a backup server available to restore all the information
* Each user has his own unique credentials and they are the only ones that can access their page
* The password of the user will be encrypted, and can be modified only by him
* The web-application will be properly updated if other needs are required

**3.4 Organizational requirements**

**3.4.1 Environmental requirements**

Environmental requirements depend on:

1. **Hardware compatibility:** The system should be compatible with a range of hardware platforms and configurations, including desktops, laptops, mobile devices, and servers.
2. **Operating system compatibility:** The system should be compatible with a range of operating systems, including Windows, macOS, Linux, and Unix.
3. **Network compatibility:** The system should be compatible with different types of networks, including LANs, WANs, and the internet.
4. **Database compatibility:** The system should be compatible with different types of databases, including relational databases, NoSQL databases, and cloud databases.
5. **Browser compatibility:** If the system is a web-based application, it should be compatible with different web browsers, including Chrome, Firefox, Safari, and Edge.
6. **Power requirements:** The system should have low power requirements, to reduce energy consumption and minimize the impact on the environment.
7. **Physical security:** The system should be physically secure, with appropriate measures in place to protect the hardware, software, and data from theft, tampering, or damage.
8. **Data center requirements:** If the system is hosted in a data center, it should meet the requirements of the data center, including temperature, humidity, and power requirements.

**3.4.2 Operational requirements**

Some of the operations that will be provided to the users are:

1. **Product management:** Users should be able to manage the items, including adding, editing, and deleting items, setting product information and pricing, and categorizing items.
2. **Inventory management:** Users should be able to manage inventory, including tracking stock levels, setting reorder points, and generating alerts for low stock.
3. **Order management:** Users should be able to manage orders, including creating and processing orders.
4. **Purchase order management:** Users should be able to manage purchase orders, including creating and processing purchase orders, and generating reports on purchasing activity.
5. **Inventory reporting:** Specific users should be able to generate inventory reports, including inventory levels, inventory turnover, and inventory value.
6. **User management:** Admin should be able to manage users, including adding, editing, and deleting users, setting user permissions, and generating reports on user activity.
7. **Dashboard:** Specific users should have access to a dashboard, with real-time information on system activity, including inventory levels, and order status.

**3.5 External requirements**

**3.5.1 Ethical requirements**

Ethical requirements of this software are mainly depended on:

* **Transparency:** The system should be transparent in its operations, with clear explanations of data collection, use, and storage.
* **Responsibility:** The system should be designed to promote ethical behavior and to prevent unethical behavior, such as discriminatory practices or data misuse.
* **Responsibility:** The system should be designed to be responsible in its operations, including ensuring the security and privacy of user data and preventing data misuse.
* **Confidentiality:** The system should be designed to protect the confidentiality of user data and to prevent unauthorized access or disclosure of sensitive information.
* **Consent:** The system should be designed to obtain informed consent from users for the collection and use of their data.
* **Responsibility for design:** The system should be designed to promote ethical behavior and to prevent unethical behavior, such as discriminatory practices or data misuse.
* **Responsibility for operation:** The system should be operated in a responsible manner, with appropriate measures in place to protect the privacy and security of user data.
* **Responsibility for maintenance:** The system should be maintained in a responsible manner, with appropriate measures in place to ensure the privacy and security of user data.

It is important to note that ethical requirements vary by the scope of the application and should be operated in a manner that is consistent with ethical principles.

**3.5.2 Legislative requirements**

These Regulations require users to consider the health and safety risks to users and to carry out a risk assessment to protect users from exposure to reasonably foreseeable risks. Those risks include work-related violence. A risk assessment is an examination to:

* Comply with data protection and privacy laws

.

* Establish the significance of the risk.
* Identify and implement prevention and control measures.

**3.5.3 Security requirements**

Security requirements are a set of requirements that ensure the protection of the system, its data, and its users from potential threats and vulnerabilities. These security requirements are broken down in:

* **Authentication:** The system should have robust authentication mechanisms to ensure that only authorized users can access the system.
* **Access control:** The system should have robust access control mechanisms to ensure that users can only access the data and resources that they need to perform their work.
* **Data encryption:** The system should encrypt sensitive data, such as passwords and credit card information, to protect it from unauthorized access or theft.
* **Data backup and recovery**: The system should have a data backup and recovery plan in place to protect against data loss or corruption.
* **Physical security:** The system should have measures in place to protect the physical security of the system and its data, such as fire suppression systems and uninterruptible power supplies (UPS).
* **Network security:** The system should have measures in place to protect against network-based threats, such as firewalls, intrusion detection systems, and antivirus software.
* **Threat and vulnerability management:** The system should have mechanisms in place to detect, assess, and respond to security threats and vulnerabilities in a timely and effective manner.
* **Incident response plan:** The system should have an incident response plan in place to handle security incidents, such as data breaches, in a controlled and effective manner.

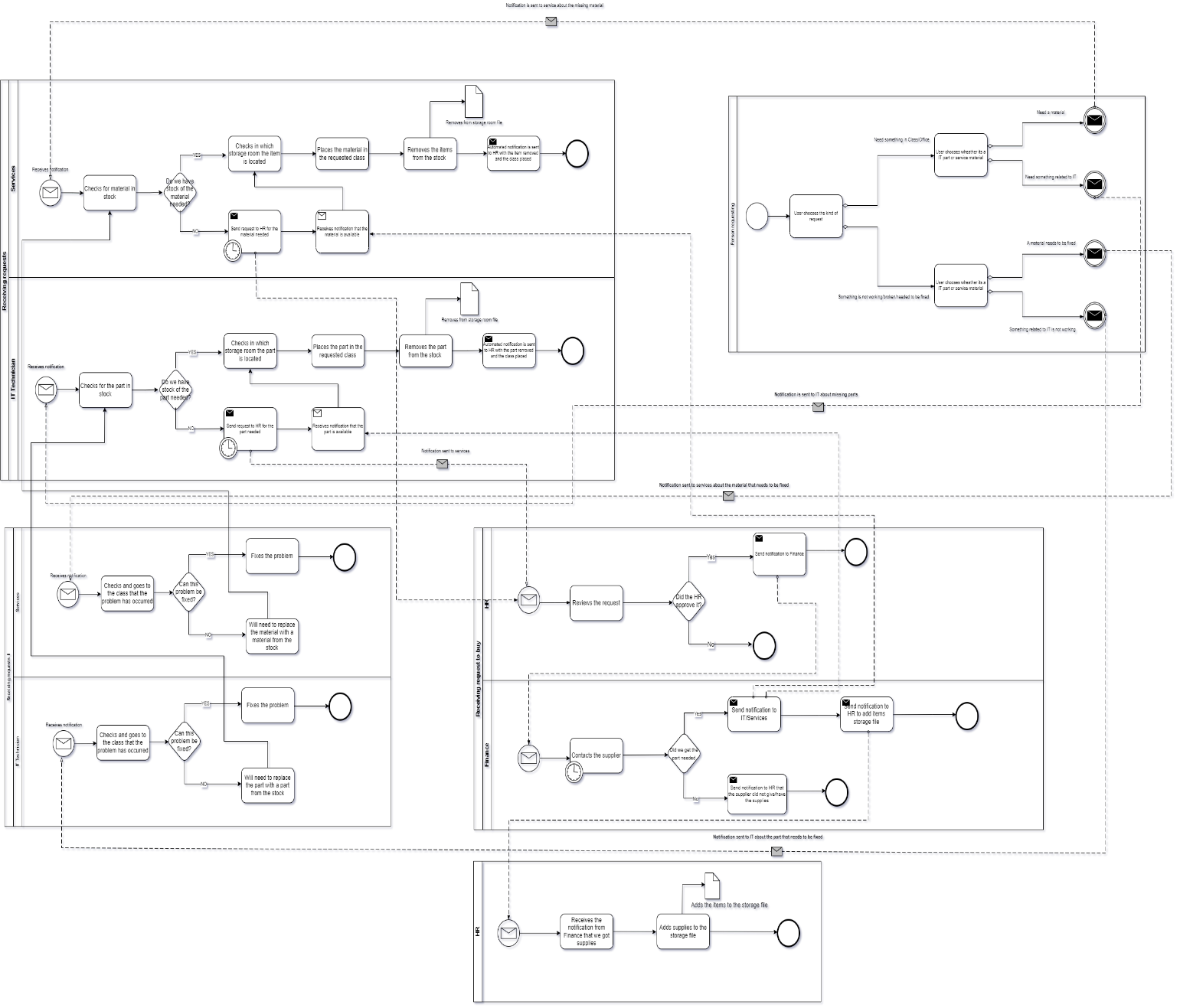
Having these security requirements in place helps to ensure that the system is protected against potential threats and vulnerabilities, and that the privacy and security of the system and its users is maintained.

**3.6 Domain requirements**

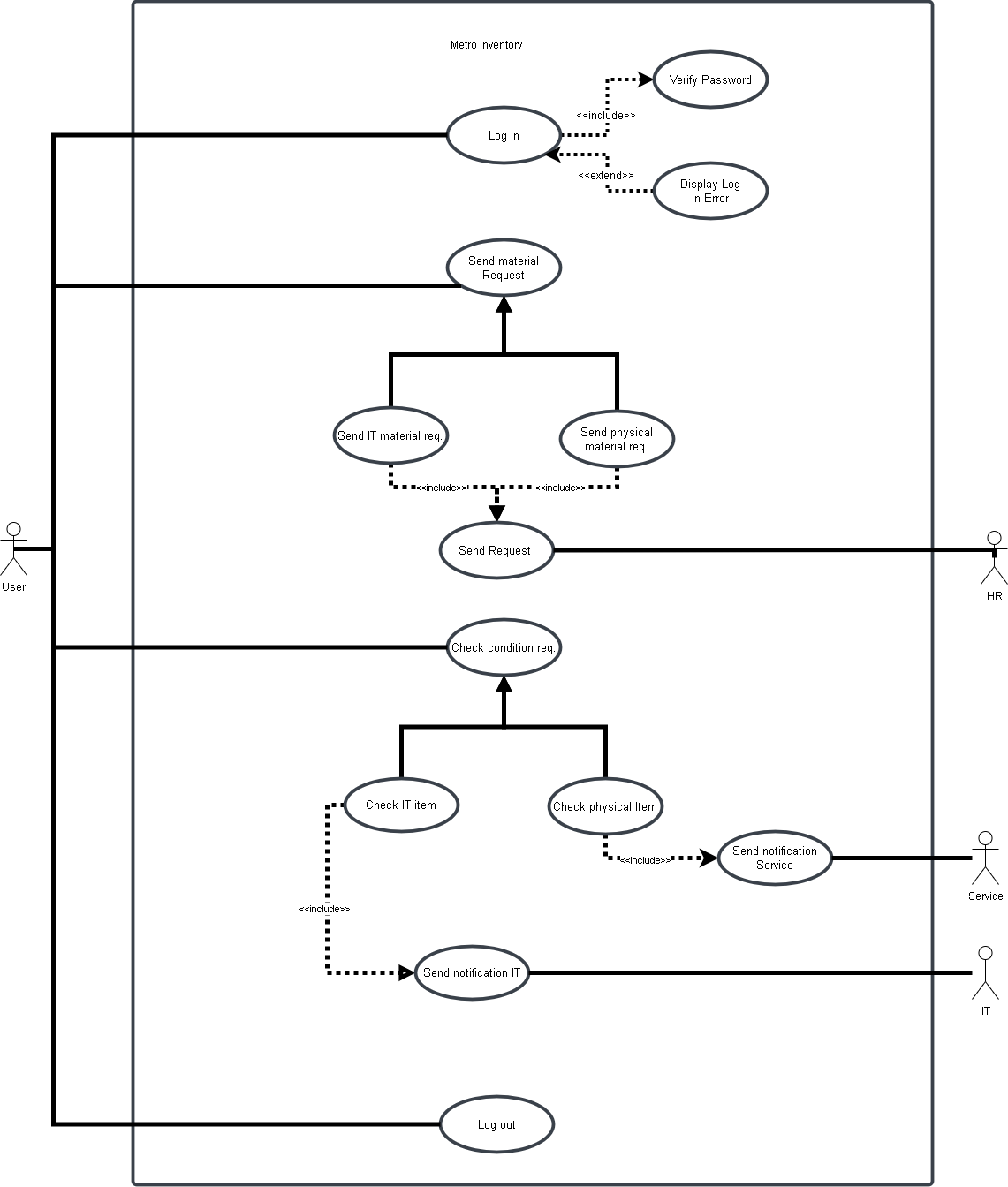
Domain requirements are requirements that are specific to the domain or industry in which the system will be used. They are determined by the unique characteristics and needs of that domain and reflect the specific constraints, processes, and regulations that apply to it.

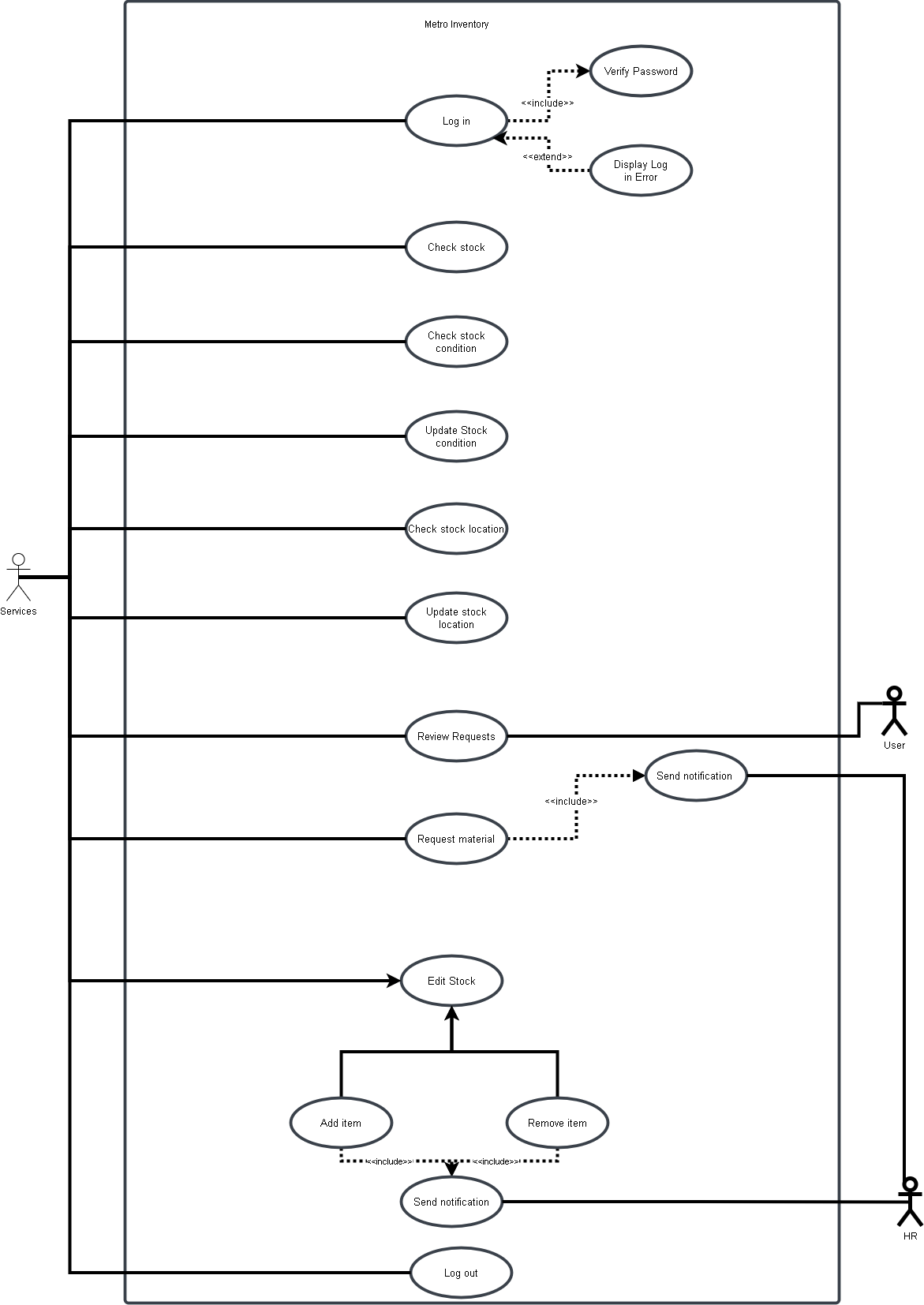
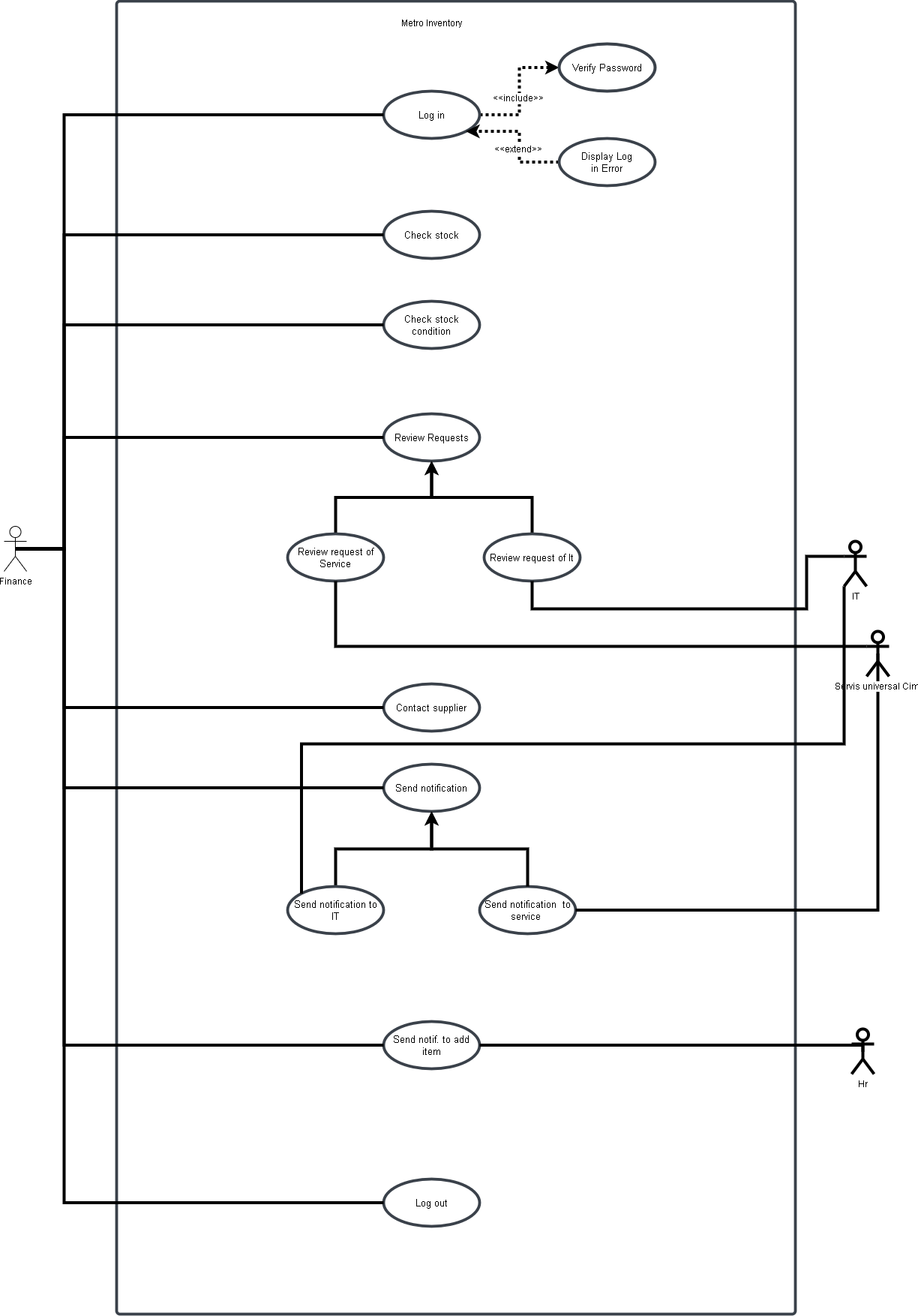
The system should have mechanisms to accurately track and manage inventory levels, including the ability to receive and process purchase orders, manage stock levels, and generate inventory reports.

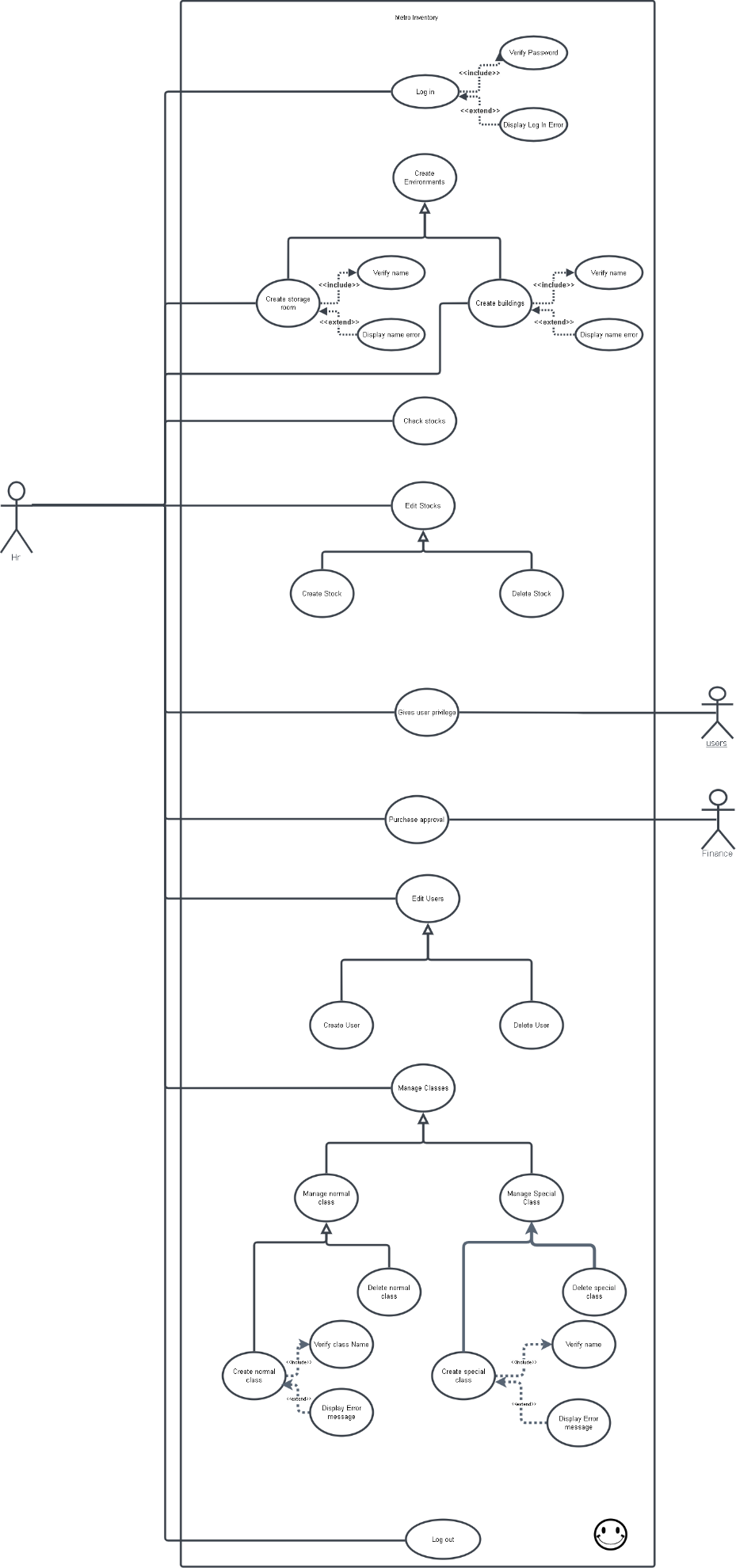
3.7 BPMN:

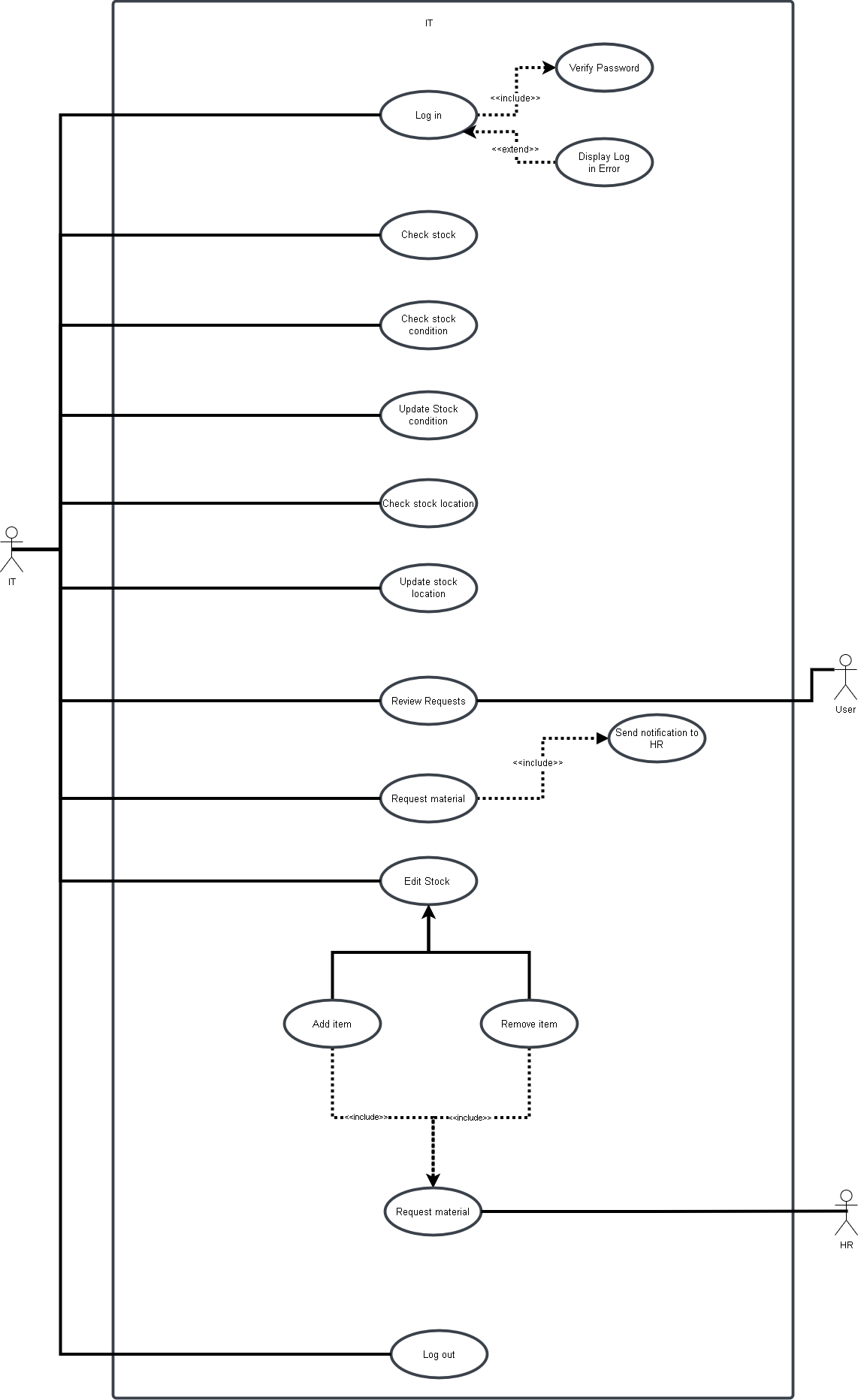


3.8 Use Case:

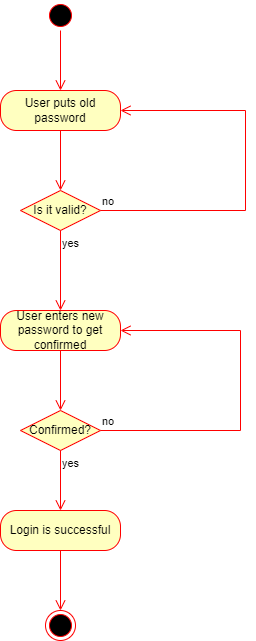


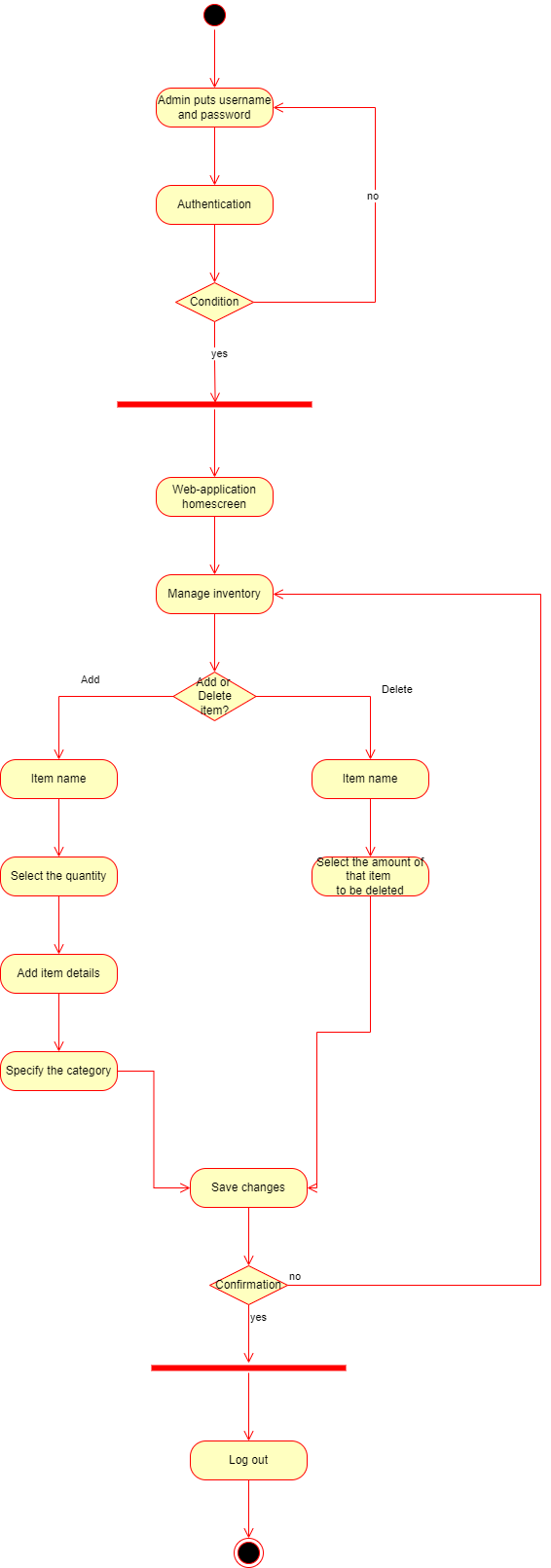


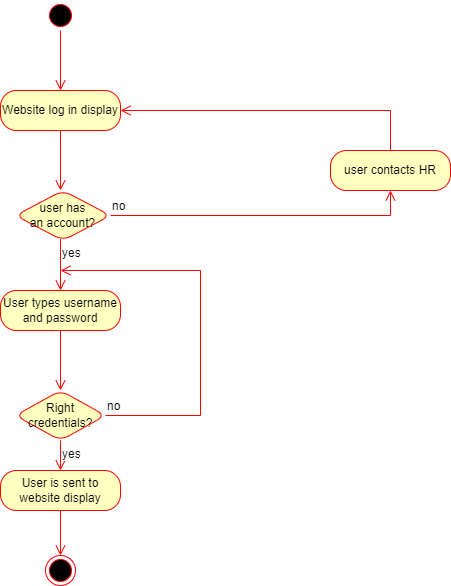


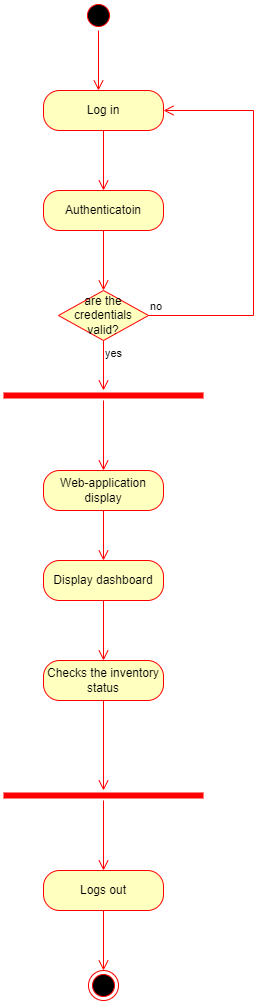


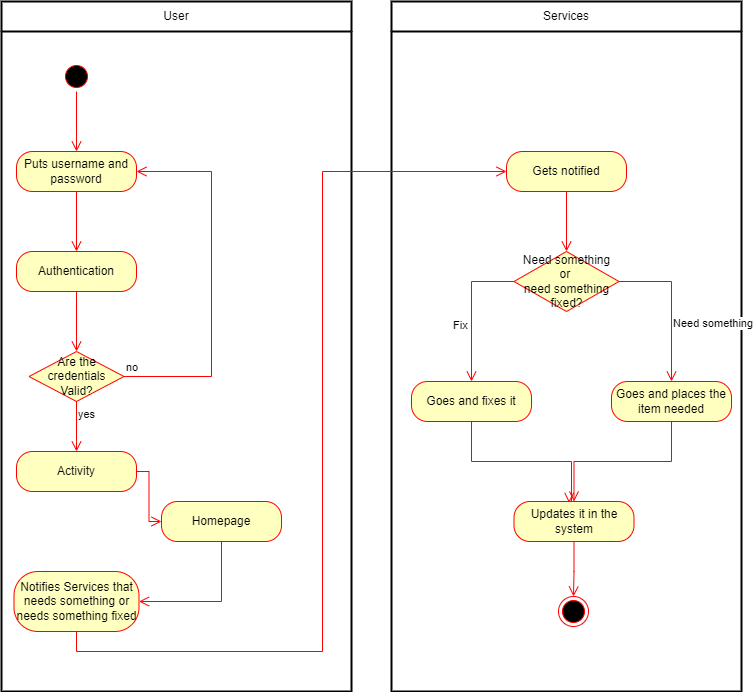
3.9 Activity Diagram:

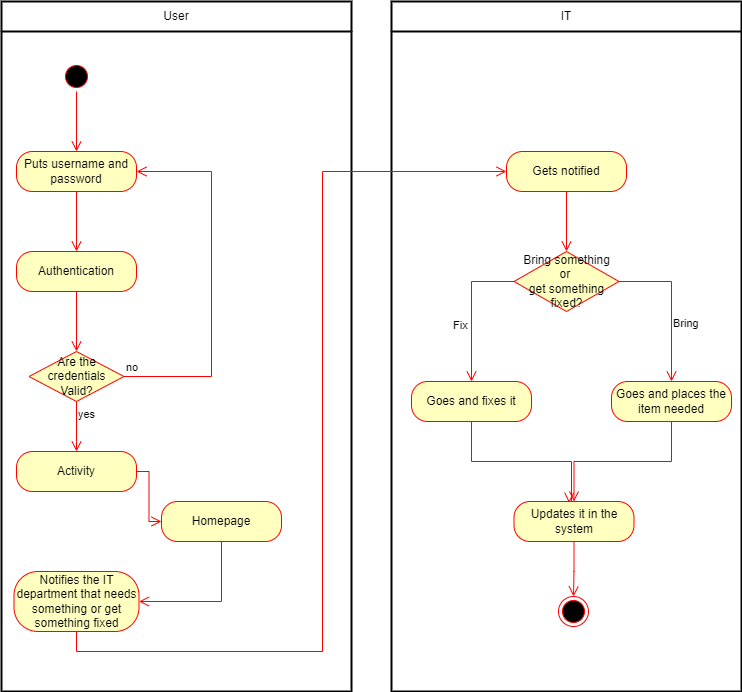




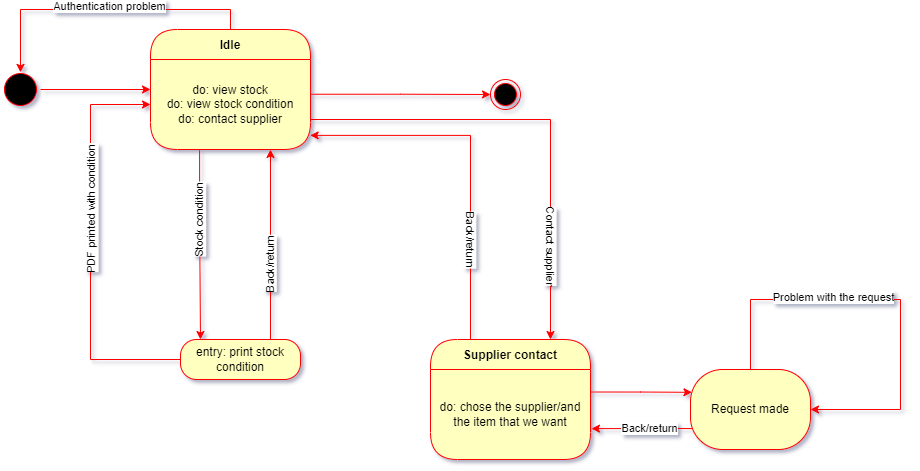


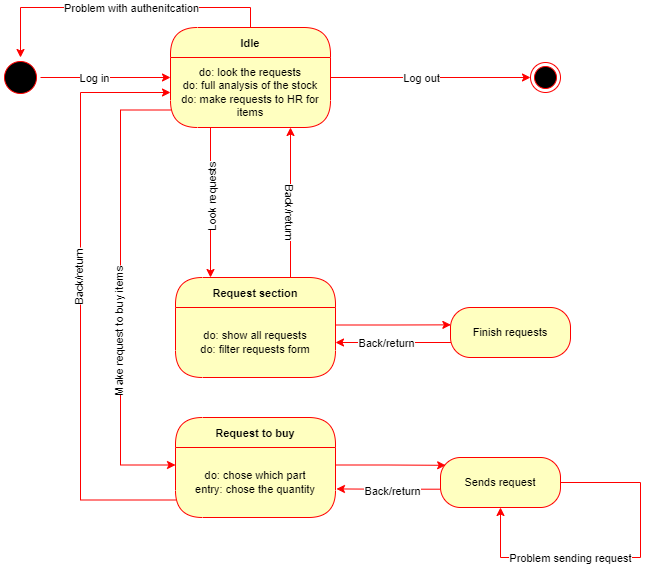


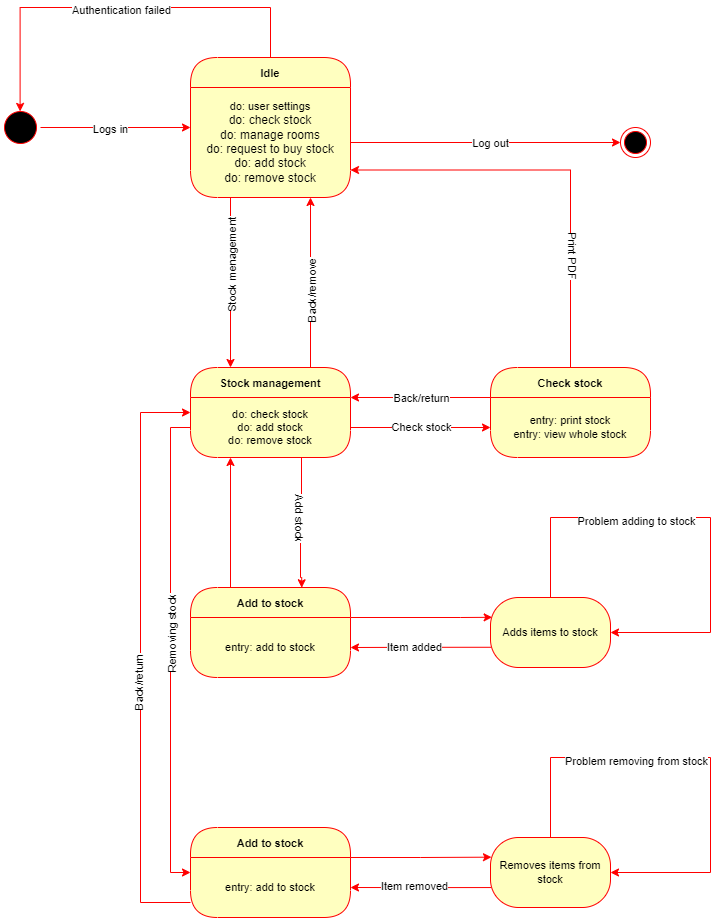


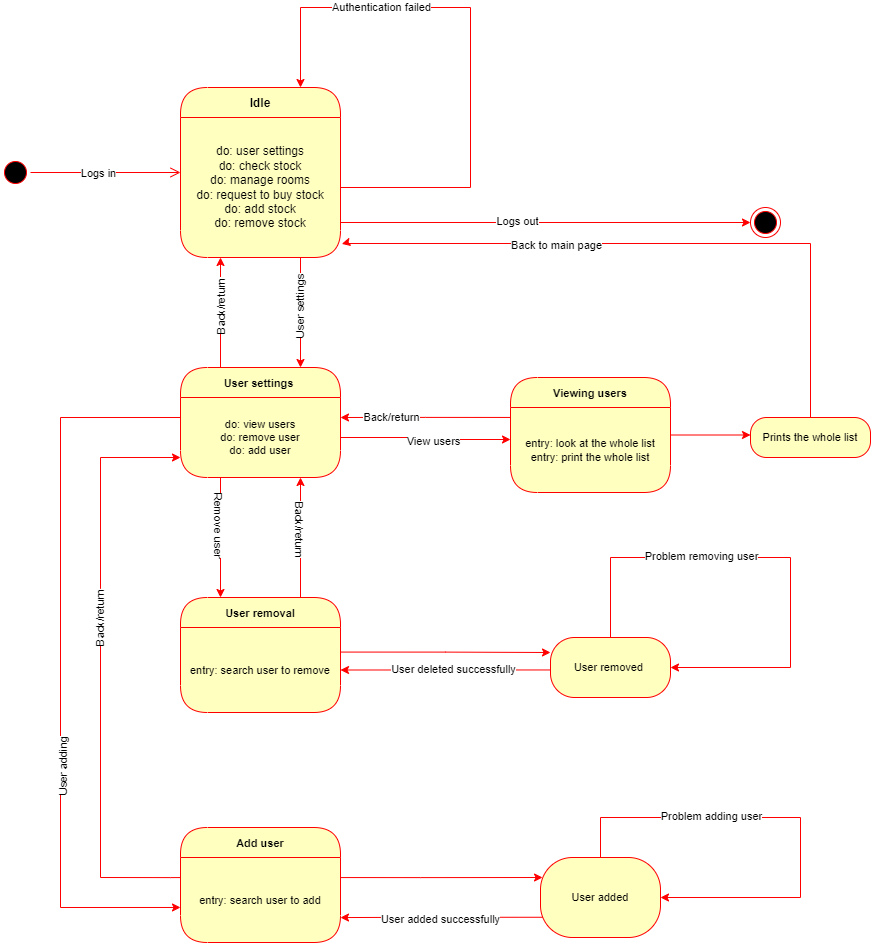


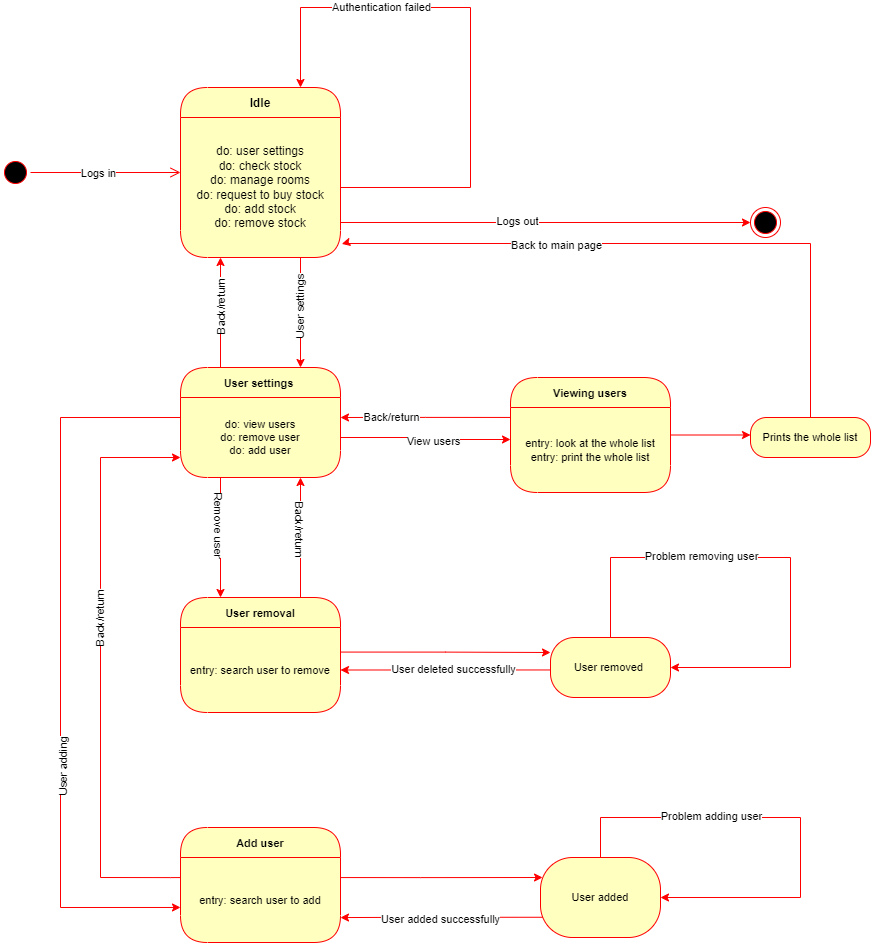
3.10 State Diagram:

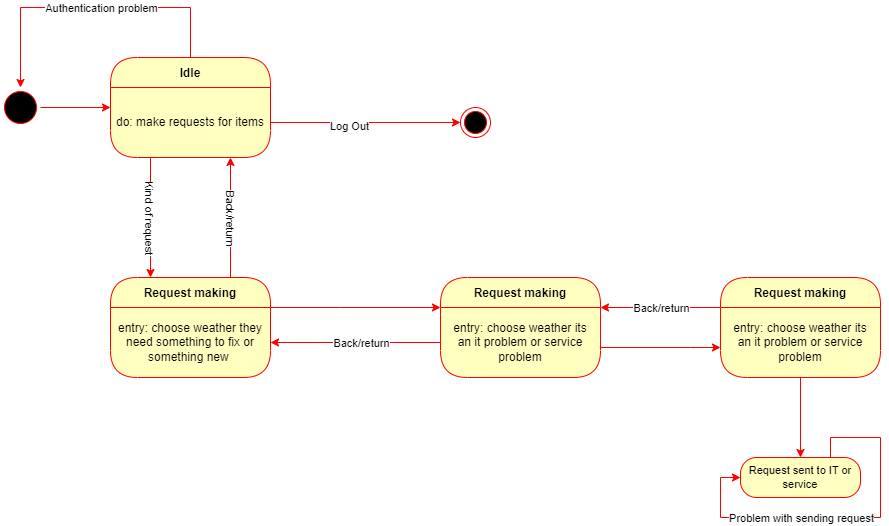


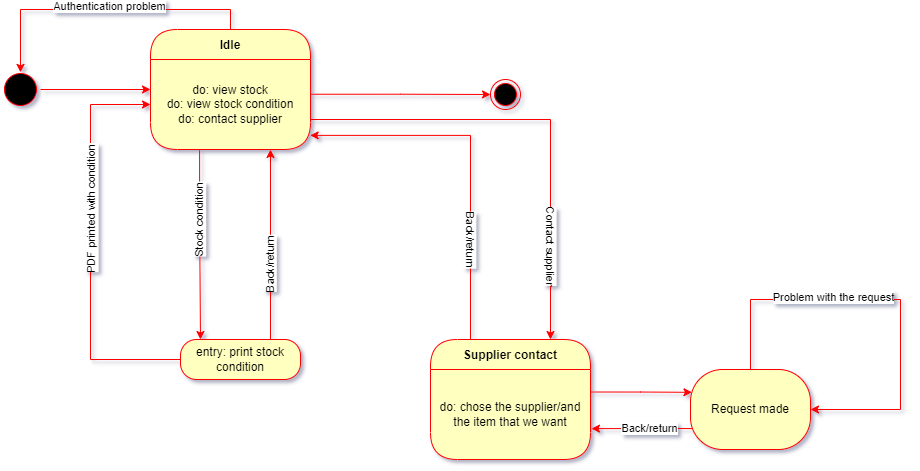


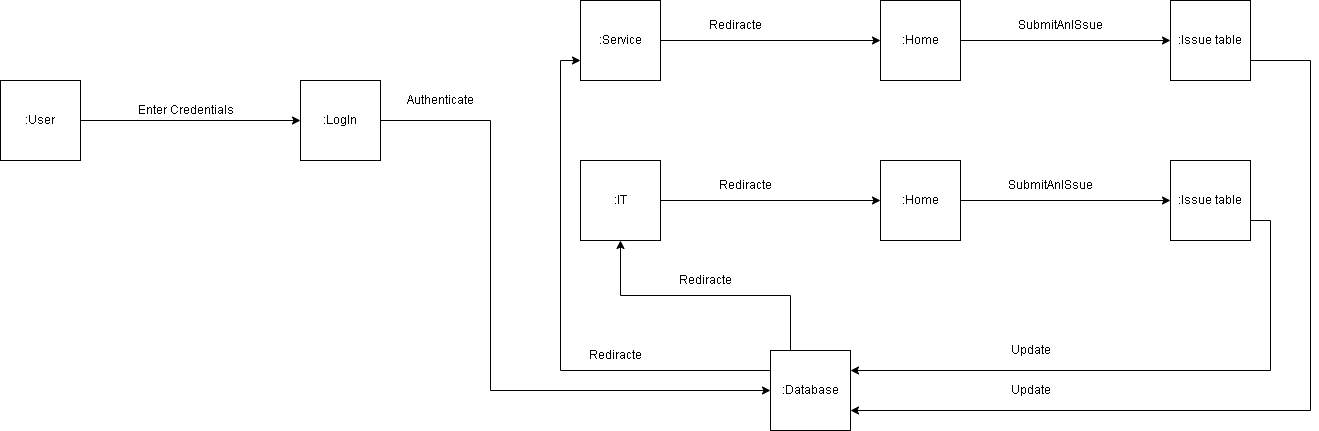


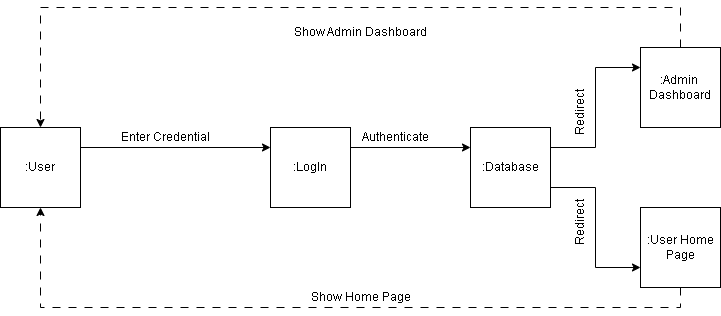


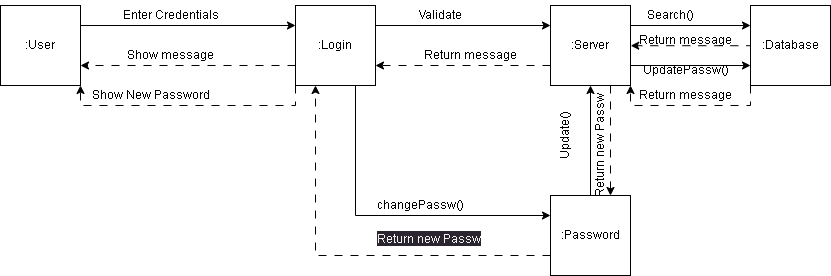


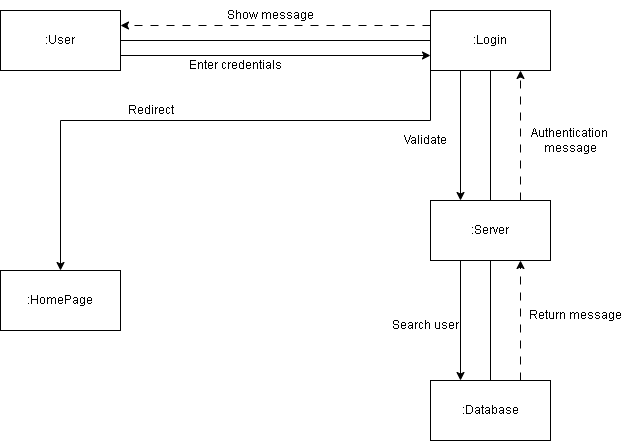


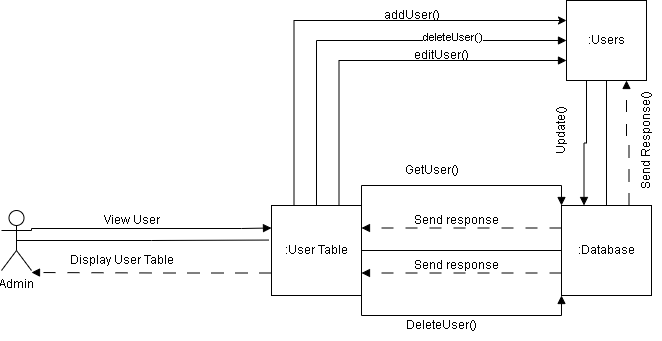


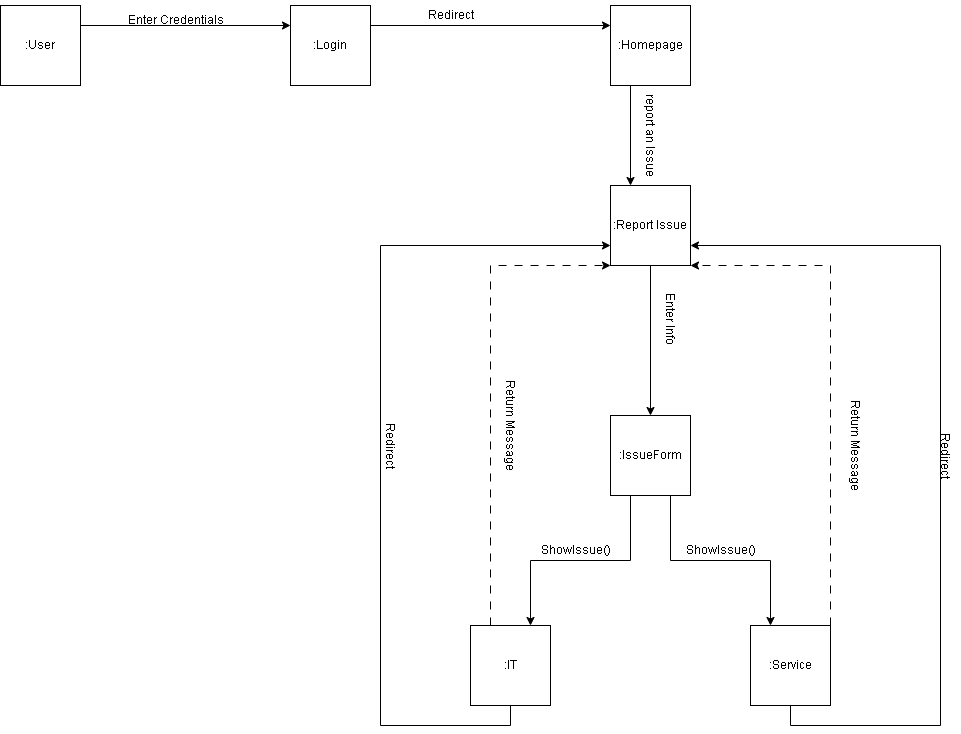


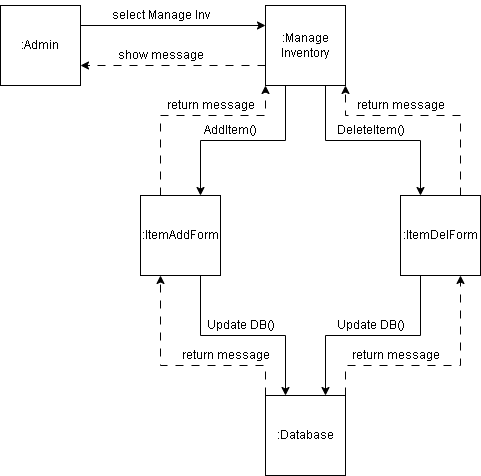




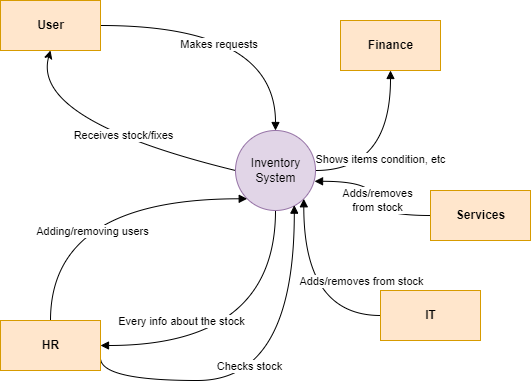


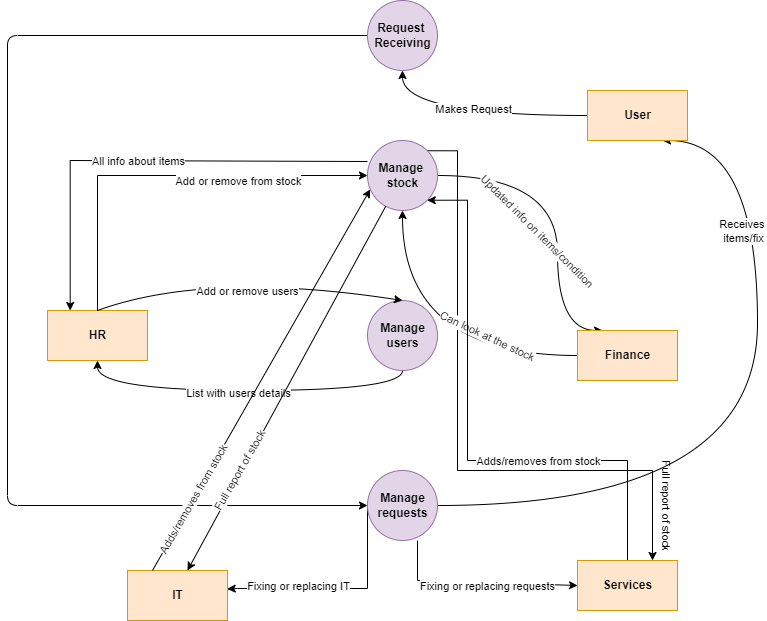


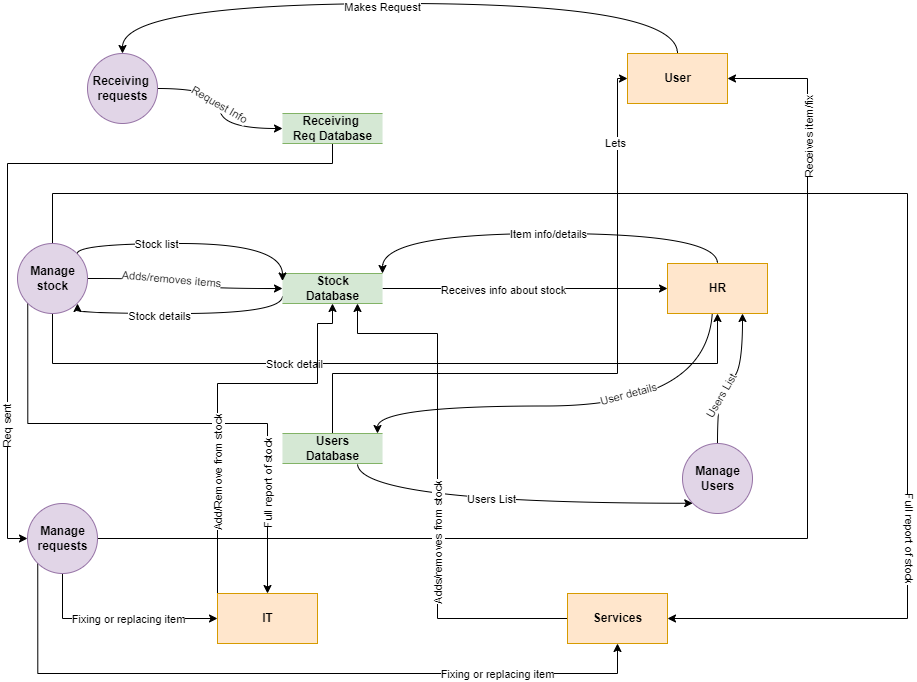




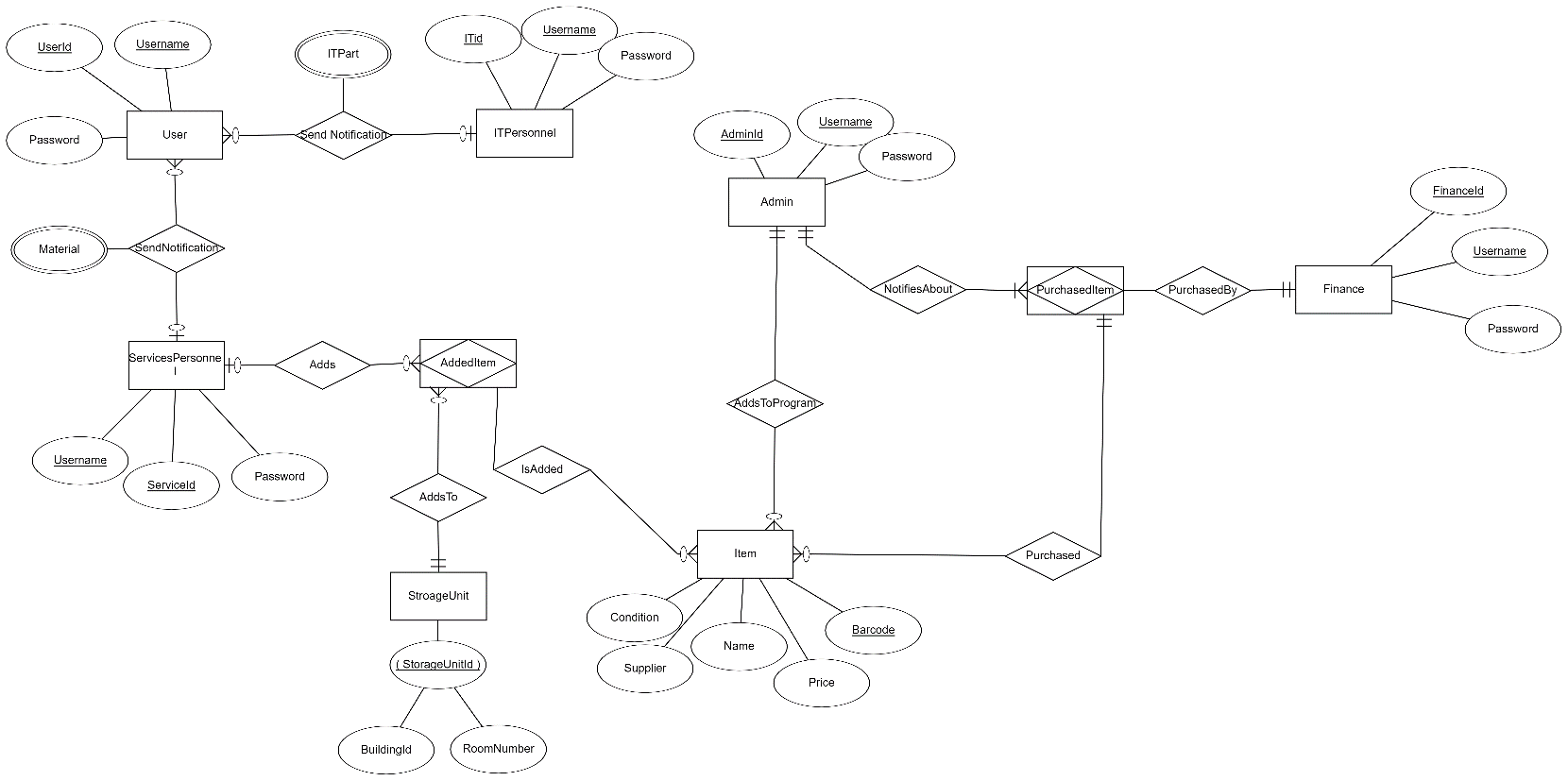
3.11 Data Flow Diagram:



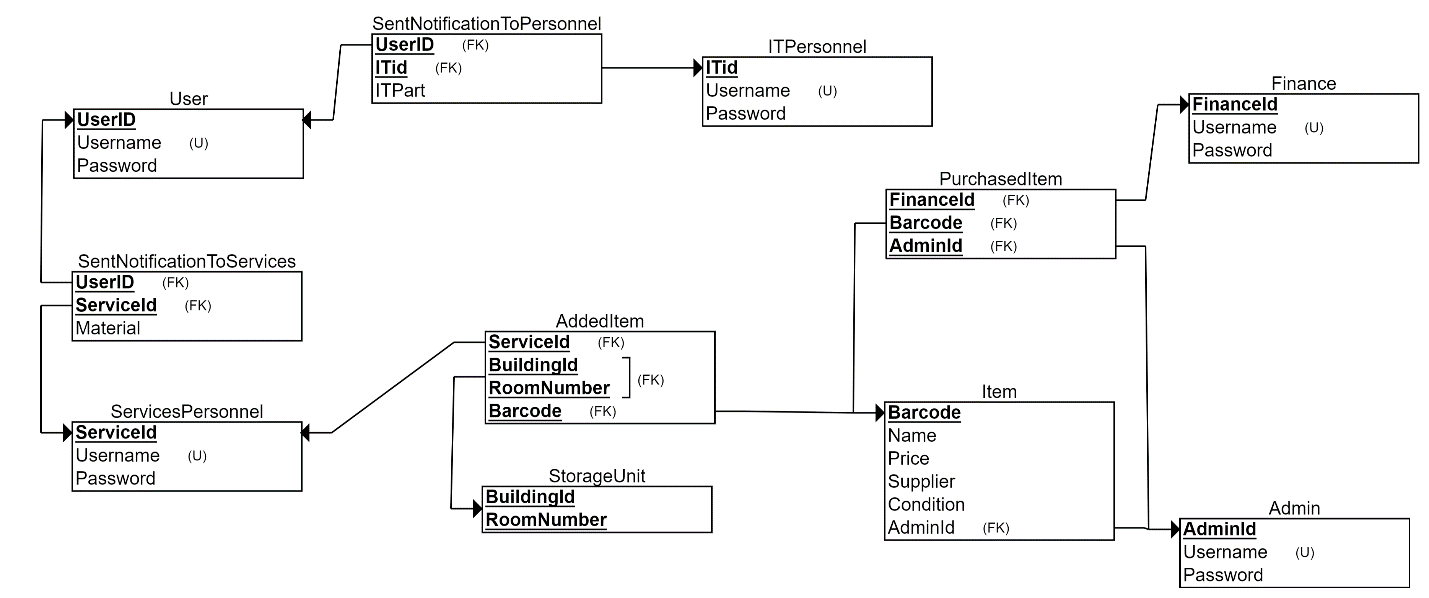




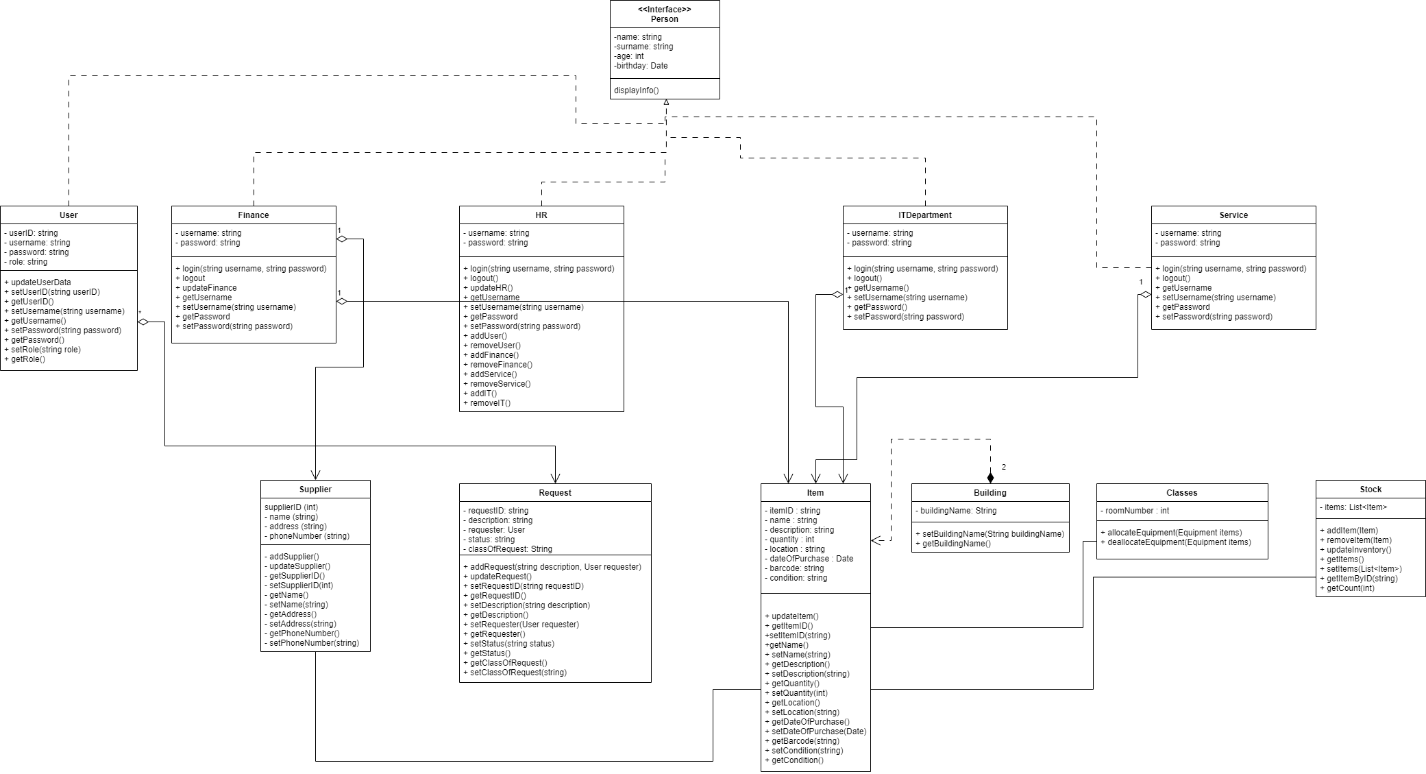
3.12 Entity Relation Diagram:



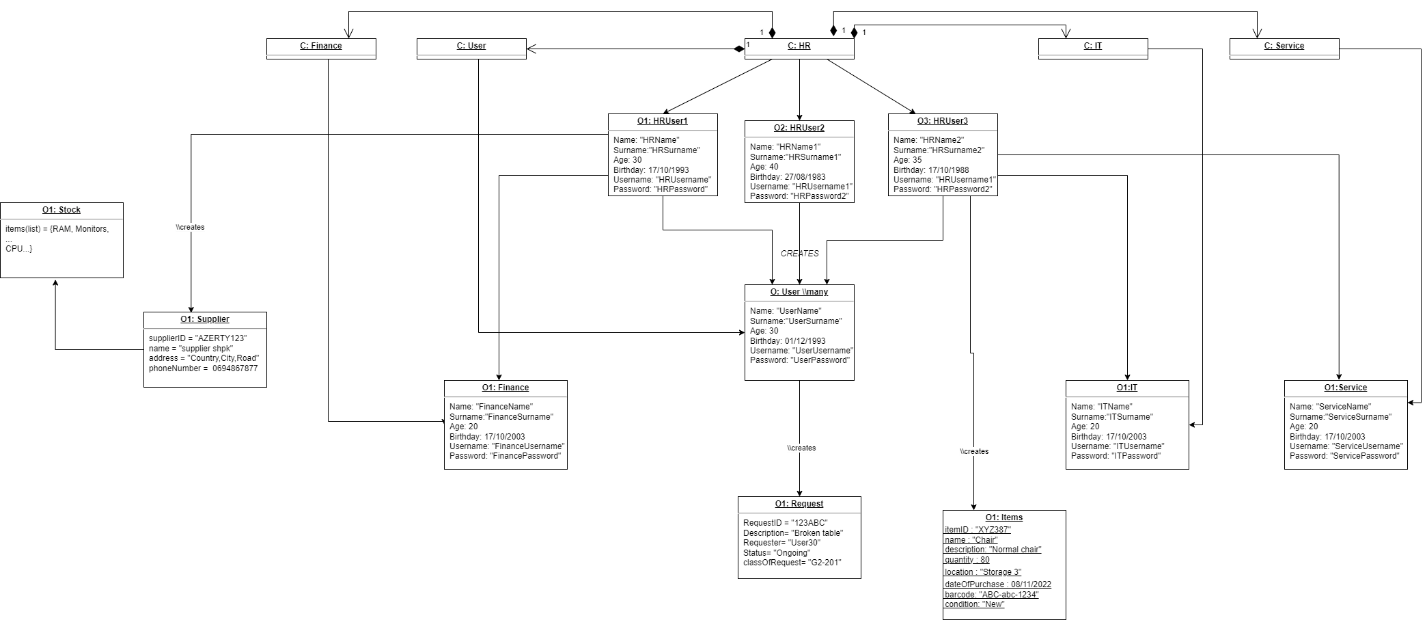
3.13 RS:



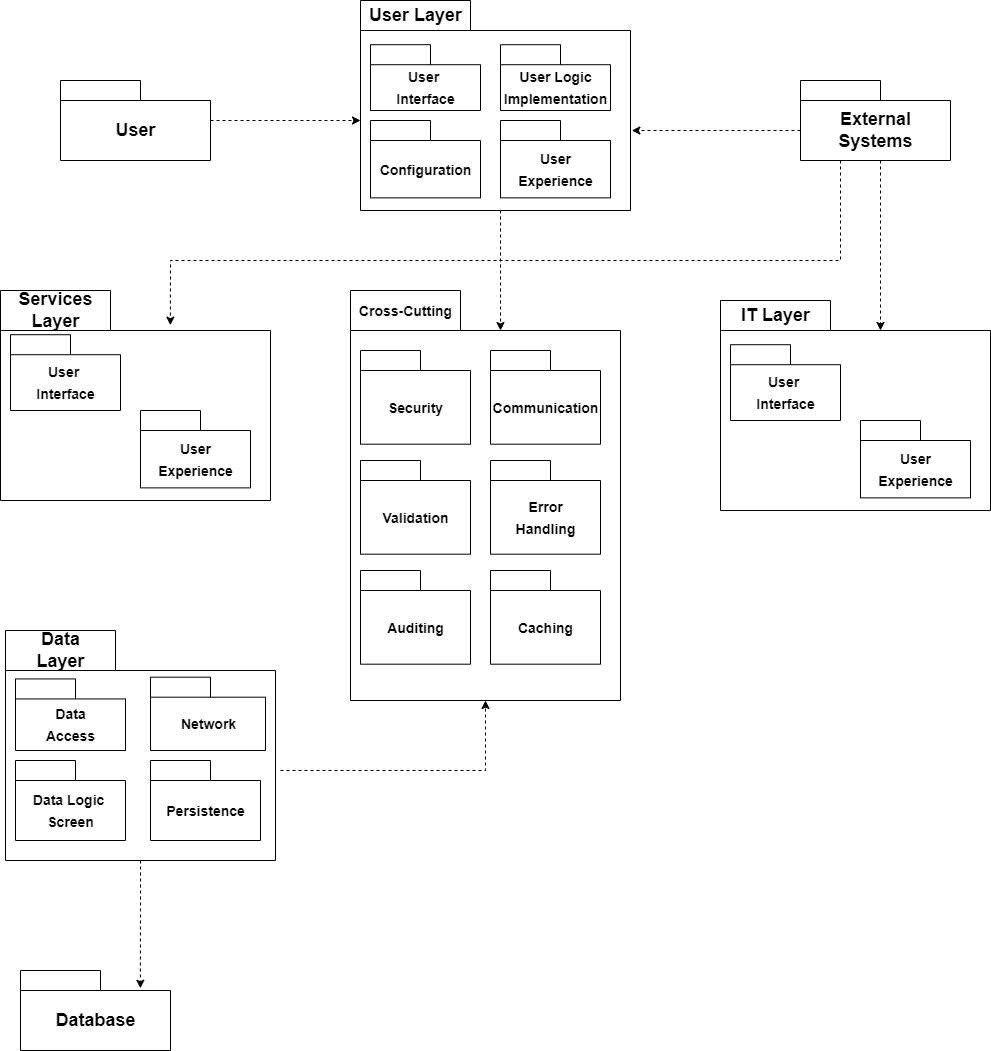
3.14 Class Diagram:



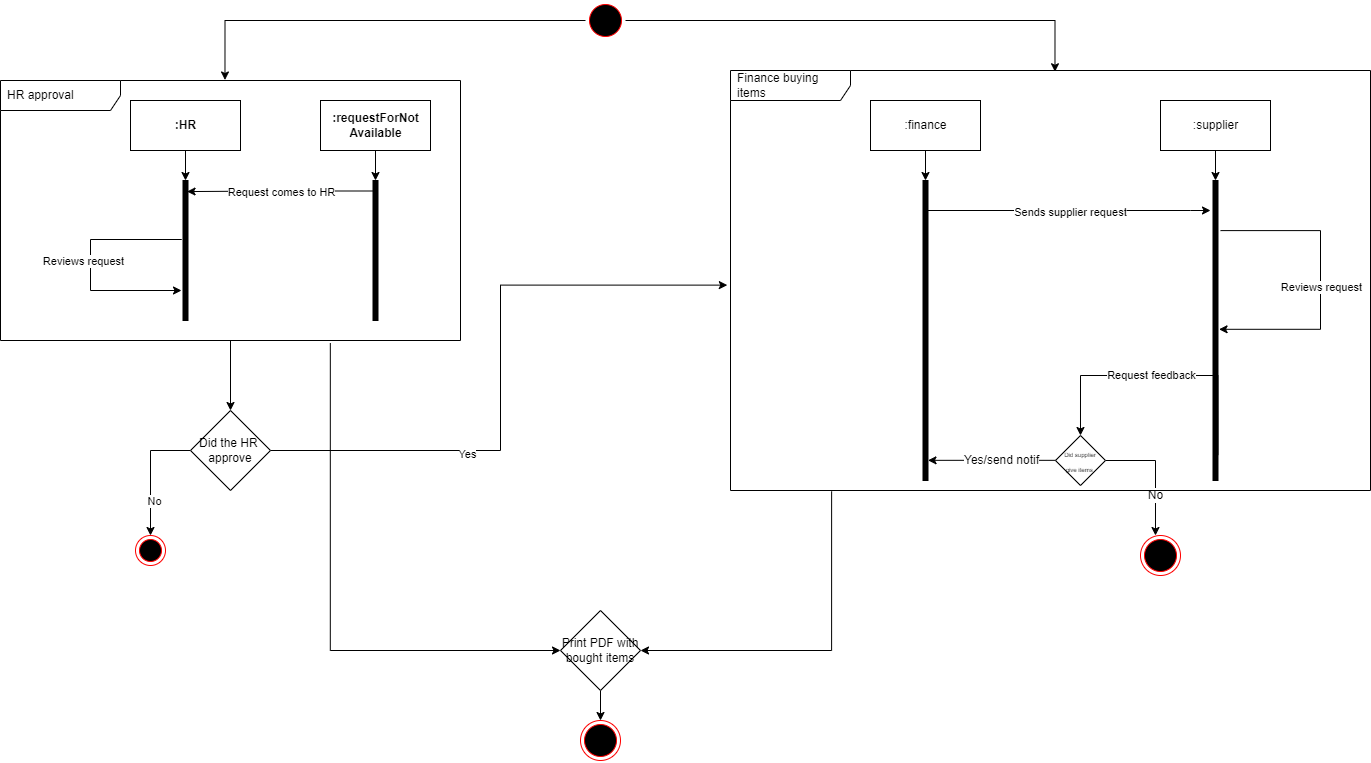
3.15 Object Diagram:

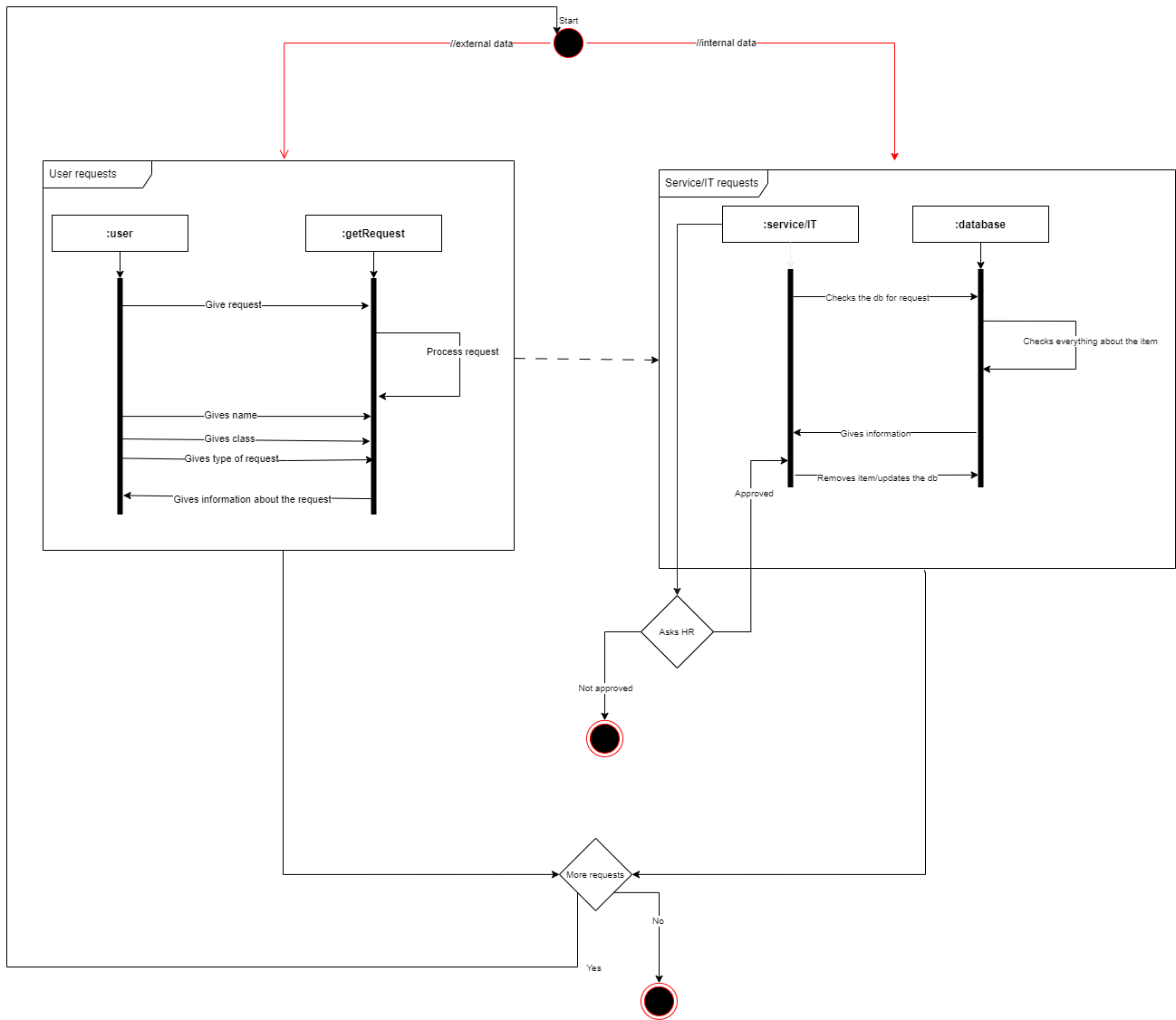


3.16 Package Diagram:

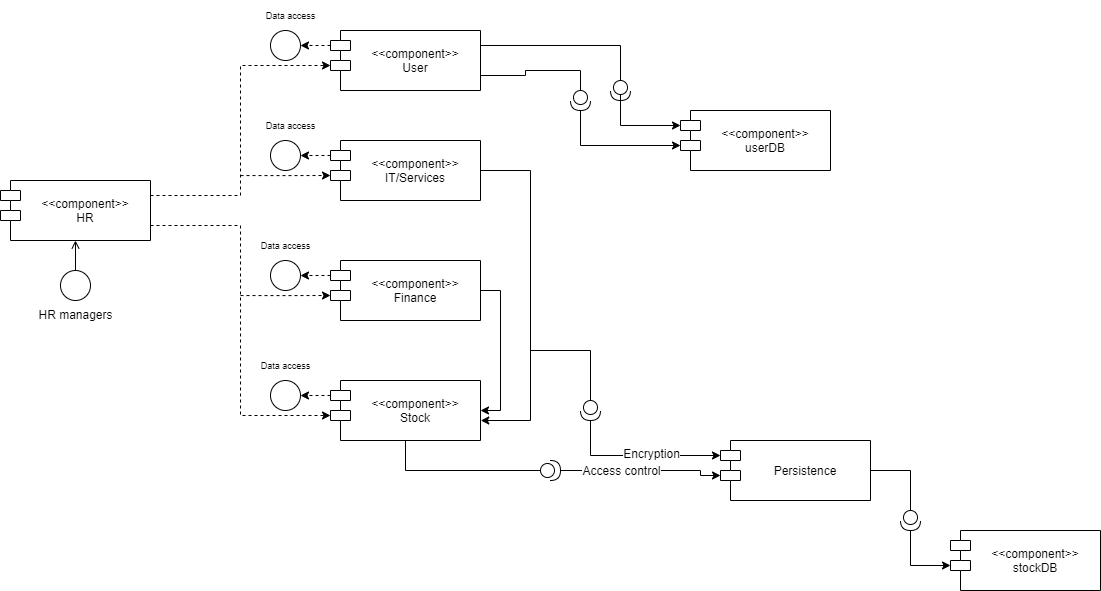
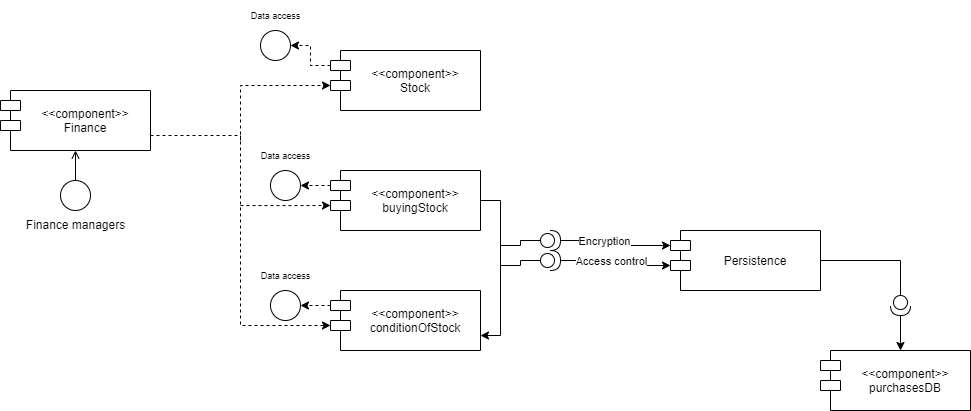


3.17 Interaction Overview Diagram:

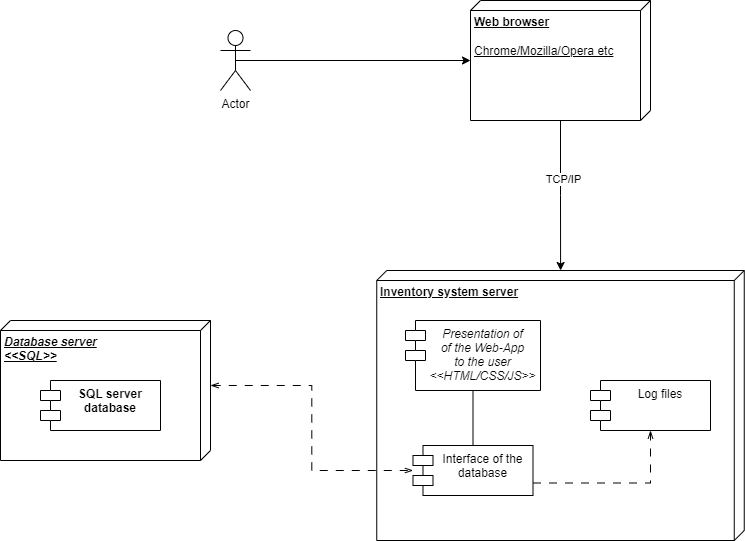




3.18 Component Diagram:



3.19 Deployment Diagram:



3.20 Sequence Diagram:

