**A picture containing text

Description automatically generated**

**School of Computing and Mathematical Sciences**

**CO7201 Individual Project**

**Preliminary Report**

**Stock Control System**

**Name - Apurva Vivek Kulkarni**

**Email Id –** [**avk7@**](mailto:avk7@)**student.le.ac.uk  
Student Id - 239064472**

**Project Supervisor: Anthony Conway**

**Principal Marker: Dr Paula Severi**

**Word Count: [XXXX]**

**[Submission Date]**

**DECLARATION**

All sentences or passages quoted in this report, or computer code of any form whatsoever used and/or submitted at any stages, which are taken from other people’s work have been specifically acknowledged by clear citation of the source, specifying author, work, date and page(s). Any part of my own written work, or software coding, which is substantially based upon other people’s work, is duly accompanied by clear citation of the source, specifying author, work, date and page(s). I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this module and the degree examination as a whole.

Name: Apurva Vivek Kulkarni

Date: 24-02-2025

Contents

[**1.** **Aims and Objectives** 3](#_Toc65510906)

[**2.** **Requirements** 3](#_Toc65510907)

[**3.** **Technical Specification** 3](#_Toc65510908)

[**4.** **Requirements Evaluation Plan** 3](#_Toc65510909)

[**5.** **Background Research and Reading list** 3](#_Toc65510910)

[**6.** **Time-plan and Risk Plan** 3](#_Toc65510911)

[**7.** **References** 3](#_Toc65510912)

# **Aims and Objectives**

A brief discussion on project relevance/motivations (why is it worth doing it?) and its main challenges.  
  
The aim of this project is to develop a **Stock Control System** that efficiently manages and monitors **computer component inventory** in a warehouse. The system will enable real-time tracking, location identification, stock assembly process management, automated reporting, and invoicing. By incorporating computer services like **maintenance, system design, software development, software installation, and disaster recovery**, the system will provide an **integrated IT management** solution.

The project seeks to **streamline stock operations, reduce inefficiencies, enhance inventory tracking accuracy, and optimize warehouse workflow** through a **scalable and secure web application**.

**1.2 Objectives**

To achieve the project aims, the following objectives have been identified:

1. Develop a **centralized stock management system** to track real-time stock levels and prevent overstocking or shortages.
2. Implement a **stock movement and location tracking module** to monitor item locations during the stock assembly process.
3. Integrate **automated alerts and notifications** for low stock levels to improve warehouse efficiency.
4. Develop **role-based authentication and access control** to enhance security and restrict access to authorized users.
5. Design and implement **reporting tools** that generate real-time inventory reports, stock movement history, and sales analytics.
6. Develop an **invoice management module** to create, store, and retrieve invoices for sales and purchases.
7. Incorporate **computer services**, including maintenance, system design, software development, software installation, and disaster recovery management.
8. Use **Python for backend development** (Flask/Django) and **TypeScript for frontend** (Angular/React) to build a responsive and efficient web application.
9. Ensure **data security and backup solutions** to prevent data loss and unauthorized access.
10. Develop a **requirement evaluation plan** to assess system performance and usability.
11. Create a **project plan with risk assessment and mitigation strategies** to ensure timely completion and successful deployment.

# **Requirements**

The system requirements are categorized into **essential, recommended, and optional** to clearly define the scope of development.

List of requirements, including high-level requirements or aims, and detailed requirements or objectives. You should separate these into essential, recommended and optional.  
  
**2.1 Essential Requirements**

1. A **centralized inventory management system** to track stock levels and manage product categories.
2. **Real-time stock movement and location tracking** to monitor items in the warehouse.
3. **Automated low-stock alerts and notifications** to optimize replenishment processes.
4. **Role-based authentication and access control** for user security.
5. **Reporting and data visualization tools** for stock analysis and forecasting.
6. **Invoice management system** for sales and purchase transactions.
7. **Secure database storage and backup mechanisms** for data protection.
8. **Integration of computer services** such as maintenance, system design, software development, installation, and recovery.

**2.2 Recommended Requirements**

1. **Barcode/QR code scanning** for seamless stock tracking.
2. **AI-driven demand forecasting** for better inventory planning.
3. **Warehouse mapping and visualization tools** for efficient storage optimization.
4. **Cloud-based deployment** for remote access.
5. **Integration with third-party accounting and ERP systems** to streamline financial management.

**2.3 Optional Requirements**

1. **Mobile application** for on-the-go stock management.
2. **Voice-command-based stock retrieval system** for hands-free operation.
3. **AI-powered chatbot assistance** for customer support.
4. **Multi-language support** to enhance accessibility.

# **Technical Specification**

Technical specifications of the project.

**3.1 Backend (Python & Flask/Django)**

* **Python** will be used for backend development due to its robust ecosystem and security features.
* **Flask or Django** will be selected based on project needs. Flask offers lightweight flexibility, while Django provides a structured framework.
* **PostgreSQL** for data storage and **SQLAlchemy/Django ORM** for database operations.
* **RESTful API** implementation to enable communication between frontend and backend.
* **Celery & Redis** for task scheduling (e.g., automated low-stock alerts).

**3.2 Frontend (TypeScript & Angular/React)**

* **TypeScript** will be used to ensure scalable and maintainable frontend code.
* **Angular or React** for frontend framework, allowing dynamic UI interactions.**Bootstrap/Tailwind CSS** for responsive design.  
    
  **3.3 Security & Authentication**
* **JWT Authentication** for user access control.
* **bcrypt** for password hashing.
* **OAuth2** for third-party authentication integration (Google login, etc.).
* **3.4 Reporting & Invoicing**
* **Pandas & Matplotlib** for stock reports and analytics.
* **ReportLab / FPDF** for generating invoices.
* **3.5 Deployment & Hosting**
* **Docker** for containerization.
* **AWS/GCP/Azure** for cloud deployment.
* **Nginx + Gunicorn** for server deployment.

# **Requirements Evaluation Plan**

What criteria to use to evaluate your system or experimental results? Who will be involved in the evaluation? What type of testing will you do to verify the quality of your software? In short, how will you verify that your project achieves what it sets out to do?  
  
The evaluation plan will assess the system based on the following criteria:

* **Functionality Testing:** Ensure that all core functionalities (stock tracking, reporting, invoicing) operate correctly.
* **Performance Testing:** Evaluate system efficiency in handling stock updates and report generation.
* **Usability Testing:** Gather feedback from potential users to improve UI/UX.
* **Security Testing:** Verify authentication and data encryption mechanisms.
* **Scalability Testing:** Assess the system’s capability to handle increased stock levels and user interactions.

# **Background Research and Reading list**

Background material (including a reading list and literature review where appropriate);  
  
**5.1 Background Research**

* **Warehouse Management Systems (WMS)** and their role in inventory tracking.
* **Best practices in inventory control and supply chain management.**
* **Modern web technologies for building scalable stock management systems.**

**5.2 Reading List**

1. “Designing Data-Intensive Applications” – Martin Kleppmann.
2. “Flask Web Development” – Miguel Grinberg.
3. “Python for Data Analysis” – Wes McKinney.
4. “JavaScript and TypeScript: Web Development Guide” – Ethan Brown.
5. Research papers on **inventory control techniques** and **warehouse management systems**.

# **Time-plan and Risk Plan**

A detailed timetable and plan for achieving the objectives of the project (this could be tabulated or in the form of a Gantt chart), including the milestones of the project and a risk plan; Explain target dates and amount of time required for the completion of aspects of the project  
  
**6.1 Project Plan (Gantt Chart)**

A detailed Gantt chart will be created to track progress in **project phases**:

* **Phase 1 (Week 1-2):** Requirement gathering and system architecture design.
* **Phase 2 (Week 3-4):** Backend and database implementation.
* **Phase 3 (Week 5-6):** Frontend UI development.
* **Phase 4 (Week 7-8):** System integration and testing.
* **Phase 5 (Week 9-10):** Final evaluation and deployment.

A graph with red and orange squares

AI-generated content may be incorrect.

**6.2 Risk Assessment & Mitigation**

| **Risk** | **Impact** | **Mitigation Strategy** |
| --- | --- | --- |
| Security vulnerabilities | High | Implement robust authentication & encryption. |
| Database failures | High | Implement automated backups & failover solutions. |
| Scope creep | Medium | Stick to essential requirements; follow Agile principles. |
| Performance issues | Medium | Optimize queries and implement caching mechanisms. |

**7. Overall Impression**

This project will deliver a **robust, scalable, and efficient Stock Control System** that improves warehouse inventory management. By leveraging **Python, TypeScript, Flask/Django, Angular/React**, and modern APIs, the system will provide **real-time tracking, invoicing, and reporting** while maintaining security and performance. The project plan ensures timely completion, while the risk plan mitigates potential challenges.

This report clearly defines the project objectives, technical stack, requirements, and a well-structured execution plan, ensuring successful project delivery.

# **References**

The reference list should contain a mixture of books, research papers (if appropriate) and internet resources.