National College of Ireland

HDip Sc in Computing

2023/2024

Sérgio Vinício da Silva Oliveira

x23170981

x23170981@student.ncirl.ie

AluControl

Technical Report



**Table of Contents**

[Executive Summary 3](#_Toc351559315)

[1 Introduction 3](#_Toc351559316)

[1.1 Background 3](#_Toc351559317)

[1.2 Aims 3](#_Toc351559318)

[1.3 Technologies 3](#_Toc351559319)

[1.4 Structure 3](#_Toc351559320)

[2 System 3](#_Toc351559321)

[2.1 Requirements 3](#_Toc351559322)

[2.1.1 Functional requirements 3](#_Toc351559323)

[2.1.2 Data requirements 3](#_Toc351559324)

[2.1.3 User requirements 3](#_Toc351559325)

[2.1.4 Environmental requirements 3](#_Toc351559326)

[2.1.5 Usability requirements 3](#_Toc351559327)

[2.2 Design and Architecture 3](#_Toc351559328)

[2.3 Implementation 3](#_Toc351559329)

[2.4 Testing 3](#_Toc351559330)

[2.5 Graphical User Interface (GUI) Layout 3](#_Toc351559331)

[2.6 Customer testing 3](#_Toc351559332)

[2.7 Evaluation 3](#_Toc351559333)

[3 Conclusions 3](#_Toc351559334)

[4 Further development or research 3](#_Toc351559335)

[5 References 3](#_Toc351559336)

[6 Appendix 3](#_Toc351559337)

[6.1 Project Proposal 3](#_Toc351559338)

[6.2 Project Plan 3](#_Toc351559339)

[6.3 Requirement Specification 3](#_Toc351559340)

[6.4 Monthly Journal 3](#_Toc351559341)

[6.5 Other Material Used 3](#_Toc351559342)

# Executive Summary

Maximum 300 words. The abstract should mention the problem being addressed, describe the technical solution and briefly report the findings of the evaluation.

# Introduction

## Background

A report by Grand View Research (2024) predicts that the global market for enterprise resource planning (ERP) software will grow by 11.0% annually between 2023 and 2030. This growth is driven by companies' need to enhance operational efficiency and transparency in business processes, coupled with the rise of mobile and cloud applications.

The implementation of ERP within an organization has the potential to decrease inventory costs by 25% to 30% and raw material costs by approximately 15% (Grand View Research, 2024). Furthermore, it can significantly increase profitability, resulting in an increase in the overall demand for ERP solutions.

Cloud-based solutions are becoming more popular due to their flexibility, scalability and cost savings, crucial features for small and medium enterprises (SMEs) operating with limited resources. Despite the positive outlook, there are obstacles such as the high initial investment and the complexity of integrating ERP systems with existing infrastructures. Nevertheless, the benefits of integrating data management, accessing real-time information and simplifying processes make ERP solutions a strategic investment for SMEs that want to compete in an increasingly digital market, moving towards modernization and making strategic decisions to ensure sustainable growth and success in the future.

AluControl, the solution developed in this project, aims to enhance operational efficiency and decrease costs, providing real-time reporting and analytics capabilities that meet the growing demand for affordable digital solutions. The cloud deployment of this solution provides modern SMEs with the scalability and flexibility they need to compete in today's digital marketplace.

## Aims

The main purpose of this project is to produce a centralized business automation system that enables users to manage their sales and inventory operations efficiently. To be more specific, the AluControl application is meant to do:

* **Manage Inventory**: Make tracking and inventory management easier with automation.
* **Sales and Rental Reports:** Deliver detailed reports on sales and rental operations, allowing for a better comprehension of the business's performance.
* **Dashboards** Provide data visualization tools, like charts and dashboards, to keep track of performance metrics in real-time.
* **Automate Business Processes:** Minimize the use of manual processes, avoid errors, and improve operational efficiency.
* **Increase productivity:** Eliminate repetitive tasks to focus on activities that bring more value to the business.
* **Facilitate the Decision-Making:** Offer access to cutting-edge analytics and real-time reporting that can assist with strategic decision making.
* **Financial Reports:** Provide detailed reports of expenses, providing a clearer understanding of the company's financial condition.
* **Cash Flow Control:** Ensure more effective financial control by managing the company's cash flow, which includes paid and receivable items.

Although AluControl's current version has the mentioned features, there are plans to expand and enhance the system in future versions. Future goals include:

* **Optimization of Stock Levels:** Apply a sophisticated algorithms to maintain precise stock levels of products and prevent shortages and overstocks.
* **Expansion of Financial Control:** Integrate additional functionalities for detailed management of revenues and expenses, as well as more comprehensive financial reports.

## Technologies

The AluControl is supported by a robust technology set that ensures efficient and scalable development and deployment.

**Frontend Development**

* **HTML** and **CSS**: The frontend interface of AluControl is built with HTML for structure and CSS for styling. Moreover, Bootstrap features are utilized to guarantee a responsive and visually appealing design.
* **JavaScript:** JavaScript is employed to enable real-time interaction with backend services, significantly enhancing the user experience. By allowing updates to specific parts of the page without a full reload, JavaScript uses dynamic interface behavior and AJAX requests to facilitate asynchronous data exchanges. The user interface is faster and more responsive, resulting in a seamless and efficient experience for users when they interact with the system.

**Backend Development**

* **Java**: The core application is built on Java, which makes it ideal for business applications and has been the foundation of this course.
* **Spring Boot:** This framework is employed to make configuration and backend development easier. It facilitates the creation of standalone applications that can run with minimal configuration.
* **Maven:** Maven is utilized for managing projects and dependencies, and also facilitates the integration of necessary libraries and plugins.
* **Jakarta:** The creation of entities and business objects was made possible by using this API to build the project models.
* **JPA (Java Persistence API):** This tool was applied to create and manage tables in the database.
* **RESTful API**: The backend services were created using the RESTful API to facilitate communication and integration with the frontend. This enable system components to communicate effectively and efficiently, enabling operations like data creation, reading, updating, and deletion.

**Hosting**

* **Heroku:** The application is hosted on the Heroku platform, which provides a reliable and scalable cloud infrastructure. This approach simplifies the deployment and management process, enabling me to focus on developing functionalities instead of maintaining the infrastructure.

**Database**

* **MySQL:** The database was developed through the MySQL tool. The course provided a valuable guide that demonstrated the simplicity and efficiency of managing data, which is why it was selected and utilized during the test and construction phases of the system.
* **PostgreSQL:** Hosting and production services are no longer possible due to the non-availability of MySQL within Heroku, so the server hosts the PostgreSQL relational database. However the tool is known for its robustness and efficiency in data management, providing secure storage and quick recovery of information, essential for the commercial operations managed by AluControl.

## Structure

Brief overview of each chapter

# System

## Requirements

This section describes the requirements of the AluControl system. The requirements were initially defined in the requirements specification document and grew as the project was developed. The necessary changes to the requirements are explained.

### Functional requirements

Functional requirements specify what the system should do. It is crucial to make sure that AluControl meets the needs of users and project objectives.

* **Inventory Management**
  + Users should have the ability to add, edit, and remove items from the inventory via the system.
  + The system needs to monitor inventory levels and prevent sales or rentals from taking place when there is insufficient balance.
* **Customer Management**
  + The system should permit users to add and modify customers.
* **Sales and Rental Reports** 
  + The system should produce detailed reports on sales and rentals, permitting the selection of monthly and annual data.
* **Dashboards** 
  + It is important for the system to provide dashboards that display real-time graphical views of key performance metrics.
* **Process Automation**
  + The system must automate repetitive tasks to minimize manual errors and improve operational efficiency. Link sales to specific customers and product records.
* **Cash flow control**
  + The system must manage cash flow by recording items paid and receivable.
* **Expense Management**
  + The system should enable the user to record expenses related to the business.
* **Management of Sales and Rental**
  + Users must be given the ability to register and edit sales or rentals through the system.

### Data requirements

Data requirements are the guidelines for managing and storing data in the system.

* **Data Structure**
  + A relational database is the ideal solution to store inventory, sales, users, and expenses data in the system. Tables like Products, Customers, Sales, and Expenses are used to organize and relate data efficiently.
* **Data Recovery**
  + Regular backup of the data and testing of restoration procedures can ensure data recovery in case of failure.
* **Data Integrity** 
  + Appropriate validations and restrictions in the database will be implemented by the system to ensure data integrity.

### User requirements

The needs and expectations of end users of the system are specified in user requirements.

* **User Interface**
  + The system should provide an intuitive and responsive interface, allowing access to all types of users and devices.

### Environmental requirements

The environmental requirements describe the conditions in which the system is to operate.

* **Development Environment**
  + Tools like IntelliJ IDEA were employed for Java development and GitHub was used for version control while in development.
  + The system was developed using modern tools and frameworks such as Java, Spring Boot, and Maven.
* **Production Environment**
  + The system is deployed on a trusted cloud platform like Heroku, ensuring scalability and ease of management.
* **Universal Access**
  + The application was developed as a web application using HTML, CSS and JavaScript instead of a desktop application via JavaFX or Swing. This choice allows universal access to the application through web browsers, facilitating accessibility and eliminating the need for local installations.

### Usability requirements

The system must meet usability requirements to be user-friendly and accessible for users.

* **Accessibility**
  + The system should be accessible to users with different levels of technical skill.
* **Performance**
  + The system should be responsive, with fast load times and a fluid user experience.

## Design and Architecture

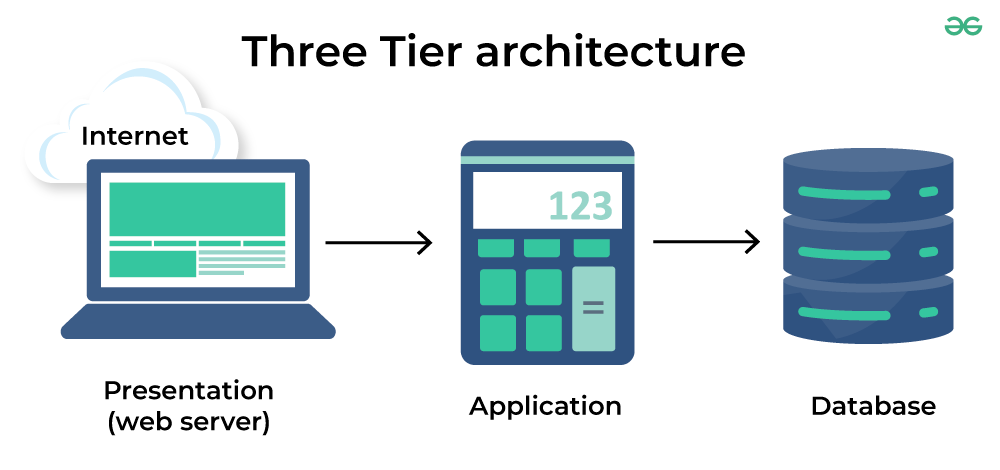
This section describes the design and architecture of the AluControl system, explaining how the system components integrate and work together to meet specified requirements.

Describe the design, system architecture and components used. Describe the main algorithms used in the project. (Note use standard mathematical notations if applicable).

An architecture diagram may be useful. In case of a distributed system, it may be useful to describe functions and/or data structures in each component separately.

### System Architecture

The AluControl system uses a layered architecture that distinctly divides the different software responsibilities, facilitating maintenance, scalability, and development. Among the various architectural approaches, the three-layer (3-tier) architecture, frontend, backend and database, stands out for its clarity in the separation of responsibilities, making a significant contribution to software development's modularity and flexibility.



<https://www.geeksforgeeks.org/three-tier-client-server-architecture-in-distributed-system/>

**Frontend – Presentation Layer:** The presentation layer is the frontend layer that handles the user interface. It consists of all the visual and interactive components that end users use to interact with the system.

**Backend - Business Logic Layer or Application:** This layer manages the system's business logic, implemented using Java, Spring Boot, Jakarta, JPA, and Maven.

**Database Layer:** Data storage and retrieval is handled by this layer, which utilizes MySQL for development and PostgreSQL for production.

#### Frontend - Presentation Layer

The presentation layer of AluControl is responsible for the user interface. By using Making use of technologies like HTML, CSS, and JavaScript, this layer ensures that the interface is intuitive, responsive and friendly. Responsibilities include:

* **Interactivity and User Experience (UX):** Build interfaces that are intuitive, responsive, and user-friendly.
* **Rendering and Data Manipulation:** Display data from the backend in a clear and organized manner, allowing users to view and manipulate information as needed.
* **Data Validation:** Perform initial checks on the user data before sending it to the backend.

The frontend’s contribution to software architecture is crucial, as it is the layer that defines the first impression of the user on the system. The success of this interface can significantly increase usability and user satisfaction, and this web approach promotes immediate access without the need for local installation. In addition, this structure facilitates the maintenance and evolution of the interface, allowing quick adaptations to changes in user needs or market.

#### Backend - Application Layer

The AluControl backend architecture is designed in layers to promote a clear separation of responsibilities and make it easier to maintain and scale the system. The main layers and components of the backend are described below:

A diagram of a service

Description automatically generated

This picture shows how Spring Boot works, like a map for how all its parts fit together.

Spring Boot Flow Architecture

<https://ezaouibiyassin.medium.com/spring-boot-architecture-annotations-e4f5dd5c7a66>

* **Controllers:** The purpose of this layer is to handle HTTP requests received from the frontend and send them to the correct services. Returning formatted responses to the frontend while creating, updating, reading, and generating statistics is your main responsibility.
* **Model:** They represent the system entities and are implemented with Jakarta Persistence API (JPA). Its main responsibility is to define the data structure and mapping between Java objects and database tables, as well as ensuring the persistence and recovery of database data.
* **Projections:** The purpose of this layer is to determine which fields of an entity or dataset should be included in the responses or queries. Providing specific and customized views of the data is your main responsibility, which promotes efficiency during database queries.
* **Repository:** Implemented with Spring Data JPA, the repositories manage access to data, offering methods for CRUD operations and complex queries.
* **Service:** Integrate the system's business logic and coordinate activities between controllers and repositories.

This layer, which is made up of Java, Spring Boot, Jakarta, JPA, and Maven, ensures system efficiency and robustness, manages business logic, and integrates seamlessly with the frontend and database.

#### Database

The AluControl database layer is responsible for managing data storage and retrieval. The system was originally created with MySQL, but it was moved to PostgreSQL for production because MySQL was unavailable on the Heroku platform. The responsibilities include:

* **Data Structure:** Store inventory, sales, rent, customers, and expenses data in a relational database.
* **Data Retrieval:** Performing routine backups and restore tests to guarantee data integrity.
* **Data Integrity:** Putting in place appropriate validations and restrictions in the database.

AluControl operations require this layer to ensure security and efficiency in managing business data.

The three-tier software architecture provides a clear separation of responsibilities, contributing to modularity, maintenance, and scalability of the system. AluControl's frontend is focused on user experience and utilizes technologies like HTML, CSS, and JavaScript to create intuitive and responsive interfaces. The backend, which is built on Java and Spring Boot, manages business logic and connects efficiently with the database. The data layer, which was initially based on MySQL and later shifted to PostgreSQL, is responsible for storing and managing crucial system data. Each layer plays a crucial role and works together to create a system that is cohesive and efficient. Understanding the responsibilities and contributions of each layer is critical to developing high-quality software that can meet the dynamic needs of users and the market.

## Implementation

Describe the main classes/functions used in the code. Consider to show and explain interesting code snippets where appropriate.

## Testing

Describe any testing tools, test plans and test specifications used in the project

## Graphical User Interface (GUI) Layout

Provide screenshots of key screens and explain.

## Customer testing

Provide evidence for and results of customer testing. This may include ratings or quotes from the customer.

## Evaluation

How was the system evaluated and what are the results? In many cases this will include usage data and user feedback. It may also include performance evaluations, scalability, correctness, etc. depending on the focus of the project.

Quantative results may be reported in tables or figures. Note that tables have their caption above the table and need to be cross referenced in the text (see **Error! Reference source not found.**). In many cases, tables are better to read if you skip the vertical lines.

Table 1: Performance with and without caching

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Nwithout** | **Nwith** | **Std.-Deviationwith** | **Std.-Deviationwithout** | **p** |
| Records | 100 | 200 | 2.54 | 3.97 | .002 |
| Data (GB) | 100 | 200 | 2.54 | 3.97 | .002 |
| Speed | 100 | 200 | 2.54 | 3.97 | .002 |

Figures have their caption below the figure as shown in **Error! Reference source not found.**. Make sure that if you use colour, the figure is still readable when printed in black & white, e.g., by using additional symbols, patterns, etc.



Figure 1: Learning gain across different experimental groups

# Conclusions

Describe the advantages/disadvantages, opportunities and limits of the project.

# Further development or research

With more resources, where could the results of this project lead to?

# References

It is recommended that students use the APA, Berkeley, Harvard or other internationally approved style. Here is an example of the APA citation style:

Wilcox, R. V. (1991). Shifting roles and synthetic women in Star Trek: The Next Generation. *Studies in Popular Culture, 13*(2), 53-65.

In the text this article can be cited as “Wilcox (1991)” or “(Wilkox, 1991)”.

References to web sites must include the access dates.

The library provides a study guide on Harvard style referencing.

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., & Stal, M. (1996). Pattern-Oriented Software Architecture: A System of Patterns. John Wiley & Sons.

Fowler, M. (2002). Patterns of Enterprise Application Architecture. Addison-Wesley Professional.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley.

Geeks for Geeks (2024) *Three-Tier Client Server Architecture in Distributed System.* Available at: <https://www.geeksforgeeks.org/three-tier-client-server-architecture-in-distributed-system/> [Accessed 26 Jul 2024].

Ezaouibi Yassin (2024) *Spring Boot Architecture & Annotations.* Available at: <https://ezaouibiyassin.medium.com/spring-boot-architecture-annotations-e4f5dd5c7a66> [Accessed 26 Jul 2024].

# Appendix

## Project Proposal

## Project Plan

## Requirement Specification

## Monthly Journal

## Other Material Used

Any other reference material used in the project for example evaluation surveys etc.

CD containing code should be glued to the technical report.