Final Year Project

FiberFlow: Textile PLM Digitalization

FYP Team

Saad Rehman (21F-9640)

Ali Shaikh (21F-9631)

Huzaifa Saqib (21F-9644)

Supervised by

Jawad Khalid Cheema

Department of Computer Science

National University of Computer and Emerging Science Chiniot, Pakistan

2024

Abstract

The Textile PLM Automation project aims to enhance the textile industry's product lifecycle management by integrating advanced AI, image processing, and cloud-based solutions. The current phase involves deploying the system on Google Cloud Platform (GCP) using Terraform, and developing algorithms for garment counting, fabric defect detection, and material quality assessment. We are also creating a marketplace platform to connect buyers and sellers and provide market analytics and billing services. So far, key infrastructure and workflow diagrams have been developed to guide the project's deployment and subsequent development phases.

Table of Contents

List of Figures

List of Tables

1. Introduction

The Textile PLM Automation project is focused on optimizing the textile production process by leveraging a centralized PLM (Product Lifecycle Management) system that integrates AI, IoT, and cloud computing. The system's goal is to improve efficiency, reduce manual labor, and provide a more transparent and connected marketplace for the textile industry. By enhancing the existing system with new functionalities like garment counting via cameras, fabric defect detection, and market analytics, the project aims to provide a robust solution that addresses current challenges and future demands of the industry.

2. Methodology

The project follows a structured methodology that is divided into distinct phases to ensure a systematic approach toward achieving the project goals:

1. **Deployment of the Existing Project on Google Cloud Platform (GCP):**
    - The first step involves deploying the existing Textile PLM Automation project on GCP using Terraform. This involves defining infrastructure as code (IaC) to automate the setup of resources such as Compute Engine instances, VPCs, subnets, firewall rules, and load balancers. The deployment ensures that the system is scalable, secure, and easy to manage.
2. **Development of New Functionalities:**
    - Following deployment, the next phase focuses on the development of new functionalities to enhance the existing Textile PLM system. This includes:
        - **Garment Counting via Camera Using Image Processing:** Developing an AI/ML model that uses CCTV cameras to count garments on a conveyor belt, improving upon existing photosensor-based methods.
        - **Fabric Quality Defect Detection via Camera:** Creating algorithms to detect fabric defects using high-grade cameras, providing a cost-effective and feasible solution.
        - **Material Quality Detection:** Implementing AI-based models to identify the type and quality of fabric and perform specific measurements.
3. **Creation of AI/ML Models to Achieve Goals:**
    - AI and ML models will be developed and trained using collected datasets to achieve the goals of garment counting, defect detection, and material quality assessment. These models will leverage image processing techniques to automate tasks traditionally performed manually, enhancing accuracy and efficiency.
4. **Marketplace Platform Development:**
    - The project will develop a marketplace platform that connects buyers and sellers, acting as a third-party intermediary. This platform will include real-time market data, billing and reminder systems, and analytics to reduce gaps and improve market transparency.

5. **Testing and Deployment in Real-world Settings:**
    - After development, the enhanced Textile PLM system and the marketplace platform will be deployed in real-world industrial settings. Feedback from these settings will be gathered to make necessary adjustments and improvements, ensuring the solution meets industry standards and user expectations.

3. Goals

1. **Garment Counting via Camera Using Image Processing:** Implement a camera-based system for counting garments on a conveyor belt, extending beyond photosensor and IoT-based methods to improve accuracy.
2. **Fabric Quality Defect Detection via Camera:** Develop an algorithm to detect fabric defects using high-grade cameras, offering a more affordable and regionally feasible solution compared to current options.
3. **Material Quality Detection:** Use AI and image processing to automatically identify fabric types (e.g., cotton, wool, Lycra) and specific measurements, automating tasks that are typically performed manually.
4. **Marketplace Bridging:** Build a platform to connect buyers and sellers in the textile market, acting as a third-party intermediary to reduce gaps and enhance transparency and trust.
5. **Market Price Analytics:** Provide users with accurate and current market pricing information based on collected data, assisting buyers in making informed decisions and preventing fraud.
6. **Billing and Reminders System:** Develop a software system for managing bills and sending payment reminders to ensure timely transactions, thus improving financial management in the textile industry.

4. Scope of the Project

Phase 1: Enhancement of the Existing Textile PLM System

This phase focuses on integrating new functionalities into the existing system, such as garment counting, fabric defect detection, and material quality assessment, by utilizing AI and image processing. These enhancements aim to replace manual methods with automated, scalable solutions that are both cost-effective and reliable.

Phase 2: Development of a Marketplace Platform and Analytics System

In this phase, a comprehensive marketplace platform will be created to connect buyers and sellers in the textile market. This platform will offer real-time market data, billing services, and analytical tools to streamline transactions and improve decision-making. The deployment of the enhanced Textile PLM system will take place in real-world industrial settings, with feedback being used to refine and optimize the platform.

5. Diagrams and Their Content

5.1 Activity Diagram: Terraform Deployment Workflow

- The activity diagram illustrates the deployment process that the client follows when using Terraform on Google Cloud Platform (GCP). The client starts by uploading application

files and running Terraform to automate the setup of the infrastructure. Depending on whether the Terraform setup succeeds, the client either proceeds to configure firewall rules, run the deployment, and monitor its status, or debugs the configuration and retries the setup. After deployment, the client monitors the results, and if the deployment is successful, they manage the resources. If the deployment fails, Terraform automatically rolls back the changes to maintain a clean environment. The process concludes with either a successful deployment or rollback.

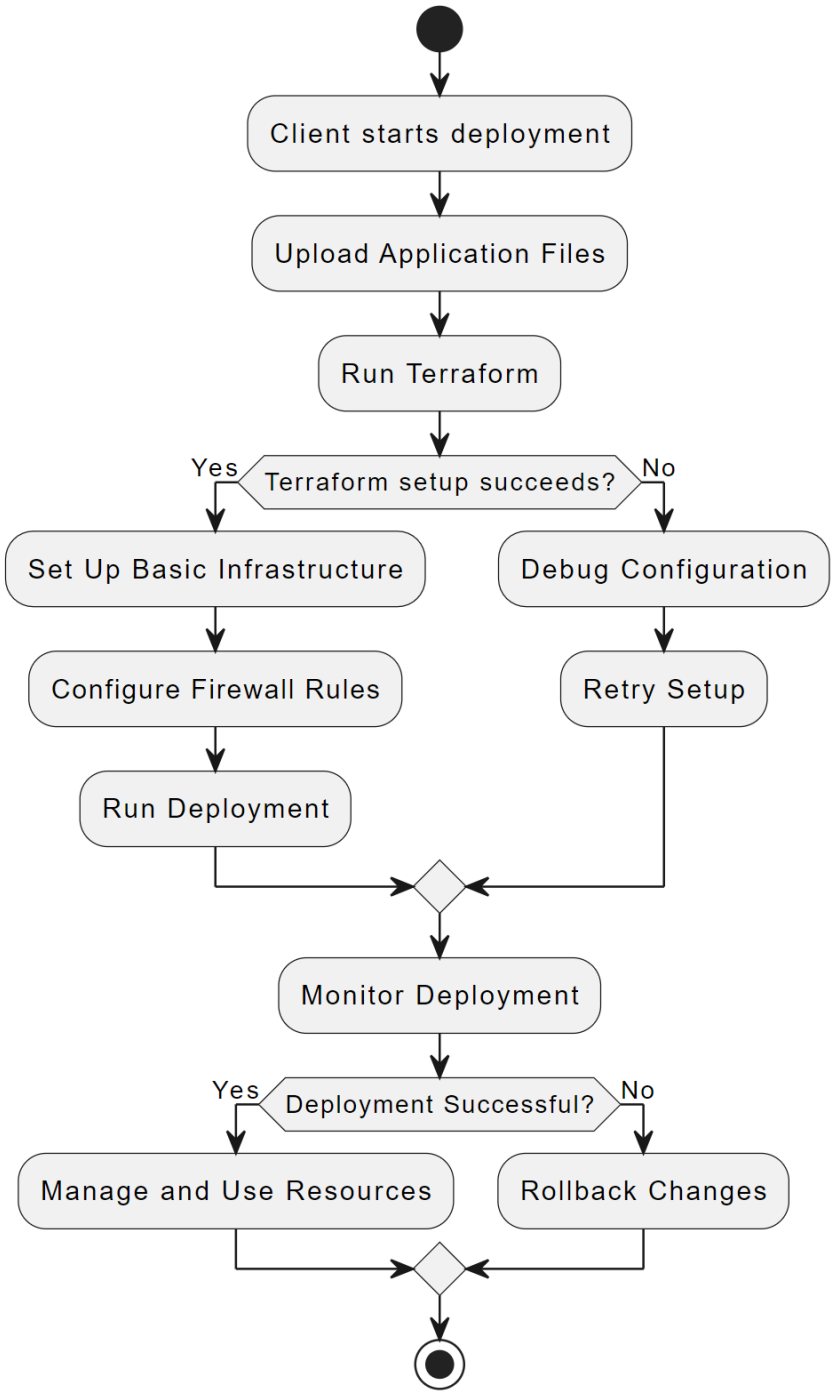

Figure 1: Activity Diagram

Step	Action/Decision	Description
1. Start	Action	Client initiates the deployment process.
2. Upload Application Files	Action	Client uploads necessary application files to GCP.

Step	Action/Decision	Description
3. Run Terraform	Action	Client runs Terraform to automate deployment.
4. Terraform setup succeeds?	Decision	Checks if Terraform setup is successful.
5a. Set Up Infrastructure	Action (Yes)	Terraform provisions infrastructure if setup is successful.
5b. Debug Configuration	Action (No)	If setup fails, the client debugs and retries the setup.
6. Configure Firewall Rules	Action	Terraform configures security settings like firewall rules.
7. Run Deployment	Action	Terraform applies configurations and runs the deployment.
8. Monitor Deployment	Action	Client monitors the deployment for errors or issues.
9. Deployment Successful?	Decision	Checks if the deployment was successful.
10a. Manage Resources	Action (Yes)	Client manages and uses deployed resources if successful.
10b. Rollback Changes	Action (No)	Terraform rolls back changes if deployment fails.
11. End	Action	The deployment process ends after success or rollback.

Table 1: Explanation of Activity Diagram

5.2 Use Case Diagram: Textile PLM Automation Deployment on GCP

- The use case diagram outlines the interactions between the client and the Google Cloud Platform (GCP) during the deployment process. The client starts by uploading the application files, followed by setting up the basic infrastructure. The client then configures the security by managing firewall rules, runs the deployment, and monitors the deployment's status to ensure it's successful. Each use case in the diagram demonstrates the key actions required to deploy the project on GCP, showcasing a streamlined process for efficient and automated infrastructure setup.

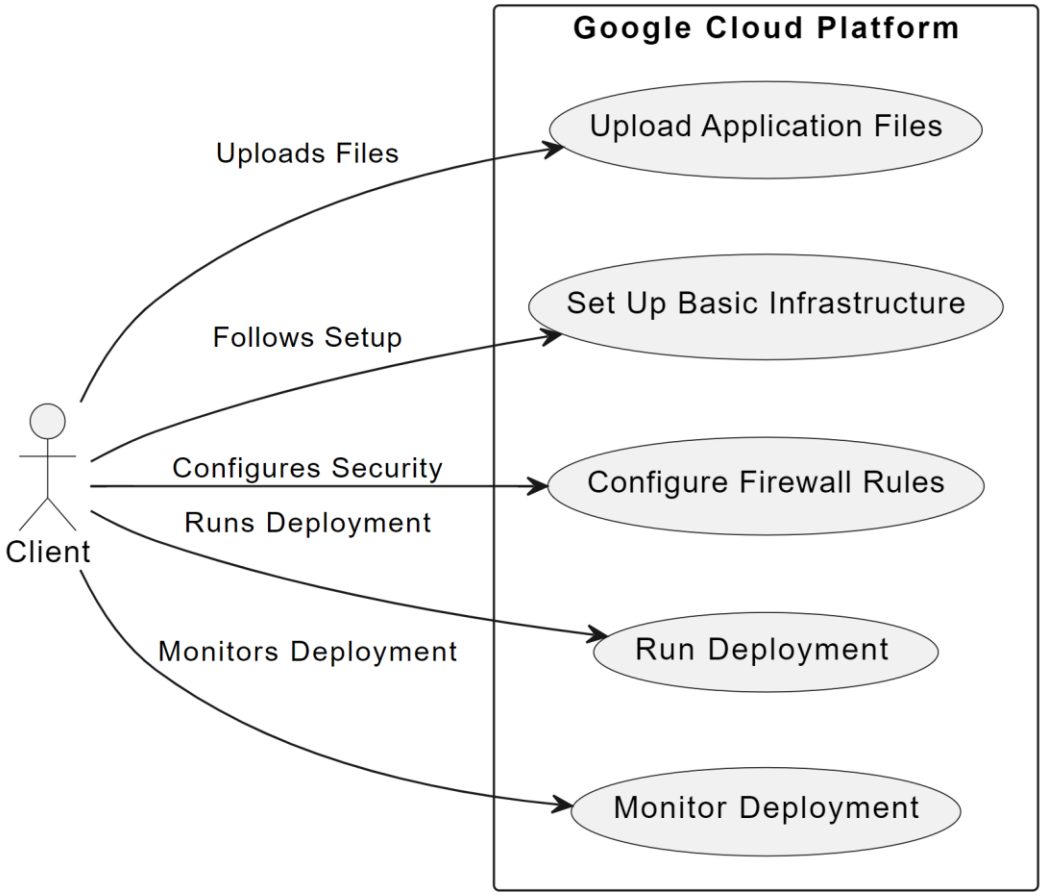

Figure 2: Use Case Diagram

Client Actions and Use Cases

Actor	Use Case	Description
Client	Upload Application Files	The client uploads project files to Google Cloud for deployment.
Client	Set Up Basic Infrastructure	The client uses the provided Terraform code to automatically set up infrastructure such as VPC, subnets, and virtual machines.
Client	Configure Firewall Rules	Optionally configures firewall rules to allow specific traffic like HTTP/HTTPS.
Client	Run Deployment	The client applies the Terraform configuration to deploy the resources.

Actor	Use Case	Description
Client	Monitor Deployment	The client monitors the deployment using GCP's monitoring tools, ensuring the application runs smoothly.

Table 2: Client Actions and Use Cases

Terraform Commands and Use Cases

Terraform Command	Use Case	Description
terraform init	Initialize Environment	Prepares the working directory to download the required GCP plugins and dependencies.
terraform plan	Preview Deployment	Generates and displays the execution plan showing what changes will be made.
terraform apply	Run Deployment	Applies the Terraform code and deploys the infrastructure on GCP.
terraform destroy	Rollback Changes	Removes all resources created by Terraform and destroys the infrastructure.

Table 3: Terraform Commands and Use Cases

6. Gantt Chart:

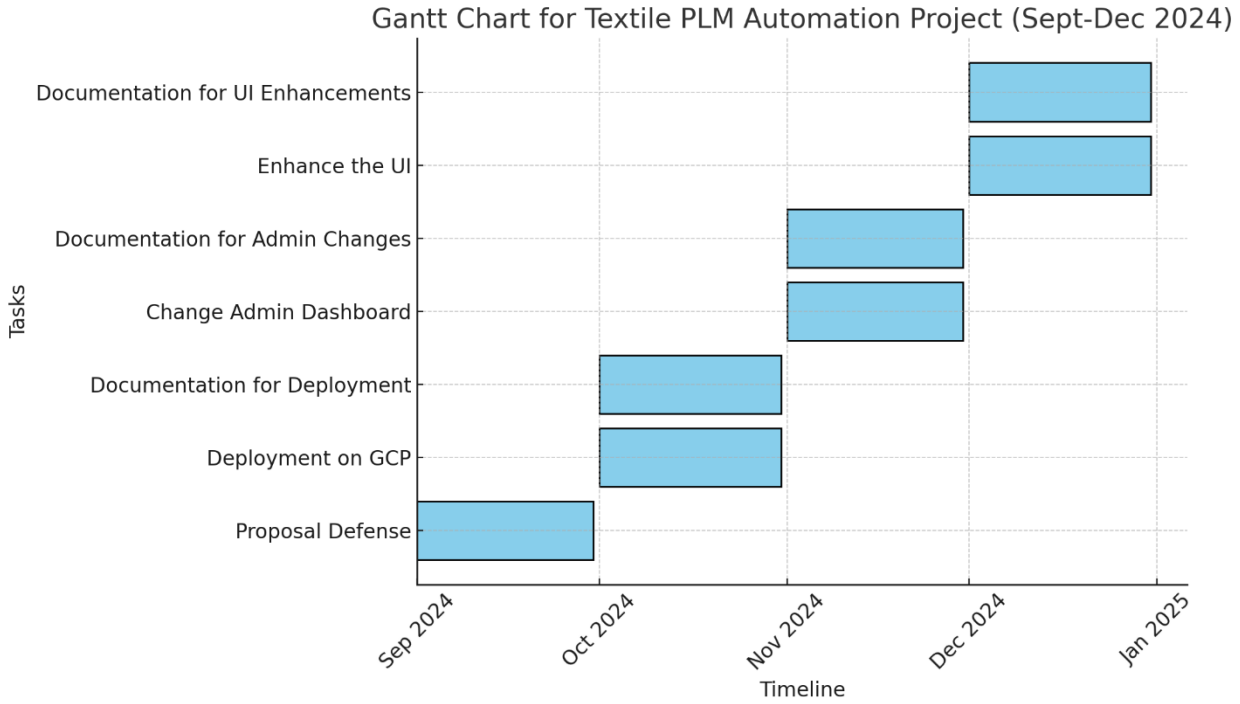

Figure 3: Gantt Chart