IMPRESS High Level Documentation

HOHENSTEIN

Bönnigheim, Germany

Illia Rohalskyi

August 2023

Contents

Chapter 1

Introduction

1.1   Purpose of the Document

The purpose of this document is to give a reader an overview of a project, it's structure and functionalities. It serves as a comprehensive yet concise document that could be read to understand the project scope, architecture and other essential details without delving into intricate technical details.

1.2   Scope

The scope of this document encompasses the entire "IMPRESS" project including its objectives, features, architecture, deployment strategy, and testing approach. The document does not delve into detailed technical implementation but provides an overview to help a reader understand the project's essence.

1.3   Definitions

· **IMPRESS:** The name of the project being documented.

· **System Architecture:** The high-level structure and organization of the project's components and modules.

· **Deployment Strategy:** The plan for deploying the project in various environments.

· **Testing Strategy:** The strategy for testing the project's functionalities and ensuring quality.

· **Use Cases:** Scenarios that describe how users interact with the project and achieve their goals.

· **Milestones:** Key points in the project timeline that mark significant achievements or progress.

· **References:** External resources and materials used as references during project planning and design.

· **Component:** A modular and self-contained unit of the system that has specific functionalities. Components can interact with each other to achieve higher-level features.

· **Continuous Integration (CI):** A software development practice that involves regularly integrating code changes into a shared repository. CI aims to detect and resolve integration issues early in the development cycle.

· **Continuous Deployment (CD):** A software development practice where changes to code are automatically built, tested, and deployed to production environments. CD aims to reduce manual intervention, minimize deployment delays, and ensure that new features and bug fixes are quickly delivered to end-users.

· **Sprint:** A timeboxed period (usually 1 to 4 weeks) in an Agile development cycle during which a team works on a set of planned tasks. Sprints are designed to deliver specific features or improvements.

· **Agile:** A flexible and iterative approach to software development that emphasizes collaboration, adaptability, and customer feedback. Agile methodologies prioritize delivering small increments of working software in short cycles.

· **Maintainability:** The measure of how easily a software system can be modified, updated, extended, or repaired over its lifecycle. A maintainable system is designed with clear and organized code, well-documented components, and modular architecture, making it more cost-effective and efficient to manage and evolve.

Chapter 2

General Description

2.1 Product Perspective

The project's primary focus is to develop a predictive model for surface tension in textiles, which holds significant importance for the textile manufacturing industry. The textile industry is a multifaceted sector that encompasses the production of a wide range of fabrics, materials, and products used in various applications. It involves processes such as dyeing, finishing, printing, coating, and treatment, each of which is influenced by surface tension.

2.2 Tools Used

2.3 General Constraints

2.4 Assumptions

Chapter 3

System Architecture

3.1 System Overview

3.2 Components and Modules

Chapter 4

Deployment

4.1   Deployment Diagram

4.2   Hardware and Software Requirements

Chapter 5

Testing Strategy

5.1 Test Plan

5.2 Testing Scenarios

Chapter 6

Use Cases

6.1 Use Case Diagram

6.2 Use Case Descriptions

Chapter 7

Project Timeline

7.1 Project Milestones

7.2 Timeline and Deliverables

Chapter 8

Retrospective

Chapter 9

References

9.1 External Resources

9.2 Documents and Materials Used