|  |
| --- |
|  |

|  |
| --- |
| REVISION HISTORY |

|  |  |  |  |
| --- | --- | --- | --- |
| **VERSION** | **DATE** | **DESCRIPTION** | **AUTHOR** |
| V1 | 21.09.25 | The first document revision for the class on 21st. | **Patrik Tamm** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| CONTENTS |

[1 SW System Overview 4](#_Toc208787546)

[1.1 Purpose 4](#_Toc208787547)

[1.2 Scope 4](#_Toc208787548)

[1.3 Use-Case Diagram 4](#_Toc208787549)

[1.4 General Constraints 4](#_Toc208787550)

[1.5 Assumptions and Dependencies 4](#_Toc208787551)

[1.6 Acronyms and Abbreviations 4](#_Toc208787552)

[2 SW Functional Requirements 5](#_Toc208787553)

[2.1 Features / Functions to be Implemented 5](#_Toc208787554)

[2.1 Acceptance Criteria 5](#_Toc208787555)

[2.2 Implementation Requirements 5](#_Toc208787556)

[3 SW Non-Functional Requirements 6](#_Toc208787557)

[3.1 Resource Consumption 6](#_Toc208787558)

[3.2 License Issues 6](#_Toc208787559)

[3.3 Coding Standard 6](#_Toc208787560)

[3.4 Modular Design 6](#_Toc208787561)

[3.5 Reliability 6](#_Toc208787562)

[3.6 Portability 6](#_Toc208787563)

[3.7 General Operational Guidelines 6](#_Toc208787564)

[4 SW Design Artifacts 7](#_Toc208787565)

[4.1 CRC Cards (Class–Responsibility–Collaboration) 7](#_Toc208787566)

[4.2 Conceptual UML Diagram (entities & relationships) 7](#_Toc208787567)

# SW System Overview

## Purpose

The purpose of the system is to create and enable the billing systems of a photo studio. The intended users are photo studios who need a system to manage orders and billing. It is being developed to reduce paperwork and reduce the time taken for administrative overhaul

## Scope

The system includes order management for photo printing and film developing, client information recording, automated pricing with express surcharges, consumable material tracking, and daily report generation for revenue and resource usage. The system excludes online customer portals, inventory procurement management, and complex scheduling features beyond basic order processing. Key benefits include a simple workflow between staff roles, accurate pricing calculations, elimination of paper-based processes, and automated reporting that reduces administrative overhead while improving operational transparency.

## Use-Case Diagram

|  |
| --- |
| *Provide a high-level UML use-case diagram showing main actors and their interactions with the system.* |

## General Constraints

*List technical and business constraints such as programming language, operating system, performance limitations, and standards*

The system is intended to be functional on both Windows and Ubuntu machines that are up to date and running hardware issued in the last 10 years. To fulfill said requirements, the program will be written in C++ 23 that can be run in most machines. Program will be CLI only.

## Assumptions and Dependencies

*State assumptions (e.g., availability of internet, supported devices) and dependencies (e.g., external APIs, hardware).*

The shop has a stable source of internet, electricity and a computer fulfilling the constraints,

## Acronyms and Abbreviations

*List all acronyms and abbreviations used in the document along with their explanations.*

|  |  |
| --- | --- |
| **Terms Used** | **Description of terms** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# SW Functional Requirements

## 2.1 Features / Functions to be Implemented

User Story 1: As a client, I want to place an order for photo printing or film developing so that I can get my photos processed by the studio.

User Story 2: As a receptionist, I want to record client details and completion times so that orders are properly tracked and scheduled.

User Story 3: As a receptionist, I want to calculate pricing with express surcharges so that clients pay the correct amount for urgent orders.

User Story 4: As a photographer, I want to receive order details and mark materials used so that I can process work efficiently and track consumables.

User Story 5: As an administrator, I want to generate daily reports on completed orders and material usage so that I can monitor studio performance and inventory.

User Story 6: As a receptionist, I want to process payments and mark orders complete so that clients receive their finished photos and the transaction is finalized.

Necessary components identified:

* An interface for receptionists to insert orders.
  + A form for the client
  + A for the photographer
* A method to identify express orders
* A way for the photographer to view orders
* An interface for the photographer to insert a report on the consumable
* A way for the administrator to view the used materials.

## Acceptance Criteria

*Define how each requirement will be validated: test cases, acceptance tests, or quality metrics.*

Requirements will be validated with test cases.

* A test to verify the validity of the forms.

## Implementation Requirements

*Provide details of specific implementation requirements if applicable. For example, integration with existing systems, supported platforms, or algorithms.*

# SW Non-Functional Requirements

## Resource Consumption

*Specify performance and resource limits (CPU, memory, storage, response time).*

## License Issues

*State licensing requirements and constraints on third-party software or libraries.*

## Coding Standard

*Define coding style and standards that must be followed.*

## Modular Design

*Specify architectural requirements such as modularity, extensibility, and maintainability.*

## Reliability

*Define requirements for reliability, error handling, and fault tolerance.*

## Portability

The application should run on Windows machines with Visual Studio installations present.

## General Operational Guidelines

*Provide guidelines for scalability, robustness, ease of use, and maintainability.*

# SW Design Artifacts

## CRC Cards (Class–Responsibility–Collaboration)

## The CRC cards can be found from the repository.

## Conceptual UML Diagram (entities & relationships)

*Draw a conceptual class diagram with key entities and their relationships; focus on nouns from User Stories/Use Cases, omit methods and low-level details.*