Functional Specification

for

Egma Ordering System

Version 1

Prepared by: Husain Raja

Reviewed by: Husni Alkhufash

Approved by: Husni Alkhufash

EGMA

SEE A BETTER WORLD

Date:22nd May 2018

Table of Contents

Revision History

Version	Date	Author	Reason For Changes	Reviewed By
V1.0	14/05/18	Husain Raja	First Draft	Husni Alkhufash

1. Introduction

1.1 Purpose (of document)

This document describes the functional requirements for Egma Ordering System used by Optical Stores to place the orders for the lens. The current system that is in place is provided by Rodenstock (German Lens Supplier) and built using C# and their legacy system.

The document is intended to provide potential suppliers with sufficient information to compile a tender document for the design and build of a new system for front end users (i.e. Stores).

1.2 Intended Audience and Primary Business Beneficiary

This document is aimed at external third party suppliers who are invited to tender for the design and build the front end of the Egma Ordering System. The backend is ready with APIs to be consumed.

The document is also intended for senior stakeholders within Egma plus all relevant associated business areas e.g. IT, Finance and Analytics. The main business areas to benefit are: Optical Stores.

2. Background & Context

EGMA is a full service optical industry solutions provider dedicated to serving the needs of the MENA region. With over 40 years of proven track record and heritage, Egma partner with optical businesses of all sizes to help them grow and improve their bottom lines and profits. Business from various fronts e.g. Manufacturing and Eye care, Egma is looking for achieving operational excellence by using Technological advancements for their direct and indirect customers.

2.1.1 ASIS Situation

Egma has two key systems i.e. Winfit and OWStudio (both provided by Optivision, Rodenstock) to manage ordering and production system.
The Winfit software is installed in every stores and enables stores to create and send the orders to factory. The OWS system is a real backbone which receives the orders from different stores and creates a job queue and a production job for the lens.

The winfit is created in .Net and C# and is a desktop software. This takes various inputs with numerous validations at frontend to make sure the correct order gets pushed. We'll not touch the OWS (backend) at this moment but the front end (Winfit) needs to be enhanced to provide more interactive and optimized user experience.

High Level Operational Workflow:

1. The customer requires a specific lens.
2. The order is created by optical store by entering the specifications of the lens in Winfit software.
3. The order is saved and sent to the OWS system.
4. The Winfit system generates the xml order file and pushes it to shared drive.
5. The directory watcher keeps checking the order xml file in the location and pushes it to OWS system for production.
6. The OWS further does the validation in line with the appropriate inputs and criteria.
7. The production card is printed for manufacturing the lens.
8. The stages are maintained in OWS and status is sent to Winfit for a specific order via .Net web service.

2.1.2 TOBE Situation

The ASIS situation described above for Stores is required to be replicated across all the areas within new system and cater for each variation of UI and UX.

The new platform will bring the following business benefits:

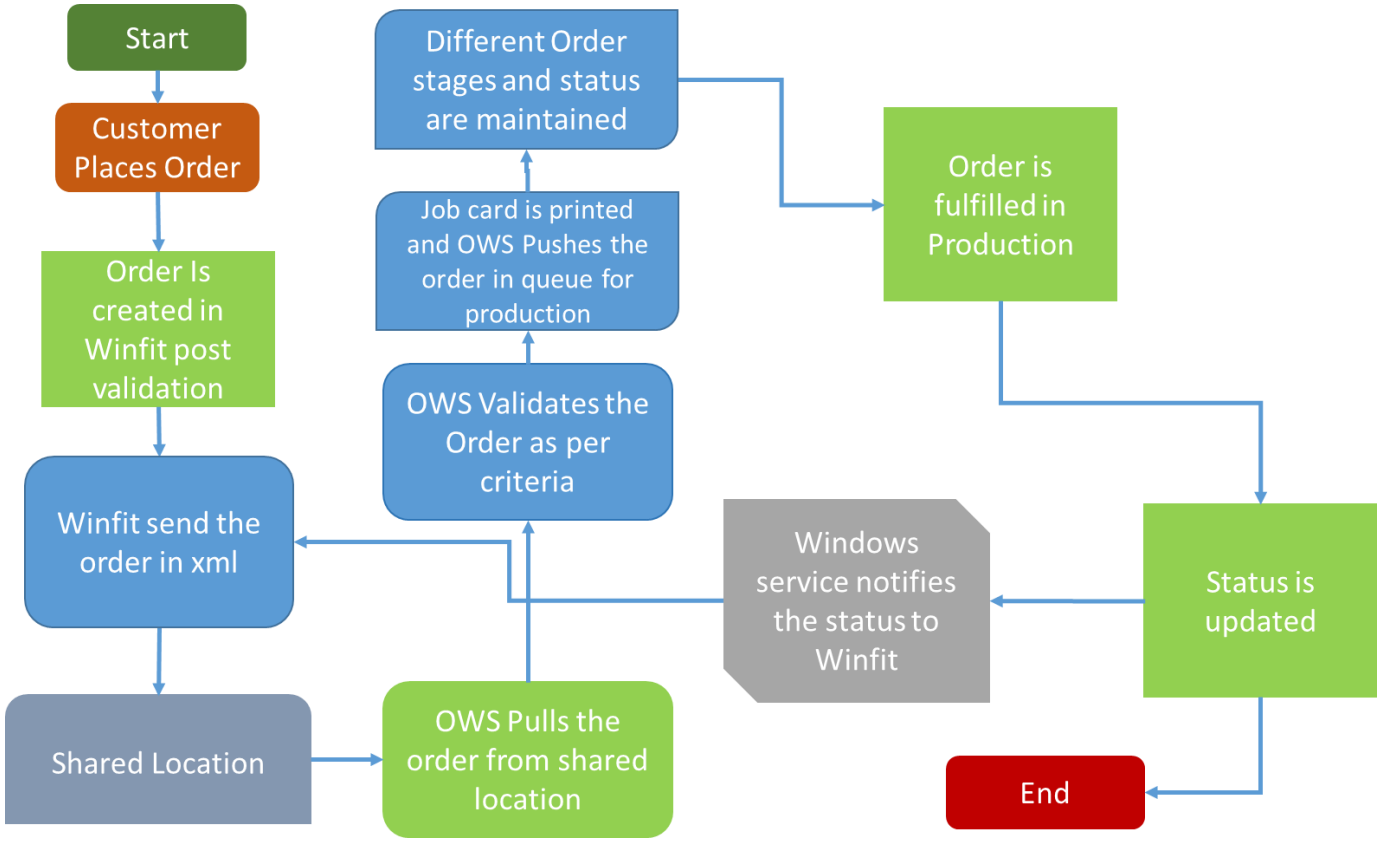
- Interactive, Intuitive and responsive UI experience to the stores as well as customers.
- Optimized usability with ease of order creation and managing for Stores.
- Reduced opportunity for error with automated error handling, validations and logging mechanism.

The Winfit requires to be replaced with new platform to provide exceptional user experience to stores as well as customers. The new system needs to be web based instead of a desktop application. The backend platform is already built using Python Django framework and the APIs are exposed to be consumed.
The frontend is required to be built with enhanced UI-UX. The frontend system will consume the APIs built in Python and xml will be generated just like Winfit to create order in OWS.

2.2 Additional Document References

- A functional workflow diagram of the current process.

3. Functional Workflow



3.1 Platform / System Functions

Winfit System Functionalities

1. New Order
    a. Lens – (Spectacle Lenses, Individual Parameters, EyeLT, Frame-and cetering data, Lens calculation)
        i. Edging
        ii. Glazing
2. Order Browser – Data Grid for each individual options.
    a. Print
        i. Fax
        ii. Framed
        iii. Order list
    b. Folders
        i. Open orders
        ii. Recycle bin
        iii. Archive
3. Options
    a. Settings
        i. Personal Data
        ii. Application Data
        iii. Printer
    b. Language

    c. Update area
    d. Service area
    e. Price calculation
4. Help
    a. Hotline
    b. Remote support
    c. Rodenstock info
        i. Measurement
        ii. Manual
    d. System information
5. Quit

The above workflow and functionalities are existing and needs to be adapted in new frontend with optimized and enhanced UX-UI experience. Several functionalities can be eliminated if they are not in use and if the APIs are not provided. Those that needs to be eliminated will be discussed later.

Additional Functionalities that are required:
1. NA

3.2 Operational Scenarios

Currently, the Winfit system is installed in every stores and inputs the order. Once the order is created and saved with all the validations passed; it is sent to Production (i.e. OWStudio). The order is sent in the form of xml file.
The Winfit is required to be replaced by Web based application which will intake the order to send it to OWStudio in the form of same xml file.

3.3 Main Users and How the System will be used

The main users / beneficiaries of the system will be stores sales/operation team.

3.4 Operating Environment & Constraints

Application would be deployed on Cloud preferably AWS in Wintel environment.

3.5 User Documentation

The user documentation requires to be done for the new application for its flow and steps.

3.6 Assumptions and Dependencies

1. The backend platform is ready and not required to be built.
2. Any modifications required in the backend platform will be done by Egma.
3. Only frontend web application requires to be built which will consume the APIs exposed from the Platform.
4. The current application is not required to be integrated and will be taken off once the new web application is in place.
5. There's no hardware device integration directly with the platform or any type of data migration.

4. System Features

1. Python backend with Restful APIs (Already created and Ready)
2. Enhanced UX-UI Experience with optimized steps and flow.
3. Web based application accessible without the need of installation.

4.1 Automation within system

Not applicable

4.2 MI & Reporting Requirements

Not applicable

5. Non-functional Requirements

5.1 System Requirements

Not applicable

5.2 Safety Requirements

Not applicable

5.3 Security Requirements

Access to the stores needs to be restricted to specific people.

5.4 Software Quality Attributes

The Platform represents a fundamental element of Egma's E2E business process. The ability to quickly manage order and submit it to production in addition of enhanced user and customer experience provides good opportunity to target multiple stores as well as customer base.

Key attributes of the Platform therefore are:

- Web based platform
- Low on maintenance
- Interactive and intuitive UI experience.

5.5 Business Rules

The data held and displayed will be governed by all current internal and external validations in place provided by Egma.

6. Other Requirements

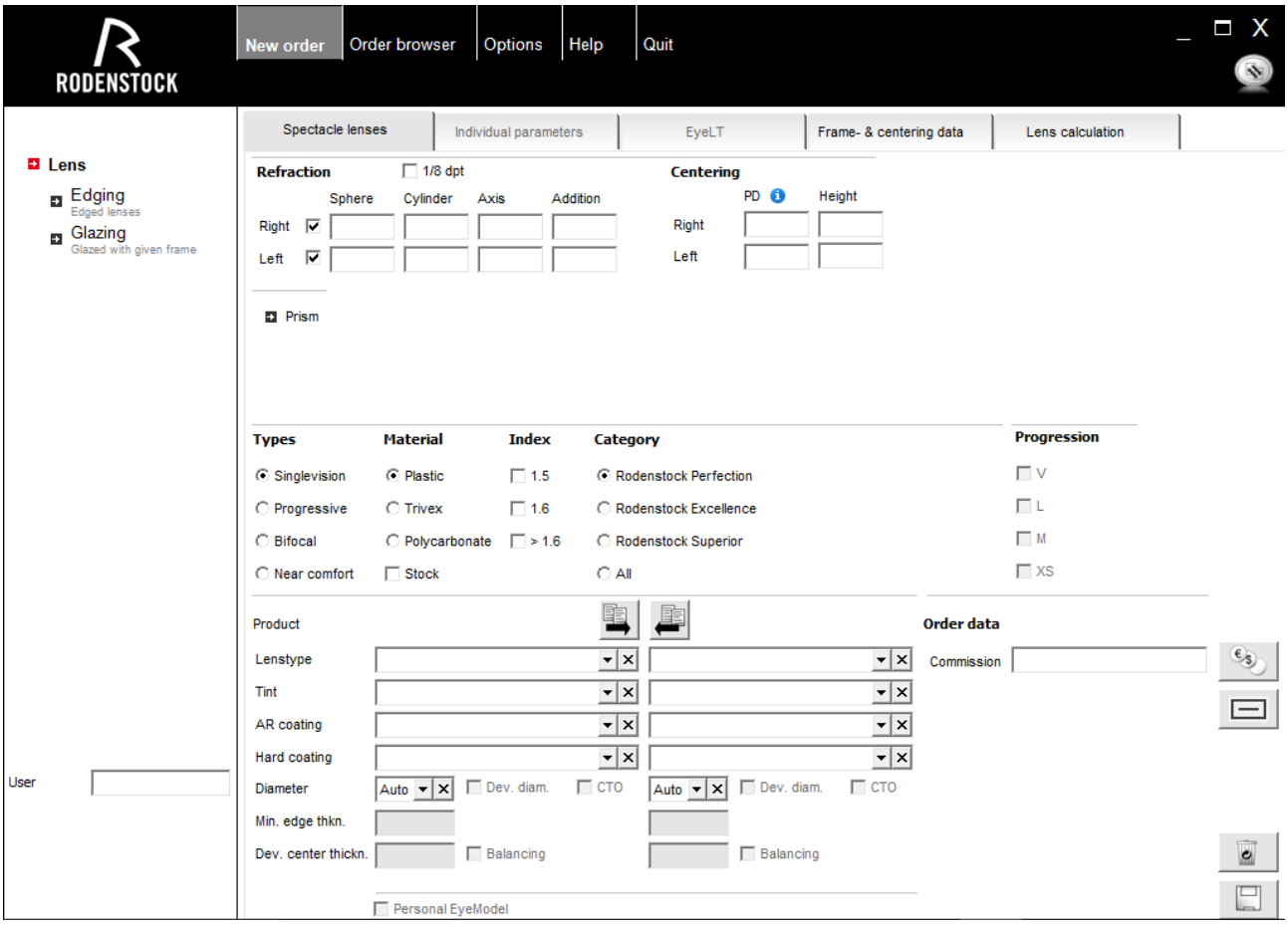
Platform support to be agreed.

7. Appendix

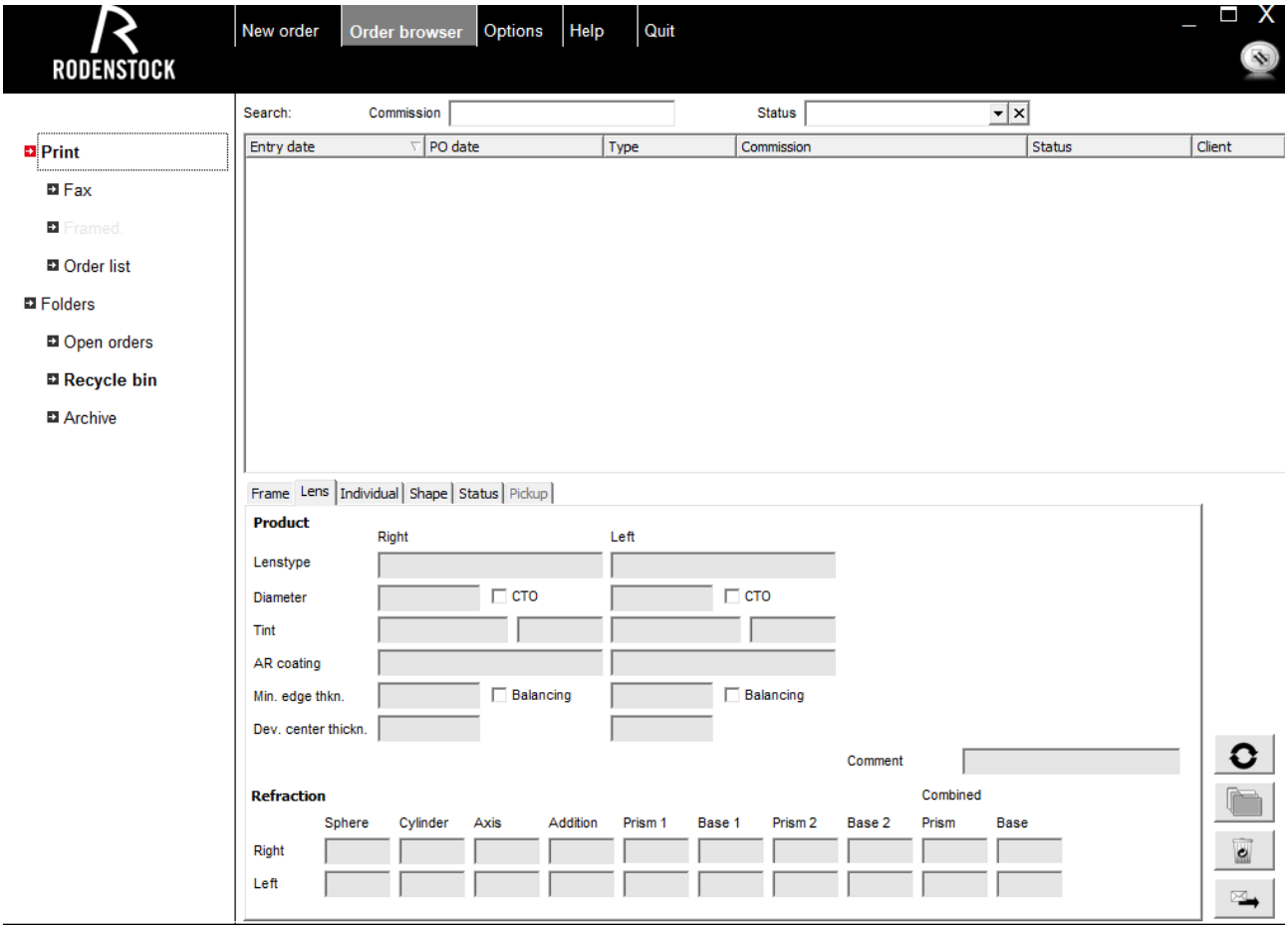
8.1 Current System

7.1.1 Winfit System

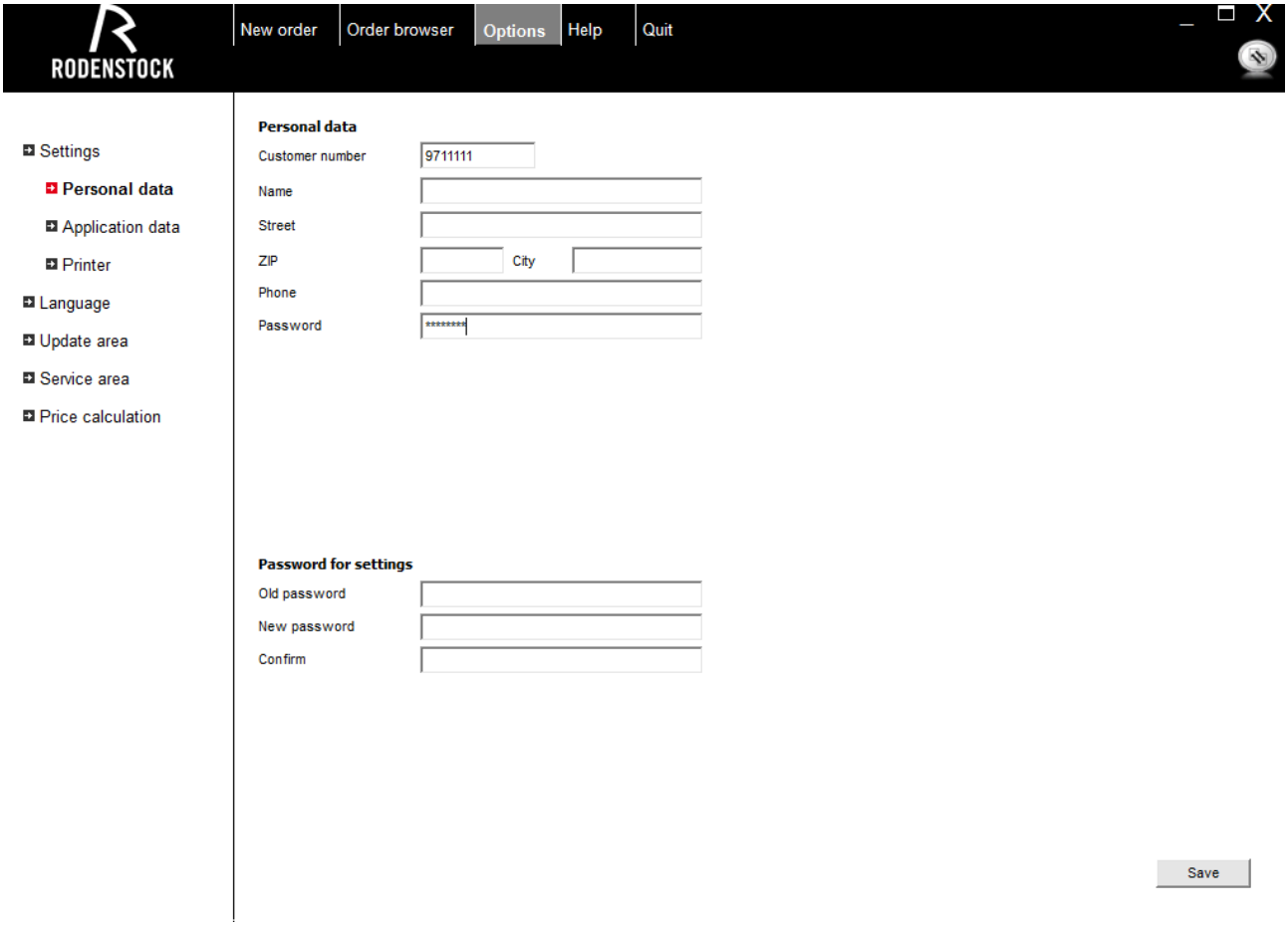
New Order:



Order Browser:



Options:



Help: