Requirements Document for
Order Entry Processing System

1. Introduction

A legacy Order Entry System needs to be renewed. The new system should receive the orders via the internet. Therefore, the current frontend components are to be replaced by a Web application implemented in Java. These include the online programs for displaying the articles on stock to the customers and for processing the orders. The old backend components in COBOL are to be taken over, wrapped and reused. These include the batch programs for:

- dispatching the ordered items
- billing the orders
- resupplying the articles on stock
- processing the back orders.

In addition, the following master files are to be migrated to relational databases:

- Customer data
- Supplier data
- Article data.

A new database for the current supplier prices is to be added.

&STARTME

2. Requirements

2.1 Functional Requirements

The following functional requirements are to be fulfilled.

FUNC-REQ-01 ArticleDisplay

The customer should have the possibility to scan the articles on stock and their prices. The articles should be ordered alphabetically and by category.

FUNC-REQ-02 OrderProcessing

The customer should have the possibility of selecting articles to be ordered. He must first enter his customer number. Then his identity and his credibility are checked. If they are ok, he may order up to 10 items at one time. If an ordered article is available on stock and in sufficient quantity, the order is to be accepted. If the article is not on stock the order is to be rejected. If the quantity on stock is too low, a back order is to be created.

FUNC-REQ-03 Dispatching

For every article item fulfilled a dispatch order should be generated and sent to the dispatch office. The ordered item should be sent to the customer within one day after it has been ordered. It is sent together with the order confirmation.

FUNC-REQ-04 Billing

For every customer order in which at least one item is fulfilled, an invoice is to be printed and sent to the customer. The invoices are to be prepared once a month at the end of the month. The invoice should include the items ordered, their prices, the total sum, the VAT and the amount due. All invoices should be checked by the accountant before being sent out.

FUNC-REQ-05 Resupplying

The system should monitor the articles on stock. If the quantity of an article on stock falls below the minimum quantity that article should be automatically reordered. The generated supply orders are to be collected and once a week, at the end of the week, they are to be processed. When a supply order is sent out the cheapest supplier at that moment in time should be selected. For this the current prices of all potential suppliers have to be checked.

FUNC-REQ-06 BackOrderProcessing

If an order cannot be fulfilled, it is put on hold to be processed later. Once a week at the end of the week the back orders are to be processed. If the order can be fulfilled, the article is dispatched, a bill prepared and the back order removed from the pool of back orders. If it is not fulfilled, it remains in the back order pool.

2.2 Non-Functional Requirements

The following non-functional requirements should be fulfilled:

NF-REQ-01 ResponseTime

The response time for a customer query should be <= 1 second. The response time for a customer order should be <= 3 seconds.

NF-REQ-02 TransactionCapacity

The system should be able to process at least 1000 orders an hour without any loss of performance.

NF-REQ-03 Availability

The system should be available 24 hours a day, 7 days a week, for at least 95% of the time.

NF-REQ-04 Security

Only authorized customers should have access to the system. Unauthorized persons are to be blocked from access. Of 10 attempts to break into the system by unauthorized persons, at least 9 should be thwarted.

NF-REQ-05 Recoverability

In case of a system crash, the system should be recovered within 30 minutes and all data restored to the state before the crash.

NF-REQ-06 DataSecurity

Customer and article data are to be secured in a backup database at the end of each working day. The backup data must be kept for at least 1 year.

NF-REQ-07 Usability

The system user interface must pass a usability test with a score of at least “0.6”.

NF-REQ-08 Maintainability

The system source code must have a maintainability index of at least “0.5” from the code audit.

NF-REQ-09 Reusability

The system source code must have a reusability rate of at least “0.6” from the code audit.

NF-REQ-10 Reliability

The system must have upon release a remaining error probability of less than “0.002”.

3. Functional Characteristics

3.1 Business-Objects

The system is composed of the following business objects:

BO-01: Customer

BO-02: Article

BO-03: Price

BO-04: Supplier

BO-05: Customer Order

BO-06: Order Item

BO-07: Dispatch Item

BO-08: Supply Order Item

BO-09: Supply Order

BO-10: Billing Item

BO-11: Invoice

BO-12: Back Order Item

BO-13: Dispatch Order.

3.2 Business-Rules

The following business rules apply to the order entry system:

BR-01: Unauthorized customers may not order articles.

BR-02: Customers with bad credibility may not order articles.

BR-03: Only articles on stock may be ordered.

BR-04: An order item can only be fulfilled if the amount of articles on stock is more than the minimum amount required.

BR-05: If the amount of an ordered article is insufficient, a back order item should be generated.

BR-06: If the amount of an article on stock falls below the minimum amount required, it should be automatically reordered.

BR-07: For each order of which at least one order item is fulfilled, a dispatch order should be sent to the dispatch office to send out the articles to the customer.

BR-08: For each supply order item, the supplier should be selected who offers the cheapest price at the moment.

BR-09: Invoices must be paid within 30 Tags after they have been sent out..

BR-10: The total amount due on an invoice is the sum of all billing items times the value added tax – VAT.

3.3 System-Actors

The new system should be used by five subjects, or system actors as well as the system itself:

Actor-01: Customer
Actor-02: Accountant
Actor-03: Dispatcher
Actor-04: Stock Clerk
Actor-05: Supplier
Actor-06: System.

Customers can query articles on stock and place orders at any time. The dispatcher receives the dispatch orders and dispatches the ordered articles to the customer. The accountant checks the invoices produced and sends them out to the customers. The stock clerk checks the supply orders and sends them out to the suppliers. The order administrator checks the back orders and sees that they are processed. The supplier receives the supply orders and delivers the articles ordered.

3.4 User-Interfaces

There should be a customer website with two web pages – one web page for informing the customers which articles are currently on stock and one web page for ordering the articles.

GUI-01i: Customer-Query-Page

With the customer query page the customer identifies himself and submits an article category.

- Customer number
- Article category.

GUI-01o: Article-List

In the Article-List the system lists out all articles of the selected category alphabetically by name with number, description and type.

- Article number
- Article name
- Article description
- Article price.

GUI-02i: Customer-order-Panel

With the customer order panel, the customer can submit an order with the following data:

- Customer number
- Customer name
- Order number
- Order items (1:n).

Each order item contains the following data:

- Order item number.
- Article number.
- Order Item Amount.

Every customer order is assigned an order number and an order date.

GUI-02o: Customer-order-Confirmation

With the customer order confirmation, either the order is confirmed or an error message is reported.

- Customer number.
- Order number.
- Confirmation message.
- Error message.

GUI-03: Report-Request-Page

With the report-request-page the user starts a backend process to either

- send out dispatches,
- to print invoices,
- to create supply orders,
- to process back-orders.

The contents of the page are:

date

Report type.

3.5 System-Reports:

The new system should provide the following reports:

Report-01: Dispatch order

OrderNumber
CustomerNumber
ArticleNumber
ArticleName
OrderAmount.

Report-02: Supply order

SupplyOrderNumber
SupplierNumber
ArticleNumber
ArticleName
SupplyAmount.

Report-03: Invoice

InvoiceNumber
CustomerNumber
CustomerName
CustomerAddress
OrderNumber
BillingItems(10)
    # BillingItemNumber

- # ArticleNumber
- # ArticleName
- # Ordermount.

Report-04: List of Back orders

- # OrderNumber
- # CustomerNumber
- # CustomerName
- # OrderItem(10)
  - # OrderItemNumber
  - # ArticleNumber
  - # ArticleName
  - # OrderAmount.

4. System Use Cases

The following use cases in the existing system should be taken over:

- Dispatch processing
- Supply order processing
- Billing
- Back order processing.

They are all implemented in COBOL.

The two new use cases are:

- Customer Query
- Customer Order processing.

They are to be implemented in Java.

UseCase-01: Customer_Query

The customer selects an article category and submits a query on the status of those articles in that category. The article database is read and all articles displayed. They are to be displayed in ascending order by article name with the article number, name and price.

Article Nr.	Article name	Article price

<999999> <Artikelname> <9999.99>

Attribute	Description
Label:	Customer_Query
Fulfill:	Func-Req-01.
Process:	Bo-01, Bo-02.
Input:	Gui-01i.
Output:	Gui-01o.
Implements	Br-01.
Trigger:	Menu_selection
Actor:	Customer
Frequency:	Continuous
PreConditions:	Customer is authorized. Articles are stored by category.
PostConditions:	Articles of the selected category are displayed by name in alphabetical order.
MainPath:	1) System displays article categories. 2) Customer selects a category. 3) System displays all articles in that category with their prices in ascending alphabetical order.
AlternatePath:	None
Exceptions:	System rejects customer if he is not authorized.
Rules:	Only authorized customers have access to the articles on stock.
Objects:	Article
Inherits:	Standard-display-function.
Uses:	XML-Creator
Extends:	Standard-query-function
Comments:	This is a read only access to the articles kept on stock by the vendor.

UseCase-02: Customer_Order_Processing

In processing the customer order the system first checks if the customer is authorized, that is, if he appears in the customer database. If he is authorized, his current credibility is checked. If the customer is authorized and his credibility is good, the customer order is processed. Otherwise the customer is rejected with a rejection message.

When processing the order, the system processes one order item at a time until all order items have been processed. First, the article ordered is checked whether or not it is still on stock. If not the customer is informed that the article is not available and the order item passed over. If the article is on stock, it should be checked if the

amount on stock is more than the amount ordered. If the amount on stock is less than the ordered amount a back order is to be created and the customer informed that his order will be fulfilled later. The back order should include:

- Order number
- Order date
- Customer number
- Article number
- Order amount.

Back orders are to be stored in a sequential file to be processed once a week by a new batch component. This component will have the same logic as the customer order processing component.

If the article amount on stock is enough to fulfill the order, the amount ordered is to be deducted from the amount on stock and a dispatch order item generated. The dispatch order item should include:

- Order number
- Customer number
- Article number
- Order amount

The dispatch orders are to be accumulated in an XML file and processed in batch mode at the end of each working day by an existing COBOL program.

Whenever the article amount on stock is reduced, it should be checked if the amount is still greater than the minimum article amount required. If it is not, then a supply order item should be created. This supply order item will contain:

- Article number and
- Supply amount.

The supply order items are be kept in an XML file to be processed later by an existing COBOL program in batch mode.

For every fulfilled order item a billing item should be created with the following data attributes:

- Order number from the customer order.
- Order date from the customer order.
- Customer number from the customer order.

- Article number from the order item.
- Order amount from the order item.
- Article price from the article data.
- Item price = Order amount x Article price.

The billing items are to be stored in an XML file and processed at the end of each month by the old billing program, which produces the bills. This component already exits as a COBOL batch program.

If anyone order item is not fulfilled, the remaining order items of that order are still to be processed. The use case is finished when all of the order items have been processed. Then the customer is given a confirmation of each order item fulfilled and an error message for each order item not fulfilled.

Attribute	**Description**
Label:	Customer_order_processing
Fulfill:	Func-Req-02, Func-Req-03, Func-Req-04, Func-Req-05.
Process:	Bo-01, Bo-02, Bo-05, Bo-06.
Input:	Gui-02i.
Output:	Gui-02o, Bo-06, Bo-07, Bo-08, Bo-12.
Implement:	Br-02, Br-03, Br-04, Br-05, Br-06, Br-07, Br-08.
Trigger:	Menu_Selection
Actor:	Customer
Frequency:	Continuous
PreConditions:	Ordered article must be on stock. Customer must be authorized to purchase articles. Customer must be credible.
PostConditions:	Article amount is reduced by the order and Dispatch order exits for each fulfilled order item and Billing item exits for each fulfilled order item Or back order exits for not fulfilled order item. A supply order item exits if the article amount falls below the minimum order amount required.
MainPath:	1) System displays customer order panel. 2) Customer submits his customer number. 3) System checks customer credibility. 4) Customer submits an article number and the amount desired. 5) System checks if the article is on stock. 6) System checks if the article amount on stock is enough. 7) System subtracts order amount from the article amount if article amount > order amount. 8) System generates a dispatch order item if article amount > order amount.

	9) System generates a billing item if article amount > order amount. 10) system generates a supply order item if article amount < minimum amount required.
AlternatePath:	11) System generates a back order if article amount <= order amount. 12) System reports a delay of delivery if article amount <= order amount.
Exceptions:	System rejects order if customer is not authorized or customer credibility is bad.
Rules:	Only authorized customers with good credit may order articles. The article amount on stock may never fall below the minimum amount required.
Objects:	Customer, customer order, order item, article, dispatch item, billing item, supply order item, back order item.
Inherits:	Standard-Ordering- process.
Uses:	XML-Creator
Extends:	GUI-Usage
Comments:	The new Java component reads the customer data and updates the article data online. It produces three XML files – dispatch items, billing items and supply order items - to be processed offline by existing COBOL programs.

UseCase-03: Dispatching

The dispatcher will start the dispatching process once every day. The process is a batch job running in the background. All pending dispatch orders are processed by customer order and enhanced with the customer data – customer name and address – to create a dispatch order list. This list is then sent to the warehouse where the articles are packed and sent to the customers together with the dispatch order.

Attribute	Description
Label:	Dispatching
Fulfill:	Func-Req-03.
Process:	
Input:	Gui-03, Bo-07.
Output:	Repo-01, BO-13.
Implement:	Br-07.
Trigger:	EndofDay
Actor:	Dispatcher
Frequency:	Daily
PreConditions:	Dispatch order items must be available.
PostConditions:	Dispatch orders are generated.
MainPath:	1) Dispatcher starts dispatch job.

	2) System sorts dispatch items by customer number. 3) System adds customer data to dispatch order. 4) System prints dispatch order
AlternatePath:	None
Exceptions:	System finds no customer to match a dispatch item.
Rules:	Dispatch orders are to be printed by customer.
Objects:	Dispatch items, dispatch orders, customers.
Inherits:	Standard-printing-function.
Uses:	COBOL-Wrapper
Extends:	Customer-order-processing.
Comments:	Dispatch orders are to be manually checked by the dispatcher prior to being sent to the warehouse.

UseCase-04: Supply_Order_Processing

Once a week the supply clerk will start the job for processing the supply orderst hat have accumulated during that week. The job is a batch process running in the background. First, the current supply order items are assigned to a supplier by selecting for each article the supplier with the cheapest price. For that, the price database has to be accessed. After assigning the cheapest suppliers, the supply order items are sorted by supplier number and for each supplier a supply order on a DIN-A4 report printed. The supply deadline is the current data plus 7 days. The supply orders are then sent out by email to the respective suppliers.

Attributes	Description
Label:	Supply_order_processing
Fulfill:	Func-Req-05
Process:	Bo-03, Bo-04.
Input:	Gui-03, Bo-08.
Output:	Repo-02, Bo-09.
Implement:	Br-08.
Trigger:	EndofWeek
Actor:	Stock clerk.
Frequency:	Weekly
PreConditions:	End of Week and at least 20 supply order items are available.
PostConditions:	Supply order has been printed for each supplier.
MainPath:	1) Supply clerk starts job. 2) System selects cheapest supplier for each article. 3) System orders supply order items by supplier number. 4) System prints a supply order for each affected supplier.
AlternatePath:	None

Exceptions:	System finds no supply order items. System finds no price for an article. System finds no supplier for an article.
Rules:	The supplier of an article is selected who offers the lowest price.
Objects:	Supply order items, prices, supplier, supply order.
Inherits:	Standard-printing-function.
Uses:	COBOL-Wrapper
Extends:	Customer-order-processing.
Comments:	The supply orders are sent out by email.

UseCase-05: Billing

At the end of each month the account should start the billing job. It is a batch process, which processes all the pending billing items. First, the billing items are sorted by customer number and order number. Then for each customer:

- the customer address data is taken from the customer data base,
- the cost of each billing item is computed as order amount * article price,
- the costs of the billing items are totaled to give the total cost,
- the total cost is multiplied by the VAT to give the amount due,
- the data due computed.

After that an invoice is printed with an invoice header containing the order data and customer data, an invoice body with all the billing items ,and an invoice trailer with the amount due and date due. The due date is exactly 30 days from the date on which the invoice is printed.

Attributes	**Description**
Label:	Billing
Fulfill:	Func-Req-04.
Process:	Bo-01.
Input:	Gui-03, Bo-10.
Output:	Repo-03, Bo-11.
Implement:	Br-09, Br-10.
Trigger:	EndofMonth.
Actor:	Accountant
Frequency:	Monthly
PreConditions:	At least one billing item must be available.
PostConditions:	An invoice is printed for each customer who ordered in the past month.
MainPath:	1) Billing job is started. 2) System sorts the billing items by customer. 3) System gets address data for each customer. 4) System computes the item cost for each billing item.

	5) System sums up the total costs of each billing item. 6) System computes the VAT of the total cost. 7) System calculates the data due. 8) System prints an invoice for each customer.
AlternatePath:	None
Exceptions:	System finds no billing items. System finds no customer data for an order.
Rules:	Due date = current date + 30 days. VAT = total cost + (total cost * 0.20).
Objects:	Billing items, Customer, Invoice.
Inherits from:	Standard_Print_Function.
Uses:	COBOL-Wrapper
Extends:	Customer_order_processing.
Comments:	The invoices are sent out by regular mail.

UseCase-06: Back_Order_Processing

At the end of each week the dispatcher will start the batch process to process the back orders. All pending back orders are processed in the same way the order items of the customer order are processing. First, the article is checked if it exists. Then the article amount on stock is checked if it is now greater than the order amount. If so, the order amount is deducted from the article amount, a dispatch item created and a billing item made. If the article amount sinks below the minimum amount on stock a supply order item is generated. This is repeated for each pending back order. If the article no longer exits an error message is printed. If the order amount is still greater than the article amount the back order remains in the back order file.

Attribute	**Description**
Label:	Back_Order_Processing
Fulfill:	Func-Req-06.
Process:	Bo-01, Bo-02, Bo-05, Bo-06.
Input:	Gui-03, Bo-12.
Output:	Bo-06, Bo-07, Bo-08, Bo-12.
Implement:	Br-02, Br-03, Br-04, Br-05, Br-06, Br-07, Br-08.
Trigger:	EndofWeek
Actor:	Dispatcher.
Frequency:	Weekly
PreConditions:	At least one back order must be available.
PostConditions:	Every back order has been processed. Dispatch order items are created. Billing items are created. Possibly one or more supply order items are created.

MainPath:	1) Dispatcher starts back order processing. 2) System processes each back order. 3) System checks if article is on stock. 4) System checks if article amount is sufficient. 5) System creates dispatch order item. 6) System creates billing item. 7) System creates supply order item if the article amount on stock falls below the minimum amount.
AlternatePath:	8) System skips over back order.
Exceptions:	System does not find the ordered article. System does not find customer.
Rules:	If article amount < minimum amount on stock, the system will create a supply order item.
Objects:	Back order, Customer, Article, Dispatch Order Item, Billing Item, Supply Order Item.
Inherits:	Standard-Ordering-Function.
Uses:	COBOL-Wrapper
Extends:	Customer-Order-Processing
Comments:	This UC should reuse as much as possible of the customer order processing use case.

5. Technical System Features

5.1 System Quantities

NF-REQ-11 Data-Quantities

There can be as many as 1,000,000 customers, 10,000 articles und 200 suppliers. The system must be able to handle up to 100,000 customer orders per day.

NF-REQ-12 Supply-Quantities

Every month a data carrier comes in with an XML File containing the latest supplier data and supplier prices. It is predicted that there will be circa 500 supply orders per month. The biggest supplier supplies up to 50 different articles.

NF-REQ-13 Article-Quantities

Also every month a data carrier comes with an XML File containing the changes to the article data, including the latest article prices. Up until now there were 1,000 updates – changes, deletions, insertions – to the 10,000 articles per month.

FUNC-REQ-07 Monthly-Reporting

At the end of every month our system is required to provide a report giving the number of articles sold and the number remaining for every article type on stock. This report goes to the warehouse manager as well as to the sales manager.

4.2 Technical Constraints

The following constraints apply to the order entry system:

- The System should be implemented as a web application in a PC-Network under a Linux operating system.
- The existing master data in VSAM files should be converted to relational data base tables in an Oracle database.
- The new online components are to be written in Java. The old batch components should remain in COBOL and be wrapped.
- The System should be designed and documented with UML-1.
- The customer orders must be processed within 3 seconds.
- The graphical user interface should require no more than 1 user hour to learn how to use.
- System responses should be in English, German, or French depending on the customer language preference.
- All order entry transactions should be logged and archived.
- Reports and Invoices should be printed on DIN-A4 pages in vertical format.

4.3 Quality Requirements

The most critical quality requirements to be fulfilled are:

- Reliability = there should be no more than one system crash per month.
- Usability = it should take no longer than 1 hour to be able to use the system.

Important quality requirements are:

- Correctness = there should be no more than two errors per 1000 statements.
- Integrity = at least 80% of all bad input should be detected and rejected.
- Performance = the response time constraint of three seconds per transaction should not be exceeded more than 10% of the tranasactions.

Middle important qualities are:

- Conformity = the design and coding rules should be adhered to by at least 75%.
- Testability = a metric rating greater than “0.60”.
- Maintainability = a metric rating greater than “0.50”.
- Security = 90% of all attempts to break into the system should be thwarted.

Of minor importance are the qualities:

- Efficiency,
- Portabilitity and

- Reusability.

A test coverage of at least 80% branch coverage is required for accepting the system.

&STOPME

4.4 Resources

The following hardware and software resources are available to the project:

- Three PC-Workstations
- MS-Windows-2007 Operating System.
- Web-2 for testing the internet connections.
- MS-Internet Explorer for the internet connection.
- MS-Word for the text documents.
- Oracle database system for the data.
- Eclipse for die JAVA component development
- MF-COBOL for the COBOL component testing
- Softwrap for wrapping the COBOL programs.
- DataTran für die Migration der Dateien
- Paradigm-Plus for the UML design.
- SoftCalc for the project cost calculation.
- COBAudit for checking the COBOL programs.
- SQLAudit for checking the Database schemas.
- JAVAudit for checking the Java sources.
- JUnit for testing the Java classes.
- TestSpec for specifying the test cases.
- DataTest for generating and validating the test data.

4.5 Project Constraints:

This project has to be finished within 3 months after starting. No more than 12 person months should be required. Costs may not exceed € 80.000.

4.6 Project Risks:

The project leader is required to perform a risk analysis to identify and evaluate the most probable risks. He should submit a risk evaluation report in which risks are ranked by their probability of occurring and their potential impact on project costs and time. For each risk at least one risk reduction measure should be identified and

quantified. In the end, the risks should be used to adjust the time and cost estimations.

Appendix: Current Master Data

CustomerData

- CustomerNumber* Dec (8)
- CustomerName* Char (40)
- CustomerCredibility Char 1 = good, 2 = acceptable, 3 = poor
- CustomerAddress
  - Street Char (40)
  - ZipCode Dec (5)
  - City Char (20)
  - Country Char (2)

ArticleData

- ArticleNumber* Dec (8)
- ArticleName Char (20)
- ArticleAmount Dec (6)
- MinimumAmount Dec (6)
- ArticlePrice Dec (6,2)
- SupplyAmount Dec (4)

SupplierData

- SupplierNumber* Dec (8)
- SupplierName* Char (40)
- SupplierCredibility Char 1 = good, 2 = acceptable, 3 = poor
- SupplierAddress
  - Street Char (40)
  - ZipCode Dec (5)
  - City Char (20)
  - Country Char (2)

 # PriceData

- SupplierNumber* Dec (8)
- ArticleNumber* Dec (8)
- SupplierPrice Dec(6,2)

* = key