Clifford Systems Requirements Specification

Version 1.0

Table of Contents

1 Executive Summary

1.1 Project Overview

Our product will feature an application that will be used by manufacture companies to manages the orders given by different distributors that want to work with that factory products. Clients will be able to make different orders and receive information regarding when the order can be shipped.

1.2 Purpose and Scope of this Specification

In scope

The following points are intended to show the core mechanics of this application:

- This project is intended to ease management of supply and production in certain company.
- Help clients make simpler and more concise orders without needing to physically discuss with the executives of the user company.
- Manage manufactured products by keeping in mind the maximum capacity of the magazine and the order of the distributors.
- Keep track of sales and make sure there are no problems.
- Manage employees in the same screen without having to disturb their environment.

Out of Scope

- Cross – platform usage so clients can order anytime and anywhere.
- Predict future sales so we can manufacture certain products at a higher rate when needed.
- Automate certain mechanics that will ease the burden on employees when dealing with manufacture production.

2 Product/Service Description

2.1 Product Context

Our application can be used by any product manufacturer. Although we are tailoring it into the food industry for now, small changes can be done to benefit any manufacturer in any industry. Application will have 2 main user interfaces: 1 for employees to manage all orders and processes happening from inside the company and another for customers who can schedule orders from anywhere.

2.2 User Characteristics

There are 3 main user classes:

Clients:

- These will be different companies that want to buy company products.
- Registration will be done by employees physically.
- Without logging in, clients can see the product lineup in a front page.
- Upon logging up, these clients can order what products they like.
- These orders will be handled by employees which will also display an approximate time when the order can be put through.
- Clients can see their current orders, finished orders and several info related to them.

Employee:

- These will be using a different interface that only deals on the company side.
- They can register new users who want to trade with the company.
- Employee accounts will only be registered by admin.
- Every employee must log in to start their shift.
- Upon logging in, they can see the current progress of the company, i.e. current magazine stock for different products, daily production by the manufacturer and pending orders.
- Pending orders can be viewed and once the employees are sure, they can approve their requests and manually schedule a date for pick up or leave it to the system to find out.

Admin:

- Admin will be a higher rank that can manage all the employees.
- He can register new employees and also delete both customers and employee accounts.
- Oversee the daily performance of the employees.

2.3 Assumptions

The application will be mostly used by the web. Clients can access the website to log in and make orders from anywhere with both phone and computers. On the other hand employees will feature a web application for starting but will eventually be converted into a windows application for faster responses.

2.4 Constraints

Possible constraints:

- Migrating data from the old application to the new one can be time consuming.
- Clients need to be connected to the internet to issue orders.
- In case of technical problems, they have to email the support team to fix them.

Dependencies

List dependencies that affect the requirements. Examples:

- No dependency is required.

3 Requirements

3.1 Functional Requirements

Clifford Systems Requirements:

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
CS_01	Customers should be able to see different products without having to log in.		1	18/04/22	
CS_02	Logged in users can order products in different amounts and work with special requests.		1	18/04/22	
CS_03	Employees should be able to review orders and either accept or reject them.		1	18/04/22	
CS_03	Employees should be able to observe daily production and stock of the magazine.		1	18/04/22	
CS_05	Employees should be able to register new clients that want to work with the company.		1	18/04/22	
CS_06	Admin should be able to register and delete employees, as well as clients.		1	18/04/22	

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
CS_07	Admin should be able to look over employees. Check their performance daily and observe their rejected and approved orders.		2	18/04/22	
CS_08	Customers should be able to contact technical support whenever they encounter a bug or a problem in the system.		2	18/04/22	
CS_09	Employees and Admin should be able to look at predicted productions in upcoming months.		3	18/04/22	

3.2 Non-Functional Requirements

In here try to use the Structure given at slide 13 in Requirements Engineering Lecture Slides, with main categories of:

3.2.1 Product Requirements

3.2.1.1 User Interface Requirements

In addition to functions required, describe the characteristics of each interface between the product and its users (e.g., required screen formats/organization, report layouts, menu structures, error and other messages, or function keys).

Client Interface

- 1. At the top of our page there will be a nav bar that will contain our logo on the left , a search bar and on the right side two icons. One of the icone is for the client account and the other one is for the shopping cart.
- 2. Below the nav bar the page will be divided into two parts where the left side will be significantly smaller as it will host the list of the function that can send the user to a different page for other information such as : Shop, Products, Categories, History and Contact Us.
- 3. The right side will be more complexe, it will contain a big box with sliding photos(these are going to our promotion products as an example), below them there are going to be 3 rows of 4 columns with pictures of the product and their names.
- 4.At the end of the products there will be a list to change the page of the products that we are looking at.
- 5. When we click on a product the page will change. Going into a page that has more photos of that product with a big one above them. On the left of the photos there will be a description of the product and its packing specifiks.
- 6. Below both of them there will be the price , quantity of the product that we want. On the right of them there will be a button to add the order into the shopping cart.

Employee Interface

- 1.The employee interface will consist of 3 parts. The top part will be the nav bar that will consist of the logo in the left side and of the user box in the right side.

- 2.The part below the nav bar will be divided into two parts. The left part will consist of 20% of the screen space and will show the list of functions that the employee can use. The will be put in the column view.
- 3.On the right part of the screen that will occupy 80% of the screen real estate it will contain boxes that each have their specific functions. Depending on the size of the screen that will be used the number of boxes displayed in a row will vary. In difference form the functions that will be shown in the list is that the ones in the boxes are the one the employee will use more frequently and can be easier to spot like : Daily Production, Clients, Register Clients, Supply Overview, Production Magazine Overview etc.

Admin Interface

- 1.From the point of construction the admin interface will be the same as the employee interface as they will have the same design.
- 2. What will change is the number of boxes as the admin has power to make changes that are out of the employee scope. As : deleting clients, registering new employee, overviewing employees, seeing the profit margin and seeing all the distributors etc.

3.2.1.2 Usability

- There will be a manual shown when the clients open the webpage.
- On every functional button, there will be a small description when you hover over them.
- For employees, a written guide will be provided to all of them.
- UI will be intuitive for both sides, being very minimal so users will not get confused at any of the functions.

3.2.1.3 Efficiency

3.2.1.3.1 Performance Requirement

- Client Web Page should be able to handle up to 1000 users simultaneously.
- Average number of clients per week should be around 100 clients.
- Since there will only be 1 employee that will handle the orders at one time, only 1 employee account will be connected at a given time.
- For automated responses, the application should handle up to 50 orders per second.
- Latency for an order to be shown on the employee side should be under 1 second.

3.2.1.3.2 Space Requirements

- Each order will feature a simple json-like structure that can go up to 5kB of data.
- These orders will be accompanied by their respective schedule and other written instructions from the employees for a total of 10kB of data.
- On average, we should need a total of around 5MB of data on a weekly basis.
- This sums up to 260 MB data needed for all orders in a year.
- Factory databases are much simpler , needing only the data for different products which will only get updated on place. On the other hand , daily production should be kept on a separate database that will add another 10kB per day.
- Total yearly production will need 3.6MB of data.

3.2.1.4 Dependability

Availability

- The Web Page should be running 24/7 so clients can see the products and even make orders at any time.

- Employees will have 2 shifts. Each shift will be 8 hours, 8:00 to 16:00 and 16:00 to 00:00. At these times, any order can be reviewed by the corresponding employee.
- Admin page will be available at any time the administration wants to check anything related to clients and employees.

Reliability

- The application should be reliable for any time of the day between 5:00 up to 4:00 the next day.
- Sometimes there can be maintenance that will be scheduled between 4:00 to 5:00 since there won't be much traffic during that hour in the morning.
- For other days, the system will be active at any hour.

Monitoring

- Daily production from the factory should be updated every 2 hours so the employees can take the latest data before proceeding with the orders.
- Every transaction should be logged and the admin can keep track of these logs on his screen.

Maintenance

- Every time we need to update the system. We should temporarily save the pending orders before proceeding.
- Then we can proceed with 1 hour downtime on the servers between 4 AM to 5 AM.
- Once the maintenance is complete, the orders are added back to the server.

Integrity

3.2.1.5 Security

- Every user, be it client or employee, will be hashed when saved on the database.
- APIs will be protected using OAuth and using only https connections.
- Every function of the system will be done with RESTful APIs which will be stateless, so authentication will not depend on sessions.
- Databases will only be accessed by the backend admin. Employees can only see data but not change anything except orders.
- Employee order responses will be logged to the admin in case the employee account has been breached from the inside.
- Orders and accounts cannot be deleted or changed unless the admin wants to. In that case, the admin activity will be logged into a separate file so other higher ups can loop for suspicious activity.

3.2.2 Organizational Requirements

Creation of customer accounts will be handled by employees, rather than creating it themselves.
Employees should always log out before finishing their shift in order for the next employee to start his shift.

3.2.2.1 Environmental Requirements

The product will be mainly accessed by PC/laptops powered by the Windows operating system.
The website for the clients however can be accessed by any type of device at any time and space.

3.2.2.2 Operational Requirements

Our product is a mix of web pages which will be used by the clients in order for the client to make different orders depending on their request but also an application for the employee and the admin for them to accept the orders and also to check the production of the factory.

3.2.2.3 Development Requirements

The technologies used in this project will be:

- Front-end: HTML , CSS , JavaScript (possibly css frameworks for better styling)
- Back-end: Python with Django framework

3.2.3 External Requirements

3.2.3.1 Regulatory Requirements

The client should accept the fact that the primary data will be provided by the employee as well as the fact that the data regarding the orders that client makes will be checked by the employee and by the admin.

3.2.3.2 Ethical Requirements

- The client must accept for their credentials to be created by the employee and also all their data regarding the orders to be shared by the employee and the admin.
- The employee should accept that all the data that he will be providing to the company will be checked by the admin in order for the work rate to be as efficient as possible.

3.2.3.3 Legislative Requirements

Our applications will comply with all the necessary Albanian laws, regarding the protection of the data of the client (in terms of the usage of the web page). All the sensitive data regarding the information of the client and also their orders will remain confidential and between the client and the company, without sharing with third parties.

3.2.3.3.1 Accounting Requirements

The website will be completely free and only accessible by the clients of the company, not requiring any kind of monetary transaction.

3.2.3.3.2 Security Requirements

The system will provide a security of data encryption which will mainly focus on the passwords being hashed in order for the employee or the admin not to have access to the clients account.

3.3 Domain Requirements

The domain requirements for our project will be a good responsive web page for the client, in order for them to have easy access to the products and also make orders accordingly to their needs. Also the application for the employee and the admin will be a simple yet compact solution for all their needs according to the requirements of each position.

4 User Scenarios/Use Cases

Provide a summary of the major functions that the product will perform. Organize the functions to be understandable to the customer or a first time reader. Include use cases and business scenarios, or provide a link to a separate document (or documents). A business scenario:

- Describes a significant business need
- Identifies, documents, and ranks the problem that is driving the scenario
- Describes the business and technical environment that will resolve the problem
- States the desired objectives
- Shows the “Actors” and where they fit in the business model
- Is specific, and measurable, and uses clear metrics for success

APPENDIX

The appendixes are not always considered part of the actual Requirements Specification and are not always necessary. They may include

- Sample input/output formats, descriptions of cost analysis studies, or results of user surveys;
- Supporting or background information that can help the readers of the Requirements Specification;
- A description of the problems to be solved by the system;
- Special packaging instructions for the code and the media to meet security, export, initial loading, or other requirements.

When appendixes are included, the Requirements Specification should explicitly state whether or not the appendixes are to be considered part of the requirements.

A Definitions, Acronyms, and Abbreviations

Define all terms, acronyms, and abbreviations used in this document.

B References

List all the documents and other materials referenced in this document.

C Requirements Traceability Matrix

The following trace matrix examples show one possible use of naming standards for deliverables (FunctionalArea-DocType-NN). The number has no other meaning than to keep the documents unique. For example, the Bargaining Unit Assignment Process Flow would be BUA-PF-01.

For example (1):

Business Requirement	Area	Deliverables	Status
BR_LR_01 The system should validate the relationship between Bargaining Unit/Location and Job Class.---Comments: Business Process = "Assigning a Bargaining Unit to an Appointment" (Priority 1)	BUA	BUA-CD-01 Assign BU Conceptual Design	Accepted
		BUA-PF-01 Derive Bargaining Unit-Process Flow Diagram	Accepted
		BUA-PF-01 Derive Bargaining Unit-Process Flow Diagram	Accepted
BR_LR_09 The system should provide the capability for the Labor Relations Office to maintain the job class/union relationship.---Comments: Business Process = "Maintenance" (Priority 1)	BUA	BUA-CD-01 Assign BU Conceptual Design	Accepted
		BUA-PF-02 BU Assignment Rules Maint Process Flow Diagram	ReadyForReview

For example (2):

BizReqlD	Pri	Major Area	DevTstItems DelivID	Deliv Name	Status
BR_LR_01	1	BUA	BUA-CD-01	Assign BU Conceptual Design	Accepted
BR_LR_01	1	BUA	BUA-DS-02	Bargaining Unit Assignment DB Modification Description	Accepted
BR_LR_01	1	BUA	BUA-PF-01	Derive Bargaining Unit-Process Flow Diagram	Accepted
BR_LR_01	1	BUA	BUA-UCD-01	BU Assign LR UseCase Diagram	ReadyForReview

BizReqID	Pri	Major Area	DevTstItems DelivID	Deliv Name	Status
BR_LR_01	1	BUA	BUA-UCT-001	BU Assignment by PC UseCase - Add Appointment and Derive UBU	Reviewed
BR_LR_01	1	BUA	BUA-UCT-002	BU Assignment by PC UseCase - Add Appointment (UBU Not Found)	Reviewed
BR_LR_01	1	BUA	BUA-UCT-006	BU Assignment by PC UseCase - Modify Appointment (Removed UBU)	Reviewed
BR_LR_09	1	BUA	BUA-CD-01	Assign BU Conceptual Design	Accepted
BR_LR_09	1	BUA	BUA-DS-02	Bargaining Unit Assignment DB Modification Description	Accepted
BR_LR_09	1	BUA	BUA-PF-02	BU Assignment Rules Maint Process Flow Diagram	Accepted
BR_LR_09	1	BUA	BUA-UCD-03	BU Assign Rules Maint UseCase Diagram	Reviewed
BR_LR_09	1	BUA	BUA-UCT-045	BU Assignment Rules Maint: Successfully Add New Assignment Rule	Reviewed
BR_LR_09	1	BUA	BUA-UCT-051	BU Assignment Rules MaintUseCase: Modify Rule	Reviewed
BR_LR_09	1	BUA	BUA-UCT-053	BU Assignment Rules MaintUseCase - Review Assignment Rules	Reviewed
BR_LR_09	1	BUA	BUA-UCT-057	BU Assignment Rules MaintUseCase: Inactivate Last Rule for a BU	Reviewed
BR_LR_09	1	BUA	BUA-UI-02	BU AssignRules Maint UI Mockups	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-021	BU Assignment Rules Maint TestCase: Add New Rule (Associated Job Class Does Not Exist) - Success	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-027	BU Assignment Rules Maint TestCase: Modify Rule - Success	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-035	BU Assignment Rules Maint TestCase: Add New Rule (Associated Job Class Does Not Exist) - Error Condition	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-049	BU Assignment Rules Maint TestCase: Modify Rule - Error Condition	ReadyForReview

For example (3):

BizReqID	CD01	CD02	CD03	CD04	UI01	UI02	UCT01	UCT02	UCT03	TC01	TC02	TC03	TC04
BR_LR_01			X		X		X			X		X	
BR_LR_09	X			X		X			X		X		X
BR_LR_10	X			X					X		X		

BizReqID	CD01	CD02	CD03	CD04	UI01	UI02	UCT01	UCT02	UCT03	TC01	TC02	TC03	TC04
BR_LR_11		X											

D Organizing the Requirements

This section is for information only as an aid in preparing the requirements document.

Detailed requirements tend to be extensive. Give careful consideration to your organization scheme. Some examples of organization schemes are described below:

By System Mode
Some systems behave quite differently depending on the mode of operation. For example, a control system may have different sets of functions depending on its mode: training, normal, or emergency.

By User Class
Some systems provide different sets of functions to different classes of users. For example, an elevator control system presents different capabilities to passengers, maintenance workers, and fire fighters.

By Objects
Objects are real-world entities that have a counterpart within the system. For example, in a patient monitoring system, objects include patients, sensors, nurses, rooms, physicians, medicines, etc. Associated with each object is a set of attributes (of that object) and functions (performed by that object). These functions are also called services, methods, or processes. Note that sets of objects may share attributes and services. These are grouped together as classes.

By Feature
A feature is an externally desired service by the system that may require a sequence of inputs to affect the desired result. For example, in a telephone system, features include local call, call forwarding, and conference call. Each feature is generally described in a sequence of stimulus-response pairs, and may include validity checks on inputs, exact sequencing of operations, responses to abnormal situations, including error handling and recovery, effects of parameters, relationships of inputs to outputs, including input/output sequences and formulas for input to output.

By Stimulus
Some systems can be best organized by describing their functions in terms of stimuli. For example, the functions of an automatic aircraft landing system may be organized into sections for loss of power, wind shear, sudden change in roll, vertical velocity excessive, etc.

By Response
Some systems can be best organized by describing all the functions in support of the generation of a response. For example, the functions of a personnel system may be organized into sections corresponding to all functions associated with generating paychecks, all functions associated with generating a current list of employees, etc.

By Functional Hierarchy
When none of the above organizational schemes prove helpful, the overall functionality can be organized into a hierarchy of functions organized by common inputs, common outputs, or common internal data access. Data flow diagrams and data dictionaries can be used to show the relationships between and among the functions and data.

Additional Comments
Whenever a new Requirements Specification is contemplated, more than one of the organizational techniques given above may be appropriate. In such cases, organize the specific requirements for multiple hierarchies tailored to the specific needs of the system under specification.

There are many notations, methods, and automated support tools available to aid in the documentation of requirements. For the most part, their usefulness is a function of organization. For example, when

organizing by mode, finite state machines or state charts may prove helpful; when organizing by object, object-oriented analysis may prove helpful; when organizing by feature, stimulus-response sequences may prove helpful; and when organizing by functional hierarchy, data flow diagrams and data dictionaries may prove helpful.