system requirements specification

<Monqez>, <2.0>

<Ehab Fawzy, Hatem Mamdoh, Hussien Ashraf, Khaled Ezzat>

<October, 19 2020>

Contents

[Section I: INTRODUCTION 3](#_Toc57053410)

[1 About thIS Document 4](#_Toc57053411)

[1.1 Document Purpose and Scope 4](#_Toc57053412)

[2 About the Project 4](#_Toc57053413)

[2.1 Stakeholders 4](#_Toc57053414)

[2.2 Business Background 4](#_Toc57053415)

[2.3 Project Purpose and Scope 5](#_Toc57053416)

[2.4 Operational Environment 5](#_Toc57053417)

[2.5 Facts, Assumptions, Dependencies, Constraints, Risks 5](#_Toc57053418)

[3 GLOSSARIES of Terms 5](#_Toc57053419)

[Section II: About the SYstem 6](#_Toc57053420)

[1 System Overview 6](#_Toc57053421)

[1.1 MAIN FEatures 6](#_Toc57053422)

[1.2 System Structure 6](#_Toc57053423)

[1.3 Context diagram 7](#_Toc57053424)

[1.4 System functions 7](#_Toc57053425)

[2 User Groups and Access Privileges 8](#_Toc57053426)

[Section III: System REQUIREMENTS 10](#_Toc57053427)

[1 Functional Requirements 11](#_Toc57053428)

[1.1 Overall Processes 11](#_Toc57053429)

[1.2 [Module, function, or page] 11](#_Toc57053430)

[2 Reports 16](#_Toc57053431)

[2.1 [REP-01: Sales Commission total per month] 16](#_Toc57053432)

[3 Common Components 18](#_Toc57053433)

[4 Data Dictionary 19](#_Toc57053434)

[5 Integration/Interfaces 20](#_Toc57053435)

[6 Logging Requirements 21](#_Toc57053436)

[7 Quality Requirements 21](#_Toc57053437)

[7.1 Capacity and Performance 21](#_Toc57053438)

[7.2 Security 22](#_Toc57053439)

[8 Other Requirements 23](#_Toc57053440)

[8.1 Technology Requirements 23](#_Toc57053441)

[8.2 Data Migration 23](#_Toc57053442)

[8.3 System Conventions 23](#_Toc57053443)

[8.4 Usability 23](#_Toc57053444)

[8.5 Compliance Requirements 23](#_Toc57053445)

[8.6 Availability 23](#_Toc57053446)

[8.7 training and Documentation Requirements 24](#_Toc57053447)

[Section IV: Appendices 25](#_Toc57053448)

[Appendix I: Dropped/Changed Requirements 26](#_Toc57053449)

[Appendix II: Client’s Wish List for Future Releases 27](#_Toc57053450)

[Appendix III: Sample Documents 28](#_Toc57053451)

[Appendix IV: Open/Closed Issues 29](#_Toc57053452)

# Section I: INTRODUCTION

## 1 About thIS Document

This document will define the design of the Monqez application. It contains specific information about expected input, output, classes and functions. The interaction between the classes to meet the desired requirements are outlined in detailed figures at the end of the document.

### 1.1 Document Purpose and Scope

This document details all the functional and quality requirements of Monqez Application. The intended reader groups for this software requirement specification are the project manager, developers, supervising professor, and any person interested in developing any application like Monqez.

## 2 About the Project

Monqez is a cross platform mobile application that aims to help people who need first aid by informing the closest first aider available from the patient's location to help them and save their lives. All registered first aider have passed a first aid course from accredited locations.

### 2.1 Stakeholders

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **ID** | **Phone Number** | **Email** |
| Ehab Fawzy | 20170072 | 01204860055 | ehab1571999@yahoo.com |
| Hatem Mamdoh | 20170085 | 01016192209 | matrix.leader@yahoo.com |
| Hussien Ashraf | 20170093 | 01016395068 | hussienashraf99@gmail.com |
| Khaled Ezzat | 20170098 | 01276016539 | khaledelhadary@outlook.com |

### 2.2 Business Background

This section may contain:

* Business concepts and general information
* Existing business processes (as-is situation)
* Client information that do not represent requirements, such as the client’s organizational chart.

Notes:

* If reading other documents would help in understanding the business, mention their names and location.
* If this section becomes too big, move it to a separate document and mention it in the Document Context section.

### 2.3 Project Purpose and Scope

**Purpose**: This software's purpose is to develop the full Monqez application, it should be able to match and connect between the user calling for help and the nearest available Monqez as well as provide some basic first aid help either via pictures or calls. The software aims to provide a quick help for the user in need until the ambulance arrives or gets well.

**Scope**: The application helps the community by saving their lives by providing first aid to people in need by qualified people as soon as possible. Users of this application should be divided into 3 main types: Normal, Helper (first aider), and Administrator. The normal user can ask for a helper either online via video call or onsite where the helper arrives to the location provided and rate them. The normal user can see the basic instructions in an easy way that is tailored to his situation. The Helper can select the time he wants to receive requests, accept or decline requests, view more information about the accepted request as well as rate the normal user after the request is fulfilled. The admin can view all the applications as well as accept or decline it and review all the complaints and ratings.

### 2.4 Operational Environment

The client can use this software on different environments such as (iOS 10.0+ and Android 7.0+). 4G connection needs to be established from the user’s mobile for stable video and voice call.

The server-side components of the software system must operate within a Windows operating system environment running Node JS.

### 2.5 Facts, Assumptions, Dependencies, Constraints, Risks

This section should include only facts, assumptions, dependencies, constraints, and risks that pertain to the project or system as a whole - not to specific requirements. Put the requirements-specific assumptions inside the requirements body close to their relevant requirement. Mention here that other assumptions are written within the below sections. Because many people don’t read this section, make sure you mention the important points within the body of the requirements or at least refer to this section.

Facts can be information about the client’s situation. Ex: users use 256k Internet connections.

Assumptions are what you had to assume because you couldn’t get a confirming answer from the client. Assumptions must be clearly stated because they put the project at a higher risk.

Dependencies can be on other systems or projects that your project depends on. Example: Implementing an integration feature depends on the availability of the system your system will integrate with.

Constraints can be:

* Time Constraints: such as delivery time
* Technical Constraints: design, tools, or technology constraints
* Budget or Resources Constraints

Risks that affect requirements can be:

* Expected changes on the client side that may affect your project
* Too many stakeholders
* No clear ownership
* Primary stakeholder not reachable

## 3 GLOSSARIES of Terms

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Monqez | It is the first aider (helper) that arrives to help the person calling for help. |
| IDE | Integrated Development Environment is a software application that provides comprehensive facilities to computer programmers for software development. |
| API | Application Programming Interface is a computing interface which defines interactions between multiple software intermediaries. |
| MVC | MVC is architectural pattern which stands for Model, View, and Controller. MVC separates an application into three components - Model, View, and Controller. |

# Section II: About the SYstem

## 1 System Overview

### 1.1 MAIN FEatures

Possibly list the main capabilities of the system. Example:

* [PP-01] System shall allow users to maintain personal profiles with personalized preferences
* [RPT-01] System shall provide its users with a reporting tool
* Connect normal to Monqez.
* Make video and voice call.
* Anyone can see the first aid instructions.
* Order for others.
* Complaints.
* Rating.
* Etc…

### 1.2 System Structure

You can partition a system by sub systems, modules, functions, or user groups. Ex:

The system is composed of the following sub systems, modules, functions:

* Customer Management: brief description
* Invoicing: brief description
* Reporting: brief description

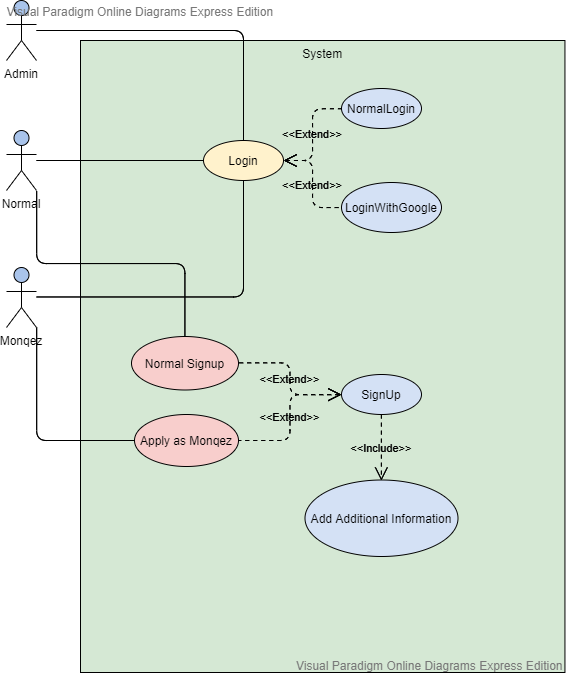
If you can, draw a decomposition diagram like the following:



### 1.3 Context diagram

If the system interfaces with other systems, departments or business entities; put the context diagram here.

### 1.4 System functions



## 2 User Groups and Access Privileges

**Normal user:** It is a user who was at the scene of the accident reporting and asking a paramedic to do first aid for him or another person. The user has the following characteristics:

* Signup:
  + The normal user could register his / her details to be able to call paramedics.
* Login:
  + The normal user could login using his phone number or his email to be able to use the application functionalities.
* Ask for Monqez:
  + The normal user has the capability to ask for Monqez in case himself or others were involved in an accident or need quick medical care.
  + The normal user can ask for either onsite or online(video/voice) Monqez.
* Display first aid instructions:
  + The normal user can display first aid instructions about many different types of injuries.
* Rate the Monqez.

**Monqez user:** He is a user who registers himself in our system as a rescuer (Monqez), and that is by attaching an accredited certificate of the first aid training course he obtained from any accredited organization and then awaits approval of his application to join the rescuers list until the certificate obtained is reviewed, and the rescuing user (Monqez) has the following characteristics:

* Signup:
  + The Monqez user can apply his information attached with his first aid certificate to the rescuers queue to be reviewed.
* Login:
  + The Monqez user could login using his phone number or his email to be able to use the application functionalities.
* Select working time:
  + The Monqez user has the option to select the status when he will be available for work.
* Accept requests:
  + The Monqez user can accept either onsite or online(video/voice) requests.
* Request Description:
  + The Monqez user can display the description of the requests provided by the normal user who made the request.
* Make complaint:
  + The Monqez user can complain about any inappropriate events that happened while fulfilling a request.

**Admin user:** The Admin user accepts or rejects requests to join the rescuers list based on the validity of the certificates that the rescuers attach when registering the account. The specialized user has the following characteristics:

* Login:
  + The Admin user could login using his phone number or his email to be able to use the application functionalities.
* View Monqez applications requests:
  + Retrieve all the non-reviewed Monqez applications requests and displays them.
* Accept/decline Monqez application requests:
  + The admin user has the capability of accept or decline the Monqez application requests after reviewing the first aid certificate.
* View ratings and complaints:
  + The admin user can review complaints and ratings.
* Block users:
  + The admin user has the capability to block any user if there are many valid complaints against him.

# Section III: System REQUIREMENTS

The concept behind this section structure is to group everything related to a particular system component in one place, so that developers see everything in one place and not have to move back and forth to find a piece of information in different sections of the document. You may however use only some of the pieces based on how applicable they are, or change the order of the sections, if it makes more sense in your situation.

## 1 Functional Requirements

### 1.1 Overall Processes

If the system as a whole automates a process end to end (such as a workflow process), draw diagram(s) that shows the major cycle here. Keep this section for the high level processes that belong to the system as a whole. Any sub processes or other processes that pertain to a specific component of the system should be put inside the component’s section below.

### 1.2 [Module, function, or page]

Categorize the system by module, function, or page depending on its size and nature.

Replace the above title with the actual name. Each section should cover a certain logical (or real) system component, such as a main function, sub process, page, feature, window, or tab. This section should be repeated for every module, function, or page in the system.

Possibly put here a brief description of the concept behind the function or the page covered in this section.

If you have atomic requirements related to the topic of this section, list them here and give them numbers.

If the module is too big to fit in one section, have a separate document for every module.

Put the first basic requirement: Prefix+number: The system shall provide the xx module

Ex: [PUR-01]: The system shall provide a Purchasing module.

Depending on the nature of what you are describing, you could use any of the following:

Process: If there is a process related to the topic of this section, include it here. Ex: If the system handles a workflow process, and each section covers a sub process; include the sub process here.

Use Cases/Scenarios: List here the numbers of the use case(s) relevant to the section’s topic from the Use Cases document.

Ex: UC-PUR-01: Create Purchase Order

System Function Specification: If there is an internal function processing involved, describe it here. By function, we mean a process carried out by the system without actor involvement (except may be to trigger the process and receive a result).

Go through the following:

* Input: data or action

Steps: detail one by one the function steps in sequence

Output: data or system result

Business entities and data to be stored: mention any data that must be captured by the system (data needed to perform the function and data that result from the function)

Example:

Function: Calculate employee net salary

Input: employee record, gross salary, tax rates, manager approval

Steps:

1. Get gross salary

2. Add up allowances

3. Subtract allowances from gross salary

4. Subtract benefits

5. Calculate tax

6. Subtract tax from result amount in # 3

7. Add transportation allowance

8. Subtract loan installment

Output: net salary, send notification email to employee

Stored information: tax rates, benefits, net salary, calculation date

Business Rules: If you have any business rules that belong to the function, include them here. Business rules can be put under their relevant requirement or separately. Do not put here field validations. Keep the field validations in the table above in the Page (or Window) Components section. Here put business rules that do not belong to a specific field (or group of fields). These are field validations that should be listed in the table below.

Each business rule must be given a number.

Example:

BR-PA-01: The monthly payroll cycle starts from 1st day of the month to the 5th.

BR-PA-02: Salary calculation must happen within the open monthly payroll cycle.

Open Issue: If there are still pending issues in a requirement, highlight it in this way.

Ex: *Open Issue: The tax rates are still to be defined.*

Screenshot/Wireframe: Insert the prototype here or refer to its location.

Page (or window) Components

If there is a page or window, list their controls here. Include data fields, buttons, or anything else.

Give all the table one ID; or, if you will refer to the same data element more than once, give each data element an ID (ex: DE-PUR-03).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element ID** | **Label** | **Type/Length** | **Data Validations** | **Business Rules** |
|  | The field label that will appear to the user | Text - list – free form – numeral – alphanumeric – Boolean | Mandatory: Y/N  Updateable: Y/N  Unique: Y/N  Auto-incrementing?  Input: base or derived  If currency: show sign? in prefix or suffix?  If date: Gregorian or Hijri, letters or numerals, order, length, and separators  Default Value? what is it and where it comes from? or put none  LOV: possible values  If the field is a result of calculation, what is the formula? How accurate the round should be? To which digit?  Dependency: or limitation based on another field or setting? | Logical Validations: accept zeros? Min and max?  For buttons, what happens when clicked (results/output)  Any other rule |

Exception Handling: List here the possible error cases related to the function you are describing and how the system will respond.

Ex: How will the system respond if the session timed out?

Messages: Put here the messages that the system will show to the user in certain conditions. Messages can be confirmation, warning, or error message. If someone else will write these messages, include them here when they are ready, so that the customer sees them and decides if they are understandable.

|  |  |  |
| --- | --- | --- |
| **Message Number** | **Message/Email Text** | **Condition** |
| MSG-YY-xx (ex: MSG-LOG-01) for message in the Login page | Show this text.. | When this happens |
|  | The email content is.. | If the system sends automatic emails |

Complete Example Showing how to put everything together:

**1) Transfer Funds**

[SR-TF-01] System shall provide an online Funds Transfer functionality.

[SR-TF-01.1] Transfer shall be allowed between accounts that belong to the same customer and within the same bank only.

**UC-TF-01: Transfer Funds**

Precondition: Actor has been authenticated

Basic Flow of Events

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. Actor requests to transfer funds between two of his/her own bank accounts. | System displays a list of the user’s accounts. |
| 2.Actor fills the required transfer information (for exact information, see data section below). | System validates entered data; and, if valid, confirms the request with message MSG-FT-01 (see messages below). |
| 3.Actor confirms the request. | System processes the request; and upon completion, sends a confirmation email to the actor. |

Post condition: Funds are debited from the origin (from) account and credited to the destination (to) account; and the transaction is logged with a time stamp.

**Data Elements**

[DT-TF-01] The following Information shall be available to the user:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Element Label** | | **Type/Length** | | **Data Validations** | | **Business Rules** |
| **Customer Information (read-only from Customer Profile)** | | | | | | |
| Customer Number |  | |  | |  | |
| Customer Name |  | |  | |  | |
| Customer Type |  | |  | |  | |
| **Transaction Information** | | | | | | |
| Origin Bank | Text box (200) | | Mandatory | |  | |
| Origin Bank Account | List | | Mandatory | |  | |
| Destination Bank | Text box (200) | | Mandatory | |  | |
| Destination Bank Account | List | | Mandatory | | Destination Bank Account must be different than the Origin Bank Account | |
| Transfer Date | Date | | - Mandatory  - Format: dd-mm-yyyy  - Default: today’s date | | Date cannot be before today’s date. If entered date is before today’s date, show message # MSG-TD-01 (see messages section) | |
| Amount | Numeric (10) | | Format: 00,000.00 | | Max = 10,000  Does not allow zeros | |

[SR-TF-02] To process the transfer, the system shall apply the following logic:

1.       Get the required amount and account numbers

2.       Check that the two accounts statuses = Active

3.       Check that the required amount is less than the origin account available balance and less than the allowed transfer funds limit

4.       Subtract the amount from the origin account available balance

5.       Add the amount to the destination account available balance

6.       Log the transaction in the audit log with a time stamp

7.       If any of the above validations is not passed, the system shall roll back the transaction and log it in the audit log with a time stamp

**Exception Handling**

* [SR-TF-03] If the user does not confirm the request, system shall warn user that the request will be discarded and display message MSG-FT-02.
* [SR-TF-04] If the status of one of the selected accounts is not Active, system shall not allow transfer and display message MSG-FT-03.
* [SR-TF-05] If the requested fund is bigger than the origin account balance or bigger than the transfer amounts limit, system shall not allow transfer and display message MSG-FT-04.

**Messages**

|  |  |  |
| --- | --- | --- |
| **Message #** | **Message/Email Text** | **Condition** |
| MSG-TF-01 | You are about to transfer the amount of <00,000.00> from account number <xxxx-xxxx-xxx> to account number <xxxx-xxxx-xxx>. Do you want to proceed? | User requests to transfer fund |
| MSG-TF-02 | Your transfer request is about to be canceled. Are you sure you want to proceed? | User cancels request or does not confirm request |
| MSG-TF-03 | One of the accounts is not active. Please contact Customer Service at <xxx-xxx-xxxx>. | The status of one of the accounts is not Active |
| MSG-TF-04 | The entered amount is not valid. You can transfer only up to <xxxxxx.xxx> | The required amount is bigger than the available balance in the origin account or than the allowed transfer limit |
| EMAIL-TF-01 | Dear <username>,  Thank you for using Online Banking. Your request to transfer funds from your account number <xxxx-xxxx-xxx> to account number <xxxx-xxxx-xxx> has been processed successfully. If you have any questions, please login to your account and use the Help Center option to contact a customer service representative.  It has been an honor to serve you! | Transaction ended successfully |

## 2 Reports

Mention common requirements that apply to all the reports in the system here. Ex: All reports must be exportable to Excel format.

[REP-01] All reports shall be …

[REP-02] All reports shall show the username in their headings.

### 2.1 [REP-01: Sales Commission total per month]

Use the name that the client uses (if available). Include the following where applicable:

Description – why do users need this report?

Distribution method: Will the report be printed? Will it be automatically generated and emailed to customer? Will a notification be emailed and the report placed on a specific path?

Specific output format: PDF, Excel, else?

Report usage frequency: daily, monthly, yearly?

Filtering: what search criteria will the user use to limit the report data? How will these criteria be ordered on the search criteria screen?

Filtering Validations: If an error occurs in filtering, which message will be displayed? Ex: Record does not exist.

Report Structure

|  |  |
| --- | --- |
| **Column** | **Specification** |
| **REPORT HEADER** | |
| Field | Value populated from data structure/element |
| **REPORT BODY** | |
| Field | Value populated from data structure/element  Sorting: Which of the report columns will be used to sort data?  Grouping rules: Mention if the report data will be grouped by this field.  Calculations: Include the formula of producing the report data (if any) |
| **REPORT FOOTER** | |
| Field | Value populated from data structure/element  Sum: Should the report provide subtotals? Grand total?  Average: Should the report provide averages for any values in the report? |

Screenshot

Put a rough report layout, or include the actual one when it is ready.

Example:

[RP-01] Users shall be allowed to print or to export all reports to PDF or XLS.

Report XYZ

[RP-XYZ-01] The XYZ report shall be generated by the system administrator to check the numbers of XYZ recorded in a certain period.

[RP-XYZ-02] Search Criteria:

|  |  |
| --- | --- |
| Criterion | Validations |
| XYZ Period | - Mandatory  - From/To Date: dd/mm/yyyy |

[RP-XYZ-03] The report shall contain the following information:

|  |  |
| --- | --- |
| Column | Specification |
| REPORT HEADER | |
| User Name | Populated from the User\_Name of the logged in user |
| REPORT BODY | |
| XYZ Number  Subscription Date | Populated from the XYZ data element in XYZ Profile  Format=dd-mm-yyyy |
| REPORT FOOTER | |
| Total | Total = grand total of all records in the report |

Exception Handling

|  |  |  |
| --- | --- | --- |
| Number | Condition | Response/Message |
| RP-XYZ-05 | If no XYZ records are found | Stop and show message: There are no XYZ in the selected period. |

## 3 Common Components

Because requirements must not be repeated, if you find shared components that can be used in different areas of the system, list them here and refer to them from the sections that use them, to avoid repeating the same info in more than one place. Do this only if you find that you have many components, not just one or two components. If the common components are just one or two, place it only once the first time it occurs, then refer to it in the subsequent ones.

Shared Components can be: Business Rules, Data Structures or Elements, anything else

## 4 Data Dictionary

If for any reason, there is a need to put all data in the system in one section, use this section. List the data structures (forms, invoices, etc). Then define the data elements within each structure as explained in the Page/Window Components section.

Example: In a workflow automation system, the same registration form can be filled by the clerk, the reviewer, and the approver; each filling his own section when it arrives to him in the workflow process. In this case, the structure is accessed from more than one component in the system, put it here and divide it to data sets by user view.

## 5 Integration/Interfaces

If the system integrates with other systems, detail this here. Mention the following, where applicable:

* What event will initiate the connection? Events can be time-based or activity-based.
* What data elements will be exchanged and in which format?
* Are there any APIs or Web Services?
* If the system is required to write in the other system database, include the data model.

Exception handling

* What happens if the connection times out?
* What happens if the data exchanged is invalid?

Note that your role is to only bring this info from the other systems owners and coordinate the information with your technical team.

## 6 Logging Requirements

If the system is required to log certain activities on the system, mention so here. Mention what exactly will be logged and any particular format requirements; where (database or log file); and when (real time, daily)?

## 7 Quality Requirements

Quality requirements are the non-functional requirements. Make sure these requirements are measurable so that testers can verify if they were satisfied.

### 7.1 Capacity and Performance

Capacity and performance are essential in any project. Put these requirements in a table format:

You can ask the PM on the customer side to provide this information from the customer’s IS department. Note that if the customer has an existing system, you might ask them to provide you with the IS log for a certain period of time (ex: one week).

|  |  |
| --- | --- |
| **Factor** | **Requirement** |
| How many users will use the system? | Ex: 1,000 |
| How many times a particular transaction will be performed per unit of time? | Ex: 10 searches per hour |
| How many records will be stored in the database of the main business entities (such as customers)? | Ex: 10,000 |
| Any potential growth in the coming few years: records and users? What is it? (Ex: if the customer plans on providing a promotional event, this may increase the number of users/records). | Ex: 10% yearly growth in the coming 5 years |
| How many users will concurrently use the system (find the worst case scenario)? And optionally, how much time they expect to reach the peak traffic and for how long it remains? | Ex: 1,000 reached after one hour and remains for 5 hours |
| What is the acceptable system response time? | Ex: 5 secs |
| What is the pages hit ratio (which pages are more likely to be used by the users)? | Ex: Login – 10 per minute |
| How much resources will the system use of the machine? | Ex: 25% of the server memory |
| Will users use a LAN or a WAN? |  |
| If WAN, Internet or leased line? |  |
| What is the bandwidth, latency, and packet loss? |  |

Exception Handling

If an exception occurs (such as if the number of users exceeds the maximum, a message may appear: System is temporarily unavailable, try again later).

### 7.2 Security

When applicable, document the following:

Authentication

* How will the system authenticate users? Passwords are the commonly used method. But sometimes users are authenticated using smart cards, photos, fingerprints, or else.
* How will passwords be structured? Are there any specific characteristics for the passwords? For example, must they be of particular length, contain numbers with letters, or mix capital with small letters (case sensitive)?
* How many failure attempts will be allowed? Will the system allow the user to retry login using wrong passwords endless times or will it stop him after certain number of attempts?
* If the system will stop users after a number of invalid trials, what happens next? How will the user recover the password? Will the system send him a new password for example?

Sensitive Data

* Is there and sensitive data that the system needs to protect (ex: royal personal data or credit card numbers)?
* Against what should the data be protected: only update or delete; or even viewing?
* How serious is the threat risk? How important it is for the client’s business to protect the data? What will the client lose?
* Communicate with the technical team to know the solution they are providing for these issues and document them here to ensure the customer is happy with the solutions. Solutions can be: HTTPS, SSO, encryption, certificates, etc.

Exception Handling

What happens when a breach is attempted or if the system fails in any aspect of the security?

## 8 Other Requirements

### 8.1 Technology Requirements

Enter here any specific technology requirements, such as .NET or SQL Server.

### 8.2 Data Migration

If the client has existing data that should be migrated, mention this here.

### 8.3 System Conventions

If the system is required to use a specific convention such as an interface template or a specific CSS, mention this here. Any UI specific standards related to the client organization should be captured here. Examples:

* System must use the client’s standard UI conventions: look and feel.
* The client’s logo must appear on all pages.
* The site shall list 10 records per page.
* Next and Previous links shall be provided to move between pages.
* All error messages shall be displayed in red.
* Titles shall have this format “Site Name:Page Name”.

### 8.4 Usability

If the customer expresses any requirements about the system usability, mention them here.

Example:

* Users shall be able to understand the interface without external guidance.
* The length of all system messages must be less than three sentences.

### 8.5 Compliance Requirements

Put here any standards that the system should comply with.

### 8.6 Availability

When applicable, mention:

* Acceptable percentage of availability
* Tolerated down time
* Meantime between failures and to recover or repair

### 8.7 training and Documentation Requirements

List the training or documentation required by the customer e.g. user manual, installation guide, Online Help, etc.

# Section IV: Appendices

## Appendix I: Dropped/Changed Requirements

If a requirement was dropped or changed, move it here along with its number to keep the history (archive), and put the new version in its original place. Requirements could be dropped or changed by the client or be the development team for many reasons such as complexity, time constraints, or else. If the dropped requirements are going to be included in future releases, add a section for that release.

In the original place, mention something like the following:

“This is a newer version of the requirement. The requirement was dropped/changed on xx/xx/xxxx by XXX based on CR # XXX. To check the previous version, go to Appendix I: Dropped/Changed Requirements.”

Notes: After sign-off, never re-assign the number of the dropped requirement to another one.

Move original requirements here ONLY after the requirements have been baselined. Before the baseline (customer sign off), no need to track requirements changes.

## Appendix II: Client’s Wish List for Future Releases

List any requirements that the customer expressed interest in, after the scope has been defined, but were out of the project (or release) scope. If no agreement is reached between all involved parties to change the scope, keep track of these requests by listing them here for future releases.

## Appendix III: Sample Documents

If you have sample documents of the client hard material, such as invoices or forms, scan them and add them here.

## Appendix IV: Open/Closed Issues

This appendix is for issues that relate to requirements only.

You can use this appendix to detail requirements that are not agreed yet. Not agreed requirements should not be included in the body of the requirements document until their issues are resolved.

In the following table, move issues that were closed and didn’t become part of the requirements.

|  |  |  |  |
| --- | --- | --- | --- |
| **Closed Issue** | **Closed Date** | **Decision Makers** | **Justification** |
|  |  |  |  |

If any of these appendices become too big, move them to another document and mention it in the Document Context section.