system requirements specification

<Monqez>, <2.0>

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

<October, 19 2020>

Contents

[Section I: INTRODUCTION 4](#_Toc57213199)

[1 About thIS Document 4](#_Toc57213200)

[1.1 Document Purpose and Scope 4](#_Toc57213201)

[2 About the Project 4](#_Toc57213202)

[2.1 Stakeholders 4](#_Toc57213203)

[2.2 Business Background 4](#_Toc57213204)

[2.3 Project Purpose and Scope 5](#_Toc57213205)

[2.4 Operational Environment 5](#_Toc57213206)

[2.5 Facts, Assumptions, Dependencies, Constraints, Risks 5](#_Toc57213207)

[3 GLOSSARIES of Terms 5](#_Toc57213208)

[Section II: About the SYstem 6](#_Toc57213209)

[1 System Overview 6](#_Toc57213210)

[1.1 MAIN FEatures 6](#_Toc57213211)

[1.2 System Structure 6](#_Toc57213212)

[1.3 Context diagram 7](#_Toc57213213)

[1.4 System functions 7](#_Toc57213214)

[2 User Groups and Access Privileges 8](#_Toc57213215)

[Section III: System REQUIREMENTS 10](#_Toc57213216)

[1 Functional Requirements 10](#_Toc57213217)

[1.1 Overall Processes 10](#_Toc57213218)

[1.2 Use Case Tables 10](#_Toc57213219)

[2 Reports 15](#_Toc57213220)

[2.1 [REP-01: Sales Commission total per month] 15](#_Toc57213221)

[3 Integration/Interfaces 17](#_Toc57213222)

[4 Logging Requirements 17](#_Toc57213223)

[5 Quality Requirements 17](#_Toc57213224)

[5.1 Performance 17](#_Toc57213225)

[5.2 Security 17](#_Toc57213226)

[6 Other Requirements 18](#_Toc57213227)

[6.1 Technology Requirements 18](#_Toc57213228)

[6.2 Usability 18](#_Toc57213229)

[6.3 Availability 18](#_Toc57213230)

[6.4 safety 18](#_Toc57213231)

[6.5 Reliability 18](#_Toc57213232)

[7 Competing softwares 19](#_Toc57213233)

[7.1 New Zealand Red Cross First Aid 19](#_Toc57213234)

[7.1 The E-Medic 19](#_Toc57213235)

# 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

* System shall allow the normal user to call for (connect to) a Monqez when needed so that the Monqez shall arrive to the site.
* System shall allow the normal user to make video and voice call with any available Monqez.
* System shall allow any user type to be able to view the first aid instructions for any type of injuries.
* System shall allow the normal user to order a Monqez for strangers in need of first aid.
* System shall allow normal users to add friends and family in order to fasten the process of asking for a Monqez in case of an emergency.
* System notifies the normal user’s selected emergency list in case any emergency happened to the user.
* System shall allow both the normal user and the Monqez user to make complaints about any incident that happened during fulfilling the request.
* System shall allow the normal user to rate the Monqez that has fulfilled his request.
* System shall allow the Admin user to view the complaints about any specific request.
* System shall allow the Admin user to view the Monqez applications and their certificates.
* System shall allow the Admin user to either approve or deny a Monqez application.
* System shall allow the Admin user to ban any account they find inappropriate.

### 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

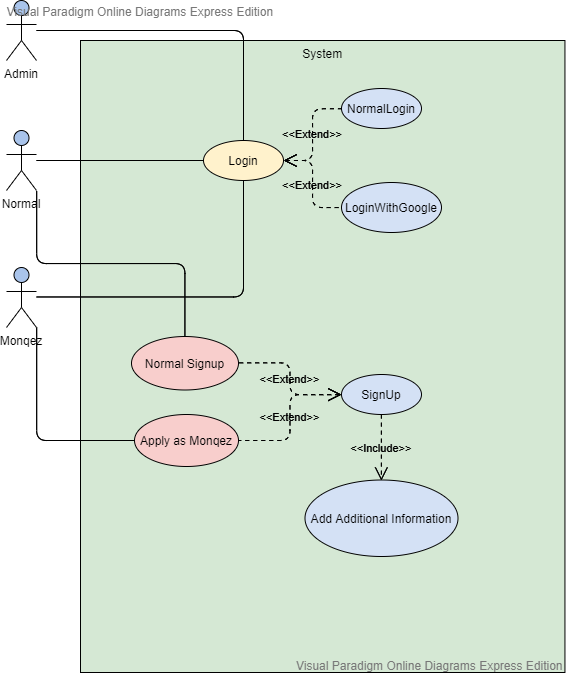
If you can, draw a decomposition diagram like the following:



### 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

## 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 Use Case Tables

|  |  |
| --- | --- |
| Use Case ID: | 1 |
| Use Case Name: | Sign up |
| Brief Description: | User creates a new account in the system. |
| Actors: | Normal user, Helper (Monqez) |
| Preconditions: | No account exists for the user in the database. |
| Postconditions: | A new account is added to the database with the user information. |
| Main Scenario: | 1. System displays the signup page. 2. User enters the credentials (email and password). 3. System stores the credentials. 4. System redirects the user to the “Add Additional Information” page. |
| Exception Flow: | 1. System displays an error message. 2. System displays the signup page. |

|  |  |
| --- | --- |
| Use Case ID: | 2 |
| Use Case Name: | Add additional information |
| Brief Description: | User fills in data about himself into the system. |
| Actors: | Normal user |
| Preconditions: | User is signed up. |
| Postconditions: | Additional information is added to the existing user. |
| Main Scenario: | 1. System displays the “Add Additional Information” page. 2. User enters the required data (full name, phone number, national ID, date of birth, address and gender). 3. System stores the information in the database. 4. System redirects the normal user to his/her home page. |
| Exception Flow: | 1. System displays an error message. |

|  |  |
| --- | --- |
| Use Case ID: | 3 |
| Use Case Name: | Apply as Monqez |
| Brief Description: | Monqez fills the signup application into the system. |
| Actors: | Helper (Monqez) |
| Preconditions: | User is signed up. |
| Postconditions: | Application is submitted and pending approval. |
| Main Scenario: | 1. System displays the “Add Additional Information” page. 2. User enters the required data (full name, phone number, national ID, date of birth, address, gender and first-aid certificate). 3. System stores the information in the database. 4. System redirects to the login page. |
| Exception Flow: | 1. System displays an error message. 2. System displays “Add Additional Information” page. |

|  |  |
| --- | --- |
| Use Case ID: | 4 |
| Use Case Name: | Log in |
| Brief Description: | Actors authenticate and get into the system. |
| Actors: | Normal user, Helper (Monqez), Administrator |
| Preconditions: | User must exist in the database and has not yet authenticated. |
| Postconditions: | User is authenticated and allowed to use the system functions. |
| Main Scenario: | 1. System displays the login page. 2. User enters the credentials (email and password). 3. System validates the credentials. 4. System checks that the user has added his additional information. 5. System displays the appropriate user home page. |
| Alternative Scenario: | 1. System displays the login page. 2. User enters the credentials (email and password). 3. System validates the credentials. 4. System checks and finds that the user has not added his additional information. 5. System displays “Add Additional Information” page. |
| Exception Flow: | 1. System displays an error message. 2. System displays the login page. |

|  |  |
| --- | --- |
| Use Case ID: | 5 |
| Use Case Name: | Log in with Google |
| Brief Description: | Actors authenticate and get into the system. |
| Actors: | Normal user, Helper (Monqez), Administrator |
| Preconditions: | - |
| Postconditions: | User is authenticated and allowed to use the system functions. |
| Main Scenario: | 1. System displays the login page. 2. User selects his Google account. 3. System validates the credentials. 4. System checks that the user has added his additional information. 5. System displays the appropriate user home page. |
| Alternative Scenario: | 1. System displays the login page. 2. User selects his Google account. 3. System validates the credentials. 4. System checks and finds that the user has not added his additional information. 5. System displays “Add Additional Information” page. |
| Exception Flow: | 1. System displays an error message. |

|  |  |
| --- | --- |
| Use Case ID: | 6 |
| Use Case Name: | Sign up with Google |
| Brief Description: | User creates a new account in the system. |
| Actors: | Normal user, Helper (Monqez) |
| Preconditions: | No account exists for the user in the database. |
| Postconditions: | User is authenticated and a new account is added to the database with the user information. |
| Main Scenario: | 1. System displays the login page. 2. User selects his Google account. 3. System validates the credentials. 4. System checks and finds that the user has not added his additional information. 5. System displays the “Add Additional Information” page. |
| Alternative Scenario: | 1. System displays the login page. 2. User selects his Google account. 3. System validates the credentials. 4. System checks that the user has added his additional information. 5. System displays the appropriate user home page. |
| Exception Flow: | 1. System displays an error message. |

## 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 Integration/Interfaces

The system will interact with Google Maps API:

* Displaying the normal user’s home page initializes the connection between the software and the Google maps API and when the Monqez user receives a request to go to a specific location.
* The longitude and latitude data of the user’s current position will be received as double from the API as well as the map.

Exception handling:

The system will interact with Google’s Firebase Authentication API:

* The connection will be initiated with the firebase as soon as the application starts running.
* The data elements exchanged will be either an access token in case of choosing Google authentication that will be sent to the Firebase Authentication server, or it could be email and password as strings and receives an access token.
* The system will write the last login date into the Firebase Authentication server as well as writes the username and password (encrypted) to the Firebase Authentication server in case of signing up.

Exception handling:

* If login credentials not correct, an error message is returned.

## 4 Logging Requirements

The system stores all the requests made by the Normal user, its date, its location, the Normal user’s rating, the Monqez that accepted the request as well as any complaint made on that specific request. These data will be stored in real-time as soon as the request is fulfilled.

## 5 Quality Requirements

### 5.1 Performance

The normal user request should be matched as quickly as possible since the duration is an important factor to save the patients’ lives. (At most 2 minutes)

The server response time must not exceed 5 seconds.

Exception Handling

If the number of users exceeds the maximum, a message may appear: System is temporarily unavailable, try again later).

### 5.2 Security

Authentication

* The system will authenticate users by their passwords or Google account.
* The password minimum length is 8 characters and must include at least one capital letter, one small letter, a number, and a special character.

Sensitive Data

* The user’s national ID number should be protected and encrypted.
* National ID must be protected against update and delete.
* 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?

## 6 Other Requirements

### 6.1 Technology Requirements

* Flutter
* Dart
* Node JS
* JavaScript
* Firebase Authentication
* Firebase Realtime database

### 6.2 Usability

* Users shall be able to understand the interface without external guidance.
* The ability of the program to do all the basic features by 3 touches as a maximum.

### 6.3 Availability

* The application should always be available to serve the users at least 95% of the time.
* The maximum tolerated down time is 72 minutes per day.
* The meantime between failure and to recover is 30 minutes.

### 6.4 safety

* During the registration process, the user is asked to attach a photo of the national id to be used in the event that anyone does anything illegal or unethical.

### 6.5 Reliability

* The ability of the program to work with full functionality in worst cases. (Peak Demand).

## 7 Competing softwares

### First Aid & Emergency – Apps on Google Play7.1 New Zealand Red Cross First Aid

* It contains step-by-step instructions for many emergencies, including CPR, what to do about a burn injury, and many other medical emergencies like heart attacks, open wounds, and more.
* It doesn't send a medic

### هيئة الهلال الأحمر السعودي7.1 The E-Medic

* The application allows receiving any new communication, as well as accepting or canceling the report, drawing the best paths to the location of the report and determining it, recording all notes on it, as well as receiving any amendment to the report from the operating room so that the operating room is aware of the emergency teams and following them on an ongoing basis.
* The application also allows emergency teams to communicate emergency cases with hospitals and choose the most appropriate hospital for the case.
* It does not send a paramedic to do first aid, rather it sends an ambulance.