

# **Requirements Document**

## **Emergency Room Flow Manager**

**Axios**

**January 17, 2023**

# Contents

<b>Revision History</b>	<b>4</b>
<b>1 Introduction</b>	<b>5</b>
1.1 Purpose	5
1.2 Project Scope	5
1.3 Glossary of Terms	5
1.4 Overview	5
<b>2 Overall Description</b>	<b>6</b>
2.1 Product Perspective	6
2.2 Product Features	6
2.3 User Classes and Characteristics	6
2.4 Operating Environment	6
2.5 Design and Implementation Constraints	6
2.6 Assumptions and dependencies	7
2.6.1 Assumptions	7
2.6.2 Dependencies	7
<b>3 System Features</b>	<b>7</b>
3.1 Display Queue Waiting Times	7
3.1.1 Description and Priority	7
3.1.2 Functional Requirements	7
3.2 Display The Number of Patients in Queue	8
3.2.1 Description and Priority	8
3.2.2 Functional Requirements	8
3.3 Submit a Form to Update Queue Information	8
3.3.1 Description and Priority	8
3.3.2 Functional Requirements	8
3.4 Display Historical Data	8
3.4.1 Description and Priority	8
3.4.2 Functional Requirements	8
<b>4 External Interface Requirements</b>	<b>9</b>
4.1 User Interfaces	9
4.2 Hardware Interfaces	9
4.3 Software Interfaces	9
4.4 Communications Interfaces	9
<b>5 Other Non Functional Requirements</b>	<b>9</b>
5.1 Performance Requirements	9
5.2 Security Requirements	9
5.3 Software Quality Attributes	10
<b>6 Other Requirements</b>	<b>10</b>
6.1 Data Retention	10

6.2 Data Retrieval Grace Period	10
<b>Appendix: Issues List</b>	<b>10</b>

## Revision History

Name	Date	Reason for Changes	Version
All	27.01.23	Skeleton Document	0.0
Bhavanvir	01.02.23	1.1-1.5 Drafting	0.1.0
Berkan	02.02.23	3.1-3.4 Drafting	0.1.1
Bhavanvir	02.02.23	2.1-2.3 Drafting	0.1.2
Dewei	02.02.23	4.1-4.4 Drafting	0.1.3
Dewei	02.02.23	6.1-6.2 Drafting	0.1.4
Jim	02.02.23	5.1-5.3, 2.4-2.5 Drafting	0.1.5
Bhavanvir	07.02.23	Proofreading and Editing	0.1.6
Jim	08.02.23	Slight Edits	0.1.7
Berkan	09.02.23	Editing	0.1.8
Bhavanvir, Jim	03.03.23	Updating Document Based on Client Team Requests	1.1

# 1 Introduction

## 1.1 Purpose

Requirements Document (henceforth known as RD) 1.0 describes the functionality of Emergency Room Flow Manager (henceforth known as ERFM), a software system to display information regarding emergency and non-emergency wait times and queue length in hospitals, as well as providing individuals with a means of reporting current conditions, all consolidated within an intuitive web application, primarily for use by the general public. Axios (henceforth known as the Client Organization or Clients) finds there to be a lack of information surrounding conditions within a hospital, giving rise to exponentially increasing wait times for those attending the emergency and non-emergency sections, and subsequently resulting in declining public confidence due to expectations in response times for treatment, so a new system must be designed to address these issues and consolidate the solution into a single web application.

## 1.2 Project Scope

The project scope is to help set expectations for individuals needing to attend a hospital for emergency and non-emergency visits and, in the process, reduce anxieties associated with the unknown nature of wait times, allowing for a more linear experience. Having this information displayed in an intuitive and simple web application where individuals quickly view vital information during a potentially time-sensitive period alleviates the anxieties associated with such events, allowing more energy to be allocated towards other aspects of the hospital visit: individuals already in the queue can report their position in line, allowing for accurate crowd-sourced data for others to view. This system will address the dwindling public confidence in an already under-represented healthcare system while simultaneously being more transparent in typical visit durations, allowing individuals to make informed decisions.

## 1.3 Glossary of Terms

<b>RD</b>	<b>Requirements Document</b>
<b>ERFM</b>	<b>Emergency Room Flow Manager</b>
<b>Clients</b>	<b>Axios</b>
<b>Client Organization</b>	
<b>Designers</b>	<b>The team responsible for the design and development of the ERFM</b>
<b>Designer Organization</b>	
<b>Individuals</b>	<b>Members of the general public who are the intended user group of the ERFM web application</b>
<b>EHR</b>	<b>Electronic Health Record</b>
<b>API</b>	<b>Application Programming Interface</b>
<b>UI</b>	<b>User Interface</b>

## 1.4 Overview

RD 1.0 begins by providing an overall description, included in this section are: the Product Perspective, the Product Features, User Classes and Characteristics, the Operating Environment, Design and Implementation Constraints, and Assumptions and Dependencies. Secondly, RD 1.0 overviews the

Functional Requirements of the system, covering four requirements: Order information Interface, Managing Orders, Wait Times, and Menu Items. Next, the External Interface Requirements section defines requirements for the User Interface, the Hardware Interface, the Software Interface, and the Communication Interface. Non-Functional Requirements follow, defining Performance Requirements, Security Requirements, and Software Quality Attributes. Finally, other requirements that have yet to be defined will be included in RD 1.1, and an Appendix issues list.

## **2 Overall Description**

### **2.1 Product Perspective**

ERFM, the system being developed, is an entirely new platform built for individuals by individuals. The perspective for the solution is listed below:

- The product will run as a single web application to allow for mass adoption and availability to members of the general public.
- The product will communicate with third-party services via API calls to allow viewers of the web application to gain a breadth of information about hospital services and availability.
- The product will not interact with the users in the form of cookie collection or any form in which data retention could result in the anonymity of a user being compromised.
- The product will increase the public trust of surrounding health authorities by providing users of the web application with authentic and accurate crowd-sourced data on the wait times of the emergency and non-emergency sections of hospitals.
- The product will retain data about historical visit times and volume for better prediction and crowd-sourced data pruning to maximize the accuracy of results.

### **2.2 Product Features**

The product will consolidate interactions between individuals and hospitals into one central web application. The product will allow individuals to gain access to information regarding hospital wait times in the emergency and non-emergency sections, and current queue lengths, as well as the extrapolation of historical data collected via crowd-sourcing; individuals currently in the queue can report the number of people directly in front of them, which is then reflected the web application after a series of validation steps.

### **2.3 User Classes and Characteristics**

The system is not divided into user classes, as there is only one type of user: individuals. Individuals only have access to viewing information on hospitals in and around their area, as well as being able to provide data in the form entering the current queue length. All of the moderation and validity checkings are done automatically through algorithms; there is no account creation permitted on the platform, and thus no administration role is needed either.

### **2.4 Operating Environment**

- Hardware platform – The system should be available for use on users' computers and mobile devices.
- Operating system and version – The web application will support the latest versions of Windows, Linux, macOS, IOS, Android, and Tizen.
- Web browsers and version – The web application will support the latest versions of Google, Firefox, and Safari.

### **2.5 Design and Implementation Constraints**

#### *Budget*

This project is to be completed within a \$400,000 budget.

#### *Timeline*

This project must be completed by April 2023.

#### *Web application*

The product is constrained to a web application development environment.

## **2.6 Assumptions and dependencies**

### **2.6.1 Assumptions**

#### *Design and Client Teams availability*

All members of both the design and client teams are available during the project's allotted period to complete assigned tasks. If any discrepancies should occur during the development process, the client team should be available for further elicitation.

#### *Consistent Internet Connectivity*

The application will have access to a steady and consistent internet connection at all times during customer interaction.

#### *Scheduling*

Every feature listed below will be completed in the specified time frame.

#### *Internal Data Storage*

Hospital data will be stored on a cloud platform and have sufficient storage to accommodate an increase in user information.

#### *Peer-to-Peer Support*

Web application users shouldn't be able to view the information of other users on the platform

#### *User Concurrency*

Servers are able to support an influx of concurrent users.

### **2.6.2 Dependencies**

#### *Third party services*

Third-party APIs will be included in the product to provide more information to users pertaining to hospital locations and services.

## **3 System Features**

The ERFM software system consists of four main features required for the functionality of the product; each feature is given a priority rating of high, medium, or low. The features, along with their description, priority, and functional requirements, are outlined below.

### **3.1 Display Queue Waiting Times**

#### **3.1.1 Description and Priority**

Patients would like to know how busy a hospital is if they do not have an appointment. By visiting our web application, patients should be able to view the estimated emergency and non-emergency wait times at the hospital they have chosen. This feature helps patients not to waste their time at hospitals as well as avoid any anxieties caused by the nature of long wait times. Aiding patients to avoid long wait times is one of the main objectives of our product. However, because this is an estimated time and it is not fully precise, this feature is of **medium priority**.

#### **3.1.2 Functional Requirements**

REQ-1: Users must be able to view all the hospitals in their area.

REQ-2: Users must be able to choose a hospital from the listed hospitals.

REQ-3: Users should be able to view data regarding non-emergency wait times.

REQ-4: Users should be able to view data regarding emergency wait times.

REQ-5: Users should be able use the web application through their mobile device(s).

## **3.2 Display The Number of Patients in Queue**

### **3.2.1 Description and Priority**

Patients would also like to know how many people are ahead of them in the queue in case of an error in estimating the waiting times. By visiting our web application, patients can view the number of patients waiting in the queue at the hospital they have chosen.

This feature provides more precise data compared to the estimated queue waiting times, and the patients would know exactly what to expect instead of an estimated wait time. Providing the most accurate data possible to our users is also our project's essential objective; therefore, this feature is of **high priority**.

### **3.2.2 Functional Requirements**

REQ-1: Users must be able to view all the hospitals in their area.

REQ-2: Users must be able to choose a hospital from the listed hospitals.

REQ-3: Users should be able to view data regarding the number of patients in queue.

## **3.3 Submit a Form to Update Queue Information**

### **3.3.1 Description and Priority**

For patients to view the queue waiting times and the number of patients in queues without being present at a hospital, the other patients who are currently at the hospital and waiting in the queue can submit a form; thus, the queue data can be updated. The validity of this webform is significant as the accuracy of the data shown to users will depend on this feature; therefore, this feature is of **high priority**.

### **3.3.2 Functional Requirements**

REQ-1: Users must be able to view all the hospitals in their area, and choose a hospital from the listed hospitals.

REQ-2: Users must open the form and fill it in with the accurate information.

REQ-3: The data on the form must be validated.

REQ-4 : If the validity of data is verified, then the queue information must be updated.

## **3.4 Display Historical Data**

### **3.4.1 Description and Priority**

Patients would like to know how good a hospital is based on the hospital rating. By visiting our web application, patients should be able to view the rating of a hospital they have chosen. This feature helps patients in decision-making when they choose a hospital, and it is a **low priority**, as our main focus is to help patients avoid long queues in hospitals.

### **3.4.2 Functional Requirements**

REQ-1: Users must be able to view all the hospitals in their area.

REQ-2: Users must be able to choose a hospital from the listed hospitals.



REQ-3: Users should click on the Information Symbol to view the historic data.

## **4 External Interface Requirements**

### **4.1 User Interfaces**

The ERFM serves as a way to help set expectations for individuals needing to attend a hospital for emergency or non-emergency visits and reduce anxieties associated with the unknown nature of wait times. The ERFM should be able to access through different web browsers on Google, Firefox and Safari. Individuals would know the average waiting time in the emergency room when they are planning to visit the designated hospitals. It would display which hospital has the least waiting time in their geolocation.

### **4.2 Hardware Interfaces**

Regarding privacy laws, company restrictions and data protection, the individual would only allow access to present-day data. It would display the average waiting time for each hospital and doctors' availability for the individual visiting the hospitals. It only serves as a suggestable waiting time for the individuals; information regarding hospital wait times in the emergency and non-emergency sections, and current queue lengths are filled by a validated form submitted by individuals already in the hospitals.

### **4.3 Software Interfaces**

Third-party technology would be used as API calls to allow viewers of the web application to gain a breadth of information about hospital services and availability. It would ask the individuals to fill out a form about how many people are in front of them and how they would rate the hospital's response time. The use of an external software interface must obey the rules of the relevant laws and copyright infringement used by federal laws and Axios company policies.

### **4.4 Communications Interfaces**

The assistant software will be able to make communication with the third-party API by utilizing the online form to indicate the average queuing time of a hospital and doctors' availability when they decide to go to the hospitals. The queuing time would automatically be updated every time a new submission form had filled out, and the admin would get notified the numbers had increased and the queuing time had increased.

## **5 Other Non Functional Requirements**

### **5.1 Performance Requirements**

ERFM will support at least 5000 simultaneous users without slowing down. The system will have to support large surges of users due to the nature and urgency of the situation the system addresses. These visits could be a critical difference in a person's life, so the system should be able to handle many clients at the same time. The system should also be available 99.9% of the time due to similar urgency reasons.

### **5.2 Security Requirements**

To maintain strict confidentiality and privacy requirements, ERFM will not communicate with any hospital's internal Electronic Health Record systems; this ensures that there is no possible way of accessing sensitive information from a hospital's EHR through ERFM. There is also no account creation required on the platform, which also reassures any individuals using the program that no personal information is taken.

### **5.3 Software Quality Attributes**

It is imperative for ERFM to be reliable and correct, which means that the information it provides is accurate and precise. The estimated time and quantity of people in the queue should be accurate within  $\pm$

5 minutes and  $\pm 3$  people. The UI designed for the system must also be usable and verified against the ten usability heuristics. ERFM needs to be easy to use as this platform will be utilized during critical situations. Additionally, the system should stay maintainable to ensure that the system stays active for distressed individuals using the system.

## **6 Other Requirements**

### **6.1 Data Retention**

Data associated with or generated by any given individual will be retained indefinitely for data mining purposes. No personal information is associated with data used for mining, as cookies or other such services are used to track, personalize, and save information about each user's session on the website.

### **6.2 Data Retrieval Grace Period**

In the event that the Client Organization ceases to pay Axios for their ERFM service, the data contained within the system pertaining to the Client Organization will be available for a reasonable amount of time to be exported; subsequent data will still retain within the Axios database for future model prediction capabilities. The remuneration structure for the project is beyond the scope of this document.

## **Appendix: Issues List**

No issues currently blocking the completion of the RD.