Software Requirement Specifications

KARE Project

Version 1.1 | 4/15/23

Contributors:

Conner Kaul

KARE Team #1

Table of Contents

Contents

Table of Figures

Introduction

Purpose

The purpose of this SRS is to show a detailed description of the KARE project. It will explain all the purposes and the features of the system, as well as what the system both intends to do, and what it will do. The functionality and requirements for the system will also be described. This document is intended for the stakeholders and developers for the system.

Scope

The scope of this project is to design and develop a mobile application for Android and a desktop application for Windows and integrate them with a cloud server. The app will allow for secure administrative access to the software and the child's private data by their parents/legal guardian. The desktop application will allow caregivers to download and visualize data collected by the Android app, making it possible to identify trends over time by using statistical analysis tools on the desktop application on the given data.

Definitions, Acronyms, and Abbreviations

1. SRS: Software Requirements Specification
2. API: Application Programming Interface
3. UI: User Interface
4. UX: User Experience
5. OS: Operating System i.e., Windows, Android OS, etc.
6. MCC: Medically complex children: children who have multiple medical conditions that require specialty care.
7. Host / Host Device: this term refers to the device that the user is utilizing to run the KARE system. e.g., desktop computer, cellphone, laptop, etc.

Overview of Document

The requirements section of the SRS covers the Overall Description of the KARE system including functional requirements which are used to establish a context for the non-functional requirements. The non-functional requirements section is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Problem Statement

The problem KARE seeks to make a difference with is to ease the transition for parents of medically complex children once they bring their child home after long-term hospital care. These are incredibly sick children who see three or more specialists in a hospital for adverse medical conditions. In a hospital environment, the child's condition would be extensively documented by doctors, and much of this documentation must continue to monitor their progress and identify the cause of any issues after they are discharged. However, when the children are released with their parents, the parents lack any sort of system to continue this monitoring at home, particularly those who are low-income, and due to the complexity of these conditions it is understandable that many parents become overwhelmed. In extreme cases, this can tragically lead to well-meaning parents losing custody of their child if they cannot provide an acceptable standard of care. KARE seeks to bridge this gap between hospital and home care by providing a simple interface for parents/legal guardians to log the necessary documentation and streamline data collection without the need for paper charts, receive notifications on when to administer medication or food, and track the progress of various therapies, all within a single Android app. This app will be installed on an Android device that can follow the child everywhere through the child's parent/legal guardian, and therefore data entry can be made by a variety of caretakers, including doctors, nurses, and therapists. However, the parents/legal guardians alone will have administrative access to the software and their child's private data. Another challenge inherent in tracking complex medical data using paper charts is the inability to make changes to the data or extrapolate the data into different contexts. The second component of KARE, an associated desktop application, will combat this challenge. The desktop app will allow the parent or legal guardian to download the data collected by the Android app and create data visualizations of the measured attributes.

Requirements

Mobile

Functional Requirements

Security

One of the main reasons that the mobile app needs to have a focus of security is due to it being related to medical information. This includes following the regulations regarding HIPAA. In order to have security the app will have a secure account, which includes logging in and out and creating accounts with a password. Another aspect is profile security, this includes account creation that has customizable accessibility for items such as: feeding, medicine, therapy. When each caregiver selects a profile, they must enter their correct passcode before gaining access.

Scheduling

The ability to schedule events. This schedule must be sorted by time and if the task has been completed or not. The ability to quickly filter by event category. A quick method to show the event information, log the event, or delete it. There should also be a way to schedule routine or repeating events.

Logging

For logging, there are two main categories: Scheduled and Spontaneous. For the scheduled events this includes feeding, medicine, and therapy activities. For feeding the data that needs to be recorded is date, time, name, dose, rate, and notes. For medicine this needs to include date, time, name, prescriber, dose, note. For therapy activities this includes date, time, category, provider, description, rating (1-5), duration (minutes), and instructions. While the spontaneous events it has to do with items related to home care, such as, temperature, diaper, emesis, gas, GTube, inhalation (Oxygen), eye drops, height, weight. Furthermore, there should be timer functionality to help caregivers record accurate timeframes. Recording this information should have the ability to input the data using Imperial or Metric data when applicable.

The logging should include information such as date, time, note, and other unique information according to relevant events.

Non-functional Requirements

Usability

The software must have a clean, intuitive, and user-friendly interface that allows users to easily access and navigate the various features. The software must also provide helpful error messages to guide users in case of any issues. The software must be designed with sufficient color contrast to ensure that all content is easily readable for users with average vision, and accessible for users with slightly impaired vision. The software must also implement a timer feature to aid parents/legal guardians in collecting accurate time data.

Security

The app should have some form of authentication and require users to create a secure login with a unique email and password to ensure that only authorized individuals can access the app and view personal health information. The app should also have some form of authorization to ensure that different users only have access to the data and functions that are relevant to them. By implementing role-based access control we can restrict access based on permissions. Once the users select their profile, they would need to enter in their valid passcode in order to access the account.

Customization

To ensure efficient use of the app. Allowing the primary user to customize the access or roles for user profiles, to make the process of logging events and tasks quicker.

Desktop

Functional Requirements

Data Visualization

The primary feature of the desktop application is the ability to create data visualizations using the data collected by the Android app. The user should be able to select any attribute that is stored in the database and plot information relevant to that attribute. This way, they can view trends over time to draw conclusions about their child's overall health. For example, if they were to select "Temperature" as the attribute to plot, they could generate a line graph or scatter plot showing measured temperatures over a particular date range. The application should include the ability to plot line graphs, scatter plots, scatter plots with a trendline or legend, and bar graphs, as appropriate. The user will have full control over which type of chart they would like to plot; however, some attributes are restricted to certain kinds of plots. Data that is categorical or ordinal should not produce line graphs or scatter plots with a trendline, as the ratio relationship between categories is nonsensical. For example, emesis volume, which is measured in the ordinal categories "Spit up," "Medium," "Big," and "Very Big," should produce only bar graphs (measuring the frequency of each volume category), scatter plots, and scatter plots with a legend that plots the date versus the time of day. The user should also have control over the date range of the data. The application, by default, should display all datapoints for the selected attribute. But if the user desires, they may also reduce the date range to only plot points before and after a certain set of dates. Finally, the desktop application should provide further statistical tools that can create regression lines and find correlations.

Database Access

The desktop application also needs to be able to retrieve data from the user database in order to create these plots. It must connect to the database via the database API, configure the data that is returned, and use the data to create a variety of charts.

Importing from File

The software should be able to generate plots with datasets besides the database. The user should be able to import a JSON file and create the same type of charts as with the database data. The JSON file that is imported is required to be formatted in the same way as the JSON returned by the database.

Adding Entries to Database

The user should be able to log entries into the database in case the mobile app is otherwise unavailable. Every one of the attributes that is available to log in the mobile app should also be available to log in the desktop application. This way, users can add an entry and immediately visualize the data with that additional entry within the same session.

Exporting Data

The data being plotted and the plot itself should be exportable. This way, the data can be sent to doctors or other specialists so they may review and look for trends with their expertise. The data itself should be exportable as a JSON or CSV, and the graphs that are generated should be exportable as a PNG file.

Non-functional Requirements

Security

As with the Android application, there is a need for security due to strict HIPAA regulations. However, because this desktop application is intended for use on a private desktop environment, usable only by the primary guardian(s) of the child who are covered under HIPAA, the security requirement is less stringent that with the Android app, which would be passed around between caretakers who are not necessarily covered under HIPAA. Users will need to log in with their user credentials to use the desktop application and access data from the database.

Usability

Usability is very important for this application, as there is no guarantee that the users will be technologically savvy. The design should be simple, uncluttered, and very easy to use. Furthermore, the data should be easily printable. Finally, the software must be designed with sufficient color contrast to

ensure that all content is easily readable for users with average vision, and accessible for users with slightly impaired vision.

Reliability

The software should have minimal bugs. This application will be working with large amounts of data, so proper error handling is required to deal with potential bad data entries. These erroneous entries should be discarded, and an error message should be presented when such a case occurs.

Compatibility

The desktop application should be Windows compatible.

Database

Functional requirements

Data storage

The software application provides an efficient mechanism for storing data in a database.

Login Details

The software application allows users to create and store login details in the database.

Event submissions

The software application allows users to submit events and store them in the database.

Scheduled Events

The software application allows users to schedule events and store them in the database.

User profiles, pins, and permissions

The software application allows users to store user profiles, pins, and permissions in the database.

Non-functional requirements

Security

The software application implements encryption to secure the data stored in the database.

Connectivity

The software application ensures a secure connection between the client and server by using HTTPS to protect data transmission.

Performance

The software application provides an efficient mechanism for accessing data in the database to ensure optimal performance. The software must be able to handle an increasing number of users and requests without a significant decrease in performance so scalability should be factored in.

High Level Design

Mobile

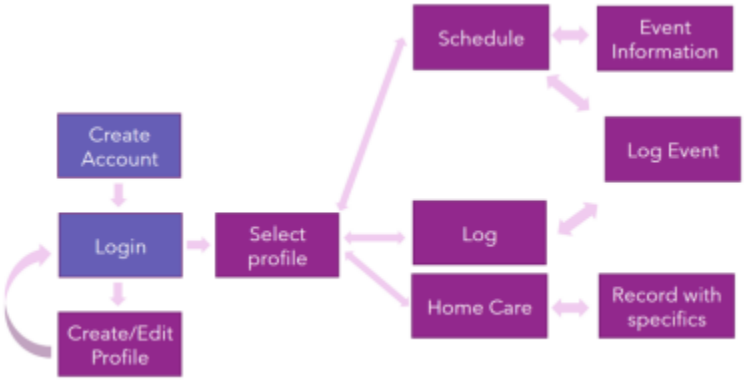

Figure 1: Mobile Design

The mobile application will first prompt the user to either create an account if they do not have one or login with previously created credentials. After the user has logged in, they will be able to create a profile, edit a previously made one, or select a profile. Once the user has selected the profile, they will be able to schedule a future event, log an event that is planned for with the applications functionality, or record an event that is outside of the capability of the app through the home care option.

Desktop

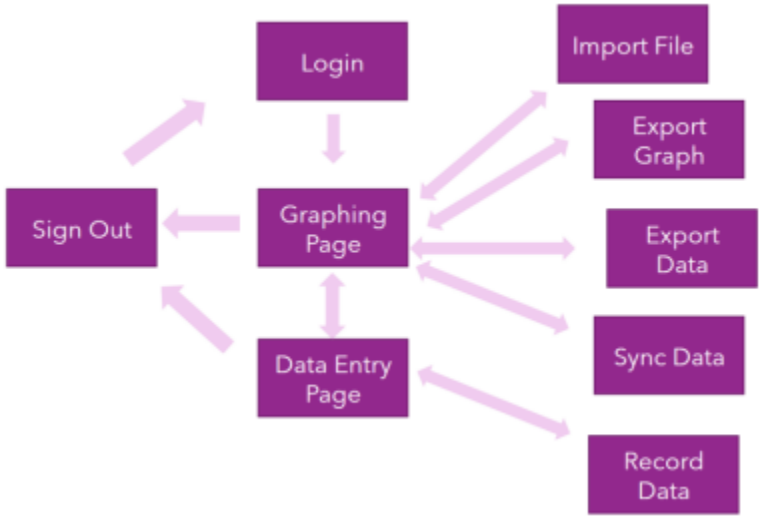

Figure 2: Desktop Design

First on the desktop application will be a login page where the user can enter their credentials that they used to sign up on the mobile Android application. After they have logged in, they will be able to synchronize the data that was entered in on the mobile application through the cloud database. After the data has been synchronized, the user can graph the data and look at various statistical methods like regression lines. From here, the user will also be able to export the statistical graphs or the data itself. The user will also have an option to record data in the case where they cannot access the mobile application.

Database

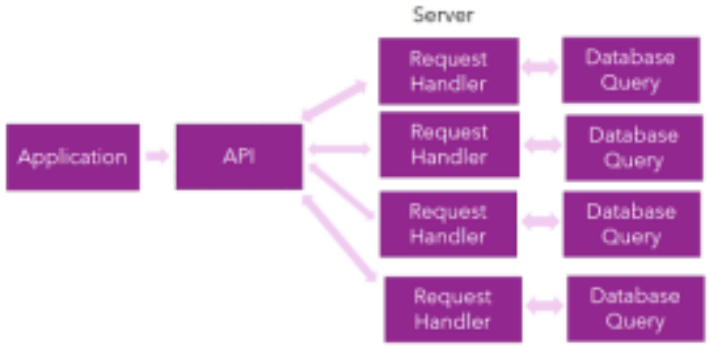

Figure 3: Database Design

The database will receive queries from the cloud server for information regarding the mobile and desktop applications.

Assumptions

Adequate hardware

1. The software is designed to be used on Windows computers and Android mobile devices.

2. The software may not function correctly on older hardware or devices with limited processing power.

Testing Plan

The testing plan for this project will be Agile testing where testing will be integrated into the weekly tasks throughout the development process. This plan was chosen due to the limited amount of time to complete the project so rapid feedback will be beneficial to the developers.

Conclusion

In conclusion, this Software Requirements Specification document serves as a comprehensive guide to the development of the KARE system. It has provided a clear and concise understanding of the system's functional and non-functional requirements, user interfaces, and system requirements. This document has been developed in collaboration with stakeholders, developers, and project managers, to ensure that all requirements are accurately conveyed. The SRS document serves as a reference throughout the development cycle to provide a baseline for testing and verification of the KARE system. This document will be used to assess the software system's compliance with requirements and ensure that it meets the needs of all stakeholders. In summary, this SRS document provides a clear specification of the software system's requirements and high-level designs. Finally, this document will guide the development, testing, and validation activities, ensuring that the KARE system is delivered on time and meets the described requirements of the stakeholders.