Software Requirements Specification

for

EKG Cloud Service

Version 1.0 approved

Prepared by Team Cirrus

Department of Computer Science, LaTech

Oct 18, 2013

Client approval:

Contents

Revision History

Name	Date	Reason For Changes	Version
Team Cirrus	10/03/13	First Draft	1.0
Team Cirrus	10/18/13	Client clarification	1.1

1. Introduction

1.1 Purpose

The present document is the result of collaboration and agreement between the main stakeholders: the project team and the client. It specifies and describes the requirements of the EKG Cloud Service system as a whole. This document is of crucial importance to both the project team and the client. It provides guidance to all the teams involved in the project realisation and assurance to the client.

1.2 Document Conventions

<N/A>

1.3 Intended Audience and Reading Suggestions

The SRS is a central reference document intended for all individuals and teams that play a role in the project. The following sections of the document deal with the project description, the interfaces, the functional and the non-functional requirements. While all the sections are highly relevant to all teams, some sections should be treated more careful by certain teams. The development team must read in detail and continuously refer to both the functional and non-functional requirements and must also be familiar with the interface description. The design team on the other hand should be more concerned with the functional requirements and their integration with the interfaces, focusing on the big picture. The testing team must ensure the quality of the implemented functional requirements and satisfaction of the non-functional requirements. They must know every detail and pay particular attention to the use cases described in the SRS. The testing team will produce a test plan having this document as starting point.

1.4 Product Scope

The EKG Cloud Service aims to provide hospitals and health institutions with a robust and and highly available electrocardiographic data management system. It is a convenient and secure way to store, visualize and make sense of patient EKG data. Instead of reading through long printed logs, the doctors will be able to easily access the EKG information for arbitrary patients. The benefits of this product range from simply EKG data management to diagnosis and disease prevention and ultimately to life saving.

1.5 References

2. Overall Description

2.1 Product Perspective

The EKG Cloud Service is a novel idea that originates in the need of health institutions to centralize patient EKG data and allow the doctors to monitor and review these vital indices. The currently described product is self-contained and restricted to EKG data but has the potential to be integrated with a wide variety of similar patient health data such as body temperature and blood pressure.

2.2 Product Functions

The major functions of the system are listed below with the afferent user class in parentheses

- ➔ user authentication (caretaker, doctor, administrator)
- ➔ CRUD operations on patient personal data (caretaker)
- ➔ patient EKG data upload (caretaker)
- ➔ visualize the EKG data (doctor)
- ➔ add and remove users to the system (administrator)

2.3 User Classes and Characteristics

There are three user classes differentiated by their respective role and privilege level:

- The caretaker class.

This class includes any medical personnel that performs basic duties around the patient and lacks medical expertize. They are able to introduce EKG data into the system but are unable to make sense of it.

- The doctor class.

It consists in the qualified medical personnel with the ability to interpret electrocardiograms. They can view the EKG data of their patients and have access to various reports

- The system administrator class.

Users in this class have the highest privilege level and are responsible with keeping the system in a good shape. Their role is solely administrative and does not require any medical education. They are able to add and remove users as well as to alter the entire database

2.4 Operating Environment

The system will operate on an Amazon EC3 Windows Server 2010 virtual machine. It will run under the XAMPP web server solution stack that consists of the Apache Server, MySQL database and PHP interpreter. The web app itself will be developed on the CodeIgniter framework.

From the client perspective the project must ensure independence in the operating environment. The user will be able to interact with system as long as his device runs one of the main operating systems (i.e. Windows, Linux, Unix, OpenSolaris, iOS) and a web browser.

2.5 Design and Implementation Constraints

The developers will work in compliance with the cloud computing standards and will use exclusively free and open source resources. Portability is an important aspect and therefore a cross-platform server package must be employed. Lastly, the system must be implemented within a well tested and documented framework.

2.6 User Documentation

User documentation will be provided by either a simple help section in the application or a standalone document.

2.7 Assumptions and Dependencies

The project is developed in the cloud and is entirely dependent on the third-party cloud services provider (i.e. Amazon Web Services), the assumption being that Amazon Inc. will maintain a high availability of their services. The EKG database is expected to grow fast, therefore scalability of the Amazon services is yet another assumption.

3. **External Interface Requirements**

3.1 User Interfaces

The user interfaces are illustrated by the following GUI prototypes. Elements visible by caretakers are shown in green, those visible by doctors are blue and those visible to administrators are red.

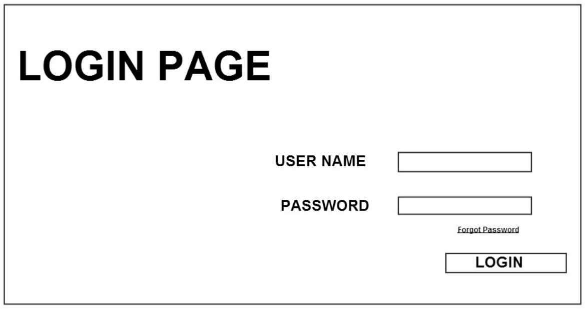


Figure 3.1 The authentication page

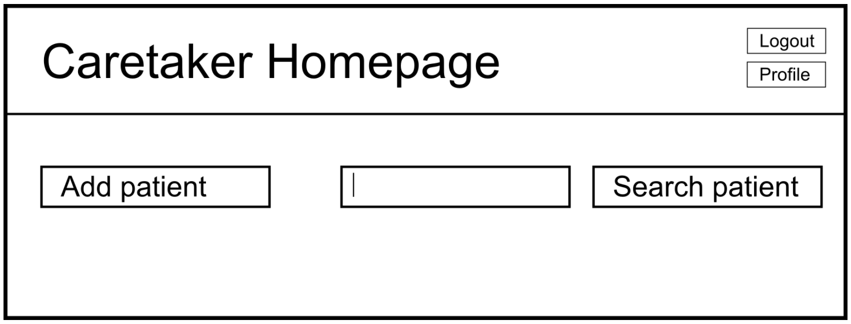


Figure 3.2 The caretaker homepage

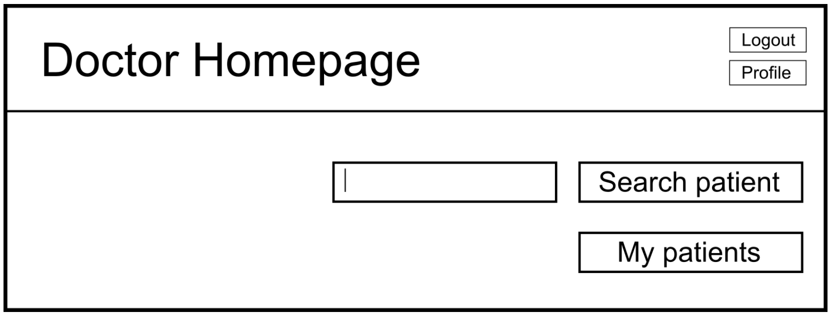


Figure 3.3 Doctor's homepage

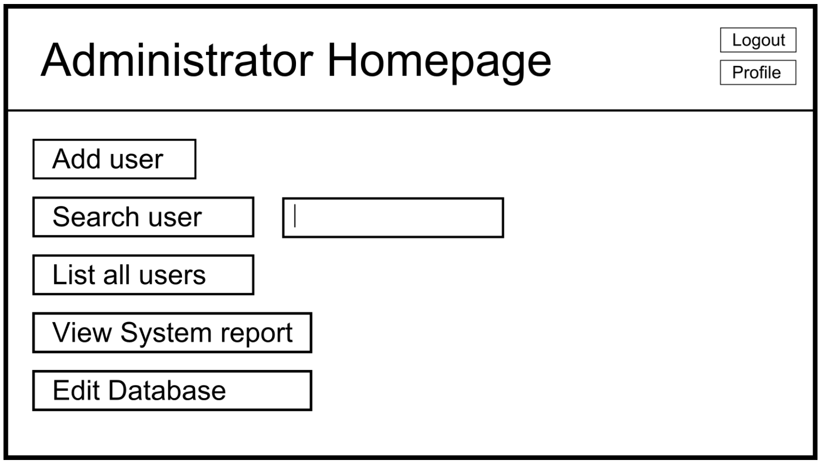
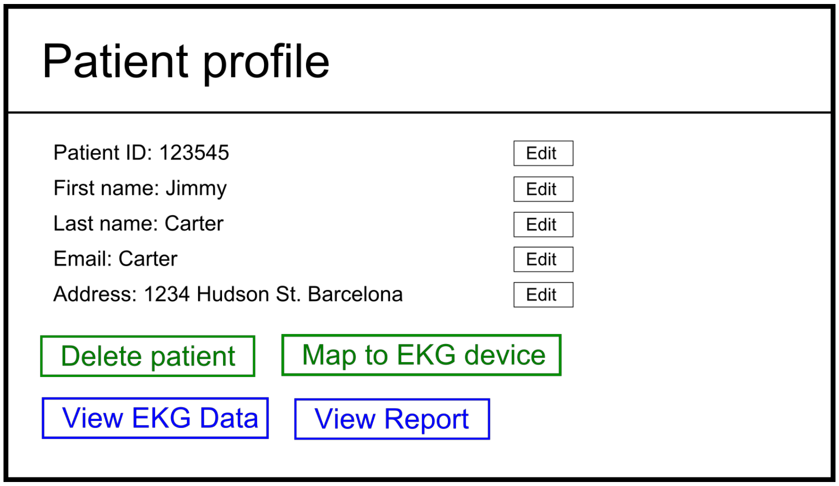
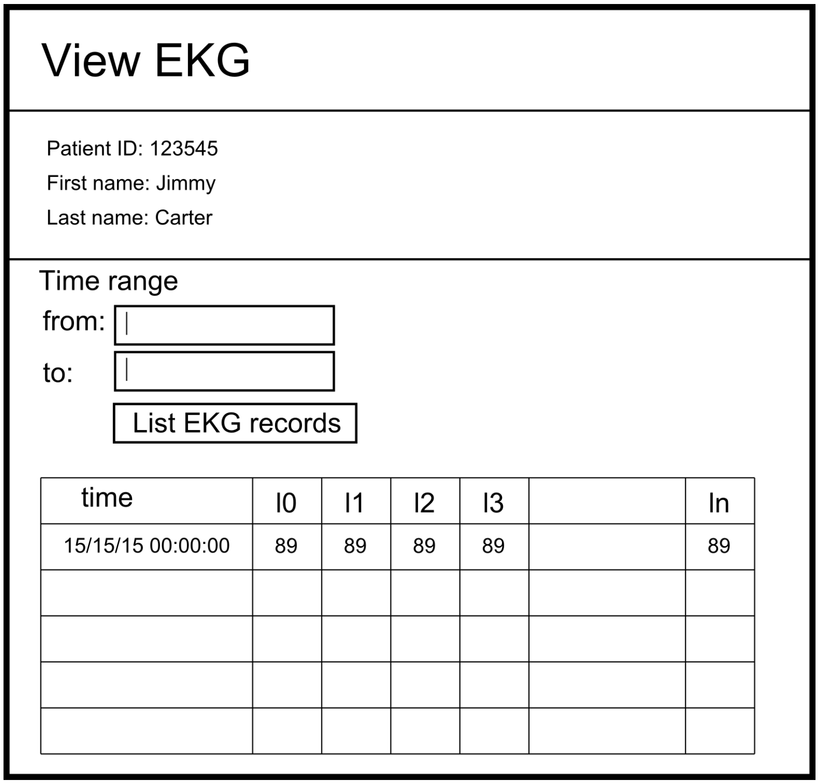


Figure 3.4 Administrator's homepage



Patient ID: 123545
First name: Jimmy
Last name: Carter
Email: Carter
Address: 1234 Hudson St. Barcelona

Figure 3.5 Patient profile page



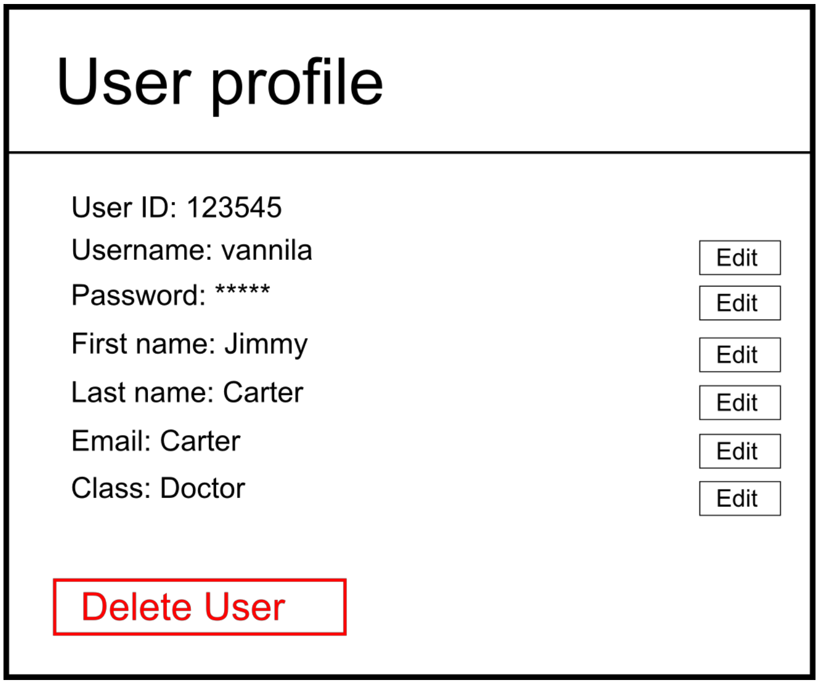
Patient ID: 123545
First name: Jimmy
Last name: Carter

from: |

to: |

time	I0	I1	I2	I3		In
15/15/15 00:00:00	89	89	89	89		89

Figure 3.6 View EKG page



User ID: 123545
Username: vannila
Password: *****
First name: Jimmy
Last name: Carter
Email: Carter
Class: Doctor

Figure 3.7 User profile page

3.2 Hardware Interfaces

There are no specific hardware components of the system as the web application can be accessed from any device running a web browser. Furthermore, the application does not require a specific hardware interface. The device on the client side may feature a keyboard, a touchscreen or any other kind of input interface.

3.3 Software Interfaces

The system may use external web services for rendering the EKG charts. The interfaces are specified by the particular web service that will be employed.

3.4 Communications Interfaces

The communication between the User device and the central system are made through the HTTP/HTTPS protocol over the Internet. The EKG device will sent JSON formatted packets over the same protocol.

4. **Domain Model**

<N/A>

5. **System Features (Use Cases)**

Use Case diagrams

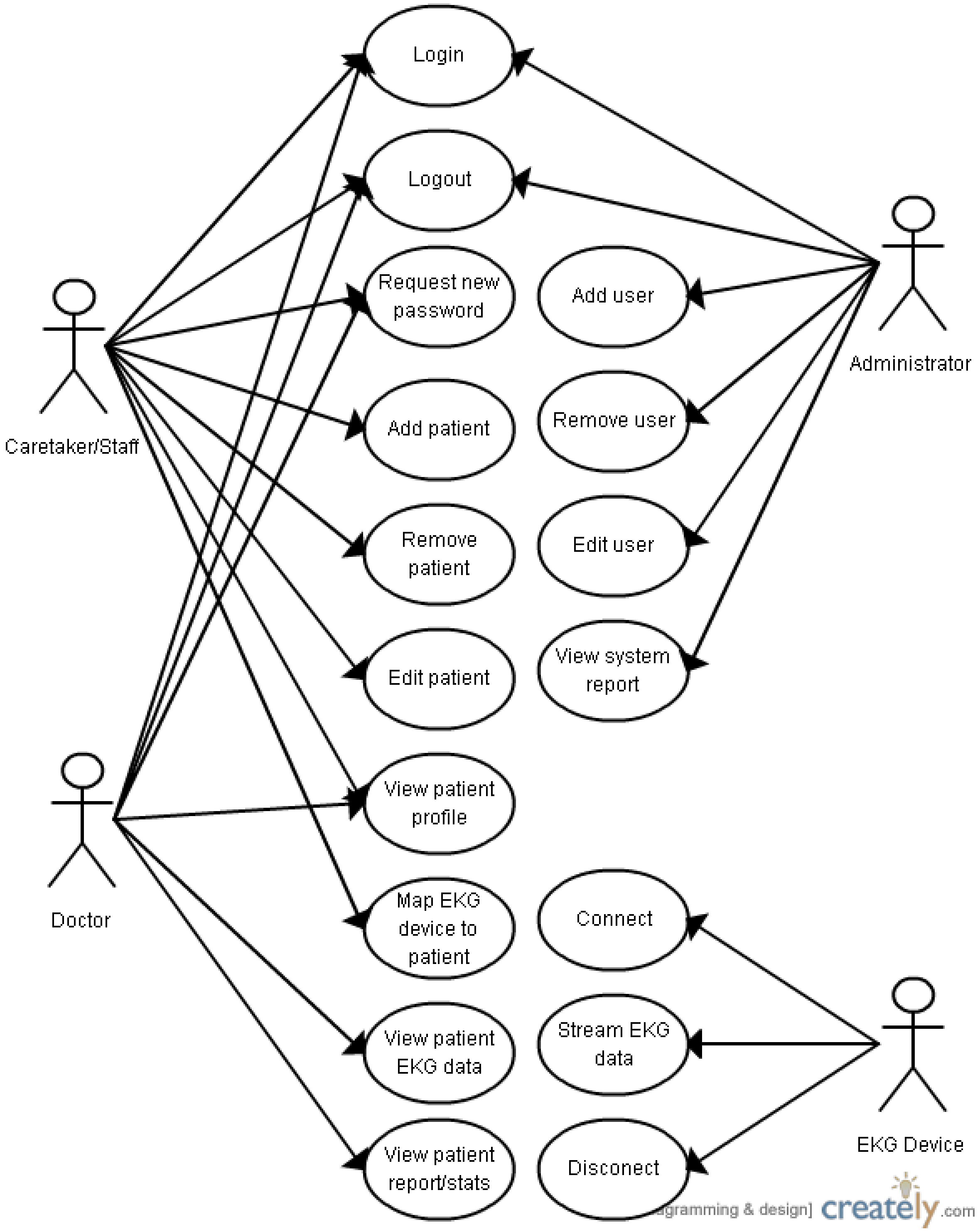


Figure 5.1 General use case diagram

5.1 Log in

Actor: caretaker, doctor, system administrator
Goal: Authenticate a user
Input: The user must input his credentials
Output: The system responds by authenticating the user
Main Scenario: The user enters his credentials correctly and is successfully authenticated
Pre-condition: The user is not yet authenticated
Steps:

- the user clicks the "login" button on the home page and the login form is displayed
- the user introduces his username and password and clicks "login"
- if the username and password are correct, the personal homepage is displayed

Post-condition: The user is authenticated and redirected to his homepage
Exceptional Scenario: nonexistent user name
If the user name does not exist, an appropriate message should be displayed
Exceptional Scenario: wrong password
If the password is incorrect, the system must prompt the user to reenter his password

5.2 Log out

Actor: caretaker, doctor, system administrator
Goal: Authenticate a user
Input: n/a
Output: The system responds by logging out the user
Main Scenario: The user clicks the "logout" button on his homepage and the current session is terminated
Pre-condition: The user is authenticated
Steps:

- the user clicks the "logout" button on the home page
- the session is terminated and the home page is displayed

Post-condition: n/a

5.3 Request a new password

Actor: caretaker, doctor, system administrator
Goal: Provide the user with a mean to access his account if he forgets his password
Input: n/a
Output: A new password is sent to the user's email
Main Scenario: The user selects "forgot password" and receives an automated email with a new password
Pre-condition: n/a
Steps:

- the user selects "forgot password" on the login page
- an email with a new password is sent to the user's registered email and a confirmation message is

displayed
Post-condition: n/a

5.4 Add a new patient

Actor: caretaker
Goal: Add a new patient in the system
Input: The caretaker will introduce the patient name, ID and other optional information (eg age, gender, adress)
Output: The system inserts the new record into the database
Main Scenario: Adding the new patient does not cause any conflicts in the database and is successfully added
Pre-condition: The mandatory fields are present: first name, last name and id
Steps:

- the caretaker clicks the “add patient” button on his homepage
- a form consisting in the patients information is displayed
- the caretaker fills up the form and clicks “submit”
- an adequate message informs the user of the result of his action

Post-condition: the database now contains the new patient
Exceptional Scenario: conflicting ID
If a patient with the same id already exist in the database then prompt the caretaker with an appropriate message

5.5 Remove an existing patient

Actor: caretaker
Goal: Remove an existing patient from the system
Input: The caretaker will introduce the patients ID
Output: The system removes the corresponding record from the database
Main Scenario: The requested patient exists in the database and is successfully removed
Pre-condition: There exists a patient with the requested ID
Steps:

- the caretaker navigates to the desired patients page
- click the “remove patient” button
- a confirmation dialog pops up and the user presses “yes”
- the patient is removed from the system and a confirmation message is displayed

Post-condition: the patient with the requested ID is no longer in the database
Exceptional Scenario: patient not found
If the database does not have the patient ID the system will notify the caretaker and will request a new ID.

5.6 View patient information

Actor: caretaker, doctor
Goal: View patient information in the system.

Input: The care taker will input a patient id
Output: The system displays the patient information to caretaker
Main Scenario: The requested patient exists in the database and its information is displayed
Pre-condition: There exists a patient with patient Id whose information is already stored in database.
Steps:

- the user types the patients ID in the search box. Alternatively a doctor can list all his patients and select the desired patient from that list.
- the patients profile page is displayed

Post-condition: <N/A>.
Exceptional Scenario: patient not found
If the database does not have the patient ID, the system will notify the caretaker and will request a new ID.

5.7 Edit patient information

Actor: caretaker
Goal: Edit patients information in the system.
Input: The caretaker will input the patient ID
Output: The system makes corresponding changes to patients information.
Main Scenario: The requested patient exists in the database and its information is displayed
Pre-condition: There exists a patient with the requested ID.
Steps:

- the user navigates to the patients profile (see use case 5.5)
- clicks the “edit” button next to the desired field
- a dialog pops up and asks for a new value and having two buttons: “ok” and “cancel”
- the user introduces a new value and presses “ok”

Post-condition: changes made to patient information are reflected in the database.
Exceptional Scenario: patient not found
If the database does not have the patient ID, the system will notify the caretaker and will request a new ID.

5.8 Map a device to a patient

Actor: caretaker
Goal: Map the unique device ID to a patient
Input: The caretaker will input the device ID in a patients profile page
Output: The device is mapped to the patient
Main Scenario: The device is successfully mapped to the patent's ID.
Pre-condition: There exists a patient with the specified ID
Steps:

- navigate to a patient’s profile page
- click the “register device” button
- enter the device ID and press “ok”

Post-condition: The device id is assigned to the desired patient
Exceptional Scenario:invalid device ID
If the device id is invalid (eg. in the wrong format) the user is warned and prompted to reenter it.

5.9 View patient EKG information

Actor: doctor
Goal: Visualize the EKG data for a certain patient and within a time interval
Input: The patient ID and a time interval
Output: The patients EKG data that for the specified time interval
Main Scenario: EKG data is available for the specified patient and time interval
Pre-condition: The patient with the specified ID exists in the database, the time interval is valid and EKG data is available
Steps:

- navigate to the patient's profile page (see use case 5.5)
- click the "view EKG data" button
- the "View EKG" page (as in fig. 3.6) is displayed
- specify the time interval and click "list EKG data" button
- the EKG records are listed in a table

Post-condition: the user is looking at the EKG diagram
Exceptional Scenario: nonexistent patient, invalid time interval, unavailable EKG data
If there is no patient with the requested ID, the time interval is invalid (eg. placed in the future) then the system will prompt the user to reenter it
Exceptional Scenario: unavailable EKG data
If there is no available EKG data for the requested patient and time interval then the system will notify the user with an appropriate message.

5.10 View patient report

Actor: doctor
Goal: The doctor should be able to request a report regarding a particular patient. The report may include cardiac anomalies captured in the EKG data with their times of occurrence and health indicators inferred from the EKG data.
Input: The patient ID
Output: A listed report
Main Scenario: The requested report is displayed
Pre-condition: <N/A>
Steps:

- the doctor navigates to the patients profile (see use case 5.5)
- click "view report" button and the report is displayed

Post-condition: <N/A>
Exceptional Scenario: nonexistent patient, invalid time interval, unavailable EKG data
If there is no patient with the requested ID, the time interval is invalid (eg. placed in the future) then the system will prompt the user to reenter it

5.11 Add user to the system

Actor: System Administrator

Goal: Add a new user in the system
Input: The system administrator will introduce a new user ID , password , authorization type and other optional information.
Output: The system inserts the new record into the database
Main Scenario: Adding the new user does not cause any conflicts in the database
Pre-condition: The mandatory fields are present: first name, last name and id
Steps:

- the system administrator selects the “add user” option on his homepage
- a form with the user information is displayed
- fill out the form and click submit
- the user is created and a confirmation message is displayed

Post-condition: the database now contains the new user
Exceptional Scenario: conflicting ID

5.12 Remove user from the system

Actor: System Administrator
Goal: To remove an existing user from the system
Input: The system administrator will introduce the user ID
Output: The system removes the corresponding record from the database
Main Scenario: The requested user exists in the database
Pre-condition: There exists a user with the requested ID
Steps:

- the administrator navigates to a user’s profile page (as sketched in fig. 3.7)
- click the “delete” button and a confirmation dialog pops up
- the administrator accepts the confirmation dialog and the user is removed from the system

Post-condition: The user with the requested ID is no longer in the database
Exceptional Scenario: User not found

5.13 Edit user information

Actor: All
Goal: To modify a user’s information
Input: The system administrator will introduce the user ID
Output: The system removes the corresponding record from the database
Main Scenario: The requested user exists in the database
Pre-condition: There exists a user with the requested ID
Steps:

- the user navigates to his own profile or the administrator navigates to a user’s profile page (as sketched in fig. 3.7)
- press the “edit” button next to the desired field
- a dialog prompts the user for a new value
- the user introduces a value and presses “ok”
- the users profile is updated with the new value

Post-condition: The user with the requested ID is no longer in the database
Exceptional Scenario: User not found

5.14 View system report

Actor: system administrator
Goal: The system administrator must be able to see the system status. Such a report includes statistics regarding the daily throughput, the number of users and the size of the database.
Input: <N/A>
Output: The system report
Main Scenario: The system report is displayed
Pre-condition: <N/A>
Steps:

- the administrator selects "view system report" on his homepage
- the system report is displayed

Post-condition: <N/A>
Exceptional Scenario: <N/A>

5.15 Connect to system (device)

Actor: device (better said, the person handling the device)
Goal: establish a connection between the streaming device and the system
Input: button pressed on the device
Output: n/a
Main Scenario: the device was prior registered to a patient and the connection is succesful
Pre-condition: n/a
Steps:

- push the "connect" button on the device

Post-condition: device is connected and ready to stream
Exceptional Scenario: unregistered device
If the device is not registered then the connection is refused.

5.16 Stream EKG data

Actor: EKG device
Goal: Stream a patients EKG data to the system
Input: push a button
Output: <n/a>
Main Scenario: data starts streaming into the system's database
Pre-condition: the device must be connected to the system
Steps:

- push a "start streaming" button on the device

Post-condition: new EKG data resides in the database

Exceptional Scenario: unconnected device
If the device is not connected, the system will refuse the stream

5.17 Disconnect from the system (device)

Actor: EKG device
Goal: disconnects the device from the system
Input: push a button
Output: <n/a>
Main Scenario: the device is disconnected
Pre-condition: the device is connected to the system
Steps:

- push a “disconnect” button on the device

Post-condition: the device is disconnected from the system

6. **Other Nonfunctional Requirements**

6.1 Performance Requirements

The system must perform satisfactory in the following aspects:

- Reliability. The system must perform in a reliable manner and robustly implement the features described in section 5.
- Responsiveness. The system features described in section 3 are expected to perform in a timely manner and with a prompt feedback.
- Throughput. Depending on the client institution size the rate at which EKG data is uploaded into the system must be proportional to the number of patients.
- Parallel access. Again, depending on the institution size, multiple users should be able to interact with the system in the same time.
- Scalability. The system must have the ability to grow an handle an increasing amount of data.
- Availability. The system must be highly available. It should be available at least 99.99% of the time
- Backup. The underlying database must feature an active backup system.
- Portability. The system must be easily deployed on any major platform.

The throughput , scalability and availability are dependent on the web cloud service provider and it has to do with an assumed scalability.

6.2 Safety Requirements

The system is required to keep logs that would allow to recover any data loss and identify malicious interactions with the system. The database management system must provide data backup functionality and the possibility to rollback.

6.3 Security Requirements

There must be different privilege levels for every user class. The caretaker and doctor user classes have limited access to the system and may perform simple data entry tasks and retrieve certain information. The system administrator is the only user class granted full access to the database. They oversee the activity of the other user classes and prevent malicious tampering with the system. Standard security protocols and encryption must be employed whenever private data is manipulated.

6.4 Software Quality Attributes

Every feature of the system must perform precisely as outlined in the SRS and detailed in the analysis and design documents. Maintainability, robustness and testability must be ensured by using standard coding conventions and by developing the system within a robust framework. Robustness must be further improved by anticipating and appropriately handling errors and exceptions.The user interface must be tailored for each user class according to their particular expertize. Finally, availability is ensured by the Amazon Cloud Services.

7. **Other Requirements**

<N/A>

Appendix A: Glossary

A list of abbreviations and acronyms:

- CRUD = **C**reate, **R**ead, **U**pdate and **D**elete
- XAMPP = is an acronym for: cross-platform (**X**), **A**pache server, **M**ySQL database, **P**HP and **P**earl interpreter

Appendix B: Analysis Models

<N/A>

Appendix C: To Be Determined List

<N/A>