****

**Pflichtenheft**

Pflichtenheft EWS

**Version 1.1**

Historie der Dokumentversionen

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Datum | Autor | Änderungsgrund / Bemerkungen |
| 0.1 | 20.03.18 | Daniel Ladwig | Ersterstellung |
| 1.0 | 24.04.18 | Daniel Ladwig | Neues Artefakt Design |
| 1.1 | 02.06.18 | Daniel Ladwig | Desktopanwendung Spezifikation Anpassung |
|  |  |  |  |

Inhaltsverzeichnis

[Historie der Dokumentversionen 2](#_Toc512359496)

[Inhaltsverzeichnis 2](#_Toc512359497)

[1 Einleitung 3](#_Toc512359498)

[1.1 Arbeitspakete 3](#_Toc512359499)

[1.2 Spezifizierung der Arbeitspakete 3](#_Toc512359500)

[2 Realisierung der Anforderungen 4](#_Toc512359501)

[2.1 Datenbank 4](#_Toc512359502)

[2.2 Server 4](#_Toc512359503)

[2.3 Android App 4](#_Toc512359504)

[2.4 Desktopanwendung 4](#_Toc512359505)

# Einleitung

## Arbeitspakete

* Datenbank
* Server
* App
* Desktop UI

## Spezifizierung der Arbeitspakete

* Datenbank mit SQL Lite auf einem Raspberry Pi
* Back-End Server für die Kommunikation
* Android APP in Java + App Inventor 2
* Windows Programm für die Anzeige der Messergebnisse und der Personalverwaltung in C# mit Visual Studio

# Realisierung der Anforderungen

## Datenbank

* + 1. Für den Betrieb der Datenbank soll ein Raspberry Pi verwendet werden.
    2. Die Kommunikation erfolgt nicht direkt über die Datenbank, sondern über eine   
        Back-End-Software, welche ebenfalls auf dem Raspberry Pi betrieben wird.
    3. Für die Erstellung der Datenbank wird SQLite verwendet.
    4. Für jeden Patienten werden individuelle Grenzwerte für Atemfrequenz, Herzschlag   
        und Blutdruck gespeichert.
    5. Die Server-Anwendung verwaltet Datenbankanfragen sowie die Benachrichtigungen   
        für das medizinische Personal bei kritischen Patienten.
    6. Die Desktopanwendung schickt die Daten an den Server.

## Server

## Android App

* + 1. Übersichtliche Darstellung der Oberfläche durch klare Kennzeichnungen.
    2. Dropdown-Listen zur Vermeidung von Tippfehlern
    3. Bei der Eingabe öffnet sich eine Tastatur, bei der nur Zahlen enthalten sind.
    4. Bei den drei Eingabewerten können nur Zahlen eingegeben werden.
    5. Beschreibung der einzugebende Werte.
    6. Nach dem die Abteilung bzw. die Station in der Dropdown-Liste ausgewählt   
       wurde, werden in der unteren Dropdown-Liste nur die Patienten dieser Abteilung   
       angezeigt.
    7. Die Messwerte können untereinander, in der App eingegeben werden.
    8. Beim Absenden wird die Zeit automatisch erfasst. Die Daten werden beim Absenden  
       an den Server versendet.

## Desktopanwendung

* + 1. Die Desktopanwendung-GUI soll mit Hilfe von Microsoft Visual Studio und   
       entsprechenden Plug-Ins visualisiert werden.   
       Die genutzte Programmiersprache soll C# sein.
    2. Über einen Button soll die Verwaltung aufrufbar sein. Dort können Patienten   
       hinzugefügt, beziehungsweise im Falle einer Entlassung deaktiviert werden.   
       Bei einem erneuten Aufenthalt muss der Patient neu angelegt werden.
    3. Durch die übermittelten Daten soll herausgefunden werden, in welchem Bereich der Patient liegt und entsprechend graphisch umgesetzt werden.
    4. Es wird angezeigt, ob es sich um ein Kind, Erwachsenen oder Senioren handelt.
    5. Die Grenzwerte werden ebenfalls in der Datenbank festgelegt. Die ermittelten Daten sollen mit den individuellen Grenzwerten aus der Datenbank verglichen werden und in die entsprechenden Bereiche eingeordnet werden.
    6. Per Buttondruck soll zwischen Tag- und Nachtmodus gewechselt werden können. Dabei werden die Patienten im roten und gelben Bereich immer angezeigt, die im grünen sind an- und ausschaltbar. Die Benachrichtigung findet ebenfalls über das Back-End Programm per E-Mail statt.