**Pflichtenheft (Grobdesign)**

Vorläufiges Dokument für den Termin Grobdesign am 24.03.18.

Vollständige Traceability, Design und andere Punkte werden in den folgenden Updates erweitert.

**Mitglieder**: Daniel Ladwig (PL), Jenne Hilberts, Rukiye Akin, Mike Durban, Raphael Panter

**Umfeld des Kunden**: Altenheime/Krankenhäuser

**Umfeld der Applikation**: Ein geschlossenes System mit WLAN Abdeckung.

-SQL Lite Datenbank auf einem Raspberry Pi **Verantwortlich** :Jenne Hilberts

-App für Android Handys mit C# und dem Tool App Inventor 2 **Verantwortlich:** Rukiye Akin

-Windows Programm für die Anzeige der Messergebnisse und für die Personalverwaltung. Programmiersprache: Java **Verantwortlich**: Raphael Panter

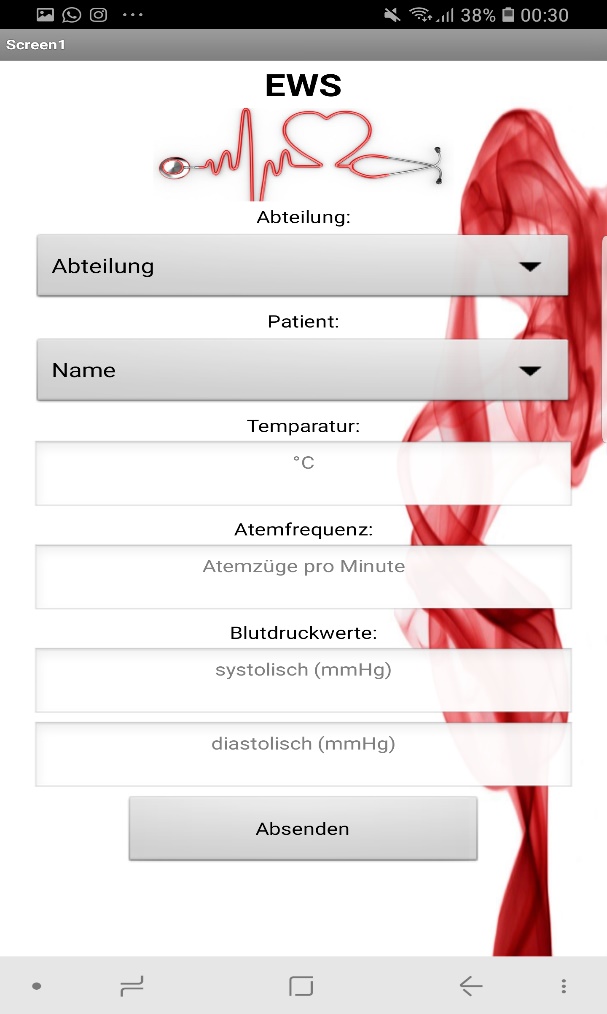
-Back-End Datenbank Anwendung **Verantwortlich**: Mike Durban

**Realisierung der Anforderungen:**

1. **Datenbank**
   1. Patientenverwaltung läuft über eine zentrale 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.   
          (D4, D5, D6, D7)
      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 Datenbandanfragen sowie die Benachrichtigungen   
          für das medizinische Personal bei kritischen Patienten. (D4, D5, D6, D7)
   2. Neue Patienten sollen über die Desktopapplikation hinzugefügt und entfernt werden.
      1. Die Desktopanwendung schickt die Daten an den Server. (D4, D5, D6, D7)
   3. Es muss pro Patient mindestens 3-mal am Tag gemessen werden können.
   4. Es müssen maximal 5 Stationen mit maximal 20 Personen verwaltet werden können.
2. **Server**
   1. Die initiale Verbindung mit der Datenbank ist innerhalb 5 Sekunden nach Start zu   
       bestehen.
   2. Anfragen an die Datenbank sind innerhalb von 1 Sekunde bearbeitet sein
   3. Die Kommunikation mit der App muss innerhalb von 5 Sekunden zu erfolgen
   4. Die Kommunikation mit der Oberfläche muss innerhalb von 5 Sekunden zu erfolgen
   5. Die Oberfläche kann verschiedene Anfragen über den Server an die Datenbank   
       senden. (Patientenkategorie rot, gelb, grün; Kritische Patienten)
3. **Android App**
   1. Die App kann einfach bedient werden. 9 von 10 Benutzer empfinden die Eingabe als  
       einfach.
      1. Übersichtliche Darstellung, (D1)
      2. Dropdown-Listen, (D1)
      3. Bei der Eingabe öffnet sich eine Tastatur, bei der nur Zahlen enthalten sind (D1)
      4. Bei den drei Werten können nur Zahlen eingegeben werden (D1)
      5. Beschreibung der einzugebende Werte (D1)
   2. Der aktuelle Patient kann in der App ausgewählt werden.
      1. 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
   3. Es können die folgenden drei Messwerte in der App eingetragen werden:  
       Temperatur, Atemfrequenz und Blutdruck
      1. Die Messwerte können untereinander, in der App eingegeben werden. (D1)
   4. Der Zeitpunkt der Messung muss ebenfalls erfasst werden.
      1. Beim Absenden wird die Zeit automatisch erfasst. Die Daten werden beim Absenden  
          an den Server versendet.
4. **Desktopanwendung**
   1. Es werden die Patienten mit ihrer Bewertung, eingeteilt in ihre Stationen auf der   
       Desktopanwendung angezeigt. (D2)
      1. Die Desktopanwendung-GUI soll mit Hilfe von Eclipse und entsprechenden Plug-Ins visualisiert werden. Die genutzte Programmiersprache soll Java sein.
   2. Patientenverwaltung wird über die Desktopanwendung gesteuert.
      1. Über einen Button soll die Verwaltung aufrufbar sein. Dort können Patienten hinzugefügt, beziehungsweise im Falle einer Entlassung deaktiviert und bereits bekannte Patienten wieder reaktiviert werden. (D2)
   3. Anhand von den Normalwerten werden die eingetragenen Messwerte in drei   
       Kategorien eingeteilt.  
       -rot (2 Werte in den kritischen Bereichen)  
       -gelb (1 Wert in den kritischen Bereichen)  
       -grün (kein Wert im kritischen Bereich)
      1. Durch die übermittelten Daten soll herausgefunden werden, in welchem Bereich der Patient liegt und entsprechend graphisch umgesetzt werden. (D2, D3)
   4. Es soll zwischen Kinder und Erwachsenen unterschieden werden.
      1. Es wird in der Anzeige über graphische Darstellung zwischen Kindern und Erwachsenen unterschieden.
   5. Die Grenzwerte des kritischen Bereichs sind für die Patienten individuell.
      1. 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. (D2, D3)
   6. Bei dem Nachtmodus sollen die Patienten mit roter und gelber Bewertung über alle Stationen hinweg angezeigt werden.  
      Bei einer gelben Bewertung soll die Oberschwester, bei einer roten Bewertung der behandelnde Arzt per Benachrichtigung informiert werden. Die Benachrichtigung muss innerhalb 30 Sekunden angelangt sein. Die Benachrichtigung enthält die relevanten Patientendaten und Messwerte.
      1. 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. (D2,D3)

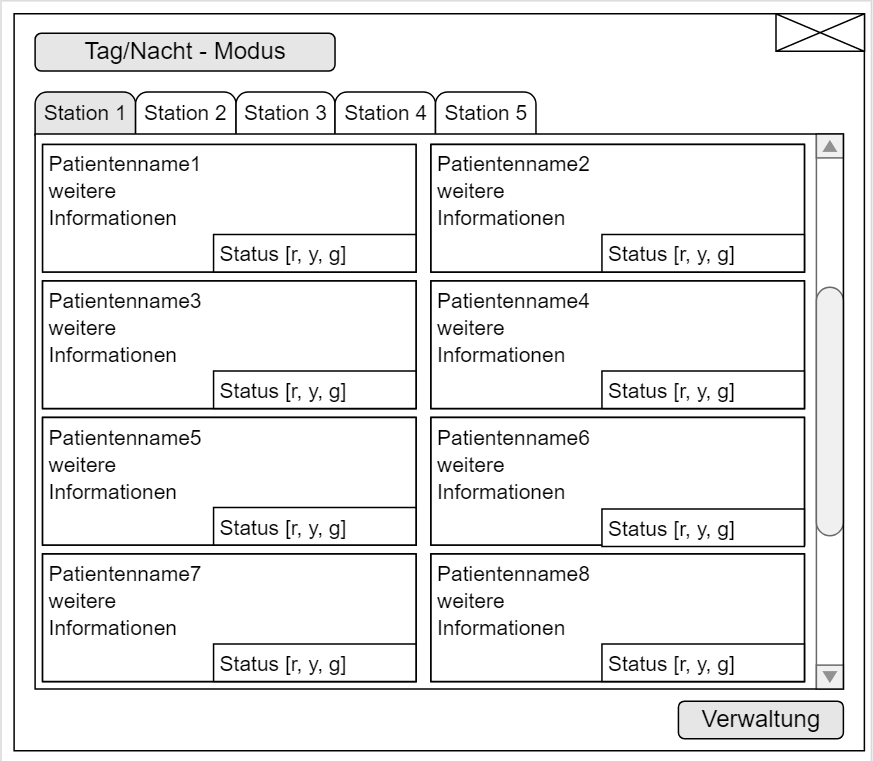
**Grobdesign (Grafiken)**

D1) Entwurf der GUI für die Android APP

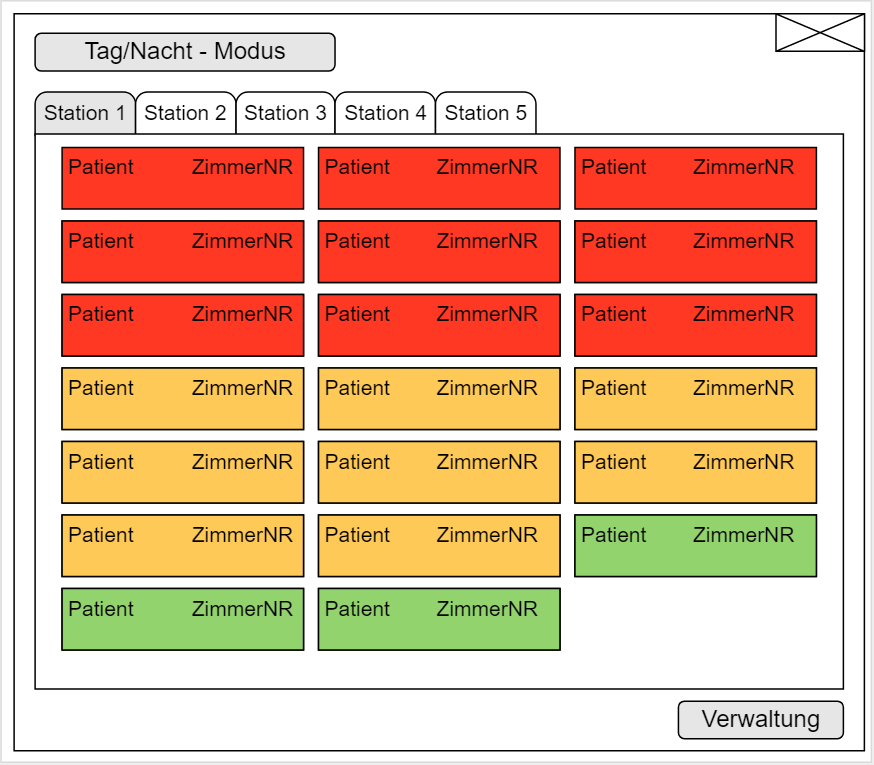


Entwurf der GUI für die Android App   
– Rukiye Akin

D2) Entwurf der GUI für die Desktopanwendung im Tag-Modus

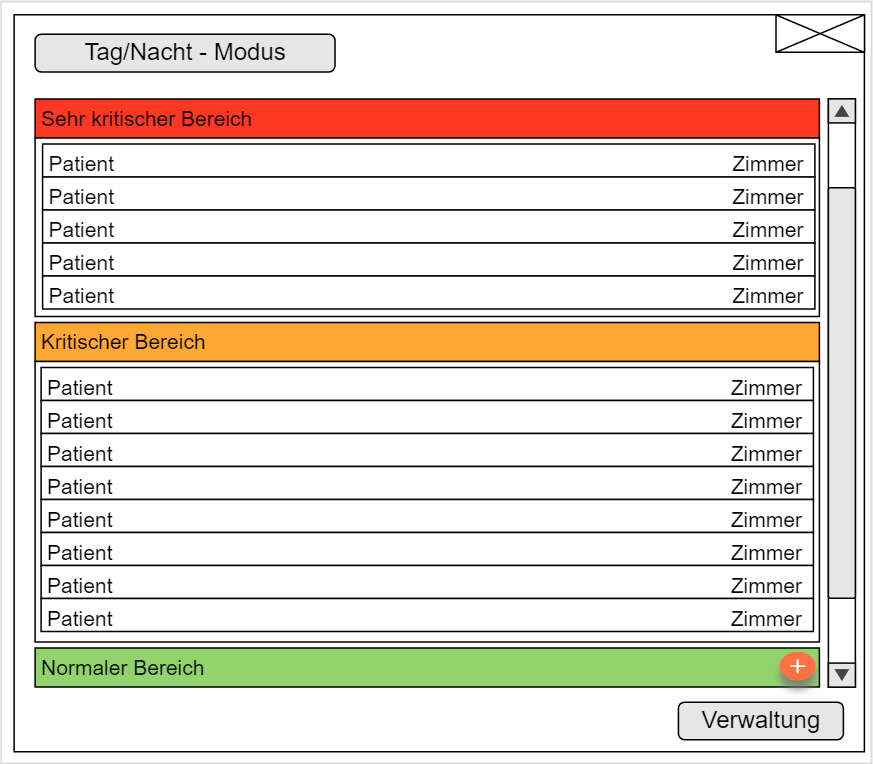


Entwurf 1 der GUI für die Desktopanwendung   
– Raphael Panter



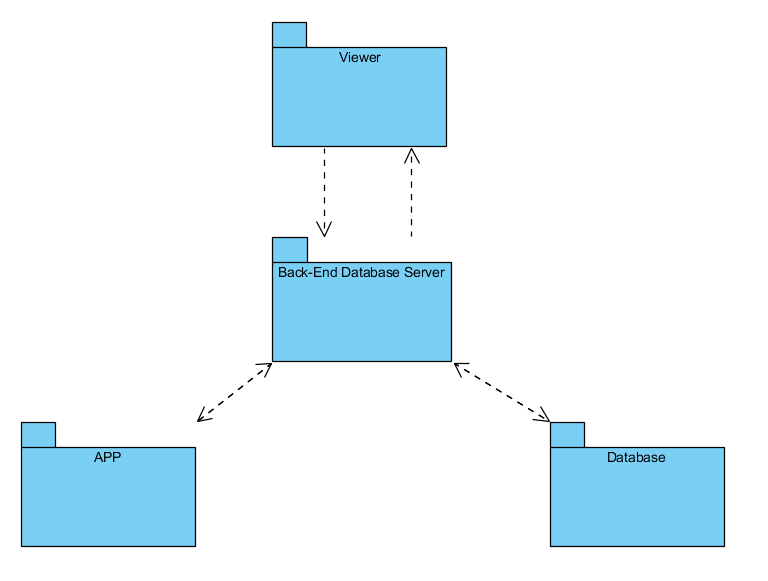
Entwurf 2 der GUI für die Desktopanwendung   
– Raphael Panter

D3) Entwurf der GUI für die Desktopanwendung im Nacht-Modus



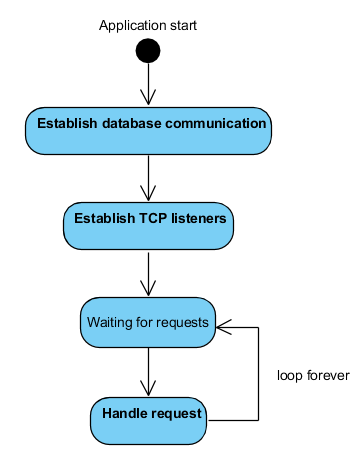
Entwurf der GUI für die Desktopanwendung im Nacht-Modus  
– Raphael Panter

D4) Packet Diagramm der größeren Module des Projekts

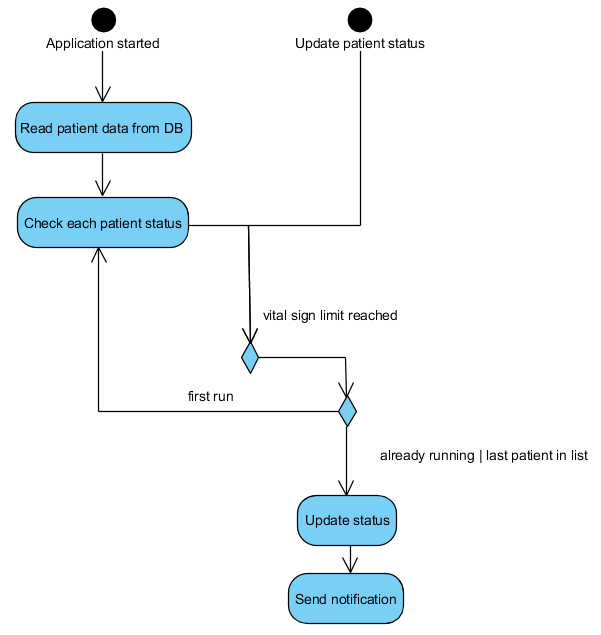


Packet Diagramm der einzelnen Module des Projekts  
– Daniel Ladwig

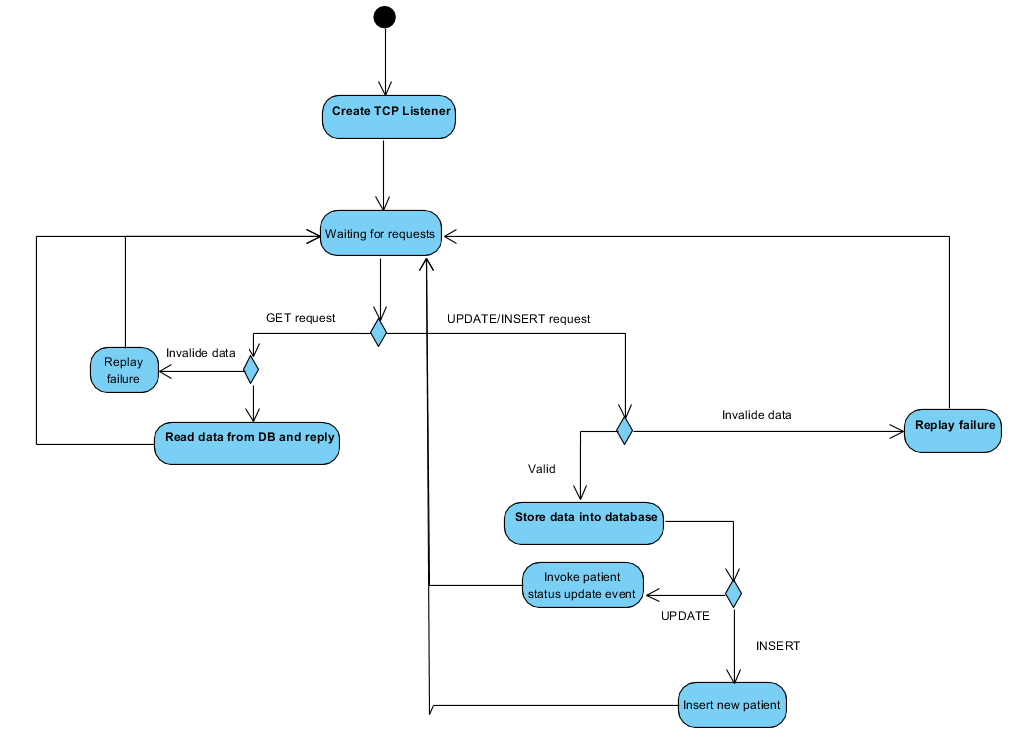
D5) Kurzübersicht des Datenbank-Servers



Quick overview des Servers  
– Mike Durban

D6) Update des Patientenstatus

Darstellung Update des Patientenstatus  
– Mike Durban

D7) Darstellung TCP, verwendet vom Datenbank Server

TCP  
– Mike Durban