**Use case diagram**

**Aktører**: Der er identificeret 5 aktører, hvilket er Administrator, Læge, Patient, Støttesystem og Database. Aktørerne Administrator, Læge og Patient anses for at være aktive aktører, da aktørerne initierer udførelsen af use cases. Støttesystem og Database anses som værende passive aktører, da disse aktører bidrager til funktionaliteten af use cases.

Aktørerne er inddelt i to use case diagrammer. Det ene diagram har aktørerne Administrator, Læge og Database, mens det andet diagram har aktørerne Patient, Støttesystem og Database.

Alle use cases, der kræver at der læses og skrives til databasen er derudover forbundet til databasen. Dette er ikke vist på use case diagrammet, da det er underforstået.

**Includes:** Der benyttes en include relation mellem to use cases, hvis den ene use case udføres som følge af den anden. Et eksempel på include relation er *base use casen* ”Indsend monitoreringsdata” og *included use casen* ”SCG-signalbehandling”. Det vil sige, når use casen ”Indsend monitoreringsdata” udføres skal use casen ”SCG-signalbehandling” også udføres.

**Extends:** Der benyttes en extend relation mellem to use cases, hvis den ene use case udføres som følge af den anden men ikke nødvendigvis at gøre det. Et eksempel på extend relation er *base use casen* ”MPI dataanalyse” og *extending use casen* “Alarmer læge”. Når ”MPI dataanalyse” udføres er det muligt men ikke obligatorisk, at use casen “Alarmer læge” udføres.

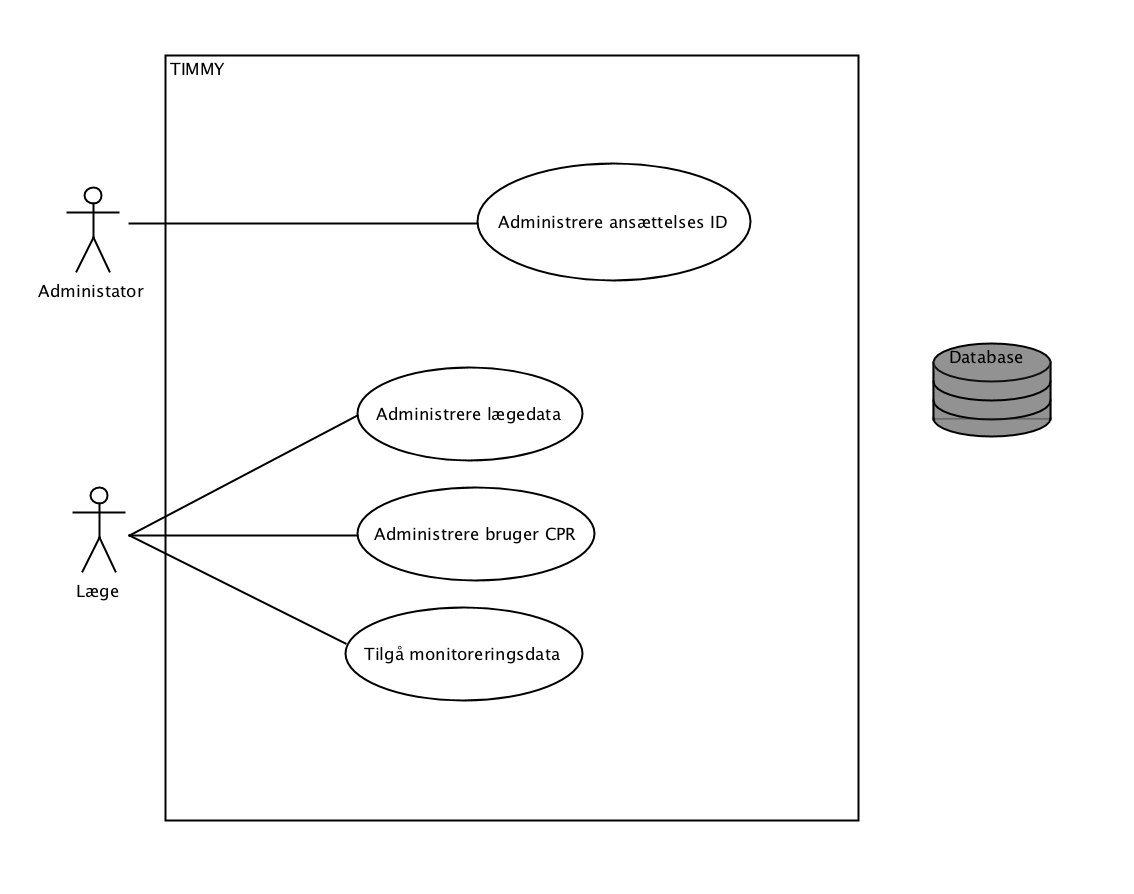
**Use cases:** Funktioner fra kravsspecifikationerne som eksempelvis at en patient kan indsende og redigere i patientdata er samlet i use casen kaldet ”Administrer patientdata”. Hvad disse præcis indebærer, beskrives i de enkelte beskrivelser af use cases.

**Støttesystem:**

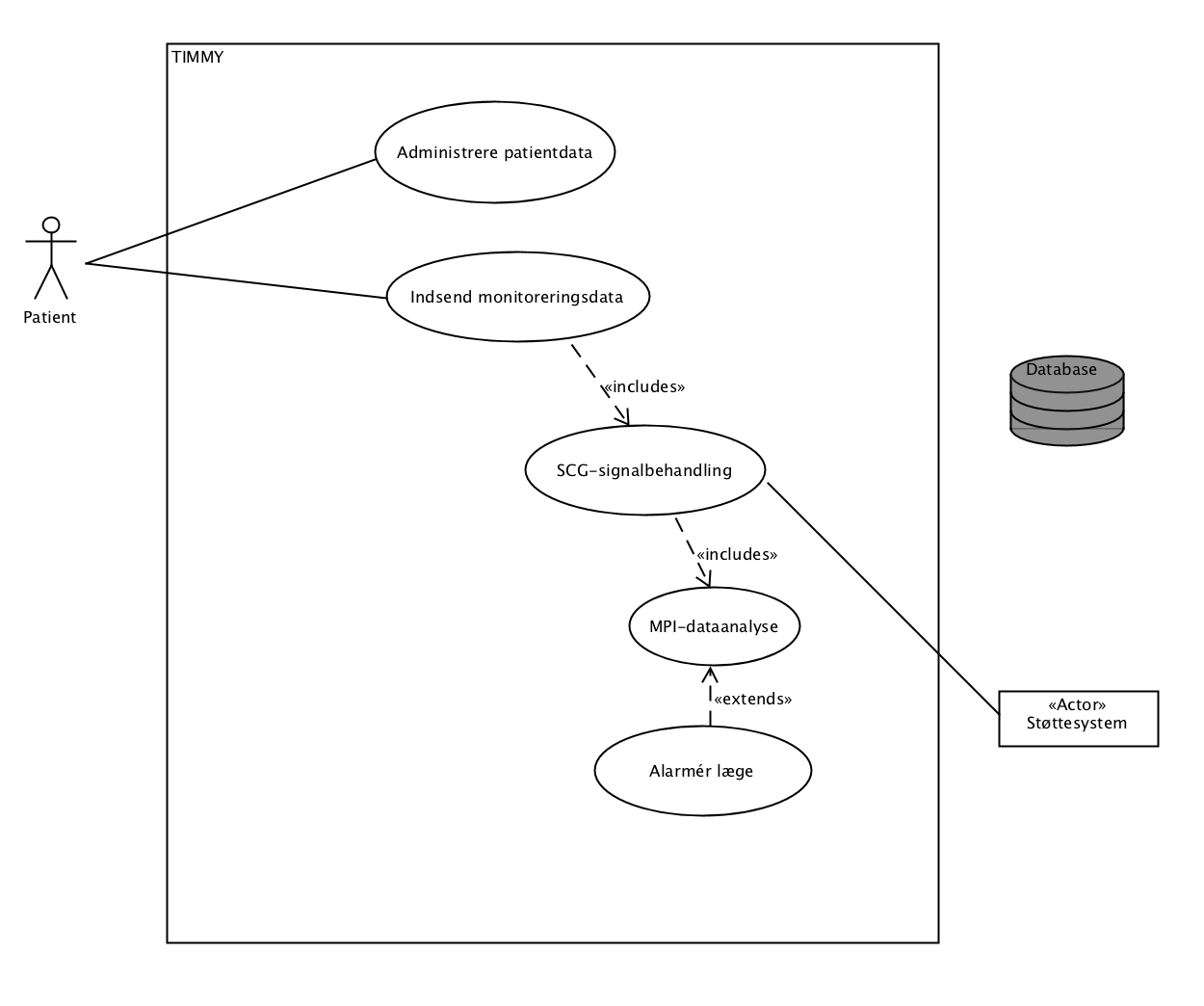
Denne aktør er oprettet, da der skal laves databehandling på det SCG-signal, der indsendes til databasen.

* **SCG-signalbehandling**. Når en patient indsender bl.a. et SCG-signal ved ”Indsend monitoreringsdata” skal støttesystemet læse denne data for at lave SCG-dataanalyse, ved at lokalisere fiducialmarkører for derefter at gemme et tilhørende MPI.
* **MPI dataanalyse.** Når et MPI skal vurderes om det ligger inden for tærskelværdien samt om der i dette tilfælde skal udføres *extending use casen* ”Alarmer læge”.

**Use case beskrivelser**

****

Ovenstående use case diagram viser aktørerne Administrator, Læge og Database.

****

Use case diagrammet viser aktørerne Patient, Støttesystem og Database.

|  |  |
| --- | --- |
| Name: | **Administrer ansættelses ID** |
| Short description: | Administrator tildeler og sletter ansættelses ID, som har adgang til at oprette brugere i systemet. |
| Precondition: | Login med administrator ID |
| Postcondition: | Ansættelses ID i databasen er opdateret |
| Error situations: | Forsøg på at slette ansættelses ID der ikke findes i databasen. |
| System state in the event of an error: | Ansættelses ID er ikke opdateret. |
| Actors: | Administrator. |
| Trigger: | Administrator ønsker at redigere ansættelses ID i databasen. |
| Standard process: | 1. Administrator logger ind. 2. Det vælges om der skal oprettes eller slettes ansættelses ID fra databasen. 3. Hvis ansættelses ID skal oprettes indskrives dette/disse i databasen.   Hvis ansættelses ID skal slettes, fjerens dette fra databasen.   1. Bekræftelse på udført kommando vises til administrator. |
| Alternative processes: |  |

|  |  |
| --- | --- |
| Name: | **Administrer lægedata** |
| Short description: | Læge kan oprette en bruger, tilgå og rette lægedata. Dette inkluderer navn, klinik, klinikkens adresse, ansættelses ID og klinikkens kontaktoplysninger. |
| Precondition: | Ansættelses ID findes i databasen. |
| Postcondition: | Lægedata er opdateret. |
| Error situations: | Forkert login.  Kan ikke oprette forbindelse til database.  Forsøg på at indtaste information af forkert dataformat. |
| System state in the event of an error: | Lægedata er ikke opdateret. |
| Actors: | Læge. |
| Trigger: | Læge logger på for at opdatere lægedata. |
| Standard process: | 1. Læge logger på. 2. Hvis det er førstegangslogin oprettes kode og læge data i form af navn, klinik, klinikkens adresse, ansættelses ID og klinikkens kontaktoplysninger indskrives. 3. Hvis det ikke er førstegangslogin indskrives kode. 4. Læge vælger at rediger lægedata. 5. Gammelt lægedata slettes hvorefter ny lægedata indskrives til databasen. 6. Bekræftelse på udført kommando vises til læge. |
| Alternative processes: |  |

|  |  |
| --- | --- |
| Name: | **Administrer bruger CPR** |
| Short description: | Bruger med ansættelses ID tildeler og sletter CPR-numre der giver brugere adgang til at oprette brugere. Yderligere kan CPR tilknyttes patientens funktionsklasse (NYHA). |
| Precondition: | Login med ansættelses ID |
| Postcondition: | CRP i databasen er opdateret |
| Error situations: | Forsøg på at slette CRP der ikke findes i databasen. |
| System state in the event of an error: | CPR er ikke opdateret. |
| Actors: | Læge med ansættelses ID. |
| Trigger: | Læge ønsker at redigere CPR i brugerregisteret. |
| Standard process: | 1. Læge med ansættelses ID logger ind. 2. Det vælges om funktionsklasse skal opdateres eller om der skal oprettes eller slettes CPR fra databasen. 3. Hvis funktionsklasse skal opdateres fjernes den gamle funktionsklasse og den nye indskrives i databasen. Hvis CPR skal oprettes indskrives dette/disse i databasen.   Hvis CPR skal slettes, fjernes dette fra databasen.   1. Bekræftelse på udført kommando vises til administrator. |
| Alternative processes: |  |

|  |  |
| --- | --- |
| Name: | **Tilgå monitoreringsdata** |
| Short description: | Læge skal være i stand til at se monitoreringsdata indsendt af patienter tilknyttet denne læge. Dette inkluderer SCG-signal og symptomer. |
| Precondition: | Login med ansættelses ID. |
| Postcondition: | Læge har monitoreringsdata tilgængelig. |
| Error situations: | Kan ikke oprette forbindelse til database. |
| System state in the event of an error: | Monitoreringsdata er ikke tilgængelig. |
| Actors: | Læge. |
| Trigger: | Læge logger på for at læse monitoreringsdata. |
| Standard process: | 1. Læge logger på. 2. CPR på den ønskede patient indtastes. 3. Den ønskede data vælges. 4. Data læses fra databasen og vises grafisk. |
| Alternative processes: |  |

|  |  |
| --- | --- |
| Name: | **Administrer patientdata** |
| Short description: | Patient kan oprette en bruger, tilgå og rette patientdata. Dette inkluderer navn, adresse og kontaktoplysninger. |
| Precondition: | Login med CPR |
| Postcondition: | Patientdata i databasen er opdateret |
| Error situations: | Forkert login.  Kan ikke oprette forbindelse til database.  Forsøg på at indtaste information af forkert dataformat. |
| System state in the event of an error: | Patientdata er ikke opdateret. |
| Actors: | Bruger med bruger login. |
| Trigger: | Bruger logger på for at opdatere lægedata. |
| Standard process: | 1. Bruger logger på. 2. Hvis det er førstegangslogin oprettes kode og patient data i form af navn, adresse og kontaktoplysninger. 3. Hvis det ikke er førstegangslogin indskrives kode. 4. Bruger vælger at rediger patientdata. 5. Gammelt patientdata slettes hvorefter ny patientdata indskrives til databasen. 6. Bekræftelse på udført kommando vises til bruger. |
| Alternative processes: |  |

|  |  |
| --- | --- |
| Name: | **Indsend monitoreringsdata** |
| Short description: | Daglig indskrivning af monitoreringsdata til databasen. Monitoreringsdata inkluderer SCG-signal og symptomer. |
| Precondition: | At symptomer er indtastet og ”Opsamling af SCG-signal er udført”. |
| Postcondition: | Symptomer og SCG-signal er indskrevet i databasen med tidsstempel. |
| Error situations: | Der kan ikke oprettes forbindelse så data kan ikke indskrives i databasen. |
| System state in the event of an error: | Intet monitoreringsdata indskrevet i databasen. |
| Actors: | Patient |
| Trigger: | Patienten skal dagligt udføre monitorering |
| Standard process: | 1. SCG-signal er tilgængeligt 2. Symptomer indtastes. 3. Data indskrives på databasen. |
| Alternative processes: |  |

|  |  |
| --- | --- |
| Name: | **Opsamling af SCG-signal.** |
| Short description: | Patientens daglig opsamling af SCG-signal gennem patientapplikationen ved brug af accelerometeret i smartphonen. |
| Precondition: | Patienten ligger på ryggen, påbegynder optagelse og placere smartphonen mellem midten af sternum og xiphoid proces. |
| Postcondition: | SCG-signal er optaget og midlertidigt tilgængeligt i patientapplikationen. |
| Error situations: | Forkert signal optages på grund af forkert placering.  Signalet er invers på grund af forkert placering.  Intet signal optages da patienten ikke kan udføre måling. |
| System state in the event of an error: | Forkert signal indskrives til databasen. Invers signal indskrives til databasen.  Intet SCG-signal målt. |
| Actors: | Patient |
| Trigger: | Patienten skal dagligt foretage opsamling af SCG-signal for at ”Indskrive monitoreringsdata”. |
| Standard process: | 1. Patienten lægger sig på en plan fast overflade. 2. Patient påbegynder nedtællingen på applikationen. 3. Smartphone placeres mellem midten af sternum og xiphoid proces. 4. Nedtælling afsluttes. 5. Opsamling i 10 sekunder. 6. Opsamling afsluttes. 7. Patienten gøres opmærksom på at opsamlingen er afsluttet. 8. Patienten slapper af. |
| Alternative processes: |  |

|  |  |
| --- | --- |
| Name: | **SCG-signalbehandling** |
| Short description: | Lokaliserer fiducialmarkører for derefter at udregne og gemme et tilhørende MPI. |
| Precondition: | Der er et nyt SCG-signal tilgængeligt og læsbart i databasen. |
| Postcondition: | MPI gemmes i databasen |
| Error situations: | SCG-signalet kan analyses at SCG-algoritmen. |
| System state in the event of an error: | Intet MPI registreres. Et fejlagtigt MPI registreres. |
| Actors: | Støttesystem |
| Trigger: | Patient indsender SCG-signal ved use case ”Indsend monitoreringsdata”. |
| Standard process: | 1. SCG-signal registreres 2. SCG-signal læses fra databasen. 3. Fiducialmarkører registreres ved algoritme. 4. Tidsintervaller til MPI beregnes. 5. MPI beregnes. 6. MPI indskrives til databasen. |
| Alternative processes: |  |

|  |  |
| --- | --- |
| Name: | **MPI dataanalyse** |
| Short description: | Læse MPI for derefter at vurdere om use casen ”Alarmer læge” skal udføres. |
| Precondition: | Der er et nyt MPI tilgængeligt og læsbart i databasen. |
| Postcondition: | Der tages beslutning om ”Alarmer læge” skal udføres. |
| Error situations: | Intet nyt MPI er tilgængeligt |
| System state in the event of an error: | Læge alarmeres ikke. |
| Actors: | Støttesystem |
| Trigger: | Use case ”SCG-signalbehandling” er udført” |
| Standard process: | 1. MPI registreres 2. MPI læses fra databasen. 3. MPI sammenholdes med tærskelværdi. 4. Hvis tærskelværdien overskrides alarmeres læge. 5. Hvis tærskelværdien ikke overskrides alarmeres læge ikke. |
| Alternative processes: |  |