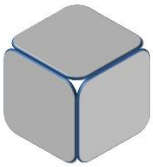
UML Model Report

A monitoring system for sleep quality

Group number: *8*

Team members:	**Student ID:**
Fosco Cancelliere	*846395*
Manuel Carzaniga	*920239*
Lorenzo Gualniera	*920820*
Francesco Sheiban	*920054*

POLITECNICO
MILANO 1863



B³ LAB
Biosignals
Bioimaging
Bioinformatics

Table of Contents

USE CASE DIAGRAM: TEXTUAL DESCRIPTION **3**

USE CASE MODEL **4**

Overview *4*

Primary actor – User *5*
Technical administrator (Tech admin) *5*
Patient *6*
Specialized practitioner (Medician) *6*

Secondary actor - Software *7*
LOG_IN_CHECK *7*
SHOW_ PROFILE *8*
WINDOW_MANAGER *8*
ADD_USER *8*
EDIT-DELETE_USER *9*
INSERT_PARAMETERS *9*
FILL_IN_SURVEY *9*
VISIT_BOOKING *10*
EDIT_VISIT *10*
ADD_VISIT_DATA *10*
SET_THRESHOLDS *11*

Secondary actor - Devices *12*

CLASS DIAGRAM **13**

ACTIVITY DIAGRAMS **15**

Technical administrator (Tech admin) *15*
Patient *18*
Specialized practitioner (Medician) *20*

ER DIAGRAM **22**

Use Case Diagram: Textual description

MAIN SUCCESS SCENARIO (MSS):

1. The user logs in
2. The user views the main screen
3. The user selects the desired function (read/write/other)
4. The user interacts with the function selected and save the changes
5. The user logs out

ALTERNATIVE SCENARIOS:

1a. The user fails to log in for wrong credentials
   1. The software notifies the user
   2. The software asks for credentials again
   3. Return to step 2 (MSS)

1b. The user fails to log in because he's not a registered user
   1. The software notifies the user
   2. The software asks to send a notification to the technical admins to take action and add the user
   3. Return to step 1 (MSS)

3a. The user selects the wrong function
   1. The software shows the back button
   2. The user goes back to the previous page
   3. Return to step 3 (MSS)

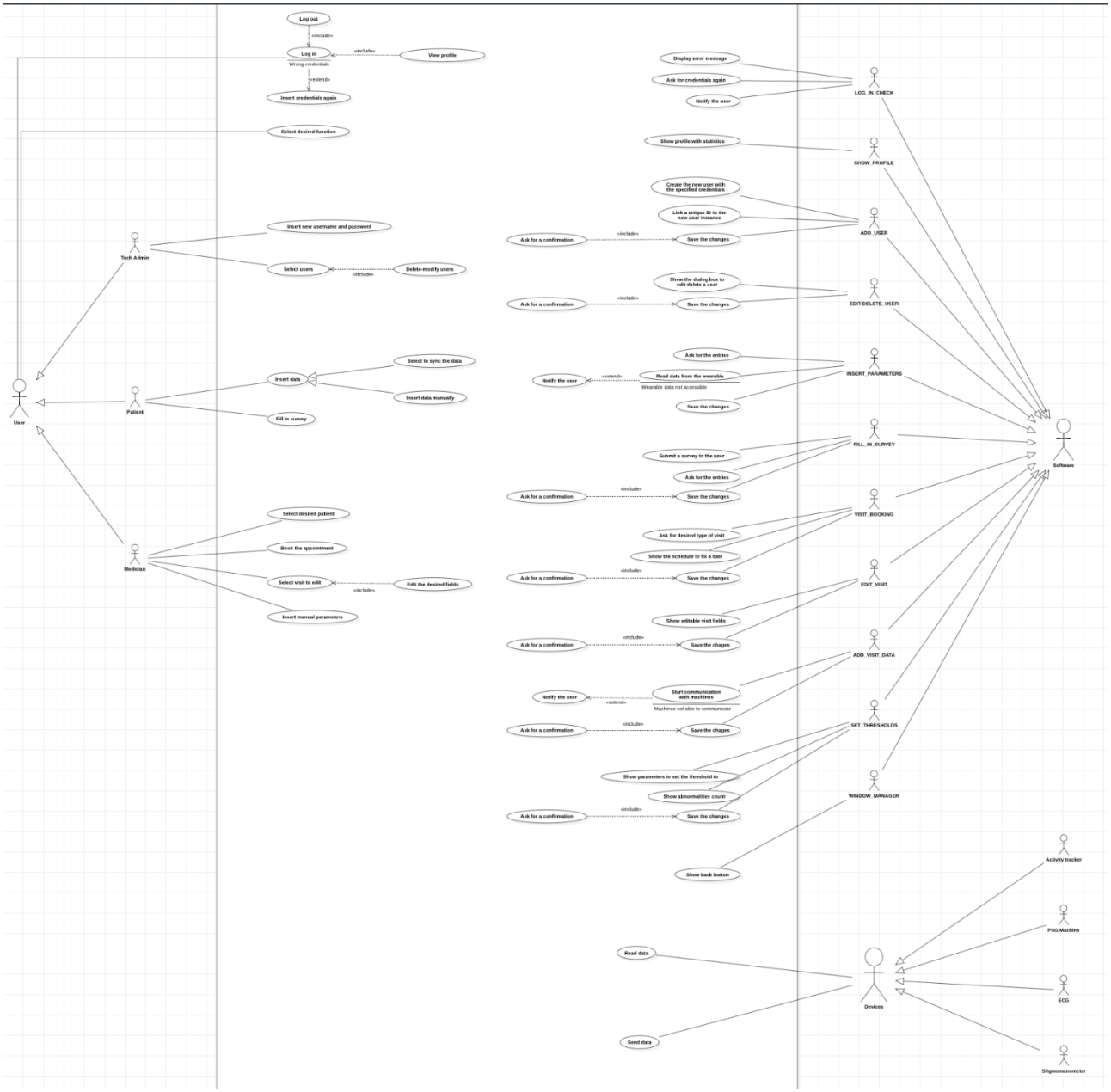
4a. The user saves the wrong changes
   1. The system allows to modify the record
   2. The user edits its interaction
   3. Return to step 5 (MSS)

Use case model – Overview

The following diagrams illustrates how actors relate to use cases.
In this implementation of a monitoring system for sleep quality we identify three main roles:

1) *User*
2) *Software*
3) *Devices*

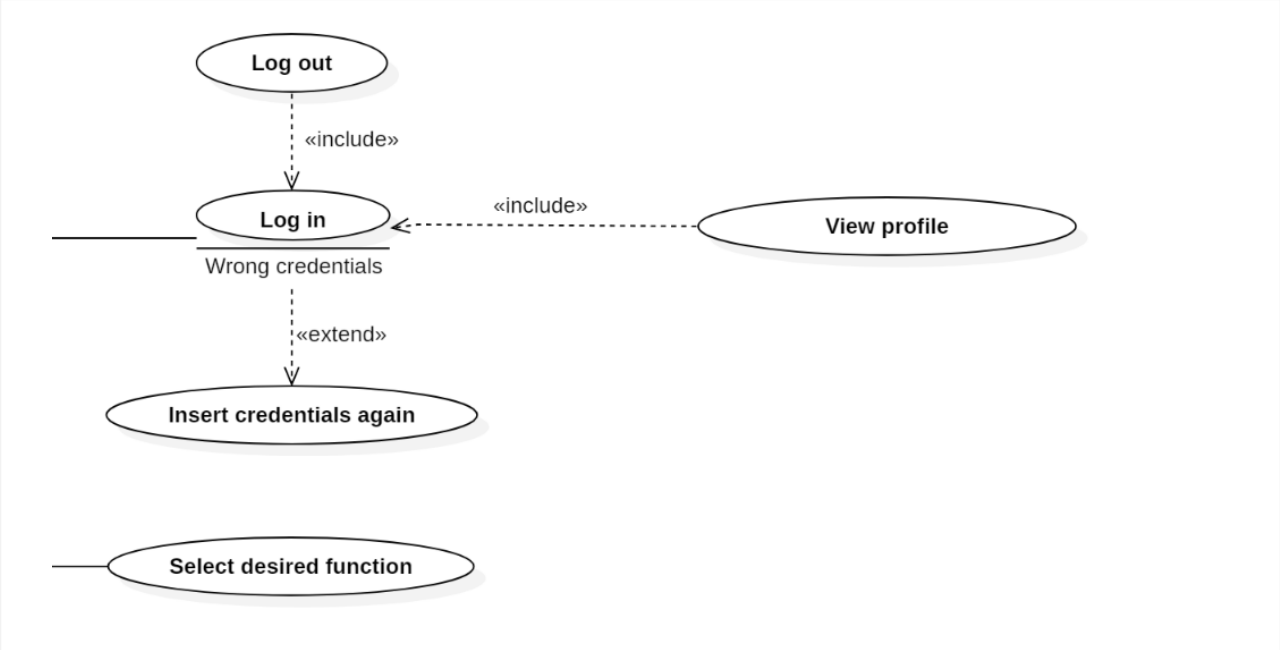
The main representation of the system in its entirety is reported below, while in the following paragraph the model will be dissected and discussed in detail.



Primary actor – User

As previously mentioned in the textual description, the primary actor of the system is represented by the *User*.

The *User* is able to carry out basic functions such as *Log in* and *Log out* in/from his personal account, which contains personal information and the access to specific functions belonging to the type of *User*.

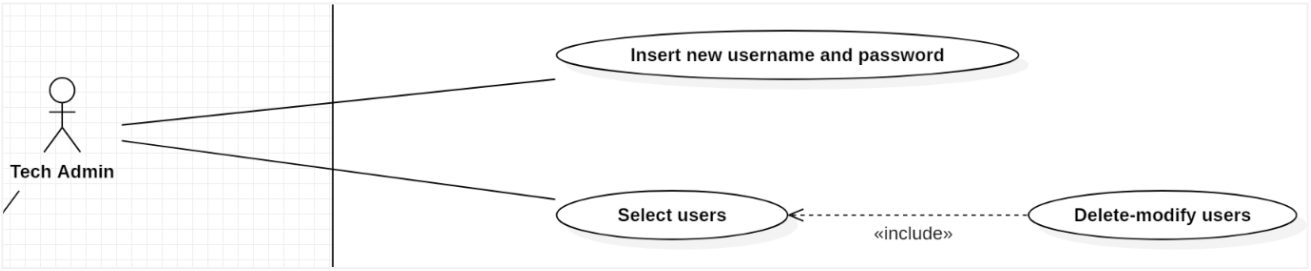


The *User* actor generalizes three different types of user:

1) *Technical administrator*
2) *Patient*
3) *Specialized practitioner*

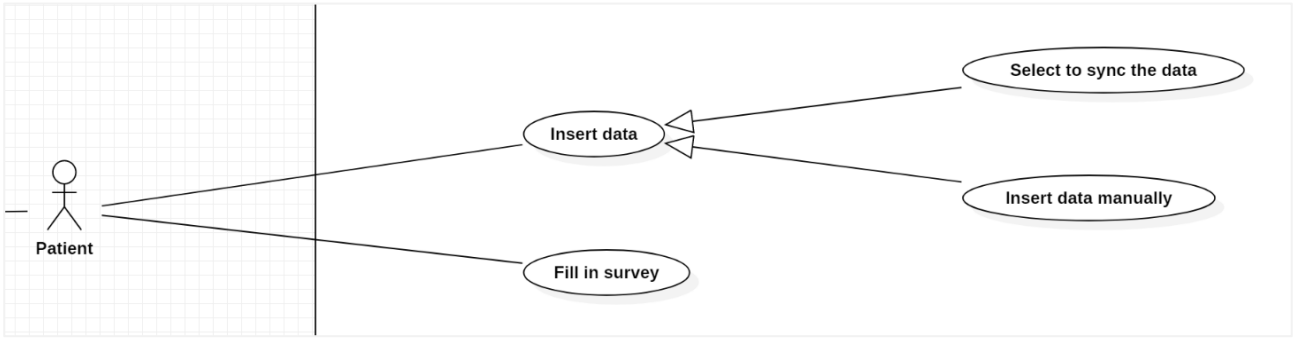
➢ Technical administrator (Tech Admin):

- Register new users
- Delete/modify existing users



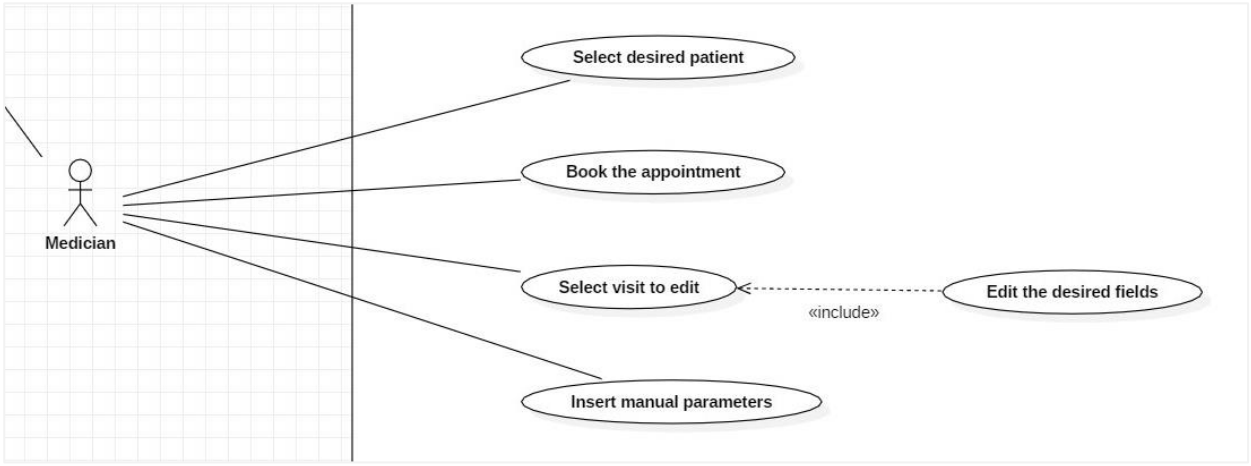
➢ Patient:

- insert parameters collected at home (sync data automatically or/and manually)
- fill in daily surveys



➢ Specialized practitioner (Medician):

- select the desired patient from a list
- book an appointment (either a visit or a video monitoring session)
- edit visits (change desired fields)
- insert parameters registered in hospital for a given patient (data collected or/and thresholds to identify abnormalities)



Secondary actor – Software

The means by which a *User* is able to perform his task is the *Software*, that represents a secondary actor in the system.

The *Software* is in turn composed by several specialized functions for a better management of the entire system, in such a way that different types of *User* are granted access to different functions.

For all types of *User*:

- LOG_IN_CHECK
- SHOW_PROFILE
- WINDOW_MANAGER

Technical administrator (Tech Admin):

- ADD_USER
- EDIT-DELETE_USER

Patient:

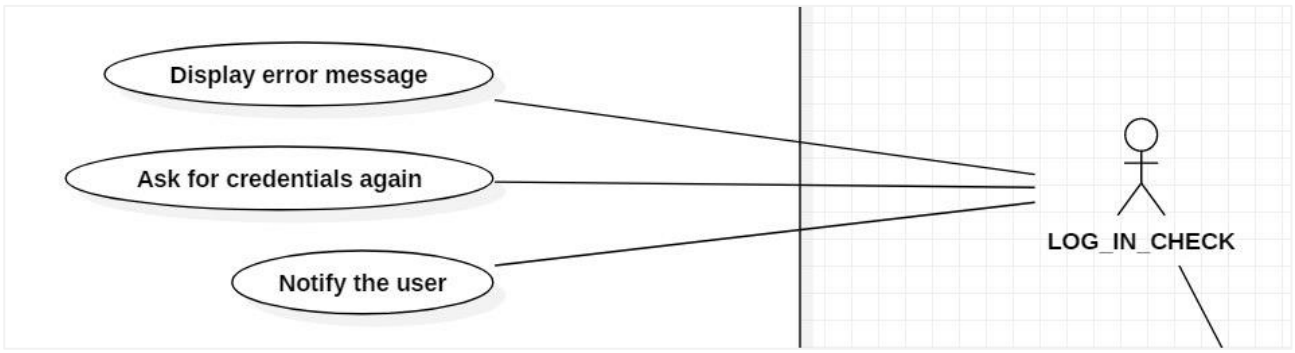
- INSERT_PARAMETERS
- FILL_IN_SURVEY

Specialized practitioner (Medician):

- VISIT_BOOKING
- EDIT_VISIT
- ADD_VISIT_DATA
- SET_THRESHOLDS

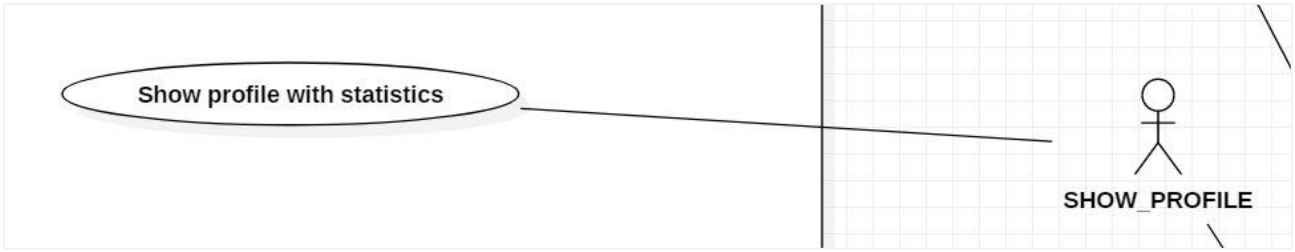
➢ LOG_IN_CHECK:

- display an error message if the inserted credentials are wrong
- after the error message the user is asked to insert the credentials once again
- after several failed attempts notify the user to contact the *Tech Admin*



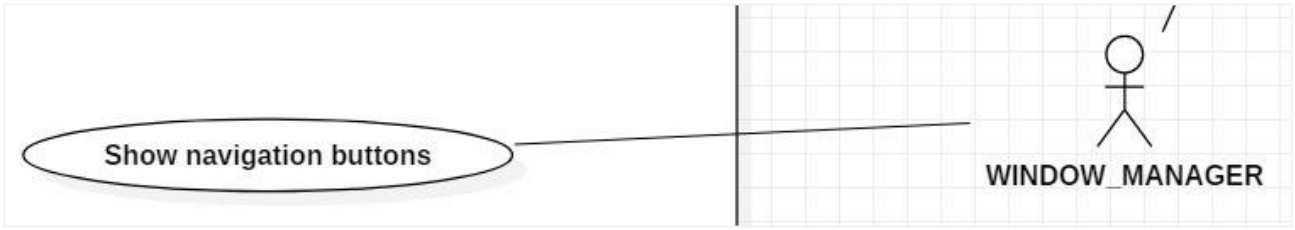
➢ SHOW_PROFILE:

- show user profile with related statistics (different types of user may have different statistics shown regarding their scope of interest)



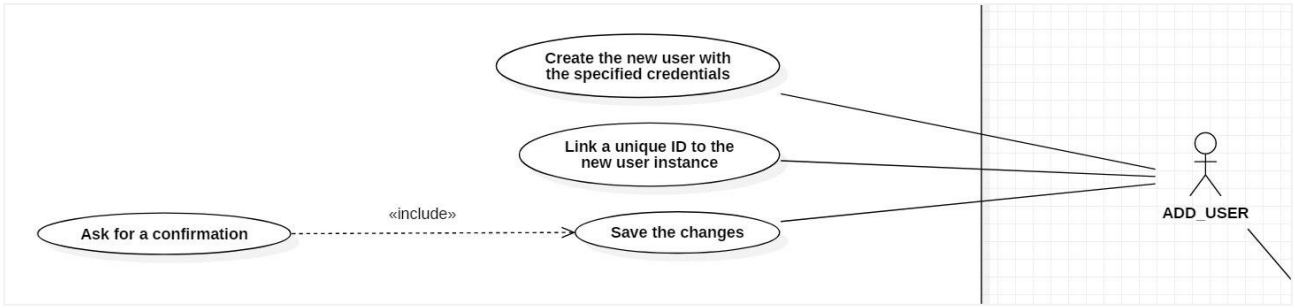
➢ WINDOW_MANAGER:

- enable the user to navigate through different pages by clicking on interactive buttons



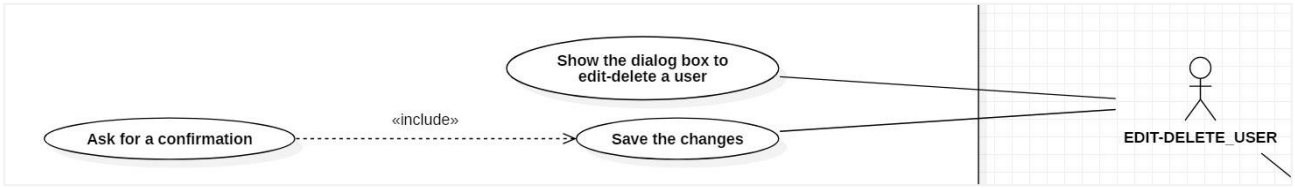
➢ ADD_USER:

- create a new user account with a username (generated from the user's personal information) and a password (generated randomly in order to meet security standards)
- link a unique ID to the new user instance to ease research in the database
- ask for confirmation and then save the changes after user's input



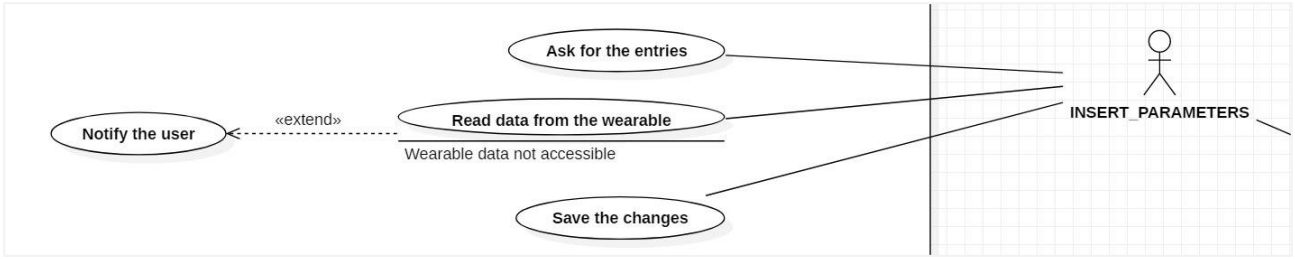
➢ EDIT-DELETE_USER:

- show the dialog box to edit-delete a user
- ask for confirmation and then save the changes after user's input



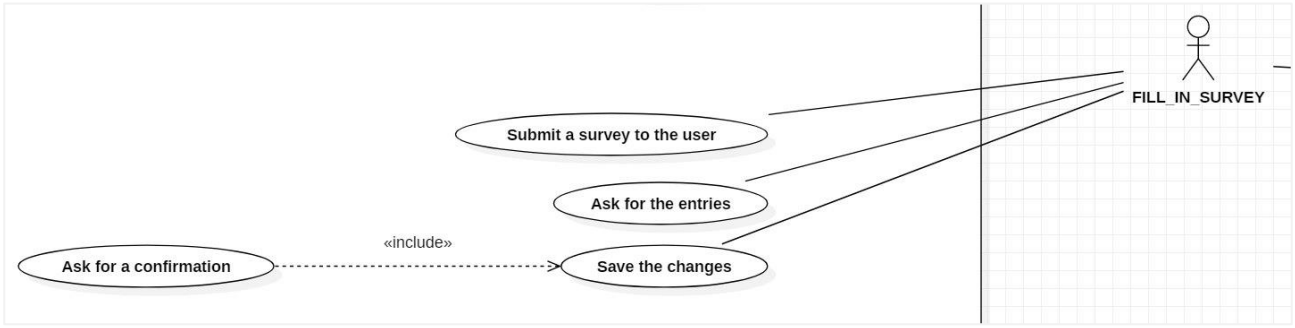
➢ INSERT_PARAMETERS:

- ask for entries
- read data from the wearable and notify the user of the connection
- save the changes



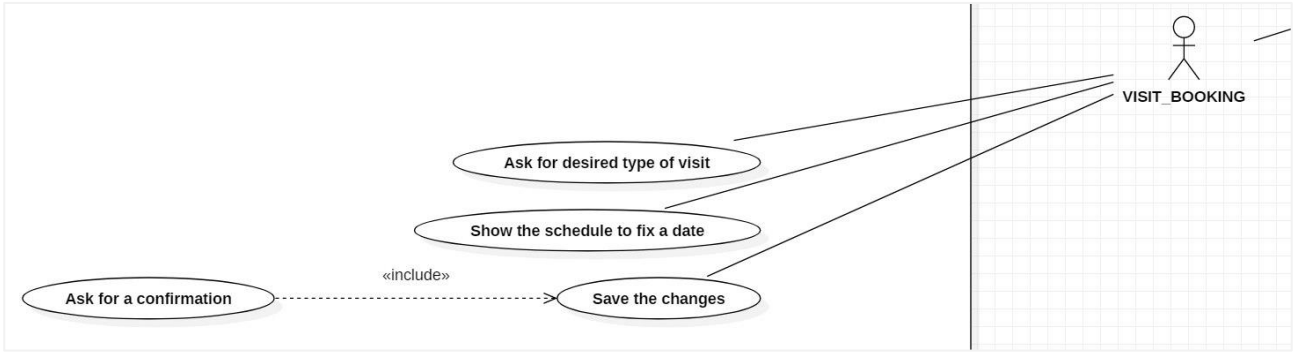
➢ FILL_IN_SURVEY:

- submit a survey to the user
- ask for the entries
- save the changes and ask for a confirmation



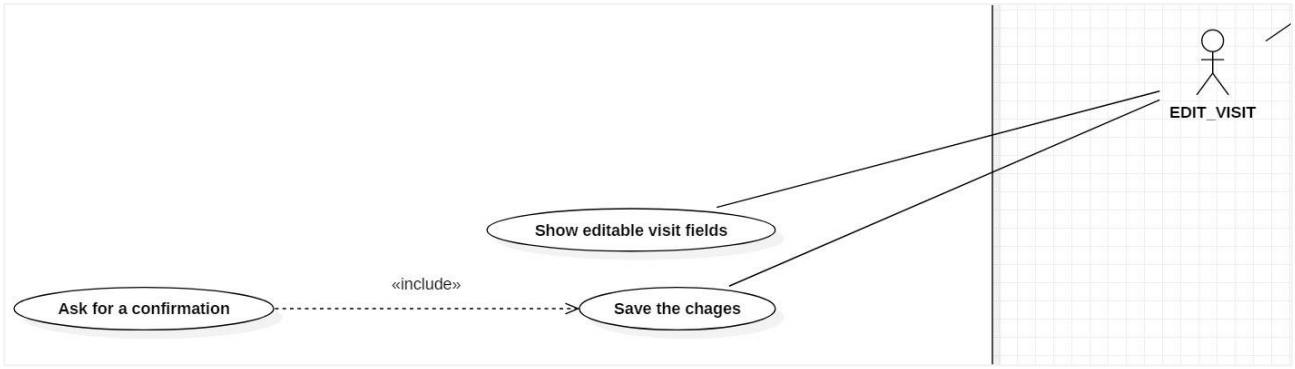
➢ VISIT_BOOKING:

- ask for desired type of visit to book
- show the schedule to a fix date
- save the changes and ask for a confirmation



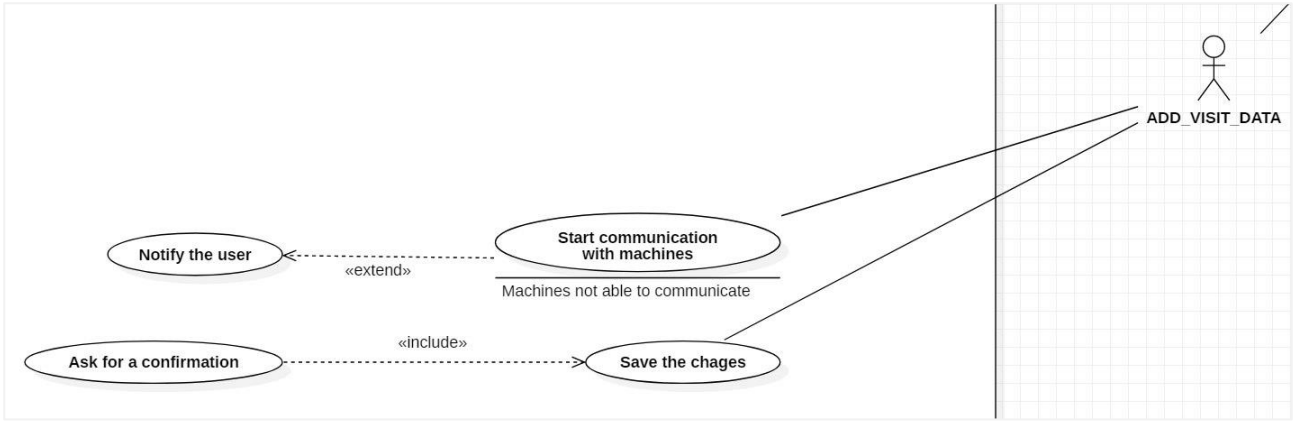
➢ EDIT_VISIT:

- show a list of editable fields for the visit (date, type)
- save the changes and ask for a confirmation



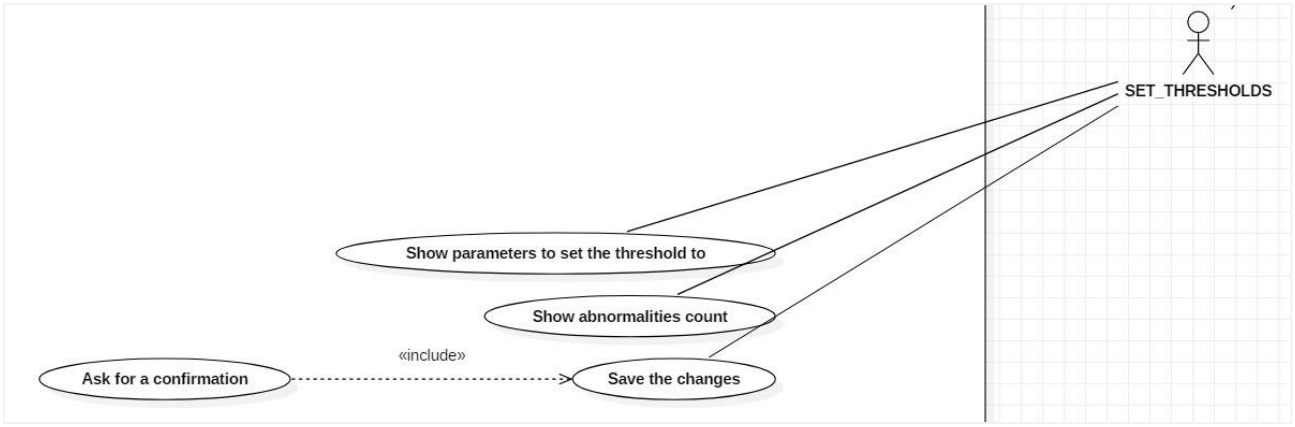
➢ ADD_VISIT_DATA:

- start communication with machines and notify the user
- save the update and ask for confirmation



➢ SET_THRESHOLD:

- show a list of parameters to set the threshold to
- show the number of abnormalities detected by the system through a counter
- save the changed threshold and ask for a confirmation



Secondary actor – Devices

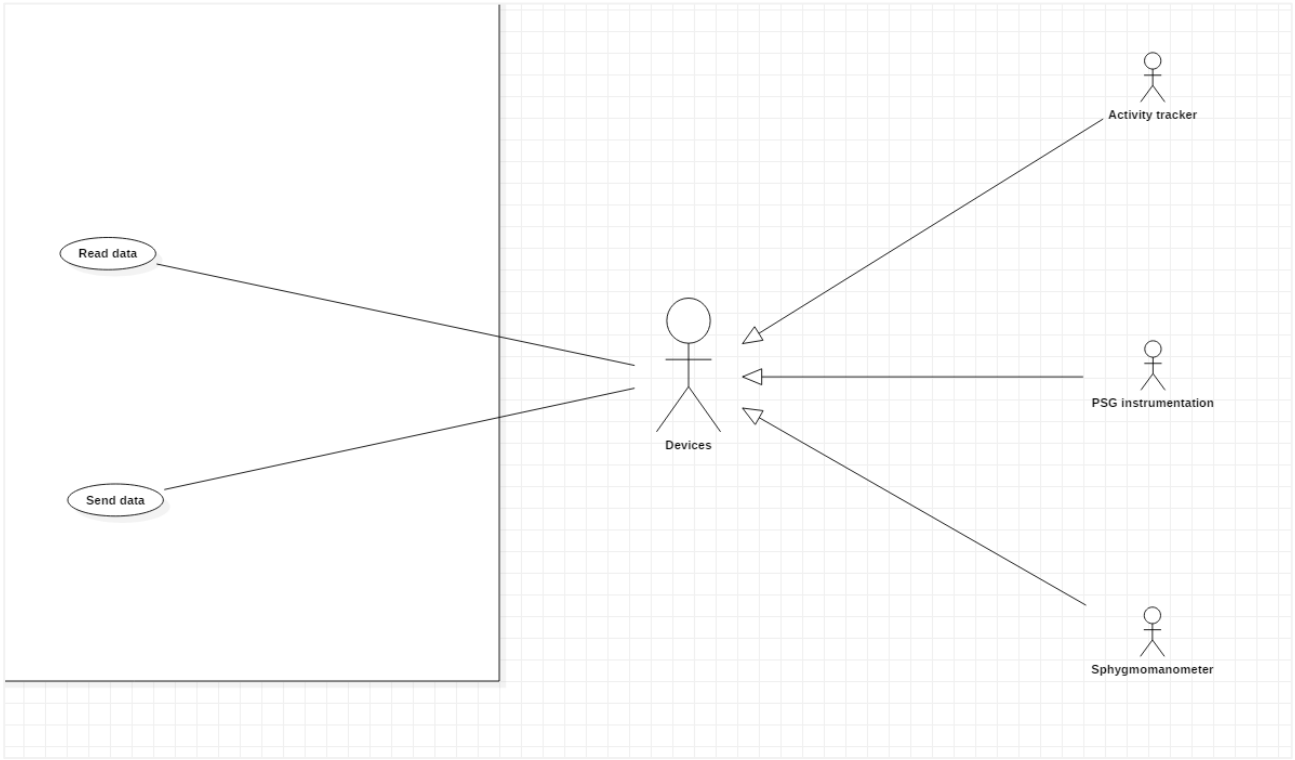
The last secondary actor of the system is represented by *Devices*, which groups three different types of instrumentation used by the *Specialized practitioner* during a visit or a sleep monitoring session.

Devices includes:

- Activity tracker
- PSG instrumentation
- Sphygmomanometer

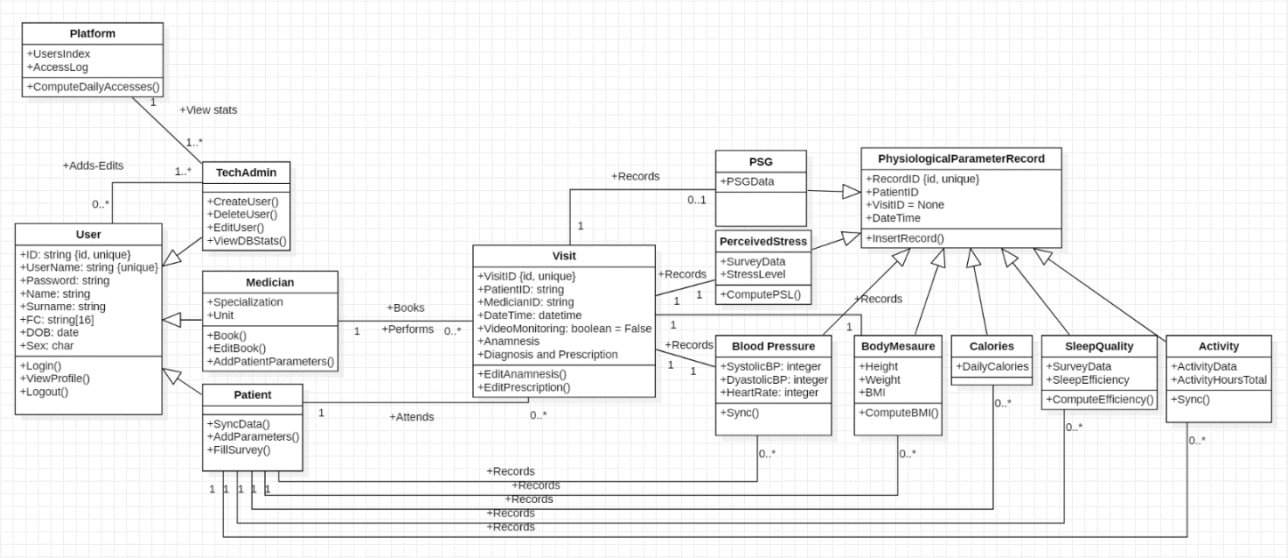
Devices utilizes two main functions of the system, which consists of the following:

- read physiological data from the medical instrument
- send collected data to the *Software* application



Class Diagram

The following diagram includes all the system classes, with their respective attributes, functions and relations.



➢ Notes:

The first design decision taken regarding the class diagram has been the generalization of the different types of people using the system: *TechAdmin*, *Medician* and *Patient*. We grouped the common attributes and functions under a parent class named *User*. In the *User* class we can find attributes regarding the credentials and personal information, such as Name, Surname, Sex and Date of Birth. The functions included in the class are basic methods to manage the access to the personal account.

A *Visit* class is able to organize and group *Patient* and *Medician's* information regarding a booked visit. Each *Visit* is linked to a unique *VisitID* in order to store the medical history of the patients.

The other central parent class worth mentioning is *PhysiologicalParameterRecord*.
This class includes both clinical parameters and home measurements in order to avoid redundancy using a *VisitID* attributes that can be set to None to discern between the two types of recording. An *InsertRecord()* function is then implemented to upload the collected data to the system from various sources (manually inserted or automatically synchronized).

Activity Diagrams

➢ Tech admin textual description & activity diagram

MAIN SUCCESS SCENARIO TECH ADMIN (MSS-T):

1. The tech admin logs in
2. The SHOW_PROFILE shows the tech admin its profile w./ the temporal statistic for active users and number of log-ins
3. The tech admin selects the desired function:

    3-1. Function selected: ADD_USER
        1. The ADD_USER asks for the username/password

    3-2. Function selected: EDIT/DELETE_USER
        1. The EDIT/DELETE_USER asks for the users to modify/delete

4. The tech admin interacts with the function selected:

    4-1. The tech admin inserts the new username and password
        1. The ADD_USER creates the new user with the specified credentials
        2. The ADD_USER links a unique ID to the user instance
        3. The ADD_USER saves the changes

    4-2. The tech admin selects the users to modify/delete
        1. The EDIT/DELETE_USER shows the dialog box to edit/delete a user
        2. The EDIT/DELETE_USER saves the changes

5. The tech admin logs out

ALTERNATIVE SCENARIOS:

1a. The tech admin fails to log in for wrong credentials
    1. The LOG_IN_CHECK notifies the user
    2. The LOG_IN_CHECK asks for credentials again
    3. The tech admin inserts the credentials again
    4. Return to step 2 (MSS-T)

1b. The tech admin fails to log in because he's not a registered user
    1. The LOG_IN_CHECK shows an error message
    2. Return to step 1 (MSS-T)

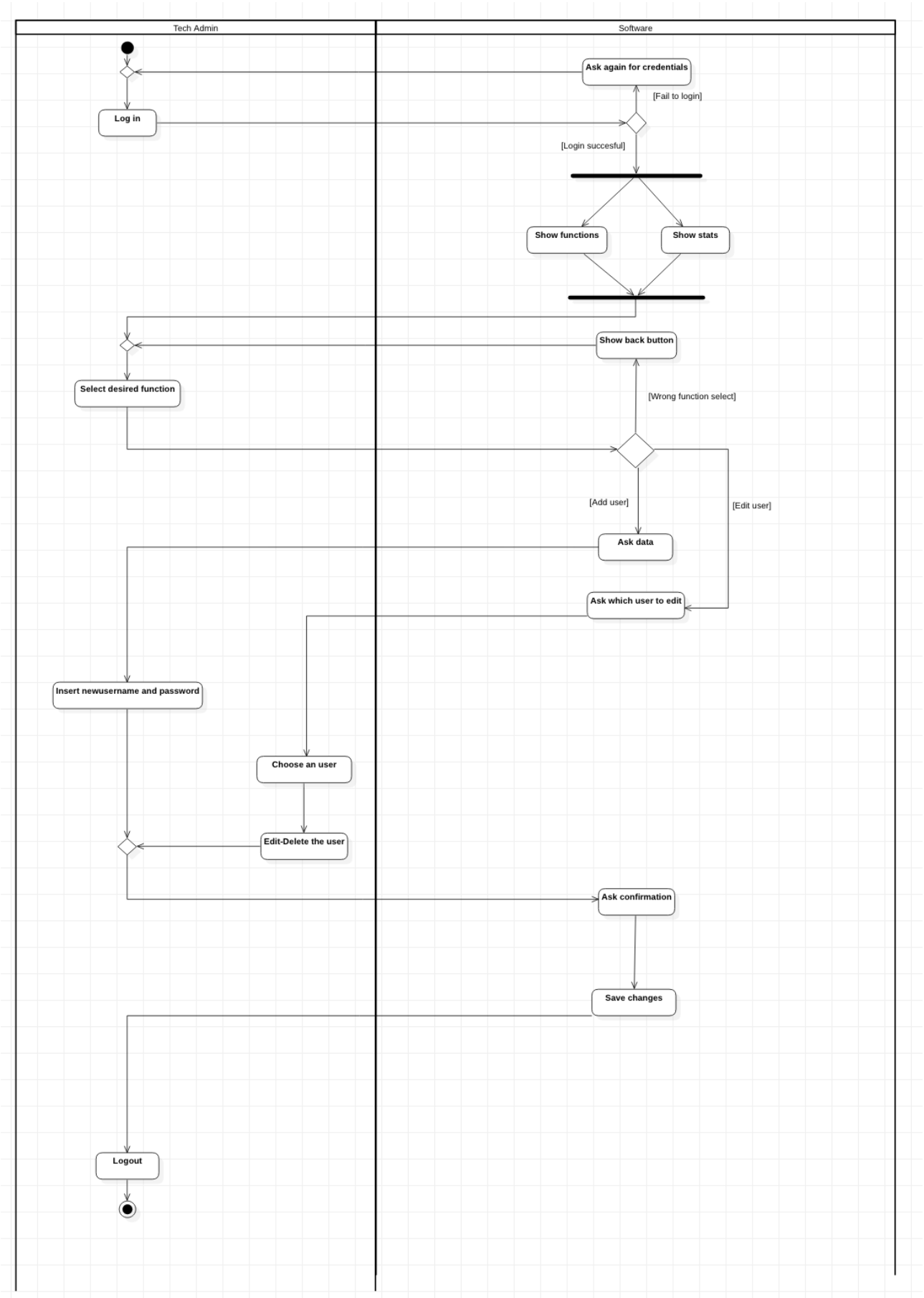
3a. The tech admin selects the wrong function
    1. The WINDOW_MANAGER shows a back button
    2. Return to step 3 (MSS-T)

4-1.a. The tech admin fails to add the proper user

1. Before saving the ADD_USER asks for a confirmation
2. Return to step 2 (MSS-T)

4-2.a. The tech admin modifies the user wrongly

1. Before saving the EDIT/DELETE_USER asks for a confirmation
2. Return to step 2 (MSS-T)


Tech Admin
Software
Ask again for credentials
[Fail to login]
Log in
[Login succesful]
Show functions
Show stats
Show back button
Select desired function
[Wrong function select]
[Add user]
[Edit user]
Ask data
Ask which user to edit
Insert newusername and password
Choose an user
Edit-Delete the user
Ask confirmation
Save changes
Logout

➢ Patient textual description & activity diagram

MAIN SUCCESS SCENARIO PATIENT (MSS-P):

1. The patient logs in
2. The SHOW_PROFILE shows the patient its profile w./ the statistics and the booked visits
3. The patient selects the desired function:

    3-1. Function selected: INSERT_PARAMETERS
        1. The INSERT_PARAMETERS asks for the entries

    3-2. Function selected: FILL_IN_SURVEYS
        1. The FILL_IN_SURVEY asks for the entries

4. The user interacts with the function selected:

    4-1. The patient selects to sync the data
        1. The INSERT_PARAMETERS reads the data from the wearable
        2. The INSERT_PARAMETERS saves the changes

    4-2. The FILL_IN_SURVEY submit a survey to the user
        1. The patient fills in the survey
        2. The FILL_IN_SURVEY saves the changes

5. The patient logs out

ALTERNATIVE SCENARIOS:

1a. The patient fails to log in for wrong credentials
    1. The LOG_IN_CHECK notifies the user
    2. The LOG_IN_CHECK asks for credentials again
    3. The patient inserts the credentials again
    4. Return to step 2 (MSS-P)

1b. The patient fails to log in because he's not a registered user
    1. The LOG_IN_CHECK shows an error message
    2. Return to step 1 (MSS-P)

3a. The patient selects the wrong function
    1. The software shows the back button
    2. The patient goes back to the previous page
    3. Return to step 3 (MSS)

4-1.a. The patient force a sync without new data
    1. The INSERT_PARAMETERS notifies the user: "data already up to date"
    2. Return to step 2 (MSS-P)

4-1.b. The INSERT_PARAMETERS fails to read data

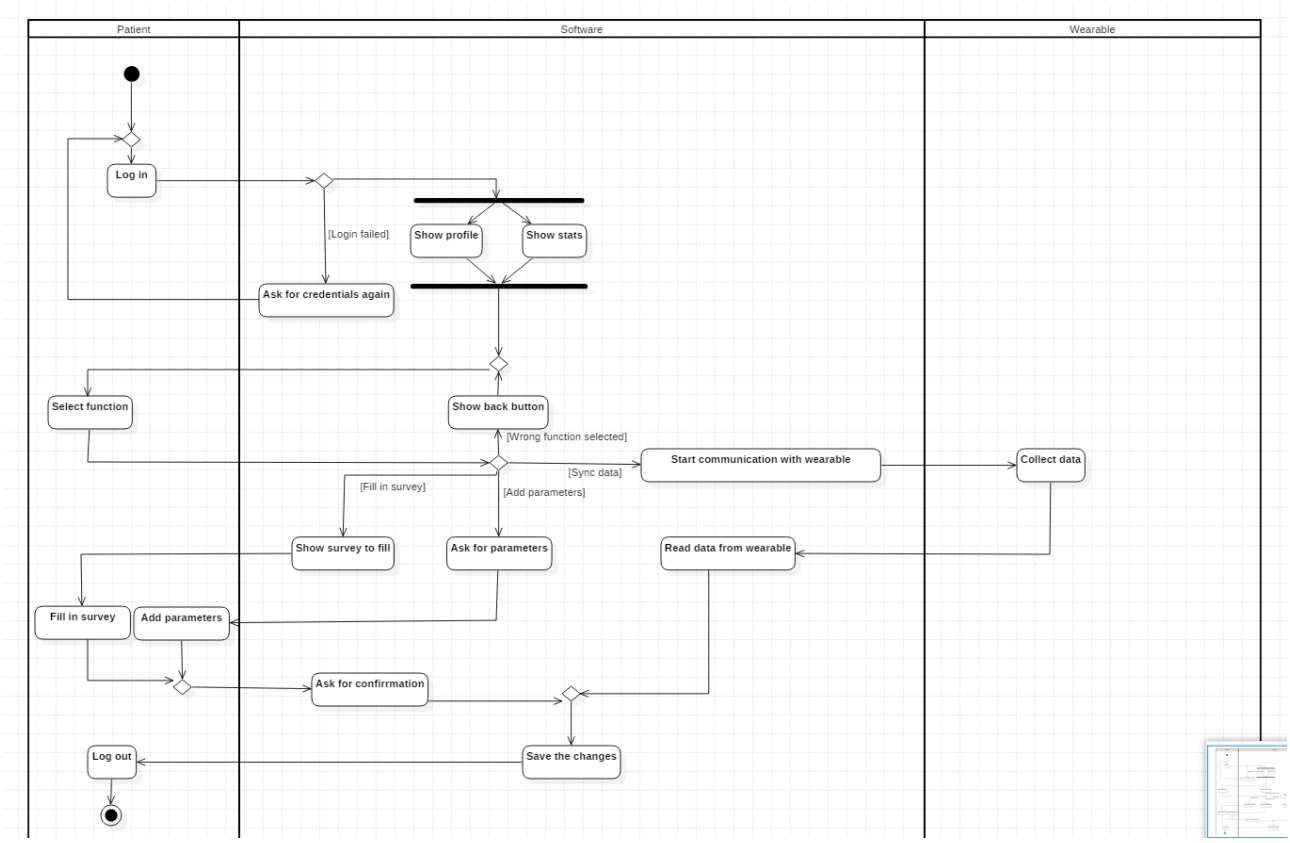
1. The INSERT_PARAMETERS notifies the user, suggesting possible solutions
2. Return to step 4-1 (MSS-P)

4-1.c. The INSERT_PARAMETERS fails to save the data

1. The INSERT_PARAMETERS tries to save the data again
2. Return to step 2 (MSS-P)

4-2.a. The patient fails to fill in the survey

1. Before saving the FILL_IN_SURVEY asks for a confirmation
2. Return to step 2 (MSS-P)



➢ Medician textual description & activity diagram

MAIN SUCCESS SCENARIO MEDICIAN (MSS-M):

1. The medician logs in
2. The SHOW_PROFILE shows the medician its profile w./ the quantities of interest
3. The medician selects the desired patient
4. The medician selects a function for the patient selected:

    4-1. The medician selects to book a visit:
        1. The VISIT_BOOKING asks for the desired type of visit
        2. The VISIT_BOOKING shows the schedule to fix a date
        3. The medician books the appointment
        4. The VISIT-BOOKING saves the changes

    4-2. The medician edits a previous visit:
        1. The EDIT_VISIT shows the editable visit fields
        2. The medician edits the desired fields
        3. The EDIT_VISIT saves the changes

    4-3. The medician adds parameters from a new visit:
        1. The ADD_VISIT_DATA starts communication with machineries
        2. The medician inserts the manual parameters
        3. The ADD_VISIT_DATA saves the changes

    4-4. The medician sets the threshold to detect abnormalities:
        1. The SET_THRESHOLDS shows a list of parameters to set the threshold to and shows the abnormalities counting
        2. The medician selects the desired parameters and set the thresholds
        3. The SET_THRESHOLDS saves the change

5. The medician logs out

ALTERNATIVE SCENARIOS:

1a. The medician fails to log in for wrong credentials
    1. The LOG_IN_CHECK notifies the user
    2. The LOG_IN_CHECK asks for credentials again
    3. The medician inserts the credentials again
    4. Return to step 2 (MSS-M)

1b. The medician fails to log in because he's not a registered user
    1. The LOG_IN_CHECK shows an error message
    2. Return to step 1 (MSS-M)

3a. The medician selects the wrong patient
    1. The WINDOW_MANAGER shows a back button
    2. Return to step 2 (MSS-M)

4a. The medician selects the wrong function

1. The WINDOW_MANAGER shows a back button
2. Return to step 4 (MSS-M)

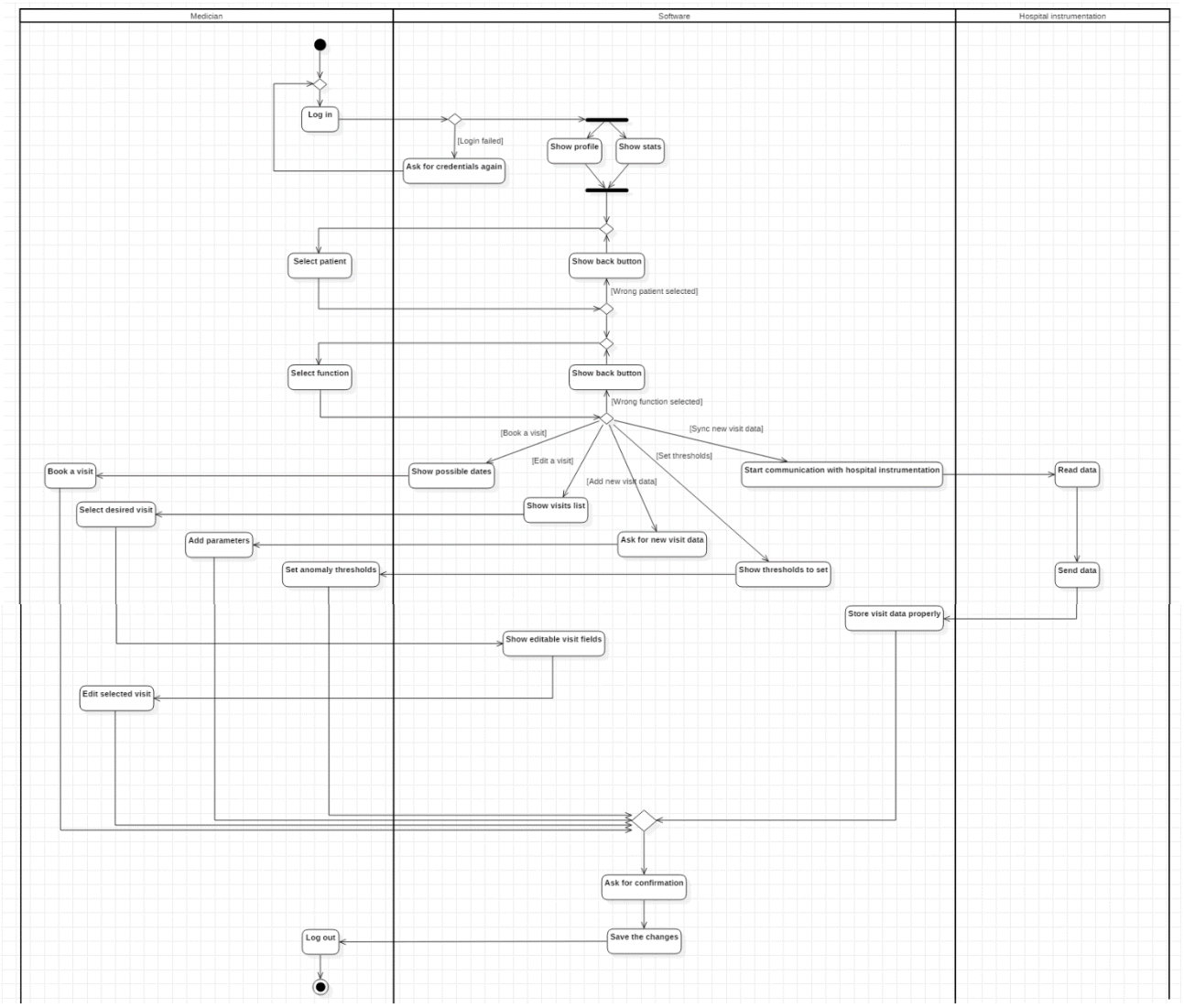
4-1.a. The medician selects the wrong date
1. Before saving the VISIT_BOOKING asks for a confirmation
2. Return to step 3 (MSS-M)

4-2.a. The medician edits the visit wrongly
1. Before saving the EDIT_VISIT asks for a confirmation
2. Return to step 3 (MSS-M)

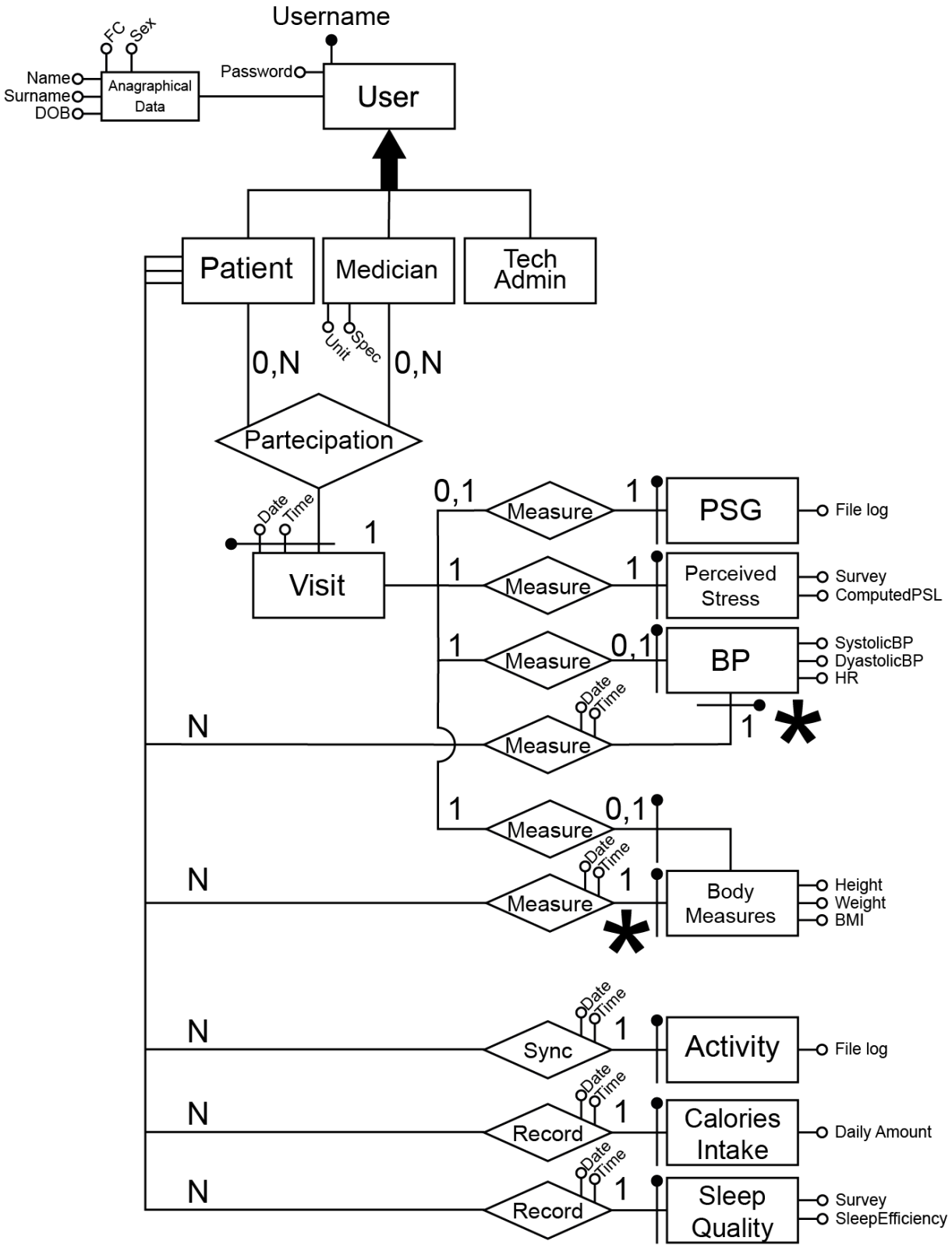
4-3.a. The ADD_VISIT_DATA fails to connect to machineries
1. The ADD_VISIT_DATA notifies the medician
2. Return to step 4-3 (MSS-M)

4-3.b. The medician add the parameters wrongly
1. Before saving the ADD_VISIT_DATA asks for a confirmation
2. Return to step 3 (MSS-M)

4-4.a. The medician set the thresholds wrongly
1. Before saving the SET_THRESHOLDS asks for a confirmation
2. Return to step 3 (MSS-M)



ER Diagram


Username
FC
Sex
Name
Surname
DOB
Anagraphical Data
Password
User
Patient
Medician
Tech Admin
Unit
Spec
0,N
0,N
Partecipation
Date
Time
1
Visit
0,1
Measure
1
PSG
File log
1
Measure
1
Perceived Stress
Survey
ComputedPSL
1
Measure
0,1
BP
SystolicBP
DyastolicBP
HR
N
Measure
1
1
Measure
0,1
N
Measure
1
Body Measures
Height
Weight
BMI
N
Sync
1
Activity
File log
N
Record
1
Calories Intake
Daily Amount
N
Record
1
Sleep Quality
Survey
SleepEfficiency

- Notes:

*

The BP and body measures are recordable both by the patient at home and by the specialised practitioner during the visit, thus these entities can be determined by their relation to a specific visit, or their home measure at a specific date and time.

During implementation phase we plan to group both cases into a single table, utilizing a nullable foreign key to the visit table to determine if the measure is linked to a visit or taken at home.