UNIVERSITY OF PADOVA
DEPARTMENT OF INFORMATION ENGINEERING

Biomedical Wearable Technologies
for Healthcare and Wellbeing

Introduction

A.Y. 2023-2024

Giacomo Cappon

DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

Agenda

- Teacher introduction
- What is the course about
- The architecture of wearable IoT systems
- Applications of wearable IoT in healthcare and wellbeing
- Course program
- The exam & The project
- Appendix: Data source characteristics

Who am I

- **Giacomo Cappon**
  - Assistant Professor, Department of Information Engineering, University of Padova
  - PhD in Information Engineering, curriculum Bioengineering
  - Main research interests: signal processing; decision support systems; wearable sensors; digital health; ai for medicine

- Contacts
  - email: giacomo.cappon@unipd.it
  - personal website: https://gcappon.github.io
  - public repos: https://github.com/gcappon

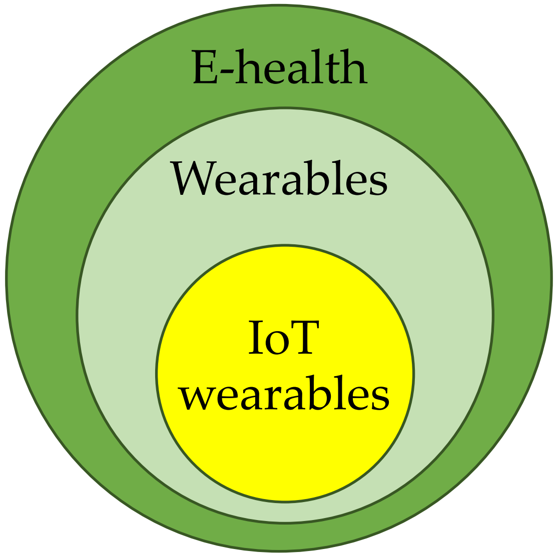
What is the course about?

Biomedical wearable technologies…

- **E-health** or **digital health**: The use of modern Information and Communication Technologies (ICT) to efficiently deliver all kinds of healthcare services
  - Examples: EHR, telemedicine systems

- **Wearables**: electronic devices worn close to and/or on the surface of the skin, which detect, analyze and transmit biomedical data, such as heart rate, activity, blood pressure, temperature, etc.

- **Internet-of-thing (IoT) wearables**: Wearables that connect and exchange data with other devices (wearable or not) and systems over the Internet or other communication networks



What is the course about?

… for healthcare and wellbeing

- Our society is facing some important global health challenges
  - **Lifestyle-related illnesses** driven by sedentary life, consumption of processed and fast foods, alcohol drinking and smoking
    - Need to change the paradigm of healthcare, from a mostly reactive system to a **proactive system**
  - **Population ageing** rises the demand of health services and the healthcare expenditure.
    - Need to **decentralizing the healthcare**
    - Covid-19 pandemic called for urgent delivery of healthcare at home

Biomedical wearable technologies are fundamental to make healthcare scalable and accessible outside the hospital setting and deliver personalized and preventive health services

Towards a new model of healthcare

- **Traditional healthcare system**:
  - **Reactive**: focused on treating
  - **Centralized**: healthcare at hospitals
  - **Population-based**: maximizing health outcomes of a certain population
  - **Doctor-centric**: only doctor responsible for the patient's care

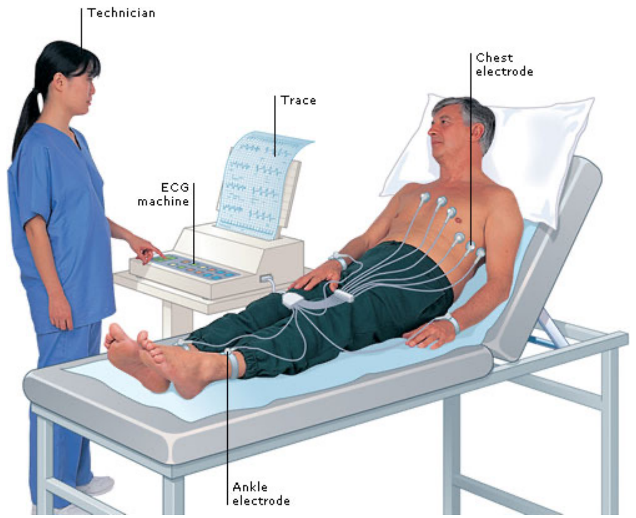
- Biomedical wearable technologies play an important role in realizing a **new model of healthcare** which is:
  - **Decentralized**: healthcare at home to reduce pressure on hospitals
  - **Proactive**: focused on prevention
  - **Personalized**: maximising the health outcomes of single individuals
  - **Participatory**: engaging individuals in the healthcare process
  - **Sustainable**: better management of resources, patient prioritization

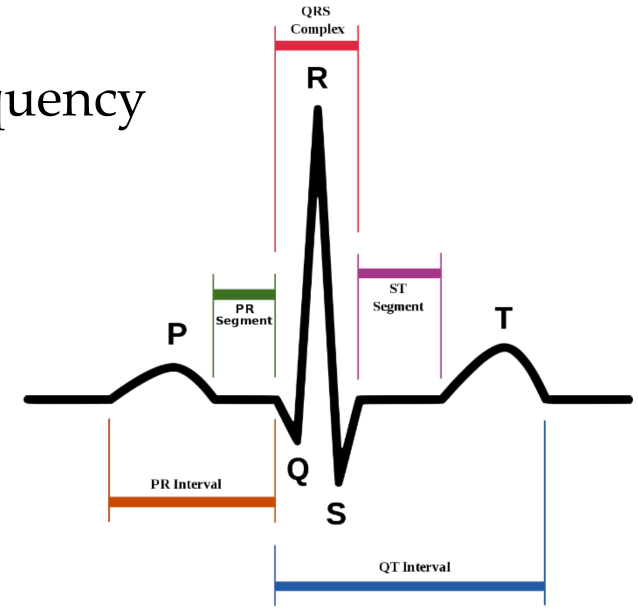
Examples of biomedical wearable technologies
Heart rate monitor

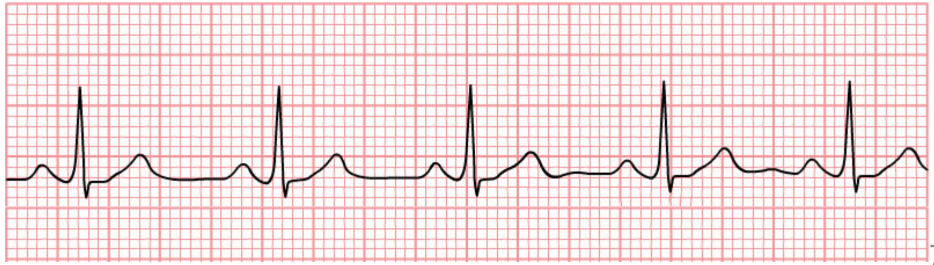
- Heart rate (HR): frequency of heart beats [beats/min]
- Monitoring HR allows to detect bradycardia (slow HR), tachycardia (fast HR) or arrhythmia (irregular HR)

In hospital:

- **Electrocardiogram (ECG)**: electrical activity of the heart measured by on skin electrodes (usually 12)
- HR determined as the frequency of peaks of the R wave



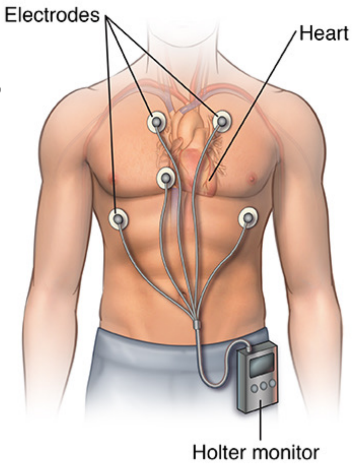




Examples of biomedical wearable technologies
Heart rate monitor

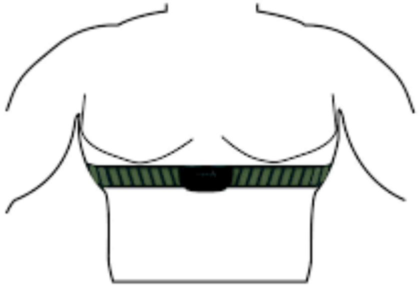
- Wearable HR monitors for home use:

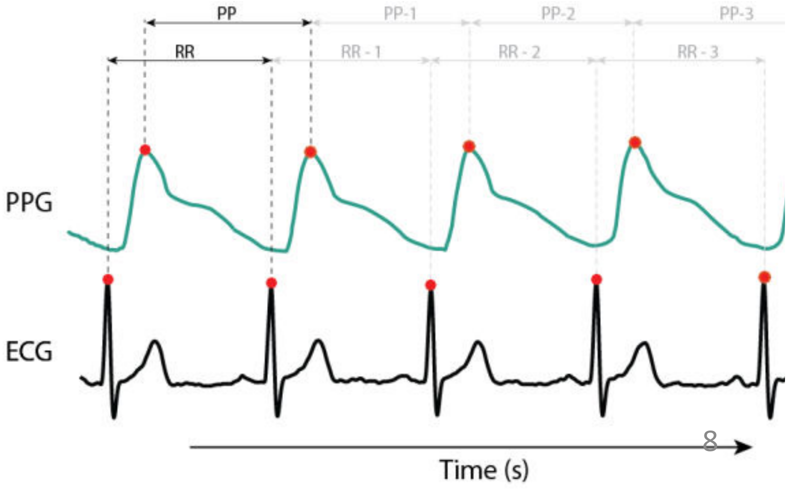
- **Holter monitor**: portable ECG monitor, 5 electrodes placed on the chest
- Used for medical purpose
- Low portability, used for 24-72 hours



- **Smartwatches** use an optical sensor that shines light and measures changes in light absorption to estimate changes in capillary blood volume
- Resulting signal: photoplethysmogram (PPG)
- HR estimated as the frequency of P waves

- **Chest straps**: electrical activity of the heart measured with electrodes integrated in a chest band/strap





Examples of biomedical wearable technologies
Glucose monitoring sensors

- Blood glucose (BG) concentration: concentration of glucose in the blood [mg/dL or mmol/L]
- Abnormal BG levels, e.g., hyperglycemia (too high BG) and hypoglycemia (too low BG), may occur in people with metabolism disorders (e.g., diabetes).

In hospital/laboratory:

- **Biochemistry analyzers**
  - Exploit a chemical reaction driven by specific enzymes (e.g., glucose-oxidase)
  - Invasive (blood sample required), only sparse measurements possible

At home:

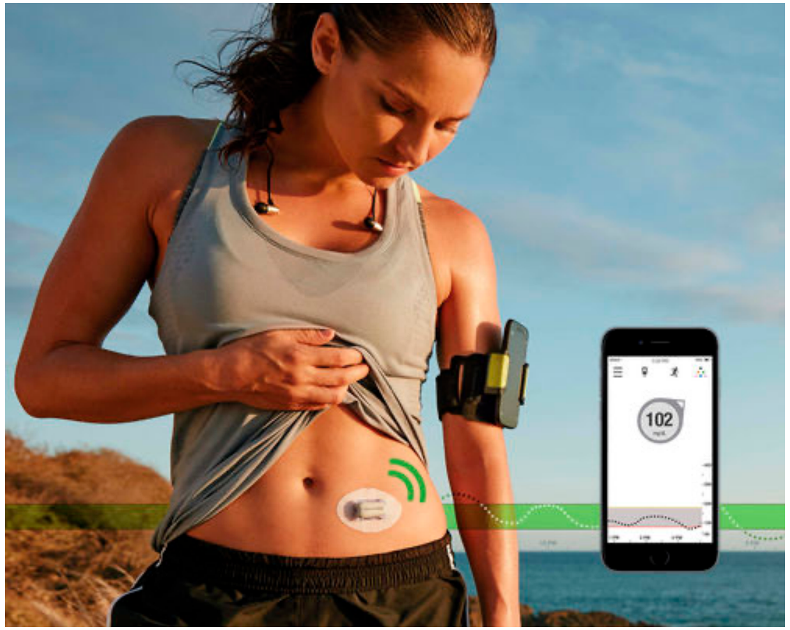
- **Self-monitoring of blood glucose (SMBG) or glucometer**:
  - drop of capillary blood collected by fingerprick and put in a test strip
  - a portable glucose analyzer, through an electrochemical sensor, detects the BG concentration in the capillary blood
  - Minimally-invasive but not wearable: only sparse measurements possible (e.g., 3-4 per day)

Examples of biomedical wearable technologies
Glucose monitoring sensors

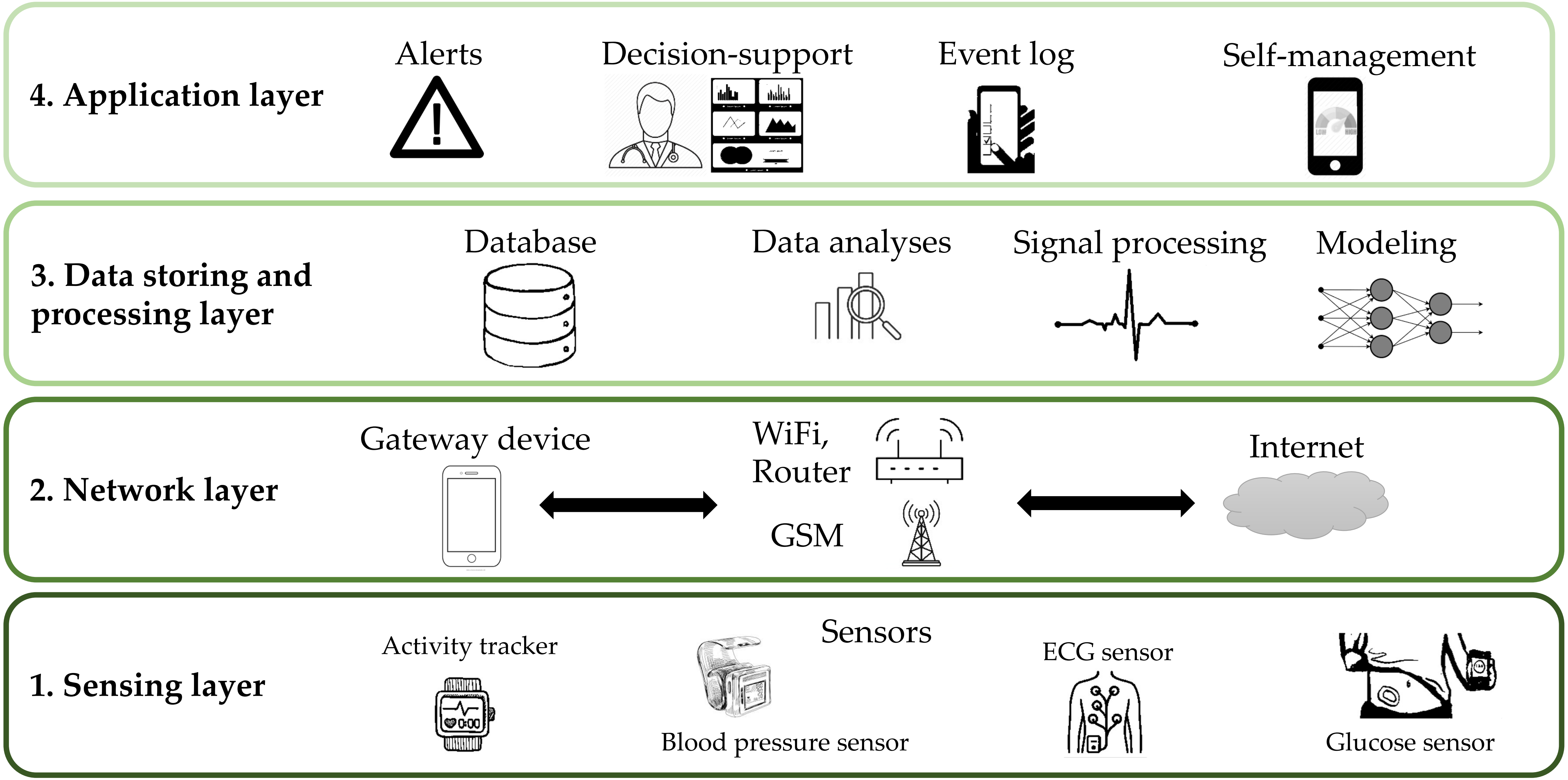
- **Continuous glucose monitoring (CGM) sensors**
  - Minimally-invasive sensors measuring glucose concentration in the interstitial fluid at almost continuous time (e.g., every 5 min) for several consecutive days (e.g., 10 days)
  - Needle electrode placed under the skin, exploiting glucose oxidation reaction
  - The transmitter, placed over the electrode, wirelessly transfers the measured data to a portable display device
  - Mainly used in people with type 1 diabetes

Modern wearable technologies become IoT

- Modern wearable sensors can wirelessly connect to internet-connected devices, such as smartphones, by which the user can share their data with other people (e.g., doctor or family/friends) and upload them in the cloud



The architecture of wearable IoT systems



1. Sensing layer

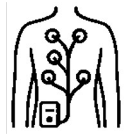
- This layer includes sensors, which are responsible for:
  - **Collecting the data** either directly from the body through contact sensors or from peripheral sensors providing indirect information of body and behaviours
  - **Preparing the data** for on-board analysis, closed-loop feedback or remote transmission for more comprehensive analysis and storing

- Key components of IoT wearable sensors:
  - Miniaturized **sensor hardware**
  - **Embedded processor** with storage capabilities, power management and communication circuits

- Typically limited computing power and storage capabilities.

Contact sensors

ECG sensor



Glucose sensor

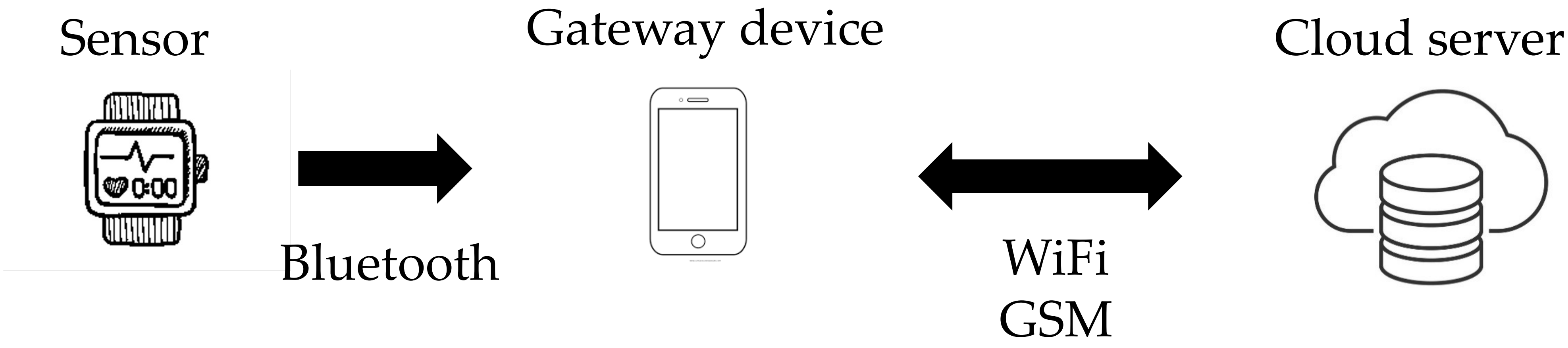
Peripheral sensors

Step counter

Microphone and camera

2. Network layer

- Fore more comprehensive data analysis and storage, the sensor layer need to transmit the data to **cloud computing servers** (remotely-located computing machines).
- This transmission can be mediated by a companion device that serves as a **gateway**, i.e., it enables the information to flow from sensors to the cloud server.
  - Example of gateway devices: smartphones, tablets and PCs.
- The communication must be **secured** through:
  - Authentication protocols for writing, modifying and accessing the data
  - Security and privacy protocols for preserving the data



3. Data storing and processing layer

- Layer responsible for data storing and complex processing

- Key components:
  - **Database** for structured data storing
    - Data organized based on a specific data model (relational, hierarchical, object-oriented,…)
  - **Algorithms** for data processing
    - Statistical analysis
    - Feature extraction (e.g., heart rate, calories expenditure, …)
    - Signal processing (e.g., filtering, artifact removal, …)
    - Data mining (e.g., cluster analysis, anomaly detection, sequential pattern extraction, …)
    - Data modeling (knowledge-based, data-driven or hybrid approaches)
    - Artificial intelligence (knowledge representation, machine learning, natural language processing, machine perception, automated planning, …)

3. Data storing and processing layer

- Layer responsible for data storing and complex processing

- Key components:

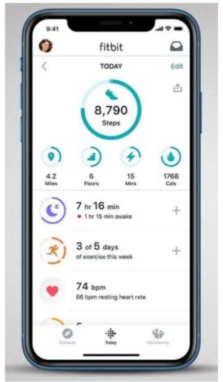
Topic not covered

  - ~~**Database** for structured data storing~~
    - ~~Data organized based on a specific data model (relational, hierarchical, object-oriented,…)~~
  - **Algorithms** for data processing
    - Statistical analysis
    - Feature extraction (e.g., heart rate, calories expenditure, …)
    - Signal processing (e.g., filtering, artifact removal, …)
    - Data mining (e.g., cluster analysis, anomaly detection, sequential pattern extraction, …)
    - Data modeling (knowledge-based, data-driven or hybrid approaches)
    - Artificial intelligence (knowledge representation, machine learning, natural language processing, machine perception, automated planning, …)

4. Application layer

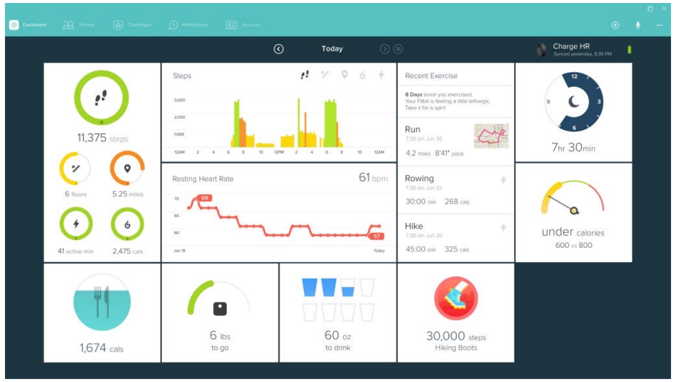
- Layer responsible for visualizing raw or processed data to end-users and for providing decision support to them through a software application
- Possible end-users: patients, physicians, caregivers, analysts, data managers, administrators, etc.
- Possible types of applications:

Mobile app



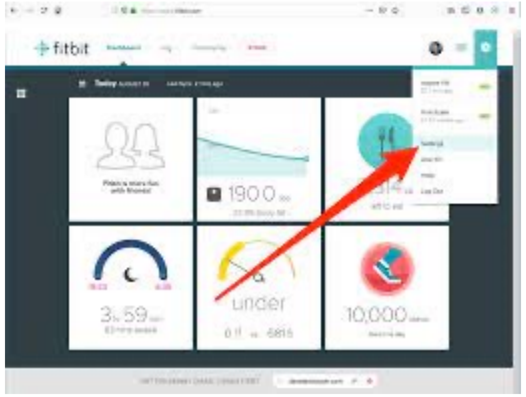
App installed in a mobile device (e.g. smartphone, smartwatch, tablet, etc.)

Desktop app



App installed in a computer.
It can work offline.

Web app



App that runs on the browser.
Installation not required.
Internet access required.

Applications of wearable IoT in healthcare and wellbeing

Some health applications:

- Telemonitoring and telemedicine in chronic diseases
- Investigative monitoring for diagnostic purposes
- Sensor-based biomechanical telerehabilitation

Some wellbeing applications:

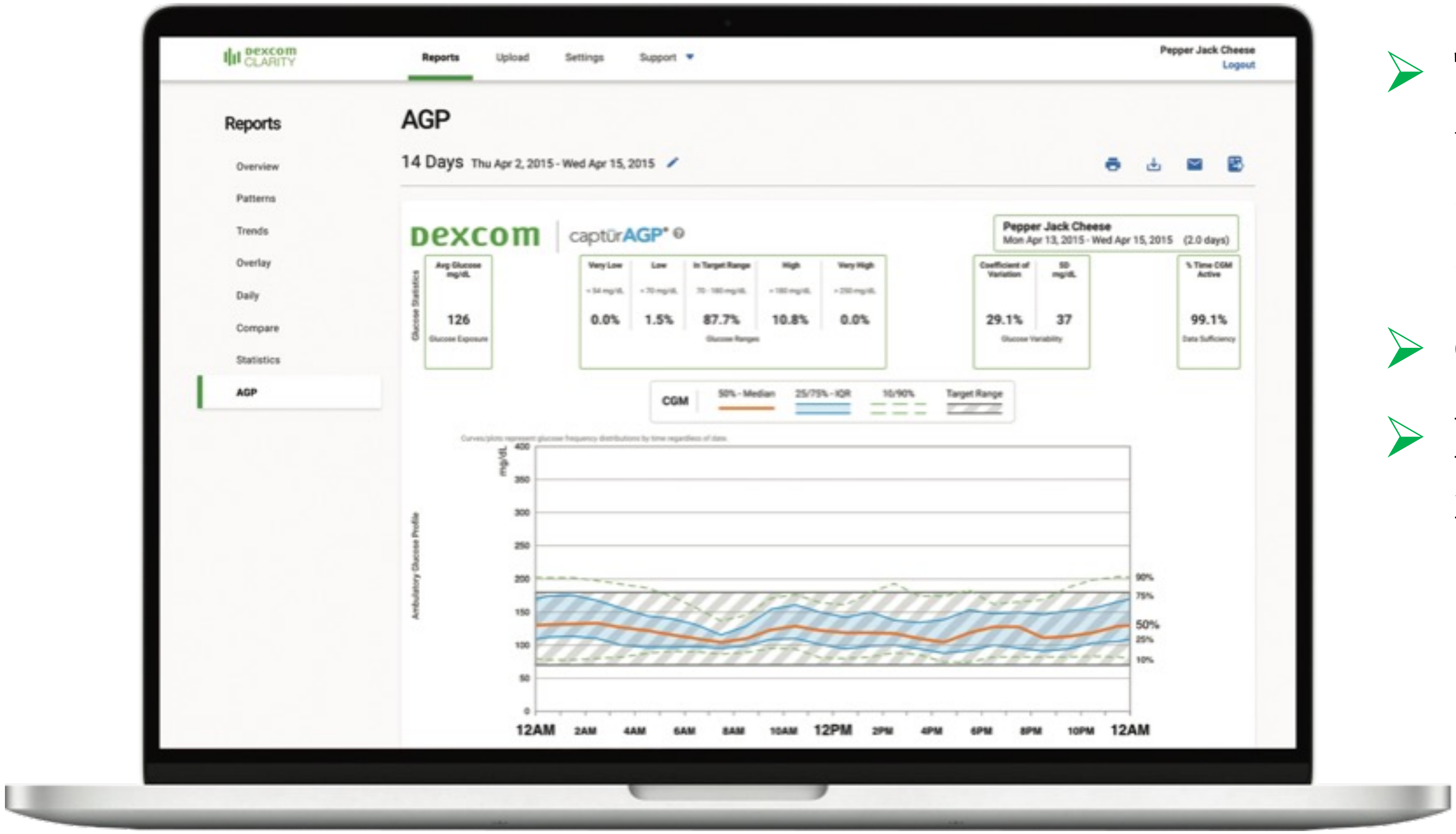
- Sport and fitness sensing
- Activity and sleep monitoring
- Falls detection

Telemonitoring and telemedicine in chronic diseases

- **Chronic diseases**:
  - Conditions that last 1 year or more and require ongoing medical attention or limit daily activities
  - About 70-80% of the entire healthcare expenditure
- **Telemonitoring**: patient's data collected by sensors are transmitted to the doctor that review the data
- **Telemedicine**: if necessary, the doctor makes decisions regarding the patient's health state and provides feedback to the patient
- Affordable, clinical-grade sensing for home use are required
- Telemonitoring and telemedicine can greatly improve chronic disease management
  - Reduce visits and their duration
  - Early detection and reaction to adverse health events
  - Notify caregivers about health changes that may go unreported or unnoticed by the patient but that may represent early warning signs of an impending health event

Telemonitoring and telemedicine in chronic diseases

- Example: Telemonitoring of glucose concentration in people with diabetes



- Through a dedicated telemonitoring application, the doctor can remotely review patients data at any time
- Continuous data synchronization
- Remote monitoring also possible for parents and caregivers

Investigative monitoring for diagnostic purpose

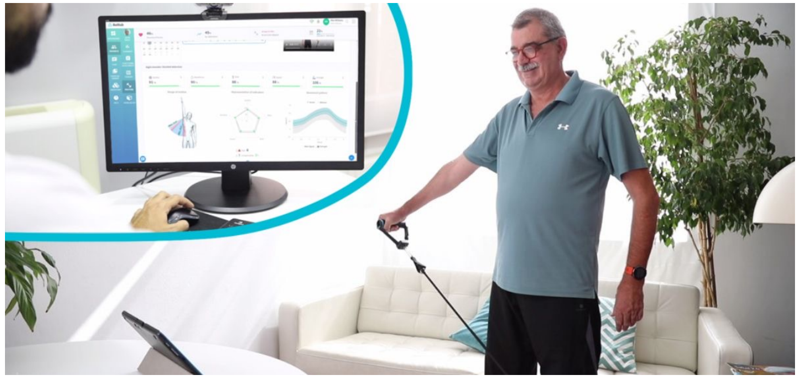
- Monitoring of health parameters in **outclinic context**, for a limited period of time (e.g., few days) to **detect** a particular health condition that cannot be observed in clinic, to track the progress of a disease, or to **monitor** recovery from a specific health event.

- Examples:
  - **Professional CGM sensor**: CGM provided to the patient for a short period (e.g., 7 days). Based on the collected data the doctor can adjust/optimize the patient’s therapy.
  - **Holter and extended-holter systems**: monitor HR for limited periods (e.g., 1-7 days) for detection of heart issues, e.g., cardiac arrhythmias
  - **At-home sleep test** for the diagnosis of obstructive sleep apnea (stops in breathing during sleep): monitoring of sleep at home for at least 1 night using wearable sensors provided by the doctor (breathing sensor, effort belt, pulse oximeter, and microphone)

Sensor-based biomechanical telerehabilitation

- Key components:
  - Wearable **inertial sensors** tracking patient's movement
  - **Interactive software**
    - displays personalized rehabilitation exercises
    - receives the sensors' data
    - provides the real-time representations of the patient
    - provides visual feedback to the patient, e.g. using an avatar
  - A **platform for clinicians** to remotely follow the therapy sessions in real-time or review the data offline

- Main application: remote monitoring of patient performance and progress in a rehabilitation programs (e.g., post-stroke, post-fall, or post-surgical operation)



Sport and fitness sensing

- Most common sensors: chest straps, wristbands, footpods and insoles, sensor patches, sensors in sportwear clothing (smart clothing), sensors in sportwear equipement
- **Real-time monitoring of performance and coaching** through portable display devices
- **Off-line review of sport statistics**, on web-based portals
- **Data sharing and social networking**
- **Gamification of fitness data**: gaming features (e.g. fitness challenges) encourage the adoption of fitness technologies.

Activity and sleep monitoring

- Measurements of activity (steps, distance traveled, etc.), calorie balance, sleep quality.
- Objectives:
  - **Encourage movement** in sedentary individuals
  - **Obesity management**: monitor an individual's participation in a weight management program and maintain engagement
  - **Sleep monitoring** by contact sensors (e.g. wristbands) or non-contact sensors (e.g. under-mattress sleep trackers)

<u>Baby sleep monitoring sensors</u>:

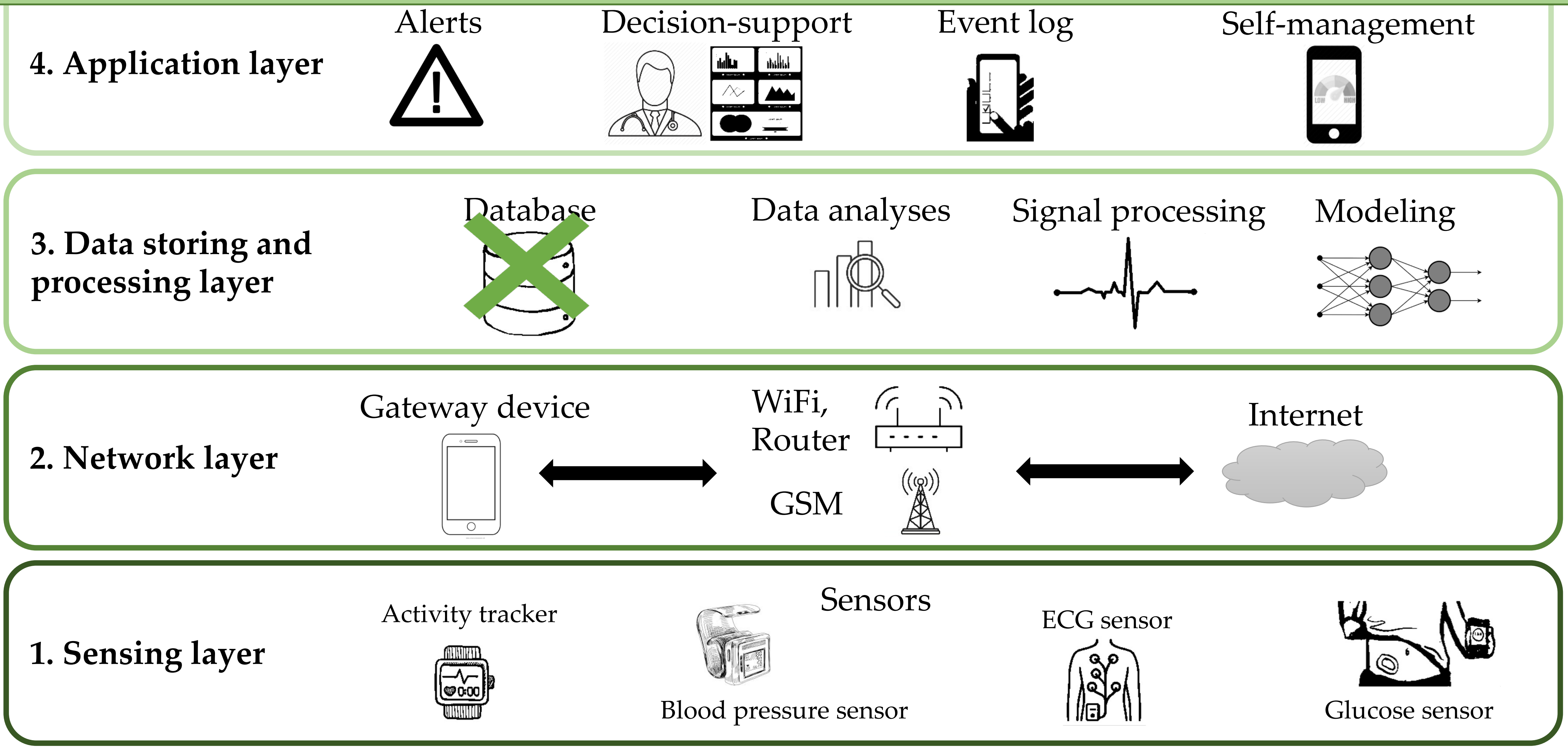
- Inertial sensors monitor movement of the chest or stomach area
- Built-in tactile stimulator triggered if no movement is detected for 15–20 seconds
- If no movement detected after use of tactile stimulation, an alert is generated

Fall detection

- Falls-related injuries are the biggest cause of injury-related death in older adults
- **Fall detection devices**:
  - Detection device based on **inertial sensors** worn on the wrist or waist or as a pendant.
  - An algorithm analyses movement measurements in real-time and sends an **alert** to an in-home base station if a fall is detected.
  - The in-home base station notifies a call center, which contacts the wearer.
  - If an emergency is determined or there is no response, the call center notifies the **emergency services** and alerts designated caregivers and family.
- Also available in smartwatches used for sport sensing

The architecture of wearable IoT systems

In this course we will go through the four layers of wearable IoT systems starting from layer 1.



Course program

- 12 theory lectures
  - Monday 12:30-14:30, Room 14E (Padiglione 14 – Fiera di Padova)

- Contents:
  - Wearable sensor technologies
  - World wide web, HTTP communication protocol
  - The representational state transfer (REST) paradigm
  - Application programming interfaces (APIs)
  - Elements of cryptography
  - Authentication protocols
  - Health design thinking
  - Evaluation of app usability
  - Regulation for health data management

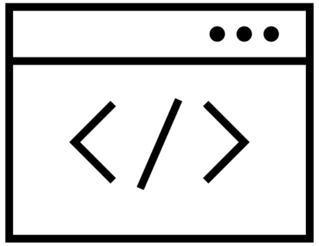
Course program

- 12 laboratory lectures
  - Tuesday 14:30-16:30, Rooms Te+Ue (Polo Didattico di Via Gradenigo)

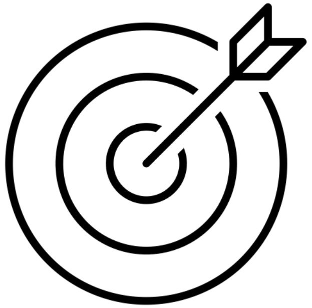
- Contents:
  - The Git version control system
  - Dart programming language and the Flutter framework
  - Building mobile/web applications in Flutter
  - User Interface design
  - Navigation
  - State management
  - Basic data persistence
  - Networking

Prerequisites & End of class goals

- Prerequisites
  - Basic:
    - Propension for programming
  - Preferable:
    - Object-oriented programming



- End of class goals
  - Learn how wearable devices communicates to the end-user
  - Design digital solutions to enhance healthcare and well-being
  - Develop a fully functioning mobile application



The exam

Dates of the written test

- First round: 20 Jun 2024, 14:00 (Room Be+Ve)
- Second round: 08 Jul 2024, 14:00 (Room Ae+Be)
- Third round: 03 Sep 2024, 14:00 (Room De)
- Fourth round: 31 Jan 2025, 14:00 (Room De)

- **Written test**: Theory exam
  - 15 closed questions
  - 3 open questions
  - Duration: ~90 min
  - Max vote: 20/30
- **Project**: Development of an application that integrates and elaborates data from a smartwatch. To be done in groups of 3 students.
  - Deadline for completion: The project code must be delivered by July 14th 2024, and no later than the day before the project discussion.
  - Discussion of the project (in group): day and time to be defined in dedicated
    - Window 1: from 01/07/2024 to 19/07/2024
    - Window 2: from 19/08/2024 to 20/09/2024
  - Duration of the discussion: 45 min (25 min presentation + 20 min Q&A)
  - Max vote: 12/30
- **Final mark**: sum of the votes of the written test and the project.

Project: What's the spirit here

- The goal is learning **"How to do SOMETHING using wearable devices"**

- What **I DO** expect from you
  - To use your imagination to create that SOMETHING
  - To learn to build something from scratch
  - To use code built by others
  - To work together
  - To use Google and StackOverflow

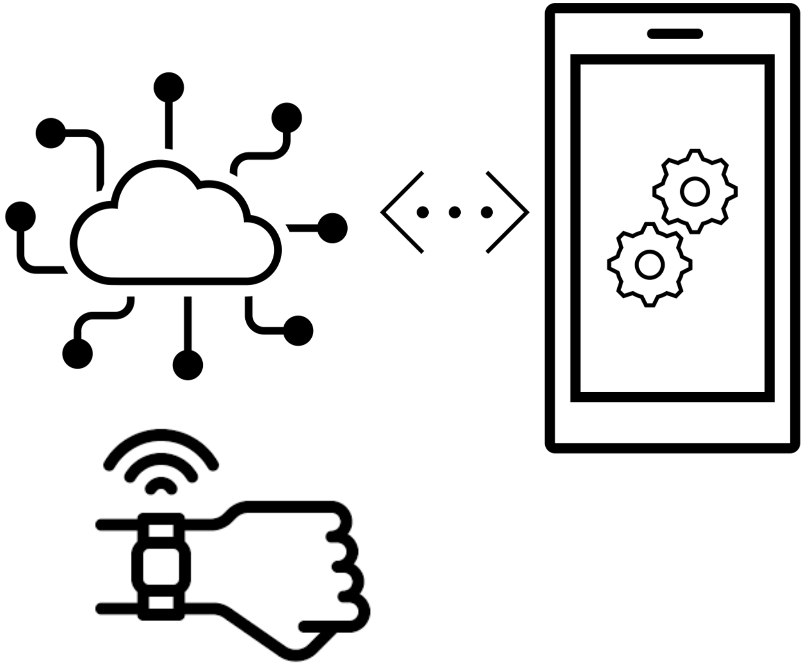
- What **I DO NOT** expect from you
  - To create the new Google
  - To reinvent the wheel (copying others code is fine)
  - To create extra complicated UI with animations and stuff
  - To implement sophisticated design pattern (even if it would be a nice thing to do)

Project structure

- The project consists of building an app for iOS or Android that collects user data from a wearable device through Web APIs, visualizes them, and does some tricks with them.

- Core functionalities:
  - Correct app flow
  - User authentication and management
  - Data collection
  - Data visualization and presentation

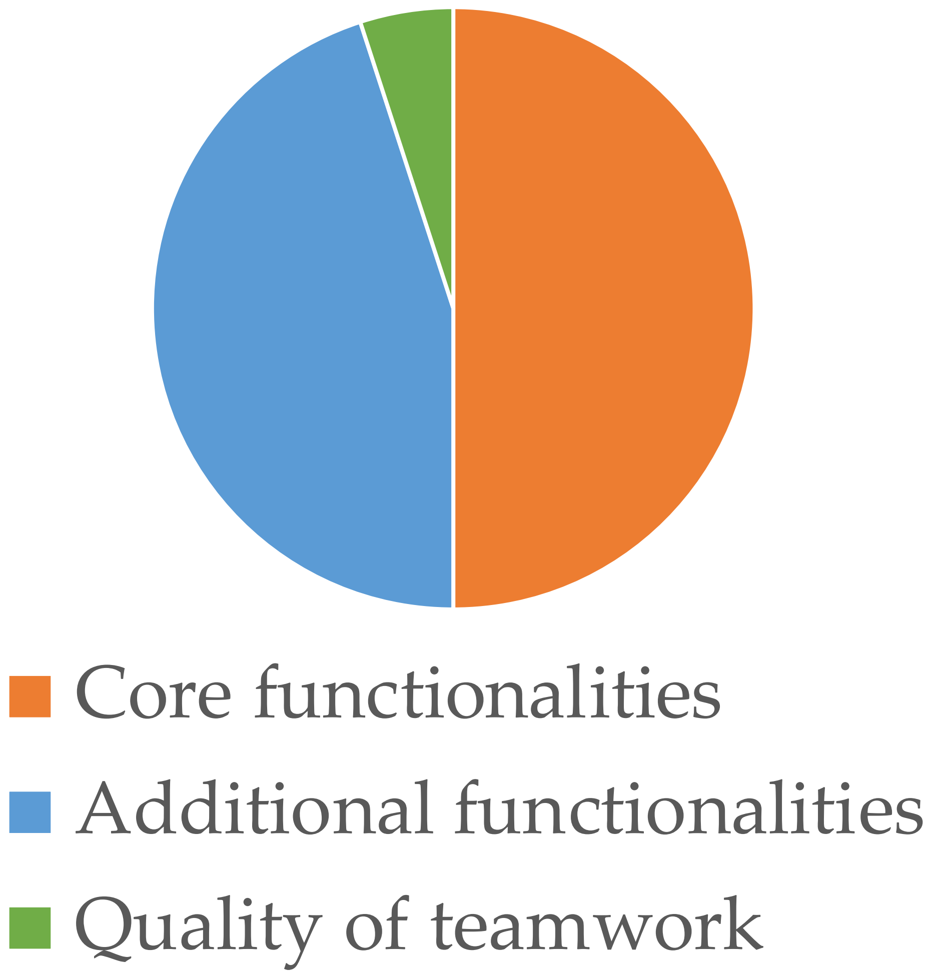
- Additional fuctionalities → It's up to you! Some examples:
  - Run some analysis on data and provide suggestions to the user
  - Implement some literature algorithm
  - …



Grading

- What are the grading criteria
  - Compliance to core functionalities
  - Originality of the additional feature
  - Quality of teamwork

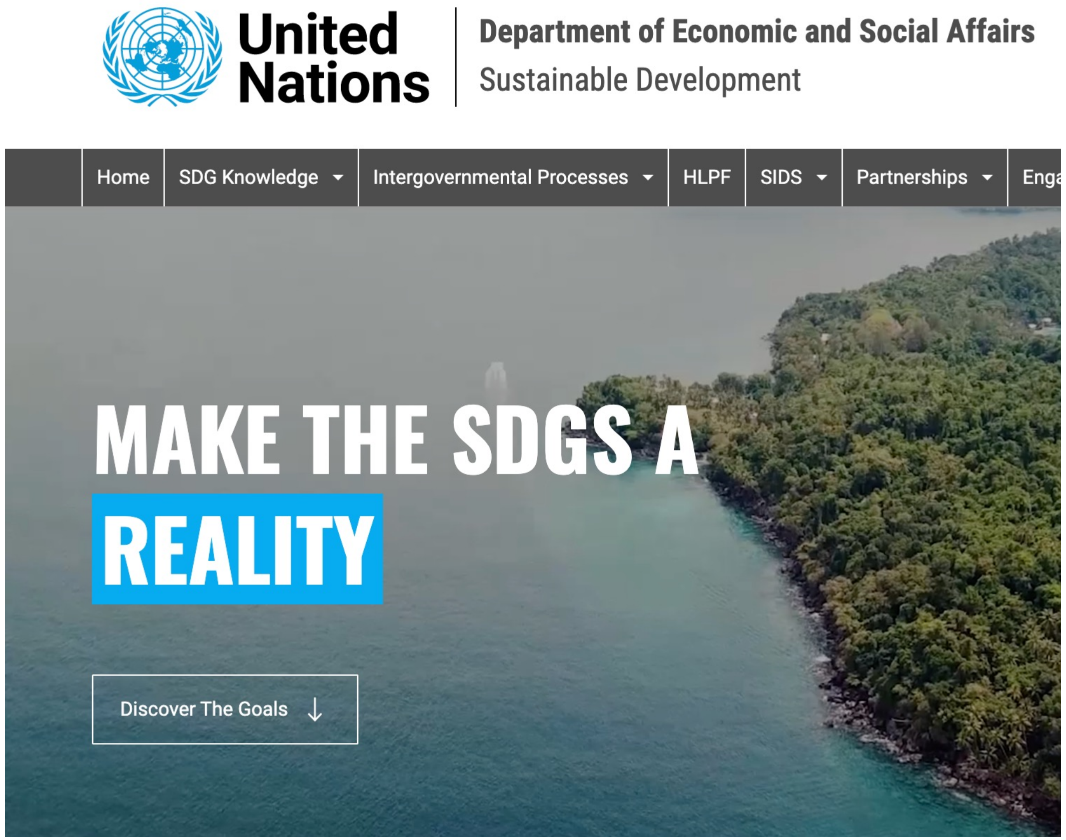
- How much these criteria matter?
  - Core functionalities: 50%
  - Additional app functionalities: 45%
  - Quality of teamwork: 5%



Choose your target

- Your app must represent a solution for the current SDGs (Sustainable Development Goals) targets

- SDG are 17 topics representing the current big challenges humanity must face and solve



Choose your target

- These are the target you can choose from (see the elearning page for more details):

  - <u>**SDG 2: Target 2.2**</u> (By 2030, end all forms of malnutrition, including achieving, by 2025, the internationally agreed targets on stunting and wasting in children under 5 years of age, and address the nutritional needs of adolescent girls, pregnant and lactating women and older persons) – **max 4 groups**
  - <u>**SDG 2: Target 2.3**</u> (By 2030, double the agricultural productivity and incomes of small-scale food producers, in particular women, indigenous peoples, family farmers, pastoralists and fishers, including through secure and equal access to land, other productive resources and inputs, knowledge, financial services, markets and opportunities for value addition and non-farm employment) – **max 4 groups**
  - <u>**SDG 3: Target 3.4**</u> (By 2030, reduce by one third premature mortality from non-communicable diseases through prevention and treatment and promote mental health and well-being) – **max 8 groups**
  - <u>**SDG 3: Target 3.5**</u> (Strengthen the prevention and treatment of substance abuse, including narcotic drug abuse and harmful use of alcohol) – **max 8 groups**
  - <u>**SDG 4: Target 4.7**</u> (By 2030, ensure that all learners acquire the knowledge and skills needed to promote sustainable development, including, among others, through education for sustainable development and sustainable lifestyles, human rights, gender equality, promotion of a culture of peace and non-violence, global citizenship and appreciation of cultural diversity and of culture's contribution to sustainable development) – **max 4 groups**
  - <u>**SDG 8: Target 8.9**</u> (By 2030, devise and implement policies to promote sustainable tourism that creates jobs and promotes local culture and products) – **max 4 groups**

Data source

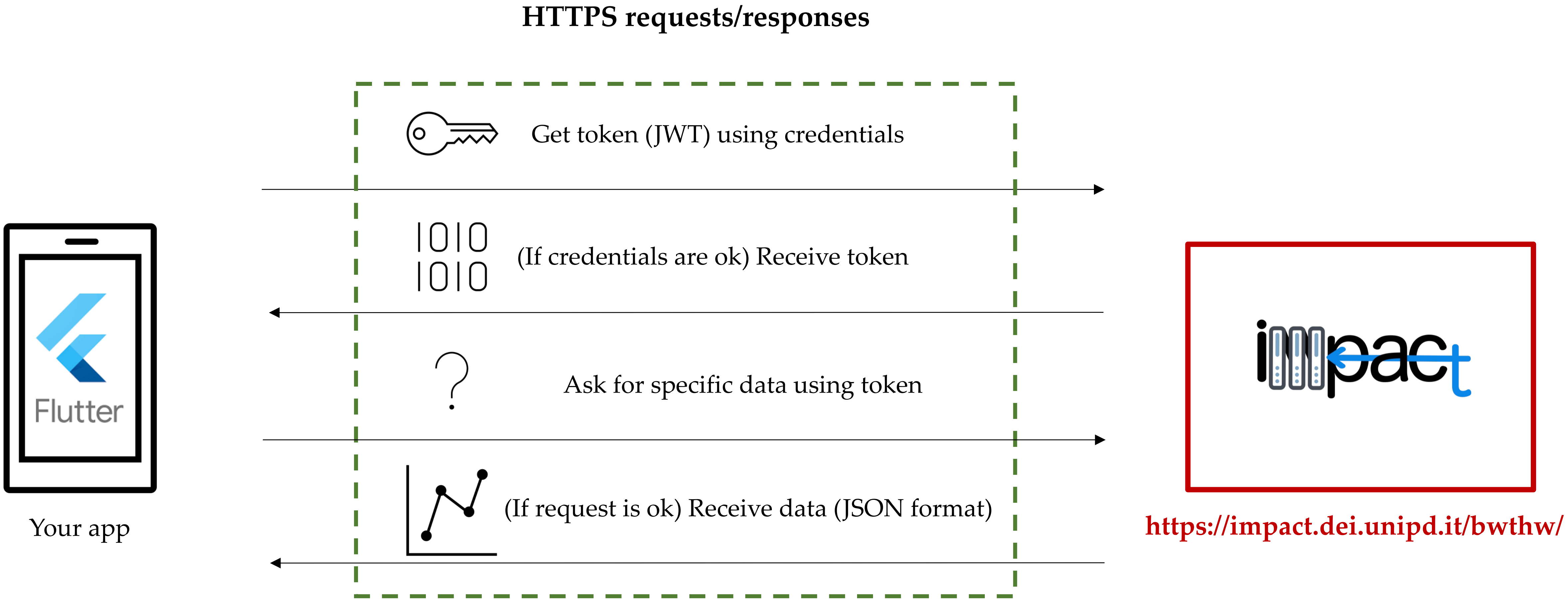
- Data that you will use are collected by me using a Fitbit Versa 2

- Data are available starting from 09/02 and will be collected up to 30/11

- The following data “types” are included(see mor details in the Appendix):
  - Calories
  - Distance
  - Exercise sessions
  - Heart rate
  - Resting heart rate (Estimated)
  - Sleep
  - Steps

- Your project must use at least 1 type of data

How to get data - The IMPACT backend



➢ You will be provided with credentials (different for each group) and learn how to get data in the next lessons (lab lesson 9 and 10)

Tutoring activities

- Need help? They are here to cover your back with answers and extra tutoring activities

- Luca Cossu

luca.cossu@phd.unipd.it

- Michele Atzeni

michele.atzeni@phd.unipd.it

Calendar of the tutoring activities

Day	Time	Room	Topic
07/03/2024	10:30 - 12:30	Ee	Tutoring 1: Exercises in Dart
14/03/2024	10:30 - 12:30	Ee	Tutoring 2: Exercises in Dart
21/03/2024	10:30 - 12:30	Ee	Tutoring 3: Sample App Presentation + Setup
11/04/2024	10:30 - 12:30	Ee	Tutoring 4: UI + Navigation
18/04/2024	10:30 - 12:30	Ee	Tutoring 5: State Management
02/05/2024	10:30 - 12:30	Ee	Tutoring 6: SharedPreferences + Authentication
09/05/2024	10:30 - 12:30	Ee	Tutoring 7: HTTP Data request
16/05/2024	10:30 - 12:30	Ee	Tutoring 8: Algorithm
23/05/2024	10:30 - 12:30	Ee	Tutoring 9: Wrapping up
28/05/2024	14:30 - 16:30	Te	Tutoring 10: Bonus

Discord server

New

- Join the Discord server to get in touch and ask questions

- **Link: https://discord.gg/j76A3D27bm**

Appendix: Data source characteristics

Focus on data – Calories

- Calories: a list with 1 data point exactly every minute

- Fields meaning:
  - time: timestamp (hh:mm:ss format) of the calory entry
  - value: the calories spent during the last 1 minute (rest + active)


```
[
        {
        "time": "00:00:00",
        "value": "1.29"
        },
        {
        "time": "00:01:00",
        "value": "1.29"
        },
        {
        "time": "00:02:00",
        "value": "1.29"
        },
        {
        "time": "00:03:00",
        "value": "1.55"
        },
        {
        "time": "00:04:00",
        "value": "1.29"
        },
        {
        "time": "00:05:00",
        "value": "1.29"
        },
        {
        "time": "00:06:00",
        "value": "1.29"
        },
        ...
]
```

Focus on data – Distance

- Distance: a list with 1 data point every minute or more depending on user’s movement
- Fields meaning:
  - time: timestamp (hh:mm:ss format) of the distance entry
  - value: the distance travelled between the last timestamp and now (in cm)


```
[
    {
        "time": "00:03:00",
        "value": "0"
    },
    {
        "time": "00:50:00",
        "value": "0"
    },
    {
        "time": "01:28:00",
        "value": "0"
    },
    {
        "time": "01:32:00",
        "value": "1440"
    },
    {
        "time": "01:33:00",
        "value": "3110"
    },
    {
        "time": "01:34:00",
        "value": "0"
    },
    {
        "time": "01:53:00",
        "value": "0"
    },
    {
        "time": "03:09:00",
        "value": "0"
    },
    ...
]
```

Focus on data – Steps

- Steps: a list with 1 data point every minute or more depending on user's movement

- Fields meaning:
  - time: timestamp (hh:mm:ss format) of the step entry
  - value: the number of steps done between the last timestamp and now

- **Note**: there can be days without data


```
[
    {
        "time": "00:07:00",
        "value": "0"
    },
    {
        "time": "00:10:00",
        "value": "0"
    },
    {
        "time": "01:57:00",
        "value": "0"
    },
    {
        "time": "02:29:00",
        "value": "0"
    },
    {
        "time": "02:40:00",
        "value": "0"
    },
    {
        "time": "02:57:00",
        "value": "11"
    },
    {
        "time": "02:58:00",
        "value": "18"
    },
    {
        "time": "02:59:00",
        "value": "0"
    },    ...
]
```

Heart rate

- Heart rate: a list with 1 data point every 5 seconds.

- Fields meaning:
  - time: timestamp (hh:mm:ss format) of the heart rate entry
  - value: the heart rate at the current time (in bpm)
  - confidence: a number from 0 to 3 defining how much the reading is reliable (this probably depends on user movement, etc. You can ignore it)

- **Note**: there can be missing values


```
[
  {
    "time": "03:10:19",
    "value": 58,
    "confidence": 1
  },
  {
    "time": "03:10:24",
    "value": 54,
    "confidence": 3
  },
  {
    "time": "03:10:29",
    "value": 52,
    "confidence": 3
  },
  {
    "time": "03:10:44",
    "value": 52,
    "confidence": 3
  },
  {
    "time": "03:10:54",
    "value": 53,
    "confidence": 3
  },
  {
    "time": "03:11:09",
    "value": 54,
    "confidence": 3
  },
  {
    "time": "03:11:19",
    "value": 53,
    "confidence": 3
  },
  ...
]
```

Resting heart rate

- Resting heart rate: an entry per day.


```
{
    "time": "00:00:00",
    "value": 51.35,
    "error": 17.86
}
```


- Fields meaning:
  - time: timestamp (hh:mm:ss format) of the resting heart rate entry (always 00:00:00)
  - value: the estimated resting heart rate at the current time (in bpm)
  - error: the estimate error (in %) (Can be ignored)

- **Note**: there can be days without data

Exercise

- Exercise: a list with 1 entry for each recorded exercise session

- Fields meaning:
  - activityName: the type of entry (“Corsa”: Run, “Bici”: bike, “Camminata”: Walk)
  - averageHeartRate: the average heart rate during the session
  - calories: the active calories spent during the session
  - distance: the distance covered during the session
  - distanceUnit: the unit of the “distance” field
  - duration: the total duration of the session (in ms)
  - activeDuration: the total active duration of the session (in ms)
  - steps: the total number of steps done during the session (if “Corsa” or ”Camminata”)
  - logType: how this record was logged (“tracker” or “auto_detected”)
  - heartRateZones: a list that summarizes the number of minutes and calories spent in specific heart rate ranges (from min to max)
  - speed: the average speed (in km/h)
  - vo2max: the estimated running VO2Max (only for “Corsa” activities)
  - elevationGain: the elevation gained during the session
  - time: the starting time of the session (format hh:mm:ss)

- **Note**: different activity types can have (slightly) different fields
- **Note**: there can be days without data


```
[
  {
    "activityName": "Corsa",
    "averageHeartRate": 143,
    "calories": 727,
    "distance": 8.474117,
    "distanceUnit": "Kilometer",
    "duration": 3.548E+6,
    "activeDuration": 3.214E+6,
    "steps": 8604,
    "logType": "tracker",
    "heartRateZones": [
        {
            "name": "Fuori zona",
            "min": 30,
            "max": 110,
            "minutes": 3,
            "caloriesOut": 30.146219999999971
        },
        {
            "name": "Grassi bruciati",
            "min": 110,
            "max": 137,
            "minutes": 8,
            "caloriesOut": 97.13782
        },
        {
            "name": "Attivita aerobica",
            "min": 137,
            "max": 170,
            "minutes": 48,
            "caloriesOut": 659.73843
        },
        {
            "name": "Picco",
            "min": 170,
            "max": 220,
            "minutes": 0,
            "caloriesOut": 0
        }
    ],
    "speed": 9.4981754822650917,
    "vo2Max": {
        "vo2Max": 48.77853
    },
    "elevationGain": 28.042,
    "time": "11:22:43"
  },
  {
    "activityName": "Bici",
    "averageHeartRate": 150,
    "calories": 3893,
    "distance": 140.143695,
    "distanceUnit": "Kilometer",
    "duration": 2.0901E+7,
    "activeDuration": 1.756E+7,
    "logType": "tracker",
    "heartRateZones": [
        {
            "name": "Fuori zona",
            "min": 30,
            "max": 110,
            "minutes": 28,
            "caloriesOut": 214.83527999999842
        },
        {
            "name": "Grassi bruciati",
            "min": 110,
            "max": 136,
            "minutes": 64,
            "caloriesOut": 664.55027999999993
        },
        {
            "name": "Attivita aerobica",
            "min": 136,
            "max": 170,
            "minutes": 232,
            "caloriesOut": 3167.1500100000012
        },
        {
            "name": "Picco",
            "min": 170,
            "max": 220,
            "minutes": 24,
            "caloriesOut": 377.76059999999995
        }
    ],
    "speed": 28.734978587699313,
    "elevationGain": 663.55,
    "time": "11:03:54"
  },...
]
```

Sleep

- Sleep: a list with 1 entry for each sleep session

- Fields meaning:
  - dateOfSleep: day associated to the sleep entry (MM-DD format)
  - startTime: the starting timestamp of the sleep (MM-DD hh:mm:ss format)
  - endTime: the ending timestamp of the sleep (MM-DD hh:mm:ss format)
  - duration: the duration of the sleep session (in ms)
  - minutesToFallAsleep: the number of minutes spent to fall asleep
  - minutesAsleep: the number of minutes asleep during the sleep entry
  - minutesAwake: the number of minutes awake during the sleep entry
  - minutesAfterWakeup: the number of minutes the user spent in bed after waking up
  - efficiency: the estimated sleep efficiency (from 0 to 100)
  - logType: how the entry was logged (always "auto_detected")
  - mainSleep: a boolean indicating if the entry corresponds to the main sleep session
  - levels: a summary of the sleeping stages (deep/wake/light/rem/restless) during the sleep session
  - data: the stage profile of the sleep session. Each entry contains the starting timestamp, the level, and how much it lasted.

- **Note**: there can be days without data


```
[
{
    "dateOfSleep": "02-14",
    "startTime": "02-13 22:44:00",
    "endTime": "02-14 06:36:30",
    "duration": 2.832E+7,
    "minutesToFallAsleep": 0,
    "minutesAsleep": 429,
    "minutesAwake": 43,
    "minutesAfterWakeup": 3,
    "timeInBed": 472,
    "efficiency": 96,
    "logType": "auto_detected",
    ”mainSleep": true,
    "levels": {
        "summary": {
            "deep": {
                "count": 5,
                "minutes": 106,
                "thirtyDayAvgMinutes": 86
            },
            "wake": {
                "count": 34,
                "minutes": 43,
                "thirtyDayAvgMinutes": 46
            },
            "light": {
                "count": 28,
                "minutes": 235,
                "thirtyDayAvgMinutes": 281
            },
            "rem": {
                "count": 11,
                "minutes": 88,
                "thirtyDayAvgMinutes": 101
            }
        },
        "data": [
            {
                "dateTime": "02-13 22:44:00",
                "level": "wake",
                "seconds": 30
            },
            {
                "dateTime": "02-13 22:44:30",
                "level": "light",
                "seconds": 150
            },
            {
                "dateTime": "02-13 22:47:00",
                "level": "deep",
                "seconds": 1860
            },
            ...
},
...
]
```