Description

Our project has been designed to provide a program which can can receive messages from a bluetooth sensor and transfer them to a cloud database anywhere on the internet.
The program innitiates connections to a remote database with a secure encrypted connection, and the database information itself can be implemented by any services that can connect to the database.
If the program cannot communicate with the server temporarily the program was also planned to archive any data it could not send to the server locally until a connection can be re-established.

Contributions:

Ms. Wang helped to design a translation method so that the data from the sensor could be understood by the program and be translated into human-readable code. Eleanor also helped provide documentation for the program to make it understandable and so that modifications can be designed for the program to compliment its open-source, 'libre' design.

Mr. Liyanage has provided the root program. This program implements threading so that each task can be managed independently of another. This program manages the data as it is used by other sections of the program as well as scheduling and prioritizing tasks.

Mr. Connell-Whitney designed the network coding for the program such that the program can communicate with the database using a secure encrypted connection. When the connection cannot be initiated or it stops functioning, the program will then send code to archive.

Mr. Manel designed an archive function for the program. The purpose of which was to provide a backup of all medical information that could not be sent to the intended server. This way any data that cannot be sent is preserved until a new connection can be made.

UML Document

The program is designed to give peace of mind to patients and allow medical professionals to provide the best care to their patients with the most accurate and timely information. The family is also encouraged to participate in the patient's health, through the use of personal interaction and web access to parts of the database.
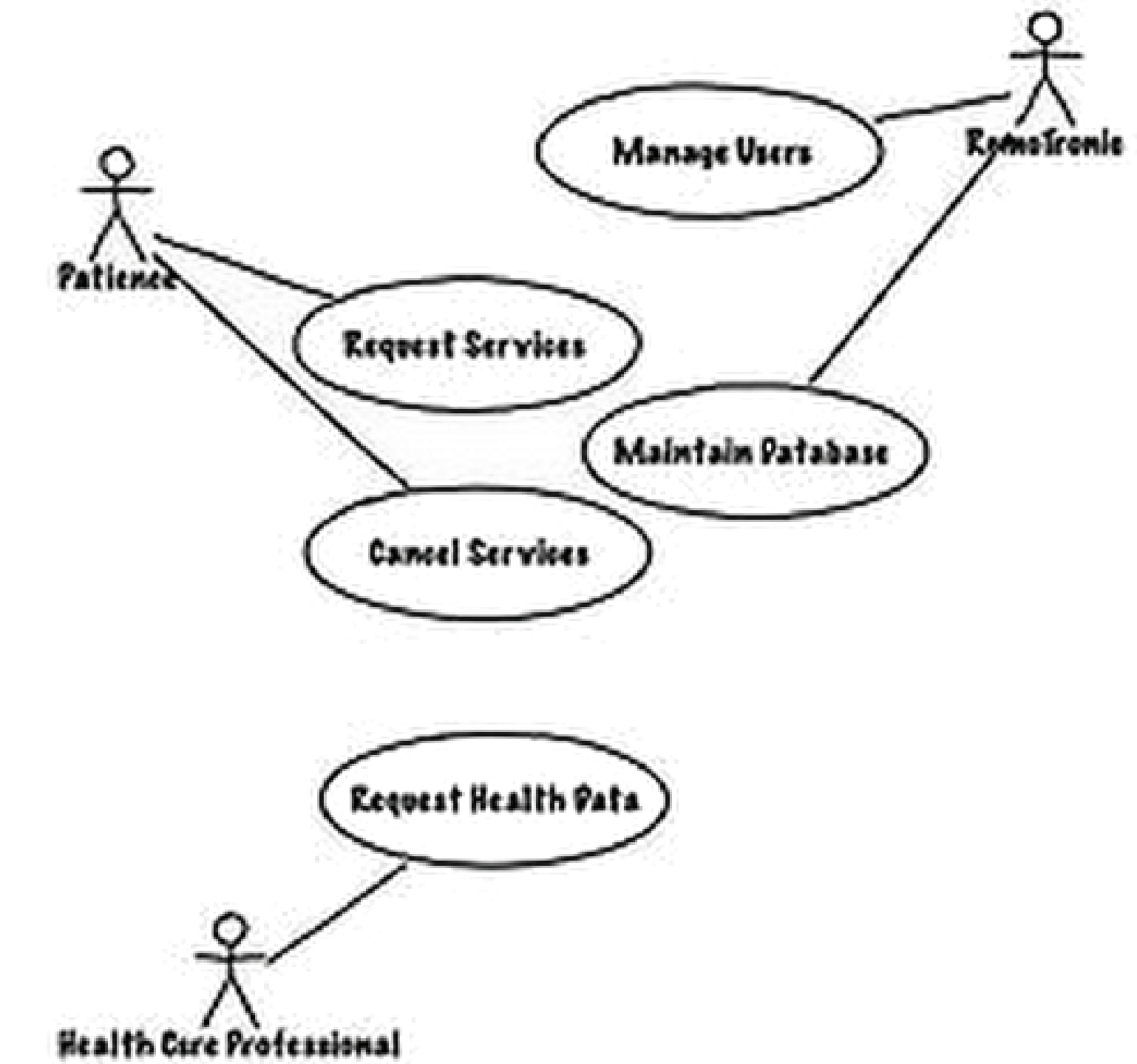
Sequence Diagram

The program is designed in such a way that a main thread is created which receives data from the sensor devices independently, processes this data so that it can be used more easily and then sends this data to a remote database.
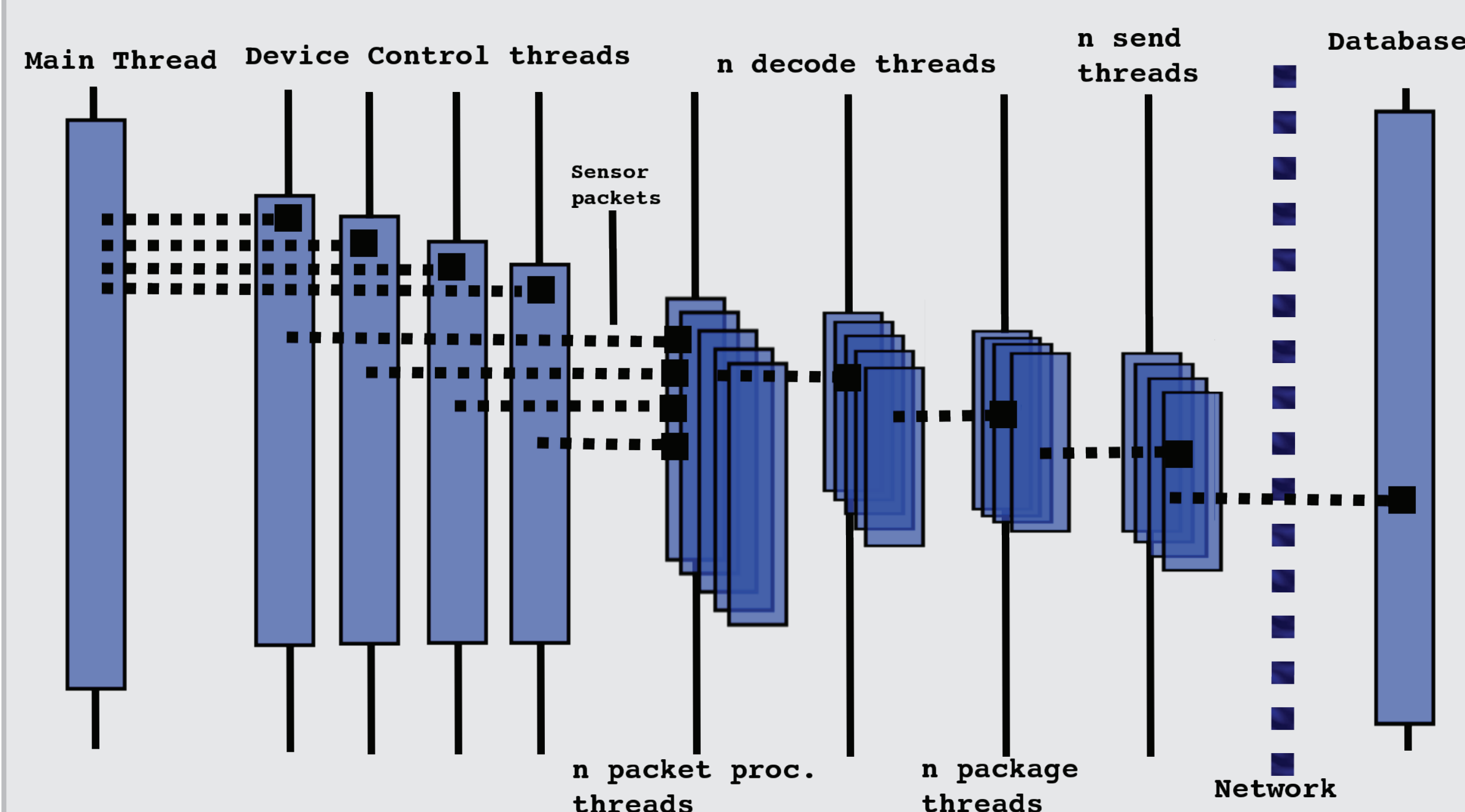
Database Schema

This program was designed:
- to use human readable and configurable files to modify settings such as the IP address of the database or to add and remove sensors
- With a modular library API to easily add a new library for a new device with minimal changes to the program
- With simple FIFO packet queuing for receiving packets with a configurable queue size
- To support up to 4 devices or more

patient_health
- id INT(11)
- timestamp TIMESTAMP
- hrt_rate INT(12)
- resp_rate INT(12)
- skin_temp CHAR(20)
- posture INT(12)
- dev_mac CHAR(30)
- est_core_temp INT(12)

Indexes

patient
- lastname VARCHAR(45)
- firstname VARCHAR(45)
- dev_mac CHAR(30)

Indexes

device_info
- id INT(11)
- timestamp TIMESTAMP
- batt_vol INT(12)
- zephyr_sys_chnl INT(12)
- dev_temp INT(12)
- link_qual INT(12)
- batt_lvl INT(12)
- batt_stat CHAR(40)
- dev_mac CHAR(30)
- dev_manufac_code CHAR(30)
- dev_type_code CHAR(30)
- status CHAR(30)

Indexes

events
- id INT(11)
- timestamp TIMESTAMP
- dev_mac CHAR(30)
- dash_detect CHAR(30)
- jmp_flight_time CHAR(30)
- jmp_peak_accel CHAR(30)

Indexes