



UNIVERSITÉ DE SHERBROOKE  
DÉPARTEMENT D'INFORMATIQUE

## IFT785 Project Report

# Annotation de données capteurs pour apprentissage machine

Report by:  
Hui Chen

Sherbrooke (Québec), Canada  
25 March 2022

---

**Contents**

<b>1</b>	<b>Annotated Application Overview</b>	<b>1</b>
1.1	Problem . . . . .	1
1.2	Objective . . . . .	1
<b>2</b>	<b>Architecture</b>	<b>2</b>
2.1	Front-end Part . . . . .	3
2.2	Back-end Part . . . . .	4
2.3	Other requirements . . . . .	6
<b>3</b>	<b>A simulated scenario of annotation application</b>	<b>7</b>
	<b>References</b>	<b>8</b>

**List of Figures**

1	Annotation Application Architecture . . . . .	2
2	Front-end UI . . . . .	3
3	Labeled raw sensor data . . . . .	6

**List of Tables**

1	Daily activity labels [1] . . . . .	4
2	The table of saving users annotation information . . . . .	5
3	The table of labeling raw sensor data . . . . .	5

## 1 Annotated Application Overview

This section describes the problem, objective and the architecture of the annotated application.

### 1.1 Problem

To promote home care, networks of sensors are deployed in the elderly's homes. The data collected makes it possible to analyze their activities of daily living(ADLs) (sleep, meals, hygiene, social life, etc.). Although the current methods give good results in Nears<sup>1</sup>, we would like to use machine learning to have better analyses. However, we have no information to establish the ground truth. It is also expensive to annotate the activities by researchers long-term in real smart homes.

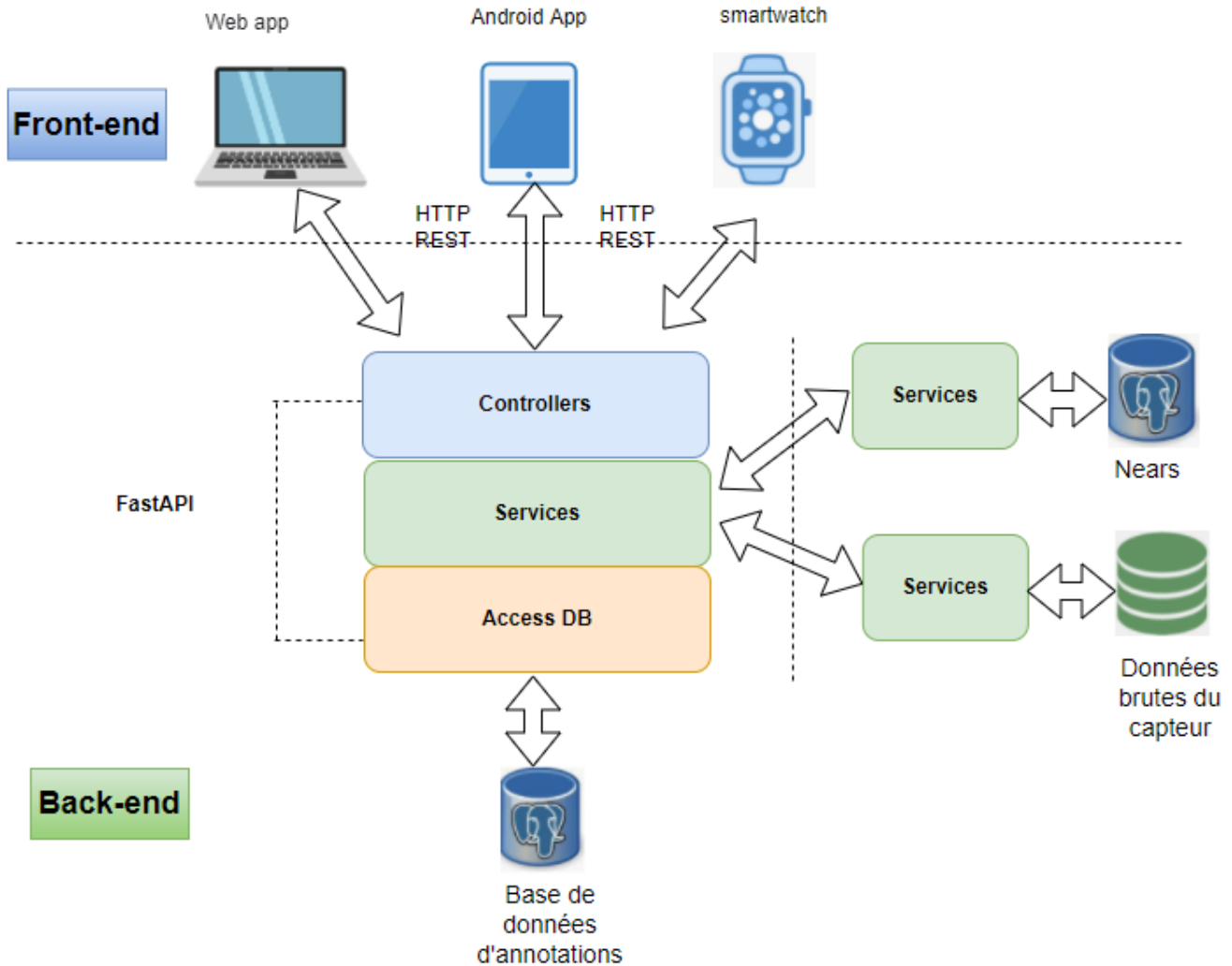
### 1.2 Objective

1. The annotation application would provide the older people with tools to give a chronology of their daily activities in terms of activities of daily living(ADLs).
2. The list of ADLs is predetermined, but it should be possible to modify or calibrate it.
3. The ADLs timeline will label the sensor events in the database.
4. Improve the quality of the labeling activities.

---

<sup>1</sup>DOMUS lab project in Université de Sherbrooke

## 2 Architecture



**Figure 1:** Annotation Application Architecture

The project includes front-end and back-end parts. The front end has three types: web application, Android application, and smartwatch. The back-end uses FastAPI and includes controllers, services, and access DB to communicate with the front-end and database.

## 2.1 Front-end Part

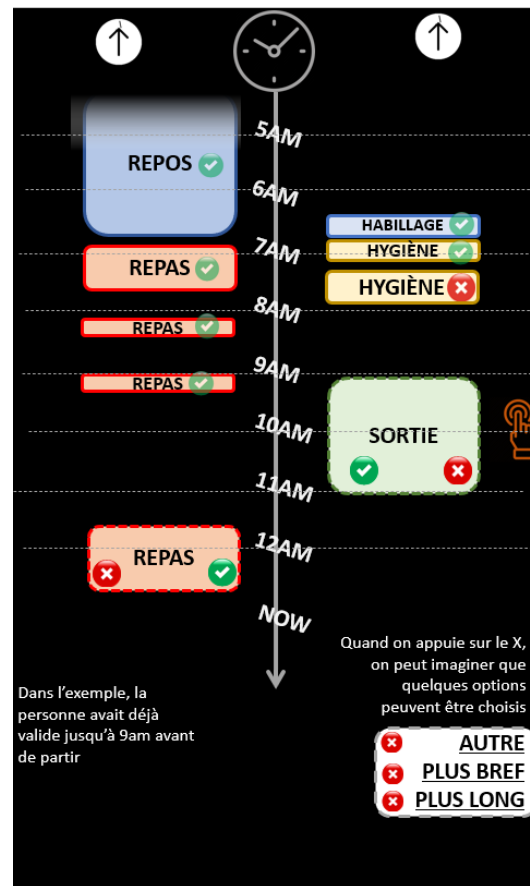


Figure 2: Front-end UI

The front-end provides the older people with tools to give a chronology of their day in terms of daily activities of daily living(ADLs). This part should include the following functionalities:

1. Users authenticate with Keycloak. <https://www.keycloak.org/>
2. Editable annotations — users can submit the results, make some changes afterward, and submit them again. No duplicated entries will be generated.
3. Each submission is stored in the database, with a timestamp field. Cache all the information and send it back to the server in real-time or every 2 minutes? (Decide by the team)
4. Display the timeline ADLs from Nears (Hygiène court, Hygiène long, Repas, repos, Inactivité).
5. User can add/delete more activities into the timeline. Table 1 show all the daily activities we need to annotate. The activities can be labeled as Bed\_Toilet\_Transition, Sleep, Wake\_Up, Personal\_Hygiene\_Long, Personal\_Hygiene\_Short, Morning\_Meds, Cook\_Breakfast, Eat\_Breakfast, Wash\_Breakfast\_Dishes, Watch\_TV, Leave\_Home, Enter\_Home, Noon\_Meds, Evening\_Meds, Cook\_Lunch, Eat\_Lunch, Wash\_Lunch\_Dishes, Nap, Cook\_Dinner, Eat\_Dinner, Wash\_Dinner\_Dishes, Relex, Sleep\_Out\_Of\_Bed, Cook\_Snack, Eat\_Snack, Unknown\_Activity [2].

ID	Activity
1	Long Hygiene
2	Short Hygiene
3	Bed to Toilet Transition
4	Cook (Cook Breakfast, Cook Lunch, Cook Dinner)
5	Eat (Eat Breakfast, Eat Lunch, Eat Dinner)
6	Wash Dishes
7	Relax (Watch TV)
8	Sleep (Go to Sleep, Sleep Out of Bed, Nap)
9	Leave Home
10	Enter Home
11	Other Activity
12	Visitors

**Table 1:** Daily activity labels [1]

6. The activity box can be move to right time.
7. The user can adjust the size according to the duration of time. The background could have a dotted line each hour shown in Figure 2.
8. For more problematic activities, additional questions may be posed. For example :« On n’a pas détecté de repas à la maison aujourd’hui, est-ce exact? » « Avez-vous fait une sieste ce matin entre X et Y » « Étiez vous bien absent lors de X? »
9. If the user clicks the "No" button, we can popup questions or other activities for him to choose.
10. The interface always shows the unconfirmed activities.
11. The user can select the date, which means he can modify previous day annotated ADLs.
12. During the test, we need to test if the web application works well using a cellphone or tablet.
13. The interface uses French.
14. Angular / React.

## 2.2 Back-end Part

We can use FastAPI<sup>2</sup> to develop the back-end part which includes controllers, services and database access.

### 1. Controllers:

- Transmit data between front-end(web application, Andriod application, smartwatch) and processing component.

<sup>2</sup><https://fastapi.tiangolo.com/tutorial/>



Name	Description
home	name of the house of the event
activity	the user perform the activity
start	the start time of the activity
end	the end time of the activity
time	the time of the user answer saving the questions

**Table 2:** The table of saving users annotation information

Name	Description
home	name of the house of the event
sendAt	the time of detecting the sensor event
sensor	the type of the sensor
subject	the specific place in
location	the room of the house
status	the value of the sensor event
label	the type of activity
labelStatus	the activity "begin" or "end"
time	the time of the user answer saving the questions

**Table 3:** The table of labeling raw sensor data

- It should be general and support other kinds of front-end.

## 2. Services:

- Manage the high level activities from NEARS and adjust the activities as a timeline for Front-end needs. The high level activities are in events.csv using "activity" filter in column "measurable".
- Manage timeline of ADLs from the users and save to Postgres database with schema in Table 2.
- Create a new table in Postgres to store low-level sensor events(from MongoDB) combined with timelines of ADLs. The schema of the new table can be in Table 3. Fig. ?? shows the details
- Use algorithm to decide popup which questions. How to handle conflict with users' timeline ADLs and Nears referenced activities. (Ex. Seniors confused their activities ).
- Use machine learning to predict activities and confirm with the users. Automatically pick a score based on the model prediction — the annotator only needs to adjust the model prediction. Also, it gives you an idea of how the model performs on unseen entries. (Can be negotiation).

```

2009-07-17 16:15:12.625001 OtherRoom ON
2009-07-17 16:15:16.906001 OtherRoom OFF
2009-07-17 16:15:17.203001 FrontDoor ON
2009-07-17 16:15:18.234001 FrontDoor OPEN Leave_Home begin
2009-07-17 16:15:18.937001 Kitchen ON
2009-07-17 16:15:21.531001 FrontDoor OFF
2009-07-17 16:15:22.843001 Kitchen OFF
2009-07-17 16:15:27.828001 FrontDoor ON
2009-07-17 16:15:31.796001 FrontDoor OFF
2009-07-17 16:15:34.421001 FrontDoor CLOSE Leave_Home end
2009-07-17 17:02:21.843001 FrontDoor OPEN Enter_Home begin
2009-07-17 17:02:24.812001 FrontDoor ON
2009-07-17 17:02:29.109001 FrontDoor OFF
2009-07-17 17:02:35.703001 FrontDoor ON
2009-07-17 17:02:37.031001 FrontDoor CLOSE Enter_Home end
2009-07-17 17:02:39.718001 FrontDoor OFF
2009-07-17 17:02:45.328001 Kitchen ON
2009-07-17 17:02:46.625001 Kitchen ON
2009-07-17 17:02:49.625001 Kitchen OFF
2009-07-17 17:02:50.937001 Kitchen OFF
2009-07-17 17:02:52.921001 Kitchen ON Cook begin
2009-07-17 17:02:56.828001 Kitchen OFF
2009-07-17 17:02:57.531001 Kitchen ON
2009-07-17 17:02:57.828001 KitchenDoor OPEN
2009-07-17 17:03:01.453001 Kitchen OFF
2009-07-17 17:03:10.734001 KitchenDoor CLOSE
2009-07-17 17:03:14.046001 Kitchen ON
2009-07-17 17:03:17.937001 Kitchen OFF
2009-07-17 17:03:44.046001 Kitchen ON
2009-07-17 17:03:48.015001 Kitchen OFF
2009-07-17 17:04:02.218001 Kitchen ON
2009-07-17 17:04:03.828001 Kitchen ON

```

**Figure 3:** Labeled raw sensor data

### 3. Database access:

- Transmit data between front-end(web application, Andriod application, smartwatch) and processing component.
- It should be general and support other kinds of front-end.

### 2.3 Other requirements

Front-end and back-end teams work together to achieve the following requirements:

#### 1. Real-time annotations :

- As soon as an activity is recognized, it is displayed, mechanisms can be added to ask immediately the user.
- The user can indicate when he will start an activity.

#### 2. A framework for exploring the annotated data:

- Displaying what is recognized and what is annotated.
- Rerunning recognition with a different algorithm with the possibility to explore various parameters.
- Displaying annotation, data, recognition for the ADLs.

### 3 A simulated scenario of annotation application

We use one scenario to show how the annotation application works. In the morning, the senior opened the annotation application for the first time. The application gets the previous week's routine of daily activities from Nears and shows the timeline for the user. The user can confirm the activity. If he answers no, popup questions to confirm if he does other activities. The user can choose other activities in the popup and adjust the time. The answers will be saved to the Postgres database. The application could use the answers of the activities to annotate raw sensor data (MongoDB) and insert this kind of information in a new table. If the user provides the conflict answers for the application, the algorithm will popup some questions to let the user confirm. In the afternoon, the user opens the application again. The interface always shows the unannotated timeline of the activities. The user also sees a popup question: "Do you start your snack?". The user will answer the question and choose the start time. So the application can achieve real-time annotations. The user remembers that he made mistake for yesterday's annotation. He can go back to modify his annotation for yesterday. The database will update the information accordingly. The user also can review what activities he has annotated and what not.

## References

- [1] Cook DJ and Krishnan Narayanan C. *Activity Learning*. Vol. 39. John Wiley Sons, Inc., 509.
- [2] Dr. Diane J. Cook. *AL-Smarthomet*. [Online; accessed 28-March-2022]. 2021. URL: [https :  
//github.com/WSU-CASAS/AL - Smarthome](https://github.com/WSU-CASAS/AL-Smarthome).