Hong Kong Institute of Vocational Education
(Tsing Yi)
Department of Information Technology
Higher Diploma in Software Engineering
Final Year Project (ITP4913M)
Final Report
(2020/2021)

Fall Detection

We declare that this is a group project and that no part of this submission has been copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part been written for us by another person.

Student	**Contribution to the project (%)** **(Total 100%)**

Abstract

Now cities are slowly beginning to become an aging society, so we need to use some innovative technologies to help them in their daily lives and one category is their health. According to past statistics, falls are most likely to affect elderly injuries, especially elderly people living alone. After they were injured, they were unable to contact their family members. It becomes one of the most serious problems because the elderly cannot be helped and treated immediately, it becomes even more serious. Fall detection is a major challenge in the public healthcare domain, especially for the elderly as the decline of their physical fitness, and timely and reliable surveillance is necessary to mitigate the negative effects of falls. So we chose to build a fall detection system. So that the elderly living alone can be protected. Our project is designed to support the independent living of the elderly. The main goal of the project is reconstructing the user's posture to detect falls in real-time.

Functionally, there will be a trained AI model which will detect the fall. The family can watch the life of the elderly through the live camera. The mobile app and website can view the records of the elderly who have fallen in the past. The family members will receive relevant notifications immediately after the elderly fall.

Table of contents

Introduction

Project Background

Nowadays the number of elderly people is increasing which brings us a continuously rising number of elderly living alone which may be very dangerous because of their lack of care. If the elderly get hurt or fainted, that will be very hard to discover. Concerning this kind of issue, we decided to develop a fall detection system to help surveillance the situation of the elderly.

Besides, we found that the technology of fall detection is very mature now. For example, the wearable devices (apple watch) that have been sold on the market. But these wearable devices have a problem, the elderly are more resistant to this type of device. The reason is that the elderly do not want something that does not belong to them to be required to wear.

Therefore, we need to think about the second method to detect falls. Due to the physical weakness and mobility difficulties of the elderly, they may easily get hurt. We expect the elderly people living alone who can't ask for help can receive medical care immediately when they fall or pass out by monitoring the human pose of the elderly people by the system in their living areas.

In conclusion, the method of using the camera is the most suitable. Because this will not affect the life of the elderly, only a camera needs to be installed at home for inspection, and the elderly will not feel uncomfortable. Besides, installation is very easy, without any tool assistance. Also, the biggest advantage is that it can be operated for a long time. Compared with traditional wearable devices such as the Apple Watch, it needs to be charged before it can run. But our method does not have to worry about that. The camera is connected to the computer to operate, and the long-term supply of power to the computer will proceed, it can be 24 hours.

The following are statistics about falls in the elderly:

Falls are one of the important causes of illness and death in the elderly.

Every year, an estimated 424,000 people die from falls worldwide, making it the second leading cause of non-temporary injury deaths after road traffic accidents. Although most falls are not fatal, there are approximately 37.3 million falls worldwide requiring treatment.

In Hong Kong, among elders aged 65 and above living in the community, approximately 1 in 5 people falls every year. Among those who fall, about 75% suffer injuries, including head trauma and fractures.

Gender

Male	125(66.8%)
Female	62 (33.2%)

Location occur

Home	83(44.4%)
Residential homes	27 (14.4%)
Trade and service	12 (6.4%)
Streets and highways	11 (5.9%)
Schools, other institutions, and public administrative places	7 (3.7%)
Industrial and construction sites	1 (0.5%)
Other locations	44 (23.5%)
Unspecified location	2 (1.1%)

This shows elderly people living alone at home will be the target of our project because they need special attention.

The Requirements

Scope of the problem

The scope of the problem is to outline the reasons for the need of having a fall-detection system for the elderly, point out other conditions that may have the elderly in danger, and state the weakness of existing simple fall-detection systems.

1. Potential hazard of falling of elderly

The elderly are inconvenienced by walking alone. When they are fallen, they gain greater damage than most young people and the harm of it may even be lethal in some situations. It is necessary to develop a system to detect if the elderly have fallen so that their family or the caretaker can take action to the harm on time.

2. Notification to elderly's family and caretakers, and alarm and help call 999

Only knowing the fallen of the elderly is meaningless. The elderly's family and the caretakers need to be informed immediately when the incident happens to let them take any action against the fallen one on time. It is necessary to include a real-time working feature to notice them when the elderly have fallen into the system. Since falling can be lethal to the elderly, it is necessary to help call the ambulance for the elderly too.

3. False-positive and False-negative on fall detection

Some simple fall-detection software may have the issue of false-positive and false-negative result since in some direction, the pose of the fallen elderly may be similar to standing to the system through the camera; or false-positive result since which the elder just want to take a break by lying down but the system recognizes it as fallen. It will cause countless false reports to the users and waste or have them miss the right time to rescue the fallen elderly. We should try our best to reduce the chance of having this kind of result.

4. False-positive and False-negative on fall detection

Some simple fall-detection software may have the issue of false-positive and false-negative result due to the fact that in some direction, the pose of the fallen elderly may be similar to standing to the system through the camera; or false-positive result since which the elder just want to take a break by lying down but the system recognizes it as fallen. It will cause countless false reports to the users and waste or have them miss the right time to rescue the fallen elderly. We should try our best to reduce the chance of having this kind of result.

5. Other situations leading elderly to dangerous

The signs of the elderly in danger are not only just falling, but also some other situations, like lying but not moving a muscle for too long. Those things happen mostly when the elderly have some health issues. There is a need to survey the status of the elderly.

6. Description of the problem environment

The problem environment mostly is in-door like an elderly's home or the nursing home for elderly, etc. Most of the nursing homes for the elderly provide a friendly environment which likes handles for assisting elderly walking and less furniture putting on the hallway to prevent the elderly from being stripped down, and there is often caretakers standby to help them. However, not all of the environment of the home of the elderly are friendly to themselves, and their families are not always with them. Thus, the need for the system is very high in this kind of environment.

Scope of the proposed system

The scope of the proposed system is to outline the system for the need of having a fall-detection system for the elderly, point out relevant solutions and suggestions for the scope of the problem.

1. Building fall detection system with the assistance of libraries

A Library is a set of programs already implemented by a 3-rd party, and well-functioning in some specific field. Building the system with libraries can save a lot of time on constructing algorithms to recognize people and their body positions. An open-source library “OpenPose '' as an example, this library can recognize the human body, record their coordinate in the view of the camera and sketch the skeleton of the human body. We can apply those data to the fall-detection system so that we can save the time of construction of collecting the data from raw materials.

2. Streaming of the web-camera

The system will stream the view of the web camera used to have surveillance on the elderly. This feature helps users understand the situation of the elderly besides reading notifications. Also, if there is any false report of the elderly in danger, the user can judge the situation by themselves through the stream.

3. Mobile and Website version of the application

People nowadays are not always sitting in front of the computer, but always have a smartphone in the pocket. To make sure the system can notify the user immediately when the elderly are in a not safe situation, we should have a mobile version of the application so that we can notify the user by the alarm function of the smartphone. Moreover, having a mobile version means that users can surveillance the elderly any time they want with their smartphone, so that they may have the chance to notice something which may cause danger to the elderly and that cannot be detected by the system.

4. Mobile and Website version of the application

People nowadays are not always sitting in front of the computer, but always have a smartphone in the pocket. To make sure the system can notify the user immediately when the elderly are in a not safe situation, we should have a mobile version of the application so that we can notify the user by the alarm function of the smartphone. Moreover, having a mobile version means that users can surveillance the elderly any time they want with their smartphone, so that they may have the chance to notice something which may cause danger to the elderly and that cannot be detected by the system.

5. Information notification after fall

After the elderly fall, automated notification processing is required. Because the elderly may need help, not only do they have to notify the family, they may also need to call 999. After using the system to process data in real-time, the important information is transmitted to people who are accustomed to it through the Internet. The data should include GIF images, live stream URLs, and text messages.

Functional Requirements

Here is the outline:

- Automatic fall-detection
- Streaming of the view of the web-camera
- Automatic notification and emergency calls when falls are detected
- Register, login, and logout function on the website and mobile app to indicate which of the elderly information belongs to who.
- Automatic reminder to family and caretaker by message if any unusual things happened to the elderly

1. Fall detect system

The family members can use this system to care about the elderly who live alone. We will train an AI model, and then cooperate with the XY coordinate checked by OPENPOSE to determine whether the elderly is falling. This is our main part to detect whether the elderly have fallen. We use the camera to capture every frame, and then pass the video through machine learning and artificial intelligence to complete this process. So we will use openpose as the basis, check the human skeleton and build the skeleton, then go through a model that has been trained to determine whether it has fallen.

2. Database system

We will choose to use MY SQL DB to store data, including elderly's self data, fall records, passwords, etc. We are using many different programming languages to support our system. The main one is the database, website, and fall detection system are our core values. So reliable and stable software is our first choice. Considering feasibility, MS SQL Server is the most suitable because we have a large number of transactions. The transaction process is related to other data. Using a Relational Database can improve operational efficiency and SQL Server also provides better security. Moreover, our system programming language is Python and PHP. SQL Server has better compatibility and a more convenient connection with Python and PHP. For easy to use and flexible. We select SQL Server.

3. Saving GIF image system

The system will record the elderly from losing their balance to being on the ground. The processed GIF image will be stored in the server, and the URL of the server stored GIF image will be saved in the DB. The idea is the system will capture 20 frames before the fall and 30 frames after the fall. So there are 50 frames in total. Finally, because the uncompressed GIF image is too large, there is no way to provide Gmail for transmission to the user. Because this will exceed the size of attachments that Gmail can transfer. Therefore, it is necessary to compress and reduce the GIF Image size first. After the compression is complete, it will be stored in a file on the Apache server.

4. Sending email system

After fall, we will use the API method to send GIF images and live stream URLs to Gmail to transmit the data to the family of the elderly. A library called “smtplib” can support this function. First, the system will select the corresponding email address of the elderly's family member in DB. And the corresponding GIF Image file in the Apache server will also be selected. The last item is the URL of the live stream, it should be (http://xxx.xxx.xxx.xxx:9000/video_feed). The xxx.xxx.xxx.xxx is the IP of the connected network of fall detection systems.

5. Sending WhatsApp message system

Considering that family members do not check email for a long time, they will first receive a WhatsApp message to notify them to check email. We will call the API method to send the message. The API is “Twilio” a communication API platform. Give priority to using its web version to set up the basic requirements and prepare the phone number to be used. Then generate your account API key and use Python for coding.

6. Live stream system

Live broadcast for family members to check the elderly after a fall. Judge the injury situation of the elderly. We use flask this set of libraries for coding. Because this is a web application framework written in Python. We require the entire system to use python, so it is very suitable. The method of operation is to keep refreshing the website and print frames one by one.

7. Register system

To confirm the identity of the user: If the user falls, the system can use the account information to call the hospital. Here is the data collection:

- Name
- HKID
- Email
- Phone
- Gender
- Address

8. Login system

Concerning security, we have two types of users. Every user must use this function to log in. family members or social workers need to enter their email and password to log in.

9. Logout system

The logout function is suitable for users who have logged in to the website and ended using the website.

10. View fall detect record system

This system is for saving some gif files that when the elderly are falling, their family can use this system to review what is happening.

11. Call help system

After fall, the caretaker will receive a GIF image and personal information of the elderly on the website page. So caretakers can use the "call help" button to call 999.

12. Mobile App

This app provides the most important function for relatives or caretakers to use. Also, this app will let the user easier to use our program and be more convenient.

Non-functional Requirements

Operational Requirements

- Technical Environment Requirements
- The fall detects system will use python as the main language.
- Provide a stable version of the system to our user

- System Integration Requirements
- The system can use a calling help system to contact their family members.

- Portability Requirements
- The system must be able to run with Windows
- The system can run on Android
- The system needs to operate at phone and PC

- Maintainability Requirements
- New versions of the system will be provided every six months

Performance Requirements

- Speed Requirements
- Response time must be 4s or less for any transaction over the network
- The data of the database must be updated and get for the real-time

- Capacity Requirements
- The user data max can store 50,000 customers for a total of about 2 GB
- All the transmit pack size require 300K of data

- Availability and Reliability Requirements
- The system should use 1 day each month except for scheduled maintenance
- The system will have over 90% uptime performance

Security Requirements

- Encryption and Authentication Requirements
- The login system of the password will be encrypted
- Data will be encrypted on the website to provide a secure look
- The account will be locked while operating in the system or maliciously illegal operations.

Cultural and Political Requirements

- Multilingual Requirements
- The system will operate in English

Architecture of the proposed system

Hardware

1. 2D camera

The 2D camera is used for Openpose to build the human skeleton and detect fall.

2. Computer

The computer is used to run our program.

3. Database Server

The database server is used to save the user information like user name, HKID, email, password, phone, gender, and address.

4. Web Server

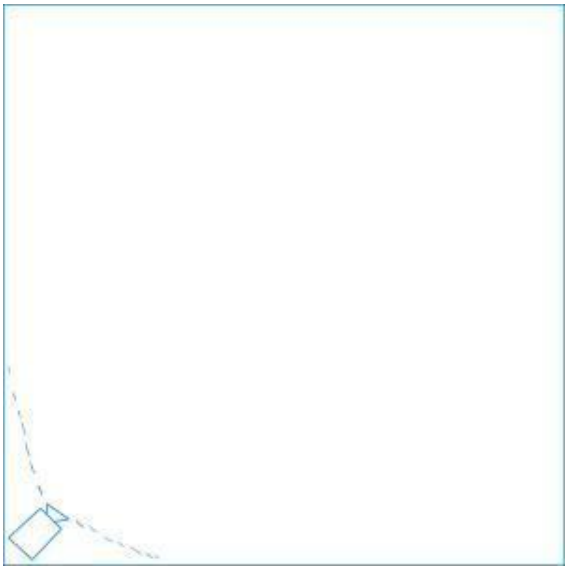
Web server is used to run the PHP website and save the GIF image in the server.

- **Server**

Database Server	MY SQL
Web Server	Apache

We choose MY SQL + Apache as a combination. It is because we need to run the PHP code and these two combinations are the easiest to install because they are free. Moreover, in our case, we use only one computer run as a Database Server, Web Server, and Fall detection system (python code), so that the XMPP can support MY SQL + Apache.

- **2D Camera Environment**



Here is the 2D camera setting:

The view is a top view. The camera will set in the corner of the room. That's why the camera will clearly find the person in the room. Also, the camera angle is 30 degrees downwards, so that can build a complete skeleton.

- **Network Environment (TCP/UDP port)**

We need to use SMTP port (587), Stream Live port (9000), HTTP port (80), and MY SQL port (3306) in our project.

1. SMTP port (587)

This port is for sending Gmail to the user. This port is designed to send email through a network and to its recipient. This is the default mail submission port. When an email client or outgoing server is submitting an email to be routed by a proper mail server, it should always use SMTP port 587 as the default port. This port, coupled with TLS encryption, will ensure that email is submitted securely and follows the guidelines set out by the IETF.

2. HTTP port (80)

This port is for using the website so that the user can access the web page. Port 80 is open for HTTP (Hypertext Transport Protocol), which is the most used protocol for the Internet. And is mainly used to transmit information on the WWW (World Wide Web) service.

3. MY SQL port (3306)

This port is for connecting to the MYSQL database. Port 3306 is the default port used for the MySQL protocol. You will use it to connect MySQL clients and utilities.

4. Stream Live port (9000)

This port is for using access to the live stream web page. Open the web access.

Software

Main composition

	SYSTEM	NOTE
Programming Language	Python	Python is an interpreted, high-level, and general-purpose programming language. Python is used to develop the fall detect system, view fall detect records, calling the help system.
Library	Openpose	OpenPose represents the first real-time multi-person system to jointly detect the human body, hand, facial, and foot key points (in total 135 keypoints) on single images. OpenPose is used to build the human skeleton.
Website	HTML5, PHP	HTML5 is a markup language used for structuring and presenting content on the World Wide Web. PHP is a general-purpose scripting language especially suited to web development. HTML5 and PHP are used to develop the website, link the database, and much more.
Database	My SQL	MySQL is an open-source relational database management system. The abbreviation for Structured Query Language.
Live Stream	Flask	Flask is a lightweight web application framework written in Python. Based on Werkzeug WSGI toolbox and Jinja2 template engine. Flask uses BSD authorization.

Addition composition

	SYSTEM	NOTE
Library	Keras	Use for the fall detection system
	Tensorflow	
	Tensorflow-GPU	
	Pandas	
	Numpy	
	Matplotlib	
	OpenCV-python	
	Pillow	
	Twilio	
	Pymysql	
	Flask	
	Imageio	
Website	Javascript	Use for the website page
	HTML5	
	PHP	
	Jquery	
	CSS	

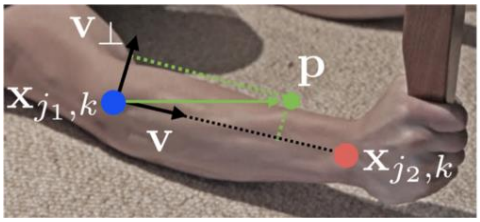
Mobile App	Java	Use for the mobile app
	Android studio	

Openpose Algorithm

The software part of the system includes some of the existing 3-rd party libraries and the main program developed by us. For the hardware part, the web camera will be needed to support the detection and streaming part. A computer with high graphic processing ability to train the model used to detect falling. There will be a database to store any required data, which includes, but may not be limited to, user accounts, elderly profiles, and data about the relations. The fall-detection system relies on the skeleton sketch of the human body in the view of the camera. Fall is detected when key points (mostly the joint part), like the hip point, of the skeleton, reach critical speed and have hit points.

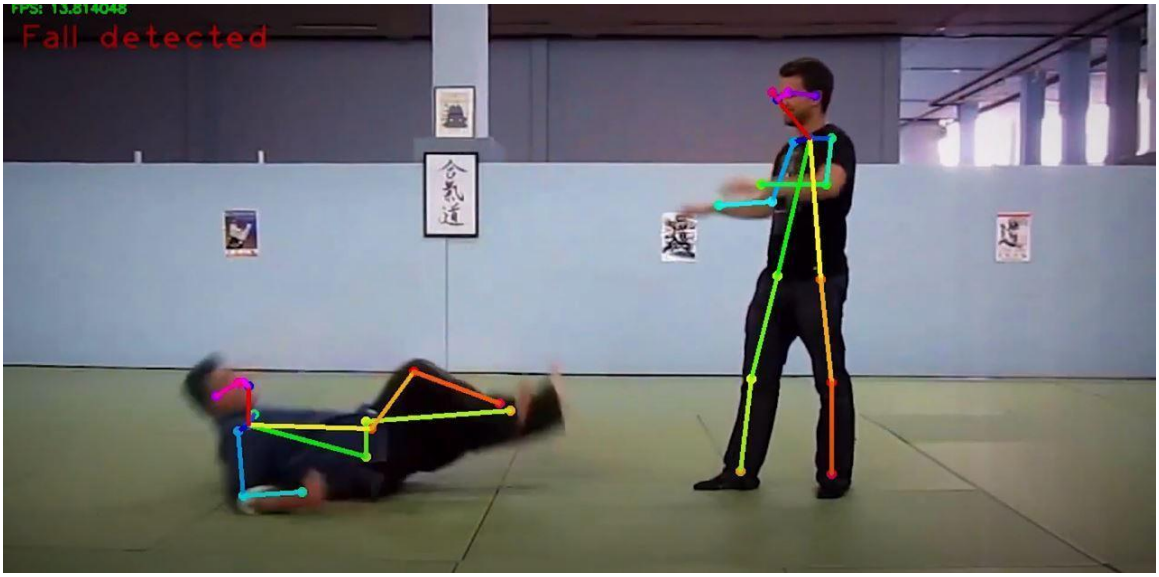
*The system sketches the skeleton of the human body

Input the picture, then the model predicts the Confidence Maps of the joint points and the PAF vector at the same time, and then post-processes the prediction of the second step, and finally obtains the overall posture.



$$\mathbf{L}^*_{c,k}(\mathbf{p}) = \begin{cases} \mathbf{v} & \text{if } \mathbf{p} \text{ on limb } c,k \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

*Part Affinity Fields for Part Association (PAF)



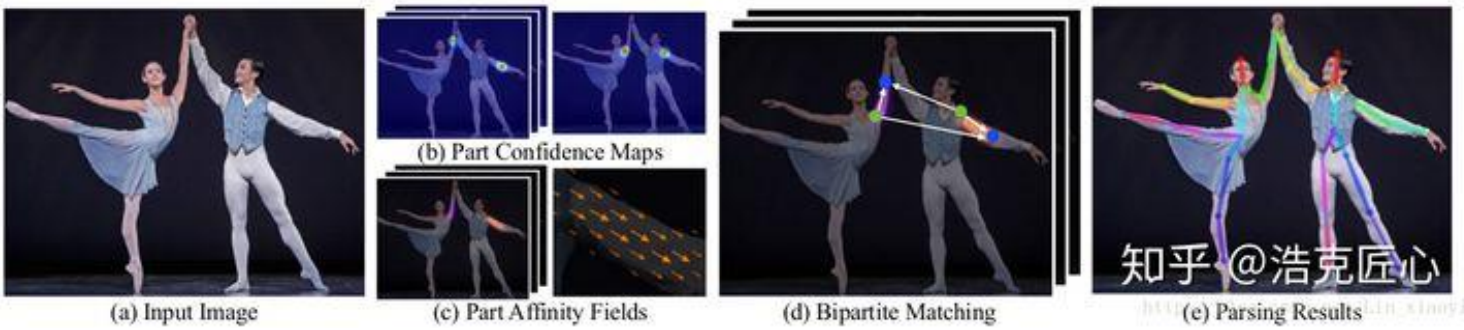
*When the person falls, there are hit points on the arms and the hip. Thus fall detected.

The flow of the algorithm

1. First, perform regression on all the people appearing in the image, and return to the points of each person's joints.

2. Use CNN network to extract Part Confidence Map (CMP) and Part Affinity Field (PAF).

3. Finally, the final result is obtained by repeatedly refining the predicted heatmap.



Traditional Human Pose

Human posture detection is usually a top-down idea. First detect whether the person exists, and then segment each person out. Based on each person, find out the key points of their respective human bodies.

Problems:

1. The result is heavily dependent on the result of the first step of human detection. If no one is found, there will be no way to find the key points.

2. The calculation time is positively correlated with the number of people, the more people, the more time-consuming.

Openpose Human Pose

OpenPose uses another way of thinking, that is bottom-up. First, find out all the key points in the picture, and then use the matching method to assemble it into a personal skeleton.

To implement this thinking, it uses PAF (Part Affinity Fields). It is responsible for encoding the 2D vector of the position and direction of the limbs in the image domain. At the same time, use CMP (Part Detection Confidence Maps) to mark the confidence of each key point (heat map). Through the two branches, a joint study of the key points and the relationship between them.

Problems:

1. The amount of calculation is very large. Based on Cuda acceleration, it is very expensive for GPU memory.

2. Low image resolution, motion blur, low brightness, dense detection targets, serious occlusion, incomplete targets, etc. The effect is not very satisfactory.

3. The original code by C++.

tf-pose-estimation

The biggest problem of openpose is it uses C++ to run and relies heavily on GPU performance. So we are looking for a library-related to openpose, that is use Tensorflow version Openpose. The 'Openpose' for human pose estimation has been implemented using Tensorflow. It also provides several variants that have made some changes to the network structure for real-time processing on the CPU or low-power embedded devices.

Tensorflow Graph File

The tf-pose-estimation has provided 2 type of Graph Model and different network size:

1. CMU (432x368, 656x368, 1312x736)
2. mobilenet_thin (432x368, 656x368, 1312x736)

The CMU model is closer to the original Openpose version, but a larger GPU memory is required. Mobilenet_thin is a model that can run under CUP or GPU and can have reliable FPS speed without the help of continuous GUP.

Conclusion

We have already tested CMU and mobilenet_thin with different network sizes. Use CMU model network size (432x368) on a graph card NVIDIA GEFORCE GTX 1060, the result is only 3-4 FPS. Therefore we are giving up on using the CMU model as our graph model. So the following is our final setting:

Molde: mobilenet_thin

Network size: 656x368

Platform: Python 3.7 + tensorflow-gpu 1.15.0

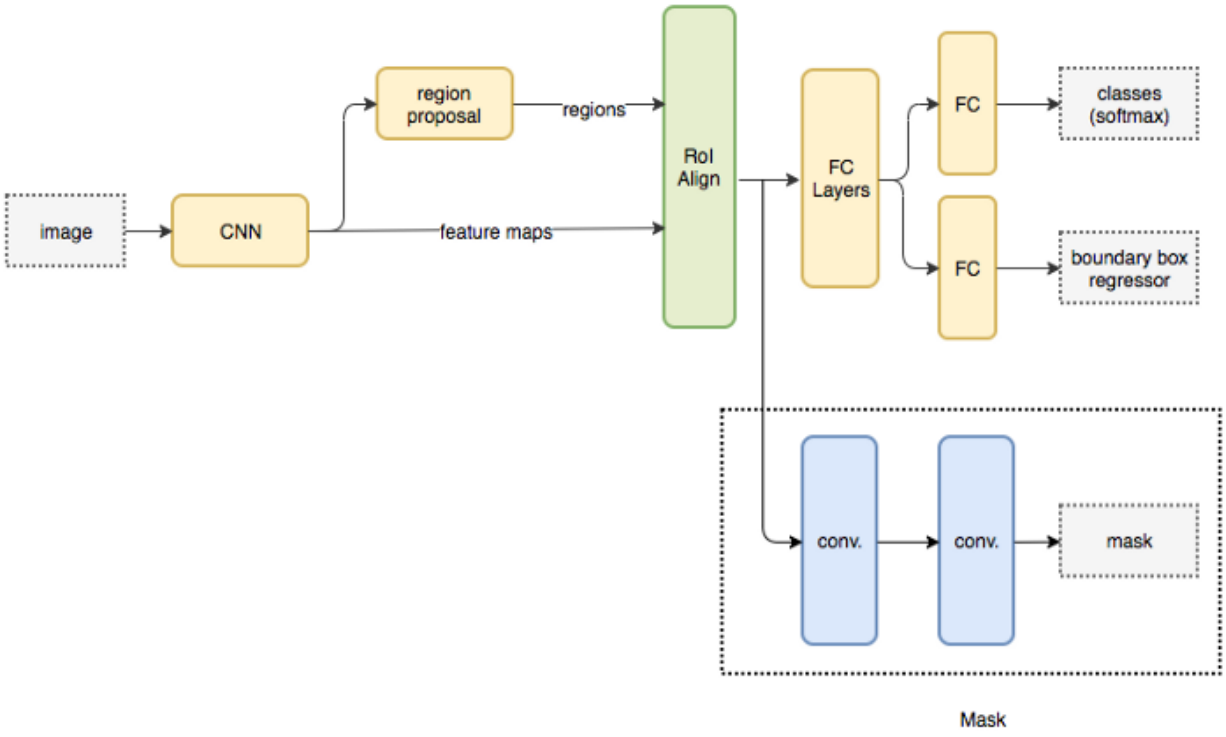
This setting runs on a graph card NVIDIA GEFORCE GTX 1060, the result is 9-10 FPS.

Model

The following models that need to be used

1. CNN

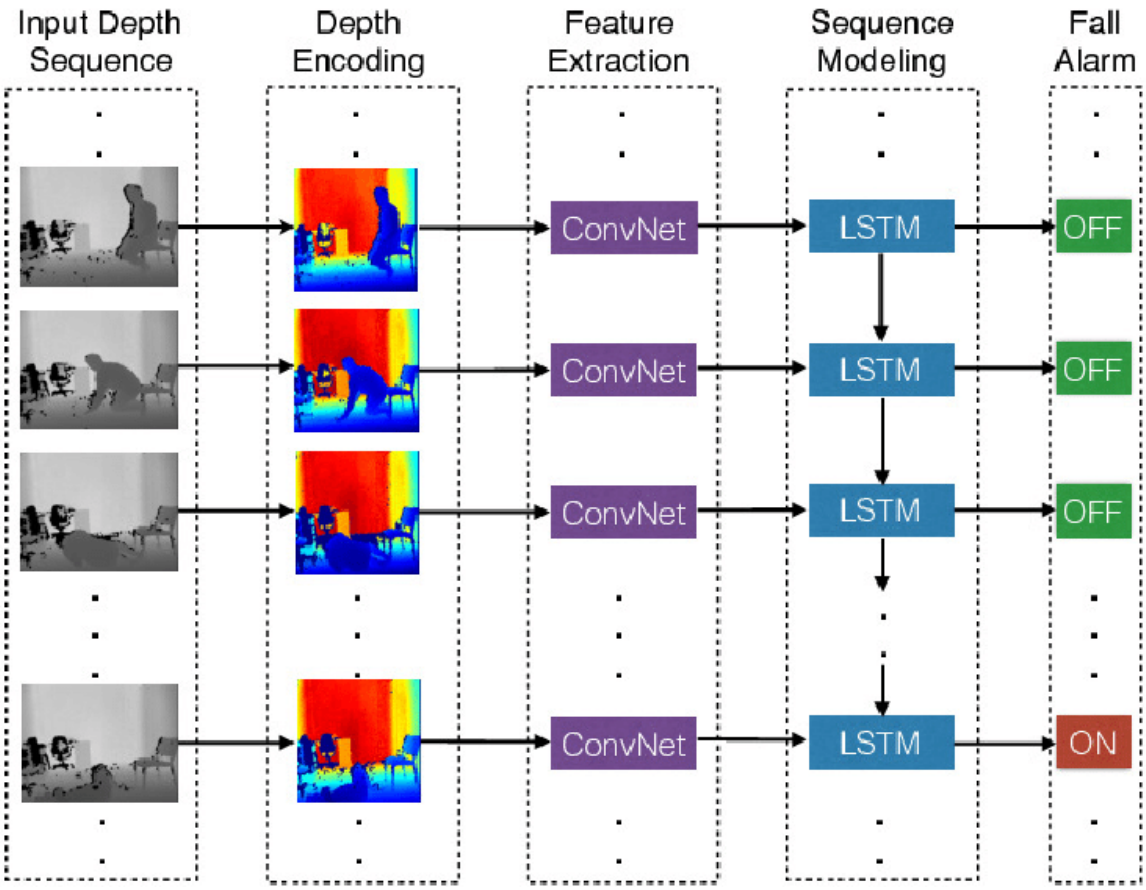
A convolutional neural network (CNN, or ConvNet) is a class of deep neural networks. CNN uses regularized versions of multilayer perceptrons. Multilayer perceptron usually represents a fully connected network, where each neuron in one layer is connected to all neurons in the next layer. So, the openpose library is run by CNN model for detecting human skeletons, because CNN is very suitable in the process of processing images. The following image illustrating the openpose library algorithms:



The openpose has been implemented by the CNN model. CNN extract Part Confidence Map (CMP) and Part Affinity Field (PAF) synthesize a skeleton. So we only need to use the CNN model to get the key points.

2. LSTM

The LSTM network is very suitable for classifying, processing, and making predictions based on time series data, because there may be lags of unknown duration between important events in the time series. Therefore, LSTM can solve the problem of vanishing gradients that may be encountered when training traditional RNNs. The relative insensitivity to gap length is the advantage of LSTM over RNN, hidden Markov model, and other sequence learning methods. So, we expect that LSTM will be used as our fall detection model.



The LSTM model is our part needed to train. Because the action of fall is a time element, the LSTM model would be an ideal model for the project. The following will describe more on how to train LSTM.

Training Method

First, we need to have a dataset of multiple sample videos. Therefore, we record video at IVE. The videos are recorded according to different light, angles, positions, and movements. There are a total of 135 videos, one action one video, and an average of about 5 seconds for each video.

Number of video

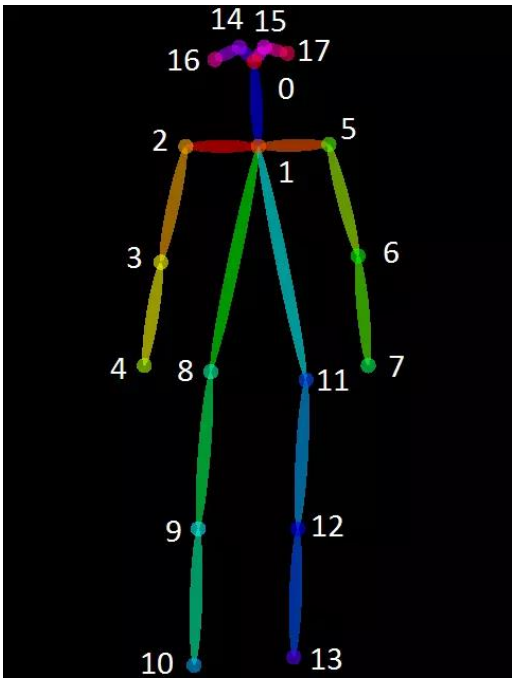
Fall Video	68
Not Fall Video	67
Total	135

Next, we need to split each frame of the video. Because in the real-time to run detection the system only can have 9-10 FPS. So we need to train the LSTM closer to real-time.

Split frame

Video	10 FPS/second

Then, we run the openpose CNN model to get the key points. The following is described of key points:



JSON Format:


```
{
   "version":1.1,
   "people":[
      {

"pose_keypoints_2d":[582.349,507.866,0.845918,746.975,631.307,0.58700
7,...],
         "face_keypoints_2d":[],
         "hand_left_keypoints_2d":[],
         "hand_right_keypoints_2d":[],
 "pose_keypoints_3d":[],
         "face_keypoints_3d":[],
         "hand_left_keypoints_3d":[],
         "hand_right_keypoints_3d":[]
      }
   ]
}
```


After getting the key points, it outputs the JSON file for each frame, so the total JSON file is:
Total JSON File = (total frame of each video) * (number of video)

The JSON data is about a total of 18 key points. For each key point include x coordinates, y coordinates, and probability score.

"pose_keypoints_2d":[582.349,507.866,0.845918,746.975,631.307,0.587007,...]

x coordinates: 582.348
y coordinates: 507.866
probability score: 0.845918

However, some frames are invalid data, for example, humans cannot be detected. The key point returns to 0, and the data cannot be trained. We need to delete this data. Similarly, on these 18 key points, the data from key points of ears and eyes are unnecessary. So, it also needs to be deleted.

Removed data:
1. the frame cannot detect people
2. left ear x coordinates (key point 17)
3. left ear y coordinates (key point 17)
4. left ear probability score (key point 17)
5. right eye x coordinates (key point 16)
6. right eye y coordinates (key point 16)
7. right eye probability score (key point 16)
8. left eye x coordinates (key point 15)
9. left eye y coordinates (key point 15)
10. left eye probability score (key point 15)
11. right eye x coordinates (key point 14)
12. right eye y coordinates (key point 14)
13. right eye probability score (key point 14)

Total training data of each JSON = (18 - 4) * 3 = 42

After our calculation we have

Row of data: 6435
Column of data: 42

Save all data as CSV file

The next step is labeling the data for each frame

Fall Label As	1
Not Fall Labe As	0

The last step is to normalize data. Because taking into account that it is performed on different webcams, the size of the image obtained is different. In this way, their XY coordinates will be different, so they need to be unified. One of the methods is to calculate the XY coordinates division by image height and weight. In our case for each key points, x and y coordinates are division by:

Height	y coordinates / 720
Weight	x coordinates / 1280

So all the key points data has been normalized XY coordinates between 0 - 1.

This is the training method and processed data to read for training.

Build the LSTM Network

To build LSTM, use the Keras library to complete. Because this method is the easiest.

Structure

1. set the batch size 16
2. one input layer (input shape: 1, 42); return_sequences = true
3. one hidden layer (shape: 8, ?); return_sequences = true
4. one output layer (shape: 1); return_sequences = false

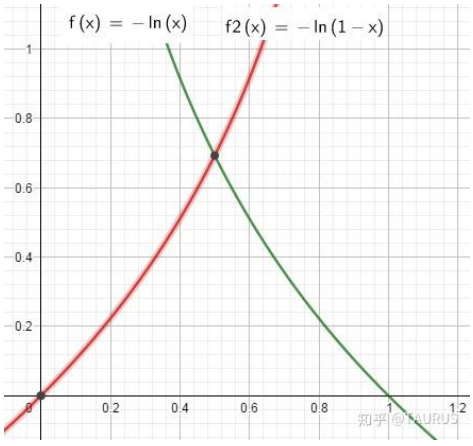
Optimizer Function

We are selected rmsprop. In order to further optimize the problem of excessive swing amplitude in the update of the loss function, and to further accelerate the convergence speed of the function, the RMSProp algorithm uses the differential square weighted average for the gradient of the weight W and the bias b.

$$
\begin{aligned}
s_{dw} &= \beta s_{dw} + (1-\beta) dW^2 \\
s_{db} &= \beta s_{db} + (1-\beta) db^2 \\
W &= W - \alpha \frac{dW}{\sqrt{s_{dw}} + \varepsilon} \\
b &= b - \alpha \frac{db}{\sqrt{s_{db}} + \varepsilon}
\end{aligned}
$$

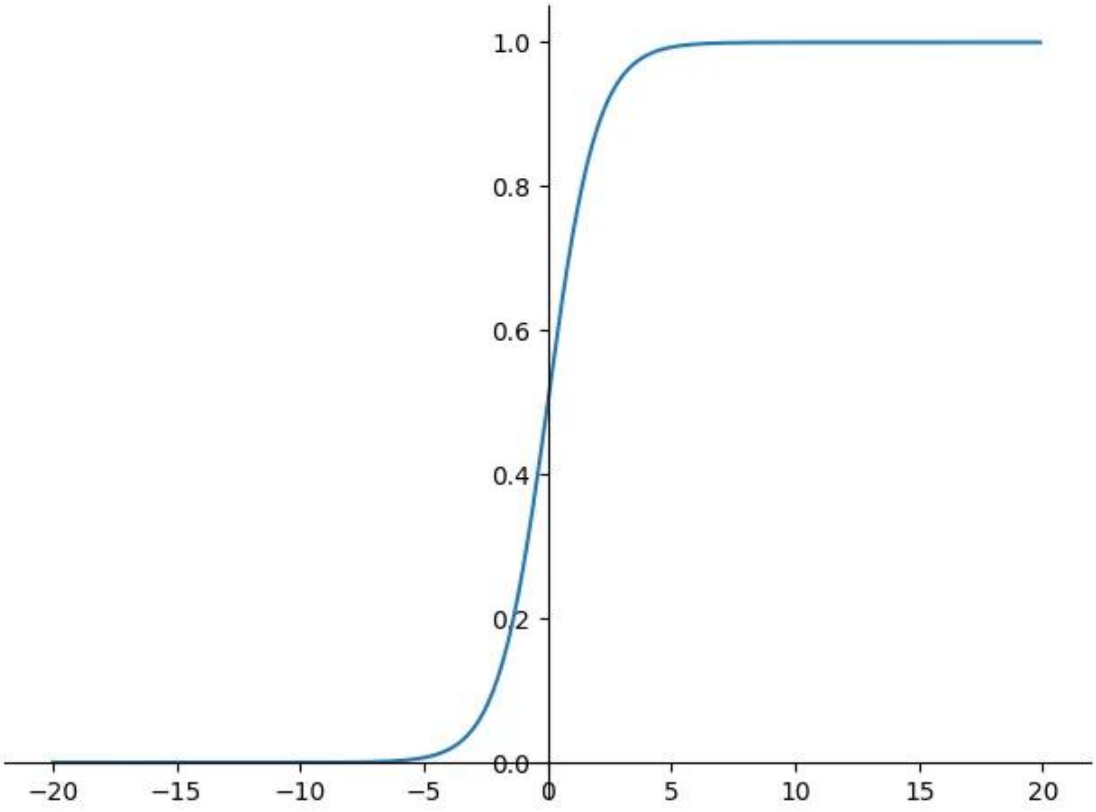
Loss Function

We are selected binary_crossentropy. In order to be used for two categories fall or not fall (1 or 0).



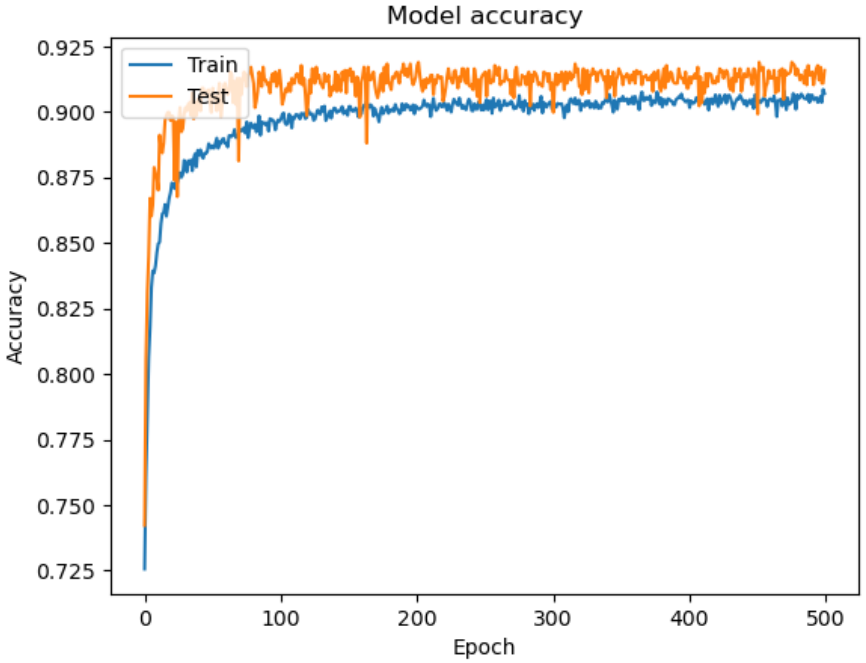
Activation Function

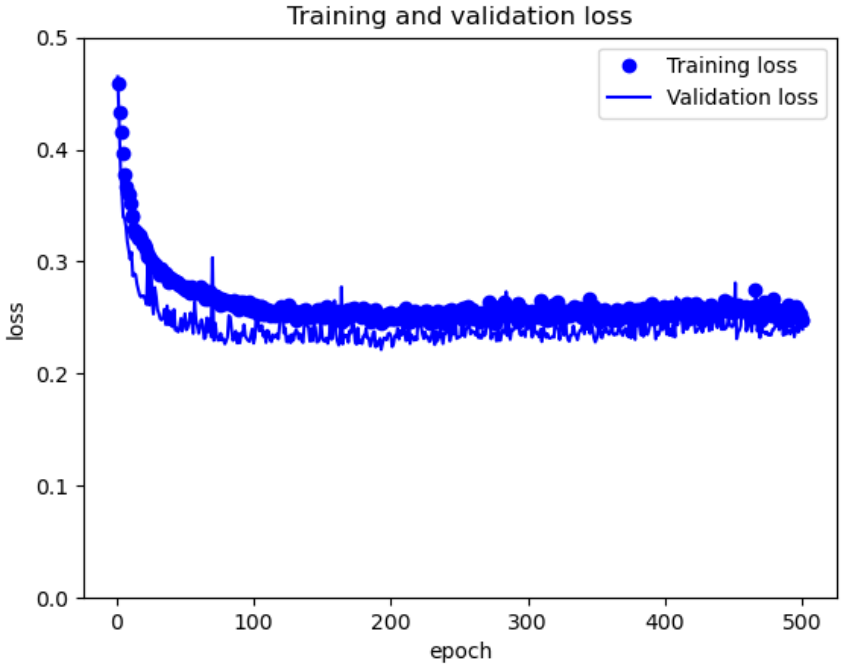
We selected sigmoid. Map any variables (these are first written as x) to between [0, 1]. It is usually used as the activation function of the neural network in the machine learning field.



Result:

Here is the training result:





System requirements

In order to use the system's software resources efficiently, the following are the requirements for users to use the system. We divide it into two groups: recommended and minimum

Recommended:

	PC
OS	Windows 10
CPU	Intel i7
Ram	32GB
Display	NVIDIA RTX 2080
Hard Disk	1 TB
Internet	1GB
Additional software	Web browser: google chrome

	Smartphone
OS	Android 10
Display	6.7 inches
Resolution	1440 x 3200
Memory	12GB
Capacity	128GB

Minimum:

	PC
OS	Windows 7
CPU	Intel i5
Ram	16GB
Display	NVIDIA GTX 1060
Hard Disk	3GB
Internet	200+KB
Additional software	Web browser: google chrome

	Smartphone
OS	Android 9.0
Display	6.4 inches
Resolution	3040 x 1440
Memory	8GB
Capacity	128GB

Problem Analysis

Use Case Descriptions

Actor description

Actor	Description
Relatives	1. Relatives can receive notifications of injuries to elders. 2. They can receive the animated picture of the elders falling to confirm the situation. 3. They can change the personal information of elders.
Caretakers	1. Caretakers can receive notifications of injuries to elders. 2. They can receive the animated picture of the elders falling to confirm the situation. 3. They can change the personal information of elders.
Elderly	1. When elders get hurt, they can receive medical care immediately

Use case

Use Case Name	Register Account
Use Case ID	UC-001
Actor(s)	Relatives and Caretakers
Brief description	The relatives or caretakers of the elder can register an account by inputting their name, valid phone number, email, HKID, password.
Preconditions	
Post-conditions	Email address and password are verified. Also, when a new account is created and the related information is stored in the database.
Flow of events	1. Enter a valid email address, phone number, HKID, password. 2. The website will ask the user to log in to the new account 3. Users enter the email address and password. Website display account activated successfully.
Alternative flows and exceptions	1. If the email address has been registered or invalid, the system will ask the user to input it again.
Non-behavioral requirements	Create a record for a new relative to the database
Assumptions	The caretakers should input valid social worker ID to ensure their identity
Issues	
Source	

Use Case Name	Login Account
Use Case ID	UC-002
Actor(s)	Relatives and Caretakers
Brief description	Before taking any action on the website, users should log in to their account to ensure security and privacy
Preconditions	Users should have
Post-conditions	Email address and password are verified.
Flow of events	1. Enter a valid email address with the correct password 2. Users can check the information and status of their relatives. 3. Website displays add the new elder successfully.
Alternative flows and exceptions	1. If the email address is invalid or the password is incorrect, the system will ask the user to input it again.
Non-behavioral requirements	
Assumptions	
Issues	
Source	

Use Case Name	Add new elders
Use Case ID	UC-003
Actor(s)	Relatives and Caretakers
Brief description	Users can add the information of their relative's elder
Preconditions	User should log in to the website
Post-conditions	Email address and password are verified.
Flow of events	1. Press the Add New Elder button on the Elder Information page. 2. Input the information of the elder such as name, HKID, address, phone number, and gender.
Alternative flows and exceptions	1. If the HKID is invalid, the system will ask the user to input it again.
Non-behavioral requirements	The website will create a new record of the new elder to the database.
Assumptions	
Issues	
Source	

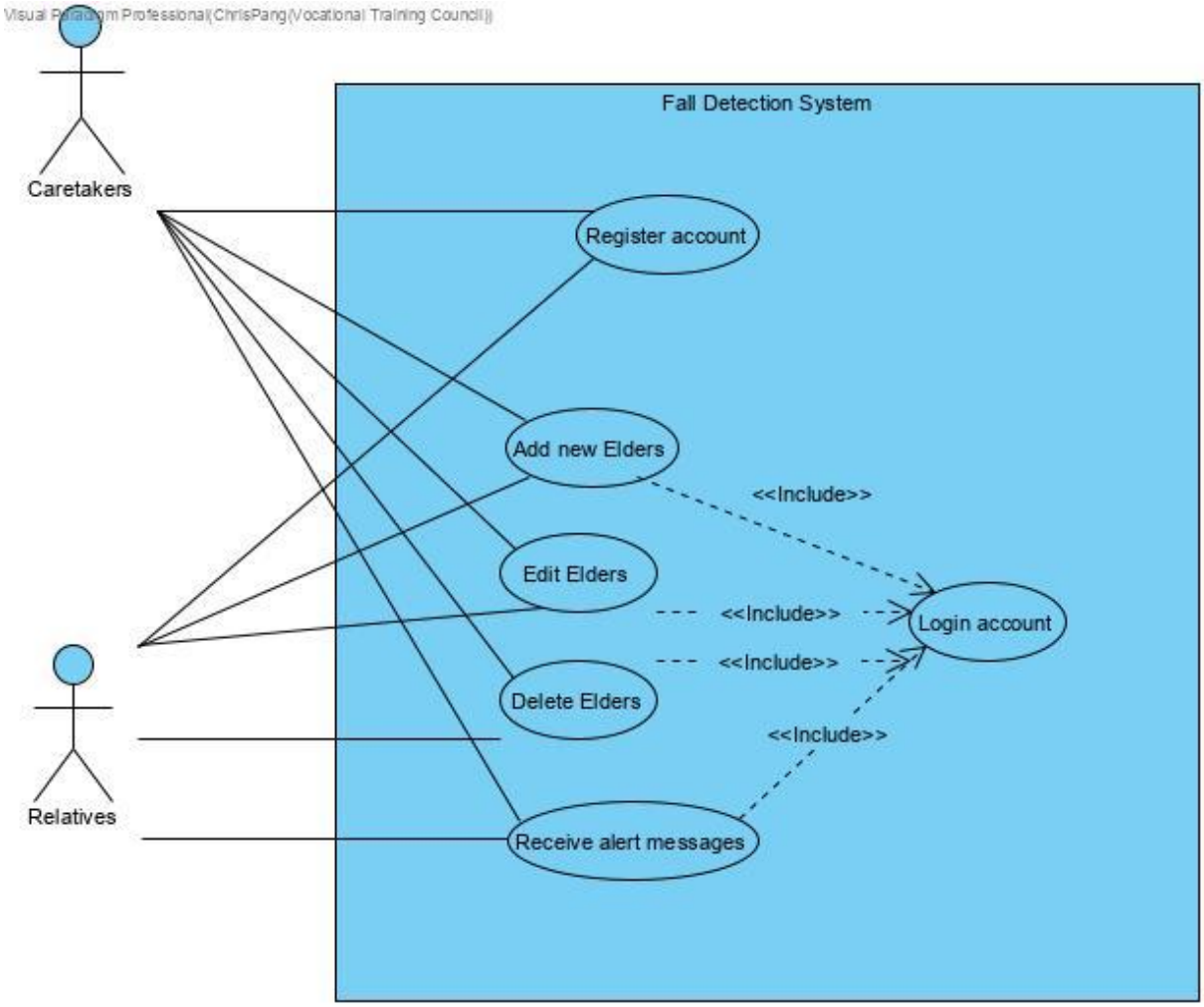
Use Case Name	Delete elder’s information
Use Case ID	UC-004
Actor(s)	Relatives and Caretakers
Brief description	Users can delete the information of their relative's elder when the elder is no longer using the service
Preconditions	User should log in to the website
Post-conditions	Email address and password are verified.
Flow of events	1. Press the Delete Elder button on the Elder Information page
Alternative flows and exceptions	
Non-behavioral requirements	The website will drop the information of the elder in the database.
Assumptions	
Issues	
Source	

Use Case Name	Edit elder’s information
Use Case ID	UC-005
Actor(s)	Relatives and Caretakers
Brief description	Users can edit the information of their relative's elder
Preconditions	User should log in to the website
Post-conditions	Email address and password are verified.
Flow of events	1. Press the Edit Elder button on the Elder Information page 2. Input the information that needs to be edited such as address and phone number.
Alternative flows and exceptions	
Non-behavioral requirements	The website will update the new information of the elder in the database.
Assumptions	
Issues	
Source	

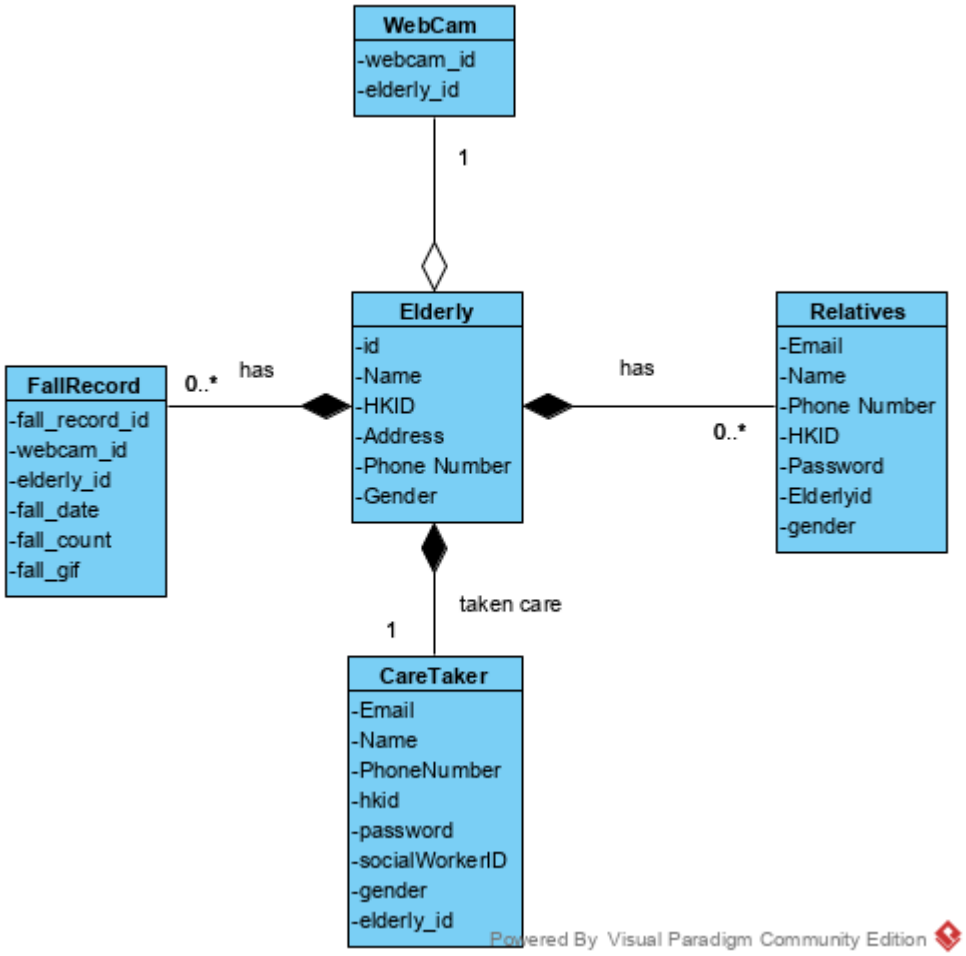
Use Case Name	Receive alert messages
Use Case ID	UC-006
Actor(s)	Relatives and Caretakers
Brief description	Users receive alert messages and emails from the system.
Preconditions	System detected the elders get hurt
Post-conditions	
Flow of events	1. The system detected elders getting hurt at home. 2. The system will cap the frames when it detected the elders fall 3. The system sends a message and emails automatically to the relatives and caretakers of the elder with the animate picture. At the same time, the system will call 999 for help.
Alternative flows and exceptions	
Non-behavioral requirements	
Assumptions	
Issues	
Source	

Use Case Diagram

Relatives Caretakers

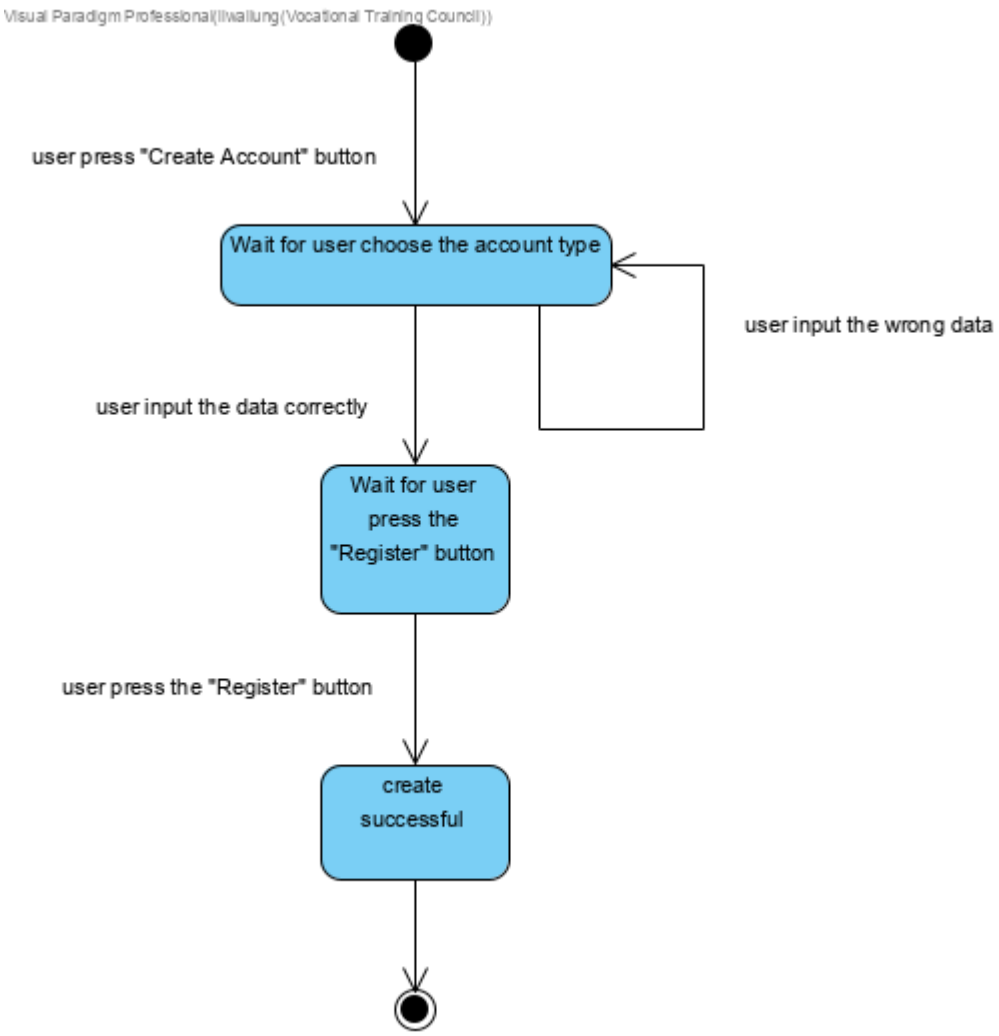


Class Diagram

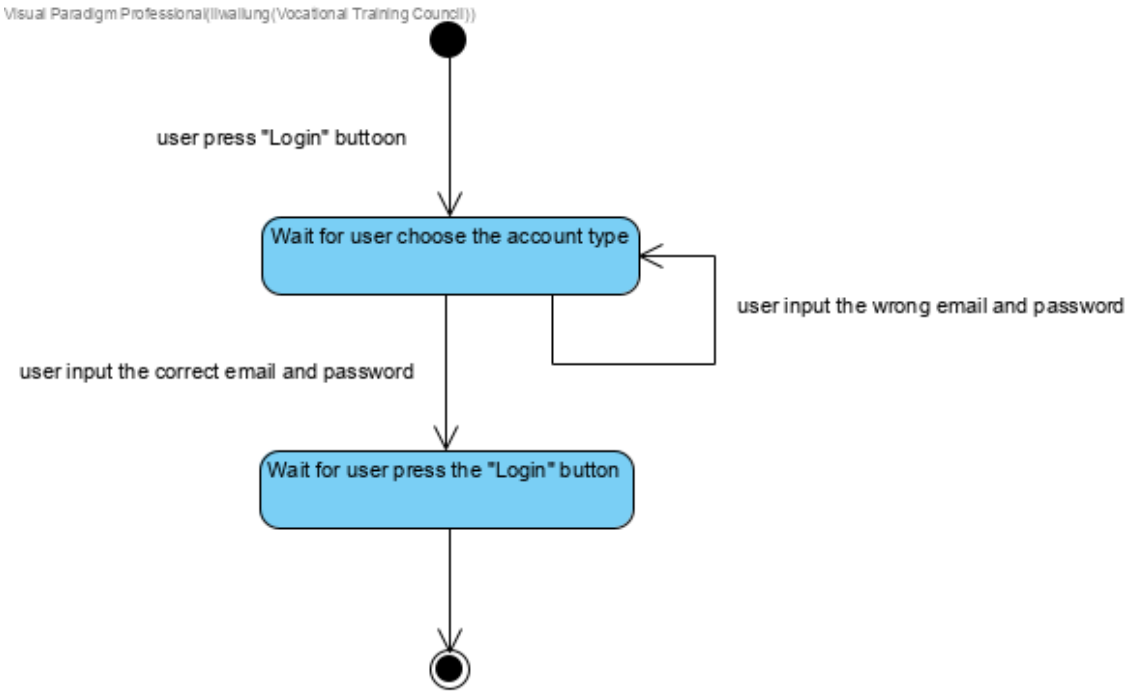


State Transition Diagram

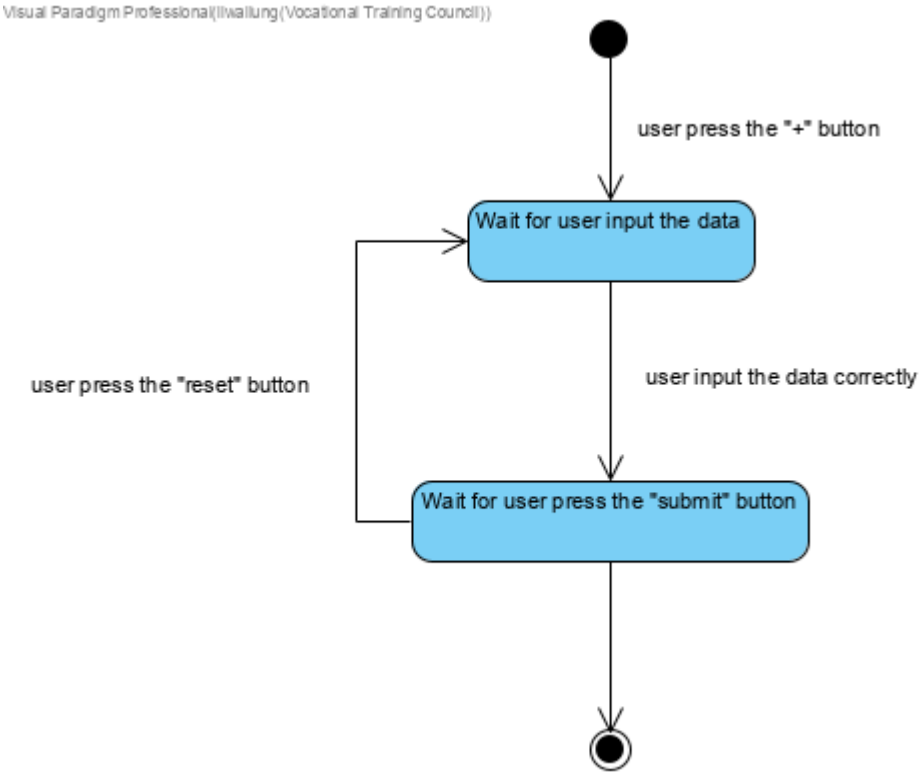
Create account



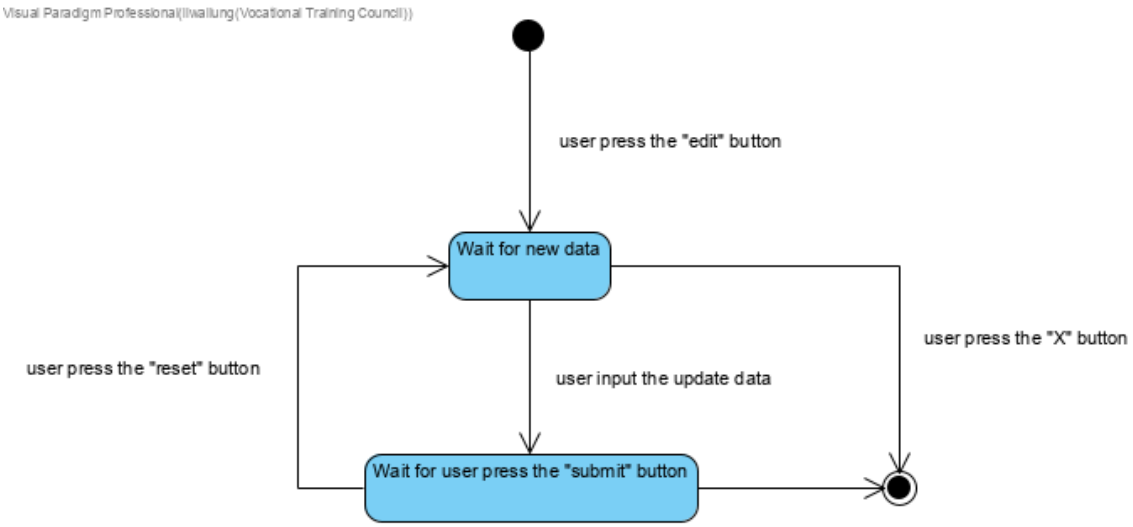
Login account



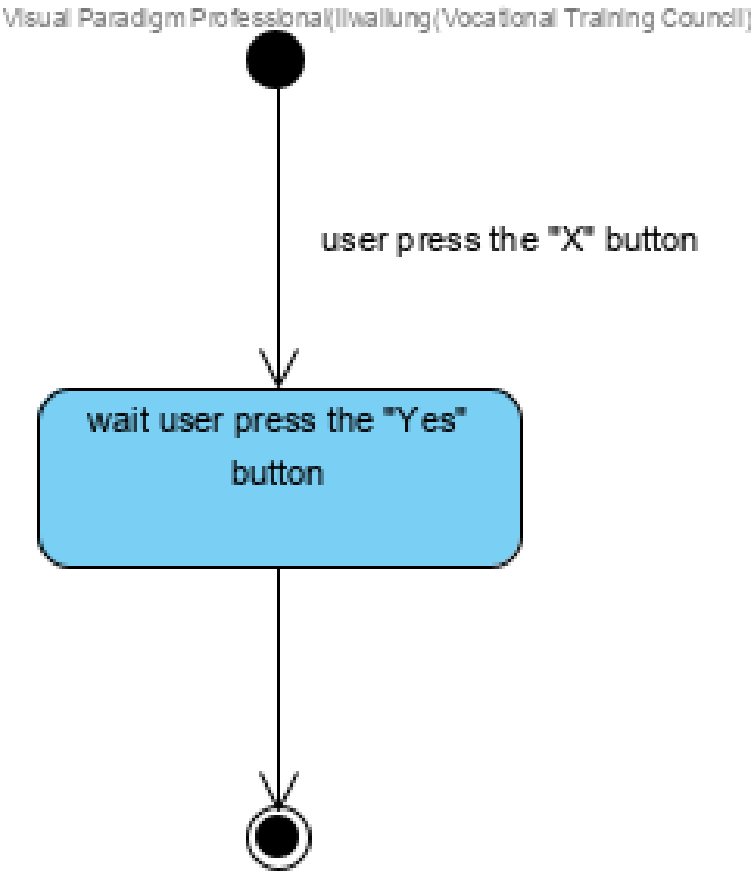
Add new elderly data



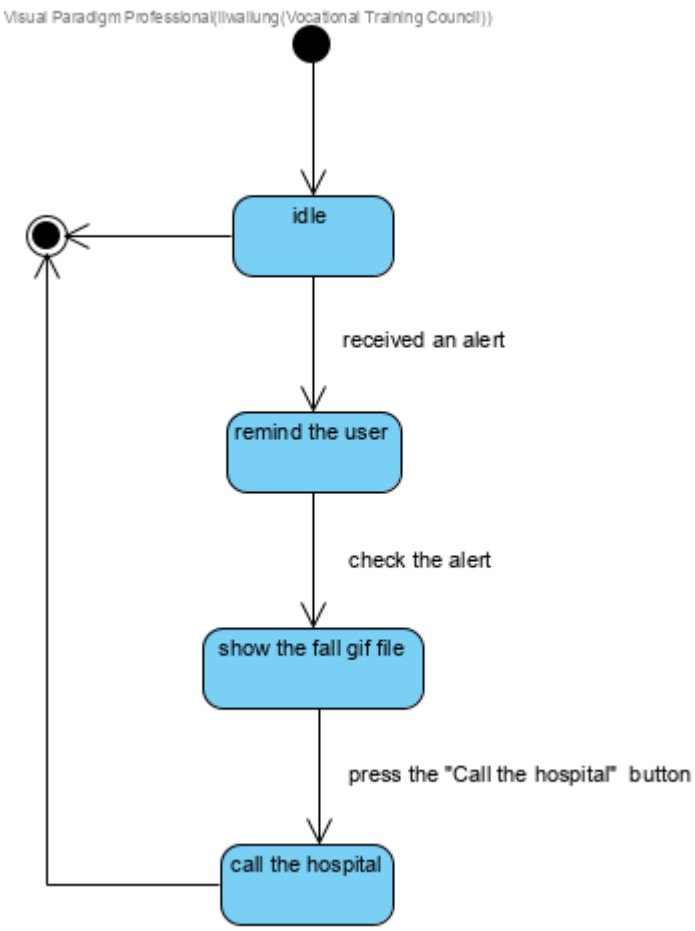
Edit elderly data



Delete the elderly data

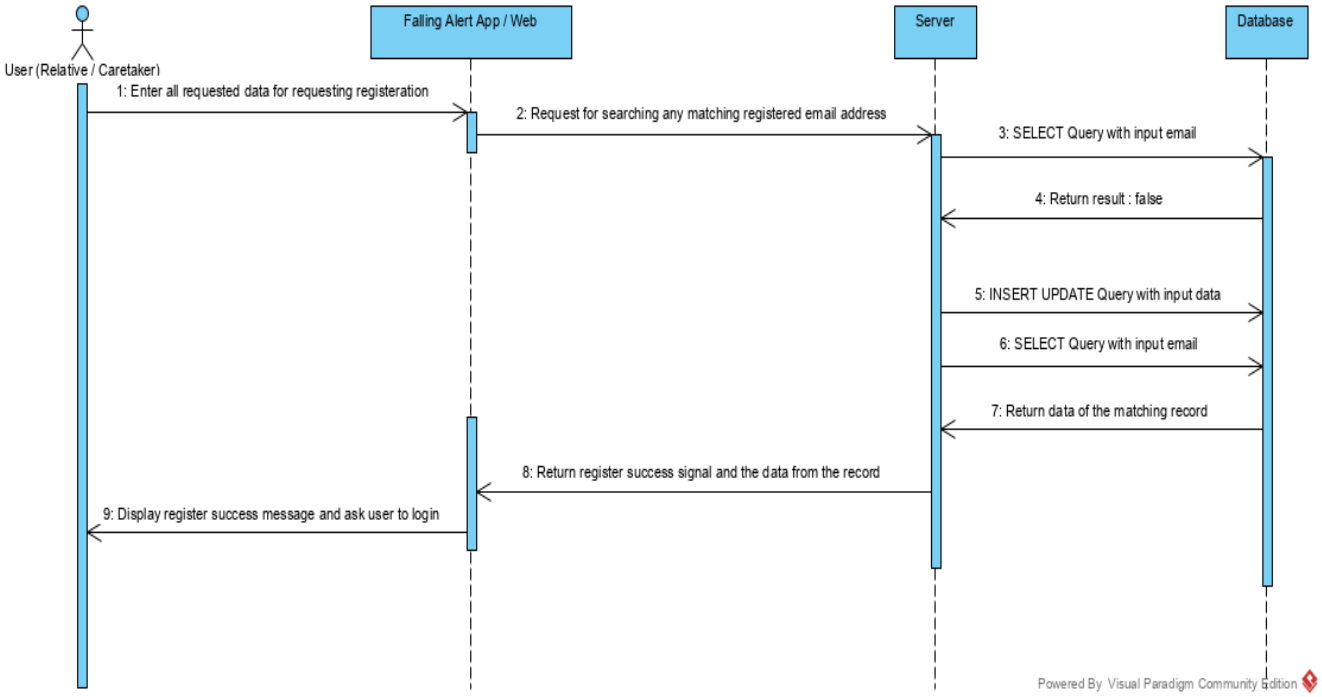


Receive alert

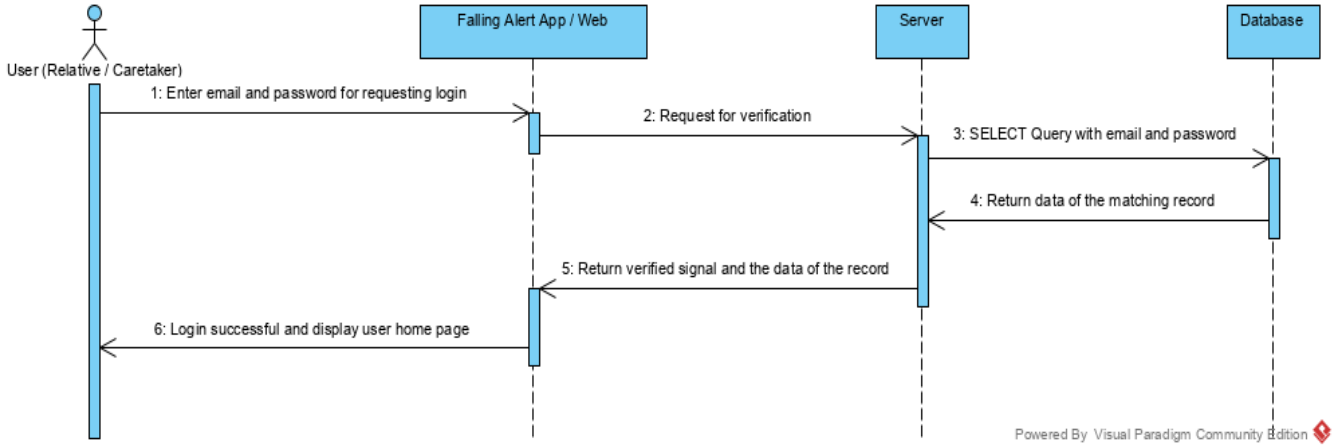


Sequence Diagram

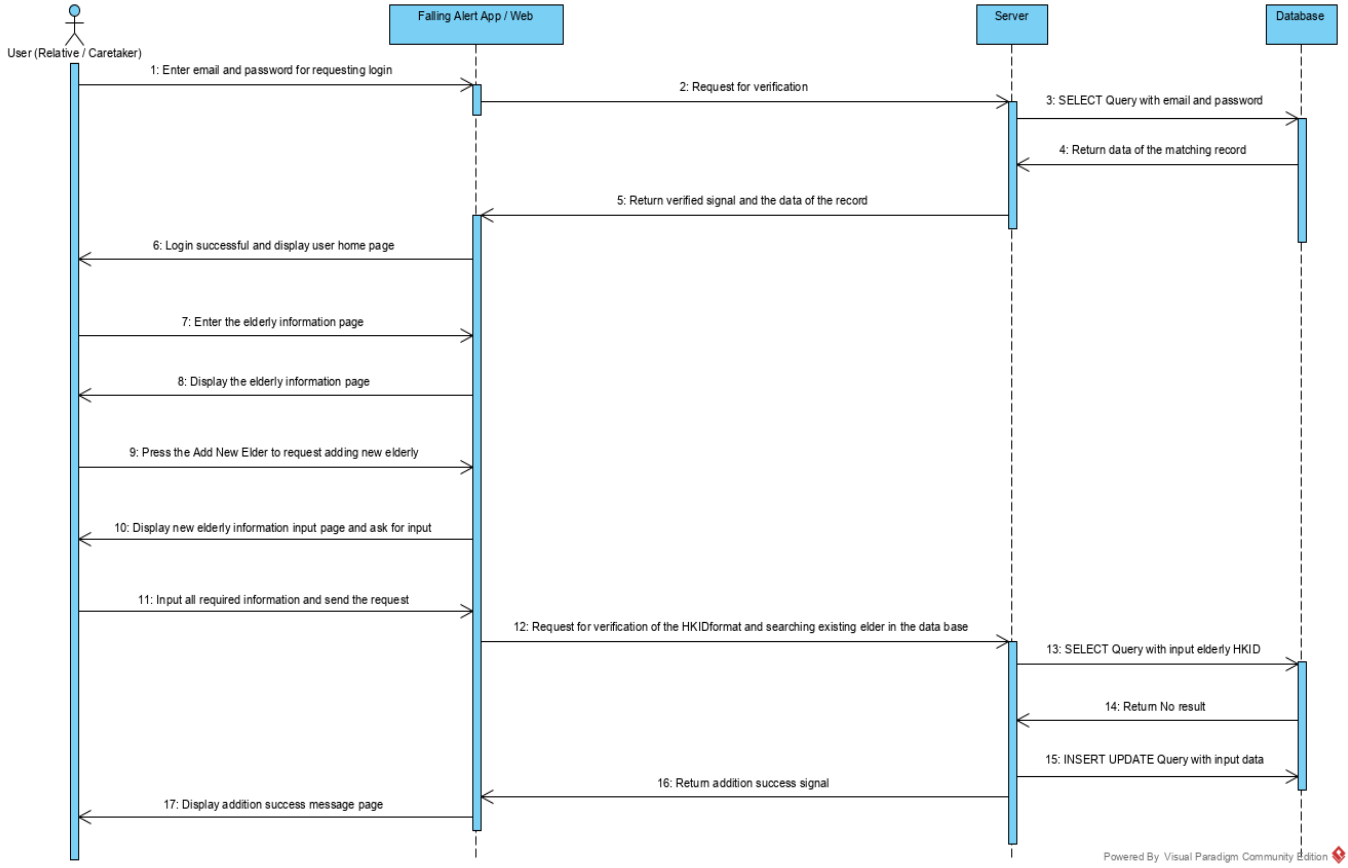
Register



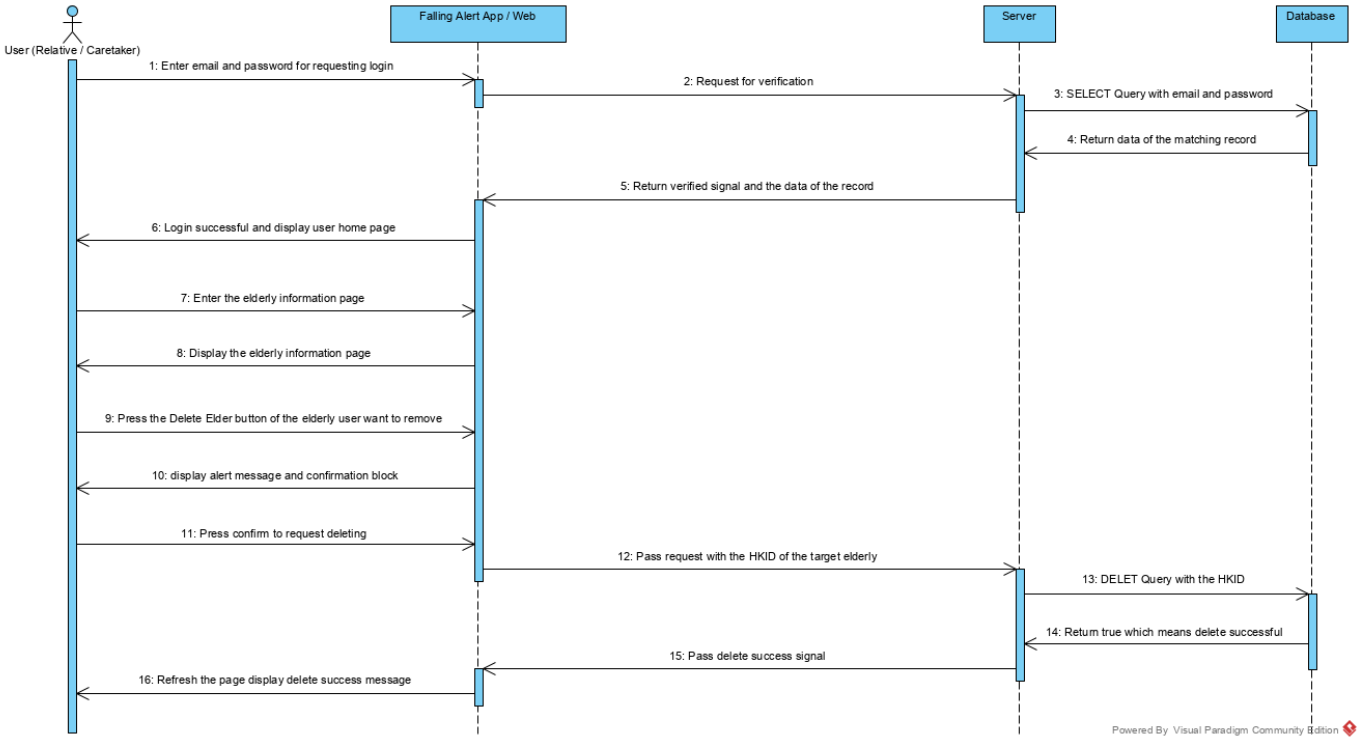
Login


User (Relative / Caretaker)
Falling Alert App / Web
Server
Database
1: Enter email and password for requesting login
2: Request for verification
3: SELECT Query with email and password
4: Return data of the matching record
5: Return verified signal and the data of the record
6: Login successful and display user home page
Powered By Visual Paradigm Community Edition

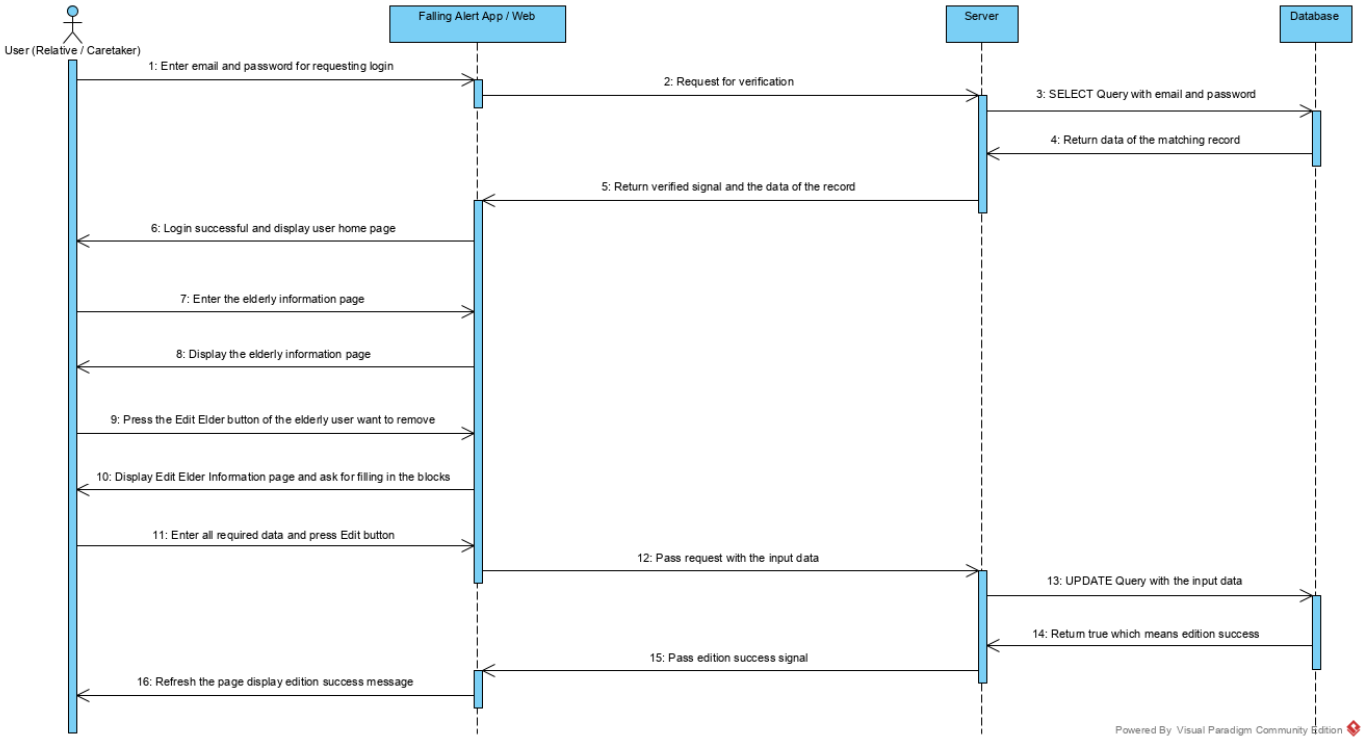
Add New Elderly



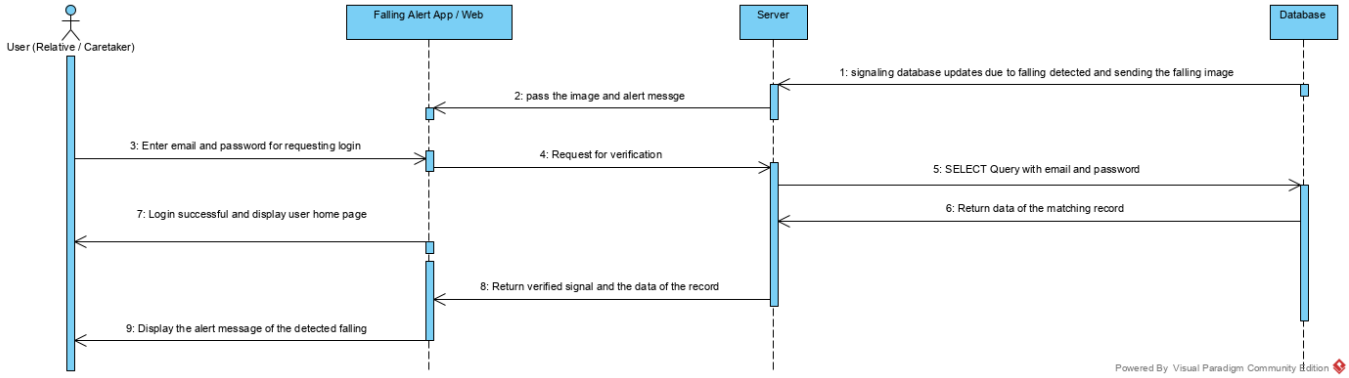
Delete Elderly



Edit Elderly Information

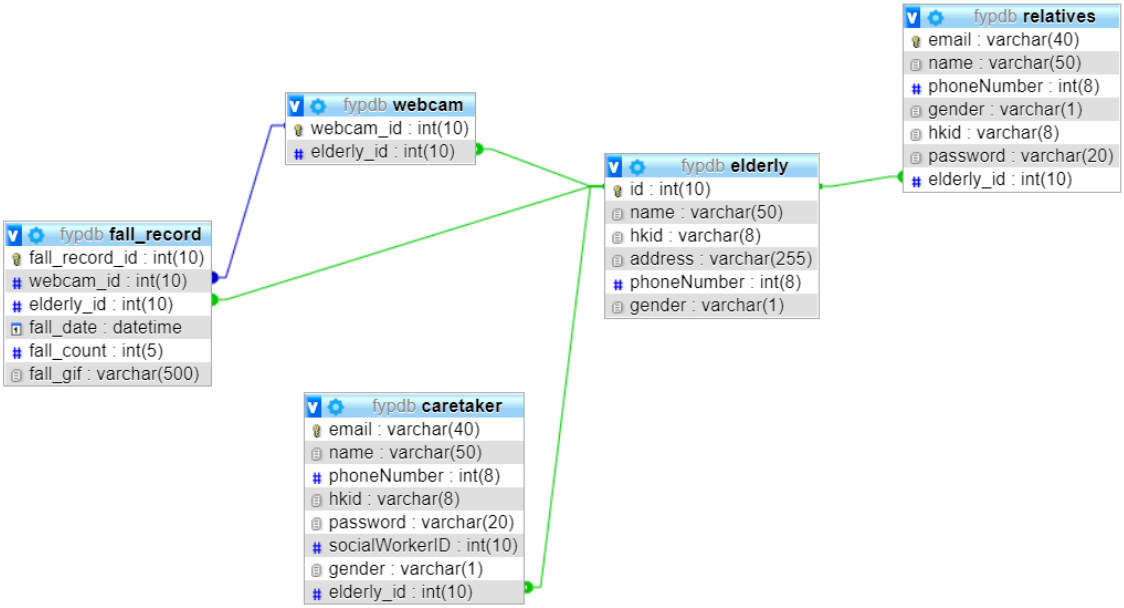


Receive Alert Message

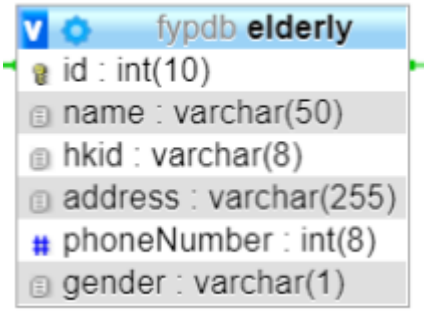


Data description

Entity Relation Diagram

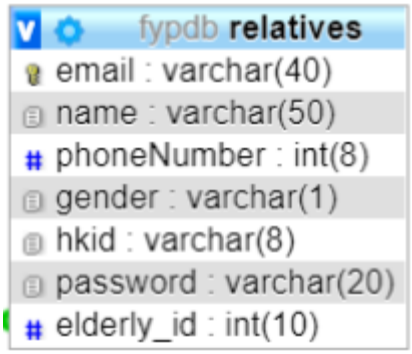


Data Dictionary



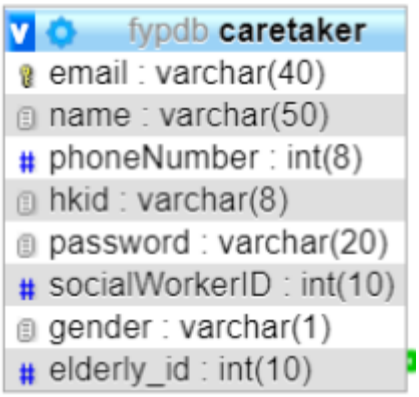
Elderly

Field Name	Data type	Data Format	Field Size	Description	Example	NULL
id	integer	NNNNNN	6	Unique number for all elderly	000001	Primary Key
Name	varchar		50	Name of the elderly	Chan Tai Man	NOT NULL
HKID	varchar		8	HKID of elderly	U123456(7)	NOT NULL
Address	varchar		255	Address of elderly		NOT NULL
Phone Number	integer	NNNNNNNN	8	Phone Number of elderlies	32442414	NULL
Gender	char		1	Gender of elderly	M	NOT NULL



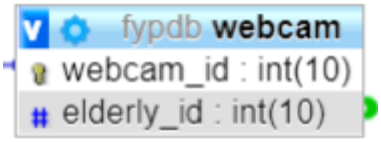
Relatives

Field Name	Data type	Data Format	Field Size	Description	Example	NULL
Email	varchar		40	Email of relatives	abc@gmail.con	Primary Key
Name	varchar		50	Name of relatives	Chan Siu Man	NOT NULL
Phone Number	integer	NNNNNNNN	8	Phone number of relatives	12346543	NOTNULL
HKID	varchar		8	HKID of relatives	A654321(7)	NOTNULL
Password	varchar		20	Password for relatives to log in to the system	aWeughasiuih	NOTNULL



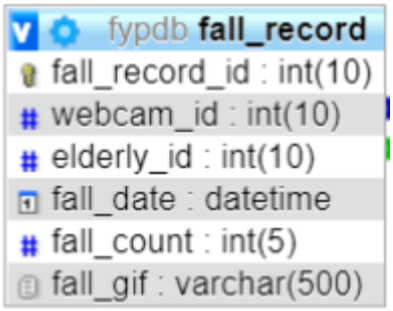
Caretaker

Field Name	Data type	Data Format	Field Size	Description	Example	NULL
Email	varchar		40	Email of caretaker	bcd@gmail.com	Primary Key
Name	varchar		50	Name of caretaker	Li Wai Lung	NOTNULL
Phone Number	integer	NNNNNNNN	8	Phone Number of caretakers	91324789	NOTNULL
SocailWorkerID	integer	NNNNNNNNNN	10	The ID of the Social Worker	0375205872	NOT NULL



Webcam

Field Name	Data type	Data Format	Field Size	Description	Example	NULL
webcam_id	int		10	id of webcams	1001	Primary Key
elderly_id	int		10	id of elderly	1	Foreign Key



Fall Record

Field Name	Data type	Data Format	Field Size	Description	Example	NULL
fall_record_id	int		10	id of fall records	100001	Primary Key
webcam_id	int		10	id of webcams	1001	Foreign Key
elderly_id	int		10	id of elderly	1	Foreign Key
fall_date	datetime			date of elderly falling	1-24-2021	NOT NULL
fall_count	int		5	times of elderly falling	1	NOT NULL
fall_gif	varchar		500	animated picture of elderly falling	D:/FYP\V1_lib/Realtime-Action-Recognition-master/gif_data/21-01-17 20:30:40-1.gif	NOT NULL

Data Storage Design

Data Formats Selection

We store most of our data with the relational database, but there are some data don't suitable for a relational database such as media and temporary information like falling gif, we will store data in different formats.

Data Examples

Data	Data Type	Format
Elders' information	Mostly text	relational
Caretakers' information	Text and Numbers	relational
Falling Gif	media	link

Estimating Storage Size

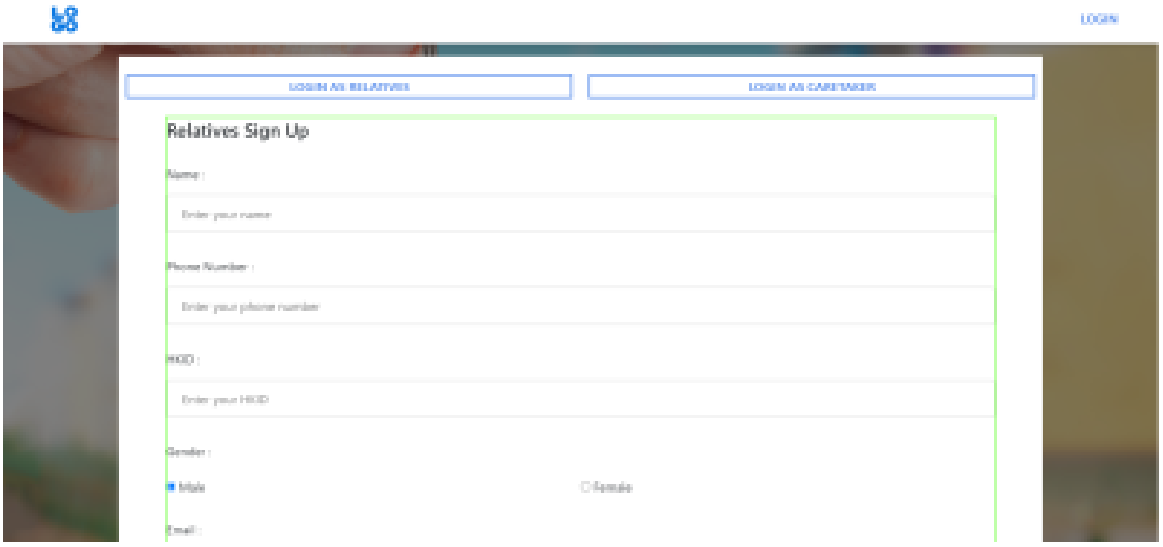
The system will perform poorly if the database server cannot handle its volume of data. We need to estimate the amount of data that the hardware will need to support.

User Interface Design (Website View)

Home Page

When users come to our website, they will see this page. This page is used to introduce our system simply.

Sign-up Page

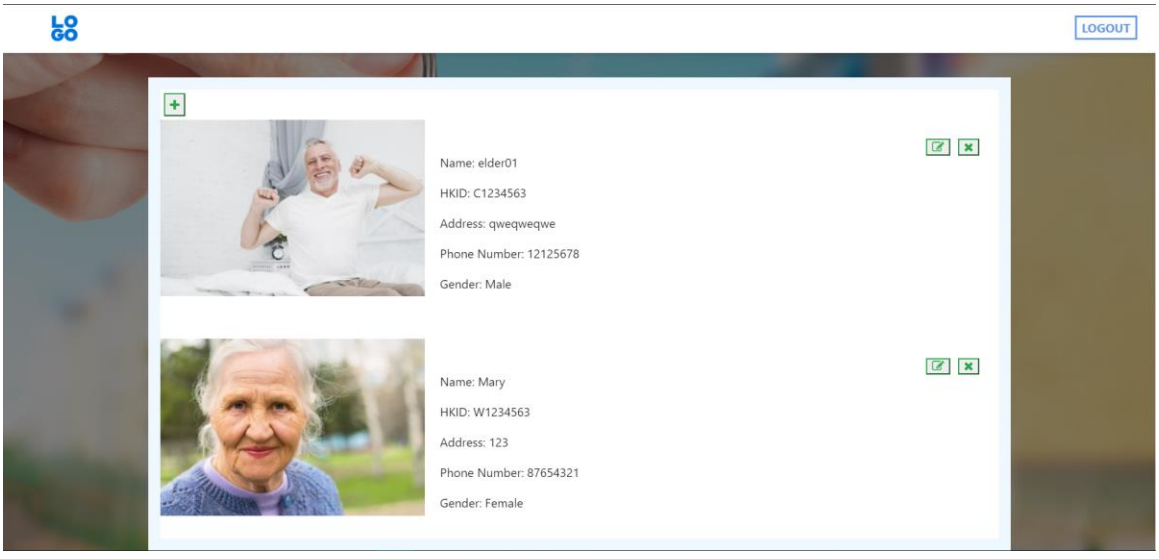


This page provides a sign-up system to the new user.

Login

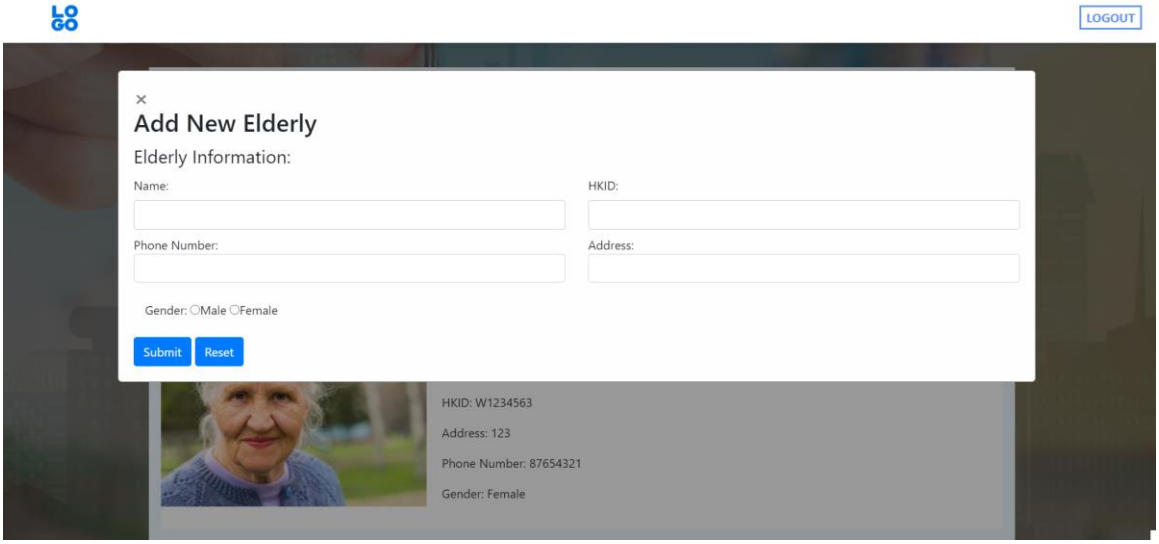
This page is inside the home page and the sign-up page which can let people log in.

Elderly Information Page



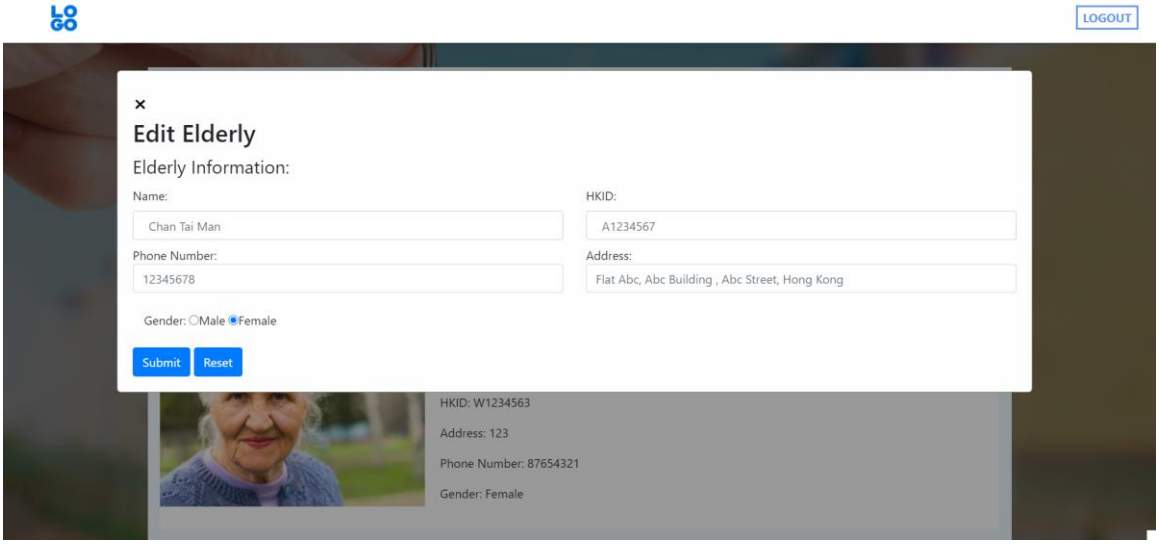
This page has the information of elders such as name, HKID, address, phone number, and gender.

Add New Elderly Page



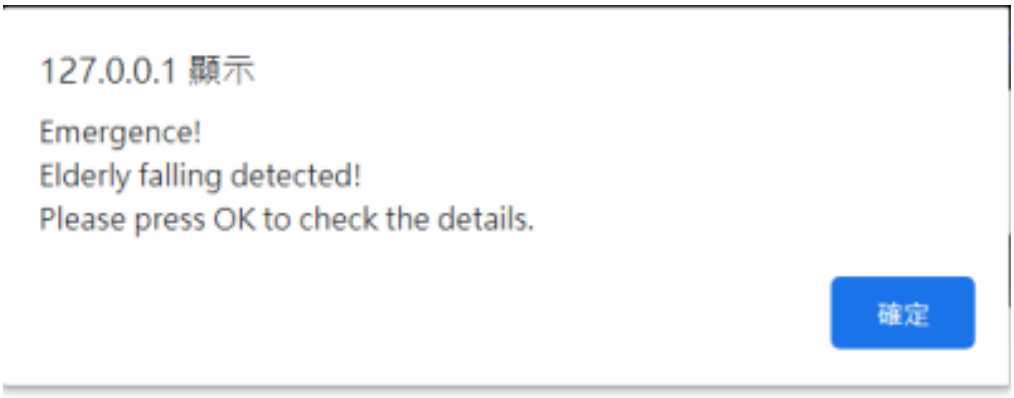
This is the page for relatives or caretakers to add a new elderly. They need to input the name, HKID, phone number, address, and gender of the elder.

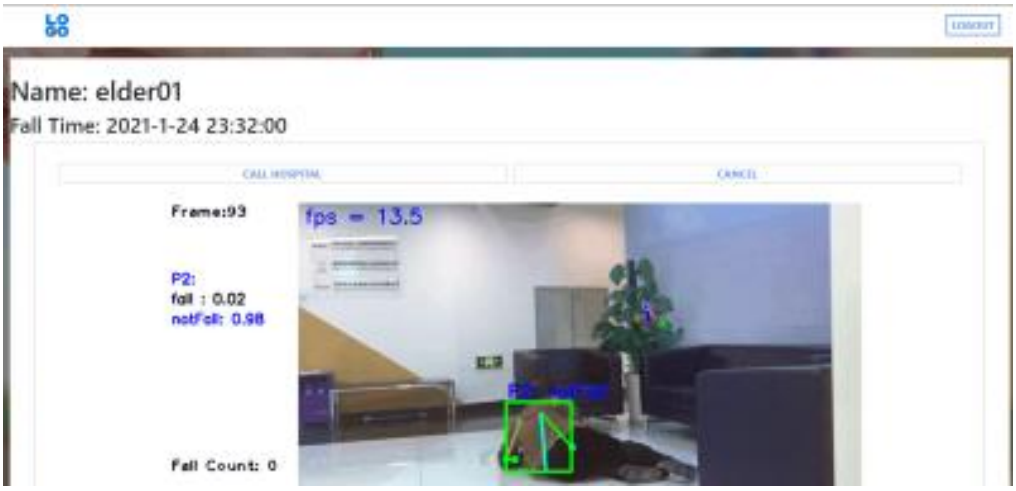
Edit Elderly Page



Relatives or caretakers can edit elders' information on this page.

Alert Page





When the system detects a fall, our website will show an emergence message to our user on the Elderly Information Page. When they press the “OK” button, the system will bring them to the alert page to learn more about the details.

User Interface Design (Phone View)

Loading Page

This page will be shown when relatives or caretakers open our app.

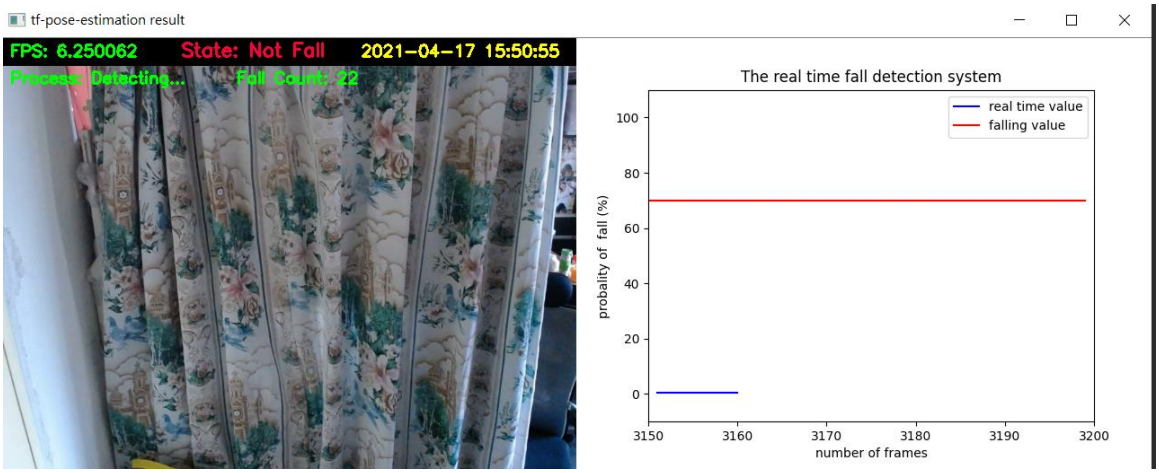
Elderly Information Page

This page allows relatives or caretakers to check the information of elders such as name, HKID, address, phone number, and gender. Also, if they want to check the latest fall detail information, they can press the "CHECK FALL DETAIL" button to get more.

Alert Page

When relatives or caretakers press the "CHECK FALL DETAIL" button, they will go to this page. This page will display some important information about falls such as name, fall time, and a gif showing the move when the elders fall.

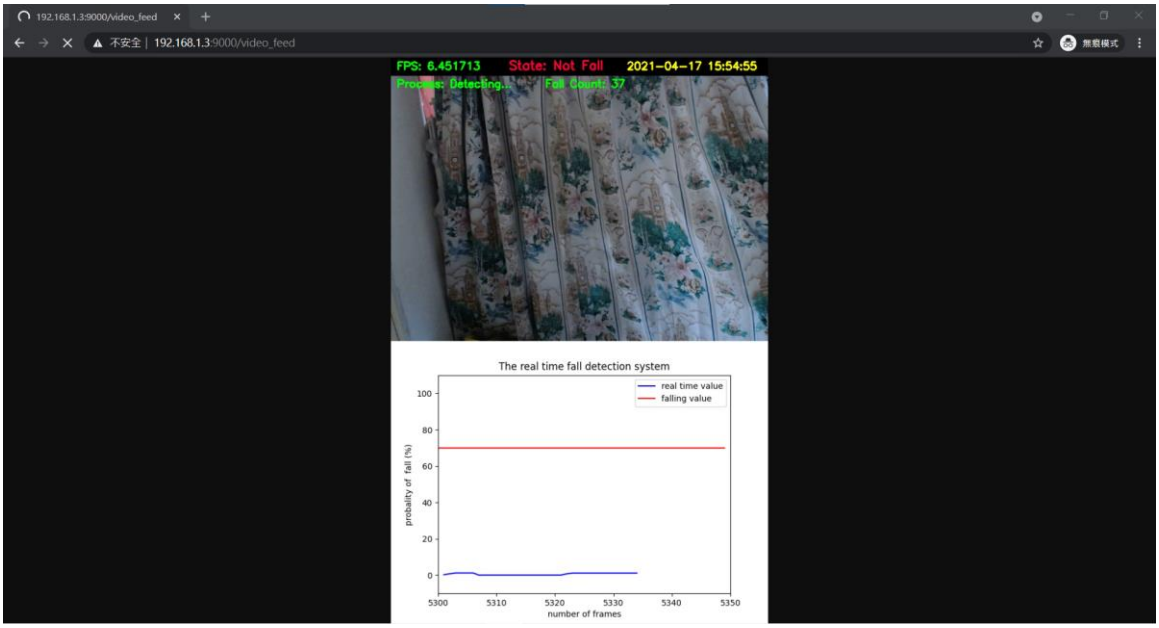
User Interface Design (Fall System)



This is the fall detection system UI shows on the computer view. The skeleton human detected by openpose will be displayed on the left side of the picture. And some data, such as FPS, State, current time, system state (Process), and the number of falls. There are 4 states of the system (Process)：

- Detecting
- Saving GIF images
- Sending email
- Sending WhatsApp

Shown on the right is a real-time line chart, the x-axis represents the current frames, and the y-axis represents how many percentages are Fall. The blue line is the current percentage, and the red line is the set Fall value of 70%. If blue surpasses red, it is Fall, and then the data will be processed and information will be sent.



This is the website's real-time view. The information displayed on the web version is the same as the information displayed on the computer version but considering that there will be mobile users in the web version, the direction has been changed. In this way, all data can be displayed on one screen.

Advantages and drawbacks of the solution

Advantages:

Since part of the system involves implementing 3-rd party open-source libraries, the development time is shorter compared to developing it from zero. Moreover, the performance of those open-source libraries is mostly well. Developing our own algorithm may have a chance that our algorithm performs worse at the end.

The streaming function makes it easier for the family or the caretaker to monitor the elderly's situation.

The automatic notification and emergency call functions have the elderly get help on time since the function activates at the moment that the incident happened to the elderly.

Drawbacks:

Since those are 3-rd party libraries there may be some difficulties when we want to tune some of the values or features in the library due to not being completely familiar with the source code or even being unable to have the source code. Also, there may be conflicts in terms of compatibility between libraries implemented by different parties which will take time to make adapters for both of them to be compatible with each other.

The streaming function required a good speed of the network. This may lead to the fact that only users who have a high-speed Wi-Fi plane for their smart are able to have streaming with a high frame rate and high resolution.

Since there is a chance to have false-positive results on fall detection, the automatic notification and emergency call functions would be annoying and maybe wasting resources of the hospital if somehow there are false-positive results that keep happening.

Implementation

Test Plan

To ensure the usability and the performance of the system, we decided to do the following four types of tests.

Unit Testing

We think Black-box testing will be suitable to test the program performs its functions as defined in the program specification.

We can test the system with:

- The accurate action performed by users
- System's interaction with inputs
- The response time of the system
- Usability issues

Testing Example:

Test case ID	TC-001
Test Case	Check invalid Login
Tester	Chris Pang
Test scenario 1	User login the system with correct email and wrong password
Test scenario 2	User login the system with wrong email and wrong password
Test scenario 3	User login the system with the correct email and correct password
Result	Passed
Objective	To ensure that the users can't log in to the system with an invalid email or password
Expected result	Users will receive an error message when they input invalid information and the system will ask them to input it again
Actual result	Same as the expected result

Integration Testing

User Interface Testing

The purpose of the user interface testing is to ensure all the elements of UI functionality works.

Here's the checklist below:

- The UI elements for size, position, width, length, and the acceptance of characters or numbers.
- Can we execute the intended functionality of the system through the UI
- The alert message is displayed correctly
- The alignment of the text and image are proper
- The images have good clarity
- The positioning of UI elements for different screen resolution
- The font used in the system is readable

Test Case	Test the input required fields
Description	If the system requires data entry on a specific field (such as password when user login the system), we will provide a red asterisk with the required and give a pop-up error message if the data is left blank
Result	Passed
Expected result	When the user didn't input the required data there will be an error message pop-up
Actual result	Same as the expected result

System Testing

Requirements testing

About the requirements testing, we will conclude all the requirements of our system, and test can the system achieve the functional and non-functional requirements.

Testing Example

Test Case	Live stream testing
Description	Live broadcast for the family members to check the elder's situation after a fall
Expected result	The family members can use the website or mobile app to check the live stream after the elder fall

Performance testing

The performance testing will test the system's ability to perform under high workloads.

Here's the checklist below:

- Speed: Determines whether the system responds quickly
- Scalability: Determines maximum user load the system can handle
- Stability: Determines if the system is stable under varying loads

Testing Example

Test Case	Test the system stable when high volumes of data generated to the system
Description	We will test multiple fall record created and all the records read and write to the database at the same time
Expected result	The database execution time below 1000ms

User Guide

Website

1. Open your browser and go to our home page.
2. If you have an account, then go to step 4. If you do not have an account, please press the "CREATE AN ACCOUNT" button and go to step 3.
3. Sign-up

i. Choose your account type.
ii. Enter all the information.
iii. Press the "REGISTER" button.

4. Login

i. Choose your account type.
ii. Enter your correct email and password.
iii. Press the "Login" button.

5. If you already have the elderly information on your home page, then please go to step 7. If you do not have the elderly information on your home page, please press the "+" button and go to step 6.
6. Add new elderly information

i. Enter all the information.
ii. Press the "Submit" button

7. If you want to edit the elderly information, please go to step 8. If you want to delete the elderly information, please go to step 9.
8. Edit elderly information

i. Press the button which is next to the button "X".
ii. Enter the information which you want to change.
iii. Press the "Submit" button.

9. Delete elderly information

i. Press the "X" button.

10. If there is an emergency alert, please press the "OK" button to check the fall details.
11. If you think the elderly are in danger you can press the "CALL HOSPITAL" button to call the hospital. If not, you can press the "CANCEL" button to cancel this alert.

Mobile App

1. If you have a message from WhatsApp which is about the elderly fall, you can open our app and go to step 2. Also, if you want to edit the elderly information, you can open our app and go to step 4. If you want to delete the elderly information, you can open our app and go to step 5.
2. Check fall detail information

i. Press the "CHECK FALL DETAIL" button

3. If you think the elderly are in danger you can press the "CALL HOSPITAL" button to call the hospital. If not, you can press the "CANCEL" button to cancel this alert.
4. Edit elderly information

i. Press the button which is next to the button "X".
ii. Enter the information which you want to change.
iii. Press the "Submit" button.

5. Delete elderly information

i. Press the "X" button.

Fall System

1. Enter “Win + R”
2. Enter “cmd” and press “Enter”
3. cd to the file
4. Enter “python run_wecam.py”
5. Wait for loading the Live Stream setup
6. After successfully loaded the live stream setup enter “CTRL + C”
7. Wait for loading the Fall Detection System setup

Installation Guide

Mobile App

1. Open the App store on your phone.
2. Search the app called "Fall Detection".
3. Press the "Install" button.

Fall System

1. Install the required library and suitable version.

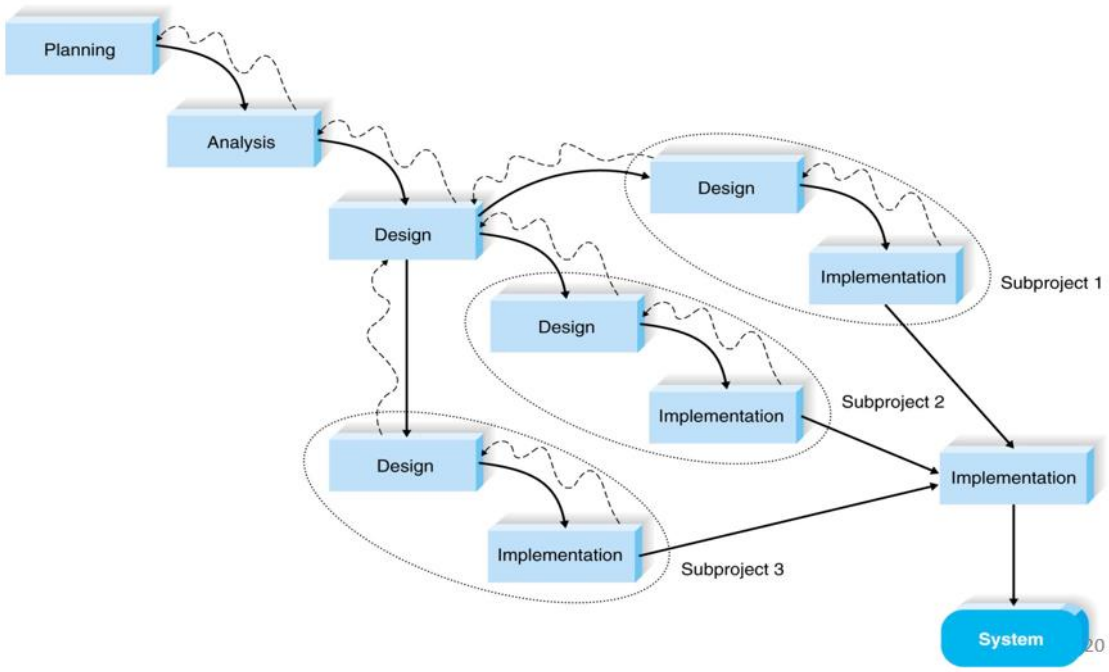
Library	Version
Keras	2.3.0
Tensorflow	1.15.0
Tensorflow-GPU	1.15.1
Pandas	1.1.2
Numpy	1.19.3
Matplotlib	3.3.2
OpenCV-python	3.4.2.17
Pillow	7.0.0
Twilio	6.45.4
Pymysql	1.0.2
Flask	1.1.2
Imageio	2.1.6

2. Install your current UPG version of CUDA

Software Process Model

Methodology

Parallel Development

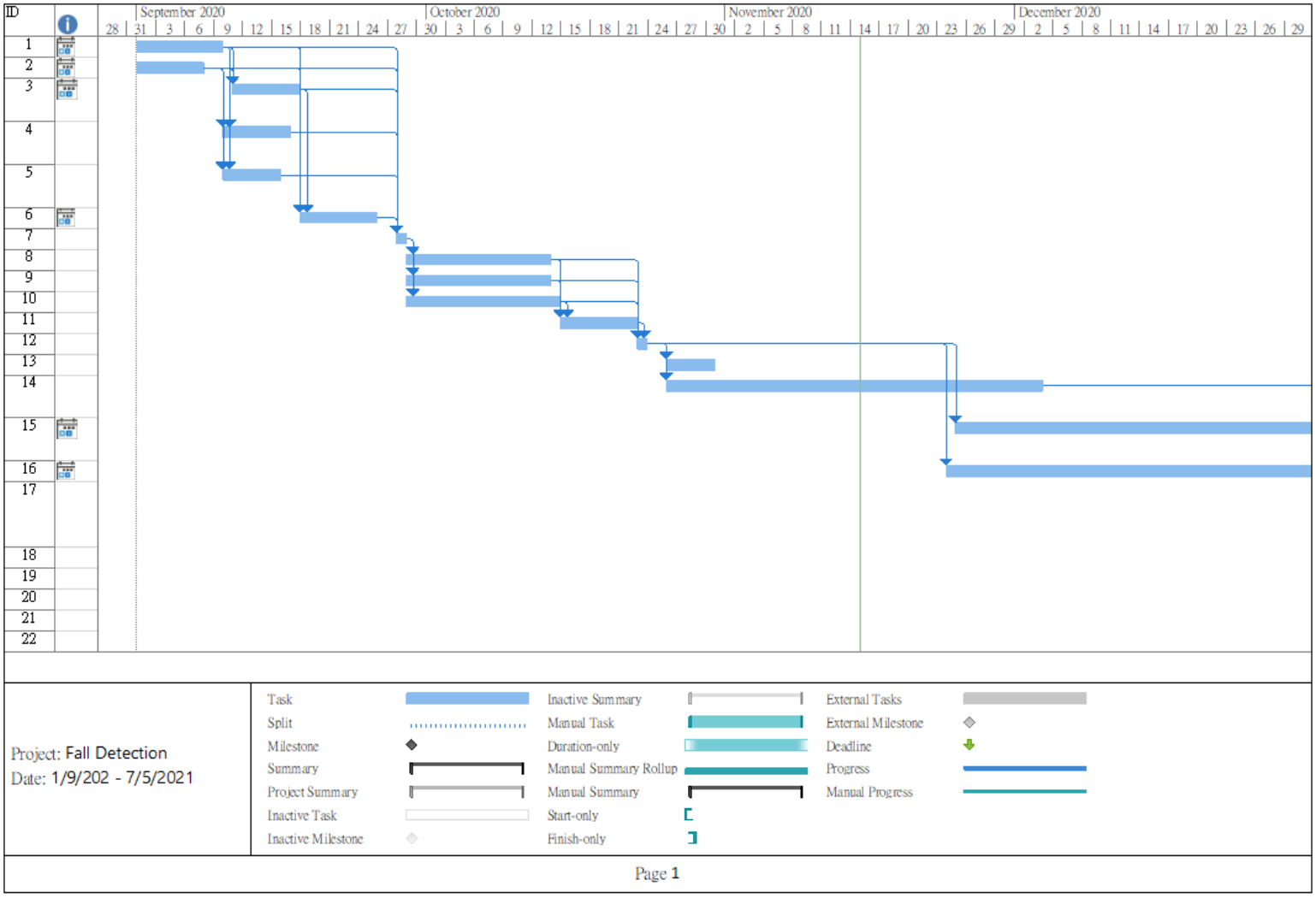


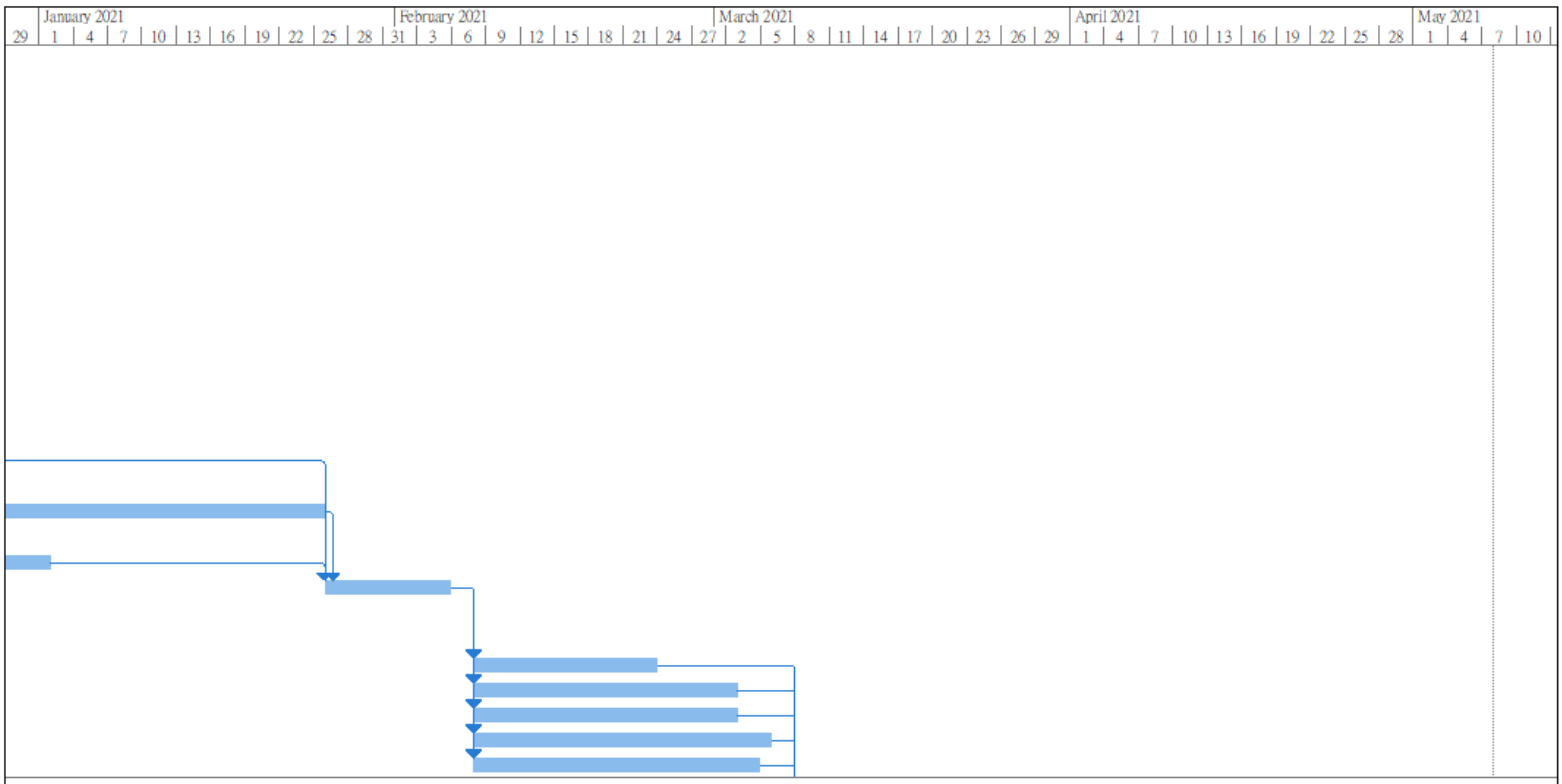
We decided to use parallel development since we only have a short time schedule to finish the project. We are going to divide the project during the design phase. Two members as a group to finish the functions such as UML design, UI design, Web/App design, Database Design, and Program design.

Project Plan

	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		System Evaluation	7 days	Tue 1/9/20	Wed 9/9/20	
2		System Requirement	5 days	Tue 1/9/20	Mon 7/9/20	
3		Create Requirement Specifications	5 days	Fri 11/9/20	Thu 17/9/20	1,2
4		Create Functional Requirement	5 days	Thu 10/9/20	Wed 16/9/20	1,2
5		Create Non-Functional Requirement	4 days	Thu 10/9/20	Tue 15/9/20	1,2
6		Schedule	6 days	Fri 18/9/20	Fri 25/9/20	1,2,3,4,5
7		Confirm System Proposal	1 day	Mon 28/9/20	Mon 28/9/20	1,2,3,4,5,6
8		Technical Feasibility	11 days	Tue 29/9/20	Tue 13/10/20	7
9		Economic Feasibility	11 days	Tue 29/9/20	Tue 13/10/20	7
10		Operation Feasibility	12 days	Tue 29/9/20	Wed 14/10/20	7
11		Create Project Plan	6 days	Thu 15/10/20	Thu 22/10/20	8,9,10
12		Confirm Initial Report 1	1 day	Fri 23/10/20	Fri 23/10/20	8,9,10,11
13		Architecture Design	5 days	Mon 26/10/20	Fri 30/10/20	12
14		Training Fall Detection Model	30 days	Mon 26/10/20	Thu 3/12/20	12
15		Test Fall Detection Model	44 days	Wed 25/11/20	Mon 25/1/21	12
16		Hardware Testing	29 days	Tue 24/11/20	Fri 1/1/21	12
17		Interim Report Presentation and Demonstration	9 days	Tue 26/1/21	Fri 5/2/21	14,15,16
18		UML Design	12 days	Mon 8/2/21	Tue 23/2/21	17
19		UI Design	17 days	Mon 8/2/21	Tue 2/3/21	17
20		Database Design	17 days	Mon 8/2/21	Tue 2/3/21	17
21		Web/App Design	20 days	Mon 8/2/21	Fri 5/3/21	17
22		Program Design	19 days	Mon 8/2/21	Thu 4/3/21	17
23		Progress Report and Mid-Semester Demonstration	5 days	Mon 8/3/21	Fri 12/3/21	18,19,20,21,22
24		Confirm Version	3 days	Mon 15/3/21	Wed 17/3/21	23
25		Testing	10 days	Thu 18/3/21	Wed 31/3/21	24
26		Documentations	11 days	Thu 1/4/21	Thu 15/4/21	25
27		Final Report	1 day	Fri 16/4/21	Fri 16/4/21	25,24,26
28		Debug	5 days	Mon 19/4/21	Fri 23/4/21	27
29		Final Report Presentation and Demonstration	10 days	Mon 26/4/21	Fri 7/5/21	28

Gantt Charts





Project: Fall Detection
Date: 1/9/202 - 7/5/2021

Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only			
Inactive Milestone		Finish-only			

Software tools needed

GitHub

Obtain resources and test some simple programs related to fall detection, compare different programs, their advantages and disadvantages, and look for libraries that may be suitable for us.

Numerous sets of training and testing data

These datasets will determine the accuracy of our model, so we need a lot of data for training and testing. If a dataset contains videos that have more types of falls and different conditions (environment settings, lighting, indoor or outdoor, etc.), the performance of real applications will be better.

Books

We need additional knowledge to understand different types of machine learning and their operating methods.

IDE

We use PyCharm as our Integrated Development Environment for Python coding because it supports many functions. So, we can easily manage the files. And the android studio for an android app with JAVA coding. Also, the PHPStorm for website development with PHP coding.

Hardware tools needed

Use advanced graphics card

High-efficiency and high-performance processing are required according to the program, so advanced GPU memory is required, which can run at a very high speed.

Critical Evaluation

GPU memory limitations

In OpenPose, two graph models are supported, namely "cmu" and "mobilenet_thin". The "cmu" has a higher accuracy rate, but also requires more GPU at the same time. Therefore, using the "cmu" model can have higher detection accuracy. But our running speed will become only 2-3 FPS. On the contrary, the FPS using the "mobilenet_thin" model is 9-10, but there is no higher accuracy. After comparing, we still used "mobilenet_thin".

2D camera angle

Because we are using a 2D camera, there is only XY coordinate. But this is a problem that some of the angles are the same as the fall angle. This problem occurs in the blind spot of the camera. For example:

1. If a person stands on the edge of the camera, the detected skeleton is only half of the body
2. If a person stands in front of the camera, the detected skeleton is only half of the body

Incomplete skeleton

As the GPU memory limitations, so we choose the "mobilenet_thin" graph model. But this is not the closest to the original Openpose version. After the elderly fall, the detected action key points will be deformed. This increases the difficulty of training the LSTM model. Also affects the accuracy of the system and becomes a congenital defect of our system.

Future features

Multi people detection

For now, our module can only detect one person on the screen and only provide for the elderly who live alone. That's why we may train another module to support the multi people detection in the future.

Use the 3D camera

At this moment, our camera is using a 2D camera. In the future, we can use a 3D camera to make the detection more accurate, detect more different human moves, and the distance between the camera and the elderly.

Upgrade the fall detection accuracy

In the future, we may try some other libraries not only the Openpose but also the OpenCV for our detection system and upgrade the fall detection accuracy.

Upgrade the fall alert function

We can build another module to replace the alert module we used now and that module can provide a faster reaction to noticing relatives or caretakers when the elderly is fallen in the future.

Conclusion

The ultimate goal of this project is to implement a fall detection system with overwatch functions for assisting elderly-caring. The fall detection section of the system should be more accurate compared to common systems existing on the internet. As a result, it can be said that the system did perform better than the commons. However, there is still a great space for the system to be improved.

This project is oriented by AI technology and the team granted valuable experience in this field. During the development of the system, we learned that the quality of the detection results can be affected by many conditions which are mostly from hardware. The greatest challenge when we are trying to train the model is the graphic card, or GPU if describing more accurately. We found that the processing power of the GPU of the graphic card affect the most because if the GPU process the streaming videos not fast enough, frame-losing may have occurred, and if the lost frame is the key frame (the frame that is able to give the highest confidence to falling), the chances that the falling is not detected will be increased. Moreover, the resolution of the webcam affects the result too due to the fact the higher resolution of the videos (no matter streaming videos or training videos) is, the easier it for Openpose to find out the human in the view of video and illustrate skeleton for detection.

Speaking of Openpose, we found that although this 3-rd party library brings conveniences to the development, its weakness gives a limitation of performance to the fall detection system. The training of the model relies on checking the key points on joints of the skeleton which is illustrated by Openpose. This means if the system cannot illustrate a complete skeleton, the probability of failing to detect falling or having false-positive results would be high. Despite that, the system still has a good performance when the webcam is set up in a place that can see the whole body of the person most of the time.

If there is any necessary need to have improvement on the system in the future, the simplest way would be using two webcams for double-checking or a 3D webcam. Since Openpose has been limiting the performance of the system, we think that the way to have a significant improvement to the system would be giving up Openpose and training the data in other ways. For example, for the method using optical flow, the accuracy rate must be higher.

References

A 2019 guide to Human Pose Estimation with Deep Learning
(https://nanonets.com/blog/human-pose-estimation-2d-guide/)

A 2019 guide to 3D Human Pose Estimation
(https://nanonets.com/blog/human-pose-estimation-3d-guide/)

CMU-Perceptual-Computing-Lab/openpose (GitHub)
(https://github.com/CMU-Perceptual-Computing-Lab/openpose)

reaktor/vzw-care-tf-pose-estimation (GitHub)
(https://github.com/reaktor/vzw-care-tf-pose-estimation)

Project Log