# loT Elderly Care Solution

PROJECT PLAN


Dec19-18
Optical Solutions
Daji Qiao
Robert Guetzlaff - Hardware
Tyler Borchert - Hardware and Testing Engineer
Siyuan Zeng-Test
Nidhi Dalvi- Meeting facilitator and hardware
Jared Griffin - Web App Engineer/GitLab Administrator
sddec19-18@iastate.edu
http://sddec19-18.sd.ece.iastate.edu


Revised: 2019-04-25

# Table of Contents

# List of Figures



# List of Tables

# 1 Introductory Material

## 1.1 Acknowledgement

Andrew Guillemette, the CEO of Optical Operations in Ames and our client, has helped significantly in guiding our group through this project. He has helped get us resources and provide us ideas when necessary. Daji Qiao, a professor in the ECpE department focusing on network engineering and our faculty advisor, has also given us welcomed advice on the ideas that we were presenting.

## 1.2 Problem Statement

The goal of this project is to make improvements and advancements of a previous capstone project. This project is intended for elderly individuals who wish to have their health monitored through the use of sensors placed around their residence [6]. This project is driven by the need for elderly care today. There is a large number of elderly people who live alone, so we need to monitor their life quality and health status to prevent health issues from arising. This is important for us to do because having a personal nurse is expensive, so we hope our project can provide a cheaper and more efficient way to achieve a cheaper and equally effective in-home healthcare.

## 1.3 Operating Environment

The sensing system will be housed within the elderly user's residence. The target audience of the product is elderly individuals who may not be interested in having visible sensors and wiring. To accommodate this, our sensors will need to blend in as much as possible with the environment and communicate wirelessly.

The database storing the user's data, the server acting as an interface to this database, the logic server, and the web application will be hosted in the Amazon Web Services (AWS) cloud environment. These components will require some level of security provided. However, as this is a prototype project, the standard protections AWS provides to systems hosted on its cloud will be sufficient.

## 1.4 Intended Users and Intended Uses

There are two intended users for this project: one being the elderly individual who interacts with the system in the kitchen, and the other being family members or another concerned party monitoring the behavior of the elderly individual.

The elderly individual will be a passive user of the system, interacting with their residence as normal, and we will be able to record what they do. This system will need to be as unobtrusive as possible as to not change the habits of this user. The system should be able to accurately monitor the eating habits of this user to predict if normal eating habits have been disrupted.

The other users of the system will interact with the web application. The user interface of the web application should be simple enough that multiple people could be tracked with little effort by one person but also provide the level of detail required to know what exactly triggered a health alert. This user could be a relative or an employee of an elderly living center.

### 1.5 Assumptions and Limitations

Assumptions:

- Sensors should be able to track activities accurately.
- The data recorded by the sensors should be passed to the server through an HTTP connection.
- The end product should be 1 sensor system per elderly individual.
- The web application should not have a maximum number of users.
- The product will not be used outside of the United States.
- Security will not be made a priority due to this being a prototype.

Limitations:

- The battery life for the sensors will last 1-2 years.
- The sensor will be wireless.
- The whole system should be indoor and requires Wi-Fi environment.
- The system must operate at 120 or 220 volts and 50 or 60 Hertz (the most common household voltages worldwide).

### 1.6 Expected end product and other deliverables

The final product will consist of a system of sensors that will be placed throughout an elderly person dwelling that will report information back to a server. This server will process the data to determine if this person is maintaining their regular routine. The server will analyze the sensor input to monitor information such as water consumption as well as whether the target person has eaten. This data will be visible on a website viewable by the caretakers or loved ones of the target.

What we will provide to the client is a set of wireless sensors that will last at least one year on a battery, a hub that will receive the data from the sensors and pass it off to the server, a server application that will process data from the sensors and identify changes in the targets routine, and a web application that will allow interested parties to see the conclusions generated by the logic server alongside the raw sensor data.

## 2 Proposed Approach and Statement of Work

### 2.1 Objective of the Task

This project has four goals to accomplish. The first goal is to fix the current wired solution to produce meaningful data. Accomplishing this goal puts our project on the correct course. The second goal is to redesign the sensors as a wireless solution. This will make installation of the sensors easier in the testing environment. The third goal is to create algorithms to implement predictive logic to interpret the data collected by our hardware. This algorithm will be put on the cloud to predict if the consumer completed activities in the kitchen area. The logic will catch anything that seems to be out of the ordinary and report it to the end user. Lastly, we will create a web application. This will present all the data collected in an easy to understand format.

## 2.2 Use Cases

There are two main use cases for our system: the elderly resident who will interact with the sensor system at their residence, and the elderly resident's loved ones or healthcare providers, who will interact with the web application. The sensor system sends data to the behavioral logic system, and the web application consumes the data the behavioral logic system produces. Below is a use case diagram that goes into further detail.


Figure 1: Use Case Diagram

## 2.3 Functional Requirements

- Fixing an old solution to produce correct data
    - Gain access to each Raspberry Pi & get the response and data collected by the Pi.
    - Check if the data is reasonable. If not, we need to check if the sensors are installed properly. And debug the code.
- Detect when the resident has missed a meal from data stored on the server (logic)
    - With the use of multiple sensor outputs, or lack thereof, to arrive at a reliable prediction.
    - Complete many test cases to ensure the logics reliability.
    - We need to recognize that the logic may not fit for everyone.
        - Machine learning. (Optional)
        - Data collecting and behavior record. (Optional)
- Create and install a wireless sensor (wireless solution)
    - Research current wireless sensors and wireless transmission technology
    - Identify the best solution based on:
        - Cost
        - Battery life
        - Data able to be recorded
        - Development time
    - Install new sensors
- The wireless solution must have a 1-2 year battery life span (wireless solution)
    - Ensure that the wireless solution researched fits this requirement.
- Display sensor event status through a web app (web app)
    - Write a web application to pull and display data from the server.

## 2.4 Constraints Considerations

The constraints for our project are:

- Cost- we will need to consider cost when moving forward in an attempt to make the most affordable end product in addition to the expense of hosting a cloud server with as minimal traffic as possible
- Ease of installation- Our project should be simple to install and add devices to so that someone who knows nothing about how it works can add monitoring to a home.
- The non-functional requirements we discussed are the following things:
- The app should be easy to use and navigate.
- Try to reuse the previous group's work.
- End door sensor solution should be wireless.
- Securely store data.
- Data gets analyzed in a timely manner.
- Nonintrusive.
- Installed easily.
- The system can handle an unexpected reboot.

These requirements are based on convenience, security, and efficiency. The users of our design are elderly people, so the design should be simple for the elderly to use. Since our project monitors the behavior of our users, we need to keep the data securely to guarantee their privacy and use the data in a just manner to avoid being intrusive. Also, since we have some work from the previous group, reusing these components can improve our working efficiency.

For the ethical standards, we follow three rules: protect users' privacy, predict risky situations that may happen to our users, and alert users to prevent risky situations to happen. Since this project is a prototype, we are not overly concerned about following these rules completely. However, we have made it clear to our client that this project should not be commercialized until these concerns are addressed.

## 2.5 Previous Work And Literature

There was a previous group working on this project last semester [1]. They created a prototype design for us to continue with. They also set up some sensors to record the time interval between opening and closing a cabinet door or drawer. The data can also be successfully sent to the database server and displayed on a mobile app.

The advantages of continuing to use this previous work are:

- Having a prototype that we can base future work on, which can reduce the amount of work we have to do
- It can serve as an evaluation tool for our future work

The shortcoming of following this previous work are:

- There are some parts not working properly in the previous work, so we have to redo these parts and fit the new parts into those working parts, which will be more difficult and time-consuming than designing a new one.
- Some works are not kept properly so we can't find them. So there is some information we want but can't be found.

Second, we also found a similar design project exist [2]. This project uses Raspberry  Pi and gateway to monitoring the elders' indoor location. We can refer to the communication between Raspberry Pi and server to implement our understanding, but since we are also monitoring the elders' behavior, this project doesn't contribute too much.

## 2.6 Proposed Design

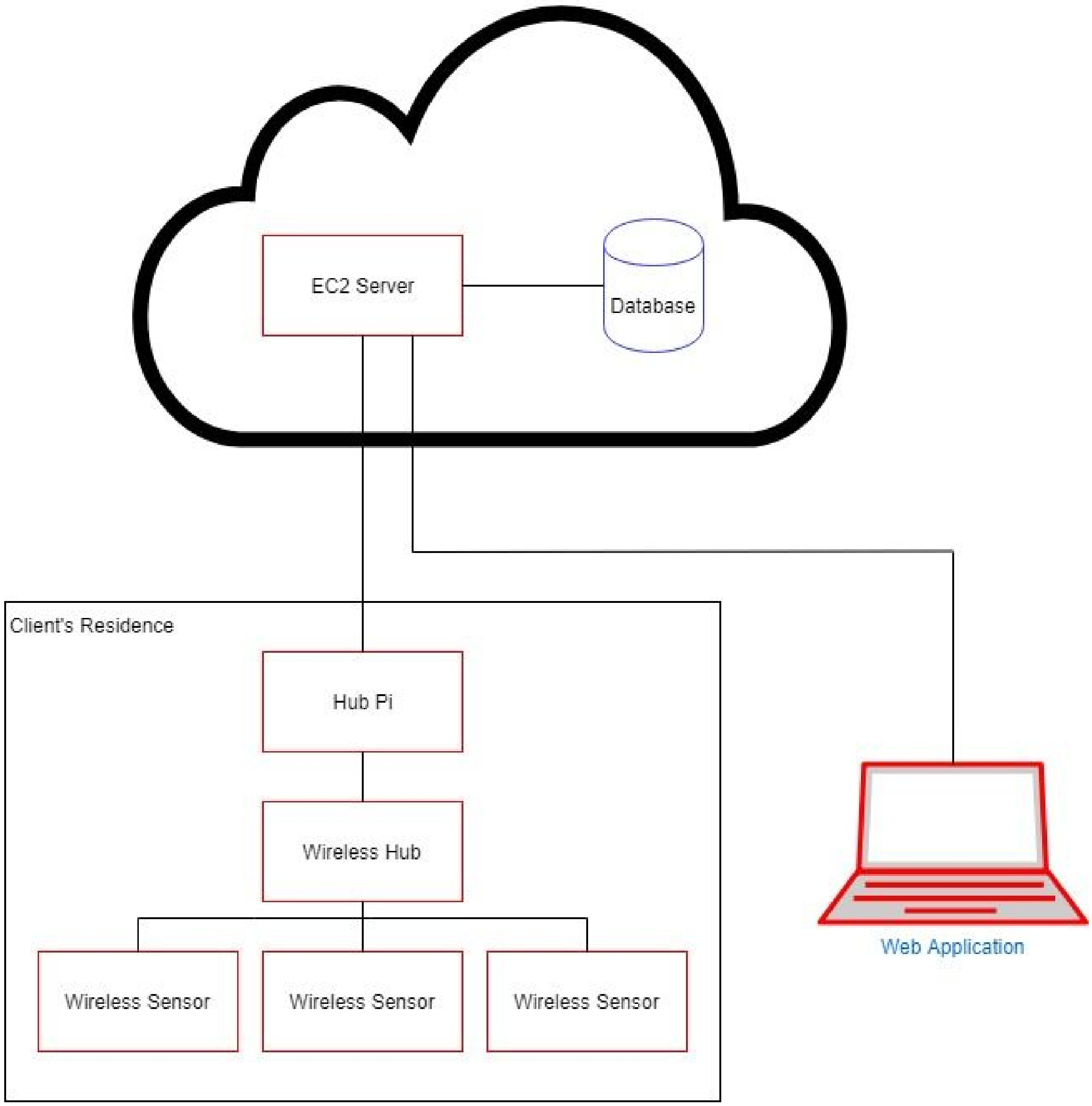


Figure 2: Block Diagram

We have three distinct design areas we will be working on, each of them will work together but be designed separately. The areas are: the sensors, the server logic, and the web application.

The sensors are physical door sensors that are placed throughout the test environment the controller that they are connected to. This controller relays back to the cloud any event data. Ten wired sensors with two controllers and one hub are currently deployed at the test environment by the previous group, however, their implementation currently throws incorrect data. The final goal for the sensors is to have them be wireless and only sending correct data. To this end find a wireless alternative to these sensors that will be more reliable and have a minimal impact on the test environments aesthetics. Alternatively, we will find the error with the current wired sensors and fix it.

The server will receive and analyze the data from the sensors to determine if the test person has completed daily tasks. Our first goal is to determine if our tester has missed a meal and if so, flag and report this data. This system will be run from an Amazon Web Services (AWS) account in the cloud and will integrate with our web app and SdMay19-36's Android application. The current alternative for this aspect is to have an onsite computer process the data to eliminate the risk of loss of internet.

The web application will provide a way for the tester to view sensor data as well and the analysis of the events. This interface will start out as our debug interface for the server logic.

## 2.7 Technology Considerations

The current setup we inherited from sddec18-14 uses magnetic sensors wired back to two Raspberry Pi Zeros, which, in turn, communicate back to a single Raspberry Pi 3 for the sensors, and an AWS EC2 instance for the database and the server. Upon taking over this project the Pi's had stopped communicating with the server. The loss of communication seems to the lineup with a reported power outage. This is our first consideration, the Raspberry Pi's run a version of Linux called Raspbian. While this is a stable and well known operating system at this time, an operating system adds additional areas for fault. In this case, since the devices were power cycled the program to record sensor data was not restarted. Physical access to the test environment may be required to fix specific faults.

The AWS server we are using for our data storage and processing is free for our purposes but requires an internet connection to receive data from the test environment. should internet go out, at best data transmission would be delayed. 4/5G failover and onsite data processing are current ways to overcome this problem. 4/5G failover would use the cellular network to transmit the data, with the drawbacks that cellular data is more expensive than traditional internet and the possibility that both systems could fail at the same time. Onsite processing of the data could be used to analyze events but requires a computer to be in every test environment, which violates our need to be as unobtrusive as possible. A combination of both of these methods could be used where we process the data if a connection cannot be made to the server, then only transmit over 4/5G if an event has been missed.

## 2.8 Safety Considerations

Nothing we are working with is inherently dangerous. Our logic triggering a false positive would only result in an extra check up on the tester. The only source of danger is our logic detecting a false negative and not creating an alert for a missed meal. As this is a prototype system, it should not be being relied on as the primary source of health information for our tester while we are developing it.

## 2.9 Task Approach

Our planned approach is: first, get the sensors working to the point where they stop giving us incorrect data. Then, with the sensor data, we will start building our logic system using the testers known activities coordinated with the data. Once we have our base logic we will test it against the data and improve the detection. Finally, we will implement our web app to show the user the data recorded by the sensors and the events detected by the logic.

## 2.10 Possible Risks And Risk Management

The largest risk to the development of our project is that all of our hardware is offsite. Our sensors and sensor controllers are at our test environment, which is a person's apartment, and our server code will run on a cloud server. Should something happen to our sensors or their controllers we have to coordinate with our tester to get physical access to them. Similarly, should AWS go offline we will lose access to our server and our data. This is a much smaller risk in our opinion, but it is something to be aware of.

## 2.11 Project Proposed Milestones and Evaluation Criteria

<u>Hardware Milestones:</u>

- The current sensors in place give off good data to the cloud to start logic.
    - Data is within reason, as in no impossible values.
    - Will be tested by numerous test cases
- Update the current solution that allows the production of good data
    - Physically push code to each of the Raspberry Pis in the testing environment.
- Researched the various type of wireless solutions and compile a list to present to the client.
    - Have enough information in order to accurately describe each proposed sensor.
- Implemented the wireless sensor into a prototype for testing.
    - Implement the wireless sensors into a given prototype to begin testing its functionality.
- Install the new wireless sensors into the testing environment.
    - Physically replace and install new wireless sensors into the testing environment.

<u>Logic Milestones:</u>

- Able to interpret the given data by hand.
    - Able to understand the data to predict by hand that the individual is eating or not eating.
- Plan and design for the logic algorithms.
    - Test and compare results with Bob's confirmed results.
- Verify and deploy the logic onto the server.

<u>Web Application Milestones:</u>

- Maintain feature parity with the Android application being built by another Senior Design group working with the same client

    - Compare application with another group to ensure functionality stays the same.
- Integrate and display the data generated by the behavior logic system
- Create a testing suite for unit, acceptance, and integration testing the web application
- Develop a continuous integration pipeline to deploy the web application on Amazon Web Services (AWS)

## 2.12 Project Tracking Procedures

Our group's progress will be tracked by our weekly reports and meetings. Our weekly reports will always mention what the group accomplished each week. While in each meeting we will discuss where we are in our project. This will help us keep on track for the entire semester. It is always helpful to know where we are as a group each week with these meetings.

## 2.13 Expected Results and Validation

The desired outcome is to have a functional testing environment where we can collect data on an individual and observe whether there are derivations in their usual daily habits. This involves having a functional sensor, proper logic, and a web application to view allocated data. The user will be able to view all this data that is collected on the said web application. At a high level, a user will be able to view the data that has been collected by the sensors. Also, they will be notified if usual daily habits have changed. If the user can complete these tasks, then we know that we have accomplished our goal.

## 2.14 Test Plan

<u>Hardware Testing:</u>

Wired Sensors

These sensors will go through tests conducted by the engineer. This will be done by using the prototype given to us by the client. Here we can run the code and test it physically. The success is determined by the data that is collected and sent off to the cloud. Some of these tests will need to be run over the course of a week or longer. This is because these sensors will be on 24/7. Seeing if there are edge cases over long periods of time are essential.

Wireless Sensors

These sensors will also be tested on the prototype. Since the wired solution is already present, we can directly compare the data that it is producing. It will go through the same tests as the wired solution. Directly comparing one solution to the old is the best way to tell of its success.

<u>Logic Testing:</u>

Code itself

The code will be tested with a provided data set. We were given a data set with an individual's habits over the course of a couple of weeks by the client. Here we can input the data to drive the code to make predictions. The code can be determined as a success if we can predict when the individual ate at the correct times as determined by the set of data.

Testing in designated Environment

The code will need to be tested with all the sensors in the environment it is designed for. Here we will have the data created over X amount of time and have that individual write down the times they ate. Then we compare the results after the testing period and evaluate the results.

Web Application Testing

Reliability Testing

Need to test to make sure that the application works in different cases. This can be set up by using the web application on different phones and different browsers.

Information Testing

The web application will be tested by opening it up to information. This is where we can see if data is getting represented correctly.

# 3 Project Timeline, Estimated Resources, and Challenges

## 3.1 Project Timeline

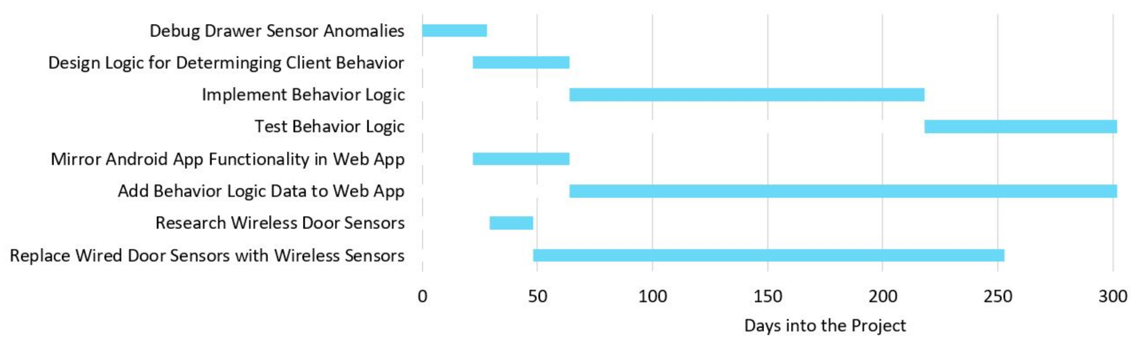


Figure 3: Gantt Chart

There are three major deliverables for this project: integrating a set of door sensors into the work a previous Senior Design group completed, designing an algorithm to take the data generated from the door sensors and predict the health of the elderly resident using the system, and creating a web-based application to display pertinent information. There have been some goals outlined for our group. Firstly, we need to devote the majority of our energy initially to resolving data anomalies from the system a prior Senior Design group created. Due to this, we have set this objective as the first one to complete. As we are wrapping up this task, our client has specified to begin working on the three other deliverables in parallel. There are some interdependencies between the three deliverables, so some of the tasks are staggered with this taken into account.

In order to finish the project on time, we have built in a large amount of time around each task. This will allow us to break the tasks up into smaller sub-tasks as we learn more about what each task will entail. Please note that the Gantt chart above does include time over the summer. We will not be working on the project during this time, but we have elected to include that time to better express the time the project will take.

## 3.2 Feasibility Assessment

We project that we will be reimplementing work once completed by a previous Senior Design group as well as implementing a web-based application for accessing data related to the logic we develop for sensing events related to the resident testing our project. Some of the challenges we will face are shifting requirements from our client since our client is a start-up company and they may have to shift their goals unexpectedly. We also are trying to build upon the work executed by a previous Senior Design group. As such, there will be challenges related to becoming familiar with what that group completed and making improvements to that work.

## 3.3 Personnel Effort Requirements

| Task | Description | Time Estimate |
|---|---|---|
| Debug Drawer Sensor Sensors Anomalies | Determine what is causing a data anomaly in the previous Senior Design group's work and resolve the issue. | ~ 20 hours |
| Design Logic for Determining Client Behavior | Create a set of relationships between how the elderly resident interacts with our sensors and signs that the individual's health is declining. | ~ 40 hours |
| Implement Behavior Logic | Take the relationships previously outlined for determining the health of the elderly resident and implement them on a server that can process the sensor data being recorded in the elderly resident's apartment. | ~ 80 hours |
| Test Behavior Logic | Create an acceptance testing suite to simulate the elderly resident's behavior in order to test that the logic system recognizes the correct behavior patterns. | ~ 40 hours |
| Mirror Android App Functionality in Web App | Take the functionality in a pre-existing Android application and reimplement it inside of a new web application. | ~ 40 hours |
| Add Behavior Logic Data to Web App | Integrate the data generated by the behavior logic server into the web | ~ 60 hours |

| | | |
|---|---|---|
| | application. | |
| Research Wireless Door Sensors | Research wireless door sensors technologies and how we could go about integrating this into our current system | ~ 40 hours |
| Replace Wired Door Sensors with Wireless | Reimplement the door sensing system using the wireless sensors previously researched | ~ 100 hours |

Table 1: Estimated time Requirement.

### 3.4 Resource Requirements

Identify the other resources aside from financial, such as parts and materials that are required to conduct the project.

| **Resource** | **Cost** |
|---|---|
| Raspberry Pi 3 | $0.00* |
| Raspberry Pi Zero | $0.00* |
| Wireless Door Sensors | ** |
| AWS EC2 Server [3] | $0.0208/hr |
| AWS RDS Database [4] | $0.017/hr |
| AWS S3 Storage Bucket [5] | $0.0237/GB |
| Onsite Server Computer | $0.00* |

Table 2: Estimated Resource Requirements

* The cost is zero because these products are already owned by the client

** The cost for the wireless sensors will not be known until research is complete in that subject

## 4 Closure Materials

### 4.1 Conclusion

For elders who live alone, careful attention needs to be paid to guarantee their safety and health. However, get a live-in nurse, or move to a nursing home to live out the rest of their days is expensive and inconvenient. Out project provides a new solution by using a series of sensors inside the home to remote monitoring the activities happened within the home and notice the elders' doctor/family member if there are potential risks that may happen.

Our solution will consist of sensors sending data to a local server. This server will perform any formatting or interpretation of the data, and data will be well analyzed to determine if good behaviors are taken by the users, or to determine if there are irregular behaviors happened.(For example, the user didn't wake up at the normal time or the user didn't eat lunch/ breakfast.) And then the server will send the results to the mobile App. The Mobile App will display the results textually or graphically and notify the user's doctor/family member.

Our team members have great understanding and experience in embedded system and programming so we have confidence for this project.

### 4.2 References


[1] http://sddec18-14.sd.ece.iastate.edu/.

team sddec18-14. "14 • IoT Passive Monitoring for Assisted Living Homes." sddec18-14, Team sddec18-14, 21 Dec. 2018, sddec18-14.sd.ece.iastate.edu/.

[2] IoT-aware AAL System for Elderly People.

L. Mainetti, L. Patrono, A. Secco and I. Sergi, "An IoT-aware AAL system for elderly people," 2016 International Multidisciplinary Conference on Computer and Energy Science (SpliTech), Split, 2016, pp. 1-6. doi: 10.1109/SpliTech.2016.7555929

[3] AWS EC2 Pricing: https://aws.amazon.com/ec2/pricing/on-demand/

Wikipedia contributors. "Amazon Elastic Compute Cloud." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 10 Mar. 2019. Web. 17 Apr. 2019.

[4] AWS RDS Pricing: https://aws.amazon.com/rds/mysql/pricing/

Wikipedia contributors. "Amazon Relational Database Service." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 19 Mar. 2019. Web. 17 Apr. 2019

[5] AWS S3 Pricing: https://aws.amazon.com/s3/pricing/

Wikipedia contributors. "Amazon S3." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 28 Mar. 2019. Web. 17 Apr. 2019.

[6] http://seniord.ece.iastate.edu/approved-projects-sddec19.pdf