FUNCTIONAL SPECIFICATION

Performing Ecological Momentary Assessments (EMA) via Picroft and mobile app

CA400

Santiago de Arribas de Renedo • 16449272
Pedro Marques Canes • 16770821

Date complete: 22/11/2019

Table of Contents

1. Introduction

1.1 Overview

1.1.1 Purpose

The system is composed of a Raspberry Pi running an open source home assistant called Picroft, our android mobile app, and our software called Daisy. The function of the system is to perform Ecological Momentary Assessments (EMA) via the Picroft or mobile app. An EMA is a method of conducting a survey in real time to when it is most valuable. This can range from commercial experiences such as consumer testing to behavioural analysis.

Another potential use of the system would be to store electronic Patient Reported Outcome (ePRO) entries from patients. The system would be an improvement on current ePRO methods as the data received from this system would be closer in real time to when the patient experiences certain effects from clinical trials. This would also reduce recall bias from patients.

1.1.2 Scope

The scope of this project will only focus on the app, Picroft and a server that is connected to Daisy. In the future the system should be able to be run on any home assistant with Daisy being hosted on a dedicated server.

The app and Picroft will send regular updates on certain information that can then be processed by Daisy to determine what is the current user situation and allow the system to act accordingly. This information can range from either microphone data to be processed (to detect if the user is in a social context) or GPS data from the app/ Picroft (to detect the users' proximity to the Picroft). Thus, the three main situations a user could be in would be as follows:

- The user is near the Picroft and alone. The Picroft is AVAILABLE for use.
- The user is near the Picroft and not alone. The Picroft is UNAVAILABLE and the app is AVAILABLE. **note: Daisy would be switching from the Picroft to the app in this situation due to privacy concerns as the user may be with people and not want to talk about sensitive information.**
- The user is not near the Picroft. The Picroft is UNAVAILABLE and the app is AVAILABLE.

Once Daisy has decided what the best device to use is to perform an EMA, it will send a request to the necessary device with a question pulled from a *questions* database and an API could potentially be used to send back data collected. This data can then be stored and processed later, in an *answers* database for a variety of applications including building a profile of the user.

The system is assuming that the user will always have their phone with them and that it is always on. It is also assuming that the home assistant is always powered on and has a stable internet connection.

1.2 Business Context

A use for this project would be to provide a method in clinical trials to receive ePRO data. Clinical Research Organisations are those in which the project could be deployed in. ePRO is a method in which to retrieve health outcomes of a clinical trial as reported by the patient – as opposed to a nurse or doctor. The benefit of ePRO is that the patients have a better experience over diary recorded entries, they are more likely to provide sensitive data and the overall cost of the data collection is reduced.

Chris Watson, PHD reports in his blog on patient clinical trial engagement via ePRO: *"When using electronic Clinical Outcome Assessments (eCOA) – especially electronic Patient Reported Outcome (ePRO) – for important patient data capture, compliance levels can reach well over 90% compared to as little as 11% with paper-based methods."*

1.3 Glossary

The following is a short list of novel terms that will appear in this document:

mycroft	An open source home assistant. Typically packaged as a physical product, this links to a dashboard where a user can decide what tasks they want it to perform.
Picroft	Mycroft software that can be uploaded to a Raspberry Pi with additional components for a cheaper solution.
skills	Tasks designed by the user for the Mycroft to execute.
daisy	The project software built that will decide when to perform the Ecological Momentary Assessments.
context detection	The process of determining the situation a user is in, either social or non-social. If the user is not alone (in a social situation), the survey will not be delivered through the voice assistant.

2. General Description

2.1 Product/ System Functions

The system is comprised of a single user, the Picroft, backend software (Daisy) connected to a server, and the mobile app. Each of these will provide their own functions to create a fully working system.

2.1.1 User Functionality

The user will interact with the system via the Picroft or mobile app at specific moments decided by Daisy. Picroft will issue questions aurally which the user can respond to vocally and have their data recorded to be processed later. The mobile app will use pop up notifications to record user responses via touch input. In both cases the user does not choose when to interact with the system.

2.1.2 Daisy Functionality

Daisy will be accessed from a Flask server during development and will connect to both the Picroft and mobile app via an API to retrieve data submitted by the user. Other data received will include the microphone data from the Picroft as well as GPS information from both the Picroft and mobile app. Daisy will analyse the GPS data to determine the proximity of the phone to the Picroft. Daisy will also analyse the microphone data to determine if the user is in a social situation.

A time interval will be set so that Daisy begins to decide asking questions when the user is most likely to be awake. E.g. Between sunrise and sunset. The Picroft and app will send updates (GPS or microphone info) synchronously, at specific intervals, which Daisy can check to make decisions on what situation the user is in.

To reiterate, the app and home assistant will send updates to Daisy at certain intervals for example every 15 minutes between sunrise and sunset.

Note: In a development environment, Daisy is run from a server on a separate machine to the Picroft, but both are on the same network to enable communication. However, during production, Daisy will either be run on a dedicated server or on the home assistant and send back data to an external database.

2.1.3 Picroft Functionality

The Picroft will send updates on its microphone and GPS data to Daisy for analysis. The Picroft uses **skills** to perform specific tasks such as setting an alarm or returning GPS data. These skills are part of the Mycroft community and are completely open source. The project will use custom skills to send data to Daisy.

If the Picroft receives a request from Daisy, it can then initialise the necessary skill to ask a user a question and return the answer to Daisy to be stored in a database and processed later.

2.1.4 App Functionality

The app, built for android smartphones, will send GPS updates to Daisy in sync with Picroft updates. As well as this, once the app receives a request from Daisy it will push a survey question via notification for the user to answer. This user submitted data will then be sent back to Daisy to be stored and potentially processed later.

2.2 User Characteristics and Objectives

The system can be targeted to two main classes of users: general and clinical trial patients. These users would most likely use the system as part of an external organisation. For example, a general user would not start the system themselves, they would use a typical home assistant and the system would be used to ask them survey questions and return data.

Another example would be clinical trial patients who use the system as part of their clinical trial but would not necessarily begin any interaction – that is decided by the clinical trial organisation.

General User
Uses any consumer grade home assistant such as Alexa. Expected to have little to moderate experience with technology. Should be able to understand how to use a home assistant and how to operate a smartphone. Will not be required to program any tasks or design survey questions.

Objective: The user should expect survey questions to be asked periodically by the system via their home assistant or mobile app. They should be able to answer these questions with little difficulty. These questions could be used to perform a mood analysis and potentially predict when is best to perform specific tasks.

Clinical Trial Patient
Much like a **general user**, this user will be expected to be able to use a home assistant and smartphone. They will not be required to initialise any interaction with either device. The activation of each device should depend on when the best time is to ask this user questions to receive the most beneficial data.

Objective: The user should expect clinical trial questions in order to perform ePRO via their home assistant or mobile app. They should be able to answer these questions with little difficulty.

Researcher
This user will expect survey questions to be asked according to some schedule. They will be able to choose what questions to ask a user and how often these will be asked.

Objective: The user should expect to be able to choose the survey questions and when these will be delivered, without much complication.

2.3 Operational Scenarios

The system is expected to work under almost any scenario the user is in provided they have their smartphone turned on and with them. The smartphone is used to determine the proximity of the user to the Picroft/ home assistant and as such Daisy can determine which device is most appropriate to use. The conditions a user might experience while using the system are as follows:

2.3.1 Use Case 1

Name:		Starting system for the first time.
Pre-conditions:	1 2	User has access to the home assistant. User has access to their smartphone.
Description:	1 2	User must connect the home assistant to internet and power. They need to download the app onto their smartphone.
Post-conditions:	1	The system is powered and ready to begin being used.

2.3.2 Use Case 2

Name:		Pairing home assistant to app.
Pre-conditions:	1	The system is ready to be connected up.
Description:	1 2	The home assistant will issue an IP address. The user must input this IP address into the app so that it knows which home assistant it needs to send data in sync with.
Post-conditions:	1 2 3	The home assistant is ready to begin sending data to Daisy. The app is ready to begin sending data to Daisy. Daisy receives both sets of data in sync and can process the data to make assessments on which devices to choose when it needs to ask a question.

2.3.3 Use Case 3

Name:		Getting asked a question on the home assistant.
Pre-conditions:	1 2	The system is fully powered and connected. Daisy has decided to send a question to the home assistant.
Description:	1	Daisy retrieves the question from the *questions* table in the database and sends it to the home assistant. The question is asked aurally.
Post-conditions:	1	A question has been asked via the home assistant.

2.3.4 Use Case 4

Name:		Answering a question on the home assistant.
Pre-conditions:	1	A question has been asked by the home assistant aurally.
Description:	1	The user answers a question asked by the home assistant. The answer doesn't have to be phrased in any way as the answer is stored in an *answers* database to be reviewed later.
Post-conditions:	1	The users answer is stored in a database.

2.3.5 Use Case 5

Name:		Getting asked a question on the app.
Pre-conditions:	1	The system is fully powered and connected.

	2	Daisy has decided to send a question to the app.
Description:	1	Daisy retrieves the question from the *questions* table in the database and sends it to the app. The question is asked via a push notification on the app.
Post-conditions:	1	A question has been asked on the app.

2.3.6 Use Case 6

Name:		Answering a question on the app.
Pre-conditions:	1	A question has been asked by the app via a push notification.
Description:	1	The user answers a question asked by the app. They select from a series of possible responses. This answer is sent back to the server to be stored in the answers database.
Post-conditions:	1	The users answer is stored in a database.

2.4 Constraints

Constraints will fall under two classes: *Development* and *End-User* with the former focusing on the constraints the developers will deal with and the latter on those affecting the user using the system.

2.4.1 Development Constraints

In developing this project there will be several constraints that will make it difficult to achieve a fully functioning system. As such, development should prioritise core functionality while keeping the system modular to allow for easy integration of additional features in the future.

This explains the schedule of developing the system with a focus on solving simpler problems such as networking and collecting data first and processing the data to perform predictions much later.

2.4.1.1 Time

Development of the core functionality of the system will need to be completed by May of 2020. This means that additional features will need to be assessed to check if they would be of benefit to the system and if so, would it be feasible to add them in before the deadline.

2.4.1.2 Hardware/ Software

The system is being tested on a Raspberry Pi, desktop computer and smartphone. Developing the system using all these devices can be troublesome because it will require having all devices working at the same time. As such an alternative should be considered. This could include using a virtual Raspberry Pi running the necessary software as well as an emulator for the mobile app.

All devices also need to be connected over a network in order to allow Daisy to process data from the Picroft and app and send the questions correctly.

The Mycroft software is being used on the Raspberry Pi. The project needs to be able to access the files that this software uses to call certain functions from the server. This may be difficult as the software code might be complicated to understand.

2.4.1.3 User Interaction

Testing the software is going to be difficult as it requires having all devices working at the same time. The system also relies on having the user use the system over a period consisting of at least 24 hours. This is because the developers need to check that the Picroft and mobile phone are

consistently sending data at regular intervals. They also need to check that Daisy is using this data to correctly assess which device to use and send questions appropriately.

The user should also be performing regular daily tasks such as going out to make sure the correct device is being used. The answers should be correctly stored in the database.

2.4.2 End-User Constraints

The end-user will deal with constraints concerning themselves or their environment. They will not have to deal with any of the technical constraints that the developer will have to deal with.

2.4.2.1 Time

The user should ideally use the system for a long enough period to collect useful data. This is to make sure there is enough data for any organisation that may find it beneficial. There may be complications if the user is unavailable from either their home assistant or smartphone for a long period of time.

2.4.2.2 Hardware/ Software

The system requires the use of a home assistant and reasonably up to date smartphone. The user should have access to electricity and a stable internet connection. The user also needs to download the app onto their smartphone and pair it to the Picroft. There may be constraints regarding storage space on the smartphone or processing power.

2.4.2.3 User Interaction

The design of the system means it can be interacted through the smartphone via the users' hands or through the home assistant via their voice. However, if the user is aphasic (they cannot speak) or if they don't have the use of their arms, then they cannot use the system properly.

3. Functional Requirements

Since the system is comprised of different components its requirements will concern multiple devices. This means that the overall functionality of the system does not depend on any one component working correctly but always on all components working together properly.

The functional requirements are listed in order of how the system needs to be operate. This means that the most important requirements are listed in descending order with most important being first.

3.1 Pairing the home assistant and app

Description:	Pair home assistant to phone app. Store this information in a database.
Criticality:	Daisy depends on knowing which phone and home assistant belong to which user. It can use this information to accurately judge the location of the user and its proximity to the home assistant.
Technical issues:	Network information needs to be provided to the app on which home assistant it needs to send data in sync with. This information also must be provided to the server so it can communicate with each device.
Dependencies with other requirements:	This requirement is dependent on the home assistant being powered on and connected to the internet. It also depends on the app being successfully installed on the smartphone device.

3.2 Connect all devices

Description:	All devices need to be connected over a network to allow communication and have necessary data sent where it is needed. E.g. GPS data from the home assistant and smartphone.
Criticality:	This requirement is necessary as all components in the system need to be properly connected or the system does not function correctly.
Technical issues:	Connecting all devices in the system may be difficult as they may each all be on separate networks. During development, these devices will be connected on the same network for ease of testing.
Dependencies with other requirements:	This requirement depends on the system being powered on and installed correctly.

3.3 Perform location check on devices

Description:	GPS information from app and home assistant updated on database. Daisy uses this info to judge the proximity of the user to the home assistant and decide which device to send questions to.
Criticality:	It is important for Daisy to have access to the GPS information from the home assistant and smartphone app to make a correct assessment of which device to use.
Technical issues:	The smartphone app and home assistant need to be connected to the server to send GPS data.
Dependencies with other requirements:	This requirement depends on having the users' smartphone and home assistant paired up correctly.

3.4 Perform context detection

Description:	Microphone data from the home assistant needs to be sent to the Daisy to allow it to perform context detection and determine if the user is in a social situation or not.
Criticality:	This requirement is important as without it, the system may send questions to the home assistant while the user is in a social situation which would be undesirable.
Technical issues:	The home assistant needs to be connected with the server correctly. Daisy also needs to be able to perform context detection correctly. This can either be done via digital software processing or by using Google's Awareness API.
Dependencies with other requirements:	This requirement depends on the system being fully connected and having all devices being able to communicate to each other especially the home assistant and the server.

3.5 Choose device

Description:	Daisy needs to be able to correctly assess the GPS and microphone data from the home assistant and smartphone and correctly choose which device to send questions to.
Criticality:	It is important that Daisy does this so that the user is always able to answer questions when it is required. This means that, for example, if they are away from the home assistant, the system can send questions to their smartphone and still collect data.
Technical issues:	GPS and microphone data need to be stored correctly and accessible by Daisy. Daisy also needs to do the appropriate processing to decide what device to use. This may difficult to program.
Dependencies with other requirements:	This requirement is dependent on the GPS and microphone data being correctly stored in the database.

3.6 Access and ask questions

Description:	Daisy needs to access questions and send these to the server to be sent to the appropriate device.
Criticality:	It is necessary to access and ask the correct questions in order to collect relevant user data.
Technical issues:	There may be difficulties with storing the questions in the database as well as accessing these and sending them to their appropriate device. There may also be an issue with posing the question properly to the user i.e. There may be limitations to what the home assistant can ask.
Dependencies with other requirements:	This requirement is dependent on the questions being stored properly in the database. It is also dependant on all devices being connected correctly to allow for questions to be sent to the appropriate device.

3.7 Retrieve and store answers

Description:	The system needs to collect user responses from their respective devices and store these properly in an answers database.
Criticality:	It is important that the data is properly collected and stored to be of use to any organisation that may find it useful.
Technical issues:	Collecting data may be an issue for the home assistant if the users' response is difficult to process. This could be due to environmental factors such as background noise. There may also be an issue concerning the database if the number of user responses collected is high.

Dependencies with other requirements:	This requirement is dependent on all devices being connected properly.

4. System Architecture

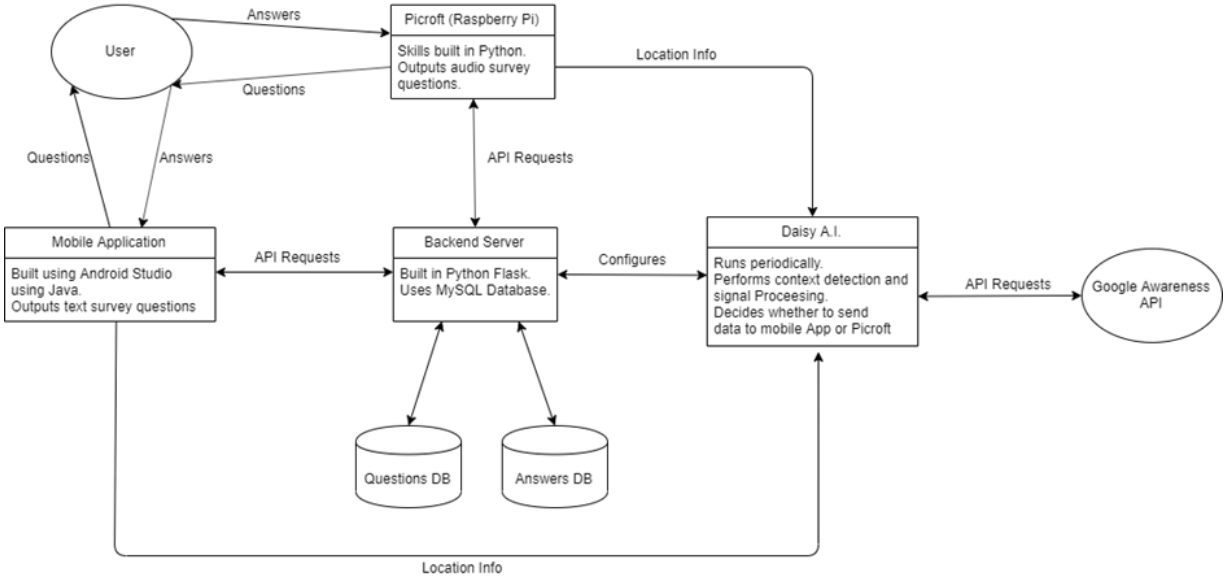

Fig 4.1 System Architecture Diagram

The system's backend server will be written in python using Flask. It will communicate with the Daisy Artificial Intelligence (AI) in order to decide which device to connect to. It will then send the data from the database to the chosen device.

The Daisy A.I. will be a scheduled task in the background of the server. It receives information from the Google Awareness API.

The Picroft and the Android mobile application will be the link from the user to the system. They will ask the users the information and get information in return, which will in turn be sent back to the server through API requests.

Both devices will send their location information to the Daisy API, which will run signal processing and decide which device to send the information to.

5. High-Level Design

5.1 Context Diagram

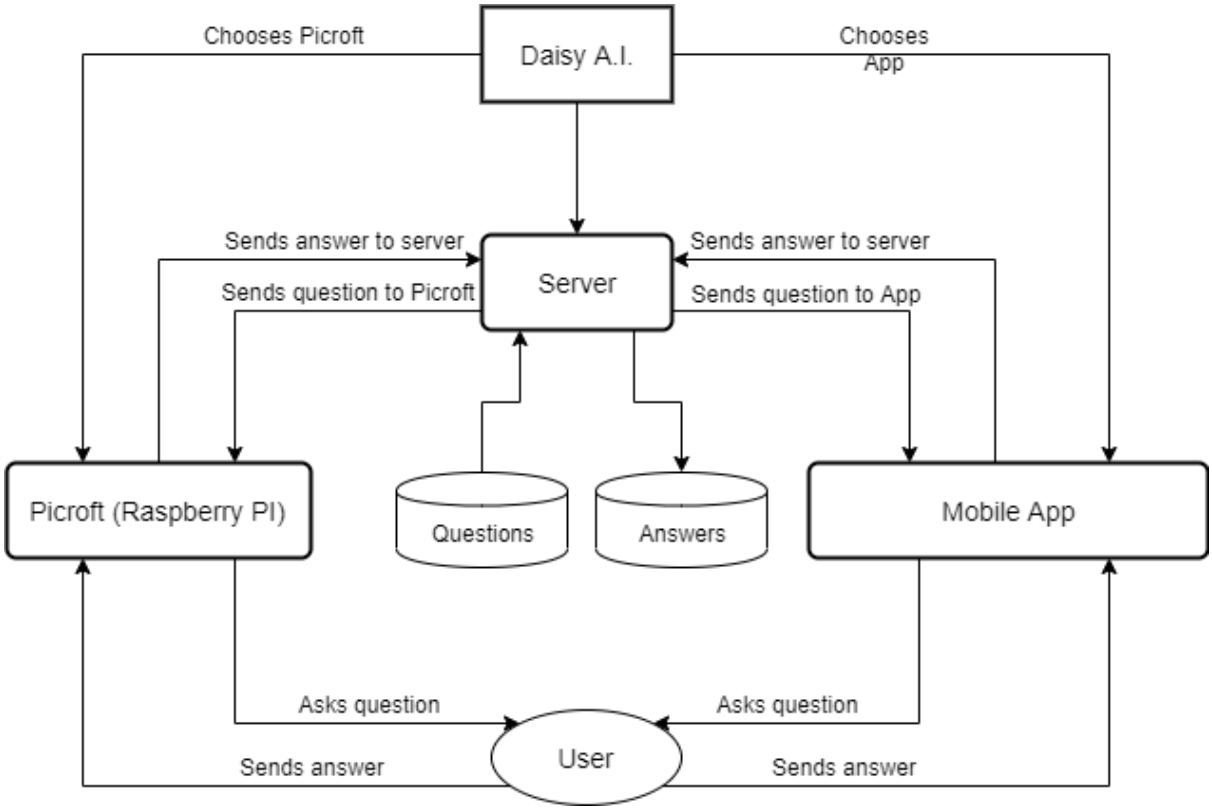

Fig 5.1 Context Diagram

The server will be the centre of operations. It will be the component responsible for linking all components together.

The Picroft and the Mobile Application will be the components responsible for communicating with the User.

Daisy will be the Artificial Intelligence responsible for deciding how to communicate with the user. The decision will be based on the components' GPS information and the Picrofts' microphone input.

A database will be used to store the questions to be asked and the answers received.

5.2 Sequence diagram

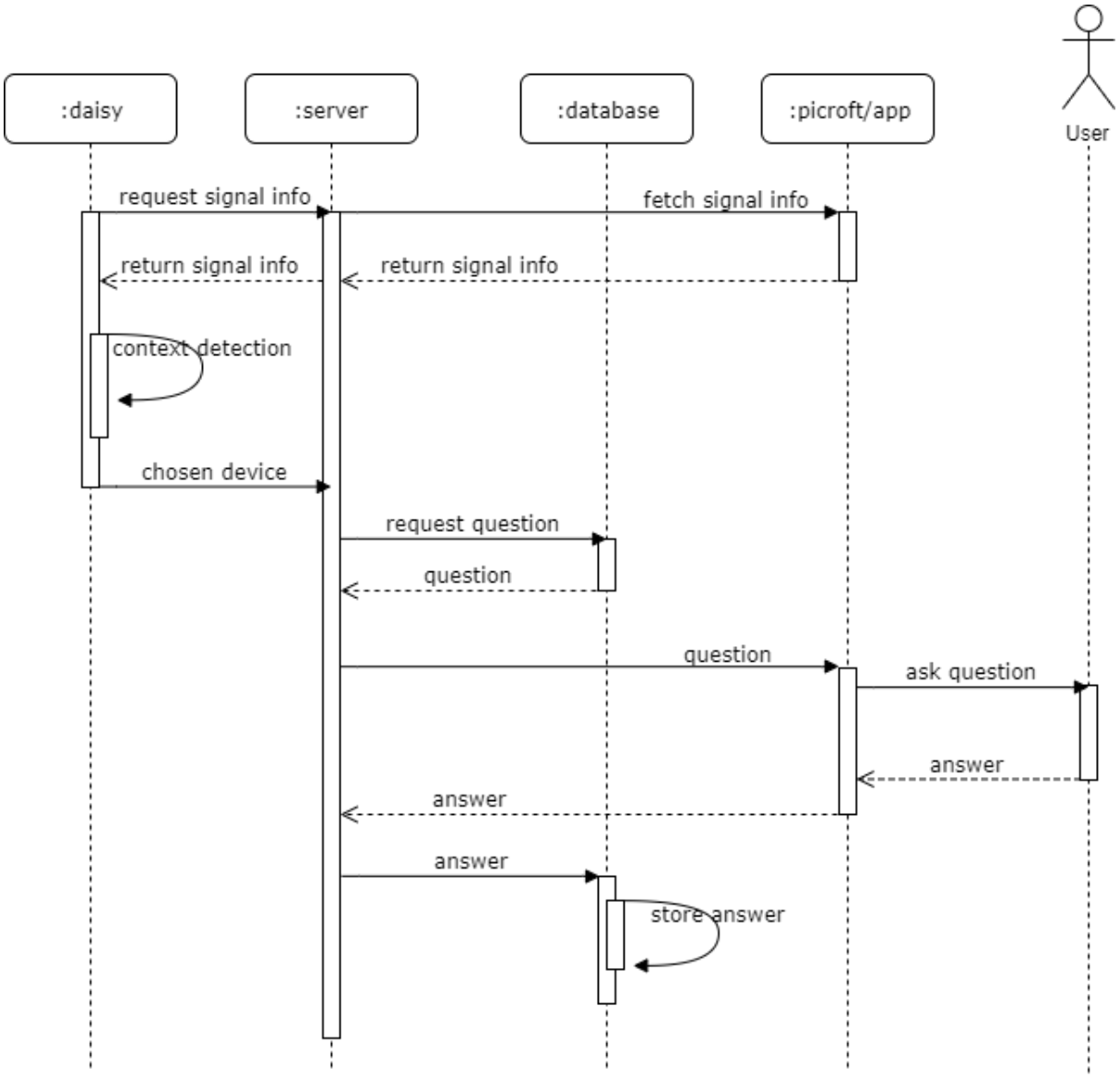

Fig 5.2 Sequence Diagram

This diagram shows the order in which the system would work. The objective of the project is to ask the user a survey question and store the answer on a database, hence the database is the endpoint.

5.3 Data-flow diagram of core functions

5.3.1 Getting asked a question

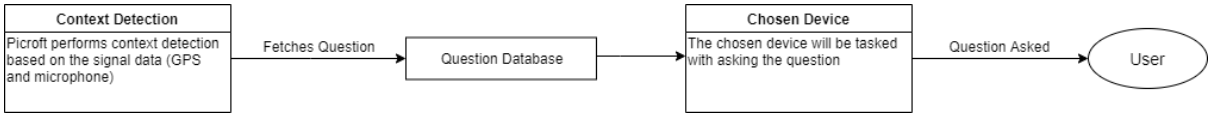

Fig 5.3.1 Core Function DFD 1

One of the core functions of the system is to ask the user a question. The first step in reaching the end goal is deciding which device to use to ask the question, and that is what happens during the context detection phase. Once the device is chosen, a question is fetched from a database and is sent to the chosen device. The device will then output the question either through the Picroft or through the application.

5.3.2 Answering a question

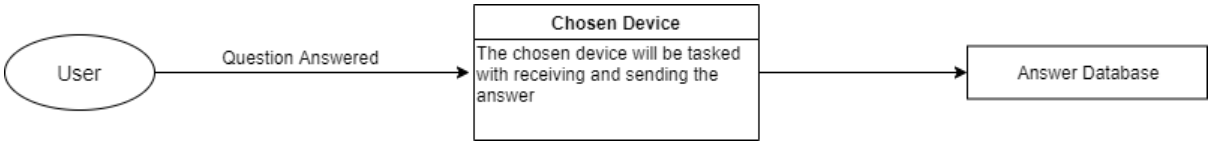

Fig 5.3.2 Core Function DFD 2

The second core function of the system is to receive the answer to the previously asked question. The user will reply to the question, either through the app or the Picroft (depending on which device was chosen). Once the device has received the answer, it will send the answer back to the server and it will then be stored in the database.

5.4 High-level design diagram - User

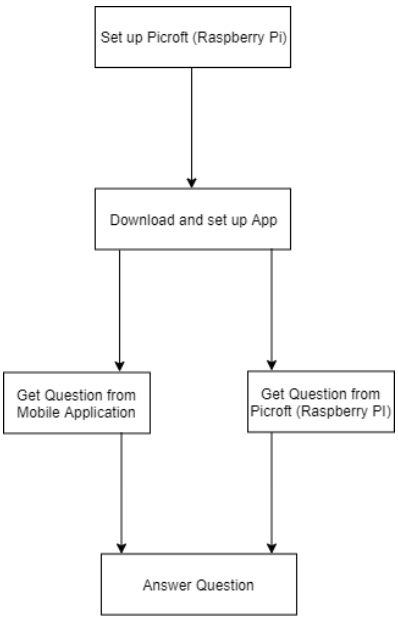

Fig 5.4 High-level design diagram

5.4.1 User diagram description

Set up Picroft (Raspberry Pi):

This is a one-time step. Once the Picroft is installed and set up, this step will no longer be necessary.

Download and set up the application:

This is another one-time step, which is only required at the start. The user will download the application and pair the phone to the Picroft.

Get question from device:

This is one of the previously mentioned core functions of the system. The user is asked a question. This can be done either by voice through the Picroft or through the mobile application with a multiple-choice answer.

Answer the question:

This is the other core function of the system. The user will reply to the device with the answer to the question. Depending on the what Daisy decides, the user will input the answer either through the phone or by voice through the Picroft.

6. Preliminary Schedule – Gantt Chart

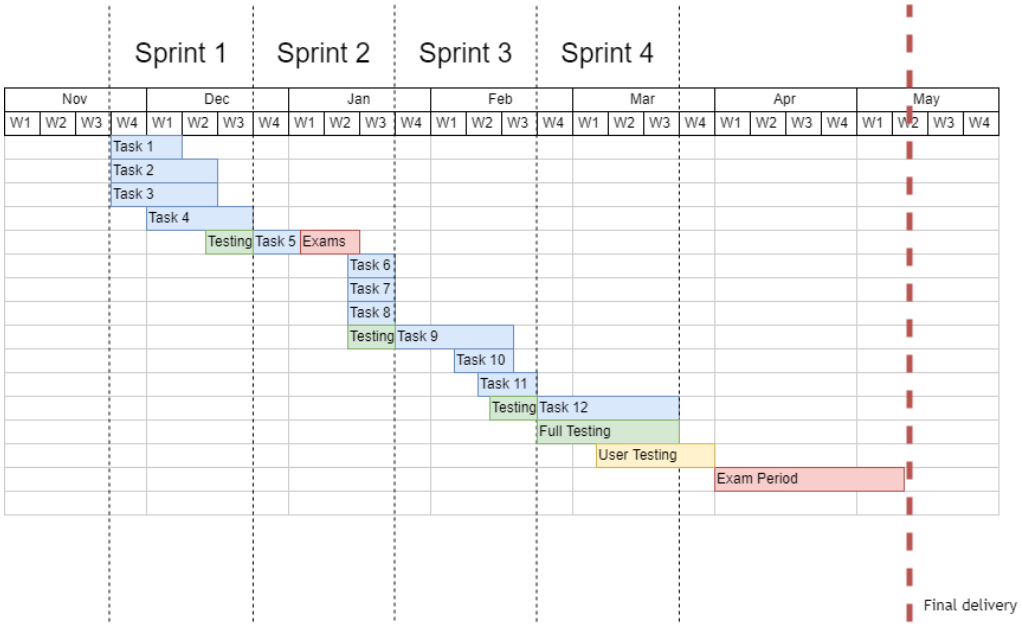

Fig 6 Gantt chart

- Task 1: Server setup. | Nov 25th – Dec 9th
- Task 2: Simple mobile application setup. | Nov 25th – Dec 16th
- Task 3: Communication between Daisy and Picroft (Raspberry Pi). | Nov 25th – Dec 16th
- Task 4: Communication between Daisy and mobile application. | Dec 2nd – Dec 23rd
- Task 5: Database setup. | Dec 23rd – Jan 3rd
- Task 6: Development of Picroft skills to return data to server. | Jan 17th – Jan 27th
- Task 7: Development of mobile application to return data to server. | Jan 17th – Jan 27th
- Task 8: Have Daisy process data and call appropriate devices. | Jan 17th – Jan 27th
- Task 9: Context detection development. | Jan 27th – Feb 13th
- Task 10: Develop questions to send from server to application/Picroft. | Feb 7th – Feb 17th
- Task 11: Collect data and store in database. | Feb 10th – Feb 21st
- Task 12: Fix any bugs or problems that arise and finalise project. | Feb 21st – Mar 23rd

The development of the project will be divided in different sprints (agile methodology), each being 4 weeks long. During the development time, tasks will be completed within the sprints with any uncompleted tasks being moved to the next sprint. The final sprint will be dedicated to finishing the project.

A whole month was assigned to it in order to account for any unfinished tasks and for the winter exam period. Each sprint will also have a period of testing allocated in order to test all the new additions to the project. There will also be a longer testing period at the end for a complete testing phase of the project. User testing will also be incorporated at the end of the development of the project.

7. Appendix

This appendix provides a more general list of terms that appear throughout this document.

survey	An investigation of the opinions or experience of a group of people based on a series of questions.
raspberry pi	The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV and uses a standard keyboard and mouse.
server	A computer or computer program which manages access to a centralized resource or service in a network.
database	A structured set of data held in a computer, especially one that is accessible in various ways.
Application Programming Interface (*API)*	A set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service.
sprint	Timeboxed iterations no longer than one month and most commonly two weeks where a team's goals can be completed.
agile (software development)	Agile is a process by which a team can manage a project by breaking it up into several stages and involving constant collaboration with stakeholders and continuous improvement and iteration at every stage. In the context of the project it will require constant testing and improvement of the system with a focus on the end goal.
bug	A software bug is an error, flaw or fault in a computer program or system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways.
Digital Signal Processing (DSP)	Digital signal processing and analog signal processing are subfields of signal processing. DSP applications include audio and speech processing, sonar, radar and other sensor array processing.

8. References

- Chris Watson, PhD (2018) Increase Clinical Trial Patient Engagement and Compliance with ePRO, Available at: https://www.ert.com/blog/increase-clinical-trial-patient-engagement-and-compliance-with-epro/ (Accessed: 13th November 2019).
- Carley Nolan () 4 Advantages of Using ePRO in the Clinical Trials Process, Available at: https://www.antidote.me/blog/4-advantages-of-using-epro-in-the-clinical-trials-process (Accessed: 13th November 2019).
- () Mycroft Home, Available at: https://mycroft.ai/ (Accessed: 2nd November 2019).
- () Mycroft AI, Available at: https://mycroft-ai.gitbook.io/docs/using-mycroft-ai/get-mycroft/Picroft (Accessed: 2nd November 2019).
- Prasanna R. Deshpande, Surulivel Rajan, B. Lakshmi Sudeepthi, and C. P. Abdul Nazir (2011) '', Patient-reported outcomes: A new era in clinical research, (), pp. [Online]. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3227331/ (Accessed: 4th November 2019).
- C Barbui, Policlinico G B Rossi, Piazzale L A Scuro (2011) '', EMA must improve the quality of its clinical trial reports, (), pp. [Online]. Available at: https://www.bmj.com/content/342/bmj.d2291.short (Accessed: 5th November 2019).