HEALTH CARE BOT

System Requirements

Project: Project Integration

Company: Saxion University Of Applied Science

Date: Enschede, 09th March 2019

Version: 0.1

Drawn up by: Hong Trinh 438443

Phuc Le 456061

Shyam Shiwakoti 426021

Minh Thanh Nguyen 444856

Minh Le 467475

Van Nguyen 444878

Supervisors: Ramazan Kirmali

Umit Guler

Client: Johan Stokkink

**Abbreviations**

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| FR | Functional Requirement |
| IQ | Installation Qualification |
| NA | Not Applicable |
| OQ | Operational Qualification |
| PQ | Performance Qualification |
| SR | System Requirements |
| TR | Technical Requirement |
| MTBF | Mean Time Between Failure |
| MTTR | Mean Time To Repair |

**Glossary**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Functional Requirement | Requirement related to the system functions (‘what?’) |
| Technical Requirement | Requirement related to the system design (‘How?’) |
|  |  |
|  |  |

**References**

|  |  |  |
| --- | --- | --- |
| **Number** | **Author(s)** | **Title** |
| [1] | J.W van Dijk | Methodological Design |
|  |  |  |
|  |  |  |

Table of contents

[**Background**](#_makihuuhr0s) **4**

[**Functional description and requirements**](#_ptigfh9g0qdp) **5**

[2.1 Functional description](#_5iugci1ha3be) 5

[2.1.1 Use Scenarios](#_w3y3uim0pqem) 5

[2.1.2 User Requirements](#_w3y3uim0pqem) 5

[2.1.3 Functions](#_w3y3uim0pqem) 5

[2.1.4 Functional architecture](#_w3y3uim0pqem) 6

[2.2 Functional requirements](#_njawgh2eck0s) 7

[2.2.1 Function 1: User interface](#_ci6tax2m5kqp) 7

[Login screen](#_75q6ua78w4zo) 7

[Main interface](#_7xmem01x54y6) 9

[2.2.2 Function 2: Control Functions](#_802pteoeiqqp) 12

[3.1 Technical requirements](#_zdh2s181iht4) 14

[3.1.1 Safety](#_c1orzexzaevw) 14

[3.1.2 Cost](#_c1orzexzaevw) 14

[3.1.3 Physical properties](#_c1orzexzaevw) 15

[3.1.4 Environmental](#_c1orzexzaevw) 16

[3.1.5 General design requirements](#_c1orzexzaevw) 16

[3.1.6 Mechanical design ( including dynamical and thermal requirements)](#_c1orzexzaevw) 17

[3.1.7 Electrical and electronic design](#_c1orzexzaevw) 17

[3.1.8 Software design requirements](#_c1orzexzaevw) 18

[3.1.9 Production and assembly](#_c1orzexzaevw) 19

[3.1.10 Reliability](#_6p6ycyz0aqt7) 19

[3.1.11 Maintainability](#_c1orzexzaevw) 19

[**Quality assurance Provisions**](#_egjje9hfm9o) **20**

[4.1 Verification definition](#_oq0c0blr0cqb) 20

[4.2 Verification setup 1: control](#_e5blf1jq29ii) 20

[4.2.1 Purpose of test/analysis](#_6qemradc31hj) 20

[4.2.2 Test setup/analysis method](#_6qemradc31hj) 20

[4.2.3 Test procedure/analysis](#_6qemradc31hj) 21

[4.2.4 Test/analysis data recorded](#_6qemradc31hj) 23

[4.2.5 Processing and evaluation of results](#_dsl1qzklc2o4) 24

# Background

In this project, the final product is building a smart healthcare bot. The project is conducted by group including EIE (Electrical Engineering) students and ACS (Apply Computer Science) students. The purpose of this project is gaining more knowledge about Echo Dot in independent hardware and how to implementation and use wireless radio system with the ESP32 as well as realizing data acquisition by sensors and graphical the user interface. The connective between hardware and software is important in this project. After the concept decision, the project continue with the first phase. System requirements of this project will be described in this document.

# Functional description and requirements

## 2.1 Functional description

### 2.1.1 Use Scenarios

A patient will be connected to a heartbeat monitor, then the doctor/nurse/patient will give command to the Alexa to measure the heartbeat. In case of emergency doctor/nurse will get notification on apps on mobile.If the doctor/nurse/patient wants to see the heart-rate of the patient later then it’s possible to get the data from a storage devices that save the data of the patient.

### 2.1.2 User Requirements

Connecting the patient will require a system that is easy to use. For live monitoring it will need a screen to display the heart rate on. Wearing the monitor 24 hours means we need a battery to sustain the 24 hours and a way to store the recorded data locally.Attaching and Detaching the patient with system, again should be easy and should not take long . Data access at a later time means, we have to store the data and we need log in system.

### 2.1.3 Functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | **Normal Operation** | **Faulty operation** | |
| Initialize | Startup the mobile device by Alexa( voice recognize) | Alexa cannot identify the mobile device | |
| Measure heartbeat | A sensor is measuring | | |
| User Interface | Display graphical data on PC and numerical data on the mobile device | A screen doesn’t display the graphical data on the mobile device. | |
| Send data | The heartbeat data is sent to a local server/internal memory when connect to Wifi. | The device cannot connect to the Wifi. | |
| Store data | Data will be stored in a database/sd card. | The heartbeat data is stored inside the internal memory (ROM/RAM) of the device. | |
| Receive data | Receive data without problem. | Receive a unexpected data from other mobile device. | |

### 2.1.4 Functional architecture

### 

The diagram depicts the function architecture of the product.

- At the first step, the customer can initialize the mobile part by Alexa (voice recognize). After that, the measurement can be performed and all the data will be displayed graphically and numerically on the PC/Smartphone. Beside, on the mobile part, there is a screen for patient to keep track easily.

- With the internet connection, the data can be upload to the database or cloud so a doctor can check it regularly. In case there isn’t a internet connection, the data can be (stored in the sd card and) displaying on the mobile part screen.

- In Addition, if there is no power supply (5V from micro usb port), the system will use the battery as a replacement.

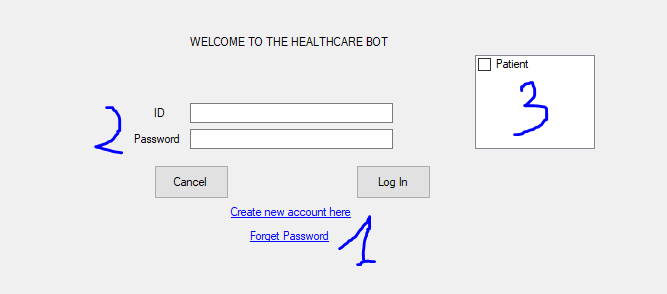
## 

## 2.2 Functional requirements

As the requirements of the client, the project result that will give out a mobility heartbeat measuring device. To keep track user’s health, a desktop user interface will be made for user include patients; specialist, who are doctors, nurses,... medical staffs in general, use in healthcare activities and contain two main parts: Login screen and main interface. Through the interface, user and medical staffs are able to access and see the result of the done measuring processes which should be saved and visible for users. Therefore, the user interface should contain these following functions which this chapter is going to describe about.

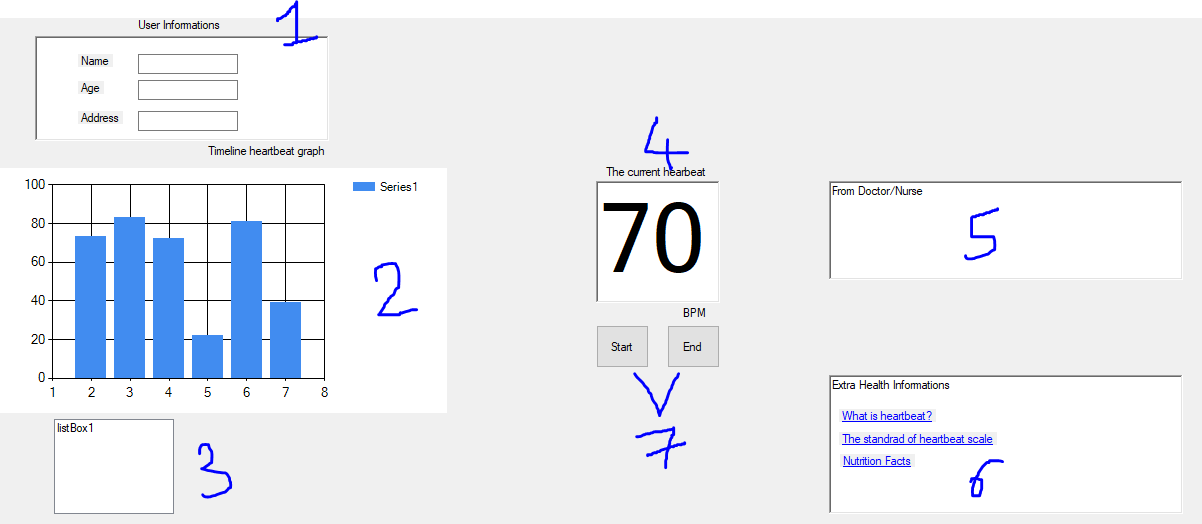
### 2.2.1 Function 1: User interface

### Login screen



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Components | Functions | Objects | Descriptions | Input | Output |
| FR01 | 1.Registering | Patient & medical staff | Patient can register for this application by using their patient ID number as ID and create their own password to make an account to use the app. Meanwhile, medical staff just use their staff ID as login ID and also create their own password. All of these account will be store in database. | New ID and password of login credentials | New login account |
| FR02 | 2.Logging in | Patient & medical staff | By using their account to as login credentials to fill in and access the app. | User’s created account | Main interface of application |
| FR03 | 3.Logging version | Patient & medical staff | By selecting a suitable category in a list which fit with the roles of users (2 main categories are patient and medical staff) | Database of users (by registering and sort by categories) | Suitable list of categories for user |

### Main interface



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Components | Functions | Objects | Descriptions | Input | Output |
| FR04 | 1.User interface | Patient & medical staff | Display information of logged user | Database of users (by registering and sort by categories) | User’s information are displayed |
| FR05 | 2.Timeline heartbeat graph | Patient & medical staff | Will be mainly used by medical staff as tracking patient health through the timeline heartbeat graph. This graph can visible the data of measurement of last recent time. Also, user can use this function as their own curious to see their heartbeat process. | Measuring heartbeat’s data which is stored in database | Visible graph for user |
| FR06 | 3.Timeline selection | Patient & medical staff | This small function allow user to choose which timeline should be shown in the graph by listing a small list of timeline: a day, 3 days, a week, 2 weeks, a month…  This is a small function to support the timeline graph function become more convenient. | Heartbeat measuring data which is stored in database | List of timeline for user choosing |
| FR07 | 4.The current heartbeat | Patient & medical staff | Patient and medical staff can easily witness the current heartbeat during the measuring procedure at that moment through a small heartbeat graph | Measuring heartbeat with online data transfer to database to be stored | Visible graph for user/Specific result at the moment when measuring procedure finished |
| FR08 | 5.Noting | Medical staff | This function allow medical staff give advices to their patients by noting to a small text box. By that way, patient when logging in to the app as a patient, the can receive the notification that come from medical staff’s message | From keyboard of computer, string text from medical staffs | String text for patient which content from medical staffs that show as a short text |
| FR09 | 6.Extra information | Patient | For those patients who have less or no idea about the scale of heartbeat, this function can help them as the extra information about heartbeat, what is good range? What is not good for health ?... whatever that related to health for patients, it can be found in this function and specially for patient’s uses | Built-in application linkable to source of information (websites,..) | Accessible link for user to look up health information |
| FR10 | 7.Start/End measuring | Patient & medical staff | Patients or medical staff can use this function for getting heartbeat data | Button to start measuring procedure of device | Starting a procedure and end a procedure |

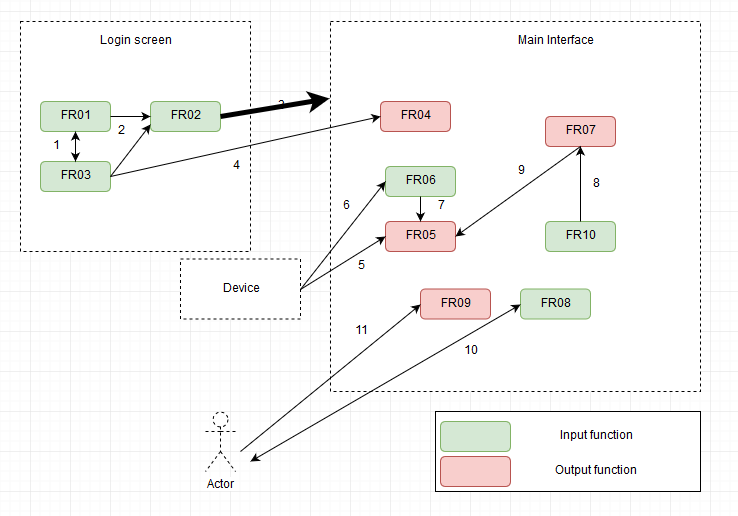
Basically, there are total ten functions for the GUI which can be seen that these function could be separated into 2 main groups : The input function that receive the input data and actions for output functions do its work as it was designed

|  |  |
| --- | --- |
| Input Functions | Output Functions |
| FR01, FR02, FR03, FR06, FR08, FR10 | FR05, FR07, FR09 |

### 

### 2.2.2 Function 2: Control Functions

According to the user interface, there are input and output functions which have relationships and support each other to make a good application for user which can be illustrated as the flowchart below



|  |  |
| --- | --- |
| FR01 | Registering function |
| FR02 | Logging in function |
| FR03 | Logging in version function |
| FR04 | User interface function |
| FR05 | Timeline heartbeat graph function |
| FR06 | Timeline selection |
| FR07 | Current heartbeat function |
| FR08 | Noting function |
| FR09 | Extra information function |
| FR10 | Start/End measuring |

In general, the login screen is the input for the main interface.

(1) Registering and logging version functions are the input to each other. Base on each user, the logging version must be picked correctly, then the registering will give out the account of that logging version. For instance, if user is a doctor, logging version will be selected as a doctor, then the created account will only work in the doctor version.

(2) The Logging in function take the input data from (2). In this case, account is created as the user version, all user have to do is fill in and press log in.

(3) Thank to logging in function, the GUI now change to the main interface where all main functions locate in.

(4) Where to display the user information. This function is as an output from the choosing logging version. For example, a patient logged in, so the information over there are belonged to that patient.

(5) The timeline graph is the output of all recent measuring process which performed by the device. The timeline selection is the one that decide the timeline graph (7). It also bases on the device (6) to know all finished measurements that it can possibly list out for user to choose to be displayed in the graph. Moreover, this graph can also display the current measuring process of the device. However, to receive those data, the start/end measuring process function will decide when the process start. When it start, the data can be shown as a specific number of heartbeat (8). Of course, the runtime data is also become input for the timeline graph (9).

Actor (medical staffs) can leave a message or a notification to patient by typing something on the given textbox (10).

(11) Actor (as patient) can get access to the given external information resource about health by actor (doctor/nurses...).

### 

### 

1. Design architecture and requirements

This chapter holds a small description of the system design technical requirements, “white box “method

is used since the internal structure/design/implementation to be used is clear to the engineers.

Some of the technical requirements are about an interpretation of the functional requirements, while

others are generated independently.

## 3.1 Technical requirements

### 3.1.1 Safety

* Safety is one of the most important feature that need to be insured in the project. The systems will be built with the EMC requirements, using CE safety medical standards. There will be the emergency alarm in the wristband, which will be buzzed when there is a problem with User’s heartbeat, it also have the fuse in order to disconnect the system whenever the trouble appears. The wristband will have the wood case to cover the electronic components inside and use the DC power supply, which is safe for human.
* In case the user need the tutorial about the system, there is the User’s manual that can give them the information in order to use the product.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number | Requirement | Relation | Value | Unit |
| TR0101 | EMC requirement | = | CE medical standards | - |
| TR0102 | Emergency Alarm | - | - | - |
| TR0103 | Fuse | - | - | - |
| TR0104 | Wristband case | = | Wood | - |
| TR0105 | Power supply | = | DC | - |
| TR0106 | User Manual | - | - | - |

### 3.1.2 Cost

* The cost is divided into 2 parts: Components and labour.
* For the components parts, there are some components that need to be supplied: Micro Controller, Display, Heartbeat sensor, Alarm, Echo Dot, Battery, Micro Controller USB…. There also have some components that the group need to be design: Input Amplifier, Micro controller USB, the wristband case…
* For the labour parts, each member in the group need the time in order to research the system, design PCB, build PCB, testing prototype, gathering information, programing, build wristband case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number | Requirement | Relation | Value | Unit |
| TR0201 | Micro Controller | = | Supplied | - |
| TR0202 | Display | = | Supplied | - |
| TR0203 | Sensor | = | Supplied | - |
| TR0204 | Alarm | = | Supplied | - |
| TR0205 | Battery | = | Supplied | - |
| TR0206 | Echo DOt | = | Supplied | - |
| TR0207 | Input Amplifier | = | Made | - |
| TR0208 | Micro controller PCB | = | Made | - |
| TR0209 | PCB design | ~ | 20 | Hours |
| TR0210 | PCB build | ~ | 7 | Hours |
| TR0211 | System research | ~ | 60 | Hours |
| TR0212 | Gathering information | ~ | 6 | Hours |
| TR0213 | Programing | ~ | 30 | Hours |
| TR0214 | Build wristband case | ~ | 5 | Hours |
| TR0215 | Testing | ~ | 20 | Hours |

### 3.1.3 Physical properties

* For the maximum dimensions, The wristband’s maximum dimensions is about 6-7cm in length and 3.5-4cm in width. The Alexa’s dimensions is 3.8 x 8.3 x 8.3 cm.

* For the maximum mass, the wristband weight’s around 0.5 kg, so it will not annoy the user because of their weight. The Alexa’s weights around 0.2kg

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number | Requirement | Relation | Value | Unit |
| TR0301 | Wristband dimensions | ~ | 7 x 4 | Cm |
| TR0302 | Alexa dimensions | = | 3.8 x 8.3 x 8.3 | Cm |
| TR0303 | Wristband mass | ~ | 0.5 | Kg |
| TR0304 | Alexa’s weight | ~ | 0.2 | Kg |
| TR0305 | Wristband mobility | = | True | - |
| TR0306 | Alexa mobility | = | True | - |
| TR0307 | Wristband color | = | Pink | - |

### 3.1.4 Environmental

* The operational and storage temperature for the system will be at 0-85. With this range, all the components in the wristband can work very well without any problems. On The order hand, the operating voltage will be 2.7-3.6V, this is the range for the micro controller can work with its best. Therefore, the power supply voltage can be used in range 3.0-3.3V. The operating current in average is 80mA and the power supply current is 500mA. The humidity for the system need to be lower than 80%.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number | Requirement | Relation | Value | Unit |
| TR0401 | Operational and Storage temperature range | = | 0-85 |  |
| TR0402 | Operating voltage | = | 2.7-3.6 | V |
| TR0403 | Operating current | = | 80 | mA |
| TR0404 | Power supply voltage | = | 3.0-3.3 | V |
| TR0405 | Power supply current | = | 500 | mA |
| TR0406 | System’s humidity | < | 80% | - |

### 

### 3.1.5 Mechanical design ( including dynamical and thermal requirements)

* For the Mechanical design, thermal and dynamical requirement have to follow application of principles about physicals , technical, and also materials science in other to analyse , design, construct or maintain machines or mechanical systems.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Requirement** | **Relation** | **Value** | **Unit** |
| TR0601 | Effect of humidity on wristband | = | True |  |
| TR0602 | Thermal conductivity of the materials | = | None |  |
| TR0603 | Electrical conductivity of the materials | = | None |  |
| TR0604 | Thermal expansion of material | = | None |  |
| TR0605 | Compatibility of the material on human skin | = | None |  |
| TR0606 | Stiffness of the wristband | < | 49.2 | MPa |
| TR0607 | Standardized major or minor components compliance with CE standards | = | True |  |

### 3.1.6 Electrical and electronic design

* For electrical and electronic design, PCB design have to follow rule and principle of EMC standard , the schematic should be indicated in detail with all information and statistic in other to achieve the final result of the product.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Requirement** | **Relation** | **Value** | **Unit** |
| TR0701 | Apply EMC standard for the design | **=** | True |  |
| TR0702 | Using Ultiboard on Matlab | **=** | True |  |
| TR0703 | Follow instruction when print and solder PCB layout | = | True |  |
| TR0704 | DC power supply | = | 5 | V |
| TR0705 | Minimum Installation Temperature of cable | > | -0 |  |
| TR0706 | Minimum Operating Temperature once installed of cable | > | -15 |  |
| TR0707 | Cable size required (5A max) | < | 0.5 |  |
| TR0708 | Conductor temperature | = | 85 |  |

### 3.1.7 Software design requirements

There will be many software requirements for this project.This Table shows the list of software Design requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number | Requirement | Relation | Value | Unit |
| TR0801 | Download Software | = | TRUE | - |
| TR0802 | Login And logout | = | TRUE | - |
| TR0803 | Python Programming Language | = | TRUE | - |
| TR0804 | Java Programming Language | = | TRUE | - |
| TR0805 | C# Programming Language | = | TRUE | - |
| TR0806 | MYSQL | = | TRUE | - |
| TR0807 | Time and Date | = | TRUE | - |
| TR0808 | Recovery Of Data loss | = | FALSE | - |
| TR0809 | Admin Interface | = | FALSE | - |
| TR0810 | Reset/Recovery Of login | = | FALSE | - |
| TR0811 | Code comment | = | TRUE | - |
| TR0812 | Encryption of Memory Card | = | TRUE | - |
| TR0813 | Standard software Development | = | TRUE | - |
| TR0814 | Test Phase | = | TRUE | - |
| TR0815 | Multiple Patient Option | = | TRUE | - |
| TR0816 | Multiple Doctors Option | = | FALSE | - |
| TR0817 | Alternative Backup Of Data | = | TRUE | - |

Notes: In case of Data recovery of loss data from both sd card and database we need more backup plans.

### 3.1.8 Production and assembly

### 

### 3.1.9 Reliability

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Requirement** | **Relation** | **Value** | **Unit** |
| TR01001 | Main Time Between Failure (MTBF) of the system | < | 2 | year |

### 3.1.10 Maintainability

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Requirement** | **Relation** | **Value** | **Unit** |
| TR01101 | Main Time To Repair (MTTR) of the system | < | 1 | day |

### 

# Quality assurance Provisions

## 4.1 Verification definition

The qualification assurance includes activities that determines the procedures and standards for the product. The quality assurance focuses on checklist, process standards, project audits, methodology and procedures for development. The aim of quality requirement is creating an environment to assure the product satisfies the quality requirements.

## 4.2 Verification setup 1: control

### 4.2.1 Purpose of test/analysis

The general purpose of testing in both hardware and software is:

* Increase quality of the upgrade
* Reduce risks to the product
* Meet the requirements of the users
* Determine capability of the product
* Define monitoring requirement in production

### 4.2.2 Test setup/analysis method

Software:

Functional testing:

|  |  |
| --- | --- |
| Unit testing | Shall be designed, implemented and executed to test each individual software components |
| Integration testing | After successful with unit test. Integrated together to perform specific tasks and activities |
| System testing | Testing the entire system for errors and bugs by interfacing the hardware and software components. Finally, testing whole system. |
| Acceptance testing | All the project requirements have been met and user have tested the system to ensure the product operates as expected and satisfies all requirements. |

Non-functional testing:

|  |  |
| --- | --- |
| Performance testing | Measuring how the system performs under the increasing load (number of users and data volumes) |
| Security testing | Test the software for confidentiality, integrity, authentication, availability. |
| Usability testing | As user aspect for using the software product. |
| Compatibility testing | The product or application is compatible with specified operating system, hardware platforms, web browsers, mobile devices,... |

### 4.2.3 Test procedure/analysis

The step of testing:

* Analyze the product

The information of product performs a vital function in testing. The product must to be learned thoroughly before testing it. The demand and expectation of clients and end users need to know clearly.

Who will use the healthcare bot?\_Patient and doctor/nurse

What is it use for?\_hearbeat measurement and alarm in emergency

How will it work?

What are software/hardware the product uses? mySQL for database, Micropython for ESP32, C# for GUI

* Design test strategy

**Scope of testing:** As the software requirement specs, the project focuses on all functions, the stress or overload of a database, performance.

**Testing type:** unit test, integration test, system test

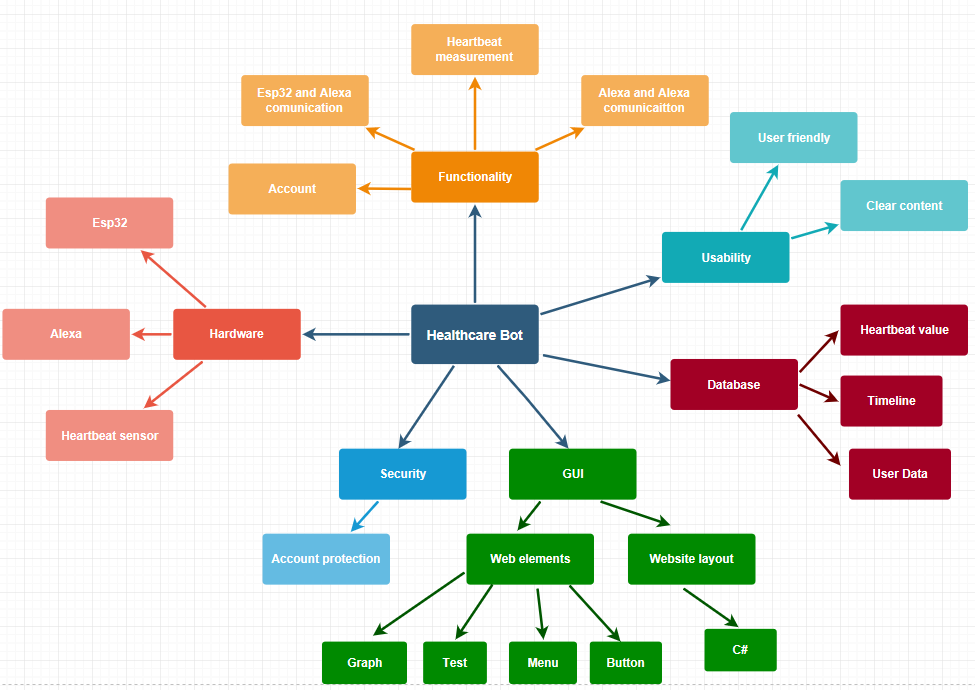
**Risk and Issues:**

|  |  |
| --- | --- |
| **Risk** | **Issue** |
| Team member lack the required skills for esp32 | Research and study how to implement with ESP32 |
| A lack of cooperation in group | Remind team member in their task |

**Test logistics**

Each member of the team will involve in testing after finishing

* Define test objectives



* Define test criteria

Planning the test cases and assure all the test has record and pass.

* Resource planning

**Human resource**

|  |  |
| --- | --- |
| **Members** | **Task** |
| Test Manager | Define project direction  Acquire appropriate resources |
| Tester | Identifying and describing test techniques, tools,..  Verify and assess the test approach  Execute and report the test |
| Developer in test | Implement the test cases, test program, test suite |
| QA member | Take in charge of quality assurance |

**Equipment**

|  |  |
| --- | --- |
| **Resources** | **Descriptions** |
| Server | Web server, database server and application server |
| Network | LAN, Wifi, Bluetooth |
| Computer | Operating system, local server |

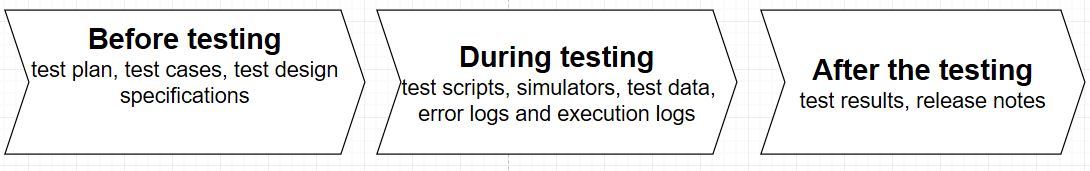
* Plan test environment

In each phase, the test environment build by a strong cooperation between test and development:



* Schedule and estimation (ACS)

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Members | Goals | Time consump (Estimate) |
| Test esp32 with LEDs | Hong Trinh | Control LEDs with esp32 making sure that esp32 Chip working well | 1 days |
| Test esp32 with serial communication | Phuc Le | Control the esp32 with serial connection for transmitting and receiving data functionality | 5 days |
| Test esp32 with wifi or bluetooth | Shyam | Be able to connect device to network by wifi or bluetooth for controlling from distance | 5 days |
| Test esp32 connecting with Alexa | Shyam, Phuc Le, Hong Trinh | By using Alexa to giving order and check if the esp32 can response to the Alexe command ? | 1 week |
| Test database | Shyam, Phuc Le, Hong Trinh | Is the data too large?  The period to store data is right? | 4 days |
| Test web server, GUI | Shyam, Phuc Le, Hong Trinh | Ensure the UI being friendly with user | 3 days |

Determine test deliverables

### 4.2.4 Test/analysis data recorded

To make sure testing and analysing procedure work effectively, each of test or analysis should be recorded by a small report. In each report will have the finished test/analysis with detail test script descriptions, test data, simulation description,.. From that point, the goals of the test/analysis are clear enough and the team can decide the result of that test/analysis with the error logs and execution logs which can provide detail result of the test/analysis

* Input data: built-in test function (debug function, built-in function in IDE), components functions (esp32, heartbeat sensor), device functions (measuring heartbeat, display, notification), application function (logging in, display data, notification)
* Output data: functionality of components, device, GUI (work - not work), testing code functions, ( pass - fail), executing code (run - error)

### 4.2.5 Processing and evaluation of results

Basing on the test/analysis data recorded, all tests/analyses will be saved with clear input and output. From that point, the team can be easy to dig deep into the problems of the test or the analysis as well.

During the testing/analysing process, the result can be considered as a success of fail which depend on the result of the project

* Coding: executing, running without error.
* Application: all function of the GUI work and meet requirements of the client: can create new account to log in, easy to use, visible all necessary data, be able to send-receive notification.
* Component functionality: the device can measure the heartbeat, built-in display of device work, be able to send-receive notification to application.

### 