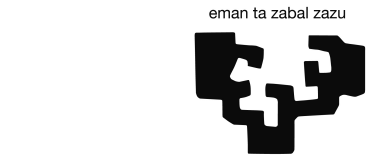


Informatika Ingeniaritzako Gradua

Konputagailuen Ingeniaritza

Gradu Amaierako Proiektua

Datu fisiologikoak eskuratzeko esperimentuak diseinatzeko sistemaren garapena: Datuak jasotzeko modulua

---

Egilea

Ezekiel Sarasua

Zuzendariak

Nestor Garay eta Edurne Larraza

2016ko Iraila

Laburpena

Proiektu honetan datu fisiologikoak eskuratzeko esperimentuak diseinatzeko sistemaren garapena egin da, bi modulutan banatzen dena: esperimentuak diseinatzeko modulua eta datu fisiologikoen eskuraketa modulua.

GAP hau, janzteko sentsoreen bidezko datu fisiologikoen eskuraketa modulua garatzeaz arduratu da. Horretarako Informatika Fakultatean aurkitzen diren bi txartelekin batera, *e-Health* eta *BITalino*, hainbat sentsore fisiologiko erabili dira: elektromiografoa, elektrokardiografoa, azelerometroa, pultsioximetroa, etab. Sentsoreetatik jasotako datuen bidalketa, proiektuan garatu den *Android* aplikazioaren bitartez egin da.

Hainbat izan daitezke beste partaidearekin batera garatu den sistemaren aplikazioak ikertzaile batek urruneko erabiltzaile baten datu fisiologikoak eskuratzearen inguruan aurretik definitutako esperimentu baten baitan: ikerkuntza, medikuntza edo kirola.

Resumen

En este proyecto se ha desarrollado un sistema de diseño de experimentos orientados a la extracción de datos fisiológicos. El proyecto se encuentra dividido en dos módulos: el módulo de diseño de experimentos y el módulo de extracción de datos.

Este PFG esta dedicado a la extracción de datos fisiológicos mediante el uso de sensores vestibles. Para ello, se han utilizado dos microcontroladores que hay en la Facultad de Informática, *e-Health* y *BITalino*. Estas dos plataformas, hacen posible la extracción de datos fisiológicos por medio de diferentes sensores: electrocardiógrafo, electromiógrafo, acelerómetro, pulsioxímetro, etc. Por otra parte, el posterior envío de los datos extraidos se realizara a través de una aplicación *Android* desarrollada en el mismo proyecto.

Pueden ser varias las aplicaciones del sistema desarrollado, en el ámbito de obtener remotamente los datos fisiológicos de un determinado usuario mediante experimentos previamente definidos: investigación, medicina o deporte.

Abstract

In this project a experiment design system that allows physiological data extraction has been developed. The project is divided in two modules: the experiment design module and the physiological data extraction module.

This DFP is dedicated to extracting physiological data by using wearable sensors. To this end, two microcontrollers which are at the Faculty of Computer Science, *e-Health* and *BITalino*, have been used. These two platforms allow the extraction of physiological data using different sensors: electrocardiograph, electromyograph, accelerometer, pulsioximeter, etc. Moreover, the subsequent delivery of the extracted data will be made through an *Android* application developed inside the project.

There are several applications for the developed system when offering the researcher the possibility to remotely obtain physiological data of a user by means of predefined experiments: research, medicine or sport.

Gaien aurkibidea

Irudien aurkibidea

Taulen aurkibidea

1. KAPITULUA

Sarrera

1.1 Motibazioa

Informatika asko eraldatu da azken urteetan, mahaigaineko ordenagailuetatik hasita nonahiko konputaziora iritsi arte. Gaur egungo teknologi berriei esker, azken urteetan seinale fisiologikoen erabilera asko zabaldu da. Sentsoreak geroz eta hedatuagoak eta txikiagoak dira, eta, ondorioz, elkarrekintza bide berriak irekitzen ari dira eta baita lor daitezkeen datuak. Elkarrekintza bide hauek janzteko gailuen (*wearable devices*) bidez lor ditzakegu.

Proiektu honetan datu fisiologikoak eskuratuz esperimentuak diseinatzeko sistema orokor bat diseinatu eta garatu nahi da, erabiltzaileen portaeran eta errendimenduan zerikusia izan dezaketen parametroak finkatu eta etorkizunean aztertu asmoz. Sistemaren helburu nagusia ikertzaileei baliabideak eta erremintak eskaintzea da, esperimentuak aurrera eramateko prozesu osoa (hauen diseinutik emaitzen analisiko fasera arte) errazteko asmoz.

Sistemaren helburua, janzteko gailuen mugikortasuna baliatuz, datu fisiologikoak ingurune erreal batean eskuratzea da. Kontuan izan behar da, jasotako datuak laborategi batean eskuratutakoak bezain zehatzak ez izan arren, ikertzaileentzat erabilgarriak izan daitezkeela, beraien naturaltasuna hobeagoa izango delakoan.

1.2 Deskribapen orokorra

Garatutako sistemak janzteko sentsore batzuen bidez pertsona baten datu fisiologikoak jaso eta zerbitzarira bidaliko ditu. Bi erabiltzaile mota identifikatu dira, alde batetik sentsoreak jantziko dituzten pertsonak (partehartzaileak edo subjektu esperimentalak) eta, bestetik, datu horiek eskuratu eta analizatuko dituzten pertsonak (ikertzaileak). Gainera, sistema ez dago mugatua erabiltzaile bakarrera, adibidez, hainbat ikertzaile egon daitezke. Zerbitzari bat egongo da sisteman partehartzaileen datuak jaso, gorde eta hauek atzitzeko eskaerak zerbitzatuko dituenak.

Sistemak bezero/zerbitzari egitura jarraitzen du, bi bezero mota desberdin daudelarik, eta ondorengo modulutan dago banatuta:

- Bezeroa:
  - Ikertzailea:
    Ikertzaileak esperimentuak diseinatzeko web aplikazio (bezero aldea) bat erabiliko du. Web aplikazio honen bidez ere, zerbitzariak biltegiratutako fitxategiak eskuratuko ditu eta aztertu ahal izango ditu.
  - Partehartzailea:
    Pertsona batek hainbat sentsore fisiologiko jantziko ditu eta lortutako datuak *Bluetooth* bidez helaraziko zaizkio bere *Android* mugikorrera. Ondoren, jasotako informazioa Internet bidez bidaliko zaio zerbitzariari, gero ikertzaileari datuak analizatzeko aukera emanez.
- Zerbitzaria:
  - Esperimentuen diseinua:
    Ikertzaileari esperimentuak diseinatzea ahalbidetuko dion web aplikazioa (zerbitzari aldea).
  - Esperimentuen emaitzen kudeaketa:
    Android aplikazioaren bidez bidalitako datuak jasotzeaz web zerbitzu bat arduratuko da. Informazioaren transmisioan segurtasuna bermatuko da protokolo seguruak erabiliz.
  - Lortutako datuen biltegiraketa:
    Jasotako datu fisiologikoen fitxategiak zerbitzarian biltegiratuko dira, *HDF5*

formatuan. Dena den, etorkizuneko lan bezala, laino sistema batera pasatzeko asmoa dago.

Sistema osoa ondorengo gailuz osatuta egongo da:

- Sentsore fisiologikoak eta hauek kudeatzeko txartela.
- Partehartzaile bakoitzeko *Android* mugikor bat sistemaren aplikazioarekin.
- Konputagailu bat, web zerbitzari bat instalatuta izango duena. Datuen transmisioaz arduratuko den web zerbitzuarekin eta ikertzaileari sarrera eskainiko dion web aplikazioarekin.
- Ikertzailearen konputagailua web aplikaziora konektatzeko eta esperimentuak kudeatzeko.

Proiektuaren irismena zabala denez, lana bi GAP-en artean banatuko da. Ezekiel Sarasua ikasleak subjektu esperimentalen elkarrekintzaren aldeko garapenaz arduratuko da, hau da, sentsore fisiologikoak eta hauek konektatzeko plakaren nondik norakoetan eta *Android* aplikazioaren garapenean. Maider Simón ikasleak, aldiz, zerbitzari aldeko garapenean egingo du lan: zerbitzariaren konfigurazioan, *API*-aren eta web aplikazioaren garapenean eta jasotako seinaleen biltegiraketan, hain zuzen ere. Bestalde, bi ikasleen artean sistema osoaren diseinua eta moduluen arteko komunikazio protokoloa egingo da.

GAP honetan subjetu esperimentalaren aldea burutzeko egin diren ikerketak eta garapena azalduko dira, proiektuaren atal komunekin batera. Beharrezkoa den ataletan ere beste GAP-eko nondik norakoak aipatuko dira.

1.3 Memoriaren egitura

Memoria honek proiektuaren nondik norakoak azaltzen ditu, hainbat kapituluetan antolatuta. 2. kapituluan Proiektuaren Helburuen Dokumentua aurki daiteke, proiektuaren plangintza eta kudeaketa azaltzen duelarik. 3. kapituluan landutako prestakuntza adierazten da eta 4. kapituluan, garapena.

Azkenik, behin proiektua garapenarekin amaitu denean, ateratako ondorioak eta etorkizunean egin daitezkeen hobekuntzak jasotzen dira 5. kapituluan. Dokumentu honen eranskinetan, garatutako sistemaren erabilpen gida eta zuzendari eta zuzendarikidearekin egin-

dako bileren aktak bildu dira. Bukaeran UCAmI-2016 [2] kongresurako prestatutako eta bidalitako artikulua gehitu da.

2. KAPITULUA

Proiektuaren Helburuen Dokumentua

Proiektuaren Helburuen Dokumentu honetan garatutako sistemaren deskribapena eta hau aurrera eramateko egin den plangintza azaltzen dira. Irismenaz gain, lanaren deskonposaketa egitura eta definitu diren atazak zerrendatu dira. Gainera, proiektuan zehar egingo den lana islatuko duen kronograma eta dedikazio aurreikuspena aurki daitezke. Segidan, komunikazio, kalitate eta arriskuen planak, eskuraketen kudeaketa eta lan metodologia zehaztu dira. Bukatzeko, benetako denbora zehazten da eta zeintzuk izan diren denboren desbiderapenaren arrazoien azterketa.

2.1 Irismena

Sistemak oso egokia den bezero/zerbitzari egitura bat jarraituko du. Esan bezala, bi bezero mota izango dira. Entitateen arteko komunikazioa bi noranzkoetan emango da, zerbitzaria dela medio; ikertzaileak subjektu esperimentalaren datu fisiologiko konkretu batzuk nahi izanez gero, esperimentua diseinatzeko fasean zehaztu beharko ditu garatuko den web aplikazioa erabiliz. Beste aldetik, partehartzaileak bere datu fisiologikoak (sentsoreak jantzita) zerbitzariari bidaliko dizkio, *Android* aplikazio baten bitartez.

Bezeroak hainbat sentsore fisiologikoen laguntzaz bere datu fisiologikoak lor ditzake (adibidez: elektrokardiografo baten bidez bihotzaren jarduera elektrikoa). Hala ere, sentsoreen teknologiaren murriztapenak direla eta, ezinezkoa suertatzen da zuzenean zerbitzarira datuak transmititzea. Honetaz ere *Android* aplikazioa arduratuko da.

Mugikorrak sistemari funtzionalitate berriak gehitzen dizkio. Batez ere, erabiltzailearekiko elkarrekintza (eskuratutako datuen bistaraketa, atazen esleipena telematikoki jasotzeko aukera, etab.). Era orokor batean, esan daiteke mugikorrak hiru helburu nagusi izango dituela: sentsoreetatik datuak jaso, erabiltzaileei datuak ikuskatzeko aukera eman, eta azkenik, ikertzaileekiko komunikazioa, bai datuak bidali, bai atazak jaso.

Azkenik, zerbitzariaren atala hiru modulu nagusiez osatuta egongo da: esperimentuen diseinua, esperimentuen emaitzen kudeaketa eta lortutako datuen biltegiraketa. Lehendabiziko modulua web aplikazio baten garapenean datza eta ikertzaileekin harremanetan egongo da era zuzenean. Ikertzaileak web aplikazioaren bidez nahi duen esperimentua diseinatuko du eta partehartzaileak burutu arte itxaron beharko du. Bigarren modulua web zerbitzu baten garapenean datza eta mugikorrarekin konexioak egiteko interfazea izango da. Honen bidez, esperimentuen emaitzak zerbitzarira bidaliko dira. Azkenik, hirugarren moduluari dagokionez, laino sistema batean datu fisiologiko kantitate handiak gordetzeko egokia den formatu batean (*HDF5*) biltegiratuko dira.

Datuen iraunkortasuna eta segurtasuna kudeatzeko web zerbitzari baten inplementazioa egingo da proiektuan. Sistemaren barnean egindako datuen transmisio guztiak seguruak izango dira. Horretarako, konexio guztiak zifratu egingo dira bidalitako datuen konfidentzialtasuna bermatu ahal izateko. Pertsonen datu fisiologikoekin lan egingo denez, pertsonek dauzkaten eskubideak [3] [4] bermatuko dira.

2.2 Lanaren deskonposaketa egitura

2.1 irudiko diagraman proiektuan zehar landuko diren atalak ikus daitezke lan-paketetan antolatuta.

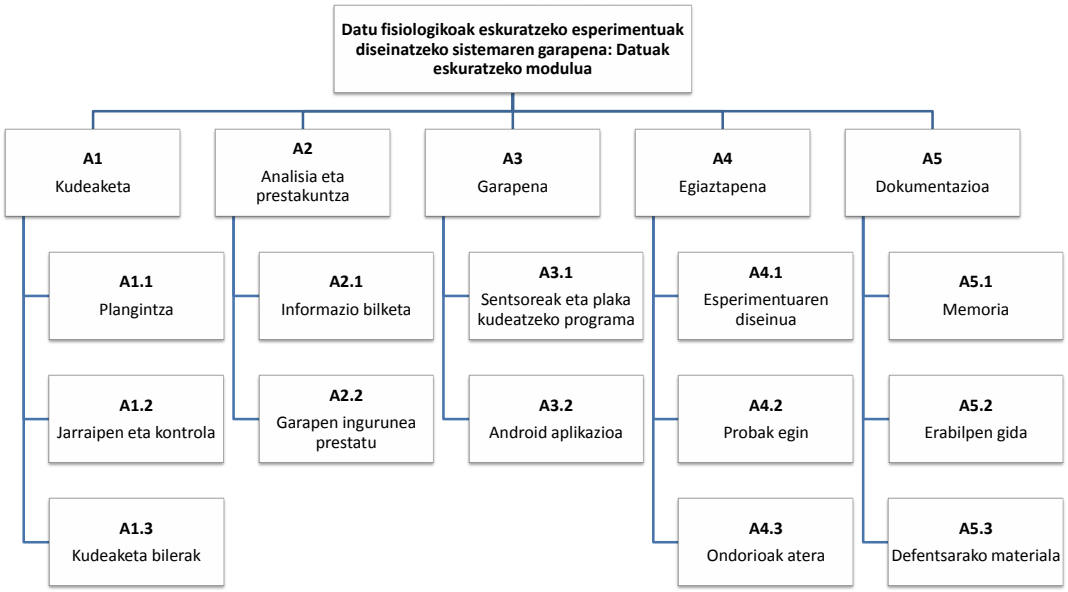
2.3 Atazak

Atal honetan proiektua aurrera eramateko beharrezkoak izango diren atazak zerrendatu dira, lanaren deskonposaketa egitura kontuan hartuz.

A1 Kudeaketa

A1.1 Plangintza

A1.1.1 Irismena zehaztu



2.1 Irudia: Lanaren deskonposaketa egituraren diagrama

A1.1.2 LDE egin

A1.1.3 Atazak definitu

A1.1.4 Kronograma egin

A1.1.5 Dedikazio aurreikuspena kalkulatu

A1.1.6 Komunikazio plana burutu

A1.1.7 Kalitate plana burutu

A1.1.8 Arriskuen plana burutu

A1.1.9 Eskuraketen kudeaketa egin

A1.1.10 Lan metodologia definitu

A1.2 Jarraipen eta kontrola

A1.2.1 Desbiderapenak kalkulatu

A1.2.2 Desbiderapenen arrazoiak bilatu

A1.3 Kudeaketa bilerak prestatu eta gauzatu

A2 Analisia eta prestakuntza

A2.1 Informazio bilketa

A2.1.1 Sentsoreak eta plakak

A2.1.2 Komunikazio teknologiak

A2.1.3 Android

A2.2 Garapen ingurunea prestatu

A3 Garapena

A3.1 Sentsoreak eta plakak kudeatzeko programa

A3.2 Android aplikazioa

A3.2.1 Txartelarekin komunikatzeko modulua

A3.2.2 Zerbitzariarekin komunikatzeko modulua

A3.2.3 Erabiltzailearekiko interfazea

A4 Egiaztapena

A4.1 Esperimentuaren diseinua

A4.2 Probak egin

A4.3 Ondorioak atera

A5 Dokumentazioa

A5.1 Memoria idatzi

A5.2 Erabilpen gida prestatu

A5.3 Defentsarako materiala prestatu

2.4 Kronograma

Hurrengo kronograman planifikatutako lana islatzen da. Kronograma asteka zatituta dago, astero burutako atazak agertzen direlarik. Aipatu beharra dago proiektua kurtso osorako planifikatu dela, hala ere, hasierako eta amaierako hilabetei ez zaizkie lan karga kantitate bera esleitu.

Lehenengo hilabetetan proiektuaren nondik norakoak finkatuko dira, informazio bilketa eta garapen inguruneak prestatzen diren bitartean.

Urritik otsailera erabiliko diren teknologien inguruan informazio bilketa eta hainbat proba egingo dira. Lortutako informazioaren helburua sentsoreak, plaka, eta Androiden nondik norakoen jakinpean egotea da. Baita ere, probak eginez, garapenerako beharrezkoa den trebetasuna lortzea.

Otsailean, prestakuntza atazak bukatuta, garapena eta dokumentazioarekin hasiko da. Apirila eta maiatzean banakako garapenarekin bukatu ostean, proiektukidearekin amankomunak diren atazak burutuko dira, sistemaren moduluen arteko komunikazio protokoloa definizioa hain zuzen ere. Proiektuaren garapenarekin batera produktuaren probak hasiko dira.

Azkenik, ekainean probak egitearekin eta memoriarekin bukatuko da, baita ere, erabilpen gida eta defentsarako erabiliko den materiala prestatzeari ekingo zaio.

Uztaila, arriskuen atalean azaltzen den moduan, gerta litezkeen ustekabei aurre egiteko utziko da.

Esan beharra dago proiektuaren garapenaren zehar hainbat kudeaketa bilera egingo direla, bai zalantzak argitzeko nola zuzendariak eta zuzendarikideak proiektuaren egoeraren berri jakin dezaten.

Zehaztasun gehiagorako ikusi 2.2 irudiko kronograma.

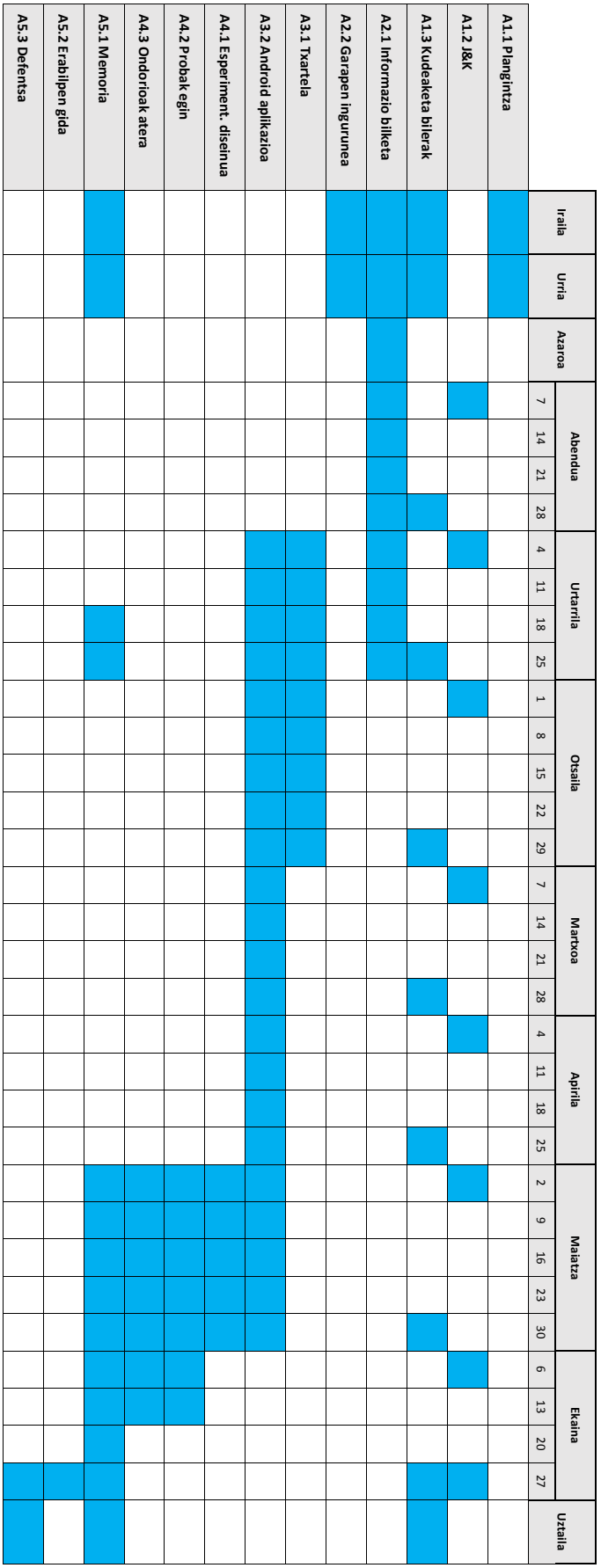
2.5 Dedikazio aurreikuspena

2.1 taulan proiektua garatzeko dedikazio aurreikuspena ikus daiteke, definitutako lan-paketeen arabera.

A3 lan-paketea, hau da, garapenari dagokiona, izango da dedikazio gehien behar duena. Hurrengoak, A5 (dokumentazioa) eta A2 (analisia eta prestakuntza) lan-paketeak izango dira, A4 (egiaztapena) lan-paketeaz jarraituta. Azkenik, A1 (kudeaketa) lan-paketea denbora gutxien behar izango duela espero da.

2.6 Komunikazio plana

Zuzendari eta zuzendariordearekin bi modutara egingo da komunikazioa. Alde batetik, zalantza edo arazoren bat izanez gero, EHU-ko posta elektronikoa erabiliko da hauek argitzeko. Bestetik, proiektuaren garapen-egoeraren arabera hainbat bilera egingo dira Informatika Fakultatean. Proiektuarekin hasi aurretik lehenengo bilera bat proiektuaren atal garrantzitsuenak definitzeko eta finkatzeko eta hasierako zalantzak argitzeko. Hurrengo bilerak behar direnean egingo dira, proiektuan aurreratu ahala. Azkenik, itxiera bilera bat egingo da proiektua bukatutzat eman ahal izateko.



2.2 Irudia: Kronograma

Lan-paketea	Dedikazio aurreikuspena (ordutan)
A1 Kudeaketa	18
A1.1 Plangintza	8
A1.2 Kudeaketa bilerak	5
A1.3 Jarraipena eta kontrola	5
A2 Analisia eta prestakuntza	60
A2.1 Informazio bilketa	55
A2.2 Garapen ingurunea prestatu	5
A3 Garapena	120
A3.1 Sentsoreak eta plakak kudeatzeko programa	20
A3.2.1 Txartelarekin komunikatzeko modulua	20
A3.2.2 Zerbitzariarekin komunikatzeko modulua	20
A3.2.3 Erabiltzailearekiko interfazea	60
A4 Egiaztapena	50
A4.1 Esperimentuaren diseinua	10
A4.2 Probak egin	20
A4.3 Zuzenketak	20
A5. Dokumentazioa	100
A5.1 Memoria	90
A5.2 Erabilpen gida	4
A5.3 Defentsa prestatu	6
GUZTIRA	**348**

2.1 Taula: Dedikazio aurreikuspena.

Bestalde, gure artean hainbat modutara komunikatuko gara. Bereziki, *WhatsApp* bidez hitz egingo dugu zalantzak azkar argitu eta bakoitzak egiten ari den lanaren berri izateko. EHU-ko posta elektronikoa ere erabiliko dugu kasu batzuetan.

Esan beharra dago proiektuaren inguruan sortzen zaizkigun zalantzak argitzeko Borja Gamecho doktorearen [5] laguntza ere izango dela.

2.7 Kalitate plana

Atal honetan proiektuaren kalitate plana azalduko da, proiektuaren kalitate betekizun eta dimentsioak definituz. Honakoak dira proiektuaren oinarrizko betekizunak:

- Sentsoreetatik eskuratutako datuak ahalik eta zehatzenak izatea (elektromiograma, elektrokardiograma, azalaren jarduera elektrikoa, argitasuna, azelerometroa, ...).

- Erabiltzaileak sentsoreen egokitasuna egiaztatzeko aukera.
- Ikertzaileak jasotako datuak sentsoreetatik lortutakoekin bat etortzea (datuen integritatea).
- Erabiltzaile eta ikertzaileen kontuen kudeaketa zuzena.
- Ikertzaileari datuak era ordenatu eta ulergarri batean helaraztea.
- Ikertzaileek beharrezkoak dituzten datuak zeintzuk diren erabiltzaileei adieraztea eta ebaluatzeko mekanismo bat izatea (atazen esleipena).
- Lortutako datu fisiologikoen iraunkortasuna datu-base batean.
- Erabiltzaile kontuak eta datu fisiologikoak biltegiratzerakoan Datuen Babeserako Lege Organikoa eta Etika Batzordearen eskakizunak bete.

Ondorengoak dira proiektuaren kalitate dimentsioak:

- Sistema, datu fisiologikoak eskuratzen dituen txartel zehatz batera mugatuta ez egotea.
- Ikerlariak, erabiltzaileak sentsoreetatik jasotako datu fisiologikoak denbora errealean ikusteko aukera.
- Erabiltzaile eta ikerlariaren interfazeak erabilgarriak eta irisgarriak izatea.
- Erabiltzaileek egindako atazak eta ebaluazioen atazak berrikusteko aukerak.

2.8 Arriskuen plana

Jarraian proiektuaren garapenean zehar gerta litezkeen arazoak aurreikusi eta hauek nola konpondu planifikatu da. Aipatu beharra dago arrisku guztiek ez dutela modu berdinean eragina izango proiektuarengan.

Ustekabe larriak ekiditeko proiektua entregatze-epea baino 15-30 egun lehenago bukatuko da, hala ere, ustekaberen bat izanez gero, galdutako orduak hurrengo asteetan berreskuratuko dira era uniforme batean. Proiektua bi pertsonen artean banatuta dagoenez muturreko ustekabe batean beste pertsona egin daiteke kargu. Arazo epe luzekoa izanez gero planifikazioa berregingo da, atazen lehentasun eta denborak birkalkulatuz.

Horren ildotik, denbora falta kudeatzeko asmoz proiektuaren amaiera benetako epea baino 15-30 egun lehenago planifikatuko da. Horrela, egun batzuk koltxoi gisa utziko dira. Arazoa gertatu eta konpondu ez gero, kalitate dimentsioak albo batera utziko dira oinarrizko betekizunetan zentratzeko.

Analisia eta prestakuntza atalean denbora nahikoa emango da garapenean zehar gerta daitezkeen arriskuak ekiditeko. Hala ere, arazoren bat izanez gero zuzendari, irakasle edota ikertzaileengana joko da.

Datu galerak ekiditeko egunero babes kopiak egingo dira bai lan egiteko erabiltzen den makinan bertan nola kanpoko euskarri batean. Hardwarean matxuraren bat izanez gero, proiektua unibertsitatean dauden baliabideekin moldatuko da.

2.9 Eskuraketen kudeaketa

Proiektu honen garapena hasi aurretik eskuraketa bat egin beharko da, bereziki, hau da, sentsoreak eta hauek konektatzeko plaka lortzea. Eskuraketa hau Borja Gamechoren bidez egingo da.

Gainerako beharrezko eskuraketak Internet bidez egingo dira, hala nola erabiliko den software-a eta erreminta desberdinak eta hauen dokumentazio orriak.

2.10 Lan metodologia

2.1 atalan azaldu den moduan, proiektu hau hiru ataletan banatu da, bi banakako atal eta atal amankomun bat. Laburbilduz:

1. Ezekiel Sarasua ikasleak: txartela, sentsoreak eta *Android* aplikazioaren garapena.
2. Maider Simón ikasleak: zerbitzariaren konfigurazioa, web zerbitzua eta web aplikazioaren garapena eta datu fisiologikoen biltegiraketa.
3. Atal amankomuna: sistemaren arkitekturaren eta moduluen arteko komunikazio protokoloen diseinua.

Ezekielek partehartzaileen atalan egingo du lan, hau da, erabiliko den plakaren funtzionamendua eta sentsoreak aztertu eta *Android* aplikazioa garatuko du. Maider, berriz, zerbitzari atalan zentratuko da, hau da, web zerbitzariaren instalazioan, konfigurazioan eta web

aplikazioaren inplementazioan egingo du lan. Bi atal hauen arteko elkarrekintza bien artean adostuko da, hau da, sistemaren egitura nagusiaren diseinua, arkitektura eta bi atalen komunikazio protokoloen diseinua.

Nahiz eta bakoitza bere zatian zentratu, biek izango dute bestearen lanaren berri momentu guztian. Horretarako, Informatika Fakultateko 302. laborategian egingo dira bilerak, egindako lana batak besteari erakutsi eta azaltzeko. Bestalde, bakoitzak bere kasa egingo du lan eta proiektuan aurrera egiteko arazoak edo zalantzak sortuko balira, proiektuko zuzendari eta zuzendarikideari galdetuko liekete.

Proiektuaren garapena hainbat fasetan banatu da. Lehenengoan, Proiektuaren Helburuen Dokumentua sortuko da plangintza moduan, hau da, zer egin behar den eta nola dokumentatu. Bigarren fasean, lana aurrera eramateko behar den informazioa bilatu, ikasi eta dokumentatuko da. Hirugarren fasean, ikasitakoa aplikatuko da sistemaren inplementazioa egiteko. Azkenik, garatutako sistema probatuko du hainbat proba kasu aztertzen.

Esan beharra dago fase bakoitzean egindako guztia dokumentatuko dela, gero GAParen memorian txertatu eta moldatzeko.

2.11 Jarraipen eta kontrola

Atal honetan, proiektuan zehar egindako atazen benetako dedikazioa aurki daiteke. Baita ere, aurreikusitako eta benetako dedikazioaren arteko desbideraketen arrazoi nagusiak azalduko dira.

2.11.1 Benetako dedikazioa

Borja Gamecho-k eskainitako laguntza dela eta, lan-pakete bakoitzari dedikatu zaion denbora ez da asko aldatu aurreikuspenetik benetako dedikaziora. Hala ere, kudeaketa eta egiaztapen ataletan desbiderapen esanguratsuak jaso dira.

Ordu kopuru totala kontutan hartuz, 43 ordu gehiago behar izan direla ikusi ahal da. Desbideraketa honen eragile nagusiak A1.2, A4.2 eta A4.3 atazak izan dira. 2.2 taulan proiektuaren benetako dedikazioa aurkitu daiteke.

Lan-paketea	Dedikazio aurreikuspena (ordutan)	Benetako dedikazioa (ordutan)
A.1. Kudeaketa	18	26
A1.1. Plangintza	8	8
A1.2. Kudeaketa bilerak	5	15
A1.3. Jarraipena eta kontrola	5	3
A2. Analisia eta prestakuntza	60	60
A2.1. Informazio bilketa	55	55
A2.2. Garapen ingurunea prestatu	5	5
A3. Garapena	120	130
A3.1 Sentsoreak eta plakak kudeatzeko programa	20	10
A3.2.1 Txartelarekin komunikatzeko modulua	20	20
A3.2.2 Zerbitzariarekin komunikatzeko modulua	20	20
A3.2.3 Erabiltzailearekiko interfazea	60	80
A4. Egiaztapena	50	85
A4.1. Esperimentuaren diseinua	10	15
A4.2 Probak egin	20	30
A4.3 Zuzenketak	20	40
A5. Dokumentazioa	100	100
A5.1. Memoria	90	90
A5.2 Erabilpen gida	4	4
A5.3. Defentsa prestatu	6	6
GUZTIRA	**358**	**401**

2.2 Taula: Benetako dedikazioa.

2.11.2 Desbideraketen arrazoiak

Aurreko azpi-atalan aipatutako A1.2, A4.2 eta A4.3 atazen desbideraketen arrazoi nagusia planifikatuta ez zegoen UCAmI-2016 kongresurako prestatu beharreko entregekin zerikusia daukate. Kongresuak eskatzen zituen sistemaren balidazioa zela eta, zegokion ataza (A4.2) era esanguratsu batean luzatu zen.

Hain balidazio sakona eginez gero, sisteman aurkitutako akatsak eta erabilgarritasun ezak konpontzen aurreikusitako baino denbora gehiago eman behar izan da, A4.3 ataza alegia.

Azkenik, aipatu beharra dago, istripu batean orkatila apurtu nuela eta hilabete batez igeltsuarekin egon behar izan nintzela. Makuluekin lantokira hurbiltzeko zailtasunak zirela eta, etxetik lan egingo nuela adostu genuen. Hala ere, proiektuaren lan erritmoa moteldu zen.

Aipatu berri diren bi arrazoiengatik ezinezkoa izan zen aurreikusita zegoen epean proiek-

tua bukatuta izatea. Hala ere, planifikazioan utzitako hilabete bateko koltxoiari esker, posible izan da proiektua azken entrega eperako prest izatea.

3. KAPITULUA

Proiektuaren prestakuntza

Kapitulu honetan, sistemaren datu eskuratze modulua garatzeko egin behar izan den prestakuntza dokumentatu da. Lau izan dira prestakuntza alor nagusiak; txartelen ezaugarrien azterketa eta haien funtzionamenduaren ikasketa, txarteletatik mugikorrera datuak transmititzeko aukeren azterketa, mugikorretik web zerbitzarira datuak bidaltzeko teknologien ikasketa, eta, erabiltzaileekin elkar-eragingo duen Android sistemaren azterketa.

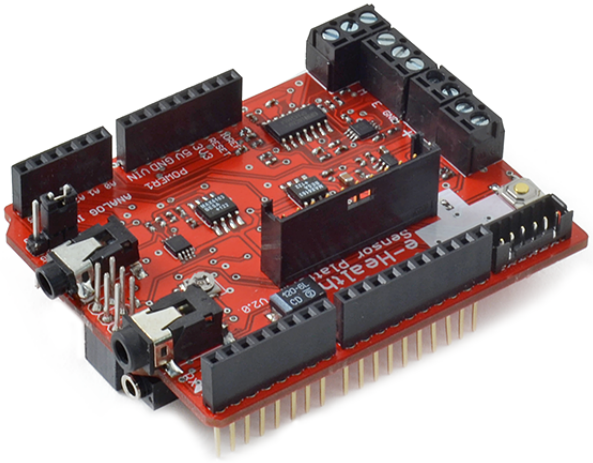
3.1 Txartelak eta sentsore fisiologikoak

Atal honek *e-Health* eta *BITalino* txartelen nondik norakoak azaltzea du helburu. Lehenik eta behin, *e-Health* plaka eta onartzen dituen sentsoreen deskribapen laburra aurkituko da. Ondoren, *BITalino*-ak eskaintzen dituen aukerak aztertuko dira.

3.1.1 *e-Health*

e-Health [6], *Arduino* edo *Raspberry Pi*-ri txertatzen zaion txartel bat da. Elkarketa honen emaitza datu fisiologikoen eskuraketa ahalbidetzen duen modulu bat da. Modulu honen helburua aplikazio mediko nola biometrikoak garatzeko euskarri bat eskaintzea da. *e-Health* plaka ondoko sentsoreez osatuta dago:

- Pultsioximetroa: Arterietako odolaren oxigeno saturazioa neurtzen du, argi trans-



3.1 Irudia: *e-Health* txartela

kutaneoaren xurgatzea arterietako odolaren oxigeno saturazioarekin erlazionatuz. Arnas jardueraren ondorioak eta pultsu kapilarra detektatzen ditu.

- Arnas fluxua neurtzeko tresna: Arnasaldi bakoitzeko eskuratzen den aire kantitate neurtzen duen tresna.

- Termometroa: Gorputzen hozberoa neurtzeko erabiltzen den tresna. Oro har, kristalezko hodi itxi bat izaten da, *Celsius* edo *Fahrenheit* graduz markatua, eta barruan isurkari bat daramana, alkohola edo merkurioa. Likidoak gora eta behera egiten du tenperaturaren arabera, beroarekin zabaldu eta hotzarekin uzkurtzen delako.

- Elektrokardiografoa: Bihotzaren jarduera elektrikoa erregistratzeko erabiltzen den tresna; ez-ohiko kinadak atzemateko erabiltzen da. Bihotzeko zenbait asalduren diagnostikoa egiten laguntzen du elektrokardiografoak.

- Glukometroa: Odolean aurkitzen den glukosa-kontzentrazioa neurtzeko erabiltzen den tresna.

- Azalaren jarduera elektrikoa neurtzeko tresna: Batez ere, esku eta oin-zoletan aurkitzen diren izerdi guruin ekrinoen jarduera neurtzen dituen tresna da. Guruin hauen berezitasun nagusia estimulazio psikikoei erantzuten dutela da, adibidez, karga handiko estimulu emozionalei; urduritasuna, poza, beldurra, etab.

- Azelerometroa: Gailuaren orientazioa eta ondorioz erabiltzen ari den pertsonaren 'orientazioa', adibidez; zutik, etzanda, etab. adieraz dezakeen tresna da

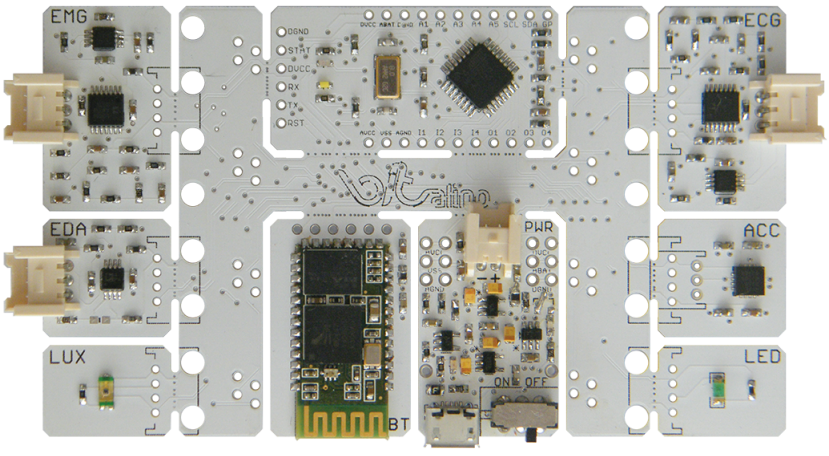
- Elektromiografoa: Uzkurdura muskularraren aktibitate elektrikoa neurtzeko tresna.

Sentsoreetatik lortutako informazioa beste gailuetara garraia daiteke *Arduino* edo *Raspberry Pi*-rekin bateragarriak diren hainbat komunikazio teknologien bidez, besteak beste: *Bluetooth*, *3G* eta *Wi-Fi*. Komunikazio horien helburua denbora errealeko edota lainoan darabilten aplikazioen garapena ahalbidetzea da.

Proiektu honetan *e-Health* txartelarekin batera *Arduino* plaka bat erabili da, hala ere, *Raspberry Pi* bat aukeratuz gero funtzionamendua berdintsua litzateke.

3.1.2 *BITalino*

e-Health plaka ez bezala, *BITalino* datu fisiologikoak eskuratu ahal dituen gailu independentea da [7]. Hiru *BITalino* mota ezberdin aurki daitezke: *Board Kit*, *Freestyle Kit* eta *Plugged Kit*. Proiektu honetan Informatika Fakultatean aurkitzen den *Board Kit*-a erabili da. Hona hemen txartelaren espezifikazioak eta onartutako sentsoreak:



3.2 Irudia: *BITalino* txartela

- Laginketa maiztasuna: 1, 10, 100 edo 1000Hz-ra konfiguragarria
- Portu analogikoak: 10 biteko 4 sarrera eta 6 biteko 2 sarrera
- Portu digitalak: bit bateko 4 sarrera eta bit bateko 4 irteera
- Datu konekzioak: Bluetooth v2.0 (>10m eremua)

- Tamaina: 100x60mm
- Pisua: 30g
- Sentsoreak:
  - Elektromiografoa
  - Azalaren jarduera elektrikoa neurtzeko tresna
  - Elektrokardiografoa
  - Azelerometroa
  - Argitasun maila neurtzeko tresna

BITalino-n eskuratutako datu fisiologikoak beste gailura transmititzeko dagoen aukera bakarra, *Bluetooth* egokitzailea erabiltzea da.

3.2 Datuen eskuraketa

Proiektu honen helburua ikertzaileei urruneko erabiltzaileen datu fisiologikoak eskuratzea ahalbidetzen duen sistema bat garatzea da. Atal honetan hasierako datu bidalketa jorratuko da, erabiltzailearen mugikorrean egin beharreko datu eskuraketa hain zuzen ere.

e-Health plakaren deskribapenean aipatu den moduan, nahiz eta *Arduino*-a interneteko zerbitzari batekin zuzenean komunikatu ahal den, sistema honetan *Android* mugikor bat erabiliko da bitartekari gisa. Android mugikorraren erabilpena erabiltzaileei orientatuta egongo da, erabiltzaileak ikertzaileengandik jasotako atazak (esperimentuak) era erraz batean ikuskatzeko aukera izateko. Honek, *Android* mugikorra eta txartelen arteko komunikazioa nahitaezkoa egiten du.

3.2.1 *e-Health*

Aipatu berri den datu eskuraketa ondoko bi entitatez osatuta egongo da; *Android* mugikorra eta *Arduino* & *e-Health* modulua. Modulu gehigarriak izanez gero, *Arduino*-ak hainbat komunikazio aukera onar ditzake. Aldiz, gaur egungo mugikor gehienek defektuz bi komunikazio aukera nagusi dakartzate; *Wi-Fi* eta *Bluetooth* teknologiak. Datu transmisio honetarako *Bluetooth*-a erabiliko da.

BITalino plaka ez bezala *Arduino & e-Health* modulua ezin da *Android* aplikazioaren bidez kontrolatu. Bi entitateen arteko komunikazioa gauzatzeko bi programa beharrezkoak dira, bata *Arduino*-rentzat eta bestea *Android*-entzat:

Arduino & e-Health funtzionamenduaren nondik norakoak hobeto ulertzeko, prestakuntza atal honetan Borja Gamecho-k egindako *Arduino* programa bat abiatu da, hona hemen atal garrantzitsuenak:

- Lehenengo pausua, maiztasun abiadura zehaztuz, mugikorrarekin serie konexio bat abiaratzea litzateke. Ondoren, datuak eskuratuko dituzten sentsoreak gaitu daiteke.

```
Serial.begin(115200);
eHealth.initPulsioximeter();
```

- Programa begizta nagusi batean sartuko da, non, serie konexiotik datozten eragiketa kodeen arabera, *Arduino*-a kontrolatuko da. Adibidez, serie konexiotik zero zenbakia etorriz gero datu eskuraketa gelditukо da, aldiz, bateko bat etortzen bada datu eskuraketa hasiko da.

  Arduinoak *C* sintaxia erabiltzen du programazio lengoaia gisa, beraz, aipaturiko aurreko kudeaketa nahiko korapilotsua egiten da: Serie konexioaren kudeatzea, jasotzen diren karaktere kateen prozesaketa, etab.

- Sentsoreetatik eskuratutako datuak hurrengo komandoarekin irakurri daiteke:

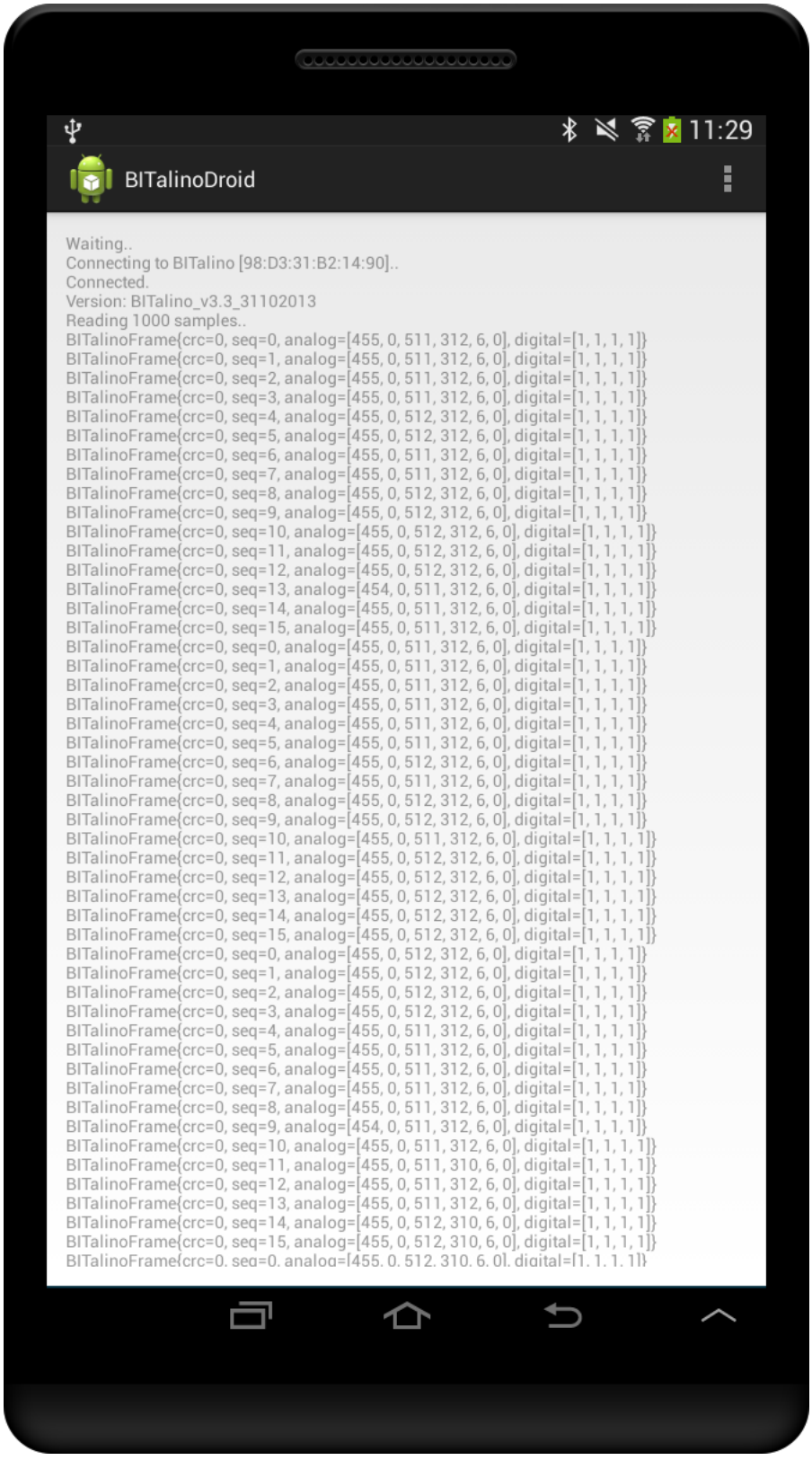
```
eHealth.readPulsioximeter();
```

- Azkenik, esan beharra dago datuen laginketa hainbat maiztasun abiaduratan egin daitekela. Hala ere, eginkizun horretarako beharrezkoa da *Arduino*-k eskura uzten dituen barne erlojuen kudeaketa egitea sistemaren etenak erabiliz.

3.2.2 *BITalino*

BITalino-ak kanpoaldearekin komunikatzeko duen aukera bakarra integratuta duen *Bluetooth* egokitzailea erabiltzea da. *BITalino*-ren funtzionamendua hobeto ulertzeko *GitHub* web orrian aurkitu daitekeen *Android*-erako adibidea [8] abiatu da.

3.3 irudian ikusi ahal den moduan, adibidea *Android* pantaila bateko *Java* programa sinple bat da, non, *BITalino*-ren sentsore guztien uneko balioak ikusten diren. Hona hemen garatu den sistemarako erabilgarriak izan diren atal garrantzitsuenak.



3.3 Irudia: *BITalino* adibidea *Android*-en

Java programa guztietan gertatzen den antzera, lehendabiziko pausua aldagaiak deklaratzea izango da. Ondorengoak dira erabiliko diren aldagai esanguratsuenak:

- *BITalino* txartela kudeatzeaz arduratu den `BITTalinoDevice` motako aldagaia.
- `BluetoothDevice` klasea sortzen duen objektua, programan, *BITalino*-ren *Bluetooth* egokitzailea adierazteaz arduratuko da.
- `BluetoothSocket` motako aldagai bat erabiliko da aurreko egokitzailearekin komunikazioa gauzatzeko erabiliko den *socket*-a kudeatzeko.

BITalino-rekin *Bluetooth* konexioa gauzatzeko ezinbestekoa izango da programan *BITalino*-ren *Bluetooth* egokitzailearen *MAC* helbidea zehaztea.

Behin bi gailuen arteko konexioa eginda dagoela, gomendagarria da mugikorraren Bluetooth begiztatzea amaitzea energia aurrezteko.

Hurrengo komandoa aipatu berri den `BITTalinoDevice` motako objetua sortzeko erabiliko da. Komandoaren lehenengo parametroak*BITalino*-ak hartuko duen laginketa abiadura adierazten du. Aldiz, bigarren parametroak, gaituko diren kanalak adierazten du (kanal bakoitzak sentsore desberdin bati dagokio). Bigarren lerroan aurkitzen den komando exekutatuz gero datu eskuraketa hasiko da.

```
bitalino = new BITalinoDevice(1000, new int[]{0, 1, 2, 3, 4, 5});
bitalino.start();
```

```
BITalinoFrame[] frames = bitalino.read(numberOfSamplesToRead);
```

Aurreko komandoarekin *BITalino*-ak eskuratzen ari den laginak irakurri daitezke. Aipatu beharra dago aurreko agindua blokeakorra dela, hau da, programaren kontrola ez da itzuliko numberOfSamplesToRead aldagaian adierazten diren lagin kopurua irakurri arte. Kasu honetan, laginketa maiztasuna 1000Hz-koa eta numberOfSamplesToRead = 1000 izanik. Aginduak, 1000 lagin irakurri dituenean itzuliko du kontrola, abiadura 1000Hz-koa denez segundo bat behar izango du 1000 lagin irakurtzeko.

Geroago azalduko den moduan, informatikako arlo honetan itxarote denbora hori onartezina da. Beraz, beharrezkoa izango da beste exekuzio hari bat erabiltzea.

3.2.3 Ondorioak

Egindako proben arabera, sistema *BITalino* txartelarekin garatzen hastea hautatu da. Hala ere, proiektuaren helburua plaka bati lotuta dagoen sistema bat eratzea ez denez, *e-Health* plaka ez da guztiz baztertuko. Garapenenean aurrera joan ahala eta esperientzia gehiago izanda, *e-Health* plaka sisteman egokituko da.

BITalino, *e-Health* plaka baino ulergarriagoa, programatzeko errazagoa, etab. dela ohar daiteke. Fabrikatzaileak eskaintzen duen *Android* liburutegia nahiko sinplea da eta goi mailako programazioa ahalbidetzen du. *e-Health* moduluan ez bezala, bere funtzionamendurako plakan ez da programa gehigarririk instalatu behar. Baita ere, kasu honetan bateria integratuta daukanez gero, erosotasunaren aldetik ez dago konputagailuari konektatzeko beharrik.

Nahiz eta *e-Health* sistemak datu fisiologiko gehiago aztertzeko aukera eskaintzen duen (sentsore gehiago onartzen baitu), haren funtzionamendua korapilatsuagoa da, gainera, probak egiterakoan hainbat arazo jaso dira serie portuaren erabilpenarekin. Bai, *Bluetooth* (mugikorrarekin komunikatzeko) nola *USB* (*Arduino*-an programa kargatzeko), serie portua erabiliz transmititzen duten komunikazioak dira. *Arduino*-a serie portu bakarra duenez gero, ezinezkoa da bi komunikazio motak aldi berean erabiltzea. Bi aukerak behin eta berriz aldizkatuz gero *Arduino*-a era egoki batean funtzionatzeari uzten zion.

3.3 Datu Bidalketa

Aipatu den moduan, sistemaren helburu nagusia ikertzaileei datu fisiologikoak helaraztea da, hori dela eta, ezinbestekoa da erabiltzaileen *Android* mugikorrean eskuratu diren datuak bidaltzea. Hona hemen *HTTP* protokoloa erabiliz web zerbitzari batera fitxategiak igotzeko atal garrantzitsuenak:

Hasierako pausua *HTTP* konexioa prestatzea litzateke, horretarako, lehenik eta behin ezinbestekoa da zerbitzariaren *URL*-a zehaztea. Konexioa ireki eta gero konexioaren ezaugarriak adierazi behar dira; eskaera metodoa eta eduki mota hain zuzen ere. Eskaera metodoari dagokionez proiektu honetan *GET*, *POST* eta *PUT* metodoak erabiliko dira. Kasu partikular honetan, fitxategi bat igo nahi denez *POST* metodoa erabiliko da.

Azkenik, konexioaren eduki mota adierazi beharko da, adibidez: *image/png*, *text/html*, etab. Fitxategi bat bidaliko dela zehazten duen eduki mota *multipart/form-data* izango da.

```
URL url = new URL("http://bitalinogap.tk/api/v1/upload");
HttpURLConnection kon = (HttpURLConnection) url.openConnection();
kon.setRequestMethod("POST");
kon.setRequestProperty("Content-Type", "multipart/form-data" ...);
```

Behin konexioa eratuta dagoela, fitxategia transmititzeko nahikoa litzateke `OutputStream` batean haren edukia idaztea. Ondoren, fitxategia ondo igo dela ziurtatzeko, hurrengo komandoa erabili ahal da zerbitzariaren erantzuna zuzena dela egiaztatzeko.

```
if (kon.getResponseCode() == HttpURLConnection.HTTP_OK)
```

3.4 *Android*

Dokumentu hasieratik aipatu da *Android* mugikor baten beharra erabiltzailearekin elkareragiteko eta datu transmisioak gauzatzeko. *Android* mugikorra, garatuko den aplikazio baten bitartez kudeatuko da. Atal honen helburua aplikazioan erabiliko diren *Android*-en alderdi nagusienak azaltzea da.

3.4.1 Hariak

Aurreko bi ataletan aztertu diren datuak eskuratzeko eta transmititzeko funtzioak blokeakorrak dira, hau da, programa nagusia blokeatuko dute haien eginkizunak bukatu arte. Aplikazioa, erabiltzailearekin uneoro elkarrekintzarako prest egon behar denez, bi aukera aztertu dira suerta daiteken blokeo egoerari aurre egiteko; *Java*-ko `Thread` eta *Android* eskaintzen duen `AsyncTask` klaseak erabiltzea.

`Thread`, *Java*-n hariak sortzeko erabili ahal den klase bat da. Klase honek sortutako hariak hari nagusiarekin paraleloan exekutatzen dira, beraz, hari nagusiak bere eginkizunetan jarrai dezake, sortutako hariak kanpoko konexioez arduratzen direlarik. Hurrengo lerroetan ikusi ahal den moduan `Thread` klasearen erabilpena oso sinplea da eta ondorengo egitura dauka:

```
Thread thread = new Thread() {
        public void run() {
                // Exekutatu nahi diren komandoak
```

```
        }
}
thread.start();
```

Aldiz, `AsyncTask` *Android*-ek dakarren klase berezi bat da, `Thread` klasean azaldu den bezala, `AsyncTask`-ek hari nagusiarekin batera exekutatuko den hari berri bat sortuko du:

```
private class AT extends AsyncTask<DatuMota, DatuMota, DatuMota> {
        @Override
        protected void onPreExecute() {
                // Exekutatu nahi diren prestakuntza komandoak
        }
           @Override
        protected DatuMota doInBackground(DatuMota... parametroak) {
                // Exekutatu nahi diren komandoak
                // return emaitza
        }
           @Override
        protected void onPostExecute(DatuMota emaitza) {
                // Exekutatu nahi diren ondorio komandoak
        }
}
```

Aipatu beharra dago `AsyncTask` klaseak aukera gehiago eskaintzen dituen arren, prestakuntza fase honetan bakarrik garatutako sisteman erabiliko diren funtzioak azaldu dira.

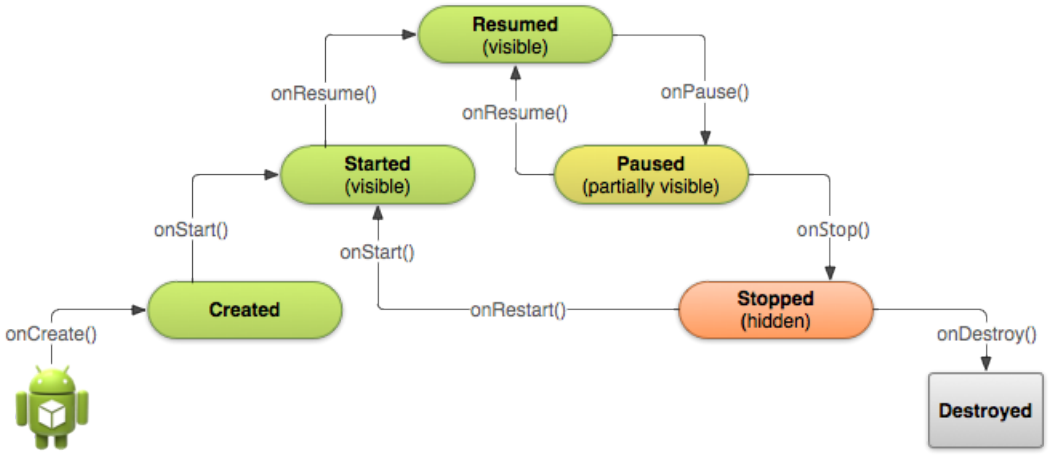
Ikus daitekeenez, `Thread` klasea `AsyncTask` baino askoz sinpleagoa dela, hala ere, hainbat probak egin ostean, erroreak jaso dira `Thread` klaseko hariak exekuzio garaian berrerabiltzean. Aldiz, `AsyncTask` klasea *Android*-en barneratuagoa dagoenez (*Android*-ek bere erabilpena gomendatzen du) kudeaketa *Android*-en esku geratzen da. Beraz, nahiz eta egitura konplexuagoa izan, sistema honetan harien kudeaketarako `AsyncTask` klasea erabiltzea erabaki da.

3.4.2 Klaseen arteko komunikazioa

Android aplikazioan ematen den komunikazio guztia ez da beti beste gailuekin gauzatzen, hain zuzen ere, aplikazio mailan ematen den klaseen arteko komunikazioa maizago izaten

da. Komunikatuko diren entitateen ezaugarrien arabera bi komunikazio aukera aztertu dira.

Androideko pantailek (*activity*-ek), 3.4 irudian aurkezten den bizitza zikloa jarraitzen dute, *activity* bakoitza *Java* klase batekin erlazionatuta dagoelarik.



3.4 Irudia: *Android activity*-en bizitza zikloa [1]

Activity-a *foreground* moduan dagoenean, hau da, pantaila ikusgarri, dagokion Java klasea exekuzioan egongo da. Aldiz, *activity*-a *background* modura igarotzen denean, hau da, pantaila ez ikusgarri, *Java* klasearen exekuzioa gelditu ko da. Beraz, nola gauzatuko da beharrezkoa den klaseen arteko komunikazioa aldi berean pantaila duen klase bakarra exekuzioan egon badaiteke?

Aukera bat uneoro exekuzioan dagoen klase global bat bitartekari gisa erabiltzea litzateke. Klase globala aplikazioa hasten denetik amaitu arte exekuzioan izango denez, ez da *activity* batekin erlazionatuta egongo, hau da, *Android* pantaila gabeko klasea izango da. *Android*-ek, halako klase globalak baimentzen eta automatikoki kudeatzen ditu. Horretarako, nahikoa da klasearen deklarazioa ondoko eran egitea.

```
public class Globala extends Application {
```

Aurreko atalan azaldutako hariak direla eta, gerta daiteke aita klase bat (pantailaduna) sortu berri duen klasearekin (ez-pantailaduna) komunikatu nahi dela. Kasu honetan, pantaila duen *Java* klase bat eta pantaila ez duen beste klase bat komunikatzeko, egokiena mezu trukaketak erabiltzea litzateke. Horretarako *Java*-ko `Handler` klasea erabiliko da.

Mezua bidaltzen duen klaseak, mezua bidali nahi dion klasearen `Handler`-a (postelekuaren helbidea) eskuratu beharko du, eta ondoren, mezua bidali.

```
Handler maneiatzailearenIzena = Globala.getManeiatzailearenIzena();
maneiatzailearenIzena.sendMessage(mezua);
```

Mezua jasoko duen klaseak berriz, `Handler`-a sortuko du eta mezuaren zain geratuko da.

```
public final Handler handlerIzena = new Handler() {
        public void handleMessage(Message mezua) {
                Bundle data = mezua.getData();
                // Datuekin egin nahi denaren kodea
        }
}
```

Esan daiteke bi komunikazio aukera hauen azpian inkesta eta etenen arteko aukeraketa dagoela. Komunikazioa klase globalaren bidez egitea hautatzen bada, behin eta berriz begiratu beharko da aldagaiaren egoeraren aldaketa. Aldiz, mezu trukaketa erabiliz gero, mezua iristean etena sortuko da.

3.4.3 *Android* proiektu baten fitxategi egitura

Nahiz eta proiektua hasi aurretik ezaguna izan, komenigarria izan da Android proiektu baten fitxategi egitura birpasatzea. Android proiektua hainbat direktorio eta fitxategiz osatuta dago, hona hemen garrantzitsuenak:

- Manifestua: fitxategi honetan aplikazioaren ezaugarriak gordeko dira, adibidez, Internet eta *Bluetooh* baimenak, pantaila kopurua eta haien orientazioa, etab.
- *java* direktorioa: direktorio honetan *java* fitxategiak gordeko dira.
- *layout* direktorioa: proiektuan erabiliko diren pantaila fitxategi guztiak direktorio honetan aurkituko dira.
- *Strings* fitxategiak: hizkuntz berri bakoitzeko fitxategi bat gehituko da..

3.4.4 Ingurunearen prestakuntza

Proiektuan zehar erabilitako garapen ingurune nagusia *Android Studio* [9] izan da. Horrez gain, hainbat liburutegi osagarrien erabilpena beharrezkoa izan da.

Android Studio

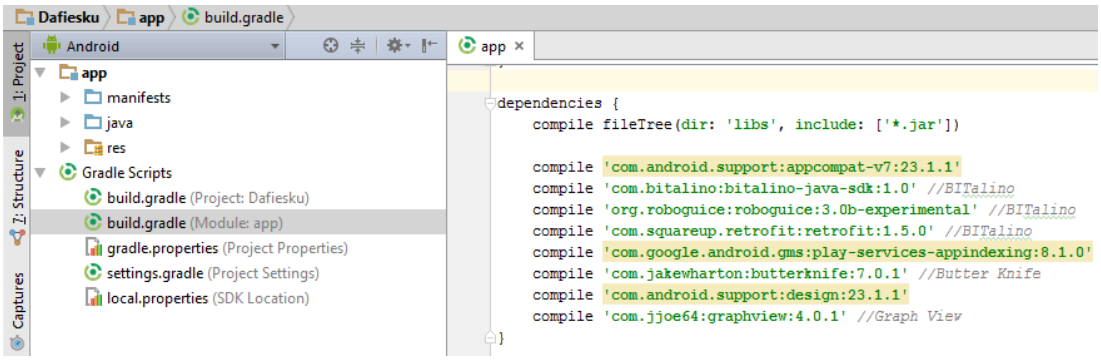
Android Studio Google-ek garatutako *Android IDE*-a da, 2014ko abenduan kaleratu zen *Eclipse* ordezkatuz. Proiektu honen garapenean 2.1.1 bertsioa erabili da.

Android Studio, *IDE* guztien antzera, *Android* programazioa errazagoa egiten duen interfaze grafiko bat baino ez da. *Android* aplikazio bat garatzeko beharrezkoak diren tresna guztiak *Android SDK*-n aurkitzen dira. Beraz, ezinbestekoa da tresna hauek *Android Studio*-n gehitzea.

Ohikoena bi osagaiak batera etortzea da, *Google*-ek, bai *IDE*-a nola *SDK*.

Aplikazioaren garapenean zehar hainbat kanpo liburutegiak erabili dira. Liburutegi gehienak hautazkoak dira; adibidez, grafikoak sortzeko (norberak gogoko duen grafiko liburutegia erabil dezake). Hala ere, oinarrizko liburutegi bat badago, *BITalino*-a kudeatzeko funtzioak eskaintzen dituzten liburutegiak hain zuzen ere.

Android proiektura BITalino liburutegiak gehitzeko, eta orokorrean liburutegi bat gehitzeko, *Gradle* fitxategiko (ikusi 3.5 irudia) *dependencies* atalean ondoko lerroak gehitu behar dira:



3.5 Irudia: *Gradle* fitxategia *Android* proiektu batean

```
compile com.bitalino:bitalino-java-sdk:1.0
```

```
compile org.roboguice:roboguice:3.0b-experimental
compile com.squareup.retrofit:retrofit:1.5.0
```

Android pantailetan aurkitzen diren osagai grafikoak kudeatzeko (adibidez: ea botoi bat sakatu den, zerrenda bat osatzeko, etab.) *Java* programetan estekatu behar dira. *Butter Knife* liburutegiak [10], estekatze hori egitea ahalbidetzen du kode ulergarriago bat erabiliz.

Hurrengo lehen lerro kodean, defektuzko erara *Android* botoi baten estekatzea ikus daiteke. Aldiz, bigarren lerroan, *Butter Knife* liburutegia erabilita egindako estekatzea ikusi ahal da.

- `Button botoia = (Button) findViewById(R.id.button1);`
- `@Bind(R.id.button1) Button botoia;`

Liburutegia *Android* proiektuan eskuragarri izateko *Gradle* fitxategian hurrengo lerroa gehitu behar da.

```
compile com.jakewharton:butterknife:7.0.1
```

Sentsoreetaik eskuratutako datuak erabiltzaileentzat ulergarriak izan dezaten grafiko batean sintetizatu eta pantailaratuko dira. Lan hori errazteko asmoz, *Graph View* [11] grafikoak irudikatzeko *Android* liburutegia erabiliko da.

Liburutegia *Android* proiektuan barneratzeko *Gradle* fitxategian ondoko lerroa gehitu behar da.

```
compile com.jjoe64:graphview:4.0.1
```

4. KAPITULUA

Proiektuaren garapena

Kapitulu honen helburua proiektuaren garapenaren nondik norakoak azaltzea da. Horretarako, bi atal nagusi desberdindu dira; sistemaren diseinua eta diseinuaren inplementazioa.

Bukatzeko, garatutako sistemaren balidazioa aurki daiteke.

4.1 Diseinua

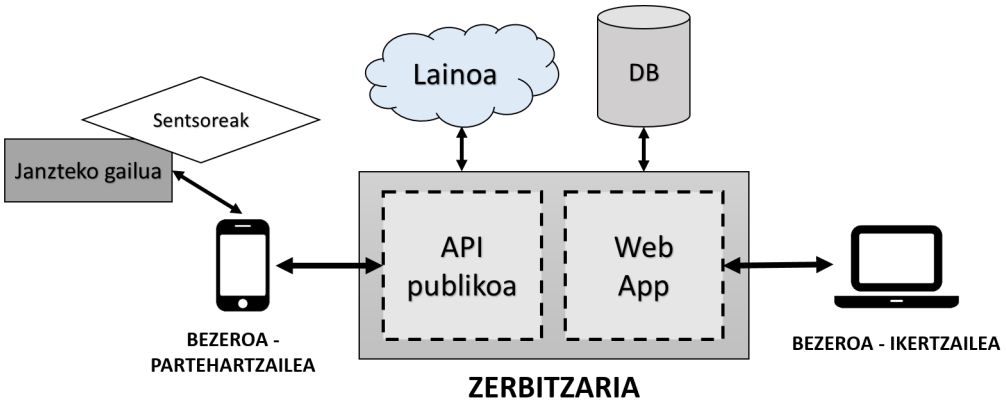
Proiektu honen irismena sistema osoa garatzea ez izan arren, sarrera gisa sistema osoaren diseinu orokorra azalduko da. Hurrengo bi azpi-ataletan, proiektu honetan landutako moduluaren diseinua eta moduluaren parte den *Android* aplikazioaren diseinua aurkitu ahal da.

4.1.1 Sistema osoaren diseinua

4.1 irudian, Maider Simon ikaslearekin batera diseinatutako sistemaren eskema orokorra ikusi ahal da:

Sistemak bezero/zerbitzari arkitektura jarraitzen du.

- Bezero aldea:
  - *Android* mugikorra (partehartzailea).



4.1 Irudia: Sistemaren arkitektura

    - BITalino txartela.
    - Sentsore fisiologikoak.
  - Konputagailua (ikertzailea).

- Zerbitzari aldea:
  - Web zerbitzaria.
    - API publikoa.
    - Web aplikazioa.
    - Datu-basea eta biltegiraketa.

Eginkizunen arabera, sistema bi modulu nagusitan banatu da: datuak jasotzeko modulua eta esperimentuen diseinuaren modulua.

Datuak jasotzeko modulua

Modulu honen helburuetako bat partehartzileari mugikortasuna eskaintzea da, *BITalino* txartela eta *Android* mugikor bat erabilita lor daitekeena, hain zuzen ere. Era honetan, partehartzaileak bere datu fisiologikoak eskura ditzake edozein ingurunean.

Lortutako mugikortasuna dela eta, partehartzaileak ez du ikertzailearengana jo beharko esperimenturen bat burutu ahal izateko. Hau da, bi entitateek ez dute ez leku ezta denbora berean konektatuta egon behar.

Izan ere, haririk gabeko konexioak erabili dira. *Android* mugikorra eta txartelaren arteko konexioa *Bluetooth* bidez egingo da. Bestalde, azpi-atal honetan geroago moduluen arteko konexioaren nondik norakoak azalduko dira zehaztasun gehiagorekin.

Ildo berean, eskuratutako datuak ingurune ez-kontrolatu batean lortzen direnez, ez dira laborategi batean eskuratutakoak bezain zehatzak izango, baina bai naturalagoak, ohiko bizimoduan murgilduta.

Azkenik, moduluak duen diseinu gardenari esker, partehartaileak ez du arlo teknikoez arduratu behar, adibidez: *BITalino* txartelaren *MAC* helbidea zehazteaz.

Esperimentuen diseinuaren modulua

Modulu honek hainbat eginkizun betetzen ditu. Zerbitzariaren kudeaketan zentratzen da batez ere, erabiltzaileentzat web aplikazioa eskainiz. Nahiz eta bereziki ikertzaileentzat diseinatuta dagoen, aplikazioaren bidez erabiltzaile motaren arabera ekintza desberdinak egin daitezke:

- Ikertzailea:
  - Sisteman izena ematea.
  - Partehartzaileen erregistroa.
  - Esperimentuen diseinua.
- Ikertzailea eta partehartzailea:
  - Esperimentuen egoera ikusi.
  - Kontuaren kudeaketa.

Erabiltzaile kontuak eta haien konfigurazioak gordetzeko datu-base sistema bat erabiliko da. Bertan ere sortutako esperimentu guztien ezaugarriak eta egoera gordeko dira.

Bestalde, web zerbitzuak (*API* publikoa) datu-basean biltegiratutako datuekin hainbat eragiketa egitea ahalbidetzen du. Hortaz, kanpoko gailuek (*Android* mugikorra), web zerbitzua eskaintzen dituen funtzioak erabilita datuen kudeaketa era garden batean egin dezakete. *Android* aplikazioak funtzio hauek sisteman sartzeko, esperimentuen kudeaketa eta datu fisiologikoen bidalketa egiteko erabiliko ditu.

Esperimentuan eskuratutako datu fisiologikoen kantitatea dela eta, egokia den *HDF5* formatuan biltegiratuko dira.

4.1 irudian ikus daitekeen moduan, proiektua nonahiko konputazioarekin erlazionatuta dagoenez, biltegiraketarako laino sistema bat erabiltzea adostu zen. Hala ere, denbora falta eta pribatutasun arazoak zirela eta, fitxategiak web zerbitzarian biltegiratzea erabaki da.

Moduluen arteko elkarrekintza

Sistemaren funtzionamendu egokirako ezinbestekoa da aipatu berri diren bi moduluen arteko komunikazioa gauzatzea. Horretarako, Maider Simon ikaslearekin batera bi moduluen arteko komunikazioa adostu da. Komunikazioa gauzatzeko *HTTP* protokoloa erabiliko da. Azaldutako bi moduluen arteko elkarrekintzak ondorengo prozedura jarraitzen du:

1. Ikertzailea sisteman sartu beharko da, web aplikazioaren bidez, bere kredentzialak erabiliz. Esperimentu bat sortuko du nahi duen partehartzaileari esleituz. Partehartzailea sisteman erregistratuta ez badago, ikertzaileak erregistratzeko aukera izango du.

2. Beste aldetik, partehartailea sisteman bere kontuarekin sartuko da *Android* aplikazioa erabiliz. Zerbitzariari *HTTP POST* eskaera egingo zaio, erabiltzaile izena eta pasahitza bidaliz. Kredentzialak zuzenak izanez gero, zerbitzariak partehartzailearen *API* gakoa eta *BITalino*-ren *MAC* helbidea itzuliko ditu. Datu transmisioak *JSON* formatuan gauzatuko dira.

3. Partehartzaileak atazak eskuratu ahal izateko *Android* aplikazioak *HTTP GET* eskaera egin behar izango du. Horretarako bigarren pausoan jasotako *API* gakoa erabiliko du identifikadore gisa. Atazak *JSON* formatuan jasoko dira.

4. Esperimenturen bat burutu eta gero, eskuratutako datuen *ZIP* fitxategia bidaliko da. Bidalketa *HTTP POST* eskaera baten bidez egingo da, eduki mota (*Content-Type*) *multipart/form-data* dela adieraziko da.

5. Behin *ZIP* fitxategia igo dela, partehartzaileak esperimentua bukatu duela adieraziko du. Esperimentuaren egoera *HTTP PUT* eskaera baten bidez eguneratuko da datu-basean.

6. Jaso berri den *ZIP* fitxategiaren edukiak *HDF5* formatua duen fitxategi batean bihurtuko dira, zerbitzarian biltegiratzen delarik.

7. Ikertzaileak web aplikazioaren bidez esperimentuen informazioa eta egoera edozein momentuan kontsulta dezake, *HDF5* fitxategia deskargatzeko aukera ere izanez.

4.1.2 Datuak jasotzeko moduluaren diseinua

Proiektu honetan datuak jasotzeko modulua landu da. Modulu honek bi gailu desberdinez osatuta dago. Batetik, partehartzailetik datu fisiologikoak eskuratuko dituen txartela, eta bestetik, esperimentuak kudeatu eta datuak zerbitzarira bidaliko dituen *Android* mugikorra.

BITalino txartela

Datu eskuraketa egiteko, *e-Health* txartelarekin ez bezala, *BITalino* txartelan ez da programarik exekutatu behar, txartela, *Android* aplikaziotik kudeatzen baita.

Prestakuntza atalean aipatu den moduan *BITalino* txartela lau laginketa maiztasun desberdin eskaintzen ditu. Sistemaren hasierako bertsio honetan eskuraketa guztietarako laginketa maiztasun berdina erabiliko da, 100Hz. Dena den, aipatu beharra dago kudeaketa hau oso eraginkorra ez dela zeren eta hainbat sentsore laginketa maiztasun txikiago batekin funtziona dezakete.

Baita ere, inplementazioa errazteko asmoz, edozein datu eskuraketan, nahiz eta beharrezkoak, ez izan kanal guztiak irekiko dira. Hau da, eskuraketa, sentsore guztietatik egingo da, hala ere, bakarrik ikertzaileek eskatutako datuak gordeko dira.

Android mugikorra

Diseinatu den *Android* aplikazioa bai mugikorrentzako nola tabletentzako erabilgarria izan daiteke. Aplikazioa intuitiboa izateko, ikasteko erraza eta irisgarria (adibidez, pantaileko osagaien letra handia) diseinatu da.

Android aplikazioaren diseinuaren nondik norakoak 4.1.3 atalan aztertuko dira.

Android mugikorrra eta *BITalino* txartelaren arteko komunikazioa

Sistema gailu espezifiko batez mugatuta ez egoteko diseinatu da, horretarako komunikazio modulu berezi bat sortzea erabaki da *Android* aplikazioaren barnean. Era honetan, ez da beharrezko izango aplikazio guztia aldatzea datuak eskuratzeko gailua aldatuz gero.

Sistema osoaren diseinuan azaldu den moduan (4.1.1 atalan), sistemaren moduluen arteko elkarrekintza *HTTP* protokoloa erabiliz gauzatuko da. Hori dela eta, mugikorrak *Wi-Fi* egokitzailea zerbitzariarekin komunikatzeko erabiliko du, eta horren ondorioz, komunikazio honetarako *Bluetooth* egokitzailea erabiliko da.

4.1.3 *Android* aplikazioaren diseinua

Azpi-atal honen helburua garatutako *Android* aplikazioaren diseinuaren zergatiak azaltzea da. Aplikazioaren egokitasunerako beharrezkoak izan diren faseak identifikatu eta azalduko dira.

Identifikazioa

Sistemak eskaintzen dituen zerbitzuak direla eta ezinbestekoa da *login* fase bat izatea. Erabiltzaileek bere burua identifikatzeko posta elektronikoaren helbidea edo erabiltzaile izena eta pasahitza aurkeztu behar izango dituzte. Kredentzialak zerbitzariarekin egiaztatu ondoren, hurrerantzean identifikazio gisa erabiliko den *API* gako bat itzuliko zaie. Era honetan segurtasun neurritzat erabiltzailearen pasahitzaren berrerabilpena galaraziko da.

Aurreko atalean aipatu den moduan, sistemaren modulu honek bi entitateez osatuta egongo da. Bi gailuen arteko konexio bat sortzea derrigorrezkoa izanik. Hasiera batean, proiektu honen helburua ahalik eta sistema nonahikoena sortzea zenez, identifikazio atal honek erabiltzaileentzako ahalik eta gardena izateko diseinatu da.

Beraz, aipatutako nonahikotasuna dela eta hurrengo arazo bat suertatzen da; nola gauzatu konexioa erabiltzailea eragotzi gabe? Hasierako proposamena erabiltzailearen *Android* mugikorrak *Bluetooth* bidez datu fisiologikoak eskuratzeko txartela (*BITalino*-a kasu honetan) detektatuz gero automatikoki konektatzea izan zen. Hala ere, inplementazio zailtasunak eta *Google*-ek egindako *Android API*-ren eguneraketa zela eta nonahikotasun kontzeptu hau aplikazioaren hurrengo bertsio baterako uztea erabaki zen.

Honen ordez, konexioa burutzeko, zerbitzaritik *API* gakoarekin batera, datu fisiologikoak

eskuratzeko txartelaren *MAC* helbidea bidaliko da. Era honetan erabiltzaileak ez da *Bluetooth* konexioaz arduratu beharko. Hala ere, zerbitzaritik *MAC* helbidea bidali ahal izateko aurretik gordeta izan behar da.

Aplikazioaren hasierako bertsio honetan erabiltzaileak ezin izango du bere burua sisteman erregistratu, horretarako beste proiektuan landu den ikertzailearen laguntza behar izango du. Pausu honetan ikertzaileak erabiltzailearen datu guztiak sistemaratuko ditu, baita ere, erabiltzaileak erabiliko duen *BITalino*-aren *MAC* helbidea. Aipatu beharra dago hurrengo bertsiotarako prozedura hau guztiz aldatu nahi dela, zeren eta sistemak eskaintzen dituen puntu indartsuenetako bat, kudeaketa telematikoa da.

Azkenik, beharrezkoa izango da uneko erabiltzailearen datuak gordetzea saioan zehar erabili direlako. Adibidez; *API* gakoa zerbitzariarekin komunikatzeko.

Identifikazioa eta gero menu nagusia agertuko da.

Menu nagusia

Sistemaren menu nagusia erabiltzaileak eginda edo egin beharreko atazak erakustea du helburu. Atazak eskuratzeko erabiltzailea identifikazio fasean jasotako *API* gakoa aurkeztu behar izango du. Atazak egoeraren arabera banatutak izango dira:

- Osorik eginda ez dagoen ataza baten gainean sakatuz gero atazaren deskribapen labur bat pantailaratuko da, ataza burutzeko edo ezeztatzeko aukera emanez.
- Osorik eginda dagoen ataza baten gainean sakatu ondoren ikertzaileak idatzitako esperimentuaren *feedback*-a ikusteko aukera izango da.

Menu nagusia erabiltzaileentzat gardena izateko pentsatuta dago. Pantailaratzen diren atazak automatikoki eguneratuko dira, atazaren bat burutzean edo aplikazioa irekitzerakoan. Hala ere, hurrengo bertsiotarako erloju bat gehitzea komenigarria litzateke minuturo (adibidez) ataza zerrendak eguneratzen dituena.

Menu nagusia atazen arteko nabigazioa ahalbidetzeko diseinatuta dago, era honetan, nahiz eta erabiltzaileak atazaren bat aukeratu, ataza egin gabe atzera (menu nagusira) itzultzeko aukera izango du.

Azkenik, aipatu beharra dago partehartzaile batek egin dezakeen esperimentu kopurua mugatuta ez dagoenez gerta daitekeela esperimentu guztiak pantailan ez agertzea. Hori ekiditeko, ataza zerrendetan *scroll* bertikalaren aukera jarriko da.

Datuen eskuraketa

Atazarekin hasi bezain laster, *BITalino* plakak eskuratutako datu fisiologikoak, *Bluetooth* bidez automatikoki bidaliko dira *Android* aplikaziora. Inplementazioa errazteko asmoz, sentsoreetatik egingo den datu laginketa beti maiztasun berdinean egingo dira, 100Hz-tara hain zuzen ere.

Nyquist-Shannon [12][13] teoremaren arabera, seinalearen laginketa maiztasuna, gutxienez laginketarako seinalearen maiztasun maximoaren bikoitza izan behar du. Soilik era honetan ikertzaileak partehartzailetik jasotako seinalea era zuzen batean berreraikitzeko aukera izango du.

Prestakuntza atalan adierazten den moduan, *BITalino*-k lau laginketa maiztasun ditu: 1, 10, 100 eta 1000Hz. 1 eta 10Hz laginketa maiztasunak oso baxuak dira, adibidez, bihotzaren seinalearen laginketarako. Aldiz, 1000Hz laginketa maiztasuna aukeratuz gero, eskuratutako lagin kopurua handiegia izango litzateke.

Sentsore bakoitzeko fitxategi bat sortuko da, fitxategiaren izena, bere barnean gordeta daukan seinalearena izango da. Eskuratutako laginak, testu lauan eta hurrengo formatuarekin gordeko dira: *lagina0, lagina1, lagina2, lagina3, ..., laginaN*.

Eskuratzen ari diren bitartean datu fisiologikoak ez dira pantailaratuko, hurrengo atalan azalduko den bezala, aplikazioaren helburuetako bat, partehartzaileak esperimentua egiten dauden bitartean ikertzailearekin nolabaiteko komunikazioa izatea litzateke.

Esperimentuen markak

Ikertzaileek definitutako esperimentuek hainbat ekintzez osaturik egon daitezke. Adibidez, 30 minutuz korrika egiteko esperimentuak ondoko ekintzak izan ditzake: aldapan gora, aldapan behera, atsedena, *sprint*-a, etab.

Partehartzaileak, ikertzaileari esperimentuan zehar ekintza bat gauzatzen ari dela adierazteko aukera izango du ekintzari dagokion marka (botoia) sakatuz. Ekintza bukatu ondoren, sakatutako marka bera berriz ere sakatu beharko du ekintzarekin amaitu duela adierazteko.

Sistemaren bertsio honetan, ezin da aldi berean marka bat baino gehiago aktibaturik izan.

Marka bakoitzak zenbaki kode bat izango du, aurreko adibidearekin jarraituz: *markarik ez = 0, aldapan gora = 1, aldapan behera = 2, etab.* Beraz, aipatutako sentsoreen fitxa-

tegiekin batera beste fitxategi bat sortuko da antzeko egiturarekin: *0, 0, 0, 0, ..., 1, 1, 1, 1, 1, ..., 0, 0, 0, 0, ...*

Esperimentuan erabilitako sentsore guztiak laginketa 100Hz-tara egiten dutenez, marken fitxategian maiztasun berdinarekin idatzi beharko da. Era honetan, ikertzaileak jakin izango du esperimentuak irauten duen denboran zehar, partehartzaileak zein momentutan sakatu dituen markak.

Sentsoreen egiaztapena

Datuen eskuraketa atalan azaldu den moduan, ataza burutzen den bitartean, datuak ez dira pantailaratuko. Beraz, gerta liteke ataza batekin hastea eta ez konturatzea eskuratzen ari diren dautak guztiz akastunak direla. Aplikazioaren fase honetan, ataza hasi baino lehen, erabiliko diren sentsore guztien egiaztapena egiteko aukera izango da.

Egiaztapen fase hau ez dago diseinatuta partehartzaileek eskuratutako datuak uler ditzaten, baizik eta sentsoreetatik eskuratzen ari diren datuen zuzentasuna ondorioztatu ahal izateko. Horretarako, lortutako laginak, grafiko batean pantailaratuko dira, oso erraza egiten baitu sentsorearen zuzentasuna egiaztatzea.

Hainbat izan daiteke jaso ahal diren arazoak, batez ere, txartela eta sentsoreekin zerikusia dutenak. Adibidez sentsoreak gorputzean era ez zuzen batean jarrita egotea, txartela itzalita egotea, etab.

Partehartzaileak sentsoreak egiaztatu ondoren atazarekin hasteko edo menu nagusira itzultzeko aukera izango du.

Datuen bidalketa

Esperimentua burutu ostean, eskuratutako datuak web zerbitzarira automatikoki igoko dira erabiltzaileari eragotzi gabe. Horretarako, aplikazioak esperimentuan sortutako fitxategi guztiak (fitxategi bat sentsore bakoitzeko eta markak dituen fitxategia), *ZIP* fitxategi batean trinkotu ondoren zerbitzarira bidaliko ditu.

Azkenik, zerbitzariari mezu bat bidaliko zaio bukatu berri den esperimentuaren egoera datu-basean egunera dezan.

Sistemaren bertsio honetan, *ZIP* fitxategiaren transmisioan ez da inolako errore tratamendurik egiten. Beraz, bidalketan erroreren bat izanez gero, partehartzaileak esperimentua

berriro egin beharko du.Sistemaren hurrengo bertsiotarako komenigarria litzateke kudeaketa hau hobetzea.

Hizkuntzak

Aplikazioa hiru hizkuntzetan eskuragarri izango da: euskera, gaztelera eta ingelesa. Hizkuntza desberdinak *Android*-ek eskaintzen dituen baliabideekin kudeatuko dira.

4.2 Inplementazioa

Atal honetan, aurreko diseinuaren funtzioak ahalbidetzen duen inplementazioa azalduko da. Lehenik eta behin aipatu beharra dago proiektu honetarako erabili den txartela *Android* aplikaziotik kudeatzen dela, beraz, esan daiteke proiektuan egindako inplementazioa *Android* aplikazioarena izan dela. Hala ere, txartela kudeatzeaz arduratzen diren kode zati guztiak klase batean barneratu dira, honela, plaka desberdin bat erabili nahi izanez gero bakarrik beharrezko litzateke komunikazio klase hori aldatzea.

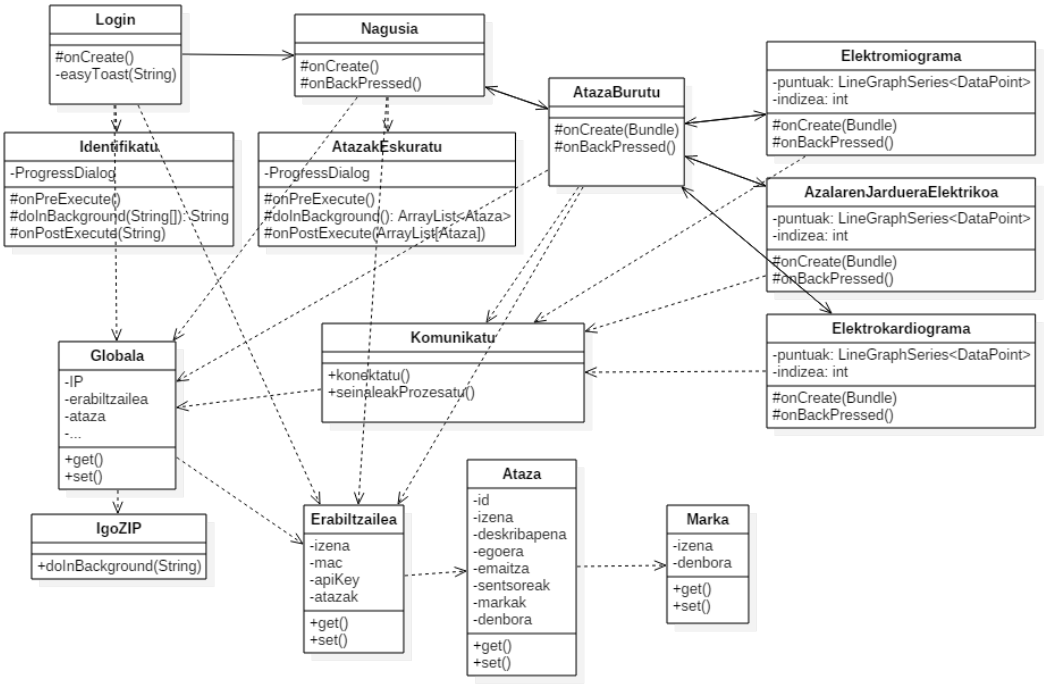
4.2 irudian *Android* aplikazioaren eskema orokorra aurki daiteke.

4.2.1 *Login* klasea

Login Java klase hau, diseinuko identifikazio fasean aztertutako eskakizunak inplementatzen ditu. Azpi-atal honen helburua, hain zuzen ere, klasearen alde nagusiak azaltzea da.

Klaseari itxura ikusteko, 4.3 irudian *Login* klasea estekatzen duen *activity*-a (*Android* pantaila) ikus daiteke. Pantaila bi eremuz osatuta izango da erabiltzaileek haien kredentzialak sar ditzaten, zehatzagoak izateko, bi `TextView` eremuak. Pasahitzaren eremuaren azpian identifikazioa gauzatzeko botoia aurkituko da.

Login botoian sakatzerakoan egiaztapen sinple bat egingo da aurreko eremuak hutsak ez direla ziurtatzeko. Ondoren, sartutako datuak zerbitzariarekin egiaztatuko dira. Kredentzialak zuzenak badira, erabiltzaileari dagokion pantaila nagusia kargatuko da, bestela, errore mezu bat pantailaratuko da. Aplikazioaren lehen bertsio honetan, ez da errore mota ezberdinduko, hala ere, bi izan daitezke errore mota nagusiak; kredentzial okerrak edo konexio errorea.



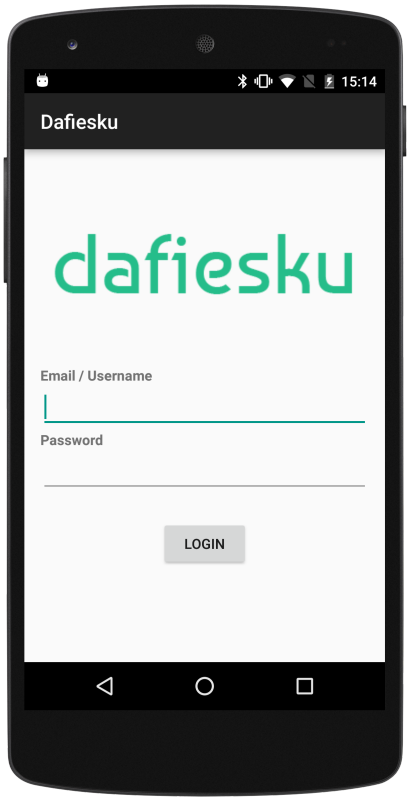
4.2 Irudia: *Android* aplikazioaren egituraren diagrama

Aipatutako egiaztatze pausu hau prozesu blokeakorra eta nahiko luzea denez, gogoratu behar da kanpo komunikazio (beste gailu batekin) bat gauzatzen ari denean, ez dela komenigarria *GUI* (*Graphic User Interface*) harian exekutatzea. Beraz, eginkizun honetarako prestakuntzan aipatutako Android `AsyncTask` klasea erabiliko da. Hona hemen klasearen atal esanguratsuenak:

Klase guztiekin gertatzen den antzera, lehendabiziko pausua klaseari dei egitea da. Dei komandoa exekutatu ondoren klasearen edukiak hari berri batean hasiko dira exekutatzen. Aipatu beharra dago egiaztatu nahi diren kredentzialak parametro bidez bidaliko direla, kasu honetan bi `TextView`-tik jasotako kredentzialak.

```
new Identifikatu().execute(iz.getText().toString(), pas.getText().toString());
```

Erabiltzailearen kredentzialak eta zerbitzariaren erantzuna *JSON* formatu batean transmitituko dira. Komunikazioa gauzatzeko erabiliko den konexioa *HTTP* motakoa izango da *POST* aukeratuz eskaera metodo gisa. Halaber, beharrezkoa izango da konexioaren eduki mota *JSON* pakete bat izango dela zehaztea. Behin konexioa eratuta, geratzen den bakarra *JSON*-a bidaltzea izango da.



4.3 Irudia: *Login* klasearen pantaila

Hurrengo kode lerroetan *JSON* objetuaren sortzea eta esleipena, konexioaren konfiguraketa, eta, kredentzialen bidalketa ikus daitezke:

```
JSONObject JSONEskaera = new JSONObject();
JSONEskaera.put("loginName", kredentzialak[0]);
JSONEskaera.put("password", kredentzialak[1]);

URL url = new URL("http://" + Globala.getIP() + "/api/v1/login");
HttpURLConnection kon = (HttpURLConnection) url.openConnection();
kon.setRequestMethod("POST");
kon.setRequestProperty("Content-Type", "application/json");

OutputStream os = kon.getOutputStream();
os.write(JSONEskaera.toString().getBytes());
```

Diseinu fasean adostu bezala, bidalitako kredentzialak zuzenak izanez gero, zerbitzariak

`HTTP_OK` kodearekin batera hurrengoetan pasahitza ordezkatuz erabiliko den *API* gakoa eta datuak eskuratuko dituen txartelaren *MAC* helbidea bidaliko ditu:

```
if (kon.getResponseCode() == HttpURLConnection.HTTP_OK) {
        InputStreamReader isr;
        isr = new InputStreamReader((kon.getInputStream());
        BufferedReader br = new BufferedReader(isr);
        String erantzuna = br.readLine();
        JSONObject erantzunaJSON = new JSONObject(erantzuna);
        erabiltzailea.setMac(erantzunaJSON.getString("bitalinoMAC"));
        erabiltzailea.setApiKey(erantzunaJSON.getString("apiKey"));
```

Bukatzeko, hurrengo klaseak erabili asmoz, erabiltzailea klase global batean gordeko da eta partehartzailea hurrengo pantailara igaroko da, pantaila nagusia alegia:

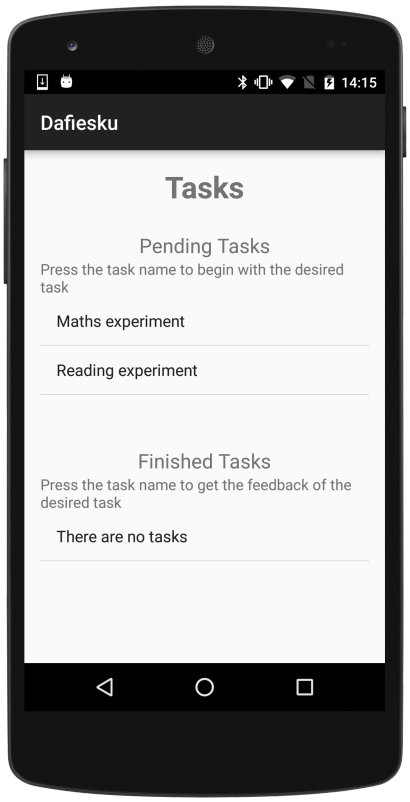
```
Globala.setErabiltzailea(erabiltzailea);
Intent nagusia = new Intent(Login.this, Nagusia.class);
startActivity(nagusia);
```

Berriz ere, aipatu beharra dago identifikazioa arrakastatsua ez bada errore mezu bat pantailaratuko dela. Aplikazioaren hasierako bertsio honetan, errore mezua onartu ondoren kredentzialak berriro sartzeko aukera eskainiko da. Inolako tratamendurik egiten ez denez, prozesu hau hainbat alditan errepika daiteke.

4.2.2 Klase nagusia

Diseinu atalean adostu den moduan, pantaila hau, erabiltzaileek atazak aukeratzeko edota egindako atazen emaitzak ikuskatzeko erabiliko da (ikusi 4.4 irudia). Bi ataza mota ezberdinak pantailaratzeko bi zerrenda erabiliko dira, zerrenda bat erabiltzaileek egiteke dituzten atazak kargateko eta bestea, bukatu dituzten atazen emaitzak ikusteko. *Login* klasearekin gertatzen zen antzera, klase honetan *HTTP* konexioa gauzatzeko `AsyncTask` klasea erabiliko da.

Hurrengo lerroan klaseari deia ikus daiteke, kasu honetan ez da kredentzialik erantsi behar. Hurrengo komandoetan ikusiko den moduan, erabiltzaileak identifikadore gisa *Login* klasean jasotako *API* gakoa erabiliko du.



4.4 Irudia: Klase nagusiaren pantaila

```
new AtazakEskuratu().execute();
```

Web zerbitzaritik atazak eskuratzeko *HTTP GET* eskaera metodoa erabiliko da. *Android*-eko `HttpURLConnection` klaseak defektuz *GET* metodoa erabiltzen duenez ez dago komando baten bidez adierazteko beharrik. Konexioarekin batera, erabiltzaileak *Login* klasean jasotako *API* gakoa bidaliko du identifikadore gisa:

```
URL url = new URL("http://" + Globala.getIP() + "/api/v1/tasks");
HttpURLConnection kon = (HttpURLConnection) url.openConnection();
kon.setRequestProperty("Authorization", "Bearer " + erabiltzailea.getApiKey());
```

Erantzuna baiezkoa bada, ataza guztiak kapsulatuta dituen *JSON* objetua deskargatuko da:

```
if (kon.getResponseCode() == HttpURLConnection.HTTP_OK){
        InputStreamReader isr = new InputStreamReader(kon.getInputStream());
        BufferedReader br = new BufferedReader(isr);
        String erantzuna = br.readLine();
        JSONArray JSONErantzuna = new JSONArray(erantzuna);
```

Diseinu atalean adierazi den moduan, pantaila honetan bi zerrenda agertuko dira, bata egiteke dauden atazekin eta bestea bukatuta daudenekin. Beraz, ataza bakoitzeko bi aukera desberdin izango dira, `egitekeAtazak` zerrendan zerrendatzea edo `bukatutakoAtazak` zerrendan gehitzea.

Ataza datu mota pantailaratu ezin denez, `String` motako zerrenda bat erabiliko da laguntzaile gisa non atazen izenak zerrendatuko diren diren:

```
for (int i = 0; i < egitekeAtazak.size(); i++)
        lista1.add(egitekeAtazak.get(i).getIzena());
for (int i = 0; i < bukatutakoAtazak.size(); i++)
        lista2.add(bukatutakoAtazak.get(i).getIzena());
```

Behin ataza guztiak era egoki batean pantailaratuta daudela, hurrengo lerroetan ataza baten gainean sakatzerakoan gertatzen dena azaltzen da:

- Alde batetik, sakatu den ataza oraindik egiteke badago, `AlertDialog` bat azalduko da atazaren izena eta deskribapen labur bat pantailaratuz. Mezuan bertan, bi botoi aurkeztuko dira, bata ezeztatzeko, eta bestea, atazara joateko.

  ```
  AlertDialog.Builder atazaraJoan = new AlertDialog.Builder(Nagusia.this);
  atazaraJoan.setPositiveButton(... // Atazara joateko botoia
  atazaraJoan.setNegativeButton(... // Mezua ezeztatzeko botoia
  ```

- Beste aldetik, eginda dagoen atazaren gainean sakatzerakoan, `AlertDialog` bat pantailaratuko da ikertzaileak esperimentuan esleitutako *feedback*-arekin.

Hurrengo bertsioetan komenigarria litzateke atal honetan egiten den atazen kudeaketa hobetzea, hain zuzen ere, hurrengo arazoari irtenbide hobeago bat emateko.

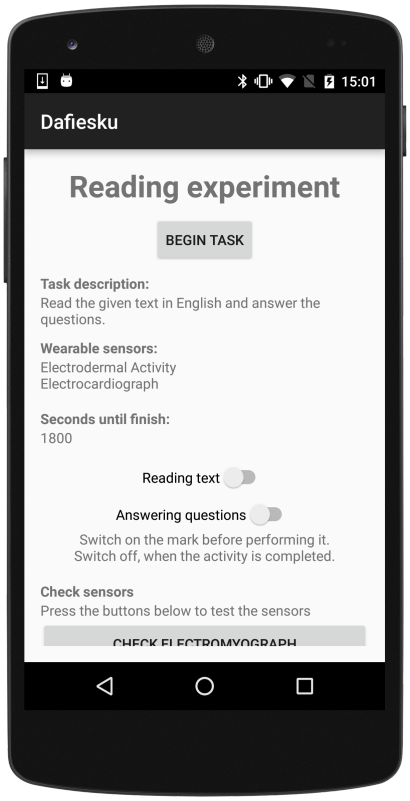
Atazaren bat burutu ostean eta klase nagusira itzultzerakoan, zerbitzaritik ataza guztiak automatikoki deskargatuko dira klase honen hasieran ikusi den `AsyncTask` klaseari deia eginez.

Komunikazioen abiadura dela eta, gerta daiteke burutu berri den atazaren eguneraketa mezua (ataza eginda dagoela) zebitzarira epean ez ailegatzea, *Android* aplikazioak une berean atazak deskargatzen izango baita. Beraz, badaezpada ere, atazen eskuraketa egiterakoan burutu den azken ataza deskargatzerakoan era berezi batean tratatuko da: nahiz eta atazaren egoera egiteke dagoela zehaztu, eginda dauden atazen zerrendan gehituko da.

4.2.3 Ataza burutzeko klasea

Klase honetan pantailaratzen diren osagaien dinamikotasuna dela eta, exekuzio denboran erabakiko baita pantailaratu beharreko osagai kopurua, inplementatzeko klaserik zailena izan da. Adibidez: burutu behar den esperimentuaren arabera pantailaratu beharreko marka kopurua desberdina izan daiteke. Atal honen helburua klasearen inplementazioaren nondik norakoak azaltzea izango da.

4.5 irudian ikusi ahal den moduan, pantaila honetan menu nagusian aukeratu den atazaren informazio guztia erakutsiko da. Gainera, ataza hasi aurretik sentsoreak egiaztatzeko aukera eskainiko da, hurrengo azpiatal batean jorratuko denez.



4.5 Irudia: Ataza burutzeko klasearen pantaiala

Markak *Android switch*-en bidez adieraziko dira. Erabiltzaileak marka adierazten duen ekintza hasterakoan *switch*-a aktibatuko du, aldiz, bukatzerakoan *switch*-a desaktibatuko du. Gainera, diseinu fasean erabaki zen bezala markak elkar bateraezinak izango dira, hau da, markaren bat sakatzerakoan besteak desgaiturik egon beharko litzateke.

Erabiltzaileak, ezin izango du marka baten gainean sakatu, marka (*switch*-a) aurretik des-

gaituta badago. Beste hitzetan esanda, ezin izango da aktibatu desgaituta dagoen marka bat. Aurreko paragrafoan azaldu den moduan, marka bat aktibatzerakoan, haien arteko elkar bateraezintasunak direla eta, beste marka guztiak desgaituko dira. Aldiz, sakatutako marka desaktibatzerakoan, pantailan aktibatutako markak izango ez direnez, beste guztiak gaituko dira.

Helburu horretarako, balio boolearrez osatutako bektore bat erabiliko da, bektore horren osagai bakoitzak dagokion markaren egoera gordeko du. Marka aktibatuta badago `true` balioa hartuko du, aldiz, `false` balioa izatea, marka desaktibatuta dagoelaren adierazgarria izango da.

Markaren bat aktibatzerakoan, markak, bektorean ezezko egoeratik baiezko egoerara iragango du. Ondoren, beste marka guztiek haien uneko egoera bektorean egiaztatuko dute; false izanez gero eta bektorean true bat detektatuz (aktibatuta dagoen markarena), bere burua desgaituko du.

Beste hitzetan esanda, marka bat aktibatu denez eta, marken arteko elkar bateraezintasunaren ondorioz, beste marka guztiek bere burua desgaituko dute.

Aldiz, marka desaktibatzerakoan, desaktibatutako markak bere egoera false-ra itzuliko du. Berriz ere, beste marka guztiek, bektoreko osagai guztiak false direla ohartuz (marka guztiak desaktibatutak), bere burua gaituko dute.

Hau da, sakatutako marka desaktibatu denez, aurreko pausuan desgaitutako markak berriz ere gaituko dira.

4.5 irudiaren goikaldean ikus daitekeen botoia sakatuz gero atazari hasiera emango zaio eta denbora dekrementatzen hasiko da.

Klase honek erabiltzalearekin elkarrekintzan izango denez, datu eskuraketa egiteaz klaseak sortuko duen hari berri bat arduratuko da. Hariak, txartelarekin komunikatzeko erabiltzen, eta, aurrerago azalduko den, klasea exekutatuko du.

```
Thread komunikatuHaria = new Thread() {
        @Override
        public void run() {
                Komunikatu komunikatu = new Komunikatu();
                komunikatu.konektatu(mac, "");
        }
```

```
};
komunikatuHaria.start();
```

Klasearekin bukatzeko, marken kopuruaren dinamikotasuna dela eta, gerta daiteke mugikorraren pantaila txikia geratzea, kasu horietan *scroll* egiteko aukera azalduko da.

4.2.4 EMG, EDA eta ECG klaseak

Hiru klase hauek, sentsoreetatik eskuratutako seinaleak pantailaratzeaz arduratuko dira. Hain zuzen ere, klase hauen helburua erabiltzaileei sentsoreen egokitasuna egiaztatzeko aukera eskaintzea izango da.

Klase hauek ataza burutzeko pantailatik bakarrik atzitu ahal dira. Erabiltzaileak, sentsoreen egiaztapena egin ondoren, mugikorreko atzera botoia sakatuz atazaren menura itzultzeko aukera izango du.

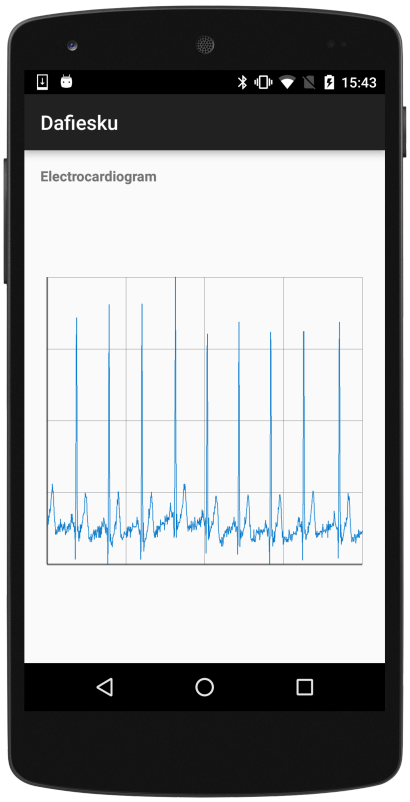
Hiru pantailek grafiko bana (4.6 irudian elektrokardiograma baten grafikoa ikus daiteke) erakutsiko dute, non aukeratutako sentsoreak eskuratutako datuak agertuko diren. Prestakuntza atalan aipatu den moduan, aplikazio honetan, grafikoak osatzeko *Graph View* liburutegia erabili da. Hala ere, beste edozein grafiko liburutegia erabiltzea posible izango litzateke, dagozkien metodoak erabiliz.

Grafikoa pantailaratzeko prozedura bi pausuz osatuta dago. Lehenengo pausua, grafikoa, inprimatu behar dituen datu segidarekin lotzea izango da.

```
        grafikoa.addSeries(puntuak);
```

Bigarrena, aipatu berri den puntu segida (`LineGraphSeries<DataPoint>` motako aldagaia) datuez osatzea izango da. Horretarako, hurrengo atalan azalduko den komunikatu klasetik eskuratzen ari diren datu fisiologikoak hiru klase hauetara denbora errealean transmitituko dira mezu trukaketaren bidez.

```
public final Handler grafikoaEguneratu = new Handler() {
        public void handleMessage(Message mezua) {
                Bundle data = mezua.getData();
                p = new DataPoint(i++, data.getInt(String.valueOf(i)));
                puntuak.appendData(p);
        }
};
```



4.6 Irudia: Elektrokardiograma klasearen pantaila

4.2.5 Komunikatu klasea

Nahiz eta hasiera batean klasea hau ataza burutzeko klasean inplementatuta zegoen, modulartasuna dela eta, banatzea erabaki da.

Memorian zehar aipatu den moduan sistema honen helburua ez da datu fisiologikoak eskuratzeko txartel espezifiko batera mugatuta egotea. Hori dela eta, edozein txartela erabili nahi izanez gero, aplikazio osoa aldatu behar ez izateko erabaki zen klase hau besteengandik banatuta inplementatzea. Beraz, datuak eskuratzeko txartela aldatuz gero, aplikazio osoan aldatu behar izango den klase bakarra hau izango litzateke.

Klase hau txartelarekin konektatzeaz eta datu fisiologikoak eskuratzeaz arduratuko da. Hona hemen klasearen egituraren azalpena:

BITalino txartela dela eta, zailtasunak zeuden *Bluetooth socket*-aren berrerabilpenaren kudeaketa egiterakoan. Beraz, komunikatu klase honi deitzen zaion bakoitzean *Bluetooth* konexio berri bat irekiko da.

Klase hau bakarrik ataza burutzeko edo sentsoreak egiaztatzeko klaseetatik dei daiteke.

Beraz, bi izango dira tratatu beharko diren kasuak; sentsoreak egiaztatzeko deia edo atazarako datu fisiologikoak eskuratzeko deia.

Egoera hori kudeatzeko *Java*-ko `Switch` kontrol egitura erabiltzea hautatu da.

```
switch (sentsorea) {
        case "EMG":
                // Komandoak
        case "EDA":
                // Komandoak
        case "ECG":
                // Komandoak
        default:
                // Komandoak
}
```

Lehenengo kasuan, komunikatu klasea sentsoreak egiaztatzeko deitzen bada, klaseari dei egiterakoan parametro gisa bidali den sentsorearen arabera, lehenengo hiru `case`-n artean aukeratuko da.

Sentsoreak egiaztatzeko bidea aukeratuz gero, datuak zuzenean aurreko atalan azaldu diren klaseetara bidaliko dira klase horiek eskuratutako laginak grafiko batean inprima ditzaten. Datu fisiologikoak ikertzaileari bidaliko ez zaizkionez, ez dira fitxategietan gordeko.

Aldiz, deia egiterakoan sentsorerik adierazten ez bada `default` bidea aukeratuko da, hau da, ataza batentzako datuak eskuratzeari ekingo zaio.

Datuak eskuratzeko suertatzen den arazo nagusia sentsoreen dinamikotasuna izango da, hau da, exekuzio denborarte ez da jakingo ataza bakoitzak erabiliko dituen sentsore kopurua.

Ataza bakoitzak bost karaktereko `String` eremu bat izango du non erabiliko diren sentsoreak adieraziko diren.

Karaktere bakoitzak sentsore bat errepresentatuko du, eta 0 edo 1 balioak izan ditzake. Karakterea 0 bada sentsoreak ez du datu eskuraketak gauzatuko, aldiz, karakterea 1 bada sentsorea datuak eskuratzeko gaituko da.

Honen ondorioz, erabiliko den sentsore bakoitzerako fitxategi bat sortu beharko da:

```
if (sentsoreak.charAt(0) == '1') {
        File fitxategiaEMG = new File(..., "elektromiograma.txt");
}
```

Baita ere, sentsore bakoitzeko datuak eskuratzen direla aukeratu behar da:

```
if (sentsoreak.charAt(0) == '1') {
        laginaEMG = frame.getAnalog(0);
}
```

Azkenik, erabili diren sentsoreen fitxategi guztiak ZIP fitxategi batean trinkotuko dira eta mezu trukaketaren bidez klase globalera bidaliko dira klase globalak web zerbitzarira bidali dezan.

4.2.6 Klase globala

Klase globala uneoro exekuzioan izango den klase bat denez, haren eginkizun nagusia aplikazioaren egoera gordetzea izango da, adibidez: uneko erabiltzailea, uneko ataza, etab.

Klase honek *get* eta *set* metodo publikoez osaturik egongo da. Horrez gain, uneoro exekuzioan dagoenez, komunikatu klasetik eskuratutako datuak zerbitzarira igotzeaz arduratuko da.

Komunikatu klasetik datu fisiologikoak iristean, klasean eten moduko bat sortuko du eta klasea fitxategien bidalketarekin hasiko da. Horretarako, `AsyncTask` klasea erabiliz hari berri bat sortuko da, klase globalak ezin baitu beste klaseei zerbitzatzeari utzi.

Hari berriak, zerbitzariarekin *HTTP Post* konexio berri bat irekiko du *ZIP* fitxategia bidali ahal izateko. Behin konexioa eratuta dagoela, irekiko duen `OutputStream` batean *ZIP* fitxategiaren edukia idazten hasiko da.

Bidalketa bukatu ostean, haria `HTTP_OK` mezuaren zain geratuko da. Zerbitzaritik mezua iristean, *Android* aplikazioak zerbitzariri atazaren egoera eguneratzeko mezua bidaliko dio. Horretarako *HTTP PUT* eskaera mota erabiliko da.

4.3 Balidazioa

Sistemaren erabilgarritasuna egiaztatzeko sistemaren balidazio bat egin da maiatzean. Balidazioan, ikertzailearen eta partehartzailearen rolak aztertu dira.

Alde batetik, ikertzailearen paperan izango den subjektu esperimentala, esperimentua bat sortuko du beharrezkoak diren datuak sartuz. Beste aldetik, partehartzaile gisa arituko den subjektu esperimentala, diseinatu berri den esperimentua gauzatuko du haren datu fisiologikoak jasotzen direlarik. Bi aplikazioen (web aplikazioa eta *Android* aplikazioa) erabilgarritasuna *SUS* galdetegien [14] bidez neurtuko da.

4.3.1 Esperimentua

Partehartzaileak

Esperimentua, Informatika Fakultatean dauden laborategietan aurkitzen ziren 5 partehartzailekin (2 emakumeak) burutu zen. Partehartzaileen adina 30 eta 57 urte artean zegoen (41.2 ± 10), guztiek esperimentua burutzeko adostasuna eman zuten.

Gailuak

Web aplikazioa eta web zerbitzua exekutatu dira *Ubuntu 14.04 LTS* sistema eragilea duen makina birtual batean, eramangarri batean instalatuta. Eramangarri hau erabili da ikertzailearen aldea balidatzeko.

Datuak eskuratzeko *Android* aplikazioa *Samsung Galaxy Tab 2.7.0* (RAM: 1GB; 1.0 GHz dual core processore; 7"pantaila 1024 * 600 pixelekin) tabletan abiarazi da. Erabili den *Android* sistema eragilearen bertsioa 4.2.2 izango da. *BITalino* plataforma arduratu da datu fisiologikoak eskuratzeaz, *ECG* eta *EDA* sentsoreak erabiltzen direlarik.

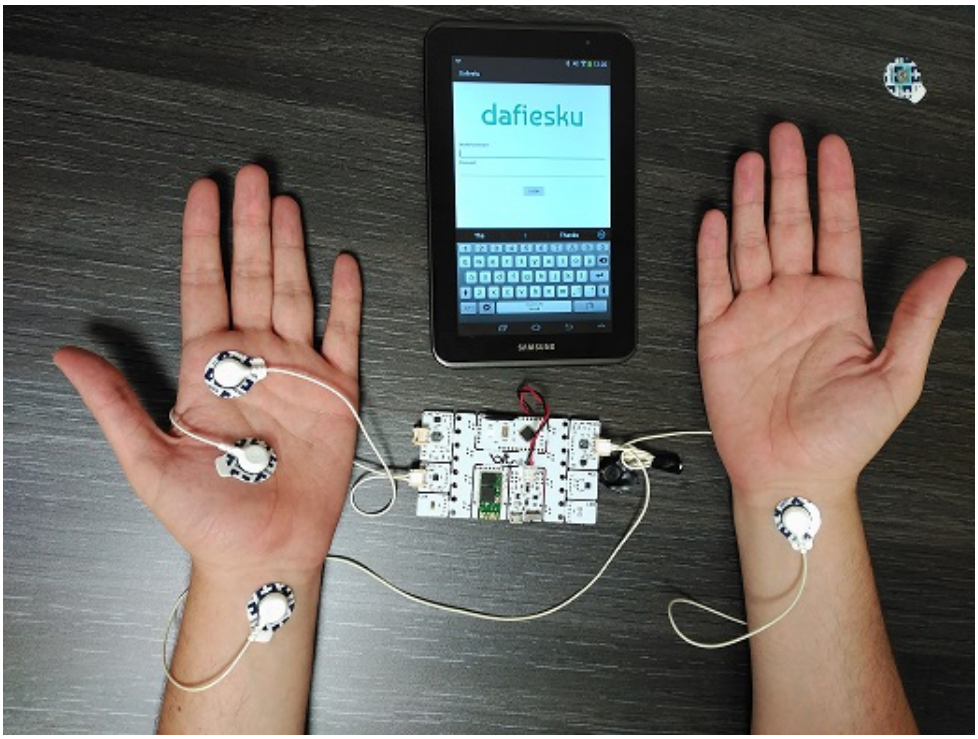
Makina birtuala alokaturik duen eramangarria eta *Android* tableta *Wi-Fi* bidez konektatuko dira *Wi-Fi* bideratzaile bat erabiliz. Alidz, *Bitalino* eta tableta *Bluetooth* bidez konektatuko dira.

Prozedura

Subjektu esperimentalen baimena jaso eta gero, inkesta demografiko bat egin zaie. Ondoren, burutu beharreko atazen azalpenak paper batean eman zaizkie.

Lehenengo atazan, ikertzaileak esperimentu bat definitu behar izan du sistemaren web aplikazioaren bidez, esperimentua, emandako paperean zehaztutako ezaugarriekin osatu da.

Ondoren, partehartzaileak sentsoreak jantzi ditu eta egiaztatu behar du aurreko atazan diseinatutako esperimentua *Android* mugikorrean atzitu daitekeela eta ondo definituta dagoela. Ataza honetan, partehartzaileak testu bat irakurri ostean, testuarekin zerikusia duten hainbat galdera erantzun behar du. Ekintza hauek paper eta boligrafoz burutu behar dira.



4.7 Irudia: Balidazioan partehartzaileak erabilitako gailuak.

Aplikazioa erabiliz, partehartzaileak adierazi behar izan du noiz dagoen testua irakurtzen, galderak irakurtzen edo galderak erantzuten. Behin ataza bukatuta, erabiltzaileak jantzitako sentsoreak kendu ditu.

Berriz ere ikertzaile gisa, web aplikazioaren bidez bigarren atazan eskuratutako datu fisiologikoak ondo igo direla egiaztatu behar du, hirugarren ataza alegia.

Ataza bakoitza burutzean emandako denbora neurtu da. Hiru atazak bukatzerakoan subjektu esperimentalek bi *SUS* galdetegi burutu behar izan dute, bata ikertzaile eta bestea esperimentu fisiologikoaren partehartzaile gisa.

Bukatzeko, sistema garatu duten bi pertsonekin elkarrizketa labur bat egin da sistemaren iritzi zabalago bat jasotzeko.

4.3.2 Emaitzak eta ondorioak

Subjektu esperimental guztiek eskatutako ataza guztiak amaitu zituzten. 4.1 taulan partehartzaileek hiru atazak bukatzeko behar izandako denbora ikus daitezke. Burutzeko denbora gehien behar izandako ataza bigarrena izan da, gutxiena hirugarrena. Hiru atazak burutzeko denborak, 25 minutu (User07) eta 44 minutu (User03) artean daude.

Erabiltzailea	1. pausoa	2. pausoa	3. pausoa	Guztira
User02	5’	21’	1’	27’
User03	7’	34’	3’	44’
User04	4’	21’	2’	27’
User05	5’	26’	2’	33’
User07	8’	16’	1’	25’
Batezbestekoa	6’	24’	2’	31’
DE *	1,47	6,09	0,75	6,94

* Desbideratze estandarra.

4.1 Taula: Balidazio atazak burutzeko beharrezko denborak.

Sistemaren erabilgarritasunari erreparatuz gero, 4.2 taulan *SUS* galdetegien emaitzak ikus daitezke. Ikertzaileek erabilitako web aplikazioa 78,5 $\pm$ 11,25 puntuko balorazioa jaso du eta Android aplikazioa 71 $\pm$ 15,62 puntu.

Bi aplikazioen kalifikazioak 70 puntutik gorago daudenez, esan daiteke bi aplikazioak erabilgarriak [15] direla. Hala ere, bi partehartzaileek aplikazioei 70 puntu baino gutxiagoko balorazioa eman dizkiote. Horietako bat, *Android* aplikazioari 42.5 kalifikazioa emanez. Honek, oraindik erabilgarritasun ezak daudelaren adierazgarria litzateke.

Subjektu esperimentalek hobekuntza batzuk iradoki zituzten, batez ere, datu eskuraketan erabiltzaile esperientzia hobetzeko asmoz.

Erabiltzailea	Ikertzailea	Partehartzailea
User02	77,5	80
User03	65	77,5
User04	67,5	42,5
User05	92,5	87,5
User07	90	67,5
Batezbestekoa	78,5	71
DE *	11,25	15,62

* Desbideratze estandarra.

4.2 Taula: *SUS* galdetegietako kalifikazioak.

Partehartzaile batek (User04), aplikazioak sentsoreak nola janzteko argibideak ekartzea iradoki zuen. Baita ere, hainbat subjektu esperimentalek sentsore fisiologikoen kableak zirela eta, bigarren ataza burutzeko zailtasuna izan zutela aipatu zuten. 4.7 irudian partehartzileak jantzi beharreko sentsoreak ikus daitezke.

Garatu den sistemaren helburua datuak eskuratzea denez, egokia litzateke hurrengo bertsioetan *Android* aplikazioak sentsoreak nola janzteko argibideak ekartzea. Aldiz, esperimentuan zehar aipatutako mugikortasun ezak proiektuaren helburuetatik kanpo aurkitzen dira, arazo hori erabilitako datu fisiologikoak eskuratzeko txartelarekin zerikusia baitu. Hala ere, kontuan izango da datu fisiologikoak eskuratzeko erabiltzen diren teknologien mugikortasun aukerak (adibidez, txartela eramateko zorroak, etab.) sistemaren erabilgarritasunean eragin negatiboa ez izateko.

Balidazioaren ondorioz, esperimentazioan aurkitutako hainbat alderdi zuzendu edo hobetu dira, balidazioko partehartzaileen iritziak eta komentarioak kontuan hartuz. Gainera, probak egin eta gero, kodean aurkitutako beste arazo batzuk ere konpondu dira.

Alde batetik, 4.5 irudian ikus daitekeen marken arteko hutsuneak oso txikiak zirenez, hainbat erabiltzaileek arazoak izan zituzten esperimentuen ekintzei zegozkien markak aktibatu edo desaktibatzeko. Erabiltzaileek, erosotasun falta zegoela aipatu zuten eta, bertsio berrian, marken arteko hutsuneak handitu dira.

Beste aldetik, erabiltzaile batzuk sentsoreak egiaztatzeko orduan, tabletaren orientazioa aldatzea nahiko desatsegina zela adierazi zuten. Inplementazio garaian uste izan zen, sentsoreek eskuratutako datuen grafikoak erakustean egokiena tableta orientazio horizontalera aldatzea izango zela grafikoa handituz eta pantaila hobeto aprobetxatuz. Azkenean, erabiltzaileen aholkuak jarraituz, aplikazio osoan orientazio bertikala mantentzea erabaki da.

Baita ere, uneko bertsioan konponduta dagoen fitxategien izenekin errore bat aurkitu zen. Nahiz eta ikertzaileak, datu eskuraketa ECG sentsoretik egitea adieraziz, eskuraketa, ACC sentsoretik gauzatzen zen.

Bukatzeko, esperimentazioan zehar kode errore larri bat aurkitu zen sistemaren funtzionamenduan: aplikazioarekin sentsoreak egiaztatu eta gero atazarekin hasterakoan, aplikazioak funtzionatzeari uzten zion. Errore hau, 3.4 ikus daitekeen *Android* pantailen bizitza zikloarekin zerikusia zeukan, hain zuzen ere, sentsoreak egiaztatzeko pantailatik ataza burutzeko pantailara bueltatzerakoan pantaila ez zen gelditzen (*Stopped*), sistematik guztiz ezabatzen zen (*Destroyed*). Beraz, pantaila aldaketa egiterakoan, pantaila, sisteman berriz ere sortu behar zen.

5. KAPITULUA

Ondorioak eta etorkizunerako lana

Azken kapitulu honetan proiektuaren garapenean zehar ondorioztatutako puntu ezberdinak komentatuko dira. Gainera, proiektu honek izan ditzakeen hobekuntzak edo etorkizunean egin daitezkeen gehigarriak azalduko dira.

5.1 Ondorioak

Proiektuaren ikuspuntu orokor bat hartuta, hasieran ezarritako helburu nagusiak bete direla esan daiteke.

Nahiz eta proiektuaren irismenean zehaztuta ez egon, proiektu hau nonahiko konputazioarekin estu erlazionatuta zegoela aipatu beharra dago. Hasiera batean, proiektuaren tutoreekin nonahiko konputazioaren eredua jarraitzen duen sistema bat garatzea adostu zen. Hala ere, ideia hau laster konputazio mugikorragatik ordeztu izan zen. Izan ere, dedikazio aldetik ezinezkoa litzateke dimentsio horietako sistema GAP batean inplementatzea. Honen inguruko ideiak 5.2 atalan aurki daitezke.

Erabilpenari dagokionez, hainbat izan daitezke sistemak eskaintzen dituen aukerak, hala nola:

- Ikerkuntza arloan ikertzaileek subjektu esperimentalen datu fisiologikoak eskuratzeko.
- Medikuntza arloan medikuek pertsonen osasun jarraipena egin ahal izateko.

- Kirol arloan kirolarien dau fisiologikoak edozein ingurunean eskuratzeko.

Bestalde, sistemaren garapena bi ikasleren artean egiteak abantailak eta desabantailak dauzka. Alde batetik, proiektuaren hainbat ataletan elkar lagundu gara. Bestetik, ordutegiak eta lan erritmo bera jarraitu behar izan dugu.

Esperientzia ona izan da guretzat landutako sistemari buruz artikulu bat UCAmI-2016 kongresura bidaltzeko aukera eduki izana eta bertan onartua izatea (ikus memoria honetako A eranskina). Gainera, artikulurako egindako sistemaren balidazioak garapena hobetzeko lagundu gaitu, aplikazioen erabilgarritasuna neurtuz.

GAP honetan aldez aurretik ezezagunak ziren hainbat teknologia berriak ikusi eta haiekin lan egiteko gaitasuna erakutsi ahal izan dugu. Aipatu beharra dago teknologia hauen erabilpena gaur egungo informatika munduan oso erabiliak direla.

Azkenik, eskertzekoa izan da tutoreez gain, Borja Gamecho doktorearen laguntza eduki ahal izana. Aditu gisa, hainbat kontuetan lagundu gaitu, batez ere *Android* aplikazioarekin, txartelen kudeaketarekin eta sistemaren balidazioarekin.

5.2 Etorkizunerako lana

Gradu Amaierako Proiektua amaitu den arren, honi jarraipena eman diezaioketen lan dezente egin daitezke, batez ere, sistemaren portaera findu eta erabilpena zabaltzeko. Hona hemen proposatutako hobekuntza posible batzuk:

Aurreko atalan aipatu den moduan, proiektu hau nonahiko konputazioaren ideia kontuan hartuta diseinatu zen hasiera batean. Honako helburuak lortu nahi ziren:

- Datu fisiologikoak eskuratzeko txartelaren bat detektatuz gero automatikoki konektatzea.
- Edozein datu fisiologikoen txartela izanda datu fisiologikoak eskuratzeko aukera izatea.
- Gailu desberdinak erabiliz erabiltzaileek sisteman egin ditzaketen ekintzak berdinak izatea, adibidez, partehartzaileek bai mugikorra nola konputagailutik esperimentuak burutzeko aukera izatea.
- Laino sistema baten erabilpena: datu fisiologikoak biltegiratu eta edozein lekutik atzitzeko aukera izatea.

Etorkizunerako lan bezala, interesgarria izan daiteke ezaugarri hauen inplementazioa egitea, sistemaren erabilpen aukerak zabalduz.

- *Android* aplikaziotik erabiltzailea sisteman bere burua erregistratzeko aukera izatea.
- Aplikazioa *Android* sistema eragilean gehiago barneratzea, adibidez: ikertzaileak erabiltzaileari atazaren bat esleitzen dion bakoitzean *Android*-eko notifikazio panelan mezu bat agertzea.
- Aplikazioak balidazioan aipatutako sentsoreak janzteko argibideak ekartzea.
- Azkenik, aplikazioaren funtzionamendua egonkorragoa izan dadin, GAP memoria honetan adierazitako hainbat puntutan kodearen egitura hobetzea gomendatzen da.

Eranskinak

A. ERANSKINA

UCAmI-2016 kongresurako idatzitako artikulua

Oharra: Han argitaratuko dena aldaketatxo batzuk izan ditzake memoria honetan eransten den bertsioarekiko.

Physiological Data Acquisition System Based on Mobile Computing

Ezekiel Sarasua, Maider Simón, Borja Gamecho, Edurne Larraza-Mendiluze and Nestor Garay-Vitoria

Egokituz Laboratory, Informatika Fakultatea (UPV/EHU); Manuel Lardizabal pasealekua 1; E-20018 Donostia (Gipuzkoa)
nestor.garay@ehu.eus

Abstract—The way to achieve enough data for data mining is accessing existing databases or directly acquiring data with the aim of creating new databases. In this paper we present the DAFIESKU system built to acquire different types of physiological data via experiments and the factors taken into account when developing it, in order to facilitate the creation of new datasets by means of mobile and wearable devices. DAFIESKU has been evaluated on a case study associated to non-classroom learning.

Keywords: Physiological data acquisition; Data storage; Mobile and Wearable Computing; Design of experiments.

1 Introduction

Before analyzing or applying machine learning techniques to physiological signals, datasets have to be obtained. There are a huge number of public repositories storing datasets which are available online. Nevertheless, in some cases researchers need to create new datasets to work with, being the task of acquiring the datasets an essential step in the research process.

The way data are obtained is an important issue related to the validity of considered data. The main approaches to record data are in controlled environments such as research laboratories or in uncontrolled environments such as in real life.

In this paper authors review topics to be taken into account and present a system aimed at acquiring physiological data on a mobile uncontrolled environment named DAFIESKU. This system allows researchers to conduct experiments remotely with the collaboration of participants, taken ethical and privacy issues into account.

2 Related Work

It is not surprising to find many frameworks, toolkits and platforms devoted to the remote evaluation in the scientific literature [1, 2]. Last decade, the high popularity of smart phones and its applications, commonly called apps, presented an opportunity to

make remote evaluations in mobile environments (for instance, Funf [3] and Aware [4]). In recent years, the use of physiological signals has emerged in the computer science area due to the vast amount of wireless wearable devices that provide such signals. Due to this fact, we can find in the literature similar systems to record and store physiological signals such as Biosignal Ignitor toolkit [5] and Physiodroid [6].

The main objective of DAFIESKU is to provide resources and tools to researchers in order to facilitate the deployment of an experiment from its design to the analysis of the results, which are not usually covered in the references found in literature.

3 Design issues

Design issues on the development of DAFIESKU system are here explained.

3.1 Mobile Sensors, Data Acquisition and Transmission

Nowadays, sensors are being more comfortable due to miniaturization. Normally, these sensors are integrated in all-in-one hardware platforms, such as Biosignalsplux [7] or Shimmer [8]. Nevertheless, these platforms still have limited storage capabilities and personal area network connectivity.

In order to overcome these limitations, a usual solution is to combine sensor platforms with mobile phones carried by users. Thanks to mobile devices, data obtained from sensors are transmitted first to the storage of the mobile device and finally to a system server available on the Internet. The main idea of DAFIESKU system is to make possible the acquisition of data in real life, without limiting it to controlled environments. These data will better reflect user natural characteristics and behavior.

3.2 Ethical Data and Privacy for Storage

Ethical and privacy issues have also to be taken into account. Physiological data are in most of the cases personal and have to be securely stored in order to ensure the people behind these data are not identified.

3.3 Further Data Analysis

Once all the ethical and privacy issues considered, data should be processed and filtered, and in order to make further analysis easier, standards, such as HDF5 [9] followed.

4 DAFIESKU System

DAFIESKU system has been designed to take advantage of wireless devices in order to obtain the less intrusive environment for the participants in the data acquisition. DAFIESKU consists of a set of wearable devices with physiological sensors coordi-

nated by a server. Wearable devices with wireless connections usually offer Bluetooth based connectivity which could be used to send physiological data to any computer. For this reason, a mobile device (e.g. smart phone or tablet) is used as intermediary between the server and the wearable devices (see Fig. 1). The main advantage of this approach is the possibility it gives to configure the data acquisition in each experiment and to guide and support the participants during the experiments.

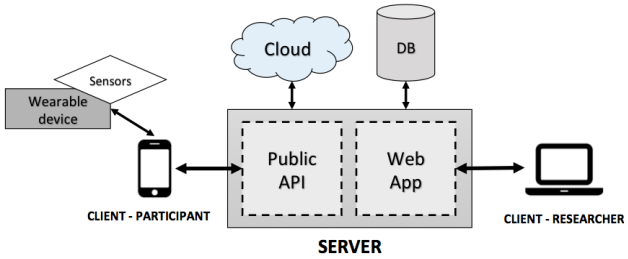


Fig. 1. Architecture of DAFIESKU system.

4.1 Configuring Data Acquisition

With DAFIESKU, researchers may establish which data, how and for how long will be acquired for creating databases with physiological information. The data to be gathered depends on the scope of the studies that the researchers plan to develop in the future, the sensors that are available for acquisition, the population, etc.

DAFIESKU system gives the researchers a web client, which offers the possibility to define how sensors (among the available ones) will be used, whether they will be used together or separately, and which kind of information will be recorded.

4.2 Providing guidance to the Participants

In DAFIESKU system, participants play a key role in the data acquisition. Among others, they decide when to start the experiments and to provide feedback to label the physiological data. In order to facilitate these activities, the researcher must also: 1) specify the experiment description and define the tasks for the participants, 2) describe the instructions to attach the sensors, and 3) set a list of predefined marks or activities related to the experiment to label the data set. All these texts must be comprehensive for the participants in order to avoid problems during the experimentation.

4.3 Experiment development

The participants will be able to download the experiment to their Android mobile device (smart phone or tablet). Once there, they will have to set the sensors and connect them to the mobile device via Bluetooth. Finally they will have to start recording data and labeling moments, following the instructions of each experiment. These last two will be mere “push button” or “select from list” tasks, respectively.

Tasks are classified as being pending or finished, depending on their status. In order to perform a task, it has to be selected from the pending tasks list. Clicking on a finished task name you can get the task feedback.

Once the user has selected a task from pending list, a screen shows all the information regarding to the selected task (description, sensors participant has to wear, maximum time estimation, and activities to perform). In order to make correct recordings, the user can check whether sensors are correctly adjusted before performing activities related to the tasks.

5 Case study to evaluate DAFIESKU system on non-classroom learning experimentation

A case study was carried out to test the adequacy of the described system. For this purpose a representative experiment on non-classroom learning was designed for DAFIESKU in order to test both roles in the system. As a researcher, the experimental subject introduced the information to create the experiment. On the other hand, as a participant of experiments, the same experimental subject performed fixed tasks while his/her physiological data were acquired. The usability of both web and mobile applications was measured by using SUS questionnaires [10].

5.1 Method

Participants and Apparatus

5 volunteers (2 females) were recruited from the surrounding research laboratories of the Faculty of Informatics of the University of the Basque Country (UPV/EHU). The experimental subjects ranged from 30 to 57 years old (41.2±10). Informed consent was obtained from all individual experimental subjects included in the study.

The Web service for designing experiments run on a virtual machine with Ubuntu 14.04 LTS operating system. This Web service was used when playing researcher role in this experiment using a laptop. Designed Apps for acquiring data run on a Samsung Galaxy Tab 2.7.0 with Android 4.2.2 operating system (RAM: 1 GB; 1.0 GHz dual core processor; 7" screen with 1024*600 pixels). BITalino [11] sensor platform was used to collect physiological data from the Electrocardiography (ECG) and Electro Dermal Activity (EDA) sensors. The virtual machine running on the laptop and the tablet were connected wirelessly using a WiFi router. The tablet and the BITalino were connected via Bluetooth interface.

Procedure

After consent has been obtained from experimental subjects, their demographic data were gathered. Then, a paper with the tasks to complete was delivered to each experimental subject.

Task 1 was defining an experiment with certain characteristics by using the Web client of DAFIESKU. This was made with the researcher role. Task 2 was made as

participant. They prepared the equipment needed in the experiment, verified they worked correctly and then they made the activities needed on the experiment defined on Task 1. This experiment consisted on reading a text printed on paper, and then reading and answering several questions related to that text, also on a paper and using a pen, like they were making exercises out of classroom. Participants had to indicate by using the system when they were reading the text, when reading questions and when answering questions. After the experiment finished, they had to put out sensors. As a researcher again, on Task 3 they verified by using the Web client again that the data of the experiment had been stored on the server.

When finishing these three tasks, experimental subjects completed two SUS questionnaires, one as researcher and the other as participant of physiological experiment. Finally, they were interviewed by up to two DAFIESKU design team members in order to get more feedback.

Design

Experimental subjects had to follow a routine composed of the abovementioned three tasks. Time required to complete each task were measured. The experimental subjects provided qualitative feedback by means of two SUS questionnaires, one for the researcher role and the other for the participant role, and the final interviews.

5.2 Results and Discussion

All the experimental subjects successfully completed tasks required. Task 2 was the one needing more time while Task 3 was the one needing less time. Minutes needed for completing three tasks went from 25' (User07) to 44' (User03). Concerning the usability of the system, the SUS scores were 78,5±11,25 for the researchers Web client application and 71±15,62 for the Android application. These scores are over 70 and therefore both applications may be considered acceptable from the usability point of view [10].

Experimental subjects also suggested several enhancements, almost with the aim of making a better participant experience for data acquisition. As DAFIESKU aims to create experiments and register data, including in the APP help or instructions to correctly adjust sensors is a good idea for future versions.

6 Conclusions and Future Work

In this paper we have presented DAFIESKU system, developed to configure data acquisition, do the collection and finally store data. Its initial aim is preparing experiments aimed at being made by students to analyze their physiological characteristics while non-classroom learning, but it could be generalized to other experimental topics and settings.

A case study has been carried out to test the validity and usability of the described system. Participants were doing fixed tasks such as reading texts and questions, and

answering questions. Achieved results are promising and show several enhancements that are being currently made on DAFIESKU system.

Next, a new version of DAFIESKU will be deployed to define experiments and acquire physiological data. Once the acquisition has been finished, data will be filtered and processed, considering all the ethical, legislation security and validity issues, in order to make data available for further research.

Acknowledgements

This work has been supported by the Ministry of Economy and Competitiveness of the Spanish Government and by the European Regional Development Fund (projects TIN2013-41123-P and TIN2014-52665-C2-1-R), and by the Department of Education, Universities and Research of the Basque Government under grant IT395-10. Last three authors belong to the Basque Advanced Informatics Laboratory (BAILab), grant UFI11/45, supported by the University of the Basque Country (UPV/EHU). B. Gamecho is supported by "Convocatoria de contratación de doctores recientes hasta su integración en programas de formación postdoctoral en la UPV/EHU 2015".

References

1. Paternò, F. (2003). Tools for remote web usability evaluation. In HCI International (pp. 828-832).
2. Christos, F., Christos, K., Eleftherios, P., Nikolaos, T., & Nikolaos, A. (2007). Remote usability evaluation methods and tools: A survey. In Proceedings of the 11th Panhellenic Conference in Informatics (PCI 2007) (pp. 151-162).
3. Aharony, N., Pan, W., Ip, C., Khayal, I., & Pentland, A.. Social fMRI: Investigating and shaping social mechanisms in the real world. Pervasive and Mobile Computing, 7(6), 643-659. (2011)
4. Ferreira, D.: Aware: A mobile context instrumentation middleware to collaboratively understand human behavior. In: Ph.D. dissertation, University of Oulu, Fac- ulty of Technology. (2013)
5. Silva, H. P., Lourenço, A., Fred, A., & Martins, R. (2014). BIT: Biosignal igniter toolkit. Computer methods and programs in biomedicine, 115(1), 20-32.
6. Banos, O., Villalonga, C., Damas, M., Gloesekoetter, P., Pomares, H., & Rojas, I. (2014). Physiodroid: Combining wearable health sensors and mobile devices for a ubiquitous, continuous, and personal monitoring. The Scientific World Journal, 2014.
7. Biosignalsplux. http://biosignalsplux.com/index.php/en/
8. Shimmer. http://www.shimmersensing.com/
9. M. Folk, G. Heber, Q. Koziol, E. Pourmal, and D. Robinson, An overview of the HDF5 technology suite and its applications. In Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases, ACM, 2011, pp. 36-47.
10. Brooke, J. (2013). SUS: a retrospective. Journal of usability studies, 8(2), 29-40.
11. H. Silva, A. Fred, and R. Martins. Biosignals for Everyone. Pervasive Computing, IEEE 13(4), 2014, pp. 64-71.

B. ERANSKINA

Erabilpen gida

Eranskin honetan garatutako *Android* aplikazioaren erabilpen gida aurkitu daiteke.

Lehenengo pausua, zure *Android* mugikorrean edo tabletan Dafiesku aplikazioa kokatzea eta irekitzea izango da.

Aplikazioa exekutatu bezain laster ohiko *login* pantaila (ikusi 4.3 irudia) bat agertuko da non, zure kredentzailak sartu beharko dituzun. Lehenengo eremuan sisteman erregistratzeko erabili duzun posta elektronikoa edo erabiltzaile izena idatzi behar izango duzu; beste eremuan, aldiz, zure pasahitza.

Login egin ondoren menu nagusia (ikusi 4.4 irudia) agertuko zaizu. Menu honetan, eginda eta oraindik egiteke dituzun atazak pantailaratuko zaizkizu.

- Alde batetik, egiteke dagoen atazaren gainean sakatzen baduzu, sakatu duzun atazaren deskribapen laburra agertuko zaizu. Deskribapen honekin batera, atazara joateko edo berriz ere menu nagusira itzultzeko aukera izango duzu.
- Beste aldetik, eginda dagoen atazaren batean sakatuz gero, ikertzaileak atazari jarritako balorazioa ikusi ahal izango duzu.

Ataza burutzea aukeratu baduzu, pantalla berri bat agertuko zaizu (ikusi 4.5 irudia). Pantaila honetan, ataza burutzeko egin beharrekoa eta jantzi beharreko sentsoreak agertuko zaizkizu, baita ere, atazaren iraupena (segundutan).

Baita ere, ataza burutzen zaren bitartean egiten ari zaren ekintzak ikertzaileari adierazi

ahal izango dizkiozu pantailan aurkitzen diren *switch*-ak aktibatuz. Ekintzarekin bukatzen duzunean gogoratu *switch*-a desaktibatzeaz .

4.5 irudiaren behekaldean dauden botoietan sakatuz gero, erabiliko diren sentsoreen egokitasuna egiaztatzeko aukera izango duzu. Sentsoreak egiaztatu ondoren, atazarekin hasteko pantaiaren goikaldean dagoen *ataza hasi* botoia sakatu behar izango duzu.

Ataza hasi aurretik ziurtatu *BITalino* txartela piztuta dagoela eta sentsoreak era egoki batean jantzita dituzula.

Behin atazarekin hasi zarela denbora dekrementazen joango da eta datu eskuraketa automatikoki hasiko da. Ataza bukatzerakoan, bai denbora bukatu delako, bai *Android*-eko atzera botoia sakatu duzulako, aplikazioa menu nagusira itzuliko da eskuratutako datuak automatikoki bidaltzen direlarik.

Azkenik, aplikazioa *Android*-eko *home* botoia sakatuz itxi dezakezu.

Oharra: Aplikazioan zehar erabiltzen diren datu pertsonalak, LOPD-a errespetatuz tratatuko dira.

C. ERANSKINA

Bilera aktak

Proiektuaren garapenaren zehar egindako bilera guztien nondik norakoak bildu dira eranskin honetan.

Bilera guztiak leku berean eta partehartzaile berdinekin egin dira:

Lekua: Informatika Fakultateko 2.1 mintegia

Bilerara hurbilduak: Ezekiel Sarasua, Maider Simón, Nestor Garay eta Edurne Larraza

C.1 Konstituzio bilera

Data: 2015/06/05, 13:00

Helburuak

Proiektuaren nondik norakoak komentatu.

Egin beharrekoak

- Ezekielek eta Maiderrek:
  Proiektuaren irismena finkatu: lana bi ikasleen artean banatu eta amankomuna finkatu.

C.2 Lehenengo bilera

Data: 2015/09/04, 15:00

Helburuak

Proiektuaren irismenaren berrikuspena.

Egin beharrekoak

- Ezekielek eta Maiderrek:
  Proiektuaren Helburuaren Dokumentua landu.

C.3 Bigarren bilera

Data: 2015/09/17, 15:00

Helburuak

Proiektuaren Helburuaren Dokumentuaren berrikuspena. Zuzendariei proiektuan lan egiteko toki bat eskatu.

Egin beharrekoak

- Ezekielek eta Maiderrek:
  Proiektuaren Helburuaren Dokumentua landu.
- Nestorrek eta Edurnek:
  Libre dagoen laborategiren bat bilatu.

C.4 Hirugarren bilera

Data: 2016/02/01, 10:30

Helburuak

Prestakuntza fasearen berri eman eta azken berrikuspena egin.

Egin beharrekoak

- Ezekielek eta Maiderrek:
  Garapenarekin hasi eta Proiektuaren Helburuen Dokumentuaren azken zuzenketak egin.

C.5 Laugarren bilera

Data: 2016/02/22, 10:45

Helburuak

Hasierako demo bat erakutsi, Proiektuaren Helburuen Dokumentuaren berrikuspena komentatu eta proiektuan eman beharreko hurrengo pausoak argitu.

Egin beharrekoak

- Ezekielek:
  BITalino eta *Android* arteko elkarrekintzarekin lan egiten jarraitu eta sentsoreetatik jasotako datuak tratatu eta modu argiago batean pantailaratu.
- Maiderrek:
  Erabiltzaileak sisteman erregistratzeko atalan lan egin eta funtzionalitate berriak gehitu.

C.6 Bosgarren bilera

Data: 2016/04/04, 10:45

Helburuak

Zalantzak argitu. Bigarren demo bat erakutsi. Orain arte egindakoa azaldu. Egiteke geratzen dena antolatu. Proiektuan eman beharreko hurrengo pausoak argitu. Proiektua ICNR2016 [16] kongresuan aurkeztearen inguruan hitz egin.

Egin beharrekoak

- Ezekielek eta Maiderrek:
  Jarraipen eta kontrola. Plangintzaren berrikuspena: atazak birdefinitu. Atazak (esperimentuak) sortzeko sistemaren garapena egin. HDF5 formatuaren inguruan informazioa bilatu. Cloud zerbitzuen inguruan informazioa bilatu.
- Ezekielek:
  Android aplikazioaren garapenarekin jarraitu.
- Maiderrek:
  Datu-basea hobetu antolatu eta osatu. Kodea txukundu. API-a osatu eta zuzendu. Web aplikazioari itxura eman. Aplikaziorako proba kasuak definitu.

- Denok:
  Kongresurako artikulua prestatu.

C.7 Seigarren bilera

Data: 2016/05/02, 10:45

Helburuak

Proiektuaren egoeraren berrikuspena. Kongresura bidalitako artikuluaren egoeraren eguneraketa.

Egin beharrekoak

- Ezekielek eta Maiderrek:
  Proiektuaren garapenarekin jarraitu eta aurkitutako akatsak zuzendu.

C.8 Zazpigarren bilera

Data: 2016/05/30, 10:30

Helburuak

ICNR2016 kongresura bidalitako artikuluaren eguneraketa. Proiektua UCAmI [**?**] kongresuan aurkeztearen inguruan hitz egin. Web aplikazioaren interfazearen diseinua komentatu.

Egin beharrekoak

- Ezekielek eta Maiderrek:
  Sistemaren balidazioa egin ahal izateko bertsio egonkor bat landu.
- Maiderrek:
  Esperimentuak definitzeko interfazea landu.
- Nestorrek eta Edurnek: Esperimentua diseinatu eta prestatu.

C.9 Zortzigarren bilera

Data: 2016/06/13, 10:30

Bilerara hurbilduak: Borja Gamecho

Helburuak

Egin beharreko sistemaren balidazioa prestatu. Kongresuko artikulua osatu.

Egin beharrekoak

- Ezekielek eta Maiderrek:
  Kodean aurkitutako akatsak zuzendu. Aplikazioaren pantailazoak egin eta hauen deskribapena idatzi.
- Maiderrek:
  Balidazioa egiteko 10 ikertzaile eta 10 partehartzaile kontu sortu.

C.10 Bederatzigarren bilera (balidazioa)

Data: 2016/06/14, 9:30

Bilerara hurbilduak: Borja Gamecho

Helburuak

Sistemaren balidazioa egin bost partehartzaileekin. Kongresuko artikulua osatzen bukatu.

Egin beharrekoak

- Ezekielek eta Maiderrek:
  Partehartzaileek aipatutako komentario guztiak kontuan hartuta sisteman aldaketak egin. Aurkitutako akatsak eta arazoak konpondu. Artikuluaren azken berrikuspena.
- Maiderrek:
  Web aplikazioaren interfazea txukundu. Esperimentuen atala osatu: Denbora marka kopurua dinamikoa izatea. Datuen konfirmazioa. “Denbora” atala berregin. Denbora markei “ordena” gehitu. Hizkuntza aldaketa egitean aurkitu diren arazoak konpondu. *API*-a zuzendu. *HDF5* formatuko C fitxategia konpondu.

C.11 Hamargarren bilera

Data: 2016/07/18, 15:00

Helburuak

GAP-en memorien egoera eta aldaketak egiteko plangintza. Zuzendarien oniritziak: noizko eta zein baldintzetan. GAUR-en bidez egin beharreko pausuak (aurreko puntuarekin oso lotuta). Entsaioak defentsak egin aurretik. UCAmI2016 artikulua. Bestelakoak.

Egin beharrekoak

GAP-en laburpena eleanitza. Eranskinak ordenatu. GAUR-en sartu eta defentsa eskaera egin. UCAmI2016 artikuluaren berrikuspena eta zuzenketak.

C.12 Itxiera bilera

Data: 2016/08/29, 11:00

Helburuak

Memoriaren azken bertsioa komentatu. Artikuluaren egoera komentatu eta aldaketak adostu. Defentsa prestatzeko entseguen plangintza egin.

Egin beharrekoak

Memoriaren azken zuzenketak egin. Artikuluaren azken bertsioa memorian txertatu. Memoriaren azken bertsioa ADDI-ra igo. Defentsa prestatu.

Bibliografia

[1] *Android activity*-en bizitza zikloa. https://developer.android.com/training/basics/activity-lifecycle/starting.html.

[2] *10th International Conference on Ubiquitous Computing and Ambient Intelligence UCAmI 2016*. http://mami.uclm.es/ucami-2016/index.html.

[3] Reglamento de la *LOPD*. https://www.agpd.es/portalwebAGPD/canaldocumentacion/informes_juridicos/reglamento_lopd/index-ides-idphp.php.

[4] *UPV/EHU*-ko etika batzordea. http://www.ehu.eus/eu/web/ceid/home.

[5] Borja Gamecho doktorea. https://bgamecho.wordpress.com/.

[6] *e-Health* txartela. https://www.cooking-hacks.com/documentation/tutorials/ehealth-biometric-sensor-platform-arduino-raspberry-pi-medical.

[7] *BITalino* txartela. http://www.bitalino.com/.

[8] *BITalino* adibidea *Android*-entzako. https://github.com/BITalinoWorld/android-example.

[9] *Android Studio* garapen ingurunea. https://developer.android.com/studio/index.html.

[10] *Butter Knife* liburutegia. https://github.com/JakeWharton/butterknife.

[11] *Graph View* liburutegia. http://www.android-graphview.org/.

[12] H. Nyquist. Certain topics in telegraph transmission theory, 1928. Trans. AIEE, 47, 617-644.

[13] "C. E. Shannon. Communication in the presence of noise, 1949. Proc. Institute of Radio Engineers, 37, 10-21.

[14] J. Brooke. SUS: a retrospective. Journal of usability studies, 2013. 8(2), 29-40.

[15] Measuring usability with the system usability scale (sus). http://www.measuringu.com/sus.php.

[16] *International Conference on Neurorehabilitation*. http://www.icnr2016.org/.