

GRADO EN INGENIERÍA INFORMÁTICA DE GESTIÓN Y
SISTEMAS DE INFORMACIÓN
TRABAJO FIN DE GRADO

***ASMARI BURUZKO KONTSULTA
INTERAKTIBOAK ETA ERATORRIKO DATUEN
ANALISIA***

Alumno/Alumna: Ibargutxi Gerrickaetxebarria, Leire
Director/Directora: Armendariz Leunda, Anaje

Curso: 2018-2019

Fecha: Bilbao, 22,07,2019

Laburpena

Proiektu honen helburua 4-11 urte arteko umeen asma era didaktiko batean antzematea da. Horretarako, UnitedHealthCare-eko Asthma Action America erakundeko testa oinarritzat hartu da. Testaren galderei erantzunez puntuazio bat lortzen da, eta horren bidez ondoriozta daiteke medikuaren asistentzia beharrezkoa den ala ez. Aipatutako aburua erdietsi ahal izateko Telegram mezularitza aplikazioko bot bat garatu da.

Bot-ak erabiltzailearen datuak bildu, testaren galderak egin eta beharrezkoa bada gaixoak esleituta duen medikuarekin hitzordua zehaztuko du. Erabiltzaileak hitzordua aukeratu eta medikuari korreo bat helaraziko zaio; honekin batera Google Calendar-en hitzordua gehituz.

Era berean, botean lortutako datuak aztertuko dira, horretarako medikuari zuzendutako web orrialde bat garatu da. Web orrialde horretan, medikuaren agenda agertuko da, haren gaixoen datuekin eta lortutako datuen analisiekin batera.

Resumen

El objetivo de este proyecto es detectar el asma infantil de niños entre 4 y 11 años de forma didáctica. Para ello, se ha hecho uso del test de la organización Asthma Action America de UnitedHealthCare. Se obtiene una puntuación respondiendo a las preguntas de dicho test mediante el cual se puede conocer si la asistencia médica es necesaria en el paciente. Estas funcionalidades se han llevado a cabo mediante la creación de un bot de la aplicación de mensajería Telegram.

Entre las funciones que puede realizar con el bot, se pueden destacar las siguientes, recoge los datos del usuario, formula las preguntas del test y si es necesario, determina la cita con el médico asignado. El usuario selecciona una fecha para la cita y se enviará un correo al médico además de sumarle la cita en Google Calendar.

Asimismo, se analizarán los datos obtenidos en el bot, para ello se ha desarrollado una página web dirigida al médico, en la que se muestran la agenda médica y los datos de sus pacientes. Entre los datos de pacientes se encuentran, sus datos personales, su histórico de citas con el médico y algunos análisis obtenidos con los datos extraídos del bot.

Summary

The aim of this project is to detect childhood asthma in children between the ages of 4 and 11, in a didactic way. In order to calculate it, it has been used a test of UnitedHealthCare's Asthma Action America organization.

By answering the questions in that test, the user gets a score which can indicate if medical assistance is necessary in the patient. These functionalities have been carried out by creating a bot from the Telegram messaging application.

Among the functions that you can perform with the bot, you can highlight the following ones: the capability to collect the user's data, and also to ask the test questions and if necessary, determine the appointment with the assigned doctor. The user selects a date for the appointment and an email will be sent to the doctor, in addition, the appointment will be added in Google Calendar.

Furthermore, the data obtained in the bot will be analyzed, for this purpose a website has been developed to the doctor's use. The website shows the medical agenda and the data of his patients. Patient data includes your personal data, the history of doctor's appointments, and some analyses obtained from the data extracted from the bot.

Gaien aurkibidea

1. Sarrera	10
1.1 Proiektuaren jatorria.....	10
1.2 Proiektuaren aukeraketaren zergatia.....	11
1.3 Negozio- testuingurua	11
1.4 Terminoak eta haien esanahiak	11
2. Hasierako planteamendua	14
2.1 Helburuak.....	14
2.2 Proiektuaren garrantzia.....	15
2.3 Proiektuak eskainitako onurak	16
2.4 Arkitektura	16
2.5 Norainokoak	16
2.5.1 Lanaren deskonposizio egitura.....	16
2.5.2 Denboraren plangintza.....	24
2.6 Tresnak.....	28
2.7 Arriskuen kudeaketa	32
2.7 Ebaluazio ekonomikoa	34
3. Betekizunen bilketa	37
3.1 Telegram bot-a	37
3.2 Web aplikazioa	39
3.3 Domeinu eredua	40
3.3.1 Gaixoa	40
3.3.2 User	41
3.3.3 TestGalderak.....	41
3.3.4 Medikua.....	42
3.3.5 LanEgun	42
3.3.6 Hitzordua	42
4. Analisia eta diseinua	44
4.1 Hasierako diseinua	44
4.1.1 Telegrameko bot-a	44
4.1.2 Web aplikazioa	45
4.2 Bukaerako diseinua.....	45
4.2.1 Telegrameko bot-a	46
4.2.2 Web aplikazioa	46
4.3 Datu-Basea	46
5. Arkitektura	48
5.1 Telegram bot	48
4.2 Web aplikazioa	50
6. Garapena	52
6.1 Telegram bot-a	52

6.1.1 Garapen ingurunea.....	52
6.1.2 Datu-basearen konexioa	54
6.1.3 Korreoa bildatzea	54
6.1.4 Google Calendar-en erabilera	55
6.2 Web aplikazioa	59
6.2.1 Garapen ingurunea.....	59
7. Probak	64
7.1 1.Prototipoa:	64
7.2 2.Prototipoa:	65
7.3 3.Prototipoa:	65
7.4 4.Prototipoa:	66
7.5 5.Prototipoa:	66
7.6 6.Prototipoa:	67
7.7 7.Prototipoa:	67
7.8 8.Prototipoa:	68
7.9 9.Prototipoa:	68
7.10 10.Prototipoa:.....	68
8. Ondorioak	70
8.1 Betetako helburuak.....	70
8.2 Egindako plangintza vs. errealitatea	71
8.3 Arriskuak eta haien agerpenak	73
8.4 Etorkizunerako lana	73
8.5 Hausnarketa pertsonala	74
Erreferentziak.....	76
I. eranskina: Telegram-eko bot-aren erabiltzaile gida	78
II. eranskina: Web aplikazioaren erabiltzaile gida.....	82

Irudien zerrenda

Irudia 1: Bezero-zerbitzariaren arkitektura	16
Irudia 2: Proiektuaren LDE diagrama	17
Irudia 3: GANTT diagrama	27
Irudia 4: Asma testa	31
Irudia 5: Gaixoak bot-ean egin ditzakeen akzioak	38
Irudia 6: Medikuak web aplikazioan egin ditzakeen akzioak	39
Irudia 7: UML diagrama	40
Irudia 8: Proiektuaren lehen UML diagrama	45
Irudia 9: Proiektuaren datu-basearen diseinua	47
Irudia 10: Telegrameko bot-aren arkitektura	48
Irudia 11: Google-eko API-en arkitektura (Iturria: Adminitracion de API-Google cloud)	49
Irudia 12: Web aplikazioaren arkitektura	50
Irudia 13: KeyboardButton adibidea Irudia 14: InlineKeyboardButton adibidea	53
Irudia 15: AnswerCallbackQuery alertaren adibidea	53
Irudia 16: DB konexioaren kodea	54
Irudia 17: Korreoa bidaltzeko kodea	55
Irudia 18: Jasotako korreoa	55
Irudia 19: OAuth 2.0 fluxua Iturria: Alejandro Diaz Torres, DRUPALERA)	56
Irudia 20: OAuth 2.0 URL-a Irudia 21: OAuth 2.0 baimentzea	57
Irudia 22: Kopiatu beharreko baimen kodea	57
Irudia 23: Google Calendar-ri deia egiteko kodea	58
Irudia 24: Leiho modal baten adibidea	60
Irudia 25: Taulen eta sekzioen adibidea	60
Irudia 26: Nabigatzeko burukoaren adibidea	60
Irudia 27: Gaixoa aukeratzeko aukera menua	61
Irudia 28: Grafiken adibidea	61
Irudia 29: egutegiaren adibidea	62
Irudia 30: Egutegiaren leiho modalaren adibidea (Abuztuak 14 eguneko hitzorduak)	62
Irudia 31: Erabiltzailearen informazioa bot-ean	78
Irudia 32: Testaren galderen eredu	79
Irudia 33: Testaren puntuazioa	79
Irudia 34: Medikuari hitzordua eskatzea	80
Irudia 35: Google Calendar-eko baimenak ematea eta autorizazio kodea	80
Irudia 36: Kakotzeko pantaila	82
Irudia 37: Gaixoaren aukeraketa	82
Irudia 38: Gaixoaren informazioa	83
Irudia 39: Hitzorduen egutegia	83
Irudia 40: Hitzorduen informazioa	84
Irudia 41: Estatistiken pantaila	84
Irudia 42: Gaixoaren grafikoaren adibidea	84

Taulen zerrenda

Taula 1: Telegram bot-eko prototipoen taula	21
Taula 2: Web aplikazioko prototipoen taula.....	22
Taula 3: Telegram bot-eko prototipoen proben taula.....	23
Taula 4: Web aplikazioko prototipoen proben taula.....	24
Taula 5: Atazen denbora estimazioen taula	25
Taula 6: Gastuen estimazioen taula.....	36
Taula 7: Denbora planifikazio errealaren taula.....	72

1. Sarrera

Atal honetan proiektuaren ikuspen global bat erakusten da, honen jatorrizko funtsa abiapuntutzat hartuz.

Proiektu hau, 4-11 urte tarteko umeen asma arazoaren kontrolera bideratuta dago. Gaixotasunaren jarraipen interaktibo bat lortzeko Telegrameko bot bat eta web aplikazio bat garatu dira. Horrela, gaitzaren kontrol zehatzagoa eta kontsultak egiteko unean erosotasuna lortzen da.

Alde batetik, bot-a gaixoen erabilerarako bideratuta dago. Telegram mezularitza aplikazioaren ChatBot-a erabili da, Bot-a programa informatiko bat da elkarrizketa bat edukitzeko aukera ematen duena, modu berean akzio bat eskatu ahal zaio, zein informazioa. Gaur egun, bot-en abantaila da, beste aplikazioak ez bezala, ez dela eguneratu behar eta gainera ez duela terminaleko memoriarik xahutzen. @OsasunBot izenpean, gaixoaren asma graduari neurri emateko erabili da.

Beste aldetik, web aplikazioa medikuei haien kontsultara esleituta dituzten asma pairatzen duten erien segimendua egiteko zenbait erraztasun ematen dizkie, hala nola, datozen hitzorduen kontsultak, erien informazioa eta estatistikak.

1.1 Proiektuaren jatorria

Gratu Amarierako Lan (GRAL) honen funtsa Kudeaketaren eta Informazio Sistemen Informatikaren Ingeniaritzako Graduak ANA JESUS ARMENDARIZ LEUNDA irakasleak proposatu zidan, haren ikerketa talderako medikuek inplementazioa behar zutelarik.

Gaur egun, mundu mailan asma ondoez kroniko ohikoenetariko bat da, 334 milioi pertsona erituz. Populazio-hazkundearekin jarraitzekotan, 2025erako asmarekiko gaixoen kopurua 100 milioitan handituko litzateke, haurtzaroaren gaixotasun kroniko nagusia bihurtuz eta osasun-kostuen kausa nagusietako bat ekarriz. (Martínez-Moragón E et al. Arch Bronconeumol 2009; 45:481-486).

Hain arrunta denez, haren diagnostikoa ez da oso erreza antzematen eta Espainian %50 paziente diagnostikatu gabe geratzen dira. @OsasunBot-ak eskaintzan duen galdetegiarekin auzi hori erraztu daiteke, testa betez jakin daiteke umeen asmaren adierazgarriak al dagoen eta medikuaren iritzia beharrezkoa baden ala ez.

1.2 Proiektuaren aukeraketaren zergatia

Proiektua egiterakoan, argi neukan nik neuk interesgarria topatzeko, auzi bat konpontzearekin lotuta egon behar zuela, eta era berean gizarteari hobekuntza bat eskaini behar ziola. Halaber, medikuntza arloa oso interesgarria aurkitu dut beti eta GRAL-a egitearekin batera zenbait ezagutza eskuratuko nituen arlo horretan.

Telegram aplikazioaren erabilera ondo ezagutzen nuen, askotan erabili nuen ikasketetatik kanpo, gehien bat txateatzeko. Baina denbora-pasa alde batera utziz, tresna baliagarri moduan erabiltzea oso erakargarria iruditu zitzaidan, ez nekien Telegram-ek funtzionalitate gehiago zituenik. Gainera, erronkatzat ikusi nuen, ez nuelako horrelakorik inoiz inplementatu.

Hasieratik banekien gutxi izan ere datuen analisiaren inguruan zerbait gehitu nahi niola nire proiektuari. Horretarako datuak behar nituen, kasu honetan Telegram-eko asma inkesta egitean haien asmaren puntuazioak. Datu horiek interpretatzeko eta horietatik ezagutzak lortzeko grafikak era oso egokia iruditzen zitzaidan, eta hori gehitu nahi nion.

1.3 Negozio- testuingurua

Proiektu hau medikuntzaren arloan kokatzen da, gaixoari banakako analisia egitea baimentzen dio eta analisiaren emaitza gaixoak medikuaren asistentzia duela bada, automatikoki gaixoak esleituta duen medikuari abisua ematen dio korreo bidez. Medikuek web aplikaziotik jarraipena egin ahal dio gaixoari.

Beste arlo batzuetan ere aplikatu ahal litzateke, proiektuaren zenbait atal aldatuz noski. Baina funts nagusia test bat eginez abisu bat ematea edo test-a periodikoki betetzen duen erabiltzailearen jarraipena egitea izango litzateke.

Kasu honetan, proiektua 4-11 urte bitarteko umeentzat zuzenduta dago, asma antzemateko edo asmaren jarraipena era eroso batean egiteko haurretan. Medikuari abisua emateko eta medikuek haren gaixoen informazioa jasotzeko datu-basean dauden datuez baliatzen da, kasu honetan Euskal herriko osasun zentroko bezeroei zuzenduta egongo litzateke.

1.4 Terminoak eta haien esanahiak

Atal honetan memorian zehar azaltzen diren termino eta siglen definizioak ageri dira.

- API: ingelesez Application Programming Interface, erregela (kode) multzo bat aplikazioak jarrai ahal dituztenak haien artean komunikatzeko.
- Chatbot/bot: erabiltzaile batek elkarrizketa bat izan dezan baimentzen duen programa informatikoa. Kasu honetan, Telegram aplikazioko bot-a.
- CSS: Cascading Style Sheets, dokumentuak erabiltzaileei aurkezteko era diseinatzeko erabiltzen den lengoia; estiloa, diseinua, etab. zehazten ditu.
- DB: Datu-basea, informazio antolatuko bilduma bat da.
- EJS: Handlebars eta beste tanto batzuk bezala, bezeroari eskaini baino lehen zerbitzarian html orriak (zeinen luzapena .ejs da) prozesatzea baimentzen du, horrela hauek zati dinamikoak izan ditzaten baimentzen du.
- GRAL: Gradu Amaierako Lana. Unibertsitateko gradu bat bukatzean derrigorrez egin beharreko proiektua.
- HTML: HyperText Markup Language, web orrialden baten edukiaren ordena zehazten duen lengoia.
- JSON: JavaScript Object Notation, datu-trukerako testu errazeko formatua.
- MYSQL: Datu-base erlazionaleko kudeaketa sistema da.
- UML: Unified Modeling Language, sistema bat ikustarazteko, zehazteko, eraikitzeke eta dokumentatzeko hizkuntza grafikoa da.
- URL: Uniform Resource Locator , formatu estandarra da sareko baliabideak izendatzen dituen

2. Hasierako planteamendua

Atal honen helburua proiektuaren norainokoak eta irismena azaltzea da, orokorki GRAL-a egin aurretik kontuan hartu beharrezko xehetasunak azaltzen dira. Horrez gain proiektuaren arriskuak, denboraren planifikazioa, analisi ekonomikoa eta erabilitako tresna nagusiak ere nabarmentzen dira.

2.1 Helburuak

Proiektu honen lehen mailako helburua 4-11 arteko umeen asma diagnostikatzearen prozesua azkartzea eta ondoez horren jarraipena erraztea da, horretarako sistema erabilerraza garatu behar da, haren erabilera sustatzearen. Gehiago zehaztearren, helburuak honako hauek dira medikuarentzako eta gaixoarentzako:

Gaixoa

- Umearen asmaren larrialdi maila ezagutzea, era autonomoan. Asmaren seinale larririk balego, osasun zentruko zerbitzuetara jo beharreko prozesua azkartzea.
- Datu-Babes-Legea betetzea.
- Era autonomoan, umearen asmaren eboluzioa kontrolatzea.

Medikua

- Haren erriaren asmaren eboluzioa grafikoki ikustea, tratamendua hobetzearen.
- Hitzorduak kontrolpean izatea.
- Haren erien informazio eta historikoa aztertzeke era erosoan.

Hala ere, badaude ere zenbait bigarren mailako helburu aplikazioak garatzean kontuan hartu behar direnak, jarraian aipatzen direnak:

Web aplikazioa

Medikuarentzako web aplikazioa eraikitzea bete nahi diren xedeak azaltzen dira:

- Erabilerraza eta sinplea izatea. Informazio atalak ondo bereizita egotea.
- Segurua izatea, gaixoen datuak agertzen baitira. Horretarako, mediku bakoitzari erabiltzaile bat eta pasahitz bat esleituko zaie.

Telegram bot-a

Gaixoarentzako Telegram-eko bot-a eraikitzean bete nahi diren xedeak azaltzen dira:

- Erabilerraza eta sinplea izatea. Uneoro zer gertatzen ari den berri izateaz gainera testaren erantzunak aukeratzeko modu bakuna agertu behar da.
- Medikuen eta gaixoaren arteko komunikazioa automatizatzea eta azkartzea, kontsulta hitzartzea alegia.
- Erabiltzaileen hitzorduaren informazioa apuntatzearen prozesua hartzea automatizatzea.

2.2 Proiektuaren garrantzia

Bi puntu nabarmen daude proiektu honen garrantzia gailentzen dituztenak.

Alde batetik, osasun zentroarekin komunikazioa automatizatzea. Gaur egun, Osakidetzaren kontsulta metodologia ez dago batere gaurkoratua, ez da teknologiaren erabilerarik egiten, hura gehituko bailitzaio osasun zerbitzua eraginkorragoa izango litzateke. Proiektu honen hobekuntzarekin, erabiltzaileak era autonomoan konproba dezake umean asma adierazgarrikeri ageri al den ala ez, ambulatiora joan Gabe. Aldi berean, prebentziorako ere erabil dezake, nahiz eta umeak asma seinalerik ez izan, erosotasunean bete dezake testa, ondoeza ez daukanaren asmoa egiaztatzeko. Prozesu hori, gaur egun bakarrik ambulatioaren idazkaritzara deitzen eta kontsulta bat eskatzen lor daiteke. Horrek zerbitzuen denbora xahutzea suposatzen du, proiektu honekin zuzenean umearen segurantza zenbakia emanez haren medikua eskuratu eta hitzordua eskuratuko luke.

Bestalde, asmaren kontrol zehatza lortzea oso zaila da gaur egun, medikuak diagnostikoak pilatu beharko lituzke horretarako, eta analizatzerako orduan banaka konparatu. Proiektu honekin, umeak periodikoki testa beteko balu, medikuari haren eboluzioa era erraz batean ikustea baimenduko lioke.

Azkenik, kontuan izan beharreko auzi garrantzitsu bat, erabilitako datuen konfidentzialtasuna da. Biltegitutako datuak osasun zentroetako bezeroen datuak izanda Datu Pertsonalak Babesteko Lege Organikoa aintzat hartu behar da, horregatik web aplikaziora konektatzen diren medikuak bakarrik osasun zentroak esleitutako erabiltzaile eta pasahitz-arekin sartu ahalko dira, ezinezkoa izango da erregistratzea edota pasahitza aldatzea

2.3 Proiektuak eskaintako onurak

Gaur egun asma antzemateko bakarkako erak eta asmaren jarraipenik egiteko tresnak ez daude osasun zentroetan erabileran, horregatik osasun zerbitzuetarako erakargarria den proiektu bat sortu nahi da.

Kasu honetan proiektuak abantailak eskaintzen dizkie medikuari zein gaixoari, 2.2 atalean (Proiektuaren garrantzia) adierazi den bezala gaixoari autonomia ematen dio asma analizatzeko baita erosotasuna kontsultaren hitzordua eskatzeko unean. Bestalde, medikuari informazioa eskaintzen dio, diagnostikoen eboluzioa zein gaixoen informazio pertsonala.

2.4 Arkitektura

Proiektu honetan zehazten den arkitektura bezero-zerbitzariarena da. Bi rolek hartzen dute parte prozesuaren ahalmenean. Honako hau da haren komunikazio prozedura:



Datuak prestatu	Eskaerak itxaron
Zerbitzua eskatu	Eskaerak itxaron
Erantzuna itxaron	Eskaera prozesatu
Erantzuna itxaron	Eraitza itzuli
<i>Erantzuna prozesatu</i>	

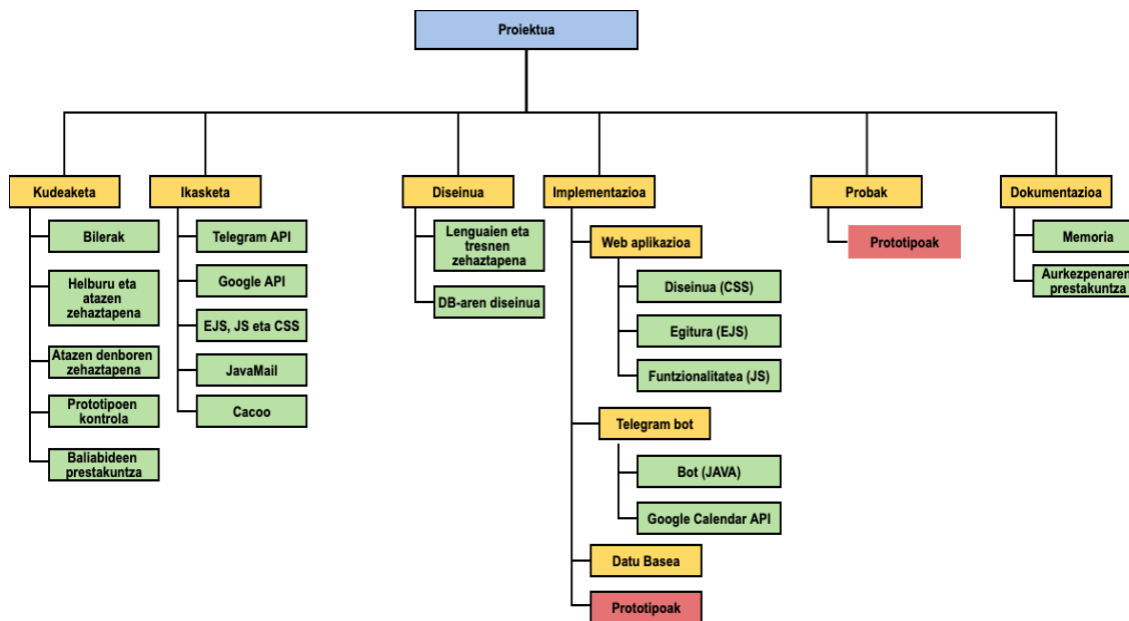
Irdia 1: Bezero-zerbitzariaren arkitektura

Bezeroak Telegram bot-a erabiltzen duen gaixoak edo web aplikazioa erabiliko duten medikuak dira eta zerbitzuak eskatuko dizkiete zerbitzariari.

2.5 Norainokoak

2.5.1 Lanaren deskonposizio egitura

Atal honetan proiektua aurrera eramateko bete beharreko funtzionalitateak azaltzen dira. Jarraian proiektuaren atazak zehzaten dituen LDE diagrama ageri da.



Irudia 2: Proiektuaren LDE diagrama

LDE diagramak bereizten dituen atal nagusiak honako hauek dira:

- **Kudeaketa:** Proiektuaren kudeaketaren atazak ditu baitan. Proiektuaren hasierako fasean izango du garrantzirik handiena, plangintza finkatu eta estimatu behar da eta baina proiektu osoan zehar luzatuko da, horretan atazak bete diren eta haien analisiak egingo dira.
- **Ikasketa:** Proiektuan erabilitako ezagutzak jasotzeko prozesuetaz arduratzen da.
- **Diseinua:** Proiektuaren funtzionaltasunak zehazten dira atal honetan.
- **Implementazioa:** Proiektuak izango dituen funtzionaltasunak implementatuko dira. Bitan banandu da ulermena erraztearren, horiz garatu diren funtzionalitateak orokorrean eta gorritz garatuko diren prototipoak garatu behar diren ordenean banan-banan azaltzen dira.
- **Probak:** Funtzionaltasunak era egokian garatu diren aztertzeaz arduratzen da. Azken prototipoa neurrikoa izan dadin aurreko prototipoen azterketak egin behar dira.
- **Dokumentazioa:** Proiektuaren garapena azaltzen da, haren ulermen bidezkoa eskuratzeko, garapenarekin paraleloki egingo da.

Jarraian, zehazki atal bakoitzaren azpi osagaiak azalduko dira, haien deskripzioa, ordu estimazioa eta erabiliko diren baliabideak zehaztuz.

2.5.1.1 Kudeaketa

Atal honetan

- **Bilerak:**
Proiektuaren arduradunarekin egindako batzarrak proiektuaren atazak zehazteko, bideratzeko eta zalantzak argitzeko.
Iraupena: 6 ordu
- **Helburu eta atazen zehaztapena:**
Proiektuaren garapenean jasango diren helburuak eta atazak finkatzea, hala nola, proiektua ataletan banatzea eta betebeharrak espezifikatzea.
Baliabideak: eskemak eta dokumentazioa
Iraupena: 3 ordu
- **Atazen denboraren zehaztapena:**
Proiektuaren atalen iraupen estimazio bat finkatu eta norberari atazak bukatzeko epeak ezarri, proiektua ez atzeratzeko eramateko.
Baliabideak: egutegia, timetable moduko bat zehazten da.
Iraupena: 2 ordu
- **Prototipoen kontrola:**
Bot-aren kodearen aldaerak kudeatzea.
Baliabideak: Github
Iraupena: 2 ordu
- **Baliabideen prestakuntza:**
Inplementatzeko beharrezkoak diren softwareen instalazioa eta prestakuntza.
Baliabideak: Eclipse, MySQL, Nodejs, Atom, Github, phpMyAdmin...
Iraupena: 10 ordu

2.5.1.2 Ikasketa

Atal honetan, edozer inplementatu orduan behar diren ezagutzak edukitzeko egin diren ikasketak aipatzen dira.

- **Telegram API:**
Telegram-eko APIaren dokumentazioa aztertu, esate baterako Telegram elkarrizketako objektuak, konexioa...

Iraupena: 15 ordu

- Google API:

Google OAuth eta Google Calendar API-ari buruzko informazioa jaso JAVA lengoian.

Iraupena: 10 ordu

- EJS, JS eta CSS:

Web Aplikazio Aberastuen Garapena irakasgaiari landu genituen web aplikazioen inplementazioak, baina apunteak errepasatu behar dira eta gainera funtzionalitate desberdinak erabili behar izango dira proiektu honetan, irakasgaiari sakondu ez genituenak.

Iraupena: 8 ordu

- JavaMail:

Eclipse-tik Java lengoian korreoak automatikoki bidaltzeko API honen informazioa bilatu

Iraupena: 4 ordu

- Cacao eta Toms Planner:

Dokumentazioa eta lanaren plangintza egiteko eskemak egiteko tresna honen funtzionalitate desberdinak ikasi.

Iraupena: 3 ordu

2.5.1.3 Diseinua

Atal honetan, proiektuan egitura bat eman behar zaien atazak diseinatu dira, gero inplementatu egingo direnak.

- Lengoia eta tresnen zehaztapena:

Telegram bot-a eta web aplikazioa zer programazio lengoian eta zer tresna erabiliz garatuko diren finkatzea, horretarako interneten erabilera eta adibide desberdinak bilatuko dira, ondo alderatu ahal izateko.

Baliabideak: Tresnei dagokienez, eclipse, nodejs, atom editorea, github.. Lengoiaei dagokienez Bot-a garatzeko JAVA eta web aplikazioa garatzeko EJS, JS eta CSS.

Iraupena: 5 ordu

- Datu-basearen diseinua:

Datuak biltegitzeko datu-basearen diseinuaren garatzea.

Iraupena: 5 ordu

2.5.1.4 Implementazioa

Atal honetan, funtzionalitate guztiak kodifikatu dira.

Lehen azaldu bezala, implementazioan bi atal bereizi dira, ulermena errazteko. Garatuko diren funtzionalitateak, proiektuaren iraupenaren ikuspegi orokor bat lortzeko, eta bestalde egingo diren prototipoak.

- Telegram bot-a:
 - Bot: Java lengoaian Telegram bot-aren implementazio osoa egingo da , Telegram APIko liburutegia erabiliz.
 - Google Calendar: Java lengoaian Google oAuth liburutegiaren bidez baimena eskatu eta Google Calendarren hitzordu bat gehitzearen implementazioa garatzea.

Baliabideak: Eclipse, Github

Iraupena: 120 ordu

- Web aplikazioa:
 - Diseinua: CSS lengoaian garatuko da web aplikazioaren diseinua, hura apaindu egiten du, koloreak, formak... aukeratzea baimenduz.
 - Egitura: EJS lengoaian garatuko da web aplikazioaren egitura, hipertestu edo "tag"-ak erabiliz web aplikazioaren informazioaren ordena finkatzen da.
 - Funtzionalitatea: JavaScript lengoaian garatuko da web aplikazioaren funtzionalitatea, script-en bidez akzioak aplikatu ahal zaizkio.

Baliabideak: Atom editorea, Github, nodeJS

Iraupena: 80 ordu

- Datu-basea:

Datu-basearen taulak eta erlazioak garatzea.

Baliabideak: Mysql, phpMyadmin

Iraupena: 20 ordu

- Prototipoak

Bot-aren prototipoak	Funtzionalitatea	Baliabideak	Iraupena
1. prototipoa	Komando komunei erantzuna eman(/start, /osasun, /info), test-a erakutsi.	Eclipse, Telegram liburutegia	40 ordu
2. prototipoa	Test-aren erantzuna lortu, datu-basearekin konektatu eta test-eko datuak biltegiratu.	Eclipse,Telegram liburutegia, Mysql, phpmyAdmin	25 ordu
3. prototipoa	Hitzordua aukeratzeko aukera eman, datu-basetik medikuaren ordu libreak eskuratzuz.	Eclipse,Mysql, , Telegram liburutegia , phpmyAdmin	20 ordu
4. prototipoa	Erabiltzaileak esleitutako medikua datu-basetik eskuratu eta korreoa bidali.	Eclipse, Telegram liburutegia , JavaMail liburutegia, Mysql, phpmyAdmin	15 ordu
5. prototipoa	Erabiltzaileak aukeratutako hitzordua Google calendarren gehitu.	Eclipse, Telegram liburutegia ,Google API-ak, Mysql, phpmyAdmin	20 ordu

Taula 1: Telegram bot-eko prototipoen taula

Telegram-eko bot-a eginda eduki eta gero, han eskuratutako datuekin medikuarentzako web aplikazioa garatu daiteke:

Web aplikazioa-ren prototipoa	Funtzionalitatea	Baliabideak	Iraupena
6. prototipoa	Login-a garatu.	Atom, nodeJs, GoogleChrome nabigatzailea	15 ordu
7. prototipoa	Datu-basearekin konektatu	Atom, nodeJs, GoogleChrome nabigatzailea, mysql, phpmyadmin	5 ordu
8. prototipoa	Medikuaren hitzorduekin egutegia erakutsi.	Atom, nodeJs, GoogleChrome nabigatzailea, mysql, phpmyadmin	20 ordu
9. prototipoa	Medikuak esleituta dituen asma gaixoen informazio pertsonala eta haren hitzorduen historikoa erakutsi.	Atom, nodeJs, GoogleChrome nabigatzailea, mysql, phpmyadmin	20 ordu
10. prototipoa	Medikuak esleituta dituen gaixoen asmaren eboluzioaren grafikak, eta bot-a erabili duten gaixoen grafika orokorrak erakutsi	Atom, nodeJs, GoogleChrome nabigatzailea, mysql, phpmyadmin	20 ordu

Taula 2: Web aplikazioko prototipoen taula

2.5.1.4 Probak

Atal honetan prototipoen funtzionalitate egokia baieztatzeko egin diren azterketak erakusten dira. Probak bakarka egiten badira ere, proiektua aurrera joan ahala aurreko prototipoen kaliatea ere probatuko da, horrela web aplikazio eta bot-aren egokitasuna azpimarratuz.

Bot-aren prototipoak	Funtzionalitatea	Baliabideak	Iraupena
1. proba	1 prototipoaren probak egingo dira, komandoei erantzun egokia agertzen baden eta test-a era egokian erakusten baden egiaztatuz.	Telegram, Eclipse	7 ordu
2. proba	2 prototipoaren probak egingo dira, datu-basean datuak sartzen diren konprobatuz.	Telegram, Eclipse, phpmyadmin, phpmyadmin	3 ordu
3. proba	3 prototipoaren probak egingo dira, hitzordua aukeratzea posible al den egiaztatuz, eta datu-basean begiratzuz ea zuzena den medikuak libre duela ordua.	Telegram, Eclipse, phpmyadmin	5 ordu
4. proba	4 prototipoaren probak egingo dira, korreo bat bidali ahal den egiaztatuz.	Gmail, Telegram, Eclipse	3 ordu
5. proba	5 prototipoaren probak egingo dira, Google caldendarrera hitzordua gehitu al den egiaztatuz.	Telegram, Google Calendar, Eclipse	5 ordu

Taula 3: Telegram bot-eko prototipoen proben taula

Web aplikazioa-ren prototipoa	Funtzionalitatea	Baliabideak	Iraupena
6. proba	1 prototipoaren probak egingo dira, login-a ondo erakusten den egiaztatuko da.	nodeJs, GoogleChrome nabigatzailea	2 ordu

7. proba	2 prototipoaren probak egingo dira.	nodeJs, GoogleChrome nabigatzailea	1 ordu
8. proba	3 prototipoaren probak egingo dira, egutegia ondo erakusten dela eta datu-baseko datu egokiak diren egiaztatuz.	nodeJs, GoogleChrome nabigatzailea, phpmyadmin	4 ordu
9. proba	4 prototipoaren probak egingo dira, gaixoaren datuak ondo erakusten direla eta datu-baseko datu egokiak diren egiaztatuz.	nodeJs, GoogleChrome nabigatzailea, phpmyadmin	2 ordu
10. proba	4 prototipoaren probak egingo dira, grafikak ondo erakusten direla eta bat datozela datuekin egiaztatuz.	nodeJs, GoogleChrome nabigatzailea, mysql, phpmyadmin	5 ordu

Taula 4: Web aplikazioko prototipoen proben taula

2.5.1.5 Dokumentazioa

Atal honetan, proiektua egin bitartean eta bukaeran biltegitratutako dokumentazioa azaltzen da.

- Memoria:
Proiektuaren inplementazioaren memoria egin.
Iraupena: 80 ordu
- Aurkezpenaren prestakuntza:
Aurkezpenean erabiliko diren tresnak eta ezagutzak prestatu
Iraupena: 20 ordu

2.5.2 Denboraren plangintza

Aurrean zehaztu diren atazen denbora estimazioak aintzat hartuz, hurrengo taulan proiektuaren iraupenaren estimazio orokorra erakusten da:

Ataza	Denbora estimazioa
KUDEAKETA	23
Bilerak	6
Helburu eta atazen zehaztapena	3
Atazen denboraren zehaztapena	2
Prototipoen kontrola	2
Baliabideen prestakuntza	10
IKASKETA	40
Telegram API	15
Google API	10
EJS,JS eta CSS	8
JavaMail	4
Cacoo eta Toms Planner	3
DISEINUA	10
Lengoaia eta tresnen zehaztapena	5
Datu-basearen diseinua	5
INPLEMENTAZIOA	200
Telegrameko bot	120
Web aplikazioa	80
PROBAK	37
Telegrameko bot-a	23
Web aplikazioa	14
DOKUMENTAZIOA	100
Memoria	80
Aurkezpena	20
GUZTIRA	410

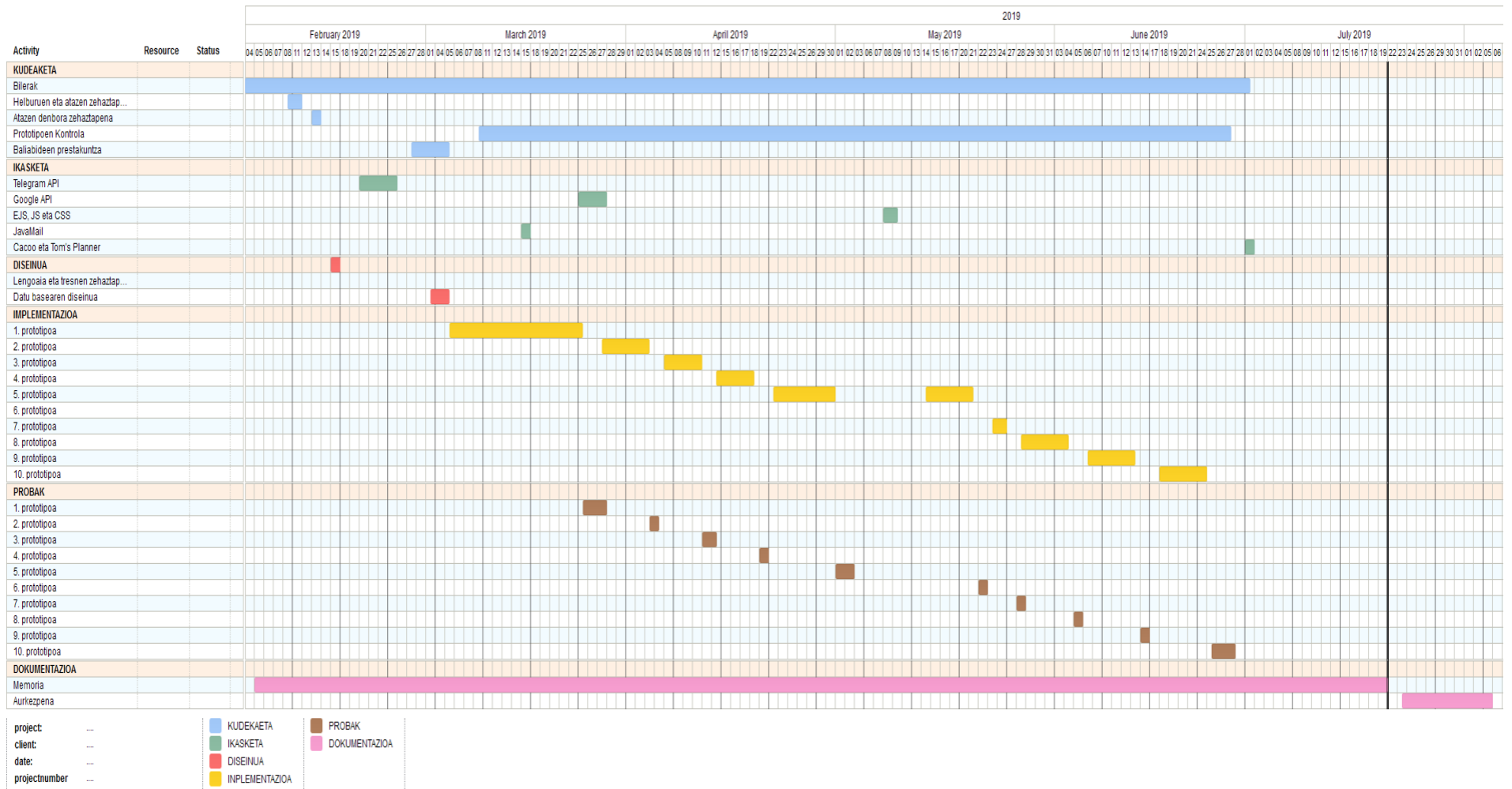
Taula 5: Atazen denbora estimazioen taula

Guztira proiektuaren iraupena 410koa izango dela estimatzen da. Gainera, ondorioztat har dezakegu atalik sakonenak inplementazioa eta dokumentazioa direla, eta berriz, denbora gutxien eskatzen dituztenak proiektuaren hasieran gauzatzen direnak dira, inplementazioa egin aurretik zehaztu behar diren faktoreak, kudeaketa eta diseinua hain zuzen ere.

Proiektuaren hazkundera era erregular batean igaro dadin, egin beharreko atazak kontrolatu behar dira, bakoitzaren sorreran eta garapenean sartuko den denboraren estimazio zehatz bat eginez, eta denboran zehaztuz. Hori Gantt diagramen bidez burutuko da, otsailean hasi zen proiektuaren garapena eta irailean bukatu da haren aurkezpenarekin.

Denbora banaketa egitean asteburuak eta jaiegunak ez dira kanpo utzi eta ez da ordutegi finko bat esleitu.

Zenbait ataza simultaneoki egin ahal direnez, haien artean teilakatzen dira, adibidez bilerak eta dokumentazioa proiektu ia osoan zehar luzatzen diren atazak dira, 3. Irudian ikusi ahal den bezala.



Irudia 3: GANTT diagrama

2.6 Tresnak

Atal honetan proiektuan zehar erabiliko diren baliabideak aipatzen dira.

- Eclipse
 - Deskripzioa: Plataforma anitzeko kode irekiko software plataforma da, garapen ingurune integratuak egiteko erabiltzen da. Javako aplikazioak edo C++ -eko programak inplementatzen dira bertan.
 - Erabilera: Proiektu honetan telegram bot-aren kodea inplementatu da bertan JAVA lengoian, 2019-03 bertsioa erabiliko da horretarako, Eclipsek 2019ko Martxoaren 20an atera zuen.

- Atom editorea
 - Deskripzioa: Kode irekiko editorea da, Node.js-n idatzitako zenbait plug-in eta Githubek garatutako Git bidezko bertsio kontrolak onartzen ditu.
 - Erabilera: Web aplikazioaren kodea garatzeko erabiliko da.

- NodeJS
 - Deskripzioa: Kode irekiko plataforma anitzeko ingurune bat da, Interneteko aplikazio eskalagarriak sortzeko diseinatua izan dena, zerbitzariak gehien bat. Javascript lengoian kodifikatzen da, gertakizunetan euskarrituta dagoen arkitektura da. Denbora azkartzea eta eskalagarritasuna maximizatzeko sarrera-irteera unitate asinkronoa erabiltzen du.
 - Erabilera: Web aplikazioaren kodea garatzeko erabiliko da.

- Telegram
 - Deskripzioa: Software libreko bat-bateko mezularitza aplikazioa da, Android, Max OSX eta IOS sistema eragileetako gailuetan dago eskuragai.
 - Erabilera: Telegrameko bot-a frogatzeko erabiliko da.

- Google Chrome kotsola
 - Deskripzioa: Google Chrome nabigatzaileak web aplikazioen garapenean laguntzeko baitan duen javascript kotsola, araztegia ere barne dauka.
 - Erabilera: Web aplikazioaren erroreak antzemateko, eta frogak egiteko.

- Google Cloud Plataform

- Deskripzioa: Google-en azpiegitura berean aplikazioak, web inguruneak eta zerbitzuak sortzea eta inplementatzea baimentzen du.
- Erabilera: Google Calendarreko API-a erabili ahal izateko.

- PHPmyAdmin:
 - Deskripzioa: PHP-n idatzitako tresna da, web-orrietaz baliatuz MySQL-ren administrazioa kudeatzeko sortua.
 - Erabilera: datu-basea era errazean kudeatu ahal izateko erabiliko da.

- Github:
 - Deskripzioa: Software lankidetzaren garapen plataforma bat da, zeinek proiektuak biltegitratzea baimentzen duen, Git bertsio kontrol sistema erabiliz.
 - Erabilera: Kodearen aldaketan jarraipena izateko eta kodea “hodeian” gordeta izateko erabiliko da.

- Stackoverflow:
 - Deskripzioa: Informatikoen komunitateak erabilitako web orria, edozein programazio lengoian sortzen diren arazoei konponbidea ateratzeko diseinatua.
 - Erabilera: Inplementazioan zehar suertatu diren arazoen konponbidea bilatzeko erabiliko da.

- Cacao
 - Deskripzioa: Online grafikak eta taulak egiteko web orrialdea.
 - Erabilera: Memoriako zenbait grafika egiteko erabiliko da.

- Tom’s planner
 - Deskripzioa: Online planifikazio ereduak egiteko web orrialdea.
 - Erabilera: Gantt diagrama egiteko erabiliko da.

- Microsoft Office
 - Deskripzioa: Microsoftek sortutako bulegotika-suitea da, testu prozesagailuak, kalkulu-orriak, aurkezpenak... egiteko aplikazioak eskaintzen ditu.

- Erabilera: Memoria eta aurkezpena egiteko erabiliko dira Word (testu-a prozesatzeko aplikazioa) eta Powerpoint (aurkezpena egiteko aplikazioa) aplikazioak. Test-aren galdera eta erantzuna .csv batean gordeko dira, hori kudeatzeko Excel aplikazioa erabiliko da.

- Gmail
 - Deskripzioa: Google-en doako posta elektronikoko zerbitzua da.
 - Erabilera: medikuari hitzorduari buruzko informazioa duen korreoa heldu zaion konprobatzeko erabiliko da.

- Test-a

Telegram bot-ak UnitedHealthCare-eko Asthma Action Americako honako test hau erabiliko da, umearen asma puntuazioa lortzeko:

Prueba para el Control del Asma Infantil para niños de 4 a 11 años.

Cómo hacer la Prueba de Control del Asma Infantil

- ▶ Paso 1 Deje que su hijo responda las primeras cuatro preguntas (1 a 4). Si su hijo necesita ayuda para leer o comprender la pregunta, usted puede ayudar, pero deje que su hijo elija la respuesta. Responda las tres preguntas restantes (5 a 7) usted solo sin dejar que la respuesta de su hijo influya en sus respuestas. No hay respuestas correctas o incorrectas.
- ▶ Paso 2 Escriba el número de cada respuesta en la casilla de puntaje provista.
- ▶ Paso 3 Sume las casillas de puntaje para obtener el total.
- ▶ Paso 4 Lleve la prueba al médico para hablar sobre el puntaje total de su hijo.

19
o menos

Si el puntaje de su hijo es de 19 o menos, puede ser un indicio de que el asma de su hijo no está tan bien controlada como podría estarlo. Sin importar cuál sea el puntaje, lleve la prueba al médico para hablar sobre los resultados de su hijo.

Pídale a su hijo que responda estas preguntas.

1. ¿Cómo está tu asma hoy?

 Muy mal	 Mal	 Bien	 Muy bien	PUNTAJE <input type="text"/>
--	--	---	---	---------------------------------

2. ¿Qué tipo de problema te causa el asma cuando corres, haces ejercicio o practicas deportes?

 Un gran problema, no puedo hacer lo que quiero hacer.	 Es un problema y no me gusta.	 Un problema pequeño pero está bien.	 No es un problema.	<input type="text"/>
--	--	--	---	----------------------

3. ¿Toses por el asma?

 Sí, todo el tiempo.	 Sí, la mayor parte del tiempo.	 Sí, un poco.	 No, nada.	<input type="text"/>
--	---	---	--	----------------------

4. ¿Te despiertas durante la noche por el asma?







 Sí, todo el tiempo.	 Sí, la mayor parte del tiempo.	 Sí, un poco.	 No, nunca.	<input type="text"/>
--	---	---	---	----------------------

Responda las siguientes preguntas usted solo.

5. Durante las últimas 4 semanas, ¿cuántos días su hijo tuvo síntomas de asma durante el día?

 Ninguno	 1 a 3 días	 4 a 10 días	 11 a 18 días	 19 a 24 días	 Todos los días	<input type="text"/>
--	---	--	---	---	---	----------------------

6. Durante las últimas 4 semanas, ¿cuántos días su hijo tuvo sibilancias durante el día debido al asma?

 Ninguno	 1 a 3 días	 4 a 10 días	 11 a 18 días	 19 a 24 días	 Todos los días	<input type="text"/>
--	---	--	---	---	---	----------------------

7. Durante las últimas 4 semanas, ¿cuántos días su hijo se despertó durante la noche debido al asma?

 Ninguno	 1 a 3 días	 4 a 10 días	 11 a 18 días	 19 a 24 días	 Todos los días	<input type="text"/>
--	---	--	---	---	---	----------------------

TOTAL



Irudia 4: Asma testa

2.7 Arriskuen kudeaketa

Proiektuan zehar suertatu ahal diren arazo kopurua gutxitzeko, suertatu daitezkeen arriskuak ezagutzea garrantzizkoa da. Arriskuen prebentzio plan zorrotza batek proiektuarentzako onura suposatuko du.

Arriskuen larrialdi maila kontrolatu ahal izateko bi faktore hartuko ditugu kontuan:

- Gauzatzeko probabilitatea: Arrisku hori proiektuan zehar suertatzeko probabilitatea
 - Baxua: %0- %33
 - Ertaina: %34-%66
 - Altua: %67-%100

- Proiektuan daukan inpaktua: Arrisku horrek proiektuaren estimatutako planifikazioan eragiten duen aldaketa.
 - Txikia: %0- %33
 - Ertaina: %34-%66
 - Handia: %67-%100

Hurrengo lerroetan proiektuan eragina izan ahal duten arriskuak aipatzen dira, prebentziorako planak eta kontingentzia planarekin bat:

- Programatzailean arazo pertsonalak edo gaixotasuna:

Programatzailean arazo pertsonal edo gaixotasunengatik lan egiteari uztea.

 - Prebentzio plana: Planifikazioa egiterako unean, egun libreak utzi.
 - Kontingentzia plana: Birplanifikatu, lan karga libre dauden egunetan ezarri.
 - Probabilitatea: Txikia
 - Inpaktua: Handia

- Hardware arriskuak:

Erabilitako hardwarearen egoera txarra, adibidez ordenagailuarena.

 - Prebentzio plana: Informazio eta kode osoa sarean gorde.
 - Kontingentzia plana: Informazioa berreskuratzen saiatu, hardwarea erreparatu.

- Probabilitatea: Txikia
- Inpaktua: Handia

- Software birusak:
Ordenagailuaren sistema eragileak akatsak jasotzea, informazioa galtzea gehien bat.
 - Prebentzio plana: Softwarea eguneratuta mantendu, “antivirus”-a instalatu, arriskutsuak diren web orrietan ez sartu ...
 - Kontingentzia plana: Informatikako espezialista batengana eraman.
 - Probabilitatea: Txikia
 - Inpaktua: Handia

- Denboraren planifikazio desegokia:
 - Prebentzio plana: Planifikazioa egiteko orduan kontuan hartu faktore guztiak, adibidez programatzailearen egun libreak, ataza bakoitzaren garrantzia eta zailtasuna...
 - Kontingentzia plana: Birplanifikatu edo proiekturen zenbait atal kendu.
 - Probabilitatea: Ertaina
 - Inpaktua: Ertaina

- Erabilitako tresnerian aldaketa:
Erabiliko diren APIetan bertsio aldaketak gertatzea, eta aurreko bertsioak bertan behera uztea.
 - Prebentzio plana: Kodifikatzean azken egunerapenerako informazioa erabili.
 - Kontingentzia plana: Askotan ez dute aurretiaz erabilitakoan eraginik bakarrik etorkizunera programatuko dituztenetan kentzen dituzte, kasu horretan ez da ezer aldatu behar, beste kasu batzuetan dena eguneratuko beharko litzateke.
 - Probabilitatea: Txikia
 - Inpaktua: Ertaina

- Funtzionalitateak diseinatzeko arazoak:
 - Prebentzio plana: Behar den denbora eman atazak zehazten.
 - Kontingentzia plana: Planifikazioa berregin, ezinezkoa bada ataza batzuk kanpoan utzi (proiektuan eragin oso handia ez dutenak).
 - Probabilitatea: Ertaina

- Inpaktua: Handia

2.7 Ebaluazio ekonomikoa

Atal honetan, proiektua osatzen duten baliabide eta zerbitzuen kostua estimatzen da. Unibertsitateko proiektua denez argi dago ez dagoela etekin ekonomikorik.

Ebaluazioa egiteko, kostua dakarten faktoreka zehaztu eta haien koste ekonomikoa kalkulatu da:

- Eskulana

Eskulan atalean enpresak pertsona bakoitzarengatik faturatzen duen diruaren kalkulua egingo da.

Proiektu hau garatzeko programatzaile eta analista lanak burutu dira, abiapuntutzat hartuko dugu 25 euroko kostua suposatzen duela orduro.

Proiektu honetan estimatutako lan orduak 2.5.2 atalean (Denbora Plangintza) ikus daitekeenez 410 ordukoak dira. Orduan lan orduak eta orduko soldata biderkatuz proiektu honen eskulanaren kostua lortuko dugu.

$$Eskulana = Proiektuko lan orduak * Orduko soldata = 410 * 25 = 10,250 \text{ €}$$

- Software

Softwareari dagokionez ez da gasturik egongo. Proiektuan zehar erabiliko diren software guztiak doakoak dira. Kostua ekar lezakeen tresna bakarrak Google Cloud Plataform Microsoft Office izango lirateke, baina horretarako unibertsitatekoagatik izateko kreditua lortu da, epe mugatu baterako.

- Hardware

Proiektua garatzeko ordenagailu eramangarri bat erabili da. Zehazki Apple Macbook Pro 13 hazpetatakoa, i5 2,3 GHz prozesagailuarekin eta 256GB-ko memoriarekin. Horren prezioa FNAC enpresan une honetan 1,900 eurokoa da. Beste aldetik hardwarearen kostuaren estimazio egoki bat egiteko ordenagailuaren bizi estimazioa behar da, 6 urtekoa da kasu honetan. Bi datu horiek zatituz hilabeteroko amortizazioa eskuratuko dugu.

$$\text{Hileroko amortizazioa} = \frac{\text{Prezioa}}{\text{Bizi iraupena}} = \frac{1900}{12 \cdot 6} = 26,39 \text{ €}$$

Lan egindako hilabeteak proiektu honen 2.5.2 atalean (Denbora Plangintza) aurkitu ahal ditugu, 6 hilabete izango dira. Aurkezpena prestatzeko behar diren hilabeteak, abuztua eta iraila dira, baina prestakuntza horrek hardwarearen erabilera konstantea suposatzen ez duenez, hau da hilabete bi horietan lan karga ez denez handia izango, hilabete bat moduan kontatuko da. Hori horrela izanga, lan egindako hilabeteak 7 izango dira. Orduan hardwarearen amortizazio osoa honako hau izango da:

$$\text{Amortizazio osoa} = \text{Hileroko amortizazioa} \cdot \text{lan egindako hilabeteak} = 26,39 \cdot 7 = 184,73 \text{ €}$$

Guztira, hardwarearen amortizazio osoa 184,73 euro dira.

Aipatu behar da, probatzeko Telegram ordenagailuko aplikazioa baita Iphoneko aplikazioa erabili direla, hau da, iphone 8 plus mugikorra erabili da zenbait frogak egiteko. Dena den Telegram ordenagailuko aplikazioa nahikoa denez frogak egiteko, ez da kostuan sartu mugikorraren erabilera.

- Zeharkako gastuak

Proiektuan zuzeneko inpaktua ez duten faktoreak osatzen dute atal hau baina garapenean funtsezkoak direnak.

Zeharkako gastutzat internet konexioa eta argindarra har daitezke. Kostu aldakorrek dituztenez tarifa zehatz bat ezarriko da hilabetero, argindarragatik 10 euro eta internetgatik 15 euro. Orduan zeharkako gastuak hilabetero 25 eurokoak izango dira.

Zeharkako gastu osoak kalkulatzeko, proiektuan lan egindako hilabeteak behar dira, aurreko atalean aurki daiteke datu hori: 6 hilabete.

$$\text{Zeharkako gastuak} = \text{Hilabetezko zeharkako gastuak} \cdot \text{lan hilabeteak} = 25 \cdot 6 = 150 \text{ €}$$

Guztira, zeharkako gastuak 150 eurokoak dira.

Jarraian, kostu guztiak banaka kalkulaturik izanda, proiektuaren kostu orokorra kalkulaturik da, hurrengo taulan ikus daitekeenez:

Gastua	Kostua €-tan
Eskulana	10.250
Hardwarea	184,73
Softwarea	0
Zeharkako gastuak	150
Guztira	10.584,73

Taula 6: Gastuen estimazioen taula

Ondoriotzat atera daiteke, proiektuaren kostuaren estimazioa 10.584,73 eurokoa dela.

3. Betekizunen bilketa

Atal honetan Telegram bot-ak eta web aplikazioak bete behar dituzten funtzionalitateak azaltzen dira.

3.1 Telegram bot-a

Atal honetan Telegram-eko bot-ak bete beharreko zereginak zerrendatuko dira.

Komando soilenei erantzuna:

Erantzuna /start /osasun eta /info komandoei. Bot-a martxan jartzeko /start komandoa sakatu beharko da, /osasun komandoa asma test-a hasteko eta /info bot-aren informazioa jasotzeko.

Test-a erakutsi:

Test-a egitea erakargarria izan behar da, hau da, galderei erantzuna ematea modu erraz batean isladatuta egon behar da.

Segurantza zenbakia eskatu:

Gaixoaren segurantza zenbakia eskatu behar da, datu-basearen bidez haren informazioa eskuragarri izateko, adibidez esleituta duen medikua. Segurantza zenbakia datu-basean ez badago ezin izango da testa egin.

Test-aren emaitzak gorde:

Gaixoarekin erlazionatu eta hark erantzundako testen erantzunak gorde, gero analisiak egin ahal izateko, eta bere asmaren eboluzioa aztertu ahal izateko.

Testa berriz egitearen posibilitatea:

Bot-ak segurantza zenbakiaren bidez antzeman behar du ea test-a aurretiaz bete duen ala ez, nahiz eta lehendabiziko aldia ez izan aukera berdinak izango ditu, hau da test-a bete ahal du eta bere medikuarekin hitzordua ere eska lezake.

Hitzordua eskatu:

Test-a bukatu eta gero haren medikuak libre dituen hitzorduetatik bat aukeratu ahal izango du erabiltzaileak, korreo bat bidaliko zaio medikuari.

Test-aren emaitza asma ez daukala bada ere, hitzordua eskatzeko posibilitatea edukiko du.

Hitzordua gorde:

Erabiltzaileak aukeratutako hitzordua Google Calendar-ean gorde ahal izango du, horretarako baimena eman beharko du haren Google-eko korreoa eta pasahitzaz kautotuz.

Bi hizkuntza erabilgarri:

Osakidetzara bideratuta dagoen bot-a denez, dena gaztelania eta euskaraz egon behar da. Telegram-ek bidaltzen dituen edozein mezu eta test-a bi hizkuntzetan egon behar dira.

Zehaztuta ez dauden mezuen kontrola:

Bot-ak bakarrik zenbait hitz baimenduko ditu, zenbait erantzunentzako mugatuta dago, test-aren erantzunak, segurantza zenbakia, komandoak... Lekuz kanpo dauden erantzuna bada, erabiltzaileak mezu bat jasoko du hori adieraziz.

Hemen ageri dira (irudia 5) erabiltzaileak egin ahal dituen akzioak, aurrerago azalduko da zein ordenean egin ahal dituen, hau da akzio batzuetara ezin izango da heldu beste akzio batzuk aurretik egin gabe. Adibidez /osasun komandoak hasiera ematen dio bot-ari, ez bada jartzen ezin da edozein beste akziorik egin.



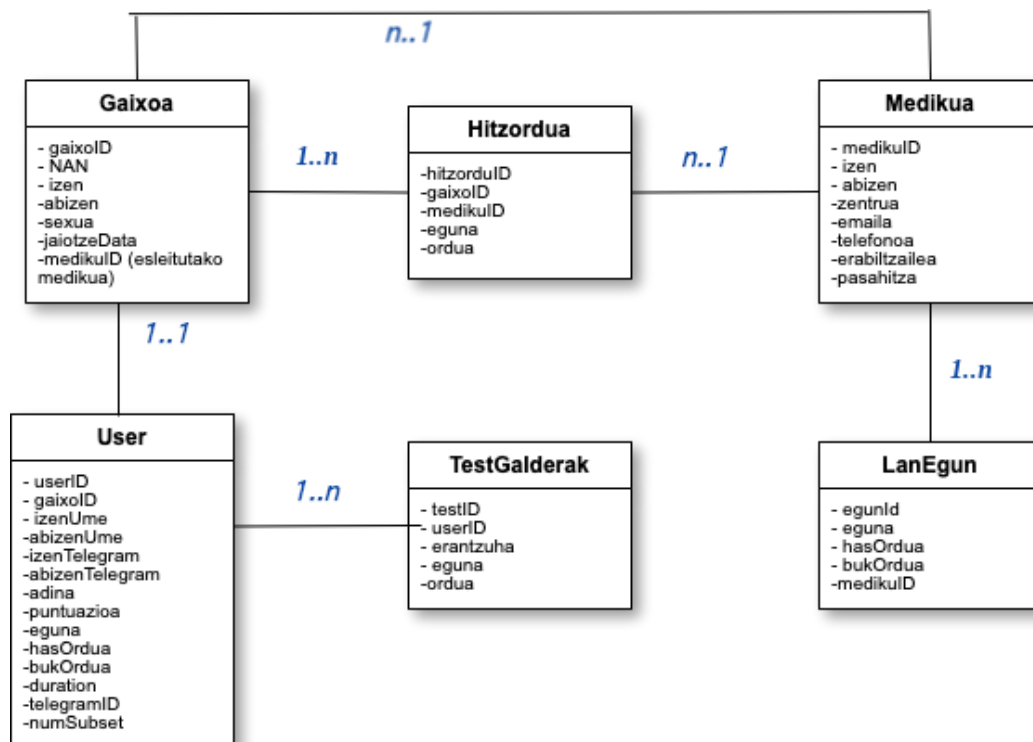
Irudia 5: Gaixoak bot-ean egin ditzakeen akzioak

Bot-aren kasuaren moduan, akzio hauek orden bat jarraitzen dute, kautotu gabe ezin izango da beste akziorik burutu.

3.3 Domeinu eredua

Atal honetan, garatutako domeinu eredua agertzen da, medikuaren eta gaixoaren arteko erlazioekin, eta Telegrameko bot-aren erabiltzaileak gaixoarekin daukan erlazioarekin.

Domeinu eredian proiektuan zehar biltegitatu nahi diren datuak gordetzen dira, haien artean erlazionatuta, ulermena erraztearren. Gainera aurrean azaldutako funtzionalitateak behar dituzten baliabideak finkatzen dira, adibidez “Hitzordua eskatu” atalean aintzat jo dezakegu mediku eta gaixoaren artean Hitzordua deritzon erlazio bat sortu behar dela.



Irudia 7: UML diagrama

3.3.1 Gaixoa

Gaixoa identifikatzeko haren segurantza zenbakia erabiltzen da (gaixoID), gizarte segurantzako zenbakia mugatuta dago, 48 (Bizkaia), 01 (Araba) edo 20 (Gipuzkoa) zenbaki sekuentzietatik hasi beharko da eta 9 zenbakiz osatuta egon beharko da. Beste

aldetik erriaren datu pertsonalak ere gordetzen dira, izen-abizena, NAN zenbakia, sexua, jaiotze data eta osasun zentroan esleituta duen medikuaren id-a batik bat.

3.3.2 User

Gaixoak Telegram-eko erabiltzaile moduan testa betetzen duenean gordetzen dira datuak taula honetan. Honako osagai hauek ditu:

- `userId`: era inkrementalean esleitzen den id-a.
- `gaixoid`: gaixoaren Segurantzaren zenbakia, bot-ak erabiltzaileari eskatuko dio, horrela gaixoaren taularekin erlazionatu daiteke eta haren datuak eskura izan.
- `IzenUme` eta `abizenUme`: Bot-ak erabiltzaileari eskatuko dizkion umearen izen eta abizenak.
- `izenTelegram` eta `abizenTelegram`: erabiltzaileak Telegram aplikazioan zehaztuta dituen izen eta abizenak.
- `adina`: Bot-ak erabiltzaileari eskatuko dio umearen adin muga, test-a bakarrik 4-11 urte tarteko umeentzat baita.
- `puntuazioa`: test-a betez lortutako asma puntuazioa.
- `eguna`: bot-a erabilitako eguna.
- `hasOrdua`, `bukOrdua` eta `duration`: bot-a erabiltzen hasitako ordua, bot-a erabiltzari utzi zaion ordua eta horien tartean igaro den denbora.
- `TelegramID`: Telegram-ek erabiltzaile bakoitzari esleitzen dion ID-a.
- `numSubSet`: test-a `subSet` deritzon .csv artxiboen bidez kargatzen da, euskerazko 6 Subset eta gaztelarazko 6 Subset erregistratu dira. Artxibo batek huts egiten badu bot-a bertan behera ez uzteko, programak ausaz 1-6rako bat hartzen du.

3.3.3 TestGalderak

Test-aren galdera bakoitzari buruzko informazioa gordetzen da. Test-aren galdera bakoitzagatik horrelako taula bat biltegitratuko da datu-basean. `userId` bidez erlazionatzen da User taularekin, `testId`-a galdera bakoitzak esleituta daukan id-a da. Galderaren id-ak ez dira inkrementalki zehaztu, .csv artxiboetan zehaztuta daude. Azkenik galderari eman dion erantzuna eta zer datan eman dion (egun eta ordua) gordeko dira.

3.3.4 Medikua

Medikua identifikatzeko medikuID-a bereizi da, MIR zenbakia da, hain zuzen ere. Bestalde medikuaren datu pertsonalak gordetzen dira, izen abizenak, lan egiten duen zentroa, korreoa eta telefono zenbakia. Gaixoak hitzordua zehazten duenean korreo horri helaraziko zaio hitzordu horren informazioa. Horregatik haren zentroko idazkaritzak bere hitzorduak kudeatzea nahi badu, medikuak korreetzat idazkaritzako korreoa eman beharko luke.

Azkenik web aplikazioan sartu ahal izateko erabiltzaile eta pasahitza biltegitratzen dira.

3.3.5 LanEgun

Medikuak lan egiten dituen egunak dira. EgunID era inkrementalean esleitzen den id-a da, eta horrek identifikatuko ditu egunLibre-ak, bakana baita. Medikuarekin erlazionatzeko, hau da jakiteko lanaldi hori mediku horri egokitzen zaiola, MedikulID-a gordeko da.

Azkenik, ze egunetan eta zer ordutik (HasOrdua) zer ordura (BukOrdua) den bere lanaldia gordeko da.

3.3.6 Hitzordua

Medikua eta gaixoaren arteko egoteko deia gordeko da Hitzordua taulan. Bereizi ahal izateko hitzorduaID gakoa dauka, era inkrementalean esleitutako id-a da. Medikua taularekin medikuID bidez erlazionatzen da, eta gaixorenarekin gaixoID bidez. Hitzorduaren eguna eta ordua ere gordeko dira. Aipatzekoa da, hitzordu bakoitzari berez 20 minutuko luzapena esleitzen zaiola.

4. Analisia eta diseinua

Atal honetan proiektuaren funtzionalitateen analisi eta diseinua azalduko dira. Proiektua egiterako orduan egokitasunez eta zehaztasunez doitu dagoen diseinuak proiektuaren garapenean eragin nabarmena izan dezake.

Proiektu bat hasteko unean, funtsezkoena garatuko den plataforma zehaztu eta hari buruzko beharrezko ezagutzak eskuratzea da. Jarraian proiektuaren eboluzioa erakusten da, hasierako faseetik, inplementatzen hasi aurreko ideiatik, gaur egun den proiekturako bidea bereiziz.

Diseinuak bi ataletan bereiziko dira, proiektua bi plataformek osatzen baitute, Telegrameko bot-a alde batetik eta bestetik web aplikazioa.

4.1 Hasierako diseinua

Atal honetan hasieran zehaztu ziren funtzionalitateak azaltzen dira, ulergarriagoa izateko inplementatutako bi ataletan banandu da.

4.1.1 Telegrameko bot-a

Proiektua diseinatzekoan hasierako asmoa, Telegrameko bot-ak hiru funtzio nagusi izatea zen:

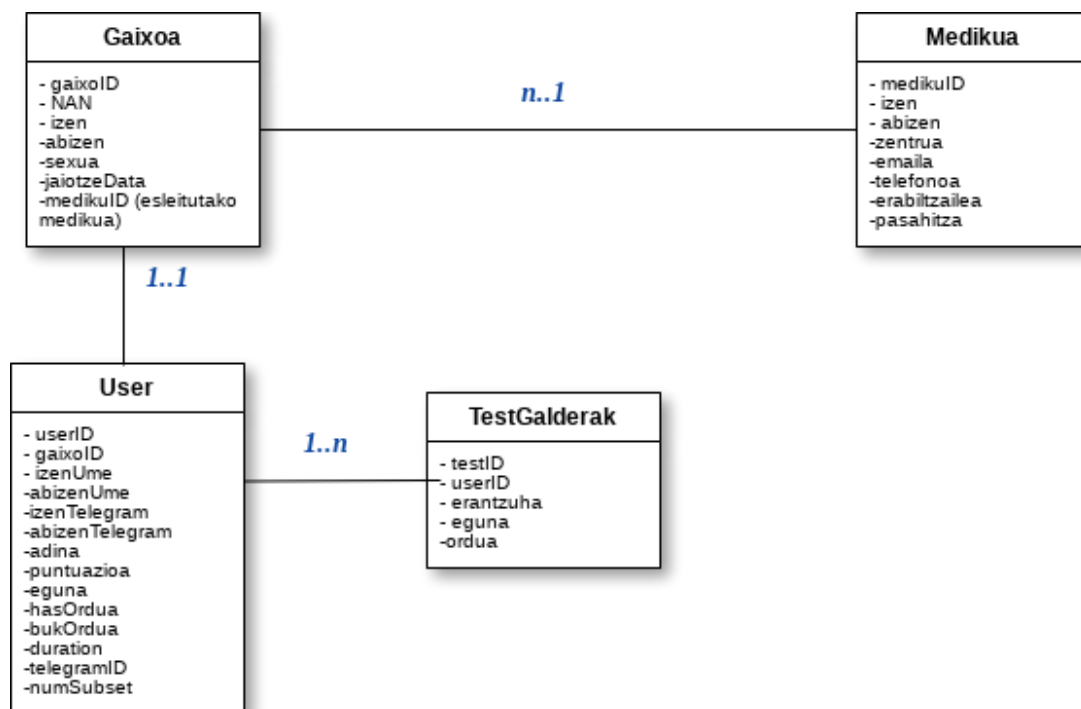
- Test-a erakutsi ahal izatea, era soil baten, hau da, galdera bakoitzari erantzuna teklatu pertsonalizatu baten bidez ematea ahalbidetzea. Bestela esanda, erabiltzaileak klik batean aukera dezake egokien ikusten duen aukera. Era horretan, erabiltzaileak ez du ezer idatzi behar, eta bot-a mugatuago, seguruago eta eraginkorragoa izango da.
- Erabiltzailea 4-11 urteko umea dela egiaztatzea eta hura identifikatzeko beharrezkoa den informazioa eskatzea, segurantzaren zenbakia eta izen-abizenak batik bat.
- Test-aren emaitzak medikuari korreo bidez helaraztea. Test-a betetzen denean, emaitza edozein izanda ere, hots, positiboa edo negatiboa izanda ere, medikuari bidaltzea.

4.1.2 Web aplikazioa

Datuen analisiarekin zerikusia zeukan atal bat garatzearren sortu zen web aplikazioaren ideia. Web aplikazio honetan lau puntu zehaztu ziren:

- Mediku moduan edo gaixo moduan sartu nahi den aukera ematea. Gaixo moduan bada bot-era birhelbideratuko du erabiltzailea, berriz medikuaren aukerak login-era bidaliko du.
- Medikuen login-a: Datu konfidentzialak direnez, eta osasun zentroko bezeroen intimitatea eskubidea babestu behar denez, Datu Pertsonalak Babesteko Lege Organikoa betez. Beharrezkoa da datuak eskuratzeko osasun zentroko baimena izatea.
- Bot-ean lortutako datuak erabiliz, asmaren eboluzioa aztertzeko lagungarriak izan daitezkeen grafikak lortu nahi ziren.
- Esleitutako gaixoen informazio pertsonala erakustea.

Aipatzekoa da hasierako UML ereduak ez zituela Hitzordu ezta LanOrdu taulak, 3.1 Domeinu Eredua atalean dagoen UML azken ereduarekin alderatuz.



Irudia 8: Proiektuaren lehen UML diagrama

4.2 Bukaerako diseinua

Diseinuan aldaketa handiena medikuaren eta gaixoaren artean hitzorduak gehitzea izan da, eraldaketa horrek beste zenbait aldakuntza ekarri ditu.

4.2.1 Telegrameko bot-a

Hasierako diseinuarekin alderatuz zenbait funtzio gehitu eta aldatu ziren proiektuan zehar, jarraian azalduko direnak:

- 4.1.1 puntuan azaldu den moduan medikuari helaraziko zitzaion informazioa test-aren emaitzak izango ziren, amaierako diseinuan erabiltzaileak bot-ean aukeratutako hitzorduaren jakingarria helarazten zaio. Gaixo bakoitzaren testaren emaitza web aplikazioan ikusi ahal izango du.
- Aurreko puntuan antzeman denez, erabiltzaileak haren medikuak libre dituen lanorduetatik hitzordua eskatzeko aukera ematen zaio.
- Hitzordua Google Calendar-en gordetzeko aukera ere ematen zaio erabiltzaileari.

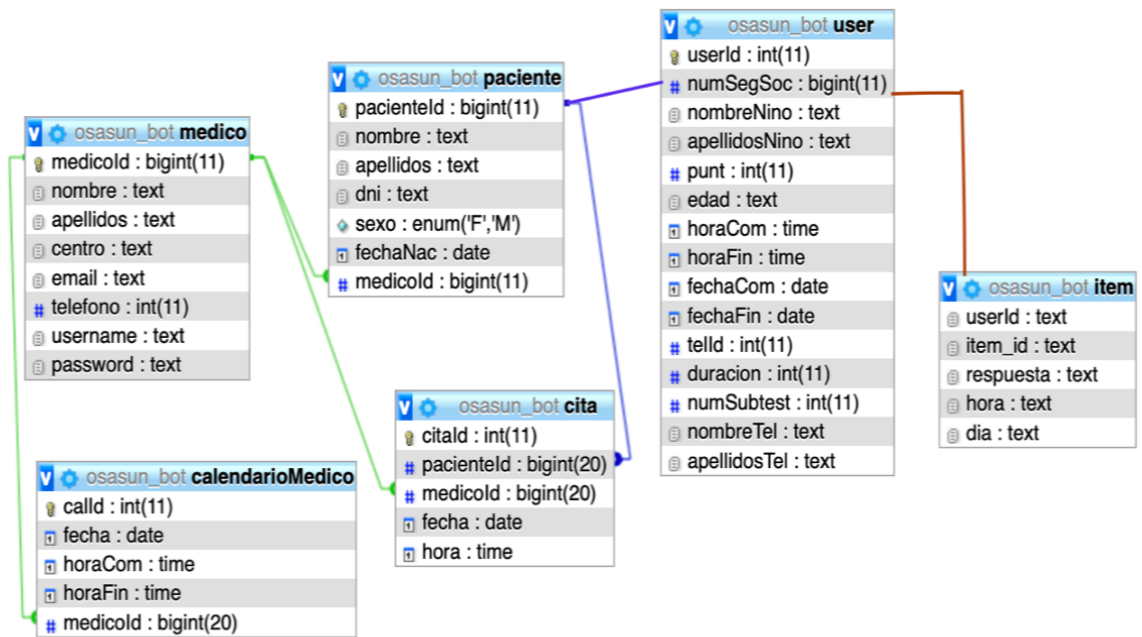
4.2.2 Web aplikazioa

Hasierako diseinuari bi funtzio gehiago atxikitu zaizkio:

- Gaixo bakoitzaren datu pertsonalekin batera izan dituen hitzorduen historikoa ere erakusten da.
- Egutegi baten bidez, medikuak haren hitzorduak konprobatu ahal ditu.

4.3 Datu-Basea

3.1 atalean erakusten den domeinu eredia aintzat hartuz, sortu da osasun_bot izenpean datu-basea. Taulen izenak eta atributuak erderaz daude publiko zabal batentzat ulergarriagoa izateko. Taula bakoitzaren informazioa 3.1 atalean azaldu da



Irudia 9: Proiektuaren datu-basearen diseinua

5. Arkitektura

Atal honetan Telegram-eko bot-ak eta web aplikazioak jarraitzen duten arkitektura erakusten da.

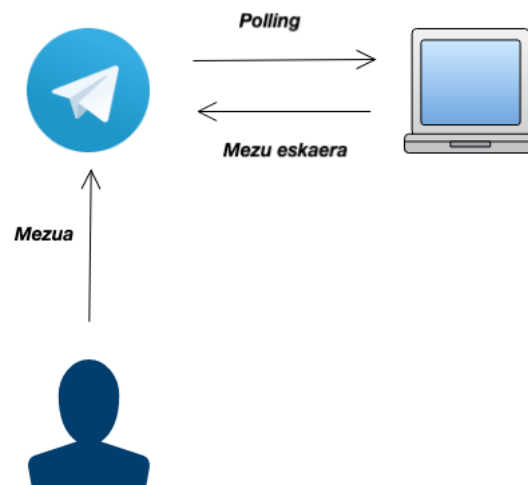
5.1 Telegram bot

Aurretiaz aipatu bezala, Telegram-eko bot-a java lengoian programatu da, Telegram-ekin konektatzeko *org.telegram* liburutegia erabili da.

Bot-ean mezuak jasotzeko bi teknika bereizten dira Polling eta Webhook.

- Polling metodoak Telegram zerbitzariari noizean behin galdetzen dio mezu berririk al dagoen (poll hitzak ingelesez bozketaren esanahia dauka), honek denbora muga eta dei asinkronoa zehaztea eskatzen du. Mezua Javako Telegram API-a erabiliz `onUpdateReceived(Update update)` metodora heltzen da.
- Webhook metodoan, Telegrameko zerbitzarian helbide bat erregistratzearen bidez komunikatzen dira programa eta Telegram zerbitzaria. Mezu berri bat heltzen denean zerbitzaria zehaztutako helbide horri eskaera bat egingo dio.

Kasu honetan polling metodoa erabili da, hurrengo irudian Telegram zerbitzari, programa eta erabiltzailearen arteko konexioak erakusten dira, 9. Irudian ikus daitekeen moduan.



Irudia 10: Telegrameko bot-aren arkitektura

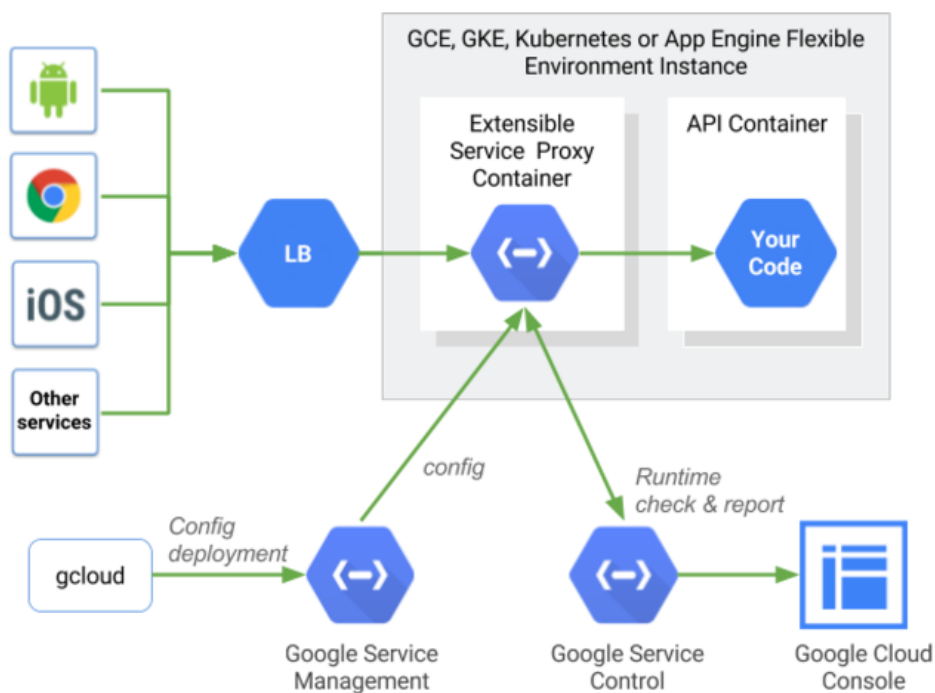
Lehen aipatu bezala jasotako mezuak, `onUpdateReceived(Update update)` metodoaren bitartez lortuko ditugu. Horrez gain, mezu bat bidali ahal izateko API-ak barne daukan `SendMessage` klasearen bidez bidali da.

Era berean, bot-ak korreo mezua bidaltzearen eta Google Calendar-en hitzorduak gehitzearen funtzioak ere burutu ahal ditu.

Alde batetik korreoa mezua bidaltzeko *javax.mail* eta *javax.activation* liburutegiak erabili dira, eta gainera mezua bidaltzeko korreo bat behar zenez, osasunbot@gmail.com korreo helbidea sortu da.

Beste aldetik, Google Calendar-en hitzorduak gehitzeko zenbait liburutegi erabili behar dira:

- Google-en baimena eskatzeko *com.google.oauth-client* eta *com.google.api-client* liburutegiak erabili dira.
- Google Calendar-en API-a erabili ahal izateko *google-api-services-calendar* liburutegia erabili da.



Irudia 11: Google-eko API-en arkitektura (Iturria: Adminitracion de API-Google cloud)

NGINXen oinarritutako proxy bat da ESP, backend aurrean exekutatzen dena eta Endpointsen funtzionalitateak barne hartzen dituena, hala nola autentifikazioa, ikuskapena eta erregistroa.

Azkenik, datu-basearekin konektatzeko *mysql-connector-java* liburutegia eta zenbait data formatu tratatzeko *DateTime* formatua adibidez, *joda-time* liburutegia erabili dira.

4.2 Web aplikazioa

Jarraian web aplikazioa egiteko jarraitu den arkitektura zehazten da.



Irudia 12: Web aplikazioaren arkitektura

Bezeroaren aldean exekutatzen diren teknologia guztiak frontend dira, hau da, web nabigatzailearen alde egiten duten teknologia guztiak, gure kasuan hiru lengoaia, ejs, CSS eta JavaScript .

Aurrekoa argituta, web aplikazioa garatzeko node JS zerbitzaria erabili da, horrek tartekari moduan jokatu du frontend eta datu-basearen artean. Barnean express liburutegia dauka, express liburutegia REST API bat da, momentu honetan teknologia honen erabilera lagungarria izan daiteke, modu erraz eta azkar batean datuak bidali eta jasotzea ahalbidetzen baitu.

REST API-ek HTTP metodoak aprobetxatzen dituzte, POST edo GET metodoetatik metodo pertsonalizatuetaraino, esate baterako, PUT edo DELETE.

Fluxua hurrengo hau izango da, frontend-ean erakusteko datuak behar direnean AJAX dei bat egingo zaio zerbitzariari, zerbitzariak MySQL bidez datu-basea kontsultatu eta informazioa bueltatuko du, JSON moduan frontend-ari itzuliko zaio, pantailan margotu dezan.

Beste aldetik beste zenbait liburutegi ere erabili dira, body-parser JSON-ak parseatzeko , chat.js grafikoak sortzeko, Jquery .ejs artxiboekin ibiltzeko, MySQL datu-basearekin konektatzeko, lehen esan den moduan eta zenbait beste, garrantzi handikoak ez direnak.

6. Garapena

Atal honetan proiektuaren funtzionalitateak garatzeko egindako garapen eta inplementazioa azalduko dira. Beste atal askotan bezala garapena bitan bananduko dugu, Telegram bot-aren garapena eta web aplikazioarena.

6.1 Telegram bot-a

Jarraian Telegram bot-aren inplementazioa azalduko da.

6.1.1 Garapen ingurunea

2.6 Tresnak atalean aipatu den moduan, Eclipse IDE for Java Developers plataforma erabili da Bot-a garatzeko 2019-03 (4.11.0) bertsioa eta JAVA lengoia programatu da.

Java lengoaiaren programazioaren paradigma objektuei zuzendutako programazioaren kontzeptuan oinarritzen da (OOP). Java lengoia C lengoaiaren deribatua da, adibidez kode blokeak metodoetan antolatzen dira eta giltzen bidez mugatzen dira (`{}` eta `}`).

Javako proiektu arrunta sortu ordez, Maven proiektua sortu da, Maven-en **Project Object Model (POM)** izeneko .xml motatako fitxategi batean eraiki beharreko software proiektua eta bere menpekotasunak definitzen dira.

Eclipseren Github plugin-a erabili da kodearen jarraipena egiteko eta kontrola egiteko.

Telegrameko zenbait osagai erabili dira bot-a osatzeko, jarraian azalduko direnak.

Mezu bat bidaltzean, hari erantzuna emango dion teklatua pertsonalizatu daiteke, Funtzio hau oso erabilgarria da nahi den erantzuna mugatu ahal delako, horrela bot-a seguruagoa izanik.

Teklatuaren botoiak bi motatakoak izan ahal dira:

- *KeyboardButton*: Teklatu arruntaren lekuan kokatzen dira zehatutako botoiak, bat sakatzean sakatutako mezua bidaliko da. Testaren erantzunetarako erabili da.
- *InlineKeyboardButton*: Teklatu arrunta ez da aldatzen, teklatu moduko bat agertzen da mezuen behealdean. Erabilera konplexuagoa dute, botoia sakatzean ez dute mezurik bidaltzen, mezua bidaltzea nahi bada callback-a erantzuna zehaztu beharko da. Gainera URL moduko botoiak ere sortu ahal dira. Hitzorduaren data eta ordua aukeratzeko erabili da, era berean Google Calendar-en baimena eskatzeko URL moduko botoia ere erabili da.

Hurrengo irudietan argiago geratuko da haien arteko diferentzia:

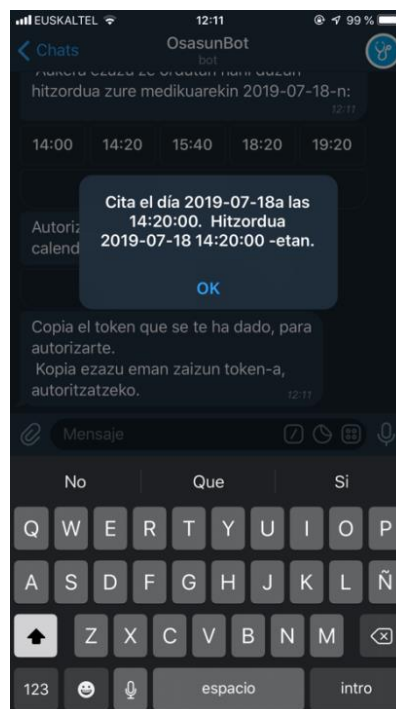


Irudia 13: KeyboardButton adibidea



Irudia 14: InlineKeyboardButton adibidea

Azkenik, alerta moduko bat agertzeko *AnswerCallbackQuery* osagaia erabili da. Osagai hau alerta moduan edo chat-aren goialdean notifikazio moduan erakuts daiteke.



Irudia 15: AnswerCallbackQuery alertaren adibidea

6.1.2 Datu-basearen konexioa

Bot-ak zenbait datu-base konexio beharrezkoak ditu, horregatik konexio eraginkor bat edukitzea funtsezkoa da. Mysql-connector-java driverra 5.1.44 bertsioduna erabili da datu-basearekin konexioa egiteko.

Jarraian azaltzen den kodearen bidez lortu da konexioa, probak egiteko datu-basea lokalean dago, horretatik localhost-era konektatzen da, erabiltzailea eta pasahitza ere pasatu behar zaizkio. Unibertsitateko zerbitzari batera igotzearen asmoa dago. Kasu horretan localhost-en ordeztu zerbitzariaren IP-a espezifikatu beharko genuke. ConectorMySQL klaseak datu-baserako kontsulta guztiak metodoetan berezita edukiko ditu eta konexioa eskatzen duen metodo bat edukiko du, jarraian azaltzen dena.

```

private ConectorMySQL() {
    try {
        try {
            Class.forName("com.mysql.jdbc.Driver");
        } catch (ClassNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/osasun_bot", "root", ' ');
    } catch (SQLException e) {
        Log.getLog().write("ERROR: Ha ocurrido un error al abrir la conexion con la base de datos.", true);
        e.printStackTrace();
    }
}

```

Irudia 16: DB konexioaren kodea

6.1.3 Korreoa bildatzea

Atal hau burutzeko JavaMail API-a erabili da, Gmaileko STMP zerbitzaria erabili da korreok bidali ahal izateko eta 587 portua. TLS protokoloaren bidez bidali dira segurtasuna bermatzeko. Garraio geruzaren segurtasuna (TLS) segurtasun-protokoloa da, pribatutasuna babesteko eta lehen konexio seguruko geruzak (SSL) betetzen zuen papera betetzen duena. Gmailek aurrez zehaztutako moduan erabiltzen du TLS, baina konexio segururik ez dagoenean (igorleak eta hartzaileak TLS erabili behar dute konexio segurua sortzeko), Gmail konexio ez-seguruen bidez entregatzen dituzte mezuak. STMP Gmail-eko arazo bat egunean 2000 mezuak bidaltzeko muga duela izan daiteke.

Guzti hau kudeatzeko Email klasea sortu da, eta metodo baten bidez egiten da guztia, jarraian azalduko dena, metodo horrek korreoa bidali beharreko datuak jasotzen ditu, Medikuen email-a eta izena, gaixoaren segurantzaren zenbakia eta izena eta haien arteko hitzorduaren datuak batik batik.

```

public class Email {
    public void sendEmail(String email, String nombreMed, String dia, String hora, String nombrePac, String numSegSoc) {
        final String username = "osasanbot@gmail.com";
        final String password = " ";
        Properties prop = new Properties();
        prop.put("mail.smtp.host", "smtp.gmail.com");
        prop.put("mail.smtp.port", "587");
        prop.put("mail.smtp.auth", "true");
        prop.put("mail.smtp.starttls.enable", "true"); //TLS

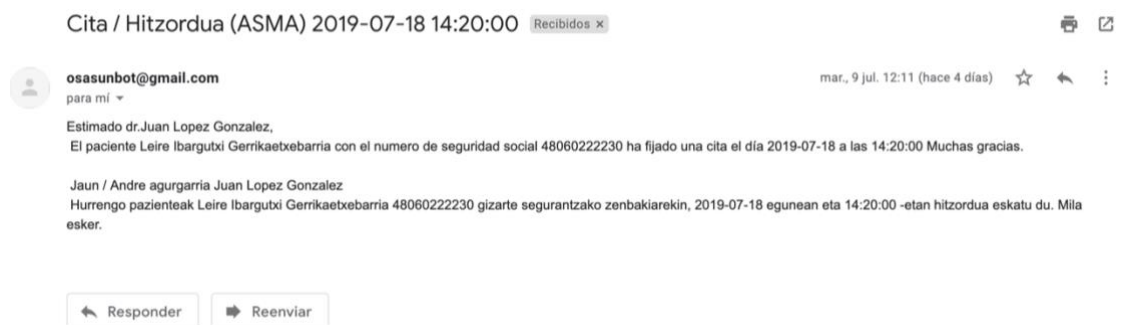
        Session session = Session.getInstance(prop,
            new javax.mail.Authenticator() {
                protected PasswordAuthentication getPasswordAuthentication() {
                    return new PasswordAuthentication(username, password);
                }
            });
        try {
            Message message = new MimeMessage(session);
            message.setFrom(new InternetAddress("from@gmail.com"));
            message.setRecipients(
                Message.RecipientType.TO,
                InternetAddress.parse(email)
            );
            message.setSubject("Cita / Hitzordua (ASMA) "+dia+ " "+hora);
            message.setText("Estimado dr."+nombreMed+", "
                + "\n El paciente "+nombrePac+" con el numero de seguridad social "+numSegSoc+" ha fijado una cita el día "+
                dia+" a las "+hora+" Muchas gracias."
                + "\n\n Jaun / Andre agurgarria "+ nombreMed +
                "\n Hurrengo pazienteak "+nombrePac+" "+numSegSoc+" gizarte segurantzako zerbakiarekin, "+ dia+" egunean eta "+
                hora+" -etan hitzordua eskatu du. Mila esker.");

            Transport.send(message);
        }
    }
}

```

Irudia 17: Korreoa bidaltzeko kodea

Korreoa jasoko den mezua horrelakoa izango da:



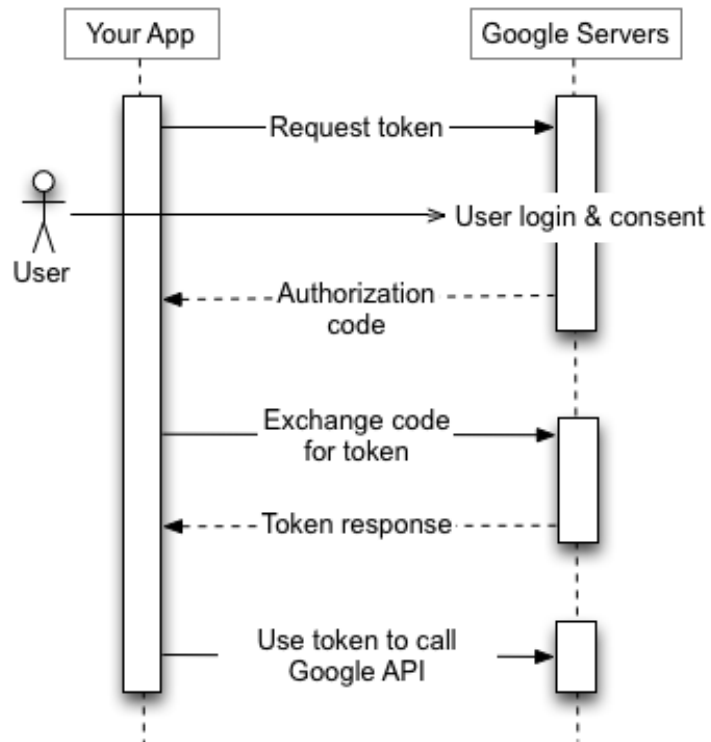
Irudia 18: Jasotako korreoa

6.1.4 Google Calendar-en erabilera

Google Calendarreko API-a erabili ahal izateko, Googleek eskaintzen duen Google Cloud Plataform-en proiektu bat sortu eta erabiliko diren API-ak baimendu behar dira, kasu honetan Google Calendar.

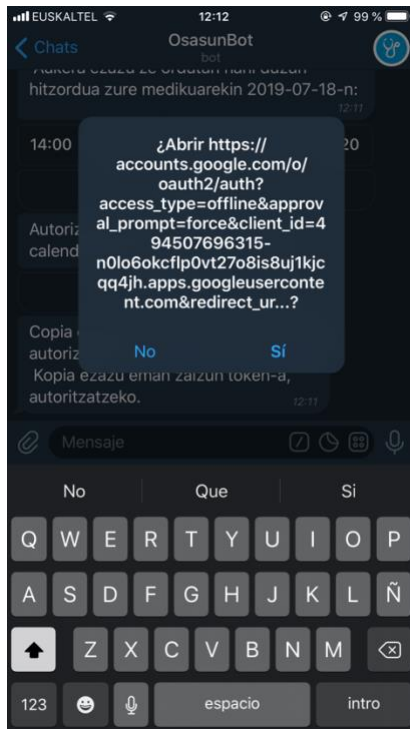
Ondoren sortutako proiektuaren aplikazioa erabiltzeko behar diren kredentzialak sortu behar dira, OAuth 2.0 bidez aplikazioa baimentzea aukeratu dugu proiektu honetarako. OAuth 2.0 eskaerak baimentzeko erabiltzen da, Google-eko erabiltzailea eta pasahitza eskatuz zenbait akzioen baldintzak baimentzen dira. Jarrian dagoen fluxuan ikus daiteke baimentzearen

ordenaren ikuspen orokorra, ondorengo lerroetan proiektuan nola garatu den modu zehatzago batean azalduko da.

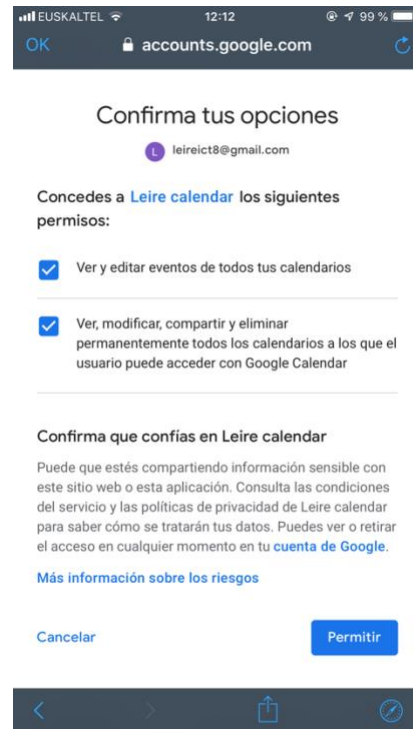


Irudia 19: OAuth 2.0 fluxua Iturria: Alejandro Diaz Torres, DRUPALERA)

Fluxuaren abiapuntua baimena eskatzea da, gure aplikazioa 19. Irudian ikus daitekeen url-ra zuzenduz hasten da, gero 17.irudian korreoa eta pasahitza sartuz baimenduko ditu beharrezko baldintzak.

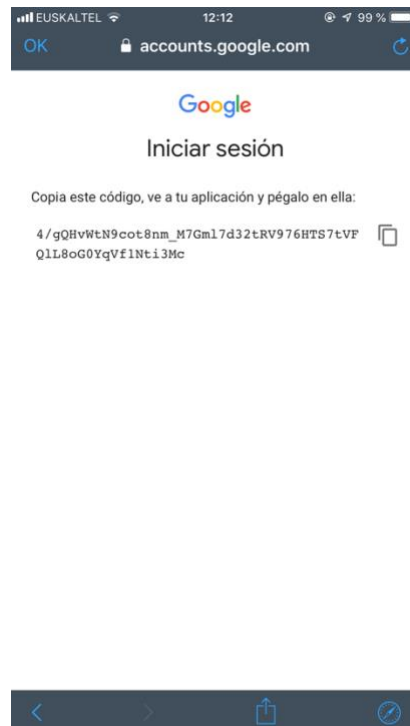


Irudia 20: oAuth 2.0 URL-a



Irudia 21: oAuth 2.0 baimentzea

Pausu horrek baimen kode bat emango digu, Telegrameko elkarrizketan kopiatuz, baimen kode hori Access token-a bihurtuko da , eta azkenik Access token horrekin baimendu diren eskaera guztiak burutu ahal izango dira



Irudia 22: Kopiatu beharreko baimen kodea

Aurrean azaldu den baimen fluxua implementatzeko google-oauth-client-jetty eta google-api-client liburutegiak erabili dira, eta GoogleAuthorizationCodeFlow bidez lortu da lehenengo baimen eskaeraren URL-a, horretarako honako eremu hauek zehaztu behar izan dira:

- Client id-a eta client secret-a: Google Cloud Plataform-en erregistratutako proiektuaren kredentzialetatik ateratzen dira bi datu hauek, client id-ak aplikazioa publikoki identifikatzen du eta client secret-ak funtzio bera dauka baina bakarrik aplikazioak eta baimen zerbitzariak ezagut dezakete.
- Redirect uri-a: Korreo bidez baimena eman eta gero, erabiltzaileari birhelbideratzen zaion web orria da, horri baimentzeko beharrezkoa den baimen kodea gehituko zaio URL-an. Aplikazio honen kasuan arazo bat eduki zen JAVA kode bitartez ez zen birhelbideratutako URL-a eskuratzen eta horrela baimen kodea lortzen; hori konpontzearen "urn:ietf:wg:oauth:2.0:oob" zehaztu da redirect-uri moduan, URL baten ordean, horrek pantailan baimen kodea agertzea ahalbidetzen du, 22. Irudian ikus daitekeenez.
- Scopes: Scopes atalean behar diren baimenak zehazten dira, aplikazio honen kasuan egutegia ikusteko eta egutegian hitzorduak aldatzeko(gehitu, moldatu, ezabatu) baimenak zehaztu dira.

Ondoren, baimen kodea GoogleAuthorizationCodeTokenRequest-ekin bihurtzen da Access token eta orduan baimen fluxua bukatutzat eman daiteke.

Azkenik, Google Calendar-en hitzordua gehitzeko google-api-services-calendar v3 liburutegia erabili da, horretan Calendar.builder -en bidez zerbitzari bat sor daiteke, Access token-a zehaztuz, eta zerbitzu horren bidez hitzorduaren datuak dituen Event objektua bidaltzan zaio. Jarraian adibide gisa hartu den kodea ikus daiteke:

```
Calendar service = new Calendar.Builder(transport, jsonFactory, cred).setApplicationName("Leire calendar")
    .build();
Event event = new Event().setSummary("Cita médico asma");
String fecha= diaCita.toString().concat("+").concat(horaCita.toString());

org.joda.time.format.DateTimeFormatter formatter = DateTimeFormat.forPattern("yyyy-MM-dd+HH:mm:ss");
org.joda.time.DateTime dt = formatter.parseDateTime(fecha);
System.out.println(dt.toString());
DateTime startDateTime = new DateTime("2019-06-17T17:00:00-07:00");
EventDateTime start = new EventDateTime().setDateTime(startDateTime).setTimeZone("Europe/Madrid");
event.setStart(start);
DateTime endDateTime = new DateTime("2019-06-17T17:00:00-07:00");
EventDateTime end = new EventDateTime().setDateTime(endDateTime).setTimeZone("Europe/Madrid");
event.setEnd(end);
String calendarId = "primary";
event = service.events().insert(calendarId, event).execute();
System.out.printf("Event created: %s\n", event.getHtmlLink());
```

Irudia 23: Google Calendar-ri deia egiteko kodea

6.2 Web aplikazioa

Jarraian web aplikazioaren inplementazioa azalduko da.

6.2.1 Garapen ingurunea

Web aplikazioa garatzeko 2.6 Tresnak atalean azaldu den moduan Atom editorea eta Nodejs erabili dira.

Atom editorean, Eclipsearen moduan, Github pluginaren erabilera egin da, bertsioren kontrola egiteko.

App.js fitxategiak kudeatzen ditu AJAX eskaera guztiak eta MySQL deiak, horrek express eta MySQL liburutegiak erabiltzen ditu. AJAX-ek aukera ematen du web orri bat dagoeneko kargatuta dagoena zerbitzariari informazio berria eskatzea, orria birkargatzearen beharrik izan gabe.

Beste aldetik, CSS estilo fitxategiak ,ejs informazioa nola antolatuta joango den adierazten duten fitxategiak dira, eta JS fitxategiak ditugu. Azken hauek funtzio fitxategiak direla esan daiteke, batik bat AJAX eskaerenbitartez app.js fitxategiari informazioa eskatzen diote eta jasotako informazioa pantailaratzen duten fitxategiak dira.

Web aplikazioan honako osagaiak ageri dira, errazago ulertzearen interfaze grafikoaen bidez erakutsiko dira eta osagai bakoitzari dagokion HTML lengoiako tag-az lagunduta azalduko dira:

.ejs artxibo asko ez izatearren eta aplikazioa modularizatzearen, leiho modalak erabili dira. Leiho modal bat, aplikazio bateko leiho nagusiaren mendeko kontrol grafikoko elementu bat da. Leiho nagusia desgaitu duen modu bat sortzen du, baina ikusgarri mantentzen du, leiho modalarekin, haren aurrean. Login-a leiho modalaren adibide argia da, 24. Irudian ikus daitekeenez.

Irudia 24: Leih modal baten adibidea

Horrez gain taulak(<table> tag-a), informazioa banatzeko sekzioak(<div> tag-a), combobox-ak aukera bat egiteko (<select> eta <option>), orrialdeen artetik nabigatzeko burukoa(<nav> tag-a) eta beste batzuk erabili dira.

Día	Hora
2019-07-17	20:20:00
2019-07-17	14:20:00
2019-08-13	17:20:00
2019-08-13	08:20:00
2019-08-13	08:40:00
2019-08-13	10:00:00
2019-08-13	09:20:00
2019-08-13	12:20:00

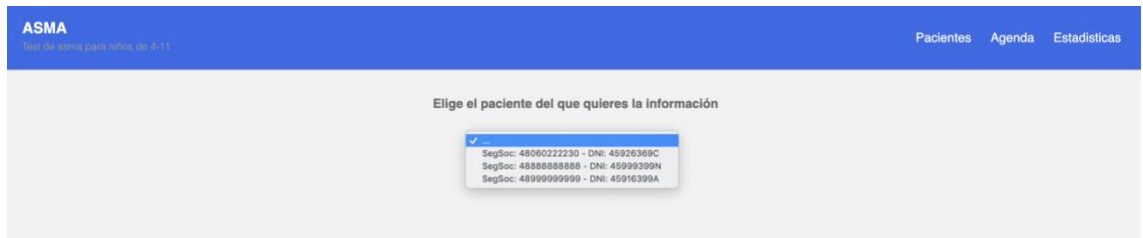
Irudia 25: Taulen eta sekzioen adibidea

25. irudian bi sekzio bereiz ditzakegu alde batetik gaixoaren informazio pertsonala eta beste aldetik hitzorduen historikoen taula.

Irudia 26: Nabigatzeko burukoaren adibidea

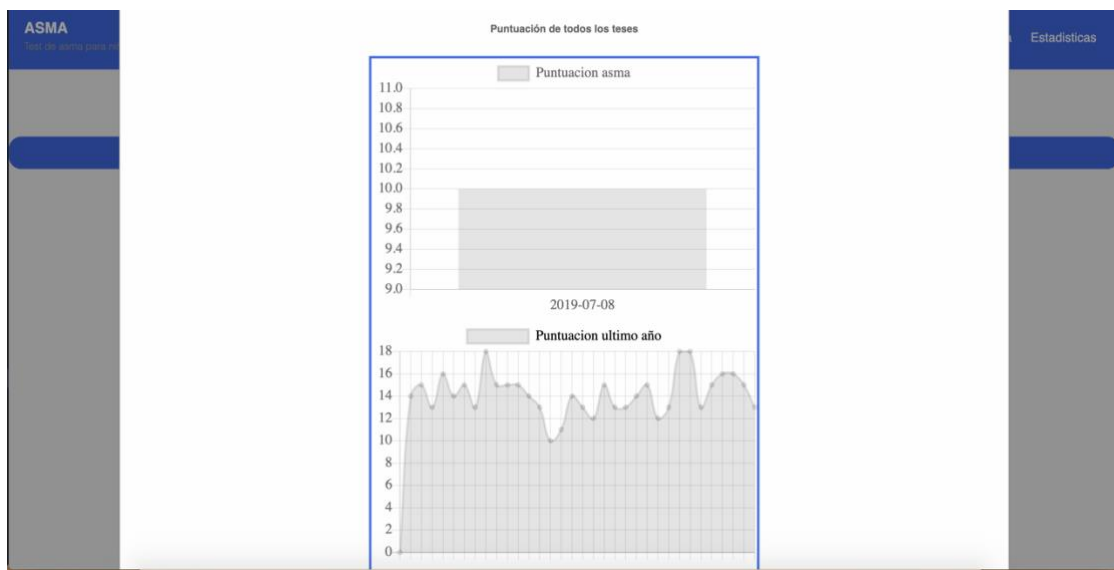
26. irudian Paciente, Agenda edo Estadísticas hitzetan klikatzearen bidez orrialde batetik bestera era eroso batean mugitu ahalko da erabiltzailea.

Gaixoa aukeratzeko <select> tag-a erabili da eta aukerak ezartzeko <option>, 27. Irudian ikus daitekeenez datu basetik mediku horren pazienteak erabili da.



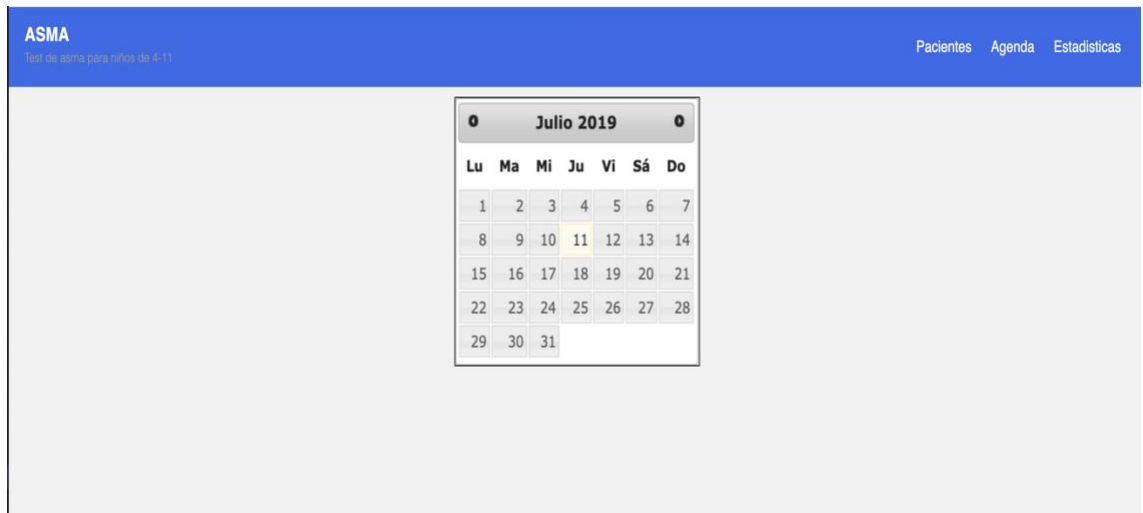
Irudia 27: Gaixoa aukeratzeko aukera menua

Beste aldetik, grafikak sortzeko chart.js liburutegia erabili da, grafikoa pantailan zehazteko <canvas> tag-a behar da. <canvas> etiketa grafikoak diseinatzeko erabiltzen da, liburutegien bidez (normalean JavaScript).



Irudia 28: Grafiken adibidea

Azkenik, egutegia erakusteko jQuery liburutegiko datepicker osagaia erabili da, sekzio ,<div>, baten barruan kokatuz.



Irudia 29: egutegiaren adibidea

Datepicker aukeratu da egun baten gainean klik egitean funtzioak exekutzatzea ahalbideratzen duelako. Web aplikazio honen kasuan, medikuak egun horretan zehaztuta dituen hitzorduen taula agertuko da.

Hora	SegSocialNum	Nombre	Apellido
08:20:00	48060222230	Leire	Ibargutxi Gerrikaetxebarria
08:40:00	48060222230	Leire	Ibargutxi Gerrikaetxebarria
09:20:00	48060222230	Leire	Ibargutxi Gerrikaetxebarria
09:40:00	48999999999	Marta	Lopez Ebebarria
10:00:00	48060222230	Leire	Ibargutxi Gerrikaetxebarria
10:20:00	48999999999	Marta	Lopez Ebebarria
10:40:00	48999999999	Marta	Lopez Ebebarria
11:00:00	48999999999	Marta	Lopez Ebebarria
11:20:00	48999999999	Marta	Lopez Ebebarria
11:40:00	48999999999	Marta	Lopez Ebebarria
12:00:00	48999999999	Marta	Lopez Ebebarria
12:20:00	48060222230	Leire	Ibargutxi Gerrikaetxebarria
12:40:00	48999999999	Marta	Lopez Ebebarria
13:00:00	48999999999	Marta	Lopez Ebebarria
13:20:00	48999999999	Marta	Lopez Ebebarria

Irudia 30: Egutegiaren leiho modalaren adibidea (Abuztuak 14 eguneko hitzorduek)

Azpimarratzekoak ez diren osagai sinpleagoak ere erabili dira botoiak <button>, idazteko formularioak <input>, etiketak <label>, etab.

7. Probak

Atal honetan inplementatutako atal bakoitzaren egokitasuna egiaztatzeko egin diren proben zerrenda aurkezten da. Proiektuaren kalitateak zerikusi zuzena dauka egin diren proben kantitate eta konplexutasunarekin. Ulergarriagoa izateko prototipo bakoitzean garatu diren probak aipatuko dira. Lehenego 5 prototipoak gaixoaren Telegram-eko bot-areei buruzko probak dira eta azken 5ak medikuaren web aplikazioarenak.

7.1 1.Prototipoa:

Komando arruntenei erantzuna ematea eta test-aren galderak erakustea zen lehen prototipoaren helburua. /start, /info eta /osasun dira tratatu diren komandoak.

1.1 Proba
Deskribapena: /start eta /info komandoei erantzun egokia, bi hizkuntzetan.
Eraitza: Proba egokia, euskaraz eta gaztelaniaz eman da informazioa. /start sakatzean testa hasteko /osasun tekleatu behar duzula agertzen da eta /info sakatzean bot-ari buruzko informazioa.

1.2 Proba
Deskribapena: /osasun komandoak segurtasun zenbakia eskatu eta umearen adina eta izena eskatu behar du.
Eraitza: Proba egokia, segurtasun zenbakia eta umearen datuak eskatzen ditu.

1.3 Proba
Deskribapena: Segurtasun zenbakia 48,01 edo 20 zenbakiez hasi behar da eta 11 luzera izan behar du
Eraitza: Beste zenbakiekin frogatu da, proba egokia.

1.4 Proba
Deskribapena: Testaren galderak bi hizkuntzatan egiteko aukera, KeyboardButton-en bidez.
Eraitza: Proba egokia. Testaren galderak gaztelaniaz agertu dira, euskaraz ere.

7.2 2.Prototipoa:

Datuak biltegitzea beharrezkoa denez, datu-basearen konexioa egiten da atal honetan. Horrez gain, testa osatzean puntuazio bat lortzen da, hori eta nabarmentasuna duen beste informazioa ere biltegitatuko da.

2.1 Proba
Deskribapena: Datu-basearekin konexioa konprobatu da, taula bateko informazioa eskatuz.
Eraitza: Proba egokia.

2.2 Proba
Deskribapena: Testaren galdera bakoitzaren erantzun guztiak gorde behar dira, item taulan.
Eraitza: Proba egokia. PhpMyAdmin-en frogatu da.

2.3 Proba
Deskribapena: Testaren galdera bakoitzaren erantzunak puntuazio bat dauka, kalkulua egokia izan behar da. Galdera batzuk 3 erantzun posible dituzte eta beste batzuk 5.
Eraitza: Proba egokia. Test originalaz baliatuz konprobatu da eskuz eta bot-ak ematen duen eraitza berdina dela.

2.4 Proba
Deskribapena: Gaixoari galdetutako datuak (Izena, adina ,segurantzaren zenbakia...) eta testari buruzko datu orokorrak (eraitza, egindako data...) gorde behar dira user taulan.
Eraitza: Proba egokia. PhpMyAdmin-en frogatu da.

7.3 3.Prototipoa:

Hitzordua aukeratzeko gaixoak esleituta duen medikuak agendan dituen une libreak agertuko dira, erabiltzaileak aukera dezan.

3.1 Proba
Deskribapena: Medikuek gaurtik aurrera libre dituen egunak agertu behar dira, InlineKeyboardButton-etan
Eraitza: Proba egokia. Telegram-en konprobatu da. Egun zuzenak diren begiratzeko phpMyAdmin ere erabili da.

3.2 Proba
Deskribapena: Eguna aukeratzean, hitzordua esleituta gabe dauden orduak atera behar dira. (Hitzordu batetik bestera 20 minutuko aldea dago)
Emaitza: Proba egokia. Telegram-en konprobatu da. Ordu zuzenak diren begiratzeko phpMyAdmin ere erabili da.

3.3 Proba
Deskribapena: Hitzordua aukeratzean konprobatu behar da cita taulara gehitu dela, mediku eta gaixoaren datu egokiekin
Emaitza: Proba egokia. PhpMyAdmin-en frogatu da.

7.4 4.Prototipoa:

Medikuari hitzorduari buruzko datuak dituen korreoa bidali zaion egiaztatuko da atal honetan.

4.1 Proba
Deskribapena: Korreoa bidali zaio, aukeratutako data eta gaixoaren informazioa adierazten dituen.
Emaitza: Proba egokia. Gmail-en frogatu da.

7.5 5.Prototipoa:

Gaixoari hitzorduaren eguna haren Google Calendar-ean apuntatuko zaiola egiaztatuko atal honetan.

5.1 Proba
Deskribapena: Baimena eskatu Google Calendar erabili ahal izateko, baimena emateko prozesua Auth 2.0 deitzen da.
Emaitza: Proba okerra. Google-en autorizazio orrira birhelbideraltzen da baina autorizazioa ez da zuzenean egitea lortzen.
Konponbidea: Baimen kode bat kopiatu behar zaio Telegram-ari, Google Calendar-en erabilera baimentzeko

5.1 proba garatzeko hainbat arazo agertu ziren. Baimena eskatzerakoan, 6.1.4 Google Calendar-en erabileran atalean azaldu moduan, baimen kode bat ematen zaio erabiltzaileari. Programatzaileak zehaztutako redirect-uri URL-ari baimen kodea gehitzen zaio eta horretara

birhelbideratzen du erabiltzailea. Kanpoko birhelbideraketa bat denez, saiakera asko egin eta gero ez zen lortu nabigatzailean agertzen den URL-a eskuratzea. Adibidez, saiakera bat JAVAko Scanner klasea erabiltzen egin zen.

JAVA lengoaiaren Google-en API-ei buruzko dokumentazio askorik ez dago, pentsatu zen dena JavaScript lengoaiara igarotzea baina arazo berdina zegoen.

5.2 Proba
Deskribapena: Google Calendar-ean erabiltzaileak aukeratutako egunean hitzordua gehitu da.
Eraitza: Proba egokia. Google Calendar-en konprobatu da egun horretan gertakizun bat ageri dela.

7.6 6.Prototipoa:

Web aplikazioaren login pantailaren erabilera zuzena frogatuko da atal honetan. 7.7 ataleko probak atal honetakoak baino lehenago konprobatu behar dira.

6.1 Proba
Deskribapena: Erabiltzaile edo pasahitz okerrak sartuz gero, erabiltzaileari horren abisua ematen zaio.
Eraitza: Proba egokia, gorri mezua ateratzen da eta erabiltzaileak berriro kautotzeko aukera dauka

6.2 Proba
Deskribapena: Sesioa gordetzen da, aldi batez kautotuz gero.
Eraitza: Proba egokia, nabigatzailea irekita dagoen bitartean ez da berriro kautotu behar da,

7.7 7.Prototipoa:

Datu-basearekin konexio egokia al den frogatu da.

7.1 Proba
Deskribapena: Datu-baseari medikuaren erabiltzailea eta pasahitza eskatu zaizkio, kontsolan erakutsiz.
Eraitza: Kontsolan medikuaren datuak agertu dira.

7.8 8.Prototipoa:

Medikuak egutegi baten bidez, egun bat aukeratzuz horretan zehaztuta dituen hitzorduak ikusi ahaliko ditu, funtzionalitate horren egokitasuna frogatu da.

8.1 Proba
Deskribapena: Hitzorduak dituen egun bat aukeratu da.
Eraitza: Froga egokia, taula bat agertzen da ordu eta pazienteen informazioarekin.

8.2 Proba
Deskribapena: Hitzordurik ez dagoen egun bat aukeratu da.
Eraitza: Froga egokia, hitzordurik ez dagoela adierazten da, alerta baten bidez.

8.3 Proba
Deskribapena: Gaixo baten Telegrameko bot-aren bidez egun batean hitzordua hartu, eta haren mediku moduan kautotu eta hitzordu hori egutegian agertzen baden frogatu da.
Eraitza: Proba egokia, froga egiteko erabilitako gaixoaren datu egokiak agertzen dira.

7.9 9.Prototipoa:

Medikuak modu egokian hark esleituta dituen pazienteen informazioa eta hitzorduen historikoa ikusi ahal du, funtzionalitate horren egokitasuna frogatu da.

9.1 Proba
Deskribapena: Esleituta agertzen diren gaixoak zuzenak diren frogatu da.
Eraitza: Froga egokia,phpMyAdmin erabili da egiaztatzeko.

9.2 Proba
Deskribapena: Gaixo bat aukeratzuz haren informazioa eta hitzorduen historikoa era egokian agertzen direna egiaztatu
Eraitza: Froga egokia.

7.10 10.Prototipoa:

Medikuak asmari buruzko grafika orokorrak eta haren paziente bakoitzaren grafikak ikusi ahal ditu, funtzionalitate horren egokitasuna frogatu da.

10.1 Proba

Deskribapena: Asmaren puntuazio urteak kontuan hartuta eta asmaren puntuazioak hilabeteak kontuan hartuz erakusten dituzten bi grafikak erakutsi behar dira.

Emitza: Froga egokia.

10.2 Proba

Deskribapena: Gaixo bakoitzak hilabetea eta urtero duen asmaren eboluzioa erkusten duten bi grafika agertu behar dira.

Emitza: Froga egokia.

10.3 Proba

Deskribapena: Gaixoak daturik ez baditu, grafikoak zuriz agertu behar dira.

Emitza: Froga egokia.

8. Ondorioak

Atal honetan, proiektua garatu eta gero atera diren ondorioak azaltzen dira. Proiektua burutzerakoan egindako hausnarketak eta denbora zein helburuen konparaketak aipatzen dira.

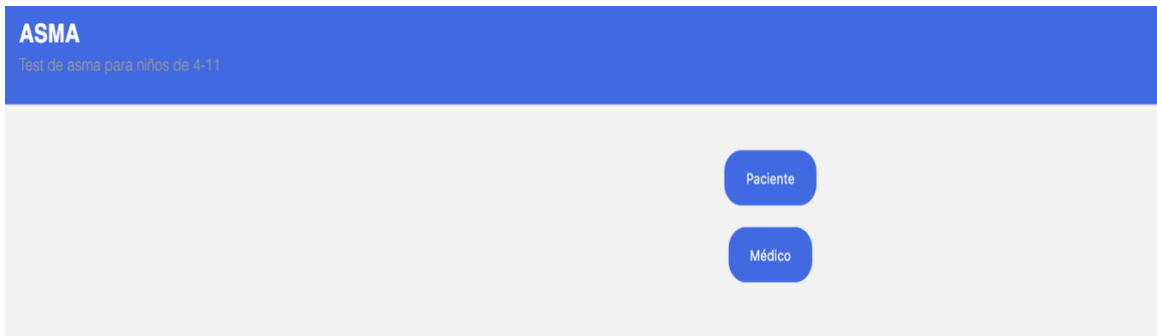
8.1 Betetako helburuak

Dokumentu honen hasieran, 2.1 Helburuak atalean, proiektuak bete beharreko xedeak aipatzen ziren. Jarraian, xede bakoitza konplitu den ala ez azaltzen da eta gainera betetzeko era justifikatzen da.

Proiektua bi erabiltzaileentzat zuzenduta dago, bakoitzak premia eta objektibo desberdinak dituztenak, medikua eta gaixoa.

Gaixoari dagokionez honako helburu hauek zehaztu ziren:

- Era autonomoan umearen asmaren larrialdi maila ezagutzearen aukera izatea eta gainera, osasun zentroarekin kontaktua automatizatzea. Hori, Telegram bot tresnaren bidez lortu da, alde batetik bot-ak eskaintzen test-a betez gaixoaren asma maila lortzen da eta gainera kontaktu zuzena eskaintzan dio erabiltzaileari haren medikuarekin, hitzordua zuzenean eska dezake.
- Datu-Babes-Legea betetzea. Gaixoak ez ditu edonoren datu pertsonal ezta klinikoak jasotzen, eta kontrolatu da gaixo bakoitzaren datu pertsonalak bakarrik hark esleituta daukan medikuak eskura ditzakeela.
- Bakarka umearen asmaren eboluzioa kontrolatzea. Helburu hau ez da guztiz bete, proiektua aurrera zihoala erabaki zen medikua izan behar zela gaixotasunaren jarraipena kontrolatzen zuena, ez gaixoa. Gaixoak testa egin dezake nahi duenean eta noski asma larrialdi maila jakin, baina eboluzioa ikusi nahi badu espezialistara joatearen ideia sustatu behar zela erabaki zen. Horregatik erabaki zen asmaren eboluzio grafikak bakarrik medikuak ikusi ahal dituela, web aplikazioan. 26. Irudian ikus daitekeenez web aplikazioa irekitzean bi aukera izango dituzu, pazientea bazara Telegram bot-era birhelbideratuko zaizu, berriz medikua izanez gero kautotu eta analisiak ikusi ahalko dituzu.



Irudia 26: Web aplikazioaren lehen pantaila

Beste aldetik, medikuaren helburuen artean honako hauek nabarmendu ahal dira:

5. Haren erriaren asmaren eboluzioa grafikoki ikustea, tratamendua hobetzearren. Lehen aipatu bezala, medikuari web aplikazioak grafikoki gaixoen jarraipena egitea ahalbidetzen dio. Gainera, ezagutzak eskuratzearen grafika orokorrak ikusi ahal ditu.
6. Hitzorduak kontrolpean izatea, horretarako egutegi bat dauka web aplikazioan.
7. Haren erien informazio eta historikoa aztertzeko era erosoak. Web aplikazioan ere, haren pazienteen artean aukeratu eta haien informazio pertsonala eta hitzorduen historikoa ikus dezake.

Azkenik, helburu orokortzat bot-a eta web aplikazioa erabilerraza, seguruak eta sinpleak izatea zehaztu zen, eta esan daiteke hirurak bete direla.

Laburbilduz, estimatutako helburu guztiak orokorrean egokitasunez bete direla ondoriozta daiteke, proiektuaren jomuga nagusia betez; jomuga umeen asma antzemateko eta haren jarraipena egiteko tresna erakargarri bat sortzea izanda.

8.2 Egindako plangintza vs. errealitatea

Proiektua zehazterako unean, denbora planifikazio bat garatu zen burutu beharreko ataza guztiak kontuan hartuz. Jarraian, azken emaitza proiektuaren hasieran egindako estimazioekin alderatzen da.

Ataza	Denbora estimazioa	Denbora errealak
KUDEAKETA	23	20
Bilerak	6	4

Helburu eta atazen zehaztapena	3	4
Atazen denboraren zehaztapena	2	2
Prototipoen kontrola	2	4
Baliabideen prestakuntza	10	6
IKASKETA	40	29
Telegram API	15	15
Google API	10	7
EJS,JS eta CSS	8	5
JavaMail	4	1
Cacoo eta Toms Planner	3	1
DISEINUA	10	5
Lengoaia eta tresnen zehaztapena	5	2
Datu-basearen diseinua	5	3
INPLEMENTAZIOA	200	200
Telegrameko bot	120	130
Web aplikazioa	80	70
PROBAK	37	35
Telegrameko bot-a	23	25
Web aplikazioa	14	10
DOKUMENTAZIOA	100	90
Memoria	80	80
Aurkezpena	20	20
GUZTIRA	410	389

Taula 7: Denbora planifikazio errealaren taula

Denbora planifikatzearen momentuan, proiektua epez entregatzearen denbora tartek gorantz estimatu ziren, egokiagoa zen denbora soberan edukitzea, faltan baino. Hori arriskuak zehaztu zirenean erabaki zen, denbora faltak atazak kanpoan uztea eragingo zuen eta. 7. taulan ikus daitekeenez, estimatutako eta errealtatearen arteko denbora diferentzia ez da oso zabala izan baina estimatutakoa altuagoa da, hori ikusita diseinu egokia egin zela ondoriozta daiteke.

Proiektuaren diseinu eta zehaztapen atazak estimatutakoa baino azkarrago burutu ziren, egin beharreko garapenaren ideia aurretiaz oso argi zegoen eta. Bestalde ikasketa nagusiak hasieran

egin ziren 29 ordukoak izan ziren gutxi gora behera, baina aipagarria da inplementazioa egitearekin batera zenbait ezagutza ere hartu behar izan zirela.

Azkenik, inplementazioari dagokionez Telegram bot-aren garapena erabat berria zen niretzat, eta aurretiaz aipatu den moduan ,6.1.4 Google Calendar-en erabilera atalean, Google-eko OAuth 2.0 baimenarekin zenbait arazo suertatu ziren, atal horretan denbora gehiago inbertitu zen horregatik.

8.3 Arriskuak eta haien agerpenak

2.6 atalean proiektuaren garapenean agertu ahal ziren arazoaren estimazioa egin zen. Jarraian suertatu direnak azaltzen dira, eta garapenean eragin negatiborik ez izatearren, konponbidea nola eman zitzaien zehazten da.

- *Programatzailean arazo pertsonalak edo gaixotasuna.* Zehazki, proiektua hasi nuenean Informatica Euskadi enpresan praktikan nebilan, proiektu berri batean sartu ninduten han, eta horren ondorioz neukan lan karga asko hazi zen. Zenbait arratsaldetan unibertsitate klaseak nituen gero, zenbait egunetan zehaztutako orduak ezin nituen bete. Horregatik, planifikatutako egutegian zenbait aldaketa egin nituen eta asteburuetan sartu nituen ordu asko.
- *Erabilitako tresnerian aldaketa.* Google Calendar-eri baimenak emateko OAuth 2.0 erabili behar da, JAVA-n programatzeari buruzko dokumentazio faltagatik, Javascript lengoaiara igarotzen hasi zen kode guztia. Lortu nahi zena Javascript-en ere ezin zenez lortu azkenean JAVA-n utzi zen. Honek, Telegram-eko bot-ean denbora gehiago ematea ekarri zuen.

8.4 Etorkizunerako lana

Proiektua era egokian garatu da, baina zenbait hobekuntza gehitu ahal zaizkio hura hobetzearren, jarraian azaltzen direnak.

- *Geolokalizazioa:* Gaixoaren kokapena eskuratuz gero, medikuak haren asmaren kontrol zehatzagoa edukiko luke. Kokapenaren arabera analizatu ahalko litzateke asma maila , adibidez Madrilera badoa, kontaminazioa dela eta asma areagotzea...

- *Web aplikazioaren hizkuntza:* Telegram-eko bot-a gaztelaniaz eta euskaraz inplementatu da, web aplikazioa berriz gaztelera bakarrik, hizkuntza aukeratzeko posibilitatea gehitzea gehigarri egokia izango litzateke.
- *Testa betetzeko abisua:* Asmaren jarraipen egokia eta zorrotza egiteko, medikuak zehaztu ahalko litzioke gaixoari testa egin beharreko egunak. Zehaztutako egunetan Telegram-ek abisua emango lioke gaixoari testa bete dezan.
- *Beste motatako gaixotasunentzat gaitzea:* Erabilitako tresnak, bot-a zein web aplikazioa, asma gaitzaren kontrolera zuzenduta daude, jarraipena behar duten beste gaixotasun kronikoen hedapena kontrolatzera ere molda litzateke. Adibidez kardiopatia, hipertentsio edo diabetesa pairatzen duten eriei zuzentzeko molda litzatekeen proiektua.

8.5 Hausnarketa pertsonala

Atal honetan, proiektuaren garapenari buruz orokorki egin den hausnarketa topatzen da.

Proiektuan zehar burututako lana oso interesgarria iruditu zait. Nire asmoa lortutako emaitza, Telegram bot-a zein web aplikazioa, osasun arlora zabaldu ahalko litzatekeen ideiatzat hartzea zen; eredutzat hartuta eta zenbait hobekuntza ezarriz lagungarria izan ahal litzatekeelako.

Informatikak daukan transbertsalitatea oso garrantzitsua dela ikusi dut, edozein arlotara aplikatuz aurreratze handiak ekar litzatekeelako.

Horrez gain, tresna berri asko erabili ditut eta proiektu handi bat eta denbora luzean luzatzen dena era autonomoan kudeatzen ikasi dut. Proiektua epean entregatzeko eta bete nahi nituen puntu guztiak biltzeko, jomugak ezarri nituen, hots data zehatz batzuetarako lan kantitate bat ezarri nuen. Hasieran kostatu zitzaidan epeak betetzea eta atzeratu egiten nituen edo epe horretarako ezarritako lan karga gutxitu, horrek zenbait aldiz birplanifikatzera eraman ninduen, baina azkenean ondo antolatzea lortu nuen.

Aurreko guztia laburbilduz, esperientzia aberasgarria topatu dut ezagutza asko eskuratu ditudalako. Proiektu honetan lagundu didatenei eskertu nahiko nieke, erabilgarria izan daitekeen zerbait sortzen lagundu didatelako, eta etorkizunean aplikatu ahal ditudan ezagutzak eman dizkidatelako.

Erreferentziak

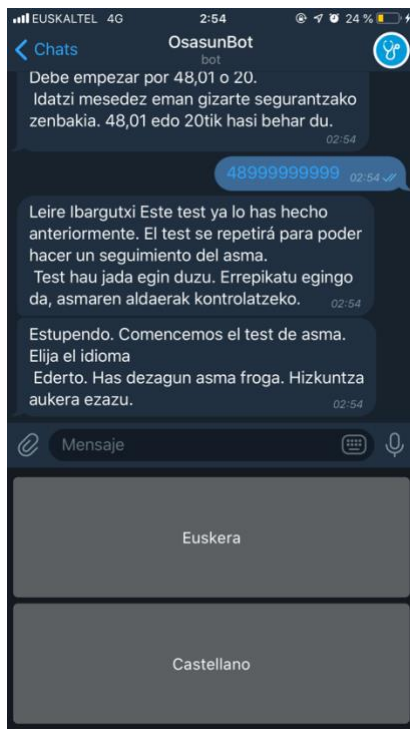
- 1- [Tom's Planner] <https://www.tomsplanner.es>.
- 2- [The Global Asthma Report 2014] www.globalasthmareport.org/burden/burden.php
- 3- [Estudio ASMACOST] (Martínez-Moragón E et al. Arch Bronconeumol 2009; 45:481-486).
- 4- [Telegram, 2019] Telegram Bot API. <https://core.telegram.org/bots/api>. [Telegram Bot API 4.3, 2019-05-31].
- 5- [Google cloud Administración del API] Arquitectura de endpoints <https://cloud.google.com/endpoints/docs/grpc/architecture-overview?hl=es-419>
- 6- [DRUPALERA] <https://www.ladrupalera.com/es/drupal/desarrollo/javascript/como-usar-una-api-de-google-con-autenticacion-traves-de-oauth2> [Díaz Torres, Alejandro, 2015-03-20]
- 7- [IBM] <https://www.ibm.com/developerworks/ssa/java/tutorials/j-introjava1/index.html> [J.Stephen Perry, 2012-12-03]
- 8- [Google API Client Libraries] Google Calendar: <https://developers.google.com/api-client-library/java/apis/calendar/v3> Google oAuth: <https://developers.google.com/api-client-library/java/google-api-java-client/oauth2>

I. Eranskina: Telegram-eko bot-aren erabiltzaile gida

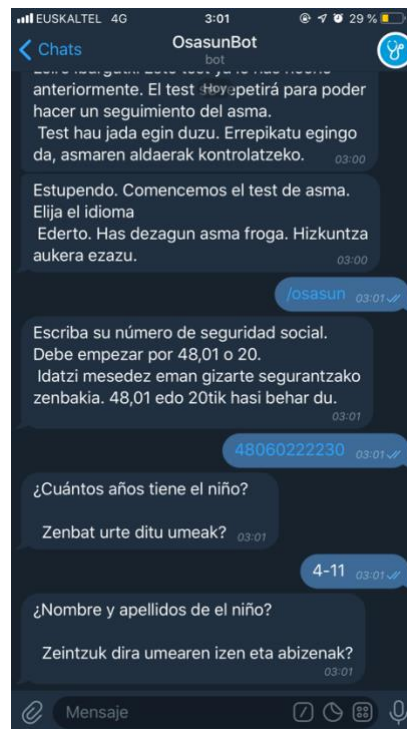
Bot-ari hasiera emateko /osasun komandoa idatzi behar da. /start komandoak bot-ari hasiera emateko informazioa emango du eta /info komandoak bot-ari buruzko informazioa.

Hasieran bot-ak segurantza zenbakia eskatuko du.

Erabiltzaileak testa aurretiaz egin badu (a. irudia) ez dio informazio pertsonalik galdetuko, berriz lehenengo aldia bada (b.irudia) izen abizen eta adinak eskatuko dizkio.



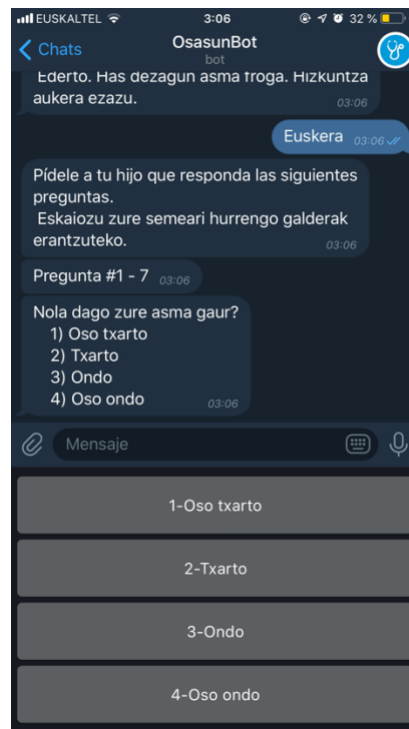
(a) Aurretiaz testa egin duen erabiltzailea



(b) Erabiltzaile berria

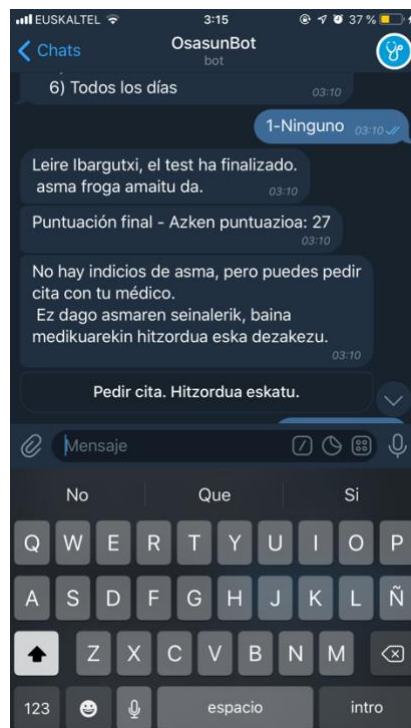
Irudia 31: Erabiltzailearen informazioa bot-ean

Hurrengo pausua, test-a egin nahi den hizkuntza aukeratzea da, hori egin eta gero erabiltzaileak testa bete beharko du.

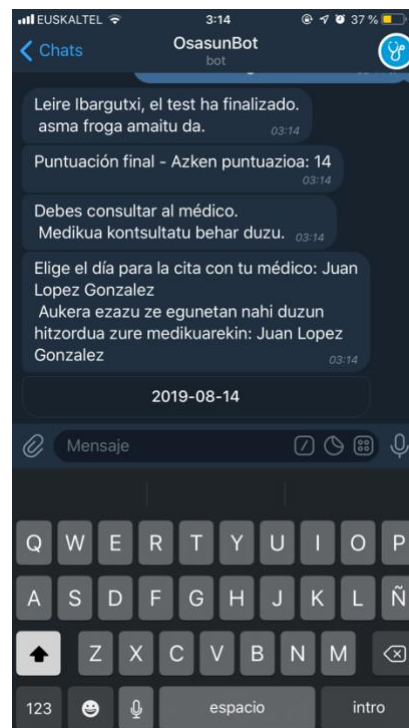


Irudia 32: Testaren galderen eredua

Testaren 7 galderak burutu eta gero emaitzaren bi posibilidadetara daude, puntuazioa 19 baino txikiagoa bada umeak asma seinaleak dituela adierazten du eta zuzenean hitzordua eskatzera eramango zaio (a. irudia), berriz handiagoa bada, posibilidadetara emango zaio hitzordua eskatu ahal izateko (b. irudia)



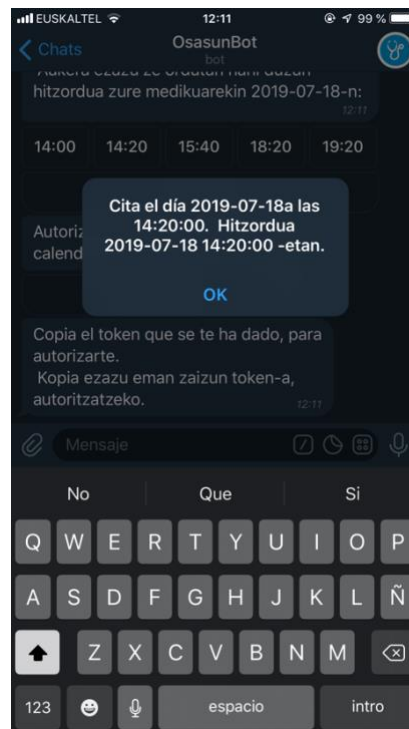
(a) Testaren emaitza negatiboa (asmarik ez)



(b) Testaren emaitza positiboa (asmaren seinaleak)

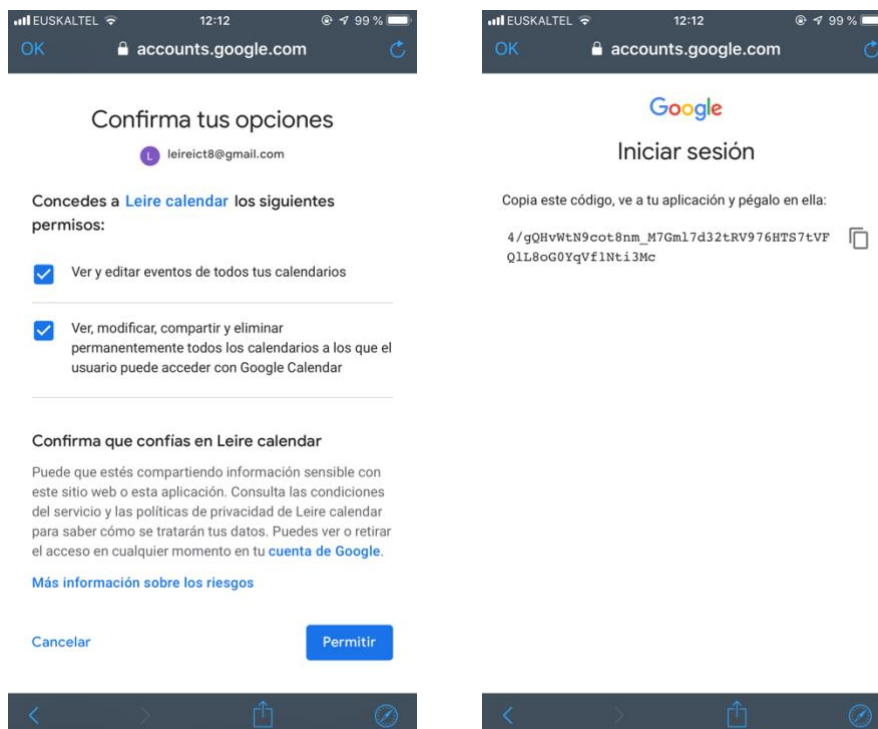
Irudia 33: Testaren puntuazioa

Azkenik hitzordua aukeratu beharko du erabiltzaileak, ondoren pantailan alerta bat (34. Irudia) agertzen denean medikuari korreo bidez hitzorduaren abisua eman zaiola esan nahi du.



Irudia 34: Medikuari hitzordua eskatzea

Azken funtzionalitatea hitzordua erabiltzailearen Google Calendar-ean gordetzea da, erabiltzaileak “Autorizar” botoia sakatuz, baimenak eman, pantailaratzen zaion kodea (35. Irudia) elkarrizketara kopiatu eta zuzenean Google Calendar-era gehituko zaio hitzordua



Irudia 35: Google Calendar-eko baimenak ematea eta autorizazio kodea

II. eranskina: Web aplikazioaren erabiltzaile gida

Web aplikazioa erabili ahal izateko erabiltzailea kakotu egin beharko da (36. Irudia).

Irudia 36: Kakotzeko pantaila

Orduan hiru funtzionalitate izango ditu erabiltzaileak, pantailaren eskumako goikaldean (37. Irudian) nabigatu ahal da batetik bestera.

1- Pacientes:

Funtzionalitate honen bidez, medikuak haren pazienteen artean aukeratu(37. Irudia) eta haien informazio pertsonala eta hitzorduen historikoa ikusi ahalko du.

Irudia 37: Gaixoaren aukeraketa

ASMA
Test de asma para niños de 4-11

Datos del paciente/Gaixoaren datuak

Nombre: Leire
Apellido: Ibargutxi Gerrikaetxebarria
DNI:
Número de Seg. Social:
Fecha Nacimiento: 2013-01-04T23:00:00.000Z
Sexo: F

Proximas citas con el paciente/Hurrengo hitzorduak gaixoarekin

Día	Hora
2019-07-17	20:20:00
2019-07-17	14:20:00
2019-08-13	17:20:00
2019-08-13	08:20:00
2019-08-13	08:40:00
2019-08-13	10:00:00
2019-08-13	09:20:00
2019-08-13	12:20:00

Cancel

Estadísticas

Irudia 38: Gaixoaren informazioa

2- Agenda:

Erabiltzaileak haren egutegiaren kontrola ahalbidetzen dio (39. Irudia), egun bakoitzean daukan hitzorduen informazioa ikus dezake (40. Irudia).

ASMA
Test de asma para niños de 4-11

Pacientes Agenda Estadísticas

Julio 2019

Lu	Ma	Mi	Ju	Vi	Sá	Do
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Irudia 39: Hitzorduen egutegia

Hora	SegSocialNum	Nombre	Apellido
08:20:00	4806022230	Leire	Ibargubi Gerrikaetxebarria
08:40:00	4806022230	Leire	Ibargubi Gerrikaetxebarria
09:20:00	4806022230	Leire	Ibargubi Gerrikaetxebarria
09:40:00	4899999999	Marta	Lopez Etxebarria
10:00:00	4806022230	Leire	Ibargubi Gerrikaetxebarria
10:20:00	4899999999	Marta	Lopez Etxebarria
10:40:00	4899999999	Marta	Lopez Etxebarria
11:00:00	4899999999	Marta	Lopez Etxebarria
11:20:00	4899999999	Marta	Lopez Etxebarria
11:40:00	4899999999	Marta	Lopez Etxebarria
12:00:00	4899999999	Marta	Lopez Etxebarria
12:20:00	4806022230	Leire	Ibargubi Gerrikaetxebarria
12:40:00	4899999999	Marta	Lopez Etxebarria
13:00:00	4899999999	Marta	Lopez Etxebarria
13:20:00	4899999999	Marta	Lopez Etxebarria

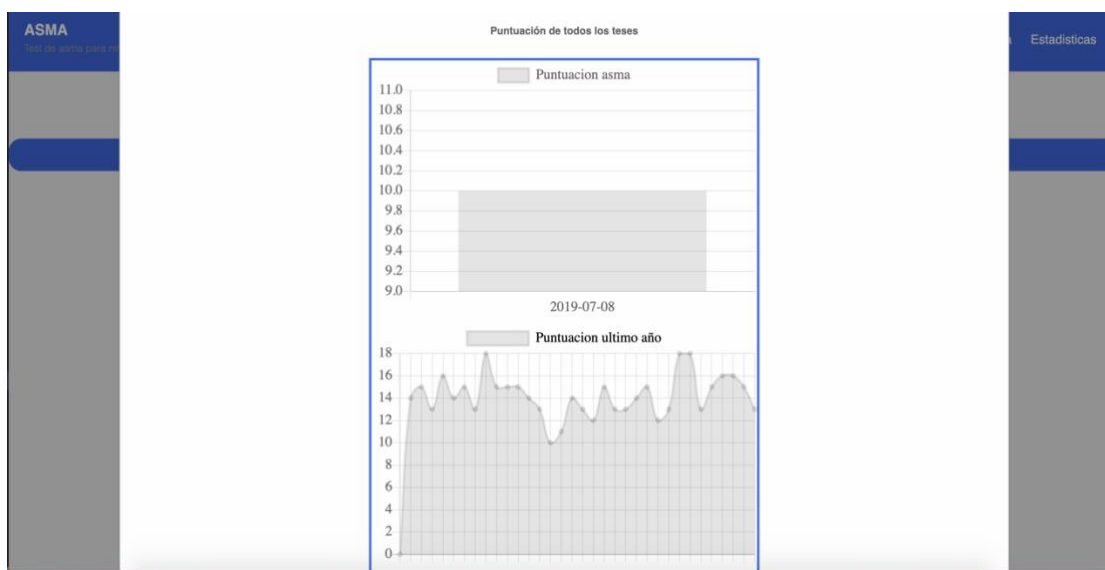
Irudia 40: Hitzorduaren informazioa

3- Estadísticas

Asmari buruzko estadistika orokorrak eta gaixoaren estadistika indibidualak ikustea ahalbidetzen du (41. Irudian), grafikak 42. Irudian moduan pantailaratuko dira.



Irudia 41: Estatistiken pantaila



Irudia 42: Gaixoaren grafikoaren adibidea

