



LATVIJAS UNIVERSITĀTES
DATORIKAS FAKULTĀTE

Android lietotne ceļu locītavu rehabilitācijai

KVALIFIKĀCIJAS DARBS

Autors: Austris Cīrulnieks

Studenta apliecības Nr.: ac14027

Darba vadītājs: B.dat. Emil Syundyukov

Elektronikas un Datorzinātņu Institūts, asistents

RĪGA 2016

ANOTĀCIJA

Kvalifikācijas darbā ir izstrādāta Android lietotne ceļa locītavu rehabilitācijai. Tā sastāv no divām daļām, kas katra paredzēta savai lietotāju grupai.

Pirmā daļa ir paredzēta pacientam, kurš veic rehabilitācijas procesu. Tā komunicē ar iegulto ierīci, kas tiek stiprināta pie pacienta locītavas un ar sensoru palīdzību raida informāciju uz Android ierīci. Tā nodrošina iespēja sekot līdzi vingrinājumu pildīšanas kvalitātei reāllaikā, papildus sūtot datus ārstam.

Savukārt otrā daļa ir paredzēta ārstam. Tajā būs iespēja sekot līdzi savu pacientu veselības stāvoklim atveseļošanās laikā, ērti un uzskatāmi pārtraugot rehabilitācijas procesu.

Izstrādāto lietotņu sistēmas mērķis ir atvieglot rehabilitācijas procesu kā ārstam, tā arī pacientam. Atveseļošanās nereti tiek pārtraukta, kas spēj izraisīt papildus traumas un pasliktināt veselības stāvokli. Pateicoties izstrādātajai Android lietotnei, ārsts spētu nodrošināt pārtraugāmu un kvalitatīvu rehabilitācijas procesu, samazinot pacienta traumu iegūšanas risku. Šis process potenciāli spētu uzlabot atveseļošanās programmas.

Atslēgas vārdi: Android, rehabilitācija, datu analīze, sensori, lietojumprogramma, mākoņkrātuve.

ABSTRACT

The qualification work consists of Android application for knee joint rehabilitation. It contains two main parts which both are made for different user groups.

The first is made for patients in order to monitor and support their rehabilitation process. It communicates with embedded device which is attached to patients' knee. The devices transmit information simultaneously to the application.

The second is made for doctors. It lets its users keep up with patients' data and monitor its rehabilitation process.

Both applications create a system which eases rehabilitation process for both patient and doctor. Aforementioned process quite often gets interrupted due to additional injuries resulting from mistreatment. One of the systems' aims is to reduce this injury cause to a minimum with potential for doctors to improve and consummate rehabilitation programmes.

Key Words: Android, rehabilitation, data analysis, sensors, application, cloud database.

SATURS

1. IEVADS	7
1.1. Tēmas izvēle	7
1.2. Darba mērķis un uzdevumi	7
1.3. Darba struktūra	7
2. Programmatūras prasību specifikācija	8
2.1. Ievads	8
2.1.1. Nolūks.....	8
2.1.2. Darbības sfēra.....	8
2.1.3. Definīcijas, akronīmi un saīsinājumi.....	8
2.1.4. Saistība ar citiem dokumentiem	9
2.2. Vispārējs apraksts	9
2.2.1. Produkta perspektīva	9
2.2.2. Produkta funkcijas	9
2.2.3. Lietotāja raksturiezīmes.....	10
2.2.4. Vispārējie ierobežojumi.....	11
2.2.5. Pieņēmumi un atkarības	11
2.3. Funkcionālās prasības	11
2.3.1. Kopējās funkcijas	11
2.3.2. Pacienta funkcijas	12
2.3.3. Ārsta funkcijas.....	19
2.4. Nefunkcionālās prasības	26
2.4.1. Veiktspējas prasības	26
2.4.2. Drošības prasības.....	27
2.4.3. Pieejamības prasības.....	27
2.4.4. Aparatūras ierobežojumi.....	27

2.4.5.	Uzturamības prasības.....	27
3.	Programmatūras projektējuma apraksts	28
3.1.	Ievads	28
3.1.1.	Nolūks.....	28
3.2.	Dekompozīcijas apraksts	28
3.2.1.	Ievads.....	28
3.2.2.	Moduļu dekompozīcija.....	28
3.3.	Firestore datu bāzes apraksts	49
3.3.1.	Pirmais līmenis – pacienti un ārsti.....	49
3.3.2.	Otrais līmenis – ārsti.....	50
3.3.3.	Otrais līmenis – pacienti	51
3.3.4.	Trešais līmenis – ārsta pacienti.....	52
3.3.5.	Trešais līmenis – pacienta rehabilitācijas nosacījumi	53
3.3.6.	Trešais līmenis – pacienta rehabilitācijas sesiju vēsture	54
3.4.	Datu plūsmu diagrammas	55
3.4.1.	Datu plūsmu diagrammas 0. līmenis	55
3.4.2.	Pacienta datu plūsmu diagramma 1. līmenis	56
3.4.3.	Pacienta datu plūsmu diagramma 2. līmenis “Pacienta pārvaldība”	57
3.4.4.	Pacienta datu plūsmu diagramma 2. līmenis “Pacienta rehabilitācijas pārvaldība” 58	
3.4.6.	Ārsta datu plūsmu diagramma 1. līmenis	59
3.4.7.	Ārsta datu plūsmu diagramma 2. līmenis “Ārsta pārvaldība”	60
3.4.8.	Ārsta datu plūsmu diagramma 2. līmenis “Rehabilitācijas datu pārvaldība”	61
3.4.9.	Ārsta datu plūsmu diagramma 2. līmenis “Pacienta datu pārvaldība”	61
3.5.	Saskarne ar ārējām sistēmām	62
3.6.	Lietotnes saskarņu diagrammas	62
3.6.1.	Pieslēgšanās, reģistrācija un autentifikācija sistēmā	63

3.6.2.	Pacienta lietotnes daļa	64
3.6.3.	Ārsta lietotnes daļa	65
3.7.	Ekrānformu projektējums	66
3.7.1.	Lietotāja tipa izvēles skats	66
3.7.2.	Sākuma izvēles skats	67
3.7.3.	Reģistrācijas skats	68
3.7.4.	Pieslēgšanās skats	69
3.7.5.	Lietotnes pacienta daļas ekrānformas	70
3.7.6.	Lietotnes ārsta daļas ekrānformas	75
4.	Testēšanas dokumentācija	82
4.1.	Ievads	82
4.2.	Testēšanas metodika	82
4.3.	Testēšanas žurnāls	82
4.3.1.	Lietotāja tipa izvēles veikšanas tests	82
4.3.2.	Pacienta reģistrācija sistēmā	83
4.3.3.	Pacienta autentifikācija sistēmā	85
4.3.4.	Pacienta navigācijas izvēlnes izveide	86
4.3.5.	Ārsta reģistrācija	86
4.2.6.	Ārsta autentifikācija sistēmā	88
4.2.7.	Ārsta navigācijas izvēlnes izveide	88
4.2.8.	Pacienta pievienošana	89
4.2.9.	Pievienoto pacientu attēlošana sarakstā	89
4.2.10.	Rehabilitācijas norādījumu uzstādīšana	90
4.2.11.	Rehabilitācijas nosacījumu dzēšana	91
4.2.12.	Rehabilitācijas veikšana un reāllaika datu aplūkošana	91
4.2.13.	Rehabilitācijas vēstures attēlošana	94
4.2.14.	Ārsta rehabilitācijas norādījumu attēlošana	94

4.2.15. Pievienoto pacientu rehabilitācijas vēstures aplūkošana	95
4.2.16. Pacienta paroles maiņa	96
4.2.17. Ārsta paroles maiņa	97
4.2.18. Pacienta konta dzēšana	98
4.2.19. Ārsta konta dzēšana	98
5. Projekta organizācija	99
6. Darbietilpības novērtējums.....	100
7. Kvalitātes nodrošināšana	101
8. Konfigurāciju pārvaldība.....	102
9. Rezultāti.....	103
10. Secinājumi	104
11. Izmantotās literatūras saraksts	105
12. PIELIKUMI.....	106

1. IEVADS

1.1. Tēmas izvēle

Šī tēma tika izvēlēta, jo tā likās interesanta un aizraujoša. Veselības tehnoloģijas ir strauji augoša un inovatīva tēma. Pasaulē tā šobrīd ļoti strauji attīstās. Veselība mūsdienās tiek negatīvi ietekmēta no dažādiem aspektiem, tāpēc eksperti uzskata, ka ir jāvērs ar vien lielāka uzmanība tās uzlabošanā un kvalitātē.

1.2. Darba mērķis un uzdevumi

Šobrīd eksistē iegulta iekārta un mobilas lietotnes prototips, kas spēj iegūt ceļa locītavu kustību datus no iegultās ierīces [1]. Darba uzdevums ir izstrādāt lietotni, ko ērti un viegli varētu lietot gan ārsti, gan pacienti rehabilitācijas procesā. Galvenie darba uzdevumi ir nodrošināt reģistrācijas un autentifikācijas iespējas, datu savstarpēju plūsmu starp ārstu un pacientu, izmantojot datu mākoņkrātuvi. Darba mērķis ir atvieglot rehabilitācijas procesu gan ārstiem, gan pacientiem, papildus sniedzot iespēju ārstiem uzlabot rehabilitāciju programmas, balstoties uz lietotnē gūtajiem datiem.

1.3. Darba struktūra

Darbs sastāv no vairākām daļām – programmatūras prasību specifikācijas, programmatūras projektējuma apraksta, testēšanas dokumentācijas, projekta pārvaldības apraksta, darbietilpības novērtējuma, dažādiem pielikumiem un papildinājumiem.

Projekta izstrādei tika izvēlēta programmēšanas valoda *Java* un izstrādes vide *Android Studio*. Papildus tika izmantota *Firebase* reāllaika mākoņkrātuve.

2. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

2.1. Ievads

2.1.1. Nolūks

Programmatūras prasību specifikācijas (PPS) nolūks ir aprakstīt Android ceļu rehabilitācijas lietotnes prasības. Šis dokuments ir paredzēts produkta izstrādātājiem un pasūtītājam, veiksmīgai tā realizācijai. Prasību specifikācijā tiek raksturota funkcionalitāte un formulētas prasības sistēmai.

2.1.2. Darbības sfēra

Android lietotne ir paredzēta divām lietotāju grupām – ceļu rehabilitāciju nodrošinošajiem ārstiem un viņu pacientiem. Tā ļaus pacientiem sekot līdzi rehabilitācijas gaitai, reāllaikā pārraugot ceļa ieliekuma leņķi, sūtot datus par katru rehabilitācijas sesiju ārstam. Ārsts šiem datiem varēs ērti un viegli sekot līdzi, veicot un izdarot secinājumus par rehabilitācijas procesu. Lietotņu sistēma atvieglos rehabilitācijas pārraudzību, kā arī ilgtermiņā ļaus veikt secinājumus par rehabilitācijas metožu efektivitāti un to potenciālu pilnveidošanu.

2.1.3. Definīcijas, akronīmi un saīsinājumi

Android - operētājsistēma mobilajām iekārtām, kas darbojas uz Linux kodola bāzes;

Java – objektorientēta programmēšanas valoda;

XML – paplašinātā iezīmēšanas valoda;

Firebase – reāllaika mākoņservisa datubāze;

Mākoņkrātuve – datu uzglabāšanas modelis, kurā tie tiek glabāti attālināti;

JSON koks – JSON failu glabāšanas struktūra;

Aktivitāte – Android grafiskās saskarnes struktūras komponente, kas iekļauj skatu, kurā lietotājs var veikt darbības;

Fragments – Android grafiskās saskarnes struktūras komponente, ko var iekļaut citos fragmentos vai aktivitātēs;

Toast – Android grafiskās saskarnes struktūras komponente, kas attēlo īsus vai vidēji īsus paziņojumus;

Bluetooth – maza darbības rādiusa bezvadu datortīklu standarts;

SHA1 algoritms – jaucējadrešēšanas algoritms failu šifrēšanai;

API – lietojumprogrammas saskarne, kas ir iepriekš definētu klašu, procedūru, funkciju, struktūru un konstanšu kopums un tiek pasniegta kā pielikums;

Android API līmenis – Android sistēmas versiju hierarhijas pamatvienība;

Google Guava bibliotēka – bibliotēka *Java* bāzētiem projektiem teksta, I/O un citu failu apstrādei.

2.1.4. Saistība ar citiem dokumentiem

Prasību specifikācija ir izstrādāta saskaņā ar standartu LVS 68:1996 “Programmatūras prasību specifikācijas ceļvedis”.

2.2. Vispārējs apraksts

2.2.1. Produkta perspektīva

Pagaidām lietotņu sistēma paredzēta tikai ceļa locītavu rehabilitācijai, taču ar laiku tā tiks pielāgota arī elkoņu, plaukstas un citām locītavām. Šī sistēma varētu uzlabot veidu, kā ārsti veic rehabilitāciju, samazinot tās laikā gūtās traumas un veicinot pacienta atveseļošanos.

2.2.2. Produkta funkcijas

Produkta funkcijas var sadalīt divus funkciju blokos – pacienta un daktera.

Kopējās funkcijas:

- Lietotāja tipa izvēles veikšana

Pacienta funkciju blokā ietilpst sekojošas funkcijas:

- Pacienta reģistrēšana sistēmā
- Pacienta autentifikācija sistēmā
- Pacienta navigācijas izvēlnes izveide
- Pacienta paroles maiņa
- Pacienta konta dzēšana
- Rehabilitācijas veikšana un reāllaika datu aplūkošana
- Rehabilitācijas vēstures attēlošana
- Ārsta rehabilitācijas norādījumu attēlošana

Ārsta funkciju blokā ietilpst sekojošās funkcijas:

- Ārsta reģistrēšanās sistēmā
- Ārsta autentifikācija sistēmā
- Ārsta paroles maiņa
- Ārsta konta dzēšana
- Ārsta navigācijas izvēlnes izveide
- Pacientu pievienošana
- Pievienoto pacientu attēlošana sarakstā
- Pievienoto pacientu rehabilitācijas vēstures aplūkošana
- Rehabilitācijas norādījumu uzstrādāšana
- Rehabilitācijas nosacījumu dzēšana

2.2.3. Lietotāja raksturiezīmes

Sistēmas lietotāji ir ārsti un pacienti, kas piedalās rehabilitācijas procesā. Tiem nav jābūt īpašām priekšzināšanām, taču pacientiem ir jābūt pieeja iegultajai ierīcei, kas komunicē ar lietotni.

2.2.4. Vispārējie ierobežojumi

Lai sāktu darbu ar lietotni, to vispirms ir nepieciešams instalēt Android ierīcē. Ierīces programmatūras API līmenim ir jābūt vienādam vai lielākam par 21. Lietotnes darbībai ir nepieciešams interneta savienojums, lai pieslēgtos mākoņkrātuvei. Pacienta pilnvērtīgai sistēmas lietošanai ir nepieciešama iegultā ierīce rehabilitācijas datu iegūšanai.

2.2.5. Pieņēmumi un atkarības

Tiek pieņemts, ka nebūs vairāk kā 100 vienlaicīgi datu pieprasījumi mākoņkrātuvei, Android ierīces programmatūras API līmenis ir vienāds vai lielāks par 21.

2.3. Funkcionālās prasības

2.3.1. Kopējās funkcijas

2.3.1.1. Lietotāja tipa izvēles veikšana

Mērķis
Šī funkcija nodrošina iespēju izvēlēties lietotāja tipu (pacients vai ārsts)
Datu ievade
Iespēja nospiegt vienu no divām pogām – pacients vai ārsts.
Datu apstrāde
Nospiežot vienu no divām pogām, izvēloties lietot aplikāciju kā ārsts vai pacients, tiek saglabāts lietotāja tips, kur to tālāk izmantos citas funkcijas un metodes.
Datu izvade
Tiek attēlots skats, kur lietotājs var izvēlēties - pieslēgties vai reģistrēties sistēmā.

2.3.2. Pacienta funkcijas

2.3.2.1. Pacienta reģistrācija sistēmā

Mērķis
Šī funkcija nodrošina pacienta reģistrāciju sistēmā.
Datu ievade
Reģistrācijas formā ir jāaizpilda sekojoši lauki (ievada lietotājs): <ul style="list-style-type: none">• Name (Vārds) – obligāts ievades lauks• Surname (Uzvārds) – obligāts ievades lauks• Gender (Dzimums) – obligāts lauks ar divām izvēles iespējām (male, female)• E-mail (E-pasta adrese) – obligāts ievades lauks• Password (Parole) – obligāts ievades lauks• Password (Parole atkārtoti) – obligāts ievades lauks• Phone number (Telefona numurs) – obligāts ievades lauks
Datu apstrāde
Nospiežot pogu “Register” notiek sekojošas darbības: <ul style="list-style-type: none">• Pārbauda, vai lauki nav atstāti tukši, e-pasts ievadīts pareizā formā, paroles abos ievades laukos sakrīt• Tiek pārbaudīts, vai lietotājs ar šādu e-pastu jau neeksistē datubāzē• Paroles tiek šifrētas pirms saglabāšanas datubāzē• Pēc veiksmīgas datu ievades, tie tiek saglabāti datubāzē Nospiežot pogu “Back” lietotājam tiek attēlots reģistrācijas vai pieslēgšanās izvēles skats.
Datu izvade
Ja kāds no ievadlaukiem ir aizpildīts nekorekti vai nav aizpildīts vispār, parādās kļūdas paziņojums tajā laukā ar attiecīgu paziņojumu (tukšs lauks vai validācija neatbilst kritērijiem – e-pasta un paroles gadījumos). Nesekmīgas reģistrēšanas gadījumā tiek parādīts paziņojums par to.

Pēc veiksmīgas datu validācijas, parādās paziņojums par lietotāja reģistrāciju sistēmā. Pēc paziņojuma parādīšanās tiek attēlots sistēmas pieslēgšanās skats ar e-pasta un paroles ievades laukiem.

2.3.2.2. Pacienta navigācijas izvēlnes izveide

Mērķis
Šī funkcija ir paredzēta, lai lietotājam attēlotu nepieciešamos izvēlnes elementus.
Datu ievade
Pēc pieslēgšanās tiek iegūta informācija par lietotāja tipu (šajā gadījumā atgriezts tips – pacients). Papildus iegūst arī pacienta vārdu, uzvārdu un e-pasta adresi.
Datu apstrāde
Tiek izveidota navigācijas izvēlne, padarot redzamus pacientam paredzētos izvēlnes elementus (Rehab, History, Precet, Bluetooth un Profile).
Datu izvade
Iespējo augstāk minētos izvēles elementus un izvēles galvenē attēlo pacienta vārdu, uzvārdu un e-pasta adresi.

2.3.2.3. Pacienta autentifikācija sistēmā

Mērķis
Šī funkcija ir paredzēta pacienta autentifikācijai sistēmā. Autentifikācija notiek, izmantojot e-pasta adresi un paroli.
Datu ievade
Autentifikācijas formā ir divi datu ievades lauki: <ul style="list-style-type: none">• E-pasts – obligāts ievades lauks• Parole – obligāts ievades lauks

Datu apstrāde
Nospiežot pogu “Login”, tiek pārbaudīts, vai ievadītā e-pasta adrese eksistē datubāzē. Apstiprinoša iznākuma gadījumā ievadītā parolei ā tiek salīdzināta ar datubāzē esošo.
Datu izvade
Neeksistējoša e-pasta ievades gadījumā parādās kļūdas paziņojums par to, ka pacients ar šādu e-pasta adresi sistēmā nav reģistrēts. Ja ievadītā e-pasta adrese eksistē datubāzē, taču ievadītā parole nesakrīt ar datubāzē esošo, tad tiek parādīts kļūdas paziņojums par nepareizas paroles ievadi. Ja e-pasta adrese datubāzē eksistē un ievadītā parole sakrīt ar datubāzē esošo, tiek parādīts paziņojums par veiksmīgu pieslēgšanos sistēmai un tiek parādīts lietotnes galvenās aktivitātes skats, kur izvēlnē tiek attēloti pacienta izvēles elementi.

2.3.2.4. Rehabilitācijas veikšana un reāllaika datu aplūkošana

Mērķis
Šī funkcija ir paredzēta, lai pacients varētu veikt rehabilitācijas sesiju un aplūkot tās datus reāllaikā, saņemot informāciju par sensoru iekārtas stāvokli lietotājam viegli uztveramā veidā.
Datu ievade
Pacients, nospiežot pogu “Start”, var uzsākt rehabilitācijas sesiju. Nospiežot pogu “Stop”, sesija tiek apturēta, pēc kuras nospiešanas pacients var ievadīt komentāru par veikto sesiju. Ar pogu “Reset” tiek notīrīti sesijas dati. Nospiežot pogu “Finish”, tiek veikta sesijas datu saglabāšana.
Datu apstrāde
Tiek sūtīts pieprasījums uz datubāzi, lai noskaidrotu, vai pacientam ārsts ir pievienojis rehabilitācijas norādījumus. Veiksmīga iznākuma gadījumā tiek ielādēta informācija par rehabilitācijas ilgumu minūtēs. Nospiežot pogu “Start”, tiek sākta laika atskaite un iespējota poga “Stop”. Pēc pogas “Stop” nospiešanas, tiek iespējots

komentāra lauks un pogas “Finish” un “Reset”. Nospiežot pogu “Finish”, tiek paziņojuma dialog, kurā lietotājam tiek vaicāts, vai vēlas beigt rehabilitācijas sesiju. Apstiprinošas izvēles gadījumā sesijas dati tiek saglabāti datubāzē.

Paralēli notiek iegultās sensoru iekārtas saņemto datu (kājas saliekšanas leņķis) attēlošana katru sekundi. Informācija tiek salīdzināta ar ārsta rehabilitācijas norādījumos esošo un tiek skaitīts, cik reizes pacients pārsniedz maksimālo ceļgala saliekšanas leņķi.

Datu izvade

Ja pacientam ārsts nav pievienojis rehabilitācijas norādījums, šī informācija tiek attēlota ar paziņojumu un visas pogas un ievadlauki šajā skatā tiek atspējoti.

Pēc apstiprinošas izvēles izdarīšanas paziņojuma dialogā, tiek parādīts paziņojums par rehabilitācijas datu saglabāšanu datubāzē.

2.3.2.5. Rehabilitācijas sesiju kopsavilkuma attēlošana

Mērķis

Šī funkcija ir paredzēta rehabilitācijas kopsavilkuma attēlošanai.

Datu ievade

Datubāzei tiek veikts pieprasījums, vaicājot pēc visiem rehabilitācijas sesiju datiem:

- Rehabilitācijas veikšanas datums un laiks
- Rehabilitācijas sesijas ilgums (minūtes un sekundes)
- Maksimālā leņķa pārsniegšanas skaita reizes
- Rehabilitācijas sesijas komentārs

Datu apstrāde

No datubāzes tiek iegūti iepriekšminētie dati un attēloti saraksta veidā.

Datu izvade

Tiek attēloti sekojoši raksturlielumi:

- Rehabilitācijas sākuma datums un laiks formātā “YYYY-MM-DD HH:MM”
- Rehabilitācijas ilgums formātā “MM min, SS sec”
- Maksimālā leņķa pārsniegšanas skaita reizes
- Rehabilitācijas sesijas komentārs

2.3.2.6. Rehabilitācijas nosacījumu aplūkošana

Mērķis
Šī funkcija ir paredzēta, lai pacients varētu apskatīt ārsta norādījumus rehabilitācijas procesam.
Datu ievade
Datubāzei tiek veikts pieprasījums, vaicājot pēc visiem rehabilitācijas sesiju datiem: <ul style="list-style-type: none"> • Rehabilitāciju veicošā ārsta vārds un uzvārds • Rehabilitāciju veicošā ārsta telefona numurs • Rehabilitācijas ilgums (minūtēs) • Optimālais ceļa saliekšanas leņķis • Maksimālais ceļa saliekšanas leņķis • Biežums (reizes nedēļā)
Datu apstrāde
No datubāzes tiek iegūti iepriekšminētie dati un ievadīti sagataves formā.
Datu izvade
Ja pacientam nav pievienots rehabilitāciju veicošais ārsts, tad tiek parādīts paziņojums ar šo informāciju. Ja pacientam ir ārstējošais ārsts, taču nav pievienoti rehabilitācijas norādījumi, tiek parādīts paziņojums ar šo informāciju. Ja pacientam ir gan ārstējošais ārsts, gan pievienoti rehabilitācijas norādījumi, to informācija tiek attēlota rehabilitācijas nosacījumu aplūkošanas skatā.

2.3.2.7. Pacienta konta informācijas attēlošana

Mērķis
Šī funkcija paredzēta, lai attēlotu pacienta pamatinformāciju. Un iespējotu funkciju 2.3.2.8. (pacienta paroles maiņa) un 2.3.2.9.(pacienta konta dzēšana) izsaukšanu.
Datu ievade
Datubāzei tiek veikts pieprasījums, vaicājot pēc visiem rehabilitācijas sesiju datiem: <ul style="list-style-type: none">• Pacienta vārds un uzvārds• Pacienta dzimums• Pacienta e-pasts• Pacienta telefona numurs
Datu apstrāde
No datubāzes tiek iegūti iepriekšminētie dati un ievadīti sagataves formā. Nospiežot pogu “Change Password”, tiek parādīts paziņojuma dialogs un izsaukta funkcija 2.3.2.9. Nospiežot pogu “Delete Account”, tiek izsaukta funkcija 2.3.2.10.
Datu izvade
No datubāzes iegūtā informācija tiek attēlota pacienta informācijas apskates skatā.

2.3.2.8. Pacienta konta paroles maiņa

Mērķis
Šī funkcija ir paredzēta, lai pacients spētu nomainīt sava konta paroli.
Datu ievade
Funkcijā 2.3.2.8., pēc pogas “Change Password” nospiešanas, tiks parādīts paziņojuma dialogs ar sekojošiem ievades laukiem: <ul style="list-style-type: none">• Old password (Vecā parole) – obligāts teksta ievades lauks• New password (Jaunā parole) – obligāts teksta ievades lauks• Confirm password (apstiprināt paroli) – obligāts teksta ievades lauks

Datu apstrāde
Vecā parole tiks salīdzināta ar datubāzē esošo paroli. Ja tās sakrītīs, tad jaunās paroles tiks savstarpēji salīdzinātas, ja tās sakrītīs, tad datubāzē veco paroli dzēsīs un tās vietā ievadīs jauno.
Datu izvade
Vecās paroles nesakrītības gadījumā tiks parādīts kļūdas paziņojums. Šāda paša tipa kļūdas paziņojums tiks parādīts brīdī, ja nesakrīt abas jaunievadītās paroles. Pēc nekļūdīgas datu ievades, nospiežot pogu “OK”, tiks parādīts paziņojums par veiksmīgu paroles maiņu.

2.3.2.9. Pacienta konta dzēšana

Mērķis
Šī funkcija paredzēta, lai nodrošinātu pacienta konta dzēšanu.
Datu ievade
Funkcijā 2.3.2.8., pēc pogas “Delete Account” nospiešanas, tiks parādīts paziņojuma dialogs par to, vai pacients vēlas dzēst savu kontu.
Datu apstrāde
Pēc “Delete Account” pogas nospiešanas, tiks parādīts dialoga paziņojums ar vaicājumu par konta dzēšanu. Nospiežot tajā pogu “Yes”, pacienta konts tiks dzēsts no datubāzes. Nospiežot pogu “cancel” pacients tiks adresēts atpakaļ uz pacienta informācijas apskates skatu.
Datu izvade
Pēc pacienta konta dzēšanas tiks parādīts paziņojums par pacienta izdzēšanu no sistēmas.

2.3.3. Ārsta funkcijas

2.3.3.1. Ārsta reģistrācija

Mērķis
Šī funkcija ir paredzēta ārsta reģistrēšanai.
Datu ievade
Reģistrācijas formā ir jāaizpilda sekojoši lauki (ievada lietotājs): <ul style="list-style-type: none">• Name (Vārds) – obligāts ievades lauks• Surname (Uzvārds) – obligāts ievades lauks• E-mail (E-pasta adrese) – obligāts ievades lauks• Password (Parole) – obligāts ievades lauks• Password (Parole atkārtoti) – obligāts ievades lauks• Phone number (Telefona numurs) – obligāts ievades lauks
Datu apstrāde
Nospiežot pogu “Register” notiek sekojošas darbības: <ul style="list-style-type: none">• Lauku validācija – vai lauki nav atstāti tukši, e-pasts ievadīts pareizā formā, paroles abos ievades laukos sakrīt• Tiek pārbaudīts, vai lietotājs ar šādu e-pastu jau neeksistē datubāzē• Pēc veiksmīgas ievaddatu validācijas tie tiek saglabāti datubāzē Nospiežot pogu “Back” lietotājam tiek attēlots reģistrācijas vai pieslēgšanās izvēles skats.
Datu izvade
Ja kāds no ievadlaukiem ir aizpildīts nekorekti vai nav aizpildīts vispār, parādās kļūdas paziņojums tajā laukā ar attiecīgu paziņojumu (tukšs lauks vai validācija neatbilst kritērijiem – e-pasta un paroles gadījumos). Nesekmīgas reģistrēšanas gadījumā tiek parādīts paziņojums. Pēc veiksmīgas datu validācijas, parādās paziņojums par lietotāja reģistrāciju sistēmā. Pēc paziņojuma parādīšanās tiek attēlota sistēmas pieslēgšanās skats.

2.3.3.2. Ārsta autentifikācija sistēmā

Mērķis
Šī funkcija ir paredzēta ārsta autentifikācijai sistēmā. Autentifikācija notiek, izmantojot e-pasta adresi un paroli.
Datu ievade
Autentifikācijas formā ir divi datu ievades lauki: <ul style="list-style-type: none">• E-pasts – obligāts ievades lauks• Parole – obligāts ievades lauks
Datu apstrāde
Nospiežot pogu “Login”, tiek sūtīts pieprasījums uz datubāzi. Tiek pārbaudīts, vai ievadītā e-pasta adrese eksistē datubāzē. Apstiprinoša iznākuma gadījumā, ievadītajai parolei tiek salīdzināta ar datubāzē esošo.
Datu izvade
Neeksistējoša e-pasta ievades gadījumā parādās kļūdas paziņojums par to, ka ārsts ar šādu e-pasta adresi sistēmā nav reģistrēts. Ja ievadītā e-pasta adrese eksistē datubāzē, taču ievadītā parole nesakrīt ar datubāzē esošo, tad tiek parādīts kļūdas paziņojums par nepareizas paroles ievadi. Ja e-pasta adrese datubāzē eksistē un ievadītā parole sakrīt ar datubāzē esošo, tiek parādīts paziņojums par veiksmīgu pieslēgšanos sistēmai un tiek parādīts lietotnes galvenās aktivitātes galvenais skats, kur izvēlnē tiek attēloti ārsta izvēlnes elementi.

2.3.3.3. Ārsta navigācijas izvēlnes izveidošanas

Mērķis
Šī funkcija ir paredzēta, lai lietotājam attēlotu nepieciešamos izvēlnes elementus.
Datu ievade

Pēc pieslēgšanās tiek iegūta informācija par lietotāja tipu (šajā gadījumā atgriezts tips – ārsts). Papildus iegūst arī ārsta vārdu, uzvārdu un e-pasta adresi.
Datu apstrāde
Tiek izveidota navigācijas izvēlne, padarot redzamus ārstam paredzētos izvēlnes elementus (Add Patient, Patients, History, Precept un Profile)
Datu izvade
Iespējo augstāk minētos izvēles elementus un izvēles galvenē attēlo ārsta vārdu, uzvārdu un e-pasta adresi.

2.3.3.4. Pacienta pievienošana

Mērķis
Šī funkcija ir paredzēta ārstējamo pacientu pievienošanai.
Datu ievade
Pacienta e-pasta adrese.
Datu apstrāde
Ievadītā e-pasta adrese tiek salīdzināta ar datubāzē reģistrēto pacientu parolēm.
Datu izvade
Nospiežot pogu “Add”:
<ul style="list-style-type: none"> • Ja datubāzē šādas e-pasta adreses nav, tiek parādīts kļūdas paziņojums par šāda pacienta neesamību sistēmā. • Ja e-pasta adrese sakrīt, taču pacientam jau ir ārsts, tiek parādīts kļūdas paziņojums. • Ja pacienta e-pasta adrese datubāzē eksistē un tam ārsta nav, šim pacientam tik piešķirts ārsts un tiek parādīts paziņojums par veiksmīgi pievienotu pacientu.

2.3.3.5. Rehabilitācijas nosacījumu definēšana pacientam

Mērķis
Šī funkcija ir paredzēta, lai ārsts varētu definēt rehabilitācijas nosacījumus konkrētam pacientam.
Datu ievade
No lejupslīdošās izvēlnes tiek izvēlēts kāds no ārsta pacientiem. Teksta ievades laukos tiek ievadīta informācija par rehabilitācijas norādījumiem – optimālais ceļa salikšanas leņķis, maksimālais ceļa saliekšanas leņķis, rehabilitācijas sesijas ilgums un rehabilitācijas biežums (reizes nedēļā). Visi ievadlauki tiek ievadītas <i>int</i> tipa vērtības.
Datu apstrāde
Lejupslīdošajā izvēlnē tiek ielasīti dati no ārsta pacientiem. Izvēloties pacientu no saraksta ir iespēja tam pievienot rehabilitācijas norādījumus, nospiežot pogu “Assign”. Pēc pogas nospiešanas notiek lauku validācija – tiek pārbaudīts, vai tie nav atstāti tukši. Ja visi lauki ir aizpildīti, tiek pārbaudīts, vai pacientam jau nav pievienoti rehabilitācijas nosacījumi. Ja pacientam nav pievienoti rehabilitācijas nosacījumi, pēc pogas “Assign” nospiešanas, datubāzē tie tiek pievienoti izvēlētajam pacientam.
Datu izvade
Ja ārstam nav neviena pacienta, tiek parādīts paziņojums par pacientu neesamību. Mēģinot pievienot daļēji aizpildītus rehabilitācijas nosacījumus, katrā neaizpildītajā laukā parādās paziņojums, ka šis lauks ir obligāts. Ja visi lauki ir aizpildīti, pēc pogas “Assign” nospiešanas, tiek pārbaudīts, vai izvēlētajam pacientam jau nav pievienoti rehabilitācijas nosacījumi. Ja ir, tad tiek parādīts paziņojums par jau esošiem rehabilitācijas datiem. Pretējā gadījumā tie tiek pievienoti un tiek parādīts paziņojums par datu pievienošanu pacientam.

2.3.3.6. Rehabilitācijas nosacījumu dzēšana

Mērķis
Šī funkcija paredzēta, lai dzēstu rehabilitācijas nosacījumus pacientam.
Datu ievade
No lejupslīdošās izvēlnes tiek izvēlēts kāds no ārsta pacientiem.
Datu apstrāde
Lejupslīdošajā izvēlnē tiek ielasīti dati no ārsta pacientiem. Pēc pogas “Remove Precept” nospiešanas, parādās paziņojuma dialogs par to, vai tiešām ārsts vēlas dzēst rehabilitācijas nosacījumus pacientam. Apstiprinošas izvēles gadījumā no datubāzes attiecīgajam pacientam tiek dzēsti rehabilitācijas nosacījumi.
Datu izvade
Ja ārstam nav neviena pacienta, tiek parādīts paziņojums par pacientu neesamību. Pēc pogas “Remove Precept” nospiešanas, parādās paziņojuma dialogs par to, vai tiešām ārsts vēlas dzēst rehabilitācijas nosacījumus pacientam. Nospiežot pogu “Yes”, tiek parādīts ziņojums par rehabilitācijas nosacījumu dzēšanu pacientam. Pretējā gadījumā, nospiežot pogu “Cancel”, tiek parādīts pacienta rehabilitācijas nosacījumu pievienošanas skats.

2.3.3.7. Pacientu sesiju vēstures attēlošana

Mērķis
Šī funkcija paredzēta pacientu rehabilitācijas sesiju vēstures attēlošanai.
Datu ievade
No lejupslīdošās izvēlnes tiek izvēlēts kāds no ārsta pacientiem. Datubāzei tiek veikts pieprasījums, vaicājot pēc atlasītā pacienta visiem rehabilitācijas sesiju datiem: <ul style="list-style-type: none">• Rehabilitācijas veikšanas datums un laiks• Rehabilitācijas sesijas ilgums (minūtes un sekundes)

- Maksimālā leņķa pārsniegšanas skaita reizes
- Rehabilitācijas sesijas komentārs

Datu apstrāde

Lejupslīdošajā izvēlnē tiek ielasīti dati no ārsta pacientiem. Pēc pacienta atlasīšanas un pogas “Show” nospiešanas, tiek izveidots saraksts ar atlasītā pacienta visu rehabilitācijas sesiju vēsturi.

Datu izvade

Ja pacientam nav nevienas rehabilitācijas sesijas, tiek parādīts paziņojums ar atbilstošu informāciju. Pretējā gadījumā zem lejupslīdošās izvēlnes parādās visa izvēlētā lietotāja rehabilitācijas sesiju vēsture.

2.3.3.8. Ārsta konta informācijas attēlošana

Mērķis

Šī funkcija paredzēta, lai attēlotu ārsta pamatinformāciju. Un iespējotu funkciju 2.3.3.9. (ārsta paroles maiņa) un 2.3.3.10.(ārsta konta dzēšana) izsaukšanu.

Datu ievade

Datubāzei tiek veikts pieprasījums, vaicājot pēc visiem rehabilitācijas sesiju datiem:

- Ārsta vārds un uzvārds
- Ārsta e-pasts
- Ārsta telefona numurs

Datu apstrāde

No datubāzes tiek iegūti iepriekšminētie dati un ievadīti sagataves formā. Nospiežot pogu “Change Password”, tiek parādīts paziņojuma dialogs un izsaukta funkcija 2.3.2.9. Nospiežot pogu “Delete Account”, tiek izsaukta funkcija 2.3.2.10.

Datu izvade

No datubāzes iegūtā informācija tiek attēlota ārsta konta apskates skatā.

2.3.3.9. Ārsta konta paroles maiņa

Mērķis
Šī funkcija ir paredzēta, lai ārsts spētu nomainīt sava konta paroli.
Datu ievade
Funkcijā 2.3.2.8., pēc pogas “Change Password” nospiešanas, tiks parādīts paziņojuma dialogs ar sekojošiem ievades laukiem: <ul style="list-style-type: none">• Old password (Vecā parole) – obligāts ievades lauks• New password (Jaunā parole) – obligāts ievades lauks• Confirm password (apstiprināt paroli) – obligāts ievades lauks
Datu apstrāde
Vecā parole tiks salīdzināta ar datubāzē esošo paroli. Ja tās sakrītīs, tad jaunās paroles tiks savstarpēji salīdzinātas, ja tās sakrītīs, tad datubāzē veco paroli dzēsīs un tās vietā ievadīs jauno.
Datu izvade
Vecās paroles nesakrītības gadījumā tiks parādīts kļūdas paziņojums. Šāda paša veida kļūdas paziņojums tiks parādīts brīdī, ja nesakrīt abas jaunievadītās paroles. Pēc nekļūdīgas datu ievades, nospiežot pogu “OK”, tiks parādīts paziņojums par veiksmīgu paroles maiņu.

2.3.3.10. Ārsta konta dzēšana

Mērķis
Šī funkcija paredzēta, lai nodrošinātu pacienta konta dzēšanu.
Datu ievade
Funkcijā 2.3.2.8., pēc pogas “Delete Account” nospiešanas, tiks parādīts paziņojuma dialogs par to, vai pacients vēlas dzēst savu kontu.
Datu apstrāde

Pēc "Delete Account" pogas nospiešanas, tiks parādīts paziņojuma ar vaicājumu par konta dzēšanu. Nospiežot tajā pogu "Yes", pacienta konts tiks dzēsts no datubāzes. Nospiežot pogu "cancel" pacients tiks adresēts atpakaļ uz ārsta konta apskates skatu.
Datu izvade
Pēc pacienta konta dzēšanas tiks parādīts paziņojums par pacienta izdzēšanu no sistēmas.

2.3.3.11. Pievienoto pacientu attēlošana sarakstā

Mērķis
Šī funkcija paredzēta, lai attēlotu ārsta pievienotos pacientus sarakstā
Datu ievade
No datubāzes tiek iegūta informācija par visiem ārsta pacientiem.
Datu apstrāde
Visi ārsta pacienti tiek saglabāti sarakstā.
Datu izvade
Pacienti tiek attēloti saraksta veidā, alfabētiskā secībā pēc to e-pasta adresēm.

2.4. Nefunkcionālās prasības

2.4.1. Veiktspējas prasības

Sistēmai ir jāspēj nodrošināt optimāla funkcionalitāte 100 lietotājiem, kas vienlaicīgi pieslēgušies un darbojas sistēmā. Tā izpaužas ar atbildes sniegšanu no pieprasījuma datubāzei ne ilgāk kā 2.5 sekunžu laikā pieslēdzoties sistēmai un 1.5 sekunžu laikā, kad lietotājs jau ir pieslēdzies sistēmai.

2.4.2. Drošības prasības

Sistēmas reģistrācijas un autorizēšanās procesā jāizmanto paroli šifrēšana. Tai jāizmanto Google Guava bibliotēka un SHA1 jaucējadrešēšanas algoritms. Jāspēj nodrošināt, ka strādāt ar sistēmu drīkst tikai reģistrēti un autentificēti lietotāji.

2.4.3. Pieejamības prasības

Sistēmai ir jābūt pieejamai visu cauru gadu, neatkarīgi no svētku dienām vai brīvdienām, 24 stundas diennaktī, 7 dienas nedēļā. Sistēmai jānodrošina stabila darbība bez, vai ar minimāliem (8 stundas viena mēneša laikā) pārtraukumiem sistēmas uzlabošanas nolūkos.

2.4.4. Aparatūras ierobežojumi

Mobilai lietotnei ir jābūt Android operētājsistēmai un tās versijai ir jābūt vienānai vai jaunākai par Android 4.0 (*Ice Cream Sandwich*). Android ierīcei jābūt interneta savienojumam un Bluetooth modulim. Pacientam ir nepieciešama pieeja speciāli izstrādātai iegultai ierīcei, kas spēj vākt ceļu locītavas kustību datus no sensoru mezgliem.

2.4.5. Uzturamības prasības

Sistēmas pirmkodam ir jābūt komentētam, ir jāglabā iepriekšējās sistēmas versijas, izmantojot versiju kontroli.

3. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

3.1. Ievads

3.1.1. Nolūks

Programmas projektējuma apraksts (PPA) paredzēts Android lietotnes ceļu rehabilitācijai projektējuma aprakstīšanai un programmatūras prasību specifikācijas izvirzīto prasību realizēšanai. Tas ir paredzēts izstrādātājiem, atvieglojot projekta funkciju implementēšanu, projekta analīzi un lēmumu pieņemšanu tā laikā.

3.2. Dekompozīcijas apraksts

3.2.1. Ievads

Par projekta pamatstruktūru tika izvēlēta Android standarta lietotnes struktūra. Visas lietotāja saskarnes klases mantos *AppCompatActivity* vai *Fragment* klases. Personalizētiem sarakstiem tiks veidotas sarakstu adapteru klases, mantojot *BaseFragment*.

3.2.2. Moduļu dekompozīcija

3.2.2.1. Klase *HistoryAdapter*

Klase paredzēta pacientu rehabilitācijas sesiju vēstures saraksta aizpildīšanai.

Metodes:

Tips	Metode	Apraksts
int	getCount()	Pārrakstīta metode no mantotās klases <i>BaseAdapter</i> . Atgriež saraksta garumu.
Object	getItem(int position)	Pārrakstīta metode no mantotās klases <i>BaseAdapter</i> . Atgriež saraksta elementu konkrētā pozīcijā.
Long	getItemId(int position)	Pārrakstīta metode no mantotās klases <i>BaseAdapter</i> . Atgriež saraksta elementa ID.

View	getView(int position, View convertView, ViewGroup parent)	Pārrakstīta metode no mantotās klases BaseAdapter. Atgriež saraksta elementa skatu ar aizpildītām vērtībām.
------	---	---

Mainīgie:

Tips	Mainīgais	Apraksts
Context	mContext	Lietotnes konteksts
List<RehabSession>	mHistoryList	Vēstures saraksts

3.2.2.2. Klase Doctor

Klase ārsta objektu izveidošanai.

Metodes:

Klase satur visu tās mainīgo uzstādītājmetodes un saņēmējmetodes.

Mainīgie:

Tips	Mainīgais	Apraksts
String	Name	Vārds
String	Surname	Uzvārds
String	Email	E-pasta adrese
String	Password	Parole
String	Phone_number	Telefona numurs

3.2.2.3. Klase Patient

Klase pacienta objektu izveidošanai.

Metodes:

Klase satur visu tās mainīgo uzstādītājmetodes un saņēmējmetodes.

Mainīgie:

Tips	Mainīgais	Apraksts
String	Name	Vārds
String	Surname	Uzvārds
String	Email	E-pasta adrese
String	Password	Parole
String	Gender	Dzimums
String	Phone_number	Telefona numurs
String	Doctor_key	Ārsta identifikācijas atslēga

3.2.2.4. Klase Precept

Klase rehabilitācijas nosacījumu objektu izveidošanai.

Metodes:

Klase satur visu tās mainīgo uzstādītājmetodes un saņēmējmetodes.

Mainīgie:

Tips	Mainīgais	Apraksts
Int	Optimal_angle	Optimālais ceļa saliekšanas leņķis
Int	Maximal_angle	Maksimālais ceļa saliekšanas leņķis
Int	Duration	Rehabilitācijas sesijas ilgums minūtēs
Int	Frequency	Rehabilitācijas veikšanas biežums (reizes nedēļā)

3.2.2.5. Klase RehabSession

Klase rehabilitācijas sesijas objektu izveidošanai.

Metodes:

Klase satur visu tās mainīgo uzstādītājmetodes un saņēmējmetodes.

Mainīgie:

Tips	Mainīgais	Apraksts
String	Date	Datums un laiks, kurā pacients ir veicis rehabilitāciju
String	Duration	Ilgums minūtēs un sekundēs, ko pacients ir pavadījis veicot rehabilitāciju
Int	Exceeded_max	Skaitis, cik reizes pacients ir pārsniedzis maksimāli atļauto ceļa saliekšanas leņķi
String	Comment	Komentārs, ko pacients pievieno pēc rehabilitācijas veikšanas

3.2.2.6. Klase UserPickActivity

Klase lietotāja tipa izvēlei. Pēc izvēles veikšanas lietotājam tiek parādīts Pieslēgšanās un reģistrācijas skats.

Metodes:

Tips	Metode	Apraksts
Void	StartPatient(View view)	SharedPreferences tiek saglabāta informācija par lietotāja tipu (pacients)
Void	StartDoctor(View view)	SharedPreferences tiek saglabāta informācija par lietotāja tipu (ārsts)

3.2.2.7. Klase StartActivity

Klase lai veiktu izvēli – pieslēgties vai reģistrēties sistēmā. Pēc izvēles veikšanas lietotājam tiek parādīts pieslēgšanās vai reģistrēšanās skats.

Metodes:

Tips	Metode	Apraksts
Void	StartLogin(View view)	Tiek attēlots sistēmas pieslēgšanās skats
Void	StartRegister(View view)	Tiek attēlots sistēmas reģistrācijas skats

3.2.2.8. Klase UserRegisterTask

Klase lietotāja reģistrēšanai sistēmā.

Metodes:

Tips	Metode	Apraksts
Boolean	userExists(Map<String,Object>pData)	Metode pārbauda, vai sistēma eksistē lietotājs
Void	addPatient(Firebase firebase)	Metode pacienta reģistrēšanai sistēmā
Void	addDoctor(Firebase firebase)	Metode ārsta reģistrēšanai sistēmā
Void	emailExistsError()	Metode kļūdas paziņojumam par sistēmā jau esošu e-pasta adresi

Mainīgie:

Tips	Mainīgais	Apraksts
String	mName	Vārds
String	mSurname	Uzvārds
String	mEmail	E-pasta adrese
String	mPassword	Parole
String	mGender	Dzimums
String	mPhoneNumber	Telefons numurs
ValueEventListener	patientListener	Pacienta notikumu uztvērējs
ValueEventListener	doctorListener	Ārsta notikumu uztvērējs
Firebase	patientFirebase	Pacienta <i>Firebase</i> datubāzes mainīgais
Firebase	doctorFirebase	Ārsta <i>Firebase</i> datubāzes mainīgais

3.2.2.9. Klase RegisterActivity

Klase, lai veiktu lietotāja reģistrācijas validāciju

Metodes:

Tips	Metode	Apraksts
Void	attemptRegister()	Metode veic lauku validāciju, sekmīga rezultāta gadījuma reģistrējot lietotāju.
Void	showProgress(final boolean show)	Metode, lai parādītu progresu animāciju reģistrācijas brīdī
Boolean	isEmailValid(String email)	Metode veic e-pasta adreses validāciju
Boolean	isPasswordValid(String password)	Metode veic paroles validāciju
Void	RegisterBack(View view)	Metode lietotāju atgriež pie iepriekšējā reģistrācijas-pieslēgšanās skata

Mainīgie:

Tips	Mainīgais	Apraksts
UserRegisterTask	mAuthTask	Lietotāja reģistrācijas mainīgais
EditText	mNameView	Vārds
EditText	mSurnameView	Uzvārds
EditText	mPasswordView	Parole
EditText	mConfirmPasswordView	Atkārota parole tās apstiprināšanai
EditText	mPhoneNumberView	Telefona numurs
EditText	mEmailView	E-pasta adrese
View	mProgressView	Progresu skats
View	mRegisterFormView	Reģistrācijas skats
Spinner	mGenderView	Lietotāja dzimums

3.2.2.10. Klase UserLoginTask

Klase, lai pieslēgtu lietotāju sistēmai.

Metodes:

Tips	Metode	Apraksts
Boolean	userExists(Map<String, Object> pData)	Pārbauda, vai lietotājs ar šādu e-pasta adresi sistēmā ir reģistrējies.
Void	tryPatientLogin(String password, String email)	Pārbauda, vai ievadīta parole sakrīt ar datubāzē esošo. Veiksmīga iznākuma gadījumā pieslēdz pacientu sistēmai, pretēji – parāda kļūdas paziņojumu.
Void	tryDoctorLogin(String password, String email)	Pārbauda, vai ievadīta parole sakrīt ar datubāzē esošo. Veiksmīga iznākuma gadījumā pieslēdz ārstu sistēmai, pretēji – parāda kļūdas paziņojumu.
Void	SharedPatient(Map<String, Object> map)	Metode saglabā pacienta datus lietotnes koplietošanas atribūtos
Void	SharedDoctor(Map<String, Object> map)	Metode saglabā ārsta datus lietotnes koplietošanas atribūtos
Void	loginFailed()	Metode kļūdas paziņošanai par nesekmīgu pieslēgšanos sistēmai
Void	noUserError()	Metode kļūdas paziņošanai par e-pasta adreses neesamību datubāzē.

Mainīgie:

Tips	Mainīgais	Apraksts
String	mEmail	E-pasta adrese
String	mPassword	Parole
ValueEventListener	patientListener	Pacienta notikumu uztvērējs
ValueEventListener	doctorListener	Ārsta notikumu uztvērējs
Firebase	patientFirebase	Pacienta <i>Firebase</i> datubāzes mainīgais
Firebase	doctorFirebase	Ārsta <i>Firebase</i> datubāzes mainīgais

SharedPreferences	sharedPreferences	Lietotnes koplietošanas atribūti
String	User_type	Lietotāja tips

3.2.2.11. Klase LoginActivity

Klase, lai veiktu lietotāja pieslēgšanās validāciju

Metodes:

Tips	Metode	Apraksts
Void	attemptLogin()	Metode veic lauku validāciju, sekmīga rezultāta gadījuma pieslēdzot lietotāju sistēmai.
Void	showProgress(final boolean show)	Metode, lai parādītu progresa animāciju sistēmas pieslēgšanās brīdī
Void	LoginBack(View view)	Metode lietotāju atgriež pie iepriekšējā reģistrācijas-pieslēgšanās skata

Mainīgie:

Tips	Mainīgais	Apraksts
UserLoginTask	mAuthTask	Lietotāja pieslēgšanās mainīgais
EditText	mEmailView	E-pasta adrese
EditText	mPasswordView	Parole
View	mProgressView	Progresa skats
View	mLoginFormView	Pieslēgšanās skats

3.2.2.12. Klase DrawerActivity

Klase Paredzēta, lai komunicētu ar visiem lietotnes fragmentiem un izpildīto to funkcijas.

Metode implementē šādus interfeisus:

NavigationView.OnNavigationItemSelectedListener,

AddPatientFragment.addPatientListener,

PatientListFragment.patientListListener,

PreceptAssignFragment.preceptAssignListener,
 PreceptFragment.preceptListener,
 RehabFragment.rehabListener,
 HistoryPatientFragment.patientHistoryListener,
 DoctorHistoryFragment.DoctorHistoryListener,
 PatientViewFragment.patientViewListener,
 DoctorViewFragment.doctorViewListener

Metodes:

Tips	Metode	Apraksts
Void	onCreate(Bundle savedInstanceState)	Pārrakstīta metode no mantotās klases <i>AppCompatActivity</i> . Atkarībā no lietotāja tipa ielādē izvēlnes fragmentus un iespējo attiecīgos izvēlnes elementus.
Void	setText(String fullName, String email)	Metode, kas iestata navigācijas izvēlnes galvenes datus.
Void	onBackPressed()	Pārrakstīta metode no mantotās klases <i>AppCompatActivity</i> . Tā izveido vaicājuma dialogu par atslēgšanos no sistēmas. Pozitīvas izvēles gadījumā lietotājs tiek atslēgts no sistēmas.
Boolean	onNavigationItemSelectedListener(MenuItem item)	Pārrakstīta metode no implementētā interfeisa <i>NavigationView</i> . <i>OnNavigationItemSelectedListener</i> , nodrošina navigācijas izvēlnes elementu izvēles gadījumā attiecīgo fragmentu parādīšanu
Void	createPatientList()	Metode, lai izveidotu pacientu sarakstu ārstam, kas pieslēdzies sistēmai.
ArrayList<String>	getFullPatientList(ArrayList<String> arrayList)	Metode, lai izveidotu visu datubāzē esošo pacientu sarakstu

String	getPatientFullName(Map<String, Object> map)	Metode, lai iegūtu pacienta vārdu un uzvārdu
Boolean	isPatient(String email)	Pārrakstīta metode no implementētā interfeisa <i>AddPatientFragment</i> . <i>addPatientListener</i> , kas pārbauda, vai pacients eksistē datubāzē
Boolean	checkPatient(String email)	Metode pārbauda, vai pacients jau ir pievienots kādam ārstam
Void	addPatient(String email)	Metode pacienta pievienošanai ārstam
Boolean	Exists(DataSnapshot dataSnapshot)	Metode, kas pārbauda, vai datubāzes vaicājums neatgriež null vērtību
Void	patientAlreadyAddedError()	Metode, kas parāda <i>Toast</i> tipa paziņojumu par to, ka pacients jau pievienots kādam ārstam
Void	patientDoesNotExistError()	Metode, kas parāda <i>Toast</i> tipa paziņojumu par to, ka pacients datubāzē neeksistē
Void	patientAdded()	Metode, kas parāda <i>Toast</i> tipa paziņojumu par to, ka pacients ir veiksmīgi pievienots ārstam
Void	populateList(ListView listView)	Pārrakstīta metode no implementētā interfeisa <i>PatientListFragment</i> . <i>patientListListener</i> ārsta pacientu saraksta aizpildīšanai.
Void	assignPrecept(String email, int optimal_angle, int maximal_angle, int duration, int frequency)	Pārrakstīta metode no implementētā interfeisa <i>PreceptAssignFragment</i> . <i>preceptAssignListener</i> rehabilitācijas nosacījumu pievienošanai konkrētam pacientam
Void	removeAssignment(final String patient_key, Context context)	Pārrakstīta metode no implementētā interfeisa <i>PreceptAssignFragment</i> . <i>preceptAssignListener</i> rehabilitācijas

		nosacījumu noņemšanai konkrētam pacientam
Void	populateSpinner(Spinner spinner)	Pārrakstīta metode no implementētā interfeisa <i>PreceptAssignFragment</i> . <i>preceptAssignListener</i> rehabilitācijas nosacījumu pacientu lejupslīdošās izvēlnes aizpildīšanai
Void	preceptAdded()	Metode, kas parāda <i>Toast</i> tipa paziņojumu par to, ka rehabilitācijas nosacījumi konkrētam pacientam ir pievienoti
Void	preceptExistsError()	Metode, kas parāda <i>Toast</i> tipa paziņojumu par to, ka rehabilitācijas nosacījumi konkrētam pacientam jau ir pievienoti
Boolean	isPrecept(final View view)	Pārrakstīta metode no implementētā interfeisa <i>PreceptFragment</i> . <i>preceptListener</i> kas pārbauda, vai rehabilitācijas nosacījumi eksistē
Void	populatePreceptDoctor(Map<String, Long< map, View view)	Metode, lai aizpildītu pacienta rehabilitācijas nosacījumu laukus “Ārsta vārds”, “Ārsta uzvārds” un “Ārsta telefona numurs”
Void	populatePreceptInfo(Map<String, Long< map, View view)	Metode, lai aizpildītu pacienta rehabilitācijas nosacījumu laukus “Rehabilitācijas ilgums”, “Rehabilitācijas biežums”, “Maksimālais ceļa saliekšanas leņķis” un “Optimālais ceļa saliekšanas leņķis”
Void	populatePreceptError()	Metode, kas parāda <i>Toast</i> tipa paziņojumu par to, ka rehabilitācijas nosacījumus ārsts nav pievienojis

Void	noDoctorError()	Metode, kas parāda <i>Toast</i> tipa paziņojumu par to, ka pacientam nav pievienots rehabilitāciju vadošais ārsts
Void	Timer(final View view)	Pārrakstīta metode no implementētā interfeisa <i>RehabFragment</i> . <i>rehabListener</i> pacienta rehabilitācijas fragmenta datu apstrādei
Void	startTimer(long millis, final TextView timerText)	Metode ritošā laika attēlošanai taimerī
Void	finishRehab(Context context)	Pārrakstīta metode no implementētā interfeisa <i>RehabFragment</i> . <i>rehabListener</i> pacienta rehabilitācijas pārtaukšanai un datu nosūtīšanai uz datubāzi
Void	populateHistoryList(ListView listView)	Pārrakstīta metode no implementētā interfeisa <i>HistoryPatientFragment</i> . <i>patientHistoryListener</i> , kas aizpilda vēstures fragmentu ar pacienta rehabilitācijas vēstures datiem
Void	addSessionToHistory(Map<String, Object> map)	Metode, kas pievieno rehabilitācijas sesiju pacienta rehabilitācijas sesiju vēsturē un datubāzē
Void	populateDoctorHistoryList(ListView listview, String patient_key)	Pārrakstīta metode no implementētā interfeisa <i>DoctorHistoryFragment</i> . <i>DoctorHistoryListener</i> , kas aizpilda ārsta vēstures fragmentu ar konkrēta pacienta rehabilitācijas sesiju vēstures datiem
Void	populateHistorySpinner(Spinner spinner)	Pārrakstīta metode no implementētā interfeisa <i>DoctorHistoryFragment</i> . <i>DoctorHistoryListener</i> , kas aizpilda ārsta vēstures fragmentu pacientu

		lejupslīdošās izvēlni ar konkrētā ārsta pacientiem
Void	deletePatientAccount(Context context)	Pārrakstīta metode no implementētā interfeisa <i>PatientViewFragment</i> . <i>patientViewListener</i> , kas izdzēš pacienta kontu no datubāzes
Void	changePatientPassword(Context context)	Pārrakstīta metode no implementētā interfeisa <i>PatientViewFragment</i> . <i>patientViewListener</i> , kas nomaina pacienta paroli datubāzē
Void	populatePatientView(View view)	Pārrakstīta metode no implementētā interfeisa <i>PatientViewFragment</i> . <i>patientViewListener</i> , kas aizpilda pacienta profila fragmentu ar pacienta informāciju.
Void	deleteDoctorAccount(Context context)	Pārrakstīta metode no implementētā interfeisa <i>DoctorViewFragment</i> . <i>doctorViewListener</i> , kas izdzēš ārsta kontu no datubāzes
Void	changeDoctorPassword(Context context)	Pārrakstīta metode no implementētā interfeisa <i>DoctorViewFragment</i> . <i>doctorViewListener</i> , kas nomaina ārsta paroli datubāzē
Void	populateDoctorView(View view)	Pārrakstīta metode no implementētā interfeisa <i>DoctorViewFragment</i> . <i>doctorViewListener</i> , kas aizpilda ārsta profila fragmentu ar ārsta informāciju.

Mainīgie:

Tips	Mainīgais	Apraksts
NavigationView	navigationView	Navigācijas izvēlnes skats
Toolvar	toolbar	Izvēlnes rīkjoslas mainīgais

String	Email	E-pasta adrese
String	Name	Vārds
String	Surname	Uzvārds
String	Gender	Dzimums
String	Phone	Telefona numurs
String	addPatientEmail	Pievienojamā pacienta e-pasta adrese
String	Doctor_key	Ārsta identifikācijas atslēga
Int	Duration_minutes	Rehabilitācijas ilgums minūtēs
ArrayList<string>	allPatients	Saraksts ar visiem pacientiem
Firebase	addPatientFierbase	<i>Firebase</i> datubāzes mainīgais pacientu pievienošanai
ValueEventListener	mListener	Datubāzes notikumu uztvērējs
ChildEventListener	cListener	Datubāzes apakšnodaļu notikumu uztvērējs
Firebase	childFirebase	<i>Firebase</i> datubāzes mainīgais datubāzes apakšnodaļu pievienošanai
ArrayList<String>	doctorPatients	Saraksts ar visiem ārsta pacientiem
Firebase	preceptFirebase	Rehabilitācijas nosacījumu <i>Firebase</i> datubāzes mainīgais
ValueEventListener	preceptListener	Rehabilitācijas nosacījumu datubāzes notikumu uztvērējs
CountDownTimer	countDownTimer	Laika atskaites mainīgais Rehabilitācijas sesijai
Long	millisLeft	Rehabilitācijas sesijas atlikušais laiks milisekundēs
String	Rehab_duration	Rehabilitācijas ilgums formātā MM min, SS sec
Long	Full_duration_millis	Pilnais rehabilitācijas ilgums milisekundēs, ko noteicis ārsts
String	Rehab_comment	Rehabilitācijas komentārs
HistroyListAdapter	historyAdapter	Sesiju vēstures saraksta adapteris
List<RehabSession>	patientHistoryList	Pacienta sesiju vēstures saraksts

Firestore	changePatientFirestore	Pacienta paroles maiņas <i>Firestore</i> datubāzes mainīgais
ValueEventListener	changePatientListener	Pacienta paroles maiņas datubāzes notikumu uztvērējs
Firestore	changeDoctorFirestore	Ārsta paroles maiņas <i>Firestore</i> datubāzes mainīgais
ValueEventListener	changeDoctorListener	Ārsta paroles maiņas datubāzes notikumu uztvērējs

Tālāk aprakstītas klases, kas implementē virsklasi “Fragment”. Katrai no tām ir savs interfeiss, kurā ir definēti metožu prototipi. To realizācijas tiek veiktas DrawerActivity klasē.

3.2.2.13. Klase AddPatientFragment

Klase pacientu pievienošanai ārstam. Tā implementē interfeisu *View.OnClickListener*.

Interfeisa *addPatientListener* metodes:

Tips	Metode	Apraksts
Boolean	isPatient(String email)	Metodes prototips

Interfeisa *View.OnClickListener* pārrakstītās metodes:

Tips	Metode	Apraksts
Void	OnClick(final View view)	Metode, kas tiek izsaukta pacienta pievienošanas pogas nospiešanas brīdī

Mainīgie:

Tips	Mainīgais	Apraksts
addPatientListener	mCommunicator	Interfeisa komunikatora mainīgais

3.2.2.14. Klase DoctoryHistoryFragment

Klase pacientu rehabilitācijas sesiju attēlošanai ārstam. Tā implementē interfeisu *View.OnClickListener*.

Interfeisa *doctorHistoryListener* metodes:

Tips	Metode	Apraksts
Void	populateDoctorHistoryList(ListView listView, String patient_key)	Metodes prototips
Void	populateHistorySpinner(Spinner spinner)	Metodes prototips

Interfeisa *View.OnClickListener* pārrakstītās metodes:

Tips	Metode	Apraksts
Void	OnClick(final View view)	Metode, kas tiek izsaukta pacienta rehabilitācijas saraksta parādīšanas pogas nospiešanas brīdī

Mainīgie:

Tips	Mainīgais	Apraksts
doctorHistoryListener	mCommunicator	Interfeisa komunikatora mainīgais

3.2.2.15. Klase DoctorViewFragment

Klase ārsta konta informācijas attēlošanai, konta dzēšanai un paroles maiņai. Tā implementē interfeisu *View.OnClickListener*.

Interfeisa *doctorViewListener* metodes:

Tips	Metode	Apraksts
Void	deleteDoctorAccount(Context context)	Metodes prototips
Void	changeDoctorPassword(Context context)	Metodes prototips
Void	populateDoctorView(View view)	Metodes prototips

Interfeisa *View.OnClickListener* pārrakstītās metodes:

Tips	Metode	Apraksts
Void	OnClick(final View view)	Metode, kas tiek izsaukta brīdī, kad ārsts nospiež kādu no pogām – “dzēst kontu” vai “mainīt paroli”

Mainīgie:

Tips	Mainīgais	Apraksts
doctorViewListener	mCommunicator	Interfeisa komunikatora mainīgais

3.2.2.16. Klase *HistoryPatientFragment*

Klase ārsta rehabilitācijas sesiju vēstures attēlošanai.

Interfeisa *patientHistoryListener* metodes:

Tips	Metode	Apraksts
Void	populateHistoryList(List View listView)	Metodes prototips

Mainīgie:

Tips	Mainīgais	Apraksts
patientHistoryListener	mCommunicator	Interfeisa komunikatora mainīgais

3.2.2.17. Klase PatientListFragment

Klase pacientu saraksta attēlošanai ārstam.

Interfeisa *patientListListener* metodes:

Tips	Metode	Apraksts
Void	populateList(ListView listView)	Metodes prototips

Mainīgie:

Tips	Mainīgais	Apraksts
patientListListener	mCommunicator	Interfeisa komunikatora mainīgais

3.2.2.18. Klase PatientViewFragment

Klase pacienta konta informācijas attēlošanai, konta dzēšanai un paroles maiņai. Tā implementē interfeisu *View.OnClickListener*.

Interfeisa *patientViewListener* metodes:

Tips	Metode	Apraksts
Void	deletePatientAccount(Context context)	Metodes prototips
Void	changePatientPassword(Context context)	Metodes prototips
Void	populatePatientView(View view)	Metodes prototips

Interfeisa *View.OnClickListener* pārrakstītās metodes:

Tips	Metode	Apraksts
Void	OnClick(final View view)	Metode, kas tiek izsaukta brīdī, kad pacients nospiež kādu no pogām – “dzēst kontu” vai “mainīt paroli”

Mainīgie:

Tips	Mainīgais	Apraksts
patientViewListener	mCommunicator	Interfeisa komunikatora mainīgais

3.2.2.19. Klase PreceptAssignFragment

Klase rehabilitācijas nosacījumu pievienošanai pacientam. Tā implementē interfeisu *View.OnClickListener*.

Interfeisa *preceptAssignListener* metodes:

Tips	Metode	Apraksts
Void	populateSpinner(Spinner spinner)	Metodes prototips
Void	assignPrecept(String patient_key, int optimal_angle, int maximal_angle, int duration, int frequency)	Metodes prototips
Void	populateSpinner(Spinner spinner)	Metodes prototips

Interfeisa *View.OnClickListener* pārrakstītās metodes:

Tips	Metode	Apraksts
Void	OnClick(final View view)	Metode, kas tiek izsaukta brīdī, kad tiek nospiesta kāda no pogām - “pievienot

		rehabilitācijas nosacījumus” vai “dzēst rehabilitācijas nosacījumus”
--	--	--

Mainīgie:

Tips	Mainīgais	Apraksts
preceptAssignListener	mCommunicator	Interfeisa komunikatora mainīgais

3.2.2.20. Klase PreceptFragment

Klase pacientu rehabilitācijas nosacījumu attēlošanai.

Interfeisa *preceptListener* metodes:

Tips	Metode	Apraksts
Boolean	isPrecept(View view)	Metodes prototips

Mainīgie:

Tips	Mainīgais	Apraksts
preceptListener	mCommunicator	Interfeisa komunikatora mainīgais

3.2.2.21. Klase RehabFragment

Klase rehabilitācijas veikšanai. Tā implementē interfeisu *View.OnClickListener*.

Interfeisa *rehabListener* metodes:

Tips	Metode	Apraksts
Void	Timer(View view)	Metodes prototips
Void	finishRehab(Context context)	Metodes prototips

Interfeisa *View.OnClickListener* pārrakstītās metodes:

Tips	Metode	Apraksts
Void	OnClick(final View view)	Metode, kas tiek izsaukta brīdī, kad pacients nospiež pogu “Beigt rehabilitācijas sesiju”

Mainīgie:

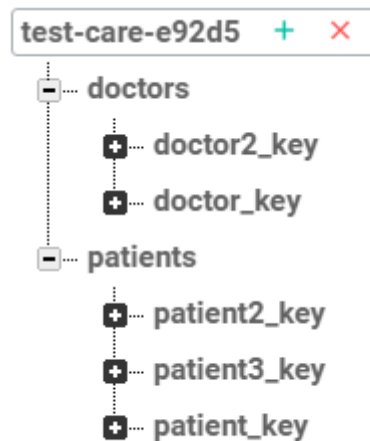
Tips	Mainīgais	Apraksts
rehabListener	mCommunicator	Interfeisa komunikatora mainīgais

3.3. Firebase datu bāzes apraksts

Projektā izmantota reāllaika mākoņa datubāze “*Firebase*”. Tā ir JSON Objektu koka veida datubāze. Dati tajā tiek glabāti kā objekti. Tā kā datubāze ir koka veida, tā tiek aprakstīta līmeņu veidā.

3.3.1. Pirmais līmenis – pacienti un ārsti

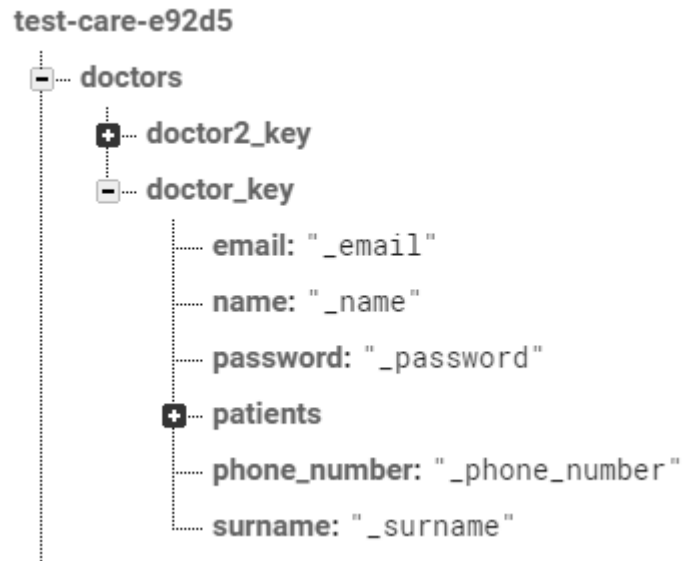
Pirmajā datubāzes līmenī būs divas ierakstu kopas – pacienti un ārsti, kur katra no tām glabās iepriekšminēto lietotāju grupu objektus. Katru no objektiem varēs identificēt pēc unikālas atslēgas.



3.3.1.1att. Pirmais līmenis – pacienti un ārsti

3.3.2. Otrais līmenis – ārsti

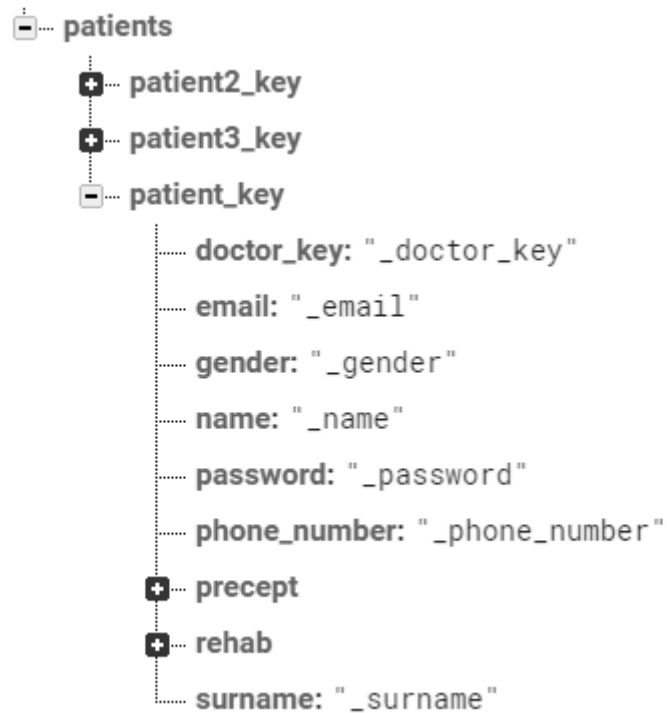
Otrajā datubāzes līmenī būs informācija par ārstiem un pacientiem. Ārsta ierakstu kopā glabāsies sekojoša informācija: vārds, uzvārds, e-pasta adrese, parole, telefona numurs un pacientu ierakstu kopa.



3.3.2.1.att. otrais līmenis - ārsti

3.3.3. Otrais līmenis – pacienti

Pacienta ierakstu kopā glabāsies sekojoša informācija: vārds, uzvārds, dzimums, e-pasta adrese, parole, telefona numurs, rehabilitācijas nosacījumu ierakstu kopa, rehabilitācijas sesiju ierakstu kopa un ārsta identifikācijas atslēga.



3.3.3.1.att. otrais līmenis - pacienti

3.3.4. Trešais līmenis – ārsta pacienti

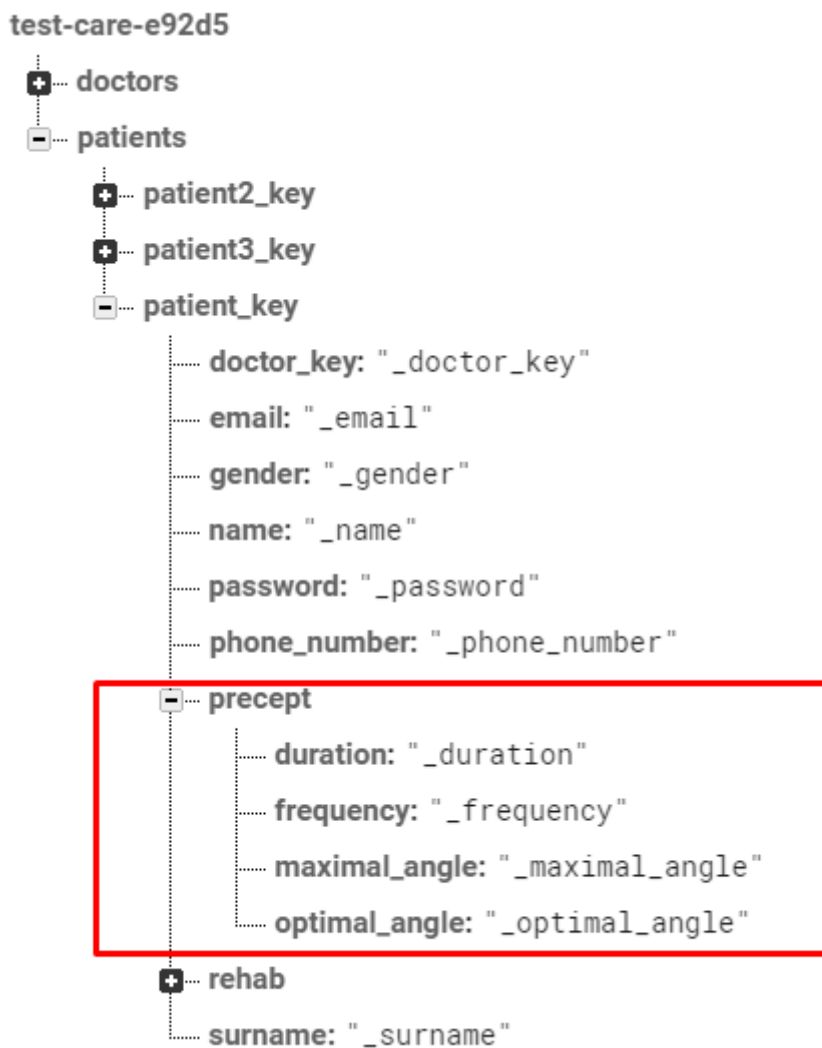
Trešajā līmenī ārsta ierakstu kopā glabāsies visu ārsta pacientu identifikācijas atslēgu kopa.



3.3.4.1. att. trešais līmenis – ārsta pacienti

3.3.5. Trešais līmenis – pacienta rehabilitācijas nosacījumi

Trešajā līmenī pacienta ierakstu kopā glabāsies informācija par pacienta rehabilitācijas nosacījumiem. Tā saturēs sekojošus laukus – rehabilitācijas sesijas ilgums, rehabilitācijas biežums, optimālais ceļgala saliekšanas leņķis un maksimālais ceļgala saliekšanas leņķis.



3.3.5.1. att. trešais līmenis – pacienta rehabilitācijas nosacījumi

3.3.6. Trešais līmenis – pacienta rehabilitācijas sesiju vēsture

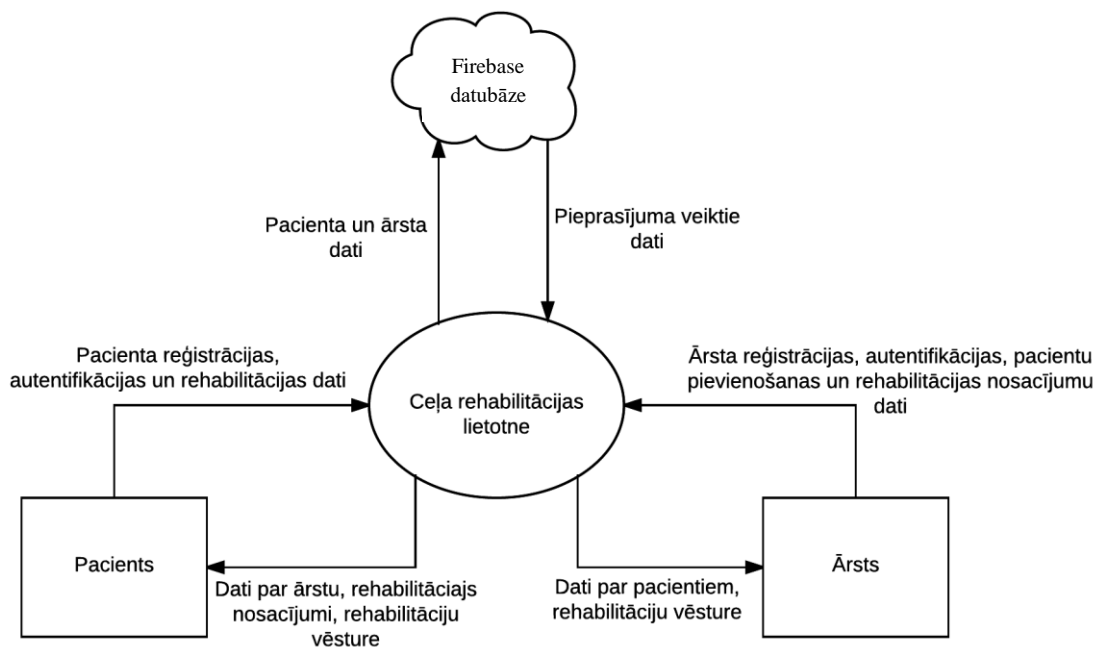
Trešajā līmenī pacienta datu kopā glabāsies arī rehabilitācijas sesiju vēsture. Katram rehabilitācijas sesijas ierakstam būs unikāla, automātiski ģenerēta identifikācijas atslēga. Katrs no šiem ierakstiem saturēs sekojošus laukus – komentārs, rehabilitācijas veikšanas datums un laiks, rehabilitācijas veikšanas ilgums, maksimālā ceļa saliekšanas leņķa reižu skaits.



3.3.6.1. att. trešais līmenis – pacienta rehabilitācijas sesiju vēsture

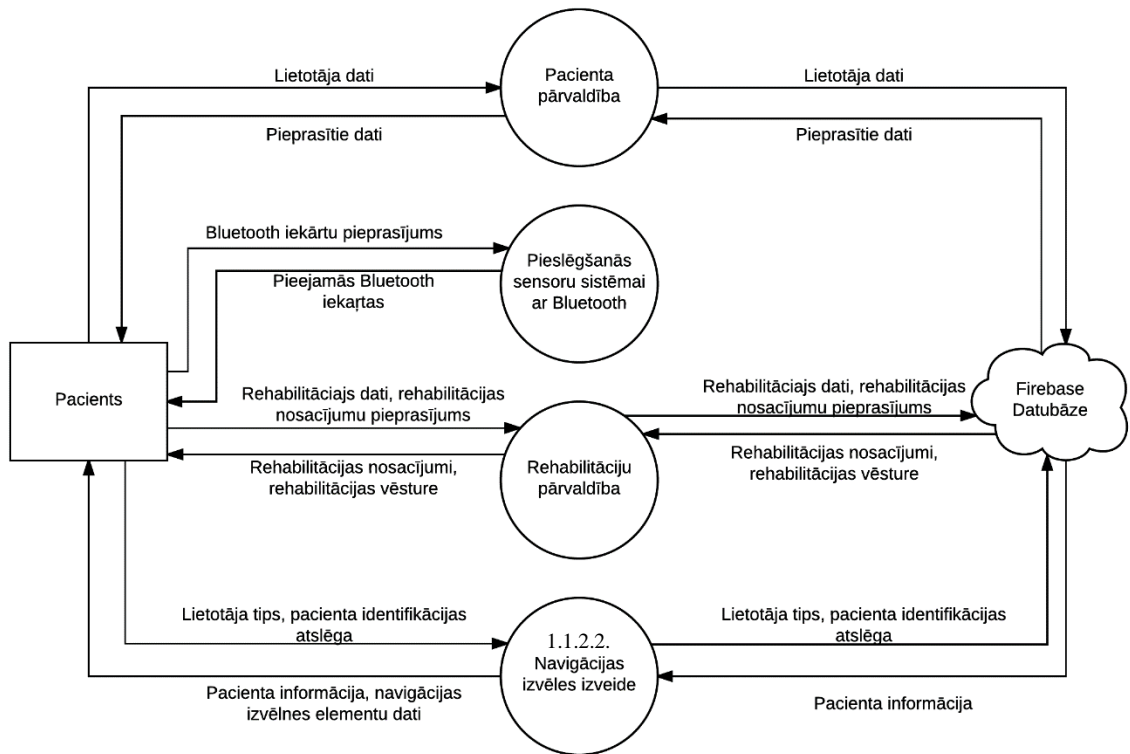
3.4. Datu plūsmu diagrammas

3.4.1. Datu plūsmu diagrammas 0. līmenis



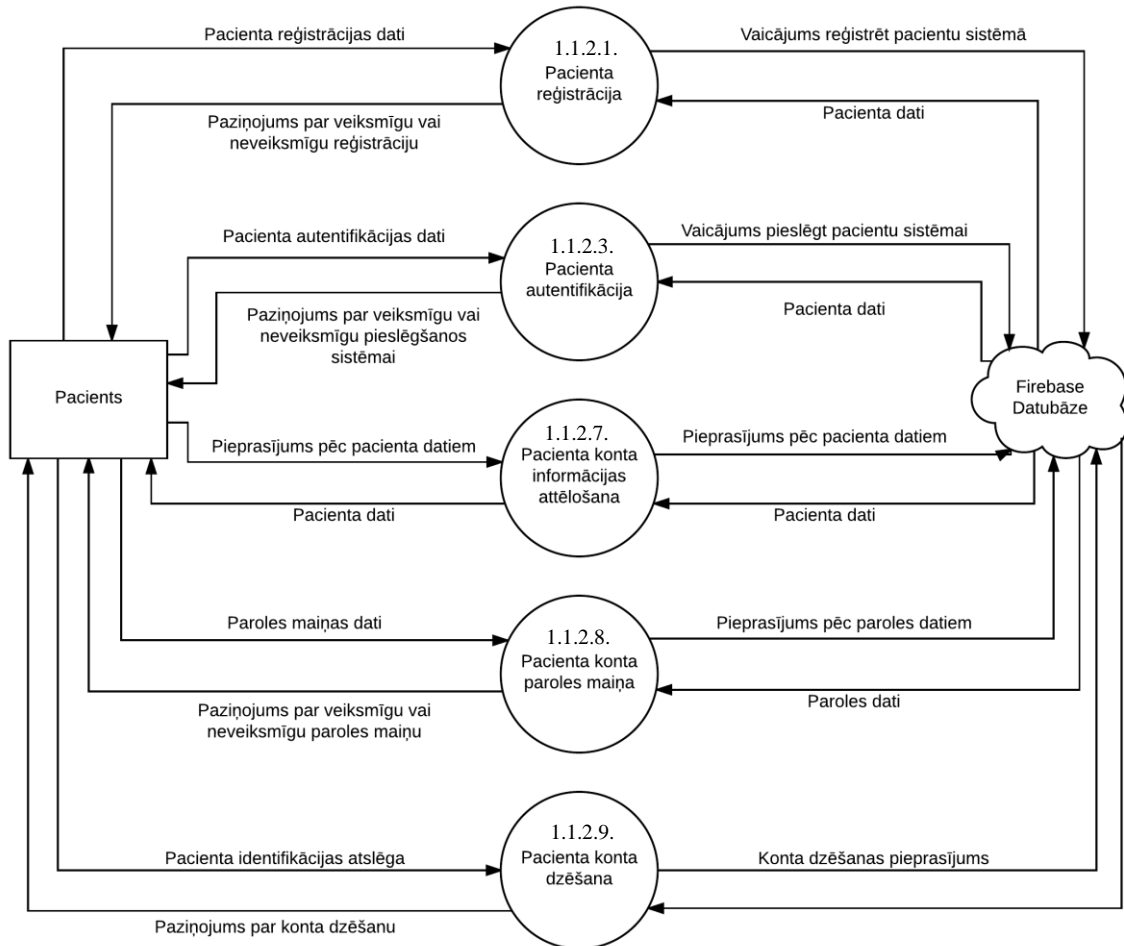
3.4.1.1. att. datu plūsmu diagrammas 0.līmenis

3.4.2. Pacienta datu plūsmu diagramma 1. līmenis



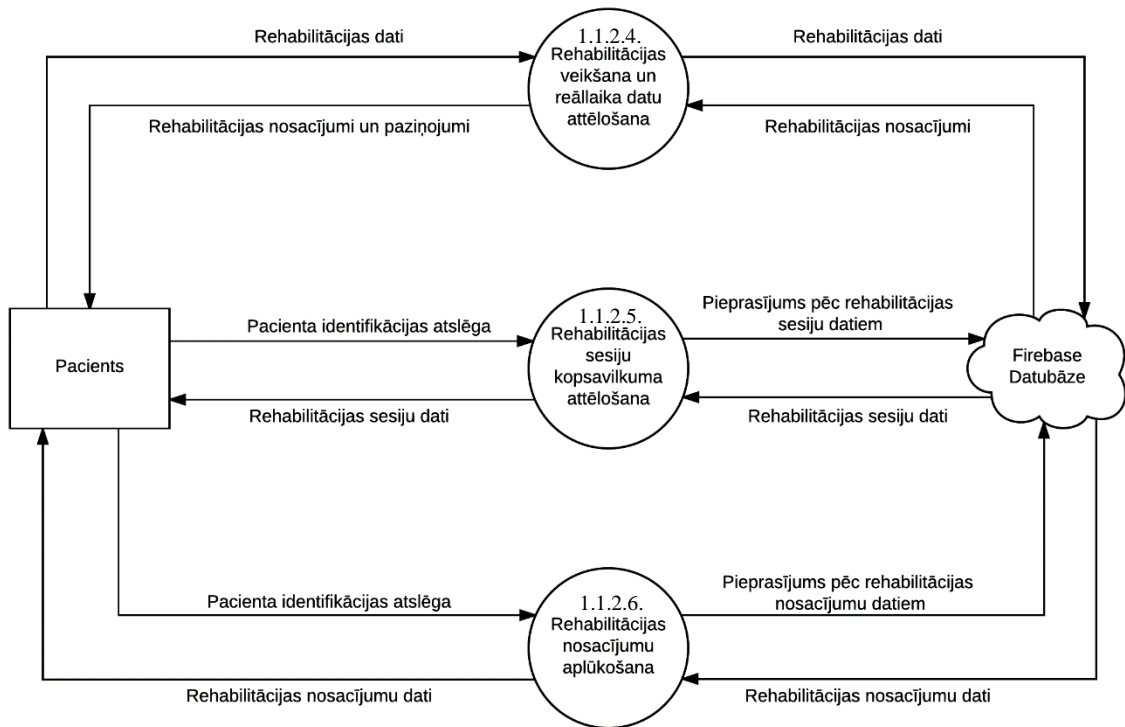
3.4.2.1 att. pacientu datu plūsmu diagrammas 1.līmenis

3.4.3. Pacienta datu plūsmu diagramma 2. līmenis “Pacienta pārvaldība”



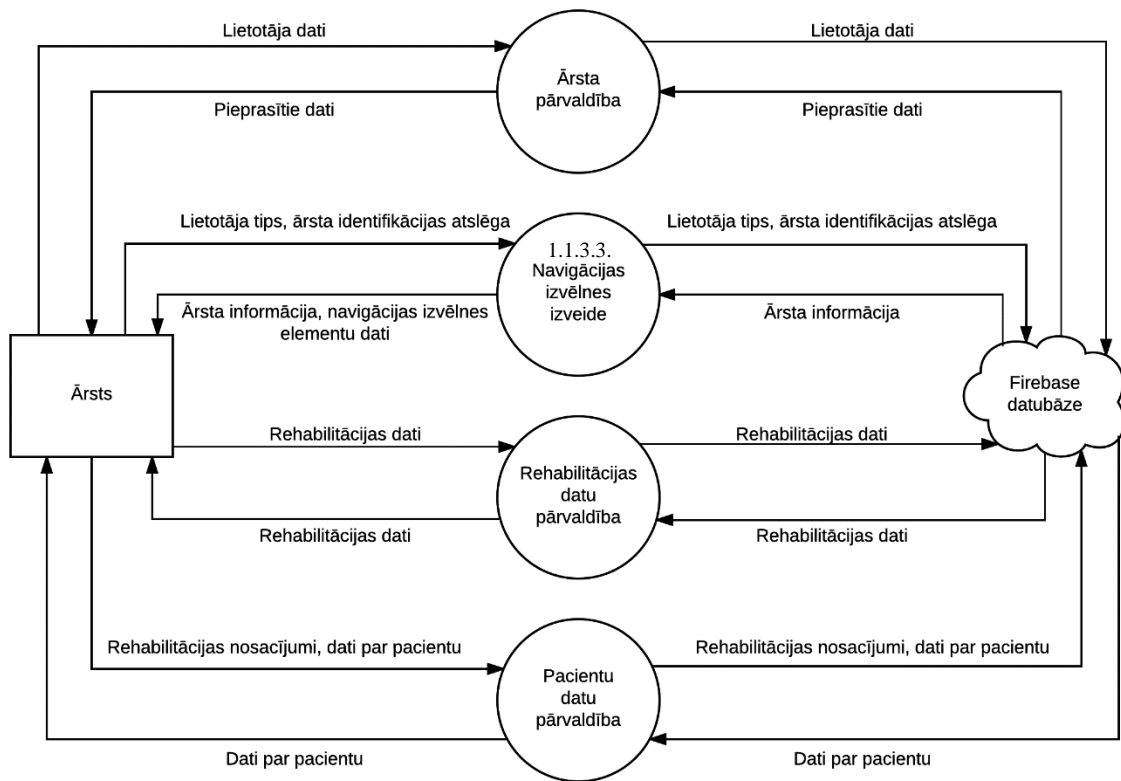
3.4.3.1 att. pacientu datu plūsmu diagrammas 2.līmenis “Pacienta pārvaldība”

3.4.4. Pacienta datu plūsmu diagramma 2. līmenis “Pacienta rehabilitācijas pārvaldība”



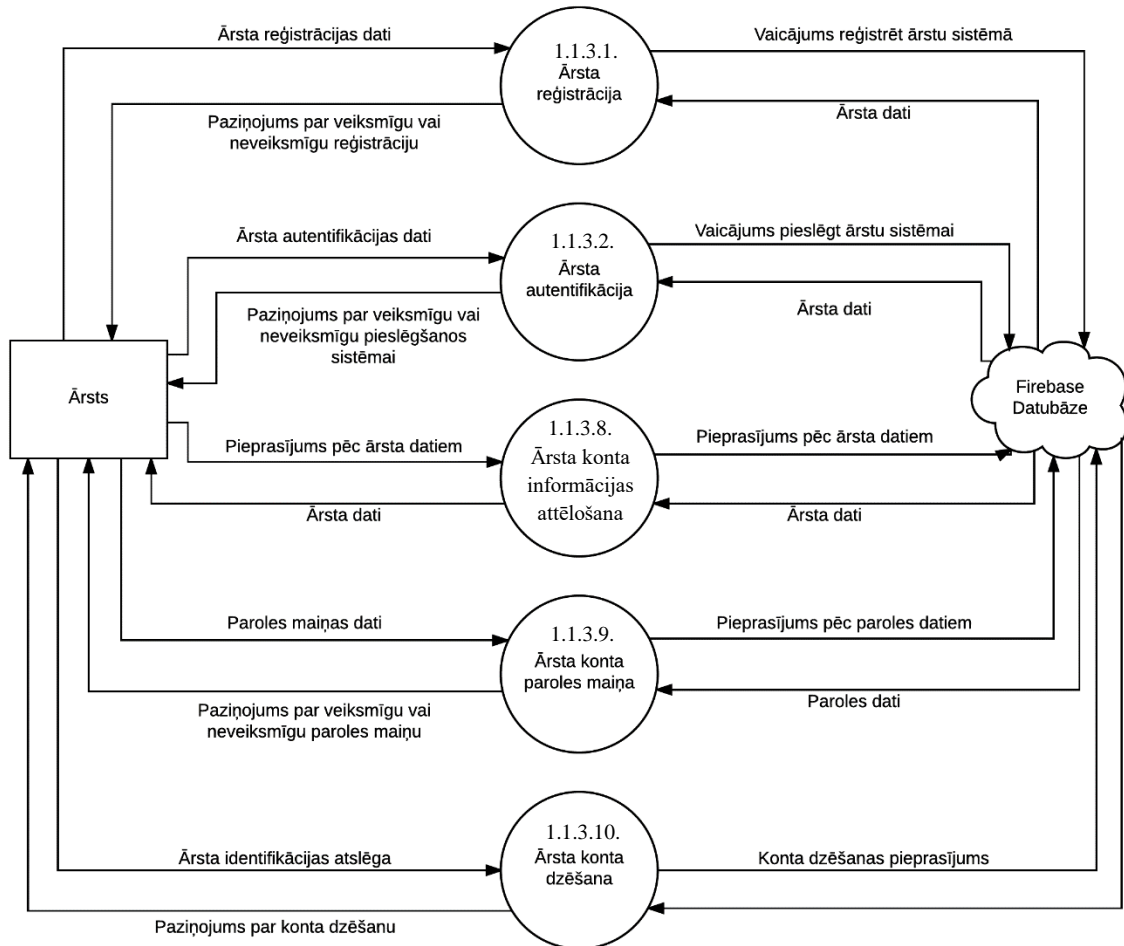
3.4.5.1. att. pacientu datu plūsmu diagrammas 2.līmenis “Pacienta rehabilitācijas pārvaldība”

2.4.5. Ārsta datu plūsmu diagramma 1. līmenis



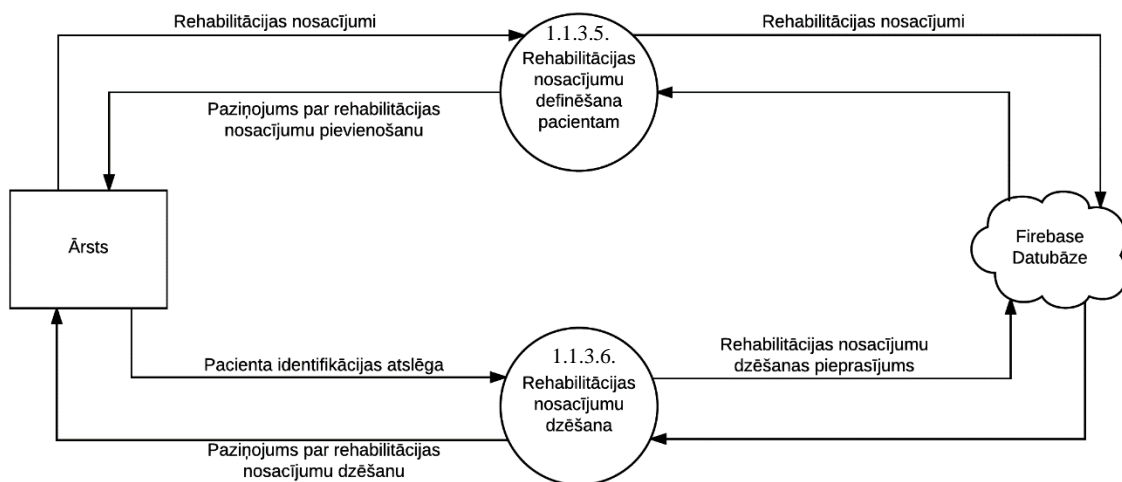
3.4.5.1 att. ārsta datu plūsmu diagramma 1. līmenis

2.4.6. Ārsta datu plūsmu diagramma 2. līmenis “Ārsta pārvaldība”



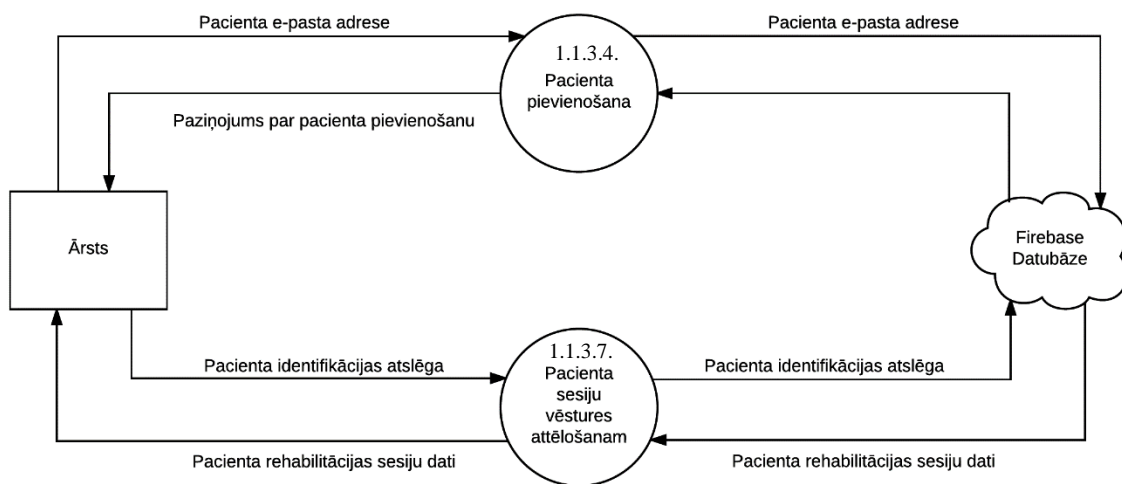
3.4.7.1. att. ārsta datu plūsmu diagrammas 2.līmenis “Ārsta pārvaldība”

2.4.7. Ārsta datu plūsmu diagramma 2. līmenis “Rehabilitācijas datu pārvaldība”



3.4.8.1. att. ārsta datu plūsmu diagrammas 2.līmenis “Rehabilitācijas datu pārvaldība”

2.4.8. Ārsta datu plūsmu diagramma 2. līmenis “Pacienta datu pārvaldība”



3.4.9.1.att. ārsta datu plūsmu diagrammas 2.līmenis “Pacientu datu pārvaldība”

2.5. Saskaņ�e ar ārējām sistēmām

Tiek lietota ārēja datu mākoņkrātuve *Firebase*. Tā sastāv no intuitīva un viegli lietojama API, tā ir responsīva, neatkarīgi no interneta aizkaves vai savienojamības, tādējādi darbojoties arī bezsaistē, sinhronizējot failus, kad savienojums ir pieejams.

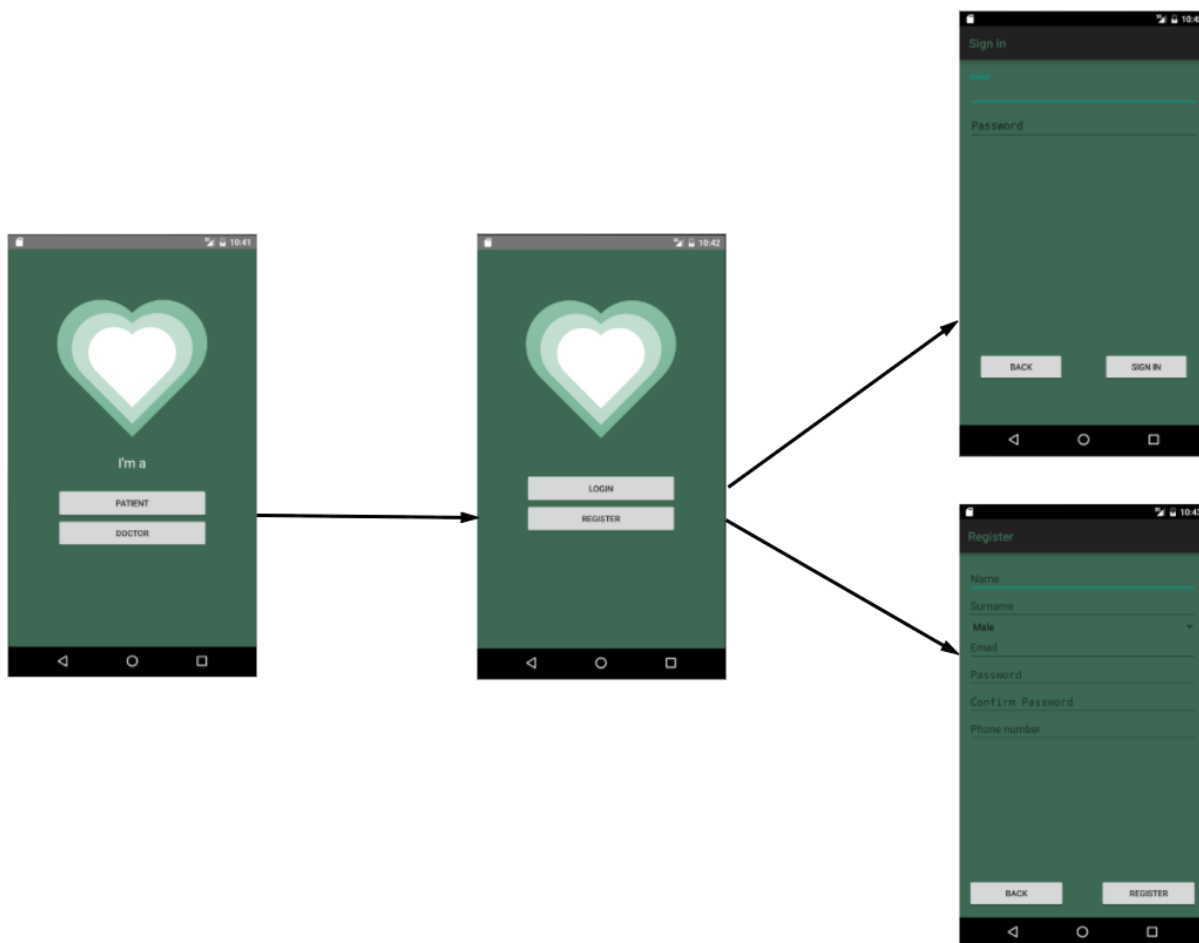
Sistēma komunicē un saņem informāciju arī no ārējās iegultās sensoru iekārtas.

2.6. Lietotnes saskaņņu diagrammas

Lietotnes saskaņņu diagrammās tiek attēlots to darbību gaita un skatu savstarpējā saistība.

2.6.1. Pieslēgšanās, reģistrācija un autentifikācija sistēmā

Tā kā reģistrēšanas un autentifikācijas process abām lietotnes daļām ir vienāds, šis process tiek attēlots vienā saskarņu diagrammā. Sākumā notiek lietotāja tipa izvēle, pēc tam tiek veikta izvēle pieslēgties sistēmai vai reģistrēties.



3.6.1.1. att. kopēja saskarņu diagramma

2.6.2. Pacienta lietotnes daļa

Lietotāja izvēles logā atzīmējot “Patient” un veiksmīgi pieslēdzoties sistēmai, tiek parādīta pacienta lietotnes daļa. Pacientam ir iespēja veikt dažādu darbību izvēli izvēlnes logā. To savstarpējā hierarhija ir attēlota diagrammā.



3.6.2.1. att. pacientu saskarņu diagramma

2.6.3. Ārsta lietotnes daļa

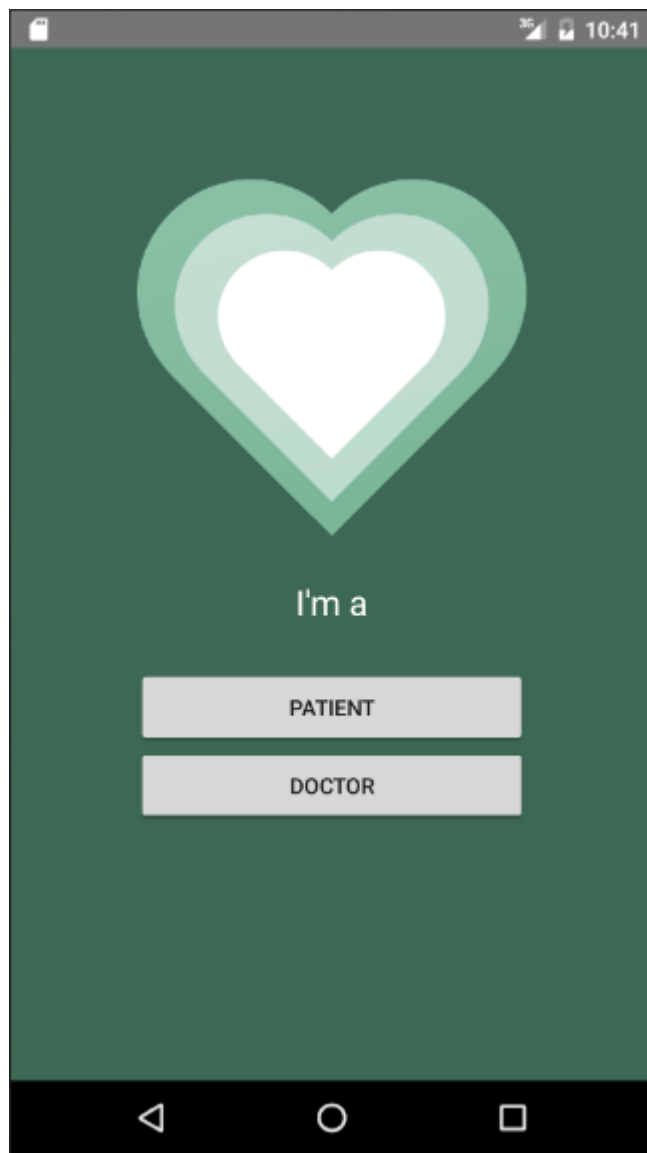
Lietotāja izvēles logā atzīmējot “Doctor” un veiksmīgi pieslēdzoties sistēmai, tiek parādīta ārsta lietotnes daļa. Ārstam ir iespēja veikt dažādu darbību izvēli izvēlnes logā. To savstarpējā hierarhija ir attēlota diagrammā.



3.6.3.1. att. ārsta saskarņu diagramma

2.7. Ekrānformu projektējums

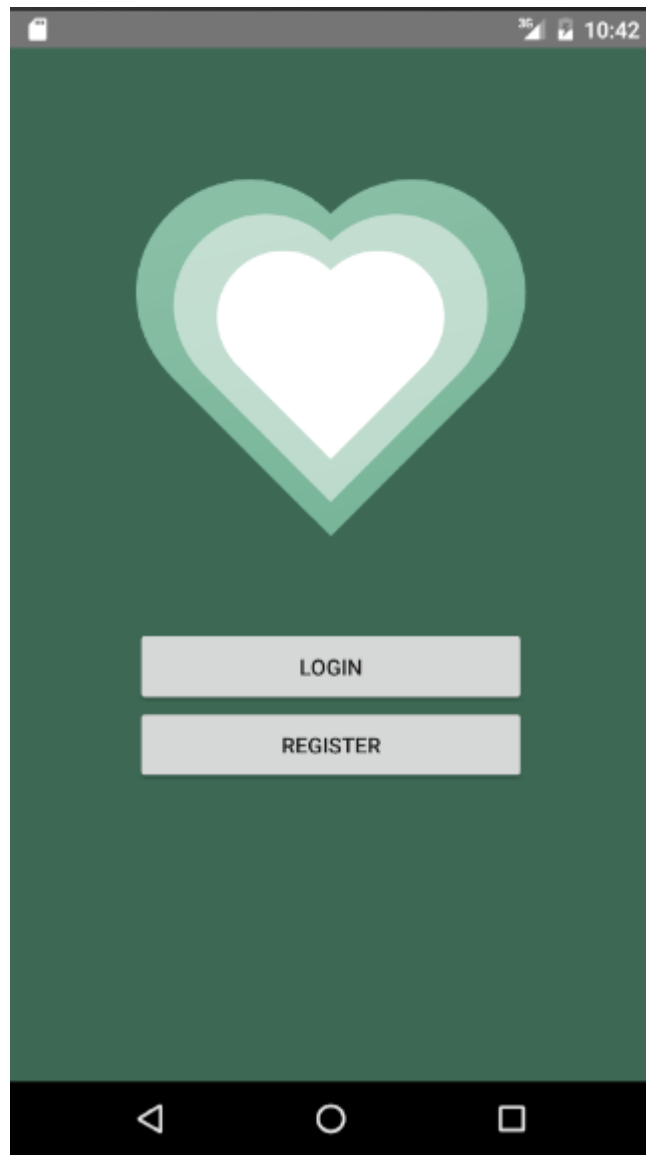
2.7.1. Lietotāja tipa izvēles skats



3.7.1.1. att. lietotāja tipa izvēles skats

Šis ir pirmais skats, ko lietotājs ierauga, atverot lietotni. Tajā tiek veikta izvēle par lietotāja tipu – ārsts vai pacients, attiecīgi spiežot uz pogām “PATIENT” vai “DOCTOR”. Pēc izvēles veikšanas tiek attēlos sākuma izvēles skats.

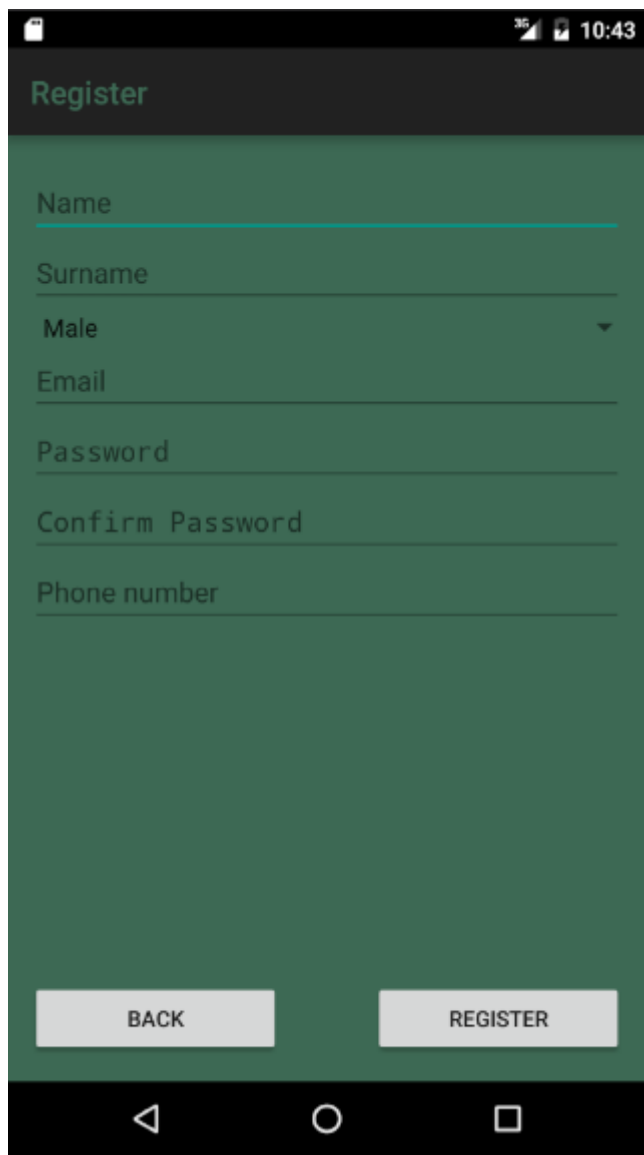
2.7.2. Sākuma izvēles skats



3.7.2.1 att. sākuma izvēles skats

Sākuma izvēles skatā lietotājam ir iespēja izdarīt izvēli starp reģistrāciju sistēmā, spiežot pogu “REGISTER”, un pieslēgšanos sistēmai, spiežot pogu “LOGIN”. Pēc izvēles veikšanas, lietotājam attiecīgi tiek attēlos reģistrācijas vai pieslēgšanās skats.

2.7.3. Reģistrācijas skats



3.7.3.1. att. reģistrācijas skats

Lietotāja reģistrācijas skatā ir 6 teksta ievades un viens lejupslidošās izvēles lauks. Nospiežot pogu “REGISTER” notiek lauku validācija. Nesekmīgas validācijas rezultātā nepareizi aizpildītajiem vai tukšajiem laukiem parādās kļūdas paziņojums. Veiksmīgas datu validācijas gadījumā lietotājam tiek attēlots pieslēgšanās skats. Nospiežot pogu “BACK” lietotājam tiek attēlots sākuma izvēles skats.

2.7.4. Pieslēgšanās skats

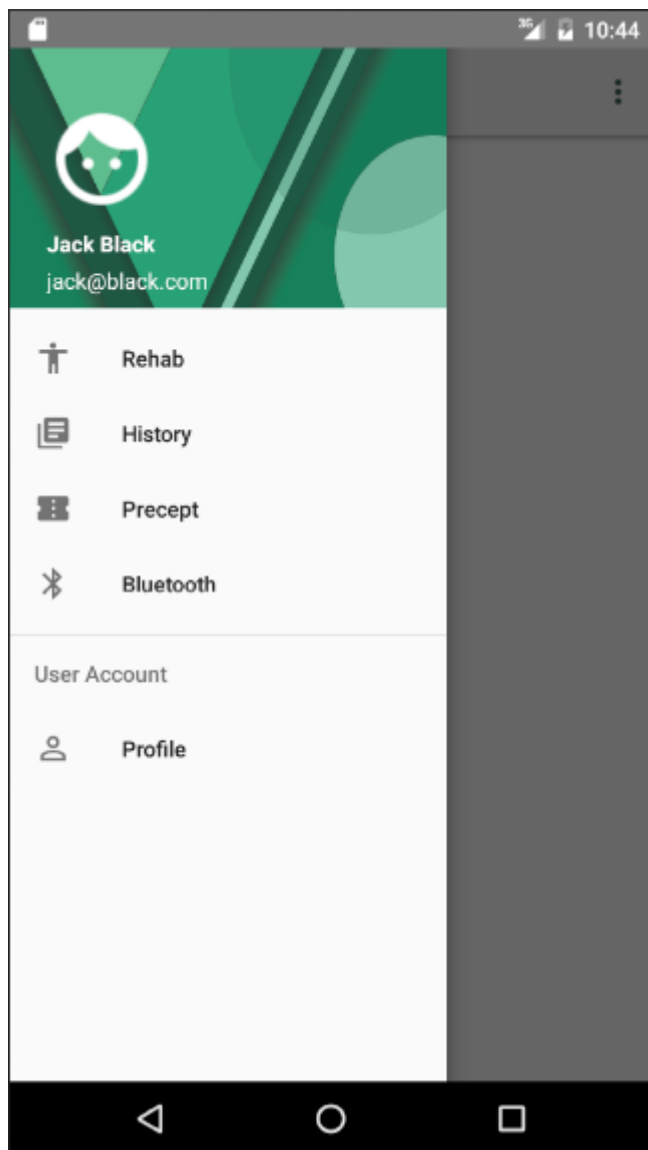


3.7.4.1. att. pieslēgšanās skats

Pieslēgšanās skatā ir divi teksta ievades lauki – e-pasta adrese un parole. Pēc pogas “SIGN IN” nospiešanas lauki tiek paslēpti, attēlota reģistrācijas animācija. Sekmīgas reģistrācijas gadījumā, atkarībā no lietotāja tipa, tiek parādīti to noklusējuma skati – pacientam rehabilitācijas nosacījumu skats un ārstam – pacienta pievienošanas skats. Nesekmīgas pieslēgšanās gadījumā tiek parādīts *Toast* tipa paziņojums. Nospiežot pogu “BACK” lietotājam tiek attēlots sākuma izvēles skats.

2.7.5. Lietotnes pacienā daļas ekrānformas

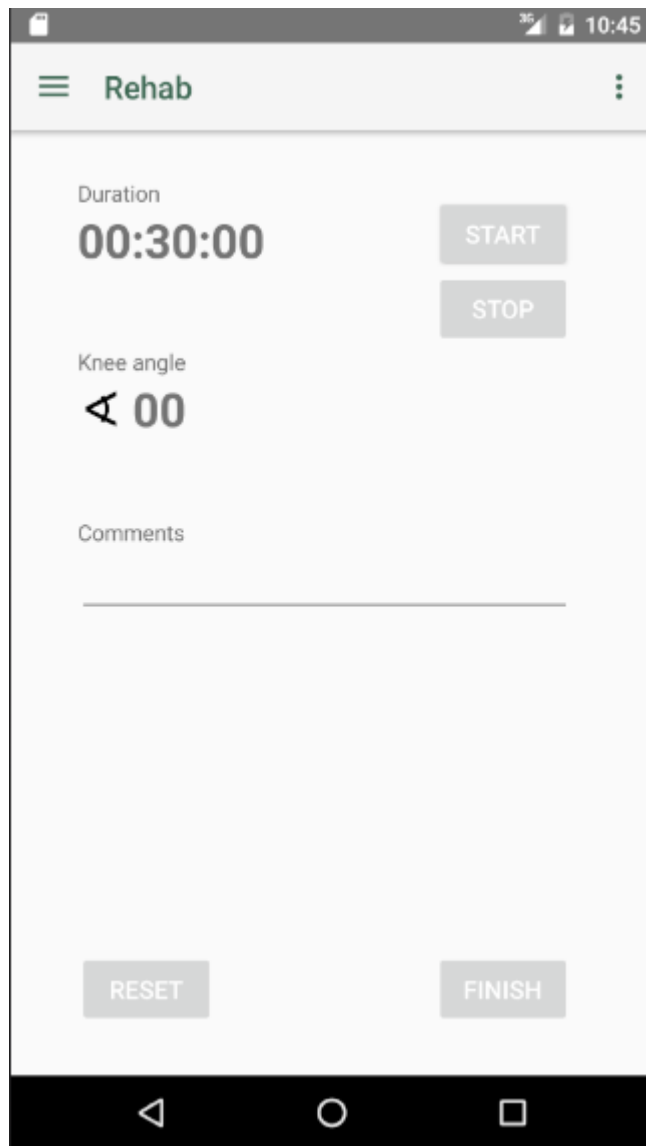
2.7.5.1. Pacienā navigācijā izvēlnes skats



3.7.5.1.1. att. pacienā izvēlnes navigācijā skats

Pacienā izvēlnes skata galvenē tiek ielādēti pacienā dati – vārds, uzvārds un tā e-pasta adrese. Izvēlnes galvenajā daļā tiek parādīti pacienā izvēlnes elementi – “Rehab”, “History”, “Precept”, “Bluetooth”, “Profile”. Nospiežot uz kādā no navigācijā izvēlnes elementiem tiek attēlots kāds no pieciem lietotāja izvēlnes skatiem.

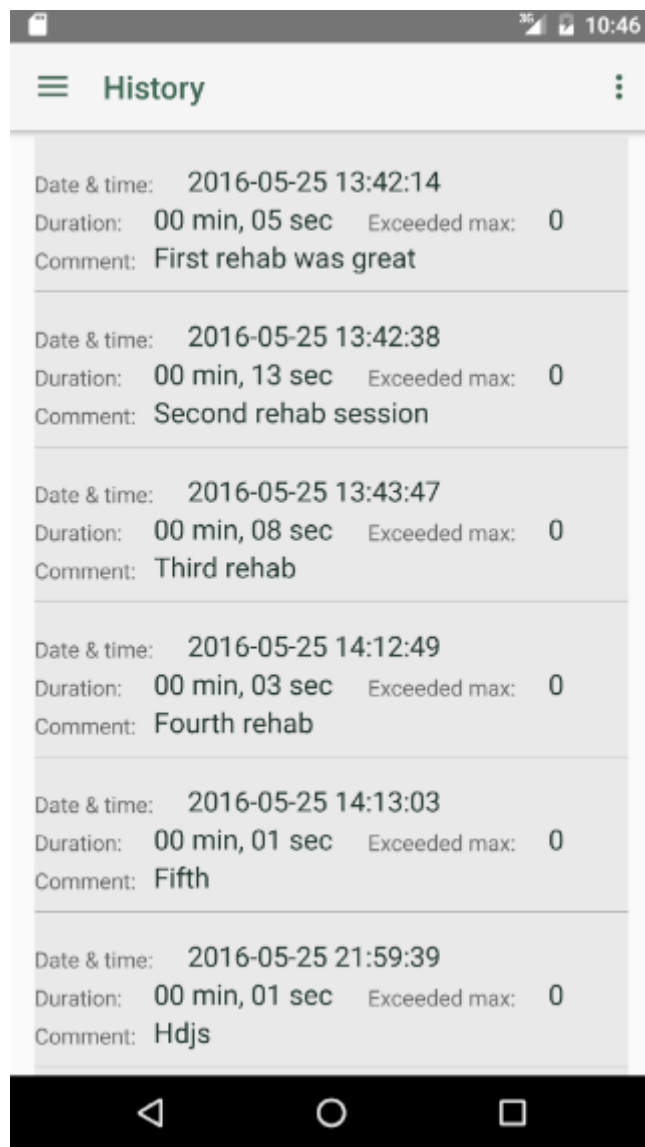
2.7.5.2. Rehabilitācijas sesijas veikšanas skats



3.7.5.2.1.att. lietotāja tipa izvēles skats

Rehabilitācijas sesijas veikšanas skatā ir redzams laika atskaites logrīks. Sākotnēji ir iespējota tikai poga "START". Pēc tās nospiešanas tiek sākta laika atskaite, iespējota poga "STOP" un atspējota poga "START". Pēc pogas "STOP" nospiešanas tiek iespējots komentāra teksta ievades lauks, pogas "RESET" un "FINISH" un apturēts laika atskaites logrīks. Nospiežot pogu "RESET", tiek atiestatīts laika atskaites logrīks, iespējota poga "START" un atspējotas pogas "FINISH" un "STOP". Nospiežot pogu "FINISH", rehabilitācijas dati tiek nosūtīti uz datubāzi un atjaunoti sākotnējie skata iestatījumi.

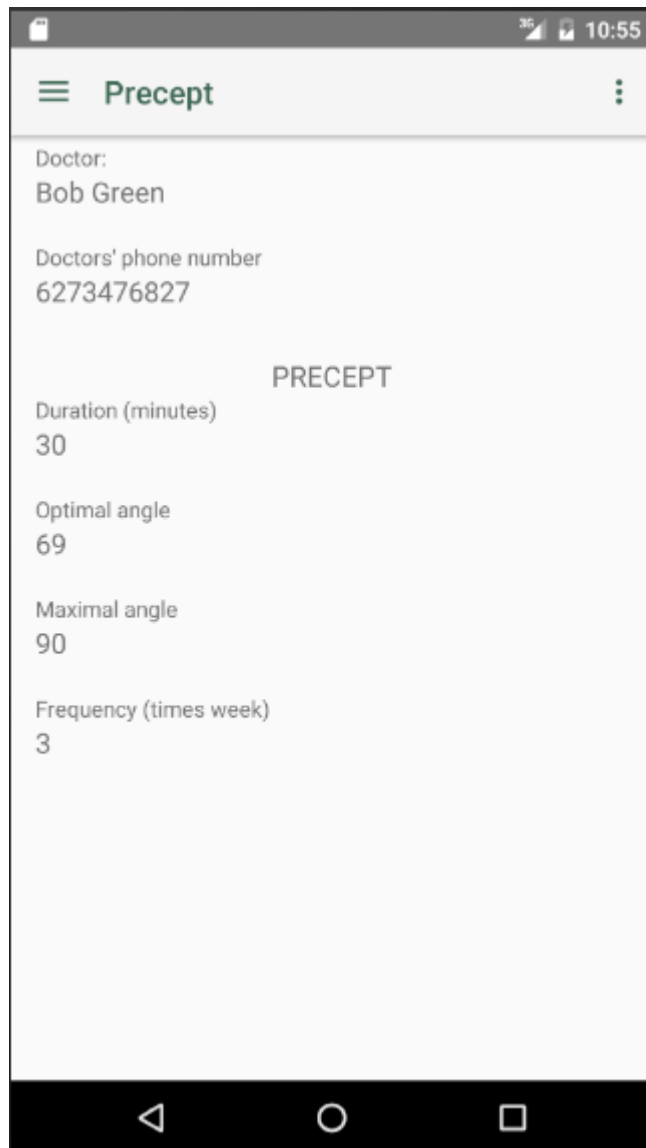
2.7.5.3. Rehabilitācijas sesiju vēstures skats



3.7.5.3.1.att. lietotāja tipa izvēles skats

Rehabilitācijas sesiju vēstures skatā ir attēloti saraksta elementi, kur katrā no tiem ir redzams rehabilitācijas sesijas datums un laiks, rehabilitācijas sesijas ilgums, maksimālā ceļa saliekšanas leņķa pārsniegšanas reižu skaits un pacienta pievienotais rehabilitācijas sesijas komentārs.

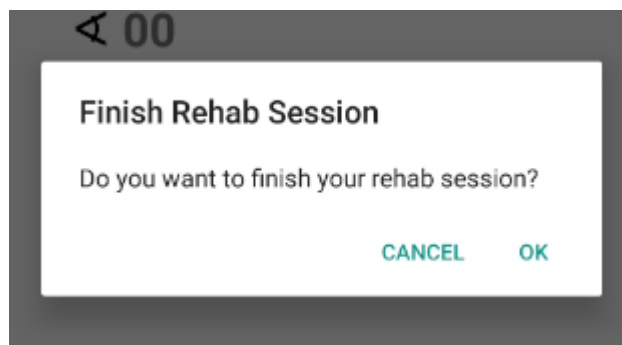
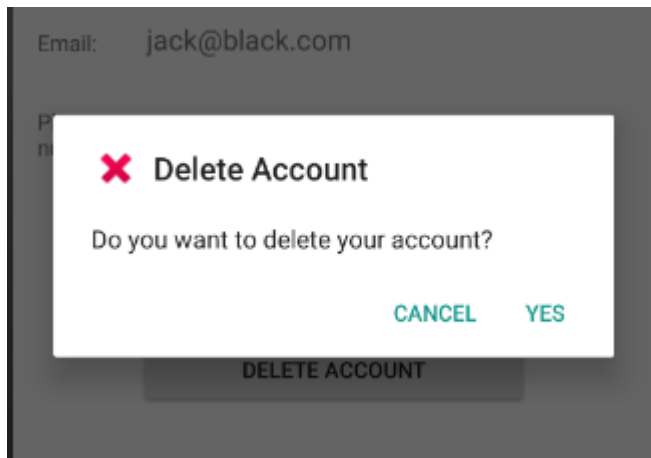
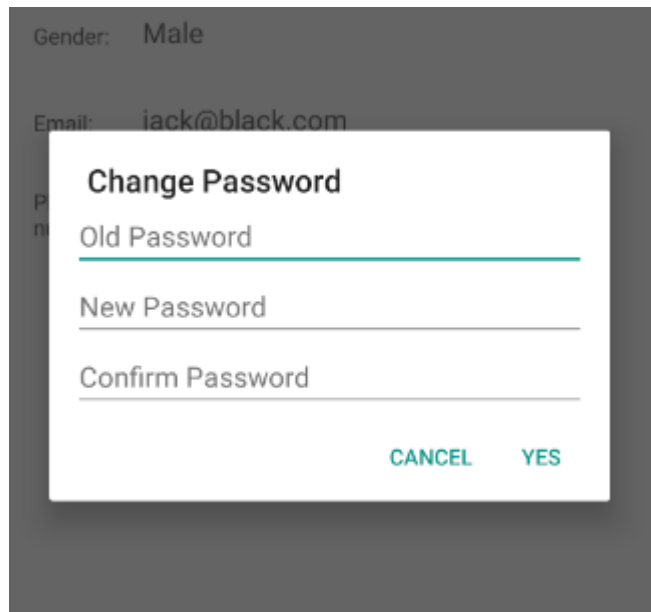
2.7.5.4. Rehabilitācijas nosacījumu skats



3.7.5.4.1.att. lietotāja tipa izvēles skats

Rehabilitācijas nosacījumu skatā ir attēloti dati par ārstējamo ārstu – tā vārds, uzvārds un telefona numurs. Ja pacientam nav pievienots rehabilitāciju veicošais ārsts, tad tiek parādīts *Toast* tipa paziņojums par ārsta neesamību. Zem ārsta datiem tiek attēloti rehabilitācijas nosacījumu dati – rehabilitācijas ilgums minūtēs, optimālais ceļa saliekššanas leņķis, maksimālais ceļa saliekššanas leņķis un rehabilitācijas veikšanas biežums (reizes nedēļā).

2.7.5.5. Paziņojumu dialogi

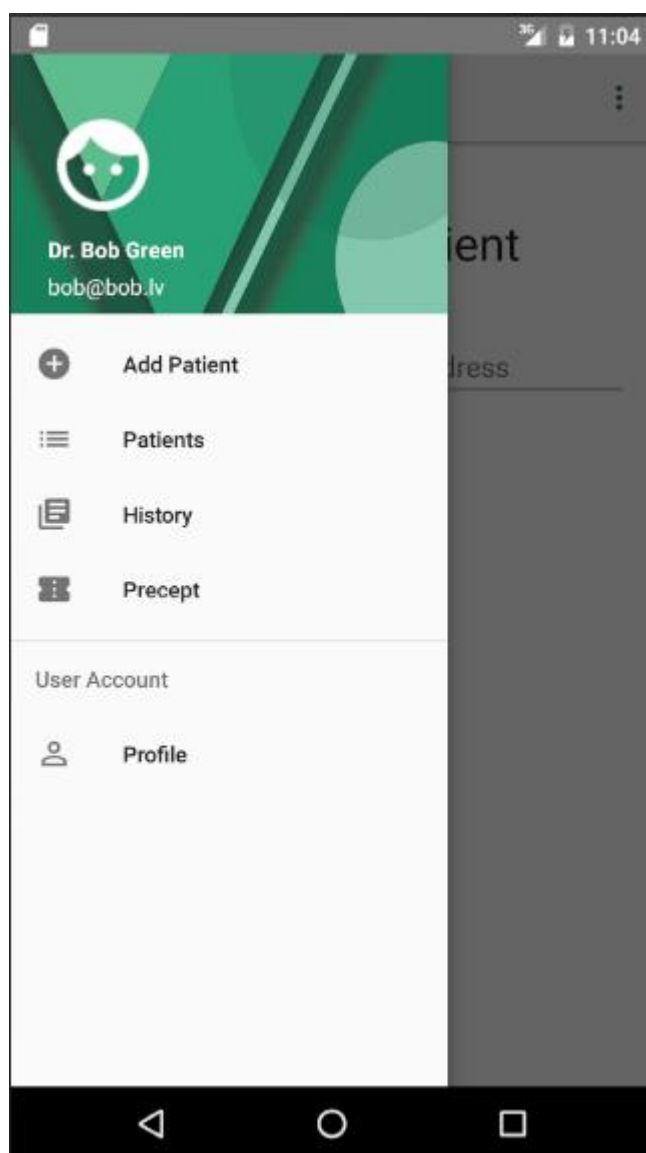


3.7.5.5.1.att. lietotāja tipa izvēles skats

Lietotnē dažām no funkcijām ir paziņojumu dialogi. Rehabilitācijas sesijas beigšanai un pacienta konta dzēšanai tie ir pārbaudes veida, mainot konta paroli paziņojuma dialogā ir jāievada vecā parole, jaunā un atkārtoti jaunā.

2.7.6. Lietotnes ārsta daļas ekrānformas

2.7.6.1. Ārsta navigācijas izvēlnes skats

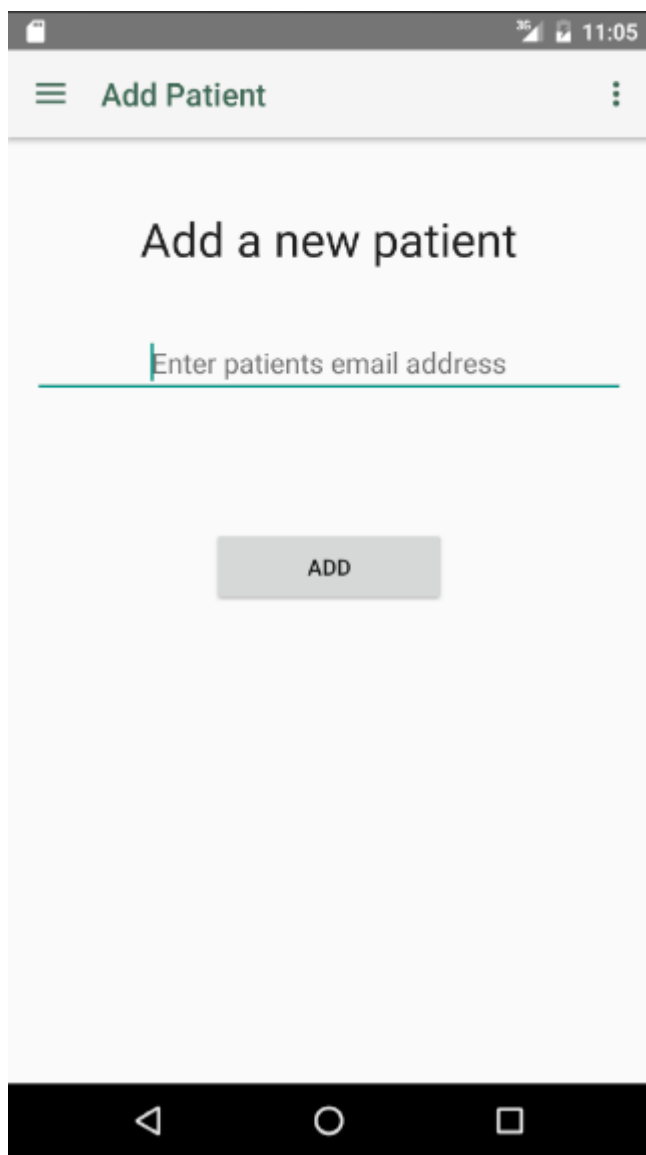


3.7.6.1.1.att. lietotāja tipa izvēlnes skats

Ārsta izvēlnes skata galvenē tiek ielādēti ārsta dati – vārds, uzvārds un tā e-pasta adrese. Izvēlnes galvenajā daļā tiek parādīti ārsta izvēlnes elementi – “Add Patient”, “Patients”,

“History”, “Precept”, “Profile”. Nospiežot uz kāda no navigācijas izvēlnes elementiem tiek attēlots kāds no pieciem lietotāja izvēlnes skatiem.

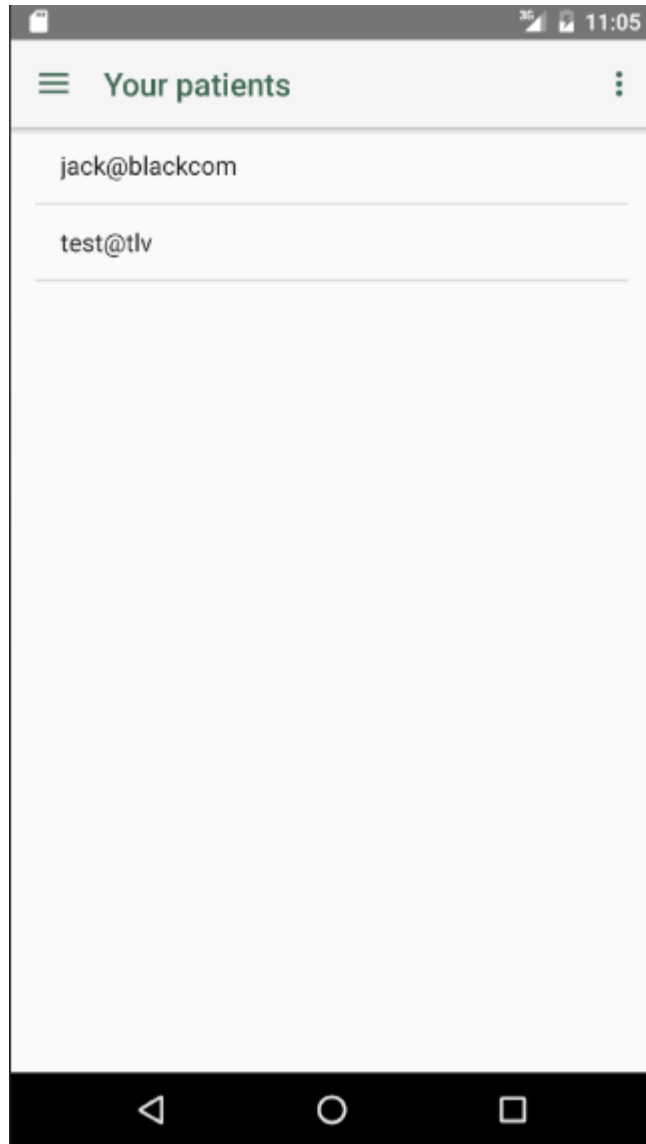
2.7.6.2. Pacienta pievienošanas skats



3.7.6.2.1.att. lietotāja tipa izvēlnes skats

Pacienta pievienošanas skatā ir teksta ievades lauks, kurā ārstam ir nepieciešams ievadīt pacienta e-pasta adresi, lai to kā savu pacientu. Pēc pogas “ADD” nospiešanas tiek veikta ievadītās e-pasta adreses pārbaude. Apstiprinošā gadījumā tiek parādīts *Toast* tipa paziņojums par pacienta pievienošanu, pretēji – par to, ka pacients ar norādītu e-pasta adresi neeksistē sistēmā vai jau ir pievienots.

2.7.6.3. Ārsta pacientu saraksta skats



3.7.5.3.1.att. lietotāja tipa izvēles skats

Ārsta pacientu saraksta skatā tiek attēlots saraksts ar ārsta pacientu e-pasta adresēm.

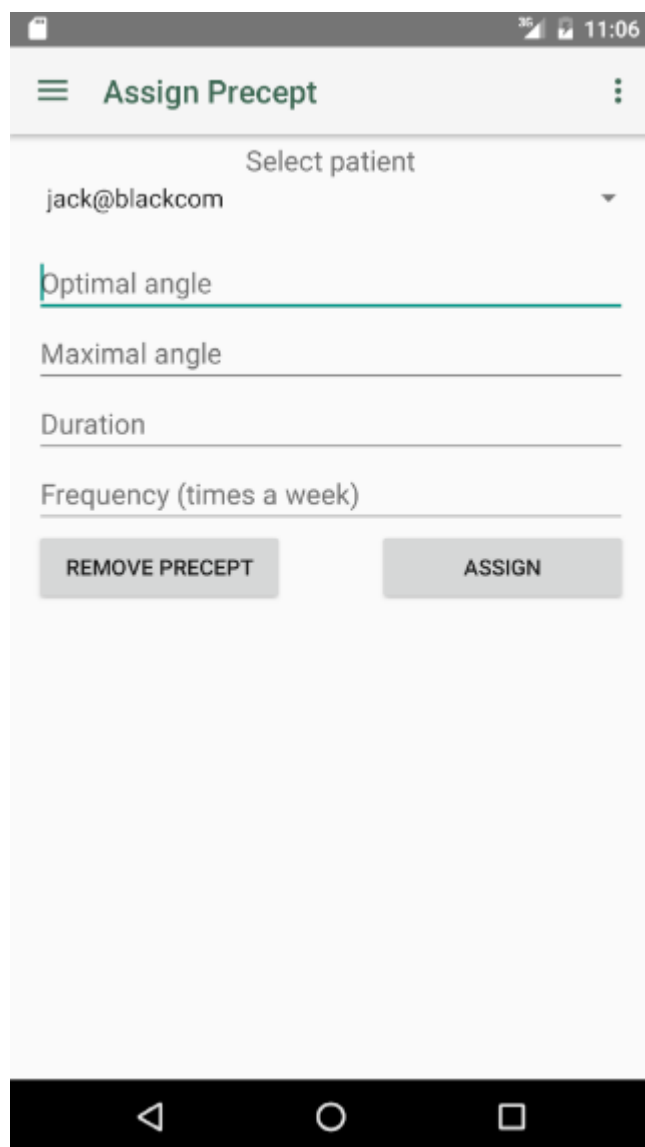
2.7.6.4. Ārsta pacientu vēstures skats



3.7.5.4.1.att. lietotāja tipa izvēles skats

Ārsta pacientu vēstures skata augšējā daļā ārsts var izvēlēties kādu no saviem pacientiem no lejupslidošās izvēlnes. Kad pacients ir atlasīts, nospiežot pogu “SHOW” tiek attēlota attiecīgā pacienta rehabilitācijas sesiju vēsture.

2.7.6.5. Rehabilitācijas nosacījumu pievienošanas un noņemšanas skats



The screenshot displays the 'Assign Precept' screen of a mobile application. At the top, there is a header with a hamburger menu icon on the left, the title 'Assign Precept' in the center, and a vertical ellipsis icon on the right. Below the header is a 'Select patient' dropdown menu showing 'jack@blackcom'. Underneath the dropdown are four text input fields: 'Optimal angle', 'Maximal angle', 'Duration', and 'Frequency (times a week)'. At the bottom of the screen, there are two buttons: 'REMOVE PRECEPT' on the left and 'ASSIGN' on the right. The entire interface is set against a light gray background. The bottom of the image shows the standard Android navigation bar with back, home, and recent apps icons.

3.7.5.5.1.att. lietotāja tipa izvēles skats

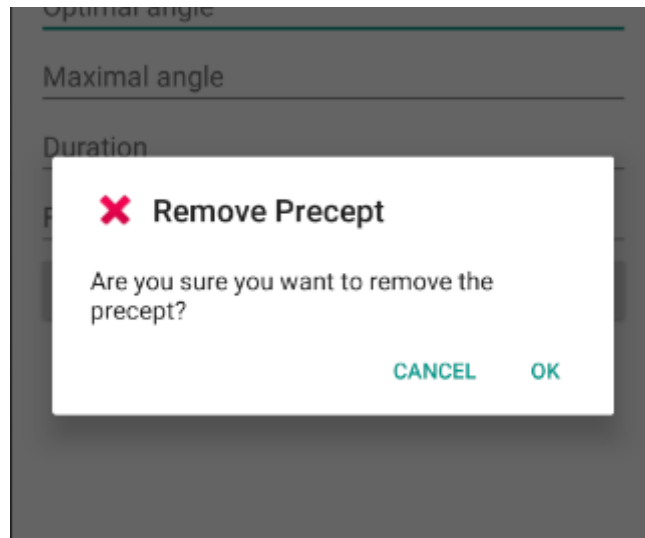
Rehabilitācijas nosacījumu pievienošanas un noņemšanas skatā augšējā daļā tiek attēlota lejupslidošā izvēlnes ar visiem ārsta pacientiem, ļaujot ārstam izvēlēties kādu no tiem. Tālāk ir četri teksta ievades lauki – optimālais ceļa saliekšanas leņķis, maksimālais ceļa saliekšanas leņķis, rehabilitācijas sesijas ilgums un rehabilitācijas biežums (reizes nedēļā). Pēc pogas “ASSIGN” nospiešanas, notiek lauku aizpildes validācija un veiksmīga iznākuma gadījumā informācija tiek nosūtīta uz datubāzi, lietotājam parādot *Toast* tipa paziņojumu par veiksmīgu rehabilitācijas nosacījumu pievienošanu. Ja visi lauki nav aizpildīti, parādās kļūdas paziņojumi zem neaizpildītajiem laukiem. Ja nosacījumi atlasītajam pacientam jau ir pievienoti, tad par to

paziņo *Toast* tipa paziņojums. Nospiežot pogu “REMOVE PRECEPT”, atlasītajam ārstam tiek attēlots paziņojuma dialoga vaicājums par pacienta rehabilitācijas nosacījumu dzēšanu. Apstiprinošas izvēles gadījumā dati tiek dzēsti no datubāzes un parādīts *Toast* tipa paziņojums.

2.7.6.6. Paziņojumu dialogi

The image displays two screenshots of mobile application dialog boxes. The top screenshot shows a 'Change Password' dialog box. It has a title 'Change Password' and three input fields: 'Old Password', 'New Password', and 'Confirm Password'. At the bottom right of the dialog are two buttons: 'CANCEL' and 'YES'. Below the dialog box is a button labeled 'DELETE ACCOUNT'. The background of the dialog is dark gray, and the dialog itself is white with a dark border. The text 'Email: bob@bob.lv' is visible at the top left of the background.

The bottom screenshot shows a 'Delete Account' dialog box. It features a red 'X' icon followed by the title 'Delete Account'. Below the title is the question 'Do you want to delete your account?'. At the bottom right of the dialog are two buttons: 'CANCEL' and 'YES'. The background of the dialog is dark gray, and the dialog itself is white with a dark border. The text 'Phone number: 6273476827' is visible at the top left of the background.



3.7.5.6.1.att. lietotāja tipa izvēles skats

Lietotnē dažām no funkcijām ir paziņojumu dialogi. Rehabilitācijas nosacījumu un pacienta konta dzēšanai tie ir pārbaudes veida, mainot konta paroli paziņojuma dialogā ir jāievada vecā parole, jaunā un atkārtoti jaunā.

3. TESTĒŠANAS DOKUMENTĀCIJA

3.1. Ievads

Testēšanas dokumentācijā ir aprakstītas ceļa rehabilitācijas lietotnes testēšanas metodes un rezultāti. Testēšanai tika izmantoti akcepttesti un vienībtesti. Lai pārliecinātos par lietotnes atbilstību programmatūras prasību specifikācijai, akcepttestus veica gan izstrādātājs, gan pasūtītājs.

3.2. Testēšanas metodika

Testēšana tika veikta uz virtuālā Android emulatora, pārbaudot lietotni uz dažādām ierīcēm. Papildus virtuālai testēšanai, tā tika veikta arī uz fiziskas Android ierīces – Motorola Moto G.

Sistēmas testēšana notika ievadot gan korektus, gan aplamus ievaddatus, pārbaudot visus iespējamus iznākuma gadījumus.

Testēšana tika sākota jau pašā sistēmas izstrādes sākumā, pārbaudot katru pievienoto funkciju.

3.3. Testēšanas žurnāls

3.3.1. Lietotāja tipa izvēles veikšanas tests

Ievaddati	Sagaidāmais rezultāts	Rezultāts (Jā/ nē)	Datums
Izvēlēts lietotāja pacienta lietotāja tips, nospiežot pogu "PATIENT"	SharedPreferences saglabā lietotāja tipu "patient"	Nē	3.03.2016.
Izvēlēts lietotāja pacienta lietotāja tips, nospiežot pogu "PATIENT"	SharedPreferences saglabā lietotāja tipu "patient"	Jā	3.03.2016.

Izvēlēts lietotāja pacienta lietotāja tips, nospiežot pogu “DOCTOR”	SharedPreferences saglabā lietotāja tipu “doctor”	Jā	3.03.2016.
---	---	----	------------

3.3.2. Pacienta reģistrācija sistēmā

Ievaddati	Sagaidāmais rezultāts	Rezultāts (Jā/ nē)	Datums
Pacienta ievaddati: name: Jack, surname: Black, gender: Male, email: jack@black.com , password: jack1234, phone number: +371 29028980	Pacients veiksmīgi pievienots datubāzē. Toast tipa paziņojums par veiksmīgu lietotāja reģistrāciju sistēmā.	Jā	18.03.2016.
Pacienta ievaddati: name: Jack, surname: Black, gender: Male, email: jackblack.com , password: jack1234, password(repeated): jack1234, phone number: +371 29028980	Kļūdas paziņojums e-pasta adreses laukā par nepareizas formas e-pasta adresi. Toast tipa paziņojums par to, ka lietotāja reģistrācija sistēmā ir neveiksmīga.	Jā	18.03.20106.
Pacienta ievaddati: name: Jack, surname: Black, gender: Male, email: jack@black.com , password: jack1234,	Kļūdas paziņojums telefona numura laukā par to, ka šis lauks ir jāaizpilda obligāti. Toast tipa paziņojums par to, ka lietotāja reģistrācija sistēmā ir neveiksmīga.	Jā	18.03.20106.

password(repeated): jack1234, phone number:			
Pacienta ievaddati: name: Jack, surname: Black, gender: Male, email: jack@black.com , password: jack1234, password(repeated): jack1234, phone number: +371 29028980 Jau pēc šāda konta reģistrācijas	Kļūdas paziņojums par to, ka lietotājs ar šādu e-pasta adresi jau ir reģistrēts sistēmā. Toast tipa paziņojums par to, ka lietotāja reģistrācija sistēmā ir neveiksmīga.	Jā	18.03.2016.
Pacienta ievaddati: name: Jack, surname: Black, gender: Male, email: jack@black.com , password: jack1234, password(repeated): jack12, phone number: +371 29028980	Kļūdas paziņojums par to, ka ievadītās paroles nesakrīt. Toast tipa paziņojums par to, ka lietotāja reģistrācija sistēmā ir neveiksmīga.	Nē	18.03.2016.
Pacienta ievaddati: name: Jack, surname: Black, gender: Male, email: jack@black.com , password: jack1234, password(repeated): jack12, phone	Kļūdas paziņojums par to, ka ievadītās paroles nesakrīt. Toast tipa paziņojums par to, ka lietotāja reģistrācija sistēmā ir neveiksmīga.	Jā	19.03.2016.

number: +371 29028980			
--------------------------	--	--	--

3.3.3. Pacienta autentifikācija sistēmā

Ievaddati	Sagaidāmais rezultāts	Rezultāts (Jā/ nē)	Datums
Pacienta autentifikācijas ievaddati: email: jack@black.com , parole: jack1111	Toast tipa paziņojums par nepareizi ievadītu paroli.	Jā	20.03.2016.
Pacienta autentifikācijas ievaddati: email: jack@black.com , parole: jack1234	Pieslēgšanās sistēmai, parādot Toast tipa paziņojumu par veiksmīgu pieslēgšanos sistēmai.	Jā	20.03.2016.
Pacienta autentifikācijas ievaddati: email: jackjack@black.com , parole: jack1234	Toast tipa paziņojums par neeksistējošu e-pasta adresi.	Nē	20.03.2016.
Pacienta autentifikācijas ievaddati: email: jackjack@black.com , parole: jack1234	Toast tipa paziņojums par neeksistējošu e-pasta adresi.	Jā	20.03.2016.

3.3.4. Pacienta navigācijas izvēlnes izveide

Ievaddati	Sagaidāmais rezultāts	Rezultāts (Jā/ nē)	Datums
Pareizi ievadīti pacienta autentifikācijas dati.	Pieslēgšanās sistēmai, attēlojot navigācijas izvēlni ar pacientam paredzētajiem elementiem	Nē	20.03.2016.
Pareizi ievadīti pacienta autentifikācijas dati.	Pieslēgšanās sistēmai, attēlojot navigācijas izvēlni ar pacientam paredzētajiem elementiem	Nē	23.03.2016.
Pareizi ievadīti pacienta autentifikācijas dati.	Pieslēgšanās sistēmai, attēlojot navigācijas izvēlni ar pacientam paredzētajiem elementiem	Jā	23.03.2016.

3.3.5. Ārsta reģistrācija

Ievaddati	Sagaidāmais rezultāts	Rezultāts (Jā/ nē)	Datums
Ārsta ievaddati: name: Bob, surname: Green, email: bob@bob.lv , password: bob123, password (repeated): bob123, phone_number: +371 29089783	Kļūdas paziņojumi par lietotāja reģistrācijas apstiprināšanu, sistēmas pieslēgšanās skata attēlošana. Lietotāja datu saglabāšana datubāzē.	Nē	26.03.2016.
Ārsta ievaddati: name: Bob, surname: Green, email:	Kļūdas paziņojumi par lietotāja reģistrācijas apstiprāšanu, sistēmas	Jā	26.03.2016.

<p>bob@bob.lv, password: bob123, password (repeated): bob123, phone_number: +371 29089783</p>	<p>pieslēgšanās skata attēlošana. Lietotāja datu saglabāšana datubāzē.</p>		
<p>Ārsta ievaddati: name: Bob, surname: Green, email: bob@bob.lv, password: bob123, password (repeated): bob123, phone_number: +371 29089783</p>	<p>Kļūdas paziņojumi par lietotāja reģistrācijas apstiprāšanu, sistēmas pieslēgšanās skata attēlošana.</p>	Nē	26.03.2016.
<p>Ārsta ievaddati: name:., surname: Green, email: bob@bob.lv, password: bob123, password (repeated): bob123, phone_number:</p>	<p>Kļūdas paziņojumi par vārda un telefona numura lauku neaizpildi. Toast tipa paziņojums par neveiksmīgu lietotāja reģistrāciju.</p>	Jā	26.03.2016.
<p>Ārsta ievaddati: name: Bob, surname: Green, email: bob@bob.lv, password: bob123, password (repeated): bob13, phone_number:</p>	<p>Kļūdas paziņojumi par telefona numura lauka neaizpildi un paroļu lauku nesakritību. Toast tipa paziņojums par neveiksmīgu lietotāja reģistrāciju.</p>	Jā	26.03.2016.

3.3.6. Ārsta autentifikācija sistēmā

Ievaddati	Sagaidāmais rezultāts	Rezultāts (Jā/ nē)	Datums
Ārsta ievaddati: epasts: bob@bob.lv , password: bob123	Lietotāja pieslēgšana sistēmai un Toast tipa paziņojums par to.	Jā	09.04.2016.
Ārsta ievaddati: epasts: bobby@bob.lv , password: bob1223	Kļūdas paziņojums par neeksistējošu e-pasta adresi.	Jā	09.04.2016.
Ārsta ievaddati: epasts: bob@bob.lv , password: bob1223	Kļūdas paziņojums par nepareizi ievadītu paroli.	Jā	09.04.2016.

3.3.7. Ārsta navigācijas izvēlnes izveide

Ievaddati	Sagaidāmais rezultāts	Rezultāts (Jā/ nē)	Datums
Pareizi ievadīti ārsta autentifikācijas dati: email: bob@bob.lv , password: bob123	Pieslēgšanās sistēmai, attēlojot navigācijas izvēlni ar ārstam paredzētajiem elementiem.	Jā	10.04.2016.
Pareizi ievadīti ārsta autentifikācijas dati: test@t.lv , password: test123	Pieslēgšanās sistēmai, attēlojot navigācijas izvēlni ar ārstam paredzētajiem elementiem.	Jā	10.04.2016.

3.3.8. Pacienta pievienošana

Ievaddati	Sagaidāmais rezultāts	Rezultāts (Jā/ nē)	Datums
john@john.lv (datubāzē neeksistējoša pacienta e-pasta adrese)	Toast tipa paziņojums par neeksistējošu pacientu.	Jā	16.04.2016.
jack@black.com (e-pasta adrese, kam jau ir pievienots ārsts)	Toast tipa paziņojums par to, ka pacientam jau ārsts ir pievienots	Nē	16.04.2016.
jack@black.com (pacienta e-pasta adrese, kam jau ir pievienots ārsts)	Toast tipa paziņojums par to, ka pacientam jau ārsts ir pievienots	Jā	16.04.2016.
test@t.lv (pacienta e-pasta adrese, kurai nav pievienots ārsts)	Toast tipa paziņojums par veiksmīgi pievienotu pacientu.	Jā	16.04.2016.

3.3.9. Pievienoto pacientu attēlošana sarakstā

Ievaddati	Sagaidāmais rezultāts	Rezultāts (Jā/ nē)	Datums
Ārsts bez pacientiem	Tukšs saraksts	Jā	17.04.2016.
Ārsts ar pievienotiem pacientiem	Aizpildīts saraksts ar ārsta pacientiem	Jā	17.04.2016.

3.3.10. Rehabilitācijas norādījumu uzstādīšana

Ievaddati	Sagaidāmais rezultāts	Rezultāts (Jā/ nē)	Datums
Atlasītais pacients: jack@black.com , duration: 30, frequency: 3, mazimal_angle: 90, optimal_angle: 69 (pievienošana pirmo reizi)	Toast tipa paziņojums par to, ka rehabilitācijas nosacījumi ir pievienoti un veikta to pievienošana pacientam datubāzē.	Jā	23.04.2016.
Atlasītais pacients: jack@black.com , duration: , frequency: 3, mazimal_angle: 90, optimal_angle: 69	Kļūdas paziņojums laukā duration, par to, ka tas ir obligāti jāaizpilda.	Jā	23.04.2016.
Atlasītais pacients: jack@black.com , duration: , frequency: 3, mazimal_angle: 90, optimal_angle: 69 (atkārtota rehabilitācijas nosacījumu pievienošana pacientam, kam tie jau ir pievienoti)	Toast tipa kļūdas paziņojums par to, ka rehabilitācijas nosacījumi šim pacientam jau ir pievienoti.	Jā	23.04.2016.

3.3.11. Rehabilitācijas nosacījumu dzēšana

Ievaddati	Sagaidāmais rezultāts	Rezultāts (Jā/ nē)	Datums
Izvēlēts pacients (john@billy.com), kuram ir pievienoti rehabilitācijas nosacījumi	Datu dzēšana no datubāzes un Toast tipa paziņojums par to, ka dati ir dzēsti.	Jā	23.04.2016.
Izvēlēts pacients (jack@black.com), kuram ir pievienoti rehabilitācijas nosacījumi	Datu dzēšana no datubāzes un Toast tipa paziņojums par to, ka dati ir dzēsti.	Jā	23.04.2016.

3.3.12. Rehabilitācijas veikšana un reāllaika datu aplūkošana

Ievaddati	Sagaidāmais rezultāts	Rezultāts (Jā/ nē)	Datums
Sistēmai pieslēdzies pacients, kuram ārsts nav pievienojis rehabilitācijas nosacījumus.	Atverot Rehab fragmentu, visas pogas un komentāra ievades lauks ir lauks ir atspējots. Parādās Toast tipa paziņojums par to, ka pacientam nav pievienoti rehabilitācijas nosacījumi.	Jā	07.05.2015.
Sistēmai pieslēdzas pacients, kuram ārsts ir pievienojis rehabilitācijas nosacījumus	Atverot Rehab fragmentu, tiek iespējota poga "START", rehabilitācijas laika atskaites logrīkā tiek ielādēts rehabilitācijas ilgums no datubāzes.	Jā	07.05.2015.

Sistēmai pieslēdzas pacients, kuram ārsts ir pievienojis rehabilitācijas nosacījumus, nospiež pogu "START"	Sākas laika atskaite, tiek iespējota "STOP" poga.	Jā	07.05.2015.
Sistēmai pieslēdzas pacients, kuram ārsts ir pievienojis rehabilitācijas nosacījumus, nospiež pogu "START", pēc tam nospiežot pogu "STOP"	Laika atskaite tiek apturēta, tiek iespējotas pogas "RESET" un "FINISH"	Jā	07.05.2015.
Sistēmai pieslēdzas pacients, kuram ārsts ir pievienojis rehabilitācijas nosacījumus, nospiež pogu "START", pēc tam nospiežot pogu "STOP", pēc tam nospiežot pogu "RESET"	Tiek atiestatīti rehabilitācijas sesijas veikšanas dati. Laika atskaites logrīkā ielādēta pilnais rehabilitācijas laiks, iespējota poga "START", citas atspējotas.	Jā	07.05.2015.
Sistēmai pieslēdzas pacients, kuram ārsts ir pievienojis rehabilitācijas nosacījumus, nospiež pogu	Tiek parādīts vaicājuma dialogs, ar jautājumu, vai pacients vēlas beigt rehabilitācijas sesiju.	Jā	07.05.2015

<p>“START”, pēc tam nospiežot pogu “STOP”, tiek ierakstīts komentārs, pēc tam nospiežot pogu “FINISH”</p>			
<p>Sistēmai pieslēdzas pacients, kuram ārsts ir pievienojis rehabilitācijas nosacījumus, nospiež pogu “START”, pēc tam nospiežot pogu “STOP”, tiek ierakstīts komentārs, pēc tam nospiežot pogu “FINISH”, izvēles dialogā nospiežot “OK”</p>	<p>Rehabilitācijas sesijas dati tiek saglabāti datubāzē, pacientam parādot Toast tipa paziņojumu par aizvadītās sesijas veiksmīgu pabeigšanu.</p>	Jā	08.05.2015
<p>Sistēmai pieslēdzas pacients, kuram ārsts ir pievienojis rehabilitācijas nosacījumus, nospiež pogu “START”, pēc tam nospiežot pogu “STOP”, tiek ierakstīts komentārs, pēc tam nospiežot pogu “FINISH”,</p>	<p>Vaicājuma dialogs pazūd un pacientam ir iespēja mainīt sesijas komentāru, atiestatīt sesijas datus, vai atkārtoti beigt sesiju, spiežot pogu “FINISH”.</p>	Jā	08.05.2015

izvēles dialogā nospiežot “Cancel”			
---------------------------------------	--	--	--

3.3.13. Rehabilitācijas vēstures attēlošana

Ievaddati	Sagaidāmais rezultāts	Rezultāts (Jā/ nē)	Datums
Pieslēdzoties sistēmai kā pacients bez rehabilitācijas sesiju vēstures	Tiek parādīts Toast tipa paziņojums par to, ka pacientam nav rehabilitācijas sesiju vēstures	Nē	14.05.2016.
Pieslēdzoties sistēmai kā pacients bez rehabilitācijas sesiju vēstures	Tiek parādīts Toast tipa paziņojums par to, ka pacientam nav rehabilitācijas sesiju vēstures	Jā	14.05.2016.
Pieslēdzoties sistēmai kā pacients ar rehabilitācijas sesiju vēsturi	Pacientam rehabilitāciju vēsture tiek ielādēta un attēlota saraksta veidā.	Jā	14.05.2016.

3.3.14. Ārsta rehabilitācijas norādījumu attēlošana

Ievaddati	Sagaidāmais rezultāts	Rezultāts (Jā/ nē)	Datums
Pieslēdzoties sistēmai kā pacients bez pievienotiem rehabilitācijas nosacījumiem.	Rehabilitācijas nosacījumu skatā tiek attēloti tikai aiļu virsraksti, informācijas lauki ir tukši. Parādās Toast tipa paziņojums par to, ka rehabilitācijas nosacījumi pacientam nav pievienoti.	Jā	15.05.2016.
Pieslēdzoties sistēmai kā pacients	Rehabilitācijas nosacījumu skatā visos laukos tiek	Jā	15.05.2016.

ar ārsta definētiem rehabilitācijas nosacījumiem.	ielādēta informācija no datubāzes, attēlojot tos pacientam.		
---	---	--	--

3.3.15. Pievienoto pacientu rehabilitācijas vēstures aplūkošana

Ievaddati	Sagaidāmais rezultāts	Rezultāts (Jā/ nē)	Datums
Pieslēdzoties sistēmai kā ārsts, kuram nav pievienots neviens pacients	Tiek parādīts Toast tipa paziņojums par to, ka ārstam nav neviena pievienota pacienta	Jā	21.05.2016.
Pieslēdzoties kā ārsts, kam ir pievienoti pacienti	Ielādējas pacientu saraksts	Jā	21.05.2016.
Pieslēdzoties kā ārsts, kam ir pievienoti pacienti, izvēloties no saraksta pacientu, kam nav rehabilitācijas sesiju vēstures	Saraksta vietā netiek nekas attēlots, parādās Toast tipa paziņojums par to, ka pacientam nav rehabilitācijas sesiju vēstures.	Jā	21.05.2016.
Pieslēdzoties kā ārsts, kam ir pievienoti pacienti, izvēloties no saraksta pacientu, kam ir rehabilitācijas sesiju vēstures	No datubāzes ielādējas saraksts ar visām atlasītā pacienta rehabilitācijas sesijām.	Nē	21.05.2016.
Pieslēdzoties kā ārsts, kam ir	No datubāzes ielādējas saraksts ar visām atlasītā	Jā	21.05.2016.

pievienoti pacienti, izvēloties no saraksta pacientu, kam ir rehabilitācijas sesiju vēstures	pacienta rehabilitācijas sesijām.		
--	-----------------------------------	--	--

3.3.16. Pacienta paroles maiņa

Ievaddati	Sagaidāmais rezultāts	Rezultāts (Jā/ nē)	Datums
Old password: jack123 (nepareiza), New password: jacky12, New password (repeated): jacky12	Toast tipa paziņojums par to, ka vecā parole nav pareiza.	Jā	28.05.2016.
Old password: jack123 (pareiza), New password: jacky11, New password (repeated): jacky12	Toast tipa paziņojums par to, jaunieievadītās paroles nesakrīt.	Jā	28.05.2016.
Old password: jack123 (pareiza), New password: jacky12, New password (repeated): jacky11	Toast tipa paziņojums par to, jaunieievadītās paroles nesakrīt.	Jā	28.05.2016.
Old password: jack123 (pareiza), New password: jacky12,	Toast tipa paziņojums par to, parole maiņa ir veikta veiksmīgi. Vecā parole datubāzē tik aizstāta ar jauno.	Jā	28.05.2016.

New password (repeated): jacky12			
-------------------------------------	--	--	--

3.3.17. Ārsta paroles maiņa

Ievaddati	Sagaidāmais rezultāts	Rezultāts (Jā/ nē)	Datums
Old password: bob1233 (nepareiza), New password: bobby12, New password (repeated): bobby12	Toast tipa paziņojums par to, ka vecā parole nav pareiza.	Jā	28.05.2016.
Old password: bob123 (nepareiza), New password: bobby122, New password (repeated): bobby12	Toast tipa paziņojums par to, jaunievadītās paroles nesakrīt.	Jā	28.05.2016.
Old password: bob123 (nepareiza), New password: bobby12, New password (repeated): bobby122	Toast tipa paziņojums par to, ka vecā parole nav pareiza.	Jā	28.05.2016.
Old password: bob123 (nepareiza), New password: bobby12, New password (repeated): bobby12	Toast tipa paziņojums par to, parole maiņa ir veikta veiksmīgi. Vecā parole datubāzē tik aizstāta ar jauno.	Jā	28.05.2016.

3.3.18. Pacienta konta dzēšana

Ievaddati	Sagaidāmais rezultāts	Rezultāts (Jā/ nē)	Datums
Pacienta kots, pogas "DELETE ACCOUNT" nospiešana	Pacienta konta dzēšana no datubāzes	Jā	28.05.2016.
Pacienta kots, pogas "DELETE ACCOUNT" nospiešana	Pacienta konta dzēšana no datubāzes	Jā	28.05.2016.

3.3.19. Ārsta konta dzēšana

Ievaddati	Sagaidāmais rezultāts	Rezultāts (Jā/ nē)	Datums
Ārsta kots, pogas "DELETE ACCOUNT" nospiešana	Ārsta konta dzēšana no datubāzes	Jā	28.05.2016.
Ārsta kots, pogas "DELETE ACCOUNT" nospiešana	Ārsta konta dzēšana no datubāzes	Jā	28.05.2016.

5. PROJEKTA ORGANIZĀCIJA

Projekta organizācija tika izvēlēta iteratīvās izstrādes pieeja. Sākumā bija skaidri definēti pamatuzdevumi un pirmās izstrādājamās metodes, taču nebija pilna izstrādājamo metožu saraksta, tāpēc tas tika papildināts izstrādes procesa laikā.

Produkta izstrādes iterācijas ilgums bija viena nedēļa. Katras nedēļas beigās notika virtuāla vai fiziska tikšanās ar pasūtītāju, atskaitoties par paveikto, pārrunājot funkciju realizāciju, optimizāciju un plānojot nākošajā iterācijā veicamos darbus.

Produkta testēšana tika veikta gan no izstrādātāja, gan no pasūtītāja puses. Visu metožu implementācijas laikā un pēc to izstrādes tika veikti akcepttesti.

6. DARBIETILPĪBAS NOVĒRTĒJUMS

Koda rindiņu kopskaita novērtēšanai tika izmantots *Android Studio* spraudnis “*Statistic*”. No tā iegūtajiem datiem tika secināts, ka tika uzrakstītas 2757 Java un 1512 XML failu rindiņas.

Darbietilpības vērtēšanai tika izmantots interneta rīks “*COCOMO II - Constructive Cost Model*”. Aprēķinot koda rindiņu skaitu un atlasot kritēriju nozīmīgumu, tika aprēķināts darbietilpības apjoms – 3.8 personmēneši. Realitātē sistēmas izstrāde aizņēma 3 personmēnešus.

Kritēriju novērtējums:

Kritērijs	Novērtējums
Izstrādes sarežģītība	Vidējs
Produkta uzticamība	Vidējs
Datu bāzes izmērs	Vidējs
Produkta sarežģītība	Vidējs
Programmētāja prasmes	Zems
Platformas pieredze	Zems
Programmēšanas valodas un rīku zināšanas	Zems
Laika ierobežojumi	Vidējs
Platformas mainīgums	Zems
Izstrādes otrreizēja lietojamība	Vidējs

7. KVALITĀTES NODROŠINĀŠANA

Sistēmas kvalitātes nodrošināšanai tika veiktas sekojošas darbības:

- Sistēma tika strukturēta balstoties uz Android lietotņu izstrādes pamatprincipiem;
- Programmatūras prasību specifikācijas izstrādes laikā tika izmantots standarts LVS 68:1996 programmatūras prasību specifikācijas ceļvedis;
- Programmatūras prasību apraksta laikā tika izmantota programmas programmatūras prasību specifikācija un standarts LVS 72:1996 ieteicamā prakse programmatūras projektējuma aprakstīšanai;
- Sistēmas izstrādes laikā regulāri tika veikta tās testēšana, kas palīdzēja savlaicīgi identificēt kļūdas un tās novērst;
- Sistēmas izstrādes laikā tika izmantots versiju kontroles rīks *Git*.

8. KONFIGURĀCIJU PĀRVALDĪBA

Konfigurācijas pārvaldībai tika izvēlēta *Git* versiju pārvaldības sistēma, izmantojot Latvijas Universitātes serveri. Pēc būtisku izmaiņu veikšanas pirmkodā, tās tika saglabātas attālinātā repozitorijā. Tās tika veidotas kā sistēmas versijas, ļaujot atsaukt veiktās izmaiņas vitālu sistēmas kļūdu gadījumā. Konfigurācijas pārvaldība nodrošina pirmkoda drošu glabāšanu un garantiju lokālās versijas pazušanas vai netišas dzēšanas brīdī.

9. REZULTĀTI

Darba izstrādes laikā tika izstrādāta strādājoša ceļa rehabilitācijas lietotne atbilstoši definētajām prasībām.

Sistēmas mērķis bija nodrošināt atvieglotu datu plūsmu starp ārstu un pacientu, atvieglojot rehabilitācijas procesu gan ārstam, gan viņa pacientiem. Līdz šim ārstam bija grūti sekot līdzi pacienta rehabilitācijas procesam, uzskatāmi vienuviet redzot visus rehabilitācijas datus. Skatoties no otras puses, pacientam ne vienmēr ir iespēja rehabilitācijas laikā komunicēt ar ārstu. Abas iepriekšminētās problēmas tiek atrisinātas ar izstrādātās lietotnes funkcionalitāti.

Izstrādes procesā tika veikta testēšana, lai nodrošinātu tās kvalitāti.

Lietotnes prototips tiks prezentēts “UPGRADED LIFE FESTIVAL” veselības tehnoloģiju iesācējuzņēmumu konferencē (Helsinki, Somija).

10. SECINĀJUMI

Veselības tehnoloģijas mūsdienās aizņem ar vien lielāku daļu no tehnoloģiju industrijas, tādēļ darba tēmas aktualitāte ir ļoti augta. Sistēma kalpos kā pamats ceļa locītavu rehabilitācijas analīzei ārstiem, papildus uzraugot un komunicējot ērtā veidā ar saviem pacientiem.

Potenciāli nākotnē lietotne varētu tik izstrādāta arī uz citām platformām, ne tikai Android. Papildus tā, līdz ar iegulto sensoru iekārtu varētu tikt pielāgota arī citām locītavām – elkoņu, plaukstu un potīšu.

Papildus nākotnē tiks veikta testēšana ar pacientiem un ārstiem, pārbaudot lietotnes reālo ietekmi uz rehabilitācijas procesu.

11. IZMANTOTĀS LITERATŪRAS SARAKSTS

1. A. Hermanis, R. Cacurs, K. Nesenbergs, M. Greitans, E. Syundyukov, L. Selavo, "Demo: Wearable Sensor System for Human Biomechanics Monitoring", The International Conference on Embedded Wireless Systems and Networks (EWSN), Graz, Austria 2016, pp. 247-248.
2. Darbietilpības novērtēšanas rīks *COCOMO II*
Pieejams: <http://csse.usc.edu/tools/COCOMOII.php> Pārbaudīts: 29.05.2016.
3. *Firebase* Android dokumentācija
Pieejams: <https://firebase.google.com/docs/android/setup> Pārbaudīts: 29.05.2016.
4. Android dokumentācija
Pieejams: <https://developer.android.com/guide/index.html> Pārbaudīts: 29.05.2016.
5. Android dizaina elementi
Pieejams: <https://design.google.com/icons/> Pārbaudīts: 29.05.2016.
6. Android dizaina vadlīnijas
Pieejams: <https://developer.android.com/design/index.html> Pārbaudīts: 29.05.2016.
7. *StackOverflow* forums problēmu risinājumu meklēšanai
Pieejams: <http://stackoverflow.com/>

12. PIELIKUMI

Lietotnes koda fragments: Klase DrawerActivity:

```
public class DrawerActivity extends AppCompatActivity
    implements NavigationView.OnNavigationItemSelectedListener,
    AddPatientFragment.addPatientListener, PatientListFragment.patientListListener,
    PreceptAssignFragment.preceptAssignListener, PreceptFragment.preceptListener,
    RehabFragment.rehabListener, HistoryPatientFragment.patientHistoryListener,
    DoctorHistoryFragment.DoctorHistoryListener,
    PatientViewFragment.patientViewListener,
    DoctorViewFragment.doctorViewListener{

    NavigationView navigationView = null;
    Toolbar toolbar = null;
    String email;
    String name;
    String surname;
    String gender;
    String phone;
    String addPatientEmail;
    String doctor_key;
    String precept;
    int duration_minutes = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        //Initializing Firebase
        Firebase.setAndroidContext(this);

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_drawer);
```

```

toolbar = (Toolbar) findViewById(R.id.toolbar);
setSupportActionBar(toolbar);

DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
    this, drawer, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
drawer.setDrawerListener(toggle);
toggle.syncState();

navigationView = (NavigationView) findViewById(R.id.nav_view);
navigationView.setNavigationItemSelectedListener(this);

SharedPreferences sharedPreferences = getSharedPreferences("my_preferences",
MODE_PRIVATE);
String user_type = sharedPreferences.getString("user_type", null);

switch (user_type) {
    case "patient":
        //Set the initial (first) fragment for patient
        PreceptFragment patientInitialFragment = new PreceptFragment();
        android.support.v4.app.FragmentTransaction patientFragmentTransaction =
            getSupportFragmentManager().beginTransaction();
        patientFragmentTransaction.replace(R.id.fragment_container,
patientInitialFragment);
        patientFragmentTransaction.commit();

        //Showing patients menu items
        navigationView.getMenu().findItem(R.id.nav_rehab).setVisible(true);
        navigationView.getMenu().findItem(R.id.nav_history).setVisible(true);
        navigationView.getMenu().findItem(R.id.nav_pac_precept).setVisible(true);
        navigationView.getMenu().findItem(R.id.nav_bluetooth).setVisible(true);
        navigationView.getMenu().findItem(R.id.nav_patient_view).setVisible(true);

```

```

//Getting patient info from shared preferences
email = sharedPreferences.getString("patient_email", "default@patient.com");
name = sharedPreferences.getString("patient_name", null);
surname = sharedPreferences.getString("patient_surname", null);
gender = sharedPreferences.getString("patient_gender", null);
phone = sharedPreferences.getString("patient_phone", null);
doctor_key = sharedPreferences.getString("doctor_key", "no_doctor");
precept = sharedPreferences.getString("precept", "no_precept");

//Setting menu name for patient
String pFullName = name + " " + surname;
setText(pFullName, email);
break;

case "doctor":
//Set the initial (first) fragment for patient
AddPatientFragment doctorInitialFragment = new AddPatientFragment();
android.support.v4.app.FragmentTransaction doctorFragmentTransaction =
    getSupportFragmentManager().beginTransaction();
doctorFragmentTransaction.replace(R.id.fragment_container,
doctorInitialFragment);
doctorFragmentTransaction.commit();

//Showing doctor menu items
navigationView.getMenu().findItem(R.id.nav_add_patient).setVisible(true);
navigationView.getMenu().findItem(R.id.nav_patient_list).setVisible(true);
navigationView.getMenu().findItem(R.id.nav_doc_precept).setVisible(true);
navigationView.getMenu().findItem(R.id.nav_doctor_view).setVisible(true);
navigationView.getMenu().findItem(R.id.nav_doctor_history).setVisible(true);

//Getting doctor info from shared preferences
email = sharedPreferences.getString("doctor_email", "default@patient.com");
name = sharedPreferences.getString("doctor_name", null);

```

```

        phone = sharedPreferences.getString("doctor_phone", null);
        surname = sharedPreferences.getString("doctor_surname", null);

        //Creates list for all doctor patients
        createPatientList();

        //Setting menu name for patient
        String dFullName = "Dr. " + name + " " + surname;
        setText(dFullName, email);
        break;
    default:
    }
}

public void setText(String fullName, String email) {
    NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
    navigationView.setNavigationItemSelectedListener(this);
    View header = navigationView.getHeaderView(0);
    TextView pFullName = (TextView) header.findViewById(R.id.drawer_name);
    TextView pEmail = (TextView) header.findViewById(R.id.drawer_email);
    pFullName.setText(fullName);
    pEmail.setText(email);
}

@Override
public void onBackPressed() {
    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    if (drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawer(GravityCompat.START);
    } else {
        new AlertDialog.Builder(this)
            .setTitle("Exit")
            .setMessage("Are you sure you want to exit?")
            .setPositiveButton("Yes", new DialogInterface.OnClickListener()

```

```

    {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            final Context context = getBaseContext();
            Intent intent = new Intent(context, StartActivity.class);
            startActivity(intent);
        }

    })
    .setNegativeButton("No", null)
    .show();
}
}

```

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.drawer, menu);
    return true;
}

```

```

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }
}

```

```

        return super.onOptionsItemSelected(item);
    }

    @SuppressWarnings("StatementWithEmptyBody")
    @Override
    public boolean onNavigationItemSelected(MenuItem item) {
        // Handle navigation view item clicks here.
        int id = item.getItemId();

        if (id == R.id.nav_rehab) {
            //Set the fragment initially
            RehabFragment fragment = new RehabFragment();
            android.support.v4.app.FragmentTransaction fragmentTransaction =
                getSupportFragmentManager().beginTransaction();
            fragmentTransaction.replace(R.id.fragment_container, fragment);
            fragmentTransaction.commit();
        } else if (id == R.id.nav_history) {
            HistoryPatientFragment fragment = new HistoryPatientFragment();
            android.support.v4.app.FragmentTransaction fragmentTransaction =
                getSupportFragmentManager().beginTransaction();
            fragmentTransaction.replace(R.id.fragment_container, fragment);
            fragmentTransaction.commit();
        } else if (id == R.id.nav_add_patient) {
            AddPatientFragment fragment = new AddPatientFragment();
            android.support.v4.app.FragmentTransaction fragmentTransaction =
                getSupportFragmentManager().beginTransaction();
            fragmentTransaction.replace(R.id.fragment_container, fragment);
            fragmentTransaction.commit();
        } else if (id == R.id.nav_patient_list) {
            PatientListFragment fragment = new PatientListFragment();
            android.support.v4.app.FragmentTransaction fragmentTransaction =
                getSupportFragmentManager().beginTransaction();
            fragmentTransaction.replace(R.id.fragment_container, fragment);
            fragmentTransaction.commit();
        }
    }

```

```

} else if (id == R.id.nav_pac_precept) {
    PreceptFragment fragment = new PreceptFragment();
    android.support.v4.app.FragmentTransaction fragmentTransaction =
        getSupportFragmentManager().beginTransaction();
    fragmentTransaction.replace(R.id.fragment_container, fragment);
    fragmentTransaction.commit();
} else if (id == R.id.nav_patient_view) {
    PatientViewFragment fragment = new PatientViewFragment();
    android.support.v4.app.FragmentTransaction fragmentTransaction =
        getSupportFragmentManager().beginTransaction();
    fragmentTransaction.replace(R.id.fragment_container, fragment);
    fragmentTransaction.commit();
} else if (id == R.id.nav_doctor_view) {
    DoctorViewFragment fragment = new DoctorViewFragment();
    android.support.v4.app.FragmentTransaction fragmentTransaction =
        getSupportFragmentManager().beginTransaction();
    fragmentTransaction.replace(R.id.fragment_container, fragment);
    fragmentTransaction.commit();
} else if (id == R.id.nav_doc_precept) {
    PreceptAssignFragment fragment = new PreceptAssignFragment();
    android.support.v4.app.FragmentTransaction fragmentTransaction =
        getSupportFragmentManager().beginTransaction();
    fragmentTransaction.replace(R.id.fragment_container, fragment);
    fragmentTransaction.commit();
} else if (id == R.id.nav_doctor_history) {
    DoctorHistoryFragment fragment = new DoctorHistoryFragment();
    android.support.v4.app.FragmentTransaction fragmentTransaction =
        getSupportFragmentManager().beginTransaction();
    fragmentTransaction.replace(R.id.fragment_container, fragment);
    fragmentTransaction.commit();
} else if (id == R.id.nav_bluetooth) {
    BluetoothFragment fragment = new BluetoothFragment();
    android.support.v4.app.FragmentTransaction fragmentTransaction =
        getSupportFragmentManager().beginTransaction();

```

```

        fragmentTransaction.replace(R.id.fragment_container, fragment);
        fragmentTransaction.commit();
    } else if (id == R.id.nav_share) {

    } else if (id == R.id.nav_send) {

    }

    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    drawer.closeDrawer(GravityCompat.START);
    return true;
}

//All patient list with pat_keys
ArrayList<String> allPatients = new ArrayList<>();

//All patient list with full names
ArrayList<String> fullPatientsList = getFullPatientsList(allPatients);

//Method for creating an initial patient list for doctor signed in
public void createPatientList() {
    String doc_key = email.replace(".", "");
    Firebase firebase = new Firebase("https://care-connect.firebaseio.com/doctors/"
        + doc_key + "/");

    Firebase childFirebase = firebase.child("patients");
    childFirebase.addChildEventListener(new ChildEventListener() {
        @Override
        public void onChildAdded(DataSnapshot dataSnapshot, String s) {
            String patient = (String) dataSnapshot.getValue();
            Log.v("ADDING PATIENT TO LIST", patient);
            if (!allPatients.contains(patient)) {
                allPatients.add(patient);
            }
        }
    });
}

```

```
}

@Override
public void onChildChanged(DataSnapshot dataSnapshot, String s) {
}

@Override
public void onChildRemoved(DataSnapshot dataSnapshot) {

}

@Override
public void onChildMoved(DataSnapshot dataSnapshot, String s) {

}

@Override
public void onCancelled(FirebaseError firebaseError) {

}
});
}
```

Kvalifikācijas darbs “*Android lietotne ceļa locītavu rehabilitācijai*” izstrādāts Latvijas Universitātes Datorikas fakultātē.

Ar savu parakstu apliecinu, ka darbs izstrādāts pastāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: *Austris Cīrulnieks* _____ .05.2016.

Rekomendēju darbu aizstāvēšanai

Darba vadītājs: *B.dat Emil Syundyukov* _____ .05.2016.

Recenzents: *M.dat. Pēteris Krastiņš*

Darbs iesniegts 30.05.2016.

Kvalifikācijas darbu pārbaudījumu komisijas sekretārs: *Darja Solodovņikova*

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

____.06.2016. prot. Nr. _____

Komisijas sekretārs(-e): _____