Универзитет у Београду

Електротехнички факултет



Андроид апликација за креирање тренинга са памћењем података на серверу

Дипломски рад

|  |  |
| --- | --- |
| Ментор: | Кандидат: |
| проф. др Бошко Николић | Марина Вучковић 2015/0230 |

Београд, Јун 2020.

Садржај

[Садржај i](#_Toc44283881)

[1. Увод 1](#_Toc44283882)

[2. Анализа проблема 3](#_Toc44283883)

[2.1. Неопходне функционалности 3](#_Toc44283884)

[2.2. Постојећа решења 3](#_Toc44283885)

[2.2.1. Home Workout 4](#_Toc44283886)

[2.2.2. Fitness & Bodybuilding 4](#_Toc44283887)

[2.2.3. Pro Gym Workout 4](#_Toc44283888)

[2.2.4. Fitvate 4](#_Toc44283889)

[2.2.5. Fitify 4](#_Toc44283890)

[2.2.6. Упоређивање постојећих апликација 4](#_Toc44283891)

[3. Архитектура система 6](#_Toc44283892)

[3.1. Релације између телесних вежби и мускулатуре 6](#_Toc44283893)

[3.1.1. Мишићи и мишићне групе 6](#_Toc44283894)

[3.1.2. Вежбе 7](#_Toc44283895)

[3.2. Тренинг 7](#_Toc44283896)

[3.2.1. Извођење тренинга 8](#_Toc44283897)

[3.3. Телесна маса 10](#_Toc44283898)

[3.4. Опрема 10](#_Toc44283899)

[3.5. Комуникација корисничке стране, сервера и базе 11](#_Toc44283900)

[3.6. Опис коришћених технологија 12](#_Toc44283901)

[3.6.1. Андроид апликација 12](#_Toc44283902)

[3.6.2. Сервер 13](#_Toc44283903)

[3.6.3. База података 13](#_Toc44283904)

[4. Опис рада система 14](#_Toc44283905)

[4.1. Регистрација и логовање 14](#_Toc44283906)

[4.2. Кориснички екрани 15](#_Toc44283907)

[4.3. Главни кориснички екран 16](#_Toc44283908)

[4.3.1. Преглед вежби у оквиру различитих група мишића 16](#_Toc44283909)

[4.3.2. Креирање тренинга 18](#_Toc44283910)

[4.3.3. Извођење тренинга 20](#_Toc44283911)

[4.3.4. Преглед одрађеног тренинга 21](#_Toc44283912)

[4.4. Преглед постојећих вежби у систему и додавање нових 22](#_Toc44283913)

[4.5. Евиденција телесне масе корисника 24](#_Toc44283914)

[4.6. Приказ корисникове опреме 25](#_Toc44283915)

[5. Реализација система 26](#_Toc44283916)

[5.1. Андроид апликација 26](#_Toc44283917)

[5.1.1. Апликација при покретању 26](#_Toc44283918)

[5.1.2. Повезивање Андроид и серверске апликације 27](#_Toc44283919)

[5.1.3. Слање захтева серверу 29](#_Toc44283920)

[5.1.4. Прихватање одговора од сервера 29](#_Toc44283921)

[5.1.5. База података 30](#_Toc44283922)

[5.1.6. Преусмеравање на одговарајући кориснички екран приликом покретања 33](#_Toc44283923)

[5.1.7. Регистрација и логовање 34](#_Toc44283924)

[5.2. Серверска апликација 35](#_Toc44283925)

[5.2.1. Прихватање захтева на серверу 35](#_Toc44283926)

[5.2.2. Слање серверског одговора 36](#_Toc44283927)

[5.2.3. Повезивање сервера и базе 37](#_Toc44283928)

[6. Закључак 38](#_Toc44283929)

[Литература 40](#_Toc44283930)

[Списак скраћеница 41](#_Toc44283931)

[Списак слика 42](#_Toc44283932)

[Списак табела 44](#_Toc44283933)

1. Увод

Технолошки напредак и развој одавно је престао да буде само грана индустрије, већ је у великој мери оставио значајан допринос у свим сферама живота. Ни слободно време резервисано за забаву, рекреацију и физичке активности није остало имуно на дигитализацију. Штавише, управо оно често директно диктира темпо, и намеће потребе и потражње за новитетима на конкурентском тржишту.

Модерни начин живота, друштвене мреже и широк спектар информација на интернету промовишу тренд здравог живота, балансиране исхране и жеље за телесном естетиком. Све већи број људи се опредељује за одлазак у теретану као видом спорта, управо зато што теретана не познаје старосне границе. За појединаца никада није касно да почне да се бави како својим спољашњим изгледом, тако и својим здрављем и унутрашњим стањем духа.

С ширењем фитнес индустрије све популарније опције тренирања постали су и кућни тренинзи и тренинзи у природи. Све више рекреативаца почиње да испуњава своје слободно време управо физичким активностима. Циљна група фитенс индустрије постала је већа него икада, с тенденцијом да се шири и даље. С тим у вези, и у овој грани живота технологија је нашла не само своје место већ и значајни удео.

Спектар различитих фитнес апликација доступан је корисницима широм света, било да су рекреативци или професионални спортисти. Такође, подржане су и најразноврсније опције и функционалности, од прављења плана исхране, до састављања програма тренинга. У складу с тим, конкуренција није заступљена само међу спортски брендовима, произвођачима суплемената и опреме, већ велику улогу игра и у компанијама које се боре да избију на тржиште израдом што практичнијих апликација како би својим клијентима помогле у њиховом фитнес путовању.

У складу с тим, ова теза представља имплементацију апликације која би кориснику пружила лагодно коришћење и праћење његових тренинга и напретка. Апликација је намењена персоналним уређајима, чиме се постиже комфор током самог извођења вежби, посебно узимајући у обзир то да је већини вежбача мобилни уређај свакако постао неизоставни део тренинга бар као један од извора музике. Циљ апликације је и да подржи креирање нових вежби од стране корисника. Наиме, већина апликација на тему телесног вежбања има ограничен скуп вежби, стога корисник нема слободу рађења вежби које нису подржане иницијално у оквиру апликације.

У наредним поглављима биће представљени детаљи саме апликације, као што су окружења у којима је развијана, као и скуп реализованих функционалности. Поглавље два описује шта се све подразумева да буде подржано у једној апликацији овог типа. Такође је дат преглед постојећих апликација на тржишту које су наишле на највећи одзив корисника. На самом крају направљен је пресек предности и мана поменутих апликација.

Поглавље три даје осврт на архитектуру самог система. Приложени су и објашњени делови базе која је имплементирана да би систем могао да подржи све корисничке захтеве. Такође је представљен и начин комуникације између делова система, и дат је осврт на технологије које су коришћене приликом његове израде.

Поглавље четири описује главне функционалности које су имплементиране и подржане у оквиру апликације. Приказане су и слике корисничких екрана, чиме се на најефикаснији начин стиче увид у рад целокупног система.

У поглављу пет дата је реализација система и имплементациони део. Приложени су и објашњени круцијални делови кода, као и они коју су били најизазовнији приликом израде саме апликације.

1. Анализа проблема

Коришћење фитнес апликација је без сумње прави корак ка телесном напретку, изградњи мускулатуре и подстицању доброг унутрашњег осећаја, посебно код појединаца који су на самом почетку. Штавише, апликације са квалитетним вежбама и детаљним упутствима могу не само дати иновативне идеје приликом вежбања, већ и смањити потребу за фитнес инструкторима и тренерима, што значи да вежбачи у недостатку новчаних средстава могу напредовати и сами, без стручне помоћи.

Намеће се питања које све критеријуме мора да испуњава једна фитнес апликација да би била конкурентна на тржишту. Управо ово поглавље ће се детаљније бавити односом понуда апликација и клијентске потражње.

* 1. Неопходне функционалности

Оно што свакако мора да има једна апликација овог типа је скуп вежби које ће бити понуђене кориснику. Битно је да свака вежба буде објашњена, и да се укаже на то које су циљне мишићне групе сваке вежбе. Визуелни приказ у виду видео записа или анимације извођења вежби свакако је бенефит, и један је од пресудних параметара који ће привући и задржати корисника. Мана великог броја оваквих апликација је немогућност да корисник унесе своју вежбу, те он остаје ограничен сетом иницијално изабраних вежби.

Наредна опција која мора бити подржана је груписање више вежби у тренинг. Мање искусним корисницима засигурно више одговара да вежбе буду унапред одабране и задате, док оне који се дуже баве вежбањем не треба ускратити за креативност и слободу одабира вежби које ће изводити.

Опције попут памћења понављања и оптерећења у оквиру серије, као и броја серија које се изводе у току саме вежбе несумњиво спадају у основне. Неизоставни део система је и тајмер за мерење паузе између вежби. Памћење датума када је корисник радио тренинг, исто сасвим извесно спада у неопходност. Тиме се постиже увид у његов напредак и редовност.

Логовање и регистрација, као неизоставни део већине апликација, у овом случају имају својих предности и мана. Мана је неоспорно дужи процес приликом првог покретања апликације, иако на дуже стазе ова опција прераста у предност, јер даје могућност памћења свих података. Наиме, корисник може несметано да настави са коришћењем апликације и са другог уређаја.

* 1. Постојећа решења

Надаље ће бити приказане неке од фитнес апликација која су наишле на највећи одзив корисника. Биће прокоментарисане неке од функционалности због којих су управо ове апликације задовољиле потребе корисника широм света, и тиме се издвојиле на тржишту.

* + 1. Home Workout

Апликације *Home Workout* заузима прво место по броју задовољних корисника, иако она не задовољава све дате параметре. Из самог назива апликације се може видети да су циљна група њених корисника почетници и рекреативци. Њима је пресудно да постоји квалитетно објашњен скуп вежби, као и редослед по којима ће се исте изводити. Памћење оптерећења с којим су вежбе извођење такође није један од пресудних параметара, с обзиром на то већина вежбача не поседује адекватну опрему и реквизите код куће. Такође, креирање сопствених вежби и тренинга није приоритет, с обзиром на то да нису још довољно упознати са тим које варијације вежби би им највише одговарале. Имајући у виду да ова апликација има преко сто милиона преузимања, закључује се да су клијенти заинтересовани за фитнес апликације управо почетници.

* + 1. Fitness & Bodybuilding

Наредна апликација, *Fitness & Bodybuilding,* је распрострањена међу чак десет пута мање корисника у односу на апликацију *Home Workout.* Међутим, стиче се утисак да ова апликација испуњава велики број захтева, и да би у великој мери задовољила како почетнике тако и искусније у тренирању. Апликација садржи и користан графички приказ одрађених вежби по данима, где се може видети напредак по серијама у виду колико понављања је рађено и са коликим оптерећењем.

* + 1. Pro Gym Workout

Апликација *Pro Gym Workout* има преко пет милиона преузимања. Међутим, уз квалитетан видео приказ вежби, детаљна објашњења као и визуелни приказ циљних мишића сваке вежбе, апликација ипакне нуде корисницима могућност да сами креирају своје вежбе.

* + 1. Fitvate

Апликација *Fitvate* има исти недостатак као претходна апликација, уз непостојање приказа неопходног оптерећења као и тајмера током извођења вежби. Међутим, уз видео објашњење вежби завидног квалитета, ова апликација броји и преко милион задовољних корисника.

* + 1. Fitify

Апликација *Fitify* има доста сличан скуп функционалности као *Pro Gym Workout*. Иако доста идентичне по ономе што нуде, апликација *Fitify* има чак 5 пута мање преузимања. Разлог томе је највероватније што је ова апликација мање интуитивна од горе наведене. Оно што ову апликацију дефинитивно издваја из мноштва других је додавање опреме коју користи клијент, а самим тим и видео приказ вежби који корисник одабира пропраћене су селектованом опремом.

* + 1. Упоређивање постојећих апликација

На самом крају дато је и визуелно поређење постојећих решења представљено табелом 2.2.6.1. Апликације су сортиране опадајући по броју корисника који су у неком временском периоду користили исту.

Иако прављене са истом наменом и истим циљем, функционалности фитнес апликација могу међусобно доста да варирају. Стиче се утисак да не постоји идеалан скуп захтева који треба да буде имплементиран да би гарантовао велики број задовољних корисника, иако, као што је већ речено, постоји минимални скуп неопходних функционалности да би једна апликација могла да се назове фитнес апликацијом.

Табела 2.2.6.1. Упоређивање постојећих апликација

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Унос сопствене вежбе | Креирање сопственог тренинга | Памћење оптерећења  по серији | Тајмер  током вежбања |
| Home  Workout | 🗶 | 🗶 | 🗶 | ✓ |
| Fitness & Bodybuilding | ✓ | ✓ | ✓ | ✓ |
| Pro Gym Workout | 🗶 | ✓ | ✓ | ✓ |
| Fitvate | 🗶 | ✓ | 🗶 | 🗶 |
| Fitify | 🗶 | ✓ | ✓ | ✓ |

1. Архитектура система

У овоме поглављу описани су кориснички захтеви и њихова реализација. Представљена је структура самог система, начини комуникације између делова система, као и технологије које су коришћење за његову реализацију.

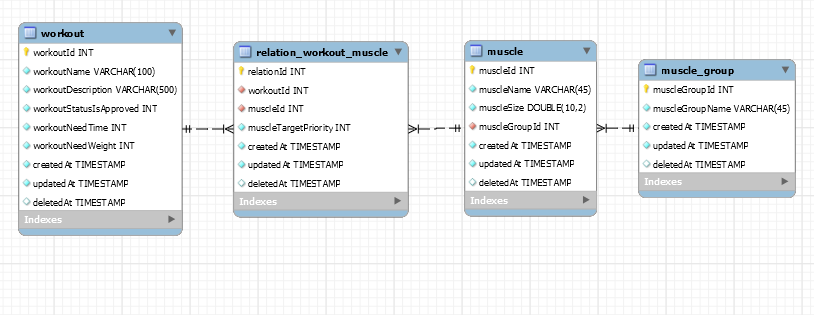
* 1. Релације између телесних вежби и мускулатуре

Као што је већ приказано у претходном поглављу, први кориснички захтев који мора да испуњава једна апликација оваквог типа је да постоји иницијални скуп вежби подржан у самом систему. Тиме се кориснику пружа могућност одабира вежби које ће да изводи. Додатно, овај систем пружа могућност кориснику да самостално уноси сопствене вежбе. На тај начин корисник не остаје ограничен сетом унапред одабраних вежби.

* + 1. Мишићи и мишићне групе

Поред скупа вежби које су понуђене кориснику, неопходно је и указати на то који телесни мишићи су циљ сваке од вежби. У складу с тим, при уносу нове телесне вежбе, неопходно је омогућити кориснику да унесе и које мишиће вежба погађа.

Да би се овакав захтев реализовао, неопходна је релација између вежбе и мишића, таква да једна вежба може да погађа више мишића, као и да исти мишић може да буде погођен од стране више вежби. Овај однос приказан је на слици 3.1.1.1. Систем је моделован тако да подржава и припадност сваког мишић једној од мишићних група. Захваљујући томе пружена је и могућност да вежбе буду груписане по мишићним групама које погађају. Стога при одлуци којој мишићној групи корисник жели да се посвети, излистају  се само вежбе за одабрану мишићну групу.

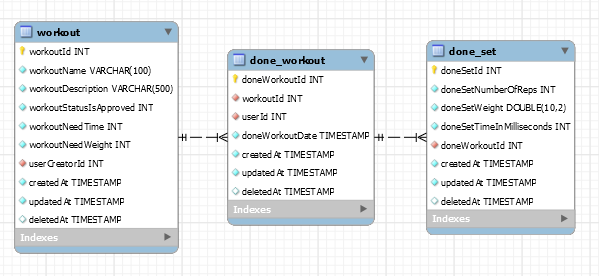


Слика 3.1.1.1. Део базе који представља релације између вежби, мишића и мишићних група

* + 1. Вежбе

Када корисник одради вежбу, пожељно је да то буде запамћено у систему, како би се несметано пратио његов напредак. За свака вежбу која је одрађена памти се датум и време када је изведена, као и број серија које су рађене. Свака серија може да има различити број понављања, као и различито оптерећење под којим је извођена. Памћење одговарајућих података омогућено је реализацијом дела базе која је приказана на слици 3.1.2.1.

Поред могућности додавања нове вежбе, подржана је и њој комплементарна операција брисања вежбе. Дакле, корисник је у могућности да брише вежбе уколико се за тим укаже потреба, али само оне које је сам креирао.



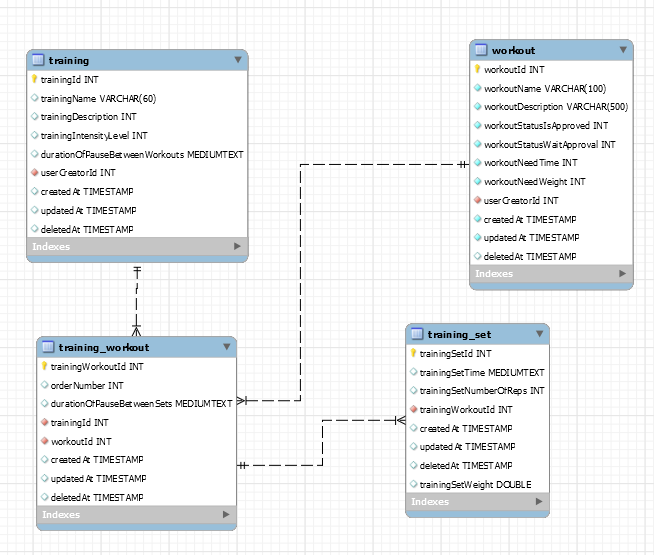
Слика 3.1.2.1. Део базе који представља однос између вежби, одрађених вежби и серија

* 1. Тренинг

Следећи кориснички захтев је креирање сопственог тренинга. Сваки тренинг треба да има свој назив и опис циља тренинга. Како се тренинг састоји из више вежби неопходно је да се сачува и информација о временској паузи између њих.

Сваки тренинг може да има више вежби, и свака вежба може да се нађе у више различитих тренинга. Да би се овај однос постигао неопходно је присуство додатне табеле између тренинга и вежби, приказано на слици 3.2.1. Такође, свака вежба може да буде задата да се изводи у више серија. Свака серија такође представља ентитет за себе, јер број понављања и оптерећење у оквиру серија може да варира.

Поред опције креирања тренинга, реализована је и комплементарна операција брисања истог.

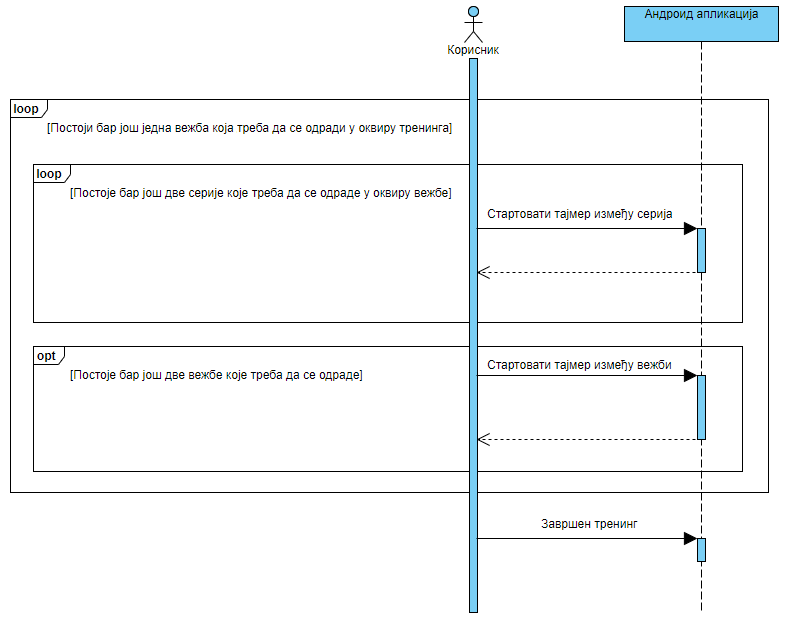


Слика 3.2.1. Део базе који представња однос између тренинга и вежби

* + 1. Извођење тренинга

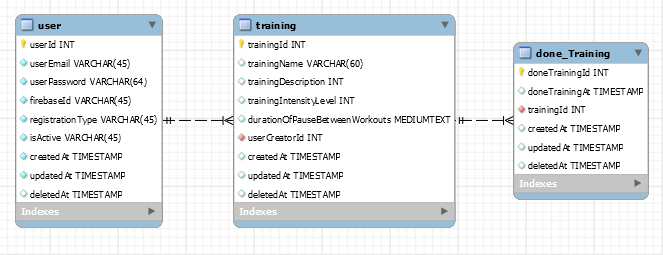
Једном када се креира тренинг, корисник може да га изводи у било које време. Тренинг је замишљен као низ одабраних вежби, са паузама између истих, које такође бира корисник при самом креирању тренинга. Свака од вежби се састоји од једне или више серија, и свакој од серија корисник је унапред задао жељени број понављања и оптерећења с којима ће је изводити, као и временске паузе између серија.

Корисник притиском на одговарајући тастер обавештава систем када је готов са извођењем једне серије. У том тренутку треба да започне пауза између серија, уколико је остала бар још једна серија у оквиру вежбе која се тренутно изводи. Ако је ипак то последња серија у оквиру вежбе, треба да започне пауза између вежби. Наравно то се дешава само уколико постоји бар још једна вежба у оквиру тренинга коју корисник треба да одради. Детаљнији приказ случајева када започиње временска пауза, као и о којој временској паузи је реч, дато је на слици 3.2.1.1.



Слика 3.2.1.1. Дијаграм секвенце за захтев стартовања тајмера

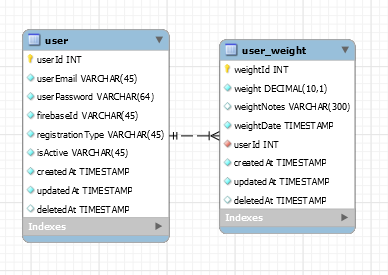
Детаљан приказ корисниковог напредовања по вежбама и серијама омогућен је, као што је већ речено, преко табела у бази података које представљају одрађену вежбу и одрађену серију. Међутим, постоји захтев да одрађени тренинзи буду приказани и по данима када је корисник тренирао. За ту потребу имплементиран је део базе са слике 3.2.1.2.



Слика 3.2.1.2. Део базе података који омогућава памћење одрађеног тренинга

* 1. Телесна маса

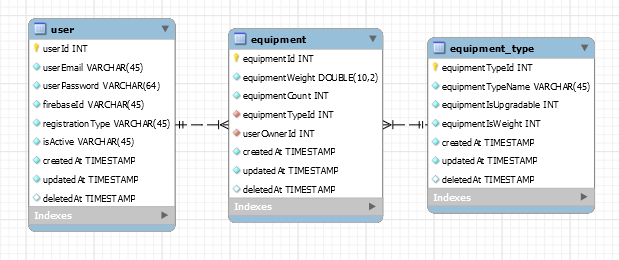
Један од додатних клијентских захтева је прикупљање података о корисниковој телесној маси. Да би се то реализовало, неопходан је део базе са слике 3.3.1. Подаци који треба да буду сачувани су датум мерења и тачна телесна маса. Опционо постоји и чување белешке, уколико корисник жели да назначи неки детаљ, рецимо да је маса измерена пре првог оброка, после тренинга и сл. Поред опције додавања телесне масе, подржана је и комплементарна операција брисања исте.



Слика 3.3.1. Део базе који омогућава вођење евиденције о корисниковој телесној маси

* 1. Опрема

Други додатни клијентски захтев је вођење евиденције о опреми коју корисник поседује. У систему су унапред дефинисани сви типови опреме који постоје, тако да се памти који тип опреме поседује корисник, у колико примерака, и колика је маса опреме уколико се ради о теговима и справама које представљају оптерећење. Детаљи реализације дати су на слици 3.4.1. Поред додавања опреме, подржана је комплементарна операције брисања.

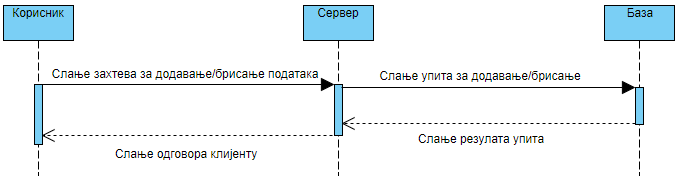


Слика 3.4.1. Део базе који омогућава вођење евиденције о опреми коју корисник поседује

* 1. Комуникација корисничке стране, сервера и базе

Сви подаци који су неопходни за извршавање апликације могу да се памте локално, у меморији корисничког уређаја. Међутим, при покретању апликације са истог налога а различитог уређаја дошло би до губитка података. Управо зато је изабрано да подаци буду памћени глобално тј. на серверском делу апликације.

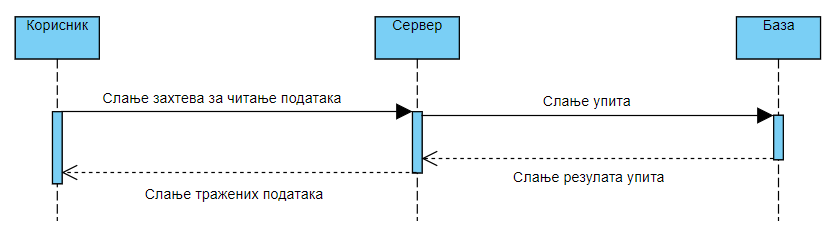
Током употребе апликације, корисник није свестан да се подаци заправо шаљу на сервер. Кориснички захтеви попут креирања/брисања вежбе, креирања/брисања тренинга, додавања/уклањања података о телесној маси, као и додавање/уклањање опреме су управо захтеви за операцијама који се заједно са одговарајућим подацима шаљу серверу, и у зависности од типа операције врши се упис или брисање из базе података. Тај вид слања захтева и чекања одговора приказан је на слици 3.5.1.



Слика 3.5.1. Дијаграм секвенце за захтев додавања/брисања података

Дакле, корисник шаље само захтеве за додавање и брисање података, али не и захтеве за читање података. При покретању апликације се ишчитају сви подаци са сервера, да би се надаље кретање кроз све функционалности апликације одвијале несметано. Међутим, и ту се разликују два случаја, у зависности да ли је корисник при последњем покретању апликације остао улогован, или се пак одјавио.

Ако се корисник нађе на страни за логовање при покретању апликације, то значи или да се никада није ни регистровао, или да с приликом последњег коришћења излогово. У том случају после регистрације одн. логовања шаљу се захтеви серверу за дохватање свих података који су релевантни за тог корисника. Та комуникација представљена је сликом 3.5.2.



Слика 3.5.2. Дијаграм секвенце за захтев читања података

Други случај је када се корисник одмах при покретању апликације нађе на главном корисничком екрану, што значи да се при последњем покретању није излогово. Тиме се постиже бржи рад, јер се од корисника не захтева да се улогује сваки пут. Сви подаци релевантни за корисника су остали у меморији персоналног уређаја, јер до њиховог брисања дође само кад се корисник одјавио. Међутим, постоји потреба да се подаци ажурирају уколико је дошло до неких промена података на серверу. Из тог разлога се и у овом случају шаље захтев серверу. Слика 3.5.2. одговара и овој комуникацији. Уколико је и дошло до промене, количина подата која ће се овим путем разменити је свакако мања од учитавања свих података. Тиме се постиже знатна уштеда у времену и боље перформансе.

Случај промене података на серверу је веома редак, јер би значило да су мењани подаци који су представљали иницијални систем, попут података о групама мишића или вежби. Предност ове могућности је свакако да уколико се укаже потреба за другачијим груписањем података, као и њиховим додавањем и мењањем, то ће бити једноставно реализовано, с обзиром на то да је омогућено да промене на серверу буду ажуриране код свих корисника апликације.

* 1. Опис коришћених технологија

У наставку је дат осврт на коришћене технологије и библиотеке захваљујући којима су олакшане реализације неких функционалности.

* + 1. Андроид апликација

Део система који директно користи клијент реализован је у окружењу *Android Studio* [5](верзија 3.6.1). Да би се клијенту пружио графички интерфејс виших перформанси, коришћени су дијалози за одабир датума и времена, као и поравнавање текста у *xml* фајловима. Да би ове и неке од наредних функционалности као и библиотека биле подржане у оквиру систему неопходно је у *build.gradle* фајл Андроид апликације додати одговарајуће зависности. Зависности које треба да буду додате, као и бенефити који оне доносе дате су табелом 3.6.1.1.

Опција регистрације путем *Google* налогa, као и регистрација и логовање пут имејл адресе омогућени су захваљујући *Firebase* платформи [6]. *Firebase* библиотека постаје доступна додавањем одговарајуће зависности у *build.gradle* фајл, што је приказано табелом 3.6.1.1.

*WebSocket API (Application Programming Interface)* је коришћен да би се омогућила комуникација између Андроид и серверске апликације. Поруке које се размењују овим путем се шаљу и примају у облику *Json* објекта.

Табела 3.6.1.1. Табела зависности које треба додати и бенефити који се тиме остварују

|  |  |
| --- | --- |
| Зависност која треба да буде додата у *build.gradle* фајл | Функционалности и библиотеке које том приликом постају доступне |
| *com.kovachcode:timePickerWithSeconds:1.0.1* [7] | Дијалог за избор времена у облику *hh:mm:ss* |
| *me.biubiubiu.justifytext:library:1.1* [8] | Поравнавање текста у *xml* фајловима |
| *com.google.firebase:firebase-auth:16.0.5* | Библиотека *com.google.firebase* коришћена за аутентикацију |
| *org.java-websocket:Java-WebSocket:1.3.0* | Библиотека *de.tavendo.autobahn* коришћена за *WebSocket* конекцију |
| *com.google.code.gson:gson:2.2.4* [9] | Библиотека *com.google.gson* коришћена за претварање објеката класа у *Json* објекатe |

* + 1. Сервер

Серверска апликација писана је у програмском језику *Java*, у развојном окружењу *IntelliJ IDEA* [10](верзија11.0.6+8-b765.25 amd64). Да би апликација ослушкивала захтеве од стране Андроид апликације на порту задате *IP (Internet Protocol)* адресе, било је неопходно да сервер буде *Spring Boot* апликација [3] (верзија2.2.6). Комуникација и повезивање серверске апликације и базе података реализовани су уз помоћ алата *Hibernate* [11](верзија 5.0.0) и конектора *MySQL Java Connector* [12] (верзија 5.1.46).

* + 1. База података

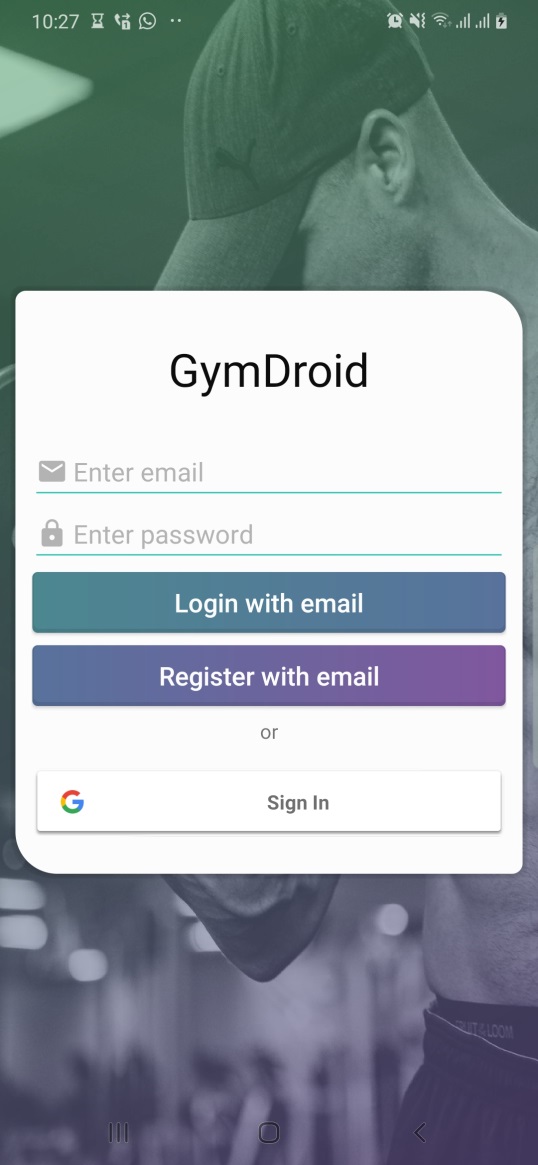
Кроз ово поглавље биле су приказане слике делова базе ради лакшег објашњења оних функционалности на којe се односе. Као што се може закључити из њих, ради се о релационој бази података. Моделовање саме базе рађено је у софтверском алату *MySQL Workbench* [4], [13] (верзија 8.0.20), који представља интегрисано развојно окружење за *MySQL Server* [14].

1. Опис рада система

У овом поглављу представљен је преглед главних функционалности подржаних у оквиру самог система. С обзиром на то да се ради о мобилној апликацији, подржан је рад и из вертикалне и из хоризонталне оријентације уређаја. Штавише, узете су у обзир и разлике у верзијама Андроида корисничких уређаја, чиме се постиже квалитетнији ефекат дизајна.

* 1. Регистрација и логовање

Почетни кориснички екран приказан је на слици 4.1.1. Корисник има могућност да приступа апликацији преко *Googlе* налога*,* или преко своје имејл адресе, уколико не поседује *Googlе* налог*.*



Слика 4.1.1. Почетни кориснички екран

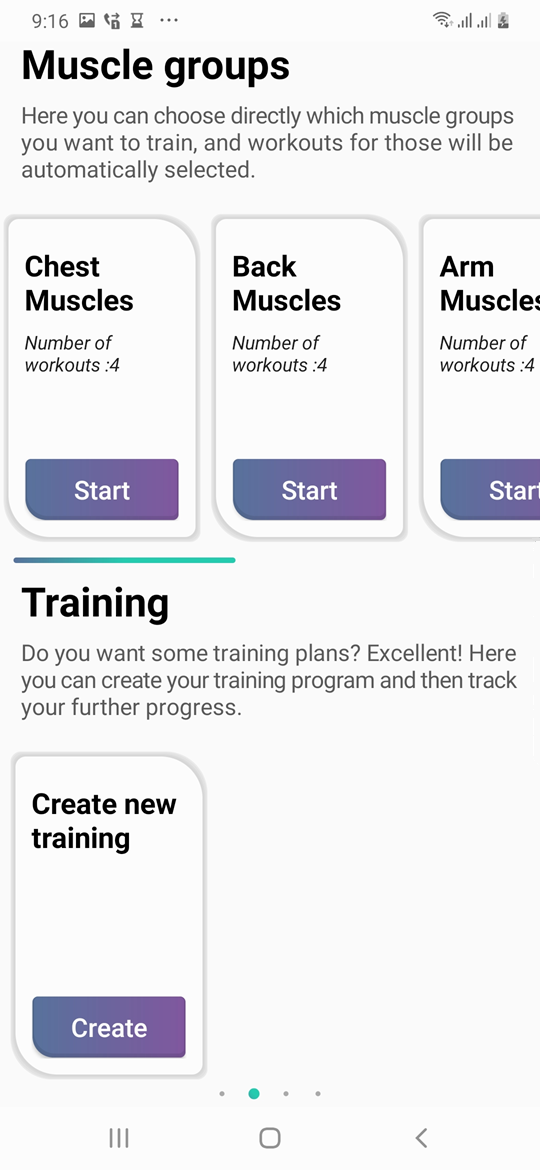
На самом почетку, уколико корисник жели да се улогује преко имејл адресе, то је омогућено уношењем адресе и шифре у одговарајућа поља. Уколико жели да се региструје преко имејл адресе, то чини притиском на дугме *Register*, која га одводи на екран за регистрацију, који садржи поља за унос имејл адресе, шифре и потврде шифре. Уколико се унесе неисправан формат имејл адресе, или некоректно поновљена шифра, исписују се одговарајуће поруке упозорења кориснику.

Као што је већ речено, корисник има могућност да се региструје одн. улогује и путем *Googla* налога, што чини притиском на дугме *Sign in*.

На самом крају, важно је напоменути да корисник једном кад се улогује у систем не мора при сваком покретању апликације то изнова да чини. Наиме, у апликацији остаје запамћено да је корисник улогован, тако да ће при сваком наредном покретању апликације бити прослеђен на главни кориснички екран са скупом свих доступних опција. Наравно, корисник има могућност и да се својевољно одјави притиском на дугме *Logout.*

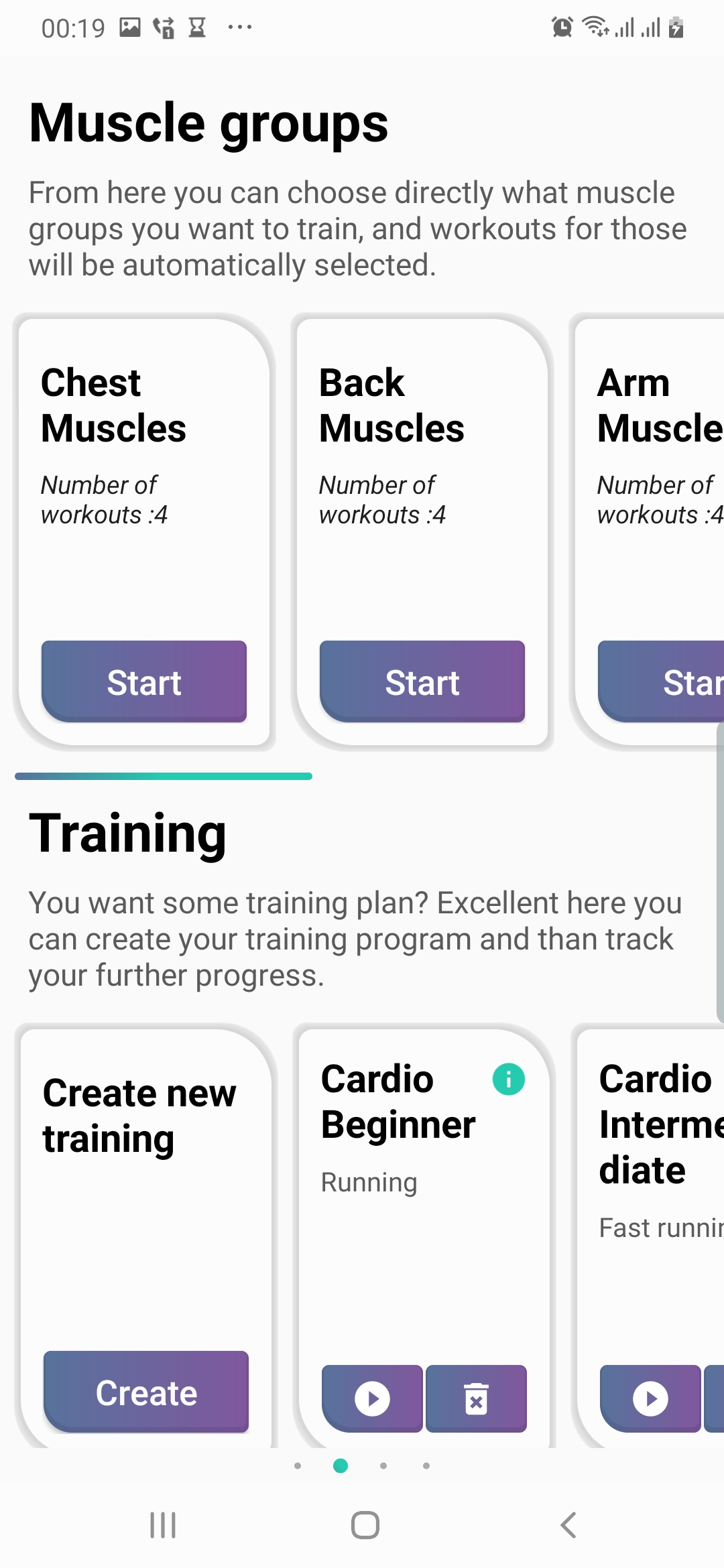
* 1. Кориснички екрани

Након успешног пријављивања на систем, кориснику се приказује главни кориснички екран представљен на слици 4.2.1.



Слика 4.2.1. Главни кориснички екран

Навигација у виду четири тачке приказана на слици 4.2.2. указује да постоји могућност померања екрана улево, као и два пута удесно. Тако се појављују и остали кориснички екрани који нуде додатне опције апликације.



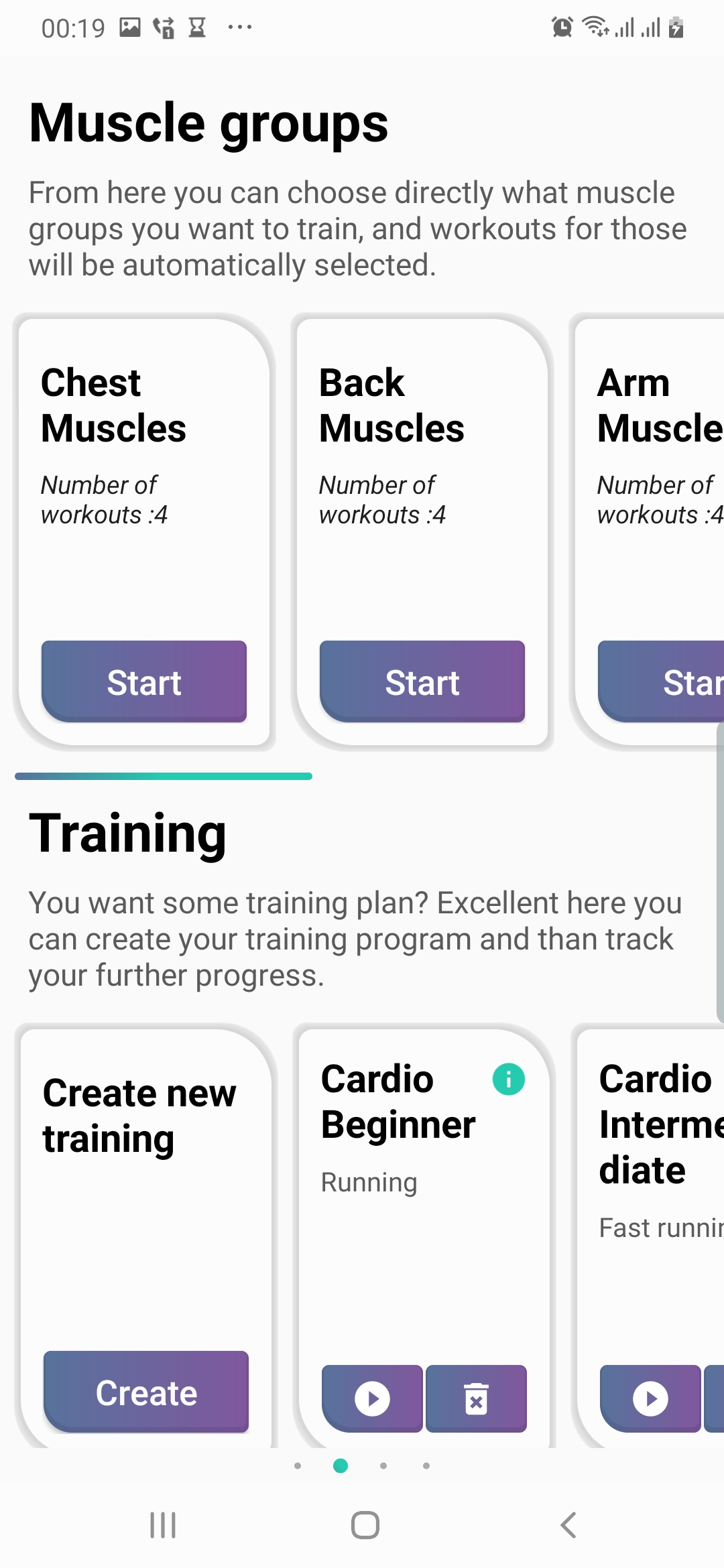
Слика 4.2.2. Навигација кроз корисничке екране

* 1. Главни кориснички екран

Као што је већ поменуто, на слици 4.2.1. је приказан главни кориснички екран. Он се састоји из два скупа функционалности. Први је везан за одабир вежби за одређену групу мишића, а други за креирање и извођење тренинга.

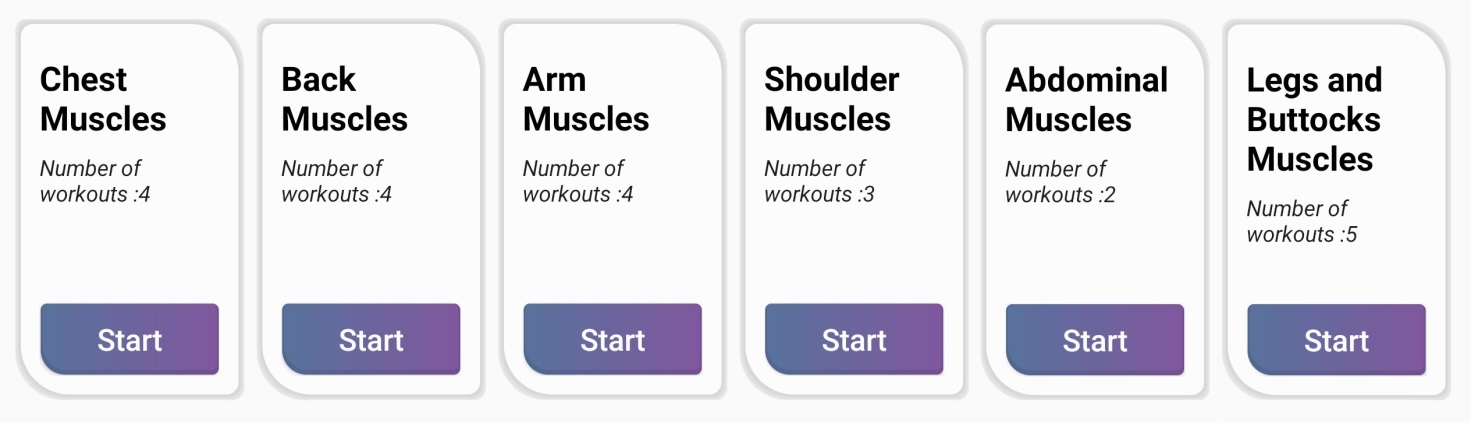
* + 1. Преглед вежби у оквиру различитих група мишића

Скролбар приказан на слици 4.3.1.1. представља заправо увећани исечак слике 4.2.1. и указује кориснику да постоји могућност померања тог дела екрана удесно, чиме се приказују све групе мишића које постоје у систему. Такође треба напоменути да би се услед већег броја креираних тренинга идентичан скролбар појавио непосредно испод картице за креирање тренинга, с истим циљем да нагласи да постоји још тренинга који су кориснику на располагању.



Слика 4.3.1.1. Скролбар

Као што је већ речено, померањем одговарајућег дела екрана удесно, кориснику се постепено приказују различите групе мишића. Приказ свих група дат је на слици 4.3.1.2. Уколико корисник још није креирао неки од својих тренинга, постоји могућност да одради вежбе понаособ. У том случају систем памти које вежбе су одрађене, у колико серија, и са коликим оптерећењем и бројем понављања је рађена свака серију засебно. Прво што корисник треба да одради у том случају је да одабере на коју групу мишића жели да се фокусира, и у складу са тим притисне дугме *Start* одговарајуће картице, што га одводи на кориснички екран приказан на слици 4.3.1.3.

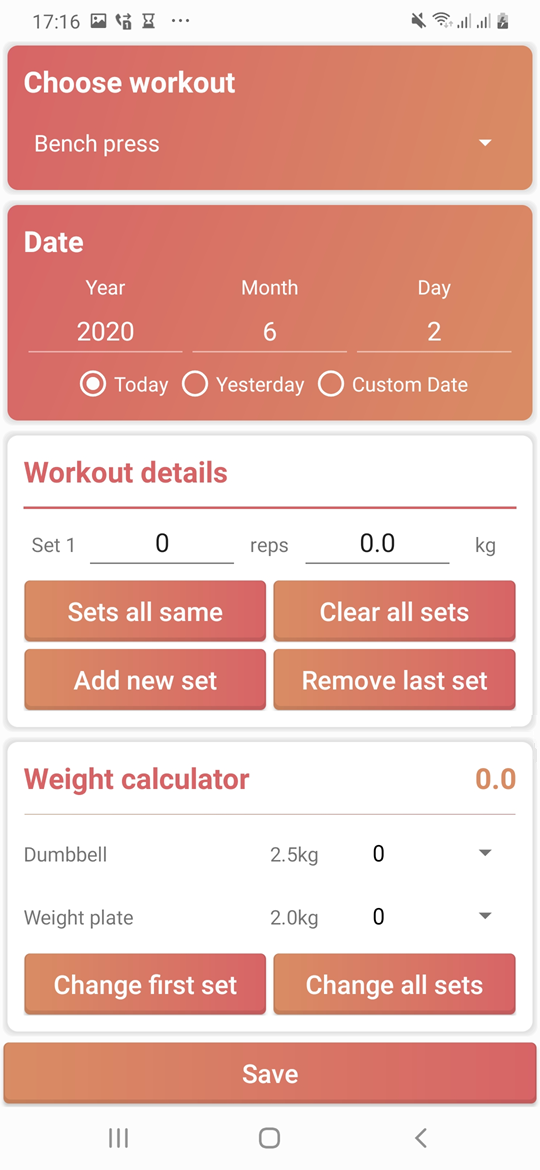


Слика 4.3.1.2. Приказ група мишића и броја вежби за сваку групу понаособ

Визуелно можемо приметити да је тај кориснички екран подељен на четири дела. Први би био одабир једне од вежби из падајућег менија за изабрану групу мишића. Други је избор датума када је вежба извођења. Ради лакшег руковања апликацијом, радио дугметом корисник може изабрати тренутни дан, или јучерашњи, с обзиром на то да је највећа вероватноћа да ће то управо бити одабир. Наравно, корисник има могућност и одабира трећег радио дугмета. Тиме се отвара цео календар и корисник може да изабере било који ранији датум. За одабир датума у будућност приказује се упозорење да не може да ради вежбе унапред.

Трећа целина на екрану представља додавање информација о томе колико има серија, колика су понављања и оптерећења у оквиру сваке серије понаособ. Дугме *Add new set* додаје нову серију, док дугме *Remove last set* брише последњу серију. Ограничење је да мора да буде изведена најмање једна серија, а највише шест. Дугме *Sets all same* ће све вредност наредних серија поставити да буду исте као код прве серије, док дугме *Clear all* поставља вредности свих серија на нулу.

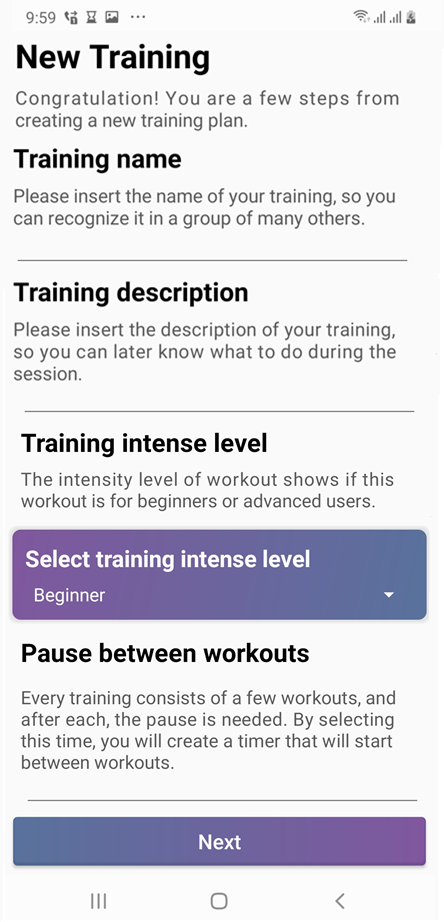
Четврти сегмент под називом *Weight calculator* олакшава кориснику унос и израчунавање оптерећења с којим изводи вежбе. У понуђеној листи дата је за сваког корисника понаособ његова опрема коју поседује, те он може да одабере шта све од опреме користи за извођење вежбе. Те масе се аутоматски сабирају и добијена вредност се уписује у у продужетку ознаке *Weight calculator.* Потом се кликом на дугме *Change first set* мења оптерећење за прву серију на ту вредност, док се притиском на дугме *Change all sets* мења оптерећење за све серије. Притиском на дугме *Save* у систему се памти одрађена вежба.



Слика 4.3.1.3. Унос података за жељену вежбу одабране групе мишића

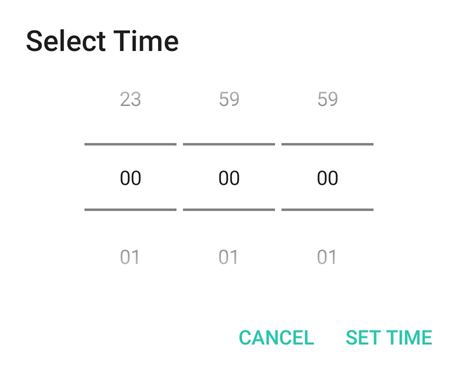
* + 1. Креирање тренинга

Креирање новог тренинга врши се притиском на дугме *Create* које се налази на доњој половини главног корисничког екрана. Тако се отвара нови кориснички дијалог приказан на слици 4.3.2.1.



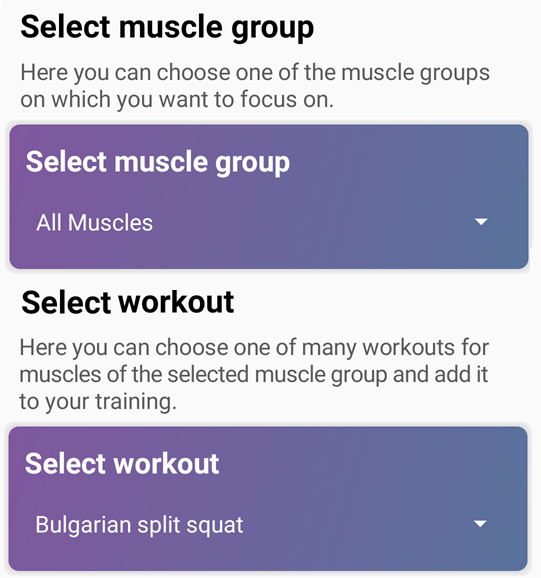
Слика 4.3.2.1. Кориснички дијалог за креирање тренинга

Од корисника се тражи да унесе име тренинга, као и опис циља и намене истог. Из падајућег менија може да се изабере интензитет тренинга тј. да ли је тренинг намењен за почетнике, средње или напредне вежбаче. На самом крају се захтева и унос временске паузе између вежби. То је реализовано отварањем дијалога за избор времена који је приказан на слици 4.3.2.2.



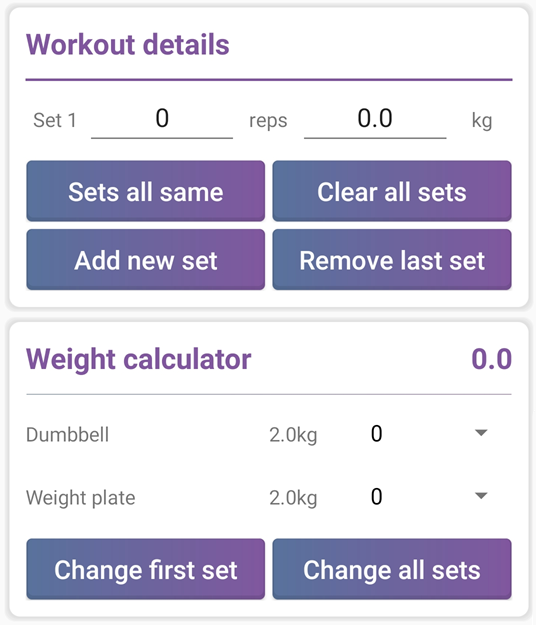
Слика 4.3.2.2. Дијалог за одабир временске паузе

Притиском на дугме *Next* отвара се дијалог за додавање вежби, чији су најбитнији делови издвојени и дати на наредним сликама. На слици 4.3.2.3. приказана су два падајућа менија. Први служи за избор жељене мишићне групе, а други се динамички прилагођава првом, нудећи само оне вежбе које погађају изабрану групу мишића. Постоји и могућност да се из првог падајућег менија одабере опција *All muscles.* Тиме у другом падајућег менију постају све вежбе доступне.



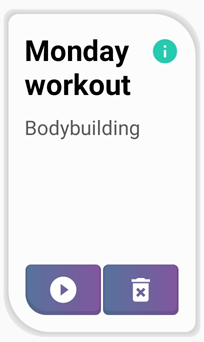
Слика 4.3.2.3. Падајући мени за одабир телесне вежбе

На слици 4.3.2.4. приказан је избор броја серија које корисник жели да ради у оквиру вежбе, као и број понављања и оптерећење у свакој серији понаособ. Такође, за сваког корисника динамички се приказује сва његова опрема у секцији *Weight calculator.* Тиме се постиже лакши одабир оптерећења које ће користити у оквиру серија. Масе одабране опреме се аутоматски сабирају, и нуди се опција да израчуната вредност оптерећења буде уписана на места које представљају масу оптерећења.



Слика 4.3.2.4. Одабир броја серија, броја понављања и масе оптерећења

Током подешавања информација о вежби, појављује се и дијалог за одабир паузе између серија, као на слици 4.3.2.2. Када сачува информације о одабраној вежби, корисник може или да дода још вежби, или да сачува тренинг. Чувањем тренинга завршава се његово креирање, и тада ће он бити додат на главном корисничком екрану, десно од картице *Create training.* Приказ ново креираног тренинга дат је на слици 4.3.2.5.

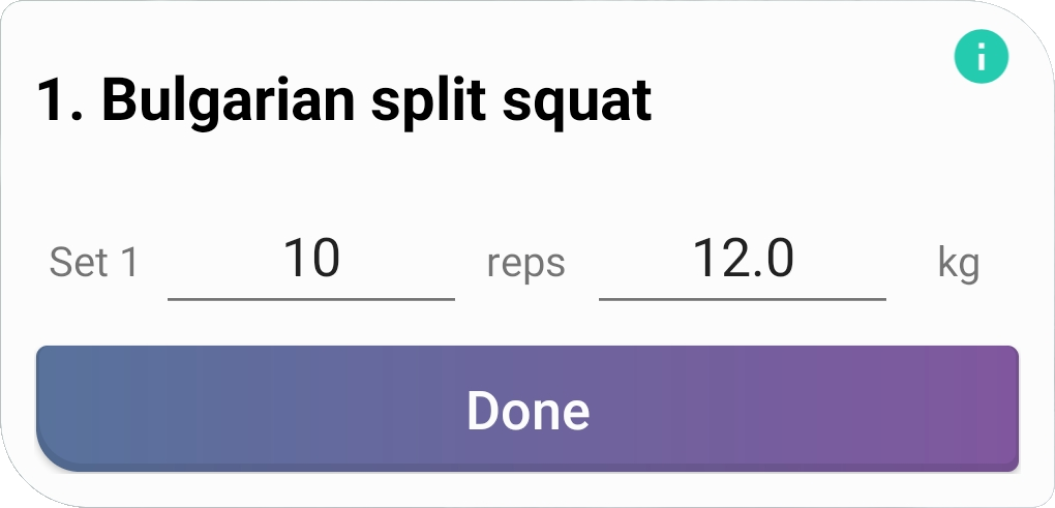


Слика 4.3.2.5. Креиран тренинг

* + 1. Извођење тренинга

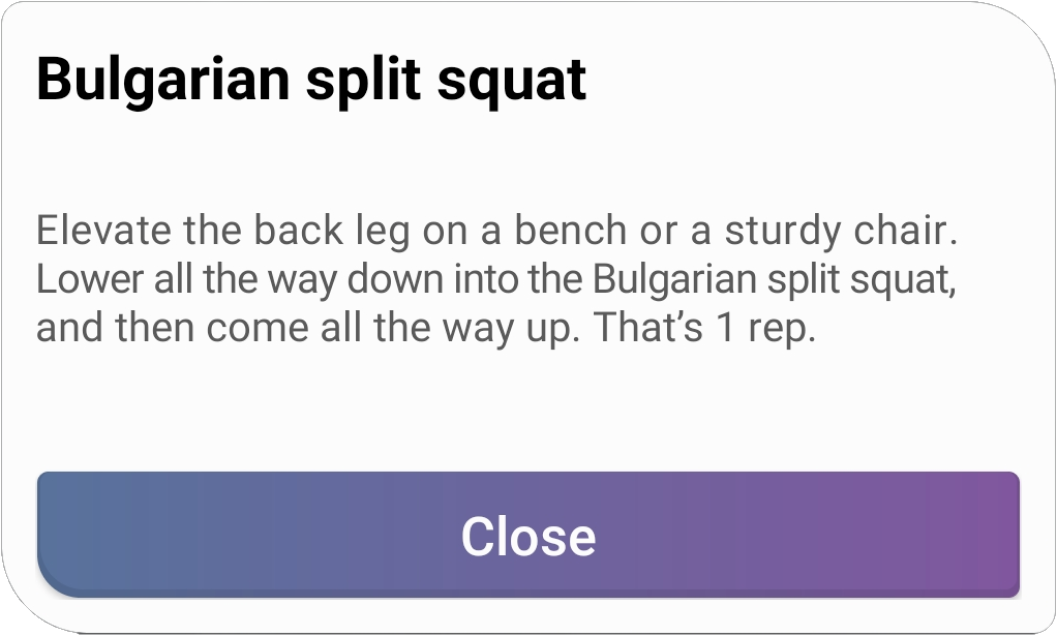
Тренинг се започиње кликом на дугме у облику знака *Play* на картици одабраног тренинга. Тиме се прелази на режим тренирања, и на корисничком екрану се одбројавају три секунде до започињања самог тренинга.

Затим се појављује дијалог као на слици 4.3.3.1. Он показује кориснику коју вежбу да ради, која је серија те вежбе у питању, као и колико понављања и са којим оптерећењем да одради тренутну серију. Уколико из неког разлога одради серију са мањим или већим бројем понављања или оптерећења него што је предвиђено, кликом на тренутне нумеричке вредности може да унесе вредности с којима је заправо одрадио серију.



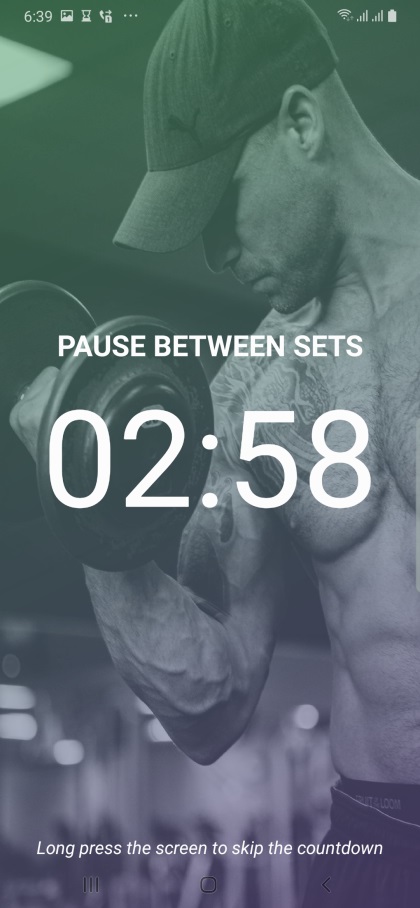
Слика 4.3.3.1. Дијалог тренутне серије и вежбе

Притиском на иконицу информација у горњем десном углу тренутног дијалога отвара се прозор као на слици 4.3.2.2. Он садржи детаље и опис вежбе која се тренутно изводи.



Слика 4.3.3.2. Опис вежбе

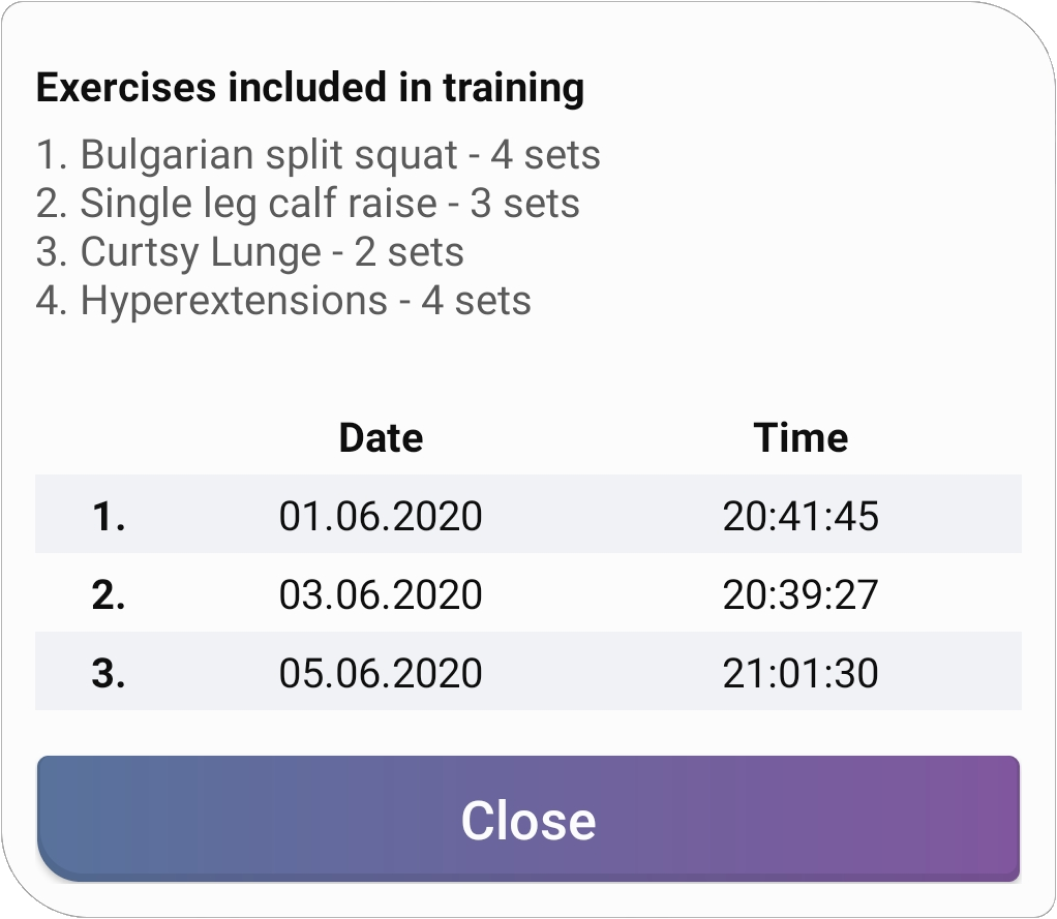
Када одради текућу серију корисник треба да притисне дугме *Done.* Тиме се активира пауза између серија, одн. пауза између вежби. Приказ екрана у том случају дат је на слици 4.3.3.3. Када истекне одбројавање, прелази се на наредну серију, одн. на наредну вежбу, уколико је то била последња серија тренутне вежбе. Тренинг се наставља у ритму смена радне серије и паузе.



Слика 4.3.3.3. Пауза између серија

* + 1. Преглед одрађеног тренинга

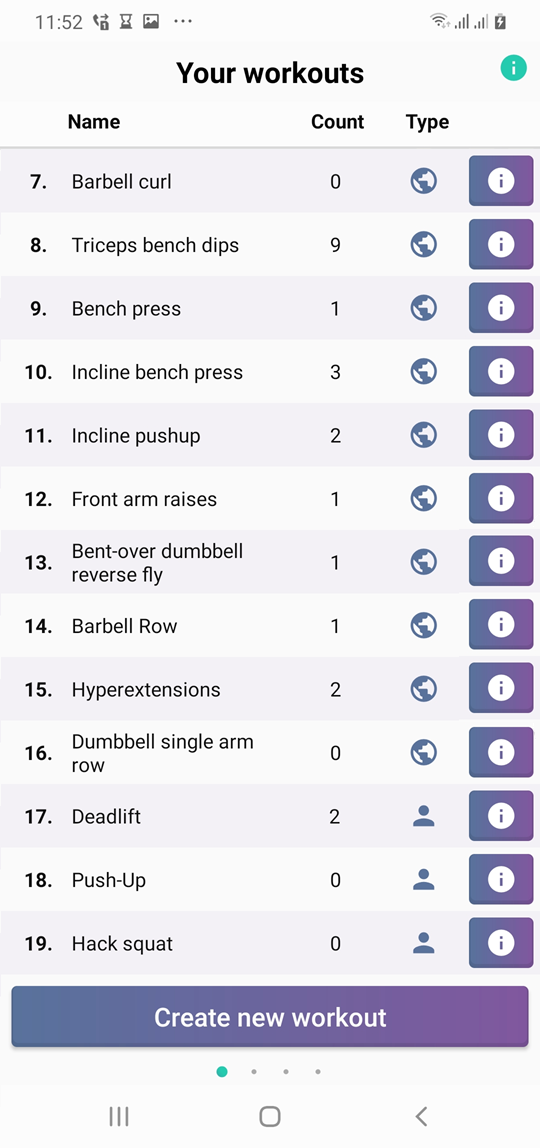
Притиском на иконицу информација у горњем десном углу картице одабраног тренинга отвара се дијалог приказан на слици 4.3.4.1. Информације које су ту садржане су редослед свих вежби у оквиру тренинга и број серија сваке вежбе. Такође се излиста и списак свих датуме и времена када је тренинг био рађен.



Слика 4.3.4.1. Преглед одрађеног тренинга

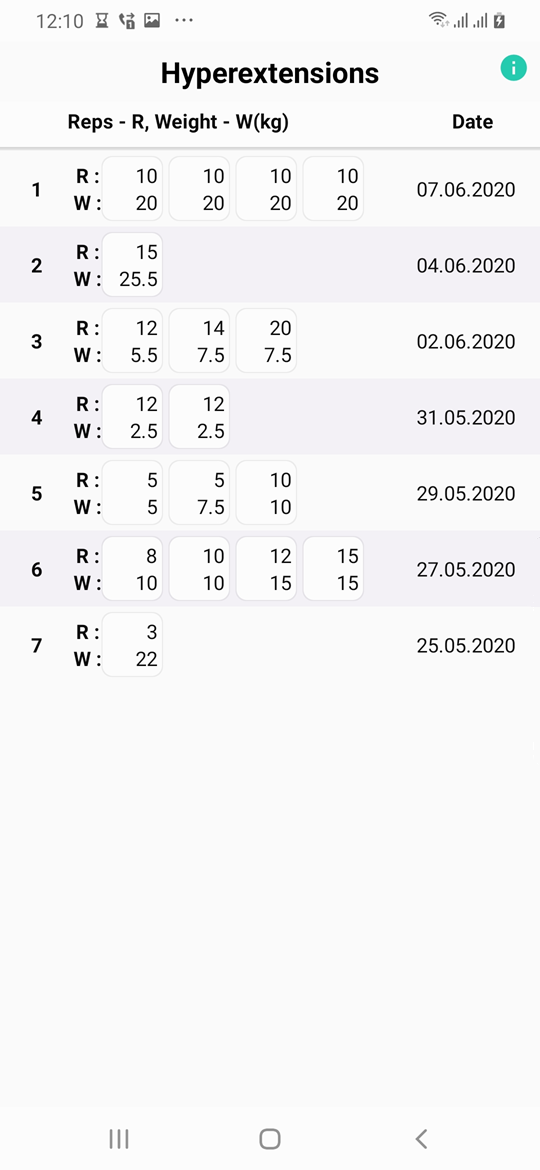
* 1. Преглед постојећих вежби у систему и додавање нових

Померањем корисничког екрана улево приказују се све вежбе доступе кориснику, као што је приказано на слици 4.4.1. У првој колони налазе се називи сваке од вежби, у другој колико пута је вежба рађена, а у трећој тип вежбе који може да буде јавни или приватни. Јавне вежбе су оне које су доступне свим корисницима, а приватне оне које је тај корисник креирао. Само приватне вежбе могу да се обришу, и то је могуће притиском на иконицу која означава да је тип вежбе приватан.



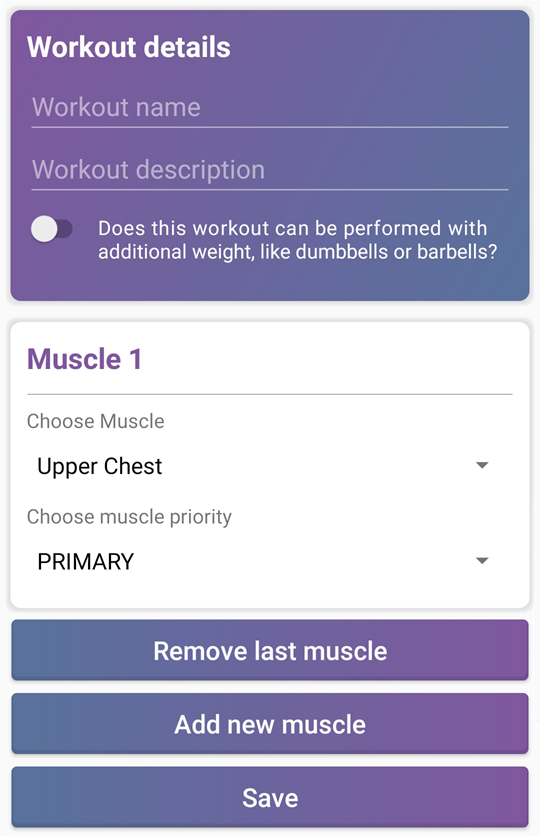
Слика 4.4.1. Списак доступних вежби

Свака вежба има информационо дугме у четвртој колони које приказује детаље свих извођења те вежбе. Притиском на то дугме отвара се прозор који одговара слици 4.4.2. Ту се приказују сва корисникова извођења изабране вежбе, сортиране по датумима. Сваки ред одговара једном извођењу вежбе, с визуелним приказом колико серија је корисник одрадио при сваком вежбању, као и колико понављања и с којим оптерећењем је одрађена свака од серија. Притиском на информационо поље које се налази у горњем десном углу отвара се дијалог као са слике 4.3.3.2. који садржи детаљан опис вежбе.



Слика 4.4.2. Детаљни приказ свих извођења вежбе

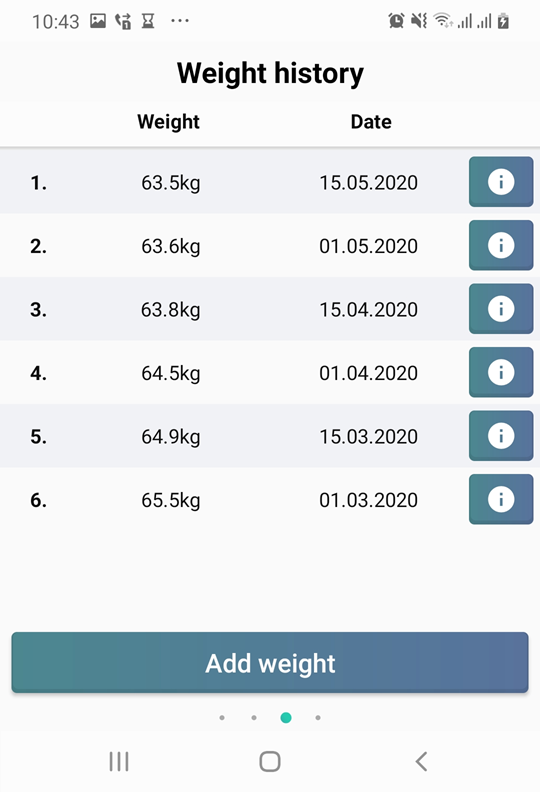
Креирање нове вежбе је омогућено притиском на дугме *Create new workout* које се налази испод листе свих вежби. Тако се отвара дијалог са слике 4.4.3. Информације које се захтевају за креирање нове вежбе су назив, опис, да ли вежба може да се ради са додатним оптерећењем као и које мишиће погађа вежба. Из првог падајућег менија се бира којим мишићима је вежба намењена, а из другог приоритет циљног мишића, јер вежба може да буде намењена већем броју мишића. Из тог разлога постоји опција додавања још циљних мишића и одређивање њиховог приоритета.



Слика 4.4.3. Додавање нове вежбе

* 1. Евиденција телесне масе корисника

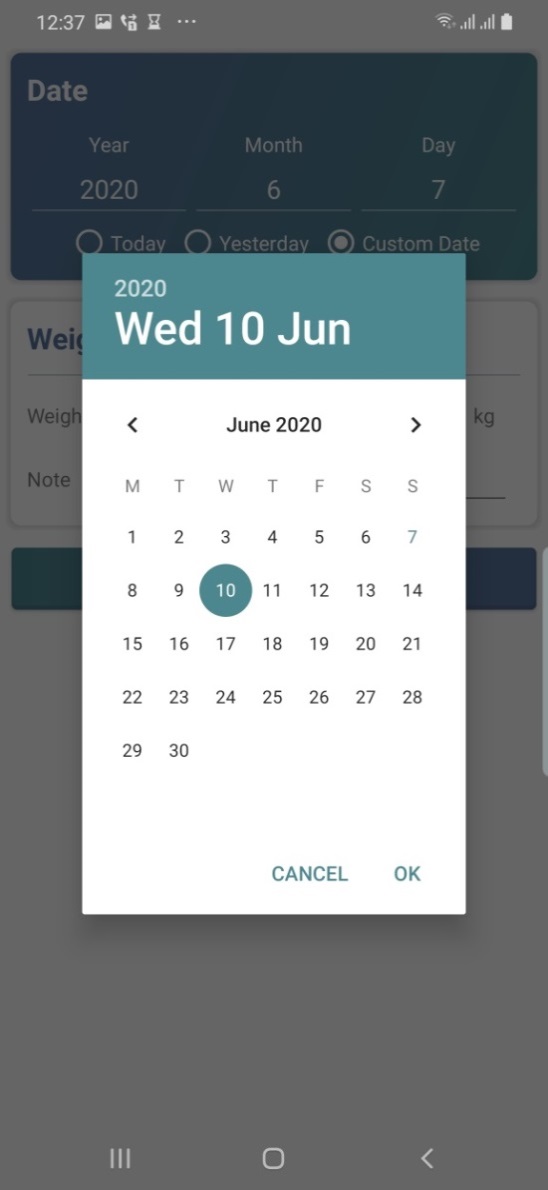
Померањем главног корисничког екрана удесно, кориснику се приказује списак његове до тада забележене телесне масе, сортиране од најскоријег датума ка ранијим. Детаљнији приказ описаног корисничког екрана дат је на слици 4.5.1.



Слика 4.5.1. Евиденција телесне масе корисника

Кликом на иконицу десно од датума отвара се дијалог који поред телесне масе и датума када је иста забележена, приказује и додатну белешку ако је корисник имао потребе да остави неку напомену у тренутку мерења.

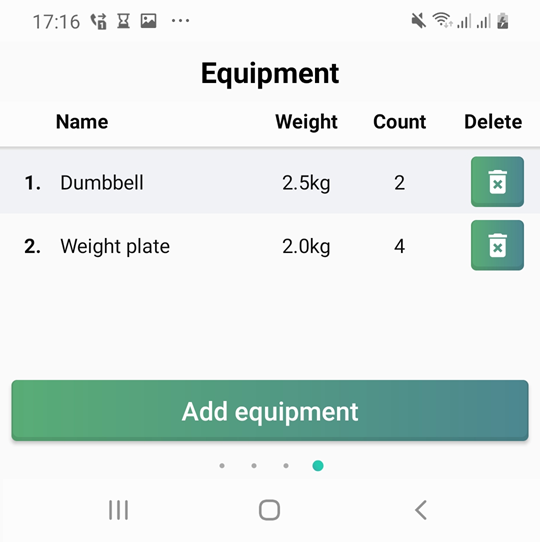
На дну листе запамћених телесних мерења налази се дугме за додавање новог. Тако се отвара кориснички дијалог који захтева од корисника да унесе телесну масу, неку напомену уколико је она потребна и датум мерења. Избор датума омогућен је отварањем календарског дијалога који је приказан на слици 4.5.1.



Слика 4.5.1. Дијалог за одабир датума

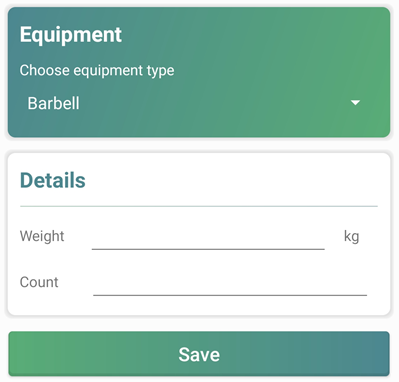
* 1. Приказ корисникове опреме

Померањем корисничког екрана још једном удесно, долази се до приказа корисникове опреме, у виду листе, која садржи име типа опреме, колико је маса опреме, као и у ком броју је корисник поседује. У сваком реду који представља корисникову опрему налази се и иконица за брисање исте. Део описаног корисничког екрана приказан је на слици 4.6.1.



Слика 4.6.1. Преглед корисникове опреме

Притиском на дугме *Add equipment* отвара се кориснички екран за додавање корисникове опреме. Из падајућег менија бира се о ком типу опреме је реч. Такође се уноси и информација о маси опреме, као и у ком броју је корисник поседује. Кликом на дугме *Save* опрема се додаје корисниковој опреми.



Слика 4.6.2. Додавање корисникове опреме

1. Реализација система

Реализација система почела је изградњом Андроид апликације. Од тренутка када је за даљи рад било неопходно памћење података и ван ње, започето је моделовање базе података и рад на серверу, као и комуникација између свих делова система. Надаље су приказани делови кода који су били најизазовнији приликом израде апликације.

* 1. Андроид апликација

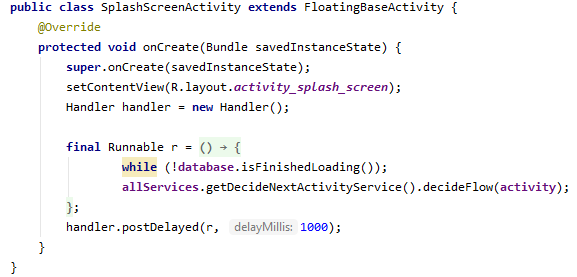
Свака Андроид апликација састоји се из тзв. активности – *Activity*. Оне представљају улазну тачку интеракције са корисником. Да би нека класа била активност она мора да проширује класу *Activity* или класу *AppCompactActivity* која је свакако проширење класе *Activity* у класној хијерархији. Самим тим има додатне функционалности као што је *action bar*, и чешће се користи у пракси.

Елементи корисничког интерфејса се дефинишу и дизајнирају у *xml* фајловима. Ако активност подржава кориснички интерфејс, онда јој се додељује одговарајући *xml* фајл приликом њеног креирања. Дакле, код који је задужен за изглед активности дефинише се кроз *xml* фајл*,* док је код саме активности задужен за њено понашање.

У наставку је приказано која активност се прва покреће, и који све процеси том приликом започињу у позадини.

* + 1. Апликација при покретању

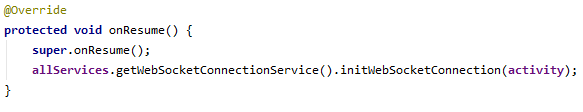
У оквиру *AndroidManifest.xml* фајла главног директоријума дефинисана је и активност која се прва креира приликом покретања самог система, а то је *SplashScreenActivity.* Код ове класе дат је на слици 5.1.1.1.



Слика 5.1.1.1. Класа *SplashScreenActivity*

За разумевање даљег рада система, битно је нагласити да методе кроз које пролази активност на самом почетку када је креирана су *onCreate()*, *onStart()* и *onResume(),* респективно. То је само почетни део животног циклуса активности који је овде издвојен као неопходан за схватање иницијализације система.

Једна од наткласа класе *SplashScreenActivity* има редефинисану методу *onResume()* чији је код дат на слици 5.1.1.2.Како ниједна од њених наткласа нема редефинисану методу *onStart()*, датој класи остаје подразумевано извршавање ове методе.

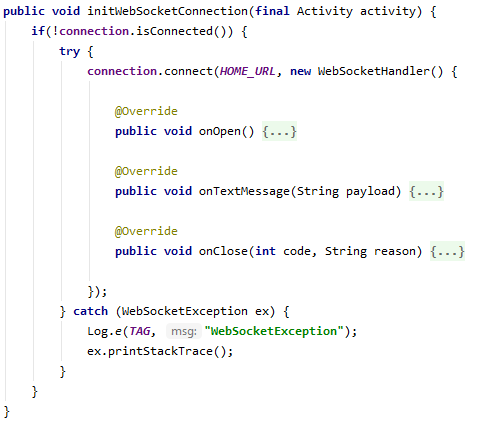


Слика 5.1.1.2. Редефинисана метода *onResume()*

Дакле, при покретању активности *SplashScreenActivity* она улази редом у методе *onCreate(), onStart()* и *onResume().* Како је код функције *run()* у методи *onCreate()* одложен за 1000*ms*, наставља се извршавања метода *onStart()* и *onResume().* Ове методе извршавају своје подразумеване операције. Метода *onResume()* поред подразумеваних наслеђених акција има позив функције *allServices.getWebSocketConnectionService().initWebSocketConnection(activity)* која иницира повезивање са серверском апликацијом.

* + 1. Повезивање Андроид и серверске апликације

*WebSocket API* омогућава размену података између Андроид апликације и сервера. Да би се успоставила ова комуникација неопходна је интернет конекција и рад сервера у виду ослушкивања порта унапред задате *IP* адресе. За реализовање ових захтева задужена је класа *WebSocketConnectionService*, која се састоји из поља *connection* типа *WebSocketConnection*, функције за дохватање наведеног поља и функције *initWebSocketConnection* чиjа је структура представљена на слици 5.1.2.1.

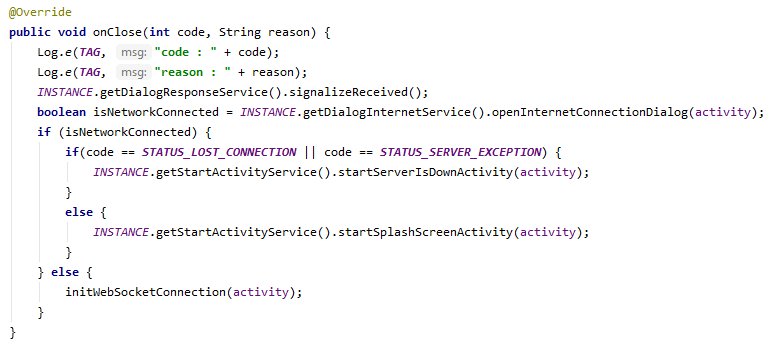


Слика 5.1.2.1. Структура функције *initWebSocketConnection*

Функција *initWebSocketConnection* проверава да ли је успостављена комуникација између Андроид апликације и сервера. Уколико то није случај, позива се метода *connect*.

Први аргумент методе је *HOME\_URL* који је за време израде пројекат имао вредност *“ws://192.168.1.25:8080/Gymdroid”*, јер је управо сервер ослушкивао порт 8080 *IP* адресе 192.168.1.25, јер је то *IP* адреса машине на којој је систем развијан. Међутим, усмеравањем сервера да ослушкује порт неке друге адресе, или покретањем сервера са друге машине морала би и вредност *HOME\_URL* да се ажурира.

Други аргумент методе је типа *WebSocketHandler* чије методе могу да се редефинишу. На слици 5.1.2.2. је приказана редефиниција методе *onClose().*



Слика 5.1.2.2. Редефинисана метода *onClose()* класе *WebSocketHandler*

У ову методу се улази када изостане конекција између Андроида и сервера. Функција *INSTANCE.getDialogInternetService().openInternetConnectionDialog(activiity)* проверава да ли постоји интернет конекција на персоналном уређају.

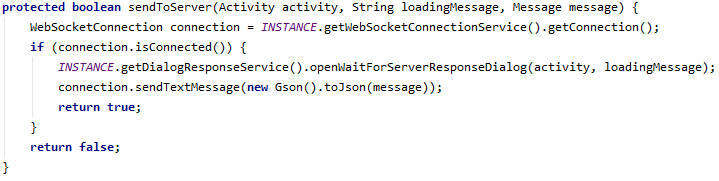
Прва могућност је да нема интернет конекције. У том случају, уколико није већ отворен, отвара се дијалог који приказује да корисник треба да се конектује на интернет, и поново се позива функција *initWebSocketConnection(activity).* Тако се непрестано проверава да ли има интернет конекције, кориснику се исписује порука да обезбеди исту, и даљи рад се наставља само у тренутку када она буде обезбеђена.

Уколико је интернет конекција обезбеђена, онда се провера да ли је дошло до сметњи на серверском делу апликације, тј. да ли је он уопште покренут. Уколико је то случај, систем одлази на кориснички екран где се исписује одговарајућа порука кориснику. Кликом на дугме тог корисничког екрана корисник поново покушава конекцију са серверском апликацијом, док се у позади заправо поново покреће активност *SplashScreenActivity.* Њена *onResume()* метода позива изнова *initWebSocketConnection().* Овај процес се понавља док се не успостави конекција Андроид и серверске апликације.

* + 1. Слање захтева серверу

Реализација функције која се позива приликом слања захтева одн. неке поруке серверу дата је на слици 5.1.3.1. Дохвата се поље *connection* класе *WebSocketConnectionService* и уколико је конектовано, методом *sendTextMessage* која као аргумент захтева *String* прослеђује се порука серверу. Из тог разлога порука *message* типа *Message* се претвара у своју *Json* репрезентацију. За ту потребу користи се *Gson* библиотека која омогућава претварање *Json* објекта у *Java* објекат и обрнуто. Примарна класа која се користи за ову намену је *Gson* која једноставно може да се креира са *new* *Gson()*.

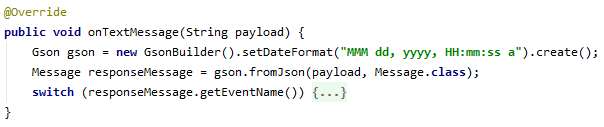
Док се чека одговор од стране серверске апликације није дозвољено радити друге операције које такође захтевају њен рад. Из тог разлога за време размене поруке између ових делова система приказује се дијалог са одговарајућом поруком.



Слика 5.1.3.1. Слање захтева серверу

* + 1. Прихватање одговора од сервера

Као што је већ речено, успостављање комуникације између Андроида и сервера врши се у функцији *initWebSocketConnection.* У тренутку успостављања конекције позива се метода  *connect* чији први аргумент представља порт *IP* адресе којег сервер ослушкује, а други аргумент је типа *WebSocketHandler.* Његова метода *onTextMessage* је директно задужена за пријем порука. Редефиниција те методе за потребе овог система дата је на слици 5.1.4.1.



Слика 5.1.4.1. Прихватање одговора од сервера

И у овој функцији се за конверзију објекта из једног облика у други користи класа *Gson.* Међутим, овде се врши конверзија из *Json* у *Java* објекат, док је код слања захтева серверу смер конверзије био обрнут. Разлог томе је што *WebSocket API* омогућава слање и пријем података који су типа *String*.

Када се изврши конверзија податка из *Json* репрезентације у *Java* објекат типа *Message* проверава се које је намене била та порука. То се врши у *switch* гранању чији имплементациони детаљи нису овде приказани због своје обимности. На основу типа поруке надаље се примењују одговарајуће операције.

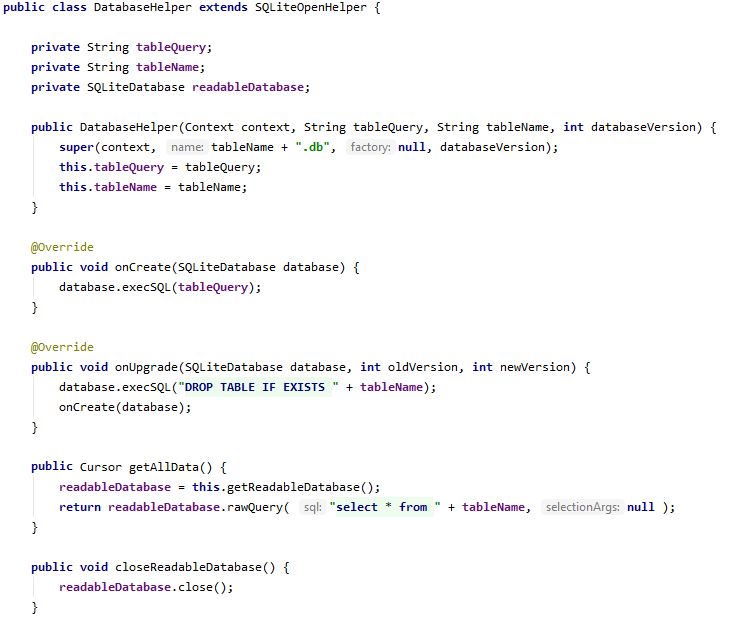
* + 1. База података

*SQLite* библиотека омогућава да се у оквиру Андроид апликације креира база података која се смешта у фолдер *data/data/NAZIV\_PAKETA/databases*.

Библиотека садржи класу *SQLiteOpenHelper* чије методе *getWritableDatabase()* и *getReadableDatabase()* враћају инстанце класе *SQLiteDatabase* које су у режиму читања и писања. У методи *onCreate()* се врши креирање и иницијализација табела када се први пут креира база, док се *onUpgrade()* позива када дође до промене верзије базе.

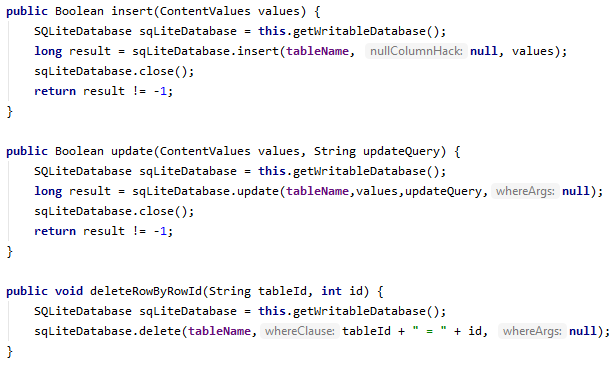
Да би се користиле методе класе *SQLiteOpenHelper* прави се класа која ће да наследи њене методе. За потребе овог система направљена је класа *DatabaseHelper* чија је реализација представљена на сликама 5.1.5.1. и 5.1.5.2.

Класа садржи поље *tableQuery* које представља *String* репрезентацију *SQL (Structured Query Language)* упита за креирање базе. Поље *tableName* представља назив табеле коју креирамо и користимо.



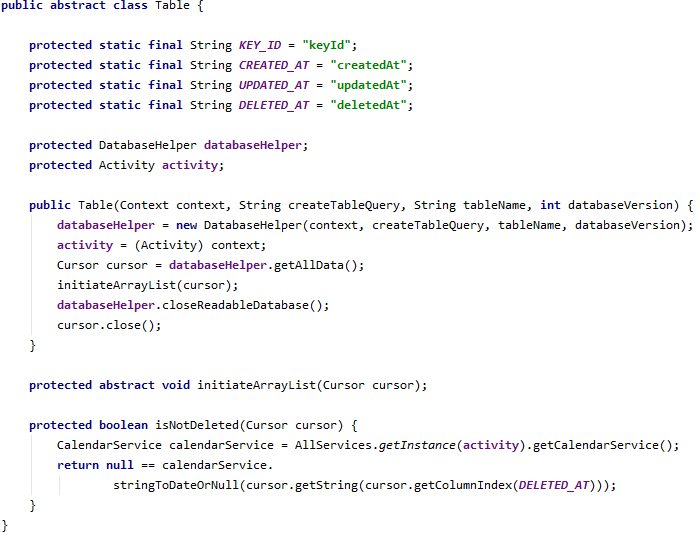
Слика 5.1.5.1. Класа *DatabaseHelper*

На слици 5.1.5.2. приказане су функције ове класе чијим позивом се врши операције убацивања, ажурирања и брисања из базе. Свака од њих се састoји из дохватања базе типа *SQLiteDatabase* која је у режиму писања, обављања тражене операције над базом и затварања исте. Уколико је резултат операције -1, операција се није успешно извршила, те је резултат функције *false*. У супротном се враћа *true* као резултат.

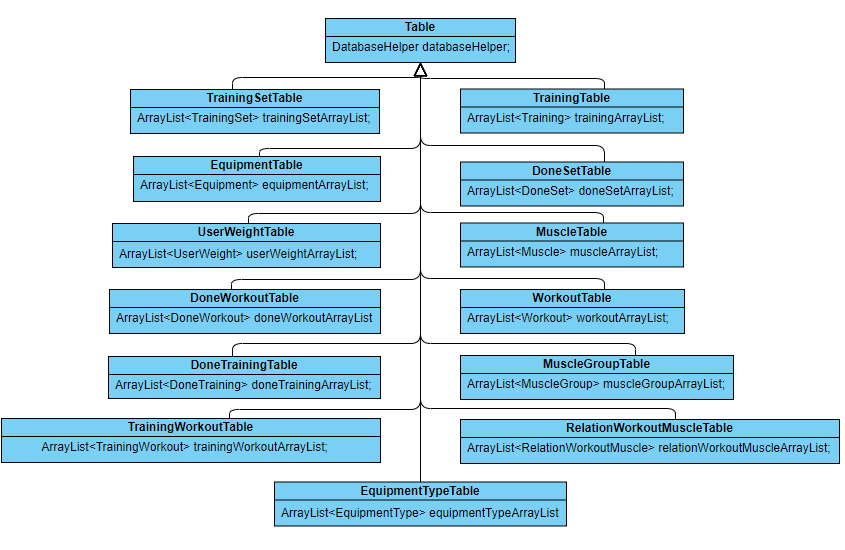


Слика 5.1.5.2. Методе за убацивање, ажурирање и брисање из базе података

На слици 5.1.5.3. приказана је класа *Table*. Она је направљена тако да представља апстракцију табеле у бази података. Све табеле које постоје у бази на серверу постојаће и као класе у Андроид апликацији и оне проширују управо ову класу. Дијаграм тих класа дат је на слици 5.1.5.4. Поред тога што проширују класу *Table,* свака од ових класа садржи и низ објеката одговарајућег типа. Класе представљају репрезентацију табела, а ови низови редове табела.

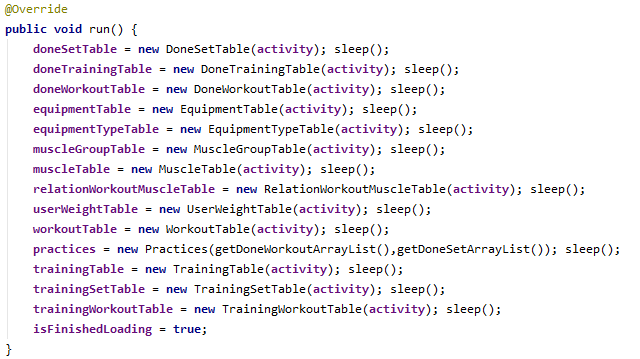


Слика 5.1.5.3. Класа *Table*



Слика 5.1.5.4. Дијаграм класа које представљају апстракцију табела базе података

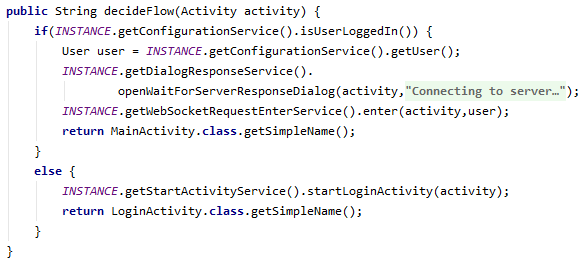
У конструктору класе *Table* и ових класа се преко поља *databaseHelper* дохватају подаци из базе која је на путањи *data/data/com.gymdroid/databases*, и добијене вредности се уписују у поменуте низове. Класа *Database* има ове класе као своја сопствена поља и омогућава једноставан рад са подацима. Приликом сваког покретања система у њеној функцији *run()* врши се дохватање података и иницијализација низова.



Слика 5.1.2.5. Метода *run()* класе *Database*

* + 1. Преусмеравање на одговарајући кориснички екран приликом покретања

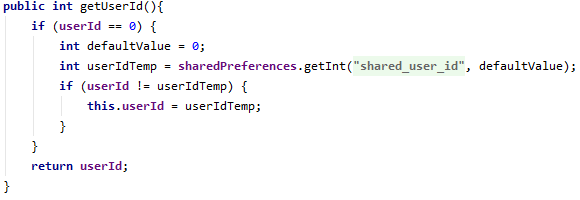
Приликом покретања система, у зависности да ли се корисник приликом последњег коришћена излогово или не, одлучује се на који кориснички екран ће бити преусмерен цео систем. Наиме, уколико је корисник остао улогован, нема потребе да се улогује поново, и стога се упућује на главни кориснички екран. Међутим, уколико се приликом последњег коришћена излогово или покреће систем по први пут, треба га преусмерити на страницу за логовање. Метода која врши ово усмеравање дата је на слици 5.1.6.1.



Слика 5.1.6.1. Функција која преусмерава корисника на одговарајући екран

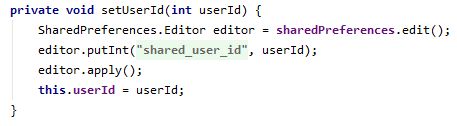
За памћења података између два покретања апликације користи се објекат класе која имплементира *SharedPreference* интерфејс. За потребе провере да ли је корисник улогован довољно је запамтити његов *id* уколико јесте, и вредност 0 уколико није.

На слици 5.1.6.2. дата је помоћна функција која преко поменутог објектапровера да ли је корисник улогован.



Слика 5.1.6.2. Функција за проверу да ли је корисник улогован

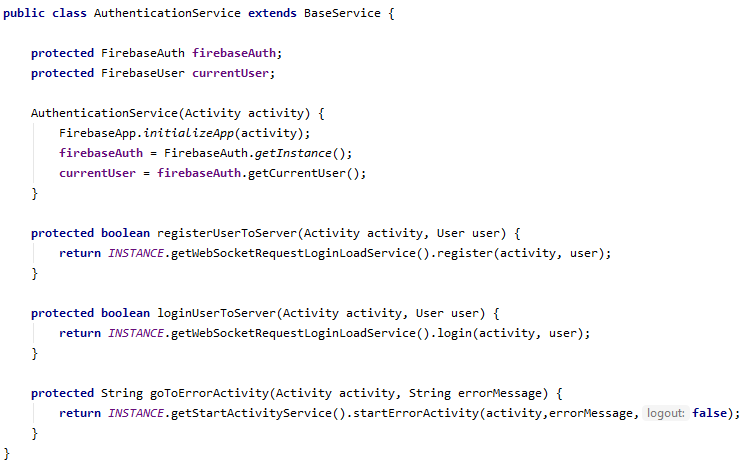
Помоћна функција која памти вредност корисниковог *id-*а приликом логовања дата је на слици 5.1.6.3. Међутим, ова функција се позива и када се корисник одјави, с тим што се тада памти вредност 0, као назнака да нико није улогован.



Слика 5.1.6.3. Фунцкија за чување информације о кориснику приликом логовања

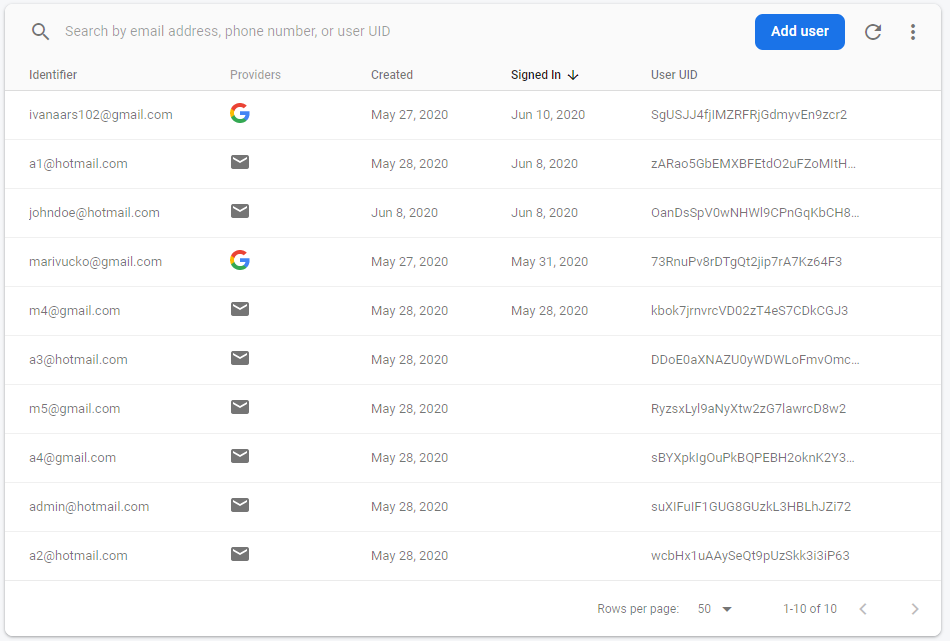
* + 1. Регистрација и логовање

Класа *AuthenticationService* приказана на слици 5.1.7.1. представља заједничке особине код пријава на систем путе имејл адресе и путем *Googlе* налога. Проширивањем ове класе добијају се класе *EmailAuthService* и *GoogleAuthService* које омогућавају да се корисник пријави на систем или преко своје имејл адресе и лозинке, или преко *Googlе* налога. Класе су реализоване коришћењем *Firebase* библиотеке.



Слика 5.1.7.1. Класа *AuthenticationService*

*Firebase* платформа омогућава и памћење корисника који се региструју у систем. Приказ табеле корисника коју су регистровани захваљујући *Firebase* аутентикацији дат је на слици 5.1.7.2. Захваљујући томе није потребно памтити податке корисника локално у оквиру апликације.

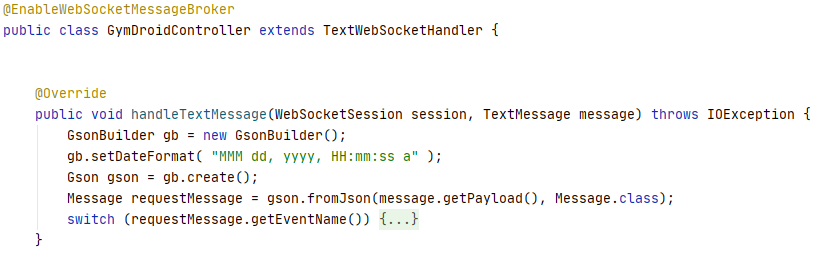


Слика 5.1.7.2. Табела корисника у оквиру *Firebase* аутентикације

* 1. Серверска апликација
     1. Прихватање захтева на серверу

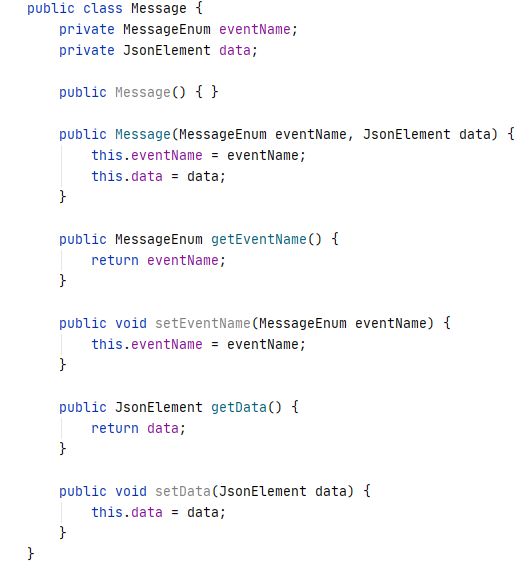
Да би сервер примио поруку од стране Андроид апликације проширена је класа *TextWebSocketHandler* и редефинисана њена метода *handleTextMessage*. Имплементациони детаљи дати су на слици 5.2.1.1.

И овде се као и код пријема у Андроид апликацији *Јson* објекат конвертује у *Јаva* објекат типа *Message*. На основу типа примљене поруке предузимају се даље акције.

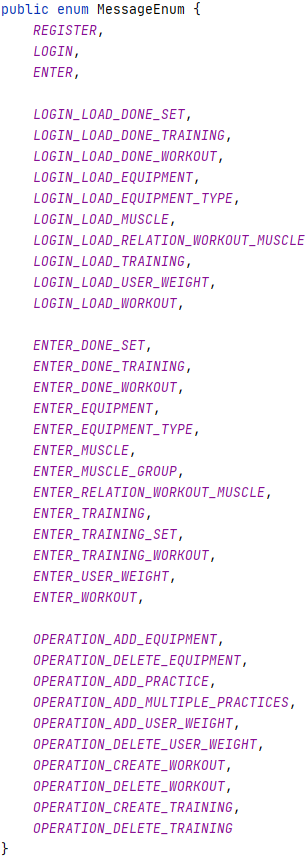


Слика 5.2.1.1. Класа задужена за пријем поруке од Андроид апликације

Поруке које се размењују између ових делова система су типа *Message.* Реализација те класе дата је на слици 5.2.1.2. Прво поље класе је типа *MessageEnum* који је приказан на слици 5.2.1.3. Друго поље је типа *JsonElement*, што омогућава да инстанца сваке класе конверзијом у *Јson* објекат може бити послата.



Слика 5.2.1.2. Класа *Меssage*

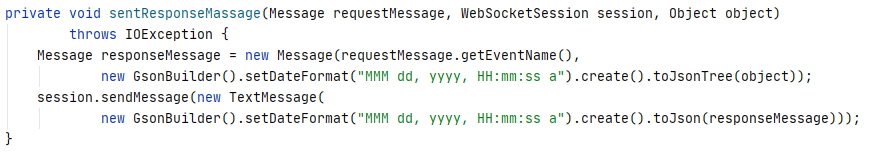


Слика 5.2.1.3. *MessageEnum*

*MessageEnum* казује коју операцију сервер треба да изврши, а самим тим указује и на то ког је ће типа бити друго примљено поље после конверзије из *Јson* објекта. *MessageEnum* који у свом имену садржи реч *LOAD* захтева дохватање података након логовања, док уколико садржи реч *ENTER* захтева дохватање само оних података који нису у бази Андроид апликације. Остале могућности за вредности од *MessageEnum* представљају сет различитих операција попут додавања и брисања података.

* + 1. Слање серверског одговора

Функција која омогућава да сервер пошаље одговор дата је на слици 5.2.2.1. Вредност првог поља порукепри пријему и слању је иста, да би се знало која операција је у питању, али је друго поље поруке различито.



Слика 5.2.2.1. Слање одговора од стране сервера

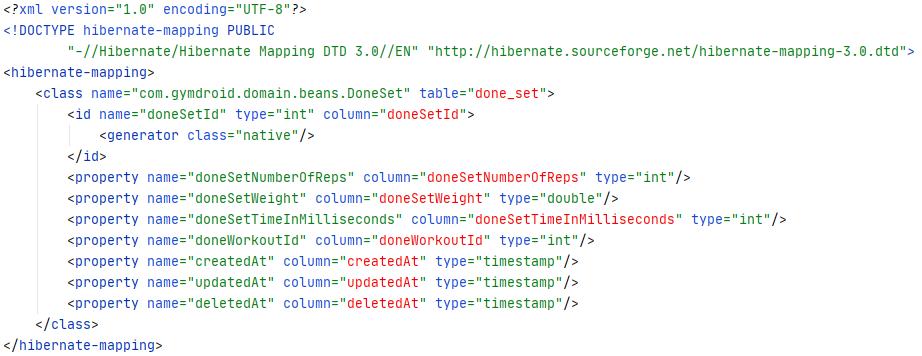
* + 1. Повезивање сервера и базе

Комуникација сервера и базе омогућена је захваљујући алату *Hibernate*. *Hibernate* конфигурациони фајл дат је на слици 5.2.3.1. Уколико се сервер покреће са различитих машина треба подесити да конфигурациони фајл и *MySQL Workbench* рачунара са којег се покреће имају исте вредности за поља *username* и за поља *password.*



Слика 5.2.3.1. Фајл *hibernate.cfg.xml*

Свака табела која се налази у бази треба да има и свој мапирајући фајл који ће је повезати са њеном класном репрезентацијом која се може надаље користити у оквиру серверске апликације. Пример мапирајућег фајла једне од табела дат је на слици 5.2.3.2. У њему се поред имена табеле, свих назива и типова колона налази и име класе у коју се мапира као и њена поља која одговарају колонама те табеле.



Слика 5.2.3.2. Фајл *done\_set.hbm.xml*

1. Закључак

Ера развоја технике и интернета, као и доступност информација популаризовали су тренд здравог живота и наметнули бављење физичком активношћу као неизоставни део свакодневице. Технолошки напредак није само мултиплицирао циљну групу фитнес индустрије, него је и умногоме омогућио прогрес сваког појединца. Наметнути су нови стандарди не само у спољашњем изгледу, него и у техничким начинима који олакшавају пут до тог циља. Сходно томе, примарни задатак ове тезе је реализација система која би корисницима омогућио једноставан увид у њихов тренинг и напредак.

Попут разних апликације сличне намене, и у реализацији овог система су подржане функционалности попут приказа сета иницијално изабраних вежби, са објашњењем како се свака вежба понаособ изводи и којим мишићима је намењена. Штавише, корисник има могућност уноса сопствене вежбе, чиме је циљна група корисника проширена. Наиме, скуп вежби у процесу рехабилитације може доста да варира од пацијента до пацијента, те их је тешко предвидети. С опцијом ручног уноса, апликацију могу користити не само активни спортисти, него и људи који су у процесу опоравка који захтева телесне вежбе.

Надаље постоји могућност организовање вежби по серијама, и њихово груписање у тренинг. Постоји и режим тренирања, током којег корисник изводи вежбе по редоследу по којем их је додавао током креирања тренигна. За сваку вежбу се приказује редни број текуће серије као и очекивани број понављања и оптерећења. По завршетку сваке серије следи режим паузе који је такође изабран од стране корисника, након чега се наставља са наредном серијом и вежбом.

Апликација укључује и преглед одрађених тренинга, као и детаље сваке изведене вежбе. Корисник има и опцију да води евиденцију своје телесне масе као и опреме коју поседује, или најчешће користи у теретани. То му олакшава каснији прорачун са којим оптерећењем изводи вежбе.

Од техничке подршке је избрано да се систем састоји из апликације намењене извршавању на Андроид оперативном систему, јер би приликом самог тренигна најпрактичнија опција за кориснике била управо употреба мобилних уређаја. Већина људи који се баве било каквим видом физичке активности свакако користи мобилне телефоне као саставни део своје рутине, бар као вид смернице колико им траје тренинг.

Поред Андроид апликације са којом корисници директно интерагују, систем се састоји и из серверске апликације. Предност постојања сервера је то што не постоји могућност губитка корисникових података. Наиме, његови подаци нису везани за уређај на којем је иницијално инстанлирана апликација. Самим тим уколико корисник покреће апликацију и са других уређаја, услед губитка претходног, или из било којих других разлога, довољно је да унесе своје креденцијале при првом покретању и да несметано настави свој рад.

Додатни допринос који обезбеђује рад сервера је то што су потребе за ажурирањем Андроид апликације сведене на минимум. Дакле, уколико се одлучи да се додају нове вежбе или типови опреме у систем, или да се изврши другачија прерасподела група мишића, довољно је жељене податке ажурирати на серверу. С обзиром на то да је свако покретање Андроид апликације реализовано тако да дохвата одређене података са сервера, сваки корисник ће услед промена података на серверу видети ажуриране податке и у апликацији. Тако се постиже да потреба за уносом или изменом података више не захтева и преузимање нове верзије апликације.

Приликом изградње сваког система па и овог, настоји се да систем буде модуларан и да се лако надограђује када се укаже потреба. Једна од ставки која би свакако требало да буде подржана у каснијим верзијама апликације је и визуелни приказ покрета који се изводи током саме вежбе. Овај захтев би могао да буде реализован или приказивањем слике вежбе, или кратке анимације извођења исте у трајању од једног или више понављања.

Наредни захтев који би могао да буде реализован у једном оваквом систему је да апликација подсећа корисника да одради тренинг. Корисник би на самом почетку бирао којим данима жели да тренира и у које време. Наравно, корисник би имао и слободу одабира којим данима ће које тренинге да ради. Тиме би могао према својим жељама и могућностима да сам креира свој програм тренинга.

Постоји још много функционалности за којима ће се појавити потреба у наредним годинама, посебно с порастом броја људи који тренирају. Циљ ове тезе је управо изградња једне фитнес апликације која ће помоћи свим корисницима који су изразили потербу за системом оваквих карактеристика и који су жељни тренирања.

Литература

1. A. Gerber, C. Craig, *Learn Android Studio,* Apress, 2015.
2. A. P. Ruiz, *Mastering Android Application Development,* Packt Publishing, 2015.
3. M. Deinum, *Spring Boot 2 Recipes*, Apress, 2018.
4. M. McLaughlin, *MySQL Workbench: Data Modeling & Development*, McGraw-Hill Education, 2013.
5. Documentation for app developers [Online]. Available: <https://developer.android.com/docs> (11.06.2020.)
6. Firebase Documentation [Online]. Available: <https://firebase.google.com/docs> (10.06.2020.)
7. Android time picker with seconds field [Online]. Available: <https://github.com/IvanKovac/TimePickerWithSeconds> (17.05.2020.)
8. Android JustifiedTextView Library [Online]. Available: <https://mvnrepository.com/artifact/me.biubiubiu.justifytext/library/1.1> (11.06.2020.)
9. Gson 2.2.4 API [Online]. Available: <https://javadoc.io/doc/com.google.code.gson/gson/2.2.4/index.html> (17.05.2020.)
10. IntelliJ IDEA [Online]. Available at: <https://www.jetbrains.com/idea/documentation/> (22.05.2020.)
11. Hibernate ORM Documentation – 5.0 [Online]. Available: <https://hibernate.org/orm/documentation/5.0/> (19.06.2020.)
12. MySQL Connector/J 5.1 Developer Guide [Online]. Available: <https://dev.mysql.com/doc/connector-j/5.1/en/> (19.05.2020.)
13. MySQL Workbench Documentation [Online]. Available: <https://dev.mysql.com/doc/workbench/en/> (28.05.2020.)
14. MySQL 8.0 Reference Manual [Online]. Available: <https://dev.mysql.com/doc/refman/8.0/en/> (28.05.2020.)

Списак скраћеница

|  |  |
| --- | --- |
| API | *Application Programming Interface* |
| IP | *Internet Protocol* |
| SQL | *Structured Query Language* |

Списак слика

[Слика 3.1.1.1. Део базе који представља релације између вежби, мишића и мишићних група 6](#_Toc43125682)

[Слика 3.1.2.1. Део базе који представља однос између вежби, одрађених вежби и серија 7](#_Toc43125683)

[Слика 3.2.1. Део базе који представња однос између тренинга и вежби 8](#_Toc43125684)

[Слика 3.2.1.1. Дијаграм секвенце за захтев стартовања тајмера 9](#_Toc43125685)

[Слика 3.2.1.2. Део базе података који омогућава памћење одрађеног тренинга 9](#_Toc43125686)

[Слика 3.3.1. Део базе који омогућава вођење евиденције о корисниковој телесној маси 10](#_Toc43125687)

[Слика 3.4.1. Део базе који омогућава вођење евиденције о опреми коју корисник поседује 10](#_Toc43125688)

[Слика 3.5.1. Дијаграм секвенце за захтев додавања/брисања података 11](#_Toc43125689)

[Слика 3.5.2. Дијаграм секвенце за захтев читања података 11](#_Toc43125690)

[Слика 4.1.1. Почетни кориснички екран 14](#_Toc43125691)

[Слика 4.2.1. Главни кориснички екран 15](#_Toc43125692)

[Слика 4.2.2. Навигација кроз корисничке екране 16](#_Toc43125693)

[Слика 4.3.1.1. Скролбар 16](#_Toc43125694)

[Слика 4.3.1.2. Приказ група мишића и броја вежби за сваку групу понаособ 16](#_Toc43125695)

[Слика 4.3.1.3. Унос података за жељену вежбу одабране групе мишића 17](#_Toc43125696)

[Слика 4.3.2.1. Кориснички дијалог за креирање тренинга 18](#_Toc43125697)

[Слика 4.3.2.2. Дијалог за одабир временске паузе 18](#_Toc43125698)

[Слика 4.3.2.3. Падајући мени за одабир телесне вежбе 19](#_Toc43125699)

[Слика 4.3.2.4. Одабир броја серија, броја понављања и масе оптерећења 19](#_Toc43125700)

[Слика 4.3.2.5. Креиран тренинг 20](#_Toc43125701)

[Слика 4.3.3.1. Дијалог тренутне серије и вежбе 20](#_Toc43125702)

[Слика 4.3.3.2. Опис вежбе 20](#_Toc43125703)

[Слика 4.3.3.3. Пауза између серија 21](#_Toc43125704)

[Слика 4.3.4.1. Преглед одрађеног тренинга 21](#_Toc43125705)

[Слика 4.4.1. Списак доступних вежби 22](#_Toc43125706)

[Слика 4.4.2. Детаљни приказ свих извођења вежбе 23](#_Toc43125707)

[Слика 4.4.3. Додавање нове вежбе 23](#_Toc43125708)

[Слика 4.5.1. Евиденција телесне масе корисника 24](#_Toc43125709)

[Слика 4.5.1. Дијалог за одабир датума 24](#_Toc43125710)

[Слика 4.6.1. Преглед корисникове опреме 25](#_Toc43125711)

[Слика 4.6.2. Додавање корисникове опреме 25](#_Toc43125712)

[Слика 5.1.1.1. Класа *SplashScreenActivity* 26](#_Toc43125713)

[Слика 5.1.1.2. Редефинисана метода *onResume()* 27](#_Toc43125714)

[Слика 5.1.2.1. Структура функције *initWebSocketConnection* 27](#_Toc43125715)

[Слика 5.1.2.2. Редефинисана метода *onClose()* класе *WebSocketHandler* 28](#_Toc43125716)

[Слика 5.1.3.1. Слање захтева серверу 29](#_Toc43125717)

[Слика 5.1.4.1. Прихватање одговора од сервера 29](#_Toc43125718)

[Слика 5.1.5.1. Класа *DatabaseHelper* 30](#_Toc43125719)

[Слика 5.1.5.2. Методе за убацивање, ажурирање и брисање из базе података 31](#_Toc43125720)

[Слика 5.1.5.3. Класа *Table* 31](#_Toc43125721)

[Слика 5.1.5.4. Дијаграм класа које представљају апстракцију табела базе података 32](#_Toc43125722)

[Слика 5.1.2.5. Метода *run()* класе *Database* 32](#_Toc43125723)

[Слика 5.1.6.1. Функција која преусмерава корисника на одговарајући екран 33](#_Toc43125724)

[Слика 5.1.6.2. Функција за проверу да ли је корисник улогован 33](#_Toc43125725)

[Слика 5.1.6.3. Фунцкија за чување информације о кориснику приликом логовања 33](#_Toc43125726)

[Слика 5.1.7.1. Класа *AuthenticationService* 34](#_Toc43125727)

[Слика 5.1.7.2. Табела корисника у оквиру *Firebase* аутентикације 34](#_Toc43125728)

[Слика 5.2.1.1. Класа задужена за пријем поруке од Андроид апликације 35](#_Toc43125729)

[Слика 5.2.1.2. Класа *Меssage* 35](#_Toc43125730)

[Слика 5.2.1.3. *MessageEnum* 36](#_Toc43125731)

[Слика 5.2.2.1. Слање одговора од стране сервера 36](#_Toc43125732)

[Слика 5.2.3.1. Фајл *hibernate.cfg.xml* 37](#_Toc43125733)

[Слика 5.2.3.2. Фајл *done\_set.hbm.xml* 37](#_Toc43125734)

Списак табела

[Табела 2.2.6.1. Упоређивање постојећих апликација 5](#_Toc44283934)

[Табела 3.6.1.1. Табела зависности које треба додати и бенефити који се тиме остварују 13](#_Toc44283935)