

**Fitness aplikacija**

**„2FIT4U“**

# **Tehnička dokumentacija projekta razvoja aplikacije „2FIT4U“**

## **1. VERZIJA**

Ovaj dokument predstavlja službenu dokumentaciju nastalu prilikom izrade aplikacije 2FIT4U.

Link na GitHub repozitorij: <https://bit.ly/2D1mD1a>

Link na ZenHub: <https://bit.ly/2r5yMPL>

# Sadržaj

1. Uvod .....	1
1.1. Općenito .....	1
1.2. Svrha dokumentacije.....	1
1.3. Djelokrug.....	1
1.4. Definicije, akronimi i skraćenice .....	1
1.5. Reference .....	2
1.6. Pretpostavke .....	2
2. Generalni opis.....	3
2.1. Pregled .....	3
2.1.1. „Pametna vaga“ – NFC realizacija.....	3
2.1.2. WebServis – zapis izmjerene težine u bazu podataka.....	4
2.2. Funkcije proizvoda .....	5
2.2.1. Korisničke priče (zahtjevi) .....	5
2.2.2. Popis funkcionalnosti aplikacije .....	6
3. Specifični zahtjevi .....	8
3.1. Zahtjevi performansi.....	8
3.2. Logički zahtjev nad bazom podataka .....	8
3.3. Ograničenja dizajna .....	8
3.4. Obilježja aplikacije .....	9
4. Opis dizajna sustava (UML dijagrami).....	10
4.1. UseCase dijagram aplikacije 2FIT4U .....	10
4.2. Dijagram klasa aplikacije 2FIT4U .....	11
4.3. ERA model.....	11

# 1. Uvod

## 1.1. Općenito

Naredna potpoglavlja predstavljaju detaljnu specifikaciju zahtjeva prema IEEE Std 830-1998. Software Requirements Specifications. U uvodu će se opisati će se svrha, djelokrug, definicije termina reference te opći pregled.

## 1.2. Svrha dokumentacije

U ovom dokumentu bit će opisane tehničke i softverske specifikacije. Također bit će navedeni korisnički zahtjevi odnosno korisničke priče u obliku sažete liste (popisa) te funkcionalnosti same aplikacije koje će biti implementirane na temelju korisničkih priča. U dokumentu će biti navedena i ograničenja sustava te popisani vanjski servisi i objašnjena komunikacija između istih i aplikacije. Ovaj dokument namijenjen je samom naručitelju aplikacije u svrhu dobivanja povratnih informacija te razvojnom timu sustava, programerima, kako bi se što bolje definirali zahtjevi te kako bi se dobila čista slika same aplikacije.

## 1.3. Djelokrug

Aplikacija je namijenjena svim korisnicima Android mobilnih uređaja i služi za praćenje težine i tjelesnih mjera kroz vrijeme. Sam sustav za isporuku sastojat će se od sljedećih komponenata:

- Mobilne aplikacije namijenjene korisnicima Android mobilnih uređaja koji si prateći težinu i tjelesne mjere kroz vrijeme mogu postavljati određene ciljeve
- Online baze podataka (Microsoft Sql Server)

## 1.4. Definicije, akronimi i skraćenice

Korisnik – osoba koja je preuzela aplikaciju i služi se njome. Može unositi težinu ručno ili preko NFC-a. Može unositi tjelesne mjere i dnevni kalorijski unos. Može pratiti sve prethodno navedeno kroz vrijeme te određivat si razne ciljeve poput željene kilaže.

NFC – skraćenica za Near Field Communication. Putem NFC-a, odnosno prislanjanjem uređaja sa NFC čipom na vagu koja koristi isto NFC čip, korisniku će se prikazana težina s vage automatski unjeti u aplikaciju.

UML – UML je engleska skraćenica za Unified Modeling Language, odnosno za standardizirani jezik za modeliranje

Use case – Statički UML dijagrami, pripadaju skupini ponašajnih dijagrama jer modeliraju moguće ponašanje korisnika sustava. Prikazuju interakciju sustava i korisnika i moguće načine korištenja.

AS – Android studio, razvojno sučelje (IDE)

MS SQL – Microsoft SQL Server, korištena baza podataka

## **1.5. Reference**

Le Vie, D.: Writing Software Requirements Specification (2010), Dostupno sa:  
<http://www.letu.edu/people/jaytevis/Software-Engineering/Examples/writing-software-requirements-specifications.doc>

## **1.6. Pretpostavke**

Pretpostavka jest da budući korisnici sustava posjeduju Android mobilni uređaj te da imaju pristup internetu.

## **2. Generalni opis**

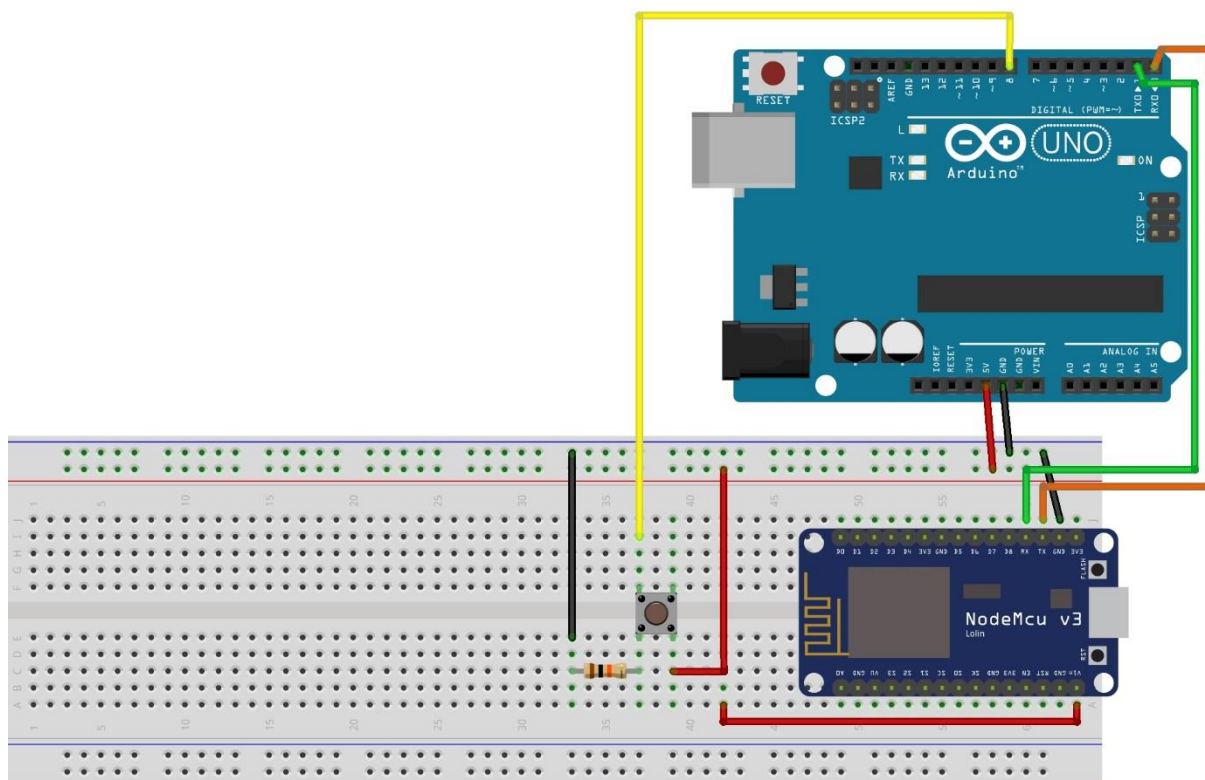
U sljedećim poglavljima bit će dan pregled čimbenika koji utječu na sam proizvod te njegovi zahtjevi. Zatim slijedi opis korisnika te sam način funkcioniranja aplikacije.

### **2.1. Pregled**

Proizvod je zamišljen kao mobilna aplikacija za Android mobilne uređaje za čije korištenje je potreban internet. Svrha aplikacije bi bila praćenje svojih tjelesnih mjera i težine kroz vrijeme te izvršavanje ciljeva koje si korisnik zadaje poput željene kilaže ili obujma struka. Korisniku se na temelju unesenog kalorijskog dnevnog unosa izračuna vrijeme potrebno da ostvari svoj cilj. Težinu korisnik može unijeti na dva načina. Ručno ili prislanjanjem uređaja s NFC čipom na vagu preko koje se težina direktno unosi u bazu podataka odnosno vidljiva je u samoj aplikaciji (nije potrebno unijeti ručno). U nastavku slijedi opis na koji način funkcionira prethodno naveden princip unosa.

#### **2.1.1. „Pametna vaga“ – NFC realizacija**

Kao vrstu unosa težine korisnik može koristiti i pametnu vagu. Pametna vaga je uređaj koji se sastoji od Arduino Uno microcontrolera na koji su povezani potrebni senzori za (mjerenje težine korisnika - Load Cell Half-bridge Strain + HX711 AD Amplifier Module, NFC čitač - RFID-RC522, 4x7 segmentni display), modula za spajanje na Internet (NodeMCU V3.2 Arduino ESP8266 ESP-12 E) te WebServisa koji služi za zapisivanje izmjerene težine u bazu podataka.



Slika 1: Pametna vaga

### 2.1.2.WebServis – zapis izmjerene težine u bazu podataka

Svrha izrađenog WebServisa je izmjerene podatke putem pametne vage zapisati u bazu podataka.

#### Poziv WebServisa

"http://www.pamtim.com/Weight/Insert/?Data=" + X1 + "-" + X2 + "-" + X3

X1 – vrijednost NFC oznake korisnika

X2 – vrijednost izmjerene težine korisnika

X3 – sigurnosna vrijednost

#### Primjer poziva WebServisa

http://www.pamtim.com/Weight/Insert/?Data=12341234-86-1341996

X1 = 12341234

X2 = 86

## 2.2. Funkcije proizvoda

U nastavku slijede korisničke priče kojima su definirani korisnički zahtjevi koje je potrebno realizirati te funkcionalnosti aplikacije.

### 2.2.1. Korisničke priče (zahtjevi)

1. Naša želja je dobiti sustav za praćenje i mjerenje napretka tjelesnih mjera kroz vrijeme
2. Želimo Android mobilnu aplikaciju koja će služiti za praćenje i mjerenje tjelesnih mjera kroz vrijeme
3. Želimo da aplikacija sadrži registraciju, prijavu te odjavu korisnika putem E-maila ili korisničkog imena i lozinke
4. Aplikacija treba podržavati funkcionalnost oporavka zaboravljene lozinka
5. Želimo da korisnik u aplikaciju unosi tjelesne mjere vrata, struka i bokova te ga aplikacija obavještava o ispravnom načinu mjerenja istih
6. Korisnik mora imati uvid u povijest svih svojih ranije spomenutih mjera koje je unosio
7. Aplikacija mora a temelju tjelesnih mjera korisnika izračunavati postotak tjelesne masti te ga prikazati korisniku
8. Korisnik mora imati mogućnost unosa svoje trenutne tjelesne težine
9. Želimo da korisnik ima dvije mogućnosti unosa trenutne tjelesne težine, a to su ručni unos tjelesne težine preko aplikacije te unos preko „pametne“ vage koja nakon što se korisnik identificira automatski sprema izmjerenu težinu za korisnika u bazu podataka. Za neidentificiranog korisnika vaga na zaslonu ispisuje težinu
10. Želimo da korisnik ima uvid u povijest svoje tjelesne težine te željene, početne i trenutne težine
11. Želimo da se korisniku napredak tjelesne težine od početne prema željenoj prikaže u obliku Progress bar-a
12. Korisniku se treba omogućiti grafički prikaz napretka odnosno statistiku korisnika koju je moguće filtrirati po određenim parametrima
13. Želimo da korisnik ima mogućnost dodavanja NFC oznaka kojima zatim može dodjeljivati naziv te također korisnik može i ukloniti već dodane NFC oznake
14. Želimo da aplikacija redovito obavještava korisnika u obliku podsjetnika i motivacijskih poruka koje korisnik može uključiti i isključiti

15. Korisnik prilikom prve prijave mora unijeti dnevni unos kalorija, količinu aktivnosti u tjednu, odabrati mjerne jedinice u kojima želi prikazivati podatke
16. Želimo da korisnik nad svim podacima ima omogućene funkcionalnosti promjene podataka

## 2.2.2. Popis funkcionalnosti aplikacije

F 1. Registracija i prijava korisnika – svaki korisnik da bi se mogao koristiti temeljnim funkcionalnostima aplikacije mora se registrirati te zatim nakon toga prijaviti

- F 1.1. Registracija – potrebno je izraditi korisnički račun, odnosno registrirati se kako bi korisnik mogao koristiti aplikaciju u potpunosti. Prilikom registracije nužno je popuniti polja: ime, prezime, e-mail, lozinka i ponovljena lozinka te pritisnuti na gumb „Registriraj se“. E-mail i lozinka su izričito važni jer su oni potrebni za prijavu u aplikaciju
- F 1.2. Prijava – funkcionalnost koja korisniku omogućuje „ulazak“, odnosno prijavu u svoj korisnički račun. Za prijavu je potrebno unesti e-mail i lozinku koje je korisnik unio prilikom registracije
  - F.1.2.1. Zaboravljena lozinka – omogućuje korisniku resetiranje lozinke ukoliko je zaboravio trenutačnu. Nova lozinka se šalje na e-mail korisnika te je isti nužno upisati kako bi se ostvarila ova funkcionalnost
- F 1.3. Odjava – korisnik ima mogućnost odjaviti se iz aplikacije (izaći iz svog korisničkog računa)

F 2. Unos i praćenje tjelesnih mjera – jedna od dvije temeljnih funkcionalnosti aplikacije koja omogućava korisniku unos određenih tjelesnih mjera te putem algoritma izračunava *Body fat*

- F. 2.1. Unos tjelesnih mjera – korisnik ima mogućnost unijeti tjelesne mjere vrata, struka i bokova. Korisnik prvo mora obaviti mjerenje sam te se preporučuje uzimanje mjera uvijek na isti način te ujutro na prazan stomak kako bi mjerenje bilo što pouzdanije:
  - ✓ Vrat – stojeći u najširem dijelu (preko Adamove jabučice)
  - ✓ Struk – stojeći u najužem dijelu ili na sredini između vrha kuka i dna prsnog koša
  - ✓ Bokovi – na najširem dijelu, gdje su mišići gluteusa (stražnjice) najveći

Na temelju uneseni tjelesnih mjera aplikacija izračunava postotak masti (eng. *Body fat*) te korisnik unosi podatke pritiskom na gumb *Save*.



- F 2.2. Prikaz tjelesnih mjera – korisnik vidi svoja zadnja mjerenja zajedno sa izračunatim postotkom tjelesne masti, početno mjerenje i sva prethodna mjerenja uz koje je „stoj“ datum kada ih je isti unio.

### F 3. Unos i prikaz težine – druga temeljna funkcionalnost aplikacije

- F3.1. Unos težine „ručno“ – korisnik može unijeti trenutnu kilažu ručno, da upiše koliko kilograma ima nakon što se izvagao
- F 3.2. Unos težine preko „pametne“ vage – korisniku je omogućen također unost težine preko vage koja ima NFC čitač kojim se korisnik identificira na način da prisloni NFC oznaku na vagu te se podaci automatski preko interneta spremaju u bazu podataka za tog korisnika
- F 3.3. Prikaz težine – korisnik vidi svoju trenutnu, početnu, željenu i sve prethodne težine te datum uz njih kad ih je unio da li ručno ili preko „pametne“ vage. Također korisnik vidi svoj napredak prikazan kao *progress bar* koji predstavlja crtu koliko je korisniku još potrebno da ostvari svoju cilj, odnosno željenu težinu.

F 4. Grafički prikaz napredka – ova funkcionalnost omogućuje korisniku da grafički vidi svoj napredak kroz vrijeme. Korisnik ima mogućnost filtriranja po određenim parametrima kako bi prilagodio prikaz.

### F 5. Dodavanje i uklanjanje NFC oznaka

- F 5.1. Dodavanje NFC oznaka – korisnik ima mogućnost dodavanja NFC oznaka kojim će ostvarivati funkcionalnost unosa težine preko „pametne“ vage (F 3.2.)
  - F 5.1.1. Davanje imena NFC oznaka – prilikom dodavanja NFC oznake korisnik mu daje ime
  - F 5.2. Uklanjanje NFC oznaka - korisnik može ukloniti već dodane NFC oznake

F 6. Obavijesti aplikacije – aplikacija ima predefinirane obavijesti koje služe kao podsjetnik i motivacija korisniku

- F 6.1. Uključivanje/isključivanje obavijesti – korisnik prema vlastitoj želji može isključiti ili uključiti obavijesti

### F 7. Ostale funkcionalnosti

- F 7.1. Unos i promjena kalorija – korisnik ima mogućnost promjene dnevnog kalorijskog unosa koji je potreban za izračun ciljeva korisnika

- F 7.2. Unos i promjena aktivnosti – korisnik unosi i može mijenjati koliko se tjedno puta bavi nekom tjelesnom aktivnosti
- F 7.3. Promjena mjernih jedinica u kojima su podaci prikazani – korisnik ima mogućnost biranja mjernih jedinica u kojima će biti prikazani podaci

F 7.4. Promjena inicijalno unesenih podataka

## **3. Specifični zahtjevi**

### **3.1. Zahtjevi performansi**

Ukoliko aplikacija odgovara prethodno navedenim osnovnim zahtjevima (popis funkcionalnosti, karakteristika korisnika, ograničenja, pretpostavke i ovisnosti) problemi sa pouzdanošću i performansama prema našim procjenama ne bih se trebali pojaviti. Pretpostavljamo, da eventualne probleme aplikacija može imati u radu sa bazom podataka, gdje on radi svoje malo vjerojatno nedostupnosti ili pak pada ne bi dozvolila ispravan rad aplikacije

### **3.2. Logički zahtjev nad bazom podataka**

Logički zahtjev nad bazom podataka kao i sama struktura pojedinih entiteta jasno je vidljiva na ERA modela priloženom u sljedećem poglavlju.

### **3.3. Ograničenja dizajna**

Prilikom izrade aplikacije 2FIT4U vodili smo industrijskim standardima koje ćemo koristiti i u budućem održavanju aplikacije, s imperativom nad korisničkom iskustvu sve u svrhu zadovoljavanja potrebe korisnika aplikacije.

Sam mockup aplikacije u obliku wireframe-a dan je na slici ispod. Ukoliko čitatelji želi pristupiti interaktivnom prototipu aplikacije, te može učinit također pritiskom na sljedeći link:

<https://marvelapp.com/jcida7e/screen/62901649>



Slika 2: Wireframe aplikacije

### 3.4. Obilježja aplikacije

**Pouzdanost** – kako bi aplikacija bila u skladu s korisničkim zahtjevima, te funkcionirala na zahtijevani način, potrebno je izvršiti testiranja, odnosno proći kroz postupak utvrđivanja grešaka. Prvenstveno će se testiranje obaviti u okolini koja zadovoljava tehničke zahtjeve vezane uz hardver i softver.

**Dostupnost** – Aplikacija je dostupna većinu vremena (99%). Slučajevi u kojima aplikacija neće funkcionirati objašnjeni su prethodno te se tiču internetske veze. Kao nedostupnost aplikacije ne računa se vrijeme koje se troši na održavanje sustava te na prelazak na novu verziju aplikacije. Navedeni dio vremena utvrđuje se sa korisnikom i ne bi trebao predstavljati utjecaj na dostupnost aplikacije.

**Održavanje** – U trenutnoj fazi razvoja proizvoda očekuju se veće promjene u programskom proizvodu te sukladno time i veća količina napora uložena u razvoj i održavanje. Konačan proizvod kakav je dogovoren neće zahtijevati veliko održavanje već ono minimalno u obliku update-ova.

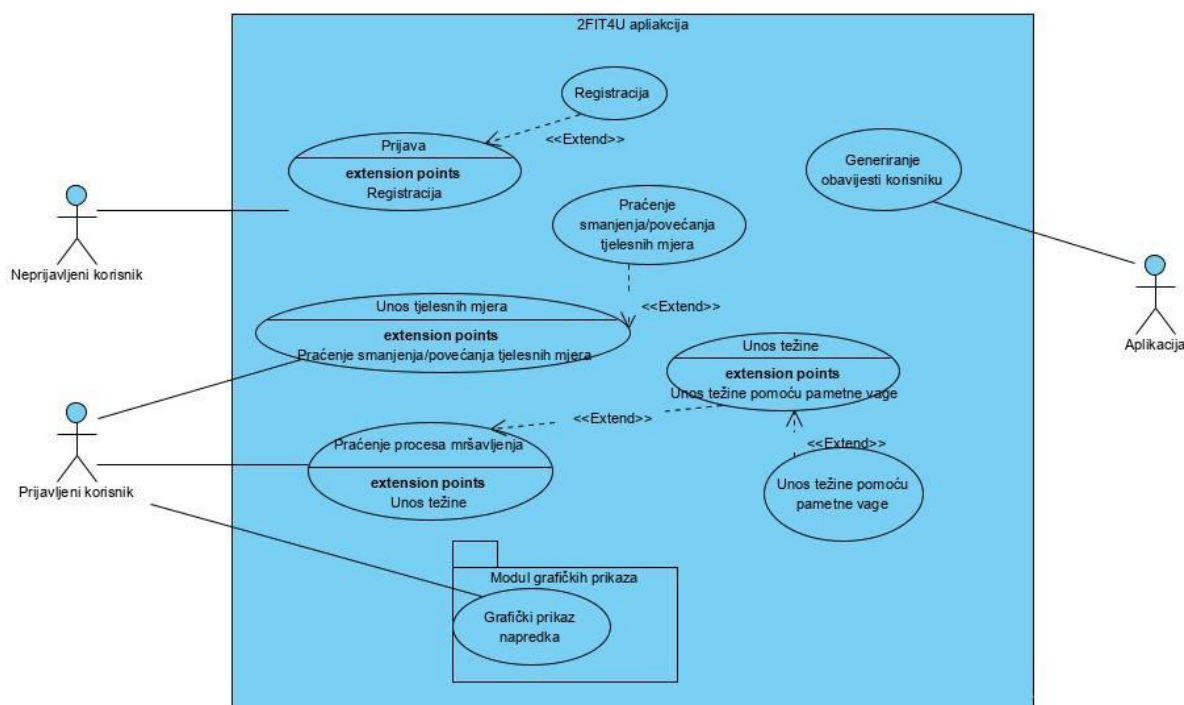
**Pokretnost** – Razvijeni proizvod, odnosno aplikacija biti će dostupna na Android mobilnim uređajima s operacijskim sustavom Lollipop 5.0 ili novijim, dok ostali sustavi trenutno nisu uključeni u obzir. Kako bi se koristila, aplikaciju je potrebno instalirati na mobilni uređaj te ju

je moguće koristiti neovisno o lokaciji, dok drugi dio sustava, sam uređaj za praćenje mora biti postavljen fiksno u vrtu kojim se upravlja te mu mora biti dostupna internetska mreža.

## 4. Opis dizajna sustava (UML dijagrami)

### 4.1. UseCase dijagram aplikacije 2FIT4U

Aplikacija 4FIT4U razlikuje 2 vrste vanjskih aktera. Prijavljenoj i neprijavljenoj korisnici. Logično, neprijavljeni korisnik ima uvelike ograničen pristup samoj aplikaciji. On se može prijaviti u samu aplikaciju pod pretpostavkom da ima kreiran korisnički račun, a s druge strane, ukoliko korisnički račun ne postoji, može se registrirati. S druge pak strane prijavljeni korisnik ima otvorena vrata za pristup svim funkcionalnostima aplikacije. Može unositi svoje tjelesne mjere u raznim vremenskim periodima kako bi aplikacija mogla započeti sa svojom metodom praćenja napretka korisnika. S druge strane može unositi težinu ručnim unosom nakon vaganja ili koristeći pametnu vagu realiziranu NFC-om kako bi se olakšalo samo korištenje aplikacije. Nadalje dostupan je pregled napretka u obliku grafičkih prikaza pogodan za oko. Na kraju, aplikacija kao unutarnji akter šalje obavijesti korisnici po postignutom rezultatu kao i podsjetnik za potrebe ponovnog vaganja.

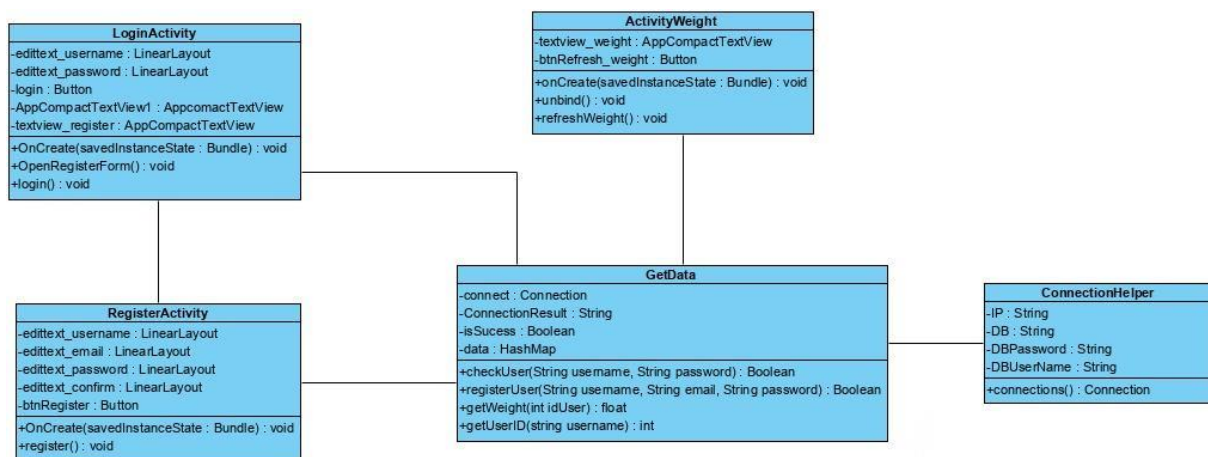


Slika 3: Use-case dijagram

## 4.2. Dijagram klasa aplikacije 2FIT4U

Na sljedećoj slici nalazi se dijagram klasa aplikacije 2FIT4U. Kako je sama aplikacija na samim počecima implementacija, na slici je dan prikaz dijagram klasa potreban za realizaciju 2 temeljne funkcionalnosti kao i klase potrebne za implementaciju baze podataka.

- Registracija i prijava korisnika
- Implementacija baze podataka



Slika 4: Dijagram klasa

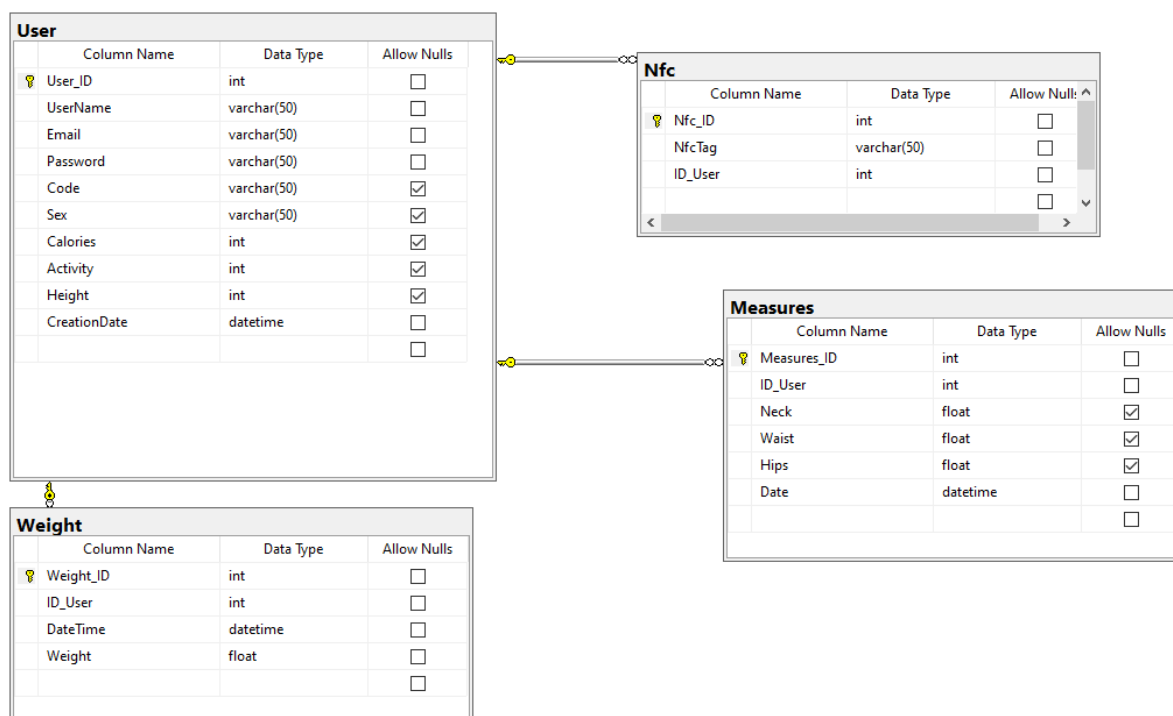
## 4.3. ERA model

Na Arvix web hostu se nalazi baza podataka. Baza podataka koje se koristi je MS SQL.

Sam ERA model se sastoji od 4 tablice. Tablica User u koju se pohranjuju podaci o

korisniku. Tablica measures u koju se pohranjuju tjelesne mjere za određenog korisnika.

Tablica Weight u koju se pohranjuje težina za određenog korisnika i tablica NFC u koju se pohranjuju podaci vezani uz NFC uređaje.



Slika 5: ERA model