

# Járásfelismerés gyorsulásérzékelő segítségével

Készítette:  
Fülöp Timea  
Mille János  
Németh Krisztián – Miklós  
Vezető tanár:  
Dr. Antal Margit

## Tartalomjegyzék

Követelmény specifikáció .....	3
Áttekintés .....	3
Gyakorlati megvalósítás .....	4
Feature Extractor Library .....	4
Model Builder Library .....	6
Beléptető és regisztráló felületek .....	7
A felhasználó feladata .....	9
Adatgyűjtő felületek .....	9
Felhasználói adatok megjelenítése .....	11
Adminisztrátori mód .....	11
Felhasználói mód .....	11
Activity Stack Management .....	12

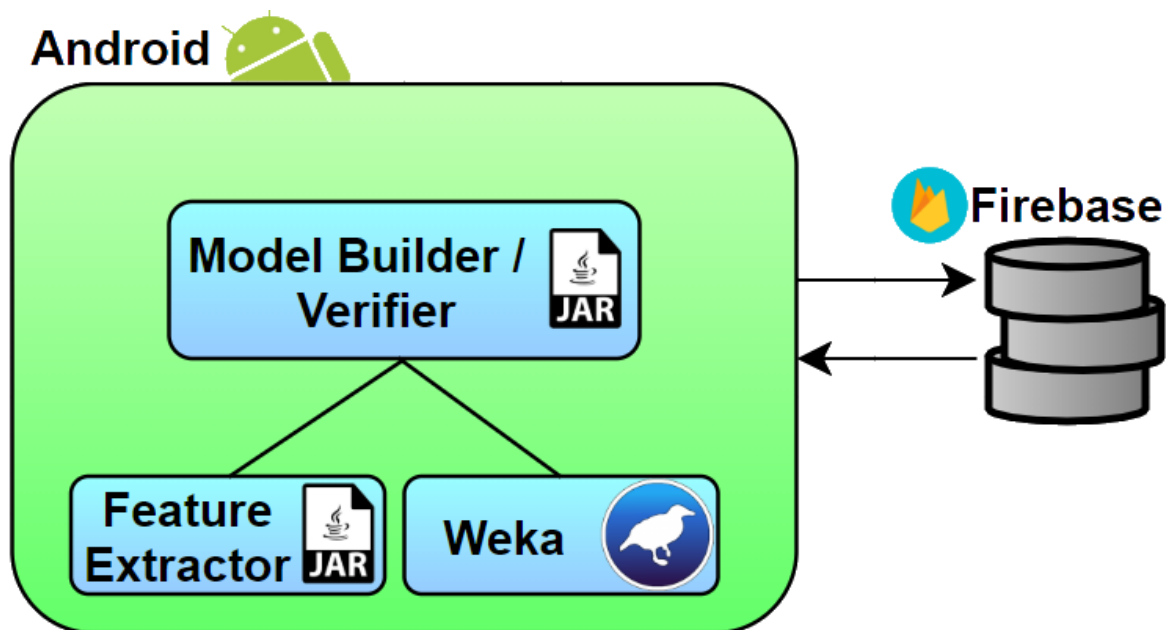
## Követelmény specifikáció

### Áttekintés

Az Android alkalmazás, amellyel a mintavételezést végeztük, három fő komponenesre osztható. Az első komponens a FeatureExtractor névre hallgat. Feladata a nyers adatok feldolgozása. Ez alatt azt értjük, hogy a gyorsulásérzékelő által érkezett jelből 59 jellemzőt von ki, ezekre a következő modulban lesz szükség. A kinyert adatokat képes többféle bájlból meghatározni, illetve beállítható az is, hogy milyen formátumba mentse el a kinyert jellemzőket.

A második komponens, a GaitModelBuilder. Ez a modul foglalkozik azzal, hogy a Feature Extractor által szolgáltatott adatokból felállítson egy bináris osztályozót, mentse le a felhasználó saját modelljét, és validáljon, vagyis ellenőrizze, hogy egy bejelentkezett felhasználónak van-e jogosultsága belépni egy adott helyiségbe.

Az előbbi két komponens egy, a harmadik komponens egyesíti. Ez maga az alkalmazás, amely képes a nyers adatokat mintavételezni, ezeket pedig feldolgozza a fent említett két modul felhasználásával. Ezen kívül biztosít egy kényelmes felületet a felhasználó számára.

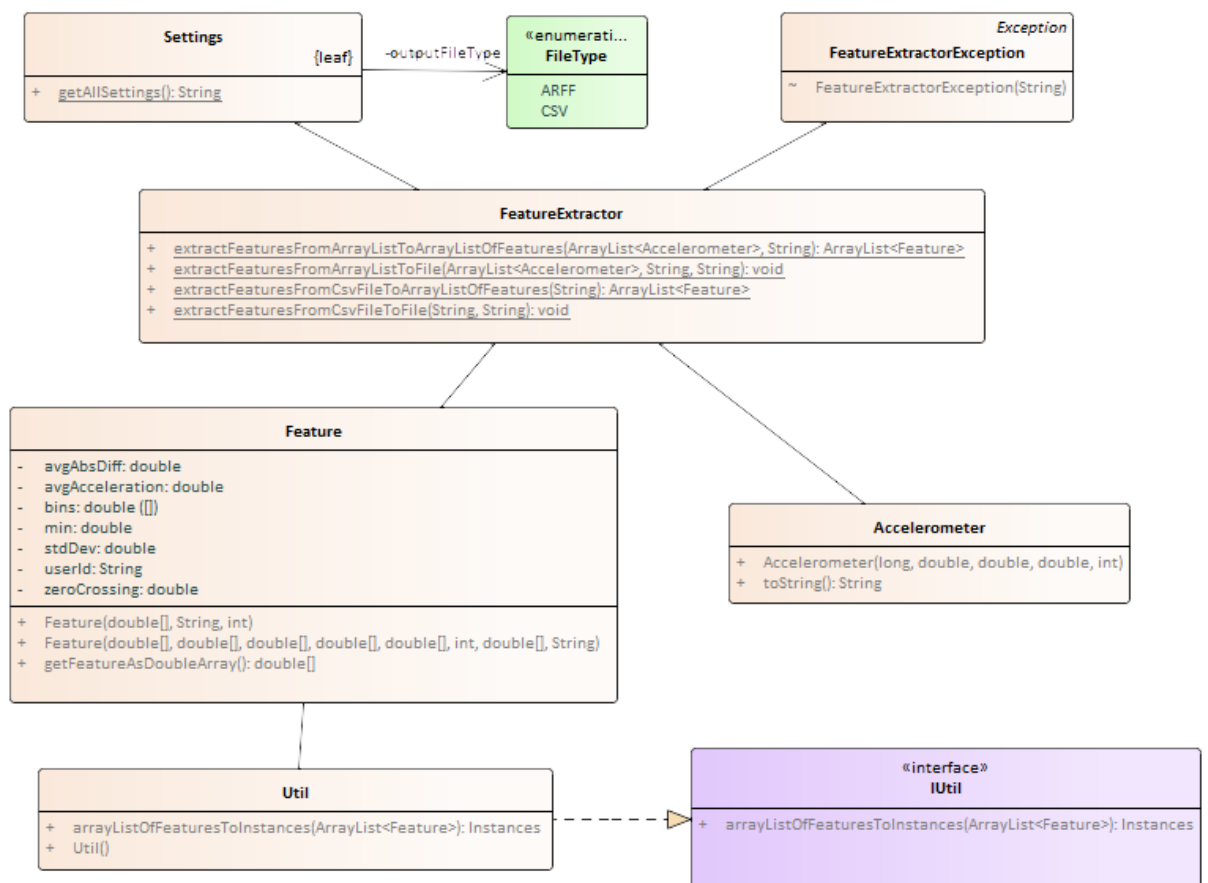


1. ábra Az alkalmazás komponensei közti kapcsolatok

## Gyakorlati megvalósítás

### Feature Extractor Library

Ahogy az előzőekben taglalva volt már, a Feature Extractor feladata a gyorsulásérzékelőtől vett nyers adatok feldolgozása és jellemzőkké alakítása. A modul felépítése az 2. ábrán található osztálydiagram szemlélteti.



2. ábra A FeatureExtractor Library-t felépítő osztályok diagramja

A FeatureExtractor osztály fő funkcionalitása a jellemzők kinyerése. Erre képes egy adatokat tartalmazó fájlt kapva mint paraméter, vagy memóriában lévő Accelerometer objektumokból álló listából és a kinyert jellemzőket egy fájlba helyezi be, vagy visszatéríti egy Feature objektumokat tartalmazó listában. A modul tartalmaz egy Settings osztályt, amely a FeatureExtractor működését határozza meg, például meghatározva a kimeneti fájl típusát és fejlécét, vagy meghatározva, hogy a bemeneti adatokat ablakokra osztva nyerje ki a jellemzőket, vagy minden egyes lépcsőciklusra nyerjen ki egy sorozat jellemzőt. A modul lehetővé teszi a Weka csomagban lévő osztályozók által használt Instances objektum generálását.

is egy Feature objektumokból álló listából a Util kényelmi osztály által nyújtott szolgáltatások segítségével.

A Feature Extractor által kinyert jellemzők az alábbi táblázat foglalja össze.

Jellemző	Magyar elnevezés	Jelentés	Jellemzők száma
minimum_for_axis_X minimum_for_axis_Y minimum_for_axis_Z minimum_for_mag	Minimum	A legkisebb érték az ax, ay, az és am jelekre.	4
average_accel_for_axis_X average_accel_for_axis_Y average_accel_for_axis_Z average_accel_for_mag	Átlag	Az ax, ay, az és am jelek átlaga.	4
standard_deviation_for_axis_X standard_deviation_for_axis_Y standard_deviation_for_axis_Z standard_deviation_for_mag	Szórás	Az ax, ay, az és am jelek szórása.	4
avg_abs_difference_for_axis_X avg_abs_difference_for_axis_Y avg_abs_difference_for_axis_Z avg_abs_difference_for_mag	Abszolút különbségek átlaga	Átlagos abszolút különbség az egyes iránymenti gyorsulások és az ennek megfelelő átlag értéke között	4

zero_crossing_rate_for_axis_X	Zéró átmenet	Előjelváltások száma	3
zero_crossing_rate_for_axis_Y			
zero_crossing_rate_for_axis_Z			
bin0_X... bin9_X	Hisztogram	10 szakaszos hisztogram	40
bin0_Y... bin9_Y			
bin0_Z... bin9_Z			
bin0_M... bin9_M			
Összesen			59

3. ábra A nyers adatokból kinyert jellemzők

## Model Builder Library

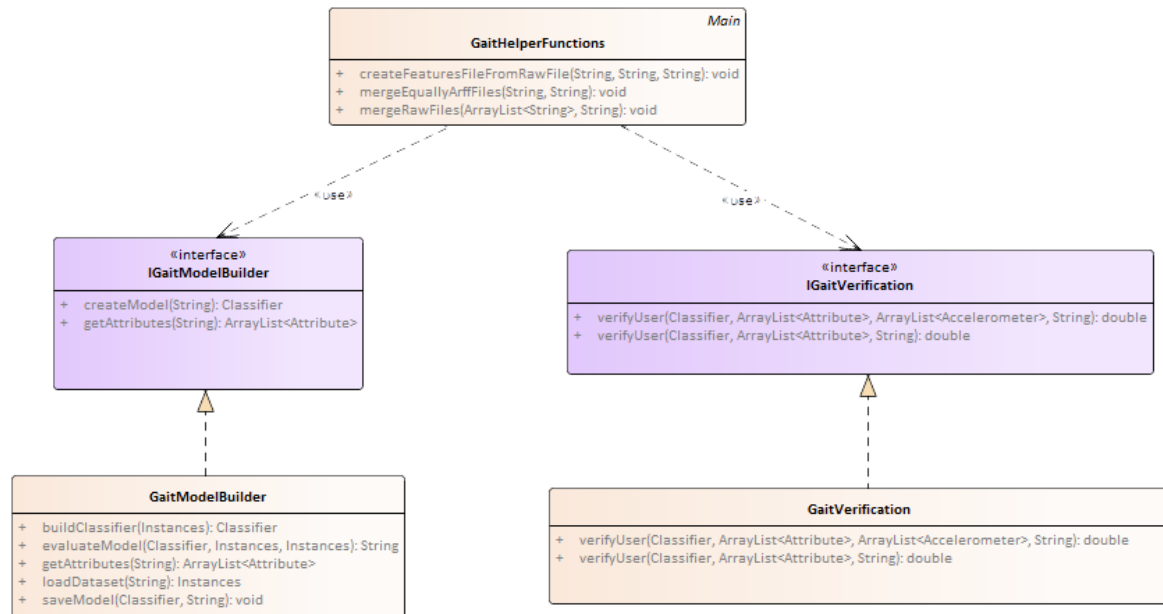
Ahhoz, hogy az embereket azonosítani lehessen a járás minta alapján, mindenkiről szükséges készítenünk egy modellt. Ezeket a modelleket később arra fogjuk használni, hogy az újonnan érkezett adatokból meg tudjuk állapítani, hogy az a bejelentkezett felhasználóhoz tartozik-e vagy inkább egy impostorhoz.

A modell elkészítéséhez szükség van nagyobb mennyiségű jellemzőkre a felhasználótól. Ezeket a Feature Extractor segítségével nyertük ki a felhasználó által generált nyers adatból. Egy bináris osztályozó betanításához ez sajnos nem elég, ezért szükség volt olyan mintákra, ami nem az aktuális felhasználótól vannak. Erre a célra létrehoztunk egy impostor felhasználót. Az ő mintáit véletlenszerűen kiválasztott emberek biztosították, így az ő mintái változatosak.

Amikor megvannak a pozitív és negatív minták, már be lehet tanítani egy osztályozót (mi esetünkben Random Forest), és el lehet menteni, mint modell, hogy a későbbiekben felhasználható legyen.

Ahhoz, hogy egy felhasználót validálni tudjunk elég a már létező modelljét előszedni (a gépet nem kell újra betanítani), és összehasonlítani az új mintáival. Az összehasonlítás végeredménye egy érték, ami azt adja meg, hogy mekkora a valószínűsége annak, hogy a

minták tényleg a felhasználóé és nem egy imposztoré.



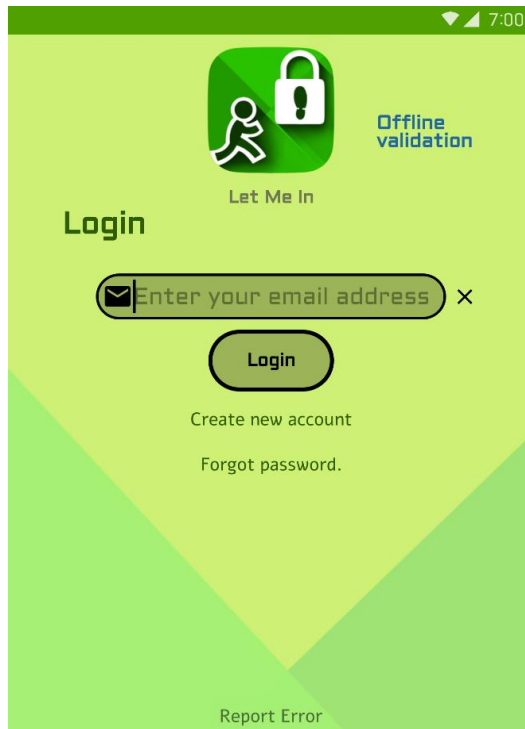
4. ábra A GaitModelBuilder Library osztálydiagrammja

## Beléptető és regisztráló felületek

Az alkalmazás indításakor egy bejelentkezést szolgáló felület látható (5. ábra). Ezt a továbbiakban Login felületnek nevezzük. Ahhoz, hogy a kliensnek lehetősége legyen használni az alkalmazást, rendelkeznie kell egy felhasználóval a rendszerben. A regisztrálást az alkalmazás megfelelő felületén kell elvégezni (6. ábra). Erre a felületre a Login felület „Create new account.” gombjára kattintva lehet elnavigálni.

A regisztrált felhasználókat a Firebase Authentication nevű szolgáltatása segítségével kezeljük. Ha a felhasználó elfelejtette a jelszavát, lehetősége van új jelszót kérni a Login felület „Forgot password.” gombjára kattintva. A rendszer küld egy e-mailt a felhasználónak egy linkkel, amelyen beállíthatja az új jelszavát.

Úgy a Login, mint a regisztrációs felület rendelkezik egy úgynevezett „Offline validation” nevű gombbal. Amennyiben a készülék nem rendelkezik internetes hozzáféréssel, akkor is lehet kérni az azonosítást, viszont ezt csak a legutoljára bejelentkezett felhasználó modelljével fogja összehasonlítani.



The login screen features a green header with a status bar at the top showing signal strength, Wi-Fi, and the time 7:00. Below the header is a green box containing a white icon of a person and a padlock, with the text "Offline validation" to its right. Below this is a "Let Me In" button. The main heading "Login" is in bold. Below it is a text input field with a mail icon, placeholder text "Enter your email address", and a clear 'x' button. Underneath is a "Login" button. Further down are links for "Create new account" and "Forgot password.". At the bottom is a "Report Error" link.

7:00

Offline validation

Let Me In

## Login

Enter your email address x

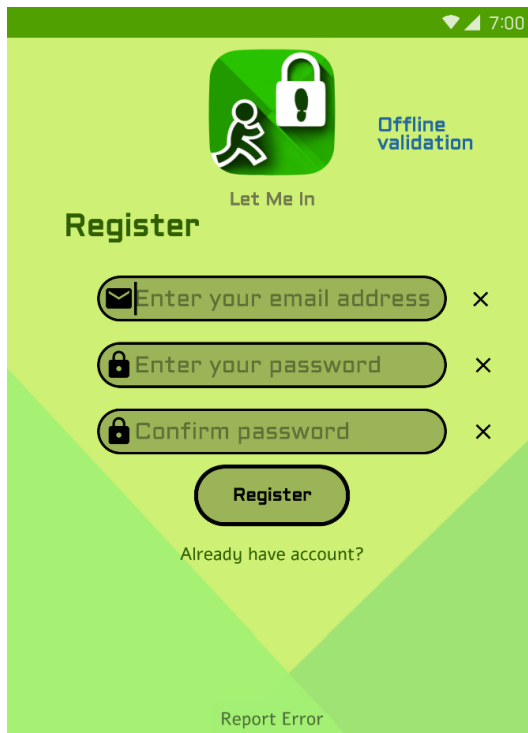
Login

Create new account

Forgot password.

Report Error

5. ábra A bejelentkezésért felelős felület



The register screen has the same green header and "Offline validation" box as the login screen. Below it is a "Let Me In" button. The main heading "Register" is in bold. Below it are three text input fields: the first for email with a mail icon, the second for password with a padlock icon, and the third for password confirmation with a padlock icon. Each field has a clear 'x' button. Below these is a "Register" button. Further down is a link for "Already have account?". At the bottom is a "Report Error" link.

7:00

Offline validation

Let Me In

## Register

Enter your email address x

Enter your password x

Confirm password x

Register

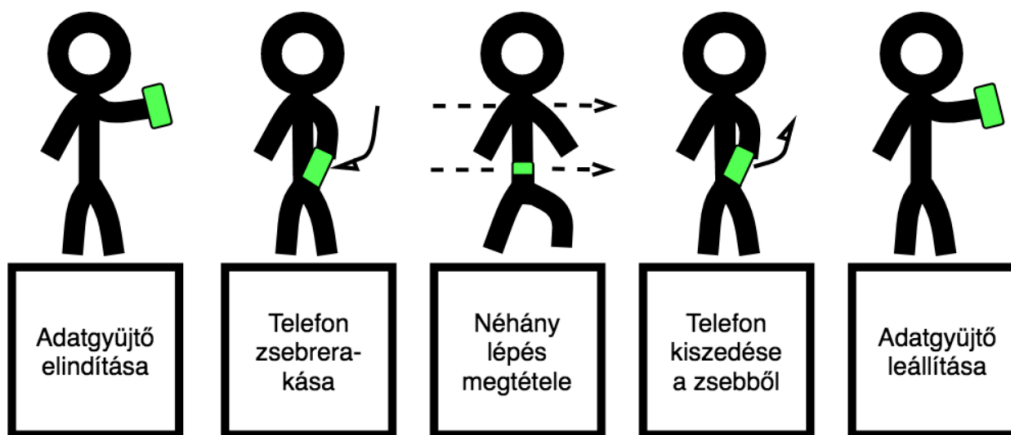
Already have account?

Report Error

6. ábra Regisztrációs felület



## A felhasználó feladata



7. ábra Mintavételezés a felhasználó szemszögéből

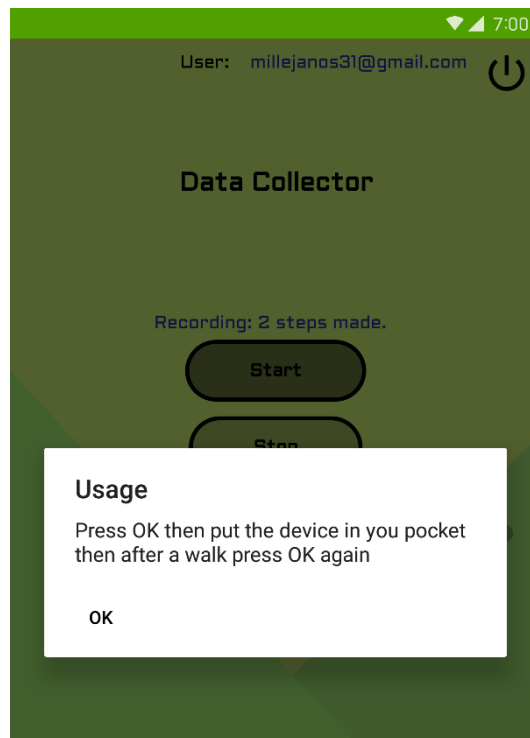
A fenti ábra magyarázza a felhasználó által elvégezendő lépéseket. Első lépésben elindítja az adatgyűjtést. Ehhez előzőleg be kellett jelentkezzen az alkalmazásban. Ez után zsebre kell rakja a készüléket (a készülék legyen a test mellett, nem jó, ha túl bő nadrágban, vagy akár térdközében helyezkedik el a készülék). Majd pár lépés után állítsa le az adatgyűjtést. Ekkor kezdődik a rendszer feladata, vagyis a feldolgozás és a kiértékelés.

## Adatgyűjtő felületek

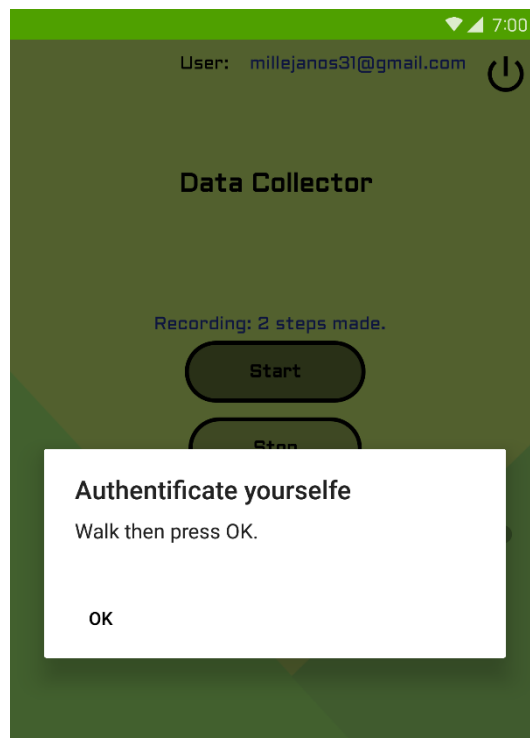
A következő három ábrán láthatóak azok a felületek, amelyek a felhasználó adatait gyűjtik. A felugró ablakok rendelkeznek a megfelelő instrukciókkal a felhasználók számára, ezek szemléltetve voltak a 7. ábrán.

Amikor a felhasználó először megnyomja az Ok gombot, a telefon elkezd a mintavételezést. A felhasználó feladata ekkor a készüléket a zsebébe rakni, illetve megtennie legalább 10 lépést, egyenes vonalban, egyenletes felületen.

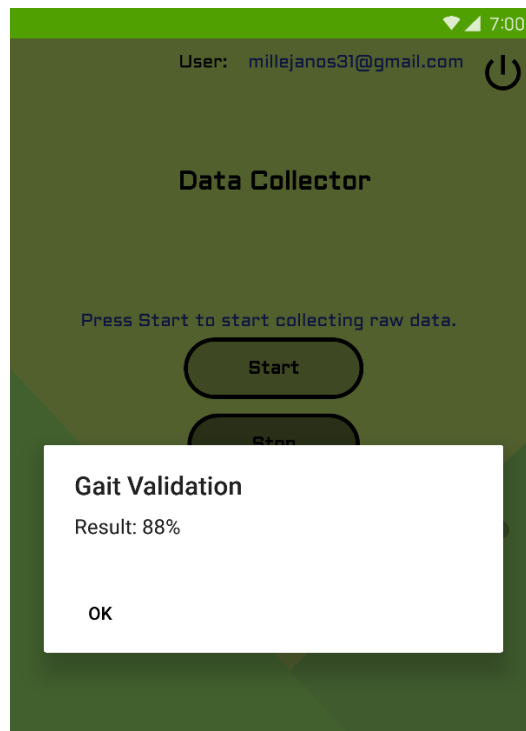
Miután ez megvan, és kiveszi a telefont a zsebéből, megint megnyomja az Ok gombot. Ennek hatására a mintavételezés megáll. Ezt követi a minták feldolgozása a Feature extractor segítségével, majd a felhasználó azonosítása, a hozzá tartozó modellel. A végeredményt egy harmadik felugró ablak adja meg. A százalék azt jelenti, hogy mekkora a valószínűsége annak, hogy tényleg az a felhasználó adta a friss járás mintákat, mint aki be van jelentkezve. Amennyiben ez az érték 80% alatt van, a rendszer automatikusan kijelentkezteti a felhasználót.



8. ábra Adatgyűjtés első lépése



9. ábra Adatgyűjtés második lépése



10. ábra Az adatgyűjtés harmadik lépése

## Felhasználói adatok megjelenítése

### Adminisztrátori mód

A `ListDataFromFirebaseActivity` megjelenít egy listát statisztikákkal a rendszer felhasználóiról. Az Activity egy `RecyclerView`-t használ fel az adatok megjelenítéséhez. Ennek az adaptere jeleníti meg az Activity-től kapott sajátos objektumokban (`UserStatssObject`) tárolt adatokat. Ezek az adatok tartalmazzák a felhasználók azonosítóját, e-mail címét, az eszközöket, amelyekről történt a mintavételezés, a szessziók számát és a felhasználó által feltöltött fájlok számát.

Mivel ezek a statisztikák érzékenyebb adatokat is tartalmaz, mint például a felhasználó e-mail címe, illetve olyan adatokat, amelyek az adatbázisban fontos szerepet töltenek be (`userID`), az Activity indítása menüből csak adminisztrátori jogokkal rendelkező felhasználók számára elérhető.

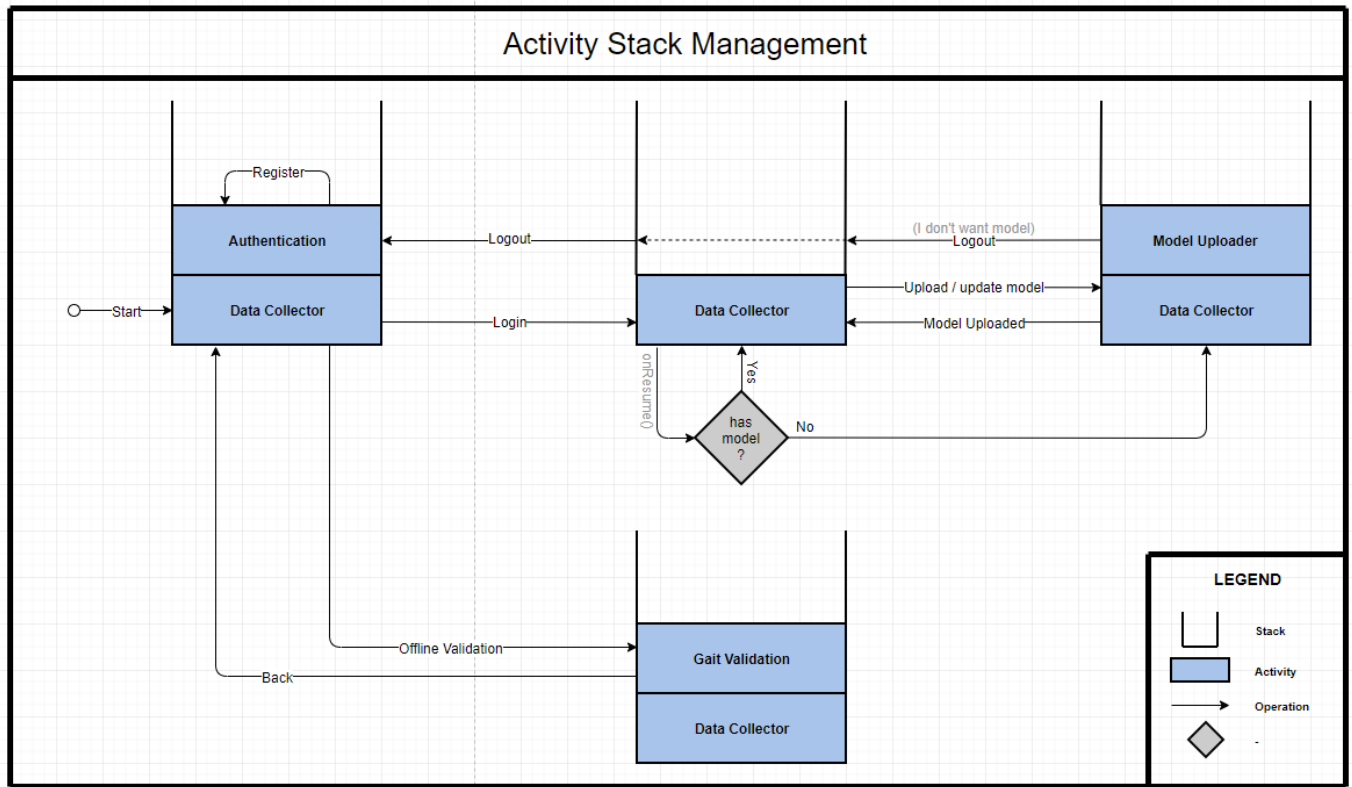
Az adatok Firebase Cloud Firestore-ban vannak tárolva. Ezek az adatok az Activity-n belül vannak lekérdezve, feldolgozva és továbbítva a `RecyclerView` adapterébe. A megjelenített adatok két oszlop formájába vannak rendezve, az első oszlop tartalmazza a másodikban levő adatok jelentését. A különböző felhasználókhöz tartozó adatok egy sajátos dekorátorral vannak elválasztva (`ItemDividerDecoration`), ami egy vízszintes zöld vonalat húz.

### Felhasználói mód

A felhasználói mód minden regisztrált felhasználó számára elérhető. Viszont minden felhasználó csak a saját adatait és statisztikáit láthatja, ellentétben az adminisztrátori módban

létező listával. A megjelenített adatai az e-mail címe, felhasználói név, egy fénykép, a szolgáltatott fájlok száma és szessziók száma. Ezen kívül biztosítva van egy felület, ahol megváltoztathatja a profilképét, felhasználói nevét és jelszavát. A e-mail cím szerkesztésére nem adunk lehetőséget, mert az adatbázisba ennek segítségével azonosítjuk a felhasználót. A változtatásokat Firebase-be.

### Activity Stack Management



11. ábra Az Activity Stack Management állapotai

Androidban a felületek úgynevezett „Activity”-k segítségével tudnak létezni. Ezek egymás utáni gyors váltogatását az Activity Stack Management vezérli. A 10. ábrán látható, hogy a felhasználó melyik cselekedete után az alkalmazás melyik állapotába kerül.