

UNIVERZITA PARDUBICE

FAKULTA ELEKTROTECHNIKY A
INFORMATIKY

BAKALÁŘSKÁ PRÁCE

2025

Evgenii Kuzmov

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Mobilní aplikace pro výzkum pohybových vzorců
Bakalářská práce

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2023/2024

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Evgenii Kuzmov**
Osobní číslo: **I21181**
Studijní program: **B0688A140009 Informační technologie**
Téma práce: **Mobilní aplikace pro výzkum pohybových vzorců**
Zadávající katedra: **Katedra informačních technologií**

Zásady pro vypracování

Cílem bakalářské práce je vytvoření mobilní aplikace pro operační systém android, která bude zachycovat údaje ze senzorů mobilu pro vyhodnocení pohybových vzorců uživatele. Zachycená data budou opatřena popisky, anonymizována a zaslána pro další zpracování na server.

Rozsah pracovní zprávy: **30**
Rozsah grafických prací:
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

OSE, S. – KUNDU, A. – MUKHERJEE, M. – BENERJEE, M. A comparative study: Java vs Kotlin programming in android application development. International Journal of advanced research in computer science, Volume 9, No. 3, ISSN 0976-5697
SIMS, Gary. Android authority [online]. Dostupné z: <https://www.androidauthority.com/develop-android-apps-languages-learn-391008/>
HARKIRAN, Kaur. Top Programming Languages for Android App Development Dostupné z: <https://www.geeksforgeeks.org/top-programming-languages-for-android-app-development/>

Vedoucí bakalářské práce: **Ing. Martin Pozdílek, Ph.D.**
Katedra informačních technologií

Datum zadání bakalářské práce: **15. prosince 2023**
Termín odevzdání bakalářské práce: **10. května 2024**

Ing. Zdeněk Němec, Ph.D. v.r.
děkan

L.S.

Ing. Jan Panuš, Ph.D. v.r.
vedoucí katedry

V Pardubicích dne 28. února 2024

Prohlašuji:

Práci s názvem mobilní aplikace pro výzkum pohybových vzorců jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 15. 05. 2025

Evgenii Kuzmov

PODĚKOVÁNÍ

Rád bych vyjádřil své díky vedoucímu bakalářské práce Ing. Martinu Pozdílkovi, Ph.D. za jeho odborné vedení a rady při vypracování této práce. Dále bych chtěl poděkovat své rodině za neustálou podporu, povzbuzení a pochopení během celého studia, zejména v náročných chvílích při dokončování této práce. V neposlední řadě děkuji také spolužákům a přátelům za jejich morální podporu a užitečné diskuze, které mi pomohly překonat mnohé překážky při vývoji aplikace.

ANOTACE

Cílem bakalářské práce je vytvoření mobilní aplikace pro operační systém Android, která umožňuje zachycovat, analyzovat a vizualizovat údaje ze senzorů mobilního telefonu pro vyhodnocení pohybových vzorců uživatele. Práce si klade za cíl především ověřit, zda je možné pomocí běžně dostupných mobilních zařízení získávat kvalitní a informativně hodnotná data, která by mohla sloužit jako podklad pro další analýzu, například pro včasnou detekci pohybových poruch nebo sledování efektivity rehabilitačního procesu. Zachycená data jsou opatřena popisky, anonymizována a bezpečně odeslána na server pro další zpracování, s důrazem na ochranu osobních údajů uživatelů.

KLÍČOVÁ SLOVA

Mobilní aplikace, Kotlin, databáze, Firebase, Python, API, Azure, pohyb, senzor

TITLE

Mobile app for motion pattern research

ANNOTATION

The aim of this bachelor thesis is to develop a mobile application for the Android operating system that enables capturing, analyzing, and visualizing data from mobile phone sensors to evaluate user movement patterns. The thesis primarily aims to verify whether it is possible to obtain quality and informatively valuable data using commonly available mobile devices, which could serve as a basis for further analysis, for example, for early detection of movement disorders or monitoring the effectiveness of rehabilitation processes. The captured data is labeled, anonymized, and securely sent to a server for further processing, with emphasis on protecting users' personal information.

KEYWORDS

Mobile application, Kotlin, database, Firebase, Python, API, Azure, movement, sensor

OBSAH

SEZNAM ILUSTRACÍ A TABULEK.....	11
SEZNAM ZKRATEK A ZNAČEK	12
ÚVOD.....	14
1 TEORETICKÁ ČÁST	15
1.1 Pohybové vzorce člověka	15
1.1.1 Definice a charakteristika pohybových vzorců.....	15
1.1.2 Význam analýzy pohybových vzorců v různých oborech	15
1.1.2.1 Medicína a rehabilitace	15
1.1.2.2 Sport a sportovní věda	16
1.1.2.3 Ergonomie a pracovní medicína	16
1.1.2.4 Umělá inteligence a vývoj pokročilých protetických pomůcek.....	16
1.1.3 Metody sledování a vyhodnocování pohybových vzorců.....	16
1.1.3.1 Laboratorní metody.....	16
1.1.3.2 Nositelné senzory	17
1.1.3.3 Mobilní technologie	17
1.1.3.4 Počítačové zpracování a analýza dat	17
1.2 Mobilní senzory a jejich využití pro analýzu pohybu.....	18
1.2.1 Přehled senzorů v moderních mobilních zařízeních	18
1.2.1.1 Akcelerometr	18
1.2.1.2 Gyroskop.....	18
1.2.1.3 Rotační vektor.....	19
1.2.1.4 Krokomeř	19
1.2.1.5 GPS (Global Positioning System).....	19
1.2.1.6 Magnetometr	20
1.2.1.7 Barometr	20
1.2.1.8 Senzor přiblížení (Proximity sensor)	20
1.2.1.9 Teploměr	20
1.2.1.10 Senzor okolního světla.....	20
1.2.2 Přesnost a limity mobilních senzorů při sběru dat.....	20
1.2.2.1 Faktory ovlivňující přesnost	20
1.2.2.2 Kompenzační techniky a algoritmy	21
1.2.2.3 Limitace mobilních senzorů pro analýzu pohybu.....	22

1.3 Zpracování a analýza dat z mobilních senzorů	22
1.3.1 Metody zpracování surových dat ze senzorů	22
1.3.1.1 Předzpracování signálu	23
1.3.1.2 Extrakce příznaků	24
1.3.2 Algoritmy pro rozpoznávání pohybových vzorců	24
1.3.2.1 Metody založené na pravidlech	25
1.3.2.2 Statistické metody a strojové učení	25
1.3.2.3 Metody pro detekci specifických událostí	26
1.4 Rešerše	27
1.4.1 Movesense Showcase app	27
1.4.2 Gait Analyzer	27
1.4.3 Porovnání s vyvíjenou aplikací	28
2 PRAKTICKÁ ČÁST	29
2.1 Cíle práce	29
2.1.1 Definice cílů práce	29
2.1.2 Funkční požadavky	29
2.1.3 Nefunkční požadavky	30
2.2 Analýza	31
2.2.1 Diagram případů užití	31
2.2.2 Datový model	32
2.2.3 Popis procesů (BPM)	36
2.3 Návrh	39
2.3.1 Architektura aplikace	39
2.3.2 Použité technologie	41
2.3.2.1 Backend	41
2.3.2.2 Frontend	41
2.3.2.3 Vývojová prostředí	42
2.4 Implementace	43
2.4.1 Databáze	43
2.4.1.1 Konfigurace Firebase	43
2.4.1.2 Struktura a přístup k datům	43
2.4.1.3 Firebase API a datový přístup	44
2.4.1.4 Základní databázové operace	44

2.4.2 Serverová část	46
2.4.2.1 Zpracování dat ze senzorů	46
2.4.2.2 Anonymizace a ochrana osobních údajů	47
2.4.2.3 Ukládání a správa dat.....	48
2.4.2.4 Python API v Azure	48
2.4.3 Uživatelská část	50
2.4.4 Rozhraní aplikace	52
2.4.4.1 Přihlašování a registrace	52
2.4.4.2 Hlavní obrazovka	53
2.4.4.3 Obrazovka senzorů	54
2.4.4.4 Seznam aktivit.....	55
2.4.4.5 Uživatelská obrazovka	56
2.4.4.6 Administrátorská obrazovka	57
2.5 Další možný rozvoj aplikace.....	59
ZÁVĚR	61
POUŽITÁ LITERATURA	62
SEZNAM PŘÍLOH.....	64

SEZNAM ILUSTRACÍ A TABULEK

Obrázek 1: Diagram případu užití (zdroj – vlastní)	32
Obrázek 2: Datový model (zdroj – vlastní)	35
Obrázek 3: Proces autentizace (zdroj – vlastní).....	36
Obrázek 4: Sběr dat o pohybových vzorcích (zdroj – vlastní)	37
Obrázek 5: Správa naměřených dat (zdroj – vlastní).....	38
Obrázek 6: Správa uživatelského účtu (zdroj – vlastní)	38
Obrázek 7: Administrátorské procesy (zdroj – vlastní).....	39
Obrázek 8: Architektura aplikace (zdroj – vlastní).....	40
Obrázek 9: Architektura uživatelského rozhraní (zdroj – vlastní).....	51
Obrázek 10: Přihlašovací obrazovka (zdroj – vlastní).....	53
Obrázek 11: Registrační obrazovka (zdroj – vlastní)	53
Obrázek 12: Hlavní obrazovka – konfigurace (zdroj – vlastní).....	54
Obrázek 13: Hlavní obrazovka (zdroj – vlastní).....	54
Obrázek 14: Obrazovka senzorů (zdroj – vlastní)	55
Obrázek 15: Seznam aktivit (zdroj – vlastní)	56
Obrázek 16: Seznam aktivit – graf (zdroj – vlastní).....	56
Obrázek 17: Uživatelská obrazovka (zdroj – vlastní).....	57
Obrázek 18: Administrátorská obrazovka (zdroj – vlastní).....	58
Obrázek 19: Administrátorská obrazovka – správa aktivit (zdroj – vlastní)	58
Tabulka 1: Funkční požadavky (zdroj vlastní)	30
Tabulka 2: Nefunkční požadavky (zdroj – vlastní)	30

SEZNAM ZKRATEK A ZNAČEK

3D	Three-dimensional
AI	Artificial Intelligence
API	Application Programming Interface
APK	Android Package Kit
AVD	Android Virtual Device
BPM	Business Process Management
CNN	Convolutional Neural Network
CRUD	Create, Read, Update, Delete
CSV	Comma-Separated Values
EMA	Exponential Moving Average
EMG	Elektromyografie
FastAPI	Fast Application Programming Interface
GPS	Global Positioning System
HMM	Hidden Markov Model
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IaaS	Infrastructure as a Service
IDE	Integrated Development Environment
IMU	Inertial Measurement Unit
JVM	Java Virtual Machine
LSTM	Long Short-Term Memory
MVVM	Model-View-ViewModel
NoSQL	Not only Structured Query Language
PaaS	Platform as a Service
PDF	Portable Document Format
REST	Representational State Transfer
RNN	Recurrent Neural Network
SaaS	Software as a Service
SDK	Software Development Kit

STFT	Short-Time Fourier Transform
SVM	Support Vector Machines
UI	User Interface
XML	eXtensible Markup Language

ÚVOD

V současné době, kdy mobilní zařízení obsahují stále více senzorů a jsou běžnou součástí každodenního života, narůstá potenciál využití těchto technologií pro sběr a analýzu dat o pohybových vzorcích člověka. Tato data mohou mít významnou hodnotu v mnoha oborech, od sportovní medicíny přes rehabilitační péči až po výzkum neurologických onemocnění. Právě schopnost detekovat a analyzovat změny v pohybových vzorcích může být klíčovým faktorem při včasné diagnostice různých zdravotních problémů nebo při monitorování efektivity rehabilitačního procesu.

Cílem této bakalářské práce je ověřit, zda je možné pomocí běžně dostupných mobilních zařízení získávat kvalitní a informativně hodnotná data o pohybu, která by mohla sloužit jako podklad pro další analýzu, například pro včasnou detekci pohybových poruch nebo sledování efektivity rehabilitačního procesu. Toho zjistíme vytvořením mobilní aplikace pro operační systém Android a ukázkou dat v grafické podobě. Aplikace je navržena s důrazem na jednoduchost ovládání a přístupnost pro širokou skupinu uživatelů včetně seniorů, což se odráží v použití vyšší minimální verze SDK Android 26, která zajišťuje kompatibilitu i se staršími zařízeními. Získaná data jsou následně opatřena popisky, anonymizována a bezpečně odeslána na server implementovaný v programovacím jazyce Python a nasazený v cloudovém prostředí Azure, kde jsou dále zpracována a vizualizována do podoby informativních grafů.

V teoretické části se zaměřuji na problematiku sběru dat ze senzorů mobilních zařízení, jejich zpracování a možnosti využití pro analýzu pohybových vzorců. Praktická část pak popisuje proces vývoje aplikace od návrhu přes implementaci až po testování a vyhodnocení nasbíraných dat. Zvláštní pozornost je věnována návrhu uživatelského rozhraní, které musí být intuitivní a přístupné i pro uživatele s nižší úrovní technologické gramotnosti.

Potenciální přínos této práce spočívá v možnosti využití vytvořené aplikace jako dostupného nástroje pro sběr dat o pohybových vzorcích v různých výzkumných oblastech, od sportovní medicíny přes rehabilitaci až po studium neurologických poruch. V dlouhodobém horizontu by data získaná pomocí podobných aplikací mohla být využita pro trénink algoritmů umělé inteligence, které by dokázaly detekovat abnormality v pohybových vzorcích a přispět tak k včasné diagnostice některých onemocnění.

1 TEORETICKÁ ČÁST

1.1 Pohybové vzorce člověka

1.1.1 Definice a charakteristika pohybových vzorců

Pohybové vzorce lze definovat jako opakující se, koordinované sekvence pohybů, které člověk vykonává při různých aktivitách. Jedná se o komplexní soubor motorických dovedností a jejich různých kombinací, které umožňují lidskému tělu efektivně interagovat s okolním prostředím. Pohybové vzorce zahrnují jak jednoduché pohyby (například chůze, běh, zvedání předmětů), tak složitější motorické sekvence (sportovní aktivity, tanec, specializované pracovní úkony). [1]

Z biomechanického hlediska je každý pohybový vzorec charakterizován několika klíčovými parametry:

- **Prostorové charakteristiky** – trajektorie pohybu, rozsah pohybu, pozice těla a jeho částí v prostoru
- **Časové charakteristiky** – rychlost, zrychlení, frekvence pohybu, rytmus
- **Silové charakteristiky** – intenzita svalové kontrakce, působení gravitace, reakční síly
- **Koordinační charakteristiky** – časování a synchronizace jednotlivých segmentů těla při pohybu

Pohybové vzorce se u každého jedince vyvíjejí od raného dětství a jsou determinovány kombinací genetických faktorů, fyzického vývoje a učení. V průběhu života se postupně automatizují a stabilizují, ale zároveň si zachovávají určitou variabilitu a adaptabilitu, což umožňuje přizpůsobení se měnícím se podmínkám prostředí. [1]

1.1.2 Význam analýzy pohybových vzorců v různých oborech

1.1.2.1 Medicína a rehabilitace

V medicínské praxi má analýza pohybových vzorců zásadní význam pro diagnostiku, léčbu a rehabilitaci různých onemocnění. Změny v pohybových vzorcích mohou signalizovat neurologické poruchy, muskuloskeletální problémy nebo jiná onemocnění. Například:

- U Parkinsonovy nemoci lze pozorovat typické změny v chůzi, jako je zmenšení délky kroku, snížení výšky zdvihu chodidla a celkové zpomalení pohybu
- Cévní mozkové příhody často vedou k asymetrii pohybů a změnám v koordinaci

- Muskuloskeletální poruchy se projevují kompenzačními pohybovými vzorci, které mohou vést k dalším zdravotním problémům

V rehabilitační medicíně je analýza pohybových vzorců klíčovým nástrojem pro hodnocení efektivity léčebných postupů a sledování průběhu rehabilitace. Umožňuje terapeutům přesněji cílit léčebné intervence a poskytuje objektivní metodu pro hodnocení zlepšení stavu pacienta. [3]

1.1.2.2 Sport a sportovní věda

Ve sportu je analýza pohybových vzorců využívána pro optimalizaci výkonu, prevenci zranění a rozvoj specifických dovedností. Sportovní vědci a trenéři analyzují pohybové vzorce vrcholových sportovců, aby identifikovali klíčové faktory úspěchu v dané disciplíně a mohli tyto poznatky aplikovat v tréninkovém procesu. [3]

Například ve vrcholovém běhu může detailní analýza pohybových vzorců odhalit neefektivní techniku, která zvyšuje energetickou náročnost běhu nebo zvyšuje riziko zranění. Na základě těchto poznatků lze pak upravit tréninkový plán a techniku pohybu. [3]

1.1.2.3 Ergonomie a pracovní medicína

V oblasti ergonomie se analýza pohybových vzorců využívá pro design pracovního prostředí a nástrojů, které minimalizují fyzickou zátěž a riziko zdravotních problémů souvisejících s prací. Studie pohybových vzorců při různých pracovních činnostech pomáhají identifikovat rizikové faktory muskuloskeletálních poruch a navrhnout preventivní opatření. [2]

1.1.2.4 Umělá inteligence a vývoj pokročilých protetických pomůcek

S rozvojem umělé inteligence a robotiky nabývá analýza pohybových vzorců na významu i v oblasti vývoje protetických pomůcek a exoskeletonů. Detailní porozumění lidským pohybovým vzorcům umožňuje vytvářet pokročilé protézy a asistenční zařízení, která přirozeně a efektivně napodobují přirozený lidský pohyb. [3]

1.1.3 Metody sledování a vyhodnocování pohybových vzorců

1.1.3.1 Laboratorní metody

Optické motion capture systémy představují zlatý standard pro analýzu pohybu v laboratorních podmínkách. Využívají vysokorychlostní kamery a reflexní nebo aktivní markery umístěné na klíčových bodech těla. Tyto systémy poskytují vysoce přesná 3D data o pohybu, ale jsou finančně nákladné a vyžadují specializované vybavení a prostředí. [4]

Elektromyografie (EMG) měří elektrickou aktivitu svalů během pohybu a poskytuje informace o timingu a intenzitě svalové aktivace. EMG je často kombinována s dalšími metodami pro komplexnější analýzu pohybových vzorců. [4]

Silové plošiny a tlaková měření umožňují analýzu reakčních sil a distribuce tlaku při stoji, chůzi nebo jiných aktivitách. Tyto informace jsou cenné především pro analýzu chůze a posturální stability. [4]

1.1.3.2 Nositelné senzory

Inerciální měřicí jednotky (IMU) kombinují akcelerometry, gyroskopy a někdy i magnetometry pro měření zrychlení, úhlové rychlosti a orientace segmentů těla. IMU jsou relativně dostupné, kompaktní a umožňují měření pohybu v přirozeném prostředí. [5]

Chytré hodinky a fitness trackery využívají zjednodušené verze IMU senzorů a další senzory pro sledování základních parametrů pohybové aktivity jako je počet kroků, rychlost pohybu nebo srdeční frekvence. Ačkoliv neposkytují tak detailní data jako specializované systémy, jejich dostupnost a uživatelská přívětivost z nich činí cenný nástroj pro dlouhodobé sledování pohybové aktivity v běžném životě. [5]

1.1.3.3 Mobilní technologie

Smartphony obsahují řadu senzorů včetně akcelerometrů, gyroskopů a GPS, které umožňují sběr dat o pohybu. Díky své všudypřítomnosti a výpočetnímu výkonu představují potenciálně revoluční nástroj pro masivní sběr dat o pohybových vzorcích v reálných podmínkách běžného života. [4]

Mobilní aplikace dokáží zpracovávat data ze senzorů smartphonů a poskytovat zpětnou vazbu uživatelům. Zároveň mohou tato data anonymizovat a odesílat k dalšímu zpracování, čímž vytvářejí rozsáhlé databáze pohybových vzorců, které lze využít pro výzkum a vývoj pokročilých algoritmů pro analýzu pohybu. [4]

1.1.3.4 Počítačové zpracování a analýza dat

Kinematická a kinetická analýza zpracovává data o pozici, rychlosti, zrychlení a působících silách během pohybu. Umožňuje kvantifikovat parametry pohybu a identifikovat případné odchylky od normálních pohybových vzorců. [5]

Statistické metody a strojové učení se využívají pro analýzu velkých objemů dat o pohybu a identifikaci vzorů a trendů, které by mohly být klinicky relevantní. Algoritmy strojového učení

dokážou například klasifikovat pohybové vzorce podle jejich normality nebo detekovat změny v pohybových vzorcích, které by mohly signalizovat zdravotní problémy. [5]

Vizualizační techniky hrají klíčovou roli v prezentaci výsledků analýzy pohybových vzorců srozumitelnou formou. Patří mezi ně například 3D animace pohybu, grafy zobrazující klíčové parametry pohybu v čase nebo heat mapy znázorňující distribuci zatížení. [5]

1.2 Mobilní senzory a jejich využití pro analýzu pohybu

1.2.1 Přehled senzorů v moderních mobilních zařízeních

Moderní chytré telefony jsou vybaveny řadou senzorů, které umožňují detekci pohybu, orientace a polohy zařízení. Tyto senzory poskytují data, která lze využít pro analýzu pohybových vzorců uživatele. Následující přehled popisuje hlavní senzory relevantní pro analýzu pohybu.

1.2.1.1 Akcelerometr

Akcelerometr je základním senzorem pro detekci pohybu v mobilních zařízeních. Měří lineární zrychlení zařízení ve třech osách (X, Y, Z). Akcelerometr dokáže detekovat:

- Gravitační zrychlení (stálá síla směřující k zemi)
- Zrychlení vyvolané pohybem zařízení
- Vibrace a otřesy

Výstupem akcelerometru jsou hodnoty zrychlení v jednotkách m/s^2 pro každou osu. Tyto hodnoty lze využít k detekci základních pohybových aktivit jako je chůze, běh nebo pád. Akcelerometrická data jsou také základem pro krokoměry a detekci orientace zařízení. [7]

1.2.1.2 Gyroskop

Gyroskop měří úhlovou rychlost (rychlost rotace) zařízení kolem tří os. Na rozdíl od akcelerometru, který detekuje lineární zrychlení, gyroskop zachycuje rotační pohyby. Výstupem jsou hodnoty v jednotkách rad/s pro každou osu (X, Y, Z). Gyroskop je klíčový pro:

- Přesnou detekci rotace zařízení
- Stabilizaci obrazu v aplikacích fotoaparátu
- Rozpoznávání gest a pohybových vzorců vyžadujících rotaci

Kombinace dat z akcelerometru a gyroskopu poskytuje komplexnější obraz o pohybu zařízení a umožňuje přesnější analýzu pohybových vzorců. [7]

1.2.1.3 Rotační vektor

Rotační vektor (Rotation Vector) je virtuální senzor, který kombinuje data z akcelerometru, gyroskopu a magnetometru pro poskytnutí přesné informace o orientaci zařízení v prostoru. Představuje rotaci zařízení jako kombinaci úhlu a osy, kolem které rotace probíhá. Tento senzor je obzvláště užitečný pro: [7]

- Přesné sledování orientace zařízení v 3D prostoru
- Aplikace rozšířené reality
- Analýzu komplexních pohybových vzorců vyžadujících informaci o absolutní orientaci

1.2.1.4 Krokoměr

Krokoměr (Step Counter) je virtuální senzor založený především na datech z akcelerometru. Využívá algoritmy pro detekci charakteristického vzoru zrychlení při kroku a poskytuje počet kroků vykonaných od posledního restartu zařízení. Moderní krokoměry dosahují vysoké přesnosti a jsou energeticky efektivní, protože zpracování dat probíhá na nízkourovňovém hardwaru. [7]

Krokoměr je užitečný pro:

- Sledování fyzické aktivity
- Výpočet ušlé vzdálenosti
- Odhad spálených kalorií

1.2.1.5 GPS (Global Positioning System)

GPS senzor poskytuje informace o geografické poloze zařízení prostřednictvím triangulace signálů z družic. V kontextu analýzy pohybu GPS umožňuje:

- Sledování trasy pohybu
- Měření rychlosti pohybu
- Určení nadmořské výšky
- Analýzu pohybových vzorců v geografickém kontextu

GPS data doplňují informace z inerciálních senzorů a poskytují kontext o prostředí, ve kterém pohyb probíhá. [7]

1.2.1.6 Magnetometr

Magnetometr měří sílu a směr magnetického pole kolem zařízení. Primárně slouží jako kompas, ale v kombinaci s dalšími senzory zlepšuje přesnost detekce orientace zařízení. Je zásadní pro fungování rotačního vektoru a významně zlepšuje detekci absolutní orientace zařízení v prostoru. [6]

1.2.1.7 Barometr

Barometr měří atmosférický tlak, což umožňuje přibližné určení nadmořské výšky. V analýze pohybu může barometr pomoci detekovat: [6]

- Změny výšky (výstupy po schodech, výtah)
- Doplnkové informace pro přesnější určení výdeje energie při pohybu v různých výškových profilech

1.2.1.8 Senzor přiblížení (Proximity sensor)

Ačkoliv primárně slouží k detekci přiblížení objektu k obrazovce, může v některých případech poskytnout doplňkové informace o způsobu držení zařízení během pohybových aktivit. [6]

1.2.1.9 Teploměr

Některá zařízení jsou vybavena teploměrem, který měří teplotu okolí nebo zařízení. Tyto údaje mohou doplňovat analýzu pohybu o kontext prostředí, ve kterém pohyb probíhá, a pomoci při interpretaci dat (například vyšší teplota může ovlivnit výkon při fyzické aktivitě). [6]

1.2.1.10 Senzor okolního světla

Měří intenzitu okolního světla a primárně slouží pro automatickou úpravu jasu displeje. V analýze pohybu může poskytnout informaci o tom, zda se uživatel pohybuje venku nebo uvnitř, případně v jakém světelném prostředí aktivita probíhá. [6]

1.2.2 Přesnost a limity mobilních senzorů při sběru dat

1.2.2.1 Faktory ovlivňující přesnost

Kvalita senzorů

Kvalita senzorů se liší mezi různými modely zařízení, což vede k rozdílům v přesnosti a spolehlivosti měření. Cenově dostupnější zařízení často obsahují méně přesné senzory s větším šumem. [9]

Drift gyroskopu

Gyroskop trpí problémem driftu – postupné odchylky měřených hodnot od skutečných. Při dlouhodobém měření se tyto chyby kumulují a vedou k nepřesnostem v určení orientace. [9]

Kalibrace senzorů

Správná kalibrace senzorů je klíčová pro přesné měření. Magnetometr je obzvláště citlivý na okolní magnetické rušení a vyžaduje pravidelnou kalibraci. [9]

Umístění a orientace zařízení

Poloha a orientace zařízení během měření významně ovlivňuje naměřená data. Pokud uživatel drží telefon v ruce, v kapse nebo má připevněný na těle, získaná data se mohou výrazně lišit. [8]

Vnější vlivy

Přesnost měření ovlivňují i vnější faktory:

- GPS signál je omezen nebo nedostupný uvnitř budov
- Magnetická pole v okolí zařízení ovlivňují magnetometr
- Teplota může ovlivnit přesnost některých senzorů

1.2.2.2 Kompenzační techniky a algoritmy

Fúze senzorů

Kombinování dat z více senzorů (sensor fusion) umožňuje kompenzovat nedostatky jednotlivých senzorů. Nejčastěji se využívá Kalmanův filtr nebo komplementární filtr pro kombinaci dat z akcelerometru, gyroskopu a magnetometru. [9]

Filtrování šumu

Surová data ze senzorů obsahují šum, který je třeba filtrovat pro získání relevantních informací. Používají se různé typy filtrů (nízkofrekvenční filtr, mediánový filtr, Kalmanův filtr) podle typu aplikace a požadované odezvy. [9]

Statistické metody a strojové učení

Pro zpracování a interpretaci dat ze senzorů se stále častěji využívají pokročilé statistické metody a algoritmy strojového učení, které dokážou: [9]

- Klasifikovat typ aktivity z dat senzorů
- Detekovat anomálie v pohybových vzorcích
- Kompenzovat nepřesnosti a šum v datech

1.2.2.3 Limitace mobilních senzorů pro analýzu pohybu

Nízká přesnost oproti profesionálním systémům

Mobilní senzory nedosahují přesnosti specializovaných systémů pro analýzu pohybu (motion capture, profesionální IMU jednotky). Poskytují však dostatečnou přesnost pro mnoho praktických aplikací. [8]

Omezenost dat o tělesných segmentech

Mobilní zařízení typicky poskytuje data pouze o pohybu jednoho bodu (kde je zařízení umístěno), nikoli o pohybu všech segmentů těla. To omezuje možnosti komplexní analýzy biomechaniky pohybu. [8]

Variabilita umístění zařízení

Uživatelé nosí mobilní zařízení různými způsoby (v kapse, v ruce, v batohu), což komplikuje standardizaci měření a interpretaci dat.

Energetická omezení

Kontinuální sběr dat ze senzorů významně zatěžuje baterii zařízení, což limituje délku měření bez dobíjení.

I přes uvedená omezení představují mobilní senzory revoluční nástroj pro sběr dat o pohybu v reálných podmínkách běžného života. Jejich všudypřítomnost, relativně nízká cena a stále se zlepšující technické parametry z nich činí ideální platformu pro masivní sběr dat o pohybových vzorcích, který by byl s profesionálními systémy ekonomicky a logisticky neproveditelný. [8]

1.3 Zpracování a analýza dat z mobilních senzorů

1.3.1 Metody zpracování surových dat ze senzorů

Surová data z mobilních senzorů vyžadují několik kroků zpracování, než je lze efektivně využít pro analýzu pohybových vzorců. Tento proces typicky zahrnuje následující metody:

1.3.1.1 Předzpracování signálu

Odstranění šumu

Surová data ze senzorů obsahují šum, který může být způsoben elektrickými interferencemi, vibracemi nebo teplotními fluktuacemi. Pro jeho odstranění se používají různé filtrovací techniky: [11]

- **Nízkofrekvenční (low-pass) filtry** odstraňují vysokofrekvenční šum a ponechávají pomalejší změny signálu
- **Mediánové filtry** jsou účinné proti impulsnímu šumu (náhlé špičky v datech)
- **Gaussovské vyhlazování** používá váženou průměrovou funkci pro vyhlazení signálu
- **Vlnková transformace (wavelet transform)** umožňuje odstranění šumu s ohledem na zachování důležitých charakteristik signálu

Kalibrace a normalizace

Data ze senzorů je třeba kalibrovat a normalizovat, aby byla srovnatelná mezi různými zařízeními: [11]

- **Odstranění offsetu** – posunutí hodnot tak, aby odpovídaly očekávanému rozsahu
- **Škálování** – úprava rozsahu hodnot na standardní interval
- **Korekce driftu** – kompenzace postupného posunu hodnot v čase
- **Převzorkování (resampling)** – úprava vzorkovací frekvence na konstantní hodnotu pro konzistentní analýzu

Segmentace dat

Pro další analýzu je často užitečné rozdělit kontinuální proud dat na menší úseky:

- **Segmentace s pevným oknem** – rozdělení dat na úseky konstantní délky
- **Adaptivní segmentace** – rozdělení dat na základě detekce změn v charakteristikách signálu
- **Segmentace založená na událostech** – identifikace úseků odpovídajících specifickým událostem (krok, otočení, změna aktivity)

1.3.1.2 Extrakce příznaků

Z předzpracovaných dat je třeba extrahovat relevantní příznaky (features), které charakterizují pohybové vzorce. Příznaky lze rozdělit do několika kategorií: [10]

Časové příznaky

- **Statistické momenty** – průměr, směrodatná odchylka, šikmost, špičatost
- **Percentily** – medián, kvantily
- **Průchody nulou** – počet průchodů signálu nulovou hodnotou
- **Extremální hodnoty** – maxima, minima, rozsah hodnot
- **Korelace** – autokorelace, křížová korelace mezi různými osami

Frekvenční příznaky

Získané transformací časových dat do frekvenční domény pomocí Fourierovy transformace: [10]

- **Dominantní frekvence** – frekvence s nejvyšší amplitudou
- **Spektrální entropie** – míra nepředvídatelnosti spektra
- **Spektrální výkon v pásmech** – energie signálu v definovaných frekvenčních pásmech
- **Spektrální centroid** – těžiště spektra

Časově-frekvenční příznaky

Získané analýzou, jak se frekvenční charakteristiky signálu mění v čase: [10]

- **Krátkodobá Fourierova transformace (STFT)** – analýza frekvencí v krátkých časových úsecích
- **Vlnková transformace (Wavelet Transform)** – dekompozice signálu s různým časovým rozlišením
- **Hilbert-Huangova transformace** – adaptivní analýza nelineárních a nestacionárních dat

1.3.2 Algoritmy pro rozpoznávání pohybových vzorců

Pro rozpoznávání a klasifikaci pohybových vzorců z mobilních senzorů se využívá široká škála algoritmů strojového učení a statistických metod:

1.3.2.1 Metody založené na pravidlech

Nejjednodušší přístup využívá předdefinovaná pravidla a prahové hodnoty pro klasifikaci pohybu:

- **Prahování signálu** – např. detekce kroků na základě překročení prahové hodnoty zrychlení
- **Stavové automaty** – modelování přechodů mezi různými fázemi pohybu
- **Heuristické algoritmy** – specifické postupy navržené na základě expertní znalosti o daném typu pohybu

Tyto metody jsou výpočetně nenáročné a vhodné pro implementaci přímo na mobilních zařízeních, ale mají omezenou přesnost a adaptabilitu na různé uživatele. [12]

1.3.2.2 Statistické metody a strojové učení

Klasifikační algoritmy

- **K-nejbližších sousedů (k-NN)** – klasifikace na základě podobnosti s trénovacími příklady
- **Rozhodovací stromy a náhodné lesy** – hierarchická klasifikace založená na pravidlech odvozených z trénovacích dat
- **Support Vector Machines (SVM)** – hledání optimální hranice mezi třídami v prostoru příznaků
- **Naivní Bayesovský klasifikátor** – pravděpodobnostní model založený na Bayesově teorému [12]

Neuronové sítě a hluboké učení

- **Konvoluční neuronové sítě (CNN)** – schopné automaticky extrahovat relevantní příznaky ze surových dat
- **Rekurentní neuronové sítě (RNN)** – vhodné pro sekvenční data díky schopnosti modelovat temporální závislosti
- **Long Short-Term Memory (LSTM)** – speciální typ RNN navržený pro dlouhodobé závislosti v datech
- **Autoenkodéry** – užitečné pro detekci anomálií v pohybových vzorcích [12]

1.3.2.3 Metody pro detekci specifických událostí

Detekce kroků

- **Detekce špiček** – identifikace lokálních maxim v signálu zrychlení
- **Zero-crossing** – detekce průchodů nulou po filtraci signálu
- **Spektrální analýza** – detekce periodicity odpovídající frekvenci kroků [13]

Rozpoznávání aktivit

- **Segmentační algoritmy** – rozdělení kontinuálního proudu dat na úseky odpovídající jednotlivým aktivitám
- **Hierarchické modely** – rozpoznávání složitějších aktivit na základě sekvence jednodušších pohybů
- **Skryté Markovovy modely (HMM)** – pravděpodobnostní modely pro sekvenční data [13]

Detekce anomálií

- **Statistické metody** – detekce odchylek od normálního rozdělení dat
- **Isolation Forest** – identifikace odlehlých hodnot na základě náhodného rozdělování prostoru
- **One-class SVM** – naučí se hranici normálního chování a detekuje odchylky
- **Rekonstrukční chyba autoenkodérů** – měření rozdílu mezi vstupními daty a jejich rekonstrukcí [13]

Efektivní zpracování a vizualizace dat z mobilních senzorů umožňuje transformovat surová data na smysluplné informace, které mohou být využity pro pochopení pohybových vzorců, detekci anomálií a poskytnutí užitečné zpětné vazby uživatelům. S rostoucím výpočetním výkonem mobilních zařízení a pokrokem v oblasti strojového učení se otevírají nové možnosti pro stále sofistikovanější analýzu pohybu přímo na mobilním zařízení nebo v kombinaci s cloudovými službami.

1.4 Rešerše

1.4.1 Movesense Showcase app

Movesense Showcase je mobilní aplikace určená pro interakci s hardwarovými senzory Movesense. Jde o aplikaci, která slouží jako demonstrační nástroj pro vývojáře a uživatele senzorů Movesense, které jsou malé, odolné a vodotěsné. Tyto senzory jsou schopné měřit pohyb, srdeční tep, teplotu a další biometrické údaje. [14]

Klíčové funkce aplikace zahrnují:

- Možnost připojení jednoho nebo dvou senzorů Movesense
- Streamování dat ze senzorů v reálném čase s automatickým ukládáním ve formátu CSV
- Prohlížení zaznamenaných dat přímo v telefonu
- Export zaznamenaných dat na Google Drive pro další analýzu
- Aktualizace firmwaru připojených senzorů

Hlavní specifikum této aplikace spočívá v tom, že vyžaduje externí hardware (senzory Movesense) pro plnohodnotné využití svých funkcí. Aplikace sama o sobě slouží především jako rozhraní pro přístup k datům z těchto externích senzorů a jako nástroj pro správu těchto zařízení. [14]

1.4.2 Gait Analyzer

Gait Analyzer je specializovaná aplikace zaměřená na analýzu chůze, která využívá integrované senzory chytrého telefonu (akcelerometr, gyroskop a magnetometr) pro výpočet parametrů chůze v reálném čase. Aplikace poskytuje širokou škálu metrik souvisejících s chůzí, jako je rychlost chůze, doba kroku, délka kroku, rychlost šlapání a symetrie. [15]

Mezi klíčové funkce aplikace patří:

- Výpočet parametrů chůze v reálném čase s využitím senzorů telefonu
- Režimy měření zahrnující časované pokusy, nepřetržité měření nebo celodenní monitorování na pozadí
- Možnost kombinovat analýzu chůze s kognitivními testy (např. Stroopův test)
- Export dat do CSV formátu nebo na Google Drive
- Možnost personalizace na základě výšky a dalších demografických údajů

- Analýza historických údajů o chůzi a kognitivních funkcích
- Generování vlastních zpráv ve formátu PDF

Gait Analyzer se zaměřuje specificky na analýzu parametrů chůze a jejich korelaci s kognitivními funkcemi, což naznačuje jeho potenciální využití v klinickém kontextu nebo pro výzkum v oblasti neurologie a rehabilitace. [15]

1.4.3 Porovnání s vyvíjenou aplikací

Vyvíjená aplikace pro výzkum pohybových vzorců se od analyzovaných existujících řešení odlišuje svým zaměřením na jednoduchý, ale efektivní sběr pohybových dat pro výzkumné účely. Na rozdíl od většiny dostupných aplikací, které jsou buď příliš specializované (Gait Analyzer), nebo vyžadují externí hardware (Movesense), naše aplikace využívá výhradně integrované senzory běžných mobilních telefonů, což ji činí dostupnou pro široké spektrum uživatelů bez dodatečných nákladů.

Klíčovým rozdílem je také cílová skupina – zatímco existující řešení jsou často navržena pro technicky zdatné uživatele nebo specifické klinické účely, naše aplikace je optimalizována s ohledem na přístupnost i pro starší uživatele nebo osoby s nižší technickou gramotností. Tomu odpovídá i zvolená minimální verze SDK Android 26, která zajišťuje kompatibilitu se staršími zařízeními, které mohou tito uživatelé vlastnit.

2 PRAKTICKÁ ČÁST

2.1 Cíle práce

Tato kapitola definuje konkrétní cíle bakalářské práce a stanovuje požadavky na mobilní aplikaci pro výzkum pohybových vzorců. Jsou zde specifikovány jak funkční požadavky popisující, co musí aplikace umožňovat, tak nefunkční požadavky určující kvalitativní charakteristiky výsledného systému. Definované cíle a požadavky slouží jako měřítko pro hodnocení úspěšnosti implementace a naplnění záměru práce.

2.1.1 Definice cílů práce

Hlavním cílem této bakalářské práce je vytvoření mobilní aplikace pro operační systém Android, která umožní efektivní sběr, zpracování a vizualizaci dat z integrovaných senzorů mobilního telefonu pro výzkum pohybových vzorců uživatelů. Práce si klade za cíl ověřit, zda je možné pomocí běžně dostupných mobilních zařízení získávat kvalitní a informativně hodnotná data o pohybu, která by mohla sloužit jako podklad pro další analýzu.

2.1.2 Funkční požadavky

V této části jsou definovány klíčové funkční požadavky na vyvíjenou mobilní aplikaci pro výzkum pohybových vzorců. Tyto požadavky stanovují, jaké konkrétní funkce musí aplikace poskytovat uživatelům. [16]

ID	Požadavek
F1	Systém umožní registraci a autorizaci uživatelů.
F2	Systém umožní správu aktivit pro sběr dat o pohybových vzorcích.
F3	Systém umožní správu uživatelů a jejich rolí, kteří mohou mít přístup k funkcím aplikace.
F4	Systém umožní nastavení vzorkovací frekvence pro různé typy aktivit.
F5	Systém umožní vizualizaci dat ze senzorů v reálném čase.
F6	Systém umožní zobrazení naměřených dat ve formě grafů.
F7	Systém umožní export naměřených dat ve formátu CSV.
F8	Systém umožní zobrazení a správu historie naměřených aktivit.

F9	Systém umožní změnu uživatelského hesla a smazání účtu.
F10	Systém umožní nastavení vzhledu aplikace (světlý/tmavý režim).
F11	Systém umožní spuštění, pozastavení a obnovení měření aktivity.
F12	Systém umožní administrátorům přidávání a úpravu typů aktivit.

Tabulka 1: Funkční požadavky (zdroj vlastní)

2.1.3 Nefunkční požadavky

V této části jsou definovány klíčové nefunkční požadavky na vyvíjenou mobilní aplikaci pro výzkum pohybových vzorců. Tyto požadavky stanovují kvalitativní charakteristiky a technické omezení systému. [16]

ID	Požadavek
NF1	Systém musí zajistit anonymizaci dat aktivit při odesílání na server.
NF2	Systém musí implementovat zabezpečení dat během přenosu i ukládání.
NF3	Mobilní aplikace musí být vyvinuta v programovacím jazyce Kotlin pro operační systém Android.
NF4	Systém musí využívat Firebase jako databázovou platformu pro ukládání dat.
NF5	Uživatelské rozhraní aplikace musí být responzivní a přizpůsobené různým velikostem displeje.
NF6	Systém musí implementovat bezpečné metody pro ochranu uživatelských účtů.
NF7	Aplikace musí podporovat operační systém Android verze 8.0 (API úroveň 26) a vyšší.
NF8	Back-end systém musí být vyvinut v Pythonu s využitím API pro zpracování a vizualizaci dat.
NF9	API musí být nasazeno na platformě Microsoft Azure pro zajištění škálovatelnosti a dostupnosti.
NF10	Systém musí minimalizovat spotřebu baterie při sběru dat ze senzorů.
NF11	Systém musí zajistit rychlou odezvu uživatelského rozhraní i během sběru dat.

Tabulka 2: Nefunkční požadavky (zdroj – vlastní)

2.2 Analýza

Tato kapitola se zaměřuje na analýzu a návrh mobilní aplikace pro výzkum pohybových vzorců. Obsahuje diagram případů užití pro vizualizaci interakcí mezi uživateli a systémem, datový model popisující strukturu ukládaných informací a architekturu celého řešení. Cílem analýzy je poskytnout jasnou představu o fungování systému před zahájením implementace.

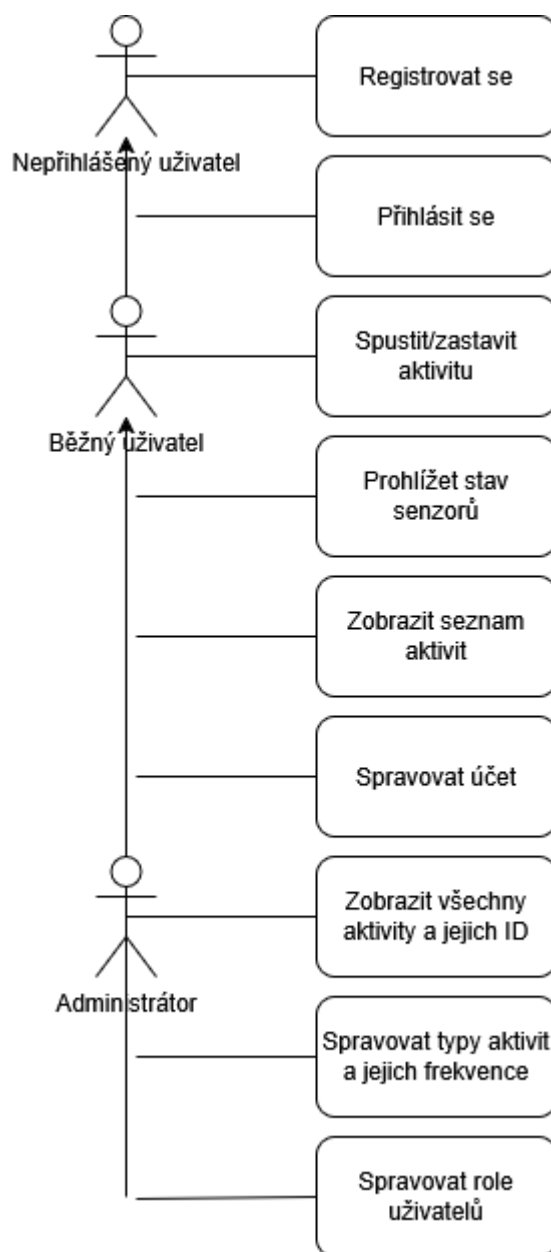
2.2.1 Diagram případů užití

Diagram případů užití (Use Case Diagram) znázorňuje interakce mezi uživateli a systémem. V naší aplikaci rozlišujeme tři typy uživatelů s různými oprávněními: nepřihlášený uživatel, běžný uživatel a administrátor. [16]

Nepřihlášený uživatel představuje osobu, která aplikaci právě nainstalovala nebo se odhlásila ze svého účtu. Tento uživatel má velmi omezený přístup k funkcím aplikace a může provádět pouze základní operace související s autentizací. Konkrétně má možnost pouze se zaregistrovat vytvořením nového účtu (zadáním e-mailu a hesla) nebo se přihlásit, pokud již má existující účet. Tato role je vstupním bodem do systému a je navržena s ohledem na bezpečnost, aby neautorizovaní uživatelé neměli přístup k citlivým funkcím a datům aplikace.

Běžný uživatel je již autentizovaný v systému a má přístup ke všem základním funkcím aplikace potřebným pro sběr a vizualizaci dat o pohybových vzorcích. Tento uživatel může spouštět a zastavovat měření různých typů aktivit, prohlížet hodnoty ze sensorů zařízení v reálném čase, zobrazovat seznam svých naměřených aktivit včetně možnosti stažení dat ve formátu CSV nebo zobrazení v grafech. Běžný uživatel může také spravovat svůj účet, což zahrnuje změnu hesla, nastavení vzhledu aplikace (světlý/tmavý režim) a možnost kompletního smazání svého účtu včetně všech spojených dat. Tato role je navržena pro koncové uživatele, kteří se účastní výzkumu pohybových vzorců.

Administrátor má rozšířená oprávnění oproti běžnému uživateli a může kromě všech základních funkcí aplikace také spravovat systém jako celek. Tento uživatel má přístup ke všem naměřeným aktivitám v systému včetně jejich jedinečných identifikátorů, což mu umožňuje kompletní přehled o shromážděných datech. V administrátorském panelu může spravovat typy aktivit – přidávat nové, upravovat existující nebo mazat nepotřebné, včetně nastavení jejich vzorkovací frekvence, která ovlivňuje přesnost měření a spotřebu baterie. Administrátor může také spravovat role ostatních uživatelů v systému – přidělovat nebo odebírat administrátorská práva. Tato role je určena pro výzkumníky a správce systému, kteří potřebují komplexní kontrolu nad aplikací a shromážděnými daty.



Obrázek 1: Diagram případu užití (zdroj – vlastní)

Diagram případů užití jasně ilustruje hierarchii uživatelských rolí a jejich oprávnění v systému. Administrátorská role dědí všechny případy užití běžného uživatele a rozšiřuje je o další funkce spojené se správou systému. Tento model slouží jako základ pro definování uživatelského rozhraní a implementaci přístupových práv v aplikaci.

2.2.2 Datový model

Tato část popisuje strukturu ukládaných dat v aplikaci, která je implementována pomocí Firebase Firestore jako NoSQL cloudové databáze. Datový model je navržen s důrazem na anonymizaci dat a ochranu soukromí uživatelů při výzkumu pohybových vzorců.

Klíčovým aspektem navrženého datového modelu je striktní oddělení uživatelských účtů a naměřených dat, což zajišťuje anonymizaci dat o pohybových vzorcích. Identifikátory měření (session ID) jsou ukládány pouze lokálně na zařízení uživatele a nejsou spojeny s uživatelským účtem v cloudové databázi. Tento přístup umožňuje uživatelům vidět pouze vlastní měření bez rizika identifikace v centrální databázi.

Kolekce „activities“

Tato kolekce obsahuje dokumenty reprezentující různé typy aktivit, které uživatelé mohou sledovat v aplikaci. Každý dokument v této kolekci obsahuje následující atributy:

- **name:** string – název typu aktivity (např. „Běh“, „Procházka“, „Cvičení“)
- **samplingRate:** number – vzorkovací frekvence pro daný typ aktivity v milisekundách

Kolekce „sessions“

Tato kolekce obsahuje dokumenty reprezentující jednotlivá měření aktivit provedená uživateli, ovšem bez přímého propojení na konkrétní uživatelské účty. Každý dokument v této kolekci obsahuje následující atributy:

- **activityId:** string – odkaz na dokument v kolekci „activities“ specifikující typ aktivity
- **duration:** number – celková doba trvání měření v milisekundách
- **startTime:** timestamp – časová značka začátku měření
- **endTime:** timestamp – časová značka konce měření
- **trajectoryBase64:** string – zakódovaná trajektorie pohybu ve formátu Base64

Každý dokument v kolekci „sessions“ také obsahuje vnořenou kolekci „data“, která uchovává jednotlivé datové body zaznamenané během měření. Každý dokument v této vnořené kolekci obsahuje:

- **accelerometer:** map – hodnoty z akcelerometru
 - **x:** number – hodnota zrychlení v ose X
 - **y:** number – hodnota zrychlení v ose Y
 - **z:** number – hodnota zrychlení v ose Z

- **gyroscope:** map – hodnoty z gyroskopu
 - **x:** number – hodnota úhlové rychlosti v ose X
 - **y:** number – hodnota úhlové rychlosti v ose Y
 - **z:** number – hodnota úhlové rychlosti v ose Z
- **rotationVector:** map – hodnoty z rotačního vektoru
 - **w:** number – hodnota W rotačního vektoru
 - **x:** number – hodnota X rotačního vektoru
 - **y:** number – hodnota Y rotačního vektoru
 - **z:** number – hodnota Z rotačního vektoru
- **timestamp:** timestamp – časová značka, kdy byl datový bod zaznamenán

Kolekce „users“

Tato kolekce obsahuje dokumenty reprezentující uživatele aplikace a jejich nastavení, ale bez vazby na naměřená data. Každý dokument v této kolekci obsahuje následující atributy:

- **email:** string – e-mailová adresa uživatele
- **isAdmin:** bool – příznak indikující, zda má uživatel administrátorská práva

Firestore Authentication

Aplikace využívá Firestore Authentication pro správu uživatelských účtů a přístupových práv. Tento systém spravuje následující informace o uživateli:

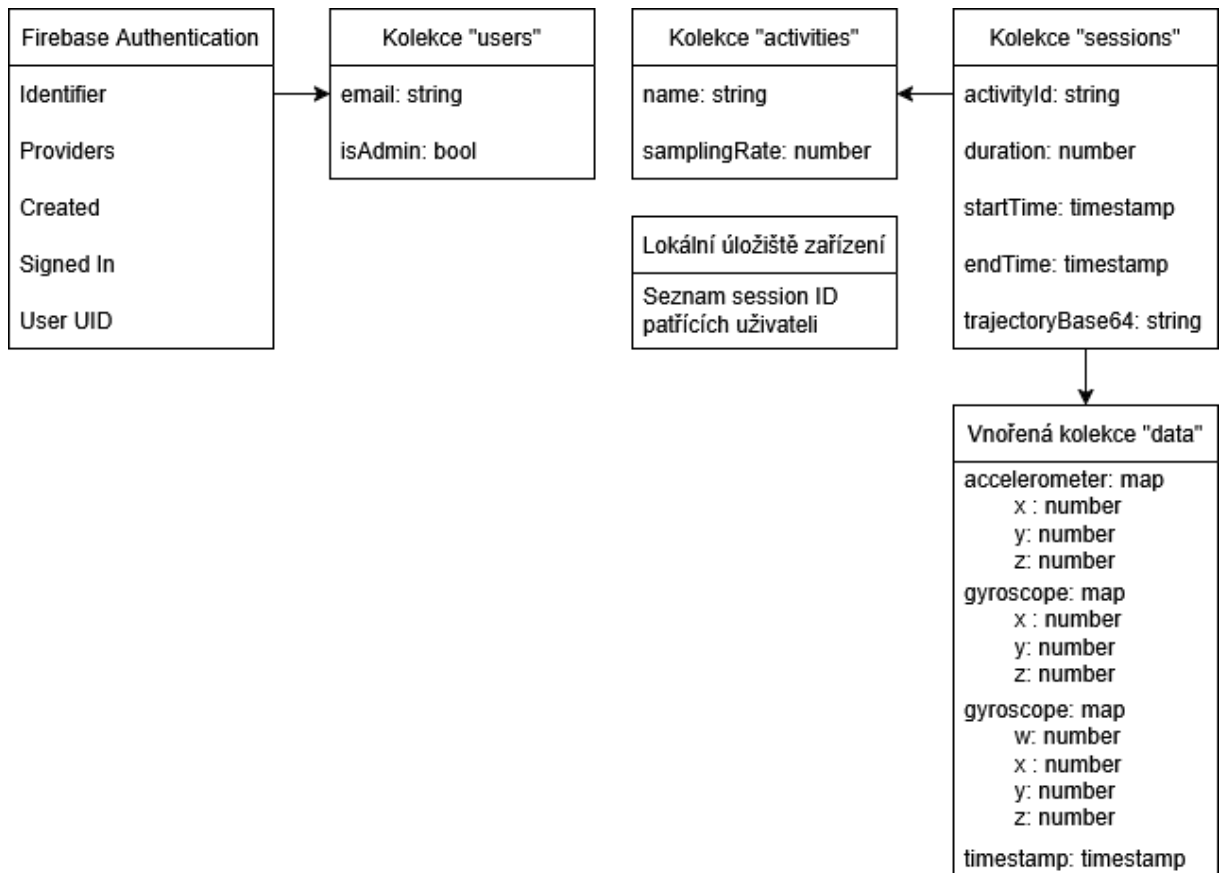
- **Identifier:** e-mailová adresa uživatele sloužící jako identifikátor
- **Providers:** metoda autentizace (v našem případě e-mail/heslo)
- **Created:** datum vytvoření účtu
- **Signed In:** datum posledního přihlášení
- **User UID:** unikátní identifikátor uživatele, který slouží k propojení záznamů v autentizačním systému s dokumenty v kolekci „users“

Lokální ukládání session ID

Pro zajištění anonymizace dat jsou ID měření (session ID) ukládána pouze lokálně v zařízení uživatele. To umožňuje:

- Uživatel vidí pouze svá vlastní měření
- Neexistuje žádná vazba mezi uživatelským účtem a naměřenými daty v cloudové databázi
- Administrátoři vidí všechna anonymní data bez možnosti identifikace konkrétních uživatelů

Tento přístup zajišťuje ochranu soukromí uživatelů při zachování možnosti shromažďovat cenná data pro výzkum pohybových vzorců.



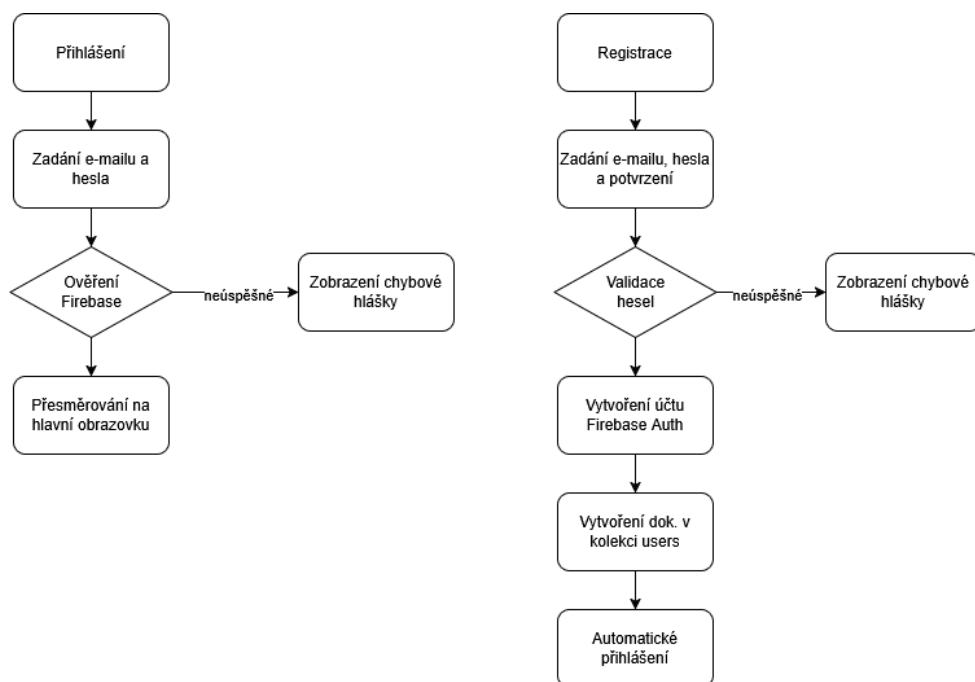
Obrázek 2: Datový model (zdroj – vlastní)

Tento datový model umožňuje efektivní ukládání a přístup k datům o pohybových vzorcích při zachování anonymity uživatelů. Absence propojení mezi uživatelskými účty a naměřenými daty v cloudové databázi zajišťuje, že ani administrátoři nemohou identifikovat, který uživatel provedl konkrétní měření, což je klíčové pro etický sběr dat v rámci výzkumu.

2.2.3 Popis procesů (BPM)

Business Process Management (BPM) představuje metodický přístup k identifikaci, návrhu, dokumentaci a monitorování klíčových procesů v rámci systému. V kontextu vyvíjené mobilní aplikace pro výzkum pohybových vzorců je analýza BPM zásadní pro pochopení logických posloupností akcí, které uživatelé a administrátoři provádějí při interakci se systémem. Tato kapitola popisuje hlavní business procesy aplikace se zaměřením na jejich tok, účastníky a výstupy. [17]

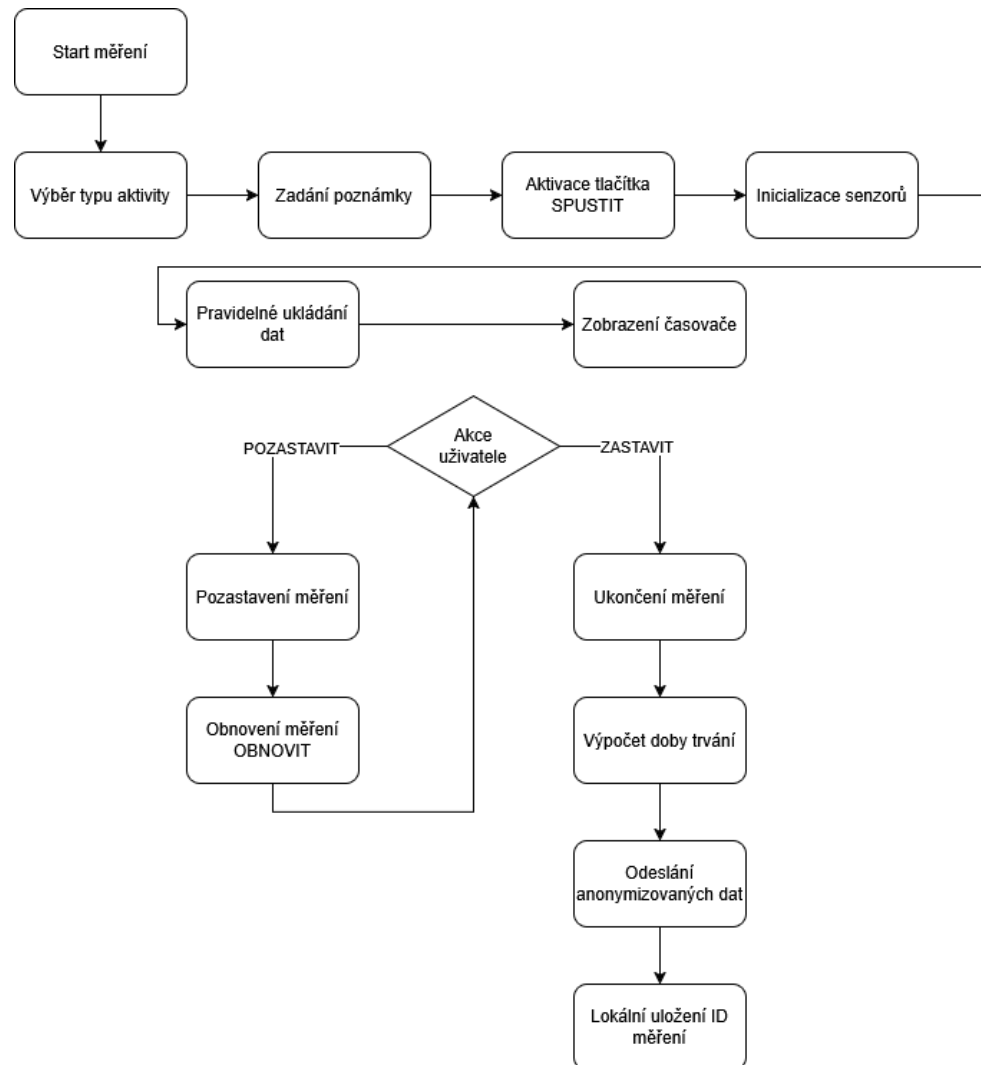
Proces autentizace uživatele zahrnuje dvě klíčové operace – přihlášení a registraci. Při přihlášení uživatel zadává své přihlašovací údaje (e-mail a heslo), které jsou ověřeny proti Firebase Authentication. Úspěšné ověření vede k přesměrování na hlavní obrazovku, zatímco neúspěšný pokus zobrazí chybovou hlášku. Proces registrace začíná zadáním e-mailu, hesla a jeho potvrzení. Systém ověřuje shodu hesel a jejich bezpečnost, následně vytváří nový účet ve Firebase Authentication a přidružený dokument v kolekci „users“ s výchozí rolí běžného uživatele. Po úspěšné registraci je uživatel automaticky přihlášen.



Obrázek 3: Proces autentizace (zdroj – vlastní)

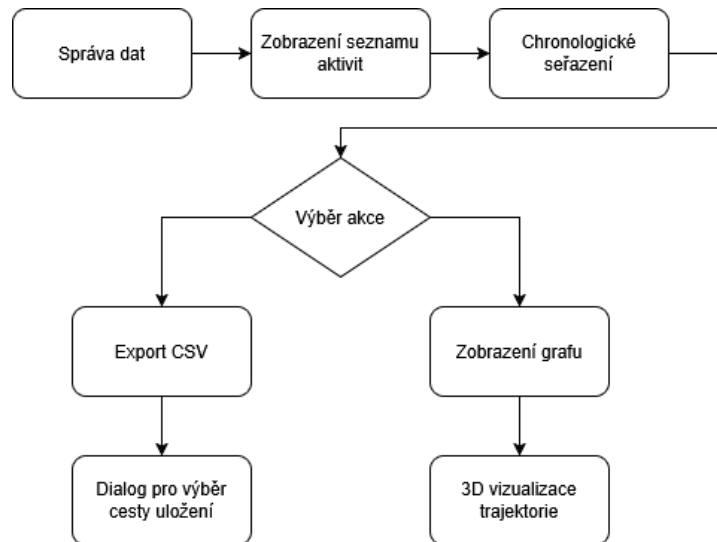
Sběr dat o pohybových vzorcích představuje jádro funkcionality aplikace. Proces zahájení měření aktivity začíná na hlavní obrazovce, kde uživatel vybírá typ aktivity a volitelně zadává poznámku. Po vyplnění povinných údajů se aktivuje tlačítko „SPUSTIT“, jehož stisknutím se inicializují senzory s příslušnou vzorkovací frekvencí. Aplikace začne pravidelně ukládat data a zobrazovat časovač. Ukončení měření nastává stisknutím tlačítka „ZASTAVIT“, čímž se

přerušit sběr dat, zaznamenat čas ukončení a vypočítá celková doba trvání. Anonymizovaná data jsou odeslána na server a uložena v kolekci „sessions“, přičemž ID měření se ukládá lokálně v zařízení. Během měření může uživatel aktivitu dočasně pozastavit tlačítkem „ZASTAVIT“ a následně obnovit tlačítkem „OBNOVIT“, což umožňuje přerušit sběr dat bez ztráty již naměřených hodnot.



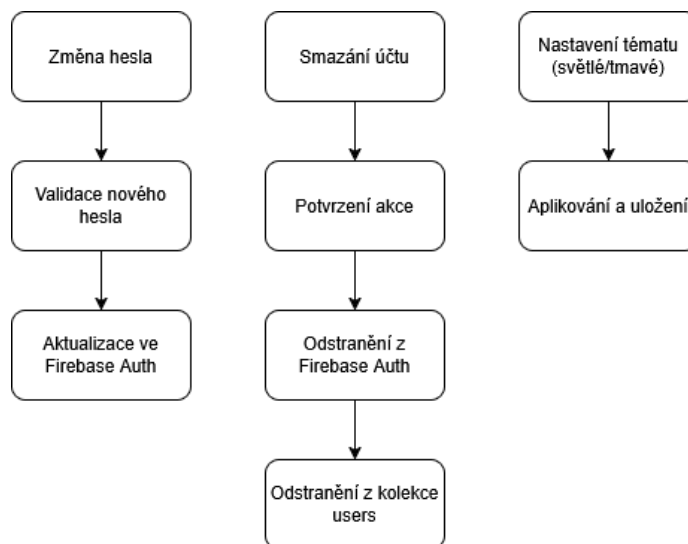
Obrázek 4: Sběr dat o pohybových vzorcích (zdroj – vlastní)

Správa naměřených dat umožňuje uživatelům pracovat s výsledky měření. V seznamu aktivit systém zobrazuje přehled všech měření uživatele na základě lokálně uložených ID, seřazených chronologicky. Pro každé měření může uživatel exportovat data ve formátu CSV stisknutím tlačítka „STÁHNOUT CSV“, což otevře dialog pro výběr cesty k uložení. Stisknutím tlačítka „ZOBRAZIT GRAF“ se naměřená data vizualizují v dialogovém okně, včetně možnosti zobrazení trajektorie pohybu v trojrozměrném prostoru.



Obrázek 5: Správa naměřených dat (zdroj – vlastní)

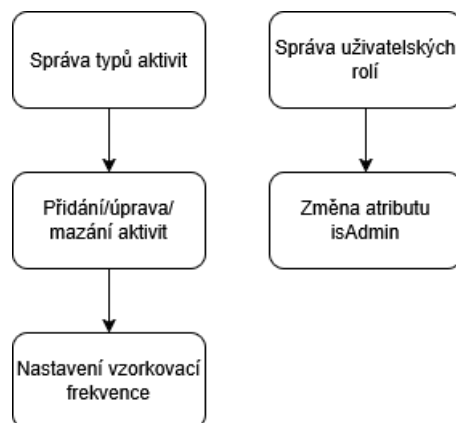
Správa uživatelského účtu poskytuje uživatelům kontrolu nad jejich profilem. Proces změny hesla probíhá v sekci správy účtu, kde uživatel zadává nové heslo a jeho potvrzení. Po úspěšné validaci je heslo aktualizováno ve Firebase Authentication. Smazání účtu představuje nevratnou akci, která po potvrzení odstraní účet z Firebase Authentication a dokument uživatele z kolekce „users“, přičemž anonymizovaná data zůstávají v databázi. V nastavení vzhledu aplikace může uživatel vybrat preferované téma (podle systému, světlé, tmavé), které je okamžitě aplikováno a uloženo pro budoucí použití.



Obrázek 6: Správa uživatelského účtu (zdroj – vlastní)

Administrátorské procesy rozšiřují možnosti správy systému. Správa typů aktivit umožňuje administrátorům přidávat, upravovat a mazat typy aktivit v kolekci „activities“, včetně nastavení jejich vzorkovací frekvence. Správa uživatelských rolí umožňuje přidělovat a

odebírat administrátorská práva jiným uživatelům jednoduchou změnou atributu „isAdmin“ v příslušném dokumentu uživatele.



Obrázek 7: Administrátorské procesy (zdroj – vlastní)

Analýza business procesů poskytuje ucelený pohled na fungování aplikace z hlediska interakcí uživatelů se systémem. Tato dokumentace pomáhá optimalizovat uživatelskou zkušenost a zajistit, že aplikace efektivně plní svůj účel jako nástroj pro výzkum pohybových vzorců.

2.3 Návrh

Tato kapitola se zabývá návrhem aplikace z hlediska její architektury a použitých technologií. Popisuje, jak jednotlivé komponenty systému vzájemně komunikují a jaké konkrétní technologie byly zvoleny pro implementaci.

2.3.1 Architektura aplikace

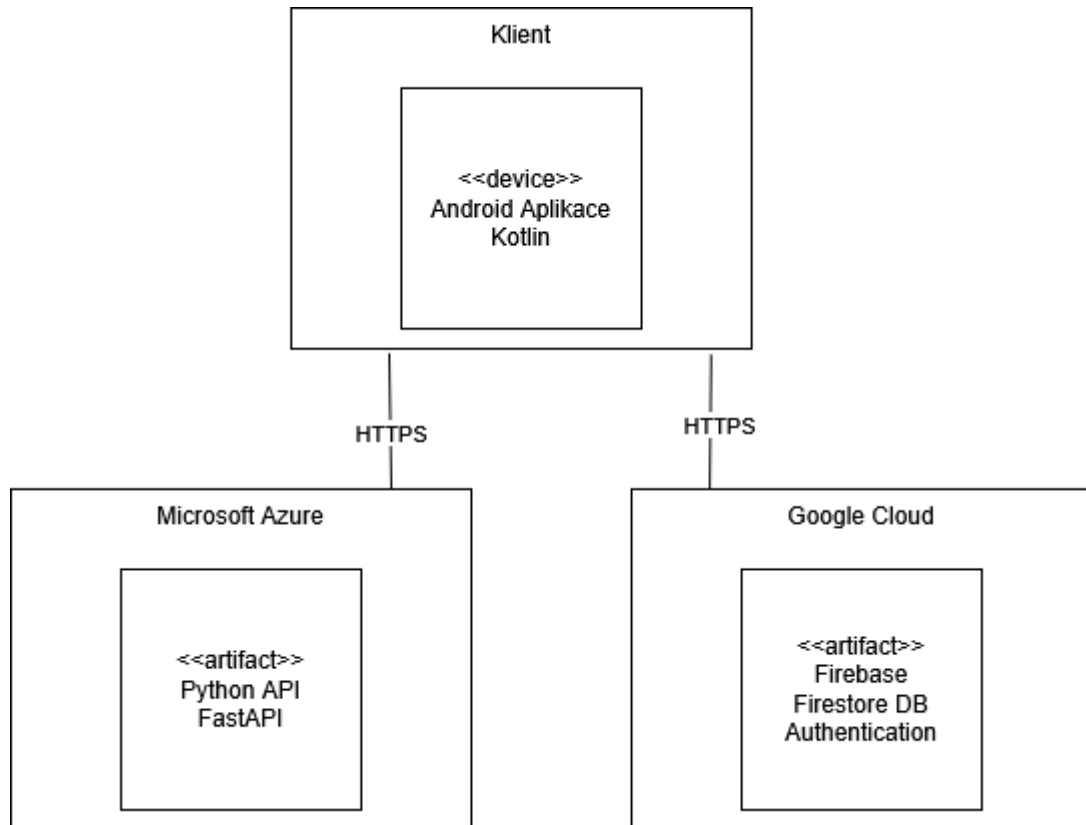
Tato sekce popisuje architekturu navrženého systému, který se skládá z mobilní aplikace pro Android, backendového API vyvinutého v Pythonu a cloudové databáze Firebase. Architektura je navržena s důrazem na škálovatelnost, bezpečnost a efektivní zpracování dat.

Architektura aplikačního systému se skládá ze tří hlavních komponent:

Android aplikace – Mobilní aplikace vyvinutá v Kotlinu pomocí Android Studia představuje klientskou část systému. Aplikace poskytuje uživatelské rozhraní a komunikuje s oběma cloudovými službami prostřednictvím HTTPS protokolu.

Firestore na Google Cloud – Cloudová služba poskytující databázové úložiště (Firestore) a autentizační služby (Authentication). Firestore zajišťuje real-time synchronizaci dat, správu uživatelských účtů a další cloudové služby. Komunikace probíhá výhradně s Android aplikací prostřednictvím Firebase API.

Python API na Microsoft Azure – Specializovaná backend služba nasazená na cloudové platformě Microsoft Azure. API zpracovává pokročilou analytickou logiku pro vizualizaci dat, kterou by nebylo efektivní implementovat přímo v mobilní aplikaci. Poskytuje služby pro generování 3D grafů trajektorií pohybu.



Obrázek 8: Architektura aplikace (zdroj – vlastní)

Komunikace mezi komponentami probíhá následovně:

- **Android aplikace ↔ Firebase:** Obousměrná komunikace prostřednictvím Firebase API pro ukládání anonymizovaných dat ze senzorů a jejich následné načítání pro zobrazení v uživatelském rozhraní
- **Android aplikace → Python API:** Jednosměrná komunikace pomocí HTTPS requestů pro odesílání CSV dat k vizualizaci
- **Python API → Android aplikace:** Jednosměrná komunikace pomocí HTTPS response pro vrácení vygenerovaných grafických vizualizací

Tato architektura zajišťuje jasné oddělení zodpovědností mezi jednotlivými komponentami. Firebase slouží jako primární úložiště dat s možností real-time synchronizace, zatímco Python API na Azure poskytuje specializované analytické funkce. Android aplikace funguje jako

centrální orchestrátor komunikace mezi oběma cloudovými službami, přičemž žádná přímá komunikace mezi Firebase a Azure neexistuje. Tento přístup umožňuje nezávislou škálovatelnost a údržbu jednotlivých služeb.

2.3.2 Použité technologie

Tato sekce popisuje klíčové technologie použité při vývoji aplikace. Každá z těchto technologií byla zvolena s ohledem na její specifické přednosti a vhodnost pro daný účel v rámci celkové architektury systému.

2.3.2.1 Backend

Kotlin

Kotlin je moderní, staticky typovaný programovací jazyk vyvinutý společností JetBrains, který běží na Java Virtual Machine (JVM). Od roku 2017 je oficiálně podporován Googlem jako preferovaný jazyk pro vývoj Android aplikací. Kotlin přináší výrazná vylepšení oproti Javě včetně nulové bezpečnosti (null-safety), lambda funkcí, rozšíření funkcí (extension functions), datových tříd a korutin pro asynchronní programování. Jeho interoperabilita s Javou umožňuje postupný přechod existujících projektů a využití všech existujících Java knihoven. [18]

Python

Python je všestranný, interpretovaný vysokoúrovňový programovací jazyk s dynamickým typováním. Je známý svou čitelnou syntaxí a filozofií „čitelnost především“. Python podporuje různá programovací paradigmaty včetně objektově orientovaného, imperativního, funkcionálního a procedurálního. Díky své flexibilitě, rozsáhlému ekosystému knihoven jako NumPy, Pandas, TensorFlow nebo Django, a rostoucí popularitě ve strojovém učení a datové analýze, je Python oblíbenou volbou pro vývoj komplexních backendových služeb, API a vědeckých aplikací. [19]

2.3.2.2 Frontend

XML

XML (eXtensible Markup Language) je flexibilní značkovací jazyk, který v Android vývojovém prostředí slouží primárně k definici uživatelského rozhraní. V rámci Android struktury aplikace se XML používá pro tvorbu layoutů (rozvržení obrazovek), definici stylů, konfiguraci zdrojů lokalizovaných podle jazykových mutací, specifikaci animací a deklaraci manifestu aplikace. XML umožňuje jasné oddělení prezentační vrstvy od logiky aplikace, což přispívá k lepší udržitelnosti a přehlednosti kódu. [20]

Material Theme

Material Design je designový jazyk vyvinutý společností Google, který sjednocuje uživatelskou zkušenost napříč různými platformami a velikostmi zařízení. Je založen na taktilní realitě inspirované fyzickými materiály a objekty, s důrazem na pohyb, hloubku a responzivní interakce. Material Theme zahrnuje komplexní sadu komponent, barevných palet, typografických pravidel a animačních principů. V Android aplikacích poskytuje uživatelům intuitivní a vizuálně konzistentní rozhraní, které respektuje zavedené vzory interakce, což vede k rychlejšímu osvojení aplikace uživatelem. [21]

2.3.2.3 Vývojová prostředí

Android Studio

Android Studio je oficiální integrované vývojové prostředí (IDE) pro platformu Android, postavené na IntelliJ IDEA od společnosti JetBrains. Nabízí komplexní sadu nástrojů pro vývoj mobilních aplikací, včetně vizuálního editoru layoutů, analyzátoru APK, inspektoru kódu a Gradle build systému. Mezi jeho pokročilé funkce patří emulátor Android zařízení s AVD Manager, nástroje pro profilování výkonu a spotřeby baterie, Firebase a Cloud Platform integrace nebo systém pro automatické refaktorování kódu. Android Studio také obsahuje nástroje pro testování a ladění, jako jsou Espresso Test Recorder nebo Layout Inspector. [22]

PyCharm

PyCharm je profesionální integrované vývojové prostředí pro Python, vyvinuté společností JetBrains. Nabízí pokročilou inteligentní asistenci kódu, včetně dokončování kódu, analýzy, zvýrazňování chyb a rychlých oprav. PyCharm poskytuje výkonný debugger, integrovaný testovací framework, nástroje pro profilování, podporu vzdálených interpretů a správu virtuálních prostředí. Profesionální verze obsahuje podporu webových technologií (HTML, CSS, JavaScript) a frameworků (Django, Flask, FastAPI, Pyramid), nástroje pro databáze, a integraci s Dockerem. PyCharm umožňuje produktivní vývoj Pythonu na různých platformách s intuitivním rozhraním a možnostmi přizpůsobení. [23]

Microsoft Azure

Microsoft Azure je cloudová platforma poskytující širokou škálu služeb pro vývoj, hostování a správu aplikací. Nabízí infrastrukturu jako službu (IaaS), platformu jako službu (PaaS) a software jako službu (SaaS). Azure poskytuje výpočetní služby jako virtuální stroje a kontejnery, technologie pro vývoj webových a mobilních aplikací, databázové služby (SQL,

NoSQL), analytické nástroje, umělou inteligenci a služby strojového učení. Pro Python aplikace Azure nabízí App Service pro hostování webových aplikací a API, Azure Functions pro serverless architekturu, Azure Kubernetes Service pro orchestraci kontejnerů, a mnoho dalších služeb. Azure se vyznačuje vysokou škálovatelností, dostupností a zabezpečením, s globální sítí datacenter umožňující dostupnost aplikací po celém světě. [24]

2.4 Implementace

Tato kapitola popisuje konkrétní implementaci jednotlivých částí systému. Zaměřuje se na klíčové komponenty aplikace a jejich funkční realizaci v kódu, včetně databázové struktury, serverové logiky, zpracování externích dat a uživatelského rozhraní.

2.4.1 Databáze

V aplikaci je implementována Firebase Firestore, cloudová NoSQL databáze, která tvoří páteř datové vrstvy celého systému. Tato technologie byla zvolena zejména pro její schopnost synchronizace dat v reálném čase, snadnou škálovatelnost a integraci s ostatními Firebase službami.

2.4.1.1 Konfigurace Firebase

Integrace Firebase služeb do Android aplikace je realizována prostřednictvím konfiguračního souboru `google-services.json`, který obsahuje všechny potřebné parametry pro připojení aplikace k projektu Firebase. Tento soubor je automaticky generován při registraci aplikace ve Firebase konzoli a obsahuje klíčové informace jako je ID projektu, API klíče a další konfigurační parametry.

V `build.gradle` souboru aplikace jsou dále definovány potřebné závislosti na Firebase SDK, které umožňují využívat různé Firebase služby. Tyto závislosti zahrnují knihovny pro Firestore, Authentication a Analytics. Pro správnou funkčnost je také přidán plugin `google-services`, který zpracovává konfigurační soubor a zpřístupňuje jeho obsah v rámci build procesu.

Tato konfigurační vrstva vytváří most mezi Android aplikací a Firebase službami, což umožňuje bezproblémovou komunikaci mezi klientskou částí aplikace a cloudovými službami bez nutnosti explicitního nastavování připojení, adres serverů nebo autentizačních parametrů v kódu aplikace.

2.4.1.2 Struktura a přístup k datům

Databázová struktura aplikace je organizována do několika klíčových kolekcí – „sessions“, „activities“ a „users“. Každá z těchto kolekcí slouží specifickému účelu v rámci aplikační

logiky. Kolekce „sessions“ obsahuje záznamy o jednotlivých relacích sledování pohybu uživatelů, včetně časových značek, trvání a odkazů na typ aktivity. V kolekci „activities“ jsou definovány různé typy sledovaných aktivit s jejich parametry, jako je vzorkovací frekvence. Kolekce „users“ uchovává informace o uživateli systému, včetně jejich rolí a oprávnění.

Pro optimalizaci výkonu při dotazování dat je implementován kompozitní index pro kolekci „sessions“, který kombinuje filtrování podle uživatelského ID a řazení podle času zahájení v sestupném pořadí. Tento index výrazně zrychluje zobrazení seznamu relací konkrétního uživatele, což je jedna z nejčastějších operací v aplikaci.

2.4.1.3 Firebase API a datový přístup

Aplikace využívá oficiální Firebase Android SDK, které poskytuje vysokoúrovňové API pro interakci s Firebase službami. Toto API abstrahuje nízkoúrovňové detaily komunikace a poskytuje vývojářům intuitivní rozhraní pro práci s cloudovými daty.

Přístup k datům je v celé aplikaci řešen asynchronně prostřednictvím Firebase API. To zajišťuje, že hlavní vlákno aplikace není blokováno čekáním na dokončení databázových operací, což přispívá k plynulé uživatelské zkušenosti. Asynchronní operace jsou realizovány pomocí callback funkcí nebo Kotlin korutin, které zjednodušují práci s asynchronními voláními.

Pro dále zlepšení výkonu je implementován mechanismus cachování dat, který minimalizuje počet dotazů na databázi. Tento caching mechanismus je využíván především při zobrazování seznamu relací, kde se lokálně ukládají informace o aktivitách. Tím se eliminuje potřeba opakovaného načítání stejných dat z databáze, což je zvláště přínosné při procházení delšího seznamu relací, kde se mohou stejné typy aktivit vyskytovat opakovaně.

Firebase SDK také poskytuje vestavěnou offline perzistenci dat, která umožňuje aplikaci fungovat i bez připojení k internetu. Data, která byla dříve načtena, jsou uložena lokálně a jsou dostupná i v offline režimu. Po obnovení připojení pak Firebase automaticky synchronizuje změny provedené offline s cloudovou databází.

2.4.1.4 Základní databázové operace

V aplikaci jsou implementovány všechny základní CRUD operace (Create, Read, Update, Delete) s využitím Firebase Firestore API. Vytváření nových záznamů probíhá pomocí metod `add()` pro automatické generování ID nebo `set()` pro explicitní nastavení ID dokumentu. Tento přístup je využíván například při vytváření nových relací nebo registraci nových uživatelů.

Čtení dat z databáze je realizováno pomocí asynchronních dotazů, které umožňují filtrování a řazení výsledků. Aplikace využívá různé přístupy k načítání dat – od jednorázových dotazů až po posluchače změn v reálném čase, které okamžitě aktualizují uživatelské rozhraní při změnách v databázi.

Pro aktualizaci existujících dokumentů je používána metoda `update()`, která umožňuje částečnou modifikaci dokumentu bez nutnosti přepisovat jeho celý obsah. Tento přístup je efektivní z hlediska množství přenášených dat a minimalizuje riziko konfliktu při současných úpravách. Typickým příkladem je ukončení relace, kdy se do existujícího dokumentu pouze doplní časová značka ukončení a celkové trvání.

Mazání dokumentů je implementováno především v administrativní části aplikace, kde je potřeba odstraňovat neplatné nebo zastaralé záznamy. Implementace těchto operací je doplněna o odpovídající aktualizace uživatelského rozhraní, aby zobrazené informace vždy odpovídaly aktuálnímu stavu databáze.

Autentizace a správa uživatelů

Pro autentizaci uživatelů je využíván Firebase Authentication, který poskytuje bezpečný způsob přihlašování pomocí emailu a hesla. Registrační proces je navržen jako dvoustupňový – nejprve se vytvoří autentizační záznam a po jeho úspěšném vytvoření se v databázi Firestore vytvoří uživatelský profil s dodatečnými informacemi a oprávněními.

Aplikace implementuje komplexní zpracování různých typů autentizačních chyb, které mohou nastat při registraci nebo přihlašování uživatelů. Toto zpracování poskytuje uživatelsky přívětivé chybové zprávy, které pomáhají rychle identifikovat a řešit problémy, jako jsou nesprávné přihlašovací údaje, již existující email nebo příliš slabé heslo.

Po úspěšném přihlášení je uživatelská session udržována, dokud se uživatel explicitně neodhlásí. Aplikace při spuštění automaticky kontroluje, zda existuje aktivní přihlášení, a pokud ano, přesměruje uživatele přímo na hlavní obrazovku bez nutnosti opětovného zadávání přihlašovacích údajů.

Správa přístupových práv

V aplikaci je implementován systém správy přístupových práv založený na rolích uživatelů. Každý uživatel má v databázi přiřazenou roli (běžný uživatel nebo administrátor), která determinuje jeho oprávnění v systému. Administrátoři mají přístup k rozšířeným funkcím, jako

je správa aktivit, úprava uživatelských práv a přístup k detailním informacím o všech relacích v systému.

Uživatelské rozhraní se dynamicky přizpůsobuje podle role přihlášeného uživatele. Administrátorům jsou zobrazeny dodatečné informace a ovládací prvky, zatímco běžným uživatelům je prezentováno zjednodušené rozhraní zaměřené pouze na funkce, ke kterým mají oprávnění. Tento přístup zajišťuje přehlednost a jednoduchost používání aplikace pro běžné uživatele při zachování plné funkčnosti pro správce systému.

Implementace Firebase Firestore a Authentication v této aplikaci poskytuje robustní základ pro ukládání, synchronizaci a zabezpečení dat. Asynchronní přístup k datům a optimalizace pomocí cachování a indexů zajišťují plynulou uživatelskou zkušenost i při větším množství dat. Integrace s Firebase pomocí konfiguračního souboru a SDK zjednodušuje implementaci cloudové databáze a autentizace bez nutnosti správy vlastní serverové infrastruktury. Systém správy přístupových práv pak umožňuje flexibilní řízení funkcí aplikace podle potřeb a oprávnění jednotlivých uživatelů.

2.4.2 Serverová část

Serverová část aplikace se skládá ze dvou hlavních komponent – Firebase Firestore pro ukládání a správu dat a Python API nasazeného v Azure pro pokročilé zpracování a vizualizaci dat. Tato hybridní architektura kombinuje výhody obou přístupů a poskytuje komplexní řešení pro zpracování dat ze senzorů mobilního zařízení.

2.4.2.1 Zpracování dat ze senzorů

Aplikace sbírá data z několika senzorů mobilního zařízení – akcelerometru, gyroskopu a senzoru rotačního vektoru. Tato data jsou klíčová pro analýzu pohybu uživatele a jsou zpracovávána v reálném čase v rámci služby běžící na pozadí.

Před odesláním dat do databáze probíhá několik důležitých kroků předzpracování:

- **Kompenzace gravitace** – Surová data z akcelerometru obsahují gravitační složku, která může zkreslit analýzu pohybu. Pro její eliminaci je implementován algoritmus využívající rotační vektor k výpočtu orientace zařízení vzhledem ke gravitačnímu poli Země. Gravitační složka je následně odečtena od naměřených hodnot akcelerometru, čímž se získají hodnoty reprezentující skutečné zrychlení způsobené pohybem.
- **Filtrace šumu** – Sensorová data obvykle obsahují vysokofrekvenční šum, který může negativně ovlivnit přesnost analýzy. Pro jeho redukci je implementován Exponential

Moving Average (EMA) filtr s konfigurovaným koeficientem alfa. Tento filtr účinně vyhlazuje data při zachování významných charakteristik pohybu.

- **Normalizace a konverze typů** – Před uložením do databáze jsou hodnoty ze senzorů převedeny do standardního formátu. Hodnoty jsou strukturovány jako mapy s klíči x, y, z (a případně w pro rotační vektor) a převedeny na datový typ Double, který je vhodnější pro následné zpracování a analýzu.
- **Časové značky** – Každý datový záznam je opatřen přesnou časovou značkou z Firebase Timestamp, která umožňuje přesnou rekonstrukci časové posloupnosti pohybu při následné analýze.

Služba běžící na pozadí zajišťuje kontinuální sběr a zpracování dat s nastaveným intervalem ukládání, který je optimalizován pro zachycení relevantních charakteristik pohybu při minimalizaci objemu přenášených dat a zatížení systému.

2.4.2.2 Anonymizace a ochrana osobních údajů

V souladu s principy ochrany osobních údajů a minimalizace dat aplikace implementuje mechanismus anonymizace uživatelských dat. Klíčovým prvkem tohoto přístupu je oddělení uživatelské identity od měřených dat:

- **Lokální ukládání identifikátorů relací** – Při vytvoření nové relace je vygenerován unikátní identifikátor, který je následně uložen pouze lokálně na zařízení uživatele pomocí LocalStoreService. V databázi Firebase není udržována přímá vazba mezi uživatelem a jeho relacemi měření.
- **Absence uživatelského ID v datech senzorů** – Data ukládaná v rámci relace neobsahují informace o uživateli, který měření prováděl. V databázi jsou tak uchovávána pouze anonymizovaná data o pohybu bez možnosti přímé identifikace konkrétního uživatele.
- **Oddělení autentizačních a měřících dat** – I když aplikace využívá Firebase Authentication pro přihlašování uživatelů, autentizační informace nejsou propojeny s naměřenými daty. Tento přístup zajišťuje, že i v případě získání přístupu k databázi sensorových dat by nebylo možné identifikovat konkrétní uživatele.

Tento přístup k anonymizaci dat je zvláště důležitý v kontextu aplikace sledující pohyb, která může potenciálně shromažďovat citlivé biometrické údaje. Implementace zajišťuje, že i když

jsou data využívána pro výzkumné nebo analytické účely, soukromí uživatelů zůstává chráněno.

2.4.2.3 Ukládání a správa dat

Data ze senzorů jsou ukládána do hierarchické struktury ve Firebase Firestore, která umožňuje efektivní organizaci a dotazování:

- **Struktura databáze** – Každá relace měření je reprezentována dokumentem v kolekci „sessions“. V rámci každé relace je vytvořena subkolekce „data“, která obsahuje jednotlivé záznamy ze senzorů. Tento hierarchický přístup umožňuje efektivní organizaci velkého množství datových bodů.
- **Optimalizace ukládání** – Pro optimalizaci výkonu a minimalizaci nákladů na datové přenosy je implementován mechanismus dávkového ukládání dat v pravidelných intervalech. Tímto způsobem je zajištěna rovnováha mezi granularitou zaznamenávaných dat a efektivitou ukládání.
- **Zpracování trajektorií** – Po dokončení relace mohou být shromážděná data zpracována do formy trajektorie pohybu, která je následně uložena v kompaktním formátu Base64. Tato funkce umožňuje efektivní vizualizaci a analýzu pohybu bez nutnosti opětovného zpracování všech původních datových bodů.

2.4.2.4 Python API v Azure

Pro pokročilé zpracování a vizualizaci pohybových dat je v aplikaci implementováno REST API postavené na frameworku FastAPI a nasazené v Microsoft Azure. Toto API poskytuje specializované funkce, které by bylo obtížné nebo neefektivní implementovat přímo v mobilní aplikaci.

Architektura API

API je vybudováno jako bezstavová služba, která přijímá data ve formátu CSV, provádí jejich zpracování a vrací výsledky ve formě vizualizací nebo analyzovaných dat. Klíčové prvky architektury API:

- **FastAPI framework** – Moderní, vysoce výkonný webový framework pro Python, který zajišťuje efektivní zpracování HTTP požadavků a automatické generování API dokumentace.

- **Asynchronní zpracování** – API využívá asynchronní přístup k obsluze požadavků, což umožňuje efektivní využití systémových zdrojů a zvládnutí většího počtu souběžných požadavků.
- **Cloudové nasazení** – API je nasazeno v Microsoft Azure, což poskytuje vysokou dostupnost, škálovatelnost a spolehlivost bez nutnosti správy vlastní serverové infrastruktury.

Vizualizace trajektorií pohybu

Jednou z hlavních funkcí Python API je vizualizace trajektorií pohybu na základě dat ze senzorů. Tato funkce zahrnuje:

- **Rekonstrukce trajektorie** – API implementuje algoritmy pro rekonstrukci 3D trajektorie pohybu zařízení z dat rotačního vektoru. Tato rekonstrukce umožňuje vizuální analýzu provedeného pohybu.
- **Vizualizace orientace** – Kromě samotné trajektorie API zobrazuje také orientaci zařízení v každém bodě trajektorie pomocí trojice vektorů reprezentujících lokální souřadný systém zařízení.
- **Generování grafů** – Pro vizualizaci jsou využívány knihovny Matplotlib a NumPy, které umožňují vytvářet interaktivní 3D grafy s možností rotace a přiblížení pro detailní analýzu.
- **Streamovaná odpověď** – Vytvořené vizualizace jsou pomocí StreamingResponse odesílány přímo klientovi bez nutnosti ukládání na disk, což minimalizuje latenci a požadavky na úložiště.

Zpracování a transformace dat

API poskytuje také funkce pro zpracování a transformaci surových dat ze senzorů:

- **Validace vstupních dat** – Před zpracováním jsou vstupní data validována, aby se zajistilo, že obsahují všechny potřebné sloupce ve správném formátu.
- **Transformace rotačních dat** – Implementace využívá knihovnu SciPy pro práci s kvaterniony a rotačními maticemi, což umožňuje přesné výpočty orientace zařízení v 3D prostoru.

- **Integrace pohybu** – API implementuje numerickou integraci pro převod rotačních dat na trajektorii pohybu, přičemž je možné konfigurovat parametry jako krok integrace pro dosažení optimálního poměru mezi přesností a výpočetní náročností.

Integrace s mobilní aplikací

Propojení mobilní aplikace s Python API je realizováno prostřednictvím HTTP požadavků:

- **Export a přenos dat** – Mobilní aplikace exportuje naměřená data do formátu CSV a odesílá je na API endpoint pomocí HTTP POST požadavku.
- **Zpracování a vizualizace** – API zpracuje přijatá data a vygeneruje odpovídající vizualizace, které jsou následně vráceny aplikaci.
- **Zobrazení výsledků** – Mobilní aplikace zobrazí vygenerované vizualizace uživateli, přičemž jsou k dispozici možnosti jako uložení, sdílení nebo další analýza.

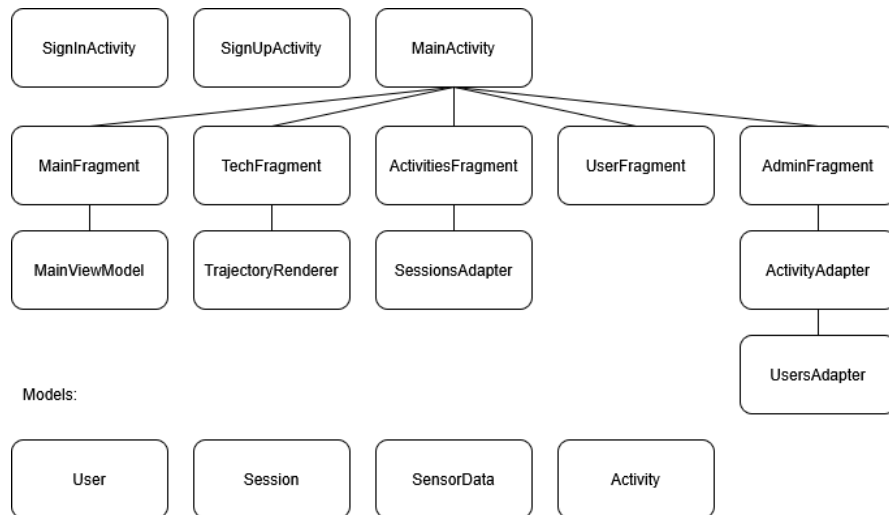
Kombinace Firebase Firestore pro ukládání a správu dat a Python API v Azure pro pokročilé zpracování a vizualizaci vytváří komplexní serverovou architekturu, která efektivně podporuje všechny požadované funkce aplikace. Tato hybridní architektura umožňuje využít silné stránky obou technologií – snadnou integraci a real-time funkcionalitu Firebase a pokročilé analytické a vizualizační schopnosti Python ekosystému.

2.4.3 Uživatelská část

Implementace uživatelské části aplikace je založena na modulární architektuře využívající principy MVVM (Model-View-ViewModel). Tato sekce popisuje strukturu a implementaci uživatelského rozhraní aplikace, jak je znázorněno ve schématu.

Architektura uživatelského rozhraní

Aplikace využívá hierarchickou strukturu, kde MainActivity slouží jako hlavní kontejner pro fragmenty a koordinuje navigaci. Přihlašování a registrace jsou řešeny samostatnými aktivitami (SignInActivity, SignUpActivity).



Obrázek 9: Architektura uživatelského rozhraní (zdroj – vlastní)

Komponenty architektury

- **Activities (View):** MainActivity jako centrální navigační uzel a samostatné aktivity pro autentizaci.
- **Fragments (View):** Pět klíčových fragmentů – MainFragment (úvodní obrazovka), TechFragment (technické nastavení), ActivitiesFragment (správa relací měření), UserFragment (uživatelský profil) a AdminFragment (administrátorské rozhraní).
- **ViewModels:** MainViewModel implementuje business logiku pro MainFragment a zajišťuje persistenci dat nezávisle na životním cyklu UI.
- **Adapters:** SessionsAdapter, ActivityAdapter a UsersAdapter zajišťují efektivní zobrazení seznamů dat v UI komponentách.
- **Models:** Základní datové entity aplikace – User, Session, SensorData a Activity – představují datovou vrstvu.
- **Specializované komponenty:** TrajectoryRenderer poskytuje funkcionalitu pro vizualizaci trajektorií pohybu.

Tato architektura zajišťuje jednosměrný tok dat: uživatelské akce jsou zpracovány ve vrstvě View, business logika je řešena ve vrstvě ViewModel a datové operace probíhají ve vrstvě Model. Výsledkem je modulární, testovatelný a udržitelný kód s jasně definovanými zodpovědnostmi jednotlivých komponent.

2.4.4 Rozhraní aplikace

Uživatelské rozhraní aplikace je navrženo s důrazem na intuitivnost, přehlednost a snadnou ovladatelnost. Aplikace implementuje moderní Material Design principy a poskytuje responzivní uživatelskou zkušenost optimalizovanou pro mobilní zařízení.

Rozhraní je strukturováno do logických celků, přičemž každá obrazovka má specifický účel a jasně definovanou funkcionalitu. Navigace mezi jednotlivými částmi aplikace je realizována prostřednictvím bottom navigace a přímých přechodů, což uživateli umožňuje rychlý a efektivní přístup ke všem funkcím.

V následujících podsekcích jsou detailně popsány jednotlivé obrazovky aplikace, jejich funkcionalita a způsob interakce s uživatelem.

2.4.4.1 Přihlašování a registrace

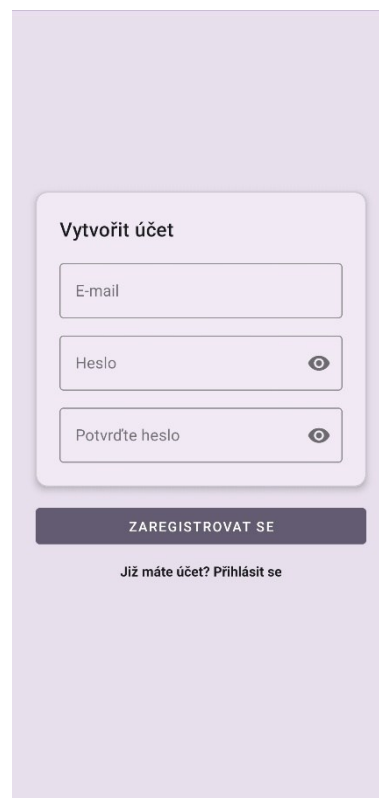
Vstupním bodem do aplikace je systém autentizace, který zajišťuje bezpečný přístup k funkcím sledování pohybu. Aplikace implementuje dvojici obrazovek pro přihlášení a registraci s konzistentním designem.

Přihlašovací obrazovka obsahuje formulář s emailem a heslem, přičemž pole pro heslo má funkci zobrazení/skrytí pomocí ikony oka. Hlavní tlačítko „PŘIHLÁŠIT SE“ spustí autentizaci a spodní odkaz umožňuje přechod na registraci.

Registrační obrazovka rozšiřuje formulář o pole „Potvrďte heslo“ s identickou funkcionalitou. Tlačítko „ZAREGISTROVAT SE“ vytvoří nový účet a spodní odkaz umožňuje návrat k přihlášení. Oba formuláře využívají validaci vstupů a poskytují přehledné chybové zprávy.



Obrázek 10: Přihlašovací obrazovka (zdroj – vlastní)



Obrázek 11: Registrační obrazovka (zdroj – vlastní)

2.4.4.2 Hlavní obrazovka

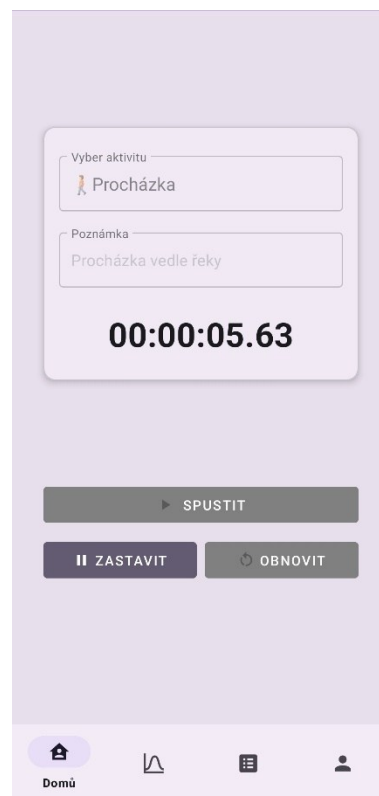
Po úspěšném přihlášení se uživatel dostává na hlavní obrazovku, která slouží jako ovládací panel pro sledování pohybu. Obrazovka obsahuje formulář pro nastavení nové relace a ovládací prvky pro řízení měření.

Konfigurace relace zahrnuje dropdown menu „Vybrat aktivitu“ s přednastavenými typy pohybových aktivit (Běh, Procházka, Cvičení) a textové pole „Poznámka“ pro volitelný popis relace. Centrální pozici zaujímá časovač zobrazující aktuální délku měření ve formátu 00:00:00.00.

Ovládací tlačítka umožňují kompletní řízení procesu měření – „SPUSTIT“ zahájí novou relaci se sbíráním dat ze senzorů, „ZASTAVIT“ ukončí aktivní měření a „OBNOVIT“ resetuje časovač. Spodní část obrazovky obsahuje navigační lištu s pěti sekcemi aplikace pro rychlý přístup k ostatním funkcím.



Obrázek 12: Hlavní obrazovka – konfigurace (zdroj – vlastní)



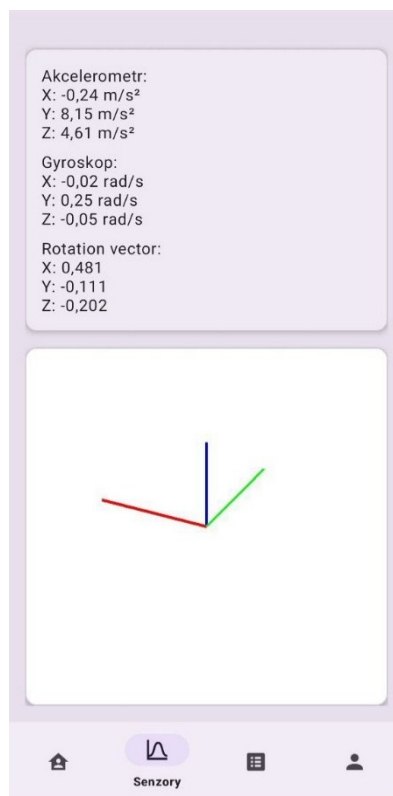
Obrázek 13: Hlavní obrazovka (zdroj – vlastní)

2.4.4.3 Obrazovka senzorů

Obrazovka senzorů poskytuje real-time monitoring a vizualizaci dat ze senzorů mobilního zařízení během měření. Slouží jako diagnostický nástroj pro kontrolu správné funkce sensorového systému a umožňuje uživateli sledovat aktuální hodnoty snímané zařízením.

Číselné hodnoty jsou zobrazeny v horní části ve třech sekcích – akcelerometr (m/s^2), gyroskop (rad/s) a rotační vektor (bezrozměrné hodnoty). Každý senzor ukazuje hodnoty pro osy X, Y a Z s přesností na dvě desetinná místa, což poskytuje detailní přehled o orientaci a pohybu zařízení.

3D vizualizace v dolní části obrazovky graficky reprezentuje orientaci zařízení pomocí trojice barevných vektorů (červená, modrá, zelená) odpovídajících lokálním osám souřadného systému. Tato vizualizace umožňuje intuitivní porozumění aktuální poloze a orientaci telefonu v prostoru.



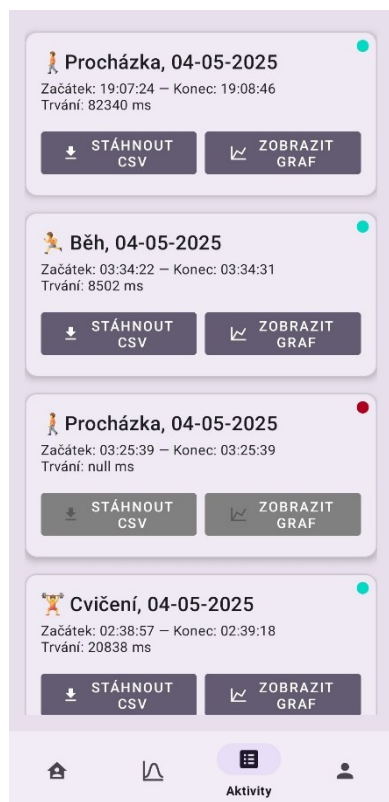
Obrázek 14: Obrazovka senzorů (zdroj – vlastní)

2.4.4.4 Seznam aktivit

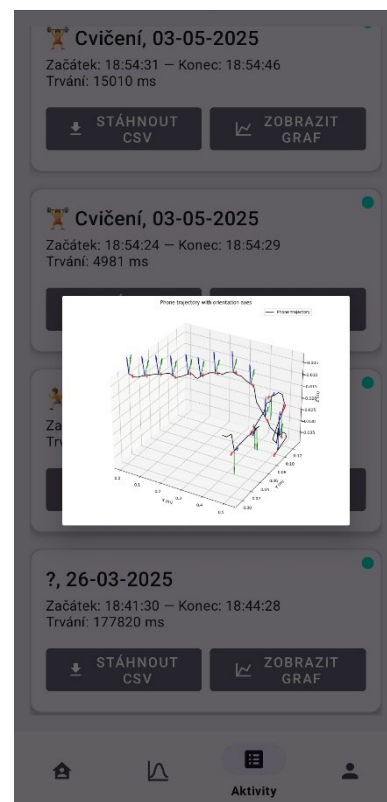
Obrazovka aktivit poskytuje přehled všech naměřených relací s možnostmi exportu a vizualizace dat. Seznam je organizován chronologicky s nejnovějšími relacemi na vrcholu.

Karta relace obsahuje hlavičku s ikonou typu aktivity, názvem a datem provedení. Pod hlavičkou jsou časové údaje zahrnující začátek a konec relace spolu s celkovým trváním v milisekundách. Barevný indikátor v pravém horním rohu signalizuje stav relace – zelený pro dokončené měření, červený pro nedokončené nebo přerušené relace.

Funkční tlačítka umožňují export dat tlačítkem „STÁHNOUT CSV“ pro následnou analýzu v externích nástrojích a „ZOBRAZIT GRAF“ pro interaktivní 3D vizualizaci trajektorie pohybu. Obě tlačítka jsou aktivní pouze u dokončených relací (zelený indikátor), u nedokončených jsou neaktivní a zobrazená šedě. Visualizace grafu zobrazuje trajektorii telefonu s orientačními vektory, které ukazují směr pohybu a natočení zařízení v prostoru.



Obrázek 15: Seznam aktivit (zdroj – vlastní)



Obrázek 16: Seznam aktivit – graf (zdroj – vlastní)

2.4.4.5 Uživatelská obrazovka

Uživatelská obrazovka poskytuje přehled účtu a možnosti správy profilu. Je rozdělena do několika logických sekcí pro snadnou orientaci a přístup k funkcím.

Statistiky a profil zobrazují počet provedených aktivit v horní kartě a emailovou adresu přihlášeného uživatele. Sekce správy účtu obsahuje tři funkční tlačítka – „ZMĚNIT HESLO“ pro aktualizaci přihlašovacích údajů, „SMAZAT ÚČET“ pro trvalé odstranění profilu a „ODHLÁSIT SE“ pro ukončení session.

Nastavení aplikace umožňuje přizpůsobení uživatelského rozhraní prostřednictvím výběru tématu. Uživatel může volit mezi třemi možnostmi – „Podle systému“ pro automatické přizpůsobení podle nastavení zařízení, „Světlé“ pro jasný vzhled aplikace nebo „Tmavé“ pro tmavý režim šetřící baterii a oči při používání za snížených světelných podmínek.



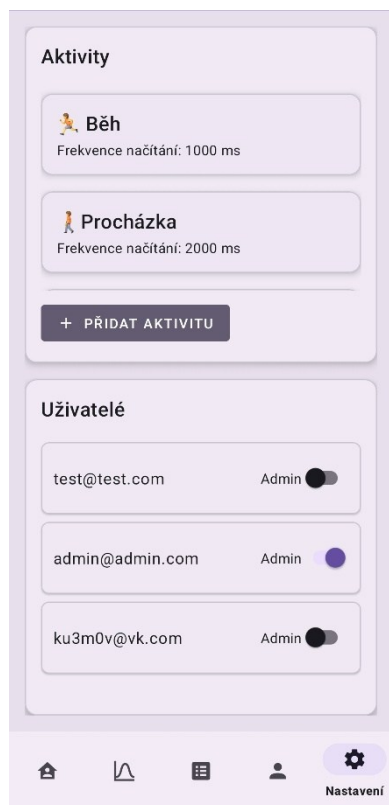
Obrázek 17: Uživatelská obrazovka (zdroj – vlastní)

2.4.4.6 Administrátorská obrazovka

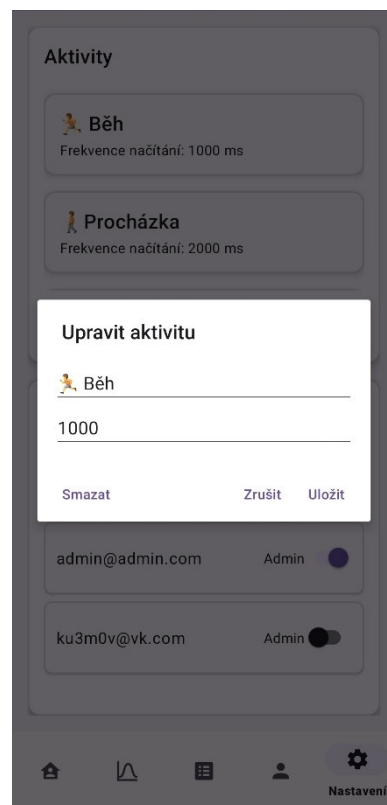
Tato obrazovka je přístupná pouze uživatelům s administrátorskými právy a poskytuje rozšířené možnosti správy systému. Je rozdělena do dvou hlavních sekcí pro komplexní správu aplikace.

Správa aktivit umožňuje administrátorovi konfigurovat typy pohybových aktivit dostupných v aplikaci. Každá aktivita má definovanou frekvenci načítání dat v milisekundách – například „Běh“ s frekvencí 1000 ms a „Procházka“ s 2000 ms. Tlačítko „PŘIDAT AKTIVITU“ umožňuje vytvoření nových typů aktivit s vlastními parametry měření.

Správa uživatelů zobrazuje seznam všech registrovaných uživatelů s možností úpravy jejich rolí. Každý uživatel je zobrazen s emailovou adresou a přepínačem role „Admin“, který administrátor může aktivovat nebo deaktivovat pro přidělení nebo odebrání administrátorských oprávnění. Tato funkce umožňuje flexibilní řízení přístupových práv v systému.



Obrázek 18: Administrátorská obrazovka (zdroj – vlastní)



Obrázek 19: Administrátorská obrazovka – správa aktivit (zdroj – vlastní)

2.5 Další možný rozvoj aplikace

Vyvinutá aplikace pro výzkum pohybových vzorců představuje solidní základ pro široké spektrum pokročilých analýz a aplikací v oblasti zdravotnictví, sportu a výzkumu. Shromážděná data ze senzorů mobilních zařízení otevírají nové možnosti v mnoha oblastech, kde může hrát klíčovou roli analýza lidského pohybu.

V oblasti rehabilitační medicíny mohou data z aplikace sloužit jako objektivní nástroj pro hodnocení pokroku pacientů během rehabilitačního procesu. Fyzioterapeuti a lékaři by mohli sledovat změny v pohybových vzorcích v průběhu času a přizpůsobovat terapeutické postupy podle konkrétních potřeb každého pacienta. Dlouhodobý sběr dat by umožnil vytvoření personalizovaných rehabilitačních programů založených na skutečném pokroku pacienta.

Pro neurologické pacienty, zejména ty s Parkinsonovou chorobou, roztroušenou sklerózou nebo po cévní mozkové příhodě, by aplikace mohla poskytovat kontinuální monitoring symptomů v domácím prostředí. Změny v charakteristikách chůze, třesu nebo koordinaci pohybů by mohly být detekovány dříve, než by byly patrné při běžných lékařských prohlídkách, což by umožnilo včasné úpravy medikace nebo léčebných postupů.

Integrace s algoritmy umělé inteligence a strojového učení představuje jednu z nejslibnějších cest budoucího rozvoje. Velké množství anonymizovaných dat o pohybových vzorcích by mohlo být využito pro trénink algoritmů schopných automaticky rozpoznávat abnormální pohybové vzorce, predikovat riziko pádu u starších osob nebo identifikovat raná stadia neurodegenerativních onemocnění. Tyto AI modely by postupem času mohly dosáhnout takové přesnosti, že by se staly cenným diagnostickým nástrojem v klinické praxi.

Ve sportovní medicíně a tréninku by data z aplikace mohla přispět k optimalizaci sportovního výkonu a prevenci zranění. Analýza pohybových vzorců během různých sportovních aktivit by umožnila trenérům a sportovním lékařům identifikovat neefektivní techniky nebo rizikové pohybové vzorce. Personalizované doporučení založená na analýze individuálních pohybových charakteristik by mohla výrazně zlepšit sportovní výkon a snížit riziko úrazů.

V oblasti geriatrické péče o seniory by aplikace mohla sloužit jako systém včasného varování před rizikem pádů. Změny v stabilitě, rychlosti chůze nebo koordinaci pohybů by mohly být detekovány dříve, než dojde k vážnému úrazu. Takový systém by mohl být integrován do chytrých domácností nebo zařízení pro seniory a poskytovat pečovatелům včasné upozornění na změny ve zdravotním stavu.

Pro ergonomické aplikace v pracovním prostředí by data o pohybových vzorcích mohla přispět k identifikaci rizikových pracovních pozic nebo opakovaných pohybů, které mohou vést k poškození pohybového aparátu. Výsledky analýz by mohly být využity při návrhu bezpečnějších pracovních postupů nebo ergonomičtějšího vybavení.

V oblasti populačního zdraví by agregovaná anonymizovaná data mohla poskytnout cenné poznatky o pohybových vzorcích různých demografických skupin. Tyto informace by mohly být využity při navrhování veřejných zdravotních politik zaměřených na podporu fyzické aktivity nebo prevenci chorob souvisejících s nedostatkem pohybu.

Možnost vytváření rozsáhlých databází pohybových vzorců by otevřela nové možnosti pro akademický výzkum v oblasti biomechaniky, neurofyzologie a sportu. Výzkumníci by mohli analyzovat korelace mezi různými faktory jako věk, pohlaví, zdravotní stav a charakteristikami pohybu, což by vedlo k hlubšímu pochopení lidského pohybu a jeho změn v průběhu života.

Integrace s dalšími zdravotnickými technologiemi, jako jsou nositelné zařízení pro monitoring vitálních funkcí, by mohla vytvořit komplexní systém pro holistické hodnocení zdraví a fyzické kondice. V budoucnu by mohla aplikace sloužit jako základ pro rozvoj chytrých asistenčních technologií, které by pomáhaly lidem s pohybovými omezeními žít nezávislejší život.

ZÁVĚR

Cílem této bakalářské práce bylo vytvoření mobilní aplikace pro operační systém Android, která umožňuje zachycovat, analyzovat a vizualizovat údaje ze senzorů mobilního telefonu pro vyhodnocení pohybových vzorců uživatele. Všechny stanovené cíle bylo úspěšně dosaženo.

Byla vyvinuta plně funkční mobilní aplikace pro Android, která efektivně využívá integrované senzory mobilních zařízení (akcelerometr, gyroskop, rotační vektor) pro kontinuální sběr data o pohybu uživatelů. Aplikace úspěšně zachycuje pohybové vzorce během různých aktivit jako je chůze, běh nebo cvičení s konfigurovatelnou vzorkovací frekvencí podle typu aktivity.

Implementovaný systém zajišťuje, že všechna zachycená data jsou automaticky opatřena popisky obsahujícími typ aktivity, časové značky a metadata měření. Klíčovým úspěchem je realizace robustního systému anonymizace dat, kdy jsou identifikátory relací ukládány pouze lokálně na zařízení uživatele a v cloudové databázi nejsou udržovány žádné vazby mezi uživatelskými účty a naměřenými daty. Tím je zajištěna ochrana soukromí uživatelů v souladu s etickými standardy výzkumu.

Serverová část systému byla úspěšně implementována jako hybridní architektura kombinující Firebase Firestore pro ukládání dat a Python API v Microsoft Azure pro pokročilé zpracování a vizualizaci. Tento přístup umožňuje bezpečné ukládání anonymizovaných dat a jejich následné zpracování do podoby informativních 3D grafů zobrazujících trajektorii pohybu uživatele.

Práce prokázala, že běžně dostupná mobilní zařízení jsou schopna poskytovat kvalitní a informativně hodnotná data o pohybových vzorcích. Implementované algoritmy pro kompenzaci gravitace a filtraci šumu zajišťují, že surová data ze senzorů jsou transformována do formy vhodné pro analýzu. Možnost exportu dat ve formátu CSV a jejich vizualizace v podobě interaktivních 3D grafů potvrzuje, že získaná data obsahují dostatečné množství informací pro následné analýzy.

POUŽITÁ LITERATURA

- [1] Biomechanics – A Comprehensive Guide[online] 2024. [cit. 2025-04-30]. Dostupné z: <https://anatomynotes.org/biomechanics-a-comprehensive-guide/>
- [2] Alfonso Penichet-Tomas. Applied Biomechanics in Sports Performance, Injury Prevention, and Rehabilitation[online] 2024. [cit. 2024-05-01] Dostupné z: <https://www.mdpi.com/2076-3417/14/24/11623>
- [3] Smita Nayak a Rajesh Kumar Das. Application of Artificial Intelligence (AI) in Prosthetic and Orthotic Rehabilitation[online] 2020. [cit. 2024-05-01] Dostupné z: <https://www.intechopen.com/chapters/73486>
- [4] Laura Alejandra Espitia-Mora, Manuel Andrés Vélez-Guerrero a Mauro Callejas-Cuervo. Development of a Low-Cost Markerless Optical Motion Capture System for Gait Analysis and Anthropometric Parameter Quantification[online] 2024. [cit. 2024-05-01] Dostupné z: <https://www.mdpi.com/1424-8220/24/11/3371>
- [5] Lilian Genaro Motti, Alison Keogh, Killian Mcmanus a Barry R Greene. Human Movement Analysis: Introduction to Motion Capture and Applications for Health[online] 2020. [cit. 2025-05-01]. Dostupné z: https://www.researchgate.net/publication/350023676_Human_Movement_Analysis_Introduction_to_Motion_Capture_and_Applications_for_Health
- [6] Miika Lumme. MOBILE PLATFORM SENSORS FROM A PROGRAMMER'S POINT OF VIEW[online] 2015. [cit. 2024-05-03] Dostupné z: https://www.theseus.fi/bitstream/handle/10024/89230/Thesis_Miika_Lumme.pdf?sequence=1
- [7] Sensors and location[online] [cit. 2025-05-03]. Dostupné z: https://developer.android.com/develop/sensors-and-location/sensors/sensors_overview
- [8] Ramzi Halabi, Rahavi Selvarajan, Zixiong Lin, Calvin Herd, Xueying Li, Jana Kabrit, Meghasyam Tummalacherla, Elias Chaibub Neto a Abhishek Pratap. Comparative Assessment of Multimodal Sensor Data Quality Collected Using Android and iOS Smartphones in Real-World Settings[online] 2024. [cit. 2025-05-03]. Dostupné z: <https://www.mdpi.com/1424-8220/24/19/6246>
- [9] Hui Yie Teh, Andreas W. Kempa-Liehr a Kevin I-Kai Wang. Sensor data quality: a systematic review[online] 2020. [cit. 2025-05-04]. Dostupné z: <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-0285-1>
- [10] Signal processing/noise removal technology for preprocessing in sensor data analysis and application to anomaly detection[online] 2024. [cit. 2025-05-06]. Dostupné z: <https://newji.ai/japan-industry/signal-processing-noise-removal-technology-for-preprocessing-in-sensor-data-analysis-and-application-to-anomaly-detection/>
- [11] Amberle McKee. A Data Scientist's Guide to Signal Processing[online] 2023. [cit. 2025-05-06]. Dostupné z: https://www.datacamp.com/tutorial/a-data-scientists-guide-to-signal-processing?dc_referrer=https%3A%2F%2Fcopilot.microsoft.com%2F

- [12] Maryam Mehrnezhad. On the Security of Mobile Sensors[online] 2017. [cit. 2025-05-06]. Dostupné z: <file:///C:/Users/ku3m0v/Downloads/Mehrnezhad,%20M.%202017.pdf>
- [13] Yang Zhou, Chaoyang Chen, Juan Ni, Guoxin Ni, Min Li, Guanghua Xu, John Cavanaugh, Mark Cheng, Stephen Lemos. EMG Signal Processing for Hand Motion Pattern Recognition Using Machine Learning Algorithms[online] 2020. [cit. 2025-05-06]. Dostupné z: <https://www.scientificarchives.com/article/emg-signal-processing-for-hand-motion-pattern-recognition-using-machine-learning-algorithms>
- [14] Movesense Showcase app[online]. [cit. 2025-05-06]. Dostupné z: <https://play.google.com/store/apps/details?id=com.movesense.showcaseapp>
- [15] Gait Analyzer[online]. [cit. 2025-05-06]. Dostupné z: <https://play.google.com/store/apps/details?id=com.matlabgeeks.gaitanalysis>
- [16] Vinicius Fulber-Garcia, Michal Aibin. Requirements: Functional vs. Non-functional[online] 2024. [cit. 2025-05-09]. Dostupné z: <https://www.baeldung.com/cs/requirements-functional-vs-non-functional>
- [17] Joseph Dewey. Business Process Management (BPM)[online] 2021. [cit. 2025-05-09]. Dostupné z: <https://www.ebsco.com/research-starters/business-and-management/business-process-management-bpm>
- [18] Clara Maine. A Comprehensive Kotlin Learning Guide for All Levels[online] 2024. [cit. 2025-05-10]. Dostupné z: <https://blog.jetbrains.com/education/2024/04/04/kotlin-learning-guide/>
- [19] Python: A Versatile Programming Language[online] 2024. [cit. 2025-05-10]. Dostupné z: <https://thecodeacademy.co.uk/docs/python/python/python-a-versatile-programming-language/>
- [20] Dr. Balvinder Taneja. Leveraging Android XML in App Development[online]. [cit. 2025-05-10]. Dostupné z: <https://drbtaneja.com/leveraging-android-xml-in-app-development/>
- [21] Material Design for Android[online]. [cit. 2025-05-11]. Dostupné z: <https://developer.android.com/develop/ui/views/theming/look-and-feel>
- [22] Google. Android Studio[online]. [cit. 2025-05-11]. Dostupné z: <https://developer.android.com/studio/>
- [23] JetBrains. PyCharm[online]. [cit. 2025-05-11]. Dostupné z: <https://www.jetbrains.com/pycharm/>
- [24] Microsoft. Hosting applications on Azure[online]. [cit. 2025-05-12]. Dostupné z: <https://learn.microsoft.com/en-us/azure/developer/intro/hosting-apps-on-azure>

SEZNAM PŘÍLOH

PŘÍLOHA A: Zdrojový kód mobilní aplikace – HumanMotionTracker	65
PŘÍLOHA B: Zdrojový kód API – HumanMotionAPI	66

PŘÍLOHA A: Zdrojový kód mobilní aplikace – HumanMotionTracker

Složka HumanMotionTracker.zip obsahuje:

1. Zdrojový kód aplikace
2. Grafické uživatelské rozhraní

PŘÍLOHA B: Zdrojový kód API – HumanMotionAPI

Složka HumanMotionAPI.zip obsahuje zdrojový kód API