

Slovenská technická univerzita v Bratislave  
Fakulta informatiky a informačných technológií  
Ilkovičova 2, 842 16, Bratislava 4

---

# Manažment zdravotného stavu pacienta prostredníctvom monitoringu emócií [eMotion]

Dokumentácia k tímovému projektu časť 2  
(Inžinierske dielo)

Pedagogický vedúci tímu: Ing. Gašpar Peter

Externý vedúci tímu: Ing. Lehocký Fedor, PhD.

Členovia tímu: Bc. Bobotová Zuzana, Bc. Černák Dávid, Bc. Gondová Veronika, Bc. Matlovič  
Tomáš, Bc. Pavlovič Tomáš, Bc. Šmihla Ján

Akademický rok: 2016 / 2017

Verzia číslo: 2

Dátum poslednej zmeny: 11.12.2016

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
1.1	Globálne ciele pre ZS . . . . .	1
<b>2</b>	<b>Celkový pohľad na systém</b>	<b>2</b>
<b>3</b>	<b>Čítanie odborných článkov</b>	<b>5</b>
3.1	Analýza detekcie stresu pomocou biometrických senzorov . . . . .	5
3.2	Analýza prístupov k monitorovaniu stresu . . . . .	7
3.3	Určovanie sentimentu z textu a hovorového slova . . . . .	13
3.4	Prístupy využívajúce akcelometer . . . . .	14
3.5	Zhrnutie . . . . .	15
<b>4</b>	<b>Moduly systému</b>	<b>17</b>
4.1	Mobil - zber dát . . . . .	17
4.1.1	PO3 (Pulse oxygen sensor) . . . . .	17
4.1.2	AM3 (krokomer) . . . . .	17
4.1.3	Bluetooth (zariadenia v okolí) . . . . .	18
4.1.4	Komunikačný modul . . . . .	19
4.1.5	Autentifikácia používateľa . . . . .	20
4.1.6	Kalendárove události . . . . .	21
4.1.7	WiFi modul . . . . .	22
4.1.8	GPS modul . . . . .	22
4.1.9	SMS modul . . . . .	24
4.1.10	Modul na detekciu hovorov . . . . .	25
4.2	Web . . . . .	27
4.2.1	Autentifikácia používateľa . . . . .	27
4.2.2	REST API na príjem dát . . . . .	28
4.2.3	Dashboard s dátami používateľa . . . . .	30
4.2.4	Modul sentimentu . . . . .	31

# 1 Úvod

Emócie, či už pozitívne alebo negatívne, majú rôzny vplyv na náš zdravotný stav. Najvýraznejšie na zdravie však vplýva stres - obzvlášť ak sa jedná o ľudí trpiacich chorobami ako je cukrovka. Stres je v dnešnej dobe rozšírený natoľko, že ho ľudia prestávajú vnímať a uvažovať nad ním ako nad problémom. Aby bolo možné stres liečiť, je ho potrebné vedieť správne monitorovať. Cieľom tohto projektu je vytvoriť systém, ktorý umožní zber dát potrebných na identifikáciu stresu, poskytovať tieto dáta lekárom, vyhodnocovať ich a zobrazovať používateľom spätnú väzbu.

Tento dokument obsahuje informácie o stave projektu po prvých 3 šprintoch. Hlavným cieľom tohto dokumentu je oboznámenie čitateľa s technickou stránkou projektu. Dokument obsahuje podrobný opis globálnych cieľov na zimný semester a celkový pohľad na systém zahŕňajúci opis architektúry systému ako aj jednotlivé moduly systému.

Moduly systému sa delia do dvoch častí - na moduly mobilnej aplikácie a moduly webovej aplikácie. Opis každého jedného modulu opisuje analýzu, návrh, implementáciu a testovanie daného modulu. V analýze sa zameriavame na preštudovanú literatúru a knižnice. V návrhu sú uvedené zvážené možnosti riešenia a v implementácii využité technológie. V testovaní je opísaný postup testovania konkrétneho modulu.

## 1.1 Globálne ciele pre ZS

Aby dokázal náš model na analýzu emócií (v prvom rade stresu) a odporúčanie správne pracovať, potrebujeme dostatočné množstvo dát. Preto je našim hlavným cieľom v zimnom semestri vytvoriť časti systému, ktoré umožnia zbierať dáta na mobilných zariadeniach používateľov a následne ich odosielať na server. K tomu je potrebné vytvoriť mobilnú aplikáciu, ktorá dokáže pracovať s rôznymi zdrojmi dát v systéme Android a taktiež externými senzormi (Oximeter, Krokomer). V mobilnej aplikácii sa budú zbierané dáta zobrazovať a zároveň sa budú odosielať na server.

Aplikácia na strane servera musí byť schopná spravovať používateľské účty, vytvárať nové účty, či už lekárov alebo pacientov, prijímať dáta z mobilnej aplikácie a efektívne ich uložiť. Prihlásený používateľ si bude môcť zobrazit dáta, ktoré odoslal jeho mobilný telefón.

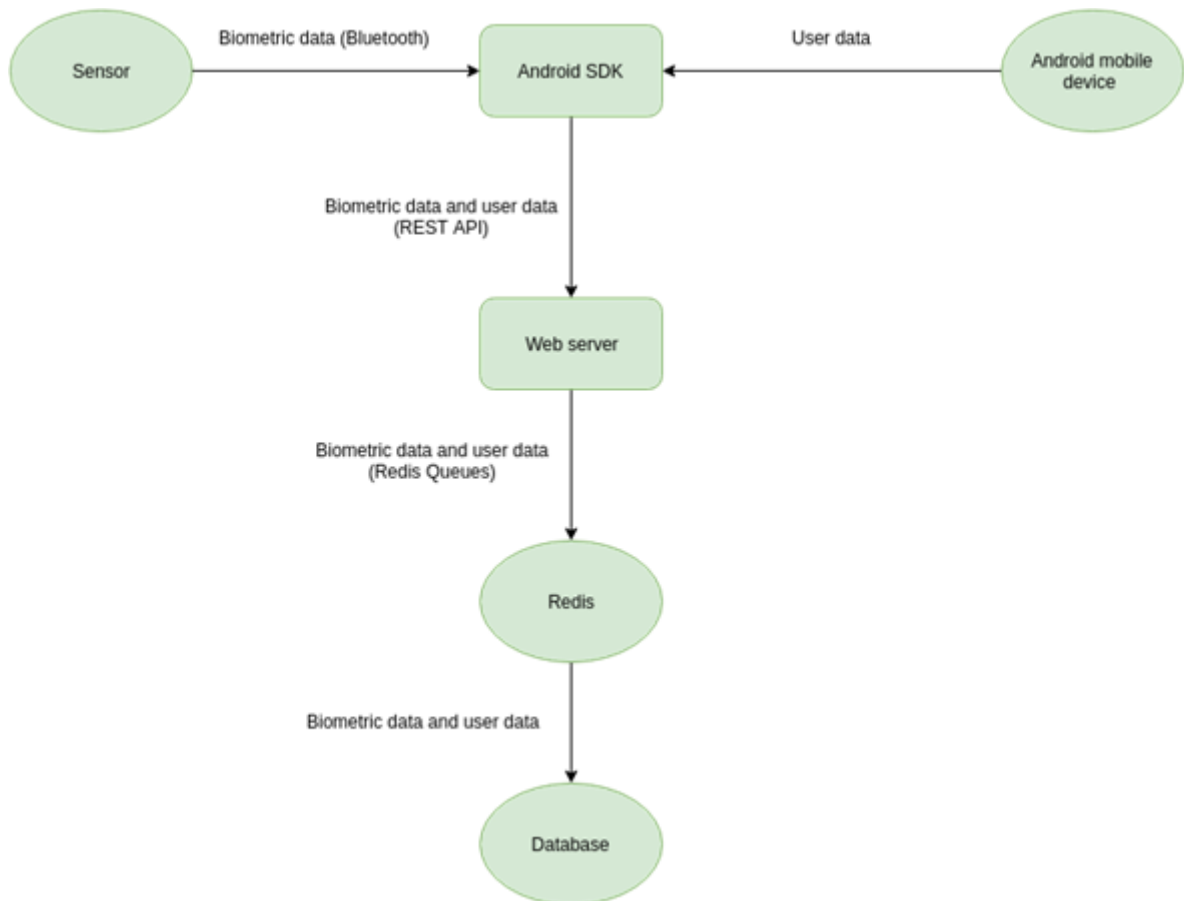
## 2 Celkový pohľad na systém

Náš systém sa skladá z dvoch hlavných častí: mobilná aplikácia a webová aplikácia. Mobilná aplikácia je vyvíjaná na platforme Android a webovú aplikáciu tvoríme s použitím Python webového frameworku - Django.

Na obrázku č. 1 môžeme vidieť diagram toku dát. Dáta z biometrických senzorov sú odosielané do mobilnej aplikácie cez Bluetooth. Následne sú odosielané na webový server pomocou nami vytvoreného REST API (ak mobilné zariadenie nemá v danom čase prístup na internet, dáta sú dočasne uložené v zariadení). Spolu s dátami zo senzorov posielame aj dáta o používateľovi a jeho zariadení - kalendár, sms, gps, dostupné wifi pripojenia, viditeľné Bluetooth zariadenia, denník hovorov.

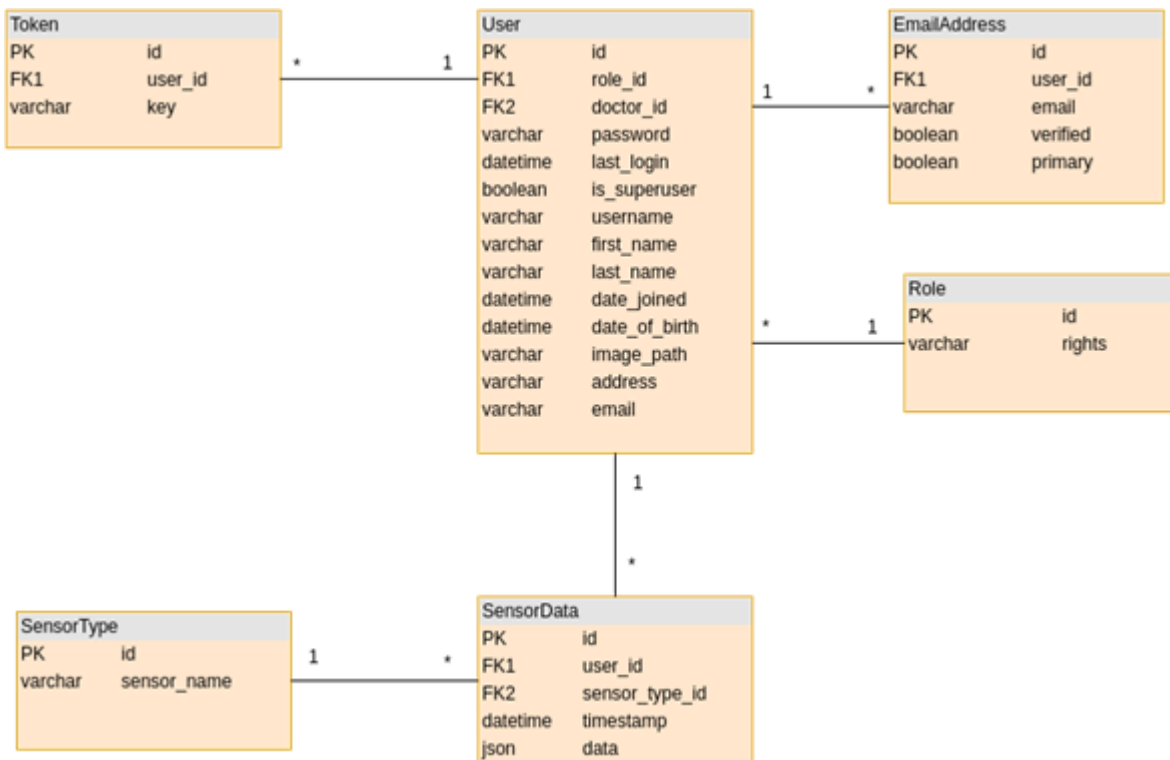
Z dôvodu zachovania čistoty dát sme implementovali na strane servera validáciu dát pomocou JSON schém. Dáta z jednotlivých senzorov a dáta z mobilného zariadenia musia mať predpísanú štruktúru, inak nebudú ďalej spracované.

Keďže biometrické dáta su zaznamenávané s relatívne veľkou frekvenciou, rozhodli sme sa použiť tzv. Redis queues. Redis dokáže vďaka využitiu pamäte RAM ukladať dáta s veľkou rýchlosťou. V prvom rade teda dáta uložíme do REDISU. Následne sú dáta postupne vyberané z Redisu a ukladané do relačnej databázy PostgreSQL. Použitím takejto architektúry budeme schopní spracovávať dáta aj keby ich prišlo veľké množstvo naraz.



Obr. 1: *Diagram toku dát*

Na obrázku č. 2 je zobrazený použitý dátový model. V tabuľke User sa nachádzajú používatelia našej aplikácie. Každý používateľ má pridelenú rolu a môže mať viac emailových adries a tokenov. Biometrické dáta a dáta o používateľovi z jeho mobilného zariadenia sa nachádzajú v tabuľke SensorData. Táto tabuľka je následne naviazaná na tabuľku SensorType a teda v dátach sa nachádza informácia o tom z akého senzoru sú nazbierané. Typ senzoru môže byť napr. Oximeter, krokomer ale aj sms alebo gps. Vďaka tomu, že dáta sú vo formáte json, môžeme ich všetky uložiť do jednej tabuľky a znížiť tak komplexitu nášho modelu.



Obr. 2: *Dátový model*

Čo sa infraštruktúry týka, webová aplikácia je nasadená na serveri s operačným systémom Linux. Nachádza sa tam aj Redis spolu s relačnou databázou PostgreSQL. Pri testovaní a nasadzovaní novej verzie aplikácie používame kontinuálnu integráciu, a teda všetko je plne automatizované. Využívame na to službu Travis CI. Testovanie vykonáva priamo Travis, akonáhle deteguje zmeny v git repozitári. Oznámenie o výsledku testovania sa nám odosielajú do Slacku, kde na to máme vyhradený kanál. Ak išlo o zmenu v hlavnej vetve master tak Travis navyše pošle požiadavku na náš server kde následne prebehne nasadenie novej verzie. V prvom rade sa stiahne aktuálna verzia z git repozitára a následne sa reštartne webový server aby sa prejavili zmeny.

## 3 Čítanie odborných článkov

### 3.1 Analýza detekcie stresu pomocou biometrických senzorov

**W.D. Scherz, J. Ortega, R. Seepold, "Towards emotion pattern extraction with the help of stress detection techniques in order to enable a healthy life", ARCA XXVII Conference on Qualitative Systems and Applications in Diagnosis, Robotics and Ambient Intelligence, ISBN: 978-84-608-5599-6, 2015.**

V tomto článku je opísaný návrh metódy na rozpoznávanie rôznych vzorov stresu za účelom klasifikácie emócií. Využívali pri tom nízkonákladové EKG - elektrokardiograf (z angl. ECG - electrocardiograph). Toto EKG je neinvazívne, zaznamenáva dáta v reálnom čase a má tri elektródy, ktoré zaznamenávajú tep srdca. Z neho vypočítali tzv. RR intervaly, čo sú zjednodušene povedané intervaly medzi jednotlivými údermi srdca. Z týchto intervalov následne určujú variability tepu srdca a z nej následne zistujú hladinu stresu. Potenciál navrhutej metódy overili experimentom. Ukázalo sa, že použitím tejto metódy je možné odhaliť stres v rôznych podmienkach, pri vysokom alebo nízkom tepe srdca.

Dáta z EKG spracúvali pomocou mikročipu Arduino UNO. Priamo na mikročipe bežal aj samotný algoritmus na meranie stresu. Dáta bolo následne možné ukladať na SD kartu alebo posielat' cez bluetooth napr. do smartfónu.

**W.D. Scherz, J. Ortega, N. Martinez Madrid, R. Seepold, "Heart Rate Variability indicating Stress visualized by Correlations Plots", Lecture Notes in Bioinformatics and Biomedical Engineering (LNBI), Volume 9044, Subseries of Lecture Notes in Computer Science, Springer International Publishing, ISSN 0302-9743, 2015.**

V tomto článku sa autori rozhodli vyvinúť nejaký ľahko nositeľný a lacný prístroj, ktorý by vedel spolupracovať s mobilom. Sústredili sa na ECG - aby ukázali, že ako veľmi to má vplyv pri strese. Vytvorili systém tak, aby sa tam dali pridať aj iné senzory.

Systém pozostáva z viacerých častí: ECG senzor a mikroprocesor, ktorý spracúva dáta a zariadenie na zobrazenie dát pre spätnú väzbu. ECG senzor funguje tak, že má tri elektródy, ktoré vysielajú analógový signál mikroprocesoru, ktorý ho transformuje na digitálny a vykoná pre-processing a filtráciu. Následne sa z digitálnych dát vypočíta HR a RR interval (interval medzi peakmi). Stress detection metóda je založená na ECG signále, z HR a RR intervalu vyjadria HRV

a to použijú na určenie stresu. Na evaluáciu využili metódu, pre ktorú je dokázane, že zvyšuje stres. Zistili teda, že stres môže byť zistený na základe meniacej sa hodnoty z ECG.

**P. Datko, J. Martínez Fernández, R. Seepold, "Aggregating individual stress-levels for dynamic clusters", AITA – Workshop on Ambient Intelligence for Telemedicine and Automotive domains, ISBN 978-84-697-0147-8, 2014.**

V tomto článku sa autori zamerali na sledovanie správania vodičov vozidiel. Dospeli k záveru, že stres negatívne ovplyvňuje správanie jednotlivca a tým zvyšuje riziká ku vzniku dopravných nehôd.

Úrovně stresu v tomto kontexte sú založené na externých a interných faktoroch. Do externých faktorov patrí všetko to, čomu vodič čelí pred jazdou, či už aktívne alebo pasívne (nezamestnanosť, finančné problémy). Do vnútorných faktorov patria všetky aktívne úkony počas jazdy, či už časový tlak, sledovanie GPS zariadenia, alebo používania mobilu, ale aj správanie ostatných vodičov (napríklad niekto nedá prednosť v jazde).

V tejto práci vyvinuli systém, ktorý pomocou zhlukovania rozdeľuje vodičov do skupín (pohodlná jazda na diaľnici, rušná jazda v meste,...), monitoruje a analyzuje individuálne úrovne stresu vodičov a šíri relevantné informácie v rámci skupiny pre zvýšenie bezpečnosti. Merajú polohu vodičov a aj srdcový tep a úroveň stresu z externého senzora a posielajú na server pomocou technológií Global System for Mobile Communications (GSM), Universal Mobile Telecommunications System (UMTS) or Long Term Evolution (LTE). Na serveri sa to spracuje a vyhodnotí. V prípade zistenia nejakého anomálneho správania sa pošle danej skupine vodičov upozornenie.

**W.D. Scherz, J. Martínez Fernández, R. Seepold, "Monitoring driving behaviour and biometric data to detect stress patterns", AITA – Workshop on Ambient Intelligence for Telemedicine and Automotive domains, ISBN 978-84-697-0147-8, 2014.**

V tomto článku sa autori zamerali na meranie stresu vodičov motorových vozidiel na základe ich správania. Cieľom práce je poskytnúť vodičovi informáciu o jeho biologickom stave a tým zlepšiť kvalitu jazdy. Použili dve merania. Najskôr merajú stres vodiča pomocou dostupných senzorov (emWave) a pri tom paralelne monitorujú aj informácie o vozidle. A potom pre zvolenie relevantných dát, potom používajú e-Health sensory (SP02, Airflow, Body temperature, ECG, Glucometer, GSR, sphygmomanometer, accelerometer, EMG), pripojený na Arduino. Namerané dáta integrujú do dát extrahované z openDS simulátora (rýchlosť, otáčky, stupeň prevodu). Porovnanie oboch meraní pomôže zlepšiť kvalitu vyhodnocovania a tým pádom pravdepodobne zmenšiť šírku parametrov nameraných pomocou senzorov.



**W.D. Scherz, R. Seepold, "Detection of heart rate characteristics as an approach to distinguish between mental stress and physical activity", 12th Russian-German Conference on biomedical engineering (RGC), 2016**

Autori článku sa snažia monitorovať stres pomocou ECK, EKG. Ale tým že aj pri fyzických aktivitách môže systém vyhodnotiť, že je človek v strese, tak sa snažia v tomto výskume nájsť správne charakteristiky, aby vedeli rozlíšiť či sa jedna o psychickú aktivitu alebo stres.

Robili experiment, v ktorom merali krvný tlak a srdcový tep participantov. Experiment bol zložený z troch fáz, v prvej fáze účastníci relaxovali, v druhej fáze vykonávali fyzické aktivity a v tretej fáze boli v stresovej situácii. Prvá fáza trvala 15 minút, v druhej fáze vykonávali účastníci rôzne cvičenia, na stacionárnom bicykli, každé z nich trvalo 5 minút. V poslednej fáze vykonali na účastníkov Stropov test, TSST a riešiť rôzne matematické úlohy v obmedzenom čase. Účastníci boli študenti vo veku od 18 do 40 rokov. Nikto z nich nebol vrcholový športovec alebo fajčiar.

Metóda rozlišovania fyzickej aktivity od stresu je založená na dvoch princípoch, prvý princíp je detekcia rýchlej zmeny frekvencie srdcového tepu, predpokladá sa, že tieto zmeny závisia od intenzity fyzickej aktivity. Druhá možnosť je založená na Furierovej analýze spektra, kde sa bude analyzovať správanie vysokej a nízkej frekvencie. Na to bude treba ale vykonať viac experimentov s väčším množstvom účastníkov.

## **3.2 Analýza prístupov k monitorovaniu stresu**

**WU, Wanqing, et al. Assessment of Biofeedback Training for Emotion Management Through Wearable Textile Physiological Monitoring System. IEEE Sensors Journal, 2015, 15.12: 7087-7095.**

Práca sa zaoberá meraním stresu prostredníctvom nositeľných senzorov a zaznamenávaním srdcovej frekvencie (HRV), ktoré môžu slúžiť na včasné varovanie pred stresom ako aj vizualizáciu emočného stavu používateľa. Výsledky štúdie ukazujú, že HRV je jeden zo základných prostriedkov na meranie stresu a úspešnosť odhadovania stresu pomocou srdcovej frekvencie je vysoká.

Práca sa taktiež zaoberala možnosťou rozlíšenia pozitívneho resp. negatívneho stresu prostredníctvom tzv. hladkých alebo nepravidelných vzorov v srdcovej frekvencii. Experimenty ukazujú, že negatívny stres je spravidla spojený s tzv. nepravidelnými vzormi v srdcovej frek-

vencii. Táto práca sa taktiež zaoberala vplyvom stresu na dýchanie človeka a korelácii medzi dýchaním a srdcovou frekvenciou pacienta.

**PAPAGEORGIU, Achilleas; ZIGOMITROS, Athanasios; PATSAKIS, Constantinos. Personalising and crowdsourcing stress management in urban environments via s-Health. In: Information, Intelligence, Systems and Applications (IISA), 2015 6th International Conference on. IEEE, 2015. p. 1-4.**

Táto práca sa zaoberá tematikou tzv. inteligentných miest. Jej cieľom je využiť možnosti, ktoré inteligentné mestá ponúkajú na zvýšenie zdravotnej starostlivosti v podobe monitoringu stresu obyvateľov a následného zásahu na hladinu stresu prostredníctvom pokročilých techník IKT. Na určovanie hladiny stresu v práci využívajú senzory, ktoré inteligentné mestá poskytujú, v rátače mobilov, ktoré využívajú samotní obyvatelia. Ich cieľom je zníženie stresu obyvateľov. Zdrojom stresu v meste môžu byť napr. preplnené ulice, zapchatá cesta alebo neprispôsobiví vodiči. Medzi možnosti odbúrania stresu, ktoré boli v práci prezentované, patrí zmena trasy cesty, prehratie obľúbenej hudby alebo nevyhnutná socializácia.

**AL OSMAN, Hussein; DONG, Haiwei; EL SADDIK, Abdulmotaleb. Ubiquitous Biofeedback Serious Game for Stress Management. IEEE Access, 2016, 4: 1274-1286.**

Táto práca je venovaná spôsobu odbúravania stresu prostredníctvom gamifikácie. Používatelia navrhutej aplikácie odbúravajú stres ovládaním hry na základe ich biometrických dát zo senzorov. To znamená, že pokiaľ má človek lepšie výsledky zdravotného stavu zo senzorov, o to lepšie skóre nadobúda v hre. Experimenty ukazujú, že ľudia, ktorí využívali prostriedky navrhutej aplikácie, dokázali lepšie ovládať svoj duševný stav a rýchlejšie sa vyrovnávali so vzniknutým stresom.

**CHEN, Kemeng, et al. Wearable sensor based stress management using integrated respiratory and ECG waveforms. In: 2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN). IEEE, 2015. p. 1-6.**

Práca sa zaoberá monitoringom stresu pacientov prostredníctvom nositeľných senzorov a mobilných zariadení a následnej spätnej väzbe v podobe dychových cvičení za účelom zníženia hladiny stresu pacienta. Na odhadovanie hladiny stresu pacienta sú využívané dve merania - variabilita srdcovej frekvencie a cyklus aktivity pacienta (prostredníctvom frekvencie dýchania pacienta). Z výsledkov vyplýva, že dychové cvičenia účinne znižujú úroveň stresu pacientov.

**ROMLI, Awanis; CHA, Arnidcha Peri. An expert system for stress management. In: Internet Technology and Secured Transactions, 2009. ICITST 2009. International Conference**

**for. IEEE, 2009. p. 1-6.**

Táto práca je venovaná expertným systémom, ktorých cieľom je určiť spôsob, ktorým jednotlivec dokáže najlepšie zvládať stres. Táto analýza je vykonaná na základe dvoch testov - záťažovým testom a testom osobnosti. Jej cieľom je vyvinúť efektívne odporúčanie aktivít používateľov, ktoré na základe osobnostných črt a na základe ich schopnosti zvládania stresu, vyhodnotí najvhodnejšiu aktivitu pre zníženie úrovni stresu. Výstupy tejto práce pomáhajú psychológom a expertom z danej oblasti rýchlejšie vyhodnotiť úroveň stresu a následnú liečbu pacientov.

**EID, Mohamad; AL OSMAN, Hussein; EL SADDIK, Abdulmotaleb. A mathematical model for personalized relaxation for stress management. In: Medical Measurements and Applications Proceedings (MeMeA), 2013 IEEE International Symposium on. IEEE, 2013. p. 201-206.**

Práca sa zaoberá vytvorením personalizovaného modelu, ktorý odporúča relaxačné cvičenia pre lepšie zvládanie stresu. Stres je monitorovaný na základe biometrických charakteristík pacientov. Cieľom tejto práce je naučiť používateľa, čo je pre neho najlepšie a ako dokáže najefektívnejšie zvládať stres. Táto práca bola overená prostredníctvom prípadových štúdií, ktoré preukázali úspešnosť matematického modelu. Navrhnutý matematický model je schopný odporúčať vhodné relaxačné cvičenia a je flexibilný na zmenu prostredia.

**CARBONARO, Nicola, et al. Wearable biomonitring system for stress management: A preliminary study on robust ECG signal processing. In: World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a. IEEE, 2011. p. 1-6.**

Táto práca je zameraná na návrh a vývoj pokročilých simulačných a snímacích technológií pre odhalovanie stresu na základe biosenzorov. Na vyhodnotenie úrovne stresu štúdia využíva EKG signál, na základe ktorého určuje srdcovú frekvenciu a variabilitu srdcovej frekvencie. V práci bol navrhnutý robustný systém, ktorý dokáže odhaliť stres pacientov a je odolný voči znečisteniu EKG signálu, prahovým hodnotám a zároveň poskytuje riešenie pre prostriedok na odhalovanie stresu s nízkou výpočtovou náročnosťou na spracovanie dát v reálnom čase.

**BAUER, Gerald; LUKOWICZ, Paul. Can smartphones detect stress-related changes in the behaviour of individuals?. In: Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on. IEEE, 2012. p. 423-426.**

Táto práca sa zaoberá odhalovaním stresu prostredníctvom mobilného zariadenia a prostriedkov,

ktoré mobilné zariadenie ponúka. Cieľom tejto štúdie je odhaliť, či je možné detegovať stres výlučne na základe bežného využívania mobilného telefónu. Štúdia vychádza z predpokladu, že človek sa správa inak pokiaľ je dlhodobo v strese a pokiaľ v strese nie je.

Súčasťou štúdie bol experiment, ktorý bol realizovaný na 7 študentoch počas skúškového obdobia. Počas tohto obdobia študenti využívali mobilné telefóny, z ktorých sa v pravidelných časových intervaloch odosielali dáta na server, ktoré následne slúžili na vyhodnotenie stresu študentov. Experiment potvrdil predpoklad o tom, že človek sa správa inak pokiaľ je vystavený dlhodobému stresu - navštevuje iné miesta, obmedzuje kontakt s ľuďmi a kontaktuje iba vyhradenú skupinu ľudí. V rámci tejto štúdie boli využité nasledovné typy dát:

Geografického správania sa používateľ'a - GPS a wifi Sociálna interakcia - Bluetooth Hovory - čísla, čas, dĺžka SMS - čísla, čas, dĺžka

**MADHURI, V. J.; MOHAN, Madhumitha R.; KAAVYA, R. Stress Management Using Artificial Intelligence. In: Advances in Computing and Communications (ICACC), 2013 Third International Conference on. IEEE, 2013. p. 54-57.**

Stres ako taký je jedným z hlavných faktorov, ktoré vedú k zdravotným problémom. V tejto práci sa zaoberajú návrhom systému na určovanie stresu z rôznych fyziologických dát - tep srdca, vodivosť pokožky, teplota tela, napätie svalov a krvný tlak. Rôzni ľudia reagujú na stresové udalosti rôznym spôsobom. Na určovanie stresu použili Fuzzy logiku, ktorá by mala zvládnuť aj takéto rozdiely. Týmto spôsobom mali taktiež možnosť určovať aj samotnú hladinu stresu a nie len či človek bol alebo nebol v strese. Po otestovaní navrhnutej metódy zistili, že prekonal doposiaľ používané prístupy ako SVM, NN a LDA.

**CERUTTI, Sergio; BIANCHI, Anna M.; REITER, Harald. Analysis of sleep and stress profiles from biomedical signal processing in wearable devices. Proceedings IEEE engineering in medicine and biology society (EMBC), 2006.**

V práci je opísaná metóda na meranie kvality spánku a manažment stresu na základe analýzy variability tepu srdca a dýchania. Analýza je zameraná najmä na dáta z prístrojov, ktoré slúžia na prevenciu proti kardiovaskulárnym chorobám a na celkové zlepšenie zdravia. Veľkou výhodou variability tepu srdca je, že vďaka nej je možné cenné informácie ohľadom stresu a kvality spánku bez potreby klinického zásahu. V práci bolo opísané ako rôzne úrovne stresu vplývajú na zdravie človeka a kvalitu jeho spánku.

**MAYYA, Subramanya, et al. Continuous monitoring of stress on smartphone using heart rate variability. In: Bioinformatics and Bioengineering (BIBE), 2015 IEEE 15th Internati-**

**onal Conference on. IEEE, 2015. p. 1-5.**

Práca popisuje metódu na meranie hladiny stresu pomocou merania tepu srdca. Metóda pozostáva zo získavania dát, predspracovania dát, vyhodnotenia a nakoniec sa modeluje skóre - hodnota stresu. Pri získavaní dát autori zvolili taký postup, že najskôr počítali hladinu stresu, keď bol človek v pokoji, a potom mu kázali vykonávať nejaké stresujúce aktivity, aby vedeli určiť zmeny. Medzi takéto aktivity patrí Stroop Colour Word Test, Mental Arithmetic Task (MAT), Memory task, Public Speaking task and Backward Counting task. Získané dáta bolo potrebné predspracovať, čo znamenalo odstránenie náhodných - chybových hodnôt. Následne boli dáta spracované, čo zahŕňalo extrakciu príznakov a štatistické analýzy príznakov. Výsledkom bola nameraná výška hladiny stresu, ktorú pekne graficky na mobilnom zariadení vizualizovali.

**GARCIA-MANCILLA, Jesus; GONZALEZ, Victor M. Stress Quantification Using a Wearable Device for Daily Feedback to Improve Stress Management. In: International Conference on Smart Health. Springer International Publishing, 2015. p. 204-209.**

Práca sa venuje novo navrhnutej metóde na kvantifikáciu stresu u človeka. Autori vychádzajú z toho, že keď je človek v strese, tak na to nervový systém nejakým spôsobom zareaguje. Na zisťovanie zmien využili nositeľné senzory, ktoré vedia merať tep srdca a teplotu kože. Namerané hodnoty potom použili v nimi navrhnutom vzorci, ktorý dokázal podľa týchto hodnôt určiť hladinu stresu. V článku bola zdôraznená využiteľnosť merania stresu pre medicínske účely, čo znamená sledovanie stresu pacienta, čo by malo dopomôcť k znižovaniu stresu a teda zlepšeniu životného štýlu a procesu liečenia.

**UDOCHUKWU, Orizu; HE, Yulan. A Rule-Based Approach to Implicit Emotion Detection in Text. In: International Conference on Applications of Natural Language to Information Systems. Springer International Publishing, 2015. p. 197-203.**

Článok hovorí o rozoznávaní emócií prostredníctvom textu. Autori uvádzajú, že pre určovanie emócií z textu využili známy OCC model. Ten pozostáva zo signifikantných slov pre dané emócie a určuje silu, akú dávajú danej emócii. Ide o rešpektovaný a uznávaný model pre určovanie emócií. V článku sa uvádza, že emócie sa zväčša rozoznávajú explicitne (napr. veta "Som šťastný, že som zvládol ten test"). Taktiež je uvedené, že implicitné rozpoznanie emócií je ťažšie (napr. veta "Spravil som ten test."). Autori vo svojom výskume na základe kľúčových slov, použitého času a pozitívnosti zvyšku vety detegovali emócie. Príklad: If Direction = "Self" and Tense = "Future" and Overall Polarity = "Positive" and Event Polarity = "Positive", then Emotion = "Hope".

**SUN, Yan, et al. Customer emotion detection by emotion expression analysis on adverbs.**

**Information Technology and Management, 2015, 16.4: 303-311.**

Práca hovorí o určovaní emócií pomocou prísloviak. Autori zhodnotili, že reč je silný nástroj na zisťovanie emócií a posúdili, že málo prác v oblasti rozpoznávania emócií sa zaoberá práve príslovkami. Ľudia pri rozhovoroch často používajú príslovky, ale nie všetky dávajú nejaké podfarbenie pre určenie pocitu. Preto ich treba extrahovať - vybrať tie najvhodnejšie (podobne ani všetky vety nedajú jasne najavo pocity). Počas práce si autori museli vyrobiť vlastný dataset, čo znamenalo rozdeliť príslovky, aby to mohli preskúmať.

**ZENONOS, Alexandros, et al. HealthyOffice: Mood recognition at work using smartphones and wearable sensors. In: 2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops). IEEE, 2016. p. 1-6.**

Cieľom práce bolo znížiť stres na pracovisku pomocou senzorov, ktoré merali dáta ako EKG, teplotu kože, akceleráciu. V tejto práci bola vyvinutá aplikácia, ktorej cieľom je zberať informácie o nálade od užívateľa. Aplikácia vyžadovala od používateľa spätnú väzbu každé dve hodiny, pričom sa užívateľ vyjadril k 8 náladám pomocou škály od 0 do 100. 0 - 19 žiadny náznak emócie 20 - 39 jemný náznak emócie 40 - 59 stredný náznak emócie 60 - 79 väčší náznak emócie 80 - 100 extrémny náznak emócie V práci predstavili svoju víziu, a dospeli k sľubným výsledkom, avšak, ako sami opisujú pracovali s malým počtom ľudí a nazbierali málo dát. Najľahšie sa detegovala emócia hnevu, za ňou nasledovali smútok, šťastie, stres, únava, nuda a pokoj. Pomocou senzorov sa podarilo dosiahnuť úspešnosť približne 70

**SANDULESCU, Virginia, et al. Mobile app for stress monitoring using voice features. In: E-Health and Bioengineering Conference (EHB), 2015. IEEE, 2015. p. 1-4.**

Cieľom tejto práce je detegovať stres pomocou hlasu. Za týmto účelom bola vyvinutá aplikácia StressID. Aplikácia sa spúšťa na pozadí určité hodiny v určitý čas, pričom pomocou knižnice TarsosDSP sa snaží nahráť 10 sekundovú audio stopu. Táto nahrávka sa následne rozdelí po 20 milisekúnd. Jednotlivé časti sa spracujú pomocou knižnice libSVM. Výsledok nahrávky sa následne odošle na server. V práci dosiahli úspešné výsledky s presnosťou 78% v určovaní pozitívnych alebo negatívnych nahrávok. Taktiež uvádzajú, že hlas je ukladaný iba dočasne - kým sa nespracuje, následne je vymazaný.

**FERDOUS, Raihana; OSMANI, Venet; MAYORA, Oscar. Smartphone app usage as a predictor of perceived stress levels at workplace. In: Proceedings of the 9th International Conference on Pervasive Computing Technologies for Healthcare. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2015. p. 225-228.**

Cieľom práce bola detekcia stresu na pracovisku pomocou detegovania, ako často užívatelia používajú konkrétne aplikácie. V experimente dali 28 pracujúcim ľuďom na 6 týždňov mobilný telefón, ktorý zaznamenával činnosti všetkých aplikácií na mobile - začiatok, koniec, dĺžka. Následne zhromaždili spätnú väzbu v ktorej účastníci tri krát denne odpovedali na úroveň stresu v konkrétnom časovom úseku. Najčastejšie využívané aplikácie boli mail, kalendár a prehliadač. Najmenej používané boli hry a nadbytočné aplikácie. Výsledky práce boli celkom prekvapivé pričom model sa učil zo 70% zozbieraných dát a zvyšných 30% určil na testovanie. Stres sa v priemere podarilo detegovať v 75%.

### 3.3 Určovanie sentimentu z textu a hovorového slova

Určovanie sentimentu zo slovenského textu je úloha, ktorou sa zaoberá viacero výskumných prác priamo na našej škole. Z tohto dôvodu sme zvážili využitie existujúcich prác, ktoré sú na našej škole dostupné.

Prvou možnosťou bolo kontaktovanie tímu Imagine Cup, ktorí vo svojej práci odhalujú nevhodné komentáre. Súčasťou ich práce je odhalovanie sentimentu textu komentárov a však primárne sa zameriavajú na anglický jazyk. Pre slovenský jazyk vytvorili jednoduchý určovač sentimentu na základe sentimentu jednotlivých slov (to znamená, že každé slovo malo hodnotu od -3 po 3 a následne jednotlivé hodnoty sčítali a predelili počtom slov v texte). Tento slovník našli na Githube a pochádza z Fakulty matematiky, fyziky a informatiku UK. Samostatné API k ich určovaču sentimentu však nemajú. Slovník, ktorý sme našli na githube:

<https://github.com/okruhlica/SlovakSentimentLexicon>

Ďalšou možnosťou bolo využitie určovania sentimentu anglického textu s prekladom slovenského textu do angličtiny. Microsoft Azure ponúka riešenie pre vyhodnocovanie sentimentu, a však iba v angličtine. Na preklad by sme mohli použiť štandardný prekladač a následne vkladať preložený text do API od Microsoft Azure, a však je otázne nakoľko by prekladač zmenil význam textu.

Poslednou možnosťou bolo využitie bakalárskej práce nášho spolužiaka Rast'a Krchňavého, ktorý sa v rámci svojej bakalárskej práce venoval problému sentimentu v slovenskom texte. Jeho práca je aktuálne nasadená na Heroku a určuje sentiment ako binárny problém - čiže kladné alebo záporné. Po komunikácii s Rast'om sme zistili, že je ochotný vytvoriť API šité na mieru. Rast'o nám taktiež poskytol možnosť preučenia jeho modelu, ktorý je aktuálne natrénovaný na facebookových komentároch.

## Emócie

Okrem samotného sentimentu slov je možné rozoznávajúce emócií z hovoreného slova. Jednou z možností je Vokaturi (<http://developers.vokaturi.com/getting-started/overview>), ktorí poskytujú zdrojový kód na stiahnutie. Je to voľne dostupné a bezplatné. Táto služba dokáže rozpoznať emóciu z hlasu, a však je dosť slabo zdokumentovaná. Samotná implementácia podporuje Python (ale len pre Mac alebo Windows, čo predstavuje problém).

Ďalšou možnosťou je napr.: <http://www.wcl.ece.upatras.gr/en/ai/resources/demo-emotion-recognition-from-speech>

Emócie sa okrem hovoreného slova dajú rozpoznať aj priamo z písaného textu. Existuje množstvo API, ktoré na základe vstupného textu a jeho následnej analýzy vrátia JSON, kde je percentuálne uvedené, k akej emócii text patrí. Problémom je opäť jazyk.

Príklad

[https://tone-analyzer-demo.mybluemix.net/?cm\\_mc\\_uid=25274456998314780159884&cm\\_mc\\_sid\\_50200000=14](https://tone-analyzer-demo.mybluemix.net/?cm_mc_uid=25274456998314780159884&cm_mc_sid_50200000=14)

Zaujímavý zdroj

<http://nordicapis.com/20-emotion-recognition-apis-that-will-leave-you-impressed-and-concerned/>

## 3.4 Prístupy využívajúce akcelometer

**RAVI, Nishkam, et al. Activity recognition from accelerometer data. In: AAI. 2005. p. 1541-1546.**

Táto práca sa zaoberá rozpoznávaním činnosti používateľ a na základe dát z akcelometra. Rozpoznávanie činností je vnímané ako problém klasifikácie. Dáta získané z akcelometra sú podrobené extrakcii funkcií a následne spracované algoritmi pre klasifikáciu. Akcelometer na základe experimentov dokáže s pomerne vysokou presnosťou rozoznávajú aktivity, ktoré sú spojené s pohybom, a však pokiaľ človek stojí na mieste a vykonáva nejakú aktivitu je to zložitejšie ju rozpoznať. Najväčšiu presnosť pri klasifikácii dosiahla metóda Plurality Voting.

**BREZMES, Tomas; GORRICH, Juan-Luis; COTRINA, Josep. Activity recognition from accelerometer data on a mobile phone. In: International Work-Conference on Artificial Neural Networks. Springer Berlin Heidelberg, 2009. p. 796-799.**



Táto práca sa zaoberá využitím bežného mobilného telefónu s akcelometrom na odhadovanie typu aktivity, ktorú používateľ vykonáva. Cieľom tejto štúdie je odhalenie možností telefónu na spracovanie dát z akcelometra v reálnom čase a ich následné vyhodnotenie. Celé spracovanie a vyhodnotenie nameraných dát bolo realizované v samotnom telefóne a teda dáta neboli preposielané na server. Súčasťou práce bol výber efektívnych metód na klasifikáciu, ktoré by boli schopné pracovať priamo v mobilnom zariadení.

**BAO, Ling; INTILLE, Stephen S. Activity recognition from user-annotated acceleration data. In: International Conference on Pervasive Computing. Springer Berlin Heidelberg, 2004. p. 1-17.**

V tejto práci sa autori zameriavali na rozpoznávanie aktivity na základe dvoch nositeľných akcelometrov, ktoré mali používatelia pripevnené na tele. Experiment bol vykonávaný na 20 účastníkoch. Samotnému experimentu predchádzala kalibrácia, kde používatelia simulovali jednotlivé typy činností. Úspešnosť navrhutej metódy je 84%. Medzi základné vlastnosti, ktoré boli skúmané patria - energia, frekvenčná entropia a korelácia dát z akcelometra.

**HONG, Yu-Jin, et al. Mobile health monitoring system based on activity recognition using accelerometer. Simulation Modelling Practice and Theory, 2010, 18.4: 446-455.**

V tejto práci sa autori zameriavajú na využitie akcelometra a RFID snímača na rozpoznávanie aktivity človeka. Na klasifikáciu 5 ľudských stavov sú v štúdiu využívané dva nositeľné snímače. Na klasifikáciu je využitý algoritmus rozhodovacieho stromu.

**SUN, Lin, et al. Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations. In: International Conference on Ubiquitous Intelligence and Computing. Springer Berlin Heidelberg, 2010. p. 548-562.**

Táto práca bola venovaná klasifikácii aktivity používateľa na základe dát z akcelometra telefónu. Medzi základné problémy, ktoré sú s mobilným akcelometrom spojené patrí poloha mobilného telefónu. Táto štúdia je zameraná na riešenie tohto problému. V práci bolo definovaných 6 základných pozícií mobilného telefónu a spôsoby klasifikácie na správne odhadnutie polohy telefónu v danom čase (SVM klasifikátor). Ďalším krokom bola samotná klasifikácia

aktivít na základe nameraných dát.

### **Odkazy na datasety:**

<https://archive.ics.uci.edu/ml/datasets/Activity+Recognition+from+Single+Chest-Mounted+Accelerometer#>

<https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones>

<https://archive.ics.uci.edu/ml/datasets/Dataset+for+ADL+Recognition+with+Wrist-worn+Accelerometer>

## **3.5 Zhrnutie**

V rámci výskumu detekcie stresu v rôznych prácach využívali biometrické senzory, hlavne EKG. Neinvazívne EKG senzory zachytávajú tep srdca pomocou elektród. Zo zachyteného signálu je následne možné vypočítať RR intervaly, čo sú jednoduche povedané, časové intervaly medzi jednotlivými údermi srdca. Z týchto intervalou následne určovali úroveň stresu. Vo viacerých prácach sa im taktiež podarilo odlíšiť stres od fyzickej aktivity. Opísané metódy boli relatívne presné a plánujeme ich použiť.

V ďalších prácach sa zaoberali tým ako odlíšiť negatívny stres od pozitívneho pomocou odhalovania korelácií medzi srdcovou frekvenciou a dýchaním. V iných výskumoch sa zaoberali možnosťami znižovaním stresu. Používali rôzne metódy ako:

- relaxačné cvičenia
- počúvanie hudby
- dychové cvičenia

Niektoré výskumy využívali taktiež dáta z mobilných telefónov a odhalili, že človek sa správa inak keď je pod vplyvom stresu. Túto metódu plánujeme použiť aj v našom projekte.

V rámci nášho projektu budeme taktiež určovať sentiment SMS správ mobilných zariadení. Tento údaj bude jedným z parametrov pre určovanie stresu. Na určovanie sentimentu využívame externé API vyvinuté na fakulte.

Plánovali sme taktiež zaznamenávať hovory z mobilných telefónov a následne určovať emócie z hlasu. Túto metódu zatiaľ neplánujeme využiť z dôvodu veľkej spotreby batérie pri zaznamenávaní hlasu. Taktiež by tu bol pravdepodobne problém s narúšaním súkromia.

Pomocou zmeny polohy mobilného telefónu je možné podľa rôznych výskumov určiť ak-

tivitu používateľ'a. Na tento účel sa využíva akcelerometer, ktorý je zabudovaný vo väčšine mobilných zariadeniach. Existuje viacero prác, ktoré sa tomuto problému venujú. Používajú buď vlastné dáta, alebo existujúce datasety. Na analýzu využívajú buď označovaný dataset alebo vlastné dáta s kalibráciou. Základným problémom môže byť poloha telefónu, ktorá sa dá riešiť klasifikáciou pozície telefónu. Úspešná klasifikácia aktivity používateľ'a môže dopomôcť k určeniu hladiny stresu.

# 4 Moduly systému

## 4.1 Mobil - zber dát

### 4.1.1 PO3 (Pulse oxygen sensor)

#### Analýza

Miera okysličenosti krvi a frekvencia krvného tepu hovorí o životnom štýle pacienta, konkrétne akým pohybovým aktivitám sa počas dňa venuje.

#### Návrh

Senzor P03 zbiera dáta pacienta ohľadom okysličenosti krvi (bloodOxygen) a krvného tepu (pulseRate). Je závislý od knižnice dodávateľa senzoru iHealth. Existuje aplikácia, ktorá dokáže komunikovať s daným sensorom cez Bluetooth, ale dáta posielajú na ich server. S kúskom št'astia by sa mohla dať vytvoriť implementácia v našej Android aplikácii, ktorá dokáže komunikovať so sensorom cez Bluetooth.

#### Implementácia

Na komunikáciu so sensorom použijeme knižnicu od iHealth<sup>1</sup>. Najskôr sensor detegujeme, pripojíme sa naň a potom budeme z neho ťahať dáta. Dáta budeme posielajú v pravidelných intervaloch (každých 60 sekúnd). Posielame ich vo formáte JSON<sup>2</sup> na adresu <http://147.175.149.141:8000/api/sensor/>.

#### Testovanie

Testovali sme odoslanie dát, či už jednorázovo, alebo v danom intervale (každých 60 sekúnd).

### 4.1.2 AM3 (krokomer)

#### Analýza

Počet prejdenných krokov a spálených kalórií dokáže čiastočne vypovedať o životnom štýle používateľa. Presne na toto sa dajú využiť hodinky AM3 od spoločnosti iHealth.

#### Návrh

---

<sup>1</sup><https://github.com/iHealthLabs/Android-SDK>

<sup>2</sup><https://github.com/Crakeh/eMotion-Web-App/wiki/API-Sensors>

Senzor AM3 zbiera dáta pacienta ohľadom počtu krokov za deň a spálených kalórií. Je závislí od knižnice dodávateľ a senzoru iHealth. Už existuje aplikácia, ktorá dokáže komunikovať s daným senzorom cez Bluetooth, ale dáta posielajú na ich server. S kúskom štastia by sa mohla dať vytvoriť implementácia v našej Android aplikácii, ktorá dokáže komunikovať so senzorom cez Bluetooth.

### **Implementácia**

Pre komunikáciu so senzorom použijeme knižnicu od iHealth<sup>3</sup>. Najskôr senzor detegujeme, potom sa na neho pripojíme a začne komunikácia. Dáta budeme posielajú v pravidelných intervaloch (každú hodinu).

### **Testovanie**

Testovali sme odoslanie dát, či už jednorázovo, alebo v danom intervale (každú hodinu).

## **4.1.3 Bluetooth (zariadenia v okolí)**

### **Analýza**

Zariadenia Bluetooth sa nachádzajú takmer všade - fotorámčeky, televízie, tlačiarne, mobilné telefóny a mnoho ďalších. Na základe okolitých Bluetooth zariadení bude teda možné zistiť približnú pozíciu používateľa. Táto pozícia môže mať vplyv na používateľovu hladinu stresu - špecifické Bluetooth zariadenia sa môžu nachádzať na pracovisku kde používateľ pracuje, v škole ktorú navštevuje, u zubára, prípadne rôzne ďalšie stresujúce prostredia. Bluetooth patrí k bežnej výbave moderných mobilných telefónov, takže pravdepodobnosť, že zariadenie, na ktorom bude bežať naša aplikácia nebude vybavené zariadením Bluetooth je prakticky nulová. Pre vyhľadanie okolitých zariadení je potrebné najprv zistiť, či je Bluetooth zapnutý, prípadne ho zapnúť. Po vykonaní skenu je potrebné Bluetooth automaticky vypnúť, pretože znižuje výdrž batérie. Zoznam získaných zariadení je následne potrebné odoslať na server.

### **Návrh**

Pred spustením skenovania je potrebné zistiť, či je zariadenie Bluetooth zapnuté. Ak nie je, treba ho zapnúť a uložiť jeho stav pred spustením skenovania. Ak bol pred skenovaním vypnutý, po ukončení skenovania ho aplikácia tiež vypne. Ak bol zapnutý, tak ho pravdepodobne používateľ práve používa a nechám ho zapnutý. Nájdene zariadenia sa následne odošlú na server pomocou komunikačného modulu, prípadne sa aj zobrazia v aplikácii.

---

<sup>3</sup><https://github.com/iHealthLabs/Android-SDK>

## Implementácia

Pre prácu so zariadením Bluetooth využijeme API, ktorú poskytuje operačný systém Android. Vytvoríme BroadcastReceiver, ktorý bude spracovávať udalosti *ACTION\_FOUND*, *BLUETOOTH\_DISCOVERY\_STARTED* a *BLUETOOTH\_DISCOVERY\_FINISHED*. Po zachytení udalosti *ACTION\_FOUND* overíme, či má zariadenie platný názov, pridáme ho do zoznamu zariadení a odošleme ho na zobrazenie do Aktivity BluetoothActivity, ak je aktívna. Po zachytení udalosti *BLUETOOTH\_DISCOVERY\_FINISHED* tento zoznam pošleme modulu na komunikáciu, ktorý ho následne odošle na server.

## Testovanie

Pre testovanie tohto modulu je potrebné zaistiť, aby bolo v okolí mobilného telefónu minimálne jedno Bluetooth zariadenie. Ak žiadne zariadenie dostupné nieje, test nebude možné uskutočniť. Následne je potrebné spustiť detekciu Bluetooth zariadení v tomto module. Detekcia prebieha automaticky po zapnutí telefónu a potom každých 5 minút, takže je potrebné počkať, alebo telefón reštartovať. Po ukončení detekcie zariadení budú na server odoslané nájdené zariadenia. Výsledok bude viditeľný medzi dátami práve prihláseného používateľa.

### 4.1.4 Komunikačný modul

#### Analýza

Zariadenie bude zbierať veľké množstvá údajov, ktoré bude potrebné následne odoslať na server. Server však môže byť často nedostupný. Či už z dôvodu výpadku internetového pripojenia alebo výpadku serveru samotného. V oboch prípadoch je potrebné dáta uložiť v mobilnom zariadení a na server ich odoslať neskôr. Služba komunikačného modulu musí byť vždy dostupná pre všetky senzory, ktoré odosielajú dáta na server. Musí teda zvládnuť paralelné požiadavky od viacerých tried.

#### Návrh

Komunikačný modul bude bežať na pozadí ako Servis. Dáta na odoslanie sa budú ukladať do zoznamu prostredníctvom volania verejnej metódy. Pre prípad, že túto metódu zavolajú dve vlákna naraz, bude potrebné zoznam chrániť mutexom. Okrem uloženia do zoznamu v pamäti, je dáta potrebné uložiť aj na perzistentné úložisko, aby sa nestratili v prípade vypnutia zariadenia. Po spustení služby bude potrebné zoznam z tohto úložiska tiež naplniť. Servis sa tieto dáta následne pokúsi odoslať na server. Ak nie je dostupný, po čase skúsi znova.

## Implementácia

Servis `DataSenderService` sa automaticky spustí po zapnutí aplikácie. Môže teda nastať prípad, že modul na zber dát zavolá metódu na odoslanie dát príliš skoro. V tomto prípade je potrebné najskôr počkať na inicializáciu `DataSenderService`. Zoznam, do ktorého sa budú ukladať dáta na odoslanie, bude inštancia triedy `ArrayList`.

Po zavolaní metódy na odoslanie dát sa zamkne mutex, ktorý tento zoznam chráni a až potom sa do neho vložia dáta. Dáta sa tiež uložia na persistentné úložisko pomocou `SharedPreferences`. Po vložení sa mutex odomkne. Servis obsahuje cyklus, ktorý čaká na naplnenie zoznamu dátami. Ak zistí, že sú v zozname dáta na odoslanie, vyberie zo zoznamu jeden `JSONArray` a pokúsi sa ho odoslať. Ak sa ho odoslať nepodarí, vráti ho do zoznamu a aj do `SharedPreferences`.

## Testovanie

Testovanie tohto modulu prebieha spoločne s testovaním ostatných modulov, ktoré odosielajú dáta na server (napr. Bluetooth modul).

### 4.1.5 Autentifikácia používateľa

#### Analýza

Dáta ktoré budú zozbierané mobilným zariadením musia byť odoslané na server spolu s informáciou, ktorému používateľovi patria, dáta ktoré nie sú priradené žiadnemu používateľovi ani nemá zmysel odosielať na server. Bude preto potrebné vytvoriť systém, ktorý umožní používateľovi prepojiť svoje mobilné zariadenie so svojim účtom vo webovej aplikácii. Pre pohodlné používanie bude tiež potrebné umožniť pacientovi registrovať sa priamo zo svojho mobilného zariadenia. Tiež bude potrebné používateľa automaticky prihlásiť po reštartovaní aplikácie.

#### Návrh

Aplikácia bude obsahovať možnosti v menu pre registráciu a prihlásenie. Po kliknutí na jednu z týchto možností sa zobrazí registračný resp. prihlasovací formulár, do ktorého používateľ zadá svoje informácie. Po používateľovom potvrdení registrácie resp. prihlásenia, bude odoslaná požiadavka na webový server, ktorý ju spracuje a ako odpoveď odošle informáciu o vykonaní požiadavky (úspech, zlyhanie a ďalšie dodatočné informácie). Po úspešnom prihlásení bude v odpovedi odoslaný token, ktorý sa uloží a bude prítomný v každej požiadavke na odoslanie dát na server. Pomocou tohto tokenu bude vedieť server priradiť dáta ku konkrétnemu používateľovi. Tiež bude potrebné vytvoriť verejnú metódu, pomocou budú môcť ostatné moduly zistiť či je/nie je používateľ práve prihlásený a získať jeho e-mail adresu.

## Implementácia

Požiadavka na prihlásenie resp. registráciu bude odoslaná na aplikačné rozhranie servera ako HTTP POST požiadavka. Ako odpoveď server odošle JSON objekt obsahujúci informácie o vykonaní požiadavky a v prípade úspešného prihlásenia aj spomínaný token. Tento token - reťazec 40 náhodných hexadecimálnych číslic - bude po úspešnom prihlásení uložený spolu s e-mailom používateľa do SharedPreferences. Automatické prihlásenie bude prebiehať v službe komunikačného modulu, ktorá sa spúšťa automaticky po zapnutí mobilného telefónu. Aby mohli ostatné moduly aplikácie zistiť, či je používateľ prihlásený, bude dostupná verejná metóda `IsLoggedIn()` a `GetLoggedEmail()`.

### 4.1.6 Kalendárove udalosti

#### Analýza

Pre lepšiu analýzu emócií a správania, je vhodné sledovať aj udalosti, ktorých sa pacient zúčastňuje. Bežať z jednej udalosti ku druhej a stíhať to načas, môže negatívne ovplyvniť hladinu stresu. Vďaka dostupným kalendárom v Android zariadení používateľa sa k nim môžeme dostať.

#### Návrh

Najskôr treba zistiť, či už sú kalendáre s udalosťami pre konkrétneho používateľa uložené na serveri. Potom vytvoríme službu, ktorá bude bežať na pozadí a sledovať všetky akcie používateľa ohľadom kalendára. To zahŕňa pridávanie, odstraňovanie udalostí alebo aktualizáciu jednotlivých udalostí. Každá akcia bude zaznamenaná na serveri.

## Implementácia

Pre získanie dát ohľadom udalostí použijeme Android API rozhranie kalendára `CalendarProvider`<sup>4</sup>. Pri spustení aplikácie jednoduchou požiadavkou skontrolujeme, či už sú kalendáre pre daného používateľa uložené na serveri. Ak nie, tak ich so všetkými udalosťami pošleme na server, všetko v jednej JSON<sup>5</sup> požiadavke. Potom implementujeme službu `CalendarService`, ktorá bude bežať na pozadí a kontrolovať akcie používateľa ohľadom kalendára a udalostí. Každá akcia sa pošle cez službu `DataSenderService` v formáte JSON na server.

## Testovanie

Modul sme otestovali tak, že skontrolujeme či prvom pustený pošle všetky dáta z kalendára.

---

<sup>4</sup><https://developer.android.com/guide/topics/providers/calendar-provider.html>

<sup>5</sup><https://github.com/Crakeh/eMotion-Web-App/wiki/API-Sensors>



Potom sme skúsili pridať, upraviť alebo aj odstrániť niektoré udalosti a skontrolovali, či sa aj tieto zmeny na serveri zaznamenali.

#### 4.1.7 WiFi modul

##### Analýza

V dnešnej dobe takmer každý jeden predaný Android má v sebe WiFi. Cieľom využitia wifi modulu v našej aplikácii je, aby sme vedeli zistiť, kde sa používateľ nachádza (domov, práca, kaviareň, ...) – aké názvy wifi sietí sa nachádzajú v jeho okolí. Android má prácu s wifi zdokumentovanú na veľmi dobrej úrovni a taktiež sa s ním ľahko robí. Prácu s wifi zabezpečuje WifiManager.

##### Návrh

Pre získanie dát z wifi použijeme WifiManager. V Androide na pozadí spustíme nami implementovaný Servis, ktorý bude spúšťaný AlarmManagerom. Dôvodom použitia AlarmManagera je to, aby Servis nebežal na pozadí stále, ale iba vtedy keď je to potrebné. AlarmManager je zodpovedný za to, aby každých 5 minút spustil Servis na detekciu wifi v okolí. Následne, keď zistíme počet a názvy wifi sietí v okolí tento servis vypneme. Týmto sa snažíme šetriť batériu mobilného telefónu. Ak spustíme detekciu wifi v okolí pomocou nášho Servisu a zistíme, že wifi je vypnutá, wifi zapneme. Samotný proces zisťovania wifi v okolí trvá pár sekúnd a po vykonaní skenovania okolia wifi zase vypneme.

##### Implementácia

Skenovanie wifi spustíme v Servise pomocou WifiManagera, kde najskôr skontrolujeme, či je wifi zapnutá alebo nie. Ak wifi zapnutá nie je, zapneme ju pomocou funkcie *setWifiEnabled(True)*. Následne zaregistrujeme Receiver, ktorý je zodpovedný za prijatie dát z Wifi. V tomto Receivery spracujeme dáta, ktoré prišli vo forme Listu. Skenovanie okolitých wifi následne zavoláme pomocou metódy *startScan()*. Na konci, keď Receiver prijme dáta, ho musíme odregistrovať a skontrolovať, či náš Servis spustil wifi alebo nie. Ak áno, tak musíme wifi vypnúť pomocou vyššie spomenutej funkcie *setWifiEnabled(False)*.

##### Testovanie

Funkčnosť modelu sme testovali v reálnych podmienkach. Po spustení aplikácie sme kontrolovali každých 5 minút, či zoznam obsahuje nové dáta zo senzorov.

## 4.1.8 GPS modul

### Analýza

Na zisťovanie polohy používateľ a je najvhodnejší spôsob používať GPS v mobilnom zariadení. Cieľom zisťovania polohy je určiť či konkrétne miesta vyvolávajú v používateľovi stres. Medzi takéto miesta môžu patriť napríklad preplnené ulice v starom meste, návšteva svokry alebo obchod s nahnevanou predavačkou. GPS patrí k prostriedkom Androidu, ktoré sa nedajú automaticky zapnúť, teda zapnutie GPS musí vykonať samotný používateľ. V Androide sa však nachádza viacero prostriedkov, vďaka ktorým sa dá zisťovať poloha. Jedným z takýchto spôsobov je napríklad internet.

### Návrh

Pre získanie dát o polohy používateľ a vytvoríme Service *MyLocationService*, ktorý sa bude spúšťať na pozadí pomocou *AlarmManager*a. Samotné vyhľadávanie bude implementované pomocou *FusedLocationApi*, ktorého cieľom je nájsť polohu používateľ a. Keď sa nám podarí nájsť používateľovu polohu, uložíme si jednotlivé súradnice – Latitude, Longitude – a čas. Na základe týchto dát budeme vedieť kde a kedy sa používateľ nachádza.

### Implementácia

Pri implementácii zisťovania polohy používateľ a treba na začiatku nastaviť konštanty o frekvenciách detekcií pohybu (ako často v milisekundách obnoviť polohu – zavolá sa metóda *onLocationChanged(Location location)*), koľko krát sa má poloha obnoviť (poloha, ktorá sa určí prvý krát nebýva veľmi presná). Na začatie vyhľadávania polohy sa musíme spojiť s *GoogleApiClient* ako môžeme vidieť na metóde *buildGoogleApiClient()* nižšie.

```
private synchronized void buildGoogleApiClient() {
    Log.i(TAG, "Building GoogleApiClient");
    mGoogleApiClient = new GoogleApiClient.Builder(this)
        .addConnectionCallbacks(this)
        .addOnConnectionFailedListener(this)
        .addApi(LocationServices.API)
        .build();
    createLocationRequest();
}
```

V metóde *createLocationRequest()* nastavíme globálnej premennej *mLocationRequest* typu *Loca-*

tionRequest interval ako často sa má poloha obnovovať (použijeme vyššie spomenutú konštantu) a taktiež s akou presnosťou má polohu vyhľadávať. V našom prípade sa jedná o typ *LocationRequest.PRIORITY\_BALANCED\_POWER\_ACCURACY*. Následne sa skontrolujú povolenie, ktoré by sa mali nachádzať v manifeste - či aplikácia má právo zisťovať polohu užívateľa. Tieto povolenia sú dvoch typov a to *Manifest.permission.ACCESS\_FINE\_LOCATION* a *Manifest.permission.ACCESS\_COARSE\_LOCATION*. Ak tieto dve povolenia sú povolené zavolá sa

```
LocationServices.FusedLocationApi.requestLocationUpdates(  
    mGoogleApiClient, mLocationRequest, this);
```

čím sa začne vykonávať vyhľadávanie polohy. Pri nájdení polohy sa zavolá vyššie spomenutá metóda *onLocationChanged(Location location)*, v ktorej už pracujeme s aktuálnou polohou používateľa.

## Testovanie

Funkčnosť modelu sme testovali v reálnych podmienkach. Po spustení aplikácie sme kontrolovali každých 5 minút, či zoznam obsahuje nové dáta zo senzorov.

## 4.1.9 SMS modul

### Analýza

SMSky sú jedným z prostriedkov, ktoré dokážu v človeku vyvolať stres. Naším cieľom bolo zachytiť číslo odosielateľa alebo adresáta a dĺžku tejto správy. Na základe týchto informácií sa budeme v budúcnosti snažiť predpovedať vyvolanie stresu aj pomocou SMS. V budúcnosti by sme taktiež chceli vypočítať sentiment správy. SMSky sa dajú v Androide zachytiť, a však len prichádzajúce. SMSky, ktoré používateľ odošle sa zachytiť pomocou nejakého Receiveru v Androide nedá. Práve preto na zachytenie odosielaných SMS bolo potrebné nájsť iný spôsob. Taktiež ak sa snažíme čítať v Androide SMS z Android databázy (miesto, kde sú uložené všetky správy), musíme mať na Android zariadeniach s API väčším alebo rovným ako API level 23 povolenie za behu systému. Pre zariadenia s nižším API level ako 23 stačí toto povolenie mať v Manifeste, kde bude automaticky povolené.

### Návrh

Pre získavanie prichádzajúci SMS v Androide, stačí nastaviť BroadcastReceiver so správnym

IntentFilterom. Pomocou IntentFiltera, dokážeme zachytiť prichádzajúce správy. Po príchode správy sa BroadcastReceiver automaticky zavolá a môžeme pracovať s SMS, ktorá bola doručená.

Odchádzajúce SMS sa však pomocou BroadcastReceiver zachytiť nedajú. Spôsob akým budeme zachytávať odchádzajúce SMS je taký, že pri prvom spustení si spočítame počet odoslaných správ. Následne pomocou AlarmManagera každých niekoľko minút budeme pomocou Service, ktorý spustí AlarmManager, kontrolovať počet odoslaných SMS. Ak sa počet odoslaných SMS zväčší, pozrieme sa na N nových odoslaných správ, ktoré následne spracujeme. Ak sa počet odoslaných zmenšil, tak si v pamäti budeme pamätať nový počet SMSiek.

### Implementácia

Na implementáciu prichádzajúcich SMS musíme v Manifeste nastaviť Receiver, ktorý bude zachytávať všetky udalosti, ktoré budú mať určitý Intent. Receiver nastavíme pomocou Intent filtra:

```
<intent-filter android:priority="999">
    <action android:name="android.provider.Telephony.SMS_RECEIVED"/>
</intent-filter>
```

V našom prípade musíme filtru nastaviť prioritu na 999, aby sa pri príchode SMS vykonal daný Receiver, nakoľko SMSky sú trochu náročnejšie na práva. Následne v android:name môžeme vidieť aký Intent sa snažíme zachytiť. Vďaka tomuto kroku sa nám zavolá náš BroadcastReceiver onReceive(). V tejto metóde už pracujeme s doručenou SMS - zistíme dĺžku správy, telefónne číslo a čas.

Pri implementácií odoslaných SMS treba pri prvom spustení vykonať inicializačné spočítanie odoslaných SMS. Ak používateľ nedá povolenie na prístup do databázy, kde sú uložené SMS, tak sa nastaví ich počet na nula. Následne sa pomocou AlarmManagera volá na pozadí každých N minút Service menom SmsOutgoingService, ktorého cieľom je spočítanie odoslaných SMS. Ak sa počet zmení, vložíme všetky odoslané SMS do ArrayListu. Tieto správy zoradíme podľa času a vyberieme aktuálny počet správ – počet uložených správ. Tieto správy následne znova spracujeme – zistíme dĺžku, komu boli odosielané a čas.

### Testovanie

Funkčnosť modelu sme testovali v reálnych podmienkach. SMS sme si posielali medzi zariadeniami a kontrolovali zoznam odoslaných a prijatých SMS v aplikácii.

## 4.1.10 Modul na detekciu hovorov

### Analýza

Telefónne hovory patria k činnostiam, ktoré nie každý človek zvláda bez stresu. Práve preto je naším cieľom detegovať prichádzajúce, odchádzajúce či zmeškané hovory. Konkrétne dáta, ktoré nás zaujímajú, sú číslo mobilu, trvanie hovoru a čas. Aj na základe týchto dát by sa dalo pracovať so stresom používateľa. V Androide sa dá použiť na detekciu prichádzajúceho ale aj odchádzajúceho hovoru `BroadcastReceiver`. Pri prichádzajúcom hovore dokonca vieme určiť kedy mobilný telefón začal zvonit', kedy bol hovor zdvihnutý a kedy bol hovor ukončený. Pri odchádzajúcom hovore vieme zistiť kedy začalo vytáčanie a kedy sa hovor ukončil, a však nevieme určiť či hovor bol zdvihnutý.

### Návrh

Pri detekcii odchádzajúcich hovorov musíme nastaviť nášmu Receiveru v Manifeste Intent Filter, na ktorý bude Receiver reagovať. Vďaka tomuto Receiveru budeme vedieť zachytiť odchádzajúce hovory. Každý Receiver môže obsahovať viacero filtrov, práve preto nastavíme tomu istému Receiveru aj filter pre zistenie stavu mobilu. Pri prichádzajúcom hovore sa totiž mení práve tento stav a vďaka nemu sa dá detegovať, kedy mobil začal zvonit', kedy bol hovor zdvihnutý a kedy ukončený.

### Implementácia

Implementácia hovorov prebiehala kvôli svojej zložitosti pomocou abstraktného Receiveru, ktorý detegoval odchádzajúce hovory a zmeny stavu mobilu. Pod zmenou stavu môžeme chápať zvonenie, prebiehanie hovoru a ukončenie hovoru. Tento Receiver obsahoval abstraktné metódy, ktoré bolo potrebné prekonať v triede, ktorá z tohto Receiveru dedila. Medzi tieto metódy patrili tieto metódy:

```
protected abstract void onIncomingCallReceived(Context ctx, String number, DateTime start);  
protected abstract void onIncomingCallAnswered(Context ctx, String number, DateTime start);  
protected abstract void onIncomingCallEnded(Context ctx, String number, DateTime start, DateTime end);  
protected abstract void onOutgoingCallStarted(Context ctx, String number, DateTime start);  
protected abstract void onOutgoingCallEnded(Context ctx, String number, DateTime start, DateTime end);  
protected abstract void onMissedCall(Context ctx, String number, DateTime start);
```

Následne ak sa zmenil stav v mobile tak sme takúto metódu zavolali v Receiveri v metóde `onCallStateChanged()` takýmto spôsobom:

```
switch (state) {  
    case TelephonyManager.CALL_STATE_RINGING:  
        isIncoming = true;  
        callStartTime = new DateTime();  
        savedNumber = number;  
        onIncomingCallReceived(context, number, callStartTime);  
        break;  
}
```

Vďaka tomuto sme v samostatnej triede, ktorú sme spomínali vyššie (trieda, ktorá dedí od abstraktného Receivera a prekonáva vyššie spomenuté metódy) vedeli pracovať so stavmi telefónu a taktiež pracovať so samotnými hovormi - odchádzajúcimi či prichádzajúcimi.

## Testovanie

Funkčnosť modelu sme testovali v reálnych podmienkach. Pomocou dvoch Android zariadení sme si navzájom telefonovali a následne kontrolovali v aplikácii, či sa podarilo tieto hovory zachytiť.

## 4.2 Web

### 4.2.1 Autentifikácia používateľa

#### Analýza

Nakoľko náš systém pozostáva z dvoch častí, z webovej aplikácie a mobilnej aplikácie, je potrebné zabezpečiť autentifikáciu na oboch stranách. Používatelia majú možnosť zaregistrovať sa na webovej stránke alebo v mobilnej aplikácii. V systéme sa nachádzajú tri typy používateľov - pacienti, lekári a administrátori. Úlohou administrátorov je registrovanie lekárov. Registrácia lekárov je realizovaná manuálne a je vykonávaná prostredníctvom Django admin panelu. Pacienti sa môžu v aplikácii registrovať svojpomocne.

#### Návrh

Do našej aplikácie sa pacienti môžu zaregistrovať pomocou webového rozhrania alebo pomocou

webovej služby. Na vytvorenie webovej registrácie sme použili django-allauth <sup>6</sup>. Táto knižnica ponúka základné funkcie potrebné pri autentifikácii používateľ a ako registrácia, prihlásenie, reset hesla, overenie e-mailovej adresy a pod. Taktiež je možné pomocou tejto knižnice vytvoriť vlastný model používateľ a. Pre registráciu používateľ a prostredníctvom mobilnej aplikácie sme vytvorili webovú službu. Na tento účel sme využili knižnicu django-rest-auth. Táto knižnica vie využiť náš model používateľ a a umožňuje jeho registráciu prostredníctvom REST API. Podobne ako django-allauth táto knižnica ponúka všetku potrebnú funkcionálnu k autentifikácii používateľ a.

### **Implementácia**

Pri registrácii či už prostredníctvom webového rozhrania, alebo mobilnej aplikácie, musí používateľ zadať email a heslo. Následne overíme či používateľ s takým e-mailom už náhodou neexistuje a taktiež či používateľ nezadal príliš jednoduché heslo. Ak je všetko v poriadku, odošleme používateľovi e-mail za účelom overenia jeho e-mailovej adresy. V tele e-mailu sa nachádza URL, pomocou ktorej používateľ potvrdí korektnosť jeho e-mailovej adresy a zároveň tým aktivuje svoj účet. Prihlásenie je možné až po úspešnej aktivácii. Po prihlásení má používateľ možnosť zmeniť svoj e-mail alebo heslo. Taktiež ak sa používateľ nevie prihlásiť z toho dôvodu, že zabudol svoje heslo, má možnosť vygenerovať si nové, ktoré mu bude zaslané na e-mailovú adresu použitú pri registrácii.

### **Testovanie**

Testovali sme registráciu používateľ a pri ktorej je potrebné zadať korektný formát emailovej adresy a taktiež dostatočne silné heslo. Následne sme otestovali odosielanie potvrdzovacieho emailu, aktiváciu účtu, prihlásenie, reset hesla a zmenu prihlasovacích údajov.

## **4.2.2 REST API na príjem dát**

### **Zvolenie vhodnej databázy**

Pri voľbe databázy nakoľko pracujeme s množstvom real-time dát bolo nevyhnutné zvážiť typ databázy, ktorý budeme používať. Demografické dáta o používateľoch ako aj prihlasovacie údaje ukladáme do relačnej databázy, konkrétne sme sa rozhodli pre Postgres. Pri výbere spôsobu ukladania dát zo senzorov a z mobilného zariadenia sme si preštudovali niekoľko zdrojov na StackOverflow ako aj odborných článkov, ktoré sa tejto tematike venujú.

Medzi zdroje podľa ktorých sme sa nakoniec rozhodli patria:

---

<sup>6</sup><http://django-allauth.readthedocs.io/en/latest/>

<http://dba.stackexchange.com/questions/13882/database-redesign-opportunity-what-table-design-to-use-for-this-sensor-data-col>

**VAN DER VEEN, Jan Sipke; VAN DER WAALJ, Bram; MEIJER, Robert J. Sensor data storage performance: Sql or nosql, physical or virtual. In: Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on. IEEE, 2012. p. 431-438.**

Z článku vyplýva, že Postgres je veľmi pomalý na zapisovanie (oproti NoSQL databázam ako sú Mongo alebo Casandra), a však s čítaním je to lepšie. Nakoniec sme sa rozhodli využiť ukladanie dát do relačnej databázy Postgres s tým, že pomalé zapisovanie riešime ukladaním do vyrovnávacej pamäti. Vyrovnávacia pamäť v podobe Redisu preklápa dáta z API, ktoré slúži na príjem dát, do samotného Postgresu. Do budúca v prípade príjmu veľkých dát je možnou alternatívou napr. Mongo s využitím Sparku na dopytovanie.

Čo sa týka samotného modelu zvažovali sme dva typy uloženia. Prvou možnosťou bolo vytvorenie samostatných tabuliek, pre každý jeden typ senzora. Toto uloženie by bolo prehládnejšie a uľahčilo by dopytovanie na konkrétny typ senzora. Jeho nevýhodou by bola náročnosť dopytovania sa na dáta zo všetkých sensorov za dané časové obdobie a taktiež spôsob pridávania nových sensorov. Nakoľko množina sensorov a dát, ktoré sa z mobilného zariadenia získavajú nemusí byť konečná, rozhodli sme sa uložiť všetky dáta v jednej tabuľke. V tabuľke je špecifikovaný typ senzora, alebo mobilného prostriedku (GPS, Wifi) a dáta zo zariadenia v podobe jsonu. Výhodou takéhoto uloženia je jednoduché pridávanie nových sensorov, nakoľko na strane servera to predstavuje pridanie jedného záznamu do databázy a nie je potrebná žiadna dodatočná implementácia (ak nerátame validáciu vstupu). Nevýhodou tohto uloženia je veľké množstvo záznamov, ktoré máme v jednej tabuľke.

### **Analýza**

Keďže náš systém pozostáva z dvoch nezávislých aplikácií (mobilná a webová), je potrebné medzi nimi zabezpečiť určitý druh komunikácie z dôvodu prenosu dát. Toto je potrebné najmä pre prenos biometrických dát z mobilného zariadenia na webový server, kde budú následne uložené do databázy. Kvôli čistote a konzistentnosti dát je potrebné vytvoriť určitý druh validácie. Komunikácia bude využívaná aj pri registrácii alebo prihlásení používateľ a do mobilnej aplikácie.

### **Návrh**

Mobilná aplikácia bude komunikovať s mobilným serverom prostredníctvom REST API. Na



tvorbu REST API použijeme Django REST framework <sup>7</sup>, ktorý je v spolupráci s Django často používaný. Navyše tento framework ponúka širokú škálu funkcionalít ako napríklad rôzne spôsoby implementácie samotnej API a taktiež viaceré možnosti autentifikácie. Na validáciu dát využijeme JSON schémy.

### **Implementácia**

Na implementáciu API sme zvolili tzv. “Class-based views”, kde každá URL je reprezentovaná jednou triedou na webovom servery. V rámci triedy sa ďalej nachádzajú metódy, ktoré predstavujú jednotlivé typy HTTP requestov. Priamo v danej triede je možné zvoliť či na prístup k danej URL je potrebná autentifikácia alebo nie. V našom prípade je potrebná autentifikácia vždy, keďže ide o dáta samotných používateľov. Zvolili sme metódu autentifikácie pomocou tokenov. Po zaregistrovaní je každému používateľovi vygenerovaný token, pomocou ktorého sa môže autentifikovať. Aktuálne token nemá žiadnu expiráciu, čo do budúcnosti zvažujeme zmeniť. Hlavnou súčasťou API je posielanie biometrických dát a dát o používateľovi, ktoré sú hneď ako dorazia na webový server validované. Ak validácia prebehne úspešne dáta sú uložené do Redisu a následne postupne ukladané do relačnej databázy.

### **Testovanie**

Testovali sme prijímanie dát z jednotlivých senzorov, validáciu dát, ukladanie dát do Redisu a do relačnej databázy PostgreSQL.

## **4.2.3 Dashboard s dátami používateľa**

### **Analýza**

Hlavným účelom nášho systému je detekcia negatívnych emócií používateľa s cieľom znížiť negatívny vplyv týchto emócií na zdravie človeka. Nevyhnutnou súčasťou takejto aplikácie je vizualizácia dát, na základe ktorých sú detekované emócie. V našej aplikácii máme dvoch typov používateľov - pacientov a lekárov. Tieto dáta sú potrebné pre obidva typy používateľov. Pacient by mal mať prístup k svojim dátam, aby si mohol prezrieť svoje historické dáta a videl ako sa zlepšuje, alebo naopak zhoršuje. Taktiež lekár by mal mať prístup k sumarizovaným dátam o jednotlivých pacientoch, ktoré mu môžu pomôcť k lepšiemu smerovaniu liečby.

### **Návrh**

Webová aplikácia bude poskytovať dva pohľady na dáta - pohľad pre lekára a pohľad pre pacienta. Pacientovi sa budú zobrazovať biometrické dáta v podobe prehľadných grafov. Zároveň

---

<sup>7</sup><http://django-allauth.readthedocs.io/en/latest/>

bude mať pacient zobrazený graf o vývoji jeho negatívnych emócií, ktoré sú vypočítané na základe dát z mobilného zariadenia. Tieto dáta (wifi, sms, hovory a pod.) však pacientovi priamo zobrazované nebudú. Lekár má k dispozícii vývoj negatívnych emócií jednotlivých pacientov sumarizovane v maximálne dvoch grafoch na každého pacienta, tak aby týmito dátami nebol zahltený.

### **Implementácia**

Z tohto modulu je aktuálne naimplementované kumulatívne zobrazovanie dát o jednotlivých používateľoch v tabuľke. To znamená, že každý používateľ si vie prezrieť všetky svoje záznamy z mobilného zariadenia alebo biometrických senzorov vo svojom účte v podobe tabuľky. Implementácia grafov zatiaľ nie je realizovaná.

### **Testovanie**

Testovali sme správne zobrazovanie dát jednotlivých používateľov.

## **4.2.4 Modul sentimentu**

### **Analýza**

Keďže cieľom našej práce je rozpoznávanie emócií a na emócie má veľký vplyv obsah komunikácie, rozhodli sme sa vyhodnocovať sentiment SMS správ z mobilného zariadenia. Na základe sentimentu dokážeme zistiť, či je obsah správ pozitívny alebo negatívny, čo nám pomôže lepšie pochopiť kontext nálady používateľa. V rámci analýzy sme zvážili niekoľko možností. Existuje množstvo nástrojov pre vyhodnocovanie anglického textu, a však nakoľko je naša aplikácia určená primárne pre slovenský jazyk, tieto nástroje nemôžeme využiť. Z tohto dôvodu sme naviazali spoluprácu s jedným z našich spolužiakov, ktorý v rámci bakalárskej práce vytvoril vyhodnocovač sentimentu pre slovenský jazyk.

### **Návrh**

Po dohode s Rast'om Krchňavým sme spoločne navrhli API na vyhodnocovanie sentimentu, ktoré vyhovuje obom stranám. Naš aktuálny návrh zahŕňa využívanie Rast'ovho API, ktoré je natrénované na facebookovských príspevkoch a dáva pomerne presné výsledky a naimplementovanie vlastného vyhodnocovania sentimentu na základe dostupného slovníka.

### **Implementácia**

V súčasnosti modul zahŕňa pravidelné volanie API po príchode údajov z mobilnej aplikácie. Keďže text SMS-iek je citlivá informácia, tento text je zašifrovaný šifrovaním AES. Po prijatí

je event s SMS-kou uložený do Redisu a následne je predspracovaný. Predspracovanie zahŕňa odšifrovanie správy, zavolanie API na určenie sentimentu, doplnenie dát o sentimente do tela eventu a následné uloženie do Postgresu. Nakoľko API nemusí byť vždy dostupné, SMS-ky ktoré neboli spracované, sú priebežne monitorované a spracovávané. Tento update je realizovaný každú hodinu v podobe cron jobu, ktorý nájde všetky SMS-ky, ktoré nemajú určený sentiment, následne tieto SMS-ky rozdelí do skupín, odšifruje a odošle na spracovanie. Posledným krokom je uloženie dodatočnej informácie do databázy. Uloženie je realizované funkciou *bulk\_update()*, ktorá umožňuje aktualizovať N objektov v rámci jednej query, čo je oveľa efektívnejšie než jednoduché *update()*.

### **Testovanie**

Modul bol testovaný simulovaním príjmu dát z mobilného zariadenia. Pri testovaní sme sa zamerali predovšetkým na funkčnosť a efektívnosť. Cron job bol testovaný prostredníctvom generovaných dát a manuálnym spúšťaním skriptu.