LUCRARE DE DISERTAȚIE

Îndrumător proiect,
Diana ȘTEFĂNESCU

Absolvent,
Andrei LUCA

Galaţi
2017

Activity Tracker – pedometru Arduino

Îndrumător proiect,
Diana ȘTEFĂNESCU

Absolvent,
Andrei LUCA

Galaţi
2017

REZUMAT

Obiectivul lucrării-"Activity Tracker – pedometru Arduino"-este realizarea unui dispozitiv de tip pedometru pentru monitorizarea activităţii personale prin calcularea numărului de paşi efectuaţi de purtător şi generarea statisticilor cu privire la stilul de viaţă al persoanei.

Dispozitivul va simţi modificarea poziţiei prin datele primite de la accelerometru şi le va transmite prin modulul bluetooth către un dispozitiv conectat. Procesarea acestor date şi generarea statisticilor şi a graficelor vor avea loc pe un server web întrucât acesta are o mai mare putere de calcul.

Plăcuţa de dezvoltare ce efectuează calculele necesare, Arduino Pro Mini, reprezintă punctul de pornire în dezvoltarea acestei lucrări. La acest dispozitiv sunt conectate un accelerometru ADXL 345 şi un modul bluetooth HC-06. Accelerometrul notifică starea sa actuală în valori ale coordonatelor X, Y, Z. Aceste date sunt citite de către Arduino, care prin intermediul unui algoritm determină dacă a fost efectuat un pas. Dacă la acest dispozitiv este conectat un dispozitiv cu sistem de operare Android, datele vor putea fi transmise în timp real pentru a putea fi vizualizate de utilizator.

Dispozitivul Android fiind conectat la Internet, va putea transmite datele către un server Web care va afişa graficele corespunzătoare activităţii fizice.

Aplicaţia web este dezvoltată folosind cele mai noi tehnologii în materie de web development ce reduc timpul de încărcare şi consumul de date mobile ale utilizatorului. De asemenea, comunicarea în timp real dintre serverul Web şi dispozivitul Android se face prin protocolul DDP.

Astfel au fost abordate următoarele probleme :

- Preluarea informaţiilor de la accelerometru
- Determinarea unei mişcări a utilizatorului
- Comunicarea între pedometru şi un dispozitiv cu sistem de operare Android prin intermediul aplicaţiei dezvoltate special în acest sens
- Comunicarea între dispozitivul Android şi un server web printr-un protocol DDP pentru ca apoi să se poată vizualiza un istoric al informaţiilor
- Implementarea unui site Web securizat pentru monitorizarea în timp real a activităţii utilizatorilor din partea unui medic
- Implementarea unei aplicaţii Android securizate pentru monitorizarea în timp real a numărului de paşi efectuaţi

Toate soluţiile propuse au fost implementate şi testate, folosind exclusiv dipozitivele embedded, aplicaţia Web şi dispozitivul mobil.

CUPRINS

INTRODUCERE

În societatea modernă a zilelor noastre în care oamenii trăiesc vieţi sedentare datorită locurilor de muncă la birou, a traficului aglomerat ce măreşte timpul de transport între casă şi locul de muncă şi a faptului că ne-am obişnuit să avem parte de servicii 24/7 direct la uşa noastră, nu mai suntem la fel de activi din punct de vedere fizic.

Potrivit ultimelor studii în acest domeniu, s-a demonstrat că simpla activitate de a merge scade riscurile unui infarct, boli de inimă, diabet şi chiar a depresiei. Un studiu efectuat de American Heart Association din 2010 a demonstrat că femeile care merg cel puţin două ore pe săptămână au cu 30% mai puţine şanse să facă un infarct.

În anul 2007, secţia de medicină a bine-cunoscutei universităţi Stanford a efectuat un test prin care demonstra că pedometrele, aceste dispozitive care măsoară numărul de paşi şi efectuează statistici cu privire la distanţa parcursă şi a numărului de calorii consumate, sunt foarte eficiente în a motiva oamenii să alerge sau să meargă mult mai des. Studiul a demonstrat că cei ce foloseau pedometre au început să efectueze cu aproximativ 2100 mai mulţi paşi decât cei ce nu purtau.

Acelaşi studiu al Universitătii Stanford a arătat că oamenii au fost mult mai motivaţi atunci când şi-au propus să atingă anumite ţinte şi când şi-au înregistrat activitatea în fiecare zi. Cu alte cuvinte, utilizatorii au fost inspiraţi de nişte ţinte concrete, măsurabile şi realiste.

Desigur, atunci când apar probleme de sănătate datorate sedentarismului, rezultate mult mai bune apar atunci când activitatea fizică este însoţită de o dietă adaptată stilului de viaţă, a greutăţii corporale şi a capacităţii fizice a utilizatorului.

Aplicaţia prezentată în această lucrare îşi propune să fie o interfaţă între o echipă formată din medici de specialitate, antrenori de fitness, nutriţionişti şi utilizatorul de rând. Astfel, utilizatorul va avea parte de rezultate cât mai bune într-un timp cât mai scurt fără ca antrenamentul fizic să fie o povară pentru el. Ţintele menţionate anterior vor fi adaptate fiecărei persoane în parte.

Astfel problemele ce au fost rezolvate cu ajutorul acestei aplicaţii sunt :

- Preluarea informaţiilor de la utilizator în ceea ce priveşte vârsta, greutatea curentă, şi informaţii personale
- Preluarea informaţiilor de la accelerometru
- Comunicarea între pedometru şi dispozitivul Android
- Comunicarea între dispozitivul Android şi serverul web
- Determinarea efectuării unei mişcări prin interpretarea rezultatelor primite de la accelerometru printr-un algoritm
- Implementarea unei aplicaţii web securizate pentru monitorizarea în timp real a activităţii utilizatorilor din partea unui doctor

- Implementarea unei aplicaţii Android securizate pentru monitorizarea în timp real a numărului de paşi efectuaţi
- Generarea graficelor şi rapoartelor în aplicaţia web

Utilizatorul va putea vizualiza datele înregistrate de către dispozitiv în timp real şi în funcţie de rezultatele obţinute pe o perioadă de timp, ţintele stabilite de către medic se vor adapta corespunzător.

Atât aplicaţia Android cât şi aplicaţia web vor prezenta grafice ale activităţii pe o anumită perioadă de timp.

În plus, aplicaţia Android va prezenta informaţii cu privire la cantitatea de apă ce trebuie consumată de utilizator potrivit cu greutatea şi efortul efectuat, alimentele cele mai benefice care pot fi consumate, numărul de calorii ce trebuie consumate pe zi. Pe langă acestea, el va primi notificări motivaţionale în timp real.

Dincolo de introducerea de faţă, lucrarea este structurată pe trei capitole:

- Primul capitol prezintă cerinţele şi specificaţiile aplicaţiei la nivel funcţional şi tehnic;
- Capitolul doi tratează proiectarea sistemului la nivelul dispozitivelor embedded, bazei de date şi a aplicaţiilor, comunicarea şi deasemeni şirul evenimentelor la nivel de principiu;
- Capitolul trei urmăreşte modul în care au fost implementate aplicaţiile la nivel tehnic

CAPITOLUL 1. CERINȚE ȘI SPECIFICAȚII

1.1 STADIUL ACTUAL AL PEDOMETRELOR

Pedometrul este soluţia ideală pentru menţinerea motivaţiei pe termen lung pentru a duce un stil de viaţă activ. Principiul de funcţionare al unui pedometru este simplu : un accelerometru înregistrează acceleraţia pe una din cele trei axe X, Y, Z. Plecând de la aceste valori, în baza unui algoritm se detectează o anumită mişcare care va fi interpretată ca fiind un pas. Există mai multe tipuri de pedometre : cu display sau fără display, cu algoritm bazat pe o valoare de referinţă sau cu algoritm bazat pe istoricul mişcărilor utilizatorului.

Desigur, pedometrele cu display vor consuma mai multă baterie şi vor necesita schimbarea bateriei acesteia într-un termen mai scurt şi merg pe acelaşi principiu ca cele fără display - de a transmite datele către un alt dispozitiv unde datele sunt interpretate.

Pedometrele ce au la bază algoritmul bazat pe stilul de mers al utilizatorului sunt mai eficiente şi au avantajul de a fi personalizate pentru fiecare purtător.

Există şi pedometre ce conţin un modul WiFi şi se pot conecta direct la o reţea wireless pentru a transmite datele către un server web, dar desigur acestea sunt mult mai scumpe.

În această categorie intră şi pedometrul Fitbit One. Acesta înregistrează numărul de paşi şi monitorizează timpul de inactivitate (somn) şi la fel ca toate celelalte pedometre inteligente, sincronizează datele activităţii cu un telefon mobil, o tabletă sau un computer. Costul foarte ridicat reprezintă desigur unul din dezavantajele acestui dispozitiv.

Nike FuelBand SE poate reprezenta o altă soluţie în ceea ce priveşte achiziţionarea unui pedometru. Deşi producţia a fost suspendată de către productor, acest model încă se poate achiziţiona în magazine de specialitate. Avantajul acestui pedometru este reprezentat de faptul că este rezistent la apă şi monitorizează şi timpul de inactivitate. Dezavantajul principal este reprezentat de faptul că acest dispozitiv nu este compatibil cu dispozitivele Android, un sistem de operare răspândit în toată lumea.

1.2 CERINȚELE SISTEMULUI

Primii paşi în dezvoltarea unui „activity tracker” au presupus identificarea şi înţelegerea necesităţilor utilizatorilor prin definirea cerinţelor sistemului. În urma studierii conceptelor ce stau la baza unui pedometru conectat cu o aplicaţie Android şi un server Web, au fost definite următoarele cerinţe :

- Realizarea montajului dispozitivului, al accelerometrului şi al modulului Bluetooth
- Citirea datelor iniţiale ale accelerometrului pentru stabilirea axei de direcţie prin determinarea poziţiei accelerometrului
- Citirea informaţiilor de la accelerometru de către placa de dezvoltare Arduino Pro Mini
- Interpretarea rezultatelor provenite de la accelerometru
- Compararea rezultatelor provenite de la accelerometru cu o valoare de referinţă pentru a determina dacă a fost efectuat un pas
- Citirea datelor provenite de la dispozitiv de către aplicaţia Android
- Proiectarea design-ului interfeţei grafice pentru aplicaţia Android şi aplicaţia Web pentru un acces facil la informaţiile importante
- Proiectarea bazei de date de tip NoSQL pe server-ul Web şi identificarea datelor ce vor fi transmise de către aplicaţia Android către aplicaţia Web

Proiectarea aplicaţiei Android

- Implementarea secţiunii de autentificare şi conectare la server
- Implementarea secţiunii de afişare a numărului de paşi efectuaţi până la momentul curent de către utilizator
- Implementarea secţiunii de afişare a sfaturilor şi dietei stabilite de către medicul supervizor
- Implementarea secţiunii de afişare a greutăţii curente a utilizatorului prin raportare la greutatea optimă stabilită de medic în funcţie de înălţime şi vârstă
- Implementarea secţiunii de setare a atributelor ce nu pot fi stabilite în mod automat : înălţime, greutate, vârstă

Proiectarea aplicaţiei Web

- Implementarea secţiunii de autentificare – poate fi făcută în mod utilizator sau mod administrator
- Transmiterea datelor de la aplicaţia Android către aplicaţia Web prin protocolul DDP (Distributed Data Protocol) în timp real
- Interpretarea datelor unui utilizator şi afişarea acestora într-un grafic în aplicaţia Web
- Interpretarea datelor provenite de la toţi utilizatorii şi afişarea în grafice a statisticilor în panoul administratorului

1.3 SPECIFICAŢII

Proiectul îşi propune să realizeze o îmbunătăţire a modului de transmitere şi afişare a datelor provenite de la un dispozitiv embedded într-o aplicaţie Web.

Mediul de programare ales a fost sistemul de operare Ubuntu 14.04 LTE, editorul Sublime Text 3 pentru implementarea aplicaţiei Web, tehnologii precum MeteorJS, ReactJS, DDP, HTML, CSS şi JavaScript.

Pentru realizarea dispozitivului au fost folosite :

- O placă de dezvoltare Arduino Pro Mini 5V / 16 Mhz
- Un accelerometru ADXL 345
- Un modul Bluetooth HC-06

Pentru proiectarea schemei dispozitivelor s-a folosit utilitarul Fritzing.

Pentru dezvoltarea programului pentru placa Arduino a fost folosit mediul de dezvoltare Arduino IDE.

Pentru realizarea aplicaţiei Android a fost folosit software-ul Android Studio.

Pentru realizarea aplicaţiei Web au fost folosite următoarele medii de dezvoltare şi tehnologii :

- Framework-ul web MeteorJS
- Baza de date MongoDB
- Librăria ReactJS dezvoltată de Facebook pentru a creea interfaţa utilizator

Serverul pentru baza de date MongoDB este pornit odată cu pornirea aplicaţiei. În folder-ul aplicaţiei se rulează comanda „meteor” care va porni atât serverul web cât şi serverul bazei de date iar apoi aplicaţia Web va putea fi accesată prin intermediul browser-ului la adresa localhost pe portul 3000 : *localhost:3000*.

CAPITOLUL 2. PROIECTAREA SISTEMULUI

Componentele hardware ale sistemului sunt:

- Placa de dezvoltare Arduino Pro Mini 5V / 16Mhz
- Un accelerometru model ADXL 345 compatibil cu Arduino
- Un modul Bluetooth model HC-06 compatibil cu Arduino

Componentele software ale sistemului sunt:

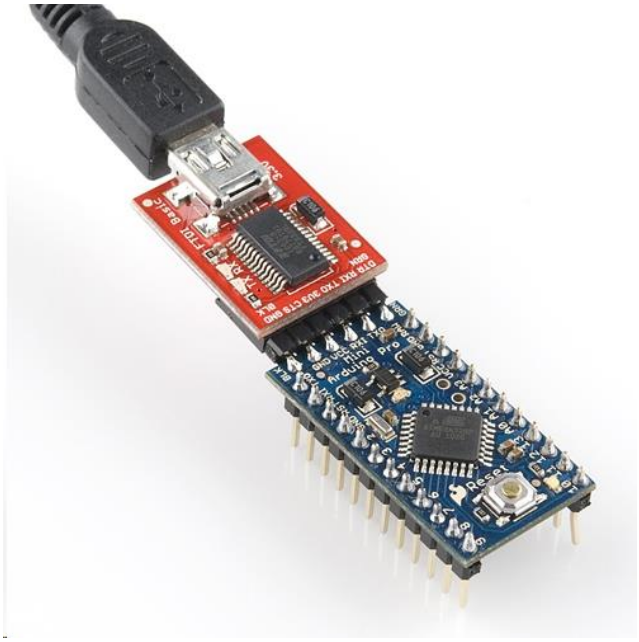
- Aplicaţia Web care conţine şi o bază de date
- Aplicaţia Android

Dispozitivul Arduino Pro Mini este programat cu ajutorul mediului de dezvoltare Arduino IDE. Algoritmul este mai întâi compilat şi apoi încărcat pe placa de dezvoltare unde va rula până la întreruperea alimentării. Modulele conectate vor comunica cu placa principală prin intermediul interfeţelor expuse de Arduino Pro Mini (I2C).

2.1 DESCRIEREA DISPOZITIVULUI EMBEDDED

2.1.1 DESCRIEREA PLĂCII DE DEZVOLTARE ARDUINO PRO MINI

Arduino Pro Mini este o placă de dezvoltare bazată pe microcontroller-ul Atmega328, ce funcţionează la o tensiune de 5 V şi la o viteză de 16MHz. Dispozitivul este de obicei folosit ca soluţie semi-permanentă în diverse obiecte sau proiecte demonstrative.



Arduino Pro Mini 5V conectat la dispozitivul FT232-RL

Caracteristici

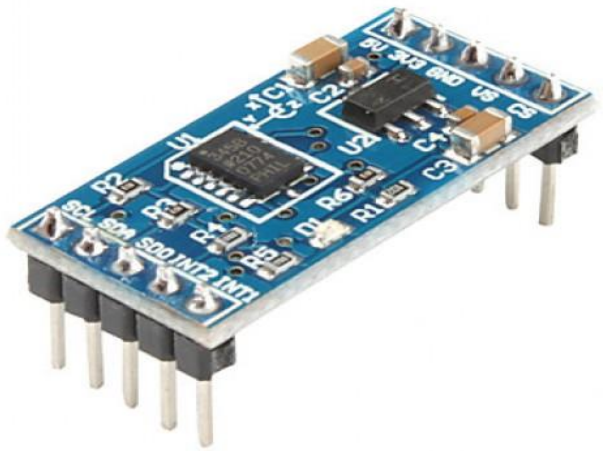
Microcontroller	ATmega328
Board Power Supply	5 - 12 V
Circuit Operating Voltage	5V
Digital I/O Pins	14
PWM Pins	6
SPI	1
I2C	1
Analog Input Pins	6
External Interrupts	2
DC Current per I/O Pin	40 mA
Flash Memory	32KB
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

Caracteristicile plăcii de dezvoltare Arduino Pro Mini

2.1.2 DESCRIEREA ACCELEROMETRULUI

ADXL345 este un accelerometru în 3 axe ce poate fi accesat prin interfaţa sa SPI sau I2C. Este potrivit pentru aplicaţiile în dispozitivile mobile pentru că măsoară acceleraţia statică a gravitaţiei precum şi acceleraţia dinamică rezultată în urma mişcării sau a producerii unui şoc.

Acesta are mai multe funcţii speciale printre care : detectarea activităţii / inactivităţii, dacă acceleraţia pe o anumită axă depăşeşte un anumit nivel prestabilit sau detectarea situaţiei în care dispozitivul se află în cădere liberă.



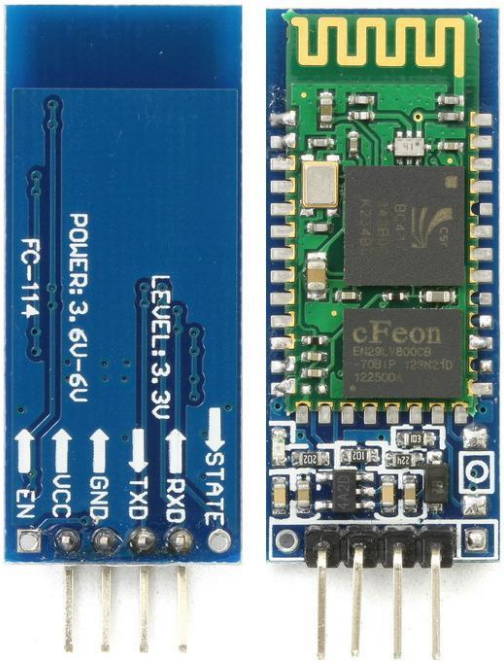
Accelerometrul ADXL-345

2.1.3 DESCRIEREA MODULULUI BLUETOOTH

Modulul bluetooth permite transmiterea datelor între dispozitivul pedometru şi aplicaţia Android prin transceiver-ul wireless digital de 2,4 Ghz. Înainte de a folosi acest dispozitiv este necesară procedura de „pairing”. Având în vedere că acest modul poate acţiona doar în starea de slave, conexiunea trebuie să fie iniţiată de clientul Android.

Dispozitivul Android trebuie să aibă şi el un modul Bluetooth încorporat la rândul său. Prin descoperirea dispozitivelor Bluetooth din proximitate, se selectează modulul HC-06 şi se va introduce parola dispozitivului.

Modulul Bluetooth este prevăzut cu un LED care indică starea conexiunii. Dacă starea LED-ului rămâne constant pe modul „aprins” înseamnă că procedura de „pairing” a fost efectuată cu succes şi se poate începe transmiterea datelor atât dinspre dispozitivul pedometru către dispozitivul Android, cât şi invers.



Modul Bluetooth HC-06

2.2 PROIECTAREA DISPOZITIVELOR EMBEDDED

Dispozitivul pedometru va fi format din placa de dezvoltare Arduino Pro Mini, accelerometrul ADXL345 şi modulul Bluetooth HC-06. Având în vedere că toate aceste dispozitive funcţionează la aceeaşi tensiune, nu vor mai fi necesare alte elemente adiţionale.

Accelerometrul va fi conectat la placa de dezvoltare prin intermediul interfeţei digitale I2C astfel :

- Pinul SDA va fi conectat la pinul A4 al plăcii Arduino
- Pinul SCL va fi conectat la pinul A5

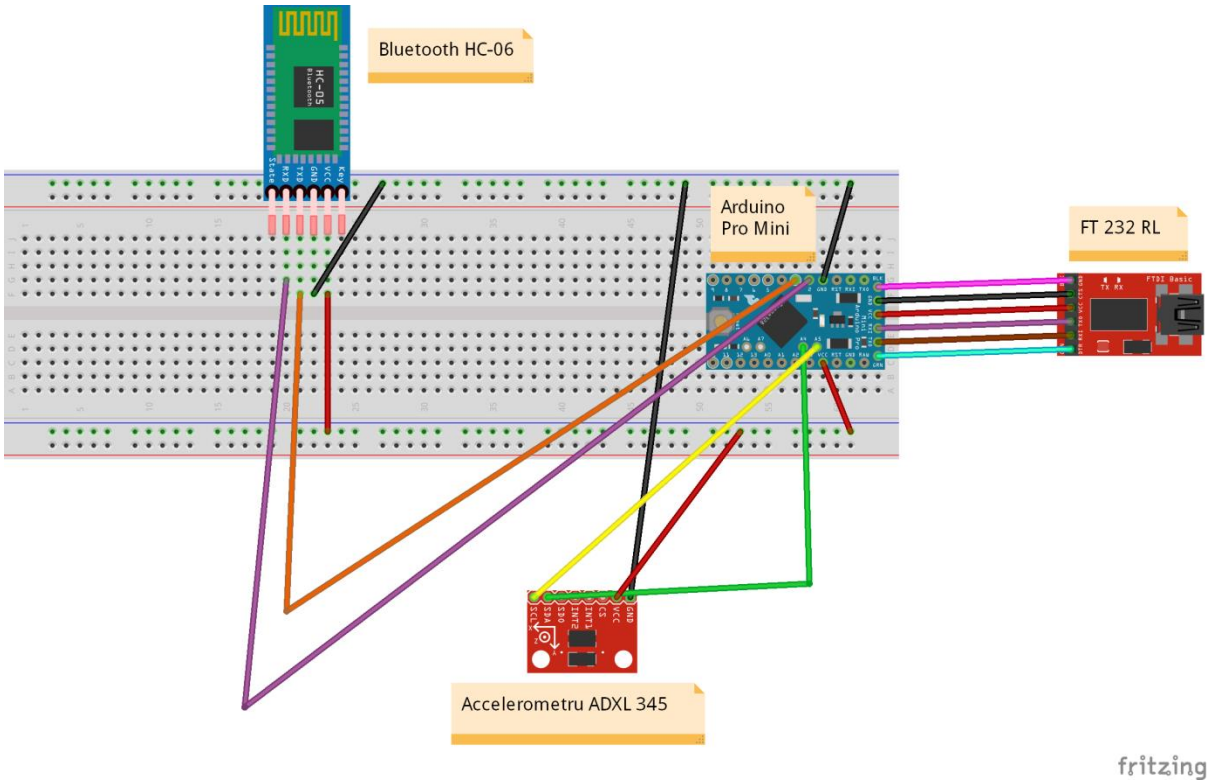
- Pinul GND (Ground) va fi conectat la pinul de masă GND
- Pinul VCC va fi conectat la pin-ul de alimentare VCC al plăcii Arduino

Modulul BT HC-06 va fi conectat astfel :

- Pinul RX (receiver) va fi conectat la pinul digital D2
- Pinul TX (transmitter) va fi conectat la pinul digital D3
- Pinul GND va fi conectat la pinul de masă GND
- Pinul VCC va fi conectat la pinul de alimentare VCC al plăcii Arduino

Pentru programarea şi alimentarea plăcii de dezvoltare Arduino Pro Mini a fost folosit un dispozitiv FT232-RL conectat astfel :

- GRN – GRN
- TXO – RXI
- RXI – TXO
- VCC – VCC
- GND – GND
- BLK – BLK



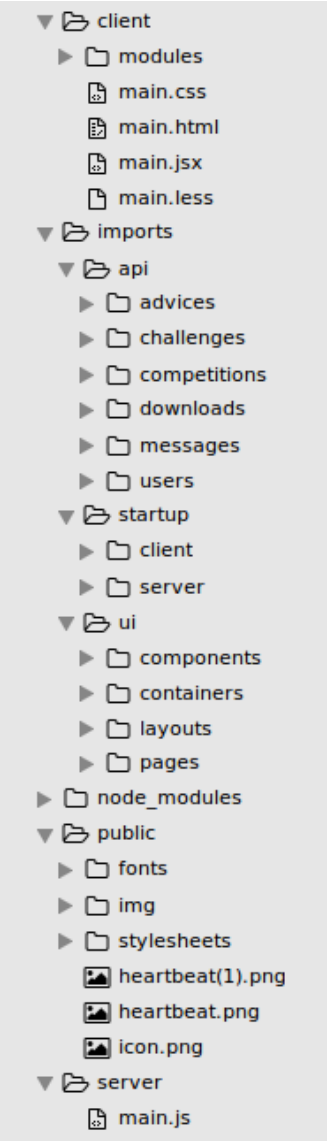
Proiectarea dispozitivului pedometru

2.3 PROIECTAREA APLICAȚIEI WEB

2.3.1 ARHITECTURA APLICAȚIEI WEB

Aplicaţia Web este bazată pe framework-ul full-stack MeteorJS şi diferă de majoritatea aplicaţiilor web prin faptul că o parte din cod rulează doar pe client, o altă parte pe server, dar are şi cod comun ce rulează atât pe client cât şi pe server.

Pentru a beneficia de acest avantaj al framework-ului trebuie ca structura folderelor şi fişierelor să fie bine definită, în folderele corespunzătoare „client” şi „server”



Structura fişierelor

Sistemul modular ne permite să importăm diverse fişiere doar atunci când avem nevoie de ele. Sistemul de compilare al framework-ului Meteor va include doar fişierele care sunt referenţiate în alte fişiere prin intermediul comenzii „*import <nume_fişier>*”.

Meteor foloseşte tehnica „date pe fir”, iar prin aceasta înţelegem faptul că serverul trimite date, nu fişiere HTML, iar clientul este cel ce se ocupă de partea de randare.

Pentru o performanţă crescută, a fost folosită librăria dezvoltată de Facebook denumită ReactJS pentru randarea datelor în browser.

Într-o aplicaţie HTTP tradiţională, clientul şi serverul comunică prin procedeul de *request-response*. Clientul trimite anumite cereri către server şi primeşte în schimb fişiere HTML sau obiecte JSON. Limitarea este dată de faptul că serverul nu are posibilitatea de a trimite date către client în cazul în care se modifică documentele bazei de date.

Pentru a îmbunătăţi performanţele, s-a folosit protocolul DDP – Distributed Data Protocol – ce permite transmiterea datelor de la client către server şi invers. Asta înseamnă că nu va trebui să creăm endpoint-uri pentru a fi accesate şi pentru a obţine informaţii ci datele vor fi oferite sub formă de publicaţii. Clientul iniţiază o subscripţie la o publicaţie şi primeşte datele.

Clientul nu are o conexiune directă la baza de date pentru a manipula datele, atât din motive de securitate cât şi din motive ale performanţei. O colecţie în cadrul clientului este de fapt un cache al bazei de date de pe server. De exemplu, în momentul în care vrem să introducem anumite date în baza de date, acest lucru se va realiza cu ajutorul unor metode existente pe server.

Metodele oferă o alternativă mult mai eficientă faţă de clasicele rute endpoint din arhitectura REST. Ele oferă toate avantajele unor endpoint-uri asincrone din NodeJS dar folosesc stilul sincron al API-ului. Pe langă asta, având în vedere că se foloseşte tehnica Websockets în loc de HTTP, avem garanţia că toate apelurile şi rezultatele metodelor vor veni în ordinea aşteptată.

2.3.2 ŞIRUL EVENIMENTELOR APLICAȚIEI WEB

Având în vedere tematica lucrării, s-a luat în considerare crearea ecranelor diferite în mod utilizator şi în mod administrator. Utilizatorul nu are dreptul de a edita sau a crea conţinut, spre deosebire de administrator.

În modul de administrare, aplicaţia Web urmăreşte să conţină următorul şir de evenimente :

- Autentificarea administratorului prin ecranul de conectare
- Înregistrarea prin subscripţie la documentele publicate ale bazei de date
- Vizualizarea statisticilor cu privire la numărul de utilizatori ce deţin un cont şi activitatea fizică înregistrată în ultimele 24 ore
- Vizualizarea detaliată a informaţiilor utilizatorilor
- Vizualizarea în timp real a numărului de paşi parcurşi pentru un utilizator selectat
- Crearea, editarea şi vizualizarea competiţiilor la care un utilizator se poate înregistra prin stabilirea traseului şi a numărului maxim de participanţi

- Crearea, editarea şi vizualizarea provocărilor la care un utilizator se poate înregistra
- Transmiterea mesajelor private către un utilizator

În modul utilizator, aplicaţia Web urmăreşte să ofere următoarele posibilităţi :

- Autentificarea prin ecranul de înregistrare sau ecranul de conectare
- Înregistrarea prin subscripţie la documentele publicate ale bazei de date în funcţie de permisiunile utilizatorului
- Vizualizarea statisticilor cu privire la numărul de paşi efectuaţi în ziua curentă prin raportare la ţinta stabilită de către medicul supervizor.
- Vizualizarea şi înregistrarea la competiţiile create
- Vizualizarea şi înregistrarea la provocările create

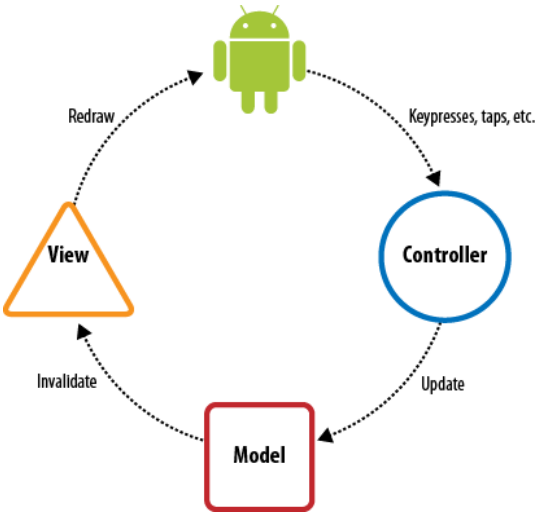
2.4 PROIECTAREA APLICAȚIEI ANDROID

2.4.1 ARHITECTURA APLICAȚIEI ANDROID

Aplicaţia Android foloseşte un pattern MVC (Model – View Controller) pentru implementarea interfeţei grafice. Acest tipar împarte clientul în trei module interconectate cu scopul de a separa reprezentarea internă a informaţiei de modul în care această informaţie este prezentată utilizatorului. Beneficiul adus este dat de posibilitatea reutilizării codului şi posibilitatea de a dezvolta codul pentru fiecare cele trei module în paralel, fără o interdependenţă.

Clientul Android conţine atât informaţii proprii, cât şi informaţii ce sunt stocate în baza de date a server-ului Web.

Atât interacţiunea dintre clientul Web şi server-ul Web, cât şi interacţiunea dintre clientul Android şi server-ul Web se bazează pe protocolul DDP. În scopul de a transmite datele într-un mod cât mai rapid şi eficient a fost dezvoltată separat o librărie ce se foloseşte de acest protocol. În momentul conectării în aplicaţie, se crează un socket către server-ul Web ce va rămane activ şi va permite transmiterea datelor în mod bidirecţional.



Modelul Model-View-Controller al aplicaţiei Android

2.4.2 ŞIRUL EVENIMENTELOR APLICAȚIEI ANDROID

Aplicaţia Android este necesară doar utilizatorului pentru a vedea progresul obţinut în timp real. Astfel el are la îndemână următoarele tool-uri în cadrul aplicaţiei :

- Autentificarea prin ecranul de înregistrare sau ecranul de conectare
- Vizualizarea într-un grafic a numărului de paşi parcurşi în ziua curentă până la momentul curent
- Vizualizarea informaţiilor şi sfaturilor medicului supervizor
- Vizualizarea greutăţii curente prin raportare la greutatea optimă la care trebuie să ajungă aşa cum a fost stabilită de către medicul supervizor
- Posibilitatea de a seta informaţiile ce nu pot fi determinate dinamic : greutatea curentă, vârsta, înălţimea cât şi conexiunea cu dispozitivul pedometru.

CAPITOLUL 3. IMPLEMENTAREA SISTEMULUI

3.1 IMPLEMENTAREA DISPOZITIVELOR EMBEDDED

Pentru programarea plăcii de dezvoltare Arduino Pro Mini a fost folosit mediul de dezvoltare Arduino IDE (Integrated Development Environment). Pentru aceste plăci de dezvoltare au fost scrise diverse librării pentru a îndeplini anumite funcţionalităţi cum ar fi comunicarea prin interfaţa digitală I2C cu senzorii sau citirea datelor primite prin modulul Bluetooth.

Limbajul de programare folosit pentru Arduino este limbajul C, iar funcţiile principale ale unui program sunt setup şi loop :

- void setup() {}

Funcţia void setup este o funcţie apelată automat la pornirea programului şi este folosită pentru definirea de variabile, încărcarea bibliotecilor etc.

- void loop() {}

Funcţia loop este funcţia principală ce va fi rulată de dispozitivul embedded la infinit. Aici se pune codul C principal.

- double variabilaA = 0;
- int variabilaB = 0;

Variabilele globale se poziţionează înafara acestor două funcţii

3.1.1 IMPLEMENTAREA DISPOZITIVULUI PEDOMETRU

3.1.1.1 PRELUAREA VALORILOR DE LA SENZORUL DE ACCELERAȚIE ADXL 345

Pentru a putea comunica cu senzorul de acceleraţie ADXL345 a fost folosită librăria I2Cdev.h, fiind inclusă prin comanda #include I2Cdev.h

Protocolul de comunicaţie I2C permite unui dispozitiv configurat în mod „slave", în cazul nostru senzorul de acceleraţie, să comunice cu un dispozitiv configurat în mod „master", în cazul nostru placa de dezvoltare Arduino.

Pentru a putea determina poziţia în care este aşezat accelerometrul, la pornirea lui se vor citi datele de pe cele trei axe, iar axa cu valoarea cea mai mare va fi folosită ca şi axă de referinta.

Pentru a determina efectuarea unei mişcări, se citesc valorile de pe axa de referinţă într-o buclă infinită. Dacă se determină că valoarea citită şi procesată este încadrată în anumiţi parametri, se va transmite un semnal prin modulul Bluetooth către dispozitivul Android conectat.

3.1.1.2. COMUNICAREA PRIN MODULUL BLUETOOTH HC-06

Pentru comunicarea cu dispozitivul Android a fost folosită librăria SoftwareSerial. Arduino permite comunicarea serială pe pinii 0 şi 1 prin o componentă hardware denumită UART. Aceşti pini sunt folosiţi şi pentru conexiunea USB cu computer-ul de pe care se dezvoltă programul ce va fi folosit pe placă.

Astfel a intervenit nevoia de comunicare serială de pe alţi pini digitali ai plăcii şi a fost dezvoltată librăria SoftwareSerial ce îndeplineşte aceleaşi funcţii ca şi comunicarea serială nativă folosind o componentă software.

Pinii folosiţi pentru comunicarea cu ajutorul acestei librării sunt pinii 10 si 11. Aşa cum a fost menţionat la pasul anterior, la fiecare detectare a unui pas, se trimite cu ajutorul comenzii "write" valoarea 1 (semnificând efectuarea unui pas) către dispozitivul Android.

3.2 IMPLEMENTAREA BAZEI DE DATE

Bazele de date de tip NoSQL se bazează pe organizarea informaţiilor sub forma documentelor, care la rândul lor pot conţine alte documente.

Beneficiul adus de utilizarea acestui tip de baze de date în contextul temei aplicaţiei şi în contextul modelului de publicare-subscripţie oferit de MeteorJS este oferit de faptul că putem obţine toate informaţiile necesare legate de un anume utilizator printr-o singură subscripţie, nefiind nevoie să facem interogări în mai multe tabele.

De exemplu, pentru a crea o colecţie denumită „messages" este necesar să rulăm următoarea comandă : Messages = new Mongo.Collection("messages");

În momentul rulării acestei comenzi se întâmplă următoarele lucruri :

- Server – este creată o colecţie cu numele trimis ca parametru pe serverul Mongo
- Client – este creată o instanţă Minimongo

    - instanţă Minimongo – o implementare în memorie, impersistentă, în Javascript a sistemului de baze de date Mongo - are rolul de a menţine un cache local al datelor de pe server
- În momentul în care se execută o operaţie în client (insert, update, remove), comanda se execută pe client şi simultan este trimisă şi către server unde va fi rulată iar. Acest lucru aduce beneficiul unei modificări a interfeţei în timp real, pentru că nu se aşteaptă rezultatul execuţiei de pe server.

3.3 IMPLEMENTAREA APLICAȚIEI WEB

3.3.1 TEHNOLOGII FOLOSITE

3.3.1.1 METEORJS

MeteorJS este o platformă full-stack Javascript ce permite dezvoltarea aplicaţiilor web şi mobile moderne. Meteor sau MeteorJS include un set de tehnologii pentru crearea aplicaţiilor reactive, un sistem de compilare şi o listă de librării atent alese dintre cele dezvoltate de către comunitatea Node.js şi Javascript.

Platforma foloseşte o combinaţie de cod JavaScript scris pentru front-end ce rulează în browser, cod JavaScript scris pentru back-end ce rulează pe un server Meteor într-un container NodeJS şi alte fişiere HTML, CSS şi imagini pentru a crea interfeţe utilizator ce se pot modifica fără a fi nevoie să efectuăm operaţia de refresh a paginii în browser.

De asemenea, se integrează foarte uşor cu librăria ReactJS folosită în aplicaţia dezvoltată în acest proiect.

Fiind un framework full-stack, ne permite să dezvoltăm aplicaţia folosind un singur limbaj de programare şi anume JavaScript. Este de asemenea un framework reactiv, ceea ce crează interfeţe grafice ce se modifică în timp real, o funcţionalitate la care utilizatorii moderni se aşteaptă.

Această capacitate de a prezenta informaţiile în timp real este foarte importantă pentru aplicaţiile în care utilizatorii doresc să împartă informaţiile chiar în timp ce se întâmplă.

Dintr-un punct de vedere tehnic, Meteor este reactiv datorită modului de a reîmprospăta datele. Pentru a face aceasta, el va menţine în browser o replică a bazei de date MongoDB, denumită „minimongo”. Toate update-urile datelor se vor întâmpla live în aplicaţie, pe ecranul utilizatorului. Acest lucru este posibil datorită modelului MVC folosit de Meteor.

În momentul în care se efectuează modificări în baza de date de către utilizator se merge pe principiul de „Optimistic UI” : când clientul doreşte să facă o modificare, se apelează o metodă care va fi procesată în browser, iar interfaţa grafică se va modifica presupunând că răspunsul venit de la server în urma cererilor de modificare este pozitiv, fără a aştepta efectiv răspunsul server-ului; simultan aceeaşi metodă se va rula şi pe server şi dacă răspunsul este într-adevăr pozitiv, se vor efectua modificările asupra colecţiilor din baza de date.

Dacă apelul metodei va rezulta într-o eroare, server-ul va notifica clientul iar acesta va executa o operaţiune de „undo” asupra modificărilor din interfaţa grafică.

3.3.1.2 MONGODB

MongoDB este o bază de date nerelaţională, scrisă în C++, un avantaj major fiind faptul că nu există câmpuri predefinite spre deosebire de bazele de date relaţionale, cum ar fi MySQL, unde există coloanele definite în momentul creării tabelelor.

În MongoDB nu există o schemă pentru câmpurile unui document sau pentru tipurile lor de date, ele având posibilitatea să varieze în funcţie de ceea ce doreşte utilizatorul să introducă; fiecare document trebuie să aibă totuşi un câmp ”_id” care reprezintă cheia unică a acelui document. Acest câmp poate fi creat fie de creatorul bazei de date, fie este generat automat după un algoritm astfel încât oricâte documente ar fi introduse, el nu se va repeta.

MongoDB memorează datele ca documente în format BSON (Binary JSON – format de interschimb şi transfer al datelor reţea), avantajul fiind oferit de faptul că se reduce nevoia de join.

Crearea indecşilor secundari şi compuşi reprezintă un alt avantaj al MongoDB, orice atribut având posibilitatea de a fi indexat.

Rezumând, dacă cititorul este familiarizat cu tehnologia bazelor de date relaţionale, se poate face o corespondenţă între acestea şi bazele de date nerelaţionale.

O bază de date MongoDB conţine mai multe colecţii, care sunt asemănătoare cu tabelele din bazele de date relaţionale. Colecţiile conţin documente, iar fiecare document al colecţiei are ca şi corespondent un rând al unui tabel.

3.3.2 PROGRAMARE WEB

3.3.2.1 HTML

HTML este un limbaj de marcare orientat către prezentarea documentelor text pe o singură pagină. Utilizând un software de redare specializat, numit agent utilizator HTML (cel mai bun exemplu de astfel de software fiind browserul web).

HTML furnizează mijloacele prin care conţinutul unui document poate fi adnotat cu diverse tipuri de metadate şi indicaţii de redare.

3.3.2.2 CSS

Crearea paginilor HTML este un lucru relativ simplu, învăţarea etichetelor HTML şi crearea unor imagini ducând la realizarea de pagini web de o complexitate medie. Odată cu dezvoltarea internetului, site-urile au devenit din ce în ce mai complexe, cu un număr mai mare de pagini, cerinţele privind grafica şi elementele din pagină au devenit mai pretenţioase şi astfel proiectarea paginilor web a devenit o sarcină ceva mai dificilă.

O problemă importantă când avem un site cu multe pagini este atunci când dorim să facem anumite schimbări în elementele paginii: fondul, grafica sau fontul textelor din pagini.

Prin utilizarea CSS (Cascading Style Sheets), în traducere "foi de stil în cascadă", acest lucru nu mai este o problemă, realizându-se relativ uşor, prin schimbarea sau adăugarea unor elemente în codul CSS.

CSS se ocupă în general cu aspectul şi controlul grafic al elementelor din pagină, cum ar fi: textul, imaginile, fondul, culorile şi aşezarea acestora în cadrul ferestrei paginii.

3.3.2.3 JAVASCRIPT

JavaScript este un limbaj de programare orientat obiect bazat pe conceptul prototipurilor. Este folosit mai ales pentru introducerea unor functionalităţi în paginile web, codul Javascript din aceste pagini fiind rulat de către browser.

Limbajul este binecunoscut pentru folosirea sa în construirea site-urilor web, dar este folosit şi pentru accesul la obiecte încastrate (embedded objects) în alte aplicaţii.

JavaScript este util pentru a verifica validitatea informaţiilor introduse într-un formular înainte ca datele să fie trimise către server. O menţiune importantă: programele care rulează pe calculatorul utilizatorului sunt numite aplicaţii client-side (aflate pe partea de client) şi programele care rulează pe server (inclusiv CGI-urile) sunt numite aplicaţii server-side (aflate pe partea de server).

3.3.2.4 REACTJS

React (React.js sau ReactJS) este o librărie JavaScript open-source ce permite dezvoltarea interfeţelor utilizator. A fost creată şi este menţinută în continuare de către Facebook, Instagram şi o comunitate de dezvoltatori şi corporaţii.

Unul dintre punctele centrale ale acestei librării este că poate crea paginile HTML pe partea de server. De asemenea, foloseşte un concept denumit Virtual DOM (Document Object Model virtual) ce permite recrearea sub-nodurilor atunci când starea paginii este modificată.

Modul în care funcţionează DOM-ul virtual este următorul : atunci când modificăm starea iniţială a unui obiect, React va executa la primul pas un algoritm de diferenţiere între starea iniţială şi cea curentă pentru a identifica schimbările efectuate, iar în cel de-al doilea pas va realiza un „update” al DOM-ului doar cu rezultatele obţinute la primul pas. Eficientizarea în acest punct este oferită de faptul că dacă un obiect-copil este modificat, doar el va fi recreat, nu şi părintele.

Pentru a putea scrie o componentă React, se va folosi limbajul JSX, o extensie a limbajului JavaScript ce permite folosirea tag-urilor HTML în componenţa sa.

Atributele componentelor sunt denumite „*props*” şi pot fi trimise ca şi parametri către sub-componente. De asemea, starea unei componente, denumită „*state*”, poate fi modificată şi este punctul central ce determină declanşarea update-urilor pentru interfaţa cu utilizatorul.

Din punct de vedere arhitectural, datele aplicaţiei au un caracter unidirecţional : într-o ierarhie cu componente multiple, componenta-părinte este responsabilă cu privire la starea aplicaţiei şi poate oferi aceste valori ca şi atribute componentelor-copil.

3.3.2.5 CHARTISTJS

ChartistJS este o librărie folosită pentru generarea de grafice în funcţie de o serie de date de intrare. Această bibliotecă oferă o flexibilitate sporită pentru generarea şi personalizarea unui grafic de tip linie, de tip radar şi de tip bară.

Această librărie a fost folosită în prezentarea statisticilor într-un mod grafic uşor şi rapid de înţeles.

3.4 IMPLEMENTAREA APLICAȚIEI ANDROID

3.4.1 TEHNOLOGII FOLOSITE

3.4.1.1 ANDROID

Android este o platformă software şi un sistem de operare pentru dispositive mobile bazată pe nucleul Linux, dezvoltată iniţial de compania Google, iar mai târziu de consorţiul commercial Open Handset Alliance. Android permite dezvoltatorilor să scrie cod gestionat în limbajul Java, controlând dispozitivul prin intermediul bibliotecilor Java dezvoltate de Google. Aplicaţiile scrise în C şi alte limbaje pot fi compilate în cod masină ARM şi executate, dar acest model de dezvoltare nu este sprijinit oficial de către Google.

În ultimii ani Android a cunoscut o dezvoltare foarte rapidă şi nu numai prin faptul că este un proiect “open source”, dar şi prin faptul că este un sistem foarte bine optimizat, uşor de utilizat şi foarte sigur din punct de vedere al securităţii. Astfel a fost actualizat pentru a putea fi folosit şi pe o altă gamă de dispozitive, putând fi utilizat şi în domeniul industrial prin intermediul dispozitivelor embedded.

Android este un mediu software construit pentru dispozitive mobile în special. Nu este o platformă hardware. Android include un sistem de operare bazat pe kernel Linux, o interfaţă utilizator bogată, aplicaţii, biblioteci, frameworkuri, suport multimedia, şi multe altele. În timp ce componentele care stau la baza sistemului de operare sunt scrise în limbajul C sau C++, aplicaţiile utilizator pentru Android sunt construite în Java. Chiar şi aplicaţiile care constitue partea integrată sunt scrise în Java.

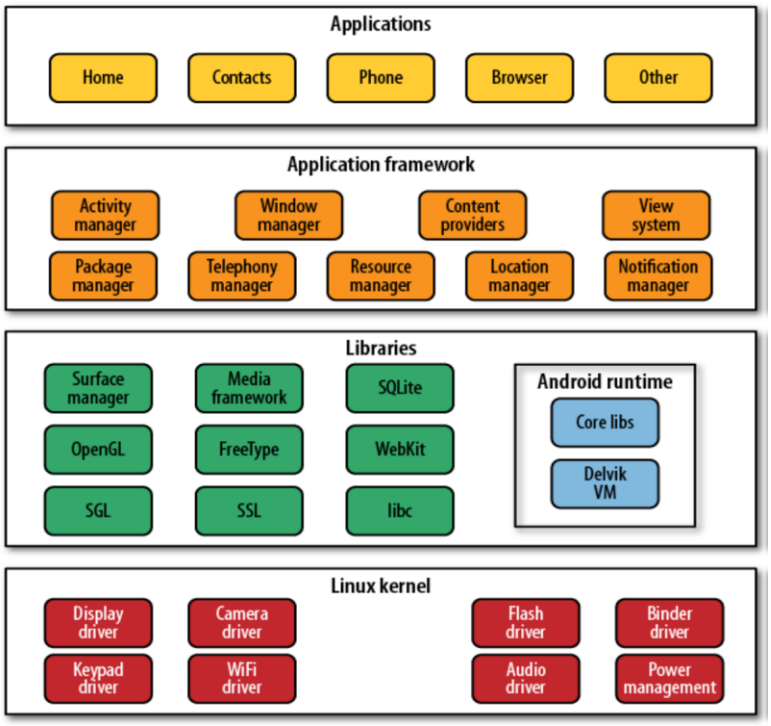
Android SDK

O caracteristică a platformei Android este că nu există nici o diferenţă între aplicaţiile built-in şi aplicaţiile create cu SDK. Acest lucru înseamnă că se pot scrie aplicaţii care pot atinge resursele care se află pe dispozitiv.

Android SDK oferă API-uri şi instrumentele necesare pentru a dezvolta, testa, şi depana aplicaţii pentru Android. Android SDK suportă majoritatea librăriilor din Java Platform, Standard Edition (Java SE), cu excepţia Abstract Window Toolkit (AWT) şi Swing. În loc de AWT şi SWING, SDK-ul Android are propriul framework pentru realizarea de interfeţe utilizator.

The Stack

Sistemul de operare Android este ca un tort format din diferite straturi. Fiecare strat are propriile caracteristici şi scop. Straturile nu sunt separate complet, ci de obicei sunt înfiltrate între ele.



Arhitectura sistemului de operare Android

3.4.2 PROGRAMARE ANDROID

3.4.2.1 MP ANDROID CHART

MP Android Chart este o bibliotecă Android folosită pentru generarea de grafice.

Aceasta este foarte simplu de folosit şi oferă o flexibilitate sporită în ceea ce priveşte personalizarea.

Pentru folosirea graficului trebuie incluse în fişierul build.gradle al aplicaţiei Android următoarele:

```
dependencies {
        compile 'com.github.PhilJay:MPAndroidChart:v3.0.0'
}
```

Utilizând această bibliotecă se pot creea şi personaliza diferitele tipuri de grafice cum ar fi Line Chart, Bar Chart şi Pie Chart.

Această librărie a fost folosită pentru crearea graficului de tip „bară” pentru a vedea statistica numărului de paşi efectuaţi în ultimele 7 zile.

3.4.3 IMPLEMENTAREA FIZICĂ A APLICAȚIEI ANDROID

Aplicaţia Android este creată sub forma unor ecrane separate ce pot fi vizualizate prin executarea acţiunii de „swipe” pe ecranul dispozitivului. Ea conţine mai multe activităţi ce sunt deschise la nevoie prin intermediul unor Intent-uri.

Înainte de a putea folosi toate opţiunile aplicaţiei, utilizatorul trebuie să fie conectat. Dacă a creat deja un cont de utilizator pe aplicaţia Web, el se poate autentifica cu e-mailul şi parola înregistrate. De asemenea, are posibilitatea de a-şi crea un cont direct în aplicaţia Android. În ambele ecrane, utilizatorului îi sunt verificate datele de intrare pentru a îndeplini toate cerinţele necesare.

Primul ecran vizualizat după conectare este cel al graficului pedometrului. Acesta indică numărul de paşi efectuaţi până la momentul curent în ziua curentă. Înainte de a reseta valoarea acestui grafic la valoarea 0 la sfârşitul fiecărei zile, datele vor fi trimise către server-ul Web, dacă este disponibilă o conexiune de date (WiFi, date mobile). De asemenea, alături de aceste date este indicat şi timpul total de activitate de când a fost detectat dispozitivul pedometru de către aplicaţia Android.

Ecranul privitor la detaliile de sănătate, sfaturi şi recomandări afişează şi informaţii cu privire la înălţimea, vârsta şi greutatea utilizatorului. Bazat pe aceste valori sunt calculate atât cantitatea de apă ce trebuie consumată într-o zi cât şi numărul de calorii necesare pentru o dietă normală în funcţie de intensitatea activităţilor fizice. De asemenea, în acelaşi ecran există posibilitatea de deschidere a unei activităţi ce conţine sfaturi cu privire la o dietă echilibrată, special concepută pentru utilizatorii ce doresc să scadă din greutate.

Ecranul de profil prezintă alături de informaţiile personale ale utilizatorului şi informaţii cu privire la distanţa, greutatea medie atinsă, cea mai lungă distanţă parcursă într-o zi precum şi cea mai mică greutate atinsă.

Ecranul cu privire la greutate, aşa cum a fost menţionat, prezintă sub forma unui grafic de tip „arc de cerc”, valoarea curentă a greutăţii exprimată în kilograme prin raportare la greutatea optimă ce ar trebui atinsă de utilizator.

Ecranul de statistici, prezintă cu ajutorul librăriei MPAndroidChart statistici cu privire la numărul de paşi parcurşi în ultimele 7 zile, precum şi numărul total de paşi, dar şi media numărului de paşi parcurşi în decurs de o zi.

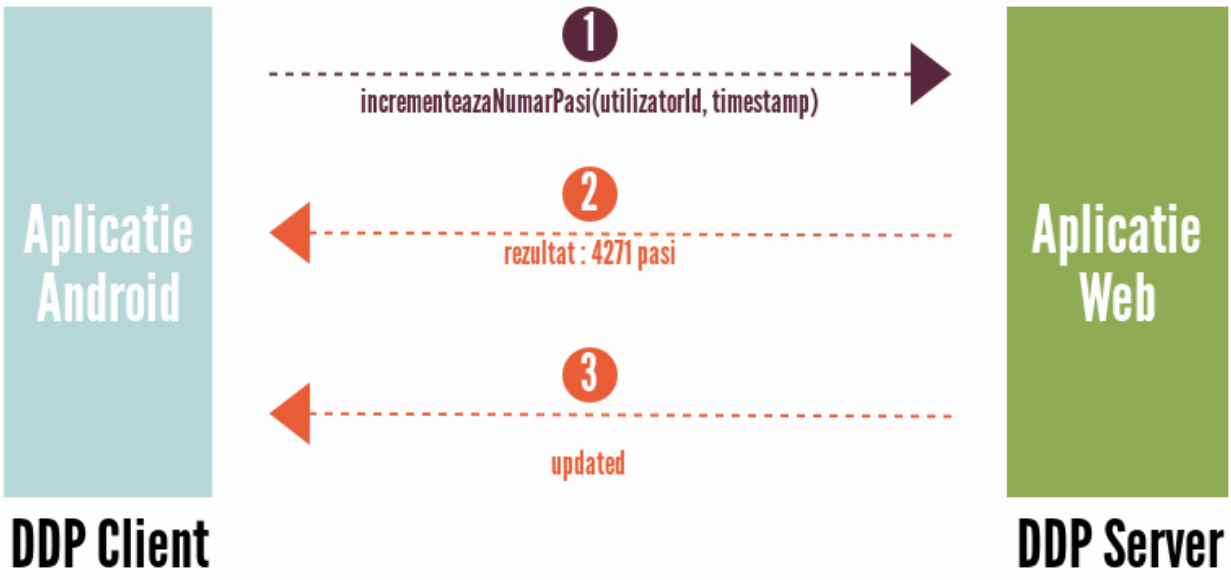
Ultimul ecran, cel al setărilor, permite utilizatorului să introducă data de naştere, greutatea curentă, înălţimea cât şi sexul. De asemenea, permite sincronizarea datelor la momentul curent cu cele de pe server, cât şi ştergerea tuturor datelor şi ieşirea din aplicaţie. Din acest ecran se poate deschide activitatea ce permite conectarea dispozitivului Android cu dispozitivul Arduino.

Prin activarea opţiunii Bluetooth, se vor detecta toate dispozitivele din proximitate ce deţin un modul Bluetooth şi se va alege dispozitivul HC-06. După ce au fost conectate cele două dispozitive, aplicaţia este pregătită pentru a primi date de la pedometru, şi de a le afişa în graficele corespunzatoare.

3.5 PROTOCOLUL DDP

DDP, sau Distributed Data Protocol, este protocolul bazat pe JSON folosit de Meteor pentru comunicarea între server şi client. Implementarea curentă a acestui protocol în Meteor se bazează pe Websockets şi SockJS. Responsabilitatea acestui protocol este de a executa apelurile procedurilor de pe server şi de a administra datele.

Cu ajutorul RPC (Remote Procedure Calls) se poate apela o metodă de pe server pentru a se primi un răspuns.



Apelul unei metode de pe clientul DDP pe serverul DDP

```
1.
    {
        "msg":"method", "method": "incrementeazaNumarPasi",
        "params": ["507f191e810c19729de860ea", "1489567999"],
        "id": "randomId-1"
    }
2.
    {
        "msg": "result", "id": "randomId-1", "result": "4271"
    }
3.
    {
        "msg": "updated", "methods": ["randomId-1"]
    }
```

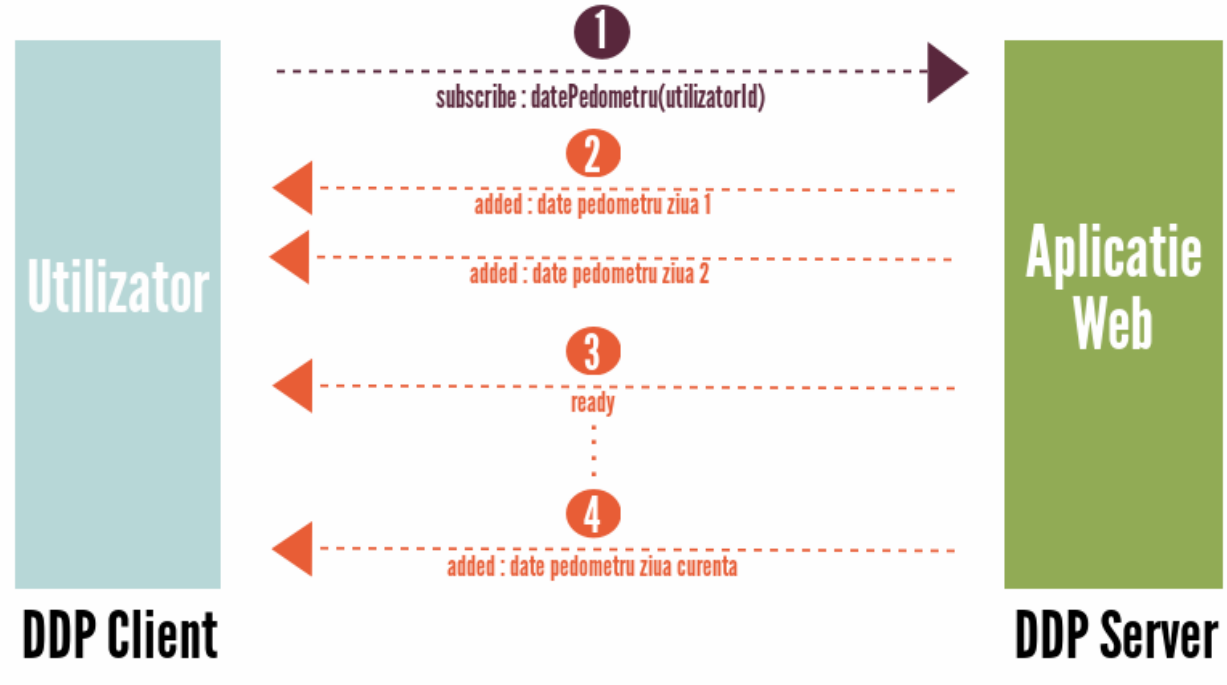
Secvenţa de mesaje schimbate între clientul DDP şi serverul DDP

Modelul de publicare-subscripţie este realizat tot cu ajutorul protocolului DDP. Un client poate primi una din următoarele trei tipuri de notificări : "*added*", "*changed*", "*removed*". Fiecare notificare este alocată la o anumită colecţie din baza de date, protocolul fiind inspirat de către MongoDB.

Acest model poate fi reprezentat cel mai bine prin următorul exemplu :

- Clientul DDP cere o cerere de subscripţie pentru un anumit set de date

- Clientul va primi o serie de notificări cu obiectele din colecţia la care s-a făcut subscripţia
- După ce toate obiectele au fost primite de către client, el va primi un mesaj special denumit "*ready*" care indică faptul că toate datele subscripţiei au fost trimise de la server către client şi se poate începe procesarea lor.



Secvenţa de subscriere la o colecţie

```
1.
    {
        "msg": "sub",
        "id": "randomId-2",
        "name": "datePedometru",
        "params": ["507f191e810c19729de860ea"]
    }
2.
    {
        "msg": "added", "collection": "datePedometruUtilizator", "id": "record-1",
        "fields": {"numarPasi": "2585", "timestamp": "1489449600"}
    }

    {
        "msg": "added", "collection": "datePedometruUtilizator", "id": "record-2",
        "fields": {"numarPasi": "6870", "timestamp": "1490227200"}
    }
3.
    {
        "msg": "ready":
        "subs": ["randomId-2"]
    }
4.
    {
        "msg": "added", "collection": "datePedometruUtilizator", "id": "record-3",
        "fields": {"numarPasi": "4271", "timestamp": "1489567999"}
    }
```

Secvenţa de mesaje schimbate între clientul DDP şi serverul DDP în momentul subscripţiei la o colecţie

3.6 BENCHMARKS

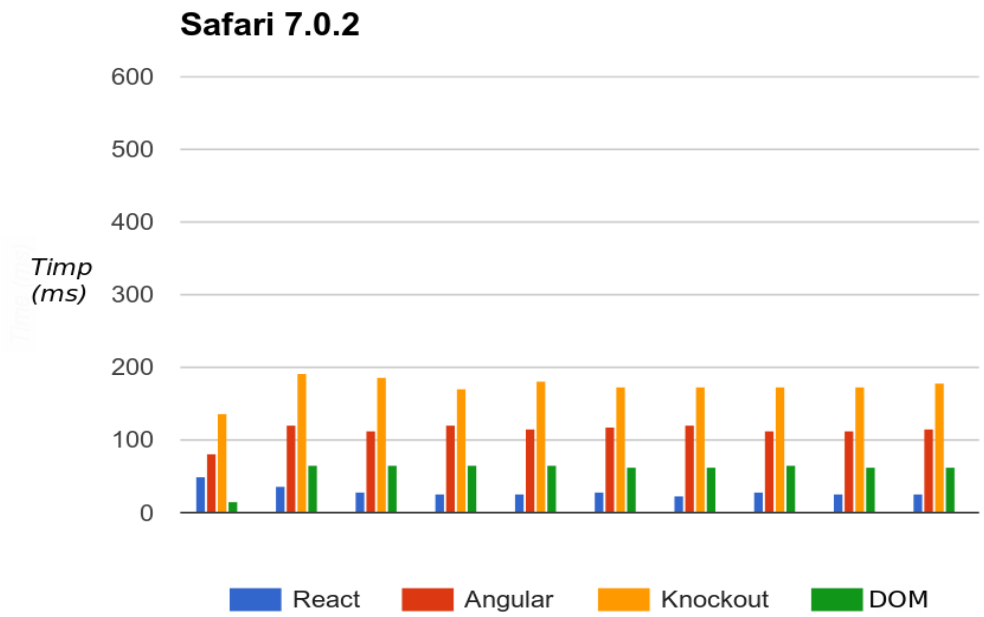
Librăria ReactJS este responsabilă cu procesarea codului JSX şi afişarea elementelor HTML în structura logică a documentelor (DOM). Pentru a testa performanţele acestei librării avem nevoie de valori de referinţă la care ne putem raporta.

Multe din modificările efectuate asupra unei pagini HTML pot fi făcute prin modificarea DOM-ului direct, astfel că acest mod va reprezenta punctul de referinţă. La fel ca React, există şi alte librării binecunoscute care îndeplinesc aceleaşi atribuţii : AngularJS sau KnockoutJS.

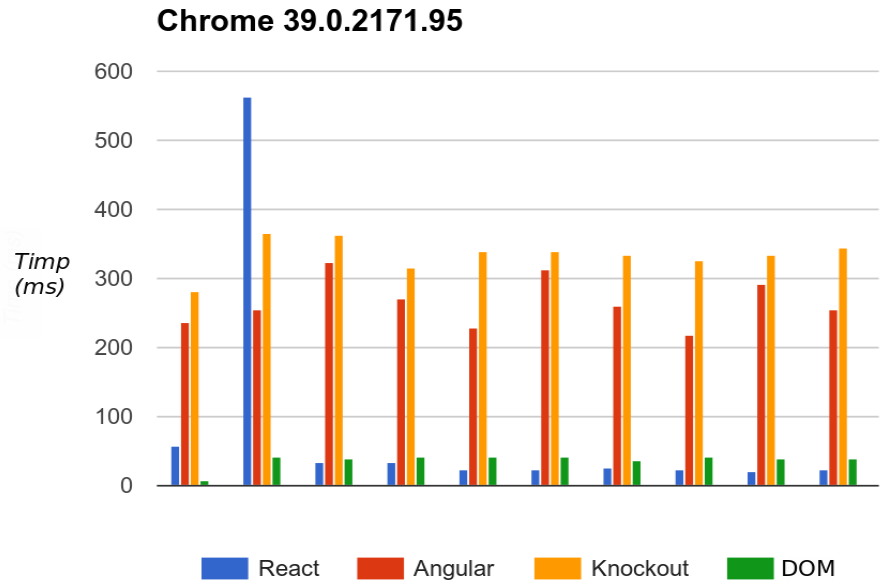
Pentru a testa performanţele librăriei ReactJS a fost efectuat următorul test :

- Se generează 1000 elemente într-o listă HTML cărora li se ataşează un ascultător de evenimente
- Se măsoară indexul timpului în momentul în care se porneşte testul
- Se măsoară indexul timpului după ce au fost afişate cele 1000 elemente
- Se calculează diferenţa între indecşi pentru a determina timpul necesar afişării pe ecran a elementelor

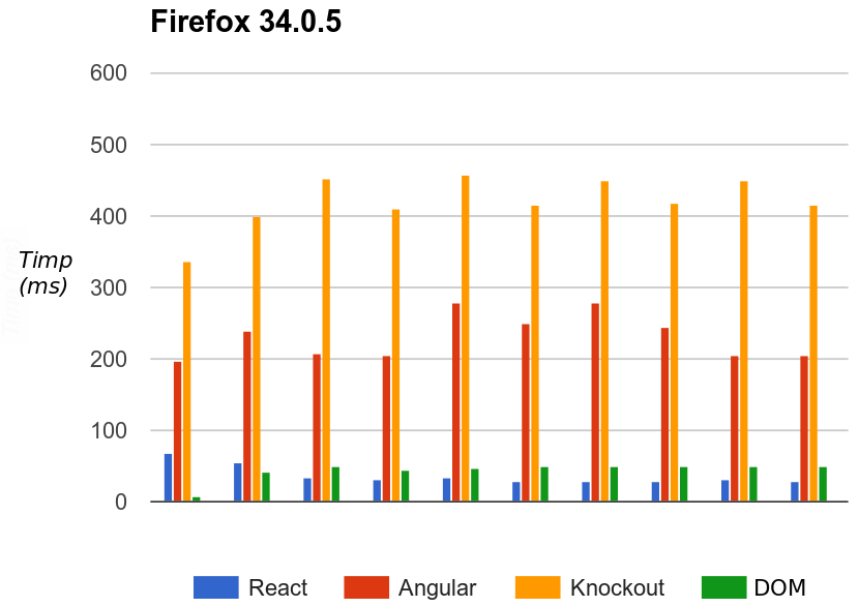
Testele au fost efectuate în browserele Chrome 39.0.2171.95, Firefox 34.0.5 şi Safari 7.0.2, iar rezultatele pot fi urmărite în graficele următoare :



Rezultatele testelor în urma testării în browser-ul Safari 7.0.2



Rezultatele testelor în urma testării în browser-ul Chrome 39.0.2171.95



Rezultatele testelor în urma testării în browser-ul Firefox 34.0.5

Conform graficelor, ReactJS este cea mai rapidă librărie în comparaţie cu AngularJS şi KnockoutJS, unele dintre cele mai folosite librării pentru generarea elementelor HTML.

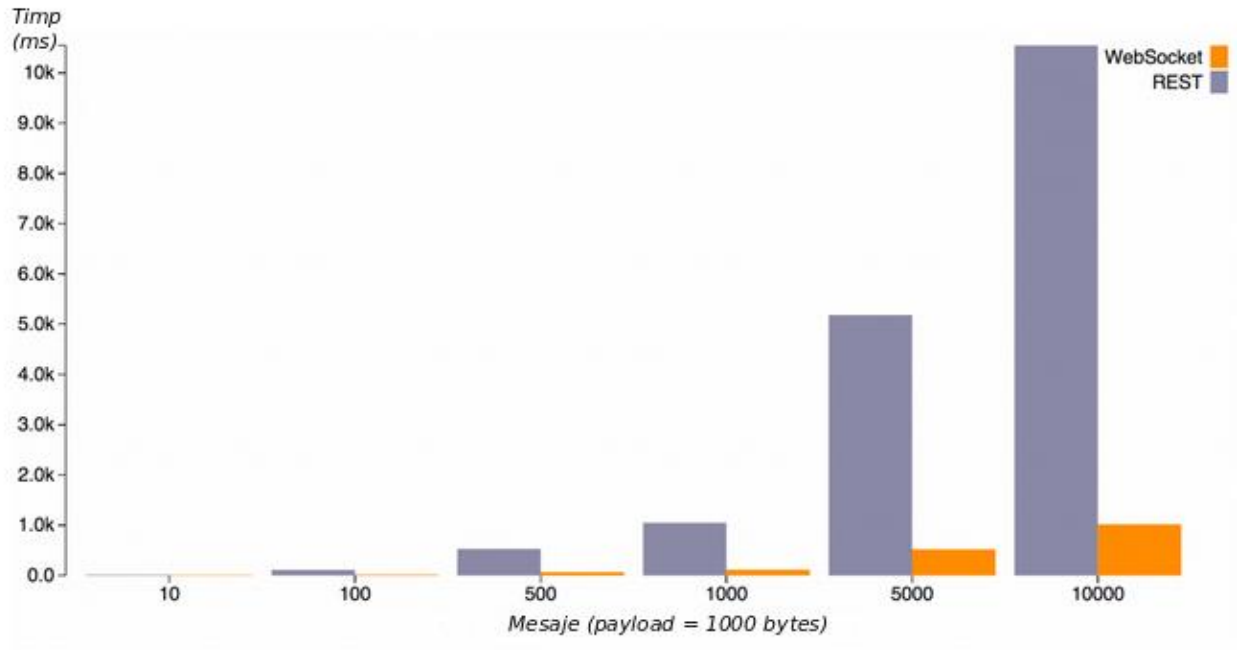
De asemenea, o altă alegere naturală a fost crearea conexiunii dintre client (web sau Android) şi server prin Websockets, în detrimentul creării rutelor HTTP pentru inserarea sau citirea datelor din baza de date.

Un websocket este un canal de comunicaţie bidirecţional şi full-duplex printr-o conexiune TCP unde atât clientul cât şi serverul pot iniţia trimiterea unui mesaj. Prin comparaţie, protocolul HTTP este unidirecţional şi doar clientul are posibilitatea de a realiza cereri serverului care le procesează şi returnează un răspuns.

Caracterul full-duplex permite atât clientului să trimită un mesaj serverului, cât şi invers, într-un mod independent.

Utilizând protocolul HTTP, de fiecare dată când un client accesează o rută de pe server, o nouă conexiune TCP este creată şi distrusă în momentul în care serverul returnează un răspuns, ceea ce nu se întâmplă în cazul folosirii Websocket-ului unde conexiunea este actualizată folosind mecanismul standard HTTP şi astfel se refoloseşte conexiunea TCP pe toată durata de viaţă a conexiunii prin Websocket.

Următorul grafic arată timpul necesar procesării unui număr variabil de mesaje cu un payload constant :



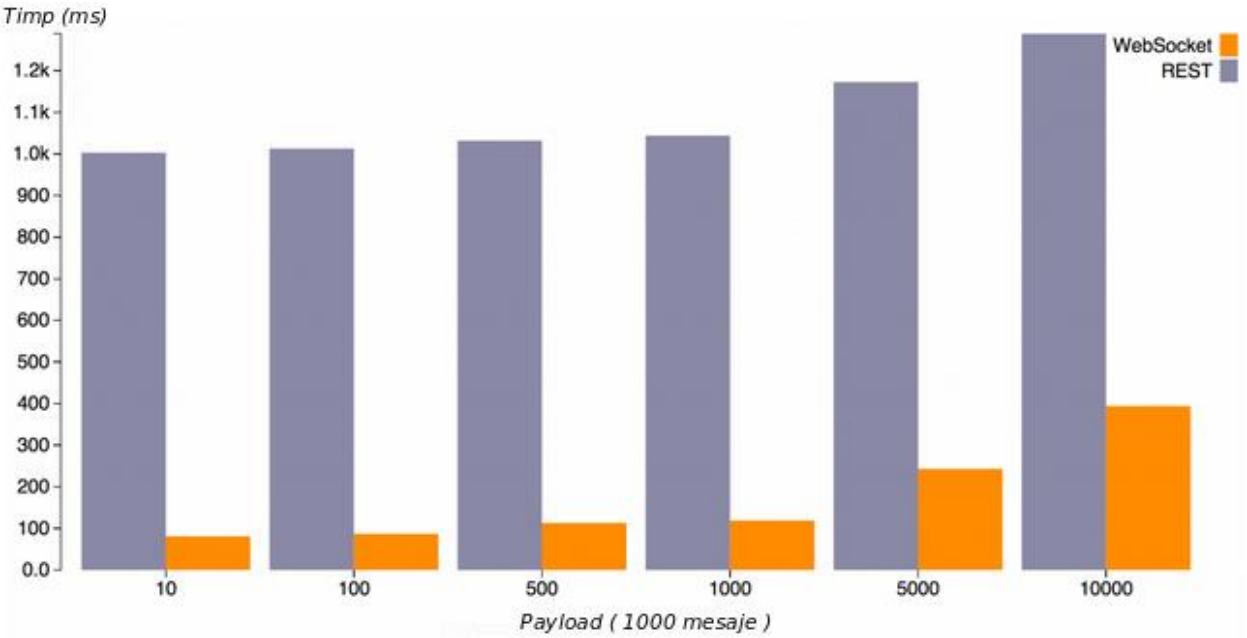
Timpul necesar procesării unui număr variabil de mesaje cu payload constant

Payload constant, număr variabil de mesaje		
Număr mesaje	REST (ms)	Websocket (ms)
10	17	13
100	112	20
500	529	68
1000	1050	115
5000	5183	522
10000	10547	1019

Timpul necesar procesării unui număr variabil de mesaje cu payload constant

Aşa cum se poate observa din tabel, timpul necesar creşte considerabil deoarece aşa cum a fost menţionat, o nouă conexiune TCP trebuie creată şi distrusă la fiecare mesaj.

Testul următor arată timpul necesar procesării unui număr constant de mesaje dar cu payload variabil :



Timpul necesar procesării unui număr constant de mesaje cu payload variabil

Payload variabil, număr constant de mesaje		
Payload (bytes)	REST (ms)	Websocket (ms)
10	1003	81
100	1013	87
500	1032	113
1000	1044	119
5000	1173	243
10000	1289	394

Timpul necesar procesării unui număr constant de mesaje cu payload variabil

Costul procesării mesajului de la client creşte foarte puţin în cazul accesării rutelor REST deoarece majoritatea timpului este dedicat creării/terminării conexiunii TCP nu procesării efective mesajului.

CONCLUZII

Lucrarea de faţă întruneşte cerinţele impuse pentru monitorizarea şi controlul stării de sănătate a utilizatorului. Acesta va putea observa în timp real informaţii despre starea sa şi poate primi sfaturi în scopul de a-şi îmbunătăţi condiţia actuală de sănătate.

Întreg sistemul foloseşte tehnologii de ultimă oră pentru a oferi utilizatorului o experienţă cât mai plăcută şi cât mai rapidă. Dispozitivele prezentate culeg datele utile cu ajutorul diferitelor tehnologii, senzorilor şi le transmit automat celorlalte dispozitive.

Protocolul DDP permite crearea între client şi server a unei conexiuni ce nu va fi refăcută la fiecare transmitere de date ci va rămâne activă pe toata durata folosirii aplicaţiei. Prototiparea unei astfel de aplicaţii web se poate face într-un timp restrâns datorită framework-ului MeteorJS şi a librăriei de randare ReactJS ce vin cu diverse componente ce pot fi aplicate fără a fi necesari paşi adiţionali pentru configurare sau modificare.

În concluzie, acest sistem permite înregistrarea mişcării utilizatorului şi posibilitatea monitorizării de către un medic în timp real prin transmiterea datelor de la dispozitivul Android către serverul web. O altă facilitate este posibilitatea comunicării în timp real cu medicul supervizor, un element foarte important chiar în cazul sportivilor de performanţă.

BIBLIOGRAFIE

Cărți:

- David Flanagan - "JavaScript: The DefinitiveGuide", 6th Edition, Editura O'Reilly Media, 2011
- Isaac Strack - “Getting Started with Meteor.js JavaScript Framework Paperback”
- Ken Rogers – “Meteor and React”
- Marcelo Reyna, Isaac Strack, Fabian Vogelsteller – “Meteor: Full-Stack Web Application Development ”
- Sabin Buraga - ”Proiectarea site-urilor Web”, Editura Polirom, 2002
- Shelley Power – “JavaScript Cookbook”, Editura O'Reilly Media, 2007
- Tom Coleman, Sacha Greif – “Discover Meteor “

Resurse Web:

- https://developer.android.com/index.html
- https://facebook.github.io/react/
- https://guide.meteor.com/
- http://www.w3resource.com/
- http://www.w3schools.com/