Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky

Aplikácie pre spracovanie údajov určujúcich kritický zdravotný stav pacienta, s využitím mobilných zariadení

Diplomová práca

2016 Nikola Cichovská

Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky

Aplikácie pre spracovanie údajov určujúcich kritický zdravotný stav pacienta, s využitím mobilných zariadení

Diplomová práca

Študijný program:	Informatika
Študijný odbor:	9.2.1 Informatika
Školiace pracovisko:	Katedra počítačov a informatiky (KPI)
Školiteľ:	Ing. Branislav Madoš, PhD.
Konzultant:	Ing. Marián Zorkovský
	Ing. Mikuláš Fedorčák

Košice 2016 **Nikola Cichovská**

Abstrakt v SJ

Diplomová práca je venovaná analýze vývoja medicínskeho softvéru pre mobilné zariadenia. Opisuje dostupné mobilné aplikácie vyvinuté pre medicínske odvetvie a porovnáva ich výhody a nedostatky. Ďalej definuje architektúry troch najrozšírenejších mobilných platforiem a venuje sa podrobnej charakteristike vývojových prostredí pre multiplatformový vývoj softvéru. V tejto práci je navrhnutá a implementovaná aplikácia pre dve mobilné platformy, ktorá prijíma EKG dáta z prídavného zariadenia Raspberry Pi, spracúva ich a následne vyhodnocuje.

Kľúčové slová

Medicínske systémy, mobilná aplikácia, Android, Windows Phone, Xamarin, EKG

Abstrakt v AJ

This Master thesis is dedicated to analysis of evolution of medical software for mobile devices. It describes available mobile applications developed for medical industry and comaperes their advantages and disadvantages. Furthermore it defines archotecture of three most widespread mobile platforms and describes in detail characteristic of development environments for crossplatform software development. In this thesis is designed and implemented application for two mobile platforms that recieves ECG data from auxilliary device Raspberry Pi, process them and evaluates them.

Kľúčové slová v AJ

Medical systems, mobile application, Android, Windows Phone, Xamarin, ECG

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
Katedra počítačov a informatiky

ZADANIE DIPLOMOVEJ PRÁCE

Študijný odbor: **9.2.1 Informatika**
Študijný program: **Informatika**

Názov práce:

Aplikácia pre spracovanie údajov určujúcich kritický zdravotný stav pacienta s využitím mobilných zariadení
Mobile device applications for data processing they indicates emergency situation of patient

Študent:	**Bc. Nikola Cichovská**
Školiteľ:	**Ing. Branislav Madoš, PhD.**
Školiace pracovisko:	**Katedra počítačov a informatiky**
Konzultant práce:	**Ing. Marian Zorkovský, Ing. Mikuláš Fedorčák**
Pracovisko konzultanta:	**Siemens, s.r.o., Healthcare, Lamačská cesta 3/A, Bratislava 841 04**

Pokyny na vypracovanie diplomovej práce:

1. Analyzovať spôsob a vhodnosť využitia prenosných mobilných zariadení, ich HW a SW vybavenia, operačných systémov, vývojových prostriedkov a pod., pre spracovanie údajov indikujúcich kritický zdravotný stav pacienta.
2. Navrhnúť a na úrovni funkčného prototypu implementovať aplikáciu pre:
i. Vstup potrebných biometrických a riadiacich údajov a ich spracovanie.
ii. Spoľahlivé vyhodnotenie a ďalšie spracovanie získaných, vstupných údajov pre účely ich aplikačného využitia (v zmysle témy DÚ) .
iii. Spracovanie kritickej situácie pomocou aplikácie v mobilnom zariadení.
3. Systém má byť navrhnutý tak, aby:
i. Využil v maximálnej možnej miere bežné, komerčne dostupné, mobilné zariadenia.
ii. Bol konfigurovateľný a jednoducho rozšíriteľný.
iii. Nadväzoval na DÚ „Systém pre monitorovanie a rýchlu diagnostiku kritického stavu pacienta, s využitím mobilných zariadení".
4. Prakticky overiť a otestovať výsledné riešenie.
5. Vypracovať dokumentáciu podľa pokynov vedúceho práce.

Jazyk, v ktorom sa práca vypracuje: slovenský
Termín pre odovzdanie práce: 29.04.2016
Dátum zadania diplomovej práce: 31.10.2015

doc. Ing. Jaroslav Porubän, PhD.
vedúci garantujúceho pracoviska

prof. Ing. Liberios Vokorokos, PhD.
dekan fakulty

Čestné vyhlásenie

Vyhlasujem, že som diplomovú prácu vypracovala samostatne s použitím uvedenej odbornej literatúry.

Košice 28. 4. 2016 .....

Vlastnoručný podpis

Poďakovanie

Na tomto mieste sa chcem poďakovať vedúcemu práce Ing. Branislavovi Madošovi, PhD., ako aj konzultantom Ing. Mariánovi Zorkovskému a Ing. Mikulášovi Fedorčákovi za cenné rady a pripomienky poskytnuté pri vypracovávaní tejto práce.

Obsah

Zoznam obrázkov

Zoznam tabuliek

Zoznam symbolov a skratiek

ARM Advanced RISC Machine

APEJ Asia/Pacific Excluding Japan

API Aplication Programing Interface

DVM Dalvik Virtual Machine

ECG Electrocardiogram

FDA Food and Drug Administration

GPS Global Positioning System

GSM Global System for Mobile

GUI Grafical User Interface

IDC International Data Corporation

IDE Integrated Development Environment

IL Intermediate Language

JNI Java Native Interface

LTE Long Term Evolution

MAC address Media Access Control address

MEA Middle East and Africa

MVC Model View Controller

ORM Object-relational Mapping

PDA Personal Digital Assistant

PDF Portable Document Format

PNG Portable Network Graphics

SQL Structured Query Language

SVG Scalable Vector Graphics

TFS Team Foundation Server

UI User Interface

UUID Universally Unique Identifier

WPF Windows Presentation Foundation

XAML Extensible Application Markup Language

XML Extensible Markup Language

XWT XML Windowing Toolkit

Slovník termínov

Back-end je podporná služba v pozadí, s ktorou používateľ neprichádza priamo do styku.

Cloud je vyhradené miesto na internete, kde je možné ukladať rôzne druhy informácií ako napríklad fotografie, dokumenty, hudbu, aplikácie, videá a pod.

Engine je počítačový termín , ktorý znamená jadro databázového systému, počítačovej hry alebo programu.

Framework je softvérový rámec. Predstavuje knižnice, ktoré majú uľahčiť prácu pri programovaní, ako napríklad prehľadnejší kód a rýchlejší vývoj systému.

Front-end označuje časti systému, ktoré sú viditeľné pre bežného používateľa.

iTunes je aplikácia slúžiaca na prehrávanie a správu hudby a videí vytvorená spoločnosťou Apple.

Layout označuje grafické rozmiestnenie základných prvkov na používateľskom rozhraní.

Middleware je počítačový termín pre softvér, ktorý umožňuje prepojiť softvérové komponenty a ich aplikácie.

Raspberry Pi je názov minipočítača s ARM procesorom.

Smartfón je inteligentné mobilné zariadenie.

Úvod

Vývoj informačných technológií v posledných desaťročiach smeruje k expanzii do všetkých odvetví ľudských činností. Vplyv technológií na ľudský život je v dnešnej dobre takmer neodmysliteľný. Najväčší prínos však môže mať najmä v medicíne, kde by prostredníctvom inteligentných telefónov, inteligentných náramkov, hodiniek, tabletov či dokonca rôznych prídavných zariadení komunikujúcich priamo s aplikáciou v spomenutých zariadeniach dokázali monitorovať každodenný život pacienta, prispievať k presnejším a konkrétnejším diagnostikám ochorení vedením záznamov o jeho zdravotnom stave či dokonca predvídať nebezpečný zdravotný stav ešte pred tým, než reálne nastane a podniknúť čo najviac krokov pre zabezpečenie okamžitej zdravotnej starostlivosti.

V súčasnosti trpí omnoho väčšie množstvo ľudí srdcovo cievnymi ochoreniami než v minulosti. Počet kardiovaskulárnych chorôb v posledných rokoch stúpol až tak, že sa zaradili k civilizačným chorobám spoločnosti. Práve tento aspekt bol rozhodujúcim faktorom pri výbere témy diplomovej práce. V prípade, že by existovali aplikácie pre inteligentné mobilné zariadenia, ktoré by zabezpečovali neustále monitorovanie srdcovej frekvencie pacienta, ukladali tieto informácie do databázy lekára, dokázali by pomocou vyhodnocovacieho algoritmu predikovať riziko infarktu alebo dokonca podniknúť kroky pre privolanie záchrannej služby, bolo by možné znížiť percento úmrtnosti v dôsledku srdcovo cievnych chorôb a tým nielen zachrániť život pacientom, ale aj zlepšiť kvalitu ich života.

Táto diplomová práca sa zameriava na vývoj aplikácie pre mobilné zariadenia, ktorá by v budúcnosti mohla uľahčiť prácu lekárom neustálym monitorovaním srdcovej frekvencie pacienta dvadsaťštyri hodín denne, sedem dni v týždni. Aplikácia by prijímala údaje z prídavného zariadenia umiestneného na hrudníku pacienta a následne by ich ďalej spracúvala a vyhodnocovala.

Prvá kapitola obsahuje formuláciu úlohy, kde stručne opisuje jednotlivé body zadávacieho listu diplomovej práce.

Druhá kapitola diplomovej práce analyzuje dostupné mobilné aplikácie z kategórie medicínskeho softvéru. Detailne opisuje jednotlivé medicínske aplikácie, zhodnocuje ich prínos, výhody, ale aj nevýhody. Ďalej definuje najrozšírenejšie mobilné platformy súčasnosti, pričom opisuje ich architektúru. Táto kapitola definuje viac platformový vývoj aplikácií a porovnáva vývojové prostredia, ktoré prevažne podporujú vývoj projektov pre viacero platforiem súčasne.

Jadro práce tvorí tretia kapitola, ktorá je venovaná podrobnému opisu prínosu práce. V úvode tejto kapitoly sú obsiahnuté funkčné požiadavky na systém, ako aj používateľské požiadavky pacienta a stručne opísaná plánovaná funkcionalita aplikácie. Nasledujúca podkapitola sa venuje návrhu aplikačného vybavenia, pričom analyzuje obmedzenia aplikácie, výber platformy a následne vývojového prostredia. Podkapitola Tvorba aplikácie poskytuje podrobný opis návrhu riešenia, analyzuje výber vhodnej metódy vývoja softvéru, rieši otázku komunikácie medzi zariadeniami, výber vhodnej knižnice pre vykresľovanie EKG, výber najvhodnejšej databázy, tvorbu testovacieho servera a samotné vyhodnocovanie dát s kritickou hodnotou.

V kapitole s názvom *Testovanie* je opísaný postup overenia riešenia implementovanej aplikácie dvoma metódami a zhrnuté závery testovania. Pre účely testovania bude aplikácia nainštalovaná na mobilné zariadenia obsahujúce rôzne verzie operačných systémov a hardvérové vybavenie. Ďalej budú pre aplikáciu vytvorené automatizované testy používateľského prostredia.

Ďalšia kapitola obsahuje zhodnotenie dosiahnutého riešenia a sústreďuje sa na prínosy implementovanej aplikácie, pričom v stručnosti opisuje jej výhody a nevýhody.

V závere práce je zhodnotené riešenie, prínos zadania diplomovej práce a návrh možných rozšírení aplikácie.

1 Formulácia úlohy

Medzi ciele tejto diplomovej práce patrí naštudovanie základov vývoja softvéru pre mobilné zariadenia a zhotovenie aplikácie vo forme funkčného prototypu. Diplomová práca sa opiera o nasledujúce body:

1. Analyzovať spôsob a vhodnosť využitia prenosných mobilných zariadení, ich HW a SW vybavenia, operačných systémov, vývojových prostriedkov a pod., pre spracovanie údajov indikujúcich kritický zdravotný stav pacienta.
2. Navrhnúť a na úrovni funkčného prototypu implementovať aplikáciu pre:
   (a) Vstup potrebných biometrických a riadiacich údajov a ich spracovanie.
   (b) Spoľahlivé vyhodnotenie a ďalšie spracovanie získaných, vstupných údajov pre účely ich aplikačného využitia (v zmysle témy DÚ) .
   (c) Spracovanie kritickej situácie pomocou aplikácie v mobilnom zariadení.
3. Systém má byť navrhnutý tak, aby:
   (a) Využil v maximálnej možnej miere bežné, komerčne dostupné, mobilné zariadenia.
   (b) Bol konfigurovateľný a jednoducho rozšíriteľný.
   (c) Nadväzoval na DÚ „Systém pre monitorovanie a rýchlu diagnostiku kritického stavu pacienta, s využitím mobilných zariadení“.
4. Prakticky overiť a otestovať výsledné riešenie.
5. Vypracovať dokumentáciu podľa pokynov vedúceho práce.

Prvý bod diplomovej práce je obsiahnutý v druhej kapitole s názvom *Analýza*, kde je zhodnotený súčasný stav medicínskych mobilných aplikácií a jednotlivých mobilných platforiem. Táto kapitola ďalej obsahuje charakteristiku architektúr troch najrozšírenejších operačných systémov pre inteligentné telefóny. V závere kapitoly

sú opísané vývojové prostredia pre multiplatformový vývoj aplikácií.
Druhý bod práce je popísaný v tretej kapitole s názvom *Návrh aplikácie pre spracovanie údajov určujúcich kritický stav pacienta* je podrobne opísaný návrh aplikácie pre prijímanie biometrických dát a ich ďalšie spracovanie. Táto kapitola je tiež zameraná na opis implementácie aplikácie do formy funkčného prototypu, ktorá je vyvíjaná pre dve platformy súčasne.
Aplikácia využíva potenciál dostupných mobilných zariadení a je v budúcnosti možné ju ďalej konfigurovať a rozširovať. Vzhľadom na to, že navrhnutá aplikácia prijíma dáta zo zariadenia Raspberry Pi, ktorých získanie a spracovanie je predmetom diplomovej práce s názvom *„Systém pre monitorovanie a rýchlu diagnostiku kritického stavu pacienta, s využitím mobilných zariadení“*, je splnený tretí bod diplomovej práce.
Štvrtý bod práce je zahrnutý v samostatnej kapitole s názvom *Testovanie* kde je prakticky overená funkčnosť navrhnutej aplikácie.
V prílohe práce sú zahrnuté príručky, ktoré predstavujú dokumentáciu k navrhnutej aplikácii, čím je splnený posledný bod zadávacieho listu diplomovej práce.

2 Analýza

Počas posledných desiatich rokov smartfóny radikálne zmenili mnoho aspektov každodenného života ľudí od nakupovania, cez finančnú sféru bankovníctva, zábavu, vzdelávanie či zdravotníctvo. A práve v tejto sfére je ich využitie ešte neprebádané, čo poskytuje možnosť vývoja medicínsky zameraných aplikácií, ktoré môžu nielen uľahčiť prácu lekárom, ale dokonca zachrániť život pacienta včasným odoslaním informácií o jeho zdravotnom stave, varovaním jeho blízkych alebo zavolaním záchrannej služby v prípade, že je pacient v ohrození života.

2.1 Analýza dostupných mobilných aplikácií

S inováciami digitálnych technológií ako sú cloudové služby, rôzne snímacie senzory a prídavné zariadenia, sa otvárajú možnosti využitia smartfónov v medicíne napríklad na monitorovanie zdravotného stavu pacienta vrátane snímania krvného tlaku, cukru v krvi, srdcovej frekvencie, počtu spálených kalórií, upozornenie pacienta na nedostatok pohybu, či príjmu tekutín, notifikovanie kedy má pacient užiť lieky, odosielanie zozbieraných dát lekárovi a podobne (1).

2.1.1 Medicínske aplikácie pre smartfóny

Ako príklad je možné uviesť aplikáciu iWander, navrhnutú pre platformu Android, ktorej úlohou je poskytovať bezpečnostné opatrenia pre ľudí s Alzheimerovou chorobou (2). Aplikácia funguje tak, že pomocou GPS v smartfóne sleduje pacienta za každých okolností. Na začiatku sú do softvéru uložené informácie o pacientovi, jeho vek, úroveň demencie a GPS súradnice jeho domova. Ak iWander zistí, že pacient je ďaleko od svojho domova (napríklad nezvyčajne neskoro v noci alebo za nepriaznivého počasia), algoritmus aplikácie vyhodnotí, že pacient môže byť zmätený a okamžite požaduje aby pacient ručne potvrdil svoj status. Neposkytnutie potvrde-

nia spustí alarm, ktorý upozorní blízku rodinu a obvodného lekára pacienta. Je však nutné podotknúť, že táto aplikácia má obmedzenia vo forme GPS a internetového pripojenia. Taktiež starší pacient s demenciou môže mať problém s využívaním moderných zariadení (3).

Iným príkladom využitia je používanie smartfónov pri rehabilitácii (4). V prípade, že pacient nie je schopný zúčastniť sa rehabilitácie priamo v zariadení na to určenom, je tu možnosť vykonávať cviky priamo z pohodlia jeho domova pripojením snímacieho senzora EKG na telo, ktoré komunikuje so smartfónom prostredníctvom Bluetooth, a tak umožňuje lekárom sledovať činnosť jeho srdca v reálnom čase zatiaľ čo pacient vykonáva cvičenie (5).

Ďalšou aplikáciou, ktorá využíva potenciál smartfónu spočíva v monitorovaní pacienta. Topánky vybavené senzormi, ktoré komunikujú s telefónom (6) je možné použiť na sledovanie úrovne pohybovej aktivity pacienta, ktorý v nedávnej dobe prekonal mozgovú príhodu.

Akcelerometrom, ktorý sa nachádza v smartfóne je tiež možné interpretovať chôdzu a rovnováhu pacienta (7), (8), (9), (10).

Elektrokardiografické senzory prepojené so smartfónom dokážu diagnostikovať a sledovať pokroky liečby spánkového apnoe, pričom poskytujú alternatívu k nákladnej a pracovne náročnej polysomnografii (11).

Pacienti s prvým stupňom diabetes sú práve tí, ktorým môže pokročilá technológia uľahčovať každodenný boj s chorobou, používaním aplikácie Diabeo (12). Táto aplikácia zbiera informácie ako pacientom nameraná hladina glukózy v krvi, počet prijatých sacharidov a plánovanú fyzickú aktivitu pred tým, než sa odporučí dávkovanie inzulínu.

2.1.2 Mobilné aplikácie pre pacientov so srdcovo-cievnymi ochoreniami

Srdcovo-cievne ochorenia sú v súčasnosti najčastejšou príčinou smrti v Európe. Každý rok zapríčinia viac ako 4 milióny úmrtí. Percentuálne až 47%, čo je tak-

mer polovica úmrtí (42% žien a 52% mužov) je spôsobených ischemickou chorobou srdca a infarktom, čo sú hlavné formy srdcovo-cievnych ochorení (13). V tabuľke 2 – 1 je možné vidieť percentuálne zastúpenie úmrtí mužov a žien na srdcovo-cievne choroby v jednotlivých krajinách Európy. Ischemická choroba srdca je sama o sebe

Tabuľka 2 – 1 Percentuálne zastúpenie úmrtí spôsobených srdcovo-cievnymi chorobami mužov a žien v Európe (14).

Krajina	**Úmrtie muži (%)**	**Krajina**	**Úmrtie ženy (%)**
Bulharsko	63	Bulharsko	72
Rumunsko	54	Rumunsko	68
Lotyšsko	50	Litva	66
Estónsko	48	Estónsko	63
Litva	47	Slovensko	61
Slovensko	47	Lotyšsko	58
Česká Republika	45	Česká Republika	56
Grécko	45	Maďarsko	55
Maďarsko	45	Poľsko	52
Poľsko	41	Grécko	49
Fínsko	39	Rakúsko	48
Švédsko	39	Slovinsko	46
Rakúsko	37	Nemecko	45
Nemecko	37	Fínsko	42
Cyprus	36	Taliansko	42
Luxembursko	35	Malta	42
Írsko	34	Švédsko	41
Taliansko	34	Cyprus	40
Malta	34	Luxembursko	38
Slovinsko	33	Belgicko	36

Krajina	**Úmrtie muži (%)**	**Krajina**	**Úmrtie ženy (%)**
Spojené Kráľovstvo	32	Portugalsko	36
Belgicko	30	Španielsko	35
Dánsko	30	Írsko	34
Holandsko	28	Dánsko	31
Portugalsko	27	Spojené Kráľovstvo	31
Španielsko	27	Holandsko	30
Francúzsko	25	Francúzsko	39

najčastejšou príčinou smrti v Európe a ročne je zodpovedná za 1,8 milióna úmrtí pozostávajúcich z 22% žien a 20% mužov (13). Z toho vyplýva, že každá piata žena a každý piaty muž umrie na následky tejto choroby.

V poradí druhou najčastejšou príčinou smrti je infarkt s takmer 1,1 miliónom úmrtí, čo predstavuje 15% žien a 10% mužov, v Európe ročne (13). To znamená, že v priemere každá siedma žena a každý desiaty muž zomrú na infarkt.

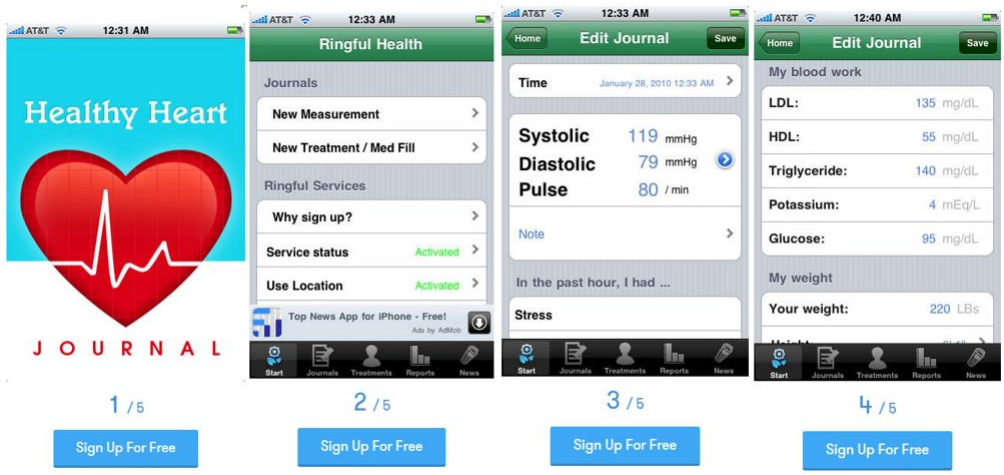
Štatistiky zostavené British Heart Foundation Health Promotion Research Group a Health Economics Research Centre poukazujú na kritickú situáciu tejto novodobej civilizačnej choroby. Ako opatrenia a možná predikcia týchto kritických stavov existujú rôzne mobilné aplikácie, ktoré sú primárne určené na zaznamenávanie informácií o srdcovej frekvencii, krvnom tlaku, užití liekov, odosielanie záznamu obvodnému lekárovi. Iné aplikácie využívajú kameru mobilných zariadení a vďaka oscilujúcemu obrazu dokážu zistiť srdcovú frekvenciu pacienta. Ako príklad je uvedených niekoľko najlepšie hodnotených aplikácií so zameraním na problematiku srdcovo-cievnych chorôb.

Healthy Heart 2

Táto aplikácia je vyvinutá pre platformu iOS a je voľne dostupná pre všetky mobilné

zariadenia od firmy Apple. Healthy Heart 2 poskytuje používateľovi možnosť zaznamenávať hodnotu krvného tlaku, srdcovej frekvencie, cholesterolu, krvnej glukózy, draslíka a informáciu o užití liekov 2 – 1. Tieto dáta sa zálohujú na Ringful online service, kde ich je možné ďalej analyzovať a zdieľať s lekármi a rodinnými príslušníkmi. Aplikácia ponúka podrobný prehľad nameraných hodnôt počas jednotlivých dní, čo je výhodou nielen pre používateľa, ale aj pre lekára, ktorý má takýmto spôsobom pacienta neustále pod kontrolou.

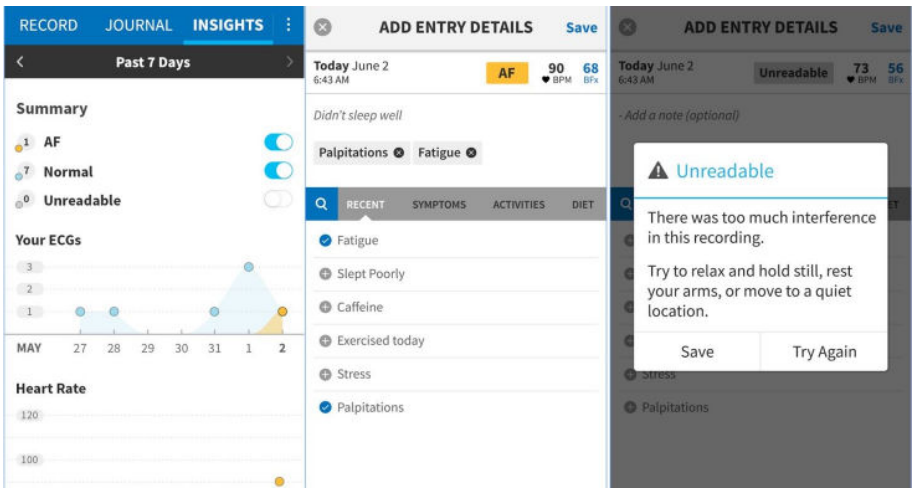
Nevýhodou aplikácie Healthy Heart 2 je nevyhnutnosť prídavných zariadení, ktoré však nekomunikujú s aplikáciou a všetky záznamy o meraniach je potrebné zadávať do systému ručne.



Obr. 2 – 1 Obrazovky aplikácie Healthy Heart 2 (15)

AliveECG

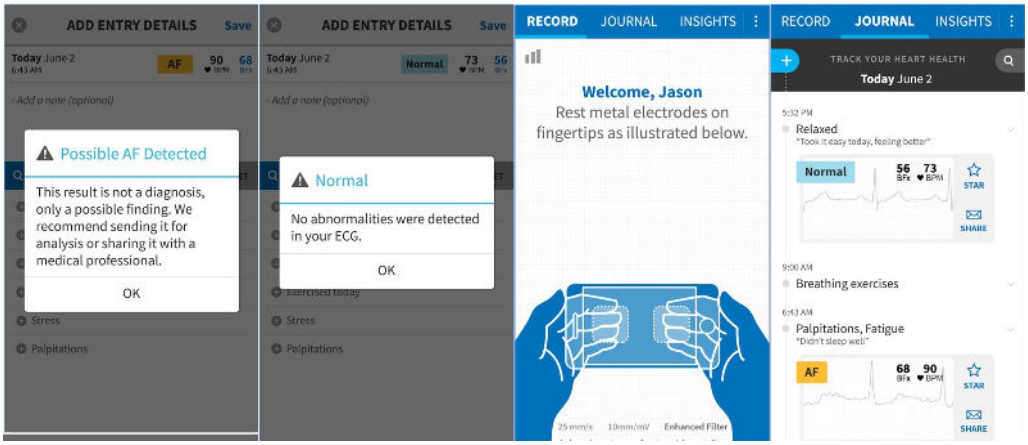
Aplikácia AliveECG je produktom firmy AliveCor Inc., ktorá sa zameriava na vývoj medicínskych aplikácií a je dostupná pre platformy iOS a Android. Táto aplikácia poskytuje analýzu, ukladanie a odosielanie nameraných výsledkov srdcovej aktiviy pomocou externého zariadenia s názvom AliveCor Heart Monitor, čo je EKG záznamník veľkosti bežného smartfónu. Používateľ položí dva prsty obidvoch rúk na zariadenie AliveCore Heart Monitor pre zaznamenanie EKG signálu 2 – 3, ktorý je



Obr. 2 – 2 Obrazovky aplikácie AliveECG (16).

bezdrôtovo prenášaný do aplikácie AliveECG 2 – 2.

Dve klinické štúdie uvádzajú, že AliveCore Heart Monitor a aplikácia AliveECG majú citlivosť nad 85% a špecifiskosť nad 90% pri identifikácií paroxymálnej fibrilácie predsiení (AF), na diagnostiku ktorej je táto aplikácia primárne určená (17). Výhodou tejto aplikácie je presnosť meraní, vďaka ktorému je klinicky preukázaná a schválená FDA. Ďalšou výhodou je prídavné zariadenie, ktoré samostatne komunikuje s aplikáciou (18).



Obr. 2 – 3 Obrazovky aplikácie AliveECG (16).

Aj napriek faktu, že samotná aplikácia je v Apple Store a Goole Play dostupná zadarmo, používateľ je nútený zakúpiť si zariadenie AliveCor Heart Monitor, ak ju chce využívať. Je taktiež nutné poznamenať, že spomínané zariadenie je k dispozícií len v USA, Austrálii, Indii, Írsku, Novom Zélande a vo Veľkej Británií (19).

eMotion ECG

Zariadenie eMotion ECG od firmy Mega Electronics Ltd je ďalšou aplikáciou monitorujúcou a vyhodnocovacou srdcovú frekvenciu pacienta. Pre správne fungovanie aplikácie je potrebné vlastniť prídavné zariadenie veľkosti zápalkovej škatuľky, ktoré obsahuje tri senzory pripojené k zariadeniu pomocou káblov ako je znázornené na obrázku 2 – 4. Tieto senzory sa pripevnia na hrudník pacienta a údaje o srdcovej frekvencii odosielajú pomocou bluetooth na aplikáciu v mobile (20).

Obr. 2 – 4 Aplikácia a prídavné zariadenie eMotion ECG (21).

Táto aplikácia spolu s prídavným zariadením ponúka pacientovi viacero možností:

- Zobrazuje prijaté údaje z prídavného zariadenia.
- Meria srdcovú frekvenciu v reálnom čase.

- Odosiela namerané hodnoty na webový server.
- Sleduje polohu pacienta.
- Upozorní pacienta na nezvyčajné namerané hodnoty.

Dáta sa odosielajú na webový server prostredníctvom mobilnej siete kde sú uložené a môžu byť ďalej spracované dvoma spôsobmi: buď pomocou webového prehliadača pre analytické účely alebo zobrazené na počítači v reálnom čase. Prostredníctvom webového prehliadača môže lekár analyzovať uložené dáta EKG a následne odoslať emailovú správu pacientovi. Pri prezeraní dát plynúcich v reálnom čase je možné pozorovať viacerých pacientov naraz. Je nutné poznamenať, že aplikácia eMotion ECG bola schválená FDA (22).
Aj napriek mnohým výhodám tejto aplikácie je stále jej slabou stránkou chýbajúca podpora pre mobilné platformy iOS a Windows Phone a iné. eMotion ECG ponúka podporu len pre platformu Android.

2.2 Analýza vhodných mobilných platforiem

Pred samotnou tvorbou aplikácie je potrebné špecifikovať, pre ktorú platformu bude aplikácia vyvíjaná. V súčasnosti na trhu dominujú tri svetové spoločnosti poskytujúce vlastné operačné systémy pre smartfóny a tablety a to sú, Google produkujúci Android, Apple a ich iOS a Microsoft vyvíjajuci Windows phone.
Podľa výskumov International Data Corporation (IDC) zaznamenal celosvetový trh so smartfónmi medziročný nárast o 13,0% v druhom štvrťroku 2015 s 341,5 miliónov zásielok. Tento nárast je predovšetkým spôsobený ziskami v rozvíjajúcich sa trhoch ako je APEJ a MEA.

Ako je možné vidieť v tabuľke 2 – 2, Android od spoločnosti Google dominuje na trhu s podielom 82,8% v období druhého štvrťroku 2015 (23). Jeho dominancia nad ostatnými mobilnými platformami spočíva v open source verzii (keďže jeho jadro

Tabuľka 2 – 2 Tabuľka zobrazujúca percentuálne zastúpenie predaja mobilných platforiem (23).

Obdobie	**Android** (%)	**iOS** (%)	**Windows Phone** (%)	**BlackBerryOS** (%)	**Iné** (%)
2015Q2	82,8	13,9	2,6	0,3	0,4
2014Q2	84,8	11,6	2,5	0,5	0,7
2013Q2	79,8	12,9	3,4	2,8	1,2
2012Q2	69,3	16,6	3,1	4,9	6,1

je založené na jadre operačného systému Linux), ktorá je výhodná pre výrobcov smartfónov tým, že eliminuje potrebu vývoja vlastného operačného systému, a tak sa výrobcovia snažia od seba odlíšiť aspoň vlastnou grafickou nadstavbou ako napríklad HTCSense od firmy HTC, či TouchWiz od spoločnosti Samsung. Ďalšou výhodou androidu je kompatibilita so širokým spektrom hardvéru, čo ho stavia na prvé miesto vo svete mobilných operačných systémov.

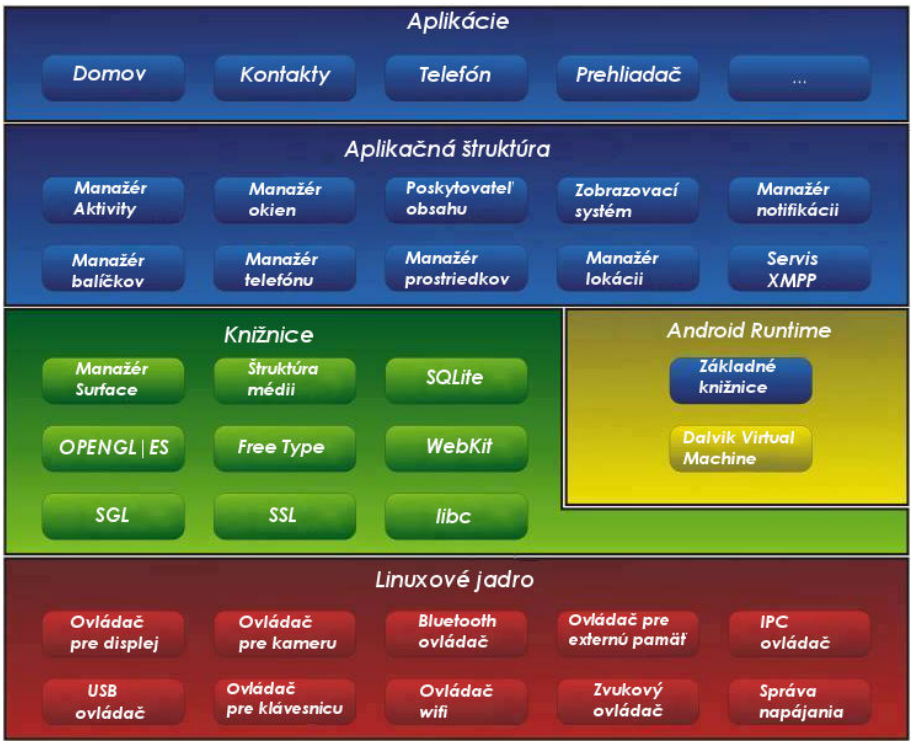
Podiel zastúpenia operačného systému iOS, ktorý zastrešuje firma Apple, na trhu poklesol o 22,3% medzi kvartálne s počtom zásielok 47,5 miliónov (23). Aj napriek sezónnemu poklesu predaja si Apple udržiava priazeň spotrebiteľov vďaka zariadeniam s väčšími displejmi. Obľúbenosť produktu iPhone 6 Plus pokračuje na mnohých trhoch vrátane Číny, kde celkový predaj týchto smartfónov vzrástol o 6,7 % (23).

Medzi kvartálny pokles operačného systému od Microsoftu dosiahol 4,2%, kde celkový počet predaných softvérových produktov v tomto štvrťroku predstavoval 8,8 milióna (23). Spoločnosť Microsoft od roku 2014 (v čase akvizície spoločnosti Nokia) pozmenil celé produktové portfólio a uviedol na trh Microsoft Lumia zariadenia.

Vzhľadom na najväčšie zastúpenie na trhu boli vybrané tri mobilné platformy a to Android, iOS a Windows Phone, ktoré sú detailne opísané v nasledujúcich podkapitolách.

2.2.1 Operačný systém Android

Ako už bolo vyššie spomenuté, Android je rozsiahla open-source platforma vytvorená pre mobilné zariadenia ako sú smartfóny, tablety, PDA, navigácie. Ako je možné vidieť na obrázku 2 – 5 architektúra operačného systému Android pozostáva z operačného systému, ktorého jadro tvorí základ operačného systému Linux, middleware vrstvy, aplikačnej štruktúry a veľkého množstva aplikácií.



Obr. 2 – 5 Architektúra Android platformy

Prvú časť architektúry tvoria dve skupiny aplikácií. Natívne Android aplikácie, ktoré sa nachádzajú v mobilnom zariadení už pri prvom spustení ako napríklad kalendár, kontakty alebo internetový prehliadač, a aplikácie tretích strán, ktoré si používateľ môže dodatočne stiahnuť a nainštalovať.

Aplikačná štruktúra je časť, ktorá zodpovedá za spravovanie a plynulý beh Android aplikácií. Obsahuje súbor základných služieb, ktoré sú potrebné pre vývoj každej aplikácie. Dostupnosť týchto služieb uľahčuje prácu vývojárom, ktorí sa tak môžu

sústrediť výhradne na programovanie vlastnej aplikácie.
Aplikačná štruktúra zahŕňa tieto služby:

- Manažér aktivít - Riadi všetky aspekty životného cyklu aplikácie.
- Poskytovateľ obsahu - Umožňuje aplikáciám navzájom zdieľať dáta.
- Manažér oznámení - Umožňuje aplikáciám zobrazenie a nastavenie upozornení, notifikácií a správ.
- Manažér zdrojov - Poskytuje prístup k nastaveniu farieb či rozvrhnutiu používateľského rozhrania.
- Zobrazovací systém - Rozšíriteľný súbor zobrazení použitých na vytvorenie používateľského rozhrania aplikácie.
- Manažér balíčkov - Pomocou balíčkov sú aplikácie schopné zistiť informácie o aktuálne nainštalovaných aplikáciách v zariadení.
- Manažér telefónu - Poskytuje aplikácii informácie o tom, aké telefónne služby sú na zariadení k dispozícii.
- Manažér polohy - Umožňuje prístup k lokalizačným službám pričom dovoľuje aplikácii prijímať aktuálne informácie o zmene polohy.

Prostredníctvom Aplikačnej štruktúry vývojári pristupujú k tretej časti architektúry OS Android, ktorá obsahuje sadu knižníc v jazyku C a C++, ktoré sú používané komponentmi systému.

- Multimediálna knižnica - Umožňuje prehrávanie audio a video súborov.
- Povrchový manažér - Vytvára grafické zložky skladaním jednotlivých 2D a 3D zložiek.
- OpenGL - Umožňuje tvorbu aplikácií využívajúcich trojrozmernú grafiku v reálnom čase.

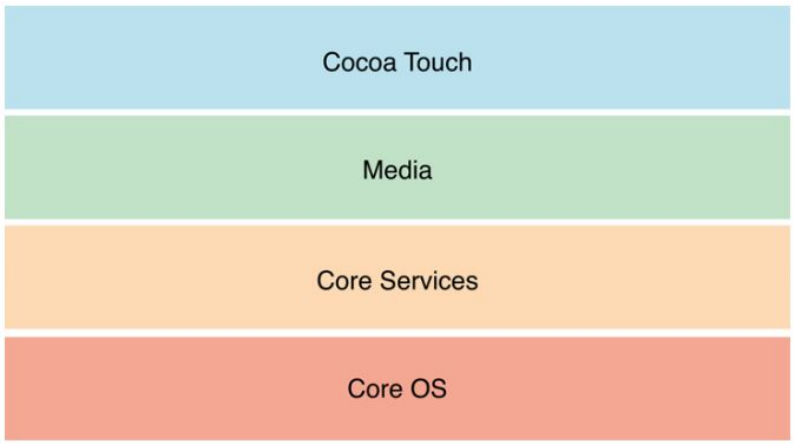
- SGL - Engine pre dvojrozmernú grafiku.
- FreeType - Umožňuje vykresľovanie bitmapových a vektorových formátov písma.
- SQLite - Poskytuje relačný databázový systém dostupný pre všetky aplikácie.

Štvrtú časť tvorí vykonávacie prostredie Android, ktoré sa skladá zo súboru základných knižníc a virtuálneho stroja Dalvik. Keďže aplikácie používajú programovací jazyk Java, je potrebný preklad programu do strojového kódu, na ktorý sa miesto Java Virtual Machine používa Dalvik. DVM je registrovo orientovaná architektúra, ktorá využíva vlastnosti linuxového jadra, ako je správa pamäte (24).
Poslednou časťou je Linuxové jadro, ktoré tvorí abstraktnú vrstvu medzi hadrvérom a softvérom, vďaka čomu je Android nezávislý na rôznych hardvérových zariadeniach. Hlavnou súčasťou jadra sú ovládače, ktoré zabezpečujú komunikáciu medzi hardvérom a softvérom, ako napríklad ovládač WiFi, fotoaparátu, Bluetooth, zvuku, atď. Ďalšou úlohou je správa procesov, napájania, pamäti a pripojenie sietí.

2.2.2 Operačný systém iOS

Architektúra iOS vychádza z architektúry operačného systému Mac OS X, používaného v počítačoch spoločnosti Apple. Vyznačuje sa veľkou stabilitou a jedinečným rozhraním oproti konkurenčným systémom, za čo vďačí operačnému systému UNIX, na ktorého jadre je postavená. Model architektúry tvorí päť vrstiev (na obrázku 2 – 6 je možné vidieť iba štyri vrstvy, pretože prvá vrstva, ktorú tvoria aplikácie sa nezvykne vyznačovať), ktoré umožňujú základnú funkčnosť a poskytujú vývojárom potrebné aplikačné rozhranie a frameworky pre vývoj aplikácií.

Prvú vrstvu tvoria natívne aplikácie a aplikácie tretích strán. Do prvej skupiny patria aplikácie, ktoré telefón pri jeho prvom zapnutí obsahuje ako sú kontakty, správy, kalkulačka, apod. Aplikácie tretích strán sú všetky aplikácie dostupné v obchode App Store, ktoré si používateľ môže nainštalovať. Pre túto platformu je však väčšina aplikácií tretích strán spoplatnená.



Obr. 2 – 6 Architektúra operačného systému iOS (25)

Cocoa Touch vrstva obsahuje všetky potrebné frameworky pre správne fungovanie grafického rozhrania systému a aplikácií. Hlavnou súčasťou tejto vrstvy je zobrazovací framework UIKit, ktorý zodpovedá za vykresľovanie aplikácií napísaných v jazyku Objective C, ako aj za rozhranie pre dotykové ovládanie aplikácií a rozpoznávanie gest.
Vrstva médií obsahuje audio, video a grafické nástroje pre správne spracovanie multimediálneho obsahu, čo umožňuje plynulé prehrávanie videí a zvuku.
Vrstva médií obsahuje tieto grafické technológie:

- Core Graphics - Umožňuje vykresľovanie 2D vektorov a renderovanie obrázkov.
- Core Animation - Poskytuje pokročilú podporu pre animácie.
- OpenGL ES - Knižnica pre hardvérovo-akcelerované vykresľovanie trojrozmerných objektov.
- Core Text - Umožňuje vykresľovanie textu.
- Image I/O - Slúži na čítanie a zápis grafických formátov.

Vrstva médií obsahuje tieto zvukové technológie:

- OpenAL - Poskytuje multiplatfomové rozhrania pre trojrozmerný zvuk.

- AV Foundation - Poskytuje sadu rozhraní Objective C pre správu záznamu zvuku a jeho prehrávania.
- The Media Player Framework - Umožňuje prístup ku knižnici iTunes a prehrávanie skladieb.
- Core Audio Framework - Zodpovedá za systémové zvuky, upozornenia, vibrovanie a umožňuje prehrávať viackanálový zvuk.

Vrstva médií obsahuje tieto video technológie:

- Media Player Framework - Umožňuje celo obrazovkové, ale aj čiastočné prehrávanie videí.
- AV Foundation - Poskytuje sadu rozhraní Objective C pre správu záznamu zvuku a jeho prehrávania.
- Core Media - Zahŕňa nízko-úrovňové funkcie, na ktorých je postavená väčšina technológií.

Vrstva služieb obsahuje systémové služby, ktoré zodpovedajú za správnu funkciu a inštaláciu aplikácií, databázové nástroje, zabezpečenie zariadení, cloudové služby a nástroje pre spracovávanie skriptov.
Posledná vrstva zodpovedá za správnu komunikáciu medzi hardvérovou a softvérovou časťou zariadenia. Obsahuje ovládače pre sieťové pripojenie, grafiku, procesor apod.

2.2.3 Operačný systém Windows Phone

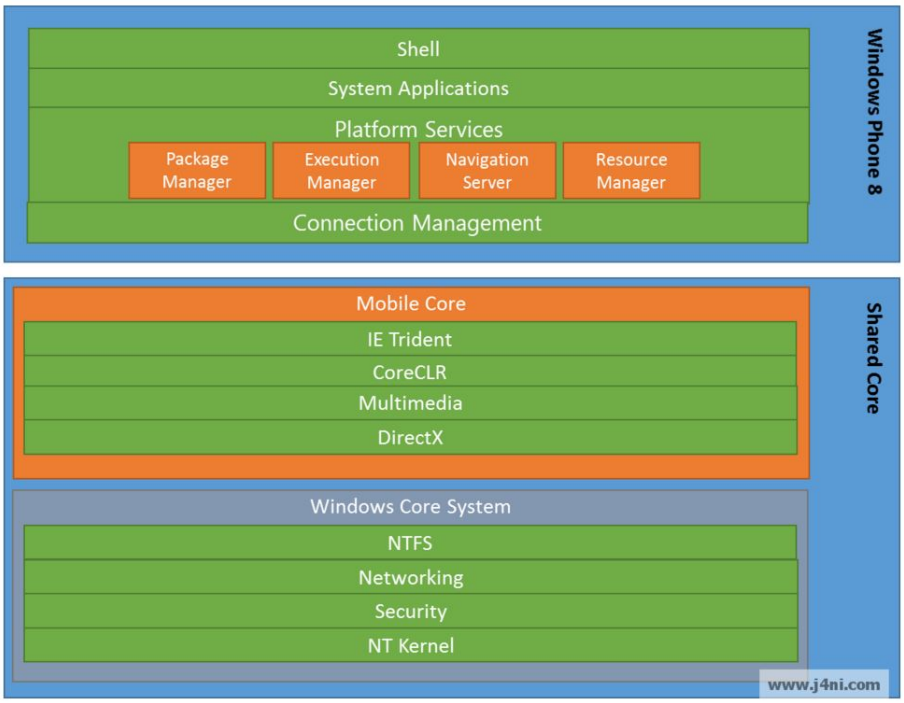
Windows Phone je pomerne mladý operačný systém, ktorý je v súčasnej verzii 8 postavený na architektúre Windows NT narozdiel od svojho predchodcu Windows Phone 7, ktorý používal architektúru Windows CE.
Zdieľané jadro sa skladá z dvoch samostatných častí, jadra hlavného systému Windows a mobilného jadra. Jadro systému Windows je tá časť, ktorá skutočne zdieľa kód s Windowsom a je hlavnou zložkou jadra operačného systému. Jadro systému a

mobilné jadro sú jediné oblasti, v ktorých sa zdieľa kód medzi dvoma platformami. Existujú oblasti s rovnakým rozhraním API, ale kód sa medzi Windowsom a Windows Phone líši. Architektúra operačného systému pozostáva zo štyroch hlavných vrstiev 2 – 7, ktoré poskytujú služby pre všetky aplikácie (26).

Manažér balíčkov je vrstva, ktorá spracováva životný cyklus aplikácie od inštalovania až po odinštalovanie aplikácie, a správu metadát počas inštalácie. Tieto dáta obsahujú informácie o možnosti pripnutia aplikácie na hlavnú obrazovku a rozšíriteľnosti aplikácie.

Manažér vykonávania spracováva aplikácie na pozadí systému a vykonáva ich spúšťanie, vypínanie a deaktiváciu.

Navigačný server zodpovedá za presúvanie aplikácií do popredia. Oznámi manažérovi vykonávania akú aplikáciu má zapnúť, alebo aktivovať vybraním z navigačného zásobníka.



Obr. 2 – 7 Architektúra operačného systému Windows Phone (28).

Správca prostriedkov monitoruje systémové prostriedky všetkých aktívnych proce-

sov. V prípade, že aplikácia nepracuje správne, správca prostriedkov ju môže ukončiť s cieľom zabezpečiť stabilnú a rýchlu odozvu telefónu.

2.3 Analýza vývojových prostredí

Táto podkapitola ponúkne podrobný prehľad vývojových prostredí pre Android, Windows Phone a iOS, a taktiež opis multiplatforiem, teda vývojových prostredí respektíve frameworkov pre návrh aplikácie kompatibilnej s viacerými platformami (27).

2.3.1 Xamarin

Xamarin je framework zameraný na multiplatformové vyvíjanie mobilných aplikácií pre iOS, Android a Windows Phone, ktorý ponúka možnosť práce nielen vo vlastnom vývojovom prostredí Xamarin Studio, ktoré je dostupné pre Windows a Mac OSX, ale aj vo forme plugin-u do známeho vývojového prostredia Visual Studio (29). Základným stavebným kameňom frameworku Xamarin je programovací jazyk C#, v ktorom je používateľovi umožnené vytvárať aplikácie pre všetky vyššie spomenuté platformy.
Stručný súhrn kľúčových bodov pre vyvíjanie multiplatformových aplikácií v prostredí Xamarin:

- Použitie C# - Xamarin umožňuje v tomto jazyku napísať kód pre Windows Phone, ktorý môže byť veľmi ľahko portovaný na iOS a Android.
- Použitie MVC - Pre vývoj aplikácií je používaná architektúra MVC (Model/View/Controller), kde môže vývojár určiť, ktoré časti aplikácie budú používať natívne prvky používateľského rozhrania z každej platformy (iOS, Android, Windows Phone).

- Kompilovanie natívnych UI - Každý operačný systém poskytuje inú vrstvu používateľského rozhrania pre svoje aplikácie (implementované v jazyku C# s pomocou natívnych návrhových nástrojov UI):

  – Pre iOS sa na vytváranie natívne vyzerajúcich aplikácií používa MonoTouch.UIKit API, prípadne Xamarin iOS designer.

  – Aby si Androidové aplikácie zachovali špecifické používateľské rozhranie, používa sa Android.Views s využitím Xamarin UI dizajneru.

  – Windows Phone používa XAML/Silverlight prezentačnú vrstvu prostredníctvom Visual Studia alebo Blend UI dizajneru.

Na obrázku 2 – 8 je možné vidieť porovnanie možností vývojového prostredia Xamarin IDE pre MAC OS X a Windows na vývoj mobilných aplikácií pre tri najrozšírenejšie mobilné platformy. Xamarin poskytuje možnosti zvýšenia kvantity zdie-

Development OS	Mac OS X	Windows	
IDE	Xamarin Studio	Xamarin Studio	Visual Studio
iOS	Y	-	Y
Android	Y	Y	Y
Windows Phone	-	-	Y

Obr. 2 – 8 Porovnanie možností Xamarin IDE na operačných systémoch MAC OS X a Windows

ľaného kódu použitím týchto multiplatformových komponentov, ktoré poskytujú bežné služby vo všetkých systémoch:

- SQLite-NET pre lokálne úložisko SQL.
- Xamarin.Mobile pre prístup k službám špecifickým pre konkrétne zariadenia

vrátane fotoaparátu, kontaktov, geolokácie a pod.

- Použitie frameworkov pre prístup k sieti, webovým službám, vypínaniu a zapínaniu zariadenia a pod.

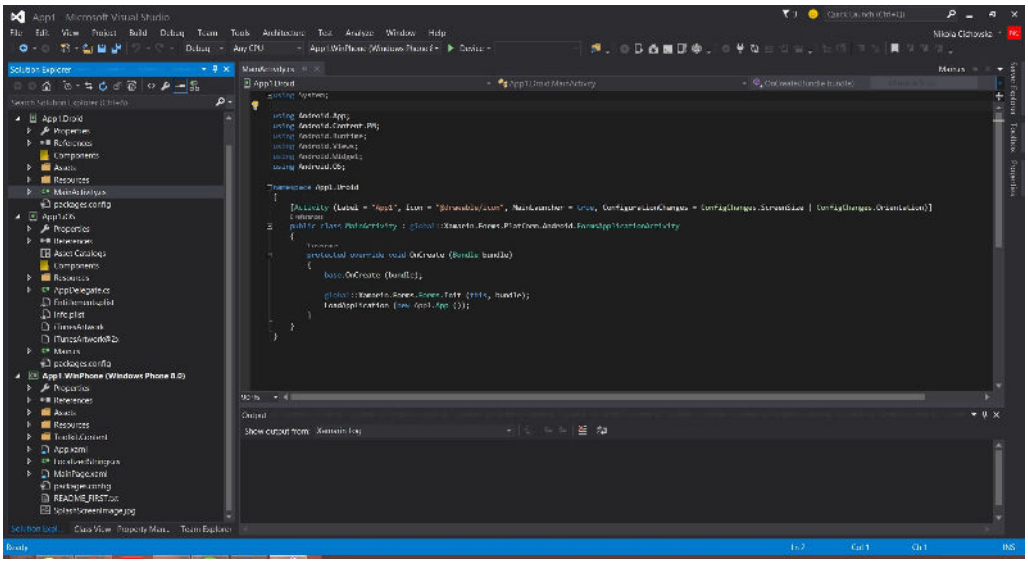
Platforma Xamarin pozostáva zo štyroch základných vrstiev, ktoré umožňujú vývoj aplikácií pre iOS a Android:

- Jazyk C# - Umožňuje používať sofistikované funkcie ako generické typy a lambda výrazy.
- Mono.NET framework - Poskytuje implementáciu multiplatformovo rozsiahlych funkcií v rámci .NET spoločnosti Microsoft.
- Kompilátor - V závislosti na platforme, vytvára natívnu aplikáciu (iOS) alebo integrované .NET aplikácie (Android).
- IDE nástroje - Xamarin ponúka vývojové prostredie Xamarin Studio IDE a Xamarin plug-in pre Visual Studio, kde je možné vytvárať projekty Xamarin.

Kompilácia aplikácií vytvorených v programovacom jazyku C# do natívnych aplikácií prebieha na každej platforme veľmi odlišne:

- iOS - C# je kompilovaný pomocou ahead-of-time compilation do ARM strojového jazyka. Úlohou .NET frameworku je odstránenie nevyužitých tried počas linkovania so zámerom zníženia veľkosti aplikácie.Keďže firma Apple neumožňuje vykonávanie generovania kódu na iOS, tak niektoré jazykové funkcie nie sú k dispozícií.
- Android - C# je kompilovaný na IL, pričom sú vytvárané balíčky pomocou MonoVM a JITing. Nevyužité triedy vo frameworku sú počas linkovania odstránené. Aplikácia beží bok po boku s JAVA/ART a spolupracuje s natívnymi typmi cez JNI.
- Windows Phne - C# je kompilovaný na IL pričom nevyžaduje nástroje Xama-

rin. Návrh Windows Phone aplikácií v prostredí Xamarin uľahčuje migrovanie kódu na iOS a Android.



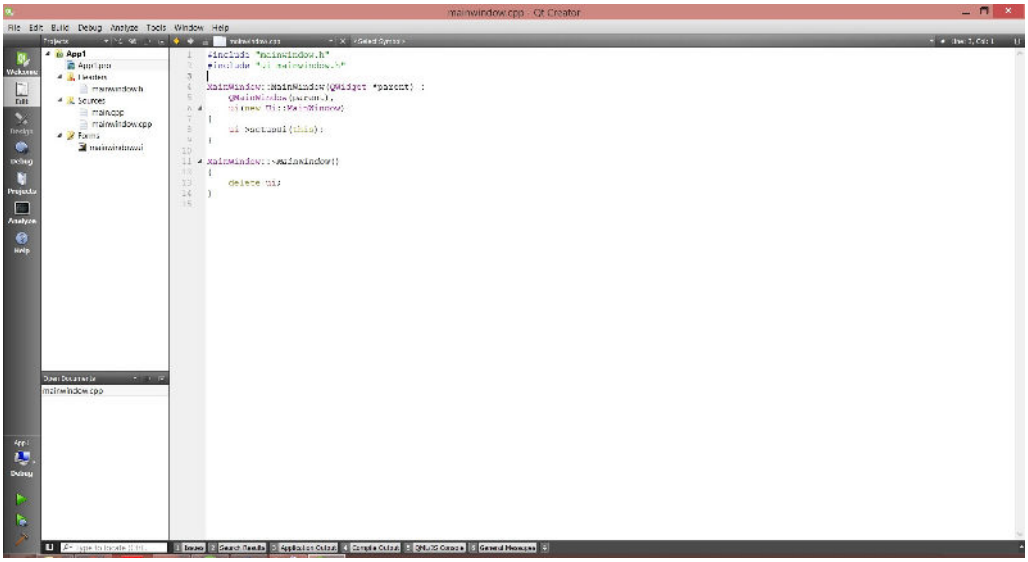
Obr. 2 – 9 Xamarin rozšírenie pre Visual Studio

2.3.2 Qt

Qt je viac platformový aplikačný framework používaný pre vývoj softvéru, ktorý môže bežať na rôznych softvérových a hardvérových platformách s malými alebo žiadnymi zmenami zdrojového kódu. Je primárne určený pre vývoj aplikačného softvéru s grafickým používateľským prostredím, pričom je možné vyvíjať aj programy bez GUI ako napríklad konzolové aplikácie. Qt používa štandard C++ s rozšíreniami zahŕňajúcimi signály a sloty, ktoré zjednodušujú manipuláciu s udalosťami, čím napomáhajú vývoju nielen GUI, ale aj serverových aplikácií. Tento framework je v súčasnosti vyvíjaný ako Qt Company, dcérskej spoločnosti Digia, ktorá vlastní ochrannú známku a copyright Qt, a ako Qt Project v rámci open-source, zahŕňajúc individuálnych vývojárov (30).

Výhodou tohto frameworku je open-source verzia, ktorá je voľne dostupná a jej funkcionalita nie je časovo obmedzená. Orientácia vo vývojovom prostredí je však

obťažnejšia a ďalšou nevýhodou je samotný jazyk frameworku, keďže vzhľadom k potrebám požadovanej aplikácie je efektívnejšie ju vyvíjať v jazyku C# alebo Java.



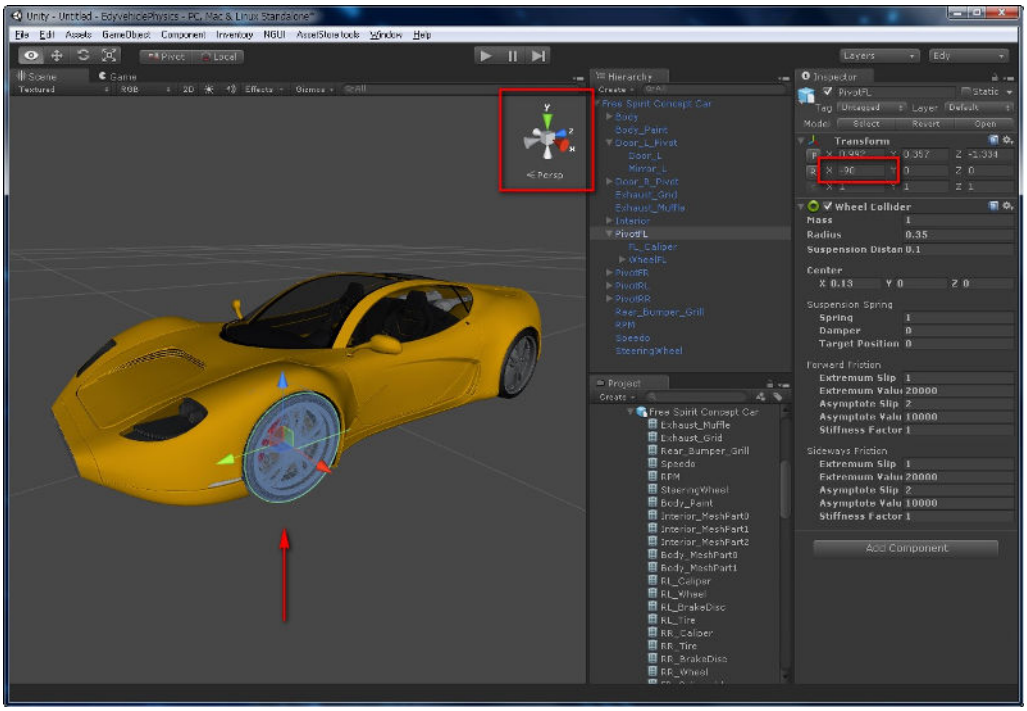
Obr. 2 – 10 Ukážka projektu vo vývojovom prostredí Qt Creator

2.3.3 Unity

Unity je multiplatfomový herný engine vyvíjaný spoločnosťou Unity Technologies a používaný najmä na vývoj video hier pre desktopové počítače, herné konzoly, mobilné zariadenia a webové stránky. Používaným štandardom je C#. Medzi podporované platformy tohto enginu patrí Android, Windows Phone 8, iOS, BlackBerry 10, Unity Web Player, PlayStation 3, PlayStation 4, PlayStation Vita, Xbox 360, Xbox One, Wii U, Nintendo 3DS a Wii (32). Unity je voľne dostupný ako samostatný editor, ale aj ako rozšírenie do Visual Studia.

Ako výhodu frameworku Unity je nutné spomenúť širokú škálu grafických možností, dostupnosť a programovací jazyk tohto prostredia. Samotné vývojové prostredie aj napriek tomu, že ponúka širokú škálu funkcionalít, je primárne určené pre vývoj vi-

deo hier a k tomu má aj prispôsobené samotné API. Vzhľadom na potreby vyvíjanej aplikácie, ktorá nebude obsahovať pokročilé grafické prvky je použitie vývojového prostredia Unity zbytočné.



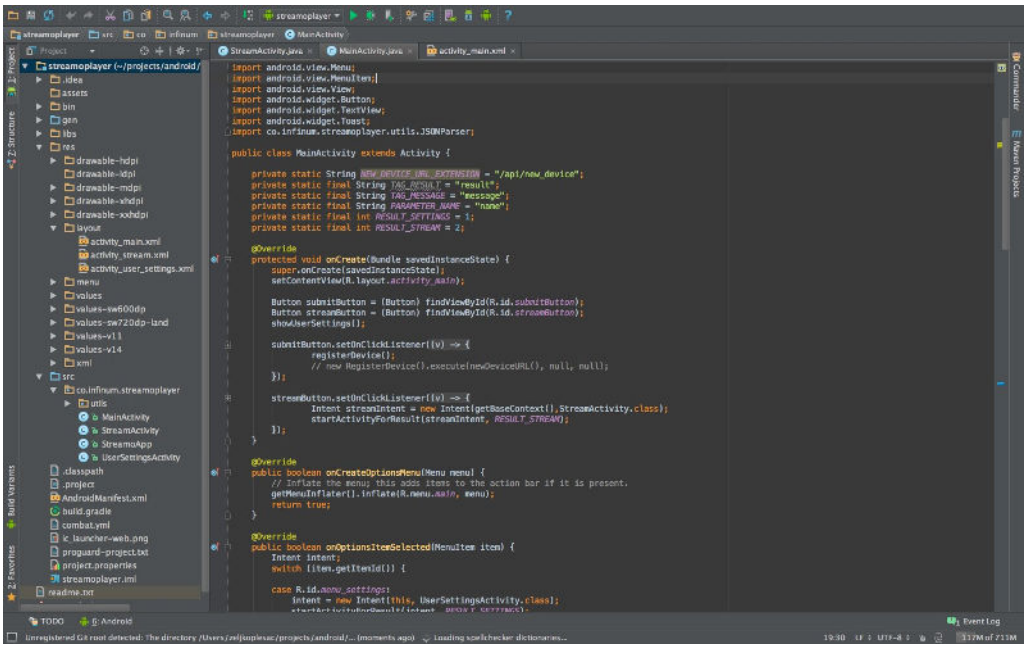
Obr. 2 – 11 Ukážka projektu vo vývojovom prostredí Unity Editor (33)

2.3.4 Android Studio

Android Studio je voľne dostupné oficiálne vývojové prostredie pre vývoj Android aplikácií, ktoré vychádza z IntelliJ IDEA. Je dostupné na stiahnutie pre platformy Windows, Mac OS X a Linux (31) a nahradil Eclipse Android Development Tools ako primárne vývojové prostredie pre natívny vývoj androidových aplikácií. Android Studio ponúka širokú škálu funkcionalít ako napríklad šablóny kódu, bohaté layouty s podporou pre drag and drop editáciu, vstavanú podporu pre Google Cloud Platform, čo uľahčuje integráciu Google Cloud Messaging atď (34).

Android Studio je narozdiel od vyššie spomínaných frameworkov zamerané na vývoj jedinej platformy, čo môže byť na jednej strane nevýhodou z hľadiska oslovenia len

jednej cieľovej skupiny, a to používajúcej smartfóny s operačným systémom Android, na druhej strane však výhodou z pohľadu na rozmanitosť natívnych API. Ďalšou výhodou je rozsiahlosť a prístupnosť k dokumentácii nástroja Android Studio, ktorá môže značne prispieť k rýchlosti osvojenia si práce v tomto vývojovom prostredí.



Obr. 2 – 12 Ukážka projektu vo vývojovom prostredí Unity Editor

Po naštudovaní práce s vyššie spomenutými frameworkmi a vývojovými prostrediami a ich následnom otestovaní sa rozhodlo, že tvorba aplikácie pre mobilné zariadenia bude realizovaná pre mobilné platformy iOS, Windows Phone a Android vo vývojovom prostredí Visual Studio s plug-inom Xamarin. K rozhodnutiu prispela dostupnosť rozsiahlej dokumentácie od spoločnosti Xamarin, poskytnutie študentskej licencie na rok zdarma a podpora developerského tímu Xamarin po dobu platnosti licencie.

Xamarin bol nainštalovaný ako plug-in do vývojového prostredia Visual Studio Enterprise 2015, kde bol vytvorený jednoduchý testovací projekt, pričom práca s týmto multiplatformovým frameworkom bola pomerne jednoduchá a efektívna. Pre testovanie aplikácie bolo potrebné pripojiť zariadenie alebo stiahnuť a nainštalovať

emulátor jednotlivých smartfónov. Je možné povedať, že tento framework splnil kritéria pre vývoj aplikácie, pričom by mohla byť spustiteľná na viacerých platformách. Jedinou nevýhodou môže byť jeho cena, keďže zadarmo je k dispozícií len trial verzia, po ktorej vypršaní platnosti nie je možné ďalej pokračovať vo vývoji projektu. Spoločnosť Xamarin však ponúka ročnú licenciu zdarma pre študentov technických vysokých škôl. Pre získanie tejto licencie je nutná registrácia na portály DreamSpark.com, kde sa zadaním identifikačného čísla študentskej karty overí, či je žiadateľ naozaj študentom.

3 Návrh aplikácie pre spracovanie údajov určujúcich kritický stav pacienta

Využívanie mobilných zariadení v medicíne by mohlo nielen uľahčiť prácu lekárom, ale aj poskytnúť väčší komfort pacientom. S prihliadnutím na fakt, že v dnešnej dobe takmer každý človek vlastní aspoň jeden mobilný telefón, sa čím ďalej, tým viac stáva táto myšlienka uskutočniteľnou. Mobilné zariadenia by mohli fungovať ako akýsi strážca napríklad pacienta so srdcovým ochorením, pričom by sa informácie o jeho srdcovej činnosti odosielali v istých časových intervaloch do databázy lekára, ktorý by tak mal pacienta neustále pod kontrolou. Pacient by mal k dispozícii aplikáciu, ktorá by monitorovala jeho zdravotný stav a neobmedzovala by ho v jeho každodenných aktivitách a jej používanie by znížilo frekvenciu EKG vyšetrení u lekára, čím by pacient ušetril veľa času nielen pri samotnom vyšetrení, ale aj času strávenom v čakárni. Takáto aplikácia by mohla obsahovať množstvo rôznych funkcionalít, ktoré by mu v prípade život ohrozujúcom stave mohli zachrániť život. Aplikácia by mohla časom úplne nahradiť tlakový holter, čo je prístroj na meranie krvného tlaku, ktorý sa dáva pacientom na dobu 24 hodín pre monitorovanie ich srdcovej aktivity. Jeho nevýhodou však je, že narozdiel od spomenutej aplikácie nedokáže sám signalizovať riziko zlyhania srdca ani následne vyhľadať pomoc.

3.1 Analýza požiadaviek

Mnohé firmy zaoberajúce sa vývojom softvéru najprv dohodnú so zákazníkom súbor požiadaviek a špecifikácií aby si utvorili predstavu o tom, čo od nich zákazník očakáva. V tejto diplomovej práci bolo prvým krokom získanie súboru požiadaviek na softvér, čo bude slúžiť ako odrazový mostík pri jeho návrhu a samotnej realizácii produktu, pričom je možné požiadavky meniť a dopĺňať počas vývoja aplikácie. Nasledujúce tabuľky obsahujú funkčné požiadavky na systém 3 – 1, ako aj používateľské

požiadavky pacienta 3 – 2.

Tabuľka 3 – 1 Zoznam funkčných požiadaviek na systém.

ID	**Definícia**
FP-1	Nadviazanie spojenia so zariadením Raspberry Pi pomocou Bluetooth.
FP-2	Vyžiadanie pridania telefónnych čísel na odosielanie sms správ so súradnicami pacienta.
FP-3	Vyžiadanie pridania názvu servera lekára pre odosielanie dát.
FP-4	Získavanie údajov zo zariadenia Raspberry Pi pomocou Bluetooth.
FP-5	Odosielanie údajov na databázový server.
FP-6	Zobrazenie základných informácií o pacientovi prijatých zo zariadenia Raspberry Pi na obrazovke.
FP-7	Vykresľovanie prijatých EKG údajov.
FP-8	Zistenie kritickej situácie pacienta z prijatých dát.
FP-9	Vytočenie čísla záchrannej služby v prípade zistenia kritických EKG údajov.
FP-10	Odoslanie SMS správ na čísla uložené v aplikácii.
FP-11	Zapnutie automatického alarmu po zistení odchýlky od normálnych hodnôt.
FP-12	Zrušenie automatického alarmu po zistení odchýlky od normálnych hodnôt.

Tabuľka 3 – 2 Zoznam funkčných požiadaviek na používateľa/pacienta.

ID	Definícia
FP-1	Uloženie telefónnych čísel do databázy aplikácie.
FP-2	Možnosť pridávania, úpravy a mazania telefónnych čísel z databázy pomocou používateľského rozhrania aplikácie.
FP-3	Uloženie adresy servera lekára, na ktorý sa budú prijaté údaje odosielať.
FP-4	Možnosť zmeny adresy servera pri zmene lekára.

3.2 Plánovaná funkcionalita práce

Predmetom tejto diplomovej práce je navrhnúť a implementovať aplikáciu, ktorá bude spustiteľná na mobilných zariadeniach. Táto aplikácia bude komunikovať so zariadením Raspberry Pi, na ktorom budú pripojené senzory na snímanie krvného tlaku pacienta. Získanie údajov s informáciou o srdcovej frekvencii pacienta a návrh algoritmu na ich spracovanie je predmetom inej diplomovej práce (36).

Najprv sa uvažovalo, že po zapnutí aplikácie by mala pacienta uvítať hlavná obrazovka, kde dostane na výber či chce aplikáciu používať ako pacient, alebo lekár. Po zvolení role pacienta aplikácia požaduje jeho registráciu, kde bude nutné uviesť meno, priezvisko, rodné číslo a heslo. V prípade, že pacient už má vytvorený účet, zvolí druhú ponúkanú možnosť, ktorou bude prihlásenie pomocou rodného čísla a hesla.

Po absolvovaní registrácie respektíve prihlásenia, bude pacientovi sprístupnené zistenie zariadení, s ktorými je jeho smartfón spárovaný a ponúkne sa mu možnosť voľby, ku ktorému zariadeniu sa pripojí. Predpokladá sa, že pacient zvolí zariadenie Raspberry Pi, na komunikáciu s ktorým je aplikácia v jeho mobilnom zariadení určená.

Nakoniec sa však rozhodlo, že mobilná aplikácia nebude disponovať registračnou a prihlasovacou databázou. Nebude potrebné pridávať možnosť pre prihlásenie lekára z dôvodu, že všetky potrebné informácie o pacientovi budú uložené v zariadení

Raspebrry Pi, ktoré bude obsahovať konfiguračný softvér pre ukladanie základných informácií o pacientovi, ako aj jeho anamnézy, alergie a všetko potrebné čo môže uľahčiť lekárom záchranu života pacienta. Taktiež sa uvažovalo, že aplikácia v mobilnom zariadení bude slúžiť len pre jedného pacienta vzhľadom na to, že v dnešnej dobe je každý človek majiteľom aspoň jedného mobilného zariadenia.
Akonáhle nadviaže mobilné zariadenie spojenie so zariadením Raspberry Pi, aplikácia spustí neustále prijímanie dát, ktoré následne začne vykresľovať na obrazovku vo forme EKG grafu. Pacient tak bude mať k dispozícii vlastný EKG prístroj priamo vo svojom smartfóne.
Vyvíjaná aplikácia bude však prijímať nielen samotnú EKG hodnotu, ale aj druhú hodnotu signalizujúcu kritický stav pacienta. V prípade, že externé zariadenie zistí odchýlky EKG od normálnych hodnôt, aplikácia dostane varovný signál, ktorý bude podnetom pre vykonanie ďalších funkcionalít, ktoré budú navrhované tak, aby pacientovi zabezpečili rýchlejšie zdravotné ošetrenie.
Aplikácia v situácii ohrozenia pacienta vytočí číslo rýchlej zdravotnej pomoci, čo by v prípade napríklad infarktu sám pacient nebol schopný vykonať. Ďalšou funkcionalitou aplikácie by malo byť odoslanie SMS správy, ktorá bude obsahovať GPS súradnice pacienta a text, v ktorom bude zrejmé, že pacient je v kritickom stave. Telefónne čísla, na ktoré sa táto SMS správa odošle si pacient bude môcť zvoliť sám v nastaveniach aplikácie.

3.3 Návrh aplikačného vybavenia

Pri návrhu aplikačného programového vybavenia bolo potrebné zvoliť vhodný programovací jazyk a vývojové prostredie. Pred tým však bolo potrebné určiť, pre ktorú platformu bude daný produkt vyvíjaný. Do úvahy prichádzali dve možnosti, buď bude softvér realizovaný na jednu konkrétnu mobilnú platformu, alebo bude vyvíjaný viac platformový, teda pre viacero platforiem súčasne.

3.3.1 Obmedzenia aplikácie

Obmedzením vyvíjaného systému je vývojové prostredie, pretože je nutné sa sústrediť len na voľne dostupné distribúcie, ktoré nie vždy ponúkajú dostatok funkcií. Hardvérovým obmedzením tejto aplikácie je potreba zariadenia s nasledujúcimi modulmi a funkciami:

- GPS
- WiFi
- Bluetooth
- Mobilné dáta (Edge, 3G, LTE)
- GSM

3.3.2 Výber platformy

V snahe pokryť čo najväčšiu plochu trhu s mobilnými zariadeniami sa uvažovalo nad vývojom aplikácie pre viacero mobilných platforiem súčasne. Ako už bolo vyššie spomenuté, lídrami na trhu s inteligentnými mobilnými zariadeniami sú v súčasnosti tri veľké korporácie: Apple, Google a Microsoft. Vzhľadom na to, že platforma každej tejto firmy má vlastný životný cyklus a odlišný natívny programovací jazyk, bolo donedávna takmer nemysliteľné vyvíjať produkt v jednom vývojovom prostredí a jediným programovacím jazykom.
Po analýze dostupných mobilných platforiem sa rozhodlo, že aplikácia sa bude vyvíjať pre tri najrozšírenejšie mobilné platformy a to Android, iOS a Windows Phone. Avšak z dôvodu absencie mobilného zariadenia s platformou iOS bude vyvíjaný systém implementovaný vo forme funkčného prototypu len na dve tieto platformy a to Android a Windows Phone.

3.3.3 Výber vývojového prostredia

Jedným z najdôležitejších rozhodnutí prí vývoji softvéru je správna voľba vývojového prostredia, ktoré by malo byť zvolené na základe programovacieho jazyka a mobilnej platformy, pre ktorú bude aplikácia realizovaná. Keďže sa zvolila cesta viac platformového vývoja aplikácie, prichádzali do úvahy len vývojové prostredia podporujúce rozšírenia viac platformových softvérových rámcov alebo samotné viac platformové vývojové prostredia, ktoré sú na realizáciu aplikácií na viacerých platformách primárne určené. Po dôkladnej analýze dostupných vývojových prostredí a softvérových rámcov sa ako najlepšie možné riešenie ukázalo vývojové prostredie Visual Studio, ktoré podporuje rozšírenie pre Xamarin, čo je multiplatformový softvérový rámec pre vývoj aplikácií na tri najrozšírenejšie platformy. Ďalšou výhodou tohto softvérového rámca je používaný programovací jazyk C#.

3.4 Tvorba aplikácie

Táto kapitola ponúkne podrobný opis návrhu a riešenia aplikácie. Opíše postup implementácie jednotlivých funkcionalít systému až po finálny stupeň funkčného prototypu.

3.4.1 Výber vhodnej metódy vývoja softvéru pre vývoj aplikácie

Vzhľadom na rozsiahlosť aplikácie a náväznosť na inú diplomovú prácu sa rozhodlo, že diplomanti budú počas písania prác využívať agilné metódy vývoja softvéru, čo by malo uľahčiť písanie a samotný vývoj projektov.
Agilné metódy vývoja softvéru sa v súčasnosti využívajú najmä vo firmách, ktoré sú zamerané na tvorbu a vývoj softvéru. Agilné metódy nepomenúvajú konkrétne kroky ako postupovať pri vývoji softvéru, opisujú skôr princíp ako by takýto vývoj mal fungovať. Ich zameraním je vývoj softvéru založený na iteratívnom a inkrementálnom

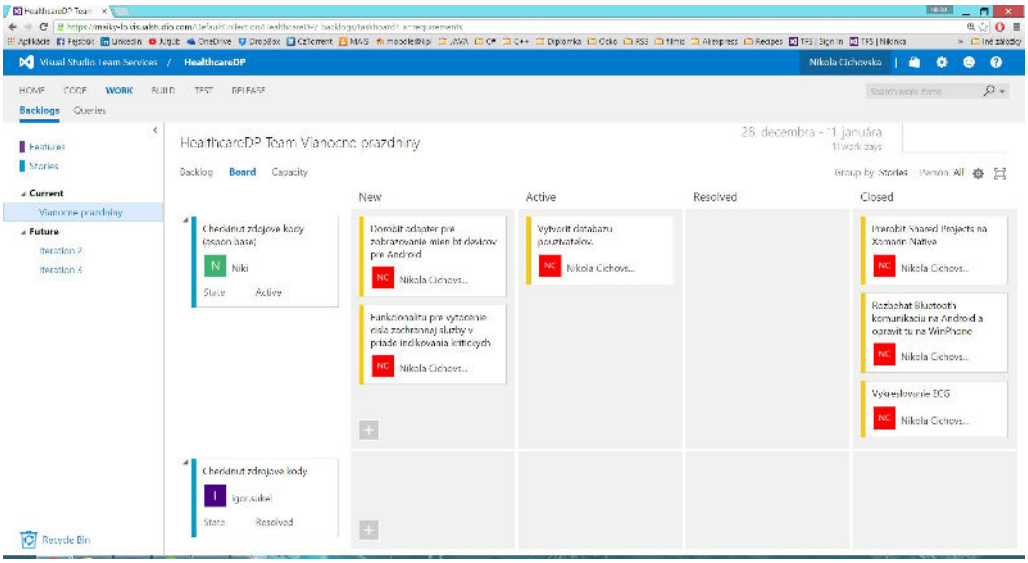
vývoji, pričom sa pracuje v seba-organizovaných a viac-odborových tímoch, čo podporuje evolučný vývoj a dodávky produktu vo vopred dohodnutých časových intervaloch. Takýto prístup k vývoji softvéru podporuje rýchle a pružné reakcie tímu programátorov ako aj manažérov na zmeny.
Medzi agilné metodiky patria:

- Extrémne programovanie
- Lean Development
- Vývoj riadený vlastnosťami softvéru
- Vývoj riadený testami
- SCRUM

Najznámejšou a najrozšírenejšou agilnou metódou je SCRUM, ktorého hlavným princípom je pravidelná komunikácia so zákazníkom, aby sa zabránilo nepochopeniu požiadaviek a následne dodaniu softvéru, ktorý sa bude líšiť od zákazníkových predstáv. Preto sa v pravidelných intervaloch, nazývaných tiež demo, stretne firma so zákazníkom a predstavuje časť produktu, na ktorej sa pracovalo od posledného stretnutia. Súčasťou tejto agilnej metódy je členenie do troch rolí: Zákazník (Product Owner), Scrum Master a Tím.
Rozhodlo sa teda rozčleniť diplomovú prácu na jednotlivé šprinty, ktoré trvali štyri týždne a ich výsledkom bolo krátke demo, kde boli prizvaní nielen konzultanti, ale aj ľudia z firmy, ktorí o túto tému prejavili záujem. Po ukončení dema sa konala retrospektíva, kde boli prítomní autori obidvoch diplomových prác, ich konzultanti a zhodnotili sa výhody, nevýhody predchádzajúceho šprintu, problémy, ktoré počas neho vznikli a ich riešenie. Jednotlivé šprinty a ich backlog, čo je množina všetkých operácií vykonávaných v projekte ako napríklad úlohy, ošetrovanie chýb, kontrola kódu a pod., boli plánované konzultantom, pričom diplomanti mali možnosť pridávania vlastných úloh. Na správu celého agilného procesu sa použil produkt TFS (Team Foundation Server) od spoločnosti Microsoft zobrazený na obrázku 3 –

1, ktorý ponúka možnosti:

- Tvorby backlogov
- Manažment zdrojového kódu
- Manažment požiadaviek
- Projektový manažment
- Testovací manažment



Obr. 3 – 1 Vytvorenie nového projektu pomocou Xamarin frameworku.

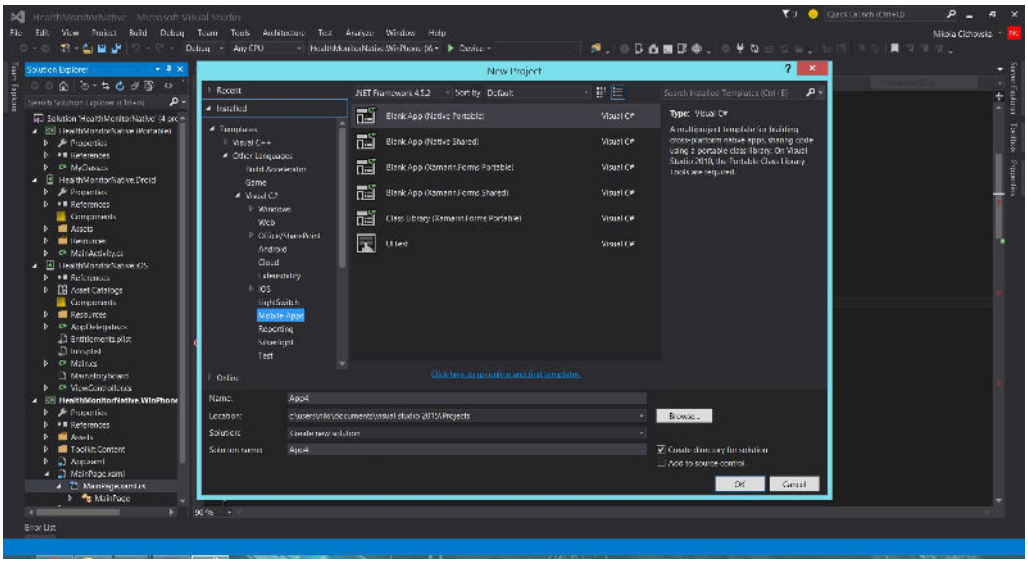
Je možné tvrdiť, že pokrýva celý životný cyklus aplikácie. Okrem webového používateľského rozhrania je možné ho spravovať aj priamo z vývojových prostredí Visual Studio alebo Eclipse.

3.4.2 Výber projektu pre tvorbu aplikácie

Pred samotným vývojom aplikácie bolo potrebné stiahnuť a nainštalovať vývojové prostredie, konkrétne Visual Studio Enterprise 2015, ktorého licencia je voľne do-

stupná študentom technických vysokých škôl prostredníctvom serveru msdn. Pre vývoj na viacerých platformách bol potrebný framework Xamarin, ktorého licencia bola získaná zadarmo po dobu jedného roka.

Ako prvé bolo potrebné vytvoriť nový projekt kde bolo možné vybrať z viacerých možností akým spôsobom bude zdrojový kód zdieľaný. Na začiatku sa zvolil projekt Blank App (Xamarin.Forms Shared), ktorý sa neskôr z dôvodu malého rozsahu knižníc potrebných pri vytváraní bluetooth spojenia medzi zariadeniami vymenil za projekt Blank App (Native Portable), kde boli k dispozícii natívne knižnice všetkých troch platforiem ako je možné vidieť na obrázku 3 – 2.



Obr. 3 – 2 Vytvorenie nového projektu pomocou Xamarin frameworku.

Po vytvorení projektu boli k dispozícii tri pod projekty, jeden pre každú platformu a hlavný projekt, ktorý obsahuje zdieľaný kód. Ku každému z troch pod projektov je k dispozícii designer, kde je možné pomocou drag'n'drop funkcie poskladať grafické rozhranie aplikácie, čo pri vývoji pre tri rozdielne platformy ušetrí veľa času. V prípade korekcií, ktoré v designeri nie je možné nastaviť bola k dispozícii druhá možnosť a to využiť XAML, Extensible Application Markup Language, čo je deklaratívny jazyk založený na XML, vyvinutý spoločnosťou Microsoft a používaný od .NET Frameworku 3.0 v technológiách Windows Presentation Foundation, Workflow Foundation

a Silverlight. Vo WPF a Silverlight sa používa na vytváranie užívateľského rozhrania. Všetko, čo sa vytvorí pomocou jazyka XAML je možné popísať aj pomocou jazykov C# či VB.NET, avšak použitie XAML predstavuje jednoduchšie riešenie.

3.4.3 Výber formy komunikácie medzi zariadeniami

Vzhľadom na to, že funkcionalita celej aplikácie je závislá od prijímania dát, bolo potrebné rozhodnúť aká komunikačná technológia bude použitá. Do úvahy pripadali dve možnosti: komunikácia bude realizovaná pomocou služby wifi direct alebo bluetooth.

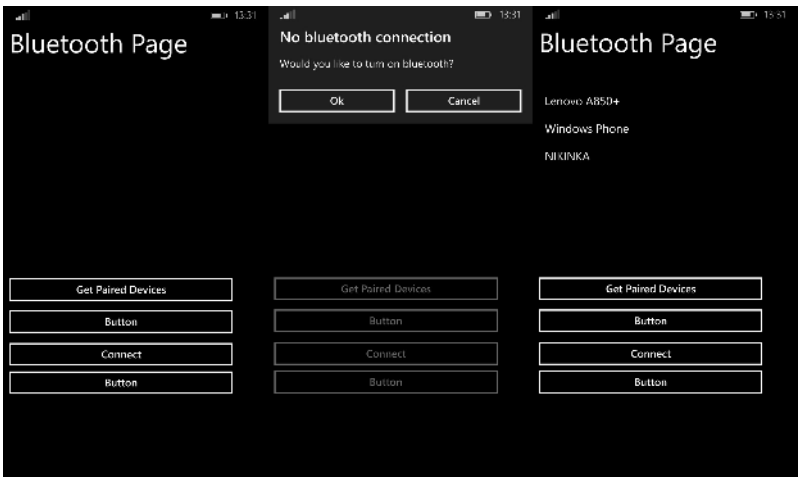
WiFi Direct je pomerne nová technológia, ktorá sa líši od klasického WiFi tým, že dokáže prepojiť dve zariadenia bez potreby pripájania sa na prístupový bod. Tento proces sa nazýva ad hoc WiFi transmssion. Táto technológia sa stala veľmi populárnou nielen v hernom priemysle (Nintendo DS, PlayStation Portable), ale aj pri digitálnych fotoaparátoch, smart TV a ďalších zariadeniach spotrebnej elektroniky. Výhodou tejto formy komunikácie je stabilnejšie spojenie a rýchlejší prenos dát medzi zariadeniami, avšak na druhej strane implementácia na zariadení Raspberry Pi by bola ťažšia a finančne náročnejšia. Nevýhodou tejto komunikačnej technológie je aj nedostatočná podpora aplikačných programových rozhraní (API).

Bluetooth vytvára komunikačné spojenie na krátky dosah (cca 100m) medzi dvoma zariadeniami. Používa rovnaké frekvenčné pásmo (2,4 GHz) ako WiFi, ale inú technológiu. Bluetooth sieť využíva model klient/server pre kontrolovanie kedy a kde sa môžu dáta odosielať. V súčasnosti sa na väčšine zariadení využíva verzia Bluetooth 4.0. Výhodou tohto komunikačného protokolu je široká dostupnosť API, nízka cena bluetooth adaptéru a najmä nižšia spotreba energie oproti WiFi.

Analýzou týchto dvoch komunikačných protokolov sa dospelo k rozhodnutiu, že pre tvorbu aplikácie, ktorá bude určená na monitorovanie zdravotného stavu pacienta bude použitá technológia Bluetooth nielen pre väčšiu dostupnosť aplikačných programových rozhraní a nižšiu cenu, ale hlavne kvôli nízkej spotrebe energie. Spotreba energie je rozhodujúcim faktom z dôvodu, že sa predpokladá, že pacient bude mať monitorovacie zariadenie pripojené počas celého dňa a ráta sa s možnosťou, že nebude mať v dosahu zdroj pre nabíjanie batérie mobilného zariadenia.
Po výbere komunikačnej technológie bol počiatočný vývoj aplikácie sústredený na vytvorenie stabilného spojenia medzi aplikáciou a iným zariadením.

Funkcionalita Bluetooth pre Windows Phone

Vzhľadom na to, že Xamarin používa programovací jazyk C# a väčšiu časť softvérového rámca .NET sa v tejto diplomovej práci využije natívne API vytvorené spoločnosťou Microsoft. Toto aplikačné programové rozhranie je možné použiť v projektoch typu Windows 8, Windows 8 Silverlight a Windows 8.1. Pozostáva zo základnej sady knižníc a tried, ktoré zabezpečujú funkcionalitu Bluetooth. Pred tvorbou samotného Bluetooth adaptéru je nutné v manifeste aplikácie povoliť prístup k tomuto adaptéru pre každú platformu osobitne.
Veľká časť funkcionality pre tvorbu Bluetooth je zahrnutá v mennom priestore Windows.Networking.Proximity. Ako prvé bolo potrebné nastaviť identitu v triede PeerFinder, ktorá je následne po tomto kroku schopná vyhľadať všetky spárované zariadenia alebo spustí hľadanie dostupných zariadení. Rozhodlo sa, že posledná spomenutá funkcionalita sa implementovať nebude, a to kvôli tomu, že párovanie zariadenia nemusí byť závislé na aplikácii podobne ako je to pri rôznych bluetooth headsetoch. Pri triede PeerFinder bolo potrebné ošetriť to, že dané zariadenie, na ktorom bude aplikácia spustiteľná nemusí mať bluetooth hardvér alebo bluetooth môže používať iná aplikácia, prípadne bude vypnutý ako je možné vidieť na obrázku 3 – 3. Tieto výnimky boli ošetrené pomocou try catch bloku. Následne sa zozbierané objekty, ktoré sa vrátili po volaní metódy PeerFinder.FindAllPeersAsync(), uložia



Obr. 3 – 3 Bluetooth rozhranie aplikácie pre Windows Phone.

do kolekcie, ktorá je vložená ako zdroj dát pre objekt typu ListView, čo je v podstate riadkový zoznam. Používateľ si vyberie zariadenie, s ktorým chce nadviazať komunikáciu, v tomto prípade zariadenie Raspberry Pi a odklikne tlačidlo Connect, pomocou ktorého sa bude snažiť nadviazať spojenie. Po kliknutí na tlačidlo Connect sa pomocou metódy PhoneApplicationService.Current.State.Add() pridá kľúč vo forme KeyValuePair (dvojica kľúč-hodnota) pre vybrané bluetooth zariadenie a jeho MAC adresu. Zároveň sa zavolá metóda NavigationService.Navigate(new Uri("/MainPage.xaml", UriKind.Relative)), ktorá používateľa presmeruje na ďalšiu stánku aplikácie.

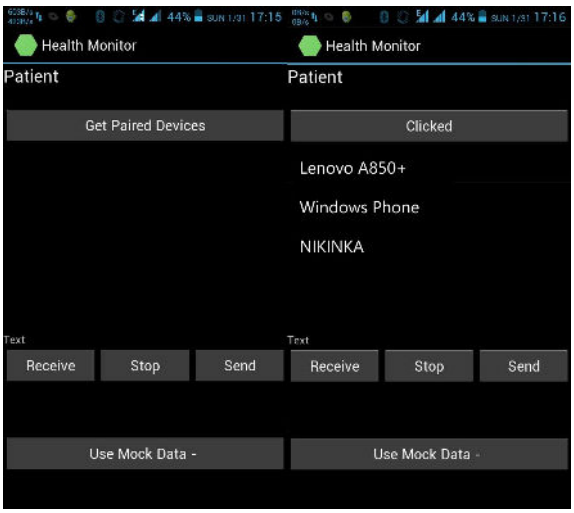
Posledná spomínaná stránka je hlavnou stránkou aplikácie a na jej pozadí prebieha bluetooth komunikácia. Hneď pri nabehnutí si aplikácia vytvorí nové vlákno, ktoré slúži na prijímanie dát z Raspberry Pi. Toto vlákno sa vytvára z dôvodu dosiahnutia plynulejšieho behu aplikácie a taktiež preto, aby hlavné UI vlákno nebolo zaťažované. Nové vlákno zavolá vlastnú metódu Receive(), v ktorej sa vytvorí socket a pomocou modelu klient/server sa nadviaže komunikácia. Základom tejto komunikácie je metóda socket.ConnectAsync(selectedBluetoothDevice.HostName, "fa87c0d0-afac-11de-8a39-0800200c9a66"). Prvým parametrom tejto metódy je hostiteľské meno zariadenia a nie MAC adresa ako by sa mohlo na prvý pohľad zdať.

Ako druhý parameter je použité UUID, čo je Universally Unique Identifier. UUID je identifikačný štandard používaný v softvérových konštrukciách. Je to 128-bitová hodnota zapísaná v hexadecimálnej sústave.Pre bluetooth sa pomocou UUID určujú profily zariadení a teda štandardy komunikácie medzi zariadeniami na rozhraní bluetooth. V tejto aplikácii sa rozhodlo, že sa vytvorí vlastný UUID a to preto, aby táto aplikácia bola unikátna a neexistovala žiadna iná aplikácia s rovnakým UUID z bezpečnostných dôvodov. Keďže táto metóda je blokovacia, bude čakať, až pokiaľ nenadviaže spojenie. V inom prípade vyhodí výnimku podľa chyby, ktorá nastala. Po úspešnom spojení sa vytvorí buffer, z rozhrania IBuffer, ktorý sa ale neskôr naplní štandardným poľom bajtov. Potom sa v slučke while načíta vstupný prúd, z ktorého sa vyčíta pole bajtov vo vopred dohodnutej veľkosti. Následne sa prečítané dáta prekonvertujú na string a ďalej spracúvajú.

Funkcionalita Bluetooth pre Android

Vzhľadom na to, že táto aplikácia je vyvíjaná pre viacero platforiem, usilovalo sa písať čo najviac zdieľaného kódu a zdieľať rovnaké postupy. Preto boli pri vývoji pre platformu Android použité podobné programovacie postupy ako pre platformu Windows Phone. Jediným rozdielom bola nutnosť implementovať triedy a metódy API, ktoré sú prispôsobené pre Android. Xamarin ponúka menný priestor Xamarin.Android, ktorý obsahuje wrapper-e natívnych javovských tried, ktoré fungujú pod natívnym Androidom.

Aplikácia je v Androide rozdelená na niekoľko obrazoviek rovnako ako aj vo Windows Phone. Na rozdiel od Widows Phone, kde je front-endová časť obrazovky pevne spätá s back-endom, pri Androide sa vytvárajú tzv. activity, ktoré obsahujú logiku vykonávanú na stránke a na ne sa napájajú layouty, ktoré sú vo formáte AXML, čo je Xamarin verzia klasického XML súboru. Takto je možné layouty využiť pri viacerých aktivitách nezávisle. V prvej aktivite, ktorá sa zaoberá bluetooth komunikáciou, sa nachádza ListView zariadení, ktoré sú už s mobilným zariadením spárované. Pre tento ListView bolo potrebné navrhnúť rozširujúci adaptér pre jeho obsah, ktorý

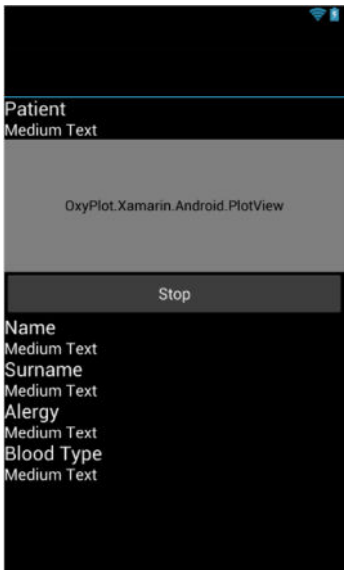


Obr. 3 – 4 Bluetooth rozhranie aplikácie pre Android.

by zobrazoval mená mobilných zariadení a nie MAC adresy, ktoré boli štandardne zobrazované. Pre toto je potrebné vytvoriť vlastnú triedu, ktorá rozširuje základný adaptér a potom do nej dodefinovať, ako sa majú informácie zobrazovať. Tu sa ukazuje dosť veľký rozdiel oproti Windows Phone, kde sa podobná situácia dá vyriešiť pomocou nastavenia zobrazovaných reťazcov v XAML zložke danej obrazovky. Pri písaní tejto aktivity sa využíva menný priestor Android.Bluetooth. Pomocou triedy BluetoothAdapter a statickým volaním BluetoothAdapter.DefaultAdapter je možné získať inštanciu adaptéra. Opäť je potrebné ošetriť, či je bluetooth adaptér voľný, či ho vôbec dané mobilné zariadenie má. Ďalej je nutné pridať povolenie pre používanie bluetooth do manifestu aplikácie. Následne sa zavolá vlastnosť BondedDevices, ktorá vráti zoznam všetkých spárovaných zariadení. Po odkliknutí želaného zariadenia je možné pokračovať ďalej. Klik na tlačidlo pokračovania vytvorí nový Intent, do ktorého sa vložia informácie o spárovanom zariadení a naštartuje sa nová aktivita, kam sa tieto údaje pošlú. Ako je možné vidieť na obrázku 3 – 4, aplikácia na platforme android sa veľmi nápadne podobá aplikácii vyvinutej pre operačný systém Windows Phone.

Po prijatí chorobopisu pacienta vo formáte json sa na hlavnej obrazovke aplikácie zobrazia základné údaje o pacientovi (ako je možné vidieť na obrázku 3 – 5), ktoré sa

následne ďalej spracúvajú. Na simuláciu odosielania dát bol na počítač nainštalovaný



Obr. 3 – 5 Hlavná obrazovka aplikácie, ktorá zobrazuje základné údaje o pacientovi.

program, ktorý odosiela prostredníctvom bluetooth string reťazec ľubovoľnej dĺžky. Pre overenie odosielania dát to síce postačovalo, ale program neumožňoval neustále odosielanie dát čo konečný produkt vyžaduje. Preto bolo potrebné vytvoriť metódu, ktorá by neustále odosielala reťazec stringov, pričom bolo taktiež potrebné dohodnúť formát, v ktorom budú dáta neskôr odosielané z Raspberry Pi. Táto metóda bola vložená do aplikácie a odosielanie dát sa realizovalo tou istou aplikáciou, ktorou sa prijímali.

3.4.4 Vykreslenie EKG grafu

Jednou z plánovaných funkcionalít tohoto projektu bolo aj vykresľovanie EKG krivky pacienta v reálnom čase. Všetky zozbierané hodnoty, namerané zariadením Raspberry Pi, sa rádovo v milisekundách odosielajú do mobilného zariadenia, kde ich aplikácia prijíma a ďalej spracúva. Pre vykresľovanie grafov a rôznych kriviek existuje v súčasnej dobe mnoho knižníc pre rôzne platformy ako napríklad:

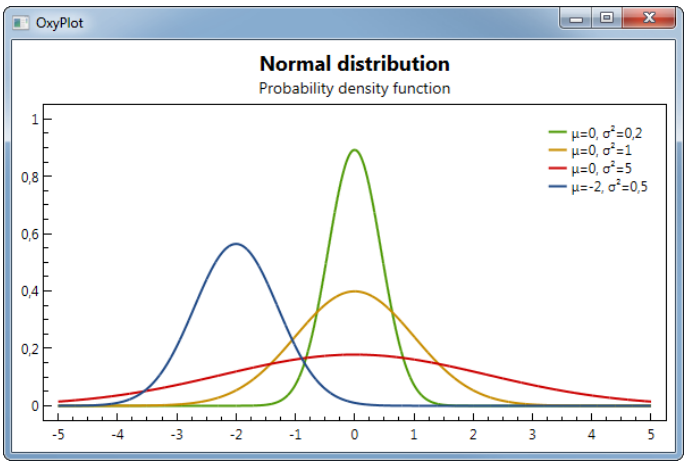
- GraphView

- ZedGraph
- NPlot
- OxyPlot

Po dôkladnej analýze sa pre samotné vykresľovanie EKG frekvencie zvolila knižnica OxyPlot, čo je viac platformová knižnica pre .Net, ktorá môže byť vďaka podpore Portable Class Library použitá na rôznych platformách. Vlastné ovládacie prvky sú implementované pre:

- WPF
- Windows 8
- Windows Phone Silverlight
- Windows Forms
- Silverlight
- XWT
- Xamarin.iOS
- Xamarin.Mac
- Xamarin.Forms
- Xamarin.Android

Najnovšie binárne súbory sú dostupné v NuGet balíčkoch, ktoré je potrebné stiahnuť a nainštalovať pomocou Manažéra NuGetových balíčkov, ktorý sa nachádza vo Visual Studiu. OxyPlot zahŕňa mnoho rôznych typov osí a sérií 3 – 6. V prípade, že je potrebná funkcionalita, ktorá nie je zahrnutá v knižnici, je možné vytvoriť vlastnú odvodenú triedu vďaka voľne šíriteľnej licencii knižnice. Ďalšou výhodou OxyPlotu je možnosť exportovania do súboru vo formátoch ako: png, pdf a svg.



Obr. 3 – 6 Ukážka vykreslenia grafov pomocou knižnice Oxyplot (35).

Funkcionalita vykreslenia EKG pre Windows Phone

Pre prácu s grafmi bolo potrebné použiť menný priestor OxyPlot a OxyPlot.Series. Pred tým ako sa implementovala samotná funkcionalita pre vykresľovanie EKG krivky bolo potrebné vytvoriť model pre front-end a navrhnúť rozloženie grafu na obrazovke aplikácie definovaním menného priestoru

```
xmlns:oxy="clr-namespace:OxyPlot.WP8;assembly=OxyPlot.WP8"
```

V back-endovom kóde je ako prvé potrebné vytvoriť pohľad modelu a inicializovať triedy PlotModel a LineSeries. Pomocou triedy OxyPlot.OxyColors sa nastaví farba pozadia grafu a samotnej kresliacej krivky. Následne sa v metóde ParseMockData(), ktorá bola vytvorená pre parsovanie prijatých dát z Raspberry Pi, pomocou menného priestoru OxyPlotWP8.PlotView nastavila minimálna a maximálna hodnota osi y.

```
plotView.ActualModel.Axes[1].Minimum = 0.5;
plotView.ActualModel.Axes[1].Maximum = 4;
```

Použitím podmieňovacieho príkazu if else sa vykonalo samotné vykresľovanie rozparsovaných dát. Ak bol počet vykreslených pixelov menší ako počet pixelov šírky celého grafu, tak sa postupne pridávali pixely. Inak sa pixel na prvom mieste vymazal a na posledné miesto sa pridal nový pixel. Krivka grafu tak vytvárala dojem, akoby

plynula sprava doľava. Obe vetvy podmieňovacieho príkazu boli navyše ošetrené try catch blokom pre prípad, že by aplikácia prestala pracovať správne. V tom prípade by sa na obrazovke objavilo okno s chybovou správou.

Funkcionalita vykreslenia EKG pre Android

V prípade platformy Andorid bol postup takmer identický. Na to, aby bolo vykresľovanie vôbec možné bolo najprv potrebné použiť menný priestor OxyPlot. Xamarin.Android a OxyPlot.Series. Pomocou triedy OxyPlot.OxyColors sa najprv nastavila farba pozadia grafu a samotnej kresliacej krivky. Následne sa v metóde ParseMockData(), ktorá bola vytvorená pre parsovanie prijatých dát z Raspberry Pi, pomocou menného priestoru OxyPlot.PlotView nastavila minimálna a maximálna hodnota osi y. Pomocou podmieňovacieho príkazu if else sa vykonalo samotné vykresľovanie rozparsovaných dát. Ak bol počet vykreslených pixelov menší ako počet pixelov šírky celého grafu, tak sa postupne pridávali pixely. Inak sa pixel na prvom mieste vymazal a na posledné miesto sa pridal nový pixel. Krivka grafu tak vytvárala dojem, akoby plynula sprava doľava. Obe vetvy podmieňovacieho príkazu boli navyše ošetrené try catch blokom pre prípad, že by aplikácia prestala pracovať správne. V tom prípade by sa na obrazovke objavilo okno s chybovou správou.

3.4.5 Výber vhodnej databázy

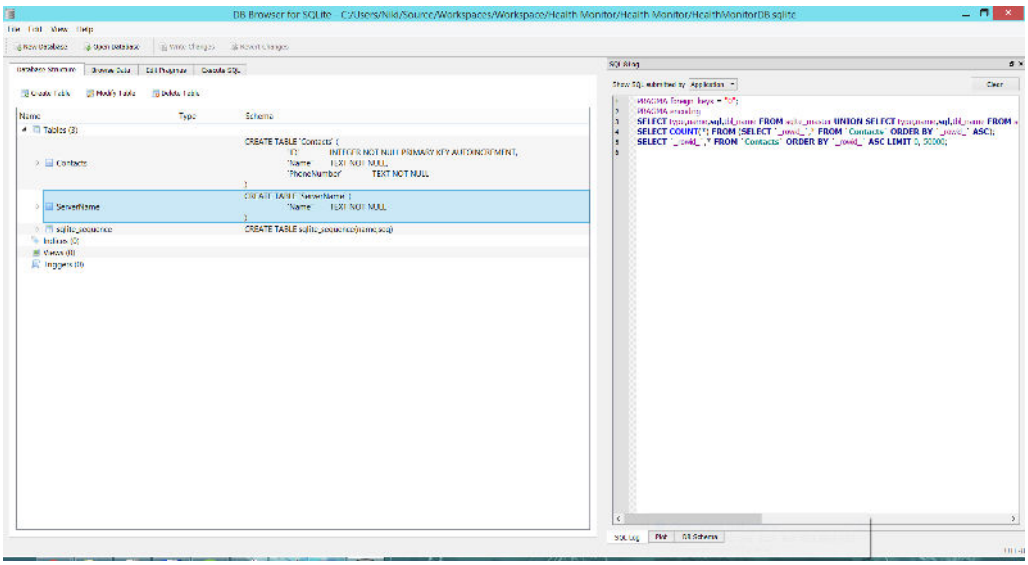
Pre správne fungovanie aplikácie bolo potrebné vytvoriť databázu, ktorá bude disponovať telefónnymi číslami, na ktoré bude v prípade núdze odoslaná sms správa s geolokáciou pacienta. Databáza bude taktiež obsahovať tabuľku, do ktorej sa uloží meno webového serveru lekára, na ktorý sa budú odosielať informácie o aktuálnom stave pacienta. Vzhľadom na nenáročnosť potrebnej databázy a jej využitie v aplikácii pre mobilné zariadenia sa rozhodlo vytvoriť databázu typu SQLite, ktorá je v čase písania tejto práce najčastejšie využívaná pre mobilné aplikácie bežiace na rôznych platformách.

SQLite je relačný databázový systém zahrnutý v knižnici napísanej v jazyku C. Na rozdiel od mnohých iných databázových systémov, SQLite nie je založená na princípe klient-server modely, kde je databázový server spustený ako samostatný proces. SQLite je knižnica, ktorá sa prilinkuje k aplikácii. Výhodou SQLite je možnosť použitia v rôznych programovacích jazykoch ako napríklad: C, C++, C#, Delphi, Haskell, Java, Objektové C, Perl, PHP, Python, R, Ruby atď.
Aby bolo možné ďalej s databázou pracovať v jednotlivých platformách, bolo potrebné ju najprv vytvoriť, čo sa udialo pomocou jednoduchého nástroja s názvom DB Browser for SQLite. Ako je možné vidieť na obrázku 3 – 7 boli v databáze vytvorené dve jednoduché tabuľky. Tabuľka Contacts obsahuje tri hodnoty:

- ID - identifikačné číslo typu integer, ktoré nesmie byť prázdne, slúži ako primárny kľúč a pridaním nových hodnôt do tabuľky sa automaticky inkrementuje
- Name - predstavuje meno pod ktorým bude telefónne číslo pridané, hodnota not null znamená, že táto hodnota musí byť do tabuľky zadaná
- PhoneNumber - znamená telefónne číslo, ktoré musí byť zadané do tabuľky

Druhá tabuľka ServerName obsahuje len jedinú hodnotu Name, teda meno serveru, na ktorý sa budú prijaté dáta odosielať.
Ďalším nutným krokom pred samotnou prácou s databázou v aplikácii je stiahnutie potrebných nugetov pre platformu Android a Windows Phone pomocou manažéra nugetových balíčkov.
Vzhľadom na to, že aplikácia je vyvíjaná na dve rôzne platformy bolo možné časť kódu databázy zdieľať tým, že v hlavnom projekte HealthMonitorNative sa vytvorili tri triedy používajúce menný priestor SQLite:

- DataBaseManagement.cs - hlavná trieda, v ktorej sa definujú metódy pre prácu s dátami v tabuľkách
- Contacts.cs - trieda obsahujúca ORM model, čo je objektovo-relačný model



Obr. 3 – 7 Ukážka editovania databázy pomocou nástroja DB Browse for SQLite.

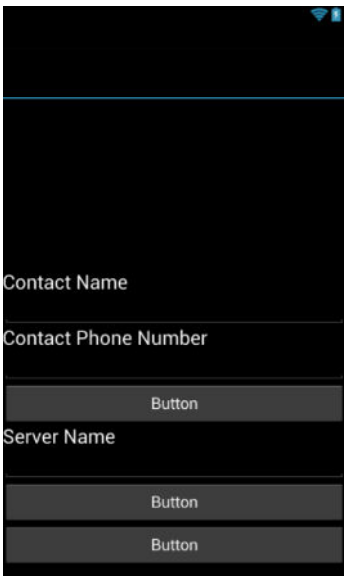
tabuľky Contacts

- ServerName.cs - trieda obsahujúca ORM model tabuľky ServerName

Práca s databázou v platforme Android a Windows Phone

V podprojekte pre platformu Android bolo potrebné pridať vytvorenú databázu do priečinka Assets. Ďalším krokom bolo vytvorenie novej obrazovky s názvom Settings, ktorá bude zobrazovať údaje z databázy a umožní úpravu a vkladanie nových dát ako je možné vidieť na obrázku 3 – 8. V prvej tretine obrazovky sa nachádza ListView obsahujúci údaje z tabuľky Contacts, čo znamená, že vypíše mená všetkých uložených čísel. Zvyšný priestor obrazovky vypĺňajú položky EditText, ktoré slúžia na pridávanie nových telefónnych čísel a názvu webového servera a tlačidlá potvrdzujúce zápis do databázy.

Po návrhu dizajnu obrazovky sa zameralo na funkcionalitu, pridala sa trieda Settings, kde bol vytvorený adaptér pre zobrazovanie obsahu v ListView, priradené eventy tlačidlám a volané metódy zo zdieľanej triedy DataBaseManagement. Ako prvá sa nastaví cesta k databáze aby s ňou bolo možné ďalej pracovať. Do privátnej premennej listview typu ListView sa priradí identifikátor listview, ktorý sa nachádza

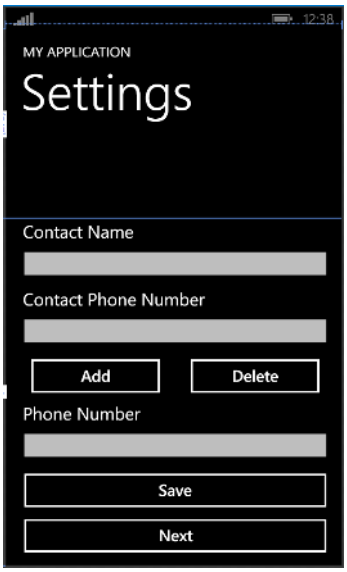


Obr. 3 – 8 Obrazovka s nastaveniami pre Android.

v obrazovke Settings. Po tom sa vytvorí adaptér pre listview, ktorý zobrazí mená všetkých telefónnych čísel uložených v databáze. Ďalej sa vytvorí akcia pre stlačenie prvého tlačila s názvom Add. Ako prvé sa získa text, ktorý bol používateľom vložený do textového poľa Name a Phone Number a následne na to sa zavolá metóda InsertUpdateData s parametrami contacts a path, čo predstavuje tabuľku a cestu k databáze. Po tom sa do obidvoch textových polí vloží prázdny string čo sa na obrazovke aplikácie prejaví ako vymazanie zadaných údajov. Po kliknutí na tlačidlo určené pre uloženie mena serveru sa rovnako získa text a zavolá metóda InsertUpdateData, ale s parametrom mena tabuľky ServerName a parametrom obsahujúcom cestu k databáze. Metóda s týmito parametrami sa správa trocha odlišne ako pri tabuľke kontaktov. V tabuľke ServerName bolo potrebné ukladať len jediný údaj s menom servera, ktorý sa nebude pridávať, ale prepisovať podľa potreby, preto táto metóda zakaždým vymaže predchádzajúci údaj v tabuľke a nahradí ho novým, ktorý ostane naďalej zobrazený v textovom poli na obrazovke aplikácie. Po implementovaní týchto metód sa uvažovalo o možnosti vymazávania telefónnych čísel z databázy preto sa vytvorila metóda na mazanie položiek z tabuľky kontaktov.

Podobne ako pri Androide, v pod projekte pre Windows Phone bolo potrebné pri-

dať databázu a vytvoriť novú obrazovku aplikácie pre zobrazovanie uložených telefónnych čísel, možnosť pridávania alebo mazania a pridanie mena webového servera ako je možné vidieť na obrázku 4 – 1. V triede Settings.xaml.cs boli následne implementované metódy na čítanie a zápis do databázy obdobne ako pri projekte pre platformu Android.



Obr. 3 – 9 Obrazovka s nastaveniami pre Windows Phone.

3.4.6 Odosielanie prijatých dát na webový server

Počas vývoja aplikácie sa rozhodlo, že sa rozšíri o funkcionalitu odosielania prijatých dát na webový databázový server lekára. Na simuláciu odosielania týchto dát bol vytvorený jednoduchý webový server v ASP.NET MVC.

Tvorba testovacieho webového servera

Na tvorbu webového servera bolo použité vývojové prostredie Microsoft Visual Studio Enterprise 2015, kde bol vytvorený nový projekt ASP.NET Web Application. Následne bola vytvorená databáza pomocou vyššie spomenutého nástroja DB Browser for SQLite, do ktorej boli pridané jednoduché tabuľky s názvom Data a Patient.

Tabuľka Data obsahuje hodnoty Value, čo predstavuje prijaté údaje z mobilného zariadenia a BirthId, teda rodné číslo pacienta slúžiace aj ako cudzí kľúč keďže je potrebné ukladať získané údaje ku konkrétnym pacientom. V tabuľke Patient sa vytvorili záznamy s potrebnými údajmi o pacientovi ako meno, priezvisko, rodné číslo, krvná skupina, adresa, alergie a lieky, ktoré pacient pravidelne užíva. Potom sa z databázy vygeneroval model a vytvorili sa kontrolóri a pohľady z modelov. Pre kontrolór PatientsController.cs bola vytvorená akcia pre prijate údajov vo forme json a následné uloženie do databázy. Trieda DataController.cs obsahuje akciu pre prijímanie hodnôt a rodného čísla pacienta, ktoré slúži ako cudzí kľúč pre tabuľku Patient, aby sa získané údaje vždy priradili správnemu pacientovi. Následne bol celý projekt za účelom testovania publikovaný na server azure.com.

Funkcionalita odosielania dát na webový server

V projekte vyvýjanej aplikácie bola v MainActivity.cs vytvorená metóda HttpPostOnServer(), ktorá odošle http požiadavku na webový server, ktorého adresu získa z databázy aplikácie a následne sa vykoná akcia ProcessData a údaje o pacientovi vo formáte json. Táto metóda je volaná v metóde Read() tesne po tom ako aplikácia prijme prvé dáta.

3.4.7 Vyhodnocovanie prijatých dát s kritickou hodnotou

Podstatnou úlohou vyvíjaného systému má byť snaha o záchranu života pacienta pomocou inteligentného mobilného zariadenia, ktoré bude obsahovať funkcionalitu na vytočenie čísla rýchlej zdravotnej pomoci a odoslanie SMS správy na uložené telefónne čísla v aplikácii.

Funkcionalita odosielania SMS správ

Ako prvé bolo potrebné povoliť vytáčanie hovorov v manifeste projektu pre Android a Windows Phone. V pod projekte pre Android v metóde ParseMockData()

je zavolaná metóda CallEmergency() v prípade, ak počet prijatých dát s kritickou hodnotou 1 je väčší nanajvýš rovný ako 1000. Tento podmieňovací príkaz bol implementovaný vzhľadom na to, že každú milisekundu aplikácia prijme jednu EKG hodnotu s údajom 0 (nepredstavuje žiadne riziko) alebo 1 (označuje prijatie kritickej hodnoty). Prijatie jedinej kritickej hodnoty však môže znamenať len nepodstatnú anomáliu srdcovej činnosti pacienta, prípadne chybne spracovaný prijatý údaj aplikáciou, preto je potrebné určiť, aký počet súvislých prijatých dát obsahujúcich kritickú hodnotu bude potrebný pre spustenie alarmu. Táto hodnota bola nastavená pre testovacie účely, pričom je možné ich v budúcnosti upraviť podľa individuálnych potrieb pacienta.

Metóda CallEmergency() v prípade, že súčasná poloha pacienta nie je prázdna, pretypuje GPS súradnice na string aby ich bolo možné zapísať do SMS správy. V tejto metóde sa súčasne nastaví cesta k databáze a pomocou metódy GetContacts() sa získajú telefónne čísla, ktoré si pacient predtým uložil v aplikácii. V metóde CallEmergency() sa nachádza cyklus foreach, ktorý pre každý kontakt uložený v databáze odošle SMS správu s polohou pacienta.

Samotná poloha pacienta sa získava v metóde GetGeoData(), kde sa vyberie poskytovateľ a zvolí sa kritérium presnosti polohy. Toto kritérium bolo v čase implementácie a testovania aplikácie nastavené na High, Medium a Low, pričom najkratší čas získania polohy sa dosiahol pri atribúte Low, teda najnižšej presnosti určenia polohy. Pre zabezpečenie rýchlejšej a neustálej aktualizácie polohy pacienta je táto metóda volaná na začiatku spustenia hlavnej obrazovky. Táto metóda sa v podobnej forme nachádza aj v pod projekte pre Windows Phone, kde však nebolo možné implementovať odosielanie SMS správ priamo vzhľadom na prísne bezpečnostné nastavenia firmy Microsoft, ktorá neposkytuje prístup k API všetkým developerom. Pre testovanie odosielania SMS správ s GPS súradnicami však bol vytvorený composer, ktorý po zavolaní metódy CallEmergency() poskladá SMS správu s textom o polohe pacienta a vyplní políčko s telefónnym číslom a následne ju zobrazí na obrazovke mobilného zariadenia, kde je však užívateľ nútený ju odoslať ručne.

Vytočenie čísla rýchlej zdravotnej pomoci

V prípade, že bude pacient v ohrození života, primárnou úlohou aplikácie v mobilnom zariadení bude vytočiť číslo rýchlej zdravotnej pomoci a zabezpečiť tak pacientovi okamžitú lekársku starostlivosť.

Na implementovanie tejto funkcionality v pod projekte pre Android platformu bol vytvorený Intent, ktorý sa používa na štartovanie aktivity pre vytočenie telefónneho čísla. V nasledujúcom úryvku kódu je možné vidieť celú implementáciu pre vytočenie čísla. Intent phone obsahuje dva parametre: typ akcie a adresu, ktorá zahŕňa formát a samotné číslo.

```
Intent phone = new Intent(Intent.ActionCall,
Android.Net.Uri.Parse(string.Format("tel:{0}", "112")));
StartActivity(phone);
```

Počas testovania sa namiesto čísla rýchlej zdravotnej pomoci použilo telefónne číslo autora diplomovej práce.

Implementácia funkcionality vytáčania hovoru bola obdobne použitá aj pre platformu Windows Phone. Bezpečnostné opatrenia operačného systému však neumožnili priame vytočenie hovoru.

Zobrazenie upozornenia na zistenie kritických údajov.

Uvažovalo sa, že počas prijímania dát obsahujúcich srdcovú frekvenciu pacienta môže dôjsť k chybnému prečítaniu údajov. V takomto prípade by sa spustili všetky funkcie pre záchranu života pacienta, čo by bolo nákladné v prípade zbytočného výjazdu rýchlej zdravotnej pomoci a upozornenia rodinných príslušníkov. Preto sa v aplikácii implementovala funkcionalita pre zobrazenie dialógového okna s časomierou pre overenie, či sa nejedná o chybný poplach.

Pre obidve platformy sa vytvorilo dialógové okno so správou či je pacient v poriadku a tlačidlom OK a 10 sekundovou časomierou. V prípade kliknutia na tlačidlo OK sa aplikácia vráti do pôvodného stavu, kde začína ďalej monitorovať srdcovú frek-

venciu pacienta. V inom prípade po uplynutí časomiery sú odoslané SMS správy a vytočené číslo RZP.

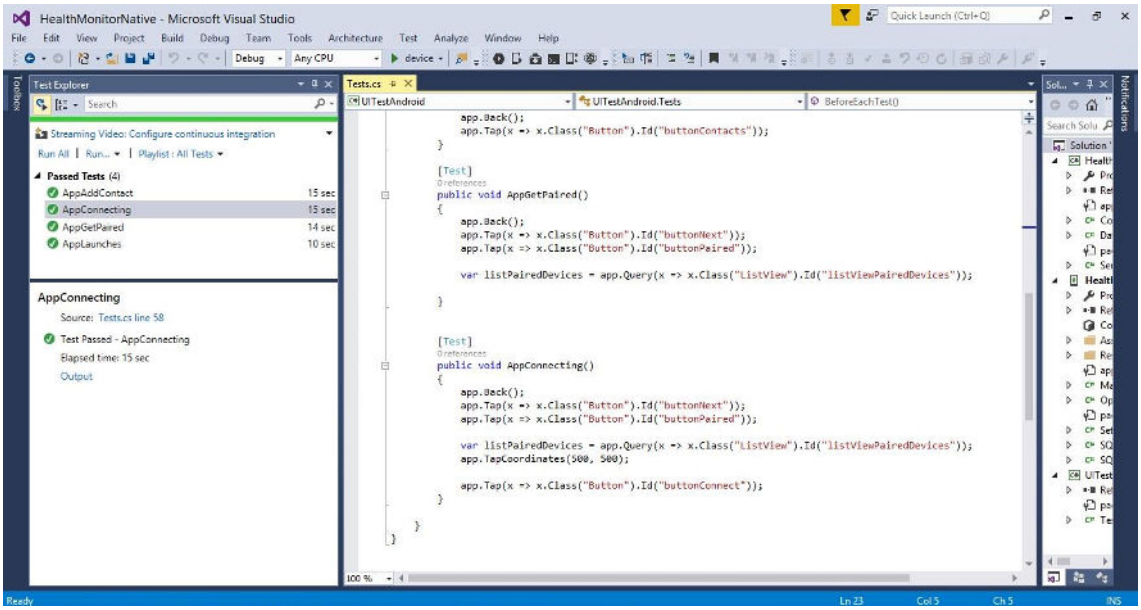
4 Testovanie

Po návrhu a následnej implementácii vo forme prototypu bolo potrebné overiť funkčnosť aplikácie najmä kvôli tomu, že sa jedná o medicínsky softvér, od ktorého presnosti a spoľahlivosti môže závisieť ľudský život.
Overenie funkčnosti aplikácie prebiehalo realizáciou týchto testov:

- Vytvorením a spustením UI unit testov
- Reálnym testovaním používateľmi na rôznych verziách operačného systému pre platformu Android a Windows Phone

4.1 Testovanie funkčnosti aplikácie prostredníctvom UI unit testov

V informačných technológiách sa počas vývoja softvéru používa unit testing, čo označuje automatické testovanie a overovanie fungovania implementácie aplikácií. Tento pojem zahŕňa spôsob, ktorého cieľom je overenie správnej funkčnosti jednotlivých časti zdrojového kódu systému.
Na testovanie implementovanej aplikácie boli použité unit testy pre používateľské rozhranie. Testovanie UI pomocou *Xamarin.UITest* sa vykonáva rovnako ako pri klasických unit testoch systémov. Výhodou je možnosť testovania celého prípadu použitia systému automatickým stláčaním tlačidiel aplikácie, pričom výsledky kliknutí sa vrátia v podobe asertov, a je tak možné vyhodnotiť či bol test vykonaný správne.
Na obrázku 2 – 4 je zobrazená syntax unit testov používateľského rozhrania. V pravej časti obrázku sa nachádzajú výsledky testov a čas, za ktorý boli vykonané. Ako je možné vidieť, všetky testy boli úspešne absolvované. Testovalo sa spustenie aplikácie, pridanie nového kontaktu do databázy, zobrazenie spárovaných zariadení a inicializovanie bluetooth spojenia.



Obr. 4 – 1 Testovanie aplikácie pomocou UI unit testov.

4.2 Testovanie funkčnosti aplikácie používateľom

Manuálne testovanie systému je neodmysliteľnou súčasťou overovania funkčnosti aplikácie. Úlohou tohto testovania bolo odhalenie chýb počas behu systému, objavenie nedostatkov a obmedzení aplikácie, ako aj zistenie minimálnych hardvérových a softvérových požiadaviek na systém.

Aplikácia *Health Monitor* bola nainštalovaná na tieto mobilné zariadenia:

- Nokia Lumia 830
- Microsoft Lumia 640 LTE
- Sony Xperia SP
- Lenovo A850+
- Xiaomi Redmi Note 2

- Xiaomi MI4
- HTC One S
- ZTE Grand X In
- Samsung Galaxy S4

Inštalácia a následné spustenie aplikácie prebehlo úspešne na všetkých testovacích mobilných zariadeniach. Problém nastal až pri zapisovaní dát do databázy, keď sa vyhodila výnimka *SQLite Exepcion*. Táto chyba sa objavila len pri zariadeniach s operačným systémom Android s verziou staršou ako 4.2.0 čím sa odhalili minimálne požiadavky aplikácie na softvérové vybavenie. Výsledky testov sú znázornené v tabuľke 4 – 1.

Okrem spomenutého obmedzenia neboli počas testovania odhalené žiadne závady aplikácie a jej beh bol plynulý počas celej doby testovania.

Tabuľka 4 – 1 Testovanie funkčnosti aplikácie na rôznych verziách operačných systémov.

Model mobilného zariadenia	Verzia operačného systému	Výsledok testovania aplikácie
Nokia Lumia 830	Windows Phone 8.1	✓
Microsoft Lumia 640 LTE	Windows Phone 8.1	✓
Sonz Xperia SP	Android v4.1	x
Lenovo A850+	Android v4.2.2	✓
Xiaomi Redmi Note 2	Android v5.0	✓
HTC One S	Android v4.0	x
Xiaomi MI4	Android v4.4.3	✓
Samsung Galaxz S4	Android v5.0.1	✓
ZTE Grand X In	Android v4.0	x

5 Vyhodnotenie riešenia

Táto kapitola obsahuje vyhodnotenie riešenia navrhnutého a implementovaného systému pre monitorovanie srdcovej frekvencie pacienta s vyhodnotením kritickej situácie. V prvej časti sú opísané prínosy diplomovej práce. Druhá časť zahŕňa výhody aplikácie *Health Monitor*, ktoré prevažujú nad nevýhodami spomenutými v poslednej časti tejto kapitoly.

5.1 Prínosy diplomovej práce

Predmetom tejto diplomovej práce bolo navrhnúť aplikáciu v najväčšej možnej miere využívajúcu softvérové a hardvérové možnosti mobilných zariadení. Táto aplikácia mala byť sústredená na oblasť medicíny, kde bolo jej úlohou monitorovať EKG dáta prijaté z prídavného zariadenia Raspberry Pi, odosielať ich na webový server a v reálnom čase zobrazovať EKG graf. Ďalšou predpokladanou funkcionalitou systému bolo vyhodnocovanie kritickej situácie, kedy by aplikácia po uplynutí časomiery odoslala SMS správu s GPS súradnicami na vopred uložené telefónne čísla, následne na to uskutočnila hovor pre privolanie rýchlej zdravotnej pomoci.

Prínosom tejto práce je návrh a úspešná implementácia aplikácie vo forme funkčného prototypu, ktorý bol vytvorený pre dve mobilné platformy, a to Android a Windows Phone. Vývoj aplikácie bol realizovaný prostredníctvom multiplatformového softvérového rámca Xamarin a je preto v budúcnosti možné jednoduchšie implementovanie aj pre platformu iOS.

Aplikácia *Health Monitor* môže byť užitočným nástrojom lekár pre monitorovanie srdcovej frekvencie pacienta počas celého dňa, pričom by ho neobmedzovala v pohybe a vykonávaní bežných dennodenných aktivít vďaka neustálemu odosielaniu dát na webový server lekára.

Prínosom aplikácie pre používateľa nie je len zobrazovanie jeho srdcovej frekvencie v reálnom čase, ale aj vyhodnocovanie kritickej situácie a privolanie záchrannej služby

čo môže používateľovi zachrániť život.
Aplikáciu je možné využiť aj v medicínskych zariadeniach na testovacie účely.

5.2 Výhody aplikácie Health Monitor

Výhody aplikácie je možné zhrnúť do týchto bodov:

- multiplatformovosť (Android, Windows Phone)
- plynulý beh aplikácie
- odosielanie dát na webový server
- zobrazovanie EKG dát v reálnom čase
- odosielanie SMS správy s GPS súradnicami na telefónne čísla zvolené používateľom
- zobrazenia návodu s prvou pomocou
- vytočenie čísla rýchlej zdravotnej pomoci
- v prípade vyhodnotenia kritickej situácie sa zobrazí okno s časomierou (výhoda v prípade prečítania nesprávnych údajov) content...

5.3 Nevýhody aplikácie Health Monitor

Nevýhody aplikácie je možné zhrnúť do týchto bodov:

- aplikácia podporuje operačný systém Android v4.2.0 a vyššie čo znamená, že na iných verziách OS Android nemusí správne fungovať
- bezpečnostné opatrenia v operačnom systéme Windows Phone (nedovoľuje priamo uskutočniť hovor ani odoslať SMS správu)
- dáta odosielané prostredníctvom bluetooth nie sú šifrované
- dáta odosielané na webový server nie sú šifrované

6 Záver

Cieľom tejto diplomovej práce bolo teoreticky analyzovať spôsob a vhodnosť využitia mobilných zariadení, ich operačných systémov a vývojových prostriedkov pre spracovanie údajov indikujúcich kritický zdravotný stav pacienta. Taktiež boli zhodnotené medicínske aplikácie pre mobilné telefóny, ich spoľahlivosť, dostupnosť a použiteľnosť. Následne bolo požadované navrhnúť a na úrovni funkčného prototypu implementovať aplikáciu pre vstup biometrických údajov a ich spracovanie. Ďalej sa požadovalo spracovanie získaných vstupných údajov pre účely aplikačného využitia a náväznosť práce na diplomovú prácu *"Systém pre monitorovanie a rýchlu diagnostiku kritického stavu pacienta, s využitím mobilných zariadení"*.

V úvodnej časti sa diplomová práca venuje analýze problematiky medicínskych aplikácií pre mobilné zariadenia, pričom sa zameriava na tri najrozšírenejšie mobilné platformy na trhu, ich architektúru, opisuje vývojové prostredia pre tieto platformy a definuje viac platformový vývoj aplikácií. Ďalej opisuje dostupné mobilné aplikácie určené pre monitorovanie zdravotného stavu pacienta.

V ďalšej časti práce je navrhnutá aplikácia pre dve mobilné platformy: Android a Windows Phone, ktoré v dobe písania tejto práce patria medzi najrozšírenejšie na trhu. Pre druhú najrozšírenejšiu pltafromu iOS sa rozhodlo aplikáciu nevyvíjať z dôvodu nemožnosti testovania aplikácie počas vývoja kvôli absencii mobilného zariadenia od firmy Apple. Vyvinutá aplikácia dokáže spracovať vstupné dáta obdržané zo zariadenia Raspberry Pi prostredníctvom bluetooth spojenia a ďalej ich spracúvať. V reálnom čase plynule vykresľuje prijaté údaje do EKG grafu, zobrazuje informácie o pacientovi prijaté z Raspberry Pi vo formáte json, ktoré následne odosiela na webový server. V prípade zistenia kritických hodnôt vytočí číslo rýchlej záchrannej pomoci a odošle sms správu s informáciou o aktuálnej polohe na telefónne čísla uložené v databáze aplikácie.

V závere práce sú opísané metódy testovania a vyhodnotenie riešenia. Aplikácia využíva v maximálnej možnej miere bežné mobilné zariadenia, bola manuálne testo-

vaná zo strany používateľa a zároveň boli vytvorené a spustené unit testy používateľského rozhrania. Aplikácia spĺňa všetky ciele, ktoré boli kladené na túto diplomovú prácu.

Aplikácia sa v dobe písania tejto práce nachádza v stave funkčného prototypu a je preto možné odporúčiť jej ďalšie spracovanie a rozširovanie funkcionalít, napríklad zabezpečenie šifrovanej bluetooth komunikácie. Vzhľadom na to, že na jej vývoj bol použitý viac platformový softvérový rámec Xamarin, je možné ju v budúcnosti jednoduchšie preniesť aj na platformu iOS vďaka funkcii zdieľania kódu, ktorú tento softvérový rámec ponúka. Ďalším možným rozšírením je úprava webového serveru pre lekára ako napríklad tvorba databázy kde sa budú prijaté údaje ukladať, zobrazovať a bude možné ich filtrovať podľa rôznych kritérií, vytvorenie registračného a prihlasovacieho formulára pre lekára a prispôsobenie používateľsky priateľského rozhrania webového servera.

Literatúra

[1] Ozdalga, E., Ozdalga, A. and Ahuja, N. (2012). The Smartphone in Medicine: A Review of Current and Potential Use Among Physicians and Students. J Med Internet Res, 14(5), p.e128.

[2] Sposaro F, e. (2015). iWander: An Android application for dementia patients. - PubMed - NCBI. [online] Ncbi.nlm.nih.gov. Dostupné na internete: http://www.ncbi.nlm.nih.gov/pubmed/21097072 [cit. 9 Dec. 2015].

[3] Sposaro F, Danielson J, Tyson G. iWander: An Android application for dementia patients. Conf Proc IEEE Eng Med Biol Soc. 2010;2010:3875–8. doi: 10.1109/IEMBS.2010.5627669.

[4] Wu HH, Lemaire ED, Baddour N. Change-of-state determination to recognize mobility activities using a BlackBerry smartphone. Conf Proc IEEE Eng Med Biol Soc. 2011;2011:5252–5. doi: 10.1109/IEMBS.2011.6091299.

[5] Worringham, C., Rojek, A. and Stewart, I. (2011). Development and Feasibility of a Smartphone, ECG and GPS Based System for Remotely Monitoring Exercise in Cardiac Rehabilitation. PLoS ONE, 6(2), p.e14669.

[6] Edgar S, Swyka T, Fulk G, Sazonov ES. Wearable shoe-based device for rehabilitation of stroke patients. Conf Proc IEEE Eng Med Biol Soc. 2010;2010:3772–5. doi: 10.1109/IEMBS.2010.5627577.

[7] Yamada M, Aoyama T, Mori S, Nishiguchi S, Okamoto K, Ito T, Muto S, Ishihara T, Yoshitomi H, Ito H. Objective assessment of abnormal gait in patients with rheumatoid arthritis using a smartphone.Rheumatol Int. 2011 Dec 23; doi: 10.1007/s00296-011-2283-2.

[8] Nishiguchi S, Yamada M, Nagai K, Mori S, Kajiwara Y, Sonoda T, Yoshimura K, Yoshitomi H, Ito H, Okamoto K, Ito T, Muto S, Ishihara T, Aoyama T.

Reliability and validity of gait analysis by android-based smartphone. Telemed J E Health. 2012 May;18(4):292–6. doi: 10.1089/tmj.2011.0132.

[9] Mellone S, Tacconi C, Chiari L. Validity of a Smartphone-based instrumented Timed Up and Go. Gait Posture. 2012 May;36(1):163–5. doi: 10.1016/j.gaitpost.2012.02.006.S0966-6362(12)00057-4

[10] Worringham, C., Rojek, A. and Stewart, I. (2011). Development and Feasibility of a Smartphone, ECG and GPS Based System for Remotely Monitoring Exercise in Cardiac Rehabilitation. PLoS ONE, 6(2), p.e14669.

[11] Bsoul M, Minn H, Tamil L. Apnea MedAssist: real-time sleep apnea monitor using single-lead ECG.IEEE Trans Inf Technol Biomed. 2011 May;15(3):416–27. doi: 10.1109/TITB.2010.2087386.

[12] Charpentier G, Benhamou PY, Dardari D, Clergeot A, Franc S, Schaepelynck-Belicar P, Catargi B, Melki V, Chaillous L, Farret A, Bosson JL, Penfornis A, TeleDiab Study Group The Diabeo software enabling individualized insulin dose adjustments combined with telemedicine support improves HbA1c in poorly controlled type 1 diabetic patients: a 6-month, randomized, open-label, parallel-group, multicenter trial (TeleDiab 1 Study) Diabetes Care. 2011 Mar;34(3):533–9. doi: 10.2337/dc10-1259.dc10-1259

[13] CardioPulse Articles European Cardiovascular Disease Statistics 4th edition 2012 : EuroHeart II European Heart Journal Elite Reviewers of 2012Top EHJ Editors 2012Highlights from European Society of Cardiology Congress 2013. (2013). Eur Heart J, 34(39), pp.3007-3013.

[14] Kestens, M. (2015). Percentage of Deaths from CVD - EHN - European Heart Network. [online] Ehnheart.org. Dostupné na internete: http://www.ehnheart.org/cvd-statistics/percentage-of-deaths-from-cvd.html [cit. 9 Dec. 2015].

[15] 2, H. and LLC, R. (2016). Healthy Heart 2 on the App Store. [online] App Store. Dostupné na internete: https://itunes.apple.com/us/app/healthy-heart-2/id352887946?mt=8 [cit. 17 Apr. 2016].

[16] Play.google.com. (2016). [online] Dostupnné na internete: https://play.google.com/store/apps/details?id=com.alivecor.aliveecg [cit. 17 Apr. 2016].

[17] Nice.org.uk. (2015). AliveCor Heart Monitor and AliveECG app for detecting atrial fibrillation — Technology-overview — Advice — NICE. [online] Dostupné na internete: https://www.nice.org.uk/advice/mib35/chapter/Technology-overview [cit. 17 Apr. 2016].

[18] Medgadget. (2012). AliveCor iPhone ECG Receives FDA Clearance —. [online] Dostupné na internete: http://www.medgadget.com/2012/12/alivecor-iphone-ecg-receives-fda-clearance.html?trendmd-shared=0 [cit. 17 Apr. 2016].

[19] Alivecor.com. (2016). AliveCor. [online] Dostupné na internete: http://www.alivecor.com/service/ [cit. 17 Apr. 2016].

[20] Megaemg.com. (2016). eMotion ECG — Mega Electronics Ltd. [online] Dostupné na internete: http://www.megaemg.com/products/emotion-ecg/ [cit. 17 Apr. 2016].

[21] Medapp.pl. (2016). MedApp - mobile medical applications — EKG Telemedyczne. [online] Dostupné na internete: http://medapp.pl/pl/produkty/ekg/ [cit. 17 Apr. 2016].

[22] MobiHealthNews. (2013). FDA clears Android-based continuous ECG monitor. [online] Dostupné na internete: http://mobihealthnews.com/27865/fda-clears-android-based-continuous-ecg-monitor [cit. 17 Apr. 2016].

[23] www.idc.com, (2015). IDC: Smartphone OS Market Share. [online] Dostupné

na internete: http://www.idc.com/prodserv/smartphone-os-market-share.jsp [cit. 1 Nov. 2015].

[24] www.tutorialspoint.com. (2016). Android Architecture. [online] Dostupné na internete: http://www.tutorialspoint.com/android/android_architecture.htm [cit. 26 Apr. 2016].

[25] Developer.apple.com. (2016). About the iOS Technologies. [online] Dostupné nainternete: https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html [cit. 26 Apr. 2016].

[26] Whitechapel, A. and McKenna, S. (2012). Windows Phone 8 development internals. Redmond, Wash.: Microsoft Press.

[27] Anon, (2016). [online] Dostupné na internete: https://sysdev.microsoft.com /en-us/Hardware/oem/docs/Getting_Started/Windows_Phone_architecture_overview [cit. 26 Apr. 2016].

[28] .NET Junkyard. (2016). .NET Junkyard. [online] Dostupné na internete: http://j4ni.com/blog/ [cit. 26 Apr. 2016].

[29] Xamarin.com, (2015). Mobile App Development & App Creation Software - Xamarin. [online] Dostupné na internete: https://xamarin.com/ [cit. 2 Dec. 2015].

[30] Qt, (2015). Qt - Home. [online] Dostupné na internete: http://www.qt.io/ [cit. 2 Dec. 2015].

[31] Haslam, O., Morris, P., Morris, P., Khan, S., Rehman, Z. and Morris, P. (2013).Download Android Studio IDE For Windows, OS X And Linux — Redmond Pie. [online] Redmond Pie. Dostupné na internete: http://www.redmondpie.com/download-android-studio-ide-for-windows-os-x-and-linux/ [cit. 3 Dec. 2015].

[32] Unity3d.com, (2015). Unity - Multiplatform - Publish your game to over 10 platforms. [online] Dostupné na internete: https://unity3d.com/unity/multiplatform [cit. 3 Dec. 2015].

[33] Anon, (2016). [online] Dostupné na internete: http://www.edy.es/dev/2014/02/a-better-blender-to-unity-3d-importer/ [cit. 27 Apr. 2016].

[34] Overview, A. (2015). Android Studio Overview — Android Developers. [online] Developer.android.com. Dostupné na internete: http://developer.android.com/tools/studio/index.html [cit. 3 Dec. 2015].

[35] Docs.oxyplot.org. (2016). Introduction — OxyPlot 2015.1 documentation. [online] Dostupné na internete: http://docs.oxyplot.org/en/latest/introduction/introduction.html [cit. 17 Apr. 2016].

[36] Sukeľ, I. (2016). Systém pre monitorovanie a rýchlu diagnostiku kritického stavu pacienta, s využitím mobilných zariadení. Technická univerzita Košice.

Zoznam príloh

Príloha A Používateľská príručka

Príloha B Systémová príručka

Príloha C CD médium