

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Android v medicínskych systémoch
Diplomová práca

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Android v medicínskych systémoch
Diplomová práca

Študijný program: Informatika
Študijný odbor: 9.2.1 Informatika
Školiace pracovisko: Katedra počítačov a informatiky (KPI)
Školiteľ: Doc. Ing. Zdeněk Havlice, CSc.
Konzultant: Ing. Juraj Vízi

Abstrakt v SJ

Cieľom diplomovej práce je upriamiť pozornosť na neustály rozvoj a napredovanie mobilných zariadení a operačných systémov nielen v bežných oblastiach, ale aj v oblastiach medicíny. Oblasť medicíny patrí medzi najväčšie a najrozmanitejšie odvetvia. Samotný vývoj aplikácií je veľmi citlivý, pretože podanie nesprávnych informácií prípadne zlé vyhodnotenie programu, či chybný krok zariadenia, môžu viesť k ublíženiu nejakej osoby. Všetky softvérové produkty využívané v tejto oblasti musia byť vyvíjané precízne a podliehajú prísny štandardom a normám. Úlohou tejto práce je oboznámiť sa s tými najdôležitejšími. Sú tu taktiež popísané súčasne najrozšírenejšie programy využívané v medicíne na rôznych mobilných zariadeniach. Ďalšou zaujímavou témou je návrh a popis oblastí pre inovačné využitie mobilnej platformy Android v medicíne. Súčasťou práce je aj návrh a implementácia vlastnej aplikácie v podobe elektronickej zdravotnej karty, ktorá je dostupná nielen pre lekára, ale aj pre pacientov.

Kľúčové slová

Mobilná aplikácia, operačný systém Android, medicínske systémy

Abstrakt v AJ

The goal of this thesis is to present development and progress of mobile devices and operating systems especially in the medical area. This area belongs to the biggest and to the most diverse sectors. The development of applications is very critical due to reliability of information, program's execution etc., all which can lead to a patient's injury or device's damage. All software products in the medical area must be developed very precisely and must follow standards and norms. This thesis represents the most important medical standards. The characteristics of mobile medicine applications are also described. The presented application is an electronic health record system for the Android operating system. The scope of access to this application is limited to doctors and patients.

Kľúčové slová v AJ

Mobile application, mobile operating system Android, medical systems

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:


Android v medicínskych systémoch
Android in Medical Systems

Študent (tituly, meno, priezvisko): **Bc. Veronika Jančová**
Školiteľ (tituly, meno, priezvisko): **doc. Ing. Zdeněk Havlice, CSc.**
Školiace pracovisko: **Katedra počítačov a informatiky**
Konzultant práce (tituly, meno, priezvisko): **Ing. Juraj Vízi**
Pracovisko konzultanta: **Siemens Program and System Engineering s.r.o.**

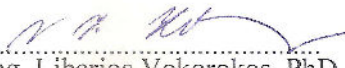
Pokyny na vypracovanie diplomovej práce:

1. Popis súčasného stavu použitia OS Android vo vložených systémoch a medicínskych systémoch.
2. Návrh a popis oblasti pre inovačné využitie platformy Android v medicínskych systémoch.
3. Návrh a implementácia mobilnej aplikácie pre medicínske účely.
4. Vytvorenie databázy používateľov a jej integrácia s aplikáciou.
5. Vypracovať dokumentáciu podľa pokynov vedúceho práce (hlavná časť v rozsahu 50-70 strán a prílohy, tlačaná forma v nerozoberateľnej väzbe a e-forma na DVD).

Jazyk, v ktorom sa práca vypracuje: slovenský
Termín pre odovzdanie práce: 26.04.2013
Dátum zadania diplomovej práce: 31.10.2012


prof. Ing. Ján Kollár, CSc.
vedúci garantujúceho pracoviska




prof. Ing. Liberios Vokorokos, PhD.
dekan fakulty

Čestné vyhlásenie

Vyhlasujem, že som celú diplomovú prácu vypracovala samostatne za pomoci uvedenej odbornej literatúry.

Košice, 20.máj 2013

.....

vlastnoručný podpis

Pod'akovanie

Ďakujem vedúcemu diplomovej práce doc. Ing. Zdeňkovi Havlicemu, CSc. a konzultantovi Ing. Jurajovi Vízimu za pomoc, všetky užitočné rady, ochotu a čas, ktoré mi venovali. Tiež by som chcela poďakovať mojej rodine za porozumenie a trpezlivosť počas vypracovávania praktickej a teoretickej časti práce.

Predhovor

V svojej práci som sa zamerala na vývoj aplikácií pre mobilné zariadenia s dotykovou obrazovkou a operačným systémom Android, verzia 4.0. V súčasnosti sa na trhu nachádza nespočetne veľa programov pre bežných používateľov. V niektorých štátoch je použitie týchto zariadení už bežné v rôznych priemysloch. Ja som sa zamerala na aplikácie v oblasti medicíny. S týmito zariadeniami si uľahčujú prácu mnohí zahraniční lekári.

Pod pojmom elektronická zdravotná karta si veľa ľudí predstaví rôzne veci. Medzi najčastejšie predstavy patrí kartička, v ktorej by boli uvedené všetky zdravotné údaje o pacientovi. Lekári by boli vybavení špeciálnou čítačkou, ktorá by vedela po priložení karty, prečítať o akého pacienta ide a akými chorobami prešiel. Taktiež by mohla disponovať aj takou funkciou, aby sa uchovávali všetky výsledky jednotlivých vyšetrení.

Reálnejšou predstavou je však elektronická zdravotná karta reprezentujúca komplexný informačný systém bežiaci na zariadení, ktoré by malo prístup na nejaký vzdialený server, kde by boli uchované dáta. Išlo by o obrovskú databázu pacientov a ich údajov. Pacienti by si na vyšetrenia k odborným lekárom nemuseli nosiť zdravotnú kartu v papierovej forme, alebo výsledky z jednotlivých vyšetrení. Všetko by bolo dostupné na nejakom špeciálne zabezpečenom serveri. Všetky dáta by boli šifrované. Takýto systém má veľa výhod, ale aj veľa nevýhod. Mohol by sa stať terčom nelegálnych útokov.

Mojou prácou by som chcela ukázať na možnosť vytvorenia systému, ktorý by predstavoval systém slúžiaci nielen pre lekára, ale aj pre pacienta. Pacient si v súčasnosti nemá možnosť prečítať záznamy zo svojej zdravotnej karty. Je veľmi ťažké si zapamätať všetky choroby, ktoré daná osoba prekonala, alebo množstvo liekov, ktoré užívala. Ide o informácie, ktoré sú veľmi dôležité pri rôznych odborných vyšetreniach. Touto mojou aplikáciou by sa niektoré údaje pacientom sprístupnili.

Obsah

Zoznam obrázkov	8
Zoznam tabuliek	9
Zoznam symbolov a skratiek	10
Slovník termínov	12
Úvod	14
1 Formulácia úlohy	15
2 Analýza.....	17
2.1 Charakteristika vložených systémov	17
2.1.1 História vložených systémov	18
2.2 Charakteristika mobilného operačného systému Android.....	18
2.3 Medicínske systémy	21
2.3.1 Štandardy pre medicínske systémy	22
2.3.2 Zabezpečenie kvality v medicínskych projektoch	27
2.4 Súčasný stav operačného systému Android a medicínskych systémov	29
2.5 Najpoužívanějšíe medicínske aplikácie vyvinuté na platforme Android	31
3 Návrh a popis oblasti pre inovačné využitie platformy Android v medicínskych systémoch	40
4 Návrh a implementácia mobilnej aplikácie pre medicínske účely.....	46
4.1 Analýza požiadaviek	46
4.2 Návrh aplikácie.....	48
4.3 Platformy použité pri vývoji aplikácie	50
5 Možnosti integrácie databázovej časti.....	52
6 Záver.....	56
Zoznam použitej literatúry	58
Prílohy	61

Zoznam obrázkov

Obr. 1 Architektúra OS Android [2].....	19
Obr. 2 V – model [11][12][13]	28
Obr. 3 Rozšírenosť mobilných operačných systémov[15]	29
Obr. 4 Rozšírenosť verzií Android [14].....	30
Obr. 5 Aplikácia Epocrates pre OS Android [16].....	32
Obr. 6 Aplikácia Skyscape pre OS Android [21]	32
Obr. 7 Aplikácia PubMed Mobile pre OS Android [18]	33
Obr. 8 Aplikácia CPR – Choking pre OS Android [23]	33
Obr. 9 Aplikácia Medscape pre tablety s OS Android [24].....	34
Obr. 10 Aplikácia MicroMedex pre tablety s OS Android [27]	35
Obr. 11 Aplikácia Kalkulačka by QxMD pre tablety s OS Android [28]	35
Obr. 12 Aplikácia Cancer.net pre OS Android [29]	36
Obr. 13 Aplikácia AHRQ ePSS pre OS Android [30].....	36
Obr. 14 Aplikácia Draw MD Series predstavujúca medicínske produkty [31]	37
Obr. 15 Aplikácia InMotion znázorňujúca stavbu rôznych častí tela [32]	38
Obr. 16 Pridanie pacienta do zoznamu	41
Obr. 17 Aplikácia v čakárni	42
Obr. 18 Aplikácia v ambulancii	42
Obr. 19 Myšlienková mapa Informačného systému pre nemocnicu	43
Obr. 20 Pokračovanie myšl. mapy – rozhranie sestrička.....	44
Obr. 21 Pokračovanie myšl. mapy – rozhranie lekár.....	45
Obr. 22 Pokračovanie myšl. mapy - rozhranie pacient.....	45
Obr. 23 Model používateľa lekár	49
Obr. 24 Model používateľa pacient	50
Obr. 25 Spôsob posielania a prijímania dát	54
Obr. 26 Logická schéma	54
Obr. 27 Fyzická schéma.....	55

Zoznam tabuliek

Tab. 1 Normy patriace k všeobecným požiadavkám pre bezpečnosť[22].....	26
Tab. 2 Špecifické požiadavky pre bezpečnosť [22].....	26
Tab. 3 Informácie o verziách OS Android a ich rozšírenosť vzhľadom na graf na Obr. 4[14].....	30
Tab. 4 Zoznam všetkých požiadaviek.....	47
Tab. 5 Zoznam požiadaviek lekára	47
Tab. 6 Zoznam požiadaviek pacienta	47
Tab. 7 Metódy v architektúre REST	53

Zoznam symbolov a skratiek

API	Application Programming Interface – rozhranie pre programovanie aplikácií
JDK	Java Development Kit – skupina nástrojov pre vývojárov v jazyku Java
MIT	Massachusetts Institute of Technology - Massachusettský technologický inštitút
2D	skratka výrazu two dimensional space, je to „dvojdimenziálny“, alebo „dvojrozmerný“ svet,
3D	skratka výrazu three dimensional space, je to „trojdimenziálny“, alebo „trojrozmerný“ svet
GUI	Graphical User Interface - grafické užívateľské rozhranie
MPEG4	Motion Picture Experts Group – skupina štandardov pre videa a pesničky
JPG	Joint Photographic Experts Group – spojenie skupín pre odborné fotografie, alebo obrázky
PNG	Portable Network Graphics – grafický formát
AAC	Advanced Audio Coding - zvukový štandard
H.264	Advanced Video Coding - štandard pre kompresiu videa
GNU	General Public License / GNU's Not Unix - je počítačový slobodný operačný systém projektu GNU.
iOS	iPhone Operating System – mobilný operačný systém od spoločnosti Apple
OHA	Open Handset Alliance – skupina výrobcov mobilných telefónov
AOSP	Android Open Software Project – dohoda o rozvíjaní operačného systému Android
ISO	International Standard Organization – Medzinárodná organizácia noriem
IEC	International Electro technical Commission - Medzinárodná elektrotechnická komisia
ITU	International Telecommunication Union - Medzinárodná telekomunikačná únia
CEN	Comité Européen de Normalisation – Európsky výbor pre normalizáciu
CENELEC	Comité Européen de Normalisation Électrotechnique – Európsky výbor pre normalizáciu v elektrotechnike

ETSI	The European Telecommunications Standards Institute - Európsky inštitút pre telekomunikačné normy
FDA	The Food and Drugs Administration – Agentúra amerického ministerstva zdravotníctva a sociálnych služieb
GMP	FDA Good Manufacturing Practices – FDA zameraná na dobré výrobné praktiky
QSR	FDA Quality System Regulations – FDA zameraná na nariadenia systému kvality
QSIT	FDA The Quality System Inspection Technique – Príručka techník na dodržiavanie systému kvality
MDD	European Medical Device Directive 93/42/EEC - európska smernica pre medicínske zariadenia
CFR	The Code of Federal Regulations – kód federálnych predpisov
QSIT	The Quality System Inspection Technique – Technika kontroly systému kvality
SQL	Structured Query Language – štruktúrovaný vyhľadávací jazyk
SOAP	Simply Object Access Protocol – jednoduchý objektovo prístupový protokol
XML	Extensible Markup Language – rozšíriteľný značkovací jazyk
XML-RPC	Extensible Markup Language –Remote Procedure Call - rozšíriteľný značkovací jazyk – Vzdialený prístup k procedúram
REST	Representational State Transfer – architektúra rozhrania navrhnutá pre distribuované prostredia
HTTP	Hypertext Transfer Protocol – hypertextový internetový protokol
URL	Uniform Resource Locator – jednotný lokalizátor
HTML	Hyper Text Markup Language – hypertextový značkovací jazyk
GIF	Graphics Interchange Format – grafický formát
MIME	Multipurpose Internet Mail Extensions – internetový rozširujúci štandard elektronickej pošty
CRUD	Create Read Update Delete – základné operácie v SQL databázach - vytvorenie, čítanie, úprava a zmazanie
iPad	iPhone Pad – iPhone tablet od firmy Apple
iOS	iPhone Operating System – iOS operačný systém od firmy Apple

Slovník termínov

Android - operačný systém pre mobilné zariadenia od spoločnosti Google

Smartphone - označenie multifunkčných telefónov

Bit mapa - počítačový formát pre ukladanie rastrovej grafiky.

Application Programming Interface - rozhranie, ktoré obsahuje funkcie a triedy, ktoré určujú akým spôsobom sa majú funkcie knižníc volať zo zdrojového kódu programu

Java Development Kit - produkt spoločnosti Oracle, je to súbor základných nástrojov pre vývoj aplikácií pre platformu Java.

Massachusettský technologický inštitút - nazývaný aj Massachusettská technická univerzita v meste Cambridge

Dvojmerný svet – označuje svet, ktorý je možné popísať dvoma rozmermi predmety v dvojmernom svete majú obsah a napr. dĺžku alebo šírku, ale nemajú objem.

Trojmerný svet - označuje svet, ktorý je možné popísať tromi rozmermi predmety v trojmernom svete majú objem.

Graphical User Interface - Grafické užívateľské rozhranie je užívateľské rozhranie, ktoré umožňuje ovládať počítač pomocou interaktívnych grafických ovládacích prvkov

Motion Picture Experts Group - je kolekcia patentovaných metód definujúcich kompresiu a uloženie zvukových a obrazových dát.

Motion Picture - je formát stratovej kompresie zvukových súborov založený na kompresnom algoritme MPEG (Motion Picture Experts Group).

Joint Photographic Experts Group - je štandardná metóda stratovej kompresie používanej pre ukladanie počítačových obrázkov v reálnej fotografickej kvalite.

Portable Network Graphics - je grafický formát určený pre bezstratovú kompresiu rastrovej grafiky.

Advanced Audio Coding - je štandard pre stratovú kompresiu zvuku.

H.264 je štandard pre kompresiu videa, ekvivalentom pre MPEG-4 part 10 a MPEG-4 AVC.

GNU's Not Unix - je počítačový slobodný operačný systém projektu GNU.

Open Handset Alliance – skupina výrobcov mobilných telefónov, telekomunikačných operátorov a technologických firiem, ktoré vyvíjajú operačný systém Android pre mobilné zariadenia

Android Open Software Project – dohoda o rozvíjaní operačný systém Android

Structured Query Language – štruktúrovaný jazyk určený na riadenie bázy dát

Simply Object Access Protocol – protokol určený na výmenu správ cez sieť

Extensible Markup Language – jazyk určený na výmenu dát medzi aplikáciami a na zverejňovanie dokumentov

Representational State Transfer – architektúra rozhrania navrhnutá pre distribuované prostredia

Hypertext Transfer Protocol – internetový protokol určený na výmenu hypertextových dokumentov vo formáte HTML

Uniform Resource Locator – reťazec znakov s definovanou štruktúrou, ktorý slúži na špecifikáciu umiestnenia zdrojov informácií

Hyper Text Markup Language – značkovací jazyk slúžiaci na vytváranie internetových stránok

Graphics Interchange Format – grafický formát určený pre rastrovú grafiku

Multipurpose Internet Mail Extensions – internetový štandard, ktorý prostredníctvom elektronickej pošty posielajú textové správy s diakritikou

CRUD - Create Read Update Delete – základné operácie v SQL databázach - vytvorenie, čítanie, úprava a zmazanie

Google Play - internetový obchod od firmy Google, slúži na sťahovanie voľne dostupných, ale aj platených aplikácií bežiacich pre operačný systém Android

iTunes - internetový obchod od firmy Apple, slúži na sťahovanie voľne dostupných, ale aj platených aplikácií bežiacich na operačnom systéme iOS

X-ray zariadenia – sú to zariadenia určené na vysielanie elektromagnetického žiarenia

Widget – miniaplikácia – je to prvok ktorý je prispôsobený tak aby bol čo najjednoduchší na používanie

Úvod

Diplomovú prácu som si vybrala na základe skúseností s vývojom pre operačný systém Android. S týmto systémom som sa zoznámila počas spracovávania bakalárskej práce, ktorej názov bol databázové aplikácie pre mobilné zariadenia. Úlohou bolo vytvoriť jednoduchý informačný systém s databázou.

Na začiatku práce som priblížila pojem vložené systémy a ich históriu. Následne som uviedla základné informácie o mobilnom operačnom systéme Android, kde som sa zamerala hlavne na jeho architektúru.

Ďalej som opísala vlastnosti medicínskych systémov a ich vývoja. Existuje niekoľko typov softvérových produktov pre medicínu. Všetky sa musia riadiť prísne stanovenými štandardmi, ktorým je venovaná kapitola 2.3.1. Tieto normy riadi viacero európskych a medzinárodných organizácií, ktoré sú v práci tiež spomenuté.

Kapitola číslo 2.4 je venovaná súčasnému stavu operačného systému Android. Tento stav je zobrazený na rôznych štatistikách, ktoré sa zaoberajú rozšírenosťou v porovnaní s inými operačnými systémami. Aktuálnu situáciu tohto systému je možné vidieť aj na rozšírenosti jednotlivých verzií Androidu. Súčasný stav systému v medicínskej oblasti je možné predstaviť napríklad pri prehľade najznámejších a najpoužívanejších aplikácií.

Zaujímavou témou je aj návrh a popis oblastí pre inovačné využitie tohto mobilného systému v medicínskej oblasti. Tu je uvedená logika navrhovaných aplikácií a príklad grafického používateľského rozhrania, ako by mohla vyzerat'. Ide o jednu menšiu aplikáciu, určenú na zlepšenie organizácie v čakárňach u jednotlivých lekárov. Druhá je obširnejšia. Jej úlohou je riadiť systém veľkej zdravotníckej prevádzky ako je nemocnica alebo poliklinika. Po charakterizovaní základných pojmov a štandardov, ktoré sa týkajú vývoja takýchto produktov nasleduje návrh vlastnej aplikácie, kde sú bližšie popísané modely jednotlivých používateľov.

Osobitná kapitola je venovaná integrácii databázovej časti, ktorej súčasťou sú aj entitno - relačné modely pre databázu. Posledná časť sa zaoberá rôznymi možnosťami integrácie s inými aplikáciami. V závere sú uvedené výhody takéhoto systému a možné rozšírenia.

1 Formulácia úlohy

Cieľom diplomovej práce bolo naštudovať si základy vývoja medicínskeho softvéru a zhotovenie medicínskej aplikácie implementovanej na mobilnej platforme Android. Spracovanie celej práce sa opieralo o tieto hlavné body:

1. Súčasný stav systému Android vo vložených a medicínskych systémoch
2. Návrh a popis oblasti pre inovačné využitie platformy Android v medicínskych systémoch.
3. Návrh a implementácia mobilnej aplikácie pre medicínske účely.
4. Vytvorenie databázy používateľov a jej integrácia s aplikáciou
5. Vypracovať dokumentáciu podľa pokynov vedúceho.

Prvý bod práce je zahrnutý v analýze v kapitole číslo 2. Obsahuje charakteristiku vložených systémov a mobilného operačného systému Android. V tejto kapitole je popísaná aj architektúra operačného systému Android. Súčasný stav Androidu a jeho verzií je uvedený v štatistikách a predstavený na medicínskych aplikáciách, ktoré sú populárne aj na iných mobilných operačných systémoch. Vývoj medicínskeho softvéru prebieha podľa štandardov a noriem, na ktoré dohliadajú rôzne organizácie. V kapitole 2.3.1 sú uvedené základné informácie o dôležitých štandardoch v medicíne ako napríklad ISO 14971. Táto norma sa zaoberá predchádzaním rizík počas vývoja medicínskeho softvéru.

Druhý bod práce je popísaný v kapitole číslo 3, kde je navrhnutých a popísaných viacero inovačných oblastí v zdravotníctve, v ktorých by mohlo byť použitie operačného systému Android nápomocné. Napríklad systém slúžiaci na organizáciu poradia pacientov v čakárňach. Tento systém má navrhnuté aj používateľské grafické rozhranie.

Tretí bod práce je zameraný na návrh a implementáciu vlastnej medicínskej aplikácie pre mobilné zariadenia, ktorá rieši systém elektronickej zdravotnej karty. Ide o inovačné využitie platformy Android. Aplikácia by bola prístupná lekárom aj pacientom. Návrh tvoria zoznamy požiadaviek a modely prípadov použitia. Nachádzajú sa tam aj detailnejšie informácie o použitých vývojových prostriedkoch využitých pri implementácii aplikácie.

Štvrtý bod obsahuje využité možnosti pri integrácií databázovej časti s aplikáciou. Sú tam uvedené aj modely nevyhnutné na vytvorenie databázy.

Piaty bod tvoria tieto prílohy:

- Používateľská príručka k ukážke aplikácie – obsahuje návod na použitie pre používateľov aplikácie - lekára a pacienta
- Systémová príručka k ukážke aplikácie – slúži pre vývojárov
- Metodická príručka k ukážke aplikácie – obsahuje metodiky pre vytvorenie aplikácie na platforme Android

2 Analýza

Počas posledných rokov je možné vnímať rýchle pokroky mobilných operačných systémov a zariadení. Inovuje sa a zlepšuje nielen ich funkcionálnosť, ale aj ich rozšírenie v rôznych priemyselných oblastiach. Smartphony už nie sú považované za obyčajné telefóny určené ľuďom, len na komunikáciu medzi sebou, ale hlavne slúžia na uľahčenie a zefektívnenie práce. Pomerne veľká časť ľudí ho využíva kvôli rôznym burzám, prepočtom a cenníkom. Ďalším slúži ako pomôcka pri dodržiavaní predpísaných pravidiel na cestách v rôznych krajinách, ale aj ako navigácia. V svojej práci by som chcela ukázať aj na možnosť využitia softvéru na prenosných zariadeniach v oblasti zdravotníctva.

Základné informácie o vývoji medicínskych softvérových produktov som získala v spoločnosti Siemens v Košiciach. Vďaka tomu som sa viac zaujímala o túto tému.

Vývoj medicínskych aplikácií pre mobilné zariadenia je v zahraničí veľmi rozšírený. Najviac takýchto aplikácií je od firmy Apple, ktorá je vlastníkom operačného systému iOS.

Cieľom mojej práce je navrhnuť a implementovať program, ktorý slúži na uľahčenie a zjednodušenie práce lekára s databázou pacientov. Na vývoj aplikácie je použitý programovací jazyk Java a vývojové prostredie Eclipse.

2.1 Charakteristika vložených systémov

Vloženým systémom (anglický názov – embedded system), nazývame jednoúčelový systém, v ktorom je riadiaci počítač zabudovaný priamo do zariadenia, ktoré riadi. Na rozdiel od klasických počítačových systémov, vložené systémy sú určené väčšinou na to, aby riešili jednu zadanú úlohu. Každý takýto systém sa skladá zo softvérovej a hardvérovej časti. Sú vyrábané v nespočetných množstvách, často ide o sériovú výrobu. Takýmito produktmi sú napríklad kalkulačky, bankomaty, pevné telefónne linky a ďalšie. Za vložený systém bol považovaný aj mobilný telefón. Avšak s postupom času patrí skôr na hranicu medzi viacúčelovými a vloženými systémami. Dôvodom je to, že sa stále viac vyrovná v mnohých funkciách už osobným počítačom.[1]

2.1.1 História vložených systémov

Za prvý moderný vložený systém je považovaný riadiaci počítač pre Apollo, ktorý zostrojil Charles Stark Draper v prístrojovom laboratóriu MIT (Massachusettský technologický inštitút). Tento navádzací systém bol považovaný za najnebezpečnejšiu časť celého projektu Apollo. Vyvinutím integrovaných obvodov, ktoré znížili hmotnosť a veľkosť sa zvýšilo riziko zlyhania.

Prvý systém, ktorý sa začal hromadne vyrábať bol určený na vojenské účely. Vojenský navádzací systém D-17 bol vyvinutý firmou Autentic v roku 1961 a bol určený špeciálne pre rakety s názvom Minuteman. Tvorili ho tranzistory. Hlavná pamäť využívala už pevný disk.

V prvých vložených systémoch ako boli kalkulačky sa využíval aj prvý mikroprocesor Intel 4004, ktorý vyžadoval vnútornú pamäť a ďalšie podporné obvody.

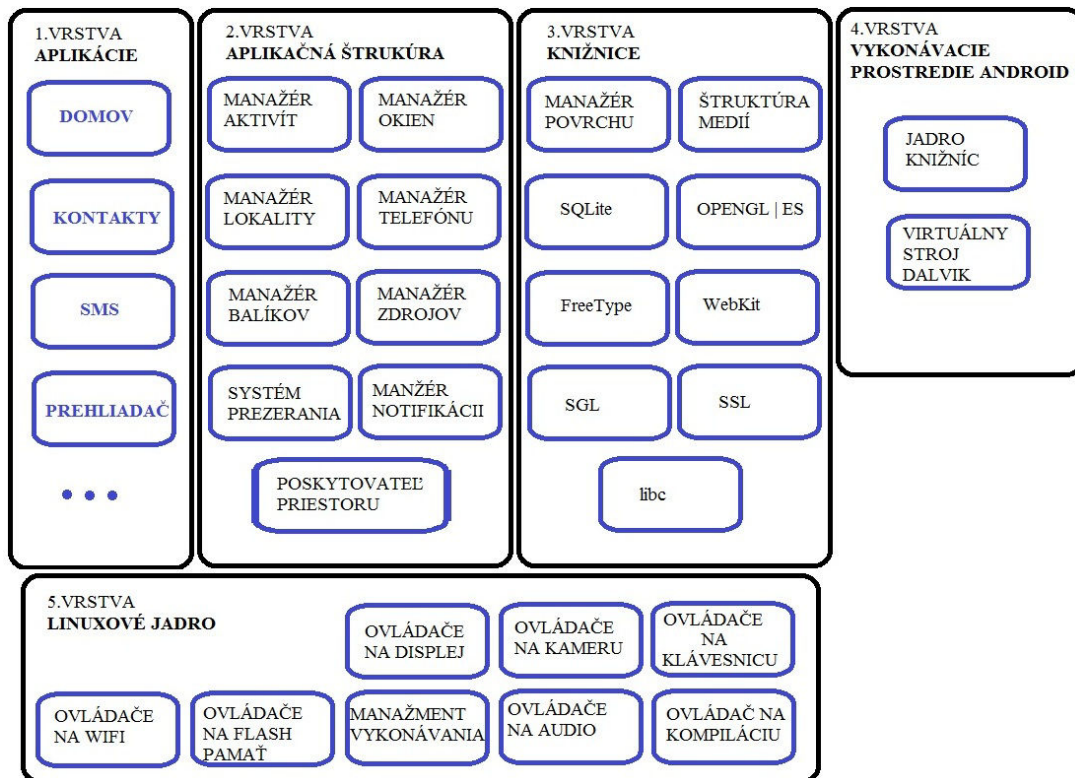
V polovicike 80-tych rokov bola väčšina externých súčiastok integrovaná na čipe spolu s procesorom. Takáto súčiastka sa nazývala jedno čipový počítač.

Vložené systémy sa naďalej zdokonaľujú a využívajú v mnohých starých aj nových elektronických zariadeniach. [1]

2.2 Charakteristika mobilného operačného systému Android

Platforma Android bola vyvinutá firmou Android Inc. Neskôr v roku 2005 ju odkúpila spoločnosť Google Inc., ktorá ju v roku 2007 uviedla ako AOSP. AOSP – Android open source project bol navrhnutý alianciou OHA(Open Handset Alliance). OHA je to združenie 86 hardvérových, softvérových a telekomunikačných spoločností ktorá sa zaoberá presadzovaním voľne dostupných štandardov pre mobilné zariadenia. Cieľom AOSP je udržiavať a rozvíjať operačný systém Android. Tento systém je určený pre mobilné zariadenia a tablety.

Architektúru systému tvorí päť častí (Obr. 1): aplikácie, aplikačná štruktúra, knižnice, vykonávacie prostredie OS Android a linuxové jadro.



Obr. 1 Architektúra OS Android [2]

Prvú časť tvoria základné aplikácie ako kontakty, informácie o telefóne, správy, internetový prehliadač a ďalšie. Tieto aplikácie sú v telefóne už pri prvom spustení zariadenia. Používateľ si môže pridať ďalšie aplikácie podľa vlastného výberu, ako napríklad nový internetový prehliadač, správca úloh a podobne.

Aplikačná štruktúra (angl. Application framework) je časť, ktorá je zaujímavá pre vývojárov. Sprístupňuje im množstvo služieb, ktoré využívajú iné aplikácie. Tieto služby poskytujú dáta umožňujúce spustenie rôznych procesov v pozadí, nastavenie notifikácií v stavovom paneli a veľa ďalších funkcií. Architektúra systému je prispôbená tak, aby mohla znova používať komponenty iných aplikácií.

Patria tu napríklad:

- Manažér aktivít, ktorý sa stará o manipuláciu s aplikáciami a o ich životný cyklus.
- Manažér okien, ktorý disponuje širokou škálou okien a zložiek, vďaka ktorým je možné vytvoriť grafické používateľské rozhranie aplikácií pomocou okien, tlačidiel, textových polí, zaškrávacích políček atď.
- Poskytovateľ priestoru (angl. Content provider) poskytuje prístup k dátam iných aplikácií.

-
- Manažér telefónu obsahuje informácie o telefóne, vyhľadáva nové aktualizácie.
 - Manažér lokality je zodpovedný za nájdenie a určenie polohy.
 - Manažér notifikácií umožňuje nastaviť panel, kde sa nachádzajú notifikácie, poznámky a statusy.

Tretiu časť tvoria knižnice programovacích jazykov C a C++. Sú dôležité hlavne pre vývojárov, ktorí k nim majú prístup pomocou aplikačnej štruktúry.

Patria tu:

- Manažér povrchu (angl. Surface manager)- je zodpovedný za vytváranie grafických zložiek a to skladaním 2D a 3D grafických zložiek.
- Multimediálna knižnica, ktorej základ tvorí Open CORE. Zodpovedá za prehrávanie audio a video záznamov, ale aj obrázkov, ktoré sú formátu – MPEG 4, MP3, JPG, PNG, AAC, H.264
- SQLite, ktorý slúži ako odľahčený databázový systém.
- OpenGL určený pre prácu s 3D grafikou.
- SGL je určený pre prácu s 2D grafikou.
- FreeType knižnica v jazyku C disponuje s veľkým počtom formátov písma. Využíva sa na vykreslenie textu do bit mapy.
- WebKit bol navrhnutý pre sprístupnenie webových prehliadačov.
- SSL je kryptografický protokol, ktorý zabezpečuje komunikáciu cez internet.
- Libc je systémová štandardná knižnica písaná v C, je upravená pre vložené systémy.

Ďalšou dôležitou vrstvou je vykonávacie prostredie operačného systému Android. Toto prostredie tvorí virtuálny stroj Dalvik (Dalvik Virtual Machine) a súbor základných knižníc (angl. Core libraries). Dalvik je registrovo orientovaná architektúra, ktorá bola vytvorená špeciálne pre Android v roku 2005. Všetky aplikácie sú písané v programovacom jazyku Java. Ich preklad prebieha tak, že kód aplikácie je skompilovaný do Java binárneho kódu, ktorý sa následne prekompiluje pomocou prekladača Dalvik. Výsledkom je binárny kód prekladača Dalvik, ktorý je spustený na virtuálnom stroji Dalvik. Táto časť obsahuje knižnice programovacieho jazyka Java, okrem knižníc používateľského rozhrania ako AWT a Swing, pretože tie musia byť špeciálne pre Android. Tiež obsahuje knižnice, ktoré slúžia na prácu so sieťou.[3]

Najdôležitejšou vrstvou je linuxové jadro, ktoré tvorí základ celého operačného systému. Jadro dokáže spravovať jednotlivé procesy, pamäte a siete. Taktiež zahrňuje vlastnosti na podporu ovládačov a spustenie viacerých aplikácií súčasne. Nepodporuje používateľské rozhranie X Window System a ani celú skupinu knižníc GNU. Výhodou využitia jadra založeného na Linuxe je jeho jednoduchá kompilácia na ľubovoľnom zariadení. [2][4]

2.3 Medicínske systémy

Medicínske systémy (angl. Medical systems) sú počítačové systémy špecializované na zdravotníctvo. Medicína patrí medzi najdôležitejšie odbory na svete. Vývoj aplikácií, ktoré majú slúžiť v medicíne by mali zodpovedať prísnyim kritériám. Súčasný stav vývoja takýchto aplikácií je pomerne dosť rozšírený na osobných počítačoch, ale pomaly sa prechádza aj na mobilné zariadenia. Medzi mobilné zariadenia sa radia mobily - Smartphony a tablety.

Smartphony sú mobilné zariadenia s mnohými funkciami určené hlavne na rýchlu komunikáciu. Viacerými funkciami sa vyrovnajú osobnému počítaču. Každé takéto zariadenie má v sebe zabudovanú wifi kartu, ktorá umožňuje bezdrôtové internetové spojenie. Pomocou tohto pripojenia je operačný systém pravidelne aktualizovaný. Vlastník telefónu si môže vytvoriť prostredie, ktoré obsahuje pre neho nevyhnutné aplikácie.

Vývoj softvéru v medicínskom odbore v zahraničí je na vyššej úrovni ako na Slovensku.

Základné rozdelenie systémov v medicínskej oblasti:

- Aplikácie, ktoré sú určené pre lekárov v ambulanciách.
- Softvér, ktorý riadi medicínske zariadenia
- Iné programy, ktoré sú voľne dostupné ľuďom.

V prvú skupinu tvoria predovšetkým databázové aplikácie. Napríklad elektronické zdravotné karty s osobnými záznamami o pacientoch a ich zdravotnom stave. Ide hlavne o veľkú databázu, do ktorej má prístup len lekár. Ten by si mal viesť podrobné záznamy o prehliadkach a všetkých chorobách každého pacienta. Na Slovensku sa už takýto typ aplikácie využíva. Ďalšia aplikácia, ktorá je dostupná len doktorom je databáza liekov. Ide o aplikáciu, ktorá obsahuje všetky lieky, ktoré môže predpísať len lekár. Taktiež by

mala obsahovať aj lieky, ktoré môžu pacientovi pomôcť a sú voľne predajné v lekárni. Takáto databáza musí byť pravidelne aktualizovaná.

Druhú skupinu tvorí softvér, ktorý je určený pre zariadenia, ktoré sa využívajú v zdravotníctve. Ide napríklad o rôzne onkologické, rádiologické a ďalšie prístroje. Takýto softvér musí spĺňať štandardy (bližšie v podkapitole 2.3.1) a musí byť pravidelne kontrolovaný.

Posledná skupina obsahuje programy, ktoré sú dostupné každému. Patria tu programy, ktoré slúžia na organizovanie podávania liekov, ale aj rôzne aplikácie, ktoré dokážu merať tep, či počítať energetickú hodnotu jedál a mnoho ďalších. Sú to aplikácie, ktoré je možné využívať každý deň. Využíva ich veľa ľudí pri chudnutí, pri športovaní, využívajú ich dokonca aj študenti na zdravotných školách. Nedávno vyšla aj slovenská aplikácia s názvom Zdravotnícky sprievodca. Táto aplikácia vie podľa aktuálneho miesta nájsť v databáze najbližších odborných lekárov, zubárov, nemocnice, polikliniky, lekárne a veterinárov.

2.3.1 Štandardy pre medicínske systémy

Každý štát má organizáciu, ktorá sa zaoberá normalizáciou. Všetky členské národné normalizačné orgány majú zastúpenie aj v medzinárodných komisiách, ktoré riadia činnosti súvisiace s normami. Norma je dokument, ktorý bol schválený po dohode príslušným orgánom. Dokument obsahuje charakteristiku, postupy činností a ich výsledky po dosiahnutí optimálnosti. Normy sa rozlišujú podľa charakteru, predmetu a spôsobu šírenia. Ich hlavnou úlohou je zabezpečiť odstránenie problémov, ktoré sa vyskytujú medzi rôznymi činnosťami v odlišných odvetviach a obchodoch. Medzinárodné normy sú výsledkom spolupráce rôznych orgánov a sú schválené dohodou reprezentujúcou spoločné záujmy rozličných oblastí. Vyvíjajú sa spolu s technologickým a sociálnym pokrokom.

Vývoj medicínskeho softvéru, zariadení a samotných produktov je riadený pomocou štandardov a noriem, ktorými sa zaoberajú medzinárodné organizácie normalizácie, regionálne organizácie normalizácie v Európe a národná organizácia normalizácie.

Medzinárodné organizácie:

- ISO – Medzinárodná organizácia pre normalizáciu

-
- Organizácia, ktorá riadi technickú činnosť zaoberajúcou sa štandardmi a normami. Tvorí ju 164 krajín, za každú jeden zástupca.[7]
 - IEC – Medzinárodná elektrotechnická komisia
 - svetová uznávaná organizácia, ktorá pripravuje a vydáva medzinárodné normy pre elektrické, elektronické a ďalšie technológie, ktoré obsahujú alebo produkujú elektrinu. [6]
 - ITU – Medzinárodná telekomunikačná únia
 - Špecializovaná organizácia OSN pre informačné a telekomunikačné technológie. Tvorí ju 3 hlavné oblasti zamerané na rádio komunikáciu, normalizáciu v telekomunikácii a vývoj telekomunikácií. [5]

Regionálne organizácie normalizácie v Európe

- CEN - Európsky výbor pre normalizáciu - nezisková organizácia, ktorá tvorí európske normy v rôznych oblastiach. Úlohou je vybudovanie jednotného trhu pre tovar a služby.
- CENELEC - Európsky výbor pre normalizáciu v elektrotechnike - zodpovedá za štandardy v oblasti elektrotechniky.
- ETSI – Európsky inštitút pre telekomunikačné normy - organizácia, ktorá vydáva európske normy elektronických komunikácií. [8]

Hardvér, softvér a konečné produkty v medicíne sú riadené viacerými normami. Medzi najdôležitejšie normy patria:

FDA (skratka z angl. Food and Drugs Administrations) je agentúra amerického ministerstva zdravotníctva a sociálnych služieb.

Zodpovedá za

- ochranu verejného zdravia zabezpečením toho, že všetky potraviny sú bezpečné, zdravé, sanitárne (hygienické) a poriadne označené
- zabezpečuje že ľudské a veterinárne liečivá, vakcíny a ďalšie biologické a zdravotnícke pomôcky určené pre ľudskú spotrebu sú bezpečné a majú daný účinok
- ochranu verejnosti pred radiáciou z elektronických produktov

-
- zabezpečenie kozmetiky a potravinových doplnkov tak, aby boli bezpečné a správne označené
 - zrýchlenie produktových inovácií, ktoré vedie k zvyšovaniu verejného zdravia
 - reguláciu tabakových výrobkov
 - pomoc pri získavaní vedecky podložených informácií, ktoré sú potrebné pre využitie v medicíne aj pre zariadenia a potraviny za účelom zlepšenia zdravia

FDA reguluje

- potraviny s výnimkou väčšiny mäsových a hydinových produktov, ktoré sú regulované americkým ministerstvom poľnohospodárstva
- prídavné látky v potravinách
- dojčenské výživy
- doplnky stravy
- ľudské lieky
- vakcíny, krvné produkty a ďalší biologický materiál
- medicínske zariadenia, od jednoduchých - používaných pri jednoduchých lekárskych zákrokoch napríklad pri výbere mandlí až po zložitejšie zariadenia ako srdcové kardiostimulátory
- elektronické produkty, ktoré vyžarujú radiáciu tak ako mikrovlnné rúry a X-ray zariadenia
- kozmetiku
- krmivá, lieky a zariadenia určené pre domáce zvieratá, hospodárske zvieratá a iné zvieratá
- tabakové výrobky

Ďalšie štandardy uvedené v rámci FDA 21 CFR 820:

FDA GMP (skratka z angl. FDA Good Manufacturing Practices) - hovorí o zavedení a udržiavaní dobrých výrobných praktík kvalitného systému

FDA QSR (skratka z angl. FDA Quality System Regulations) – nariadenia systému kvality sú dôležité pre výrobcov. QSR definuje presné postupy a požiadavky určené pre

zabezpečenie hotových zariadení, tak aby vyhovovali zákonom o potravinách, liekoch a kozmetike.

FDA QSIT (skratka z angl. The Quality System Inspection Technique) – verejná príručka určená pre priemysel medicínskych zariadení a ďalších priemyselných operácií spadajúcich pod GMP, kde vyšetrovatelia kontrolujú presne stanovený režim. Výsledkom je uľahčenie hodnotenia a udržiavania pravidiel.[9][10]

MDD (angl. Medical Device Directive) - európska smernica pre medicínske zariadenia. Cieľom smernice je minimalizovať problémy ktoré nastanú počas návrhu, výrobe a pri balení zdravotníckych pomôcok. Všetky produkty vyhovujúce tejto smernici majú označenie CE. Takýmto označením disponuje len ten, ktorý prešiel radom hodnotení, skúškami kvality systému, skúškou produktového typu na základe navrhnutých cieľov, ktoré má výsledný produkt splňať.

ISO 14971 – medzinárodný štandard reprezentujúci požiadavky pre riziká manažmentu pre zdravotnícke zariadenia. Je uznávaný v Spojených štátoch, Európskej Únii a v Kanade. Jeho úlohou je

- overiť určené použitie výrobku,
- identifikovať a kvantifikovať možné hazardy už počas návrhu a vývojových procesov
- zníženie rizík na prijateľnú úroveň a potvrdenie jeho prijateľnosti pred uvedením na trh
- stanovenie procesov, ktoré monitorujú produkty po ich uvedení na trh a prijatie opatrenia či sú riziká väčšie ako sa predpokladalo a či vznikajú nové.

Riziká je potrebné riešiť a dohliadať na nich počas celého životného cyklu vývoja zariadenia určeného na zdravotnícke účely. Identifikácia rizík je nevyhnutná pri návrhu a vývoji.

ISO 13485 – medzinárodne uznávaná norma, ktorá definuje presne systém manažmentu kvality pomocou požiadaviek vzťahujúcich sa na návrh a výrobu zdravotníckych zariadení. Nevyhnutnou činnosťou je neustála kontrola implantovateľných prístrojov, kontrola čistoty výrobku a znečistenia. Špecifikuje požiadavky pre dokumentáciu a postupy validácie sterilných medicínskych zariadení a pomôcok. Vyžaduje sa aj norma týkajúca sa rizík – ISO 14971. Taktiež obsahuje požiadavky na overenie efektívnosti správnych a preventívnych činností.

IEC 60601 - skupina technických štandardov zameraná na bezpečnosť a efektívnosť medicínskych elektrických zariadení. Prvá časť IEC 60601 – 1 je zameraná na všeobecné požiadavky pre bezpečnosť. V tabuľke (Tab. 1) sú uvedené presné STN a ich popis. [22]

STN	POPIS
STN EN 60601 - 1 - 1	Bezpečnostné požiadavky pre elektrické zdravotnícke systémy.
STN EN 60601 - 1 - 2	Elektromagnetická kompatibilita -požiadavky a testy
STN EN 60601 - 1 - 3	Všeobecné požiadavky radiačnej ochrany v diagnostických- röntgenových zariadeniach
STN EN 60601 - 1 - 4	Programovateľné zdravotnícke elektrické systémy
STN EN 60601 - 1 - 6	Použitelnosť
STN EN 60601 - 1 - 8	Všeobecné požiadavky, testy a rady pre alarmové systémy v elektrických zdravotníckych zariadeniach a systémoch.

Tab. 1 Normy patriace k všeobecným požiadavkám pre bezpečnosť[22]

Druhá časť IEC 60601 – 2 sa zaoberá jednotlivými druhmi zariadení, ktoré patria do medicínskej oblasti. Používajú sa vždy v prepojení s prvou časťou. V tabuľke (Tab. 2) je uvedených prvých 10 noriem. Všetkých je 52. [22] [34]

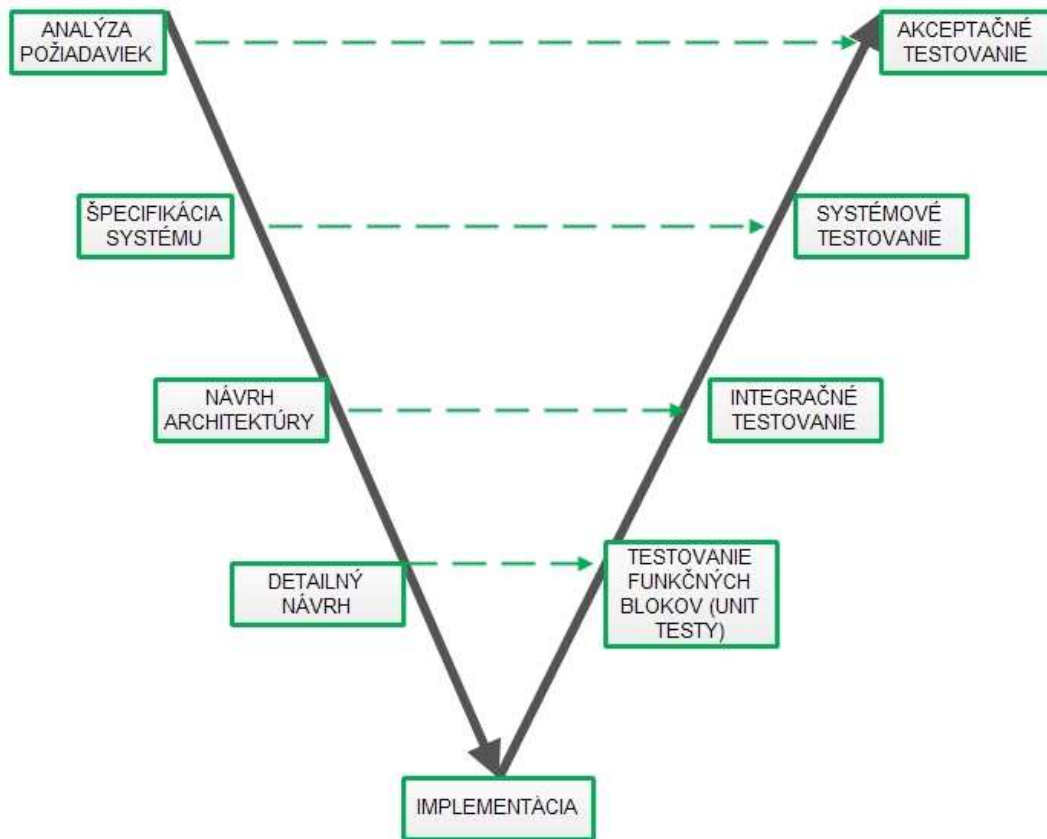
STN	POPIS
STN EN 60601 - 2 - 1	Špecifické požiadavky pre bezpečnosť elektrónových urýchľovačov v rozsahu od 1MeV do 50 MeV
STN EN 60601 - 2 - 2	Špecifické požiadavky pre bezpečnosť vysokofrekvenčných chirurgických prístrojov
STN EN 60601 - 2 - 3	Špecifické požiadavky pre bezpečnosť prístrojov na liečbu krátkymi vlnami
STN EN 60601 - 2 - 4	Špecifické požiadavky pre bezpečnosť srdcových defibrilátorov
STN EN 60601 - 2 - 5	Špecifické požiadavky pre bezpečnosť ultrazvukových fyzioterapeutických prístrojov
STN EN 60601 - 2 - 6	Špecifické požiadavky pre bezpečnosť prístrojov na liečbu mikrovlnami
STN EN 60601 - 2 - 7	Špecifické požiadavky pre bezpečnosť vysokonapäťových zdrojov diagnostických - röntgenových generátorov
STN EN 60601 - 2 - 8	Špecifické požiadavky pre bezpečnosť terapeutických röntgenových prístrojov pracujúcich v rozsahu od 10kV do 1MV
STN EN 60601 - 2 - 9	Špecifické požiadavky pre bezpečnosť kontaktných dozimetrov pacienta používaných v rádioterapii s elektricky pripojenými detektormi žiarenia
STN EN 60601 - 2 - 10	Špecifické požiadavky pre bezpečnosť nervových a svalových stimulátorov

Tab. 2 Špecifické požiadavky pre bezpečnosť [22]

2.3.2 Zabezpečenie kvality v medicínskych projektoch

Zabezpečenie kvality v medicínskych projektoch prebieha nielen pomocou dodržiavania štandardov a noriem, ale aj zvolením vhodného životného cyklu vývoja projektu. Medzi najznámejšie životné cykly vývoja patrí vodopádový model, prototypovací model, model prieskumník, špirálový model a V – model (Obr. 2). Na obrázku je znázornená štruktúra životného cyklu V – modelu. Najprv sa urobí analýza požiadaviek pre daný projekt. Požiadavky sa spíšu a zoradia podľa priority určenej zvyčajne zákazníkom. Následne sa urobí špecifikácia celého systému a navrhne sa architektúra. Potom softvéroví inžinieri spracujú detailný návrh, po ktorom je možné začať s implementáciou samotného softvérového produktu. Po naprogramovaní základných blokov je možné hneď robiť základné testy funkčných blokov a testovať, či sú splnené požiadavky. Toto testovanie sa nazýva aj biela krabica (angl. white box). Ide o testovanie kódu, o ktorom vieme, čo má vykonávať. V tejto fáze je možné postupovať aj tak, že najprv sa spravia testy na zadané požiadavky. Až potom sa píše program a následne sa spúšťajú testy. Takto sa tiež overuje, či kód splňa to, čo bolo zadané.

Keď sú naprogramované väčšie celky, kde sú už prepojenia blokov, tak sa vykonáva integračné testovanie. Potom sa vykonáva testovanie celého systému, pri ktorom sa validuje špecifikácia systému. Toto testovanie sa nazýva čierna krabica (angl. black box). Ide o testovanie funkcionality systému. Nakoniec prebieha akceptačné testovanie, ktoré je zvyčajne už v prostredí, kde reálne bude produkt nasadený.

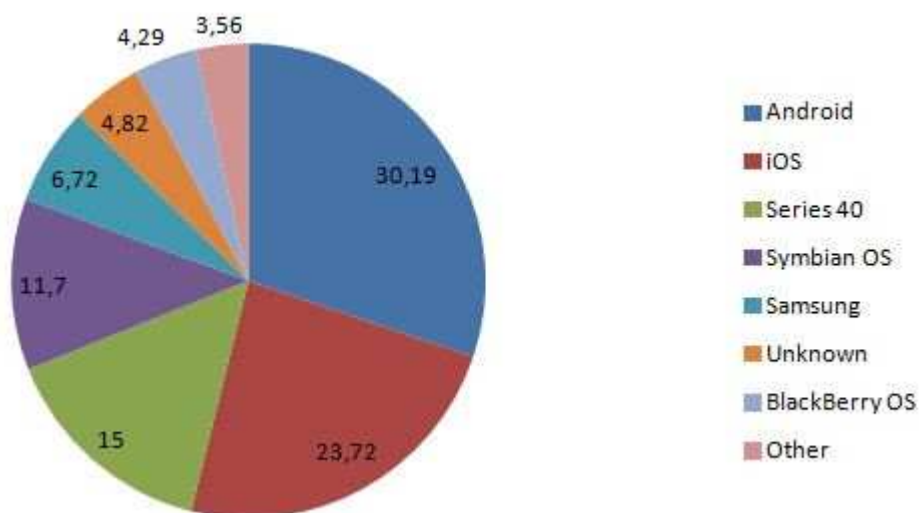


Obr. 2 V – model [11][12][13]

Testovanie je veľmi dôležité pri vývoji akéhokoľvek produktu. Spoločnosti vyvíjajúce takýto typ softvéru majú vyškolených ľudí, ktorí dozerajú na dodržiavanie štandardov a noriem. V medicínskych projektoch sú dôležité napríklad tieto štandardy: ISO 9001, ISO 14001, ISO 13485, ISO 14971, EN 62304, EN 60601-1-x, FDA-QSR 21 Part 820, CMMI. Väčšina z nich je spomenutá v podkapitole 2.3.1. Zvyčajne na jednom projekte všetci pracujúci nie sú vyškolení na všetky štandardy. Záleží to väčšinou od typu role, ktorú pracovníci vykonávajú. Významným človekom v tíme je aj manažér kvality, ktorý musí byť oboznámený s vyššie uvedenými štandardmi pre medicínske projekty. Manažér kvality sa snaží dohliadať aj na to, ktorí pracovníci by mali byť vyškolení tak, aby projekt splňal ciele kvality. Ďalšou významnou úlohou manažéra kvality je koordinovanie auditov, ktoré sú neoddeliteľnou súčasťou medicínskych softvérových projektov. Audit je druh kontroly, pri ktorom sa overuje používanie procesov v rámci projektu aj na základe predloženej dokumentácie.

2.4 Súčasný stav operačného systému Android a medicínskych systémov

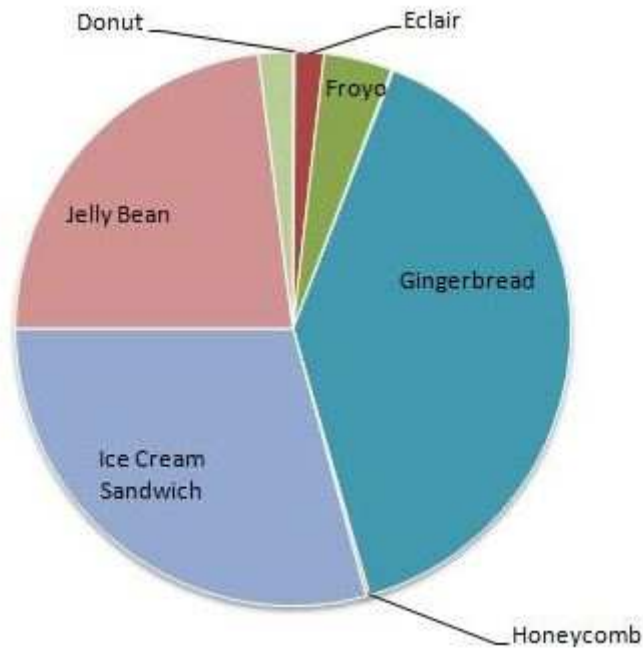
Operačný systém Android je v súčasnosti veľmi populárny. A jeho obľúbenosť naďalej rastie. Podľa štatistiky z decembra 2012 ako vidno na obrázku (Obr. 3) je najpoužívanější a hneď po ňom nasleduje operačný systém firmy Apple - iOS.



Obr. 3 Rozšírenosť mobilných operačných systémov[15]

V porovnaní štatistík z marca 2013 a minulým rokom (Obr. 4), najpoužívanějšíou verziou Androidu ostala verzia s názvom Gingerbread 2.3.3-2.3.7. Ďalšou je Ice Cream Sandwich 4.0, ktorá je využívaná aj na tabletoch. Treťou najviac rozšírenou verziou je Jelly Bean 4.1 a 4.2, tiež pomerne dosť využívanou aj na tabletoch a ešte verzia Froyo 2.2, ktorá je implementovaná na starších zariadeniach. Najnovšie telefóny a tablety majú implicitne verziu 4.x, pretože táto séria je špeciálne upravená tak, aby vývoj aplikácií pre mobily a tablety bol zjednotený. Tak sa uľahčila práca vývojárov, ktorý už nemusia vytvárať aplikáciu pre každé zariadenie zvlášť. [14]

V máji tohto roku by mala spoločnosť Google predstaviť novú verziu Androidu 5.0, s názvom Key Lime Pie. Jej cieľom je dosiahnuť vylepšenie chodu aplikácií v každej situácii. V tejto verzii si používatelia budú vedieť rozdeliť aplikácie na osobné a pracovné, s prístupom rôznych používateľských kont. Určite prinesie mnoho ďalších zaujímavých noviniek.



Obr. 4 Rozšírenosť verzií Android [14]

V Nasledujúcej tabuľke zoznam aktuálnych verzií operačného systému Android s ich špecifickými názvami, stupňom API a distribúciou. [14]

VERSION	CODENAME	API	DISTRIBUTION
1.6	Donut	4	0,1%
2.1	Eclair	7	1,7%
2.2	Froyo	8	4,0%
2.3 - 2.3.2	Gingerbread	9	0,1%
2.3.3 - 2.3.7		10	39,7%
3.2	Honeycomb	13	0,2%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	29,3%
4.1.x	Jelly Bean	16	23,0%
4.2.x		17	2,0%

Tab. 3 Informácie o verziách OS Android a ich rozšírenosť vzhľadom na graf na Obr. 4[14]

Vývoj softvérových produktov pre túto platformu umožňuje aj vývoj bez poplatkov. Integrácia s príslušným zariadením je robená veľmi jednoducho, takže je možné svoju aplikáciu vyskúšať hneď na ľubovoľnom zariadení s týmto systémom.. Systém je vytvorený tak, aby s ním vedel pracovať hocikto a preto je aj veľmi obľúbený.

2.5 Najpoužívanejšie medicínske aplikácie vyvinuté na platforme Android

Výhodou platformy Android je linuxové jadro, v rámci ktorého nie je táto architektúra náročná na hardvér a nezaťažuje procesor tak, ako iné operačné systémy. Vývoj na tejto platforme je veľmi jednoduchý. Obchod určený na sťahovanie rôznych programov sa nazýva Google Play. Z medicínskej oblasti sa ich tam nachádza cez 1000. Obľúbenosť Androidu každým rokom v rôznych oblastiach stúpa. Jeho využitie v zdravotníctve je zatiaľ, len vo vývoji v porovnaní s operačným systémom iOS od firmy Apple. V internetovom obchode od firmy Apple - iTunes existuje cez 2500 voľne dostupných medicínskych aplikácií. Sú rozdelené do dvoch kategórii. Prvá je určená pre registrovaných užívateľov. Tu sa predpokladá, že prístup majú predovšetkým ľudia, ktorí sa špecializujú v tomto odbore. Ide o odborných lekárov a študentov. Väčšina lekárov ich využíva na svojich tabletoch, ktoré majú názov iPad Druhá kategória je prístupná pre hocikoho. Nie je potrebná registrácia. Tablet je vložený systém, ktorý je možné výhodne využiť v rôznych medicínskych oblastiach.

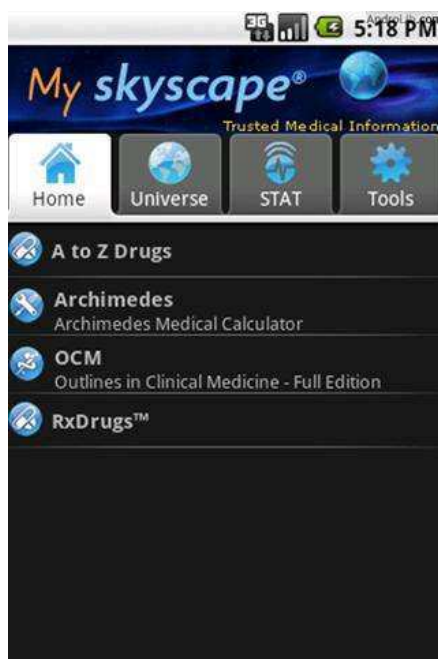
Medzi najznámejšie aplikácie [20], ktoré boli vyvinuté ako prvé na platforme Android patrí Epocrates, ktorá je vyvinutá aj pre iné systémy. Je určená predovšetkým pre ľudí, ktorí poskytujú zdravotnú starostlivosť. Poskytuje informácie o liekoch, chorobách a diagnostických nástrojoch. Dokáže kontrolovať pôsobenie viacerých liekov. Na obrázku (Obr. 5) je možné vidieť, aké možnosti ponúka po vybratí konkrétneho lieku zo zoznamu. Sú to napríklad rôzne typy dávkovania, upozornenia, kontraindikácie a nežiaduce účinky. Táto aplikácia je veľmi rozšírená a je možné ju nájsť aj zadarmo v obmedzených verziách. Epocrates bola vytvorená najskôr pre stolové počítače a až neskôr bola aplikovaná aj pre prvé Smartphony.



Obr. 5 Aplikácia Epocrates pre OS Android [16]

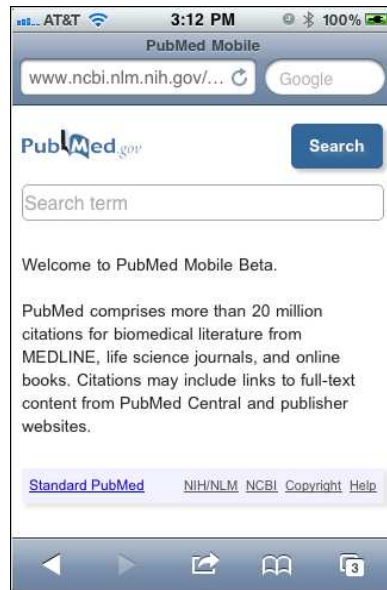
Pre spustenie tejto aplikácie je potrebná registrácia na oficiálnej stránke, ktorá má doménu ePOCRATES.[16] [17]

Ďalšou rozšírenou aplikáciou je SKYSCAPE. Slúži ako skupina medicínskych informácií a zdrojov pre podporu lepšieho rozhodovania pri liečbe. Vo voľne dostupnej verzii sú obsiahnuté informácie, ktoré je možné vidieť aj na obrázku (Obr. 6). Sú to informácie o liekoch, kalkulačka Archimedes, informácie o symptómoch a jednotlivých ochoreniach a aj medicínsky časopis MedAlert, ktorý informuje o novinkách a výsledkoch. [21]



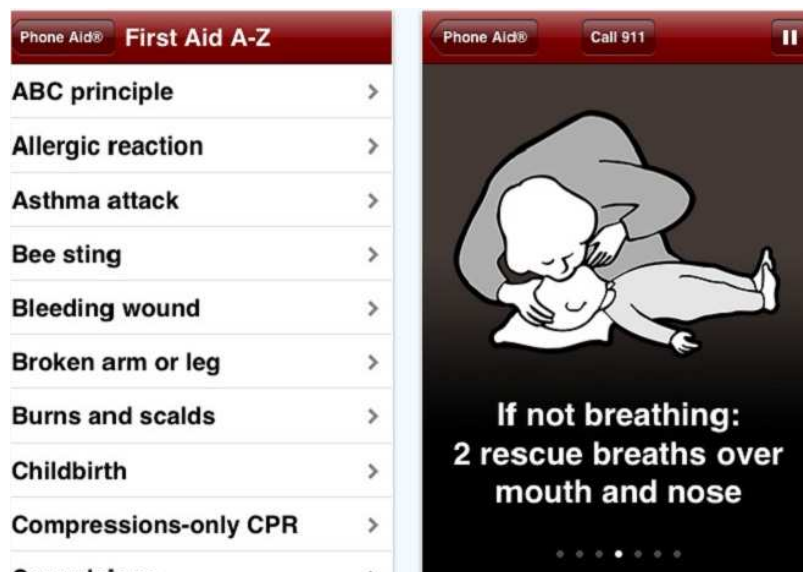
Obr. 6 Aplikácia Skyscape pre OS Android [21]

Známou medicínskou aplikáciou je aj PubMed Mobile, ktorá je zameraná na vyhľadávanie medicínskych výrazov v článkoch. Ide o databázu, v ktorej si je možné uložiť hľadané údaje. [19]



Obr. 7 Aplikácia PubMed Mobile pre OS Android [18]

CPR - Choking, je skratka anglického výrazu „cardiopulmonary resuscitation“, čo v preklade znamená kardiopulmunálna resuscitácia. Ide o voľne dostupnú aplikáciu, ktorá obsahuje videá slúžiace na výučbu základných zručností v zdravotníckej oblasti. Na obrázku (Obr. 8) možno vidieť ukážku aplikácie, kde je zoznam situácií, pri ktorých je potrebné podať prvú pomoc. Taktiež je vidieť aj ukážku videa, pri ktorom je aj popis činnosti, ktorá sa vykonáva. Vytvorili ju profesori a lekári z univerzity vo Washingtone. [23]



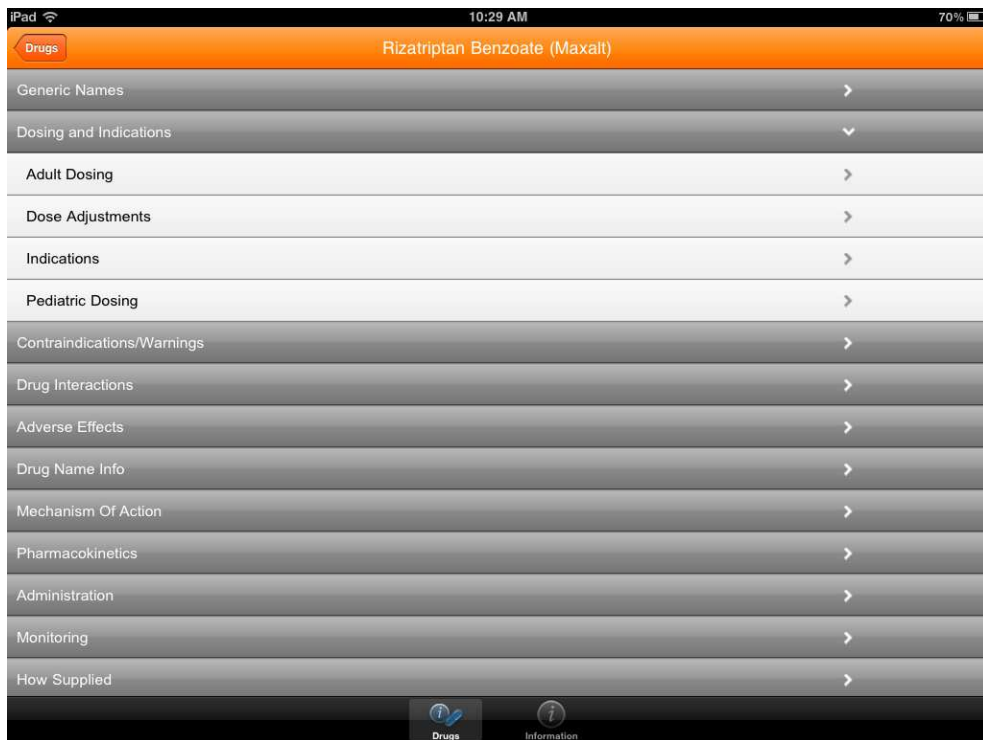
Obr. 8 Aplikácia CPR – Choking pre OS Android [23]

Najvyužívanejšie medicínske aplikácie na platforme iOS nútli autorov vyvinúť ďalšie aplikácie aj pre Android. Medzi takéto aplikácie patrí aj Medscape. Cieľom tejto aplikácie je poskytovať referencie o liekoch. Okrem širokého zoznamu liekov je možné si vybrať z rôznych ochorení a stavov. K liekom je uvedené dávkovanie, podmienky, kontraindikácie a ďalšie dôležité informácie, ktoré bývajú priložené pri liekoch. Na obrázku (Obr. 9) je možné vidieť úvodnú obrazovku a menu tejto aplikácie. [25]



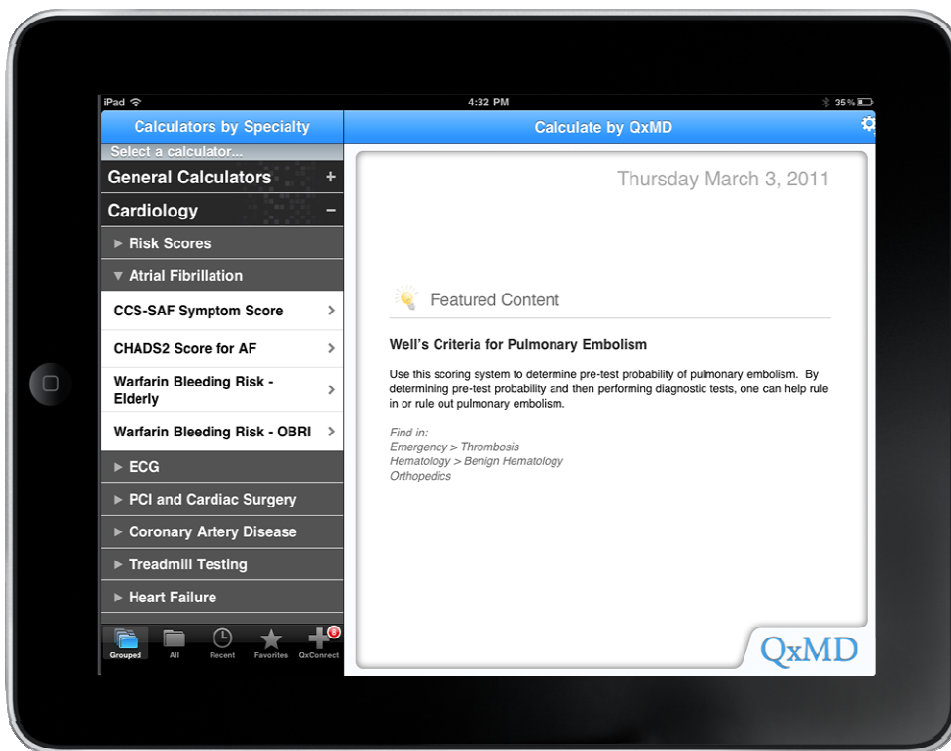
Obr. 9 Aplikácia Medscape pre tablety s OS Android [24]

Veľmi podobnou aplikáciou je MicroMedex, ktorá sa odlišuje od predchádzajúcej tým, že nie je potrebná registrácia a je využívaná prevažne lekárnikmi. Na obrázku (Obr. 10) je vidieť ponuku informácií po výbere konkrétneho lieku.[26]



Obr. 10 Aplikácia MicroMedex pre tablety s OS Android [27]

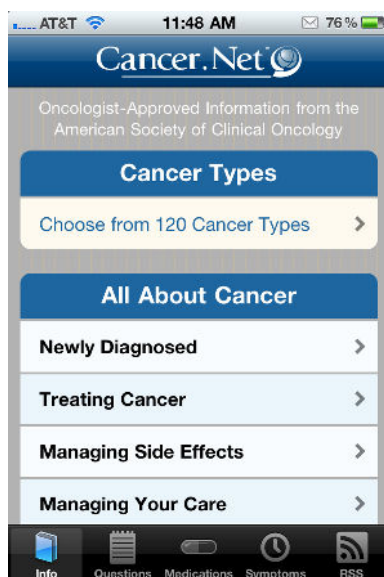
Ďalšou užitočnou aplikáciou, ktorú ocení každý lekár je špeciálna kalkulačka s názvom Calculate by QxMD (Obr. 11). Je veľmi rozšírenou a užitočnou. [28]



Obr. 11 Aplikácia Kalkulačka by QxMD pre tablety s OS Android [28]

Ľudia trpiaci rakovinou potrebujú neustály kontakt s lekárom. Tento kontakt vie zabezpečiť pomocou on-line aplikácie Cancer.Net (Obr. 12), ktorá umožňuje

každodenné záznamy. Na základe týchto záznamov, vie lekár všetky reakcie na lieky. Taktiež pacient si môže prečítať viac o rôznych typoch rakoviny, o nežiaducich účinkoch a ako im predchádzať. [29]



Obr. 12 Aplikácia Cancer.net pre OS Android [29]

Poslednou najvyužívanejšou aplikáciou pre iOS a potom neskôr aj pre Android je AHRQ EPSS (Obr. 13). Bola vyvinutá na základe amerického ministerstva zdravia a sociálnych vecí. Je to vlastne štúdia zozbieraná počas niekoľkých rokov. Na základe vyplneného formulára, v ktorom sa zadávajú všeobecné údaje ako pohlavie, vek, fajčiar a iné. Program vyhodnotí aké choroby môže človek očakávať. Môže si prečítať rôzne informácie ako týmto chorobám predísť. [30]

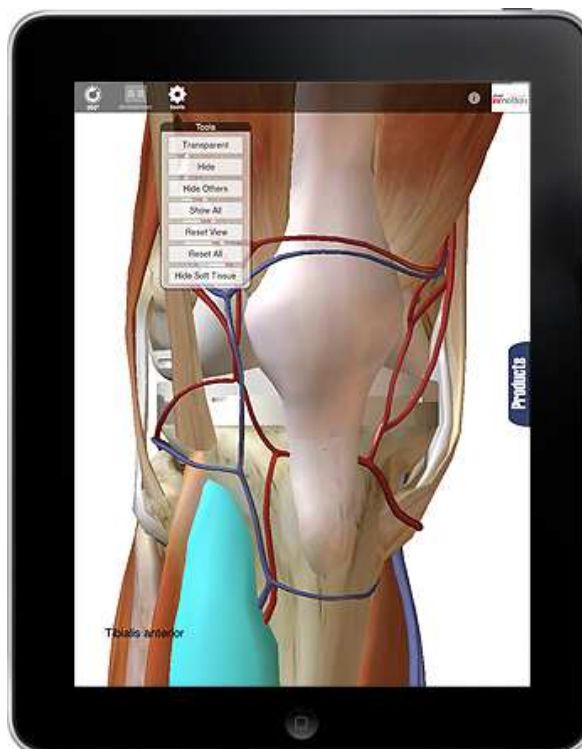
Obr. 13 Aplikácia AHRQ ePSS pre OS Android [30]

Tieto aplikácie patria medzi najpoužívanejšie, preto boli vyvinuté pre viac systémov. Avšak v Amerike je ich využitie oveľa väčšie a preto existuje mnoho ďalších aplikácií, ktoré sú zaujímavejšie a časom budú možno vyvinuté aj pre iné operačné systémy. Väčšinou ide o programy, ktoré sú zamerané na nejakú oblasť. Napríklad Draw MD series. Je rozdelená na viac menších častí, z ktorých každá sa venuje inej medicínskej časti ako kardiológia, ortopédia, urológia a iné. Vyvinula ju firma zaoberajúca sa výrobou medicínskych zariadení. Ide o rôzne náhrady kĺbov a pomôcky. Cieľom je predstaviť pacientom náhradu niektorých častí. A taktiež oboznámiť ich s rôznymi možnosťami a tak dať do popredia výroby firmy. V aplikácií si pacient, alebo lekár môže vybrať z rôznych častí tela a označiť napríklad časť tela, ktorej sa týka chirurgický zákrok (Obr. 14). Program mu ponúkne výber z rôznych produktov. [31]



Obr. 14 Aplikácia Draw MD Series predstavujúca medicínske produkty [31]

Medzi zaujímavé programy patrí aj InMotion 3D, ktorá je zameraná na ortopédiu. Pomocou nej sa môžu pacienti alebo aj študenti naučiť anatómiu častí tela. Obsahuje aj názorné ukážky pohybu. Na obrázku (Obr. 15) je možné vidieť svaly, kĺby, kosti a dôležité žily a tepny. [32]



Obr. 15 Aplikácia InMotion znázorňujúca stavbu rôznych častí tela [32]

Existuje veľa voľne dostupných medicínskych aplikácií pre študentov, doktorov, pacientov. Keďže pre operačný systém Android sa v tejto oblasti začali vyvíjať softvérové produkty omnoho neskôr ako pre iOS, aj preto je Android na nižšej úrovni. Firma Apple vytvorila stránku s doménou ioshealthsystems [33], kde je možné sa dočítať súčasné novinky, informácie o nových programoch a zariadeniach.

Medicínske aplikácie patria k skupine, ktorej vývoj musí byť neustále kontrolovaný. Medzi najdôležitejšie úlohy patrí poskytovanie kvalitných pomôcok pre študentov, dôležité informácie pre pacientov a lekárom poskytnúť pomôcku, ktorá im urýchli mnohé činnosti.

Ďalšiu skupinu produktov pre zdravotnícky priemysel tvorí softvér využívaný na medicínskych zariadeniach. Patria tu napríklad prístroje používané na onkológií, kardiológii, rádiológii a v ďalších. Vývoj pre túto skupinu je najnáročnejší, pretože mnohokrát chyba môže znamenať smrť človeka. Preto je celý životný cyklus vyvíjaný podľa mnohých štandardov, ktoré sa môžu časom meniť.

Medicínska oblasť neustále napreduje a preto sa radí medzi tie najväčšie. Existuje tu mnoho možností vyvíjať nové a vylepšovať staré produkty na základe nových poznatkov. Vzhľadom na pokrok mobilných zariadení a tabletov, ktorých výhoda je

hlavne v ich prenosnosti a ľahkosti sa začali aplikácie vytvorené na klasických počítačoch presúvať a vyvíjať pre mobilné zariadenia.

Výhodou mobilných zariadení je, že umožňujú si prečítať novinky a študovať nové veci pohodlne a rýchlo. Ďalšou výhodou môže byť aj poskytnutie rýchlej pomoci. V súčasnosti existujú aplikácie, ktoré presne popisujú postup podania rýchlej pomoci. Niektoré disponujú aj videami. K užitočným radíme aj programy, ktoré na základe toho kde sa zariadenie aktuálne nachádza, vie určiť najbližšie pohotovosti, nemocnice, polikliniky, odborných lekárov, lekárne a dokonca aj veterinárov. Mnoho aplikácií sa cez internet len aktualizuje, ale je možné k nim pristupovať aj bez neho. V súčasnosti je podiel smartphonov veľmi vysoký.

3 Návrh a popis oblasti pre inovačné využitie platformy Android v medicínskych systémoch

Android ako súčasne najrozšírenejšia platforma by mohol byť časom využívaný aj vo viacerých odborných inštitúciách. Výhodou nie je len jeho aktuálny nárast, ale aj veľmi jednoduché a intuitívne ovládanie. Taktiež dôležitým faktorom je rýchla komunikácia a vyhodnocovanie aplikácií, čím by sa dosiahla efektívnosť. Veľké plus sa pripisuje aj dotykovému ovládaniu a multitaskingu.

Softvérové produkty, ktoré sú určené pre nemocnice a všeobecne pre medicínu sú označované za rizikové. Pretože musia byť podrobne kontrolované, aby nedošlo k chybe. Ak by náhodou došlo k zlyhaniu, musí byť situácia ošetrená tak, aby sa neohrozil stav pacienta a doktora.

V medicínskych odvetviach by sa mohol operačný systém Android zamerať na aplikácie pre konkrétne oddelenia a produkty zamerané na jednotlivé choroby.

Firmy, ktoré sa venujú výrobe výrobkov nevyhnutných pre niektorých pacientov, by mohli vyvinúť aplikácie, ktoré by umožnili pacientom bližšie spoznať ich chorobu a možnosti liečenia. Takýmto spôsobom by mohli predstaviť výhody ich produktov. Napríklad ľudia trpiaci cukrovkou sú nútení kontrolovať si výšku cukru v krvi pomocou špeciálneho pera. V súčasnosti existuje mnoho typov tohto vloženého systému. Inteligentné inzulínové perá už obsahujú aj technológiu Bluetooth. Pomocou tejto technológie by bolo možné uskutočniť komunikáciu tohto systému s ďalším, ktorý by výsledky zbieral a vyhodnocoval. Najlepšie by bolo keby sa informácie zbierali počas celého dňa do mobilného telefónu prípadne do tabletu. Ten by údaje zapisoval a mohol by pomocou nich urobiť nejaké grafy. Prostredníctvom zaznamenávania údajov by vedel doktor povedať, či sa stav zlepšuje alebo zhoršuje a či je potrebné upraviť dávkovanie alebo typ lieku.

Ďalšou zaujímavou inovačnou implementáciou by mohlo dôjsť v nemocniciach. Pacienti, ktorý ležia na lôžkach by nemuseli byť pripojení na rôzne prístroje pomocou káblov, ale stačili by im senzory, ktoré by dokázali sledovať životné funkcie. Náramky, ktoré vedia snímať tep človeka už existujú. Určite by bolo lepšie keby sa dali kontrolovať životné funkcie bezdrôtovo. Bolo by to pre pacientov príjemnejšie a pohodlnejšie.

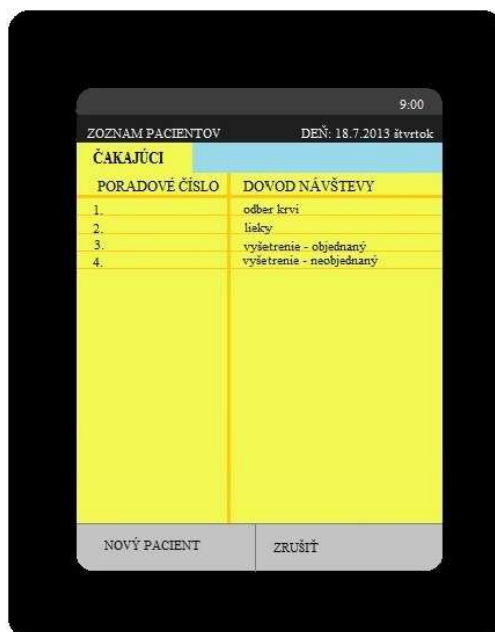
Zavedením vhodného elektronického systému, ktorý by riešil poradie pacientov, by sa mohlo predísť častým nedorozumeniam v čakárňach. V čakárňach u lekárov funguje, buď číselný poradový systém, alebo papierové zapisovanie. Takýto systém by sa dal zefektívniť práve tabletom, na ktorom by bola prístupná len aplikácia, ktorá by zlučovala obe terajšie systémy. Pacienti by zapísali do zoznamu svoje meno, priezvisko, rok narodenia a vybrali by si z ponuky dôvod kvôli čomu prišli k lekárovi. (Obr. 16)



The image shows a tablet displaying a mobile application interface for adding a new patient. The screen is titled "NOVÝ PACIENT" and has a status bar at the top showing "9:00" and "DEN: 18. 7. 2013 štvrtok". Below the title, there are input fields for "PRIEZVISKO", "MENO", and "DÁTUM NARODENIA". Below these fields, there are radio button options for "DOVOD NÁVŠTEVY": "ODBER KRVI", "TESTY", "VYŠETRENIE - OBJEDNANÝ", "VYŠETRENIE - NEOBJEDNANÝ", and "LIEKY". At the bottom of the screen, there are two buttons: "ZRUŠIŤ ZÁZNAM" and "SPÄT".

Obr. 16 Pridanie pacienta do zoznamu

Mnohokrát sa totiž stáva, že do zoznamu sa niektorí ľudia nezapíšu, alebo sa zapíšu aj tí čo idú len po lieky, alebo sa nezdržia dlho. Táto aplikácia by umožňovala sestričke samostatne zoradiť pacientov podľa priority. Ďalšou možnosťou by bolo keby sa zoradovanie robilo automaticky podľa prednastavenej priority v systéme. Po zapísaní sa do formulára, by sa pacientovi zobrazilo poradové číslo, ktoré si musí zapamätať. Potom by sa mu zobrazil aktuálny zoznam (Obr. 17).



Obr. 17 Aplikácia v čakárni

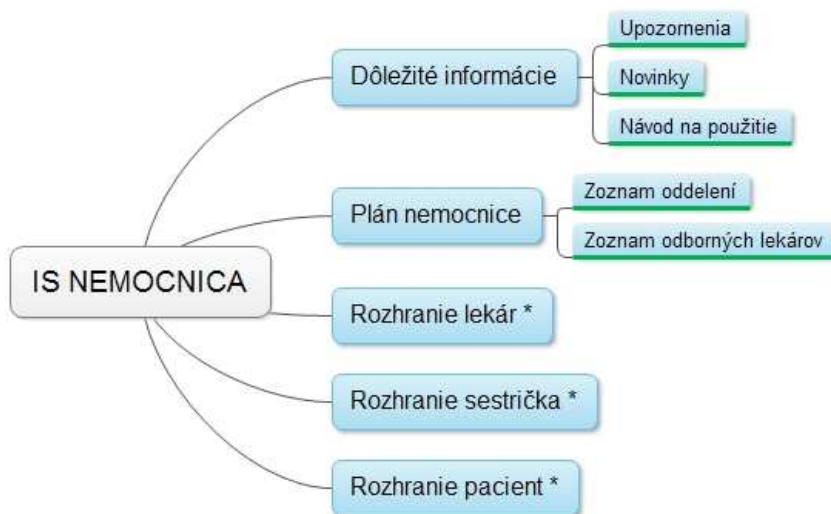
Keďže meno a dátum narodenia sú citlivé informácie, ktoré by nemali byť prístupné iným osobám, preto by sa v čakárni zobrazoval, len zoznam poradových čísel a ešte dôvod kvôli čomu pacienti prišli.

Sestričke v ambulancii by sa zobrazovali všetky mená s poradovými číslami, dátumom narodenia a dôvodom návštevy (Obr. 18). Úlohou sestričky by bolo ošetrovaných pacientov označiť v programe ako vybavených a tak by sa presunuli do inej tabuľky a sa zoznam čakajúcich ľudí by sa aktualizoval.



Obr. 18 Aplikácia v ambulancii

Užitočným softvérovým produktom by mohol byť systém, ktorý by uľahčil chod oddelení na poliklinikách alebo v nemocniciach. Išlo by o rozsiahly produkt, do ktorého by mali prístup rôzne skupiny ľudí s rozličnými právami. Jeho hlavným cieľom by bola jednoduchá a rýchla dostupnosť informácií. Zamestnanci by vedeli pracovať s rozpisom služieb, zoznamom pacientov a izieb. Taktiež by bolo možné zistiť obsadenosť odborných vyšetrení. Jednotlivé funkcie sú rozpisované pomocou myšlienkového mapy, ktorá kvôli svojej veľkosti musela byť rozdelená na viacero častí. Na prvom obrázku sú zobrazené základné možnosti systému (Obr. 19). V takomto systéme by nemali chýbať dôležité informácie, pre používateľov systému. Patria tu rôzne notifikácie o nových možnostiach, o upozorneniach na rôzne odstávky systému alebo upozorneniach, ktoré sa týkajú údržby nemocnice. Taktiež by mal byť dostupný návod na ďalšie použitie systému. Jednoduchý a výstižný popis funkcií, aby používatelia s ním vedeli pracovať rýchlo. Ďalšou možnosťou systému zavedeného v nemocnici by malo byť prezeranie areálu nemocnice a jednotlivých budov s označením. Po zobrazení budovy by sa zobrazili konkrétne oddelenia, prípadne ich špecializácia.



Obr. 19 Myšlienková mapa Informačného systému pre nemocnicu

Ďalšie možnosti systému by sa zobrazovali podľa typu prihláseného používateľa. Rozhrania prihlásených používateľov sú rozdelené do samostatných častí:

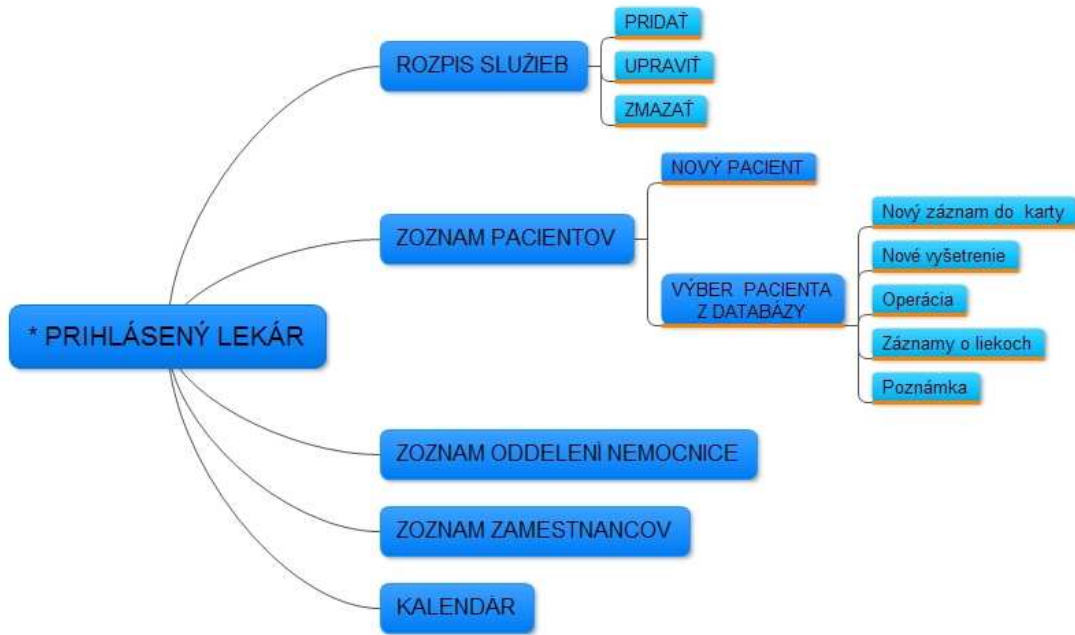
- Prihlásená sestrička (Obr. 20) – by mala možnosť vidieť rozpis služieb zdravotného personálu v nemocnici, ktorý by vedela upravovať. Taktiež by mala prístup k zoznamu pacientov. V tomto zozname by si mohla pozrieť, kde pacient leží, alebo na aké vyšetrenie, alebo operáciu sa chystá. Tiež by mala k dispozícii zoznam zamestnancov nemocnice a oddelení. Mohla by mať

implementovaný aj vlastný kalendár, kde by sa mohli zobrazovať jej denné služby, a dôležité termíny, ktoré by si vedela sama upravovať.



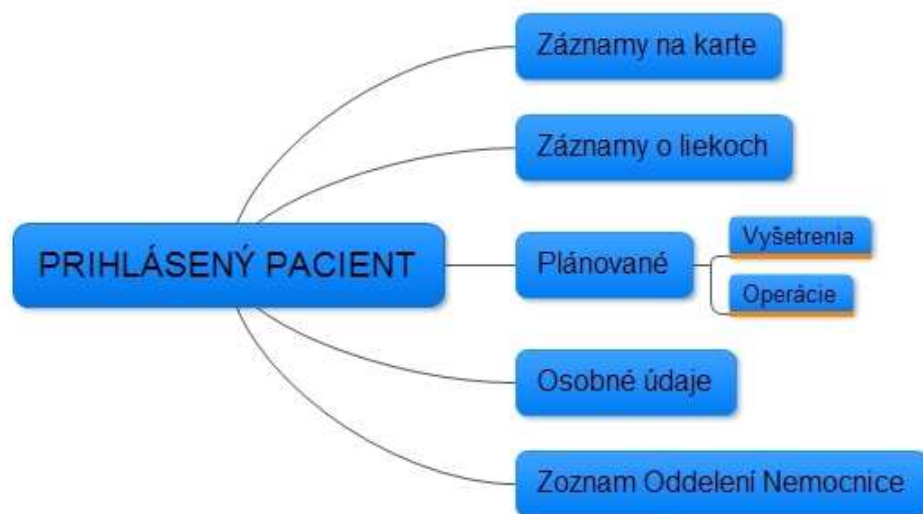
Obr. 20 Pokračovanie myšl. mapy – rozhranie sestrička

- Prihlásený lekár (Obr. 21) – má podobné funkcie ako sestrička. Mal by možnosť prezerat' si rozpis služieb zdravotného personálu. Mohol by ho upravovať. Všetky zmeny by sa vyznačovali tak, aby sa zobrazilo aj meno osoby, ktorá služby upravovala. Osobám, ktorých sa zmena služby týkala, napríklad – zrušenie služby, presun na inú službu, doplnenie do služby, tak by sa posielala notifikácia. Každý lekár by mal k dispozícii zoznam pacientov. Vedel by pridávať nového pacienta do databázy a pridávať mu nové chorobopisy, záznamy o liekoch, ale aj plánovať operácie, alebo odborné vyšetrenia. Taktiež by mal k dispozícii zoznam zamestnancov a oddelení nemocnice. Lekárov kalendár by mohol slúžiť podobne ako diár. Vedel by ho upozorňovať na stretnutia. Tiež by si lekár mohol zaznamenať dôležité termíny operácií a iných udalostí.



Obr. 21 Pokračovanie myšl. mapy – rozhranie lekár

- Prihlásený pacient (Obr. 22) – pacient by mal najmenej možností z rôznych používateľov. Vedel by si prezerať osobné údaje v systéme, jednotlivé záznamy na karte, záznamy o liekoch, ale aj plánované odborné vyšetrenia a operácie. Taktiež by mal k dispozícii zoznam oddelení nemocnice.



Obr. 22 Pokračovanie myšl. mapy - rozhranie pacient

4 Návrh a implementácia mobilnej aplikácie pre medicínske účely

Zaujímavou predstavou o inovačnom systéme v medicínskej oblasti je aj zdravotná karta. V niektorých krajinách majú zavedené karty, ktoré obsahujú čip. Nachádzajú sa v nej osobné a zdravotné údaje o pacientovi.

Mnohí lekári využívajú ešte aj papierové formy zdravotných kariet o pacientovi. Niektorí zapisujú údaje aj do systému a následne ich vytlačia a založia do karty. Do týchto systémov nemá prístup pacient a nie je možná komunikácia s inými lekármi.

Moja aplikácia je návrhom ako by taká elektronická zdravotná karta mohla vyzerat' a fungovat'. Keďže platforma Android sa snaží svoje funkcie rozširovat' a zlepšovāt tak je možné, že onedlho bude možné implementovat' aj všetky navrhnuté funkcie.

4.1 Analýza požiadaviek

Analýza požiadaviek je znázornená v tabuľke (Tab. 4), ktorá zobrazuje zoznam požiadaviek usporiadaný podľa logickej postupnosti. Ku každej požiadavke je uvedená jej priorita. Priorita číslo 1 znamená, že požiadavka musí byť najskôr implementovaná, teda je nevyhnutná pre používateľa v aplikácii. Priorita číslo 2 znamená, že po implementovaní požiadaviek s prioritou 1 sa môžu implementovat' požiadavky s prioritou 2. Požiadavky s prioritou číslo 3 nie sú nevyhnutné pre celkový beh systému a môžu byť zapracované úplne nakoniec.

Vývoj produktu by mal vždy začínat' analýzou požiadaviek so zákazníkom, pre ktorého je produkt vyvíjaný. Počas samotného vývoja by sa mali vývojári pridržať požiadaviek a neskôr aj testovat', či všetky požiadavky boli splnené. Prípadne počas vývoja doplnit' ďalšie aj s príslušnou prioritou.

PORADIE	NÁZOV POŽIADAVKY	PRIORITA
1	registrácia lekára	1
1.2	prihlásenie lekára	1
2	registrácia pacienta	1
2.1	prihlásenie pacienta	1
3	rozdielne používateľské práva	1
4	vytváranie zoznamu pacientov	1
4.1	pridanie nového pacienta	1
4.2	pridanie osobných údajov pre pacienta	2

4.3	prezeranie osobných údajov pacienta	3
4.4	pridanie základných zdravotných údajov pre pacienta	2
4.5	prezeranie základných zdravotných údajov pre pacienta	3
5	vytvorenie zoznamu chorobopisov	2
5.1	pridanie nového záznamu do chorobopisu	2
5.2	zoradenie chorobopisov podľa dátumu	2
6	úprava osobných údajov pacienta	2
6.1	úprava základných zdravotných údajov pacienta	2
7	zmazanie pacienta	3
8	zadanie údajov o lekárovi a jeho činnosti	2
8.1	úprava údajov o lekárovi a jeho činnosti	2
9	prezeranie kategorizácie liekov	3
10	prezeranie všetkých liekov	3

Tab. 4 Zoznam všetkých požiadaviek

Klasifikácia požiadaviek podľa typu používateľa je znázornená v nasledujúcich dvoch tabuľkách - pre lekára (Tab. 5) a pre pacienta (Tab. 6).

PORADIE	NÁZOV POŽIADAVKY	PRIORITA
1	registrácia lekára	1
1.2	prihlásenie lekára	1
2	registrácia pacienta	1
3	vytváranie zoznamu pacientov	1
3.1	pridanie nového pacienta	1
3.2	pridanie osobných údajov pre pacienta	2
3.3	úprava osobných údajov pacienta	2
3.4	pridanie základných zdravotných údajov pre pacienta	2
3.5	úprava základných zdravotných údajov pacienta	2
3.6	pridanie nového záznamu do chorobopisu	2
3.7	zoradenie chorobopisov podľa dátumu	2
4	zmazanie pacienta	3
5	zadanie údajov o lekárovi a jeho činnosti	2
5.1	úprava údajov o lekárovi a jeho činnosti	2
6	prezeranie kategorizácie liekov	3
7	prezeranie všetkých liekov	3

Tab. 5 Zoznam požiadaviek lekára

PORADIE	NÁZOV POŽIADAVKY	PRIORITA
1	prihlásenie pacienta	1
2	prezeranie údajov o lekárovi a jeho činnosti	2
3	prezeranie vlastných osobných údajov	1
4	prezeranie vlastných zdravotných údajov	1
5	prezeranie zoznamu chorobopisov	1
6	prezeranie jednotlivých záznamov chorobopisov	1

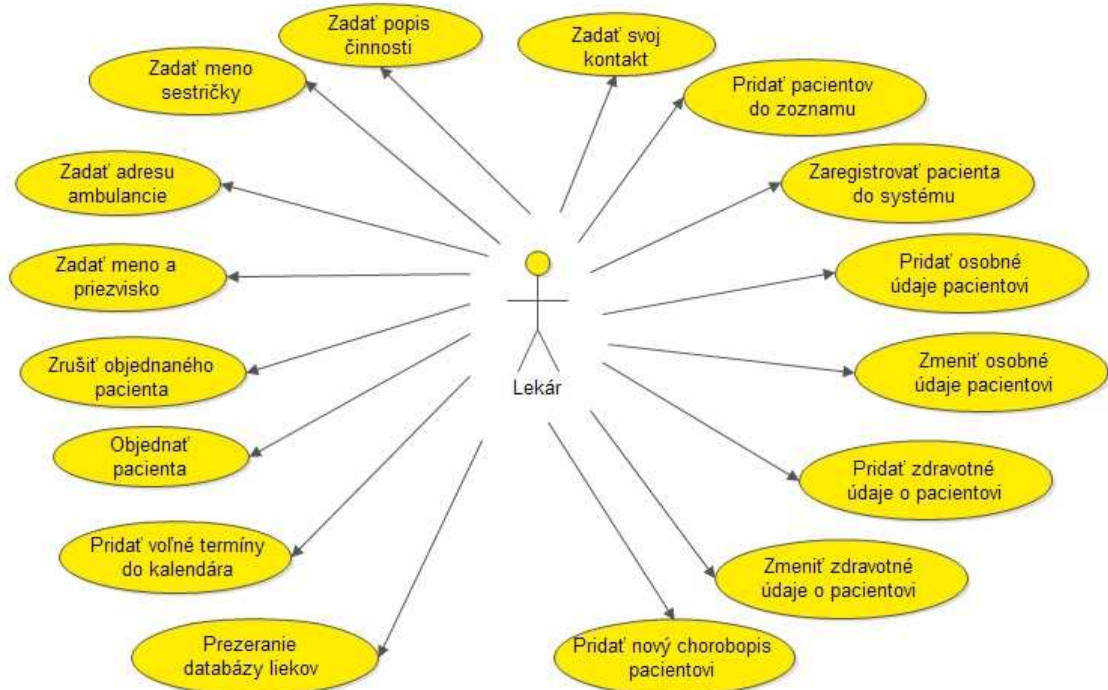
Tab. 6 Zoznam požiadaviek pacienta

4.2 Návrh aplikácie

Hlavnou myšlienkou mojej práce je vytvoriť elektronickú zdravotnú kartu, ktorá bude aplikovaná na tabletoch. Výhodou tabletov nie je len ich váha a veľkosť, ale aj neustále vylepšovaný výkon, ktorý sa už vyrovná aj notebookom.

Aplikácia je vytváraná pre Android s verziou 4.0 a vyššie. Rozhodla som sa pre túto verziu preto, lebo ako jediná je aplikovateľná aj pre mobilné telefóny a tablety. Táto verzia umožňuje ovládať systém len za pomoci dotykového displeja. Hlavné menu je troška upravené oproti starším verziám a to tak, že je rozdelené na aplikácie a widgety. Widget je označenie pre miniaplikáciu - prvok, ktorý je prispôbený tak, aby ho bolo možné ovládať čo najjednoduchšie. Hlavné menu je takto prehľadnejšie a používatelia si môžu ikony jednotlivých aplikácií presúvať a rozdeľovať ako sa im páči. Táto verzia má už aj odomykanie zariadenia pomocou rozpoznania tváre prostredníctvom kamery. Internetový prehliadač už dokáže synchronizovať záložky s prehliadačom Google Chrome, ktorý beží na osobnom počítači. Taktiež už disponuje funkciou, ktorá uloží internetové stránky tak, aby boli čitateľné v režime bez internetového pripojenia. Táto verzia je známa pod názvom Ice Cream Sandwich.

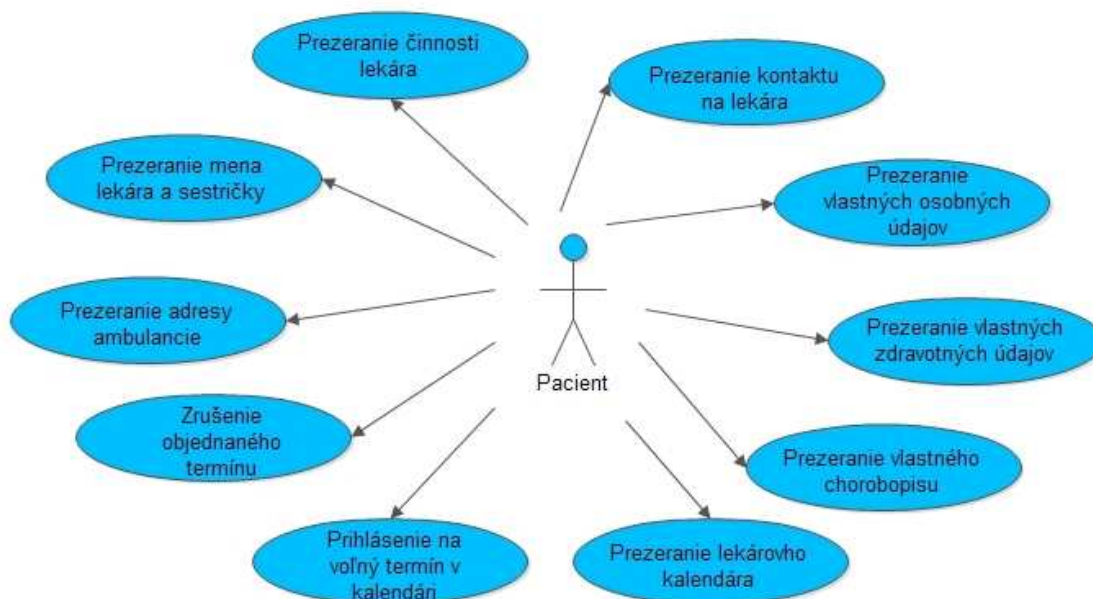
Aplikácia je určená pre dve skupiny ľudí s rôznymi právami. Prvou skupinou bude lekár (Obr. 23), ktorý sa po spustení aplikácie zaregistruje. Tlačidlo na registráciu sprístupňuje administrátor aplikácie, z dôvodu, aby sa nemohol do aplikácie registrovať hocikto za lekára. Prihlasovacie údaje sa uložia do lokálnej databázy SQLite na danom zariadení, aby ich pri ďalšom spustení aplikácie používateľ nemusel zadávať.



Obr. 23 Model používateľa lekár

Ako vidno na obrázku (Obr. 23) lekár si vytvára databázu pacientov. Pri zadávaní jednotlivých pacientov im najprv musí prideliť prihlasovacie meno a heslo. Až potom je možné zadať osobné a zdravotné údaje. Len lekár má možnosť meniť, pridávať a mazať jednotlivé údaje. Taktiež má k dispozícii aj databázu liekov.

Prihlásený pacient má možnosť si všetky údaje len prezerať (Obr. 24). Po prihlásení do systému sa jeho údaje tiež uložia do SQLite databázy na zariadení. V tejto aplikácii si vie prezerať aj záznamy jednotlivých chorobopisov zoradených podľa dátumu. Pri každom zázname je aj prehľad predpísaných liekov. Výhodou pre pacienta je tiež to, že môže sledovať aktuality o lekárovi. Napríklad či mu pribudli nejaké nové činnosti, alebo sa presťahoval na inú adresu, alebo zmenil telefónne číslo.



Obr. 24 Model používateľa pacient

Pridanie nového pacienta prebieha tak, že lekár, ktorý pridáva pacienta do databázy musí najprv zadať prihlasovacie meno a heslo pre daného pacienta. Potom zadá osobné údaje a zdravotné údaje a môže hneď zadať záznam do chorobopisov.

4.3 Platformy použité pri vývoji aplikácie

Na vývoj aplikácií pre operačný systém Android je uprednostnené vývojové prostredie Eclipse IDE, ktoré je považované za oficiálny vývojový nástroj. Toto prostredie patrí medzi voľne dostupné programy. Výhodou je jednoduché a intuitívne ovládanie, ale taktiež aj to, že samotný program nezaberá veľa miesta. Programovanie v špecifických jazykoch, alebo pre individuálne systémy ako je aj Android, je prispôbené doplnkami, ktoré si musí užívateľ stiahnuť. Tie sú tiež voľne stiahnuteľné. Konkrétne pre vývoj aplikácií na platformu Android, na oficiálnej stránke pre vývojárov je možné si stiahnuť súbor s názvom – Android nástroj pre vývojárov, ktorý obsahuje emulátor s manažérom a Eclipse IDE, v ktorom sú už pridané doplnky na vývoj. Stačí, len nastaviť manažér emulátora, v ktorom je potrebné si stiahnuť súbory slúžiace pre vývoj na konkrétnu verziu Androidu. V tomto vývojovom prostredí sa využíva programovací jazyk Java na vývoj aplikačnej logiky a na grafické používateľské rozhranie je použitý programovací jazyk XML. Vytváranie grafického rozhrania prebieha pomocou špeciálneho editora, ktorý umožňuje výber z predvolených najpoužívanejších typov a následne ich úpravy cez vlastnosti. Je to napríklad vkladanie

tlačidiel, textu, zoznamu a ďalších. Tak aj človek, ktorý neovláda XML si vie vytvoriť obrazovku so základnými prvkami. [37]

Návrh databázovej časti som robila v programe Microsoft Office Visio 2007, ktorý je dostupný na MSDNA pre študentov. Tento program slúži na vyvíjanie rôznych návrhov diagramov, procesov, štruktúr. Práca s týmto programom je veľmi rýchla a jednoduchá, pretože je zostavený ako väčšina produktov od firmy Microsoft konkrétne v skupine programov Office. Výhodou tohto programu pri návrhu databázových diagramov sú predvolené možnosti, pomocou ktorých je program schopný vygenerovať kód v jazyku SQL.

Na interpretáciu vzdialeného prístupu s databázou som použila program XAMPP. Je to program, ktorý je vyvíjaný skupinou Apache Friends. Ide o program, ktorý obsahuje Apache, MySQL, PHP a Perl. Je voľne dostupný a nie je potrebná žiadna inštalácia. Stačí stiahnuť verziu pre daný systém, rozbaliť a spustiť. Apache je program, ktorý predstavuje webový server. Apache je taktiež voľne dostupný. Apache spolu s PHP a MySQL sú veľmi často využívané k vývoju dynamických internetových stránok. MySQL je databázový server. Údaje mojej aplikácie sa ukladajú do MySQL databázy pomocou nástroja phpMyAdmin. To je nástroj v internetovom prehliadači, ktorý je prispôbený na vytváranie databázových skriptov. Komunikácia databázy a mojej aplikácie prebieha pomocou skriptov v PHP programovacím jazyku. [38]

5 Možnosti integrácie databázovej časti

Existuje viacero možností ako prepojiť databázovú časť s aplikačnou. Dáta potrebné pre aplikáciu môžu byť uložené napríklad v jednoduchých súboroch, v lokálnej databáze SQLite, alebo na vzdialenom serveri, ku ktorému sa pristupuje prostredníctvom internetu.

SQLite – je lokálna databáza v systéme Android, ktorá je zabudovaná vo vykonávacom prostredí. A práve to umožňuje každej aplikácii vytvoriť vlastnú databázu. Práca s ňou je veľmi jednoduchá pre ľudí, ktorí už majú nejaké skúsenosti s inými rozhraniami SQL. Patrí medzi najrozšírenejšie, pretože obsahuje čisté SQL rozhranie, zaberá veľmi málo miesta (275 kB) a je pomerne rýchla.

SQLite používa pre dotazy, manipuláciu s dátami a definíciu dát dialekt jazyka SQL. Je efektívna pri práci s pamäťou, pretože ju môže vykonávacie prostredie systému obsahovať celú a nie len jednu časť, aby sa znížil jej celkový objem. Veľký rozdiel medzi SQLite a inými databázami je v typovaní dát. Do akéhokoľvek stĺpca je možné uložiť dáta ľubovoľnej hodnoty. SQLite nepodporuje niekoľko štandardných vlastností SQL:

- obmedzenie typu použitím cudzieho kľúča (angl. FOREIGN KEY),
- vnorená transakcia,
- čiastočné vonkajšie spojenie typu sprava (angl. RIGHT OUTER JOIN)
- úplné vonkajšie spojenie (angl. FULL OUTER JOIN),
- niektoré príkazy vykonávajúce modifikáciu tabuľky (angl. ALTER TABLE)

Pre aplikácie, v ktorých majú prístup viacerí používatelia je lepšie mať dáta uložené na vzdialenom serveri. Po spustení aplikácie sa načítajú aktuálne údaje z databázy. Komunikácia s webovými službami prebieha pomocou protokolu SOAP (angl. Simple Object Access Protocol). Je to protokol, ktorý zabezpečuje výmenu správ založených na XML, prostredníctvom siete. SOAP je nástupcom XML-RPC protokolu. XML-RPC uchováva pravidlá, ktoré definujú ako sa používajú štandardné technológie pre potreby RPC (vzdialené volanie procedúr). V systéme Android sa nenachádzajú API protokolu SOAP a ani protokolu XML-RPC. Táto komunikácia je v Androide vyriešená tak, že súčasťou systému Android je knižnica Apache HttpComponents, nad ktorou je

možné umiestniť vrstvu protokolov SOAP alebo XML-RPC, alebo je možné používať prístup k webovým službám založeným na architektúre REST (angl. Representational State Transfer). Rest je architektúra, ktorá umožňuje prístupovať k dátam na určitom mieste pomocou štandardných metód http. Definuje ako majú byť špecifikované a adresované. Využíva štandardy :

- prenosný protokol (HTTP)
- zdrojový protokol (URL)
- zdrojová reprezentácia (XML/HTML/GIF/JPEG/...)
- MIME typy (text/xml, text/html, image/gif, image/jpeg,...)

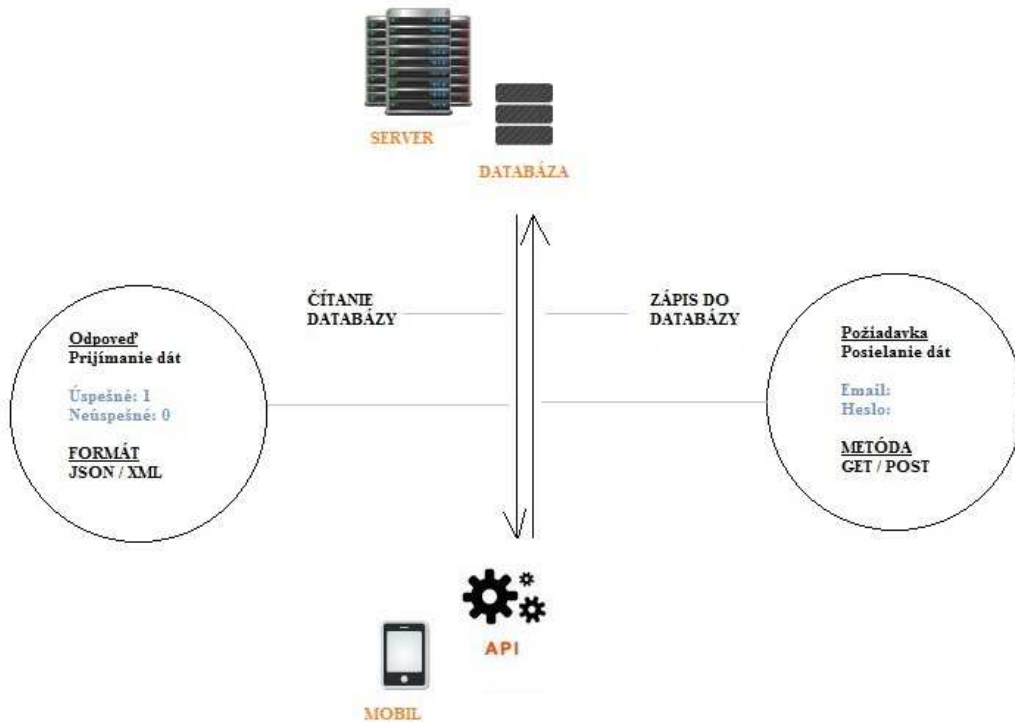
Prístup k zdrojom je jednotný a jednoduchý. REST umožňuje využitie CRUD operácií pomocou štandardných HTTP dotazov. REST definuje 4 základné metódy, ktoré sú v tabuľke (viď Tab. 7). [35]

REST	CRUD
POST	Create
GET	Read
PUT	Update or Create
DELETE	Delete

Tab. 7 Metódy v architektúre REST

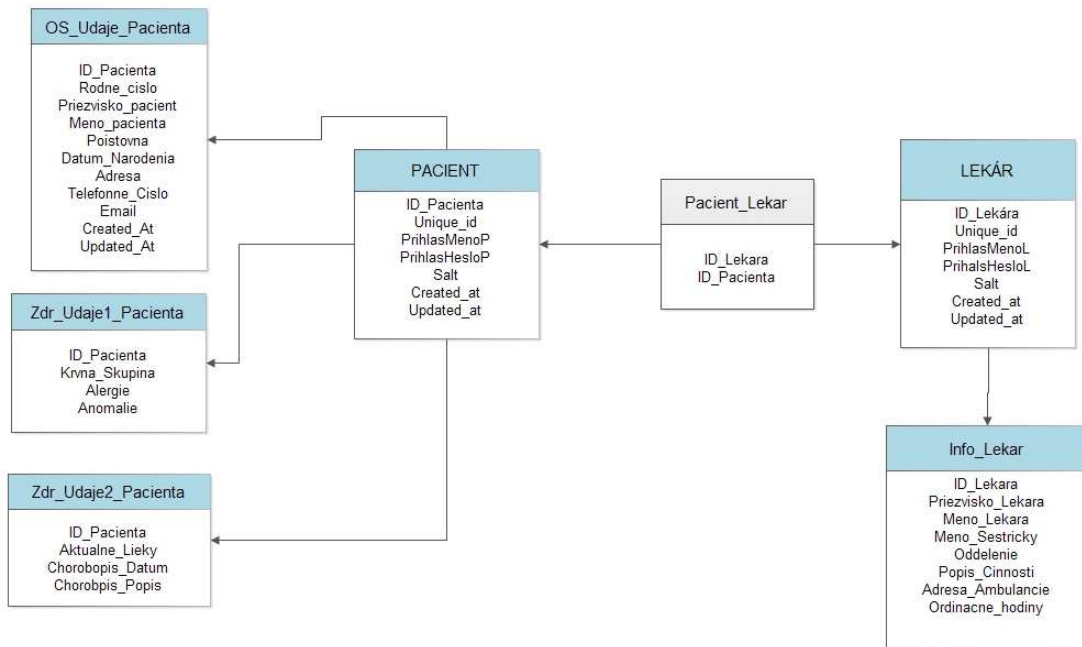
Moja aplikácia bude interpretovať pripojenie na vzdialený server. Využívam v nej architektúru REST a knižnicu Apache HttpComponents. Najprv je potrebné si vytvoriť objekt typu httpClient. Vzhľadom na to, že ide o rozhranie, bude sa musieť vytvoriť inštancia implementácie tohto rozhrania, napríklad trieda DefaultHttpClient. Do inštancie triedy HttpRequest sa zabalia požiadavky s rôznymi implementáciami pre jednotlivé http príkazy (napr. get). Následne je potrebné vytvoriť inštanciu triedy httpRequest s danou implementáciou. Do nej sa vloží cieľová URL adresa a ďalšie konfiguračné údaje. Potom sa metóda predá klientovi, aby sa http požiadavka vykonala prostredníctvom metódy execute(). Vrátí sa objekt typu HttpResponse s kódom odpovede. [36]

Na obrázku (Obr. 25) je znázornená databázová komunikácia mobilného zariadenia so serverom.

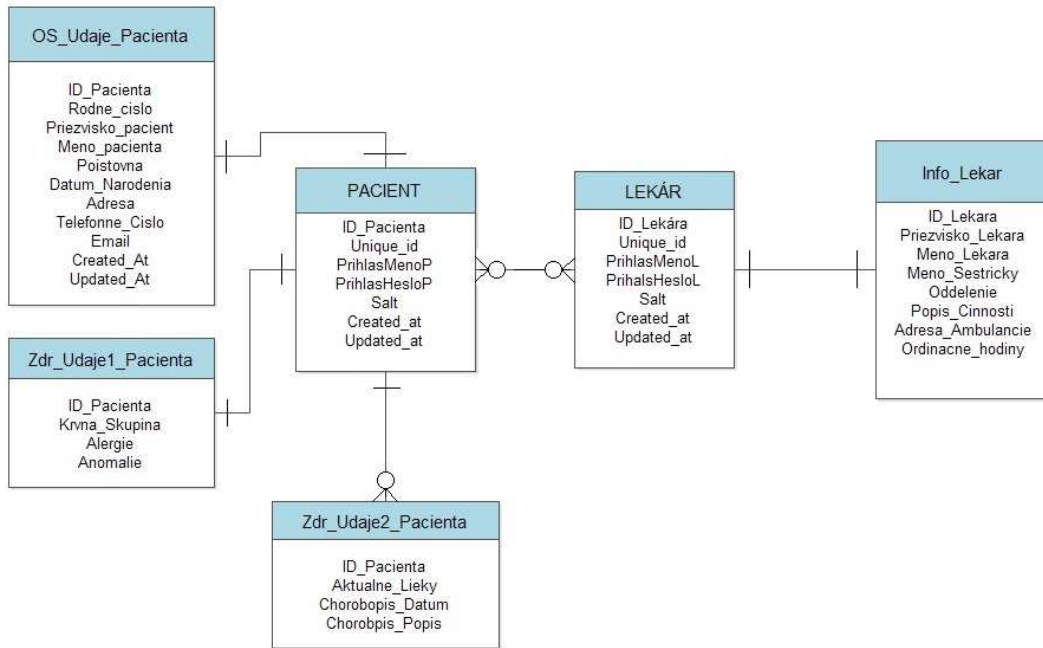


Obr. 25 Spôsob posielania a prijímania dát

Pred samotnou implementáciou databázy je potrebné navrhnuť logickú (Obr. 26) a fyzickú (Obr. 27) schému databázy. Databázový skript je súčasťou CD.



Obr. 26 Logická schéma



Obr. 27 Fyzická schéma

6 Záver

Úlohou mojej diplomovej práce bolo naštudovať poznatky o vložených systémoch, operačnom systéme Android a medicínskych systémoch. Taktiež som popísala súčasný stav tejto mobilnej platformy pre medicínsku oblasť a následne som navrhla inovačné oblasti, kde by mohla byť táto platforma užitočná. Výsledkom spracovania je aplikácia, ktorou som sa snažila ukázať možnosť vytvorenia elektronickej zdravotnej karty aj na inom operačnom systéme, ktorý sa využíva na iných zariadeniach ako sú osobné počítače. Ide o program určený hlavne pre všeobecných lekárov, ktorí svojim pacientom umožnia prístup do systému. Pacienti po nainštalovaní tohto programu do mobilu a prihlásení sa, môžu vidieť svoje osobné a zdravotné údaje. Nemôžu nič meniť.

Túto aplikáciu som sa rozhodla robiť pre operačný systém Android, pretože jeho rozšírenosť veľmi rýchlo napreduje. Myslím si, že je len otázkou času, keď sa rozšíri aj v medicínskej oblasti. V zahraničí využívajú bežne v tejto oblasti operačný systém iOS, ktorý je tiež určený pre mobilné zariadenia. Mnoho aplikácií, ktoré sa uplatnili na systéme iOS museli vyvinúť aj pre Android. Avšak v súčasnosti existuje cez 1000 medicínskych aplikácií, no sú v angličtine. V starších verziách Androidu bol problém s vývojom aplikácií pre príslušné zariadenia, pretože existuje veľa zariadení s týmto systémom, ale majú rozličné rozlíšenie obrazovky a rozličnú verziu systému. Firma Google vytvorila verziu 4.0 s názvom Ice Cream Sandwich, ktorá umožňuje vyvíjať aplikácie, ktoré sú aplikovateľné aj na rôzne typy mobilov a tabletov. Ale zariadenia musia mať túto verziu, alebo niektorú z novších.

V tejto aplikácii bolo potrebné urobiť aj nejakú formu zabezpečenia, len oprávnených osôb. Zaregistrovať sa do systému môžu len lekári, ktorí po pridávaní pacientov do zoznamu im vedia prideliť prihlasovacie údaje. Prihlasovanie je implementované za pomoci funkcie hash, ktorá slúži na šifrovanie mena a hesla. Je to kvôli tomu, že aj keď sa dostane do systému nejaká neoprávnená osoba tak nevie zistiť konkrétne prihlasovacie údaje. Využitie takéhoto systému by sprehládnilo a zjednodušilo prácu doktorov.

Tento program je možné rozšíriť o ďalšie funkcie ako vytvorenie dynamického kalendára, do ktorého by sa vedeli pacienti prihlasovať a videli by napríklad aj voľné termíny lekára. Lekárovi by mohla chodiť emailová notifikácia. Taktiež by sa dala viac prepracovať databáza liekov, ktorá by mohla obsahovať aj externé linky na internetové

odkazy s recenziami na daný liek. Existuje množstvo rozdelení liekov podľa typu chorôb, podľa špecializácie lekára, podľa dostupnosti a ďalšie. Takýto systém by sa mohol prepojiť aj s programom, ktorý som navrhla v kapitole - 3. Aktuálne zoznamy v čakárni by mohli byť uvedené na stránke určenej špeciálne pre nejakú polikliniku, v ktorej sa nachádza viaceró lekárov. Pacienti by si vedeli pozrieť cez internet aktuálny stav ľudí v čakárni, ktorý by sa pravidelne aktualizoval.

Programy v tejto oblasti majú množstvo výhod. Dostupnosť niektorých informácií na internete je pre pacientov lákavá, ale takéto zaujímavé systémy sa môžu stať lákavé aj pre páchanie počítačovej kriminality.

Informácie o rôznych medicínskych produktoch sú zaujímavé pre mnohých obyčajných ľudí. Ľudia s rôznymi problémami by si radi prečítali možnosti pri ich chorobách a možnosti rôznych operácií, no väčšina takýchto informácií je dostupných v aplikáciách, ktoré sú v angličtine. Napríklad možnosti pri výmene kĺbov a výber z rôznych náhrad by sa určite uplatnil.

Študenti zdravotných stredných a vysokých škôl sa môžu lepšie pripraviť na svoje povolanie aj vďaka rôznym aplikáciám. Existuje množstvo programov, ktoré obsahujú zahraničné odborné časopisy a aktuálne novinky v konkrétnych odboroch zdravotníctva. Aj rôzne interaktívne programy, ktoré umožňujú prejsť časťami ľudského tela s popismi a videami sú veľmi pekne spracované a zábavné.

Možnosti vývoja softvérových produktov pre oblasť medicíny je veľmi široká. Uplatnenie nemusí byť až také náročné, ak sú produkty spracované kvalitne a podávajú vecné a presné informácie. Dúfam, že postupne pribudne viac zaujímavých aplikácií v slovenskom jazyku, ktoré budú inšpirované aj zahraničnými programami, ktoré sa už v praxi uplatnili.

Zoznam použitej literatúry

- [1] Embedded systems – Wikipedia, the free encyclopedia [online]. [cit.2012-3-4] Dostupné na internete: < http://en.wikipedia.org/wiki/Embedded_system >
- [2] Vivek SHAH: June 2010 – [online]. [cit.2012-3-4] Dostupné na internete: <http://reply2viveksshah.blogspot.sk/2010_06_01_archive.html>
- [3] Android (operating system) - [online]. [cit.2012-3-4] Dostupné na internete: <http://en.wikipedia.org/wiki/Android_%28operating_system%29>
- [4] Android - Android Wiki [online]. [cit.2012-3-4] Dostupné na internete: <<http://wiki.androidforum.cz/index.php/Android>>
- [5] ITU: Committed to connecting the world [online]. [cit.2012-5-20] Dostupné na internete: <<http://www.itu.int/en/Pages/default.aspx>>
- [6] International Electrotechnical Commission [online]. [cit.2012-5-20] Dostupné na internete: <<http://www.iec.ch/>>
- [7] International Organization for Standardization [online]. [cit.2012-5-20] Dostupné na internete: < <http://www.iso.org/iso/home.htm> >
- [8] SUTN – Úloha normalizácie [online]. [cit.2012-5-20] Dostupné na internete: <<http://www.sutn.sk/default.aspx?page=81f79981-12c5-491e-87e3-491c73710141>>
- [9] U S Food and Drug Administration Home Page [online]. [cit.2012-5-20] Dostupné na internete: <<http://www.fda.gov/default.htm>>
- [10] FDA.COM Information Portal encyclopedia [online]. [cit.2012-3-4] Dostupné na internete: <<http://www.fda.com/>>
- [11] V-model – Software Testing Mentor [online]. [cit. 2013-04-10] Dostupné na internete: <<http://swtmentorsite.appspot.com/sdlc/vmodel.php>>
- [12] Development Model: V Model For Testing ~ “Testing A Software“ world [online]. [cit.2013-04-10] Dostupné na internete: <<http://testing-a-software.blogspot.sk/2011/11/development-model-v-model.html>>
- [13] Chapter 2 – Software Reliability> A Federal Highway Administration Preliminary Handbook, September 2004 – FHWA-RD-04-080 [online]. [cit.2013-04-10] Dostupné na internete: <<http://www.fhwa.dot.gov/publications/research/safety/04080/02.cfm>>
- [14] Dashboards | Android developer [online]. [cit.2012-4-7] Dostupné na internete: <<http://developer.android.com/about/dashboards/index.html>>
- [15] Mobile OS | worldwide market share [online]. [cit.2012-4-17] Dostupné na internete: <http://stats.areppim.com/stats/stats_mobiosxsnapshot.htm>
- [16] Point of Care Medical Applications | Epocrates [online]. [cit.2012-4-7] Dostupné na internete: <<http://www.epocrates.com/>>

-
- [17] Epocrates Rx Beta for the Android – Android App Review encyclopedia [online]. [cit.2012-4-7] Dostupné na internete: <<http://www.imedicalapps.com/2010/02/epocrates-android-medical-app/>>
- [18] PubMed Mobile – aplikácie pre Android v aplikácii Google Play [online]. [cit.2012-4-7] Dostupné na internete: <<https://play.google.com/store/apps/details?id=com.bim.pubmed&hl=sk>>
- [19] Home – PubMed – NCBI [online]. [cit.2012-4-7] Dostupné na internete: <<http://www.ncbi.nlm.nih.gov/pubmed>>
- [20] The Top Free Android Medical Apps [online]. [cit.2012-4-15] Dostupné na internete: <<http://www.imedicalapps.com/2010/03/free-android-medical-apps/5/>>
- [21] Mobile medical resources for iOS, Android & Blackberry | Skyscape [online]. [cit.2012-4-15] Dostupné na internete: <<http://www.skyscape.com/index/home.aspx>>
- [22] List of IEC 60601 Standards encyclopedia [online]. [cit.2012-6-10] Dostupné na internete: <http://www.601help.com/Other_601_Standards/other_601_standards.html>
- [23] CPR Choking Android [online]. [cit.2012-4-15] Dostupné na internete: <http://www.androidzoom.com/android_applications/health_and_fitness/cprchoking_bawn.html>
- [24] WebMed Launches Medscape for iPad and Android [online]. [cit.2012-4-15] Dostupné na internete: <<http://appsout.pcmag.com/apple-ios-iphone-ipad-ipod/268899-webmd-launches-medscape-for-ipad-and-android>>
- [25] Medscape: Medical News [online]. [cit.2012-04-15] Dostupné na internete: <<http://reference.medscape.com/>>
- [26] MicroMedex Mobile [online]. [cit.2012-04-15] Dostupné na internete: <<http://healthcare.thomsonreuters.com/micromedexMobile/>>
- [27] MicroMedex | PharmApps [online]. [cit.2012-4-15] Dostupné na internete: <<http://www.pharmapps.com/micromedex/>>
- [28] Calculate by QxMD | QxMD Medical Apps [online]. [cit.2012-4-15] Dostupné na internete: <<http://www.qxmd.com/apps/calculate-by-qxmd>>

-
- [29] Cancer.Net Mobile – Aplikácie pre Android v aplikácií Google Play [online]. [cit.2012-4-15] Dostupné na internete: <<https://play.google.com/store/apps/details?id=com.fueled.cancernet&hl=sk>>
- [30] US Department of Health and Human Services mobile app for health care providers emphasizes prevention [online]. [cit.2012-4-15] Dostupné na internete: <<http://www.imedicalapps.com/2011/05/practice-evidencebased-preventive-care-department-health-human-services-ahrq-epss-app-app-review/>>
- [31] Draw MD [online]. [cit.2012-6-10] Dostupné na internete: <<http://www.drawmd.com/>>
- [32] inMotion 3D for iPad on the iTunes App Store [online]. [cit.2012-6-10] Dostupné na internete: <<https://itunes.apple.com/us/app/inmotion-3d/id441783434?mt=8>>
- [33] Certified Web Based HER, EMR & Practice Management Solution | IOS Health Systems [online]. [cit.2012-6-10] Dostupné na internete: <<https://itunes.apple.com/us/app/inmotion-3d/id441783434?mt=8>>
- [34] IEC 60601 Wikipedia, the free encyclopedia [online]. [cit.2012-6-10] Dostupné na internete: <http://en.wikipedia.org/wiki/IEC_60601>
- [35] REST: architektura pro webové API | Zdroják [Online]:[cit.2013-02-15] Dostupné na internete: <<http://www.zdrojak.cz/clanky/rest-architektura-pro-webove-api/>>
- [36] Mark L.Murphy: Android 2 – Průvodce programováním mobilních aplikací. Brno: Computer Press, a.s., 2011. 377 s. ISBN 978-80-251-31
- [37] About the Eclipse Foundation [online]. [cit.2013-04-04] Dostupné na internete: <<http://www.eclipse.org/org/>>
- [38] Apache friends – XAMPP for Windows [online]. [cit.2013-04-04] Dostupné na internete <<http://www.apachefriends.org/en/xampp-windows.html>>

Prílohy

Príloha A: Používateľská príručka

Príloha B: Systémová príručka

Príloha C: Metodická príručka

Príloha D: CD médium – diplomová práca v elektronickej podobe, zdrojové kódy aplikácie, modely k aplikácii a príručky v elektronickej podobe.

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Android v medicínskych aplikáciách

Diplomová práca

ElIdCard

POUŽIVATEĽSKÁ PRÍRUČKA

Študijný program:	Informatika
Študijný odbor:	9.2.1 Informatika
Školiace pracovisko:	Katedra počítačov a informatiky (KPI)
Školiteľ:	Doc. Ing. Zdeněk Havlice, CSc.
Konzultant:	Ing. Juraj Vízi

Obsah

Zoznam obrázkov	3
1 Formulácia úlohy	4
2 Inštalácia programu.....	5
2.1 Požiadavky na technické prostriedky	5
2.2 Požiadavky na programové prostriedky	5
2.3 Inštalácia programu	6
2.4 Zoznam služieb poskytovaných pre používateľov	6
3 Použitie aplikácie.....	7
4 Obmedzenia programu.....	15
4.1 Chybové hlásenia.....	15
5 Možné rozšírenia	16

Zoznam obrázkov

Obrázok 1: Ikona aplikácie	7
Obrázok 2: Prihlásenie do systému	8
Obrázok 3: Registrácia.....	8
Obrázok 4: Úvodné menu lekára	9
Obrázok 5: Informácie o lekárovi	9
Obrázok 6: Zoznam pacientov	10
Obrázok 7: Úvodné menu pre pacienta.....	10
Obrázok 8: Osobné údaje pacienta	11
Obrázok 9: Zdravotné údaje	11
Obrázok 10: Zoznam chorobopisov a pridanie chorobopisu	12
Obrázok 11: Vytvorenie účtu pre pacienta	12
Obrázok 12: Osobné údaje pre pacienta	13
Obrázok 13: Lieky	13
Obrázok 14 Úvodné menu pre pacienta.....	14
Obrázok 15: Chybné prihlasovacie údaje	15

1 Formulácia úlohy

Aplikácia ELIdCard má predstavovať elektronickú zdravotnú kartu. Ide o program určený lekárom a jeho pacientom. Tento program je možné využívať na zariadeniach s operačným systémom Android verziou 4.0 -Ice Cream Sandwich, alebo s novším. Tento systém bol vytvorený pomocou voľne dostupného softvéru ako balíček vývojových nástrojov pre Android, ktorý obsahuje vývojové prostredie Eclipse s importovaným doplnkom pre vývoj Android aplikácií. V tomto balíčku sa tiež nachádza emulátor - SDK manažér, pomocou ktorého je možné skúšať aplikácie na rôznych verziách platformy Android. Aplikačná časť ELIdCard bola naprogramovaná v jazyku Java. Grafické používateľské rozhranie je vytvorené v jazyku XML. Databázová časť je v programovacom jazyku SQL, ktorá je s aplikačnou časťou prepojená pomocou skriptov napísaných v php. Dokumentácia obsahuje softvérové a hardvérové požiadavky, podrobný postup práce s aplikáciou, obmedzenia aplikácie a možné rozšírenia.

2 Inštalácia programu

2.1 Požiadavky na technické prostriedky

1. Optimálne požiadavky pre mobilné zariadenie:

- Zariadenie s operačným systémom Android 4.0 Ice Cream Sandwich
- Potrebný priestor pre aplikáciu - 7MB

2. Optimálne požiadavky pre osobný počítač:

- CPU- Intel (R) Core™ i5-2430M CPU @ 2,40 GHz
- RAM- 4,00 GB
- Potrebný priestor pre vývoj a samotnú aplikáciu - 3 GB

2.2 Požiadavky na programové prostriedky

Operačný systém:

- Windows XP 32 bitový,
- Windows Vista 32/ 64 bitový,
- Windows 7 32/64 bitový.

Vývojové prostriedky:

- Android vývojové nástroje (Build: v21.0.1-543035):
 - Eclipse Platform, JDT, CDT, EMF, GEF and WTP s doplnkom pre vývoj aplikácií pre Adnroid – ADT plugin
 - Android SDK manager, Revision 21.0.1
 - SQLite3
 - Java SE JDK 7
- XAMPP
 - Apache server
 - MySQL
 - PHP
- Microsoft Office Visio 2007

2.3 Inštalácia programu

Aplikácia má viacero možností inštalácie. Ak sa aplikácia nachádza na internetovom obchode Google Play, ktorý slúži aj na siahnutie aplikácií, tak je potrebné ju vyhľadať podľa názvu a stiahnuť. Následne zbehnú aj inštalácia. Ak nie je možný prístup k internetu a teda nie je možné aplikáciu nájsť v internetovom obchode, tak je možné stiahnuť aplikáciu aj z počítača, pripojením smartphonu pomocou kábla. V mobilnom zariadení je potrebné v nastaveniach nájsť položku, ktorá má na starosti nastavenia aplikácií. V tej je možné spravovať aplikácie, povoliť inštaláciu neznámych zdrojov a ďalšie. Tam je potrebné nájsť nastavenia pre vývoj a potvrdiť ladenie USB.

Na osobnom počítači je potrebné mať vývojové prostredie Eclipse s doplnok pre Android a emulátor. Do tohto vývojového prostredia je potrebné importovať aplikáciu, ktorá je na CD. Ďalej je potrebné mať nainštalovaný program pre konkrétne zariadenie, pre mobilné zariadenia značky HTC sa volá program - HTC sync Slúži na synchronizáciu počítača a mobilného zariadenia. Teraz keď aplikáciu v Eclipse dáme spustiť na mobilnom zariadení, ktoré je pripojené káblom. Aplikácia by sa mala preniesť a hneď aj spustiť.

2.4 Zoznam služieb poskytovaných pre používateľov

1. Lekár

- Aktualizácia údajov o ambulancii a jeho činnosti
- Pridanie nových pacientov do databázy
- Prehľadnejšie oddelené osobné a zdravotné informácie
- Zoznam chorobopisov zoradený podľa dátumu
- Prezeranie databázy liekov
- Možnosť prečítať si viac informácií o liekoch aj na internete

2. Pacient

- Prezeranie osobných údajov
- Prezeranie aktuálnych údajov o lekárovi a jeho činnosti
- Prezeranie zdravotných údajov
- Prezeranie záznamov z jednotlivých návštev u lekára
- Prehľad liekov predpísaných pri jednotlivých chorobopisoch

3 Použitie aplikácie

Aplikácia sa nachádza priamo v hlavnom menu medzi ostatnými položkami pod názvom EIdCard, ako je možné vidieť na Obrázok 1.



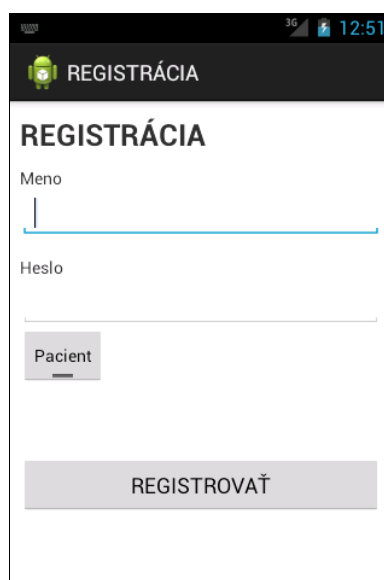
Obrázok 1: Ikona aplikácie

Spustenie aplikácie je veľmi jednoduché, pomocou tlačidiel sa dostaneme na ikonku aplikácie a potvrdíme. Zobrazí sa hlavné okno s názvom PRIHLÁSENIE DO SYSTÉMU(Obrázok 2).



Obrázok 2: Prihlásenie do systému

Ak je aplikácia spustené prvýkrát lekárom, tak je potrebné, aby sa zaregistroval. Registráciu spustí kliknutím na slovíčko Registrácia, ktoré sa nachádza pod tlačidlom Prihlásiť. Zobrazí sa okno (Obrázok 3), v ktorom je potrebné zadať prihlasovacie meno a heslo, ktoré bude lekár využívať. Tieto údaje sa ukladajú v databáze v šifrovanej podobe.



Obrázok 3: Registrácia

Po zaregistrovaní sa ako lekár sa objaví úvodné menu s tlačidlami: informácie o lekárovi, pacienti, lieky a odhlásenie (Obrázok 4)



Obrázok 4: Úvodné menu lekára

Zvolením prvej možnosti – informácie o lekárovi (Obrázok 5), si môže lekár zapísať údaje o svojej činnosti a ambulancii, ktoré sa zobrazujú aj jeho pacientom. Tieto informácie môže kedykoľvek zmeniť.



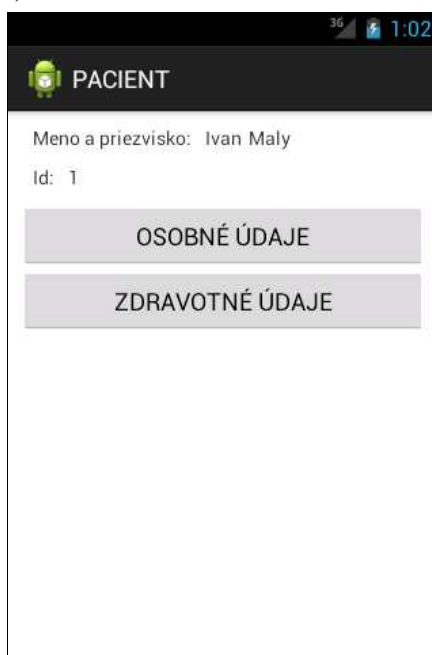
Obrázok 5: Informácie o lekárovi

Výberom druhej možnosti sa dostane do zoznamu pacientov. V zozname pacientov sa zobrazujú mena, priezviská a rodné čísla pacientov ().



Obrázok 6: Zoznam pacientov

Po výbere jednotlivého pacienta sa zobrazí menu, ktoré obsahuje tlačidlá: osobné údaje, zdravotnú údaje (Obrázok 7).



Obrázok 7: Úvodné menu pre pacienta

Kliknutím na osobné údaje sa zobrazia údaje ako meno, priezvisko, adresa, poisťovňa, rodné číslo, email a telefónne číslo (Obrázok 8). Tieto údaje vie pacientovi meniť len lekár.



ÚPRAVA OSOBNÝCH ÚDAJOV ...

Rodné číslo
9305079234

Priezvisko pacienta
Malý

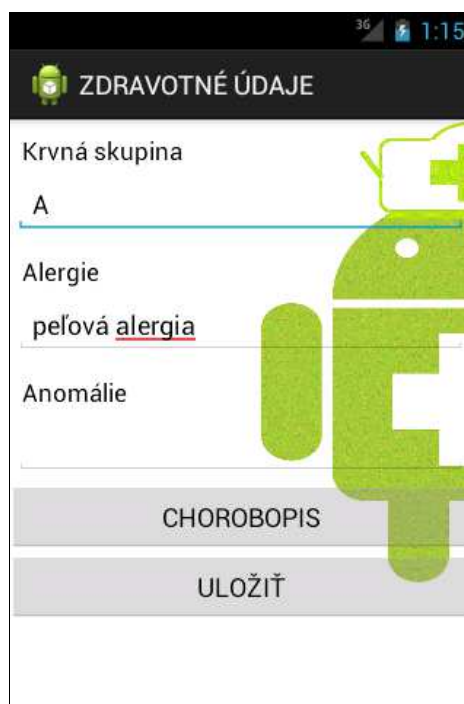
Meno pacienta
Ivan

Poistovňa
DÔVERA

Dátum
1993-03-02

Obrázok 8: Osobné údaje pacienta

Zdravotné údaje (Obrázok 9) obsahujú tieto položky - alergie, anomálie a krvnú skupinu.



ZDRAVOTNÉ ÚDAJE

Krvná skupina
A

Alergie
peľová alergia

Anomálie

CHOROBOPIS

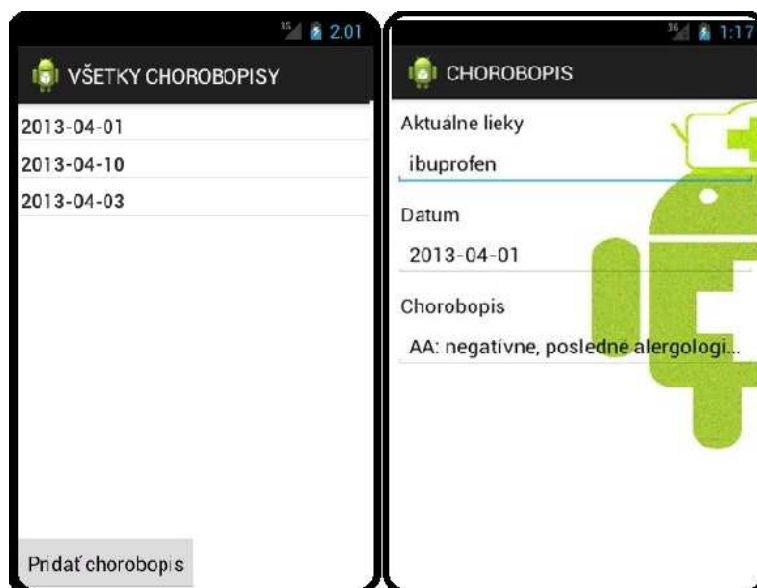
ULOŽIŤ

Obrázok 9: Zdravotné údaje

Cez toto okno sa vie lekár dostať k chorobopisu pacienta. Je to vlastne zoznam dátumov chorobopisov. Ak sa lekár rozhodne pridať chorobopis pacienta, tak musí vyplniť –

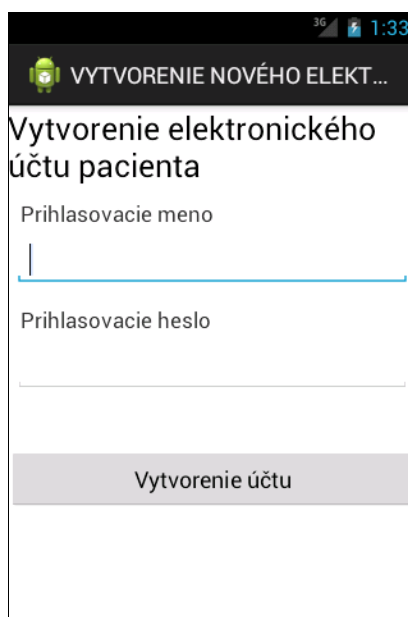
- dátum, ktorý sa zobrazí v zozname,
- aktuálne lieky, ak mu niečo predpisuje

- záznam s akým problémom pacient prišiel



Obrázok 10: Zoznam chorobopisov a pridanie chorobopisu

Ak si lekár pridá pacienta do databázy je potrebné najprv zadať prihlasovacie meno a heslo pre pacienta (Obrázok 11) a potom zadať osobné údaje a zdravotné údaje. Prípadne doplniť hneď prvý záznam do chorobopisu.

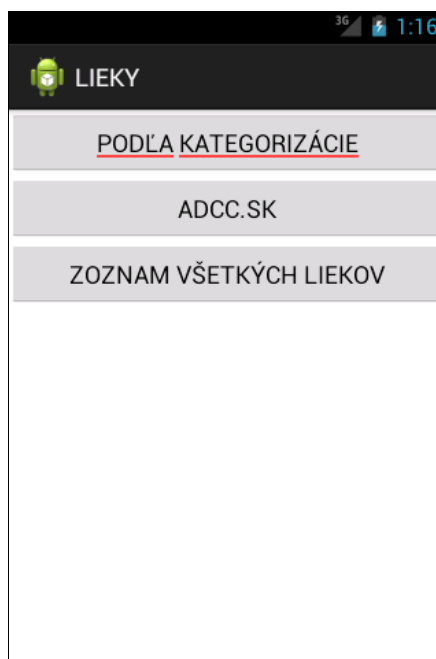


Obrázok 11: Vytvorenie účtu pre pacienta



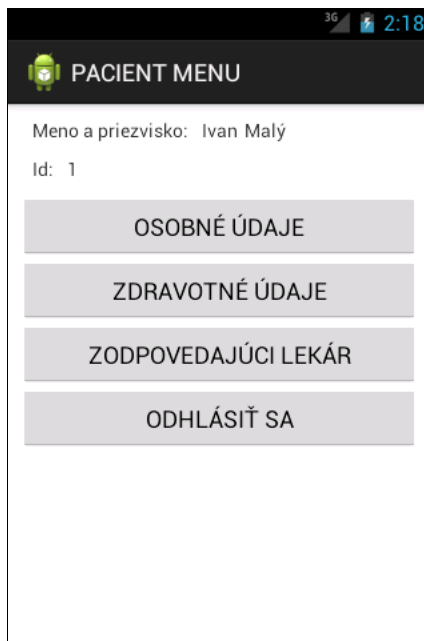
Obrázok 12: Osobné údaje pre pacienta

Lekár má v úvodnom menu ešte tlačidlo – Lieky (Obrázok 13). Po kliknutí na toto tlačidlo sa mu zobrazí obrazovka s ďalšími možnosťami. Vzhľadom na to, že existuje mnoho databáz liekov, ale nie sú prístupné ľuďom, ktorí nie sú lekármi bolo potrebné túto časť pripraviť inak. Keďže existuje mnoho rozdelení liekov, či už podľa typu chorôb, podľa špecializácií lekárov, podľa rôzneho charakteru liekov tak som v aplikácii uviedla možnosť rozdelenia napríklad podľa kategórií, zoznam všetkých liekov a externý link na stránku s dôležitými údajmi.



Obrázok 13: Lieky

Pacient sa vie prihlásiť do systému, len vtedy ak mu vytvorí účet lekár. Pacient nemôže meniť údaje. Všetky údaje sú mu prístupné len na čítanie. Úvodné menu, ktoré sa zobrazí pacientovi je vidieť na obrázku - Obrázok 14. Po kliknutí na osobné údaje sa zobrazia vyplnené polia s osobnými informáciami.



Obrázok 14 Úvodné menu pre pacienta

4 Obmedzenia programu

Program by mal byť dostatočne optimalizovaný na väčšiu databázu pacientov a rôznych lekárov. Počas behu aplikácie sa nevyskytujú žiadne obmedzenia ak sú údaje napísané správne.

4.1 Chybové hlásenia

V tejto verzii sa objavuje len jedno chybové hlásenie. Zobrazí sa vtedy, keď sa pokúsi prihlásiť niekto kto nie je pridaný v databáze(Obrázok 15).



Obrázok 15: Chybné prihlasovacie údaje

5 Možné rozšírenia

Aplikáciu je ďalej možné rozšíriť napríklad zameraním sa na lieky, ktoré vidí lekár. Databáza liekov je príliš obširná na aplikovanie. Ide o veľkú databázu údajov, ktorá môže byť rozdelená podľa viacerých kategórií. Tieto jednotlivé rozdelenia by sa mohli implementovať, aby sa lekárovi uľahčilo hľadanie a tak mal k dispozícii menšiu skupinu liekov, z ktorej je schopný vybrať ten správny. Taktiež by mohlo byť spracované aj to, že keď chce lekár priradiť liek pacientovi tak by sa prepojil na databázu liekov a vybral by si. Tu by sa mohlo kontrolovať či liek sa nenachádza medzi položkou kde sú alergie daného pacienta.

Databázy liekov už existujú. Táto databáza musí byť neustále aktualizovaná. Ide o lieky, ktoré môže predpísať len lekár. V aplikácií by sa mohli nachádzať aj voľne dostupné lieky, bez predpisu, ktoré by mohli byť sprístupnené aj pacientom. Pre pacientov by mohlo byť zaujímavé ak by pri vyhľadávaní konkrétneho lieku sa zobrazili aj iné možné alternatívy s rovnakým účinkom.

Ďalším možným rozšírením aplikácie by mohlo byť spracovanie kalendára, ktorý by mohol byť zdieľaný. Lekár by si mohol nastaviť termíny, na ktoré sa môžu pacienti prihlasovať. Pacienti by videli voľné termíny v konkrétne dni, videli by lekárovi neprítomnosti prípadne dovolenku a kto ho dané dni zastupuje. Pacient po výbere termínu návštevy by mohol poslať žiadosť lekárovi, ktorý by ju potvrdil. Lekár by mohol pridávať pacientov tiež, a tieto termíny by sa v zdieľanej časti nezobrazovali. Týmto spôsobom by sa uľahčil proces objednávanie pre pacientov. Výhodou pre lekára by bol interaktívny kalendár, ktorý by sprehľadnil návštevy a vedel by si napríklad v kalendári kliknúť na meno pacienta, ktoré by sa zobrazilo z databázy s konkrétnymi údajmi. A tak by nemusel vyhľadávať v celom zozname.

Vylepšením aplikácie môže byť možná integrácia s inými systémami. Avšak najlepšie slovenskými.

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Android v medicínskych aplikáciách
Diplomová práca
Systemová príručka

Študijný program: Informatika
Študijný odbor: 9.2.1 Informatika
Školiace pracovisko: Katedra počítačov a informatiky (KPI)
Školiteľ: Doc. Ing. Zdeněk Havlice, CSc.
Konzultant: Ing. Juraj Vízi

Obsah

Zoznam obrázkov	3
1 Funkcie programu.....	4
2 Analýza riešenia	5
3 Popis programu	6
3.1 Modely.....	6
3.2 Popis riešenia.....	11
4 Popis tried programu.....	15
4.1 Balíček com.medical.elidcard1	17
4.2 Balíček com.medical.elidcard1.kniznica.....	18
4.3 Balíček com.medical.elidcard1.lieky	19
4.4 Balíček com.medical.elidcard1.pacient.chorobopis	20
4.5 Balíček com.medical.elidcard1.pacient.d'alšie	21
5 Obmedzenia programu.....	25
5.1 Chybové hlásenia.....	25
6 Požiadavky	26
6.1 Požiadavky na technické prostriedky	26
6.2 Požiadavky na programové prostriedky pri preklade.....	26
7 Zhrnutie riešenia	27
Zoznam použitej literatúry	28

Zoznam obrázkov

Obr. 1 Diagram znázorňujúci beh aplikácie	6
Obr. 2 Logická schéma	7
Obr. 3 Fyzická schéma.....	7
Obr. 4 Používateľský model lekára.....	10
Obr. 5 Používateľský model pacienta	11
Obr. 6 Spôsob posielania a prijímania dát	14
Obr. 7 Diagram tried	16
Obr. 8 Chybné prihlasovacie údaje.....	25

1 Funkcie programu

Aplikácia EIdCard patrí medzi medicínske aplikácie určené pre lekárov a ich pacientov. Ide o aplikáciu vytvorenú pre operačný systém Android s verziou 4.0 Ice Cream Sandwich. K aplikácii majú prístup 2 skupiny ľudí s rôznymi právami.

Prvou skupinou sú lekári, ktorí sa do aplikácie musia zaregistrovať. Po zaregistrovaní môžu pridať informácie o svojej činnosti, adresu ambulancie, telefónne číslo, meno sestričky a emailovú adresu. Lekári si najprv musia pridať pacientov do databázy. Pridanie pacienta do databázy prebieha tak, že lekár musí zadať prihlasovacie údaje pacienta a potom môže zadávať osobné údaje pacienta ako meno, priezvisko, rodné číslo, adresa trvalého pobytu, emailová adresa, telefónne číslo a názov poisťovne. Ďalej je potrebné pridať základné zdravotné údaje o pacientovi a tu patria – typ krvnej skupiny, anomálie, alergie. Zo základných zdravotných údajov sa vie prepnúť do chorobopisu daného pacienta. Chorobopis je zoznam jednotlivých návštev pacienta. Ak chce pridať záznam návštevy, tak musí zadať dátum návštevy, aktuálne lieky ak nejaké lekár predpíše a popis dôvodu návštevy. Lekár má k dispozícii aj databázu liekov, ktorá je v aplikácii len navrhnutá. Ide o rozdelenie liekov podľa kategórii, zoznam všetkých liekov a externý odkaz na ďalšie informácie o liekoch

Druhou skupinou sú pacienti, ktorí majú na rozdiel od lekárov obmedzené funkcie. Nemôžu žiadne údaje meniť. Pacientom sa zobrazí po prihlásení menu s 3 tlačidlami: Informácie o lekárovi a jeho činnosti, osobné údaje a zdravotné údaje. Po kliknutí na informácie o lekárovi sa zobrazia údaje, ktoré zadal lekár do systému. Ďalej si môže prezerat' osobné údaje a zdravotné údaje. Zo zdravotných údajov sa vie ďalej dostať aj na chorobopisy a teda si vie prezerat' jednotlivé záznamy svojich návštev.

2 Analýza riešenia

Ide o aplikáciu s väčším množstvom údajov, ktoré budú časom stále pribúdajú. Preto databázová časť nemôže byť riešená prostredníctvom lokálnej databázy – SQLite3, ktorá beží na operačnom systéme Android. SQLite3 je využívaná len na uchovanie prihlasovacích údajov na danom zariadení. Celá databáza je vytvorená na serveri MYSQL, ktorý predstavuje vzdialený databázový server. Prihlasovacie údaje sú šifrované pomocou funkcie hash.

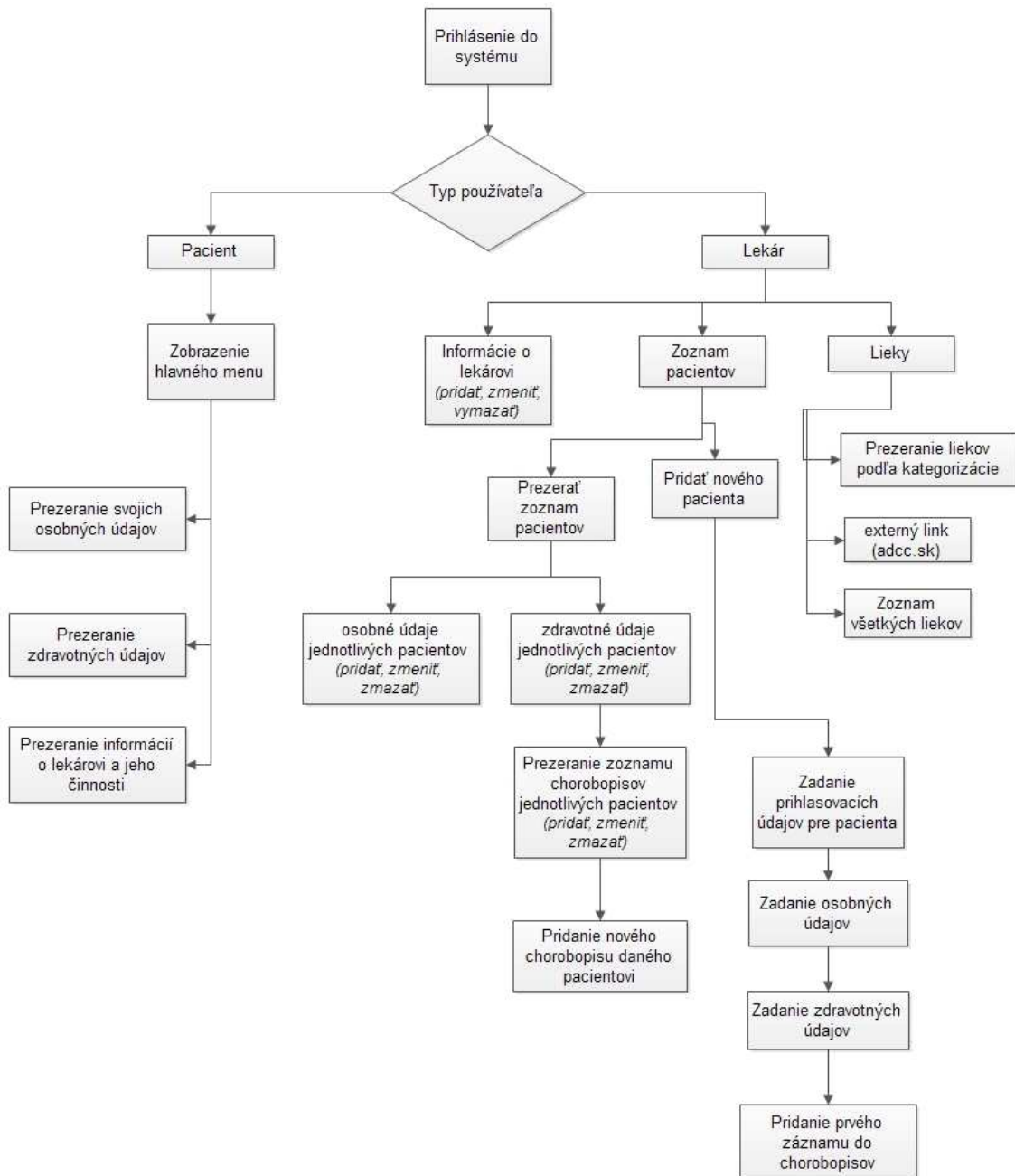
Aplikáciu je určená pre mobilné zariadenia s operačným systémom Android a verziou 4.0 alebo novšou. Vybrala som túto verziu, pretože vývoj aplikácii je zjednotený pre Smartphony a tablety. Takže stačí urobiť jednu aplikáciu. Na samotnú implementáciu som použila vývojový nástroj Android. Je to balíček programov a doplnkov potrebných pre tvorbu Android aplikácií. Nachádza sa v ňom vývojové prostredie Eclipse, ktoré má už pridaný a nastavený doplnok pre vývoj Android aplikácií. Taktiež obsahuje SDK manažér, ktorý slúži ako emulátor. Tento emulátor je potrebné nastaviť podľa toho pre aké zariadenia a akú verziu Androidu bude implementovaná aplikácia.

Túto platformu som si vybrala kvôli jeho súčasnému rozšíreniu vo svete a neustálym pribúdaním aplikácií pre tento systém v rôznych odvetviach

3 Popis programu

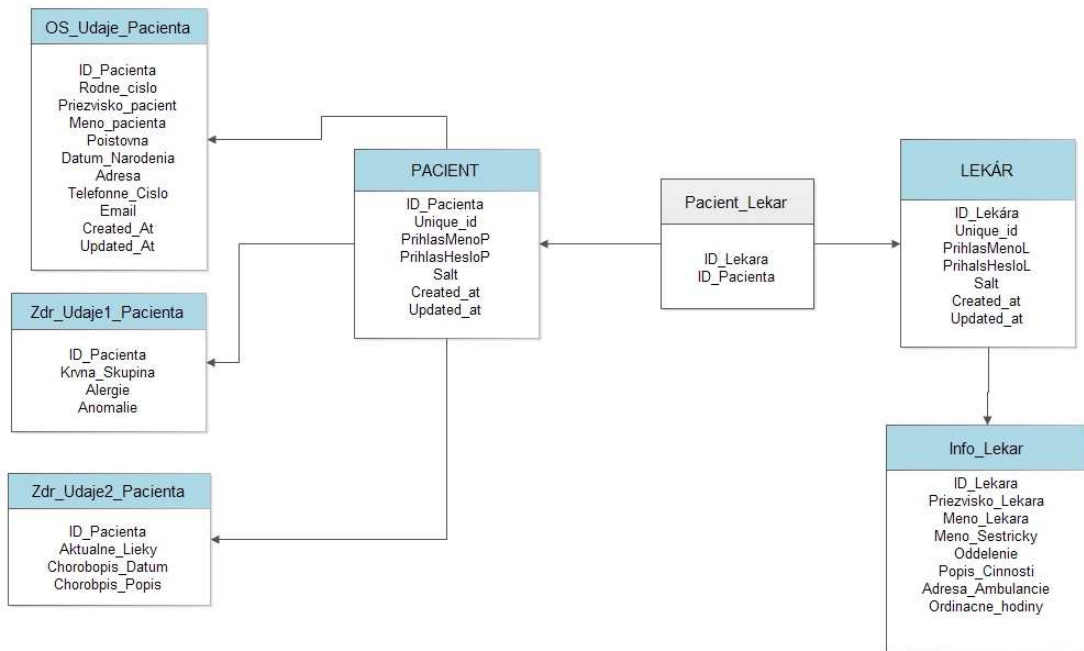
3.1 Modely

Diagram (Obr. 1) znázorňuje beh aplikácie a možnosti jednotlivých prihlásených používateľov.

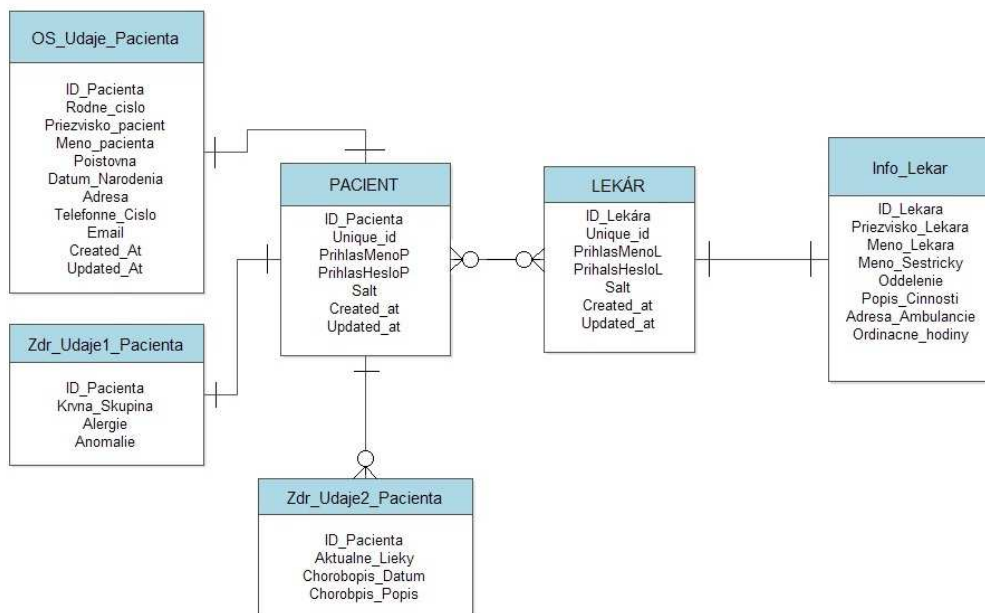


Obr. 1 Diagram znázorňujúci beh aplikácie

Pred samotnou implementáciou databázy je potrebné navrhnuť logickú (Obr. 2) a fyzickú (Obr. 3) schému databázy. Logická schéma slúži na deklaráciu dát v databáze. Fyzická schéma slúži na popis dát na fyzickej báze. Znázorňuje vzťahy medzi tabuľkami.



Obr. 2 Logická schéma



Obr. 3 Fyzická schéma

Databázový skript:

```
create table lekar(
```

```
ID_lekara int(2) primary key auto_increment not null unique,
```

```
unique_id varchar(23)not null unique,
```

```
PrihlasMenoL varchar(30) not null unique,
```

```
PrihlasHesloL VARCHAR(30)not null,
```

```
salt varchar(10),
created_at datetime,
updated_at datetime null);
```

```
create table os_udaje_lekara(
ID_lekara INT(2) not null,
Priezvisko_lekara varchar (50) not null,
Meno_lekara varchar (20) not null,
Meno_sestricky varchar (70),
Oddelenie varchar (70) not null,
Popis_Cinnosti varchar (150),
Adresa_amb varchar (50) not null,
Ord_hodiny integer (100),
foreign key (ID_lekara) references lekar (ID_lekara)
ON DELETE RESTRICT
ON UPDATE CASCADE) ;
```

```
create table pacient(
ID_pacienta int(5) primary key auto_increment not null unique,
unique_id varchar(23)not null unique,
PrihlasMenoP VARCHAR(40) not null,
PrihlasHesloP VARCHAR(30) not null,
salt varchar(10),
created_at datetime,
updated_at datetime null);
```

```
create table os_udaje_pacienta(
ID_pacienta int(5) not null,
Priezvisko_pacienta VARCHAR (50),
Meno_pacienta VARCHAR(20),
Poistovna VARCHAR(30) not null,
Datum_narodenia date not null,
Adresa VARCHAR(50) not null,
```

```
Email VARCHAR(100)not null unique,  
MOBIL INTEGER(10),  
created_at datetime,  
updated_at datetime null,  
foreign key (ID_pacienta) references pacient (ID_pacienta)  
ON DELETE RESTRICT  
ON UPDATE CASCADE) ;
```

```
create table zdr_udaje1_pacienta(  
ID_pacienta int(5) not null,  
Krvna_skupina CHAR(2),  
Alergie VARCHAR(100),  
Anomalie VARCHAR(100),  
foreign key (ID_pacienta) references pacient (ID_pacienta)  
ON DELETE RESTRICT  
ON UPDATE CASCADE);
```

```
create table zdr_udaje2_pacienta(  
ID_pacienta int(5) not null,  
Aktualne_lieky VARCHAR(500),  
Chorobopis_datum DATE,  
Chorobopis_popis VARCHAR(500),  
created_at datetime,  
updated_at datetime null,  
foreign key (ID_pacienta) references pacient (ID_pacienta)  
ON DELETE RESTRICT  
ON UPDATE CASCADE) ;
```

```
Create table pacient_lekar(  
ID_lekara int(2) not null,  
ID_pacienta int(5) not null,  
Foreign key (ID_lekara) REFERENCES lekar (ID_lekara)  
ON DELETE RESTRICT
```

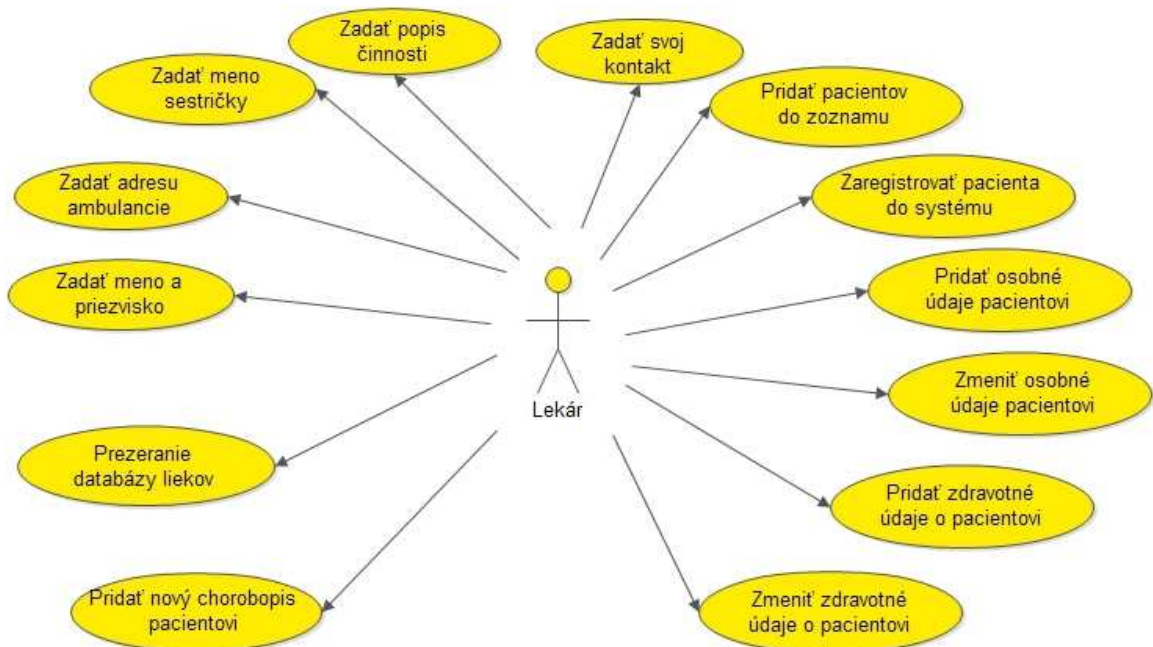
ON UPDATE CASCADE,

Foreign key (ID_pacienta) REFERENCES pacient (ID_pacienta)

ON DELETE RESTRICT

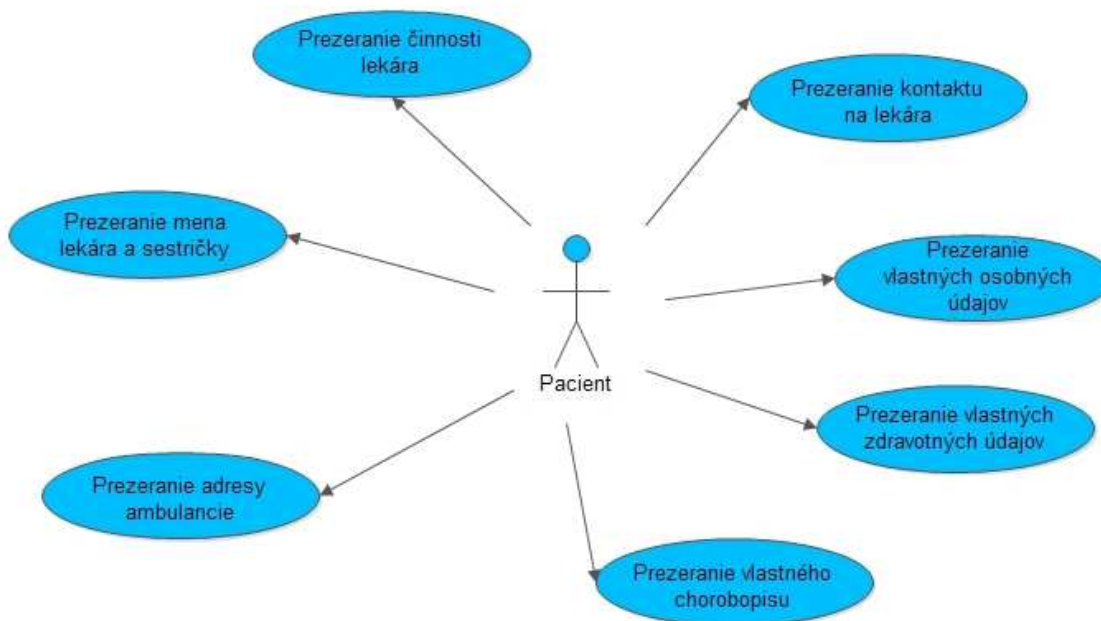
ON UPDATE CASCADE);

Používateľské modely zobrazujú možnosti daného používateľa v aplikácii. Na prvom obrázku (Obr. 4) je uvedený lekár. Lekár môže zadať údaje o sebe a svojej činnosti do systému. Taktiež môže pridávať pacientov do systému tak že ich najskôr zaregistruje, aby sa mohli do systému prihlásiť aj oni. Po registrácii je možné zadať osobné údaje pacienta a zdravotné údaje, v ktorých je možnosť zadať chorobopis pre pacienta. Ďalej môže lekár prezerat' rôzne kategórie liekov.



Obr. 4 Používateľský model lekára

Druhým používateľom systému tvoria pacienti (Obr. 5). Tí sa nemôžu do systému registrovať. Prihlasovacie údaje dostanú od lekára. Vedia si prezerat' údaje o lekárovi – jeho meno, priezvisko, adresu ambulancie, meno sestričky, kontakt na lekára a ďalšie. Má možnosť prezerat' si údaje svoje osobné údaje, ktoré má k dispozícii lekár. Taktiež môže prezerat' základné zdravotné údaje a zoznam vlastných chorobopisov.



Obr. 5 Používateľský model pacienta

3.2 Popis riešenia

3.2.1 Platformy použité pri vývoji aplikácie

Na vývoj aplikácií pre operačný systém Android je uprednostnené vývojové prostredie Eclipse IDE, ktoré je považované za oficiálny vývojový nástroj. Toto prostredie patrí medzi voľne dostupné programy. Výhodou je jednoduché a intuitívne ovládanie, ale taktiež aj to, že samotný program nezaberá veľa miesta. Programovanie v špecifických jazykoch alebo pre individuálne systémy ako je aj Android je prispôbené doplnkami, ktoré si musí užívateľ stiahnuť. Tie sú tiež zadarmo. Konkrétne pre vývoj aplikácií na platformu Android na oficiálnej stránke pre vývojárov je možné si stiahnuť súbor s názvom – Android nástroj pre vývojárov, ktorý obsahuje emulátor s manažerom a Eclipse IDE, v ktorom sú už pridané doplnky na vývoj. Stačí len nastaviť manažér emulátora, v ktorom je potrebné stiahnuť si súbory slúžiace pre vývoj na konkrétnu verziu Androidu. V tomto vývojovom prostredí sa využíva programovací jazyk Java na aplikačnú logiku a na grafické používateľské rozhranie je programovací jazyk XML. Vytváranie grafického rozhrania prebieha pomocou špeciálneho editora, ktorý umožňuje výber z predvolených najpoužívanejších typov a následne ich úpravy cez vlastnosti. Je to napríklad vkladanie tlačidiel, textu, zoznamu a ďalších. Tak aj človek, ktorý neovláda XML si vie vytvoriť obrazovku so základnými prvkami. [2]

Návrh databázovej časti som robila v programe Microsoft Office Visio 2007, ktorý je dostupný na MSDNA pre študentov. Tento program slúži na vyvíjanie rôznych návrhov diagramov, procesov, štruktúr. Práca s týmto programom je veľmi rýchla a jednoduchá, pretože je zostavený ako väčšina produktov od firmy Microsoft konkrétne v skupine programov Office. Výhodou tohto programu pri návrhu databázových diagramov sú predvolené možnosti, pomocou ktorých je program schopný vygenerovať kód v jazyku SQL.

Ako interpretáciu vzdialeného prístupu s databázou používam program XAMPP. Je to program, ktorý je vyvíjaný skupinou Apache Friends. Ide o program, ktorý obsahuje Apache, MySQL, PHP a Perl. Je voľne dostupný a nie je potrebná žiadna inštalácia. Stačí stiahnuť verziu pre daný systém, rozbaľiť a spustiť. Apache je program, ktorý predstavuje webový server. Apache je taktiež voľne dostupný. Apache spolu s PHP a MySQL sú veľmi často využívané k vývoju dynamických internetových stránok. MySQL je databázový server. Údaje mojej aplikácie sa ukladajú do MySQL databázy pomocou phpMyAdmin. To je nástroj v internetovom prehliadači ktorý je prispôsobený na vytváranie databázových skriptov. Komunikácia databázy a mojej aplikácie prebieha pomocou skriptov v PHP programovacím jazyku. [3]

3.2.2 Možnosti integrácie databázovej časti

Existuje viacero možností ako prepojiť databázovú časť s aplikačnou. Dáta potrebné pre aplikáciu môžu byť uložené napríklad v jednoduchých súboroch, v lokálnej databáze SQLite, alebo na vzdialenom serveri, ku ktorému sa prístupuje prostredníctvom internetu.

SQLite – je lokálna databáza v systéme Android, ktorá je zabudovaná vo vykonávacom prostredí. A práve to umožňuje každej aplikácii vytvoriť vlastnú databázu. Práca s ňou je veľmi jednoduchá pre ľudí, ktorí už majú nejaké skúsenosti s inými rozhraniami SQL. Patrí medzi najrozšírenejšie, pretože obsahuje čisté SQL rozhranie, zaberá veľmi málo miesta (275 kB) a je pomerne rýchla.

SQLite používa pre dotazy, manipuláciu s dátami a definíciu dát dialekt jazyka SQL. Je efektívna pri práci s pamäťou, pretože ju môže vykonávacie prostredie systému obsahovať celú a nie len jednu časť, aby sa znížil jej celkový objem. Veľký rozdiel medzi SQLite a inými databázami je v typovaní dát. Do akéhokoľvek stĺpca je možné uložiť dáta ľubovoľnej hodnoty. SQLite nepodporuje niekoľko štandardných vlastností SQL:

- obmedzenie typu použitím cudzieho kľúča (angl. FOREIGN KEY),
- vnorená transakcia,
- čiastočné vonkajšie spojenie typu sprava (angl. RIGHT OUTER JOIN)
- úplné vonkajšie spojenie (angl. FULL OUTER JOIN),
- niektoré príkazy vykonávajúce modifikáciu tabuľky (angl. ALTER TABLE)

Pre aplikácie, v ktorých majú prístup viacerí užívatelia je lepšie mať dáta uložené na vzdialenom serveri. Po spustení aplikácie sa načítajú aktuálne údaje z databázy. Komunikácia s webovými službami prebieha pomocou protokolu SOAP (Simple Object Access Protocol). Je to protokol, ktorý zabezpečuje výmenu správ založených na XML prostredníctvom siete. SOAP je nástupcom XML-RPC protokolu. XML-RPC uchováva pravidlá, ktoré definujú ako sa používajú štandardné technológie pre potreby RPC (vzdialené volanie procedúr). V systéme Android sa nenachádzajú API protokolu SOAP a ani protokolu XML-RPC. Táto komunikácia je v Androide vyriešená tak, že súčasťou systému Android je knižnica Apache HttpComponents, nad ktorou je možné umiestniť vrstvu protokolov SOAP alebo XML-RPC, alebo je možné používať prístup k webovým službám založeným na architektúre REST (Representational State Transfer). Rest je architektúra, ktorá umožňuje pristupovať k dátam na určitom mieste pomocou štandardných metód http. Definuje ako majú byť špecifikované a adresované. Využíva štandardy :

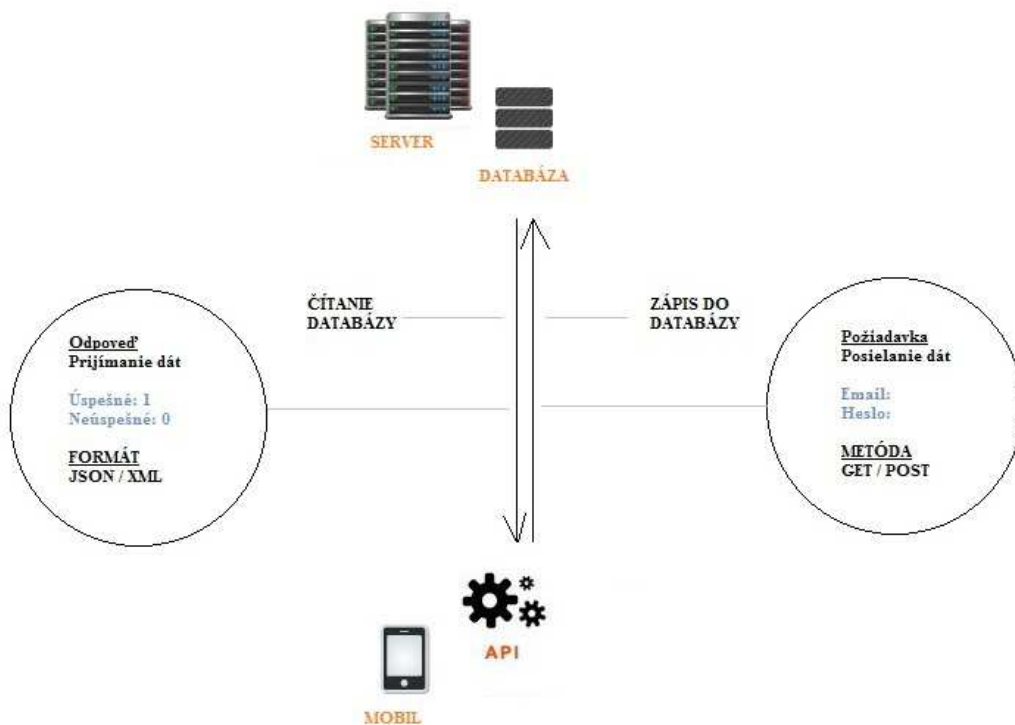
- prenosný protokol (HTTP)
- zdrojový protokol (URL)
- zdrojová reprezentácia (XML/HTML/GIF/JPEG/...)
- MIME typy (text/xml, text/html, image/gif, image/jpeg,...)

Prístup k zdrojom je jednotný a jednoduchý. REST umožňuje využitie CRUD operácií pomocou štandardných HTTP dotazov. REST definuje 4 základné metódy, ktoré sú v tabuľke (vid' Tab. 1).

REST	CRUD
POST	Create
GET	Read
PUT	Update or Create
DELETE	Delete

Tab. 1 Metódy v architektúre REST

Moja aplikácia interpretuje pripojenie na vzdialený server. Využívam v nej architektúru REST a knižnicu Apache HttpComponents. Najprv bolo potrebné si vytvoriť objekt typu `httpClient`. Vzhľadom na to že ide o rozhranie, bude sa musieť vytvoriť inštancia implementácie tohto rozhrania, napríklad trieda `DefaultHttpClient`. Do inštancie triedy `HttpRequest` sa zabalia požiadavky s rôznymi implementáciami pre jednotlivé http príkazy (napr. `get`). Následne je potrebné vytvoriť inštanciu triedy `HttpRequest` s danou implementáciou. Do nej sa vloží cieľová URL adresa a ďalšie konfiguračné údaje. Potom sa metóda predá klientovi, aby sa http požiadavka vykonala prostredníctvom metódy `execute()`. Vrátí sa objekt typu `HttpResponse` s kódom odpovede. [1]



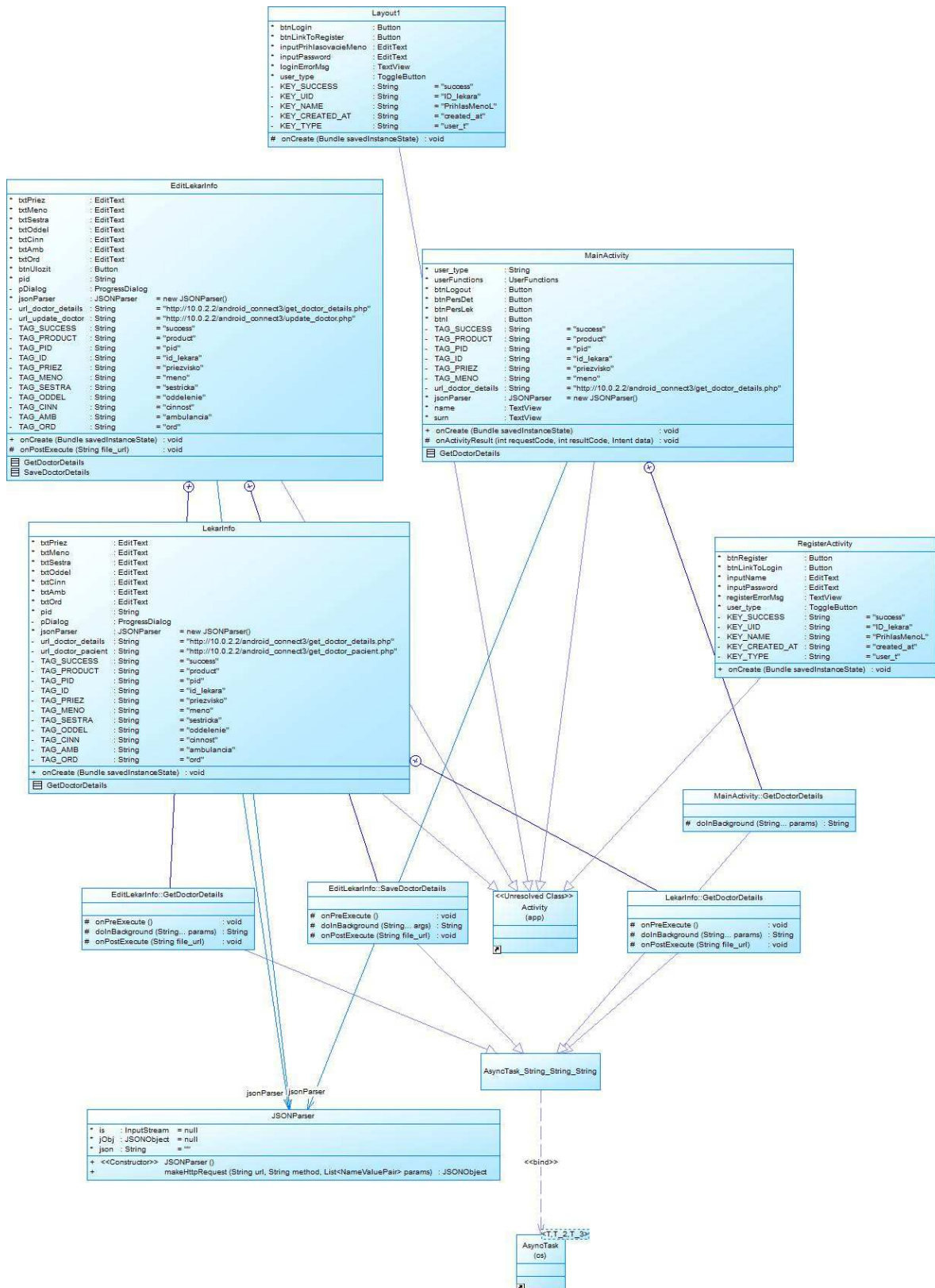
Obr. 6 Spôsob posielania a prijímania dát

4 Popis tried programu

Hlavná časť aplikácie je tvorená balíčkami:

- `com.medical.elidcard1` – obsahuje triedy, ktoré majú na starosti hlavnú aktivitu na zobrazenie úvodnej obrazovky pre lekára, zadávanie údajov o lekárovi a jeho činnosti. Taktiež je tam trieda, ktorá je dôležitá na registráciu.
- `com.medical.elidcard1.kniznica` – obsahuje triedy zaoberajúce sa lokálnou databázou SQLite3 a prácou s prihlasovacími údajmi
- `com.medical.elidcard1.lieky` – obsahuje triedy určené na prácu s liekmi, zobrazovanie kategorizácie, aktivovanie tlačidiel a iné
- `com.medical.elidcard1.pacient.chorobopis` - obsahuje triedy zaoberajúce sa zobrazovaním a pridávaním chorobopisov pre rôznych používateľov.
- `com.medical.elidcard1.pacient.d'alšie` – obsahuje triedy na vytváranie zoznamu pacientov, pridávanie nových pacientov do zoznamu, zobrazenie úvodného menu pre pacienta a iné

Pomocou diagramu tried je možné zobraziť štruktúru systému pomocou balíčkov, tried a vzťahov medzi nimi (Obr. 7).



Obr. 7 Diagram tried

Vzhľadom na veľkosť diagramu tried je priložený v prílohe na CD.

4.1 Balíček com.medical.elidcard1

4.1.1 EditLekarInfo.java

- Nastaví sa príslušný layout a získajú sa dáta o lekárovi, ktoré sa vypíšu na obrazovku zariadenia. Po stlačení tlačidla na uloženie sa spustí trieda `SaveDoctorDetails`, ktorá údaje uloží do databázy.

Metódy triedy:

`onCreate(Bundle savedInstanceState)` – slúži na zobrazenie layoutu a zavolá podtriedu `GetDoctorDetails`.

Podtriedy:

`GetDoctorDetails` - získa údaje o doktorovi z databázy a zobrazí ich na obrazovke

`SaveDoctorDetails` - zadané údaje vloží do databázy

4.1.2 JSONParser.java

Najprv sa zistí typ http požiadavky. V tejto triede sa vytvára `HttpClient`. Potom sa vytvorí http požiadavka pomocou URL adresy a parametrov. Ďalej sa vykoná http požiadavka a to čo vráti, sa uloží v http response. Z http response sa získajú http entity, z ktorých sa následne získa obsah. `BufferedReader` číta riadky z obsahu a postupne ich pridáva do sb typu `StringBuilder`, ktorý sa uloží do premennej `json`. Vytvorí sa nový objekt - `JSONObject` pomocou premennej `json`, ktorú sme získali. Ak nenastala žiadna chyba vráti sa `JSONObject`.

4.1.3 Layout1.java

- spracuje prihlasovanie používateľov

Metódy triedy:

`onCreate(Bundle savedInstanceState)` – čaká na stlačenie tlačidla – prihlásenie. Po stlačení tlačidla sa odošlú údaje do triedy `UserFunctions.java` a podtriedy `LoginUsers`. To čo vráti tá metóda sa uloží do premennej. Ak sa vrátilo v premennej že zadaný užívateľ sa úspešne prihlásil tak sa jeho údaje uložia do SQLite databázy pomocou `DatabaseHandler` a podtriedy `AddUser`. Ďalej sa zistí typ používateľa. Ak bol typ používateľa pacient tak sa spustí trieda `PacientUvod`. Ak bol lekár tak sa spustí trieda `MainActivity`. Ukončia sa všetky ostatné aktivity a nastaví sa hlavná.

4.1.4 LekarInfo.java

- nastaví sa príslušný layout a získajú sa dáta o lekárovi, ktoré sa vypíšu na obrazovku zariadenia

Metódy triedy:

onCreate(Bundle savedInstanceState) – slúži na zobrazenie layoutu a zavolá metódu GetDoctorDetails(), ktorá získa údaje o doktorovi z databázy a zobrazí ich na obrazovke.

4.1.5 MainActivity.java

- Zapne sa pri spustení aplikácie a zisťuje či je prihlásený používateľ. Ob

Metódy triedy:

onCreate(Bundle savedInstanceState) – zistí sa či je používateľ prihlásený. Ak nie je prihlásený tak sa spustí trieda Layout. Ak je prihlásený tak sa zistí typ používateľa. Ak je typ používateľa pacient, tak sa spustí PatientUvod. Ak je typ lekár tak sa pokračuje vo vykonávaní aktivity a zobrazí sa hlavné menu pre lekára.

Podtrieda:

GetDoctorDetails – získa údaje o lekárovi a vypíše jeho meno a priezvisko na layout.

4.1.6 RegisterActivity.java

- Slúži na zaregistrovanie lekára

Metódy triedy:

onCreate(Bundle savedInstanceState) – zobrazí príslušný layout a čaká na stlačenie tlačidla – zaregistrovať. Po stlačení tlačidla – zaregistrovať. Prebehne pokus o zaregistrovanie lekára do databázy. Ak bol pokus úspešný tak sa používateľské údaje uložia do SQLite databázy a spustí sa trieda MainActivity.

4.2 Balíček com.medical.elidcard1.kniznica

4.2.1 DatabaseHandler.java

- trieda určená na prácu s databázou SQLite3

Metódy triedy:

DatabaseHandler(Context context) - vytvorenie databázy

onCreate(SQLiteDatabase db) - vytvorenie tabuľky v databáze

onUpgrade() – vymazanie tabuľky

addUser() - pridanie používateľa do databázy

HashMap<String, String> getUserDetails() – získanie používateľových údajov z databázy
getRowCount() – kontrola či je používateľ prihlásený
resetTables() – vymazanie všetkých riadkov v tabuľke

4.2.2 JSONParser.java

V tejto triede sa vytvára HttpClient. Potom sa vytvorí http požiadavka pomocou URL adresy a parametrov. Ďalej sa vykoná http požiadavka a to čo vráti, sa uloží v http response. Z http response sa získajú http entity, z ktorých sa následne získa obsah. BufferedRedear číta riadky z obsahu a postupne ich pridáva do sb typu StringBuilder, ktorý sa uloží do premennej json. Vytvorí sa nový objekt - JSONObject pomocou premennej-json, ktorú sme získali. Ak nenastala žiadna chyba vráti sa JSONObject.

4.2.3 UserFunctions.java

- Obsahuje metódy na registráciu a prihlasovanie používateľov

JSONObject loginUser(String PrihlasMenoL, String password, String user_t) – spracuje prihlasovacie údaje používateľov a pošle ich triede JSONParser.java. Vráti odozvu zo servera

JSONObject registerUser(String PrihlasMenoL, String password, String user_t, String id_lekara) – spracuje zadané údaje pri registrácií používateľa s pošle ich JSONParser.java - vráti odozvu zo servera.

isUserLoggedIn(Context context) – zistí či už používateľ bol prihlásený.

logoutUser(Context context) – odhlási používateľa

4.3 Balíček com.medical.elidcard1.lieky

4.3.1 Lieky_kateg.java

- Zobrazí zoznam kategórii a po výbere niektorej kategórie sa zobrazí stránka

4.3.2 Lieky_uvod.java

- Zobrazí úvodné menu, kde sa nachádzajú rôzne ďalšie možnosti pre lieky

4.3.3 Lieky_zozn.java

- Zobrazí zoznam liekov. Po kliknutí na jednotlivé lieky sa zobrazí stránka s informáciami o tomto lieku

4.3.4 Lieky.java

- Zobrazí stránku, ktorá obsahuje informácie o všetkých liekoch

4.3.5 Web.java

- Zobrazí url adresu

4.4 Balíček com.medical.elidcard1.pacient.chorobopis

4.4.1 Pacient_ch_view.java

- Zobrazenie konkrétneho chorobopisu pre pacienta

Metódy triedy:

onCreate(Bundle savedInstanceState) – zobrazí layout a spustí podtriedu

GetPatientDetails()

Podtrieda:

GetPatientDetails() – vypíše už konkrétny chorobopis

4.4.2 Pacient_chorobopis.java

- Slúži na pridanie chorobopisu pre pacienta. Vykonáva to lekár.

Metódy triedy:

onCreate(Bundle savedInstanceState) – čaká na stlačenie tlačidla, na vytvorenie nového chorobopisu pre daného pacienta. Po stlačení tlačidla spustí podtriedu SavePatientDetails().

Podtrieda:

SavePatientDetails() – uloží nový chorobopis do systému

4.4.3 Pacient_chorobopis_all_view.java

- Slúži na zobrazenie zoznamu chorobopisov pre každého pacienta zvlášť. Je to zobrazenie určené pre lekára

Metódy triedy:

onCreate(Bundle savedInstanceState) – zobrazí príslušný layout a spustí triedu LoadAllPatients(). Taktiež sa zobrazí aj tlačidlo na pridanie chorobopisu

onActivityResult(int requestCode, int resultCode, Intent data) – ak bol pridaný chorobopis tak sa znovu vykoná aktivita

onPostExecute() - počká kým sa všetky úlohy na pozadí ukončia a potom aktualizuje zoznam pomocou patientList

Podtrieda:

LoadAllPatients() – získa z databázy všetky chorobopisy daného pacienta a uloží ich do premennej patientList

4.4.4 Pacient_chorobopis_all.java

- Slúži na zobrazenie zoznamu chorobopisov pre každého pacienta zvlášť. Je to zobrazenie určené pre pacienta

Metódy triedy:

onCreate(Bundle savedInstanceState) – zobrazí príslušný layout a spustí triedu LoadAllPatients().

onActivityResult(int requestCode, int resultCode, Intent data) – ak bol pridaný chorobopis tak sa znovu vykoná aktivita

Podtrieda:

LoadAllPatients() – získa z databázy všetky chorobopisy daného pacienta a uloží ich do premennej patientList

4.4.5 Pacient_zu_view.java

- Zobrazí lekárovi zdravotné údaje pacienta a tlačidlá na zobrazenie chorobopisov a uloženie zdravotných údajov

Metódy triedy:

onCreate(Bundle savedInstanceState) – zobrazí príslušný layout a spustí podtriedu GetPatientDetails

onActivityResult(int requestCode, int resultCode, Intent data) - po uložení všetkých údajov sa aktivita spustí znova

Podtriedy:

GetPatientDetails – získa údaje z databázy a vypíše ich na obrazovku

SavePatientDetails – po stlačení tlačidla uloží údaje do databázy

4.5 Balíček com.medical.elidcard1.pacient.d'alšie

4.5.1 AllPatientsActivity.java

- pomocou tejto triedy sa získa zoznam používateľov z databázy a dynamicky sa vytvorí zoznam. Zobrazia sa tlačidlá na pridanie pacienta.

Metódy triedy:

onCreate(Bundle savedInstanceState) – slúži na zobrazenie príslušného layoutu a taktiež zavolá metódu LoadAllPatients(), ktorá získa všetkých pacientov a uloží ich do premennej list.

onActivityResult(int requestCode, int resultCode, Intent data) – zistí či boli osobné údaje pacienta editované a prípadne aktualizuje údaje.

onPostExecute(String file_url) – ak na pozadí nebežia už žiadne metódy potom sa spustí táto metóda a aktualizuje zoznam

4.5.2 EditPatientActivity.java

- Slúži na vypísanie osobných údajov pacienta a ich editáciu

Metódy triedy:

onCreate(Bundle savedInstanceState) – zavolá podtriedu GetPatientDetails

Podtriedy:

GetPatientDetails – umožňuje vypísanie osobných informácií pacienta

SavePatientDetails – slúži na ukladanie osobných údajov pacienta

4.5.3 New_Patient.java

- Pri pridávaní pacienta lekárom do systému je potrebné zadať osobné údaje pacienta, čo sa vykoná touto triedou.

Metódy triedy:

onCreate(Bundle savedInstanceState) – čaká na stlačenie tlačidla na vytvorenie elektronického osobných údajov pacienta. Po jeho stlačení sa spustí podtrieda CreateNewPatient

Podtrieda:

CreateNewPatient – vytvorí nové osobné údaje pre daného pacienta

4.5.4 NewPatient.java

- Vytvorí nového pacienta v databáze

Metódy triedy:

onCreate(Bundle savedInstanceState) - spustí príslušný layout a čaká na stlačenie tlačidla na vytvorenie nového elektronického účtu pacienta. Po stlačení tlačidla sa vytvorí elektronický účet pacienta.

onActivityResult(int requestCode, int resultCode, Intent data) – vráti premennú do predošlej aktivity, len ak boli zadané osobné údaje.

4.5.5 Pacient_ou.java

- Vypíše údaje osobné údaje pacienta

Metódy triedy:

onCreate(Bundle savedInstanceState) – zobrazí layout a spustí podtriedu
GetPatientDetails

Podtredy:

GetPatientsDetails – získa osobné údaje z databázy a vypíše ich na obrazovku

4.5.6 Pacient_uvod.java

- Zobrazí úvodné menu pre pacientov

Metódy triedy:

onCreate(Bundle savedInstanceState) – zobrazí layout a spustí podtriedu
GetPatientDetails

Podtriedy:

GetPatientDetails – získa meno a priezvisko pacienta a vypíše ich na layout

4.5.7 Pacient_view.java

- Slúži lekárovi pri zobrazení nejakého konkrétneho pacienta. Zobrazí sa úvodné menu, kde si môže vybrať osobné alebo zdravotné údaje pacienta

Metódy triedy:

onCreate(Bundle savedInstanceState) – zobrazí layout a spustí podtriedu
GetPatientDetails

onActivityResult(int requestCode, int resultCode, Intent data) - zistí či boli osobné údaje upravované. Ak boli upravované tak spustí aktivitu znova

Podtriedy:

GetPatientDetails – zistí meno a priezvisko pacienta a vypíše ho na obrazovku

4.5.8 Pacient_zu.java

- Zobrazí pacientovi jeho zdravotné údaje a tlačidlo na zobrazenie chorobopisov

Metódy triedy:

onCreate(Bundle savedInstanceState) – zobrazí príslušný layout a spustí podtriedu

GetPatientDetails

Podtriedy:

GetPatientDetails – získa údaje z databázy a vypíše ich na obrazovku

5 Obmedzenia programu

Program by mal byť dostatočne optimalizovaný na väčšiu databázu pacientov a rôznych lekárov. Počas behu aplikácie sa nevyskytujú žiadne obmedzenia ak sú údaje napísané správne.

5.1 Chybové hlásenia

V tejto verzii sa objavuje len jedno chybové hlásenie. Zobrazí sa vtedy, keď sa pokúsi prihlásiť niekto, kto nie je pridaný v databáze(Obr. 8).



Obr. 8 Chybné prihlasovacie údaje

6 Požiadavky

6.1 Požiadavky na technické prostriedky

1. Optimálne požiadavky pre mobilné zariadenie:

- Zariadenie s operačným systémom Android 4.0 Ice Cream Sandwich
- Potrebný priestor pre aplikáciu - 7MB

2. Optimálne požiadavky pre osobný počítač:

- CPU- Intel (R) Core™ i5-2430M CPU @ 2,40 GHz
- RAM- 4,00 GB
- Potrebný priestor pre vývoj a samotnú aplikáciu - 3 GB

6.2 Požiadavky na programové prostriedky pri preklade

Operačný systém:

- Windows XP 32 bitový,
- Windows Vista 32/ 64 bitový,
- Windows 7 32/64 bitový.

Vývojové prostriedky:

- Android vývojové nástroje (Build: v21.0.1-543035):
 - Eclipse Platform, JDT, CDT, EMF, GEF and WTP s doplnkom pre vývoj aplikácií pre Adnroid – ADT plugin
 - Android SDK manager, Revision 21.0.1
 - SQLite3
 - Java SE JDK 7
- XAMPP
 - Apache server
 - MySQL
 - PHP
- Microsoft Office Visio 2007

7 Zhrnutie riešenia

Vývoj aplikácií pre systém Android je jednoduchý a pomocou emulátora je možné si programy hneď vyskúšať.

Databázová časť aplikácie je interpretovaná vzdialeným serverom MYSQL, ktorý som vytvorila pomocou balíčka programov XAMPP, kde sa nachádza webový server APACHE, databáza MYSQL a php. Vďaka webovému serveru je možné sa spojiť s databázou a tak interpretovať vzdialený prístup. Toto spojenie som si zvolila preto, lebo síce existujú aj voľné databázové serveri, ale kvôli ich nadmernej vyťažnosti by spojenie trvalo dlho a aplikácia by neustále padala. Takto to funguje na rovnakom princípe, len databáza je uložená na disku notebooku ako localhost. Na tento localhost sa aplikácia musí pripojiť klasicky pomocou údajov a adresy.

Aplikácia je vytvorená tak, aby ju bolo možné ďalej rozširovať o nové funkcie podľa požiadaviek lekárov. Napríklad je možné dorobiť kalendár, ktorý by slúžil na zdieľanie voľných termínov u lekára, ktoré by si sám navolil. Pacienti by sa takto mohli pohodlne prihlasovať a taktiež by mohli dať vedieť lekárovi, keby sa nemohli dostať. A tiež, ak by lekár musel zrušiť niektorých objednaných pacientov, vedel by im to oznámiť cez systém a navrhnúť nové termíny.

Programy v medicínskej oblasti majú množstvo výhod. Dostupnosť niektorých informácií na internete je pre pacientov lákavá, ale takéto zaujímavé systémy sa môžu stať lákavé aj pre páchanie počítačovej kriminality.

Možnosti vývoja softvérových produktov pre oblasť medicíny je veľmi široká. Uplatnenie nemusí byť až také náročné ak sú produkty spracované kvalitne a podávajú vecné a presné informácie.

Zoznam použitej literatúry

- [1] Mark L.Murphy: Android 2 – Průvodce programováním mobilních aplikací. Brno: Computer Press, a.s., 2011. 377 s. ISBN 978-80-251-31
- [2] About the Eclipse Foundation [online]. [cit.2013-04-04] Dostupné na internete: <<http://www.eclipse.org/org/>>
- [3] Apache friends – XAMPP for Windows [online]. [cit.2013-04-04] Dostupné na internete <<http://www.apachefriends.org/en/xampp-windows.html>>

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Android v medicínskych aplikáciách
Diplomová práca
Metodická príručka

Študijný program: Informatika
Študijný odbor: 9.2.1 Informatika
Školiace pracovisko: Katedra počítačov a informatiky (KPI)
Školiteľ: Doc. Ing. Zdeněk Havlice, CSc.
Konzultant: Ing. Juraj Vízi

Obsah

Zoznam obrázkov	3
Úvod	4
1 Čo je potrebné pred vytvorením projektu?	5
1.1 Ako vytvoriť aplikáciu	13
2 Ako prepojiť aplikáciu s databázovou časťou	20
3 Ako si aplikáciu preniesť na vlastné zariadenie.....	24
Zoznam použitej literatúry	28

Zoznam obrázkov

Obr. 1 Ponuka vývojových prostriedkov Eclipse	5
Obr. 2 Verzie SDK manažéra pre rôzne systémy	6
Obr. 3 Súhlas s podmienkami používania	6
Obr. 4 Úvod inštalácie	7
Obr. 5 Overenie JAVA SE JDK	7
Obr. 6 Výber používateľov tohto nástroja	8
Obr. 7 Nastavenie adresy kde sa má SDK na inštalovať	8
Obr. 8 Začiatkové menu	9
Obr. 9 Inštalácia Android SDK nástroja	9
Obr. 10 Ukončenie inštalácie	10
Obr. 11 Spustenie Android SDK manažéra	10
Obr. 12 AVD manažér	11
Obr. 13 Nastavenie nového zariadenia	12
Obr. 14: Nastavenie adresára SDK	13
Obr. 15 Vytvorenie Android projektu	14
Obr. 16 Ponuka pre Android projekty	14
Obr. 17 Názov aplikácie, projektu a balíčka	15
Obr. 18 Nastavenie ikonky aplikácie a vytvorenie hlavnej aktivity	16
Obr. 19 Ikonka aplikácie	16
Obr. 20 Vytvorenie aktivity	17
Obr. 21 Nastavenie hlavnej aktivity	18
Obr. 22 štruktúra aplikácie	19
Obr. 23 Oficiálna stránka na stiahnutie XAMPP pre rôzne paltformy	21
Obr. 24 Program XAMPP	21
Obr. 25 XAMPP v internetovom prehliadači	22
Obr. 26 Nástroj phpMyAdmin pre vytváranie MySQL databázy	22
Obr. 27 Doinštalovanie doplnku Google USB Driver pomocou SDK manažéra	24
Obr. 28 Vývojárske nastavenia v OS Android	26
Obr. 29 Režim ladenia s USB zariadením	27

Úvod

V metodickéj príručke je popísané ako postupovať pri vytváraní aplikácie, ktorá je určená pre operačný systém Android. Operačný systém Android je využívaný v mobilných zariadeniach. Ide o tablety a Smartphony. Smartphony sú multifunkčné zariadenia, ktoré sa funkciami už skoro vyrovnajú normálnemu počítaču.

Vývoj aplikácií pre platformu Android je v súčasnosti veľmi populárny. Je to možné sledovať aj na denne pribúdajúcich aplikáciách v internetovom obchode Google Play. Ide o virtuálny obchod, z ktorého je možné sťahovať aplikácie nielen zadarmo, ale je tam aj možnosť ich zakúpenia.

Samotný vývoj aplikácie pre tento operačný systém nie je veľmi náročný. Poskytovatelia Androidu sa snažia uľahčiť nielen prácu bežných používateľov, ale aj vývojárov. Vzhľadom na to, že Android je voľne dostupná platforma, tak aj preferované prostriedky na vývoj Android aplikácií sú voľne dostupné.

V ďalších častiach sa budem venovať týmto témam:

- čo je nevyhnutné pre vývoj aplikácií bežiacich na Androide,
- ako s vývojom začať
- ako prepojiť aplikačnú časť s databázovou
- ako si aplikáciu preniesť na vlastné zariadenie

1 Čo je potrebné pred vytvorením projektu?

Pred samotným vytvorením aplikácie je potrebné mať nainštalovanú Javu SE JDK-7. Potom nasleduje inštalácia a konfigurácia vývojárskych nástrojov. Existujú dve spôsoby. Prvý spôsob, ľahší je taký, že z hlavnej stránky pre Android vývojárov je možné si stiahnuť ADT bundle [1]. Ide o balíček vývojárskych nástrojov, ktoré sú voľne dostupné a prispôbené na vývoj Android aplikácií. ADT bundle obsahuje vývojové prostredie Eclipse už s nainštalovaným doplnkom pre Android a tiež obsahuje SDK manažér, ktorý slúži ako emulátor. Emulátor vie simulovať zariadenia s operačným systémom Android. Nastaviť je potrebné, len SDK manažér.

Druhý spôsob, starší je taký, že najprv si treba stiahnuť Eclipse z hlavnej stránky [3]. Postačí aj verzia klasik, ako je vidno na obrázku (Obr. 1)



Obr. 1 Ponuka vývojových prostriedkov Eclipse

Eclipse nemá žiadnu inštaláciu. Stačí si ho stiahnuť a otvoriť exe súbor.

Ďalej je potrebné stiahnuť SDK manažér, ktorého inštalácia potrebuje Javu SE JDK-7. Z oficiálnej stránky pre Android vývojárov je možné si stiahnuť osobitne aj SDK manažér [1], ako je vidno aj na obrázku (Obr. 2). Je vyvinutý pre rôzne

systemy.

SDK Tools Only			
Platform	Package	Size	MD5 Checksum
Windows 32 & 64-bit	android-sdk_r21.1-windows.zip	99360755 bytes	dbece8859da9b66a1e8e7cd47b1e647e
	installer_r21.1-windows.exe (Recommended)	77767013 bytes	594d8ff8e349db9e783a5f2229561353
Mac OS X 32 & 64-bit	android-sdk_r21.1-macosx.zip	66077080 bytes	49903cf79e1f8e3fde54a95bd3666385
Linux 32 & 64-bit	android-sdk_r21.1-linux.tgz	91617112 bytes	3369a439240cf3dbe165d6b4173900a8

Obr. 2 Verzie SDK manažera pre rôzne systémy

Po zvolení verzie sa zobrazí okno s podmienkami, ktoré je potrebné potvrdiť, aby bolo možné stiahnuť manažér (Obr. 3).

Get the Android SDK

Before installing the Android SDK, you must agree to the following terms and conditions.

Terms and Conditions

This is the Android Software Development Kit License Agreement

1. Introduction

1.1 The Android Software Development Kit (referred to in this License Agreement as the "SDK" and specifically including the Android system files, packaged APIs, and Google APIs add-ons) is licensed to you subject to the terms of this License Agreement. This License Agreement forms a legally binding contract between you and Google in relation to your use of the SDK.

1.2 "Android" means the Android software stack for devices, as made available under the Android Open Source Project, which is located at the following URL: <http://source.android.com/>, as updated from time to time.

1.3 "Google" means Google Inc., a Delaware corporation with principal place of business at 1600 Amphitheatre Parkway, Mountain View, CA 94043, United States.

I have read and agree with the above terms and conditions

[Download installer_r21.1-windows.exe](#)

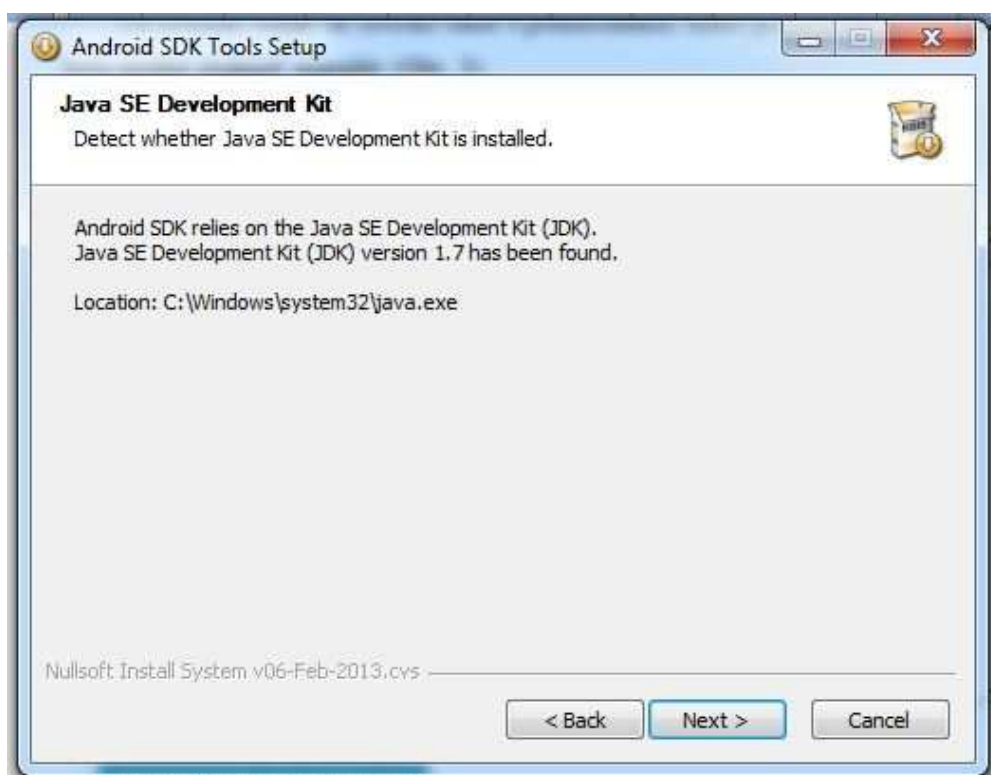
Obr. 3 Súhlas s podmienkami používania

Nasleduje inštalácia SDK manažera. Ktorá je riadená typickým sprevádzaním wizard, kde sa odporúča vypnúť iné bežiacie aplikácie počas tejto inštalácie (Obr. 4).



Obr. 4 Úvod inštalácie

Nasleduje overenie či je na danom zariadení nainštalovaná verzia JAVA SE Development Kit, potrebná na zobrazenie emulátora (Obr. 5).



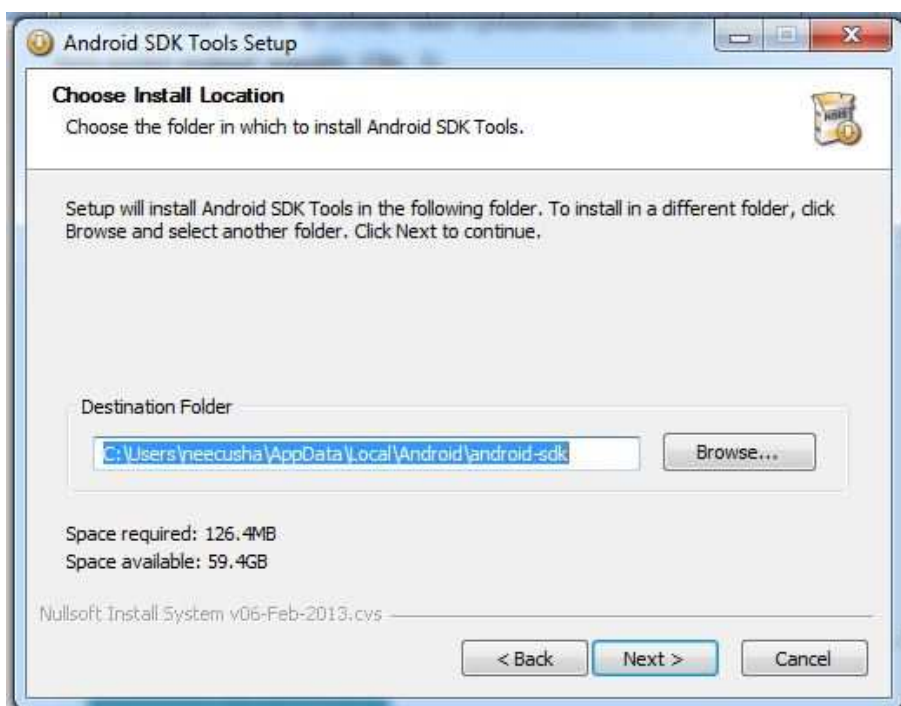
Obr. 5 Overenie JAVA SE JDK

Ďalším krokom je potrebné nastaviť pre koho bude tento nástroj prístupný (Obr. 6).



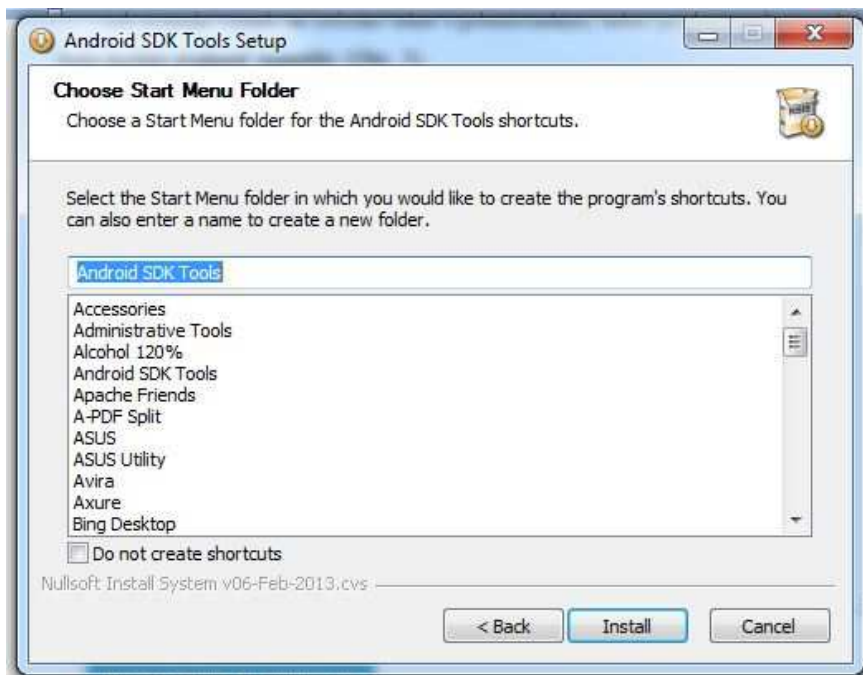
Obr. 6 Výber používateľov tohto nástroja

Ďalej je potrebné zvoliť kam bude SDK manažér na inštalovaný (Obr. 7).



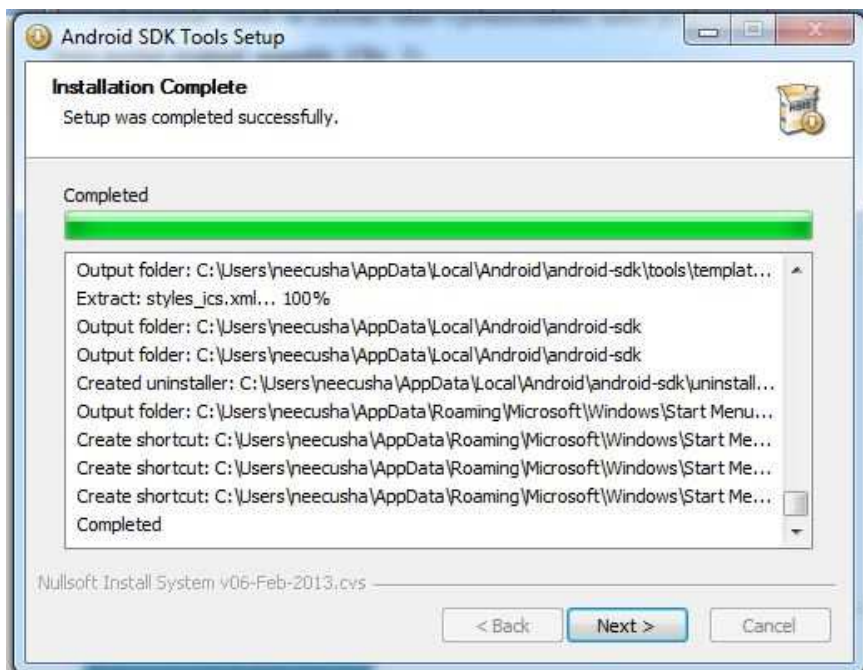
Obr. 7 Nastavenie adresy kde sa má SDK na inštalovať

Nastavenie začiatočného menu (Obr. 8).



Obr. 8 Začiatočné menu

Po tejto obrazovke by sa mala spustiť už inštalácia (Obr. 9).



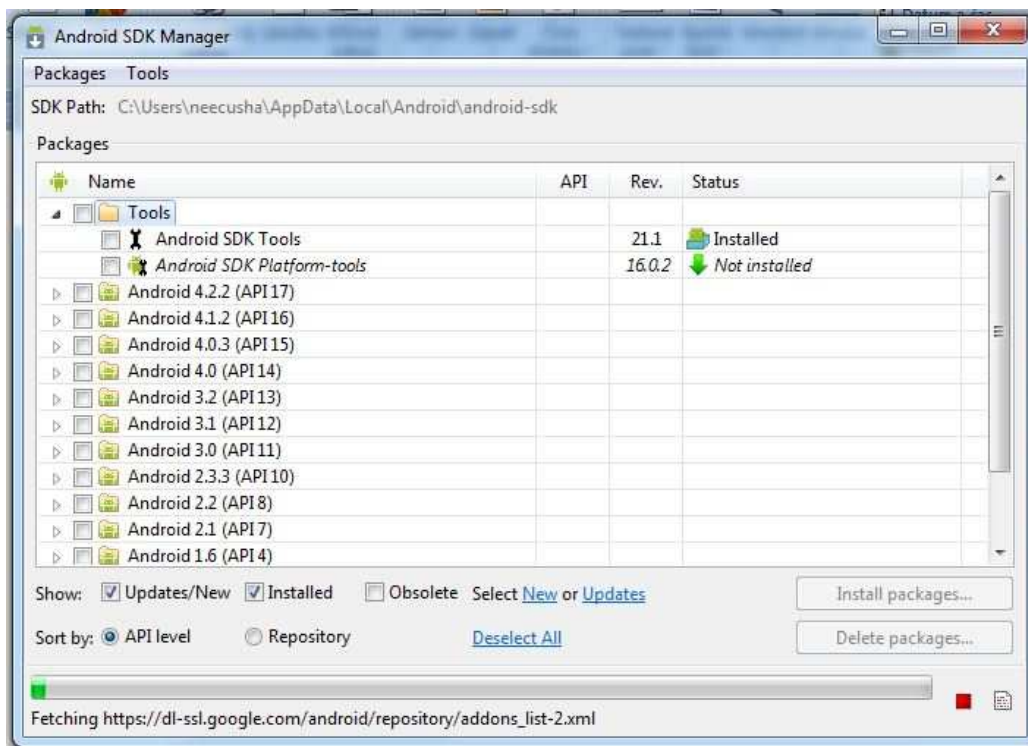
Obr. 9 Inštalácia Android SDK nástroja

Po úspešnej inštalácii by sa mala zobrazit' obrazovka na Obr. 10, kde je možné si zapnúť hneď SDK manažér.



Obr. 10 Ukončenie inštalácie

Na Obr. 11 je vidieť už spustený SDK manažér. Tu je možné si stiahnuť potrebné API na vývoj aplikácií pre príslušné verzie Androidu.



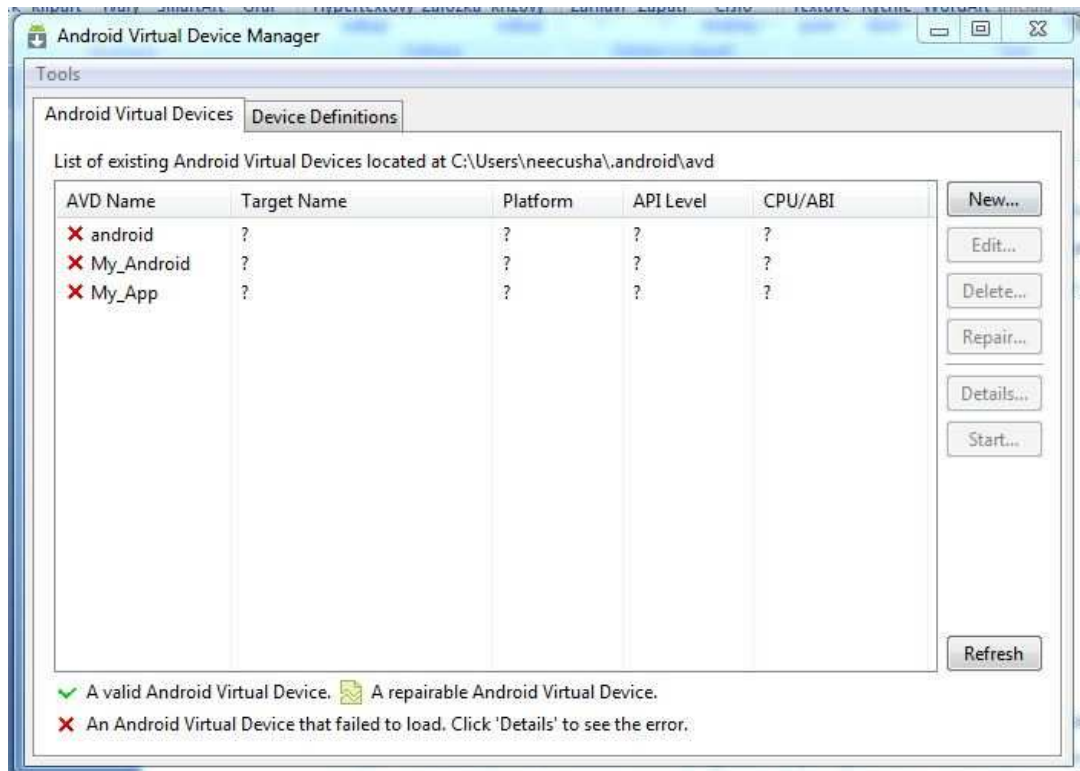
Obr. 11 Spustenie Android SDK manažéra

Každá verzia Androidu využíva iné API (Tab. 1). To znamená aj to, že aplikácie, ktoré boli vytvorené pre staršie verzie Androidu je možné spustiť na nových verziách. Ale aplikácie vyvinuté na nové verzie napríklad pre 4.0 Ice Cream Sandwich, nie je možné spustiť na zariadení, ktoré má implementovanú verziu 2.3 –Gingerbread. [2]

VERSION	CODENAME	API
1.6	Donut	4
2.1	Eclair	7
2.2	Froyo	8
2.3 - 2.3.2	Gingerbread	9
2.3.3 - 2.3.7		10
3.2	Honeycomb	13
4.0.3 -4.0.4	Ice Cream Sandwich	15
4.1.x	Jelly Bean	16
4.2.x		17

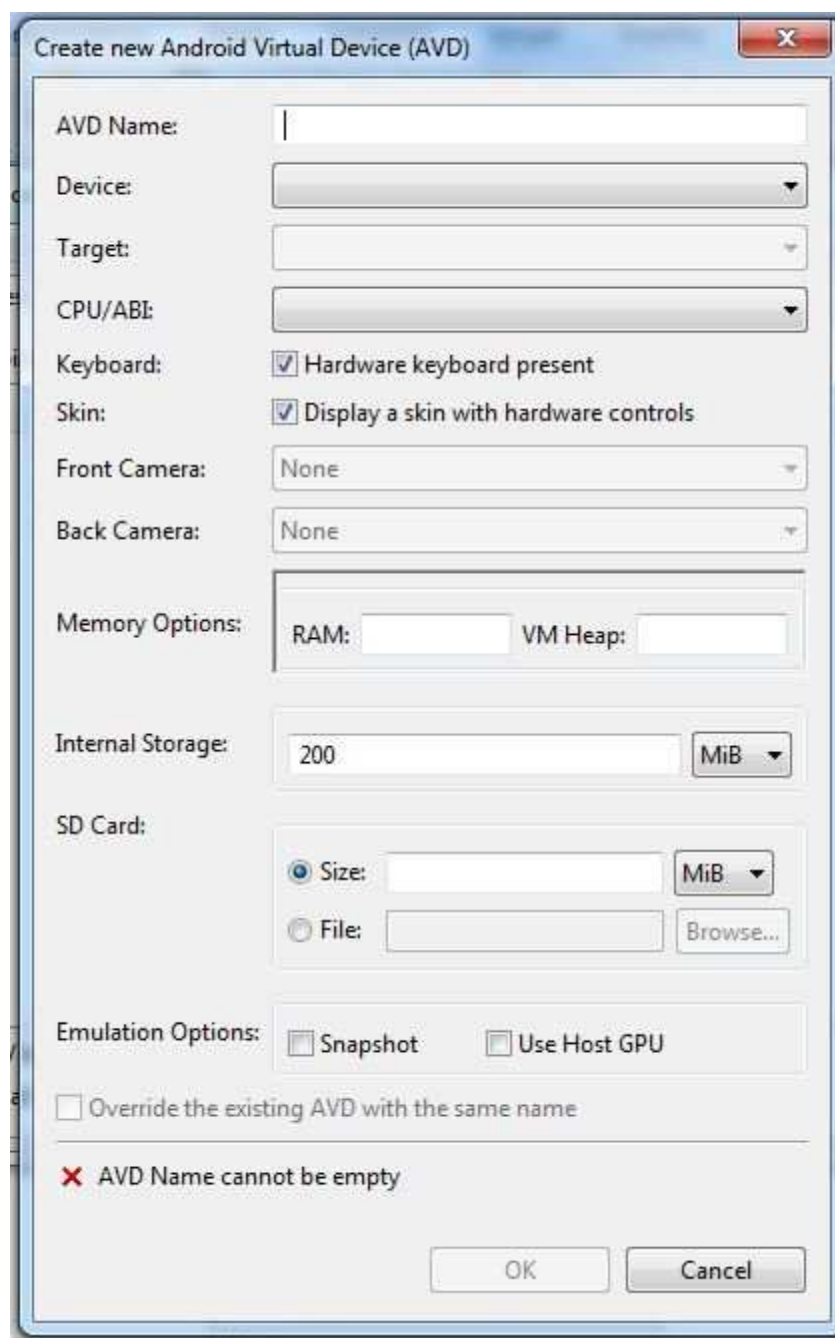
Tab. 1 Prehľad verzií OS Android a k nim príslušné API [2]

Teraz je potrebné už len vytvoriť emulátor zariadenia, pre ktoré chceme aplikáciu vyvíjať. Výhodou je, že je možné robiť programy aj pre úplne nové verzie aj keď nie je k dispozícii reálne zariadenie s touto verziou. Spolu s SDK manažérom sa nainštaloval aj AVD manažér (Obr. 12), kde je potrebné nastaviť emulátor.



Obr. 12 AVD manažér

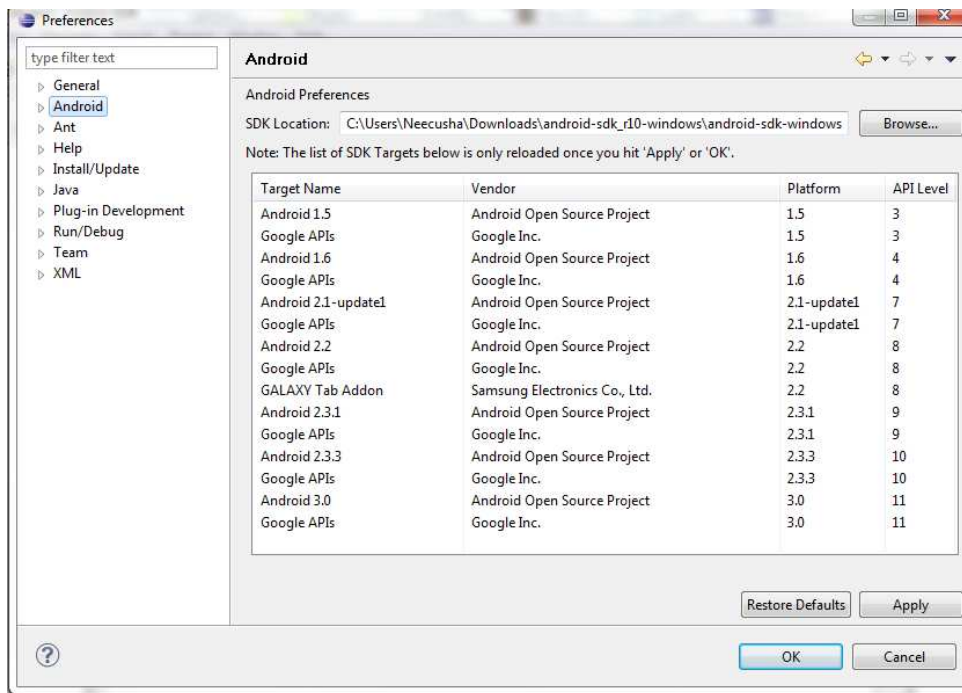
Pre vytvorenie nového virtuálneho zariadenia, je potrebné kliknúť na tlačidlo New, zobrazí sa obrazovka ako na Obr. 13.



Obr. 13 Nastavenie nového zariadenia

V tomto okne je potrebné zadať názov emulátora, vybrať verziu Androidu pre ktorú bude aplikácia. Dôležité je nastaviť veľkosť SD karty, ktorá je odporúčaná 512-1024 MB.

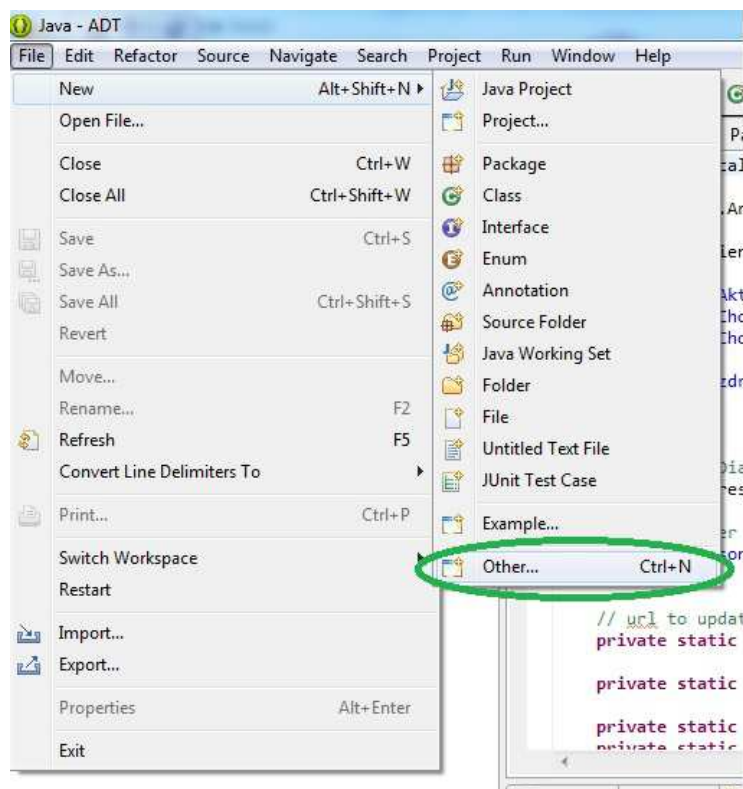
Na vývoj aplikácií pre Android je potrebné pripojiť doplnok Android Development Tools k Eclipse. Po spustení Eclipse, v hlavnom menu Help > Install New Software sa zobrazí okno v ktorom je potrebné zadať ADT plugin - <http://developer.android.com/sdk/eclipse-adt.html>. Po pridaní týchto prvkov je nutné prepojiť emulátor s prostredím Eclipse. V hlavnom menu Window > Preferences, otvorí sa okno v ktorom je potrebné sa prepnúť do položky Android, v ktorej je potrebné nastaviť cestu do adresára, kde je nainštalovaný SDK manažér.



Obr. 14: Nastavenie adresára SDK

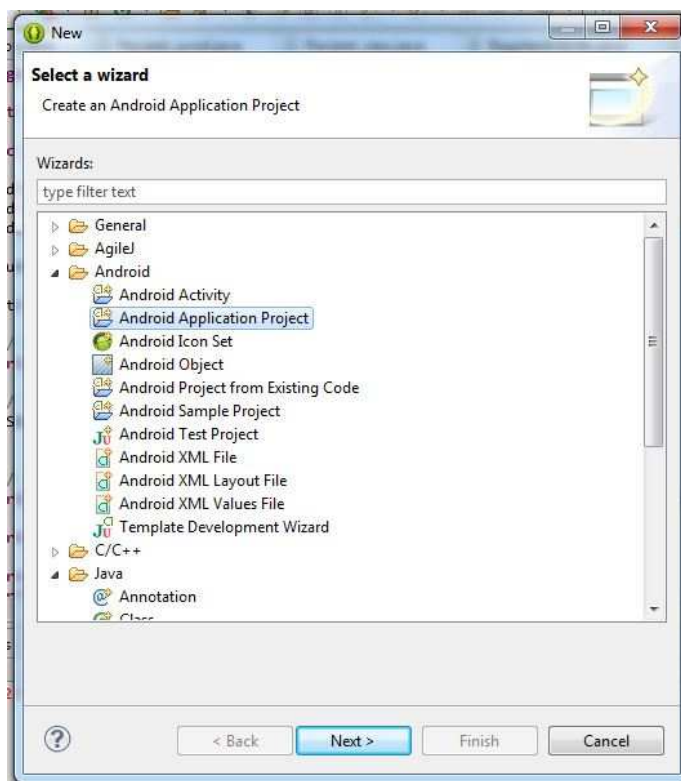
1.1 Ako vytvoriť aplikáciu

Po nainštalovaní všetkých potrebných nástrojov na vývoj aplikácií je možné začať s vytvorením samotného projektu. Vo vývojom prostredí Eclipse v hlavnom menu je prvá položka Súbor (angl. File) po kliknutí na ňu sa objaví ponuka tak ako je vidno na Obr. 15. Android projekty sa nezobrazujú v základnej ponuke pre vytvorenie projektov. Je potrebné kliknúť na zobrazenie ďalšej ponuky.



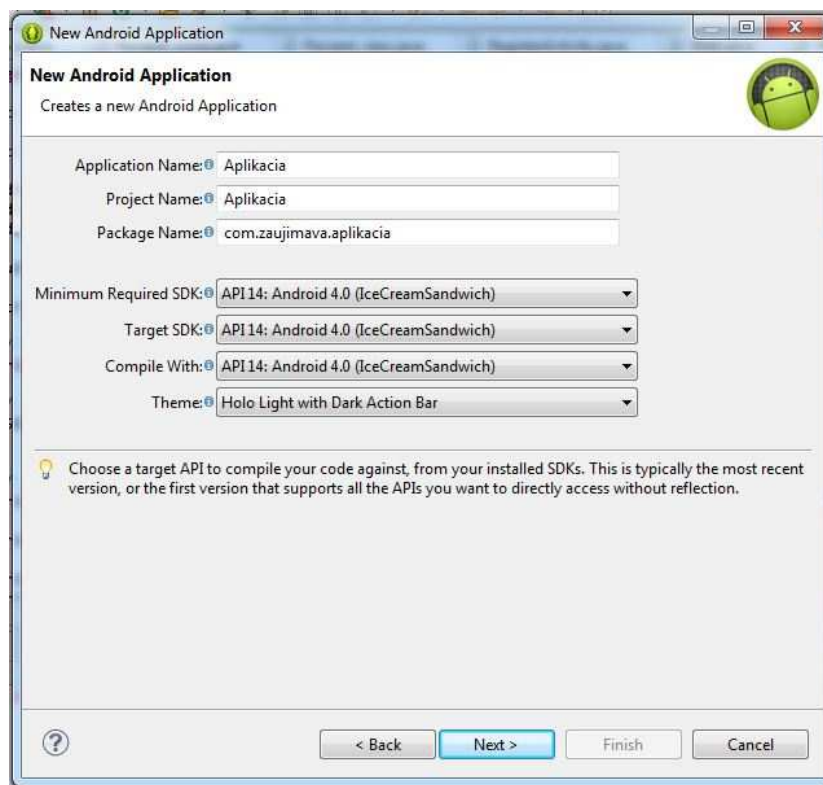
Obr. 15 Vytvorenie Android projektu

V ďalšom menu sa zobrazia zložky pre vytváranie projektov, ktoré boli napríklad doinštalované ako doplnky do vývojového prostredia Eclipse (Obr. 16).



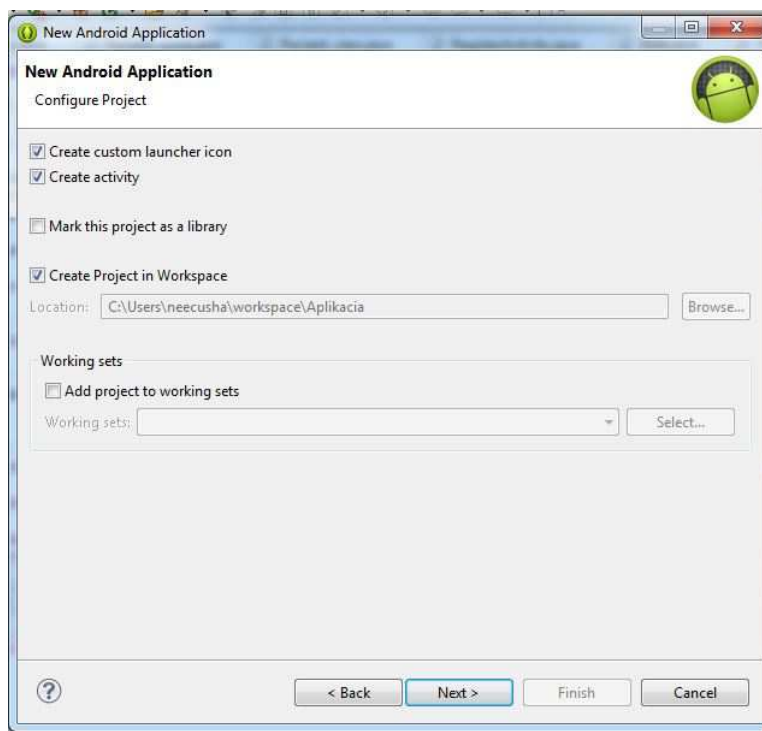
Obr. 16 Ponuka pre Android projekty

Po výbere niektorej z ponuky Android, napríklad Android Application Project sa zobrazí obrazovka, kde sa zadáva meno aplikácie, projektu a balíčka (Obr. 17). Meno balíčka je hneď nastavené na com.example. za slovo example sa dopíše zadaný názov pre aplikáciu a v okne bude stále chyba. Je potrebné zmeniť example na nejaký iný vhodný názov. V tomto okne je tiež potrebné vedieť na akom zariadení chceme aplikáciu skúšať. Nastaví sa minimálne požadované API podľa ktorého sa bude aplikácia vyvíjať.



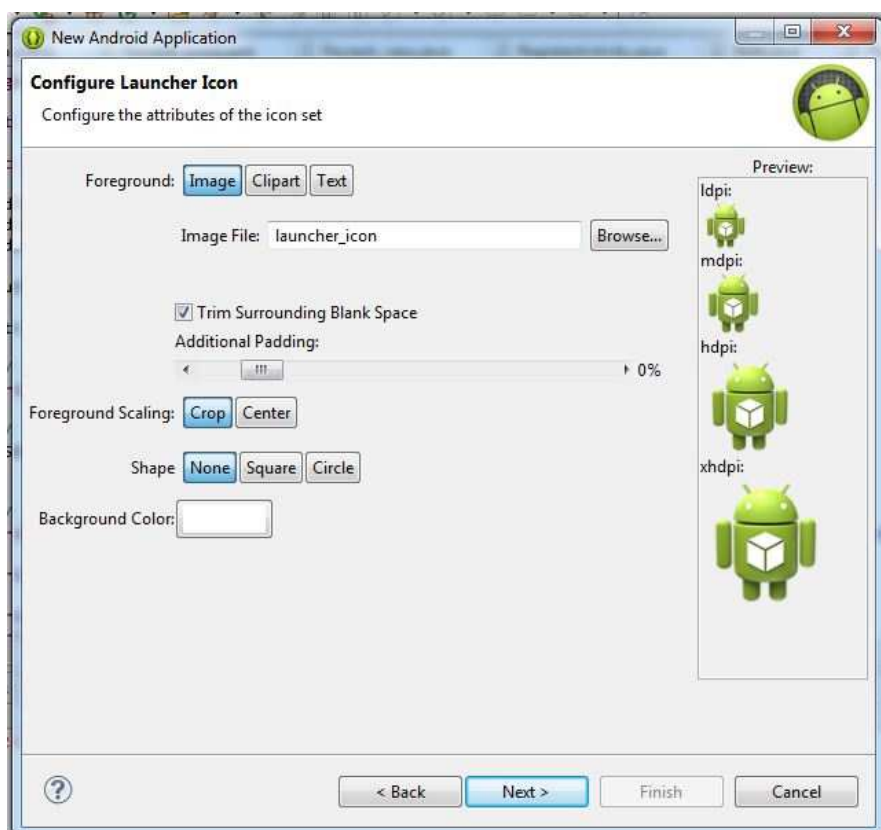
Obr. 17 Názov aplikácie, projektu a balíčka

Na ďalšej obrazovke sa nastavuje či chcem nastaviť ikonku pre aplikáciu a či je potrebné vytvoriť hneď aj hlavnú triedu (Obr. 18). Tu je možné zvoliť si aj inú adresu umiestnenia projektu.



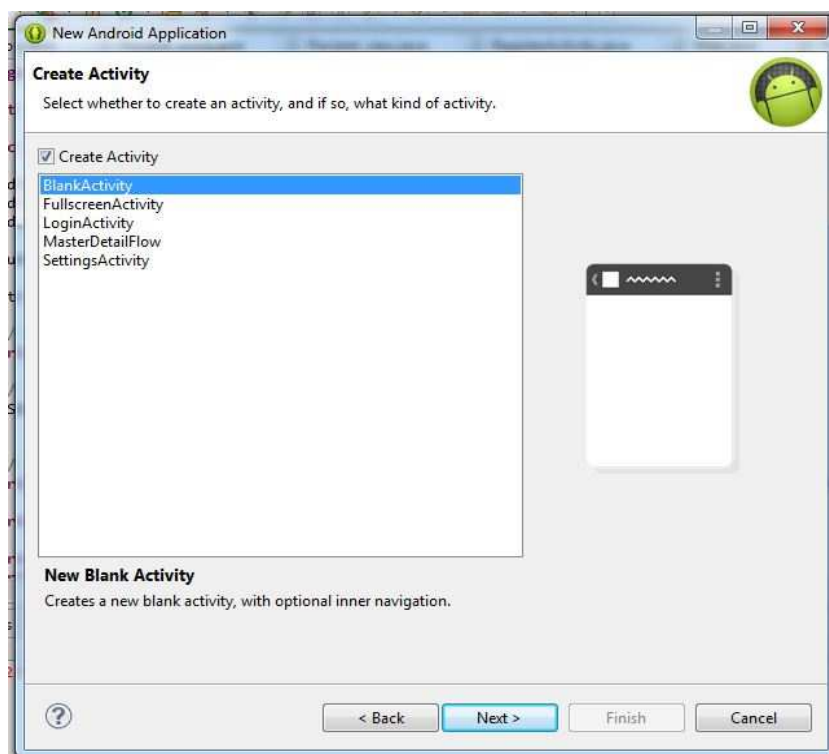
Obr. 18 Nastavenie ikonky aplikácie a vytvorenie hlavnej aktivity

Ďalšia obrazovka je zameraná len na ikonku pre aplikáciu- typ, farba, veľkosť (Obr. 19)



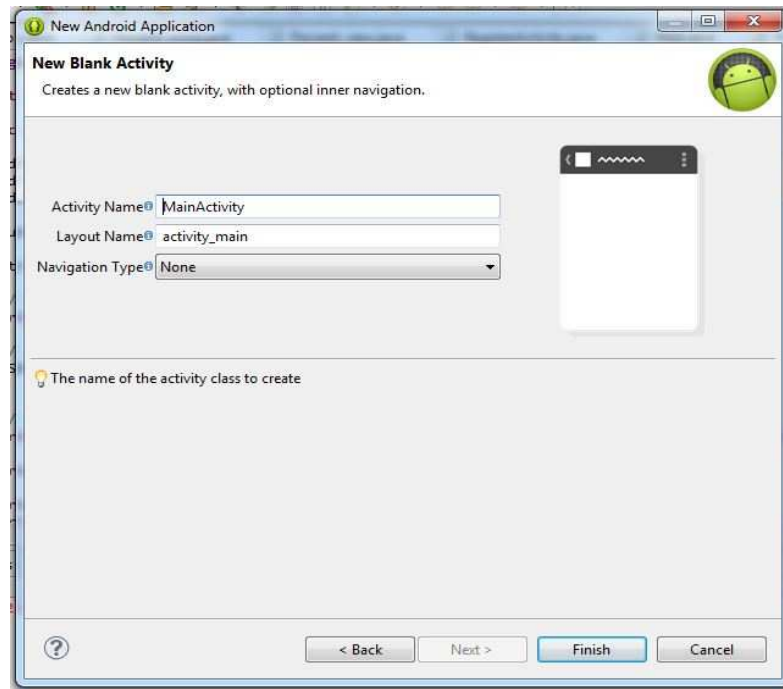
Obr. 19 Ikonka aplikácie

Po nastavení ikony nasleduje nastavenie aktivity aplikácie. Môže to byť napríklad prihlasovacia aktivita, nastavovacia aktivita alebo aj prázdna, ktorú je možné dotvoriť(Obr. 20).



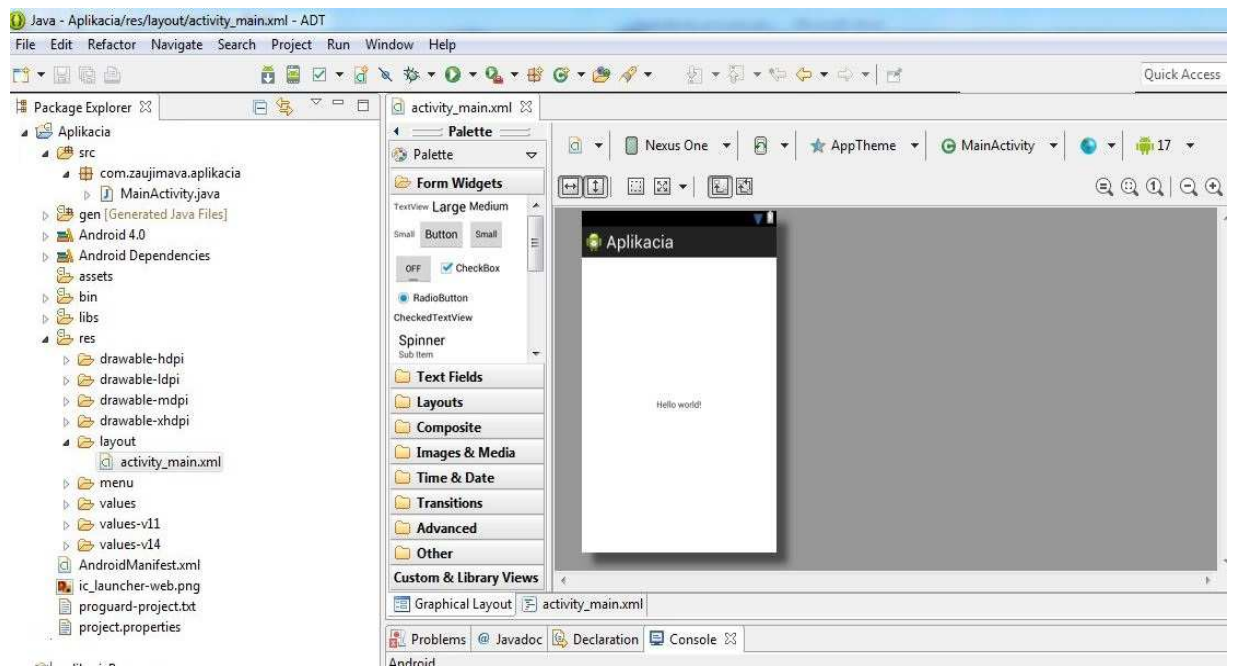
Obr. 20 Vytvorenie aktivity

Ak sme si zvolili typ aktivity, ďalej je potrebné nastaviť jej názov a layout, na ktorý sa bude aktivita odkazovať(Obr. 21). To znamená že po spustení aplikácie sa nám zobrazí hlavná aktivita tak že zobrazí obrazovku, ktorú sme teraz nastavili.



Obr. 21 Nastavenie hlavnej aktivity

Ak sme všetko nastavili a dostali sa na koniec tak sa zobrazí štruktúra nového Android projektu na ľavej lište. Môžeme tam vidieť aj hlavnú triedu a layout, ktorý sme pre ňu nastavili. Na väčšej ploche obrazovky sa nachádza pracovný priestor, kde sa zobrazí automaticky po vytvorení projektu grafické používateľské rozhranie projektu. Používateľské rozhranie Android projektov prebieha pomocou špeciálneho editora, v ktorom sa dá veľmi ľahko a rýchlo navrhnuť vzhľad obrazoviek aplikácie. Obsahuje základné prvky na panely nástrojov, ktoré je možné veľmi rýchlo popresúvať na obrazovku zariadenia. Celé grafické používateľské rozhranie je robené v jazyku XML a je možné sa prepnúť priamo aj na kód, ktorý je tiež možné automaticky upraviť na prehľadnejšie zobrazenie.



Obr. 22 štruktúra aplikácie

2 Ako prepojiť aplikáciu s databázovou časťou

Dáta potrebné pre aplikáciu môžu byť uložené napríklad v jednoduchých súboroch, v lokálnej databáze SQLite, alebo na vzdialenom serveri, ku ktorému sa pristupuje prostredníctvom internetu. SQLite je lokálna databáza v systéme Android, ktorá je zabudovaná vo vykonávacom prostredí. A práve to umožňuje každej aplikácii vytvoriť vlastnú databázu. Práca s ňou je veľmi jednoduchá pre ľudí, ktorí už majú nejaké skúsenosti s inými rozhraniami SQL. Patrí medzi najrozšírenejšie, pretože obsahuje čisté SQL rozhranie, zaberá menej miesta ako iné databázy (275 kB). SQLite používa pre dotazy, manipuláciu s dátami a definíciu dát dialekt jazyka SQL. Je efektívna pri práci s pamäťou, pretože ju môže vykonávacie prostredie systému obsahovať celú a nie len jednu časť, aby sa znížil jej celkový objem. Veľký rozdiel medzi SQLite a inými databázami je v typovaní dát. Do akéhokoľvek stĺpca je možné uložiť dáta ľubovoľnej hodnoty [4]. SQLite nepodporuje niekoľko štandardných vlastností SQL:

- obmedzenie typu použitím cudzieho kľúča (angl. FOREIGN KEY),
- vnorená transakcia,
- čiastočné vonkajšie spojenie typu sprava (angl. RIGHT OUTER JOIN)
- úplné vonkajšie spojenie (angl. FULL OUTER JOIN),
- niektoré príkazy vykonávajúce modifikáciu tabuľky (angl. ALTER TABLE)

Pre aplikácie, v ktorých majú prístup viacerí užívatelia je lepšie mať dáta uložené na vzdialenom serveri. Po spustení aplikácie sa načítajú aktuálne údaje z databázy.

Na interpretáciu vzdialeného prístupu s databázou som použila program XAMPP [5]. Je to program, ktorý je vyvíjaný skupinou Apache Friends. Ide o program, ktorý obsahuje Apache, MySQL, PHP a Perl (Obr. 24). Je voľne dostupný a nie je potrebná žiadna inštalácia. Stačí stiahnuť verziu pre daný systém, rozbaľiť a spustiť.

XAMPP

Many people know from their own experience that it's not easy to install an Apache web server and it gets harder if you want to add MySQL, PHP and Perl.

XAMPP is an easy to install Apache distribution containing MySQL, PHP and Perl. XAMPP is really very easy to install and to use - just download, extract and start.

At the moment there are four XAMPP distributions:



XAMPP for Linux [↗](#)

The distribution for Linux systems (tested for SuSE, RedHat, Mandrake and Debian) contains: Apache, MySQL, PHP & PEAR, Perl, ProFTPD, phpMyAdmin, OpenSSL, GD, Freetype2, libjpeg, libpng, gdbm, zlib, expat, Sablotron, libxml, Ming, Webalizer, pdf class, ncurses, mod_perl, FreeTDS, gettext, mcrypt, mhash, eAccelerator, SQLite and IMAP C-Client.

XAMPP for Windows [↗](#)

The distribution for Windows 2000, 2003, XP, Vista, and 7. This version contains: Apache, MySQL, PHP + PEAR, Perl, mod_php, mod_perl, mod_ssl, OpenSSL, phpMyAdmin, Webalizer, Mercury Mail Transport System for Win32 and NetWare Systems v3.32, Ming, FileZilla FTP Server, mcrypt, eAccelerator, SQLite, and WEB-DAV + mod_auth_mysql.

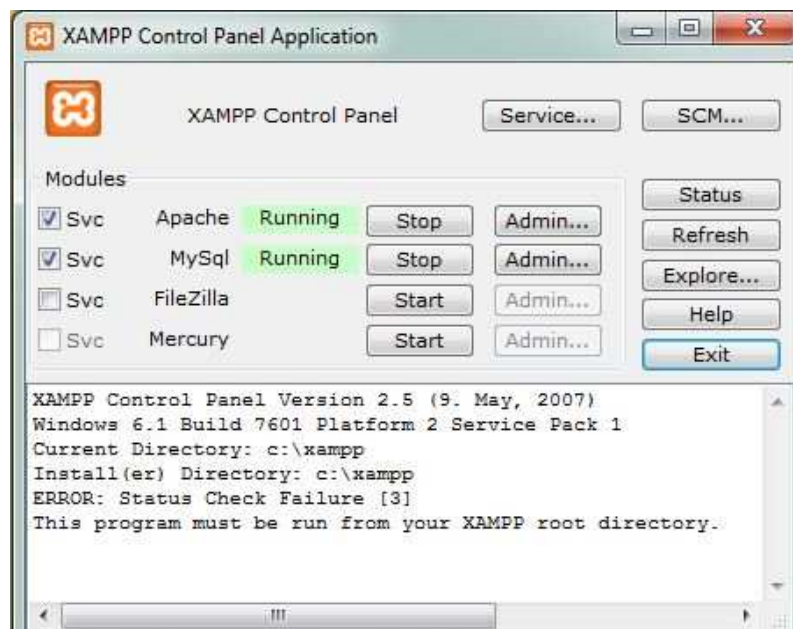
XAMPP for Mac OS X [↗](#)

The distribution for Mac OS X contains: Apache, MySQL, PHP & PEAR, SQLite, Perl, ProFTPD, phpMyAdmin, OpenSSL, GD, Freetype2, libjpeg, libpng, zlib, Ming, Webalizer, mod_perl.

XAMPP for Solaris [↗](#)

Obr. 23 Oficiálna stránka na stiahnutie XAMPP pre rôzne platformy

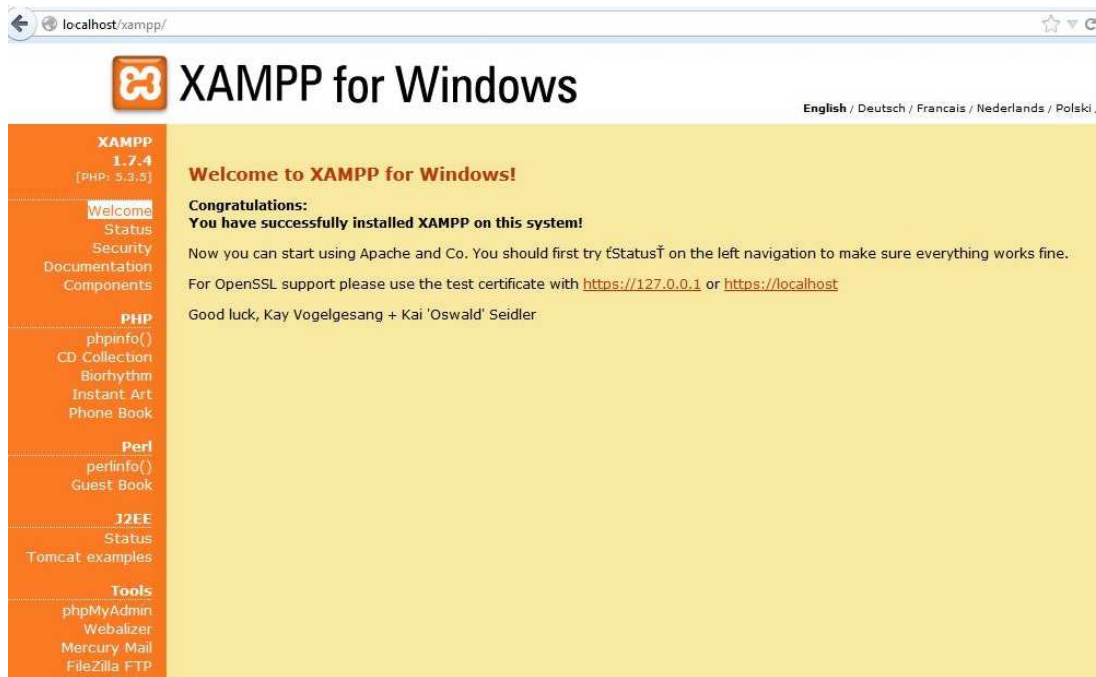
Apache je program, ktorý predstavuje webový server. Apache je taktiež voľne dostupný.



Obr. 24 Program XAMPP

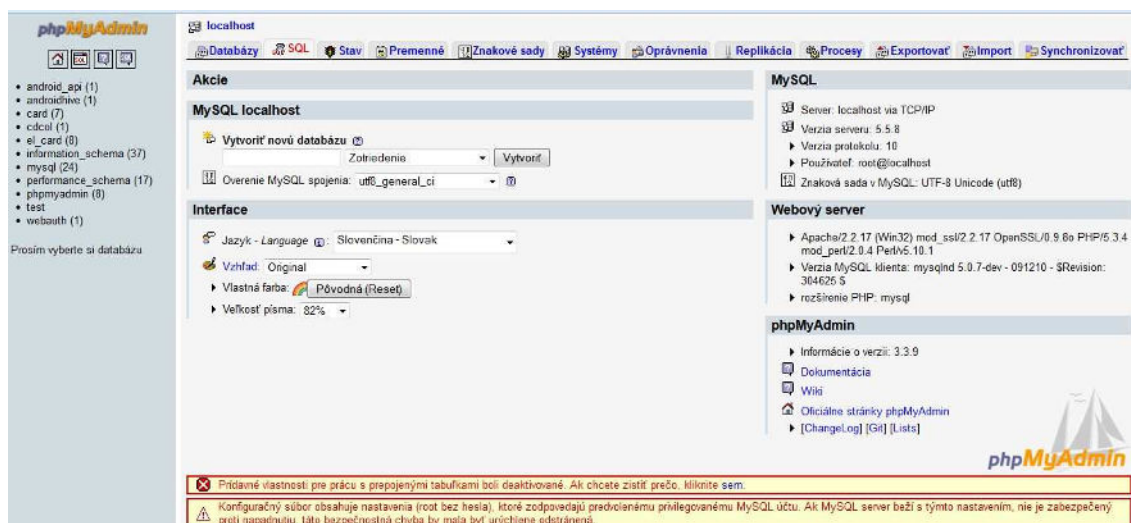
Apache spolu s PHP a MySQL sú veľmi často využívané k vývoju dynamických internetových stránok. MySQL je databázový server. Ďalšie funkcie pre XAMPP je

možné prezrieť aj v ľubovoľnom internetovom prehliadači, kde stačí napísať localhost a mala by sa zobrazíť stránka ako na Obr. 25.



Obr. 25 XAMPP v internetovom prehliadači

Údaje mojej aplikácie sa ukladajú do MySQL databázy pomocou nástroja phpMyAdmin. To je nástroj v internetovom prehliadači, ktorý je prispôsobený na vytváranie databázových skriptov (Obr. 26). Komunikácia databázy a mojej aplikácie prebieha pomocou skriptov v PHP programovacom jazyku. [5]



Obr. 26 Nástroj phpMyAdmin pre vytváranie MySQL databázy

Komunikácia s webovými službami prebieha pomocou protokolu SOAP (Simple Object Access Protocol). Je to protokol, ktorý zabezpečuje výmenu správ založených na XML prostredníctvom siete. SOAP je nástupcom XML-RPC protokolu. XML-RPC

uchováva pravidlá, ktoré definujú ako sa používajú štandardné technológie pre potreby RPC (vzdialené volanie procedúr). V systéme Android sa nenachádzajú API protokolu SOAP a ani protokolu XML-RPC. Táto komunikácia je v Androide vyriešená tak, že súčasťou systému Android je knižnica Apache HttpComponents, nad ktorou je možné umiestniť vrstvu protokolov SOAP alebo XML-RPC, alebo je možné používať prístup k webovým službám založeným na architektúre REST (Representational State Transfer). Rest je architektúra, ktorá umožňuje pristupovať k dátam na určitom mieste pomocou štandardných metód http. Definuje ako majú byť špecifikované a adresované. Využíva štandardy :

- prenosný protokol (HTTP)
- zdrojový protokol (URL)
- zdrojová reprezentácia (XML/HTML/GIF/JPEG/...)
- MIME typy (text/xml, text/html, image/gif, image/jpeg,...)

Prístup k zdrojom je jednotný a jednoduchý. REST umožňuje využitie CRUD operácií pomocou štandardných HTTP dotazov. REST definuje 4 základné metódy, ktoré sú v tabuľke (viď Tab. 2).

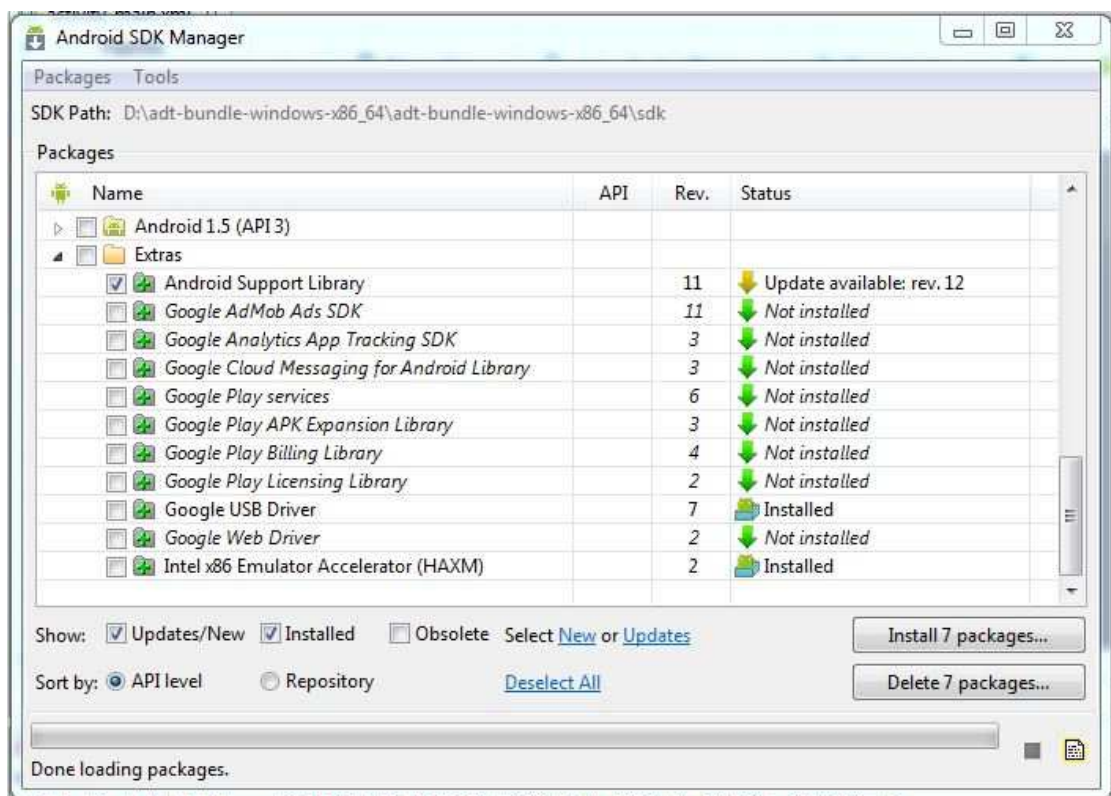
REST	CRUD
POST	Create
GET	Read
PUT	Update or Create
DELETE	Delete

Tab. 2 Metódy v architektúre REST

Moja aplikácia interpretuje pripojenie na vzdialený server. Využívam v nej architektúru REST a knižnicu Apache HttpComponents. Najprv bolo potrebné si vytvoriť objekt typu `httpClient`. Vzhľadom na to že ide o rozhranie, bude sa musieť vytvoriť inštancia implementácie tohto rozhrania, napríklad trieda `DefaultHttpClient`. Do inštancie triedy `HttpRequest` sa zabalia požiadavky s rôznymi implementáciami pre jednotlivé http príkazy (napr. `get`). Následne je potrebné vytvoriť inštanciu triedy `HttpRequest` s danou implementáciou. Do nej sa vloží cieľová URL adresa a ďalšie konfiguračné údaje. Potom sa metóda predá klientovi, aby sa http požiadavka vykonala prostredníctvom metódy `execute()`. Vráti sa objekt typu `HttpResponse` s kódom odpovede. [6]

3 Ako si aplikáciu preniesť na vlastné zariadenie

Na spustenie aplikácie na vlastnom reálnom zariadení je potrebné mať usb kábel, ktorý sa zapojí do počítača, kde je aplikácia naprogramovaná. Ďalej je potrebné mať v SDK manažéri okrem API pre príslušnú verziu aj ovládač USB pre ADB – Android Debug Bridge. Ide o ovládač, ktorý sprostredkuje prenesenie aplikácie do reálneho zariadenia pomocou USB kábla. SDK manažér je možné spustiť z vývojového prostredia Eclipse, v hornom menu je záložka Window, kde nájdeme Android SDK manažér. Po spustení SDK manažéra sa zobrazia všetky balíčky, ktoré je možné doinštalovať, alebo aktualizovať tie, ktoré sme už prevzali. Nachádza sa tam aj balíček s názvom Extra. V tomto balíčku sa nachádzajú doplnky pre vývoj Android aplikácií. Ako vidno na obrázku (Obr. 27) tretia položka zdola je Google USB driver, ktorý je už nainštalovaný. Tento ovládač je možné nainštalovať aj na začiatku vývoja aplikácie, keď vieme, že aplikáciu chceme vytvoriť pre také zariadenie, na ktorom ju môžeme aj reálne vyskúšať.



Obr. 27 Doinštalovanie doplnku Google USB Driver pomocou SDK manažéra

Je možné stiahnuť aj proprietárny ovládač pre konkrétne zariadenie. Nakoniec treba už len v danom projekte upraviť Android Manifest.xml, kde je potrebné pridať Android: debuggable="true", tak ako je vidno v kóde nižšie.

```
<?xml version="1.0" encoding="utf-8"?>
```

```

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.medical.elidcard1"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="8"/>

    <application
        android:configChanges="keyboardHidden|orientation"
        android:icon="@drawable/ic_launcher"
        android:label="ELldCard"
        android:theme="@style/AppTheme"
        android:debuggable="true" >
        <activity
            android:label="ÚVODNÉ MENU"
            android:name=".MainActivity" >
            <intent-filter >
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <!-- Login Activity -->
        <activity android:label="PRIHLÁSENIE DO SYSTÉMU"
            android:name=".Layout1"></activity>
        <!-- Register Activity -->
        <activity android:label="REGISTRÁCIA"
            android:name=".RegisterActivity"></activity>
        <activity android:name=".ALLPatientsActivity"
            android:label="ZOZNAM PACIENTOV"></activity>
        <activity android:name=".EditPatientsActivity"
            android:label="ÚPRAVA OSOBNÝCH ÚDAJOV PACIENTA"></activity>
        <activity android:name=".EditLekarInfo"
            android:label="ÚPRAVA OSOBNÝCH ÚDAJOV LEKÁRA"></activity>
        <activity android:name=".Patient_uvod"
            android:label="PACIENT MENU"></activity>
        <activity android:name=".Patient_ou"
            android:label="OSOBNÉ ÚDAJE"></activity>
        <activity android:name=".Patient_view"
            android:label="PACIENT"></activity>
        <activity android:label="ZDRAVOTNÉ ÚDAJE"
            android:name=".Patient_zu"></activity>
        <activity android:name=".Patient_chorobopis"
            android:label="CHOROBOPIS"></activity>
        <activity android:label="ZDRAVOTNÉ ÚDAJE"
            android:name=".Patient_zu_view"></activity>
        <activity android:label="VYTVORENIE NOVÉHO ELEKTRONICKÉHO ÚČTU PACIENTA"
            android:name=".NewPatient"></activity>
        <activity android:label="VYTVORENIE OSOBNÝCH ÚDAJOV PACIENTA"
            android:name=".New_Pacient"></activity>
        <activity android:name=".Patient_chrobopis_all"
            android:label="VŠETKY CHOROBOPISY"></activity>
        <activity android:name=".Patient_ch_view"
            android:label="CHOROBOPIS"></activity>
        <activity android:name=".Patient_chrobopis_all_view"
            android:label="VŠETKY CHOROBOPISY"></activity>
        <activity android:name=".LekarInfo"
            android:label="LEKÁR"></activity>
        <activity android:name=".Liekky"

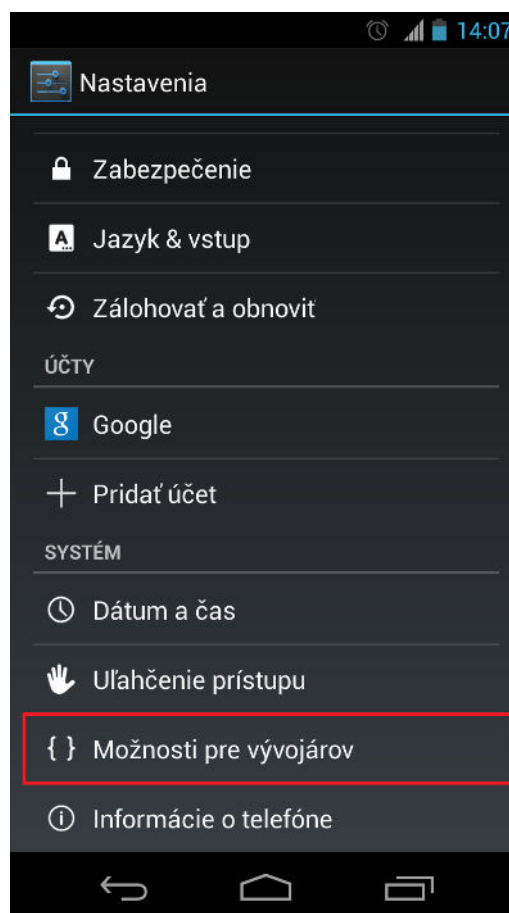
```

```
        android:label="LIEKY"></activity>
<activity android:name=".Liekky_uvod"
        android:label="LIEKY"></activity>
<activity android:name=".Liekky_kateg"
        android:label="KATEGÓRIE"></activity>
<activity android:name=".Web"
        android:label="WEB"></activity>
<activity android:name=".Liekky_zozn"
        android:label="ZOZNAM LIEKOV"></activity>

</application>

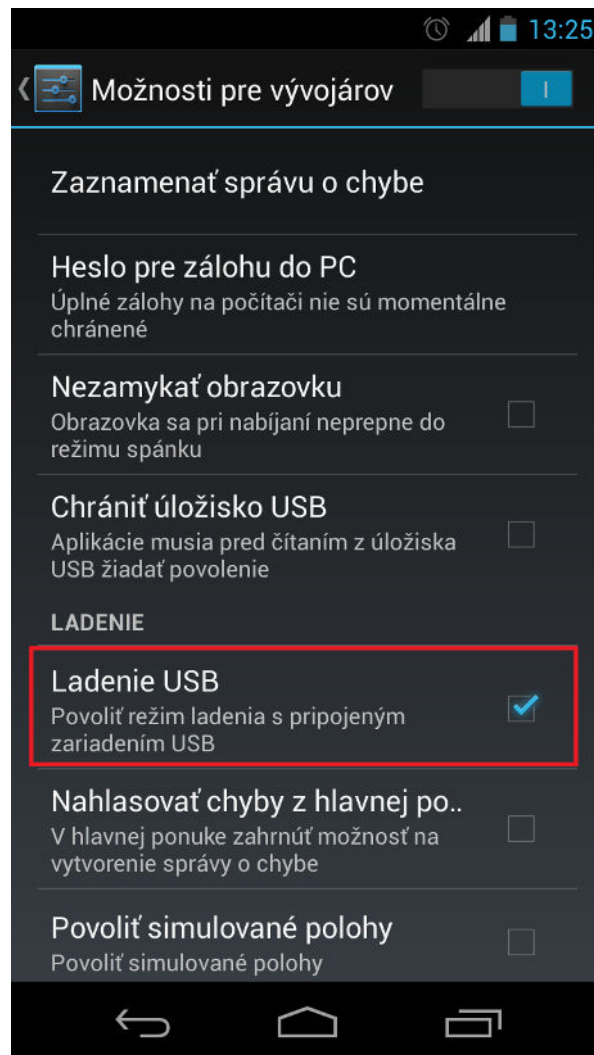
<!-- Internetové Práva --><uses-permission
android:name="android.permission.INTERNET" />
```

V reálnom zariadení je potrebné vojsť do nastavení, kde sa nachádza položka – Možnosti pre vývojárov, ako je vidno na obrázku nižšie (Obr. 28).



Obr. 28 Vývojárske nastavenia v OS Android

Po potvrdení sa zobrazí ďalšia obrazovka, na ktorej je potrebné povoliť režim ladenia s pripojeným zariadením USB (Obr. 29).



Obr. 29 Režim ladenia s USB zariadením

Po týchto krokoch je možné zapojiť presunúť aplikáciu na reálne zariadenie. Avšak ešte je možné, že si konkrétne zariadenie bude vyžadovať nainštalovať program určený od výrobcu. Napríklad pre mobilné telefóny HTC je potrebné mať v osobnom počítači nainštalovaný program HTC Sense pre konkrétny typ zariadenia. Nie sú potrebné žiadne nastavenia, len registrácia.

Zoznam použitej literatúry

- [1] Android SDK | Android Developers [online]. [cit.2013-05-23] Dostupné na Internete: <<http://developer.android.com/sdk/index.html>>
- [2] Dashboards | Android developer [online]. [cit.2013-05-23] Dostupné na internete: <<http://developer.android.com/about/dashboards/index.html>>
- [3] Eclipse Downloads [online]. [cit.2013-05-23] Dostupné na internete: <<http://www.eclipse.org/downloads/>>
- [4] About SQLite [online].[cit.2013-05-23] Dostupné na internete: <<http://www.sqlite.org/about.html>>
- [5] Apache friends – xampp [online]. [cit.2013-05-23] Dostupné na internete: <<http://www.apachefriends.org/en/xampp.html>>
- [6] Mark L.Murphy: Android 2 – Průvodce programováním mobilních aplikací. Brno: Computer Press, a.s., 2011. 377 s. ISBN 978-80-251-31