ŠIAULIŲ UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

INFORMATIKOS KATEDRA

EINARAS DAUNYS

Informatikos specialybės magistrantūros II kurso dieninio skyriaus studentas

Gedimų registracijos sistema delniniams kompiuteriams

Failure registration system for PDAs

MAGISTRO DARBAS

Darbo vadovė:
Doc. Dr. S. Turskienė

Darbo recenzentas:
Lekt. Dr. G. Felinskas

Šiauliai, 2009

Turinys

1. Įvadas

1.1 Temos aktualumas

Augant kompiuterinės technikos infrastruktūrai, svarbu yra užtikrinti atskirų jos komponentų nenutrūkstamą funkcionavimą, nes pavieniai komponentai gali būti gyvybiškai svarbūs visos sistemos stabiliam darbui. Išėjus iš rikiuotės pagrindiniams kompiuterinio tinklo mazgams, sustoja ir visos infrastruktūros darbas. Tad operatyviai gauta informacija apie sistemų gedimus, gali sumažinti iki minimumo technologines prastovas. Šiuo atveju yra labai svarbu pasirinkti tinkamą gedimų pranešimų bei registravimo sistemos modelį. Tokios sistemos modelis turi būti mažiausiai technologiškai priklausomas nuo eksploatuojamos kompiuterinės įrangos infrastruktūros.

Dažniausiai vidutinėse bei didelėse įmonėse ar įstaigose įvairūs programinės įrangos sprendimai projektuojami pasinaudojus kliento – serverio modeliu. Kliento – serverio modelis programinės įrangos architektūra naudojama centralizuotai teikianti tam tikras gimininges paslaugas paskirstytose sistemose[1].

Paskutiniu metu labai populiarėja delniniai kompiuteriai PDA (Personal Digital Assistant). Vien 2006 metų trečią ketvirtį PDA pardavimai padidėjo 36 procentais (Gartner Research). Jie yra maži, patogūs, lengvi, o svarbiausia jie turi visas savybes būdingas personaliniams kompiuteriams. Delninius kompiuterius gamina didžiausi pasaulio kompiuterinės technikos bei mobiliosios elektronikos gamintojai: *Palm Inc., Hewlett Packard (HP), HTC, Asus, Dell, Apple, SHARP* ir t.t.

Pagrindinės delninio kompiuterio savybės – tai jo maži gabaritai, o svarbiausia įvairių duomenų perdavimo technologijų bei sąsajų palaikymas (Wi-Fi, IrDA, BlueTooth). Tai leidžia priimti bei perduoti reikalingus duomenis internetu, perkelti informaciją į personalinį kompiuterį ar į mobilųjį telefoną. Taigi, mobiliosios technologijos tampa neatsiejama buities, pramogų, verslo, mokslo dalimi ir Lietuvoje.

Šio darbo idėja – perkelti Šiaulių Universiteto kompiuterinės technikos gedimų registracijos sistemą į delninius kompiuterius. Tai leis atsakingiems už kompiuterines sistemas asmenis

[1] *Charles Consel Laurent Réveillère A Programmable Client-Server Mode*l ENSEIRB 1, avenue du docteur Albert Schweitzer,Domaine universitaire - BP 99

operatyviau gauti informaciją apie gedimus, užregistruoti ir nusiųsti gedimą esant vidinio kompiuterinio tinklo sutrikimams, kas būtų visiškai neįmanoma su jau eksploatuojama sistema, pasiekti reikalingus duomenis, būnant bet kurioje pasaulio vietoje, kur yra įmanomas Wi-Fi, GPRS, UMTS duomenų perdavimas, ir visa tai telpa atsakingo inžinieriaus kišenėje.

1.2. Mokslinis naujumas

Šiuo metu delniniams kompiuteriams yra sukurta daugybė programinės taikomosios įrangos. Kaip žinome, paprastai programos yra rašomos ir kompiliuojamos tam tikrai pasirinktai kompiuterinei architektūrai bei platformai. Populiariausios delniniuose kompiuteriuose naudojamos operacinės sistemos: *Palm OS* bei *Windows CE, Windows Mobile*, užimančios virš 80% rinkos. Po truputį plinta, *Symbian* bei *Linux* operacinių sistemų naudojimas. Kas tinka *Palm OS*, netinka *Windows Mobile*, kas tinka *Windows mobile*, netinka *Linux* ar *Symbian*. Taip pat skiriasi įvairių gamintojų ir aparatinė „delninukų“ architektūra. Visi šie skirtumai labai apsunkina unifikuotos programinės įrangos kūrimą. Norint sukurti programą veikiančia įvairiose platformose, reikia atsižvelgti į labai daug faktorių ir visus juos tinkamai įvertinti, o kartais tai būna ir neįmanoma. Tai gi ką privalo pasirinkti programinės įrangos kūrėjas? Ar adaptuoti programos išeities tekstus įvairioms platformoms, kas reikalauja daug laiko resurso bei specifinių žinių, ar pasirinkti kitą kelią? Taip, belieka rinktis kitą kelią ir tas kelias yra tai *Java*^{TM} vykdančioji aplinka bei *Java* programavimo kalba.

Java – objektiškai orientuota programavimo kalba, 1991 metais sukurta Džeimso Goslingo ir kitų Sun Microsystems inžinierių. Apie ją oficialiai paskelbta 1995 metų gegužės 23 d., o išleista tų pačių metų lapkritį. Java (pradžioje vadinta Oak) kalbos pirminis tikslas buvo pakeisti C++ kalbą. Šios programavimo kalbos yra dvi pagrindinės savybės, kurių neturi kitos objektiškai orientuotos kalbos, tai virtualiosios mašinos atsiradimas, bei keletas Java kalbos sukurtų platformų, kurios turi tam tikrus iškeltus uždavinius. Naudodamiesi šia technologija, parašytas bei sukompiliuotas programas *Java* kalba, galime paleidinėti įvairiose platformose bei operacinėse sistemose tokiose kaip *Windows, Linux, AIX, Solaris, HP-UX, OpenVMS*, ten kur yra įdiegta *Java* aplinka. *Java* programuotojų devizas – „rašai vieną kartą, naudoji bet kur“. Kuriant Java, pagrindiniai tikslai buvo šie:

- Kalba turi būti objektiškai orientuota.

- Kalba turi būti nepriklausoma nuo naudojamos platformos (beveik nepriklausoma).
- Savyje kalba turi turėti priemones ir bibliotekas komunikacijai tinklu.
- Kalba turi būti suprojektuota taip, kad kodas iš nutolusio šaltinio būtų vykdomas saugiai.

Java platforma yra sudaryta iš trijų pagrindinių dalių:

- J2SE – Java 2 Platform, Standard Edition. Tai *Javos* širdis, šioje platformoje yra pateikiamos visos bazinės bibliotekos ir įrankiai, kurie naudojami komandinės eilutės ir grafinę sąsają turinčioms programoms kurti.
- J2EE – Java 2 Platform, Enterprise Edition. Ši dalis skirta kurti informacines verslo sistemas. Tiesiog J2SE yra papildoma įvairiomis technologijomis, suteikiančiomis galimybę kurti Web programas (*Java Servlet, JavaServer Pages, JavaServer Faces* ir t.t.), išskirstytas sistemas, apibrėžia daugkartinio panaudojimo komponentus (Enterprise JavaBeans), pateikia šūsnį standartų ir t.t.
- J2ME – Java 2 Platform, Micro Edition. Tai platforma, kuri pateikia įrankių rinkinį kurti programas tokiems mobiliems įrenginiams, kaip mobiliesiems telefonams ar sumaniesiems telefonams.

Deja programinis kodas kuris buvo rašomas taikant J2SE platforma neveiks J2ME ir atvirkščiai. Tas pats liečią ir J2EE į tai reikia atkreipti didelį dėmėsi pasirenkant naudojimo platformą.

Mobiliuosiuose įrenginiuose, dėl ribotų techninių išteklių, paprastai taikoma J2ME technologija, vadinasi negalėsime paleisti programas, kurios sėkmingai veikia personaliniuose kompiuteriuose.

Nei *Sun Microsystems* nei *Microsoft* nepasirūpino išleisti J2SE delniniams kompiuteriams, nors šių kompiuteriukų pajėgumai visiškai laisvai leistų interpretuoti bei vykdyti *Java byte* kodą. Senesnėse *PalmOS* versijoje buvo realizuojama Java virtuali mašiną, bet nuo 2008 metų *Palm Inc.* nutraukė jos palaikymą.

1.3. Praktinė vertė

Didėjant delninių kompiuterių vartojimui, bei gausėjant įvairioms platformoms, atsiranda poreikis sukurti programinę įrangą, kuri veiks daugumoje įrenginių. Vartotojas, norėdamas naudotis tam tikromis programomis, neturėtų rinktis konkrečios platformos delninį kompiuterį. Pasinaudojus J2SE technologiją programa tampa nepriklausoma nuo kompiuterio architektūros. Tad ją galime naudoti ir delniniame kompiuteryje.

1.4. Tyrimo metodai

Literatūros analizė. Delninių kompiuterių technologijų analizė (aparatinė architektūra, operacinės sistemos, programų kūrimo įrankiai). Tinkamiausios programinės įrangos parinkimas, vartotojų poreikio analizė.

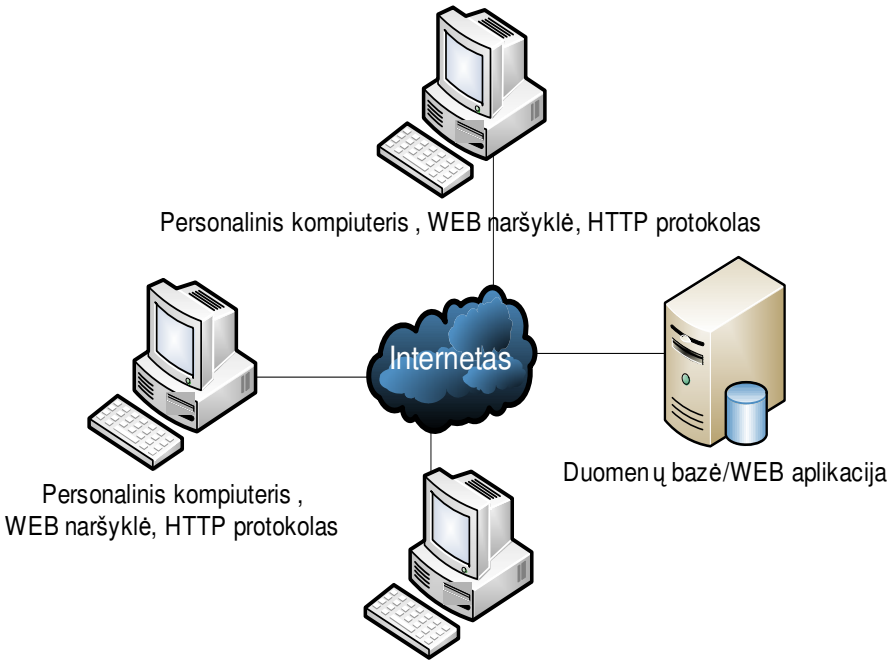
1.5. Darbo tikslas ir uždaviniai

Darbo tikslas – sukurti universalią kompiuterinių sistemų gedimų registravimo programinę įrangą delniniams kompiuteriams. Atsižvelgiant į darbo tikslą uždaviniai yra tokie:

- Išanalizuoti delninių kompiuterių technologines galimybes.
- Išanalizuoti naudojamas operacines sistemas delniniuose kompiuteriuose.
- Parinkti optimalią programos kūrimo platformą.
- Ištirti jau esamus panašius produktus.
- Sukurti veikiantį programinės įrangos modelį.

2. Problemos apibrėžtumas

Dabartinė gedimų registravimo sistema eksploatuojama Šiaulių Universitete struktūriškai atrodo taip:



Gedimų registracija vykdoma per WEB naršyklę ir HTTP protokolą, prisijungus prie tarnybinės stoties. Tokia sistema turi sekančius technologinius trūkumus:

- Išėjus iš rikiuotės tarnybiniai stočiai, sistema visiškai nebefunkcionuoja.
- Nutrūkus interneto ar vietinio tinklo ryšiui, nebeįmanoma prisijungti prie sistemos, užregistruoti naujo gedimo, ar peržiūrėti gedimų žurnalo.
- Sistema jautri pagalbinėms interneto tarnyboms, tokioms kaip DNS.

Gedimų registracijos sistema delniniams kompiuteriams, praplės esamas stacionarias gedimų registravimų sistemas. Techniniams darbuotojams leis nevaržomai judėti pastoviai prisijungus prie duomenų bazių bei operatyviai reaguoti į tam tikrus įvykius. Išskaidant sistemą į atskirus modulius, išvengiama technologinių bei gamybinių prastovų, sugedus tik vienam moduliui. Priešingai jei sistema yra sukonstruota kaip vieninga visuma, išėjus iš rikiuotės vienam komponentui sustoja visa sistema.

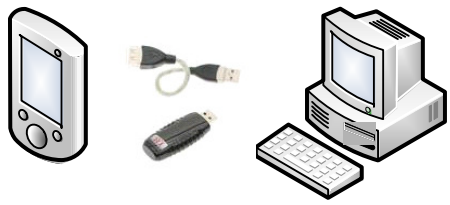
3. Teorinė dalis

3.1. Delninių kompiuterių technologinė analizė

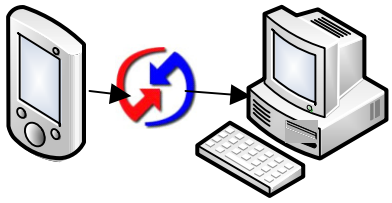
Pirmasis delninis kompiuteris buvo pagamintas 1983 metų gegužės mėnesį. Tai buvo CASIO PF-3000 modelis. Terminas PDA (Personal digital assistant) pirmą kartą buvo pavartotas 1992 metų sausio 7-ą dieną elektronikos parodoje Las Vegase, kai Apple Inc. pristatinėjo savo Apple Newton delninį kompiuterį. 1996 metais Nokia pristatė komunikatorių Nokia 9000.

Visi delniniai kompiuteriai nuo atsiradimo iki šių dienų išlaikė pagrindines savybes:

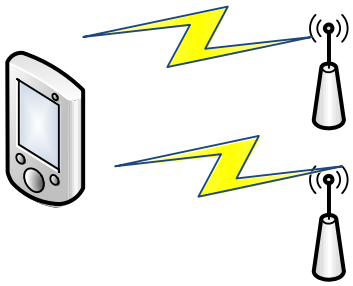
- Nedideli gabaritai, leidžiantys įrenginį nešioti kišenėje.
- Lietimui jautrus ekranas, kuris atstoja įprastą klaviatūrą.
- Papildoma sąsaja išoriniam duomenų kaupikliui prijungti (MMC, SD atminties kortelės).
- Bevielės duomenų perdavimo sąsajos (IrDa – infraraudonųjų spindulių, Bluetooth – bevielis duomenų perdavimas nedideliu atstumu iki 10 metrų, Wi-Fi spartus duomenų perdavimas vidutiniu atstumu. Taip pat gali būti naudojamos ir kitos bevielės technologijos GSM, GPRS, UMTS, HSPDA paprastai naudojamos su integruotu mobiliuoju telefonu).
- Laidinė duomenų perdavimo sąsaja, skirta sinchronizuoti duomenis su personaliniu kompiuteriu.



1 pav. Delninis kompiuteris sujungtas per Infraraudonųjų spindulių sąsają



2 pav. Delninis kompiuteris sujungtas per laidinę sąsają



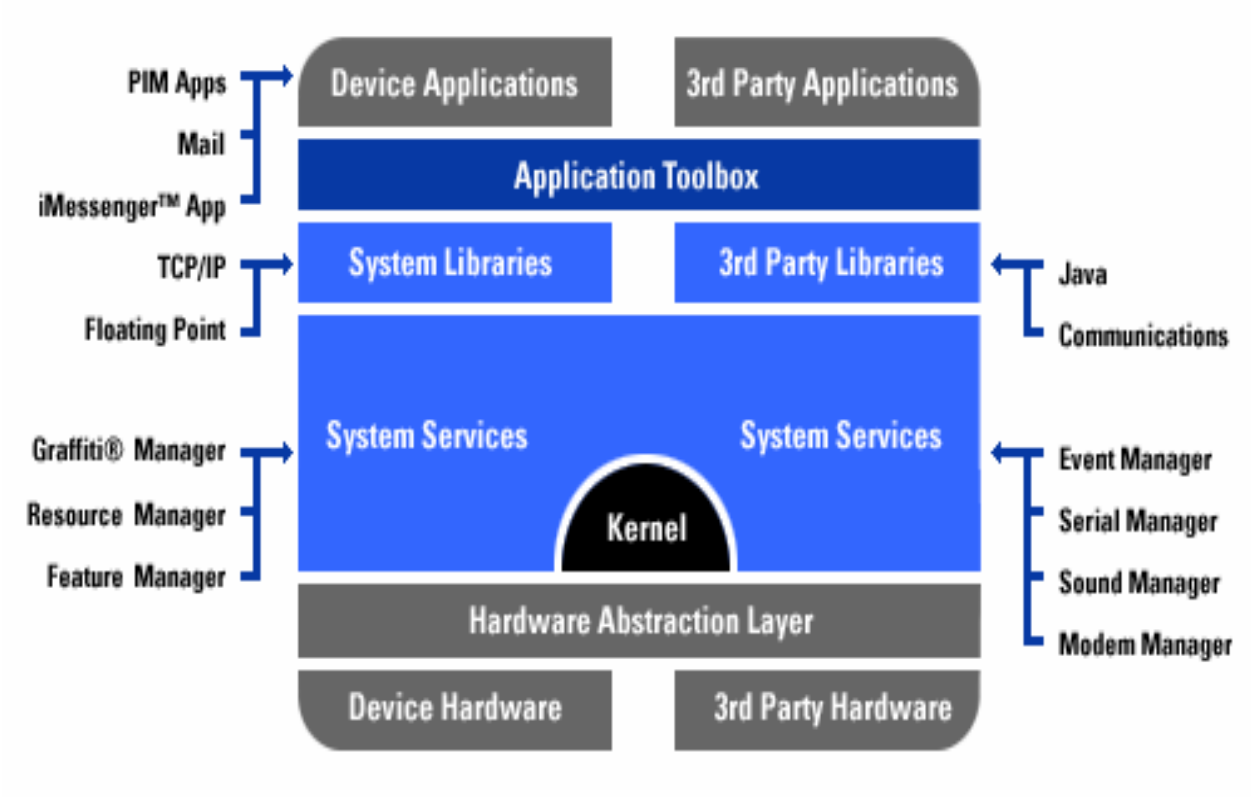
3 pav. Delninis kompiuteris prisijungęs tiesiogiai į GSM/GPRS/UMTS tinklą

Nors iš pažiūros ir turimų savybių įvairių gamintojų gaminiai atrodo vienodi, tačiau juose slypi didžiuliai skirtumai aparatiniame bei programiniame aspekte. Kiekvienas gamintojas naudoja tam tikrų tipų procesorius (ARM, Motorola, Intel), kurie savo instrukcijų komandomis tarpusavyje yra nesuderinami. Taip pat skiriasi ir delninių kompiuterių operacinės sistemos. Rinkoje labiausiai yra paplitusios PalmOS bei WindowsCE / Windows Mobile operacinės sistemos. Jų užimama rinkos dalis yra apie 80 procentų. Jei žvelgtume, giliau ta pati Windows Mobile skirtingose versijose turi tam tikrų programinių apribojimų.

	Mobilieji telefonai	**„Protingieji“ telefonai**	**Delniniai kompiuteriai**	**Personaliniai kompiuteriai PC**
Ekranas	~170x220 taškų, 65000 spalvų.			
Procesoriaus	~50 MHz	~ 300 MHz	~ 500 MHz	~3GHz

dažnis				
Vidinė atmintinė	~64MB	~128MB	~256MB	Priklauso nuo duomenų kaupiklio
Duomenų perdavimas	IrDA, Bluetooth, laidinis	IrDA, Bluetooth, laidinis, Wi-Fi GSM, GPRS, UMTS, HSPDA	IrDA, Bluetooth, laidinis, Wi-Fi (GSM, GPRS, UMTS, HSPDA – su integruotu telefonu)	Ethernet, Wi-FI, IrDA, Bluetooth ir t.t.
Operacinė sistema	Gamintojo programinė įranga	Symbian OS, Palm OS, Windows Mobile, Pocket PC, BlackBerry, Linux	Symbian OS, Palm OS, Windows Mobile, Pocket PC, BlackBerry, Linux	Windows, Linux, MacOS X, UNIX ir t.t.

Lentelė Nr1. Įrenginių palyginimas



Pav. 4 PalmOS operacinės sistemos struktūra[3]

PalmOS operacinės sistemos pagrindinės savybės:

- Padidintas saugumas maskuoti ar slėpti privačius duomenis.
- Auto užrakinimas.
- Stipri apsauga vartotojo įrašytiems slaptažodžiams.
- Laiko zonų palaikymas telefoninės paslaugos.
- Java palaikymas.

[3] Aarne Klemetti, PDA Operating Systems

3.2. Aplikacijų kūrimo priemonės

Kiekviena kompiuterinė platforma turi jai skirtus programų kūrimo įrankius, specifikuotas bibliotekas ir skirtingas struktūras. Kuriant universalias programas tinkančias įvairioms platformoms bei operacinėms sistemoms, programuotojai gali susidurti su begalę sunkumų, kai reikia įvertinti tam tikras specifikas. Programuotojai tada privalo rašyti labai lankstų programinį kodą, naudoti begales kompiliatorių direktyvų bei sudaryti painų kompiliatoriaus konfigūracijos komandų rinkinį. Tam, kad viso to išvengti, galima pasinaudoti specialiais sprendimais. Vienas iš sprendimų būtu pasinaudoti virtualia mašina arba kitaip unifikuoto kodo interpretatoriumi.

Virtuali mašina tai programinės įrangos rinkinys interpretuojantis sukompiliuotą pseudo mašininį kodą arba tam tikrus scenarijus (*scripts*). Programuotojui rašančiam programas naudojant šį modelį nebereikia rūpintis platformos specifikacijomis. Jis rašo unifikuotą kodą pasirinktai virtualiai mašinai. Tam kad pasinaudoti ir paleisti programą, belieka tik įdiegti tinkamą virtualią mašiną į kompiuterį.

Labiausiai paplitusios kompiuterinėse sistemose yra Java™ virtualios mašinos vadinamos Hot Spot‘ais. Jas galima naudoti labai plačiame kompiuterinių platformų bei operacinių sistemų spektre. Specialiai šioms virtualioms mašinoms, arba dažnai vadinamoms vykdymo aplinkomis (*runtime environments*) yra sukurta Java programavimo kalba. Java programavimo kalbos geroji savybė yra ta, kad yra sukurta nemažai aplinkų, kaip NetBeans BlueJ, Eclipse, JDeveloper ir kitos, kurios orientuotos į vartotojo grafinę sąsają bei veikia įvairiose operacinėse sistemose.

Java laikoma revoliuciniu programavimo žingsniu pirmiausia dėl programavimo galimybių internete – kalbant tiek apie klientinį, tiek apie serverinį programavimą. Pagrindinė klientinių - serverinių sistemų idėja ta, jog centralizuotas informacijos bankas yra serveryje, į kurį gali kreiptis grupė klientinių kompiuterių ar mobiliųjų įrenginių. Serveryje saugomos informacijos pokyčiai retai prieinami klientiniams įrenginiams.

4. Projektinė dalis. Įrankių ir priemonių pasirinkimas

4.1. Java integruotų programavimų aplinkų apžvalga

Patogiam programavimui egzistuoja labai daug integruotų programavimo aplinkų. Šios aplinkos programuotojams labai palengvina darbą lyginant su darbu elementariajame tekstų redaktoriuje. Programuotojas tam tikrose srityse iš karto gali matyti failų navigatorių, klasių ir objektų tarpusavio ryšius, *auto completion* – viena iš geriausių savybių, automatiškai parodantį objektų ar klasių metodus, parametrus ir t.t.

Integruota programavimo aplinka	**Licenzija**	**Java virtuali aplinka**	**Platformos**	**Grafinės vartotojo sąsajos dizaineris**
BEA Workshop for WebLogic	Apribota (Proprietary)	Taip	Windows, Linux	Taip
BlueJ	GPL v2	Taip	Windows, Mac OS X, Linux, Solaris	Ne
DrJava	laisva	Taip	Windows, Mac OS X, Linux, Solaris	Ne
Eclipse JDT	Eclipse Public License	Taip	Windows, Mac OS X, Linux, Solaris	Taip
Geany	GPL	Ne	Windows, Linux	Ne
Greenfoot	Pusiau laisva	Taip	Windows, Mac OS X, Linux, Solaris	Ne
IntelliJ IDEA	Apribota (Proprietary)	Taip	Windows, Mac OS X, Linux	Taip
JBuilder	Apribota	Taip	Linux, Solaris, Windows	Taip

	(Proprietary)			
JCreator	Apribota (Proprietary)	Ne	Windows	Ne
JDeveloper	Apribota (Proprietary)	Taip	Windows, Mac OS X, Linux, generic JVM	Taip
KDevelop	GPL	Ne	Linux	Taip
NetBeans	CDDL GPL2	Taip	Windows, Mac OS X, Linux, Solaris	Taip

Lentelė Nr2. Java programavimo aplinkų palyginimas

Katrą programavimo aplinką reikėtų rinktis, viskas priklauso nuo to kokioje platformoje jūs dirbate, kokiu papildomų savybių reikalaujate iš sistemos ir t.t. Atsižvelgiant į tai jei reikia aplinkos su laisva licenzija, su papildomai teikiama JVM ir vartotojo sąsajos dizaineriu, bei veikiančia daugelyje platformų belieka du pasirinkimai: *NetBeans* bei *Elipse*. Šios sistemos savo funkcijomis ir galimybėmis yra labai panašios. Jas abi galima plėsti papildomais moduliais bei programavimo paketais. Ir jos abi nėra vien tiktai pritaikytos programavimui *Java* kalba.

4.2. NetBeans integruota programavimo aplinka

NetBeans yra atviro kodo integruota programavimo aplinka. Kadangi ji pati parašyta *Java* programavimo kalba, tai šį programinį produktą galima naudoti daugumoje populiariųjų operacinių sistemų. *NetBeans* yra sukonstruotas taip, kad jo funkcionalumą galima būtų plėsti papildomais įskiepiais (*plugins*), arba moduliais. Todėl šis produktas tampa lanksčiu ir plačiai taikomu praktikoje.

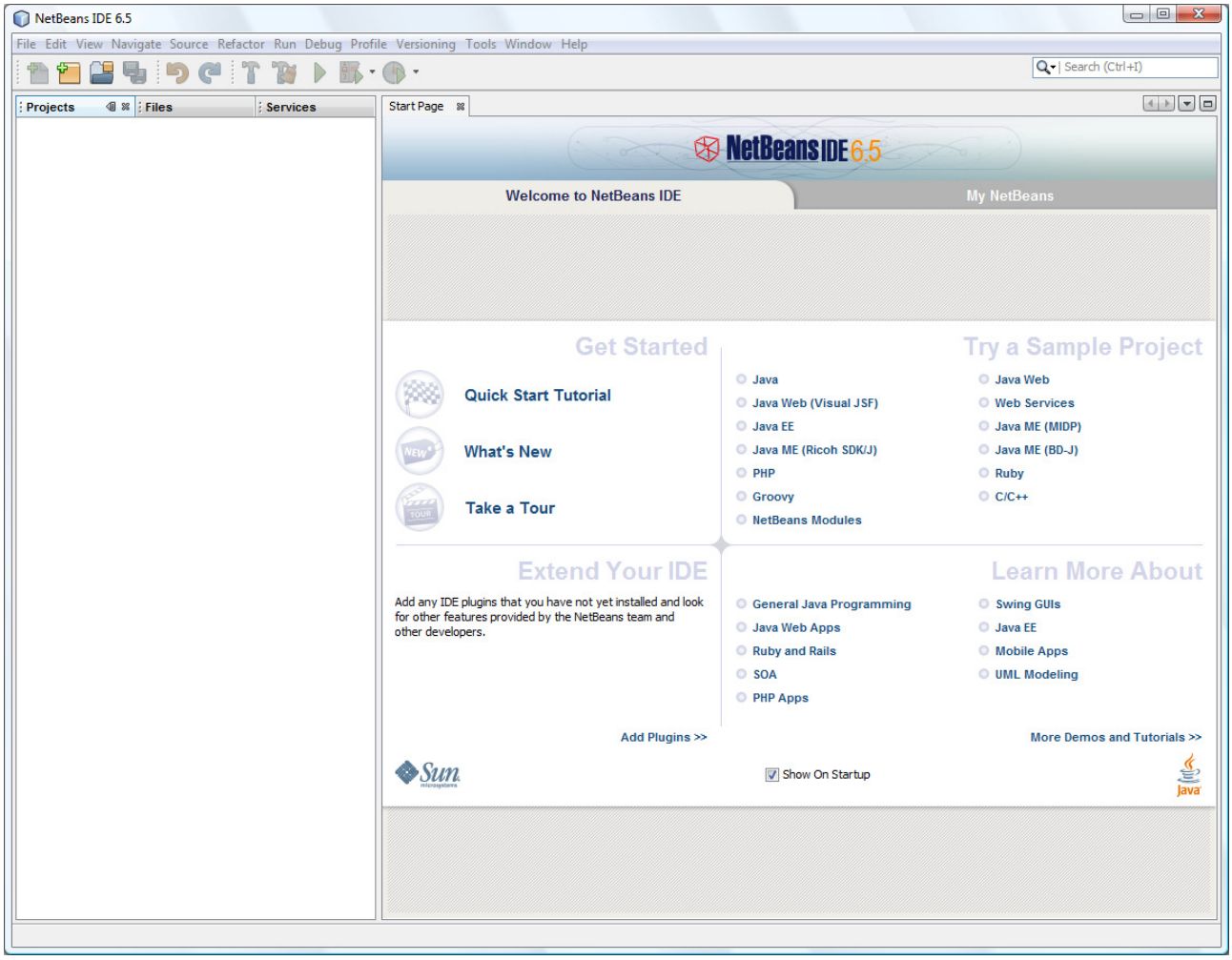
NetBeans savo gyvavimą pradėjo 1997 metais kaip *Xelfi* projektas. Vėliau 1999 metais šį projektą įsigijo *Sun Microsystems* ir laisvai išplatino jo išeities tekstus. Taip projektas susilaukė didelio dėmesio visame pasaulyje ir sėkmingai gyvuoja iki šių dienų.

Paskutinė projekto 6.5 versija labai praplėtė JavaEE (Java Persistence, EJB 3 bei JAX-WS) technologijomis paremtų projektų kūrimą. Kaip Java EE 5 enterprise applications, įskaitant SOA vaizdinio modeliavimo priemones, XML schemų įrankius, *WEB* tarnybas, ir UML modeliavimas.

Dėka savo modulinės struktūros NetBeans ženkliai palengvina programuotojo darbą. Įtraukiant į sistemą papildomus modulius, pilnai arba iš dalies galima automatizuoti programos kūrimą. Pavyzdžiui programuotojui nereikia rūpintis pasikartojančiais kodo blokais, sistema pati automatiškai sukuria šabloninius procedūrų kūnus. Taip pat jei kuriama desktop tipo aplikacija, integruoti grafiniai dizaineriai sugeneruoja kodą, priklausomai nuo atliekamų veiksmų.

Pilnas NetBeans paketas apima šiuos komponentus:

- NetBeans Base IDE, bazinė platforma.
- Java SE aplikacijų kūrimas.
- Web bei Java EE aplikacijų kūrimas.
- Mobility, aplikacijų mobiliems įrenginiams kūrimas.
- SOA, į paslaugas orientuotų sistemų kūrimas.
- Ruby posistemė.
- C/C++ programų rašymo galimybė bei kompiliavimas.
- PHP (nuo 6.5 versijos) aplikacijų kūrimas.
- GlassFish aplikacijų serveris.
- Apache Tomcat servletų ir JSP konteineris (serveris).



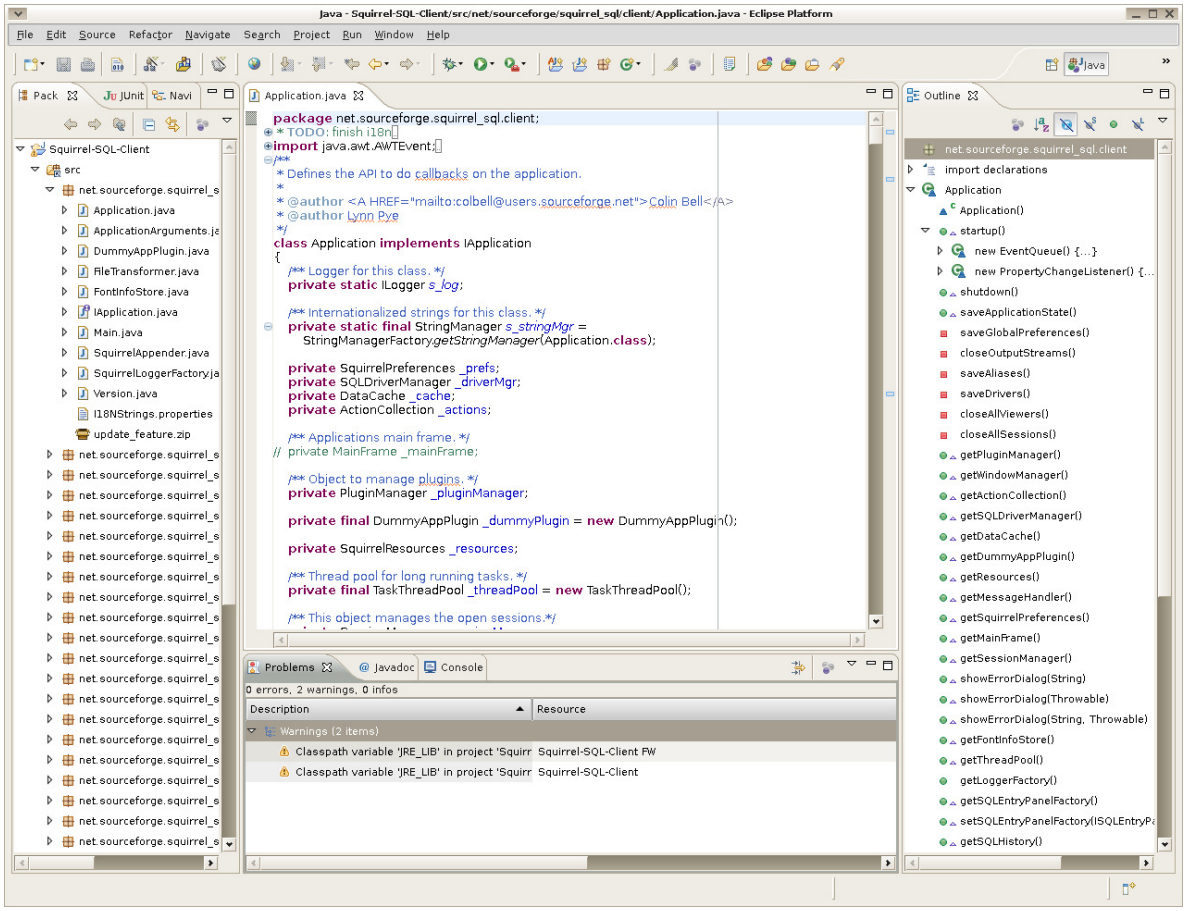
Pav 5. NetBeans aplinkos pagrindinis langas

4.3. Eclipse integruota programavimo aplinka

Eclipse yra daugiakalbė programavimo aplinka. Daugiakalbė reiškia, kad šios platformos pagalba galima rašyti programas įvairiomis programavimo kalbomis. Dėl modulinės struktūros galima į sistemą integruoti įvairius papildomus programavimo modulius. Pati *Eclipse* kaip ir *NetBeans* parašyta Java programavimo kalba, tačiau įskiepių pagalba galima įtraukti ir kitų programavimo kalbų palaikymą, kaip *C/C++, Cobol, Python, Perl, PHP* bei daugelį kitų.

Eclipse pradinis kodas buvo perimtas iš *IBM VisualAge* programavimo aplinkos programavimui Java kalboje. Vėliau ši sistema buvo realizuota *Eclipse Public License* licenzija ir tapo atviro kodo produktu.

Eclipse sistemos architektūra sukonstruota taip, lyg ji būtu atskira operacinė sistema. Ši sistema susideda iš pagrindinio branduolio ir atskirų įskiepių. Tokia sistema be galo paprasta plėsti ir tobulinti. Norint papildyti tam tikru funkcionalumu nereikia perrašyti ar perkompiliuoti produktą, o užtenka integruoti papildomus modulius.



Pav 6. *Eclipse* pagridinis langas

Taigi abi sistemos tiek *NetBeans* tiek *Eclipse* turi labai daug funkcinių ir technologinių panašumų. Vertinti kuri yra pranašesnė būtų labai sunku, šiame darbe iš pradžių naudosime abi šias sistemas ir nuo darbo eigos rezultatų bus parinkta pagrindinė programavimo aplinka.

4.4. Duomenų bazių valdymo sistemų apžvalga

Reliacinės duomenų bazių valdymo sistemos - tokios duomenų bazių valdymo sistemos (DBVS), kurios remiasi Edgaro Kodo (Edgar F. Codd) aprašytu reliaciniu modeliu, kurio pagrindinis elementas yra reliacinė lentelė. Dauguma šiuolaikinių populiariųjų DBVS palaiko didžiąją dalį E. Kodo suformuluotų taisyklių: Oracle, Microsoft SQL Server, IBM DB2, Microsoft Access, MySQL, PostgreSQL, tačiau jomis neapsiriboja. Kiekvienas gamintojas siūlo savus plėtinius ir papildymus.

Šiuo metu vienas iš didžiausių duomenų bazių valdymo sistemų uždavinių yra gebėjimas kuo daugiau sukaupti duomenų ir kuo greičiau juos apdoroti bei pateikti vartotojui. Praktiškai visos duomenų bazių valdymo sistemos apart įprasto duomenų kaupimo turi ir papildomas funkcines savybes, kurios leidžia rašyti tam tikrus programavimo scenarijus (procedūras, funkcijas, įvykių apdirbimus).

4.7. Populiarių duomenų bazių valdymo sistemų palyginimas pagal tam tikrus kriterijus

Lentelė Nr 3. Pagrindinė informacija

DBVS	**Platintojas**	**Išleidimo metai**	**Paskutinė versija**	**Licenzija**
DB2	IBM	1982	9.5	Apribota
Firebird	Firebird project	2000	2.1.2	IPL bei IDPL
Informix	IBM	1985	11.50	Apribota
Microsoft SQL Server	Microsoft	1989	2008 (v10.0)	Apribota
MySQL	Sun Microsystems	1996	5.1.32	GPL arba proprietary
Oracle	Oracle Corporation	1979	11g Release 1	Apribota

PostgreSQL	PostgreSQL Global Development Group	1989	8.3.7	BSD
SQLite	D. Richard Hipp	2000	3.6.7	Public domain

Lentelė Nr 4. Operacinių sistemų palaikymas

DBVS	**Windows**	**Mac OS X**	**Linux**	**BSD**	**UNIX**	**Symbian**
DB2	Taip	Ne	Taip	Ne	Taip	Ne
Firebird	Taip	Taip	Taip	Taip	Taip	Ne
Informix	Taip	Taip	Taip	Taip	Taip	Ne
Microsoft SQL Server	Taip	Ne	Ne	Ne	Ne	Ne
MySQL	Taip	Taip	Taip	Taip	Taip	Taip
Oracle	Taip	Taip	Taip	Ne	Taip	Ne
PostgreSQL	Taip	Taip	Taip	Taip	Taip	Ne
SQLite	Taip	Taip	Taip	Taip	Taip	Taip

Lentelė Nr 5. Funkciniai apribojimai

DBVS	**Maksimalus dydis**	**Maksimalus lentelės dydis**	**Maksimalus įrašo dydis**	**Maksimalus laukų dydis**	**Maksimalus BLOB / CLOB dydis**	**Maksimalus simbolių dydis**	**Maksimalus skaičių dydis**
DB2	512 TB (512 TiB)	512 TB	32,677 B	1012	2 GB	32 KB	64 bitai
Firebird	Neribojama	~32 TB	65,536 B	Priklauso nuo duomenų tipo	2 GB	32,767 B	64 bitai
Microsoft SQL Server	524,258 TB	524,258 TB	Neribojamas	1024	2 GB	8000 B	64 bitai
MySQL	Neribojama	2 GB (Win32 FAT32) iki 16 TB (Solaris)	64 KB	4096	4 GB	64 KB	64 bitai
Oracle	Neribojama	4 GB	Neribojama	1000	Neribojama	4000 B	126 bitai
PostgreSQL	Neribajama	32 TB	1.6 TB	Priklauso nuo duomenų tipo	1 GB	1 GB	Neribojama
SQLite	32 TB	?	?	2000	1 GB	1 GB	64 bitai

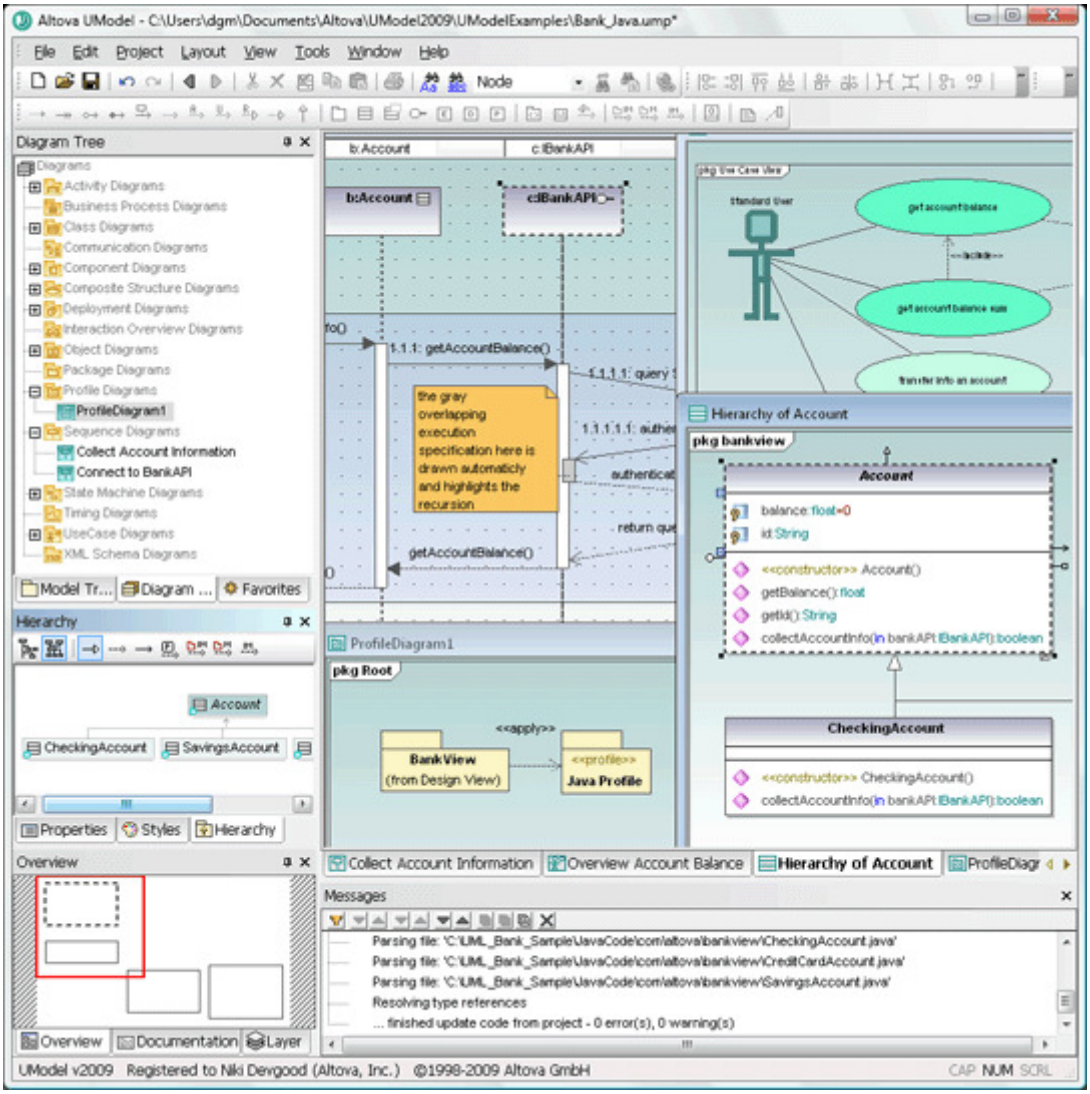
Atsižvelgiant į licenzijos apribojimus, palaikomų operacinių sistemų kiekį bei funkcines savybes artimiausios DBVS būtų *MySQL* bei *SQLite*. *MySQL* duomenų bazė tinkama kaip serverinė duomenų bazių valdymo sistema, o *SQLite* lokali duomenų bazė mobiliajame įrenginyje.

4.8. Aplikacijų modeliavimo priemonės

Objektiškai orientuotų programų projektavimo etape dažniausiai naudojamos UML (Unified Modeling Language, vieninga modeliavimo kalba). Ši kalba skirta specifikuoti, atvaizduoti ir konstruoti objektiškai orientuotų programų dokumentus. Šiuo metu UML yra labiausiai paplitęs programinės įrangos specifikavimo standartas, palaikomas įvairių gamintojų *Borland, IBM, Telelogic, No Magic*.

UML diagramų kūrimo priemonių ratas yra labai gausus. Egzistuoja kelias dešimtys atviro kodo produktų bei komercinių paketų. Produktai daugiausia skiriasi operacinių sistemų palaikymu, naudojimosi licencija, programavimo kalbų palaikymu bei UML standarto palaikymu.

Šiame darbe buvo parinktas Altanota kompanijos UML projektavimo paketas *UModel*. Šis produktas pasižymi plačiomis techninėmis galimybėmis: UML diagramų generacija į išeities tekstus Java, C#, or Visual Basic .NET, automatinis projekto dokumentavimas, grafinis dizaineris, palaiko 14 UML 2.2 versijos diagramų, patogi ir intuityvi vartotojo sąsaja. Diagramas patogu skaityti, jos išsiskiria spalvomis, bei skirtingais stiliais. Šį produktą nemokamai galima bandyti 30 dienų.



Pav 7. *Altanota UModel* programinis paketas

5. Gedimų registravimo aplikacijos realizavimas

5.1. Įvadas

Gedimų registravimo aplikacijos paskirtis yra praplėsti jau esamos ir naudojamos Šiaulių Universitete gedimų sistemos funkcionalumą bei ergonomiškumą. Aplikacija leis techniniams darbuotojams registruoti, bei gauti informaciją apie sistemų gedimus būnant bet kurioje pasaulio vietoje ten kur yra bet kokia prieiga prie interneto. Programa yra realizuota Java programavimo kalba, tad ją bus galima paleisti ten kur yra įdiegta Java vykdomoji aplinka.

Kadangi šio darbo tikslas yra realizuoti aplikaciją delniniams kompiuteriams *PDA*, tai analizuosime programines technologijas tinkančias šiems įrenginiams. Modelio realizavimas atliekamas keliais etapais:

- Techninių sąlygų bei užduoties analizė.
- Programinės įrangos diegimas.
- Duomenų bazės projektavimas ir kūrimas.
- Aplikacijos projektavimas ir kūrimas.
- Sistemos testavimas.
- Tobulinimo perspektyvų numatymas.

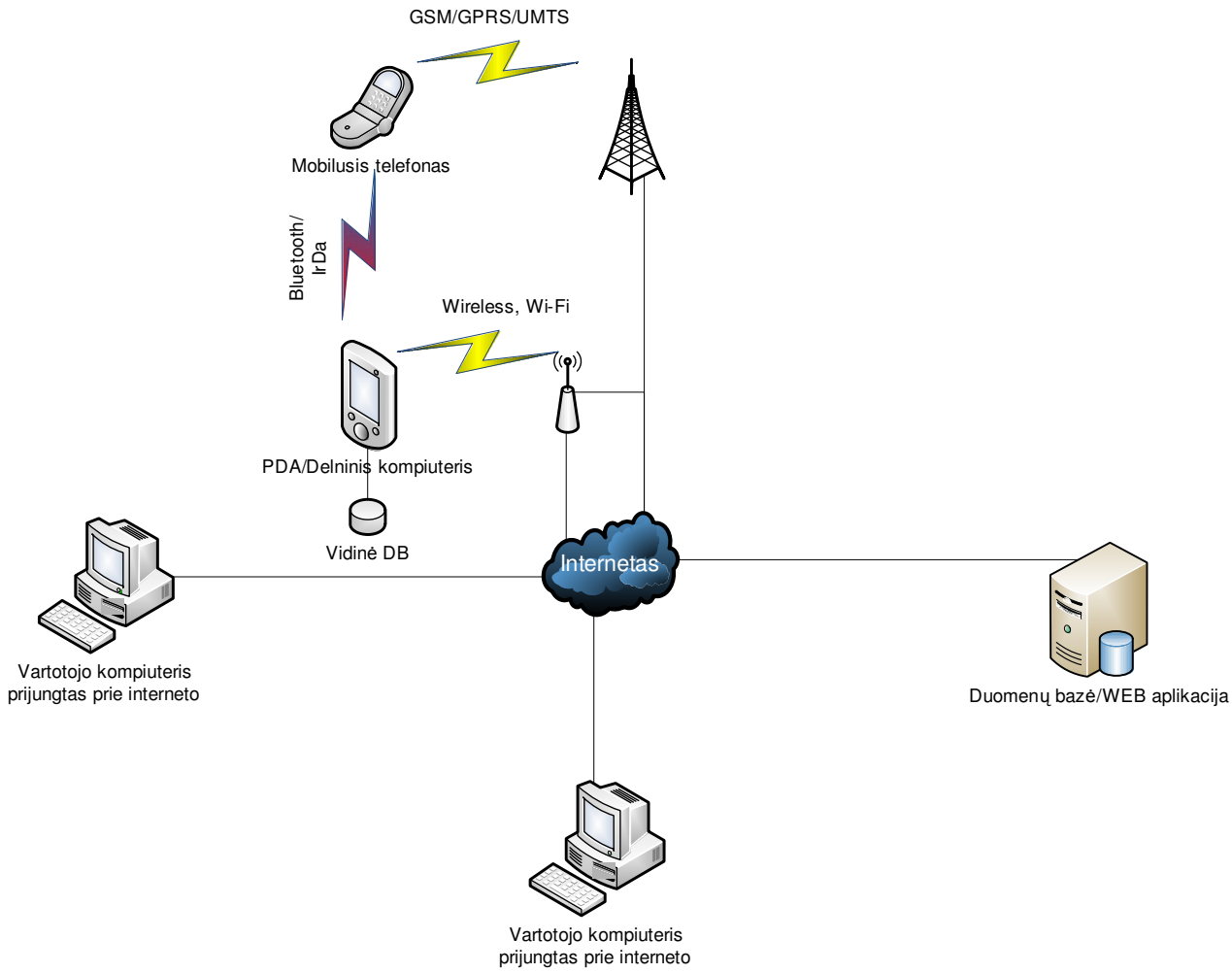
5.2. Funkciniai reikalavimai sistemai

Vienas iš svarbiausių reikalavimų yra, kad būtų galimybė naudotis realizuota programa bet kuriame įrenginyje, kuris gali vykdyti sukompiliuotas J2SE platformoje parašytas programas. Programa gali naudoti tik autorizuoti vartotojai. Autorizuoti vartotojai gali registruoti, bei peržiūrėti jau užregistruotus gedimus. Vartotojai turi turėti galimybę suvesti naujus parametrus į duomenų bazę (nauja gedimo kategorija, nauja gedimo subkategorija, nauji vartotojai ir t.t.).

Sistemos programiniai reikalavimai turi būti tokie, kad sistemoje būtų įdiegta J2SE platforma, o serveryje bet kokia duomenų valdymo sistema.

5.3. Sistemos architektūra

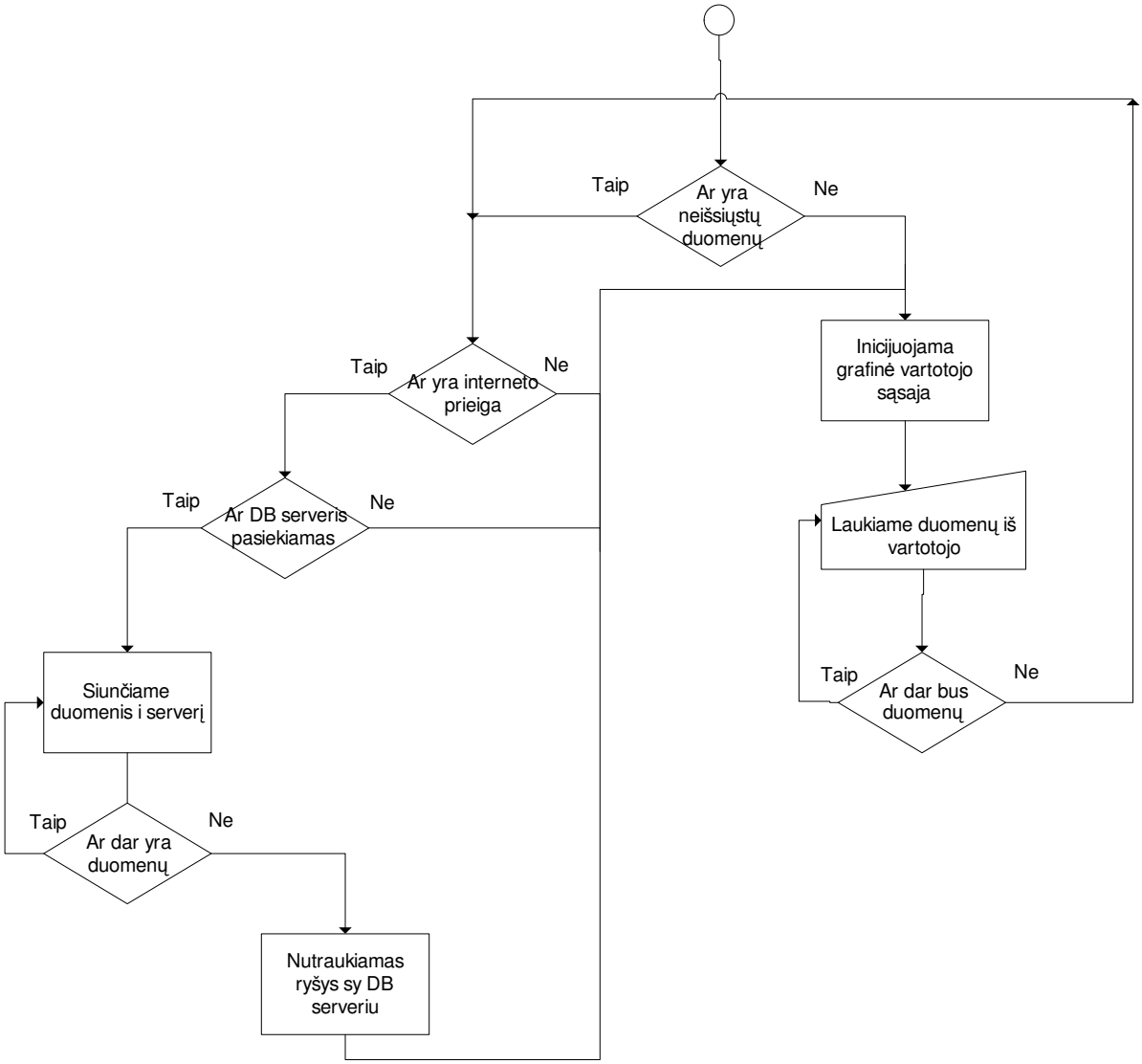
Sistemos architektūra yra paremta kliento – serverio modeliu. Įrenginyje yra vykdoma aplikacija, kur esant interneto ryšiui dirba su nutolusiu duomenų bazių serveriu, o nesant interneto ryšio galimybei, duomenis saugo lokalioje duomenų bazėje. Pasinaudojus JDBC modeliu, serverinė duomenų bazių valdymo sistemą galima rinktis laisvai.



Pav.8 Projekto struktūrinė schema

Duomenys tarp aplikacijos ir serverio perduodami TCP/IP protokolu, kas leidžia sistemą eksploatuoti bet kokiuose kompiuterių tinkluose.

Programos principinę veikimo diagrama atrodo taip:



Pav. 9 Programos veikimo principinė diagrama

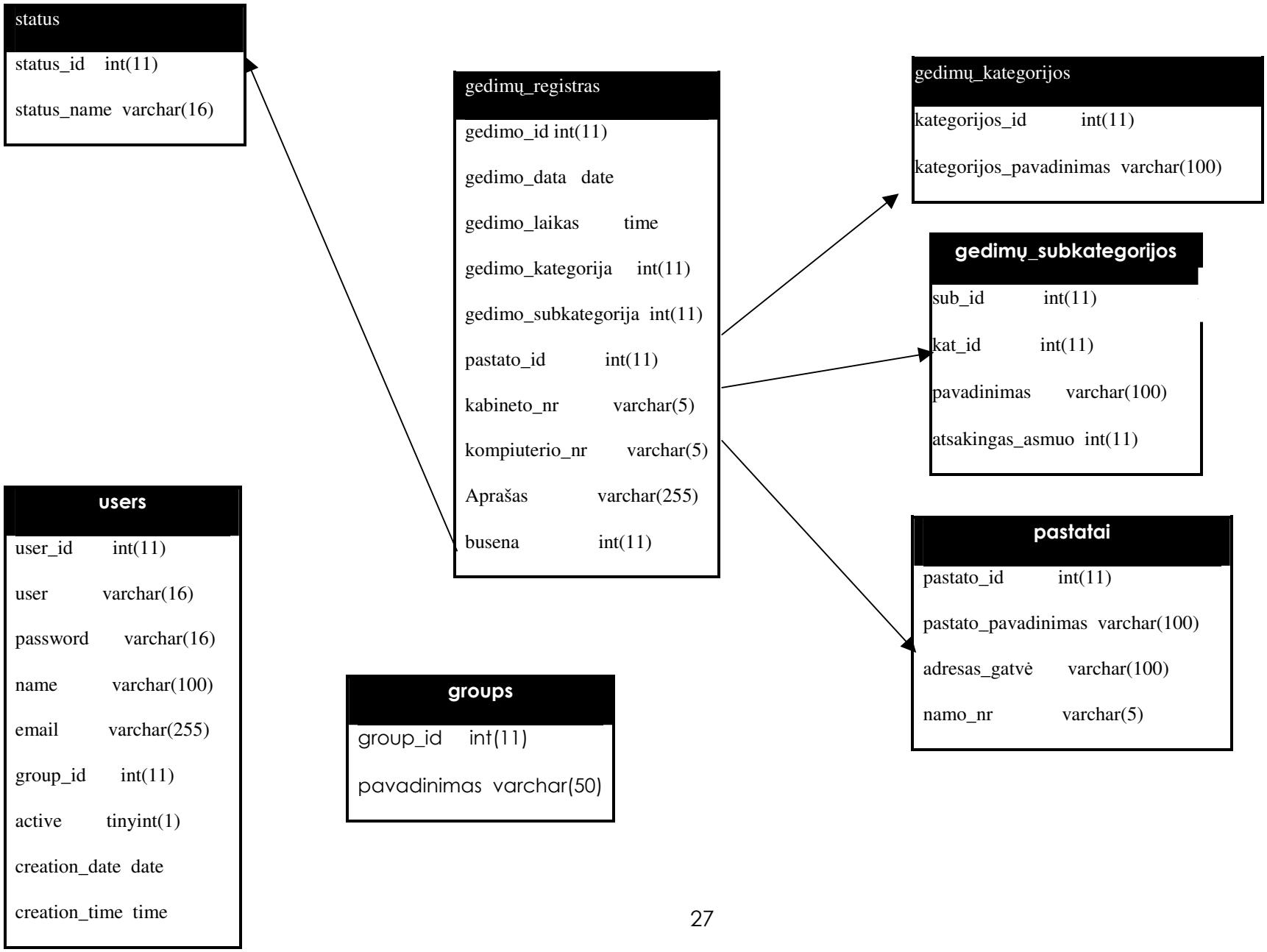
Iš diagramos matome, kad tik paleidus programą, sistema tikrina ar yra neišsiųstų duomenų, jei yra tokių duomenų sistema toliau tikrina ar yra prieiga prie interneto. Jei prieiga yra, sistema sėkmingai išsiunčia duomenis ir pereina į darbo režimą inicijuodama vartotojo sąsają.

5.5. Duomenų bazės realizavimas ir projektavimas

Kadangi jau naudojamos gedimų registravimo sistemos duomenų bazė yra suprojektuota *MySQL* duomenų bazių valdymo sistemos pagrindu, tai ir šiame darbe taip pat yra naudojama

MySQL. Projektavimo patogumo dėlei buvo naudojamas grafinis įrankis *SQLyog*. Šio įrankio pagalba duomenų bazės projektuotojas grafiniu režimu projektuoja duomenų bazę. Tai yra kur kas patogiau ir efektyviau, negu naudojant įprasta komandinės eilutės klientą. Duomenų bazės struktūra pavaizduota paveiksle Nr 10.

Pav. 10 Duomenų bazės struktūrinė schema ir lentelių tarpusavio ryšiai



Duomenų bazės lentelių aprašai

Lentelės pavadinimas	***gedimu_kategorijos***		
Lauko pavadinimas	**Duomenų tipas**	**Pradinė reikšmė**	**Komentaras**
kategorijos_id	int(11)	auto_increment	Laukas naudojamas kaip indeksas surišimui su kitomis lentelėmis
kategorijos_pavadinimas	varchar(100)		Gedimų kategorijos pavadinimas

Lentelės pavadinimas	***gedimu_registras***		
Lauko pavadinimas	**Duomenų tipas**	**Pradinė reikšmė**	**Komentaras**
gedimo_id	int(11)	auto_increment	Laukas naudojamas kaip indeksas surišimui su kitomis lentelėmis
gedimo_data	date	Now()	Gedimo registracijos data
gedimo_laikas	time	Now()	Gedimo registracijos laikas
gedimo_kategorija	int(11)		Imama indekso reikšmė iš *gedimu_kategorijos* lentelės
gedimo_subkategorija	int(11)		Imama indekso reikšmė iš *gedimu_subkategorijos* lentelės
pastato_id	int(11)		Imama indekso reikšmė iš *pastai* lentelės
kabineto_nr	varchar(5)		Kabineto numeris, kur įvyko gedimas
kompiuterio_nr	varchar(5)		Kompiuteris, kur įvyko gedimas

Aprašas	varchar(255)		Gedimo aprašymas
busena	int(11)	1	Imama indekso reikšmė iš *status* lentelės

Lentelės pavadinimas	***gedimu_subkategorijos***		
Lauko pavadinimas	**Duomenų tipas**	**Pradinė reikšmė**	**Komentaras**
sub_id	int(11)	auto_increment	Laukas naudojamas kaip indeksas surišimui su kitomis lentelėmis
kat_id	int(11)		Imama indekso reikšmė iš *gedimu_kategorijos* lentelės
pavadinimas	varchar(100)		Gedimų subkategorijos pavadinimas
atsakingas_asmuo	int(11)		Imama indekso reikšmė iš *users* lentelės

Lentelės pavadinimas	***groups***		
Lauko pavadinimas	**Duomenų tipas**	**Pradinė reikšmė**	**Komentaras**
group_id	int(11)	auto_increment	Laukas naudojamas kaip indeksas surišimui su kitomis lentelėmis
pavadinimas	varchar(100)		Vartotojų grupės pavadinimas

Lentelės pavadinimas	***pastatai***		
Lauko pavadinimas	**Duomenų tipas**	**Pradinė reikšmė**	**Komentaras**
pastato_id	int(11)	auto_increment	Laukas naudojamas kaip indeksas surišimui su kitomis lentelėmis
pastato_pavadinimas	varchar(100)		Pastato pavadinimas
adresas_gatvė	varchar(100)		Gatvės pavadinimas
namo_nr	varchar(5)		Namo numeris

Lentelės pavadinimas	***status***		
Lauko pavadinimas	**Duomenų tipas**	**Pradinė reikšmė**	**Komentaras**
status_id	int(11)	auto_increment	Laukas naudojamas kaip indeksas surišimui su kitomis lentelėmis
status_name	varchar(16)		Gedimo būklės pavadinimas

Lentelės pavadinimas	***users***		
Lauko pavadinimas	**Duomenų tipas**	**Pradinė reikšmė**	**Komentaras**
user_id	int(11)	auto_increment	Laukas naudojamas kaip indeksas surišimui su kitomis lentelėmis
user	varchar(16)		Vartotojo prisijungimo vardas
password	varchar(16)		Vartotojo prisijungimo slaptažodis
name	varchar(100)		Vartotojo tikras vardas, pavardė
email	varchar(255)		Vartotojo el. pašto adresas
group_id	int(11)		Imama indekso reikšmė iš *groups* lentelės vartotojo grupei nurodyti.
active	tinyint(1)	1	Nurodo ar vartotojas aktyvus sistemoje
creation_date	date	Now()	Vartotojo sukūrimo data
creation_time	time	Now()	Vartotojo sukūrimo laikas

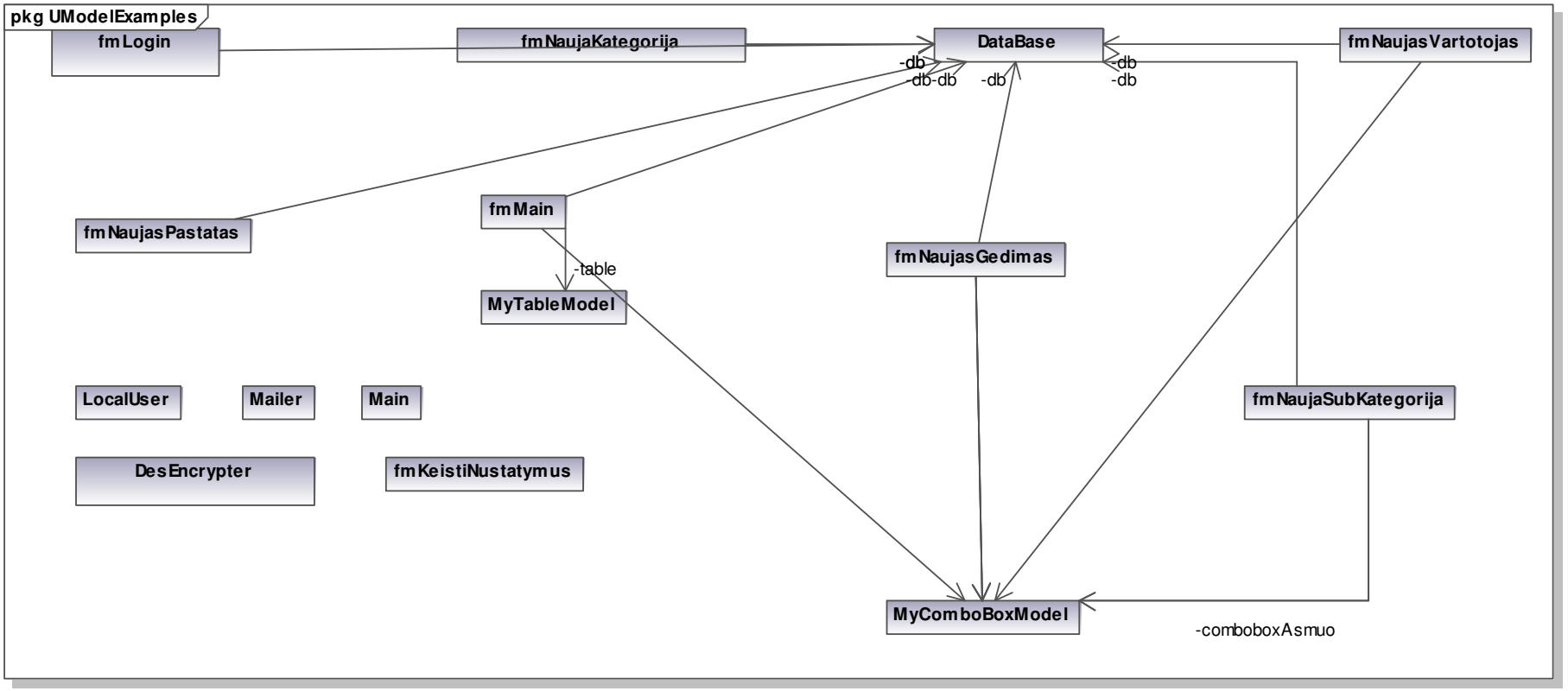
5.6. Aplikacijos projektavimas ir realizavimas

Programą sudaro dvi pagrindinės dalys, tai darbui su duomenų baze specialiai parašytos klasės, bei vartotojo grafinės sąsajos klasės. Grafinė vartotojo sąsaja sukonstruota *Java Swing* komponentų pagrindu, kadangi tai labai turtingas komponentų atžvilgiu bei nepriklausomas nuo operacinės sistemos komponentų rinkinys.

Visų klasių kodas buvo rašomas *Eclipse IDE* pagalba, o klasės projektuojamos *Altanota UModel*.

Startuojant programai, *Main.java* pagrindinėje klasėje vykdomas visas programos parengimas ir vartotojo sąsajos paleidimas. Iš pradžių programa patikrina ar egzistuoja sistemos nustatymo failas, reikalingas tinkamai programos veiklai. Jei šio failo nėra, programa sukuria šį faila su pradinėmis reikšmėmis vartotojo „namų“ direktorijoje. Toliau sistema inicijuoja duomenų bazės klases ir perduoda jų objektus grafinei vartotojo sąsajai. Toliau inicijuojama *fmMain.java* klasė, kuri yra pagrindinė programos forma. Joje vaizduojamas gedimų žurnalas, bei pagrindinis valdymo meniu.

Aplikacijos klasių tarpusavio ryšiai pavaizduoti paveiksle Nr 11.



Pav 11. Programos klasių struktūrinė schema ir tarpusavio ryšiai

Iš šios diagramos matome, kad visos aplikacijos formos susijusios su duomenų valdymų, tiesiogiai sąveikauja su duomenų bazės klase. Diagramoje taip pat matome pagalbines programos klases, tai:

- *LocalUser* klasė, lokalaus vartotojo objekto klasė, naudojama aprašyti lokalų sistemos vartotoją.
- *Mailer* klasė naudojama pašto pranešimui išsiųsti, kai užregistruojamas naujas gedimas.
- *DesEncrypter* klasė naudojama teksto šifravimui.

5.7. Sistemos testavimas

Sistemos testavimui buvo parinkta keletas platformų: delninukas *ASUS MYPAL* su *Microsoft Windows Mobile* 5.0 operacine sistema, i386 architektūros personalinis kompiuteris su *Microsoft Windows XP* operacine sistema bei i386 architektūros kompiuteris su *GNU Linux Knoppix 5.0* operacine sistema.

Testavimų kriterijai tokie:

- Pilnas aplikacijos funkcionalumas.
- Korektiškai vaizduojama grafinė vartotojo sąsaja.
- Aplikacijos veikimo greitis.
- Operatyvios atminties suvartojimas.

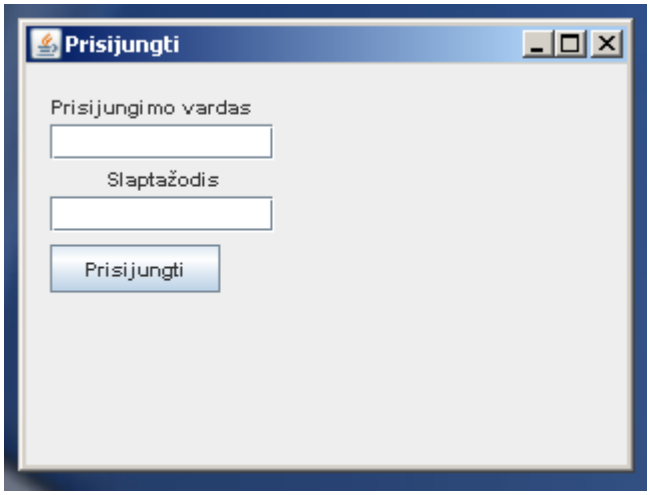
2. lentelė Sistemos testavimas

	ASUS MYPAL **Windows Mobile 5.0, 600 MHz CPU, 64 MB RAM**	**i386 PC Windows XP,** **1400 MHz CPU, 512 MB RAM**	**i386 GNU/Linux Knoppix 5.0, 1400MHz CPU, 512 MB RAM**
Pilnas aplikacijos funkcionalumas	Taip	Taip	Taip
Korektiškai vaizduojama vartotojo sąsaja (diakritiniai ženklai)	Taip	Taip	Taip

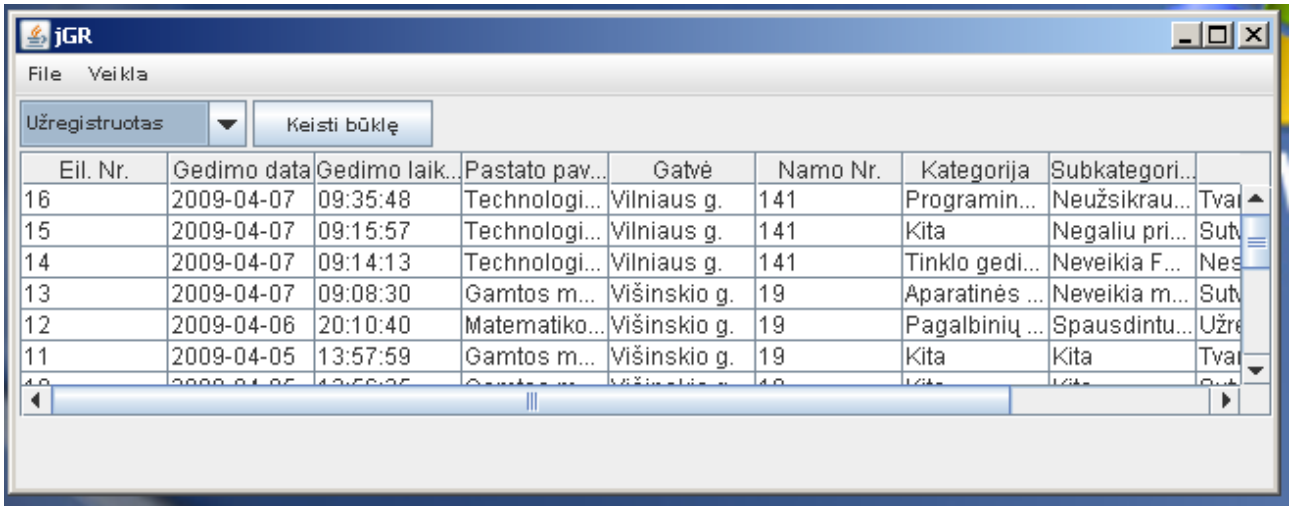
Grafinės vartotojo sąsajos formų sukūrimo bei atvaizdavimo greitis.	~3s	<1s	<1s
RAM sunaudojimas	~10MB	~12MB	~12MB
Kompiuterinio tinklo sparta	iki 384 Kb/s	iki 2Mb/s	iki 2 Mb/s

Visose platformose aplikacijos funkcionalumas išliko korektiškas. Programa „elgėsi“ kaip ir buvo numanyta. Visos procedūros ir funkcijos tinkamai atliko skaičiavimų rezultatus.

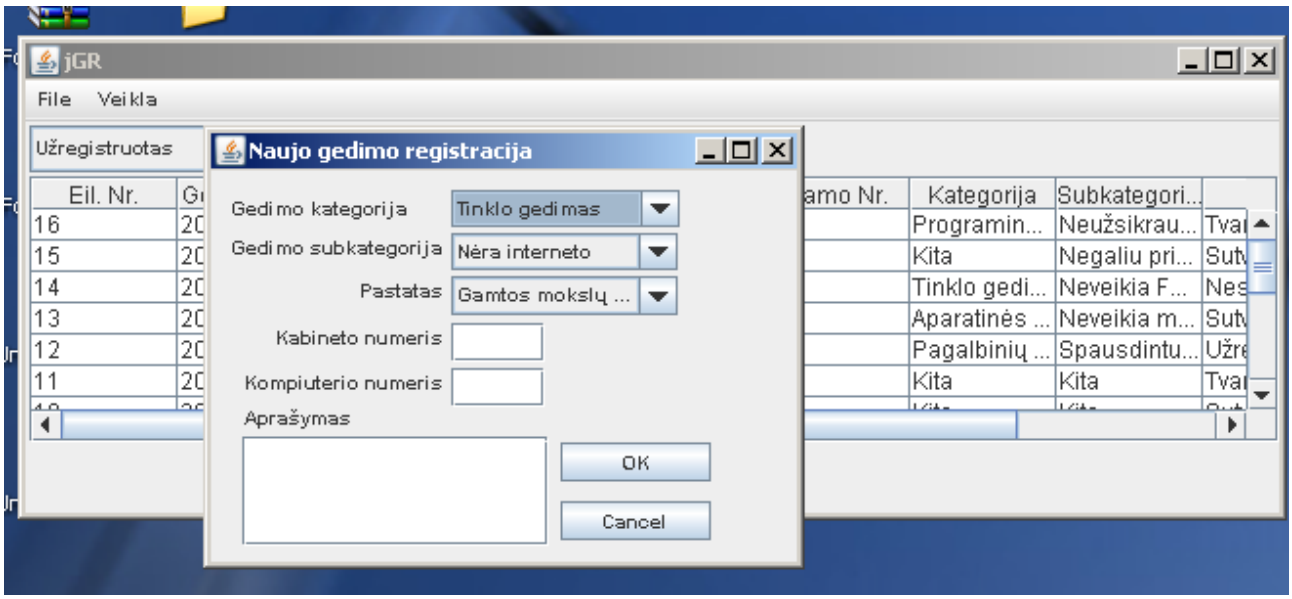
Kadangi projektuojant aplikaciją, buvo parinkta *Java Swing* komponentų posistemė, aplikacija įvairiuose platformose atrodė vienodai bei korektiškai.



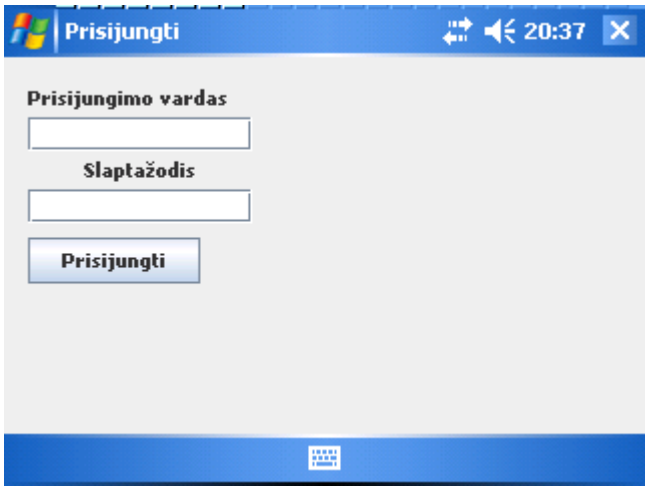
Pav. 12 Programos prisijungimo langas *Windows XP* aplinkoje



Pav. 13 Pagrindinis programos langas *Windows XP* aplinkoje

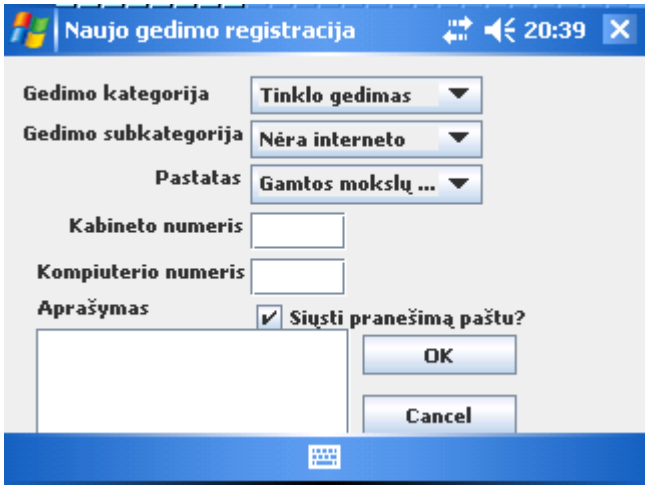


Pav. 14 Pagrindinis langas ir naujo gedimo registravimo dialogas *Windows XP* aplinkoje

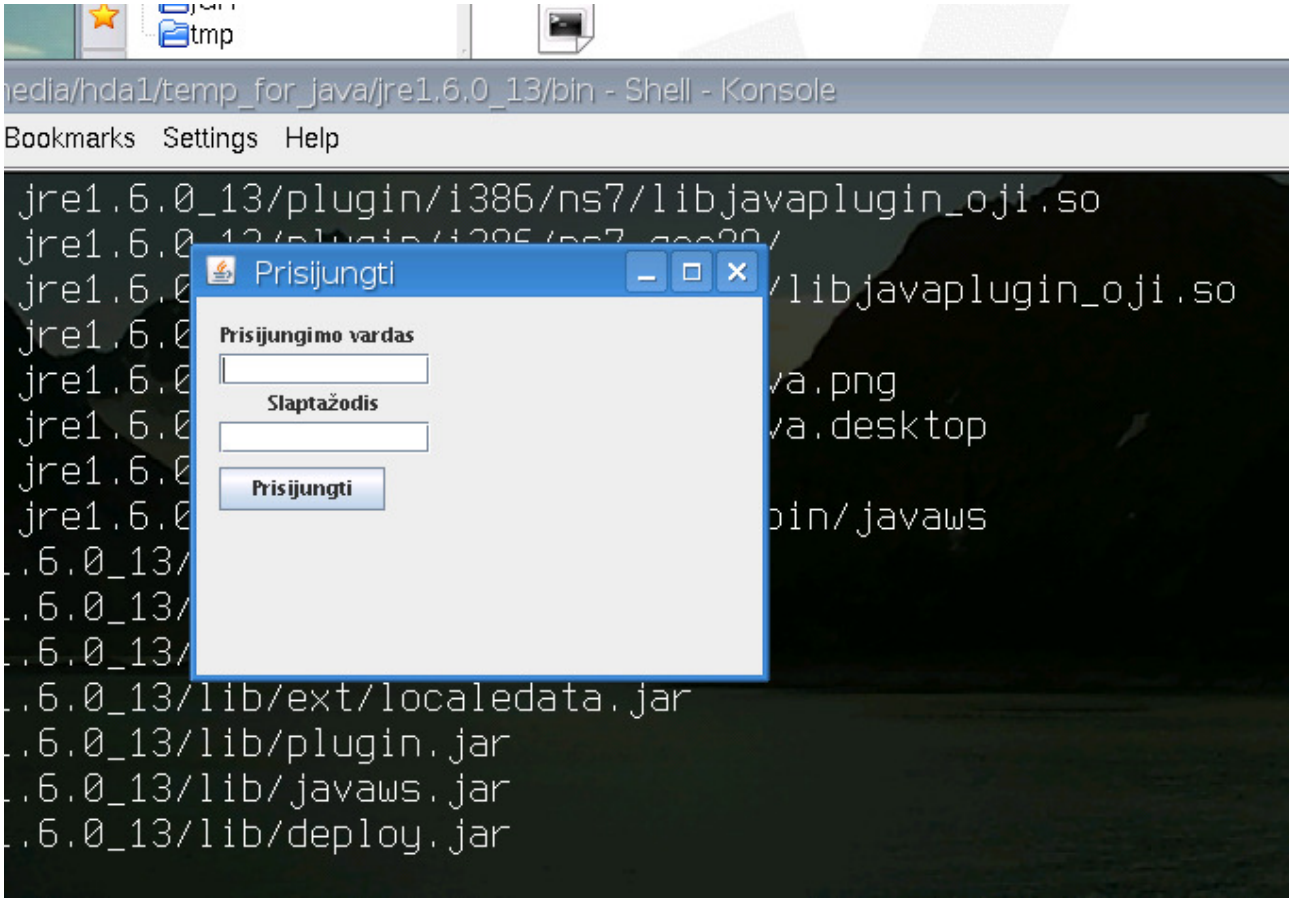


Pav. 15 Programos prisijungimo langas *Windows Mobile 5.0* aplinkoje

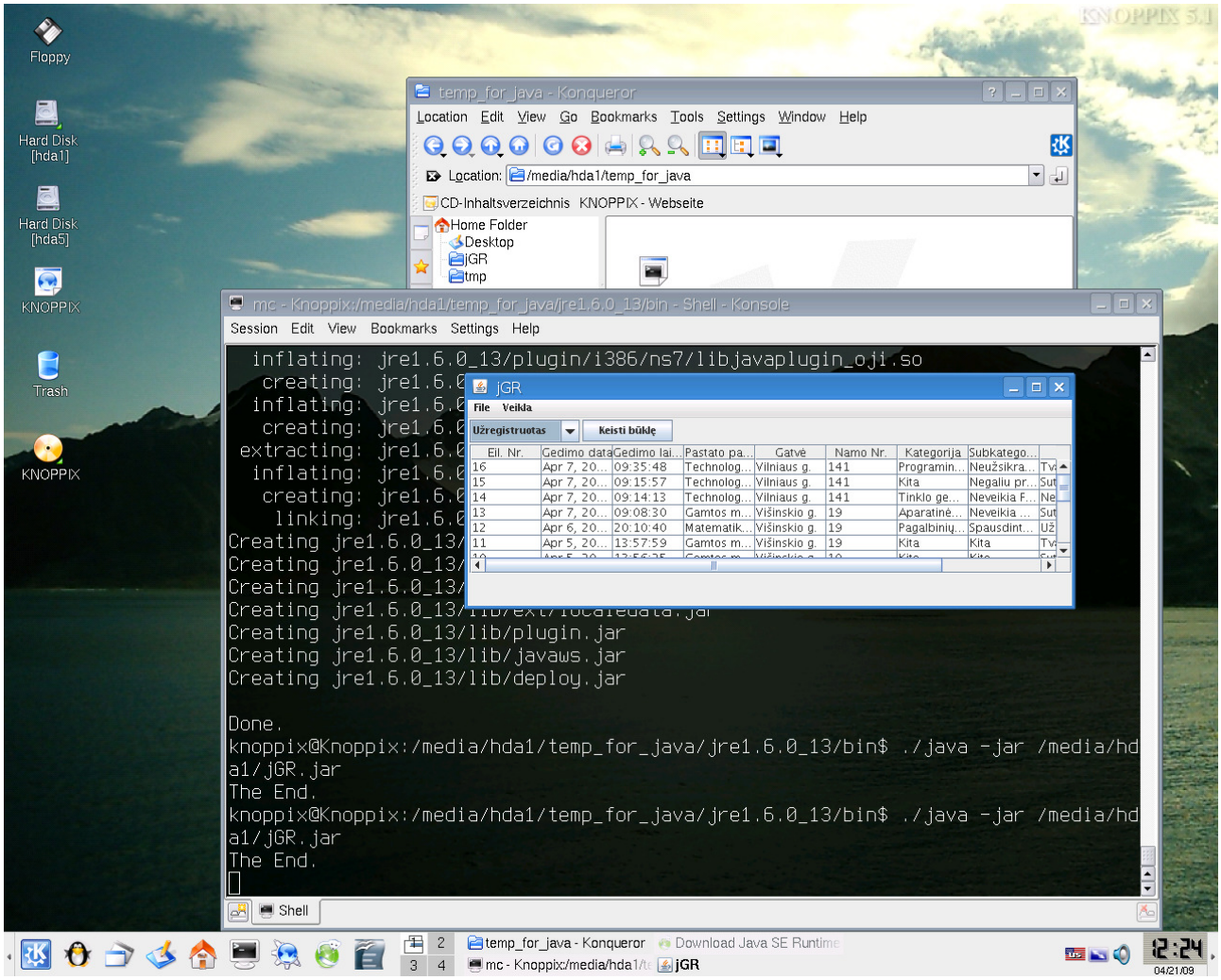
Pav. 16 Pagrindinis programos langas *Windows Mobile 5.0* aplinkoje



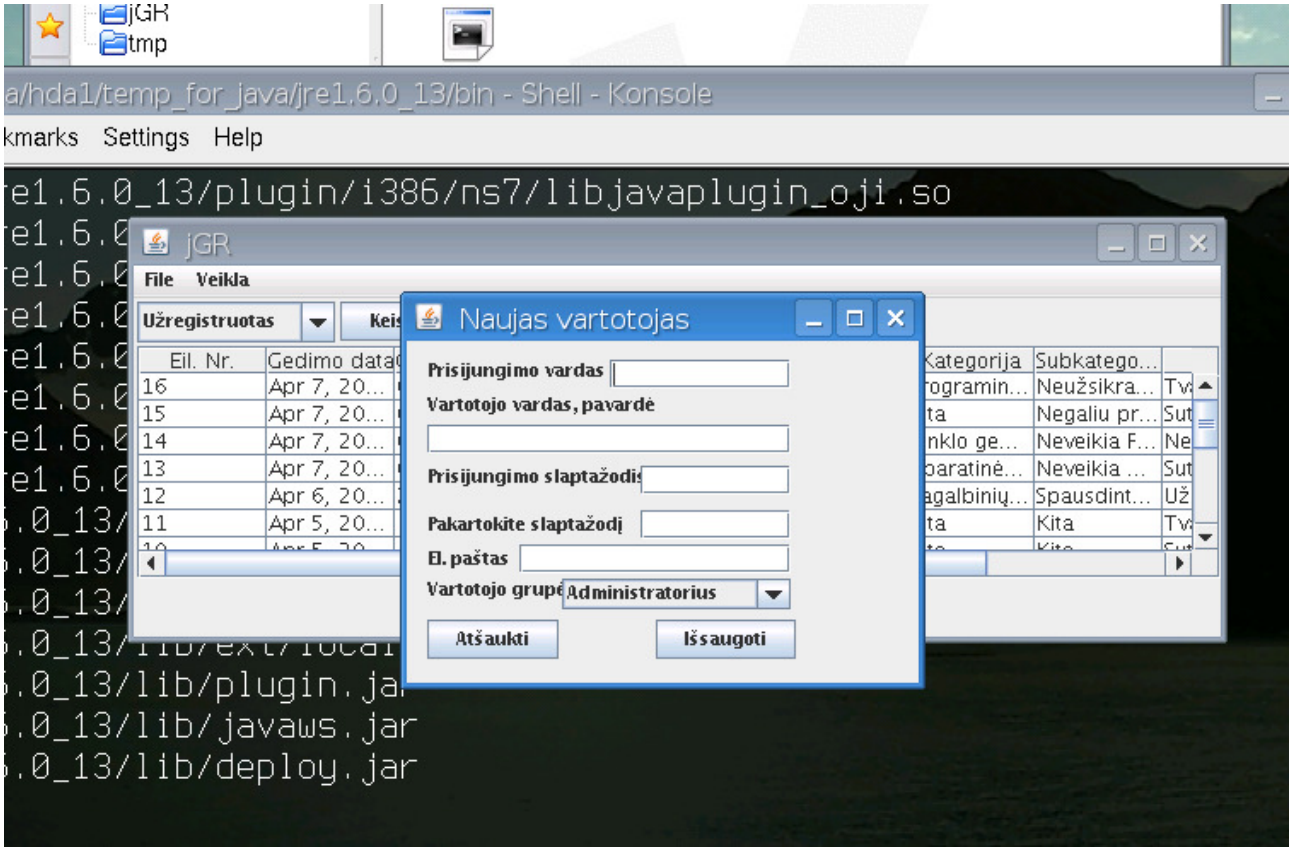
Pav. 17 Naujo gedimo registravimo dialogas *Windows Mobile 5.0* aplinkoje



Pav. 18 Programos prisijungimo langas *GNU Linux KDE 3.*5 aplinkoje



Pav. 19 Pagrindinis programos langas *GNU Linux KDE 3*.5 aplinkoje



Pav. 20 Pagrindinis programos langas ir naujo vartotojo registracijos dialogas *GNU Linux KDE 3*.5 aplinkoje

Iš pateiktų paveikslėlių matome, kad jokioje platformoje nėra problemų su lietuviškų simbolių vaizdavimu, nors lokalės buvo skirtingos. Windows XP sistemoje buvo nustatyta lietuviška lokalė, o likusiuose platformose JAV lokalė. Aplikacija naudoja UTF-8 simbolių koduotę, todėl nepriklausomai nuo kompiuterio kalbos nustatymų, programos sąsaja išlieka korektiška.

Programos veikimo greitis personaliniuose kompiuteriuose buvo idealus, tik delniniame kompiuteryje dėl mažoko centrinio procesoriaus dažnio jautėsi tam tikri sulėtėjimai.

Programa visuose platformose naudojo apie 10MB operatyvinės atminties, tad lyginant su šiuolaikiniais kompiuterių bei PDA resursais tai nebuvo didelis suvartojimas.

5.8. Problemų analizė ir jų sprendimas

Realizuojant gedimų registravimo programą kilo tam tikrų techninių problemų, kurias toliau ir apžvelgsime.

1. Visų pirma ir pati svarbiausia problema, tai *Java2SE* vykdomosios posistemės nebuvimas iš oficialiųjų platintojų (*Microsoft* arba *Sun Microsystems*). Nei viena iš šių bendrovių neplatina *Java2SE* posistemės *Windows Mobile* platformai. Be šios posistemės neįmanoma paleisti nei vienos programos sukompiliuotos į *Java bytecode*. Dėka *GNU ClassPath* bei *Maisafu Java* projektų pavyko adaptuoti *Java2SE* posistemę delniniams kompiuteriams bei *Windows Mobile* operacinėms sistemoms.

Mysaifu JVM yra skirta vykdyti *java byte* kodą *Windows Mobile* operacinėje sistemoje. Palaikomos šios operacinės sistemos versijos:

- Windows Mobile 6.0.
- Windows Mobile 5.0.
- Windows Mobile 2003 Second Edition software for Pocket PC (Pocket PC 2003 SE).
- Windows Mobile 2003 software for Pocket PC (Pocket PC 2003).

Ši Java aplinka yra sukurta GNU Classpath bibliotekų pagrindu. GNU Classpath bibliotekos yra nuolat aktyviai tobulinamos ir palaikomos daugumos GNU bendruomenės programuotojų. Paskutinės GNU Classpath bibliotekų versijos pilnai tenkina JDK 1.4 JDK 1.5 platformas. Tad visi norintys perkelti *Java* aplinką į kitas operacinių sistemų platformas gali laisvai naudotis šiomis bibliotekomis, nes jos yra išleistos pagal *GNU General Public License* licenziją.

2. Sekanti problema buvo išplaukianti iš pirmosios. Kadangi *Maisafu Java* nėra oficialus produktas ir nėra sertifikuotas *Sun Microsystems* bendrovės, joje trūksta naujausių bibliotekų, kurios jau yra įprastai platinamos su oficialiosiomis versijomis. Konkrečiai šiuo atveju trūko *Swing* vizualių komponentų išdėstymo bibliotekų. Ši problema išspręsta taip, į pačią aplikaciją buvo „įskiepytos“ šios klasės atskiros bibliotekos pagrindu.

3. Trečia problema tai duomenų atvaizdavimas grafinėje vartotojo sąsajoje. *Java Swing* komponentų rinkinys neturi vizualiųjų komponentų, gebančių tiesiogiai atvaizduoti duomenis iš duomenų bazių objektų. Ši problema buvo išspręsta papildomų klasių pagalba. Specialiai šiai aplikacijai buvo parašytos dvi klasės: MyTableModel.java bei MyComboBoxModel.java. Šiuos klasės atlieka duomenų tarpininko vaidmenį – skaito duomenis iš duomenų bazės objektų ir tinkama struktūra perduoda į vizualiuosius komponentus.

5.9. Galutinis modelio apibendrinimas

Tyrime buvo realizuotos visos programos dalys, numatytos iškeltuose darbo tiksluose. Programa tiek delniniame kompiuteryje tiek kitose kompiuterių architektūrose veikė kaip ir numatyta projekte, su tam tikrais trūkumais (sąlyginai lėtas vykdymas dėl lėtesnių procesorių). Toliau tobulinant sistemą, reiktų atkreipti dėmesį į programos greitaveikos didinimą, bei papildomų funkcinių galimybių praplėtimą.

Pilnai ištestuotą ir ištobulintą programą būtų galima panaudoti kaip papildymą esamai kompiuterinių sistemų gedimų registravimo sistemai Šiaulių Universitete.

6. Darbo išvados

Išanalizavus delninių kompiuterių technologines galimybes, programų kūrimo priemones bei naudojamas operacines sistemas delniniuose kompiuteriuose padarytos išvados:

1. Naudojantis šiuolaikinėmis aplikacijų projektavimo bei kūrimo technologijomis, buvo sukurta pilnavertė kompiuterinių sistemų gedimų registravimo programa delniniams kompiuteriams. Ši aplikacija gali papildyti ar netgi pakeisti jau esamas ir eksploatuojamas sistemas.

2. PDA įrenginių aparatinės bei programinės charakteristikos, leidžia naudoti *Java2SE* technologijas.

3. Pasinaudojus *Java2SE* platforma, išnyksta skirtumas tarp stacionariųjų ir delninių kompiuterių, programos sėkmingai veikia daugelyje operacinių sistemų.

4. Bevielio interneto teikiamų paslaugų greičių per *GPRS, UMTS, HSPD* visiškai pakanka sėkmingam prisijungimui prie nutolusių duomenų bazių.

7. Literatūra ir informacijos šaltiniai

1. PDA technologinių duomenų svetainė [žiūrėta 2008-04-09]. Prieiga per internetą: http://pdadb.net
2. *Sun microsystems Java™* technologijų svetainė [žiūrėta 2008-04-09]. Prieiga per internetą: http://java.sun.com
3. *Sun microsystems* internetinė svetainė: J2SE platformos apžvalga [žiūrėta 2008-04-10]. Prieiga per internetą: http://java.sun.com/javase/
4. *Sun microsystems* internetinė svetainė: J2EE platformos apžvalga [žiūrėta 2008-04-10]. Prieiga per internetą: http://java.sun.com/javaee/
5. *Sun microsystems* internetinė svetainė: J2ME platformos apžvalga [žiūrėta 2008-04-10]. Prieiga per internetą: http://java.sun.com/javame/
6. *Microsoft Windows Mobile* svetainė: *Windows Mobile* apžvalga [žiūrėta 2008-04-12]. Prieiga per internetą: http://www.microsoft.com/windowsmobile
7. *Garnet OS(PalmOS)* svetainė [žiūrėta 2008-04-12]. Prieiga per internetą: http://www.access-company.com/products/platforms/garnet/index.html
8. Palm™ delninių kompiuterių informacinė svetainė [žiūrėta 2008-04-12]. Prieiga per internetą: http://www.palminfocenter.com
9. *Maisafu Java* internetinė svetainė: *Java aplinka Windows Mobile* platformai [žiūrėta 2008-05-13]. Prieiga per internetą: http://www2s.biglobe.ne.jp/~dat/java/project/jvm/index_en.html
10. *GNU Classpath* internetinė svetainė: *GNU Java* bibliotekos [žiūrėta 2008-05-13]. Prieiga per internetą: http://www.gnu.org/software/classpath/classpath.html
11. *NetBeans* internetinė svetainė: *NetBeans* integruota programavimo aplinka [žiūrėta 2008-05-28]. Prieiga per internetą: http://www.netbeans.org/
12. *Eclipse IDE* internetinė svetainė: E*clipse IDE* integruota programavimo aplinka [žiūrėta 2008-05-28]. Prieiga per internetą: http://www.eclipse.org/
13. *Eclipse Visual Editor* internetinė svetainė: *Eclipse* grafinis vartotojo sąsajos dizaineris [žiūrėta 2008-06-15]. Prieiga per internetą: http://www.eclipse.org/vep/WebContent/main.php
14. *BEA Workshop for WebLogic* internetinė svetainė [žiūrėta 2008-06-15]. Prieiga per internetą: http://www.oracle.com/technology/software/products/ias/bea_main.html
15. *BlueJ* internetinė svetainė: *BlueJ* programavimo aplinka [žiūrėta 2008-09-10]. Prieiga per internetą: http://bluej.org
16. *DrJava* internetinė svetainė: *DrJava* programavimo aplinka [žiūrėta 2008-09-11]. Prieiga per internetą: http://www.drjava.org
17. *Geany* internetinė svetainė: *Geany* programavimo aplinka [žiūrėta 2008-09-13]. Prieiga per internetą: http://www.geany.org

18. *Greenfoot* internetinė svetainė: *Greenfoot* programavimo aplinka [žiūrėta 2008-10-08]. Prieiga per internetą: http://www.greenfoot.org

19. *IntelliJ IDEA* internetinė svetainė: *IntelliJ IDEA* programavimo aplinka [žiūrėta 2008-10-08]. Prieiga per internetą: http://www.jetbrains.com/idea

20. *Jbuilder* internetinė svetainė: *Jbuilder* programavimo aplinka [žiūrėta 2008-10-08]. Prieiga per internetą: http://www.codegear.com/Products/JBuilder

21. *Jcreator* internetinė svetainė: *Jcreator* programavimo aplinka [žiūrėta 2008-11-06]. Prieiga per internetą: http://www.jcreator.com

22. *Jdevelop* internetinė svetainė: *Jdevelop* programavimo aplinka [žiūrėta 2008-11-06]. Prieiga per internetą: *www.oracle.com/technology/products/jdev*

23. *Kdevelop* internetinė svetainė: *Kdevelop* programavimo aplinka [žiūrėta 2008-11-06]. Prieiga per internetą: *www.kdevelop.org/*

24. *DB2* internetinė svetainė: *DB2* duomenų bazių valdymo sistema [žiūrėta 2009-01-13]. Prieiga per internetą: *www.ibm.com/db2*

25. *Firebird* internetinė svetainė: *Firebird* duomenų bazių valdymo sistema [žiūrėta 2009-01-13]. Prieiga per internetą: *www.firebirdsql.org*

26. *Informix* internetinė svetainė: *Informix* duomenų bazių valdymo sistema [žiūrėta 2009-01-13]. Prieiga per internetą: *www.ibm.com/software/data/informix*

27. *Microsoft SQL Server* internetinė svetainė: *Microsoft SQL Server* duomenų bazių valdymo sistema [žiūrėta 2009-02-02]. Prieiga per internetą: *www.microsoft.com/SQL/default.mspx*

28. *MySQL* internetinė svetainė: *MySQL* duomenų bazių valdymo sistema [žiūrėta 2008-12-20]. Prieiga per internetą: *www.mysql.com*

29. *Oracle* internetinė svetainė: *Oracle DB* duomenų bazių valdymo sistema [žiūrėta 2008-12-20]. Prieiga per internetą: *www.oracle.com/database/index.html*

30. *PostgreSQL* internetinė svetainė: *PostgreSQL* duomenų bazių valdymo sistema [žiūrėta 2008-12-20]. Prieiga per internetą: *www.postgresql.org*

31. *SQLite* internetinė svetainė: *SQLite* duomenų bazių valdymo sistema [žiūrėta 2009-02-21]. Prieiga per internetą: *www.sqlite.org*

32. *Altanova* internetinė svetainė: *Altanova Umodel* modeliavimno sistema [žiūrėta 2009-04-28]. Prieiga per internetą: *www.altova.com*

33. *Дюбуа, Поль, MySQL*, Издательский дом "Вильямс", 2001. ISBN 5-8459-0158-8.

34. *Justin Couch and Daniel H. Steinberg, Java 2 Enterprise Edition Bible*, New York, NY 10022

35. *Todd M. Thomas, Java Data Access—JDBC, JNDI, and JAXP*, New York, NY 10022

36. *Jonathan B. Knudsen, Java Cryptography*, O'REILLY ISBN: 1-56592-402-9 1998

37. *Aarne Klemetti, PDA Operating Systems*, EVTEK, Media Technology.

38. *Charles Consel Laurent Réveillère A Programmable Client-Server Mode*l ENSEIRB 1, avenue du docteur Albert Schweitzer,Domaine universitaire - BP 99

8. Anotacija

Įvairi skaičiavimo technika sparčiai veržiasi į mūsų kasdieninį gyvenimą, ne taip ir jau seniai kompiuteriais naudojosi tik mokslininkai. Šiandien kompiuterius galime rasti beveik pas kiekvieną žmogų. Ne išimtis ir įvairūs nešiojami įrenginiai, dėl mažų gabaritų bei funkcinių galimybių, prilygstančių įprastiems kompiuteriams, besiveržiantys į daugelio vadybininkų, projektų vadovų, ar techninių darbuotojų kasdieninį darbą. Taigi, mobiliosios technologijos tampa neatsiejama buities, pramogų, verslo, mokslo, politikos dalimi ir Lietuvoje.

Šiame darbe buvo analizuojamos bei projektuojamos galimybės, kuo labiau adaptuoti kompiuterinių sistemų gedimų registravimo programinį produktą delniniams kompiuteriams ir visa tai pritaikyti kasdieniniame darbe.

Tyrime išanalizuotos delninių kompiuterių operacinės sistemos ir jų ypatybės, programų kūrimo įrankiai, duomenų bazių panaudojimo galimybės.

9. Summary

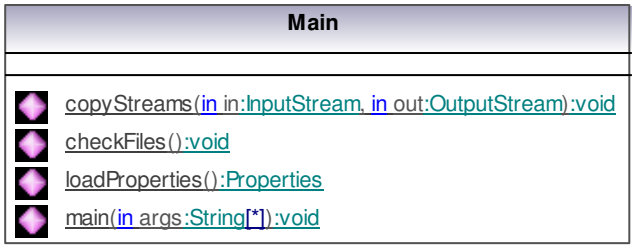
Various computing systems quickly forces its way into our daily lives, not so long ago and only used by computer scientists. Today, computers can be found in almost every human being. No exception, and various portable devices, the small-size and functionality equivalent to conventional computers, coming to many managers, project managers, technical staff, or routine work. Therefore, mobile technology is an integral part of household appliances, entertainment, business, science, policy, and in Lithuania to.

This work was analyzed and designed opportunities, as much as possible to adapt computer systems failure recording software products, and handheld computers all of which adapt everyday work.

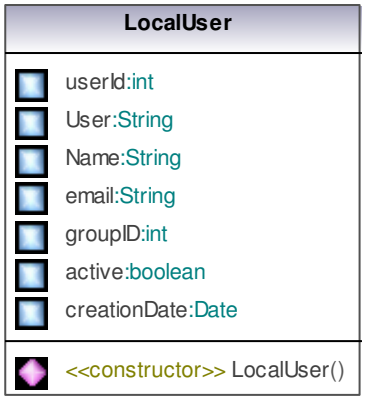
The study analyzed the handheld computer operating systems and their characteristics, application development tools, database options.

10. Priedai

10.1 Klasių diagramos

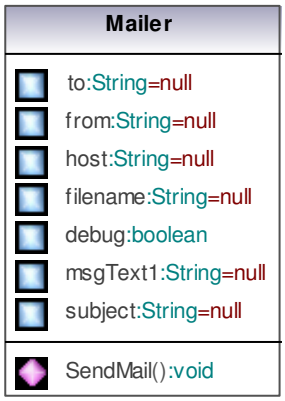


Pav. 21 *Main* klasė, programos inicijavimas ir paleidimas *Main.java*

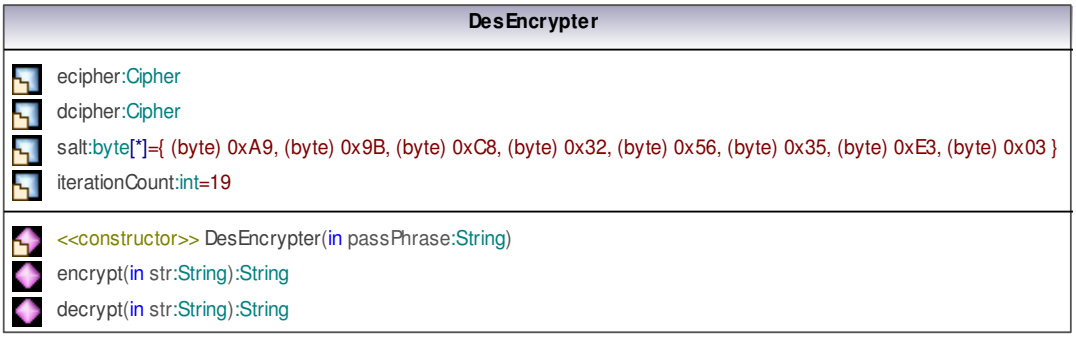


Pav. 22 Lokalaus vartotojo klasė *LocalUser.java*

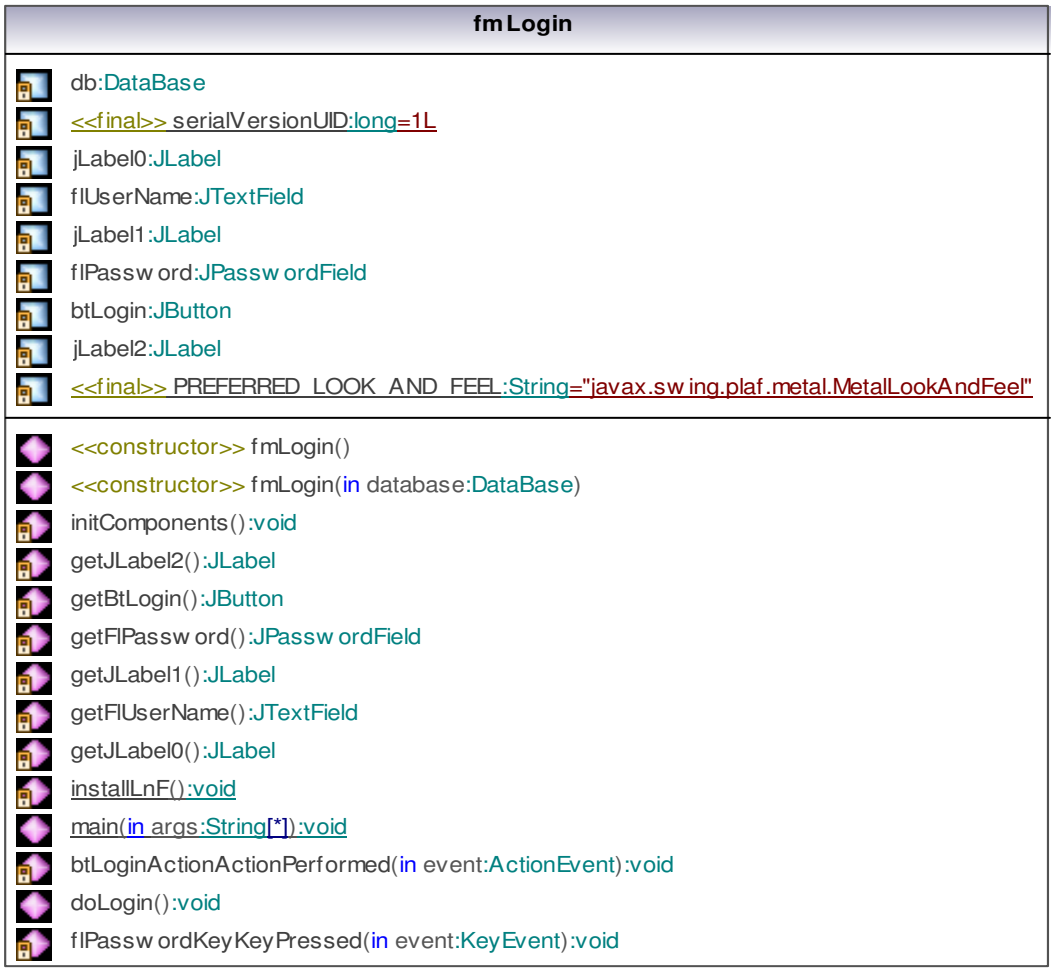
Pav. 23 Duomenų bazės klasė D*ataBase.java*



Pav. 24 Pašto siuntimo per SMTP protokolą klasė *Mailer.java*

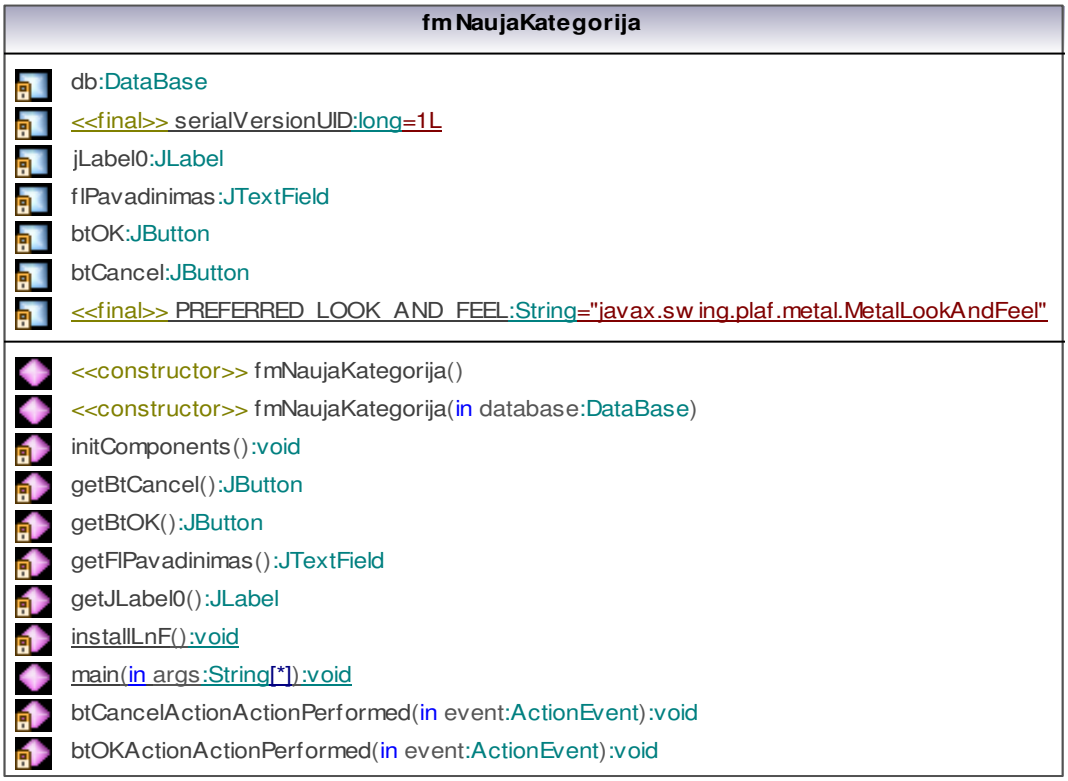


Pav. 25 Teksto šifravimo klasė *DesEcrypter.java*



Pav. 26 Prisijungimo formos klasė *fmLogin.java*

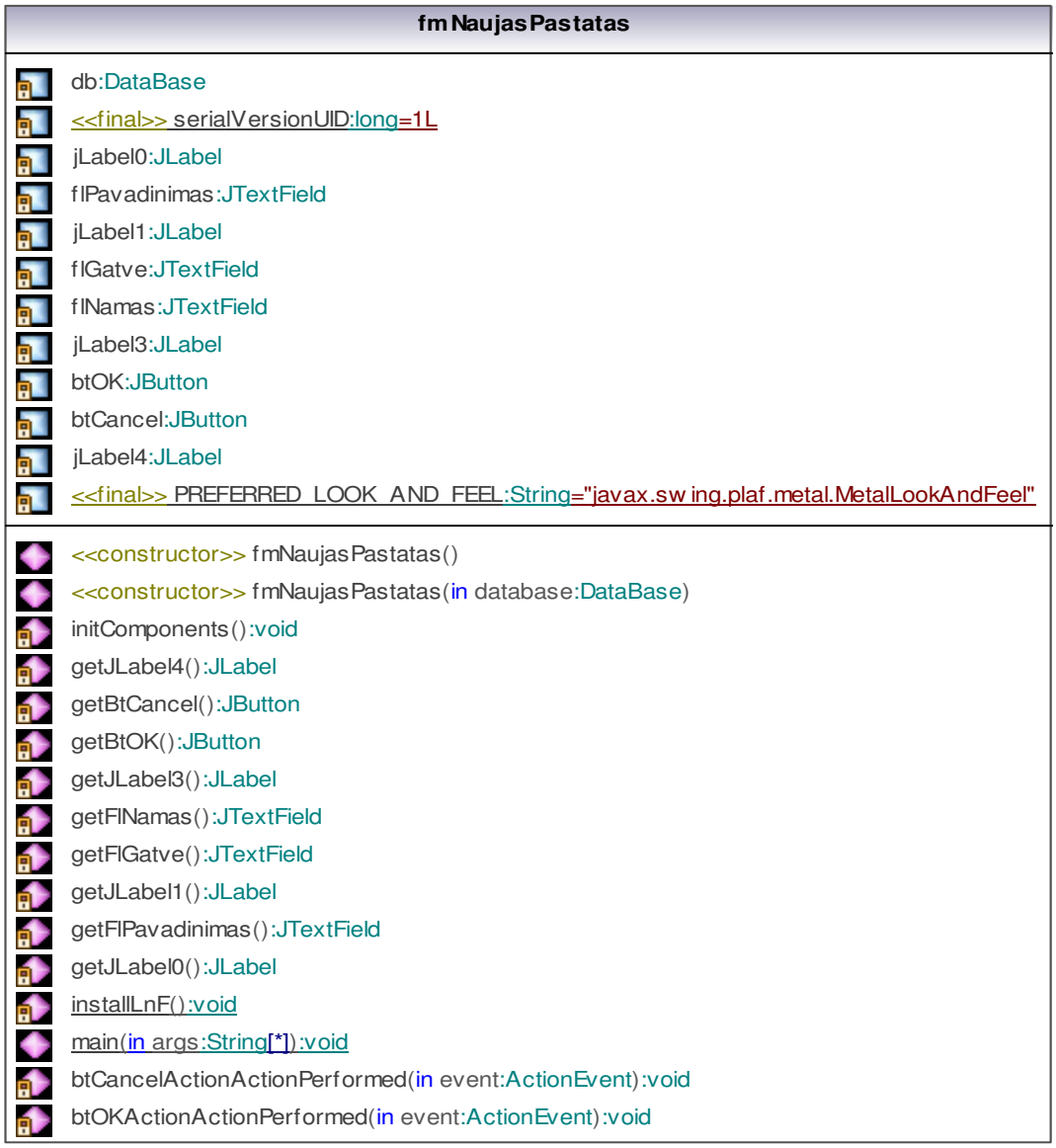
Pav. 27 Programos pagrindinės formos klasė *fmMain.java*



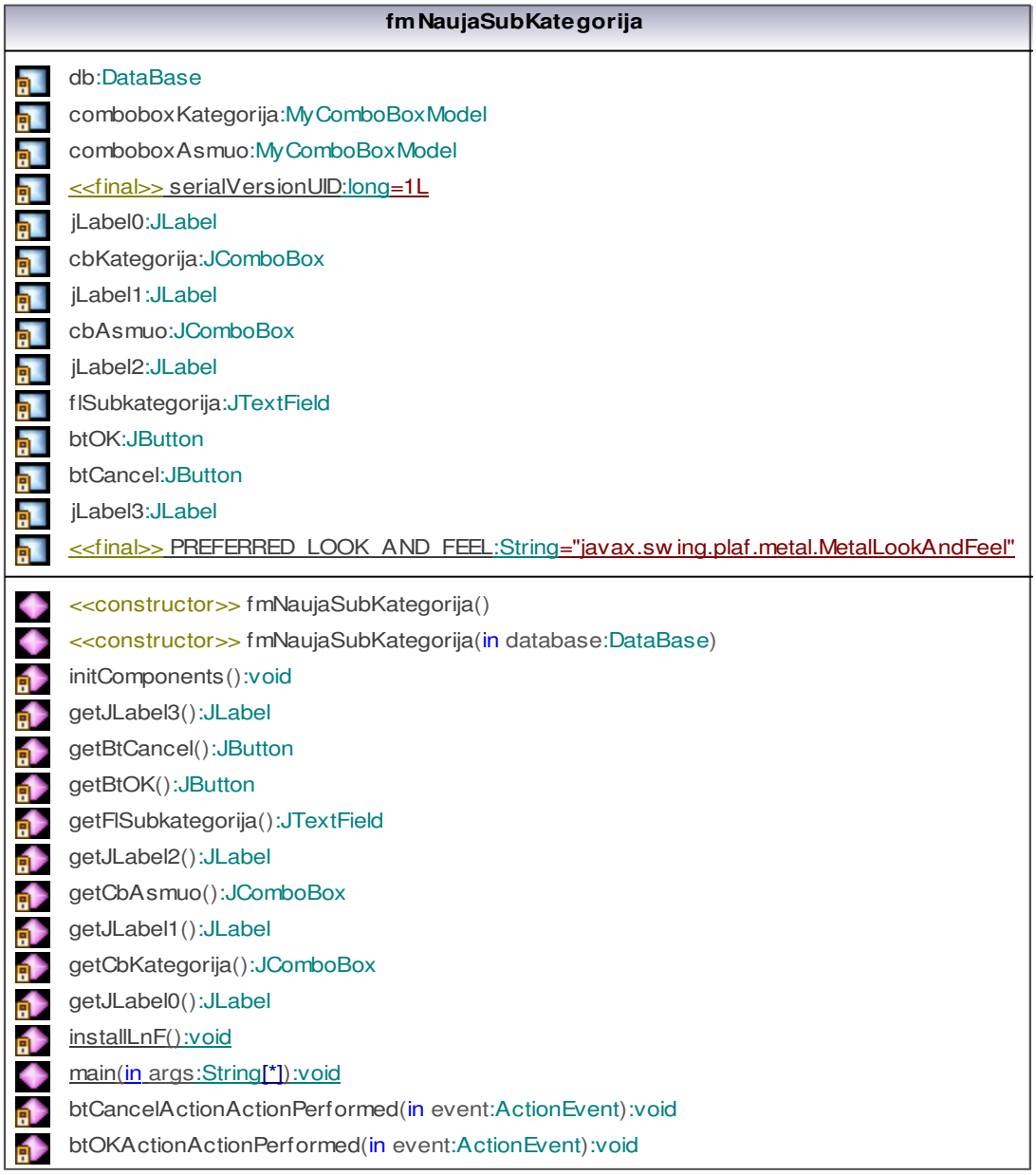
Pav. 28 Naujos kategorijos formos klasė *fmNaujaKategorija.java*

Pav. 29 Programos nustatymų keitimo formos klasė

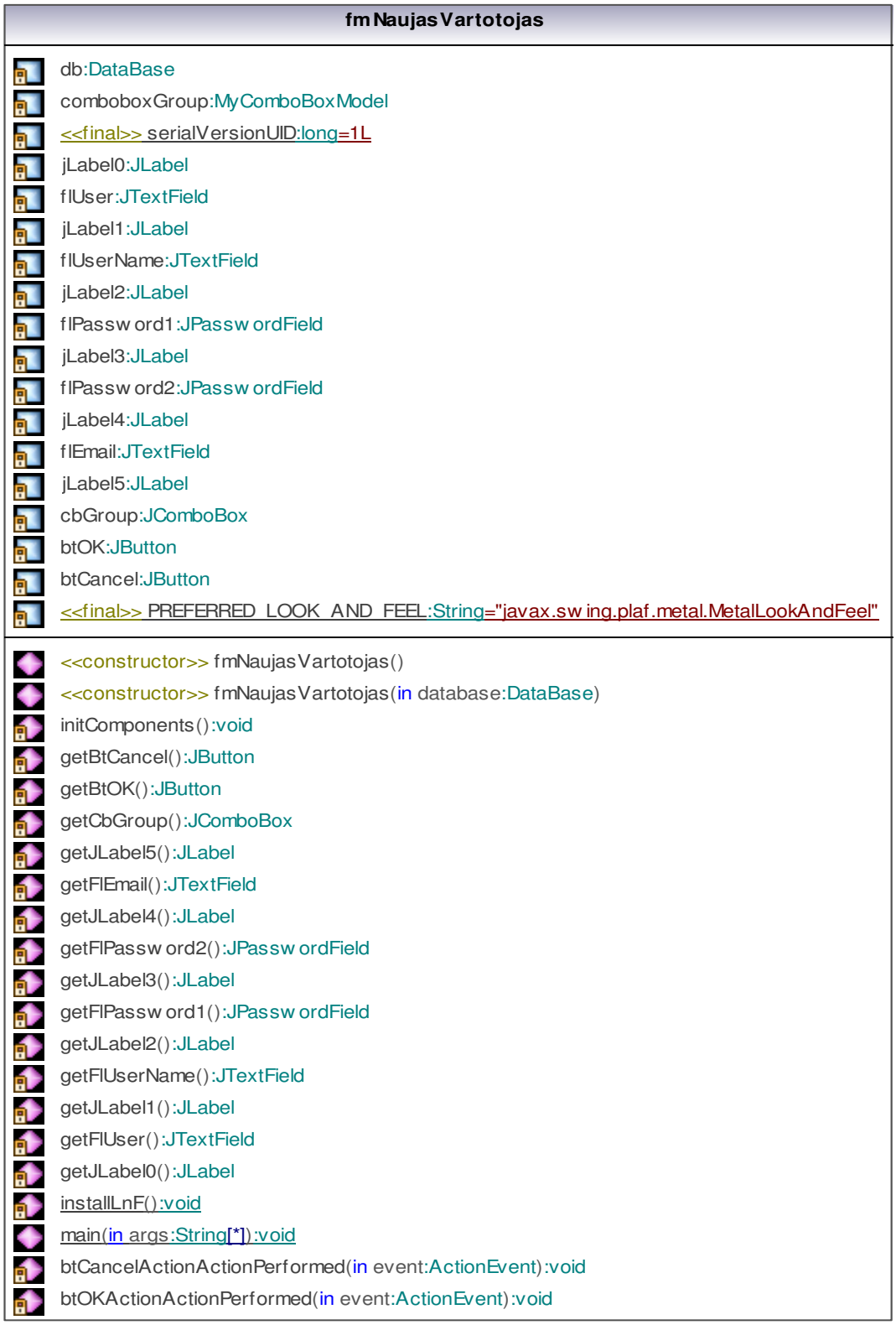
Pav. 30 Naujo gedimo registracijos formos klasė



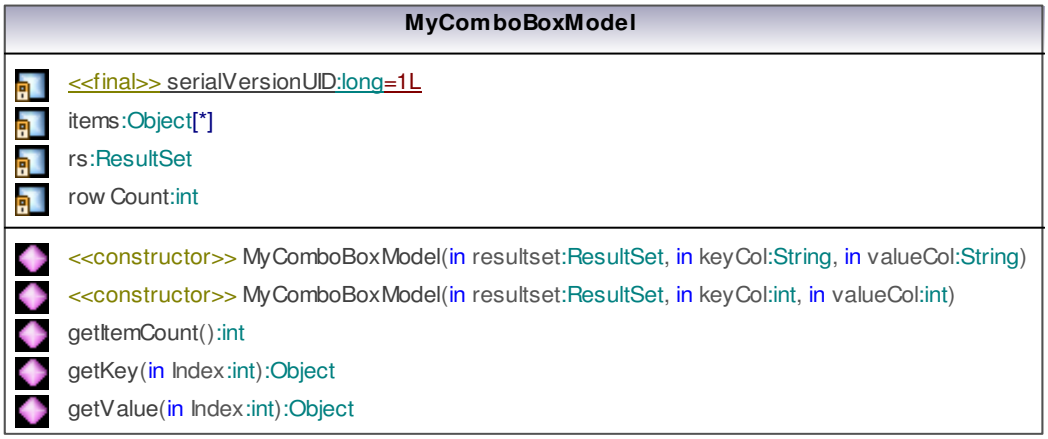
Pav. 31 Naujo pastato kūrimo formos klasė



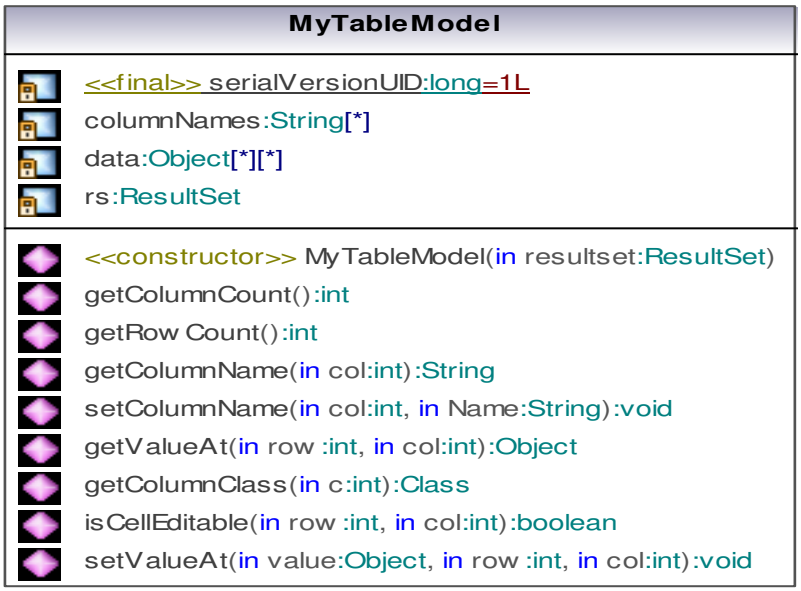
Pav. 32 Naujos subkategorijos kūrimo formos klasė



Pav. 33 Naujo vartotojo kūrimo formos klasė



Pav. 34 *MyComboBoxModel* klasė naudojama duomenų apsikeitimui su duomenų baze ir *jComboBox* komponentais



Pav. 35 *MyTableModel* klasė naudojama duomenų apsikeitimui su duomenų baze ir *jTable* komponentais

10.2 Duomenų bazės išeities kodas

Duomenų bazės išeities tekstai pridėti elektroninėje laikmenoje.

10.1 Programos išeities kodas

Programos išeities tekstai pridėti elektroninėje laikmenoje.