



Univerza v Ljubljani  
Fakulteta *za računalništvo  
in informatiko*

Univerza v Ljubljani  
**Fakulteta za računalništvo in informatiko**  
Šol. l. 2019/2020

# Dodatna seminarska naloga

Pri predmetu uporabniški vmesniki



## **IZVLEČEK**

Cilj dodatne seminarske naloge je izdelati aplikacijo z uporabo Java Swing grafičnega ogrodja. Namenjena je za zajemu podatkov posameznih avtomobilskih zavarovanj in splošni uporabi v zavarovalnici. Ciljni uporabniki so zaposleni v avtomobilski zavarovalnici. V seminarski nalogi je predstavljen razvoj aplikacije ter tehnologije in orodja, ki so bile uporabljene pri izdelavi aplikacije.

**Ključne besede:** NetBeans, Java, Swing, zavarovalnica, aplikacija.

## **ABSTRACT**

The goal of the project is to create an application using the Java Swing graphical framework. It is intended for data collection of individual car insurances and general use in the insurance company. The target users are employees of a car insurance company. The seminar paper presents the development of the application, technologies and tools used in the development of the application.

**Keywords:** NetBeans, Java, Swing, insurance company, application.

# KAZALO

<b>1.</b>	<b>UVOD.....</b>	<b>7</b>
<b>2.</b>	<b>PROGRAMSKI JEZIKI IN ORODJA.....</b>	<b>8</b>
2.1.	PROGRAMSKI JEZIK JAVA .....	8
2.2.	RAZVOJNO OKOLJE APACHE NETBEANS.....	8
2.3.	KNJIŽNICA JCALENDAR .....	9
2.4.	KNJIŽNICA JSON-SIMPLE .....	9
<b>3.</b>	<b>JEDRO .....</b>	<b>10</b>
3.1.	DESET NIELSEN-OVIH PRINCIPOV NAČRTOVANJA UPORABNIŠKIH VMESNIKOV .....	10
3.1.1	<i>Prilagodi se realnemu svetu.....</i>	<i>10</i>
3.1.2	<i>Konsistentnost in standardi.....</i>	<i>10</i>
3.1.3	<i>Pomoč in dokumentacija.....</i>	<i>10</i>
3.1.4	<i>Uporabnikov nadzor in svoboda .....</i>	<i>10</i>
3.1.5	<i>Vidljivost sistema.....</i>	<i>10</i>
3.1.6	<i>Fleksibilnost in učinkovitost.....</i>	<i>11</i>
3.1.7	<i>Izogibanje napakam.....</i>	<i>11</i>
3.1.8	<i>Raje prepoznavaj, kot si zapomni .....</i>	<i>11</i>
3.1.9	<i>Javljanje napak, diagnoza, reševanje.....</i>	<i>11</i>
3.1.10	<i>Estetika in minimalistično načrtovanje .....</i>	<i>11</i>
3.2.	IZBOR GRADNIKOV .....	12
3.2.1	<i>Vrstični meni .....</i>	<i>12</i>
3.2.2	<i>Gumbi .....</i>	<i>12</i>
3.2.3	<i>Vnosna polja.....</i>	<i>12</i>
3.2.4	<i>Izključujoča stikala.....</i>	<i>12</i>
3.2.5	<i>Ne izključujoča stikala.....</i>	<i>12</i>
3.2.6	<i>Kombinirani izvlečni seznammi .....</i>	<i>12</i>
3.2.7	<i>Kombinirana polja za vnos ali izbiro ( krožna polja) .....</i>	<i>13</i>
3.2.8	<i>Gradnik za vnos datuma .....</i>	<i>13</i>
3.2.9	<i>Pojavna okna.....</i>	<i>13</i>
3.3.	ARANŽIRANJE GRADNIKOV .....	13
3.4.	IZBOR BARV, BESEDILA IN SLIK .....	14
3.5.	POVRATNA INFORMACIJA.....	14
3.6.	ANALIZA UPORABNIKOV .....	14
3.7.	ODZIVNOST APLIKACIJE .....	14
<b>4.</b>	<b>ZAKLJUČEK .....</b>	<b>16</b>

## KAZALO SLIK

Slika 1: NetBeans .....	9
Slika 2: Primer gradnikov za izbor datuma .....	9
Slika 3: Najmanjša velikost .....	15
Slika 4: Največja velikost .....	15



## 1. UVOD

Pri zavarovalnici Nč bat so se odločili, da bodo zamenjali dosedanja aplikacijo, s katero so zajemali podatke za posamezna avtomobilska zavarovanja.

Naloga je na novo načrtati in implementirati aplikacijo. Nuditi mora možnost vnosa vseh podatkov, kot jih nudi tudi prejšnja aplikacija. Nov vmesnik bo načrtovan v skladu z desetimi Nielsen-ovimi principi in navodili načrtovanja uporabniških vmesnikov. Posamezne podobe oziroma gradnike aplikacije bo potrebno zamenjati z drugimi, saj nekateri obstoječi gradniki niso primerno izbrani. Pri izbiranju je potrebno upoštevati navodila in pravila izbora grafičnih gradnikov za interakcijo glede na razmere. Gradnike vmesnika je potrebno tudi pravilno aranžirati. Pri aranžiranju gradnikov je potrebno upoštevati navodila aranžiranja grafičnih gradnikov za interakcijo: tehnike za doseg preprostosti, balansiranje, grupiranje (beli presledki, okvirji), orientacija in poravnavanje. Vmesnik je treba tudi pravilno grafično načrtovati. V skladu z navodili načrtovanja vmesnikov morajo biti pravilno izbran tekst in barve.

Aplikacija mora omogočati tudi ponastavitev vnesenih vrednosti na začetne vrednosti ("Ponastavi vnose") in zapis vnesenih podatkov v podatkovno bazo ("Shrani vnos"), ki pa bo kar tekstovna datoteka. Vmesnik mora tudi preverjati, če so bili vneseni vsi podatki. Če niso bili, vmesnik to sporoči uporabniku. Prav tako mora vmesnik tudi testirati pravilnost podatkov (npr. obstoj določenega datuma) in pravilnost vnosa (npr. poštna številka - poštna številka abc123 ne obstaja). Po želji lahko vmesnik tudi dopolnimo.

## **2. PROGRAMSKI JEZIKI IN ORODJA**

Aplikacija je namenjena uporabi na vseh platformah in je izdelana v programskem jeziku Java. V veliko pomoč mi je bila knjižnica JCalendar, ki poenostavi vnos datumov. Za lažje shranjevanje podatkov sem uporabil knjižnico Json-simple, ki poenostavi uporabo json objektov. Aplikacijo sem razvil s pomočjo razvojnega okolja Apache NetBeans.

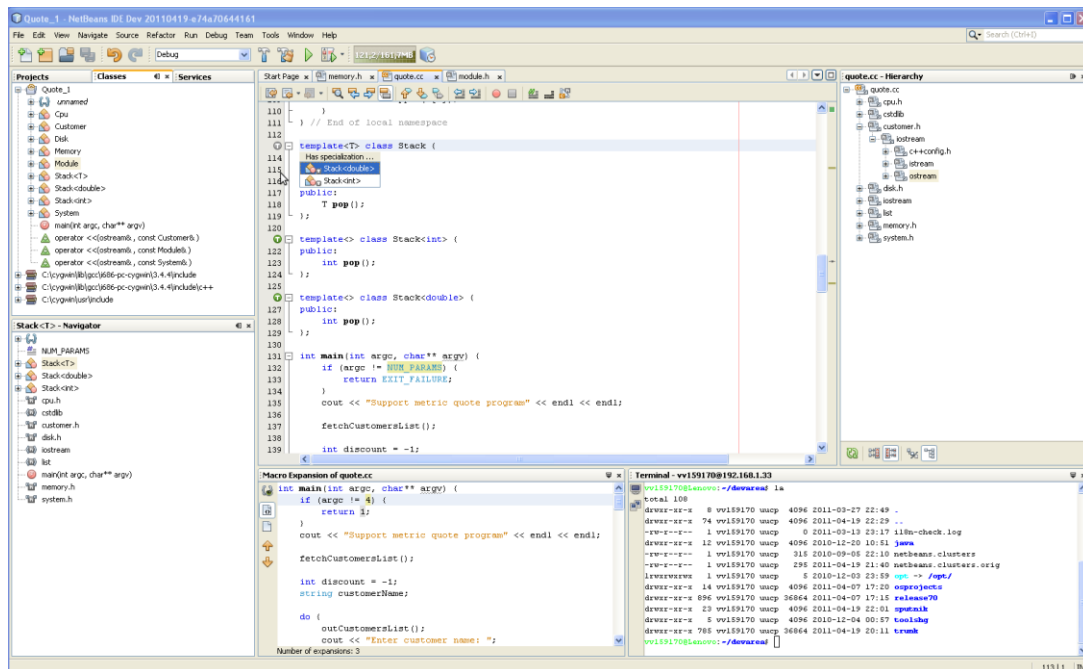
### **2.1. Programski jezik Java**

Java je splošno namenski, objektno usmerjen in prenosljiv programski jezik. Razvil ga je James Gosling pri podjetju Sun Microsystems. Projekt se je na začetku imenoval Oak oz. Hrast in je bil razvit kot zamenjava za programski jezik C++. Večji del sintakse jezika izhaja iz programskega jezika C in C++, vendar je Java preprostejša in lažja za učenje. Velika prednost je neodvisnost izvajanja od arhitekture naprave. Je eden izmed najbolj priljubljenih programskih jezikov in se pogosto uporablja za razvoj mobilnih aplikacij. [1]

### **2.2. Razvojno okolje Apache NetBeans**

NetBeans je integrirano razvojno okolje (IDE) namenjeno Java programskemu jeziku. Omogoča razvoj aplikacij iz nabora modularnih komponent programske opreme, imenovanih moduli. Deluje v sistemih Windows, macOS, Linux in Solaris. Poleg razvoja Java ima razširitve za druge jezike, kot so PHP, C, C ++, HTML5 in JavaScript. Aplikacije, ki temeljijo na NetBeansu, vključno z IDE NetBeans, lahko razširijo tretji razvijalci. [2]



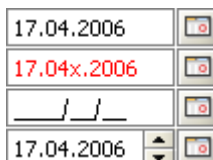


Slika 1: NetBeans

(Vir: <https://bit.ly/2TzHuRP>)

## 2.3. Knjižnica JCalendar

Pri razvijanju aplikacije sem uporabil knjižnico JCalendar. To je knjižnica za grafično izbiranje datuma in časa. Vsebuje več gradnikov, kot so: JDayChooser, JMonthChooser in JYearChooser. Knjižnico JCalendar sem izbral zaradi preproste implementacije in številnih funkcij, ki jih ponuja. [3]



Slika 2: Primer gradnikov za izbor datuma

(Vir: <https://bit.ly/2XtQdpU>)

## 2.4. Knjižnica Json-simple

Pri shranjevanju podatkov sem si pomagal s knjižnico Json-simple. To je preprosto Java orodje za manipulacijo z JSON objekti. [4]

### **3. JEDRO**

#### **3.1. Deset Nielsen-ovih principov načrtovanja uporabniških vmesnikov**

##### **3.1.1 Prilagodi se realnemu svetu**

Vmesnik je uporabniku prilagojen. Prilagaja se temu, da uporabniki niso računalničarji. Ne vsebuje tehničnih in nerazumljivih izrazov. Uporabniku je dovoljena uporaba šumnikov. Metafore niso uporabljene, saj ne bi bile koristne. Aplikacija je popolnoma odzivna.

##### **3.1.2 Konsistentnost in standardi**

Vmesnik vsebuje standardne gradnike. Omogočena je enostavna izbira datuma s primernim gradnikom. Nekateri klikajoči elementi so modre barve (namigi). Gradniki so vsebinsko razporejeni in grupirani. Celotni vmesnik je enostaven. Podobne stvari izgledajo in se obnašajo podobno. Različne stavi izgledajo drugače. Konsistentnost omogoča enostaven prenos obstoječega znanja. Uporabnik zna vmesnik uporabljati dokaj hitro. Primera konsistentnosti sta vrstični meni in gradnik za vnos datuma.

##### **3.1.3 Pomoč in dokumentacija**

Vmesnik je v osnovi zelo enostaven za uporabo, zato dodatne pomoči niti ne potrebuje. Pomoč je vseeno na voljo, če se uporabnik znajde v težavah. Je konkretna, kratka in vsebuje vse potrebne informacije.

##### **3.1.4 Uporabnikov nadzor in svoboda**

Uporabnik ima vedno nadzor nad delovanjem aplikacije. Vedno lahko prekliče vnos, ponastavi podatke in jih tudi uredi. Izhodi aplikacije so dobro označeni. Uporabnik se ne počuti kot ujetnik.

##### **3.1.5 Vidljivost sistema**

Aplikacija vsebuje statusno vrstico, katera prikazuje sporočila o delovanju. Ob odpiranju in shranjevanju datoteke se v statusni vrstici izpiše ime. Uporabnik je obveščen o statusu sistema. Ob napakah se napačno vnesen podatke obarva rdeče in v statusni vrstici se prikaže sporočilo o napaki. Kurzor miške se spreminja glede na različne gradnike. Poudarja se gradnik pod miško. Vsi vnosi so prikazani na eni strani aplikacije, kar omogoča zelo dobro vidljivost. Dodajanje več strani ne bi bilo smiselno, saj je podatkov malo. Odzivni časi akcij so hitri.

### **3.1.6 Fleksibilnost in učinkovitost**

Aplikacij omogoča uporabo bližnjic preko tipkovnice (memonikov in pospeševalnikov). Med gradniki se lahko uporabnik premika z uporabo tabulatorja. Ko uporabnik uporablja aplikacijo dlje časa, lahko poveča odzivnost aplikacije z uporabno bližnjic, ki omogočajo hitro uporabo določenih funkcionalnosti. Uporabljene so kombinacije, ki jih uporabnik pozna iz drugih programov ( CTRL+S, CTRL+O, CTRL+Q).

### **3.1.7 Izogibanje napakam**

Vsa potrebna informacija je vidna, zato pride težje do napak. Kjer je to mogoče, so uporabljeni kombinirani izvlečni sezname namesto besedilnih območij, saj je izbira manj podvržena napakam, kot samo pisanje. Pri izbirah ima uporabnik omejeno izbiro možnosti. Pri pisanju si mora uporabnik podatke zapomniti in še zatipka se lahko. Aplikacija sledi principu “manj tipkanja, več izbiranja”. Vsebuje potrditvene dialoge za nekatere destruktivne akcije.

Vmesnik se izogiba akcijam z podobnim opisom. Končne akcije niso postavljene zraven destruktivnih akcij.

### **3.1.8 Raje prepoznav, kot si zapomni**

Aplikacija uporablja kombinirane izvlečne sezname in ne besedilna območja, kjer je to mogoče. Uporaba izbire manj obremenjuje uporabnikov spomin. Sledi principu “manj tipkanja, več izbiranja”. Vsa potrebna informacija je vidna. Aplikacija uporablja menije in ne ukazni jezik. Uporablja generične ukaze ( Odpri, Shrani, Izhod).

### **3.1.9 Javljanje napak, diagnoza, reševanje**

Aplikacija vsebuje natančna obvestila o napakah, vendar so tehnične podrobnosti skrite. Ne vsebuje nekoristnih dialogov. Ob neuspelem odpiranju datoteke je prikazano tudi ime datoteke. Ob napačnem vnosu se napačno vnesen podatek obarva rdeče. Nepotrebni dialogi se ne prikazujejo. Omejitve programa so sporočene uporabniku (npr. registrska številka). Pomoč je konstruktivna, pove kje se je zgodila napaka. Dialogi so za uporabnika moteči in zmanjšujejo učinkovitost, zato jih ni veliko. Obvestila o napakah so do uporabnika prijazna.

### **3.1.10 Estetika in minimalistično načrtovanje**

Aplikacija je preprosta in dobro grafično načrtovana. Ne vsebuje odvečnih informacij, grafik in lastnosti. Vmesnik vsebuje le nekaj dobro izbranih pisav, barv in podrobnosti. Gradniki so pravilno poravnani in grupirani z belimi presledki. Besedilo je kratko in jedrnato.

## **3.2. Izbor gradnikov**

Večina izbranih gradnikov omogoča izbiranje, kar omogoča hitrejši vnos podatkov in zmanjšuje možnost napak. Poenostavljeno je tudi preverjanje napačnih vnosov.

Celoten vmesnik je enostaven, prilagojen uporabniku in uporablja standardne gradnike.

### **3.2.1 Vrstični meni**

Vsebuje pogoste in standardne akcije. Omogoča shranjevanje in odpiranje datoteke, ponastavitev vnosa, prikaz navodil, izpis avtorja in izhod iz aplikacije.

### **3.2.2 Gumbi**

Uporabljeni so, ker so vedno vidni, omogočajo hitre akcije in imajo razumljiv opis. Uporabnik razume opis gumba. Uporabljeni so memoniki in pospeševalniki. Gumbi so uporabljeni za klic akcij aplikacije.

### **3.2.3 Vnosna polja**

Uporabljena so, ker so fleksibilna in potrebujejo malo prostora. Omogočajo fleksibilni vnos brez omejitev. Če se le da, so bila nadomeščena z drugimi gradniki, ker zahtevajo uporabo tipkovnice in se uporabniki veliko motijo. Vnosna polja so uporabljena za vnos imena, priimka, ulica, hišne številke, oznake avtomobila in registrske označbe.

### **3.2.4 Izključujoča stikala**

Omogočajo eno izbiro. Uporabljena so za nastavljanje vrednosti. So lahko dostopna, omogočajo enostavno primerjavo, razumljiv opis in uporabo. Izključujoča stikala so uporabljena pri izbirah vrste osnovnega in kasko zavarovanja, saj se izbire izključujejo.

### **3.2.5 Ne izključujoča stikala**

Omogočajo več izbir. So lahko dostopna, omogočajo lahko primerjavo, razumljiv opis in uporabo. Ne izključujoča stikala so uporabljena za izbiro mladega voznika in dodatnega zavarovanja, saj se izbire ne izključujejo.

### **3.2.6 Kombinirani izvlečni sezname**

Omogočajo izbiro ene možnosti. Prikažejo se na zahtevo in omogočajo uporabo tipkovnice. Imajo neomejeno število možnosti, ne zasedejo veliko prostora in podpirajo prepoznavanje. Uporabljeni so za izbiro kraja, poštno številko, kraja prve registracije, registrske označbe, znamke, barve in vrste vozila.

### **3.2.7 Kombinirana polja za vnos ali izbiro ( krožna polja)**

Omogočajo izbiro ene možnosti. Možen je vnos preko tipkovnice. So fleksibilna in ne zasedejo veliko prostora. Omogočajo prost vnos in izbiro.

Uporabljena so za vnos prostornine, števila sedežev in moči avtomobila.

### **3.2.8 Gradnik za vnos datuma**

Omogočena je enostavna izbira datuma s primernim gradnikom. Uporabljen je standardni gradnik, torej ga uporabnik že zna uporabljati. Zaradi omejene izbire je možnost napake veliko manjša. Za uporabo je bilo potrebno naložiti knjižnico JCalendar.

### **3.2.9 Pojavna okna**

Uporabniku podajo bolj podrobna navodila in pridobijo uporabnikovo potrditev za izvajanje destruktivnih operacij.

Ker so lahko za uporabnika moteča, jih ni veliko.

## **3.3. Aranžiranje gradnikov**

Aranžiranje omogoča uporabniku enostavno in tekočo izvedbo njegovih nalog.

Gradniki so poravnani vertikalno in horizontalno. So enakomerno in simetrično razporejeni. Poskrbljeno je za enakomerno distribucijo in simetrijo. Pri tem so oznake na levi strani (razen pri opisih stikal, kjer so tudi na desni strani), vnosna polja pa na desni strani. Oznake so vertikalno poravnane po levem robu, vnosna polja pa vertikalno po levem robu in po desnem robu, kjer je to mogoče oziroma smiselno. Oznake in vnosna polja so horizontalno poravnana po spodnjem robu.

Sorodni gradniki so bližje skupaj. Grupirani so glede na tip gradnika, kontekst interakcije in kontekst podatkov. Uporabljeni so beli presledki in okvirji, ki ločujejo gradnike. Beli presledki dobro ločijo alternative. Vsi štirje presledki znotraj in zunaj okvirjev so enaki.

Podatke, ki sodijo skupaj so zaradi boljše vidljivosti in grupiranja ločeni z uporabo okvirjev, kjer so uporabljeni tudi beli presledki.

Komponente so primerno orientirane glede na tok aplikacije in glede na tok uporabnikovega premikanja skozi aplikacijo.

Uporabljena je tehnika redukcije. Jezik je tak, da uporabniku pomaga.

Uporabniški vmesnik je preprost. Nepomembni elementi so odstranjeni.

Uporabljeni so regularni vzorci. Omejene so nepomembne variacije med elementi ( pisave, barve, debeline črti, dimenzije in orientacije).

Uporabljeno je le eno glavno okno.

Organizacija gradnikov odraža organiziranosti okolja realnega sveta, ki mu služi aplikacija.

Velikost istih tipov gradnika je enaka.

Uporabljena je samo ena pisava

### **3.4. Izbor barv, besedila in slik**

Besedilo izvira iz zahtev naloge in se navezuje na izposojno avtomobilov.

Uporabniški vmesnik ne vsebuje veliko barv in slik, saj bi lahko to uporabnika motilo pri uporabi.

Rdeča barva predstavlja opozorila na napake in posebej izstopa.

Besede so preproste, čiste in vljudne. Besedila za gumbe so primerno izbrana. Besedilo je lepo poravnano in ni preblizu roba.

Nesmiselne razlike so zanemarljive.

Pisava je povsod enaka, ker omogoča le primerjavo glede ne enakost.

Uporabljenih je le nekaj barv. Nasičene barve niso uporabljene.

### **3.5. Povratna informacija**

Uporabniški vmesnik vsebuje statusno vrstico, v kateri se izpisujejo sporočila o delovanju in tudi napake.

Napačno vneseni podatki se obarvajo rdeče.

Vmesnik vsebuje tudi pojavna okna, ki uporabniku podajo bolj podrobna navodila in obvestila.

Odzivnost sistema je prilagojena hitrosti človekovih miselnih procesov. Informacija se pojavi hitro, da ne pride do nižje učinkovitosti.

Obvestila so direktna, kratka in preprosta.

Aplikacija uporablja dialoge za potrjevanje destruktivnih akcij, odpiranj in shranjevanj datotek.

Dialogi za napake niso uporabljeni.

### **3.6. Analiza uporabnikov**

Ciljni uporabnik je oseba, zaposlena v zavarovalnici vozil.

Uporabniški vmesnik omogoča zaposlenemu urediti zavarovanje vozil glede na želje strank.

Posebno znanje ni potrebno, saj je vmesnik enostaven za uporabo. Vmesnik omogoča učinkovito sklepanje zavarovanj.

### **3.7. Odzivnost aplikacije**

Aplikacije je popolnoma odzivna in se prilagaja različnim uporabnikovim zahtevam.

Aplikacija za zavarovanje avtomobilov

Datoteka Uredi Pomoč

**Podatki o zavarovancu**

Ime:  Ulica:

Preimek:  Hišna številka:

Datum rojstva:  Kraj:

Mlad voznik: ☒ Poštna številka:

**Podatki o vozilu**

Znamka:  Gorivo:

Oznaka:  Prostornina (ccm³):

Barva:  Št. sedežev:

Vrsta vozila:  Moč (kW):

**Podatki o registraciji**

Datum prve reg.:  Registrska označba:

Kraj prve reg.:

**Podatki o zavarovanju**

Osnovno zavarovanje: ☐ AO ☒ AO+ Kasko: ☐ Polno ☒ Odbitna franšiza ☐ Brez

Dodatno zavarovanje: ☒ Zavarovanje stekel ☐ Zavarovanje tretje osebe ☒ Zavarovanje na parkirišču

☐ Zavarovanje proti toči ☒ Zavarovanje potnikov ☐ Zavarovanje proti kraji

☐ Zavarovanje gum

Aplikacija pripravljena na uporabo.

Slika 3: Najmanjša velikost

Aplikacija za zavarovanje avtomobilov

Datoteka Uredi Pomoč

**Podatki o zavarovancu**

Ime:  Ulica:

Preimek:  Hišna številka:

Datum rojstva:  Kraj:

Mlad voznik: ☒ Poštna številka:

**Podatki o vozilu**

Znamka:  Gorivo:

Oznaka:  Prostornina (ccm³):

Barva:  Št. sedežev:

Vrsta vozila:  Moč (kW):

**Podatki o registraciji**

Datum prve reg.:  Registrska označba:

Kraj prve reg.:

**Podatki o zavarovanju**

Osnovno zavarovanje: ☐ AO ☒ AO+ Kasko: ☐ Polno ☒ Odbitna franšiza ☐ Brez

Dodatno zavarovanje: ☒ Zavarovanje stekel ☐ Zavarovanje tretje osebe ☒ Zavarovanje na parkirišču

☐ Zavarovanje proti toči ☒ Zavarovanje potnikov ☐ Zavarovanje proti kraji

☐ Zavarovanje gum

Aplikacija pripravljena na uporabo.

Slika 4: Največja velikost

## **4. ZAKLJUČEK**

V seminarski nalogi sem opisal postopek razvoja aplikacije za sklepanje zavarovanj vozil. Razvil sem vmesnik namenjen zaposlenim v raznih zavarovalnicah vozil.

Veliko časa sem porabil za odzivnost aplikacije, saj je narediti dobro odziven vmesnik precej zahtevno. Težave sem imel tudi z uporabo gradnika za izbiro datuma. Ob izdelavi aplikacije sem poglobil svoje znanje Swing ogrodja in programskega jezika Java, kar bom koristno uporabil v prihodnosti.

Ugotovil sem, da je izdelovanje vmesnikov v NetBeansu težje, kot izdelovanje teh v Android studio. Za izdelavo aplikacije, ki bo vseh uporabnikom, tekoče delovala, lepo izgledala in delovala na veliko napravah, je potrebno veliko dela in testiranja. Največ težav sem imel z izdelavo odzivnega vmesnika aplikacije.



## **VIRI**

1. Java Wikipedija [URL: <https://bit.ly/1kLzC7h>], 24. 5. 2020
2. NetBeans Wikipedija [URL: <https://bit.ly/36nWjvS>], 24. 5. 2020
3. JCalendar [URL: <https://bit.ly/2TwRICy>], 24. 5. 2020
4. Json-simple [URL: <https://bit.ly/2XrW4vC>], 24. 5. 2020