|  |  |  |
| --- | --- | --- |
|  | УНИВЕРЗИТЕТ У НОВОМ САДУ  **ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У НОВОМ САДУ** |  |

Тамара Глишић

**СИСТЕМ ЗА ПРОДАЈУ КАРАТА НА СКИЈАЛИШТУ И ПРЕПОРУКУ ПОЛИСЕ ОСИГУРАЊА**

ДИПЛОМСКИ РАД

- Основне академске студије -

Нови Сад, 2021.

|  |  |  |
| --- | --- | --- |
|  | УНИВЕРЗИТЕТ У НОВОМ САДУ ⚫ **ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА**  21000 НОВИ САД, Трг Доситеја Обрадовића 6 | Датум: |
|  |
| **ЗАДАТАК ЗА ИЗРАДУ ДИПЛОМСКОГ (BACHELOR) РАДА** | Лист: |
| 1/1 |

*(Податке уноси предметни наставник - ментор)*

| Врста студија: | **Основне академске студије** |
| --- | --- |
| Студијски програм: | **Софтверско инжењерство и информационе технологије** |
| Руководилац студијског програма: | **ван. проф. др Мирослав Зарић** |

|  |  |  |  |
| --- | --- | --- | --- |
| Студент: | **Тамара Глишић** | Број индекса: | **SW38/2017** |
| Област: | **Системи базирани на знању** | | |
| Ментор: | **доц. др Синиша Николић** | | |
| НА ОСНОВУ ПОДНЕТЕ ПРИЈАВЕ, ПРИЛОЖЕНЕ ДОКУМЕНТАЦИЈЕ И ОДРЕДБИ СТАТУТА ФАКУЛТЕТА  ИЗДАЈЕ СЕ ЗАДАТАК ЗА ДИПЛОМСКИ (Bachelor) РАД, СА СЛЕДЕЋИМ ЕЛЕМЕНТИМА:   * проблем – тема рада; * начин решавања проблема и начин практичне провере резултата рада, ако је таква провера неопходна; * литература | | | |

**НАСЛОВ ДИПЛОМСКОГ (BACHELOR) РАДА:**

|  |
| --- |
|  |

**ТЕКСТ ЗАДАТКА:**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| Руководилац студијског програма: | Ментор рада: |
|  |  |

|  |
| --- |
| Примерак за:  - Студента;  - Ментора |

САДРЖАЈ

[1. УВОД 1](#_Toc81393896)

[2. СТАЊЕ У ОБЛАСТИ 2](#_Toc81393897)

[3. КОРИШЋЕНЕ ТЕХНИКЕ И ТЕХНОЛОГИЈЕ 3](#_Toc81393898)

[3.1. Angular 3](#_Toc81393899)

[3.2. Spring 4](#_Toc81393900)

[3.3. Drools 6](#_Toc81393901)

[3.4. REST сервис 7](#_Toc81393902)

[4. СПЕЦИФИКАЦИЈА 8](#_Toc81393903)

[4.1. Дијаграм случајева коришћења 8](#_Toc81393904)

[4.2. Дијаграм класа 10](#_Toc81393905)

[4.3. Дијаграм секвенци 13](#_Toc81393906)

[5. ИМПЛЕМЕНТАЦИЈА 15](#_Toc81393907)

[5.1. Клијентска апликација 15](#_Toc81393908)

[5.2. Серверска апликација 16](#_Toc81393909)

[5.3. *Drools* апликација 18](#_Toc81393910)

[5.3.1. Правила за израчунавање цене карте 18](#_Toc81393911)

[5.3.2. Правила за препоруку полисе осигурања 26](#_Toc81393912)

[5.3.3. *CEP* правила 30](#_Toc81393913)

[6. ПРИКАЗ ИМПЛЕМЕНТИРАНОГ РЕШЕЊА 32](#_Toc81393914)

[6.1. Неулоговани корисник 32](#_Toc81393915)

[6.2. Улоговани корисник 35](#_Toc81393916)

[6.3. Администратор 41](#_Toc81393917)

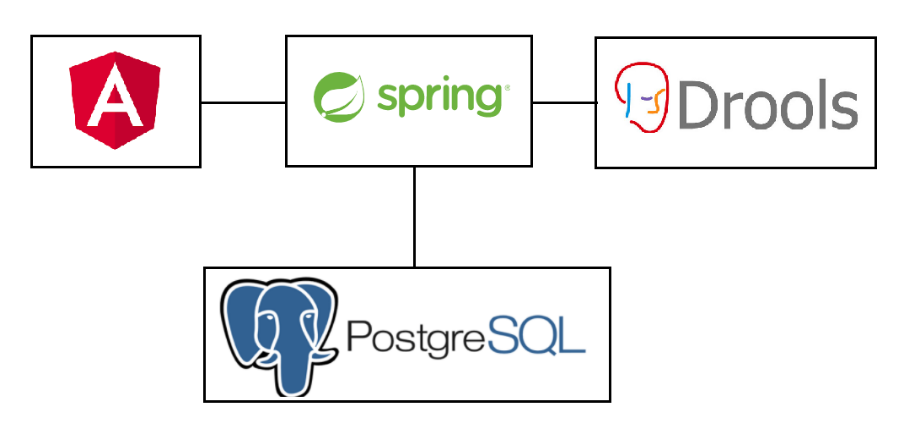
[ЛИТЕРАТУРА 46](#_Toc81393918)

# УВОД

# СТАЊЕ У ОБЛАСТИ

# КОРИШЋЕНЕ ТЕХНИКЕ И ТЕХНОЛОГИЈЕ

Решење је имплементирано као веб апликација чија архитектура обухвата клијентски и серверски део, базу података и систем базиран на правилима (Слика 3.1). Клијентски део имплементиран је употребом *Angular* радног оквира, за серверски део коришћен је *SpringBoot* заједно са *REST (Representational State Transfer)*  сервисима који за рамену података користе *JSON (JavaScript Object Notation)* формат. Размена података и комуникација обавља се путем *HTTP (Hyper Text Transfer Protocol)* протокола. Као база података коришћен је *PostgreSQL*, а систем базиран на правилима је *Drools.*



Слика 3.1 Архитектура веб апликације

## Angular

*Angular* је радни оквир за развој клијентских апликација употребом *HTML-a* (*Hyper Text Markup Language*) и *TypeScript-a.* Сам радни оквир написан је у *TypeScript-у,* а његов основни градивни блок су компоненте. Поред компонената, клијентска апликација писана у *Angular-у* састоји се још и од модула и сервиса, што чини три главна дела апликације.

***Компоненте***

Компоненте обједињују приказ, програмску логику која га контролише и податке. Програмска логика налази се у *TypeScript* класама које су повезане са *HTML* темплејтом који дефинише начин на који ће клијенту бити приказани подаци.

***Модули***

*Angular* апликације су модуларне и састоје се из бар једног (коренског) модула. Модули су *TypeScript* класе које обједињују различите компоненте, и омогућују да се компоенете из конкретног домена извезу како би могле бити коришћене у неком другом модулу или обрнуто, да могу да се увезу из неког другог модула и користе у конкретном модулу.

***Сервиси***

Да би клијентска апликација комуницирала са серверским делом, користе се сервиси. Сервиси су *TypeScript* класе у којима се налазе подаци и логика која није карактеристична за само једну компоненту већ може бити дељена са више компоненти.

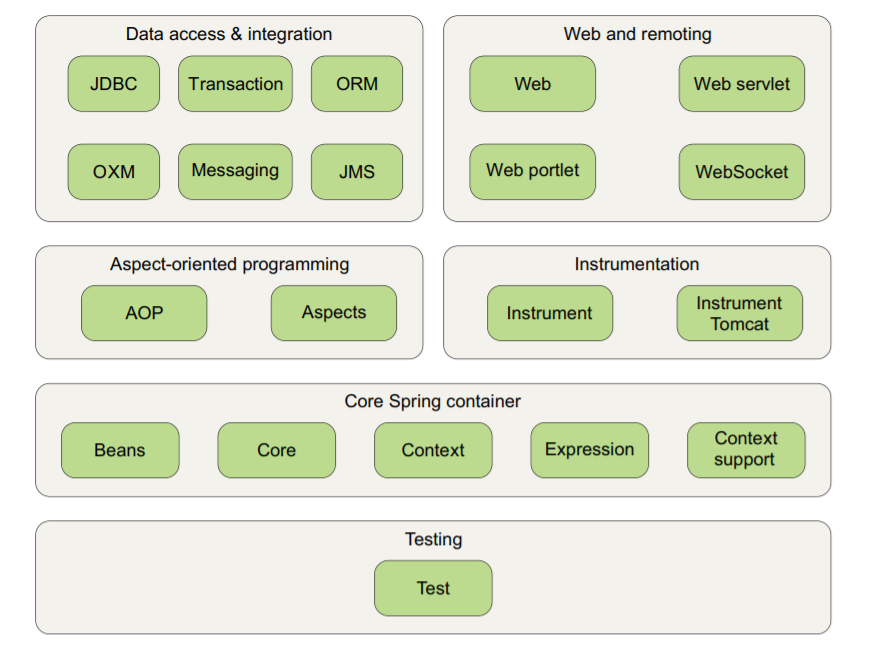
## Spring

*Spring* је *open-source* радни оквир написан у *Java* програмском језику. Настао је са циљем да омогући једноставнији развој *Java* пословних апликација. (референца на кнјигу СПринг ин ацтион). Главни концепти *Spring* радног оквира су *Dependency injection* (DI) и *Aspect-oriented programming* (AOP).

DI је механизам за повезивање објеката уз помоћ којег објекат добија своје зависности уместо да их сам добавља.

АОР омогућује поновну искоришћење компоненти. На овај начин функционалности се независно дефинишу, а затим се аспекти примењују у потребним деловима апликације.

*Spring* радни оквир се састоји из шест јасно дефинисаних модула (Слика 3.2).



Слика 3.2 Приказ модула у Spring-у [ ref na knjigu ]

***Core Spring container***

Главни део *Spring* радног оквира је Контејнер енгл. *Container* који управља животним циклусом објеката, тј начином на који ће они бити креирани, конфигурисани и коришћени. Овај модул садржи фабрику објеката која омогућава DI механизам при коришћењу објеката.

***Aspect – oriented programming***

Један од главних концепата, овај модул пружа основе за разовј личних аспеката у *Spring* апликацији. Са АОР проблеми везани за апликацију, као што су трансакције и безбедност, су одвојени од објеката на које се примењују.

***Data access and integration***

Модул за управљање подацима и интеграцију апстрахује везу са базом података. Рад са *Java Database Connectivity (JDBC)* и *Data – access objects (DAO)* модулима омогућују краћи и једоставнији код одвојен од базе и на тај начин омогућује експесивније поруке о SQL грешкама независно од система за управљање базом података. *Spring* има подршку за објектно – релационо мапирање (ОRM) али овај модул нема своју ORM имплементацију већ користи неку од постојећих као што су *Hibernate, OpenJPA* итд.

***Web and remoting***

Овај модул садржи радни оквир по *Model – View – Controller (MVC)* парадигми која је општеприхваћен приступ за изградњу и развој комплетних веб апликација. MVC гарантује да је *User interface (UI)* одвојен од апликативне логике. Поред веб апликација, овај модул пружа подршку за позиве удањених метода и на тај начин омогућују комуникацију са другим апликацијама.

***Instrumentation***

Омогућује прилагођавање рада *Java Virtual Machine (JVM)*, конкретно, садржи измењене начине за учитавање класа тј. трансформише их уз помоћ специфичне имплементације *ClassLoader* објеката.

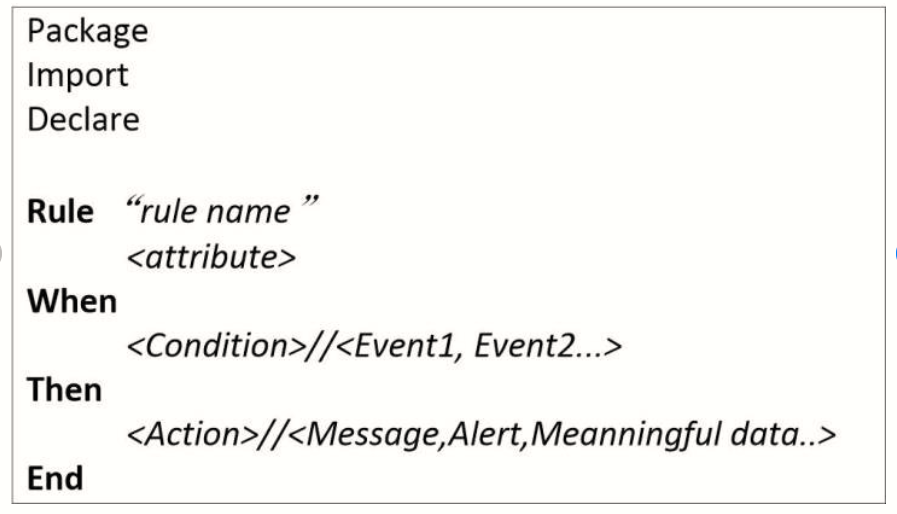
***Testing***

Овај модул омогућује јединично и интеграционо тестирање засновано на *Junit* и *TestNG* библиотекама. Садржи и имплементацију *mock* објеката који могу бити коришћени за јединичне тестове изоловано од других делова апликације. *Mock* објекат је објекат двојник који симулира понашање стварног објекта.

## Drools

*Drools* је систем за управљање пословним правилима и имплементацију експертских система који опонашају доношење одлука људског стручњака из неког домена. Заснован је на закључивању унапред и уназад и подржава *Complex event processing (CEP)* за откривање и обраду значајних догађаја у систему.

*Drools* има своју синтаксу за писање правила која прати структуру приказану на слици 3.3.



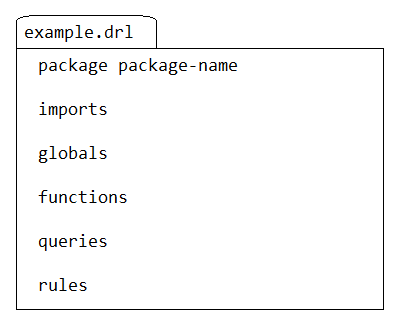
Слика 3.3 Структура правила [1]

*Attribute –* атрибути су опциони и означавају како правило треба да се понаша.

*Condition –* услови се постављају према правилима и служе за верификацију правила. Ако су дефинисани услови испуњени, правило се примењује, тј. реализују се акције дефинисане у *then* делу правила.

*Action –* пише се у стандардном *Java* коду и садржи акције које треба извршити уколико су испуњени услови.

Правила се пишу у фајлу који има .drl екстензију и структуру приказану на слици 3.4. Сви елементи су опциони.



Слика 3.4 Структура .drl фајла

## REST сервис

*Representational State Transfer (REST)* је стил софтверске арфитектуре заснован на постојању ресурса и униформног управљања њима преко скупа предефинисаих операција којег чине методе:

* *CREATE –* метода за креирање ресурса.
* *READ –* метода за читање ресурса.
* *UPDATE –* метода за измену ресурса.
* *DELETE –* метода за брисање ресурса.

REST сервис користи ресурсе чија је репрезентација у виртуалној форми у облику JSON објекта или неке друге репрезентације која одговара кориснику тих ресурса. Ти ресурси се трансферују из једне у другу апликацију.

REST дефинише шест архитектонских ограничења која треба да испуни сваки сервис да би био RESTful.

* Униформни интерфејс – како би било задовољено ово ограничење неопходно је испунити захтеве као што су: идентификација ресурса, манипулација ресурсима кроз њихову репрезентацију, самоописна порука као информација клијенту и хипертекст што значи да ће клијент након приступа ресурсу моћи да користи хиперлинкове како би могао да реализује све доступне акције карактеристичне за тај ресурс.
* Клијент – сервер као две независне апликације побољшавамо преносивост корисничког интерфејса на више платформи и на тај начин поједностављујемо компоненте сервера.
* *Stateless –* овај вид комуникације захтева да сваки захтев који долази од клијента на сервер мора да садржи све потребне информације за његову обраду. Стање сесије се у потпуности чува на страни клијента.
* Могућност кеширања – подаци у одговору или захтеву се имплицитно или експлицитно означавају као подаци који могу или не могу да се кеширају. То значи да је могућа поновна искористивост једног те истог одговора уколико корисник шаље еквивалентне захтеве.
* Слојевитост – начин организације компонената у хијерархије слојева, при чему свака компонента не може да интерагује са компонентама које су изван њеног непосредног слоја.
* Код на захтев (опционо) – REST омогућава проширење функционалности преузимањем и извршавањем кода у облику аплета или скрипти.

# СПЕЦИФИКАЦИЈА

Систем за куповину карата за скијалиште и препоруку осигурања на стази ради тако што обрачунава цену карата за унете критеријуме. Након што је корисник купио карте, он може да купи и осигурање које ће му бити предложено на основу његових захтева и здравствеог стања које је од значаја за скијаше. Његове функционалности биће представљене кроз различите *Unified Modeling Language (UML)* дијаграме.

## Дијаграм случајева коришћења

Дијаграми случајева коришћења илуструју на које све начине је могуће користити апликацију. На тај начин увиђамо сценарије у којима апликација интерагује са крајњим корисником, као и циљеве до којих ти сценарији могу да доведу.

Апликацију могу да користе три типа корсника, то су:

* Неулоговани корисник
* Улоговани корисник
* Администратор

Сва три типа корсника имају приступ странци за логовање и прегледу и претрази постојећих скијалишта и информацијама колика је попуњеност сваког од скијалишта за наредних 5 дана.

Права улогованог корисника су следећа:

* Куповина карата за конкретно скијалипте. Корисник уноси потребне информација као што су избор скијалишта, период коришћења, тип транспорта, дневни период корићења карте, уноси кориснике карата и да ли има права на неке погодности и попусте.
* Преглед свих купљених карата.
* Преглед купљених полиса уз конкретну карту.
* Отказивање карата. Корисник може да откаже и уклони карте минимум два дана пре почетка важења карата.
* Куповина осигурања за кориснике карата. Корисник попуњава упитник у којем уноси податке неопходне за процену идеалне полисе која би задовољила његове критеријуме, способности и здравствено стање.
* Корисник бира полису из листе предложених полиса осигурања. Изабрану полису може да купи само једном или да исту полису одмах купи и за све остале кориснике карата.
* Отказивање полиса осигурања. Корисник може да откаже купљене полисе минимум два дана пре почетка важења.

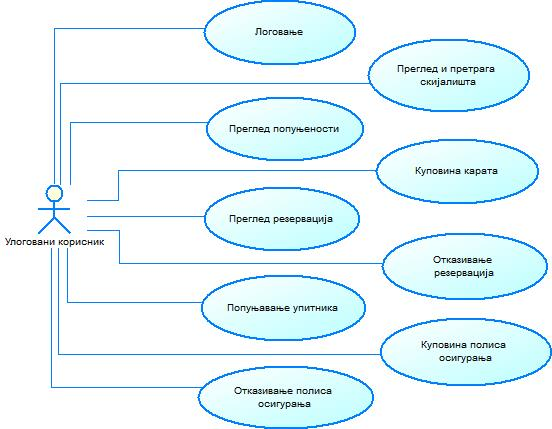
Права администратора су следећа:

* Додавање, измена, брисање и претрага скијалишта.
* Увид у посећеност скијалишта за конкретан датум.
* Претрага купљених полиса осигурања.
* Унос података везаних за искоришћење полисе осигурања.

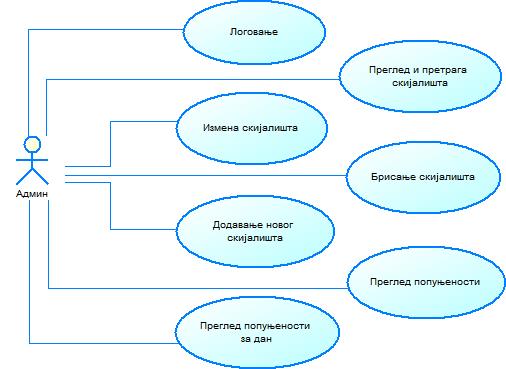
На сликама 4.1, 4.2, и 4.3 се могу видети дијаграми случајева коришћења за сва три типа корисника.



Слика 4.1 Дијаграм случајева коришћења неулогованог корисника



Слика 4.2 Дијаграм случајева коришћења за улогованог корисника



Слика 4.3 Дијаграм случајева коришћења за администратора

## Дијаграм класа

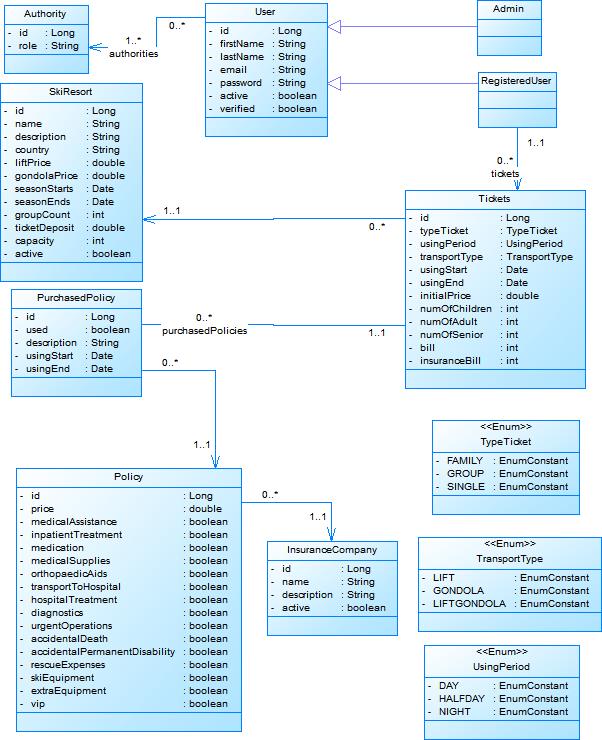
Дијаграм класа служи за моделовање система и приказ саме структуре система на нивоу класа и интерфејса [3]. Класе у класном дијаграму одговарају класама које чине модел података у систему. У дијаграму си приказана имена класа, атрибути које те класе садрже као и однос и повезаност између класа.

Систем је моделован са девет класа и три енумерације (Слика 4.4). Класе овог система су:

* *Authority:* класа која представља корисничке улоге у систему. Садржи атрибут под називом *role.* У систему су регистроване две корисничке улоге а то је улога администратора (*ROLE\_ADMIN*) и улога регистрованог корисника (*ROLE\_REGISTERED\_USER*).
* *User:* класа која садржи основне податке о кориснику као што су име, презиме, подаци за логовање, и стање налога као и улоге корисника у систему. Ову класу наслеђују класе *RegisteredUser* и *Admin.*
* *RegisteredUser:* класа која наслеђује класу *User* и садржи додатно листу свих корисникових резервисаних карата
* *Admin:* класа која наслеђује класу *User.*
* *SkiResort:* класа која описује скијалиште. Као атрибуте садржи назив скијалишта, кратак опис, земљу у којој се налази то скијалиште, почетак и крај сезоне, цену ски лифта, цену гондоле, депозит за издавање карте, број корисника који означава групну карту и поље које означава да ли је то скијалиште и даље активно тј. да ли се за њега могу резервисати карте.
* *InsuranceCompany:* класа са атрибутима за опис компаније тј. осигуравајуће куће. Садржи назив компаније, кратак опис и поље за проверу валидности компаније.
* *Policy:* поред поља која чува цену полисе, ова класа садржи поља која означавају шта покрива конкретна полиса осигурања. Сва та поља су *boolean* типа, чија вредност *true* означава да полиса покрива захтев, а вредност *false* да не покрива захтев.
* *PurchasedPolicy:* класа за купљене полисе која чува податке о томе која конкретна полиса је купљена и за које карте. Поред тих атрибута, садржи и датуме од кад до кад је период важења, поље које представља искоришћеност полисе и поље за кратак опис и разлог искоришћења полисе.
* *Tickets:* класа које одговара једној резервацији карата за скијање. Као атрибуте садржи три енумерације: тип карте, тип транспорта, и период коришћења карте. Поред енумерација, садржи атрибуте које представљају појединачан број корисника који могу бити деца, одрасли или сениори, арибут као укупна цена карата и атрибут за укупну цену купљених полиса.

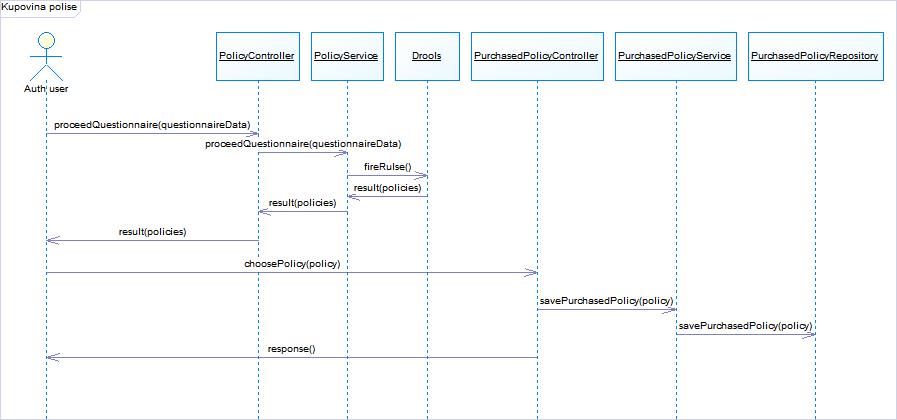
Три енумерације обухватају:

* *TypeTicket:* тип карте. Вредности које може имати енумерација типа карте је *SINGLE* (појединачна)*, FAMILY* (породична)*, GROUP* (групна).
* *TransportType:* тип транспорта. Подржава вредности *LIFT, GONDOLA* и *LIFTGONDOLA* што симболизује комбиновани тип транспорта.
* *UsingPeriod:* период коришћења карте дозвољава вредности као што су *DAY* (дневна карта), *HALFDAY* (полудневна карта) и *NIGHT* (ноћна карта).



Слика 4.4 Дијаграм класа

## Дијаграм секвенци



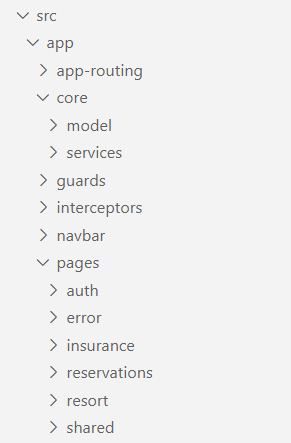
Слика 4.5 Дијаграм секвенци

# ИМПЛЕМЕНТАЦИЈА

Систем за куповину карата за скијалиште и препоруку осигурања састоји се из клијентске апликације имплементиране коришћењем *Angular* радног оквира, серверског дела уз помоћ *Spring-a* и система базираног на правилима који је имплеметиран употребом *Drools* алата. Код апликације налази се на линку: гитхуб.цом

## Клијентска апликација

Клијентска апликација имплементирана је уз помоћ *Angular* радног оквира. Састоји се од коренског модула, модула за рутирање и још шест модула у који обухватају логички сродне компоненте. Структура апликације приказана је на слици 5.1.



Слика 5.1 Структура Angular апликације

*app – routing –* овде се налази модул за рутирање који импортује руте. Руте на које су намапиране компоненте, налазе се у фајлу *routes.ts.*

*core –* садржи модел и сервисе за комуникацију клијентске апликације са серверским делом.

*guards –* пакет у коме се налази *loginGuard* и *roleGuard* за проверу права приступа рутама. Поједине руте су доступне само улогованим корисницима, док су неке доступне само за конкретне улоге корисника, админа или регистрованог корисника.

*interceptors –* сервис за пресретање и измену захтева. Дешифрује токен који се налази у *localStorage* и шаље га заједно са захтевом као *Authorization* параметар.

*navbar –* у овом пакету груписани су навигацион менији. Саджи навигациони мени за неулогованог корисника, улогованог корисника и админа.

*auth –* модул саджи компоненту за логовање и компоненту за регистрацију.

*error –* модул у коме се налазе компоненте за приказ грешака уколико, на пример, корисник покуша да приступи страници за коју нема приступ.

*insurance –* модул који обједињује компонете за приказ, претрагу и измену полиса осигурања као и за попуњавање упитика за предлог полиса.

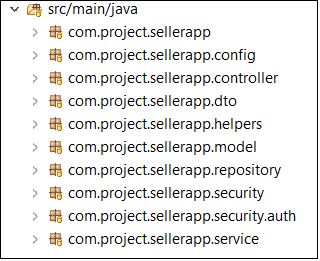
*reservations –* модул чине компонента за приказ резервација конкретног корисника и компонента за добављање неопходних информација за креирање нове резервације.

*resort –* у овом модулу груписане су компоненте за приказ, едитовање, додавање нових скијалишта, као и компонента за приказ капацитета скијалишта.

*shared –* овде се налази дијалог компонента за потврду акција. Компонента је прилагодљива различитим акцијама.

## Серверска апликација

Серверска апликација имплементирана је у *Java* програмском језику коришћењем *SpringBoot* радног оквира. Структура апликације (Слика 5.2) прати препоручене конвенције. Пакети групишу логичке целине.



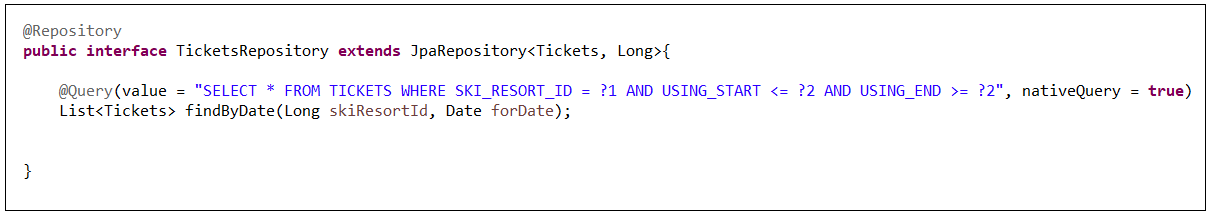
Слика 5.2 Структура серверске апликације

*controller –* обједињује класе контролера. Котролери су задужени за комуникацију између клијента и сервера. Сваки захтев који је упућен серверу пролази кроз контролер и кроз проверу права приступа. Контролери инјектују потребне сервисе и након пристизања захтева позивају потребне методе из сервиса. Након обрађеног захтева клијенту се шаље статус о успешности обрађеног захтава, и резултати обраде захтева. Овај пакет садржи следеће контролере:

* *AuthenticationController –* обрађује захтеве за логовање, регистрацију, и одјаву са система.
* *InsuranceCompanyController –* обрађује захтеве који се тичу осигуравајућих компанија.
* *PolicyController –* обрађује захтев за процесуирање података из упитника.
* *PurchasedPolicyController –* за обраду захтева који се извршују над купљеним полисама.
* *SkiResortController –* за обраду захтева за добављање, претрагу, измену, додавање нових скијалишта.
* *TicketsController –* обрађује захтеве везане за карте.

*service –* у овом пакету се налазе класе сервиса. Класе сервиса имплементирају пословну логику и инјектују потребне интерфејсе репозиторијума преко који се врши комуникација са базом. Поред тих сервиса, овај пакет садржи и сервис који активира и комуницира са правилима у *Drools* апликацији. То је сервис *KieService.* У њему се налазе методе за позивање правила везаних за куповину карата и препорку осигурања, као и за позивње *CEP (Complex Event Processing)* правила.

*repository –* садржи интерфејсе који директно комуницирају са базом. Ови интерфејси наслеђују *JpaRepository* [2] из *Spring* радног оквира. *JpaRepository* поред стандардних *CRUD (Create Update Delete)* операција, пружа подршку за додавање и нових, нестандардних операција над подацима. Специфичне операције не морају се имплемнтирати, довољно је у интерфејсу написати декларације метода пратећи конвенције у именовању. Нестандардни упити који се не могу на овај начин изразити, задају се у *JPQL* језику који је део *JPA* имплементације (Слика 5.3).



Слика 5.3 Пример нестандардног упита задатог у JPQL језику, проналази карте тако да задати датум упада у период коришћења карата

*config –* садржи класе неопходне за конфигурисање безбедности апликације. Конфигурација безбедности се врши редефинисањем метода класе *WebSecurityConfigurerAdapter.* Дефинишемо начин утврђивања идентитета корисника при аутентикацији, дефинишемо права приступа за захтеве ка одређеним *URL* – овима *(енгл. Uniform Resource Locator)* и дефинишемо генералну безбедност веб апликације, као што је игнорисање аутентикације за одређене захтеве идт.

*dto –* пакет који саджи *DTO (Data Transfer Object)* класе које се користе за пренос података између клијента и сервера. Ове класе моделују податке из базе.

*model –* садржи класе које моделују систем. Како су ове класе повезане између себе представњено је у поглављу.............\*\*\*

*helpers –* пакет који садржи классе за мапирање података из ентитета у *DTO* класе. Такође садржи и класу *Utility* са методама које манипулишу датумима, конвертујућих у *String,* или за рачунање нових датума неопходних за извршавање неких задатака.

*security –* класе за рад са токенима. У овом случају *JSON* веб токенима. Ту се налазе класе са методама за генерисање, преузимање токена из захтева и читање података из токена, класа која дефинише филтер који ће пресретати сваки захтев клијента упућен серверу и проверавати валидност токена.

## *Drools* апликација

*Drools* апликација чини засебну апликацију базирану на знању. У њој се налазе правила на основу којих систем опонаша експерта специфичне области и доноси одлуке у зависности од тих правила. Правила су подељена у логичке целине и налазе се у три пакета. Први пакет садржи правила за израчунавање цене карата, други пакет је пакет са правилима за препоруку полисе осигурања и трећи пакет садржи *CEP* правила која служе за детектовање сумњивих активности или праћења догађаја од значаја, више о тим догађајима у наставку.

### Правила за израчунавање цене карте

Након уношења корисникових захтева на клијентској страни, подаци се обрађују, пролаазе кроз правила, и рачуна се најповољнија цена за његове карте. Клијентов захтев се састоји из:

* одабира скијалишта,
* одабира типа транспорта (може бити ски лифт, гондола или комбиновано),
* одабира периода (дневна карта, полудневна или ноћна),
* почетак / крај коришћења карте,
* уношење појединачно броја корисника карата. Корисници карата могу бити деца, одрасли или сениори.
* Навођење одређених привилегија за попусте (студентски попуст или *loyalty)*.

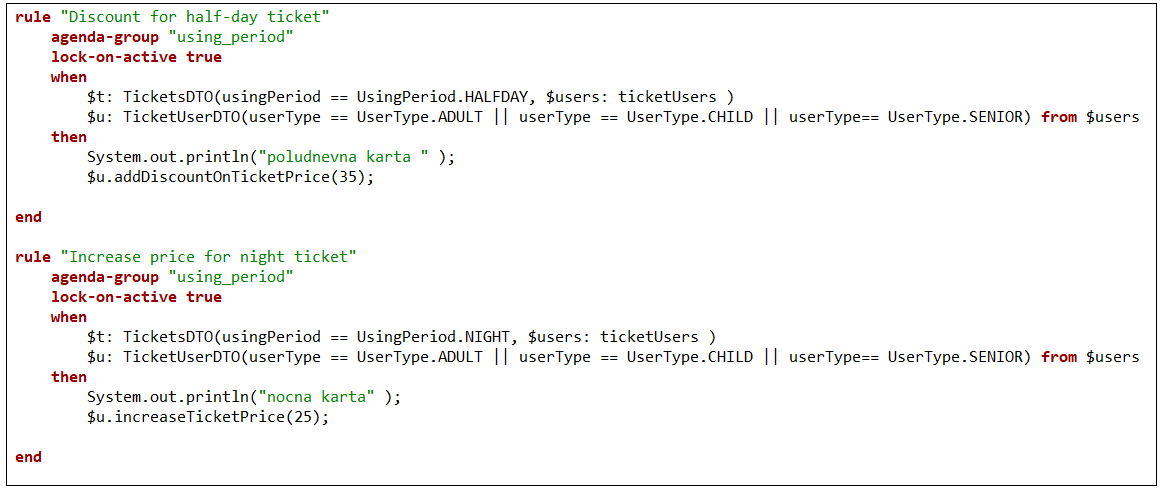
Правила су груписана у агенда групе*,* што нам омогућава да правила сврстамо у логички сродне групе и на тај начин ставимо фокус на извршавање одређене групе правила. Водећи се тиме, правила за израчунавање цене груписана су у следеће агенда групе:

* *transport\_type* (Слика 5.4):како свако скијалиште поседује иницијалну цену карте за коришћење ски лифта или гондоле, ова група правила на основу одабраног типа транспорта, сваком кориснику карата поставља почетну цену карте која ће се даље прилагођавати унетим захтевима.



Слика 5.4 Правила за одабран тип транспорта

* *using\_period:* цена карте је иницијално за дневне вожње. Уколико је корисник ипак одабрао полудневне или ноћне вожње, његова цена карте се умањује или повећава респективно. За полудневну карту цена је мања за 35%, док је за ноћну вожњу она скупља 25%.



Слика 5.5 Правила за одабрани период коришћења

* *type\_ticket* (Слика 5.6):систем генерише тип карте на основу броја корисника. Свака карта је иницјално појединачна. Да би карта била групна, мора да има бар 20 корисника, а да би била породична мора да задовољи услов да има бар два одрасла корисника и једно дете и да нема сениор корисника.



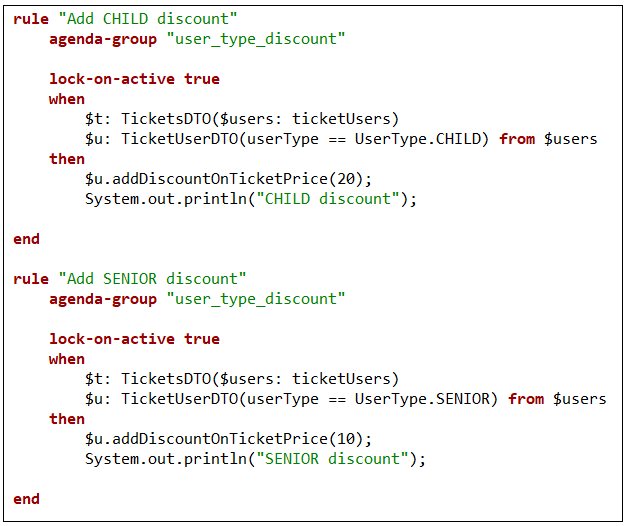
Слика 5.6 Правила која генеришу чињеницу типа карте

* *type\_ticket\_discount* (Слика 5.7): различити типови карата доносе одређене погодности, па тако на основу изгенерисане чињенице типа карте додељује се попуст. Уколко је карта групна, њена вредност се умањује 10%, а ако је породична 15%. Додатни попуст се остварује уколико породична карта има корисника са *loyalty* погодностима. Онда се одобрава још 15% попуста за сваког корисника.



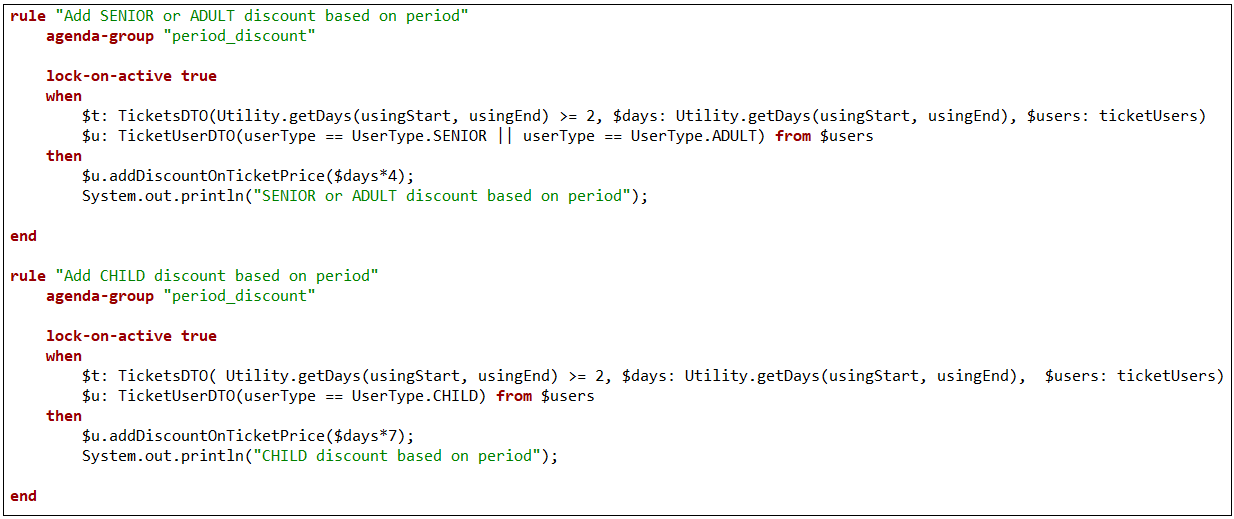
Слика 5.7 Правила за одобравање попуста на основу типа карте

* *user\_type\_discount* (Слика 5.8): иницијална цена карте односи се на корисника карте типа одрасли (енгл. *Adult*), с тога ова група правила умањује цену карте за 20% ако је корисник карте дете, или 10% ако је корисник сениор.



Слика 5.8 Правила за умањење цене на основу типа корисника

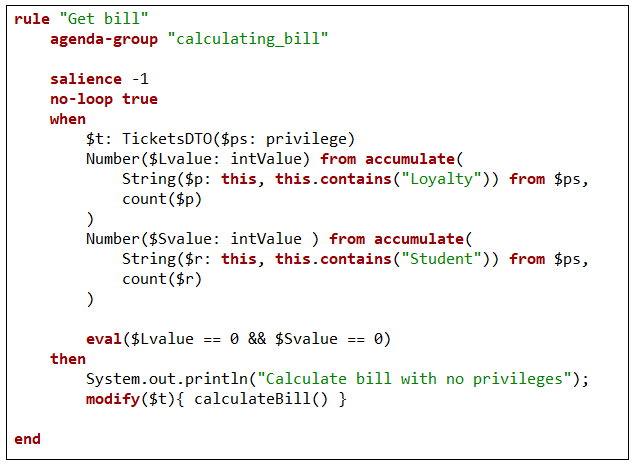
* *period\_discount* (Слика 5.8): уколико се карте купују за минимум два дана коришћења, њихова цена се умањује. Проценат умањења цене за сениоре и одрасле се рачуна по формули бр\_дана \* 4, а за децу бр\_дана \* 7.



Слика 5.9 Правила за попуст на дужину периода коришћења карте

* *calculating\_bill:* након што је формирана коначна цена карте која се односи на један дан, систем рачуна укупну цену за све дане коришћења. У овој групи правила утврђујемо да ли корисници карата имају одређене привилегије, тј. да ли су студенти или поседују *loyalty* погодности.

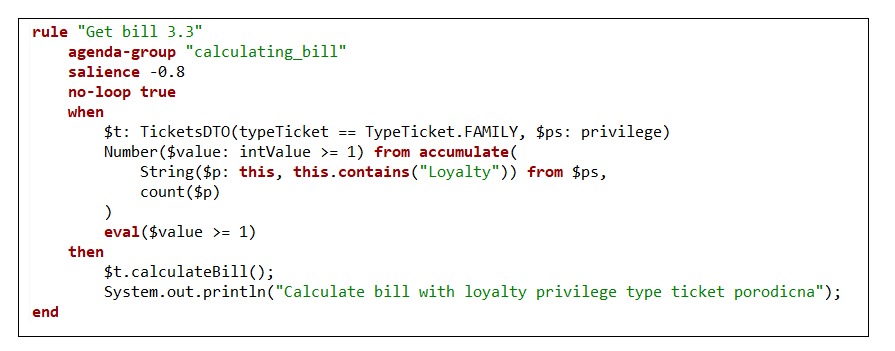
Уколико нема никаквих привилегија (Слика 5.10) укупна цена се рачуна на једноставан начин, позивањем методе *calculateBill()* која иницијалну цену карте за сваког корисника, која је претходно утврђена проласком кроз правила, помножи са бројем дана.



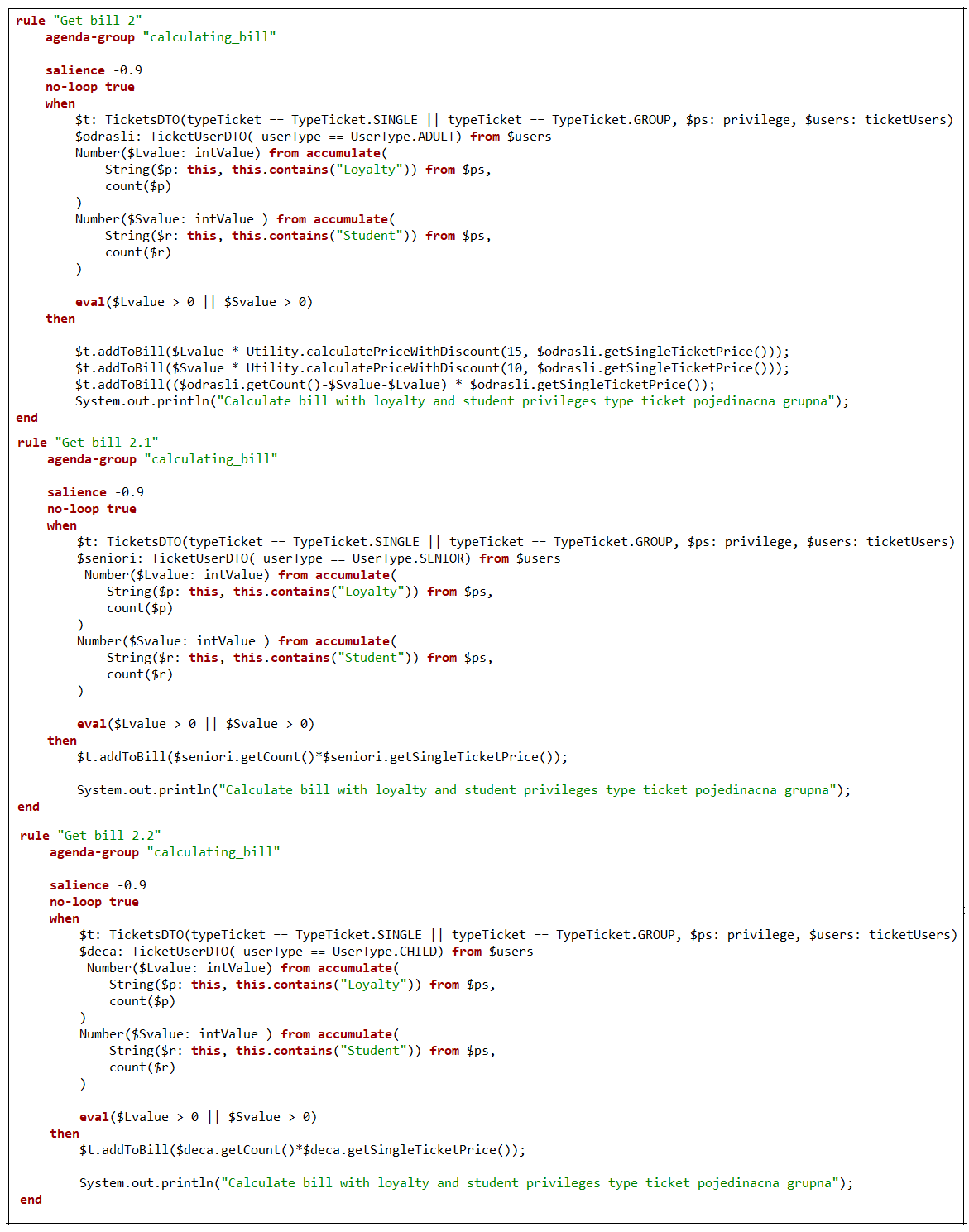
Слика 5.10 Рачунање укупне цене за кориснике без привилегија

Ако неко од корисника има одређене погодности, и ако је у питању појединачна или групна карта (Слика 5.12), то значи да ће само одређени корисници добити право на те погодности, тј. да ће попуст бити примењен на онај број карата колико постоји привилегија. Слично важи и уколико је тип карте породична, а неко од корисника има право на студентски попуст (Слика 5.13). Тада се умањује онолико иницијалних цена карата, чији су корисници типа одрали, колико има права на студентски попуст. Сениори и деца немају права на студентски попуст.

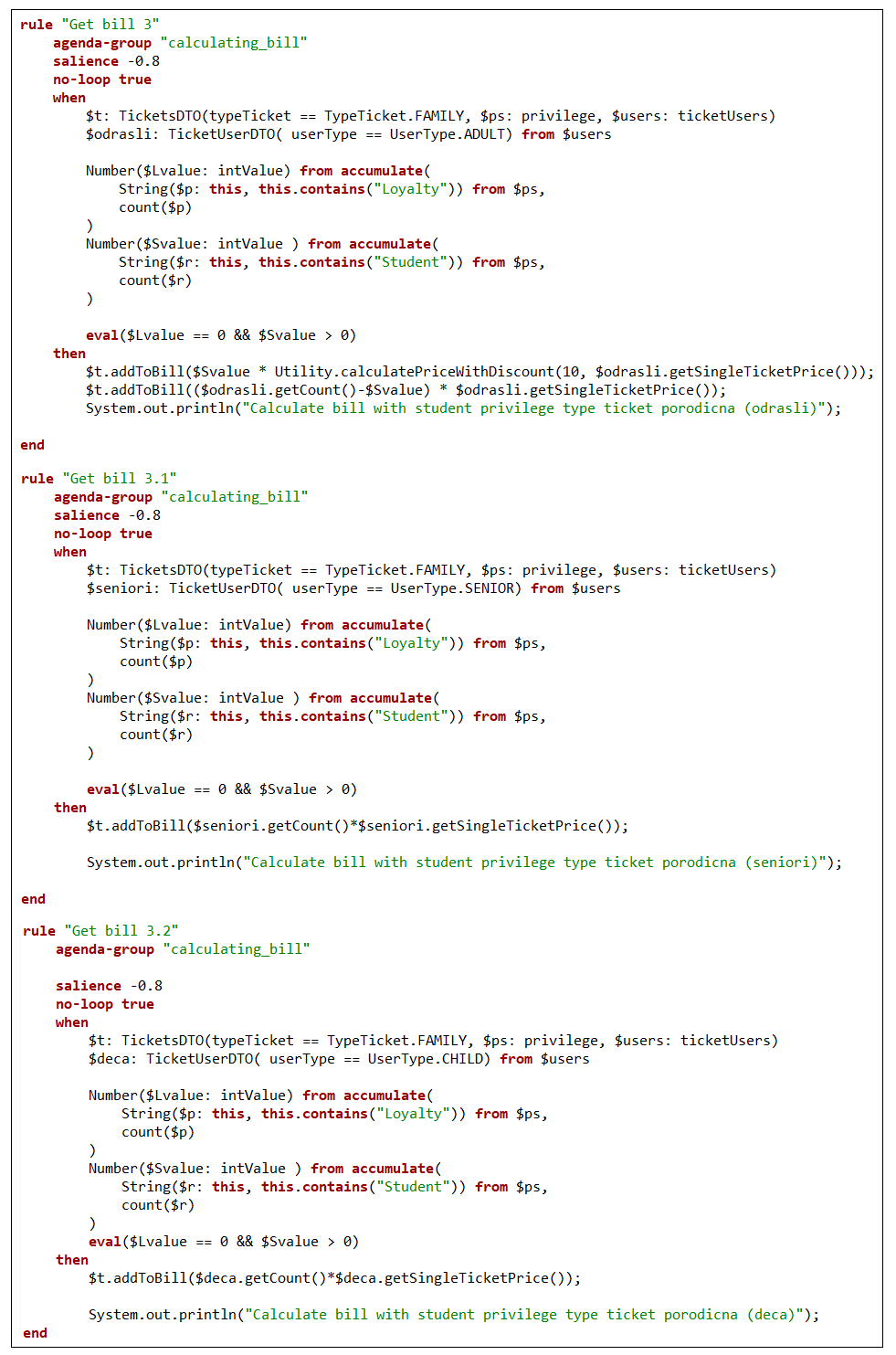
Изузетак је *loyalty* погодност када је породични тип карте. Тада се попуст обрачунава на крају и односи се на цео рачун. Правило за то приказано је на слици 5.11.



Слика 5.11 Правило за породични тип карте са loyalty привилегијама

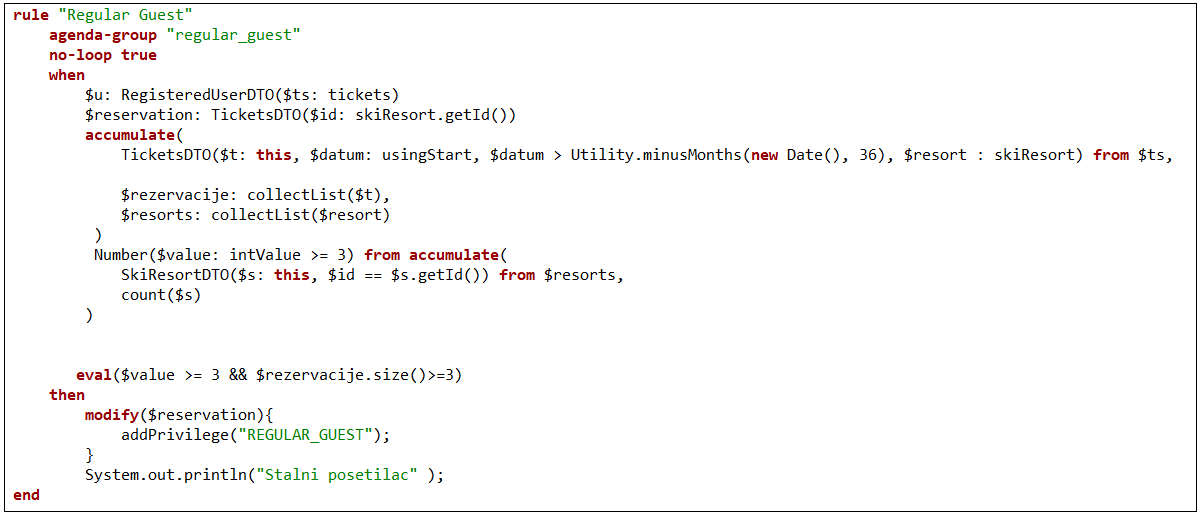


Слика 5.12 Правила за рачунање укупне цене карата типа појединачна или групна са привилегијама



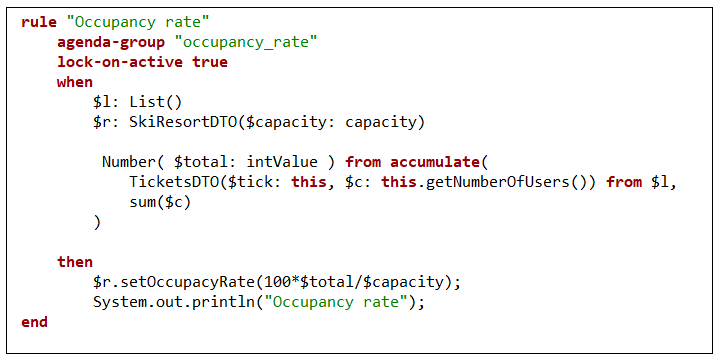
Слика 5.13 Правила за рачунање укупне цене карата типа породична са студентским попустом

* *regular\_guest* (Слика 5.14): правило се односи на формирање чињенице да је улоговани корисник који купује кате редован гост конкретног скијалишта за које купује карте. Овај статус му се додељује као привилегија која му може донети одређене бенефите. Да би корисник био редован гост неког скијалишта, неопходно је да је за то скијалиште купио бар три пута карте у последње три године.



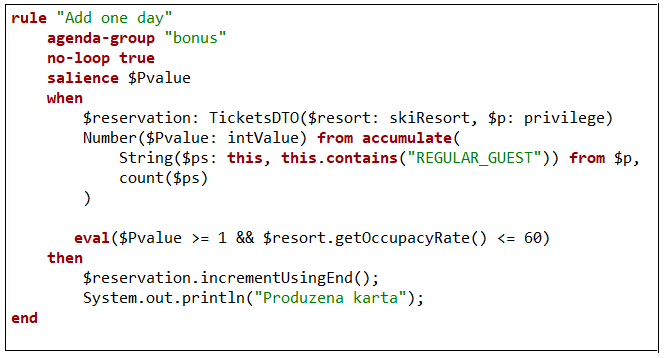
Слика 5.14 Правило за генерисање чињенице да је корисник редован гост скијалишта

* *occupancy\_rate* (Слика 5.15): на основу резервисаних карата, ово правило рачуна колика је посећеност скијалишта у процентима за неки конкретан датум.



Слика 5.15 Правило за рачунање процента посећености скијалишта

* *bonus* (Слика 5.16): ово правлио доноси одлуку о продужетку важења карте за још један дан уколико су испуњени услови, а то је, да корисник има привилегију сталног посетиоца (енлг. *Regular guest*) и да посећеност скијалишта за тај дан не прелази 60%.



Слика 5.16 Правило продужетка важења карте

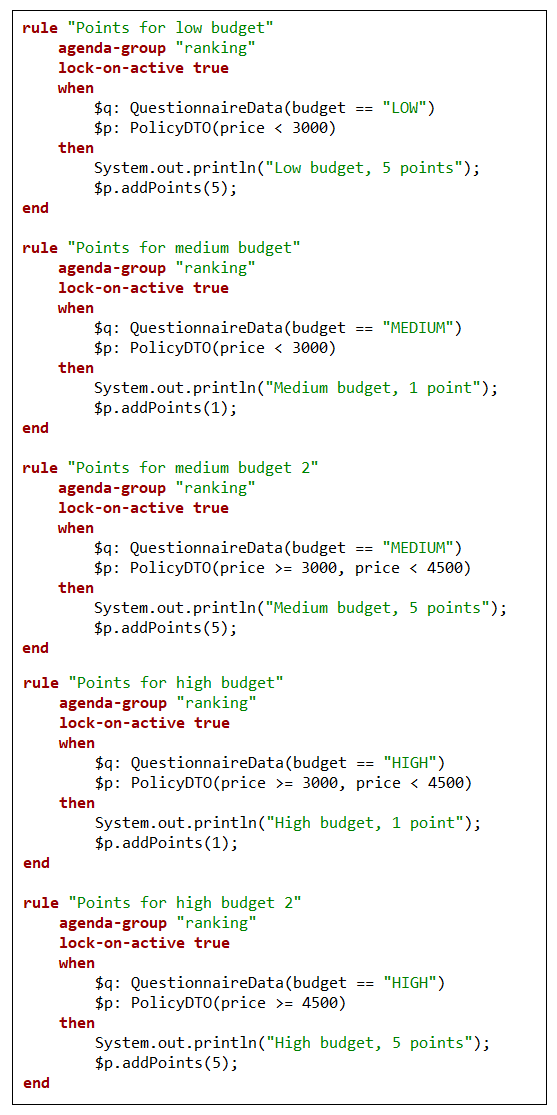
### Правила за препоруку полисе осигурања

Како би полисе осигурања што боље одговарале корисниковим захтевима и способностима, од њега се захтева попуњавање упитника. Упитник садржи одговоре на питање и која су од значају за процену. И упитник и правила су пажљиво осмишљена пратећи изведене закључке у истраживањима која се баве моделом повреда у скијању и сновбордингу (поглавље ). Од корисника се захтева да унесе:

* процену буџета
* одабере осигуравајуће куће уколико има неких преференција
* да ли је скијаш или сновбордер
* да оцени своју вештину скијања
* да унесе каква му вожња одговара
* коју стазу најчешће користи
* да ли користи заштитну опрему и коју
* да ли је претходно имао повреду на стази
* и да унесе његово тренутно здравствено стање у смислу да ли пати од неких урођених деформитета, да ли је имао преломе, операције итд.

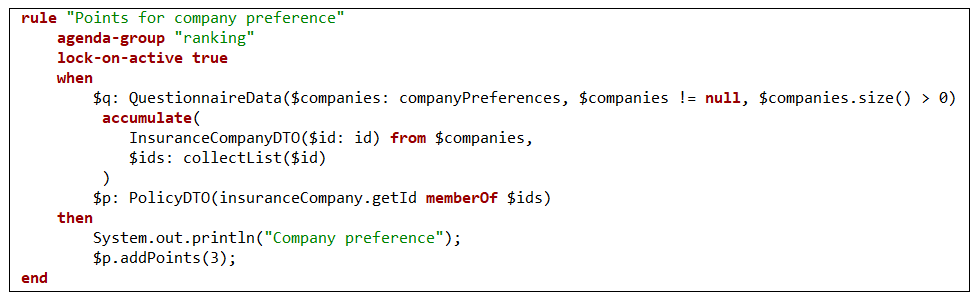
Сва правила за препоруку полисе осигурања груписана су у једну агенда групу под називом *ranking*. У преводу са енгленског – рангирање, што нам директно указује да ће та правила рангирати полисе на основу критеријума индиректно задатих од стране корисника.

Корисник уноси буџет којим располаже. Његов буџет може бити низак, средњи или висок. У зависности од тога одређеним полисама се додају поени на основу којих ће се она касније рангирати (Слика 5.17). Како су цене изражене у динарима, процењено је да полисе чија вредност не прелази 3000 динара одговарају ниском буџету, полисе чија је цена између 3000 и 4500 динара припадају средњем буџету, а изнад тога је висок буџет.



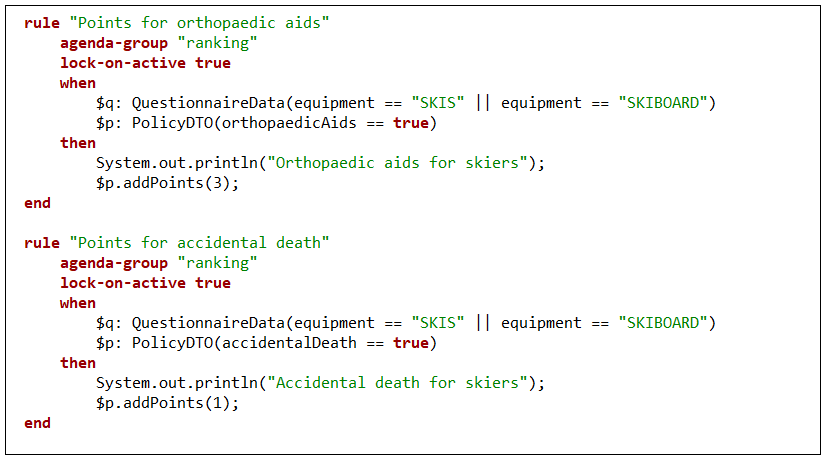
Слика 5.17 Правила за рангирање полиса у зависности од буџета

Уколико корисник има преференција ка одређеним осигуравајућим кућама, полисама које им припадају биће додељени додатни поени који ће их подићи на више ранг место.



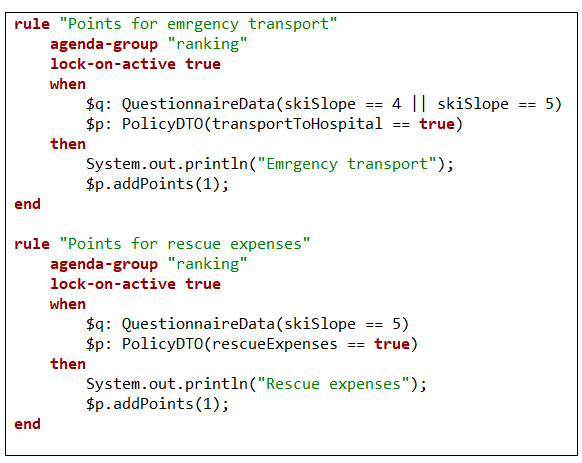
Слика 5.18 Правило за доделу поена полисама одговарајућих осигуравајућих кућа

Како су резултати истраживања показали да су повреде код скијаша углваном повреде доњих екстремитета, а код сновбордера горњих, прво правило на слици 5.19 додаје поене полисама које покривају ортопедска помагала уколико се корисник изјаснио као скијаш. Друго правило на слици 5.19 додаје поене полисама које покриваје осигурање у случају изненадне смрти. Ово правило додаје само 1 поена пошто смртни исход није тако чест, али су повреде скијаша озбиљнији и већа је смртност у односу на сновбордере.



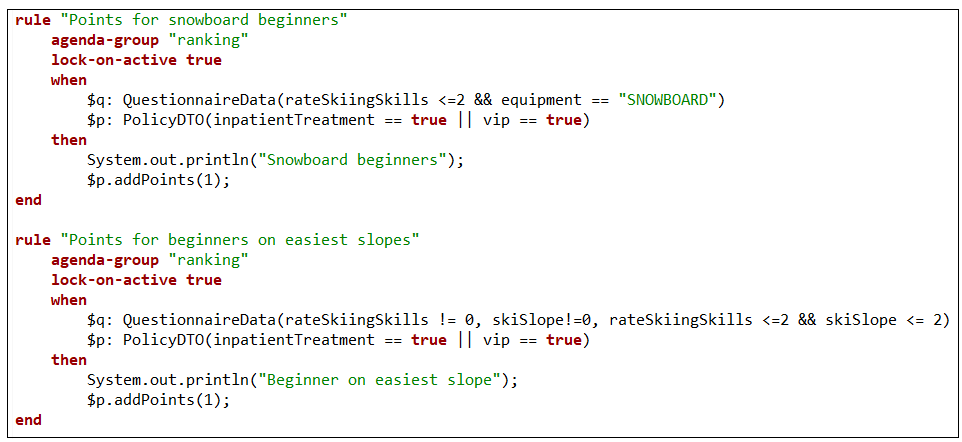
Слика 5.19 Правила која додају поене полисама за ортопедска помагала и изненадну смрт

У зависности од тога коју стазу корисник преферира, и вођени закључком да је хитан транспорт био потребан за скијаше или бордере на екстремим стазама и слободним теренима, правила на слици 5.20 додају поене полисама које покривају трошкове хитног транспорта и спашавања уколико корисник слободне терене за скијање.



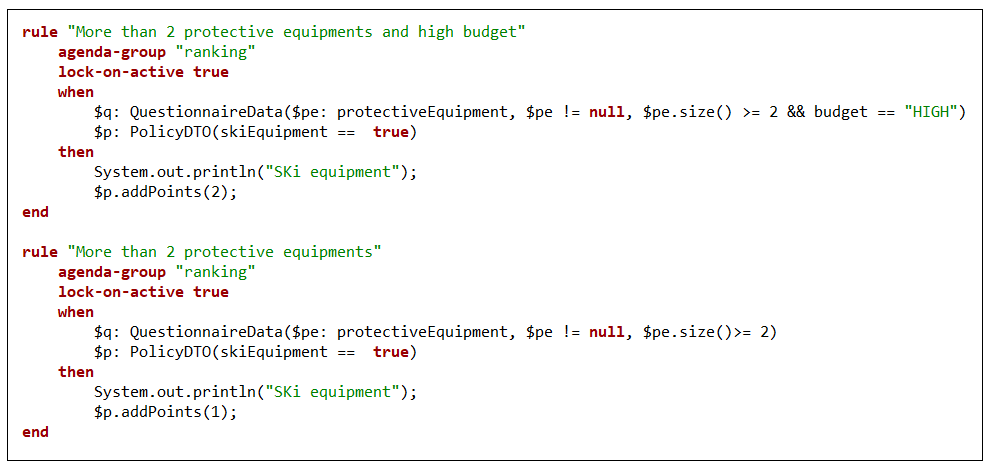
Слика 5.20 Правила која додају поене полисама које покривају трошкове транспорта и спашавања

На основу податка да је чак 49% повреда код сновбордера почетника, прво правило на слици 5.21 додаје поене полисама које покривају стационарно лечење за сновбордере који су проценили своје вештине као „не умем да скијам“ и „почетник“. Друго правило на слици 5.21 додаје поене истим полисама уколико је корисник на исити начин проценио своје вештине али и ако користи најлакше и мало теже стазе тј. стазе са благим и умерено стрмим нагибом. Баш на тим стазама догодио се највећи проценат повреда 47,3 % и то од стране почетника, чак 73% повреда направили су почетници у скијању или сновбордингу.



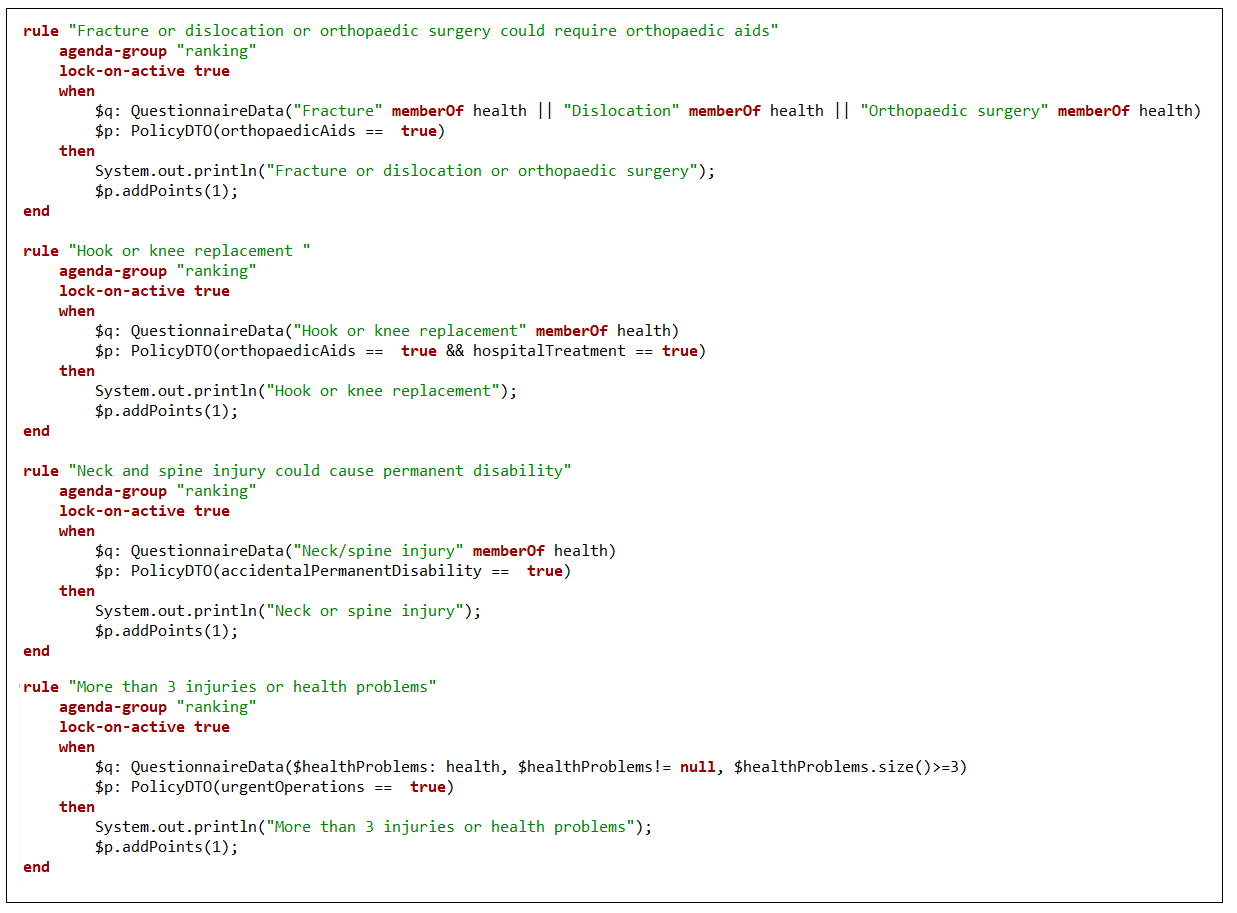
Слика 5.21 Правила које додају поене полисама које покривају стационарно лечење

Ако је корисник означио да користи заштитну опрему, наве бар две заштите и уколико је његов буџет висок, два поена ћемо додати полисама које осигуравају и опрему, у супротном ако нема висок буџет, онда не фаворизујемо полисе са 2 поена, већ додајемо 1 поен (Слика 5.22).



Слика 5.22 Правила која фаворизују полисе које осигуравају опрему

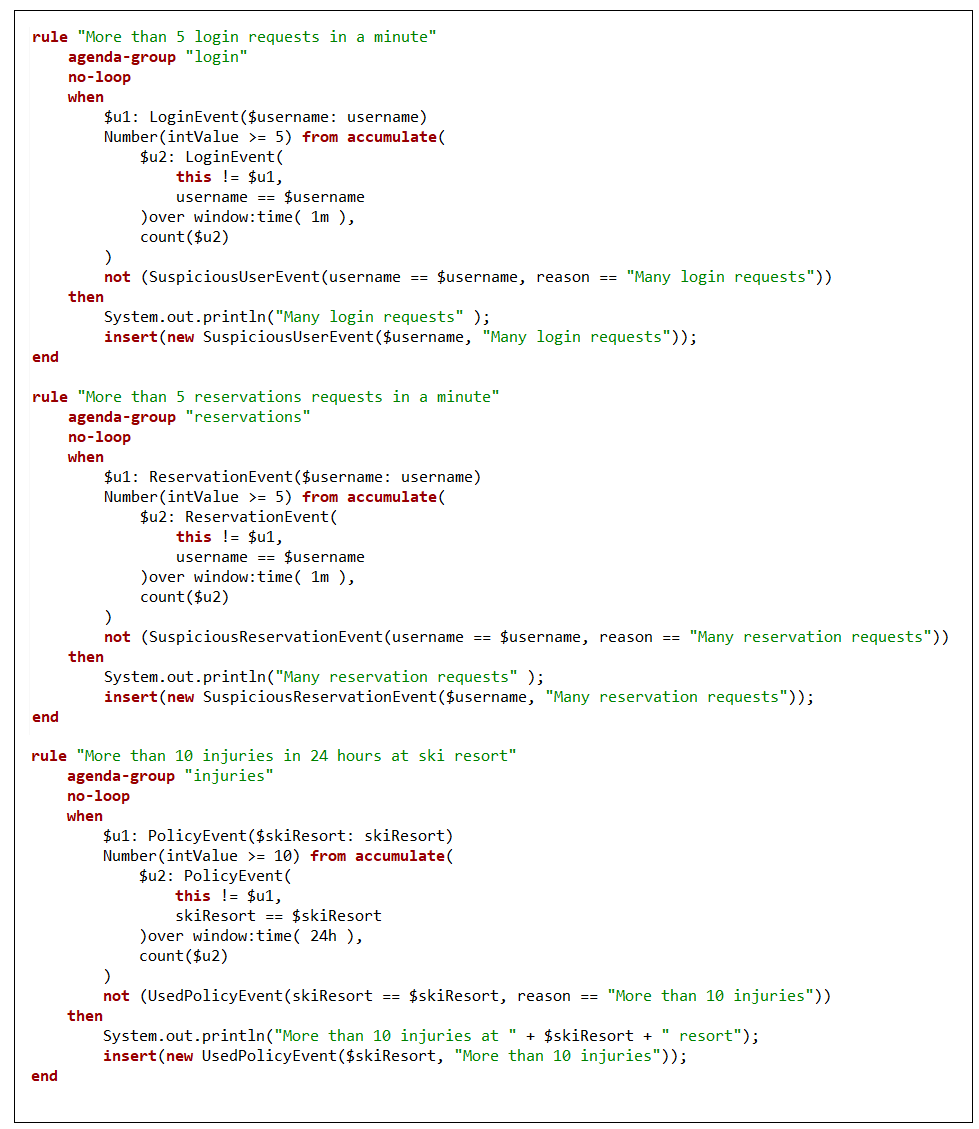
На слици 5.23 су приказана правила која тумаче здравствено стање скијаша / бордера и предност дају полисама које покривају ортопедска помагала, болничко лечење, хитне операције или исплату уколико дође до изненадне трајне непокретљивости. Уколико је корисник има претходно фрактуре, ишчашења или ортопедску операцију постоји вероватноћа да доћи до повреде и да би можда била неопходна ортопедска помагала па поене дајемо управо тим полисама. Сходно том закључивању, корисницима са вештачким куком или коленом може бити неопходно и болничко лечење. Ако је корисник имао повреду врата или кичме поене ћемо дати полисама које покривају изненадну трајну непокретљивост. И још ако је корисник пати од бар три раличита здравствена проблема, повреде и слично, можемо закључити да је склон повредама и да би му можда било неопходна и хитна операција.



Слика 5.23 Правила која тумаче здравствено стање корисника

### *CEP* правила

*CEP* правила служе нам да пратимо корисникове акције у систему и у опште акције које могу бити од значаја за наш систем. На слици 5.24 су приказана правила која се тичу сумњивих активности као што је неуспешно логовање више од 5 пута у једном минуту и више од 5 захтева за резервацију у једном минуту. Како администратор овог система прати и уноси податке везано за искоришћење полисе осигурања, ова правила бележе да је дошло до повреде на одређеном скијалишту. Уколико се десило више од 10 повреда на једном истом скијалишту у последња 24 сата емитује се догађај који сигнализира да можда постоји проблем на некој од стаза.

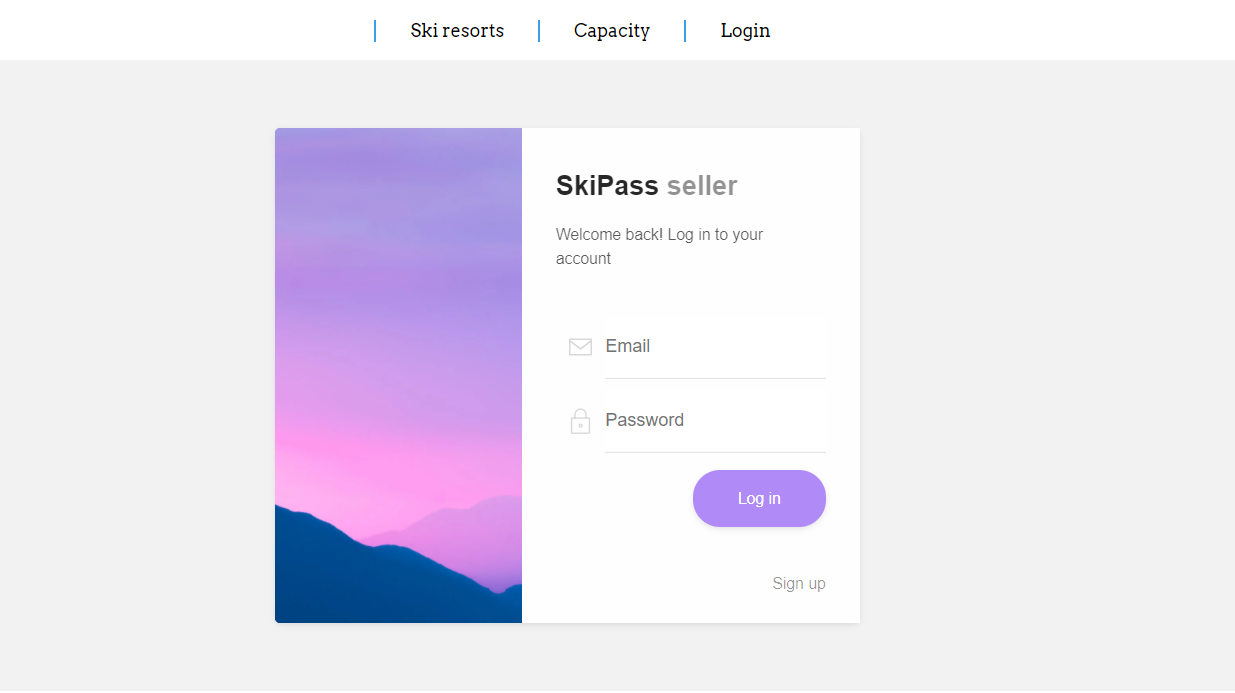


Слика 5.24 Правила која имплементирају CEP механизам

# ПРИКАЗ ИМПЛЕМЕНТИРАНОГ РЕШЕЊА

Корисници овог система могу бити неулоговани корисници, улоговани корисници и администратори. За сваког корисника персонализоване су странице у зависности од типа коме припадају. Оно што је заједничко за све, јесте страница за логовање, преглед свих доступних ски центара (енгл. *Ski resorts*) и увид у заузетост скијалишта за тренутни дан и наредна 4 дана.

Након приступа апликацији отвара се страница за логовање (Слика 6.1) где корисник уноси своје податке или одакле корисник може да пређе на страницу за регистрацију уколико нема креиран налог.

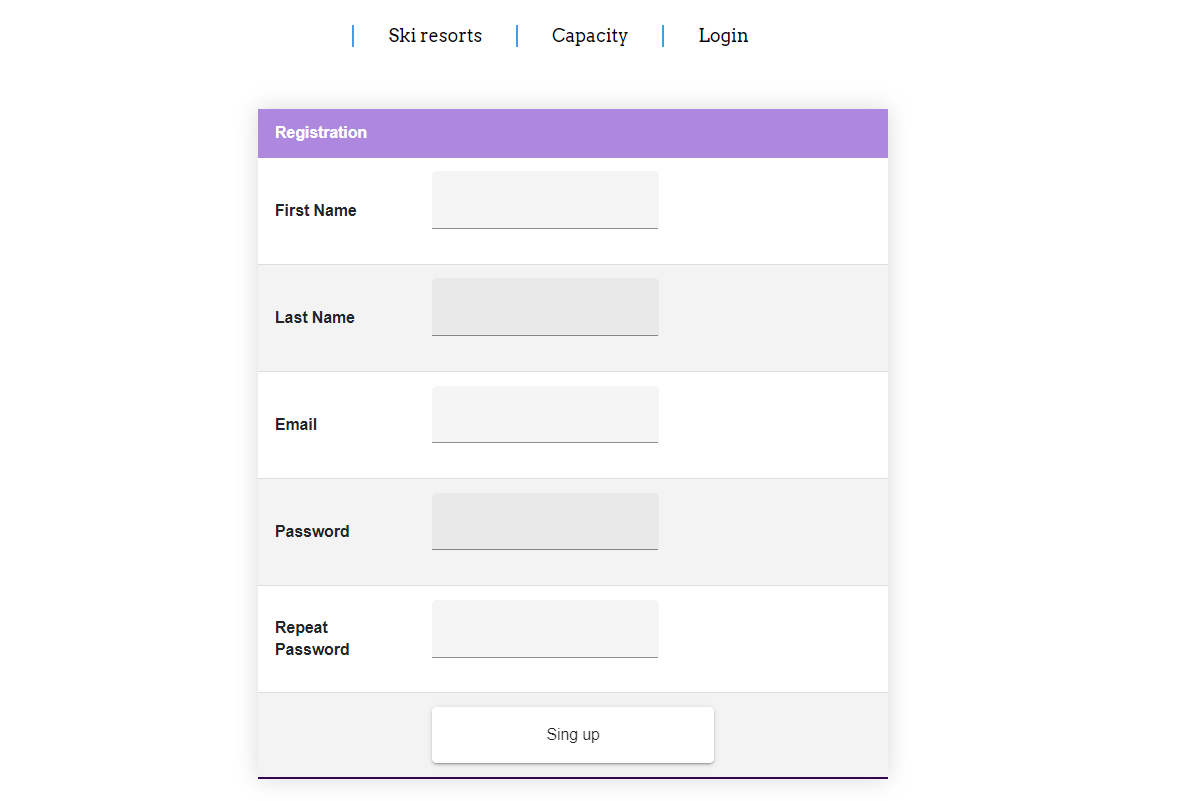


Слика 6.1 Страница за логовање

## Неулоговани корисник

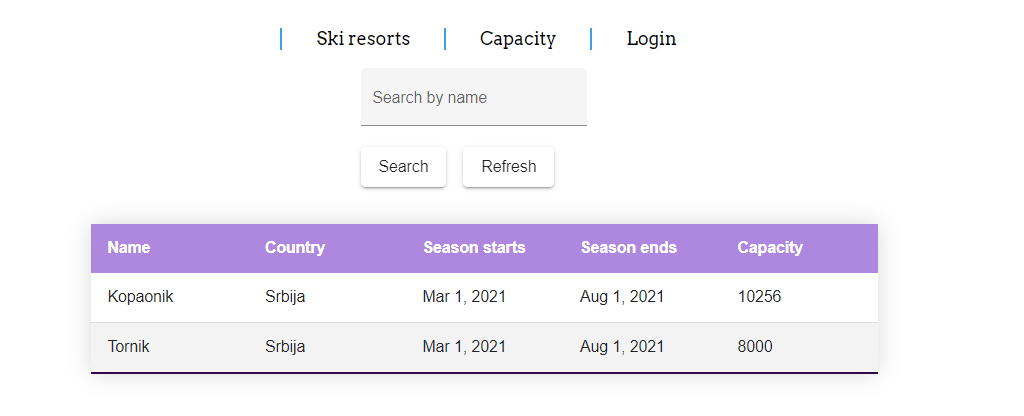
Неулоговани корисник има приступ једино горе наведеним страницама које су заједничке за сва три типа корисника. Да би корисник могао да се улогује, неопходно је да има креиран налог. Уколико нема налог потребно је да се региструје. Прелазак на страницу за регистрацију реализује се кликом на дугме „*Sign up“* које се налази на страници за логовање (Слика 6.1).

Кориснику се отвара страница са формом за унос података (Слика 6.2). Од корисника се захтева да попуни сва поља: име, презиме, емаил, лозинку и поновљену лозинку. Да би регистрација била успешна, емаил адреса треба да је јединствена у систему, тј. да не постоји налог са том емаил адресом. Кликом на дугме *„Sign up“* проверавају се унети подаци и корисник добија потврду о успешној или неуспешној регистрацији.

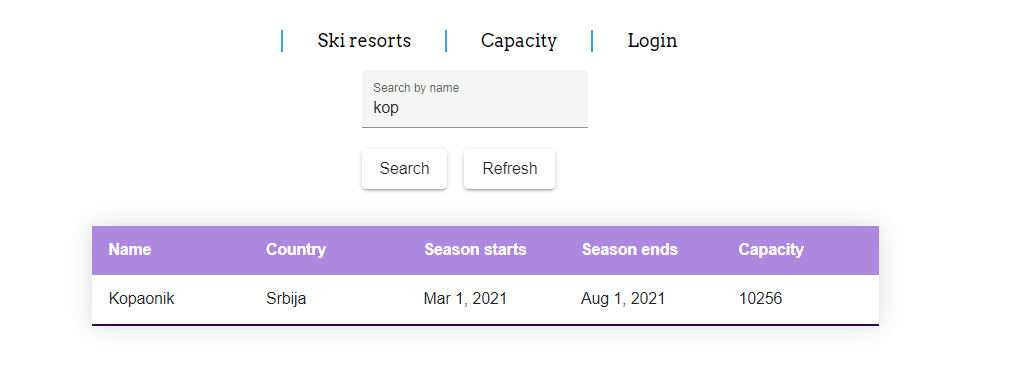


Слика 6.2 Страница за регистрацију

Преглед свих скијалишта за које је могуће направити резервацију, налази се на страници „*Ski resorts*“ одабиром на ту каартицу отвара се нова страница (Слика 6.3). Корисник има могућност претраге на основу имена, тако сто у поље за претрагу уноси име или део имена (Слика 6.4).



Слика 6.3 Страница за преглед скијалишта



Слика 6.4 Пример претраге

Кликом на дугме „*Refresh*“, поништавају се резултати претраге, и могуће је поново видети сва доступна скијалишта.

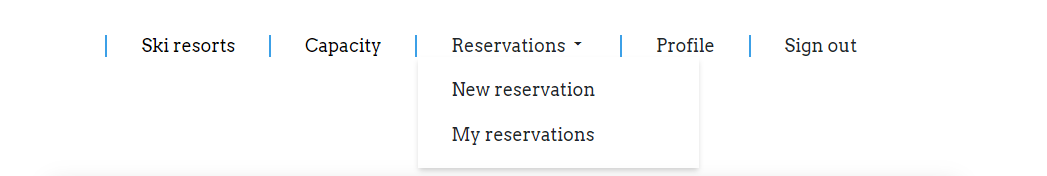
Одабиром картице „*Capacity*“ кориснику се приказује график који илуструје заузетост конкретног скијалишта. Преласком курсором преко графика за у зависности од позиције приказује се тачан проценат попуњеног капацитета.



Слика 6.5 Приказ попуњености конкретног скијалишта

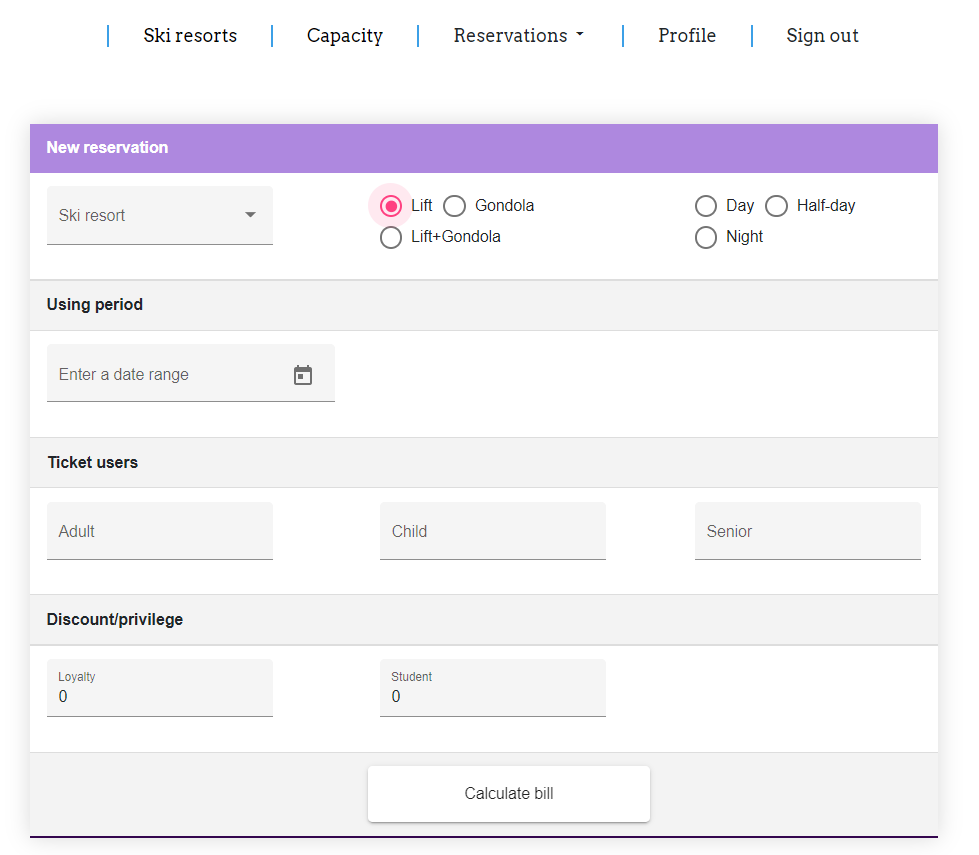
## Улоговани корисник

Након успешног логоавања (Слика 6.1), улоговани корсник добија приказ новог навигационог бара са њему доступним страницама и акцијама (Слика 6.6). Поред заједничких страница, улоговани корисник може да прегледа своје резервације или да направи нову резервацију. То може урадити одабиром на једне од две опције из падајућег менија који се налази под ознаком „*Reservations*“ у навигационом бару.



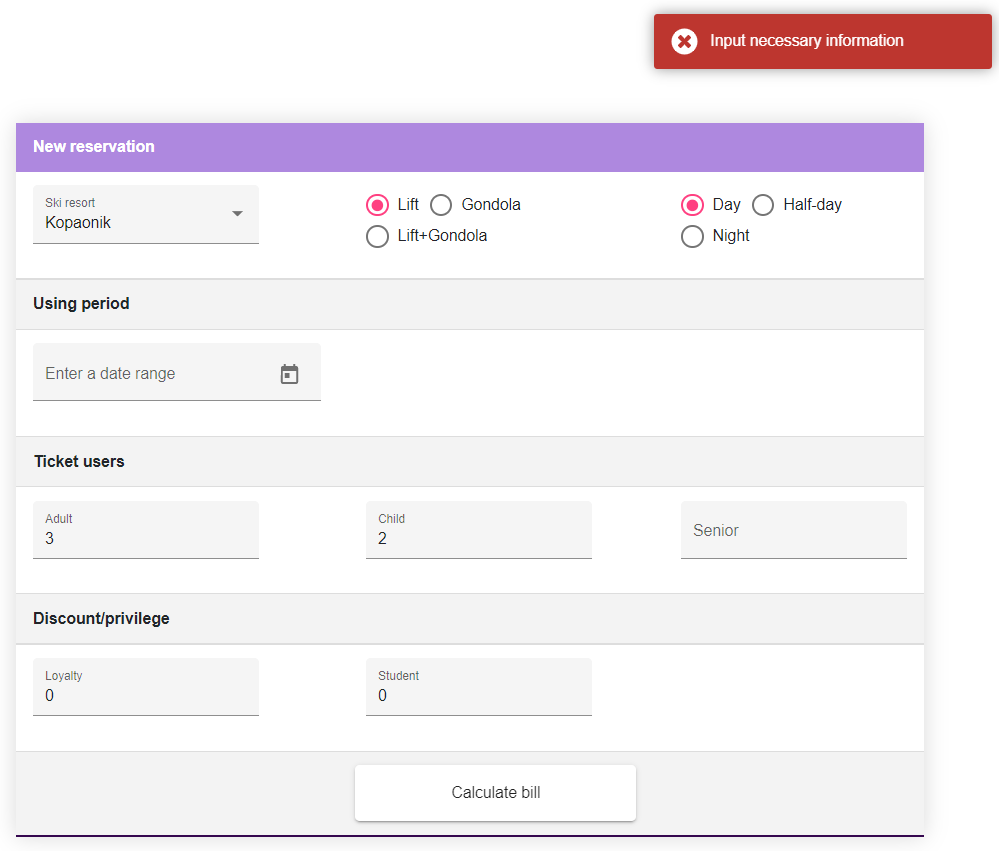
Слика 6.6 Навигациони бар улогованог корисника

Када корисник одабере опцију „*New reservation*“, одлази на страницу са формом за креирање нове резервације.



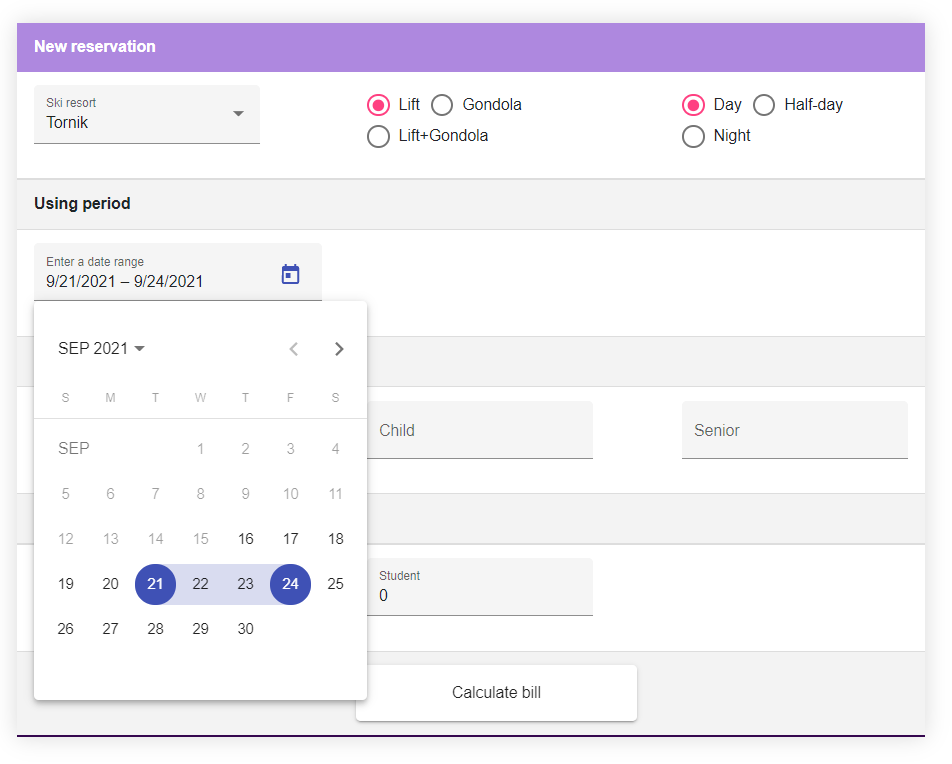
Слика 6.7 Страница за креирање нове резервације

Од корисника се захтева да унесе све потребне информације. Уколико корисник не унесе неку од потребих информација, добија поруку грешке која сигнализира да нису унети сви захтеви.



Слика 6.8 Пример неисправне форме

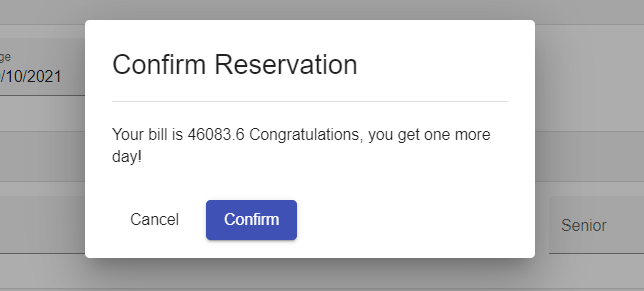
Корисник прво бира скијалиште за које жели да резервише карте, затим тип транспорта. Уколико је тип транспорта лифт онда се кориснику приказује и одабир периода коришћења карте, у смислу да ли жели дневну, полудневну или ноћну карту. Како свако скијалиште има свој почетак и крај сезоне скијања, корисник уз та ограничења бира од када до када ће користити своју пропусницу. Корисник не може да резервише карте за период који је већ прошао, а исто тако не може да резервише карте за датуме који су ван трајања сезоне за одабрано скијалиште. За потребе приказа имплементираног решења, почетак ски сезоне за скијалиште Торник је 16. септембар 2021. године. Пример уноса периода важења карата на слици 6.9.



Слика 6.9 Унос периода важења карата

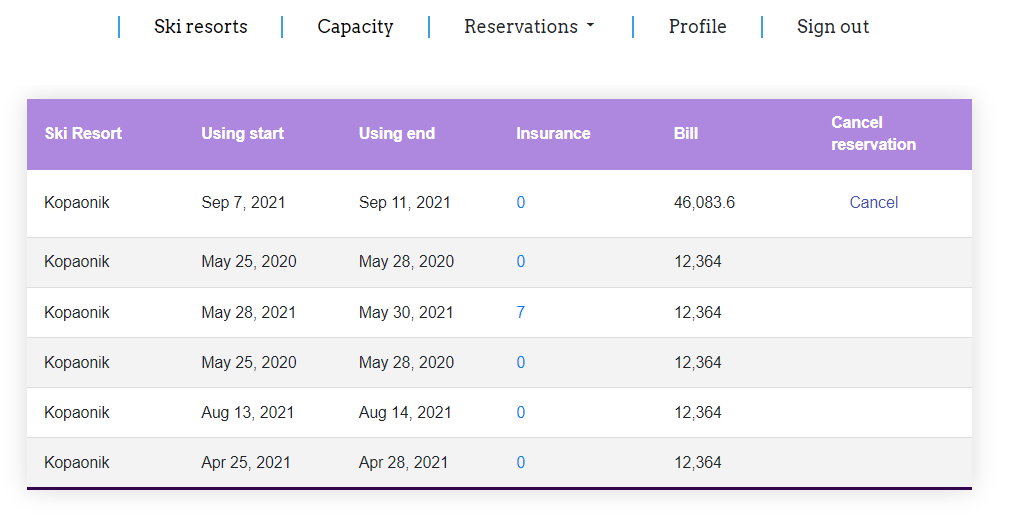
Даље корисник уноси колико жели да купи карата за одрасле, за децу или за сениоре. Уколико не унесе неки од тих података подразумевано је нула карата за те типове корисника. Ако неко од корисника карата има одређене привилегије, уноси се само број тих привилегија. На пример, ако два корисника имају права на студентски попуст, онда се у поље предвиђено за студентски попуст уноси број 2. Провера валидности тих права није предвиђена овом апликацијом.

Када унети подаци прођу кроз правила у *Drools* апликацији, корисник добија приказ укупног рачуна у дијалог прозору, и од њега се захтева потврда (Слика 6.10). Ако су правила утврдила да је овај корисник редован посетилац скијалишта и ако постоје услови за продужетак карте, корисник добија обавештење и о томе.



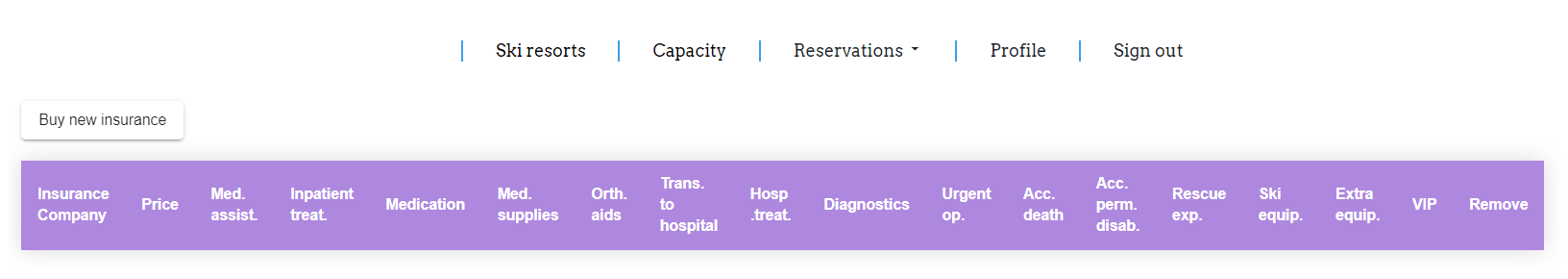
Слика 6.10 Приказ укупног износа рачуна

Када корисник потврди резервацију, он своје карте може да прегледа на страници „*My reservations*“ (Слика 6.11). Са те странице корисник за одабрану карту може да купи осигурање и може да откаже резервацију два дана пре почетка важења карата. Отказивање резервације се врши након потврде акције кроз дијалог прозор.



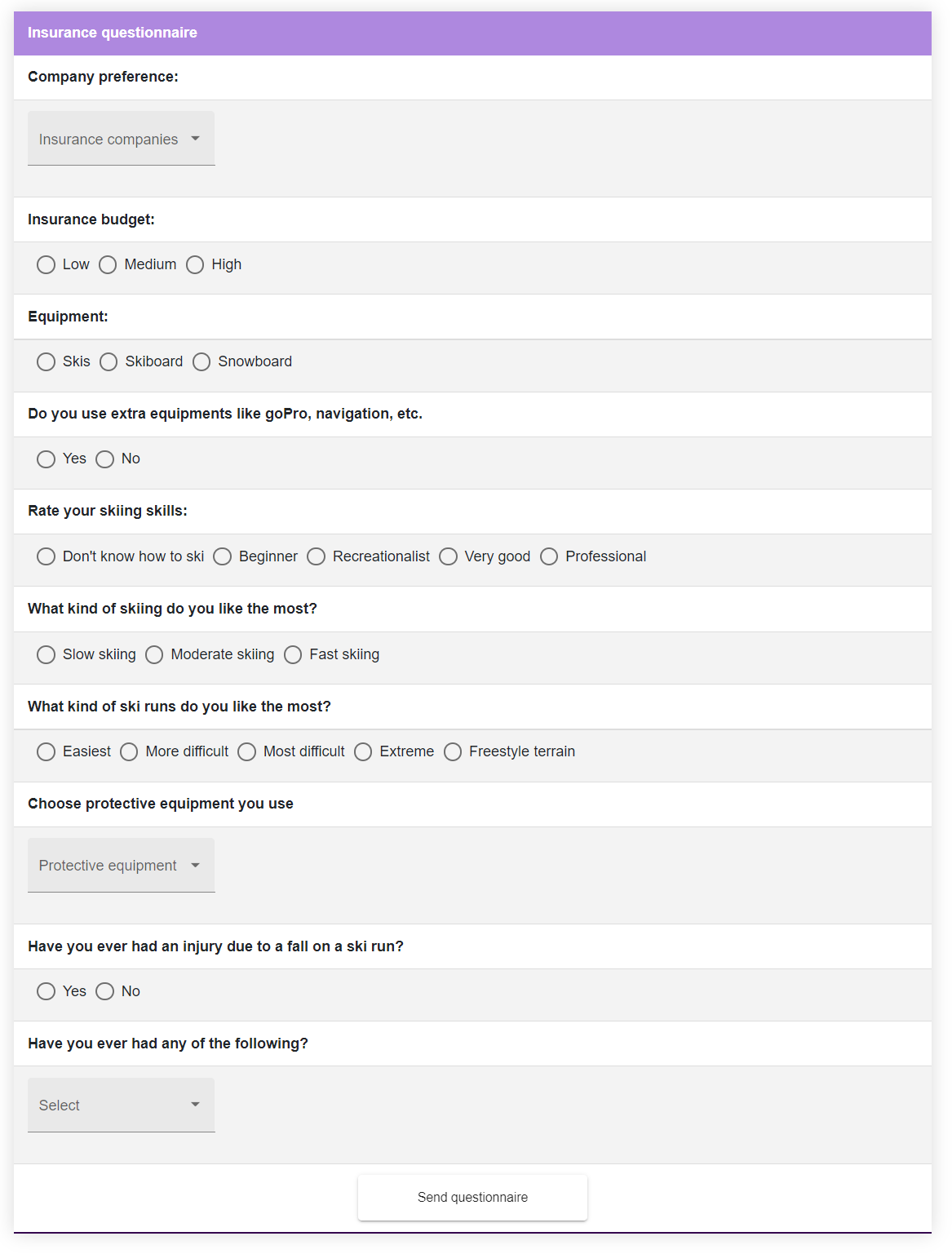
Слика 6.11 Страница за приказ свих резервација улогованог корисника

Кликом на број у табели који се налази у колони „*Insurance*“ за конкретну карту, отвара се страница за приказ купљених полиса за ту карту (Слика 6.12). Како за прву карту немамо купљених полиса, табела за приказ биће празна, а нама ће бити дозвољен клик на дугме „*Buy new insurance*“. Клик на то дугме дозвољено је два дана до почетка коришћења карата и ако постоје корисници карата за које још увек није купљена полиса.



Слика 6.12 Стрница за приказ купљених полиса

Кликом на дугме „*Buy new insurance*“ корисник се шаље на страницу за попуњавање упитника (Слика 6.13) на основу којег ће му бити препоручене полисе које се најбоље уклапају у његове захтеве.



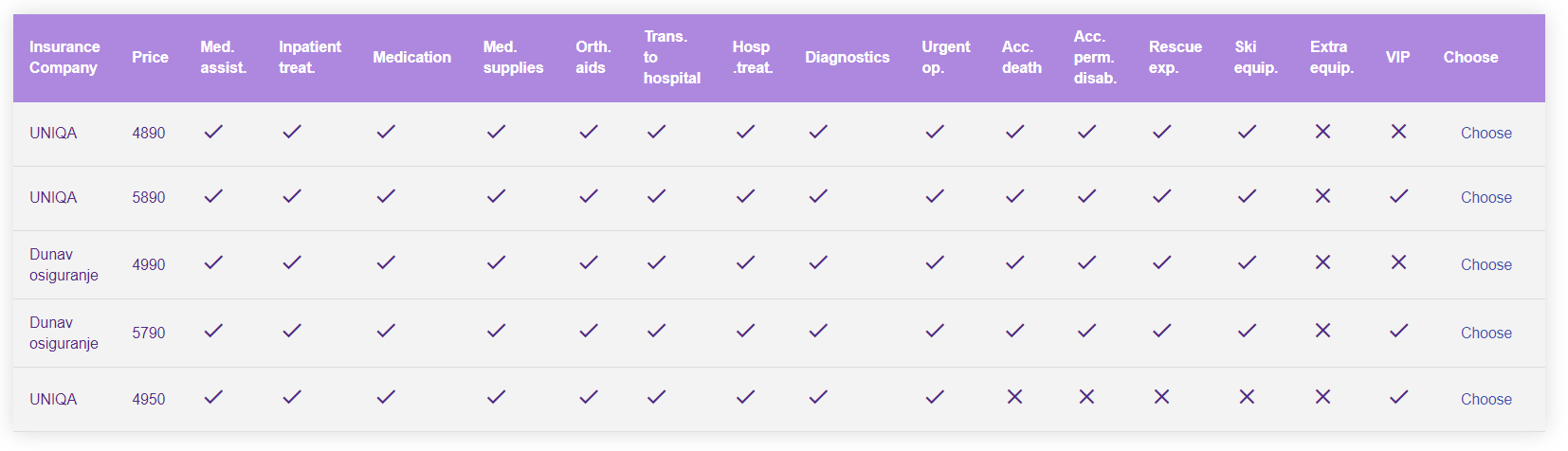
Слика 6.13 Упитник за препоруку полиса

Корисник уноси потребне информације, редом:

* Из листе доступних осигуравајућих кућа, бира компаније ка којима има преференција
* Процењује буџет (низак, средњи или висок)
* Уноси коју опрему користи (скије, кратке скије, сновборд)
* Да ли користи нешто од додатне опреме или додатних уређаја
* Процењује своје скијашке способности (не умем да скијам, почетник, рекреативац, веома добар, професионалац)
* Бира вожњу која му највише одговара (споро, умерено, брзо)
* Уноси терен који најчешће користи (лаке стазе, мало теже, тешке, екстремне, слободан терен)
* Ако користи заштитну опрему уноси шта користи
* Да ли је некада имао повреду на стази
* И уноси своје тренутно здравствено стање, бира из листе понуђених повреда, стања итд.

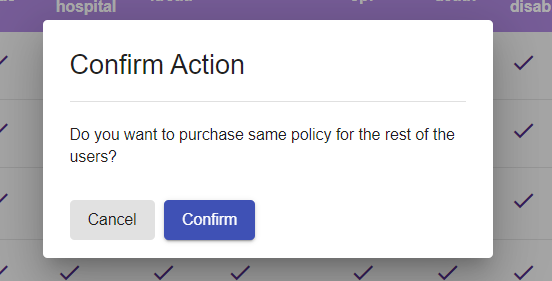
Корисник није у обавези да попуни сва поља.

Кликом на дугме „*Send questionnaire*“, упитник се обрађује и пролази кроз *Drools* апликацију и корисник добија листу првих 5 најбоље рангираних полиса (Слика 6.14).



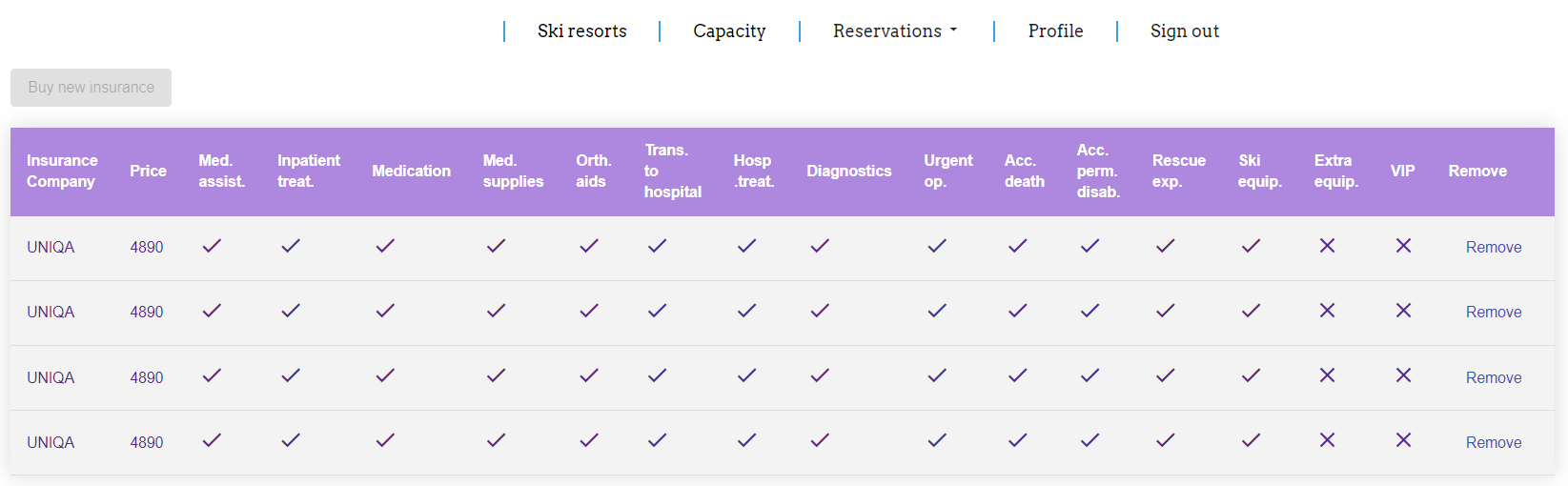
Слика 6.14 Табела предложених полиса

Кликом на дугме „*Choose*“ корисник купује одабрану полису. Њему се нуди избор да одабрану полису купи само једном или да купи и за све остале кориснике карата. Потврдом акције са слике 6.15, корисник је купио онолико полиса колико је остало корисника који немају полису.



Слика 6.15 Куповина полисе и за остале кориснике карата

Сада страница за приказ свих купљеих полиса изгледа као на слици 6.16. Корисник сада више нема могућност куповине нових полиса, и може уколико жели, да уклони неку од купљених полиса кликом на дугме „*Remove*“ и потврдом акције уклањања.



Слика 6.16 Приказ купљених полиса

## Администратор

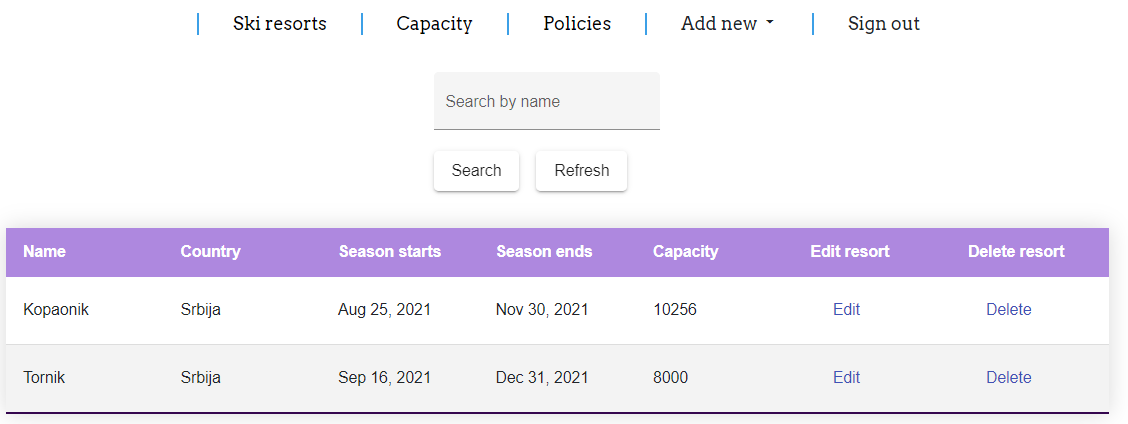
Поред претходно описаних заједничких страница и фукционалности, администратор има и додатне опције на тим страницама, као што је уређивање конкретног скијалишта и претрага попуњености скијалишта за конкретан датум, а не за тренутних 5 дана као што је случај код неулогованог и улогованог корисника. Његов навигациони бар приказан је на слици 6.17.



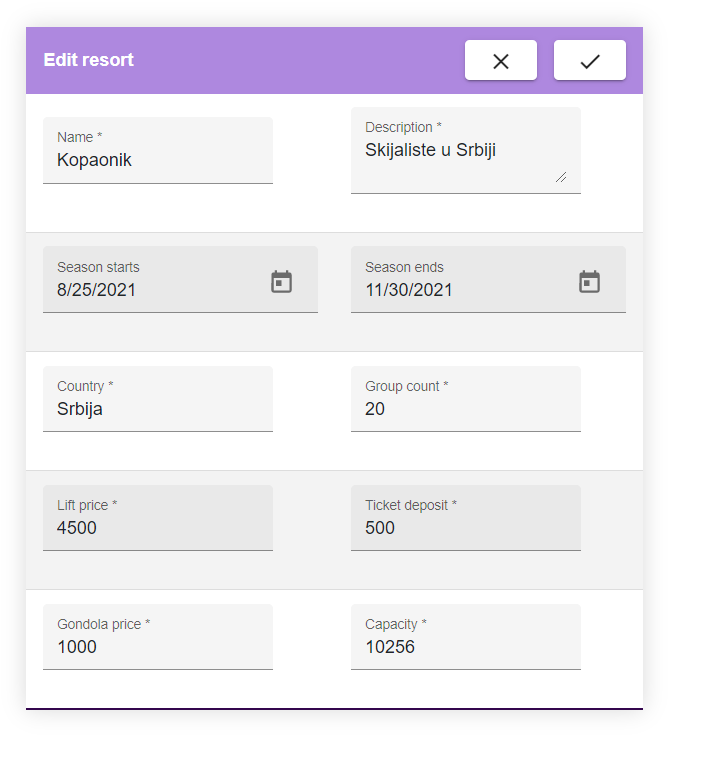
Слика 6.17 Навигациони бар за администратора

У табели у којој су приказана сва скијалишта, а која се налази на страници „*Ski resort*“ (Слика 6.18), администратор има могућност да измени податке неког конкретног скијалишта кликом на опцију „*Edit*“ које одговара колони жељеног скијалишта илл да на исти начин обрише неко скијалиште. Брисање се реализује тек након потврде акције кроз дијалог прозор.

Након што је администратор одабрао опцију едитовања, отвара му се нова страница са формом за едитовање (Слика 6.19). Та форма у себи садржи тренутне податке који се односе на конкретно скијалиште, а које админ може да измени.



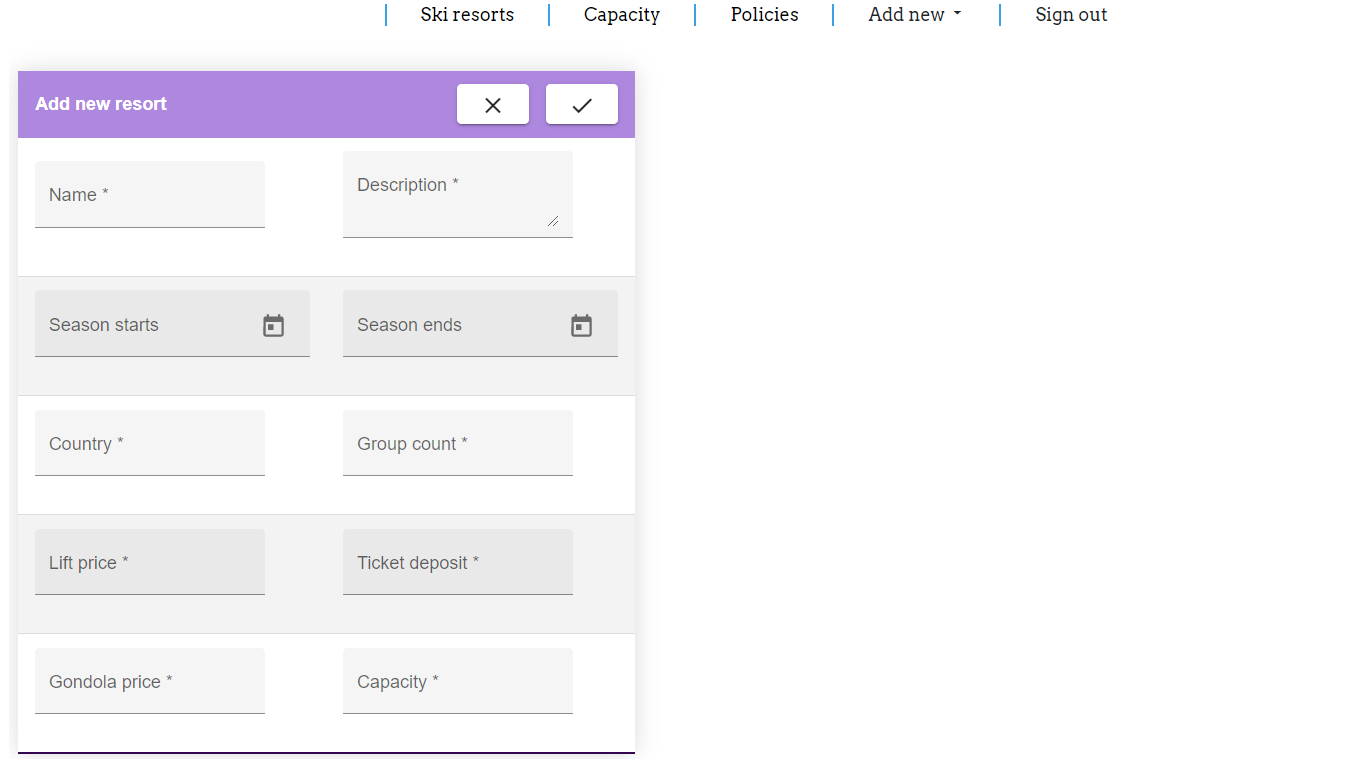
Слика 6.18 Админов приказ свих скијалишта



Слика 6.19 Форма за едиваање података о ски центру

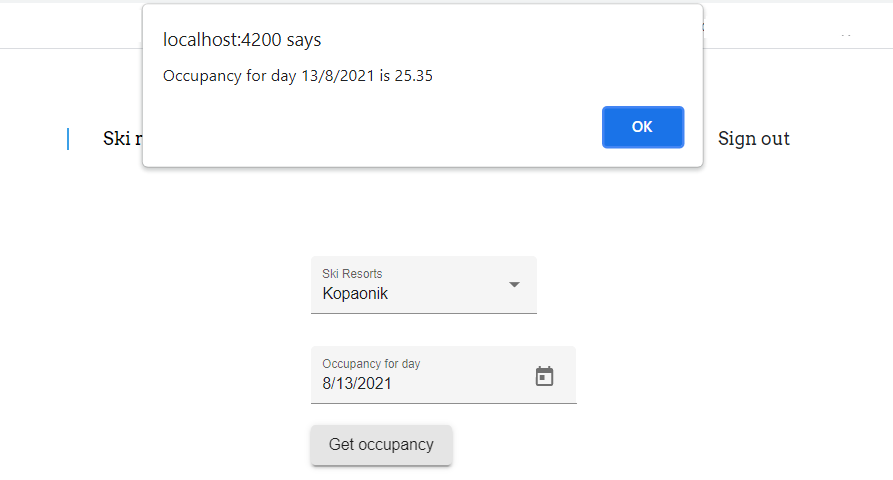
Кликом на дугме са симболом „🗸“ које означава потврду унетих информација, подаци бивају измњени и сачувани у бази података. Уколико је чување било успешно, администратор добија поруку да су подаци успешно сачувани, исто важи и за неуспешно чување. Кликом на дугме са симболом „🗴“ промене се одбацују.

Одабиром опције „*Add new ski resort*“ отвара се страница за унос података о новом скијалишту (Слика 6.21). Страница садржи исту форму као и за едитовање, с тим што је сада ова форма празна и од администратора се очекује да попуни сва поља. Администратор мора да води рачуна да је име скијалишта јединствено у систему.



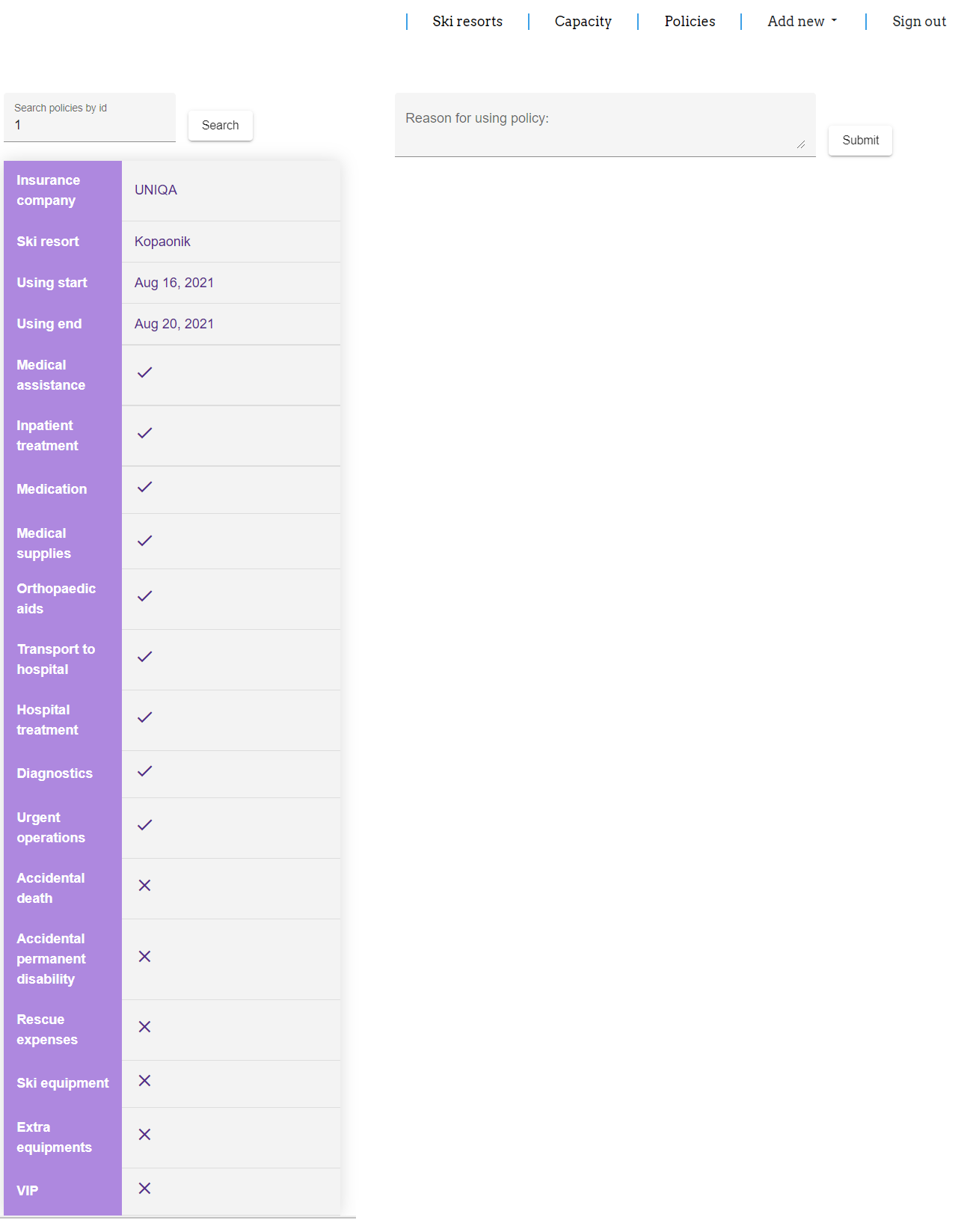
Слика 6.20 Страница за унос података о новом скијалишту

На страници за приказ попуњености скијалишта, поред стандардног графика који садржи данашњи дан и наредна 4 дана, администратор може да за одабрано скијалиште унесе и датум и тако провери попуњеност капацитета за тај дан (Слика 6.21).



Слика 6.21 Приказ заузећа скијалишта у процентима

Администратор у систем уноси податке о искоришћењу осигурања тј. наводи разлог зашто је нека полиса искоришћена. Та акција се релизује на страници „*Policies*“ (Слика 6.22). Киком на ту картицу из навигацион бара, отвара се страница где администратор претражује купљене полисе по јединственом идентификатору. Након успешне претраге, администратор види све податке о тој полиси као и за које је скијалиште купљена, и на тај начин процењује да ли је полиса валидна.



Слика 6.22 Страница за претрагу купљених полиса

# ЛИТЕРАТУРА

1. Drools rule structure - <https://www.researchgate.net/figure/Drools-Rule-Structure_fig1_330211452>, преузето августа 2021.
2. JpaRepository - <https://docs.spring.io/spring-data/jpa/docs/current/reference/html/#jpa.repositories>, преузето августа 2021.
3. UML Class and Object Diagrams - <https://www.uml-diagrams.org/class-diagrams-overview.html>, преузето августа 2021.