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

Јелена Хрњак

**Наменски језик за генерисање *spring* апликације са безбедносном конфигурацијом**

Мастер рад

- Мастер академске студије -

Нови Сад, 2023.

|  |  |  |  |
| --- | --- | --- | --- |
| Редни број, **РБР**: | |  | |
| Идентификациони број, **ИБР**: | |  | |
| Тип документације, **ТД**: | | Монографска документација | |
| Тип записа, **ТЗ**: | | Текстуални штампани материјал | |
| Врста рада, **ВР**: | | Мастер рад | |
| Аутор, **АУ**: | | Јелена Хрњак | |
| Ментор, **МН**: | | др Владимир Димитриески, доцент | |
| Наслов рада, **НР**: | | Наменски језик за генерисање... | |
| Језик публикације, **ЈП**: | | Српски / ћирилица | |
| Језик извода, **ЈИ**: | | Српски | |
| Земља публиковања, **ЗП**: | | Република Србија | |
| Уже географско подручје, **УГП**: | | Војводина | |
| Година, **ГО**: | | 2023 | |
| Издавач, **ИЗ**: | | Ауторски репринт / Факултет техничких наука | |
| Место и адреса, **МА**: | | Нови Сад, Трг Доситеја Обрадовића 6 | |
| Физички опис рада, **ФО**: (поглавља/страна/ цитата/табела/слика/графика/прилога) | | 6/45/0/50/36/0/0 | |
| Научна област, **НО**: | | Електротехничко и рачунарско инжењерство | |
| Научна дисциплина, **НД**: | | Примењене рачунарске науке и информатика | |
| Предметна одредница/Кqучне речи, **ПО**: | | Доменски оријентисано моделовање и језици | |
| **УДК** | |  | |
| Чува се, **ЧУ**: | | Библиотека Факултета техничких наука, Нови Сад | |
| Важна напомена, **ВН**: | |  | |
| Извод, **ИЗ**: | | TODO | |
| Датум прихватања теме, **ДП**: | |  | |
| Датум одбране, **ДО**: | |  | |
| Чланови комисије, **КО**: | Председник: |  |
|  | Члан: |  | Потпис ментора |
|  | Члан, ментор: | др Владимир Димитриески, доцент |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Accession number, **ANO**: | |  | |
| Identification number, **INO**: | |  | |
| Document type, **DT**: | | Monographic publication | |
| Type of record, **TR**: | | Textual printed material | |
| Contents code, **CC**: | | Graduate-master Thesis | |
| Author, **AU**: | | Jelena Hrnjak | |
| Mentor, **MN**: | | Vladimir Dimitrieski, PhD, Assistant Professor | |
| Title, **TI**: | | ... | |
| Language of text, **LT**: | | Serbian | |
| Language of abstract, **LA**: | | Serbian | |
| Country of publication, **CP**: | | Republic of Serbia | |
| Locality of publication, **LP**: | | Vojvodina | |
| Publication year, **PY**: | | 2023 | |
| Publisher, **PB**: | | Author’s reprint/Faculty of Technical Sciences | |
| Publication place, **PP**: | | Novi Sad, Faculty of Technical Sciences, Dositeja Obradovica sq. 6 | |
| Physical description, **PD**: (chapters/pages/ref./tables/pictures/graphs/appendixes) | | 6/45/0/50/36/0/0 | |
| Scientific field, **SF**: | | Electrical and computer engineering | |
| Scientific discipline, **SD**: | | Applied computer science and informatics | |
| Subject/Key words, **S**/**KW**: | | Domain-Specific Modeling Languages | |
| **UC** | |  | |
| Holding data, **HD**: | | The Library of Faculty of Technical Sciences, Novi Sad, Serbia | |
| Note, **N**: | |  | |
| Abstract, **AB**: | | TODO | |
| Accepted by the Scientific Board on, **ASB**: | |  | |
| Defended on, **DE**: | |  | |
| Defended Board, **DB**: | President: |  |
|  | Member: |  | Menthor's sign |
|  | Member, Mentor: | Vladimir Dimitrieski, PhD, Assistant Professor |  |

|  |  |  |
| --- | --- | --- |
|  | УНИВЕРЗИТЕТ У НОВОМ САДУ ⚫ **ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА**  21000 НОВИ САД, Трг Доситеја Обрадовића 6 | Датум: |
|  |
| **ЗАДАТАК ЗА ДИПЛОМСКИ РАД** | Лист/Листова: |
|  |

| Студијски програм: | **Рачунарство и аутоматика** |
| --- | --- |
| Руководилац стидијског програма: | др Милан Рапаић, редовни професор |

*(Податке уноси предметни наставник - ментор)*

|  |  |  |  |
| --- | --- | --- | --- |
| Студент: | **Јелена Хрњак** | Број индекса: | **Е2 64/2022** |
| Област: | **Примењене рачунарске науке и информатика** | | |
| Ментор: | **др Владимир Димитриески, доцент** | | |
| НА ОСНОВУ ПОДНЕТЕ ПРИЈАВЕ, ПРИЛОЖЕНЕ ДОКУМЕНТАЦИЈЕ И ОДРЕДБИ СТАТУТА ФАКУЛТЕТА ИЗДАЈЕ СЕ ЗАДАТАК ЗА МАСТЕР РАД, СА СЛЕДЕЋИМ ЕЛЕМЕНТИМА:   * проблем – тема рада; * начин решавања проблема и начин практичне провере резултата рада, ако је таква провера неопходна; * литература | | | |

**НАСЛОВ ДИПЛОМСКОГ РАДА:**

|  |
| --- |
| **Наменски језик за генерисање...** |

**ТЕКСТ ЗАДАТКА:**

|  |
| --- |
| * ... * ... |

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

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

**Садржај**

[1. Увод 1](#_Toc142147110)

[1.1 Структура рада 1](#_Toc142147111)

[2. Преглед постојећег стања у области 2](#_Toc142147112)

[3. Технологије коришћене при развоју наменског језика 3](#_Toc142147113)

[4. Апстрактна синтакса наменског језика 4](#_Toc142147114)

[4.1.1 Концепт *Application* 5](#_Toc142147115)

[4.1.2 Концепт Database 7](#_Toc142147116)

[4.1.3 Концепт *Entity* 8](#_Toc142147117)

[4.1.4 Концепт *Attribute* 8](#_Toc142147118)

[4.1.5 Концепт *User* 9](#_Toc142147119)

[4.1.6 Концепт *Role* 9](#_Toc142147120)

[4.1.7 Концепт *RoleInstance* 9](#_Toc142147121)

[4.1.8 Концепт *Controller* 9](#_Toc142147122)

[4.1.9 Концепт *Authentication* 10](#_Toc142147123)

[4.1.10 Концепт *Endpoint* 10](#_Toc142147124)

[4.1.11 Концепт *Security* 11](#_Toc142147125)

[4.1.12 Концепт *BasicAuthentication* 11](#_Toc142147126)

[4.1.13 Концепт *JWT* 11](#_Toc142147127)

[4.1.14 Концепт *Claim* 11](#_Toc142147128)

[4.1.15 Концепт *RegisteredClaim* 12](#_Toc142147129)

[4.1.16 Концепт *OAuth2* 12](#_Toc142147130)

[4.1.17 Концепт *Provider* 12](#_Toc142147131)

[5. Закључак 14](#_Toc142147132)

[Литература 16](#_Toc142147133)

[Биографија 17](#_Toc142147134)

# Увод

## Структура рада

Након уводног поглавља следи поглавње ,,Преглед постојећег стања у области”...

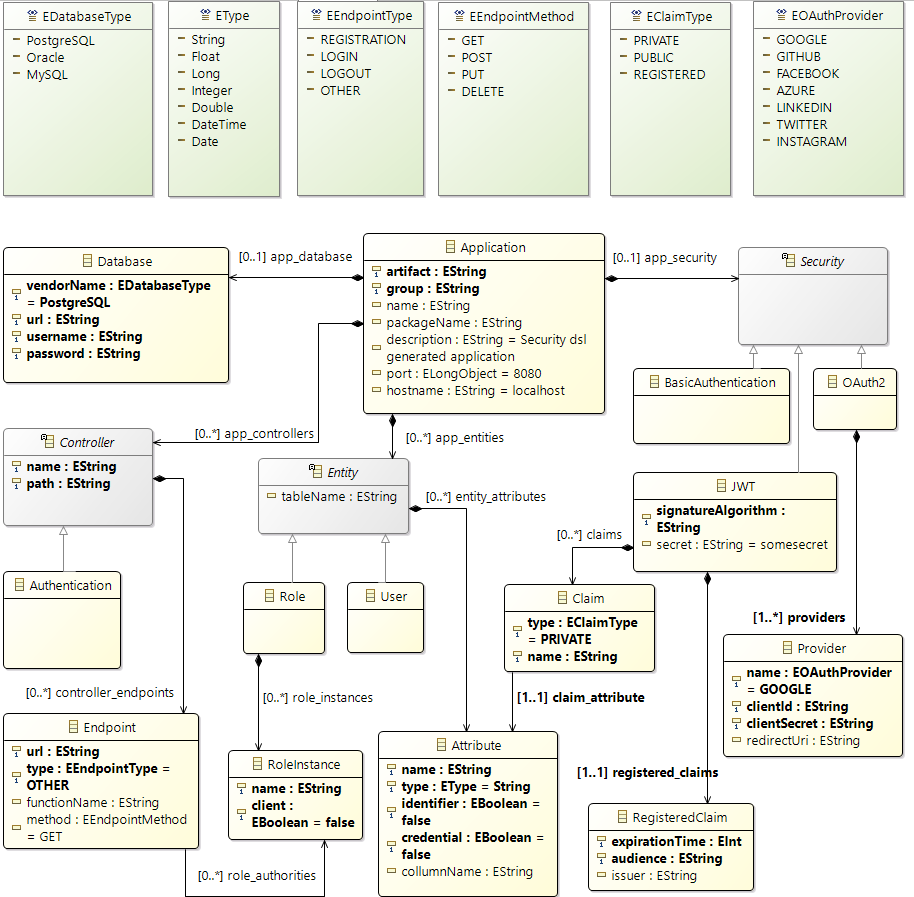
# Преглед постојећег стања у области

# Технологије коришћене при развоју наменског језика

# Апстрактна синтакса наменског језика

Апстрактна синтаска омогућава опис структуре наменског језика *securityDSL* и представљена је помоћу мета-модела (слика 4.1). Апликација која се генерише, а самим тим и апстрактна синтакса наменског језика *securityDSL*, може се поделити у пет целина:

* параметри који се односне на саму апликацију (као што су артефакт, опис итд.),
* параметри базе података,
* слој за репрезентацију података (ентитети),
* слој за обраду захтева корисника (контролери) и
* сигурносни слој који се односи на аутентификацију и контролу приступа корисника.



Слика 4.1 – Апстракнта синтакса наменског језика *securityDSL*

У даљем тексту дат је опис концепата апстрактне синтаксе, где су енглески називи концепата који су приказани на слици наведени курзивом.

Коренски концепт апстрактне синтаксе је апликација (*Application*) и садржи податке о параметрима неоходним за иницијализацију апликације. Кориснику је остављена могућност проширења генерисане апликације додатним концептима уз ограничења која ће бити наведена у наставку поглавља. Могуће је подесити параметре за повезивање са базом података *(Database)*, а поред тога, могу се дефинисати ентитети (*Entity*) који се односе на кориснике (*User*) и роле (*Role*). При дефинисању ентитета, неопходно је навести обележја (*Attribute*) за сваки ентитет. Концепт који се односи на обраду захтева корисника (*Controller*) је повезан са концептом *Endpoint* где је остављена могућност додавања метода које се односе на различите функционалности апликације. Навођењем конкретних рола (*RoleInstance*) које ће се налазити у систему и повезивањем са одређеним *endpoint-*овима омогућена је контрола приступа. Контролер за аутентификацију (*Authentication*) може да садржи методе за регистрацију, пријаву на систем и одјаву са система.

Посебан део апстрактне синтаксе односи се на сигурносни слој (*Security*), где су подржана три безбедносна механизма: основна аутентификација (*BasicAuthentication*), аутентификација заснована на *JWT* токенима (*JWT*) и *OAuth2.0* аутентификација *(OAuth2).* У зависности од жељеног механизма могу се дефинисати додатни параметри описани адекватним концептима.

Сваки од наведених концепата могу се додати или изоставити у зависности од потреба корисника. Оваква апстрактна синтакса омогућава кориснику флексибилност и једноставно проширење генерисане апликације у складу са потребама.

У наставку поглавља описани су сви концепти апстрактне синтаксе наменског језика *securityDSL* уз опис обележја и асоцијација концепата. Поред тога, наведена су и појашњена ограничења имплементирана помоћу декларативног језика *OCL.*

### Концепт *Application*

Коренски концепт апстрактне синтаксе *Application* садржи обележја која се односе на основне параметре неопходне за иницијализацију апликације.

|  |  |  |  |
| --- | --- | --- | --- |
| Назив обележја | Тип податка | Обавезно | Опис |
| *artifact* | *EString* | да | Назив артефакта |
| *group* | *EString* | да | Назив групе који представља организацију, компанију или тип који развија апликацију |
| *name* | *EString* | не | Назив апликације |
| *packageName* | *EString* | не | Назив пакета, *namespace* у ком се налази изворни код апликације |
| *description* | *EString* | не | Опис апликације |
| *port* | *ELongObject* | не | Порт на ком апликација слуша захтеве |
| *hostname* | *EString* | не | Адреса рачунара или сервера на ком се извршава апликација |

Табела 4.1 – Обележја концепта *Application*

Портови у опсегу од 1024 до 49151 представљају регистроване портове који се могу доделити апликацији. Ограничење *validRegisteredPort* (листинг 4.1) гарантује да апликација користи валидан и регистрован порт.

**invariant** validRegisteredPort:

**self**.*port* >= 1024 **and** **self**.*port* <= 49151;

Листинг 4.1 – Порт апликације мора бити унутар опсега регистрованих портова

|  |  |  |  |
| --- | --- | --- | --- |
| Назив асоцијације | Референцирани концепт | Кардиналитет | Опис |
| *app\_database* | *Database* | 0..1 | База података апликације |
| *app\_entitites* | *Entity* | 0..\* | Eнтитети у апликацији |
| *app\_controllers* | *Controller* | 0..\* | Контролери у апликацији |
| *app\_security* | *Security* | 0..1 | Безбедносна подешавања апликације |

Табела 4.2 – Асоцијације концепта *Application*

Апликација може садржати ентитете, контролере, базу података и аспекте безбедности. Кориснику је остављен простор да наведе или изостави одређене целине. Међутим, навођење неких целина повлачи одређена ограничења. Уколико се наведу ентитет, неопходно је да апликација има дефинисану базу података која складишти податке о тим ентитетима. Ово је омогућено *OCL* ограничењем *hasDatabawseForEntity* (листинг 4.2). С друге стране, уколико постоји контролер за аутентификацију, неопходно је да буду дефинисани ентитети који представљају кориснике и роле. Контролер за аутентификацију обрађује захтеве који се тичу регистрације корисника, пријаве и одјаве корисника са система, те су информације о корисницима и њиховим ролама неопходни (листинг 4.3).

**invariant** hasDatabaseForEntity:

**self**.*app\_entities* -> *isEmpty*() **or** **not** **self**.*app\_database*->*isEmpty*();

**invariant** hasUserAndRoleForController:

**self**.*app\_controllers* -> *exists*(c | *c*.*oclIsTypeOf*(*Authentication*)) **implies**

(

**self**.*app\_entities* -> *exists*(e | *e*.*oclIsTypeOf*(*User*))

**and**

**self**.*app\_entities* -> *exists*(e | *e*.*oclIsTypeOf*(*Role*))

);

Листинг 4.2 – Уколико апликација има дефинисане ентитете, мора да има дефинисану базу података

Листинг 4.3 – Уколико апликација има дефинисан контролер за аутентификацију, неопходно је да има дефинисане ентитете за кориснике и роле

### Концепт Database

Апстрактни концепт *Database* представља параметре за повезивање апликације са базом података. Тип података *EDatabaseType* представља енумерацију заназив компаније која је развила жељени систем за управљање базама података. Корисник је дужан да унесе валидне креденцијале и конекциони стринг како би апликација успешно успоставила везу са базом података. Ово омогућава да апликација чита, додаје и ажурира податке сачуване у бази података.

|  |  |  |  |
| --- | --- | --- | --- |
| Назив обележја | Тип податка | Обавезно | Опис |
| *vendorName* | *EDatabaseType* | да | Систем за управљање базама података, при чему су могуће вредности *PostgreSQL, MySQL* и *Oracle* |
| *url* | *EString* | да | Конекциони стринг за повезивање са базом података |
| *username* | *EString* | да | Корисничко име за приступ бази података |
| *password* | *EString* | да | Лозинка за приступ бази података |

Табела 4.3 – Обележја концепта *Database*

### Концепт *Entity*

Ентитети апликације моделовани су помоћу концепта *Entity.* Асоцијација ентитета са концептом *Attribute* омогућава да ентитети садрже сва релевантна обележја за домен апликације*.*

|  |  |  |  |
| --- | --- | --- | --- |
| Назив обележја | Тип податка | Обавезно | Опис |
| *tableName* | *EString* | не | Назив табеле у бази података која се односи на ентитет |

Табела 4.4 – Обележја концепта *Entity*

|  |  |  |  |
| --- | --- | --- | --- |
| Назив асоцијације | Референцирани концепт | Кардиналитет | Опис |
| *entity\_attributes* | *Attribute* | 0..\* | Обележја ентитета |

Табела 4.5 – Асоцијације концепта *Entity*

Ограничење *uniqueTableName* (листин 4.4) гарантује да сви ентиети имају јединствен назив табеле у бази података што спречава потенцијалне проблеме приликом рада са подацима. За ентитете за које није наведен, назив табеле биће изједначен са називом ентитета у множини (*users* или *roles*).

**invariant** uniqueTableName:

Entity.allInstances() -> *select*(e | *e*.*tableName* <> **null**) ->

*isUnique*(e | *e*.*tableName*.*toLower*());

Листинг 4.4 – Сви ентитети имају јединствен назив табеле у бази података

### Концепт *Attribute*

Концепт *Attribute* моделује обележја ентитета у апликацији која описују различите карактеристике ентитета. *EType* представља енумерацију за тип податка обележја моделованог концептом *Attribute.* Атрибут представља идентификатор ентитета уколико је вредност обележја *identifier* једнака *true,* док је атрибут креденцијал уколико је вредност обележја *credential* једнака *true.*

|  |  |  |  |
| --- | --- | --- | --- |
| Назив обележја | Тип податка | Обавезно | Опис |
| *name* | *EString* | да | Назив обележја |
| *type* | *EType* | да | Тип обележја, при чему су могуће вредности *String, Float, Long, Integer, Double, DateTime* и *Date* |
| *identifier* | *EBoolean* | да | Да ли је обележје идентификатор |
| *credential* | *EBoolean* | да | Да ли је обележје креденцијал |
| *collumnName* | *EString* | не | Назив колоне у табели базе података који се односи на обележје |

Табела 4.6 – Обележја концепта *Attribute*

Јединственост назива атрибута омогућена је ограничењем  *uniqueAttributeName* (листинг 4.5), док јединственост назива колона у табели, уколико је оно наведено, гарантује ограничењне *uniqueCollumnName* (листинг 4.6). Неопходно је ентитети да поседују тачно један идентификатор, што је обезбеђено ограничењем onlyOneIdentifier (листинг 4.7).

**invariant** uniqueAttributeName:

**self**.*entity\_attributes* -> *isUnique*(a | *a*.*name*);

Листинг 4.5 – Сви атрибути имају јединствене називе

**invariant** uniqueCollumnName:

**self**.*entity\_attributes* -> *exists*(a | *a*.*collumnName* <> **null**) **implies**

**self**.*entity\_attributes* -> *isUnique*(a | *a*.*collumnName*);

Листинг 4.6 – Сви атрибути имају јениствене називе колона у табели у бази података

**Invariant** onlyOneIdentifier:

**self**.*entity\_attributes* -> *size*() > 0 **implies**

**self**.*entity\_attributes* -> *select*(a | *a*.*identifier*) -> *size*() = 1;

Листинг 4.7 – За ентитет мора да постоји тачно један атрибут који је идентификатор

### Концепт *User*

Приликом генерисања кода, концепт *User* биће искориштен за генерисање класе које представља ентитет корисника апликације. Наслеђује концепт *Entity,* па садржи обележја која ближе описују корисника.

Атрибут који представља лозинку је подразумеван и биће генерисан са остатком кода, те не постоји потреба за експлицитним навођењем истог (листинг 4.8). Како би аутентификација била омогућена неопходно је да класа *User*, поред постојеће лозинке, поседује још тачно један атрибут који представља креденцијал (нпр. корисничко име). Ово је гарантовано ограничењем *oneStringTypeCredentialForUser* (листинг 4.9). Увођењем креденцијала, потребно је онемогућити креденцијале у класама које не представљају кориснике система (листинг 4.10).

**invariant** noAttributeNamedPassword:

**self**.*entity\_attributes* -> *forAll*(a | *a*.*name*.*toLower*() <> 'password');

Листинг 4.8 – За концепт *User* не сме да постоји атрибут који представља лозинку

**Invariant** oneCredentialForUser:

**self**.*entity\_attributes* -> *select*(a | *a*.*credential*) -> *size*() = 1;

Листинг 4.9 – За концепт *User* мора да постоји тачно један атрибут који представља креденцијал

**invariant** otherEntitiesDoesntHaveCredential:

**not** **self**.*oclIsTypeOf*(*User*) **implies**

**self**.*entity\_attributes* -> *select*(a | *a*.*credential*) -> *isEmpty*();

Листинг 4.10 – Само концепти *User* могу да поседују атрибут који представља креденцијал

### Концепт *Role*

Концепт *Role* наслеђује концепт *Entity* и моделује ентитет који се односи на роле апликације.

|  |  |  |  |
| --- | --- | --- | --- |
| Назив асоцијације | Референцирани концепт | Кардиналитет | Опис |
| *role\_instances* | *RoleInstance* | 0..\* | Конкретне роле у апликацији |

Табела 4.7 – Асоцијације концепта *Role*

За аутентификацију помоћу токена *JWT* роле представљају класу са обележјима. Како би конкретне роле биле омогућене, неопходно је да у случају одабира аутентификације на основу *JWT* токена за класу *Role* постоји тачно једно обележје које је типа *String.* Поред тога, дозвољено је само обележје које представља идентификатор. У случају да је идентификатор типа *String,* нису дозвољена додатна обележја (листинг 4.11). У случају одабира основне аутентификације (*BasicAuthentication*) роле ће у апликацији бити представљене као енумерација. Самим тим нису дозвољена обележја, већ само конкретне роле (листинг 4.12)

**Invariant** roleCanHaveIdAndStringAttribute:

**self**.*app\_security*.*oclIsTypeOf*(*JWT*) **implies**

(**self**.*app\_entities* -> *select*(e | *e*.*oclIsTypeOf*(*Role*))

-> *forAll*(role | *role*.*entity\_attributes* ->

*select*(a | *a*.*type* = *EType*::*\_'String'*

**and** *a*.*identifier* = **true**) -> *size*() = 1 )

**or**

**self**.*app\_entities* -> *select*(e | *e*.*oclIsTypeOf*(*Role*)) ->

*forAll*(role | *role*.*entity\_attributes* ->

*select*(a | *a*.*type* = *EType*::*\_'String'* **and**

*a*.*identifier* = **false**) -> *size*() = 1

**and**

*role*.*entity\_attributes* ->

*select*(a | *a*.*identifier* = **true**) -> *size*() = 1));

Листинг 4.11 – У случају аутентификације помоћу *JWT,* рола може да има тачно једно обележје типа *String*

**Invariant** basicAuthNoRoleAttributes ('Basic authentication cannot have role attributes!'):

**self**.*app\_security*.*oclIsTypeOf*(*BasicAuthentication*) **implies**

**self**.*app\_entities* -> *select*(e | *e*.*oclIsTypeOf*(*Role*)) ->

*forAll*(role | *role*.*entity\_attributes* -> *size*() = 0);

Листинг 4.12 – У случају основне аутентификације нису дозвољена придружена обележја конепту *Role*

### Концепт *RoleInstance*

Конкретне роле, било да је у питању енумерација или ентитет са обележјима, моделоване су помоћу концепта *RoleInstance*. Представљају роле које је могуће доделити корисницима система како би била омогућена контрола приступа ресурсима апликације.

|  |  |  |  |
| --- | --- | --- | --- |
| Назив обележја | Тип податка | Обавезно | Опис |
| *name* | *EString* | да | Назив конретне роле |
| *client* | *EBoolean* | да | Ознака да ли је рола клијент. Уколико је вредност обележја *false* рола је администраторска |

Табела 4.8 – Обележја концепта *RoleInstance*

### Концепт *Controller*

Концепт *Controller* описује контролере апликације. Омогућава навођење назива и путање контролера, при чему је неопходно да обе вредности буду јединствене за целу апликацију (листинг 4.13 и листинг 4.14). Уобичајено је да се називи контролера разликују од путања како би се избегли конфликти приликом рутирања захтева што је гарантовано ограничењем *uniqueControllerPath*. Путања контролера представља апсолутну путању у оквиру апликације, те је неопходно да почиње карактером '/'. Ово такође омогућава конзистентност генерисаног кода (листинг 4.15).

|  |  |  |  |
| --- | --- | --- | --- |
| Назив обележја | Тип податка | Обавезно | Опис |
| *name* | *EString* | да | Назив контролера |
| *path* | *EString* | да | Путања контролера |

Табела 4.9 – Обележја концепта *Controller*

|  |  |  |  |
| --- | --- | --- | --- |
| Назив асоцијације | Референцирани концепт | Кардиналитет | Опис |
| *controller\_endpoints* | *Endpoint* | 0..\* | *Endpoint-*ови контролера |

Табела 4.10 – Асоцијације концепта *Controller*

**invariant** uniqueControllerName ('Controllers should have unique names!'):

*Controller*.*allInstances*() -> *isUnique*(c | *c*.*name*);

Листинг 4.13 – Називи контролера унутар апликацјие морају да буду јединствени

**invariant** uniqueControllerPath ('Controller paths should be unique and different from names!'):

*Controller*.*allInstances*() -> *isUnique*(c | *c*.*path*) **and** *Controller*.*allInstances*() ->

*forAll*(c | '/' + *c*.*name*.*toLower*() <> *c*.*path*.*toLower*());

Листинг 4.14 - Путање контролера унутар апликације морају да буду јединствени и да се разлику од назива контролера

**invariant** controllerPath('Controller path should start with \'/\'!'):

**self**.*path*.*at*(1) = '/';

Листинг 4.15 – Путања контролера мора да почиње караткером ‘/’

### Концепт *Authentication*

### Концепт *Endpoint*

|  |  |  |  |
| --- | --- | --- | --- |
| Назив обележја | Тип податка | Обавезно | Опис |
| *url* | *EString* | да | Путања, односно *URL* адреса *endpoint-*а, |
| *type* | *EEndpointType* | да | Тип *endpoint*-a, где су могуће вредности регистрација (*REGISTRATION*), пријављивање (*LOGIN*), одјављивање (*LOGOUT*) и друго (*OTHER*) |
| *methodName* | *EString* | да | Назив методе на коју се односи *endpoint* |
| *method* | *EEndpointMethod* | не | *HTTP* метод који означава каква је врста захтева, при чему су могуће вредности *GET, POST, PUT* и *DELETE* |

Табела 4.11 – Обележја концепта *Endpoint*

|  |  |  |  |
| --- | --- | --- | --- |
| Назив асоцијације | Референцирани концепт | Кардиналитет | Опис |
| *role\_authorities* | *RoleInstance* | 0..\* | Роле које имају право приступа *endpoint*-у |

Табела 4.12 – Асоцијације концепта *Endpoint*

### Концепт *Security*

### Концепт *BasicAuthentication*

### Концепт *JWT*

|  |  |  |  |
| --- | --- | --- | --- |
| Назив обележја | Тип податка | Обавезно | Опис |
| *signatureAlgorithm* | *EString* | да | Алгоритам који се користи за потписивање *JWT* токена |
| *secret* | *EString* | не | Тајни кључ који се користи за потписивање *JWT* токена |

Табела 4.13 – Обележја концепта *JWT*

|  |  |  |  |
| --- | --- | --- | --- |
| Назив асоцијације | Референцирани концепт | Кардиналитет | Опис |
| *claims* | *Claim* | 0..\* | Тврдње у оквиру *JWT* токена |
| *registered\_claims* | *RegisteredClaim* | 1..1 | Регистроване тврдње *JWT* токена |

Табела 4.14 – Асоцијације концепта *JWT*

### Концепт *Claim*

|  |  |  |  |
| --- | --- | --- | --- |
| Назив обележја | Тип податка | Обавезно | Опис |
| *type* | *EClaimType* | да | Тип тврдње, при чему су могуће вредности *PRIVATE, PUBLIC* и *REGISTERED* |
| *name* | *EString* | да | Назив трвдње |

Табела 4.15 – Обележја концепта *Claim*

|  |  |  |  |
| --- | --- | --- | --- |
| Назив асоцијације | Референцирани концепт | Кардиналитет | Опис |
| *claim\_attributes* | *Attribute* | 1..1 | Атрибут на ког се тврдња односи и који садржи додатне информације о њој |

Табела 4.16 – Асоцијације концепта *Claim*

### Концепт *RegisteredClaim*

|  |  |  |  |
| --- | --- | --- | --- |
| Назив обележја | Тип податка | Обавезно | Опис |
| *expirationTime* | *EInt* | да | Време истека токена. Након истека, *JWT* токен више није валидан |
| *audience* | *EString* | да | Идентификатор примаоца ком је *JWT* токен намењен |
| *issuer* | *EString* | не | Идентификатор издравача *JWT* токена |

Табела 4.17 – Обележја концепта *RegisteredClaim*

### Концепт *OAuth2*

|  |  |  |  |
| --- | --- | --- | --- |
| Назив асоцијације | Референцирани концепт | Кардиналитет | Опис |
| *providers* | *Provider* | 1..\* | Конфигурисани провајдери у апликацији |

Табела 4.18 – Асоцијације концепта *OAuth2*

### Концепт *Provider*

|  |  |  |  |
| --- | --- | --- | --- |
| Назив обележја | Тип податка | Обавезно | Опис |
| *name* | *EOAuthProvider* | да | Назив провајдера, при чему су могуће вредности *Google, Github, Facebook, Microsoft azure, Linkedin, Twitter* и *Instagram* |
| *clientId* | *EString* | да | Идентификациони број клијента додељен од стране провајдера |
| *clientSecret* | *EString* | да | Тајни кључ клијента додељен од стране провајдера за сигурносну контролу приликом комуникације са провајдером |
| *redirectUri* | *EString* | не | Адреса за преусмерење након успешне аутентификације |

Табела 4.19 – Обележја концепта *Provider*

# Закључак

скраћенице

* URI (Uniform Resource Identifier)
* URL
* JWT
* HTTP
* *OCL*

# Литература

*[1] Java Documentation, <https://docs.oracle.com/en/java/>*

*[2] Spring Boot Documentation,* [*https://spring.io/projects/spring-boot/*](https://spring.io/projects/spring-boot/)

*[3] PostgreSQL,* [*https://www.postgresql.org/about/*](https://www.postgresql.org/about/)

# Биографија

Јелена Хрњак рођена је 21. августа 1999. године у Бачкој Тополи где је стекла основно образовање у основној школи ,,Никола Тесла”. Даље школовање је наставила у Суботици где је завршила Гимназију ,,Светозар Марковић”, природно-математички смер. Школске 2018/2019 године уписује се на Факултет техничких наука Универзитета у Новом Саду, смер Рачунарство и аутоматика. Основне академске студије завршила је 2022. године и исте године уписује се на мастер академске студије на студијском програму Рачунарство и аутиматика Факултета теничких наука. Положила је све испите предвиђене планом и програмом мастер академских студија.