

Универзитет у Београду
Факултет организационих наука
Катедра за софтверско инжењерство

Семинарски рад из предмета Пројектовање софтвера
Тема: Софтверски систем за евиденцију запослених на
факултету

Ментор
Др. Душан Савић

Студент
Алекса Крсмановић 2018/30

Београд, 2023.

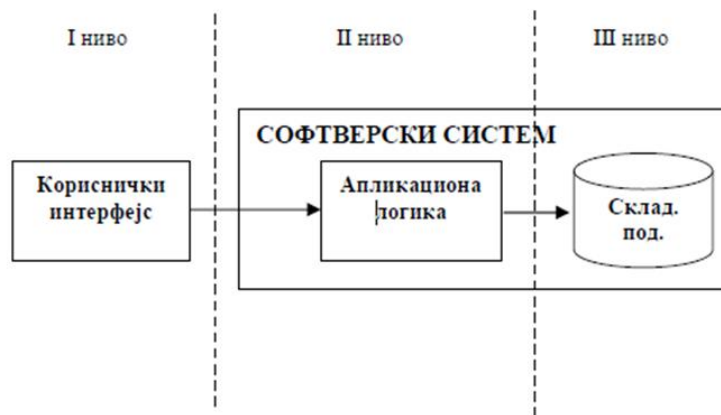
Пројектовање	2
Архитектура софтверског система	2
Пројектовање корисничког интерфејса	2
СК1: Случај коришћења – Унос нове катедре	5
СК2: Случај коришћења – Унос новог запосленог	7
СК3: Случај коришћења – Измена запосленог	9
СК4: Случај коришћења – Деактивација запосленог	12
СК5: Случај коришћења – Приказ свих запослених	15
СК6: Случај коришћења – Приказ свих катедри	17
СК7: Случај коришћења – Направи извештај за катедру	19
СК8: Случај коришћења – Обриши ставку историјата академског звања	20
Пројектовање контролера корисничког интерфејса	22
Пројектовање апликационе логике	22
Пројектовање контролера апликационе логике	23
Пројектовање понашања софтверског система (системске операције)	24
Уговор УГ1: Креирај ентитет	24
Уговор УГ2: Обриши ентитет	25
Уговор УГ3: Ажурирај ентитет	26
Уговор УГ4: Врати све ентитете	27
Уговор УГ5: Врати филтриране ентитете	28
Уговор УГ6: Запамти листу ентитета	29
Пројектовање структуре софтверског система (доменске класе)	30
Пројектовање складишта података	33

Пројектовање

Архитектура софтверског система

Фаза пројектовања описује физичку структуру и понашање софтверског система (архитектуру софтверског система).

Архитектура софтверског система је тронивојска и састоји се од корисничког интерфејса, апликационе логике и складишта података.



Слика 1: Тронивојска архитектура

Ниво корисничког интерфејса је на страни клијента, а апликациона логика и складиште података на страни сервера.

Пројектовање корисничког интерфејса

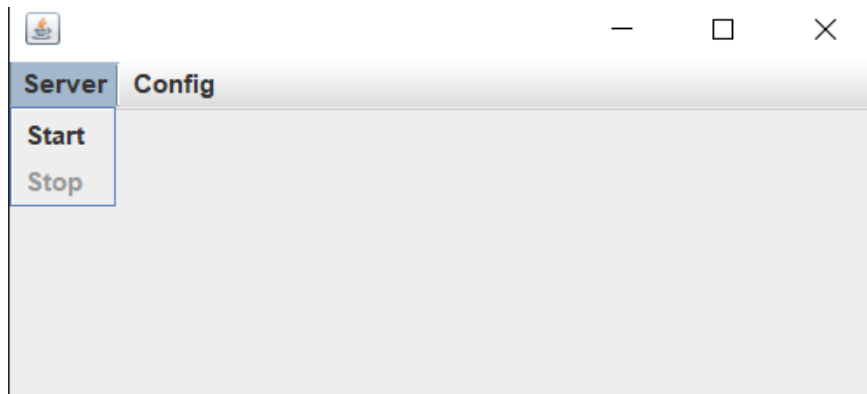
Кориснички интерфејс представља реализацију улаза и/или излаза софтверског система.



Слика 2: Структура корисничког интерфејса

Корисник комуницира са екранском формом где уноси, мења и брише податке користећи графичке компоненте. Такође, на тај начин креира и догађаје. Екранска форма прихвата податке које је корисник унео, као и догађаје које корисник прави, позива контролера корисничког интерфејса и прослеђује му ове податке. Екранска форма приказује податке које прима од контролера.

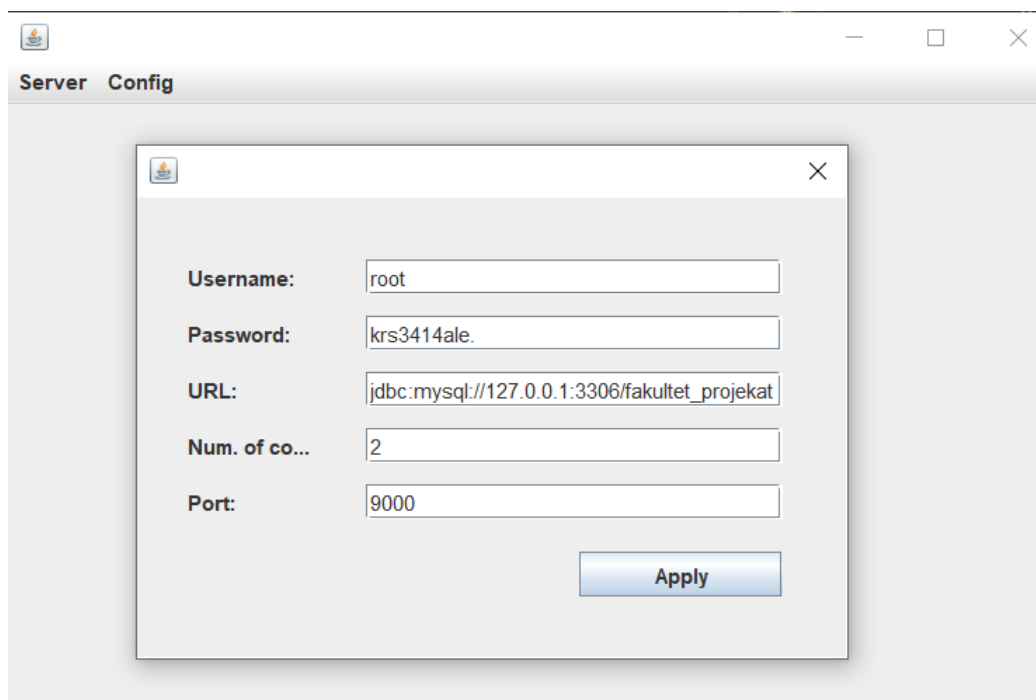
На серверској страни програма се налази форма са два менија. Мени Server има ставке за покретање и заустављање сервера.



Слика 3: Серверски програм 1

Кликом на „Start“, сервер ће се покренути и биће спреман за рад са клијентима. Уколико се кликне на „Stop“, сервер ће прекинути везе са свим клијентима.

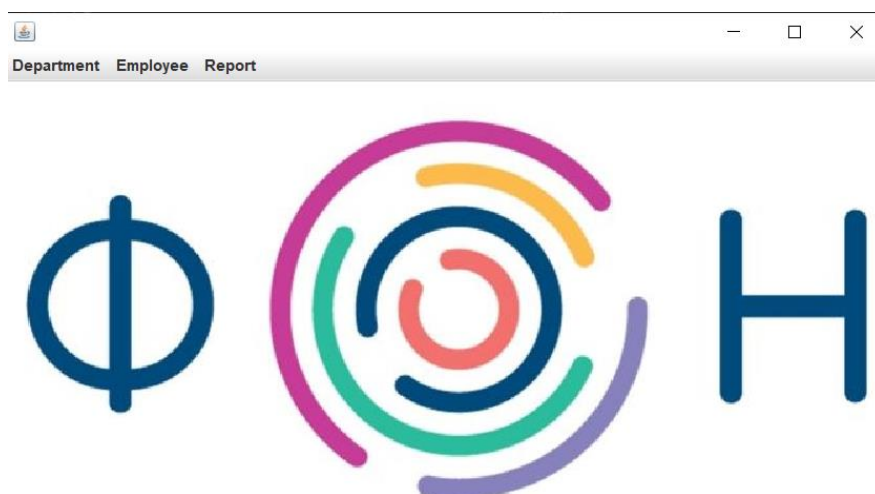
Мени Config служи за подешавање параметара за успостављање комуникације са базом података. Кликом на дугме “View” отвара се следећи дијалог и у њему су уписане тренутне вредности параметара:



Слика 4: Серверски програм 2

Кликом на дугме “Apply” нове вредности параметара се уписују у конфигурациони фајл.

Изглед главне клијентске форме:



Слика 5: Клијентски програм

СК1: Случај коришћења –Унос нове катедре

Назив СК

Креирање нове катедре

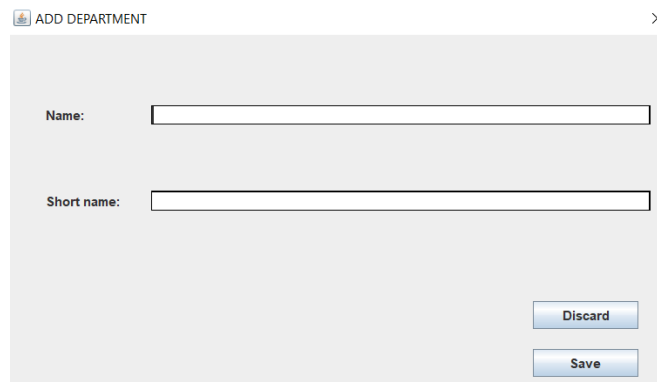
Актори СК

Корисник

Учесници СК

Корисник и систем (програм)

Предуслов: Систем је укључен и корисник је улогован. Систем приказује форму за рад са катедром.



ADD DEPARTMENT

Name:

Short name:

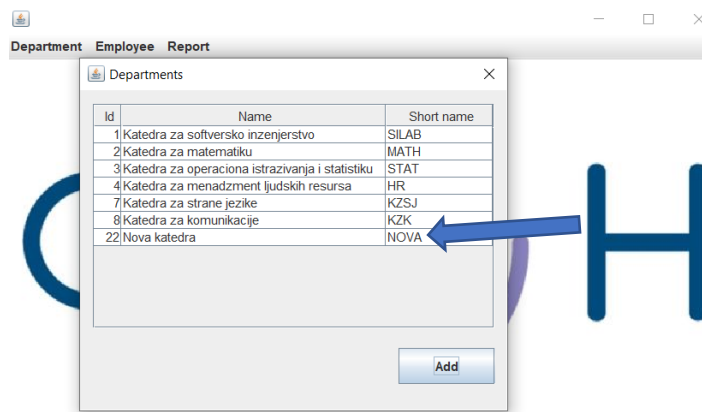
Discard

Save

Слика 6: Форма за унос нове катедре

Основни сценарио СК

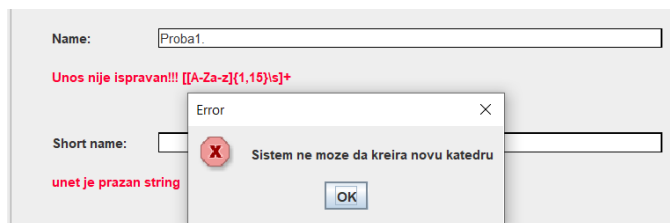
1. **Корисник** уноси податке о новој катедри. (АПУСО)
2. **Корисник** проверава податке о новој катедри. (АНСО)
3. **Корисник** позива **систем** да запамти податке о катедри. (АПСО)
4. **Систем** памти податке о катедри. (СО)
5. **Систем** приказује **кориснику** запамћену катедру и затвара форму за унос.(ИА)



Слика 7: Успешно креирање нове катедре

Алтернативна сценарија

5.1 Уколико **систем** не може да креира **нову катедру** он приказује **кориснику** поруку: “**Систем** не може да креира **нову катедру**”. Прекида се извршење сценарија. (ИА)



Слика 8: Неуспешно креирање нове катедре

СК2: Случај коришћења –Унос новог запосленог

Назив СК

Креирање новог запосленог

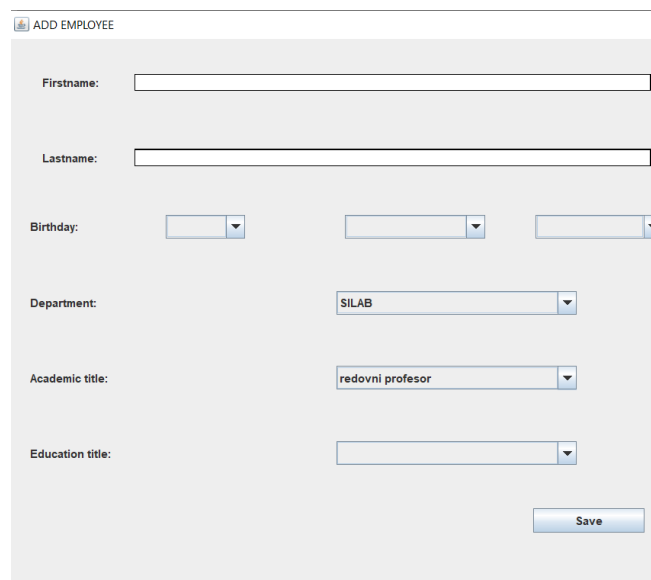
Актори СК

Корисник

Учесници СК

Корисник и систем (програм)

Предуслов: Систем је укључен и корисник је улогован. Систем приказује форму за рад са запосленим.



ADD EMPLOYEE

Firstname:

Lastname:

Birthday:

Department:

Academic title:

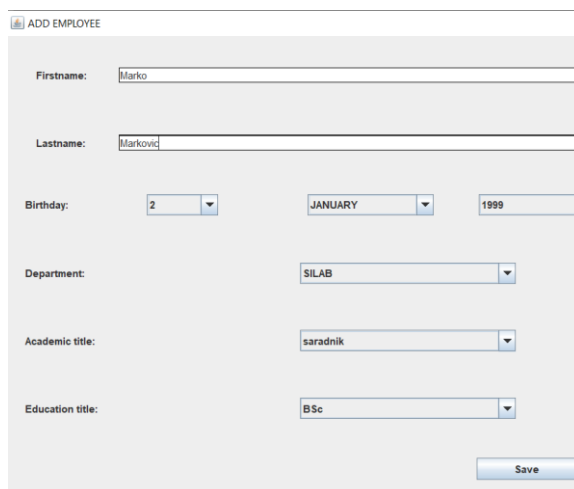
Education title:

Save

Слика 9: Форма за унос новог запосленог

Основни сценарио СК

1. **Корисник** уноси податке о **новом запосленом**. (АПУСО)



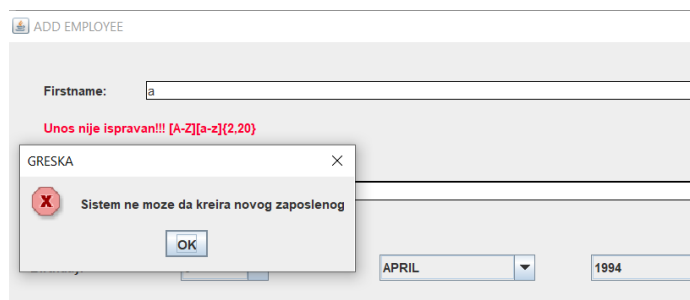
The screenshot shows a web form titled "ADD EMPLOYEE". It contains several input fields and dropdown menus. The "Firstname" field is filled with "Marko". The "Lastname" field is filled with "Markovic". The "Birthday" field has three parts: a day dropdown set to "2", a month dropdown set to "JANUARY", and a year dropdown set to "1999". The "Department" dropdown is set to "SILAB". The "Academic title" dropdown is set to "saradnik". The "Education title" dropdown is set to "BSc". A "Save" button is located at the bottom right of the form.

Слика 10: Попуњена форма за унос новог запосленог

2. **Корисник** **проверава** податке о **новом запосленом**. (АНСО)
3. **Корисник** **позива систем** да запамти податке о запосленом. (АПСО)
4. **Систем** **памти** податке о **запосленом**. (СО)
5. **Систем** **затвара** форму за унос.(ИА)

Алтернативна сценарија

5.1 Уколико **систем** не може да креира **новог запосленог** он приказује **кориснику** поруку: “**Систем** не може да креира **новог запосленог**”. Прекида се извршење сценарија. (ИА)



The screenshot shows the "ADD EMPLOYEE" form with the "Firstname" field containing the letter "a". A red error message is displayed above the field: "Unos nije ispravan!!! [A-Z][a-z]{2,20}". A modal dialog box titled "GRESKA" (Error) is open in the foreground. It contains a red 'X' icon and the text "Sistem ne moze da kreira novog zaposlenog" (System cannot create new employee). There is an "OK" button at the bottom of the dialog. In the background, the "Birthday" dropdowns are visible, set to "APRIL" and "1994".

Слика 11: Неуспешно креирање новог запосленог

СКЗ: Случај коришћења – Измена запосленог

Назив СК

Промена запосленог

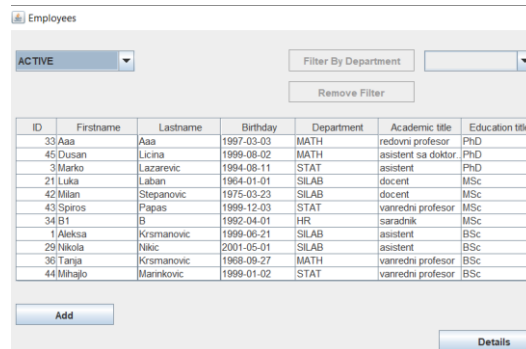
Актори СК

Корисник

Учесници СК

Корисник и систем (програм)

Предуслов: Систем је укључен и корисник је улогован. Систем приказује форму за приказ запослених. Учитана је листа запослених.



ID	Firstname	Lastname	Birthday	Department	Academic title	Education title
33/Aaa	Aaa	Aaa	1997-03-03	MATH	redovni profesor	PhD
45/Dusan	Licina		1999-08-02	MATH	asistent sa doktor	PhD
3/Marko	Lazarevic		1994-08-11	STAT	asistent	PhD
21/Luka	Laban		1994-01-01	SILAB	docent	MSc
42/Milan	Stepanovic		1975-03-23	SILAB	docent	MSc
43/Spiros	Papas		1999-12-03	STAT	vanredni profesor	MSc
34/B1	B		1992-04-01	HR	saradnik	MSc
1/Aleksa	Krsmanovic		1999-06-21	SILAB	asistent	BSc
29/Nikola	Nikic		2001-05-01	SILAB	asistent	BSc
36/Tanja	Krsmanovic		1968-09-27	MATH	vanredni profesor	BSc
44/Mhajo	Marinkovic		1999-01-02	STAT	vanredni profesor	BSc

Слика 12: Форма за приказ запослених

Основни сценарио СК

1. **Корисник** бира запосленог којег жели да измени. (АПУСО)
2. **Корисник** позива **систем** да учита запосленог. (АПСО)
3. **Систем** учитава изабраног запосленог. (СО)
4. **Систем** приказује **кориснику** изабраног запосленог.(ИА)

VIEW EMPLOYEE

Firstname:

Lastname:

Birthday:

Department:

Academic title:

Education title:

Academic title	Begin date	End date

ADD SAVE REMOVE Edit

Слика 13: Форма за приказ података о запосленом

5. **Корисник** позива **систем** да омогући измену запосленог. (АПСО)
6. **Систем** омогућава измену запосленог. (СО)
7. **Систем** приказује форму са запосленим у режиму за измену.(ИА)
8. **Корисник** уноси (мења) податке о запосленом. (АПУСО)

EDIT EMPLOYEE

Firstname:

Lastname:

Birthday:

Department:

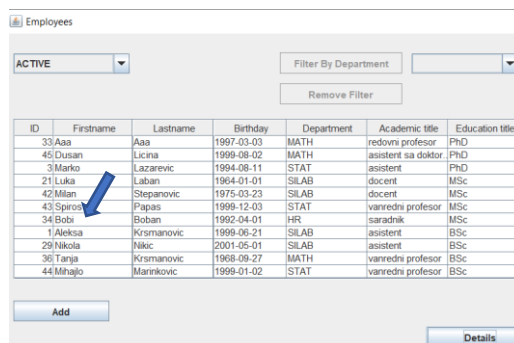
Academic title:

Education title:

Save

Слика 14: Форма за измену података о запосленом

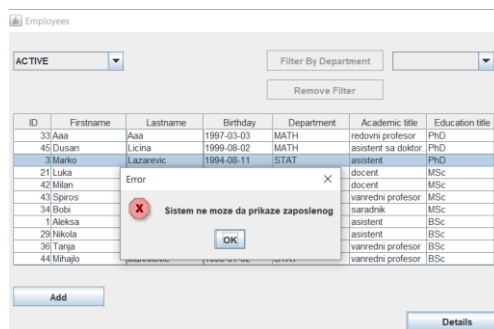
9. **Корисник** контролише да ли је коректно унео податке о запосленом. (АНСО)
10. **Корисник** позива **систем** да запамти податке о запосленом. (АПСО)
11. **Систем** памти податке о запосленом. (СО)
12. **Систем** приказује **кориснику** запамћеног запосленог.(ИА)



Слика 15: Успешна измена запосленог

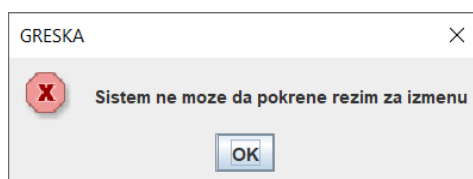
Алтернативна сценарија

4.1 Уколико **систем** не може да прикаже **запосленог**, он приказује **кориснику** поруку: “**Систем** не може да прикаже **запосленог**”. Прекида се извршење сценарија. (ИА)



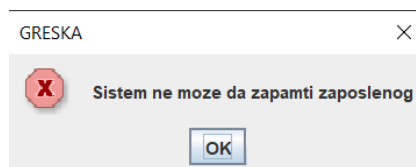
Слика 16: Неуспешан приказ запосленог

7.1 Уколико **систем** не може да прикаже форму са **запосленим у режиму за измену** он приказује **кориснику** поруку “**Систем** не може да покрене режим за измену”. Прекида се извршење сценарија. (ИА)



Слика 17: Неуспешна промена режима форме

12.1 Уколико **систем** не може да запамти **запосленог**, он приказује **кориснику** поруку: “**Систем** не може да запамти **запосленог**”. Прекида се извршење сценарија. (ИА)



Слика 18: Неуспешна измена запосленог

СК4: Случај коришћења – Деактивација запосленог

Назив СК

Деактивација запосленог

Актори СК

Корисник

Учесници СК

Корисник и систем (програм)

Предуслов: Систем је укључен и корисник је улогован. Систем приказује форму за приказ катедре. Учитана је листа запослених.

ID	Firstname	Lastname	Birthday	Department	Academic title	Education title
21	Luka	Laban	1964-01-01	SILAB	docent	MSc
42	Milan	Stepanovic	1975-03-23	SILAB	docent	MSc
1	Aleksa	Krsmanovic	1999-06-21	SILAB	asistent	BSc
29	Nikola	Nikic	2001-05-01	SILAB	asistent	BSc
20	Bobi	Krsmanovic	2012-03-25	SILAB	redovni profesor	
46	Lili	Lili	2000-01-02	SILAB	redovni profesor	
48	Prekdzija	Dorna	1931-01-02	SILAB	redovni profesor	
49	Petar	Miosevic	2002-12-03	SILAB	asistent sa doktor.	

Слика 19: Форма за приказ катедре

Основни сценарио СК

1. **Корисник** бира запосленог којег жели да деактивира. (АПУСО)

View Department

ACTIVE

ID: 1

Name: Katedra za softversko inženjerstvo

ID	Firstname	Lastname	Birthday	Department	Academic title	Education title
21	Luka	Laban	1964-01-01	SILAB	docent	MSc
42	Milan	Stepanovic	1975-03-23	SILAB	docent	MSc
1	Aleksa	Krsmanovic	1999-06-21	SILAB	asistent	BSc
29	Nikola	Nikic	2001-05-01	SILAB	asistent	BSc
20	Bob	Krsmanovic	2012-03-25	SILAB	redovni profesor	
46	Lili	Lili	2000-01-02	SILAB	redovni profesor	
48	Predzija	Dorna	1931-01-02	SILAB	redovni profesor	
49	Petar	Milosevic	2002-12-03	SILAB	asistent sa doktor.	

Слика 20: Форма за приказ катедре 2

2. **Корисник** позива **систем** да деактивира запосленог. (АПСО)
3. **Систем** захтева потврду одлуке од корисника. (СО)
4. **Систем** приказује **кориснику** форму за потврду жеље **корисника**. (ИА)

View Department

ACTIVE

ID: 1

Name: Katedra za softversko inženjerstvo

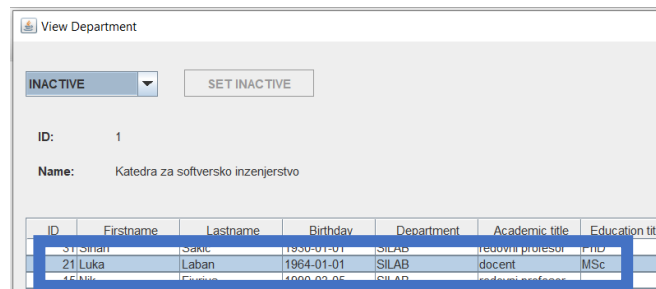
POTVRDITE

Da li zelite da deaktivirate zaposlenog?

ID	Firstname	Lastname	Birthday	Department	Academic title	Education title
21	Luka	Laban	1964-01-01	SILAB	docent	MSc
42	Milan	Stepanovic	1975-03-23	SILAB	docent	MSc
1	Aleksa	Krsmanovic	1999-06-21	SILAB	asistent	BSc
29	Nikola	Nikic	2001-05-01	SILAB	asistent	BSc
20	Bob	Krsmanovic	2012-03-25	SILAB	redovni profesor	
46	Lili	Lili	2000-01-02	SILAB	redovni profesor	
48	Predzija	Dorna	1931-01-02	SILAB	redovni profesor	
49	Petar	Milosevic	2002-12-03	SILAB	asistent sa doktor.	

Слика 21: Форма за потврду жеље

5. **Корисник** потврђује **систему** да деактивира запосленог. (АПСО)
6. **Систем** деактивира изабраног запосленог. (СО)
7. **Систем** приказује **кориснику** деактивираниог **запосленог**. (ИА)

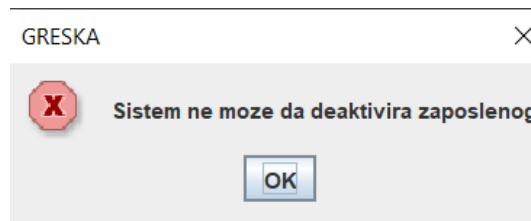


Слика 22: Приказ деактивираних запослених

Алтернативна сценарија

4.1 Уколико **систем** не може да прикаже форму за потврду жеље **корисника**, он **приказује** поруку: “**Систем** не може да прикаже форму за потврду жеље **корисника**”. Прекида се извршење сценарија. (ИА)

7.1 Уколико **систем** не може да деактивира **запосленог**, он приказује **кориснику** поруку: “**Систем** не може да деактивира **запосленог**”. Прекида се извршење сценарија. (ИА)



Слика 23: Неуспешна деактивација запосленог

СК5: Случај коришћења – Приказ свих запослених

Назив СК

Приказ свих запослених

Актори СК

Корисник

Учесници СК

Корисник и систем (програм)

Предуслов: Систем је укључен и корисник је улогован. Отворена је главна клијентска форма.



Слика 24: Главна форма

Основни сценарио СК

1. **Корисник** позива **систем** да прикаже све запослене. (АПСО)
2. **Систем** учитава листу запослених. (СО)
3. **Систем** приказује **кориснику** све запослене.(ИА)

The screenshot shows a web application window titled 'Employees'. It features a filter section with a dropdown menu set to 'ACTIVE', a 'Filter By Department' button, and a 'Remove Filter' button. Below this is a table with the following data:

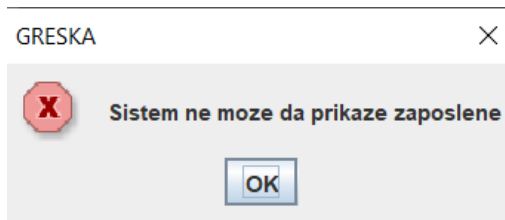
ID	Firstname	Lastname	Birthday	Department	Academic title	Education title
3	Marko	Lazarevic	1994-08-11	MATH	asistent	PhD
33	Aaa	Aaa	1997-03-03	MATH	redovni profesor	PhD
45	Dusan	Licina	1999-08-02	MATH	asistent sa doktor.	PhD
21	Luka	Laban	1964-01-01	SILAB	docent	MSc
42	Milan	Stepanovic	1975-03-23	SILAB	docent	MSc
43	Spiros	Papas	1999-12-03	STAT	vanredni profesor	MSc
34	Bobo	Boban	1992-04-01	HR	saradnik	MSc
1	Aleksa	Krsmanovic	1999-06-21	SILAB	asistent	BSc
29	Nikola	Nikic	2001-05-01	SILAB	asistent	BSc
36	Tanja	Krsmanovic	1968-09-27	MATH	vanredni profesor	BSc
44	Mihajlo	Marinkovic	1999-01-02	STAT	vanredni profesor	BSc

At the bottom of the form, there are two buttons: 'Add' and 'Details'.

Слика 25: Форма за приказ свих запослених

Алтернативна сценарија

3.1 Уколико **систем** не може да прикаже **све запослене**, он приказује **кориснику** поруку: “**Систем** не може да прикаже **запослене**”. Прекида се извршење сценарија. (ИА)



Слика 26: Неуспешан приказ запослених

СК6: Случај коришћења – Приказ свих катедри

Назив СК

Приказ свих катедри

Актори СК

Корисник

Учесници СК

Корисник и систем (програм)

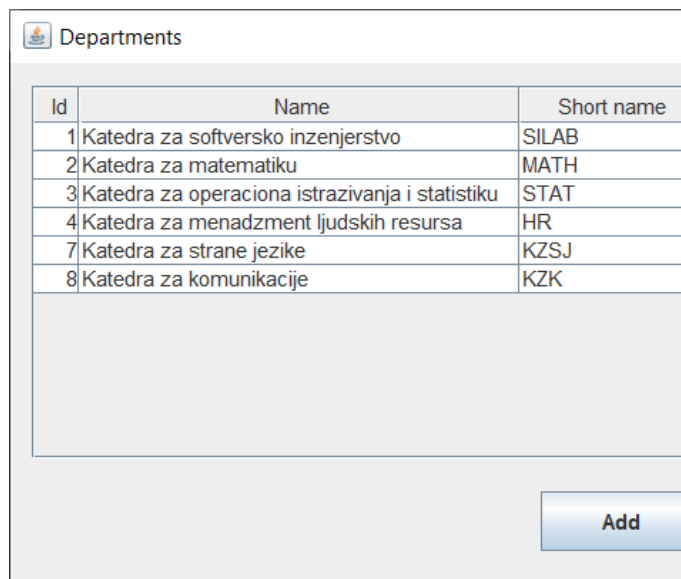
Предуслов: Систем је укључен и корисник је улогован. Отворена је главна клијентска форма.



Слика 27: Главна форма

Основни сценарио СК

1. **Корисник** **позива** **систем** да прикаже све катедре. (АПСО)
2. **Систем** **учитава** листу катедри. (СО)
3. **Систем** **приказује** **кориснику** све катедре.(ИА)



The screenshot shows a window titled 'Departments' with a table containing the following data:

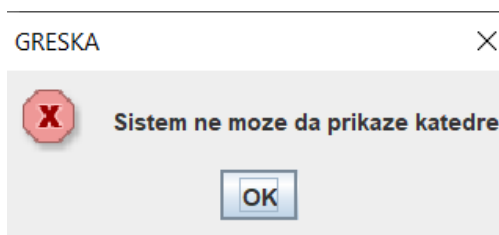
Id	Name	Short name
1	Katedra za softversko inzenjerstvo	SILAB
2	Katedra za matematiku	MATH
3	Katedra za operaciona istrazivanja i statistiku	STAT
4	Katedra za menadzment ljudskih resursa	HR
7	Katedra za strane jezike	KZSJ
8	Katedra za komunikacije	KZK

Below the table is a large empty rectangular area and an 'Add' button at the bottom right.

Слика 28: Форма за приказ свих катедри

Алтернативна сценарија

3.1 Уколико **систем** не може да прикаже **све катедре**, он приказује **кориснику** поруку: “**Систем** не може да прикаже **запослене**”. Прекида се извршење сценарија. (ИА)



Слика 29: Неуспешан приказ катедри

СК7: Случај коришћења – Направи извештај за катедру

Назив СК

Направи извештај за катедру

Актори СК

Корисник

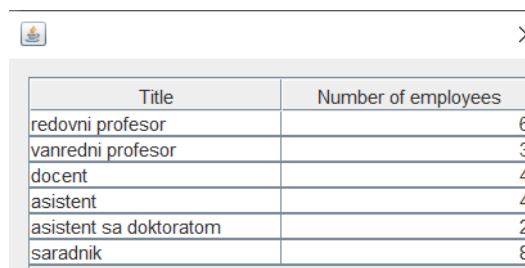
Учесници СК

Корисник и систем (програм)

Предуслов: Систем је укључен и корисник је улогован. Отворена је главна клијентска форма. Учитане су листе академских звања и запослених.

Основни сценарио СК

1. Корисник позива систем да направи извештај . (АПСО)
2. Систем креира извештај о академским звањима запослених. (СО)
3. Систем приказује кориснику извештај.(ИА)

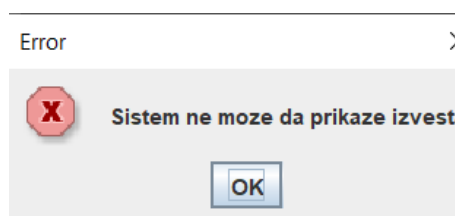


Title	Number of employees
redovni profesor	6
vanredni profesor	3
docent	4
asistent	4
asistent sa doktoratom	2
saradnik	6

Слика 30: Форма за приказ извештаја

Алтернативна сценарија

3.1 Уколико систем не може да прикаже извештај он приказује кориснику поруку: “Систем не може да прикаже извештај”. Прекида се извршење сценарија. (ИА)



Слика 31: Неуспешан приказ извештаја

СК8: Случај коришћења – Обриши ставку историјата академског звања

Назив СК

Обриши ставку историјата академског звања

Актери СК

Корисник

Учесници СК

Корисник и систем (програм)

Предуслов: Систем је укључен и корисник је улогован. Отворена је форма рад са запосленим у режиму “Детаљи”.

The screenshot shows a web form titled "VIEW EMPLOYEE". It contains several input fields and dropdown menus for employee information:

- Firstname: Milan
- Lastname: Stepanovic
- Birthday: 23, MARCH, 1975
- Department: SILAB
- Academic title: docent
- Education title: MSc

Below these fields is a table showing academic titles and their durations:

Academic title	Begin date	End date
vanredni profesor	2000-01-04	2005-02-02
redovni profesor	2005-02-02	

At the bottom of the form are four buttons: ADD, SAVE, REMOVE, and Edit.

Слика 32: Форма за приказ запосленог

Основни сценарио СК

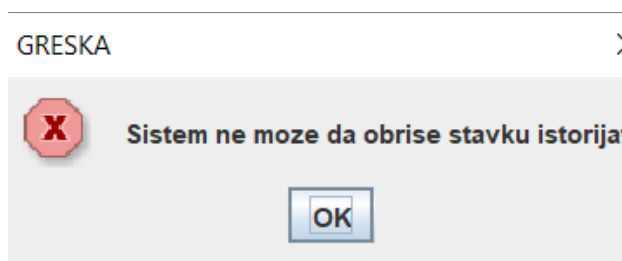
1. **Корисник** бира ставку историјата за брисање. (АПУСО)
2. **Корисник** позива **систем** да обрише ставку историјата . (АПСО)
3. **Систем** брише ставку историјата. (СО)
4. **Систем** приказује **кориснику** историјат без обрисане ставке.(ИА)

Academic title	Begin date	End date
vanredni profesor	2000-01-04	2005-02-02

Слика 33: Успешно брисање ставке историјата

Алтернативна сценарија

4.1 Уколико **систем** не може да обрише ставку историјата он приказује **кориснику** поруку: “**Систем** не може да обрише ставку историјата”. Прекида се извршење сценарија. (ИА)



Слика 34: Неуспешно брисање ставке историјата

Пројектовање контролера корисничког интерфејса

Контролер корисничког интерфејса је одговоран да:

1. прихвати податке које шаље екранска форма
2. конвертује податке (који се налазе у графичким елементима) у објекат који представља улазни аргумент СО која ће бити позвана
3. шаље захтев за извршење системске операције до апликационог сервера (софтверског система)
4. прихвата објекат (излаз) софтверског система настао као резултат извршења системске операције
5. Прослеђује објекат форми корисничког интерфејса

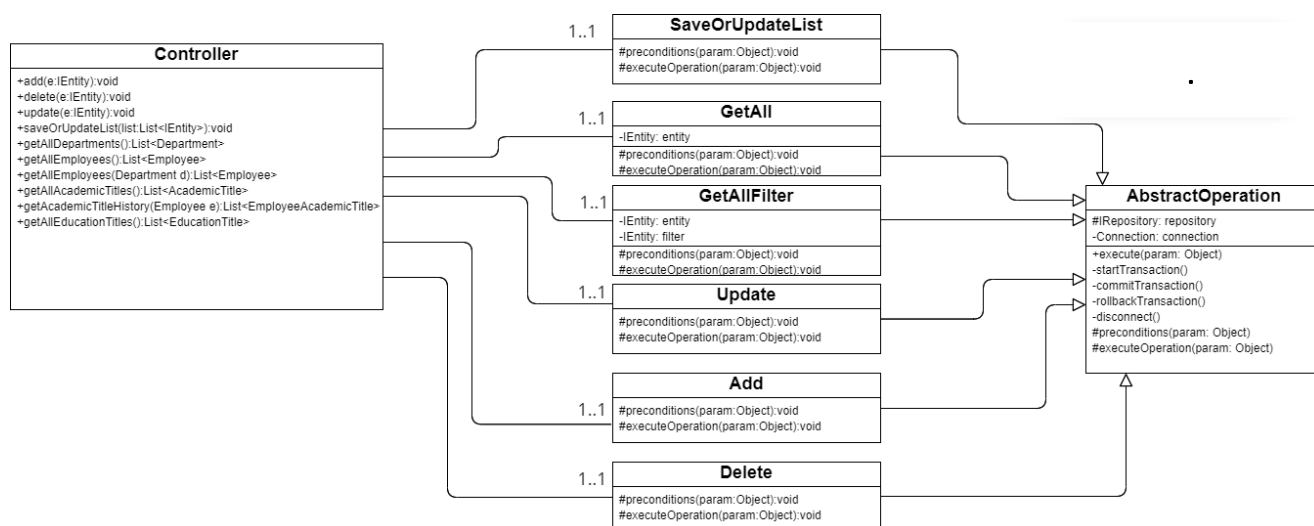
Пројектовање апликационе логике

Апликациона логика описује структуру и понашање система. Апликациони сервер се састоји из:

1. Контролера апликационе логике – Служи за комуникацију са клијентом и одговоран је да прихвати захтев за извршење системске операције од клијента и проследи га до пословне логике која је одговорна за извршење СО
2. Пословна логика – описана је структуром (доменске класе) и понашањем (системске операције)
3. Брокера базе података – служи за комуникацију између пословне логике и базе података.

Пројектовање контролера апликационе логике

Софтверски систем реализован је као клијент-сервер апликација. На серверској страни је нит `ServerThread` која садржи објекат класе `ServerSocket`. Нит константно позива методу асерт која чека да се покрене клијентска апликација која ће, кад се то деси, покушати да се повеже на сервер. Слање и примање података од клијента се остварује преко сокета, метода асерт креира објекат класе `Socket`. Клијент шаље захтев за извршење неке од СО до одговарајуће нити (коју смо назвали `ClientThread`), која је повезана са тим клијентом. `ClientThread` прима захтев и даље га преусмерава до класа које су одговорне за извршење СО. Након извршења СО резултат се враћа до апликационе логике, односно до класе `ClientThread` на серверској страни која тај резултат шаље назад до клијента путем сокета.



Слика 35: Контролер апликационе логике

Пројектовање понашања софтверског система (системске операције)

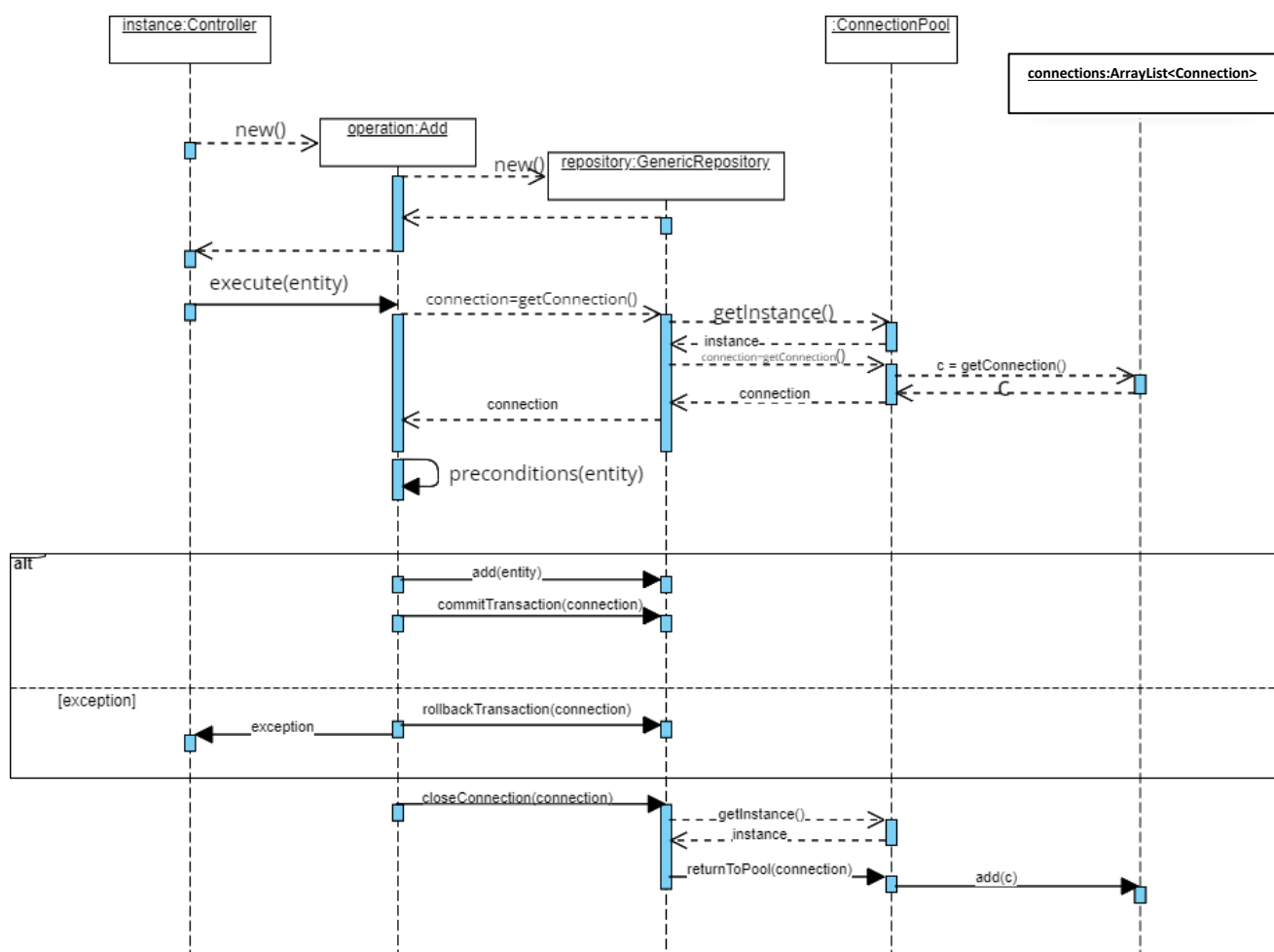
Уговор УГ1: Креирај ентитет

Операција: add IEntity),

Веза са СК: СК1

Предуслови: Структурна и вредносна ограничења над ентитетом морају бити задовољена.

Постуслови: Креиран је нови ентитет.



Слика 36: СД за операцију add(IEntity)

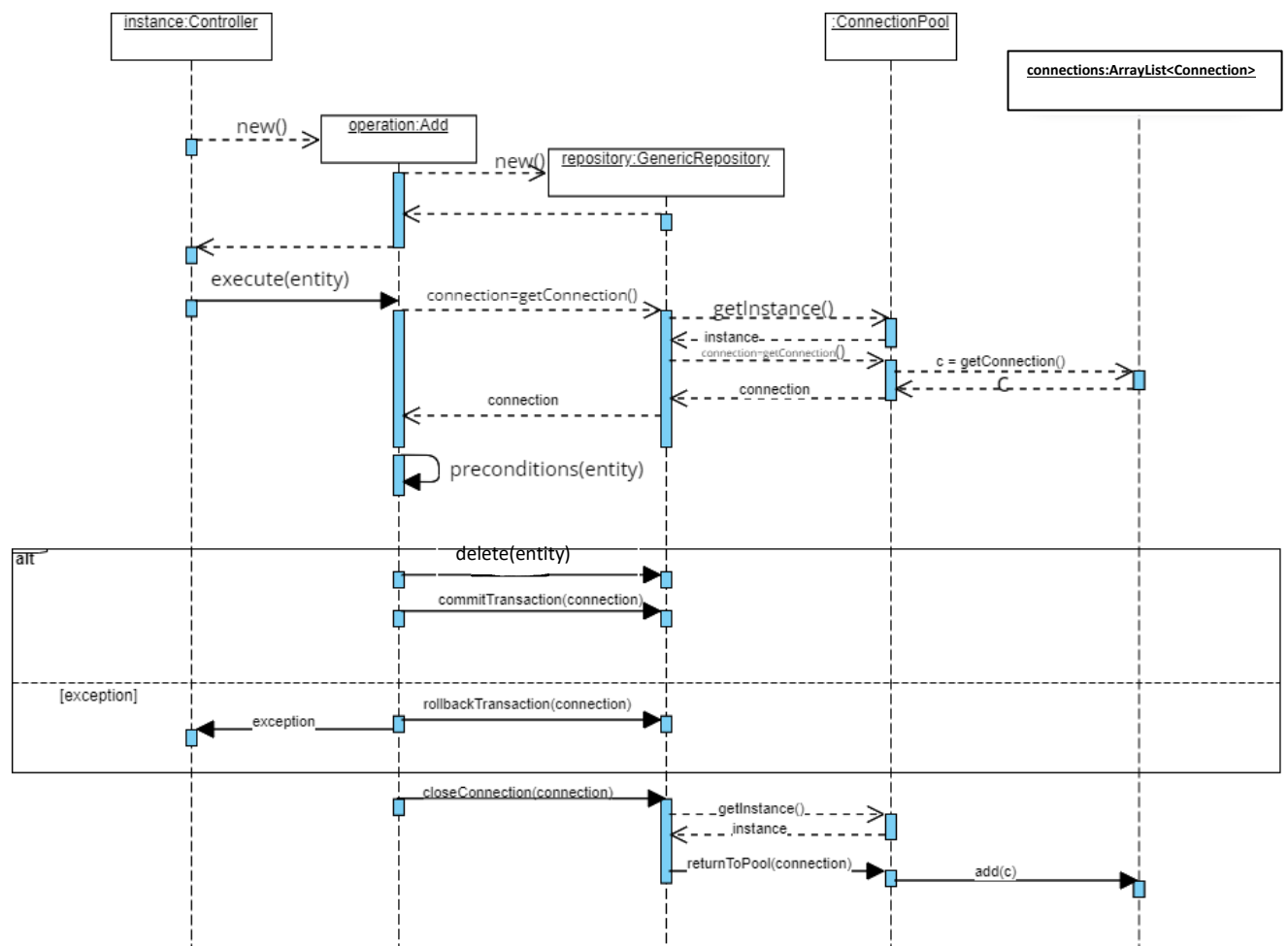
Уговор УГ2: Обриши ентитет

Операција: delete(IEntity),

Веза са СК: СК1

Предуслови: Очуван је интегритет базе података.

Постуслови: Обрисан је ентитет.



Слика 37: СД за операцију delete(IEntity)

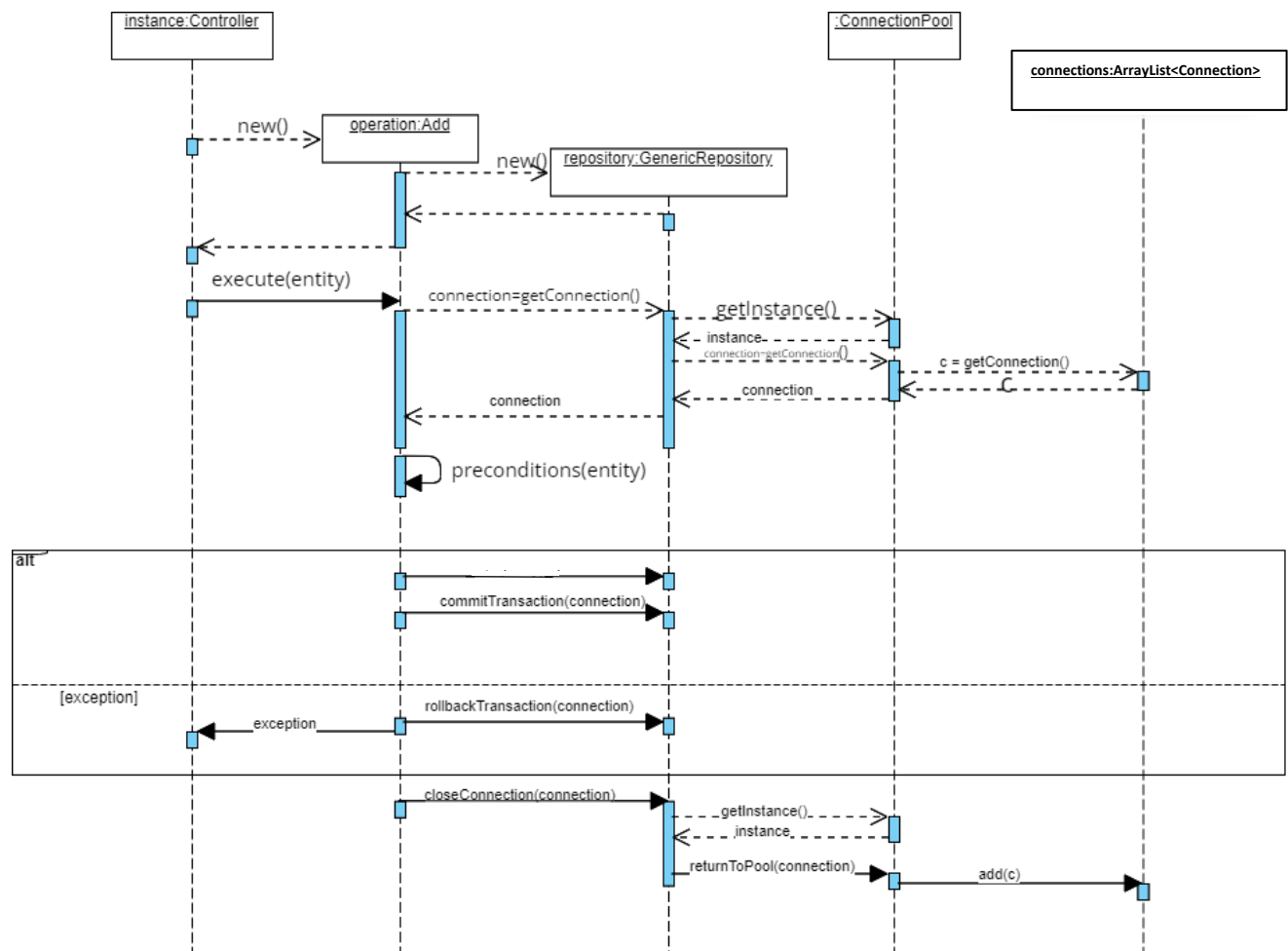
Уговор УГ3: Ажурирај ентитет

Операција: update(IEntity),

Веза са СК: СК1

Предуслови: Структурна и вредносна ограничења над ентитетом морају бити задовољена.

Постуслови: Ажуриран је ентитет.



Слика 38: СД за операцију edit(IEntity)

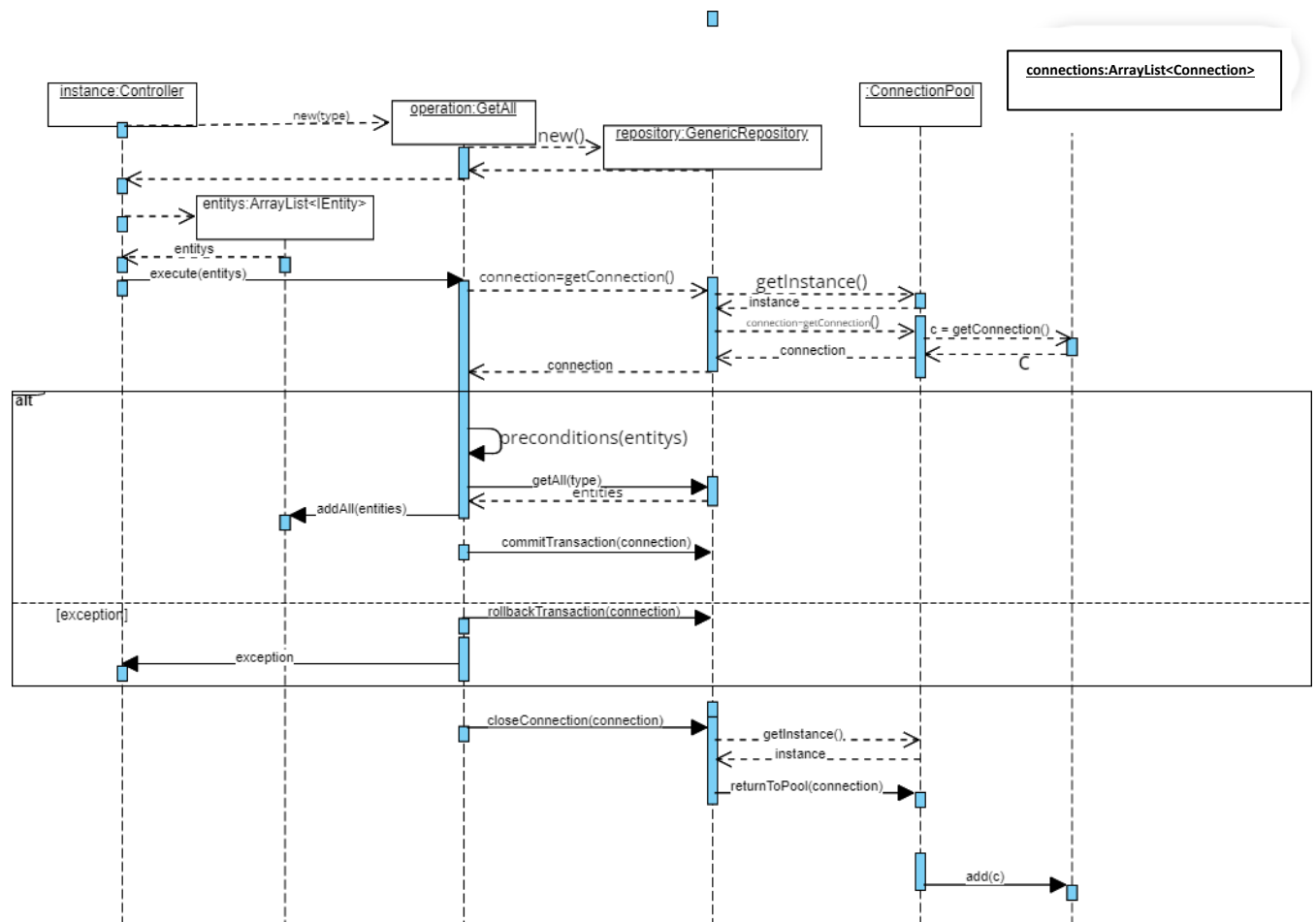
Уговор УГ4: Врати све ентитете

Операција: getAll(IEntity): List<IEntity>,

Веза са СК: СК1

Предуслови: /.

Постуслови: Учитана је листа ентитета.



Слика 39: СД за операцију `getAll(IEntity)`

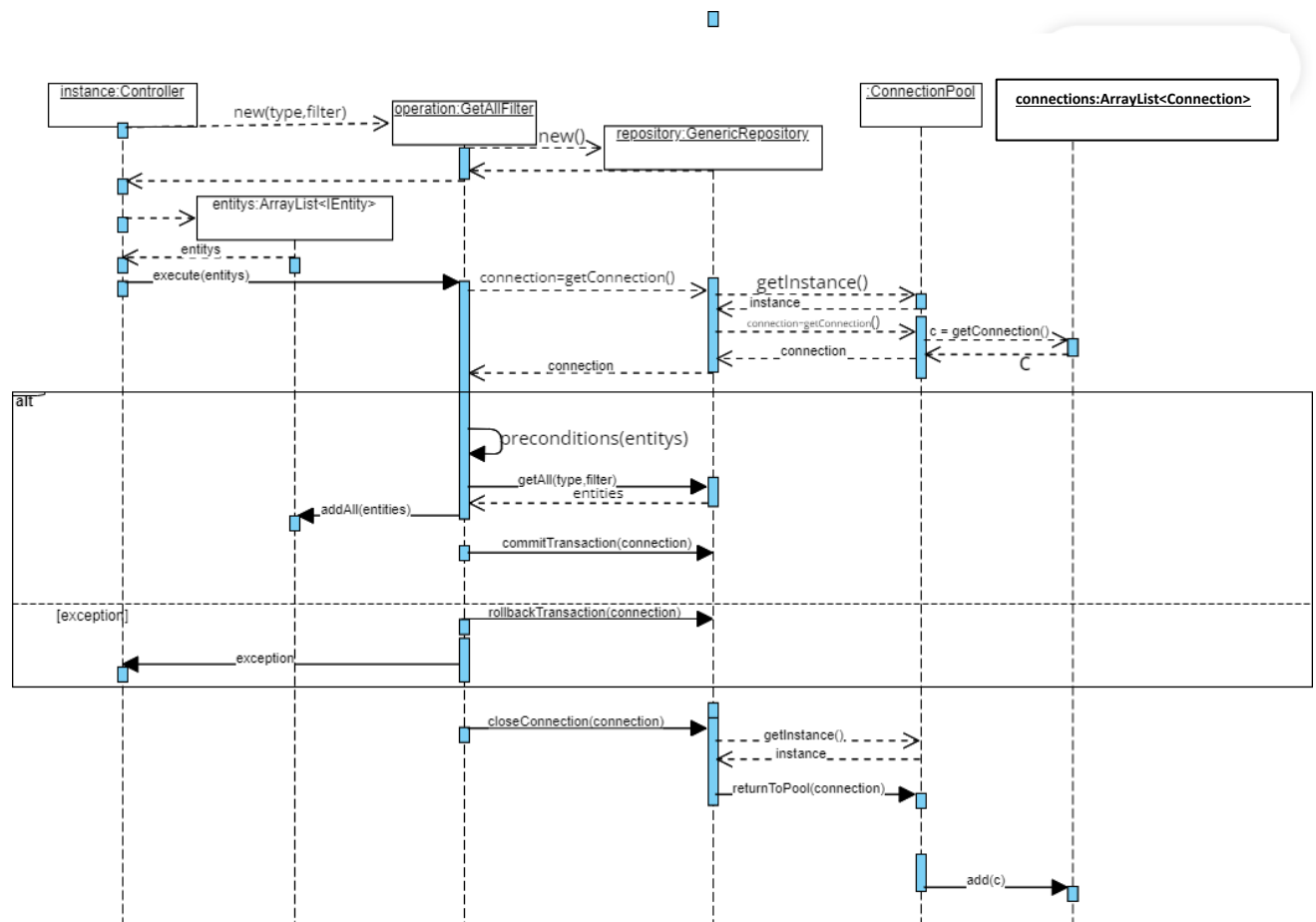
Уговор УГ5: Врати филтриране ентитете

Операција: getAllFilter IEntity, IEntity): List<IEntity>,

Веза са СК: СК1

Предуслови: /.

Постуслови: Учитана је листа ентитета.



Слика 40: СД за операцију `getAllFilter(IEntity, IEntity)`

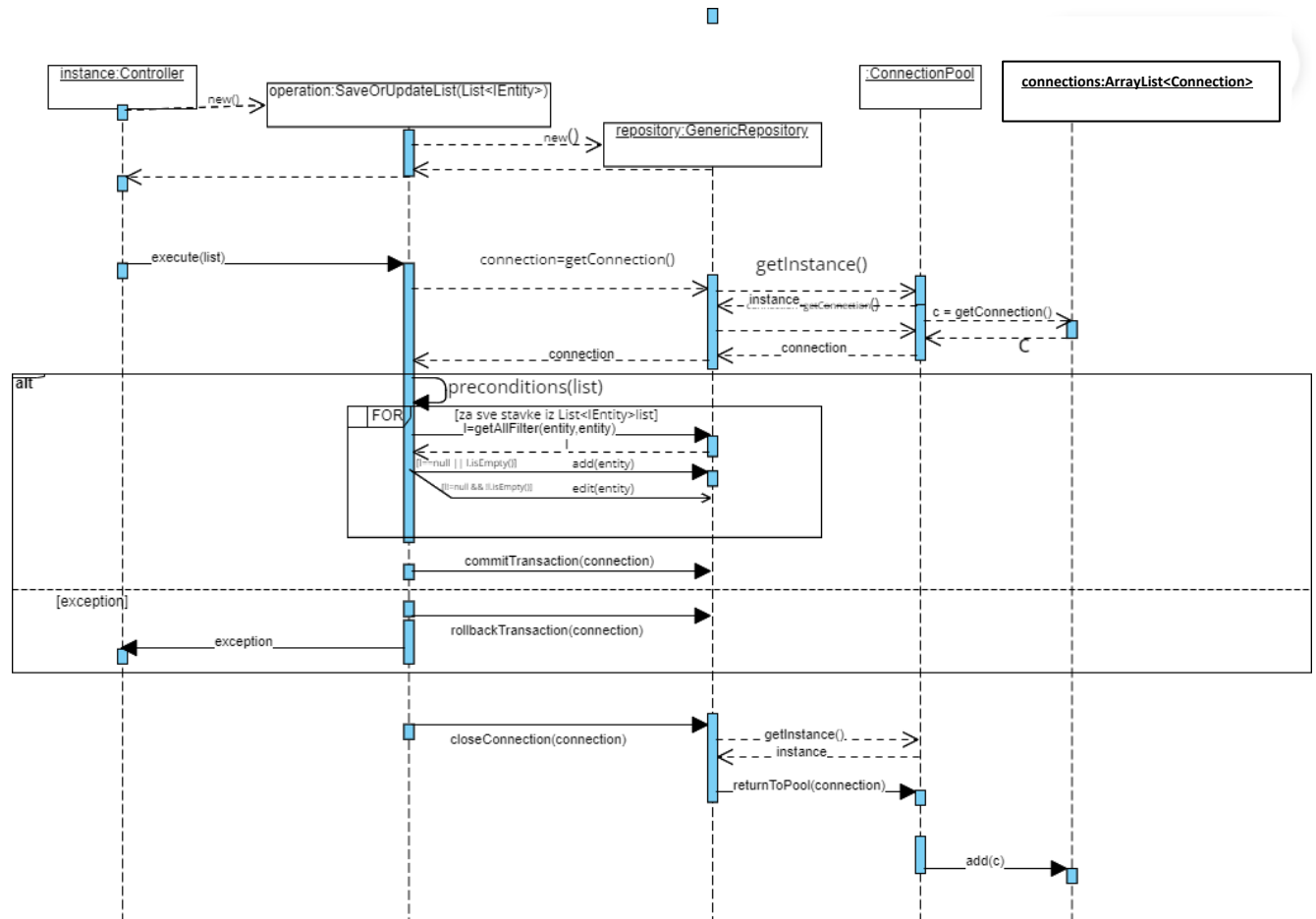
Уговор УГ6: Запамти листу ентитета

Операција: saveOrUpdateList(List<IEntity>)

Веза са СК: СК1

Предуслови: Структурна и вредносна ограничења над ентитетима морају бити задовољена.

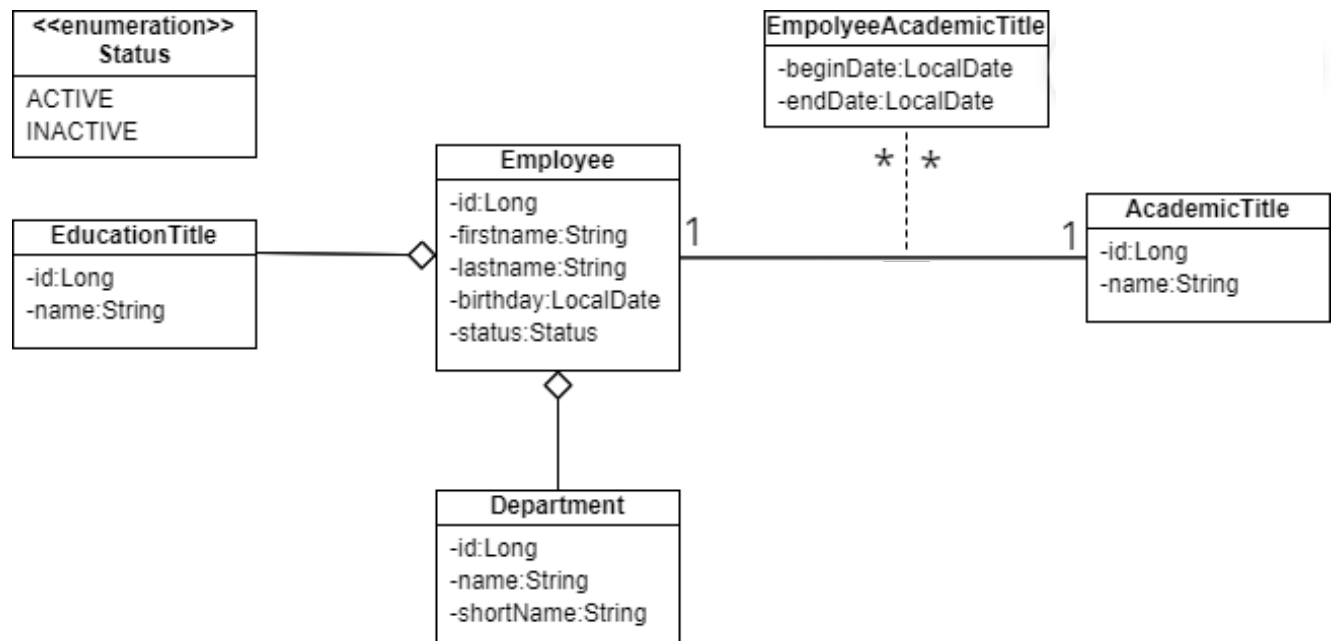
Постуслови: Ажурирана је листа ентитета.



Слика 41: СД за операцију `saveOrUpdateList(List<IEntity>)`

Пројектовање структуре софтверског система (доменске класе)

На основу концептуалних класа креирају се софтверске класе структуре.



Слика 42: Концептуални модел

Идентификоване су следеће класе:

- Employee

```
public class Employee implements IEntity {

    private Long id;
    private String firstname;
    private String lastname;
    private LocalDate birthday;
    private Department department;
    private AcademicTitle academicTitle;
    private EducationTitle educationTitle;
    private Status status;

    public Employee() {
    }
    public Employee(String firstname, String lastname, LocalDate birthday,
        Department department, AcademicTitle academicTitle, EducationTitle educationTitle, Status status) {
        this.firstname = firstname;
        this.lastname = lastname;
        this.birthday = birthday;
        this.department = department;
        this.academicTitle = academicTitle;
        this.educationTitle = educationTitle;
        this.status = status;
    }
}
```

Слика 43: Софтверска класа Employee

- Department

```

public class Department implements IEntity {

    private Long id;
    private String name;
    private String shortName;

    public Department() {
    }

    public Department(Long id, String name, String shortName) {
        this.id = id;
        this.name = name;
        this.shortName = shortName;
    }
}

```

Слика 44: Софтверска класа Department

- AcademicTitle

```

public class AcademicTitle implements IEntity{
    private Long id;
    private String name;

    public AcademicTitle() {
    }

    public AcademicTitle(Long id, String name) {
        this.id = id;
        this.name = name;
    }
}

```

Слика 45: Софтверска класа AcademicTitle

- EducationTitle

```

public class EducationTitle implements IEntity{

    private Long id;
    private String name;

    public EducationTitle() {
    }

    public EducationTitle(Long id, String name) {
        this.id = id;
        this.name = name;
    }
}

```

Слика 46: Софтверска класа EducationTitle

- EmployeeAcademicTitle

```
public class EmployeeAcademicTitle implements IEntity {  
  
    private Employee employee;  
    private AcademicTitle academicTitle;  
    private LocalDate beginDate;  
    private LocalDate endDate;  
  
    public EmployeeAcademicTitle() {  
    }  
    public EmployeeAcademicTitle(Employee employee, AcademicTitle academicTitle,  
        LocalDate beginDate, LocalDate endDate) {  
        this.employee = employee;  
        this.academicTitle = academicTitle;  
        this.beginDate = beginDate;  
        this.endDate = endDate;  
    }  
}
```

Слика 47: Софтверска класа EmployeeAcademicTitle

- Status

```
public enum Status implements Serializable{  
    ACTIVE,  
    INACTIVE;  
}
```

Слика 48: Софтверска класа Status

Пројектовање складишта података

Складиште података пројектује се на основу релационог и концептуалног модела.

Табела *Employee*:

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update
<input checked="" type="checkbox"/>	id	bigint	20		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	firstname	varchar	30		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	lastname	varchar	30		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	birthday	date			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	department	bigint	20		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	academicTitle	bigint	20		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	educationTitle	bigint	20		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	status	varchar	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Слика 49: Табела *Employee*

Табела *AcademicTitle*

Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update
id	bigint	20		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
name	varchar	50		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Слика 50: Табела *AcademicTitle*

Табела *EducationTitle*

Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update
id	bigint	20		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
name	varchar	30		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Слика 51: Табела *EducationTitle*

Табела *Department*

Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Updat
id	bigint	20		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
name	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
shortName	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

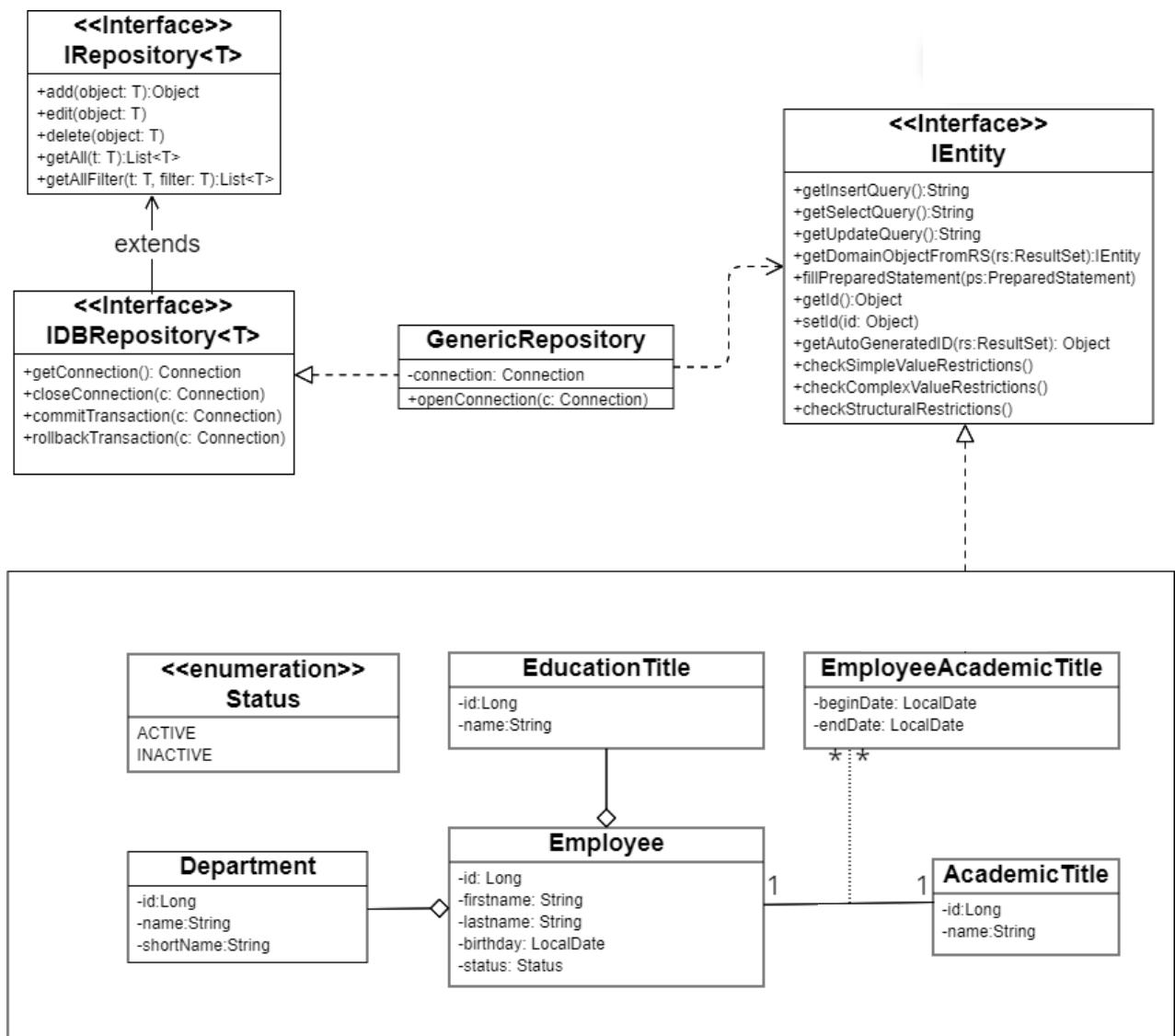
Слика 52: Табела *Department*

Табела *EmployeeAcademicTitle*

Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Updat
employee	bigint	20		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
academicTitle	bigint	20		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
beginDate	date			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
endDate	date			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

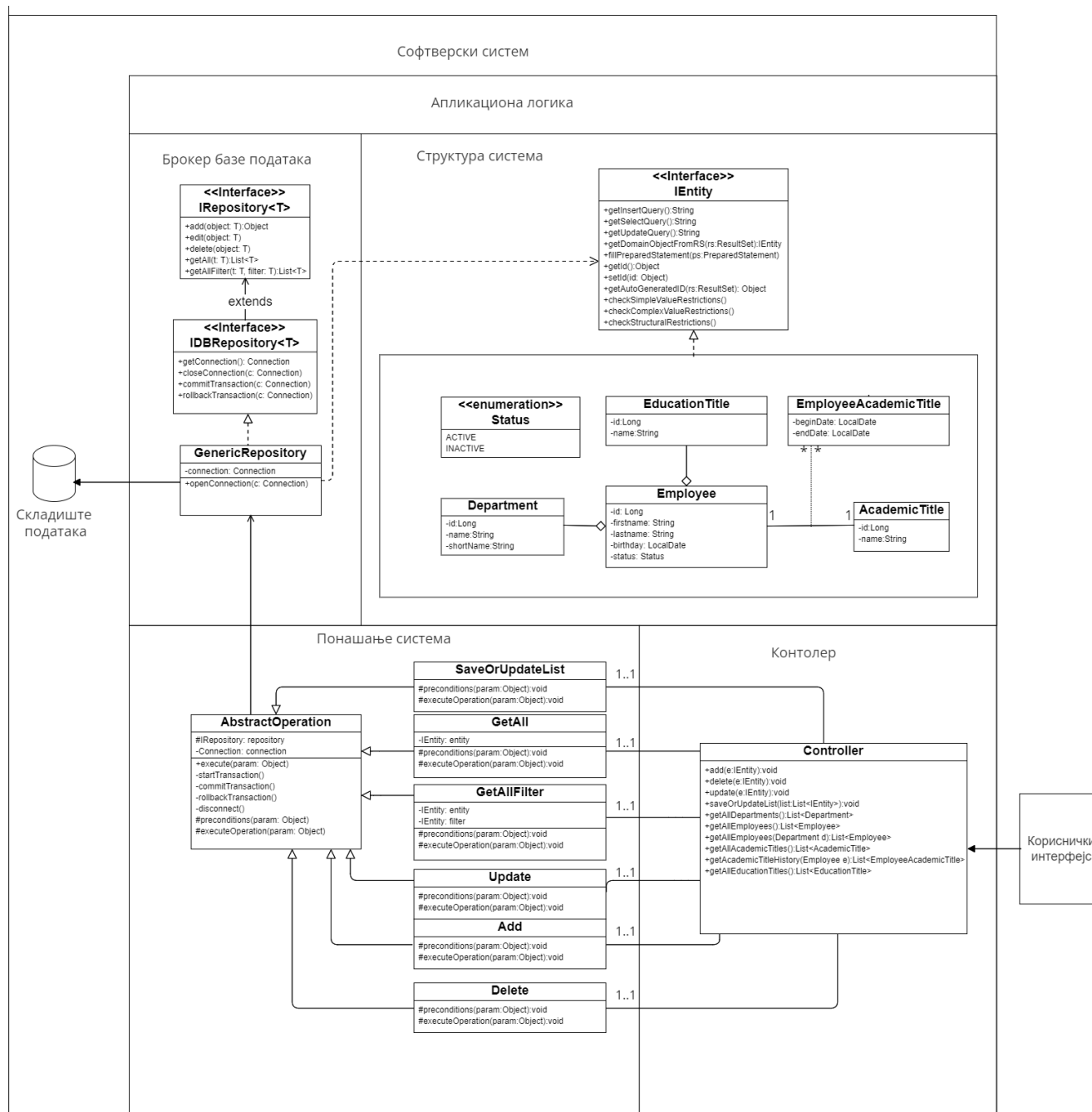
Слика 53: Табела *EmployeeAcademicTitle*

За комуникацију са базом података користимо такође и генеричку класу `AbstractOperation` која садржи методе за проверу предусlova, за извршавање операције и за управљање трансакцијом. Методе за проверу предусlova, за извршавање операције су апстрактне и њих ће имплементирати свака класа системске операције. Методе за управљање трансакцијом су имплементиране у класи `AbstractOperation`.



Слика 54: `GenericRepository` се повезује са `IEntity`

На основу претходних целина може се саставити цела архитектура софтверског система за евиденцију запослених на факултету.



Слика 55: Коначна архитектура софтверског система