

Универзитет у Београду
Факултет организационих наука
Катедра за софтверско инжењерство



Семинарски рад из предмета
Пројектовање софтвера

**Софтверски систем за резервацију одмора у
студентском одмаралишту у Јава окружењу**

Ментор: проф. др Синиша Влајић

Студент: Лука Стајковић

Број индекса: 2021/0129

Београд, 2024.

Садржај

1. Увод	3
2. Прикупљање корисничких захтева	4
2.1 Вербални опис	4
2.2 Случајеви коришћења	6
3. Анализа	17
3.1 Понашање софтверског система - Одређивање системских операција на основу сценарија случаја коришћења	17
3.2 Понашање софтверског система - Секвенцни дијаграми случаја коришћења	20
3.3 Понашање софтверског система - Дефинисање уговора о системским операцијама	28
3.4 Структура софтверског система – Концептуални (доменски) модел	30
3.5 Структура софтверског система – Релациони модел	31
3.6 Табела структурних и вредносних ограничења релационог модела	32
4. Пројектовање	35
4.1 Пројектовање корисничког интерфејса	36
4.2 Пројектовање апликационе логике	37
4.3 Пројектовање складишта података	38
5. Имплементација	39
6. Тестирање	40
7. Закључак	41
Литература	42

1. Увод

2. Прикупљање корисничких захтева

2.1 Вербални опис

Потребно је направити **Софтверски систем за резервацију одмора у студентском одмаралишту у Јава окружењу**. Софтверски систем, односно његова пословна логика (у општем смислу), састоји се из следећих *апстрактних концепата*:

а) **пружалац услуге**

б) **прималац услуге**

ц) **документ** који описује процес пружања услуге

д) **шифарници** у којима се налазе подаци о конкретним концептима који се користе у процесу пружања услуге, а који нису пружалац услуге, прималац услуге или документ који описује процес пружања услуге.

У наведеном софтверском систему *пружалац услуге* је **Службеник**, *прималац услуге* је **Студент**, *документ* који описује процес пружања услуге је **Фактура одмора**. *Шифарници* су **Ноћење**, **Смена**, **Факултет**.

Конкретни концепти су између себе повезани на следећи начин:

Преко фактуре је могуће уписати више ноћења. Фактура је повезана са једним службеником и једним студентом. Студент је везан за један факултет. Службеник може бити распоређен у више смена, док смена може бити везана за више службеника.

При пријављивању на софтверски систем потребно је обезбедити **аутентификацију** (преко корисничког имена и шифре) корисника софтверског система.

Потребно је обезбедити следеће функционалности за наведене конкретне концепте:

Редни број концепта	Концепт	Функционалности
1.	Фактура одмора	Креирај, Претражи, Промени
2.	Студент	Креирај, Претражи, Промени, Обриши
3.	Службеник	Пријави, Креирај, Претражи, Промени, Обриши
4.	Ноћење	Креирај, Претражи, Промени, Обриши
5.	Факултет	Креирај, Претражи, Промени, Обриши
6	Смена	Убаци, Претражи, Промени, Обриши
7.	Ставка фактуре	
8.	Службеник Смена	

Табела 1: Повезивање концепата и функционалности

Функционалностима софтверског система се приступа преко главног менија који има следећу структуру:

1. Документи
 - 1.1 Фактура одмора
2. Пружалац услуге
 - 2.1 Службеник
3. Прималац услуге
 - 3.1 Студент
4. Шифарници
 - 4.1 Ноћење,
 - 4.2 Смена,
 - 4.3 Факултет,
5. Подешавања софтверског система
6. О програму

2.2 Случајеви коришћења

На основу наведених функционалности концепата уочени су следећи случајеви коришћења:

Шифра случаја коришћења	Назив случаја коришћења	Предуслови С.К. (листе)	Критеријуми претраживања се односе на:
SK1	СК1- Креирај фактура одмора	Учитане су листе: а) Службеник б) Студент с) Ноћење	
SK2	СК2- Претражи фактура одмора		а) Фактура одмора б) Службеник в) Студент д) Ноћење
SK3	СК3- Промени фактура одмора	Учитане су листе: а) Службеник б) Студент с) Ноћење	а) Фактура одмора б) Службеник в) Студент д) Ноћење
SK4	СК4- Креирај студент	Учитане су листе: а) Факултет	
SK5	СК5- Претражи студент		а) Студент б) Факултет
SK6	СК6- Промени студент	Учитане су листе: а) Факултет	а) Студент б) Факултет
SK7	СК7- Обриши студент		а) Студент б) Факултет
SK8	СК8- Пријави службеник		
SK9	СК9- Креирај службеник	Учитане су листе: а) Смена	
SK10	СК10- Претражи службеник		а) Службеник б) Смена
SK11	СК11- Промени службеник	Учитане су листе: а) Смена	а) Службеник б) Смена
SK12	СК12- Обриши службеник		а) Службеник б) Смена
SK13	СК13- Креирај ноћење		
SK14	СК14- Претражи ноћење		а) Ноћење
SK15	СК15- Промени ноћење		а) Ноћење
SK16	СК16- Обриши ноћење		а) Ноћење
SK17	СК17- Креирај факултет		
SK18	СК18- Претражи факултет		а) Факултет
SK19	СК19- Промени факултет		а) Факултет
SK20	СК20- Обриши факултет		а) Факултет
SK21	СК21- Убаци смена		
SK22	СК22- Претражи смена		а) Смена
SK23	СК23- Промени смена		а) Смена
SK24	СК24- Обриши смена		а) Смена

Табела 2: Случајеви коришћења софтверског система

За следеће случајеве коришћења ћемо дати детаљан опис:

СК1- Креирај фактура одмора
СК2- Претражи фактура одмора
СК3- Промени фактура одмора
СК4- Креирај студент
СК5- Претражи студент
СК6- Промени студент
СК7- Обриши студент
СК8- Пријави службеник
СК21- Убаци смена

Табела 3: Случајеви коришења за које ће бити дат детаљан опис

СК1- Креирај фактура одмора

Назив СК

Креирај фактура одмора

Актори СК

Службеник

Учесници СК

Службеник, кориснички интерфејс (клијентски програм) и систем (серверски програм)

Предуслови: Кориснички интерфејс (клијентски програм) и систем (серверски програм) су покренути. Службеник је пријављен под својом шифром. Систем приказује форму за рад са фактуром одмора. *Учитане су листе: а) Службеник б) Студент с) Ноћење*

Основни сценарио СК:

1. Службеник позива систем да креира фактуру одмора. (АПСО)
2. Систем креира фактуру одмора. (СО)
3. Систем приказује службенику фактуру одмора и поруку: “Систем је креирао фактуру одмора”. (ИА)
4. Службеник уноси податке о фактури одмора. (АПУСО)
5. Службеник контролише да ли је коректно унео податке о фактури одмора. (АНСО)
6. Службеник позива систем да запамти податке о фактури одмора. (АПСО)
7. Систем памти податке о фактури одмора. (СО)
8. Систем приказује службенику фактуру одмора и поруку: “Систем је запамтио фактуру одмора.” (ИА)

Алтернативна сценарија:

- 3.1 Уколико систем не може да креира фактуру одмора он приказује службенику поруку: “Систем не може да креира фактуру одмора”. Прекида се извршење сценарија. (ИА)
- 8.1 Уколико систем не може да запамти податке о фактури одмора он приказује службенику поруку: “Систем не може да запамти фактуру одмора”. (ИА)

СК2- Претражи фактура одмора

Назив СК

Претражи фактура одмора

Актори СК

Службеник

Учесници СК

Службеник, кориснички интерфејс (клијентски програм) и систем (серверски програм)

Предуслови: Кориснички интерфејс (клијентски програм) и систем (серверски програм) су покренути. Службеник је пријављен под својом шифром. Кориснички интерфејс приказује форму за рад са фактуром одмора. На наведеној екранској форми су дефинисани критеријуми, који се односе на: а) Фактура одмора б) Службеник с) Студент д) Ноћење, који ће да врате листу фактуре одмора.

Основни сценарио СК:

1. Службеник бира критеријуме на основу којих претражује фактуре одмора. (АПУСО)
2. Службеник позива систем да нађе фактуре одмора по задатим критеријумима. (АПСО)
3. Систем тражи фактуре одмора по задатим критеријумима. (СО)
4. Систем приказује службенику фактуре одмора и поруку: "Систем је нашао фактуре одмора по задатим критеријумима". (ИА)
5. Службеник бира фактуру одмора. (АПУСО)
6. Службеник позива систем да нађе фактуру одмора. (АПСО)
7. Систем тражи фактуру одмора. (СО)
8. Систем приказује службенику фактуру одмора и поруку: "Систем је нашао фактуру одмора". (ИА)

Алтернативна сценарија:

4.1 Уколико систем не може да нађе фактуре одмора он приказује службенику поруку: "Систем не може да нађе фактуре одмора по задатим критеријумима". Прекида се извршење сценарија. (ИА)

8.1 Уколико систем не може да нађе фактуру одмора он приказује службенику поруку: "Систем не може да нађе фактуру одмора". (ИА)

СКЗ- Промени фактура одмора

Назив СК

Промени фактура одмора

Актери СК

Службеник

Учесници СК

Службеник, кориснички интерфејс (клијентски програм) и систем (серверски програм)

Предуслови: Кориснички интерфејс (клијентски програм) и систем (серверски програм) су покренути. Службеник је пријављен под својом шифром. Кориснички интерфејс приказује форму за рад са фактуром одмора. На наведеној екранској форми су дефинисани критеријуми, који се односе на: а) Фактура одмора б) Службеник с) Студент д) Ноћење, који ће да врате листу фактуре одмора. Учитане су листе: а) Службеник б) Студент с) Ноћење

Основни сценарио СК:

1. Службеник бира критеријуме на основу којих претражује фактуре одмора. (АПУСО)
2. Службеник позива систем да нађе фактуре одмора по задатим критеријумима. (АПСО)
3. Систем тражи фактуре одмора по задатим критеријумима. (СО)
4. Систем приказује службенику фактуре одмора и поруку: "Систем је нашао фактуре одмора по задатим критеријумима". (ИА)
5. Службеник бира фактуру одмора. (АПУСО)
6. Службеник позива систем да нађе фактуру одмора. (АПСО)
7. Систем тражи фактуру одмора. (СО)
8. Систем приказује службенику фактуру одмора и поруку: "Систем је нашао фактуру одмора". (ИА)
9. Службеник уноси (мења) податке о фактури одмора. (АПУСО)
10. Службеник контролише да ли је коректно унео податке о фактури одмора. (АНСО)
11. Службеник позива систем да запамти податке о фактури одмора. (АПСО)
12. Систем памти податке о фактури одмора. (СО)
13. Систем приказује службенику фактуру одмора и поруку: "Систем је запамтио фактуру одмора." (ИА)

Алтернативна сценарија:

- 4.1 Уколико систем не може да нађе фактуре одмора он приказује службенику поруку: "Систем не може да нађе фактуре одмора по задатом критеријуму". Прекида се извршење сценарија. (ИА)
- 8.1 Уколико систем не може да нађе фактуру одмора он приказује службенику поруку: "Систем не може да нађе фактуру одмора". Прекида се извршење сценарија. (ИА)
- 13.1 Уколико систем не може да запамти податке о фактури одмора он приказује службенику поруку: "Систем не може да запамти фактуру одмора". (ИА)

СК4- Креирај студент

Назив СК

Креирај студент

Актори СК

Службеник

Учесници СК

Службеник, кориснички интерфејс (клијентски програм) и систем (серверски програм)

Предуслови: Кориснички интерфејс (клијентски програм) и систем (серверски програм) су покренути. Службеник је пријављен под својом шифром. Систем приказује форму за рад са студентом. *Учитане су листе: а) Факултет*

Основни сценарио СК:

1. Службеник позива систем да креира студента. (АПСО)
2. Систем креира студента. (СО)
3. Систем приказује службенику студента и поруку: "Систем је креирао студента". (ИА)
4. Службеник уноси податке о студенту. (АПУСО)
5. Службеник контролише да ли је коректно унео податке о студенту. (АНСО)
6. Службеник позива систем да запамти податке о студенту. (АПСО)
7. Систем памти податке о студенту. (СО)
8. Систем приказује службенику студента и поруку: "Систем је запамтио студента." (ИА)

Алтернативна сценарија:

- 3.1 Уколико систем не може да креира студента он приказује службенику поруку: "Систем не може да креира студента". Прекида се извршење сценарија. (ИА)
- 8.1 Уколико систем не може да запамти податке о студенту он приказује службенику поруку: "Систем не може да запамти студента". (ИА)

СК5- Претражи студент

Назив СК

Претражи студент

Актори СК

Службеник

Учесници СК

Службеник, кориснички интерфејс (клијентски програм) и систем (серверски програм)

Предуслови: Кориснички интерфејс (клијентски програм) и систем (серверски програм) су покренути. Службеник је пријављен под својом шифром. Кориснички интерфејс приказује форму за рад са студентом. На наведеној екранској форми су дефинисани критеријуми, који се односе на: а) Студент б) Факултет, који ће да врате листу студената.

Основни сценарио СК:

1. Службеник бира критеријуме на основу којих претражује студенте. (АПУСО)
2. Службеник позива систем да нађе студенте по задатим критеријумима. (АПСО)
3. Систем тражи студенте по задатим критеријумима. (СО)
4. Систем приказује службенику студенте и поруку: "Систем је нашао студенте по задатим критеријумима". (ИА)
5. Службеник бира студента. (АПУСО)
6. Службеник позива систем да нађе студента. (АПСО)
7. Систем тражи студента. (СО)
8. Систем приказује службенику студента и поруку: "Систем је нашао студента". (ИА)

Алтернативна сценарија:

- 4.1 Уколико систем не може да нађе студенте он приказује службенику поруку: "Систем не може да нађе студенте по задатим критеријумима". Прекида се извршење сценарија. (ИА)
- 8.1 Уколико систем не може да нађе студента он приказује службенику поруку: "Систем не може да нађе студента". (ИА)

СК6- Промени студент

Назив СК

Промени студент

Актери СК

Службеник

Учесници СК

Службеник, кориснички интерфејс (клијентски програм) и систем (серверски програм)

Предуслови: Кориснички интерфејс (клијентски програм) и систем (серверски програм) су покренути. Службеник је пријављен под својом шифром. Кориснички интерфејс приказује форму за рад са студентом. На наведеној екранској форми су дефинисани критеријуми, који се односе на: а) Студент б) Факултет, који ће да врате листу студената. Учитане су листе: а) Факултет

Основни сценарио СК:

1. Службеник бира критеријуме на основу којих претражује студенте. (АПУСО)
2. Службеник позива систем да нађе студенте по задатим критеријумима. (АПСО)
3. Систем тражи студенте по задатим критеријумима. (СО)
4. Систем приказује службенику студенте и поруку: "Систем је нашао студенте по задатим критеријумима". (ИА)
5. Службеник бира студента. (АПУСО)
6. Службеник позива систем да нађе студента. (АПСО)
7. Систем тражи студента. (СО)
8. Систем приказује службенику студента и поруку: "Систем је нашао студента". (ИА)
9. Службеник уноси (мења) податке о студенту. (АПУСО)
10. Службеник контролише да ли је коректно унео податке о студенту. (АНСО)
11. Службеник позива систем да запамти податке о студенту. (АПСО)
12. Систем памти податке о студенту. (СО)
13. Систем приказује службенику студента и поруку: "Систем је запамтио студента." (ИА)

Алтернативна сценарија:

- 4.1 Уколико систем не може да нађе студенте он приказује службенику поруку: "Систем не може да нађе студенте по задатом критеријуму". Прекида се извршење сценарија. (ИА)
- 8.1 Уколико систем не може да нађе студента он приказује службенику поруку: "Систем не може да нађе студента". Прекида се извршење сценарија. (ИА)
- 13.1 Уколико систем не може да запамти податке о студенту он приказује службенику поруку: "Систем не може да запамти студента". (ИА)

СК7- Обриши студент

Назив СК

Обриши студент

Актори СК

Службеник

Учесници СК

Службеник, кориснички интерфејс (клијентски програм) и систем (серверски програм)

Предуслови: Кориснички интерфејс (клијентски програм) и систем (серверски програм) су покренути. Службеник је пријављен под својом шифром. Кориснички интерфејс приказује форму за рад са студентом. На наведеној екранској форми су дефинисани критеријуми, који се односе на: а) Студент б) Факултет, који ће да врате листу студената.

Основни сценарио СК:

1. Службеник бира критеријуме на основу којих претражује студенте. (АПУСО)
2. Службеник позива систем да нађе студенте по задатим критеријумима. (АПСО)
3. Систем тражи студенте по задатим критеријумима. (СО)
4. Систем приказује службенику студенте и поруку: "Систем је нашао студенте по задатим критеријумима". (ИА)
5. Службеник бира студента. (АПУСО)
6. Службеник позива систем да нађе студента. (АПСО)
7. Систем тражи студента. (СО)
8. Систем приказује службенику студента и поруку: "Систем је нашао студента". (ИА)
9. Службеник позива систем да обрише студента. (АПСО)
10. Систем брише студента. (СО)
11. Систем приказује службенику поруку: "Систем је обрисао студента." (ИА)

Алтернативна сценарија:

- 4.1 Уколико систем не може да нађе студенте он приказује службенику поруку: "Систем не може да нађе студенте по задатим критеријумима". Прекида се извршење сценарија. (ИА)
- 8.1 Уколико систем не може да нађе студента он приказује службенику поруку: "Систем не може да нађе студента". Прекида се извршење сценарија. (ИА)
- 11.1 Уколико систем не може да обрише студента он приказује службенику поруку: "Систем не може да обрише студента". (ИА)

СК8- Пријави службеник

Назив СК

Пријави службеник

Актори СК

Службеник

Учесници СК

Службеник, кориснички интерфејс (клијентски програм) и систем (серверски програм)

Предуслови: Кориснички интерфејс (клијентски програм) и систем (серверски програм) су покренути. Кориснички интерфејс приказује форму за **пријављивање**.

Основни сценарио СК:

1. Службеник **уноси** корисничко име и шифру. (АПУСО)
2. Службеник **контролише** да ли је коректно унео корисничко име и шифру. (АНСО)
3. Службеник **позива систем** да провери корисничко име и шифру. (АПСО)
4. Систем **проверава** корисничко име и шифру. (СО)
5. Систем **приказује службенику** поруку: "Корисничко име и шифра су исправни." (ИА)
6. Кориснички интерфејс **позива** главни форму и мени. (КИПГФМ)

Алтернативна сценарија:

- 5.1 Уколико систем провером установи да корисничка шифра и/или шифра нису исправни он **приказује службенику** поруку: "Корисничко име и шифра нису исправни". (ИА)
- 6.1 Уколико кориснички интерфејс не може да отвори главну форму и мени **приказује продавцу** поруку: "Не може да се отвори главна форма и мени". (НПГФМ)

Постуслови: Отворена главна форма и мени.

СК21- Убаца смена

Назив СК

Убаца смена

Актери СК

Службеник

Учесници СК

Службеник, **кориснички интерфејс** (клијентски програм) и **систем** (серверски програм)

Предуслови: **Кориснички интерфејс** (клијентски програм) и **систем** (серверски програм) су покренути. **Службеник** је пријављен под својом шифром. Систем приказује форму за рад са **сменом**.

Основни сценарио СК:

1. **Службеник уноси** податке о **смени**. (АПУСО)
2. **Службеник контролише** да ли је коректно унео податке о **смени**. (АНСО)
3. **Службеник позива систем** да запамти податке о **смени**. (АПСО)
4. **Систем памти** податке о **смени**. (СО)
5. **Систем приказује службеник сменом** и поруку: “**Систем** је запамтио **сменом**.” (ИА)

Алтернативна сценарија:

5.1 Уколико **систем** не може да запамти податке о **смени** он **приказује службеник** поруку: “**Систем** не може да запамти **сменом**”. (ИА)

3. Анализа

3.1 Понашање софтверског система - Одређивање системских операција на основу сценарија случаја коришћења

На основу наведених случајева коришћења уочене су следеће системске операције:

Шифра случаја коришћења	Назив случаја коришћења	Системске операције
SK1	СК1- Креирај фактура одмора	1. signal KreirajFakturaOdmora(FakturaOdmora) 2. signal PromeniFakturaOdmora(FakturaOdmora) 3. signal vratiListuSviSluzbenik(Lista<Sluzbenik>) 4. signal vratiListuSviStudent(Lista<Student>) 5. signal vratiListuSviNocenje(Lista<Nocenje>)
SK2	СК2- Претражи фактура одмора	1. signal PretraziFakturaOdmora(FakturaOdmora) 2. signal vratiListuFakturaOdmora(kriterijumFakturaOdmora, Lista<FakturaOdmora>) 3. signal vratiListuFakturaOdmora(kriterijumSluzbenik, Lista<FakturaOdmora>) 4. signal vratiListuFakturaOdmora(kriterijumStudent, Lista<FakturaOdmora>) 5. signal vratiListuFakturaOdmora(kriterijumNocenje, Lista<FakturaOdmora>)
SK3	СК3- Промени фактура одмора	1. signal PromeniFakturaOdmora(FakturaOdmora) 2. signal PretraziFakturaOdmora(FakturaOdmora) 3. signal vratiListuSviSluzbenik(Lista<Sluzbenik>) 4. signal vratiListuSviStudent(Lista<Student>) SO6
SK4	СК4- Креирај студент	1. signal KreirajStudent(Student) 2. signal PromeniStudent(Student) 3. signal vratiListuSviFakultet(Lista<Fakultet>)
SK5	СК5- Претражи студент	1. signal PretraziStudent(Student) 2. signal vratiListuStudent(kriterijumStudent, Lista<Student>) 3. signal vratiListuStudent(kriterijumFakultet, Lista<Student>)
SK6	СК6- Промени студент	1. signal PromeniStudent(Student) 2. signal PretraziStudent(Student) 3. signal vratiListuSviFakultet(Lista<Fakultet>) 4. signal vratiListuStudent(kriterijumStudent, Lista<Student>) 5. signal vratiListuStudent(kriterijumFakultet, Lista<Student>)
SK7	СК7- Обриши студент	1. signal ObrisiStudent(Student) 2. signal vratiListuStudent(kriterijumStudent, Lista<Student>) 3. signal vratiListuStudent(kriterijumFakultet, Lista<Student>)
SK8	СК8- Пријави службеник	1. signal PrijaviSluzbenik(korisnickolme, sifra)
SK9	СК9- Креирај службеник	1. signal KreirajSluzbenik(Sluzbenik) 2. signal PromeniSluzbenik(Sluzbenik) 3. signal vratiListuSviSmena(Lista<Smena>)
SK10	СК10- Претражи службеник	1. signal PretraziSluzbenik(Sluzbenik) 2. signal vratiListuSluzbenik(kriterijumSluzbenik, Lista<Sluzbenik>) 3. signal vratiListuSluzbenik(kriterijumSmena, Lista<Sluzbenik>)
SK11	СК11- Промени службеник	1. signal PromeniSluzbenik(Sluzbenik) 2. signal PretraziSluzbenik(Sluzbenik) 3. signal vratiListuSviSmena(Lista<Smena>) 4. signal vratiListuSluzbenik(kriterijumSluzbenik, Lista<Sluzbenik>) 5. signal vratiListuSluzbenik(kriterijumSmena, Lista<Sluzbenik>)
SK12	СК12- Обриши службеник	1. signal ObrisiSluzbenik(Sluzbenik) 2. signal vratiListuSluzbenik(kriterijumSluzbenik, Lista<Sluzbenik>) 3. signal vratiListuSluzbenik(kriterijumSmena, Lista<Sluzbenik>)
SK13	СК13- Креирај ноћење	1. signal KreirajNocenje(Nocenje) 2. signal PromeniNocenje(Nocenje)
SK14	СК14- Претражи ноћење	1. signal PretraziNocenje(Nocenje) 2. signal vratiListuNocenje(kriterijumNocenje, Lista<Nocenje>)
SK15	СК15- Промени ноћење	1. signal PromeniNocenje(Nocenje) 2. signal PretraziNocenje(Nocenje) 3. signal vratiListuNocenje(kriterijumNocenje, Lista<Nocenje>)

SK16	СК16-Обриши ноћење	1. signal ObrisiNocenje(Nocenje) 2. signal vratiListuNocenje(kriterijumNocenje, Lista<Nocenje>)
SK17	СК17-Креирај факултет	1. signal KreirajFakultet(Fakultet) 2. signal PromeniFakultet(Fakultet)
SK18	СК18-Претражи факултет	1. signal PretraziFakultet(Fakultet) 2. signal vratiListuFakultet(kriterijumFakultet, Lista<Fakultet>)
SK19	СК19-Промени факултет	1. signal PromeniFakultet(Fakultet) 2. signal PretraziFakultet(Fakultet) 3. signal vratiListuFakultet(kriterijumFakultet, Lista<Fakultet>)
SK20	СК20-Обриши факултет	1. signal ObrisiFakultet(Fakultet) 2. signal vratiListuFakultet(kriterijumFakultet, Lista<Fakultet>)
SK21	СК21-Убаци смена	1. signal UbaciSmena(Smena) 2. signal PromeniSmena(Smena)
SK22	СК22-Претражи смена	1. signal PretraziSmena(Smena) 2. signal vratiListuSmena(kriterijumSmena, Lista<Smena>)
SK23	СК23-Промени смена	1. signal PromeniSmena(Smena) 2. signal PretraziSmena(Smena) 3. signal vratiListuSmena(kriterijumSmena, Lista<Smena>)
SK24	СК24-Обриши смена	1. signal ObrisiSmena(Smena) 2. signal vratiListuSmena(kriterijumSmena, Lista<Smena>)

Табела 4: Системске операције случаја коришћења

Наводимо све уочене системске операције:

1. signal KreirajFakturaOdmora(FakturaOdmora)
2. signal KreirajFakultet(Fakultet)
3. signal KreirajNocenje(Nocenje)
4. signal KreirajSluzbenik(Sluzbenik)
5. signal KreirajStudent(Student)
6. signal ObrisiFakultet(Fakultet)
7. signal ObrisiNocenje(Nocenje)
8. signal ObrisiSluzbenik(Sluzbenik)
9. signal ObrisiSmena(Smena)
10. signal ObrisiStudent(Student)
11. signal PretraziFakturaOdmora(FakturaOdmora)
12. signal PretraziFakultet(Fakultet)
13. signal PretraziNocenje(Nocenje)
14. signal PretraziSluzbenik(Sluzbenik)
15. signal PretraziSmena(Smena)
16. signal PretraziStudent(Student)
17. signal PrijaviSluzbenik(korisnickolme, sifra)
18. signal PromeniFakturaOdmora(FakturaOdmora)
19. signal PromeniFakultet(Fakultet)
20. signal PromeniNocenje(Nocenje)
21. signal PromeniSluzbenik(Sluzbenik)
22. signal PromeniSmena(Smena)
23. signal PromeniStudent(Student)
24. signal UbaciSmena(Smena)
25. signal vratiListuFakturaOdmora(kriterijumFakturaOdmora, Lista<FakturaOdmora>)
26. signal vratiListuFakturaOdmora(kriterijumNocenje, Lista<FakturaOdmora>)
27. signal vratiListuFakturaOdmora(kriterijumSluzbenik, Lista<FakturaOdmora>)
28. signal vratiListuFakturaOdmora(kriterijumStudent, Lista<FakturaOdmora>)
29. signal vratiListuFakultet(kriterijumFakultet, Lista<Fakultet>)
30. signal vratiListuNocenje(kriterijumNocenje, Lista<Nocenje>)
31. signal vratiListuSluzbenik(kriterijumSluzbenik, Lista<Sluzbenik>)
32. signal vratiListuSluzbenik(kriterijumSmena, Lista<Sluzbenik>)
33. signal vratiListuSmena(kriterijumSmena, Lista<Smena>)
34. signal vratiListuStudent(kriterijumFakultet, Lista<Student>)
35. signal vratiListuStudent(kriterijumStudent, Lista<Student>)

36. signal vratiListuSviFakultet(Lista<Fakultet>)
37. signal vratiListuSviNocenje(Lista<Nocenje>)
38. signal vratiListuSviSluzbenik(Lista<Sluzbenik>)
39. signal vratiListuSviSmena(Lista<Smena>)
40. signal vratiListuSviStudent(Lista<Student>)

Табела 5: Системске операције софтверског система

3.2 Понашање софтверског система - Секвенци дијаграми случаја коришћења

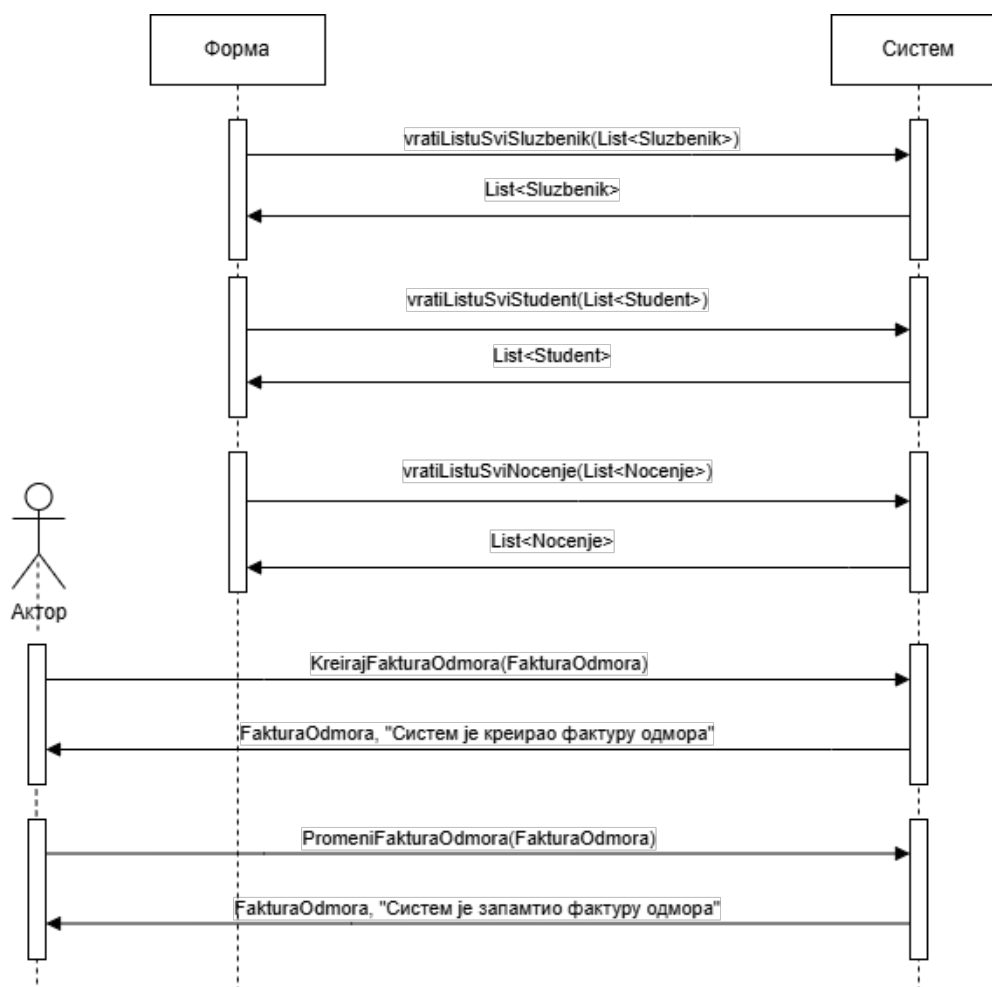
ДС1: Дијаграми секвенци случаја коришћења – Креирај фактура одмора

Предуслови:

1. **Форма** позива систем да **врати** листу свих службеника. (АПСО)
2. **Систем** враћа **форми** листу свих службеника. (ИА)
3. **Форма** позива систем да **врати** листу свих студената. (АПСО)
4. **Систем** враћа **форми** листу свих студената. (ИА)
5. **Форма** позива систем да **врати** листу свих ноћења. (АПСО)
6. **Систем** враћа **форми** листу свих ноћења. (ИА)

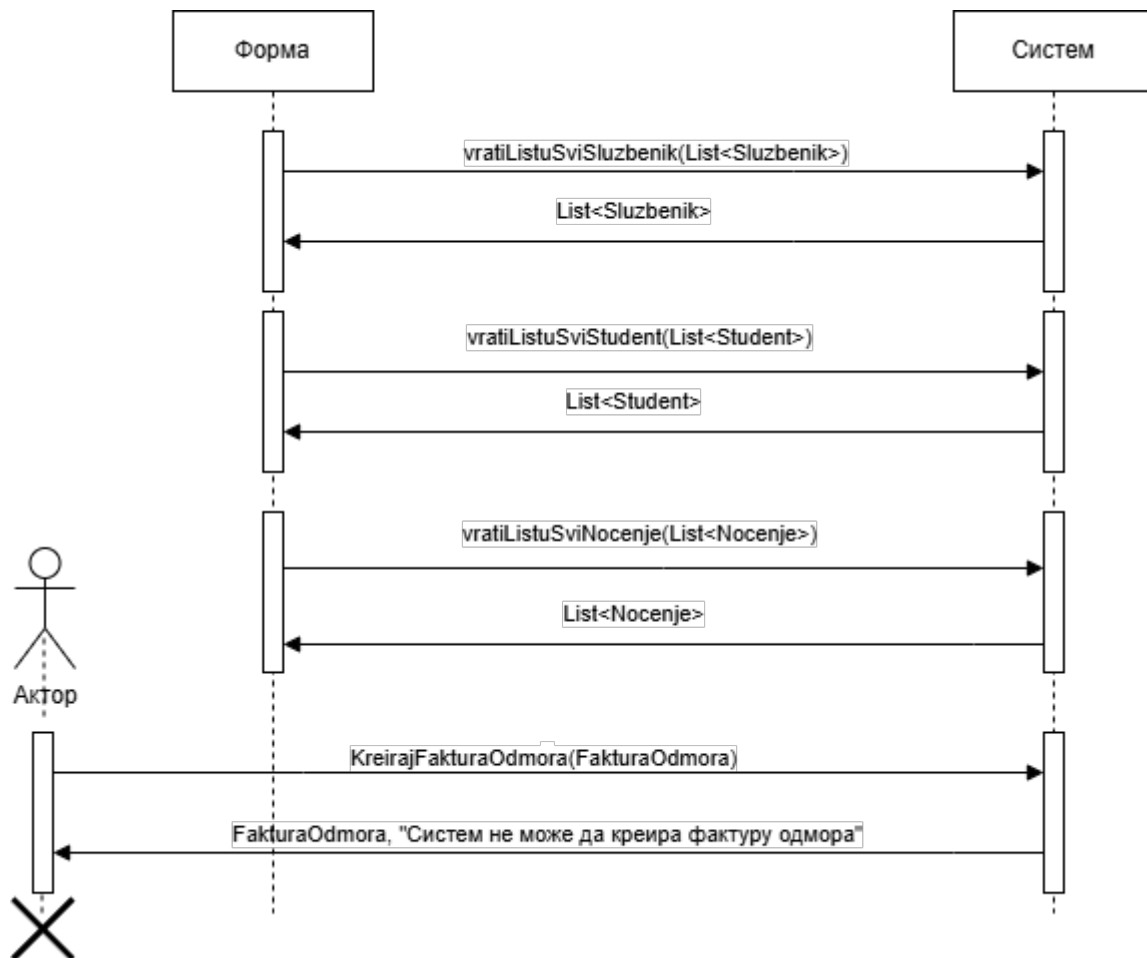
Основни сценарио СК:

7. **Службеник** позива **систем** да **креира** фактуру одмора. (АПСО)
8. **Систем** приказује **службенику** фактуру одмора и поруку: “**Систем** је креирао фактуру одмора”. (ИА)
9. **Службеник** позива **систем** да **запамти** податке о фактури одмора. (АПСО)
10. **Систем** приказује **службенику** фактуру одмора и поруку: “**Систем** је запамтио фактуру одмора.” (ИА)

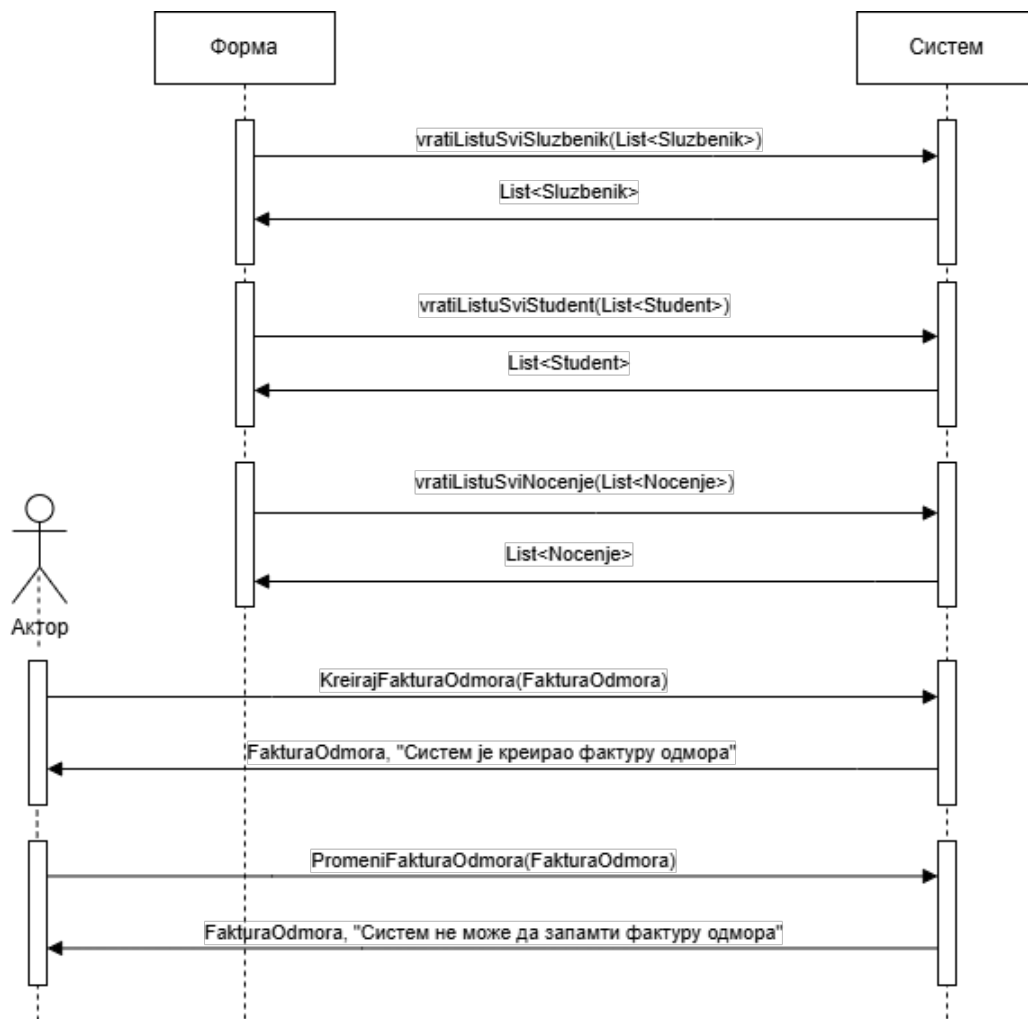


Алтернативна сценарија:

7.1 Уколико **систем** не може да креира **фактуру одмора** он **приказује** **службенику** поруку: "Систем не може да креира **фактуру одмора**". Прекида се извршење сценарија. (ИА)



9.1 Уколико **систем** не може да запамти податке о **фактури одмора** он **приказује** **службенику** поруку: “**Систем** не може да запамти **фактуру одмора**”. (ИА)



Са наведених секвенцих дијаграма уочавају се 5 системских операција које треба пројектовати:

signal KreirajFakturaOdmora(FakturaOdmora)
signal PromeniFakturaOdmora(FakturaOdmora)
signal vratiListuSviSluzbenik(Lista<Sluzbenik>)
signal vratiListuSviStudent(Lista<Student>)
signal vratiListuSviNocenje(Lista<Nocenje>)

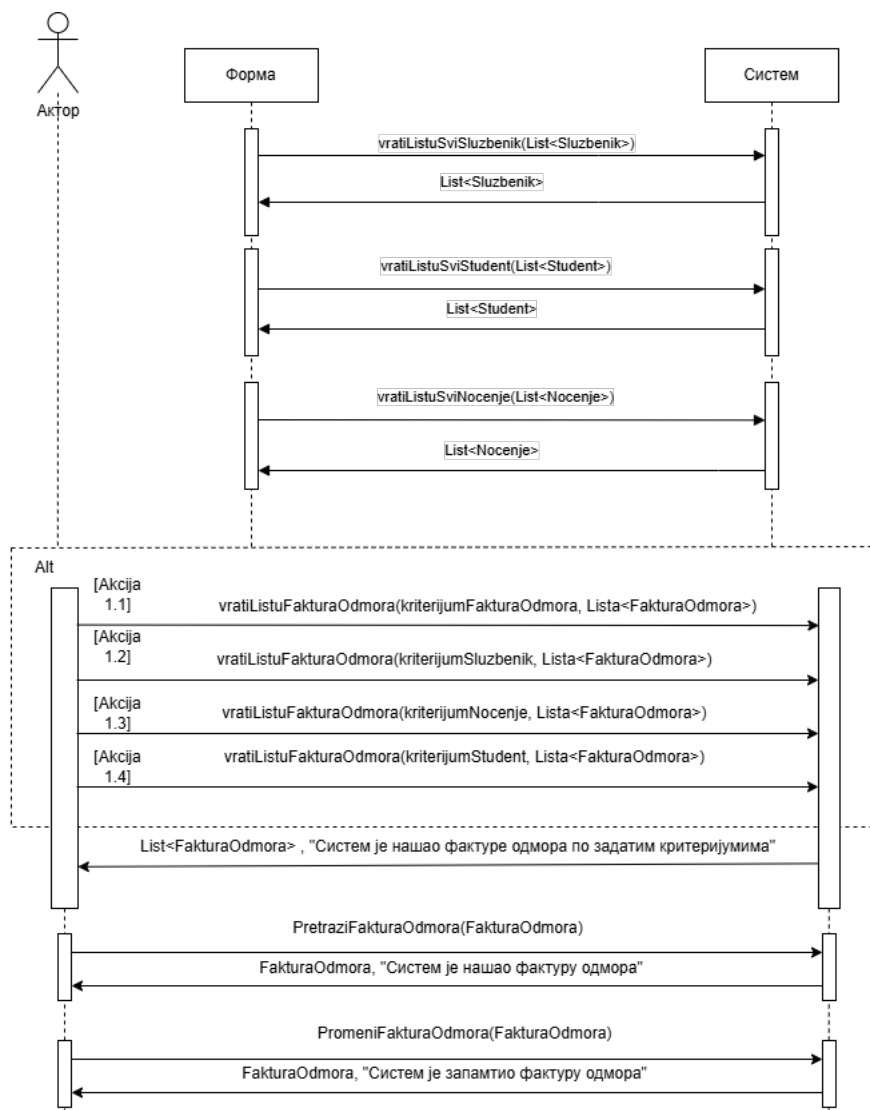
ДС2: Дијаграми секвенци случаја коришћења – Промени фактура одмора

Предуслови:

1. **Форма** позива систем да **врати** листу свих службеника. (АПСО)
2. **Систем** враћа **форми** листу свих службеника. (ИА)
3. **Форма** позива систем да **врати** листу свих студената. (АПСО)
4. **Систем** враћа **форми** листу свих студената. (ИА)
5. **Форма** позива систем да **врати** листу свих ноћења. (АПСО)
6. **Систем** враћа **форми** листу свих ноћења. (ИА)

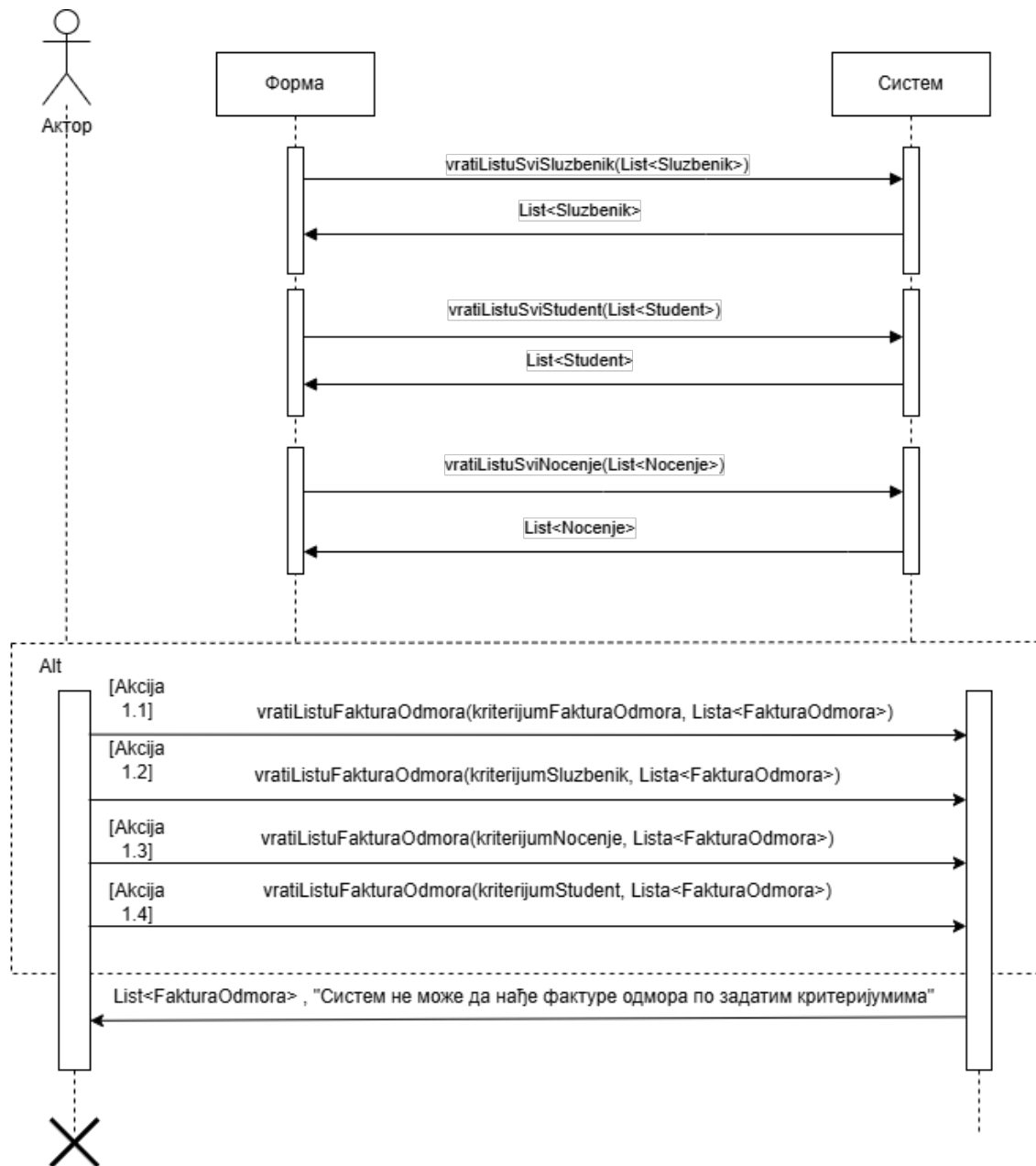
Основни сценарио СК:

7. **Службеник** позива **систем** да нађе фактуре одмора по задатим критеријумима. (АПСО)
8. **Систем** приказује **службенику** фактуре одмора и поруку: “Систем је нашао фактуре одмора по задатим критеријумима”. (ИА)
9. **Службеник** позива **систем** да нађе фактуру одмора. (АПСО)
10. **Систем** приказује **службенику** фактуру одмора и поруку: “Систем је нашао фактуру одмора”. (ИА)
11. **Службеник** позива **систем** да запамти податке о фактури одмора. (АПСО)
12. **Систем** приказује **службенику** фактуру одмора и поруку: “Систем је запамтио фактуру одмора.” (ИА)

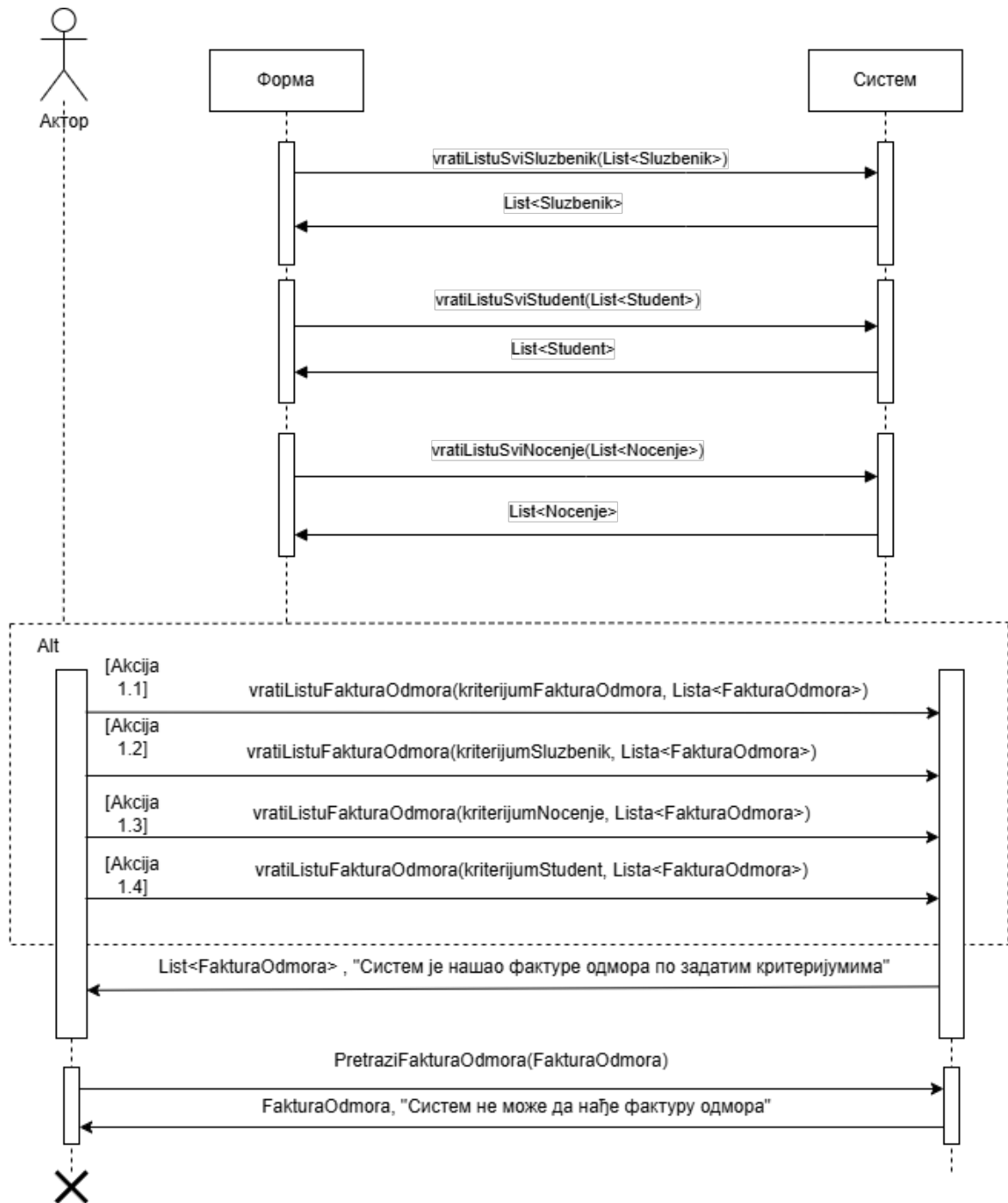


Алтернативна сценарија:

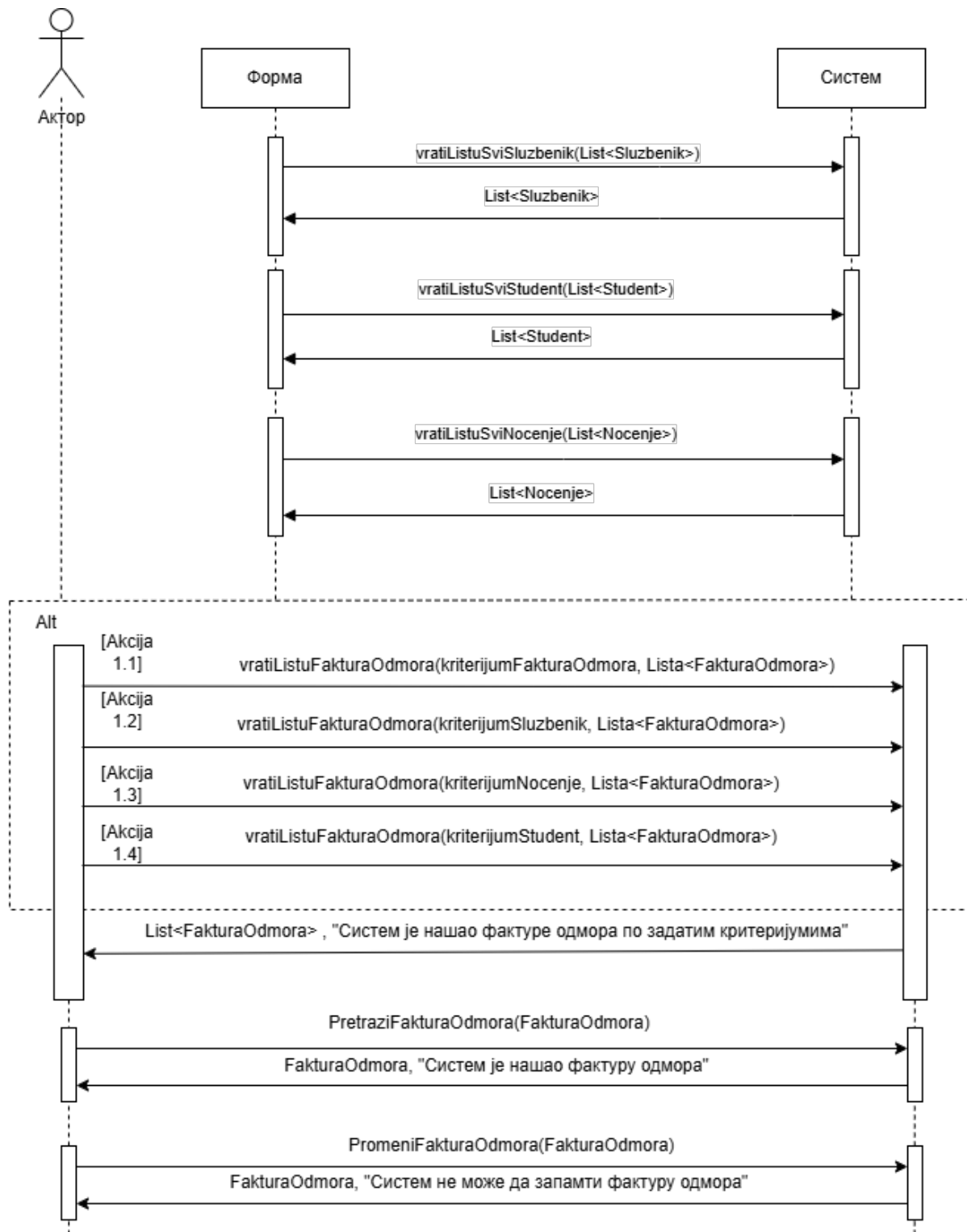
8.1 Уколико **систем** не може да нађе **фактуре одмора** он **приказује службенику** поруку: “**Систем** не може да нађе **фактуре одмора** по задатом критеријуму”. Прекида се извршење сценарија. (ИА)



10.1 Уколико **систем** не може да нађе **фактуру одмора** он **приказује службенику** поруку:
 “Систем не може да нађе **фактуру одмора**”. Прекида се извршење сценарија. (ИА)



12.1 Уколико **систем** не може да запамти податке о **фактури одмора** он **приказује** **службенику** поруку: “**Систем** не може да запамти **фактуру одмора**”. (ИА)



Са наведених секвенцих дијаграма уочавају се 9 системских операција које треба пројектовати:

1.	signal PromeniFakturaOdmora(FakturaOdmora)
2.	signal PretraziFakturaOdmora(FakturaOdmora)
3.	signal vratiListuSviSluzbenik(Lista<Sluzbenik>)
4.	signal vratiListuSviStudent(Lista<Student>)
5.	signal vratiListuSviNocenje(Lista<Nocenje>)
6.	signal vratiListuFakturaOdmora(kriterijumFakturaOdmora, Lista<FakturaOdmora>)
7.	signal vratiListuFakturaOdmora(kriterijumNocenje, Lista<FakturaOdmora>)
8.	signal vratiListuFakturaOdmora(kriterijumSluzbenik, Lista<FakturaOdmora>)
9.	signal vratiListuFakturaOdmora(kriterijumStudent, Lista<FakturaOdmora>)

3.3 Понашање софтверског система - Дефинисање уговора о системским операцијама

За системске операције се праве уговори. Овде ћемо навести осам различитих (типских) уговора за системске операције.

1. Уговор UG1: PrijaviSluzbenik

Операција: `PrijaviSluzbenik (KorisnickoIme, Sifra):signal;`

Веза са СК: СК8

Предуслови:

Постуслови: *Службеник је пријављен на систем.*

2. Уговор UG2: KreirajStudent

Операција: `KreirajStudent(Student):signal;`

Веза са СК: СК1

Предуслови: *Структурна и вредносна ограничење над објектом класе Студент морају бити задовољена.*

Постуслови: *Направљен је нови објекат класе Студент.*

3. Уговор UG3: UbaciSmena

Операција: `KreirajSmena(Smena):signal;`

Веза са СК: СК21

Предуслови: *Структурна и вредносна ограничење над објектом класе Смена морају бити задовољена.*

Постуслови: *Направљен је нови објекат класе Смена.*

4. Уговор UG4: PromeniStudent

Операција: `PromeniStudent(Student):signal;`

Веза са СК: СК4, СК6

Предуслови: *Структурна и вредносна ограничење над објектом класе Студент морају бити задовољена*

Постуслови: *Објекат класе Студент је промењен.*

5. Уговор UG5: ObrisiStudent

Операција: `ObrisiStudent(Student):signal;`

Веза са СК: СК7

Предуслови: *Структурна и вредносна ограничење над објектом класе Студент морају бити задовољена.*

Постуслови: *Објекат класе Студент је обрисан.*

6. Уговор UG6: PretraziSluzbenik

Операција: PretraziSluzbenik(*Sluzbenik*):signal;

Веза са СК: СК10, СК11

Предуслови:

Постуслови: *Пронађен је тражени објекат класе Службеник.*

7. Уговор UG7: vratiListuFakturaNocenja

Операција: vratiListuFakturaNocenja(*Kriterijum* ,*Lista*<*FakturaNocenja*>):signal;

Веза са СК: СК2

Предуслови:

Постуслови: *Пронађена је листа тражених објеката класе ФактураНоћења.*

8. Уговор UG8: vratiListuSviStudent

Операција: vratiListuSviX (*Kriterijum*,*Lista*<*Student*>):signal;

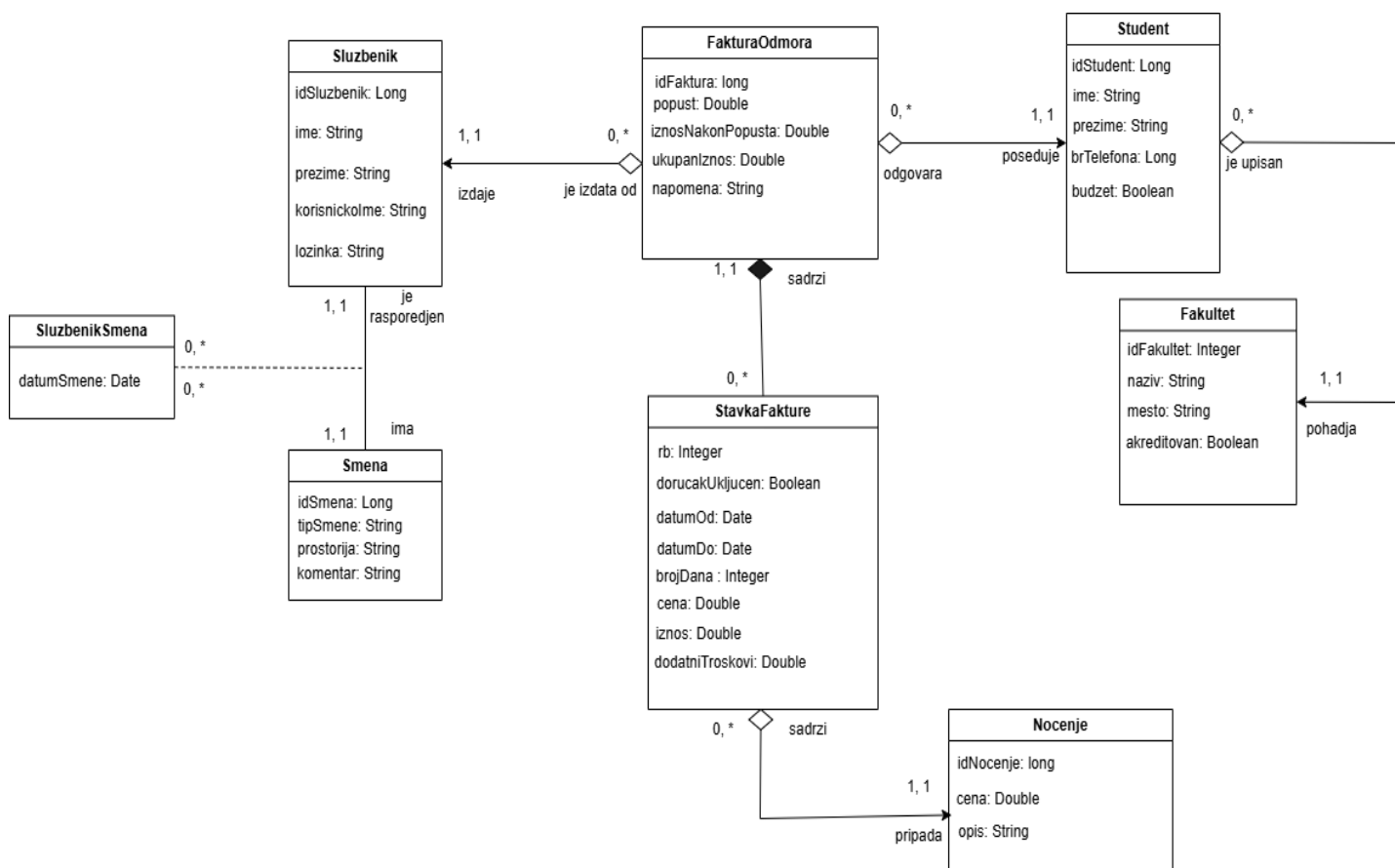
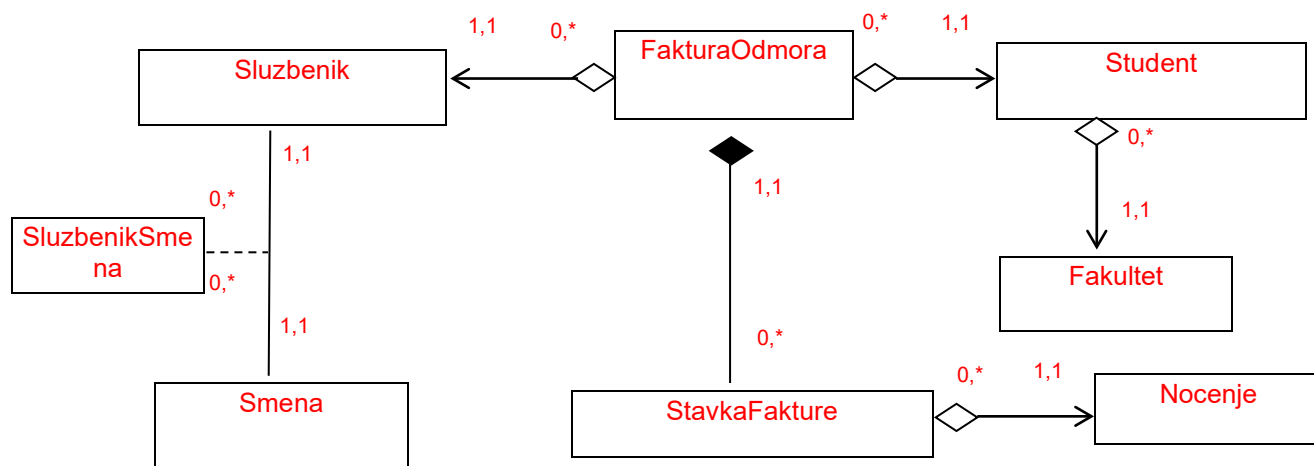
Веза са СК: СК1, СК3

Предуслови:

Постуслови: *Пронађена је листа свих објеката класе Студент.*

3.4 Структура софтверског система – Концептуални (доменски) модел

Тип концептуалног модела 1:



3.5 Структура софтверског система – Релациони модел

На основу концептуалног модела се прави релациони модел .

Релациони модел добијен из **првог** типа концептуалног модела:

1. **Sluzbenik** (idSluzbenik, ime, prezime, korisnickoIme, lozinka)
2. **Nocenje** (idNocenje, cena, opis)
3. **Fakultet** (idFakultet, naziv, mesto, akreditovan)
4. **Smena** (idSmena, prostorija, komentar, tipSmene)
5. **Student** (idStudent, ime, prezime, brTelefona, budzet, *idFakultet*)
6. **FakturaOdmora** (idFakturaOdmora, popust, iznosNakonPopusta, ukupanIznos, napomena *idSluzbenik*, *idStudent*)
7. **StavkaFakture** (*idFakturaOdmora*, rb, dorucakUkljucen, datumOd, datumDo, brojDana, cena, iznos, dodatniTroskovi, *idNocenje*)
8. **SluzbenikSmena** (idSluzbenik, idSmena, datumSmene,)

3.6 Табела структурних и вредносних ограничења релационог модела

За сваку релацију се прави табела структурних и вредносних ограничења.

Табела структурних и вредносних ограничења релационог модела који је добијен из **првог** типа концептуалног модела:

1. Табела Sluzbenik		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Назив	Тип атрибута	Вредност атрибута	Међузав. атрибута једне табеле	Међузав. атрибута више табела	INSERT / UPDATE CASCADES FakturaNocenja , SluzbenikSmena DELETE RESTRICTED FakturaNocenja , SluzbenikSmena
	idSluzbenik	Long	Not null and >0			
	ime	String	Not null			
	prezime	String	Not null			
	korisnickoIme	String	Not null			
	lozinka	String	Not null			

2. Табела TipNocenja		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Назив	Тип атрибута	Вредност атрибута	Међузав. атрибута једне табеле	Међузав. атрибута више табела	INSERT / UPDATE CASCADES StavkaFakture , DELETE RESTRICTED StavkaFakture ,
	idNocenja	Long	Not null and >0			
	cena	Double	Not null and >0			
	opis	Integer	Not null			

3. Табела Fakultet		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Назив	Тип атрибута	Вредност атрибута	Међузав. атрибута једне табеле	Међузав. атрибута више табела	INSERT / UPDATE CASCADES Student , DELETE RESTRICTED Student ,
	idFakultet	Integer	Not null and >0			
	naziv	String	Not null			
	mesto	String	Not null			
	akreditovan	Boolean	True or false			
		Date				

4. Табела Smena		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Назив	Тип атрибута	Вредност атрибута	Међузав. атрибута једне табеле	Међузав. атрибута више табела	INSERT / UPDATE CASCADES SluzbenikSmena , DELETE RESTRICTED SluzbenikSmena ,
	idSmena	Long	Not null and >0			
	prostorija	String	Not null			
	komentar	String	Not null			
	tipSmene	String	Not null			

5. Табела Student		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Назив	Тип атрибута	Вредност атрибута	Међузав. атрибута једне табеле	Међузав. атрибута више табела	INSERT RESTRICTED idStatus UPDATE RESTRICTED idStatus UPDATE CASCADES FakturaNocenja DELETE RESTRICTED FakturaNocenja
	idStudent	Long	Not null and >0			
	ime	String	Not null			
	prezime	String	Not null			
	brTelefona	Long	Not null and >0			
	budzet	Boolean	True or false			

6. Табела FakturaOdmora		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Назив	Тип атрибута	Вредност атрибута	Међузав. атрибута једне табеле	Међузав. атрибута више табела	INSERT RESTRICTED Student Sluzbenik UPDATE RESTRICTED Student Sluzbenik UPDATE CASCADES StavkaRezervacije DELETE RESTRICTED StavkaRezervacije
	idFaktura	Long	Not null and >0			
	popust	Double	0.3 or 1		If (budzet = true) then popust = 0.3 else popust = 1	
	cenaNakonPopusta	Double	Not null and >0		ukupanIznos * popust	
	ukupanIznos	Double	Not null and >0		SUM(StavkeFakture.iznos)	
	napomena	String				

7.Табела StavkaFakture		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Назив	Тип атрибута	Вредност атрибута	Међузав. атрибута једне табеле	Међузав.атрибута више табела	INSERT RESTRICTED FakturaOdmora TipNocenja UPDATE RESTRICTED FakturaOdmora TipNocenja DELETE /
	<u>idFakture</u>	Long	Not null and >0			
	<u>rb</u>	Integer	Not null and >0			
	dorucakUkljucen	Boolean		If (dorucakUkljucen = true) then cena = cena +		
	datumOd	Date	Not null	<datumDo		
	datumDo	Date	Not null	>datumDo		
	brojDana	Integer	Not null and >0	datumDo-datumOd		
	cena	Double	Not null and >0			
	iznos	Double	Not null and >0	brojDana * cena + dodatniTroskovi		
	dodatniTroskovi	Double	Not null	If (dorucakUkljucen = true) then dodatniTroskovi= brojDana*200		

8.Табела SluzbenikSmena		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Назив	Тип атрибута	Вредност атрибута	Међузав. атрибута једне табеле	Међузав.атрибута више табела	INSERT RESTRICTED Sluzbenik Smena UPDATE RESTRICTED Sluzbenik Smena DELETE /
	<u>idSluzbenik</u>	Long	Not null and >0			
	<u>idSmena</u>	Long	Not null and >0			
	<u>datumSmene</u>	Date	Not null			

4. Пројектовање

4.1 Пројектовање корисничког интерфејса

4.2 Пројектовање апликационе логике

4.3 Пројектовање складишта података

5. Имплементација

6. Тестирање

Напомена за студенте: Треба неколико системских операција тестирати ручно или аутоматизовано (нпр . коришћењен JUnit оквира). Посебно ћемо ценити ако то урадите аутоматизовано.

7. Закључак

Напомена за студенте: У закључку се наводи шта је резултат семинарског рада и ваш субјективни осврт на изазове које сте имали током израде овог семинарског рада.

Литература