

Универзитет у Београду
Факултет организационих наука



Пројектовање софтвера

Систем за управљање библиотеком

Ментор:
др Илија Антовић

Студент:
Дејан Чолић 260/13

Београд. 2025

Садржај

Садржај.....	2
1. Увод.....	5
2. Методологија развоја.....	5
3. Прикупљање корисничких захтева.....	6
3.1 Вербални опис.....	6
3.2. Листа случајева коришћења.....	7
3.3. Модел случајева коришћења.....	8
3.4 Случајеви коришћења.....	9
1. СК1: Случај коришћења – Унос нове књиге (сложен).....	9
2. СК2: Случај коришћења – Унос новог задужења (сложен).....	10
3. СК3: Случај коришћења – Унос новог члана.....	11
4. СК4: Случај коришћења – Претраживање чланова.....	12
5. СК5: Случај коришћења – Претраживање књиге.....	13
6. СК6: Случај коришћења – Претраживање задужења.....	14
7. СК7: Случај коришћења – Измена члана.....	15
8. СК8: Случај коришћења – Пријављивање радника.....	17
4. Анализа.....	18
4.1. Понашање система – Дијаграм секвенци.....	18
ДС1: Дијаграм секвенци случаја коришћења – Унос нове књиге.....	18
ДС2: Дијаграм секвенци случаја коришћења – Унос новог задужења.....	22
ДС3: Дијаграм секвенци случаја коришћења – Унос новог члана.....	26
ДС4: Дијаграм секвенци случаја коришћења – претраживање члана.....	29
ДС5: Дијаграм секвенци случаја коришћења – претраживање књиге.....	32
ДС6: Дијаграм секвенци случаја коришћења – претраживање задужења.....	35
ДС7: Дијаграм секвенци случаја коришћења – измена члана.....	38
ДС8: Дијаграм секвенци случаја коришћења – пријављивање радника.....	41
4.2. Понашање система – Дефинисање уговора о системским операцијама.....	43

Уговор УГ1	43
Уговор УГ2	43
Уговор УГ3	44
Уговор УГ4	44
Уговор УГ5	44
Уговор УГ6	44
Уговор УГ7	44
Уговор УГ8	45
Уговор УГ9	45
Уговор УГ10	45
Уговор УГ11	45
Уговор УГ12	45
Уговор УГ13	46
Уговор УГ14	46
Уговор УГ15	46
Уговор УГ16	46
4.3. Структура Система – Концептуални (доменски модел)	47
4.4. Структура система – релациони модел	48
5. Пројектовање	52
5.1 Архитектура софтверског система	52
5.2 Пројектовање корисничког интерфејса	53
5.3 Пројектовање екранских форми	53
5.3.1 СК1: Случај коришћења – Унос нове књиге (сложен)	53
5.3.2 СК2: Случај коришћења – Унос новог задужења (сложен)	56
5.3.3 СК3: Случај коришћења – Унос новог члана	59
5.3.4 СК4: Случај коришћења – Претраживање чланова	61
5.3.5 СК5: Случај коришћења – Претраживање књига	63
5.3.6 СК6: Случај коришћења – Претраживање задужења	67
5.3.7 СК7: Случај коришћења – Измена члана	69
5.3.8 СК8: Случај коришћења – Пријављивање радника	72
5.4 Пројектовање контролера корисничког интерфејса	74

5.5 Пројектовање апликационе логике	74
5.5.1. Комуникација са клијентима	74
5.5.2. Контролер апликационе логике	75
5.5.3 Пословна логика	75
Уговор УГ1	75
Уговор УГ2	76
Уговор УГ3	76
Уговор УГ4	77
Уговор УГ5	78
Уговор УГ6	78
Уговор УГ7	79
Уговор УГ8	80
Уговор УГ9	81
Уговор УГ10	81
Уговор УГ11	82
Уговор УГ12	83
Уговор УГ13	83
Уговор УГ14	84
Уговор УГ15	84
Уговор УГ16	85
5.5.4. Пројектовање структуре софтверског система	86
5.5.5. Брокер базе података	86
5.6 Пројектовање складишта података	89
6. Имплементација	90
7. Тестирање	94
8. Литература	95

1. Увод

У овом раду разматра се процес развоја софтверског система за управљање библиотеком. Као методолошки оквир коришћена је упрошћена Ларманова метода, која омогућава систематичан приступ анализи и дизајну. Рад обухвата фазе од прикупљања захтева, преко моделовања, до саме имплементације у програмском језику Јава, при чему је посебан нагласак стављен на практичну примену објектно оријентисаних принципа.

2. Методологија развоја

Приликом развоја овог софтвера коришћена је упрошћена Ларманова метода.

Процес се састоји из пет делова:

1. Прикупљање захтева
2. Анализа
3. Пројектовање
4. Имплементација
5. Тестирање

У првој фази прикупљамо захтеве од корисника. Ту дефинишемо својства и услове које софтвер треба да задовољи. Те захтеве формално описујемо преко УМЛ дијаграма случајева коришћења (UML Use-Case diagram).

У другој фази се дефинише пословна логика која се састоји од структуре система и од понашања система. Структуру представљамо преко концептуалног и релационог модела, док понашање представљамо преко УМЛ дијаграма секвенци (UML Sequence diagram).

У трећој фази се описује архитектура система која се састоји из корисничког интерфејса, апликационе логике, и складишта података.

У четвртој фази имплементирамо досадашњи модел у некој конкретној технологији (у овом случају у програмском језику JAVA).

У петој и финалној фази свака од компоненти се тестира.

3. Прикупљање корисничких захтева

3.1 Вербални опис

Потребно је направити софтверски систем за управљање библиотеком.

Библиотека има скуп књига, свака књига има своје примерке, библиотека има чланове и има раднике који обављају административне послове.

Радник мора да приступи систему путем пријаве на систем, а затим може да уноси, претражује и мења књиге, чланове, и задужења.

Задужење настаје када члан задужи барем један примерак неке књиге. Члан може задужити и по један примерак више различитих књига, а може и задужити више примерака исте књиге.

Члан мора да врати књигу библиотеци, али не мора све књиге да врати у исто време.

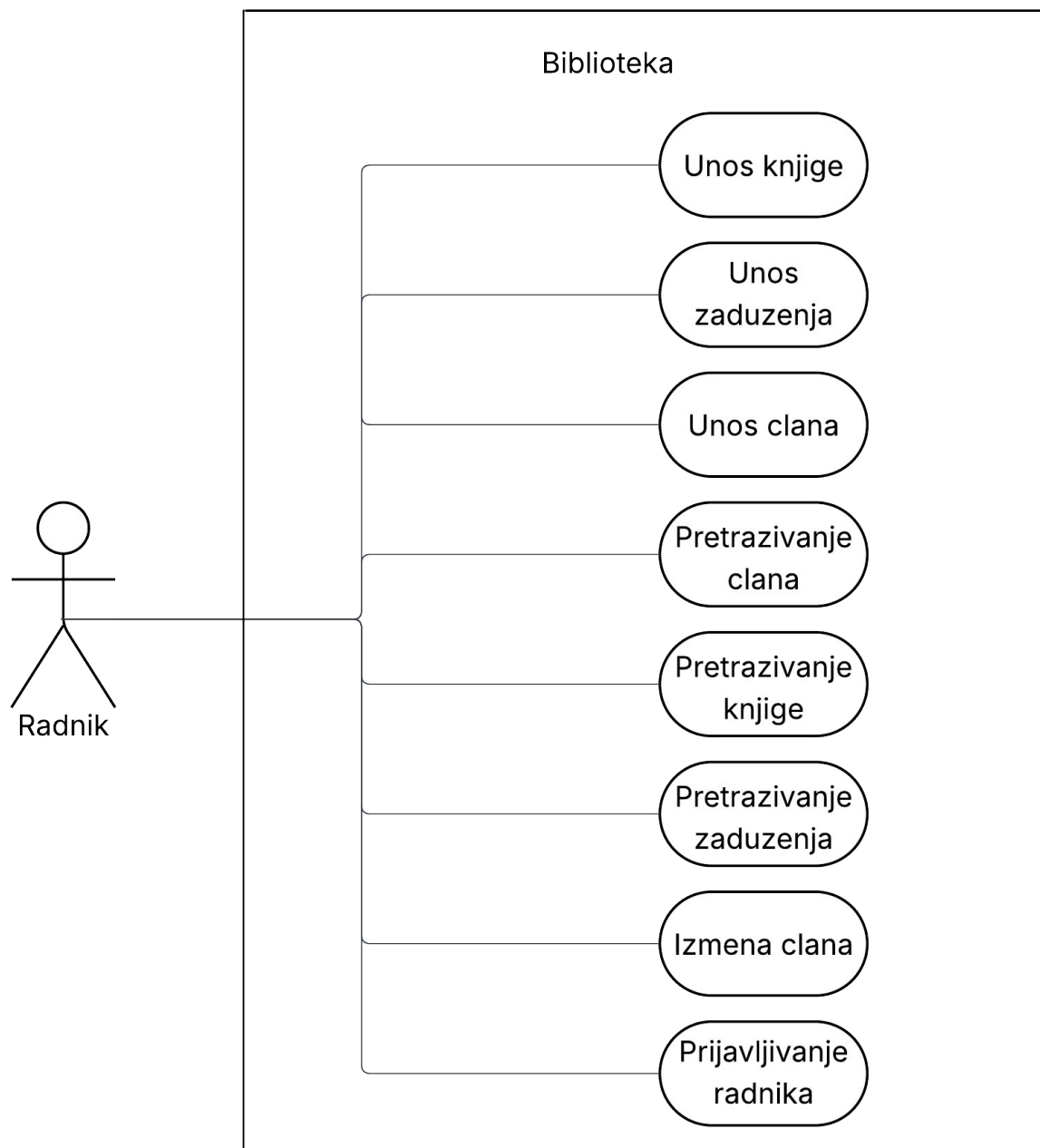
Сваки примерак књиге коју је члан задужио представља једну ставку задужења.

Ставке задужења и примерци књига се не могу директно креирати и мењати, већ се то ради у оквиру креирања, односно измене задужења или књиге за коју су везани.

3.2. Листа случајева коришћења

1. Унос нове књиге (Сложен)
2. Унос новог задужења (Сложен)
3. Унос новог члана
4. Претраживање члана
5. Претраживање књига
6. Претраживање задужења
7. Измена члана
8. Пријављивање радника

3.3. Модел случајева коришћења



Слика 1: Дијаграм случајева коришћења

3.4 Случајеви коришћења

1. СК1: Случај коришћења –Унос нове књиге (сложен)

Назив СК

Унос нове књиге

Актори СК

Радник

Учесници СК

Радник и систем (програм)

Предуслов: Систем је укључен и радник је улогован. Систем је омогућио рад са књигом.

Основни сценарио СК

1. Радник позива систем да креира нову књигу. (АПСО)
2. Систем креира књигу. (СО)
3. Систем приказује раднику књигу и поруку: “Систем је креирао књигу“. (ИА)
4. Радник уноси податке о књизи. (АПУСО)
5. Радник контролише да ли је тачно унео податке о књизи. (АНСО)
6. Радник позива систем да запамти податке о књизи. (АПСО)
7. Систем памти податке о књизи. (СО)
8. Систем приказује раднику запамћену књигу и поруку: “Систем је запамтио књигу“. (ИА)

Алтернативна сценарија

3.1 Уколико систем не може да креира књигу он приказује раднику поруку: “Систем не може да креира књигу”. Прекида се извршење сценарија. (ИА)

8.1 Уколико систем не може да запамти податке о књизи он приказује раднику поруку “Систем не може да запамти књигу”. (ИА)

2. СК2: Случај коришћења – Унос новог задужења (сложен)

Назив СК

Унос новог задужења

Актори СК

Радник

Учесници СК

Радник и систем (програм)

Предуслов: Систем је укључен и радник је улогован. Систем је омогућио рад са задужењима, учитана је листа књига, листа примерака, листа чланова.

Основни сценарио СК

1. Радник позива систем да креира ново задужење. (АПСО)
2. Систем креира ново задужење. (СО)
3. Систем приказује раднику ново задужење и поруку: "Успешно креирано ново задужење". (ИА)
4. Радник уноси податке о задужењу. (АПУСО)
5. Радник позива систем да запамти податке о задужењу. (АПСО)
6. Систем памти податке о задужењу. (СО)
7. Систем приказује раднику ново задужење и поруку: "Успешно запамћено ново задужење". (ИА)

Алтернативна сценарија

3.1 Уколико систем не може да креира задужење, приказује раднику поруку: "Систем не може да креира задужење". Прекида се извршење сценарија. (ИА)

7.1 Уколико систем не може да запамти задужење, приказује раднику поруку: "Систем не може да запамти задужење". (ИА)

3. СКЗ: Случај коришћења – Унос новог члана

Назив СК

Унос новог члана

Актори СК

Радник

Учесници СК

Радник и систем (програм)

Предуслов: Систем је укључен и радник је улогован. Систем је омогућио рад са члановима.

Основни сценарио СК

1. Радник позива систем да креира члана. (АПСО)
2. Систем креира члана. (СО)
3. Систем приказује раднику члана и поруку: “Систем је креирао члана“. (ИА)
4. Радник уноси податке о члану. (АПУСО)
5. Радник контролише да ли је тачно унео податке о члану. (АНСО)
6. Радник позива систем да запамти податке о члану. (АПСО)
7. Систем памти податке о члану. (СО)
8. Систем приказује раднику запамћеног члана и поруку: “Систем је запамтио члана“. (ИА)

Алтернативна сценарија

3.1 Уколико систем не може да креира члана он приказује раднику поруку: “Систем не може да креира члана”. Прекида се извршење сценарија. (ИА)

8.1 Уколико систем не може да запамти податке о члану он приказује раднику поруку “Систем не може да запамти члана”. (ИА)

4. СК4: Случај коришћења – Претраживање чланова

Назив СК

Претраживање чланова

Актори СК

Радник

Учесници СК

Радник и систем (програм)

Предуслов: Систем је укључен и радник је улогован. Систем је омогућио рад са члановима.

Основни сценарио СК

1. **Радник** уноси вредности по којима претражује чланове. (АПУСО)
2. **Радник** позива **систем** да пронађе чланове по задатим вредностима. (АПСО)
3. **Систем** тражи чланове по задатим критеријумима. (СО)
4. **Систем** приказује листу чланова и поруку "Пронађени су одговарајући чланови". (ИА)
5. **Радник** бира члана. (АПУСО)
6. **Радник** позива **систем** да учита податке о изабраном члану. (АПСО)
7. **Систем** учитава податке о изабраном члану. (СО)
8. **Систем** приказује податке о члану уз поруку "Успешно учитан члан". (ИА)

Алтернативна сценарија

4.1 Уколико **систем** не може да нађе чланове он приказује **раднику** поруку: "Систем не може да нађе чланове по задатим вредностима". Прекида се извршење сценарија. (ИА)

8.1 Уколико **систем** не може да учита податке о члану он приказује **раднику** поруку: "Систем не може да учита изабраног члана". (ИА)

5. СК5: Случај коришћења – Претраживање књига

Назив СК

Претраживање књига

Актори СК

Радник

Учесници СК

Радник и систем (програм)

Предуслов: Систем је укључен и радник је улогован. Систем је омогућио рад са књигама.

Основни сценарио СК

1. Радник уноси вредности по којима претражује књиге. (АПУСО)
2. Радник позива систем да пронађе књиге по задатим вредностима. (АПСО)
3. Систем тражи књиге по задатим критеријумима. (СО)
4. Систем приказује листу књига и поруку "Пронађене су одговарајуће књиге". (ИА)
5. Радник бира књигу. (АПУСО)
6. Радник позива систем да учита податке о изабраној књизи. (АПСО)
7. Систем учитава податке о изабраној књизи. (СО)
8. Систем приказује податке о књизи уз поруку "Успешно учитана књига". (ИА)

Алтернативна сценарија

- 4.1 Уколико систем не може да нађе књиге он приказује раднику поруку: "Систем не може да нађе књиге по задатим вредностима". Прекида се извршење сценарија. (ИА)
- 8.1 Уколико систем не може да учита податке о књизи он приказује раднику поруку: "Систем не може да учита изабрану књигу". (ИА)

6. СК6: Случај коришћења – Претраживање задужења

Назив СК

Претраживање задужења

Актори СК

Радник

Учесници СК

Радник и систем (програм)

Предуслов: Систем је укључен и радник је улогован. Систем је омогућио рад са задужењима.

Основни сценарио СК

1. **Радник** уноси вредности по којима претражује задужења. (АПУСО)
2. **Радник** позива **систем** да пронађе задужења по задатим вредностима. (АПСО)
3. **Систем** тражи задужења по задатим критеријумима. (СО)
4. **Систем** приказује листу задужења и поруку "Пронађена су одговарајућа задужења". (ИА)
5. **Радник** бира задужење. (АПУСО)
6. **Радник** позива **систем** да учита податке о изабраном задужењу. (АПСО)
7. **Систем** учитава податке о изабраном задужењу. (СО)
8. **Систем** приказује податке о задужењу уз поруку "Успешно учитано задужење". (ИА)

Алтернативна сценарија

- 4.1 Уколико **систем** не може да нађе задужења он приказује **раднику** поруку: "Систем не може да нађе задужења по задатим вредностима". Прекида се извршење сценарија. (ИА)
- 8.1 Уколико **систем** не може да учита податке о задужењу он приказује **раднику** поруку: "Систем не може да учита изабрано задужење". (ИА)

7. СК7: Случај коришћења – Измена члана

Назив СК

Измена члана

Актори СК

Радник

Учесници СК

Радник и систем (програм)

Предуслов: Систем је укључен и радник је улогован. Систем је омогућио рад са члановима. Учитана је листа чланова.

Основни сценарио СК

1. Радник уноси вредности по којима претражује чланове. (АПУСО)
2. Радник позива систем да пронађе чланове по задатим вредностима. (АПСО)
3. Систем тражи чланове по задатим критеријумима. (СО)
4. Систем приказује листу чланова и поруку "Пронађени су одговарајући чланови". (ИА)
5. Радник бира члана из листе чланова. (АПУСО)
6. Радник позива систем да учита податке о изабраном члану. (АПСО)
7. Систем учитава податке о изабраном члану. (СО)
8. Систем приказује податке о члану уз поруку "Успешно учитан члан". (ИА)
9. Радник мења податке о члану. (АПУСО)
10. Радник проверава да ли је тачно унео податке. (АНСО)
11. Радник позива систем да запамти нове податке о члану. (АПСО)
12. Систем памти измењене податке о члану. (СО)
13. Систем приказује раднику поруку: "Подаци успешно измењени". (ИА)

Алтернативна сценарија

4.1 Уколико **систем** не може да нађе члана он приказује **раднику** поруку: “Систем не може да нађе члана по задатој вредности”. Прекида се извршење сценарија. (ИА)

8.1 Уколико **систем** не може да пронађе податке о члану, **систем** приказује **раднику** поруку: " Систем не може да пронађе податке о члану". Прекида се извршење сценарија. (ИА)

13.1 Уколико **систем** не може да сачува измене, приказује **раднику** поруку: " Систем не може да измени члана". (ИА)

8. СК8: Случај коришћења – Пријављивање радника

Назив СК

Пријављивање радника

Актори СК

Радник

Учесници СК

Радник и систем (програм)

Предуслов: Систем је укључен и омогућава пријављивање радника.

Основни сценарио СК

1. Радник уноси податке за аутентификацију радника. (АПУСО)
2. Радник позива систем да пронађе радника са задатим подацима (АПСО)
3. Систем претражује раднике (СО)
4. Систем приказује поруку “Радник је успешно пријављен” и почетну форму (ИА)

Алтернативна сценарија

- 4.1 Уколико систем не може да нађе радника он приказује поруку: “Систем не може да нађе радника на основу унетих вредности”.(ИА)

4. Анализа

Након фазе прикупљања захтева од корисника приступамо фази анализе. Резултат фазе анализе је структура и понашање софтверског система.

Понашање описујемо помоћу дијаграма секвенци и уговора о системским операцијама.

Структуру описујемо помоћу концептуалног и релационог модела.

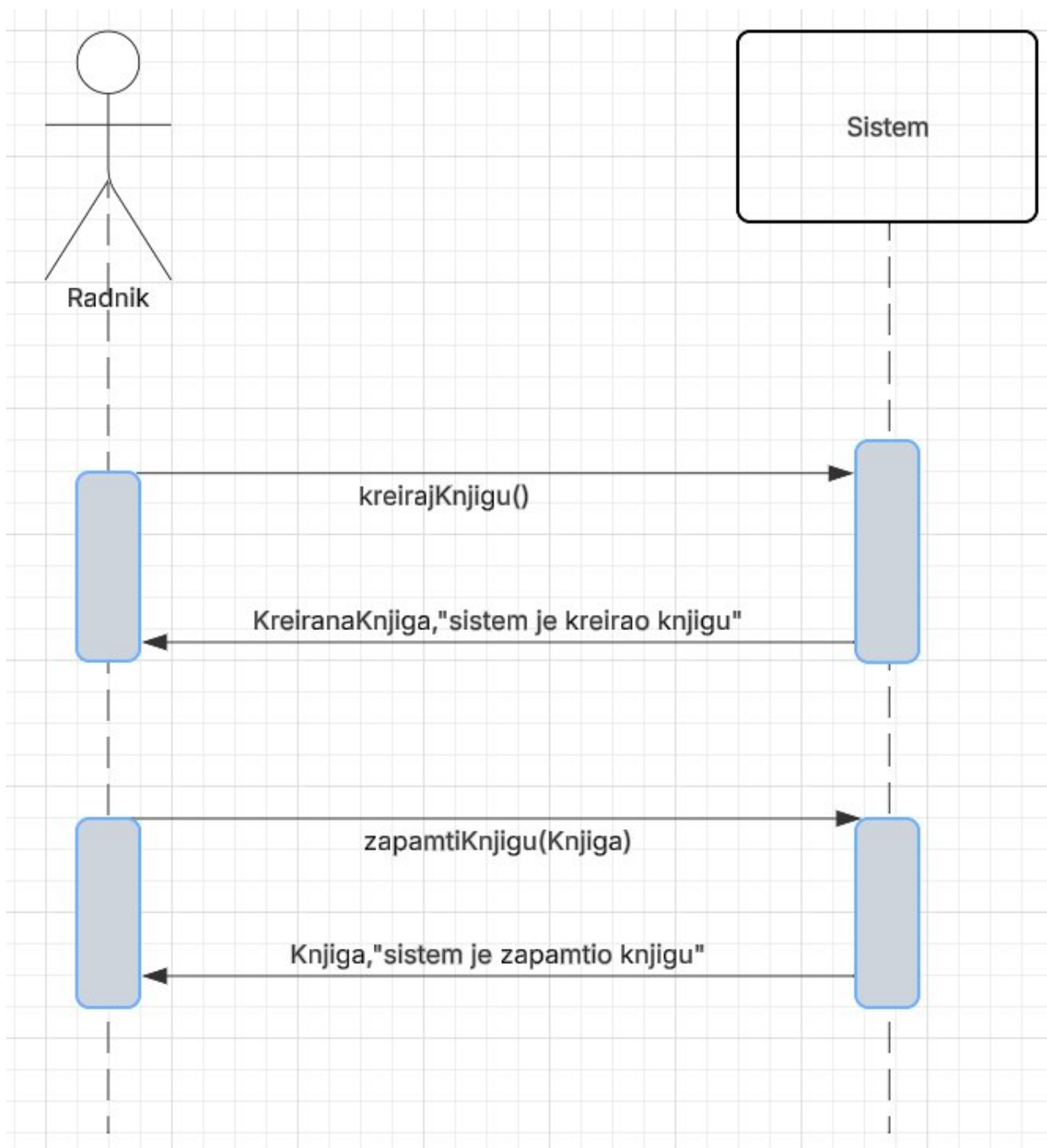
4.1. Понашање система – Дијаграм секвенци

Дијаграм секвенци је један од УМЛ дијаграма који служи за опис понашања система. Системски дијаграм секвенци приказује догађаје у одређеном редоследу, за одређени сценарио случаја коришћења, који успостављају комуникацију између актора и система.

Као резултат анализе сценарија СК добијају се захтеви за извршење системских операција. За сваку системску операцију се праве уговори(контракти).

ДС1: Дијаграм секвенци случаја коришћења – Унос нове књиге

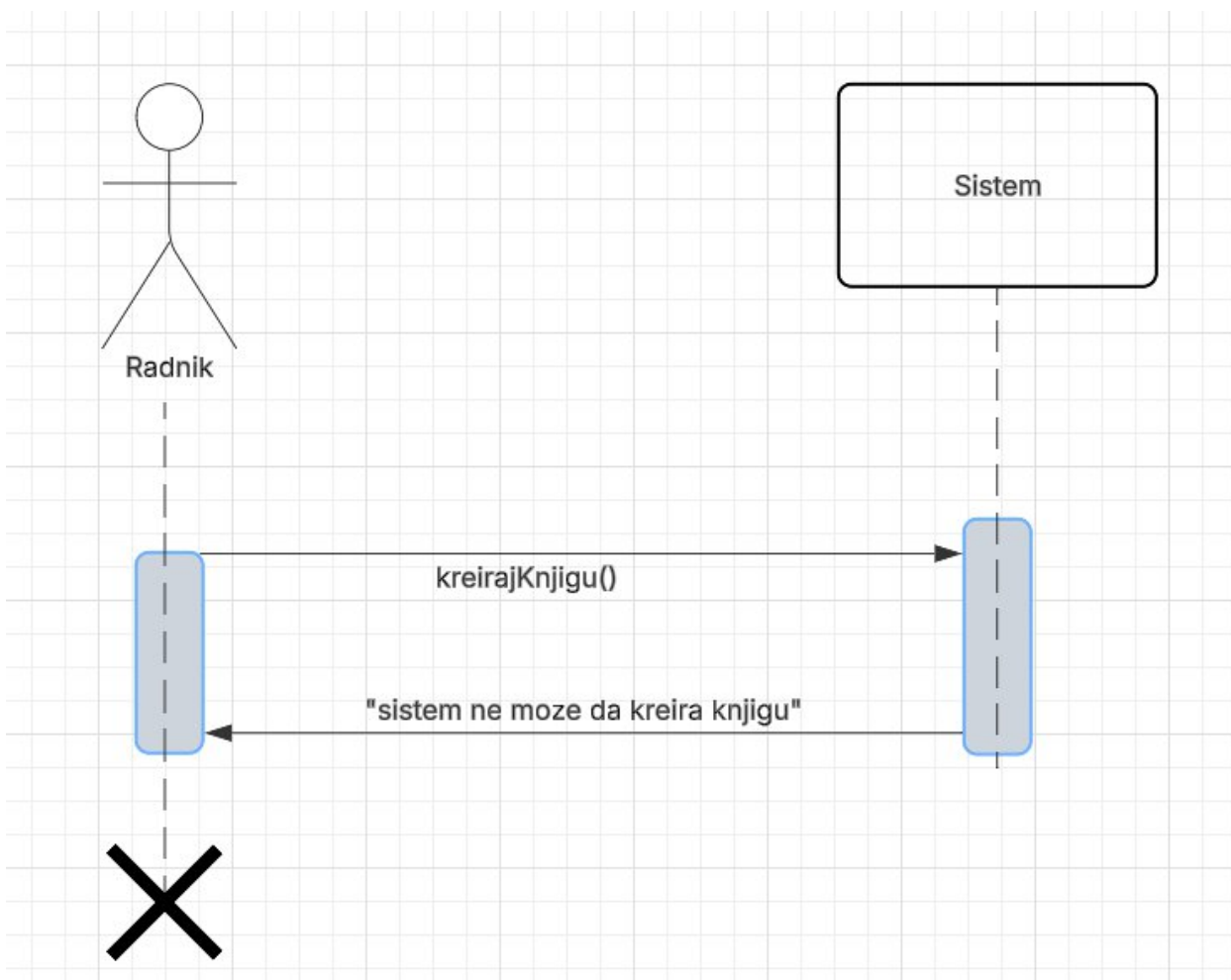
1. **Радник** позива **систем** да креира нову књигу. (АПСО)
2. **Систем** приказује **раднику** књигу и поруку: “Систем је креирао књигу“. (ИА)
3. **Радник** позива **систем** да запамти податке о књизи. (АПСО)
4. **Систем** приказује **раднику** запамћену књигу и поруку: “Систем је запамтио књигу“. (ИА)



Слика 2: СК1 – Унос нове књиге

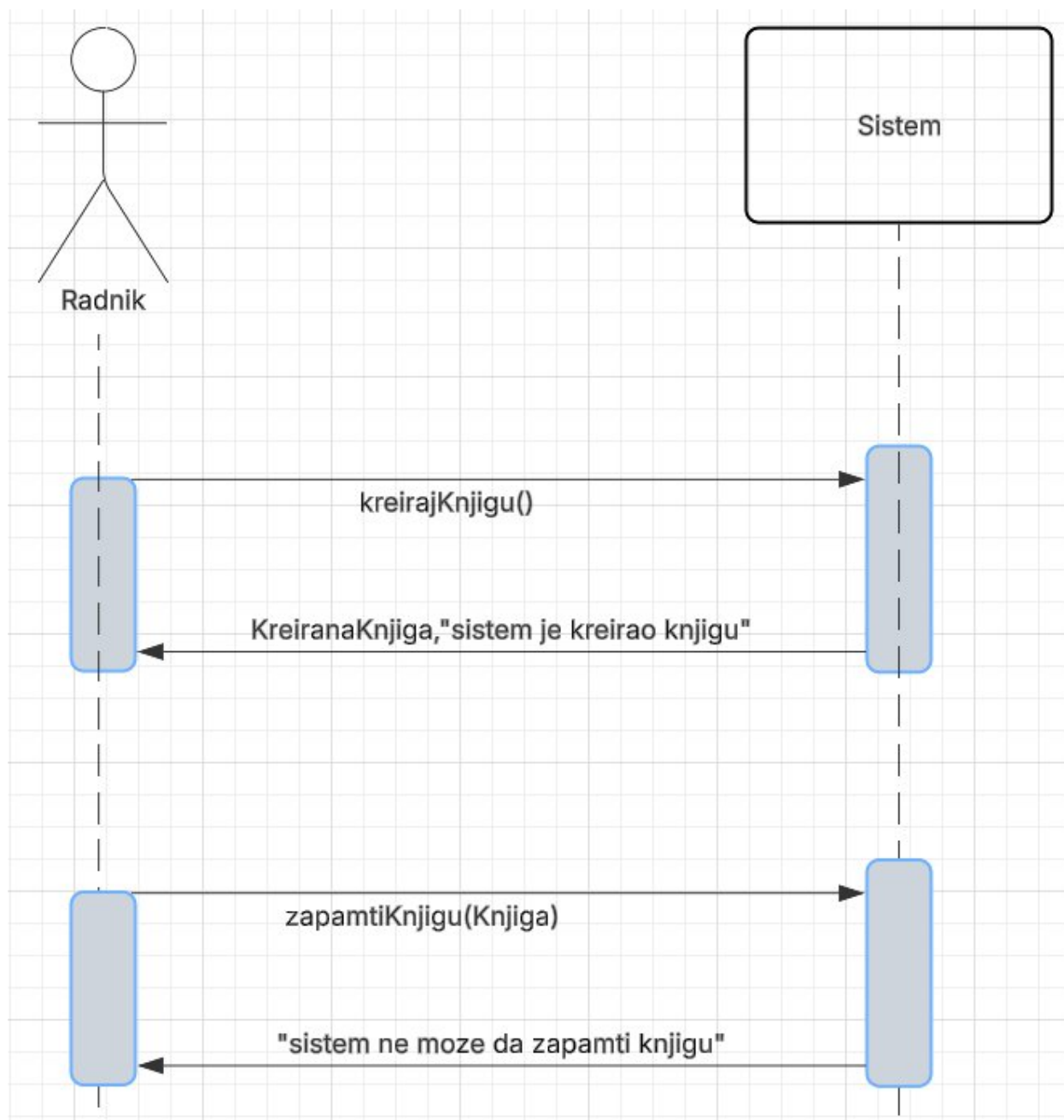
Алтернативна сценарија

2.1 Уколико **систем** не може да креира књигу он приказује **раднику** поруку: “Систем не може да креира књигу”. Прекида се извршење сценарија. (ИА)



Слика 3 : СК1 – алтернативни сценарио 1

4.1 Уколико **систем** не може да запамти податке о књизи он приказује **раднику** поруку “Систем не може да запамти књигу”. (ИА)



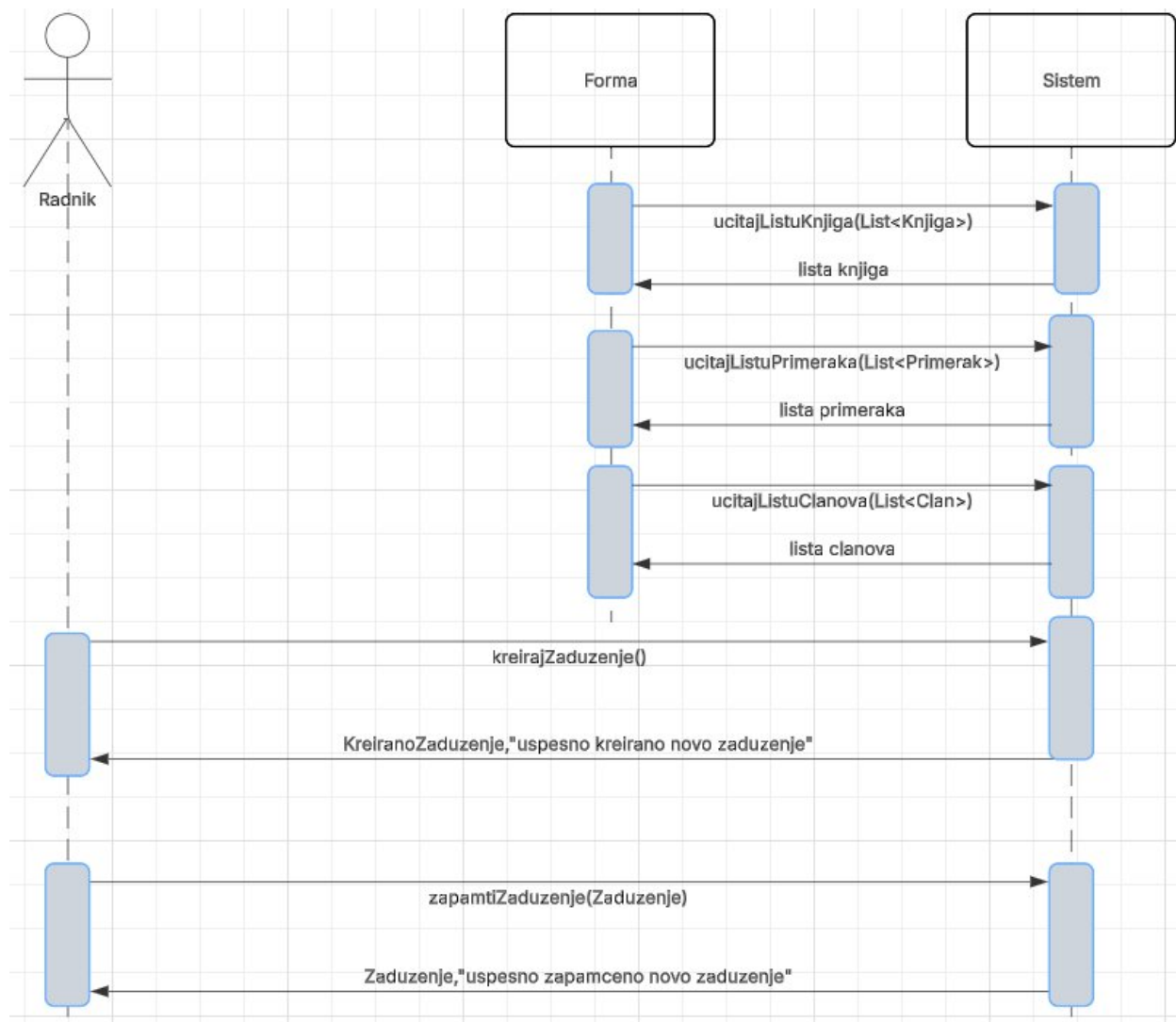
Слика 4 : SKI – алтернативни сценарио 2

Са наведених секвенцијалних дијаграма уочавају се следеће системске операције:

1. Signal **kreirajKnjigu(Knjiga)**;
2. Signal **zapamtiKnjigu(Knjiga)**;

ДС2: Дијаграм секвенци случаја коришћења – Унос новог задужења

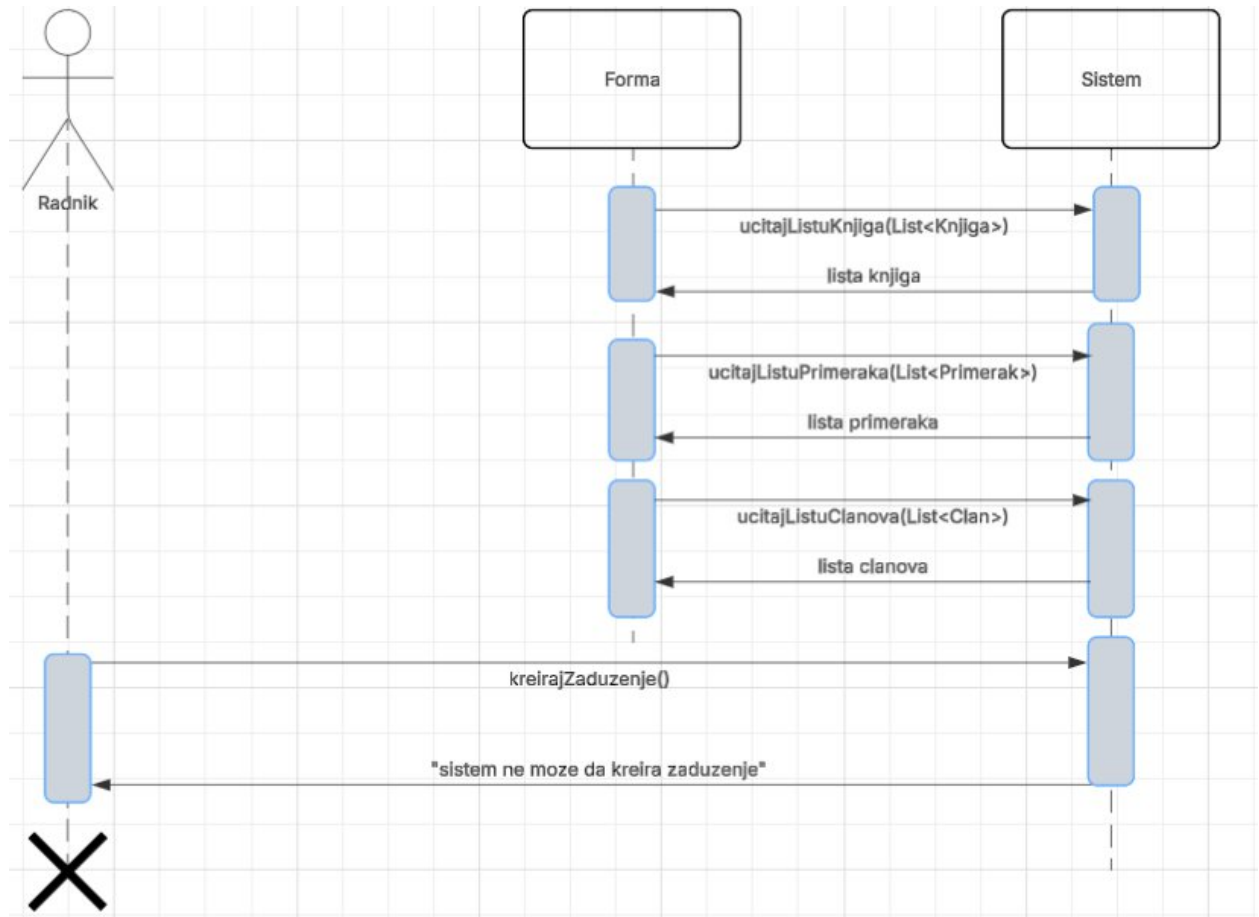
1. **Форма** позива **систем** да учита листу књига. (АПСО)
2. **Систем** враћа **форми** листу књига. (ИА)
3. **Форма** позива **систем** да учита листу примерака. (АПСО)
4. **Систем** враћа **форми** листу примерака. (ИА)
5. **Форма** позива **систем** да учита листу чланова. (АПСО)
6. **Систем** враћа **форми** листу чланова. (ИА)
7. **Радник** позива **систем** да креира ново задужење. (АПСО)
8. **Систем** приказује **раднику** ново задужење и поруку: "Успешно креирано ново задужење". (ИА)
9. **Радник** позива **систем** да запамти податке о задужењу. (АПСО)
10. **Систем** приказује **раднику** ново задужење и поруку: "Успешно запамћено ново задужење". (ИА)



Слика 5 : CK2 – унос новог задужења

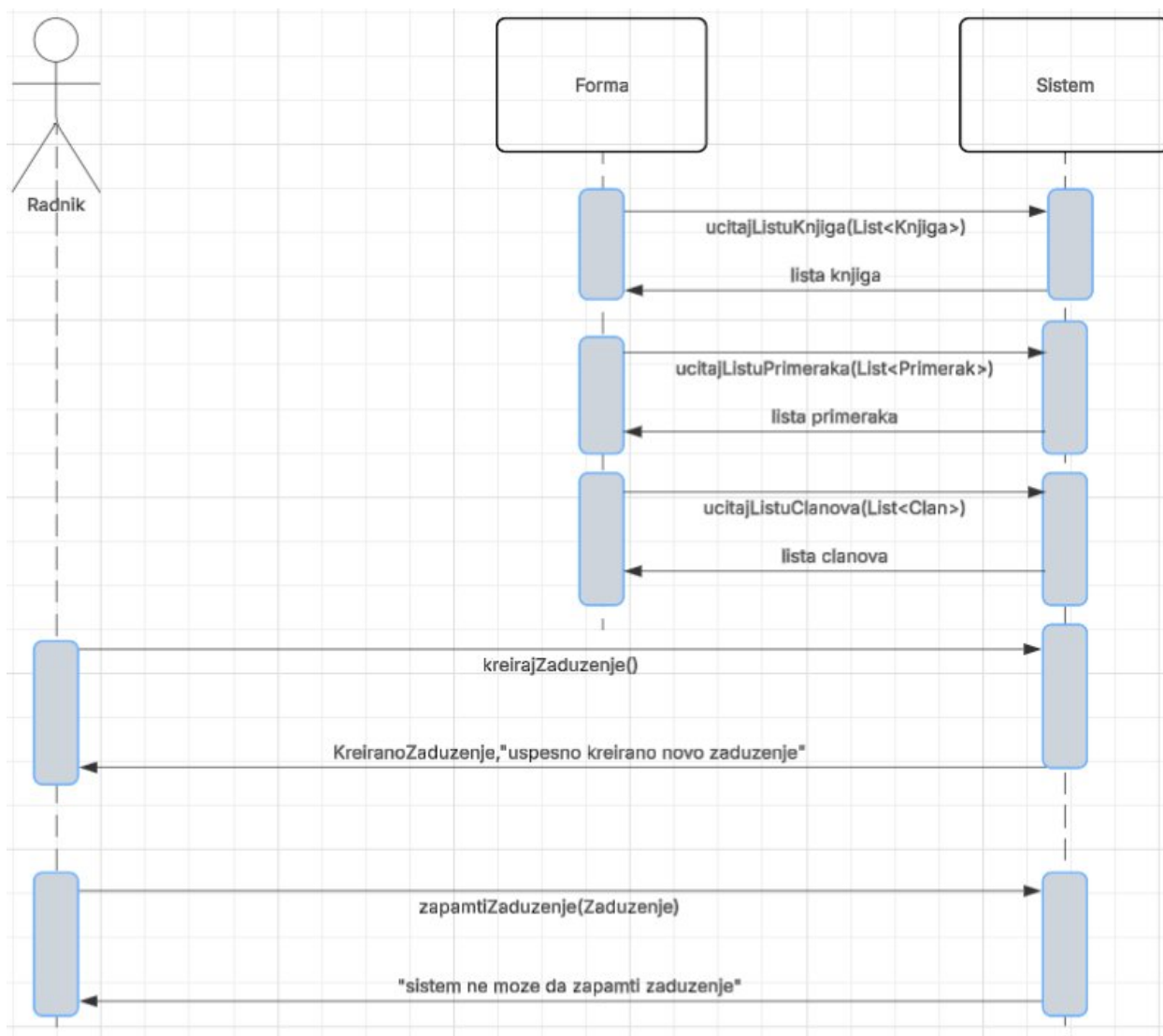
Алтернативна сценарија

8.1 Уколико **систем** не може да креира задужење, приказује **раднику** поруку: "Систем не може да креира задужење". Прекида се извршење сценарија. (ИА)



Слика 6 : СК2 – алтернативни сценарио 1

10.1 Уколико **систем** не може да запамти задужење, приказује **раднику** поруку: " Систем не може да запамти задужење". (ИА)



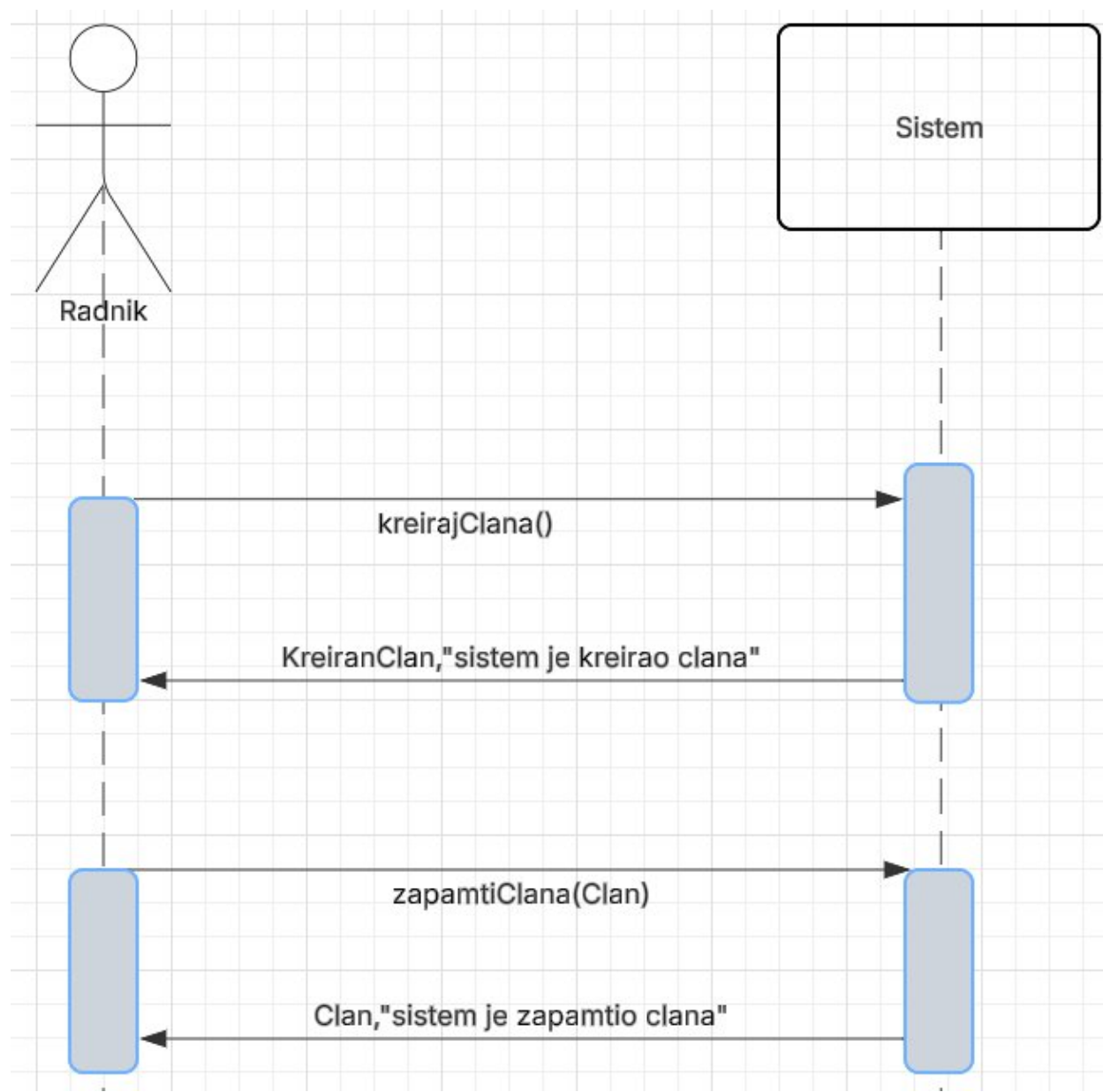
Слика 7 : CK2 – алтернативни сценарио 2

Са наведених секвенцијалних дијаграма уочавају се следеће системске операције:

1. Signal **ucitajListuKnjiga(List<Knjiga>);**
2. Signal **ucitajListuPrimeraka(List<Primerak>);**
3. Signal **ucitajListuClanova(List<Clan>);**
4. Signal **kreirajZaduzenje(Zaduzenje);**
5. Signal **zapamtiZaduzenje(Zaduzenje);**

ДСЗ: Дијаграм секвенци случаја коришћења – Унос новог члана

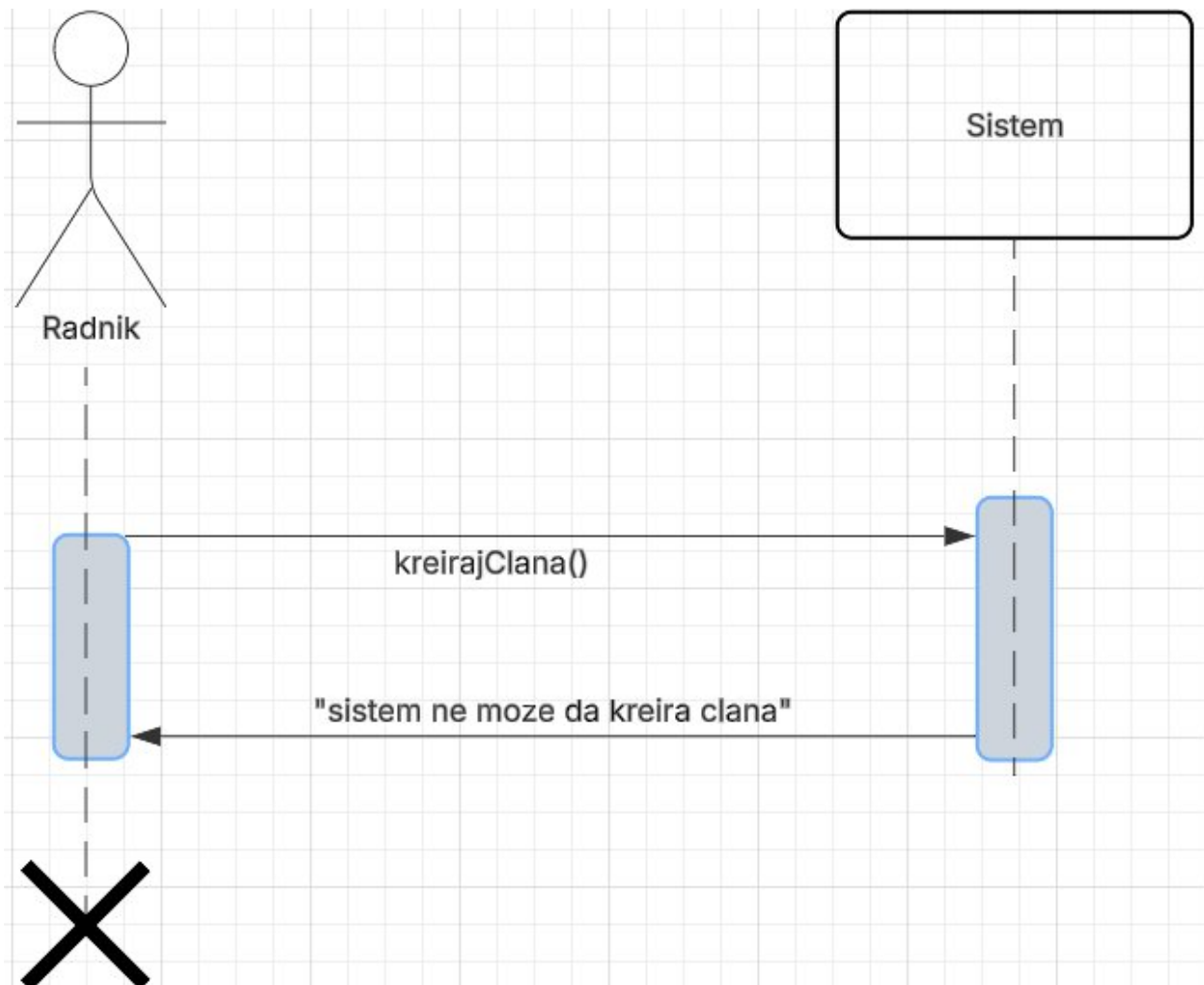
1. **Радник** позива **систем** да креира члана. (АПСО)
2. **Систем** приказује **раднику** члана и поруку: “Систем је креирао члана“. (ИА)
3. **Радник** позива **систем** да запамти податке о члану. (АПСО)
4. **Систем** приказује **раднику** запамћеног члана и поруку: “Систем је запамтио члана“. (ИА)



Слика 8 : СКЗ – унос новог члана

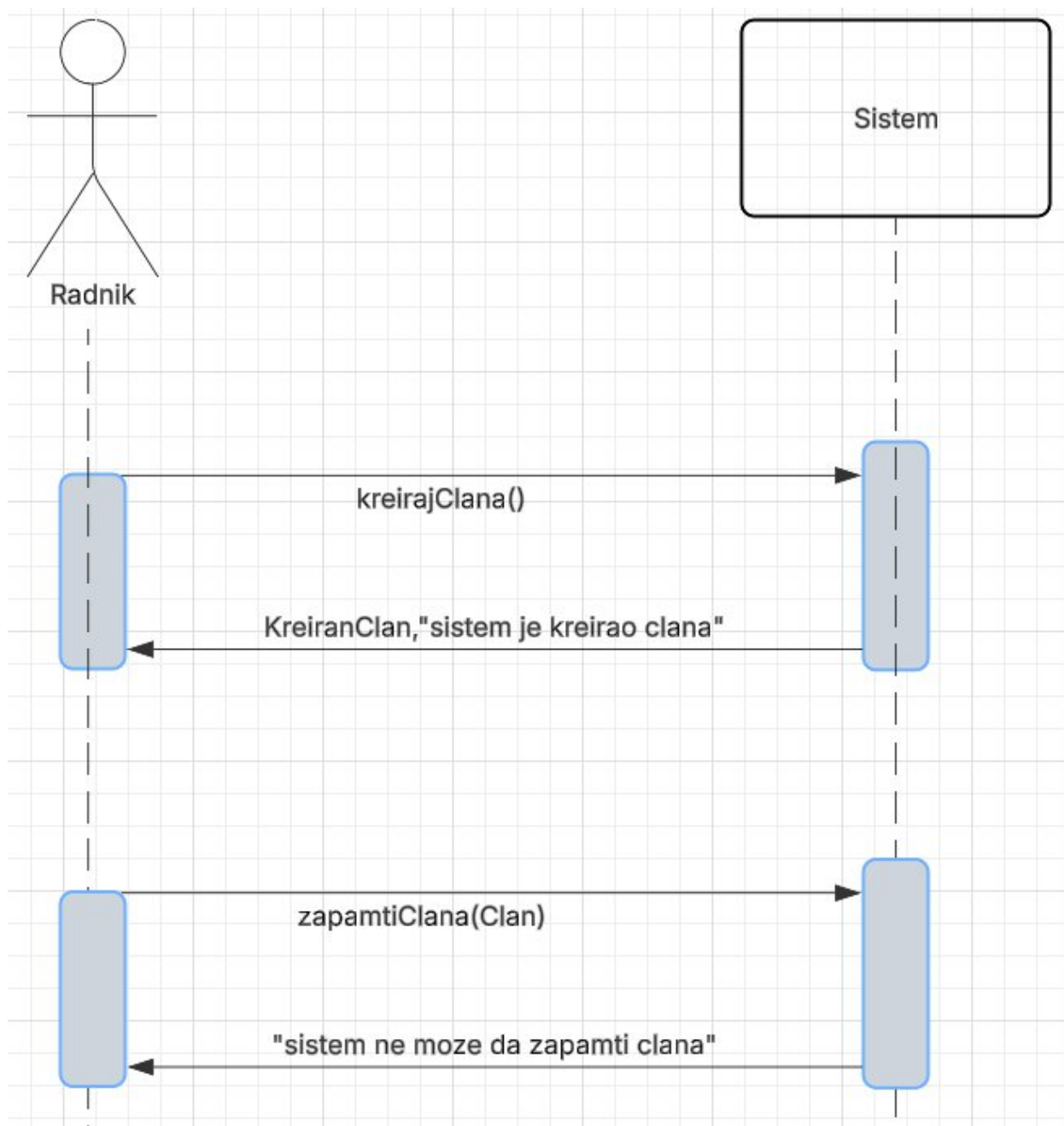
Алтернативна сценарија

2.1 Уколико **систем** не може да креира члана он приказује **раднику** поруку: “Систем не може да креира члана”. Прекида се извршење сценарија. (ИА)



Слика 9 : СКЗ – алтернативни сценарио 1

4.1 Уколико **систем** не може да запамти податке о члану он приказује **раднику** поруку “Систем не може да запамти члана”. (ИА)



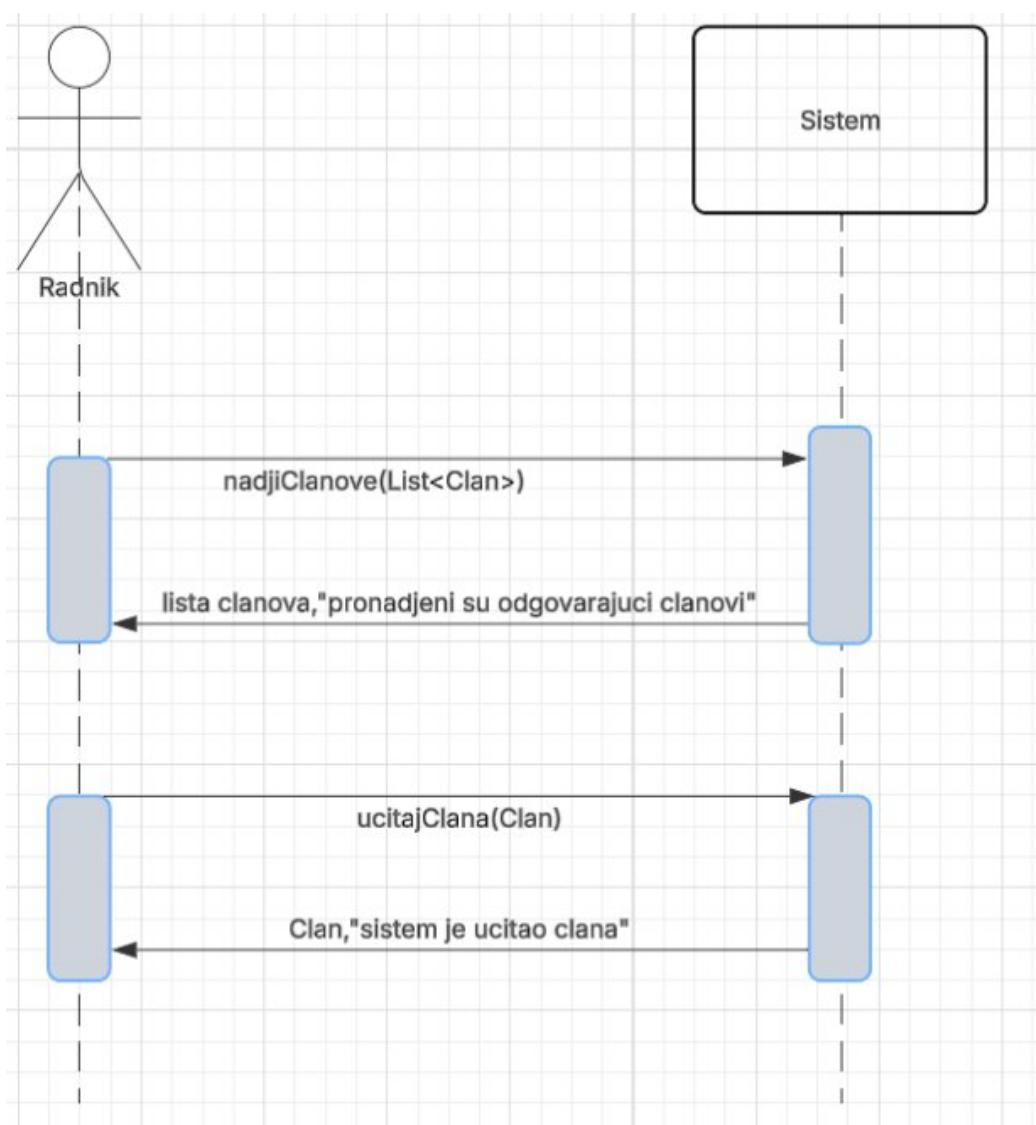
Слика 10 : СК3 – алтернативни сценарио 2

Са наведених секвенцијалних дијаграма уочавају се следеће системске операције:

1. Signal **kreirajClana(Clan);**
2. Signal **zapamtiClana(Clan);**

ДС4: Дијаграм секвенци случаја коришћења – претраживање члана

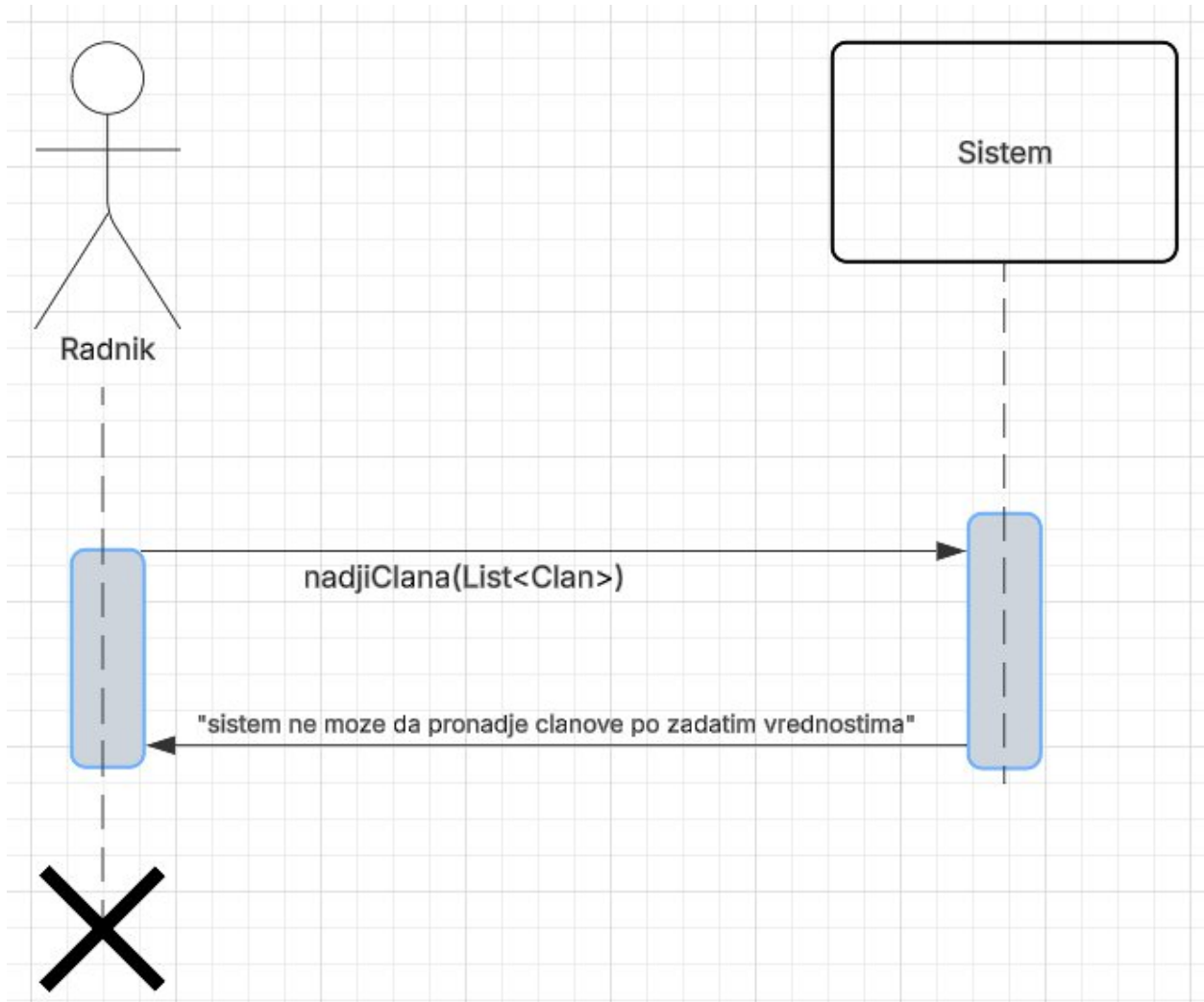
1. **Радник** позива **систем** да пронађе чланове по задатим вредностима. (АПСО)
2. **Систем** приказује листу чланова и поруку "Пронађени су одговарајући чланови". (ИА)
3. **Радник** позива **систем** да учита податке о изабраном члану. (АПСО)
4. **Систем** приказује податке о члану уз поруку "Успешно учитан члан". (ИА)



Слика 11 : СК4 – претраживање члана

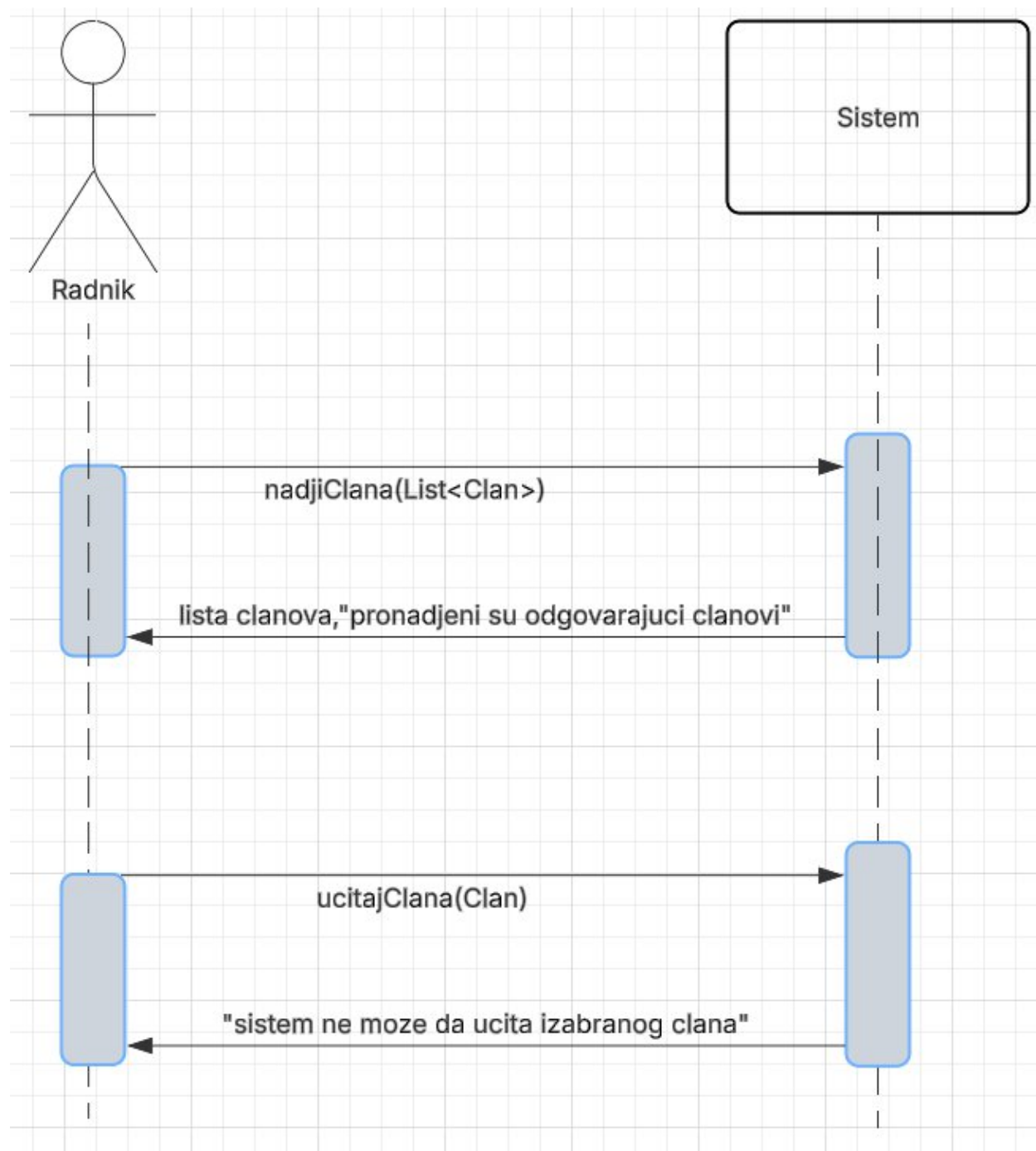
Алтернативна сценарија

2.1 Уколико **систем** не може да нађе чланове он приказује **раднику** поруку: “Систем не може да нађе чланове по задатим вредностима”. Прекида се извршење сценарија. (ИА)



Слика 12 : СК4 – алтернативни сценарио 1

4.1 Уколико **систем** не може да учита податке о члану он приказује **раднику** поруку:
"Систем не може да учита изабраног члана". (ИА)



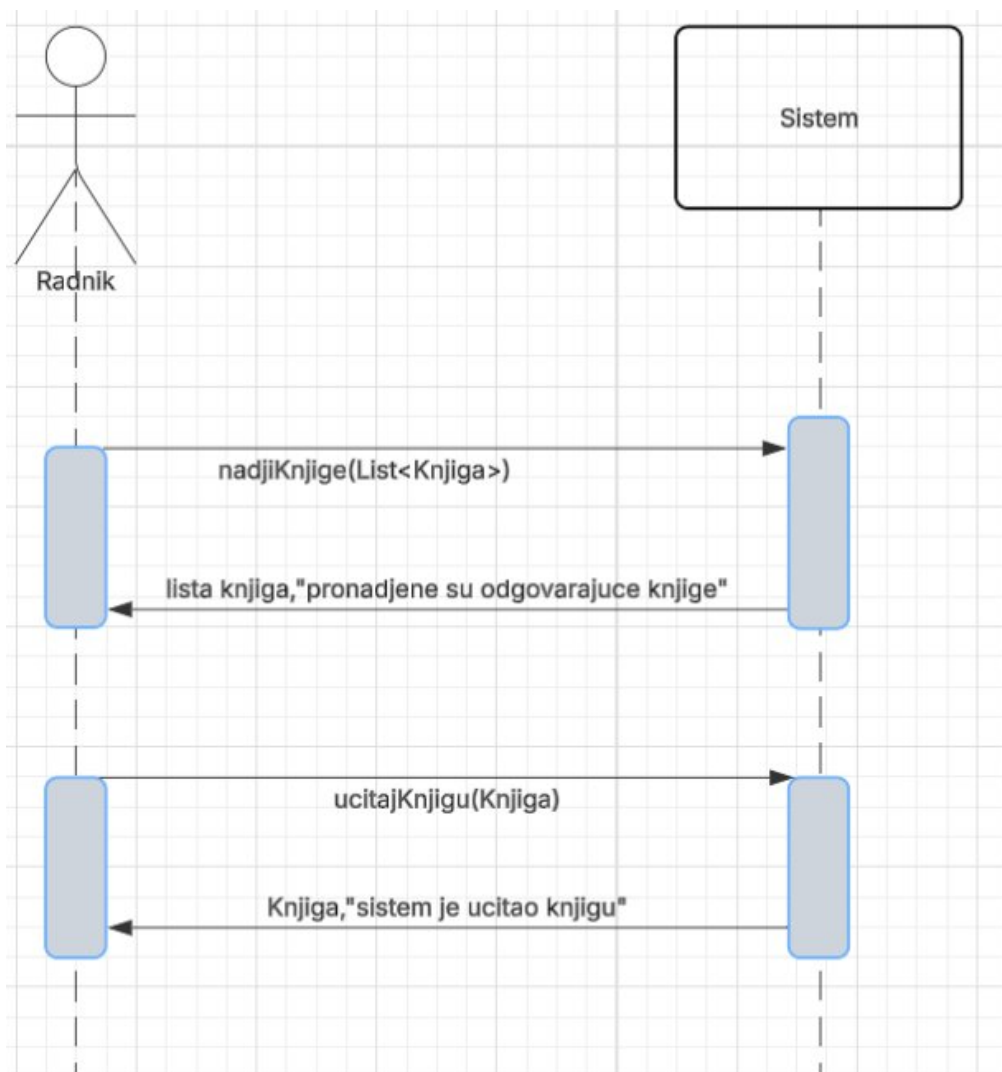
Слика 13 : СК4 – алтернативни сценарио 2

Са наведених секвенцијалних дијаграма уочавају се следеће системске операције:

1. Signal **nadjiClanove**(List<Clan>);
2. Signal **ucitajClana**(Clan);

ДС5: Дијаграм секвенци случаја коришћења – претраживање књиге

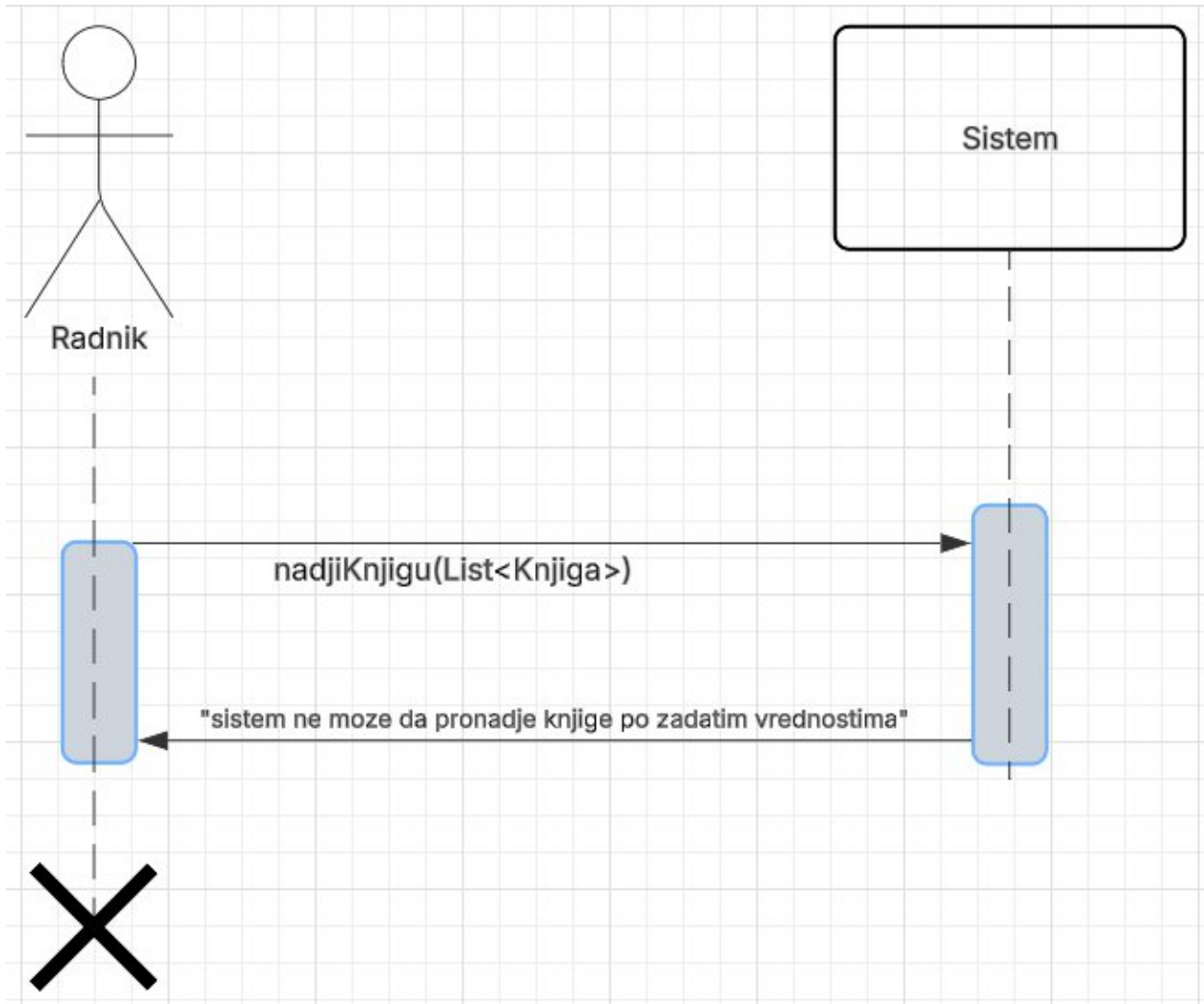
1. **Радник** позива **систем** да пронађе књиге по задатим вредностима. (АПСО)
2. **Систем** приказује листу књига и поруку "Пронађене су одговарајуће књиге". (ИА)
3. **Радник** позива **систем** да учита податке о изабраној књизи. (АПСО)
4. **Систем** приказује податке о књизи уз поруку "Успешно учитана књига". (ИА)



Слика 14 : СК5 – претраживање књиге

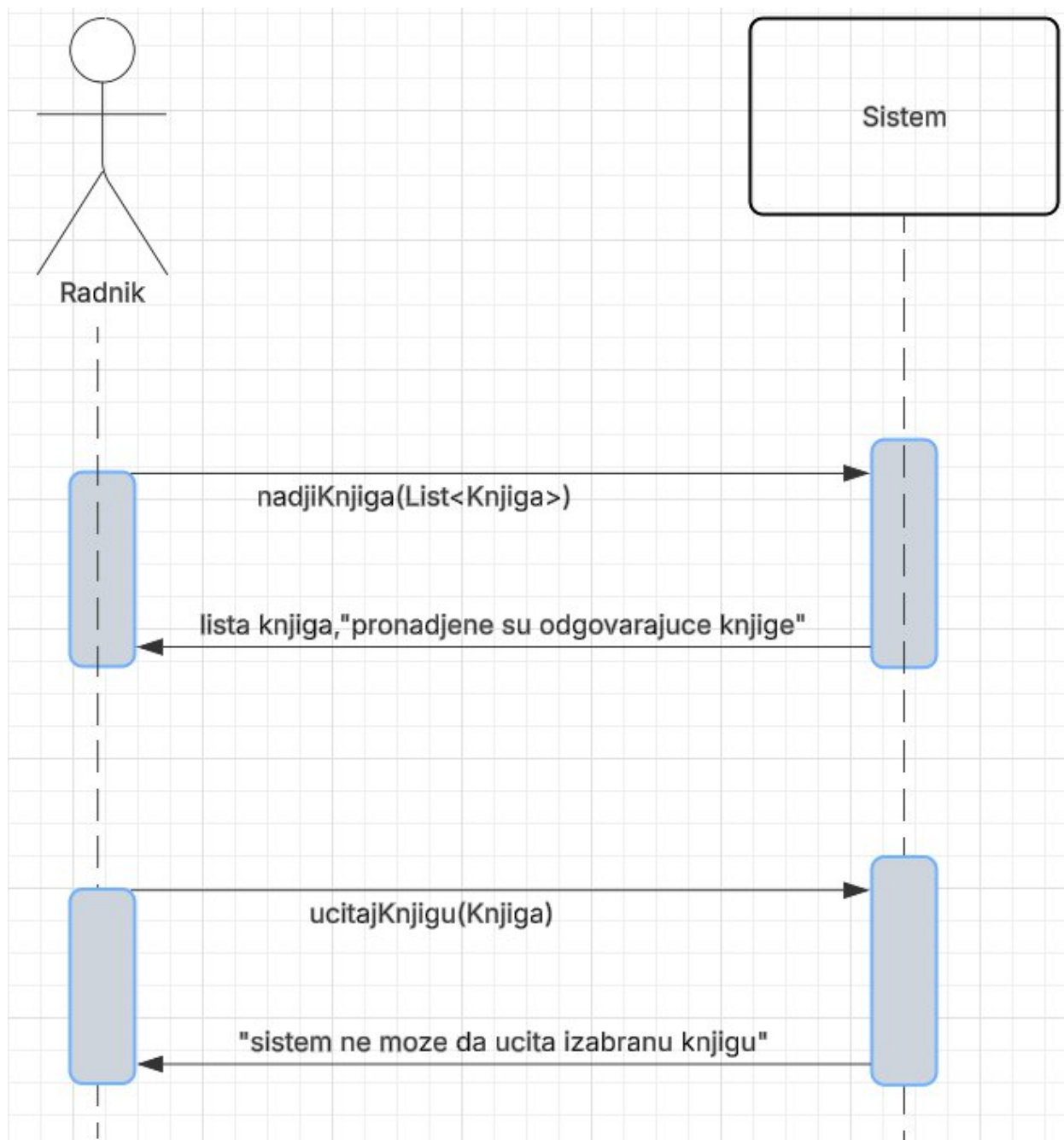
Алтернативна сценарија

2.1 Уколико **систем** не може да нађе књиге он приказује **раднику** поруку: “Систем не може да нађе књиге по задатим вредностима”. Прекида се извршење сценарија. (ИА)



Слика 15 : СК5 – алтернативни сценарио 1

4.1 Уколико **систем** не може да учита податке о књизи он приказује **раднику** поруку: "Систем не може да учита изабрану књигу". (ИА)



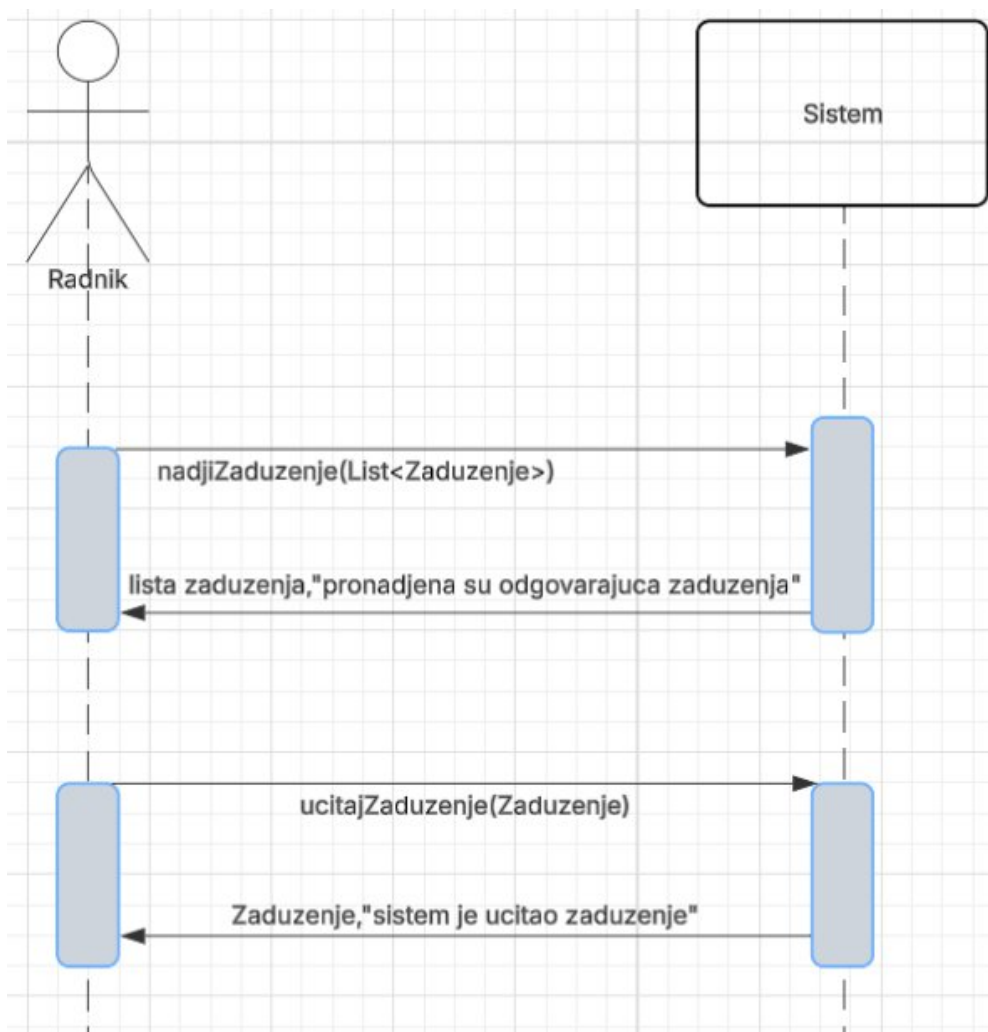
Слика 16 : CK5 – алтернативни сценарио 2

Са наведених секвенцијалних дијаграма уочавају се следеће системске операције:

1. Signal **nadjiKnjige(List<Knjiga>);**
2. Signal **ucitajKnjigu(Knjiga);**

ДС6: Дијаграм секвенци случаја коришћења – претраживање задужења

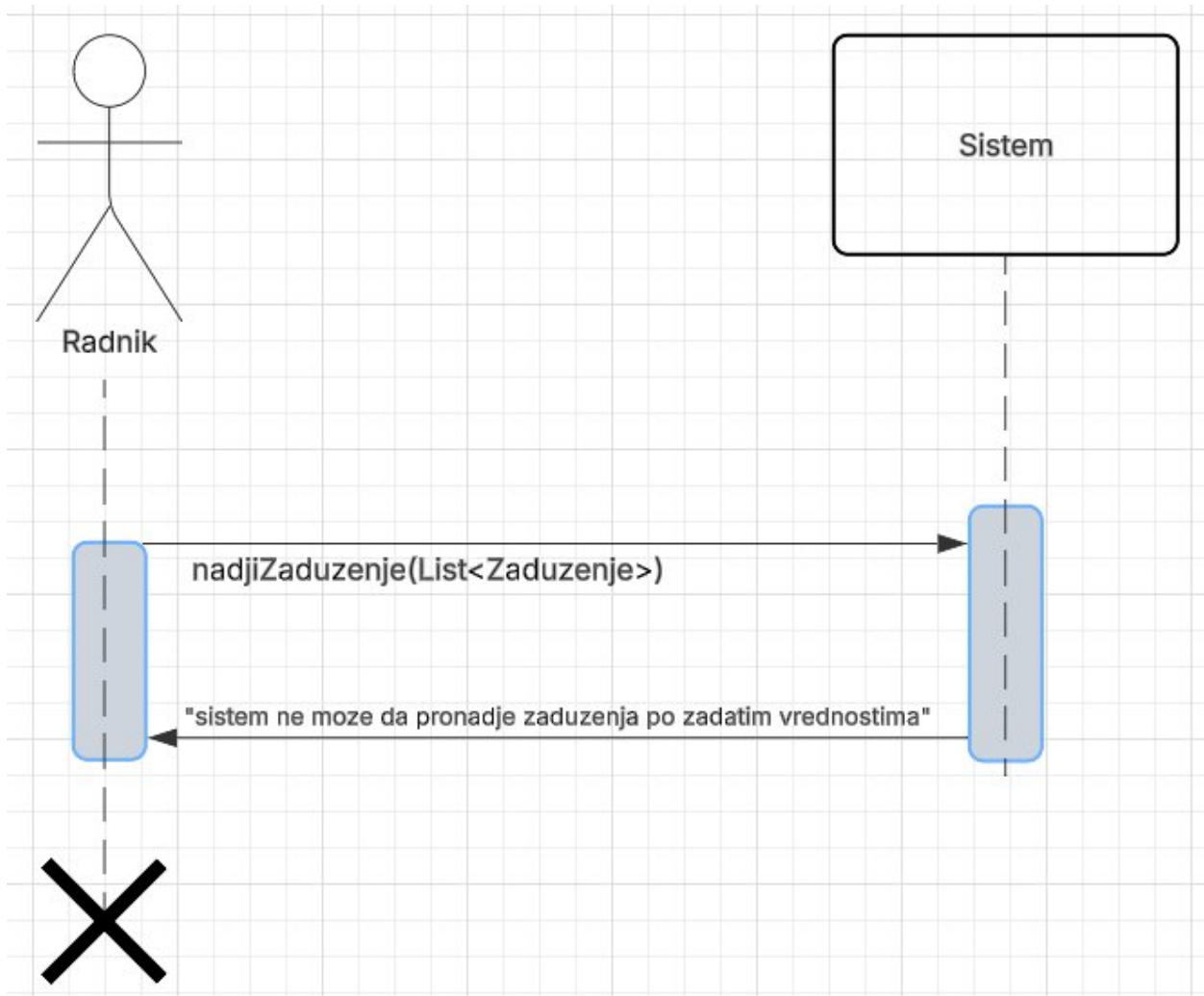
1. **Радник** позива **систем** да пронађе задужења по задатим вредностима. (АПСО)
2. **Систем** приказује листу задужења и поруку "Пронађена су одговарајућа задужења". (ИА)
3. **Радник** позива **систем** да учита податке о изабраном задужењу. (АПСО)
4. **Систем** приказује податке о задужењу уз поруку "Успешно учитано задужење". (ИА)



Слика 17 : СК6 – претраживање задужења

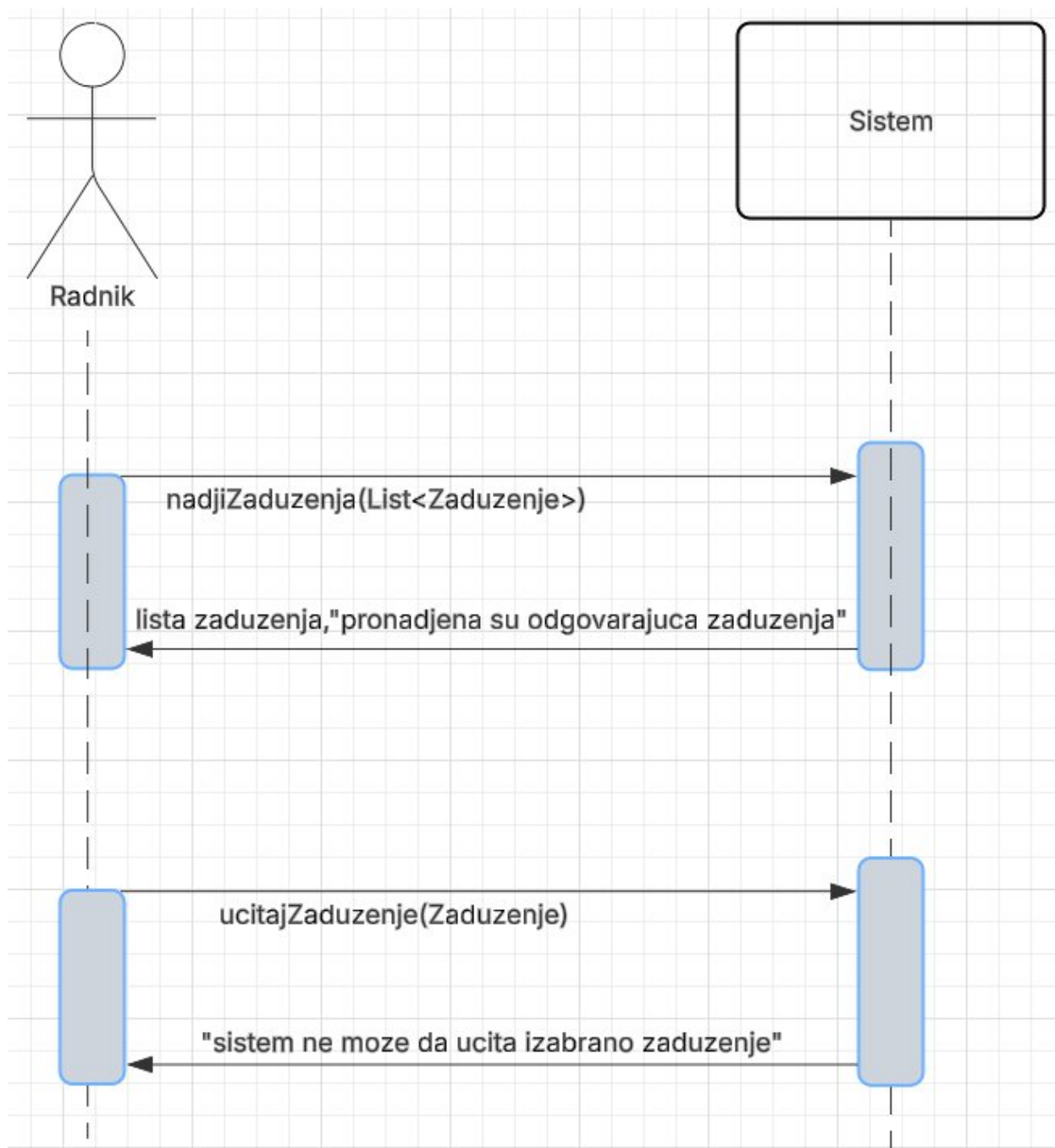
Алтернативна сценарија

2.1 Уколико **систем** не може да нађе задужења он приказује **раднику** поруку: “Систем не може да нађе задужења по задатим вредностима”. Прекида се извршење сценарија. (ИА)



Слика 18 : СК6 – алтернативни сценарио 1

4.1 Уколико **систем** не може да учита податке о задужењу он приказује **раднику** поруку: "Систем не може да учита изабрано задужење". (ИА)



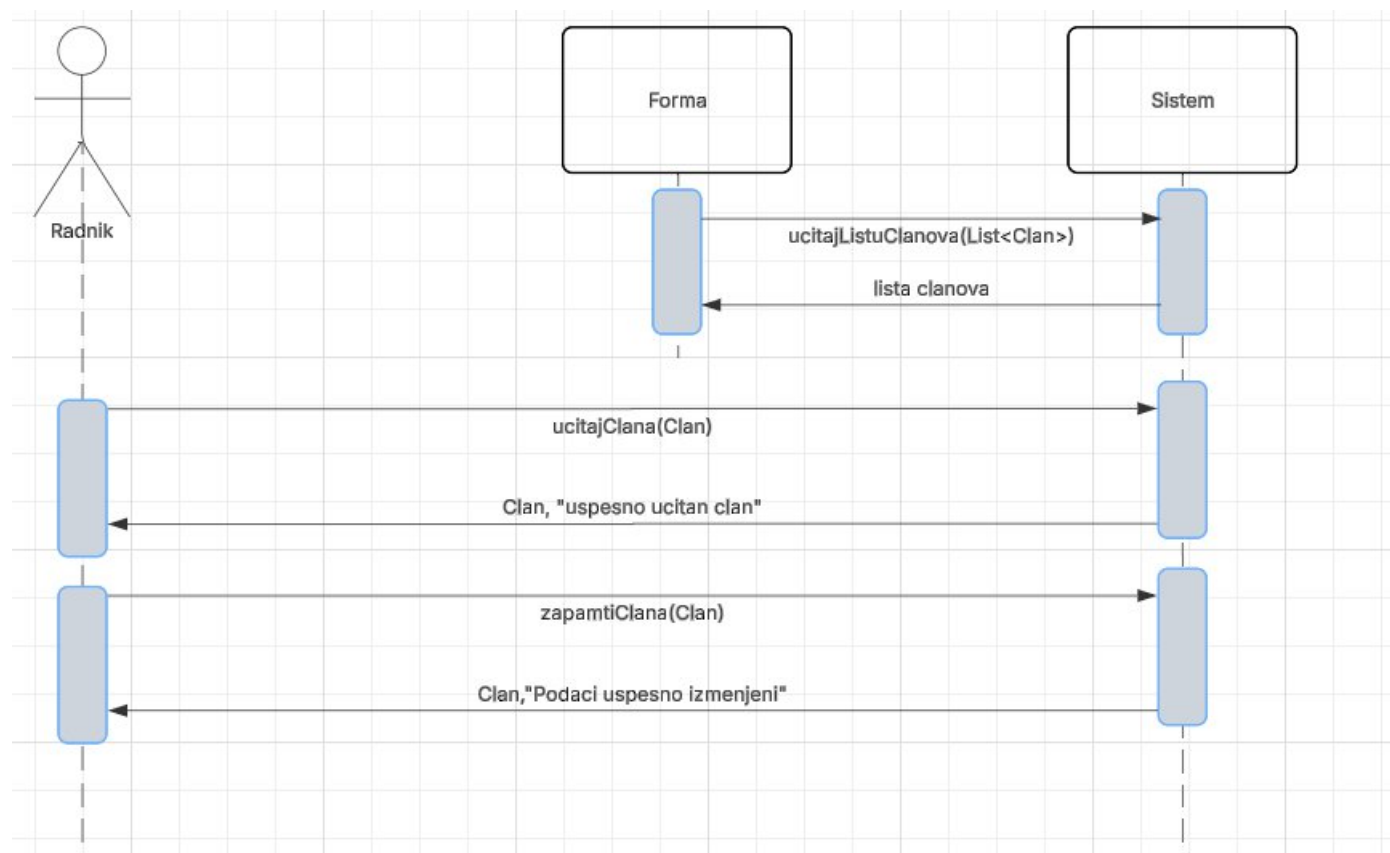
Слика 19 : СК6 – алтернативни сценарио 2

Са наведених секвенцијалних дијаграма уочавају се следеће системске операције:

1. Signal **nadjiZaduzenja(List<Zaduzenje>);**
2. Signal **ucitajZaduzenje(Zaduzenje);**

ДС7: Дијаграм секвенци случаја коришћења – измена члана

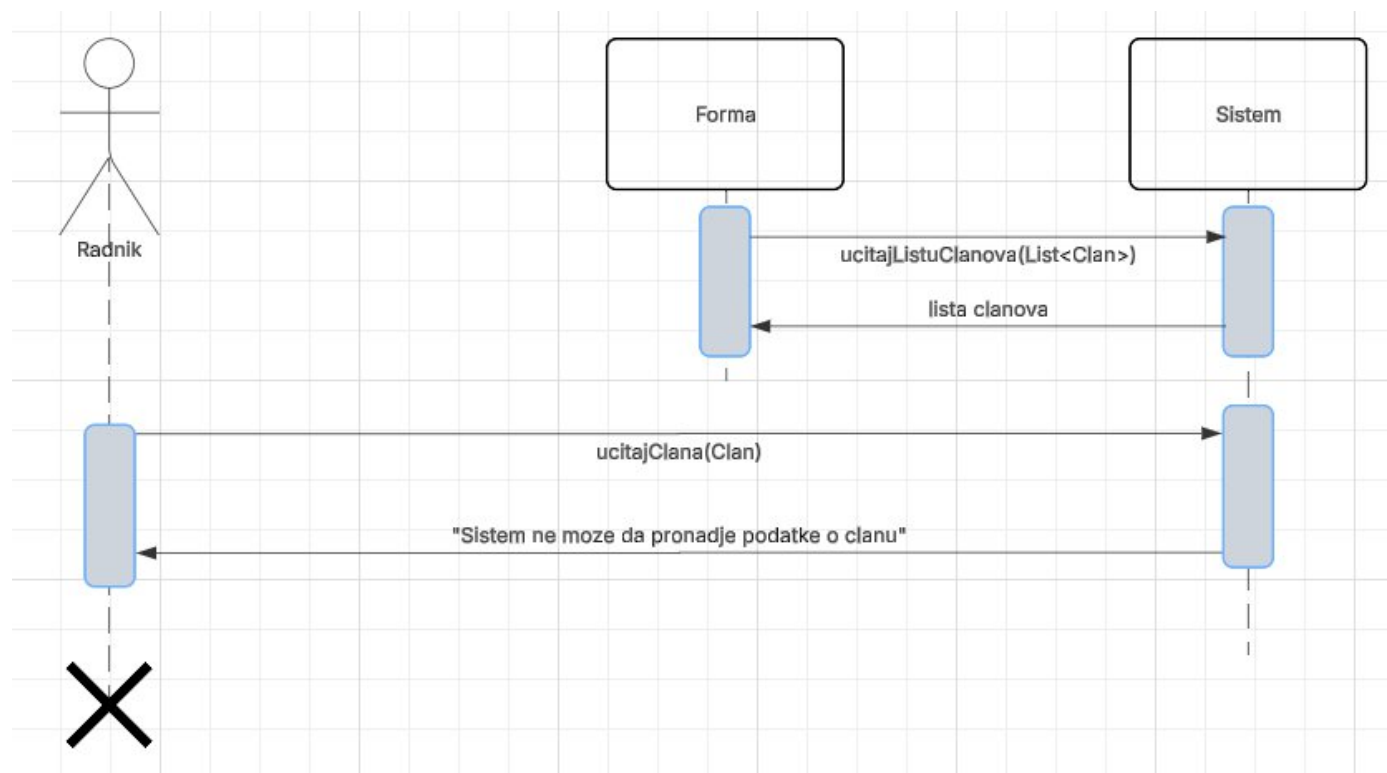
1. **Форма** позива **систем** да учита листу чланова. (АПСО)
2. **Систем** враћа **форми** листу чланова. (ИА)
3. **Радник** позива **систем** да учита податке о изабраном члану. (АПСО)
4. **Систем** приказује податке о члану уз поруку "Успешно учитан члан". (ИА)
5. **Радник** позива **систем** да запамти нове податке о члану. (АПСО)
6. **Систем** приказује **раднику** поруку: "Подаци успешно измењени". (ИА)



Слика 20 : СК7 – измена члана

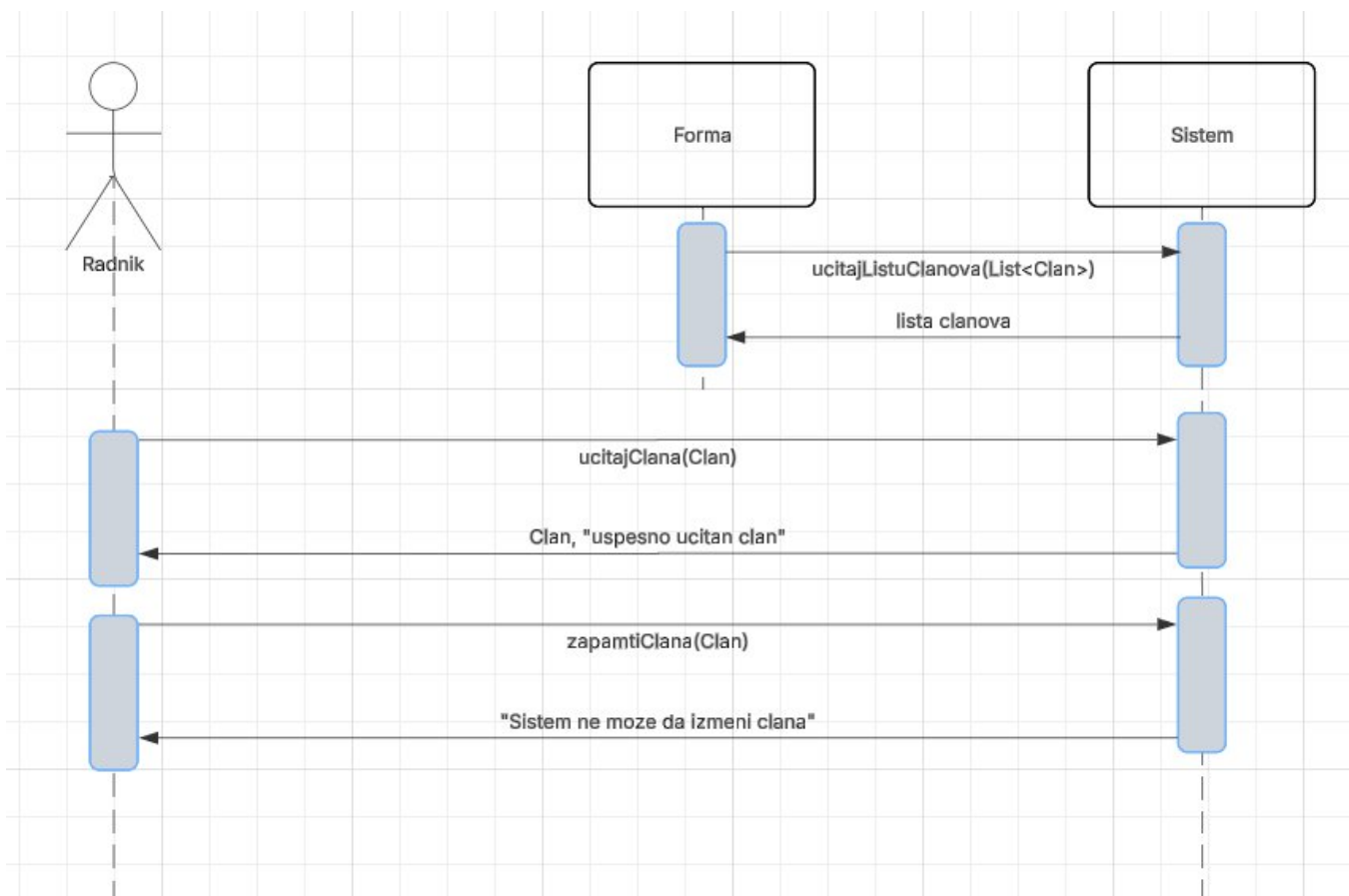
Алтернативна сценарија

4.1 Уколико **систем** не може да пронађе податке о члану, **систем** приказује **раднику** поруку: " Систем не може да пронађе податке о члану". Прекида се извршење сценариа. (ИА)



Слика 21 : СК7 – алтернативни сценарио 1

6.1 Уколико **систем** не може да сачува измене, приказује **раднику** поруку: " Систем не може да измени члана". (ИА)



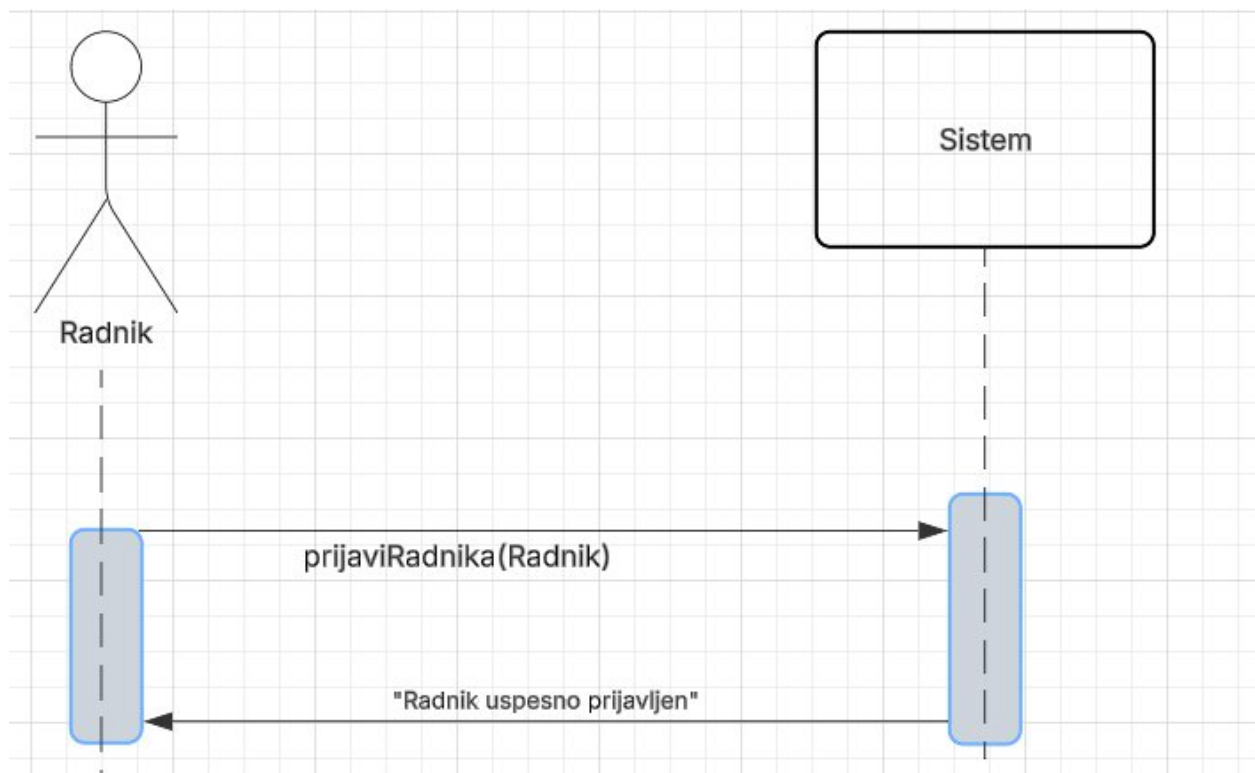
Слика 22 : CK7 – алтернативни сценарио 2

Са наведених секвенцијалних дијаграма уочавају се следеће системске операције:

1. Signal **ucitajListuClanova(List<Clan>);**
2. Signal **ucitajClana(Clan);**
3. Signal **zapamtiClana(Clan);**

ДС8: Дијаграм секвенци случаја коришћења – пријављивање радника

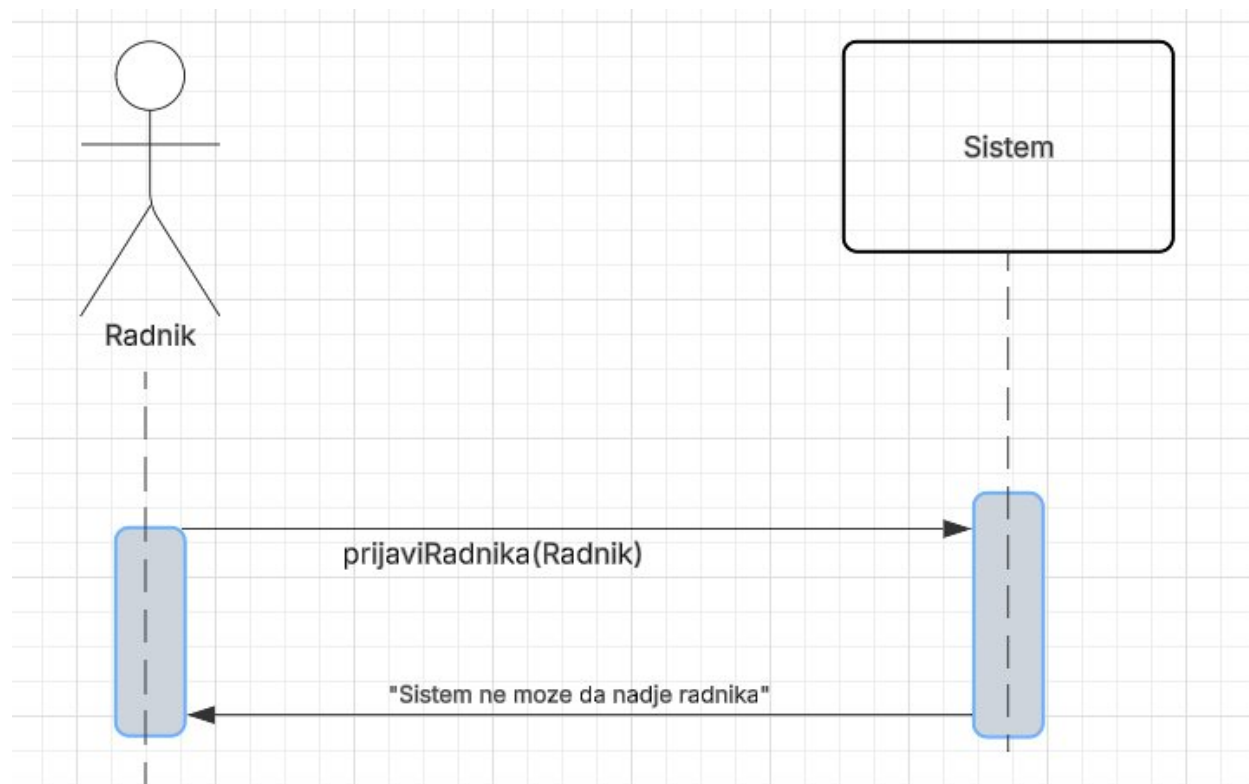
1. **Радник** позива **систем** да пронађе **радника** са задатим подацима (АПСО)
2. **Систем** приказује поруку “Радник је успешно пријављен” (ИА)



Слика 23 : СК8 – пријављивање радника

Алтернативна сценарија

2.1 Уколико **систем** не може да нађе **радника** он приказује поруку: “Систем не може да нађе радника на основу унетих вредности”.(ИА)



Слика 24 : СК8 – алтернативни сценарио 1

Са наведених секвенцијалних дијаграма уочавају се следеће системске операције:

1. Signal **prijaviRadnika(Radnik);**

На основу анализе сценарија добијено је 16 системска операција:

1. Signal **kreirajKnjigu(Knjiga);**
2. Signal **zapamtiKnjigu(Knjiga);**
3. Signal **ucitajListuKnjiga(List<Knjiga>);**
4. Signal **ucitajListuPrimeraka(List<Primerak>);**
5. Signal **ucitajListuClanova(List<Clan>);**
6. Signal **kreirajZaduzenje(Zaduzenje);**
7. Signal **zapamtiZaduzenje(Zaduzenje);**
8. Signal **kreirajClana(Clan);**
9. Signal **zapamtiClana(Clan);**
10. Signal **nadjiClanove(List<Clan>);**
11. Signal **ucitajClana(Clan);**
12. Signal **nadjiKnjige(List<Knjiga>);**
13. Signal **ucitajKnjigu(Knjiga);**
14. Signal **nadjiZaduzenja(List<Zaduzenje>);**
15. Signal **ucitajZaduzenje(Zaduzenje);**
16. Signal **prijaviRadnika(Radnik);**

4.2. Понашање система – Дефинисање уговора о системским операцијама

Уговор УГ1

Операција: **kreirajKnjigu(Knjiga):Signal;**

Веза са СК: СК1

Предуслови: Структурна и вредносна ограничења над Knjiga објектом морају бити задовољена.

Постуслови: Направљена је нова књига.

Уговор УГ2

Операција: **zapamtiKnjigu(Knjiga):Signal;**

Веза са СК: СК1

Предуслови: Структурна и вредносна ограничења над Knjiga објектом морају бити задовољена.

Постуслови: Књига је запамћена.

Уговор УГ3

Операција **ucitajListuKnjiga(List<Knjiga>):Signal;**

Веза са СК: СК2

Предуслови: /

Постуслови: /

Уговор УГ4

Операција: **ucitajListuPrimeraka(List<Primerak>):Signal;**

Веза са СК: СК2

Предуслови: /

Постуслови: /

Уговор УГ5

Операција: **ucitajListuClanova(List<Clan>):Signal;**

Веза са СК: СК2, СК7

Предуслови: /

Постуслови: /

Уговор УГ6

Операција: **kreirajZaduzenje(Zaduzenje):Signal;**

Веза са СК: СК2

Предуслови: Структурна и вредносна ограничења над *Zaduzenje* објектом морају бити задовољена.

Постуслови: Направљено је ново задужење.

Уговор УГ7

Операција: **zapamtiZaduzenje(Zaduzenje) :Signal;**

Веза са СК: СК2

Предуслови: Структурна и вредносна ограничења над *Zaduzenje* објектом морају бити задовољена.

Постуслови: Задужење је запамћено.

Уговор УГ8

Операција: **kreirajClana(Clan) :Signal;**

Веза са СК: СК3

Предуслови: Структурна и вредносна ограничења над Clan објектом морају бити задовољена.

Постуслови: Направљен је нови члан.

Уговор УГ9

Операција: **zapamtiClana(Clan):Signal;**

Веза са СК: СК3,СК7

Предуслови: Структурна и вредносна ограничења над Clan објектом морају бити задовољена.

Постуслови: Члан је запамћен.

Уговор УГ10

Операција: **nadjiClanove(List<Clan>) :Signal;**

Веза са СК: СК4

Предуслови: /

Постуслови: Пронађен је тражени Члан.

Уговор УГ11

Операција: **ucitajClana(Clan) :Signal;**

Веза са СК: СК4, СК7

Предуслови: /

Постуслови: /

Уговор УГ12

Операција: **nadjiKnjige(List<Knjiga>) :Signal;**

Веза са СК: СК5

Предуслови: /

Постуслови: Пронађена је тражена Књига.

Уговор УГ13

Операција: **ucitajKnjigu(Knjiga):Signal;**

Веза са СК: СК5

Предуслови: /

Постуслови: /

Уговор УГ14

Операција: **nadjiZaduzenja(List<Zaduzenje>):Signal;**

Веза са СК: СК6

Предуслови: /

Постуслови: Пронађено је тражено задужење.

Уговор УГ15

Операција: **ucitajZaduzenje(Zaduzenje):Signal;**

Веза са СК: СК6

Предуслови: /

Постуслови: /

Уговор УГ16

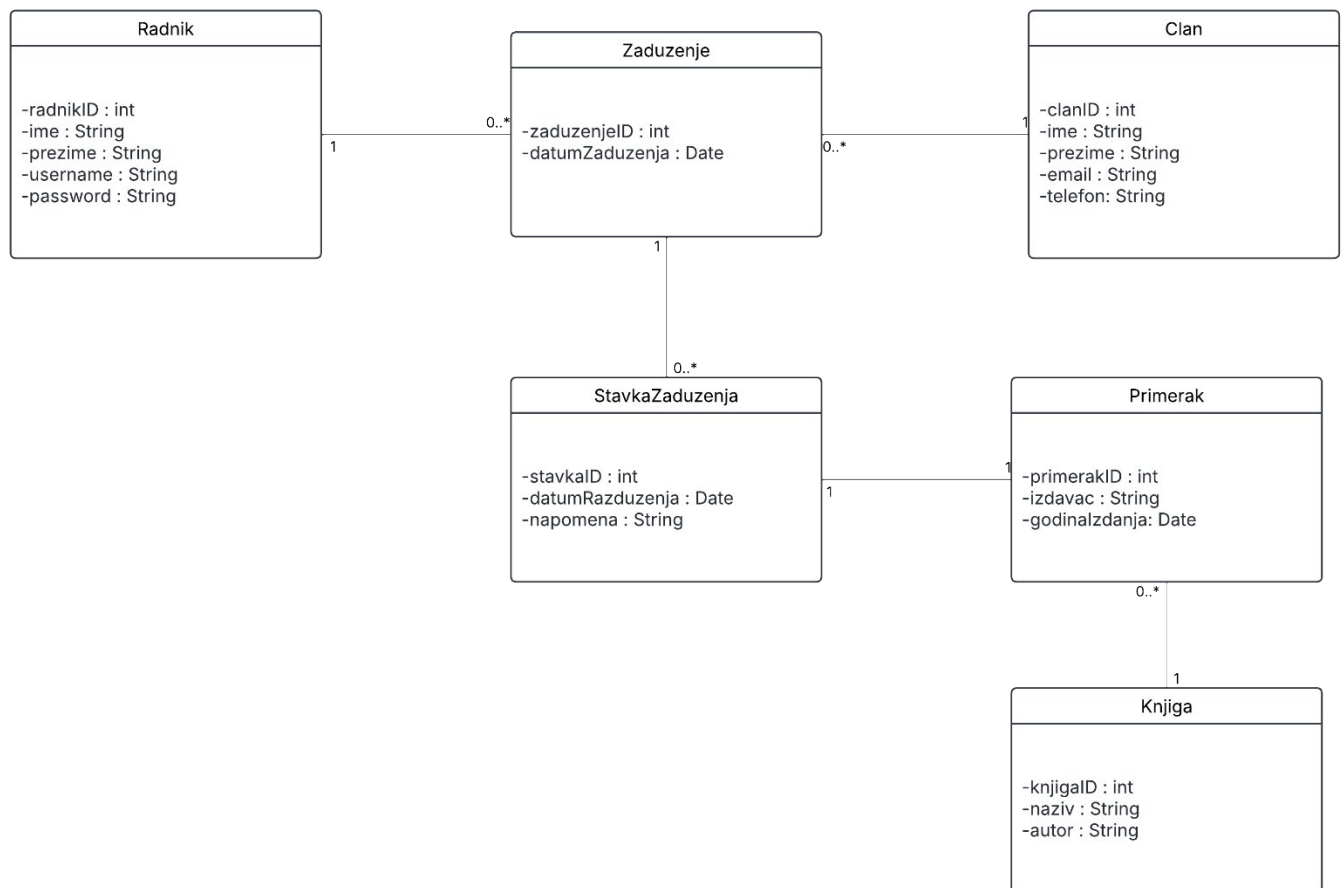
Операција: **prijaviRadnika(Radnik):Signal;**

Веза са СК: СК8

Предуслови: Структурна и вредносна ограничења над објектом Radnik морају бити задовољена. Радник са истим крединцијалима не може већ бити улогован.

Постуслови: Радник је улогован на систем.

4.3. Структура Система – Концептуални (доменски модел)



Слика 25: Концептуални модел

4.4. Структура система – релациони модел

Radnik (radnikID, ime, prezime, username, password)

Clan (clanID, ime, prezime, telefon, email)

Zaduzenje (zaduzenjeID, datumZaduzenja, *radnikID*, *clanID*)

Knjiga (knjigaID, naziv, autor)

Primerak (primerakID, izdavac , godinaIzdanja, *knjigaID*)

StavkaZaduzenja (**zaduzenjeID**, **stavkaID**, datumRazduzenja, napomena, , *primerakID*)

Tabela Radnik		Prosto vrednosno ogranicenje		Slozeno vrednosno ogranicenje		Strukturno ogranicenje
Atributi	Ime	Tip atributa	Vrednost atributa	Medjuzavisnost atributa jedne tabele	Medjuzavisnost atributa vise tabela	INSERT: / UPDATE: Cascades Zaduzenje DELETE: Restricted Zaduzenje
	radnikID	integer	Not null			
	ime	String				
	prezime	String				
	username	String				
	password	String				

Tabela Clan		Prosto vrednosno ogranicenje		Slozeno vrednosno ogranicenje		Strukturno ogranicenje
Atributi	Ime	Tip atributa	Vrednost atributa	Medjuzavisnost atributa jedne tabele	Medjuzavisnost atributa vise tabela	INSERT: / UPDATE: Cascades Zaduzenje DELETE: Restricted Zaduzenje
	clanID	integer	Not null			
	ime	String				
	prezime	String				
	telefon	String				
	email	String				

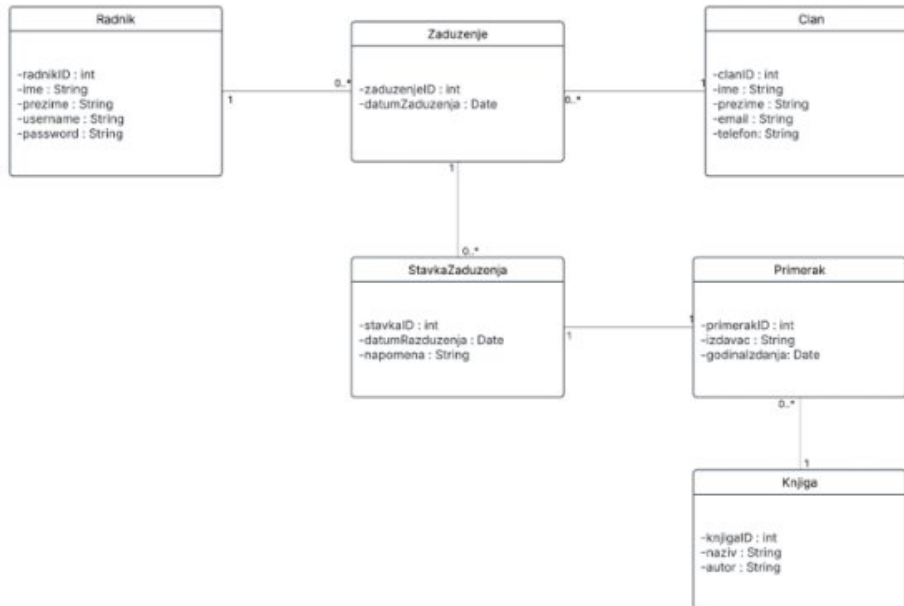
Tabela Zaduzenje		Prosto vrednosno ogranicenje		Slozeno vrednosno ogranicenje		Strukturno ogranicenje
Atributi	Ime	Tip atributa	Vrednost atributa	Medjuzavisnost atributa jedne tabele	Medjuzavisnost atributa vise tabela	INSERT: / UPDATE: Cascades StavkaZaduzenja DELETE: Cascades StavkaZaduzenja
	zaduzenjeID	integer	Not null			
	datumZaduzenja	date				
	clanID	integer	Not null			
	radnikID	integer	Not null			

Tabela Knjiga		Prosto vrednosno ogranicenje		Slozeno vrednosno ogranicenje		Strukturno ogranicenje
Atributi	Ime	Tip atributa	Vrednost atributa	Medjuzavisnost atributa jedne tabele	Medjuzavisnost atributa vise tabela	INSERT: / UPDATE: Cascades Primerak DELETE: Cascades Primerak
	knjigaID	integer	Not null			
	naziv	String				
	autor	String				

Tabela Primerak		Prosto vrednosno ogranicenje		Slozeno vrednosno ogranicenje		Strukturno ogranicenje
Atributi	Ime	Tip atributa	Vrednost atributa	Medjuzavisnost atributa jedne tabele	Medjuzavisnost atributa vise tabela	INSERT: Restricted Knjiga UPDATE: Restricted Knjiga DELETE: /
	primerakID	integer	Not null			
	izdavac	String				
	godinaIzdanja	date				
	knjigaID	integer	Not null			

Tabela StavkaZaduzenja		Prosto vrednosno ogranicenje		Slozeno vrednosno ogranicenje		Strukturno ogranicenje
Atributi	Ime	Tip atributa	Vrednost atributa	Medjuzavisnost atributa jedne tabele	Medjuzavisnost atributa vise tabela	INSERT: Restricted Zaduzenje Primerak UPDATE: Restricted Zaduzenje Primerak DELETE: /
	zaduzenjeID	integer	Not null			
	stavkaID	integer	Not null			
	datumRazduzenja	date	Default: Null			
	napomena	String				
	primerakID	integer	Not null			

Struktura Sistema



Ponašanje Sistema

Sistemske Operacije

1. Signal **kreirajKnjigu**(Knjiga);
2. Signal **zapamtiKnjigu**(Knjiga);
3. Signal **ucitajListuKnjiga**(List<Knjiga>);
4. Signal **ucitajListuPrimeraka**(List<Primerak>);
5. Signal **ucitajListuClanova**(List<Clan>);
6. Signal **kreirajZaduzenje**(Zaduzenje);
7. Signal **zapamtiZaduzenje**(Zaduzenje);
8. Signal **kreirajClana**(Clan);
9. Signal **zapamtiClana**(Clan);
10. Signal **nadjiClanove**(List<Clan>);
11. Signal **ucitajClana**(Clan);
12. Signal **nadjiKnjige**(List<Knjiga>);
13. Signal **ucitajKnjigu**(Knjiga);
14. Signal **nadjiZaduzenja**(List<Zaduzenje>);
15. Signal **ucitajZaduzenje**(Zaduzenje);
16. Signal **prijaviRadnika**(Radnik);

Слика 26 : Структура и понашање система

5. Пројектовање

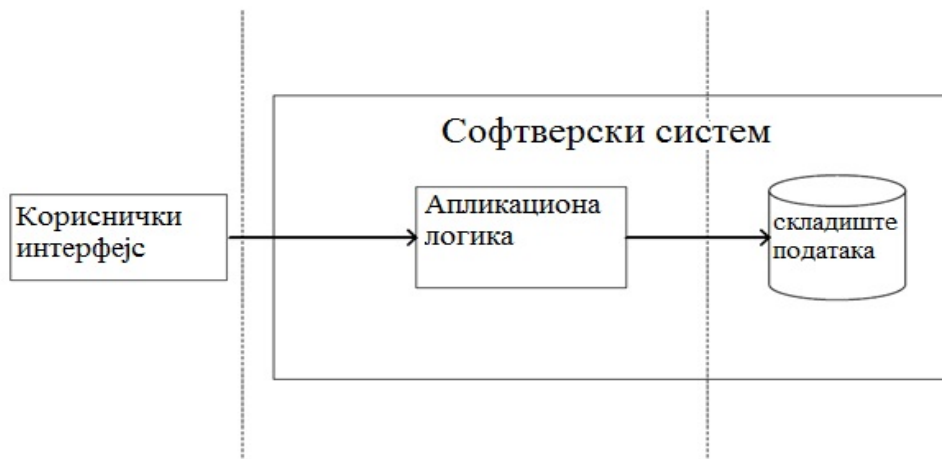
Фаза пројектовања описује физичку структуру и понашање софтверског система (архитектуру софтверског система).

5.1 Архитектура софтверског система

Архитектура софтверског систем је тронивојска и састоји се од следећих нивоа:

- Кориснички интерфејс
- Апликациона логика
- Складиште података

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



Слика 27 : Архитектура софтверског система

5.2 Пројектовање корисничког интерфејса

Кориснички интерфејс представља реализацију улаза и/или излаза софтверског система и састоји се од екранске форме и контролера корисничког интерфејса.



Слика 28 : Приказ корисничког интерфејса

5.3 Пројектовање екранских форми

Кориснички интерфејс је дефинисан преко скупа екранских форми. Сценарија коришћења екранских форми су директно повезани са сценаријима случајева коришћења. Екранска форма има улогу да прихвати податке које уноси актор, прихвата догађаје које прави актор, позива контролера корисничког интерфејса како би му проследио те податке и приказује податке добијене од контролера корисничког интерфејса.

5.3.1 СК1: Случај коришћења –Унос нове књиге (сложен)

Назив СК

Унос нове књиге

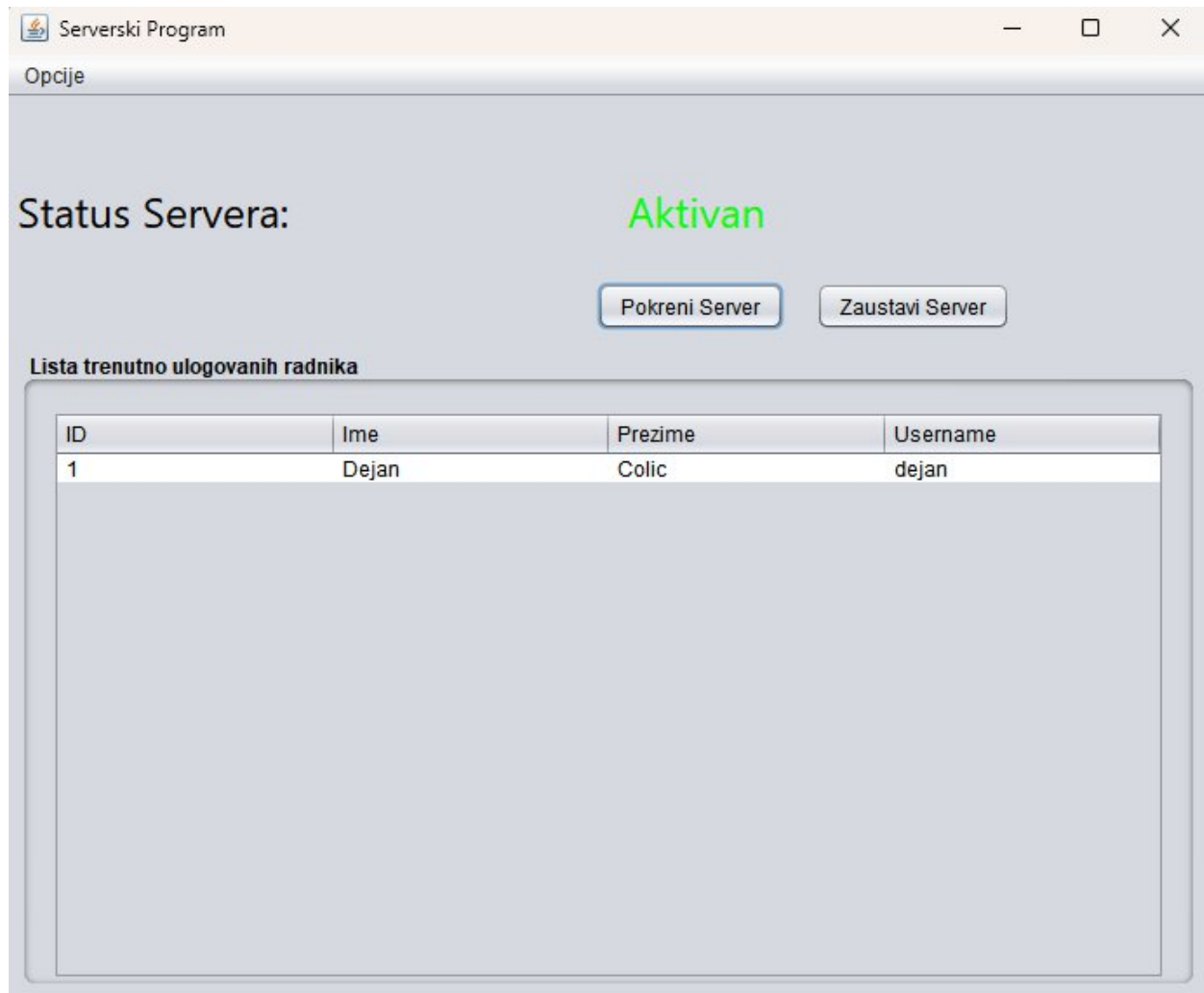
Актори СК

Радник

Учесници СК

Радник и систем (програм)

Предуслов: Систем је укључен и радник је улогован. Систем је омогућио рад са књигом.



Основни сценарио СК

1. **Радник** позива **систем** да креира нову књигу. (АПСО)
2. **Систем** креира књигу. (СО)
3. **Систем** приказује **раднику** књигу и поруку: “Систем је креирао књигу“. (ИА)
4. **Радник** уноси податке о књизи. (АПУСО)

Nova Knjiga

Knjiga

Naziv : TestKnjiga

Autor : TestAutor

Primerci

Izdavac :

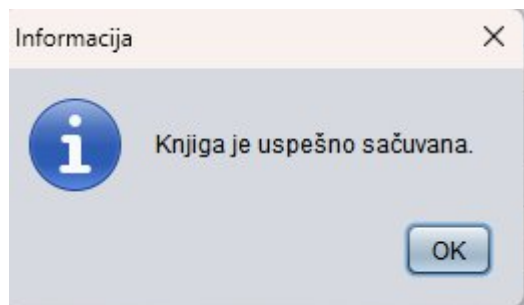
Godina izdanja :

Izdavač	Godina izdanja
TestIzdavac	2000
TestIzdavac2	2001

Sacuvaj Primerak Izmeni Primerak Obrisi Primerak

Sacuvaj Knjigu Odustani

5. **Радник** контролише да ли је тачно унео податке о књизи. (АНСО)
6. **Радник** позива **систем** да запамти податке о књизи. (АПСО)
7. **Систем** памти податке о књизи. (СО)
8. **Систем** приказује **раднику** запамћену књигу и поруку: “Систем је запамтио књигу”. (ИА)



Алтернативна сценарија

3.1 Уколико **систем** не може да креира књигу он приказује **раднику** поруку: “Систем не може да креира књигу”. Прекида се извршење сценарија. (ИА)

8.1 Уколико **систем** не може да запамти податке о књизи он приказује **раднику** поруку “Систем не може да запамти књигу”. (ИА)

5.3.2 СК2: Случај коришћења – Унос новог задужења (сложен)

Назив СК

Унос новог задужења

Актери СК

Радник

Учесници СК

Радник и **систем** (програм)

Предуслов: Систем је укључен и радник је улогован. Систем је омогућио рад са задужењима, учитана је листа књига, листа примерака, листа чланова.

Основни сценарио СК

1. **Радник** позива **систем** да креира ново задужење. (АПСО)
2. **Систем** креира ново задужење. (СО)
3. **Систем** приказује **раднику** ново задужење и поруку: "Успешно креирано ново задужење". (ИА)

Rad sa Zaduzenjima

Radnik : Dejan Colic

Datum zaduzenja :

Član : Pera Peric

Stavke Zaduzenja

PrimerakID	Izdavač	Godina	Datum razduženja	Napomena

Datum Razduzenja : (Ostaviti prazno ako knjiga nije vratena)

Knjiga : Knjiga pet prstenova

Primerak : ID 1 - Laguna 2000

Napomena :

Dodaj Stavku Izmeni Stavku Obrisi Stavku

Član	Radnik	Datum
Pera Peric	Admin Admin	30.07.2025
Pera Peric	Admin Admin	09.09.2025
Aleksandar Veliki	Dejan Colic	31.07.2025

Dodaj Zaduzenje Izmeni Zaduzenje Obrisi Zaduzenje Nazad

4. **Радник** уноси податке о задужењу. (АПУСО)
5. **Радник** позива систем да запамти податке о задужењу. (АПСО)

Rad sa Zaduzenjima

Rad sa Zaduzenjima

Radnik : Dejan Colic

Datum zaduzenja : 10.9.2025

Clan : test test

Stavke Zaduzenja

PrimerakID	Izdavač	Godina	Datum razduženja	Napomena
71	TestIzdavac	2000		test

Datum Razduzenja :

(Ostaviti prazno ako knjiga nije vracena)

Knjiga : TestKnjiga

Primerak : ID 71 - TestIzdavac 2000

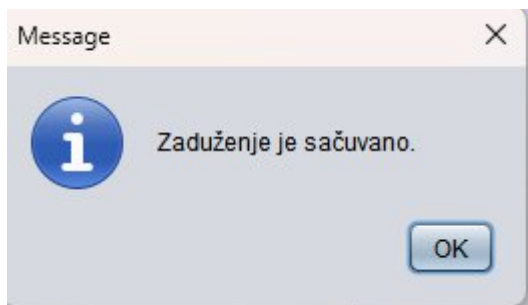
Napomena : test

Dodaj Stavku
Izmeni Stavku
Obrisi Stavku

Član	Radnik	Datum
Pera Peric	Admin Admin	30.07.2025
Pera Peric	Admin Admin	09.09.2025
Aleksandar Veliki	Dejan Colic	31.07.2025

Dodaj Zaduzenje
Izmeni Zaduzenje
Obrisi Zaduzenje
Nazad

6. **Систем** памти податке о задужењу. (CO)
7. **Систем** приказује раднику ново задужење и поруку:"Успешно запамћено ново задужење". (ИА)



Алтернативна сценарија

3.1 Уколико **систем** не може да креира задужење, приказује **раднику** поруку: "Систем не може да креира задужење". Прекида се извршење сценарија. (ИА)

7.1 Уколико **систем** не може да запамти задужење, приказује **раднику** поруку: " Систем не може да запамти задужење". (ИА)

5.3.3 СК3: Случај коришћења – Унос новог члана

Назив СК

Унос новог члана

Актори СК

Радник

Учесници СК

Радник и **систем** (програм)

Предуслов: Систем је укључен и радник је улогован. Систем је омогућио рад са члановима.

Основни сценарио СК

1. **Радник** позива **систем** да креира члана. (АПСО)
2. **Систем** креира члана. (СО)
3. **Систем** приказује **раднику** члана и поруку: "Систем је креирао члана". (ИА)

Rad sa clanovima

Rad sa clanovima

Pretrazi po imenu:

Ime	Prezime	Telefon	Email	Ima zaduženja
Pera	Peric	061123123	pera@pera.com	Da
Stefan	Krunic	062123123	stefan@gmail.com	Ne
Aleksandar	Veliki	069135135	aca@yahoo.com	Da
Petar	Petrovic	063245245	petar@gmail.com	Ne
test	test	3124235	af@asf	Da
proba	probic	123	asf@asg	Ne
asfasfttt	asfasftt	35253	asfasf@asfasf	Ne
test21	test21	123123123000	test2@test.com0	Ne

4. **Радник** уноси податке о члану. (АПУСО)
5. **Радник** контролише да ли је тачно унео податке о члану. (АНСО)
6. **Радник** позива **систем** да запамти податке о члану. (АПСО)

Dodaj ili Izmeni Clana

Clan

Ime :

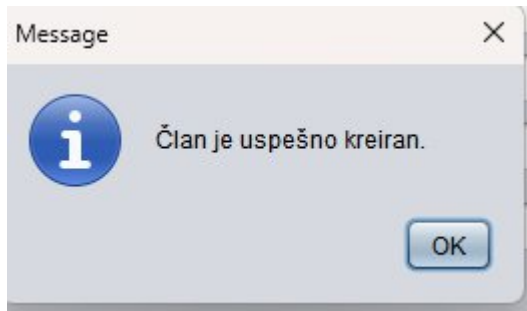
Prezime :

Telefon :

E-mail :

7. **Систем** памти податке о члану. (СО)

8. **Систем** приказује **раднику** запамћеног члана и поруку: “Систем је запамтио члана”. (ИА)



Алтернативна сценарија

- 3.1 Уколико **систем** не може да креира члана он приказује **раднику** поруку: “Систем не може да креира члана”. Прекида се извршење сценарија. (ИА)

- 8.1 Уколико **систем** не може да запамти податке о члану он приказује **раднику** поруку “Систем не може да запамти члана”. (ИА)

5.3.4 СК4: Случај коришћења – Претраживање чланова

Назив СК

Претраживање чланова

Актори СК

Радник

Учесници СК

Радник и **систем** (програм)

Предуслов: Систем је укључен и радник је улогован. Систем је омогућио рад са члановима.

Основни сценарио СК

1. **Радник** уноси вредности по којима претражује чланове. (АПУСО)
2. **Радник** позива **систем** да пронађе чланове по задатим вредностима. (АПСО)

Rad sa clanovima

Pretrazi po imenu:

Ime	Prezime	Telefon	Email	Ima zaduženja
Pera	Peric	061123123	pera@pera.com	Da
Stefan	Krunic	062123123	stefan@gmail.com	Ne
Aleksandar	Veliki	069135135	aca@yahoo.com	Da
Petar	Petrovic	063245245	petar@gmail.com	Ne
test	test	3124235	af@asf	Da
proba	probic	123	asf@asg	Ne
asfasfttt	asfasftt	35253	asfasf@asfasf	Ne
test21	test21	123123123000	test2@test.com0	Ne
TestClan0	TestClan0	1231231230	123@123.com	Ne

3. **Систем** тражи чланове по задатим критеријумима. (CO)
4. **Систем** приказује листу чланова и поруку " Пронађени су одговарајући чланови". (ИА)
5. **Радник** бира члана. (АПУСО)
6. **Радник** позива **систем** да учита податке о изабраном члану. (АПСО)
7. **Систем** учитава податке о изабраном члану. (CO)
8. **Систем** приказује податке о члану уз поруку "Успешно учитан члан". (ИА)

Rad sa clanovima

Pretrazi po imenu:

Ime	Prezime	Telefon	Email	Ima zaduženja
Stefan	Krunic	062123123	stefan@gmail.com	Ne

Алтернативна сценарија

4.1 Уколико **систем** не може да нађе чланове он приказује **раднику** поруку: “Систем не може да нађе чланове по задатим вредностима”. Прекида се извршење сценарија. (ИА)

8.1 Уколико **систем** не може да учита податке о члану он приказује **раднику** поруку: "Систем не може да учита изабраног члана". (ИА)

5.3.5 СК5: Случај коришћења – Претраживање књига

Назив СК

Претраживање књига

Актори СК

Радник

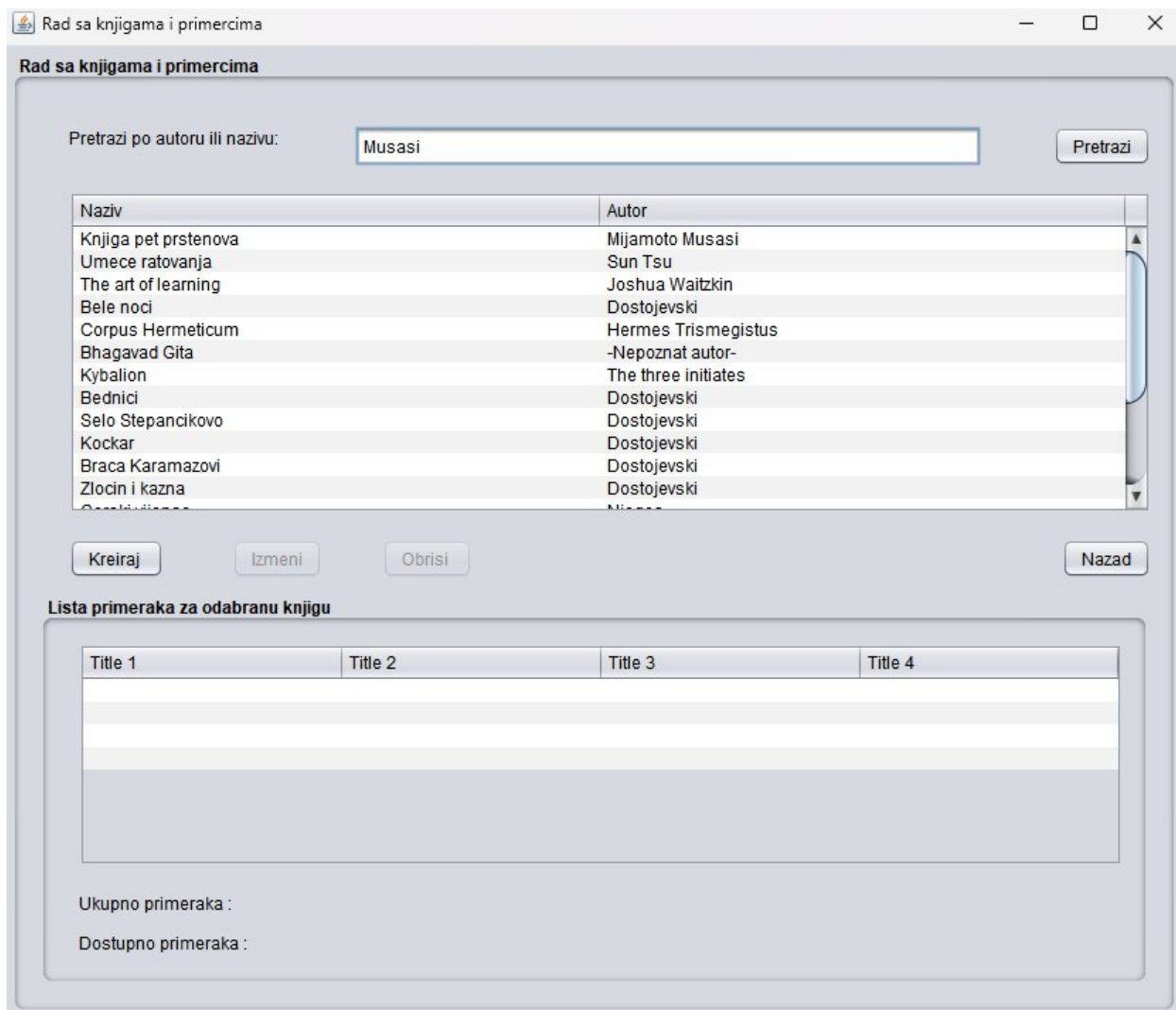
Учесници СК

Радник и **систем** (програм)

Предуслов: Систем је укључен и радник је улогован. Систем је омогућио рад са књигама.

Основни сценарио СК

1. **Радник** уноси вредности по којима претражује књиге. (АПУСО)
2. **Радник** позива **систем** да пронађе књиге по задатим вредностима. (АПСО)



3. **Систем** тражи књиге по задатим критеријумима. (CO)
4. **Систем** приказује листу књига и поруку " Пронађене су одговарајуће књиге". (ИА)

Rad sa knjigama i primercima

Pretrazi po autoru ili nazivu:

Naziv	Autor
Knjiga pet prstenova	Mijamoto Musasi

Lista primeraka za odabranu knjigu

Title 1	Title 2	Title 3	Title 4

Ukupno primeraka :
Dostupno primeraka :

5. **Радник** бира књигу. (АПУСО)
6. **Радник** позива **систем** да учита податке о изабраној књизи. (АПСО)
7. **Систем** учитава податке о изабраној књизи. (СО)
8. **Систем** приказује податке о књизи уз поруку "Успешно учитана књига". (ИА)

Rad sa knjigama i primercima

Pretrazi po autoru ili nazivu: Musasi Pretrazi

Naziv	Autor
Knjiga pet prstenova	Mijamoto Musasi

Kreiraj Izmeni Obrisi Nazad

Lista primeraka za odabranu knjigu

Izdavač	Godina izdanja	Dostupan
Laguna	2000	Da
IzdavacNEki	2005	Da

Ukupno primeraka : 2
Dostupno primeraka : 2

Алтернативна сценарија

4.1 Уколико **систем** не може да нађе књиге он приказује **раднику** поруку: “Систем не може да нађе књиге по задатим вредностима”. Прекида се извршење сценарија. (ИА)

8.1 Уколико **систем** не може да учита податке о књизи он приказује **раднику** поруку: "Систем не може да учита изабрану књигу". (ИА)

5.3.6 СК6: Случај коришћења – Претраживање задужења

Назив СК

Претраживање задужења

Актори СК

Радник

Учесници СК

Радник и систем (програм)

Предуслов: Систем је укључен и радник је улогован. Систем је омогућио рад са задужењима.

Основни сценарио СК

1. **Радник** уноси вредности по којима претражује задужења. (АПУСО)
2. **Радник** позива **систем** да пронађе задужења по задатим вредностима. (АПСО)
3. **Систем** тражи задужења по задатим критеријумима. (СО)
4. **Систем** приказује листу задужења и поруку "Пронађена су одговарајућа задужења". (ИА)
5. **Радник** бира задужење. (АПУСО)
6. **Радник** позива **систем** да учита податке о изабраном задужењу. (АПСО)
7. **Систем** учитава податке о изабраном задужењу. (СО)
8. **Систем** приказује податке о задужењу уз поруку "Успешно учитано задужење". (ИА)

Rad sa Zaduzenjima

Radnik : Dejan Colic

Datum zaduzenja : 10.09.2025

Clan : test test

Stavke Zaduzenja

PrimerakID	Izdavač	Godina	Datum razduženja	Napomena
71	Testizdavac	2000		test

Datum Razduzenja : (Ostaviti prazno ako knjiga nije vracena)

Knjiga : Knjiga pet prstenova

Primerak : ID 1 - Laguna 2000

Napomena :

Dodaj Stavku Izmeni Stavku Obris Stavku

Član	Radnik	Datum
Pera Peric	Admin Admin	30.07.2025
Pera Peric	Admin Admin	09.09.2025
Aleksandar Veliki	Dejan Colic	31.07.2025
test test	Dejan Colic	10.09.2025

Dodaj Zaduzenje Izmeni Zaduzenje Obris Zaduzenje Nazad

Алтернативна сценарија

4.1 Уколико **систем** не може да нађе задужења он приказује **раднику** поруку: “Систем не може да нађе задужења по задатим вредностима”. Прекида се извршење сценарија. (ИА)

8.1 Уколико **систем** не може да учита податке о задужењу он приказује **раднику** поруку: "Систем не може да учита изабрано задужење". (ИА)

5.3.7 СК7: Случај коришћења – Измена члана

Назив СК

Измена члана

Актори СК

Радник

Учесници СК

Радник и **систем** (програм)

Предуслов: Систем је укључен и радник је улогован. Систем је омогућио рад са члановима. Учитана је листа чланова.

Основни сценарио СК

1. **Радник** уноси вредности по којима претражује чланове. (АПУСО)
2. **Радник** позива **систем** да пронађе чланове по задатим вредностима. (АПСО)
3. **Систем** тражи чланове по задатим критеријумима. (СО)
4. **Систем** приказује листу чланова и поруку " Пронађени су одговарајући чланови". (ИА)

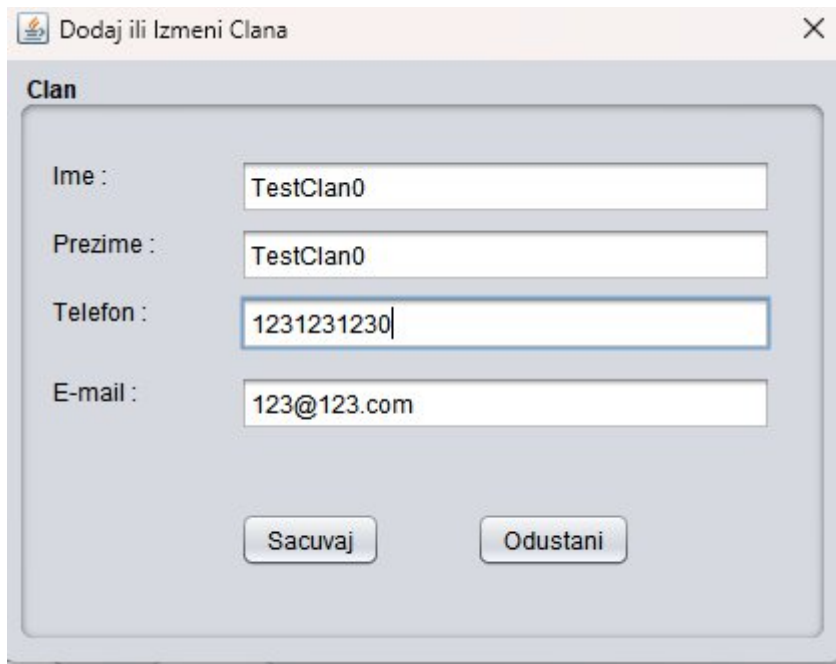
Rad sa clanovima

Rad sa clanovima

Pretrazi po imenu:

Ime	Prezime	Telefon	Email	Ima zaduženja
Pera	Peric	061123123	pera@pera.com	Da
Stefan	Krunic	062123123	stefan@gmail.com	Ne
Aleksandar	Veliki	069135135	aca@yahoo.com	Da
Petar	Petrovic	063245245	petar@gmail.com	Ne
test	test	3124235	af@asf	Da
proba	probic	123	asf@asg	Ne
asfasfttt	asfasftt	35253	asfasf@asfasf	Ne
test21	test21	123123123000	test2@test.com0	Ne
TestClan	TestClan	123123123	123@123.com	Ne

5. **Радник** бира члана из листе чланова. (АПУСО)
6. **Радник** позива **систем** да учита податке о изабраном члану. (АПСО)
7. **Систем** учитава податке о изабраном члану. (СО)
8. **Систем** приказује податке о члану уз поруку "Успешно учитан члан". (ИА)
9. **Радник** мења податке о члану. (АПУСО)
10. **Радник** проверава да ли је тачно унео податке. (АНСО)
11. **Радник** позива **систем** да запамти нове податке о члану. (АПСО)



Dodaj ili Izmeni Clana

Clan

Ime : TestClan0

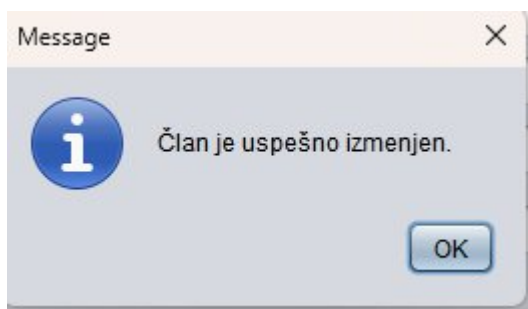
Prezime : TestClan0

Telefon : 1231231230

E-mail : 123@123.com

Sacuvaj Odustani

12. Систем памти измењене податке о члану. (CO)
13. Систем приказује раднику поруку: "Подаци успешно измењени". (ИА)



Алтернативна сценарија

- 4.1 Уколико систем не може да нађе члана он приказује раднику поруку: "Систем не може да нађе члана по задатој вредности". Прекида се извршење сценарија. (ИА)
- 8.1 Уколико систем не може да пронађе податке о члану, систем приказује раднику поруку: "Систем не може да пронађе податке о члану". Прекида се извршење сценарија. (ИА)
- 13.1 Уколико систем не може да сачува измене, приказује раднику поруку: "Систем не може да измени члана". (ИА)

5.3.8 СК8: Случај коришћења – Пријављивање радника

Назив СК

Пријављивање радника

Актори СК

Радник

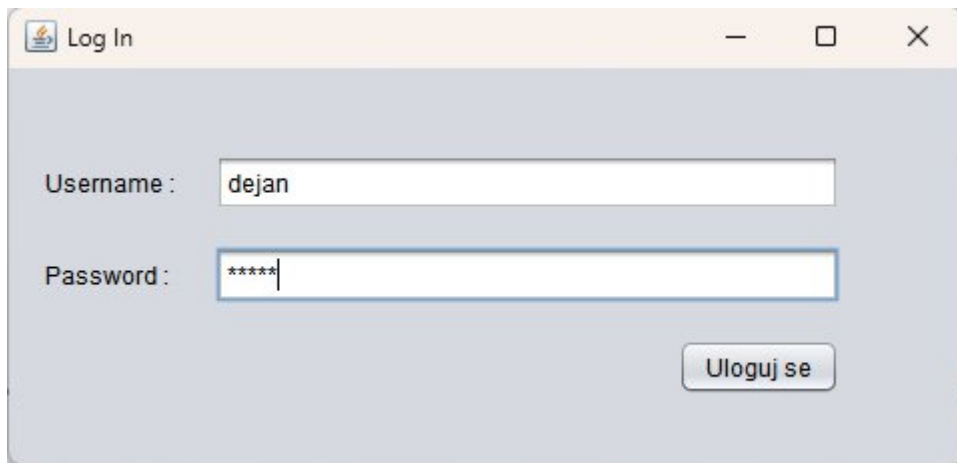
Учесници СК

Радник и систем (програм)

Предуслов: Систем је укључен и омогућава пријављивање радника.

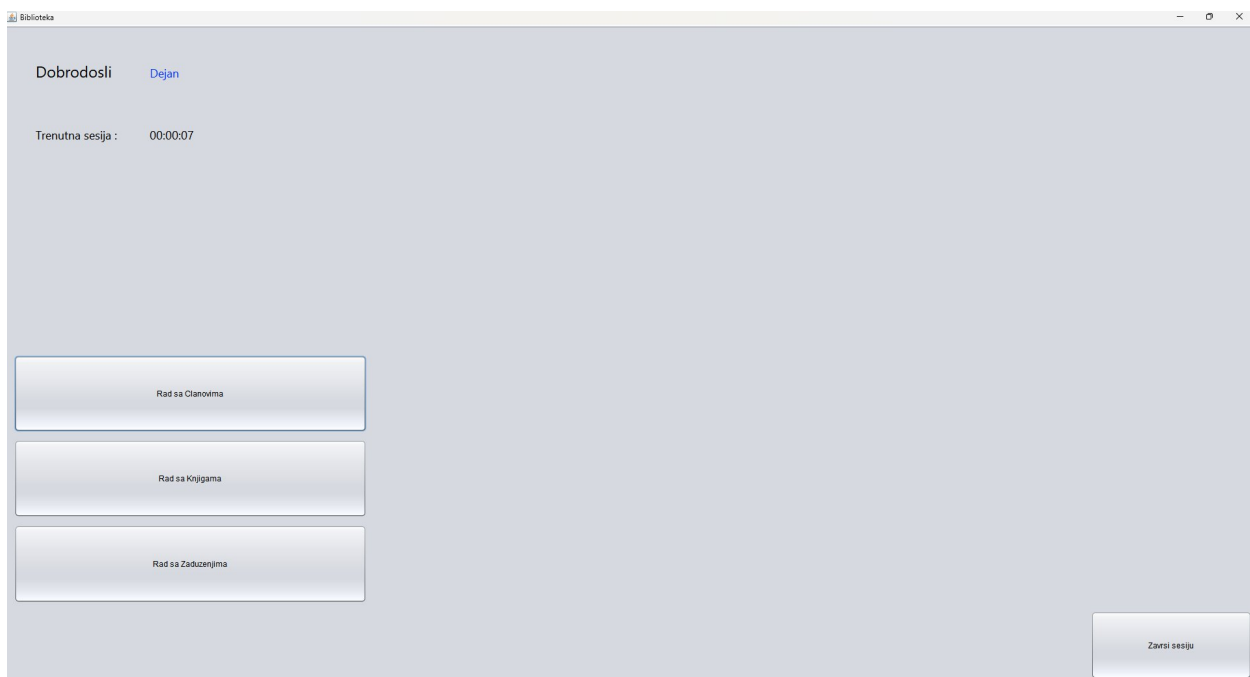
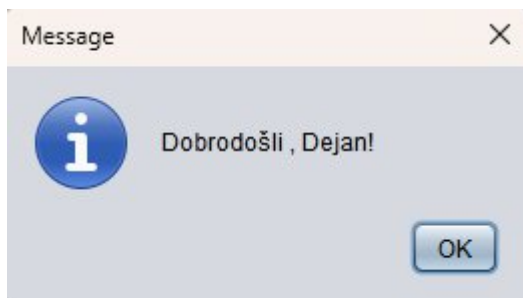
Основни сценарио СК

1. Радник уноси податке за аутентификацију радника. (АПУСО)
2. Радник позива систем да пронађе радника са задатим подацима (АПСО)



The image shows a screenshot of a web application window titled "Log In". The window has a light gray background and a white border. Inside, there are two input fields: "Username :" with the text "dejan" and "Password :" with masked characters "*****". Below the password field is a button labeled "Uloguj se". The window has standard OS window controls (minimize, maximize, close) in the top right corner.

3. Систем претражује раднике (СО)
4. Систем приказује поруку “Радник је успешно пријављен” и почетну форму (ИА)



Алтернативна сценарија

4.1 Уколико **систем** не може да нађе **радника** он приказује поруку: “Систем не може да нађе радника на основу унетих вредности”.(ИА)

5.4 Пројектовање контролера корисничког интерфејса

Контролер корисничког интерфејса је одговоран за:

- прихватање графичких објеката од екранске форме,
- конвертовање података који се налазе у графичким објектима у доменске објекте који ће бити прослеђени преко мреже до апликационог сервера,
- конвертовање доменских објеката у графичке објекте и прослеђује их до екранске форме.

5.5 Пројектовање апликационе логике

Апликациони сервери су одговорни да обезбеде сервисе који ће да омогуће реализацију апликационе логике софтверског система. Пројектовани апликациони сервер садржи:

- део за комуникацију са клијентима,
- контролер апликационе логике,
- део за комуникацију са складиштем података (брокер базе података),
- део који садржи пословну логику.

5.5.1. Комуникација са клијентима

Део за комуникацију подиже серверски сокет који ће да ослушкује мрежу. Када клијентски сокет успостави конекцију са серверским сокетом, тада сервер генерише нит која ће успоставити двосмерну везу са клијентом.

Слање и примање података од клијента се обавља разменом објеката класе Request и Response и остварује се преко сокета.

Клијент шаље захтев за извршење неке од системских операција до одговарајуће нити која је повезана са тим клијентом. Та нит прихвата захтев и прослеђује га до контролера апликационе логике. Након извршења системске операције, резултат се преко контролера апликационе логике враћа до нити клијента која тај резултат шаље назад до клијента.

5.5.2. Контролер апликационе логике

Контролер апликационе логике прихвата захтев за извршење системске операције од нити клијента и даље га преусмерава до класа које су одговорне за извршење системских операција. Након извршења системске операције контролер апликационе логике прихвата резултат и прослеђује га позиваоцу (нити клијента).

5.5.3 Пословна логика

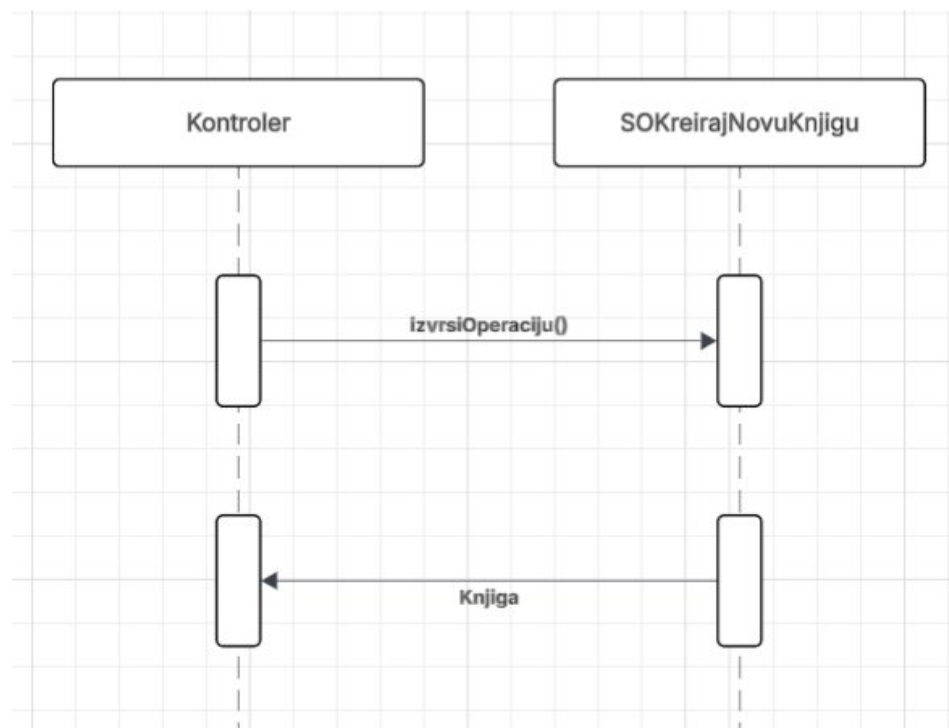
Уговор УГ1

Операција: **kreirajKnjigu(Knjiga):Signal;**

Веза са СК: СК1

Предуслови: Структурна и вредносна ограничења над **Knjiga** објектом морају бити задовољена.

Постуслови: Направљена је нова књига.



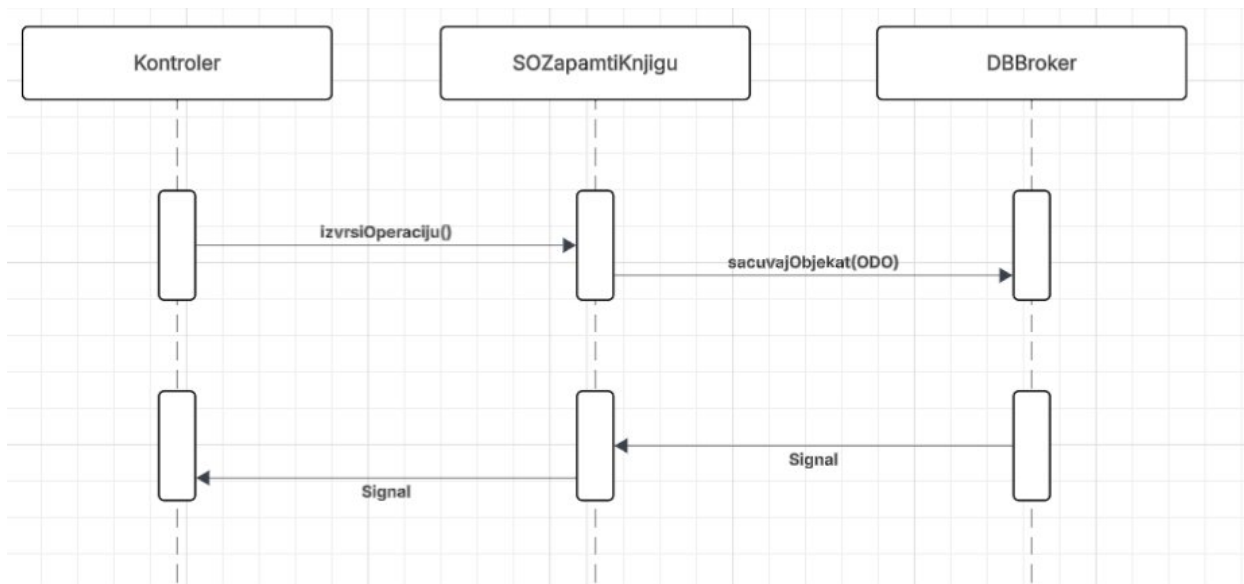
Уговор УГ2

Операција: **zapamtiKnjigu(Knjiga):Signal;**

Веза са СК: СК1

Предуслови: Структурна и вредносна ограничења над Knjiga објектом морају бити задовољена.

Постуслови: Књига је запамћена.



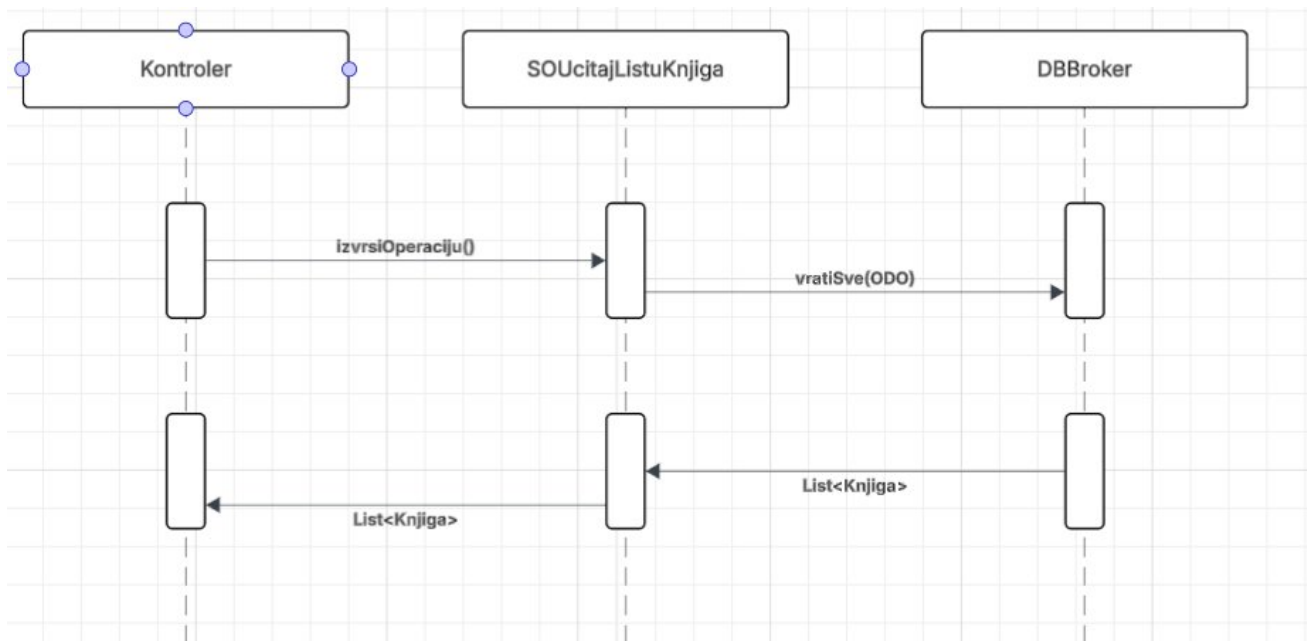
Уговор УГ3

Операција **ucitajListuKnjiga(List<Knjiga>):Signal;**

Веза са СК: СК2

Предуслови: /

Постуслови: /



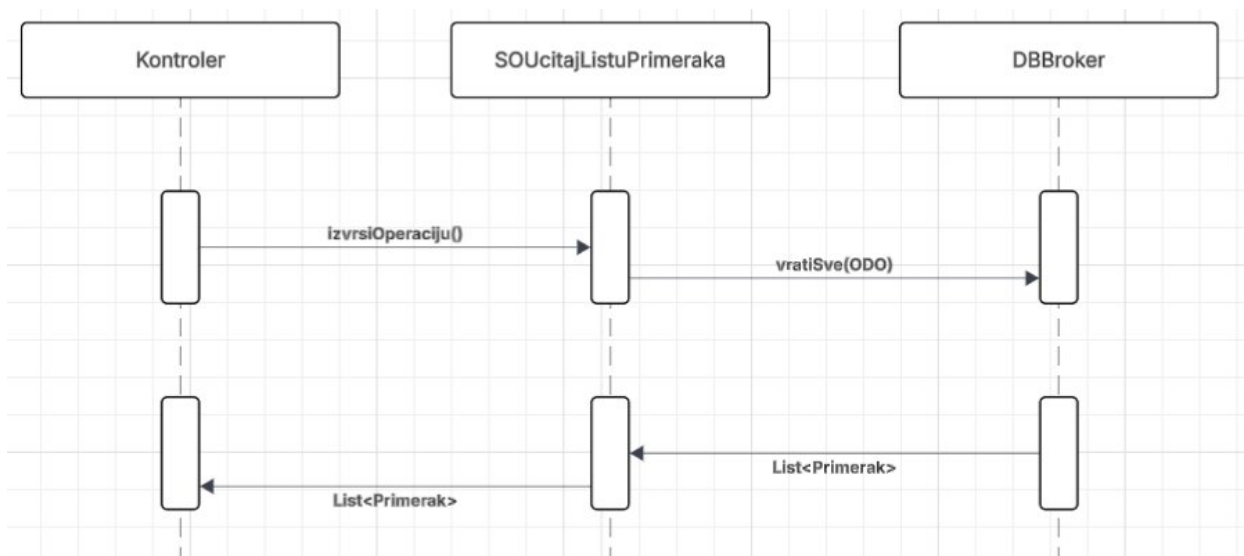
Уговор УГ4

Операција: **ucitajListuPrimeraka(List<Primerak>):Signal;**

Веза са СК: СК2

Предуслови: /

Постуслови: /



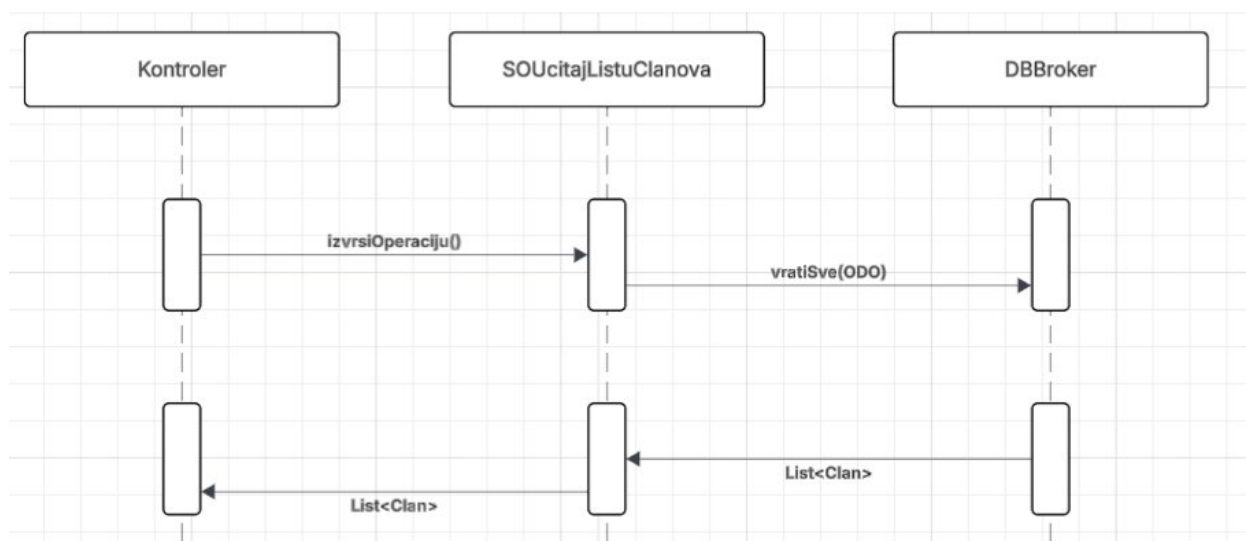
Уговор УГ5

Операција: **ucitajListuClanova(List<Clan>):Signal;**

Веза са СК: СК2, СК7

Предуслови: /

Постуслови: /



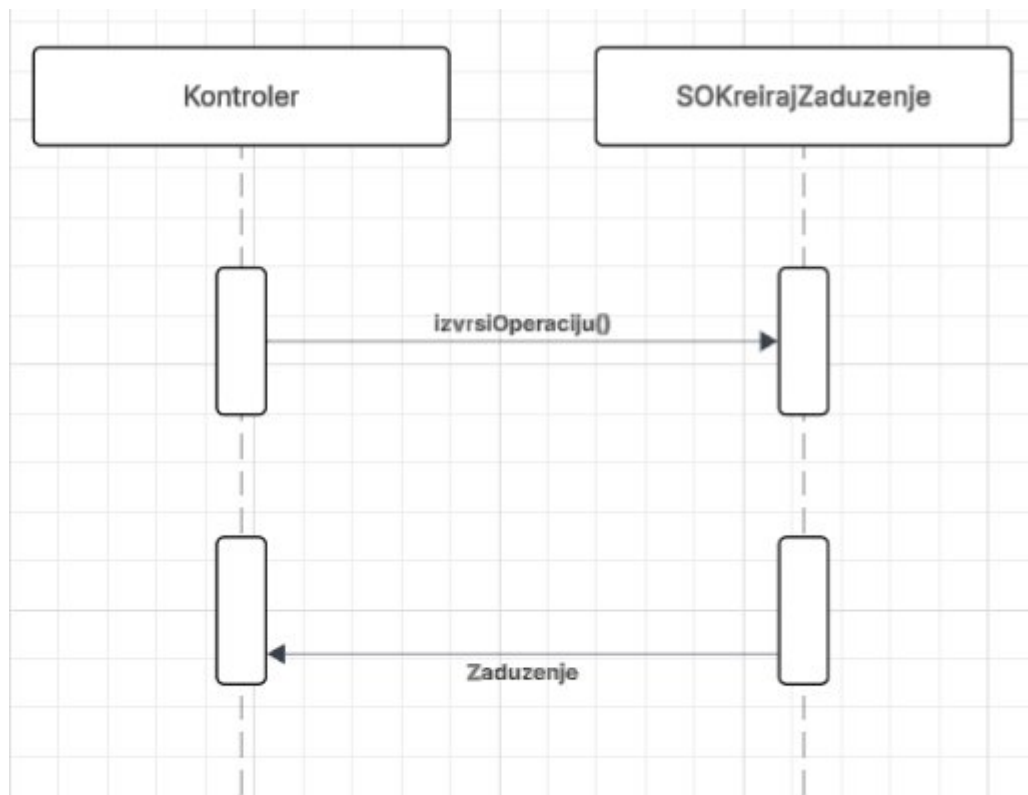
Уговор УГ6

Операција: **kreirajZaduzenje(Zaduzenje):Signal;**

Веза са СК: СК2

Предуслови: Структурна и вредносна ограничења над *Zaduzenje* објектом морају бити задовољена.

Постуслови: Направљено је ново задужење.



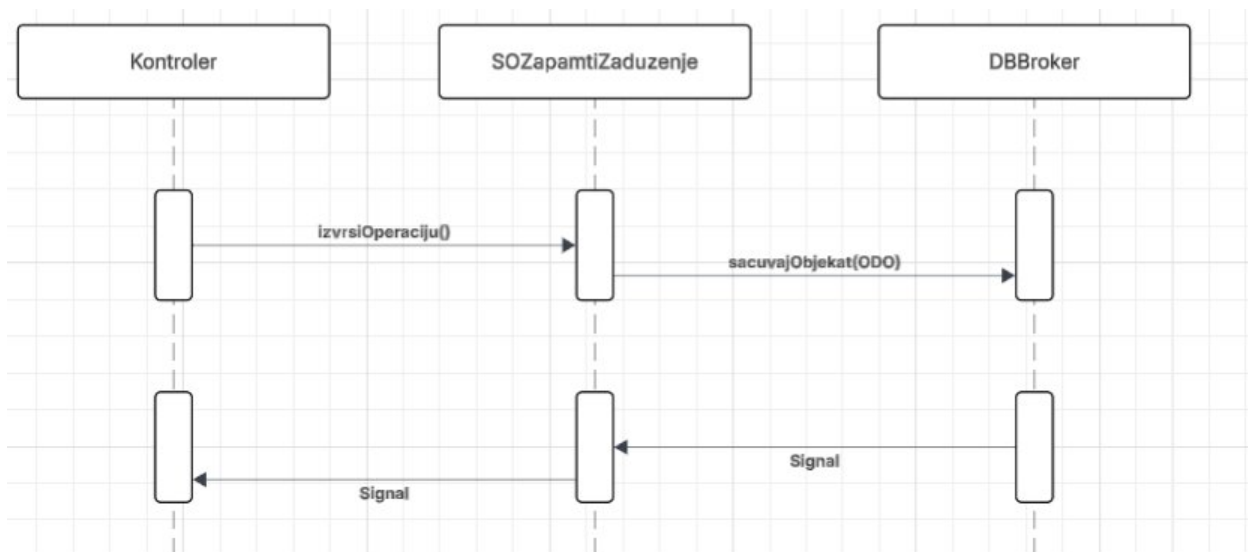
Уговор УГ7

Операција: **zapamtiZaduzenje(Zaduzenje) :Signal;**

Веза са СК: СК2

Предуслови: Структурна и вредносна ограничења над *Zaduzenje* објектом морају бити задовољена.

Постуслови: Задужење је запамћено.



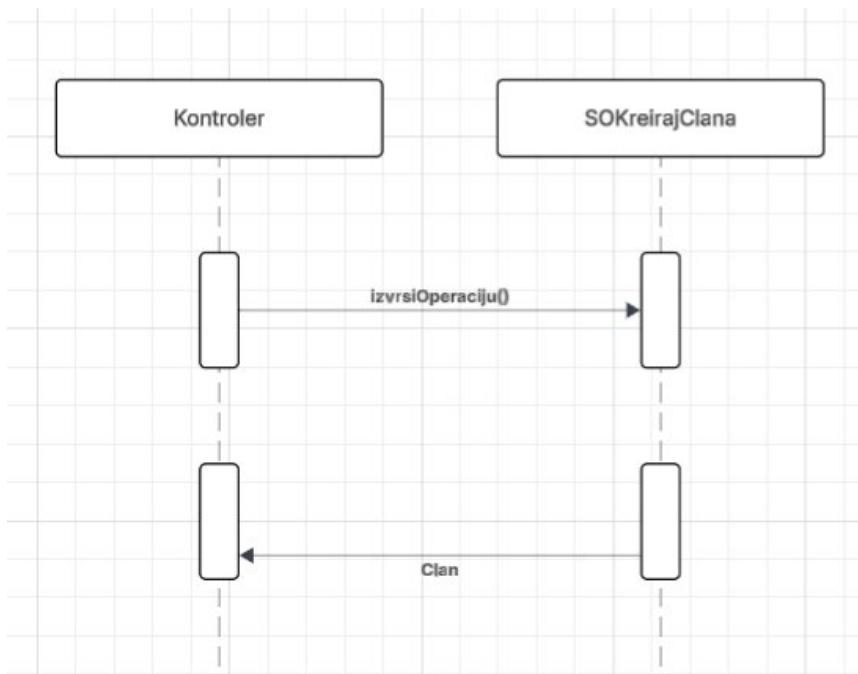
Уговор УГ8

Операција: **kreirajClana(Clan) :Signal;**

Веза са СК: СК3

Предуслови: Структурна и вредносна ограничења над Clan објектом морају бити задовољена.

Постуслови: Направљен је нови члан.



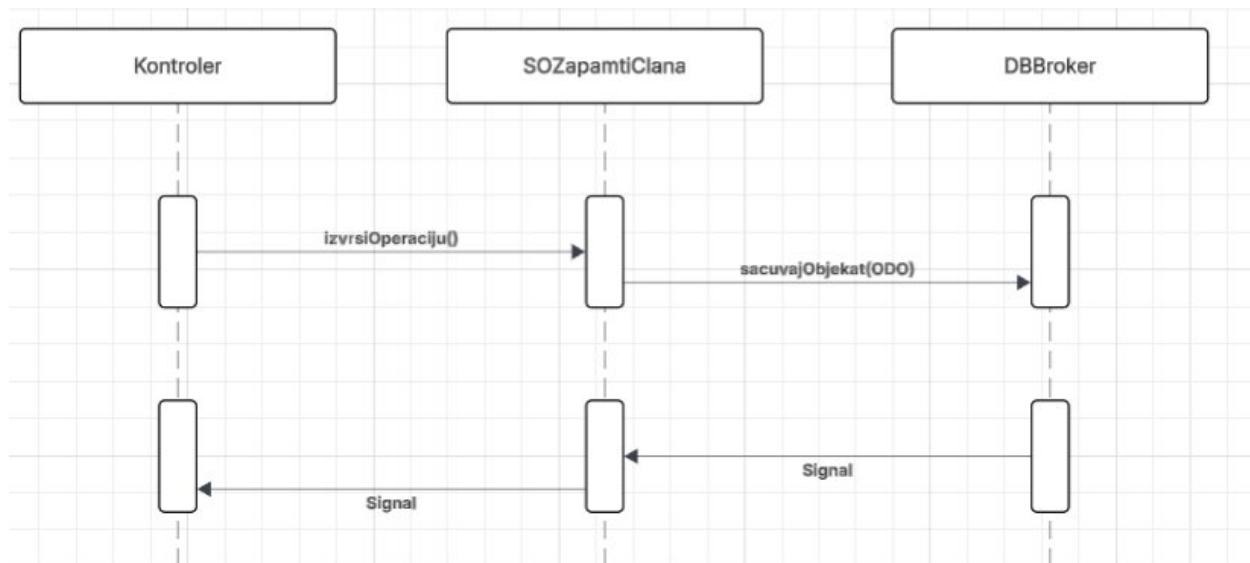
Уговор УГ9

Операција: **zapamtiClana(Clan):Signal;**

Веза са СК: СК3,СК7

Предуслови: Структурна и вредносна ограничења над Clan објектом морају бити задовољена.

Постуслови: Члан је запамћен.



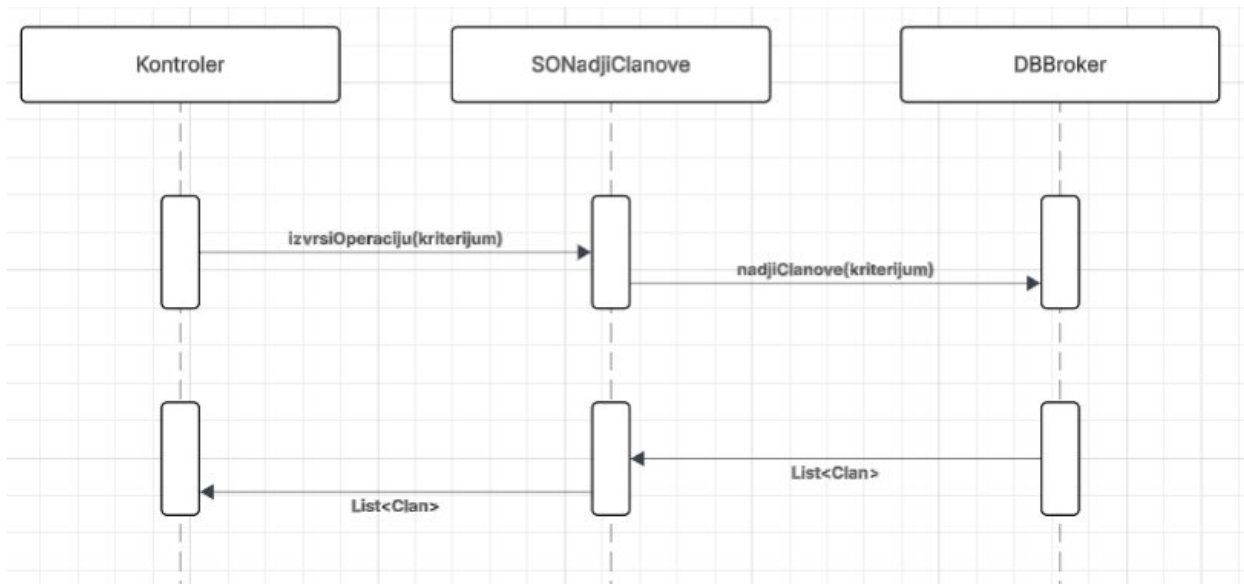
Уговор УГ10

Операција: **nadjiClanove(List<Clan>) :Signal;**

Веза са СК: СК4

Предуслови: /

Постуслови: Пронађен је тражени Члан.



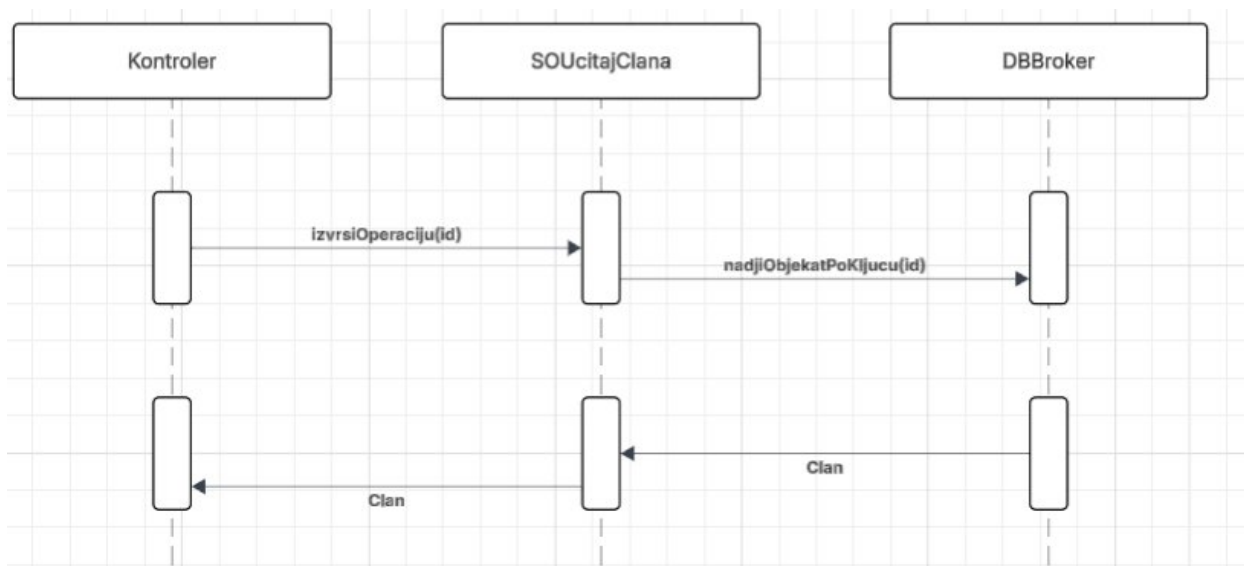
Уговор УГ11

Операција: **ucitajClana(Clan) :Signal;**

Веза са СК: СК4, СК7

Предуслови: /

Постуслови: /



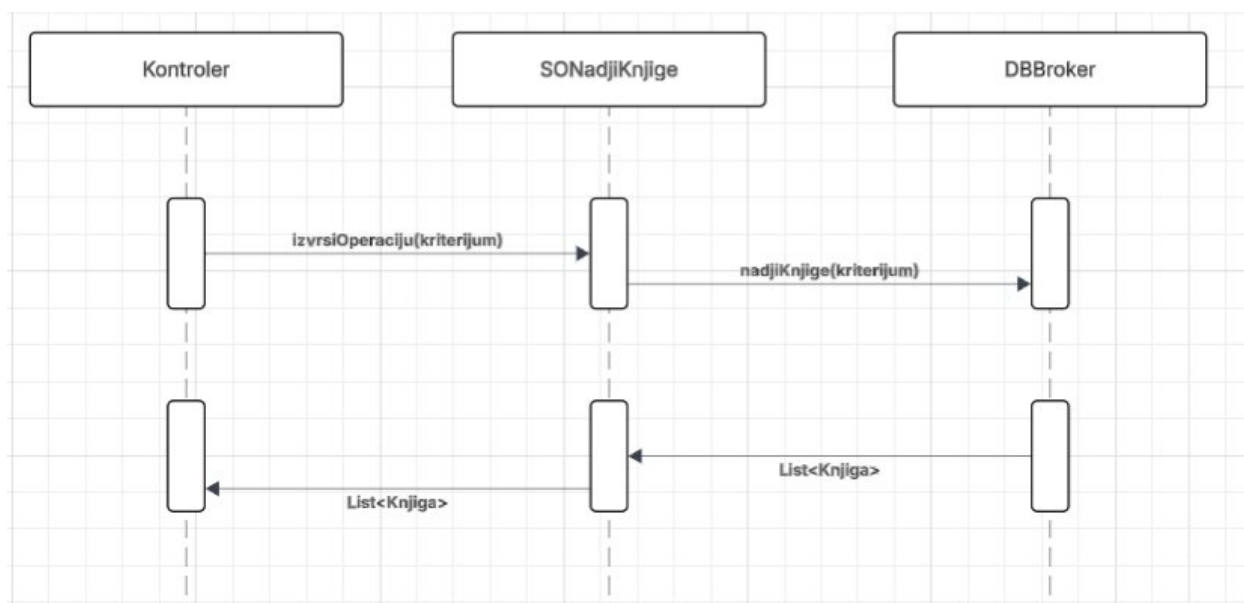
Уговор УГ12

Операција: **nadjiKnjige(List<Knjiga>) :Signal;**

Веза са СК: СК5

Предуслови: /

Постуслови: Пронађена је тражена Књига.



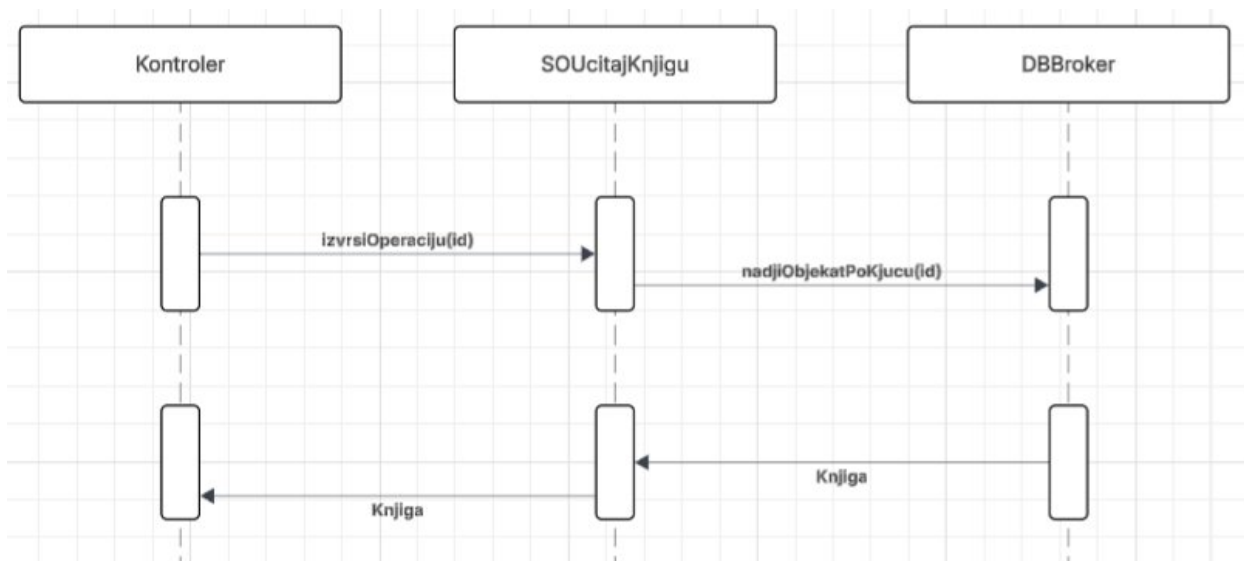
Уговор УГ13

Операција: **ucitajKnjigu(Knjiga):Signal;**

Веза са СК: СК5

Предуслови: /

Постуслови: /



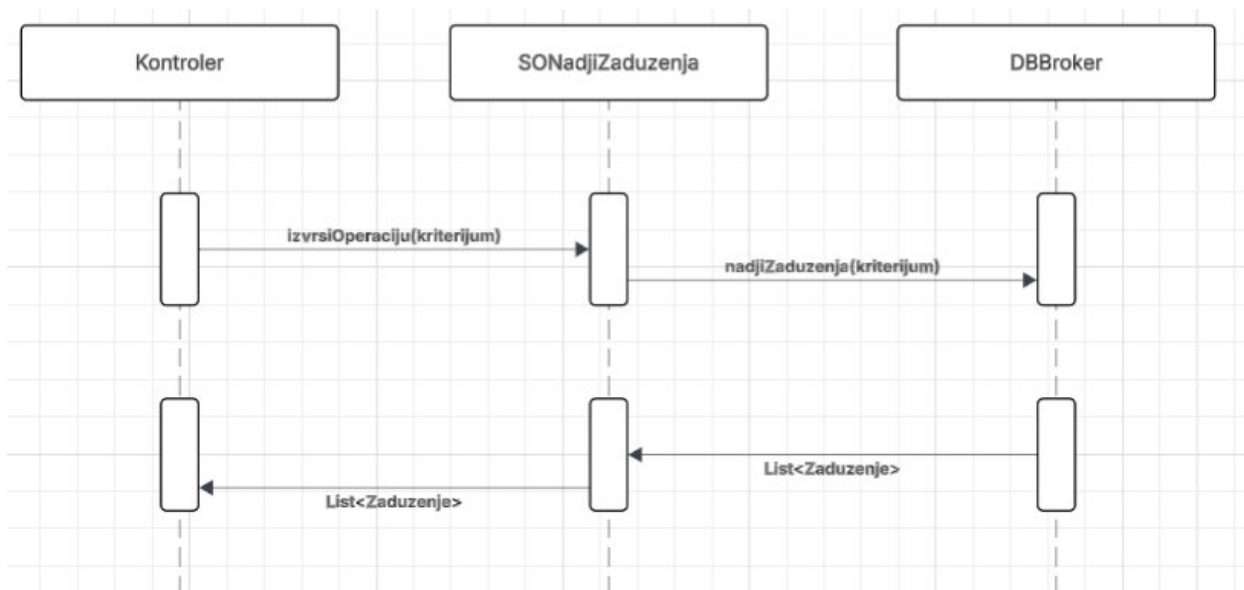
Уговор УГ14

Операција: **nadjiZaduzenja(List<Zaduzenje>):Signal;**

Веза са СК:СК6

Предуслови: /

Постуслови: Пронађено је тражено задужење.



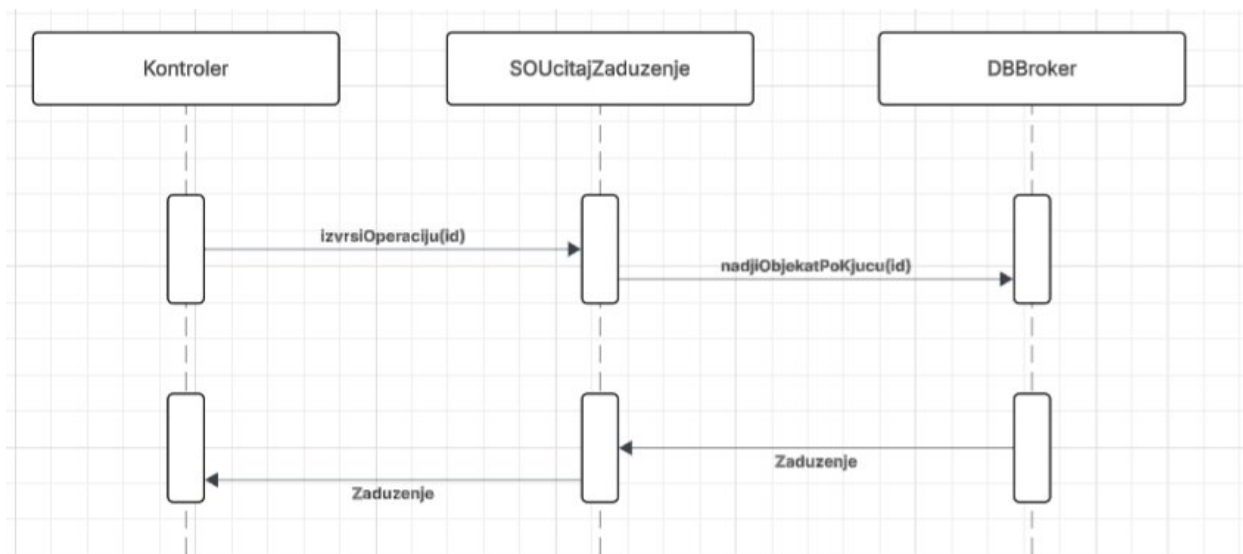
Уговор УГ15

Операција: **ucitajZaduzenje(Zaduzenje):Signal;**

Веза са СК: СК6

Предуслови: /

Постуслови: /



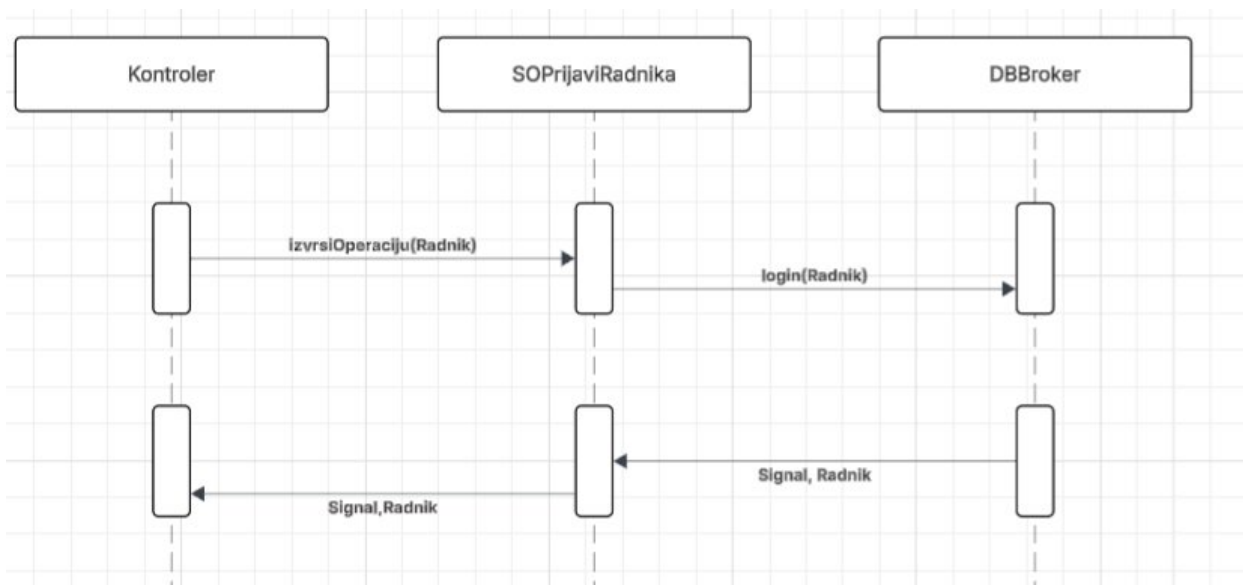
Уговор УГ16

Операција: **prijaviRadnika(Radnik):Signal;**

Веза са СК: СК8

Предуслови: Структурна и вредносна ограничења над објектом Radnik морају бити задовољена. Радник са истим крeдинцијалима не може већ бити улогован.

Постуслови: Радник је улогован на систем.



5.5.4. Пројектовање структуре софтверског система

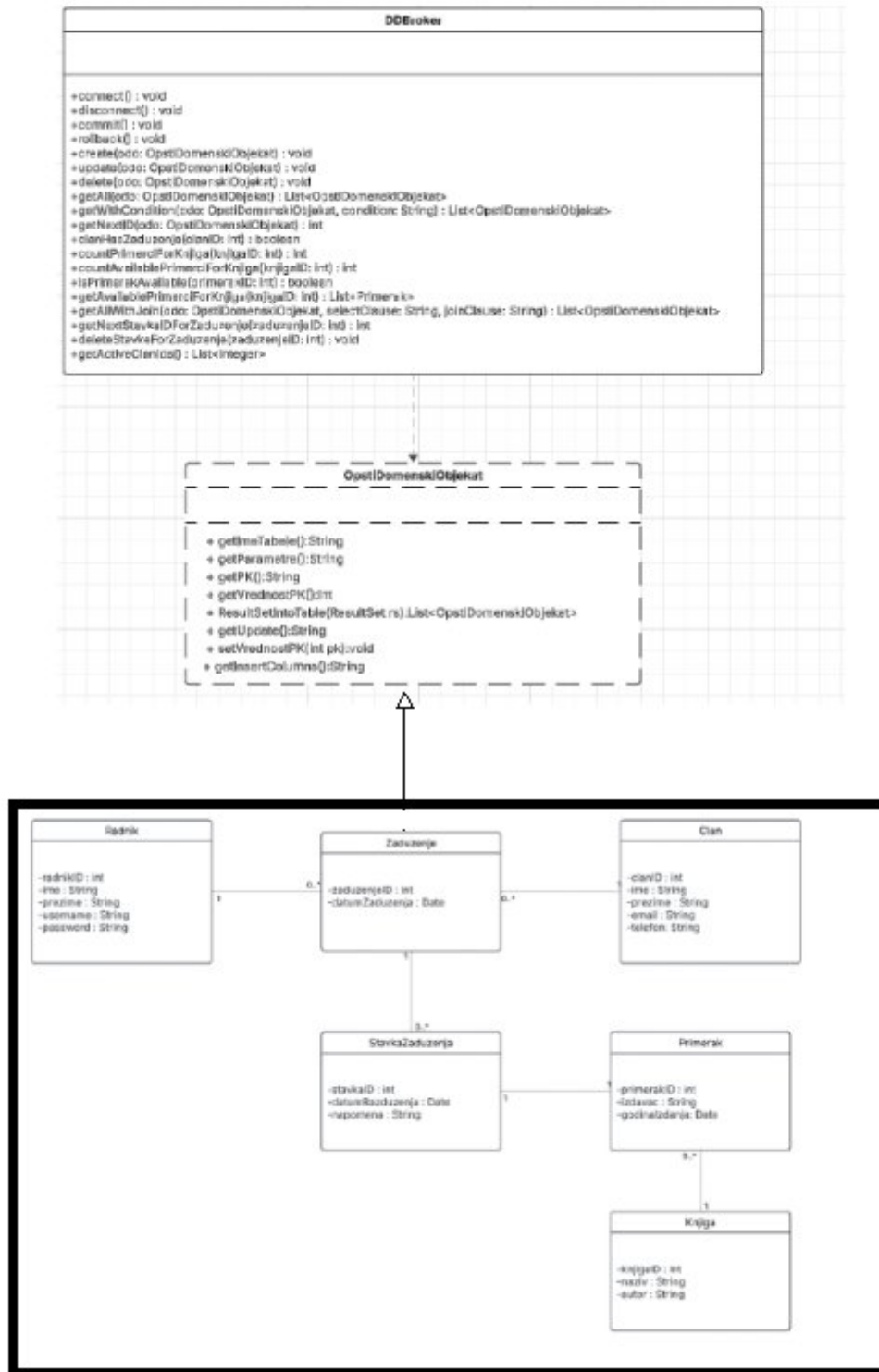
На основу концептуалних класа праве се софтверске класе структуре. Свака класа има приватна поља атрибута, гет-ере и сет-ере за атрибуте, безпараметарски и параметарски конструктор.

5.5.5. Брокер базе података

Класа DBBroker се приказује како би се обезбдио презистентни сервис објектима доменских класа које се чувају у бази података. Тако ова класа представља оквир који посредује у свим операцијама над базом података и реализује следеће методе:

```
public void connect()  
public void disconnect()  
public void commit()  
public void rollback()  
public void create(OpstiDomenskiObjekat odo)  
public void update(OpstiDomenskiObjekat odo)  
public void delete(OpstiDomenskiObjekat odo)  
public List<OpstiDomenskiObjekat> getAll(OpstiDomenskiObjekat odo)  
public List<OpstiDomenskiObjekat> getWithCondition(OpstiDomenskiObjekat odo,  
String condition)  
public int getNextID(OpstiDomenskiObjekat odo)  
public boolean clanHasZaduzenja(int clanID)  
public int countPrimeraciForKnjiga(int knjigaID)  
public int countAvailablePrimeraciForKnjiga(int knjigaID)  
public boolean isPrimerakAvailable(int primerakID)  
public List<Primerak> getAvailablePrimeraciForKnjiga(int knjigaID)  
public List<OpstiDomenskiObjekat> getAllWithJoin(OpstiDomenskiObjekat odo, String  
selectClause, String joinClause)  
public int getNextStavkaIDForZaduzenje(int zaduzenjeID)
```

```
public void deleteStavkeForZaduzenje(int zaduzenjeID)  
public List<Integer> getActiveClanIds()
```



Слика 28: Веза општег доменског објекта са брокером базе података

5.6 Пројектовање складишта података

На основу софтверских класа структуре пројектоване су табеле (складишта података) релационог система за управљање базом података (MySQL).

Табела Члан:

<input type="checkbox"/>	Column Name	Data Type	Length
<input type="checkbox"/>	clanID	int	
<input type="checkbox"/>	ime	varchar	255
<input type="checkbox"/>	prezime	varchar	255
<input type="checkbox"/>	telefon	varchar	255
<input type="checkbox"/>	email	varchar	255
<input type="checkbox"/>			

Табела Књига:

<input type="checkbox"/>	Column Name	Data Type	Length
<input type="checkbox"/>	knjigaID	int	
<input type="checkbox"/>	naziv	varchar	255
<input type="checkbox"/>	autor	varchar	255
<input type="checkbox"/>			

Табела Примерак:

<input type="checkbox"/>	Column Name	Data Type	Length
<input type="checkbox"/>	primerakID	int	
<input type="checkbox"/>	izdavac	varchar	255
<input type="checkbox"/>	godinaIzdanja	year	
<input type="checkbox"/>	knjigaID	int	
<input type="checkbox"/>			

Табела Радник:

<input type="checkbox"/>	Column Name	Data Type	Length
<input type="checkbox"/>	radnikID	int	
<input type="checkbox"/>	ime	varchar	255
<input type="checkbox"/>	prezime	varchar	255
<input type="checkbox"/>	username	varchar	255
<input type="checkbox"/>	password	varchar	255
<input type="checkbox"/>			

Табела Задужење:

<input type="checkbox"/>	Column Name	Data Type	Length
<input type="checkbox"/>	zaduzenjeID	int	
<input type="checkbox"/>	datumZaduzenja	date	
<input type="checkbox"/>	radnikID	int	
<input type="checkbox"/>	clanID	int	
<input type="checkbox"/>			

Табела СтавкаРаздужења:

<input type="checkbox"/>	Column Name	Data Type	Length
<input type="checkbox"/>	zaduzenjeID	int	
<input type="checkbox"/>	stavkaID	int	
<input type="checkbox"/>	datumRazduzenja	date	
<input type="checkbox"/>	napomena	varchar	255
<input type="checkbox"/>	primerakID	int	
<input type="checkbox"/>			

6. Имплементација

Софтверски систем је развијен у програмском језику Java. Систем је пројектован као клијент – сервер апликација. Коришћен је MySQL систем за управљање базом података, а NetBeans IDE 26 је коришћен као развојно окружење.

На основу архитектуре софтверског система добијени су следећи пројекти и класе:

6.1 Client projekat:

- communication/CommunicationWithServer.java
- controller/UIController.java
- models/TableModelClan.java
- models/TableModelKnjiga.java
- models/TableModelPrimerak.java
- models/TableModelPrimerakTemp.java
- models/TableModelStavkaZaduzenja.java
- models/TableModelZaduzenje.java
- ui/FormaLogin.java
- ui/FormaMain.java
- ui/clanovi/DijalogNoviClan.java
- ui/clanovi/FormaClanovi.java
- ui/knjige/DijalogIzmeniPrimerak.java
- ui/knjige/DijalogNovaKnjiga.java
- ui/knjige/FormaKnjige.java
- ui/zaduzenja/FormaZaduzenje.java

6.2 Server projekat:

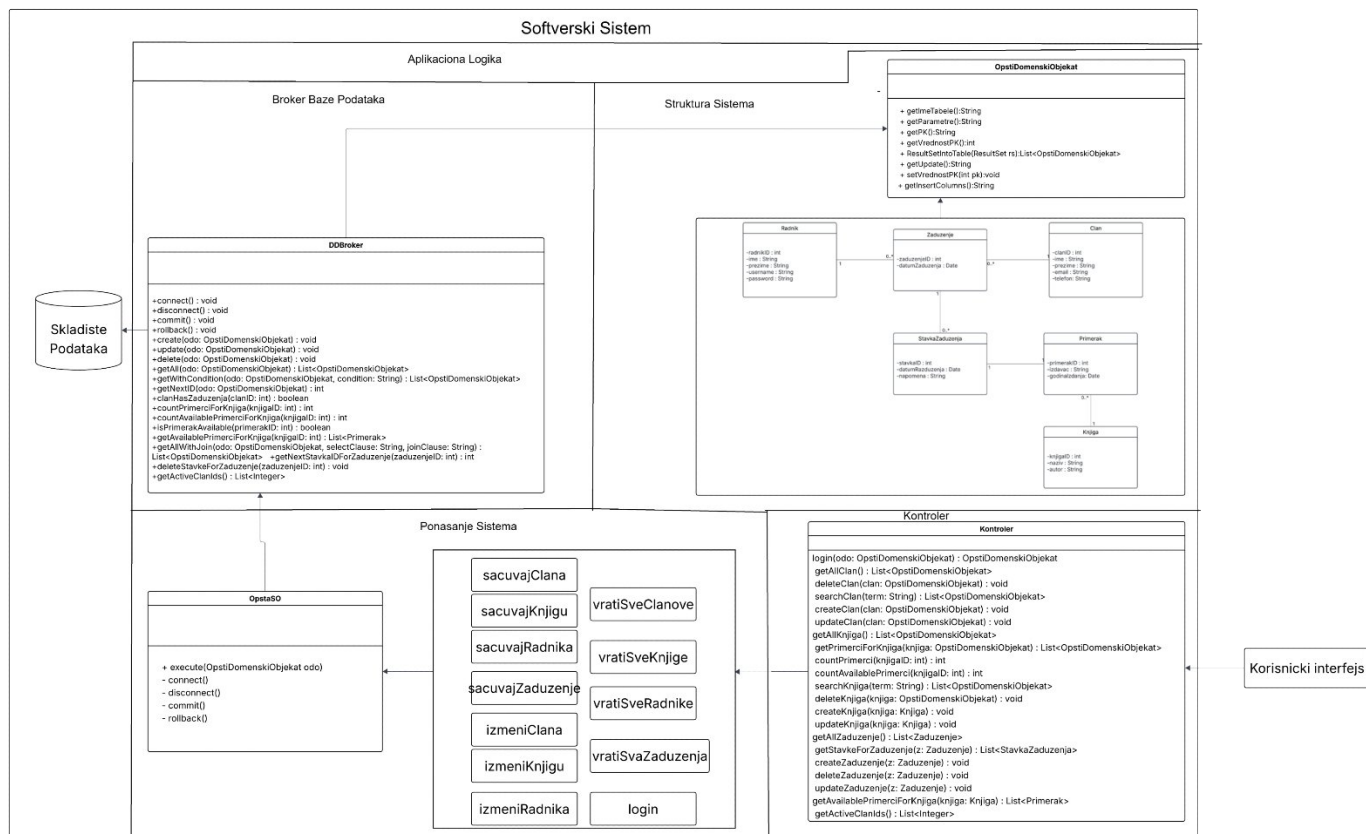
- controller/Controller.java
- db/DBBroker.java
- db/DatabaseUtil.java
- models/TableModelRadnik.java
- so/GenericSO.java

- so/clan/SOcreateClan.java
- so/clan/SOdeleteClan.java
- so/clan/SOgetAllClan.java
- so/clan/SOsearchClan.java
- so/clan/SOupdateClan.java
- so/knjiga/SOcountAvailablePrimerici.java
- so/knjiga/SOcountPrimerici.java
- so/knjiga/SOcreateKnjiga.java
- so/knjiga/SOdeleteKnjiga.java
- so/knjiga/getAllKnjiga.java
- so/knjiga/SOgetAvailablePrimericiForKnjiga.java
- so/knjiga/SOgetPrimericiForKnjiga.java
- so/knjiga/SOsearchKnjiga.java
- so/knjiga/SOupdateKnjiga.java
- so/login/SOlogin.java
- so/zaduzenje/SOcreateZaduzenje.java
- so/zaduzenje/SOdeleteZaduzenje.java
- so/zaduzenje/SOgetActiveZaduzenje.java
- so/zaduzenje/SOgetAllZaduzenje.java
- so/zaduzenje/SOgetStavkeForZaduzenje.java
- so/zaduzenje/SOupdateZaduzenje.java
- threads/ClientServiceThread.java
- threads/StartServerThread.java
- ui/DialogInfo.java
- ui/DialogLogs.java

- ui/FormServerConfig.java
- ui/FormServerMain.java
- util/ServerLogger.java

6.3 Common projekat:

- communication/Request.java
- communication/Response.java
- constants/Constants.java
- constants/Operations.java
- domain/OpstiDomenskiObjekat.java
- domain/Clan.java
- domain/Knjiga.java
- domain/Primerak.java
- domain/Radnik.java
- domain/Zaduzenje.java
- domain/StavkaZaduzenja.java



Слика 29 : Архитектура Система

7. Тестирање

Сваки од имплементираних случајева коришћења је тестиран. Приликом тестирања сваког случаја коришћења поред правилно унетих података, уношени су и неправилни подаци како би се утврдило какав ће бити резултат извршења. На основу извршених тестирања отклоњени су уочени недостаци.

Нажалост рађено је само функционално, мануелно тестирање, аутоматско тестирање је ван оквира овог предмета.

8. Литература

Др Синиша Влајић, Пројектовање софтвера (скрипта), Београд, 2015.