

Универзитет у Београду
Факултет организационих наука
Катедра за софтверско инжењерство



Семинарски рад из предмета
Пројектовање софтвера

**Софтверски систем за дечију ординацију у Јава
окружењу**

Ментор: проф. др Синиша Влајић

Студент: **Огњен Ковачевић**

Број индекса: **2022/0091**

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	35
3.4 Структура софтверског система – Концептуални (доменски) модел	37
3.5 Структура софтверског система – Релациони модел	38
3.6 Табела структурних и вредносних ограничења релационог модела	39
4. Пројектовање	43
4.1 Пројектовање корисничког интерфејса	44
4.2 Пројектовање апликационе логике	62
4.3 Пројектовање складишта података	72
5. Имплементација	76
6. Тестирање	79
7. Закључак	80
Литература	81

1. Увод

Софтверске апликације све чешће налазе примену у решавању конкретних, свакодневних проблема. У овом раду приказано је решење за рад дечије ординације, са циљем да се на једноставан начин воде подаци о деци, лекаrima, лековима и рецептима. Поред основне евиденције, систем омогућава и рад са специфичностима као што су школска и предшколска деца, као и специјализације доктора.

Програм је подељен на клијентски и серверски део, а кроз различите форме и контролере омогућено је додавање, измена и приказ података, као и комуникација између корисника и система. Циљ овог рада је да се покаже како једноставан софтвер може да помогне у организовању и убрзавању свакодневних послова у једној ординацији.

.

2. Прикупљање корисничких захтева

2.1 Вербални опис

Потребно је направити **Софтверски систем за дечију ординацију 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 **Доктор**, *прималац услуге* је **Дете**, *документ* који описује процес пружања услуге је **Рецепт**. *Шифарници* су **Лек**, **Специјализација**.

Конкретни концепти су између себе повезани на следећи начин:

Преко рецепта је могуће преписати више лекова. Рецепт је повезан са једним доктором и једним дететом. Дете може бити школског или предшколско. Доктор може бити везан за више специјализација, док специјализација може бити везана за више доктора.

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.	Рецепт	Креирај, Убаци, Претражи, Промени, Обриши
2.	Дете	Креирај, Убаци, Претражи, Промени, Обриши
3.	Школско дете	Креирај, Убаци, Претражи, Промени, Обриши
4.	Предшколско дете	Креирај, Убаци, Претражи, Промени, Обриши
5.	Доктор	Пријави, Креирај, Убаци, Претражи, Промени, Обриши
6.	Лек	Креирај, Убаци, Претражи, Промени, Обриши
7.	Специјализација	Креирај, Убаци, Претражи, Промени, Обриши
8.	Ставка Рецепта	
9.	ДрСп	

Табела 1: Повезивање концепата и функционалности

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

1. Документи
 - 1.1 Рецепт
2. Пружалац услуге
 - 2.1 Доктор
3. Прималац услуге
 - 3.1 Дете
 - 3.2 Предшколско дете
 - 3.2 Школско дете
4. Шифарници
 - 4.1 Лек,
 - 4.2 Специјализација,
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 рецепт	Учитане су листе: а) Доктор б) Дете с) Лек	
SK2	СК2- Убаци рецепт	Учитане су листе: а) Доктор б) Дете с) Лек	
SK3	СК3- Претражи рецепт		а) Рецепт б) Доктор с) Дете d) Лек
SK4	СК4- Промени рецепт	Учитане су листе: а) Доктор б) Дете с) Лек	а) Рецепт б) Доктор с) Дете d) Лек
SK5	СК5- Обриши рецепт		а) Рецепт б) Доктор с) Дете d) Лек
SK6	СК6- Креирај дете		
SK7	СК7- Убаци дете		
SK8	СК8- Претражи дете		а) Дете
SK9	СК9- Промени дете		а) Дете
SK10	СК10- Обриши дете		а) Дете
SK11	СК11- Креирај школско дете	Учитане су листе: а) Дете	
SK12	СК12- Убаци школско дете	Учитане су листе: а) Дете	
SK13	СК13- Претражи школско дете		а) Школско дете б) Дете
SK14	СК14- Промени школско дете	Учитане су листе: а) Дете	а) Школско дете б) Дете
SK15	СК15- Обриши школско дете		а) Школско дете б) Дете
SK16	СК16- Креирај предшколско дете	Учитане су листе: а) Дете	
SK17	СК17- Убаци предшколско дете	Учитане су листе: а) Дете	
SK18	СК18- Претражи предшколско дете		а) Предшколско дете б) Дете
SK19	СК19- Промени предшколско дете	Учитане су листе: а) Дете	а) Предшколско дете б) Дете
SK20	СК20- Обриши предшколско дете		а) Предшколско дете б) Дете
SK21	СК21- Пријави доктор		
SK22	СК22- Креирај доктор	Учитане су листе: а) Специјализација	
SK23	СК23- Убаци доктор	Учитане су листе: а) Специјализација	
SK24	СК24- Претражи доктор		а) Доктор б) Специјализација
SK25	СК25- Промени доктор	Учитане су листе: а) Специјализација	а) Доктор б) Специјализација
SK26	СК26- Обриши доктор		а) Доктор б) Специјализација
SK27	СК27- Креирај лек		
SK28	СК28- Убаци лек		
SK29	СК29- Претражи лек		а) Лек

SK30	SK30- Промени лек		а) Лек
------	-------------------	--	--------

Табела 2: Случајеви коришћења софтверског система

За следеће случајеве коришћења ћемо дати детаљан опис:

СК1- Креирај рецепт
СК2- Убаци рецепт
СК3- Претражи рецепт
СК4- Промени рецепт
СК5- Обриши рецепт
СК6- Креирај дете
СК21- Пријави доктор
СК22- Креирај доктор
СК23- Убаци доктор

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

СК1- Креирај рецепт

Назив СК

Креирај рецепт

Актори СК

Доктор

Учесници СК

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

Предуслови: Кориснички интерфејс (клијентски програм) и систем (серверски програм) су покренути. Доктор је пријављен под својом шифром. Систем приказује форму за рад са рецептом. *Учитане су листе: а) Доктор б) Дете с) Лек*

Основни сценарио СК:

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

Алтернативна сценарија:

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

СК2- Убази рецепт

Назив СК

Убази рецепт

Актори СК

Доктор

Учесници СК

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

Предуслови: Кориснички интерфејс (клијентски програм) и систем (серверски програм) су покренути. Доктор је пријављен под својом шифром. Систем приказује форму за рад са рецептом. *Учитане су листе: а) Доктор б) Дете с) Лек*

Основни сценарио СК:

9. Доктор уноси податке о рецепту. (АПУСО)
10. Доктор контролише да ли је коректно унео податке о рецепту. (АНСО)
11. Доктор позива систем да запамти податке о рецепту. (АПСО)
12. Систем памти податке о рецепту. (СО)
13. Систем приказује доктору рецепт и поруку: "Систем је запамтио рецепт." (ИА)

Алтернативна сценарија:

- 5.1 Уколико систем не може да запамти податке о рецепту он приказује доктору поруку: "Систем не може да запамти рецепт". (ИА)

СКЗ- Претражи рецепт

Назив СК

Претражи рецепт

Актори СК

Доктор

Учесници СК

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

Предуслови: Кориснички интерфејс (клијентски програм) и систем (серверски програм) су покренути. Доктор је пријављен под својом шифром. Кориснички интерфејс приказује форму за рад са рецептом. На наведеној екранској форми су дефинисани критеријуми, који се односе на: а) Рецепт б) Доктор с) Дете д) Лек, који ће да врате листу рецепта.

Основни сценарио СК:

14. Доктор бира критеријуме на основу којих претражује рецепте. (АПУСО)
15. Доктор позива систем да нађе рецепте по задатим критеријумима. (АПСО)
16. Систем тражи рецепте по задатим критеријумима. (СО)
17. Систем приказује доктору рецепте и поруку: "Систем је нашао рецепте по задатим критеријумима". (ИА)
18. Доктор бира рецепт. (АПУСО)
19. Доктор позива систем да нађе рецепт. (АПСО)
20. Систем тражи рецепт. (СО)
21. Систем приказује доктору рецепт и поруку: "Систем је нашао рецепт". (ИА)

Алтернативна сценарија:

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

СК4- Промени рецепт

Назив СК

Промени рецепт

Актери СК

Доктор

Учесници СК

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

Предуслови: Кориснички интерфејс (клијентски програм) и систем (серверски програм) су покренути. Доктор је пријављен под својом шифром. Кориснички интерфејс приказује форму за рад са рецептом. На наведеној екранској форми су дефинисани критеријуми, који се односе на: а) Рецепт б) Доктор с) Дете д) Лек, који ће да врате листу рецепта. Учитане су листе: а) Доктор б) Дете с) Лек

Основни сценарио СК:

22. Доктор бира критеријуме на основу којих претражује рецепте. (АПУСО)
23. Доктор позива систем да нађе рецепте по задатим критеријумима. (АПСО)
24. Систем тражи рецепте по задатим критеријумима. (СО)
25. Систем приказује доктору рецепте и поруку: "Систем је нашао рецепте по задатим критеријумима". (ИА)
1. Доктор бира рецепт. (АПУСО)
2. Доктор позива систем да нађе рецепт. (АПСО)
3. Систем тражи рецепт. (СО)
4. Систем приказује доктору рецепт и поруку: "Систем је нашао рецепт". (ИА)
5. Доктор уноси (мења) податке о рецепту. (АПУСО)
6. Доктор контролише да ли је коректно унео податке о рецепту. (АНСО)
7. Доктор позива систем да запамти податке о рецепту. (АПСО)
8. Систем памти податке о рецепту. (СО)
9. Систем приказује доктору рецепт и поруку: "Систем је запамтио рецепт." (ИА)

Алтернативна сценарија:

- 4.1 Уколико систем не може да нађе рецепте он приказује доктору поруку: "Систем не може да нађе рецепте по задатом критеријуму". Прекида се извршење сценарија. (ИА)
- 8.1 Уколико систем не може да нађе рецепт он приказује доктору поруку: "Систем не може да нађе рецепт". Прекида се извршење сценарија. (ИА)
- 13.1 Уколико систем не може да запамти податке о рецепту он приказује доктору поруку: "Систем не може да запамти рецепт". (ИА)

СК5- Обриши рецепт

Назив СК

Обриши рецепт

Актори СК

Доктор

Учесници СК

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

Предуслови: Кориснички интерфејс (клијентски програм) и систем (серверски програм) су покренути. Доктор је пријављен под својом шифром. Кориснички интерфејс приказује форму за рад са рецептом. На наведеној екранској форми су дефинисани критеријуми, који се односе на: а) Рецепт б) Доктор с) Дете д) Лек, који ће да врате листу рецепта.

Основни сценарио СК:

10. Доктор бира критеријуме на основу којих претражује рецепте. (АПУСО)
11. Доктор позива систем да нађе рецепте по задатим критеријумима. (АПСО)
12. Систем тражи рецепте по задатим критеријумима. (СО)
13. Систем приказује доктору рецепте и поруку: "Систем је нашао рецепте по задатим критеријумима". (ИА)
14. Доктор бира рецепт. (АПУСО)
15. Доктор позива систем да нађе рецепт. (АПСО)
16. Систем тражи рецепт. (СО)
17. Систем приказује доктору рецепт и поруку: "Систем је нашао рецепт". (ИА)
18. Доктор позива систем да обрише рецепт. (АПСО)
19. Систем брише рецепт. (СО)
20. Систем приказује доктору поруку: "Систем је обрисао рецепт." (ИА)

Алтернативна сценарија:

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

СК6- Креирај дете

Назив СК

Креирај дете

Актори СК

Доктор

Учесници СК

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

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

Основни сценарио СК:

21. Доктор позива систем да креира дете. (АПСО)
22. Систем креира дете. (СО)
23. Систем приказује доктору дете и поруку: "Систем је креирао дете". (ИА)
24. Доктор уноси податке о детету. (АПУСО)
25. Доктор контролише да ли је коректно унео податке о детету. (АНСО)
26. Доктор позива систем да запамти податке о детету. (АПСО)
27. Систем памти податке о детету. (СО)
28. Систем приказује доктору дете и поруку: "Систем је запамтио дете." (ИА)

Алтернативна сценарија:

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

СК21- Пријави доктор

Назив СК

Пријави доктор

Актери СК

Директор

Учесници СК

Директор, 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:

29. Директор уноси корисничко име и шифру. (АПУСО)
30. Директор контролише да ли је коректно унео корисничко име и шифру. (АНСО)
31. Директор позива систем да провери корисничко име и шифру. (АПСО)
32. Систем проверава корисничко име и шифру. (СО)
33. Систем приказује директору поруку: "Корисничко име и шифра су исправни." (ИА)
34. 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 директору 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.

СК22- Креирај доктор

Назив СК

Креирај доктор

Актори СК

Директор

Учесници СК

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

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

Основни сценарио СК:

35. Директор позива систем да креира доктором. (АПСО)
36. Систем креира доктором. (СО)
37. Систем приказује директору доктором и поруку: "Систем је креирао доктором". (ИА)
38. Директор уноси податке о доктору. (АПУСО)
39. Директор контролише да ли је коректно унео податке о доктору. (АНСО)
40. Директор позива систем да запамти податке о доктору. (АПСО)
41. Систем памти податке о доктору. (СО)
42. Систем приказује директору доктором и поруку: "Систем је запамтио доктором." (ИА)

Алтернативна сценарија:

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

СК23- Убаца доктор

Назив СК

Убаца доктор

Актори СК

Директор

Учесници СК

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

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

Основни сценарио СК:

43. Директор уноси податке о доктору. (АПУСО)
44. Директор контролише да ли је коректно унео податке о доктору. (АНСО)
45. Директор позива систем да запамти податке о доктору. (АПСО)
46. Систем памти податке о доктору. (СО)
47. Систем приказује директору доктором и поруку: "Систем је запамтио доктором." (ИА)

Алтернативна сценарија:

- 5.1 Уколико систем не може да запамти податке о доктору он приказује директору поруку: "Систем не може да запамти доктором". (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 рецепт	1. signal KreirajRecept(Recept) 2. signal PromeniRecept(Recept) 3. signal vratiListuSviDoktor(Lista<Doktor>) 4. signal vratiListuSviDete(Lista<Dete>) 5. signal vratiListuSviLek(Lista<Lek>)
SK2	СК2- Убаци рецепт	1. signal UbaciRecept(Recept) 2. signal PromeniRecept(Recept) 3. signal vratiListuSviDoktor(Lista<Doktor>) 4. signal vratiListuSviDete(Lista<Dete>) 5. signal vratiListuSviLek(Lista<Lek>)
SK3	СК3- Претражи рецепт	1. signal PretraziRecept(Recept) 2. signal vratiListuRecept(kriterijumRecept, Lista<Recept>) 3. signal vratiListuRecept(kriterijumDoktor, Lista<Recept>) 4. signal vratiListuRecept(kriterijumDete, Lista<Recept>) 5. signal vratiListuRecept(kriterijumLek, Lista<Recept>)
SK4	СК4- Промени рецепт	1. signal PromeniRecept(Recept) 2. signal PretraziRecept(Recept) 3. signal vratiListuSviDoktor(Lista<Doktor>) 4. signal vratiListuSviDete(Lista<Dete>) 5. signal vratiListuSviLek(Lista<Lek>) 6. signal vratiListuRecept(kriterijumRecept, Lista<Recept>) 7. signal vratiListuRecept(kriterijumDoktor, Lista<Recept>) 8. signal vratiListuRecept(kriterijumDete, Lista<Recept>)
SK5	СК5- Обриши рецепт	1. signal ObrisiRecept(Recept) 2. signal vratiListuRecept(kriterijumRecept, Lista<Recept>) 3. signal vratiListuRecept(kriterijumDoktor, Lista<Recept>) 4. signal vratiListuRecept(kriterijumDete, Lista<Recept>) 5. signal vratiListuRecept(kriterijumLek, Lista<Recept>)
SK6	СК6- Креирај дете	1. signal KreirajDete(Dete) 2. signal PromeniDete(Dete)
SK7	СК7- Убаци дете	1. signal UbaciDete(Dete) 2. signal PromeniDete(Dete)
SK8	СК8- Претражи дете	1. signal PretraziDete(Dete) 2. signal vratiListuDete(kriterijumDete, Lista<Dete>)
SK9	СК9- Промени дете	1. signal PromeniDete(Dete) 2. signal PretraziDete(Dete) 3. signal vratiListuDete(kriterijumDete, Lista<Dete>)
SK10	СК10- Обриши дете	1. signal ObrisiDete(Dete) 2. signal vratiListuDete(kriterijumDete, Lista<Dete>)
SK11	СК11- Креирај школско дете	1. signal KreirajSkolskoDete(SkolskoDete) 2. signal PromeniSkolskoDete(SkolskoDete) 3. signal vratiListuSviDete(Lista<Dete>)
SK12	СК12- Убаци школско дете	1. signal UbaciSkolskoDete(SkolskoDete) 2. signal PromeniSkolskoDete(SkolskoDete) 3. signal vratiListuSviDete(Lista<Dete>)
SK13	СК13- Претражи школско дете	1. signal PretraziSkolskoDete(SkolskoDete) 2. signal vratiListuSkolskoDete(kriterijumSkolskoDete, Lista<SkolskoDete>) 3. signal vratiListuSkolskoDete(kriterijumDete, Lista<SkolskoDete>)
SK14	СК14- Промени школско	1. signal PromeniSkolskoDete(SkolskoDete) 2. signal PretraziSkolskoDete(SkolskoDete) 3. signal vratiListuSviDete(Lista<Dete>) 4. signal vratiListuSkolskoDete(kriterijumSkolskoDete, Lista<SkolskoDete>) 5. signal vratiListuSkolskoDete(kriterijumDete, Lista<SkolskoDete>)

	дете	
SK15	СК15-Обриши школско дете	1. signal ObrisiSkolskoDete(SkolskoDete) 2. signal vratiListuSkolskoDete(kriterijumSkolskoDete, Lista<SkolskoDete>) 3. signal vratiListuSkolskoDete(kriterijumDete, Lista<SkolskoDete>)
SK16	СК16-Креирај предшколско дете	1. signal KreirajPredskolskoDete(PredskolskoDete) 2. signal PromeniPredskolskoDete(PredskolskoDete) 3. signal vratiListuSviDete(Lista<Dete>)
SK17	СК17-Убаци предшколско дете	1. signal UbaciPredskolskoDete(PredskolskoDete) 2. signal PromeniPredskolskoDete(PredskolskoDete) 3. signal vratiListuSviDete(Lista<Dete>)
SK18	СК18-Претражи предшколско дете	1. signal PretraziPredskolskoDete(PredskolskoDete) 2. signal vratiListuPredskolskoDete(kriterijumPredskolskoDete, Lista<PredskolskoDete>) 3. signal vratiListuPredskolskoDete(kriterijumDete, Lista<PredskolskoDete>)
SK19	СК19-Промени предшколско дете	1. signal PromeniPredskolskoDete(PredskolskoDete) 2. signal PretraziPredskolskoDete(PredskolskoDete) 3. signal vratiListuSviDete(Lista<Dete>) 4. signal vratiListuPredskolskoDete(kriterijumPredskolskoDete, Lista<PredskolskoDete>) 5. signal vratiListuPredskolskoDete(kriterijumDete, Lista<PredskolskoDete>)
SK20	СК20-Обриши предшколско дете	1. signal ObrisiPredskolskoDete(PredskolskoDete) 2. signal vratiListuPredskolskoDete(kriterijumPredskolskoDete, Lista<PredskolskoDete>) 3. signal vratiListuPredskolskoDete(kriterijumDete, Lista<PredskolskoDete>)
SK21	СК21-Пријави доктор	1. signal PrijaviDoktor(korisnickolme, sifra)
SK22	СК22-Креирај доктор	1. signal KreirajDoktor(Doktor) 2. signal PromeniDoktor(Doktor) 3. signal vratiListuSviSpecijalizacija(Lista<Specijalizacija>)
SK23	СК23-Убаци доктор	1. signal UbaciDoktor(Doktor) 2. signal PromeniDoktor(Doktor) 3. signal vratiListuSviSpecijalizacija(Lista<Specijalizacija>)
SK24	СК24-Претражи доктор	1. signal PretraziDoktor(Doktor) 2. signal vratiListuDoktor(kriterijumDoktor, Lista<Doktor>) 3. signal vratiListuDoktor(kriterijumSpecijalizacija, Lista<Doktor>)
SK25	СК25-Промени доктор	1. signal PromeniDoktor(Doktor) 2. signal PretraziDoktor(Doktor) 3. signal vratiListuSviSpecijalizacija(Lista<Specijalizacija>) 4. signal vratiListuDoktor(kriterijumDoktor, Lista<Doktor>) 5. signal vratiListuDoktor(kriterijumSpecijalizacija, Lista<Doktor>)
SK26	СК26-Обриши доктор	1. signal ObrisiDoktor(Doktor) 2. signal vratiListuDoktor(kriterijumDoktor, Lista<Doktor>) 3. signal vratiListuDoktor(kriterijumSpecijalizacija, Lista<Doktor>)
SK27	СК27-Креирај лек	1. signal KreirajLek(Lek) 2. signal PromeniLek(Lek)
SK28	СК28-Убаци лек	1. signal UbaciLek(Lek) 2. signal PromeniLek(Lek)
SK29	СК29-Претражи лек	1. signal PretraziLek(Lek) 2. signal vratiListuLek(kriterijumLek, Lista<Lek>)
SK30	СК30-Промени лек	1. signal PromeniLek(Lek) 2. signal PretraziLek(Lek) 3. signal vratiListuLek(kriterijumLek, Lista<Lek>)

Табела 4: Системске операције случаја коришћења

Наводимо све уочене системске операције:

1. signal KreirajDete(Dete)
2. signal KreirajDoktor(Doktor)
3. signal KreirajLek(Lek)
4. signal KreirajPredskolskoDete(PredskolskoDete)
5. signal KreirajRecept(Recept)
6. signal KreirajSkolskoDete(SkolskoDete)
7. signal KreirajSpecijalizacija(Specijalizacija)
8. signal ObrisiDete(Dete)
9. signal ObrisiDoktor(Doktor)
10. signal ObrisiLek(Lek)
11. signal ObrisiPredskolskoDete(PredskolskoDete)
12. signal ObrisiRecept(Recept)

13. signal ObrisiSkolskoDete(SkolskoDete)
14. signal ObrisiSpecijalizacija(Specijalizacija)
15. signal PretraziDete(Dete)
16. signal PretraziDoktor(Doktor)
17. signal PretraziLek(Lek)
18. signal PretraziPredskolskoDete(PredskolskoDete)
19. signal PretraziRecept(Recept)
20. signal PretraziSkolskoDete(SkolskoDete)
21. signal PretraziSpecijalizacija(Specijalizacija)
22. signal PrijaviDoktor(korisnickolme, sifra)
23. signal PromeniDete(Dete)
24. signal PromeniDoktor(Doktor)
25. signal PromeniLek(Lek)
26. signal PromeniPredskolskoDete(PredskolskoDete)
27. signal PromeniRecept(Recept)
28. signal PromeniSkolskoDete(SkolskoDete)
29. signal PromeniSpecijalizacija(Specijalizacija)
30. signal UbaciDete(Dete)
31. signal UbaciDoktor(Doktor)
32. signal UbaciLek(Lek)
33. signal UbaciPredskolskoDete(PredskolskoDete)
34. signal UbaciRecept(Recept)
35. signal UbaciSkolskoDete(SkolskoDete)
36. signal UbaciSpecijalizacija(Specijalizacija)
37. signal vratiListuDete(kriterijumDete, Lista<Dete>)
38. signal vratiListuDoktor(kriterijumDoktor, Lista<Doktor>)
39. signal vratiListuDoktor(kriterijumSpecijalizacija, Lista<Doktor>)
40. signal vratiListuLek(kriterijumLek, Lista<Lek>)
41. signal vratiListuPredskolskoDete(kriterijumDete, Lista<PredskolskoDete>)
42. signal vratiListuPredskolskoDete(kriterijumPredskolskoDete, Lista<PredskolskoDete>)
43. signal vratiListuRecept(kriterijumDete, Lista<Recept>)
44. signal vratiListuRecept(kriterijumDoktor, Lista<Recept>)
45. signal vratiListuRecept(kriterijumLek, Lista<Recept>)
46. signal vratiListuRecept(kriterijumRecept, Lista<Recept>)
47. signal vratiListuSkolskoDete(kriterijumDete, Lista<SkolskoDete>)
48. signal vratiListuSkolskoDete(kriterijumSkolskoDete, Lista<SkolskoDete>)
49. signal vratiListuSpecijalizacija(kriterijumSpecijalizacija, Lista<Specijalizacija>)
50. signal vratiListuSviDete(Lista<Dete>)
51. signal vratiListuSviDoktor(Lista<Doktor>)
52. signal vratiListuSviLek(Lista<Lek>)
53. signal vratiListuSviSpecijalizacija(Lista<Specijalizacija>)

Табела 5: Системске операције софтверског система

3.2 Понашање софтверског система - Секвенцни дијаграми случаја коришћења

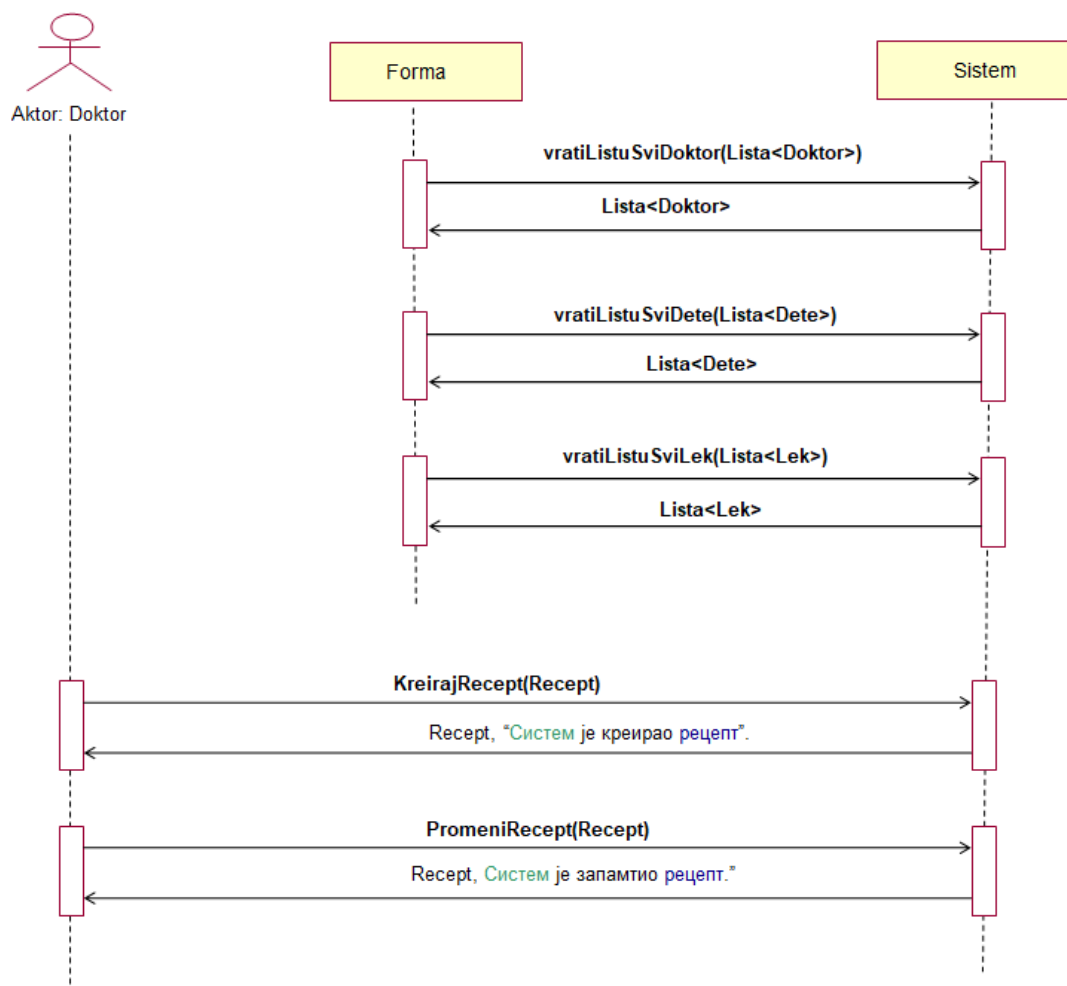
ДС1: Дијаграми секвенци случаја коришћења – Креирај рецепт

Предуслови:

1. **Форма** позива систем да **врати** листу свих доктора. (АПСО)
2. **Систем** враћа **форми** листу свих доктора. (ИА)
3. **Форма** позива систем да **врати** листу све деце. (АПСО)
4. **Систем** враћа **форми** листу све деце. (ИА)
5. **Форма** позива систем да **врати** листу свих лекова. (АПСО)
6. **Систем** враћа **форми** листу свих лекова. (ИА)

Основни сценарио СК:

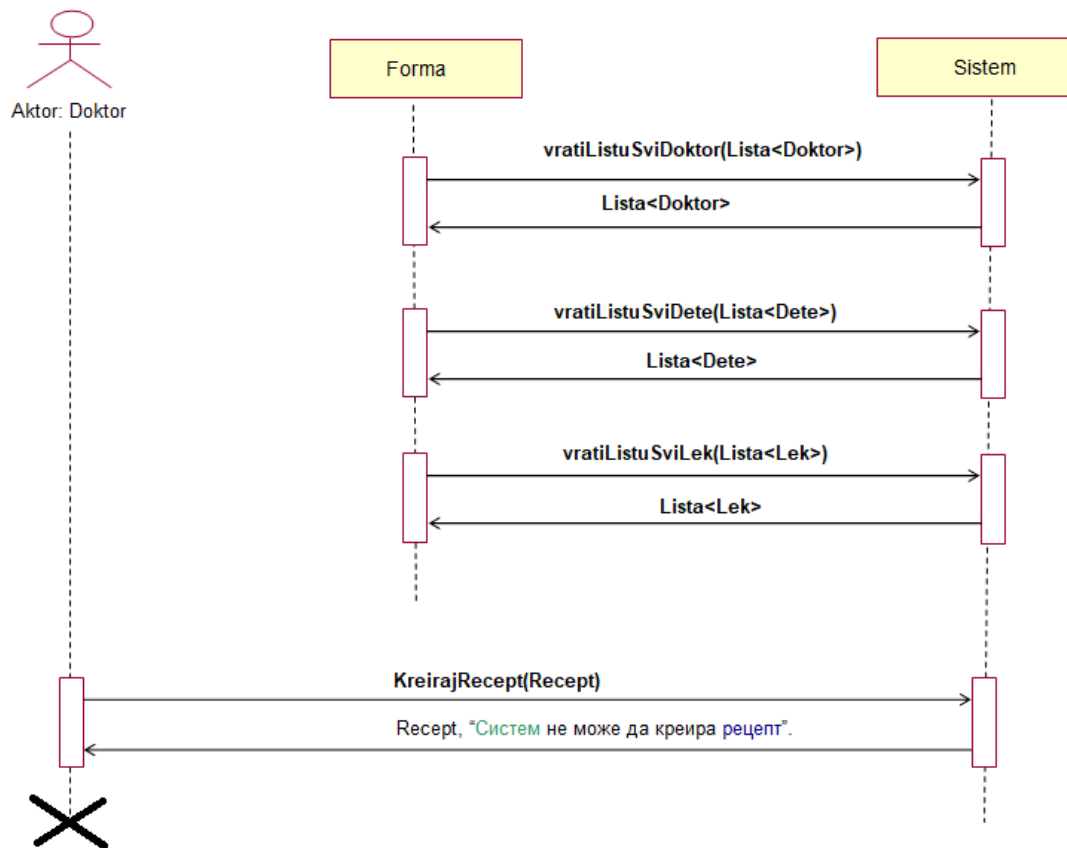
7. **Доктор** позива **систем** да **креира** **рецепт**. (АПСО)
8. **Систем** приказује **доктору** **рецепт** и поруку: “**Систем** је креирао **рецепт**”. (ИА)
9. **Доктор** позива **систем** да запамти податке о **рецепту**. (АПСО)
10. **Систем** приказује **доктору** **рецепт** и поруку: “**Систем** је запамтио **рецепт**.” (ИА)



Слика 1. ДС1 Основни сценарио

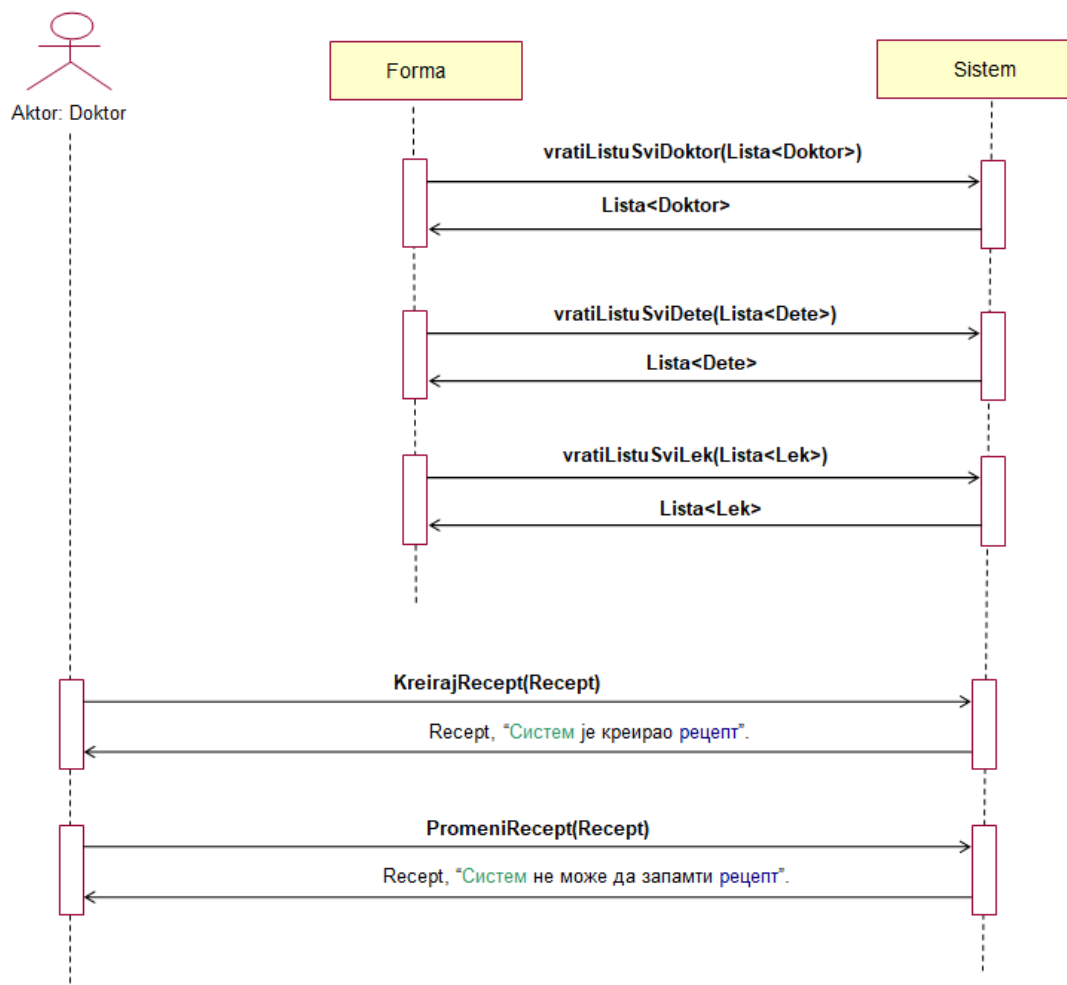
Алтернативна сценарија:

8.1 Уколико **систем** не може да креира **рецепт** он **приказује доктору** поруку: “**Систем** не може да креира **рецепт**”. Прекида се извршење сценарија. (ИА)



Слика 2. ДС1 Алтернативни сценарио 1

10.1 Уколико **систем** не може да запамти податке о **рецепту** он **приказује доктору** поруку: “**Систем** не може да запамти **рецепт**”. (ИА)



Слика 3. ДС1 Алтернативни сценарио 2

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

1.	signal KreirajRecept(Recept)
2.	signal PromeniRecept(Recept)
3.	signal vratiListuSviDoktor(Lista<Doktor>)
4.	signal vratiListuSviDete(Lista<Dete>)
5.	signal vratiListuSviLek(Lista<Lek>)

ДС2: Дијаграми секвенци случаја коришћења – Претражи рецепт

Основни сценарио СК:

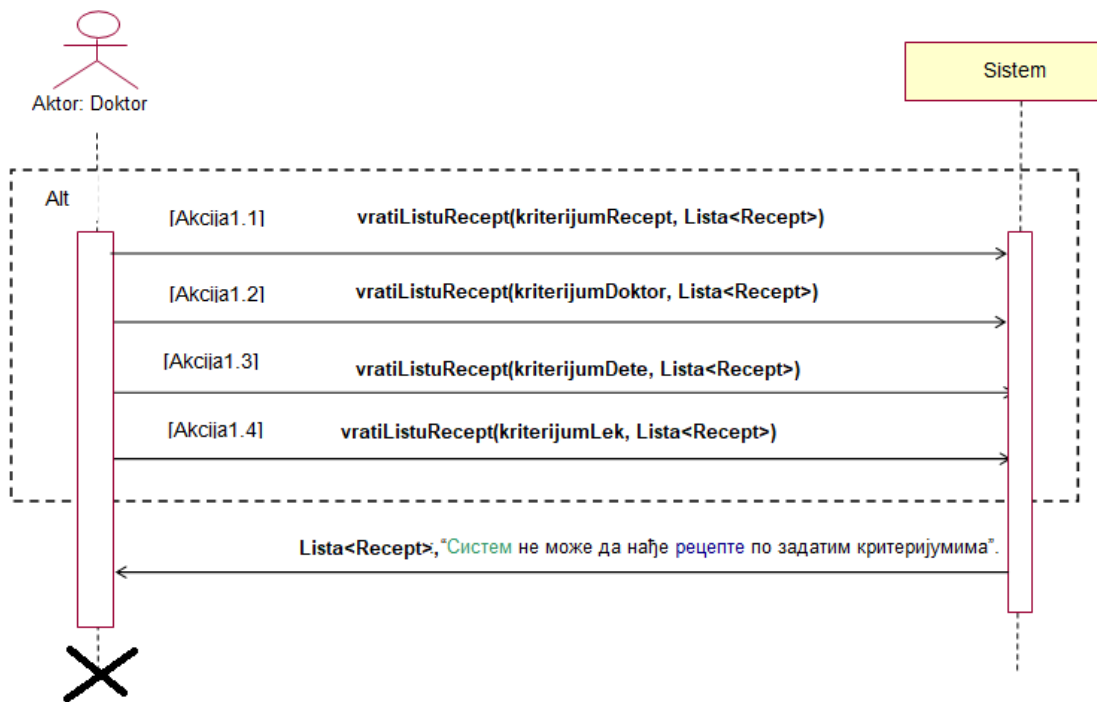
1. **Доктор** позива **систем** да нађе **рецепте** по задатим критеријумима. (АПСО)
2. **Систем** приказује **доктору** **рецепте** и поруку: “**Систем** је нашао **рецепте** по задатим критеријумима”. (ИА)
3. **Доктор** позива **систем** да нађе **рецепт**. (АПСО)
4. **Систем** приказује **доктору** **рецепт** и поруку: “**Систем** је нашао **рецепт**”. (ИА)



Слика 4. ДС2 Основни сценарио

Алтернативна сценарија:

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



Слика 5. ДС2 Алтернативни сценарио 1

4.1 Уколико **систем** не може да нађе **рецепт** он **приказује доктору** поруку: “**Систем** не може да нађе **рецепт** ”.(ИА)



Слика 6. ДС2 Алтернативни сценарио 2

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

1.	signal PretraziRecept(Recept)
2.	signal vratiListuRecept(kriterijumRecept, Lista<Recept>)
3.	signal vratiListuRecept(kriterijumDoktor, Lista<Recept>)
4.	signal vratiListuRecept(kriterijumDete, Lista<Recept>)
5.	signal vratiListuRecept(kriterijumLek, Lista<Recept>)

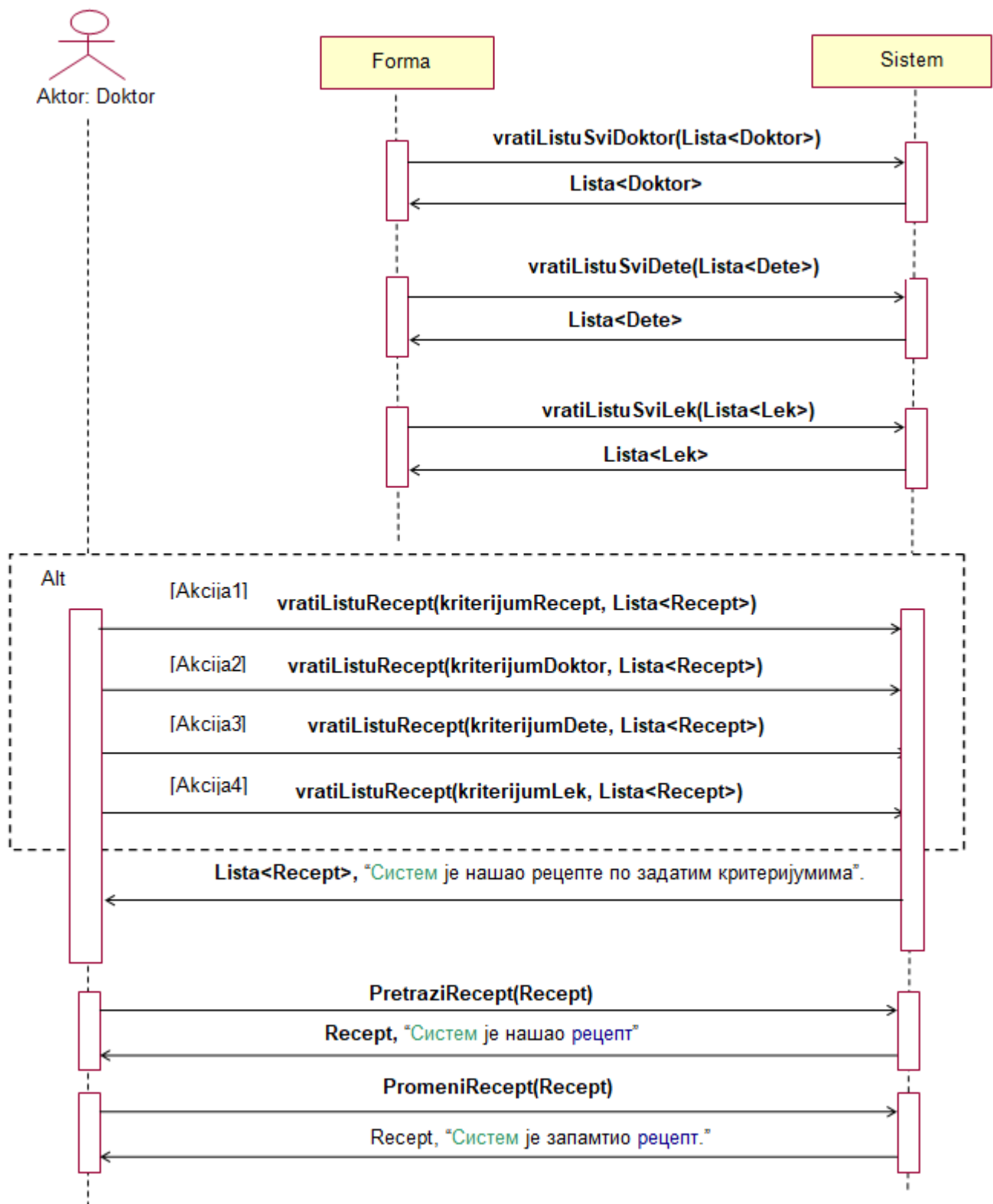
ДСЗ: Дијаграми секвенци случаја коришћења – Промени рецепт

Предуслови:

1. **Форма позива** систем да **врати** листу свих продаваца. (АПСО)
2. **Систем враћа форми** листу свих продаваца. (ИА)
3. **Форма позива** систем да **врати** листу свих пословних партнера. (АПСО)
4. **Систем враћа форми** листу свих пословних партнера. (ИА)
5. **Форма позива** систем да **врати** листу свих роба. (АПСО)
6. **Систем враћа форми** листу свих роба. (ИА)

Основни сценарио СК:

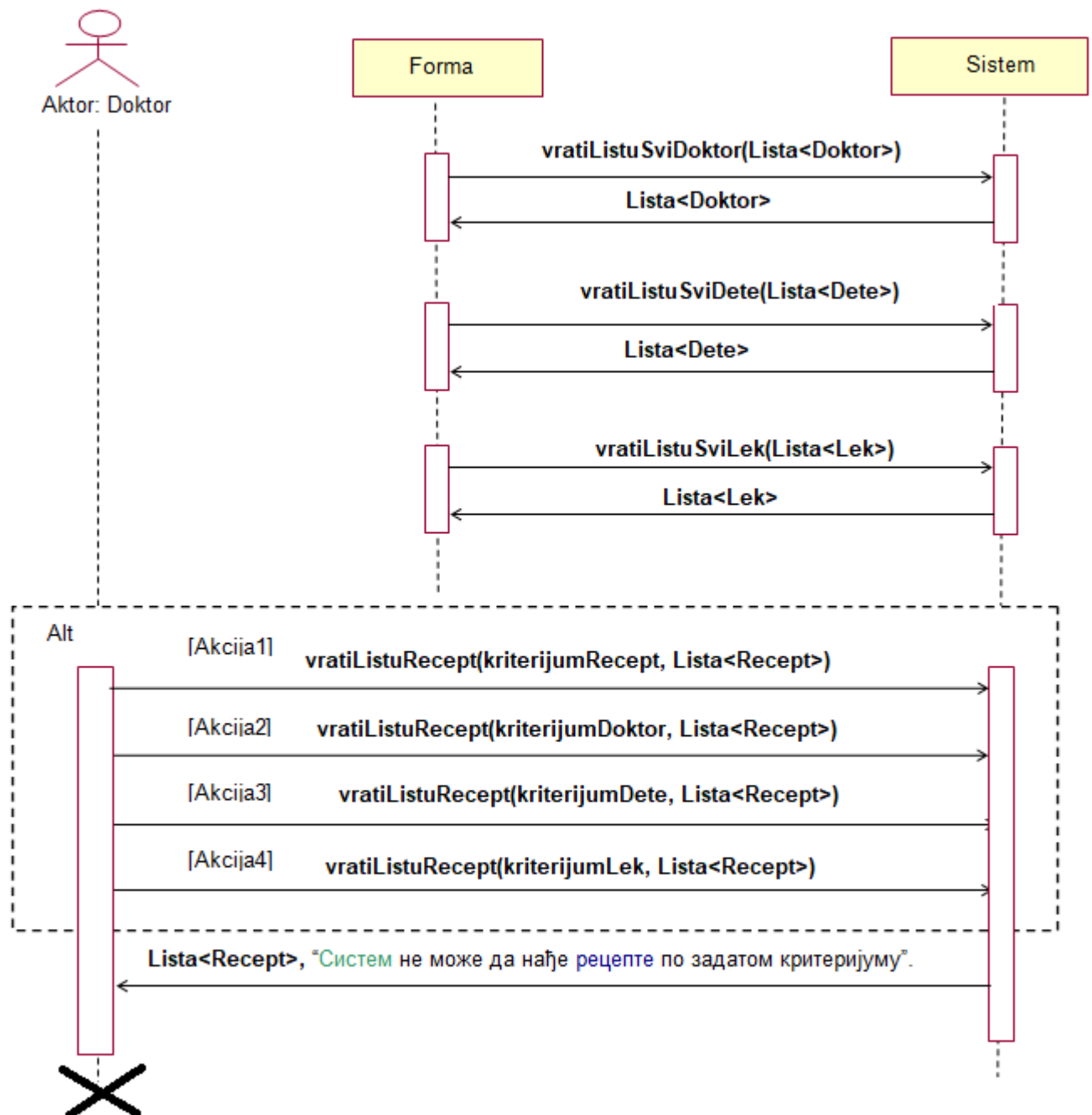
7. **Доктор позива систем** да нађе **рецепте** по задатим критеријумима. (АПСО)
8. **Систем приказује доктору** **рецепте** и поруку: "**Систем** је нашао **рецепте** по задатим критеријумима". (ИА)
9. **Доктор позива систем** да нађе **рецепт**. (АПСО)
10. **Систем приказује доктору** **рецепт** и поруку: "**Систем** је нашао **рецепт**". (ИА)
11. **Доктор позива систем** да запамти податке о **рецепту**. (АПСО)
12. **Систем приказује доктору** **рецепт** и поруку: "**Систем** је запамтио **рецепт**." (ИА)



Слика 7. ДСЗ Основни сценарио

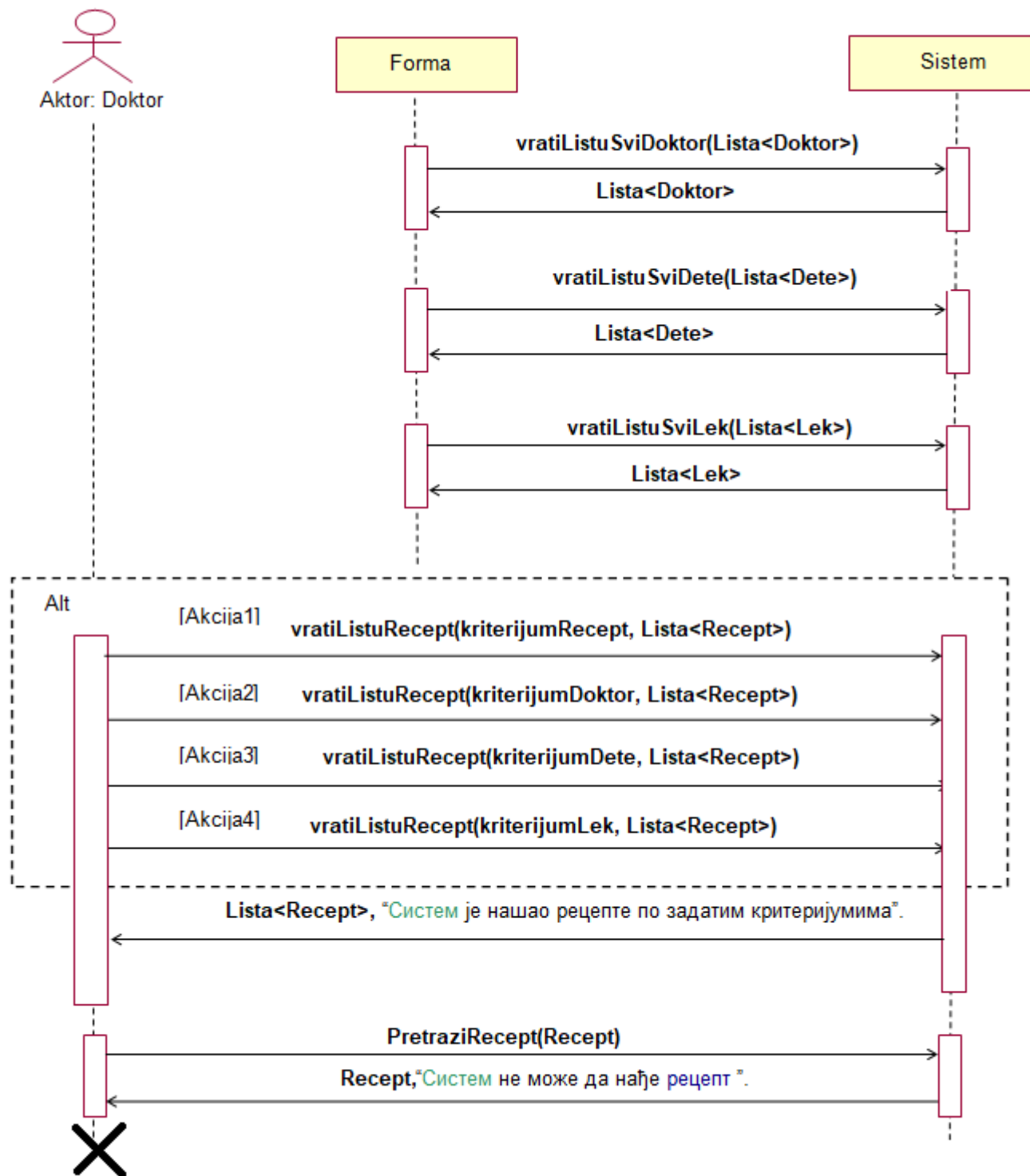
Алтернативна сценарија:

8.1 Уколико **систем** не може да нађе **рецепте** он **приказује доктору** поруку: “**Систем** не може да нађе **рецепте** по задатом критеријуму”. Прекида се извршење сценарија. (IIA)



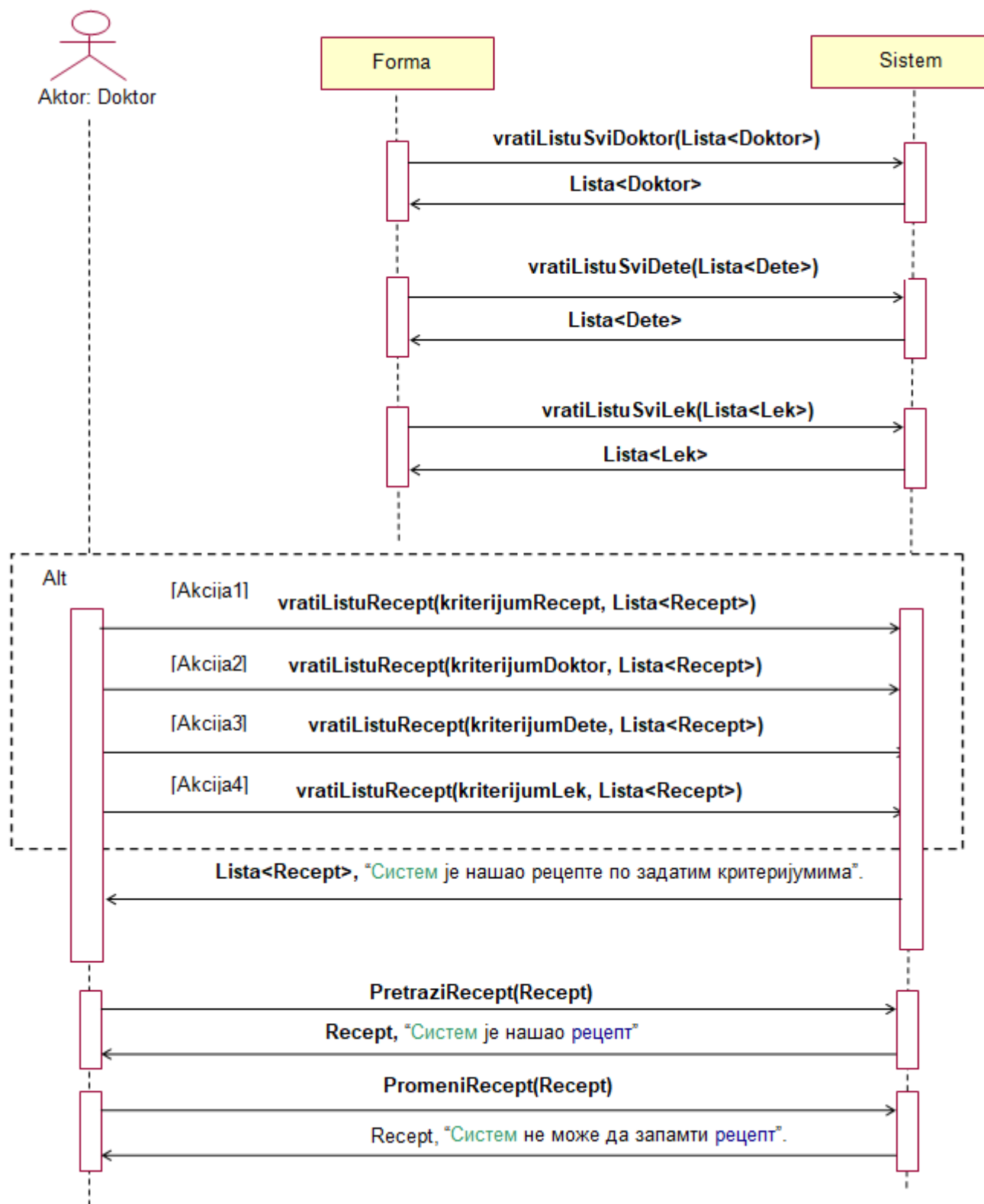
Слика 8. ДСЗ Алтернативни сценарио 1

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



Слика 9. ДСЗ Алтернативни сценарио 2

12.1 Уколико **систем** не може да запамти податке о **рецепту** он **приказује доктору** поруку: “Систем не може да запамти **рецепт**”. (ИА)



Слика 10. ДСЗ Алтернативни сценарио 3

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

1.	signal PromeniRecept(Recept)
2.	signal PretraziRecept(Recept)
3.	signal vratiListuSviDoktor(Lista<Doktor>)
4.	signal vratiListuSviDete(Lista<Dete>)
5.	signal vratiListuSviLek(Lista<Lek>)
6.	signal vratiListuRecept(kriterijumRecept, Lista<Recept>)
7.	signal vratiListuRecept(kriterijumDoktor, Lista<Recept>)
8.	signal vratiListuRecept(kriterijumDete, Lista<Recept>)
9.	signal vratiListuRecept(kriterijumLek, Lista<Recept>)

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: *PrijaviDoktor(korisnickolme, sifra)*

Операција: *PrijaviDoktor(korisnickolme, sifra)*:signal;

Веза са СК: СК21

Предуслови:

Постуслови: *Доктор је пријављен на систем.*

2. Уговор UG2: *KreirajRecept(Recept)*

Операција: *KreirajRecept(Recept)*:signal;

Веза са СК: СК1

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

Постуслови: *Направљен је нови објекат класе ЕвиденцијаКурса.*

3. Уговор UG3: *UbacijRecept(Recept)*

Операција: *UbacijRecept(Recept)*:signal;

Веза са СК: СК2

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

Постуслови: *Направљен је нови објекат класе Рецепт.*

4. Уговор UG4: *PromeniRecept(Recept)*

Операција: *PromeniRecept(Recept)*:signal;

Веза са СК: СК4

Предуслови: *Структурна и вредносна ограничење над објектом класе Рецепт морају бити задовољена.¹*

Постуслови: *Објекат класе Рецепт је промењен.*

¹ *Ако је објекат класе X сторниран не може се извршити системска операција.*

Ако је објекат класе X обрађен не може се извршити системска операција осим превођења објекта класе X у стање сторниран.

5. Уговор UG5: *ObrisiRecept(Recept)*

Операција: *ObrisiRecept(Recept)*:signal;

Веза са СК: СК5

Предуслови *Структурна и вредносна ограничење над објектом класе Рецепт морају бити задовољена.*²

Постуслови: *Објекат класе Рецепт је обрисан.*

6. Уговор UG6: *PretraziRecept(Recept)*

Операција: *PretraziRecept(Recept)*:signal;

Веза са СК: СК3

Предуслови:

Постуслови: *Пронађен је тражени објекат класе Рецеом.*

7. Уговор UG7: *vratiListuDoktor(kriterijumDoktor, Lista<Doktor>)*

Операција: *vratiListuDoktor(kriterijumDoktor, Lista<Doktor>)*:signal;

Веза са СК: СК24 СК25 СК26

Предуслови:

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

8. Уговор UG8: *vratiListuSviDete(Lista<Dete>)*

Операција: *vratiListuSviDete(Lista<Dete>)*:signal;

Веза са СК: СК1 СК2 СК4

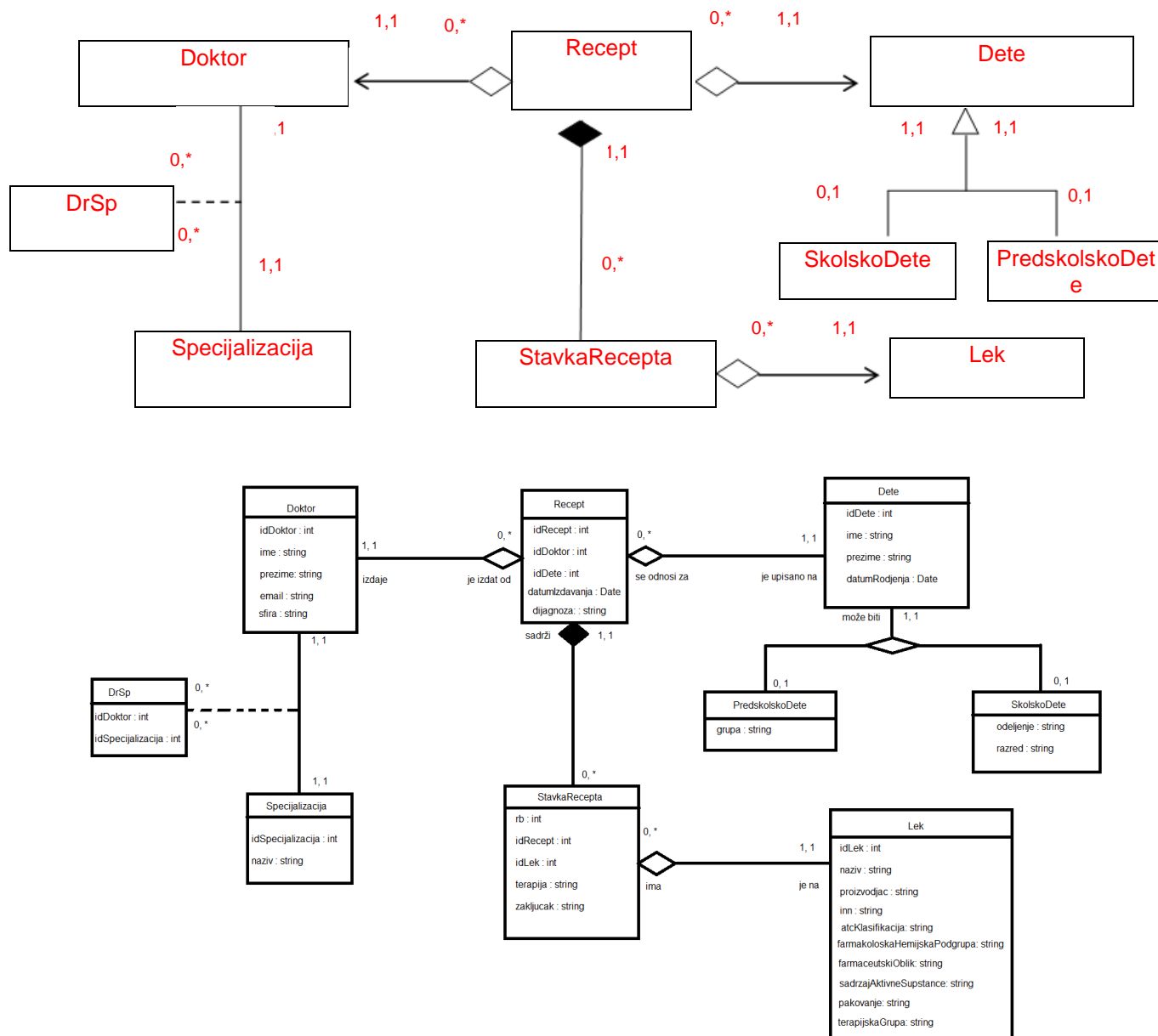
Предуслови:

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

² *Ако је објекат класе X обрађен или сторниран не може се извршити системска операција.*

3.4 Структура софтверског система – Концептуални (доменски) модел

Тип концептуалног модела 3:



Слика 11. Концептуални модел

3.5 Структура софтверског система – Релациони модел

На основу концептуалног модела се прави релациони модел .

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

1. **Doktor** (idDoktor, ime, prezime, email, sifra)
2. **Lek** (idLek, naziv, proizvođjac, inn, atcKlasifikacija, farmakoloskaHemijskaPodgrupa, farmaceutskiOblik, sadrzajAktivneSupstance, pakovanje, terapijskaGrupa)
3. **Specijalizacija** (idSpecijalizacija, naziv)
4. **Dete** (idDete, ime, prezime, datumRodjenja)
5. **SkolskoDete** (idDete, odeljenje, razred)
6. **PredskolskoDete** (idDete, .grupa)
7. **Recept** (idRecept, datumIzdavanja, *idDoktor*, *idDete*)
8. **StavkaRecepta** (*idRecept*, rb, terapija, zakljucak, *idLek*)
9. **DrSp** (*idDoktor*, idSpecijalizacija)

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 **трећег** типа концептуалног модела:

1. Табела Doktor		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 Recept , DrSp DELETE RESTRICTED Recept , DrSp
	idDoktor	Integer	Not null and >0			
	Ime	String	Not null			
	Prezime	String	Not null			
	Email	String	Like ‘%@%’ and not null			
	Sifra	String	Not null			

2. Табела Lek		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 StavkaRecepta , DELETE RESTRICTED StavkaRecepta ,
	idLek	Integer	Not null and >0			
	Naziv	String	Not null			
	Proizvodjac	String	Not null			
	inn	String	Not null			
	atcKlasifikacija	String	Not null			
	farmakoloskaHemijskaPodgrupa	String	Not null			
	farmaceutskiOblik	String	Not null			
	sadrzajAktivneSupstance	String	Not null			
	pakovanje	String	Not null			
	terapijskaGrupa	String	Not null			

3. Табела Specijalizacija		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 DrSp , DELETE RESTRICTED DrSp ,
	idSpecijalizacija	Integer	Not null and >0			
	Naziv	String	Not null			

4. Табела Dete		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Назив	Тип атрибута	Вредност атрибута	Међузав. атрибута једне табеле	Међузав. атрибута више табела	INSERT /
	idDete	Integer	Not null and >0			UPDATE CASCADES SkolskoDete PredskolskoDete Recept
	Ime	String	Not null			DELETE RESTRICTED SkolskoDete PredskolskoDete Recept
	Prezime	String	Not null			
	datumRodjenja	Date	Not null			

5. Табела SkolskoDete		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Назив	Тип атрибута	Вредност атрибута	Међузав. атрибута једне табеле	Међузав. атрибута више табела	INSERT RESTRICTED
	idDete	Integer	Not null and >0			Dete
	Odeljenje	String	Not null			UPDATE RESTRICTED Dete DELETE /
	Razred	String	Not null			

6. Табела PredskolskoDete		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Назив	Тип атрибута	Вредност атрибута	Међузав. атрибута једне табеле	Међузав. атрибута више табела	INSERT RESTRICTED
	idDete	Integer	Not null and >0			Dete
	Grupa	String	Not null			UPDATE RESTRICTED Dete DELETE /

7. Табела Recept		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Назив	Тип атрибута	Вредност атрибута	Међузав. атрибута једне табеле	Међузав. атрибута више табела	INSERT RESTRICTED Dete Doktor UPDATE RESTRICTED Dete Doktor UPDATE CASCADES StavkaRecepta DELETE RESTRICTED StavkaRecepta
	<u>idRecept</u>	Integer	Not null and >0			
	<u>idDoktor</u>	Integer	Not null and >0			
	<u>idDete</u>	Integer	Not null and >0			
	DatumIzavanja	Date	Not null			

8. Табела StavkaRecepta		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Назив	Тип атрибута	Вредност атрибута	Међузав. атрибута једне табеле	Међузав. атрибута више табела	INSERT RESTRICTED Recept Lek UPDATE RESTRICTED Recept Lek DELETE /
	<u>idRecept</u>	Integer	Not null and >0			
	<u>rb</u>	Integer	Not null and >0			
	<u>idLek</u>	Integer	Not null and >0			
	Terapija	String	Not null			
	Zakljucak	String	Not null			

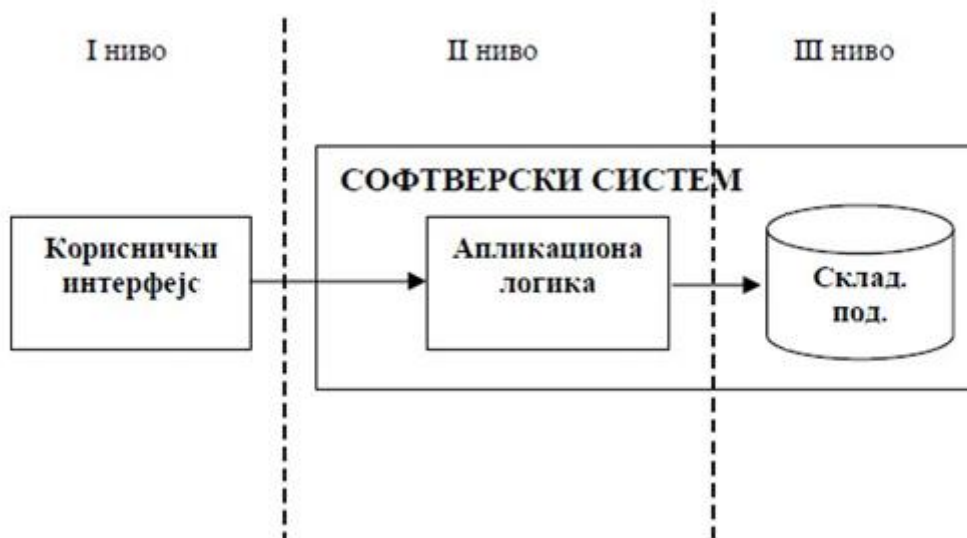
9. Табела DrSp		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Назив	Тип атрибута	Вредност атрибута	Међузав. атрибута једне табеле	Међузав. атрибута више табела	INSERT RESTRICTED Doktor Specijalizacija UPDATE RESTRICTED Doktor Specijalizacija DELETE /
	<u>idDoktor</u>	Integer	Not null and >0			
	<u>idSpecijalizacija</u>	Integer	Not null and >0			

4. Пројектовање

Фаза пројектовања обухвата физичко пројектовање структуре и понашања софтверског система. Обухвата пројектовање екранских форми, апликационе логике и складишта података. Пројектовање корисничког интерфејса обухвата пројектовање екранских форми и контролера корисничког интерфејса. У оквиру апликационе логике се пројектују контролер апликационе логике, пословна логика и брокер базе података. Пројектовање пословне логике обухвата пројектовање логичке структуре и понашања софтверског система. Пројектовање складишта података обухвата креирање релационе базе на основу концептуалног и релационог модела.

Тронивојска архитектура пројектованог софтверског система састоји се из:

- Корисничког интерфејса
- Апликационе логике
- Складишта (базе) података



Слика 12. Тронивојска архитектура

4.1 Пројектовање корисничког интерфејса

Кориснички интерфејс представља улазно-излазну реализацију софтверског система.

Састоји се од:

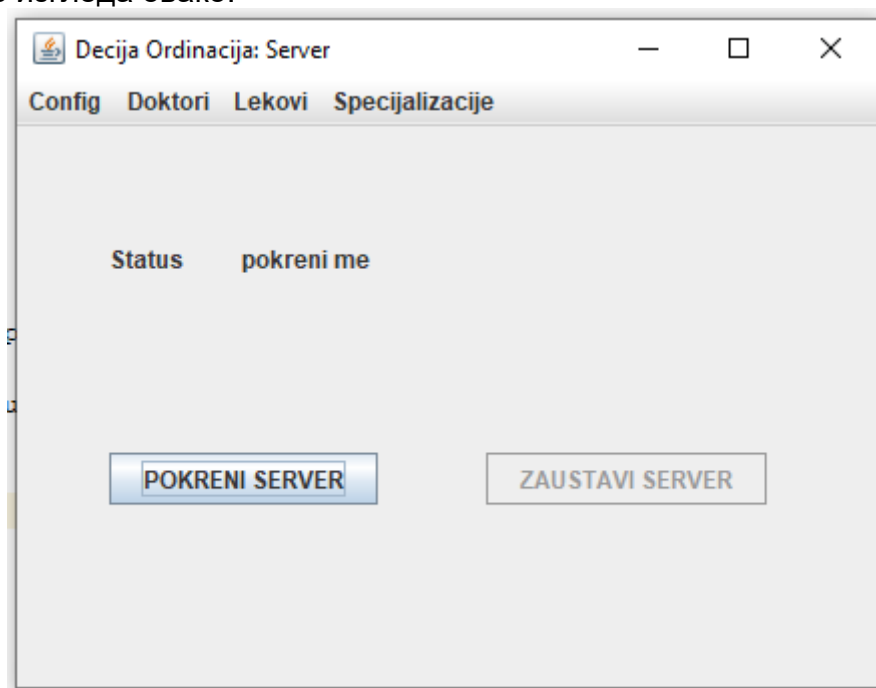
1. Екранске форме
2. Контролера корисничког интерфејса



Слика 13. Структура корисничког интерфејса

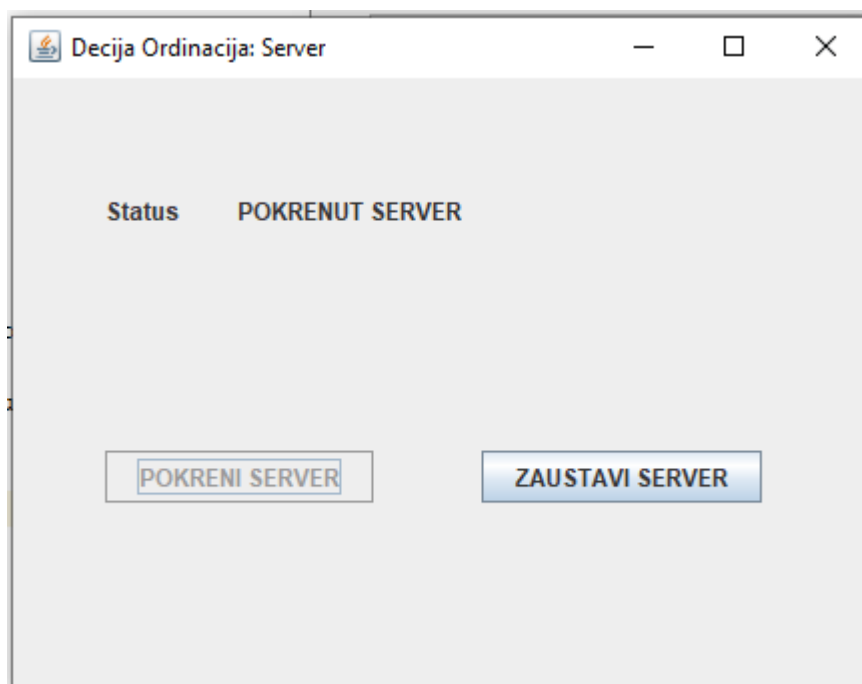
4.1.1 Пројектовање екранских форми

Кориснички интерфејс је изграђен кроз низ екранских форми чија су сценарија коришћења директно повезана са сценаријима случајева коришћења. На серверској страни програма пројектована је серверска екранска форма која пре активације изгледа овако:



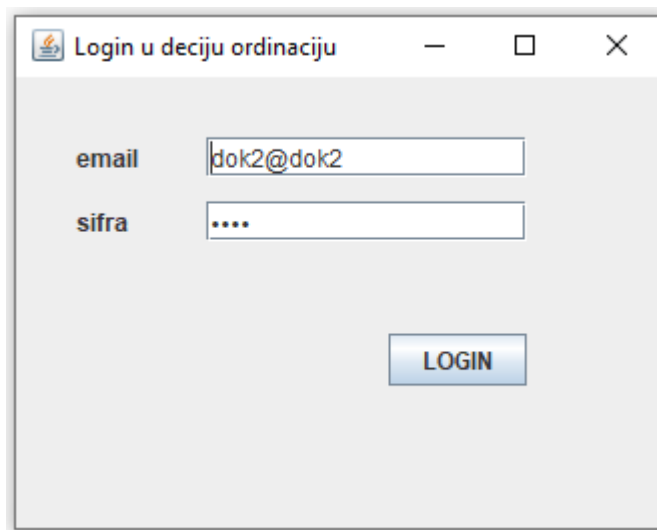
Слика 14. Серверска форма пре покретања

Након активације, серверска форма изгледа овако:



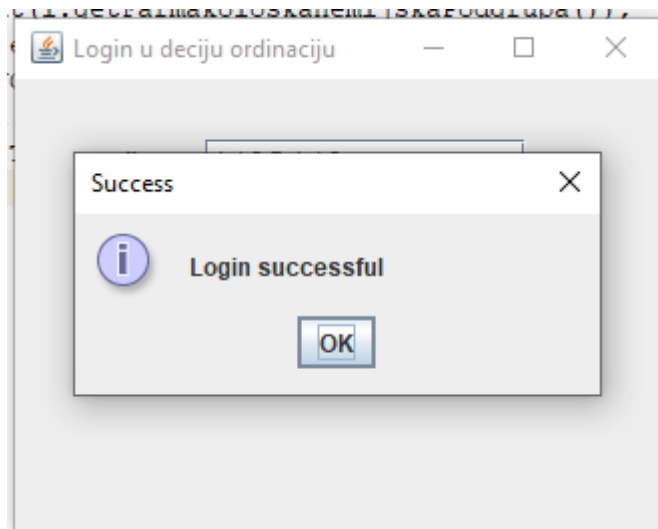
Слика 15. Серверска форма након покретанња

Да би се апликација користила, клијент се претходно треба пријавити са својим креденцијалима на клијентској страни. Login форма изгледа овако:



Слика 16. Login форма

У случају успешног пријављивања клијент добија обавештење и приказује му се главна екранска форма са навигационим менијем ка осталим екранским формама.



Слика 17. Login форма – порука о успешном пријављивању

Decija ordinacija: Izdaj recept

Dete Recept Obrazovanje deteta

Ulogovani Doktor: dok2 dok2

Log out

Izdajte recept za dete:

Dete{ime=nikola, prezime=nikolic}

Dijagnoza

Povišena telesna temperatura (febrilnost)

Zaključak:

Terapija:

Lek:

LEK - PROIZ

Izdaj recept

Слика 18. Главна клијентска форма

СК1- Креирај рецепт

Назив СК

Креирај рецепт

Актори СК

Доктор

Учесници СК

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

Предуслови: Кориснички интерфејс (клијентски програм) и систем (серверски програм) су покренути. Доктор је пријављен под својом шифром. Систем приказује форму за рад са рецептом. *Учитане су листе: а) Доктор б) Дете с) Лек*

Коментар: Приликом израде софтверског система направљена је одлука да улоговани доктор може да изда рецепт само на своје име, и тиме није потребно да систем учита листу доктора.

Decija ordinacija: Izdaj recept

Dete Recept Obrazovanje deteta

Ulogovani Doktor: dok2 dok2

Log out

Izdajte recept za dete:

Dete{ime=nikola, prezime=nikolic}

Dijagnoza

Povišena telesna temperatura (febrilnost)

Zakljucak:

Terapija:

Lek:

LEK - PROIZ

Izdaj recept

Слика 19. Форма за креирање новог рецепта

Основни сценарио СК:

- 48. Доктор позива систем да креира рецепт. (АПСО)
- 49. Систем креира рецепт. (СО)
- 50. Систем приказује доктору рецепт и поруку: "Систем је креирао рецепт". (ИА)
Коментар: Приликом израде софтверског система одлучено је да се рецепт креира и памти након уноса података.
- 51. Доктор уноси податке о рецепту. (АПУСО)

Decija ordinacija: Izdaj recept

Dete Recept Obrazovanje deteta

Ulogovani Doktor: dok2 dok2

Log out

Izdajte recept za dete:

Dete{ime=nikola, prezime=nikolic}

Dijagnoza

Povišena telesna temperatura (febrilnost)

Zakljucak:

Dete ima povišenu telesnu temperaturu i neće moći da učestvuje

Terapija:

Detetu treba da se daje jedan lek dnevno 3 dana za redom

Lek:

LEK - PROIZ

LEK - PROIZ

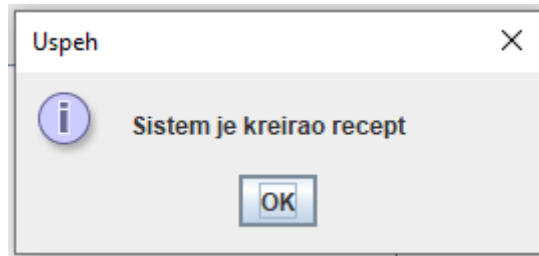
TRIDERM - PROIZ1

NAZIV - PROIZVODJAC

Izdaj recept

Слика 20. Уност података за нови рецепт

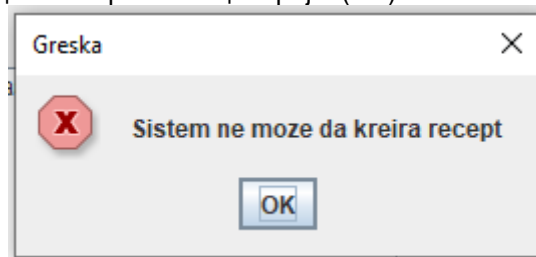
- 52. Доктор контролише да ли је коректно унео податке о рецепту. (АНСО)
- 53. Доктор позива систем да запамти податке о рецепту. (АПСО)
- 54. Систем памти податке о рецепту. (СО)
- 55. Систем приказује доктору рецепт и поруку: "Систем је запамтио рецепт." (ИА)



Слика 21. Успешно креиран рецепт

Алтернативна сценарија:

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



Слика 22. Систем не може да креира рецепт

8.1 Уколико **систем** не може да запамти податке о **рецепту** он **приказује доктору** поруку: “**Систем** не може да запамти **рецепт**”. (ИА)

Коментар: Приликом израде софтверског система одлучено је да рецепт креира и памти у оквиру једне операције.

СКЗ- Претражи рецепт

Назив СК

Претражи рецепт

Актори СК

Доктор

Учесници СК

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

Предуслови: Кориснички интерфејс (клијентски програм) и систем (серверски програм) су покренути. Доктор је пријављен под својом шифром. Кориснички интерфејс приказује форму за рад са рецептом. На наведеној екранској форми су дефинисани критеријуми, који се односе на: а) Рецепт б) Доктор с) Дете д) Лек, који ће да врате листу рецепта.

Коментар: Приликом израде софтверског система одлучено је да неће бити претраге према критеријум Лек.

Prikaz izdatih recepta

Dijagnoza:

Datum izdavanja: (1.1.2001)

Ime deteta:

Prezime deteta:

Ime doktora:

Prezime doktora:

idRecept	datumizavanja	Doktor	Dete	dijagnoza
1	28.12.2024	dok1	nikola	1
2	2.2.2024	dok1	shaolin	1
3	28.2.2025	dok1	nikola	1
4	28.2.2025	dok1	shaolin	1
5	28.2.2025	dok1	dete2	1
6	28.2.2025	dok1	dete2	1
7	28.2.2025	dok1	asdffffff	1
9	14.3.2025	dok2	nikola	Poremećaji u ponašan...
10	19.3.2025	dok2	nikola	Povišena telesna tem...
11	1.6.2025	dok2	nikola	Povišena telesna tem...
12	1.6.2025	dok2	nikola	Povišena telesna tem...

Слика 23. Форма за претраживање рецепта

Основни сценарио СК:

56. Доктор бира критеријуме на основу којих претражује рецепте. (АПУСО)

The form contains the following fields and values:

- Dijagnoza: (empty)
- Datum izdavanja: 1.6.2025 (1.1.2001)
- Ime deteta: nikola
- Prezime deteta: (empty)
- Ime doktora: dok
- Prezime doktora: (empty)
- Buttons: Pretrazi, Resetuj

Слика 24. Критеријуми за претраживање рецепта

57. Доктор позива систем да нађе рецепте по задатим критеријумима. (АПСО)

58. Систем тражи рецепте по задатим критеријумима. (СО)

59. Систем приказује доктору рецепте и поруку: “Систем је нашао рецепте по задатим критеријумима”. (ИА)

Prikaz izdatih recepta

The form shows the same search criteria as Slika 24. Below the form is a table of results:

idRecept	datumizavanja	Doktor	Dete	dijagnoza
1	28.12.2024	dok1	nikola	1
2	2.2.2024	dok1	shaolin	1
3	28.2.2025	dok1	nikola	1
4	28.2.2025	dok1	shaolin	1
5	28.2.2025	dok1	dete2	1
6	28.2.2025	dok1	dete2	1
7	28.2.2025	dok1	dete2	1
9	14.2.2025	dok1	dete2	Poremećaji u ponašan...
10	19.2.2025	dok1	dete2	Povišena telesna tem...
11	1.3.2025	dok1	dete2	Povišena telesna tem...
12	1.3.2025	dok1	dete2	Povišena telesna tem...

A success message dialog box is displayed over the table:

Uspeh
 Sistem je nasao recepte po zadatik kriterijumima
 OK

Buttons at the bottom: AZURIRAJ RECEPT, OBRISI RECEPT

Слика 25. Систем је пронашао рецепт по задатом критеријуму

60. Доктор бира рецепт. (АПУСО)

61. Доктор позива систем да нађе рецепт. (АПСО)


62. Систем тражи рецепт. (СО)

63. Систем приказује доктору рецепт и поруку: “Систем је нашао рецепт”. (ИА)

Коментар: Приликом израде софтверског решења одлучено је да неће постојати детаљан приказ објекта Рецепт, већ да ће та функционалност бити имплементирана кроз Ажурирај форму.

Алтернативна сценарија:

4.1 Уколико систем не може да нађе рецепте он приказује доктору поруку: “Систем не може да нађе рецепте по задатим критеријумима”. Прекида се извршење сценарија. (ИА)

 Prikaz izdatih recepta

Dijagnoza:

Datum izdavanja: (1.1.2001)

Ime deteta: nikolaASasSA


Prezime deteta:

Ime doktora:

Prezime doktora:

idRecept	datumizavanja	Doktor	Dete	dijagnoza
----------	---------------	--------	------	-----------

Greska

 Sistem ne moze da nadje recept po zadatik kriterijumima

Слика 26. Систем је не може да нађе рецепт по задатом критеријуму

8.1 Уколико систем не може да нађе рецепт он приказује доктору поруку: “Систем не може да нађе рецепт”.(ИА)

Коментар: Приликом израде софтверског решења одлучено је да неће постојати детаљан приказ објекта Рецепт, већ да ће та функционалност бити имплементирана кроз Ажурирај форму.

СК4- Промени рецепт

Назив СК

Промени рецепт

Актори СК


Доктор

Учесници СК

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

Предуслови: Кориснички интерфејс (клијентски програм) и систем (серверски програм) су покренути. Доктор је пријављен под својом шифром. Кориснички интерфејс приказује форму за рад са рецептом. На наведеној екранској форми су дефинисани критеријуми, који се односе на: а) Рецепт б) Доктор с) Дете д) Лек, који ће да врате листу рецепта. Учитане су листе: а) Доктор б) Дете с) Лек

Коментар: Приликом израде софтверског система одлучено је да неће бити претраге према критеријум Лек.

 Prikaz izdatih recepta

Dijagnoza:

Datum izdavanja:

Ime deteta:

Prezime deteta:

Ime doktora:

Prezime doktora:

(1.1.2001)

Pretrazi

Resetuj

idRecept	datumIzdavanja	Doktor	Dete	dijagnoza
1	28.12.2024	dok1	nikola	1
2	2.2.2024	dok1	shaolin	1
3	28.2.2025	dok1	nikola	1
4	28.2.2025	dok1	shaolin	1
5	28.2.2025	dok1	dete2	1
6	28.2.2025	dok1	dete2	1
7	28.2.2025	dok1	asdfffffff	1
9	14.3.2025	dok2	nikola	Poremećaji u ponašan...
10	19.3.2025	dok2	nikola	Povišena telesna tem...
11	1.6.2025	dok2	nikola	Povišena telesna tem...
12	1.6.2025	dok2	nikola	Povišena telesna tem...

AZURIRAJ RECEPT

OBRISI RECEPT

Слика 27. Форма за претраживање рецепта

Основни сценарио СК:

64. **Доктор бира** критеријуме на основу којих претражује **рецепте**. (АПУСО)




Dijagnoza:	<input type="text"/>	
Datum izdavanja:	<input type="text" value="1.6.2025"/>	(1.1.2001)
Ime deteta:	<input type="text" value="nikola"/>	
Prezime deteta:	<input type="text"/>	
Ime doktora:	<input type="text" value="dok"/>	
Prezime doktora:	<input type="text"/>	<input type="button" value="Pretrazi"/> <input type="button" value="Resetuj"/>

Слика 28. Критеријуми за претраживање рецепта

65. **Доктор позива систем** да нађе **рецепте** по задатим критеријумима. (АПСО)

66. **Систем тражи** **рецепте** по задатим критеријумима. (СО)

67. **Систем** приказује **доктору** рецепте и поруку: “**Систем** је нашао рецепте по задатим критеријумима”. (ИА)

 Prikaz izdatih recepta

Dijagnoza:

Datum izdavanja:

(1.1.2001)

Ime deteta:

nikola

Prezime deteta:

Ime doktora:


Prezime doktora:

Pretrazi

Resetuj

idRecept	datumizavanja	Doktor	Dete	dijagnoza
1	28.12.2024	dok1	nikola	1
2	2.2.2024	dok1	shaolin	1
3	28.2.2025	dok1	nikola	1
4	28.2.2025	dok1	shaolin	1
5	28.2.2025	dok1	dete2	1
6	28.2.2025	dok1	dete2	1
7	28.2.2025	dok1	dete2	1
9	14.2.2025	dok1	dete2	Poremećaji u ponašan...
10	19.2.2025	dok1	dete2	Povišena telesna tem...
11	1.3.2025	dok1	dete2	Povišena telesna tem...
12	1.3.2025	dok1	dete2	Povišena telesna tem...

Uspeh

 Sistem je nasao recepte po zadatik kriterijumima

OK

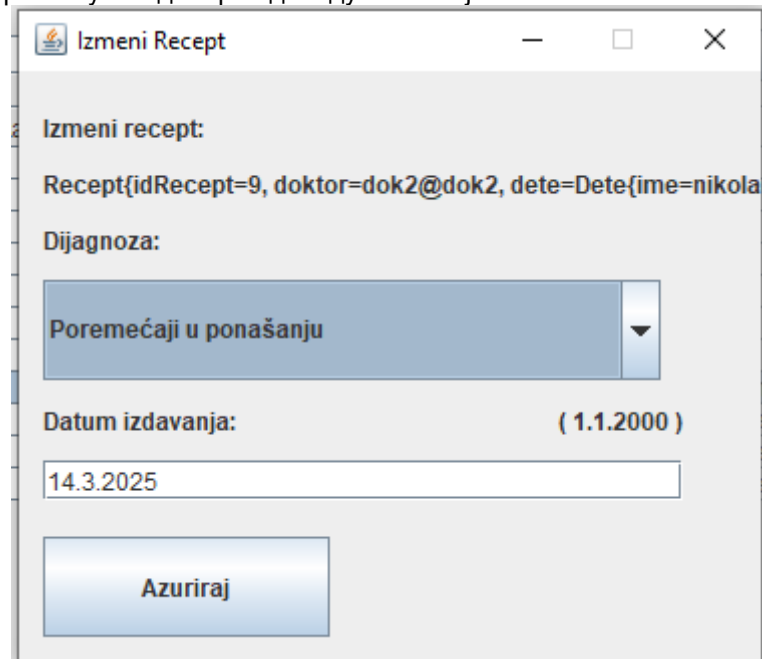
AZURIRAJ RECEPT

OBRISI RECEPT

Слика 29. Систем је пронашао рецепт по задатом критеријуму

68. **Доктор** бира рецепт. (АПУСО)
69. **Доктор** позива **систем** да нађе рецепт. (АПСО)
70. **Систем** тражи рецепт. (СО)

71. **Систем** приказује **доктору** **рецепт** и поруку: “**Систем** је нашао **рецепт**”. (ИА)
Коментар: Приликом израде софтверског решења одлучено је да неће постојати обавештење кориснику за одабир индивидуалног објекта.



Izmeni Recept

Izmeni recept:

Recept{idRecept=9, doktor=dok2@dok2, dete=Dete{ime=nikola

Dijagnoza:

Poremećaji u ponašanju

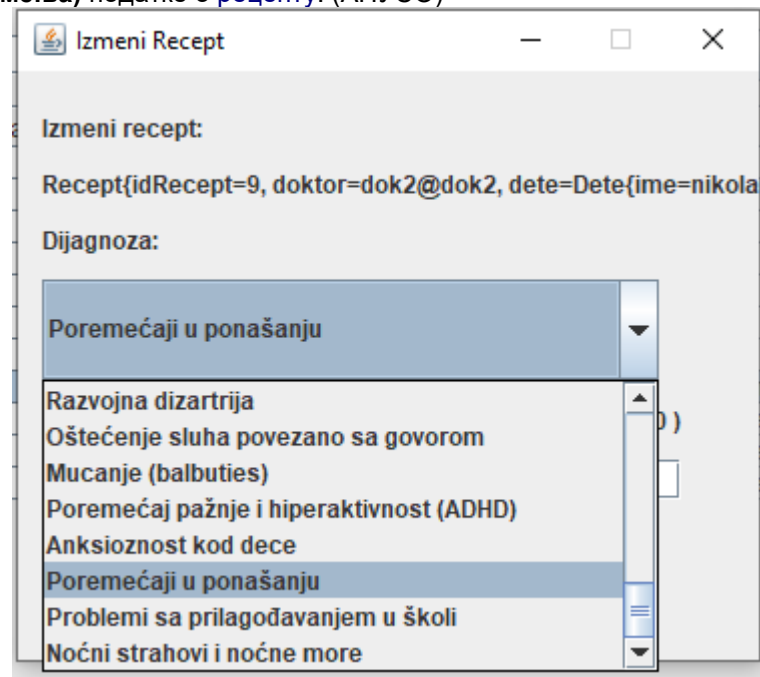
Datum izdavanja: (1.1.2000)

14.3.2025

Azuriraj

Слика 30. Приказ форме за измену рецепта

72. **Доктор** уноси (**мења**) податке о **рецепту**. (АПУСО)



Izmeni Recept

Izmeni recept:

Recept{idRecept=9, doktor=dok2@dok2, dete=Dete{ime=nikola

Dijagnoza:

Poremećaji u ponašanju

Razvojna dizarija

Oštećenje sluha povezano sa govorom

Mucanje (balbuties)

Poremećaj pažnje i hiperaktivnost (ADHD)

Anksioznost kod dece

Poremećaji u ponašanju

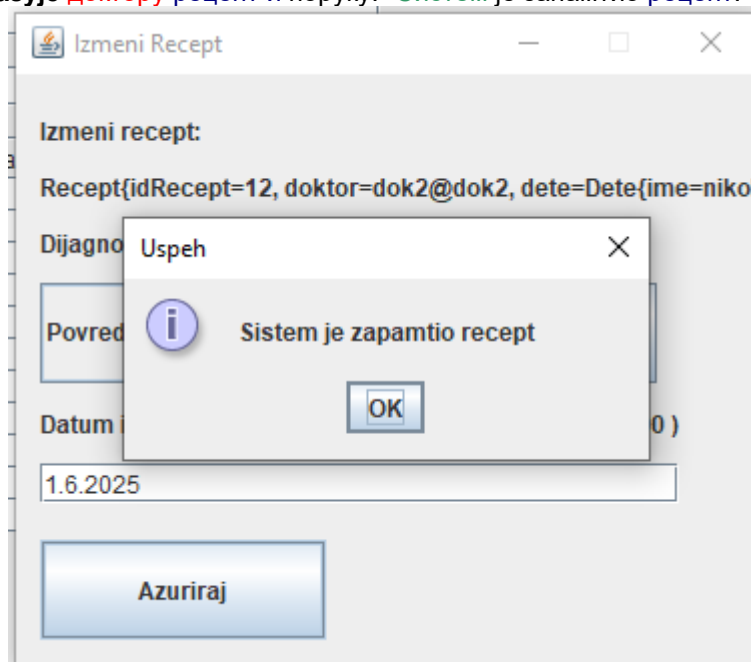
Problemi sa prilagođavanjem u školi

Noćni strahovi i noćne more

Слика 31. Мењање података о рецепту

73. **Доктор** контролише да ли је коректно унео податке о **рецепту**. (АНСО)
74. **Доктор** позива **систем** да запамти податке о **рецепту**. (АПСО)
75. **Систем** памти податке о **рецепту**. (СО)


76. Систем приказује доктору рецепт и поруку: “Систем је запамтио рецепт.” (ИА)




Слика 32. Успешна измена рецепта

Алтернативна сценарија:

4.1 Уколико **систем** не може да нађе **рецепте** он **приказује доктору** поруку: “**Систем** не може да нађе **рецепте** по задатом критеријуму”. Прекида се извршење сценарија. (ИА)

 Prikaz izdatih recepta

Dijagnoza:	<input type="text"/>	
Datum izdavanja:	<input type="text"/>	(1.1.2001)
Ime deteta:	<input type="text" value="nikolaASasSA"/>	
Prezime deteta:	<input type="text"/>	
Ime doktora:	<input type="text"/>	
Prezime doktora:	<input type="text"/>	<input type="button" value="Pretrazi"/> <input type="button" value="Resetuj"/>

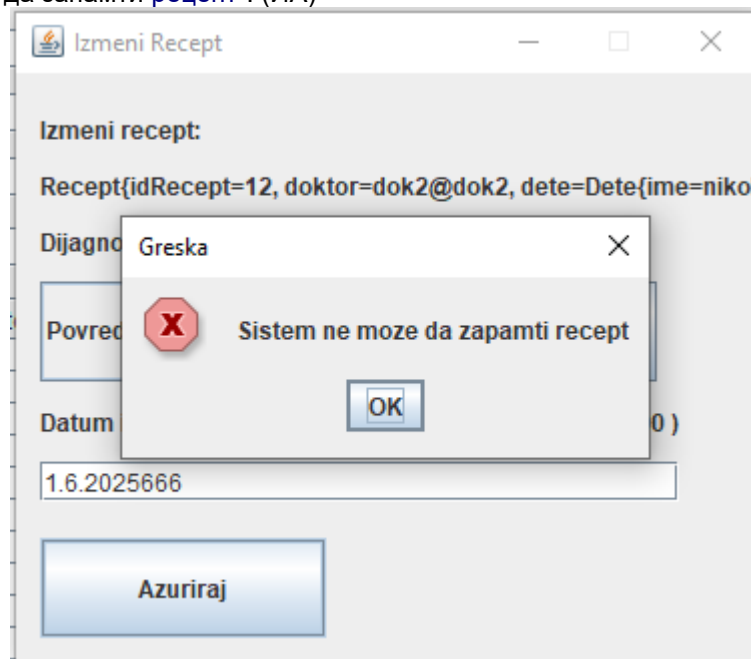
idRecept	datumIzavanja	Doktor	Dete	dijagnoza
<div><div>Greska</div><div> Sistem ne moze da nadje recept po zadatik kriterijumima</div><div><input type="button" value="OK"/></div></div>				

Слика 33. Систем је не може да нађе рецепт по задатом критеријуму

8.1 Уколико **систем** не може да нађе **рецепт** он **приказује доктору** поруку: “**Систем** не може да нађе **рецепт**”. Прекида се извршење сценарија. (ИА)

Коментар: Приликом израде софтверског решења одлучено је да неће постојати обавештење кориснику за одабир индивидуалног објекта.

13.1 Уколико **систем** не може да запамти податке о **рецепту** он **приказује** **доктору** поруку:
“**Систем** не може да запамти **рецепт**”. (ИА)



Слика 34. Систем не може да запамти рецепт

4.1.2 Пројектовање контролера корисничког интерфејса

Контролер корисничког интерфејса има следеће одговорности:

1. Прихвата податке од екранске форме.
2. Конвертује податке из графичких елемената у објекат који ће послужити као улазни аргумент системске операције (CO).
3. Прослеђује захтев за извршење системске операције до апликационог сервера (софтверског система).
4. Прихвата објекат који софтверски систем генерише као резултат извршења системске операције (CO).
5. Конвертује добијени објекат у податке који ће бити приказани у графичким елементима.

4.2 Пројектовање апликационе логике

Апликациона логика дефинише структуру и понашање система. Апликациони сервер

обухвата следеће компоненте:

1. Контролер апликационе логике – одговоран је за покретање серверског сокета који ће ослушкивати мрежне захтеве. Овај контролер служи за комуникацију са клијентом, прихвата захтеве за извршење системских операција и прослеђује их пословној логици, која је задужена за њихово извршење
2. Пословна логика – описана је кроз структуру (доменске класе) и понашање (системске операције)
3. Брокер базе података – посредује у комуникацији између пословне логике и базе података

4.2.1 Контролер апликационе логике

Контролер апликационе логике прихвата захтев за извршење системске операције од клијентске нити и даље га преусмерава до класа које су одговорне за извршење системских операција. Након извршења системских операција, контролер апликационе логике прихвата резултат и прослеђује га оном ко га позива. Свака системска операција имплементирана је као засебна класа.

4.2.2 Пословна логика

Пројектовање понашања софтверског система (системске операције)

За сваки уговор креирамо системску операцију која пројектује понашање софтверског система наслеђивањем апстрактне класе AbstractSO. Ова класа садржи методу Execute која као параметар има DomainObject и позива апстрактне методе validate и execute које свака системска операција имплементира на јединствен начин. Након провере предуслова и извршавања операције, метода commit позива истоимену методу над конекцијом DatabaseBroker-а која чува измене над подацима у бази, уколико није дошло до грешке. На основу Response објекта клијент закључује да ли је операција успешно извршена на серверској страни или је дошло до грешке.

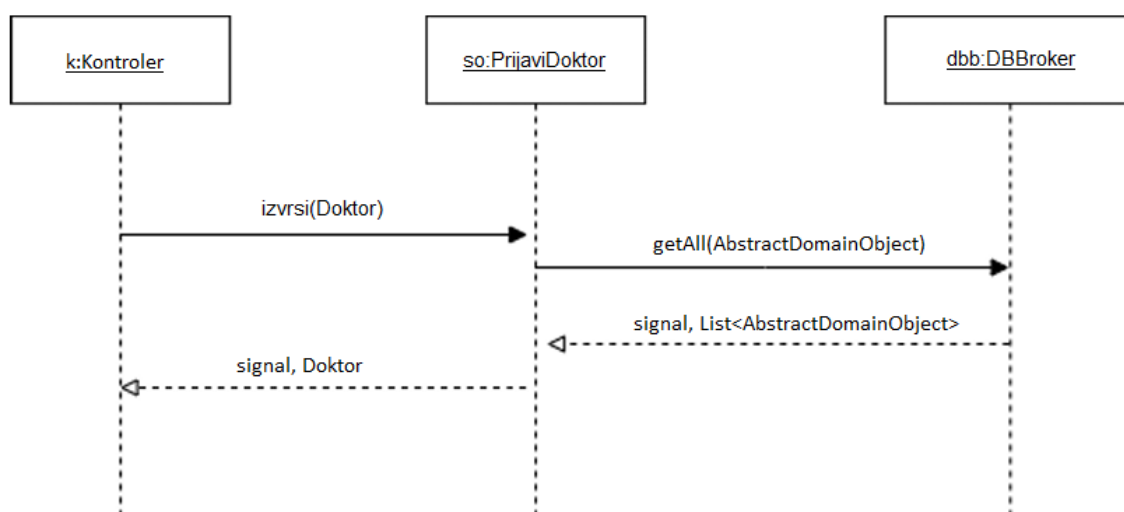
1. Уговор UG1: PrijaviDoktor(korisnickolme, sifra)

Операција: PrijaviDoktor(korisnickolme, sifra):signal;

Веза са СК: CK21

Предуслови:

Постуслови: Доктор је пријављен на систем.



Слика 35. Уговор УГ1

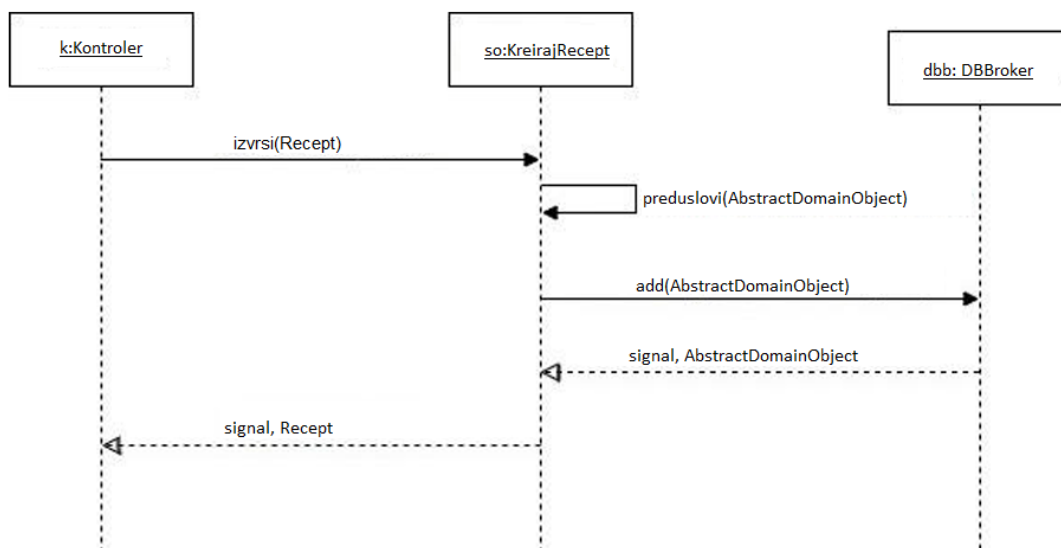
2. Уговор UG2: KreirajRecept(Recept)

Операција: KreirajRecept(Recept):signal;

Веза са СК: СК1

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

Постуслови: Направљен је нови објекат класе ЕвиденцијаКурса.



Слика 36. Уговор УГ2

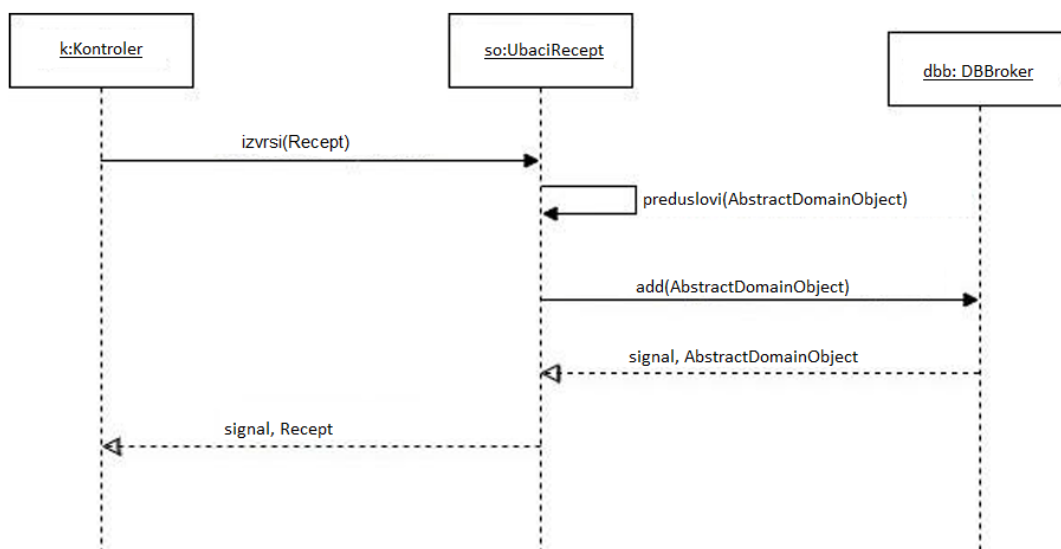
3. Уговор UG3: UbacijRecept(Recept)

Операција: UbacijRecept(Recept):signal;

Веза са СК: СК2

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

Постуслови: Направљен је нови објекат класе Рецепт.



Слика 37. Уговор УГ3

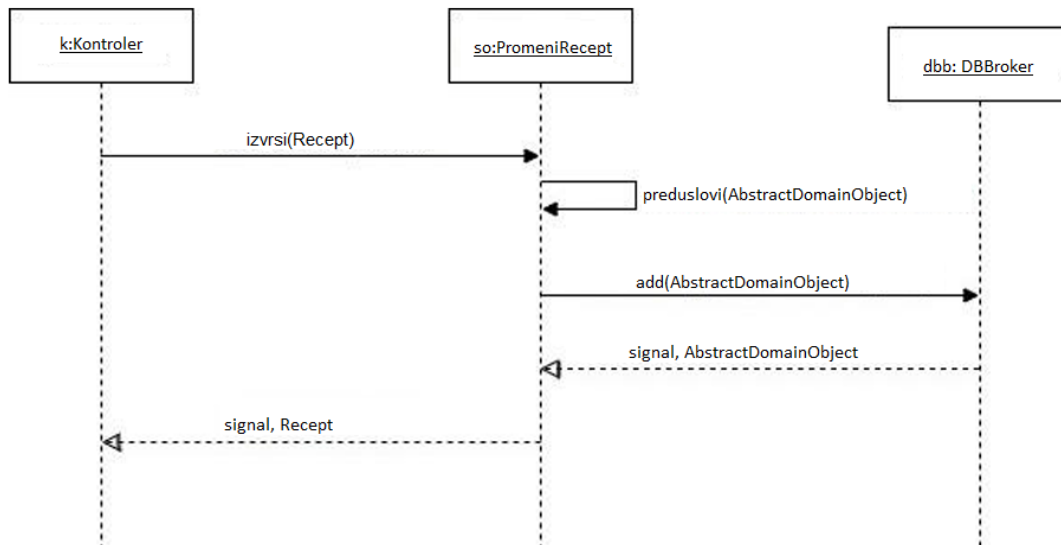
4. Уговор UG4: PromeniRecept(Recept)

Операција: PromeniRecept(Recept):signal;

Веза са СК: СК4

Предуслови: Структурна и вредносна ограничење над објектом класе Рецепт морају бити задовољена.1

Постуслови: Објекат класе Рецепт је промењен.



Слика 38. Уговор УГ4

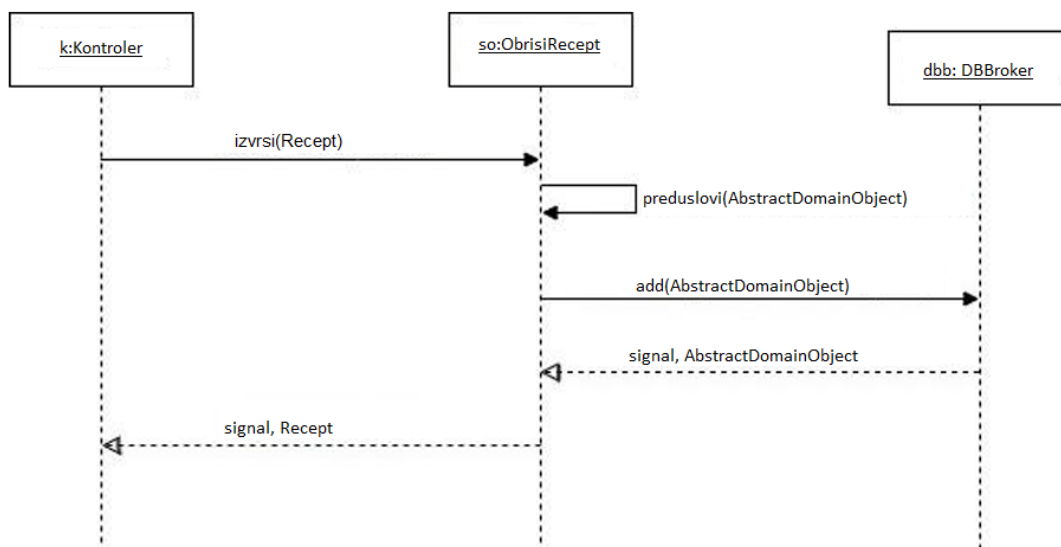
5. Уговор UG5: ObrisiRecept(Recept)

Операција: ObrisiRecept(Recept):signal;

Веза са СК: СК5

Предуслови: Структурна и вредносна ограничење над објектом класе Рецепт морају бити задовољена.2

Постуслови: Објекат класе Рецепт је обрисан.



Слика 39. Уговор УГ5

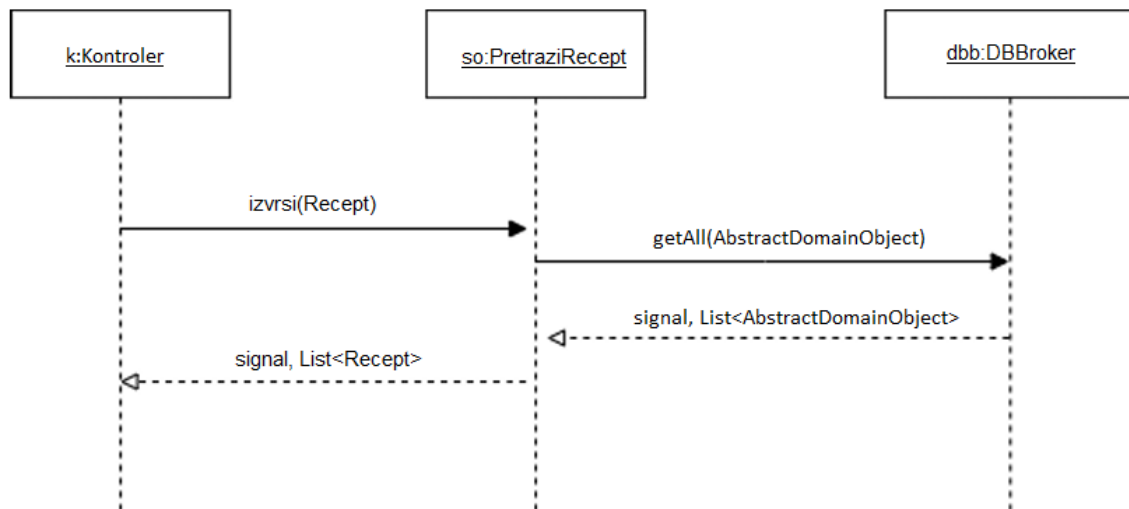
6. Уговор UG6: PretraziRecept(Recept)

Операција: PretraziRecept(Recept):signal;

Веза са СК: СК3

Предуслови:

Постуслови: Пронађен је тражени објекат класе Рецеот.



Слика 40. Уговор УГ6

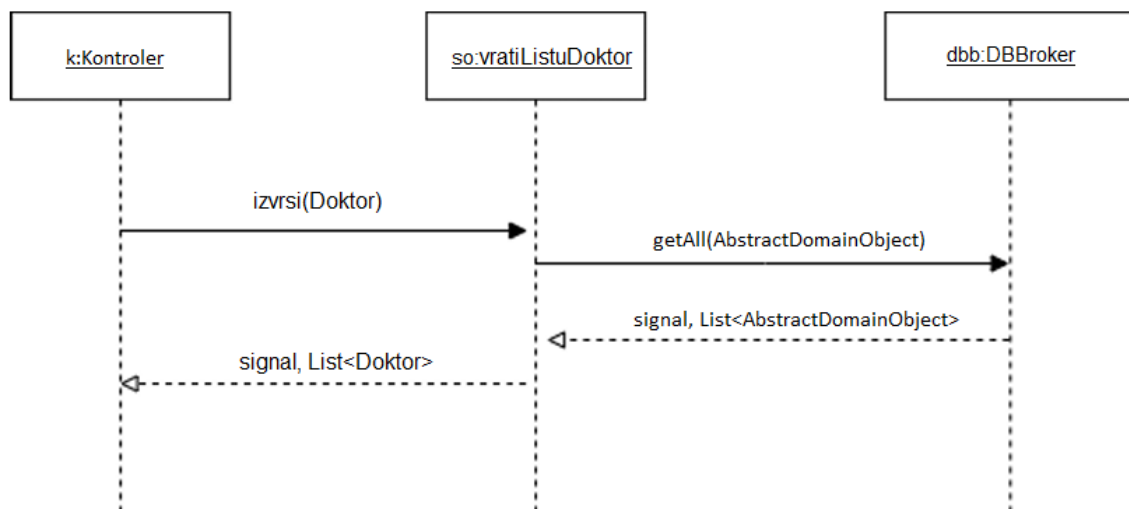
7. Уговор UG7: vratiListuDoktor(kriterijumDoktor, Lista<Doktor>)

Операција: vratiListuDoktor(kriterijumDoktor, Lista<Doktor>):signal;

Веза са СК: СК24 СК25 СК26

Предуслови:

Постуслови: Пронађена је листа тражених објеката класе Доктор.



Слика 41. Уговор УГ7

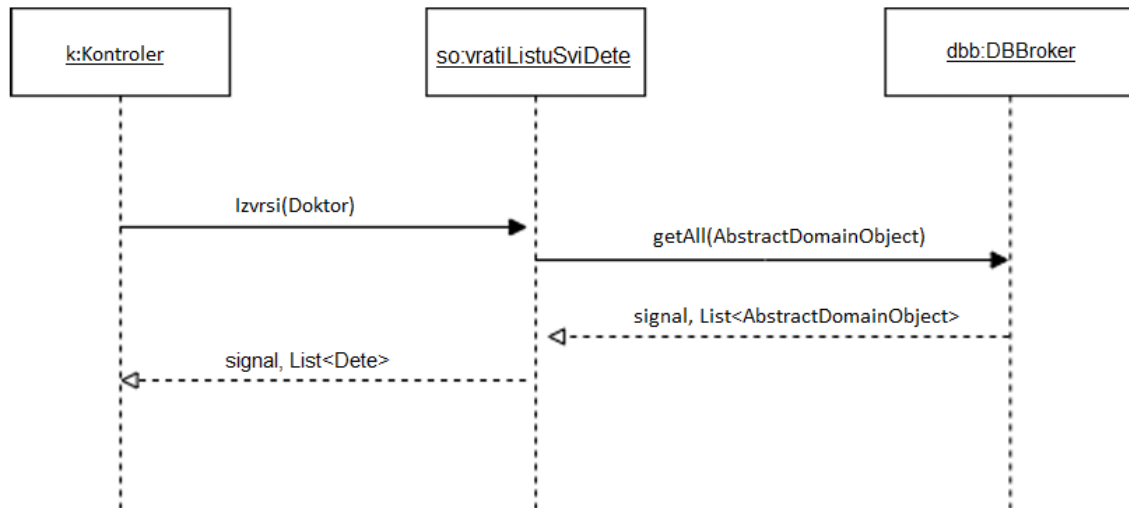
8. Уговор UG8: vratiListuSviDete(Lista<Dete>)

Операција: **vratiListuSviDete**(Lista<Dete>):signal;

Веза са СК: CK1 CK2 CK4

Предуслови:

Постуслови: Пронађена је листа свих објеката класе Дете.



Слика 42. Уговор УГ8

4.2.3 Пројектовање структуре софтверског система (доменске класе)

Софтверске класе структуре:

- Dete

```
public class Dete implements AbstractDomainObject {  
    private int idDete;  
    private String ime;  
    private String prezime;  
    private LocalDate datumRodjenja;  
  
    public Dete() {  
    }  
  
    public Dete(int idDete, String ime, String prezime, LocalDate datumRodjenja) {  
        this.idDete = idDete;  
        this.ime = ime;  
        this.prezime = prezime;  
        this.datumRodjenja = datumRodjenja;  
    }  
}
```

Слика 43. Класа Дете

- Doktor

```
public class Doktor implements AbstractDomainObject {  
    private int idDoktor;  
    private String ime;  
    private String prezime;  
    private String email;  
    private String sifra;  
  
    public Doktor() {  
    }  
  
    public Doktor(int idDoktor, String ime, String prezime, String email, String sifra) {  
        this.idDoktor = idDoktor;  
        this.ime = ime;  
        this.prezime = prezime;  
        this.email = email;  
        this.sifra = sifra;  
    }  
}
```

Слика 44. Класа Доктор

- DrSp

```
public class DrSp implements AbstractDomainObject {  
  
    private Doktor doktor;  
    private Specijalizacija specijalizacija;  
  
    public DrSp() {  
    }  
  
    public DrSp(Doktor doktor, Specijalizacija specijalizacija) {  
        this.doktor = doktor;  
        this.specijalizacija = specijalizacija;  
    }  
}
```

Слика 45. Класа Доктор-Специјализација

- Lek

```
public class Lek implements AbstractDomainObject {

    private int idLek;
    private String naziv;
    private String proizvođjac;
    private String inn;
    private String atcKlasifikacija;
    private String farmakoloskaHemijskaPodgrupa;
    private String farmaceutskeOblik;
    private String sadrzajAktivneSupstance;
    private String pakovanje;
    private String terapijskaGrupa;

    public Lek() {
    }

    public Lek(int idLek, String naziv, String proizvođjac, String inn, String atcKlasifikacija,
        String farmakoloskaHemijskaPodgrupa, String farmaceutskeOblik, String sadrzajAktivneSupstance, String pakovanje, String terapijskaGrupa) {
        this.idLek = idLek;
        this.naziv = naziv;
        this.proizvođjac = proizvođjac;
        this.inn = inn;
        this.atcKlasifikacija = atcKlasifikacija;
        this.farmakoloskaHemijskaPodgrupa = farmakoloskaHemijskaPodgrupa;
        this.farmaceutskiOblik = farmaceutskeOblik;
        this.sadrzajAktivneSupstance = sadrzajAktivneSupstance;
        this.pakovanje = pakovanje;
        this.terapijskaGrupa = terapijskaGrupa;
    }
}
```

Слика 46. Класа Лек

- PredskolskoDete

```
public class PredskolskoDete extends Dete implements AbstractDomainObject {
    private String grupa;

    public PredskolskoDete() {
    }

    public PredskolskoDete(String grupa, int idDete, String ime, String prezime, LocalDate datumRodjenja) {
        super(idDete, ime, prezime, datumRodjenja);
        this.grupa = grupa;
    }
}
```

Слика 47. Класа ПредшколскоДете

- SkolskoDete

```
public class SkolskoDete extends Dete implements AbstractDomainObject {

    private String odeljenje;
    private String razred;

    public SkolskoDete() {
    }

    public SkolskoDete(String odeljenje, String razred, int idDete, String ime, String prezime, LocalDate datumRodjenja) {
        super(idDete, ime, prezime, datumRodjenja);
        this.odeljenje = odeljenje;
        this.razred = razred;
    }
}
```

Слика 48. Класа ШколскоДете

- Specijalizacija

```
public class Specijalizacija implements AbstractDomainObject {

    private int idSpecijalizacija;
    private String naziv;

    public Specijalizacija() {
    }

    public Specijalizacija(int idSpecijalizacija, String naziv) {
        this.idSpecijalizacija = idSpecijalizacija;
        this.naziv = naziv;
    }
}
```

Слика 49. Класа Специјализација

- StavkaRecepta

```
public class StavkaRecepta implements AbstractDomainObject {

    private int rb;
    private Recept recept;
    private Lek lek;
    private String terapija;
    private String zakljucak;

    public StavkaRecepta() {
    }

    public StavkaRecepta(int rb, Recept recept, Lek lek, String terapija, String zakljucak) {
        this.rb = rb;
        this.recept = recept;
        this.lek = lek;
        this.terapija = terapija;
        this.zakljucak = zakljucak;
    }
}
```

Слика 50. Класа СтавкаРецепта

- AbstractDomainObject

```
public interface AbstractDomainObject extends Serializable {

    public String returnTableName();

    public List<AbstractDomainObject> returnListFromRS(ResultSet rs) throws Exception;

    public AbstractDomainObject returnObjectFromRS(ResultSet rs) throws Exception;

    public String vratiKoloneZaUbacivanje();

    public String vratiVrednostZaUbacivanje();

    public String vratiPrimarniKljuc();

    public String vratiVrednostiZaIzmenu();
}
```

Слика 51. Класа Апстрактни Доменски Објекат

- Operation

```
public enum Operation implements Serializable {
    LOGIN,
    UCITAJ_DECU,
    OBRISI_DETE,
    DODAJ_DETE,
    IZMENI_DETE,
    UCITAJ_RECEPTE,
    UCITAJ_STAVKE,
    DODAJ_PREDSKOLSKODETE,
    IZMENI_PREDSKOLSKODETE,
    DODAJ_SKOLSKODETE,
    IZMENI_SKOLSKODETE,
    UCITAJ_SKOLSKUDECU,
    UCITAJ_PREDSKOLSKUDECU,
    OBRISI_SKOLSKODETE,
    OBRISI_PREDSKOLSKODETE,
    LOGOUT,
    OBRISI_STAVKURECEPTA,
    DODAJ_STAVKURECEPTA,
    OBRISI_RECEPT,
    DODAJ_RECEPT,
    UCITAJ_LEKOVE,
    IZMENI_RECEPT
}
```

Слика 52. Класа Операција

- Request

```
public class Request implements Serializable {
    private Operation operation;
    private Object payload;

    public Request() {
    }

    public Request(Operation operation, Object payload) {
        this.operation = operation;
        this.payload = payload;
    }
}
```

Слика 53. Класа Захтев

- Response

```
public class Response implements Serializable {
    private Object payload;

    public Response() {
    }

    public Response(Object payload) {
        this.payload = payload;
    }
}
```

Слика 54. Класа Одговор

4.3 Пројектовање складишта података

На основу релационог модела и ограничења пројектоване су табеле базе података које користи наш софтверски систем:

- Табела дете

idDete	int	10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ime	varchar	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
prezime	varchar	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
datumRodjenja	date		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Слика 55. Табела Дете

- Табела доктор

Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update
idDoktor	int	10		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ime	varchar	50		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
prezime	varchar	50		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
email	varchar	50		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
sifra	varchar	50		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Слика 56. Табела Доктор

- Табела ДрСп

idDoktor	int	10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
idSpecijalizacija	int	10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Слика 57. Табела ДрСп

- Табела Лек

idLek	int	10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
naziv	varchar	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
proizvodjac	varchar	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
inn	varchar	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
atcKlasifikacija	varchar	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
farmakoloskaHemija	varchar	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
farmaceutskiOblik	varchar	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
sadrzajAktivneSup	varchar	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
pakovanje	varchar	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
terapijskaGrupa	varchar	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Слика 58. Табела Лек

- Табела Предшколско дете

idDete	int	10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
grupa	varchar	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Слика 59. Табела Предшколско дете

- Табела Рецепт

idRecept	int	10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
idDoktor	int	10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
idDete	int	10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
datumIzdavanja	date		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
dijagnoza	varchar	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Слика 60. Табела Рецепт

- Табела Школско дете

idDete	int	10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
odeljenje	varchar	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
razred	varchar	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Слика 61. Табела Школско дете

- Табела Специјализација

idSpecijalizacija	int	10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
naziv	varchar	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

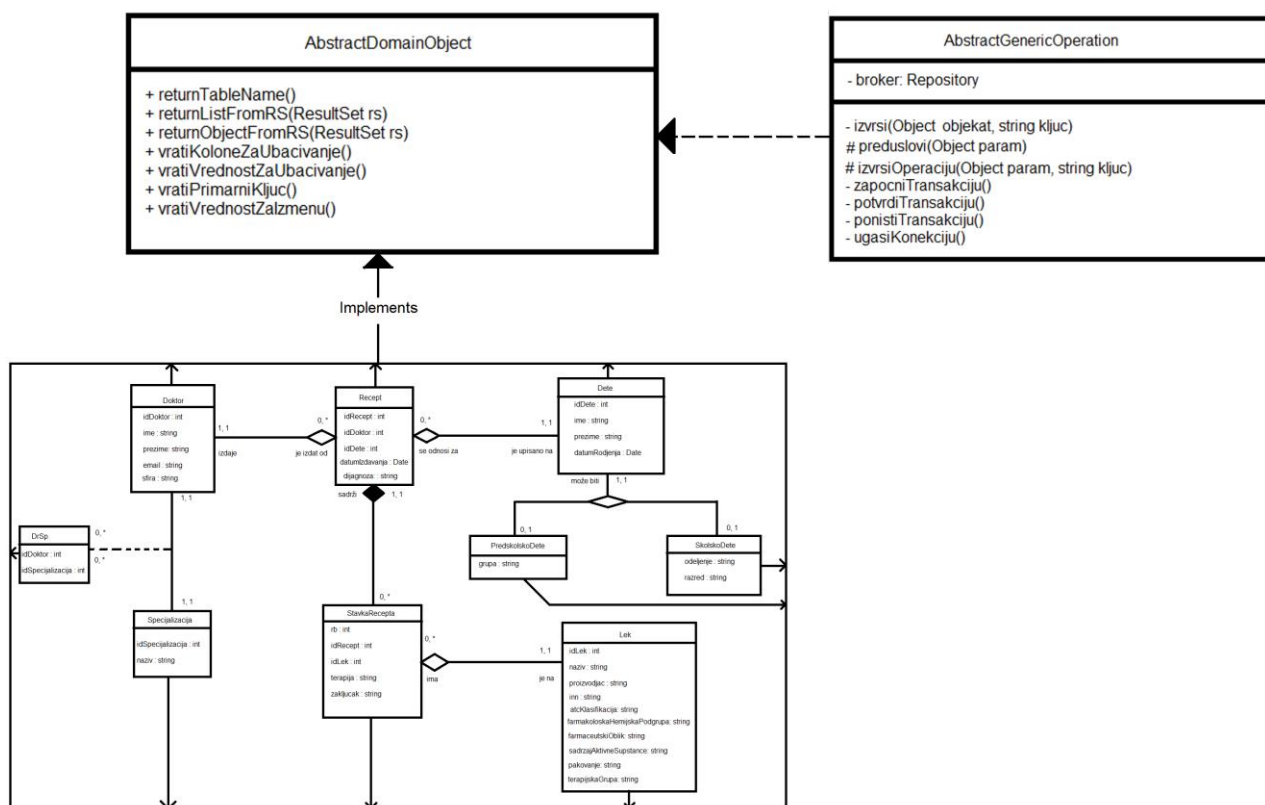
Слика 62. Табела Специјализација

- Табела Рецепт

idRecept	int	10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
rb	int	10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
idLek	int	10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
terapija	varchar	500	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
zakljucak	varchar	500	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

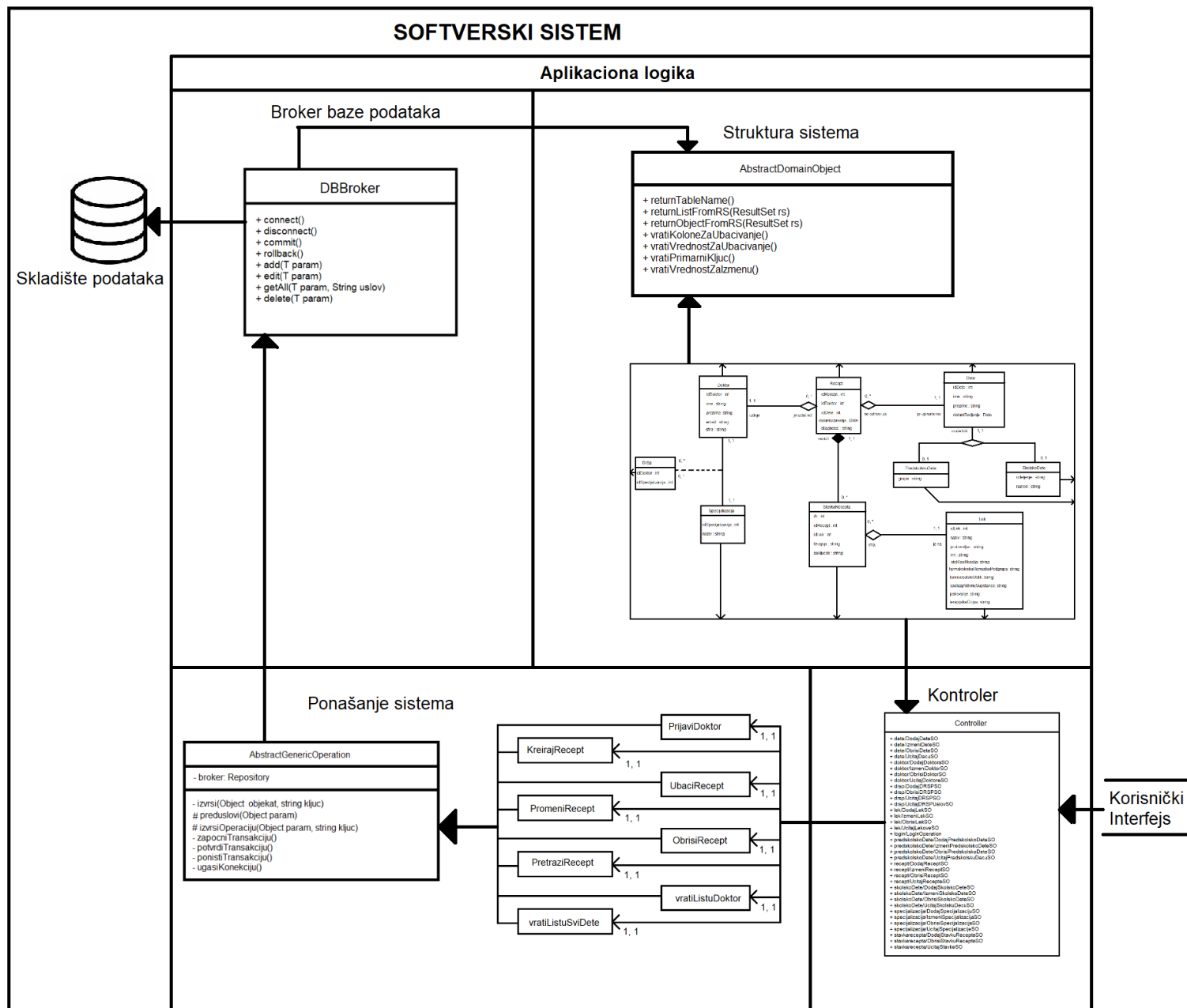
Слика 63. Табела Рецепт

За комуникацију са базом података, креирамо генеричку класу `AbstractGenericOperation`, која садржи апстрактне методе за валидацију и извршење трансакција. Те методе ће бити имплементирane у свакој специфичној класи системских операција која се буде извршавала. Поред тога, `AbstractGenericOperation` укључује и методе за потврђивање и поништавање трансакција. Класа се ослања на `DBBroker`, која је реализована помоћу Singleton патерна и која чита параметре за повезивање са базом података из текстуалне датотеке са свим потребним подацима. На овај начин, `AbstractGenericOperation` успоставља конекцију са базом података. Као резултат пројектовања класе `AbstractGenericOperation`, `DatabaseConnection` и доменских објеката добијамо следећи дијаграм класа:



Слика 64. Дијаграм класа добијен након пројектовања доменских класа и класе `AbstractDomainObject`

На основу претходних целина, може се саставити цела архитектура софтверског система за евиденцију рецепта за школску и предшколску децу у дечијој ординацији.



Слика 65. Архитектура софтверског система

5. Имплементација

Софтверски систем је развијен у програмском језику Java 18 и пројектован као клијент сервер апликација. За управљање базом података коришћен је SqlYog, док је као развојно окружење коришћен NetBeans IDE 15. На основу архитектуре система дефинисане су следеће софтверске класе:

0_SEMINARSKI_CLIENT:

- communication/Communication
- controllers/DodajDeteController
- controllers/DodajPredskolskoDeteController
- controllers/DodajSkolskoDeteController
- controllers/DodajStavkuReceptaController
- controllers/IzmeniReceptController
- controllers/LoginController
- controllers/MainController
- controllers/PrikazDeceController
- controllers/PrikazObrazovanjaController
- controllers/PrikazReceptaController
- cordinator/Cordinator
- forms/DodajDeteForm
- forms/DodajPredskolskoDeteForm
- forms/DodajSkolskoDeteForm
- forms/DodajStavkuReceptaForm
- forms/FormMode
- forms/IzmeniReceptForm
- forms/LoginForm
- forms/MainForm
- forms/PrikazDeceForm
- forms/PrikazObrazovanjaForm
- forms/PrikazReceptaForm
- forms.models/ModelTabeleDeca
- forms.models/ModelTabeleRecepti
- forms.models/ModelTabeleStavke
- main/Main

0_SEMINARSKI_COMMON:

- domain/AbstractDomainObject
- domain/Dete
- domain/Doktor
- domain/DrSp
- domain/Lek
- domain/PredskolskoDete
- domain/Recept
- domain/SkolskoDete
- domain/Specijalizacija
- domain/StavkaRecepta
- network/Operation
- network/Receiver
- network/Request
- network/Response

network/Sender

0_SEMINARSKI_SERVER:

controller/Controller

cordinator/Cordinator

forms/DatabaseDialog

forms/DodajDoktoraForm

forms/DodajLekForm

forms/FormMode

forms/PortDialog

forms/PrikazDoktoraForm

forms/PrikazLekovaForm

forms/PrikazSpecijalizacijaForm

forms/ServerForm

forms.controllers/DodajDoktoraController

forms.controllers/DodajLekController

forms.controllers/PrikazDoktoraController

forms.controllers/PrikazLekovaController

forms.controllers/PrikazSpecijalizacijaController

forms.models/ModelTabeleDoktori

forms.models/ModelTabeleLekovi

forms.models/ModelTabeleSpecijalizacije

konfiguracija/Konfiguracija

main/Main

niti/ObradaKlijentskihZahteva

operations/AbstractGenericOperation

operations.dete/DodajDeteSO

operations.dete/IzmeniDeteSO

operations.dete/ObrisiDeteSO

operations.dete/UcitajDecuSO

operations.doktor/DodajDoktoraSO

operations.doktor/IzmeniDoktorSO

operations.doktor/ObrisiDoktorSO

operations.doktor/UcitajDoktoreSO

operations.drsp/DodajDRSPSO

operations.drsp/ObrisiDRSPSO

operations.drsp/UcitajDRSPSO

operations.drsp/UcitajDRSPUsllovSO

operations.lek/DodajLekSO

operations.lek/IzmeniLekSO

operations.lek/ObrisiLekSO

operations.lek/UcitajLekoveSO

operations.login/LoginOperation

operations.predskolskoDete/DodajPredskolskoDeteSO

operations.predskolskoDete/IzmeniPredskolskoDeteSO

operations.predskolskoDete/ObrisiPredskolskoDeteSO

operations.predskolskoDete/UcitajPredskolskuDecuSO

operations.recept/DodajReceptSO

operations.recept/IzmeniReceptSO

operations.recept/ObrisiReceptSO

operations.recept/UcitajRecepteSO

operations.skolskoDete/DodajSkolskoDeteSO
operations.skolskoDete/IzmeniSkolskoDeteSO
operations.skolskoDete/ObrisiSkolskoDeteSO
operations.skolskoDete/UcitajSkolskuDecuSO
operations.specijalizacija/DodajSpecijalizacijuSO
operations.specijalizacija/IzmeniSpecijalizacijaSO
operations.specijalizacija/ObrisiSpecijalizacijaSO
operations.specijalizacija/UcitajSpecijalizacijeSO
operations.stavkarecepta/DodajStavkuReceptaSO
operations.stavkarecepta/ObrisiStavkuReceptaSO
operations.stavkarecepta/UcitajStavkeSO
repository/Repository
repository.db/DbConnectionFactory
repository.db/DbRepository
repository.db.impl/DbRepositoryGeneric
server/Server

6. Тестирање

Након имплементације извршено је тестирање софтверског система. Тестирање је извршено уношењем и валидних и невалидних података. Уношени су валидни подаци, како би се тестирали сви дефинисани случајеви коришћења. Уношењем неодговарајућих и неисправних података, вршене су провере исправности валидација и логике проблема. Сви откривени недостаци су исправљени, како би софтверски систем нормално и неометано функционисао.

7. Закључак

Пројекат је реализован применом упрошћене Ларманове методе, кроз фазе прикупљања захтева, анализе, пројектовања и имплементације софтверског система за евиденцију пацијената у дечијој ординацији. Систем покрива основне функционалности и може да послужи као добра основа за даље проширење и унапређење.

Током рада, највише ме је изненадила улога документације. Испоставило се да добро написана документација значајно олакшава цео процес — кад једном постоји јасна структура, она функционише као упутство које програмер може директно да прати. Тај део ми је био посебно занимљив јер показује колико унапред осмишљена логика може да убрза развој и смањи број одлука које се доносе у ходу.

Рад на овом пројекту био је прилика да у пракси повежем знања из више предмета и применим их на конкретан пример, кроз развој једноставног, али функционалног софтверског решења.

Литература

Влајић, С. (2024). Пројектовање софтвера (скрипта). Београд, Србија: Факултет организационих наука