

Универзитет у Београду
Факултет организационих наука
Лабораторија за софтверско инжењерство

Семинарски рад из предмета
Пројектовање софтвера

Тема: Развој софтвера за медицинске установе
у Јава окружењу

Ментор:
Проф. др. Синиша Влајић

Студент:
Маја Павловић 0226/2018

Београд, 2021.

Садржај

Прикупљање корисничких захтева	3
Вербални опис система.....	3
Случајеви коришћења	3
СК1: Случај коришћења – Креирање картона пацијента	4
СК2: Случај коришћења – Претраживање картона пацијента	5
СК3: Случај коришћења – Измена података у картону пацијента	6
СК4: Случај коришћења – Креирање упута за лабораторијске анализе – сложен СК ...	7
СК5: Случај коришћења – Преузимање упута за лабораторијске анализе	8
СК6: Случај коришћења – Унос резултата лабораторијских анализа – сложен СК	9
СК7: Случај коришћења – Преузимање резултата лабораторијских анализа	10
Анализа.....	11
Системски дијаграми секвенци.....	11
ДС1: Дијаграм секвенци случаја коришћења – Креирање картона пацијента	11
ДС2: Дијаграм секвенци случаја коришћења – Претраживање картона пацијента	13
ДС3: Дијаграм секвенци случаја коришћења – Измена података у картону пацијента	14
ДС4: Дијаграм секвенци случаја коришћења – Креирање упута за лабораторијске анализе – сложен СК	16
ДС5: Дијаграм секвенци случаја коришћења – Преузимање упута за лабораторијске анализе	17
ДС6: Дијаграм секвенци случаја коришћења – Унос резултата лабораторијских анализа – сложен СК	18
ДС7: Дијаграм секвенци случаја коришћења – Преузимање резултата лабораторијских анализа	19
Понашање софтверског система – Дефинисање уговора о системским апликацијама ...	21
Структура софтверског система – Концептуални (доменски) модел	22
Структура софтверског система – Релациони модел.....	22
Структурна и вредносна ограничења	23
Софтверски систем за медицинске установе	26

Прикупљање корисничких захтева

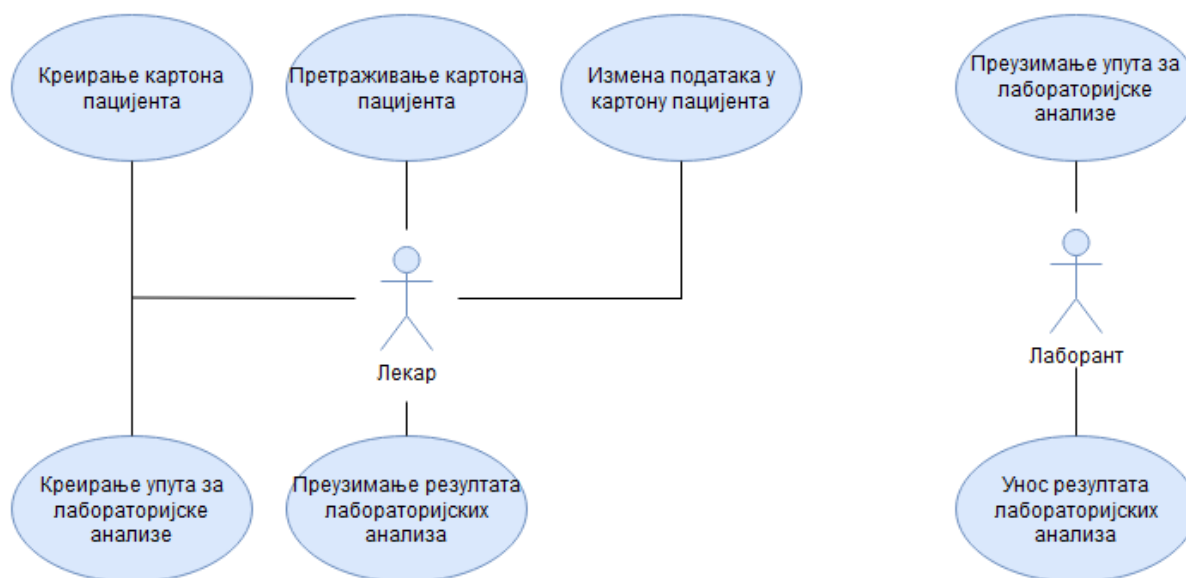
Вербални опис система

Медицинским установама неопходна је софтверска подршка како би се олакшала и убрзала међусобна комуникација. Систем који развијамо ће омогућити лекарима који раде са пацијентима да ефикасније комуницирају са лабораторијама које обављају неопходне анализе. У систему ће бити чувани картони свих пацијената који се лече у датој установи. Лекари ће имати опцију да уносе податке о новим пацијентима и претражују картоне постојећих, као и да мењају основне податке о пацијентима по потреби. Моћи ће да помоћу система издају упуте за неопходне лабораторијске анализе. С друге стране, лаборанти ће са система преузимати упуте и након урађених анализа уписивати резултате у систем.

Случајеви коришћења

Разликујемо следеће случајеве коришћења:

1. Креирање картона пацијента
2. Претраживање картона пацијента
3. Измена података у картону пацијента
4. Креирање упута за лабораторијске анализе – сложен СК
5. Преузимање упута за лабораторијске анализе
6. Унос резултата лабораторијских анализа – сложен СК
7. Преузимање резултата лабораторијских анализа



СК1: Случај коришћења – Креирање картона пацијента

Назив СК

Креирање картона пацијента

Актори СК

Лекар

Учесници СК

Лекар и систем (програм)

Предуслов: Систем је укључен и лекар је улогован под својом шифром. Систем приказује форму за рад са картоном пацијента.

Основни сценарио СК

1. Лекар уноси податке у картон пацијента. (АПУСО)
2. Лекар контролише да ли је коректно унео податке у картон пацијента. (АНСО)
3. Лекар позива систем да запамти податке о картону пацијента. (АПСО)
4. Систем памти податке о картону пацијента. (СО)
5. Систем приказује лекару запамћени картон пацијента и поруку: “Систем је запамтио картон пацијента”. (ИА)

Алтернативна сценарија

- 3.1 Уколико систем не може да креира картон пацијента он приказује лекару поруку: “Систем не може да креира картон пацијента”. Прекида се извршење сценарија. (ИА)
- 8.1 Уколико систем не може да запамти податке о картону пацијента он приказује лекару поруку “Систем не може да запамти картон пацијента”. (ИА)

СК2: Случај коришћења – Претраживање картона пацијента

Назив СК

Претраживање картона пацијента

Актори СК

Лекар

Учесници СК

Лекар и систем (програм)

Предуслов: Систем је укључен и лекар је улогован под својом шифром. Систем приказује форму за рад са картоном пацијента.

Основни сценарио СК

1. Лекар уноси вредност по којој претражује картон пацијента. (АПУСО)
2. Лекар позива систем да нађе картон пацијента по задатој вредности. (АПСО)
3. Систем тражи картон пацијента по задатој вредности. (СО)
4. Систем приказује лекару податке о картону пацијента и поруку: “Систем је нашао картон пацијента по задатој вредности”. (ИА)

Алтернативна сценарија

- 4.1 Уколико систем не може да нађе картон пацијента он приказује лекару поруку: “Систем не може да нађе картон пацијента по задатој вредности”. (ИА)

СК3: Случај коришћења – Измена података у картону пацијента

Назив СК

Измена података у картону пацијента

Актори СК

Лекар

Учесници СК

Лекар и систем (програм)

Предуслов: Систем је укључен и лекар је улогован под својом шифром. Систем приказује форму за рад са картоном пацијента.

Основни сценарио СК

1. Лекар уноси вредност по којој претражује картон пацијента. (АПУСО)
2. Лекар позива систем да нађе картон пацијента по задатој вредности. (АПСО)
3. Систем тражи картон пацијента по задатој вредности. (СО)
4. Систем приказује лекару картон пацијента и поруку: “Систем је нашао картон пацијента по задатој вредности”. (ИА)
5. Лекар уноси (мења) податке у картону пацијента. (АПУСО)
6. Лекар контролише да ли је коректно унео податке у картон пацијента. (АНСО)
7. Лекар позива систем да запамти податке о картону пацијента. (АПСО)
8. Систем памти податке о картону пацијента. (СО)
9. Систем приказује лекару запамћени картон пацијента и поруку: “Систем је запамтио картон пацијента.” (ИА)

Алтернативна сценарија

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

9.1 Уколико систем не може да запамти податке о картону пацијента он приказује лекару поруку “Систем не може да запамти картон пацијента”.(ИА)

СК4: Случај коришћења – Креирање упута за лабораторијске анализе – сложен СК

Назив СК

Креирање упута за лабораторијске анализе

Актори СК

Лекар

Учесници СК

Лекар и систем (програм)

Предуслов: Систем је укључен и лекар је улогован под својом шифром. Систем приказује форму за унос упута и листу могућих анализа и врста узорака.

Основни сценарио СК

1. Лекар уноси податке о пацијенту и потребним анализама у упут. (АПУСО)
2. Лекар контролише да ли је коректно унео податке у упут. (АНСО)
3. Лекар позива систем да запамти податке о упуту. (АПСО)
4. Систем памти податке о упуту. (СО)
5. Систем приказује лекару запамћени упут и поруку: “Систем је запамтио упут”. (ИА)

Алтернативна сценарија

- 5.1 Уколико систем не може да запамти податке о упуту он приказује лекару поруку “Систем не може да запамти упут”. (ИА)

СК5: Случај коришћења – Преузимање упута за лабораторијске анализе

Назив СК

Преузимање упута за лабораторијске анализе

Актори СК

Лаборант

Учесници СК

Лаборант и систем (програм)

Предуслов: Систем је укључен и лаборант је улогован под својом шифром. Систем приказује листу са упутима.

Основни сценарио СК

1. Лаборант бира упут који ће да прочита. (АПУСО)
2. Лаборант позива систем да отвори задати упут за читање. (АПСО)
3. Систем тражи упут по задатој вредности. (СО)
4. Систем приказује лаборанту податке о упуту. (ИА)

Алтернативна сценарија

- 4.1 Уколико систем не може да отвори упут он приказује лаборанту поруку: “Систем не може да отвори дати упут”. (ИА)

СК6: Случај коришћења – Унос резултата лабораторијских анализа – сложен СК

Назив СК

Унос резултата лабораторијских анализа

Актори СК

Лаборант

Учесници СК

Лаборант и систем (програм)

Предуслов: Систем је укључен и лаборант је улогован под својом шифром. Систем приказује форму за унос резултата.

Основни сценарио СК

1. Лаборант уноси податке о резултату у форму система. (АПУСО)
2. Лаборант контролише да ли је коректно унео податке у резултат. (АНСО)
3. Лаборант позива систем да запамти резултат. (АПСО)
4. Систем памти резултат. (СО)
5. Систем приказује лаборанту запамћени резултат и поруку: “Систем је запамтио резултат”. (ИА)

Алтернативна сценарија

- 5.1 Уколико систем не може да запамти податке о резултату он приказује лаборанту поруку “Систем не може да запамти резултат”. Прекида се извршење сценарија. (ИА)

СК7: Случај коришћења – Преузимање резултата лабораторијских анализа

Назив СК

Преузимање резултата лабораторијских анализа

Актори СК

Лекар

Учесници СК

Лекар и систем (програм)

Предуслов: Систем је укључен и лекар је улогован под својом шифром. Систем приказује листу са резултатима.

Основни сценарио СК

1. Лекар уноси вредност по којој претражује резултат. (АПУСО)
2. Лекар позива систем да нађе резултат по задатој вредности. (АПСО)
3. Систем тражи резултат по задатој вредности. (СО)
4. Систем приказује лекару податке о резултату и поруку: “Систем је нашао резултат по задатој вредности”. (ИА)

Алтернативна сценарија

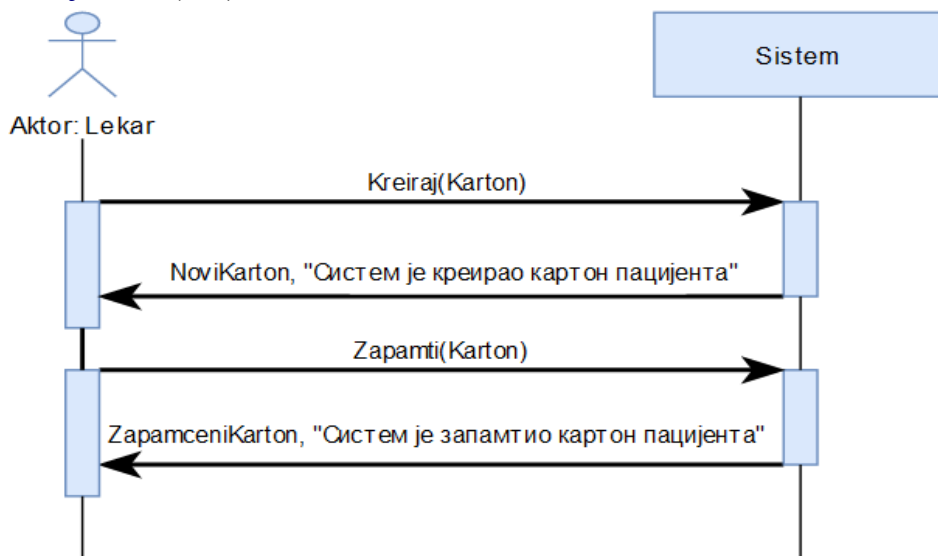
- 4.1 Уколико систем не може да нађе резултат он приказује лекару поруку: “Систем не може да нађе резултат по задатој вредности”. (ИА)

Анализа

Системски дијаграми секвенци

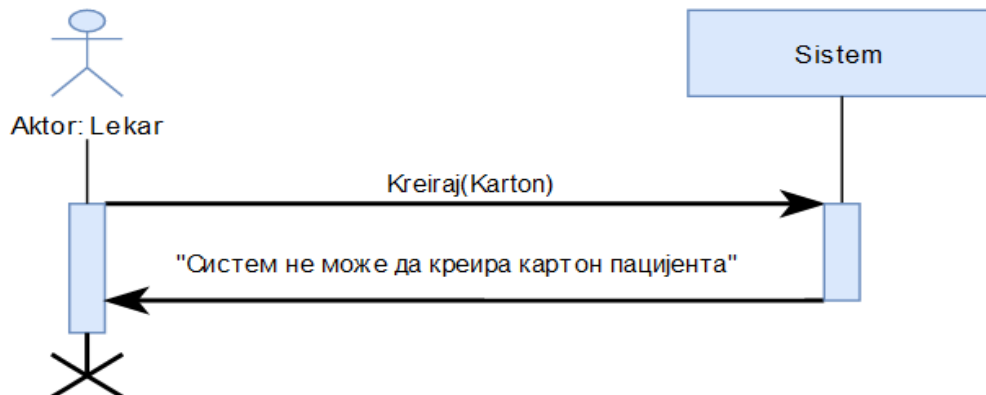
ДС1: Дијаграм секвенци случаја коришћења – Креирање картона пацијента

1. **Лекар** позива **систем** да креира **картон пацијента**. (АПСО)
2. **Систем** приказује **лекару** картон пацијента и поруку: “**Систем** је креирао картон пацијента”. (ИА)
3. **Лекар** позива **систем** да запамти податке о **картону пацијента**. (АПСО)
4. **Систем** приказује **лекару** запамћени **картон пацијента** и поруку: “**Систем** је запамтио картон пацијента”. (ИА)

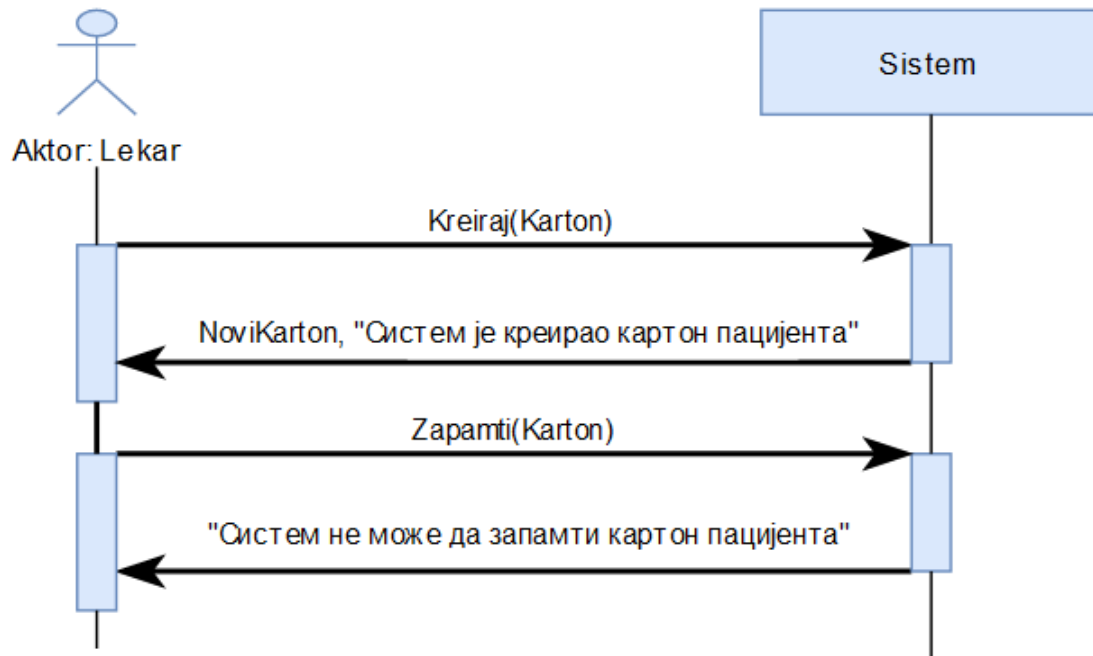


Алтернативна сценарија

- 2.1 Уколико **систем** не може да креира **картон пацијента** он приказује **лекару** поруку: “**Систем** не може да креира **картон пацијента**”. Прекида се извршење сценарија. (ИА)



4.1 Уколико **систем** не може да запамти податке о **картону пацијента** он приказује **лекару** поруку “**Систем** не може да запамти **картон пацијента**”. (ИА)

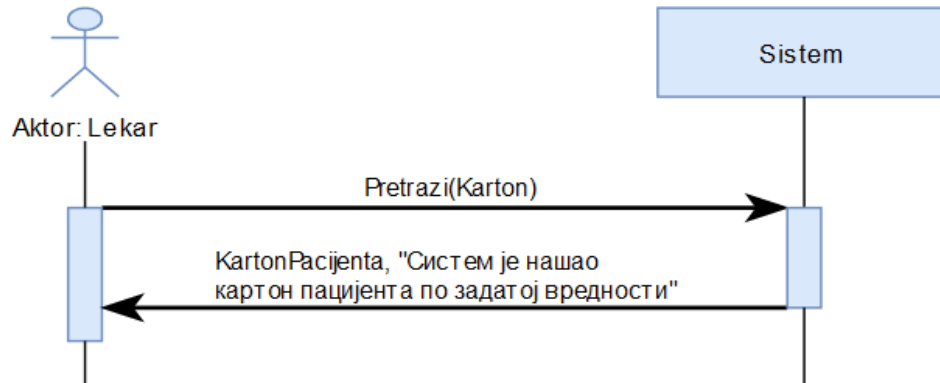


Из наведених секвенцих дијаграма уочавају се две системске операције које је потребно пројектовати:

1. Signal Kreiraj(Karton);
2. Signal Zapamti(Karton);

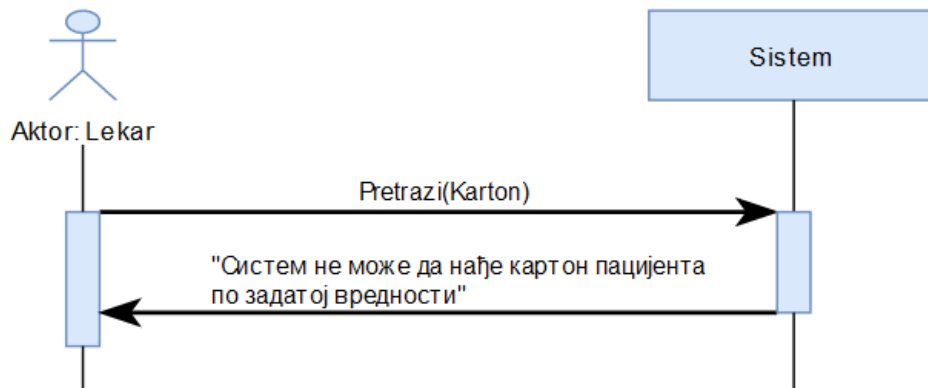
ДС2: Дијаграм секвенци случаја коришћења – Претраживање картона пацијента

1. **Лекар** позива **систем** да нађе **картон пацијента** по задатој вредности. (АПСО)
2. **Систем** приказује **лекару** податке о **картону пацијента** и поруку: “**Систем** је нашао **картон пацијента** по задатој вредности”. (ИА)



Алтернативна сценарија

- 2.1 Уколико **систем** не може да нађе **картон пацијента** он приказује **лекару** поруку: “**Систем** не може да нађе **картон пацијента** по задатој вредности”. (ИА)

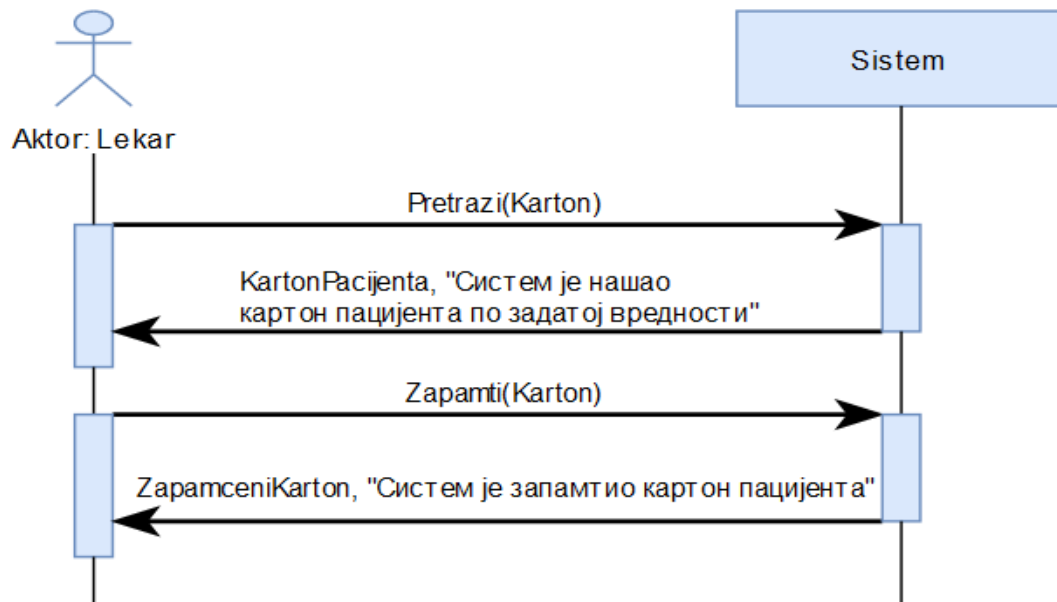


Из наведених секвенцих дијаграма уочава се системска операција коју је потребно пројектовати:

1. Signal Pretrazi(Karton);

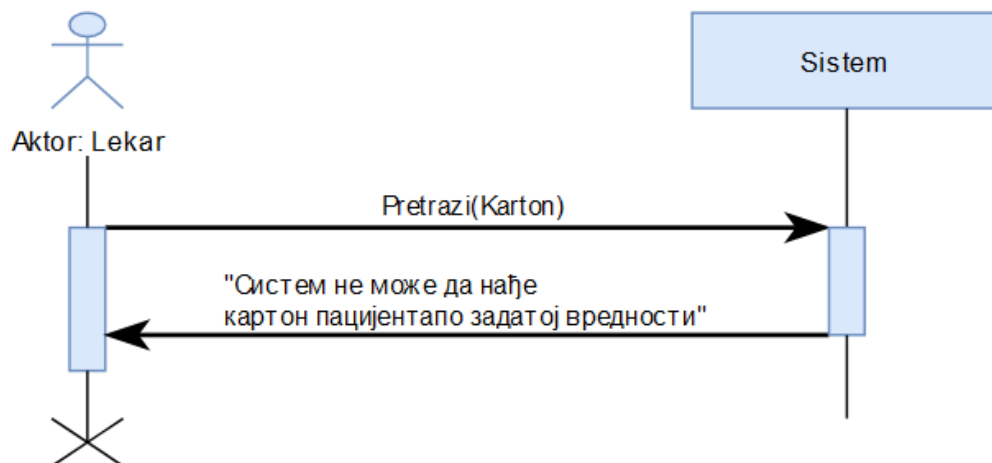
ДСЗ: Дијаграм секвенци случаја коришћења – Измена података у картону пацијента

1. **Лекар** **позива** **систем** да нађе **картон пацијента** по задатој вредности. (АПСО)
2. **Систем** приказује **лекару** **картон пацијента** и поруку: “**Систем** је нашао **картон пацијента** по задатој вредности”. (ИА)
3. **Лекар** **позива** **систем** да запамти податке о **картону пацијента**. (АПСО)
4. **Систем** **приказује** **лекару** запамћени **картон пацијента** и поруку: “**Систем** је запамтио **картон пацијента**.” (ИА)

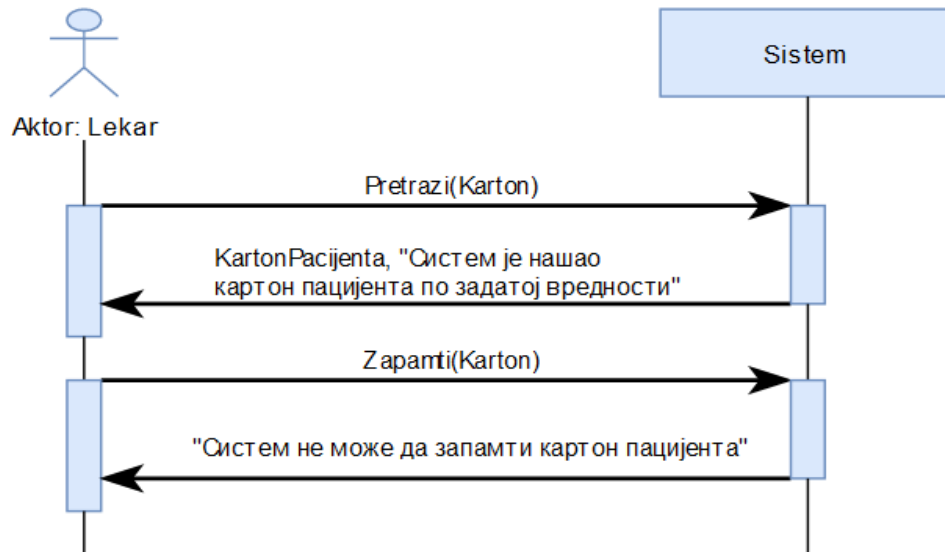


Алтернативна сценарија

- 2.1 Уколико **систем** не може да нађе **картон пацијента** он приказује **лекару** поруку: “**Систем** не може да нађе **картон пацијента** по задатој вредности”. Прекида се извршење сценарија. (ИА)



4.1 Уколико **систем** не може да запамти податке о **картону пацијента** он приказује **лекару** поруку “**Систем** не може да запамти **картон пацијента**”.(ИА)

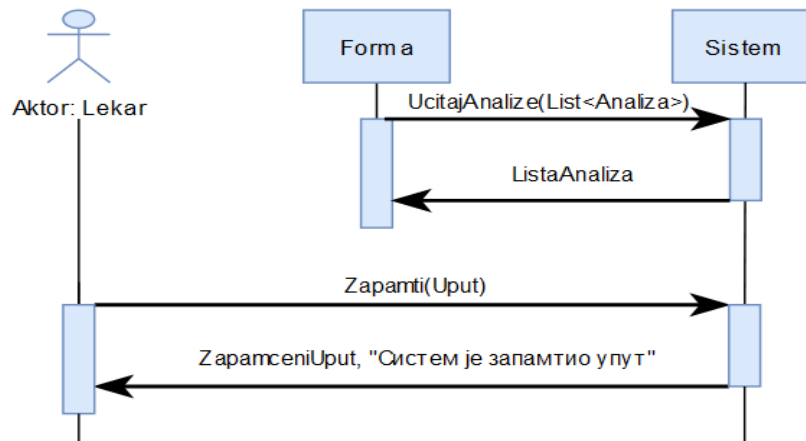


Из наведених секвенчних дијаграма уочавају се две системске операције које је потребно пројектовати:

1. Signal Pretrazi(Karton);
2. Signal Zapamti(Karton);

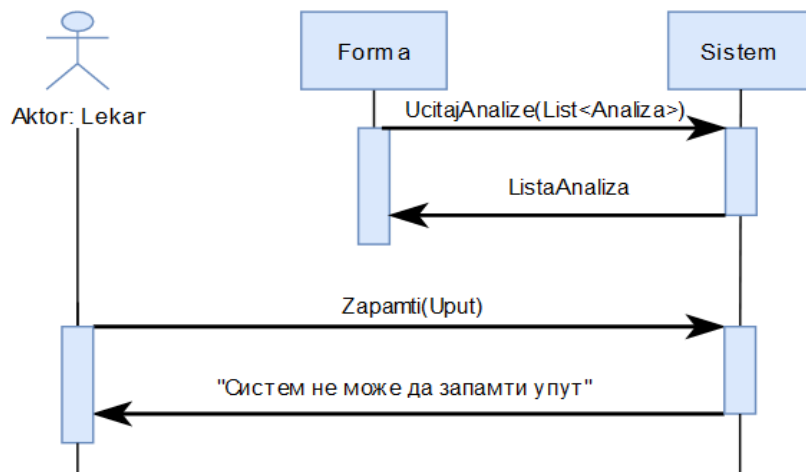
ДС4: Дијаграм секвенци случаја коришћења – Креирање упута за лабораторијске анализе – сложен СК

1. **Форма** позива **систем** да учита листу анализа. (АПСО)
2. **Систем** враћа **форми** листу анализа. (ИА)
3. **Лекар** позива **систем** да запамти податке о **упуту**. (АПСО)
4. **Систем** приказује **лекару** запамћени **упут** и поруку: “**Систем** је запамтио **упут**”. (ИА)



Алтернативна сценарија

- 4.1 Уколико **систем** не може да запамти податке о **упуту** он приказује **лекару** поруку “**Систем** не може да запамти **упут**”. (ИА)

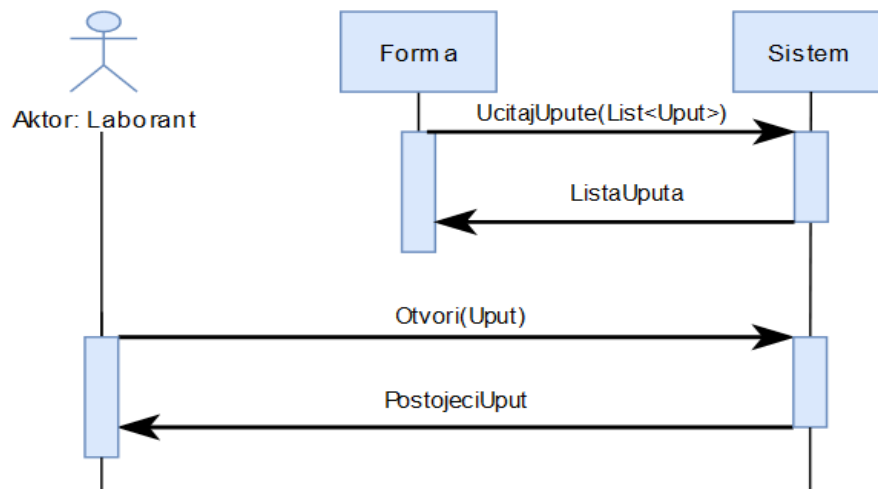


Из наведених секвенчних дијаграма уочавају се две системске операције које је потребно пројектовати:

1. Signal UcitajAnalize(List<Analiza>);
2. Signal Zapamti(Uput);

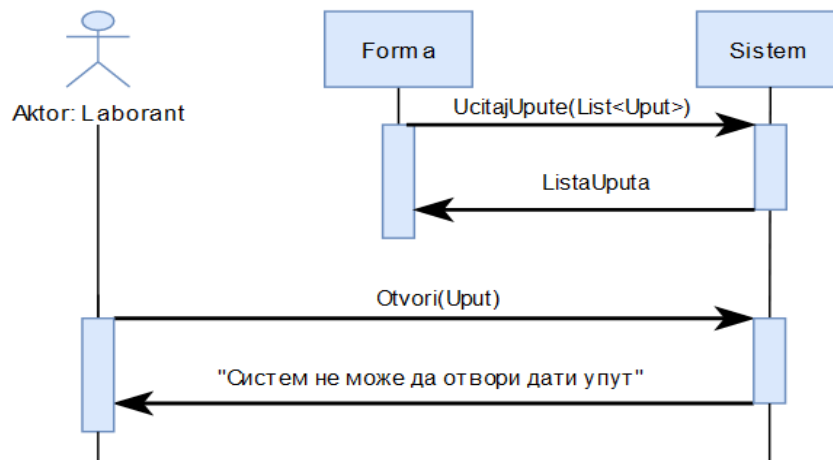
ДС5: Дијаграм секвенци случаја коришћења – Преузимање упута за лабораторијске анализе

1. **Форма** **позива** **систем** да учита листу упута. (АПСО)
2. **Систем** **враћа** **форми** листу упута. (ИА)
3. **Лаборант** **позива** **систем** да отвори задати упут за читање. (АПСО)
4. **Систем** приказује **лаборанту** податке о упуту. (ИА)



Алтернативна сценарија

- 4.1 Уколико **систем** не може да отвори **упут** он приказује **лаборанту** поруку: “**Систем** не може да отвори дати **упут**”. (ИА)

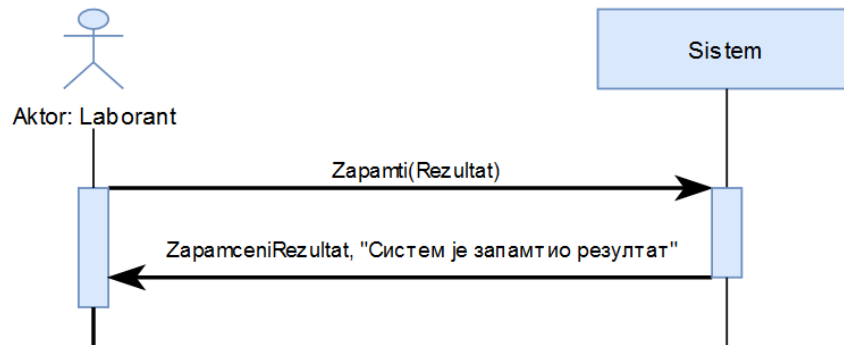


Из наведених секвенчних дијаграма уочавају се две системске операције које је потребно пројектовати:

1. Signal UcitajUpute(List<Uput>);
2. Signal Otvori(Uput);

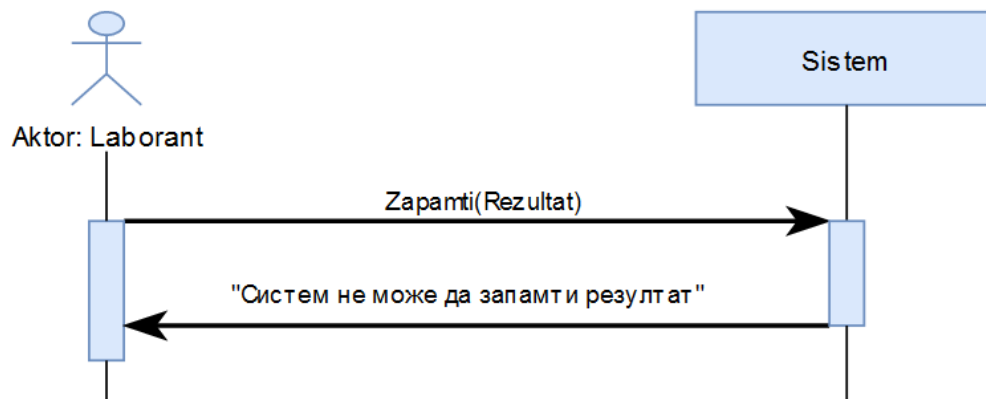
ДС6: Дијаграм секвенци случаја коришћења – Унос резултата лабораторијских анализа – сложен СК

1. **Лаборант** **позива** **систем** да запамти **резултат**. (АПСО)
2. **Систем** **приказује** **лаборанту** запамћени **резултат** и поруку: “**Систем** је запамтио **резултат**”. (ИА)



Алтернативна сценарија

- 2.1 Уколико **систем** не може да запамти податке о **резултату** он приказује **лаборанту** поруку “**Систем** не може да запамти **резултат**”. (ИА)

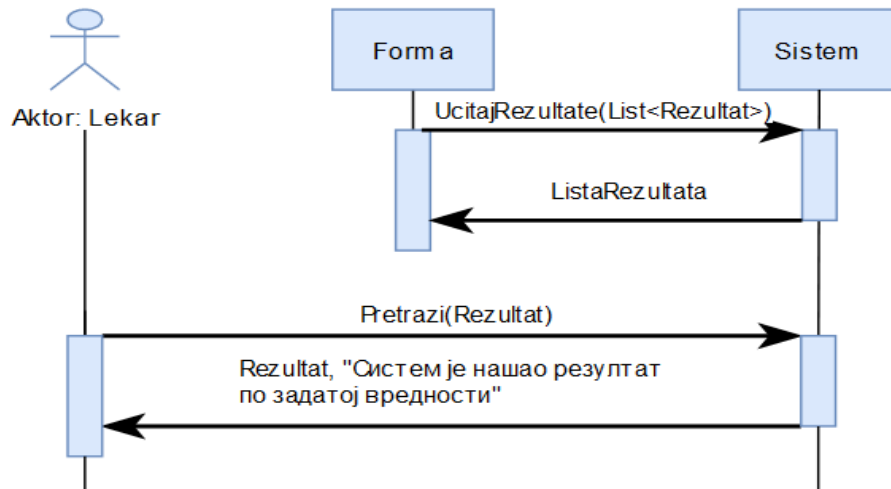


Из наведених секвенцих дијаграма уочавају се две системске операције које је потребно пројектовати:

1. Signal Zapamti(Rezultat);

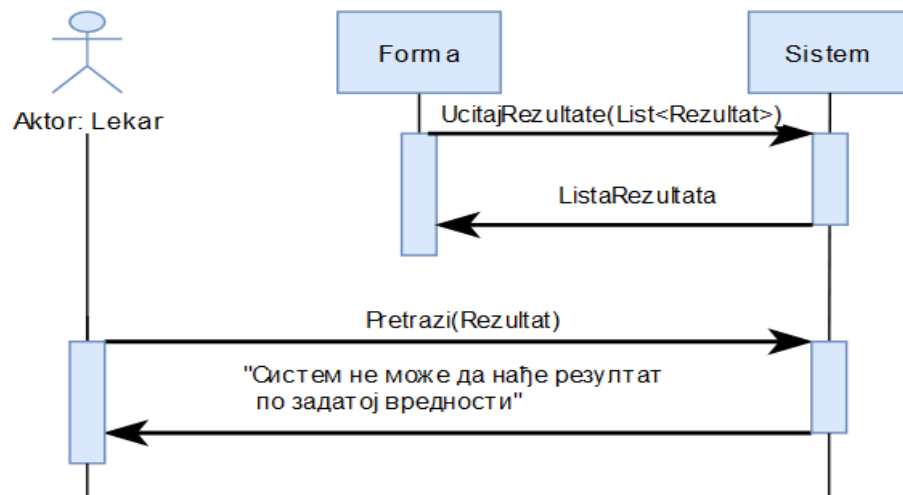
ДС7: Дијаграм секвенци случаја коришћења – Преузимање резултата лабораторијских анализа

1. **Форма** позива **систем** да учита листу резултата. (АПСО)
2. **Систем** враћа **форми** листу резултата. (ИА)
3. **Лекар** позива **систем** да нађе резултат по задатој вредности. (АПСО)
4. **Систем** приказује **лекару** податке о резултату и поруку: “**Систем** је нашао резултат по задатој вредности”. (ИА)



Алтернативна сценарија

- 4.1 Уколико **систем** не може да нађе резултат он приказује **лекару** поруку: “**Систем** не може да нађе резултат по задатој вредности”. (ИА)



Из наведених секвенчних дијаграма уочавају се две системске операције које је потребно пројектовати:

1. Signal UcitajRezultate(List<Rezultat>);
2. Signal Pretrazi(Rezultat);

Као резултат анализе сценарија свих случајева коришћења добијено је 11 системских операција које је потребно пројектовати:

1. Signal Kreiraj(Karton);
2. Signal Zapamti(Karton);
3. Signal Pretrazi(Karton);
4. Signal UcitajAnalyze(List<Analiza>);
5. Signal Zapamti(Uput);
6. Signal UcitajUpute(List<Uput>);
7. Signal Otvori(Uput);
8. Signal Zapamti(Rezultat);
9. Signal Obradi(Rezultat);
10. Signal UcitajRezultate(List<Rezultat>);
11. Signal Pretrazi(Rezultat);

Понашање софтверског система – Дефинисање уговора о системским апликацијама

1. Уговор UG7: UcitajRezultat

Операција: FindRezultatAnalyze(Rezultat):signal;

Веза са СК: CK7

Предуслови:

Постуслови: Учитан је резултат анализе

2. Уговор UG6: Zapamti

Операција: InsertRezultat(Rezultat): Signal

Веза са СК: CK6

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

Постуслови: Резултат је запамћен.

3. Уговор UG5: UcitajAnalizaUputa

Операција: UcitajAnalyzeUputa(Uput u):signal;

Веза са СК: CK5

Предуслови:

Постуслови: Учитана је листа анализа упута.

4. Уговор UG4 UcitajUpute

Операција: FindUputi(List<Uput>):signal;

Веза са СК: CK5

Предуслови:

Постуслови: Учитана је листа упута.

5. Уговор UG3: Zapamti uput

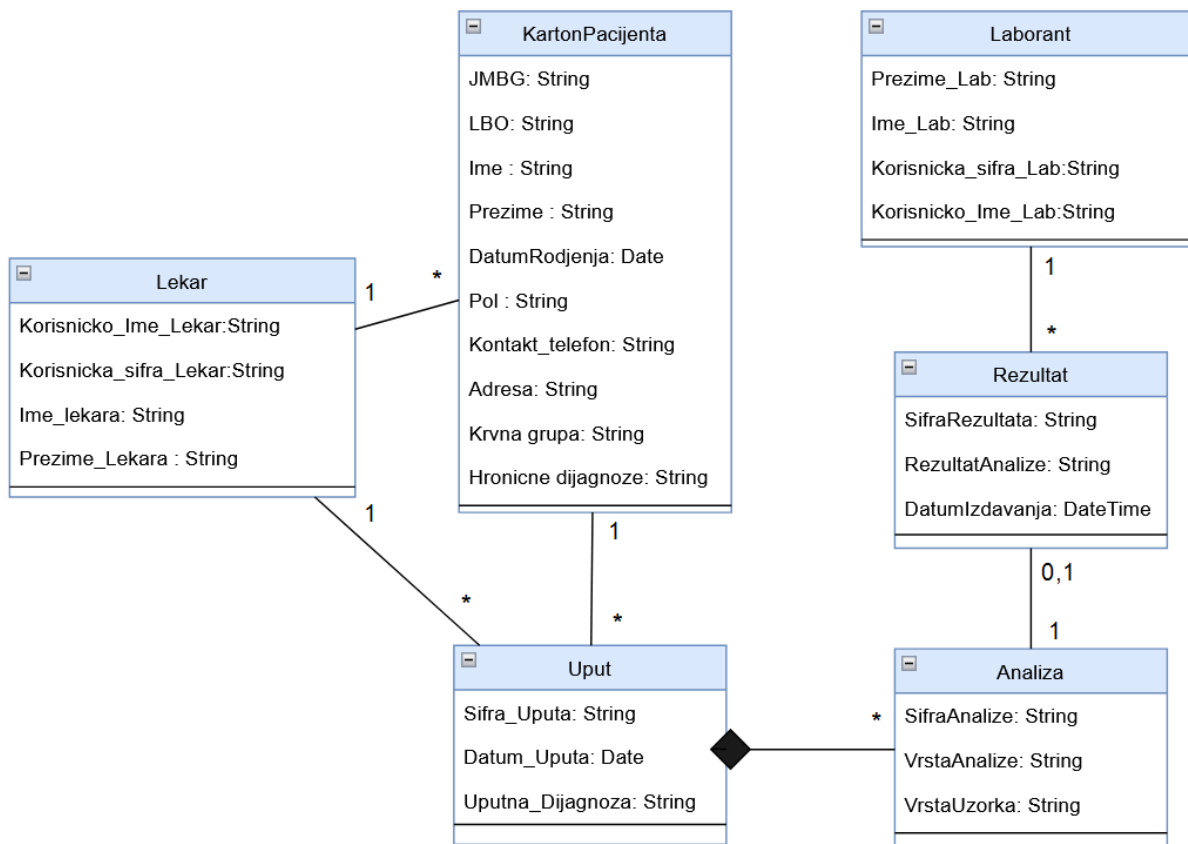
Операција: InsertUput(Uput):signal;

Веза са СК: CK4

Предуслови: Структурна и вредносна над објектом Uput морају бити задовољена.

Постуслови: Запамћен је упут за лабораторијске анализе

Структура софтверског система – Концептуални (доменски) модел



Структура софтверског система – Релациони модел

Lekar(Korisnicko_Ime_Lekar, Korisnicka_Sifra_Lekar, Ime_Lekara, Prezime_Lekara)
 KartonPacijenta(JMBG, LBO, Ime, Prezime, DatumRodjenja, Pol, Kontakt_telefon, Adresa, KrvnaGrupa, HronicnaDijagnoza, *Korisnicko_Ime_Lekar*)
 Laborant(Korisnicko_Ime_Lab, Korisnicka_Sifra_Lab, Ime_Lab, Prezime_Lab)
 Uput(SifraUputa, DatumUputa, UputnaDijagnoza, *Korisnicko_ime_Lekar*, JMBG)
 Rezultat(SifraRezultata, RezultatAnalize, DatumIzdavanja, *SifraLaboranta*, *SifraAnalize*)
 Analiza(SifraAnalize, SifraUputa, VrstaAnalize, VrstaUzorka)

Структурна и вредносна ограничења

Табела Лекар		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Име	Тип атрибута	Вредност атрибута	Међузависност атрибута једне табеле	Међузависност атрибута више табела	INSERT / UPDATE CASCADES KartonPacijenta, Uput DELETE RESTRICTED KartonPacijenta, Uput
	KorisnickoIme	String	Not null			
	KorisnickaSifra	String	Not null			
	ImeLekara	String				
	PrezimeLekara	String				

Табела КартонПацијента		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибут и	Име	Тип атрибута	Вредност атрибута	Међузависност атрибута једне табеле	Међузависност атрибута више табела	INSERT RESTRICTED Lekar UPDATE RESTRICTED Lekar, CASCADES Uput DELETE RESTRICTED Uput
	JMBG	String	Not null			
	LBO	String				
	ImePrezime	String				
	DatumRodjenja	Date				
	Pol	String				
	Kontakt	String				
	Adresa	String				
	KrvnaGrupa	String				
	HronicnaDijagnoza	String				
	SifraLekara	String				

Табела Лаборант		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Име	Тип атрибута	Вредност атрибута	Међузависност атрибута једне табеле	Међузависност атрибута више табела	INSERT / UPDATE CASCADES Rezultat DELETE RESTRICTED Rezultat
	KorisnickoIme	String	Not null			
	KorisnickaSifra	String				
	Ime	String				
	Prezime	String				

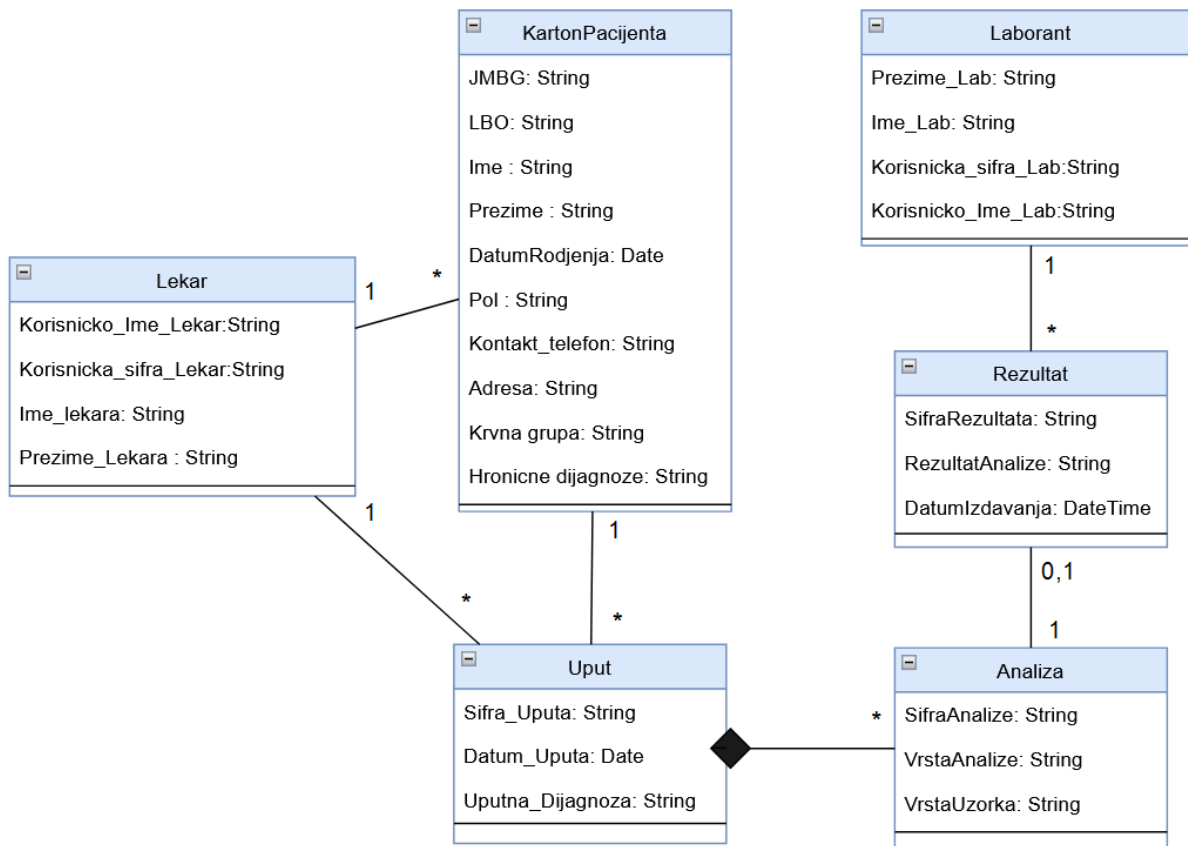
Табела Упут		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Име	Тип атрибута	Вредност атрибута	Међузависност атрибута једне табеле	Међузависност атрибута више табела	INSERT RESTRICTED Lekar, KartonPacijenta UPDATE RESTRICTED Lekar, KartonPacijenta, CASCADES Analiza DELETE RESTRICTED Analiza
	SifraUputa	String	Not null			
	DatumUputa	Date	Not null			
	UputnaDijagnoza	String	Not null			
	Komentar	String	Not null			
	SifraLekara	String	Not null			
	JMBG	String	Not null			

Табела Резултат		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Име	Тип атрибута	Вредност атрибута	Међузависност атрибута једне табеле	Међузависност атрибута више табела	INSERT RESTRICTED Laborant, Analiza UPDATE RESTRICTED Laborant, Analiza DELETE /
	SifraRezultata	String	Not null			
	RezultatAnalize	String	Not null			
	DatumIzdavanja	DateTime	Not null			
	Korisnicko_Ime_Lab	String	Not null			
	SifraAnalize	String	Not null			

Табела Анализа		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
----------------	--	-----------------------------	--	------------------------------	--	-----------------------

Атрибути	Име	Тип атрибута	Вредност атрибута	Међузависност атрибута једне табеле	Међузависност атрибута више табела	INSERT RESTRICTED Uput
	SifraAnalyze	String	Not null			UPDATE RESTRICTED Uput, CASCADES Rezultat
	VrstaAnalyze	String	Not null			
	VrstaUzorka	String	Not null			
	SifraUputa	String	Not null			DELETE RESTRICTED Rezultat

Софтверски систем за медицинске установе



Ponašanja sistema

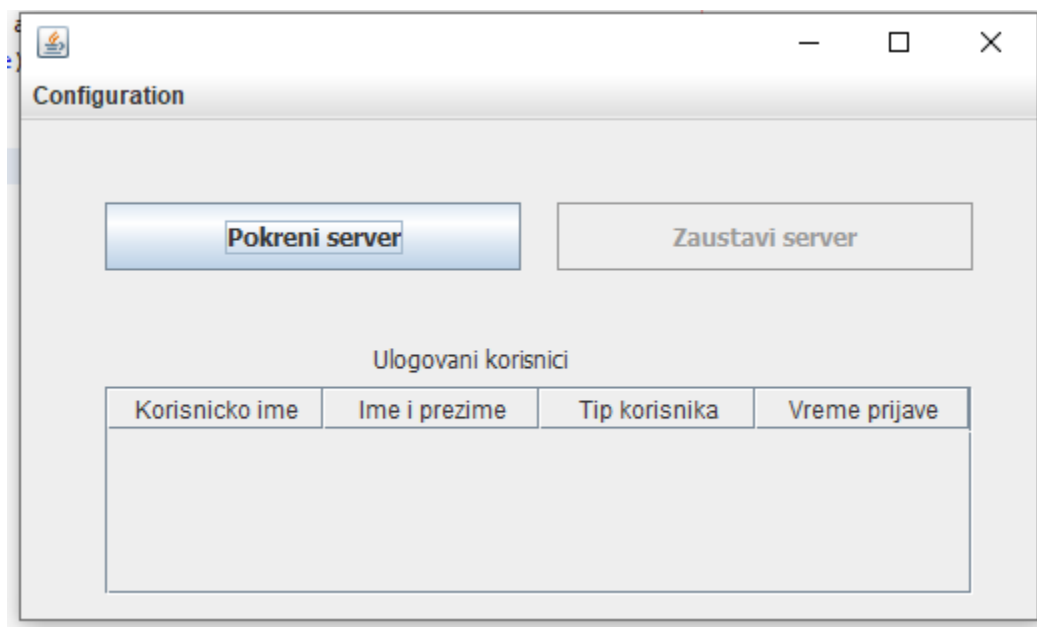
1. Kreiraj(Karton) : signal;
2. Zapamti(Karton) : signal;
3. Pretrazi(Karton) : signal;
4. UcitajAnalize(List<Analiza>) : signal;
5. Zapamti(Uput) : signal;
6. UcitajUpute(List<Uput>) : signal;
7. Otvori(Uput) : signal;
8. Zapamti(Rezultat) : signal;
9. UcitajRezultate(List<Rezultat>) : signal;
10. Pretrazi(Rezultat) : signal;

Фаза пројектовања

Пројектовање корисничког интерфејса

Кориснички интерфејс дефинише се преко екранских форми које види крајњи корисник.

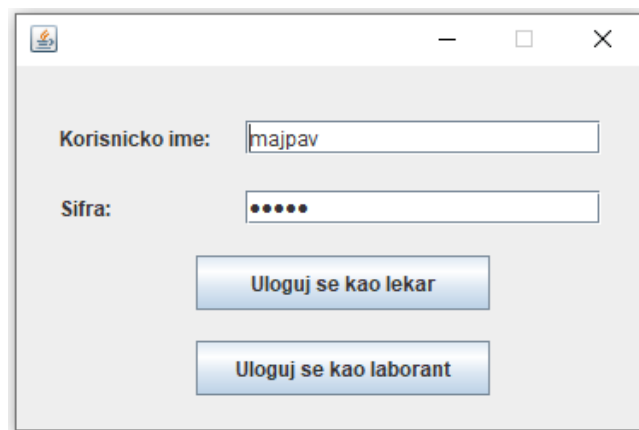
Екранска форма на серверској страни система изгледа овако:



У табели се приказују тренутно улоговани корисници.

На клијентској страни потребно је прво улоговати се како би корисник приступио систему.

Екранска форма за пријављивање изгледа овако:



Постоје два различита начина за пријаву : као корисник и као лаборант, у зависности од наше улоге у систему.

СК1: Случај коришћења – Креирање картона пацијента

Назив СК

Креирање картона пацијента

Актори СК

Лекар

Учесници СК

Лекар и систем (програм)

Предуслов: Систем је укључен и **лекар** је улогован под својом шифром. Систем приказује форму за рад са **картоном пацијента**.

The screenshot shows the 'MedicalIS' application window. At the top right, the user 'Maja Pavlovic (majpav)' is logged in. Below the title bar, there is a search section with the text 'Pretraži karton pacijenta', a text input field for 'JMBG', and two buttons: 'Nadji' and 'Očisti'. The main area is titled 'Podaci o pacijentu' and contains several form fields: 'JMBG' and 'LBO' (text inputs), 'Ime' and 'Prezime' (text inputs), 'Pol' (a dropdown menu with 'Muski' selected), 'Datum rođenja' (a date picker), 'Adresa' (a text input), and 'Kontakt telefon' (a text input). To the right of these fields are 'Krvna grupa' (a dropdown menu with 'A+' selected) and 'Hronicne dijagnoze' (a text area). Below the 'Hronicne dijagnoze' field is a label 'Izabrani lekar' and a text input field. On the far right, there are two tables. The first table is titled 'Lista uputa' and has columns: 'Uput', 'Uputna dijagnoza', 'Datum', and 'Lekar'. The second table is titled 'Analize i rezultati' and has columns: 'Sifra', 'Vrsta uzorka', 'Analiza', and 'Rezultat'. At the bottom of the form, there are three buttons: 'Kreiraj karton novog pacijenta', 'Zameni', and 'Kreiraj novi uput'.

Основни сценарио СК

1. **Лекар** уноси податке у **картон пацијента**. (АПУСО)

The screenshot shows the 'MedicalIS' application window. At the top right, the user 'Maja Pavlovic (majpav)' is logged in. Below the title bar, there is a search bar with the text 'Pretraži karton pacijenta' and a text input field containing 'JMBG 2703999710888'. To the right of this input are two buttons: 'Nadji' and 'Očisti'. Below this search bar is a section titled 'Podaci o pacijentu'. This section contains several form fields: 'JMBG' (2703999710888), 'LBO' (15232678451), 'Ime' (Bojan), 'Prezime' (Bojic), 'Pol' (Muski), 'Datum rođenja' (Mar 27, 1999), 'Adresa' (Resavska 28), and 'Kontakt telefon' (0631111222). There are also dropdown menus for 'Krvna grupa' (A+) and 'Hronične dijagnoze' (Nema). To the right of these fields is a table titled 'Lista uputa' with columns 'Uput', 'Uputna dijagnoza', 'Datum', and 'Lekar'. Below this table is another table titled 'Analize i rezultati' with columns 'Sifra', 'Vrsta uzorka', 'Analiza', and 'Rezultat'. At the bottom of the form are three buttons: 'Kreiraj karton novog pacijenta', 'Zameni', and 'Kreiraj novi uput'.

MedicalIS

Maja Pavlovic (majpav)

Pretraži karton pacijenta JMBG 2703999710888 Nadji Očisti

Podaci o pacijentu

JMBG LBO
2703999710888 15232678451

Ime Prezime
Bojan Bojic

Pol
Muski

Datum rođenja
Mar 27, 1999

Adresa
Resavska 28

Kontakt telefon
0631111222

Krvna grupa
A+

Hronične dijagnoze
Nema

Izabrani lekar
Maja Pavlovic

Lista uputa

Uput	Uputna dijagnoza	Datum	Lekar
------	------------------	-------	-------

Analize i rezultati

Sifra	Vrsta uzorka	Analiza	Rezultat
-------	--------------	---------	----------

Kreiraj karton novog pacijenta Zameni Kreiraj novi uput

2. **Лекар** контролише да ли је коректно унео податке у **картон пацијента**. (АНСО)
3. **Лекар** позива **систем** да запамти податке о **картону пацијента**. (АПСО)

MedicalIS

Maja Pavlovic (majpav)

Pretraži karton pacijenta JMBG

Podaci o pacijentu

JMBG LBO

Ime Prezime

Pol

Datum rođenja

Adresa

Kontakt telefon

Krvna grupa

Hronicne dijagnoze

Izabrani lekar

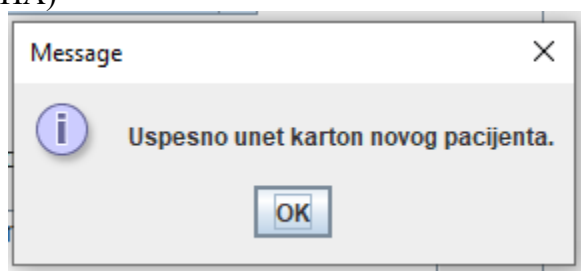
Lista uputa

Uput	Uputna dijagnoza	Datum	Lekar

Analize i rezultati

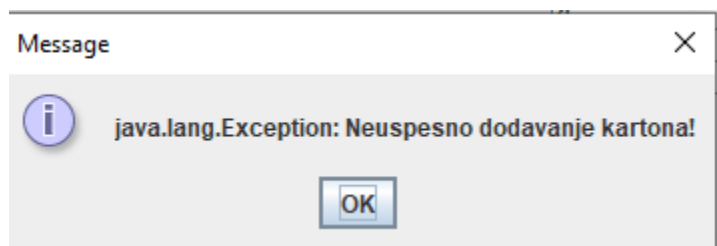
Sifra	Vrsta uzorka	Analiza	Rezultat

4. **Систем памти** податке о **картону пацијента**. (СО)
5. **Систем приказује лекару** запамћени **картон пацијента** и поруку: “**Систем** је запамтио **картон пацијента**“. (ИА)



Алтернативна сценарија

- 3.1 Уколико **систем** не може да креира **картон пацијента** он приказује **лекару** поруку да не може да запамти **картон пацијента**. (ИА)



СК2: Случај коришћења – Претраживање картона пацијента

Назив СК

Претраживање картона пацијента

Актори СК

Лекар

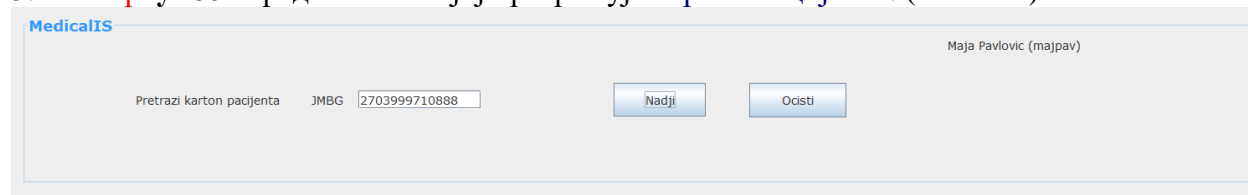
Учесници СК

Лекар и систем (програм)

Предуслов: Систем је укључен и **лекар** је улогован под својом шифром. Систем приказује форму за рад са **картоном пацијента**.

Основни сценарио СК

5. **Лекар** уноси вредност по којој претражује **картон пацијента**. (АПУСО)



6. **Лекар** позива **систем** да нађе **картон пацијента** по задатој вредности. (АПСО)

7. **Систем** тражи **картон пацијента** по задатој вредности. (СО)

8. **Систем** приказује **лекару** податке о **картону пацијента** и поруку: “**Систем** је нашао **картон пацијента** по задатој вредности”. (ИА)

MedicalIS Maja Pavlovic (majpav)

Pretraži karton pacijenta JMBG Nadji Očisti

Podaci o pacijentu

JMBG LBO

Ime Prezime

Pol Muski

Datum rođenja

Adresa

Kontakt telefon

Krvna grupa A+

Hronicne dijagnoze

Izabrani lekar

Lista uputa

Uput	Uputna dijagnoza	Datum	Lekar
8	N50	2022-02-20	Maja Pavlovic

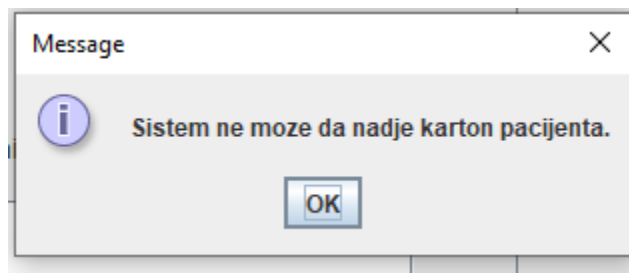
Analize i rezultati

Sifra	Vrsta uzorka	Analiza	Rezultat
11	Bris grla	Bakterioloska	Rezultat nije spr...
12	Bris nosa	Bakterioloska	Rezultat nije spr...

Kreiraj karton novog pacijenta
Zameni
Kreiraj novi uput

Алтернативна сценарија

4.1 Уколико **систем** не може да нађе **картон пацијента** он приказује **лекару** поруку:
 “Систем не може да нађе **картон пацијента**.”. (ИА)



СК3: Случај коришћења – Измена података у картону пацијента

Назив СК

Измена података у картону пацијента

Актери СК

Лекар

Учесници СК

Лекар и систем (програм)

Предуслов: Систем је укључен и лекар је улогован под својом шифром. Систем приказује форму за рад са картоном пацијента.

Основни сценарио СК

1. Лекар уноси вредност по којој претражује картон пацијента. (АПУСО)
2. Лекар позива систем да нађе картон пацијента по задатој вредности. (АПСО)
3. Систем тражи картон пацијента по задатој вредности. (СО)
4. Систем приказује лекару картон пацијента. (ИА)

MedicalIS Maja Pavlovic (majpav)

Pretraži karton pacijenta JMBG 2703999710888 Nadji Ocisti

Podaci o pacijentu

JMBG	LBO	Krvna grupa
2703999710888	15232678451	A+
Ime	Prezime	
Bojan	Bojic	
Pol		Hronične dijagnoze
Muski		Nema
Datum rođenja		
Mar 27, 1999		
Adresa		Izabrani lekar
Resavska 28		Maja Pavlovic
Kontakt telefon		
0631111222		

Kreiraj karton novog pacijenta Zameni Kreiraj novi uput

Lista uputa

Uput	Uputna dijagnoza	Datum	Lekar
8	N50	2022-02-20	Maja Pavlovic

Analize i rezultati

Sifra	Vrsta uzorka	Analiza	Rezultat
11	Bris grla	Bakterioloska	Rezultat nije spr...
12	Bris nosa	Bakterioloska	Rezultat nije spr...

5. Лекар уноси (мења) податке у картону пацијента. (АПУСО)
6. Лекар контролише да ли је коректно унео податке у картон пацијента. (АНСО)

7. **Лекар** **позива** **систем** да запамти податке о **картону пацијента**. (АПСО)

The screenshot shows a patient registration form with the following fields and values:

- JMBG: 2703999710888
- LBO: 15232678451
- Krvna grupa: A+
- Ime: Bojan
- Prezime: Bojic
- Pol: Muski
- Datum rođenja: Mar 27, 1999
- Adresa: Kraljice Natalije 8
- Kontakt telefon: 0631111222
- Hronične dijagnoze: Nema
- Izabrani lekar: Maja Pavlovic

Buttons at the bottom: Kreiraj karton novog pacijenta, Zameni

8. **Систем** **памти** податке о **картону пацијента**. (СО)
9. **Систем** **приказује** **лекару** запамћени **картон пацијента** и поруку да је успешно ажуриран **картон пацијента**. (ИА)

The screenshot shows a patient record update interface. A message box is displayed in the center with the text: "Uspesno azuriran karton pacijenta." The background shows the patient record form with the following fields and values:

- Krvna grupa: A+
- Hronične dijagnoze: Nema
- Izabrani lekar: Maja Pavlovic

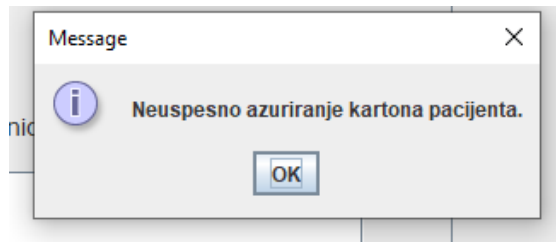
Buttons in the message box: OK

Алтернативна сценарија

- 4.1 Уколико **систем** не може да нађе **картон пацијента** он приказује **лекару** поруку: “**Систем** не може да нађе **картон пацијента**.”. Прекида се извршење сценарија. (ИА)

The screenshot shows a message box with the text: "Sistem ne moze da nadje karton pacijenta." The message box has a close button (X) and an OK button.

9.1 Уколико **систем** не може да запамти податке о **картону пацијента** он приказује **лекару** поруку о неуспешном ажурирању картона пацијента.(ИА)



СК4: Случај коришћења – Креирање упута за лабораторијске анализе – сложен СК

Назив СК

Креирање упута за лабораторијске анализе

Актори СК

Лекар

Учесници СК

Лекар и **систем** (програм)

Предуслов: Систем је укључен и **лекар** је улогован под својом шифром. Систем приказује форму за **унос упута** и листу могућих анализа.

Основни сценарио СК

6. **Лекар уноси** податке о пацијенту и потребним анализама у **упут.** (АПУСО)
7. **Лекар контролише** да ли је коректно унео податке у **упут.** (АНСО)

Kreiranje uputa za laboratorijske analize

JMBG pacijenta: 2703999710888 Datum: 2022-02-20

Sifra uputa:

Uputna dijagnoza: Tonsilitis

Vrsta uzorka: Bris grla Vrsta analize: Virusoloska

Vrsta uzorka	Analiza
Bris grla	Bakterioloska
Bris grla	Virusoloska

8. **Лекар** позива **систем** да запамти податке о **упуту**. (АПСО)
9. **Систем** памти податке о **упуту** и **анализама упута**. (СО)
10. **Систем** приказује **лекару** запамћени **упут** и поруку да је **упут** успешно сачуван. (ИА)

Kreiranje uputa za laboratorijske analize

JMBG pacijenta: 2703999710888 Datum: 2022-02-20

Sifra uputa:

Uputna dijagnoza: Tonsilitis

Vrsta uzorka: Vrsta analize: Virusoloska

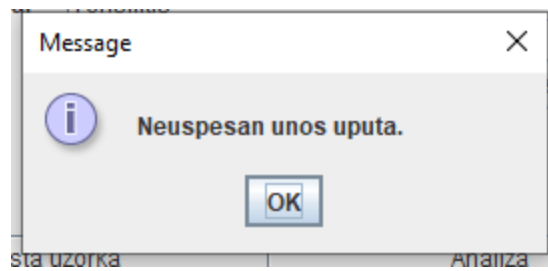
Vrsta uzorka	Analiza
Bris grla	Bakterioloska
Bris grla	Virusoloska

Message

Uspesno unet uput. Sifra uputa: 9

Алтернативна сценарија

- 5.1 Уколико **систем** не може да запамти податке о **упуту** он приказује **лекару** поруку о неуспешном чувању упута. (ИА)



СК5: Случај коришћења – Преузимање упута за лабораторијске анализе

Назив СК

Преузимање упута за лабораторијске анализе

Актори СК

Лаборант

Учесници СК

Лаборант и систем (програм)

Предуслов: Систем је укључен и лаборант је улогован под својом шифром. Систем приказује листу са упутима.

Основни сценарио СК

- Лаборант бира упут који ће да прочита. (АПУСО)
- Лаборант позива систем да отвори задати упут за читање. (АПСО)
- Систем тражи упут по задатој вредности. (СО)
- Систем приказује лаборанту податке о упуту. (ИА)

The screenshot displays the MedicalIS application window. The title bar shows the application name and standard window controls. The main content area is divided into two sections. The top section, titled 'Lista uputa', contains a table with columns: Uput, Uputna dijagnoza, Datum, and Lekar. The bottom section, titled 'Prikaz uputa i unos rezultata', contains a form for entering analysis details. The form includes fields for Sifra analize, Datum uputa, Vrsta uzorka, and Analiza. A 'Rezultat:' label is followed by a large text area for the result. A 'Podaci o rezultatu' box contains fields for 'Datum izdavanja' and 'Laborant'. A 'Lekar' field is also present. A 'Zapamti' button is located at the bottom of the form.

Uput	Uputna dijagnoza	Datum	Lekar
10	Tonsilitis	2022-02-20	Maja Pavlovic
9	Tonsilitis	2022-02-20	Maja Pavlovic
8	NSO	2022-02-20	Maja Pavlovic
7	1	2022-02-19	Maja Pavlovic
6	NSO	2022-02-19	Maja Pavlovic
5	Proba	2022-02-19	Maja Pavlovic
4	Proba	2022-02-19	Maja Pavlovic
3	R50	2022-02-19	Pavle Pajic
2	Upala bronhija	2022-02-19	Pavle Pajic
1	R50	2022-02-18	Maja Pavlovic

Sifra	Vrsta uzorka	Analiza	Rezultat
13	Bris grla	Bakterioloska	Rezultat nije spreman
14	Bris grla	Virusoloska	Rezultat nije spreman

Prikaz uputa i unos rezultata

Sifra analize: 13 Datum uputa: 2022-02-20 Vrsta uzorka: Bris grla Analiza: Bakteriološka

Rezultat:

Podaci o rezultatu

Datum izdavanja:

Laborant:

Lekar: Maja Pavlovic

Zapamti

Алтернативна сценарија

4.1 Уколико **систем** не може да отвори **упут** он приказује **лаборанту** поруку: “**Систем** не може да отвори дати **упут**”. (ИА)

СК6: Случај коришћења – Унос резултата лабораторијских анализа – сложен СК

Назив СК

Унос резултата лабораторијских анализа

Актори СК

Лаборант

Учесници СК

Лаборант и систем (програм)

Предуслов: Систем је укључен и **лаборант** је улогован под својом шифром. Систем приказује форму за унос **резултата**.

Основни сценарио СК

6. **Лаборант** уноси податке о **резултату** у форму **система**. (АПУСО)
7. **Лаборант** контролише да ли је коректно унео податке у **резултат**. (АНСО)

The screenshot shows the MedicalIS application interface. At the top, there's a header with the application name 'MedicalIS' and the user name 'Maja Pavlovic(majpav)'. Below the header, there are two tables. The left table, titled 'Lista uputa', lists medical orders with columns for 'Uput', 'Uputna dijagnoza', 'Datum', and 'Lekar'. The right table, titled 'Maja Pavlovic(majpav)', shows a list of results with columns for 'Sifra', 'Vrsta uzorka', 'Analiza', and 'Rezultat'. Below these tables, there's a section titled 'Prikaz uputa i unos rezultata' which contains a form for entering results. The form has fields for 'Sifra analize', 'Datum uputa', 'Vrsta uzorka', and 'Analiza'. There's also a 'Rezultat:' field with a text area for entering the result. A 'Zapamti' button is at the bottom of the form. On the right side of the form, there's a 'Podaci o rezultatu' section with fields for 'Datum izdavanja', 'Laborant', and 'Lekar'.

Uput	Uputna dijagnoza	Datum	Lekar
10	Tonsilitis	2022-02-20	Maja Pavlovic
9	Tonsilitis	2022-02-20	Maja Pavlovic
8	NSO	2022-02-20	Maja Pavlovic
7	1	2022-02-19	Maja Pavlovic
6	NSO	2022-02-19	Maja Pavlovic
5	Proba	2022-02-19	Maja Pavlovic
4	Proba	2022-02-19	Maja Pavlovic
3	RSO	2022-02-19	Pavle Pajic
2	Upala bronhija	2022-02-19	Pavle Pajic
1	RSO	2022-02-18	Maja Pavlovic

Sifra	Vrsta uzorka	Analiza	Rezultat
13	Bris grla	Bakterioloska	Rezultat nije spreman
14	Bris grla	Virusoloska	Rezultat nije spreman

Prikaz uputa i unos rezultata

Sifra analize: 13 Datum uputa: 2022-02-20 Vrsta uzorka: Bris grla Analiza: Bakteriološka

Rezultat: Izolovan je Streptococcus

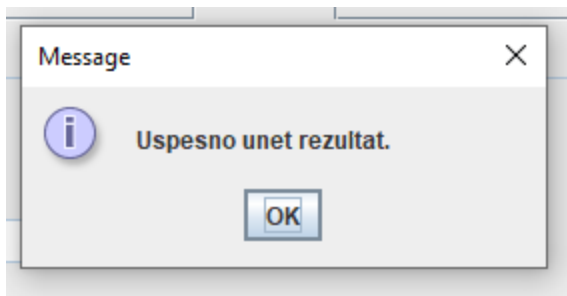
Lekar: Maja Pavlovic

Zapamti

Podaci o rezultatu

Datum izdavanja: Laborant: Lekar:

8. **Лаборант** позива **систем** да запамти **резултат**. (АПСО)
9. **Систем** памти **резултат**. (СО)
10. **Систем** приказује **лаборанту** запамћени **резултат** и поруку о успешно сачуваном резултату. (ИА)



Алтернативна сценарија

5.1 Уколико **систем** не може да запамти податке о **результату** он приказује **лаборанту** поруку “**Систем** не може да запамти **результат**. (ИА)

СК7: Случај коришћења – Преузимање резултата лабораторијских анализа

Назив СК

Преузимање **результата лабораторијских анализа**

Актори СК

Лекар

Учесници СК

Лекар и **систем** (програм)

Предуслов: Систем је укључен и **лекар** је улогован под својом шифром. Систем приказује листу са упутим и **результатима**.

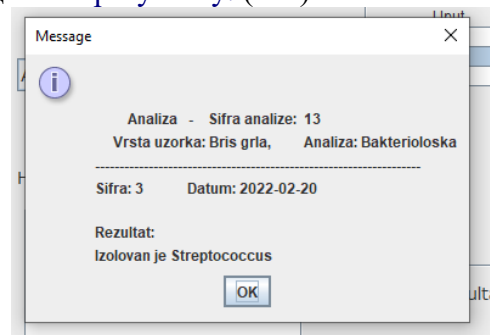
Основни сценарио СК

5. **Лекар бира** анализу за коју жели да прочита **результат**. (АПУСО)
6. **Лекар позива систем** да нађе **результат**. (АПСО)

Lista uputa			
Uput	Uputna dijagnoza	Datum	Lekar
10	Tonsilitis	2022-02-20	Maja Pavlovic
9	Tonsilitis	2022-02-20	Maja Pavlovic
8	N50	2022-02-20	Maja Pavlovic

Analize i rezultati			
Sifra	Vrsta uzorka	Analiza	Rezultat
13	Bris grla	Bakterioloska	Izolovan je Stre...
14	Bris grla	Virusoloska	Rezultat nije spr...

7. Систем **тражи** **резултат** по задатој вредности. (CO)
8. Систем приказује **лекару** податке о **резултату**. (ИА)



Алтернативна сценарија

- 4.1 Уколико **систем** не може да нађе **резултат** он приказује **лекару** поруку: “ N/A”. (ИА)

Пројектовање структуре софтверског система

Општа доменски интерфејс који имлментирају конкретне класе система:

```

package domain;

] import java.io.Serializable;
- import java.sql.*;

public interface GeneralDObject extends Serializable{

    String getAtrValue();
    String setAtrValue();
    String getClassName();
    String getWhereCondition();
    String getNameByColumn(int column);

    GeneralDObject getNewRecord(ResultSet rs) throws SQLException;

] public default String getFields() { return ""; }

] public default String getOrderBy() { return ""; }

] public default String getWhereCondition1() {return ""; }
}

```

Конкретне доменске класе

```
public class KartonPacijenta implements Serializable, GeneralDObject {

    String jmbg, lbo;
    String ime;
    String prezime;
    String pol;
    Date datumRodjenja;
    String adresa;
    String kontaktTelefon;
    String krvnaGrupa;
    String hronicneDijagnoze;
    Lekar lekar;

    List<Uput> uputi;

    public KartonPacijenta() {
    }

    public KartonPacijenta(String jmbg) {
        this.jmbg = jmbg;
    }

    public KartonPacijenta(String jmbg, String lbo, String ime, String prezime, String pol, Date datumRodjenja,
        String adresa, String kontaktTelefon, String krvnaGrupa, String hronicneDijagnoze, Lekar lekar) {
        this.jmbg = jmbg;
        this.lbo = lbo;
        this.ime = ime;
        this.prezime = prezime;
        this.pol = pol;
        this.datumRodjenja = datumRodjenja;
        this.adresa = adresa;
        this.kontaktTelefon = kontaktTelefon;
        this.krvnaGrupa = krvnaGrupa;
        this.hronicneDijagnoze = hronicneDijagnoze;
        this.lekar = lekar;
    }

    public String getJmbg() {
        return jmbg;
    }

    public void setJmbg(String jmbg) {
        this.jmbg = jmbg;
    }
}
```

```

public String getLbo() {
    return lbo;
}

public void setLbo(String lbo) {
    this.lbo = lbo;
}

public String getTime() {
    return ime;
}

public void setTime(String ime) {
    this.ime = ime;
}

public String getPrezime() {
    return prezime;
}

public void setPrezime(String prezime) {
    this.prezime = prezime;
}

public String getPol() {
    return pol;
}

public void setPol(String pol) {
    this.pol = pol;
}

public Date getDatumRodjenja() {
    return datumRodjenja;
}

public void setDatumRodjenja(Date datumRodjenja) {
    this.datumRodjenja = datumRodjenja;
}

public String getAdresa() {
    return adresa;
}

public void setAdresa(String adresa) {
    this.adresa = adresa;
}

```

```

public String getKontaktTelefon() {
    return kontaktTelefon;
}

public void setKontaktTelefon(String kontaktTelefon) {
    this.kontaktTelefon = kontaktTelefon;
}

public String getKrvnaGrupa() {
    return krvnaGrupa;
}

public void setKrvnaGrupa(String krvnaGrupa) {
    this.krvnaGrupa = krvnaGrupa;
}

public String getHronicneDijagnoze() {
    return hronicneDijagnoze;
}

public void setHronicneDijagnoze(String hronicneDijagnoze) {
    this.hronicneDijagnoze = hronicneDijagnoze;
}

public List<Uput> getUputi() {
    return uputi;
}

public void setUputi(List<Uput> uputi) {
    this.uputi = uputi;
}

public Lekar getLekar() {
    return lekar;
}

public void setLekar(Lekar lekar) {
    this.lekar = lekar;
}

@Override
public String getAtrValue() {
    SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");
    return "'" + jmbg.toString() + "', '" + lbo.toString() + "', '" + ime + "', '"
        + prezime + "', '" + formatter.format(datumRodjenja) + "', '" + pol.toString()
        + "', '" + adresa + "', '" + kontaktTelefon + "', '" + krvnaGrupa.toString() +
        "', '" + hronicneDijagnoze + "', '" + lekar.getUsername() + "'";
}

@Override
public String setAtrValue() {
    SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");
    return "jmbg= '" + jmbg + "', lbo='" + lbo + "', ime='" + ime
        + "', prezime='" + prezime + "', pol='" + pol.toString()
        + "', datumrodjenja='" + formatter.format(datumRodjenja)
        + "', adresa='" + adresa + "', kontakt_telefon='"
        + kontaktTelefon + "', krvna_grupa='" + krvnaGrupa.toString()
        + "', hronicne_dijagnoze='" + hronicneDijagnoze
        + "', lekar='" + lekar.getUsername() + "'";
}

@Override
public String getClassName() {
    return "KartonPacijenta";
}

@Override
public String getWhereCondition() {
    return "jmbg = '" + jmbg + "'";
}

```

```

@Override
public GeneralDObject getNewRecord(ResultSet rs) throws SQLException {
    return new KartonPacijenta(
        rs.getString("jmbg"), rs.getString("lbo"), rs.getString("ime"),
        rs.getString("prezime"), rs.getString("pol"),
        rs.getDate("datumrodjenja"), rs.getString("adresa"),
        rs.getString("kontakt_telefon"), rs.getString("krvna_grupa"),
        rs.getString("hronicne_dijagnoze"), new Lekar(rs.getString("lekar"))
    );
}

@Override
public String toString() {
    return "jmbg= '" + jmbg + "', lbo='" + lbo + "', ime='" + ime
        + "', prezime='" + prezime + "', pol='" + ((pol == null) ? "" : pol.toString())
        + "', datumrodjenja='" + ((datumRodjenja == null) ? "" : datumRodjenja.toString())
        + "', adresa='" + adresa + "', kontakt_telefon='"
        + kontaktTelefon + "', krvna_grupa='" + krvnaGrupa.toString()
        + "', hronicne_dijagnoze'" + hronicneDijagnoze + "', lekar=" + lekar.getIme() + " " + lekar.getPrezime();
}

@Override
public String getWhereCondition1() {
    return "jmbg = '" + jmbg + "' AND lekar='" + lekar.getUsername() + "'";
}

```

```

public class Lekar implements Serializable, GeneralDObject {

    String username;
    String password;
    String ime;
    String prezime;

    public Lekar() {
    }

    public Lekar(String username, String password, String ime, String prezime) {
        this.username = username;
        this.password = password;
        this.ime = ime;
        this.prezime = prezime;
    }

    public Lekar(String username) {
        this.username = username;
    }

    public Lekar(String username, String password) {
        this.username = username;
        this.password = password;
    }
}

```

```

public String getTime() {
    return ime;
}

public void setTime(String ime) {
    this.ime = ime;
}

public String getPrezime() {
    return prezime;
}

public void setPrezime(String prezime) {
    this.prezime = prezime;
}

public String getUsername() {
    return username;
}

public void setUsername(String username) {
    this.username = username;
}

public String getPassword() {
    return password;
}

public void setPassword(String password) {
    this.password = password;
}

@Override
public String getAtrValue() {
    return "'" + username + "', '" + password + "', '" + ime + "', '" + prezime + "'";
}

@Override
public String setAtrValue() {
    return "korisnicko_ime='" + username + "', korisnicka_sifra='" + password +
        "', ime='" + ime + "', prezime='" + prezime + "'";
}

@Override
public String getClassName() {
    return "Lekar";
}

@Override
public String getWhereCondition() {
    return "korisnicko_ime='" + username + "' AND korisnicka_sifra='" + password + "'";
}

@Override
public GeneralDBObject getNewRecord(ResultSet rs) throws SQLException {
    return new Lekar(rs.getString("korisnicko_ime"), rs.getString("korisnicka_sifra"),
        rs.getString("ime"), rs.getString("prezime"));
}

@Override
public String getWhereCondition1() {
    return "korisnicko_ime='" + username + "'";
}

```



```

public class Laborant implements Serializable, GeneralDBObject {

    String username;
    String password;
    String ime;
    String prezime;

    public Laborant() {
    }

    public Laborant(String username, String password, String ime, String prezime) {
        this.username = username;
        this.password = password;
        this.ime = ime;
        this.prezime = prezime;
    }

    public Laborant(String username) {
        this.username = username;
    }

    public String getIme() {
        return ime;
    }

    public void setIme(String ime) {
        this.ime = ime;
    }

    public String getPrezime() {
        return prezime;
    }

    public void setPrezime(String prezime) {
        this.prezime = prezime;
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    @Override
    public String getAtrValue() {
        return "\"" + username + "\", \"\" + password + "\", \"\" + ime + "\", \"\" + prezime + "\"";
    }

}

```

```

@Override
public String setAtrValue() {
    return "korisnicko_ime='" + username + "', korisnicka_sifra='" +
        password + "', ime='" + ime + "', prezime='" + prezime + "'";
}

@Override
public String getClassName() {
    return "Laborant";
}

@Override
public String getWhereCondition() {
    return "korisnicko_ime='" + username + "' AND korisnicka_sifra='" + password + "'";
}

@Override
public GeneralDBObject getNewRecord(ResultSet rs) throws SQLException {
    return new Laborant(rs.getString("korisnicko_ime"), rs.getString("korisnicka_sifra"),
        rs.getString("ime"), rs.getString("prezime"));
}

@Override
public String getWhereCondition1() {
    return "korisnicko_ime='" + username + "'";
}

```

```

public class Uput implements Serializable, GeneralDBObject {

    Long sifraUputa;
    Date datumUputa;
    String uputnaDijagnoza;
    Lekar lekar;
    KartonPacijenta pacijent;

    List<Analiza> analize;

    public Uput() {
    }

    public Uput(Long sifraUputa, Date datumUputa, String uputnaDijagnoza, Lekar lekar,
        KartonPacijenta pacijent, List<Analiza> analize) {
        this.sifraUputa = sifraUputa;
        this.datumUputa = datumUputa;
        this.uputnaDijagnoza = uputnaDijagnoza;
        this.lekar = lekar;
        this.pacijent = pacijent;
        this.analize=analize;
    }

    public Uput(Long sifraUputa, Date datumUputa, String uputnaDijagnoza,
        Lekar lekar, KartonPacijenta pacijent) {
        this.sifraUputa = sifraUputa;
        this.datumUputa = datumUputa;
        this.uputnaDijagnoza = uputnaDijagnoza;
        this.lekar = lekar;
        this.pacijent = pacijent;
    }
}

```

```

public Uput(Long sifraUputa, Date datumUputa, String uputnaDijagnoza, String vrstaUzorka) {
    this.sifraUputa = sifraUputa;
    this.datumUputa = datumUputa;
    this.uputnaDijagnoza = uputnaDijagnoza;
}

public Uput(Long sifraUputa) {
    this.sifraUputa = sifraUputa;
}

public Long getSifraUputa() {
    return sifraUputa;
}

public void setSifraUputa(Long sifraUputa) {
    this.sifraUputa = sifraUputa;
}

public Date getDatumUputa() {
    return datumUputa;
}

public void setDatumUputa(Date datumUputa) {
    this.datumUputa = datumUputa;
}

public String getUputnaDijagnoza() {
    return uputnaDijagnoza;
}

public void setUputnaDijagnoza(String uputnaDijagnoza) {
    this.uputnaDijagnoza = uputnaDijagnoza;
}

public Lekar getLekar() {
    return lekar;
}

public void setLekar(Lekar lekar) {
    this.lekar = lekar;
}

public KartonPacijenta getPacijent() {
    return pacijent;
}

public void setPacijent(KartonPacijenta pacijent) {
    this.pacijent = pacijent;
}

```

```

public List<Analiza> getAnalyze() {
    return analyze;
}

public void setAnalyze(List<Analiza> analyze) {
    this.analyze = analyze;
}

@Override
public String getAtrValue() {
    SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");
    return sifraUputa + ", " + formatter.format(datumUputa) + ", " +
        uputnaDijagnoza + ", " + lekar.getUsername() + ", " +
        pacijent.getJmbg() + "";
}

@Override
public String setAtrValue() {
    return "sifra_uputa=" + sifraUputa + ", datum_uputa=" + datumUputa + ", uputna_dijagnoza=" +
        uputnaDijagnoza + ", sifra_lekara=" + lekar.getUsername() + ", jmbg=" +
        pacijent.getJmbg() + "";
}

@Override
public String getClassName() {
    return "Uput";
}

@Override
public String getWhereCondition() {
    return "jmbg = " + pacijent.getJmbg().toString() + " ORDER BY sifra_uputa DESC";
}

@Override
public GeneralDBObject getNewRecord(ResultSet rs) throws SQLException {
    return new Uput(rs.getLong("sifra_uputa"), rs.getDate("datum_uputa"),
        rs.getString("uputna_dijagnoza"), new Lekar(rs.getString("sifra_lekara")),
        new KartonPacijenta(rs.getString("jmbg")));
}

@Override
public String toString() {
    SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");
    return "sifra_uputa=" + sifraUputa + ", datum_uputa=" + formatter.format(datumUputa) + ", uputna_dijagnoza=" +
        uputnaDijagnoza + ", sifra_lekara=" + lekar.getUsername() + ", jmbg=" +
        pacijent.getJmbg() + "";
}

@Override
public String getFields() {
    return "sifra_uputa";
}

@Override
public String getOrderBy() {
    return "sifra_uputa DESC";
}

```

```

public class Analiza implements GeneralDObject, Serializable {

    Long sifraAnalize;
    String vrstaAnalize, vrstaUzorka;
    Uput uput;

    public Analiza(Long sifraAnalize) {
        this.sifraAnalize = sifraAnalize;
    }

    public Analiza(Long sifraAnalize, String vrstaAnalize, String vrstaUzorka, Uput uput) {
        this.sifraAnalize = sifraAnalize;
        this.vrstaAnalize = vrstaAnalize;
        this.vrstaUzorka = vrstaUzorka;
        this.uput = uput;
    }

    public Analiza() {
    }

    public Long getSifraAnalize() {
        return sifraAnalize;
    }

    public void setSifraAnalize(Long sifraAnalize) {
        this.sifraAnalize = sifraAnalize;
    }

    public String getVrstaAnalize() {
        return vrstaAnalize;
    }

    public void setVrstaAnalize(String vrstaAnalize) {
        this.vrstaAnalize = vrstaAnalize;
    }

    public String getVrstaUzorka() {
        return vrstaUzorka;
    }

    public void setVrstaUzorka(String vrstaUzorka) {
        this.vrstaUzorka = vrstaUzorka;
    }

    public Uput getUput() {
        return uput;
    }

    public void setUput(Uput uput) {
        this.uput = uput;
    }

    @Override
    public String getAtrValue() {
        return sifraAnalize + ", " + vrstaAnalize + ", " + vrstaUzorka + ", " + uput.getSifraUputa();
    }
}

```

```

@Override
public String setAtrValue() {
    return "sifra_analize = " + sifraAnalyze + ", vrsta_analize=" +
        vrstaAnalyze + ", vrsta_uzorka=" + vrstaUzorka + ", sifra_uputa=" + uput.getSifraUputa();
}

@Override
public String getClassName() {
    return "Analiza";
}

@Override
public String getWhereCondition() {
    return "sifra_uputa = " + uput.getSifraUputa();
}

@Override
public GeneralDBObject getNewRecord(ResultSet rs) throws SQLException {
    return new Analiza(
        rs.getLong("sifra_analize"),
        rs.getString("vrsta_analize"),
        rs.getString("vrsta_uzorka"),
        new Uput(rs.getLong("sifra_uputa")));
}

@Override
public String getFields() {
    return "sifra_analize";
}

public class Rezultat implements Serializable, GeneralDBObject {

    Long sifra_rezultata;
    String rezultat_analize;
    Date datumIzdavanja;
    Laborant laborant;
    Analiza analiza;

    public Rezultat() {
    }

    public Rezultat(Long sifra_rezultata, String rezultat_analize, Date datumIzdavanja, Laborant laborant, Analiza analiza) {
        this.sifra_rezultata = sifra_rezultata;
        this.rezultat_analize = rezultat_analize;
        this.datumIzdavanja = datumIzdavanja;
        this.laborant = laborant;
        this.analiza = analiza;
    }

    public Long getSifra_rezultata() {
        return sifra_rezultata;
    }

    public void setSifra_rezultata(Long sifra_rezultata) {
        this.sifra_rezultata = sifra_rezultata;
    }

    public Analiza getAnaliza() {
        return analiza;
    }
}

```

```

public void setAnaliza(Analiza analiza) {
    this.analiza = analiza;
}

public String getRezultat_analize() {
    return rezultat_analize;
}

public void setRezultat_analize(String rezultat_analize) {
    this.rezultat_analize = rezultat_analize;
}

public Date getDatumIzdavanja() {
    return datumIzdavanja;
}

public void setDatumIzdavanja(Date datumIzdavanja) {
    this.datumIzdavanja = datumIzdavanja;
}

public Laborant getLaborant() {
    return laborant;
}

public void setLaborant(Laborant laborant) {
    this.laborant = laborant;
}

@Override
public String getAtrValue() {
    SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");
    return sifra_rezultata + ", " + rezultat_analize + ", " + formatter.format(datumIzdavanja)
        + ", " + laborant.getUsername() + ", " + analiza.getSifraAnalize();
}

@Override
public String setAtrValue() {
    return "sifra_rezultata=" + sifra_rezultata + ", rezultat_analize=" + rezultat_analize
        + ", datum_izdavanja=" + datumIzdavanja + ", laborant=" + laborant.getUsername()
        + ", sifra_analize=" + analiza.getSifraAnalize();
}

@Override
public String getClassName() {
    return "Rezultat";
}

@Override
public String getWhereCondition() {
    return "sifra_analize = " + analiza.getSifraAnalize();
}

```

```

@Override
public GeneralDObject getNewRecord(ResultSet rs) throws SQLException {
    return new Rezultat(rs.getLong("sifra_rezultata"), rs.getString("rezultat_analize"),
        rs.getDate("datum_izdavanja"), new Laborant(rs.getString("laborant")),
        new Analiza(rs.getLong("sifra_analize")));
}

@Override
public String getOrderBy() {
    return "sifra_rezultata ASC";
}

@Override
public String getFields() {
    return "sifra_rezultata";
}

@Override
public String toString() {
    return "sifra_rezultata=" + sifra_rezultata + ", rezultat_analize='" + rezultat_analize
        + "', datum_izdavanja=" + datumIzdavanja + "";
}

```

Брокер базе података

Брокер базе података посредује у свим операцијама над базом података и имплементиран је кроз следеће методе:

```

public abstract class BrokerBazePodataka {
    public abstract boolean makeConnection();
    public abstract boolean insertRecord(GeneralDObject odo);
    public abstract boolean updateRecord(GeneralDObject odo, GeneralDObject odoold);
    public abstract boolean deleteRecord(GeneralDObject odo);
    public abstract GeneralDObject findRecord(GeneralDObject odo);
    public abstract GeneralDObject findRecord1(GeneralDObject odo);
    public abstract List<GeneralDObject> findAllRecords(GeneralDObject odo);
    public abstract List<GeneralDObject> findAllRecords_NoCondition(GeneralDObject odo);
    public abstract Long findMaxRecord(GeneralDObject odo);
    public abstract boolean commitTransation();
    public abstract boolean rollbackTransation();
    public abstract boolean getCounter(GeneralDObject odo, AtomicInteger counter);
    public abstract boolean increaseCounter(GeneralDObject odo, AtomicInteger counter);
    public abstract void closeConnection();
}

```


Методе брокера базе података пројектоване су као генеричке и прихватају објекте класа које имплементирају интерфејс GeneralDBObject:

```
package domain;

import java.io.Serializable;
import java.sql.*;

public interface GeneralDBObject extends Serializable{

    String getAtrValue();
    String setAtrValue();
    String getClassName();
    String getWhereCondition();

    GeneralDBObject getNewRecord(ResultSet rs) throws SQLException;

    public default String getFields() { return ""; }

    public default String getOrderBy() { return ""; }

    public default String getWhereCondition1() {return ""; }

}
```

Пројектовање складишта података

Табела KartonPacijenta

1 Columns 2 Indexes 3 Foreign Keys 4 Advanced 5 SQL Preview							
<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?
<input type="checkbox"/>	jmbg	varchar	13		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	lbo	varchar	11		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	ime	varchar	20		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	prezime	varchar	20		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	datumrodjenja	date			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	pol	varchar	20		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	adresa	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	kontakt_telefon	varchar	20		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	krvna_grupa	varchar	4		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	hronicne_dijagnoza	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	lekar	varchar	15		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Табела Lekar

1 Columns 2 Indexes 3 Foreign Keys 4 Advanced 5 SQL Preview								
<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	A
<input type="checkbox"/>	korisnicko_ime	varchar	15		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	korisnicka_sifra	varchar	15		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	ime	varchar	20		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	prezime	varchar	20		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Табела Laborant

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	A
<input type="checkbox"/>	korisnicko_ime	varchar	15		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	korisnicka_sifra	varchar	15		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	ime	varchar	20		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	prezime	varchar	20		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Табела Uput

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Au
<input type="checkbox"/>	sifra_uputa	bigint	20		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	datum_uputa	date			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	uputna_dijagnoza	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	sifra_lekara	varchar	15		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	jmbg	varchar	13		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Табела Analiza

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Uns
<input type="checkbox"/>	sifra_analize	bigint	20		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	vrsta_analize	varchar	30		<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	vrsta_uzorka	varchar	20		<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	sifra_uputa	bigint	20		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>	

Табела Rezultat

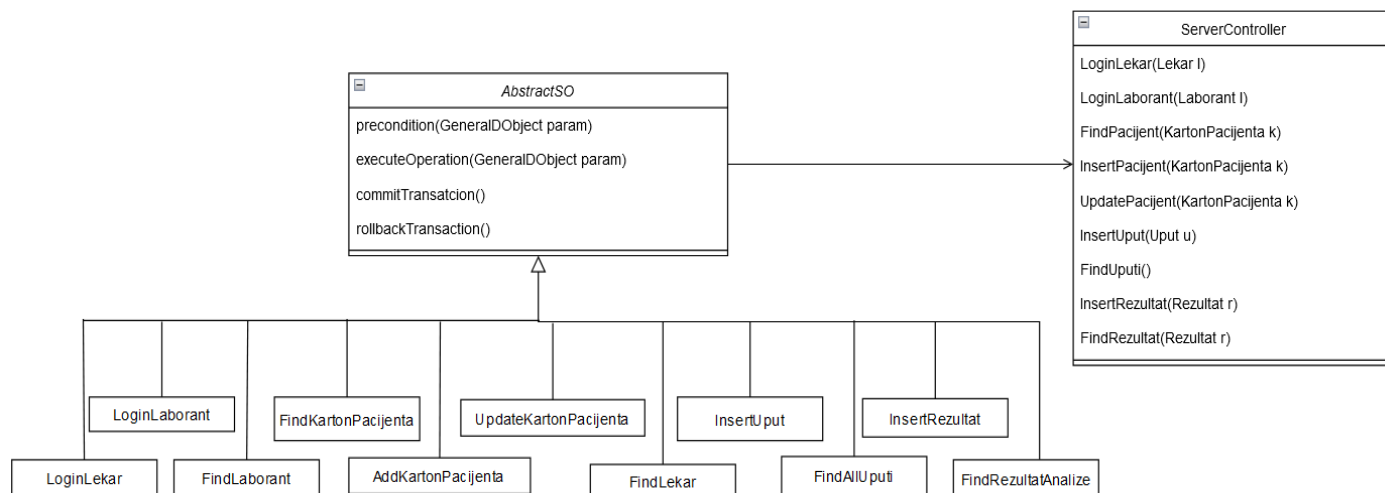
<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	A
<input type="checkbox"/>	sifra_rezultata	bigint	20		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	rezultat_analize	varchar	50		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	datum_izdavanja	date			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	laborant	varchar	15		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	sifra_analize	bigint	20		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Пројектовање контролера корисничког интерфејса

Контролер корисничког интерфејса одговоран је за прихватање података са екранске форме, конвертовање тих података у објекте доменских класа и прослеђивање мрежи, као и приказ приказ објеката доменских класа на екранским формама.

Контролер апликационе логике

Контролер апликационе логике прослеђује захтев за извршење системских операција до одговорних класа.



Пословна логика система

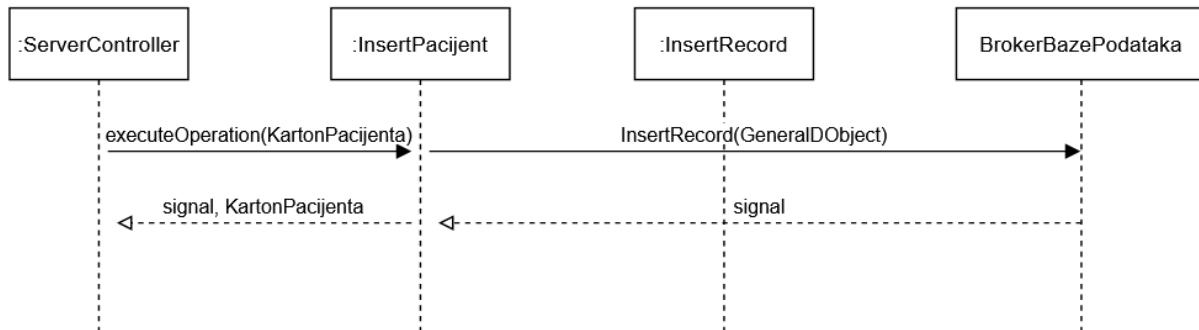
6. Уговор UG1:

Операција: KreirajPacijent(*KartonPacijenta*):signal;

Веза са СК: CK1

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

Постуслови: Креиран је нов картон пацијента.



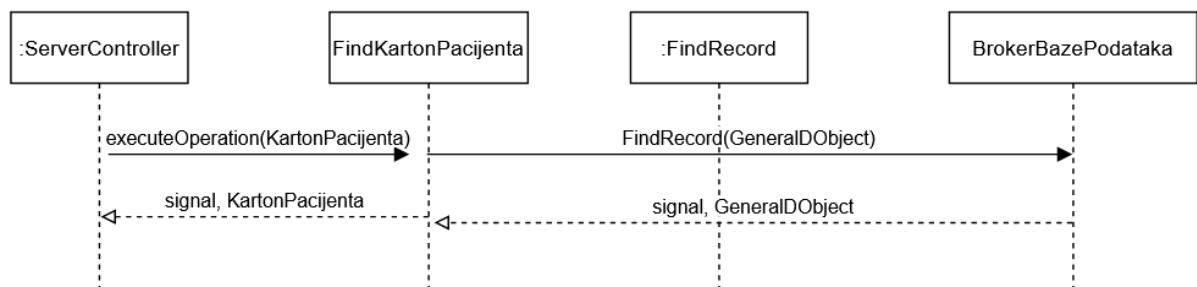
7. Уговор UG2: Pretrazi

Операција: FindPacijent(*KartonPacijenta*):signal;

Веза са СК: CK2, CK3

Предуслови:

Постуслови: Пронађен је тражени картон.



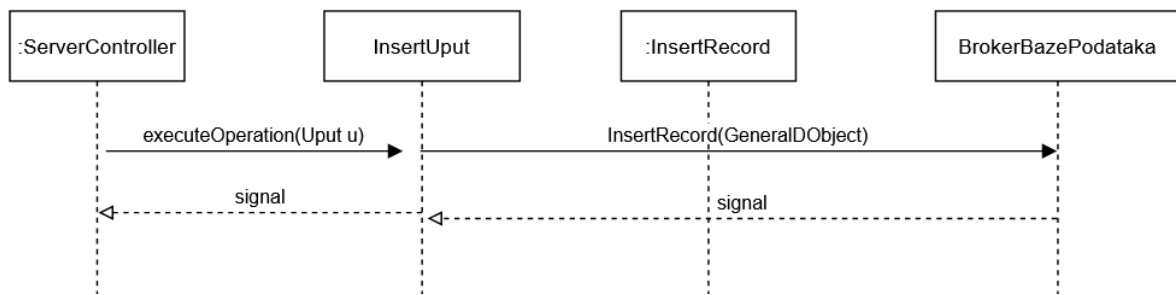
8. Уговор UG3: *Zapamti uput*

Операција: **InsertUput**(Uput):signal;

Веза са СК: *CK4*

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

Постуслови: *Запамћен је упут за лабораторијске анализе*



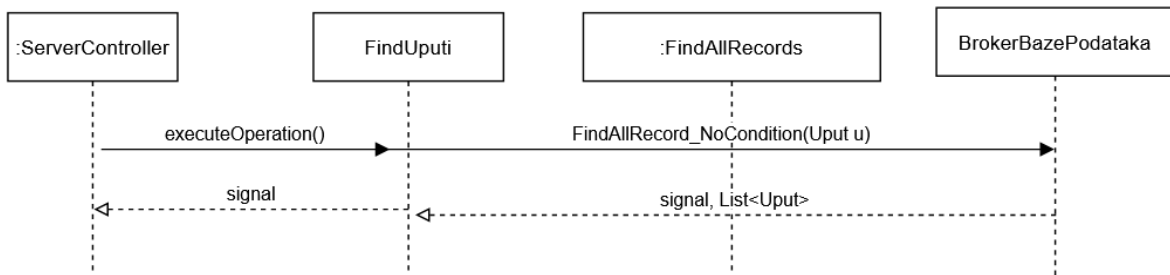
9. Уговор UG4 *UcitajUpute*

Операција: **FindUputi**(List<Uput>):signal;

Веза са СК: *CK5*

Предуслови:

Постуслови: *Учитана је листа упута.*



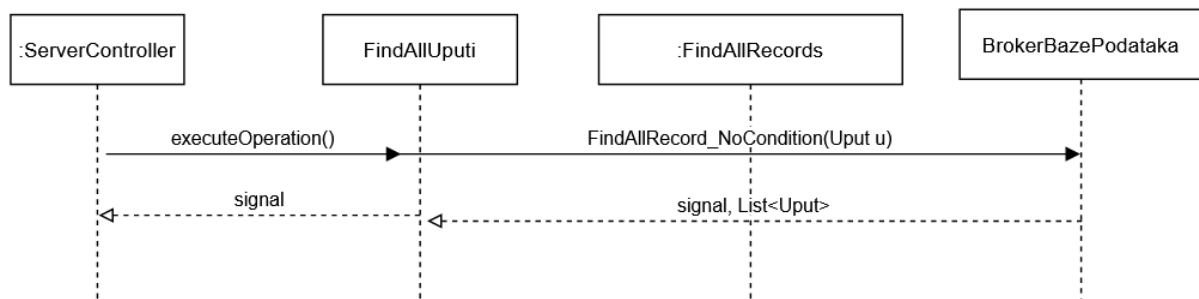
10. Уговор UG5: *UcitajAnalizaUputa*

Операција: **UcitajAnalyzeUputa**(Uput u):signal;

Веза са СК: *CK5*

Предуслови:

Постуслови: Учитана је листа анализа упута.



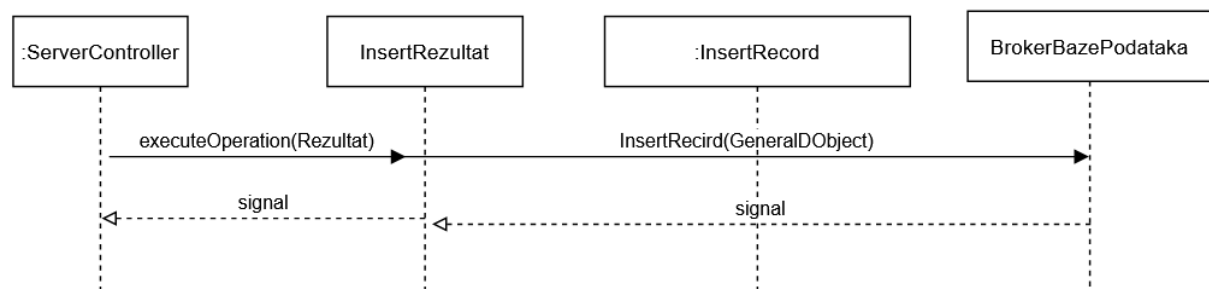
11. Уговор UG6: *Zapamti*

Операција: InsertRezultat(Rezultat): Signal

Веза са СК: CK6

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

Постуслови: Резултат је запамћен.



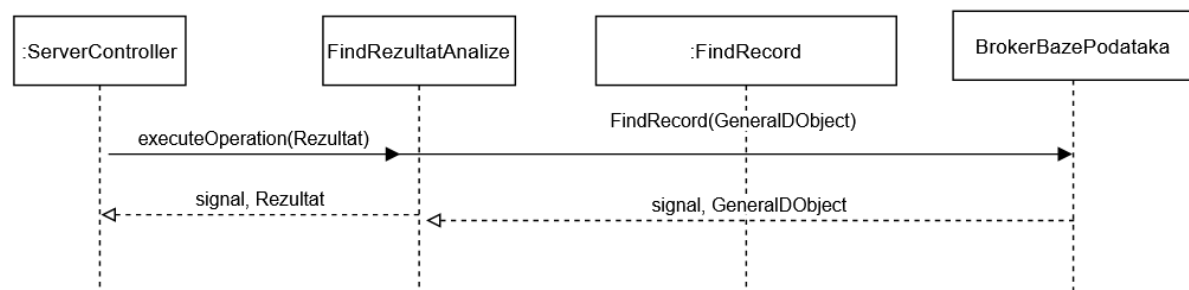
12. Уговор UG7: *UcitajRezultat*

Операција: FindRezultatAnalyze(Rezultat):signal;

Веза са СК: CK7

Предуслови:

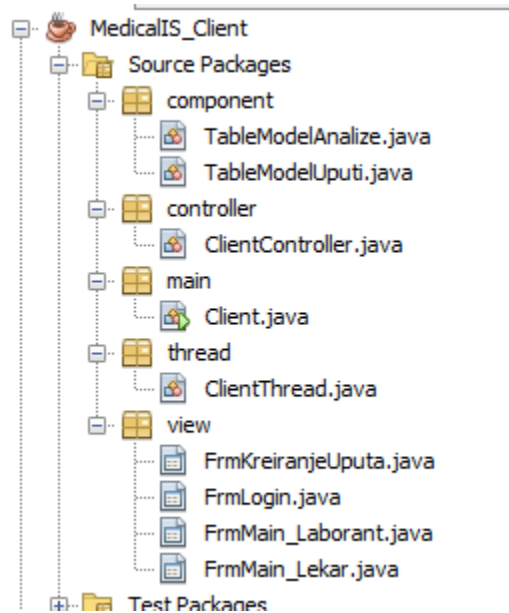
Постуслови: Учитан је резултат анализе



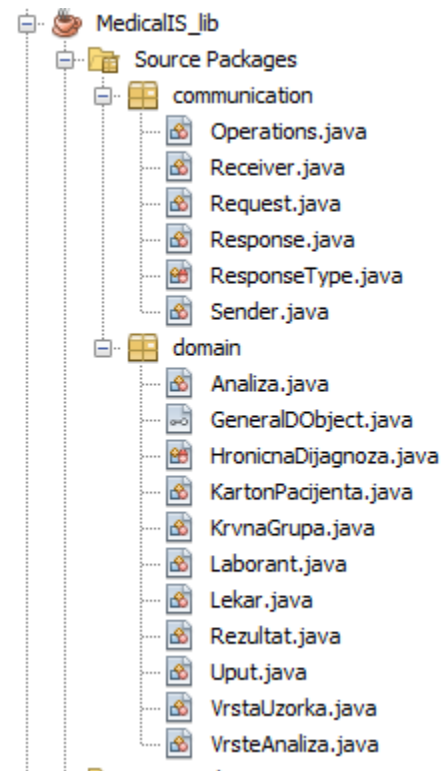
Имплементација

Систем је имплементиран кроз следеће класе:

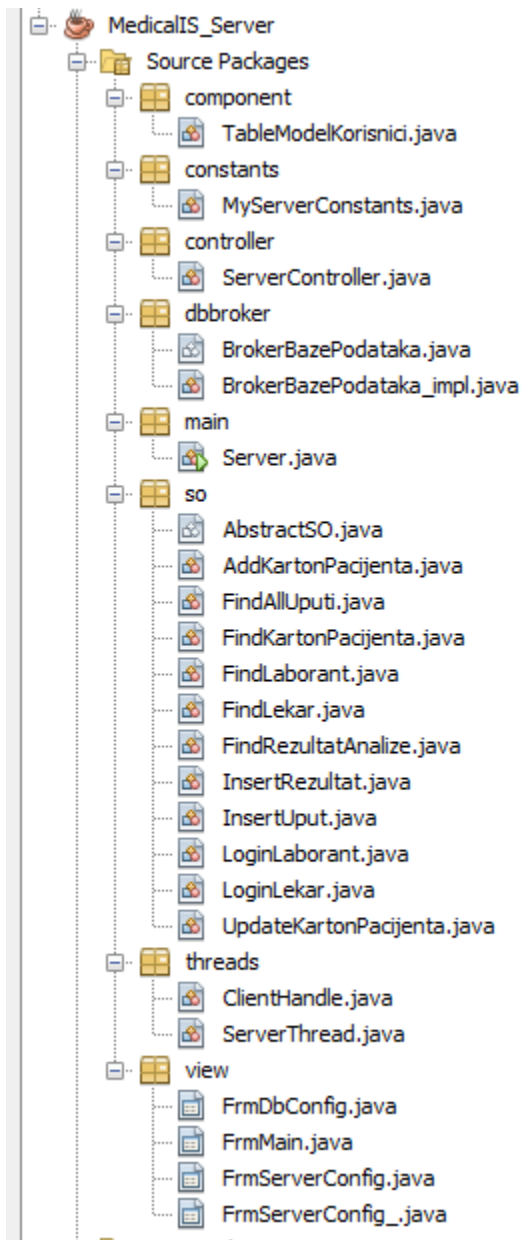
Клијентска страна



Заједничке класе



Серверска страна



Тестирање

Као финалну фазу пројектовања система, урађено је тестирање целог програма, како би се провериле функционалности.