

УНИВЕРЗИТЕТ У БЕОГРАДУ
ФАКУЛТЕТ ОРГАНИЗАЦИОНИХ НАУКА

ЗАВРШНИ РАД

Развој софтверског система за паровање
на шаховским турнирима у Јава окружењу

Ментор

Др Илија Антовић

Студент

Марија Тодоровић 64/2018

Београд, 2023. Године

*За моје маму и баку, захваљујући којима сам
ценила вредност образовања и пре него што сам
сама успела да је спознам.*

АПСТРАКТ

У овом раду пројектовано је софтверско решење за организацију шаховских турнира које се састоји из два међусобно зависна модула. Један од њих намењен је судијама шаховских турнира и осим за уношење резултата партија, генерисање свих кола у току турнира и проглашавање победника турнира, служи и као база играча, турнира и клубова. Други модул користе играчи како би из листе активних турнира изабрали на ком желе да учествују, пријавили се на исти и евентуално поништили пријаву у случају одустајања.

Циљ овог рада је да поједностави горе наведени процес, што је потребно пре свега зато што у односу на остале спортиве, шах има специфичан начин одређивања такозваног жреба, односно тога ко ће против кога играти и у ком колу. Осим тога што се двојица такмичара не могу састати више пута у току турнира (осим на специфичним турнирима као што је првенство за првака света и слично, који нису обрађени у овом раду), постоје многобројни критеријуми који одређују ко када треба да игра против кога. Сви они укључени су у логику овог рада који осим те основне свхре представља и базу играча, турнира и клубова за судије, а сајт везан за актуелности у свету шаха за играче.

У раду је коришћен програмски језик Java 18.0.2.1, окружење NetBeans 15 IDE, MySQL server, JDBC библиотека, JavaScript програмски језик, PHP програмски језик, Laravel framework PHP-а, ReactJS библиотека и језик форматирања CSS. Коришћени су и неки од напредних концепата набројаних технологија које представљају основу овог рада.

Без обзира на компликоване захтеве које је са собом носила идеја овог рада, наведени алати показали су се као успешни у решавању најразличитијих проблема конкретног пословног процеса и њиховог превођења у “програмерске” задатке. Креирање ове апликације показатељ је колико увођење софтверских решења у конвенционалне процесе доприноси ефикасности спровођења истих.

САДРЖАЈ

1.УВОД.....	1
2.ПРЕГЛЕД КОРИШЋЕНИХ ТЕХНОЛОГИЈА.....	2
2.1 РАЗВОЈНА ОКРУЖЕЊА.....	2
2.2 Java програмски језик.....	3
2.2.1 Java with Ant.....	5
2.2.2 JDBC.....	5
2.2.3 Java Swing.....	6
2.3 ReactJS.....	6
2.4 Laravel.....	8
2.5 JavaScript.....	8
2.6 Bootstrap.....	9
2.7 CSS.....	9
2.8 AJAX.....	10
2.9 Комуникација између два модула.....	11
3.СТУДИЈСКИ ПРИМЕР.....	8
3.1 РАЗВОЈ СУДИЈСКОГ МОДУЛА ЗА ИЗВРШАВАЊЕ АДМИНИСТРАТОРСКИХ ФУНКЦИЈА.....	8
3.1.1 Вербални опис система.....	8
3.1.2 Прикупљање корисничких захтева.....	9
3.1.2.1 СК1: Случај коришћења - Пријављивање судије на систем.....	10
3.1.2.2 СК2: Случај коришћења - Креирање играча.....	11
3.1.2.3 СК3: Случај коришћења - Претрага клубова.....	12
3.1.2.4 СК4: Случај коришћења - Претраживање турнира.....	12
3.1.2.5 СК5: Случај коришћења - Претраживање играча.....	13
3.1.2.6 СК6: Случај коришћења - Унос резултата кола.....	14
3.1.2.7 СК7: Случај коришћења - Креирање кола.....	15
3.1.2.8 СК8: Случај коришћења - Креирање турнира.....	15
3.1.2.9 СК9: Случај коришћења - Провера пласмана.....	16
3.1.2.10 СК10: Случај коришћења - Измена података о турниру.....	17
3.1.3 Фаза анализе.....	18
3.1.3.1 ДС1: Дијаграм секвенце случаја коришћења - пријављивање на систем.....	18
3.1.3.2 ДС2: Дијаграм секвенце случаја коришћења - Креирање играча...	19
3.1.3.3 ДС3: Дијаграм секвенце случаја коришћења - Претрага клубова..	20
3.1.3.4 ДС4: Дијаграм секвенце случаја коришћења - Претраживање турнира.....	21
3.1.3.5 ДС5: Дијаграм секвенце случаја коришћења - Претраживање играча.....	23
3.1.3.6 ДС6: Дијаграм секвенце случаја коришћења - Унос резултата кола..	25

3.1.3.7 DC7: Дијаграм секвенце случаја коришћења - Креирање кола.....	26
3.1.3.8 DC8: Дијаграм секвенце случаја коришћења - Креирање турнира.	27
3.1.3.9 DC9: Дијаграм секвенце случаја коришћења - Провера пласмана.	28
3.1.3.10 DC10: Дијаграм секвенце случаја коришћења - Измена података турниру.....	29
3.1.3.11 Понашање софтверског система - дефинисање уговора о системским операцијама.....	31
3.1.3.12 Структура софтверског система - концептуални (доменски) модел.	34
3.1.3.13 Структура софтверског система - Релациони модел.....	34
3.1.4 ФАЗА ПРОЛЕКТОВАЊА.....	39
3.1.4.1. Архитектура софтверског система.....	39
3.1.4.2 Пројектовање корисничког интерфејса.....	39
3.1.4.3. Пројектовање екранских форми.....	40
3.1.4.4 CK1: Случај коришћења - Пријављивање судије на систем.....	45
3.1.4.5 CK2: Случај коришћења - Креирање играча.....	46
3.1.4.6 CK3: Случај коришћења - Претрага клубова.....	49
3.1.4.7 CK4: Случај коришћења - Претрага турнира.....	52
3.1.4.8 CK5: Случај коришћења - Претрага играча.....	54
3.1.4.9 CK6: Случај коришћења - Унос резултата кола.....	56
3.1.4.10 CK7: Случај коришћења - Креирање кола.....	58
3.1.4.11 CK8: Случај коришћења - Креирање турнира.....	60
3.1.4.12. CK9: Случај коришћења - Провера пласмана.....	61
3.1.4.13 CK10: Случај коришћења - Измена података о турниру.....	62
3.1.4.14 Пројектовање контролера корисничког интерфејса.....	66
3.1.4.15 Пројектовање апликационе логике.....	66
3.1.4.16 Комуникација са клијентима.....	67
3.1.4.17 Пројектовање контролера апликационе логике.....	69
3.1.4.18 Пројектовање пословне логике.....	71
3.1.4.19 Пројектовање структуре софтверског система.....	77
3.1.4.20 Пројектовање брокера базе података.....	77
3.1.4.21 Пројектовање складишта података.....	78
3.1.4.22 Комплетна архитектура софтверског система.....	80
3.1.5 Имплементација.....	81
3.1.5.1 Пројекат SahKlijent:.....	81
3.1.5.2 Пројекат SahZajednicki.....	82
3.1.5.3 Пројекат SahServer.....	82
3.1.6 Тестирање.....	83
3.2 РАЗВОЈ ИГРАЧКОГ МОДУЛА ЗА ПРИЈАВУ/ОДЈАВУ УЧЕШЋА НА ТУРНИРИМА.....	83
3.2.1 Вербални опис система.....	84
3.2.2 Прикупљање корисничких захтева.....	84

3.2.2.1 СК1: Случај коришћења - Пријављивање играча на систем.....	84
3.2.2.2 СК2: Случај коришћења - Регистрација новог играча.....	85
3.2.2.3 СК3: Случај коришћења - Пријављивање играча на турнир.....	85
3.2.2.4 СК4: Случај коришћења - Претрага турнира.....	86
3.2.2.5 СК5: Случај коришћења - Одјављивање играча са турнира.....	87
3.2.3 Фаза анализе.....	87
3.2.3.1 ДС1: Дијаграм секвенце случаја коришћења - Пријављивање играча на систем.....	87
3.2.3.2 ДС2: Дијаграм секвенце случаја коришћења - Регистрација новог играча.....	88
3.2.3.3 ДС3: Дијаграм секвенце случаја коришћења - Пријављивање играча на турнир.....	89
3.2.3.4 ДС4: Дијаграм секвенце случаја коришћења - Претрага турнира..	90
3.2.3.5 ДС5: Дијаграм секвенце случаја коришћења - Одјављивање играча са турнира.....	91
3.2.3.6 Понашање софтверског система - дефинисање уговора о системским операцијама.....	92
3.2.3.7 Структура софтверског система - концептуални (доменски) модел...	92
3.2.3.8 Структура софтверског система - релациони модел.....	93
3.2.4 Пројектовање.....	96
3.2.4.1 Архитектура софтверског система.....	96
3.2.4.2 Пројектовање корисничког интерфејса.....	96
3.2.4.3 Пројектовање екранских форми.....	96
3.2.4.4 СК1: Случај коришћења - Пријављивање играча на систем.....	98
3.2.4.5 СК2: Случај коришћења - Регистровање новог играча.....	99
3.2.4.6 СК3: Случај коришћења - Претрага турнира.....	100
3.2.4.7 СК4: Случај коришћења - Пријава на турнир.....	102
3.2.4.8 СК5: Случај коришћења - Одјављивање са турнира.....	103
3.2.4.9 Пројектовање контролера корисничког интерфејса.....	104
3.2.4.10 Пројектовање апликационе логике.....	105
3.2.4.11 Комуникација са клијентима.....	105
3.2.4.12 Пројектовање контролера апликационе логике.....	107
3.2.4.13 Пројектовање пословне логике.....	107
3.2.4.14 Пројектовање структуре софтверског система.....	110
3.2.4.15 Пројектовање складишта података.....	111
3.2.4.16 Комплетна архитектура софтверског система.....	112
3.2.5 Имплементација.....	115
3.2.6 Тестирање.....	117
4. ЗАКЉУЧАК.....	117
Литература.....	119

1.УВОД

До недавно се о појави интернета и укључивању технологија у многе сегменте привреде, а затим и свакодневице причало као о актуелном тренду чија се експанзија неочекивано и брзо дододила. Међутим, укорењеност ИТ сектора у све сфере пословања врло брзо је престала да буде новитет и постала стандард. Уска повезаност дигитализације са унапређивањем је додатно продубљена и тешко је замислiti да савремено предузеће функционише без постојања исте.

Сходно све већој заступљености и потреби за прилагођавањем, интернет технологије су се убрзано развијале. Пратећи хронологију развоја апликација какве су најчешће биле потребне компанијама, очигледна је значајна разлика између статичких веб страница са основним информацијама какве су развијане у почетку и многобројним могућностима за решавање проблема било које врсте које постоје за кориснике интернета данас.

Код предузећа, група самосталних програмера и појединача чије пословање је рас прострањено на више географских локација јавила се потреба за апликацијама које ће бити велике, дистрибуиране и преносиве. Прикупљање, анализа и обрада података поједностављени су појавом могућности за приступање апликацијама са више различитих уређаја. Како су ширењем спектра могућности које пружају, веб апликације постале непходан део пословања сваке компаније, тимовима за њихово креирање додељене су и нове одговорности. Апликације су престале бити само извор информација за оне који са њима раде и полако почеле добијати улогу у гаранцији поузданости компаније, комуникацији са клијентима и заинтересованим странама, архивирању разних садржаја, грађењу имиџа компаније, спровођењу интернет куповине и многим другим сферама пословања.

Интеграција интернет технологија довела је до оптимизације пословања, смањења трошкова, повећања ефикасности и ефективности и дефинисања степена технолошког развитка као новог параметра угледа компаније. Уз повећање обима неопходних функционалности, дошло је и до унапређења самих апликација. Како не би заостале за новим захтевима, компаније све више прелазе са централизованих десктоп апликација на веб апликације. Основна разлика између ова два појма је у томе што веб апликација нуди могућност извршавања на више различитих уређаја. Осим тога, у савременим веб апликацијама пројектује се интерактиван и прилагодљив кориснички интерфејс што даје велики допринос у задовољењу потреба крајњих корисника, док из угла инжењера који их пројектују, чињеница да сви клијенти приступају истој апликацији значајно олакшава одржавање, дистрибуцију и сам развој апликације.

Узевши у обзир брзе промене у савременим трендовима и експоненцијални пораст потребе за веб програмирањем, очекивана је и појава многобројних нових алата којој сведочимо. Неки од њих обрађени су у наставку овог рада.

У уводном поглављу дат је кратак приказ основних спецификација сваке од коришћених технологија и објашњење елемената кључних за разумевање начина функционисања исте.

У трећем поглављу је представљен студијски пример, односно практична апликација реализована коришћењем технологија наведених у другом поглављу. Конкретан пословни процес за који је развијан софтверски систем је организација шаховских турнира и то је рађено применом ReactJS и Java технологија кроз поједностављени поступак Ларманове методе чије су фазе: дефинисање корисничких захтева, анализа, пројектовање, имплементација и тестирање. Дефинисање корисничких захтева је описано кроз моделе случајева коришћења где је за сваки детаљно написан редослед извршавања операција са податком о врсти операције којој наведена припада (АПУСО, АПСО, АНСО, ИА). У фази анализе представљени су претходно дефинисани случајеви коришћења преко дијаграма секвенци. Фаза пројектовања садржи по корацима описано пројектовање корисничког интерфејса, апликационе логике и складишта података, а на самом крају поглавља приказана је комплетна архитектура система. У фази имплементације описане су практичне имплементационе компоненте система, које представљају реализацију компоненти из фазе пројектовања. Компоненте добијене у фази имплементације, ручно су тестиране у фази тестирања која осим резултата тестирања, садржи и опис технологија коришћених за вршење истог. Закључно поглавље даје предлоге за даље унапређење програма и правце развоја које би требало пратити у свакој од фаза пројектовања овог софтверског система, као и разматрања везаних за тренутне и потенцијалне могућности истог.

2. ПРЕГЛЕД КОРИШЋЕНИХ ТЕХНОЛОГИЈА

2.1 РАЗВОЈНА ОКРУЖЕЊА

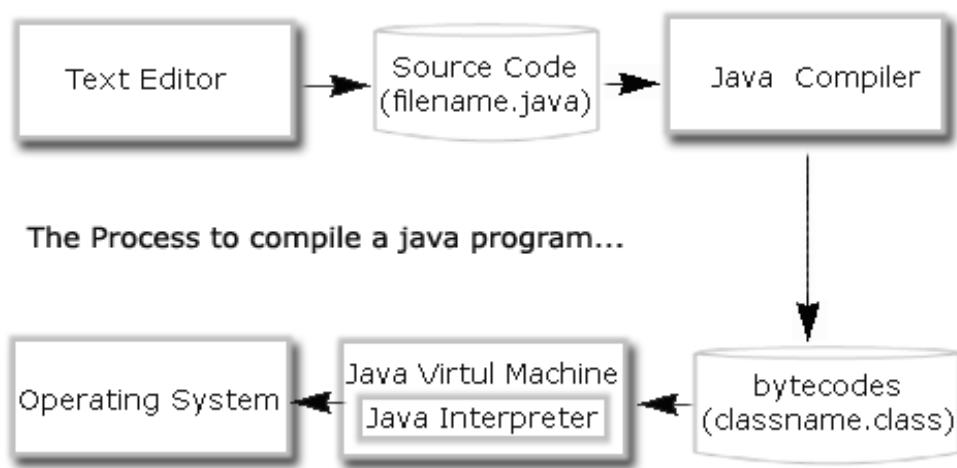
За писање кода на серверској страни ове апликације коришћен је Apache NetBeans IDE 15. То је интегрисано развојно окружење које се углавном окристи за пројекте у Јава програмском језику, али може се прилагодити и другим. Подржава велики број алата и технологија које позитивно утичу на побољшање квалитета развојног процеса апликације. Врло је лако користити га, поготово када су у питању функционалности везане за прављење графичког корисничког интерфејса. Такође, огроман број корисника сматра да има одличну корисничку подршку и да је најпогоднији за почетнике.

За писање кода који се тиче клијентске стране, за фронтенд као и за бекенд, коришћен је Visual Studio Code. Једноставан је за коришћење и може се лако инсталорати на било којој платформи. Има уградјену подршку за Node.js, TypeScript, JavaScript. Поседује велики број екstenзија које се могу користити у раду са већином актуелних програмских језика.

2.2 Java програмски језик

Јава је објектно оријентисани програмски језик заснован на класама чија је главна карактеристика и предност у томе што минимално зависи од имплементације. То је програмски језик опште намене намењен да дозволи програмерима апликација да пишу код једном, а покрећу га било где и било кад, што значи да компајлирани Јава код може да ради на свим платформама које подржавају Јаву без потребе за поновним компајлирањем.

Све то постиже се захваљујући Јава виртуелној машини(Java Virtual Machine - JVM). То је виртуелни процесор и један од два неопходна дела за извршавање Јава апликација. Други од њих је Java Runtime Environment (JVE).

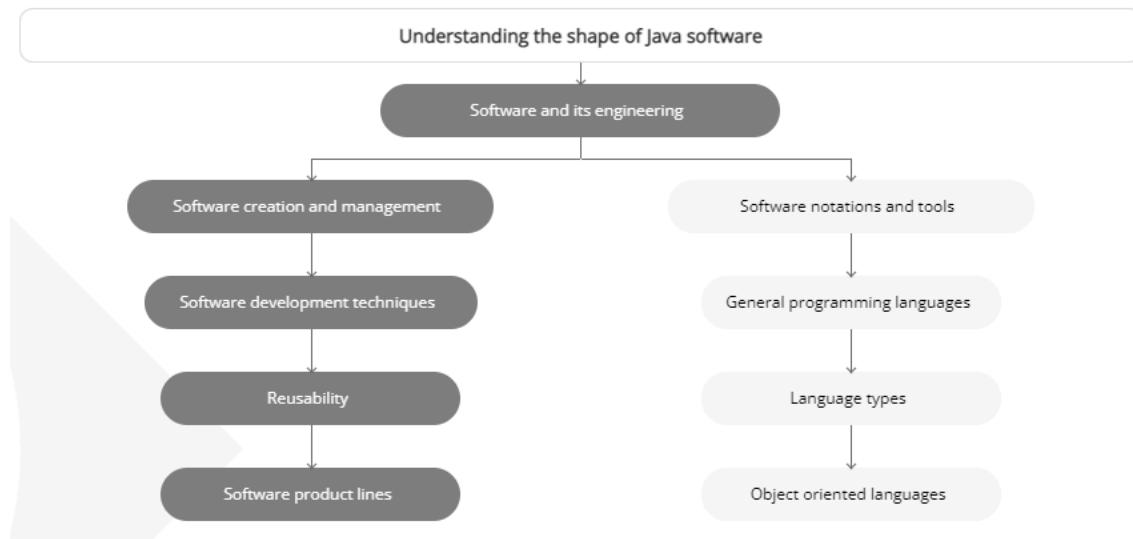


Слика 1. Извршавање Јава програма

Јава апликације се обично компајлирају у бајт код који може да ради на било којој Јава виртуелној машини без обзира на основну архитектуру рачунара.¹[1]

Синтакса Јаве је слична Џ и Џ++, али има мање објеката ниског нивоа од било које од њих. Java Runtime пружа динамичке могућности (као што су рефлексија и модификација кода током извршавања) које у већини програмских језика нису доступне.

¹ Gosling, J., Holmes, D. C., & Arnold, K. (2005). The Java programming language.



Слика 2. Конструкција софтвера програмског језика Јава²[2]

У зависности од специфичности софтвера и домена примене постоје следећа издања Јава платформе:

1. Java card – Користи се за развој апликација које ће се покретати на паметним картицама и другим уређајима мале меморије.
2. Java ME (Micro Edition) – је скуп извршних окружења за развој апликација за мобилне уређаје као што су телефони, пејџери и навигациони уређаји.
3. Java SE (Standard Edition) – садржи пакете који се најчешће користе у стандардном програмирању. Чини је стандардна библиотека која се користи у аплетима, Јава ЕЕ апликацијама, Јава МЕ и конзолним уређајима.
4. Java EE (Enterprise Edition) – представља проширење и надоградњу Java SE.

Користи се за развој и извршавање великих, вишенивојских, скалабилних, поузданих и сигурних мрежних апликација.

У породици Java Standard Edition два главна софтверска производа обезбеђује Oracle:

- Java SE Runtime Environment (JRE) - обезбеђује библиотеке, Јава виртуелну машину и друге компоненте неопходне за покретање аплета и апликација написаних у програмском језику Јава. Ово окружење за извршавање може се редистрибуирати са апликацијама како би биле самостојеће.
- Java SE Development Kit (JDK) - развојни комплет ЈДК укључује ЈРЕ и развојне алате командне линије као што су компајлери и програми за отклањање грешака који су неопходни или корисни за развој аплета и апликација.

² Baxter, G., Frean, M., Noble, J., Rickerby, M., Smith, H., Visser, M., ... & Tempero, E. (2006, October). Understanding the shape of Java software. In *Proceedings of the 21st annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications* (pp. 397-412).

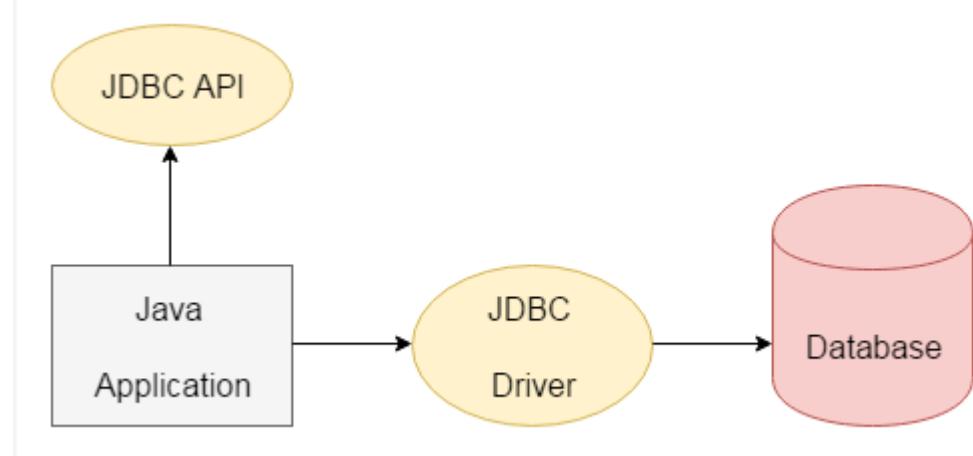
2.2.1 Java with Ant

[Apache Ant](#) (“Another Neat Tool”) је Јава библиотека која се користи за аутоматизацију процеса изградње за Јава апликације. Датотеке прављење у Ант-у су писане у XML и по конвенцији се зову build.xml.³[3]

Главна предност Анта је његова флексибилност. Ант не намеће никакве конвенције кодирања или структуре пројекта. Сходно томе, то значи да Ант захтева од програмера да сами напишу све команде, што понекад доводи до огромних XML датотека за изградњу које је тешко одржавати.

2.2.2 JDBC

JDBC је скраћеница од Java Database Connectivity и представља ODBC (Open Database Connectivity) који обезбеђује Microsoft. То је Java API за повезивање и извршавање упита над базом података. Саставни је део JavaSE (Java Standard Edition) и користи JDBC драјвере за повезивање са базом. Користећи овај програмски и интерфејс апликације, могуће је вршити CRUD операције над подацима из било које релационе базе.



Слика 3. Улога JDBC driver-а, извор: [11]

Често коришћени интерфејси који припадају овом API-ју су: Driver, Connection, Statement, PreparedStatement, ResultSet, RowSet, CallableStatement, ResultSetMetaData, DatabaseMetaData... Класа коришћена у овом раду је DriverManager, а још неке од познатих класа су Blob, Clob, Types.

Главне предности коришћена jdbc у односу на остале начине повезивања програма са базом огледају се у могућности да се ради са већ постојећим подацима, односно да се може доћи до информација из база имплементираних у другим системима. Осим тога, имплементација је врло једноставна, нема потребе за инсталацијом нити за мрежном конфигурацијом већ се конекција на базу идентификује уз помоћ URL-а. Омогућен је приступ мета подацима који су предуслов за развој комплексних апликација.

³ <https://www.baeldung.com/ant-maven-gradle>

2.2.3 Java Swing

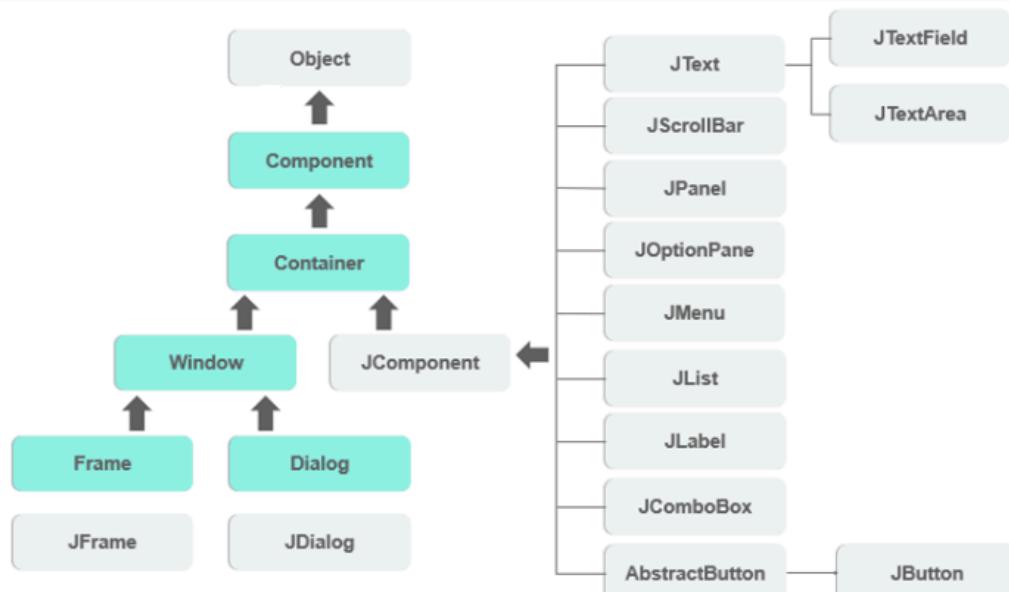
Java SE има три библиотеке које се користе за графичко програмирање: AWT, Swing, JavaFX. У овом раду коришћена је Swing библиотека која је првобитно настала како би се превазишли недостаци прве међу набројаним библиотекама који се пре свега огледају у поседовању само оних компонената које постоје на свим оперативним системима.

Swing библиотека је објављена у оквиру J2SE 1.2. У потпуности је написана у Јави и садржи многобројне могућности за креирање комплексних визуелних апликација уз помоћ класа и интерфејса. Она је део веће библиотеке Java Foundation Classes (JFC) која осим основне AWT библиотеке, укључује и библиотеку Java 2D за рад са графиком у равни.

Container class се у Swing-у назива свака класа која има друге компоненте у себи. Да би се покренула апликација са графичким корисничким интерфејсом, неопходно је да постоји бар једна класа која припада овој групи. Постоје три типа container класе;

- Панел (JPanel) – Користи се за организовање компоненти на прозору
- Оквир (JFrame) – Потпуно функционалан прозор са иконама и насловима
- Дијалог (JDialog) – То је као искачући прозор, али није потпуно функционалан као оквир

Све компоненте у свингу као што су JButton, JComboBox, JList, JLabel и остале су наслеђене класе класе JComponent која се може додати у панел, дијалог или оквир. Основне Swing компоненте су саставни блокови било које графичке апликације.



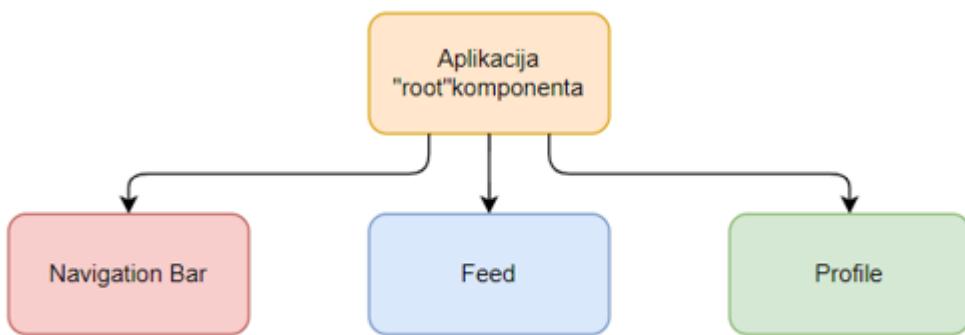
Слика 4. Хијерархија елемената у библиотеци Java Swing, извор: [10]

2.3 ReactJS

React (ReactJS) је Јаваскрпит библиотека отвореног кода која обезбеђује преглед података записаних преко XTMЛ-а у виду single-page страница.

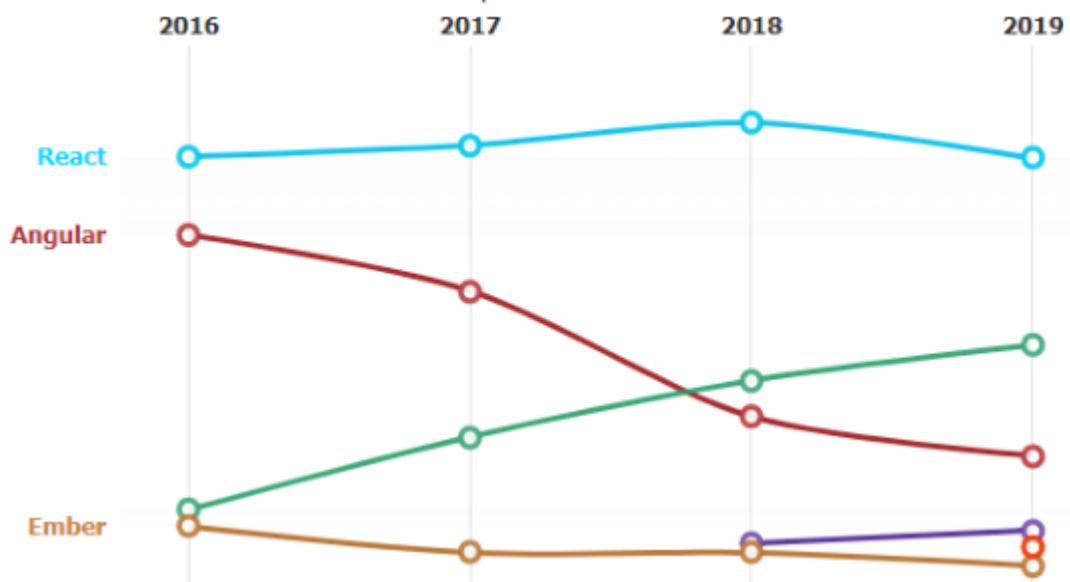
Она омогућава једноставно прављење интерактивног корисничког интерфејса и писање кода високих перформанси користећи концепт компонената. Компоненте служе да поједноставе прављење жељеног корисничког интерфејса тако што ће програмер избеги да води рачуна о томе на који начин се оне на крају приказују на страници.

Свака апликација састоји се од скупа независних компонената које се међусобно повезују у сложену апликацију. Свака апликација мора имати “root” компоненту.



Слика 5. Пример компонената које чине једну Реакт апликацију

У односу на остале веб странице, оне прављење у Ријекту су значајно брже с обзиром на то да се при промени странице не учитава цела апликација, већ само та одређена страница. Такође, омогућено је прављење интерактивних елемената интерфејса.

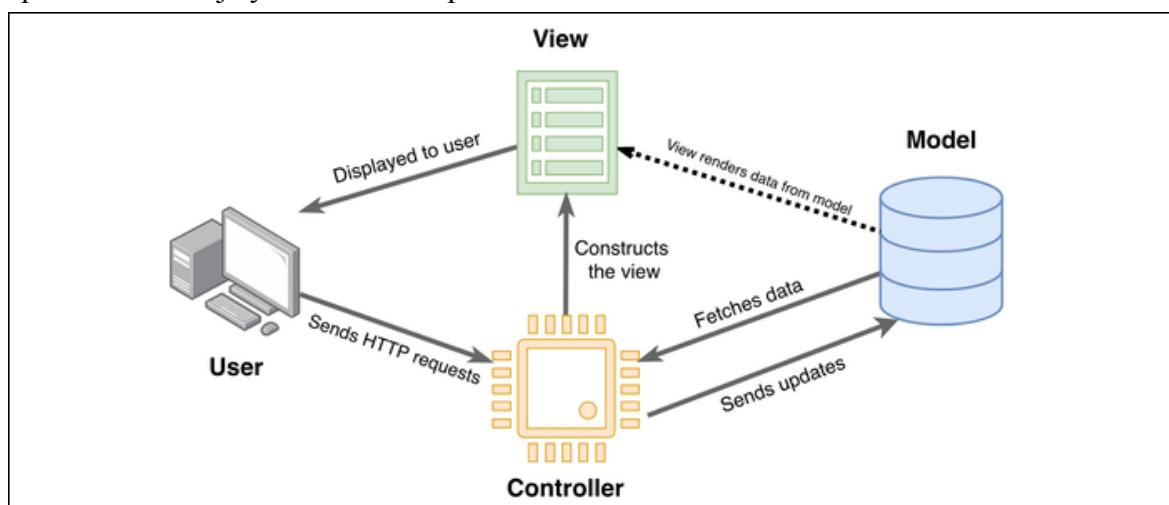


Слика 6. Употреба ReactJS у поређењу са употребом других JavaScript фрејмворка

2.4 Laravel

Laravel је бесплатни PHP веб оквир отвореног кода који је креирао Тейлор Отвел и намењен је развоју веб апликација које прате архитектонски образац модел-поглед-контролер (MVC) и заснован на Sifoni-ju. Неке од карактеристика Ларавела су модуларни систем паковања са наменским менаџером зависности, различити начини за приступ релационим базама података, услужни програми који помажу у примени и одржавању апликација.

Предност ларавела у односу на остале php оквире је у томе што је већи део основног кода аутоматски генерисан, па се при раду са великим количином података не губи време на писање подразумеваних функција. Ларавел са базом комуницира уз помоћ http протокола, слањем захтева којима се врше основне операције над базом (CRUD). Ларавел са фронт делом апликације (React) комуницира путем axios-а. Axios је http клијент који упућује захтеве API сервису тако да се комуникација у овој апликацији практично одвија уз помоћ rest api.



Слика 7. Модел по ком ради Ларавел

Ларавел се првенствено користи за прављење прилагођених веб апликација помоћу PHP-а. То је веб оквир који се бави многим стварима које је досадно да сами направите, као што су рутирање, шаблонски HTML и аутентификација.⁴[4]

2.5 JavaScript

JavaScript је програмски скрипт језик за чији је развој заслужна компанија коју данас зnamо као Mozilla. Функционише на свим познатим прегледачима, а популарност коју је брзо стекао дuguје, између остalog и сличности са Јава програмским језиком иако је, што многи не знају, потпуно засебан програмски језик.

Првенствено се користи за додавање функционалности и динамичког понашања интернет страницама прављеним као делови десктоп апликација. Функције написане у њему углавном се укључују тако што се пишу у XTMЛ страницама.

⁴ Anthony Heedings, 2020, What is Laravel and how do you get started with it?
<https://www.howtogeek.com/devops/what-is-laravel-and-how-do-you-get-started-with-it/>

Служи за развој мобилних и рачунарских апликација, а приметно велику примену има и у развоју игрица.

На страницу може се додати на два начина, а како се у овом пројекту додавао на PHP странице, приказани су начини како се то ради управо у том случају:

1. Унутар етикета:

```
<script> nekiKodNapisanUJavaScriptuUnutarPHPStranice </script>
```

2. Увежен из спољашње датотеке

```
<script src="dokumentUJAvAScriptu.js"></script>
```

2.6 Bootstrap

Bootstrap је један од најкоришћенијих радних оквира за развој корисничког интерфејса веб апликација. Представља колекцију алата за израду веб страница уз помоћ HTML-а, CSS-а и JavaScript-а.⁵[5]

Састоји се из низа CSS компоненти попут грид система за мобилне уређаје, посебно стилизованих компоненти, дугмади, падајућих менија, као и Javascript библиотеке које подржавају стилизоване компоненте.

2.7 CSS

CSS је технологија која служи за раздвајање изгледа странице од њене већ претходно дефинисане структуре уз помоћ неке друге технологије, најчешће html. HTML странице служе да прикажу информације које треба да буду презентоване на страници, а опис изгледа странице и начин представљања тих информација је улога CSS-а.

Стилови у CSS-у служе да дефинишу изглед елемената странице на једном месту, мењају изглед целих страница или делова страница одједном, мењају изглед свих елемената неке странице, дефинишу величину слова и остале ствари везане за формат и фонт текстуалног дела странице, позиционирају садржај странице са прецизношћу на нивоу пиксела, редефинишу изглед постојећих тагова... Они се могу применити на три начина:

- Повезивањем HTML странице и неког екстерног фајла са екstenзијом .css
- Унутар странице у head делу кода у оквиру `<style>` тага
- Директно у атрибуту неког тага који треба да се форматира

⁵ Bootstrap, <https://getbootstrap.com>, Јануар 2021.

2.8 AJAX

AJAX је скраћеница од “Asynchronous JavaScript and XML”. То је развојна техника која служи за креирање интерактивних садржаја на вебу. Заснива се на размени што мање количине података са сервером у циљу спречавања појаве да корисник стално мора да освежава страницу. То чини страницу интерактивнијом и ефикаснијом.

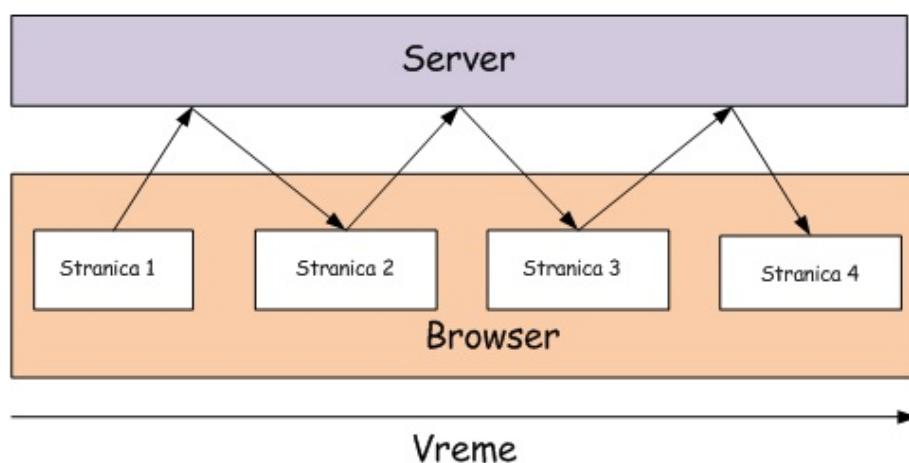
Ајакс се не може назвати технологијом сам по себи, већ представља термин који означава коришћење групе технологија. Обухвала презентацију базирану на стандардима користећи XHTML и CSS, динамички приказ и интеракцију преко Document Object Model-а, размену и манипулацију над подацима користећи XML, асинхроно прикупљање података уз помоћ XMLHttpRequest и JavaScript који повезује све у целину.

Циљ AJAX-а је да омогући да се свака страница на којој се одвија веб апликација учита само једном, што сакрива комуникацију са сервером од корисника тако што избегава поновно учитавање целе странице. Поједностављено речено, уз помоћ ајакса, сваки пренод података између сервера и клијента се врши у позадини што је могуће захваљујући JavaScript.

У уводном поглављу рада било је речи о предности веб апликација у односу на десктоп апликације. Ако ствари погледамо из другог угла, предност десктоп апликација у односу на веб била је у томе што су десктоп апликације давале веће могућности у виду функција и информација које нуде. Управо да би се овај недостатак надоместио, у изради веб апликација почeo је да се користи AJAX.

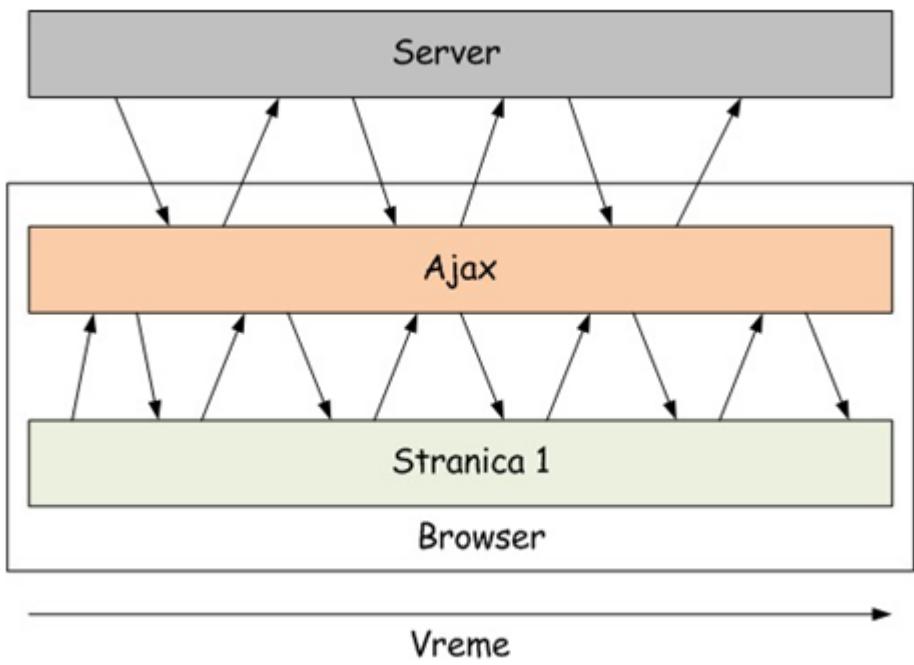
Значајно за разумевање главног доприноса коришћења AJAX-а је разумети разлику између синхроне и асинхроне комуникације између клијента и сервера.

- Код синхроне комуникације, након што клијент пошаље захтев серверу, подаци дођу до сервера, сервер их обрађује и онда враћа. Дакле, не постоји начин да се избегне чекање због тога што се захтеви обављају један за другим чак и када за тим нема потребе.



Слика 8: синхронна комуникација сервера и прегледача, извор: [9]

- Асинхронна комуникација омогућава да се корисничко коришћење веб апликације никада не прекида. Захваљујући AJAX-у, корисник може да настави да користи апликацију без обзира на то да ли ће систем одговорити на његов захтев одмах или после неког времена.



Слика 9: асинхрона комуникација између сервера и претраживача којој посредује ајах, извор: [9]

2.9 Комуникација између два модула

У раду су вршene операције над SQL базом (релациона база података) и као платформа за спровођење истих коришћен је SQLyog community. Измене над базом које се дешавају са клијентске стране спровођене су преко REST API протокола који је из PHP Laravel-а прослеђивао http захтеве.

Серверска страна са базом комуницира уз помоћ дефинисаних интерфејса и класи које садрже Јавине библиотеке за слање SQL упита. Коришћени предефинисани елементи су: java.sql.Connection, java.sql.DriverManager, java.sql.Exception. Сви наведени припадају једној од најзаступљенијих библиотека Јава програмског језика: jdbc. Информације о порту преко којег ће се повезати корисник серверске стране

апликације (судија) са базом и самим програмом и подацима неопходним за приступ локалној бази података налазе се у properties фајлу у самом програму.

3.СТУДИЈСКИ ПРИМЕР

3.1 РАЗВОЈ СУДИЈСКОГ МОДУЛА ЗА ИЗВРШАВАЊЕ АДМИНИСТРАТОРСКИХ ФУНКЦИЈА

3.1.1 Вербални опис система

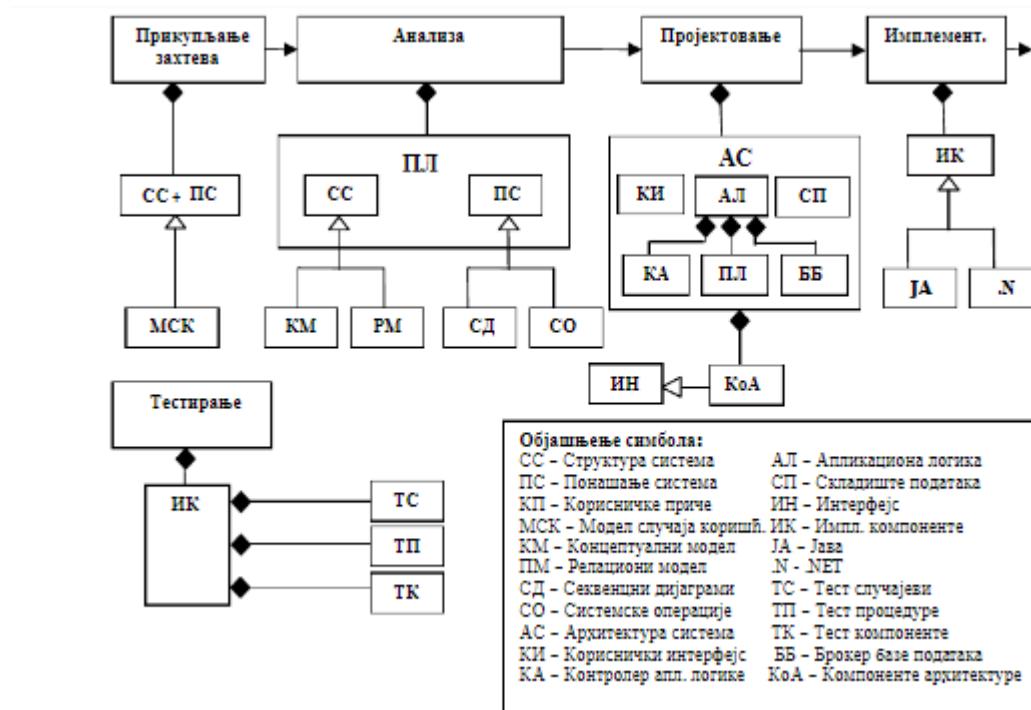
У овом раду направљена је апликација чија је главна сврха организација шаховских турнира и комуникација између актера истих.

Модул за судије је предвиђен за коришћење у току неког шаховског турнира. Судије задужене за суђење тих турнира имају приступ апликацији. Главна сврха јој је да уместо њих уноси резултате партија, генерише парове противника за свако коло на основу унапред утврђених критеријума, чува резултате кола и турнира, мења статус турнира у бази активних након што се заврши и води евиденцију о тренутним бодовима сваког играча. Једноставније речено, апликација служи за вођење турнира и његово закључивање.

Осим тога, може бити корисна и као база активних турнира, играча и клубова с обзиром на то да судија све време има могућност претраживања истих и приступ детаљима о њима (не и било какве врсте измене осим додавања нових играча уколико одлуче да се пријаве на турнир, а немају могућност да то ураде преко веб апликације или измене података о турниру уколико дође до потребе за истом у току организације). Да би користио апликацију, судија мора бити пријављен на систем.

Животни циклус софтверског система састоји се од пет подједнако битних фаза:

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



3.1.2 Прикупљање корисничких захтева

У овој фази дефинишу се својства и услови које неки програм мора да испуни.

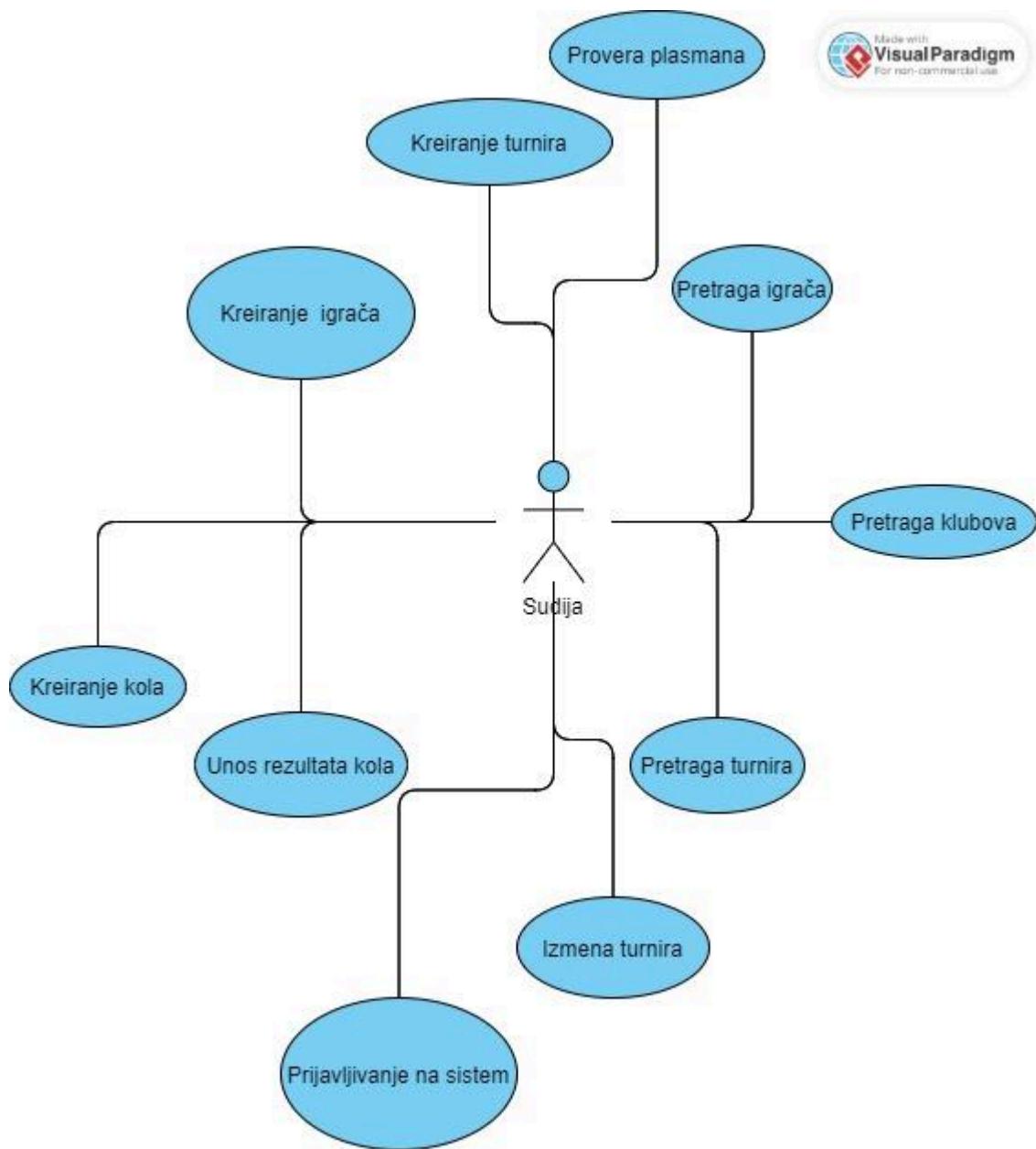
Захтеви се деле на функционалне (дефинишу функције система) и нефункционалне (употребљивост, поузданост, перформансе, подрживост).⁶ [6]

Функционални захтеви представљају случајеве коришћења.

У овој апликацији идентификовани су следећи случајеви коришћења:

1. Пријављивање судије на систем
2. Креирање играча
3. Претраживање клубова
4. Претраживање турнира
5. Претрага играча (Сложен СК)
6. Унос резултата кола
7. Креирање кола (Сложен СК)
8. Креирање турнира
9. Провера пласмана
10. Измена података о турниру (Сложен СК)

⁶ Пројектовање софтвера скрипта, Др Синиша Влајић, Београд 2015.



Слика 11: дијаграм случаја коришћења (цртан на <https://www.visual-paradigm.com/>)

3.1.2.1 СК1: Случај коришћења - Пријављивање судије на систем

Назив СК

Пријављивање судије на систем

Актори СК

Судија

Учесници СК

Судија и систем (програм)

Предуслов: Систем је укључен и приказује форму за пријављивање судије или играча на систем.

Основни сценарио СК

1. **Судија** уноси корисничко име и лозинку. (АПУСО)
2. **Судија** контролише да ли је коректно унео податке за пријављивање. (АНСО)
3. **Судија** позива **систем** да изврши пријављивање. (АПСО)
4. **Систем** проверава унешене податке. (СО)
5. **Систем** приказује **судији** поруку: “успешно сте се пријавили на систем”. (ИА)

Алтернативна сценарија

- 5.1. Уколико **судија** није унео правилне податке, **систем** приказује поруку “пријављивање је неуспешно”. (ИА)

3.1.2.2 СК2: Случај коришћења - Креирање играча

Назив СК

Креирање играча

Актори СК

Судија

Учесници СК

Судија и **систем** (програм)

Предуслов: Систем је укључен и судија је улогован под својом шифром. Систем приказује форму за уношење података о играчу. Учитана је листа свих регистрованих шаковских клубова.

Основни сценарио СК

1. **Судија** **позива** **систем** да креира **играча**. (АПСО)
2. **Систем** **креира** **играча**. (СО)
3. **Систем** **приказује** поруку: “**Систем** је креирао играча“. (ИА)
4. **Судија** **уноси** податке о играчу у недавно креирани ентитет играч. (АПУСО)
5. **Судија** **контролише** да ли је коректно унео податке о играчу. (АНСО)
6. **Судија** **позива** **систем** да запамти податке о **играчу**. (АПСО)
7. **Систем** **памти** податке о **играчу**. (СО)
8. **Систем** **приказује** **судији** запамћеног **играча** и поруку: “**Систем** је запамтио **играча**“. (ИА)

Алтернативна сценарија

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

8.1 Уколико **систем** не може да запамти податке о **играчу**, приказује **судији** поруку “Систем не може да запамти **податке о игрachu**”. (ИА)

3.1.2.3 СК3: Случај коришћења - Претрага клубова

Назив СК

Претраживање клубова

Актори СК

Судија

Учесници СК

Судија и **систем** (програм)

Предуслов: Систем је укључен и **судија** је улогован под својом шифром. Систем приказује форму за рад са **клубовима**.

Основни сценарио СК

1. **Судија** уноси вредност по којој претражује **клубове** . (АПУСО)
2. **Судија** позива **систем** да нађе **клубове** по задатој вредности. (АПСО)
3. **Систем** тражи **клубове** по задатој вредности. (СО)
4. **Систем** приказује **судији** све **клубове** чији се подаци поклапају са задатом вредношћу и поруку: “Систем је нашао **клубове** по задатој вредности”. (ИА)
5. **Судија** бира неки од приказаних клубова. (АПУСО)
6. **Судија** позива систем да учита **клуб**. (АПСО)
7. **Систем** учитава **клуб**. (СО)
8. **Систем** приказује **судији** податке о **играчу** и поруку “Систем је учитао **клуб**.” (ИА)

Алтернативна сценарија

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

8.1 Уколико **систем** не може да учита **клуб**, он показује **судији** поруку “**систем** не може да учита **клуб**”. (ИА)

3.1.2.4 СК4: Случај коришћења - Претраживање турнира

Назив СК

Претраживање турнира

Актори СК

Судија

Учесници СК

Судија и систем (програм)

Предуслов: Систем је укључен и Судија је улогован под својом шифром. Систем приказује форму за рад са турнирима. Учитана је листа играча.

Основни сценарио СК

1. Судија уноси вредност по којој претражује турнире . (АПУСО)
2. Судија позива систем да нађе турнире по задатој вредности. (АПСО)
3. Систем тражи турнире по задатој вредности. (СО)
4. Систем приказује Судији све турнире чији се подаци поклапају са задатом вредношћу и поруку: “Систем је нашао турнире по задатој вредности”. (ИА)
5. Судија бира неки од приказаних турнира. (АПУСО)
6. Судија позива систем да учита турнир. (АПСО)
7. Систем учитава турнир. (СО)
8. Систем приказује судији податке о турниру и поруку “Систем је учитао турнир.” (ИА)

Алтернативна сценарија

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

8.1 Уколико систем не може да учита турнир, он показује судији поруку “систем не може да учита турнир”. (ИА)

3.1.2.5 СК5: Случај коришћења - Претраживање играча

Назив СК

Претраживање играча

Актори СК

Судија

Учесници СК

Судија и систем (програм)

Предуслов: Систем је укључен и судија је улогован под својом шифром. Систем приказује форму за рад са играчима.

Основни сценарио СК

1. Судија уноси вредност по којој претражује играче . (АПУСО)
2. Судија позива систем да нађе играче по задатој вредности. (АПСО)
3. Систем тражи играче по задатој вредности. (СО)
4. Систем приказује судији све играче чији се подаци поклапају са задатом вредношћу и поруку: “Систем је нашао играче по задатој вредности”. (ИА)

Алтернативна сценарија

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

3.1.2.6 СК6: Случај коришћења - Унос резултата кола

Назив СК

Унос резултата партија за једно коло

Актори СК

Судија

Учесници СК

Судија и систем (програм)

Предуслов: Систем је укључен и судија је улогован под својом шифром. Систем приказује форму за рад са колом турнира.

Основни сценарио СК

1. Судија бира одговарајуће коло. (АПУСО)
2. Судија позива систем да учита коло. (АПСО)
3. Систем учитава коло. (СО)
4. Систем приказује судији податке о колу и поруку “Систем је учитао коло”. (ИА)
5. Судија уноси (мења) податке о резултатима сваког парта кола. (АПУСО)
6. Судија контролише да ли је коректно унео податке о колу (результате партија) . (АПСО)
7. Судија позива систем да запамти податке о резултатима кола . (АПСО)
8. Систем памти податке о колу. (СО)
9. Систем приказује запамћене резултате и поруку: “Систем је запамтио резултате кола.” (ИА)

Алтернативна сценарија

4.1. Уколико систем не може да учита коло, он приказује судији поруку “систем не може да пронађе коло по задатој вредности”. Прекида се извршење сценарија. (ИА)

9.1. Уколико систем не може да запамти податке о резултатима кола он приказује судији поруку “Систем не може да запамти резултате кола ”. Прекида се извршење сценарија. (ИА)

3.1.2.7 СК7: Случај коришћења - Креирање кола

Назив СК

Креирање кола на турниру

Актори СК

Судија

Учесници СК

Судија и систем (програм)

Предуслов: Систем је укључен и судија је улогован под својом шифром. Систем приказује форму за рад са колом турнира. Учитани су подаци о турниру.

Основни сценарио СК

1. Судија позива систем да креира парове за то коло . (АПСО)
2. Систем креира парове конкретног кола турнира . (СО)
3. Систем приказује судији парове кола и поруку: “Систем је креирао парове овог кола “. (ИА)

Алтернативна сценарија

3.1 Уколико систем не може да креира парове једног кола он приказује судији поруку: “Систем не може да креира парове за ово коло”. (ИА)

3.1.2.8 СК8: Случај коришћења - Креирање турнира

Назив СК

Креирање турнира

Актори СК

Судија

Учесници СК

Судија и систем (програм)

Предуслов: Систем је укључен и судија је улогован под својом шифром. Систем приказује форму за уношење података о турниру.

Основни сценарио СК

1. Судија позива систем да креира турнир. (АПСО)
2. Систем креира турнир. (СО)

3. Систем приказује поруку: “Систем је креирао турнир“. (ИА)
4. Судија уноси податке о турниру у недавно креирани ентитет турнир. (АПУСО)
5. Судија контролише да ли је коректно унео податке о турниру. (АХСО)
6. Судија позива систем да запамти податке о турниру. (АПСО)
7. Систем памти податке о турниру. (СО)
8. Систем приказује судији запамћени турнир и поруку: “Систем је запамтио турнир“. (ИА)

Алтернативна сценарија

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

8.1 Уколико систем не може да запамти податке о турниру, приказује судији поруку “Систем не може да запамти податке о турниру”. (ИА)

3.1.2.9 СК9: Случај коришћења - Провера пласмана

Назив СК

Провера бодова сваког играча

Актори СК

Судија

Учесници СК

Судија и систем (програм)

Предуслов: Систем је укључен и судија је улогован под својом шифром. Систем приказује форму за рад са партијама једног кола.

Основни сценарио СК

1. Судија позива систем да врати листу до сада остварених поена сваког играча. (АПСО)
2. Систем враћа тражене податке и уписује их на форму. (СО)
3. Систем приказује судији тренутни пласман и поруку: “Систем је учитао поене“. (ИА)

Алтернативна сценарија

3.1 Уколико систем не може да врати резултате он приказује судији поруку: “Систем не може да врати тренутни пласман”. (ИА)

3.1.2.10 CK10: Случај коришћења - Измена података о турниру

Назив СК

Измена података о турниру

Актори СК

Судија

Учесници СК

Судија и систем (програм)

Предуслов: Систем је укључен и судија је улогован под својом шифром. Систем приказује форму за рад са турниром. .

Основни сценарио СК

1. Судија уноси вредност по којој претражује турнире. (АПУСО)
2. Судија позива систем да нађе турнире по задатој вредности. (АПСО)
3. Систем тражи турнире по задатој вредности. (СО)
4. Систем приказује све турнире чији се подаци поклапају са задатом вредношћу и поруку: “Систем је нашао турнире по задатој вредности”. (ИА)
5. Судија бира неки од приказаних турнира. (АПУСО)
6. Судија позива систем да учита турнир. (АПСО)
7. Систем учитава турнир. (СО)
8. Систем приказује судији податке о турниру и поруку: “Систем је учитао турнир”. (ИА)
9. Судија уноси (менја) податке о турниру. (АПУСО)
10. Судија контролише да ли је коректно унео податке о турниру. (АНСО)
11. Судија позива систем да запамти податке о турниру. (АПСО)
12. Систем памти податке о турниру. (СО)
13. Систем приказује судији запамћени турнир и поруку: “Систем је запамтио турнир .” (ИА)

Алтернативна сценарија

- 4.1 Уколико систем не може да нађе турнире, он приказује судији поруку: “Систем не може да нађе турнире по задатој вредности”. Прекида се извршење сценарија. (ИА)

8.1 Уколико **систем** не може да учита **турнир**, он приказује **судији** поруку “**Систем не може да учита турнир**”. Прекида се извршење сценарија.(ИА)

13.1 Уколико систем не може да запамти турнир, приказује судији поруку “Систем не може да запамти турнир”. (ИА)

3.1.3 Фаза анализе

У овој фази описује се логичка структура и понашање софтверског система, па се њен резултат назива *пословна логика софтверског система*. Структура се описује концептуалним и релационим моделом, док су за опис понашања поред системских операција потребни неки од UML дијаграма активности.

Процес генерирања дијаграма Unified Modeling Language (UML) из захтева написаних природним језику (NL) сматра се сложеним и изазовним задатком.

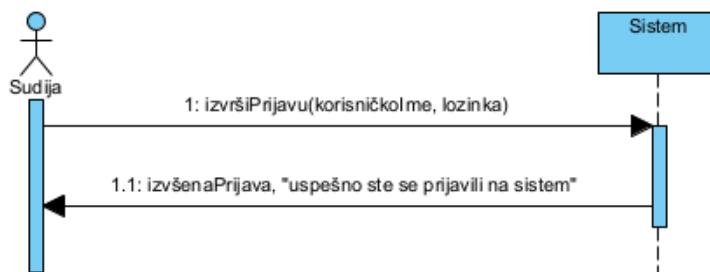
Спецификација софтверских захтева је често написана природним језиком, што изазива потенцијалне проблеме.⁷[7]

Конкретно у овом раду, врста дијаграма активности која је коришћена је дијаграм секвенци. Добра ствар код дијаграма секвенци, која је између осталог и заслужна за толико честу употребу истих, је сличност са изврним програмским кодом, али записана тако да је могу разумети и они који нису програмери. Такође, не зависи од програмског језика.

3.1.3.1 ДС1: Дијаграм секвенце случаја коришћења - пријављивање на систем

Основни сценарио:

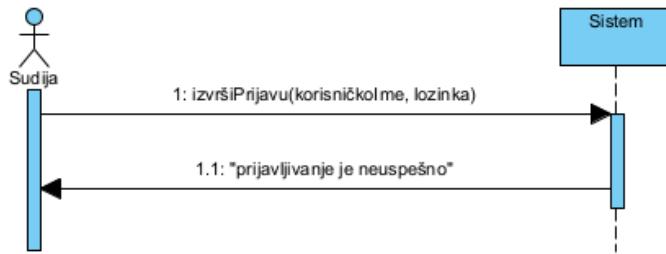
- 1.Судија позива систем да изврши пријављивање. (АПСО)
- 2.Систем приказује судији поруку “успешно сте се пријавили на систем”. (ИА)



Алтернативни сценарио:

- 2.1 Уколико судија/играч није унео правилне податке, систем приказује поруку “пријављивање је неуспешно”. (ИА)

⁷ M. Maatuk, A., & A. Abdelnabi, E. (2021, April). Generating uml use case and activity diagrams using nlp techniques and heuristics rules. In *International Conference on Data Science, E-learning and Information Systems 2021* (pp. 271-277).



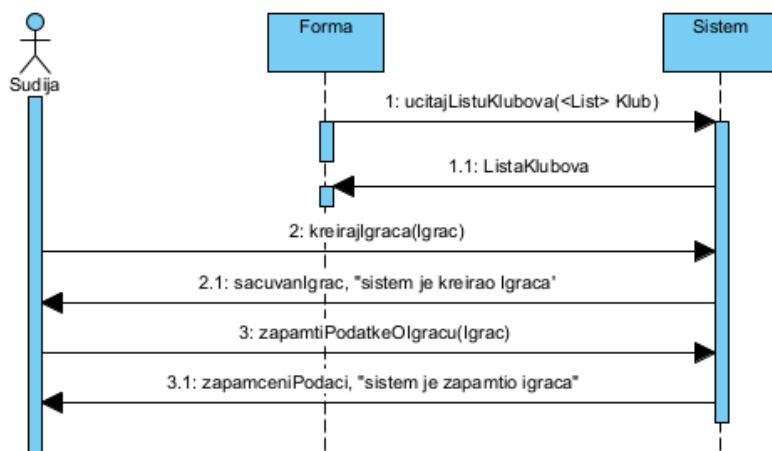
Са наведених секвенцних дијаграма уочавају се системске операције:

1. Signal izvrshiPrijavu(korisnickoIme, lozinka)

3.1.3.2 ДС2: Дијаграм секвенце случаја коришћења - Креирање играча

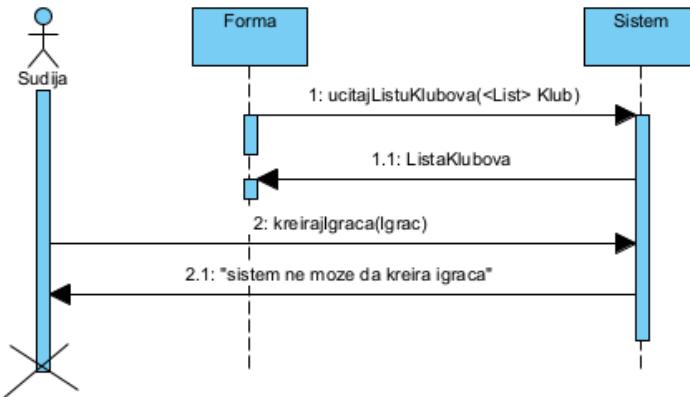
Основни сценарио:

1. Форма позива систем да учита листу шаховских клубова. (АПСО)
2. Систем враћа форми листу клубова. (ИА)
3. Судија позива систем да креира играча. (АПСО)
4. Систем приказује поруку: "Систем је креирао играча". (ИА)
5. Судија позива систем да запамти податке о играчу. (АПСО)
6. Систем приказује судији запамћеног играча и поруку: "Систем је запамтио играча". (ИА)

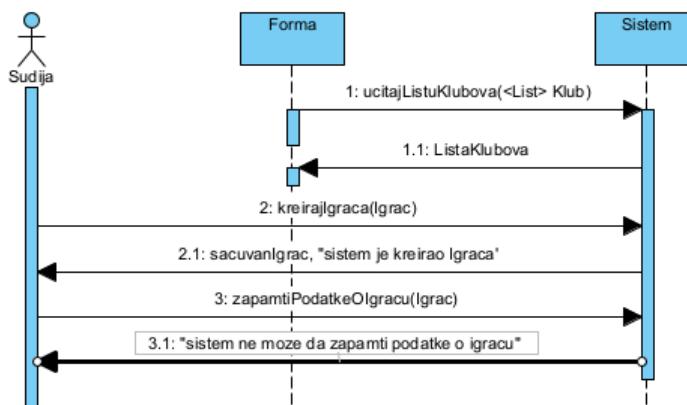


Алтернативни сценарији:

- 4.1 Уколико систем не може да креира играча он приказује судији поруку: "Систем не може да креира играча". Прекида се извршење сценарија. (ИА)



6.1 Уколико систем не може да запамти податке о играчу, приказује судији поруку "Систем не може да запамти податке о играчу". (ИА)



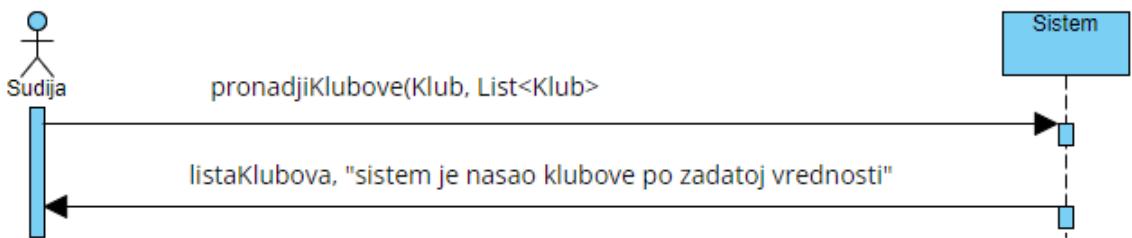
Са наведених секвенцних дијаграма уочавају се системске операције:

1. Signal ucitajListuKlubova(<List> Klub)
2. Signal kreirajIgraca(Igrac)
3. Signal zapamtiPodatkeOligracu(Igrac)

3.1.3.3 ДС3: Дијаграм секвенце случаја коришћења - Претрага клубова

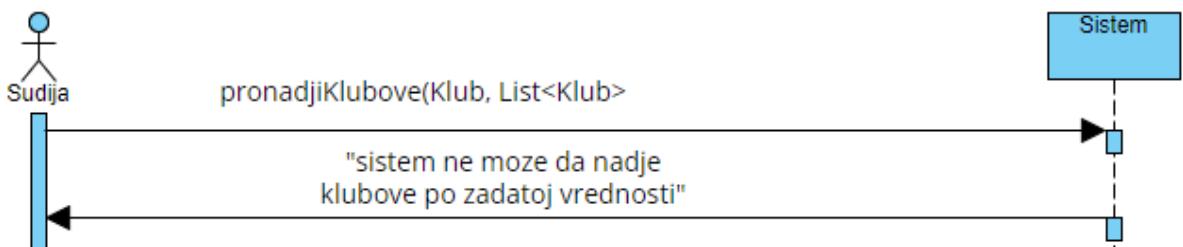
Основни сценарио:

1. Судија позива систем да нађе клубове по задатој вредности. (АПСО)
2. Систем приказује судији све клубове чији се подаци поклапају са задатом вредношћу и поруку: "Систем је нашао клубове по задатој вредности". (ИА)



Алтернативни сценарио:

2.1 Уколико систем не може да нађе клубове он приказује судији поруку: “Систем не може да нађе клубове по задатој вредности”. (ИА)



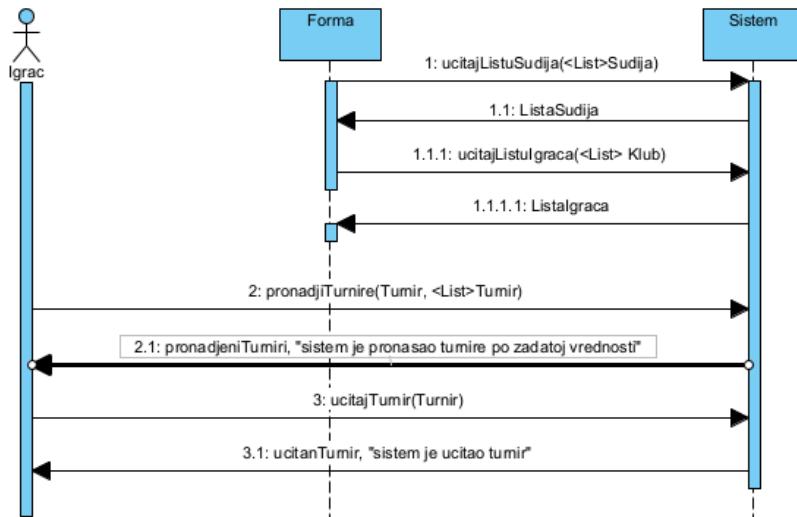
Са наведених секвенцних дијаграма уочавају се системске операције:

1. Signal `pronadjiKlubove(Klub, <List> klub)`

3.1.3.4 ЏС4: Дијаграм секвенце случаја коришћења - Претраживање турнира

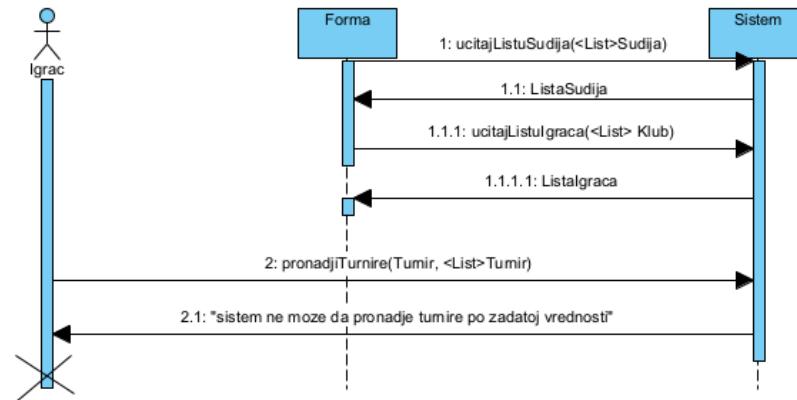
Основни сценарио:

1. Форма позива систем да учита листу играча. (АПСО)
2. Систем враћа форми листу играча. (ИА)
3. Форма позива систем да учита листу судија. (АПСО)
4. Систем враћа форми листу судија. (ИА)
5. Судија позива систем да нађе турнире по задатој вредности. (АПСО)
6. Систем приказује играчу све турнире чији се подаци поклапају са задатом вредношћу и поруку: “Систем је нашао турнире по задатој вредности”. (ИА)
7. Судија позива систем да учита турнир. (АПСО)
8. Систем приказује играчу податке о турниру и поруку “Систем је учитао турнир.” (ИА)

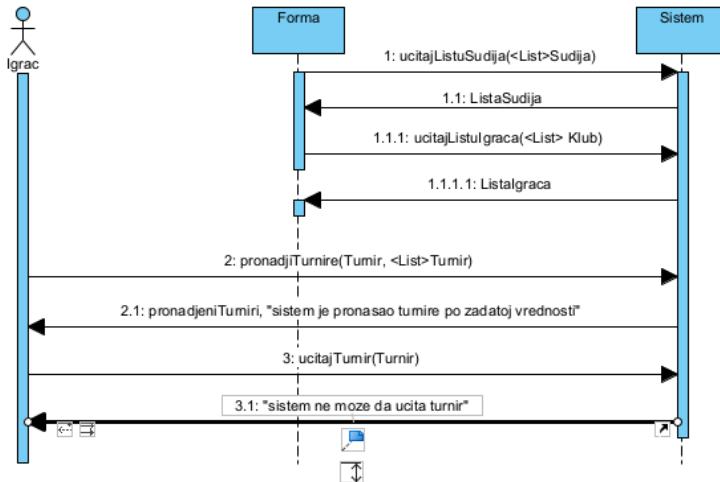


Алтернативни сценарији:

6.1 Уколико систем не може да нађе турнире он приказује играчу поруку: “Систем не може да нађе турнире по задатој вредности”. Прекида се извршење сценарија. (ИА)



8.1 Уколико систем не може да учита турнир, он показује играчу поруку “систем не може да учита турнир”. (ИА)



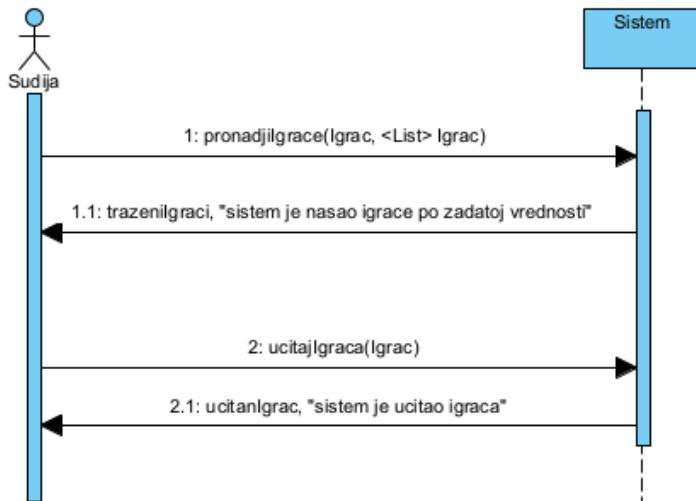
Са наведених секвенцних дијаграма уочавају се системске операције:

1. Signal ucitajListulgraca(<List> Igrac)
2. Signal ucitajListuSudija (<List> Sudija)
3. Signal pronadjiTurnire (Turnir, <List> Turnir)
4. Signal ucitajTurnir (Turnir)

3.1.3.5 DC5: Дијаграм секвенце случаја коришћења - Претраживање играча

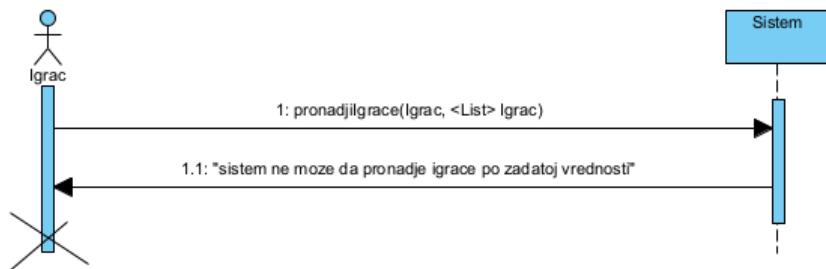
Основни сценарио:

1. Играч позива систем да нађе играче по задатој вредности. (АПСО)
2. Систем приказује играчу све играче чији се подаци поклапају са задатом вредношћу и поруку: "Систем је нашао играче по задатој вредности". (ИА)
3. Играч позива систем да учита играча. (АПСО)
4. Систем приказује играчу податке о играчу и поруку "Систем је учитао играча." (ИА)

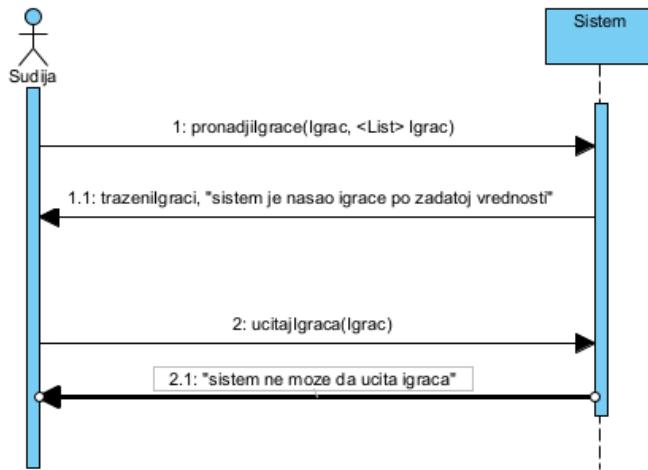


Алтернативни сценарији:

2.1 Уколико систем не може да нађе играче он приказује играчу поруку: “Систем не може да нађе играче по задатој вредности”. Прекида се извршење сценарија. (ИА)



4.1 Уколико систем не може да учита играча, он показује играчу поруку “систем не може да учита играча”. (ИА)



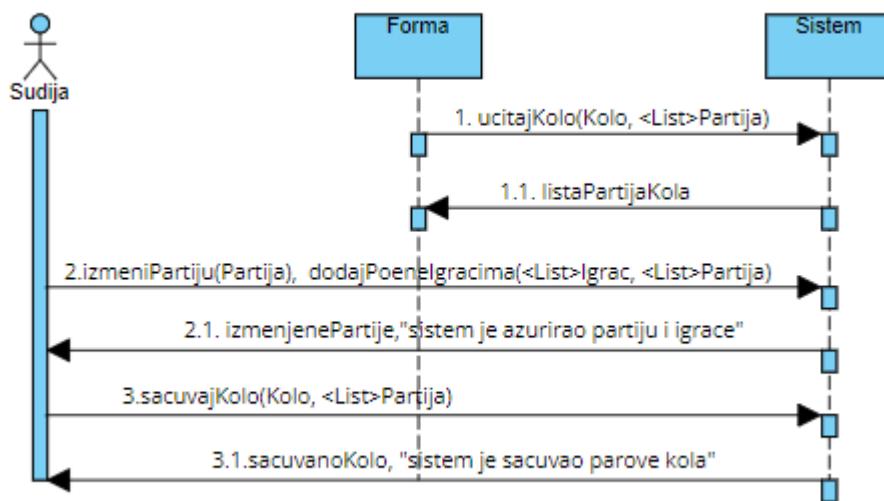
Са наведених секвенцних дијаграма уочавају се системске операције:

2. Signal `pronadjilgrace(Igrac, <List> Igrac)`
3. Signal `ucitajIgraca (Igrac)`

3.1.3.6 DC6: Дијаграм секвенце случаја коришћења - Унос резултата кола

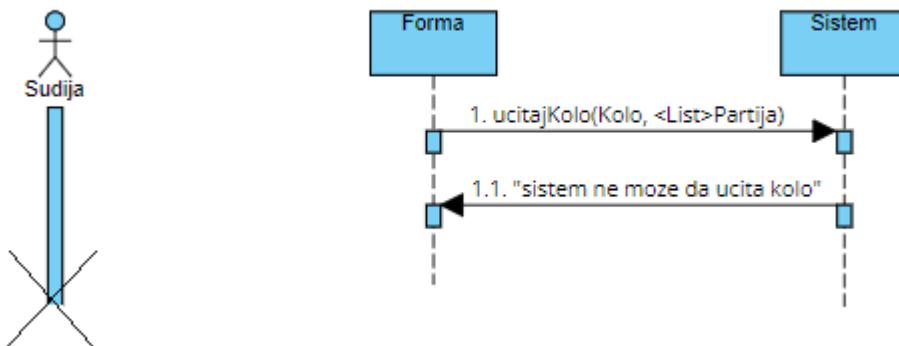
Основни сценарио:

1. Судија позива систем да учита коло. (АПСО)
2. Систем приказује судији податке о колу. (ИА)
3. Судија захтева од система да измене сваку од партија. (АПСО)
4. Систем извршава измену резултата партија у оперативној меморији. (ИА)
5. Судија захтева од система да сачува коло и листу свих одиграних партија у том колу. (АПСО)
6. Систем чува резултате кола и приказује поруку о успешном чувању. (ИА)

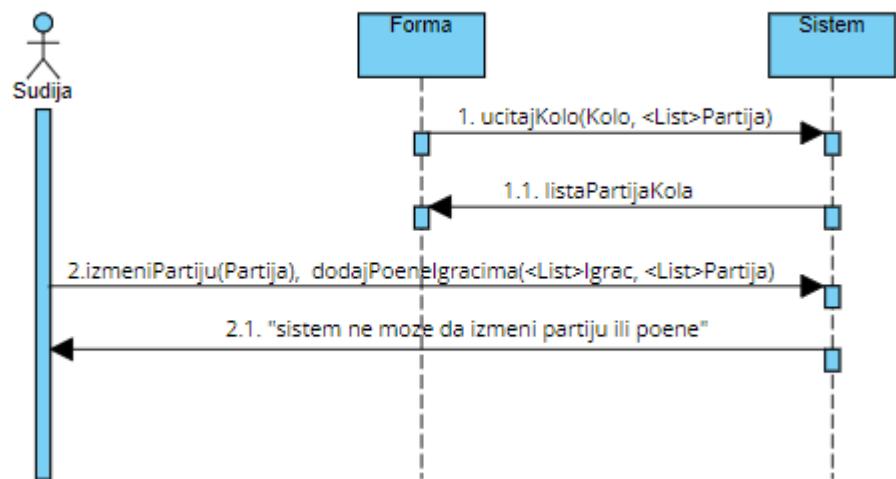


Алтернативни сценарији:

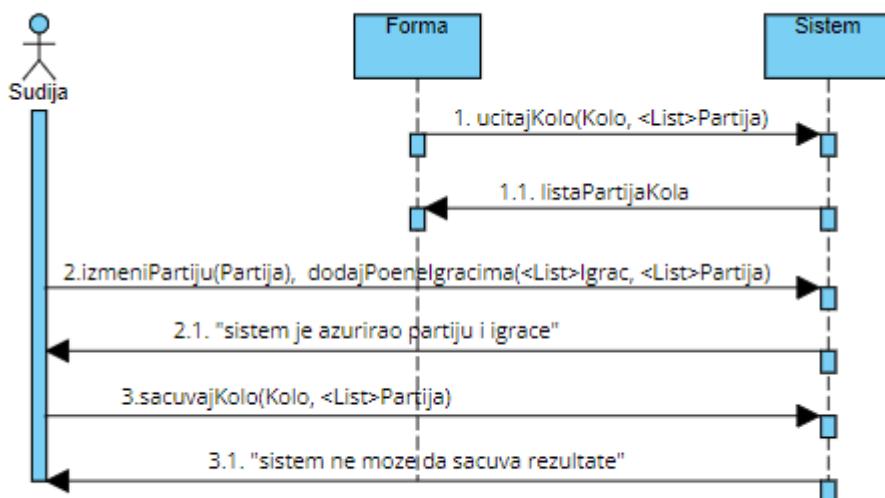
2.1. Уколико систем не може да учита коло, он приказује судији поруку “систем не може да учита партије овог кола”. Прекида се извршење сценарија.



4.1. Уколико систем не може да изменi резултат партије, он приказује судији поруку “систем не може да изменi партију”.



6.1 .Уколико систем не може да сачува партије, он приказује судији поруку “систем не може да сачува резултате”. (ИА)



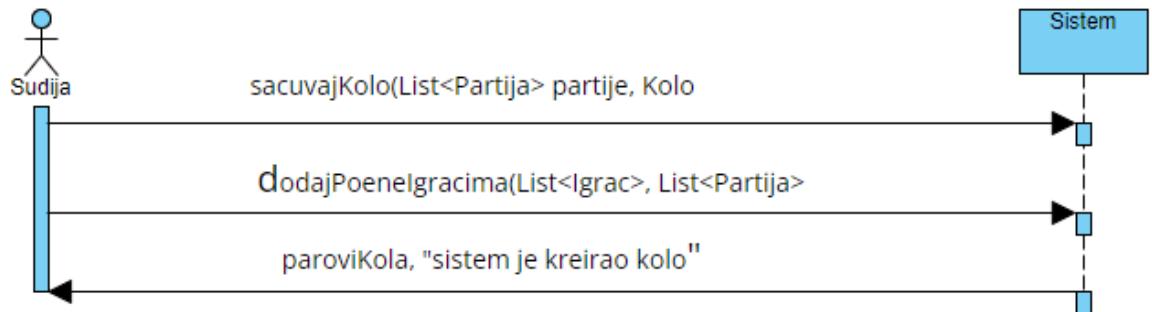
Са наведених секвенцних дијаграма уочавају се системске операције:

1. Signal dodajPoeneIgracima(Igrac, List<Partija>)
2. Signal ucitajKolo(Kolo, <List> Partija)
3. Signal sacuvajKolo(Kolo, <List>Partija)

3.1.3.7 ДС7: Дијаграм секвенце случаја коришћења - Креирање кола

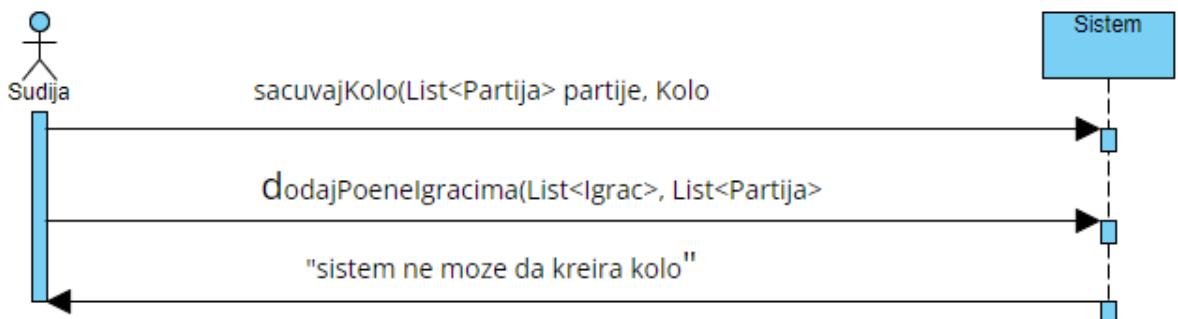
Основни сценарио:

- Судија позива систем да сачува податке о резултатима партија. (АПСО)
- Систем чува партије, коло, ажурира број поена играча и приказује судији парове новог кола и поруку: "Систем је креирао парове овог кола". (ИА)



Алтернативни сценарио:

- Уколико систем не може да креира парове једног кола он приказује судији поруку: "Систем не може да креира парове за ово коло". (ИА)



Са наведених секвенцних дијаграма уочавају се системске операције:

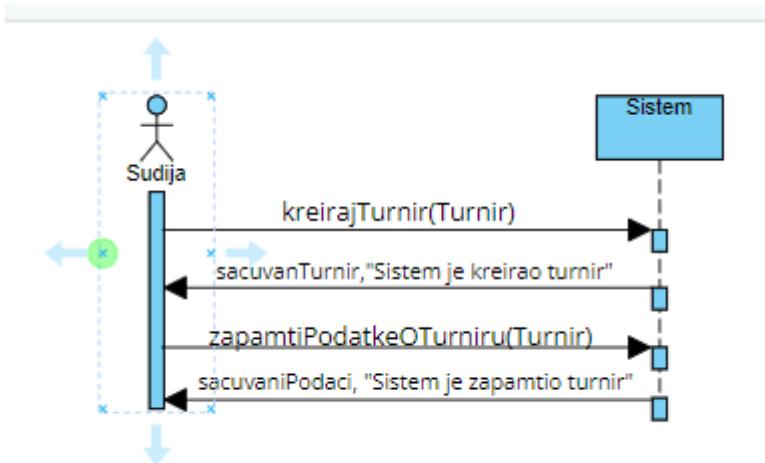
- Signal `dodajPoeneIgracima(List<Igrac>, List<Partija>)`
- Signal `kreirajKolo(Kolo, List<Partija>)`

3.1.3.8 ДС8: Дијаграм секвенце случаја коришћења - Креирање турнира

Основни сценарио:

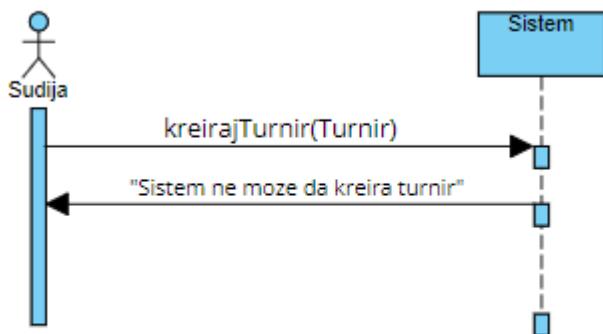
- Судија позива систем да креира турнир. (АПСО)

2. Систем приказује поруку: "Систем је креирао турнир". (ИА)
3. Судија позива систем да запамти податке о турниру. (АПСО)
4. Систем приказује судији запамћени турнир и поруку: "Систем је запамтио турнир". (ИА)

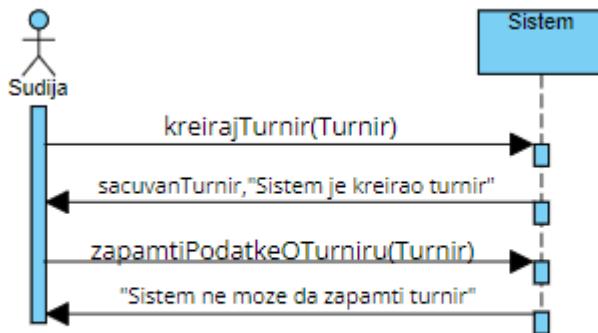


Алтернативни сценарији:

- 2.1 Уколико систем не може да креира турнир он приказује судији поруку: "Систем не може да креира турнир". Прекида се извршење сценарија. (ИА)



- 4.1 Уколико систем не може да запамти турнир он приказује судији поруку: "Систем не може да запамти турнир". (ИА)



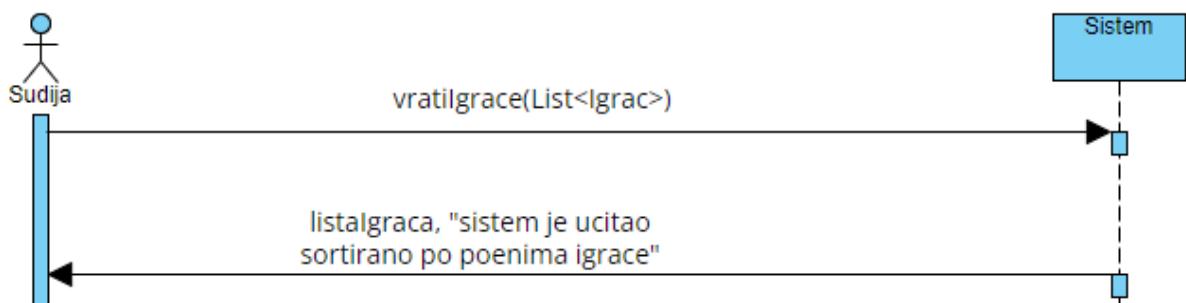
Са наведених секвенцних дијаграма уочавају се системске операције:

1. Signal `ucitajListuIgraca(<List> Igrac)`
2. Signal `pronadjilgrace(Igrac, <List> Igrac)`
3. Signal `obrisiTTurnir(Turnir)`

3.1.3.9 ДС9: Дијаграм секвенце случаја коришћења - Провера пласмана

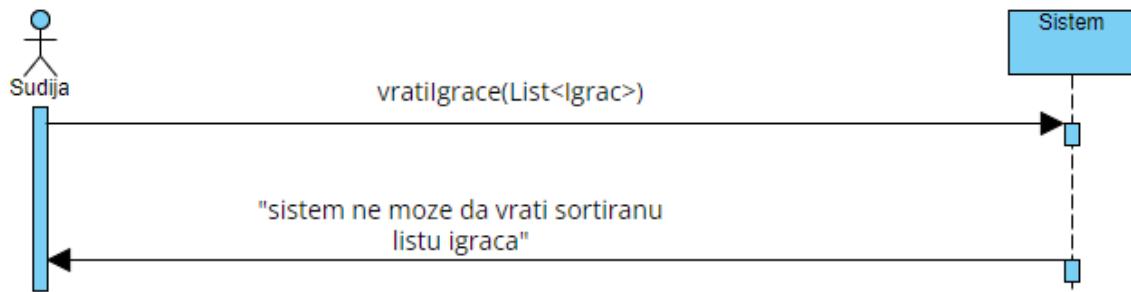
Основни сценарио:

1. судија позива систем да прикаже податке о тренутном пласману (АПСО)
2. Систем приказује судији сортирану листу играча са бројем поена (ИА)



Алтернативни сценарио:

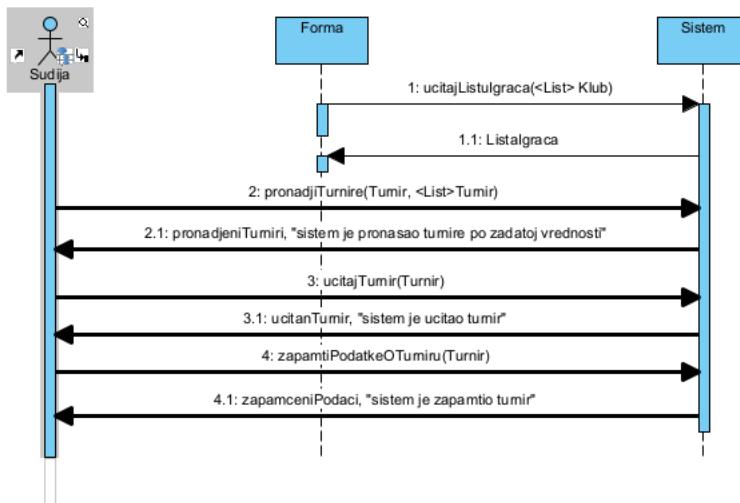
1. Систем приказује судији поруку да не може да прикаже пласман(ИА)



3.1.3.10 DC10: Дијаграм секвенце случаја коришћења - Измена података турниру

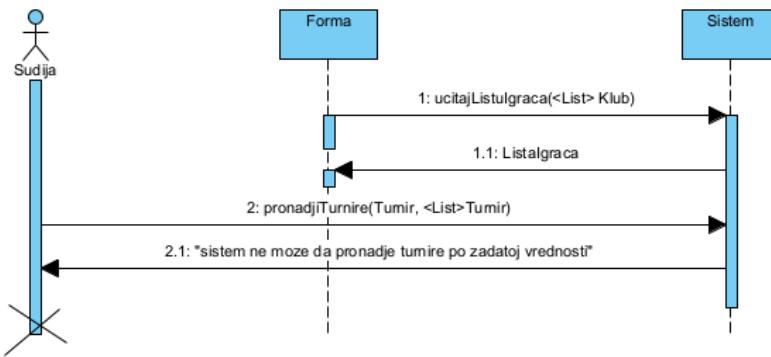
Основни сценарио:

1. Форма позива систем да учита листу турнира. (АПСО)
2. Систем враћа форми листу турнира. (ИА)
3. Судија позива систем да нађе турнире по задатој вредности. (АПСО)
4. Систем приказује турнире и поруку: "Систем је нашао турнире по задатој вредности". (ИА)
5. Судија позива систем да учита турнир. (АПСО)
6. Систем приказује судији податке о турниру и поруку: "Систем је учитао турнир". (ИА)
7. Судија позива систем да запамти податке о турниру. (АПСО)
8. Систем приказује судији запамћени турнир и поруку: "Систем је изменио турнир." (ИА)

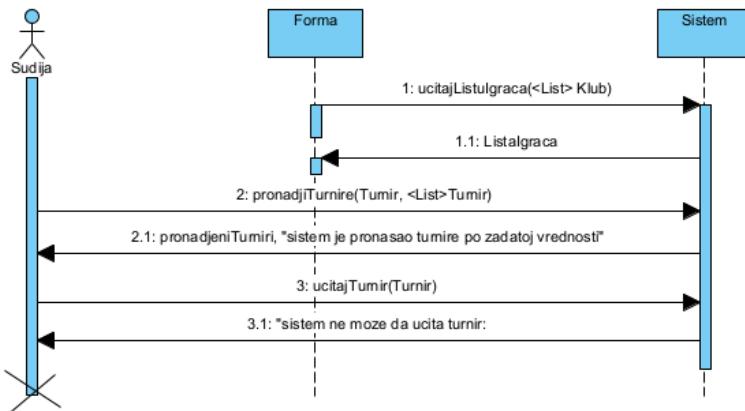


Алтернативни сценарији:

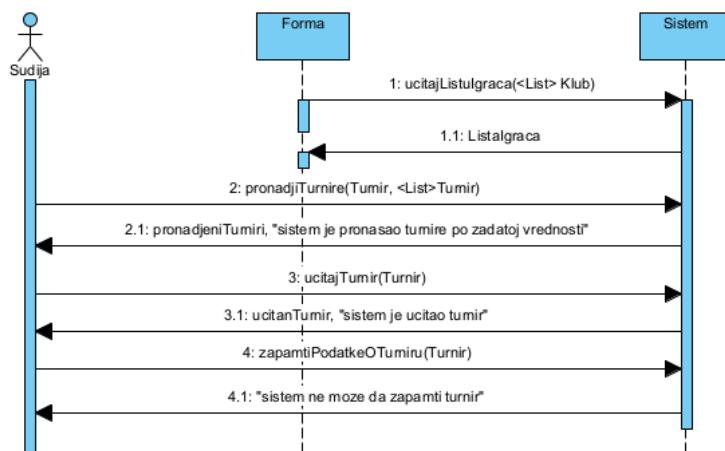
- 4.1 Уколико систем не може да нађе турнир, он приказује судији поруку: "Систем не може да нађе турнир по задатој вредности". Прекида се извршење сценарија. (ИА)



8.1 Уколико систем не може да учита партију, он приказује судији поруку “Систем не може да учита турнир”. Прекида се извршење сценарија. (ИА)



13.1 Уколико систем не може да запамти измену података о турниру, приказује судији поруку “Систем не може да запамти турнир” (ИА)



Са наведених секвенцних дијаграма уочавају се системске операције:

1. Signal `ucitajListuTurnira (<List> Turnir)`
2. Signal `pronadjiTurnire (Turnir, List <Turnir>)`
3. Signal `ucitajTurnir(Turnir)`

4. Signal zapamtiPodatkeOTurniru(Turnir)

На основу анализе сценарија добијено је :

1. Signal izvršiPrijavuNaSistem(korisničkoIme, lozinka)
2. Signal ucitajListuKlubova(<List> Klub)
3. Signal pronadjiKlubove(<List>Klub, Klub>)
4. Signal kreirajIgraca(Igrac)
5. Signal ucitajListulIgraca (<List> Igrac)
6. Signal pronadjiTurnire (<List> Turnir, Turnir)
7. Signal zapamtiPodatkeOTurniru(Turnir)
8. Signal pronadjiIgrace(Igrac, <List> Igrac)
9. Signal ucitajListuTurnira(Turnir, <List> Turnir)
10. Signal kreirajKolo(Kolo)
11. Signal izmeniIgraca(Igrac)
12. Signal obrisiTurnir(Turnir)

3.1.3.11 Понашање софтверског система - дефинисање уговора о системским операцијама

Уговор УГ1: Signal izvršiPrijavuNaSistem(korisničkoIme, lozinka)

Веза са СК: CK1

Предуслови:

Постуслови:

Уговор УГ2:Signal ucitajListuKlubova(<List> Klub)

Веза са СК: CK2, CK3

Предуслови:

Постуслови:

Уговор УГ3:Signal kreirajIgraca(Igrac)

Веза са СК: CK2

Предуслови: Вредносна и структурна ограничења над објектом Igrac морају бити задовољена.

Постуслови: Подаци о играчу су запамћени.

Уговор УГ4:Signal izmeniIgraca(Igrac)

Веза са СК: CK6, CK7

Предуслови: Вредносна и структурна ограничења над објектом Igrac морају бити задовољена.

Постуслови: Подаци о играчу су запамћени.

Уговор УГ5:Signal ucitajListuIgraca (<List> Igrac)

Веза са СК: CK5, CK9

Предуслови:

Постуслови:

Уговор УГ6: signal ucitajListuTurnira(<List> Turnir)

Веза са СК: CK4, CK10

Предуслови:

Постуслови:

Уговор УГ7:Signal pronadjiTurnire (<List> Turnir, Turnir)

Веза са СК: CK4

Предуслови:

Постуслови:

Уговор УГ8:Signal zapamtiPodatkeOTurniru(Turnir)

Веза са СК: CK10

Предуслови: Вредносна и структурна ограничења над објектом Turnir морају бити задовољена.

Постуслови: Подаци о турниру су запамћени.

Уговор УГ9:Signal pronadjiIgrace(Igrac, <List> Igrac)

Веза са СК: CK5

Предуслови:

Постуслови:

Уговор УГ10:Signal pronadjiKlubove(Klub, <List> Klub)

Веза са СК: CK3

Предуслови:

Постуслови:

Уговор УГ11:Signal kreirajKolo(Kolo)

Веза са СК: CK7, CK6

Предуслови: Вредносна и структурна ограничења над објектом Коло морају бити задовољена.

Постуслови: Подаци о колу су запамћени.

Уговор УГ12:Signal kreirajTurnir(Turnir)

Веза са СК: CK8

Предуслови: Вредносна и структурна ограничења над објектом Turnir морају бити задовољена.

Постуслови: Подаци о турниру су запамћени.

Уговор УГ13: Signal ucitajKolo(Kolo, <List>Partija)

Веза са СК: CK6

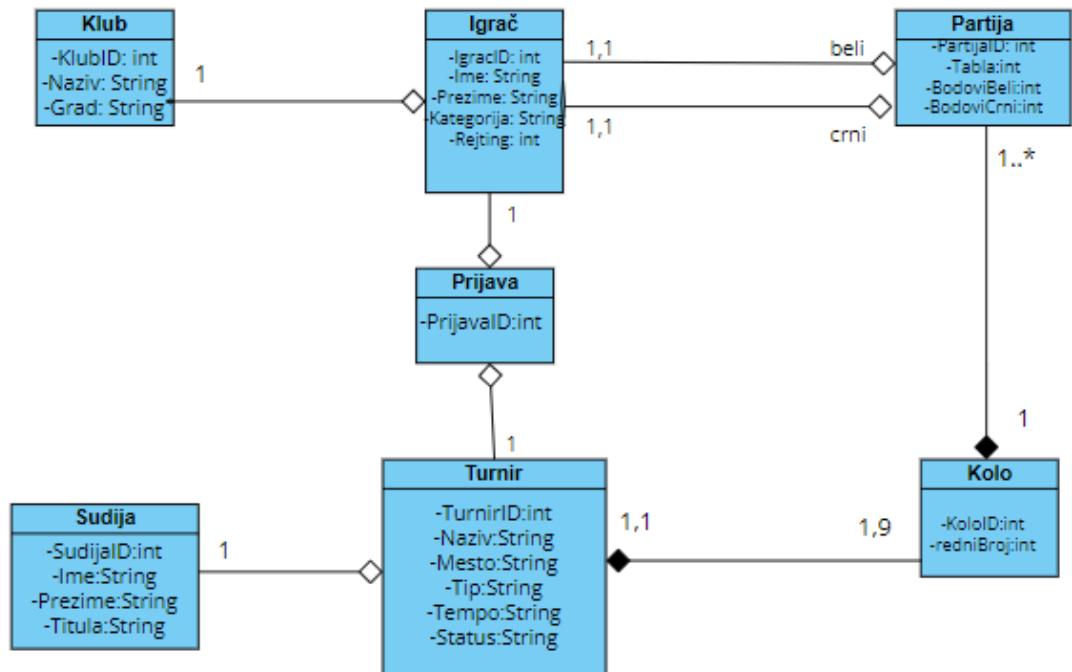
Предуслови:

Постуслови:

3.1.3.12 Структура софтверског система - концептуални (доменски) модел

За приказ концептуалног модела овог софтверског система коришћен је дијаграм класа. Он приказује скуп класа и атрибута сваке од њих (и информације о типу податка) и релације између класа. Технички, то је граф који специфицира логичке и статичке аспекте модела.

Дијаграм класа се не користи само за визуелизацију, описивање и документовање различитих аспеката система, већ и за конструисање извршног кода софтверске апликације.⁸[8]



Слика 12: Концептуални модел (цртан на <https://www.visual-paradigm.com/>)

3.1.3.13 Структура софтверског система - Релациони модел

Klub (klubID, naziv, grad)

Sudija (IDSudije, ime, prezime, titula, prezime, korisnicko, lozinka)

Turnir(IDturnira, IDsudije naziv, mesto, tip, tempo, status)

Igrač(IgračID, klubID, ime, prezime, kategorija, rejting)

Kolo (KoloID, IDturnira, redniBroj)

Partija (partijaID, KoloID, IDTurnira, beli, crni, tabla, bodovibeli, bodovicrni)

Prijava(prijavaID, IDTurira, IDIgraca)

Табела 1: табела структурних и вредносних ограничења класе “Клуб”

Tabela Klub (атрибути)	Просто вредносно ограничење	Слојено вредносно ограничење	Структурно ограничење
---------------------------	-----------------------------	------------------------------	-----------------------

⁸ https://www.tutorialspoint.com/uml/uml_class_diagram.htm

klubID	Int , not null and >0		INSERT / UPDATE CASCADES IGRAC
naziv	String, not null		
grad	String		DELETE RESTRICTED IGRAC

Табела 2: табела структурних и вредносних ограничења класе “Судија”

Tabela Sudija (atributi)	Prosto vrednosno ogranicenje	Slozeno vrednosno ogranicenje	Strukturno ogranicenje
IDsudije	Int , not null and >0		INSERT / UPDATE CASCADES TURNIR
ime	String, not null		
titula	String		DELETE RESTRICTED TURNIR

Табела 3: табела структурних и вредносних ограничења класе “Турнир”

Tabela Turnir (atributi)	Prosto vrednosno ogranicenje	Slozeno vrednosno ogranicenje	Strukturno ogranicenje
IDturnira	Long , not null and >0		INSERT RESTRICTED SUDIJA
naziv	String, not null		UPDATE RESTRICTED SUDIJA
Mesto	String		CASCADES KOLO, PRIJAVA
tip	String, not null		
tempo	String, not null		DELETE RESTRICTED KOLO, PRIJAVA
status	String, not null		

Табела 4: табела структурних и вредносних ограничења класе “Играч”

Tabela Igrac (atributi)	Prosto vrednosno ogranicenje	Slozeno vrednosno ogranicenje	Strukturno ogranicenje
IgracID	Long, not null and >0		INSERT RESTRICTED KLUB
klubID	Integer, not null and >0		UPDATE RESTRICTED KLUB
imeIPrezime	String, not null		CASCADES PARTIJA, PRIJAVA
rejting	Integer		DELETE RESTRICTED PARTIJA, PRIJAVA
kategorija	String		

Табела 5: табела структурних и вредносних ограничења класе “Партија”

Tabela Partija (atributi)	Prosto vrednosno ogranicenje	Slozeno vrednosno ogranicenje	Strukturno ограничење
partijaID	Int , not null and >0		INSERT RESTRICTED IGRAC, KOLO
tabla	Integer		UPDATE RESTRICTED IGRAC, KOLO
otvaranje	String		DELETE /
beli	integer, not null and >0		
crni	integer, not null and >0		
KoloID	Integer, not null and >0		
IDTurnira	Integer , not null and >0		

Табела 6: табела структурних и вредносних ограничења класе “Коло”

Tabela Kolo	Prosto vrednosno	Slozeno vrednosno	Strukturno

(atributi)	ogranicenje	ogranicenje	ogranicenje
redniBrojKola	Integer , not null and $0 < 9$		INSERT RESTRICTED TURNIR
IDTurnira	Long, not null and > 0		UPDATE RESTRICTED TURNIR
KoloID	Int, not null and > 0		CASCADES PARTIJA DELETE RESTRICTED PARTIJA

Табела 7: табела структурних и вредносних ограничења класе “Пријава”

Tabela Prijava	Prosto vrednosno ogranicenje	Slozeno vrednosno ogranicenje	Struktuno ogranicenje
IDIgraca	Integer, not null and > 0		INSERT RESTRICTED IGRAC, TURNIR
IDTurnira	Integer , not null and > 0		UPDATE RESTRICTED IGRAC, TURNIR
IDPrijava	Integer, not null and > 0		DELETE /

Као резултат анализе сценарија СК и прављења концептуалног модела добија се логичка структура и понашање софтверског система. Тиме је завршена фаза анализе чији је крајњи резултат пословна логика система.

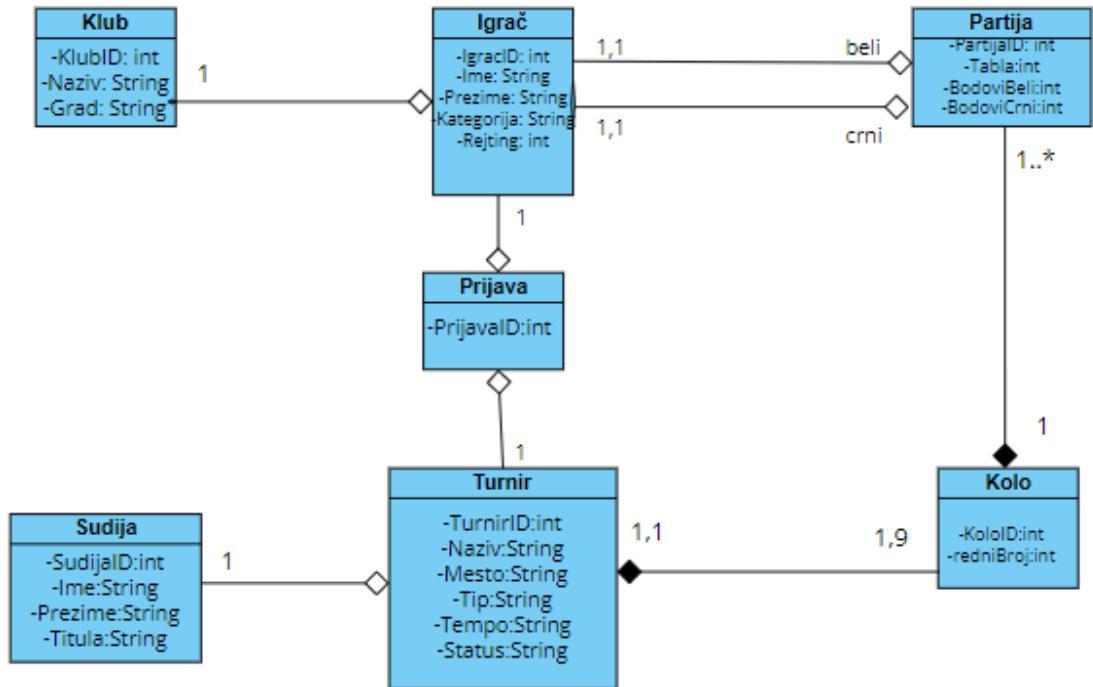
У фази прикупљања захтева елементи структуре и понашања софтверског система се налазе заједно у случајевима коришћења и они су представљени преко именица, глагола и придева. Структура софтверског система (концепти и атрибути концепата), из фазе анализе, се добија на основу именица и придева који су дефинисани у случајевима коришћења. Системске операције, из фазе анализе се добијају на основу глагола који су дефинисани у случајевима коришћења.

Секвенцни дијаграми, из фазе анализе, се добијају на основу сценарија случајева коришћења.⁹ [6]

⁹ Пројектовање софтвера-скрипта, др Синиша Влајић, Београд 2015.

СОФТВЕРСКИ СИСТЕМ

Структура софтверског система:



Понашање:

Системске операције:

- Signal izvršiPrijavuNaSistem(korisničkoIme, lozinka)
- Signal ucitajListuKlubova(<List> Klub)
- Signal pronadjiKlubove(<List> Klub, Klub>)
- Signal kreirajIgraca(Igrac)
- Signal ucitajListuIgraca (<List> Igrac)
- Signal pronadjiTurnire (<List> Turnir, Turnir)
- Signal zapamtiPodatkeOTurniru(Turnir)
- Signal pronadjiIgrace(Igrac, <List> Igrac)
- Signal ucitajListuTurnira(Turnir, <List> Turnir)
- Signal ucitajTurnir (Turnir)
- Signal kreirajKolo(Kolo)
- Signal kreirajTurnir(Turnir)

3.1.4 ФАЗА ПРОЈЕКТОВАЊА

Фаза пројектовања описује физичку структуру и понашање софтверског система (архитектуру софтверског система). У овој фази користимо све до сада добијене ентитете од којих сваки одговара неком од елемената потребних за каснију имплементацију система:

- Складиште података се пројектује на основу релационог модела

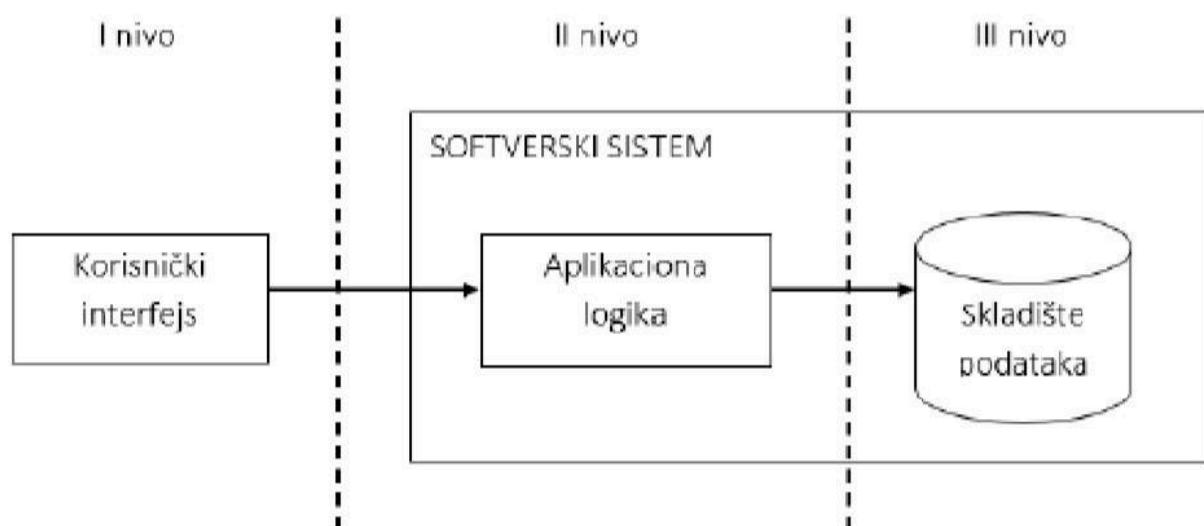
- Пословна логика постаје део апликационе логике
- Кориснички интерфејс дефинише се преко скупа екранских форми
- Сценарија коришћења екранских форми у директној су вези са сценаријима случајева коришћења

3.1.4.1. Архитектура софтверског система

У овом раду коришћена је тронивојска архитектура која се састоји из следећих нивоа:

- Кориснички интерфејс који представља улазно-излазну репрезентацију софтверског система
- Апликационе логике која описује структуру и понашање софтверског система
- Складиште података које чува стање атрибута софтверског система

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



Слика 13: тронивојска архитектура система

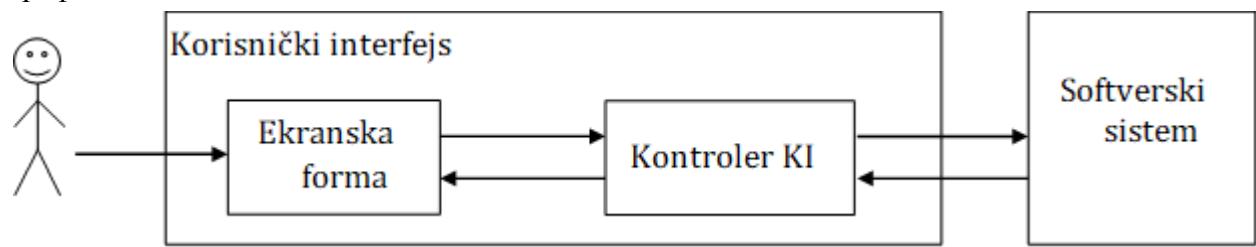
3.1.4.2 Пројектовање корисничког интерфејса

У овом одељку(да не би дошло до забуне због постојања клијентске стране целог овог пројекта) под “клијент” и “корисник” подразумева се судија тј корисник серверске стране ове апликације.

Контролер корисничког интерфејса и екранске форме не представљају софтверски систем, већ реализацију улаза и излаза софтверског система.

Екранска форма има улогу да прихвати податке које корисник уноси, приhvata догађаје које прави корисник, позива контролера корисничког интерфејса како би му проследила податке и приказује податке добијене од контролера корисничког интерфејса. Улога контролера корисничког интерфејса је да прихвати податке које му шаље екранска форма, конвертује податке у објекат који представља улазни аргумент системске операције која ће бити позвана, пошаље захтев за извршење системске операције до софтверског система, приhvата излаз софтверских система који настаје

као резултат извршења системске операције и конвертује објекат у податке графичких елемената.

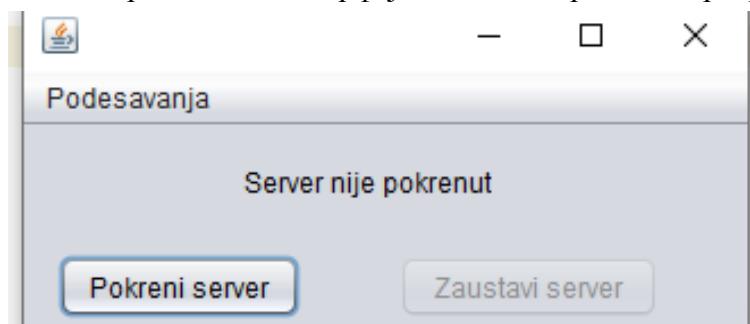


Слика 14. структура корисничкох интерфејса

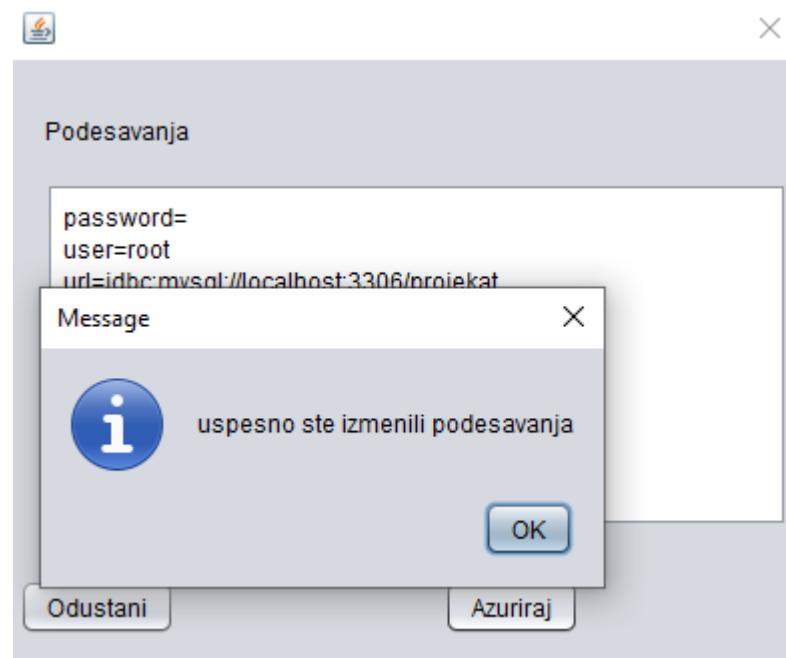
3.1.4.3. Пројектовање екранских форми

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

Изглед корисничког интерфејса након покретања серверског дела апликације:

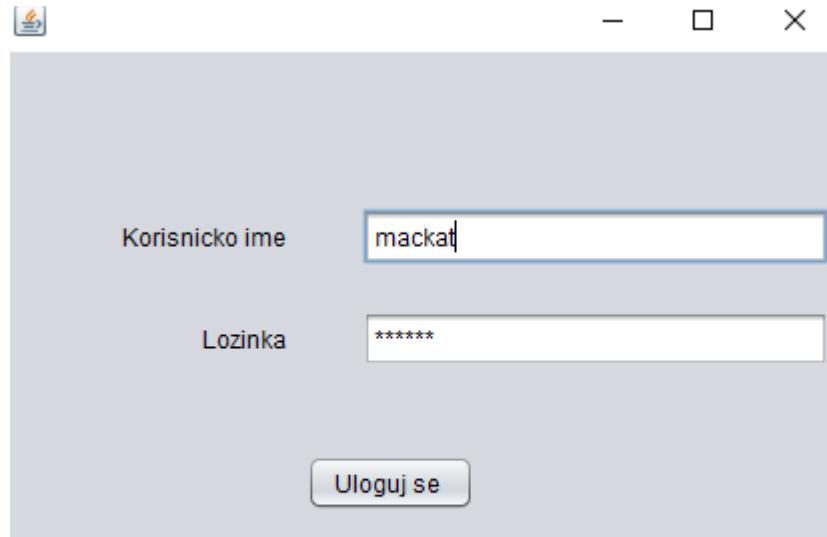


Из менија подешавања могуће је конфигурисати базу података и конекцију:

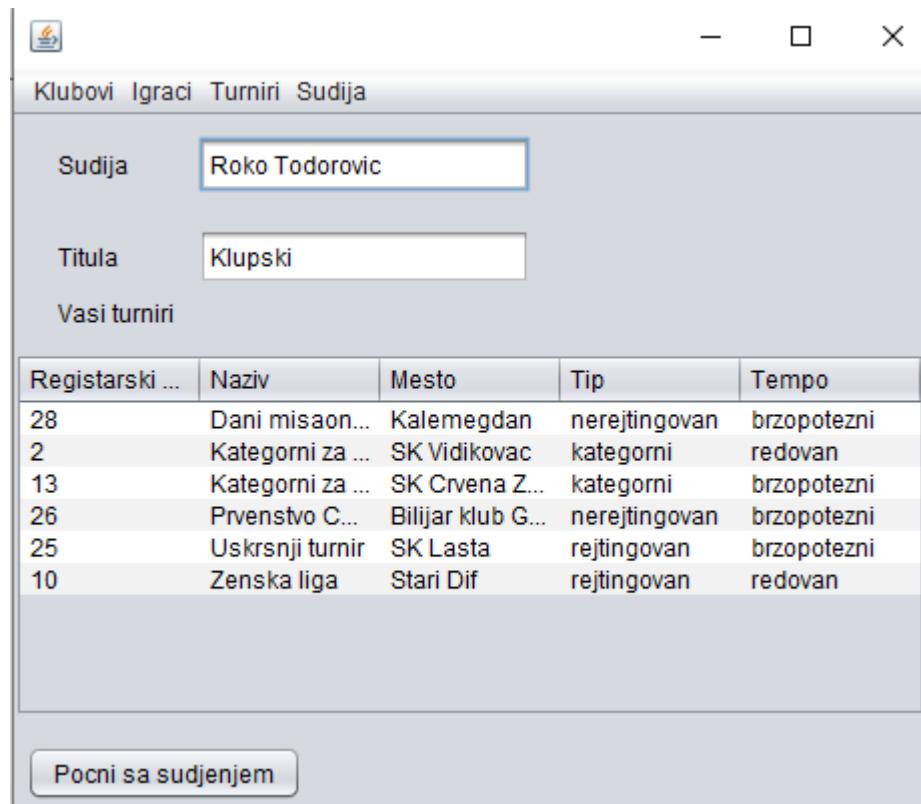


На дугме “покрени сервер”, сервер се покреће и чека судије да се улогују. Дугме “Заустави сервер” затвара серверски сокет. Подешено је да се сервер може више пута покретати и заустављати с тим што није омогућено притискање дугмића у исто време тј у једном тренутку само један од њих је доступан.

Изглед корисничког интерфејса након покретања клијентског дела апликације:



Након што укуца одговарајуће корисничко име и лозинку, клијенту се отвара главна форма клијентског дела која изгледа овако:



На овој форми учитани су само они турнири из базе за чије је суђење задужен тренутно пријављени судија.

Кликом на неки турнир у табели, а затим дугме “почни са суђењем” отвара се форма за рад са конкретним турниром која изгледа овако:

Ulogovani sudija:	Beli Macka				
Aktivan turnir:	Memorijalni turnir Gliga				
Broj odigranih kola: 0 Pogledaj trenutni plasman					
Trenutno aktivno kolo: 1					
Tabla	Beli	Crni	Rezb	Rezc	▼
1	RonnieJam...	Robert Plant	-1.0	-1.0	▲
2	OzzyOsbour...	MylesKenne...	-1.0	-1.0	▼
3	DuffMcKagan	IzzyStradlin	-1.0	-1.0	▼
4	RitchieBlack...	LarsUlrich	-1.0	-1.0	▼
5	JimiHendrix	JoshKizska	-1.0	-1.0	▼
6	DavidGilmour	PhilAnselmo	-1.0	-1.0	▼
7	Slash/	AgnusYoung	-1.0	-1.0	▼
8	StevenAdler	LemmyKilm...	-1.0	-1.0	▼
9	AxlRose	Jon BonJovi	-1.0	-1.0	▼
10	VinceNeil	KirkHammet	-1.0	-1.0	▼

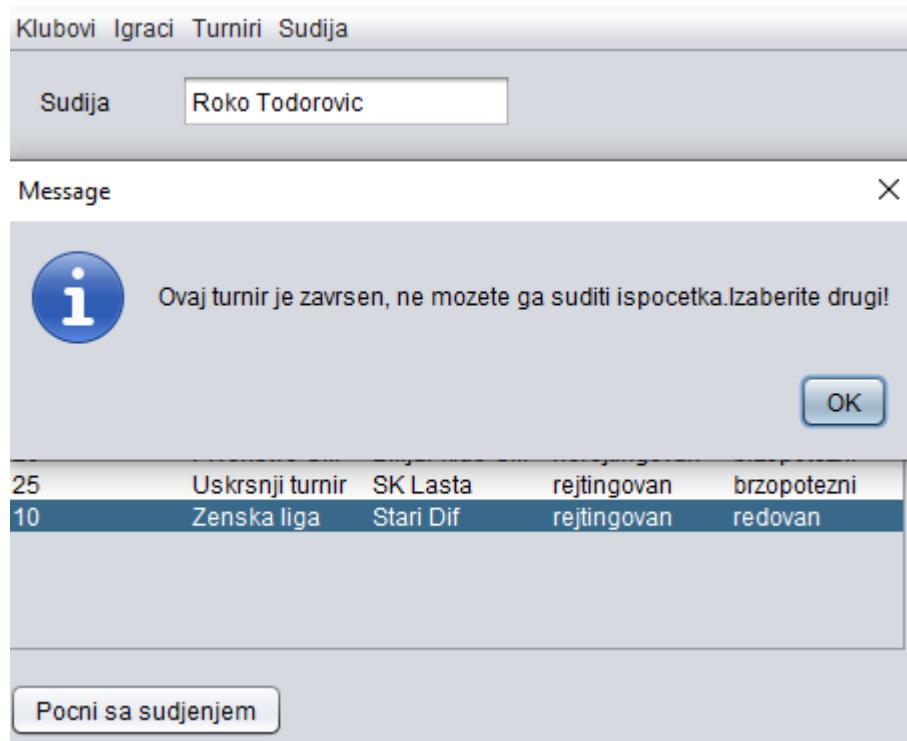
Unesi rezultat odabrane partije

Sacuvaj rezultate kola

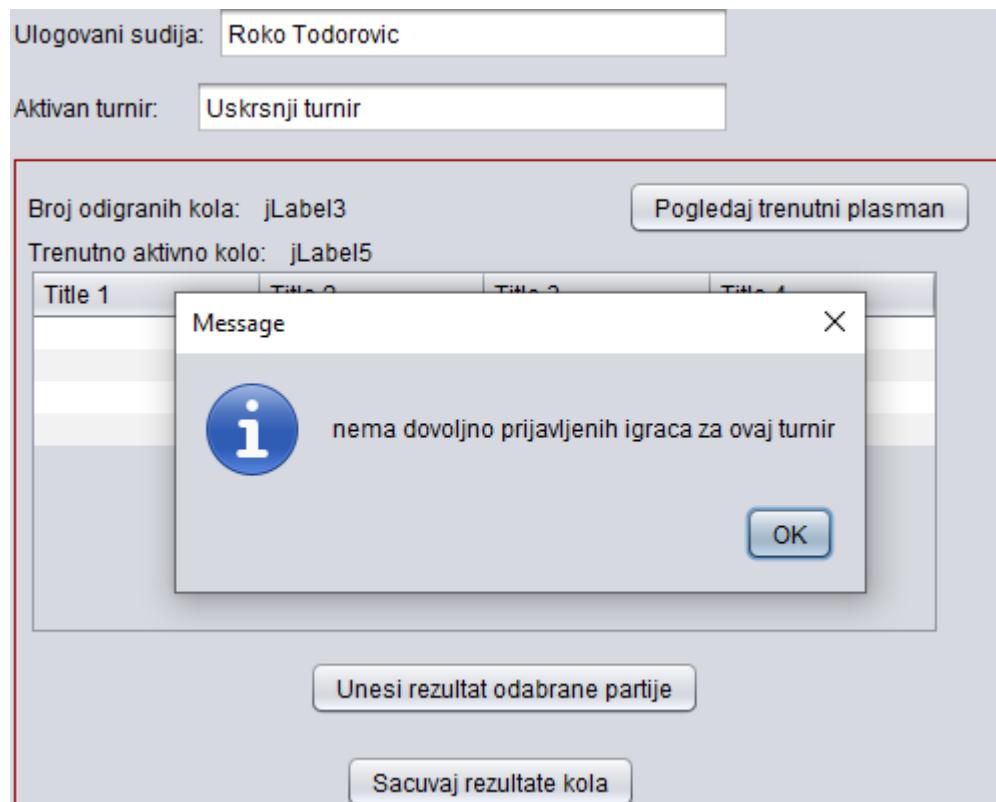
Уоквирени панел представља форму за рад са тренутно активним турниром који суди улоговани судија. Прво коло је аутоматски испаровано од стране система који зна који судија суди турнир јер тај податак постоји у бази. Тако је учитано име судије на форми, турнир који суди и парови првог кола. Резултат партија је постављен на -1 што значи да још није унешен резултат за ту партију, то ће судија радити у наставку сесије.

Изузетак су два случаја:

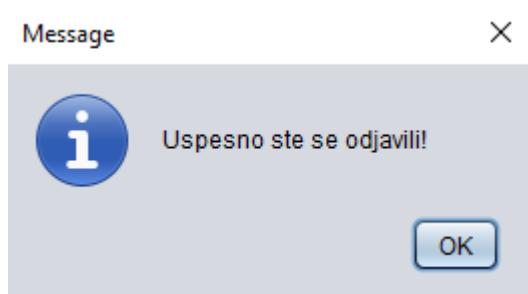
1. Када је одабрани турнир завршен - његов атрибут статус постављен је на "завршен". Тада се приказује следећа порука:



2. Када за одабрани турнир нема довољно пријављених играча (минимум је три):



Кликом на ставку менија судија, судија добија опцију да се одјави. Кликом на опцију “одјави се” судија није више пријављен на систем који га о томе обавештава:



Кроз случајеве коришћења пројектоване су и остале екранске форме које ће апликација поседовати, које се позивају избором неких од датих опција из менија на главној форми или притиском дугмића.

3.1.4.4 CK1: Случај коришћења - Пријављивање судије на систем

Назив СК : Логовање судије

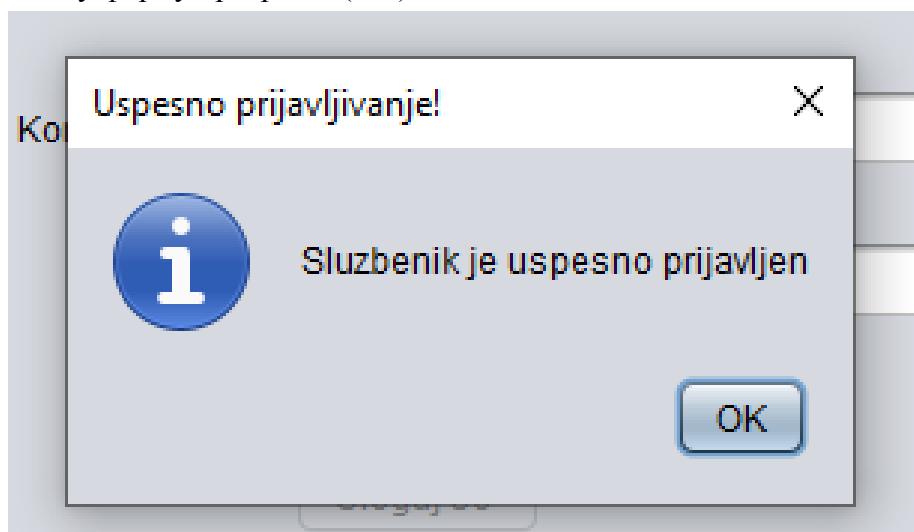
Актори СК : Судија

Учесници СК : Судија и систем (програм)

Предуслов: Систем је покренут и отворена је форма за пријављивање.

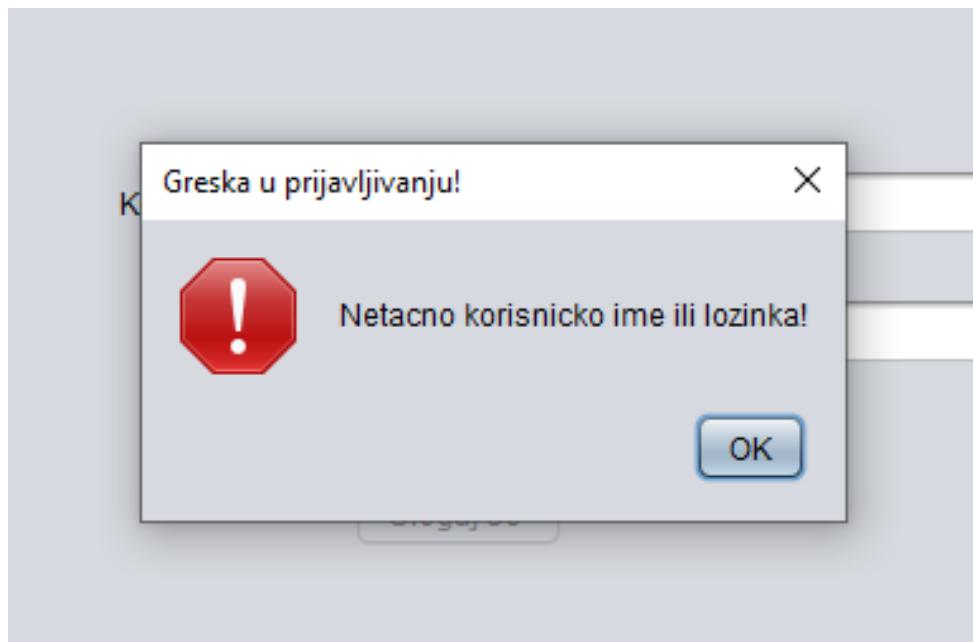
Основни сценарио:

1. Судија уноси корисничко име и лозинку (АПУСО)
2. Кликом на дугме Улогуј се судија позива систем да изврши пријављивање. (АНСО)
3. Систем проверава унешене податке (СО)
4. Систем приказује судији поруку да је успешно пријављен на систем и отвара главну форму програма. (ИА)



Алтернативни сценарио:

- 4.1 Систем приказује судији поруку да није унео одговарајуће параметре.



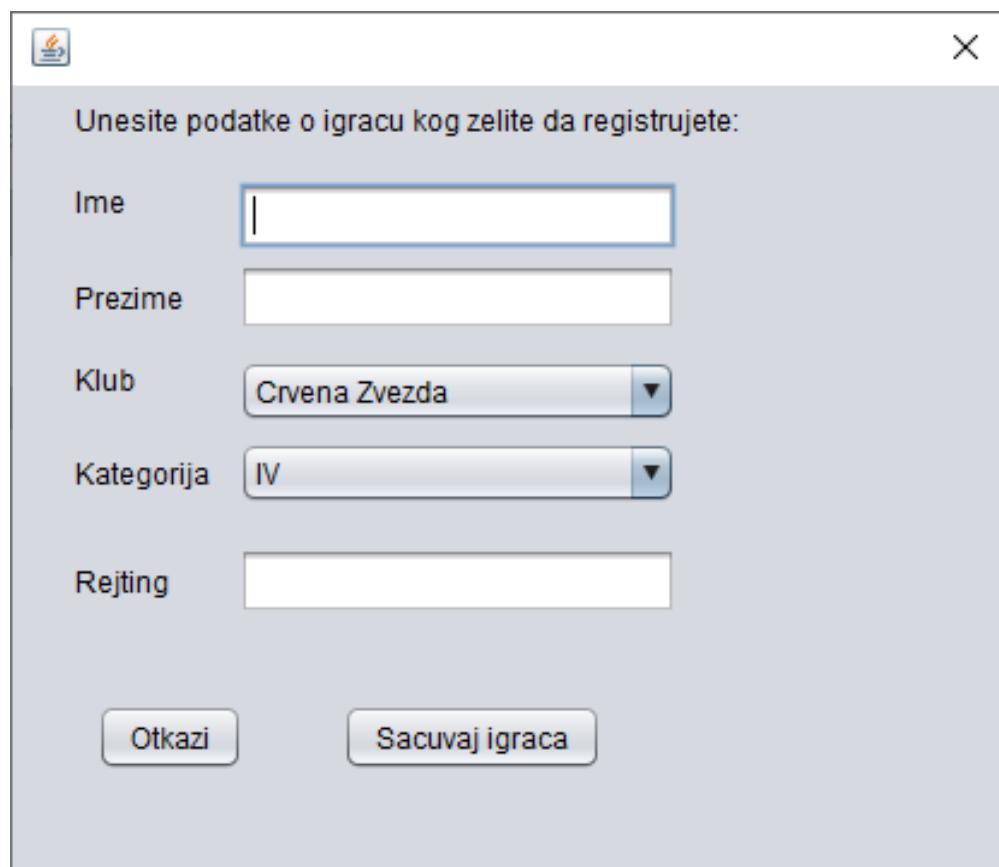
3.1.4.5 CK2: Случај коришћења - Креирање играча

Назив СК : Креирање играча

Актори СК : Судија

Учесници СК : Судија и систем (програм)

Предуслов: Систем је укључен и судија је улогован под својом шифром. Систем приказује форму за уношење података о играчу. Учитана је листа свих регистрованих шаковских клубова.



Unesite podatke o igracu kog zelite da registrujete:

Ime

Prezime

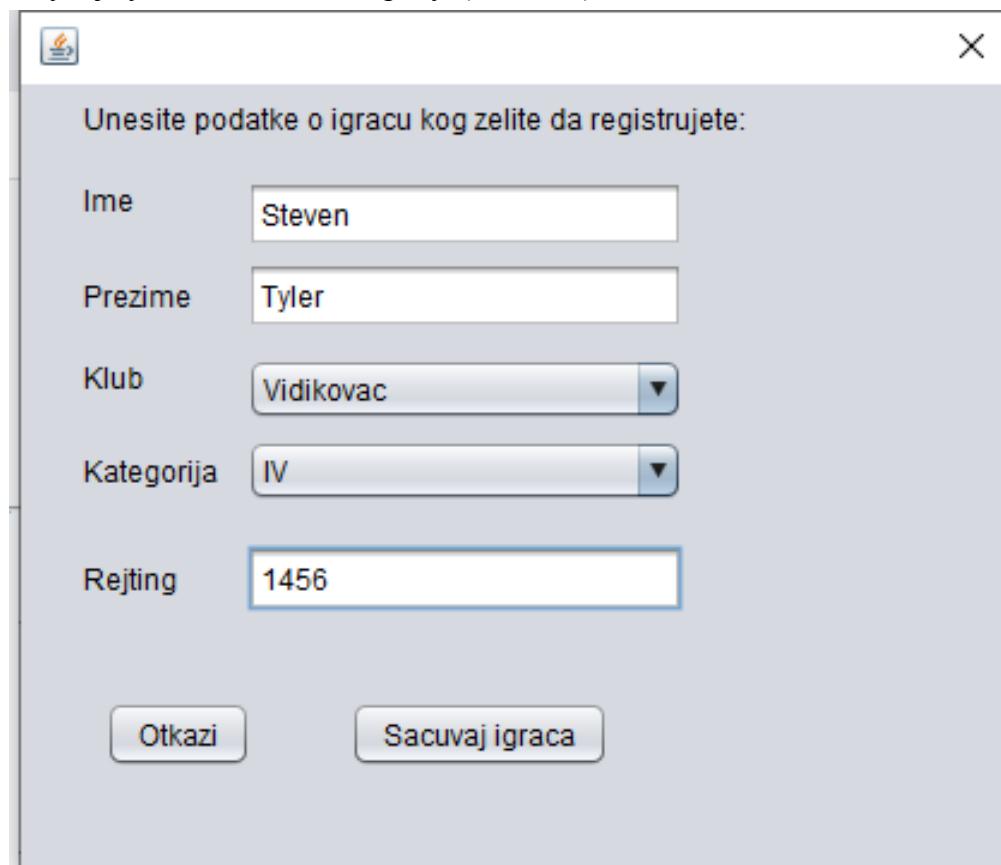
Klub

Kategorija

Rejting

Основни сценарио:

- Судија уноси податке о играчу. (АПУСО)



Unesite podatke o igracu kog zelite da registrujete:

Ime

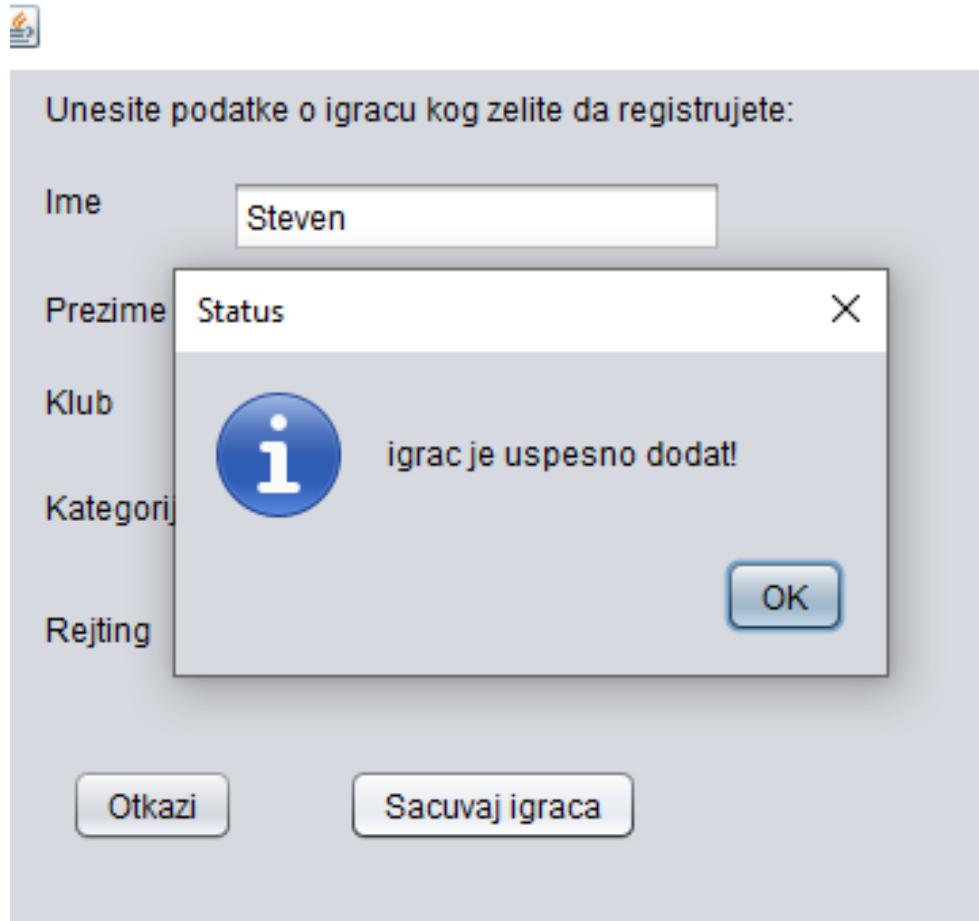
Prezime

Klub

Kategorija

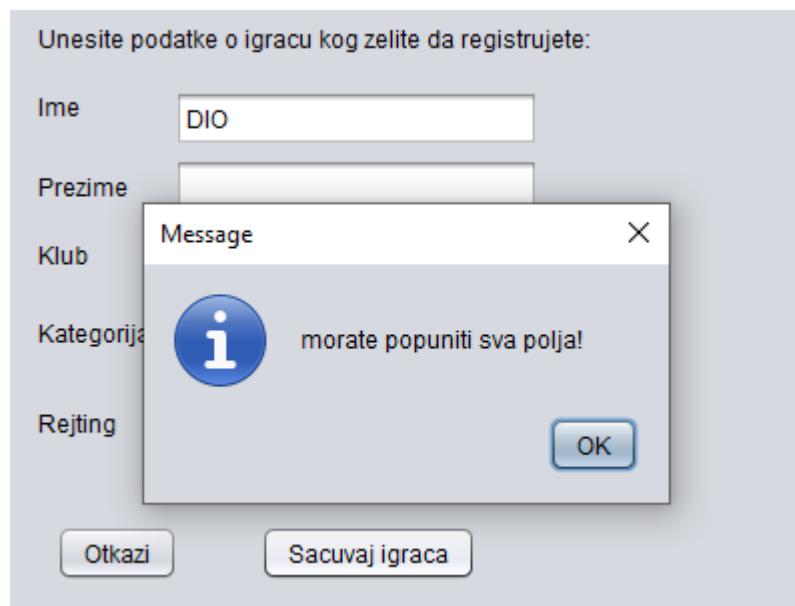
Rejting

2. Судија контролише да ли је коректно унео податке (AHCO)
3. Судија позива систем да запамти податке о играчу (APCO)
4. Систем креира новог играча и памти податке о њему (CO)
5. Систем приказује судији поруку да је запамтио играча (IA)



Алтернативни сценарио:

- 5.1. Уколико систем не може да запамти податке о новом играчу он приказује поруку грешке судији.



(На пример, у случају да се не унесу подаци у сва поља).

3.1.4.6 СК3: Случај коришћења - Претрага клубова

Назив СК : Претрага клубова

Актори СК : Судија

Учесници СК : Судија и систем (програм)

Предуслов: Систем је укључен и судија је улогован под својом шифром. Систем приказује форму за претрагу клубова.

Основни сценарио СК:

1. Судија уноси вредност по којој претражује клубове (АПУСО)



Odaberite kriterijum za pretragu:

Naziv kluba
 Grad
 Prikazi sve

Unesite vrednost kriterijuma:

Pretrazi

Registarski broj	Naziv	Grad

2. Судија позива систем да нађе клуб по задатој вредности (АПСО)
3. Систем тражи клубове по задатој вредности (СО)
4. Систем враћа клубове по задатој вредности (ИА)



Odaberite kriterijum za pretragu:

- Naziv kluba
 Grad
 Prikazi sve

Unesite vrednost kriterijuma:

Pretrazi

Registarski broj	Naziv	Grad
5	Crvena Zvezda	Beograd
2	Kralj	Beograd
6	Partizan	Beograd
1	Vidikovac	Beograd

При промени тренутно активног radio-button мења се и критеријум претраге:

Odaberite kriterijum za pretragu:

- Naziv kluba
 Grad
 Prikazi sve

Unesite vrednost kriterijuma:

Pretrazi

Registarski broj	Naziv	Grad
5	Crvena Zvezda	Beograd
3	Cukaricki	Novi Sad
7	Delfin	Bar
12	Kapablanka	Niksic
2	Kralj	Beograd
4	Lasta	Bar
9	Mala Rokada	Uzice
11	Nestor	Podgorica
10	Nikola Tesla	Uzice
6	Partizan	Beograd
8	Tadic	Nis
1	Vidikovac	Beograd

Алтернативни сценарио:

4.1. Уколико не може да изврши претрагу, систем шаље судији поруку уз објашњење зашто не може (ИА)

Odaberite kriterijum za pretragu:

Naziv kluba
 Grad
 Prikazi sve

Unesite vrednost kriterijuma:

Message ×

morate izabrati kriterijum pretrage

Registarski broj	Naziv

3.1.4.7 СК4: Случај коришћења - Претрага турнира

Назив СК : Претрага турнира

Актори СК : Судија

Учесници СК : Судија и систем (програм)

Предуслов: Систем је укључен и судија је улогован под својом шифром. Систем приказује форму за претрагу турнира.

Odaberite i unesite kriterijum pretrage:

Naziv
 Tip
 Tempo
 Region
 Prikazi sve

Registar...	Naziv	Mesto	Tip	Tempo	Region

Основни сценарио СК:

1. Судија уноси вредност по којој хоће да претражи турнире (АПУСО)

Odaberite i unesite kriterijum pretrage:

Naziv

Tip

Tempo

Region

Prikazi sve

Registar...	Naziv	Mesto	Tip	Tempo	Region
2	Kategor...	SK Vidik...	kategorini	redovan	drzavno
3	Letnji O...	Restora...	rejtingov...	redovan	drzavno
8	Memorij...	Gimnazi...	rejtingov...	redovan	svetsko
11	Prva liga...	Stadion ...	rejtingov...	redovan	drzavno
1	Prvenstv...	Matemat...	nerejtin...	redovan	balkans...
4	Prvenstv...	Hotel Fo...	nerejtin...	redovan	opstinsko
29	Svetski ...	Beograd...	rejtingov...	redovan	svetsko
10	Zenska l...	Stari Dif	rejtingov...	redovan	svetsko

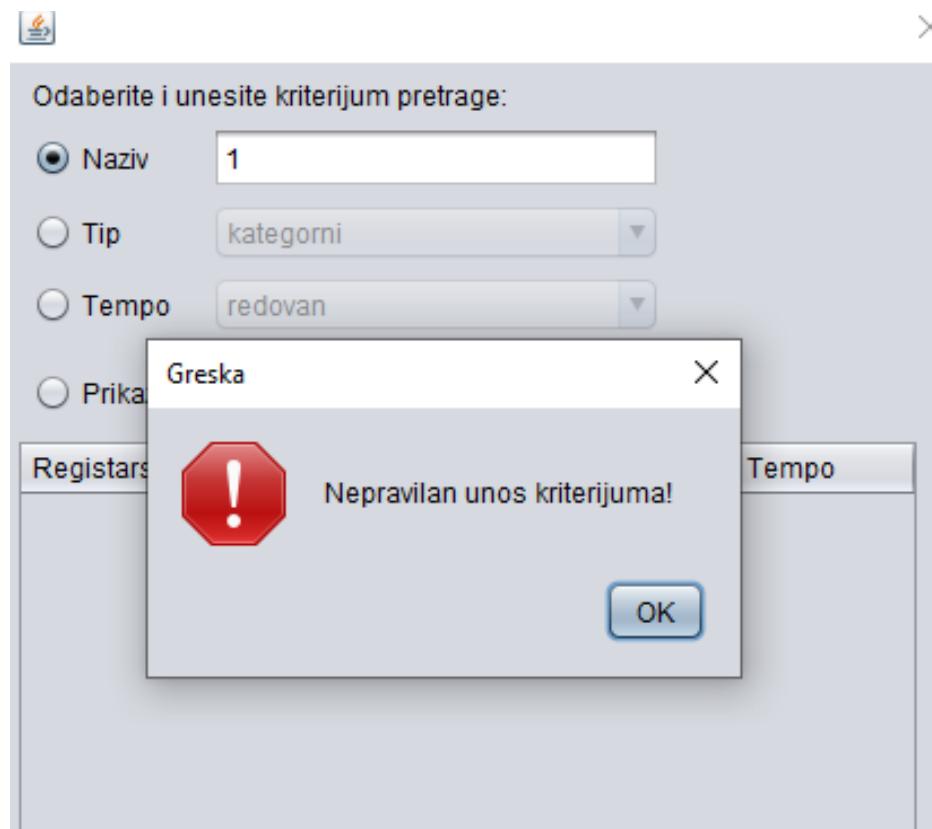
2. Судија позива систем да нађе турнир по задатој вредности (АПСО)

3. Систем тражи турнире по задатој вредности (СО)

4. Систем враћа судији турнире по задатој вредности (ИА)

Алтернативни сценарио:

- 4.1. Систем враћа судији поруку да не може да врати турнире по задатој вредности уколико је то случај (ИА)



На пример, када су подаци за неки од атрибута унешени у неодговарајућем формату.

3.1.4.8 CK5: Случај коришћења - Претрага играча

Назив СК : Претрага играча

Актори СК : Судија

Учесници СК : Судија и систем (програм)

Предуслов: Систем је укључен и судија је улогован под својом шифром. Систем приказује форму за претрагу играча. Учитана је листа клубова.

Основни сценарио СК:

1. Судија уноси вредност по којој хоће да претражи играче (АПУСО)
2. Судија позива систем да нађе играче по задатој вредности (АПСО)
3. Систем тражи играче по задатој вредности (СО)
4. Систем враћа играче по задатој вредности (ИА)

Odaberite kriterijum za pretragu i unesite/odaberite vrednost:

<input type="radio"/> klub	Crvena Zvezda
<input checked="" type="radio"/> kategorija	MK
<input type="radio"/> ime	
<input type="radio"/> prezime	
<input type="radio"/> prikazi sve	<input type="button" value="Pretraga"/>

registarski...	klub	ime	prezime	kategorija	rejting
11	Kralj	Ozzy	Osbourne	MK	1900
6	Cukaricki	Lars	Ulrich	MK	2200
14	Partizan	Robert	Plant	MK	1700
27	Crvena Zv...	Jon	BonJovi	MK	1890
1	Vidikovac	Axl	Rose	MK	1987

Алтернативни сценарио:

4.1. Уколико не може да пронађе играче по задатој вредности, систем враћа поруку судији (ИА)

Odaberite kriterijum za pretragu i unesite/odaberite vrednost:

<input type="radio"/> klub	Crvena Zvezda
<input type="radio"/> kategorija	IV
<input checked="" type="radio"/> ime	
<input type="radio"/> prezime	
<input type="radio"/> prikazi sve	<input type="button" value="Pretraga"/>

klub	ime	Message	rejting
Nikola Tesla	Agnu	 nije unet kriterijum pretrageae	56
Kapablanka	Amy		57
Vidikovac	Axl		70
Mala Rokada	Davi		70
Nestor	Davi		70
Kralj	Davi		70
Cukaricki	Duff		70
Crvena Zvezda	Ian		1676
Lasta	Izzy		

На пример, ако је одабрано по ком критеријуму ће да се врши претрага, а вредност критеријума није унешена.

3.1.4.9 СК6: Случај коришћења - Унос резултата кола

Назив СК : Унос резултата кола

Актори СК : Судија

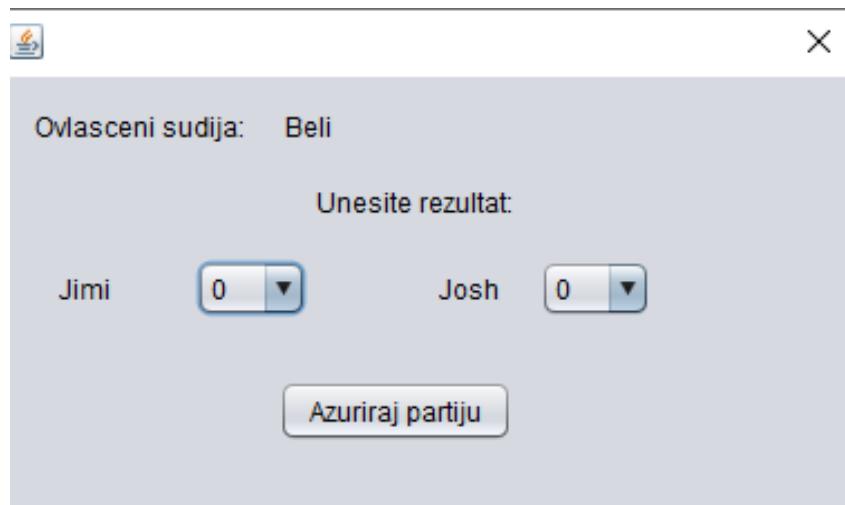
Учесници СК : Судија и систем (програм)

Предуслов: Систем је укључен и судија је улогован под својом шифром. Систем приказује главну форму која се отвара при одабиру турнира са почетне форме која се отвара при логовању и садржи панел за рад са колом турнира. Учитани су подаци о броју одиграних кола, редном броју тренутно активних кола, турниру и судији. Прво коло је аутоматски испаровано па је учитана и табела партија тог кола чији резултати тек треба да се унесу.

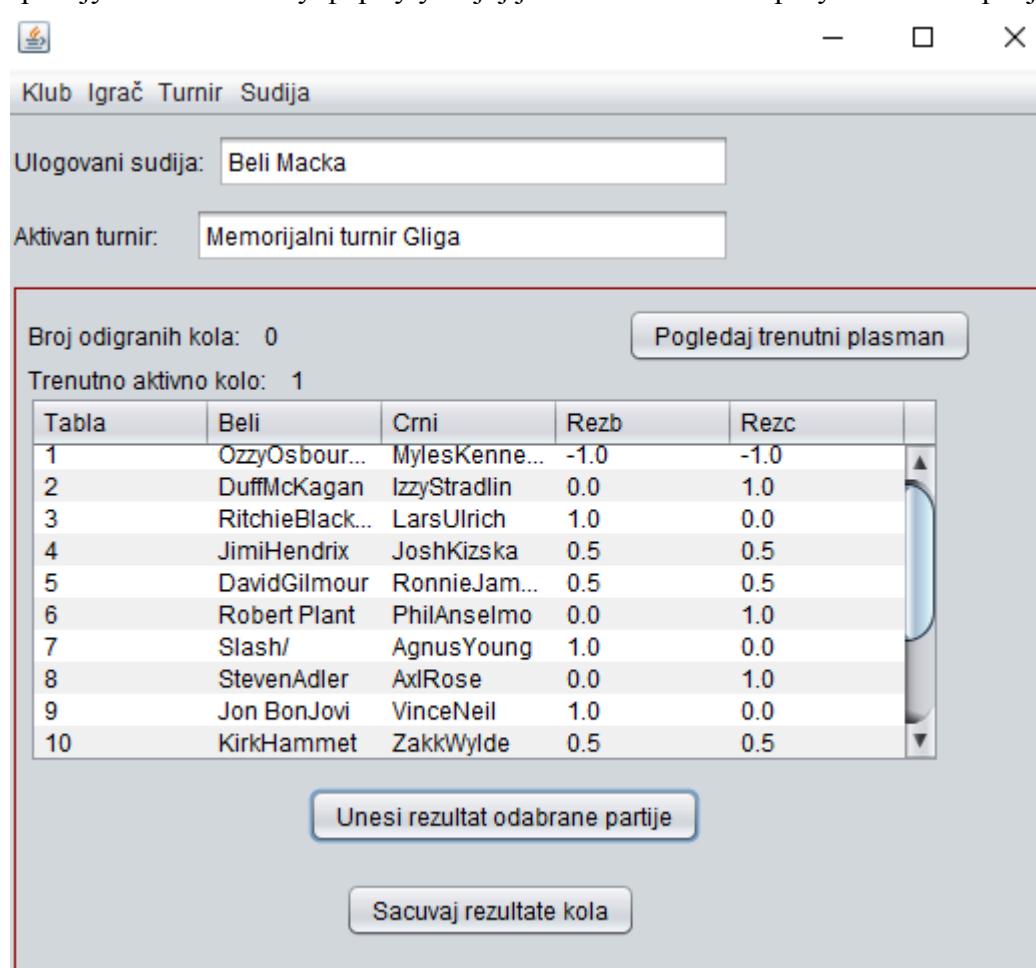
Tabla	Beli	Crni	Rezb	Rezc
1	OzzyOsbour...	MylesKenne...	-1.0	-1.0
2	DuffMcKagan	IzzyStradlin	-1.0	-1.0
3	RitchieBlack...	LarsUlrich	-1.0	-1.0
4	JimiHendrix	JoshKizska	-1.0	-1.0
5	DavidGilmour	RonnieJam...	-1.0	-1.0
6	Robert Plant	PhilAnselmo	-1.0	-1.0
7	Slash/	AgnusYoung	-1.0	-1.0
8	StevenAdler	AxlRose	-1.0	-1.0
9	Jon BonJovi	VinceNeil	-1.0	-1.0
10	KirkHammet	ZakkWylde	-1.0	-1.0

Основни сценарио СК:

- Судија бира партију из табеле за коју хоће да унесе резултат (АПУСО)
- Судија кликом на дугме “Унеси резултат одабране партије” захтева од система да измени партију. (АПСО)
- Систем отвара форму за унос резултата те партије. (СО)

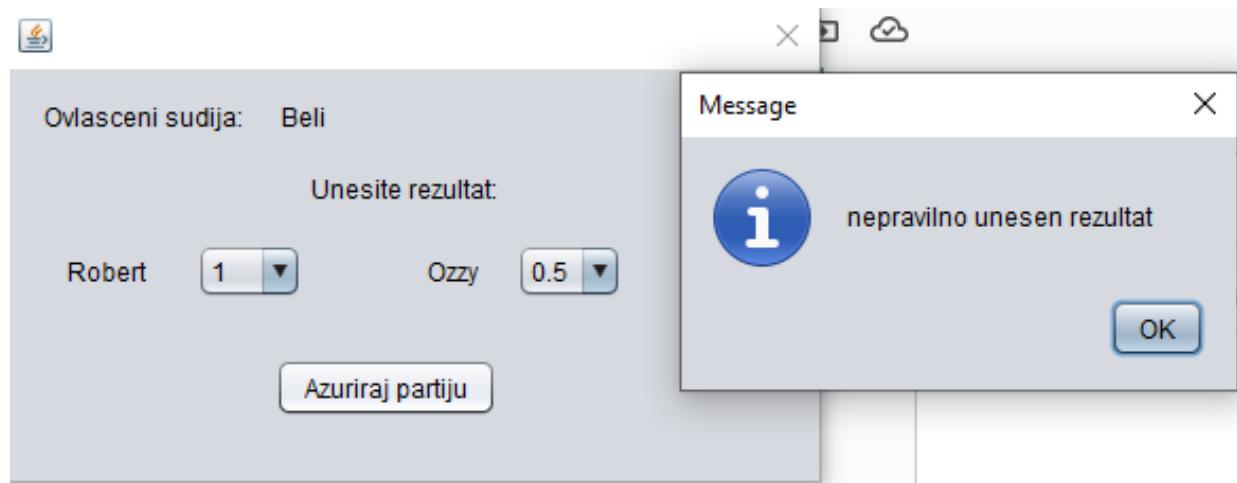


4. Судија уноси резултат партије. (АПУСО)
5. Кликом на дугме “Ажурирај партију” судија захтева од система да промени резултат партије (АПСО)
6. Систем извршава измену у оперативној меморији и затвара дијалог партије враћајући се на главну форму у чијој је табели изменјен резултат ове партије.(СО)



Алтернативни сценарији:

- 6.1. Систем враћа судији поруку уколико не може да сачува резултат партије (ИА)



3.1.4.10 CK7: Случај коришћења - Креирање кола

Назив СК : Креирање кола

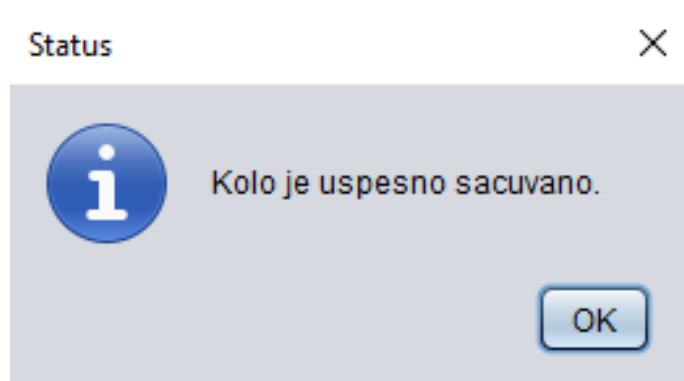
Актори СК : Судија

Учесници СК : Судија и систем (програм)

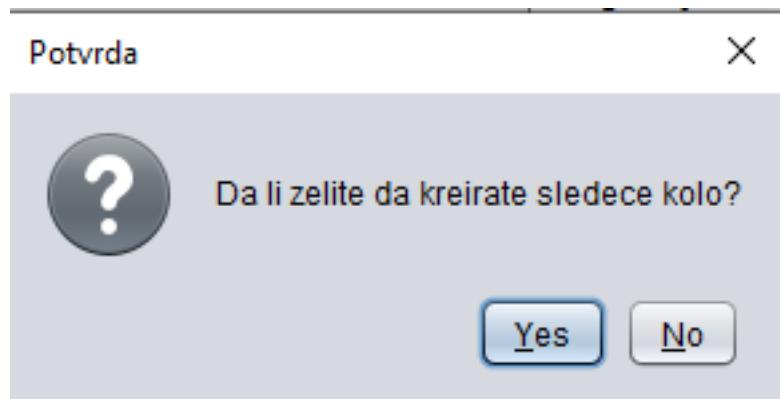
Предуслов: Систем је укључен и судија је улогован под својом шифром. Систем приказује главну форму која се отвара при пријављивању и садржи панел за рад са колом турнира. Учитани су подаци о броју одиграних кола, редном броју тренутно активних кола, турниру и судији. Измењен је резултат свих партија, односно све партије су завршене.

Основни сценарио СК:

- 1.Судија кликом на дугме “Сачувај резултате кола” позива систем да сачува податке о резултатима партија. (АПУСО)
- 2.Систем чува партије, коло и ажурира број поена играча. (СО)
- 3.Систем приказује поруку о успешно сачуваном колу (ИА)



- 4.Систем захтева од судије да одабере хоће ли генерисати следеће коло (ИА)



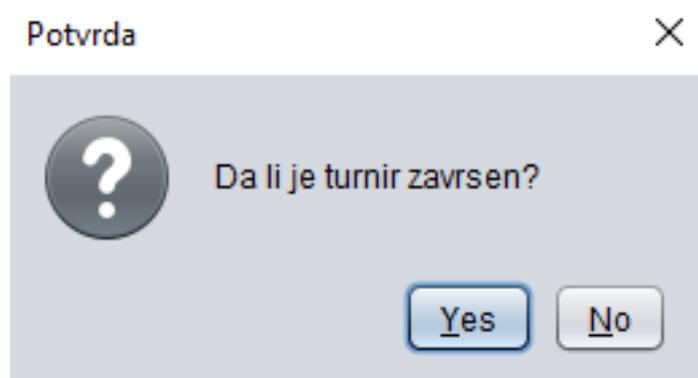
5. Судија бира једну од опција. (АПСО)

Опција да:

6. Уз ажурирање података на форми се понавља исти случај коришћења

Опција не:

6. Систем захтева од судије да одабере да ли је турнир завршен(ИА);



Опција ДА: прелази се на СК10

Опција НЕ: Систем обавештава судију да је немогуће да прекине са генерирањем кола уколико турнир није завршен тј да не може да прекине сесију суђења пре краја турнира.

Ulogovani sudija: Eko Snaucerovic

Aktivan turnir: Atomac

Broj odigranih kola: 5

Pogledaj trenutni plasman

Message: Ne mozete zavrsiti sesiju ili poceti novi turnir pre nego sto se turnir zavrsi. nastavite sa uspesnim sudjenjem. srecno!

OK

8	SebastianB...	RobertTrujillo	1.0	0.0
9	MalcolmYou...	KirkHammet	1.0	0.0
10	John Lennon	JamesHetfield	1.0	0.0

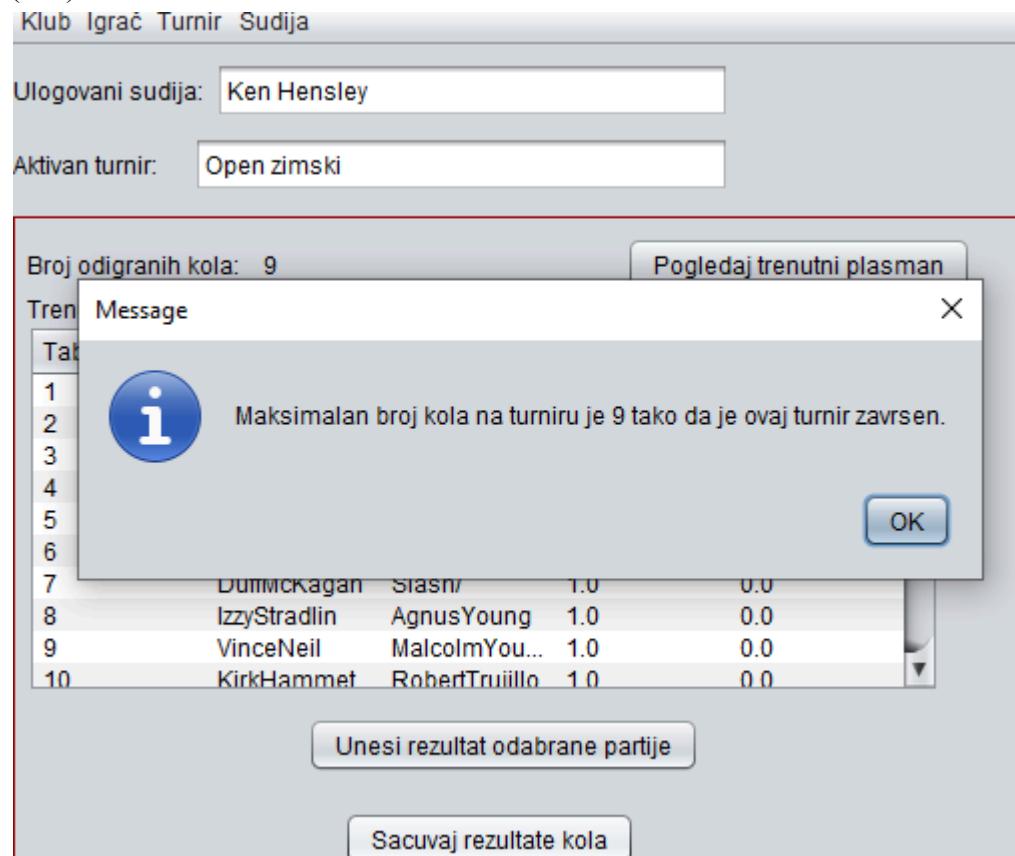
Unesi rezultat odabране partije

Naziv	Mesto	Tip
Atomac	Prostorije ...	reitingovar
Prva liga ...	Stadion O...	reitingovar
Prvenstvo	Matematic...	neratingovo

Кликом на OK дијалог нестаје и судија поново може да генерише следеће коло.

Алтернативна сценарија:

3.1.У случају да не може да се генерише више кола, систем обавештава судију о томе (ИА)



3.1.4.11 CK8: Случај коришћења - Креирање турнира

Назив CK : Креирање турнира

Актори CK : Судија

Учесници CK : Судија и систем (програм)

Предуслов: Систем је укључен и судија је улогован под својом шифром. Отворена је форма за унос података о новом турниру.

- 1.Судија уноси податке о новом турниру (АПУСО)
- 2.Судија контролише да ли је коректно унео податке (АНСО)

Sudija Roko Todorovic

Naziv	<input type="text" value="Kup sampiona Evrope"/>
Mesto	<input type="text" value="Hotel Hajat"/>
Tip	<input type="text" value="rejtingovan"/>
Tempo	<input type="text" value="redovan"/>
Region	<input type="text" value="evropsko"/>

3. Судија позива систем да запамти податке о турниру(АПСО)
4. Систем креира нови турнир и памти податке о њему (СО)
5. Систем приказује продавцу поруку да је запамтио турнир(ИА)

Sudija Roko Todorovic

Status

i TURNIR je uspesno dodat!

Naziv	<input type="text" value="Status"/>
Mesto	<input type="text"/>
Tip	<input type="text"/>
Tempo	<input type="text"/>
Region	<input type="text" value="evropsko"/>

Алтернативни сценарио:

3.1. Уколико систем не може да запамти турнир, шаље судији поруку о томе (ИА)

3.1.4.12. СК9: Случај коришћења - Провера пласмана

Назив СК : Провера пласмана

Актори СК : Судија

Учесници СК : Судија и систем (програм)

Предуслов: Систем је укључен и судија је улогован под својом шифром. Систем приказује главну форму која се отвара при пријављивању и садржи панел за рад са колом турнира. Учитани су подаци о броју одиграних кола, редном броју тренутно активних кола, турниру и судији.

Основни сценарио СК:

1. Притиском на дугме “Погледај тренутни пласман” судија позива систем да прикаже податке о тренутном пласману (АПСО)
2. Систем приказује судији сортирану листу играча тог турнира са бројем поена (CO)

Ime	Prezime	Poeni
Zakk	Wylde	2.0
Myles	Kennedy	2.0
Ian	Gilan	2.0
Sebastian	Bach	2.0
Agnus	Young	2.0
Tony	Martin	2.0
Duff	McKagan	2.0
Jon	BonJovi	2.0
Vince	Neil	2.0
Kirk	Hammet	1.5
David	Coverdale	1.5
Izzy	Stradlin	1.5
Ritchie	Blackmour	1.5
Jimi	Hendrix	1.0
Steven	Adler	1.0
Duff	Plants	1.0

zb	Rez
	0.0
	1.0
	0.5
	0.0
	0.0
	1.0
	0.0
	0.0
	0.0

Pogledaj trenutni plasman

Unesi rezultat odabrane partije

Sacuvaj rezultate kola

3.1.4.13 CK10: Случај коришћења - Измена података о турниру

Назив СК : Измена података о турниру

Актори СК : Судија

Учесници СК : Судија и систем (програм)

Предуслов: Систем је укључен и судија је улогован под својом шифром. Систем приказује форму за претрагу турнира.

Основни сценарио СК:

1. Судија бира који турнир из табеле жели да измени (АПСО)

Odaberite i unesite kriterijum pretrage:

<input type="radio"/> Naziv	<input type="text"/>
<input type="radio"/> Tip	<input type="text" value="kategorini"/>
<input type="radio"/> Tempo	<input type="text" value="redovan"/>
<input type="radio"/> Region	<input type="text" value="evropsko"/>

Prikazi sve

Regista...	Naziv	Mesto	Tip	Tempo	Region
5	Atomac	Prostori...	rejtingo...	brzopot...	drzavno
14	Bozicni ...	Hotel Sl...	nerejtin...	ubrzani	drzavno
28	Dani mi...	Kaleme...	nerejtin...	brzopot...	evropsko
9	Ekipno ...	Lestane...	nerejtin...	ubrzani	evropsko
12	Kategor...	SK Cuk...	kategorini	ubrzani	opstins...
2	Kategor...	SK Vidik...	kategorini	redovan	drzavno
13	Kategor...	SK Crve...	kategorini	brzopot...	balkans...
30	Kup sa...	Hotel H...	rejtingo...	redovan	evropsko
3	Letnji O...	Restora...	rejtingo...	redovan	drzavno
8	Memorij...	Gimnazi...	rejtingo...	redovan	svetsko
11	Prva lig...	Stadion ...	rejtingo...	redovan	drzavno
1	Prvenst...	Matema...	nerejtin...	redovan	balkans...
26	Prvenst...	Bilijar kl...	nerejtin...	brzopot...	svetsko
15	Prvenst...	SK Kralj	nerejtin...	ubrzani	opstins...

2. Судија захтева од система (кликом на дугме измени турнир) да отвори форму за измену турнира са учитаним подацима о истом. (AHCO)

3. Систем отвара форму за измену са учитаним подацима о означеном турниру, таквим да се све осим назива може мењати.(CO)

Naziv turnira

Mesto odrzavanja

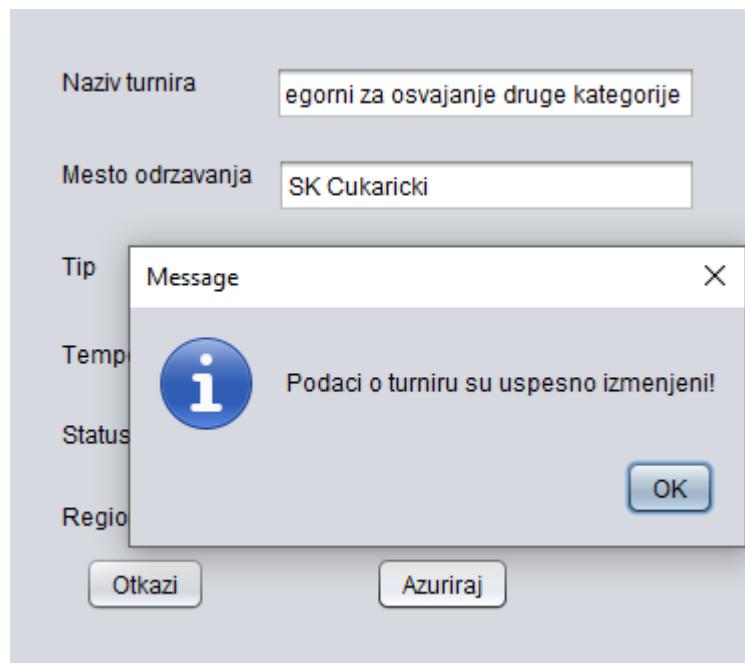
Tip

Tempo

Status

Region

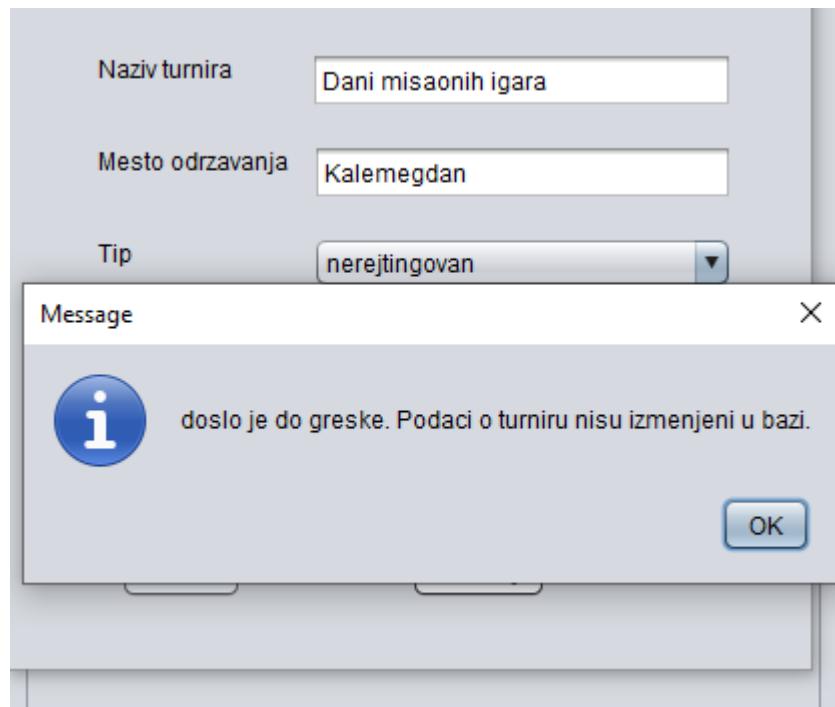
4. Судија мења податке које жели (АПСО)
5. Кликом на дугме ажурирај, судија шаље систему захтев да измени податке о турниру. (АНСО)
6. Систем мења податке у бази и враћа судији поруку. (ИА)



(Не могу се мењати сви атрибути-поље са називом турнира није могуће изменити)

Алтернативна сценарија:

- 6.1. Уколико не може да измени турнир, систем враћа судији поруку о грешци.

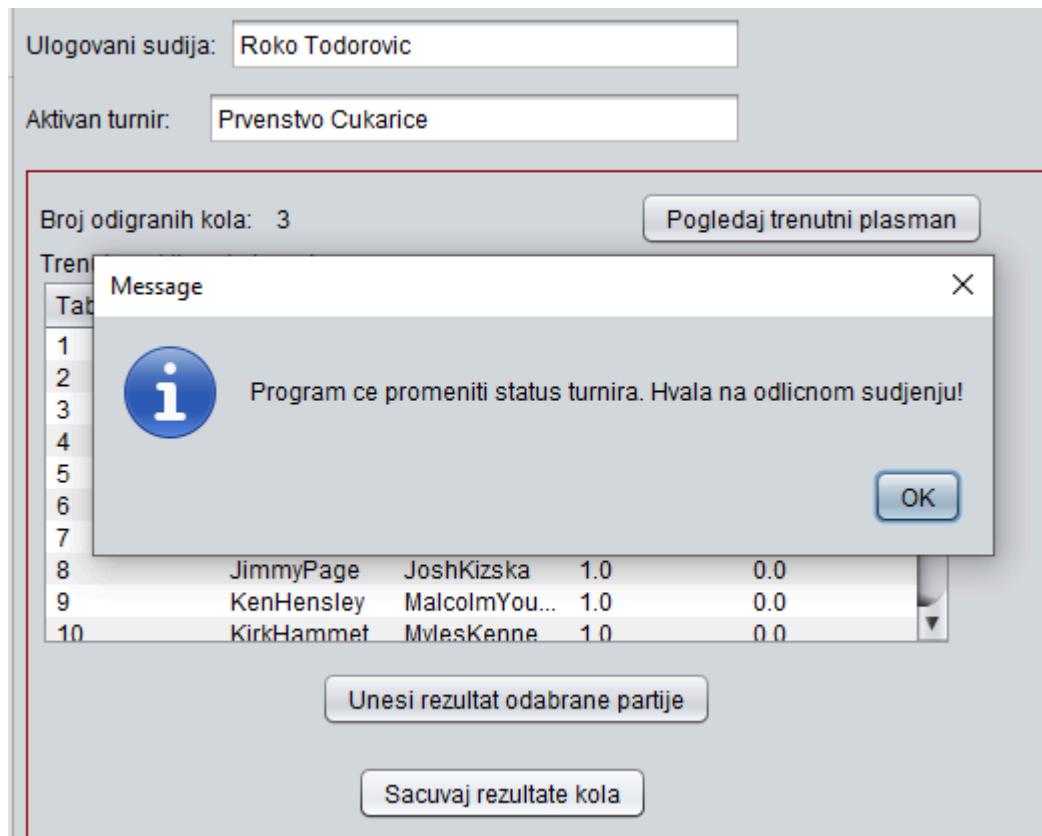


На пример, у случају када судија није изменио ниједан атрибут.

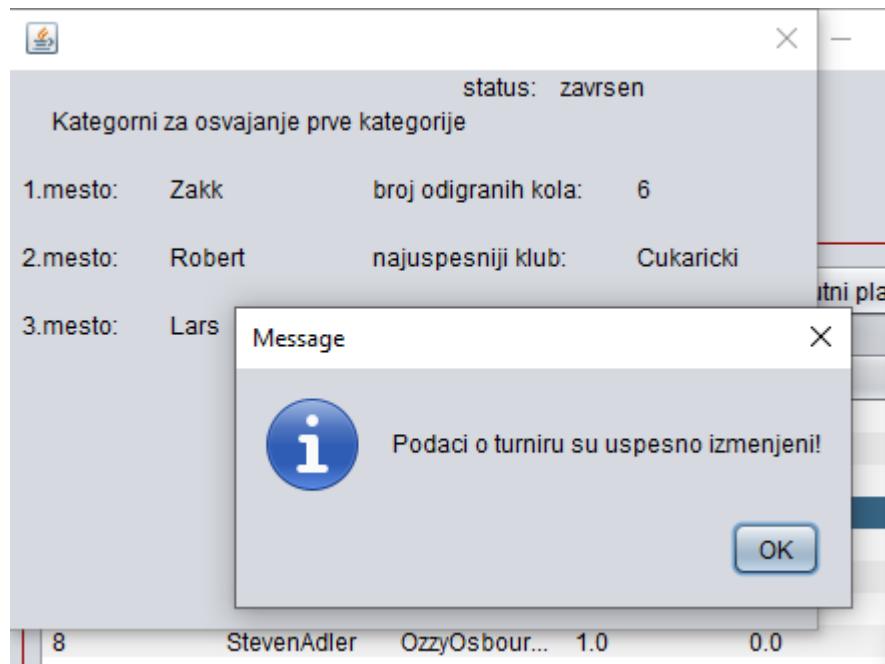
- Још једно појављивање овог случаја коришћења

Предуслов: Систем је укључен и судија је улогован под својом шифром. Одиграно је макар једно коло турнира и судија је одабрао опцију (СК4, корак 6) да турнир јесте завршен или је достигнут максималан број кола (СК4, корак 3.1).

1. Систем обавештава судију да ће статус турнира бити промењен у “завршен”(ИА)
2. Систем мења статус турнира у табели активних турнира (СО)
3. Систем обавештава судију да је измена извршена (ИА)



Одмах након тога, појављује се форма са детаљима о турниру - освојена прва три места, број одиграних кола и клуб чији су се играчи нашли на највишим позицијама.



Након затварања ових дијалога, судији је омогућено да затвори и главну форму за вођење турнира што до сада није био случај (сесија суђења једног турнира се не може прекинути).

3.1.4.14 Пројектовање контролера корисничког интерфејса

Контролер корисничког интерфејса је одговоран за:

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

3.1.4.15 Пројектовање апликационе логике

Апликациони сервери треба да обезбеде сервисе који ће омогућити реализацију апликационе логике софтверског система. Пројектовани апликациони сервер садржи:

- Део за комуникацију са клијентом
- Контролер апликационе логике
- Део за комуникацију са складиштем података (брокер базе података)
- Део који садржи пословну логику

3.1.4.16 Комуникација са клијентима

Део за комуникацију подиже серверски сокет који даље ослушкује мрежу. Када клијент успостави конекцију, сервер генерише нит која ће бити одговорна за двосмерну везу са клијентом.

Клијент шаље захтев за извршење неке од системских операција, одговарајућа нит (додељена клијенту) прихвата захтев и прослеђује га до контролера апликационе логике. Након извршења системске операције, контролер враћа резултат “нити клијента”. Резултат се затим прослеђује клијенту.

Комуникација између клијента и сервера се обавља разменом објекта класе “Клијентски захтев” и класе “СерверскиОдговор” које припадају пакету трансфер.

```
public class KlijentskiZahtev implements Serializable {  
    private int operacija;  
    private Object parametar;  
  
    public KlijentskiZahtev() {  
    }  
  
    public KlijentskiZahtev(int operacija, Object parametar) {  
        this.operacija = operacija;  
        this.parametar = parametar;  
    }  
  
    public Object getParametar() {  
        return parametar;  
    }  
  
    public void setParametar(Object parametar) {  
        this.parametar = parametar;  
    }  
  
    public int getOperacija() {  
        return operacija;  
    }  
  
    public void setOperacija(int operacija) {  
        this.operacija = operacija;  
    }  
  
}  
  
public class ServerskiOdgovor implements Serializable {  
    private Object odgovor;
```

```
private int operacija;
private boolean uspesno;
private String greska;

public ServerskiOdgovor() {
}

public ServerskiOdgovor(Object odgovor, int operacija, boolean uspesno, String greska)
{
    this.odgovor = odgovor;
    this.operacija = operacija;
    this.uspesno = uspesno;
    this.greska = greska;
}

public Object getOdgovor() {
    return odgovor;
}

public void setOdgovor(Object odgovor) {
    this.odgovor = odgovor;
}

public boolean isUspesno() {
    return uspesno;
}

public void setUspesno(boolean uspesno) {
    this.uspesno = uspesno;
}

public String getGreska() {
    return greska;
}

public void setGreska(String greska) {
    this.greska = greska;
}

public int getOperacija() {
    return operacija;
}

public void setOperacija(int operacija) {
```

```

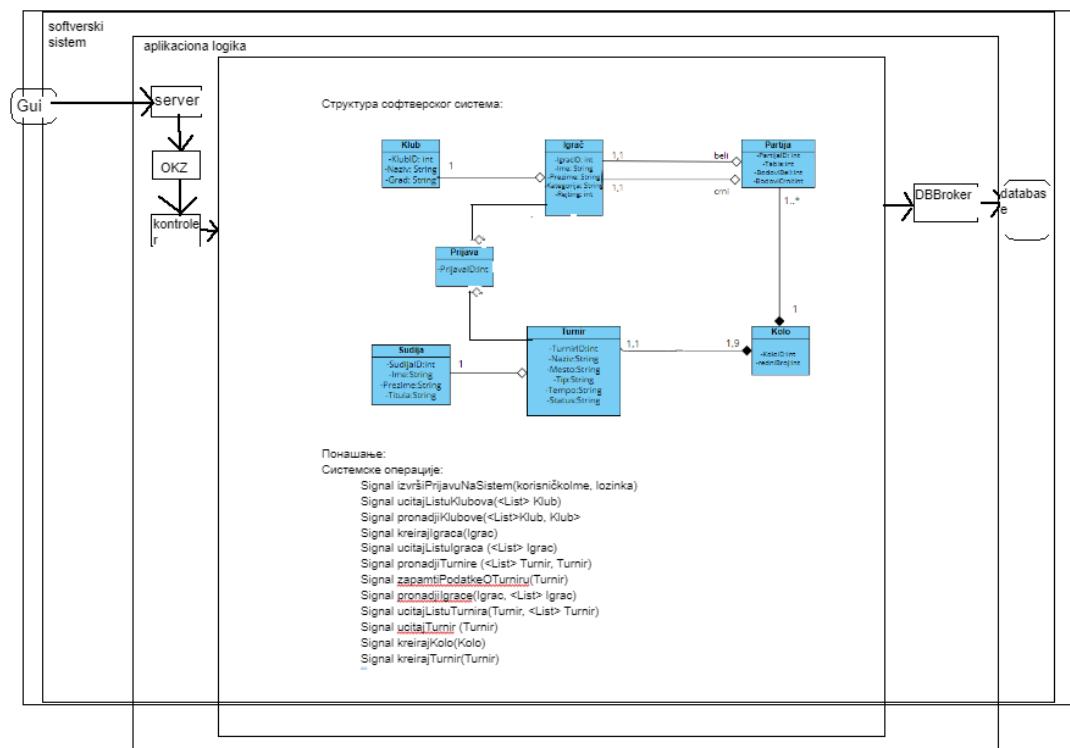
        this.operacija = operacija;
    }
}

```

```
}
```

3.1.4.17 Пројектовање контролера апликационе логике

Контролер апликационе логике прихвата захтеве за извршење системских операција и исте прослеђује до конкретне системске операције. Након извршења системске операције, контролер прихвата одговор и враћа назад позиваоцу (нити клијента). Преко класе КонтролерСервер прихватају се захтеви од нити клијента за извршење системских операција, који се затим прослеђују до одговарајућих класа које су одговорне за извршење системских операција.



Слика 15: архитектура софтверског система након пројектовања контролера апликационе логике

Контролер прима захтеве од нити клијента за извршавање системских операција, који се затим прослеђују класама одговорним за извршење системских операција. За сваку системску операцију праве се софтверске класе које треба да реализују системску операцију. Све класе за реализацију системске операције наслеђују апстрактну класу ГенеричкаОперација. Њена метода костурИзврши(ОпштиДоменскиОбјекат одо) одрађује конекцију на базу, комитује и ролбекује конекције. Остале класе имплементирају методу за валидацију и за извршавање.

Општа системска операција представљена је класом ГенеричкаОперација:

```

public abstract class GenerickaOperacija {
    protected InterfaceDBBroker dbb;

    public GenerickaOperacija() {
        dbb = new KlasaDBBroker();
    }

    public final void Kosturizvrsi(OpstiDomenskiObjekat odo) throws SQLException {
        try {
            pokreniTransakciju();
            validacija(odo);
            izvrsi(odo);
            komitujTransakciju();
        } catch (SQLException ex) {
            rolbekujTransakciju();
            System.out.println("ne valja upit ili nesto sa bazom");
            Logger.getLogger(GenerickaOperacija.class.getName()).log(Level.SEVERE, null,
            ex);
        } catch (Exception ex) {
            System.out.println("ne valja validacija");
            Logger.getLogger(GenerickaOperacija.class.getName()).log(Level.SEVERE, null,
            ex);
        }
    }

    protected abstract void validacija(OpstiDomenskiObjekat odo) throws Exception ;
    protected abstract void izvrsi(OpstiDomenskiObjekat odo) throws Exception ;

    private void pokreniTransakciju() throws SQLException {
        KonekcijaSaBazom.getInstance().getConnection().setAutoCommit(false);
    }

    private void komitujTransakciju() throws SQLException {
        KonekcijaSaBazom.getInstance().commit();
    }

    private void rolbekujTransakciju() throws SQLException {
        KonekcijaSaBazom.getInstance().rollback();
    }

}

```

3.1.4.18 Пројектовање пословне логике

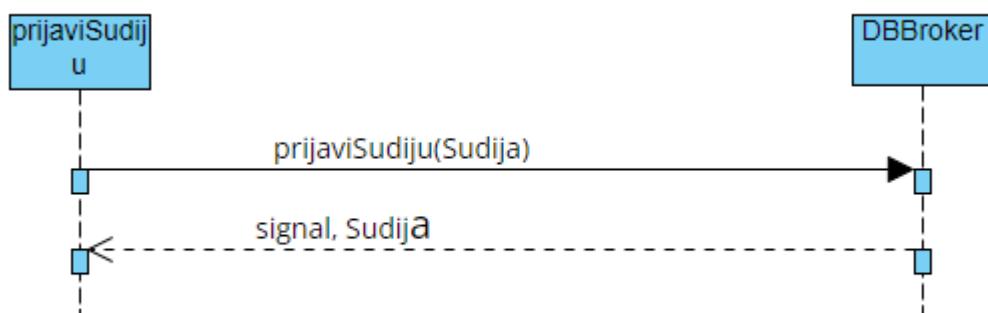
Ово се односи на пројектовање понашања софтверског система односно системске операције. За сваку системску операцију треба направити концептуална решења која су директно повезана са логиком проблема. За сваки од уговора пројектује се концептуално решење.

Уговор УГ1: Signal izvršiPrijavuNaSistem(korisničkoIme, lozinka)

Веза са СК: СК1

Предуслови:

Постуслови:

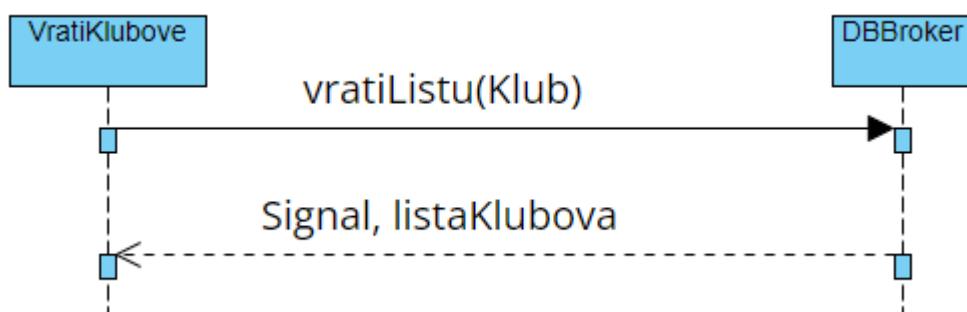


Уговор УГ2:Signal ucitajListuKlubova(<List> Klub)

Веза са СК: СК2, СК3

Предуслови:

Постуслови:

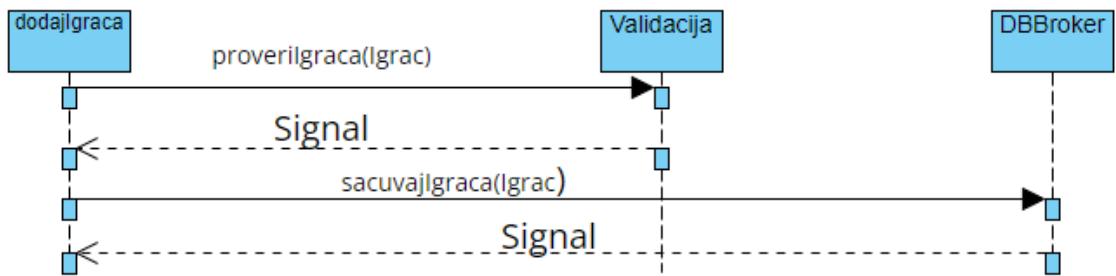


Уговор УГ3:Signal kreirajIgraca(Igrac)

Веза са СК: СК2

Предуслови: Вредносна и структурна ограничења над објектом Igrac морају бити задовољена.

Постуслови: Подаци о играчу су запамћени.

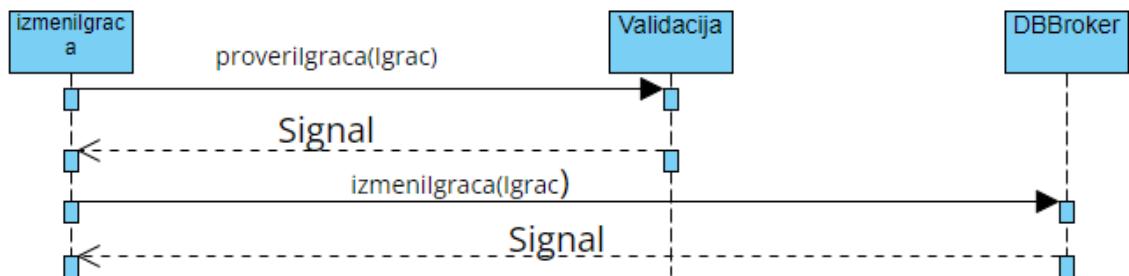


Уговор УГ4:Signal izmeniIgraca(Igrac)

Веза са СК: СК6, СК7

Предуслови: Вредносна и структурна ограничења над објектом Igrac морају бити задовољена.

Постуслови: Подаци о играчу су запамћени.

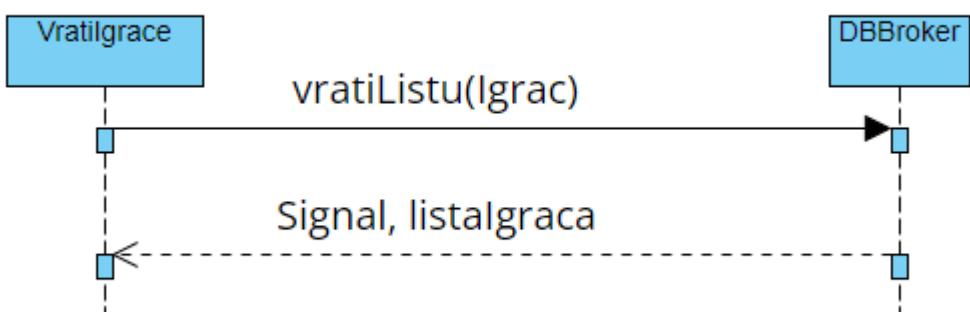


Уговор УГ5:Signal ucitajListuIgraca (<List> Igrac)

Веза са СК: СК5, СК9

Предуслови:

Постуслови:

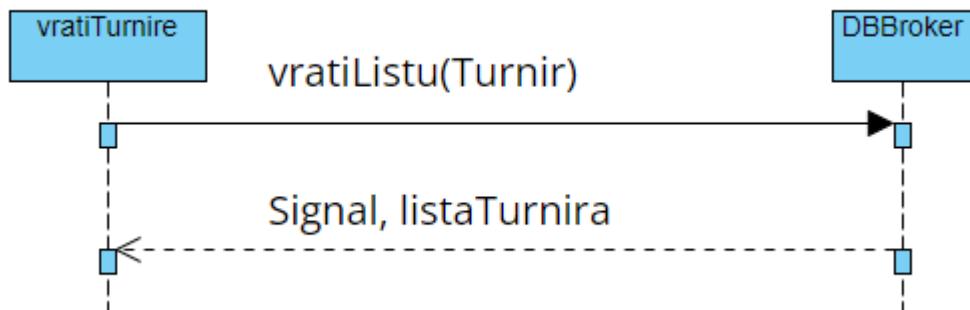


Уговор УГ6: signal ucitajListuTurnira(<List> Turnir)

Веза са СК: СК4, СК10

Предуслови:

Постуслови:

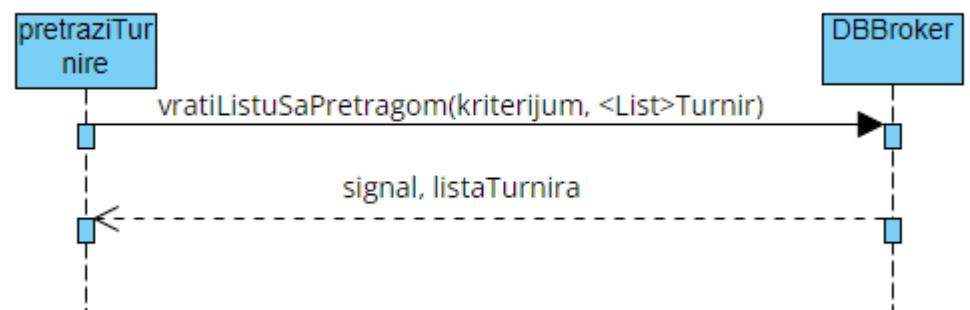


Уговор УГ7:Signal pronadjiTurnire (<List> Turnir, Turnir)

Веза са СК: СК4

Предуслови:

Постуслови:

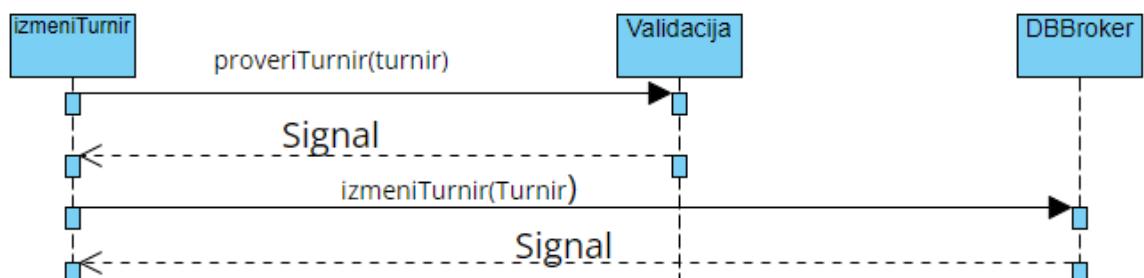


Уговор УГ8:Signal izmeniTTurnir(Turnir)

Веза са СК: СК10

Предуслови: Вредносна и структурна ограничења над објектом **Turnir** морају бити задовољена.

Постуслови: Подаци о турниру су запамћени.

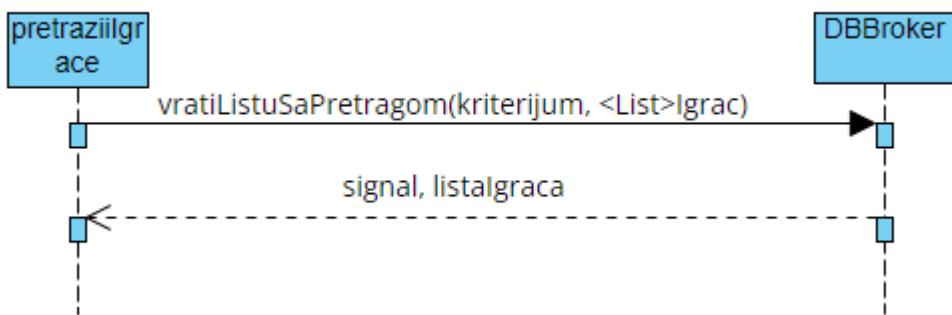


Уговор УГ9:Signal pronadjiIgrace(Igrac, <List> Igrac)

Веза са СК: СК5

Предуслови:

Постуслови:

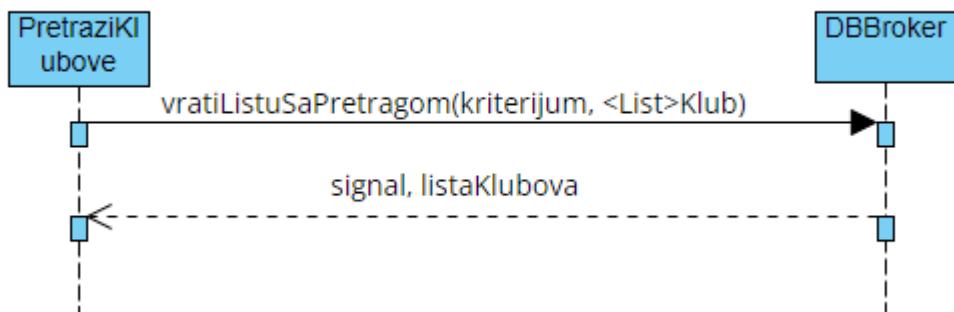


Уговор УГ10:Signal pronadjiKlubove(Klub, <List> Klub)

Веза са СК: СК3

Предуслови:

Постуслови:

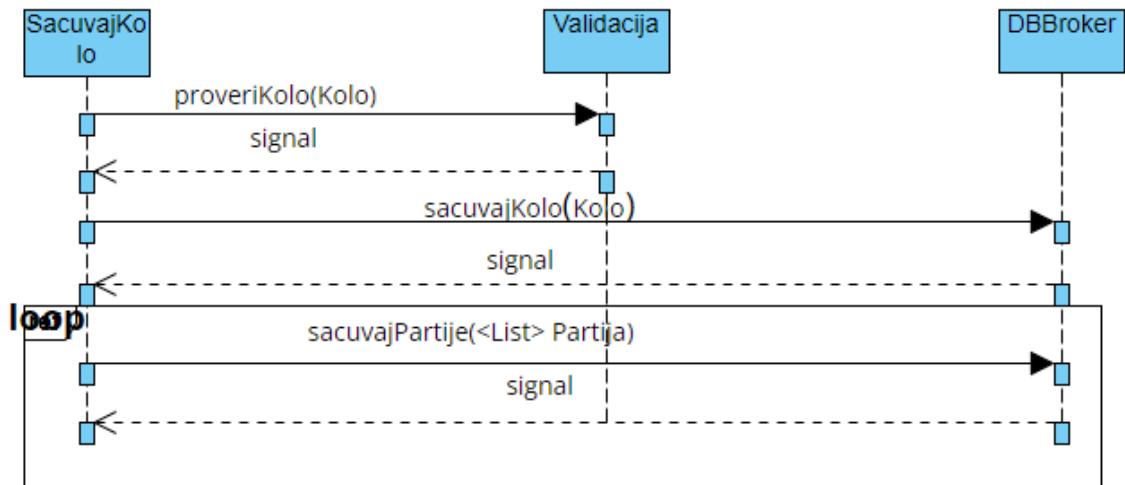


Уговор УГ11:Signal kreirajKolo(Kolo)

Веза са СК: СК7

Предуслови: Вредносна и структурна ограничења над објектом Коло морају бити задовољена.

Постуслови: Подаци о колу су запамћени.

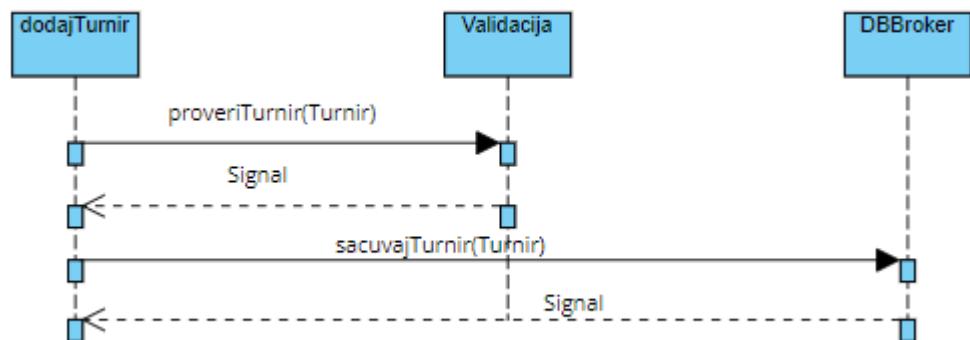


Уговор УГ12:Signal kreirajTurnir(Turnir)

Веза са СК: СК8

Предуслови: Структурна ограничења над објектом Turnir морају бити задовољена.

Постуслови: Подаци о турниру су запамћени.

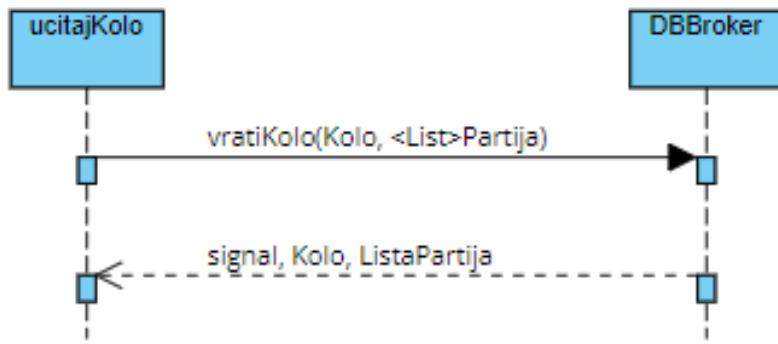


Уговор УГ13: Signal ucitajKolo(Kolo, <List>Partija)

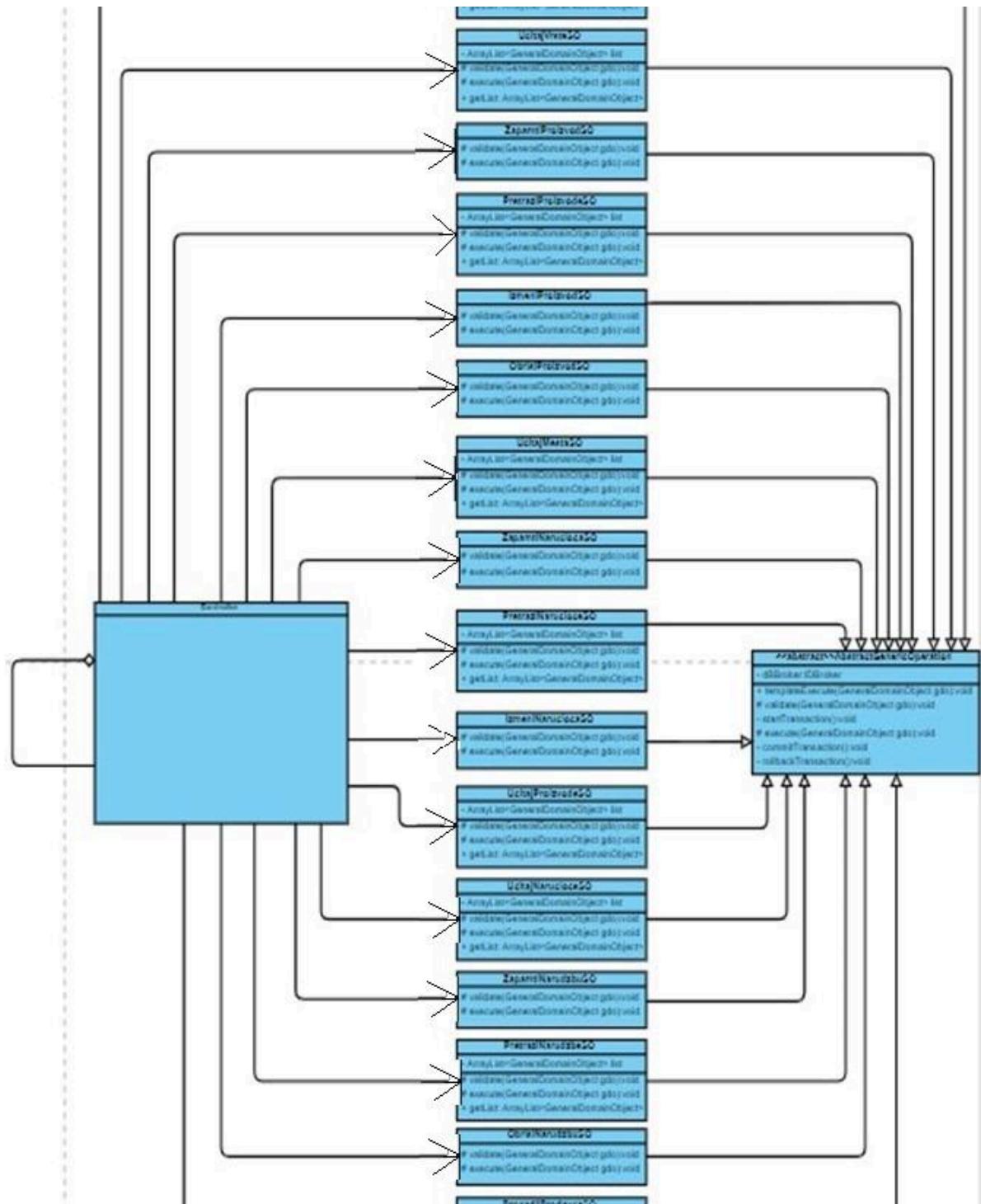
Веза са СК: СК6

Предуслови:

Постуслови:



На основу уговора, направљен је велики дијаграм класа који показује везу између контролера апликационе логике и класа одговорних за извршење системских операција које наслеђују класу ОпштаГенеричкаОперација.



Слика 16: дијаграм класа који показује везу између контролера апликационе логике и класа одговорних за извршење системских операција које наслеђују класи **Општа** **Генеричка** **Операција**

3.1.4.19 Пројектовање структуре софтверског система

На основу концептуалних класа праве се софтверске класе структуре. Свака класа има приватна поља атрибута, гетере и сетере за те атрибуте, беспараметарски конструктор и параметарски конструктор. Доменске класе имплементирају интерфејс ОпштиДоменскиОбјекат да би омогућиле лакше имплементирање метода

ДББрокера. На тај начин брокер базе података прима интерфејс уместо самих класа и олакшава креирање генеричких упита.

Интерфејс ОпштиДоменскиОбјекат:

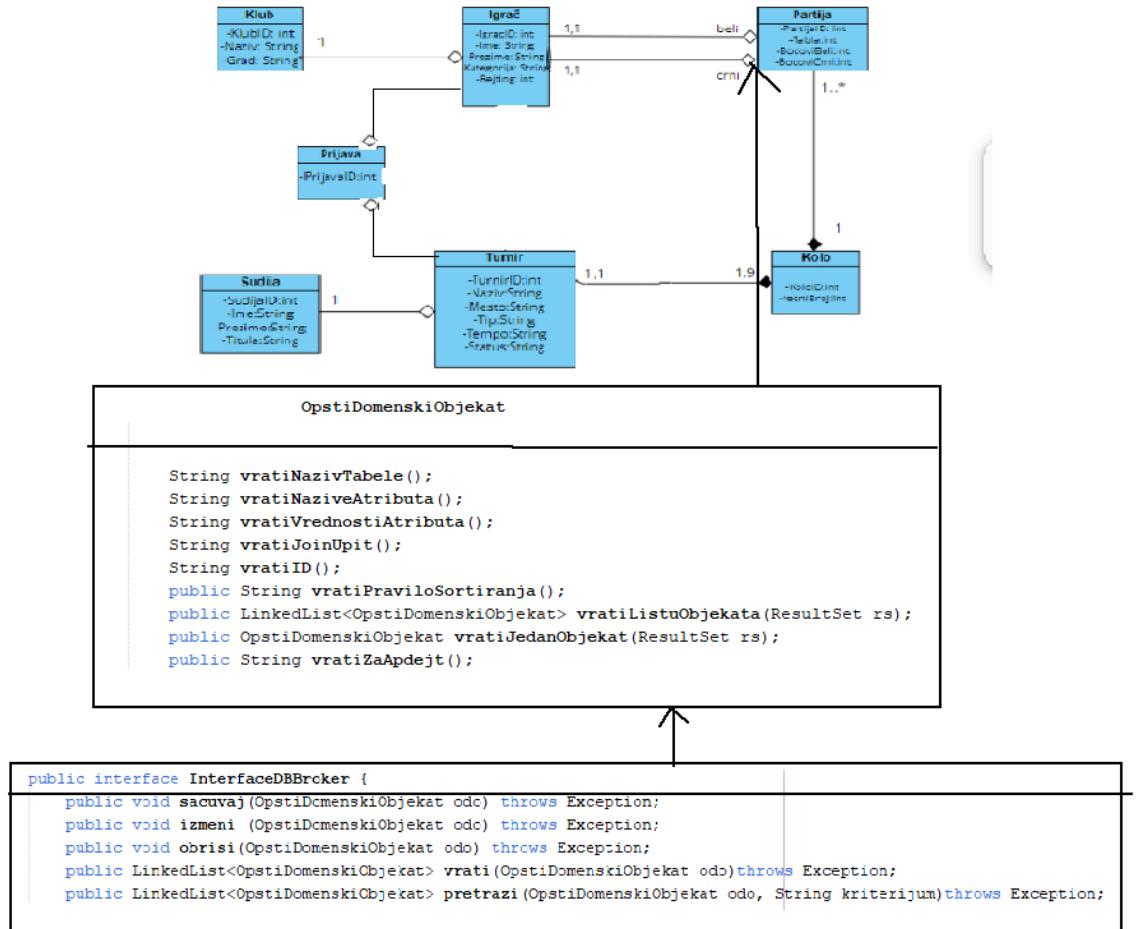
```
public interface OpstiDomenskiObjekat extends Serializable{
```

```
    String vratiNazivTabele();
    String vratiNaziveAtributa();
    String vratiVrednostiAtributa();
    String vratiJoinUpit();
    String vratiID();
    public String vratiPraviloSortiranja();
    public LinkedList<OpstiDomenskiObjekat> vratiListuObjekata(ResultSet rs);
    public OpstiDomenskiObjekat vratiJedanObjekat(ResultSet rs);
    public String vratiZaApdejt();
}
```

3.1.4.20 Пројектовање брокера базе података

Класа КласаДББрокер представља оквир који посредује у свим операцијама над базом података и имплементира интерфејс ДББрокер који је у коду задат на следећи начин:

```
public interface InterfaceDBBroker {
    public void sacuvaj(OpstiDomenskiObjekat odo) throws Exception;
    public void izmeni (OpstiDomenskiObjekat odo) throws Exception;
    public void obrisi(OpstiDomenskiObjekat odo) throws Exception;
    public LinkedList<OpstiDomenskiObjekat> vrati(OpstiDomenskiObjekat odo) throws
Exception;
    public LinkedList<OpstiDomenskiObjekat> pretrazi(OpstiDomenskiObjekat odo,
String kriterijum) throws Exception;
}
```



Слика 17: Повезаност класа ДББрокер и ОпштиДоменскиОбјекат

3.1.4.21 Пројектовање складишта података

На основу софтверских класа структуре пројектоване су табеле релационог система за управљање базом података. Коришћен је MySQL server, алат графичког корисничког интерфејса SQLyog Community 64.

Табела играч:

Column Name	Data Type
IgracID	int
KlubID	int
Ime	varchar
Prezime	varchar
Kategorija	varchar
Rejting	int

Табела клуб:

	Column Name	Data Type
	KlubID	int
	Naziv	varchar
	Grad	varchar

Табела коло:

	Column Name
	id
	turnirid
	redni

Табела партија:

Delete selected columns(s) (Alt+Del)		
	id	int
	tabla	int
	sudija	int
	beli	int
	crni	int
	kolo	int
	turnir	int
	bodovibeli	double
	bodovicrni	double

Табела судија:

	Column Name	Data Type
	SudijaID	int
	Ime	varchar
	Prezime	varchar
	Titula	varchar
	Lozinka	varchar
	Korisnicko	varchar
	TurnirID	int

Табела турнир:

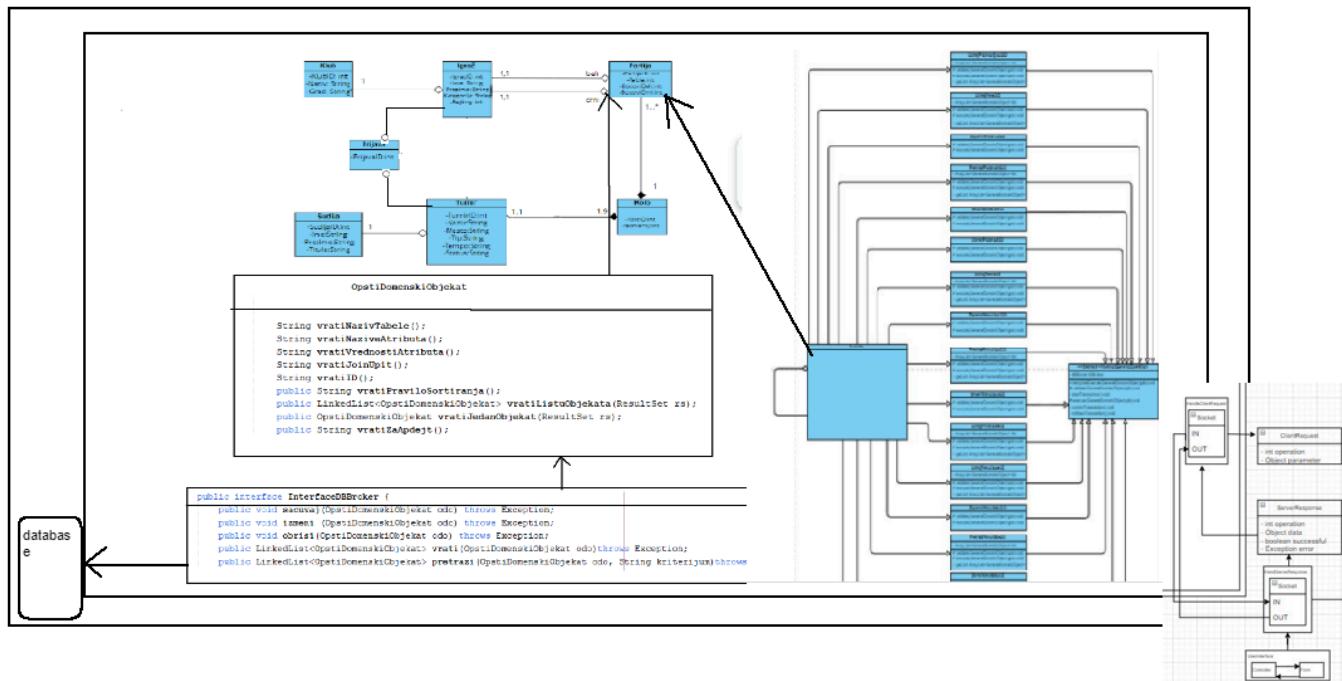
<input type="checkbox"/>	Column Name
<input type="checkbox"/>	TurnirID
<input type="checkbox"/>	Naziv
<input type="checkbox"/>	Mesto
<input type="checkbox"/>	Tip
<input type="checkbox"/>	Tempo
<input type="checkbox"/>	SudijaID
<input type="checkbox"/>	Status
<input type="checkbox"/>	Slika
<input type="checkbox"/>	Region

Табела пријава:

Column Name	Data Type	
PrijavaID	int	▼
TurnirID	int	▼
IgracID	int	▼
updated_at	datetime	▼
created_at	datetime	▼
		▼

3.1.4.22 Комплетна архитектура софтверског система

Након пројектовања табела из базе података добијамо слику комплетне архитектуре софтверског система.



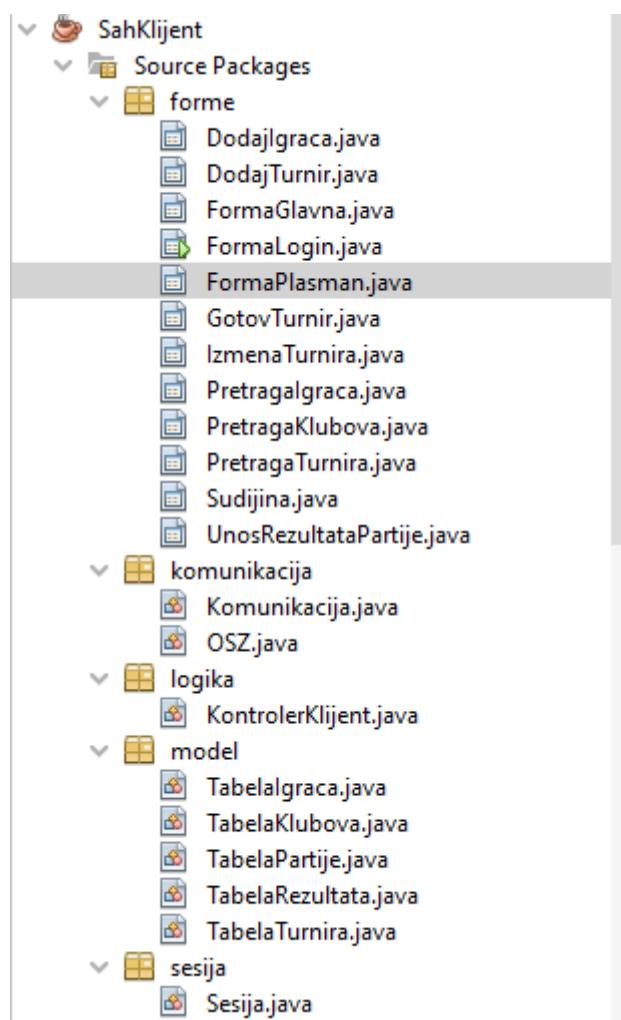
Слика 18: архитектура софтверског система

3.1.5 Имплементација

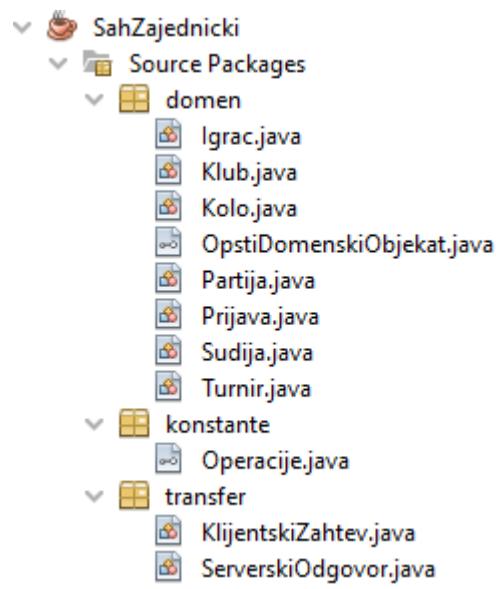
Софтверски систем креиран као овај пројекат је развијен у програмском језику Java и пројектован као клијент - сервер апликација. Као развојно окружење коришћен је Apache Netbeans IDE 15, а као систем за управљање базом података коришћен је MySQL сервер.

Читав систем је реализован као три пројекта: ШахКлијент, ШахСервер и ШахЗаједнички. На основу архитектуре софтверског система добијене су следеће софтверске класе:

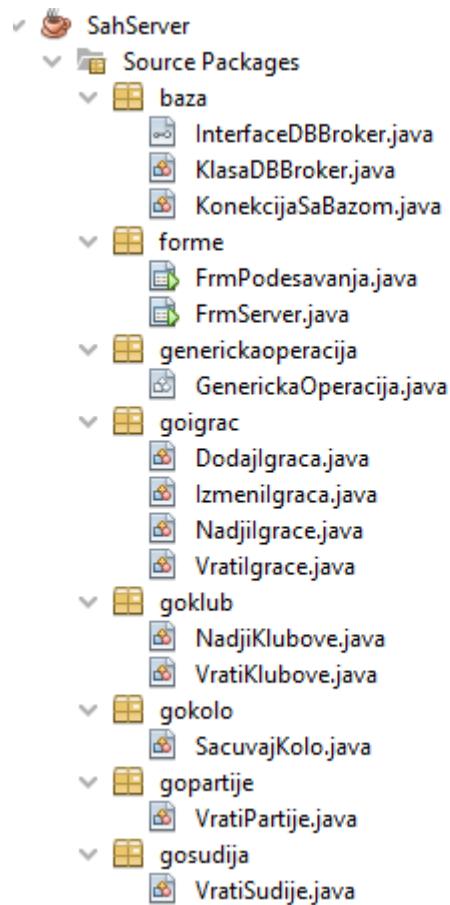
3.1.5.1 Пројекат SahKlijent:

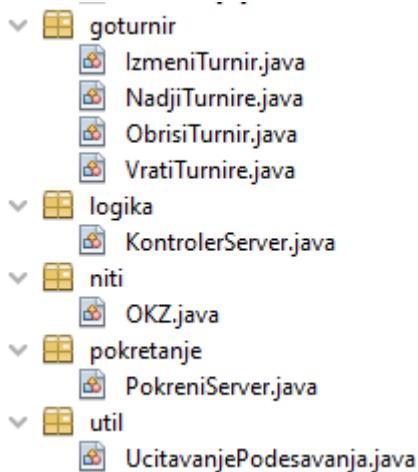


3.1.5.2 Пројекат SahZajednicki



3.1.5.3 Пројекат SahServer





3.1.6 Тестирање

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

Тестирање је обављено ручно. У Java пројекту на backend-у је коришћен Ant где је једна од зависности (енг. dependency) mysql-connector-java преко које backend комуницира са базом података, где се налазе сви објекти у виду табела. Покретањем самог програма у развојном окружењу и постизањем конекције са базом преко localhost-а где су преко екранских форми извршаване функционалности, тестиране су све функције.

3.2 РАЗВОЈ ИГРАЧКОГ МОДУЛА ЗА ПРИЈАВУ/ОДЈАВУ УЧЕШЋА НА ТУРНИРИМА

Сврха постојања овог могула је у поједностављивању процеса пријављивања на турнир. Уместо да судија уноси податке о сваком играчу што би му одузело доста времена или да се пријаве врше неким конвенционалнијим путем (долазак лично, позив, пријава пред сам турнир), омогућено је играчима да се за само неколико секунди пријаве, посетом свима доступног сајта.

Такође, на backend-у модулас задате су функције које решавају све неправилности које би потенцијално настале као последица једноставности алгоритма за пријављивање на frontend-у модила (када се корпа играча пријављених за одређен турнир узима из базе - не узимају се дупликати, уколико неком играчу није дозвољено да учествује, метода ће га такође искључити на основу података којима добије приступ након што се играч региструје/покуша да се пријави).

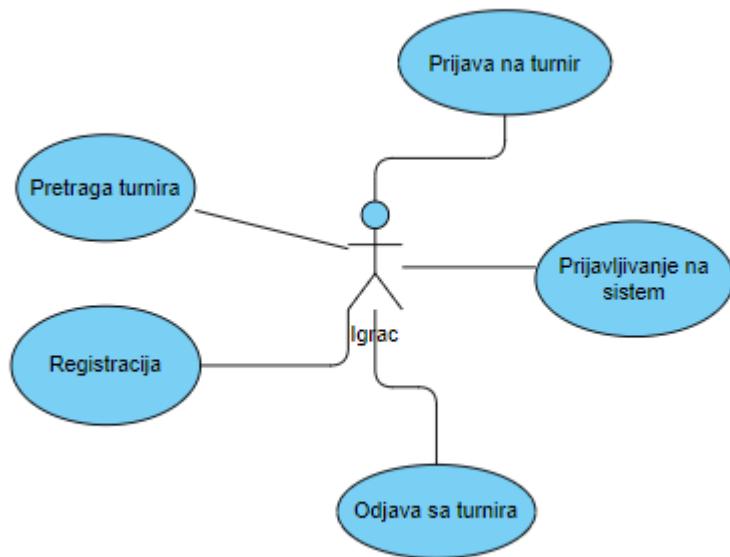
3.2.1 Веरбални опис система

Креирана је веб апликација на којој свако може направити налог, а пријавом на исти постати потенцијални играч неког од тренутно доступних турнира. Осим актуелности и чланака из света шаха, апликација садржи страницу са табелом свих турнира (уз могућност претраге) на којој играч има могућност да се пријави на неки од њих и страницу где се налазе турнири на које је он већ пријављен која садржи опцију за одустајање од турнира уколико се играч предомислио, а опцију за проверу резултата који је остварио уколико је тај турнир завршен.

3.2.2 Прикупљање корисничких захтева

У овом модулу идентификовано је пет случајева коришћења:

- 1.Пријављивање играча на систем
- 2.Регистрација новог играча
- 3.Претрага турнира
- 4.Пријављивање играча на турнир
- 5.Одјављивање играча са турнира



Слика 19. Дијаграм случајева коришћења

3.2.2.1 СК1: Случај коришћења - Пријављивање играча на систем

Назив СК

Пријављивање играча на систем

Актори СК

Играч

Учесници СК

Играч и систем (програм)

Предуслов: Систем је укључен и приказује форму за пријављивање играча на систем.

Основни сценарио СК

- 1.Играч уноси корисничко име и лозинку. (АПУСО)
- 2.Играч контролише да ли је коректно унео податке за пријављивање. (АНСО)
- 3.Играч позива систем да изврши пријављивање. (АПСО)
- 4.Систем проверава унешене податке. (СО)
- 5.Систем приказује играчу поруку: “успешно сте се пријавили на систем”. (ИА)

Алтернативна сценарија

- 5.1. Уколико играч није унео правилне податке, систем приказује поруку “пријављивање је неуспешно”. (ИА)

3.2.2.2 СК2: Случај коришћења - Регистрација новог играча

Назив СК

Регистрација играча

Актори СК

Играч

Учесници СК

Играч и систем (програм)

Предуслов: Систем је укључен и приказује форму за уношење података о играчу.

Основни сценарио СК

1. **Играч** **позива** **систем** да креира **играча**. (АПСО)
2. **Систем** **креира** **играча**. (СО)
3. **Систем** **приказује** поруку: “**Систем** је креирао играча“. (ИА)
4. **Играч** **уноси** своје податке у недавно креирани ентитет играч. (АПУСО)
5. **Играл** **контролише** да ли је коректно унео своје податке. (АНСО)
6. **Играл** **позива** **систем** да запамти податке о **играчу**. (АПСО)
7. **Систем** **памти** податке о **играчу** . (СО)
8. **Систем** **приказује** **играчу** запамћеног **играча** и поруку: “**Успешно сте се регистровали**“. (ИА)

Алтернативна сценарија

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

8.1 Уколико **систем** не може да запамти податке о **играчу**, приказује **играчу** поруку “Систем не може да запамти **податке о играчу**”. (ИА)

3.2.2.3 СК3: Случај коришћења - Пријављивање играча на турнир

Назив СК

Пријава на турнир

Актори СК

Играч

Учесници СК

Играч и **систем** (програм)

Предуслов: Систем је укључен и **играч** је улогован под својом шифром. Систем приказује форму за рад са **турнирима**. Учитана је листа турнира.

Основни сценарио СК

1. **Играч** бира неки од приказаних турнира. (АПУСО)
2. **Играч** кликом на дугме поред назива турнира позива систем да креира **пријаву**. (АПСО)
3. **Систем** креира **пријаву**. (СО)
4. **Систем** приказује **играчу** податке о **пријави** и поруку “**Успешно сте се пријавили на турнир!**.” (ИА)

Алтернативна сценарија

4.1 Уколико **систем** не може да креира **пријаву**, он показује **судији** поруку “**систем** не може да креира **пријаву**”. (ИА)

3.2.2.4 СК4: Случај коришћења - Претрага турнира

Назив СК

Претраживање турнира

Актори СК

Играч

Учесници СК

Играч и систем (програм)

Предуслов: Систем је укључен и Играч је улогован под својом шифром. Систем приказује форму за рад са турнирима. Учитана је листа турнира.

Основни сценарио СК

1. Играч уноси вредност по којој претражује турнире . (АПУСО)
2. Играч позива систем да нађе турнире по задатој вредности. (АПСО)
3. Систем тражи турнире по задатој вредности. (СО)
4. Систем приказује играчу све турнире чији се подаци поклапају са задатом вредношћу и поруку: “Систем је нашао турнире по задатој вредности”. (ИА)

Алтернативна сценарија

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

3.2.2.5 СК5: Случај коришћења - Одјављивање играча са турнира

Назив СК

Одјава са турнира

Актори СК

Играч

Учесници СК

Играч и систем (програм)

Предуслов: Систем је укључен и играч је улогован под својом шифром. Систем приказује форму за рад са турнирима. Учитана је листа турнира на које је пријављен улоговани играч.

Основни сценарио СК

1. Играч бира неки од приказаних турнира. (АПУСО)
2. Играч кликом на дугме поред назива турнира позива систем да обрише пријаву. (АПСО)
3. Систем брише пријаву. (СО)
4. Систем приказује играчу поруку “Ваша пријава је успешно обрисана!.” (ИА)

Алтернативна сценарија

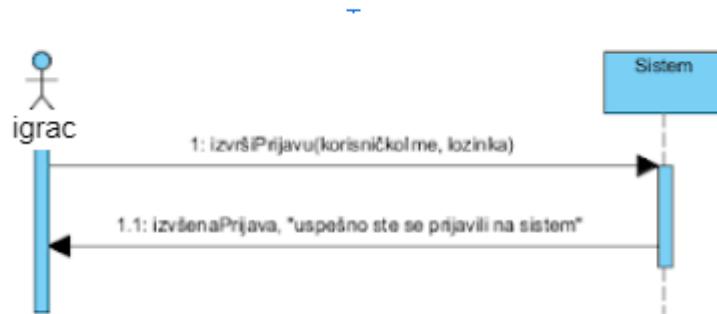
4.1 Уколико систем не може да обрише пријаву, он показује судији поруку “систем не може да обрише пријаву”. (ИА)

3.2.3 Фаза анализе

3.2.3.1 ДС1: Дијаграм секвенце случаја коришћења - Пријављивање играча на систем

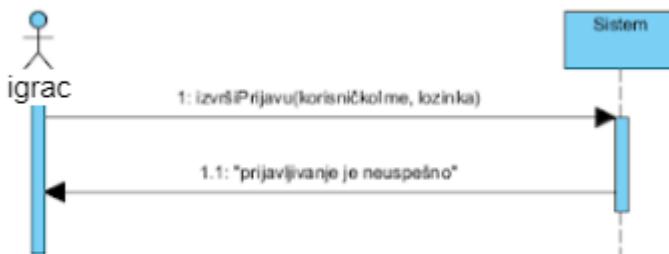
Основни сценарио:

- 1.играч позива систем да изврши пријављивање. (АПСО)
- 2.Систем приказује судији поруку “успешно сте се пријавили на систем”. (ИА)



Алтернативни сценарио:

- 2.1 Уколико судија/играч није унео правилне податке, систем приказује поруку “пријављивање је неуспешно”. (ИА)



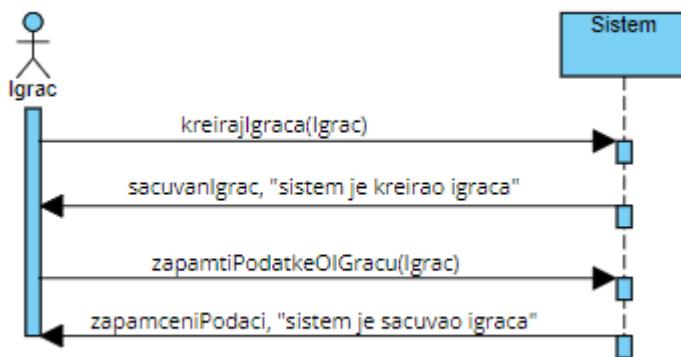
Са наведених секвенцних дијаграма уочавају се системске операције:

1. Signal izvrshiPrijavuNaSistem(korisnickoIme, lozinka)

3.2.3.2 ДС2: Дијаграм секвенце случаја коришћења - Регистрација новог играча

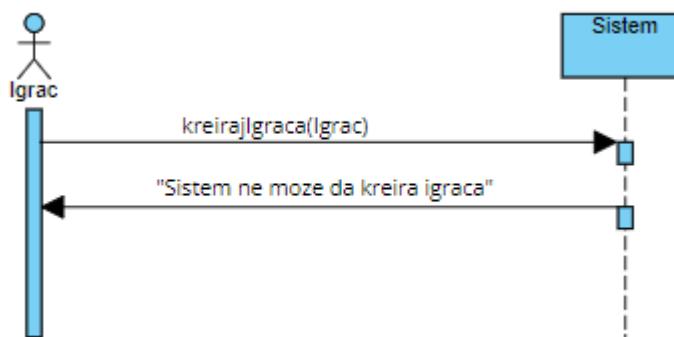
Основни сценарио:

- 1.Играч позива систем да креира играча. (АПСО)
- 2.Систем приказује поруку: “Систем је креирао играча“. (ИА)
- 3.Играч позива систем да запамти податке које је унео. (АПСО)
- 4.Систем приказује играчу поруку: “Успешно сте се регистровали“. (ИА)

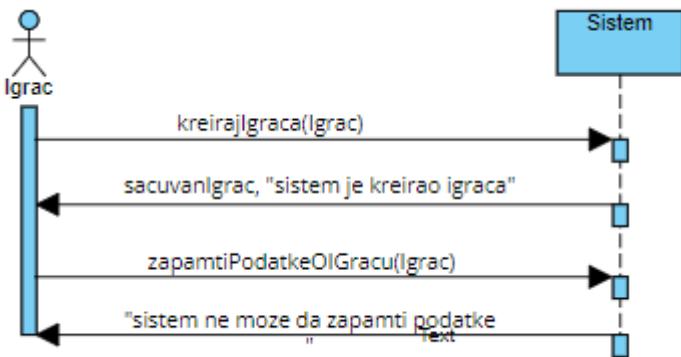


Алтернативни сценарији:

2.1 Уколико систем не може да креира играча он приказује играчу поруку: "Систем не може да креира играча". Прекида се извршење сценарија. (ИА)



4.1 Уколико систем не може да запамти податке о играчу, приказује судији поруку "Систем не може да запамти податке о играчу ". (ИА)



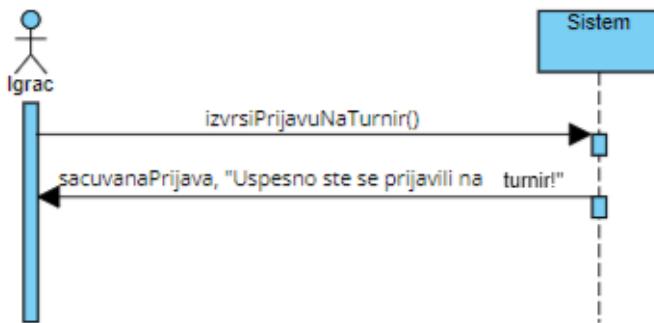
Са наведених секвенцних дијаграма уочавају се системске операције:

- 1.Signal `kreirajIgraca(Igrac)`
- 2.Signal `zapamtiPodatkeOIGracu(Igrac)`

3.2.3.3 ДС3: Дијаграм секвенце случаја коришћења - Пријављивање играча на турнир

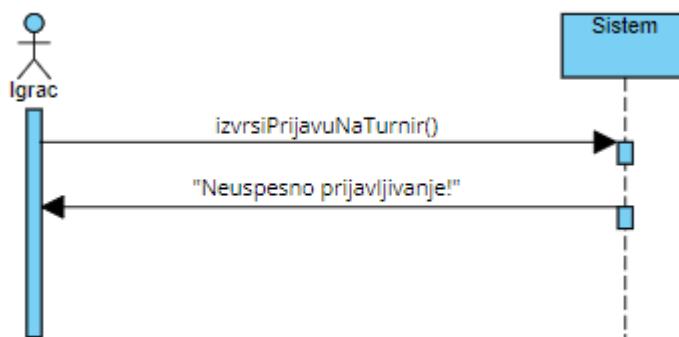
Основни сценарио:

- 1.Играч кликом на дугме позива систем да креира пријаву (АПСО)
- 2.Систем приказује играчу поруку "успешно сте се пријавили на турнир". (ИА)



Алтернативни сценарио:

2.1 Уколико играч не може да се пријави на тај турнир, систем приказује поруку "пријављивање је неуспешно". (ИА)



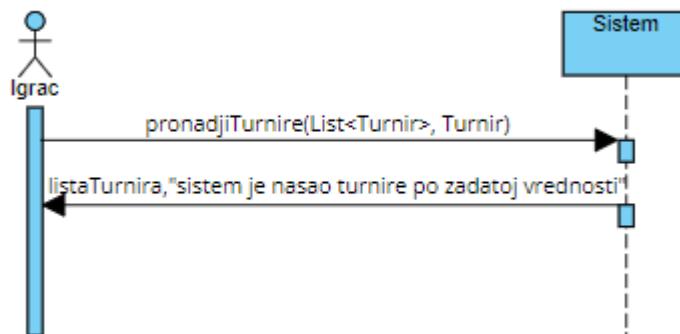
Са наведених секвенцних дијаграма уочавају се системске операције:

1.Signal izvrsiPrijavuNaTurnir()

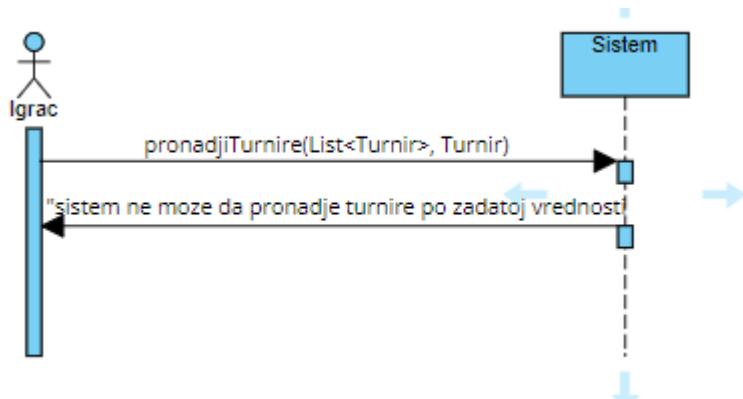
3.2.3.4 ДС4: Дијаграм секвенце случаја коришћења - Претрага турнира

Основни сценарио:

- 1.Играч позива систем да нађе турнире по задатој вредности. (АПСО)
- 2.Систем приказује играчу све турнирчији се подаци поклапају са задатом вредношћу и поруку: "Систем је нашао турнире по задатој вредности". (ИА)



2.1 Уколико систем не може да нађе турнире он приказује играчу поруку: "Систем не може да нађе турнире по задатој вредности". (ИА)



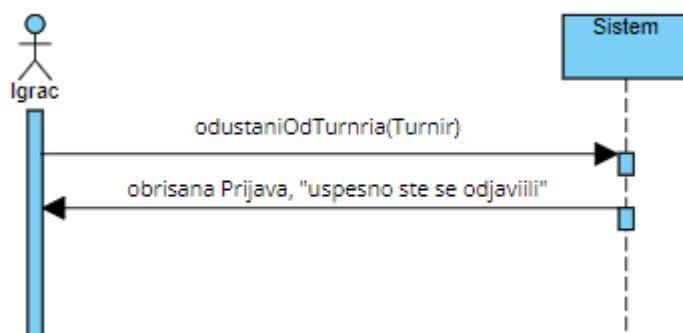
Са наведених секвенцних дијаграма уочавају се системске операције:

- 1.Signal pronadjiTurnire(Turnir, <List> Turnir)

3.2.3.5 ДС5: Дијаграм секвенце случаја коришћења - Одјављивање играча са турнира

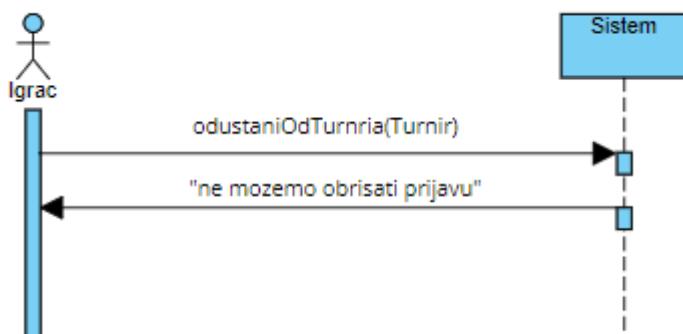
Основни сценарио:

- 1.Играч кликом на дугме поред назива турнира позива систем да обрише пријаву.
- 2.Систем приказује поруку “Ваша пријава је успешно избрисана”



Алтернативни сценарио:

- 2.1 Уколико систем не може да обрише пријаву , приказује поруку “не можемо обрисати вашу пријаву”



Са наведених секвенцних дијаграма уочавају се системске операције:

- 1.Signal odustaniOdTurnira(Turnir)

Како резултат анализе сценарија добијено је укупно шест операција које треба пројектовати:

- 1.Signal izvrsiPrijavuNaSistem(korisnickoIme, Lozinka)
- 2.Signal kreirajIgraca(Igrac)
- 3.Signal zapamtiPodatkeOIgracu(Igrac)
- 4.Signal pretraziTurnire (<List>Turnir, Turnir)
- 5.Signal prijaviSeNaTurnir(Prijava)
- 6.Signal odustaniOdTurnira(Prijava)

3.2.3.6 Понашање софтверског система - дефинисање уговора о системским операцијама

Уговор УГ1: Signal izvrsiPrijavuNaSistem()

Веза са СК: СК1

Предуслови:

Постуслови:

Уговор УГ2: Signal kreirajIgraca(Igrac)

Веза са СК: СК2

Предуслови: Вредносна и структурна ограничења над објектом Igrac морају бити задовољена.

Постуслови: Подаци о играчу су запамћени.

Уговор УГ3: signal ucitajListuTurnira(<List> Turnir)

Веза са СК: СК3

Предуслови:

Постуслови:

Уговор УГ4: Signal pronadjiTurnire (<List> Turnir, Turnir)

Веза са СК: СК3

Предуслови:

Постуслови:

Уговор УГ5: Signal kreirajPrijavu(Prijava)

Веза са СК: СК4

Предуслови: Вредносна и структурна ограничења над објектом Prijava морају бити задовољена.

Постуслови: Подаци о колу су запамћени.

Уговор УГ6:Signal obrisiPrijavu(Prijava)

Веза са СК: СК5

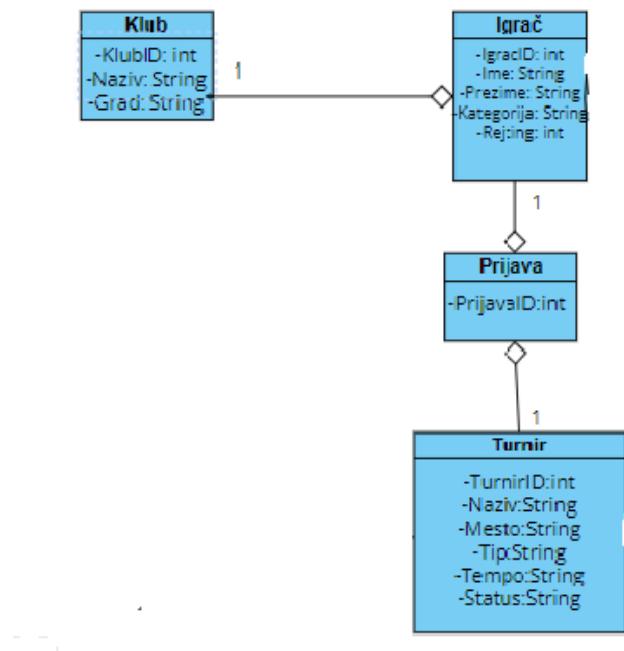
Предуслови: Ако је пријава обрисана не може се извршити системска операција.

Структурна ограничења над објектом Пријава морају бити задовољена.

Постуслови: Пријава је обрисана

3.2.3.7 Структура софтверског система - концептуални (доменски) модел

Структура софтверског система описана је помоћу концептуалног модела. За те потребе коришћен је UML дијаграм класа.



Слика 20. Концептуални (доменски) модел играчког модула

3.2.3.8 Структура софтверског система - релациони модел

На основу претходно приказаног концептуалног модела описаног помоћу дијаграма класа следи опис релационог модела посматраног софтверског система.

Klub (klubID, naziv, grad)

Turnir(IDturnira, naziv, mesto, tip, tempo, status)

Igrač(IgračID, klubID, ime, prezime, kategorija, rejting)

Prijava(prijavID, IDTurira, IDIgraca)

Табела 8: табела структурних и вредносних ограничења класе “Клуб”

Tabela Klub (atributi)	Prosto vrednosno ogranicenje	Slozeno vrednosno ogranicenje	Strukturno ogranicenje
klubID	Int , not null and >0		INSERT / UPDATE CASCADES IGRAC
naziv	String, not null		
grad	String		DELETE RESTRICTED IGRAC

Табела 9: табела структурних и вредносних ограничења класе “Турнир”

Tabela Turnir (atributi)	Prosto vrednosno ogranicenje	Slozeno vrednosno ogranicenje	Strukturno ограничење
IDturnira	Long , not null and >0		INSERT RESTRICTED SUDIJA
naziv	String, not null		
Mesto	String		
tip	String, not null		
tempo	String, not null		
status	String, not null		

Табела 10: табела структурних и вредносних ограничења класе “Играч”

Tabela Igrac (atributi)	Prosto vrednosno ограничење	Slozeno vrednosno ограничење	Strukturnо ограничење
IgracID	Long, not null and >0		INSERT RESTRICTED KLUB
klubID	Integer, not null and >0		UPDATE RESTRICTED KLUB
imeIPrezime	String, not null		CASCADES PRIJAVA
rejting	Integer		

			DELETE RESTRICTED PRIJAVA
kategorija	String		

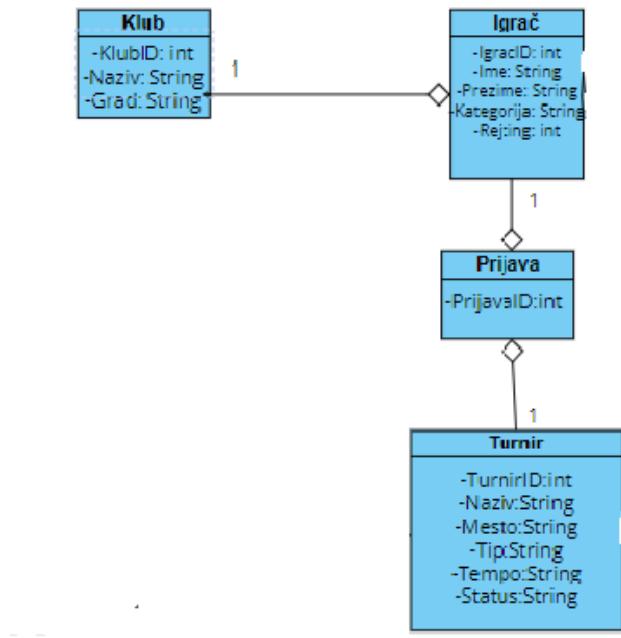
Табела 11: табела структурних и вредносних ограничења класе “Пријава”

Tabela Prijava	Prosto vrednosno ogranicenje	Slozeno vrednosno ogranicenje	Struktuno ogranicenje
IDIgraca	Integer, not null and >0		INSERT RESTRICTED IGRAC, TURNIR UPDATE RESTRICTED IGRAC, TURNIR DELETE /
IDTurnira	Integer , not null and >0		
IDPrijava	Integer, not null and >0		

Као резултат анализе сценарија СК и прављења концептуалног модела добија се логичка структура и понашање софтверског система. Тиме је завршена фаза анализе чији је крајњи резултат пословна логика система.

СОФТВЕРСКИ СИСТЕМ

Структура софтверског система:



Понашање софтверског система

Системске операције:

1. Signal izvrsiPrijavuNaSistem(korisnickoIme, Lozinka)
2. Signal kreirajIgraca(Igrac)
3. Signal zapamtiPodatkeOIgracu(Igrac)
4. Signal pretraziTurnire (<List>Turnir, Turnir)
5. Signal prijaviSeNaTurnir(Prijava)
6. Signal odustaniOdTurnira(Prijava)

3.2.4 Пројектовање

Циљ фазе пројектовања и уједно њен излазни елемент је целокупна архитектура софтверског система. У наставку биће приказани њени делови појединачно, а затим обједињени.

3.2.4.1 Архитектура софтверског система

Овај модул има REST API архитектуру. Backend рађен је у Laravel framework-у и он је задужен за комуникацију са релационом базом података (SQL). Кориснички део имплементиран је преко ReactJS-а који корисничке захтеве путем HTTP захтева шаље у одређеном формату. У зависности од типа захтева, корисник може послати

ахтев са GET, POST, PUT или DELETE методом, услед чега је задатак сервера да покрене одговарајућу методу и врати кориснику одговор у траженом формату

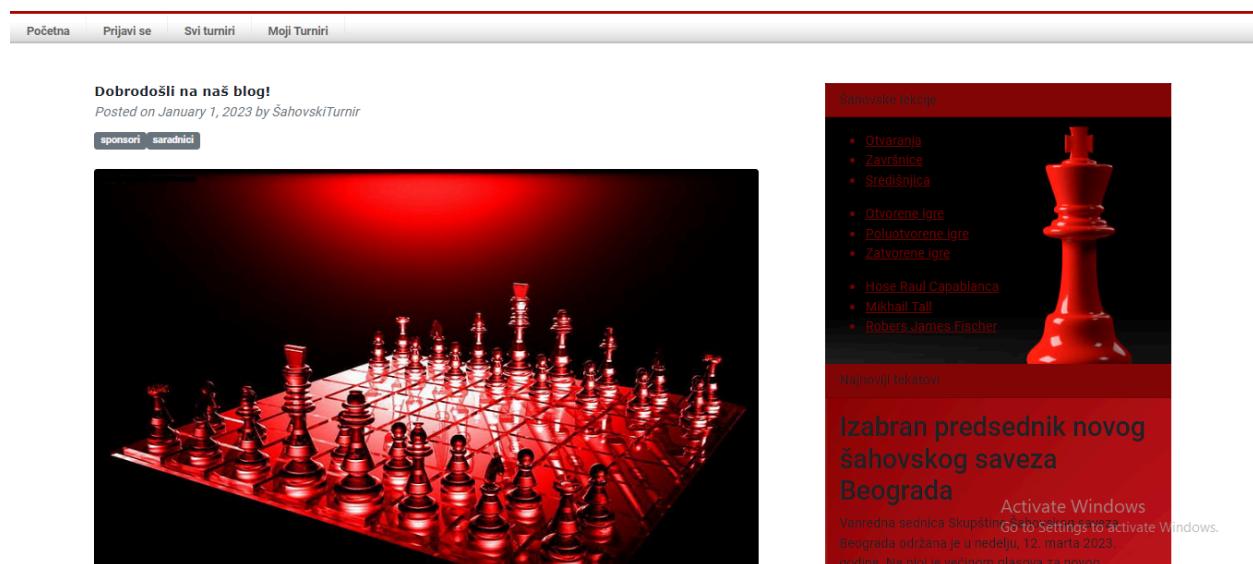
3.2.4.2 Пројектовање корисничког интерфејса

Екранска форма у ReactJS налази се у виду single-page страница које омогућавају такозвано “сакривање странице” на којој се корисник налази и приказ оне на коју кликне што је знатно брже него класично учитавање сваке стране и њених пратећих фајлова посебно.

Све што корисник унесе или покрене на екранским формама, прослеђује се контролеру корисничког интерфејса који даље комуницира са склadiштем података. У овом модулу, контролер је имплементиран на backend-у, а захтеви до њега стижу уз помоћ axios библиотеке која обезбеђује API комуникацију.

3.2.4.3 Пројектовање екранских форми

Модул за играче замишљен је као мала веб апликација преко које ће се они пријављивати на турнире. При отварању апликације, приказује се почетна страна која садржи најновије занимљивости, актуелности и чланке из света шаха и она има чисто информативни карактер. Ту су и секције за коментаре и линкови ка шаховским лекцијама које су, као и садржај ове странице доступне сваком ко отвори сајт.



The screenshot shows a chess-themed website. At the top, there is a navigation bar with links: 'Početna', 'Prijavi se', 'Svi turniri', and 'Moji Turniri'. Below the navigation bar, there is a header section with the text 'Dobrodošli na naš blog!' and 'Posted on January 1, 2023 by ŠahovskiTurnir'. There are also buttons for 'sponsori' and 'saradnici'. The main content area features a large image of a chessboard with red and black pieces. To the right of the image, there is a sidebar with the title 'Šahovske lekcije' and a list of topics: 'Otvaranja', 'Završnice', 'Srednjica', 'Otvorene igre', 'Polotvorene igre', 'Zatvorene igre', 'Hose Raul Capablanca', 'Mikhail Tal', and 'Robers James Fischer'. Below this, there is a section titled 'Najnoviji tekstovi' with the heading 'Izabran predsednik novog šahovskog saveza Beograda'. At the bottom of the sidebar, there is a note about activating Windows: 'Activate Windows' and 'Vanredna sednica Skupštine Šahovskog saveza Beograda održana je u nedelju, 12. marta 2023. godine. Na njoj je većinom glasova za novog'.



Šah je sinonim za mudrost, strategiju i inteligenciju i ima puno zanimljivih priča i legendi vezanih za ovu igru. Izuzetno je zanimljiva legenda o nastanku šaha, poznata kao priča o „šahu i pirlaću“.

Novi potez? Danas neko teško može dokazati da je odigrao novi potez, jer će pre ili kasnije neko doći i pokazati da je taj potez odigrao deceniju pre na nekom turniru C kategorije ili ga odigrao u nekoj kafanskoj partiji. Ali, mnogi potezi koji su odlicni su skroz zaboravljeni. -Aleksandar Aljehin

Pogled na šahovsku olimpijadu me uvek uzbudjuje. Svi ti ljudi iz raznih zemalja-neki iz veoma bogatih, neki iz komunističkih, neki iz kapitalističkih- ali svi dele zajedničko oduševljenje za šah. Na prvi pogled izgleda mi savršen primer kako svi ljudi na svetu mogu da se takmiče u prijateljstvu i miru. A zatim počinje borba. -Garry Kasparov.

Šah na Balkanu u ekspanziji!

Uvođenje šaha kao izbornog predmeta u školama i sve veći broj turnira za mlade doveli su do značajnog povećanja zainteresovanosti za ovu igru na našim prostorima u prethodnih nekoliko godina.

Sve je veći broj dece koja u ranom uzrastu pokazuju interesovanje za učenje ove igre pa postaju članovi šahovskih klubova kojih su takođe sve brojniji. Nije iznenadujuće da to dobro utiče i na kvalitet igre s obzirom na to da se treneri i bivši šahisti sve više posvećuju radu sa mališanima.

Našim igre s obzirom na to da se učenje i razvoj djece svi više posvećuju radu sa mališanima.

Pridružite se diskusiji i ostavite svoj komentar!

Dr.Viktor Mihajlović

Dzabe, cim deca porastu nametnu im se druge vrednosti i sah vise nije "in"



sasneRole

Lepo je cuti ovako nesto! Nije samo do misaone igre, razvijanje takmicarskog duha kroz turnire je super



mackatron

I ti klubovi su samo uzimanje para



Gost

Sah pomaze intelektualnom razvoju dece

predsednika SSB izabran je Andrija Jorgić - Od osnivanja Saveza 1957. do danas mnogi predsednici SSB bili su ljudi različitih profesija, ali ono je što je svima bila zajedničko jeste ljubav prema šahu. Svako od njih je ostavio trag da se šah u Beogradu unapredi, a tim putem ču i ja, jer smatram da beogradski šah zasluguje da bude lokomotiva srpskog šaha i učinimo sve da tako i bude - izjavio je Jorgić.

Queen's Online turnir za žene

FIDE i FIDE omisija za ženski šah (WOM) organizuje online turnir Karen memorial 2022. Ce se na Turniru platformi. Kvalifikacije ce se odrzati po kontinentima: to: 05. jun – Amerika, 11-12. jun – Afrika i Azija, 19. jun – Evropa i finala (pojedinačno i ekipno) 25-26. jun.

Ding je novi svetski šampion u šahu!

ASTANA - Kineski šahista Liren Ding osvojio je titulu svetskog šampiona, pošto je danas u Astani pobedio Rusa Jana Nepomjačnog preko dodatnog tri krajla rezultatom 2,5 - 1,5. Kineski i ruski šahisti su danas odigrali četiri partije sa kontrolnim vremenom 25 minuta, plus 10 sekundi po potezu. Međutim, u poslednjoj WS partiji klasičnog šaha završen je u zadnjim 7,75 minuta. Ding je prvi Kinez, koji je postao svetski šampion u pojedinačnoj konkurenциji. Ding je postao svetski

partija klasičnog šaha završen bez vremenskog, 7,75. Ding je prvi Kinez, koji je postao svetski šampion u šahu u muškoj konkurenциji. On je postao 21. šampion u historiji šaha. Ding i Nepomjačni su igrali za titulu svetskog šampiona pošto je Norvežanin Magnus Carlsen odlučio da ne brani titulu.

Fudbaler Alexander Arnold izazvao Carlsena na mec

Alexander-Arnold tako se po drugi put pojavio na velikom takmičenju nakon jednog nastupa na SP 2018., a na velikoj sceni igrao je i šah, i to mi manje ni više protiv velikog šampiona Magnusa Carlsena. Bilo je to 2018. u sklopu kampanje za promovisanje sporta, a Trent je u meču protiv velemaštira izdržao 17 poteza. Čini se malo? Poredjenja radi, poznati tehnološki gigant Bill Gates četiri godine ranije uspeo je odigrati tek devet poteza protiv Carlsena.

Activate Windows
Go to Settings to activate Windows.

Međutim, da bi se koristile funkcionalnosti sajta zloga kojih i postoji, потребно da je da se korisnik prijaviti (formi za prijavljivanje može pristupiti iz navigacionog menija) i to predstavlja prvi slučaj korишćenja.

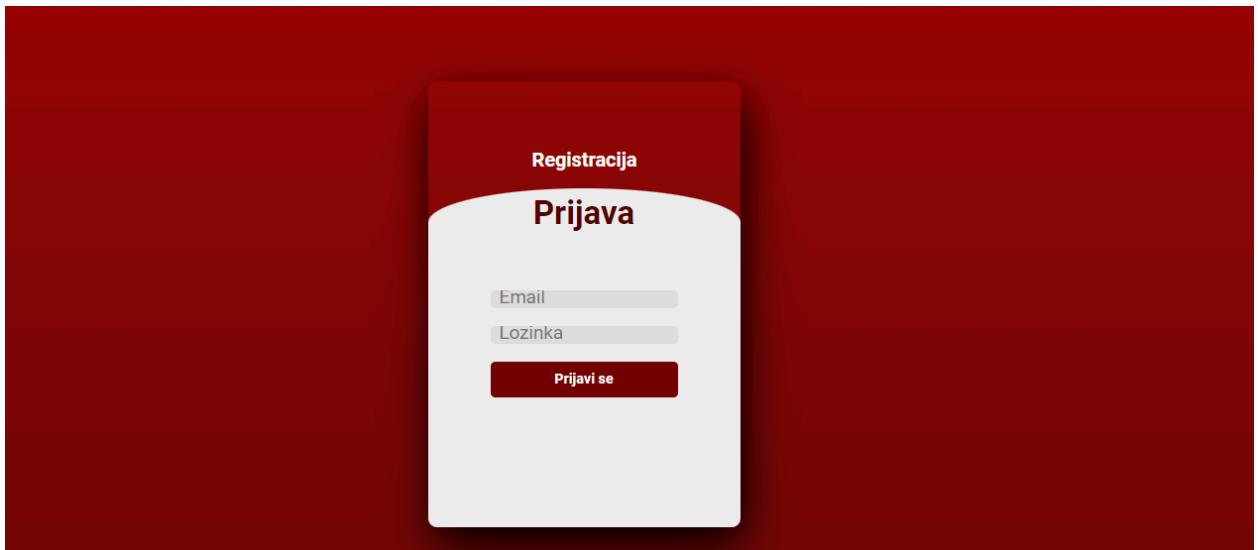
3.2.4.4 CK1: Slučaj korишćenja - Prijavljivanje igrača na sistem

Назив СК : Пријављивање играча

Актори СК : Играч

Учесници СК : Играч и систем (програм)

Предуслов: Систем је покренут и отворена је форма за пријављивање.



Основни сценарио:

- 1.Играч уноси корисничко име и лозинку(АПУСО)
- 2.Кликом на дугме Пријави се играч позива систем да изврши пријављивање. (АНСО)
- 3.Систем проверава унешене податке (СО)
- 4.Систем приказује играчу поруку да је успешно пријављен на систем и отвара главну форму програма. (ИА)

Dobrodošli u bazu aktuelnih šahovskih turnira!

Početna Prijava se Svi turniri Moji Turniri

Dobrodošli na naš blog!
Posted on January 1, 2023 by ŠahovskiTurnir

sponsori saradnici



Šahovske lekcije

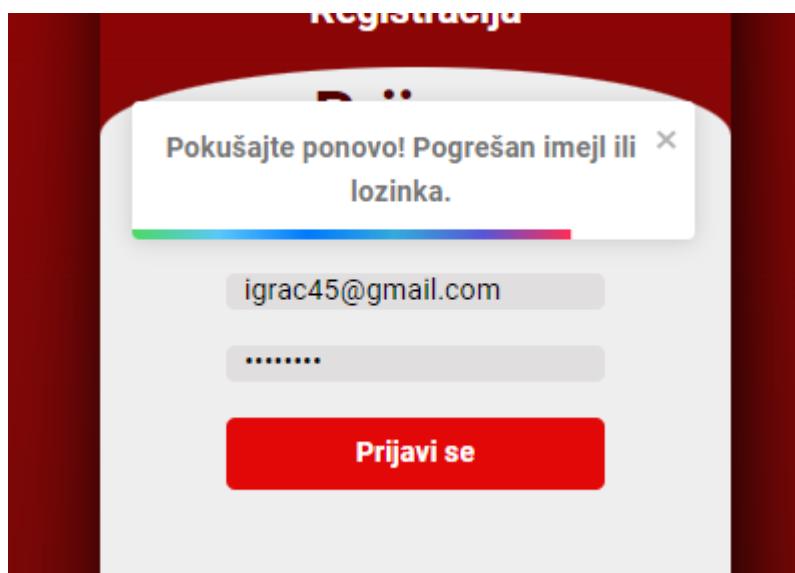
- [Otvaranja](#)
- [Završnice](#)
- [Središnica](#)
- [Otvorene igre](#)
- [Poluotvorene igre](#)
- [Zatvorene igre](#)
- [Hoće Raul Capablanca](#)
- [Mikhail Tal](#)
- [Roberts James Fischer](#)

Najnoviji tekstovi

Izabran predsednik novog šahovskog saveza Beograda

Алтернативни сценарио:

- 4.1 Систем приказује играчу поруку да није унео одговарајуће параметре.



3.2.4.5 СК2: Случај коришћења - Регистровање новог играча

Уколико корисник нема отворен налог, то значи да његови подаци не постоје у бази играча и неће моћи да се пријави док не направи налог. Прелазак на страницу за регистрацију врши се кликом на део екрана у ком пише регистрација. Ту корисник уноси своје податке: корисничко име, имејл адресу и лозинку.



Назив СК : Регистрација новог играча

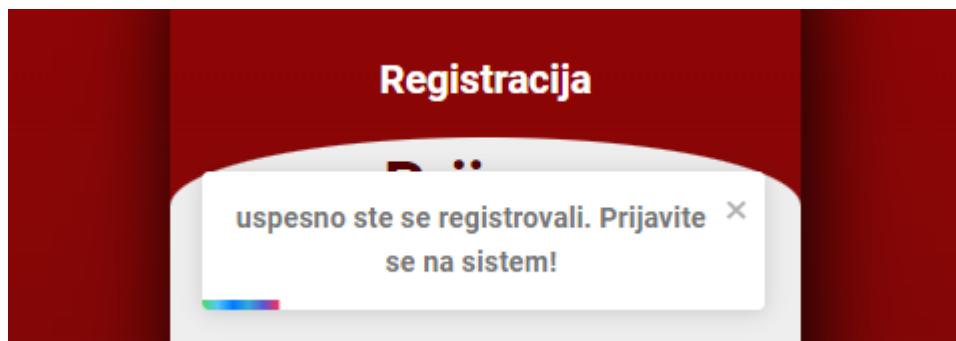
Актори СК : Играч

Учесници СК : Играч и систем (програм)

Предуслов: Систем је укључен и приказује форму за уношење података о играчу.

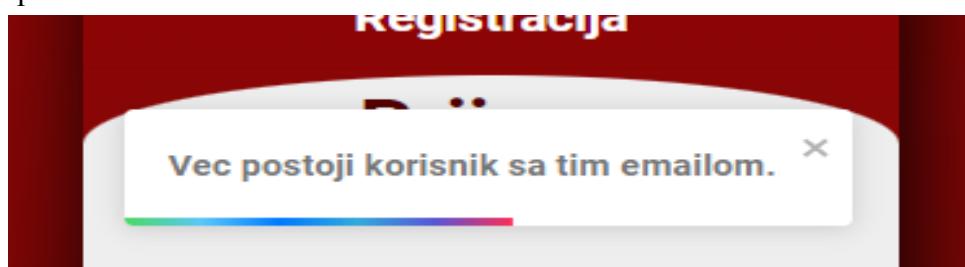
Основни сценарио:

- 1.Играч уносује своје податке (АПУСО)
- 2.Играч контролише да ли је коректно унео податке (АНСО)
- 3.Играч позива систем да запамти податке (АПСО)
- 4.Систем креира новог играча и памти податке о њему (СО)
- 5.Систем приказује играчу поруку да је његове податке(ИА)



Алтернативни сценарио:

- 5.1. Уколико систем не може да запамти податке о новом играчу он приказује поруку грешке.



3.2.4.6 СК3: Случај коришћења - Претрага турнира

Најзад, ако су подаци тачно унешени и налазе се у бази, корисник се преусмерава на почетну страницу. Одатле из навигационог менија може приступити осталим страницама. Прва међу њима је “Сви турнири” на којој се налази табела са турнирима.

Početna	Prijavi se	Svi turniri	Moji Turniri				
Registarski broj	Turnir	Mesto	Tip	Tempo	Prijava		
1	Prvenstvo Beograda	Matematička gimnazija	neređingovan	redovan	Pridruži se turniru		
2	Kategoriji za osvajanje MK	SK Vidikovac	kategoriji	redovan	Pridruži se turniru		
3	Letnji Open	Restoran Balasević	ređingovan	redovan	Pridruži se turniru		
4	Prvenstvo Srbije	Hotel Fontana	neređingovan	redovan	Pridruži se turniru		
5	Atomac	Prostorije novog sahovskog saveza	ređingovan	brzopotezni	Pridruži se turniru		
7	Školsko ubrzano prvenstvo	OS Pavle Savic	neređingovan	ubrzani	Pridruži se turniru		
8	Memorijalni turnir Gliga	Gimnazija Batajnica	ređingovan	redovan	Pridruži se turniru		
9	Ekipno Klupsko	Lestane opština	neređingovan	ubrzani	Pridruži se turniru		
10	Zenska liga	Stari Dif	ređingovan	redovan	Pridruži se turniru		
11	Prva liga Beograda	Stadion Obilić	ređingovan	redovan	Pridruži se turniru		
12	Kategoriji za osvajanje druge kategorije	SK Čukaricki	kategoriji	ubrzani	Pridruži se turniru		
13	Kategoriji za osvajanje prve kategorije	SK Crvena Zvezda	kategoriji	brzopotezni	Ovaj turnir je završen.		
14	Božićni turnir	Hotel Slavija	neređingovan	ubrzani	Pridruži se turniru		
15	Prvenstvo kluba	SK Kralj	neređingovan	ubrzani	Pridruži se turniru		
25	Uskrsnji turnir	SK Lasta	ređingovan	brzopotezni	Pridruži se turniru		
26	Prvenstvo Čukarice	Bilić klub Garaza	neređingovan	brzopotezni	Pridruži se turniru		
28	Dani misaonih igara	Kalemegdan	neređingovan	brzopotezni	Pridruži se turniru		
29	Svetski kup	Beogradski sajam	ređingovan	redovan	Pridruži se turniru		
30	Kup šampiona Evrope	Hotel Hajat	ređingovan	redovan	Pridruži se turniru		

Prijavite se brzo i jednostavno na svaki
SahovskiTurnir

[Početna](#) [Blog](#) [O nama](#) [Kontakt](#)

ŠahovskiTurnir © 2023

[Email](#)

[Poruka](#)

Activate Windows
Go to Settings to activate Windows
[Poslati](#)

Овде се налази списак предстојећих и недавно завршених турнира. Уколико је турнир завршен (турнир 13 на слици) то је назначено у последњој колони табеле. Поље за претрагу приказује се изнад табеле и у њега се може унети критеријум за претрагу што представља следећи случај коришћења.

Назив СК : Претрага турнира

Актори СК : Играч

Учесници СК : Играч и систем (програм)

Предуслов: Систем је укључен и играч је улогован под својом шифром. Систем приказује форму за претрагу турнира.

Основни сценарио СК:

1.Играч уноси вредност критеријума за претрагу турнира по називу (АПУСО)

Pretraži turnire po nazivu					
Registarski broj	Turnir	Mesto	Tip	Tempo	Prijava
1	Prvenstvo Beograda	Matematička gimnazija	nerejtingovan	redovan	<input type="button" value="Pridruži se turniru"/>
2	Kategorini za osvajanje MK	SK Vidikovac	kategorini	redovan	<input type="button" value="Pridruži se turniru"/>
3	Letnji Open	Restoran Balasević	rejtingovan	redovan	<input type="button" value="Pridruži se turniru"/>
4	Prvenstvo Srbije	Hotel Fontana	nerejtingovan	redovan	<input type="button" value="Pridruži se turniru"/>

2. Систем тражи турнире по задатој вредности (CO)

3. Систем враћа турнире по задатој вредности (IA)

prvens					
Registarski broj	Turnir	Mesto	Tip	Tempo	Prijava
1	Prvenstvo Beograda	Matematička gimnazija	nerejtingovan	redovan	<input type="button" value="Pridruži se turniru"/>
4	Prvenstvo Srbije	Hotel Fontana	nerejtingovan	redovan	<input type="button" value="Pridruži se turniru"/>
7	Školsko ubrzano prvenstvo	OS Pavle Savić	nerejtingovan	ubrzani	<input type="button" value="Pridruži se turniru"/>
15	Prvenstvo kluba	SK Kralj	nerejtingovan	ubrzani	<input type="button" value="Pridruži se turniru"/>
26	Prvenstvo Cukarice	Biljari klub Garaza	nerejtingovan	brzopotezni	<input type="button" value="Pridruži se turniru"/>

Prijavite se brzo i jednostavno na svaki
Šahovski Turnir
Početna | Raspored | O nama | Kontakt

[Kontaktirajte nas](#) [Email](#)

Не постоји алтернативни сценарио с обзиром на то да у случају уноса критеријума за ког нема резултата или неправилног критеријума, систем неће урадити ништа.

3.2.4.7 СК4: Случај коришћења - Пријава на турнир

Уколико није још почео, играч се на турнир може пријавити кликом на дугме “придружи се турниру”. Систем ће приказати поруку о успешној пријави (чак и уколико постоји нека неправилност у пријави, нпр да је играч већ пријављен за тај турнир - то се решава на серверској страни).

11	Prva liga Beograda	Stadion Obilić	rejtingovan	redovan	<input type="button" value="Pridruži se turniru"/>
12	Kategorini za osvajanje druge kategorije	SK Cukaricki	kategorini	ubrzani	<input type="button" value="Pridruži se turniru"/>
13	Kategorini za osvajanje prve kategorije	SK Crvena Zvezda	kategorini	brzopotezni	Ovaj turnir je završen.
14	Božićni turnir	Hotel Slavija	nerejtingovan	ubrzani	<input type="button" value="Pridruži se turniru"/>

Назив СК: Пријава на турнир

Актори СК: Играч

Учесници СК: Играч и систем (програм)

Предуслов: Систем је укључен и играч је улогован под својом шифром. Систем приказује форму за рад са турнирима. Учитана је листа турнира.

Основни сценарио СК

- Играч бира неки од приказаних турнира. (АПУСО)
- Играч кликом на дугме поред назива турнира позива систем да креира пријаву. (АПСО)
- Систем креира пријаву. (CO)

4. Систем приказује играчу податке о пријави и поруку “Успешно сте се пријавили на турнир!.” (ИА)

Dobrodošli u bazu aktuelnih šahovskih turnira!

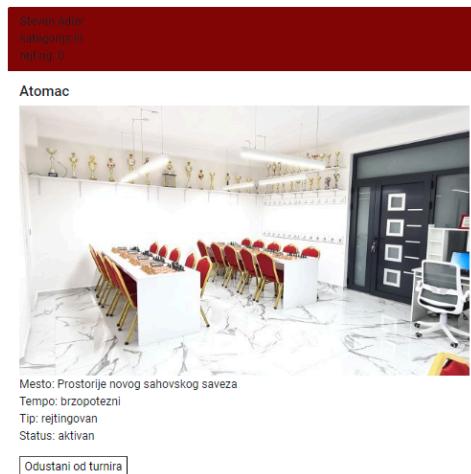
Početna	Prijavi se	Svi turniri	Moji Turniri	Uspešno ste se prijavili na turnir!		
Registarski broj	Turnir	Mesto	Tip	Tempo	Prijava	
1	Prvenstvo Beograda	Matematička gimnazija	nerejtingovan	redovan	Pridruži se turniru	
2	Kategorini za osvajanje MK	SK Vidikovac	kategorini	redovan	Pridruži se turniru	
3	Letnji Open	Restoran Balasevic	rejtingovan	redovan	Pridruži se turniru	
4	Prvenstvo Srbije	Hotel Fontana	nerejtingovan	redovan	Pridruži se turniru	

Алтернативна сценарија

4.1 Уколико систем не може да креира пријаву, он показује судији поруку “систем не може да креира пријаву”. (ИА)

3.2.4.8 СК5: Случај коришћења - Одјављивање са турнира

На страници “Моји турнири” налазе се турнири на које је играч пријављен (и они које је већ одиграо уколико се налазе и у табели свих турнира).



Турнири су приказани као картице са детаљима о сваком. Кликом на дугме “Одустани од турнира” брише се пријава датог играча на одабрани турнир и он добија поруку о томе да је то извршено.

Назив СК: Одјава са турнира

Актори СК: Играч

Учесници СК: Играч и систем (програм)

Предуслов: Систем је укључен и играч је улогован под својом шифром. Систем приказује форму за рад са турнирима. Учитана је листа турнира на које је пријављен улоговани играч.

Основни сценарио СК

- Играч бира неки од приказаних турнира. (АПУСО)
- Играч кликом на дугме поред назива турнира позива систем да обрише пријаву. (АПСО)

3. Систем брише пријаву. (СО)

4. Систем приказује играчу поруку “Ваша пријава је успешно обрисана!.” (ИА)

Steven Adler
kategorija III
rejting: 0

Skolsko ubrzano prvenstvo

Steven Adler
kategorija III
rejting: 0

Zenska liga

Steven Adler
kategorija III
rejting: 0

Kategorjni za osvajanje MK

Obrisali ste prijavu.

1

?

ШАХОВСКИ КЛУБ "ВИДИКОВАЦ"
БЕОГРАД

Mesto: OS Pavle Savic
Tempo: ubrzani
Tip: nerejtingovan
Status: aktivan

Mesto: Stari Dif
Tempo: redovan
Tip: rejtingovan
Status: aktivan

Mesto: SK Vidikovac
Tempo: redovan
Tip: kategorjni
Status: aktivan

Oduštani od turnira

Oduštani od turnira

Oduštani od turnira

Уколико се ради о турниру који је играч већ одиграо, кликом на дугме “Ваш резултат” он добија поруку о томе које је пласман остварио.

Steven Adler
kategorija III
rejting: 0

Kategorjni za osvajanje prve kategorije

ШК

Mesto: SK Crvena Zvezda
Tempo: brzopotezni
Tip: kategorjni
Status: zavrsen

Steven Adler
kategorija III
rejting: 0

Prvenstvo Cukarice

Niste osvojili nista na ovom turniru.

Principe

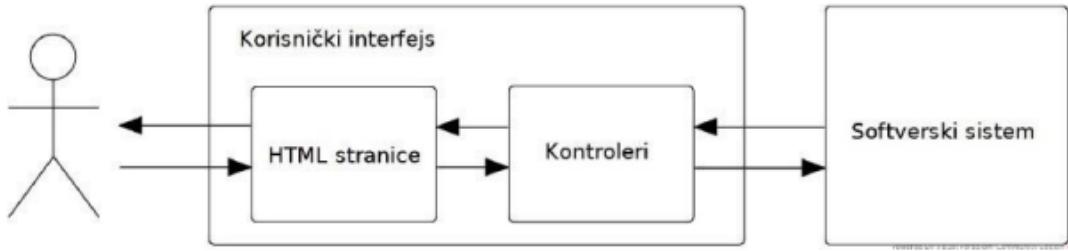
Алтернативна сценарија

4.1 Уколико систем не може да обрише пријаву, он показује судији поруку “систем не може да обрише пријаву”. (ИА)

3.2.4.9 Пројектовање контролера корисничког интерфејса

Контролер корисничког интерфејса је одговоран за:

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



Слика 21. Контролер корисничког интерфејса

3.2.4.10 Пројектовање апликационе логике

Апликациони сервери треба да обезбеде сервисе који ће омогућити реализацију апликационе логике софтверског система. Пројектовани апликациони сервер садржи:

- Део за комуникацију са клијентом
- Контролер апликационе логике
- Део за комуникацију са складиштем података (у овом случају то представља класа контролер која је надкласа сваког контролера који је пројектован и наслеђује класу BaseController генерисану од стране Ларавела која комуницира са базом)
- Део који садржи пословну логику

3.2.4.11 Комуникација са клијентима

Frontend шаље http захтев backend-у путем axios библиотеке и тако индиректно комуницира са базом података. Backend садржи контролере који комуницирају са базом (јер наслеђују класу коју је ларавел аутоматски генерисао и која служи за комуникацију са базом) па се самим тим и са клијентске и са серверске стране врше промене над базом и узимају благовремено ажурирани подаци.

API руте дефинисане на backend-у уз позивање преко контролера за сваку класу:

```

Route::post('/register', [AuthController::class, 'register']);
Route::post('/login', [AuthController::class, 'login']);
Route::post('/logout', [AuthController::class, 'logout']);
Route::resource('turnir', TurnirController::class)->only(['index', 'show']);
Route::resource('igrac', IgracController::class)->only(['index', 'show', 'store']);
Route::resource('prijava', PrijavaController::class)->only(['index', 'store', 'destroy']);
Route::resource('klub', KlubController::class)->only(['index']);
Route::resource('igrac/{id}/prijava', IgracTurniriController::class)->only(['index']);
Route::resource('prijava/{id}/igrac', IgracPrijavaController::class)->only(['index']);

```

У сваком од контролера налазе се методе помоћу којих се врше CRUD операције.

На примери табеле играч, оне су овако имплементиране у контролеру:

```

class IgracController extends Controller
{
    public function index()
    {
        $igrac = Igrac::all();
        return IgracResource::collection($igrac);
    }

    public function show(Igrac $igrac)
    {
        return new IgracResource($igrac);
    }

    public function store(Request $request)
    {
        $igrac = Igrac::create([
            'ime' => $request->ime,
            'prezime' => $request->prezime,
            'klub' => $request->klub,
            'kategorija' => $request->kategorija,
            'rejting' => $request->rejting,
            'password' => $request->password,
            'email' => $request->email
        ]);

        $igrac->save();

        return response()->json(['Igrac je uspesno kreiran.', new
IgracResource($igrac), 'success' => true]);
    }
}

```

Као што се може приметити, ова класа наслеђује класу Controller која повезује апликацију са базом.

```

<?php

namespace App\Http\Controllers;

use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
use Illuminate\Foundation\Validation\ValidatesRequests;
use Illuminate\Routing\Controller as BaseController;

class Controller extends BaseController

```

```
{
    use AuthorizesRequests, ValidatesRequests;
}
```

Пример слања захтева са frontend-а (играч жели да поништи своју пријаву на турнир):

```
function obrisi (){
    const FormData = require('form-data');
let data = new FormData();

let config = {
    method: 'delete',
    maxBodyLength: Infinity,
    url: 'http://127.0.0.1:8000/api/prijava/'+prijava.PrijavaID,
    data : data
} ;
```

3.2.4.12 Пројектовање контролера апликационе логике

Након слања захтева за извршење неке системске операције од стране клијента, Frontend шаље HTTP захтев backend-у који га преусмерава захтев до контролера одговорних за комуникацију са базом података и извршење системских операција путем API путање. Резултат извршења системске операције враћа се до апликационе логике и затим приказује на страници web апликације.

3.2.4.13 Пројектовање пословне логике

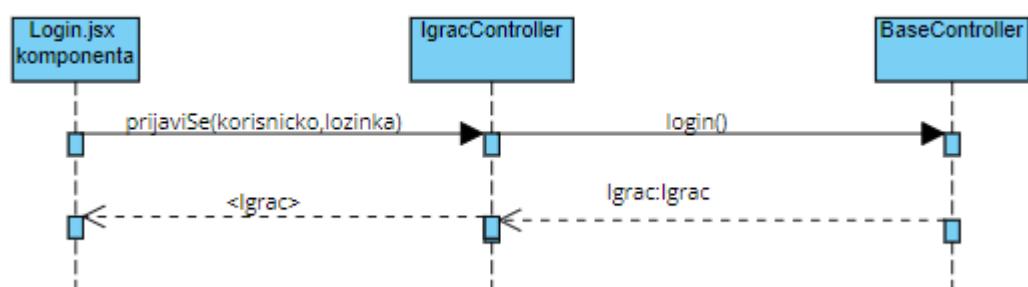
Да би се пројектовало понашање система, потребно је за сваки од претходно дефинисаних уговора направити системску операцију.

Уговор УГ1: Signal izvršiPrijavuNaSistem()

Веза са СК: СК1

Предуслови:

Постуслови:

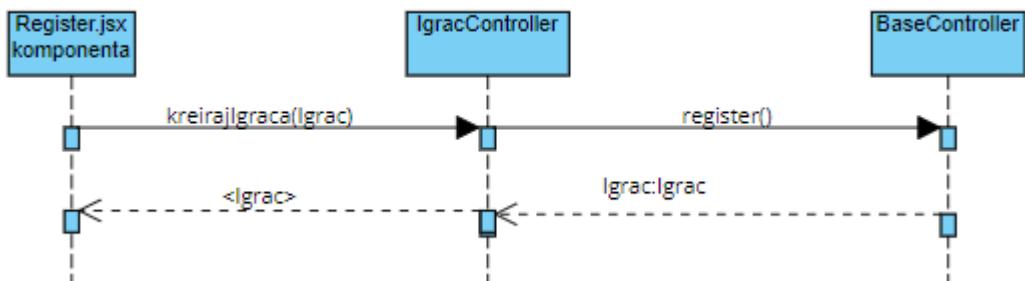


Уговор УГ2:Signal kreirajIgraca(Igrac)

Веза са СК: СК2

Предуслови: Вредносна и структурна ограничења над објектом Igrac морају бити задовољена.

Постуслови: Подаци о играчу су запамћени.

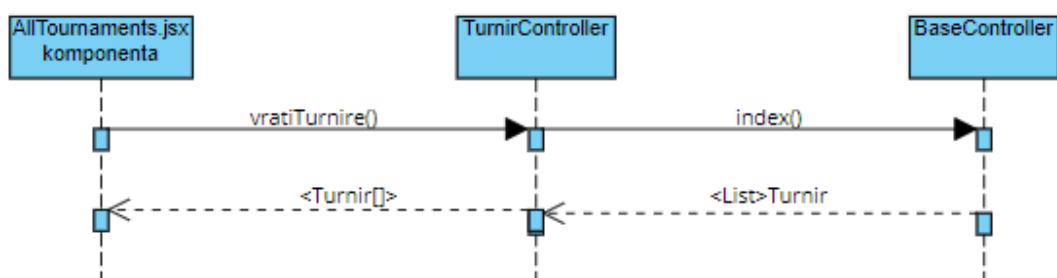


Уговор УГ3: signal ucitajListuTurnira(<List> Turnir)

Веза са СК: СК3

Предуслови:

Постуслови:

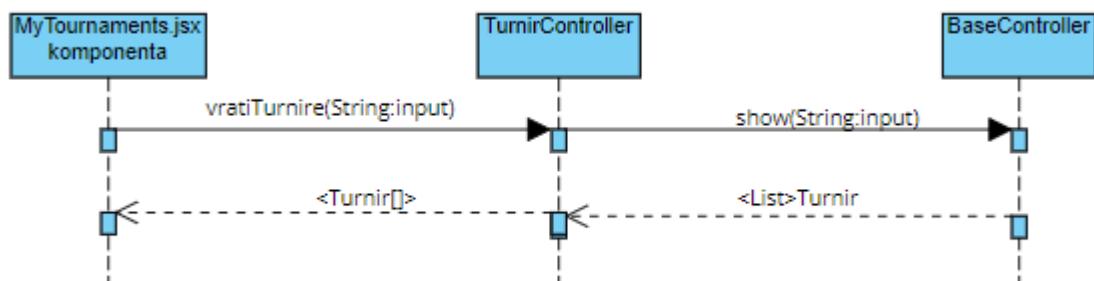


Уговор УГ4:Signal pronadjiTurnire (<List> Turnir, Turnir)

Веза са СК: СК3

Предуслови:

Постуслови:

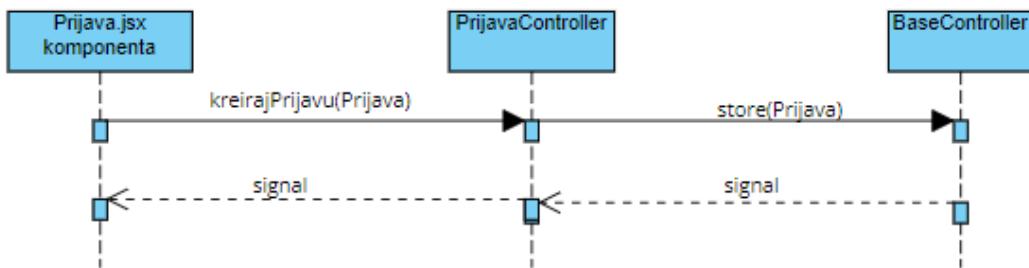


Уговор УГ5:Signal kreirajPrijavu(Prijava)

Веза са СК: СК4

Предуслови: Вредносна и структурна ограничења над објектом Prijava морају бити задовољена.

Постуслови: Подаци о колу су запамћени.



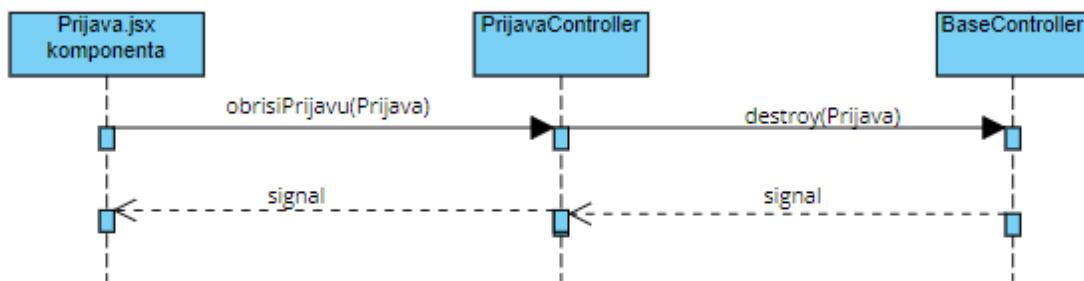
Уговор УГ6:Signal obrisiPrijavu(Prijava)

Веза са СК: СК5

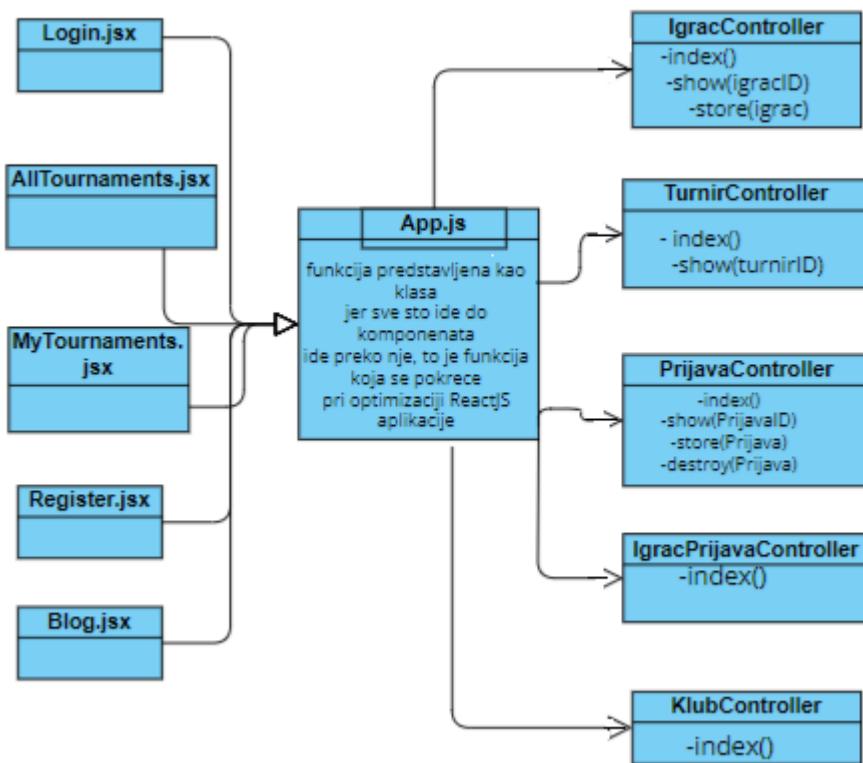
Предуслови: Ако је пријава обрисана не може се извршити системска операција.

Структурна ограничења над објектом Пријава морају бити задовољена.

Постуслови: Пријава је обрисана



Након дефинисања системских операција за сваки од уговора, следи приказ имплементације системских операција:



Слика 22.имплементација системских операција

App.js је заправо функција, али овде је представљена као класа јер она у ствари представља целу ReactJS апликацију. Све методе извршавају се унутар ове функције којој такође и стижу сви подаци из базе пре него што их она проследи компонентама. Дакле, уместо могао је писати назив react апликације.

3.2.4.14 Пројектовање структуре софтверског система

Класе које одговарају табелама из базе наслеђују класу Model која је аутоматски генерисана од стране Laравела. У њим се функцијама дефинишу односи између табела и атрибути.

```

class Igrac extends Model
{
    use HasFactory;

    protected $table = 'igrac';
    public $primaryKey = 'IgracID';
    protected $fillable = [
        'ime',
        'prezime',
        'klub_id',
    ]
  
```

```

        'kategorija',
        'rejting',
        'email',
        'password'
    ] ;

    public function klub()
    {
        return $this->belongsTo(Klub::class);
    }
    public function prijava()
    {
        return $this->hasMany(Prijava::class);
    }
}

```

Ознаке:

Protected \$table - ово поље означава име табеле из базе којој одговара класа

Public \$primaryKey - примарни кључ

Protected \$fillable - атрибут који говори о томе да су вредности ових атрибута променљиве

Функције испод говоре о односу између класа. Конкретно у овом случају, приказана је припадност играча клубу, односно да играч има спољни кључ ка табели клуб и да један играч може имати више пријава тј да табела пријава садржи спољни кључ ка табели играч.

3.2.4.15 Пројектовање складишта података

На основу релационог модела и ограничења пројектоване су табеле базе података које користи се користе у овом модулу. Све ове табеле коришћене су и у првом модулу где је приказана њихова структура, па ће овде бити приказани подаци.

Табела играч:

	IgracID	KlubID	Ime	Prezime	Kategorija	Rejting	email	password
	0	1	Steven	Adler	III		0 player0@gmail.com	11111111
	1	1	Axl	Rose	MK	1987	player1@gmail.com	11111111
	2	2	Slash	/	II	1783	player2@gmail.com	11111111
	3	3	Duff	McKagan	I	1900	player3@gmail.com	11111111
	4	4	Izzy	Stradlin	II	1676	player4@gmail.com	11111111
	5	2	James	Hetfield	IV		0 player5@gmail.com	11111111
	6	3	Lars	Ulrich	MK	2200	player6@gmail.com	11111111
	7	4	Kirk	Hammet	FM	2500	player7@gmail.com	11111111
	8	3	Robert	Trujillo	I	1800	player8@gmail.com	11111111
	9	3	Myles	Kennedy	IV		0 player9@gmail.com	11111111
	11	2	Ozzy	Osbourne	MK	1900	player10@gmail.com	11111111
	12	4	RonnieJames	Dio	III	1567	player11@gmail.com	11111111
	13	5	Tom	Ciljan	IV		0 player12@gmail.com	11111111

Табела турнир:

TurnirID	Naziv	Mesto	Tip	Tempo	SudijaID	Status	Slika
1	Prvenstvo Beograda	Matematicka gimnazija	nerejtingovan	redovan	1	aktivran	https://j
2	Kategorjni za osvajanje MK	SK Vidikovac	kategorjni	redovan	2	aktivran	https://j
3	Letnji Open	Restoran Balasevic	rejtingovan	redovan	3	aktivran	https://j
4	Prvenstvo Srbije	Hotel Fontana	nerejtingovan	redovan	4	aktivran	https://j
5	Atomac	Prostorije novog sahovskog saveza	rejtingovan	brzopotezni	1	aktivran	https://j
7	Školsko ubrzano prvenstvo	OS Pavle Savic	nerejtingovan	ubrzani	3	aktivran	https://j
8	Memorijalni turnir Gliga	Gimnazija Batajnica	rejtingovan	redovan	4	aktivran	https://j
9	Ekipno Klupsko	Lestane opština	nerejtingovan	ubrzani	3	aktivran	https://j
10	Zenska liga	Stari Dif	rejtingovan	redovan	2	aktivran	data:image
11	Prva liga Beograda	Stadion Obilić	rejtingovan	redovan	1	aktivran	https://n
12	Kategorjni za osvajanje druge kategorije	SK Cukaricki	kategorjni	ubrzani	4	aktivran	https://j
13	Kategorjni za osvajanje prve kategorije	SK Crvena Zvezda	kategorjni	brzopotezni	2	zavrsen	https://j

Табела пријава:

PrijavaID	TurnirID	IgracID	updated_at	created_at
21	2	17	(NULL)	(NULL)
22	2	26	(NULL)	(NULL)
23	2	3	2023-05-13 17:58:20	2023-05-13 17:58:20
24	2	3	2023-05-14 20:38:59	2023-05-14 20:38:59
25	12	6	2023-05-14 21:44:21	2023-05-14 21:44:21
26	2	3	2023-05-14 21:46:38	2023-05-14 21:46:38
28	13	0	2023-05-14 21:47:48	2023-05-14 21:47:48

Табела клуб:

KlubID	Naziv	Grad
1	Vidikovac	Beograd
2	Kralj	Beograd
3	Cukaricki	Novi Sad
4	Lasta	Bar
5	Crvena Zvezda	Beograd
6	Partizan	Beograd
7	Delfin	Bar
8	Tadic	Nis
9	Mala Rokada	Uzice
10	Nikola Tesla	Uzice
11	Nestor	Podgorica
12	Kapablanka	Niksic

3.2.4.16 Комплетна архитектура софтверског система

Идентификоване класе на основу класа из концептуалног модела су:

- Играч
- Турнир
- Пријава

Осим њих, у модул су додате и следеће класе:

1. На frontend-y:

Компоненте које представљају приказе појединачних страница или делова страница видљивих кориснику. Компоненте нису везане за конкретне класе већ се у оквиру њиховог потписа позивају функције које учитавају објекте класа и повезују их са ентитетима дефинисаним у компонентама. Додате компоненте су странице са JSX екстензијом JavaScript-а које омогућавају да се заједно користе JavaScript и HTML (у

овом случају и PHP). Неке од компонената су reusable (NavBar, Footer) што значи да се идентичне јављају на више single-page страница.

2. На backend-у:

Класе колекција, ресурса и контролера за сваку од основних класа које се у ларавелз имплементирају као модели. На примеру класе клуб:

Collection:

```
class KlubCollection extends ResourceCollection
{
    /**
     * Transform the resource collection into an array.
     *
     * @return array<int|string, mixed>
     */
    public static $wrap = 'klub';

    public function toArray(Request $request): array
    {
        return parent::toArray($request);
    }
}
```

Ова класа служи да објекте класе Клуб враћа као низове. \$wrap представља ознаку којом се приступа подацима елемената низа.

Resource:

```
class KlubResource extends JsonResource
{
    /**
     * Transform the resource into an array.
     *
     * @return array<string, mixed>
     */
    public static $wrap = 'klub';

    public function toArray(Request $request): array
    {
        return [
            'KlubID' => $this->resource->KlubID,
            'Naziv' => $this->resource->Naziv,
            'Grad' => $this->resource->Grad,
        ];
    }
}
```

Класа KlubResource дефинише у ком облику ће се кориснику при позивању враћати подати о објекту класе Клуб. Упрошћено се може рећи да она служи за форматирање података.

Controller:

```
class KlubController extends Controller
{
    public function index()
    {
        $klub = Klub::all();
        return KlubResource::collection($klub);
    }
}
```

Beđ je bilo речи о улози контролерских класа у овом модулу. У њима се налазе функције којима се извршавају основне операције над подацима из базе. На примеру се види повезаност трију наведених типова класа. (подаци из табеле Клуб из базе враћају се као низ док сваки од њих такође представља низ података који чине његови атрибути и бивају враћени у формату наведеном у ресурс класи).

Специфични контролери за сложене случајеве коришћења:

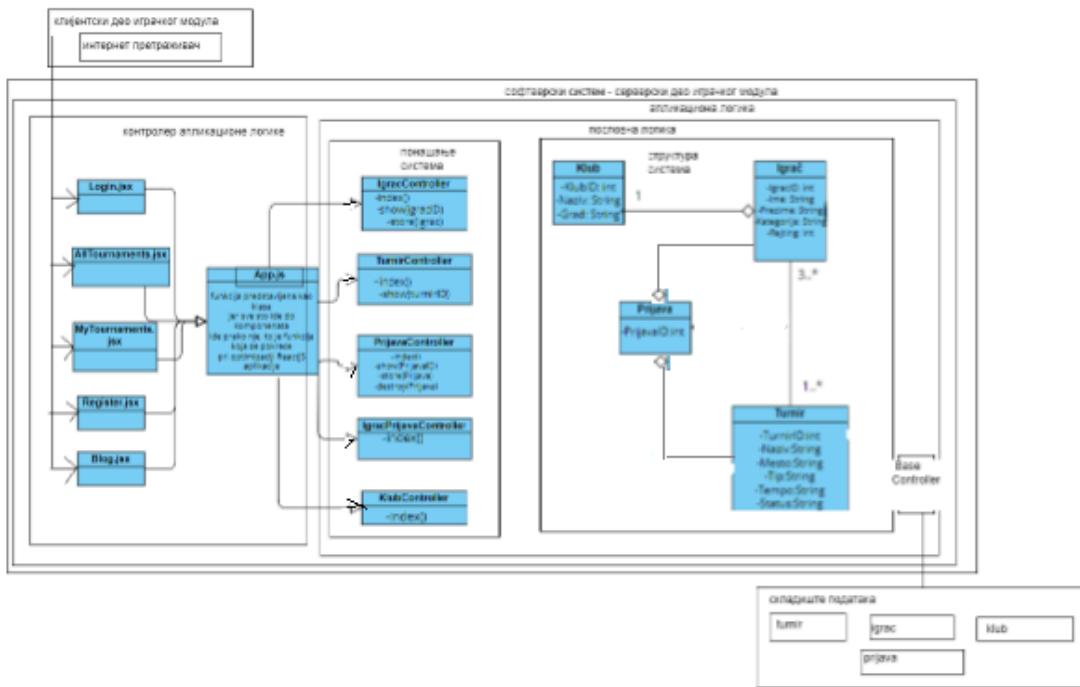
```
class IgracPrijavaController extends Controller
{
    public function index($id)
    {
        $prijave = Prijava::get()->where('IgracID', $id);

        if (count($prijave) == 0) {
            return response()->json('No data!', 404);
        }

        return new PrijavaCollection($prijave);
    }
}
```

Ова класа служи да врати све пријаве једног играча. Њен контролер позива се као ruta независна од контролера класе играч и контролера класе пријава појединачно.

Сада су објашњене све компоненте потребне за приказ целокупне архитектуре софтверског система:



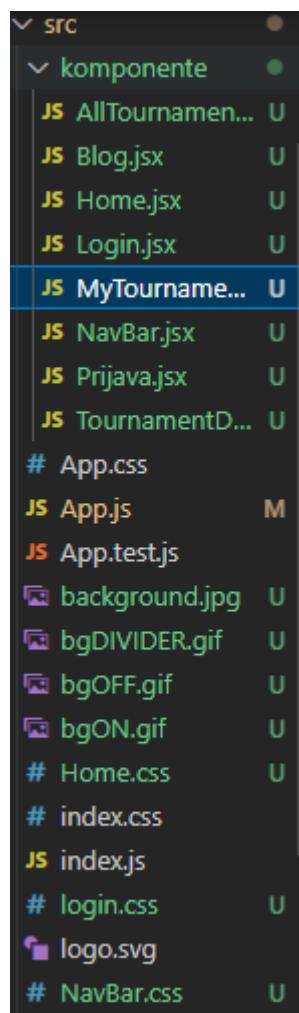
Слика 23. Архитектура софтверског система

3.2.5 Имплементација

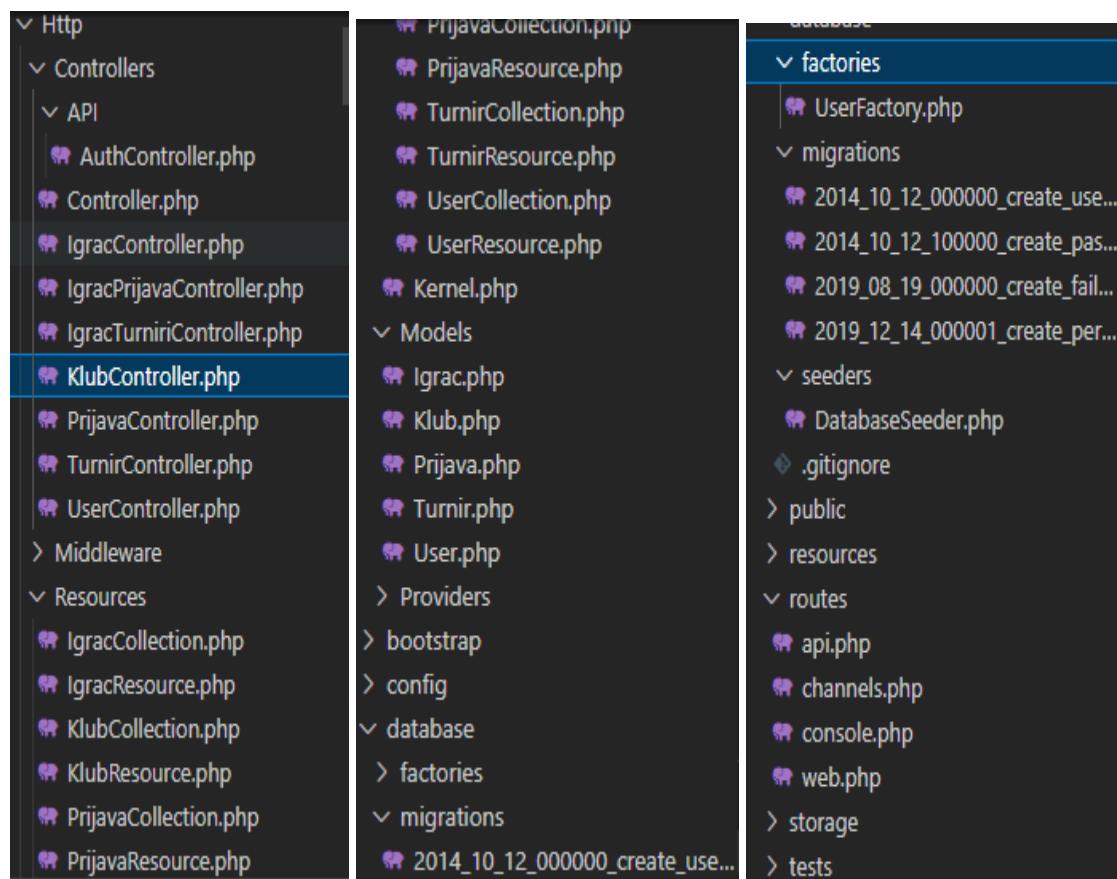
Софтверски систем креиран као овај пројекат је развијен у програмским језицима JavaScript и PHP и пројектован као клијент - сервер апликација. Као развојно окружење коришћен је Visual Studio code, а као систем за управљање базом података коришћен је MySQL сервер.

Читав систем је реализован као два пројекта, један везан за frontend, а други за backend.

Frontend:



Backend:



3.2.6 Тестирање

Тестирање је обављано ручно. ReactJS на frontend-у комуницира са backend-ом преко HttpClient-а. У терминалу у Visual Studio Code-у је укуцано прт start који покреће пројекат. Приступљено је веб апликацији путем линка <http://localhost:3000> који је укуцан на browser-у где се отвара почетна страница и тестирају се све функционалности које апликација нуди.

Пројекат је покренут на backend-у и уписан је линк <http://localhost:8000/api> у апликацији Postman или на browser-у, где су тестиране све методе чије смо позиве дефинисали у Laravel-у у окружењу Visual Studio Code.

4. ЗАКЉУЧАК

С обзиром на то да је програмски језик Јава платформски независан и можда најсличније конципиран људском размишљању међу свим програмским језицима, јасно је зашто се многи одлучују за њега при креирању апликација најразличитијих намена. То је случај и са овим радом у ком је коришћена поједностављена Ларманова метода за развој софтвера.

Одлуку о теми рада било је врло једноставно донети јер сам одавно хтела да направим нешто слично овоме. Бавим се шахом већ дugo и између осталог сам и судија, па сам имала прилике да увидим колико је потребно имати добар програм за паровање и што мање ствари радити ручно.

Као што се већ много пута испоставило у пракси, софтверска решења значајно доприносе убрзању и повећању ефективности конвенционалних процеса, што је показао и овај рад. Једном дефинисане функције за одређивање критеријума паровања могу се примењивати у било којој ситуацији и по било ком правилнику организације шаховских турнира, а у већини случајева и на одређивање жреба у другим спортивима. Део апликације за претрагу по разним критеријумима и измене над подацима је такође користан јер тренутно не постоји актуелан програм везан за шах који обухвата све наведено - одвојено се користи база играча, база партија, сајт са новостима, програм за паровање (све то је у овом раду обједињено) док онлајн пријављивање на турнире још није почело да се практикује, осим у изузетним случајевима када су у питању интерни турнири клубова или турнири мањих капацитета, па се с те стране играчки модул ове апликације може сматрати иновативним.

Правац у ком би требало унапређивати овај рад је пре свега у тестирању. Ручно тестирање је превазиђено како се технологије развијају и представља непотребан губитак времена, па самим тим и смањење ефикасности у тој фази. Требало би радити на развоју аутоматизованих тестова и прављења класа за тестирање сваког од модула. Тестирање се у почетку може вршити над насумичним базама података, а онда би се морало извршити и са одговарајућом како би се проверило евентуално одступање од контекста.

Један од циљева који би у случају наставка рада на овом пројекту морао бити испуњен је постизање независности модула. Да би сваки од модула имао примену без оног другог, потребно је увести коришћење додатних технологија. За побољшање модула за вођење турнира који користе судије требало би користити неку од технологија као што је Spring оквир. Креирањем REST API у Javi у овом модулу, била би могућа директна комуникација са фронтендом, па у играчком модулу не би морао да постоји Ларавел, нити било који други начин комуницирања са базом. Тиме би судијски модул остао клијент-сервер апликација у којој клијента представља судија који користи програм, а играчки модул би био само ReactJS апликација која би директно комуницирала са судијским модулом.

Уколико се пође од модула за пријављивање на турнир за играче, постоје недостаци везани за архитектуру. Требало би имплементирати контролерске класе и на серверској страни програма (уз претпоставку да потреба за њом још постоји, тј да претходни предлог унапређења није урађен).

Степен развоја софтверског система који је достигнут у овом раду не обухвата никакве методе заштите (осим основне мере чувања података о конекцији са базом у properties фајлу, а не у некој од класа програма и hash-овању лозинке и у једном и у другом модулу) па је то још један од праваца у ком би даљи развој требало да тече. Потребно је заштитити софтвер пре свега од вируса, а затим и од потенцијалних покушаја упада појединачних хакера.

Комбинација технологија коришћених у раду можда није оптимална, али је одличан показатељ прилагођавања различитих алата намени система. Свака од технологија употребљена је баш за део система за који се њена спецификација сматра одговарајућом. Чињеница је да постоји још доста простора за унапређивање, али и ова верзија софтверског система у потпуности је применљива у сврхе у које је и замишљена, што је одлична основа за даље истраживање, развој и проширење скупа области примене исте.

Литература

- [1] Gosling, J., Holmes, D. C., & Arnold, K. (2005). The Java programming language. https://hmong.ru/wiki/Java_language
- [2] Baxter, G., Frean, M., Noble, J., Rickerby, M., Smith, H., Visser, M., ... & Tempero, E. (2006, October). Understanding the shape of Java software. In Proceedings of the 21st annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications (pp. 397-412). <https://dl.acm.org/doi/abs/10.1145/1167473.1167507>
- [3] Baeldung, <https://www.baeldung.com/ant-maven-gradle>, Јун 2020.
- [4] Anthony Heedings, 2020, What is Laravel and how do you get started with it? <https://www.howtogeek.com/devops/what-is-laravel-and-how-do-you-get-started-with-it/>
- [5] Bootstrap, <https://getbootstrap.com>, Јануар 2021.
- [6] Синиша Влајић, Projektovanje softvera (Skripta), Београд, Факултет организационих наука, Јануар 2015.
- [7] M. Maatuk, A., & A. Abdelnabi, E. (2021, April). Generating uml use case and activity diagrams using nlp techniques and heuristics rules. In International Conference on Data Science, E-learning and Information Systems 2021 (pp. 271-277). <https://dl.acm.org/doi/abs/10.1145/3460620.3460768>
- [8] Briand, L. C., Labiche, Y., & Leduc, J. (2006). Toward the reverse engineering of UML sequence diagrams for distributed Java software. IEEE Transactions on Software Engineering, 32(9), 642-663.
- [9] Katedra za elektronsko poslovanje Fakulteta organizacionih nauka, Skripta iz predmeta Internet Tehnologije, Beograd 2022.
- [10] <https://www.edureka.co/blog/java-swing/>
- [11] <https://www.javatpoint.com/java-jdbc>