



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У
НОВОМ САДУ




Милица Петровић

Систем за дијагностику менталног стања пацијента

ДИПЛОМСКИ РАД
- Основне академске студије -

Нови Сад, 2023.

	УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА 21000 НОВИ САД, Трг Доситеја Обрадовића 6	Датум:
	ЗАДАТАК ЗА ИЗРАДУ ДИПЛОМСКОГ (BACHELOR) РАДА	Лист:
		3/57

(Податке уноси предметни наставник - ментор)

Врста студија:	<input type="checkbox"/> Основне академске студије
Студијски програм:	Софтверско инжењерство и информационе технологије
Руководилац студијског програма:	ван. проф. др Мирослав Зарић

Студент:	Милица Петровић	Број индекса:	SW46/2019
Област:	Електротехничко и рачунарско инжењерство		
Ментор:	доц. др Сениша Николић		

НА ОСНОВУ ПОДНЕТЕ ПРИЈАВЕ, ПРИЛОЖЕНЕ ДОКУМЕНТАЦИЈЕ И ОДРЕДБИ СТАТУТА ФАКУЛТЕТА ИЗДАЈЕ СЕ ЗАДАТАК ЗА ДИПЛОМСКИ (Bachelor) РАД, СА СЛЕДЕЋИМ ЕЛЕМЕНТИМА:

- проблем – тема рада;
- начин решавања проблема и начин практичне провере резултата рада, ако је таква провера неопходна;
- литература

НАСЛОВ ДИПЛОМСКОГ (BACHELOR) РАДА:

Систем за дијагностику менталног стања пацијента

ТЕКСТ ЗАДАТКА:

Описати систем за дијагностику менталног стања пацијента на основу анализе попуњених упитника, са посебним освртом на складиштење и коришћење доменског знања. Специфицирати и описати архитектуру система. Анализирати правила за анализу одговора и доношење закључка. Документовати решење.

Руководилац студијског програма:	Ментор рада:

Примерак за: ☐ - Студента; ☐ - Ментора

Садржај

1. Увод	1
2. Стање у области.....	3
2.1 BetterHelp.....	3
2.2 Апликација Центар срце	3
2.3 Wysa.....	5
2.4 Научна област	6
2.5 Пресек стања	6
3. Коришћене технике и технологије.....	7
3.1 Angular	7
3.2 REST.....	8
3.3 Spring	8
3.4 Знање	10
3.4.1 Системи базирани на знању.....	10
3.4.2 Експертски системи базирани на правилима	11
3.4.3 Разрешавање конфликта	12
3.4.4 RETE алгоритам.....	12
3.4.5 Уланчавање.....	13
3.5 Drools	13
3.5.1 Drools правила	13
3.5.2 Drools упит.....	14
3.5.3 Rule template.....	14
4. Спецификација	17
4.1 Дијаграм случајева коришћења.....	17
4.2 Дијаграм класа.....	18
4.3 Дијаграм секвенци	20
5. Имплементација система	23
5.1 Клијентска апликација	23
5.2 Серверска апликација	24
5.3 Drools апликација	28
6. Приказ имплементираног система.....	35
7. Закључак	43
Литература.....	45

Биографија	47
------------------	----

1. Увод

Модеран начин живота намеће људима све више обавеза и захтева. Стрес, нервоза, временски рокови и јурњава су присутни код већине данашњих занимања. Такво радно окружење и брз начин живота значајно неповољно утичу на физичко и ментално здравље људи. Као један од начина да се спрече поменути неповољни утицаји било би буђење свести заједнице о важности менталног здравља и популаризације истог. Доступност разних *online* материјала који едукују људе о психологији, међуљудским односима и комуникацији, представљају битан искорак у покушају умањивања узрока неповољних утицаја. Међутим, и поред експанзије садржаја многима је одлазак на психотерапију неопходан, што неретко представља табу тему, те због тога изостаје подршка околине да се људи одреде за једну такву одлуку. Много људи се суочава са анксиозности, социјалном анксиозности и паничним нападима, те би таквим људима итекако била неопходна психотерапеутска помоћ.

У данашње време са развојем нових ИТ технологија сасвим је реално за очекивати да је могуће изградити паметне *online* сервисе који би представљали иницијални корак у дијагностици и лечењу пацијената у домену психотерапије. Примена система базираних на знању у креирању експертског система био би један од поменутих решења. Постојање експертског система који помаже при дијагностици наведених менталних стања би пружио корисницима први сусрет са психотерапијом и притом омогућио да попуњавањем упитника добију резултат о свом менталном стању и на тај начин започну рад на себи.

Тема рада је систем базиран на правилима који омогућава корисницима дијагностику одређених психичких болести и поремећаја, као што су анксиозности, генералног анксиозног поремећаја, паничног напада, паничног поремећаја, социјалне анксиозности и социјалне фобије. Анксиозни поремећаји су група поремећаја чији је централна карактеристика немогућност контролисања страха или бриге [13]. Страх је окарактерисан као реакција на специфичну, видљиву опасност, док је анксиозност окарактерисана као беспредметна стрепња [14]. Анксиозност је стање расположења оријентисано на будућност повезано са припремом за могуће, предстојеће негативне догађаје; а страх је алармни одговор на садашњу или непосредну опасност [15]. Централна карактеристика паничног поремећаја је неочекивани напад панике, односно дискретни период интензивног страха праћен физиолошким симптомима као што су убрзање срца, кратак дах, знојење или вртоглавица. [13]. Социјалну фобију карактерише упорни страх од друштвених ситуација или ситуација које особу излажу потенцијалној критици и пажњи од стране других људи [13].

Циљ рада је изградња апликације чијом употребом би се психологија приближила својим корисницима и омогућио „*рад на себи*“ кроз психолошку подршку. Апликација би омогућила корисницима који немају доменско знање и подршку за бављењем собом да јој на једноставан начин приступе, провере своје ментално стање и повежу се са психологом. Такође, апликација би потенцијално могла да се користи од стране психолога као помоћно средство у раду са њиховим клијентима.

Решење ће бити имплементирано као веб апликација с циљем да апликација буде лако доступна корисницима и да може да се користи на свим уређајима који поседују веб претраживач.

Доменско знање о анализи упитника који попуњава корисник и дијагностиковању менталних стања биће имплементирано дефинисањем правила у *Drools* [7] алату.

У другом поглављу је приказано стање у области где су анализирани тренутно најпопуларније апликације које се баве менталним здрављем. Треће поглавље описује технике и технологије које су коришћене при изради решења. Спецификација решења са одговарајућим дијаграмима приказана је у четвртом поглављу. Пето поглавље описује детаље имплементације решења, а у шестом поглављу је дат приказ имплементираних решења. У последњем поглављу наведене су закључне одреднице и наведени потенцијални правци будућег рада.

2. Стање у области

Дефиниција здравља према Светској здравственој организацији (*World Health Organization - WHO*) гласи *Здравље представља комплексно физичко, ментално и социјално благостање, а не само одсуство болести и онеспособљености*. Наведена дефиниција јасно истиче важност менталног здравља као неизоставни део општег здравља.

На основу системске анализе опште популације у већини европских земаља показано је да ће 27% одрасле популације старости 18-65 година некада током свог живота имати ментални поремећај (то укључује поремећаје у вези са злоупотребом супстанци, психозе, депресије, анксиозност и поремећаје исхране). За жене поменути ризик од оболевања износи око 33,2%, а за мушкарце 21,7%. Код 32% опште популације се очекује да ће имати један ментални поремећај, код 18% се могу очекивати два, а код 14% и три ментална поремећаја током живота [1].

Превенција и ментална хигијена би били неки од начина којим би се могла умањити наведена негативна статистика. Данас постоје апликације које могу пружити подршку људима у поменутом. У наставку су анализиране апликација *Центар срце*, *BetterHelp* и *Wysa*.

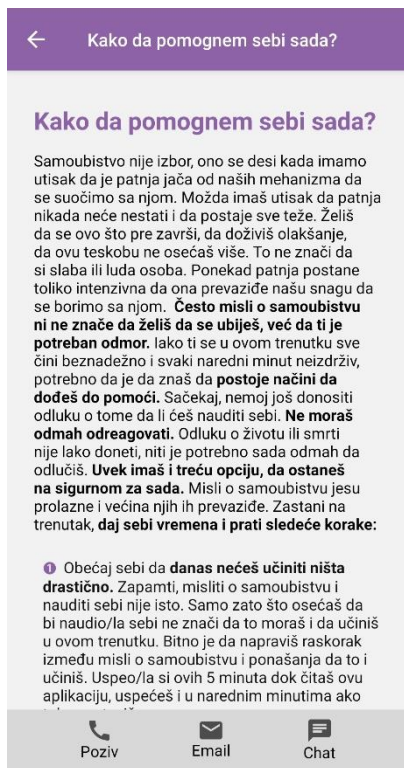
2.1 BetterHelp

BetterHelp је данас најпопуларнија апликација која пружа лако повезивање корисника са психолозима широм света. Приликом инсталације апликације попуњава се упитник који покрива питања о жељеној терапији (индивидуална, партнерска или терапија за децу), личним подацима, љубавном статусу, религији, религијским навикама, духовности, разлозима за терапију, очекивањима од терапеута, претходне терапије, коришћењу лекова, тренутном стању по питању неких болова или тренутној анксиозности, депресији, финансијском статусу, навикама за спавање, преференције везане за терапеута и слично. Свакодневна комуникација са терапеутом се одвија путем аудио, видео и текстуалних порука унутар чета. Уживо сесије у трајању од 30-45 минута могу се одвијати путем телефонског и видео позива или чета. Коришћење апликације се плаћа 65 долара недељно односно 260 долара месечно. У ову цену су укључене 4 сесије и подршка унутар чета. Ово је значајно повољније за америчке стандарде где је цена једне сесије 150-180 долара. У случају да корисник није у могућности да плати сесију има могућност да се обрати за финансијску помоћ.

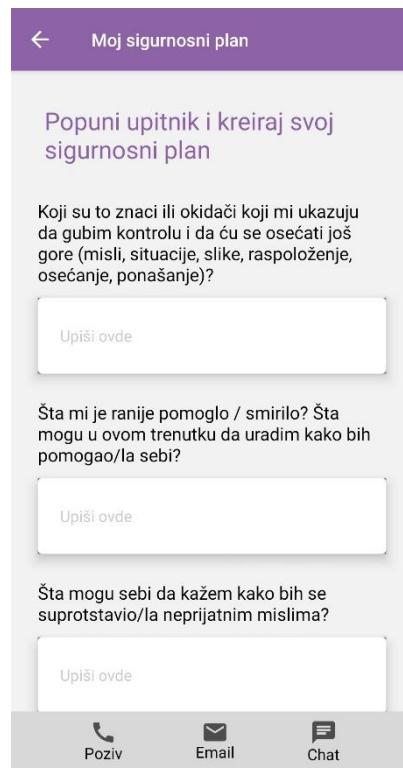
2.2 Апликација Центар срце

Апликација *Центар срце* је мобилна апликација за подршку људима у тренуцима када им је јако тешко и када се чини да нема излаза. Подршка је упућена свима који имају суицидне мисли и намере.

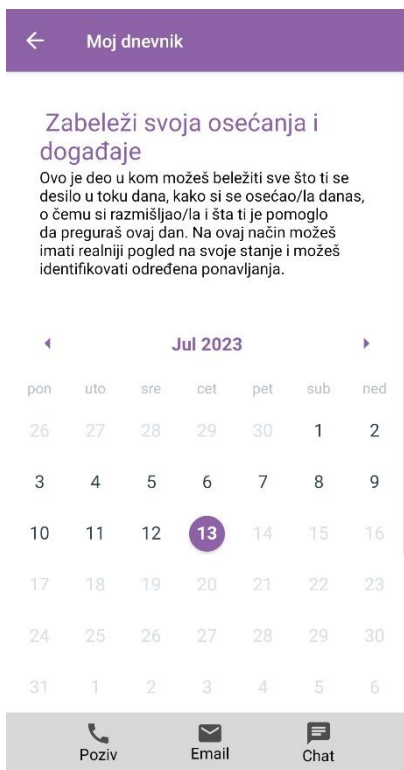
Након инсталације, апликација се може одмах користити без остављања података. Регистрација није потребна тако да је анонимност загарантована. На почетној страници се налази мени којим је омогућена навигација кроз апликацију. У апликацији се налазе текстови и опште информације о суициду као и текстуални водич под називом *Како да помогнем себи сада?* (Слика 1) који кроз текстуални опис корака пружа прву помоћ у безизлазним тренуцима. Одељак *Мој сигурносни план* је упитник који наводи корисника да одговорима на питања направи свој план са акцијама које може предузети (Слика 2). Доступан је и дневник за евидентирање осећања и стања (Слика 3). Корисник има могућност да ступи у контакт са волонтерима преко позива, мејла или чета (Слика 4).



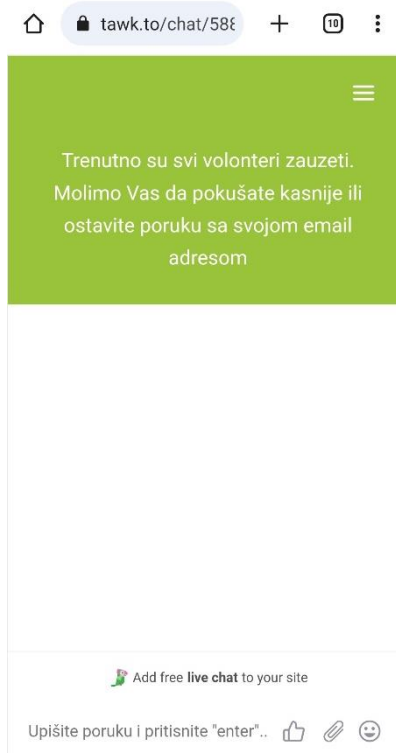
Слика 1 Водич за помоћ



Слика 2 Сигурносни план



Слика 3 Дневник



Слика 4 Контакт са волонтерима

2.3 Wusa

Wusa апликација је заснована на вештачкој интелигенцији која пружа подршку и смернице корисницима. Корисник се дописује са вештачком интелигенцијом која користи когнитивне бихејвиоралне технике засноване на доказима, технике дијалектичког понашања, медитацију, технике дисања, јогу [2]. Клинички је доказано да Wusa вештачка интелигенција успоставља терапијски однос еквивалентан људском терапеуту у току прве недеље [2]. Апликација пружа низ алата и функција, укључујући праћење расположења, технике медитације и опуштања и постављање циљева. За свакодневни стрес, Wusa нуди специфичне програме самопомоћи кроз више од 150 вежби заснованих на доказима, укључујући материјале за анксиозност, стрес, депресију, проблеме са спавањем (Слика 5, Слика 6).

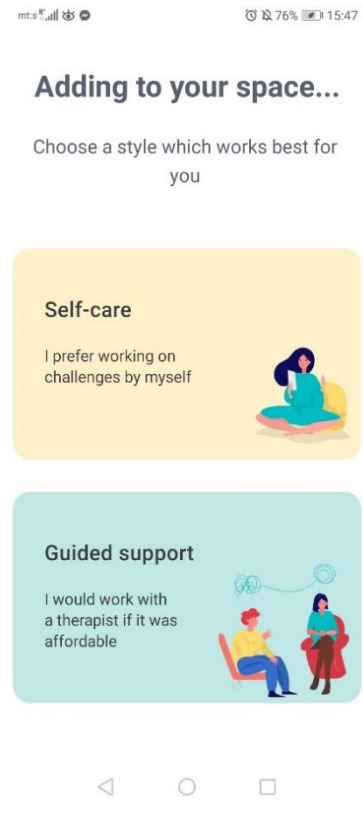
Након инсталације апликације није потребна регистрација нити остављање личних података, довољно је само унети надимак. Након уноса неког имена корисник бира области на којима жели да ради (Слика 5, Слика 6), а затим бира и да ли жели да у његов рад буде укључен терапеут или ће самостално радити на одабраним областима (Слика 7).



Слика 6 Области за рад



Слика 5 Области за рад



Слика 7 Одабир начина рада на себи

2.4 Научна област

Системи базирани на знању се ослањају на знање које је формално исказано и складиштено у систему тако да систем може да доноси одлуке на основу тог знања. Експертски системи представљају специјализоване системе базирани на знању јер интегришу доменско знање експерта у базу знања која се користи за закључивање.

У раду [16] представљен је систем који је намењен предвиђању менталног стања појединца. Један модул система је имплементиран као систем базиран на правилима и представља упитник од 10 питања на која одговара корисник и чије се вредности одговора сабирају и скалирају и утичу на закључак о коначном стању појединца поред осталих метода као што су детекција емоција на основу израза лица.

У раду [17] представљен је систем базиран на правилима који омогућава детекцију менталних поремећаја унутар ког је интегрисано знање експерта. У систем је интегрисано знање *MINI ICD-10* упитника који се састоји од питања која детектују главне и споредне симптоме менталног поремећаја. Дати систем је тестиран у Индонезији и показао је тачност од 96%.

Рад [18] даје концептуални модел експертског система базираног на знању који идентификује анксиозност и депресију код жена током трудноће. Систем је намењен да помогне здравственим радницима у пружању менталне подршке трудницама током и након пандемије *Covid-19* вируса. Систем се базира на *EPDS* упитнику.

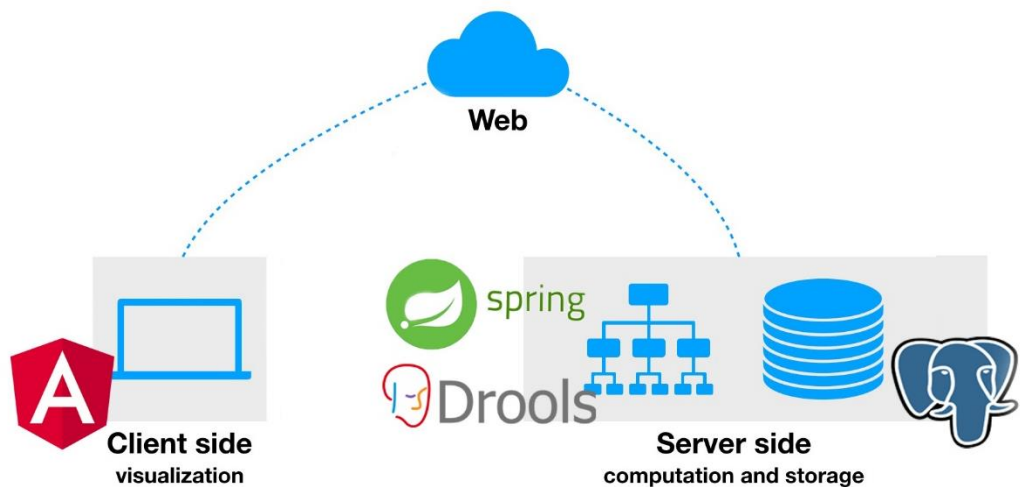
2.5 Пресек стања

Добре стране описаних апликација су што повезују кориснике са психологом уз загарантовану анонимност или омогућавају коришћење апликације без остављања приватних података. Такође, апликације обезбеђују доста материјала у текстуалној форми и видео снимака које могу бити први корак у раду на себи и/или самопомоћи. Корисник може да оствари интерактивност преко чета или разговора, добија обавештења и упутства о темама од интереса.

Кориснику је дата могућност да самостално бира области којима жели да се бави попут анксиозности, мотивације, стреса, депресије, траума, веза што доводи корисника у ситуацију да сам себи одреди поље рада. Како корисник није стручан, може доћи у ситуацију да избегава област која би управо била његово поље рада. Апликација која је описана у раду се бави дијагностиковањем менталних стања тако да апликација упућује корисника на области којима би могао да се бави и да притом у свој рад укључи психолога. Научни радови који се баве анализом упитника у циљу дијагностиковања неког менталног поремећаја или стања представљају мотивацију за овај рад. Апликација пружа кориснику брзу дијагностику анксиозности, генералног анксиозног поремећаја, паничног напада, паничног поремећаја, социјалне анксиозности и социјалне фобије и повезивање са психологом.

3. Коришћене технике и технологије

Систем чине клијентска апликација, серверска апликација, систем базиран на правилима и база података. Клијентска апликација је развијена у *Angular*-у [3], а серверски део је развијен у *Spring*-у [4] уз употребу *REST* [5] сервиса. Подаци се складиште у *PostgreSQL* бази података. Систем базиран на правилима је имплементиран у *Drools*-у [7]. На Слика 8 је приказана архитектура система.



Слика 8 Архитектура апликације

3.1 Angular

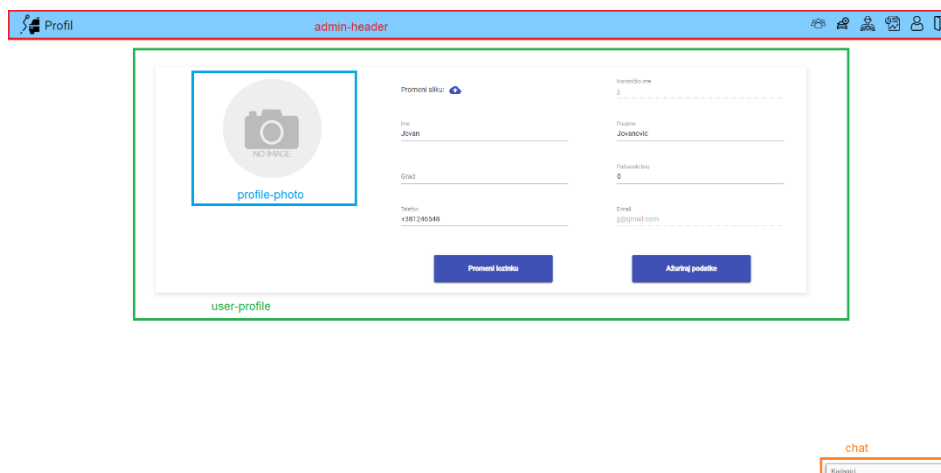
Angular је радни оквир за дизајн апликација и развојна платформа за креирање апликација на једној страни (*Single-Page*) заснован на *HTML*-у и *TypeScript*-у [3].

Основни градивни елементи апликације су компоненте. Компонента обједињује приказ у *HTML*-у и апликативну логику у *TypeScript*-у. Компоненте су повезане тако да чине стабло. Угњежавањем компоненти постиже се читљивост кода и избегавање понављања истог (Слика 9).

Скуп компоненти који гради логичку целину представља модул. Свака *Angular* апликација мора се састојати од барем једног модула који мора садржати коренску компоненту док се остале компоненте угњежавају унутар коренске. У *TypeScript* класи која представља модул наводе се све компоненте и сервиси који припадају том модулу, компоненте које се извозе у друге модуле и модули који се увозе у дати модул.

За комуникацију са серверском апликацијом користе се сервиси, односно *TypeScript* класе које су анотиране са *@Injectable* анотацијом. Инјектор је задужен за инстанцирање сервиса, управљање његовим животним циклусом и прибављањем по потреби. У сервисима се налази

логика коју користи више компоненти. Сервиси се могу користити и за комуникацију између две удаљене компоненте.



Слика 9 Приказ уоквирених угњеждених компоненти

3.2 REST

REST (Representational state transfer) је архитектонски стил који се користи на веб сервисима. *Rest API*, интерфејс за програмирање апликације, је скуп правила која дефинишу како апликације или уређаји могу да се повежу, међусобно комуницирају и размењују податке. Др Роја Филдинг је први пут дефинисао *REST API* у својој докторској дисертацији 2000. године [6].

Флексибилност је основна предност *REST* архитектуре. *REST API* најчешће комуницира путем *HTTP* захтева и податке може размењивати у различитим форматима од којих је *JSON* најпопуларнији. *REST* дефинише шест ограничења:

- Клијент-сервер архитектура
- Непостојање стања
- Кеш меморија
- Униформни интерфејс
- Слојевити систем
- Код на захтев (опционо)

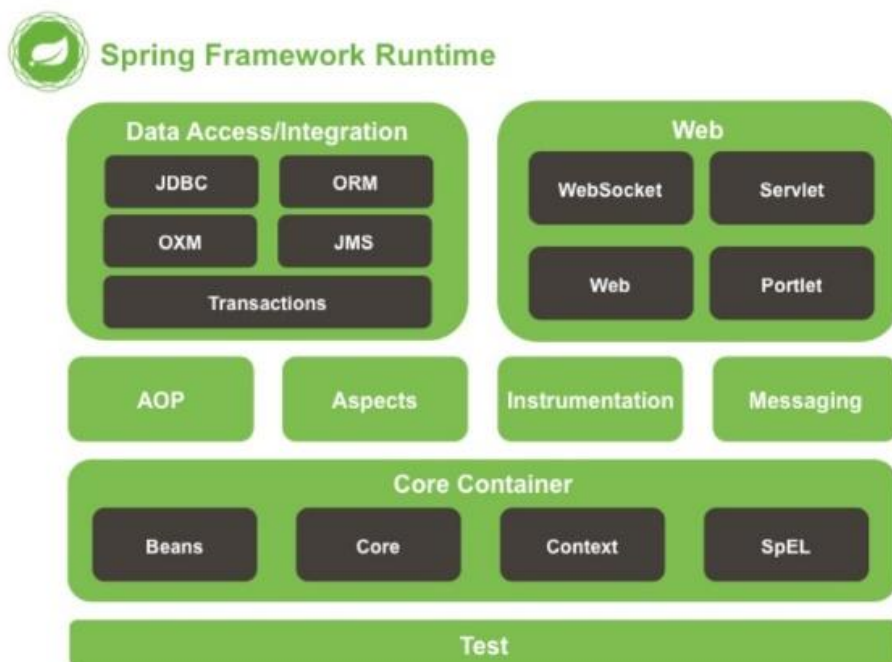
Применом наведених ограничења постиже се скалабилност, могућност измене, једноставност, преносивост.

3.3 Spring

Spring је радни оквир за развој апликација у *Java* програмском језику. *Spring* омогућава креирање *Java EE* апликација на једноставнији начин јер обезбеђује инфраструктуру тако да програмер може да се фокусира на доменске проблеме. Прва верзија *Spring* радног оквира публикована је 2003. године.

Предност *Spring*-а је што користи обичне класе - *Plain old Java objects(POJO)* које су анотиране анотацијама којима се дефинише како *Spring* управља објектима тих класа. *Spring* обезбеђује повезивање објеката, брине о животном циклусу објеката и добавља тражене објекте на захтев пратећи *Dependency Injection (DI)* механизам. Такође, у *Spring*-у је подржано декларативно и аспектно-оријентисано програмирање.

Spring се састоји од функција организованих у 20 модула. Модули су груписани у *Core Container*, *Web*, *Data Access/Integration*, *Aspect Oriented Programming (AOP)*, *Aspects*, *Instrumentation* и *Test* (Слика 10).



Слика 10 *Spring* модули [4]

Core Container је основни модул који обезбеђује фундаменталне делове оквира попут *Inversion of Control* и *Dependency Injection* механизма. *Beans* модул садржи фабрику објеката која омогућава одвајање конфигурације и спецификације зависности од стварне програмске логики уместо употребе *Singleton* шаблона. Модул *Context* се ослања на *Core* и *Beans* модуле. *Context* наслеђује функције *Beans* модула и додаје подршку за пропагацију догађаја, учитавање конфигурације апликације и свих објеката. Централна тачка *Context* модула је *ApplicationContext* интерфејс. *Expression language* модул пружа језик за писање упита и манипулацију графом објеката током времена извршавања (*run-time*).

Data Access/Integration модул се састоји из подмодула *JDBC*, *ORM*, *OXM*, *JMS* и *Transactions*. *JDBC* модул апстрахује везу са базом података и уклања потребу за заморним *JDBC* програмирањем и парсирањем грешака специфичних за добављање података из базе података. *ORM* модул обезбеђује објектно-релационо мапирање чија конкретна имплементација може бити *OpenJPA*, *Hibernate* [8]. Сви оквири за објектно-релационо-мапирање могу се користити у комбинацији са функцијама које нуди *Spring* попут декларативног управљања трансакцијама.

Web модул пружа подршку за различите аспекте развоја веб апликација попут инјекције објеката у сервлете, постављање фајлова на сервер. Модул *Web-Servlet* садржи *Spring*-ову имплементацију *MVC* обрасца за веб апликације. Коришћена је подршка за *REST* јер је клијентски део имплементиран помоћу *Angular*-а.

Test модул омогућује тестирање засновано на *JUnit* и *TestNG* библиотекама. Омогућује спровођење јединичног и интеграционог тестирања. За интеграционо тестирање омогућује учитавање објеката у *Spring application context*. Такође, садржи и имплементацију *mock* објеката који се користе за јединичне тестове и изолацију дела апликације који се тестира.

3.4 Знање

Знање је резултат анализе информација које се доводе у контекст. Знање представља информацију коју људи користе за решавање проблема. По Трипартитној дефиницији знање је образложено истинито уверење ако је исказ тачан, истинит и оправдан. Компоненте знања су:

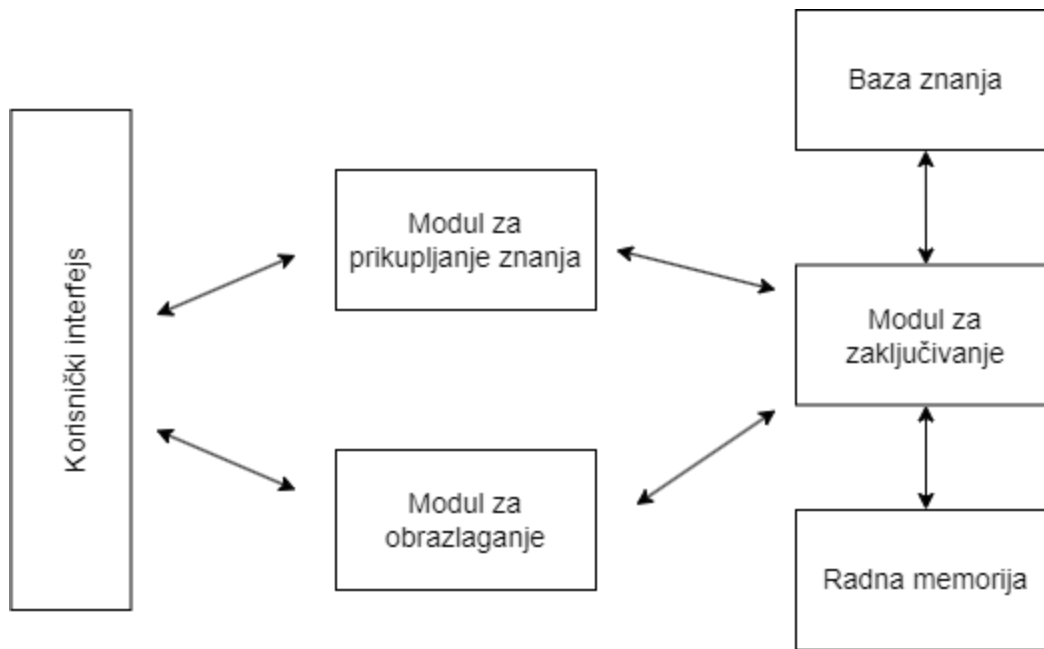
- Чињеница – ствар која се зна или је доказано да је она истинита
- Концепт – апстракција или уопштење искуства
- Процедура – низ акција које се спроводе по одређеном редоследу и начину
- Модел – репрезентација система или сложеног ентитета
- Хеуристика – искуствено базирана техника за решавање проблема
- Пример – део нечега што се узима да се покаже карактер целине

3.4.1 Системи базирани на знању

Системи базирани на знању се ослањају на знање које је формално исказано и складиштено у систему тако да систем може да доноси одлуке на основу тог знања. Експертски системи представљају специјализоване системе базиране на знању јер интегришу доменско знање експерта у базу знања која се користи за закључивање.

Експертски системи су применљиви на уске области знања односно домене знања. Разлози за прављење експертских система јесу чување знања експерта, основа за обуку почетницима, обезбеђивање уједначености одлука и омогућавање доступности знања експерта.

Системи базирани на знању се састоје из базе знања, модула за закључивање, радне меморије, модула за резонување, модула за прикупљање знања и корисничког интерфејса (Слика 11).



Слика 11 Структура система базираног на знању

3.4.2 Експертски системи базирани на правилима

Експертски системи базирани на правилима су системи чија база знања садржи доменско знање стручњака представљено у форми *if-then* правила. Тврдње у *if-then* форми називају се продукциона правила. У експертским системима базираним на правилима знање стручњака је формулисано продукционим правилима која се примењују на чињенице из радне меморије у циљу доношења закључака.

Претпоставке које је потребно да буду задовољене се пишу у *if* делу правила, односно у левој страни правила. Потребно је да су задовољене све претпоставке како би се извршила акција или донео закључак. У случају да само једна претпоставка није задовољена, акција се неће извршити.

Над претпоставкама се могу писати логички оператори попут: *and*, *or*, *not*. Оператор *not* се тумачи као да не постоји чињеница у меморији која задовољава услов. Заграде се користите за груписање. Претпоставка се састоје из објекта, вредности и оператора. У примеру *The temperature is 36 degrees - > the temperature* је објекат, *36 degrees* је вредност и *is* је оператор.

Акције/закључци се пишу у десној страни правила, након *then* кључне речи. Десна страна правила представља закључак до ког је дошао експерт сагледавањем свих чињеница. Акција/закључак представља предвиђени корак за решавање проблема.

Пре почетка резонувања продукциона правила попуњавају базу знања, а чињенице се иницијализују у радној меморији. Након покретања модула за закључивање упарују се чињенице из радне меморије са патернима који су дефинисани у претпоставкама у циљу проналаaska правила која су задовољена.

Правило је задовољено ако се лева страна правила слаже са чињеницама из радне меморије. Модул за закључивање одређује правила чији су предуслови задовољени. Задовољена правила се смештају у агенду која се налази унутар модула за закључивање. У једном тренутку само једно правило може да се активира. У случају да се у агенди нађе више правила примењује се неки од поступака за разрешавање конфликта.

Након извршавања десне стране правила, правило се уклања из агенде. Извршавање правила може довести до промена у радној меморији додавањем нових чињеница или изменом и уклањањем постојећих тако да се врши поново упаривање и додавање нових задовољених правила у агенду и избацивање оних која више не важе. Модул за закључивање завршава са радом у тренутку када у агенди више нема задовољених правила или експлицитном *stop* командом.

3.4.3 Разрешавање конфликта

Сет правила која су задовољена се назива конфликт сет (*conflict-set*). Разрешавање конфликта (*conflict resolution*) представља метод одабира правила које ће се извршити из конфликт сета.

Стратегије за одабир правила:

1. Одабир правила на основу приоритета – најчешће редослед правила у бази знања одређује и приоритет правила. Приоритет се може и експлицитно дефинисати.
2. Одабир правила на основу позиције правила – бира прво правило које дефинисано у тексту
3. Одабир најспецифичнијег правила – правило је специфичније ако његово задовољење захтева испуњење више предуслова
4. Одабир на основу мета-правила – односно на основу правила о правилима. Већи приоритет има правило који је писао експерт него правило које је писао почетник
5. Одабир на основу најсвежијих података – одабрати правило на основу чињеница које су најскорије додате

Најчешће се користе стратегије 1 и 2 у разрешавању конфликта.

3.4.4 RETE алгоритам

RETE алгоритам је развијен крајем седамдесетих година двадесетог века захваљујући др *Charles L. Forgy*. *RETE* алгоритам је алгоритам који проверава да ли чињенице задовољавају правила у систему правила (*rule-engine*). Правила су скуп претпоставки и акција. Чињеница је вектор својстава.

RETE на латинском значи мрежа. *RETE* мрежу дефинишу усмерени ациклични графови који представљају сетове правила. Сваки чвор у мрежи одговара услову или делу услова из неке претпоставке у левој страни правила. Путања од коренског чвора до крајњег чвора представља леву стару правила. Сваки чвор има своју меморију у којој се налазе све чињенице које задовољавају дати услов. За сваку чињеницу креира се елемент радне меморије (*Working Element Memory - WME*) који представља ту чињеницу.

Свака чињеница улази у *RETE* мрежу кроз коренски чвор мреже. Коренски чвор прослеђује чињеницу чворовима наследницима. На тај начин чињеница пролази кроз неку од грана стабла док не стигне до неког од крајњих чворова.

3.4.5 Уланчавање

Два основна механизма за закључивање су уланчавање унапред и уланчавање уназад.

Уланчавање унапред (*forward chaining*) је резонување које је вођено подацима (*data-driven*) од чињеница ка закључку. На основу чињеница које задовољавају претпоставке се активирају правила чијим извршењем се потенцијално мења радна меморија или доноси закључак. Промена радне меморије доводи до поновне евалуације правила. Стање радне меморије је познато, али коначни закључак, ако се до њега дође, није познат.

Уланчавање уназад (*backward chaining*) је резонување вођено упитом (*query-driven*) од закључка ка чињеницама. Знамо шта желимо да докажемо, али не знамо да ли ћемо то доказати са чињеницама које имамо у радној меморији. Полази се од хипотезе односно упита и прво се траже правила чија је десна страна хипотеза. У случају да таква правила не постоје, хипотеза је оборена. У случају да таква правила постоје, проверава се да ли чињенице задовољавају леве стране. Ако чињенице задовољавају леве стране, хипотеза је доказана, а ако не задовољавају онда лева страна правила постаје хипотеза и циклус се понавља рекурзивно.

3.5 Drools

Drools је део *KIE (Knowledge Is Everything)* пројекта који нуди алате за решења за аутоматизацију пословања и управљање. *Drools* представља систем за управљање пословним правилима користећи механизам за закључивање заснованим на уланчавању унапред и уназад. Поддржава и комплексно процесирање догађаја (*Complex Event Processing - CEP*). *Drools* представља основни градивни елемент за развој експертских система. Имплементира проширен *RETE* алгоритам.

3.5.1 Drools правила

Правила се пишу у *.drl* фајловима. Структура фајла обухвата назив пакета, импорт секцију, декларацију глобалних варијабли, типова и догађаја, правила и упите. У Листинг 1 је приказана структура фајла и пример правила.

Називи правила морају бити јединствени. У *when* секцији се пишу услови помоћу *DRL* језика. *When* секција представља леву страну правила (*Left Hand Side - LHS*). Потребно је да чињенице испуњавају услове из *when* дела како би се извршило правило. У *then* секцији, односно у десној страни правила (*Right Hand Side - RHS*) се пишу последице активирања правила.

За свако правило се могу дефинисати и атрибути који омогућавају филтрирање правила. Евалуација атрибута се ради након што је задовољен *when* део правила. *Salience* атрибут дефинише приоритет правила, *no-loop* атрибутом се спречава бесконачна петља где правило само себе активира док *look-on-active* спречава да се правило поново изврши за исту чињеницу. Правила се могу груписати помоћу *agenda-group* атрибута.

У *LHS* делу правила могу се користити оператори, *Java* изрази попут *and*, *or*, *not*, *collect*, *contains*, *memberOf*, *from*, *accumulate*, *exists*, *forall*. У *RHS* делу правила се пишу акције које могу оперисати над радном меморијом попут *insert*, *modify*, *update* и *delete*. Помоћу наведених кључних речи могуће је додавати нове чињенице у радну меморију, мењати и брисати постојеће.

```

// naziv paketa
// importi
// globalne promenljive

rule "naziv pravila"
// atributi: sailence, no-loop, agenda-group
when
    // Left Hand Side
then
    // Right Hand Side
end

```

Листинг 1 Структура .drl фајла и пример правила

3.5.2 Drools упит

Упит (*query*) представља једноставан начин за претрагу радне меморије. Упити се састоје само из *LHS* дела и могу да приме параметре. Погодни су за писање рекурзивних упита односно за уланчавање уназад. Упит се састоји из кључне речи *query* након које се пише назив упита и опциони параметри које прима сам упит. Након тога пише се *LHS* што је приказано на Листинг 2.

```

query "naziv upita" (Tip parametar)
    Left Hand Side
end

```

Листинг 2 Пример упита

3.5.3 Rule template

Rule template представљају начин за генерисање *DRL* правила у времену извршавања користећи *template* фајлове и податке који се могу представити табеларно.

Основна предност *rule template*-а је да се може користити један *rule template* фајл за различите сетове података и обрнуто. Текстуални фајлови у којима се пишу темплејти имају *.drt* (*Drools rule template*) екстензију и кључне речи којима се означавају различите секције темплејта и променљиве.

На Листинг 3 је приказана структура темплејта. Након заглавља темплејта наводе имена променљивих која се налазе и у подацима која ће се интегрисати у генерисана правила. Празан ред означава крај заглавља. Након импорт секције и декларација наводи се кључна реч *template* и јединствено име темплејта. Унутар темплејта дефинише се правило које ће бити генерисано са различитим подацима. Јединственост назива правила постиже се бројачем који је додат на крају назива правила.

```
template header
// nazivi promenljivih

package diagnostics; // naziv paketa

// import sekcija, definisanje tipova i funkcija

template "templ-1" // početak templejta

rule "nazi_pravila_{row.rowNumber}"

when
    // LHS
then
    // RHS
end

end template // kraj templejta
```

Листинг 3 Структура темплејта

4. Спецификација

Систем за дијагностику менталног здравља пацијената је веб апликација која је замишљена као помоћно средство у раду клијената и психолога, али и веб сајт који би био лако доступан корисницима који желе да се баве менталним здрављем. У овом поглављу систем је представљен различитим *UML(Unified Modeling Language)* [9] дијаграмима.

4.1 Дијаграм случајева коришћења

У комуникацији наручиоца софтвера и тима који развија софтвер важну улогу има презентација захтева. Технику за прикупљање, анализу и бележење захтева базирану на случајевима коришћења (*use case*) осмислио је и Иван Јакобсон 1994. године [10]. Дијаграми случајева коришћења се користе за описивање скупа радњи (случајева коришћења) које софтвер треба да изврши у сарадњи са једним или више учесника.

Случај коришћења представља функционалност или сервис која се пружа учеснику система. Сваки случај коришћења треба да пружи видљив и вредан резултат учесницима [10].

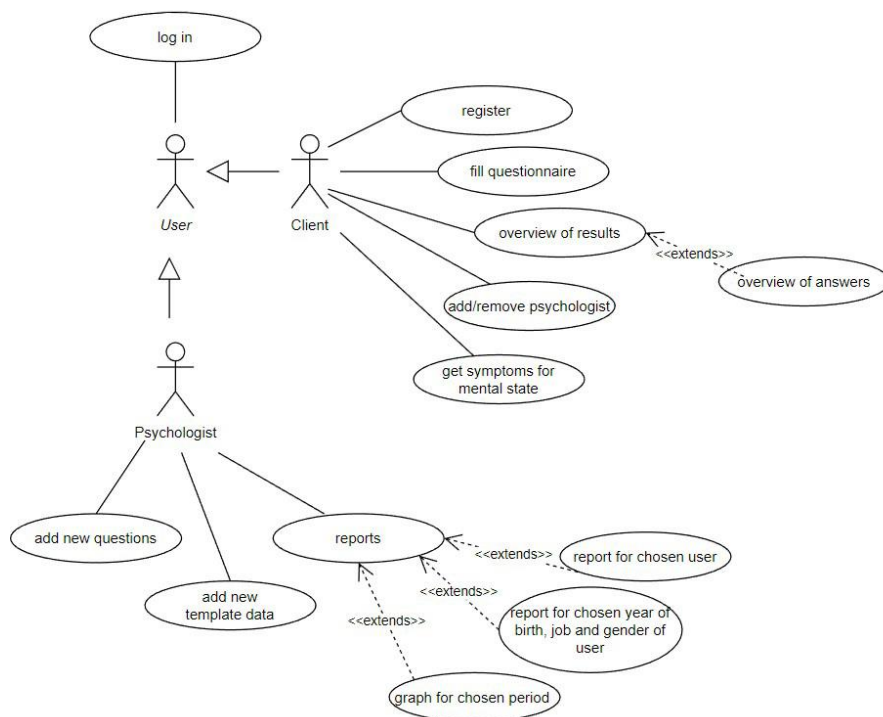
Учесник (*actor*) представља улогу у систему. Улоге се додељују корисницима система и једном кориснику се може доделити више улога. На Слика 12 приказан је дијаграм случајева коришћења за систем описан у раду .

Апликација подржава два типа корисника клијента и психолога.

Клијент је корисник који се може регистровати на систем и након регистрације може да се пријави. Клијент има могућност да попуни упитник за дијагностиковање, додаје и уклања психологе којима дозвољава приступ својим одговорима. Такође, има преглед својих дијагностикованих стања и одговора на питања. Клијент може да за одабрано ментално стање добије листу свих стања која су симптоми одабраном стању.

Психолог има могућност да се пријави и додаје нове упитнике и правила за обраду тих упитника као и увид у извештаје. Психолог може да прегледа резултате за кориснике који су му претходно одобрили приступ, добије извештај на основу године рођење, професије и пола клијената као и графички приказ дијагностикованих стања у временском опсегу.

Обрада одговора на упитник који попуњава корисник и упит за добијање симптома за одабрано стање је реализована коришћењем система базираних на правилима која се ослањају на доменском знању експерта.



Слика 12 Дијаграм случајева коришћења

4.2 Дијаграм класа

Дијаграм класа се користи за скицирање структуре пословног домена за који се имплементира софтвер, комуникацију идеја унутар тима, спецификацију имплементације, генерисање кода и документовање решења зависно у којој се фази развоја налази софтвер [10].

Описани систем се састоји из осам класа и четири енумерације. Набројане су класе:

- User
- Role
- Privileges
- Question
- DetectionType
- Answer
- Result
- DiagnosticTemplate

Класе *User*, *Role* и *Privileges* служе за пријаву и ауторизацију корисника као и бележење основних података о кориснику. *Privileges* класа садржи атрибут *name* који описује конкретну адресу на серверу којој се може приступити. *Role* класа описује улоге у систему и садржи атрибут *name*.

Одговарајући парови инстанци класа *Privileges* и *Role* представљају дозволе које одређују која улога има могућност да приступи одређеној адреси на серверу.

Класа *User* има атрибуте *name*, *surname*, *email*, *password enabled*, *role*, *yearOfBirth*, *gender* и *job* што су основни подаци корисника. Класа *User* има листу психолога који су додати, листу одговора које је дао корисник и лису резултата корисника.

DetectionType класа представља ментално стање које систем може дијагностиковати и садржи атрибуте *name* што представља назив менталног стања и *detectionTypeGroup* односно назив групе којој припада стање.

Класама *Question*, *Answer* и *Result* описана су питања и одговори које даје корисник и резултат дијагностиковања на основу одговора. *Question* класа садржи атрибуте *text*, *questionLayer*, *detectionType* и *positive* који описују текст питања, ментално стање на које се питање односи и слој ком питање припада.

Answer класа представља одговор корисника и садржи референцу на питање, одговор у виду бројчане вредности и датум и време када је одговор дат.

Result класа моделује резултат дијагностиковања и садржи дијагностиковано стање и време када је дијагноза утврђена.

DiagnosticTemplate класом описан је темплејт за дијагностику који садржи назив стања које се детектује, слој у ком се детектује дато стање, интензитет стања, минималан и максималан број бодова који одговарају описаном стању, текст који описује стање и даје информацију кориснику и информацију да ли је описано стање крајње.

Поред класа, систем чине и четири енумерације:

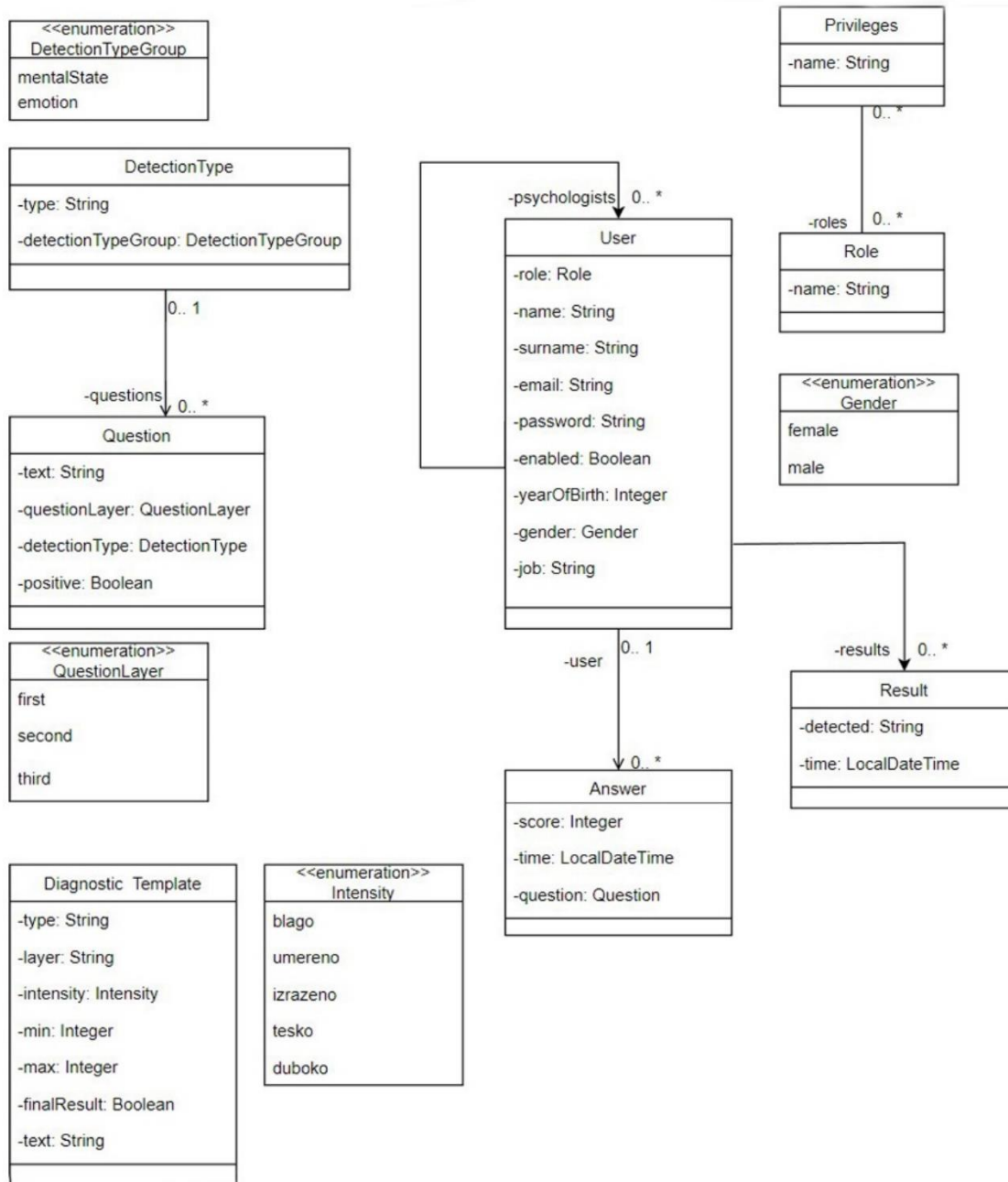
- *DetectionTypeGroup*
- *QuestionLayer*
- *Intensity*
- *Gender*

DetectionTypeGroup енумерација има вредности *mentalState* и *emotion* и тако одваја стања од емоција с обзиром да дијагностиковање менталних стања представља подсистем већег система. Користи се као атрибут класе *DetectionType*.

QuestionLayer енумерација има вредности *first*, *second* и *third* којим се описује редослед питања у упитнику, то јест подела питања по нивоима, круговима. У класи *Question* се користи за опис питања.

Intensity означава ниво озбиљности дијагностикованог стања и има вредности *blago*, *umereno*, *izrazeno*, *tesko* и *duboko*. На основу нивоа озбиљности дијагностикованог стања се доносе одлуке о наредним дијагнозама.

Gender представља пол и има вредности *male* и *female*. Користи се у класи *User* и представља основни податак о кориснику. На Слика 13 приказан је дијаграм класа.



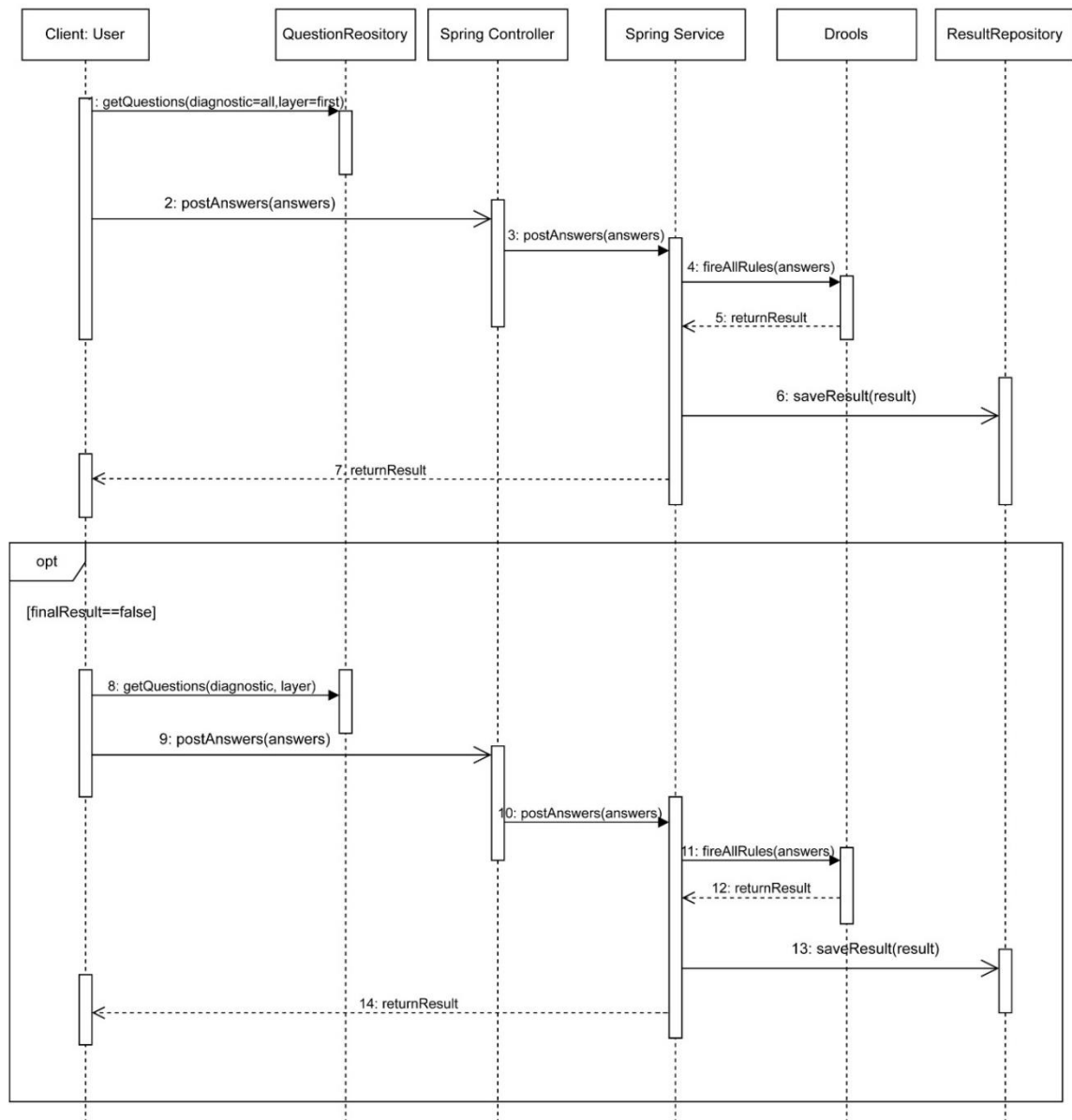
Слика 13 Дијаграм класа

4.3 Дијаграм секвенци

Дијаграм секвенци приказује комуникацију између учесника у систему и елемената система у времену [10]. Дијаграм секвенци описује интеракцију фокусирајући се на редослед порука које се размењују.

На Слика 14 је приказан дијаграм секвенци за клијента приликом попуњавања упитника. Прва акција је добијање првог сета питања. Клијент добије први слој питања позивањем методе

getQuestions *QuestionController*-a. Затим клијент шаље одговоре на питања контролеру. Контролер позива *postAnswers* методу сервиса. Сервис покреће правила и добија резултат који се затим чува у бази података позивом *saveResult* методе репозиторијума. Добијени резултат се враћа клијенту. У случају да клијент има услове за даља испитивања, односно *finalResult==false*, клијент тражи нови сет питања и наставља се дијагностика за ментална стања за која су утврђене предиспозиције у претходном сету питања. Претходно описани поступак се понавља до краја упитника кроз сва три слоја питања.



Слика 14 Дијаграм секвенци

5. Имплементација система

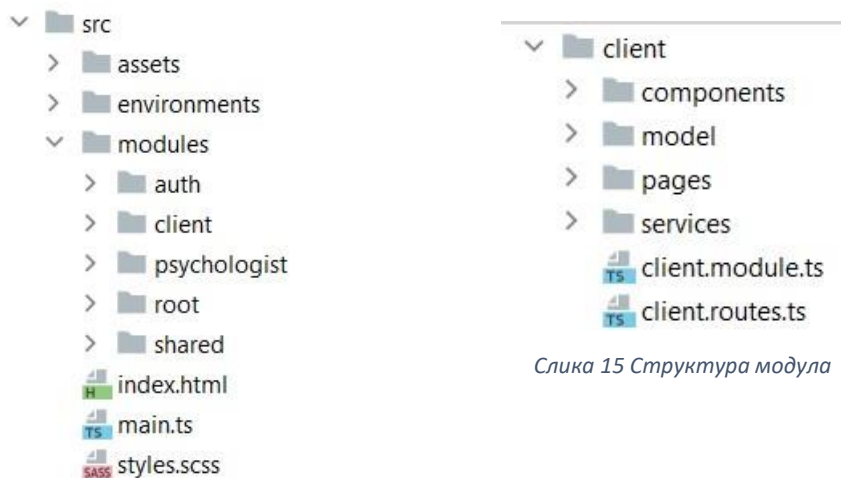
У овом поглављу биће описано како је имплементиран систем то јест детаљнији приказ сваког дела апликације.

Апликација за дијагностиковање менталних стања се састоји из четири дела: клијентски део писан у *Angular*-у, серверски део, *PostgreSQL* база података и систем базиран на правилима имплементиран помоћу *Drools* алата.

Изворни код апликације се налази на GitHub репозиторијуму <https://github.com/MilicaPe/diplomski.git>.

5.1 Клијентска апликација

Клијентска апликација је имплементирана коришћењем *Angular* радног оквира. Изворни код апликације има структуру као на Слика 16. У *assets* фолдеру се налази статички садржај односно све слике и иконице које се користе за дизајн. У *Environments* фолдеру је дефинисана константа која представља адресу на којој се налази серверска апликација. Имплементирана је лења евалуација модула. Апликација се састоји из пет модула.



Слика 16 Структура клијентске апликације

Слика 15 Структура модула

Унутар сваког модула се налазе фолдери (Слика 15):

- *components* – унутар *components* фолдера налазе се компоненте које су део неке од страница
- *pages* – унутар *pages* фолдера налазе се компоненте на које усмерава рутер
- *models* – у *models* фолдеру су дефиниције свих интерфејса који моделују податке које се размењују између клијентског и серверског дела. На основу интерфејса се аналогно формирају *DTO(Data Transfer Object)* класе на серверском делу
- *services* – унутар *services* фолдера се налазе све *TypeScript* класе које комуницирају са серверским делом апликације
- *naziv.module.ts* - *TypeScript* класа унутар које су декларисане све компоненте модула и сви модули које се увозе (импортују) у дати модул

- *naziv.routes.ts* - *TypeScript* класа унутар које су дефинисане руте до компоненти из *pages*

Root модул је коренски модул апликације у ком се налази коренска компонента (*RootComponent*). У *root* модулу су смештена заглавља за клијента и психолога као и страница на коју се усмерава корисник уколико покуша да приступи путањи која не постоји. *TitleService* класа обезбеђује одговарајуће наслове зависно од путање на коју се усмерава корисник. У класи *AppModules* су дефинисане све компоненте *root* модула и увезени су сви модули које овај модул користи. У *AppRoutingModule* класи је дефинисана лења евалуација модула. Дефинисано је који модул је потребно учитати на основу путање.

Shared модул садржи компоненте које су заједничке за више модула, а то су компоненте за преглед већ попуњеног упитника. Овде су дефинисани и *pipes* који омогућавају униформан приказ за датуме и енумерације. У *interceptors* фолдеру је дефинисан *interceptor* који обрађује сваки захтев који се упућује ка серверској апликацији тако што уграђује токен за ауторизацију у сваки захтев.

У *auth* модулу се налазе компоненте за аутентификацију корисника. То су компонента за пријаву корисника коју може користити било који корисник, компонента за регистрацију новог корисника коју користе клијенти.

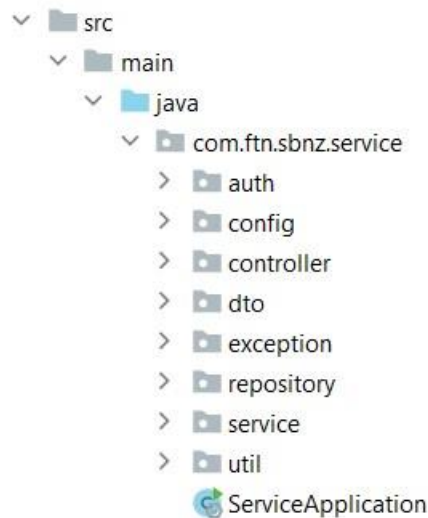
У *client* модулу се налазе све компоненте које су доступне клијенту. То су компоненте за попуњавање упитника, преглед добијене дијагнозе и преглед одговора на питања, преглед симптома за одабрано ментално стање. Такође, ту су и компоненте за приказ психолога који имају приступ подацима клијента и компоненте за додавање у уклањање психолога.

У *psychologist* модулу налазе се компоненте којима може да приступа психолог. Психолог може да додаје нова питања за упитник и дефинише правила за обраду тих питања. Психологу су доступна три типа извештаја. Први тип извештаја се односи на преглед дијагностикованог стања и одговора на питања за одабраног корисника за кога психолог има приступ. Други извештај приказује резултате упита на основу године рођења, професије и пола клијента. Трећи извештај представља график на ком се може пратити одабрано ментално стање у одабраном временском опсегу.

5.2 Серверска апликација

Серверска апликација је имплементирана коришћењем *Spring* радног оквира. Клијентска и серверска апликација комуницирају по принципу захтева и одговора. Сервер комуницира и са апликацијом која представља модел, системом базираним на знању и базом података.

Изворни код апликације је структуриран у фолдере: *auth*, *config*, *controller*, *dto*, *exception*, *repository*, *service* и *util* (Слика 17). У *auth* фолдеру су смештене класе које се баве ауторизацијом и токенима. *Config* фолдер садржи класе које се баве конфигурацијом. *Util* фолдер садржи класу за рад са *JSON* веб токенима. У *exception* фолдеру су класе које моделују изузетке које се могу јавити у апликацији. *Data Transfer Objects* класе које моделују податке за размену са клијентском апликацијом се налазе у *dto* фолдеру.



Слика 17 Структура серверског дела

Hibernate [8] је коришћен за објектно-релационо мапирање. Интерфејси репозиторијума који су смештени у *repository* фолдеру наслеђују и проширују *JpaRepository* интерфејс. Репозиторијуми који постоје у апликацији су:

- UserRepository
- RoleRepository
- QuestionRepository
- AnswerRepository
- ResultRepository
- DetectionTypeRepository
- DiagnosticTypeRepository

Методе репозиторијума се позивају у класама сервиса. Пример репозиторијума је приказан на Листинг 4 где су приказане методе *DetectionTypeRepository*-а.

```
@Repository
public interface DetectionTypeRepository extends
    JpaRepository<DetectionType, Long> {

    DetectionType getDetectionTypeByType(String type);
    DetectionType getDetectionTypeById(Long id);
    List<DetectionType>
    getDetectionTypeByDetectionGroup(DetectionTypeGroup group);
}
```

Листинг 4 Пример репозиторијума - *DetectionTypeRepository*

Сервиси су смештени у *service* фолдеру. Сервиси су имплементирани као интерфејси који имају своје имплементације. У класама сервиса је имплементирана пословна логика система. Сервиси који постоје у апликацији су:

- UserService
- QuestionService
- AnswerService
- ResultService
- DetectionTypeService
- DiagnosticService
- DiagnosticTemplateService
- PsychologistTemplateService
- ResultService

Пример интерфејса једног од сервиса дат је на Листинг 5 где су приказане све јавне методе *DiagnosticService*-а.

```
public interface DiagnosticService {
    ArrayList<String> getAllSymptoms(String diagnostic);
    ResultDTO getDiagnostics(List<AnswerDTO> answers, String
loggedInUser) throws IOException;
    List<DepressionMarkDTO> getResultForDepressionMark(String
userEmail);
    String makeNewRules(RuleDTO ruleDTO) throws IOException;
    void addTemplateRules(List<TemplateParamDTO> templateParamDTOs)
throws IOException;
}
```

Листинг 5 Пример сервиса - Diagnostic service

Клијентска и серверска апликација комуницирају тако што клијентски део шаље захтеве серверу, а сервер одговара на добијене захтеве. Захтеви клијентске апликације покрећу функције које су смештене у класама које су анотиране *@RestController* анотацијом. Контролери који постоје у систему су:

- AuthController
- UserController
- QuestionController
- DiagnosticController
- DetectionTypeController

- ReportController
- PsychologistController

Како су модел, кјар и сервис посебни *maven* [11] пројекти, потребно је у *pom.xml* фајл сервис пројекта додати одговарајуће зависности и повезати ове пројекте. У Листинг 6 приказане су зависности које су додате у *pom.xml* фајл како би се повезали модел и кјар пројекат са сервис пројектом.

```
<dependency>
  <groupId>com.ftn.sbnz</groupId>
  <artifactId>kjar</artifactId>
  <version>0.0.1-SNAPSHOT</version>
</dependency>

<dependency>
  <groupId>com.ftn.sbnz</groupId>
  <artifactId>model</artifactId>
  <version>0.0.1-SNAPSHOT</version>
</dependency>
```

Листинг 6 Зависности модел и кјар пројектс у *pom.xml* фајлу

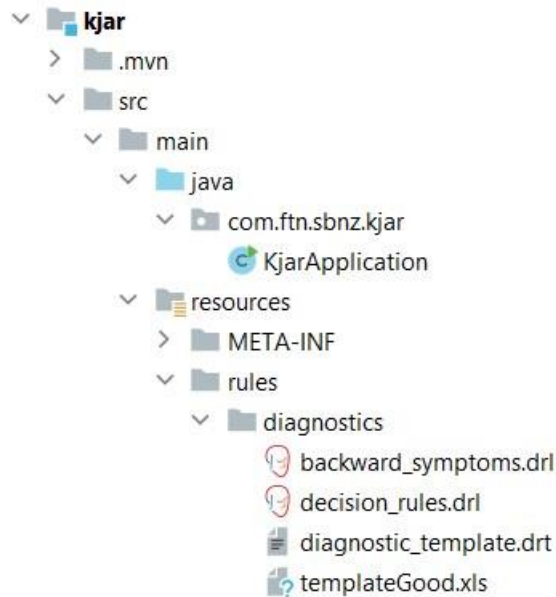
За инстанцирање сесије која омогућава додавања чињеница у базу знања и покретање модула за закључивање је потребно додати инстанцу *KieContainer*-а. Зато је у *main* класу потребно додати код из Листинг 7.

```
@Bean
public KieContainer kieContainer() {
    KieServices ks = KieServices.Factory.get();
    KieContainer kContainer = ks
        .newKieContainer(ks.newReleaseId("com.ftn.sbnz", "kjar",
            "0.0.1-SNAPSHOT"));
    KieScanner kScanner = ks.newKieScanner(kContainer);
    kScanner.start(1000);
    return kContainer;
}
```

Листинг 7 Код који омогућава инстанцирање *KieContainer*-а

5.3 Drools апликација

Систем базиран на правилима који доноси одлуке на основу *if-then* правила имплементиран је као засебни *maven* пројекат и користи *Drools* алат. Структура кјар пројекта приказана је на Слика 18. Сви фајлови са правилима смештени су пакет *resources/rules/diagnostics*.



Слика 18 Структура кјар пројекта

Анализа одговора на питања почиње правилом за трансформацију одговора. Правило проверава да ли је питање постављено у позитивној или негативној форми и на основу тога модификује вредност одговора. Одговори се дају по Ликертовој скали [12] у вредностима од 1, што значи *уопште се не слажем*, до 5 односно у *потпуности се слажем*. На Листинг 8 је приказано правило за трансформацију. У *when* делу се траже објекти класе *FullAnswer* чији је атрибут *positive* постављен на *false* вредност што значи да је питање негативно формулисано и да је потребно трансформисати вредност одговора. У *then* делу се вредност одговора модификује и поставља се вредност *positive* атрибута на *true*.

```
rule "transform-answer-diagnostic"
no-loop
agenda-group "transform-diagnostic"
when
    $a: FullAnswer(positive==false, $s: score)
then
    modify($a){setScore(5-$s);}
    modify($a){setPositive(true);}
end
```

Листинг 8 Правило за трансформацију одговора за питања постављена у негативној форми

Група правила која обрађује одговоре након трансформације имплементирана је као темплејт. За исту групу питања, односно за питања која дијагностикују исто ментално стање и припадају истом слоју упитника примењује се правило.

Први слој питања садржи питања која детектују назнаке односно услове за анксиозност, паничне нападе и социјалну анксиозност. На основу одговора из првог упитника формира се наредни упитник који садржи питања из области која су у претходном имала висок збир бодова, односно чија стања су дијагностикована као тешка или дубока што је уједно и симптом за дијагностику наредног стања.

Темплејт је приказан на Листинг 9 где се на основу збира одговора у радну меморију убацује објекат класе *Diagnostic* са одговарајућим атрибутима. У заглављу темплејта су наведени атрибути које користи темплејт као што су *type*, *layer*, *intensity*, *min*, *max* и *finalResult*. Након импорт секције наведен је темплејт на основу ког се генеришу правила. У *when* делу се тражи објекат класе *User*. Затим се рачуна сума свих одговора за објекте *FullAnswer* класе који припадају већ пронађеном кориснику и који имају одговарајуће вредности из темплејт података за ментално стање које се испитује и слој питања. Израчуната сума се пореди са минималном и максималном вредности из темплејт података и на основу тога се убацује објекат *Diagnostic* класе у радну меморију. Објекат *Diagnostic* класе садржи *id* корисника, детектовано ментално стање, интензитет, слој питања, текст, наредни слој питања и препоруку за детекцију наредног стања и информацију да ли је детектовано стање крајње стање.

За одређивање дијагнозе услова за неко од претходно наведених менталних стања, потребно је да у радној меморији постоји објекат класе *Diagnostic* који указује на неко дијагностиковано стање и да не постоје објекти исте класе који би указивали на присутност још неког стања за истог корисника. Пример правила које доноси ту одлуку приказано је у Листинг 10. У *when* делу правила се тражи објекат класе *Diagnostic* чија питања су из првог слоја и дијагностикују услове за анксиозност, односно чија вредност интензитет атрибута је тешко или дубоко. Потребно је да у радној меморији не постоје *Diagnostic* објекти чија питања су из другог и трећег слоја односно која би детектовала анксиозност и генерални анксиозни поремећај. Ако су услови задовољени извршава се *then* део и модификује постојећи *Diagnostic* објекат тако што се сетују вредности које указују на наредни слој питања и наредно стање које треба испитати.

За дијагностиковање анксиозности, паничног напада и социјалне анксиозности потребно је да је већ дијагностикован услов за дато ментално стање и да је упитник који представља то ментално стање обрађен и окарактерисан као тешко или дубоко стање. Пример за правило које дијагностикује анксиозност приказано је у Листинг 11. У *when* делу се траже два објекта *Diagnostic* класе. Први се односи на услове за анксиозност односно први слој питања, а други на анксиозност односно други слој питања. За оба објекта је потребно да имају интензитет постављен на тешко или дубоко. У радној меморији не треба да постоји објекат класе *Diagnostic* који описује трећи слој питања. Ако је правило задовољено постављају се вредности за наредни сет питања и наредно стање које је потребно испитати.

За дијагностиковање генералног анксиозног поремећаја, паничног поремећаја и социјалне фобије потребно је да је претходно утврђено одговарајуће ментално стање, услови за то стање и обрађен одговарајући упитник за финално стање. Пример правила за дијагностиковање генералног анксиозног поремећаја приказано је на Листинг 12. Потребно је да у радној меморији постоје три објекта класе *Diagnostic* и да сва три објекта припадају истом кориснику. Такође је потребно да сваки од објеката представља први, други или трећи слој питања односно услове за анксиозност, анксиозност и генерални анксиозни поремећај и да је за сва три објекта вредност интензитета

постављена на тешко или дубоко. Акција правила поставља вредност атрибута *finalResult* на *true* што показује да је детектован најтежи облик менталног стања.

Корисник попуњава упитнике један по један и након сваког упитника добија резултате и сазнања о свом менталном стању уз питање да ли жели да настави даље. Питања у упитницима се формирају динамички на основу претходних одговора. Ако је корисник имао благе услове за анксиозност, а тешке или дубоке услове за панични напад, следећи упитник ће се састојати само од питања која дијагностикују панични напад. По истом принципу као на Листинг 10, Листинг 11 и Листинг 12, писана су правила и за остала ментална стања.

```

template header
type
layer
intensity
min
max
finalResult
text

package diagnostics;

import com.ftn.sbnz.model.QuestionLayer;
import com.ftn.sbnz.model.Question;
import com.ftn.sbnz.model.Intensity;
import com.ftn.sbnz.model.User;
import com.ftn.sbnz.model.helper.FullAnswer;
import com.ftn.sbnz.model.helper.Diagnostic;

template "templ-1"
rule "Diagnostic_@{row.rowNumber}"
agenda-group "first-diagnostic"

when
    $u: User($uId: id)
    $total: Number() from accumulate(
        $a1: FullAnswer(
            user.id==$uId,
            question.detectionType.getType() == "@{type}",
            question.questionLayer == QuestionLayer.@{layer},
            $s: score
        ),
        sum($s)
    )
    eval($total.intValue() >= @{min} && $total.intValue() <=
@{max})

then
    insert(new Diagnostic($uId, Intensity.@{intensity}, "@{type}",
QuestionLayer.@{layer}, null, null, "@{text}", @{finalResult}));
end

end template

```

Листинг 9 Темплејт који генерише правила за обраду одговора на питања

```

rule "r1-anksioznost"
agenda-group "result-diagnostic"
  when
    $d: Diagnostic($uId: userId,
                  detectionType == "USLOVI_ZA_ANKSIOZNOST",
                  questionLayer == QuestionLayer.FIRST,
                  finalResult == false,
                  (intensity == Intensity.TESKO || intensity ==
Intensity.DUBOKO))

    not (Diagnostic(userId == $uId, detectionType ==
"ANKSIOZNOST", questionLayer==QuestionLayer.SECOND))
    not (Diagnostic(userId == $uId,
detectionType == "GENERALNI_ANKSIOZNI_POREMECAJ",
questionLayer==QuestionLayer.THIRD))
  then
    modify($d){setNextLayer(QuestionLayer.SECOND);}
    modify($d){setNextDetection("ANKSIOZNOST");}
  end

```

Листинг 10 Пример правила које дијагностикује услове за анксиозност

```

rule "r2-anksioznost"
agenda-group "result-diagnostic"
  when
    $d1: Diagnostic($uId: userId,
                  detectionType == "USLOVI_ZA_ANKSIOZNOST",
                  questionLayer == QuestionLayer.FIRST,
                  finalResult == false,
                  (intensity == Intensity.TESKO || intensity ==
Intensity.DUBOKO))
    $d2: Diagnostic(userId == $uId,
                  detectionType == "ANKSIOZNOST",
                  questionLayer == QuestionLayer.SECOND,
                  finalResult == false,
                  (intensity == Intensity.TESKO || intensity ==
Intensity.DUBOKO))

    not (Diagnostic(userId == $uId,
detectionType == "GENERALNI_ANKSIOZNI_POREMECAJ",
questionLayer==QuestionLayer.THIRD))
  then
    modify($d2){ setNextLayer(QuestionLayer.THIRD);}
    modify($d2){
      setNextDetection("GENERALNI_ANKSIOZNI_POREMECAJ");}
  end

```

Листинг 11 Правило које дијагностикује анксиозност

```

rule "r3-anksioznost"
agenda-group "result-diagnostic"
  when
    $d1: Diagnostic($uId: userId,
                    detectionType == "USLOVI_ZA_ANKSIOZNOST",
                    questionLayer == QuestionLayer.FIRST,
                    (intensity == Intensity.TESKO || intensity ==
Intensity.DUBOKO))
    $d2: Diagnostic(userId == $uId,
                    detectionType == "ANKSIOZNOST",
                    questionLayer == QuestionLayer.SECOND,
                    (intensity == Intensity.TESKO || intensity ==
Intensity.DUBOKO))
    $d3: Diagnostic(userId == $uId,
                    detectionType == "GENERALNI_ANKSIOZNI_POREMECAJ",
                    questionLayer == QuestionLayer.THIRD,
                    (intensity == Intensity.TESKO || intensity ==
Intensity.DUBOKO))
  then
    modify($d3){ setFinalResult(true);}
  end
end

```

Листинг 12 Пример правила које дијагностикује генерални анксиозни поремећај

За добијање листе симптома, односно стања која морају бити идентификована пре стања за које се врши упит се користе правила. Рекурзивни упит који претражује сва ментална стања која су симптом за дато је приказан у Листинг 13. Упит прима два параметра. Први параметар представља симптом, а други представља ментално стање. Тражи се *DiagnosticState* објекат који садржи прослеђене параметре. Ако такав објекат не постоји тражи се објекат који може имати било коју вредност за симптом и дато ментално стање и позива се упит рекурзивно тако што се шаље вредност за симптом, а за ментално стање је могућа било која вредност.

```

query getSymptomsOf(String symptom, String mentalState)
  DiagnosticState( symptom, mentalState; )
  or
  ((DiagnosticState(x, mentalState; )) and (getSymptomsOf
(symptom, x; )))
end

```

Листинг 13 Упит за претраживање симптома за дато ментално стање

Правило које позива упит из Листинг 13 је приказано у Листинг 14. У *when* делу се тражи стринг и покреће се упит тако што пронађени стринг представља ментално стање за које се траже симптоми. У *then* делу се убацују сви пронађени симптоми у радну меморију. Пре покретања правила потребно је у радну меморију убацити чињенице у *Java* коду. Код је приказан у Листинг 15 и представља убацавање одговарајућих чињеница у радну меморију.

```

rule "get all symptoms"
when
    $s:String();
    getSymptomsOf( thing, $s ; )
then
    insert(thing);
end

```

Листинг 14 Правило које покреће претрагу симптома

```

public ArrayList<String> getAllSymptoms(String diagnostic) {

    kieSession.insert(new
DiagnosticState("USLOVI_ZA_ANKSIOZNOST", "ANKSIOZNOST"));

    kieSession.insert(new DiagnosticState("ANKSIOZNOST",
"GENERALNI_ANKSIOZNI_POREMECAJ"));

    kieSession.insert(new
DiagnosticState("USLOVI_ZA_PANICNI_NAPAD", "PANICNI_NAPAD"));
    kieSession.insert(new DiagnosticState("PANICNI_NAPAD",
"PANICNI_POREMECAJ"));
    kieSession.insert(new
DiagnosticState("USLOVI_ZA_SOCIJALNU_ANKSIOZNOST",
"SOCIJALNA_ANKSIOZNOST"));
    kieSession.insert(new
DiagnosticState("SOCIJALNA_ANKSIOZNOST", "SOCIJALNA_FOBIJA"));

    FactHandle symptom = kieSession.insert(diagnostic);
    int fired = this.kieSession.fireAllRules();
    this.kieSession.delete(symptom);

    ...

}

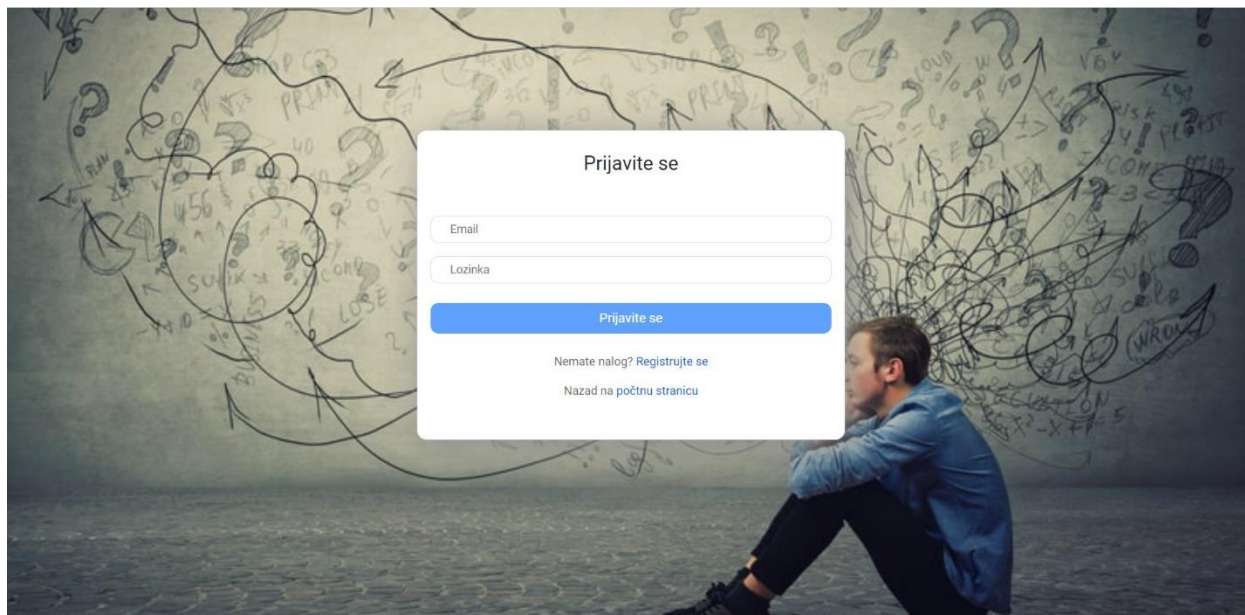
```

Листинг 15 Убацивање чињеница у ранду меморију

6. Приказ имплементираног система

У овом поглављу ће бити приказано решење из перспективе оба типа корисника клијента и психолога.

На Слика 19 је приказана почетна страница апликације на којој се могу пријавити претходно регистровани корисници тако што унесу свој мејл и лозинку.



Слика 19 Пријава на систем

У случају да корисник није регистрован на систем, може се усмерити на страницу за регистрацију кликом на *Registrujte se*. Пример попуњене форме за регистрацију приказан је на Слика 20 где су унети основни подаци о кориснику.

Слика 20 Форма за регистрацију

Након пријаве на систем клијенту је доступан упитник. На свако питање је могуће одговорити бројчано од 1 – *уопште се не слажем* до 5 – *у потпуности се слажем*. На Слика 21 је приказан упитник.

Dijagnostika

U upitniku su pitanja i za svako pitanje odgovorite jednim od brojeva.
1 - uopšte se ne slažem, 2 - ne slažem se 3 - niti se slažem, niti se ne slažem, 4 - slažem se, 5 - u potpunosti se slažem

Pitanja

Imam utisak da mogu da kontrolišem svoju brigu.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Iako su određene situacije neprijatne, ja ih mogu podneti.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

U određenim situacijama osećam tenziju.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Ne predstavlja mi problem da se fokusiram na određene zadatke.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Iako se ponekad ne osećam najbolje, nemam utisak da sam na „ivici pucanja“

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Items per page: 5 1 - 5 of 17

Pošalji upitnik

Слика 21 Упитник за клијента

Након слања датих одговора на постављена питања, добијају се резултати. Резултати упитника након првог слоја питања приказани су на Слика 22.

Dijagnostika

Prikazani su rezultati upita za

1. Uslovi za panični napad
Primećeni su uslovi za panični napad. Naredni set pitanja će sadržati i pitanja o paničnom napadu.

2. Uslovi za socijalnu anksioznost
Primećeni su uslovi za socijalnu anksioznost. Naredni set pitanja će sadržati i pitanja o socijalnoj anksioznosti.

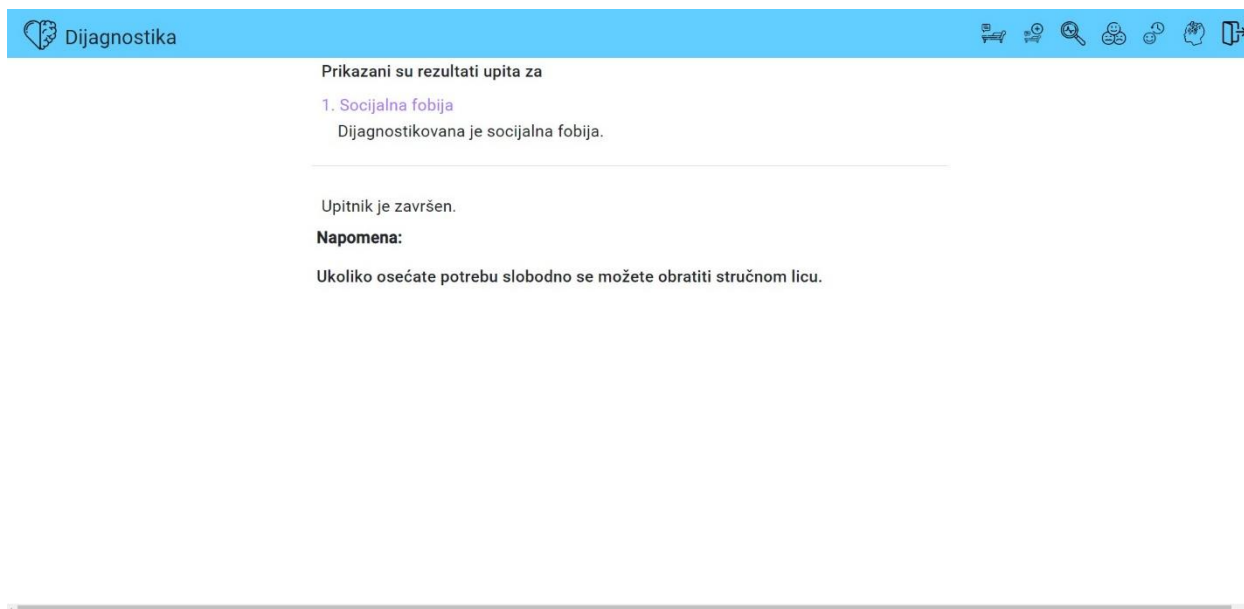
3. Uslovi za anksioznost
Nemate uslova za anksioznost. Nemate razloga za brigu.

Da li želite da nastavite upitnik?

Napomena:
Ukoliko osećate potrebu slobodno se možete obratiti stručnom licu.

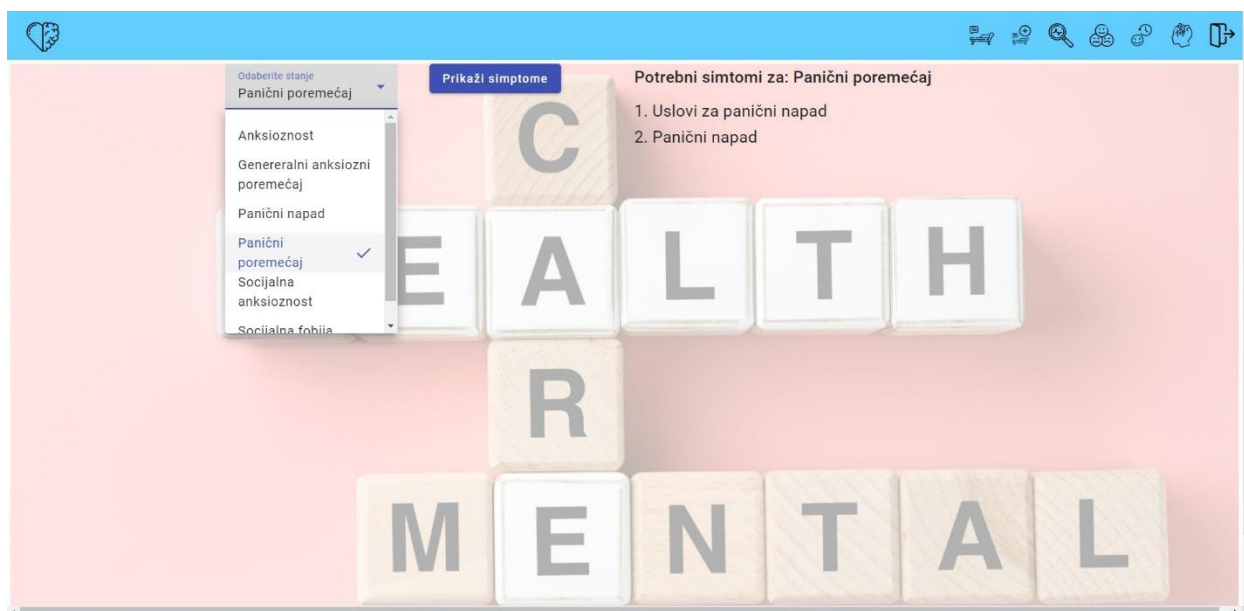
Слика 22 Приказ резултата након првог слоја питања

Кликом на *Nastavi upitnik* добија се нови сет питања који ће наставити да пропитује клијента о оним менталним стањима за које има предиспозиције. Када корисник прође све кругове питања добиће и крајњу дијагностику. На Слика 23 је приказан крај упитника.



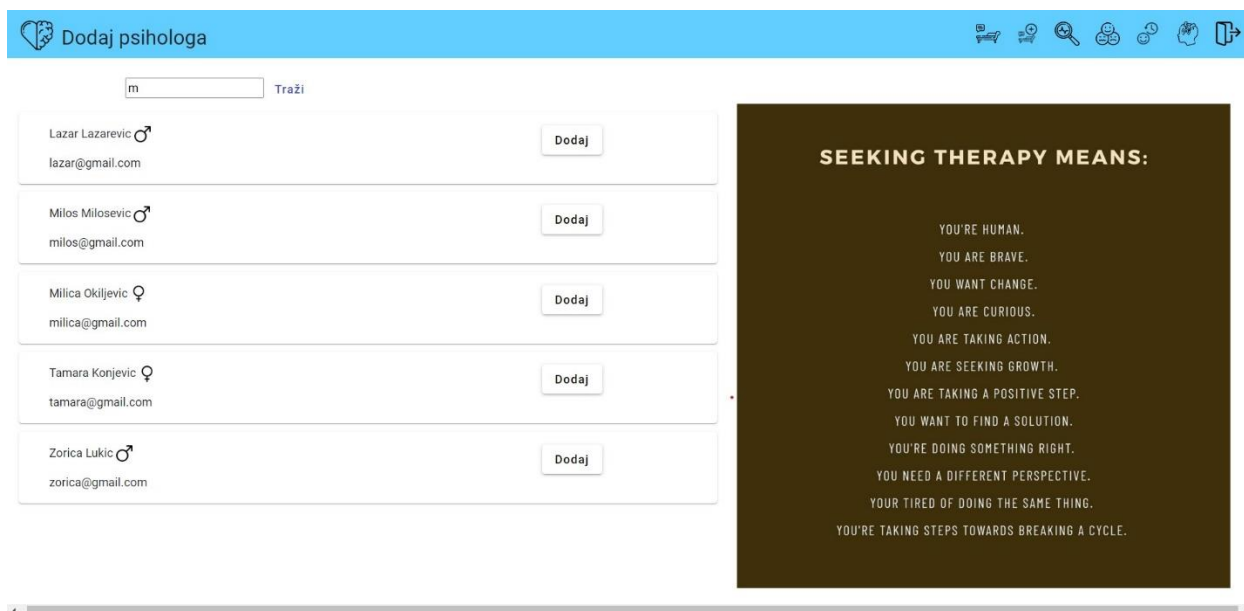
Слика 23 Приказ завршетка упитника

Клијент може да тражи симптоме за ментална стања која се могу дијагностиковати у апликацији. Упит за приказ симптома за панични поремећај је приказан на Слика 24.



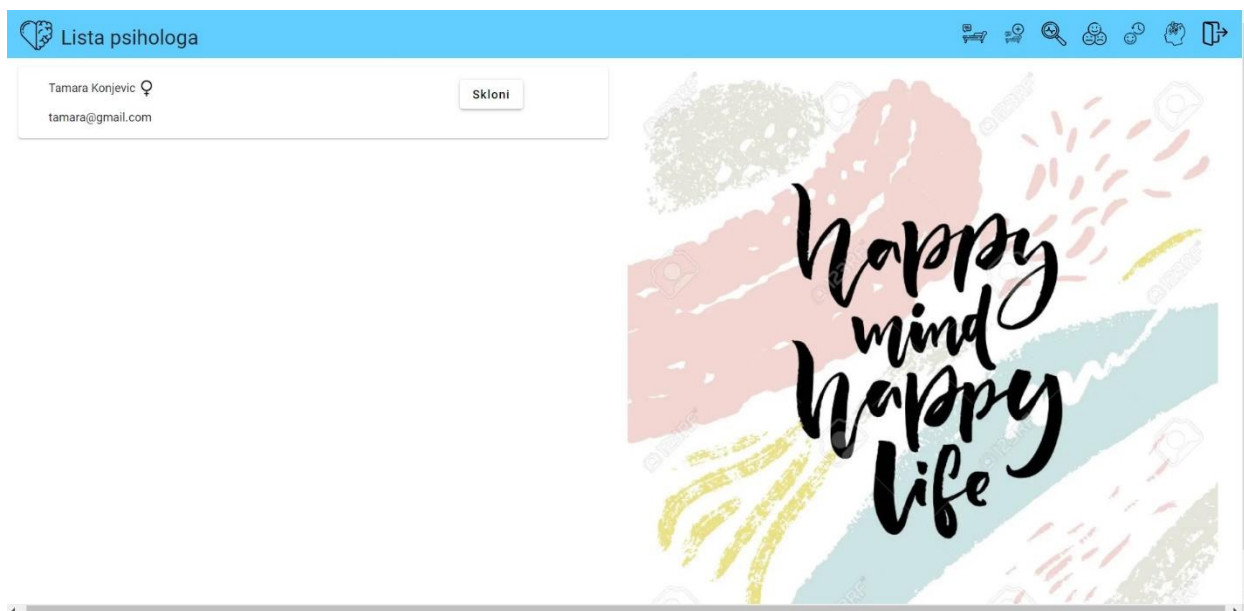
Слика 24 Страница за упит симптома

Претрага психолога и омогућавање приступа одговорима на питања и резултатима упитника приказано је на Слика 25.



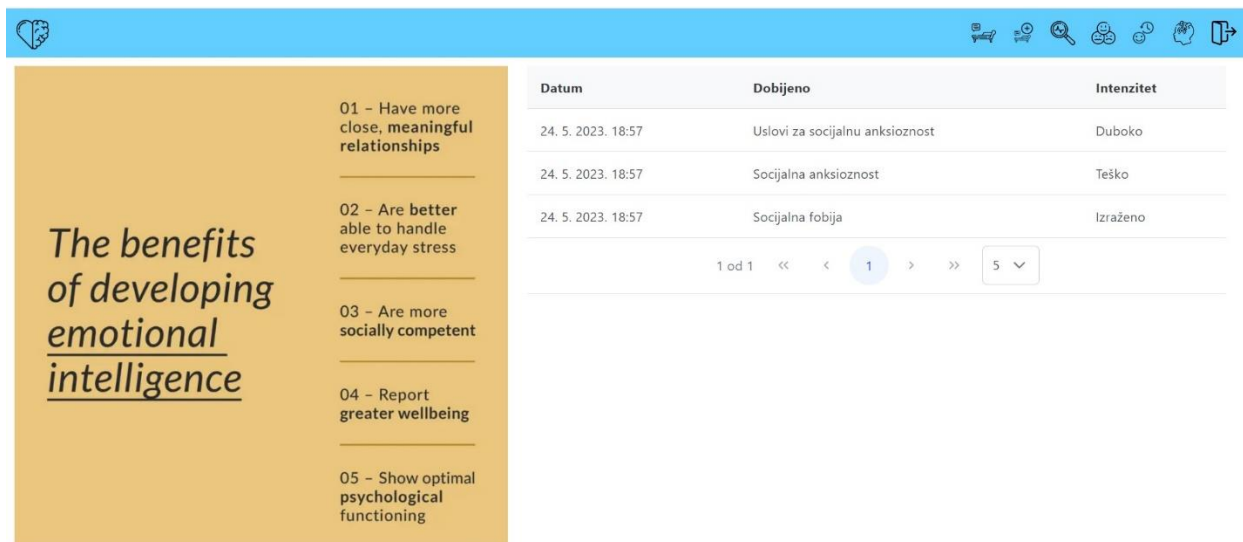
Слика 25 Претрага и додавање психолога

Приказ психолога који имају приступ клијентовим резултатима приказан је на Слика 26.



Слика 26 Приказ додатих психолога

Преглед свих добијених резултата приказан је на Слика 27. Кликом на неки од резултата добија се приказ свих одговора на питања која се тичу дијагностикованог менталног стања као што је приказано на Слика 28.



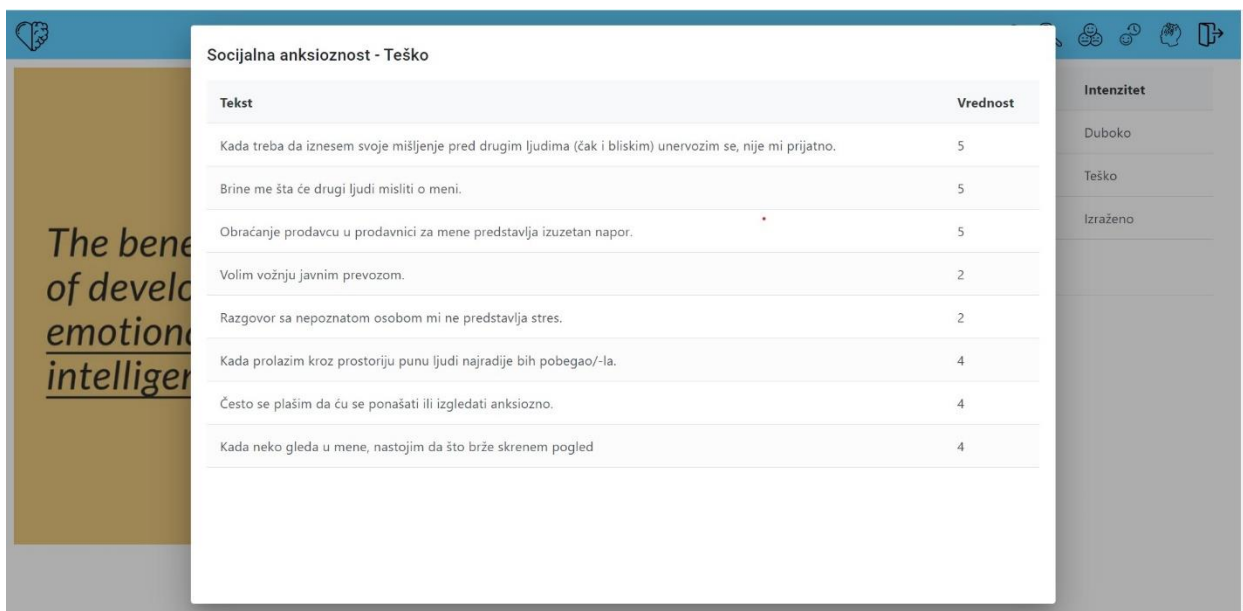
The benefits of developing emotional intelligence

- 01 – Have more close, meaningful relationships
- 02 – Are better able to handle everyday stress
- 03 – Are more socially competent
- 04 – Report greater wellbeing
- 05 – Show optimal psychological functioning

Datum	Dobijeno	Intenzitet
24. 5. 2023. 18:57	Uslovi za socijalnu anksioznost	Duboko
24. 5. 2023. 18:57	Socijalna anksioznost	Teško
24. 5. 2023. 18:57	Socijalna fobija	Izraženo

1 od 1 << < 1 > >> 5 ▾

Слика 27 Приказ добијених резултата



The benefits of developing emotional intelligence

Socijalna anksioznost - Teško

Tekst	Vrednost
Kada treba da iznesem svoje mišljenje pred drugim ljudima (čak i bliskim) unervozim se, nije mi prijatno.	5
Brine me šta će drugi ljudi misliti o meni.	5
Obraćanje prodavcu u prodavnici za mene predstavlja izuzetan napor.	5
Volim vožnju javnim prevozom.	2
Razgovor sa nepoznatom osobom mi ne predstavlja stres.	2
Kada prolazim kroz prostoriju punu ljudi najradije bih pobegao/-la.	4
Često se plašim da ću se ponašati ili izgledati anksiozno.	4
Kada neko gleda u mene, nastojim da što brže skrenem pogled	4

Intenzitet

Duboko

Teško

Izraženo

Слика 28 Детаљнији приказ резултата са питањима и одговорима

Психолог има могућност да погледа резултате за оне пацијенте који су му одобрили приступ. Приказ једног од резултата је приказан на Слика 29. Преглед одговора на питања је приказан на Слика 28 .

Izveštaj za pojedinačnog korisnika

Izveštaj po zanimanju, starosti, polu

Izveštaj po mentalnim stanjima

Odaberite klijenta
Maša Mašić

Prikaži

Datum	Stanje	Intenzitet
16. 6. 2023. 13:30	Uslovi za anksioznost	Teško
16. 6. 2023. 13:30	Anksioznost	Teško
16. 6. 2023. 14:30	Generalni anksiozni poremećaj	Umereno

1 od 1 << < 1 > >> 5

Слика 29 Приказ извештаја за појединачног корисника

Друга врста извештаја је извештај на основу опсега године рођења, занимања и пола. Извештај представља детектована стања и укупан број. Пример таквог извештаја је дат на Слика 30.

Izveštaj za pojedinačnog korisnika

Izveštaj po zanimanju, starosti, polu

Izveštaj po mentalnim stanjima

Početna godina rođenja
1990

Krajnja godina rođenja
2000

Odaberite zanimanje
student

Odaberite pol
Oba

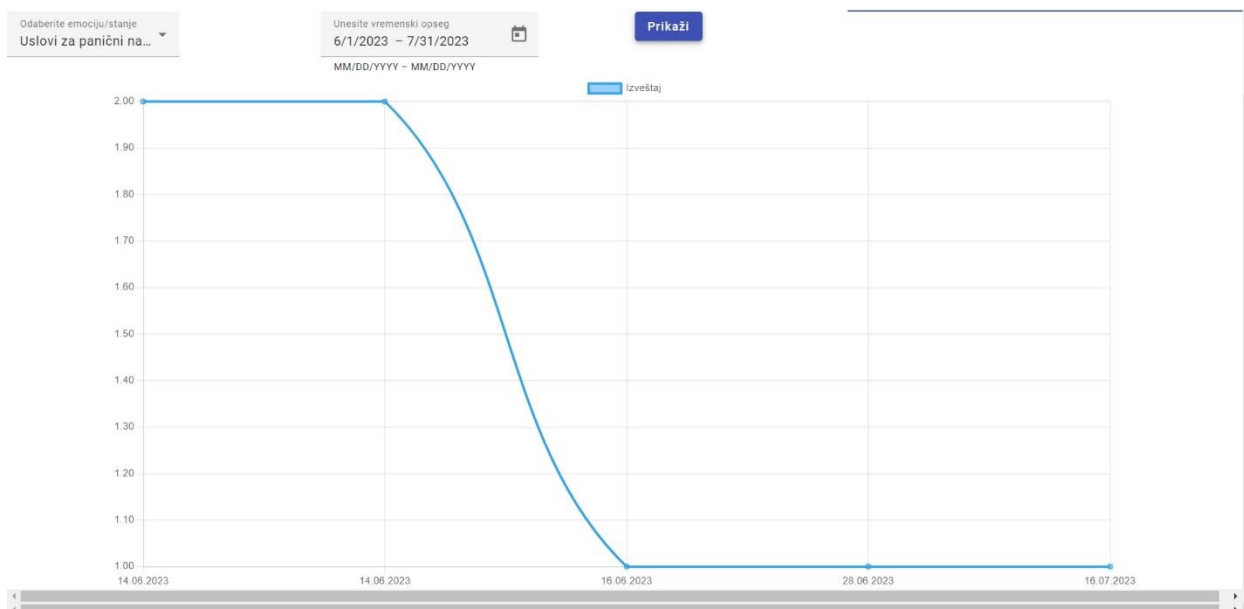
Prikaži

1. Uslovi za panični napad	4
2. Socijalna fobija	2
3. Tužno	1
4. Uslovi za socijalnu anksioznost	5
5. Socijalna anksioznost	2
6. Panični napad	1
7. Uslovi za anksioznost	4

1 od 1 << < 1 > >> 5

Слика 30 Извештај на основу године рођења, занимања и пола

Праћење детектованих стања кроз време приказано је на Слика 31 где се на основу одабраног стања и временског периода добије график.




Слика 31 Извештај за ментално стање кроз време





Психолог има могућност да додаје нова питања која ће се уврстити у упитник. Пример попуњене форме за додавање нових питања приказан је на Слика 32.

The screenshot shows a form titled 'Definisanje novih pitanja' (Defining new questions). It has a blue header bar with a heart icon and the title. Below the header, there are two rows of input fields. The first row is for 'Pitanje 1' (Question 1) and the second row is for 'Pitanje 2' (Question 2). Each row has a 'Sloj pitanja*' (Question layer*) dropdown menu, a 'Da li je pitanje pozitivno formulisano?' (Is the question positively formulated?) radio button group, and a 'Dijagnoza*' (Diagnosis*) dropdown menu. For 'Pitanje 1', the 'Sloj pitanja*' is 'Prvi' (First), the radio button for 'Da' (Yes) is selected, and the 'Dijagnoza*' is 'Depresija' (Depression). For 'Pitanje 2', the 'Sloj pitanja*' is 'Drugi' (Second), the radio button for 'Ne' (No) is selected, and the 'Dijagnoza*' is 'Drugo' (Other). There is a 'Dodaj pitanja' (Add questions) button at the bottom. A 'Nova dijagnoza' (New diagnosis) button is also visible on the right.

Слика 32 Додавање нових питања/дијагноза у упитник

Дефинисање темплејта на основу ког се доносе одлуке је приказано на Слика 33.

 Dodavanje novog templejta

Dijagnoza* Depresija	Sloj pitanja* Prvi	Minimalni* 1	Maksimalni* 9
Nivo ozbiljnosti* Blago	Poruka* Dijagnosticovana je blaga depresija.	Da li je krajnji rezultat? <input checked="" type="radio"/> Da <input type="radio"/> Ne	
Dijagnoza*	Sloj pitanja*	Minimalni* 0	Maksimalni* 0
Nivo ozbiljnosti*	Poruka*	Da li je krajnji rezultat? <input type="radio"/> Da <input type="radio"/> Ne	

+

-

Dodaj pravila

Слика 33 Додавање нових templejта

7. Закључак

У раду је представљен систем за дијагностику менталног здравља пацијената као широко доступна апликација и помоћно средство у раду између клијента и психолога. Систем је имплементиран коришћењем експертског знања представљеног у виду продукционих правила. Анализирано је тренутно стање у области и популарне апликације, описане су технологије које су коришћене приликом имплементације, приложена је спецификација система описана кроз *UML* дијаграме. Објашњена је детаљна имплементација система који се састоји из четири дела. У систему базираном на правилима интегрисано је знање експерта коришћењем *Drools* алата. Упутство за коришћење апликације за сваку од улога – клијента и психолога приказано је у приказу имплементираних система.


Систем тренутно дијагностикује анксиозност, генерални анксиозни поремећај, панични напад, панични поремећај, социјалну анксиозност и социјалну фобију. Систем је могуће проширити новим правилима која би анализирали повезаност између наведених менталних стања и депресије јер се често у наведеним менталним стањима може идентификовати и стање депресије у мањој или већој мери. Такође, могуће проширење система је *live chat* који би служио за комуникацију између клијента и психолога.


Литература

- [1] “ПРОГРАМ О ЗАШТИТИ МЕНТАЛНОГ ЗДРАВЉА У РЕПУБЛИЦИ СРБИЈИ ЗА ПЕРИОД 2019-2026. ГОДИНЕ,” n.d.
- [2] Wysa, <https://www.wysa.com>
- [3] Angular, <https://angular.io/docs>
- [4] Spring, <https://docs.spring.io/spring-framework/docs/4.0.x/spring-framework-reference/html/overview.html>
- [5] REST, https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
- [6] REST <https://www.ibm.com/topics/rest-api>
- [7] Drools, https://docs.jboss.org/drools/release/7.35.0.Final/drools-docs/html_single/
- [8] Hibernate, <https://hibernate.org/>
- [9] UML diagrami, <https://www.uml-diagrams.org/>
- [10] Milosavljević, Gordana. *Uvod u modelovanje softvera*. Novi Sad: Fakultet tehničkih nauka, 2020.
- [11] Maven, <https://maven.apache.org/what-is-maven.html>
- [12] Liketova skala, <https://www.scribbr.com/methodology/likert-scale/>
- [13] World Health Organization, и Disease Control Priorities Project, *yp. Disease control priorities related to mental, neurological, developmental and substance abuse disorders*. Geneva: World Health Organization, 2006.
- [14] Barlow, David H. *Anxiety and Its Disorders: The Nature and Treatment of Anxiety and Panic*. 2. ed., Paperback ed. New York, NY: Guilford Press, 2004.
- [15] Craske, Michelle G., Scott L. Rauch, Robert Ursano, Jason Prenoveau, Daniel S. Pine, и Richard E. Zinbarg. „What Is an Anxiety Disorder?” *FOCUS* 9, изд. 3 (2011.): 369–88. <https://doi.org/10.1176/foc.9.3.foc369>.
- [16] Jain, Manav Pradeep, Soumyaprakash Sribash Dasmohapatra, и Stevina Correia. „Mental Health State Detection Using Open CV and Sentimental Analysis”. У *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, 465–70, 2020. <https://doi.org/10.1109/ICISS49785.2020.9315984>.
- [17] Windriyani, Paramaresthi, S Kom, ST Wiharto, M Kom, и Sari Widya Sihwi. „Expert system for detecting mental disorder with Forward Chaining method”, 1–7. IEEE, 2013.
- [18] Sari, Dewi Nirmala, Kemal N Siregar, и Hadi Pratomo. „Conceptual Model of an Expert System from Systematic Literature Review: Mental Health Screening in Pregnancy During the COVID-19 Pandemic”. *Women, Midwives and Midwifery* 2, изд. 3 (2022.): 51–63.

Биографија

Милица Петровић је рођена 21. априла 2000. године у Сремској Митровици. Завршила је Основну школу „Бранко Радичевић“ у Шиду. Гимназију „Јован Јовановић Змај“ завршила је у Новом Саду. Године 2019. уписала је Факултет техничких наука у Новом Саду, смер Софтверско инжењерство и информационе технологије. Све испите полаже и студије завршава у року, 2023. године.

	УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА 21000 НОВИ САД, Трг Доситеја Обрадовића 6	
	КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА	
Редни број, РБР:		
Идентификациони број, ИБР:		
Тип документације, ТД:	Монографска публикација	
Тип записа, ТЗ:	Текстуални штампани документ	
Врста рада, ВР:	Дипломски рад	
Аутор, АУ:	Милица Петровић	
Ментор, МН:	доц. Др Синиша Николић, ФТН Нови Сад	
Наслов рада, НР:	Систем за дијагностику менталног стања пацијента	
Језик публикације, ЈП:	Српски	
Језик извода, ЈИ:	Српски/енглески	
Земља публикавања, ЗП:	Србија	
Уже географско подручје, УГП:	Војводина	
Година, ГО:	2023	
Издавач, ИЗ:	Ауторски репринт	
Место и адреса, МА:	Факултет техничких наука, Трг Доситеја Обрадовића 6, Нови Сад	
Физички опис рада, ФО:	бр. Поглавља 7/страница 57/цитата 18/табела 0/слика 48	
Научна област, НО:	Електротехничко и рачунарско инжењерство	
Научна дисциплина, НД:	Системи базирани на знању	
Предметна одредница/кључне речи, ПО:	системи базирани на правилима, анализа упитника, представа доменског знања	
УДК		
Чува се, ЧУ:	Библиотека ФТН, Трг Доситеја Обрадовића 6, Нови Сад	
Важна напомена, ВН:		
Извод, ИЗ:	У раду је описан систем за менталну подршку који омогућава клијентима дијагностиковање менталних стања, коришћењем система базираних на правилима. Детаљно је описана спецификација и имплементација система. Приказано је коришћење апликације од стране психолога и клијента.	
Датум прихватања теме, ДП:		
фДатум одбране, ДО:		
Чланови комисије, КО:	Председник:	Др Драган Ивановић, ред. проф., ФТН Нови Сад
	Члан:	Др Марко Марковић, доц., ФТН Нови Сад
	Члан, ментор:	Др Синиша Николић, доц., ФТН Нови Сад
		Потпис ментора

	UNIVERSITY OF NOVI SAD • FACULTY OF TECHNICAL SCIENCES 21000 NOVI SAD, Trg Dositeja Obradovića 6	
	KEY WORDS DOCUMENTATION	
Accession number, ANO :		
Identification number, INO :		
Document type, DT :	Monographic publication	
Type of record, TR :	Textual material	
Contents code, CC :	Bachelor thesis	
Author, AU :	Milica Petrovic	
Mentor, MN :	Sinisa Nikolic, PhD	
Title, TI :	System for diagnosing the patient's mental states	
Language of text, LT :	Serbian	
Language of abstract, LA :	Serbian/English	
Country of publication, CP :	Serbia	
Locality of publication, LP :	Vojvodina	
Publication year, PY :	2023	
Publisher, PB :	Author's reprint	
Publication place, PP :	Faculty of Technical Sciences, Trg Dositeja Obradovića 6, Novi Sad	
Physical description, PD :	no. of chapters 7/pages 57/quotes 18/tables 0/pictures 48	
Scientific field, SF :	Electrical and computer engineering	
Scientific discipline, SD :	Knowledge based systems	
Subject/Key words, S/KW :	Rule based systems, questionnaire analysis, representation of domain knowledge	
UC		
Holding data, HD :	Library of Faculty of Technical Sciences, Trg Dositeja Obradovića 6, Novi Sad	
Note, N :		
Abstract, AB :	This paper describes a system for mental support that enables clients to diagnose mental states, using rule-based system. The specification and implementation of the system are described in detail. The use of application by the psychologist and the client is shown.	
Accepted by the Scientific Board on, ASB :		
Defended on, DE :		
Defended Board, DB :	President:	Dragan Ivanović, PhD, prof., FTN Novi Sad
	Member:	Marko Marković, PhD, assist. prof., FTN Novi Sad
	Member, Mentor:	Siniša Nikolić, PhD, assist. prof., FTN Novi Sad
		Menthor's sign