



Универзитет у Крагујевцу
Факултет инжењерских наука



Назив студијског програма: Рачунарска техника и софтверско инжењерство

Ниво студија: Основне академске студије

Предмет: Е-пословање

Број индекса: 562/2015

Никола М. Самарцић

**Развој Интернет апликације за управљање процесима и
ресурсима базираној на RFID технологији**

дипломски рад

Комисија за преглед и одбрану:

1. проф. др Миладин Стефановић
2. _____
3. _____

Датум одбране: _____

Оцена: _____

У оквиру овог мастер рада кандидат треба да:

Дефинише технологије које ће користити, као и архитектуру апликације, опис реалног система, опис развојног модела и развојног софтвера. Треба обезбедити одговарајуће услове за тестирање апликације. Кандидат треба да развије апликацију за управљање процесима и ресурсима базираној на *RFID* технологији. Потребно је јасно дефинисати примену поменуте апликације, као и бољитке и предности у односу на остале системе сличне природе. Закључним разматрањем описати интеграцију система и могућности имплементација.

Крагујевац, 22.05.2019

Ментор:

проф. др Миладин Стефановић

Садржај

1	Увод.....	7
2	Теоријске основе и преглед технологије.....	8
2.1	RFID технологија и њене примене.....	8
2.1.1	Како ради RFID?	10
2.2	RC522 сензор.....	13
2.2.1	RC522 RFID пинови.....	15
2.3	Raspberry PI 3	16
2.3.1	RPI 3 пинови.....	17
2.3.2	Повезивање RC522 сензора на RPI3.....	19
2.4	WAMP сервер.....	20
2.4.1	Apache HTTP сервер	21
2.4.2	MySQL.....	21
2.4.3	PHP	22
2.5	Laravel Интернет framework	23
2.5.1	MVC.....	23
3	Архитектура и имплементација апликације.....	24
3.1	Комуникација два подсистема.....	25
3.2	Структура система	26
4	Функционално моделирање	28
4.1	Дијаграм контекста	28
4.2	Стабло активности	29
5	Информационо моделирање	30
5.1	Дефинисање детаљних захтева.....	30
5.2	Креирање ЕР дијаграма	31
5.2.1	Идентификација ентитета и веза	31
5.2.2	Креирање атрибута	32
6	Дефинисање физичког слоја базе података.....	33
6.1	Схема базе података.....	35
7	UML дијаграми.....	36
7.1	Дијаграм активности.....	36

7.2	Дијаграм машине стања	38
7.3	Дијаграм слулајева коришћења	39
7.4	Дијаграм секвенци	41
7.4.1	Сваки корисник	41
7.4.2	Администратор	42
7.4.3	Професор	44
8	Приказ решења апликације	45
9	Закључак	54
10	Литература	55

Abstract:

This paper describes the implementation of an WEB application which tracks resources and processes using RFID technology. Whole system is divided into two smaller systems, one is hardware driven while the other one is softwer. The first mentioned small system is developed on the famous Raspberry PI 3 computer, while the other one is implemented in Laravel WEB framework. While reading this paper, the reader will gain technical knowledge not only of the system, but the aftermentioned technologies, and also would learn about the RFID technology and all the various ways it can be integrated. In the later part of the paper, the reader can gain knowledge of how the relational database is implemented, and will gain more insight into the usage of the application through UML diagrams. In the last chapter, reader will get a small how-to tutorial on the usage the application and all of its advantages.

Key words: RFID, Laravel, process, tracking, authentication, faculty

Резиме рада:

У раду су описане модерне технологије као што су *Laravel*, *Raspberry PI* и *RFID*. У уводном делу рада читалац ће бити упознат са свим поменутиим технологијама и њиховим примена, док ће каснија поглавља објаснити систем апликације, како она функционише и како су ове, наизглед, различите технологије у спрези и како зависе једне од других, чинећи јединствену целину. На самом крају рада, читалац ће добити кратке инструкције како користити апликацију (описане сликама), где ће видети све предности описаног система.

Кључне речи: *RFID*, *Laravel*, процеси, праћење, аутентикација, факултет

1 Увод

На ивици смо технолошке револуције која ће променити из корена наш начин живота, начин на који радимо и односимо се једни према другима. Последице ће утицати на то какви смо и како се односимо једни према другима, чак и у најудаљенијим деловима света. Револуција ће утицати на пословно тржиште, будућност посла, неједнакости у приходима, а њихови последњи трзаји утицаће на геополитичку сигурност и етичке одреднице. [1]

Параграф изнад је цитат Клауса Шваба, аутора књиге *Четврта индустријска револуција* из 2016. године. Није случајно да овај рад почиње таквим цитатом, свака даља тачка која описује овај пројекат спада у ту тзв. четврту индустријску револуцију.

Пројекат који ће бити описан овим радом јесте заправо дигитализација и део индустријске револуције. Тема овог дипломски рада представља јединствен систем за, условно речено, праћење запослених у оквиру неке институције, која је у овом случају факултет. *RFID* аутентикација и праћење је замишљена као један део много већег пројекта, прилагођеног акдамеским заједницама и институцијама. Целокупни пројекат би требало да још садржи дигиталну оверу/упис семестра, дигиталне потписе професора (за студенте), СМС пријаву испита, дигитални упис оцена, могућност праћења докумената када се пошаљу са једне локације на другу (нешто попут система праћења поштом наше поште, или великих светских коорпорација попут *AliExpress-a*) и много тога још.

Међутим, фокус овог рада биће на праћењу и аутентикацији корисника коришћењем *RFID* технологије, како теоријски, тако и практично. У првом делу овог рада читалац ће бити упућен у коришћене технологије, начин имплементирања Интернет апликације, и њене спреге са хардвером *RFID-a* и *Raspberry Pi-a*. Након теоријског увода, који читаоца припрема за разумевање целог система, представљена је структура система, као и архитектура истог, релациона база података као и начин коришћења апликације кроз UML дијаграме.

2 Теоријске основе и преглед технологије

2.1 *RFID технологија и њене примене*

Radio frequency identification – RFID је систем даљинског слања и пријема података помоћу картица/привеска. *RFID* картица је мали објекат који се може залепити или уградити у жељени производ или уређај, може се и користити као обична картица (као што је у овом раду) за контролу приступа одређеним просторијама. Поменуте картице садрже у себи антену која им омогућава пријем и слање радио-талаза од примопредајника.[2]

Већу примену, поготову у апликацијама за праћење, *RFID* је добио осамдесетих година прошлога века, због своје способности да прати покретне објекте. Прва оваква примена је у готово свим продавницама, где се налази на артиклима, и сузбија могућност крађе на минимум. Као префињена технологија, са неслућеним могућностима примене, он се стално развија.



Слика 1: Европски симбол *RFID-a*

RFID има предност у односу на остале, сличне технологије, јер омогућава праћење јединственог производа од његовог настанка до крајњег потрошача. Нпр. стандардни бар-код идентификује само произвођача и производ, али не и јединствени артикал. Бар-код на омоту чоколаде је исти на сваком омоту исте врсте чоколаде, те је немогуће путем самог бар-кода издвојити тачно одређени производ. *RFID* транспондер носи идентификатор – серијски број јединствен само за тај специфични производ. Када је реч о апликацији, описаној овим радом, свака картица има свој, уникатни серијски број, који је преписан тачно једном кориснику, и на основу тога има се увид кретања корисника (провлачењем своје картице кроз сензоре у одређеним просторијама). Нпр. професор може знати ко му је присуствовао предавању, јер да би студент отворио врата учионице мора прислонити своју картицу на сензор.

У већини окружења *RFID* постиже 99.5% од 100% читавања у првом скенирању. [2] Нема покретне делове или оптичке компоненте, те је одржавање далеко једноставније. Примена овог система је у почетној фази, не мора у потпуности заменити систем идентификације и праћења заснован на бар-коду, али га свакако може допунити.

2.1.1 Како ради RFID?

Осим носилаца информације, *RFID* систем захтева и средство којим ће те информације бити прочитане, затим пренесене рачунару, односно информационом систему. *RFID* уређај (читач) користи радио трансмисију за слање енергије транспондеру ¹(*RFID* таг) који онда емитује повратну информацију: јединствени серијски број и/или низ података, раније смештених у самом транспондеру. Поменуте податке могуће је даље обрађивати. [2]

Физички постоје три категорије:

- ✓ Транспондер (таг)
- ✓ „паметне“ налепнице
- ✓ *RFID* картица

Носилац информације у облику транспондера, налепнице или картице се поставља на објекат, амбалажу или чак на сам производ (или га корисник може носити са собом, као што је случај са нашим пројектом), тако да може путовати са њиме и на сваком кораку га идентификовати.

RFID транспондер – таг. Транспондери се производе у различитим облицима, величинама, с различитим капацитетима меморије и отпорношћу. *RFID* транспондер може бити довољно мали како би се сместио под кожу животиње, може бити као ексер или шраф за означавање дрвене грађе или у облику кредитне картице за коришћење у апликацијама контроле приступа.

Велики пластични привесци за спречавање крађе прикачени на гардеробу у трговинама такође су *RFID* транспондери, а слични су и врло отпорни транспондери у облику блока којима се означавају контејнери у интерним процесима производње, или радне машине и камиони у сврху праћења и одржавања. Готово сви су заштићени неком врстом кућишта од удараца, хемикалија, влаге и прашине. [3]

RFID налепница. Бар-код као технологија аутоматске идентификације у употреби је већ деценијама и врло је добро прихваћен. Ипак, једном одштампане, бар-код налепнице не могу више бити промењене, а да би је скенер прочитао мора бити у видљивом домету скенера. Нова генерација „паметних” (смарт) налепница опремљена је РФИД технологијом и нема ограничења као традиционални бар-код. Интегрирани електронички склоп садржи дигиталну меморију и може бити програмиран или ре-програмиран коришћењем радио-таласа. Смарт налепнице имају очигледну предност пред традиционалним бар-код налепницама у апликацијама где је потребна комбинација ефикасности читања и визуелна, људском оку разумљива информација. [2]

¹ **Транспондер.** Изведена од термина *transmitter/responder*, према функцији тог уређаја који на трансмисију читача одговара податком. Основне компоненте транспондера су микрочип и антена.

RFID плочица. РСВ плочица (Printed Circuit Board) је намењена уградњи у производ или амбалажу. Предности јој је нижа цена и способност подношења услова околине које *RFID* налепнице не би поднеле.



Слика 2: Примена *RFID* технологије

Читање, као и уписивање у транспондер могуће је извршити на следећа три начина а зависи од типа меморије [2]:

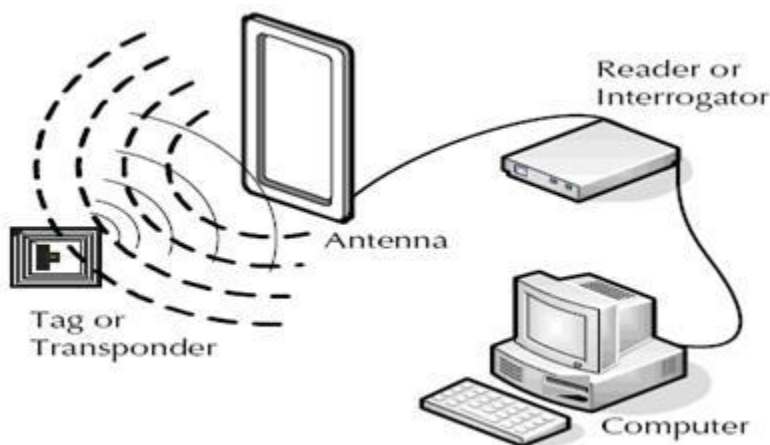
- ✓ **Read only (R)** - само читање транспондера који у процесу производње добија свој јединствени серијски број. Једном смештена информација не може се мењати (коришћено у овом раду).
- ✓ **Write Once Read Many (WORM)** - корисник сам програмира меморију транспондера, али податак може уписати само први пут, након чега он остаје перманентно похрањен.
- ✓ **Read/Write (R/W)** - корисник може много пута уписати информацију на транспондер. Рead-write транспондери обично имају серијски број који се не може избрисати, а подаци који се уписују, додају се томе. Рead-write транспондери су корисни у комплекснијим апликацијама, али будући скупљи, нису практични за означавање јефтиних производа.

Транспондери комуницирају с читачем путем радио-таласа. Радио-таласи су део електромагнетског спектра за који у свакој држави постоји законска регулатива. Проблем с *RFID* комуникацијом је у томе што су у различитим земљама света делови спектра различито расподељени према намени. Значи, транспондер који ради на 915 MHz у једном делу света, ће бити неупотребљив негде друго, где је то фреквенцијско подручје намењено каквој другој апликацији. [2]

Свака држава управља фреквенцијама у складу с регулативом трију постојећих подручја: Европа и Африка представљају Регион 1, Северна и Јужна Америка Регион 2, а Аустралија и Азија Регион 3. Постоји иницијатива за постизање одређеног степена слагања у погледу кориштења фреквенцијских подручја до 2010. године, али их је тренутно за примену *RFID* технологије врло мало доступно на глобалном нивоу. [3]

RFID систему се могу класификовати у три подручја [3]:

- ✓ **Подручје ниске фреквенције** (LF^2) – између 100 и 500 kHz, најкраћи домет сигнала и најмања брзина читавања и преноса.
- ✓ **Подручје високе фреквенције** (HF^3) – између 10 и 15 MHz, кратког до средњег домета сигнала, средње брзине читавања и преноса.
- ✓ **Подручје веома високе фреквенције** (UHF^4) – између 433 и 915 MHz, највећег домета сигнала, веће врзине преноса. Код ових сигнала не сме бити препреке између читача и транспондера, јер таласи на овим фреквенцијама не продиру тако брзо кроз материјале и захтева више енергије за трансмисију у датом опсегу него талас ниже фреквенције.



Слика 3: Функционалност *RFID* технологије

Дакле, да резимирамо како заправо *RFID* технологија функционише. Читач у континуитету шаље радио-сигнале (зависности од фекренције ти сингали могу имати домет од десетак центиметара до петнаест метара), што значи да читач мора бити повезан на извор напајања (у овом раду то је *Raspberry PI 3*). *RFID* таг (састоји се од транспондера и антене), који може бити величине привеска или кредитне картице, условно речено „прима“ сигнале од читача, те му је читач извор напајања, и преко антене враћа сигнал назад читачу (тај сигнал садржи податке из транспондера). Добијене податке читач обрађује у свом микроконтролеру. Битно је напоменути да је могуће писање на тагу.

² Low Frequency

³ High Frequency

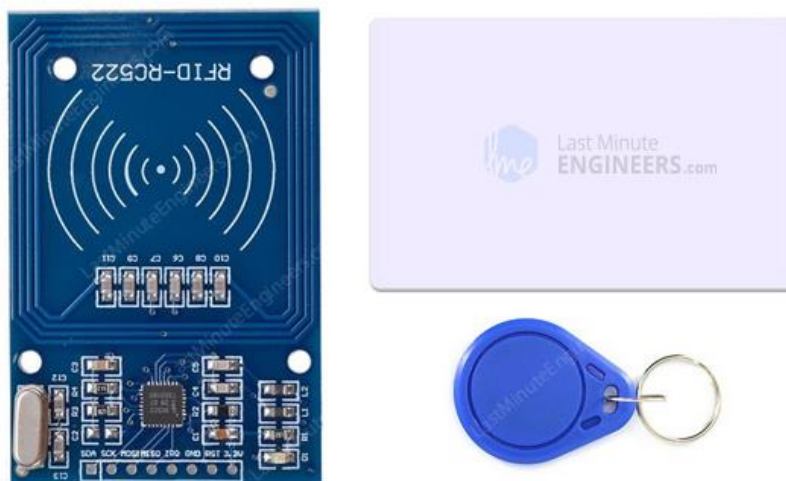
⁴ Ultra High Frequency

Меморија транспондера је реда величине неколико килобајта, јер веће меморије не би биле исплативе, узимајући у обзир да би онда били неопходни сигнали јачи од радио-сигнала. *RFID* технологија је свуда око нас, како у продавницама, тако и у нашим кућним љубимцима. Пракса је да се у домаће животиње уграђују *RFID* тагови, или што ветеринари воле да кажу, микрочипови. Тиме, сваки ветеринар, уз помоћ свог *RFID* читача може да прочита податке са, нпр. пса, ако се негде изгуби (ти подаци су име и презиме власника, број телефона, име пса итд.). Ова технологија се усталила и у јавном превозу у Србији (бус плус), као и у нашим кредитним картицама, што нам уштеди неколико секунди куцања ПИН кода, али може бити ризично ако се кредитна/дебитна картица изгуби, јер онда свако може подићи новац са исте.

У овом дипломском раду, читач са картице прочита серијски број, упоређује га са базом података и накнадно утврђује да ли особа са картицом има ауторизацију да отвори одређену просторију. Уколико да, просторија се отвара и у бази података се уписује име и презиме корисника, као и тачно време када је приступио одговарајућој просторији. Уколико не, врата се не отварају, након три неуспешна покушаја, кориснику се блокира картица, док јој админ не склони рестрикције кроз Интернет апликацију.

2.2 *RC522* сензор

RC522 RFID модул, базиран на *MFRC522 IC* технологији, је један од најјефтинијих *RFID* компоненти коју је могуће пронаћи, по цени мањој од четири америчка долара. У пакету, поред читача, добија се и једна картица и привезак (тагови), оба имају меморију од једног килобајта. Најбоље од свега, може се писати у тим таговима. [4]



Слика 4: *RC522 RFID* читач и тагови

Модул је дизајниран тако да генерише електромагнетно поље, на фреквенцији од 13.56 MHz, преко којег је остварена комуникација са таговима. Читач комуницира са микроконтролером преко четворопинског Серијског Периферног Интерфејса (*SPI*⁵), са максималним протоком података од 10Mbps. Такође, подржава комуникацију преко *I2C* и *UART* протокола. [4]

Модул, такође, садржи пин прекида (енг. *interrupt pin*), што је корисно јер уместо да се стално пита „да ли је прислоњена картица“, модул ће нас обавестити уколико дође до комуникације са тагом. [4]

Напон на којем ради сензор је у распону од 2.5 до 3.3 волта, међутим, логички пинови су толерантни на 5 волти, што значи да се, без проблема, може повезати на Ардуино или било који логички контролер (чије је напајање максимално 5V) без коришћења било каквих конвертера. У овом пројекту је повезан на *Raspberry PI*, који ради на 3.3V. На следећој слици се табеларно могу видети све спецификације сензора.

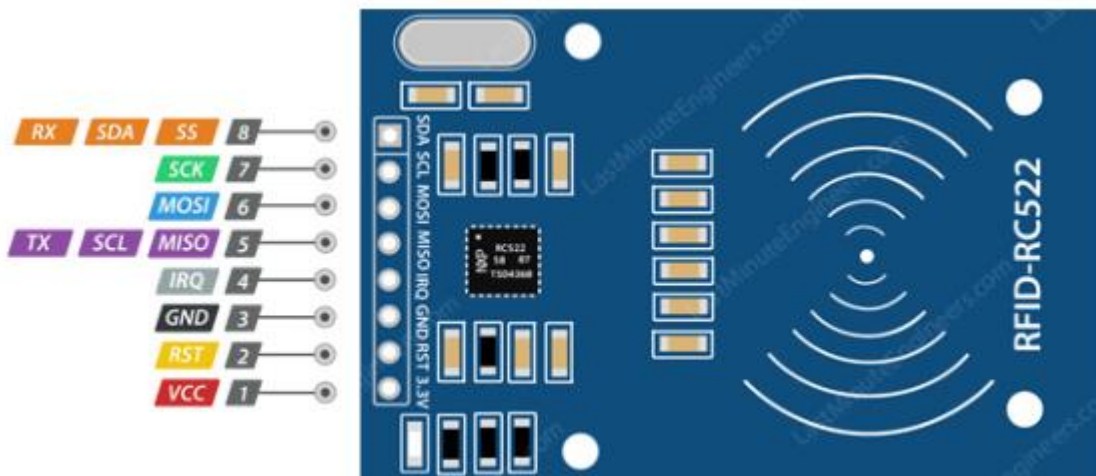
Frequency Range	13.56 MHz ISM Band
Host Interface	SPI / I2C / UART
Operating Supply Voltage	2.5 V to 3.3 V
Max. Operating Current	13-26mA
Min. Current(Power down)	10µA
Logic Inputs	5V Tolerant
Read Range	5 cm

Слика 5: Спецификације RC522 сензора

⁵ SPI- је синхрони протокол, који користе микроконтролери за комуникацију са периферним уређајима (сензорима). Такође, може се користити за комуникацију два микроконтролера. Функционише по принципу *master-slave*, где је један уређај *master* (углавном микроконтролер), док су остали „слуге“ (сензори). [5]

2.2.1 RC522 RFID пинови

RC522 RFID модул има укупно осам пинова, који га повезују са спољашњим светом. На следећој слици се исти могу видети.



Слика 6: Пинови RC522 сензора

- **VCC** – повезује сензор на извор напајања. Као што је раније поменуто, тај извор може бити између 2.5V и 3.3V.
- **RST** – улаз за *Reset* и *power-down*. Када напон на пину постане мали, сензор ради у режиму штедне енергије. Ово гаси сви интерну струју, укључујући и осцилатор, пинови више не комуницирају са спољашњим светом. На растућој ивици клока, модул (сензор) се ресетује.
- **GND** – је пин за уземљење, и мора бити повезан на *GND* пин микроконтролера. Напомена: напон уземљења је 0V.
- **IRQ** – је пин прекида, који је поменут раније у тексту, чија је улога да овабести микроконтролер када *RFID* таг „уђе“ у домет читача.
- **MISO/SCL/Tx** – пин се понаша као *Master-In-Slave-Out* када је омогућен SPI интерфејс, понаша се као серијски такт када је омогућен I2C интерфејс и као серијски излаз података када је омогућен UART интерфејс.
- **MOSI (Master Out Slave In)** – SPI улаз модула.
- **SCK (Серијски такт)** – прихвата импулсе такта, које шаље SPI магистрала „господара“ (у овом случају *Raspberry PI*).
- **SS/SDA/Rx** – пин се понаша као улаз сигнала када се омогућен SPI интерфејс, понаша се као серијски ток података када је I2C протокол омогућен, и понаша се као серијски улаз података када је UART интерфејс омогућен.

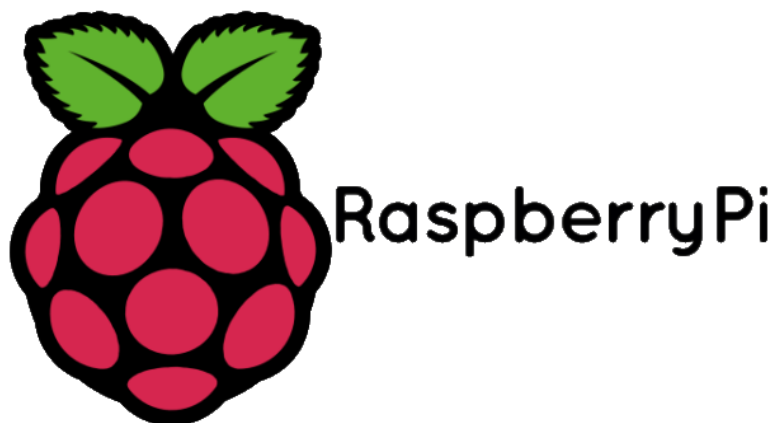
2.3 Raspberry PI 3

До сада сте могли прочитати неколико пута у овом тексту *Raspberry PI*, у овом одељку, коначно сазнајемо шта ја то.

Raspberry PI је нискобуџетни рачунар, величине кредитне картице, који се може повезати на монитор или ТВ, користи стандарну тастатуру и миш (докле год имају USB улазе). Способни, мали уређај који дозвољава људима свих узрасних доба да истражују рачунарство и да уче програмирање у најпопуларнијим језицима, као што је нпр. *python*. Може све што може и десктоп рачунар, од *сурфовања* Интернетом, гледања HD видео записа, играња игрица итд. [6]

Међутим, оно што га одваја од обичног персоналног рачунара (осим цене) јесте способност комуникације са спољашњим светом. У нашем случају, то је малопре поменути *RC522* модул. Оно што треба поменути јесте да *Raspberry PI* имају своју едукацијску фондацију, чији је циљ да побољша образовање деце и одраслих широм света, у области рачунарства. [6]

Оно што одваја *RPI*⁶ од својих претходника јесте што има картицу за бежични интернет, као и уграђене Wi-Fi и Bluetooth компоненте. [7] Мој провајдер Интернета и телефоније је SBB, скоро су ми донели тзв. *EON Smart Box*, који од обичног ТВ уређаја прави паметни ТВ уређај, који је на даљински управљач повезан Bluetooth-ом. Разлог због којег помињем ово јесте што се исти тај уређај може направити и на *RPI*-у. SBB-ов уређај је на Интернет повезан преко Етернет кабла, што подржава и *RPI*, такође има и Bluetooth, тако да је могуће повезати и даљински управљач на *RPI*. Намеће се закључак да су примене овог малог уређаја бескрајне.



Слика 7: Симбол *RaspberryPi-a*

⁶ RPI – скраћеница за *Raspberry PI*, често ће бити коришћена у овом раду.

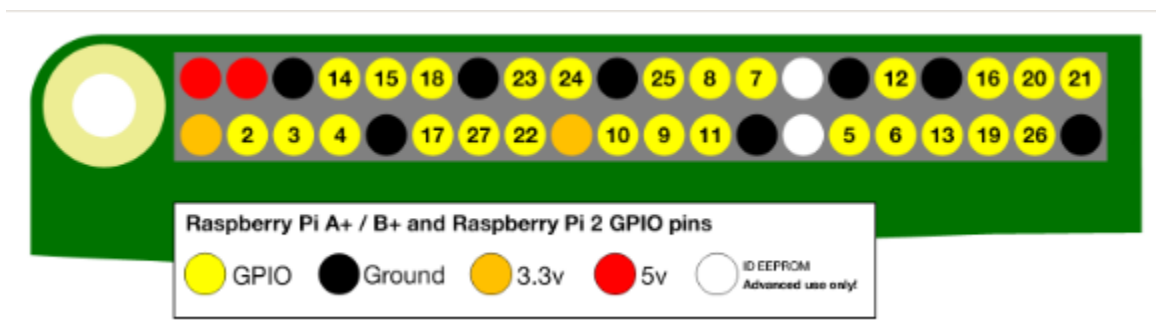
2.3.1 RPI 3 пинови

Моћни атрибут ове генерације RPI плоча је нови ред *GPIO*⁷ пинова. Глава од 40 пинова се налази на свим најновијим RPI рачунарима, док су старији модели имали по 26. На слици број 8 можете видети поменуте пинове.



Слика 8: RPI пинови

На следећој слици налази се схематска репрезентација пинова. Напомена: пинови нису у нумеричком редоследу, физички пинови 27 и 28 су за напредније коришћење. [7]



Слика 9: Схематска репрезентација RPI пинова

Пинови обележени црвеном бојом представљају извор напајања од 5V (превише за RC522 сензор, ризикујемо пржење микроконтролера), док су наранџасти пинови напајање од 3.3V. Остали пинови су сви пинови опште намене, што значи да су улази и излази толерантни на 3.3V. [7]

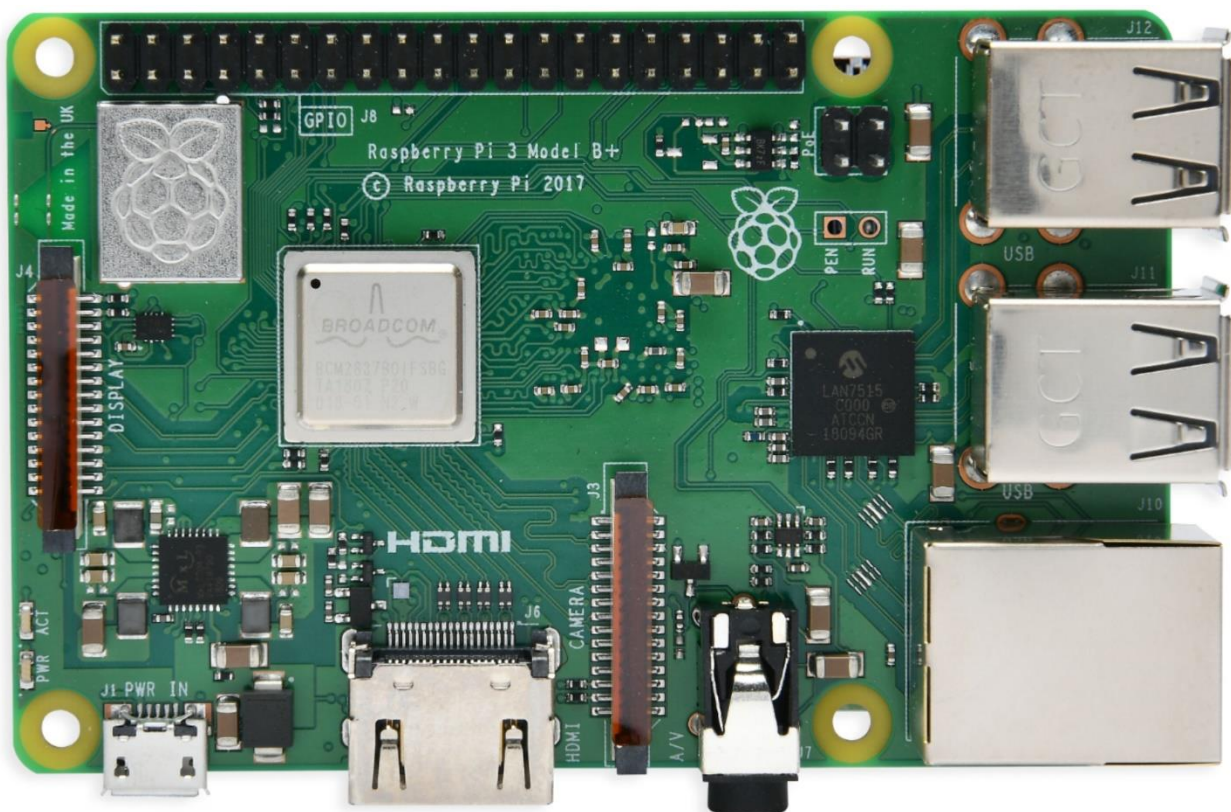
Пинови опште намене могу бити излазни као логичка јединица (3.3V) или као логичка 0 (0V). [7]

Такође, улазни пинови могу бити прочитани као логичка јединица (3.3V) или као логичка 0 (0V). Процес је олакшан коришћењем интерних *pull-up* или *pull-down* отпорника. Пинови 2 и 3 имају фиксне *pull-up* отпорнике, док се остали могу конфигурирати софтверски.

⁷ General-Purpose Input/Output

Пинови поред опште намене, могу имати и алтернативне функције, неке су доступне на свим пиновима, а неке на специфичним. [7]

- *PWM (pulse-width modulation)*
 - Софтверски *PWM* доступан на свим пиновима
 - Хардверски доступан на пиновима 12, 13, 18 и 19
- *SPI*
 - *SPI0*: *MOSI* (пин 10), *MISO* (пин 9), *SCLK* (пин 11), *CE0* (пин 8), *CE1* (пин 7)
 - *SPI1*: *MOSI* (пин 20), *MISO* (пин 19), *SCLK* (пин 21), *CE0* (пин 18), *CE1* (пин 17), *CE2* (пин 16)
- *I2C*
 - Подаци (пин 2), такт (пин 3)
 - *EEPROM* подаци (пин 0), *EEPROM* такт (пин 1)
- *Серијска комуникација*
 - *TX* (пин 14), *RX* (пин 15)



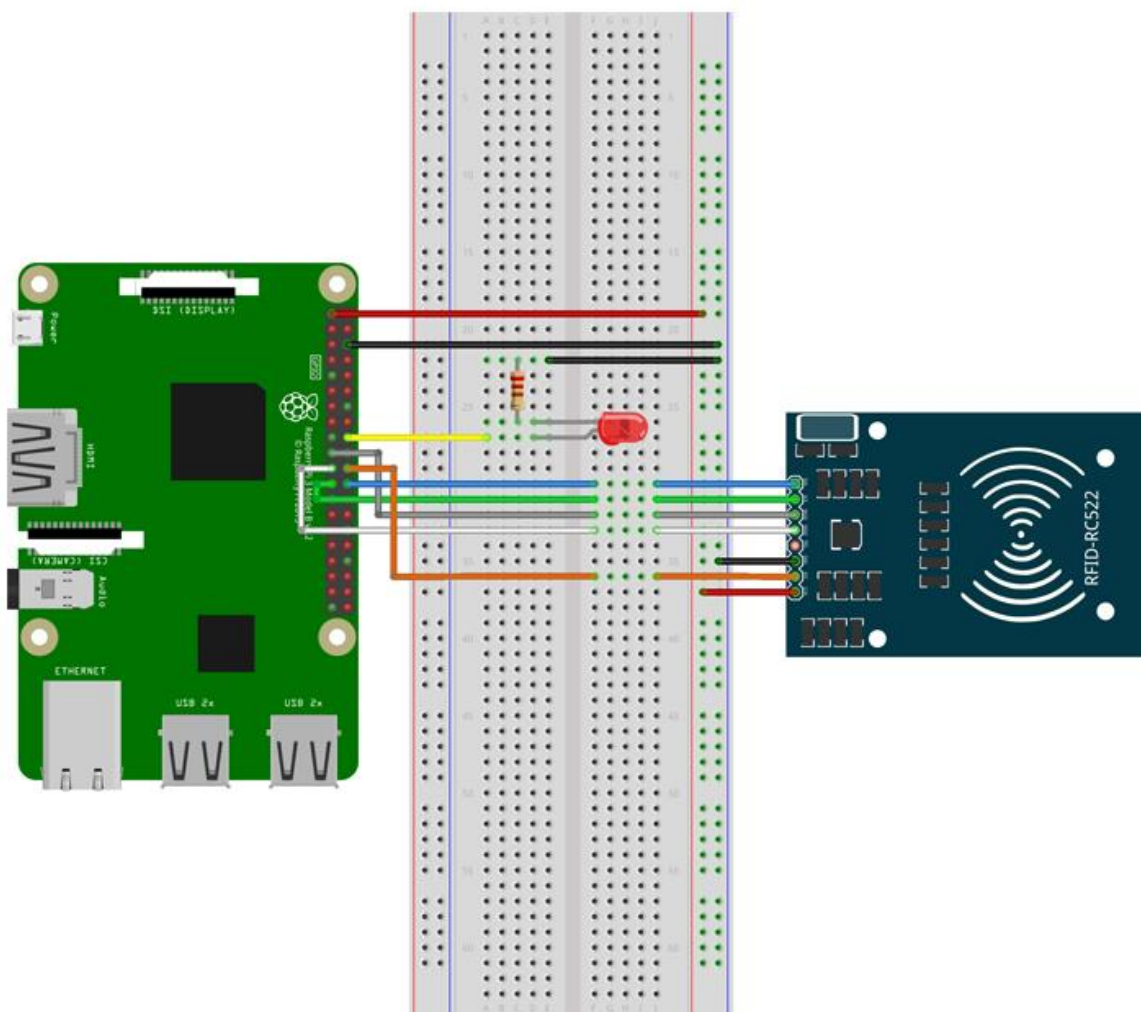
Слика 10: Изглед *RPI3* рачунара

2.3.2 Повезивање RC522 сензора на RPI3

За повезивање RC522 модула на RPI3, користиће се SPI протокол за комуникацију између два уређаја, неопходно је инсталирати библиотеку за SPI комуникацију, као и библиотеку за модул, *MSRF522*. [9, 10]

Поред два поменута уређаја, повезиваће се и диоде (зелена и црвена) које представљају комуникацију са корисником, добијају сигнале од пинова опште намене. [11]

Повезивање два уређаја врши се преко прото плоче и мушко-женских џампера. На слици број 11 може се видети схематски приказ, као смерница за повезивање, док је на слици број 12 приказана табела RPI пин на RC522 пин. [8]



Слика 11: Схематски приказ повезивања RPI и RC522

RC522 Pin	Raspberry Pi Pin	Wire Color
3.3V	Pin 1	Red
RST	Pin 22	Orange
GND	Pin 6	Black
MISO	Pin 21	White
MOSI	Pin 19	Grey
SCK	Pin 23	Green
SDA	Pin 24	Blue

Слика 12: Табела повезивања (RC522 пин на RPI пин)

2.4 WAMP сервер

WAMP је пакет софтвера који се инсталира на *Windows* оперативном систему. Користи се за програмирање Интернет апликација и интерно тестирање, такође, може серверати и праве интернет странице (живе странице). [12] WAMP је изведена скраћеница од следећих речи:

1. **W** – означава *Windows* оперативни систем, такође постоје *LAMP* (за Линукс) и *MAMP* (за Мек).
2. **A** – означава *Apache*. *Apache* је HTTP сервер софтвер који је одговоран за „сервирање“ интернет страница.
3. **M** – означава *MySQL*, чији је задатак да буде систем управљања база података за сервер, у којима су сачуване све релевантне информације.
4. **P** – означава *PHP*. То је скриптни програмски језик, који представља, условно речено, лепак за скуп наведених софтвера. *PHP* ради у спони са *Apache* интернет сервером и комуницира са *MySQL*-ом. Напомена: **P** у WAMP-у може означавати и *Perl* или *Python*, програмске језике, који су такође скриптни.



У наставку рада биће још речи о појединачним компонентама WAMP-а.

2.4.1 *Apache HTTP* сервер

Apache је *open-source* и бесплатан Интернет сервер, на којем је подигнуто око 46% Интернет страница широм света. Званични назив је *Apache HTTP Server*, одржава га *Apache* софтвер фондација. [13]

Дозвољава власницима да сервисирају садржај на Интернету – отуд назив Интернет сервер. Један је од најстаријих и најпоузданијих сервера, где је прва верзија обелодањена пре 20 година, 1995. [13]

Иако *Apache* називамо сервером, он није физички сервер, већ софтвер који ради на неком серверу. Примарни задатак овог софтвера је да успостави везу између неког сервера и претраживача (*Firefox*, *Google Chrome*, *Safari*, итд.) док доставља податке између истих. *Apache* је *cross-platform* софтвер, што значи да ради и на *Windows* и *UNIX* оперативним системима (серверима).

Када посетилац хоће да учира страну са наше Интернет странице, њихов Интернет претраживач шаље захтев нашем серверу и *Apache* враћа одговор са свим захтеваним подацима (текст, слика, итд.). Сервер и клијент комуницирају преко **HTTP** протокола, *Apache* је задужен за глатку и сигурну комуникацију између две машине. [13]

WAMP даје могућност „мењања“ и прилагођавања *Apache* софтвера, преко фајла који се зове *httpd*, што поразумева да Интернет страницу можемо ограничити, у смислу да се њен приказ могућ само на одређеним IP адресама, или чак само једној итд. [13]

2.4.2 *MySQL*

MySQL је најпопуларнији *open-source SQL* систем управљања базама података, који развија и одржава *Oracle* коорпорација. [14]

База података је структурирана колекција података. Може бити нешто једноставно попут најобичнијег списка намирница за куповину, до нечега комплексног попут галерије слика, са разноврсним подацима у великој мрежи. Како бисмо добили приступ и процесуирали податке, наведени подаци морају бити у бази података, а да бисмо тој бази приступили неопходан нам је систем управљања, какав нуди *MySQL* сервер. [14]

MySQL је релациона база података, што значи да податке одваја по табелама, уместо да их чува све на једном месту. Структура базе података је организована као физичка датотека, ради оптимизације. Логички модели, са објектима као што су базе података, табеле, погледи, редови и колоне, даје додатну флексибилност за програмерско окружење. Релационе базе имају „правила“ по којим знамо везу између оређених табела, или пак колоне из тих табела, што могу бити један према један, један према више, више према више итд. Такве базе, са наведеним правилима, имају такав ефекат да апликација не делује неконстантно, да нема *memory leak-ова*, недовољно простора итд.

SQL део *MySQL* је скраћеница од *Structured Query Language*, који је најстандарднији језик за приступ базама података, веома је интуитиван и лак за учење. [14]

2.4.3 PHP

PHP је, такође, *open-source*, скриптни језик, веома популаран и коришћен широм света. Језик је опште намене, међутим, највећу примену проналази у Интернет апликације, због добре спреге са *HTML*-ом.

Оно што одваја *PHP* од других језика јесте што се код извршава на серверу, где генерише *HTML*, који се шаље клијенту. Клијент добија само резултат покретања те скрипте, али не зна који је код иза тог резултата. Корисник не зна какве трикове кријемо у рукаву.

2.5 *Laravel Интернет framework*

Laravel је *framework*⁸ за интернет апликације са експресном, елегантном синтаксом. Ларавел олакшава развој апликације тиме што аутоматски имплементира честе таскове, који се користе у већини апликација, попут аутентикације корисника, рутирања, сесија и кеширања података. [17]

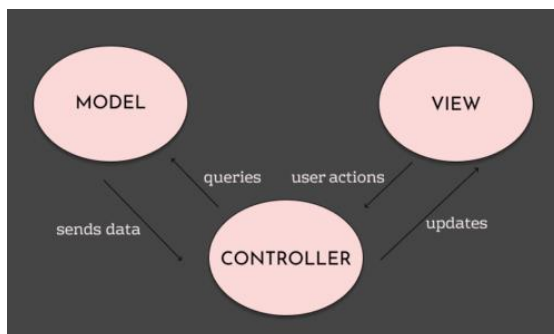
Ларавел за циљ има да развој апликација буде једноставнији за програмера, без жртвовања функционалности апликације. Срећни програмери праве најлепше кодове. [17] Ларавел је комбинације других *framework-ова*, из других програмских језика, те се могу видети утицаји *Ruby-a (Rails framework)*, *ASP.NET MVC* и *Sinatra*.

Ларавел је приступачан, те моћан, садржи алате неопходне за велике апликације. Одлична инверзија контролних контејнера, експресиван систем миграција и феноменално *UNIT* тестирање.

2.5.1 *MVC*

MVC је акроним за *Model View Controller*, који је архитектура коју су програмери усвојили како би правили апликације. Поменута архитектура одваја домен/апликацију/бизнис итд. логику од корисничког интерфејса, што постиже одвајањем апликације на три дела: [17]

1. *Model* – управља фундаменталним понашањима и подацима апликације. Може да реагује на захтева за информацију, инструкцију промена информације и да чак информише посматрача о промени информације. Модел може бити база података, или било која структура података или систем складиштења. Укратко, модел је „податак“ и ситем управљања података у датом систему.
2. *View* – кориснички интерфејс у апликацији. „Рендероваће“ податке из модела у нешто што корисник може да разуме.
3. *Controller* – спона корисника и целог система. Узима улазе од корисника и „позива“ објекте модела, као и *view* како би исти извео одговарајуће акције.



Слика 13: *MVC*

⁸ *Framework* – обезбеђује код који „гради“ комплетну апликацију која зна **шта** да уради, али не зна **како**, програмер има задатак да „објасни“ апликацији како.

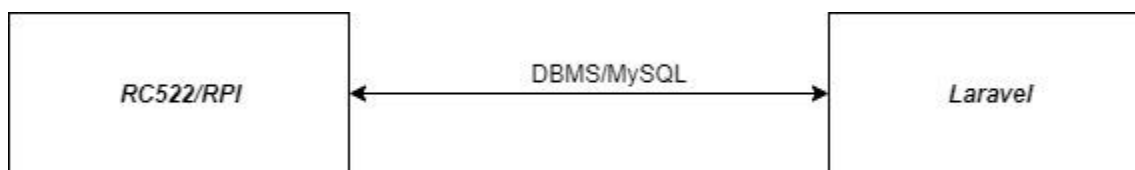
3 Архитектура и имплементација апликације

У овом поглављу описана је архитектура решења, заједно са детаљном комуникацијом између елемената система. Такође, описана је детаљна структура целог система. Циљ је читаоца припремити за детаљно разумевање система, као и додавање нових теоријских основа.

На следећој слици (блок-шема) приказана је основна архитектура целог система и основно функционисање истог. Систем се састоји од два подсистема:

- *RPI*
- *WAMP/Laravel*

Модул *RC522* повезан је на *RPI*, као што је раније речено, преко *SPI* протокола шаље очитане информације са *RFID* тага, на *RPI*, који дате информације обрађује у *python* програмском језику. Обрађивање тих информација подразумева очитавање серијског броја картице, затим проверавање из базе података да ли та картица постоји, ако постоји, коме припада и којим просторијама има приступ. Уколико се тренутна просторија налази међу просторијама са дозвољеним приступом, диода светли зелено. У бази података се бележи лог са јединственим идентификационим бројем корисника, просторије, као и тачног времена када је поменути корисник ушао у просторију. Уколико тренутна картица не постоји у бази, или уколико дата картица нема приступ поменутој просторији, диода светли црвено. Кориснику се пише, условно речено, „минус“. Уколико три пута покуша да уђе, без дозволе, картица се блокира. Намеће се закључак да је веза између два подсистема заправо њихова заједничка база података, као и *Apache* Интернет сервер.



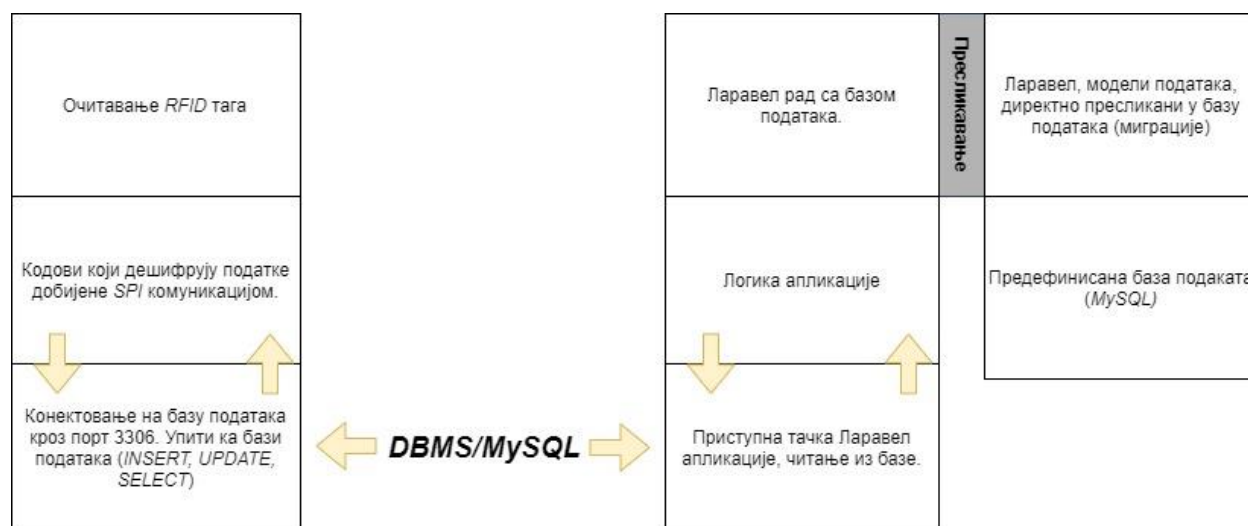
Слика 14: Блок схема система

Слика је апстрактна блок-схема функционалних делова система, са основним приказом функционисања система. На слици бр. 14 приказана је и комуникација између подсистема. Технички они комуницирају кроз порт 3306, заједничком манипулацијом базе података. Опширнје објашњење налази се у другом делу рада, поглавље *Комуникације два подсистема*.

3.1 Комуникација два подсистема

Као што је раније најављено, у овом поглављу, читалац ће бити упознат, детаљно, како два система функционишу. Дакле, Ларавел систем представља виртуелни сервер, то је локална машина која ће сервисирати Интернет апликацију. *Apache* софтвер обезбеђује комуникацију два подсистема, заједно са базом података која се налази на локалној машини, и која ради на *Windows* оперативном ситему (отуда *WAMP*). Да би комуникација била дозвољена, неопходно је „отворити“ порт 3306 (на локалној машини), што свако може учинити у свом *Windows Firewall-u*, тако што дода порт 3306 у „дозвољене апликације“. Зашто баш тај порт? Уобичајно је да *WAMP* тај порт користи за приступ бази података, могуће је мануелно променити тај порт у *my.ini* фајлу. Такође, неопходно је да *WAMP* ради у *online* моду (*WAMP* наглашава да је ова опција само за експерте). Из сигурносних разлога, у *MySQL*-у додат је нови корисник, преко којег се може улоговати само преко *RPI*. Специфицирана је IP адреса *RPI-a*, и уписани корисничко име и лозинка. Као тај корисник могуће је улоговати се само преко *RPI-a*, и само тај корисник има приступ бази са локалне машине (сигурносни разлози).

Напомена: Ради бољег увида, омогућено је, преко *WAMP* поставки, прегледати Интернет и *phpmyadmin* страницу. Неопходно је у *Windows Firewall-u* *httpd* фајл ставити у „дозвољене апликације“, као и у поменутом фајлу *localhost* отворити за све машине са локалне мреже (команда *Require all granted*).

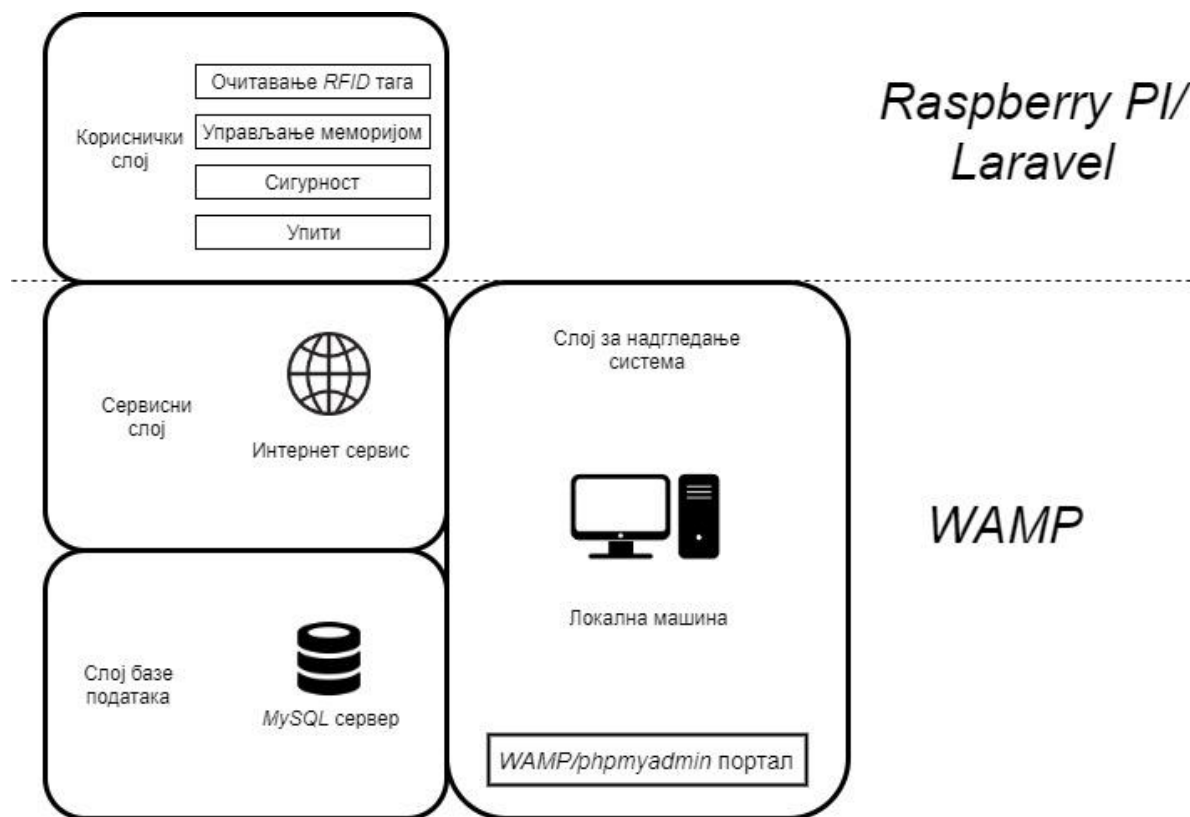


Слика 15: Комуникација система

Тешко дизање у овој апликацији је на *RPI*-у, док Ларавел као задатак има да податке скупљене са хардвера приказује, тако да сваки корисник разуме.

3.2 Структура система

Структура система се може представити кроз апстрактну поделу система на слојеве. Основна подела сваког система може бити на слој који корисници виде и користе и на слој који обаља послове које корисник задаје. Могуће је даље ширити ту поделу, на слој базе података и сервер базе података.



Слика 16: Структура система

Слојеви система, приказани на слици бр. 16 су:

- **Кориснички слој:** *RPI/Laravel* део система је оно што корисник види. Подразумева очитавање *RFID* тага, као и Интернет апликацију коју корисник види. Један део је ауторизација корисника, тј. контрола приступа, док је други манипулација подацима. Кроз једноставан кориснички интерфејс даје контролисано управљање података кориснику. Повезује корисника са слојем испод: омогућава писање упита, дешифровање комуникације, приказ одговора добијених од сервисног слоја.
- **Сервисни слој:** Сервисни слој применом Интернет сервиса (*Apache*) обезбеђује комуникацију апликације са слојем испод (слојем базе података). Имплементиран је кроз *WAMP* (*Apache*), обрађује захтеве које добије од слоја изнад. Поменути захтеви су углавном *REST* типа (*GET*, *POST*, *PUT* или *DELETE*). Након обраде

захтева враћа одговарајући одговор. Сума сумарум, овај слој је као лепак који спаја оно што корисник види и базу података.

- **Слој базе података:** Налази се на самом дну система. Обезбеђује манипулацију података или кроз терминал или апликацију или путем локалне машине. *WAMP* интегрише слој базе података са слојем изнад (сервисни слој). У *WAMP*-у, као што је раније поменуто, база је имплементирана кроз *MySQL* софтвер. *RPI* комуницира са овим слојем кроз порт 3306, детљаније у одељку *Комуникација два подсистема*.
- **Слој за надгледање система:** Налази се паралелно уз цео систем, обезбеђује проверу рада система, као и контролу података, тј. измену истих. *WAMP* интегрише сва три ова слоја, приступ овом слоју се може добити преко *phpmyadmina* на локалној машини.

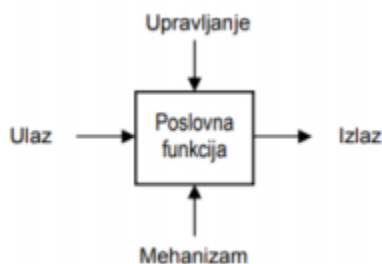
4 Функционално моделирање

У оквиру функционалног моделирања, спроводи се анализа свих елемената потребних за развој базе података и омогућава декомпоновање пословних функција и планирање потребних ресурса за реализацију функција. Функционално моделирање је везано за технике моделирања активности базираних на комбинацији графике и текста, који су представљени на организован и систематичан начин да би се повећала разумљивост, која подржава анализу, обезбеђује логику за потенцијалне измене, специфицира захтеве и подржава анализу система по нивоима и интегрише активности. [18]

Модел се састоји од хијерархијског низа дијаграма који постепено приказују све више детаља о функцијама и њиховој међувези (interface) са осталим деловима система и омогућава анализу особина одређеног пословног процеса ради његовог максималног унапређења.

4.1 Дијаграм контекста

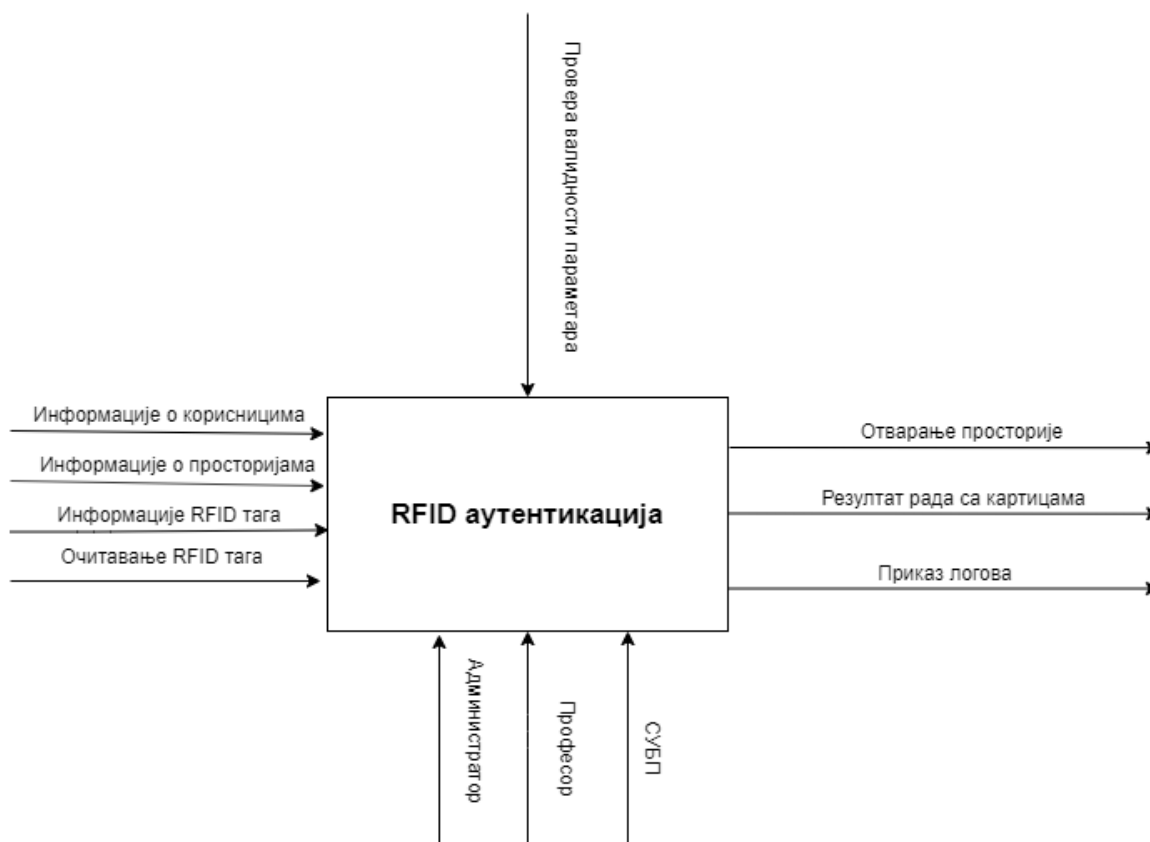
Први корак у оквиру функционалног моделирања је израда дијаграма контекста. Дијаграм контекста је кључан за дефинисање граница система. Дијаграм је дефинисан једним павоугаоником који представља границу модела који се проучава. У систему и ван њега, информације теку преко стрелица. Контекстни дијаграм је навиши ниво апстракције који се касније преводи у нижи ниво апстракције. Изглед дијаграма контекста приказан је на следећој слици. [18]



Слика 17: Општи облик дијаграма контекста

- Улаз (*Input*) представља нешто што се троши у процесу
- Управљање (*Control*) представља ограничење на извршавање процеса
- Излаз (*Output*) представља резултат процеса
- Механизам (*Mechanism*) представља оно што се користи у процесу али се не троши

Дијаграм контекста за апликацију описну овим радом приказан је на слици бр. 18.



Слика 18: Дијаграм контекста *RFID* аутентикације

4.2 Стабло активности

Након креирања дијаграма контекста, следи дефинисање стабла активности. На основу дефинисаних граница система, помоћу стабла активности се успостављају вертикалне (хијерархијске) везе између активности. Сложена активност се раставља на више подређених активности. [18] На тај начин, стабло активности представља хијерархију дефинисаних активности и омогућава функционалну декомпозицију разматраног проблема. Стабло активности се дефинише применом методе решавања проблема одозго на доле (top-down). Сложена активност се раставља на више подређених активности, а затим се приступа решавању једноставних, подређених, активности. Полазна сложена активност се развија у структуру типа стабла. Корен стабла (то је највиши чвор стабла) садржи полазну активност, док листови, тј. чворови који немају потомке, садрже активности чије је решавање релативно једноставно. Решавањем свих активности из листова решена је и полазна сложена активност. Разбијање активности „родитељ“ на своју децу треба да има минимум две до максималног броја шест подређених активности.

Уколико је број подређених активности већи од дозвољеног броја, значи да је покушано да се смести превише детаља за један ниво. Стабло активности апликације описане овим радом налази се на следећој слици. [19]



Слика 19: Стабло активности

5 Информационо моделирање

Информационо моделирање изводи се на основу дефинисаних критичних функција за претходно урађену активност функционално моделирање. Критичне функције представљају, свака за себе, по један главни пројекат који се развија у оквиру активности информационо моделирање. [18] Информациони модел приказује у каквом су међусобном односу подаци у неком реалном систему. Информационо моделирање омогућава:

1. Дефинисање системске документације која се може користити за базу података и за развој апликација да би корисници могли да дефинишу системске захтеве
2. Бољу комуникацију међусобно и са крајњим корисницима
3. Јасну слику о пословним правилима
4. Логичку слику базе података коју могу користити аутоматски алати за генерисање система за управљање базама података

5.1 Дефинисање детаљних захтева

Дефинисање детаљних захтева се изводи на основу претходно урађене активности функционално моделирање, тј. дефинисане студије којом се одређују оквири који се у овој активности, за изабране подсистеме, детаљно разграђују. [19]

Дефинисање детаљних захтева треба да представља ревизију постављених елемената из студије уз даље детаљно продубљивање за изабрану функцију за коју се спроводи реинжењеринг пословних процеса.

5.2 Креирање ЕР дијаграма

Креирање ЕР дијаграма представља моделирање података тј. дефинисање ентитетног дијаграма. За дијаграм треба користити технику за описивање структуре података и пословних правила под називом модел података, којим се дефинишу ентитети. [18] Сваки ентитет има своје особине (атрибуте) а све је то повезано везама. Према устаљеним конвенцијама, великим словима у једнини пишу се ентитети док се атрибути и везе пишу малим словима. [18]

5.2.1 Идентификација ентитета и веза

За активност идентификација ентитета полази се од објеката посматрања. Објекат посматрања је све што се може једнозначно идентификовати, па самим тим и изоловати из околине и описати. Тако је објекат посматрања и сам ентитет. Ентитет је особа, ствар, догађај, појам (реални или апстрактни) који је од трајног интереса тј. нешто што се жели појединачно посматрати. [17] Конкретно за ову апликацију, ентитети су:

- Корисник
- Улога
- Картица
- Просторија
- Аутентикација просторије
- Логови

Оваквим одабиром ентитета обезбеђено је да редуванса података приликом складиштења буде минимална.

Идентификација веза дефинише везе између ентитета. Веза представља интеракцију међу објектима тј. знање о њиховој међусобној повезаности. Веза се дефинишу зависности између ентитета односно описују начини повезивања (узајамна дејства) ентитета. Везе у реченици су глаголске фразе које показују какав је однос међу ентитетима. Премда, глаголске фразе не описују правила прецизно, оне дозвољавају особи која посматра модел да стекне основни осећај о повезаности ентитета. Добра је пракса ако читање модела даје

ваљане реченице (исказе). [17] У оквиру ове апликације, везе које се користе за ентитете су:

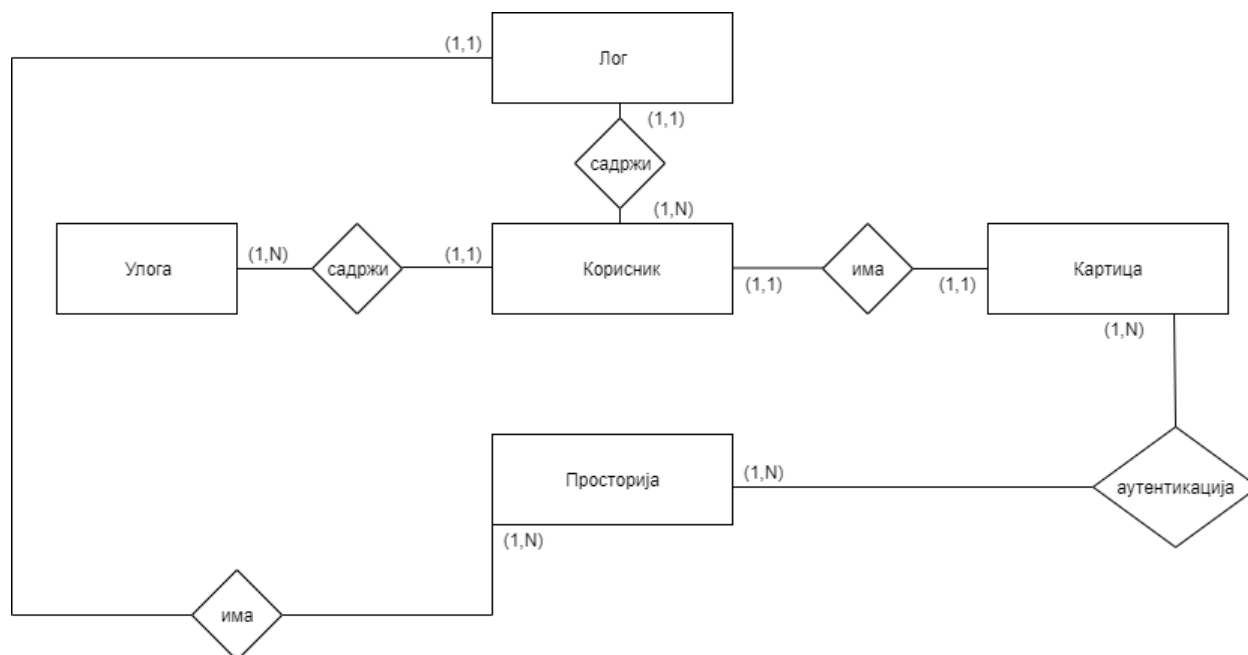
- Корисник има улогу
- Корисник има картицу
- Картица аутентификује просторију
- Лог садржи корисника
- Лог садржи просторију

5.2.2 Креирање атрибута

Сваки објекат реалног света је дефинисан особинама па по тој аналогiji и ентитети имају атрибуте којима се описују карактеристична својства. Након дефинисања атрибута, одређују се примарни кључеви који једнозначно одређују сваки ентитет. Атрибути или групе атрибута који могу бити изабрани као примарни кључеви називају се атрибути кандидати. Кандидат за кључ мора јединствено да идентификује сваки примерак ентитета. Из тога следи да ниједан део примарног кључа не сме бити Null односно празан или недостајући. Од кандидата за кључеве бира се један који се назива примарни кључ. Избор кључа неког ентитета може бити један атрибут којим се дефинише једноставни примарни кључ или комбинација атрибута којом се дефинише сложени примарни кључ. Примарни кључеви се додељују аутоматски и раде по принципу аутоматског инкрементовања, што значи да корисницима није омогућена ручна додела вредности за примарне кључеве. [17]

- Ентитет Корисник има следеће атрибуте:
Корисник (ID, име, презиме, ID_улоге, *RFID*)
- Ентитет Улога има следеће атрибуте:
Улога (ID, име)
- Ентитет Картица има следеће атрибуте:
Картица (ID, блокирана)
- Ентитет Просторија има следеће атрибуте:
Просторија (ID, име)
- Ентитет Аутентикација просторија има следеће атрибуте:
Аутентикација (*RFID*, ID_просторија)
- Ентитет Логови има следеће атрибуте:
Логови (ID, ID_просторија, ID_корисник, време)

На следећој слици приказан је ЕР модел базе података.



Слика 20: ЕР дијаграм

6 Дефинисање физичког слоја базе података

Циљ ове фазе је превођење логичког у физички дизајн, односно превођење ентитета у табеле, атрибута у колоне, као и одговарајућа ограничења. Најпре је потребно извршити избор СУБП (Систем за Управљање Базом Података). [18] База података ће бити креирана у оквиру MySQL сервера. Затим је потребно дефинисати табеле и колоне.

Релациони модел базе података тј. сама база података је скуп информација које се односе на постављену тему, односно систем. Тему, односно систем, чине одређени ентитети базе података. Подаци у оквиру базе су уређени и груписани по ентитетима. Сваки ентитет са собом носи одређене врсте податка који га карактеришу, а то су атрибути ентитета. Складиштење (чување) података врши се у табелама. У релационим базама података, то је више табела које носе податке о ентитетима. [18]

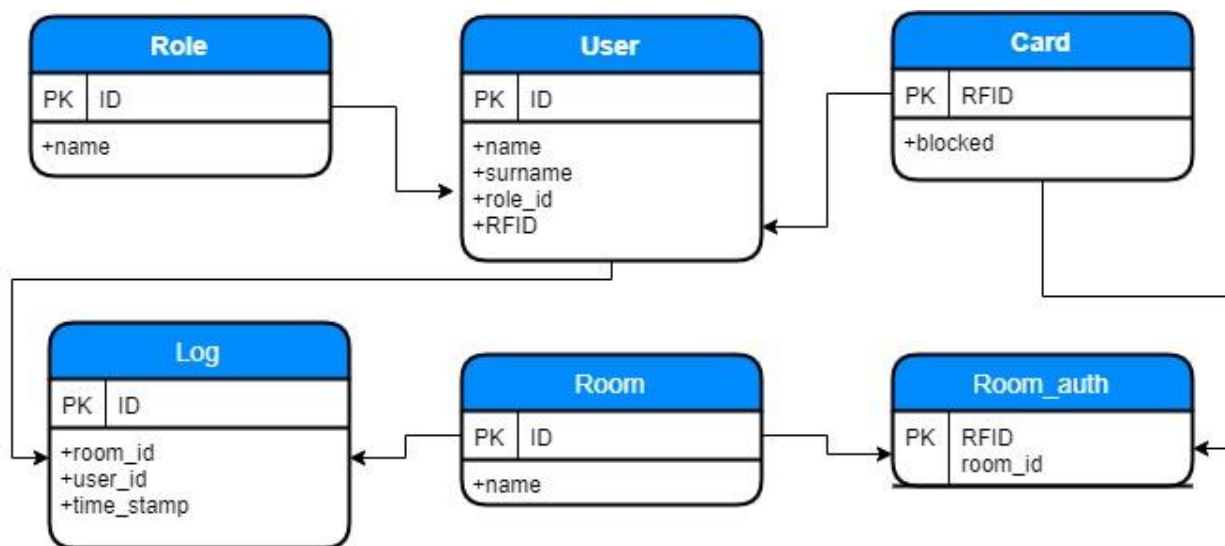
Табеле су међусобно повезане и функционишу као целина. Релациони модел базе података омогућава максималну флексибилност и економичност у чувању и коришћењу података. Информације о ентитетима се смештају у табеле. За сваки ентитет формирана је засебна табела. Ентитет чини мноштво субјеката, а атрибутима се одређују карактеристични подаци за све субјекте ентитета. Атрибут има своје име по којем се разликује од осталих у истој релацији. Вредност једног атрибута су подаци истог типа.

Дакле дефинисан је скуп дозвољених вредности за атрибут који се зове домен атрибута. [17]

Вредност атрибута је једнострука и једноставна (не може се раставити на делове). Неки атрибути захтевају своје атрибуте, онда их треба сматрати новим ентитетом. У табели сваки субјекат је једнозначно одређен, што подразумева постојање врсте података, односно атрибута који то омогућава. Поље јединствених вредности дефинише се као поље примарног кључа. На основу те врсте податка, односно преко поља примарног кључа врши се повезивање табела.

Пројектовано решење у основи је релациона база података која се налази на локалној машини којој корисници приступају ради уноса и обраде података и користе јединствене шифарнике за идентификацију чиме се омогућава конзистентност базе података. Комплетно решење је реализовано тако да омогућава лак рад у мрежи која ради по принципу клијент - сервер.

6.1 Схема базе података



Слика 21: Схема базе података

User (id, name, surname, role_id, RFID)

Role (id, name)

Card (RFID, blocked)

Room (ID, name)

Room_auth (RFID, room_id)

Log (ID, room_id, user_id, time_stamp)

Међурелациона ограничења:

User (role_id) $\subseteq$ **Role** (id)

User (RFID) $\subseteq$ **Card** (RFID)

Room_auth (RFID) $\subseteq$ **Card** (RFID)

Room_auth (room_id) $\subseteq$ **Room** (id)

Log (room_id) $\subseteq$ **Room** (id)

Log (user_id) $\subseteq$ **User** (id)

7 UML дијаграми

Unified Modeling Language (UML) је стандардни графички језик за моделовање објектнооријентисаног софтвера. *Unified* нам говори да он уједињује све постојеће нотације, *Modeling* зато што се користи за моделовање софтверских елемената док *Language*, као што и сама реч каже, представља средство комуникације. [20] Основни елементи UML-a су:

- Елементи (битни концепти - класе, објекти, поруке)
- Релације (повезивање индивидуалних ствари)
- Дијаграми (груписање међусобно повезаних колекција ствари и релација)

Оно што је битно напоменути је да се UML дијаграми деле на две целине, и то на: [20]

- Структурне дијаграме (*Structure diagram*) унутар којих се налазе дијаграми класа (*class diagram*), објекта (*object diagram*), компоненти (*component diagram*), постављања (*deployment diagram*), пакета (*package diagram*), профила (*profile diagram*), композитне структуре (*composite structure diagram*)
- Дијаграме понашања (*Behaviour diagram*) који обухватају дијаграм случајева коришћења (*use case diagram*), активности (*activity diagram*), машине стања (*state machine diagram*), као и дијаграме интеракције (*interaction diagram*) унутар којих се налазе дијаграми комуникација (*communication diagram*), редоследа/секвенци (*sequence diagram*) и времена (*timing diagram*)

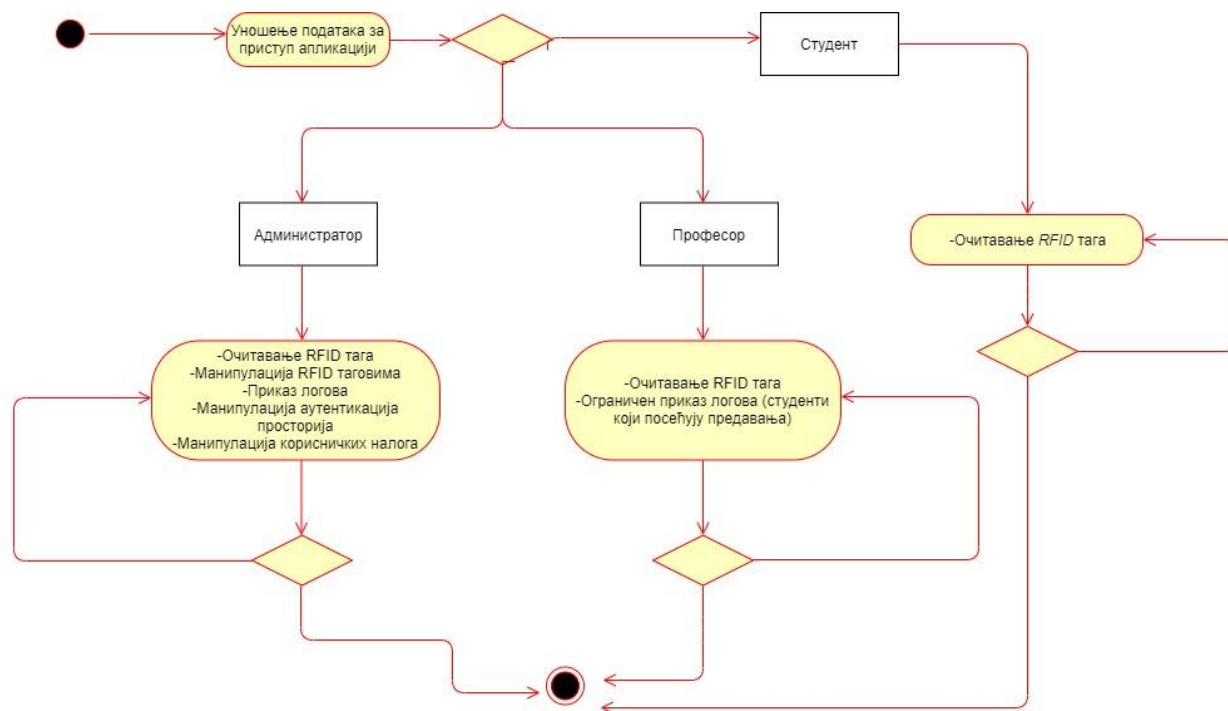
У оквиру овог рада биће представљена четири UML дијаграма и то они који се најчешће користе, а то су:

1. Дијаграм активности (*Behaviour diagram -> Activity diagram*)
2. Дијаграм стања машине (*Behaviour diagram -> State Machine diagram*)
3. Дијаграм случајева коришћења (*Behaviour diagram -> Use Case diagram*)
4. Дијаграм секвенци (*Behaviour diagram -> Sequence diagram*)

7.1 Дијаграм активности

Дијаграми активности су намењени моделирању динамичких аспеката (понашања) система. Овај дијаграм приказује ток активности коју извршавају објекти као и евентуално ток објекта између корака активности. [21]

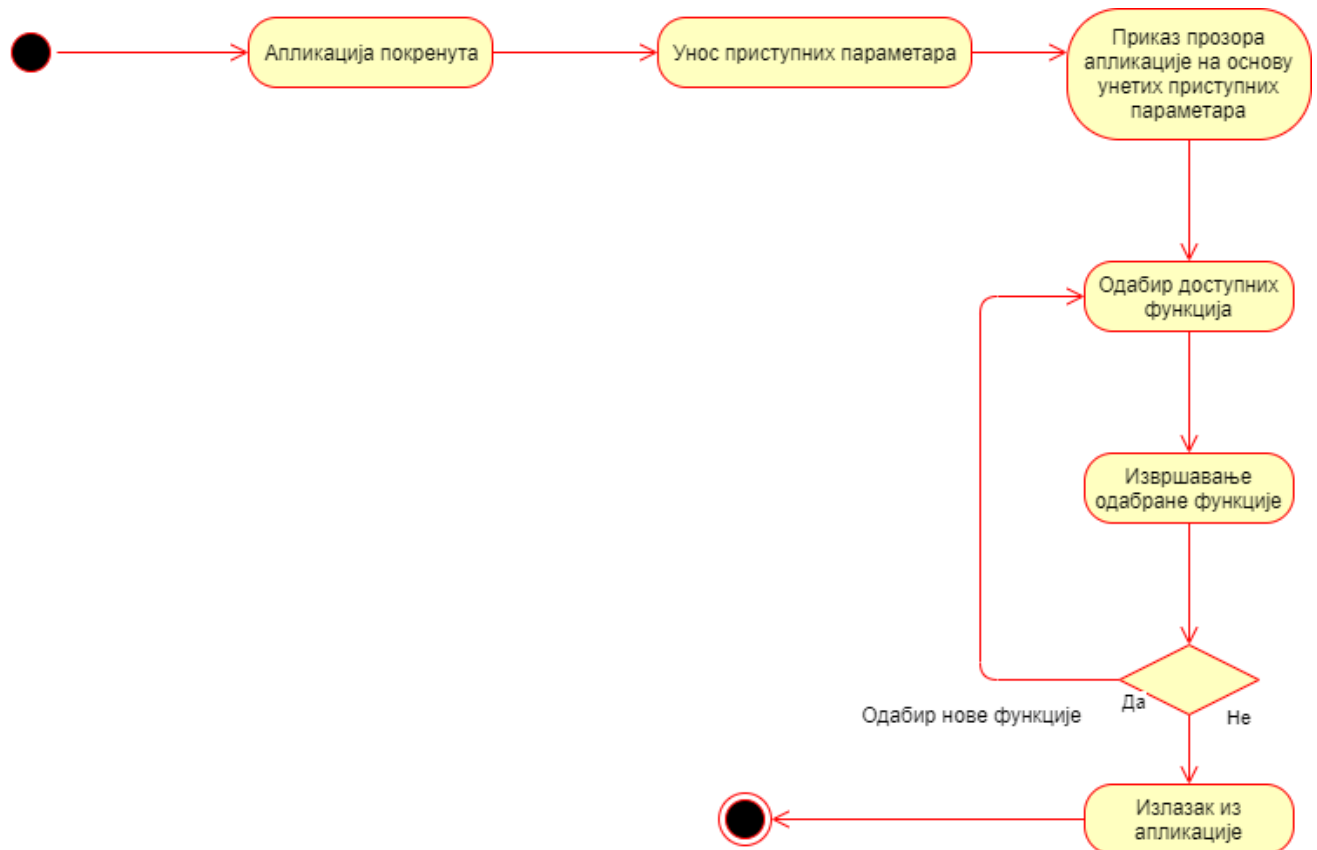
Активност је спецификација понашања која се изражава кроз ток извршења преко секвенцирања и конкурисања подактивности. [21] Акција је основна јединица спецификације понашања која репрезентује неку трансформацију или обраду у моделираном систему.



Слика 22: Дијаграм активности

7.2 Дијаграм машине стања

Дијаграм стања је граф који приказује аутомат (машину) стања - чворови су стања док су гране прелази. Ови дијаграми се углавном фокусирају на ток активности као и на понашање самих догађаја. Они се креирају за ентитете који показују битно динамичко понашање унутар система.



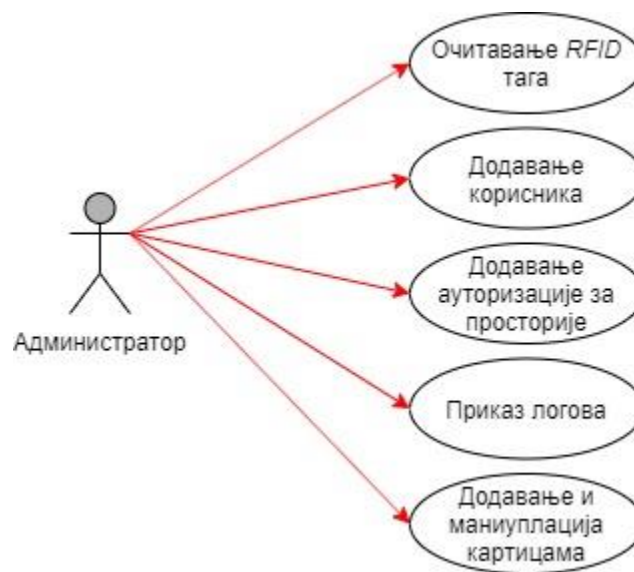
Слика 23: Дијаграм машине стања

7.3 Дијаграм случајева коришћења

Дијаграм случајева коришћења приказује скуп случајева коришћења и актера. Типично се користи да специфицира неку функционалност и понашање неког субјекта - углавном пројектованог система. Дијаграм визуелизује понашање система, подсистема или чак класе и интерфејса. [22]

Ауторизован корисник администратор има следеће случајеве коришћења:

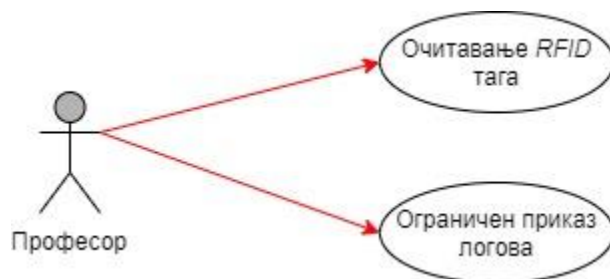
1. Очитавање *RFID* тага
2. Додавање и манипулација картицама
3. Додавање корисника
4. Додавање ауторизације за просторије
5. Приказ логова



Слика 24: Use-case администратора

Ауторизован корисник професор има следеће случајеве коришћења:

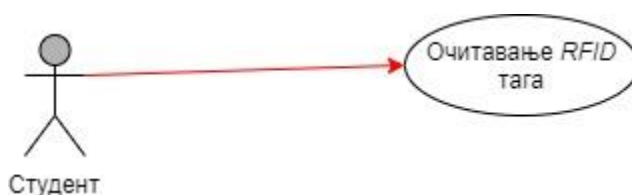
1. Очитавање *RFID* тага
2. Ограничен приказ логова



Слика 25: *Use-case* професора

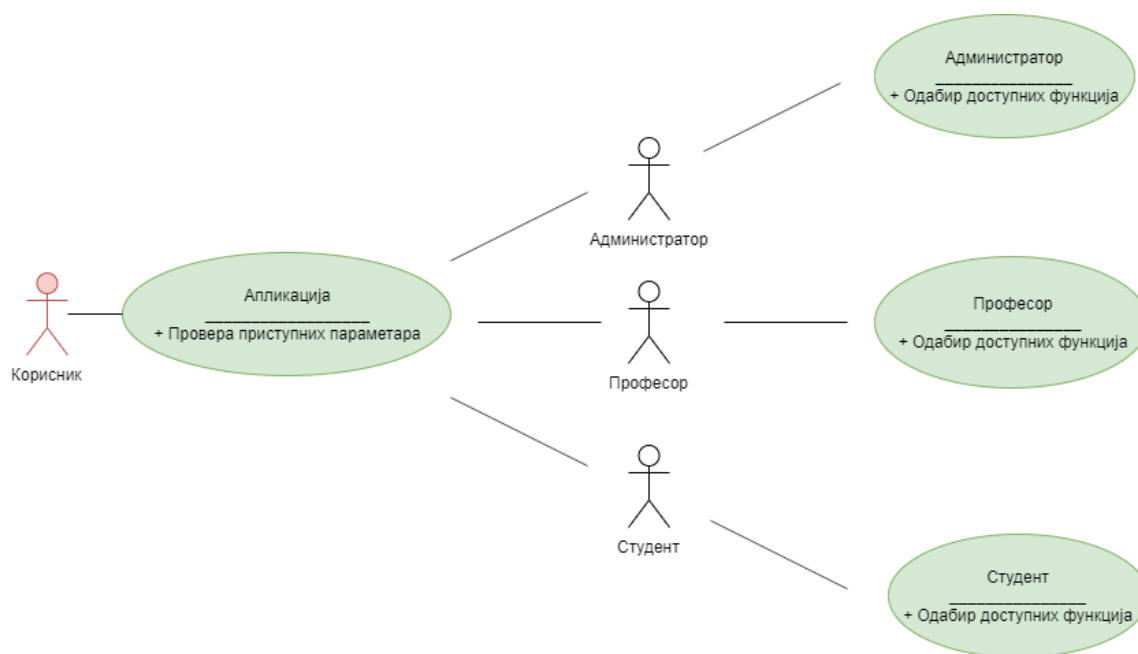
Ауторизован корисник студент има следеће случајеве коришћења:

1. Очитавање *RFID* тага



Слика 26: *Use-case* студента

Приказ дијаграма случајева коришћења за целокупну апликацију дат је на следећој слици.



Слика 27: *Use-case* целокупне апликације

7.4 Дијаграм секвенци

Дијаграм секвенци користи се за спецификацију временских захтева у опису сложених сценарија – опис тока порука између објеката којима се реализује одговарајућа операција у систему. Користи се за моделовање динамичких аспеката система. Идентификују се системски догађаји и системске операције за одговарајући случај употребе. Дијаграм секвенци је један од дијаграма интеракције. Интеракција је понашање које обухвата скуп порука које се размењују између скупа објеката у неком контексту са неком наменом. Порука је спецификација комуникације између објеката која преноси информацију. Пријем поруке изазива акцију (извршење наредбе). [23]

7.4.1 Сваки корисник

Контрола приступа:

1. Корисник прислања *RFID* таг на *RC522* модул, долази до очитавања тага
2. Апликација проверава да ли корисник има право приступа
3. Уписује се лог у апликацију
4. Кориснику се отварају врата



Слика 28: Дијаграм секвенци за очитавање тага

Изузетак:

Корисник покушава да приступи просторији, међутим, ауторизација му није дозвољена.

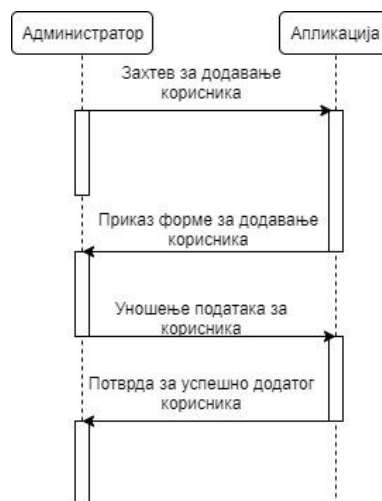


Слика 29: Изузетак приликом читавања тага

7.4.2 Администратор

Додавање новог корисника:

1. Администратор упућује захтев за додавање корисника
2. Апликација одговара слањем форме за унос података
3. Администратор уноси неопходне податке и додељује слободну *RFID* картицу
4. Апликација одговара потврдом за успешно додатог корисника



Слика 30: Дијаграм секвенци за додавање корисника

Изузетак:

Администратор погрешно уноси неки од података (потврду лозинке, лошу електронску адресу или већ постојећу), апликација узвраћа поруком за неуспешни унос корисника.



Слика 31: Изузетак дијаграма секвенци, за погрешно уношење корисника

Манипулација ауторизованим просторијама:

1. Администратор упућује захтев за манипулацију ауторизованим просторијама
2. Апликација одговара слањем табеле за манипулацију ауторизованим просторијама
3. Администратор додаје/брише ауторизацију за дату просторију
4. Апликација одговара потврдом за успешну промену ауторизације



Слика 32: Дијаграм секвенци за промену аутуризације просторија

7.4.3 Професор

Приказ ограничених логова:

1. Професор шаље захтев апликацији за приказ логова
2. Апликација шаље као одговор списак просторија за које професор може да види логове (отуда ограничени логови)
3. Професор бира просторију за коју жели да види логове
4. Апликација одговара приказом логова за дату просторију

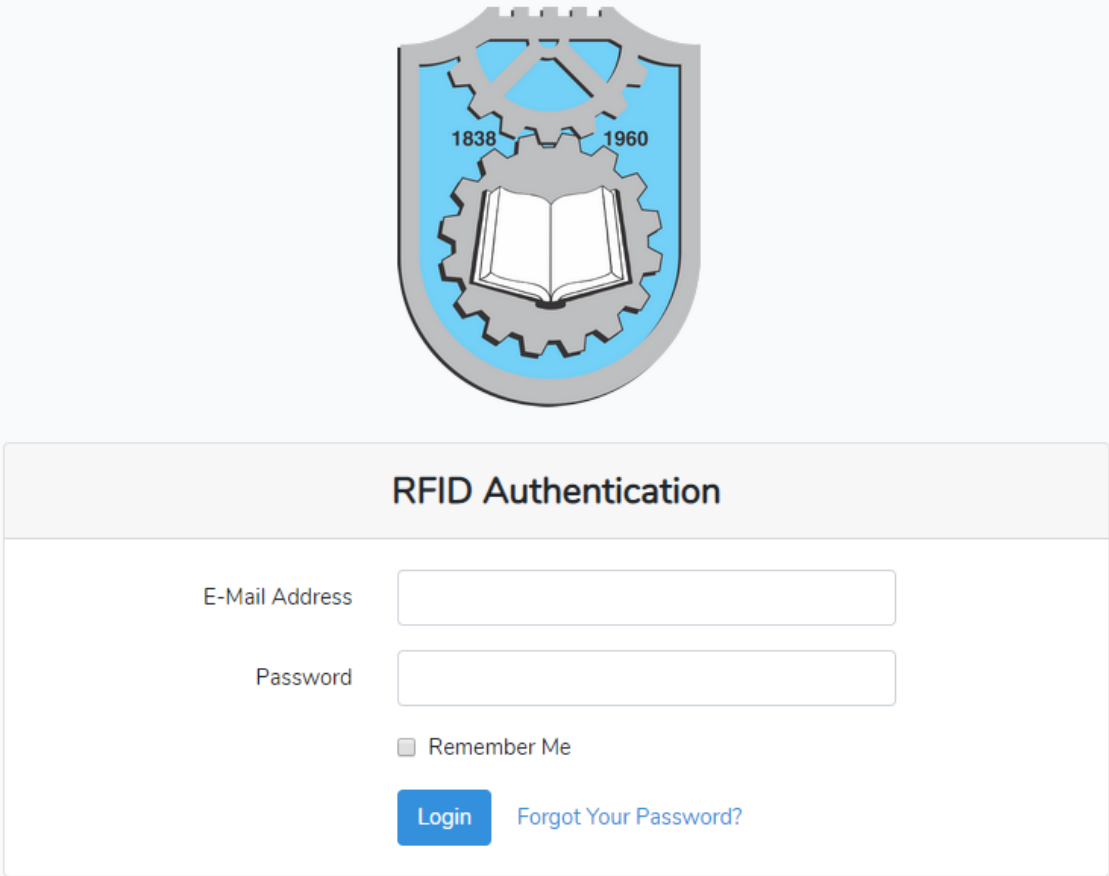


Слика 33: Дијаграм секвенци за приказ логова код професора

8 Приказ решења апликације

Циљ овог одељка јесте да читаоца упозна са начином коришћења ове апликације, као и да му „покаже“ дизајн (на фронту).

Почетни екран целе апликације јесте једна форма за пријављивање (логовање), регистрација је избачена са ове стране, јер је Интернет апликација дизајнирана за академске и универзитетске установе, што значи да се не може свако регистровати. [24]

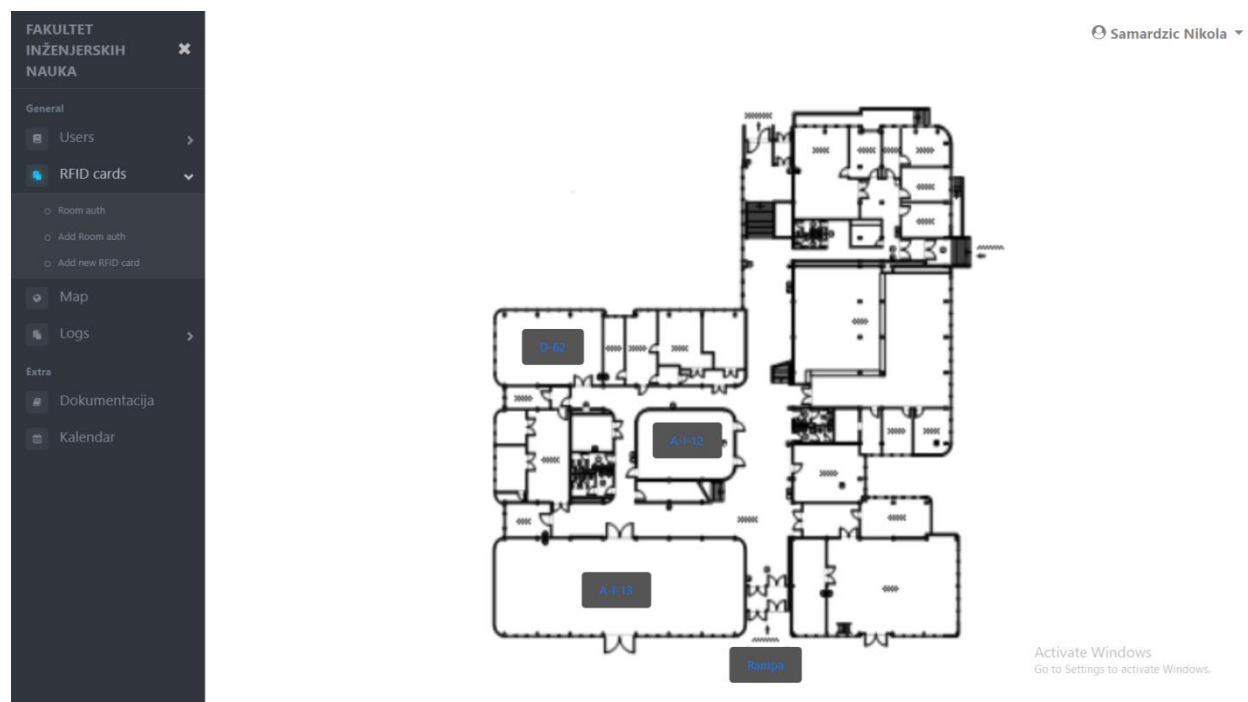


Слика 34: Изглед почетног екрана апликације

Пријављивање у систем је прилично једноставно, и успостављено је глобално решење где се корисник логује преко своје електронске адресе и лозинке. Уколико корисник унесе погрешне креденцијале, бива истог момента обавештен (црвеном поруком испод поља за

унос). Дизајн почетног екрана, као и целе апликације, је прилично једноставан и може се рећи елегантан, са природним бојама, пријатним за око посматрача.

Када се администратор улогује, његова почетна страна је мапа целог факултета или установе, где су просторије обележене (тастерима, на којима пише име дате просторије, слика бр. 35).



Слика 35: Почетни екран администратора

Као што је раније речено, дизајн је елегантан и пријатан за око. Цела апликација је дизајнирана тако да буде интуитивна свим корисницима, различитих образовних профила.

Напомена: Мапа са слике јесте мапа Факултета инжењерских наука, тачније Објекат А, приземље. Убачене учионице нису „праве“ учионице, већ су ради тестирања измишљене.

Кликом на било који од тастера (слика 35), администратор има увид у све логове за дату просторију, што подразумева име и презиме корисника који је приступио просторији, као и тачно време када је ушао. Овакав систем омогућава лакше праћење запослених на факултету, као и студената (рецимо, профсеор тачно може знати ко му је и када присуствовао предавању, што знатно олакшава праћење присутности уколико велики број студената долази на предавање, као што је рецимо амфитеатар).

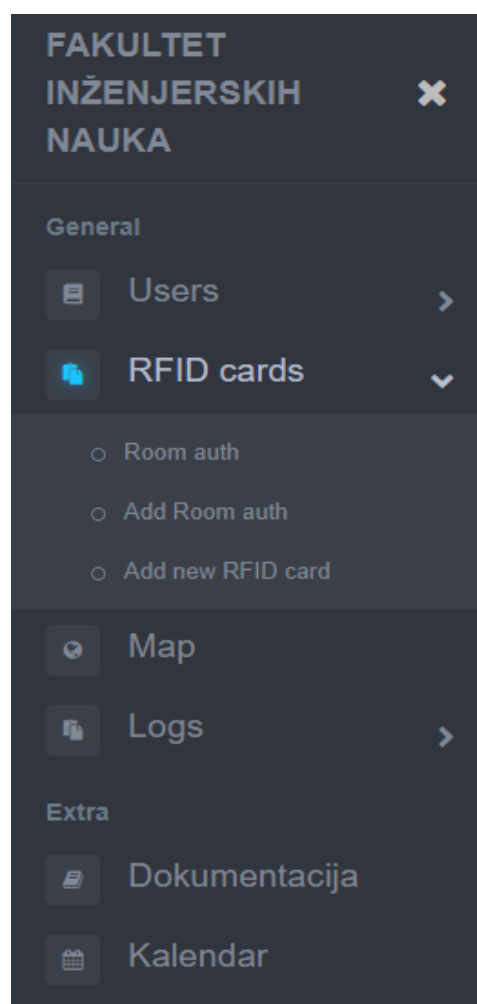
На следећој слици, демонстрирања ради, погледаћемо логове за просторију Д-62.

Search by Room

NAME	SURNAME	ROOM NAME	TIME STAMP
Mika	Lazic	D-62	29/09/2019 18:51:50
Pera	Peric	D-62	29/09/2019 18:51:56
Nikola	Samardzic	D-62	29/09/2019 18:52:11
Pera	Peric	D-62	29/09/2019 18:52:28

Слика 36: Логови просторије Д-62

Можемо видети да су корисници Мика Лазич, Никола Самарџић и Пера Перић (два пута) ушли у просторију Д-62, истог датума и у исто време (29. Септембар 2019, мало пре 7 сати увече). О тастеру *Search by Room* биће речи касније.



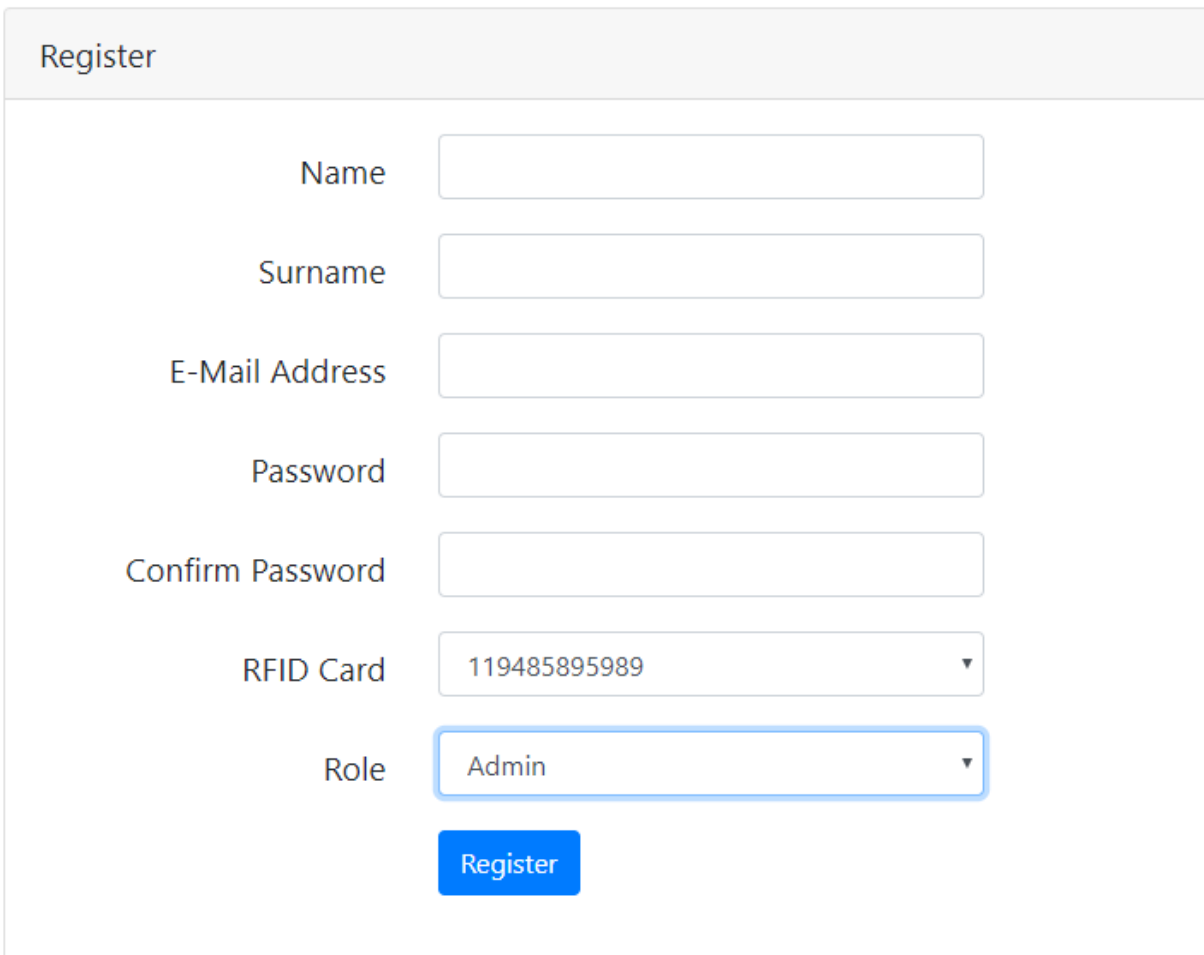
Слика 37: sidebar

Прва ствар која се примећује на почетној страници админа (као и сваког корисника, на свакој страни) јесте тзв. *sidear* (слика 37).

Сврха *sidebara* јесте лакша навигација корисника, у апликацији. Излистава све могуће опције за улогованог корисника (разликује се у односу на дату ауторизацију). На слици 37 могу се уочити све опције којима располаже администратор. Кликом на врх (*Fakultet inženjerskih nauka*) корисник ће бити враћен на своју почетну страницу. Кликом на опцију *User* админ може да бира да ли ће унети новог корисника, или може видети списак свих корисника, као и додељене *RFID* картице. Таб *RFID cards*, који је приказан на слици поред, даје увид у ауторизоване просторије свих картица, додавање/брисање ауторизације за конкретну просторију или уношење нове *RFID* картице. Таб *Map* админу приказује мапу факултета (као и на почетној страници), док кликом на *Logs* може видети све логове, затим претражити исте по кориснику или по просторији.

Кликом на документацију, може видети овај рад у *PDF* формату.

Додавање новог корисника је прилично једноставно и интуитивно (слика 38).



Register

Name

Surname

E-Mail Address

Password

Confirm Password

RFID Card

Role

Register

Слика 38: Додавање новог корисника

Попуњавају се поља, уколико дође до неке грешке администратор ће бити обавештен. *RFID card* и *Role* се разликују, по томе што су опадајуће листе. Респективно, излиставају све слободне картице, односно корисничку ауторизацију. Кликом на *register* дугме, нови корисник (уколико нема грешака) унет је у базу података и има приступ целом систему (Интернет апликацији и *RFID* читачу).

NAME	SURNAME	E-MAIL ADDRESS	RFID	BLOCKED	
Kosta	Eric	kosta@gmail.com	1036091465225	yes	Delete
Nikola	Samardzic	samar@gmail.com	185834545879	no	
Pera	Peric	pera@gmail.com	108479606485	no	Delete
Mika	Lazic	mika@gmail.com	39005023890	no	Delete

Слика 39: Приказ свих корисника

Приказ свих корисника дат је на слици изнад. Поред основних информација, као што су име, презиме, електронска адреса, додељена картица, може се видети да ли је иста блокирана, и могућа је манипулација стања картице. Картица се блокира аутоматски ако корисник покуша 3 пута да приступи недозвољеној просторији (три пута узастопце, принцип сличан банкоматима), међутим, могућа је и манулена блокада картице. Када се картица блокира, једино админ може да јој скине рестрикције, преко ове табеле (слика 39).

Такође, могуће је и брисање корисника, кликом на дугме *Delete*. У том случају, заузета картица се „ослобађа“, међутим рестрикције остају исте. Корисник више не постоји у бази података и губи све привилегије апликације. Може се приметити да корисник *Никола Самарџић* не може бити избрисан, то је зато што је он администратор, ради заштите, њега (као и све остале администраторе) није могуће избрисати из апликације, већ једино директно из базе података.

На слици бр. 40 види се форма за додавање нове картице.

ADD RFID CARD

RFID

Enter id of card

Submit

Слика 40: Форма за додавање нове *RFID* картице

Уколико администратор унесе већ постојећу картицу, систем ће му пријавити грешку.

На следећој слици можемо видети ауторизацију појединачних картица, у смислу којим просторијама наведене картице имају приступ, и да ли су блокиране.

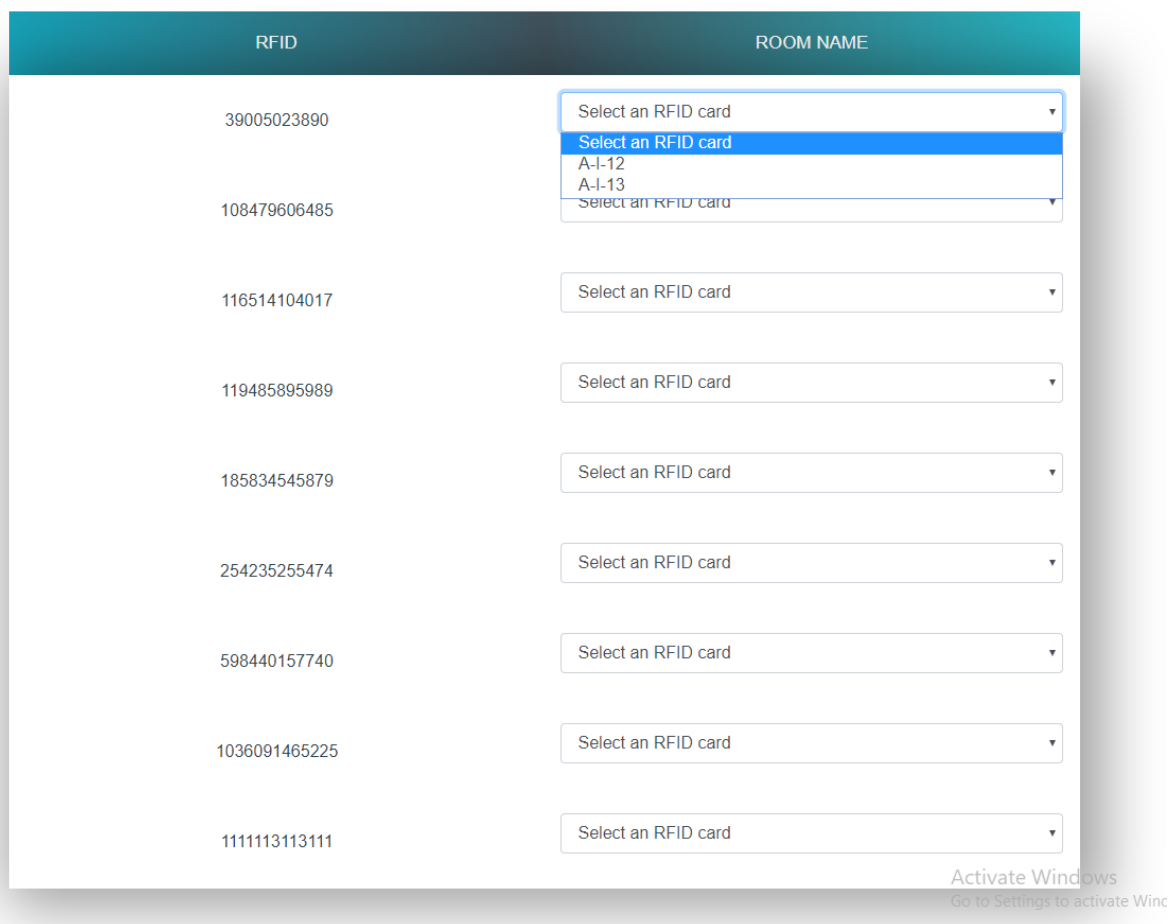
RFID	BLOCKED CARD	ROOM NAME	
39005023890	no	Rampa	Delete
		D-62	Delete
108479606485	no	Rampa	Delete
		A-I-12	Delete
		A-I-13	Delete
185834545879	no	D-62	Delete
		Rampa	Delete
		A-I-12	Delete
		D-62	Delete

Слика 41: Приказ картица

У последња колоне, дугме *Delete*, уклања ауторизацију одговарајуће собе за дату картицу. Нпр. ако администратор кликне на последњи тастер у колони, картица серијског броја 185834545879 губи приступ просторији Д-62.

Као што је већ поменуто, апликација је интуитивна и дизајнирана тако да је свако може научити, веома брзо.

Додавање нових ауторизованих просторија за дату картицу функционише по истом принципу као и брисање ауторизације. (слика 42)



Слика 42: Додавање ауторизације картицама

Поред серијског броја картице, налази се опадајућа листа свих просторија којима картица **нема** приступ. Одабиром једне од ставки из листе, страница се аутоматски освежава и дата картица сада има приступ новој просторији.

На слици 41. сада бисмо видели и нову, додату просторију. Следећи таб у *sidebar*-у јесу мапе, међутим, објашњавање тога било би редундантно, обзиром да отвара почетну страницу која је раније у тексту појашњена.

Пре него што буде објашњена претрага логова, битно је објаснити шта се подразумева када помињемо *лог*. Лог је информација о томе кад је, који корисник приступио одређеној просторији и у које време.

Претрагу логова могуће је урадити на два начина:

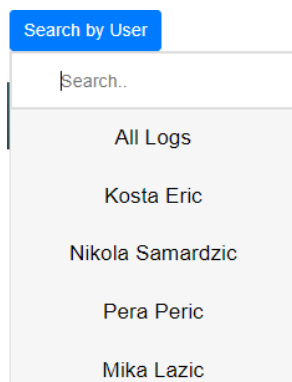
1. По корисницима
2. По просторијама

Обе претраге биће објашњене.

Search by User

NAME	SURNAME	ROOM NAME	TIME STAMP
Nikola	Samardzic	Rampa	22/09/2019 21:46:24
Nikola	Samardzic	Rampa	22/09/2019 21:46:40
Nikola	Samardzic	Rampa	22/09/2019 22:02:35
Nikola	Samardzic	Rampa	22/09/2019 22:03:48
Nikola	Samardzic	Rampa	23/09/2019 14:46:18
Nikola	Samardzic	Rampa	23/09/2019 14:46:55
Nikola	Samardzic	Rampa	23/09/2019 14:47:31
Nikola	Samardzic	A-I-12	29/09/2019 18:53:52
Nikola	Samardzic	A-I-12	29/09/2019 18:54:06
Nikola	Samardzic	D-62	29/09/2019 18:52:11
Pera	Peric	Rampa	29/09/2019 18:50:17
Pera	Peric	A-I-12	29/09/2019 18:53:44
Pera	Peric	A-I-12	29/09/2019 18:54:23
Pera	Peric	D-62	29/09/2019 18:51:56
Pera	Peric	D-62	29/09/2019 18:52:28
Mika	Lazic	Rampa	29/09/2019 18:50:07
Mika	Lazic	D-62	29/09/2019 18:51:50

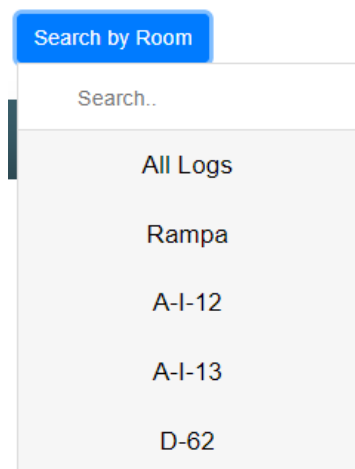
Слика 44: Претрага логова по корисницима



Слика 43: Опадајућа листа корисника

На слици бр. 43, налазе се сви досадашњи логови. Након само једног дана коришћења апликације број логова биће огроман, отуда, ради једноставнијег увида, уведена је претрага, у овом случају по корисницима. Кликом на дугме *Search by User*, добија се опадајућа листа (слика 44). Кликом на име билог ког корисника, можемо видети све логове истог. Међутим, узето је у обзид да би једна образовна институција имала преко 1000 корисника, те да би се олакшала претрага истих, уведена је опција претраге куцањем.

Претрага логова по просторијама урађена је по истом принципу. Сматра се да би број просторија, такође, био велики те је претрага олакшана увођењем преграге куцањем (слика 44). Кликом на било коју просторију, могуће је прегледати све логове дате просторије, такође, кликом на *All Logs* могу се прегледати сви логови.



Слика 45: Опдајућа листа просторија

9 Закључак

Овим радом описана је апликација за праћење ресурса и процеса коришћењем *RFID* технологије, делом реализована у *Laravel frameworku*, делом на *RaspberryPI3*. Читалац је могао да се упозна са применом *RFID* технологије, која иако постоји дуги низ година, још увек није реализовала свој потенцијал. *RFID* нас свуда окружује, како у тржним центрима, обичним самопослугама, тако и у нашим домовима, угравираним у наше кућне љубимце, чак се налази и у кључевима нашим аутомобила.

RFID аутентикација је апликација направљена искључиво за образовне и академске институције, међутим њена примена може бити шири. Може се адаптирати за праћење ресурса и процеса у великим коорпорацијама, чак своју примену може пронаћи и у нашим домаћинствима.

Као што је најављено раније, апликација је замишљена само као један део много већег система, у којем би била омогућена пријава испита СМС-ом, дигитална овера/упис семестра, дигитални потписи професора, алгоритамски распоред испита по салама, увид у оцене и факултетске обавезе (за студенте), увид у предмете, обавезе, пројекте, праћење докумената, као и електронски упис оцена (за професоре).

Тренутно живимо у доба четврте индустријске револуције, које је давно стигло у Западни свет, али полако хвата прикључак и код нас, где овакве апликације могу бити од виталног значаја за рад, не само образовних, већ свих институција, па чак и домаћинстава. Уместо да се плашимо развоја и експанзије модерних технологија, треба да пригласимо оно што оне доносе, а то је потпуно нову димензију и увид у неки нови (сајбер) свет. Апликација описана у овом дипломском раду, управо за циљ има то, тај нови поглед на свет јесте заобилажење непотребне бирократије и олакшавање послова обичног човека.

10 Литература

- [1]
Klaus Schwab, The Fourth Industrial revolution, Currency
Издато: 3. јануар 2017.
[референцирано септембра 2019.]
- [2]
atlasrfidstore.com, *What is RFID?*
Издато: 28. фебруар 2019.
[референцирано септембра 2019.]
Доступно на: <https://www.atlasrfidstore.com/rfid-beginners-guide/>
- [3]
electronics.howstuffworks.com, *Kevin Bonsor & Wesley Fenlon, How RFID Works*
[референцирано септембра 2019.]
Доступно на: <https://electronics.howstuffworks.com/gadgets/high-tech-gadgets/rfid.htm>
- [4]
NXP Semiconductors, MFRC522 - Standard performance MIFARE and NTAG frontend
Издато: 27. април 2016.
[референцирано септембра 2019.]
- [5]
arduino.cc, *A brief Introduction to the Serial Peripheral Interface (SPI)*
[референцирано септембра 2019.]
Доступно на: <https://www.arduino.cc/en/reference/SPI>
- [6]
raspberrypi.org, *What is a Raspberry PI?*, Raspberry PI Fondation
[референцирано септембра 2019.]
Доступно на: <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>
- [7]
raspberrypi.org, *GPIO*, Raspberry PI Fondation
[референцирано септембра 2019.]
Доступно на: <https://www.raspberrypi.org/documentation/usage/gpio/>
- [8]
deviceplus.com, *How to integrate RFID module with RPI*, DevicePlus Editorial Team
Издато: 21. мај 2018.
[референцирано септембра 2019.]
Доступно на: <https://www.deviceplus.com/connect/integrate-rfid-module-raspberry-pi/>

- [9]
github.com, *SPI-Py: Hardware SPI as a C Extension to Python*, Louis Thiery
Издато: 5. јануар 2013.
[референцирано септембра 2019.]
Доступно на: <https://github.com/lthiery/SPI-Py>
- [10]
github.com, *MFRC522-python*, Mario Gomez
Издато: 1. јул 2013.
[референцирано септембра 2019.]
Доступно на: <https://github.com/mxgxw/MFRC522-python>
- [11]
raspberrypi.org, *GPIO in Python*
[референцирано септембра 2019.]
Доступно на:
<https://www.raspberrypi.org/documentation/usage/gpio/python/README.md>
- [12]
techterms.com, *WAMP*
Издато; 23. мај 2013.
[референцирано септембра 2019.]
Доступно на: <https://techterms.com/definition/wamp>
- [13]
hostinger.com, *An In-Depth overview of Apache Web Server*, Gediminas B.
Издато: 18. јануар 2019.
[референцирано септембра 2019.]
Доступно на: <https://www.hostinger.com/tutorials/what-is-apache>
- [14]
dev.mysql.com, *What is MySQL?*
[референцирано септембра 2019.]
Доступно на: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>
- [15]
php.net, *What is PHP+?*
[референцирано септембра 2019.]
Доступно на: <https://www.php.net/manual/en/intro-what-is.php>
- [16]
laravel.com, *Laravel Philosophy*
[референцирано септембра 2019.]
Доступно на: <https://laravel.com/docs/4.2/introduction>

- [17]
blog.pusher.com, *How Laravel implements MVC*, Neo Ighodaro
Издато: 24. мај 2018.
[референцирано септембра 2019.]
Доступно на: <https://blog.pusher.com/laravel-mvc-use/>
- [18]
Бранислав Лазаревић, *Базе података*, Факултет организационих наука
Издато: 2003. Године
[референцирано септембра 2019.]
- [19]
moodle.mfkg.rs, *Базе података: Материјал са вежви и предавања*, Милан Ерић
Издато: јануар 2018.
[референцирано септембра 2019.]
Доступно на: <http://moodle.mfkg.rs/course/view.php?id=526>
- [20]
tallyfy.com, *All you need to know about UML diagrams*
[референцирано септембра 2019.]
Доступно на: <https://tallyfy.com/uml-diagram/>
- [21]
visual-paradigm.com, *Activity Diagram*
[референцирано септембра 2019.]
Доступно на: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>
- [22]
visual-paradigm.com, *Use-case Diagram*
[референцирано септембра 2019.]
Доступно на: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>
- [23]
visual-paradigm.com, *Sequence Diagram*
[референцирано септембра 2019.]
Доступно на: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>
- [24]
laravel.com, *Laravel Philosophy*
[референцирано септембра 2019.]
Доступно на: <https://laravel.com/docs/5.8/authentication>