

**ЈЕДНА ИМПЛЕМЕНТАЦИЈА СИСТЕМА ЗА КОНТРОЛУ ПРИСТУПА
МАКСИМАЛНЕ УПОТРЕБЉИВОСТИ****ONE IMPLEMENTATION OF ACCESS CONTROL SYSTEM WITH MAXIMUM
USABILITY**

Мирко Ивић, Факултет техничких наука, Нови Сад

Област – РАЧУНАРСТВО И АУТОМАТИКА

Кратак садржај – У овом раду представљен је развој једног система за контролу приступа објектима у чијој су имплементацију коришћене различите технологије. На почетку су дефинисани процеси и корисници система. Након тога описан је доменски модел система, а затим и имплементација сваке компоненте система.

Кључне речи: Контрола приступа, Ардуино, RFID, NFC, Веб апликација,

Abstract – This paper presents the development of access control system for facilities, in which implementation various technologies were used. At the beginning, processes and users of the system are defined. After that, the domain model of the system is described, followed by the implementation of each component of the system.

Keywords: Access control, Arduino, RFID, NFC, , Web application, Web application

1. УВОД

Контролисан приступ стамбеним и пословним објектима представља значајан фактор за безбедност и заштиту особа које бораве унутра, као и поверљивих докумената чије би компромитовање могло да угрози пословање. Систем, који је тема овог рада, представља решење за контролу приступа као и за контролу кретања унутар једног предузећа. Осим запослених, у предузеће долазе разни посетиоци, којима треба омогућити ауторизован приступ објекту. Осим контроле приступа, систем има и могућност бележења колико времена запослени проведе на послу у току радног дана. На овај начин обезбеђује се максимална употребљивост и аутоматизација у раду, а смањује могућност грешке и злоупотребе. На брз и поуздан начин бележи се радно време запосленог, а степен безбедности унутар објекта се подиже на виши ниво.

Најпоузданији начин за идентификацију особа је свакако идентификација путем скенирања отиска прста или зенице ока. Међутим, ови подаци спадају у биометријске податке особе што захтева посебне

дозволе од надлежних институција и посебан третман приликом чувања ових података. У овом случају, паметни телефони, које велики део популације користи, представља идеалан уређај. Запослени не може бити приморан да поседује паметни уређај. Због тога је за такве кориснике потребно имплементирати и алтернативни начин идентификације.

2. ОПИС СИСТЕМА

У систему се препознају три типа корисника, тј три улоге:

- администратор,
- запослени,
- посетилац.

Улоге администратор и запослени имају директну интеракцију са системом, што значи да преко апликације имају могућност извршавања акција над ентитетима и мењања стања система. Улога посетилац непосредно интерагује са системом, тј. посетилац не користи софтвер директно и није свестан своје интеракције са системом.

Већина функционалности су у надлежности администратора, који има главну улогу и уређује систем према потребама предузећа. Задатак администратора је да одржава систем у конзистентном стању, тј. да систем у сваком моменту одсликава реално стање ствари у предузећу.

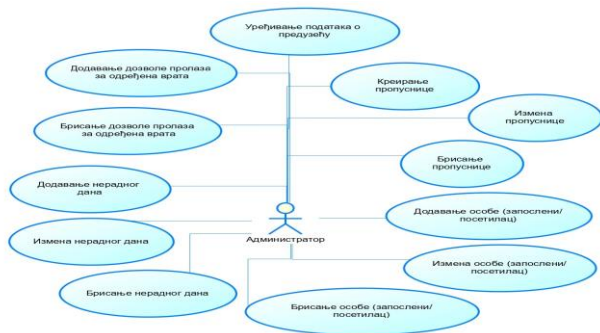
На слици 1. и на слици 2., путем дијаграма случајева коришћења [2], приказане су функционалности које су доступне администратору.



Слика 1. Функционалности доступне администратору (први део)

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Драган Иветић, ред. проф.



Слика 2. Функционалности доступне администратору (други део)

Фокус система је да омогући запосленима ауторизован приступ предузећу и унутрашњим просторијама истог. Из тог разлога запослени има знатно мање доступних функционалности у односу на корисника типа администратор. Осим чекирања на вратима, запослени може да формира захтев за годишњи одмор и да преузима одређене документе. С обзиром на то да администратор ради у предузећу, њему су такође доступне све функционалности које има и запослени.

Као што је већ речено, корисник типа посетилац нема директне интеракције са системом. Једна функционалност која је њему додељена је чекирање на вратима. Њему се издаје привремена пропусница како би имао приступ тачно одређеним просторијама унутар предузећа и на тај начин се спречио неовлашћени приступ просторијама у којима се налазе поверљиви документи или одвијају процеси који су део пословне тајне.

Систем има могућност генерисања следећих докумената:

- потврда о радном односу, тј. потврда о запослењу,
- листинг са именима свих запослених и њиховим подацима,
- листинг са именима свих запослених и укупним бројем радних сати са жељени временски период,
- листинг са датумом и временом откуцавања запосленог у току месеца и бројем сати проведених на радном месту за сваки дан.

Потврда о радном односу се генерише на основу захтева запосленог и може бити извезена (експортирана) у *PDF* (*Portable Document Format*) формату. Потврда треба да садржи личне податке запосленог, податке о предузећу и намену за коју се потврда издаје. Листинг са именима свих запослених и њиховим подацима се генерише на захтев администратора и може бити извезен у *PDF* или *CSV* (*Comma-Separated Values*) формату. Пре самог генерисања администратор може да одреди по ком критеријуму ће листинг бити сортиран. Овај листинг поред имена и презимена запослених садржи и додатне податке о њима попут адресе, јмбг-а, занимања, позиције итд. Листинг са именима свих запослених и укупним бројем радних сати за жељени временски период генерише се на захтев администратора и може бити извезен у *PDF* или *CSV* формату. Администратор пре самог генерисања

дефинише временски период који га интересује и критеријум по ком ће листинг бити сортиран. На документу се налазе датуми за изабрани период, подаци о запосленима и збир сати проведених на послу за сваког запосленог. Листинг са датумом и временом откуцавања запосленог и бројем радних сати за сваки дан у току месеца се генерише на захтев запосленог или администратора и може бити извезен у *PDF* или *CSV* формату. На листингу се налазе подаци о запосленом и назив месеца за који се листинг издаје. Осим тога, приказани су и појединачни подаци за сваки дан у месецу као што су време доласка и одласка са посла и разлика та два времена тј. колико је времена запослени провео на послу тог дана.

Да би систем осликавао реално стање у предузећу задужен је администратор. Путем апликативног софтвера администратор уноси податке о предузећу и о ентитетима унутар предузећа као што су: врата, нерадни дани, позиције, запослени, посетиоци итд.

Приликом запошљавања новог запосленог администратор уноси његове податке у систем. Након тога запосленом се додељује једна или више пропусница које може да користи за приступ предузећу и кретање кроз њега. Запосленом се издаје перманентна пропусница, тј. пропусница без датума истека. Уколико запосленом престaje радни однос, тада се дефинише датум престанка валидности пропуснице. Доласком до одређених врата запослени користи своју пропусницу ради идентификације. Потом систем врши проверу да ли тај запослени има дозволу за пролаз. Уколико има, биће му омогућен пролаз кроз врата, у супротном биће му сигнализировано да нема приступ тим вратима. Такође, пропусница се издаје и за посетиоце. За разлику од пропуснице која се издаје запосленом, посетиоцу не може бити издата перманентна пропусница. Када су нека од врата дефинисана као улазно-излазна, тада се приликом дозволе пролаза, запосленом бележи датум и време проласка. На основу тих записа, остварује се могућност евиденције колико је времена запослени провео на свом радном месту.

Евиденцију коришћења годишњих одмора и боловања води администратор. Његов задатак је да бележи свако коришћење како би се на ефикасан начин водила евиденција колико дана одмора је запослени искористио у току године, тј. колико дана је провео на боловању. Запослени би требало да свој годишњи одмор најави унапред тако што поднесе захтев. Администратор врши одобравање захтева уколико одсуство запосленог за тај период не ремети унапред испланирану активност предузећа.

3. МОДЕЛ ПОДАТАКА

Централни ентитети овог система су *Preduzece* и *Osoba*. Све битне информације које описују један пословни субјекат попут назива, адресе, телефона, матичног броја се налазе у истоименом ентитету. За поља која репрезентују ПИБ и матични број предузећа потребно је увести ограничење јединствености тј. забрану дефинисања два ентитета предузећа са истим вредностима за ова два поља.

Нерадни дани у предузећу дефинисани су путем ентитета *NeradniDan*. Ентитет садржи поља датума нерадног дана и описа који објашњава зашто се тог дана није радило. Поред ова два поља постоји и поље *BrojSati* које, у предузећима где се зарада рачуна на основу броја радних сати, говори колико се радних сати приписује раднику тог дана. Врата на којима се врши контрола приступа репрезентована су путем ентитета *Vrata*. Једно од поља овог ентитета је и поље *Ulazna* типа *boolean*. На основу овог поља дефинише се да ли су одређена врата улазно-излазна тј. да ли се особи приликом проласка бележи време и датум. Ентитетом *Pozicija* дефинисане су позиције унутар једног предузећа.

Ентитет *Osoba* репрезентује људе који интерагују са системом. У овом ентитету налаза се поља која дефинишу једног човека унутар система као на пример име, презиме, адреса, број телефона итд. Ентитети *Zapolseni* и *Posetilac* наслеђују ентитет *Osoba*. Како би се запослени могао пријављивати на систем истоименом ентитету додата су поља корисничко име и лозинка. Тренутно, ентитету *Posetilac* нису додата никаква додатна поља. Разлог имплементирања у три табеле је у томе што би у случају једне табеле, у којој су и запослени и посетиоци, дошло до усложњавања проблема. Тада би приликом уноса новог посетиоца требало изоставити унос података за поља која су везана само за запосленог, а нека од тих поља су неопходна у случају уноса запосленог. Ту долази до компликације приликом валидације уноса која је избегнута овако дизајнираним моделом. Свим особама у систему додељују се пропуснице како би се могли чекирати на вратима. Ентитет *Propusnica* садржи податке о пропусници попут датума издавања, датума истека и којег је типа. Датум истека дефинише се уколико је пропусница додељена посетиоцу или уколико се зна датум престанка радног односа запосленог. За сваку позицију унутар предузећа, ентитетом *PozicijaVrataPreduzece* унапред су предодређена врата за која та позиција има пролаз. То значи да је за додељивање дозволе пролаза особи, довољно да се приликом уноса особе у систем дефинише њена позиција. Дозволе пролаза, које се односе на неку особу и одређена врата, репрезентоване су путем ентитета *DozvolaProlaza*. На овај начин остварена је могућност прилогађавања приликом доделе пропусница, тј. особа може имати приступ још неким вратима, иако то није предвиђено у односу на позицији. Такође, приступ неким вратима може бити и одузет.

Евидентирање годишњег одмора имплементирано је коришћењем ентитета *GodisnjiOdmor* и *KoriscenjeOdmora*. Календарску годину током које запослени може да користи годишњи одмор описује ентитет *GodisnjiOdmor*. Овај ентитет садржи поље које даје информацију о календарској години, чија вредност мора бити јединствена за једног запосленог, и поље које показује колико је дана годишњег одмора запослени искористио у тој години.

Коришћење одмора у години евидентра се путем ентитета *KoriscenjeOdmora*. Најпре треба поменути да

ентитет *KoriscenjeOdmora* садржи поље *StatusGodisnjiOdmor* које је типа *enum*, а могуће вредности су *Predlog* и *Prihvacen*. Ово поље сигнализира да ли је то изнети предлог запосленог у вези са годишњим одмором или годишњи који треба урачунати у укупан збир дана.

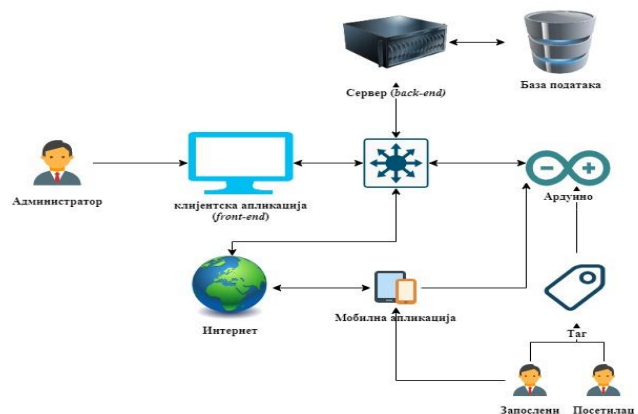
Осим овог поља, ентитет *KoriscenjeOdmora* садржи податке о датуму почетка и датуму завршетка одмора и броју искоришћених дана. Приликом рачунања броја искоришћених дана не рачунају се дани који су у предузећу дефинисани као нерадни.

Имплементација евидентирања боловања решена је на сличан начин као евиденција годишњих одмора. Евиденција боловања садржи ентитете *Bolovanje* и *KoriscenjeBolovanja*. Ентитет *Bolovanje* описује календарску годину у којој запослени користи боловање.

Поље *Godina* представља годину коју описује ентитет и вредност овог поља мора бити јединствена за запосленог. Вредност поља *BrojDana* представља укупан број дана које је запослени провео на боловању током године. Појединачно коришћење боловања у години евидентира се помоћу ентитета *KoriscenjeBolovanja*. Овај ентитет садржи податке о датуму отварања, датуму затварања и броју дана проведених на боловању том приликом.

4. ИМПЛЕМЕНТАЦИЈА СИСТЕМА

Систем је изграђен од неколико компоненти које међусобно интерагују. У наставку је описана свака појединачна компонента архитектуре система уз додатни опис технологија које су кориштене приликом имплементације. На слици 3. приказана је архитектура система.



Слика 3. Архитектура система

4.1 Серверска апликација

Серверска апликација заузима централно место у систему. Све остале компоненте комуницирају са њом или преко ње. За развијање серверске апликације коришћено је *IntelliJ IDEA* интегрисано радно окружење. Програмски код написан је у програмском језику *Java* верзије 17 и у радном оквиру *Spring Boot*.

Приликом дизајнирањаа дизајнирање комуникације између сервера са клијентом и ардуином коришћен је *REST* архитектонски стил [3], а сама комуникација се одвија путем *HTTP* протокола.

4.2 База података

Подаци који циркулишу унутар система чувају се у јединственој бази података. За потребе овог система коришћена је *SQL Server 2012* релациона база података, а комуникација сервера са базом података остварена је уз помоћ *Hibernate JPA (Java Persistence API)* радног оквира. База података генерисана је *Code-First* методом. То значи да су прво направљене класе модела, а потом на основу анотација које се додају пољима класе *Hibernate JPA* креира табеле унутар базе. Мапирање доменског модела на базу података врши се на следећи начин. На основу ентитета доменског модела формиране су објектне класе, а потом коришћењем функционалности *Hibernate JPA*, објектне класе су намапиране на табеле унутар базе података. Атрибути ентитета су мапирани на поља објектне класе, а они на колоне у табели. Сваки ентитет поседује атрибут *id*, који представља примарни кључ ентитета.

4.3 Клијентска апликација

Преко клијентске апликације администратор врши ажурирање података унутар система. Администратор има своје креденцијале помоћу којих се пријављује на систем. Приступ систему није дозвољен непријављеним корисницима. Клијентска апликација је развијана у *Visual Studio Code* интегрисаном радном окружењу, а имплементирана је коришћењем *Angular JavaScript* библиотеке верзије 15.0.2.

Приликом уноса података, администратор користи различите форме за унос. Приликом уноса потребно је обезбедити валидацију унетих података. За валидацију унетих података кориштени су *Strategy* шаблон и *Angular*-ове библиотеке за валидацију форме уноса. Код *Strategy* шаблона свака класа модела имплементира интерфејс са методом *validate*. Унутар сваке класе модела, метода *validate* је редефинисана и имплементирана тако да се изврши валидација према захтевима за специфичну класу.

4.4 Ардуино платформа

Ардуино је платформа отвореног кода (енг. *open-source*) која омогућава реализацију разноврсних пројеката у домену електронике [4]. Једна од великих предности ове платформе јесте њена отвореност. То значи да су сви делови платформе отворени и доступни за проучавање и измене. Ардуино штит (енг. *Shield*) је хардверска компонента која се надограђује на врх микроконтролера и садржи склопове који су специјализовани за одређену намену. Ово значи да корисник, уколико жели да омогући контролеру приступ интернету или управљање електро мотором, не мора самостално да развија модул за ту функционалност. Уместо тога, довољно је да прикључи одговарајући штит контролеру и користи већ доступне библиотеке како би управљао том додатном функционалношћу.

Ардуино контролер сам по себи нема могућност приступа интернету, нити могућност детектовања *RFID* или *NFC* сигнала. За потребе система ардуино уређаји придодати су интернет штит (енг. *Ethernet Shield*) *W5100* и *RFID* читач *RC522*. Захваљујући овим додацима, ардуино има приступ интернету и

могућност читања *RFID* тагова. Пошто овај читач ради на фреквенцији од 13,56 MHz, то му омогућава детектовање и *NFC* сигнала.

4.5 Андроид апликација

Мобилна апликација комуницира са сервером преко интернета. Запослени користе апликацију ради преузимања докумената и увида у своје радне сате, а запослени чији паметни телефони имају *NFC* могу се чекирати на вратима и помоћу паметног телефона. У супротном, запослени користе *RFID* таг за чекирање. Приликом развијања мобилне апликације коришћено је *Android Studio* интегрисано развојно окружење, а апликација је рађена у андроид радном оквиру и писана у *Java* програмском језику.

5. ЗАКЉУЧАК

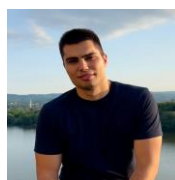
Тема овог рада јесте креирање система за контролу приступа објекту једног предузећа употребом различитих технологија ради добијања максималне ефикасности у раду. Представљен је систем који путем додељених пропусница препознаје особе на вратима, а затим на основу дефинисаних дозвола омогућава или забрањује даљи пролаз. Такође, једна од бенефиција овог система јесте аутоматско израчунавање броја радних часова које запослени порведе у предузећу.

Даљи правци развоја и унапређивања система може бити имплементација *HTTPS (Hypertext Transfer Protocol Secure)* протокола како би се осигурала сигурна комуникација како између клијента и сервера, тако и између андорид апликације и сервера. Ардуино платформа се показала врло захвалном за интеграцију унутар система. Доступност компоненти и кодова за учење знатно су олакшали имплементацију. Међутим, *RFID* читач *RC522* се није нарочито показао приликом употребе са *NFC*-ом. Иако читач ради на фреквенцији од 13,56 MHz, дешавало се да не успева да прочита сигнал који му је упућен од стране паметног телефона.

6. ЛИТЕРАТУРА

- [1] Службени гласник РС, „Правно информациона систем“, https://www.paragraf.rs/propisi/zakon_o_zastiti_podat_aka_o_licnosti.html.
- [2] UML, „UML Use Case Diagrams“, <https://www.uml-diagrams.org/use-case-diagrams.html>.
- [3] R. Thomas Fielding, *Architectural Styles and the Design of Network -Based Software Architectures*, ProQuest Dissertations Publishing, 2000, pp.76-105.
- [4] Arduino, „What is Arduino?“, <https://docs.arduino.cc/learn/starting-guide/whats-arduino>.

Кратка биографија:



Мирко Ивић рођен је у Новом Саду 1996. год. Основне академске студије завршио је 2020. год. на Факултету техничких наука у Новом Саду. Мастер рад на Факултету техничких наука из области Рачунарство и аутоматика одбранио је 2023.