Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Мамандығы 5В070400 – Есептеу техникасы және бағдарламалық қамтамасыз ету

Студент Әбдіманап Ғалымжан

Тақырыбы Университет аудиториясына уақытты ескере отырып кіруді бақылау жүйесін құру

ҒЫЛЫМИ ЖЕТЕКШІНІҢ СЫН-ПІКІРІ

Диплом жобасын жасаушы Әбдіманап Ғалымжанның алдына университет аудиториясына уақытты ескере отырып кіруді бақылау жүйесін құру тапсырмасы қойылған.

Дипломдық жобада университет немесе кез келген ғимарат аудиториясына уақытты ескере отырып кіруге рұқсат беретін жүйе құрылған. Жүйе интернетке шыға алатын, веб браузерді қолдайтын кез келген операциондық жүйеде жұмыс жасайды. Жүйе магнитті карта арқылы және адам бетін идентификациялау арқылы жұмыс жасайды.

Жұмыс барысындағы басты мәселелер: аудиторияға кіру жүйесін құруда қолданылатын әдістерді талдау, жүйенің интерфейсі үшін бағдарламалық қамтамасыз етуді әзірлеу құралдарын таңдау, жүйеның камерадан адам бетін іздеу және идентификациялау алгоритмін іске асыру, әрі қарай дамыту және программалық камтаманың модулдігін қамтамасыз ету болып табылады. Бұл қазіргі заманауи ең өзекті және келешегі бар технологияларды қолданудағы маңызды мәселелердің бірі.

Менің пікірімше, диплом жазушы алдына қойылған тапсырманы толығымен орындады және ақпараттық жүйелердің заманауи технологияларын меңгергендігін көрсетті.

Жоба жетекшісі ретінде бұл дипломдық жобаны өз деңгейіне сәйкес деп есептей отырып Әбдіманап Ғалымжанға 5В070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету» мамандығы бойынша «Техника және технологиялар бакалавры» академиялық дәрежесін тағайындауға болады деп есептеймін.

ҒЫЛЫМИ ЖЕТЕКШІ _____ **М. Канат**
«Компьютерлік және программалық инженерия»
кафедрасының лекторы «30» 04 2019 жыл

Отчет подобия

Университет:	Satbayev University
Название:	Abdimanap_vpb_15_1k.docx
Автор:	Әбдіманап Ғалымжан
Координатор:	Мақсат Қанатов
Дата отчета:	2019-05-03 17:17:48
Коэффициент подобия № 1:	**6,0%**
Коэффициент подобия № 2:	**3,0%**
Длина фразы для коэффициента подобия № 2:	25
Количество слов:	4 093
Число знаков:	32 765
Адреса пропущенные при проверке:	
Количество завершенных проверок:	2

Ғылыми жетекші _____ М. Қанат

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Программалық инженерия кафедрасы

Әбдіманап Ғалымжан Сейтахметғалиұлы

«Университет аудиториясына уақытты ескере отырып кіруді бақылау жүйесін құру»

Дипломдық жобаға

ТҮСІНІКТЕМЕЛІК ЖАЗБА

5В070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету» мамандығы

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Программалық инженерия кафедрасы

ҚОРҒАУҒА ЖІБЕРІЛДІ
Кафедра меңгерушісі
тех. ғыл. кандидаты, доцент,
ассистен-профессор
_____ Р. Юнусов
"17" мая 2019 ж.

Дипломдық жобаға
ТҮСІНІКТЕМЕЛІК ЖАЗБА

Тақырыбы: «Университет аудиториясына уақытты ескере отырып кіруді бақылау жүйесін құру»

5В070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»

Орындаған Ғ. С. Әбдіманап

Ғылыми жетекші
техн. ғыл. магистрі, лектор
_____ М. Қанат
"30" 04 2019 ж.

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Программалық инженерия кафедрасы

5В070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»

БЕКІТЕМІН
Кафедра меңгерушісі
тех. ғыл. кандидаты, доцент,
ассистен-профессор
_____ Р. Юнусов
"07" мая 2019 ж.

Дипломдық жоба орындауға
ТАПСЫРМА

Білім алушы: *Әбдіманап Ғалымжан Сейтахметғалиұлына*
Тақырыбы: *«Университет аудиториясына уақытты ескере отырып кіруді бақылау жүйесін құру»*
Университет академиялық мәселелер жөніндегі проректорының бұйрығымен *«14» наурыз 2018ж. № 1841-б шешімімен бекітілген.*
Орындалған жобаның өткізу мерзімі *«13» мамыр 2019 ж.*
Дипломдық жұмыстың бастапқы мәліметтері: *Кіруге рұқсат беретін программалық қамтамалардың жұмыс істеу принципі, бағдарламалық қамтаманы, қосымшаны қолданушылардың, өтінімді орындау қызметін атқаратын әкімші-басқарушының деректер қорына енгізу жүйелерін ұйымдастыру.*
Есеп – түсініктеме жазбаның талқылауға берілген сұрақтардың тізімі:
а) аналитикалық шолу;
б) жобалау бөлімі;
в) өңдеу құралдары және жобаны талдау;
г) қолданбалы бөлім;
д) А Қосымшасы – техникалық тапсырма.
е) Б Қосымшасы – бағдарлама мәтіні.
Жобаның презентациялық 18 слайды ұсынылған.
Ұсынылған негізгі әдебиеттер: *26 атау.*

Дипломдық жобаны орындау
КЕСТЕСІ

Бөлімдердің атаулары, зерттелген мәселелердің тізімі	**Ғылыми жетекшіге және кеңесшілерге ұсыну мерзімі**	**Ескерту**
1. Дипломдық жобаның жоспарын құру	14.01.2019	ҰСОҚ
2. Тапсырма қойылымы және бағдарламалау ортасын таңдау	18.01.2019	ҰСОҚ
3. Бағдарламалық қамтаманы жобалау және талдау.	01.02.2019	ҰСОҚ
4. Бағдарламаны әзірлеу	15.02.2019	ҰСОҚ
5. Жоба бағдарламасын тестілеу	18.03.2019	ҰСОҚ
6. Дипломдық жоба түсіндірме жазбасын жазу	26.04.2019	ҰСОҚ

Дипломдық жұмыс бөлімдерінің кеңесшілерінің аяқталған жұмысқа қойған **қолтаңбалары**

Бөлімдер атауы	**Кеңес берушілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)**	**Қолтаңба қойылған мерзімі**	**Қолы**
Нормалық бақылаушы	Таурбекова А. А. лектор	04.05.19	
Бағдарламалық бөлім	Қалдыбеков С. Б. сениор-лектор	06.05.19	

Ғылыми жетекші _____ М. Қанат

Тапсырманы орындауға қабылдап алған студент _____ Ғ. С.Әбдіманап

Күні «3» қазан 2018 ж.

АҢДАТПА

Машина (техникалық) жүйелерінің жүйесі визуалды ақпаратты талдау арқылы өндірілетін өнімдерді бақылауды және өндірістік процестерді басқаруды автоматтандыруға мүмкіндік береді. Бейнелерді қалыптастыру үшін бейнекамералар пайдаланылады. Компьютерлік көру жүйелерінің бағдарламалық жасақтамасы бұл ақпаратты операторға, автоматтандырылған процестерді басқару жүйесіне, роботқа немесе тікелей өндіріс процесін басқаруға арналған атқарушы механизмдерге жібереді. Машиналық көру жүйелері, талданатын ақпараттың көлемі, жылдамдығы немесе күрделілігі оператордың қабілетінен асып кететін жағдайларда тиімді.

Құру объектісі – университеттің аудиторияларына уақытты ескере отырып кіруді бақылау және басқару жүйесі. Бағдарламалық бөлік магниттік карталарды оқу программалық қамтамасымен және бетті идентификациялау программалық қамтамасынан тұрады. Аппараттық бөлік кіріктірілген электрондық карталарды оқу құрылғысы және ip-бейне камералар болып табылады. Бағдарламалық қамтама python және java бағдарламалау тілдерінде жазылған. Жүйе қызметкердің физикалық күшін үнемдеу және университеттің аудиториясына қызметкерлерге қол жетімділікті бақылауды күшейту мақсатында құрылды.

АННОТАЦИЯ

Системы машинного (технического) зрения позволяют автоматизировать контроль выпускаемой продукции и управление производственными процессами путем анализа визуальной информации. Для формирования изображений используются промышленные видеокамеры. Программное обеспечение систем машинного зрения анализирует увиденное и передает эту информацию оператору, АСУ ТП, роботу или напрямую исполнительным механизмам для управления производственным процессом. Системы машинного зрения особенно эффективны в тех случаях, когда объем, скорость или сложность анализируемой информации существенно превышает способности оператора.

Объектом разработки является система контроля и управления доступом в аудитории университета с учетом времени , состоящая из программной и аппаратной части. Программная часть состоит из считывателя магнитных карт и идентификации лиц. Аппаратной частью являются контроллеры со встроенным считывателем электронных карт и ip-видеокамеры. Программная часть написана на языке программирования python и java. Система разрабатывалась с целью экономии физической силы сотрудника и повышения контроля доступа персонала в аудиторию университета.

ABSTRACT

Systems of machine (technical) vision allow you to automate the control of manufactured products and the management of production processes by analyzing visual information. For the formation of images used industrial camcorders. The software of computer vision systems analyzes the seen and transfers this information to the operator, the automated process control system, the robot, or directly to the executive mechanisms for controlling the production process. Machine vision systems are particularly effective in cases where the volume, speed or complexity of the analyzed information significantly exceeds the ability of the operator.

The object of the development is a control and management system for access to the university's lecture hall, consisting of software and hardware. The software part consists of a magnetic card reader and face identification. The hardware part are controllers with built-in electronic card reader and ip-video camera. The software is written in the programming languages python and java. The system was developed with the aim of facilitating the physical strength of an employee and increasing the control of personnel access to the university auditorium.

МАЗМҰНЫ

КІРІСПЕ

Қатынасты басқару жүйесі (АСС) – АБЖ объектілеріне қолжетімділікті бақылауға және қорғалатын аумақта адамдардың қозғалысына мониторинг жүргізуге мүмкіндік беретін техникалық құралдар мен ұйымдастырушылық шаралар жиынтығы [16]. Қазіргі уақытта қол жеткізуді бақылау жүйелері объектілер үшін кешенді қауіпсіздік тапсырмаларын шешудің ең тиімді әдістерінің бірі деп танылады.

Дипломдық проектінің өзектілігі аудиторияға кіруге рұқсат беру жолдарын автоматтандыру болып табылады. Проектінің қызметі аудиторияға кіруге келген адамды идентификациялау және автоматты түрде базаға түсіретін веб қосымшаны қолдайтын жүйені іске асыру.

Менің дипломдық проектімнің түпкі мақсаты, университет аудиторияларына немесе бөлмеге кіруге рұқсат керек кез-келген организацияның рұқсат беру жолын барынша жеңілдету және жылдамдату. Аудиторияға кіру магнитті карта арқылы немесе адам бетін тану арқылы жұмыс жасайды.

Менің дипломдық жобам университет немесе кез-келген организация қызметкерлерінің уақытын үнемдеуге және кіріп шыққан адамның нақты кім екенін анықтауға арналған. Менің дипломдық жобам веб-бет қосымшасын қолдайды.

Менің дипломдық жобамды іске асырудағы алға қойған міндетерім:

- менің жүйеме ұқсас жүйелерді зертеу;
- ұқсас желілердің жұмысын талдау;
- өз жүйемді диаграммалар салып жобалау;
- алынған нәтижелер бойынша алгоритм құрып оны іске асыру.

1 Негізгі бөлім

1.1 Жүйені талдау

Жүйе клиент-сервер технологиясымен жұмыс істейді. Жүйе қауіпсіздікті жоғарлату мақсатында құрылғандықтан веб-сайт қосымшасында негізгі артықшылықтар админ рөліне жүктеледі. Бұл веб-сайт қосымшасына кез келген адам тіркеле алмайды. Тіркелу қатаң түрде админ рөлі арқылы өтеді. Дипломдық жоба екі программалық қосымшадан тұрады, бірінші программалық қамтама Java Spring тірегінде жазылған ал екінші программалық қамтама нейрондық желілер көмегімен коптеген кітапханалар арқылы жасап шығарылған, қазіргі кезде кеңінен тарап келе жатқан тіректердің бірі болғандықтан, жоба көп уақыт бойы қолайлы басқаруға ие болады.

Қатынасты басқару жүйесі (ACC) – АБЖ объектілеріне қолжетімділікті бақылауға және қорғалатын аумақта адамдардың қозғалысына мониторинг жүргізуге мүмкіндік беретін техникалық құралдар мен ұйымдастырушылық шаралар жиынтығы. Қазіргі уақытта қол жеткізуді бақылау жүйелері объектілер үшін кешенді қауіпсіздік тапсырмаларын шешудің ең тиімді әдістерінің бірі деп танылады [16].

Қол жеткізуді бақылау жүйесіндегі қызығушылықтың үнемі өсуіне және оларды кеңінен қолдану болашағына қарап, қол жеткізуді бақылау жүйесі сәйкестендіру процесін жеңілдетеді, уақытты үнемдейді және компанияның қауіпсіздік қызметтерінің тиімділігін арттырады, бірақ сонымен бірге, ол қазіргі күнге дерлік адамдық бақылауды қажет етеді. Жүйеге тағайындалған ықтимал қауіптердің деңгейі мен міндеттері жүйедегі адамдар мен техникалық ресурстардың арасындағы оңтайлы ара-қатынасты таңдау қажеттілігін анықтайды. Қатынасты басқару жүйесін орнату тек қана жалпы қауіпсіздік деңгейін арттырып қана қоймай, оны қамтамасыз етудің өзіндік құнын азайтады, себебі қол жеткізуді бақылау жүйелері техникалық жағынан қызмет көрсету кезінде көптеген қызметкерлерді қажет етпейді, олар электр энергиясын тұтынуда үнемді болады. Сондықтан менің жасап жатқан жүйемнің түпкі мақсаты қызметкерлердің уақытын үнемдеу, артық адам күшін қажет етпеу және қауіпсіздікті жоғарлату.

1.2 Ұқсас жобаларды шолу

Бұл жазба уақытында қазақстан нарығы ұсынылды кішігірім кеңселерге кіруді бақылаудың көптеген жүйелерінің кейбіреулері:

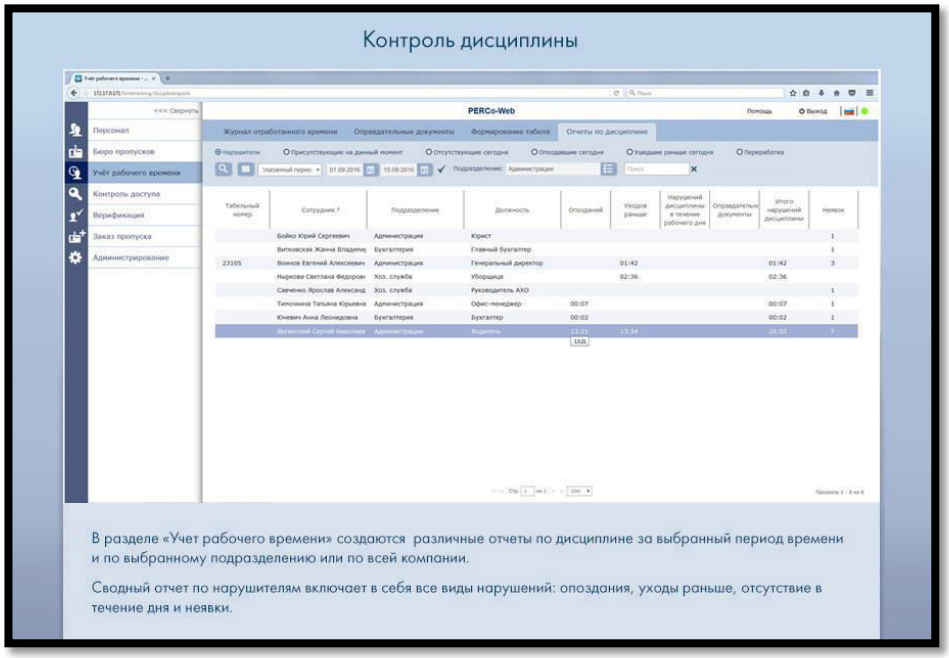
- «PERCo-Web»;
- «ParsecNET офисі»;
- «BOLID ORION PRO»;

– «Elsys Bastion 2».

PERCo – Web жүйесінде рұқсат беретін веб интерфейсі бар режимдегі әлемнің кез келген жерінен офис келушілер туралы барлық ақпаратты қадағалаңыз нақты уақыт. Жүйелік интерфейстің құрамында 7 секция бар көптеген түймелер мен қондырғылар, бұл оны шамадан тыс жүктейді және қиынға соғады түсіну [17].

Қолданыстағы тариф бойынша бағдарламалық қамтамасыз ету лицензиясының құны шамамен 228 мың теңгені құрайды. Сынақ мерзімі 60 күнге дейін бар. Бұл жүйе Windows Server ОЖ-де серверді қажет етпейді 2008 жылы R2 Firebird 2.0 DBMS орнатылған. Жабдықтардың жалпы құны оның ішінде екі контроллер, төрт карточкалық оқу құралы және екеуі бар электромеханикалық құлып, ағымдағы бағытта шамамен 444 мың теңге. Сондай-ақ, орнатудың, кабельдердің және қуат көздерінің құнын, өзара кабельдер [17].

Жүйенің негізгі артықшылықтары – бұл веб-интерфейстің болуы тегін ДҚБЖ қолдану. Кемшіліктердің ішінен шығындар, жабық бағдарлама коды, қолайсыз интерфейс дизайны, ескірген ДҚБЖ, тек Windows операциялық жүйелеріне қолдау көрсету [17]. PERCo-Web жүйесінің веб бетін 1.1-суретінен карай аласыз.

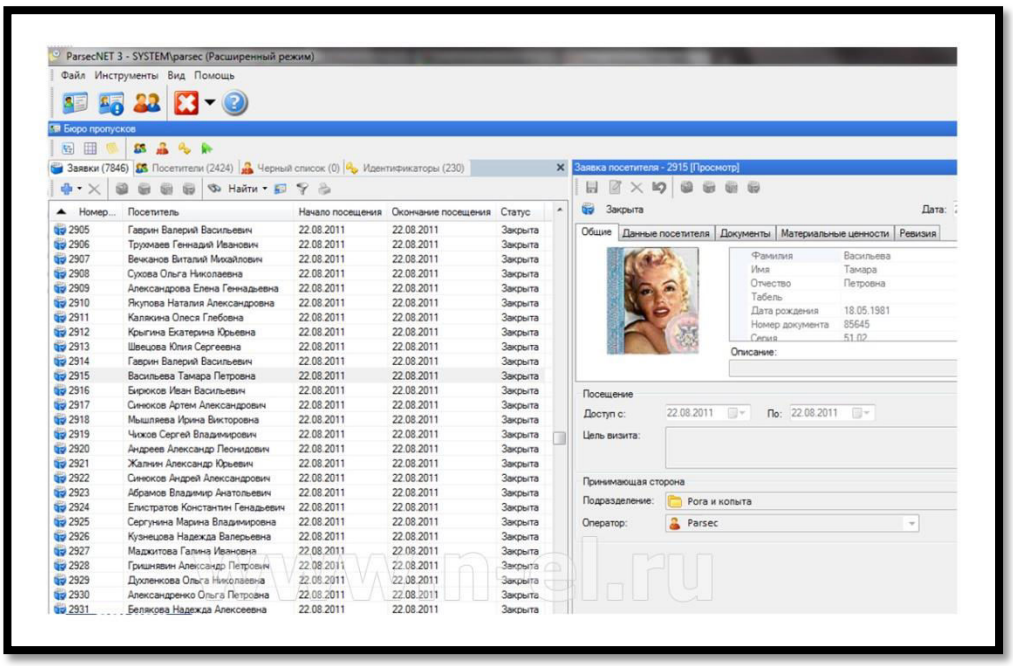


1.1-сурет – PERCo-Web жүйесінің веб беті

«ParsecNET офисі» жүйесі объектілерге арналған 16 ұпайлардан тұратын шағын масштабта. Компания кепілдік береді жоғары сапалы өнімдер, барлық оның кеңейтілген кепілдік береді сондай-ақ ұйымның қажеттіліктері үшін жабдықтар мен бағдарламалық қамтамасыз етуді еркін таңдау. Шектеулі функционалдығы бар тегін бағдарламалық жасақтама нұсқасы қол жетімді [18].

Жүйенің жұмыс істеуі үшін сізге Windows Server ОЖ-де сервер қажет. Нұсқа 2008 R2 және Microsoft SQL Server DBMS орнатылған, нұсқасы төмен емес 2008 R2. Сондай-ақ, жүйені басқару үшін, жұмыс станцияларын жабдықтау қажет операторлар. Жүйеге арналған ең аз жабдық жиынтығы екіден тұрады контроллерлер және жалпы құны 486 мың теңге болатын төрт карталарды оқуға арналған. Компания электромеханикалық құлыптарды сатумен және қосумен айналыспайды. Лицензиялардың жалпы құны – 816 мың теңге [18].

Жүйенің артықшылықтары жоғары сапалы өнім болып табылады және тегін қолдау Кемшіліктердің ішінен өте жоғары құны анықталуы мүмкін: бағдарлама коды, веб-интерфейстің болмауы, операциялық жүйелерге қолдау көрсету Windows отбасылық жүйелері [18]. ParsecNET жүйесінің қосымшасын 1.2-суретінен карай аласыз.



1.2-сурет – ParsecNET жүйесінің қосымшасы

«BOLID ORION PRO» жүйесі мониторинг жүйесіне қосымша және кіруді бақылау, сондай-ақ өрт дабылы, бейнебақылау, автоматты өрт сөндіру, сондай-ақ мониторинг жүйесі диспетчерлік объектілер. Қол жеткізуді басқару жүйесін пайдалану үшін сатып алу қажет жүйенің толық нұсқасы [19].

Бағдарламалық жасақтаманы кез-келген операциялық жүйеде орнатуға болады. Windows отбасылық жүйесі XP нұсқасынан бастап. ДББЖ рөлінде қолданылады Microsoft SQL Server Express 2012. Жүйенің жұмыс істеуі үшін сіз екі сатып алуыңыз керек қол жеткізу контроллері, төрт картаны оқу құрылғысы және екі электромагниттік құлыптар құны 210 мың теңге болды. Бағдарламалық жасақтаманың жалпы құны болады 210 мың теңгені құрайды [19].

Жүйе салыстырмалы түрде арзан бағаға ие болды үшінші тарап сенсорларының үлкен санын қосу мүмкіндігі. Минус туралы Жабық бағдарлама кодын таңдауға болады, ол тек жұмыс істейді Windows отбасылық жүйелері [19].

«Бастион-2» аппараттық-бағдарламалық кешені интеграциялануға мүмкіндік береді бір бейнебақылау жүйесінде, өрт және қауіпсіздік сигналдары, жүйелер периметрлік қауіпсіздік, қауіпсіздік жарықтандыру жүйелері, қол жеткізуді басқару жүйелері және олар үшін техникалық процестер басқару жүйелері. Oracle 11g DBMS ретінде пайдаланылады. Қызмет ету үшін Кіруді басқару жүйесі бағдарламалық қамтамасыз етудің серверлік нұсқасын және кіруді бақылау модулін сатып алу үшін жеткілікті. Жалпы құны 306 мың теңге болады. Екі контроллердің құны төрт Есепшілер 360 мың теңге болады. Сату және қосылу Компания электромеханикалық құлыптармен айналыспайды [20].

Жүйенің артықшылықтары веб-интерфейс арқылы кіруді қамтиды жүйені қосымша модульдермен жабдықтау мүмкіндігі. Минус туралы шығындарды, жабық коды мен қолдауды көрсете алады Windows операциялық жүйелері ғана. BOLID ORION PRO жүйесінің қосымшасын 1.3 – суретінен карай аласыз.

1.3-сурет – BOLID ORION PRO жүйесінің қосымшасы

Барлық жүйелерді талдағаннан кейін біз әзірленгенде үнемдеуге тырысамыз барлық артықшылықтар мен кемшіліктерді жою. Жүйені жетілдіру бірнеше талаптарға жауап береді [20]:

– тегін ашық бастапқы коды, ол тек қана іске қосылуы мүмкін Windows отбасының ОС, сонымен қатар Unix отбасында OS;

– веб-интерфейстің болуы;

– қазіргі заманғы еркін ДҚБЖ-мен жұмыс істеу.

1.3 Web – технологияның қазіргі жағдайы

Internet кез-келген дербес компьютерді ғаламшарда орналасқан одан басқа жұмыс орындарымен, яғни телефон байланыстарында қосылған басқа дербес компьютермен тез байланыстыратын бүкіл әлемдік желі. Дүниедегі ең алып ауқымды желі болып табылады. Телефон желісі арқылы байланыса алатын дербес компьютерлер бір-бірімен тығыз байланыста TCP/IP хаттама нұсқаулықтарыен ақпарат алмасады, бір нұсқада олардың бәрін, яғни, бір тілде болатындай «мәлімет алмасады» деп айтса болады. Бүкіл әлемдік халықаралық телефон байланысы секілді оны ешқандай мекеме басқармайды, ол ешқандай компанияның жекеменшігі емес. Сондықтан, осы Интернет байланысы көмегі арқылы электрондық поштаның көмегімен хабар алмасып (беріп), басқа дербес компьютерлердегі мәліметті көріп, алыстан телеконференцияларға қатысу жұмыстарын өткізуге мүмкіндік болады. TCP/IP – Интернет байланысына қосылған дербес компьютерлер арасындағы мәлімет алмасуды қамтамасыздандыратын ақпаратты бір құрылымға келтіру ережелері, болмаса оларды құру хаттамасы. IP – ақпаратты, оны алушының адресі бойынша көрсетілгендей шағын параметрлері бар болғандықтан, бірнеше бөліктерге немесе бөлімдерге бөлетін желі арасындағы хаттама болып табылады. TCP – мәліметті жөнелтуді баскаратын хаттама болады, ол желі бойындағы мәлімет дестелерін толықтай дұрыс жеткізуге жауапты болып табылады. Интернет байланысын пайдаланып, үйден шықпай ақ, мыңдаған қалаларды аралап мұрағаттарды, кітапханалар көріп, бүкіләлемдік ғылыми және мәдени жетістіктермен танысып, оларға өркениетті елдің қолданушысы ретінде өз пайдаңызды қосып, өзіңізді әлемнің бір кішкене бөлігі болатындай сезіну мүмкіншілігі пайда болуы мүмкін. Ғаламтор –TCP/IP хаттамаларына негізделіп желі арасындағы байланысу технологиясының бірі болып табылады. Бағдарламалық қамтама етуінің құрастырушысы және қарапайым пайдаланушысы үшін Web-технологияның мәні – бұл ең алдын заманауи, қажет технология екенін бұрыннан анық. Web-технологиялардың қызығушылығы – қолданушы мен дербес компьютер арасында жүретін ақпаратты тасымалдау құрал есебінде көбінесе әмбебап құрал бола отырып интерфейсті анықтайды [5].

Бүкіләлемдік ғаламтордың іргетасы – HTML гипермәтіндік белгілеу тілі болып табылады. Ол құжаттың (веб-беттің) физикалық белгілеу үшін қызметін атқарады. Веб-беттегі мазмұнның бейнесін басқару үшін (CSS) стильдердің каскадты кестелері үшін арналған. Көбінесе CSS, белгілі мәтіндік Word жүйесінде қолданылатын стильдерімен өте ұқсас болып кездеседі. Ол құжаттың (веб-беттің) физикалық белгілеу үшін қызметін атқарады. Веб-беттерге қимыл (жылжымалы мәзірі, анимация) еннгізу үшін JavaScript бағдарламалау тілі қолданылады.

1.4 Дипломдық жобада қолданылған технологиялар

Менің дипломдық жобамда қолданылатын технологиялар:

- Java объектіге бағытталған бағдарламалау тілі;
- MySQL деректер қорымен басқару жүйесі;
- Apache веб-сервері;
- CSS тіректері;
- HTML еренмәтін белгілеу тілі;
- Spring Java тілінде жазылған тірек;
- Bootstrap веб-сайттың беткі көрінісін әшекелеуге арналған тірек;
- Python объектіге бағытталған бағдарламалау тілі;
- Tensorflow машина оқытуға арналған ашық кітапхана;
- Keras ашық нейрожүйелік кітапхана;
- PostgreSQL деректер қорымен басқару жүйесі;
- OpenCV ашық компьютерлік көзқарас кітапханасы;
- Flask веб-тірегі.

2 Жобалау бөлімі

2.1 Унифицирленген моделдеу тілі

UML – бағдарламалық дамытуға графикалық объект модельдеу сипаттау үшін тілді, бизнес-процесстерді модельдеу, жүйелер инженерлік және ұйымдастырушылық құрылымдарды салыстыру [21].

2.2 Rational Rose құралы

Rational Rose – визуалды модельдеу және кәсіпорын деңгейіндегі бағдарламалық жасақтама қосымшаларын компонентті құрастыру үшін арналған объектілі-бағдарланған бірыңғай үлгілеу тілі (UML) бағдарламалық жасақтама құралы. Rational Rose пайдаланады, кейс элементтерін (oval), нысандарды тіктөртбұрыштар) және хабарлар / қатынастар (көрсеткілер) апарып тастау таңбаларын пайдаланып жүйелі диаграммада болады. Театр режиссері ойын ойнатуға кедергі келтіреді, бағдарламалық жасақтаманың құрастырушысы актерлермен сабақтарды оқшаулау арқылы (кесте фигуралары) оқшаулау арқылы қосымшаның негізін құруға (модельдеуге) арналған Rational Rose пайдаланады, кейс элементтерін (oval), нысандарды тіктөртбұрыштар) және хабарлар / қатынастар (көрсеткілер) апарып тастау таңбаларын пайдаланып жүйелі диаграммада болады. Rational Rose схемасын жасайды, ол құрылады, содан кейін конструктор C ++, Visual Basic, Java, Oracle8, Corba немесе деректерді анықтау тілі [22].

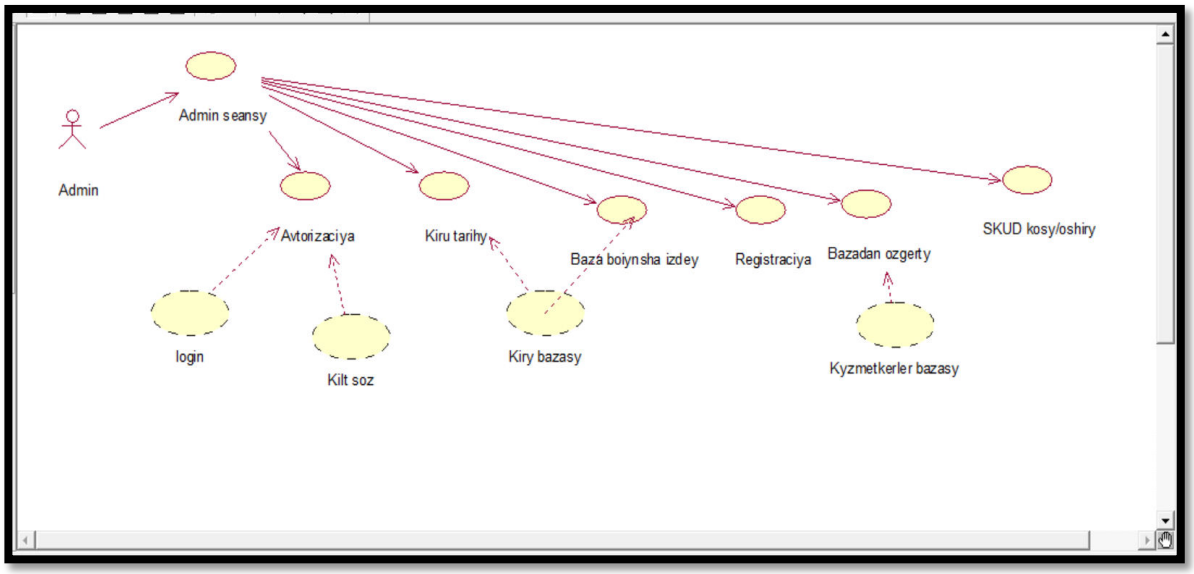
2.1-сурет – Rational Rose құралы

2.3 Прецеденттер диаграммасын құру

Жүйені құру үшін арналған зат обласын толықтай білу қажет. Сонымен қоса жүйенің қалай жұмыс жасайтыны туралы ақпараттар болу қажет. Диаграмма жүйесі ретінде орындайтын операция тізімдерін құру үшін тағайындалған. Ол берілген методтағы жүйе объектісі болып анықталған тізім жүйесін табжылмай орындайды. Осындай түрде жасалатын жүйенің функция құрылымы құрылады, сонымен қоса дүйемен байланысқан объектілердің барлығының іс-әрекетінің жоспарлары жазылады. Прецедент диаграммасы процесін модельдеу үшін пайдаланушы іс-әрекетін толықтай көрсетуіне, сонымен қоса толық реакциясына да бағытталған. Use Case диаграммасы 2.2, 2.3-суреттерде көрсетілген.

Қолдану типтері (Use Case) жүйе және активті субьект арасында болатын диалогты модельдеуге мүмкіндік ұсынады, сонымен қоса функцияны соңына қарай бейнелейді. Жүйені қолдану типтер жиыны оны пайдаланудың көптеген амалдар назарына бағытталған. Қолдану типтері – транзакция жүйесі бойынша орындалатын тізбектілік, осы арқылы табылған активті субьект, яғни, керекті нәтижені оңай алуға болады [21].

Қолдану диаграммасы – бұл активті денелердің көптеген графикалық көрінісі оны қолданудың, ол болмаса басқа нұсқалардың амалдарымен өзара байланысады. Жүйені жоспарлағанда құрылымның кілттік функциясын, сонымыен қатар көптеген қолданушылар ұсынатын негізгі диаграмма (Main Use Case Diagram) осылай конструкцияланады [21].



2.2-сурет – Администратор прецеденттер диаграммасы

Use Case – бұл белгілі актерге (Actor) арналған, белгіленген бір нәтиже бере алатындай, бағдарлама жүйесімен орындалатын іс-қимылдар тізбегінің

сипаттамасы. Ал, прецедент негізінен моделдегі берілген мәндердің тәртіптік структурасын жасайды, сонымен қатар прецеденттер кооперацияның қатысуымен таратылады [21].

Актер (Actor) – бұл берідген жүйедегі негізгі деген элементтермен байланыста болатын негізгі обьектінің рөлі болып саналады. Актер мен қолданушының салыстырмалы түрде айырмашылығы келесідей: қолданушы жүйені толықтай пайдаланатын шынайы обьект. Ол бірнеше рөлде ойнауы мүмкін болады, соған байланысты ол бір емес, бірнеше актер болуы мүмкін [21].

2.4 Тізбек диаграммасын құру

Тізбек диаграммасы уақыт бойынша берілген объектілердің өзара ұйымдасу операцияларының жасалу ретін безендіреді және жоспарлармен қарастырылған мүмкіндіктердің орындалу процесіндегі өзара алмасатын объектілердің хабарлама тізбектерінің реттелген класстар және обьектілер бейнеленеді [21].

Тізбектелген диаграммалар жалпы түрде Logical View пакетіндегі көзделген қолдану нұсқаларының жүзеге асырылуымен бірге белгіленеді. Тізбек диаграмма негізгі төрт элементтен тұрады:

– прецеденттегі артындағы мәтінінің іс-қимылы. Ол жоғарыдан төменге жазылады, және сол жақтан басталады. Сол терезеде іс-қимыл сипатталуы болып, орындалатын жұмыс уақытындағы белгілі ақпараттар қызмет етеді;

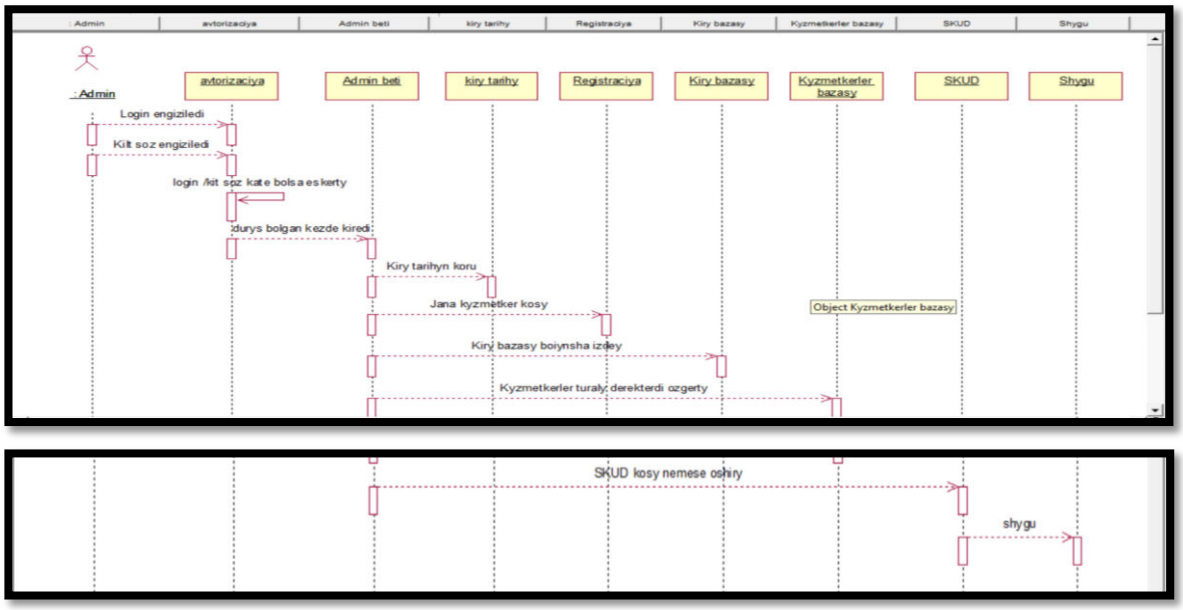
– объектілер "объект-класс" кейпінде аты болмаса объект данасының нөмірі және класс объектісінің атауы жазылады;

– хабарландыру, бір бағытта көрсетілген белгілі бір объектіден келесі обьектіге бағытталған іс-қимыл туралы мәлімет хатынан тұрады. Берілген бір уақытта орындалуы мүмкін, және осы іс-әрекеттегі берілген жүйенің қайтарымды реакциясы болуы мүмкін болып келеді;

– әдістері (операциялар). Тікбұрыш түрінде болады. Олар үздік сызықпен бейнеленген. Яғни, берілген әдістерге кіретін сол объектілер болып табылады. Тік бұрыштың ұзындығы артынан қуушыда басқару нүутесін көрсетуде қолданса болады: Тікбұрыш аяқталатын немесе бітетін әдіс толықтай нүктесіне дейін басқарумен игеріледі. Бұл, берілген үшбұрыштар циклге арналған объекті түзуі деп саналады.

UML талаптарына байланысты объектінің тізбектелген диаграммасында тіктөртбұрыш кейпінде көрсетіледі. Объектіні 3 түрлі есіммен атауға болады: тек оның атауын көрсетіп кету қажет, объект пен класстың атауын беру қажет, не класстың атауымен шектелу керек [21].

Тізбектер диаграммасы 2.4-суретте көрсетілген.



2.3-сурет – Тізбек диаграммасы

Тізбектелген диаграммада активті объектілермен өзара бірігу мәліметтер жүйесінің көрсетілуін рұқсат ететін шекутеулер класы қосылады (пайдаланушылар мен және басқа жүйелермен).

2.5 Кооперация диаграммасын құру

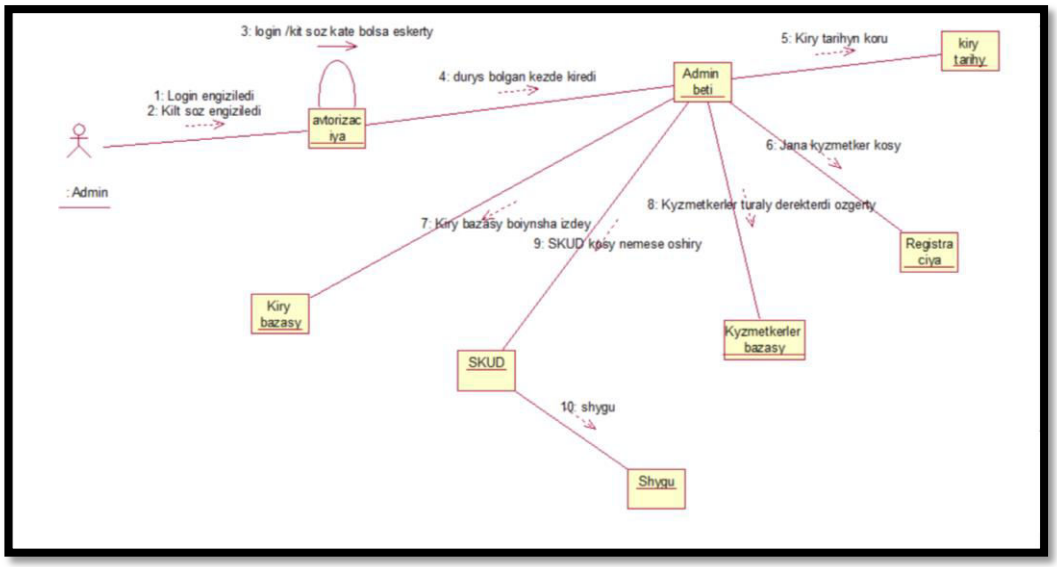
Кооперация диаграммасы дегеніміз бұл хаттарды жіберетін және қабылдайтын субъектілерді құрылымдық жақтан жан-жағынан ұйымдасуын айқындайтын әрекеттесу диаграммасы. Кооперация диаграммалары жүйе жұмысының барысында субъектілердің өзара әрекеттесуін баяндайды. Мұндай диаграммалар жүйе жоспарының сценарийлерін бейнелейді. Объект атыуның асты сызылады және әрдайым ұсынылады, ал қасиеттері таңдалып көрсетіледі [21].

Кооператив диаграммалар белгілі оқиғаларда пайдалы субъектілерде істелген өзгерістердің зардаптары керек бағалауда және қандай басқа субъектілерге әсер етеді.

Кооперативтік диаграммалардың бірін жатқызуға болады, олар негізінен оқиға детализациясы үшін пайдаланылады, олар уақиғалар ортасын және олардың арасындағы қарым-қатынасты анықтайды, қосымша пайдаланушыларды анықтайды, олардың жалпы, оған қоса мінездемелік анықтамаларын береді – ол дегеніміз, соңғы «Класстар диаграммасын» – сызу үшін қажетті деректердің барлығын жүктеуге мүмкіндік береді [21].

Кооперация сызбалары жүйе жұмысы барысында субъектілердің өзара қарым-қатынасын бейнелейді. Мұндай сызбалар жүйе тәртібінің жоспарларын

модельдейді. Кооперация диаграммасы 2.5-суретте көрсетілген.



2.4-сурет – Кооперация диаграммасы

2.6 Күй диаграммасын құру

Күй (State) – бұл кейбір нышандарды орындау кезінде белгіленген әрекет болмаса басқа оқиғаның келуін күтуде мекендеуді жүзеге асыратын жиынның шарты. Объектінің күйі класы бірнеше немесе бір деректермен бейнеленеді [21].

Күй диаграммасы – нысанның бір қалпынан басқасына ауысуын айыратын, және іс-әрекеттері қалпыларын алмастырумен ескертілінген, оқиғалардың, болмаса хабарлаулардың объекттік жағдайларының графикалық түрдегі ұсынысы [21].

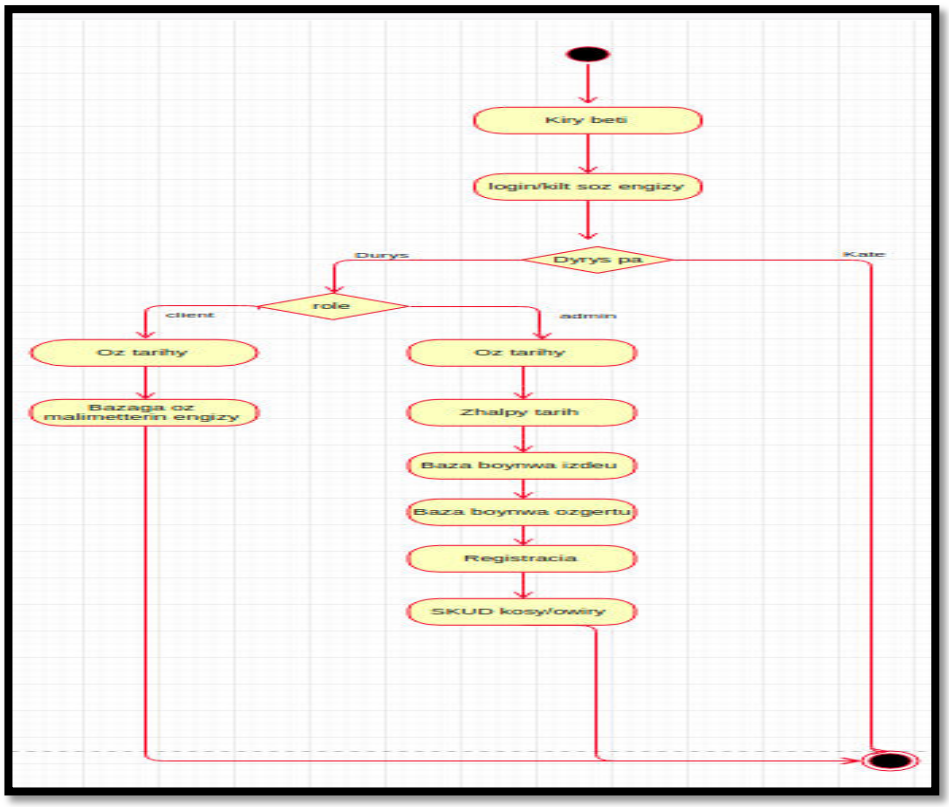
Объектінің өз қалпын уақыт бойында қай бағытта өзгертетінін түсіну үшін жағдайдың өзгеруінің спецификациясы бар. Субъектінің жағдайы оның атрибуттарының, және қатынас атрибуттарының мағынасымен сипатталады.

Күй спецификациясы класс атрибутын сипаттайды. Тәртіп спецификациясы класс оталарын анықтайды,олардың кейбірі нысан күйін басқаға өзгерте алады.

Объектілер жағдайларын моделдеу күйлер диаграммасының арқасында жүзеге асады. Тәртіп спецификациясы класс оталарын анықтайды,олардың кейбірі нысан күйін басқаға өзгерте алады.Күйлер сызбасы бұл күйлер мен өзгерістер графы. Күйлер моделі системаға маңызды кластар үшін сызылады [21].

Обьектілер белгілі бір нүктеден шығып, соңғы нүктеге жеткенше барлық нүктелерден өтеду, және соңғы нүктеге жетеді.

Күй диаграммасы 2.6-суретте көрсетілген.

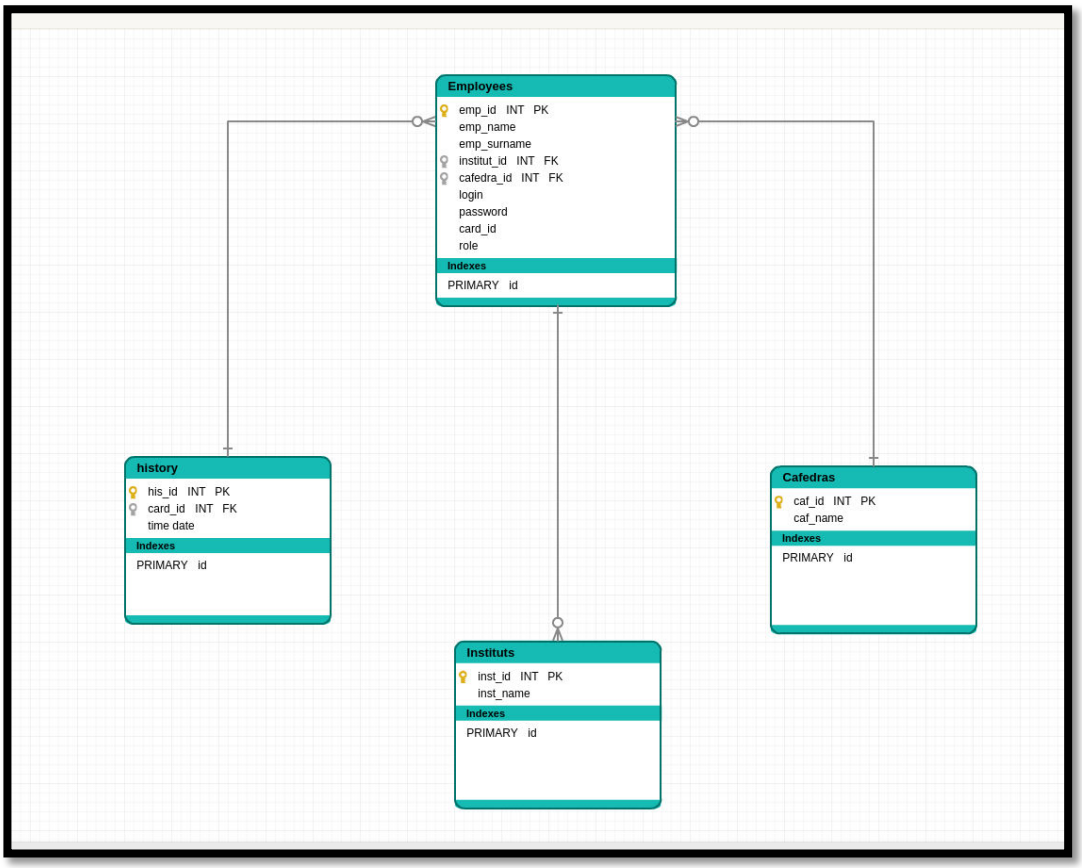


2.5-сурет – Күй диаграммасы

2.7 ER диаграмманы құру

ERD, ER диаграммасы немесе ER үлгісі деп те аталатын элемент қатынасы схемасы дерекқордың дизайнын пайдалану үшін құрылымдық схеманың түрі болып табылады. ERD екі маңызды ақпаратты бейнелейтін әр түрлі рәміздер мен қосқыштарды қамтиды: жүйелік ауқымдағы негізгі субъектілер және осы ұйымдар арасындағы өзара қарым-қатынас [23].

Проектінің ER диаграммасы 2.6-суретте көрсетілген.



2.6-сурет – Проектінің ER диаграммасы

3 Қолданылған бағдарламалық қамтамалар

3.1 MySQL технологиясы

MySQL – бұл қазіргі замандағы ең көп қолданылатын деректер қорының бірі [11]. MySQL технологиясын мен өз жүйемнің деректер базасы ретінде алдым. Базамен байланысатын барлық операциялар ос технология арқылы іске асады. Нақты айтатын болсақ магнитті карта оқылған жағдайда осы орындалған әрекетті базаға енгізу үшін және базадан қызметкерлердің деректерін енгізу үшін немесе базадан алу үшін қолданамын.

3.2 HTML тілі

HTML – веб-сайтты жасауға мүмкіндік беретін компьютер тілі. Бұл веб-сайттарды интернетке қосылған кез келген адам көре алады. Үйрену өте қарапайым, негіздері көптеген адамдар үшін бір отырыста қол жетімді және сіз жасауға мүмкіндік беретін өте қуатты [14].

HTML тілін барлық веб беттін қаңқасын жасауда қолдандым.

3.3 Apache2 веб сервері

Apache HTTP-сервері (ағыл.тілінен аударғанда a patchy server) – еркін Web-сервер [15]. Менің жасап шыққан жүйемді тестілеу үшін менің веб бетім локалды хостингта болу керек болды. Сол үшін мен Apache2 веб серверін қолдандым.

3.4 Bootstrap кітапханасы

Bootstrap (сондай-ақ Twitter Bootstrap) – веб-сайттарды және веб бағдарламаларды жасаудың құралдарының еркін жиынтығы. HTML және CSS рәсімдеуге қойылатын жобалау үлгілерін, веб нысандарының түймелерін, жапсырмаларды, навигациялық бірлік, соның ішінде басқа веб-интерфейс компоненттерді және JavaScript-кеңейтуді қамтиды [13].

Менің веб қосымшамды кез келген құрылғыдан ашқан жағдайда өз стилін жоғалтпай бір қалыпта болуы керек. Бұл жағдайда маған Bootstrap кітапханасы көмекке келді. Bootstrap кітапханасында бұл жағдайдың барлығы ескерілген, сондықтан осы кітапхананы қолдандым.

3.5 Java бағдарламалау тілі

Java жалпы мақсаттағы компьютерлік бағдарламалау тілі, класс негізделген, объектілі-бағытталған, және арнайы бірнеше іске асыру ретінде болуы арналған тәуелділіктерді мүмкіндігінше [10]. Жүйенің негізін java тіліндегі spring фреймворкы арқылы жазып шықтым. Spring фреймворкы қолдануға өте тиімді және қауіпсіздік жағынан өте сенімді.

3.6 Python бағдарламалау тілі

Python – интерпретацияланған, жоғары деңгейлі, жалпы мақсаттағы бағдарламалау тілі. Құрастырған Guido van Rossum және 1991 жылы алғаш рет шығарылған, Python жобалық философиясы бар, ол кодты оқуға назар аударады, атап айтқанда айтарлықтай бос кеңістікті қолданады. Бұл кішігірім және үлкен ауқымда нақты бағдарламалауға мүмкіндік беретін конструкцияларды қамтамасыз етеді [12]. Python программалау тілі өзін нейрондық желілер аумағында өте тиімді тіл ретінде көрсетті. Мен бұл тілді бетті тану арқылы жұмыс жасайтын программалық қамтаманың негізінде қолдандым. Адам бетін тану үшін қолданылатын моделді оқыту синтаксисін осы тілде жаздым.

3.7 Tensorflow және Keras кітапханасы

TensorFlow еркін және ашық-көзі бағдарламалық қамтамасыз ететін кітапхана үшін ағымы және дифференциалданатын міндеттер ауқымы арқылы бағдарламалау. Бұл символикалық математикалық кітапхана, сондай-ақ, нейрондық желілер сияқты машина оқуға арналған қосымшаларда қолданылады. Бұл зерттеулер және өндіру үшін де пайдаланылады Google. Бұл стандартты күту болып табылады индустрияда машина оқытуда жұмыс істеу мақсаты болса Тензор Флоудағы тәжірибесі болуы керек [24].

Keras – бұл Python-да жазылған ашық көзді нейрондық-желілік кітапхана. Ол TensorFlow, Microsoft Cognitive Toolkit, Theano немесе PlaidML үстінде жұмыс істей алады. Терең нейрондық желілермен жылдам эксперимент жасау үшін әзірленген, ол пайдаланушыға ыңғайлы, модульдік және кеңейтілуге бағытталған [25].

Бұл кітапханаларды мен бет тану арқылы жұмыс жасайтын программалық қамтаманы нейрондық желілер көмегімен оқытқан кезде осы кітапханаларды қолдандым.

3.8 OpenCV ашық компьютерлік көзқарас кітапханасы

OpenCV ашық көзден компьютерлік көзқарас және компьютерді оқу бағдарламалық жасақтамасы болып табылады. OpenCV компьютерді көру бағдарламалары үшін ортақ инфрақұрылымды қамтамасыз ету және коммерциялық өнімдерде машина қабылдауын жеделдету үшін салынды. BSD-лицензияланған өнім бола отырып, OpenCV бизнес үшін кодты қолдануға және өзгертуге мүмкіндік береді [26].

OpenCV кітапханасын адам бетін тану арқылы жұмыс жасайтын программалық қамтамада видеокамерадан ағынды алу үшін, оны өңдеу үшін алынған фотосуреттерді өңдеу үшін қолдандым.

3.9 Flask веб-тірегі

Flask – құралдар жиынтығын пайдаланып Python бағдарламалау тілінде веб-қосымшаларды жасаудың негізі, сондай-ақ Jinja2 үлгілік қозғалтқышы. Ол микрофремалар деп аталатын санатқа жатады, ең қарапайым мүмкіндіктерді әдейі қамтамасыз ететін ең аз веб-бағдарлама шеңберлері [15].

Бұл фрейворкты адам бетін тану арқылы жұмыс жасайтын программалық қамтаманы веб қосымша ретінде шығару үшін қолдандым.

3.10 Программалық қамтаманы іске асыру

Менің дипломдық жобам екі жүйеден тұрады. Бірінші жүйе Java Spring тірегінде жазылған магнитті карта оқу арқылы жұмыс жасайды, ал екінші жүйе нейрондық желілер арқылы адам бетін тану арқылы жұмыс жасайды. Екі жүйеде Веб серверде жұмыс жасайды және веб-сайт қосымшасын қолдайды.

Веб-сайттың қызметі – қызметкерге ыңғайлап, жылдамдатып, жеңілдетіп керекті жазбаларды толықтырып жүктеуге ұсыну.

3.11.1Қолданушы интерфейсі

Бұл проектіні жасаудағы түпкі мақсаты желіде орналастыру болып табылады. Веб-бетті қарауға кез-келген браузер жарайды, Opera, Explorer, Mozilla немесе Google Chrome. Веб-бет тоқтамай жұмыс жасап тұру үшін веб-бет толықтай веб-серверде орналасып, желіге хостинг арқылы таратылыу қажет.

Веб-сервер ретінде Apache 2.0 веб сервері таңданып алынды.

Веб-бетке кірген бойда кіру бетіне түсесіз. Веб-беттің кіру беті 3.1-суретте көрсетілген.

3.1-сурет – Веб-беттің кіру беті

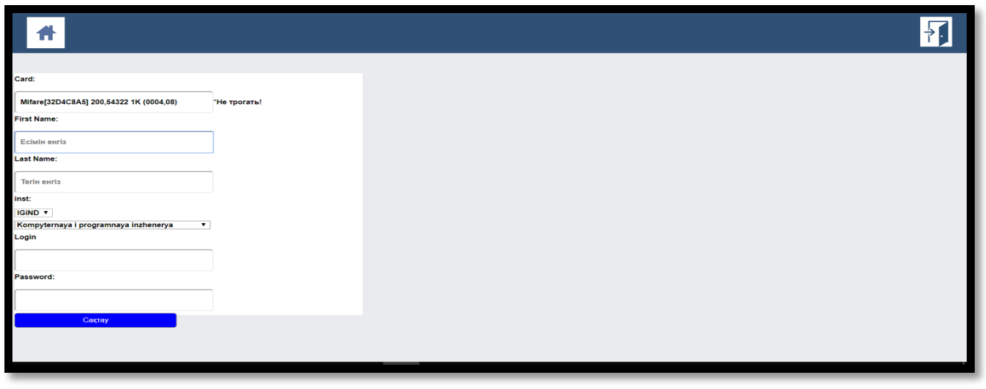
Веб-бет Boostrap тірегін пайдаланғандықтан қандай да құрылғымен ашса да, ол адам көзіне жағымды болып көрінеді.

Ұялы телефоннан, планшеттен немесе дербес компьютермен ашса да веб-беттің ешқандай элементі назардан жоғалып кетпейді.

Жоғарыда тұрған мәзір элементтері Bootstrap тірегіне сай бір жылжымалы блок сияқты бір элементтің астына жасырынып тұрады. Сол батырманы басқанда, мәзір элементтері астыға жылжып, барлығы көрінеді. Кез-келген құрылғыдан ашқанда адаптивті дизайн өзінің жұмысын жасап, сол экранға бейімдеп береді.

Мысалы, планшетпен веб-бетке кірсе адаптивті емес тірекпен жасағандықтан веб-бет экран бетіне сыймай тұрады.

Веб-беттің «Тіркелу» атты бетіне өтіп, керек ұяшықтарды толтырып тіркелсеңіз болады. Тіркелу қызметі тек админ ролінде ғана болады. Егер де, алдын ала тіркеліп қойған болсаңыз, «Кіру» атты бетке өтіп қолданушы аты мен кілтсөзіңізді жазып веб-бетке кірсеңіз болады. Веб-беттің тіркелу парақшасы 3.2-суретте көрсетілген.



3.2-сурет – Веб-беттің тіркелу парақшасы

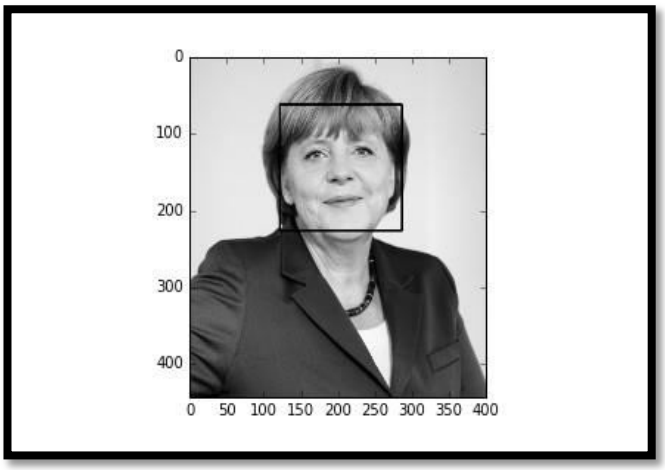
Веб-беттің «Басты» парақшасы 3.3-суретте көрсетілген. Ол жерде колданушының тарихы, тіркеу, деректер қорымен іздеу, деректер қорынан өзгерту, программалық қамтаманы іске қосу ұяшықтары бар.

3.3-сурет – Веб-беттің «Басты»парақшасы

3.11.2 Адам бетін тану арқылы жұмыс жасайтын программалық қамтаманы іске асыру

Бет әлпетті тану оңай айтылып, естілгеніне қарамастан, оны іс жүзіне асыру өте үлкен есептеу методтарын талап етеді. Бет әлпетті тану дегеніміз – алдымен бетті тауып алу содан соң оған ұқсас адамды базадан алып шығу.

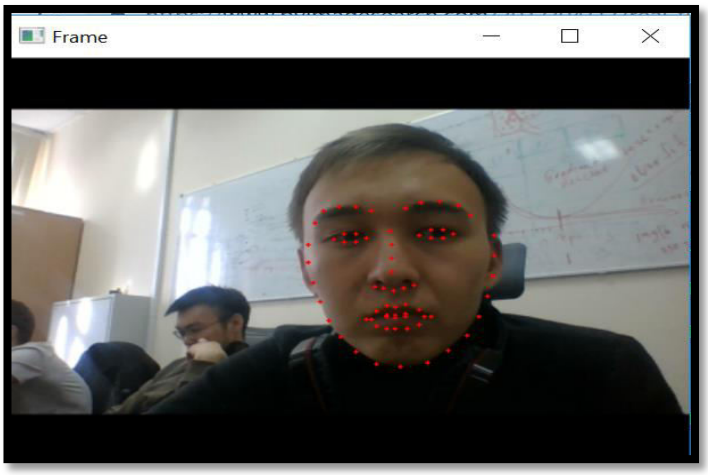
Жалпы айтқанда, бетті бақылау алгоритмі екі модульге бөлінеді. Алғашқы модуль – бұл бет әлпетті анықтау, ал екіншісі беттің өзін бақылау. Суреттегі бетті табу үшін, Haar негізінде алгоритм және dlib нейрондық желілер кітапханасының алгоритмдері қолданылады. Нақты уақыттағы нәтижелер ұсынылған алгоритм бет ерекшеліктері нүктелерін нақты шығара алады. Алгоритм нақты уақыттағы камераның кірісіне және нақты уақыттағы қоршаған ортаның жағдайына қолданылуы тиімді болып есептелінеді [6]. Haar негізіндегі алгоритм реализациясын 3.4-суреттен көре аласыздар.



3.4-сурет – Haar негізіндегі алгоритм реализациясы

Dlib – нақты әлемдік проблемаларды шешу үшін C ++-де күрделі бағдарламалық жасақтама жасау үшін машина алгоритмдерін және құралдарын қамтитын C ++ заманауи құралдар жиынтығы [1].

Ал dlib нейрондық желі кітапханасының адам бетін табу алгоритмінің реализациясын 3.5 суреттен көре аласыздар.

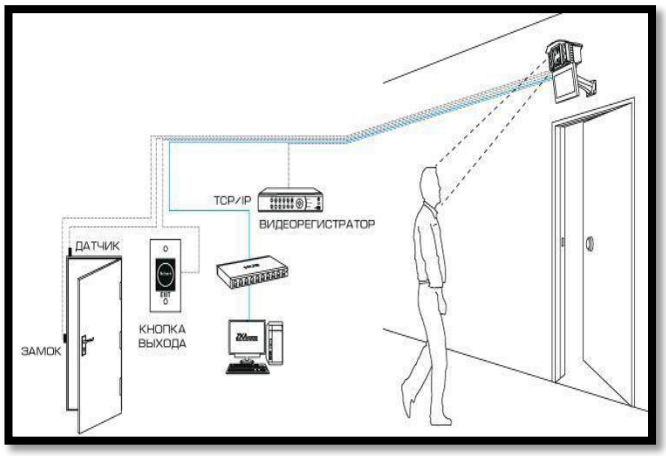


3.5-сурет – Dlib кітапханасының бет табу алгоритм реализациясы

Мен оз жобамда VGGFace2 моделінде алдын ала оқытылған модельді алып қолдандым. Бұл модель сурет қабылдайды және көптеген есептеулер нәтижсінде берілген суреттің векторын қайтарып береді.

Менің жобамның құрылымына келетін болсақ, бейнекамераның алдында адам беті пайда болған жағдайда адам беті анықталып алдын ала оқытылған модельге жіберіліп сол суреттің векторын аламын. Осылай 5 кадр жасалады және өңделуге жіберіледі. Өңдеу барысында камера алдындағы адамның суретінің векторлары базадағы кіруге рұқсат берілген адамдардың суреттерінің векторларымен салыстырылып арасындағы бұрыштың косинусы есептелінеді.

Жүйе құрылымын 3.6 – суреттен көре аласыздар.

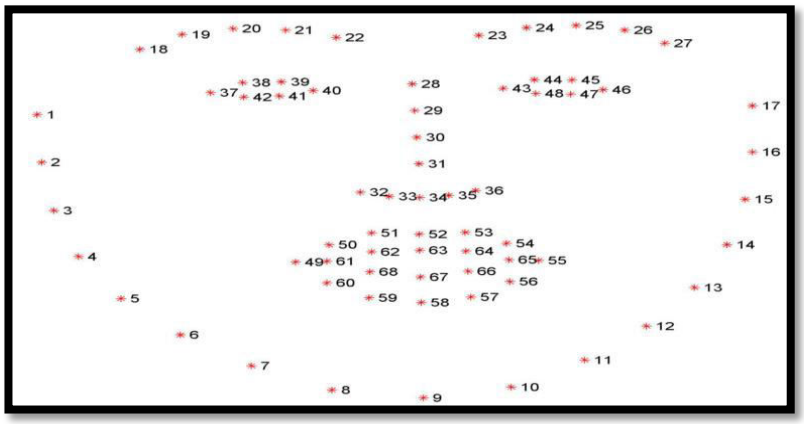


3.6-сурет – Жүйе құрылымы

Нәтижесінде екі вектор арасындағы бұрыштың косинусы 0 градусқа неғұрлым жақын болса, нақты кім екені анықталады. Егер адам беті базадағы адамдармен сәйкестігі 85%-тен жоғары болса, онда жүйе бейнекамера алдындағы адамды сол адам деп есептейді және есік ашуға рұқсат береді.

Біз камера алдындағы адам шынымен адам екеніне көз жеткізу мақсатында бірнеше алгоритмдерді қолдана аламыз. Олар адамның көзінің ашып жұмылуын есептеп анықтау және адамның күлгенін тану алгоритмдері.

Адамның көзінің ашып жұмылуын немесе күлгенін анықтау үшін біз dlib нейрондық желілер кітапханасының бет ориентирлерінің индекстерін қолданамыз. Бұл dlib нейрондық желілер кітапханасының ішінде орналасқан детектор бізге адамның бет ориентирлерінің 68 индексін береді. Бұл 68 нүктелік салыстырулар iBUG 300-W [7] деректер жиынтығындағы белгісі бойынша пішінді болжаушылар арқылы алынған. Ол индекстерді 3.7-суреттен көре аласыздар.

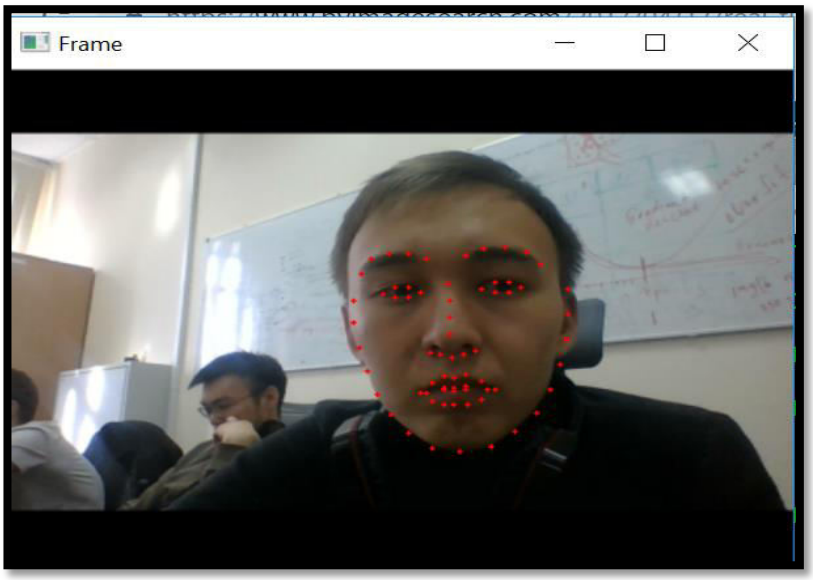


3.7-сурет – IBUG 300-W деректер жиынтығынан 68 жеке координаттар нүктелерінің әрқайсысын визуализациялау

Кескінді қарап шыққаннан кейін, алдыңғы бөліктерге Python қарапайым индекстеу арқылы кесіп ала аламыз (Python көмегімен нөлдік индекстеуді ескере отырып, жоғарыда көрсетілген сурет бір индекстелген):

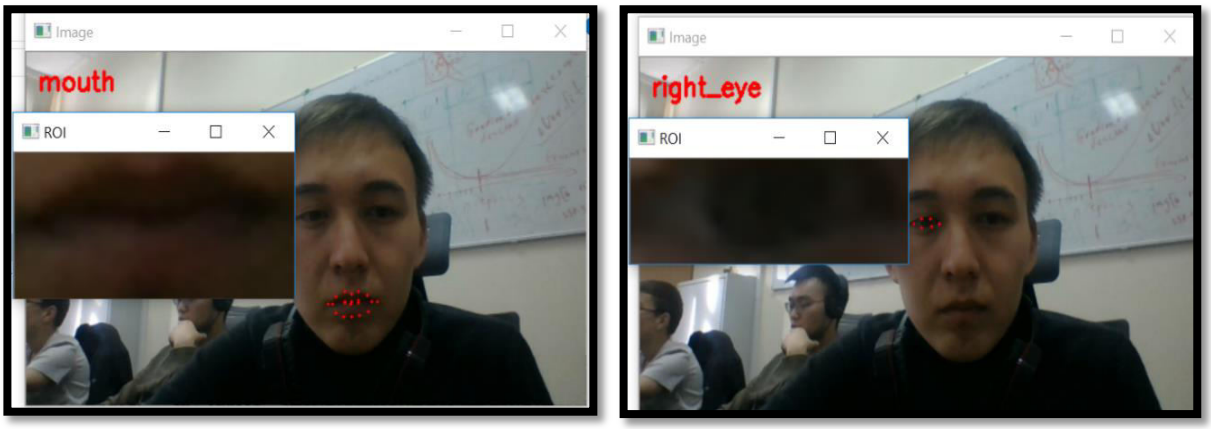
– аузына нүктеден өтуге болады [48, 68];
– оң жақ қасы арқылы нүктелер [17, 22];
– сол жақ қастар арқылы нүктелер [22, 27];
– оң нүктелерді көзге [36, 42];
– сол жақтан нүктелер арқылы өтіңіз [42, 48];
– ноза арқылы нүктелер [27, 35];
– жуын нүктелер арқылы [0, 17].

Бет бетінің маңызды жерлерін, соның ішінде көздің, қастың, мұрынның, құлақтың және ауыздың орналасуын анықтау үшін қолдануға болады, 68 жеке координаттар нүктелерінің әрқайсысын визуализациялауын 3.8-суретте көрсетілген.



3.8-сурет – 68 жеке координаттар нүктелерінің әрқайсысын визуализациялау.

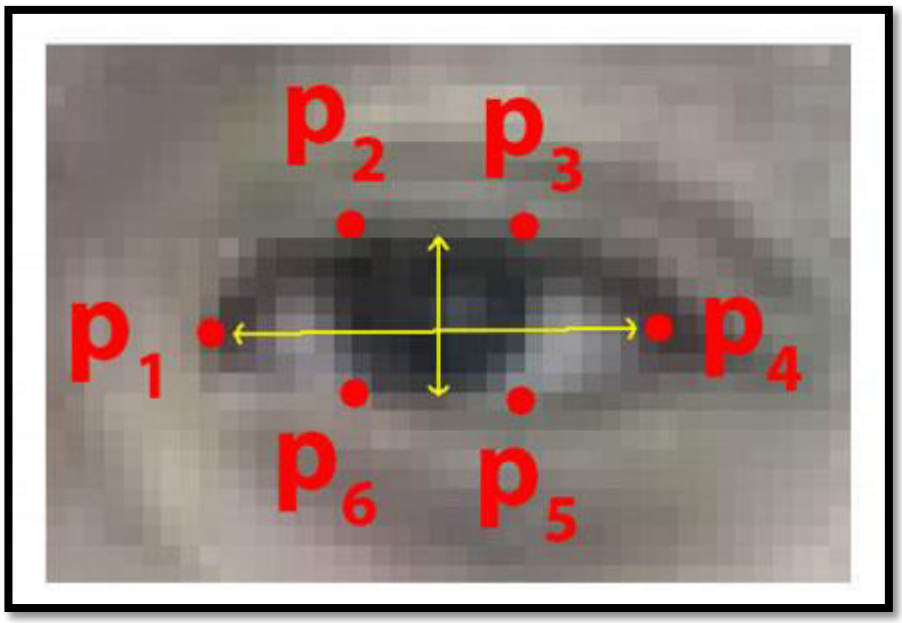
Бұл сондай-ақ беттің белгілі бір бөліктерінің көрсеткіштерін біле отырып, белгілі бір құрылымдарды шығара алатындығын білдіреді, белгілі бір құрылымдарды шығаруды 3.9 – суретте бейнеленген.



3.9-сурет – Белгілі координаттар нүктелерінің әрқайсысын визуализациялау.

Жыпықтауды анықтау тұрғысынан біз тек екі бет құрылымын – көзді қызықтырамыз.

Әр көзді көздің сол жақ бұрышынан бастаған 6 (x, y) координаттары бейнелейді (егер сіз адамға қарағандайсыз) 3.10 – суретте бейнеленген, содан соң облыстың қалған бөлігіне сағат тілімен жұмыс жасай аласыз.



3.10-сурет – Көзбен байланысты 6 бет белгісі

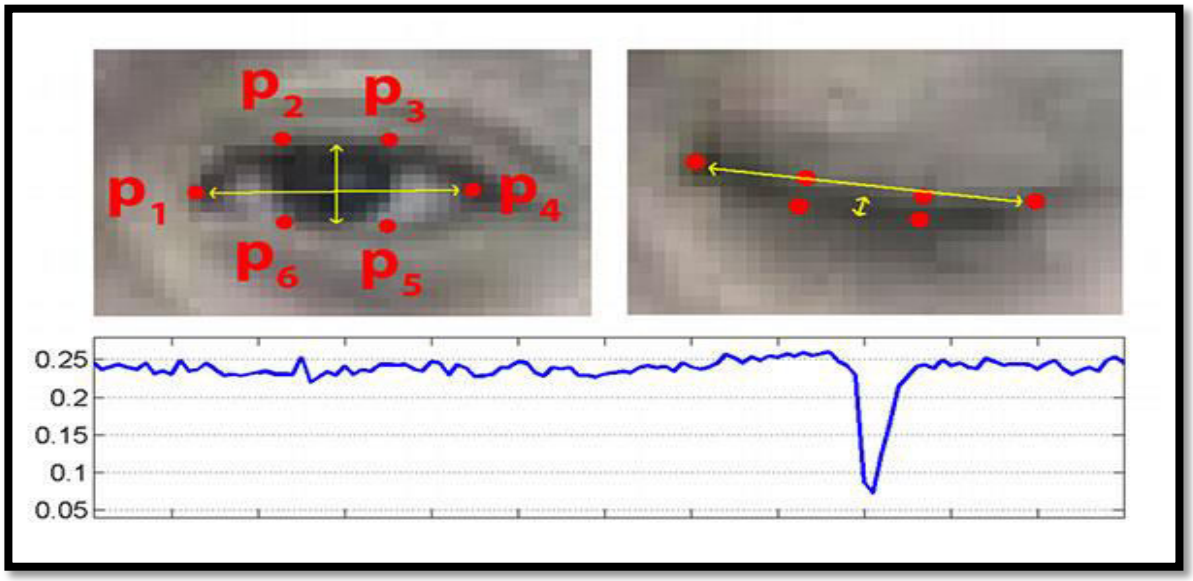
Бұл координаттардың ені мен биіктігі арасында байланыс бар. Сукупова мен Чехтың [8] жұмысына негізделген, өздерінің 2016 бапта, Face Guides арқылы нақты уақыттағы жыпықтауды анықтаған кезде, біз көздің арақатынасы (EAR) деп аталатын бұл қатынасты көрсететін теңдеуді шығара аламыз 3.11 сурет.

$$\mathrm{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

3.11-сурет – Көздің пропорционалдық теңдеуі

Мұнда p1, ..., p6 – 2D бағдарланған. Пропорция бағдар бағыты мен тиісті көлденең позицияның көлденең позициясы арасындағы бұрыш тек бір көлденең нүкте табылғанымен анықталады, бірақ екі тік нүкте салынған тік бұрышты екі жақтың арасындағы қашықтықты бағалайды [8].

Бізге ұнайтын нәрсе галлереяның глоссарийіне сәйкес келеді, мұнда айна ашылады, бірақ ай келгенде нөлге дейін тым тез емес. Осы қарапайым теңдеуді қолдану арқылы біз өңдеу әдістерін елемей, сәйкестендіру, көзқарасымызды анықтау, сәйкестендіру, байланыстыру немесе адамнан аулақ бола аламыз. Оны жарқын ету үшін келесі 3.12 - суретті қараңыз.



3.12-сурет – Жоғарғы сол жақ: көздің ашық болған кезде көзге көрінетін жерлерді визуализациялау. Жоғарғы оң жақ: көз жабылған кезде көздің бағдары. Төменде: уақыт аралығындағы кадр пішімінің кестесі. Көздің арақатынасының төмендеуі жыпылықтайды.

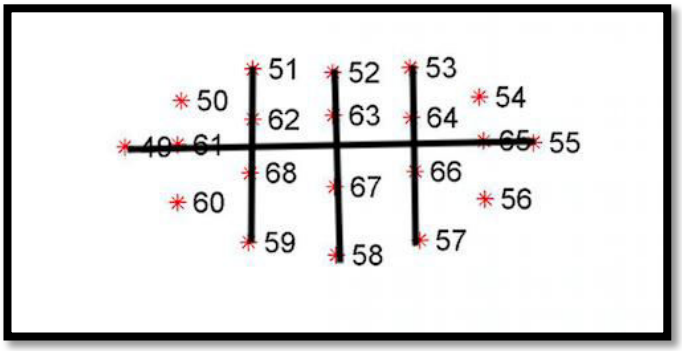
Жоғарғы сол жақ бұрышта көздеріміз толығымен ашылады – бұл жерде ара-қатынасы үлкен (r) және уақыт бойынша тұрақты болады. Алайда, адам жыпылықтаған кезде (жоғарғы оң жақта) көздің арақатынасы күрт төмендейді, нөлге жақындайды. Жоғарғы суретте бейнеклипке арналған көздің арақатынасы кестесі көрсетіледі. Көріп отырғанымыздай, көздің арақатынасы тұрақты, содан кейін тез нөлге дейін төмендейді, содан кейін қайтадан артады, бұл бір жыпылықтайды [8]. Көз жыпылықтауды анықтаудың визуализациясы 3.13 –

суретте.

3.13-сурет – Көз жыпылықтауды анықтау

Алайда бейне ағынының шуылына, бет-әлпет белгілерінің төмен деңгейіне немесе көру бұрышының тез өзгеруіне байланысты, көздің қарапайым арақатынасы шын мәнінде адам жыпылықтаған кезде жыпылықтаудың орын алғанын көрсететін жалған позитивті анықтауға әкелуі мүмкін. Бір медициналық мақалада [9] оқығанымыздай, адам орташа есеппен минутына 20 рет жыпылықтайды, бұл оның әр 3 секунд сайын жыпылықтайтынын көрсетеді. Осыған сүйене отырып, біздің жыпылықтайтын детекторды осы мәселелерге төзімді болу үшін, жыпықтауды оқығаннан кейін 3 секундтан кейін өту керек және жыпылықтаған кезде кемінде 3 кадр қажет.

Дәл осындай алгоритммен адамның күлгенін анықтауға болады. Адам ернінің индекстерін алып аламыз 3.14-сурет және белгілі нүктелердің ара қашықтығын санаймыз 3.15-сурет. Егер ара қашықтық белгіленген нормадан асып тұрса, адам күліп тұрған болып саналады. Ерін индекстерін, күлуді анықтайтын формуланы және олардың реализациясын 3.16-суреттен коре аласыздар:

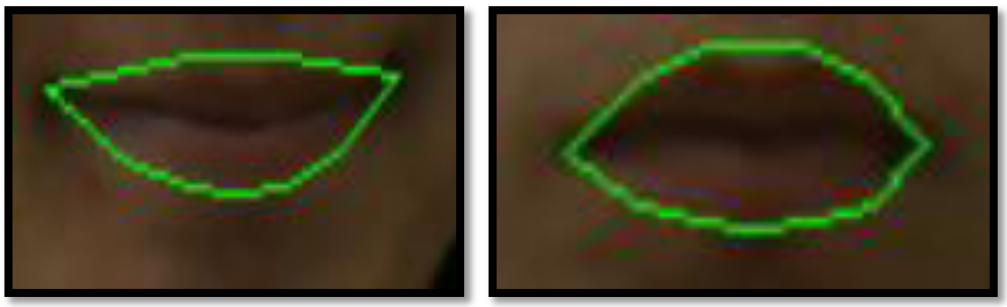


3.14-сурет – Ерінмен байланысты 20 бет белгісі

$$\frac{\|p_{51} - p_{59}\| + \|p_{52} - p_{58}\| + \|p_{53} - p_{57}\|}{3\,\|p_{48} - p_{55}\|}$$

MAR equation

3.15-сурет – Еріннің пропорционалдық теңдеуі



3.16-сурет – Күлуді анықтау

Осы алгоритмерді қолдана отырып мен камера алыдындағы адамды шын адам екеніне көз жеткізе аламын.

Менің жобам университетте немесе кез келген кәсіпорындарда бір қатар мәселелерді шешеді. Олар:

– университет қызметкерінің уақытын және энергиясын үнемдейді;

– адам еңбегінде үнемдейді;

– университет қызметкерінің аудиторияға кіруін бақылауды автоматизациялайды;

– есіктің құлыпқа ашылуын автоматизациялайды.

ҚОРЫТЫНДЫ

Бұл проектінің жасалу барысында ең бастысы қауіпсіздік жоғары болу көзделуде. Сол мақсатта бірнеше күрделі алгоритмдер жазылды.

Проектінің қызметі веб сайт қосымшасын қолдайтын аудиторияға кіруге рұқсат беруді бақылау жүйесін құру. Менің дипломдық проектімнің түпкі мақсаты, университет аудиторияларына немесе бөлмеге кіруге рұқсат керек кез-келген организацияның рұқсат беру жолын барынша жеңілдету және жылдамдату. Аудиторияға кіру магнитті карта арқылы немесе адам бетін тану арқылы жұмыс жасайды.

Осы дипломдық жоба барысында алдыма қойылған міндеттерді толықтай орындадым, яғни кіруге рұқсат беретін жүйелерді зерттеп, жобалап, талдап тиімді алгоритм жазып шығылды.

Құрылған жүйе қажетті ақпаратты жылдам алуға мүмкіндік береді және қызметкердің университет ішіндегі уақыты мен орны туралы нақты ақпарат ала аламыз, бұл өз кезегінде жұмыс сапасын жақсартады және қызметкер керек болған жағдайда жылдам тауып алуға мүмкіндік береді.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

1 Библиотека dlib // Сайттың электронды нұсқасы http://dlib.net

2 A. Asthana, S. Zafeoriou, S. Cheng, and M. Pantic. Incremental face alignment in the wild. – In Conference on Computer Vision and Pattern Recognition, 2014. – 1-7 p.

3 L. M. Bergasa, J. Nuevo, M. A. Sotelo, and M. Vazquez. Real-time system formonitoring driver vigilance. – In IEEE Intelligent Vehicles Symposium, 2004. – 1 p.

4 J. Cech, V. Franc, and J. Matas. A 3D approach to facial landmarks: Detection, refinement, and tracking. – In Proc. International Conference on Pattern Recognition, 2014. – 97 p.

5 M. Chau and M. Betke. Real time eye tracking and blink detection with USB cameras. Technical Report 2005-12. – Boston University Computer Science, May 2005. – 10 p.

6 Opencv кітапханасының көмегімен бет әлпетті табу әдістері. Т.М.Абдраимов, М.К. Қанатов, 2018. 4

7 Датасет для обучения библиотеки dlib // Сайттың электронды нұсқасы https://ibug.doc.ic.ac.uk/resources/300-W/

8 Tereza Soukupova and Jan ´ Cech. Real-Time Eye Blink Detection using Facial Landmarks, 21st Computer Vision Winter Workshop Luka Cehovin, Rok Mandeljc, Vitomir ˇ Struc (eds.) ˇ Rimske Toplice, Slovenia, February 3–5, 2016

9 В цифрах и фактах // Сайттың электронды нұсқасы http://www.aif.ru/health/life/v_cifrah_i_faktah_17_raz_v_minutu_v_srednem_morgayut_ nashi_glaza

10 Основы языка программирования java // Сайттың электронды нұсқасы https://www.java.com/en/download/faq/whatis_java.xml

11 Основы фреймворка java spring // Сайттың электронды нұсқасы https://docs.spring.io/spring/docs/current/spring-framework-reference/overview.html

12 Основы языка программирования python // Сайттың электрондынұсқасы https://www.python.org/doc/essays/blurb/

13 Основные понятия о bootstrap // Сайттың электронды нұсқасы https://www.simbla.com/post/what-is-bootstrap

14 Основные понятия о html // Сайттың электронды нұсқасы https://www.w3schools.com/whatis/whatis_html.asp

15 Основы фреймворка flask // Сайттың электронды нұсқасы http://flask.pocoo.org/

16 СКУД в наше время // Сайттың электронды нұсқасы https://searchsecurity.techtarget.com/definition/access-control

17 О системе perco-web // Сайттың электронды нұсқасы https://www.perco.ru/products/po-sistemy-kontrolya-dostupa-perco-web/

18 О системе parsec // Сайттың электронды нұсқасы https://www.parsec.ru/security-system/

19 О системе bolid orion pro // Сайттың электронды нұсқасы https://bolid.ru/production/orion/po-orion/po-arm/arm_orion_pro.html

20 О системе bastion2 // Сайттың электронды нұсқасы http://www.trevog.net/catalog/bastion/item/962/

21 UML диаграммы для ПО // Сайттың электронды нұсқасы https://www.edx.org/course/uml-class-diagrams-for-software-engineering

22 О продукте rational rose // Сайттың электронды нұсқасы https://www.predictiveanalyticstoday.com/ibm-rational-rose/

23 Как использовать Er – диаграмму // Сайттың электронды нұсқасы https://www.webopedia.com/TERM/E/entity_relationship_diagram.html

24 Библиотека tensorflow // Сайттың электронды нұсқасы https://www.tensorflow.org/

25 Библиотека keras // Сайттың электронды нұсқасы https://keras.io/

26 Библиотека opencv // Сайттың электронды нұсқасы https://opencv.org/

А Қосымшасы
(міндетті)

Университет аудиториясына уақытты ескере отырып кіруді бақылау жүйесін құруға арналған техникалық тапсырма

А.1 Жұмыстың мақсаты мен міндеттері

А.1.1 Жүйенің мақсаты

Бұл бағдарламалық қамтама университетіміздің аудиторияларына кіру мәселесін оптимизациялау үшін бағытталған.

Аудиторияға кіру өзінде белгілі бір мағлұматтар сақтай алатын магнетикті карталар арқылы жүзеге асады. Бұл БҚ арқылы біз мұғалімдердің сол аудиторияның кілтін алып келуге кететін уақытын және күш жігерін үнемдейміз және де аудиторияға кім қашан кіріп шыққаны туралы ақпаратты аламыз.

А.1.2 Жүйенің міндеттері

ACS келесі функцияларды қамтамасыз етуі керек:

Сәйкестендіру функциясының (кодының) тіркелген жадыдағы шағын жүйесін оқығанда, аудиториялар есігінде орнатылған атқарушы құрылғыларға басқару командаларын қалыптастыру және беру;

– жүйенің күйі туралы жұмыс станцияларына ақпарат беру;
– қызметкерлердің уақытын есепке алу;

А.2 Жүйенің жалпы сипаттамасы

А.2.2 АЭЖ жалпы сипаттамасы

Клиенттің сайтында құрылған ACS (PTC ACS) бағдарламалық-аппараттық кешені бағдарламалық және аппараттық қамтамасыз етуді қамтуы керек.

Бағдарлама бөлігінде келесі компоненттер бар:

– серверлік және пайдаланушы бағдарламалық қамтамасыз ету жиынтығы;
– жабдықты орнату және баптау үшін қосымша утилиталар;

– ACS жүйесін басқа тұтынушы жүйелерімен біріктіруді қамтамасыз ететін әзірлеу құралдары (SDK) жиынтығы.

Техникалық бөлімде:

– ACS контроллері;

– перифериялық жабдықтар: RFID оқырмандары, құлыптар, шығу түймелері және т.б.

– STS ACS – осы техникалық тапсырма талаптарын орындау үшін объектіде орнатылған ACC контроллерлерінің бөлінген құрылымы. Ақпарат серверлік бағдарламалық жасақтама орнатылған орталық серверде өңделеді. Деректерді өңдеу серверлерінің соңғы құрылғылармен өзара әрекеттестігі байланыс арналары арқылы жүзеге асырылуға тиіс: CAN және Ethernet.

А.3 Жүйелік талаптар

А.3.1 Жалпы жүйелік талаптар

А.3.1.1 Қол жеткізуді бақылау жүйесіне қойылатын жалпы талаптар

ACS келесі типтегі бөлмелерге және құрылғыларға қызмет көрсетеді: дәріс аудиториясы, практикалық және семинар сабақтары үшін аудиториялар, зертханалар.

Сіз оператор қолмен өрт дабылы жүйесін бастау немесе қашан осындай AL-300 сияқты кесілген-ығысу электромагниттік құлыптармен жабдықталған авариялық шығу, автоматты түрде ашу қажет. Күту режимінде эвакуациялық шығу қорғауға жатады.

Дәріс залдары, оқу сыныптары, ведомстволары мен зертханалар үшін, бақылау функцияларын кіру үшін қосымша, жүйе қауіпсіздік мүмкіндіктерінің іске асыруды қолдау керек, яғни қол картасының қорғауында үй-жайларынан орнату және жою болып табылады. Стандартты емес есіктерді құлыптау кезінде әрбір бөлмеде батырма орнатылады.

Өрт сөндіру жүйесін іске қосу кезінде барлық бақыланбаған үй-жайлардың есіктері ашық күйге түседі. Өрт сөндіру сигнализациясының жүйесі іске қосылған кезде қорғалатын аудиториялардың ерекшелігі.

А.3.1.2. Жазатайым оқиғалар кезінде ақпаратты сақтауға қойылатын талаптар

АБЖ бағдарламалық қамтамасыз ету дұрыс қайта іске қосу аппараттық

оның жұмыс істеуін қалпына келтіру керек. Ол қолмен сақтық көшірме жүйесі жүйесін және негізгі бағдарламалық қамтамасыз ету (операциялық жүйені, ДҚБЖ), күрделі бағдарламалық және аппараттық тапсырыс берушінің компонентін білдіреді автоматты ұйымдастыру мүмкін болады және (немесе) тиіс.

А.3.2 ACS контроллерлеріне және бағдарламалық жасақтамаға қойылатын талаптар

А.3.2.1 Қатынас контроллерлеріне қойылатын талаптар

– контроллерлері АБЖ әмбебап болуы тиіс және қатынау нүктелері бірнеше түрлерін қолдау үшін: есік, екі есік, Card Reader, оқырман карточкасын бағдаршам бақылау қақпасы бар магнетиктер;

– контроллерлер қосымша батарея (7 Ah кем емес) бар 220 50 Гц-ден ішкі қуат көзін болуы тиіс. Ол батареяның терең разрядты кезінде қанға автоматты жүктемені сақталуы тиіс. Барлық ПД режимі (220 жоғалуы, және батарея разряд Өзгертеді жүйесі серверге берілетін және кіруді бақылау құралдарын және жеңіл индикациялау үшін контроллер көрсетіледі үшін, қажет болған жағдайда қайталанады дыбысты белгі;

– контроллер сервер қатысуынсыз аппараттық деңгейде жазбалар кестесін, мерекелер және өзгерістер сақтауға тиіс.

А.3.2.2 ACS бағдарламалық жасақтамасына қойылатын талаптар

– ACS бағдарламалық жасақтамасы кез келген браузер ашылатын операциялық жүйелерде жұмыс істеуі тиіс;

– ACS клиенттік-сервердің архитектурасына ие болуы керек. Сервер және қашықтағы жұмыс станциялары домендік ұйыммен бөлінген желілерде жұмыс істеуі керек;

– бағдарламалық қамтамасыз ету ACS VPN туннельдер, және басқа да ұйымдастыру қажеттілігінсіз Inerrnet желісі арқылы қашықтағы жұмыс станциясы қосылу мүмкіндігі болуы керек. Толығырақ сәулет (АБЖ сервер IP, домендік атауы тыс бар. Workstations емес қоғамдық, «Сұр» IP арқылы қосылған).

Б Қосымшасы
(міндетті)

Бағдарлама мәтіні

```
//SecurityConfiguration.java
package kz.kaznitu.lessons.cofig;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.Configuration;
importorg.springframework.security.config.annotation.authentication.builders.
AuthenticationManagerBuilder;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.builders.WebSecurity;
import org.springframework.security.config.annotation.web.configuration.
EnableWebSecurity;
import org.springframework.security.config.annotation.web.configuration.
WebSecurityConfigurerAdapter;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.web.util.matcher.AntPathRequestMatcher;
import javax.sql.DataSource;
@Configuration
@EnableWebSecurity
public class SecurityConfiguration extends WebSecurityConfigurerAdapter {
   @Autowired
   private BCryptPasswordEncoder bCryptPasswordEncoder;
   @Autowired
   private DataSource dataSource;
   @Value("${spring.queries.users-query}")
   private String usersQuery;
   @Value("${spring.queries.roles-query}")
   private String rolesQuery;
   @Override
   protected void configure(AuthenticationManagerBuilder auth)
         throws Exception {
      auth.
            jdbcAuthentication()
            .usersByUsernameQuery(usersQuery)
            .authoritiesByUsernameQuery(rolesQuery)
            .dataSource(dataSource)
            .passwordEncoder(bCryptPasswordEncoder);
   }
   @Override
```

Б Қосымшасының жалғасы

```
    protected void configure(HttpSecurity http) throws Exception {
        http.
                authorizeRequests()
                //.antMatchers("/").permitAll()
                .antMatchers("/login").permitAll()
                .antMatchers("/registration").permitAll()
                .antMatchers("/admin/**").hasAuthority("ADMIN").anyRequest()
                .authenticated().and().csrf().disable().formLogin()
                .loginPage("/login").failureUrl("/login?error=true")
                .defaultSuccessUrl("/")
                .usernameParameter("email")
                .passwordParameter("password")
                .and().logout()
                .logoutRequestMatcher(new AntPathRequestMatcher("/logout"))
                .logoutSuccessUrl("/login").and().exceptionHandling()
                .accessDeniedPage("/access-denied");
    }
    @Override
    public void configure(WebSecurity web) throws Exception {
        web
                .ignoring()
                .antMatchers("/resources/**", "/static/**", "/css/**", "/js/**",
"/images/**");
    }}

//
//WebMvcConfig.java
package kz.kaznitu.lessons.cofig;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;
@Configuration
public class WebMvcConfig implements WebMvcConfigurer {
    @Bean
    public BCryptPasswordEncoder passwordEncoder() {
        BCryptPasswordEncoder bCryptPasswordEncoder = new
BCryptPasswordEncoder();
        return bCryptPasswordEncoder;
    }}
//
//LoginController.java
```

Б Қосымшасының жалғасы

```
package kz.kaznitu.lessons.controllers;
import kz.kaznitu.lessons.mod.User;
import kz.kaznitu.lessons.mod.Users;
import kz.kaznitu.lessons.service.UserService;
import kz.kaznitu.lessons.spm.listener;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.core.Authentication;
import org.springframework.security.core.context.SecurityContextHolder;
import org.springframework.stereotype.Controller;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.servlet.ModelAndView;
import javax.validation.Valid;
import java.util.Optional;
@Controller
@SessionAttributes("users")
public class LoginController {
   kz.kaznitu.lessons.spm.listener listener = new listener();
   @Autowired
   private UserService userService;
   @RequestMapping(value={"/login"}, method = RequestMethod.GET)
   public ModelAndView login(){
      ModelAndView modelAndView = new ModelAndView();
      modelAndView.setViewName("login");
      return modelAndView;
   }
   @RequestMapping(value="/registration", method = RequestMethod.GET)
   public ModelAndView registration(){
      ModelAndView modelAndView = new ModelAndView();
      User user = new User();
      modelAndView.addObject("user", user);
      modelAndView.setViewName("registration");
      return modelAndView;
   }
   @RequestMapping(value = "/registration", method = RequestMethod.POST)
   public ModelAndView createNewUser(@Valid User user, BindingResult
bindingResult) {
      ModelAndView modelAndView = new ModelAndView();
      User userExists = userService.findUserByEmail(user.getEmail());
      if (userExists != null) {
         bindingResult
```

Б Қосымшасының жалғасы

```
                .rejectValue("email", "error.user",
                        "There is already a user registered with the email provided");
        }
        if (bindingResult.hasErrors()) {
            modelAndView.setViewName("registration");
        } else {
            userService.saveUser(user);
            modelAndView.addObject("successMessage", "User has been registered
successfully");
            modelAndView.addObject("user", new User());
            modelAndView.setViewName("registration");

        }
        return modelAndView;
    }
    @RequestMapping(value="/admin/home", method = RequestMethod.GET)
    public ModelAndView home(){
        ModelAndView modelAndView = new ModelAndView();
        Authentication auth = SecurityContextHolder.getContext().getAuthentication();
        User user = userService.findUserByEmail(auth.getName());
        System.out.println(auth.getName());
        modelAndView.addObject("userName", "Welcome " + user.getName() + " " +
user.getLastName() + " (" + user.getEmail() + ")");
        modelAndView.addObject("adminMessage","Content Available Only for Users
with Admin Role");
        modelAndView.setViewName("admin/home");
        return modelAndView;
    }
    @RequestMapping(value="/", method = RequestMethod.GET)
    public ModelAndView home1() {
        ModelAndView model = new ModelAndView();
        Authentication auth = SecurityContextHolder.getContext().getAuthentication();
        Optional<Users> users = userService.findUserByLogin(auth.getName());
        System.out.println(auth.getName() + users.get().getName());
        userService.findUserByLogin(auth.getName());
        model.addObject("userName1",  users.get().getSurname()+ " " +
users.get().getName());
        listener.pauseRead();
        model.setViewName("main_admin");
        return model;
    }
}
```

Б Қосымшасының жалғасы

```
//
//MainController.java
package kz.kaznitu.lessons.controllers;
import kz.kaznitu.lessons.mod.*;
import kz.kaznitu.lessons.mod.Timestamp;
import kz.kaznitu.lessons.repas.*;
import kz.kaznitu.lessons.service.UserService;
import kz.kaznitu.lessons.spm.*;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.core.Authentication;
import org.springframework.security.core.context.SecurityContextHolder;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.servlet.ModelAndView;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import java.util.*;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import kz.kaznitu.lessons.repas.RoleRepository;
import javax.validation.Valid;
@Controller
public class MainController {
    listener listener = new listener();
    SKUD skud = new SKUD();
    @Autowired
    private RegistrationRepas registrationRepas;
    @Autowired
    private UsersRepas usersRepas;
    private long a;
    @Autowired
    private UserService userService;
    private BCryptPasswordEncoder bCryptPasswordEncoder  =new
BCryptPasswordEncoder();
    private RoleRepository roleRepository;
    @Autowired
    private InstitutRepas institutRepas;
    @Autowired
    private CafedraRepas cafedraRepas;
    @Autowired
    private TimestampRepas timestampRepas;
```

Б Қосымшасының жалғасы

```
    public MainController(BCryptPasswordEncoder bCryptPasswordEncoder) {
        this.bCryptPasswordEncoder = bCryptPasswordEncoder;
    }
    @RequestMapping(value = "/Korzina", method = RequestMethod.GET)
    public ModelAndView footballsList2(Model model) {
        ModelAndView modelAndView = new ModelAndView();
        skud.pauseRead1();
        listener.pauseRead();
        listener.startListener();
        Authentication auth = SecurityContextHolder.getContext().getAuthentication();
        Optional<Users> users = userService.findUserByLogin(auth.getName());
        modelAndView.addObject("userName1",  users.get().getSurname()+ " " +
users.get().getName());
        modelAndView.addObject("userName2", users.get().getSurname());
        model.addAttribute("users", usersRepas.findAll());
        modelAndView.setViewName("Korzina");
        return modelAndView;
    }
    @RequestMapping(value = "/delete", method = RequestMethod.GET)
    public ModelAndView deleteAuthor(@RequestParam("id") long idd) {
        usersRepas.deleteById(idd);
        return new ModelAndView("redirect:/Korzina");
    }
    @RequestMapping(value = "/registrations", method = RequestMethod.GET)
    public String clubsAdd(Model model) {
        model.addAttribute("registration", new Registration());
        return "registrations";
    }

    @RequestMapping(value = "/users", method = RequestMethod.GET)
    public String clubsAdd1(Model model) {
        model.addAttribute("user", new Users());
        model.addAttribute("users", usersRepas.findAll());
        return "users";
    }
    @RequestMapping(value = "/registrations", method = RequestMethod.POST)
    public String clubsAdd(@ModelAttribute("registration") @Valid Registration
registration, BindingResult result) {
        if (result.hasErrors()) {
            return "registrations";
        }
        registrationRepas.save(registration);
```

Б Қосымшасының жалғасы

```
        return "redirect:/registrations";
    }
    @RequestMapping(value = "/users", method = RequestMethod.POST)
    public String clubsAdd1(@ModelAttribute("users") @Valid Users users,
BindingResult result) {
        if (result.hasErrors()) {
            return "users";
        }
        usersRepas.save(users);
        return "redirect:/users";
    }
    @RequestMapping(value="/history_admin", method = RequestMethod.GET)
    public ModelAndView footballsList435(Model model) {
        ModelAndView modelAndView = new ModelAndView();
        Authentication auth = SecurityContextHolder.getContext().getAuthentication();
        Optional<Users> users = userService.findUserByLogin(auth.getName());
        modelAndView.addObject("userName1",  users.get().getSurname()+ " " +
users.get().getName());
        modelAndView.addObject("userName2", users.get().getId_user());
        model.addAttribute("users", usersRepas.findAll());
        model.addAttribute("user", new Users());
        model.addAttribute("timestamp", new Timestamp());
        modelAndView.setViewName("history_admin");
        return modelAndView;
    }
    @RequestMapping(value="/users_db", method = RequestMethod.GET)
    public ModelAndView footballsList436(Model model) {
        ModelAndView modelAndView = new ModelAndView();
        Authentication auth = SecurityContextHolder.getContext().getAuthentication();
        Optional<Users> users = userService.findUserByLogin(auth.getName());
        modelAndView.addObject("userName1",  users.get().getSurname()+ " " +
users.get().getName());
        model.addAttribute("users", usersRepas.findAll());
        modelAndView.setViewName("users_db");
        return modelAndView;
    }
    @RequestMapping(value="/find_db", method = RequestMethod.GET)
    public ModelAndView findDB(Model model) {
        ModelAndView modelAndView  = new ModelAndView();
        model.addAttribute("users", usersRepas.findAll());
        model.addAttribute("user", new Users());
        model.addAttribute("timestamp", new Timestamp());
```

Б Қосымшасының жалғасы

```
        model.addAttribute("instituts", institutRepas.findAll());
        model.addAttribute("cafedras", cafedraRepas.findAll());
        model.addAttribute("times", timestampRepas.findAll());
        Authentication auth = SecurityContextHolder.getContext().getAuthentication();
        Optional<Users> users = userService.findUserByLogin(auth.getName());
        modelAndView.addObject("userName1",  users.get().getSurname()+ " " +
users.get().getName());
        modelAndView.setViewName("find_db");

        return modelAndView;
    }
    @RequestMapping(value="/add_lessons", method = RequestMethod.GET)
    public String clubsList(Model model) {
        model.addAttribute("registrations", registrationRepas.findAll());
        skud.startListener1();
        return "BookingList";
    }
    @RequestMapping(value = "/drop", method = RequestMethod.GET)
    public ModelAndView deleteAuthor1(@RequestParam("id") Long id) {
        registrationRepas.deleteById(id);
        return new ModelAndView("redirect:/BookingList");
    }
    @GetMapping("/all2")
    public String allUsers2(Model model) {
        List<Users> users = (List<Users>) usersRepas.findAll();
        model.addAttribute("users", users);
        return "users";
    }
    @PostMapping("/editUser")
    public ModelAndView modelAndView(@Valid Users users) {
        ModelAndView modelAndView = new ModelAndView();
        users.setId_user(a);
        users.setCard(users.getCard());
        users.setSurname(users.getSurname());
        users.setName(users.getName());
        users.setInst(users.getInst());
        users.setCaf(users.getCaf());
        users.setLogin(users.getLogin());
        users.setPassword(users.getPassword());
        userService.saveUserss(users);
        modelAndView.setViewName("redirect:/users_db");
        return  modelAndView;
```

```
    }
    @RequestMapping(value = "/editUser",method = RequestMethod.GET)
    public ModelAndView editUser(Model model,@RequestParam("id") long id){
        ModelAndView modelAndView = new ModelAndView();
        a=id;
        Optional<Users> users = (Optional <Users>) usersRepas.findById(id);
        Optional<Institut> institut = (Optional <Institut>) institutRepas.findById(id);
        Optional<Cafedra> cafedra = (Optional <Cafedra>) cafedraRepas.findById(id);
        model.addAttribute("instituts", institutRepas.findAll());
        model.addAttribute("cafedras", cafedraRepas.findAll());
        model.addAttribute("user",users);
        model.addAttribute("institut",institut);
        model.addAttribute("cafedra",cafedra);
        Authentication auth = SecurityContextHolder.getContext().getAuthentication();
        Optional<Users> users1 = userService.findUserByLogin(auth.getName());
        modelAndView.addObject("userName1",  users1.get().getSurname()+ " " +
users1.get().getName());
        modelAndView.addObject("users", users);
        return new ModelAndView("smp");
    }
    @RequestMapping("/editUser")
    public String showForm2(Model model){
        model.addAttribute("user",new Users());
        model.addAttribute("user",new Institut());
        model.addAttribute("user",new Cafedra());
        return "smp";
    }
}
//
//Registration.java
package kz.kaznitu.lessons.mod;
import kz.kaznitu.lessons.spm.listener;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Size;
@Entity
public class Registration {
    public String GN;
    @Id
```

Б Қосымшасының жалғасы

```
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;
    @NotNull
    @Size(min = 3,max = 50, message = "Error")
    private String card;
    @NotNull
    @Size(min = 3,max = 50, message = "Error")
    private String surname;
    @NotNull
    @Size(min = 3,max = 50, message = "Error")
    private String name;
    @NotNull
    @Size(min = 0, max = 50, message = "Error")
    private String cafedra;
    @NotNull
    @Size(min = 0, max = 50, message = "Error")
    private String institut;
    @NotNull
    @Size(min = 0, max = 50, message = "Error")
    private String login;
    @NotNull
    @Size(min = 0, max = 50, message = "Error")
    private String password;
    @NotNull
    @Size(min = 0, max = 2, message = "Error")
    private String status;
    public String ListenerId;
    public Registration(){
        super();
    }

    public Registration(int id, String card, String surname, String name, String cafedra,
String institut, String login, String password, String status, String ListenerId){
        this.id=id;
        this.card=card;
        this.surname=surname;
        this.name=name;
        this.cafedra=cafedra;
        this.institut=institut;
        this.login=login;
        this.password=password;
        this.status=status;
```

```
        this.ListenerId=ListenerId;
    }
    public Registration(String card, String surname, String name, String cafedra, String
institut, String login, String password, String status, String ListenerId){
        this.id=id;
        this.card=card;
        this.surname=surname;
        this.name=name;
        this.cafedra=cafedra;
        this.institut=institut;
        this.login=login;
        this.password=password;
        this.status=status;
        this.ListenerId= ListenerId;
    }
    public int getId(){
        return id;
    }
    public void setId(){
        this.id=id;
    }
    public String getCard() {
        return card;
    }
    public void setCard(String card){
        this.card=card;
    }
    public String getSurname() {
        return surname;
    }
    public void setSurname(String surname){
        this.surname=surname;
    }
    public String getName() {
        return name;
    }
    public void setName(String name){
        this.name=name;
    }
    public String getCafedra() {
        return cafedra;
    }
```

Б Қосымшасының жалғасы

```
    public void setCafedra(String cafedra){
        this.cafedra=cafedra;
    }
    public String getInstitut() {
        return institut;
    }
    public void setInstitut(String institut){
        this.institut=institut;
    }
    public String getLogin() {
        return login;
    }
    public void setLogin(String login){
        this.login=login;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password){
        this.password=password;
    }
    public String getStatus(){
        return status;
    }
    public void setStatus(String status){
        this.status=status;
    }
    public String getListenerId(){
        return ListenerId;
    }
    public void setListenerId(String ListenerId){
        System.out.println(ListenerId + "asd");
        this.ListenerId=ListenerId;
        GN=ListenerId;
    }
   @Override
    public String toString() {
        setListenerId(ListenerId);
        System.out.println(ListenerId+" otvet");
        return ListenerId + " qwerty";
    }
}
```

Б Қосымшасының жалғасы

```
//
//Role.java
package  kz.kaznitu.lessons.mod;
import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;
import javax.persistence.*;
@Data
@Builder
@AllArgsConstructor
@NoArgsConstructor
@Entity
@Table(name = "role")
public class Role {
   @Id
   @GeneratedValue(strategy = GenerationType.AUTO)
   @Column(name = "role_id")
   private int id;
   @Column(name = "role")
   private String role;
}

//
//RegistrationRepas.java
package kz.kaznitu.lessons.repas;
import kz.kaznitu.lessons.mod.Registration;
import kz.kaznitu.lessons.spm.listener;
import org.springframework.data.repository.CrudRepository;
import java.util.List;
import java.util.Optional;
public interface RegistrationRepas extends CrudRepository<Registration, Long> {
   Optional<Registration> findById(Long id);
}

//
//RoleRepository.java
package kz.kaznitu.lessons.repas;
import kz.kaznitu.lessons.mod.Role;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;
@Repository("roleRepository")
```

Б Қосымшасының жалғасы

```
public interface RoleRepository extends JpaRepository<Role, Integer> {
   Role findByRole(String role);
}

//
//UserService.java
package kz.kaznitu.lessons.service;
import kz.kaznitu.lessons.mod.Role;
import kz.kaznitu.lessons.mod.User;
import kz.kaznitu.lessons.mod.Users;
import kz.kaznitu.lessons.repas.RoleRepository;
import kz.kaznitu.lessons.repas.UserRepository;
import kz.kaznitu.lessons.repas.UsersRepas;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.stereotype.Service;
import java.util.Arrays;
import java.util.HashSet;
import java.util.Optional;
@Service("userService")
public class UserService {
   private UserRepository userRepository;
   private UsersRepas usersRepas;
   private RoleRepository roleRepository;
   private BCryptPasswordEncoder bCryptPasswordEncoder;
   private long a;
   @Autowired
   public UserService(UsersRepas usersRepas,
                      UserRepository userRepository,
                      RoleRepository roleRepository,
                      BCryptPasswordEncoder bCryptPasswordEncoder) {
      this.usersRepas= usersRepas;
      this.userRepository= userRepository;
      this.roleRepository = roleRepository;
      this.bCryptPasswordEncoder = bCryptPasswordEncoder;
   }
   public User findUserByEmail(String email) {
      return userRepository.findByEmail(email);
   }
   public Optional<Users> findUserByLogin(String login) {
      return (Optional<Users>) usersRepas.findByLogin(login);
   }
```

Б Қосымшасының жалғасы

```
  public User saveUser(User user) {
      user.setPassword(bCryptPasswordEncoder.encode(user.getPassword()));
      user.setActive(1);
      Role userRole = roleRepository.findByRole("ADMIN");
      user.setRoles(new HashSet<Role>(Arrays.asList(userRole)));
      return userRepository.save(user);
   }
   public Users saveUserss(Users user) {
      user.setPassword(bCryptPasswordEncoder.encode(user.getPassword()));
      user.setActive(1);
      user.setId_user(a);
      user.setCard(user.getCard());
      user.setSurname(user.getSurname());
      user.setName(user.getName());
      user.setInst(user.getInst());
      user.setCaf(user.getCaf());
      user.setLogin(user.getLogin());
      return usersRepas.save(user);
   }
}

//
//config.java
package kz.kaznitu.lessons.spm;
public class config {
   public static final String COM = "COM3";
   public static final String myDriver = "org.gjt.mm.mysql.Driver";
   public static final String myUrl =
"jdbc:mysql://localhost:3301/db_skud?useUnicode=yes&characterEncoding=UTF-
8";
   public static final String db_username="root";
   public static final String db_password="1234";
}

//
//listener.java
package kz.kaznitu.lessons.spm;
import jssc.SerialPort;
import jssc.SerialPortEvent;
import jssc.SerialPortEventListener;
import jssc.SerialPortException;
import kz.kaznitu.lessons.mod.Registration;
```

Б Қосымшасының жалғасы

```
import org.springframework.stereotype.Controller;
import java.io.*;
import java.sql.*;
@Controller
public class listener {
    private static String card;
    boolean status=true;
    public static String SS="";
   public void pauseRead()  {
        if (status == false) {
            System.out.println("попало");
            status=true;
            if (serialPort != null || serialPort.isOpened ()) {
                System.out.println("Не пустой!");
                try {
                    serialPort.purgePort (1);
                    serialPort.purgePort (2);
                    serialPort.closePort();
                    System.out.println("Закрыт");
                } catch (SerialPortException ex) {
                    System.out.println(ex);
                }
            }
        }else{System.out.println("для начало норм");   }
    }
     static void   codeRead() throws IOException{
        if(SS.indexOf("M")>-1&&SS.indexOf(')')>0){
            SS=SS.substring(SS.indexOf("M"), SS.indexOf(')')+1);
            try    {
                Class.forName(config.myDriver);
                Connection conn = DriverManager.getConnection(config.myUrl,
config.db_username, config.db_password);
                Statement st =  conn.createStatement();
                String sql = "SELECT * FROM users";
                ResultSet rs = st.executeQuery(sql);
                String ss="";
                while(rs.next()){
                    ss=rs.getString("card");
                    if(SS.equals(ss)){
                        System.out.println("Данная карта уже была добавлена ранее.");
                   break;}
            }
```

Б қосымшасының жалғасы

```
            if(!SS.equals(ss)){
               System.out.println(SS);
               listener listener1 =new listener();
               listener1.fetchById(SS);
               card = SS;
               String sql1 = "INSERT INTO users(card, surname, name, active) " +
                     "VALUES ('"+card+"', ",",1)";
               st.executeUpdate(sql1);
               st.close();
            }
         } catch (Exception e){
            System.err.println("Got an exception! 111111 ");
            System.err.println(e.getMessage());
         }
      }
      else {System.out.println("Ваша карта не поддерживается, обратитесь
администратору");}
   }
   static void printS(String s){
      SS=SS+s;
            if(checkStr(SS)){
         try {
            codeRead();
         } catch (IOException e) {
            e.printStackTrace();
         }
         SS="";}
   }
   static boolean checkStr(String s){
      for(int i=0;i<s.length()-3;i++){
         if(s.substring(i, i+4).equals("card")){return true;}}
      return false;}
   private static SerialPort serialPort; /*Создаем объект типа SerialPort*/
   public void startListener(){
      status=false;
      System.out.println(status+" статус");
      serialPort = new SerialPort (config.COM); /*Передаем в конструктор
суперкласса имя порта с которым будем работать*/
      try {
         serialPort.openPort (); /*Метод открытия порта*/
serialPort.setParams (SerialPort.BAUDRATE_9600, SerialPort.DATABITS_8,
```

Б Қосымшасының жалғасы

```
SerialPort.STOPBITS_1, SerialPort.PARITY_NONE); /*Задаем основные параметры протокола UART*/
        serialPort.setEventsMask (SerialPort.MASK_RXCHAR); /*Устанавливаем маску или список события на которые будет происходить реакция. В данном случае это приход данных в буффер порта*/
        serialPort.addEventListener (new EventListener ()); /*Передаем экземпляр класса EventListener порту, где будет обрабатываться события. Ниже описан класс*/
      }
      catch (SerialPortException ex) {
        System.out.println (ex);
      }}
   public static class EventListener implements SerialPortEventListener {
/*Слушатель срабатывающий по появлению данных на COM-порте*/
      String s="";
      public void serialEvent (SerialPortEvent event) {
        if (event.isRXCHAR () && event.getEventValue () > 0){ /*Если происходит событие установленной маски и количество байтов в буфере более 0*/
            try {
              String data = serialPort.readString (event.getEventValue()); /*Создаем строковую переменную  data, куда и сохраняем данные*/
              printS(data);
            }
            catch (SerialPortException ex) {
              System.out.println (ex);
            }}}}
   public Registration fetchById(String string){
      Registration registration = new Registration();
      registration.setListenerId(string);
      return registration;
   }}

//
//SKUD.java
package kz.kaznitu.lessons.spm;
import jssc.SerialPort;  /*Импорт классов библиотеки jssc*/
import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
```

Б Қосымшасының жалғасы

```
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.PrintStream;
import java.nio.charset.Charset;
import java.sql.*;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Random;
import javax.swing.JOptionPane;
import jssc.SerialPortEvent;
import jssc.SerialPortEventListener;
import jssc.SerialPortException;
import kz.kaznitu.lessons.spm.config;
import org.springframework.stereotype.Controller;
import java.awt.*;
import java.awt.event.*;
@Controller
public class SKUD{  /*Класс чтения из порта*/
   private  static String card;
   private  static String currentTime;
   private  static long iid1;
   boolean status=true;
   private static String SS="";
   private static int t = 0;
   public void pauseRead1()  {
      if (status == false) {
         System.out.println("попало");
         status=true;
         if (serialPort != null || serialPort.isOpened ()) {
            System.out.println("Не пустой!");
            try {
               serialPort.purgePort (1);
               serialPort.purgePort (2);
               serialPort.closePort();
               System.out.println("Закрыт");
            } catch (SerialPortException ex) {
               System.out.println(ex);
            }
         }
      }else{System.out.println("для начало норм");   }
   }
   static void printUser(String s) throws IOException{
```

Б Қосымшасының жалғасы

```
      try    {
         Class.forName(config.myDriver);
         Connection conn = DriverManager.getConnection(config.myUrl,
config.db_username, config.db_password);
         Statement st =  conn.createStatement();
         String sql = "SELECT * FROM users";
         ResultSet rs = st.executeQuery(sql);
         String ss="";
         long iid;
         while(rs.next()){
            if(s.equals(rs.getString("card"))){ss=rs.getString("card");
iid=rs.getLong("id_user");
               card = ss;
               iid1 = iid;
               System.out.println(card + "считывающиеся карта");
               long curTime = System.currentTimeMillis();
               Date curDate = new Date(curTime);
               String curStringDate = new SimpleDateFormat("dd.MM.yyyy,
HH:mm:ss").format(curDate);
               currentTime=curStringDate;
               System.out.println(currentTime + "Current time");
            }
         }
         if(!s.equals(ss)){
            System.out.println("Не зарегистрированная карта!");
         }
         sql = "INSERT INTO timestamp(card, time_in, id_user) " +
               "VALUES ('"+card+"', '"+currentTime+"', '"+iid1+"')";
         st.executeUpdate(sql);
         st.close();
      } catch (Exception e){
         System.err.println("Got an exception! ");
         System.err.println(e.getMessage());      }
   }
   static void codeRead(){
      if(SS.indexOf("M")>-1&&SS.indexOf(')')>0){
         try{
            printUser(SS.substring(SS.indexOf("M"), SS.indexOf(')')+1));}
         catch(IOException e){System.out.println(e);}
      }
      else {
```

Б Қосымшасының жалғасы

```
            System.out.println("Ваша карта не поддерживается, обратитесь
администратору");
        }
    }
    static void printS(String s){
        SS=SS+s;
        if(checkStr(SS)){
            codeRead();
            SS="";}
    }
    static boolean checkStr(String s){
        for(int i=0;i<s.length()-3;i++){
            if(s.substring(i, i+4).equals("card")){return true;}}
        return false;}
    private static SerialPort serialPort; /*Создаем объект типа SerialPort*/
    public void startListener1(){
        status = false;
        serialPort = new SerialPort (config.COM); /*Передаем в конструктор
суперкласса имя порта с которым будем работать*/
        try {
            serialPort.openPort (); /*Метод открытия порта*/
            serialPort.setParams (SerialPort.BAUDRATE_9600,
SerialPort.DATABITS_8, SerialPort.STOPBITS_1, SerialPort.PARITY_NONE);
/*Задаем основные параметры протокола UART*/
            serialPort.setEventsMask (SerialPort.MASK_RXCHAR); /*Устанавливаем
маску или список события на которые будет происходить реакция. В данном
случае это приход данных в буффер порта*/
            serialPort.addEventListener (new EventListener ()); /*Передаем экземпляр
класса EventListener порту, где будет обрабатываться события. Ниже описан
класс*/
        }
        catch (SerialPortException ex) {
            System.out.println (ex);
        }
    }
    private static class EventListener implements SerialPortEventListener {
/*Слушатель срабатывающий по появлению данных на COM-порте*/
        String s="";
        public void serialEvent (SerialPortEvent event) {
            if (event.isRXCHAR () && event.getEventValue () > 0){ /*Если
происходит событие установленной маски и количество байтов в буфере более
0*/
```

Б Қосымшасының жалғасы

```
            try {
               String data = serialPort.readString (event.getEventValue()); /*Создаем
строковую переменную  data, куда и сохраняем данные*/
            printS(data);
            }
            catch (SerialPortException ex) {
               System.out.println (ex);
            }}}}}

//
//app.py
#!/usr/bin/env python
from importlib import import_module
import os
from flask import Flask, render_template, Response, jsonify
os.environ["CUDA_DEVICE_ORDER"]="PCI_BUS_ID"
os.environ["CUDA_VISIBLE_DEVICES"]="0"
from camera import Camera
app = Flask(__name__)
@app.route('/')
def index():
   """Video streaming home page."""
   return render_template('index.html')
def gen(camera):
   """Video streaming generator function."""
   while True:
      try:
         frame = camera.get_frame()
         yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
      except Exception as e:
         print("error : ", e)
@app.route('/face_video')
def video_feed():
   """Video streaming route. Put this in the src attribute of an img tag."""
   return Response(gen(Camera()),
                   mimetype='multipart/x-mixed-replace; boundary=frame')
if __name__ == '__main__':
Camera.set_video_source('rtsp://admin:Qq12345678@172.16.3.52:554/Streaming/C
hannels/101')
   gen(Camera())
   app.run(host='172.16.3.99', port=8824, threaded=True)
```

Б Қосымшасының жалғасы

```
//
//camera.py
import cv2
from base_camera import BaseCamera
from time import localtime, strftime
import os
import time
import numpy as np
import sys
import glob
from func import FaceControl
import random
import h5py
import dlib
from matplotlib.pyplot import imshow
import psycopg2
from psycopg2.extras import DictCursor
from datetime import datetime
connect = psycopg2.connect(database='FaceRecognizer', user='postgres',
host='192.168.150.18', password='Idet2050!')
cursor = connect.cursor()
os.environ["CUDA_DEVICE_ORDER"]="PCI_BUS_ID"
os.environ["CUDA_VISIBLE_DEVICES"]="0"
faceCascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
dnnFaceDetector =
dlib.cnn_face_detection_model_v1("mmod_human_face_detector.dat")
def normalize(image):
    faceRects = dnnFaceDetector(image, 0)
    if len(faceRects)>0:
        return "face"
    else:
        return "notface"
threshold=0.2
(width, height) = (224, 224)
idx=0
imgDim = 224
a = glob.glob('dataset/*')
#print(a)
class Camera(BaseCamera):
    video_source = 0
    @staticmethod
    def set_video_source(source):
```

Б Қосымшасының жалғасы

```
        Camera.video_source = source
    @staticmethod
    def frames():
        fc = FaceControl()
        camera = cv2.VideoCapture(Camera.video_source)
        model = fc.loadVggFaceModel()
        count=0
        while True:
            ret, frame = camera.read()
            try:
                x_sh=int(frame.shape[1]/6)
                y_sh=int(frame.shape[0]/6)
                rrec = cv2.rectangle(frame, (x_sh*2, y_sh*2), (x_sh*4, y_sh*4), (0, 0,
255), 2)
                crop_rrec = frame[y_sh*2:y_sh*4, x_sh*2:x_sh*4]
                faces = faceCascade.detectMultiScale(
                    crop_rrec,
                    scaleFactor=1.1,
                    minNeighbors=5,
                    minSize = (200, 200)
                )
                for (x,y,w,h) in faces:
                    face = crop_rrec[y:y+h,x:x+w]
                    check = normalize(face)
                    #print(check)
                    if check=='face':
                        face_resize = cv2.resize(face, (224, 224))
                        face_reshape = face_resize.reshape(1,224,224,3)
                        #print(face_reshape.shape)
                        cv2.rectangle(crop_rrec, (x,y), (x+w,y+h), (157,237,33), 2)
                        img1_representation =
model.predict(fc.preprocess_image(face_reshape))[0,:]
                        facenames = []
                        result = []
                        avarages = []
                        for i in range(len(a)):
                            faces = glob.glob(a[i]+'/*.npy')
                            name = os.path.basename(a[i])
                            facenames.append(name)
                            cosines=[]
                            for j in range(len(faces)):
                                read = np.load(faces[j]).item()
```

Б Қосымшасының жалғасы

```
                    cosine_similarity = fc.findCosineSimilarity(img1_representation,
read[0])
                    cosines.append(cosine_similarity)
                minimum = cosines.index(min(cosines))
                avarage = sum(cosines)/len(cosines)
                avarages.append(avarage)
                result.append(cosines[minimum])


            avg_min = min(avarages)
            id = result.index(min(result))
            print(facenames[id])
            print(avg_min)
            print(result[id])
            if result[id] < 0.15:
                external_id = str(facenames[id]),
                camera_id = 1
                photo_time = datetime.now()
                photo_url = 'camera_photos/'+str(facenames[id])+'-
'+str(photo_time)+'.jpg'
                cv2.imwrite(photo_url, face_resize)
                lf_name = ''
                access_allowed = None
                cursor.execute("""SELECT * FROM people WHERE
external_id=%s""", external_id)
                for row in cursor:
                    access_allowed = row[5]
                    lf_name = row[2]+' '+row[3]
                    person_id = row[0]
                print(person_id)
                cursor.execute("SELECT nextval('camera_photos_id_seq')")
                for row in cursor:
                    cp_id = row[0]
                postgres_insert_query = """INSERT INTO camera_photos (id,
camera_id, person_id, photo_time, file_path) VALUES (%s,%s,%s,%s,%s)"""
                record_to_insert = (cp_id,camera_id,person_id,photo_time,
photo_url)
                cursor.execute(postgres_insert_query, record_to_insert)
                connect.commit()
                if access_allowed:
                    print(lf_name+' WELCOME')
                    text = 'ACCESS IS ALLOWED'
```

Б Қосымшасының жалғасы

```
                        elif access_allowed==False:
                        text = 'ACCESS IS DENIED'
                    else:
                        text = 'UNVERIFIED'
                        external_id = 'none',
                        photo_time = datetime.now()
                        photo_url = 'none/face-'+str(photo_time)+'.jpg'
                        cursor.execute("""SELECT * FROM people WHERE
external_id=%s""", external_id)
                        for row in cursor:
                            person_id = row[0]
                        print(person_id)
                        cv2.imwrite(photo_url, face_resize)
                        cursor.execute("SELECT nextval('camera_photos_id_seq')")
                        for row in cursor:
                            cp_id = row[0]
                        postgres_insert_query = """INSERT INTO camera_photos (id,
camera_id, person_id, photo_time, file_path) VALUES (%s,%s,%s,%s,%s)"""
                        record_to_insert = (cp_id,camera_id,person_id,photo_time,
photo_url)
                        cursor.execute(postgres_insert_query, record_to_insert)
                        connect.commit()
                    print(text)
                cv2.putText(frame,text,(x_sh*2-50,y_sh*2-50),
cv2.FONT_HERSHEY_SIMPLEX, 1,(0,255,0),1,cv2.LINE_AA)
                yield cv2.imencode('.jpg', frame)[1].tobytes()
            yield cv2.imencode('.jpg', frame)[1].tobytes()
        except:
            pass

//
//func.py
from keras.models import Model, Sequential
from keras.layers import Input, Convolution2D, ZeroPadding2D, MaxPooling2D,
Flatten, Dense, Dropout, Activation
from PIL import Image
import numpy as np
from keras.preprocessing.image import load_img, save_img, img_to_array
from keras.applications.imagenet_utils import preprocess_input
from keras.preprocessing import image
import matplotlib.pyplot as plt
import cv2
```

Б Қосымшасының жалғасы

```
import dlib
class FaceControl():
   def loadVggFaceModel(self):
      model = Sequential()
      model.add(ZeroPadding2D((1,1),input_shape=(224,224, 3)))
      model.add(Convolution2D(64, (3, 3), activation='relu'))
      model.add(ZeroPadding2D((1,1)))
      model.add(Convolution2D(64, (3, 3), activation='relu'))
      model.add(MaxPooling2D((2,2), strides=(2,2)))
      model.add(ZeroPadding2D((1,1)))
      model.add(Convolution2D(128, (3, 3), activation='relu'))
      model.add(ZeroPadding2D((1,1)))
      model.add(Convolution2D(128, (3, 3), activation='relu'))
      model.add(MaxPooling2D((2,2), strides=(2,2)))
      model.add(ZeroPadding2D((1,1)))
      model.add(Convolution2D(256, (3, 3), activation='relu'))
      model.add(ZeroPadding2D((1,1)))
      model.add(Convolution2D(256, (3, 3), activation='relu'))
      model.add(ZeroPadding2D((1,1)))
      model.add(Convolution2D(256, (3, 3), activation='relu'))
      model.add(MaxPooling2D((2,2), strides=(2,2)))
      model.add(ZeroPadding2D((1,1)))
      model.add(Convolution2D(512, (3, 3), activation='relu'))
      model.add(ZeroPadding2D((1,1)))
      model.add(Convolution2D(512, (3, 3), activation='relu'))
      model.add(ZeroPadding2D((1,1)))
      model.add(Convolution2D(512, (3, 3), activation='relu'))
      model.add(MaxPooling2D((2,2), strides=(2,2)))
      model.add(ZeroPadding2D((1,1)))
      model.add(Convolution2D(512, (3, 3), activation='relu'))
      model.add(ZeroPadding2D((1,1)))
      model.add(Convolution2D(512, (3, 3), activation='relu'))
      model.add(ZeroPadding2D((1,1)))
      model.add(Convolution2D(512, (3, 3), activation='relu'))
      model.add(MaxPooling2D((2,2), strides=(2,2)))
      model.add(Convolution2D(4096, (7, 7), activation='relu'))
      model.add(Dropout(0.5))
      model.add(Convolution2D(4096, (1, 1), activation='relu'))
      model.add(Dropout(0.5))
      model.add(Convolution2D(2622, (1, 1)))
      model.add(Flatten())
      model.add(Activation('softmax'))
```

Б Қосымшасының жалғасы

```
        from keras.models import model_from_json
        model.load_weights('vgg_face_weights.h5')
        vgg_face_descriptor = Model(inputs=model.layers[0].input,
outputs=model.layers[-2].output)
        return vgg_face_descriptor
    def preprocess_image(self,image_path):
        img_vector = preprocess_input(image_path)
        return img_vector
    def findCosineSimilarity(self,source_representation, test_representation):
        a = np.matmul(np.transpose(source_representation), test_representation)
        b = np.sum(np.multiply(source_representation, source_representation))
        c = np.sum(np.multiply(test_representation, test_representation))
        return 1 - (a / (np.sqrt(b) * np.sqrt(c)))
```

Формат	Зона	Поз.	Обозначение	Наименование	Кол.	Приме-чание

					Дипломный проект	Лист 68