

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

Макежанұлы Бауыржан

Жұмыс берушінің тест критерийлері бойынша студенттерді жұмыспен қамту
жүйесін құру

ДИПЛОМДЫҚ ЖҰМЫС

5B070300 – «Ақпараттық жүйелер» мамандығы

Алматы 2022

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы



ҚОРҒАУҒА ЖІБЕРІЛДІ
КАӨЖС кафедрасы меңгерушісі

Т.ғ.к., қауым. профессор

Р.Ж. Сатыбалдиева

« 20 » мамыр 2022 ж.

ДИПЛОМДЫҚ ЖҰМЫС

Тақырыбы: «Жұмыс берушінің тест критерийлері бойынша студенттерді жұмыспен қамту жүйесін құру»

5B070300 – «Ақпараттық жүйелер» мамандығы

Орындаған

Макежанұлы Б.

Рецензент

PhD, қауымдастырылған-профессор

А.К. Мамырова А.К. Мамырова

" 20 " мамыр 2022 ж.

Ғылыми жетекші

Лектор, магистр

С.А. Алдаберген С.А. Алдаберген

" 20 " мамыр 2022 ж.

Алматы 2022

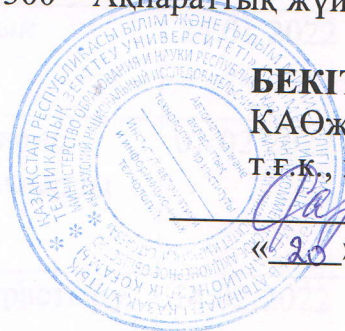
ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

5В070300 – Ақпараттық жүйелер



БЕКІТЕМІН

КАӨЖС кафедрасы меңгерушісі

Т.ғ.к., қауым. профессор

Р.Ж. Сатыбалдиева

« 20 » мамыр 2022 ж.

**Дипломдық жұмысты орындауға
ТАПСЫРМА**

Білім алушы Макежанұлы Бауыржан

Тақырыбы: «Жұмыс берушінің тест критерийлері бойынша студенттерді жұмыспен қамту жүйесін құру»

Университет Ректорының 2021 жылғы «24» желтоқсан №489-П/Ө бұйрығымен бекітілген.

Орындалған жұмыстың өткізу мерзімі « 25 » 05 2022 ж.

Дипломдық жұмыстың бастапқы мәліметтері:

Дипломдық жұмыста қарастырылатын мәселелер тізімі:

1. Зерттеу бөлімі
2. Технологиялар бөлімі
3. Жұмыс моделі бөлімі
4. Тәжірибелік бөлімі

Графикалық материалдардың тізімі (міндетті түрде қажет сызбалар көрсетілген): жұмыстың 24 слайдтан тұратын презентациясы көрсетіледі. Ұсынылған негізгі әдебиет 17 кітаптан тұрады.

Дипломдық жұмысты даярлау
КЕСТЕСІ

Бөлім атауы, қарастырылатын мәселелер тізімі	Ғылыми жетекші мен кеңесшілерге көрсету мерзімі	Ескерту
Дипломдық жұмыс жоспарын құрастыру	24.01.2022	<i>орындау</i>
Жұмыс өзектілігін зерттеу және жұмысқа қажетті технологияларды анықтау	01.02.2022	<i>орындау</i>
Техникалық тапсырманы құрастыру, жұмыс моделін дайындау	14.02.2022	<i>орындау</i>
UML диаграммалары және дерекқор моделі	10.03.2022	<i>орындау</i>
Веб-қосымшаны құру	24.03.2022	<i>орындау</i>
Дипломдық жұмыстың түсіндірме жазбасын аяқтау	09.05.2022	<i>орындау</i>

Дипломдық жұмыс бөлімдерінің кеңесшілері мен норма бақылаушының аяқталған жұмысқа қойған
қолтаңбалары

Бөлімдердің атауы	Кеңесшілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)	Қол қойылған мерзімі	Қолы
Нормалық бақылаушы	Аристомбаева М.Т., Лектор, магистр	<i>20.05.2022</i>	<i>[Signature]</i>
Негізгі бөлім	Алдаберген С.А., Лектор, магистр	<i>20.05.2022</i>	<i>[Signature]</i>

Ғылыми жетекшісі *[Signature]* Алдаберген С.А.

Тапсырманы орындауға қабылдаған білім алушы *[Signature]* Макежанұлы Б.

Күні « 24 » қараша 2021 ж.

АҢДАТПА

Дипломдық жұмыста «JobSearch» веб-қосымшасы құрылған. Веб-қосымша өз мамандығымен жұмыс жасауға ниетті жас мамандар мен оқуын аяқтаған студенттерді өз қатарынан көруді қалайтын компаниямен, яғни жұмыс ұсынушымен байланысын сипаттайды.

Веб-қосымшада жұмыс беруші кадрларды қарай алады, нақты айтқанда жұмыс іздеуші түйіндемесін көріп, қызығушылық танытқан жағдайда өз тарапынан тапсырмаларды ұсына алады. Сәйкесінше жас маман жұмыс беруші тарапынан қойылған барлық критерийлерді орындап, тапсырманы жібереді. Сәтті өткен жағдайда жұмыс берушімен байланысып болашақ жұмысын бастау мүмкіндігіне ие болады.

Бұл жұмыс жұмыссыздық деңгейін төмендетуге өз үлесін қосады. Өз біліктілігіне сенімсіз мамандарды жігерлендіріп, шыңдайды. Кез келген өз ісін енді бастап жатқан компанияларға кадр жетіспеушілік мәселесін шешеді.

Қосымша python тілінің Django фреймворкін пайдаланады. Оны пайдалану оңтайлы, қорғалған, жеңіл, кеңейтуге ығайлы және басқада ерекшеліктері бар. ReactJS кітапханасын фронтенд бөлігіне пайдаланатын болады. Мәліметтер базасы ретінде PostgreSQL таңдалды.

АННОТАЦИЯ

Дипломный проект включает в себя веб-приложение «JobSearch». Веб-приложение описывает взаимоотношения с компанией, т.е. с работодателем, который хочет видеть молодых специалистов и выпускников, желающих работать по специальности.

В веб-приложении работодатель может просмотреть персонал, в частности посмотреть резюме соискателя и, в случае заинтересованности предложить задания. Соответственно, молодой специалист выполняет все критерии, установленные работодателем, и отправляет задание. В случае успеха вы можете связаться со своим работодателем, чтобы начать будущую работу.

Проект поможет снизить уровень безработицы. Поощряет и укрепляет специалистов, не уверенных в своей квалификации. Решает проблему нехватки кадров для компании.

Приложение использует фреймворк Django языка python. Он удобен в использовании, защищен, прост, легко расширяется и имеет другие возможности. Библиотека ReactJS будет использоваться для лицевой части. В качестве базы данных был выбран PostgreSQL.

ANNOTATION

The graduation project includes the «JobSearch» web application. The web application describes the relationship with the company, that is, with the employer who wants to see young professionals and graduates who want to work in their specialty.

In the web application, the employer can view the staff, in particular, view the resume of the applicant and, if interested, offer tasks. Accordingly, the young specialist fulfills all the criteria set by the employer and submits the assignment. If successful, you can contact your employer to start a future job.

The project will help reduce unemployment. Encourages and strengthens specialists who are not confident in their qualifications. Solves the problem of shortage of personnel for the company.

The application uses the Django python framework. It's easy to use, secure, simple, easy to expand, and more. The ReactJS library will be used for the front end. PostgreSQL was chosen as the database.

МАЗМҰНЫ

	Кіріспе	9
1	Зерттеу бөлімі	11
1.1	Жұмыс өзектілігі	11
1.2	Жұмыстың мақсаты	11
1.3	Термин сөздер және анықтамалар	13
2	Технологиялар бөлімі	14
2.1	Python	14
2.2	Django	14
2.3	PostgreSQL	15
2.4	Бағдарламалау ортасы	16
3	Жұмыс моделі бөлімі	17
3.1	Жұмысты модельдеу	17
3.2	Жүйені модельдеу ортасы	17
3.2.1	Use Case Diagram(пайдалану жағдайының диаграммасы)	18
3.2.2	Sequence Diagram(тізбек диаграммасы)	19
3.2.3	Communication Diagram(байланыс диаграммасы)	21
3.2.4	State Machine Diagram(күй машинасының диаграммасы)	23
3.2.5	Collaboration Diagram(ынтымақтастық диаграммасы)	23
3.3	Дерекқорды модельдеу	24
3.3.1	Өзірлеу ортасы	24
3.3.2	ER диаграмма	24
4	Тәжірибелік бөлімі	26
4.1	Веб-қосымша интерфейсі	26
	Қорытынды	36
	Пайдаланылған әдебиеттер тізімі	37
	А қосымшасы. Техникалық тапсырма	38
	Б қосымшасы. Бағдарлама мәтіні	43

КІРІСПЕ

JobSearch веб-қосымшасы жұмыссыз жүрген кез келген маманды, студенттерді мамандығы бойынша жұмыс істеуге және жұмыссыздық деңгейін төмендетуге негізделген.

Еліміздегі көрсетіп отырған жұмыссыздық деңгейін төмендету мақсатында әзірленіп жатқан бұл жұмыс өз нәтижесін келешекте көрсетеді деп ойлаймын. Оқу бітіргеннен кейін көптеген білім алушылар мамандығы бойынша жұмыс істеуден бас тартып жатады. Бұл әрекет өз кезегінде жұмыссыздық көрсеткішінің кері әсеріне алып келеді, яғни деңгейді жоғарлатады. Білім алушылардың өз мандығы бойынша қабілеттілігін көрсетуге және өз білімін көтеруге септігін тигізеді.

Жұмыста қарастырылатын мәселе елдегі кадр тапшылығын азайту және оның алдын алу. Келешекте бұндай мәселенің шешімін табуға ойланып, уақыт жоғалпас үшін мен осы қосымшаны жүзеге асыруды жөн деп таптым. Қосымша бүгінгі таңда өте өзекті мәселе шешімін табудың оңтайлы жолын көрсетеді. Қосымшаны пайдалану жеңіл, ешбір арнайы түсіндірмені талап етпейтін, заман талабына сай, оны қолдану қосымша пайдаланушыларына ешқандай мәселе алып келмейді. Қосымша жұмысын қысқаша қарастырып өтсек.

Қосымша жұмыс беруші мен жұмыс іздеуші арасындағы байланысты сипаттайды. Жұмыс беруші жүйеге сәтті тіркелгеннен кейін бос қызмет орындарын жариялай алады және қызығушылық танытқан жұмыс іздеуші түйіндемесін көре отырып, критерийлері бойынша тапсырма құрастырып оны жүктейді. Тапсырманы сәтті орындаған кадрмен байланыс орнатып алдағы жоспарларын талқылайды.

Өз кезегінде жұмыс іздеуші жүйеге сәтті тіркелгеннен кейін түйіндемесін жариялау мүмкіндігіне ие болады. Бос қызмет орындарын көре алады, егер талаптарға сәкес болса өз өтінішін қалдыра алады. Кейін жұмыс беруші тарапынан жолданған тапсырмаларды орындап, қайта жүктейді. Сәтті болған жағдайда өз ісін бастауға мүмкіндік алады. Егер сәтсіз болса кейіннен қайта байқап көруіне мүмкіндігі болады.

Жұмыс беруші өзінің тапсырмаларын құрастырып, оларға керек деп тапқан сұрақтарын ойластырып, критерийлерін анықтап және өз тапсырмасын бөлісе алуын қадағалайды. Менің ойымша тапсырмаларды құрастырып онымен бөлісуге жұмыс берушінің өз еркіне қалдырған дұрыс деп таптым. Сенімді түрде өзі құрастырған тапсырманың нәтижесін көре отырып жұмыс іздеген маманның қабілетін бағалай алады. Тест тапсырмаларын құрастырып оны жүкеуде жұмыс беруші екі түрлі жолмен өз тапсырмасын жұмыс іздеушіге жібере алады.

Жұмыс берушінің тапсырмалармен бөлісу жолдары:

- қосымшада қолданушылар (жұмыс іздеуші мен жұмыс беруші) арасында байланыс орнағаннан кейін тапсырмаларды сілтеме арқылы жұмыс беруші тарата алады;
- екінші жолы жұмыс іздеушіге жеке электрондық почта көмегімен өз тапсырмасын жібере алады.

Жұмыс берушінің тапсырмаларын құрастыру ортасы ретінде кез-келген өзіне қолайлы ортаны пайдалануды ұсынамыз. Бұндай орта өте көптеп кездеседі. Ашық сұрақ немесе жылдамдыққа тапсырмаларды орындайтын ортаны жасау үшін жаңа бағдарлама құрастыруды дұрыс шешім деп есептемеймін. Сол себепті дайын формаларды пайдалану қолданушы үшін дұрыс шешім болары анық.

Жұмыс толық сұранысқа ие болған сәттен бастап еліміздегі жұмыссыздық мәселесін шешуге оң нәтижесін тигізеді. Білікті мамандардың қатарын арттырып, бәсекеге қабілетті жұмыс орындарының санының артуына өз септігін тигізеді. Бұл жұмыс бүгінгі кезде сұранысқа ие. Осы себептен бұл жұмыстын өзекті деп шештім.

1 Зерттеу бөлімі

1.1 Жұмыстың өзектілігі

Еліміздегі жұмыссыз жүргендердің көрсеткіші 4,9%-ды көрсетуде[1]. Өткен жылдың зерттеу жұмыстары көрсеткен нәтиже бойынша шамамен 450 мың адам жұмыссыз. Жұмыс өзектілігі жұмыссыздық деңгейін төмендету болып табылады.



1-сурет – Қазақстандағы жұмыссыздық көрсеткіші "Қазақстан" ұлттық арнасының фейсбук парақшасынан

Бұл жұмыс жас мамандар мен оқуын аяқтаған студенттерді өз мамандығымен жұмысқа орналасу мүмкіншілігін арттыру мақсатында құрастырылуда. Өз біліміне сеніміз жастарды жігерлендіріп, құлшынысын арттыруға және жұмыспен қамтамасыз етуге негізделген қосымша.

Қосымшаның негізгі қолданушылары – өз мамандығымен жұмыс жасауды қалайтын жас мамандар мен компаниялар болып табылады.

Негізгі бағдарлама python тілінің Django фреймворкін пайдаланады[9]. Оны пайдалану оңтайлы, қорғалған, жеңіл, кеңейтуге ығайлы және басқада ерекшеліктері бар. ReactJS кітапханасын фронтенд бөлігіне пайдаланатын болады[10].

1.2 Жұмыстың мақсаты

Жұмыс мақсаты жоғарыда айтылғандай жұмыссыздық мәселесін шешу негізгі мақсаты. Жұмыс ұсынушы компания мен жұмыс іздеген маман арасын байланыстыратын веб-қосымша.

Веб-қосымша міндетті түрде қамтиды:

- авторизация және жүйеге тіркелу;
- қосымша пайдаланушыларын топтастыру;
- деректермен алмасу;
- қосымша дизайны.

Қосымша пайдаланушылары келесідей топтарға бөлінеді:

- administrator (деректерді басқару);
- жұмыс іздеуші (жүйеге тіркелу, түйіндеме жариялау және өңдеу, бос жұмыс орындарын қарау, тапсырмаларды орындау);
- жұмыс беруші (жүйеге тіркелу, бос жұмыс орындарын жариялау және өңдеу, түйіндеме қарау, маманды шақыру, тапсырмаларды жүктеу);
- the guest (материалдарды оқу, бос жұмыс орындарын қарау).

Жүйеге тіркелген қолданушы кез келген мүмкіндікті пайдалана алады. Қосымшаны толықтай қолдана алады. Ал, жүйеге тіркелмеген қолданушы жүйенің тек белгілі қол жетімді мүмкіндіктерін ғана пайдаланады.

«JobSearch» қосымшасы – жас мамандарды, студенттерді, өз мамандығы бойынша жұмыс жасауды қалайтын жандарға өз қалауы бойынша жұмысқа орналасуға көмектесу құралы болмақ.

Бұл қосымша жұмыс беруші мен оны іздеген маман арасында байланыс орнатып, қажеттілігін орындауға жағдай жасайды.

Негізгі ерекшелігі жұмыс берушінің жас мамандарға қызығушылығы. Жұмыс іздеген маман өз парақшасын толтырған соң, жұмыс беруші тарапынан арнайы сұрақтарды шешу керек, егер тиесілі балды жинап сәтті өткен болса маман жұмыс беруші тарапынан шақырту алады. Бұл өз кезегінде жас маманның өзі қалайтын ісімен айналысуына таптырмас мүмкіндік сыйлайды.

Құжат жұмыстың жай-күйін сипаттайды, онда жұмысты жүзеге асыру кезінде ескеру қажет барлық ақпараттар қамтылады, жұмыс мақсаты мен жұмысқа қойылатын талаптар жазылады.

Құжатта келтірілген мәліметтер негізінде сайттың тұжырымдамасын әзірлеу жұмыстары жүргізілуде. Құжатты бекіткеннен кейін өзгерістер енгізу қосымша бекітуді талап етеді.

Халқымыздың жарқын болашағы үшін көптеген компаниялар мен мамандарды біріктіреміз.

1.3 Термин сөздер мен анықтамалар

Пәндік аймақта және жұмысты модельдеу барысында пайдалнатын термин сөздер мен олардың анықтамалары 1-кестеде көрсетілген.

1-кесте - Термин сөздер мен анықтамалар

Аббревиатура немесе термин	Анықтама
Django	Веб-қосымша фреймворкі
HTML	Гипермәтінді белгілеу тілі
ReactJS	Фронтенд үшін қолданылатын кітапхана
JS	Көп парадигмалық бағдарламалау тілі
JSON	JavaScript негізіндегі мәтіндік деректермен алмасу үлгісі
JWT	Қол жеткізу токендерін құрастыру
API	Қосымшаның бағдарламалау интерфейсі
NodeJS	Javascript-ке арналған серверлік платоформа
npm	Пакет менеджері
nrx	Тәуелділіктерді орнатуды және басқаруды қадағалайды
DB	Деректер қоры
PostgreSQL	Объектті-реляциялық ДҚБЖ
UML	Біріңғай модельдеу тілі
MVT	Model-View-Template Django құрылымының архитектуралық үлгісі

2 Технологиялар бөлімі

2.1 Python

Python - әртүрлі типтегі қосымшаларды жасауға арналған кең қолданыстағы жоғары деңгейдегі бағдарламалау тілі. Бұл веб-қосымшалар, ойындар және деректер қорымен жұмыс жасайды. Python машиналық оқыту және жасанды интеллект саласында кеңінен қолданылады[7].

Python бағдарламалау тілін алғаш рет 1991 жылы голландиялық әзірлеуші Гвидо Ван Россуммен ойлап тапқан. Сол уақыттан бері бағдарламалау тілі үлкен даму жолынан өтті. Тілдің ағымдағы нұсқасы 2021 жылдың қазан айында шыққан 3.10 болып табылады.

Python бағдарламалау тілінің басты мүмкіншіліктері:

- сценарий тілі. Бағдарлама коды сценарийлер түрінде анықталады.
- бағдарламалау парадигмаларының кең ауқымын, соның ішінде объектіге бағытталған және функционалдық парадигмаларды қолдау.
- бағдарламаны түсіндіру. Сценарийлермен жұмыс істеу үшін сценарийді іске қосатын және орындайтын аудармашы қажет.

2.2 Django

Django – жылдам ойлап табуға және таза прагматикалық дизайнға ықпал ететін жоғары деңгейлі Python веб-фреймворкі[8]. Тәжірибелі әзірлеушілер жасаған бұл фреймворк көп қиындықтарды алып тастайды, осылайша жаңа дүниені ойлап таппай-ақ қосымшаңызды жазуға назар аудара аласыз.

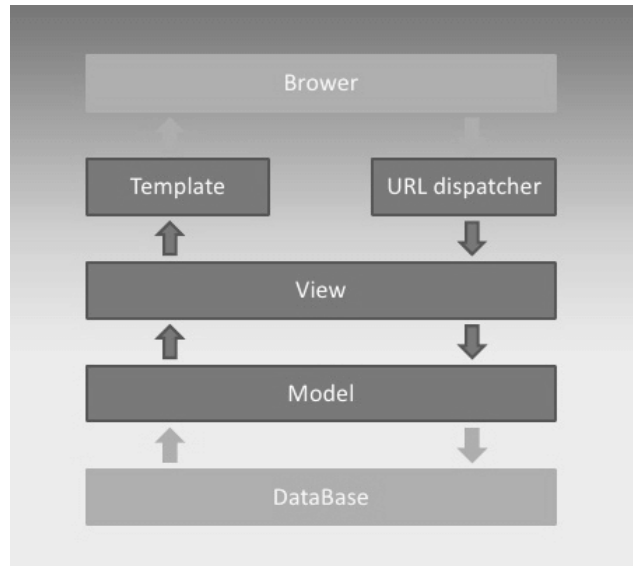
Django үлкен сұранысқа ие. Django 2005 жылы веб-қосымшаларды құру тілі ретінде Python тілін пайдалана бастаған кезде құрылды. 2008 жылы фреймворктің алғашқы шығарылымы(релиз) шықты. Қазіргі кезеңде қарқынды даму үстінде. Ағымдағы нұсқасы 2021 жылдың желтоқсанында шыққан 4.0 нұсқасы болып табылады.

Django келесідей қолдану артықшылығына ие:

- жылдам. Жобаның басталуы мен аяқталуына дейінгі аралықты мейлінше тезрек құруға арналған.
- ауқымды. Қосымшаны тәуелсіз бөліктерге бөледі. Бұл жылдамдық пен икемділік береді.
- жан-жақты. Қолдану аймағы ауқымды. Компаниялар әлеуметтік желіден ғылыми есептеу платформаларына дейін қосымшалар құра алады.
- толық жүктелген. Жалпы веб-әзірлеу мәселелерін шешу үшін пайдалануға болатын ондаған толықтыруларды қамтиды.

- сенімді қауіпсіз. Қауіпсіздікке өте мұқият қарайды. Көптеген шабуыл түрлерінен қорғалған. Тіркеу деректерін сақтаудың қауіпсіз түрін пайдаланады.

Django құрылымы Model-View-Template архитектуралық үлгісін немесе қысқаша MVT-ді жүзеге асырады. Негізі бұл Model-View-Controller үлгісінің модификациясы болып табылады. Келесі түрде бейнеленеді.



2-сурет – MVT архитектурасы

Негізгі элементтері:

- url dispatcher: Сұрау кезінде URL негізінде қай ресурс осы сұрауды өңдеу керектігін анықтайды.
- model: Қосымша қолданылатын деректерді сипаттайды.
- view: Сұрауды қабылдайды, оны өңдейді және пайдаланушыға жауап қайтарады. Егер сұрауды өңдеу Model және DB кіруді қажет етсе, онда View олармен әрекеттеседі.
- template: Жасалған html белгілеу ретінде көрсетілім логикасын көрсетеді.

2.3 PostgreSQL

PostgreSQL - ең танымал деректер қорын басқару жүйелерінің бірі. Негізінен шығу тарихы Ingres жобасымен байланысты. Даму тарихы 1986 жылдан бастау алады. Ол кезде POSTGRES деп аталды. Ал 1996 жылы жоба PostgreSQL деп өзгертілді, осы жылдың 8 шілдесінде өнімнің алғашқы шығарылымы шықты. Көптеген нұсқасы жарық көрген, ағымдағы нұсқасы 14 нұсқа.

PostgreSQL негізгі командалары[13]:

- alter database – дерекқор атрибуттарын өзгерту
- alter foreign table – кестенің анықтамасын өзгерту
- alter function – функцияның анықтамасын өзгерту
- alter group – рөл атауын өзгерту
- alter index – индекс анықтамасын өзгерту
- alter language – процедуралық тілдің анықтамасын өзгерту
- alter table – кесте анықтамасын өзгерту
- alter type – түр анықтамасын өзгерту
- alter user – дерекқордағы рөлді өзгерту
- alter view – көрініс анықтамасын өзгерту
- create database – мәліметтер қорын құру
- create function – функция құру
- create group – дерекқорда рөл құру
- create index – индекс құру
- create language – процедуралық тілді құру
- create type – жаңа деректер түрін жасау
- create user – мәліметтер базасында рөлді құру
- create view - көрініс жасау
- delete – кесте жазбаларын жою
- drop function – функцияны жою
- drop group – дерекқордағы рөлді жою
- drop index – индексті жою
- truncate – кестені немесе кестелер жинағын босату
- update – кесте жолдарын өзгерту
- select – кестеден немесе көріністен жолдарды алу

PostgreSQL артықшылықтары:

- шектеусіз көлемдегі мәліметтер базасын қолдау;
- қуатты және сенімді транзакция және репликация механизмдері;
- кірістірілген бағдарламалау тілдерінің кеңейтілген жүйесі және C-үйлесімді модульдерді жүктеуге қолдау көрсету;
- мұрагерлік;
- оңай кеңейту мүмкіндігі.

2.4 Бағдарламалау ортасы

PyCharm Professional Edition — Python бағдарламалау тіліне арналған әзірлеу ортасы. Кодты қатесіз жазуға, кодты анализдеуге, сапалы жұмыс жасауға және Django веб-фреймворкімен жұмыс жасауға мүмкіндік береді. PyCharm - IntelliJ IDEA негізінде JetBrains компаниясы жасаған. Алғашқы нұсқасы 2010 жылы шыққан[7].

3 Жұмыс моделі бөлімі

3.1 Жұмысты модельдеу

Жұмысты модельдеу құрылғалы жатқан жүйе жайлы барлық ақпаратты визуалды түрде көрсетеді. Сенімді жүйе құруда нақты нәтижеге жету үшін бірінғай модельдеу түрі қолданылады.

UML(бірінғай модельдеу тілі) - бұл объектіге бағытталған бағдарламалауды талдау, жұмысты модельдеу және жүйені жобалауға арналған графикалық сипаттау тілі[4].

UML диаграммаларының 14 түрі бар. Негізінен екі топқа бөлуге болады. Құрылым диаграммалары және мінез-құлық диаграммалары.

Құрылым диаграммалары:

- class diagram(Сынып диаграммасы);
- component diagram(Компонент диаграммасы);
- deployment diagram(Орналастыру диаграммасы);
- object diagram(Объект диаграммасы);
- package diagram(Пакет диаграммасы);
- profile diagram(Профиль диаграммасы);
- composite structure diagram(Құрама құрылым диаграммасы).

Мінез-құлық диаграммалары:

- use case diagram(Пайдалану жағдайының диаграммасы);
- activity diagram(Белсенділік диаграммасы);
- state machine diagram(Күй машинасының диаграммасы);
- sequence diagram(Тізбек диаграммасы);
- communication diagram(Байланыс диаграммасы);
- interaction overview diagram(Өзара әрекеттестікке шолу диаграммасы);
- timing diagram(Уақыт диаграммасы).

Бұл жұмыста мінез-құлық диаграммасының 5 түрі пайдаланылды.

- use case diagram (пайдалану жағдайының диаграммасы);
- sequence diagram (тізбек диаграммасы);
- communication diagram (байланыс диаграммасы);
- state machine diagram (күй машинасының диаграммасы);
- collaboration diagram (ынтымақтастық диаграммасы).

3.2 Жүйені модельдеу ортасы

Жүйені модельдеу ортасы ретінде Creately қосымшасы пайдаланылды[5]. Қолдануға ыңайлы сонымен қатар қолдану мүмкіндігін

арттыру мақсатында бейне оқулық қамтылған. Оны толығымен көргеннен кейін ешқандай қиындықсыз қолдана аламыз.

Creately — Cinergix әзірлеген көрнекі диаграмма және дизайн құралы. Creately екі нұсқасы бар: онлайн нұсқасы және Windows, Mac және Linux жүйелерімен үйлесімді жүктелетін желіден тыс жұмыс үстелі нұсқасы. Қосымша негізінен блок-схемаларды, ұйымдастыру диаграммаларын, жоба диаграммаларын, UML диаграммаларын, ақыл-ой карталарын және басқа бизнес көрнекілігін жасау үшін қолданылады.

Заманауи үлгіде, ерекше дизайнда, жақсы сапада визуалды бағытта жұмыс жасауды пайдаланатын болсаңыз Creately қосымшасын пайдалануды кеңес етемін.

3.2.1 Use Case Diagram(пайдалану жағдайының диаграммасы)

Use Case Diagram(пайдалану жағдайының диаграммасы) – UML мінез-құлық диаграммасының бірі[6]. Графикалық түрде жүйе қатысушылары(actor) мен прецеденттер(use case) арасындағы қарым-қатынасты суреттейді. Диаграмманың негізгі элементтері қатысушы(actor) және прецедент(use case) болып табылады.

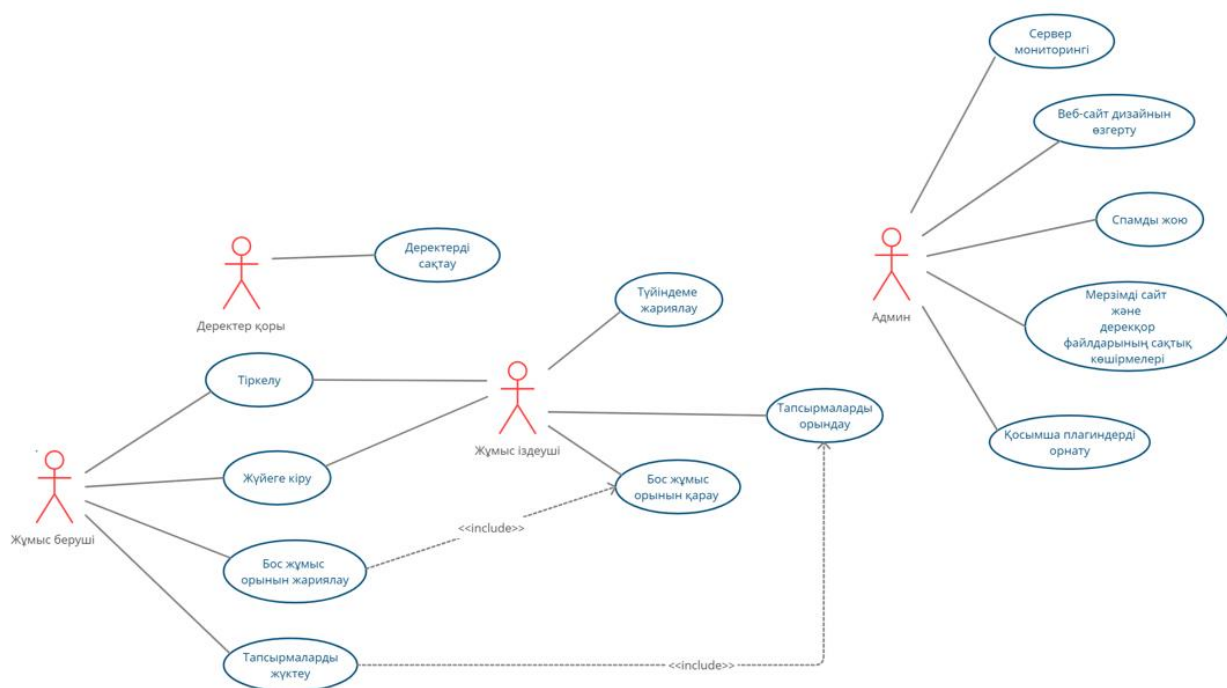
Қатысушы(actor) – прецеденттермен байланыс кезінде орындалатын логикалық рөлдер. Тек адам ғана емес сонымен қатар кез келген жүйе қатысушы бола алады. Actor адам бейнесінде көрсетіледі.

Прецедент(use case) – қатысушымен байланысты іс әрекет жиынтығы. Графикалық бейнесі эллипс түрінде белгіленеді.

Төмендегі 3.1-суретінде веб-қосымшаның пайдалану жағдайының диаграммасы көрсетілген.

- диаграммада 4 қатысушы(actor) бар:
- жұмыс беруші;
- жұмыс іздеуші;
- админ;
- деректер қоры.

Әр қатысушының(actor) жүзеге асыратын прецеденттері(use case) көрсетілген.



3.1-сурет – Пайдалану жағдайының диаграммасы (Use case diagram)

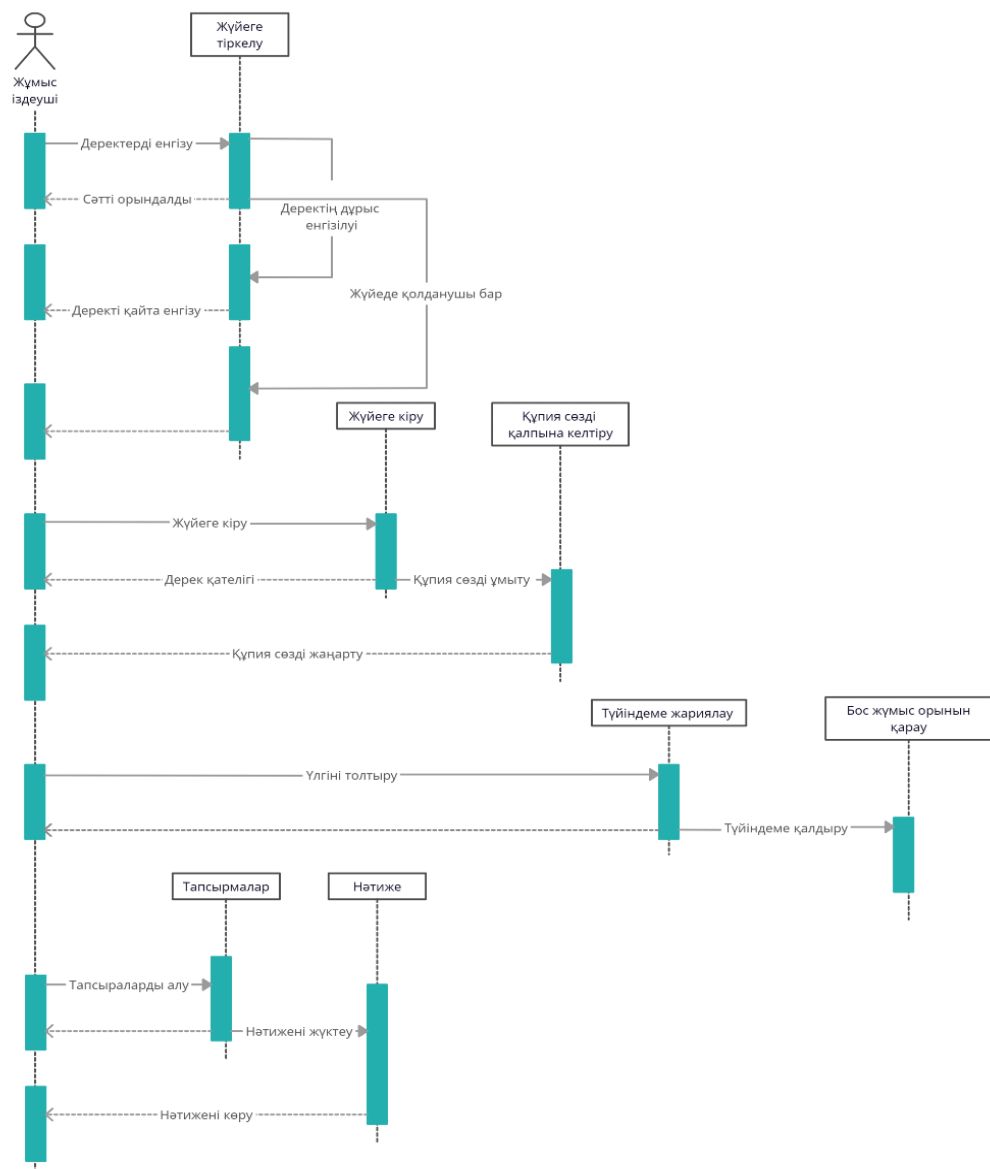
3.2.2 Sequence Diagram(тізбек диаграммасы)

Sequence Diagram (тізбек диаграммасы) – бұл пайдалану жағдайларының диаграммаларын нақтылау үшін қолданады[6]. Пайдалану жағдайының логикасын егжей-тегжейлі сипаттайды.

Негізгі элементтері:

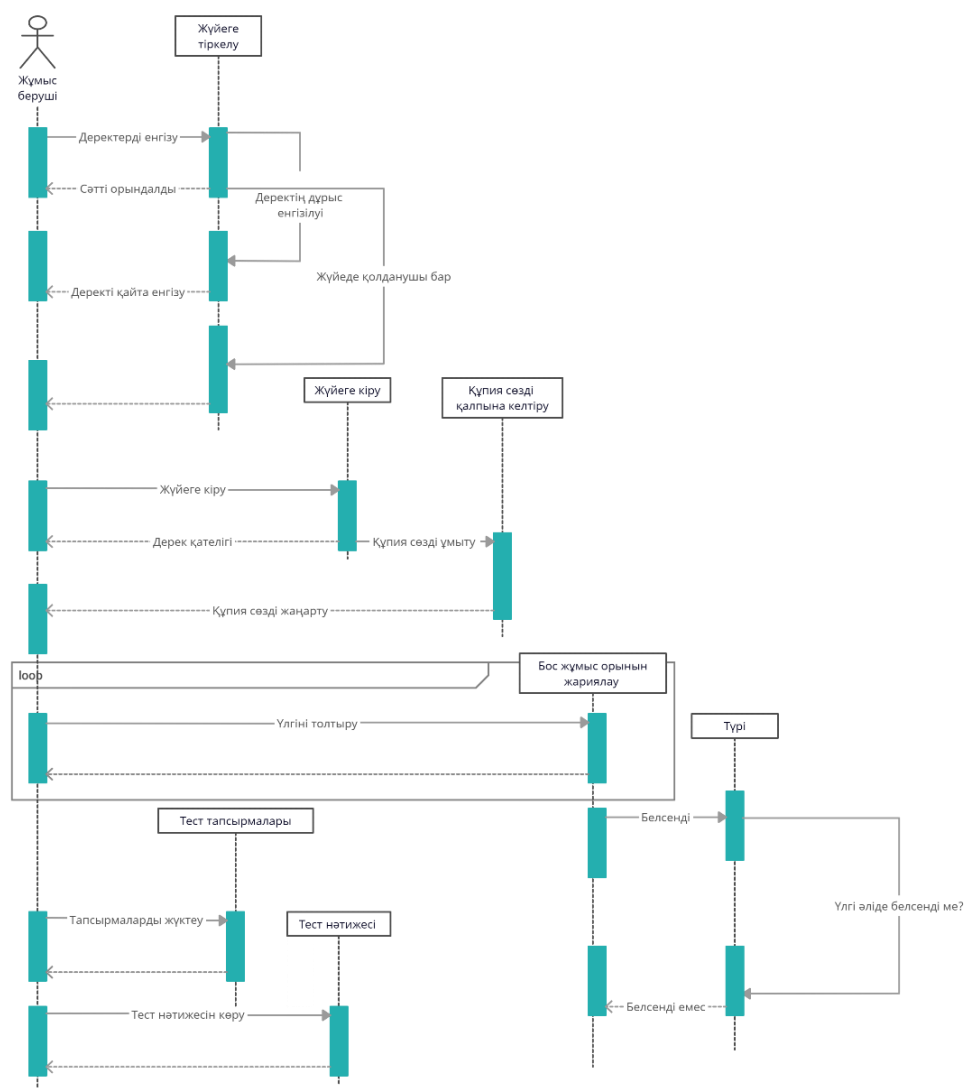
- объекті - тіктөртбұрыштармен белгіленеді;
- хабарлама – бағыты бар сызықтар;
- қайтарылған нәтижелер бағыты бар үзік сызықтар.

Тізбек диаграммасы жұмыста екі бөлікке бөлінген. Бірінші бөлікте жұмыс іздеуші үшін құрастырылған, оны 3.2-суретте көрсетілген.



3.2-сурет – Жұмыс іздеушінің тізбек диаграммасы

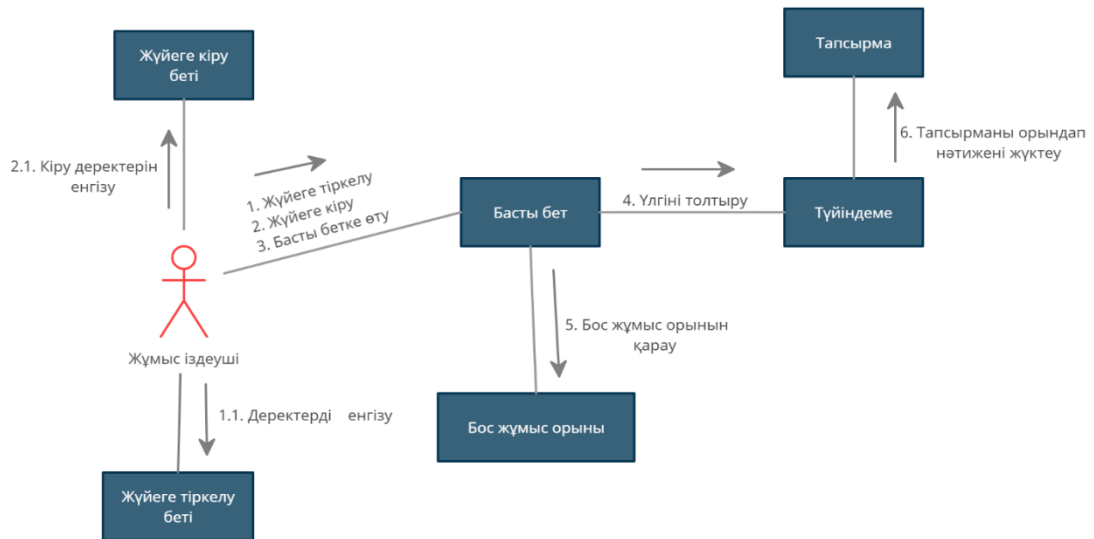
Екінші бөлікте жұмыс беруші үшін тізбек диаграммасы сипатталған, оны 3.3-суретте көрсетілген.



3.3-сурет – Жұмыс берушінің тізбек диаграммасы

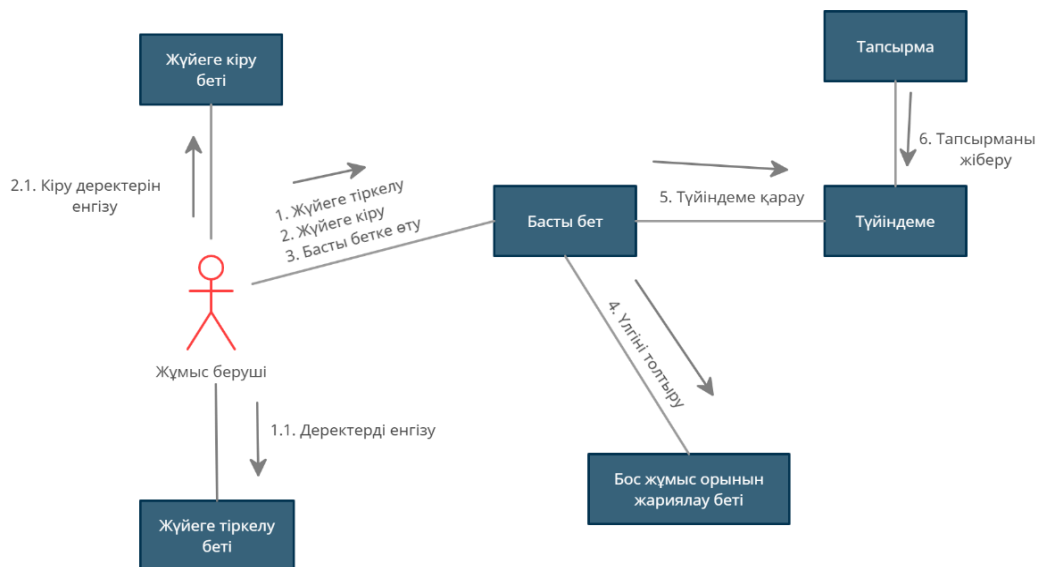
3.2.3 Communication Diagram(байланыс диаграммасы)

Communication Diagram(байланыс диаграммасы) - тізбек диаграммаларына ұқсас, бірақ басты назар объектілер арасында жіберілетін хабарламаларға аударылады[6]. Жұмыста байланыс диаграммасын екі бөлік түрінде сипатталған. Бірінші бөлігінде жұмыс іздеушінің байланыс диаграммасы сипатталған, оны 3.4-суреттен көрсетілген.



3.4-сурет – Жұмыс іздеушінің байланыс диаграммасы

Екінші бөлігінде жұмыс берушінің байланыс диаграммасы сипатталған, оны 3.5-суреттен көруге болады.

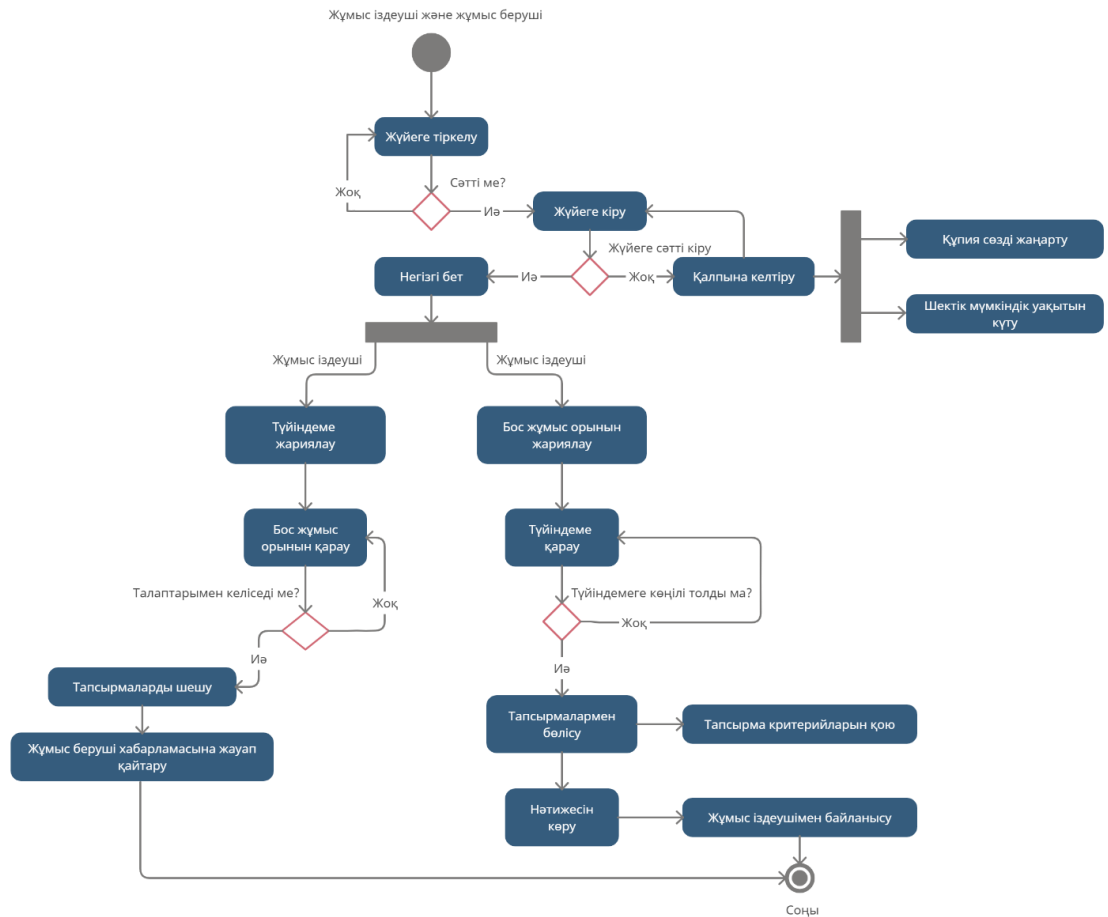


3.5-сурет – Жұмыс берушінің байланыс диаграммасы

3.2.4 State Machine Diagram(күй машинасының диаграммасы)

State Machine Diagram (күй машинасының диаграммасы) – бұл объектінің бір күйден басқа бір екінші күйге өту процесін сипаттайды[6]. Бұл диаграмма объектінің өмірлік циклін модельдеу үшін пайдалы.

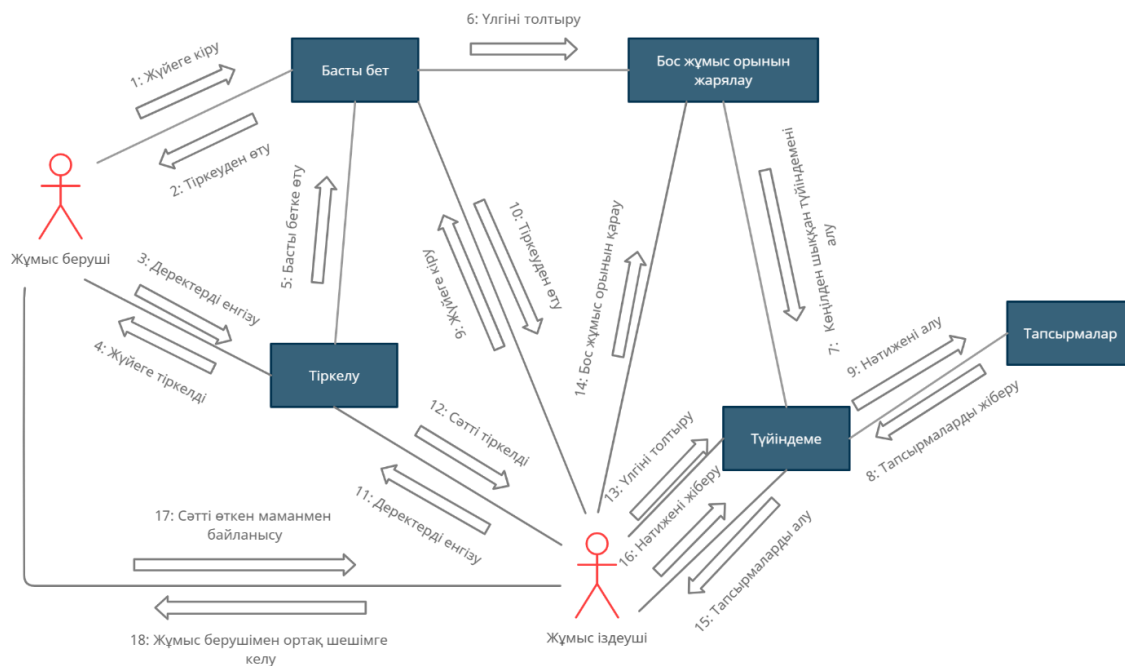
Жұмыс беруші мен жұмыс іздеушінің күй диаграммасы 3.6-суретте нақты көрсетілген.



3.6-сурет – Жүйенің күй диаграммасы

3.2.5 Collaboration Diagram(ынтымақтастық диаграммасы)

Collaboration Diagram (ынтымақтастық диаграммасы) - бұл өзара әрекеттесетін объектілердің жиынын модельдеуге арналған. Ынтымақтастық диаграммасы кейбір функцияларды көрсету үшін бірге пайдаланылатын өзара әрекеттесетін рөлдер жинағын анықтайды. 3.7-суретте жүйенің ынтымақтастық диаграммасы келтірілген[6].



3.7-сурет – Жүйенің ынтымақтастық диаграммасы

3.3 Дерекқорды модельдеу

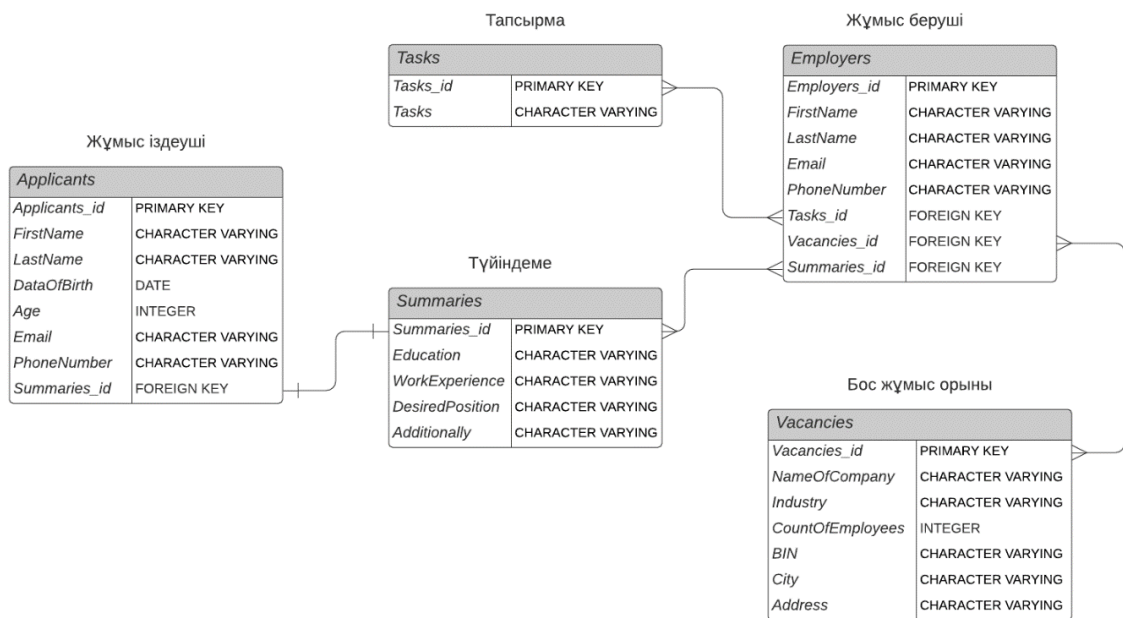
3.3.1 Бағдарламалау ортасы

Дерекқор моделін әзірлеу ортасы – Lucidchart веб-қосымшасы. Жұмысты әзірлеу барысында қолданылатын диаграммалар мен кестелерді визуалды түрде ұсынуға арналған, жоғарғы деңгейде жасалған орта. Қолдану өте оңай, арнайы үлгілерді немесе бастан аяқ өзіңе диаграмма немесе басқада визуалды әрекеттерді орындауға мүмкіндік береді.

Бұл жұмыста Lucidchart көмегімен дерекқор моделін құру үшін ER(Entity-Relationship) диаграмма пайдаланылады[12].

3.3.2 ER диаграмма

ER(Entity-Relationship) диаграмма – бұл пәндік аймақтың схемаларын көрсетуге мүмкіндік сыйлайтын деректер моделі болып табылады[12]. Оның көмегімен жоғары деңгейлі(концептуалды) деректерқорынның жобасын жасауда қолданылады. ER диаграмма көмегімен мәліметтер қорының негізгі нысандарын бөліп қарастыруға болады және олардың арасындағы байланысты нақты сипаттап көрсетуге болады. Дерекқор моделінің ER диаграммасы 3.8-суретте көрсетілген.

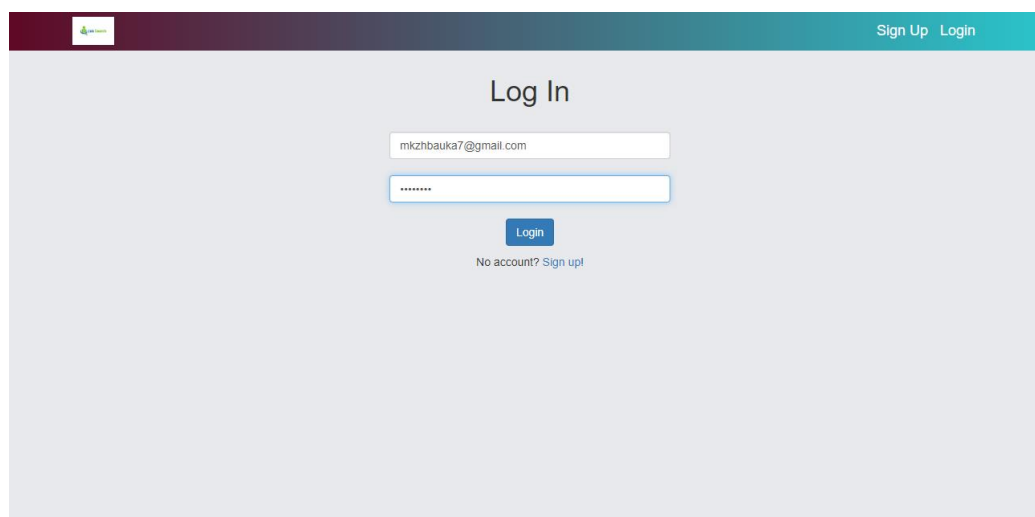


3.8-сурет – Дерекқор моделінің ER диаграммасы

4 Тәжірибелік бөлімі

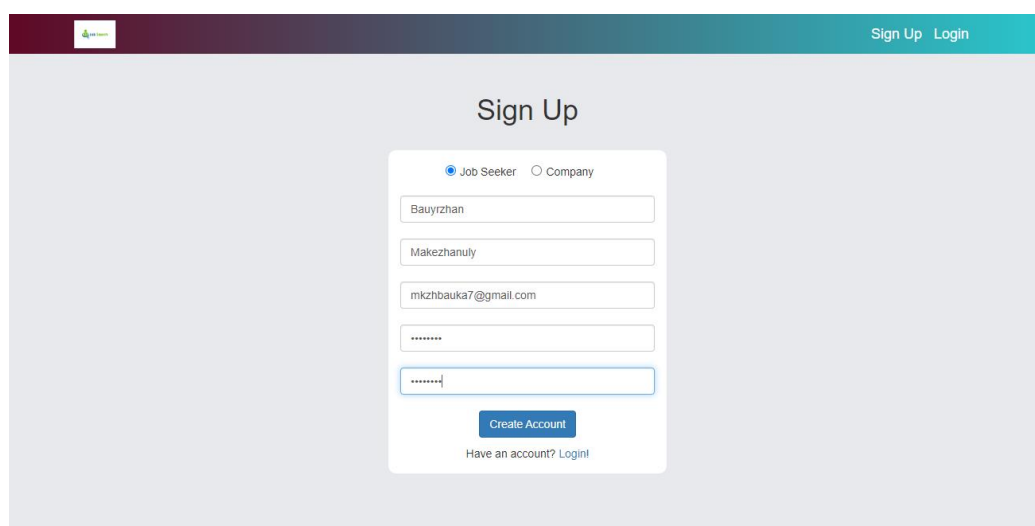
4.1 Веб-қосымша интерфейсі

Веб-қосымшаның алғашқы беті авторизация және жүйеге тіркелу беттерінен тұрады. 4.1 және 4.2-суретте көрсетілген. Жүйеге кіру деректерін енгізгеннен кейін негізгі бетке өтеді.



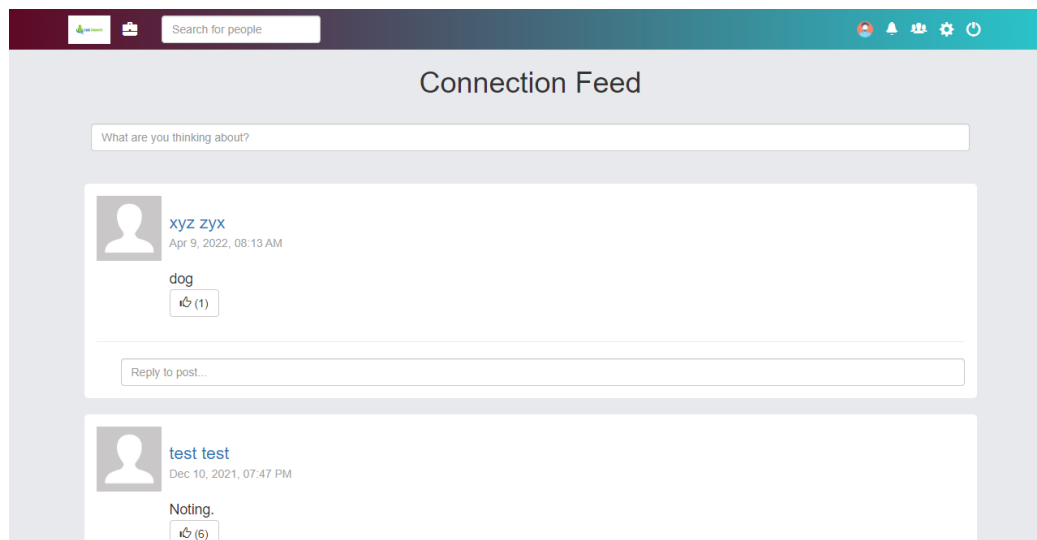
4.1-сурет – Авторизация беті

Жүйеге тіркелу деректері 4.2-суретте көрсетілген. Қолданушы өзінің жұмыс іздеуші немесе жұмыс беруші екендігін көрсетуі тиіс.



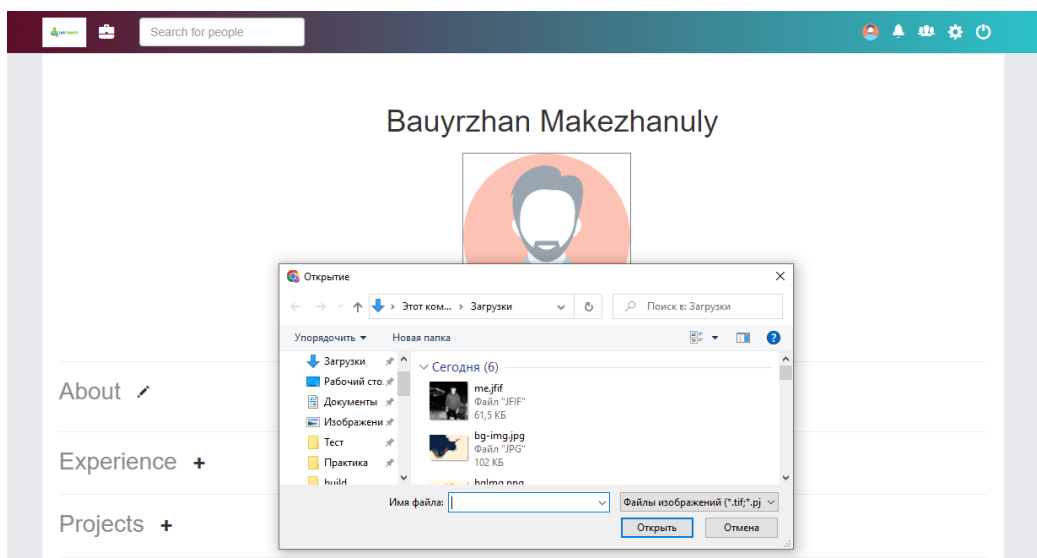
4.2-сурет – Жүйеге тіркелу беті

Жүйеге тіркеуден өтіп, кіргеннен кейін негізгі бетке өтеді 4.3 суретте көрсетілген. Ол жерде жүйе тіркелушілерінің салған жазбаларын көруге болады.

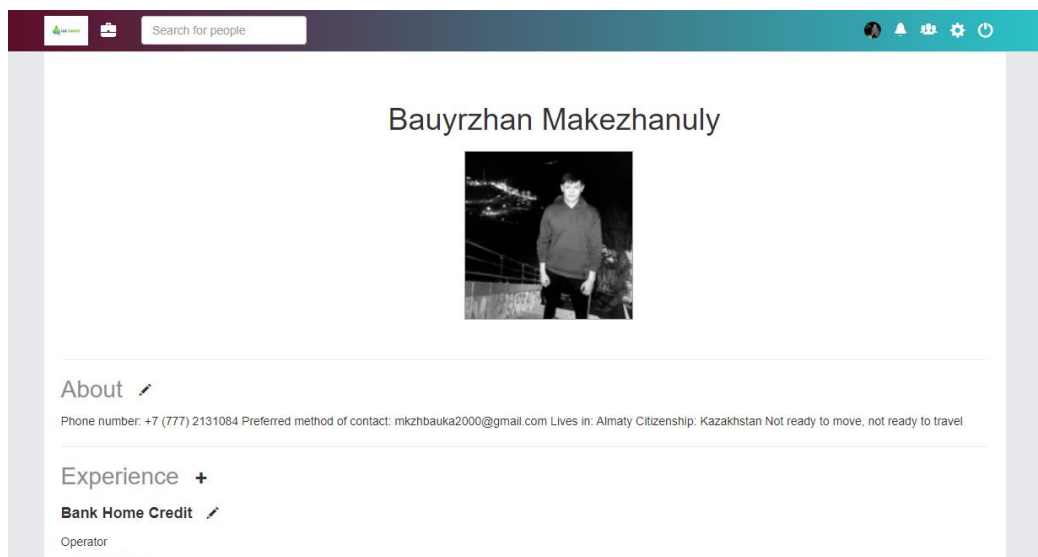


4.3-сурет – Жүйенің негізгі беті

Келесі кезекте жүйе тіркелушісінің өз түйіндемесін жариялау беті. Негізгі деректерді толтыра отырып өз түйіндемесін құра алады. 4.4, 4.5, 4.6 – суреттерде көрсетілген.

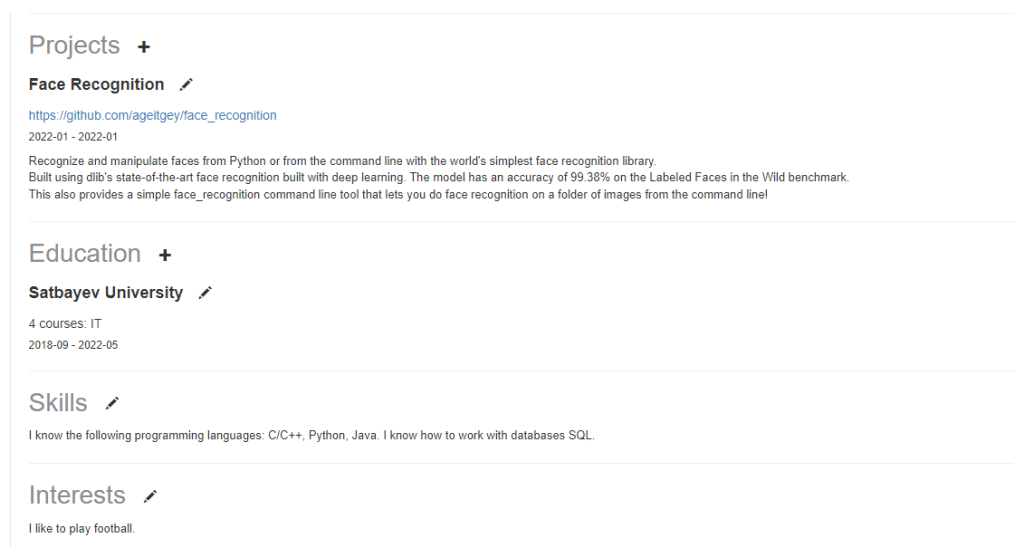


4.4-сурет – Түйіндеме жариялау беті



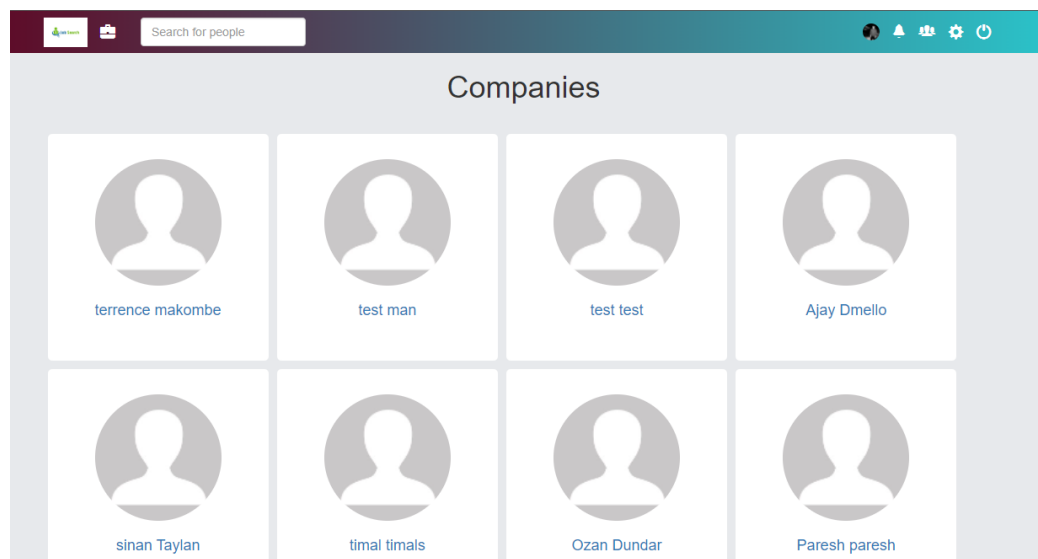
4.5-сурет – Түйіндеме жариялау беті

4.6-суретте түйіндемеде нақты қандай деректерді толтыру керектігі берілген. Осы деректерді толтыра отырып қолданушы өзі жайлы барлық ақпаратпен бөлісе алады.



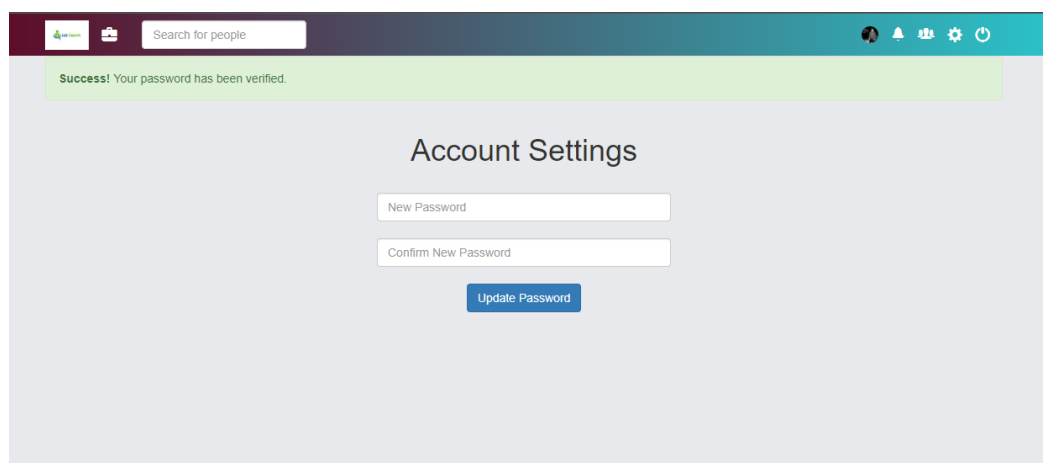
4.6-сурет – Түйіндеме жариялау беті

Жүйе қолданушысы компаниялар тізімін көре алады және өзіне ұнаған бос жұмыс орынына өтінім қалдыра алады. 4.7 – суретте көрсетілген.



4.7-сурет – Бос жұмыс орындарының тізімі

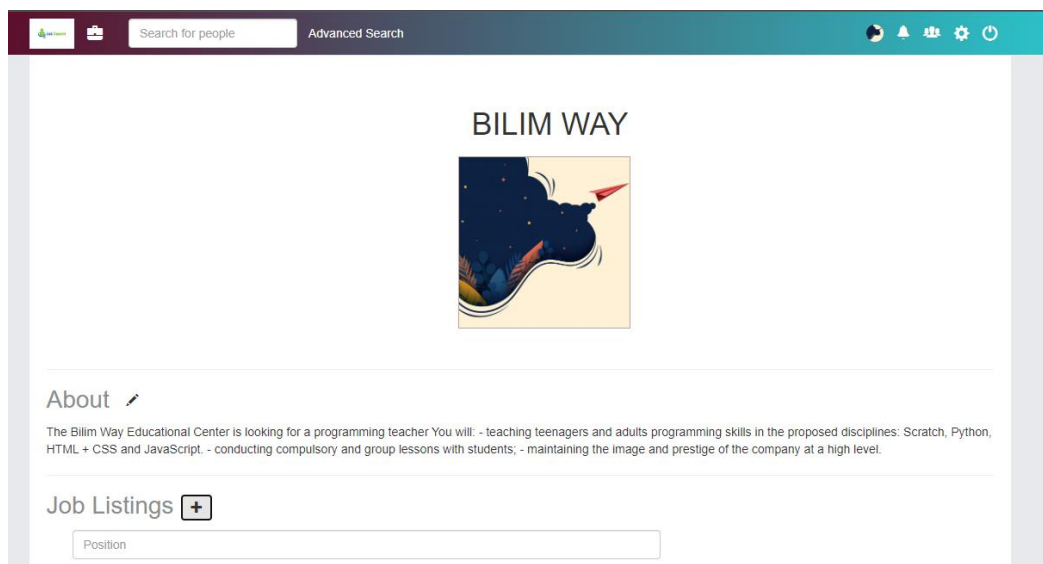
Жүйе қолданушысы өз деректерін өзгерте алады соның бірі құпия сөзді жаңарту оны 4.8-суреттен көруге болады.



4.8-сурет – Құпия сөзді жаңарту

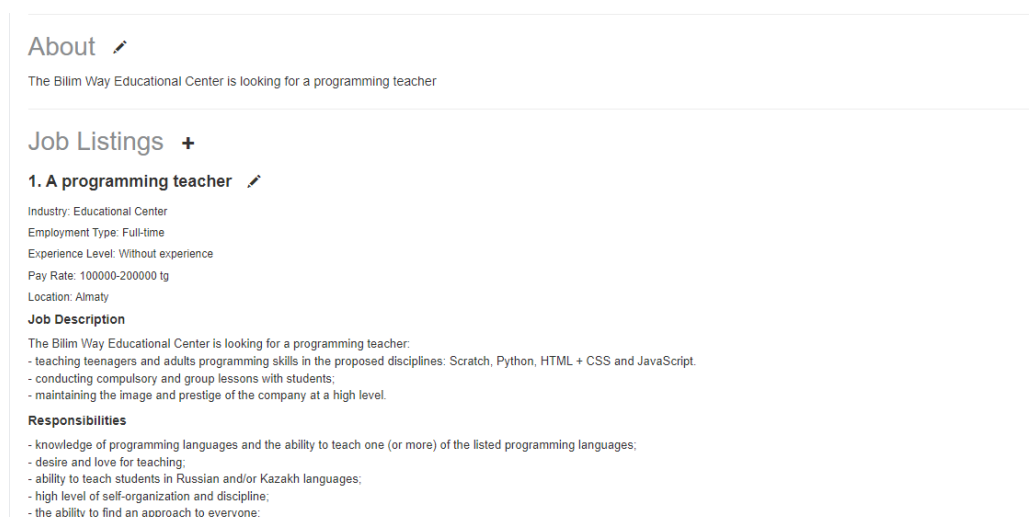
Жұмыс беруші жүйге тіркегеннен кейін бос жұмыс орынын жариялау, жұмыскерлерді іздеу және олармен байланыс орнату амалдарын орындай алады. 4.9 және 4.10 – суретте бос жұмыс орынын жариялау көрсетілген.

4.9 – суретте бос жұмыс орынын жариялау үшін қандай деректерді толтыру керектігі көрсетілген.



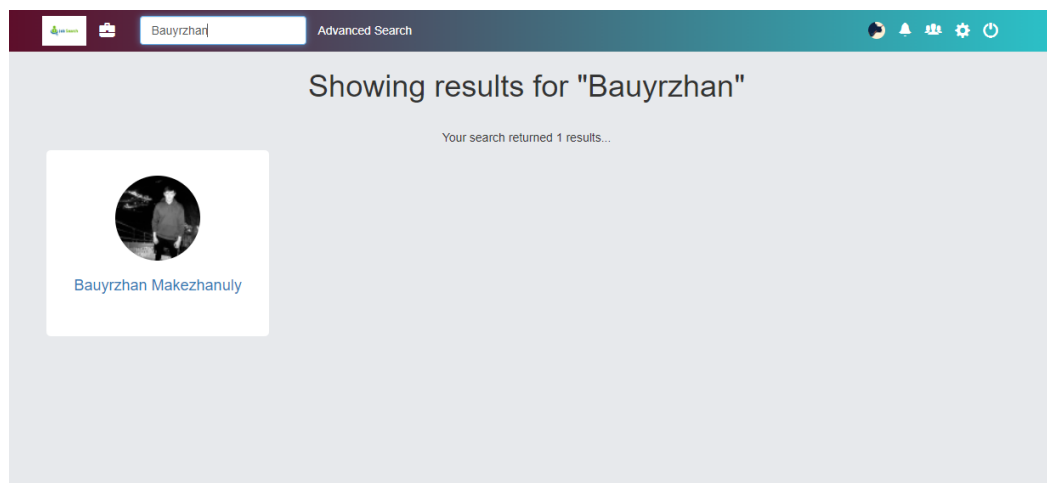
4.9-сурет – Бос жұмыс орынын жариялау

Төмендегі суретте бос жұмыс орыны жайлы барлық ақпарат көрсетілген. Қандай маманды іздейтіндігі, немен айналысатыны келтірілген.



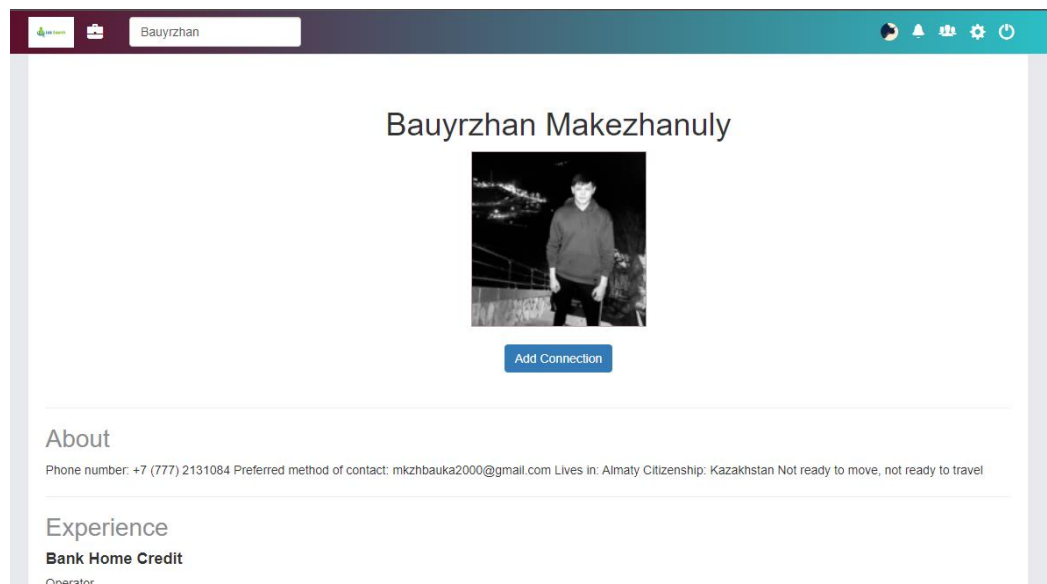
4.10-сурет – Бос жұмыс орыны жайлы ақпарат

Жұмыскерді іздеу беті 4.11-суретте көрсетілген және онымен байланыс орнату 4.12-суретте, ал жұмыс берушімен кері байланыс орнату 4.13-суретте көрсетілген. 4.14-суретте байланыс орнатқан жұмыс іздеушілер тізімі көрсетілген.

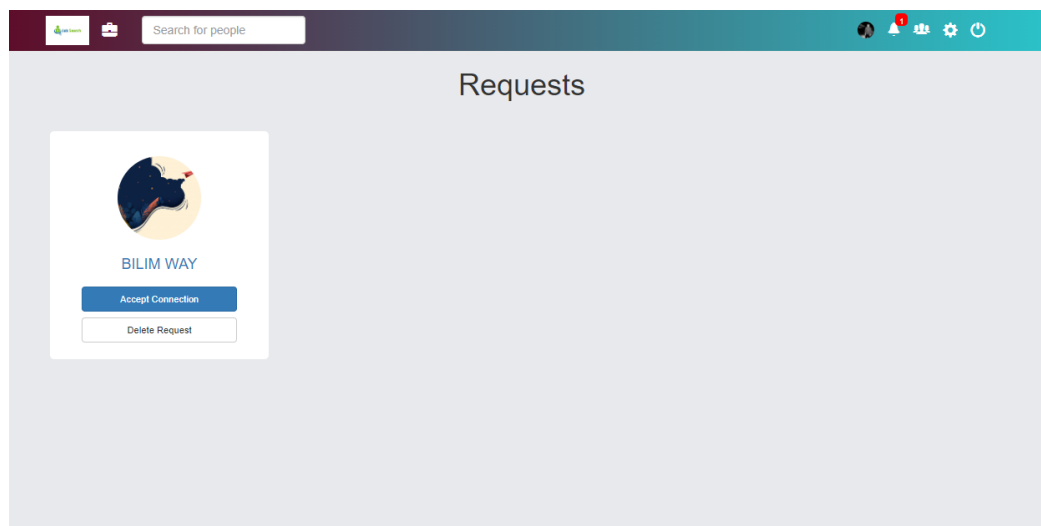


4.11-сурет – Жұмыскерді іздеу беті

Жұмыс іздеуші маманмен байланысу үшін «add connection» батырмасын басу жеткілікті.

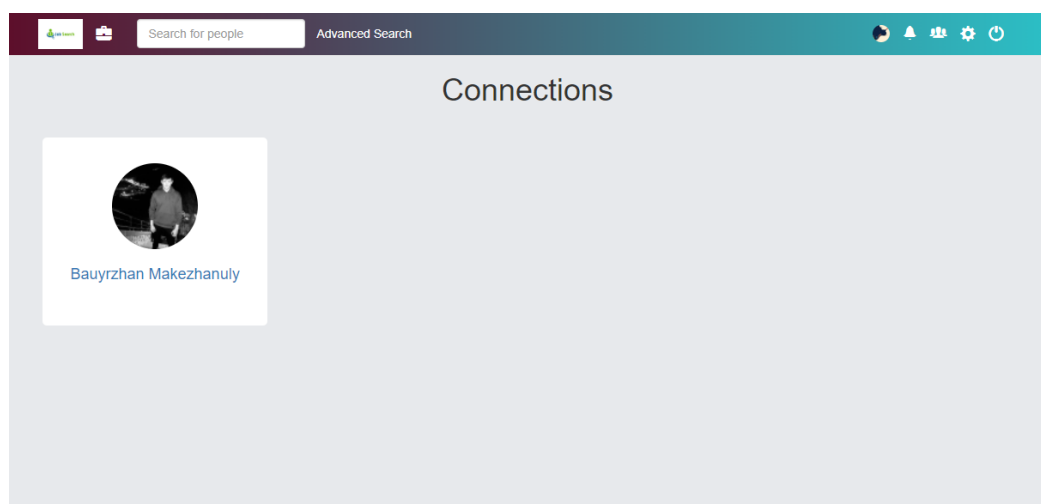


4.12-сурет – Жұмыс іздеушімен байланыс орнату



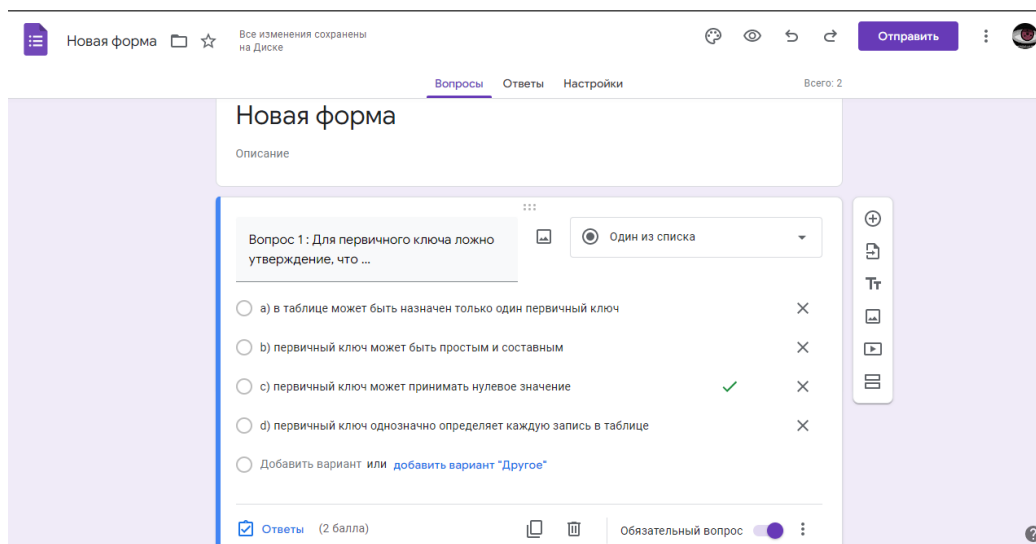
4.13-сурет – Жұмыс берушімен кері байланыс орнату

Жұмыс іздеуші байланысты растаған болса, байланыс бетінде мамандар тізімі пайда болады.



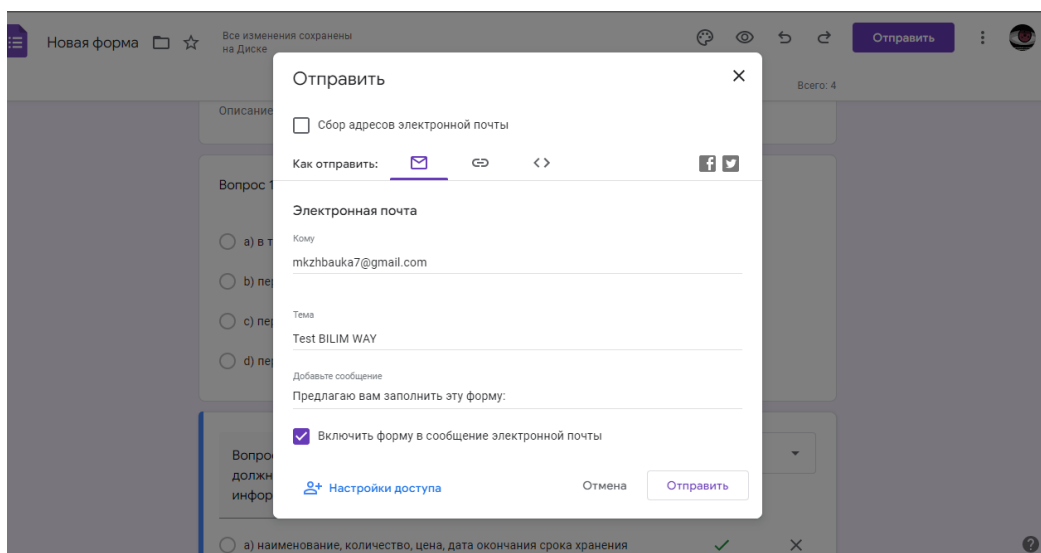
4.14-сурет – Байланыс орнатқан жұмыс іздеушілер тізімі

Жұмыс беруші өз тапсырмаларын жұмыс іздеушімен байланыс орнатқаннан кейін жібере алады. Тапсырмаларды кез келген ортада дайындап сілтемесімен бөлісу арқылы немесе электронды почтасына жіберу арқылы жұмыс беруші өз тапсырмасымен бөлісе алады. Сондай ортаның бірі Google Forms тапсырмаларды құрастыруға және онымен бөлісуге, нәтижелерін көруге өте ыңғайлы орта [16]. 4.15, 4.16, 4.17-суреттерде көрсетілген.

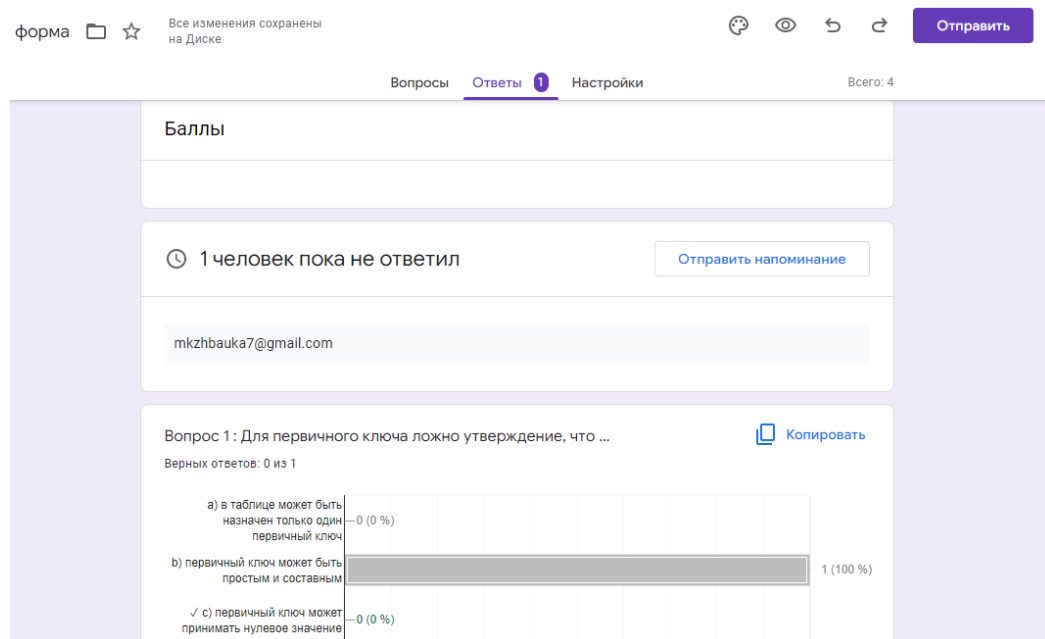


4.15-сурет – Тапсырмаларды құрастыру

Жұмыс беруші өз тапсырмасын электрондық почта арқылы немесе жүйеге жұмыс іздеуші түйіндемесіне сілтемені бекіту арқылы бөлісе алады.



4.16-сурет – Тапсырмалармен бөлісу



4.17-сурет – Тапсырма нәтижелерін көру

Жұмыс беруші тапсырма сілтемесін жұмыс іздеуші түйіндемесіне бекіте алады. Осылайша өз тапсырмаларымен бөлісе алады. 4.18-суретте көрсетілген. 4.19-суретте тест тапсырмасын түйіндемеген бекітілгенін жұмыс іздеуші көре алады.

Education
Satbayev University
 4 courses: IT
 2018-09 - 2022-05

Skills
 I know the following programming languages: C/C++, Python, Java. I know how to work with databases SQL.

Interests
 I like to play football.

Contact +

I liked your resume. That is why I am sending you the following assignment. Try to find a solution and if successful, we will contact you.

Follow the link below to take the test!
https://docs.google.com/forms/d/1sLpw_eJcL9GxwoVzyaNh0cjyXC7yfsX_qDao5y07Os/edit

Save Cancel

4.18-сурет – Тест тапсырмаларын түйіндемеге бекіту

Recognize and manipulate faces from Python or from the command line with the world's simplest face recognition library. Built using dlib's state-of-the-art face recognition built with deep learning. The model has an accuracy of 99.38% on the Labeled Faces in the Wild benchmark. This also provides a simple face_recognition command line tool that lets you do face recognition on a folder of images from the command line!

Education +

Satbayev University ✎

4 courses: IT
2018-09 - 2022-05

Skills ✎

Interests ✎

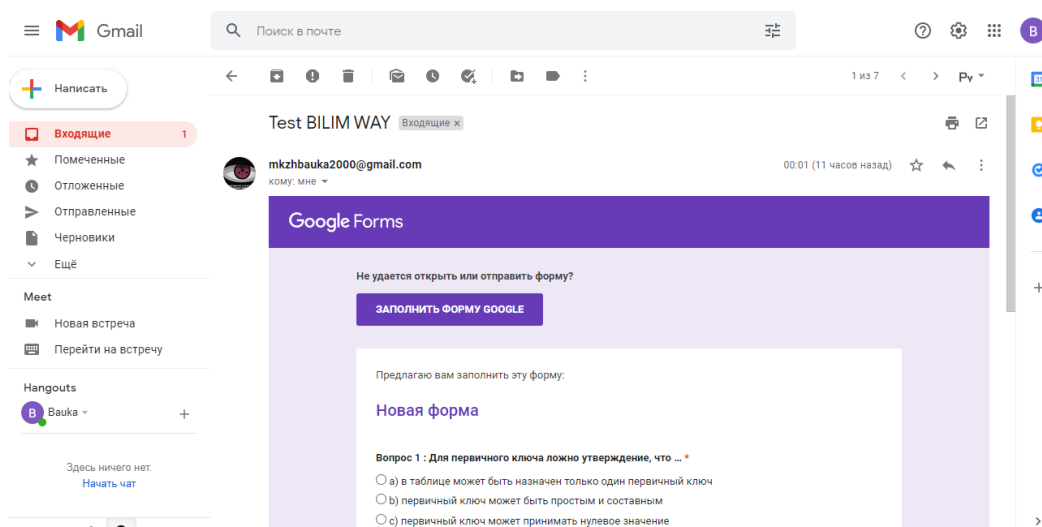
Contact



"I liked your resume. That is why I am sending you the following assignment. Try to find a solution and if successful, we will contact you. Follow the link below to take the test: https://docs.google.com/forms/d/1sLlpw_eJcl9GxwoVzyaNh0cjyXC7yfsX_qDao5y07Os/edit"

4.19-сурет – Тест тапсырмаларын түйіндемеден көру

Жұмыс беруші тарапынан келген тапсырманы орындау 4.20-суретте көрсетілген.



4.20-сурет – Тест тапсырмаларын орындау

ҚОРЫТЫНДЫ

Дипломдық жұмысты қорытындылай келе, жұмыспен қамтуды қадағалайтын веб-қосымша өз біліктілігін арттырғысы келетін, үнемі дамуды қалайтын маманды өз ісін табуына көмегін тигізеді деп үміттенемін.

Дипломдық жұмыста негізгі бағдарлама мәтіні python тілінің Django фреймворкін пайдаланады. ReactJS кітапханасын фронтенд бөлігіне пайдаланылды. Мәліметтер базасы ретінде PostgreSQL таңдалды.

Дипломдық жұмысты жобалауда UML диаграммасының 5 түрі пайдаланылды олар:

- use case diagram (пайдалану жағдайының диаграммасы);
- sequence diagram (тізбек диаграммасы);
- communication diagram (байланыс диаграммасы);
- state machine diagram (күй машинасының диаграммасы);
- collaboration diagram (ынтымақтастық диаграммасы).

Деректер қорын жобалауда ER диаграммасы пайдаланылды.

Тапсырмаларды құрастыру ортасына келесі 7 қосымшаны пайдалану ұсынылды, атап айтқанда:

- madtest
- kahoot
- socrative
- learningapps
- quizizz
- quizlet
- online test pad
- google forms

Елдегі жұмыссыздық деңгейі төмендеп, жаңа жұмыс орындарының көптеп ашылуына, кадр тапшылығын жоюға бар мүмкіндігін көрсетеді деп ойлаймын. Еліміздің өркендеп дамуына, білікті мамандарды көбейтуге, бәсекеге қабілетті жұмыс орындарының артуына көмектеседі деп сенім білдіремін.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

- 1) Қазақстанда жастар арасындағы жұмыссыздық деңгейі // Электрондық нұсқа <https://egemen.kz/article/283314-qazaqstanda-dgastar-arasyndaghy-dgumyssyzdyq-denhgeyi-tomendedi>
- 2) Қазақстандағы жұмыссыздық деңгейі 4.9% құрады // Электрондық нұсқа <https://kazgazeta.kz/news/114156>
- 3) Техникалық тапсырманы әзірлеу // Электрондық нұсқа <https://leally.ru/kk/browsers/v-sootvetstvii-s-tehnicheskim-zadaniem-tehnicheskoe-zadanie-opredelenie>
- 4) UML диаграмма түрлері // Электрондық нұсқа <https://creately.com/blog/diagrams/uml-diagram-types-examples>
- 5) UML диаграммаларын әзірлеу ортасы // Электрондық нұсқа <https://creately.com>
- 6) UML диаграммаларының сипаттамасы // Электрондық нұсқа https://flexberry.github.io/ru/fd_activity-diagram.html
- 7) Python бағдарламалау тілі // Электрондық нұсқа <https://metanit.com/python/tutorial/1.1.php>
- 8) Не себепті Django // Электрондық нұсқа <https://www.djangoproject.com/start/overview>
- 9) Django баптаулары // Электрондық нұсқа <https://metanit.com/python/django/1.2.php>
- 10) React JS-ті үйрену // Электрондық нұсқа <https://ru.reactjs.org/docs/getting-started.html#learn-react>
- 11) Жаңа React қосымшасын құру // Электрондық нұсқа <https://ru.reactjs.org/docs/create-a-new-react-app.html#create-react-app>
- 12) Дерекқорды жобалау ER-диаграммасы // Электрондық нұсқа <https://www.lucidchart.com/pages/ru/erd-%D0%B4%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0>
- 13) PostgreSQL дерекқоры // Электрондық нұсқа <https://www.postgresql.org>
- 14) Дерекқорды орнату // Электрондық нұсқа <https://www.postgresql.org/download/>
- 15) Дерекқорды құру // Электрондық нұсқа <https://metanit.com/sql/postgresql/2.1.php>
- 16) Тест критерийлерін құруда пайдаланылатын жеті қосымша // Электрондық нұсқа <https://www.eduneo.ru/7test/>
- 17) Google Forms // Электрондық нұсқа <https://www.google.com/intl/en-GB/forms/about/>

А қосымшасы

«JobSearch» сайтыን әзірлеуге арналған техникалық тапсырма

А.1 Кіріспе

Құжатта JobSearch сайттын әзірлеуге арналған тапсырма түсіндірмесін қамтиды.

Құжат жұмыстың жай-күйін сипаттайды, онда жұмысты жүзеге асыру кезінде ескеру қажет барлық ақпараттар қамтылады, жұмыс мақсаты мен жұмысқа қойылатын талаптар жазылады.

Құжатта келтірілген мәліметтер негізінде сайттың тұжырымдамасын әзірлеу жұмыстары жүргізілуде. Құжатты бекіткеннен кейін өзгерістер енгізу қосымша бекітуді талап етеді.

А.1.1 Негізгі анықтамалар

JobSearch сайты – жас мамандарды, студенттерді, өз мамандығы бойынша жұмыс жасауды қалайтын жандарға өз қалауы бойынша жұмысқа орналасуға көмектесу құралы болмақ.

Бұл сайт жұмыс беруші мен оны іздеген маман арасында байланыс орнатып, қажеттілігін орындауға жағдай жасайды.

Негізгі ерекшелігі жұмыс берушінің жас мамандарға қызығушылығы. Жұмыс іздеген маман өз парақшасын толтырған соң, жұмыс беруші тарапынан арнайы сұрақтарды шешу керек, егер тиесілі балды жинап сәтті өткен болса маман жұмыс беруші тарапынан шақырту алады. Бұл өз кезегінде жас маманның өзі қалайтын ісімен айналысуына таптырмас мүмкіндік сыйлайды.

Халқымыздың жарқын болашағы үшін көптеген компаниялар мен мамандарды біріктіреміз.

А.2 Дамыту аумағы

Дамыту аумағы – студенттердің жұмысқа орналасу порталы, мұнда барлығы тіркеліп, түйіндеме жазып, жұмыс іздей алады. Компаниялар жұмыс беруші ретінде тіркеледі және оған бос жұмыс орындары мен тапсырмалар жариялау мүмкіндігі беріледі. Кейін өз қатарына жаңа жас маманды таба алады.

А қосымшасының жалғасы

Портал міндетті түрде қамтуы керек:

- жұмысты немесе маманды тиімді іздеуге үлес қосу;
- бос жұмыс орындарын оңай көруге жағдай жасау;
- еліміздегі жұмыстың қызушылығына үлес қосу;
- компаниялардың жас мамандарды өзіне қабылдауына үлес қосу;
- жас маманның өз мамандығымен жұмыс жасауына үлес қосу;
- жұмыссыздық деңгейін төмендету.

А.3 Сайт құрылымы

- басты бет;
- жұмыс беруші үшін тіркелу беті;
- жұмыс іздеуші үшін тіркелу беті;
- жүйеге кіру беті;
- түйіндеме жариялау беті;
- бос жұмыс орынын жариялау беті;
- байланыс орнату беті;
- баптаулар беті;
- іздеу беті.

А.4 Графикалық дизайнға қойылатын талаптар

Порталды әзірлеу кезінде көбінесе көзге ұнайтын түстер, ашық, қарама-қарсы түстер пайдаланылуы керек. Дизайн еркін стильде жасалуы керек.

- веб-қосымша дизайнында мыналар болмауы керек:
- жиі жыпылықтайтын баннерлер;
- мәтінді біріктіру;
- қараңғы және агрессивті түс комбинациясы және графикалық шешімдер;
- көзге ауыртпашылық түсіретін көптеген элементтер жиынтығы;
- керісінше көптеген бос орынның болуы;
- өте ұсақ қаріптерді пайдалану;
- түсініксіз бейнелерді қолдану.

А.5 Функционалды талаптар

А қосымшасының жалғасы

А.5.1 Қол жетімділікті бөлу талаптары

Сайттың бос жұмыс орындар тізімін пайдаланушы аутентификациясыз оқуға қолжетімді болуы керек.

Пайдаланушылар топтары:

- administrator (деректерді басқару)
- жұмыс іздеуші (жүйеге тіркелу, түйіндеме жариялау және өңдеу, бос жұмыс орындарын қарау, тапсырмаларды орындау)
- жұмыс беруші (жүйеге тіркелу, бос жұмыс орындарын жариялау және өңдеу, түйіндеме қарау, маманды шақыру, тапсырмаларды жүктеу)
- the guest (материалдарды оқу, бос жұмыс орындарын қарау)

Жеке бөлімге кіру әрекеті кезінде аутентификациядан өтпеген пайдаланушыдан логин мен құпия сөз сұралуы керек. Аутентификациядан өткеннен кейін жүйе пайдаланушының сұралған бөлімге кіру рұқсаттарын тексеруі керек. Егер қол жеткізуге тыйым салынса, пайдаланушыға жеке бөлімге кіру мүмкін еместігі туралы хабарлама сұралуы керек.

Қол жетімділік деңгейі:

The guest (тіркеуден өтпеген қолданушы) сайт материалдарын келесі санаттар бойынша оқу:

- бастапқы бет ақпараттары;
- бос жұмыс орындары туралы ақпарат;
- компания профилі;
- тіркеуден өту мүмкіндігі;

Жұмыс іздеуші сайт материалдарын келесі санаттар бойынша оқу:

- бастапқы бет ақпараттары;
- бос жұмыс орынын қарау және өтінім тастау;
- компания профилі;
- түйіндеме жарялау және өзгерту;
- жеке деректерін басқару, қажеттілік болған кезде өзгерту;
- тапсырмаларды орындау мүмкіндігі.

Жұмыс беруші

- сайт материалдарын келесі санаттар бойынша оқу;
- бастапқы бет ақпараттары;
- түйіндеме қарау және байланысу;
- жеке деректерін басқару, қажеттілік болған кезде өзгерту;
- бос жұмыс орынын жариялау және өзгерту;
- компания деректерін өңдеу мүмкіндігі;
- тапсырмаларды жасау және өңдеу мүмкіндігі.

А қосымшасының жалғасы

А.5.3 Атқаратын іс-әрекет талаптары

Administrator

- барлық пайдаланушылардың жеке деректерін көру;
- барлық пайдаланушылардың жеке деректерін өңдеу;
- барлық компаниялардың деректерін қарау;
- барлық компаниялардың деректерін өңдеу.

Жұмыс іздеуші

- түйіндемені қарау;
- түйіндемедегі деректерді өңдеу;
- компания тарапынан шақыртуларды көру;
- тесттерден өту;
- бос жұмыс орынын қарау.

Жұмыс беруші

- компания туралы деректерді және жарияланған бос орындарды, тапсырмаларды қарау;
- компания туралы, жарияланған бос орындар, тапсырма деректерін өңдеу;
- жұмыс іздеуші маманды өз компаниясына шақыру.

А.6 Функционалды емес талаптар

А.6.1 Орналасу талаптары

Сайт беттерінің орналасуы дұрыс (визуалды және техникалық қателерсіз) көрсетілуі және келесі браузерлерде кросс-браузерлермен үйлесімді болуы керек:

- mozilla firefox версии 45+;
- google chrome версии 49+;
- apple safari версии 8+.

Windows және Mac OS операциялық жүйелерінде графикасы қосылған, пайдаланушы экранының ажыратымдылығы 1280 пиксель, кең және одан жоғары. Экранның енін орналасу арқылы анықтау керек.

Сайт беттерінің орналасуы iOS 9+, Android 5+, Windows Phone 8.1+ операциялық жүйелерін пайдаланатын мобильді құрылғыларда дұрыс (визуалды және техникалық қателерсіз) көрсетілуі керек.

А қосымшасының жалғасы

А.6.2 Ақпараттық қамтамасыз етуге қойылатын талаптар

Барлық сайт деректері ДҚБЖ бақылауымен құрылымдық пішінде сақталуы керек. Ерекшеліктер - көруге және жүктеп алуға арналған деректер файлдары (суреттер, бейнелер, құжаттар және т.б.). Мұндай файлдар файлдық жүйеде сақталады, ал оларға сілтемелер мәліметтер базасында орналастырылады.

Барлық бейнелер .gif, .png, .jpg пішімінде болуы керек.

Сайттың бір стандартты жүктеп алынатын бетінің көлемі орта есеппен 170 кб аспауы керек.

Б қосымшасы

Бағдарлама мәтіні

Б.1 Жүйеге қолданушыны тіркеу *components\user\new.js*

```
var React = require('react');
var firebase = require('firebase');
var Link = require('react-router').Link;
var hashHistory = require('react-router').hashHistory;
var SignUpForm = React.createClass({
  getInitialState: function(){
    return{hasError: false};
  },
  handleSignUp: function(){
    var that = this;
    var recruiter = this.state.recruiter === "true" ? true : false;
    var firstName = this.refs.firstName.value;
    var lastName = this.refs.lastName.value;
    var email = this.refs.email.value;
    var password = this.refs.password.value;
    var password_confirmation = this.refs.password_confirmation.value;
    if(firstName && lastName){
      firebase.auth().createUserWithEmailAndPassword(email,
password == password_confirmation ? password : "nil").catch(function(error) {
        if(error){
          that.setState({hasError: true});
          that.setState({errorMsg: "Please enter a valid email
address with a password of at least 6 characters."});
        }else{
          that.setState({hasError: true});
        }
      });
    }
  }
});
```

Б қосымшасының жалғасы

```
        that.setState({errorMsg: "First or last name field cannot be
empty."}))
    }
    this.unsubscribe = firebase.auth().onAuthStateChanged(function(user)
{
    if (user) {
        var userData = {
            email: email,
            first: firstName,
            last: lastName,
            recruiter: recruiter,
componentWillUnmount: function(){
    if (typeof this.unsubscribe == 'function')
    {
        this.unsubscribe();
    }
},
handleKeyPress: function(e){
    if(e.key == 'Enter'){
        try{
            this.handleSignUp();
        }
        catch(e){};
    }
},
accountChange: function(e){
    this.setState({recruiter: e.target.value});
},
errorMessage: function(){
```

Б қосымшасының жалғасы

```
        return <div className="alert alert-danger"><strong>Error!
</strong>{this.state.errorMsg}</div>;
    },
    noErrorMessage: function(){
        return <div></div>;
    },
    render: function(){
        var errorAlert;
        if(this.state.hasError){
            errorAlert = this.errorMessage();
        }else{
            errorAlert = this.noErrorMessage();
        }
        return (
            <div>
                {errorAlert}
                <div className="col-md-4">
                    <input type="text" ref="lastName"
placeholder="Last Name" className="form-control"
onKeyPress={this.handleKeyPress} /><br />
                    <input type="email" ref="email"
placeholder="Email Address" className="form-control"
onKeyPress={this.handleKeyPress} /><br />
                    <input type="password"
ref="password" placeholder="Password" className="form-control"
onKeyPress={this.handleKeyPress} /><br />
                    <input type="password"
ref="password_confirmation" placeholder="Password Confirmation"
className="form-control" onKeyPress={this.handleKeyPress} /><br />
                    <button onClick={this.handleSignUp}
className="btn btn-primary margin-bottom-10">Create Account</button><br />
                    Have an account? <Link
```

Б қосымшасының жалғасы

```
                to="/login">Login!</Link>
            </div>
        </center>
    </div>
    <div className="col-md-4">
    </div>
</div>
    );
}
});
module.exports = SignUpForm;
```

Б.2 Жүйеге кіру *components\session\login.js*

```
var React = require('react');
var firebase = require('firebase');
var Link = require('react-router').Link;
var hashHistory = require('react-router').hashHistory;
var LogInForm = React.createClass({
    getInitialState: function(){
        return{hasError: false};
    },
    handleLogIn: function(){
        var that = this;
        var email = this.refs.email.value;
        var password = this.refs.password.value;
        firebase.auth().signInWithEmailAndPassword(email,
password).catch(function(error) {
            var errorCode = error.code;
```

Б қосымшасының жалғасы

```
var errorMessage = error.message;
if(error){
    that.setState({hasError: true});
    that.setState({errorMsg: "Invalid email or password
combination."});
}
});
this.unsubscribe = firebase.auth().onAuthStateChanged((user) => {
    if (user) {
        hashHistory.push("/");
    } else {
        hashHistory.push("/login");
    }
});
},
componentWillUnmount: function(){
    if (typeof this.unsubscribe == 'function')
    {
        this.unsubscribe();
    }
},
handleKeyPress: function(e){
    if(e.key == 'Enter'){
        this.handleLogIn();
    }
},
errorMessage: function(){
    return <div className="alert alert-danger"><strong>Error!
</strong>{this.state.errorMsg}</div>;
```

Б қосымшасының жалғасы

```
    },
    noErrorMessage: function(){
        return <div></div>;
    },
    render: function(){
        var errorAlert;
        if(this.state.hasError){
            errorAlert = this.errorMessage();
        }else{
            errorAlert = this.noErrorMessage();
        }
        return (
            <div>
                {errorAlert}
                <div className="col-md-4">
                    </div>
                    <div className="col-md-4">
                        <center>
                            <h1 className="margin-top-30">Log
In</h1><br />
                            <input type="email" ref="email"
placeholder="Email Address" className="form-control"
onKeyPress={this.handleKeyPress}/><br />
                            <input type="password" ref="password"
placeholder="Password" className="form-control"
onKeyPress={this.handleKeyPress}/><br />
                            <button className="btn btn-primary
margin-bottom-10" onClick={this.handleLogIn}>Login</button><br />
                            No account? <Link to="/signup">Sign
up!</Link>
                        </center>
                    </div>
                </div>
            )
        );
    }
}
```

Б қосымшасының жалғасы

```
</div>  
<div className="col-md-4">  
</div>  
</div>
```

```
module.exports = LoginForm;
```

Б.3 Түйіндеме және бос жұмыс орынын жариялау
components\profileinfo\profile.js

```
var React = require('react');  
var firebase = require('firebase');  
var Link = require('react-router').Link;  
var hashHistory = require('react-router').hashHistory;  
var Summary = require('./summary.js');  
var Education = require('./education.js');  
var Projects = require('./projects.js');  
var Interests = require('./interests.js');  
var Experience = require('./experience.js');  
var Skills = require('./skills.js');  
var ProfileImage = require('./profileImage.js');  
var Connection = require('./connection.js');  
var JobListings = require('./joblisting.js');  
var Endorsements = require('./endorsements.js');  
var Profile = React.createClass({  
  getInitialState: function() {  
    return { user_name: "", recruiter: false, isCurrentUser: false, pageID:  
    "", currentUserID: "" };  
  }  
});
```


Б қосымшасының жалғасы

```
componentWillMount: function(){
    var that = this;
    this.setState({pageID: this.props.params.id});
    this.unsubscribe = firebase.auth().onAuthStateChanged((user) => {
    this.setState({isCurrentUser: user.uid == this.props.params.id});
    this.setState({currentUserID: user.uid});
    });

    this.userRef =
firebase.database().ref().child('users/'+this.props.params.id);
    this.userRef.on("value", snap=>{
        var user = snap.val();
        this.setState({user_name: user.first + " " + user.last});
        this.setState({recruiter: user.recruiter});
    });
},
componentWillReceiveProps: function(nextProps){
    this.setState({pageID: nextProps.params.id});
    this.unsubscribe = firebase.auth().onAuthStateChanged((user) => {
    this.setState({isCurrentUser: user.uid == nextProps.params.id});
    this.setState({currentUserID: user.uid});
    });

    this.userRef = firebase.database().ref().child('users/'+
nextProps.params.id);
    this.userRef.on("value", snap=>{
        var user = snap.val();
        this.setState({user_name: user.first + " " + user.last});
        this.setState({recruiter: user.recruiter});
    });
componentWillUnmount: function(){
```

Б қосымшасының жалғасы

```
    this.userRef.off();
    this.unsubscribe();
  },
  render: function(){
    var show;
    if(this.state.recruiter){
      show = <div>
        <Summary pageID={this.state.pageID}
isCurrentUser={this.state.isCurrentUser}/>
        <JobListings pageID={this.state.pageID}
isCurrentUser={this.state.isCurrentUser}/>
        <Endorsements pageID={this.state.pageID}
isCurrentUser={this.state.isCurrentUser}
currentUserID={this.state.currentUserID}/>
        </div>;
    }else{
      show = <div>
        <Summary pageID={this.state.pageID}
isCurrentUser={this.state.isCurrentUser}/>
        <Experience pageID={this.state.pageID}
isCurrentUser={this.state.isCurrentUser}/>
        <Projects pageID={this.state.pageID}
isCurrentUser={this.state.isCurrentUser}/>
        <Education pageID={this.state.pageID}
isCurrentUser={this.state.isCurrentUser}/>
        <Skills pageID={this.state.pageID}
isCurrentUser={this.state.isCurrentUser}/>
        <Interests pageID={this.state.pageID}
isCurrentUser={this.state.isCurrentUser}/>
```

Б қосымшасының жалғасы

```

                                <Endorsements pageID={ this.state.pageID }
isCurrentUser={ this.state.isCurrentUser }
currentUserID={ this.state.currentUserID } />
                                </div>
        }
        return (
                <div className="profile">
                        <center>
                                <div className="container profile-container">
                                        <center>
                                                <h1>{ this.state.user_name }</h1>
                                                <ProfileImage pageID={ this.state.pageID }
isCurrentUser={ this.state.isCurrentUser }/>
                                                <Connection pageID={ this.state.pageID }
isCurrentUser={ this.state.isCurrentUser }
currentUserID={ this.state.currentUserID }/>
                                        </center>
                                </div>
                        </center>
                        <br />
                        <hr />
                        { show }
                </div>
        );
    },
});
module.exports = Profile;
```

Қ.И.Сәтбаев атындағы Қазақ Ұлттық Техникалық Зерттеу Университеті
Автоматика және ақпараттық технологиялар институты
5B070300 – «Ақпараттық жүйелер»
ПҚІР

Студент Макежанұлы Б.

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

Ғылыми жетекші Алдаберген С.А.

Дипломдық жұмыс тақырыбы: Жұмыс берушінің тест критерийлері бойынша студенттерді жұмыспен қамту жүйесін құру

Қабылданған жұмыс көлемі:

Түсіндірме жазба көлемі 35 бет

Қосымша көлемі 17 бет

Дипломдық жұмыс тақырыпқа және техникалық тапсырмаға сәйкес келеді. Дипломдық жұмыс мақсаты жұмыс берушінің тест критерийлері бойынша студенттерді жұмыспен қамту жүйесін құруға негізделген. Жүйе жұмыс беруші мен жұмыс іздеген маман арасындағы байланысты сипаттайды.

Жұмыс мақсаты – мамандарды жұмыспен қамтуға көмектесетін программалық қосымша болып табылады. Жүйеге тіркелген қолданушылар келесіндей іс-әрекеттерді жасай алады. Жұмыс беруші өзінің тапсырмаларын құрастырып, оларға керек деп тапқан сұрақтарын ойластырып, критерийлерін анықтап және өз тапсырмасын бөлісе алуын қадағалайды. Өз кезегінде жұмыс іздеуші жүйеге сәтті тіркелгеннен кейін түйіндемесін жариялау мүмкіндігіне ие болады. Бос қызмет орындарын көре алады, егер талаптарға сәкес болса өз өтінішін қалдыра алады. Кейін жұмыс беруші тарапынан жолданған тапсырмаларды орындап, қайта жүктейді. Сәтті болған жағдайда өз ісін бастауға мүмкіндік алады.

Жұмыстың он жақтары келесі нәтижелерді қамтиды:

Кешендік тақырыптын клиенттік бөлімі құрылды, қызметкерлердің функционалдық диаграммалар арқылы зерттеу жұмыстары жүргізілді.

Жұмыс кемшіліктері:

Әдибиеттерге, көздеректерге сілтемелер көрсетілмеген Жұмыс толықтай аяқталған және 96% А деп бағалауға болады. Сәтті қорғаудан кейін студент Макежанұлы Б. “Ақпараттық жүйелер” 5B070300 мамандығы бойынша техника және технология бакалаврі деңгейін алуға лайықты.

Пікірлеуші:

Тұран университетінің

“Ақпараттық технологиялар” институтының
қауымдастырылған профессоры



AK Мамырова А.К.

Подпись *AK Mamirova*
заверяю
Начальник ОПР
университета «Туран»
AK
19.05.2022г.

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев университеті

Ғылыми жетекшінің пікірі

Дипломдық жұмыс

Макежанұлы Бауыржан

5B070300 – Ақпараттық жүйелер

Тақырыбы: Жұмыс берушінің тест критерийлері бойынша студенттерді жұмыспен қамту жүйесін құру

Бұл дипломдық жұмыс өзінің логикалық құрылымымен ерекшеленген. Түсіндірме жобаның құрамы кіріспеден, 4 бөлімнен, қорытындыдан, әдебиеттер тізімінен және қосымшадан тұрады.

Менің пікірімше, дипломдық жұмысты орындаушы алдына қойылған тапсырманы толығымен орындады және кейінгі технологияларын меңгергендігін көрсетті.

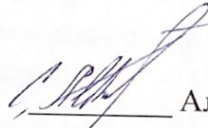
Жалпы дипломдық жоба профессионалдық деңгейде орындалған. Түсіндірме жазба сауатты бейнеленген, жоба бойынша барлық қажетті ақпараттар бар. Кемшілік ретінде кейбір шағын стилистикалық қателерді атап кетуге болады.

Жоғарыда айтылғандарға сүйене отырып, дипломдық жұмыс 5B070300 – «Ақпараттық жүйелер» мамандығының бітіру жұмыстарына қойылатын талаптарына сәйкес және дипломдық жұмыс қорғауға жіберіле алады, ал оның авторы Макежанұлы Бауыржан бакалавр академиялық дәрежесін алуға лайықты деп есептеймін.

Ғылыми жетекші:

Техникалық ғылымдар магистрі, лектор

«20» мамыр 2022 ж.



Алдаберген С.А.

Протокол анализа Отчета подобия

заведующего кафедрой / начальника структурного подразделения

Заведующий кафедрой / начальник структурного подразделения заявляет, что ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Макежанұлы Бауыржан

Название: Жұмыс берушінің тест критерийлері бойынша студенттерді жұмыспен қамту жүйесін құру

Координатор: Алдаберген С.А.

Коэффициент подобия 1: 1.41

Коэффициент подобия 2: 1.01

Замена букв:2 Интервалы:0

Микропробелы:9

Белые знаки:0

После анализа отчета подобия заведующий кафедрой / начальник структурного подразделения констатирует следующее:

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, работа признается самостоятельной и допускается к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы, по существу, и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, работа не допускается к защите.

Обоснование:

После анализа отчета по плагиату и работы дипломника выявлено, что заимствования являются добросовестными и не обладают признаками плагиата, так в основном связаны с применением общеизвестных терминов.

.....
Дата. « 20 » мая 2022 г.

.....
Подпись заведующего кафедрой /
начальника структурного подразделения

**Протокол анализа Отчета подобия
Научным руководителем**

Заявляю, что я ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Макежанұлы Бауыржан

Название: Жұмыс берушінің тест критерийлері бойынша студенттерді жұмыспен қамту жүйесін құру

Координатор: Алдаберген С.А.

Коэффициент подобия 1: 1.41

Коэффициент подобия 2: 1.01

Замена букв:2 Интервалы:0

Микропробелы:9

Белые знаки:0

После анализа Отчета подобия констатирую следующее:

обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, работа признается самостоятельной и допускается к защите;

обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы, по существу, и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;

обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, работа не допускается к защите.

Обоснование:

После анализа отчета по плагиату и работы дипломника выявлено, что заимствования являются добросовестными и не обладают признаками плагиата, так в основном связаны с применением общеизвестных терминов.

.....
Дата. «20» мәс 2022 г.

.....
Подпись научного руководителя