

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ БІЛІМ МИНИСТРЛІГІ

Қ.И.Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

Әлімбек Бақашқан

«Университет жатақханасы» веб-қосымшасын әзірлеу

ТҮСІНДІРМЕ ЖАЗБА

дипломдық жобаға

5B070400 – «Есептеу технологиясы және программамен қамтамасыз ету»

Алматы 2022

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ БІЛІМ МИНИСТРЛІГІ

Қ.И.Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

«ҚОРҒАУҒА ЖІБЕРІЛДІ»

ПИ кафедра меңгерушісі

физ-мат.ғыл.канд., профессор

_____ А.Н. Молдагулова

« _____ » _____ 2022 ж.

ТҮСІНДІРМЕ ЖАЗБА

дипломдық жобаға

Тақырыбы: «Университет жатақханасы» веб-қосымшасын әзірлеу

5B070400 – «Есептеу технологиясы және программамен қамтамасыз ету»
мамандығы

Орындады

Әлімбек Б.

Рецензент:

Физ-мат.ғыл.канд., доцент

_____ Мансурова М.Е.

« _____ » _____ 2022 ж.

Ғылыми жетекшісі:

Техн.ғыл.канд., қауымдастырылған
профессор

_____ Бейсембекова Р.Н.

« _____ » _____ 2022 ж.

Алматы 2022

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ БІЛІМ МИНИСТРЛІГІ

Қ.И.Сәтбаев атындағы Қазақ Ұлттық Техникалық Зерттеу Университеті

Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

БЕКІТЕМІН

ПИ кафедрасының меңгерушісі
физ.-мат.ғыл.канд., профессор
_____ А.Н. Молдагулова
" ____ " _____ 2022 ж.

**Дипломдық жобаны орындауға
ТАПСЫРМА**

Білім алушыға Әлімбек Бақашқан

Тақырыбы: «Университет жатақханасы» веб-қосымшасын әзірлеу

Академиялық мәселелер жөніндегі проректор бұйрығының
_____ " ____ " _____ 2022 ж. шешімімен бекітілген.

Орындалған жобаның өткізу мерзімі " ____ " _____ 2022 ж.

Дипломдық жобаның бастапқы мәліметтері:

Есеп – түсініктеме жазбаның талқылауға берілген сұрақтардың тізімі:

а) Түсініктеме жазбаның талқылауға берілген сұрақтар тізімі;

б) Тақырып бойынша таңдау және есептің қойылымын жасау;

в) Жобаны жобалау және пәндік сала бойынша талдау;

г) Бағдарламаны құру, кітапханаларды қосу, деректерді қосу және тестілеу;

Графикалық материалдар тізімі (міндетті суреттердің нақты көрсетілуімен):
презентацияның _22_ слайдпен берілген құжат түрінде ұсынылған.

Ұсынылған негізгі әдебиеттер: _21_ пайдаланылған әдебиеттер тізімінен

Дипломдық жобаны орындау
КЕСТЕСІ

Бөлімдердің атаулары, зерттелген мәселелердің тізімі	Ғылыми жетекшіге және кеңесшілерге ұсыну мерзімі	Ескерту
1. Дипломдық жоба тақырыбына байланысты әдебиеттерді іздеу және талдау	12.01.2022	орындалды
2. Зерттеліп отырған тақырыптың жалпы сипаттамасы	26.01.2022	орындалды
3. Әзірленген бағдарламалау ортасын таңдау	09.02.2022	орындалды
4. Дипломдық жобаның веб қосымшасын дайындау және іске асыру	10.03.2022	орындалды
5. Қосымша материалдар мен презентацияны дайындау	22.04.2022	орындалды

Дипломдық жұмыс бөлімдерінің кеңесшілерінің аяқталған жұмысқа қойған
қолтаңбалары

Бөлімдер атауы	Кеңес берушілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)	Қолтаңба қойылған мерзімі	Қолы
Нормалық бақылаушы	Жекамбаева М.Н. PhD, қауымдастырылған профессор		
Бағдарламалық бөлім	Марғұлан Қ. Тех.ғыл.магистрі, лектор		

Ғылыми жетекші _____ Р.Н. Бейсембекова

Тапсырманы орындауға қабылдап алған студент _____ Б. Әлімбек

Күні _____ «__» _____ 2021 ж.

АҢДАТПА

«Университет жатақханасы» веб-қосымшасын құрудың негізгі мақсаты студенттердің қай университетте болсын жатақхана тапшылығы мен университетте қағаз бастылықты әкеліп соғатын өзекті мәселені шешу үшін арналады.

Дипломдық жұмыс кіріспеден, негізгі үш бөлімнен және қорытындыдан құралған:

Кіріспеде жалпы тақырыптың өзектілігі ашылып, жобаның басты мақсаты мен тапсырмасы айқындалды.

Бірінші бөлімде университеттердің жатақханасына тапсырыс берудегі қадамдардан тартып негізгі идеяларға дейінгі барлық қадамдар жазылды.

Екінші бөлімде «Университет жатақханасына» бағытталған веб қосымшасына UML модельдеу тілі арқылы диаграммалар келтіріліп, жобада қолданылған бағдарламалар мен бағдарламау тілдері туралы кеңінен жазылған. Бұл веб қосымшада vite фронтэнд құрастыру құралы және React кітапханасын қолданылды, бэкэндке джанго, деректер базасына sqllite қолданылды. Дипломдық жоба нарықтағы ең тиімді веб технологиялармен жасалынды.

Үшінші бөлімде жоба құрылымы сипатталған және веб қосымшалардың барлық интерфейстерінің скриншоттары бейнеленген.

«Университет жатақханасының» веб қосымшасының түсініктемесі жалпы 48 бет, оның ішінде 24 сурет, 2 қосымшадан тұрады. Жұмысты жазуға 21 әдебиеттер мен сілтемелер қолданылды.

АНОТАЦИЯ

Основной целью создания веб-приложения «Общежитие университета» является решение проблемы нехватки общежитий в каком-либо университете и оформления документов в университете.

Дипломный проект состоит из введения, трех основных разделов и заключения: Введение раскрывает актуальность общей темы и определяет основные цели и задачи проекта.

В первом разделе описаны все этапы начиная от бронирования университетского общежития до основных идей.

Во втором разделе приведены схемы веб-приложения, направленного на «Общежитие университета» с использованием языка моделирования UML, и подробно описаны программы и языки программирования, используемые в проекте. В этом веб-приложении использовался конструктор frontend vite и библиотека React, в backend использовала Django, а в базе данных использовала Sqlite. Дипломный проект основан на самых эффективных веб-технологиях на рынке.

В третьем разделе описывается структура проекта и показаны скриншоты всех интерфейсов веб-приложения.

Концепция веб-приложения «Общежитие университета» состоит из 48 страниц, в том числе 24 изображений, 2 приложение. При написании работы было использовано 21 ссылки и статьи.

ANNOTATION

The main purpose of creating the University Dormitory web application is to solve the problem of a lack of dormitories in any university or the impossibility for students and complete documents at the university.

The graduation project consists of an introduction, three main sections and a conclusion: The introduction reveals the relevance of the general topic and defines the main goals and objectives of the project.

The first section describes all the steps from booking a university dormitory to the main ideas.

The second section provides diagrams of a web application aimed at the "University Dormitory" using the UML modeling language and describes in detail the programs and programming languages used in the project. This web app used the frontend vite builder and the React library, used Django for the backend, and used Sqlite for the database. The graduation project is based on the most effective web technologies on the market.

The third section describes the structure of the project and shows screenshots of all web application interfaces.

The University Dormitory web application concept consists of 48 pages, including 24 images, 2 applications. When writing the work, 21 links and articles were used.

МАЗМҰНЫ

Кіріспе	9
1 Негізгі бөлім	11
1.1 Жобаның өзектілігі	11
1.2 Веб қосымша жасаудың негіздері	12
1.3 Жатақханаға іздеу - қабылдау процессіне шолу	13
1.4 Жүйені модельдеу	13
2 Технологиялар бөлімі	17
2.1 React кітапханасы	17
2.2 Django	18
2.3 SQLLITE деректер базасы	21
3 Жоба құрылымы	23
3.1 «Университет жатақханасы » жобасы туралы	23
3.2 Веб қосымша интерфейстері	23
Қорытынды	32
Қолданылған әдебиеттер тізімі	33
А Қосымшасы	35
Б Қосымшасы	38

КІРІСПЕ

Заманымыздың қарқындап дамуына байланысты көптеген кірістер және құжаттандырулар мен өндірістер заман талабына сай компьютерлендіру үдеріснің ең қызғынды масштабында. Мемлекет басшыларының тапсырмалары бойынша қағазбастылықты жоғалту үшін бізге ИТ саласында жетістіктерге жету керек. Жақында болып жатқан дүние жүзі назарындағы оқиғада (Украина Ресей қақтығысында) бәрімізге аян тек қана шынайы қарулы соғыс ғана емес оның ішінде ИТ соғысы да болды десек артық емес. Хакерлер арасындағы жойқын соғыс бір-бірінің мемлекеттік сайттарын бұғаттап, атакалар жасау арқылы қарсы жаққа дер кезінде дүние жүзіне нақтылы хабарламалар жеткізе алмау сияқты фактлерге әкеліп соқты. Информатикалық технология тек атакамен ғана тоқталып қалмады ол гугл сияқты мықты халқаралық платформаны Ресейге тиімсіздік ету сияқты қатаң жазадан да тұрды. Гугл - Америка қоғамдық корпорациясына тиесілі кеңейтілген іздеу жүйесі. Танымалдығы бойынша бірінші орында тұр және 79.65 %ды құрайды [1]. Демек оның рөлі арқылы бір мемлекетті дүниеден хабарсыз етеді деп білсекте болады. Сол үшін елімізде ИТ дамыту ғана емес Америка құрама штаты сияқты түпкі интернет, бұлтты есептеулер, үлкен деректер, заттардың интернеті сияқты салаларда да табысқа жеткеніміз дұрыс. ИТ саласында жаңадан шыққан программалау тілдері немесе технологиялық стектерді үйренуде активті болу маңызды. Бірақ менің жасаған веб қосымшамда бұрын хит болған, қазірде басым орында тұрған JS кітапханасы REACT-ті қолдандым. Ол веб сайтты онай түрде жалдам қалыптастыру үшін арналған. Vite-қа келетін болсақ, Vite- бұл алдыңғы қатарлы фронтенд develop-ының жаңа құралы. Сіз оны қораптан тыс әзірлеу сервері + орау құралдарының тіркесімі ретінде қарастыра аласыз, бірақ жеңілдірек және жылдамырақ.

«Университет жатақханасы» веб қосымшаның негізгі аудиториясы – студенттер, яғни «Универ жатақханасы» веб қосымшасы арқылы әр қаладағы , әр универдегі студенттер өздеріне керекті жатақхана тандай алады және өтінішті онлайн форматта беру арқылы қолжетімділікке жете алады.

Дипломдық жұмыстың мақсаты: студент өміріндегі маңызды мәселенің бірі жатақханамен қамтамасыз ету, жатақхана админдерінің жұмысын оңтайландыру.

Дипломдық жоба өзектілігі: «Универ жатақханасы» веб қосымшасы арқылы студент пен жатақхана админдерінің арасындағы байланысты электронды форматқа көшіру, жасырын ерекшелігі оның бос қалған жатақхананы студент веб қосымша арқылы өтініш беріп сол арқылы тұрғын мәселесін шеше алады.

Артықшылықтары, біріншісіден тенант немесе студент “Жатақхананы іздеу” арқылы әр қаладағы әр университеттің әр жатақханасына өтініш қалдыра алады, жасырын ерекшелігі егер бір қаладағы сол университетте бос орын бар болса Организация менеджері ол студентке бөлмеден орын бөліп береді. Ал “Өтініш беру” арқылы ондағы талап етілген құжаттарды скан түрде жүктеп жатақхана админның жауабын күте алады. Екіншіден организация менеджері

жатақхана құрады және бөлме, төсек орын сияқты барлық керекті бөлімдерді реттейді және кімнің қандай бөлмеде жататынын басқарады. Үшіншіден site-admin организация админдерінің жіберген өтініштері арқылы жаңадан университет және жатақхана көздерін қосып береді.

1 Негізгі бөлім

1.1 Жобаның өзектілігі

Мемлекетімізде мобильді қосымшаларға ұқсайтын веб қосымшалардың үлесі артқан сайын оның күнделікті қолданылуы да артып келеді. Тікелей ғаламтор желісіне қосылуы талап етілетін бұл веб қосымшаның түрі барлық құрылғылардың бураузерлерінде қолжетімді.

«Университет жатақханасы» веб қосымшасы негізінен үйі басқа қалада бірақ өзі басқа қалада оқып жатқан студенттерге немесе университеттен алыс тұратын студенттерге арналған. Мұнда айтып өткендей веб қосымшаны кез-келген құрылғылардың браузерлерімен кіру арқылы өзінің талабына немесе қалағаны жайлы жатақхана іздей алады. Әрине бұл университет жатақханасы болғандықтан студенттердің өтінішін қабылдағанда шарттарға қараймыз. Оның ішінде ерекше жағдайы бар студенттер қосымша құжатты өтінішке қосып жүктейді содан кейін барып жатақхана админдеріне бұл студенттер бірінші кезекте көрсетілетін болады. Менің веб қосымшам осы функциясымен ерекшеленеді.

«Университет жатақханасы» веб қосымшасы негізінен студенттерге арналғандықтан біз оларға ең нақты және жылдам үдерістегі қызмет көрсетеміз. Яғни болашақта тек жатақхана көру, таңдау, өтініш беру ғана емес жатақханаға төленетін сумманы да осы веб қосымша арқылы төлеулеріне болады.

Бұл жоба төрт түрлі веб қолданушыға арналған, яғни атап өтетін болсақ, Жатақханаға сұраныс беретін, қаланы, университетті, жатақхананы, бөлмені, төсек орынға тапсырыс беретін студенттер яғни *tenant*. Екіншісі *Org-manager* яғни өздерінің жатақханасын осы веб қосымшасына жүктеуге өтінішті қабылдағаннан кейін жатақхананы және бөлмелердің, төсек орындардың нақты фотоларын жүктеуші, студенттерден келіп үскен өтініштерді тексеріп қабылдаушы, жатақхана кері байланысын реттеуші болып табылады. Үшіншісі *Site-admin* бұл рөлдегі админ тек жатақхана админінің өтінішін өткізуші, реттеуші немесе жоюшы болып табылады. Ең соңғысы осы веб қосымшаның негізін қалаушы. Веб қосымшадағы *environment construction*-ға келетін болсақ фронтенді *Node.js* нұсқасы 16.13.0 және *Node* Түйін пакетінің менеджері керек. Бэкэндке *Python* 3.9.6 және *Django* 3.2.8 керек.

Ал қолданушыға көрінетін бетті *React + vite* арқылы жасалынды, қосымша *HTMLtoImage* кітапханасы қолданылды. *Джанго* және *питонмен* бэкэнд жасадым.

1.2 Веб қосымша жасаудың негіздері

1-кестеде әзірленетін жобаның пәндік саласында қолданылатын барлық терминдер мен аббревиатуралар, сондай-ақ әзірлеуде қолданылатын технологиялар арқылы жобаны бағдарламалық қамтамасыз етуді енгізуге қатысты нақты терминдер көрсетілінген.

1.1-кесте – Қысқартулар, терминдер және олардың анықтамалары

Аббревиатура немесе термин	Анықтама
МБ	Мәліметтер базасы
МҚБЖ	Мәліметтер қорын басқару жүйесі
API	Application Programming Interface
UML	Unified Modeling Language
URL	Uniform Resource Locator
ER	Entity Relationship
JS	Java Script

Веб қосымша – бұл гиперсілтемелермен байланысты веб беттер жиынтығы. Веб беттер немесе гипермәтін құжаттары тег деп аталатын арнайы командалар (tags) бар мәтін болып табылады. Бұл тегтер бет элементтерін пішімдеуді қамтамасыз етеді және графикалық нысандарды, суреттерді, гиперсілтемелерді және т.б. орналастыруға мүмкіндік береді.

Веб қосымшаны құру үшін негізгі қажеттер:

- сайтты құру мақсатын анықтау;
- техникалық тапсырмаларды әзірлеу;
- сайттың доменін белгілі бір аймақта тіркеу (com, ru, kz, net);
- сайтты әзірлеу;
- хостингке сайт орналастыру;
- кез-келген веб қосымшаның негізгі мақсаты клиентті веб сервермен байланыстырып, қалаған сұраныстарды жүргізуге рұқсат беру болып табылады.

Веб қосымшаның бірқатар артықшылықтарын атап өтсек:

- ғаламторға кірген кез-келген қолданушыға қолжетімді;
- бағдарламалық қосымша орнатудың қажеті жоқ;
- уақытылы жаңартулар оңай жасалынады;
- жадыда орын алмайды;
- беттер динамикалық түрде жаңарады;
- іздеу жүйелері мен тақырыптық каталогтарда тіркеу.

1.3 Жатақханаға іздеу - қабылдау процессіне шолу

Студент «Универ жатақханасы» веб қосымшасына кіріп, жатақхананы іздеу батырмасын басып ондағы қаланы және университетті таңдайды, сосын келесі қадамда өзінде егер ерекше жағдайы болса, (мысалы: көп балалы отбасынан, мүгедектік құжаты болса, акесі немесе анасы ерте қайтыс болған сияқты) осы құжаттарды жүктеу арқылы бірінші мәмілеге немесе кезекте жатақхананы ала алады. Өтініш «Универ жатақханасы» жүйесіндегі жатақхана менеджеріне келіп түседі, жатақхана менеджері бұл тапсырысты көріп, анықтап мақұлдайды немесе кері қайтарады сосын жауабын жүктейді. Мұндағы жатақхана менеджері мен студент арасындағы айналым оңайланды және студентке тиімділі ерекше.

«Универ жатақханасы» веб қосымшасы бірқатар мәселелерді шешті:

- деректер мен құжаттар орталықтандырылған деректер қорында сақталған;
- әрбір университет студенттеріне ақпарат қолжетімді және ыңғайлы;
- студент пен жатақханаға онлайн түрде байланыс орнату;
- әрбір жатақхана және бөлмелердің фотосының шынайылығы;
- өтінішке қажетті құжаттарды сәйкес бағанға жүктеу;
- ерекше жағдайы бар студенттерге екінші рет ұтымды орай беру.

1.4 Жүйені модельдеу

Веб қосымшаның техникалық тапсырмаларын әзірлеу барысында жобаны модельдеу жұмыстары қатар жүреді. Ауқымды жүйе құруда нақты мақсатқа жету үшін бірыңғай модельдеу тілі қолданылады.

UML (бірыңғай модельдеу тілі) – бұл объектіге бағытталған бағдарламалауды талдау және жобалау үшін қолдануға болатын белгілеу жүйесі. Ол бағдарламалық жасақтаманы визуалдау, нақтылау, жобалау және құжаттау үшін қолданыла алады.

Модельдеу жүйені түсіну үшін қажет. Сонымен қатар, жүйені құру кезінде бір модель түрі жеткіліксіз болады. Сондықтан, кез-келген тривиалды емес жүйені түсіну үшін бір-бірімен өзара байланыстағы көптеген модельдерді жасау қажет. UML арқылы қосымшаның функция және архитектурасын диаграммалық, тексттік, сызба түрінде құруға болады. Бағдарламалық жасақтамаға қолданған кезде, бұл оның жүйелік архитектурасының әртүрлі көріністерін бейнелеу үшін қолданылатын тіл екенін білдіреді.

Бірыңғай модель диаграммаларында нысандар және олардың арасындағы қатынастар түрінде ұсынылады. UML моделдеу тілінде диаграммалардың көптеген түрлері бар, бірақ осы дипломдық жұмыста, түсінікті әрі оңай болғандықтан прецеденттер диаграммасын қолдандым. Прецеденттер

диаграммасы – UML диаграммаларының бірі, жүйенің динамикалық аспектілерін модельдеу үшін қолданылады.

Прецеденттер диаграммалары элементтің іс-әрекетін визуализациялау, нақтылау және құжаттау үшін маңызды. Олар жүйелерді, ішкі жүйелерді немесе класстарды түсінуді жеңілдетеді, бұл элементтерді контекстте қалай қолдануға болатындығы туралы сыртқы көріністі ұсынады. Прецеденттер диаграммасы экторлар (actor) мен олардың іс-әрекеттері арқылы сипатталады. Эктор (actor) – жүйедегі белгілі бір қолданушылар [1,2].

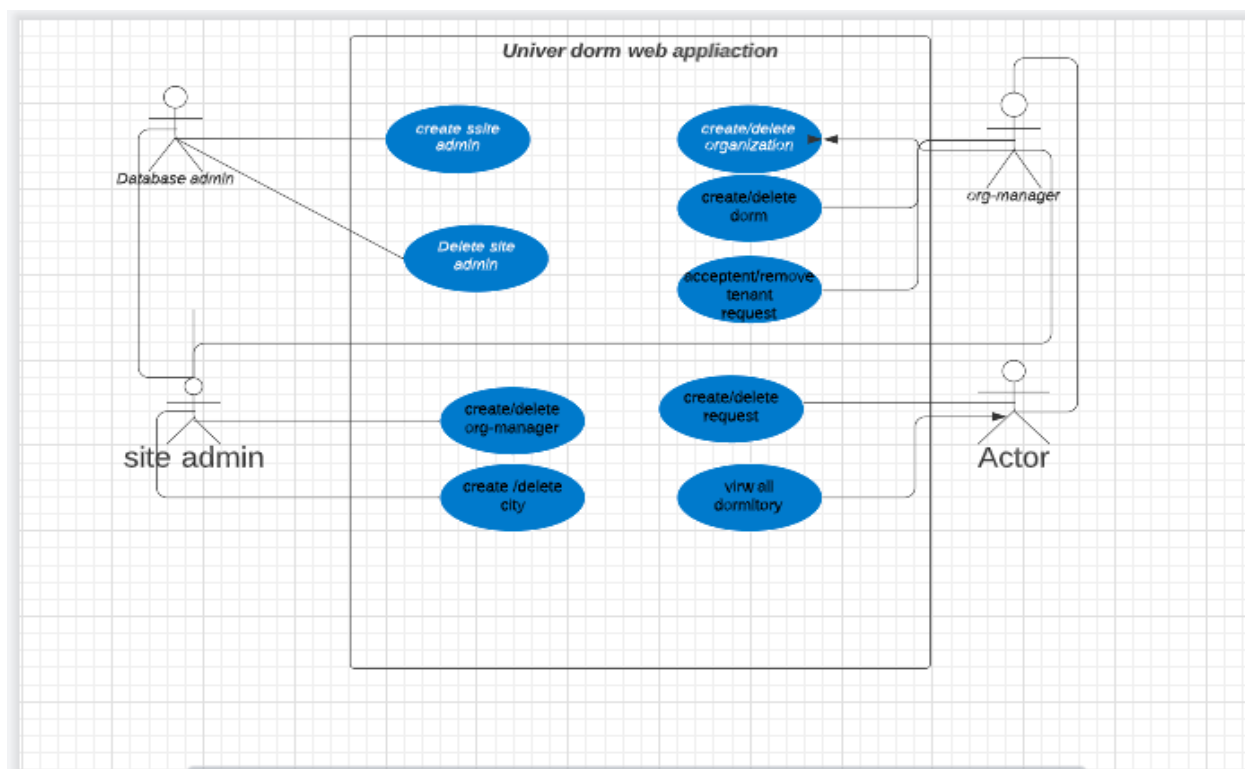
Модельдеу жүйесіне қойылатын талаптар:

- жүйедегі негізгі экторларды анықтап алу;
- әр эктор үшін жүйеден күтілетін немесе талап етілетін іс-әрекеттерін қарастыру;
- белгілі экторлармен байланысатын іс-әрекеттерді біріктіріп диаграммаға сызу.

Схемада пайдалану жағдайларының моделін көрсету үшін қолданылады:

- жүйелік шекара;
- үстіңгі жағында аты бар тікбұрыш және ішінде эллипс (пайдалану жағдайлары). Пайдалы ақпаратты жоғалтпастан жиі шығарып тастауға болады;
- актер (ағылшынша актер);
- кейбір құрылымдармен (жүйемен, ішкі жүйемен, сыныппен) өзара әрекеттесетін қолданушы рөлдерінің жиынтығын білдіретін (кең мағынада: тұлға, сыртқы тұлға, сынып, басқа жүйе) стильдендірілген адам. Актерлер бір-бірімен байланысты бола алмайды (жалпылау / мұрагерлік 14 қатынастарды қоспағанда);
- прецедент–актерлер байқайтын нәтижеге әкелетін, жүйе орындайтын әрекеттерді білдіретін (мүмкін нұсқаларды қамтуы мүмкін) жазуы бар эллипс. Субтитр жүйенің не істейтінін («қалай» емес) атауы немесе сипаттамасы болуы мүмкін (актер тұрғысынан);
- мінез-құлықты бейнелейтін іс-әрекеттердің нақты бірізділігі.

Сценарий кезінде актерлер жүйемен хабарлама алмасады. Сценарий UML түсініктемесі ретінде use-case диаграммасында көрсетілуі мүмкін. Бір пайдалану жағдайымен бірнеше түрлі сценарийлерді байланыстыруға болады.

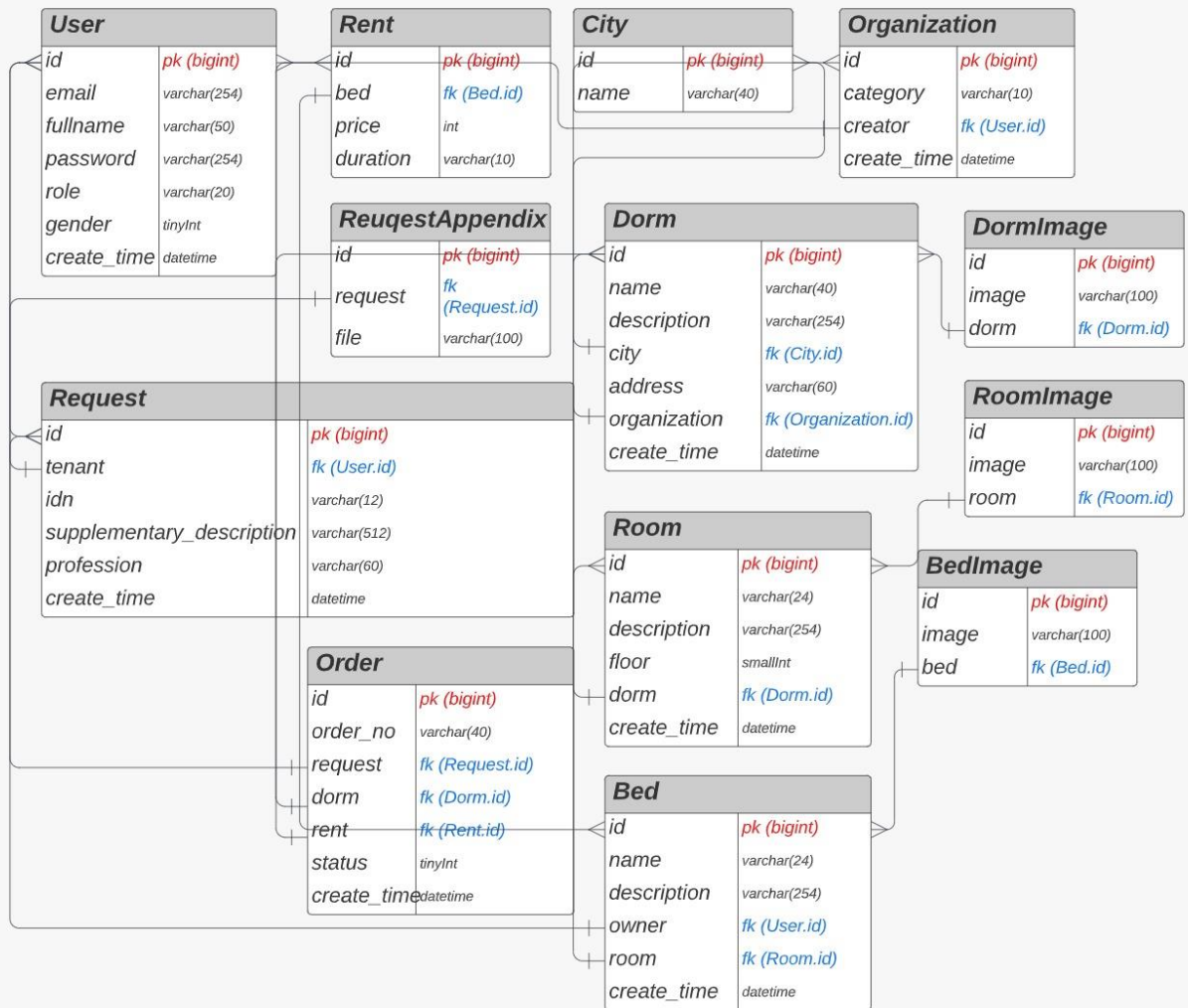


1.1-сурет – Use-case диаграммасы

Деректер қорындағы кестелердің байланыс сызбасы төменде 1.2-суретте көрсетілген:

Dormitory management system

Бақашқан



1.2-сүпет – ER diagram

2. Техникалық бөлім

2.1. React кітапханасы

React – бұл веб үшін де, мобильді қолданбалар үшін де жоғарғы интерфейсті құруға көмектесетін ашық бастапқы JavaScript кітапханасы. Ол басқа JavaScript фреймворктарымен және кітапханаларымен ыңғайлы үйлеседі және компоненттер деп аталатын шағын, дербес код биттерін қамтиды.[3]

Мен бұл бағдарламалық заманауи технологиялар негізінде құрастырдым. Қазіргі заманда көптеген програмистер жаңа технологиялар құруда

React мүмкіндіктері:

1. Декларативті дизайн – React қолданбаны оңай сипаттай алатын декларативті парадигманы қабылдайды.

2. Тиімді – React DOM эмуляциясы арқылы DOM-мен әрекеттесуді азайтады.

3. Икемді – React белгілі кітапханалармен немесе фреймворктермен жақсы жұмыс істейді.

4. JSX – JSX — JavaScript синтаксисінің кеңейтімі. React Development міндетті түрде JSX қолданбайды, бірақ біз оны ұсынамыз. (Менің программда jsx қолдандым).

5. Компоненттер – React арқылы құрамдастарды құру кодты қайта пайдалануды жеңілдетеді және үлкен жобаларды әзірлеуде жақсы пайдаланылуы мүмкін.

6. Бір жақты жауап беретін деректер ағыны – React бір жақты жауап беретін деректер ағынын жүзеге асырады, осылайша кодтың қайталануын азайтады, сондықтан ол дәстүрлі деректерді байланыстырудан оңайырақ.[4]

React ерекшеліктерін атап өтсек:

Facebook қолдауы;

– Әзірлеуші құралдары бар болуы(CLI);

– TypeScript (қатты типизацияланған код жаза аласың);

– JSX көмегімен реактивті бағдарламалау;

– HTML-ге негізделген шаблондар;

– Кроссбраузерлік Shadow DOM;

– HTTP арналған кроссбраузерлік қолдау;

– Әлем бойынша қолданушылардың көптігі;

React кез-келген «Болжамды» және «декларативті» веб-пайдаланушы интерфейстерін құруда ол веб-қосымшаларды тезірек әзірлеу үшін келесі тұжырымдамаларда сипатталған:

SPA (single page application) – Сайт басты бетке бірінші рет кірген кезде пайдаланушы тәжірибесі үшін қажетті барлық HTML/JS файлдарын жүктейді, ал кейінгі бет ауысуларында браузерді жүктейді [5]. Мен осы веб қосымшамен жұмыс жасағанда SPA қолдандым. Оның бір себебі ретінде Backend-frontend архитектураларын бөлек істеу. Егер осы бөлек архитектурамен жұмыс жасағанда

MVC (model view controller) ішіндегі V қолдана алмадым. Сол үшін оның орнына SPA қолдандым.

Жалпы React ішіндегі қоймалардың төмендегі біренешеуін басты қолдандым. Олар:

1. [@ant-design/ icons](#)- Барлық icons дизайндері;
2. Antd- UI қоймасындағы барлық бізге көрінетін батырма, card, component-тер;
3. Axios- frontend желіге сұранысы
4. Ahook- userequest сияқты hook-тармен сұранысты өңдеу кітапханасы

Бұл кітапхана қоймалары package.json ішінде тәуелділіктерде тұрады. Ол төменде 2.1-сурет тәуелді кітапханалар суретінде көрсетілген:

```
"dependencies": {  
  "@ant-design/icons": "^4.7.0",  
  "ahooks": "^3.1.9",  
  "antd": "^4.19.2",  
  "axios": "^0.25.0",  
}
```

2.1-сурет – Тәуелді кітапханалар

2.2 Django

Django - Қауіпсіз және техникалық қызмет көрсетуге болатын веб-сайттарды жылдам жасауға мүмкіндік беретін жоғары деңгейлі Python веб-жақтамасы [6]. Өз сөзіммен айтар болсам Backend архитектурасын құрудағы маңызды құрал.

ORM(object relationship mapping) frame-ын қазақша айтқанда жақтауын ұйымдастырылған мәліметтер базасының модельін және MVC(model view controller) арқылы API жасалыны.

ORM фреймын құру кезінде класс құру және сол класстың ішіндегі таблица арқылы жұмыс жасалынады. Мысалға: Backend-apps-models папкасының ішіндегі класс бағанына қарасақ ол жерде db_table='city' көре аламыз яғни бұл класстың ішіндегі таблица атауы. Келесіде әрбір отрубуттары бір өріс болып табылады және әрбір өрісте аргументтері болады. Төменде 2.2-суретте көрсетілгендей.

```

models.py X
backend > apps > dorm > models.py > City > Meta
1  from django.db import models
2  from django.conf import settings
3  from django.dispatch import receiver
4  from django.db.models.signals import post_delete
5
6  from os import remove
7
8
9  class City(models.Model):
10     '''城市'''
11     name = models.CharField(max_length=40, unique=True,
12                             verbose_name='Қала атуы')
13
14     class Meta:
15         db_table = 'city'
16         verbose_name = 'Қала'
17         verbose_name_plural = 'Қалалар'
18
19     def __str__(self):
20         return self.name
21
22
23  class Organization(models.Model):
24     '''机构'''
25     name = models.CharField(max_length=40, verbose_name='Ұйым атуы')
26     CATEGORY_CHOICES = (
27         ('university', 'Университет'),
28         ('college', 'Колледж'),
29         ('school', 'Мектеп'),
30         ('other', 'Басқа'),
31     )
32     category = models.CharField(
33         max_length=10, choices=CATEGORY_CHOICES, verbose_name='Санат')
34     creator = models.ForeignKey(
35         'user.User', on_delete=models.CASCADE, verbose_name='Құрушы')
36     create_time = models.DateTimeField(

```

2.2-сурет – ORM

MVC - (model view controller) Деректерді және кодты ұйымдастыру үшін интерфейс дисплейін бөлу және бизнес логикасын бір құрамдас бөлікке жинау үшін бизнес логикасын пайдаланатын бағдарламалық жасақтаманың дизайн парадигмасы. Интерфейс пен пайдаланушы әрекетін жақсарту және теңшеу кезінде бизнес логикасын қайта жазу қажет емес.

Негізгі идея: ажырату (деректерді бөлу, бизнесті өңдеу және бетті көрсету)

M: Модель (деректер операцияларын инкапсуляциялайды, ең көп қолданылатыны дерекқор);

V: Көрініс (деректерді көрсетуге және көрсетуге жауап береді, View - пайдаланушыға тікелей шығару);

C: Контроллер (контроллер бизнесті басқарады, сұрауларды қабылдайды, үлгі мен көріністі үйлестіреді) [7].

Мен Фронтэнд және Бэкэнд бөлек архитектурасында жазғандықтан бұл жерде V жоқ. Оны React SPA –да жаздым. Модель мен контроллерге келетін болсақ әрбір URL-де адрес болады. Мысалы: dorm/city кіргенде ол CityView контроллерін яғни классын қозғайды. Ал CityView классында get және post запростарымен жұмыс жасайды. Қысқаша түсіндіріп өтсем төмендегі кодта бірінші барлық қаланың тізімі алынады сосын ол қалалар тізімі атты хабарламаны қайтарады.

url.py:

```

urlpatterns = [
    url(r'^dorm/city/$', CityView.as_view()),
    url(r'^dorm/city/(?P<id>\d+)/$', CitySingleView.as_view()),

```

view.py:

```

class CityView(View):

```

```

def get(self, request):
    cities = City.objects.all()
    return JsonResponse({'message': 'Қалалар тізімі', 'cities':
serializer_data(cities)}, status=200)
def post(self, request):
    is_valid, user_or_response_content = verify_token(request)
    if not is_valid:
        return JsonResponse(user_or_response_content, status=401)

```

Төменде екі запрос жеңіске жеткенде қайтаратын статус коды берілген:
 Get-status code= 200
 Post-status code= 201

Авторизациялау бөлігіне мен токен әдісін қолдандым. Басында қолданушы браузерде логин пароль жазады сосын ол джангоға келеді де қолданушы папкасының ішіндегі url.py құжатына келеді. Сосын urls ішінде лайықты адресі бойынша лайықты контроллерді қозғайды. Логин жасау үшін мен пост запросын қолдандым. Бірінші қадамда get_data арқылы логин пароль аламыз. Екінші verify_login арқылы оны тексереміз егер қате болса қате жібереміз ал дұрыс болса онда келесі try-ға өтеміз, жазылған пошта адресс бар ма жоқ па соны тексереміз. Алдында айтып өткендей цикл орындайды да дұрыс болса келесі қадамға өтеміз. Үшінші пароль тексереді. Бұл кезде generate_token арқылы токен пайда болады. Келесі әрбір запроста осы токен арқылы тексерілетін болады. Нақты код түрі төменде 2.2.2-суретте көрсетілгендей:

```

class LoginView(View):
    def post(self, request):
        data = get_data(request)

        is_valid, response = verify_login(data)
        if not is_valid:
            return response

        try:
            user = User.objects.get(email=data.get('email'))
        except User.DoesNotExist:
            return JsonResponse({'message': 'авторизация сәтсіз болды'}, status=400)

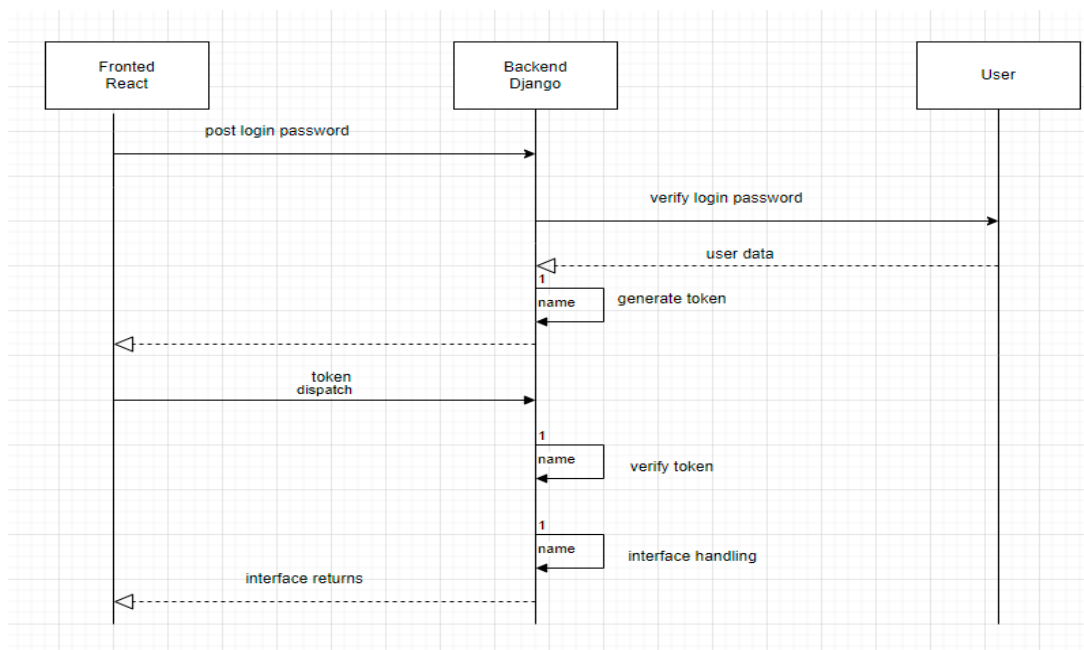
        if check_password(data['password'], user.password):
            return JsonResponse({'message': 'авторизация сәтті болды',
                                'token': generate_token(user.id),
                                'user': serializer_data(user, {'is_multiple': False, 'exclude_fields': ['password']}),
                                }, status=200)

        return JsonResponse({'message': 'авторизация сәтсіз болды'}, status=400)

```

2.3-сурет – Контроллер

Схема түрінде төменде 2.4-сурет токене көрсетілгендей:



2.4-сурет – Токен

Осымен Django бөлігінің жалпы логикасын тамамдалды.

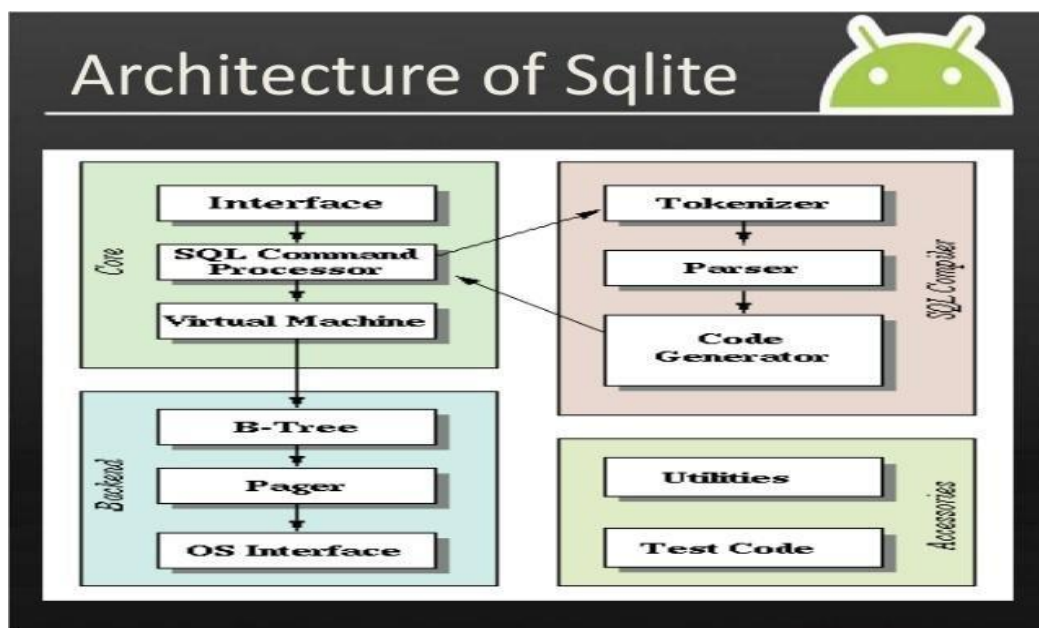
2.3 Sqlite деректер базасы

Мәліметтер қорымен жұмыс істеу үшін визуалды режимде деректерді басқара алатын дерекқор менеджерін пайдаланады. Барлық ДҚ бір тапсырманы орындайтынына қарамастан, олар пайдаланушыларға жасауға, өзгертуге және қосуға мүмкіндік береді.

Деректер қорларында сақталған ақпаратқа қол жеткізу үшін тапсырманың өзін әртүрлі тәсілдермен орындауға болады. Сондай-ақ, әрбір ДҚБЖ функциялары мен мүмкіндіктері әртүрлі болуы мүмкін. Әртүрлі ДҚБЖ әртүрлі тәсілдермен құжатталады: қысқаша немесе егжей-тегжейлі. Әртүрлі танымал дерекқорларды салыстыру кезінде біз мыналарды ескердік: пайдаланудың қарапайымдылығы, жобада бұрыннан қолданылған басқа өнімдермен дұрыс біріктіру (Django).

SQLite studio-ның артықшылықтарына мыналар жатады: ақысыз қол жетімді, барлық операциялық жүйелер үшін қолжетімділік, әртүрлі форматтарда импорттау және экспорттау мүмкіндігі, орыс тіліндегі және әртүрлі қондырмалар. Бағдарламаның интерфейсі мен қолданылуы қарапайым, сонымен қатар ол жергілікті. Файлдарды ашу және орнату үшін бірнеше минут қажет болады.

Төменде 2.5-сурет sqlite архитектурасы ахитетктурасы көрсетілген :

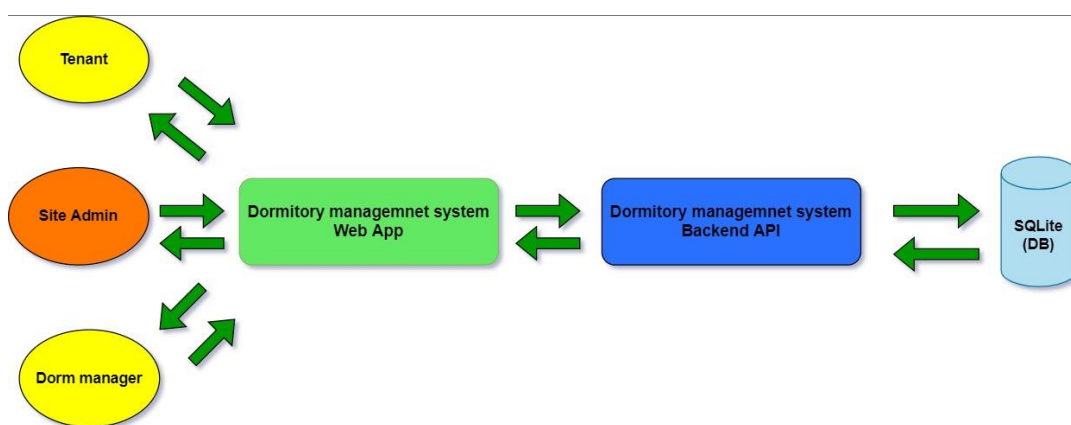


2.5-сурет – Sqlite архитектурасы

3 Жобаның құрылымы

3.1 «Университет жатақханасы» жобасы туралы

Бұл жоба бір веб қосымшадан тұрады. «Университет жатақханасы» тенант яғни студенттердің университет жатақханасына өтініш жасауына, қалалардағы жоғары оқу жатақханаларының жағдайына шолу жасауына икемділігін тигізеді. Және бұл тек жатақхана қабылдау өтініш процесстерімен шектеледі. Атап айтсақ тұрғын мәселесін шешуге ғана жауап береді. Веб қосымша деректерінің айналым жүйесін төменде 3.1-суреттен көруге болады. Байқап отырғанымыздай, бұл веб қосымша “site admin”, “org-manager”, “tenant” сынды үш түрлі қолданушыдан тұрады.

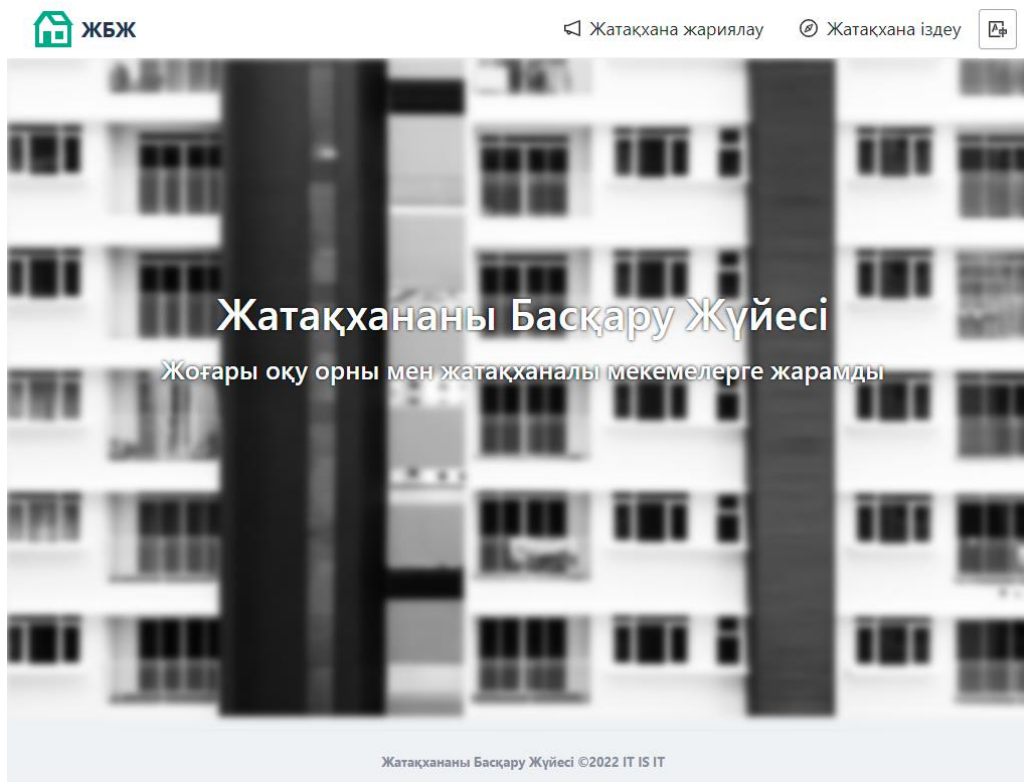


3.1-сурет – Жобаның өзара байланысу көрінісі

3.2 Веб қосымша интерфейстері

«Университет жатақханасы» веб қосымшасының интерфейсі басты бет және бірнеше қосымша беттен құралған, олар қарапайым тіркелу немесе кіру беті және ары қарай актер рөліне қарай ерекше функциялы бар беттерден ұрады. Беттер: басты бет, тіркелу немесе кіру беті, Тенант(жатақхана іздеу, өтініш қалдыру беттерінен), организация менеджері (организация құру, жатақхана құру, бөлме құру, өтініш қабылдау), сайт админ (қала құру және статика) беттерінен тұрады. Барлық актерда профил беті болады.

Басты беттегі “Жатақхана жариялау” батырмасын басу арқылы организация менеджері рөліне тіркелуге болады. Бұл университет және коледж сияқты жоғары оқу орындарының жатақхана админдеріне арналған. ж «Жатақхана іздеу» батырмасын басу арқылы тенантқа, яғни студент рөліне тіркеле аласыз. Төменде 3.2-суретте көрсетілген.



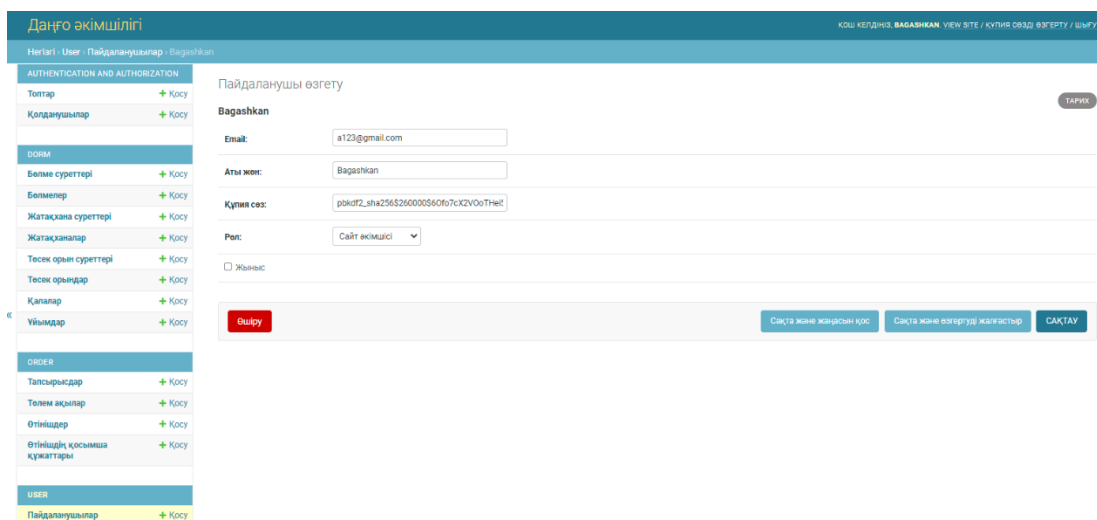
3.2-сурет – Басты бет көрінісі

«Жатақхана жариялау» немесе «Жатақхана іздеу» батырмасын басқан соң, тіркелу немесе кіру беті ашылады. Бұл бетте организация менеджері болсын немесе студент-тенант болсын өзіне керекті батырманы дұрыс таңдау қажет. Формаға сәйкес жолдарды толтыру арқылы өз рөліндегі бетке кіре алады. Төменде тіркелу немесе кіру беті 3.3-суретте көрсетілген.

3.3-сурет – Тіркелу немесе кіру беті

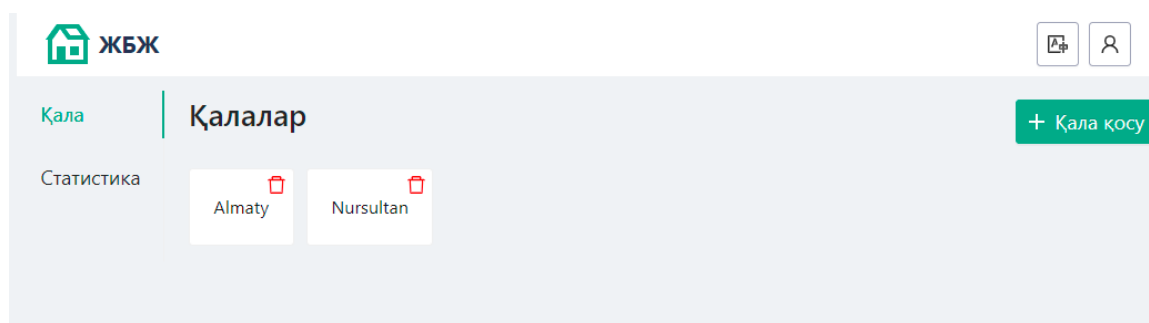
«Университет жатақханасы» веб қосымшасының негізгі идеясына келсек бізде ең бірінші джанго әкімшілігі арқылы сайт әкімшілігін құрып аламыз. Сайт әкімшілігі қаланы қосқаннан кейін барып сол қаладағы организациялар

өздерінің жатақханаларын біздің веб қосымшамызға жүктеу туралы өтініш бере алады. Төмнде джанго әкімшілігі арқылы сайт әкімшілігін қосқанымыз жайында 3.4-суретте көрсетілген.

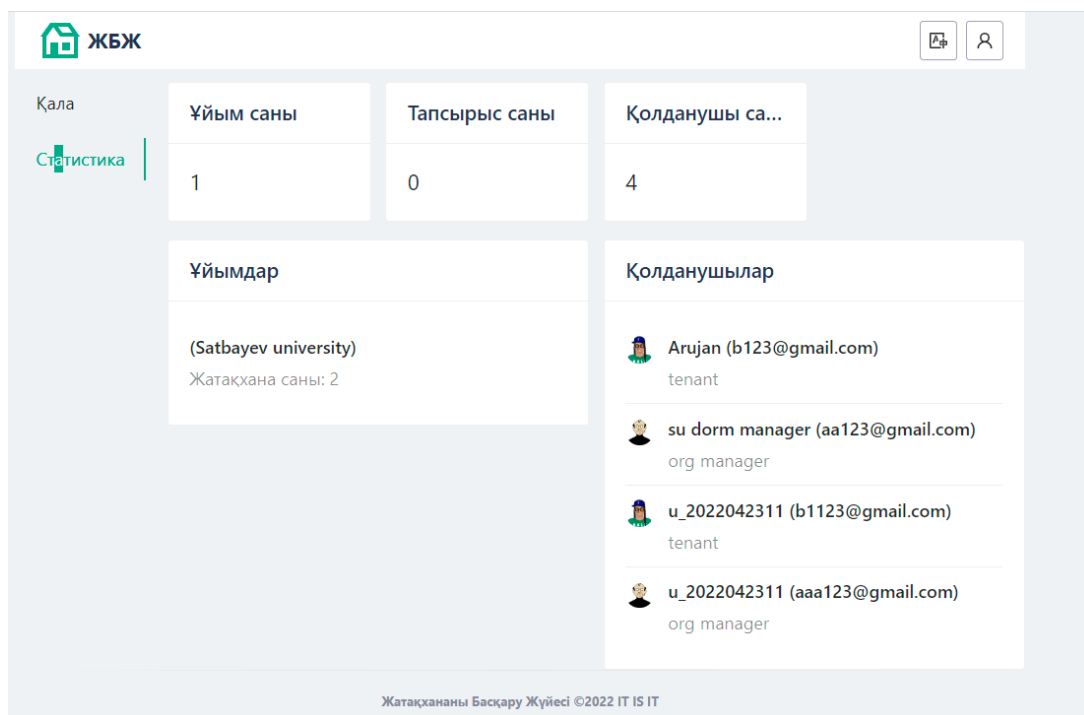


3.4-сурет – Джанго әкімшілігі

Сайт әкімшілігі қаланы қосады ол төменде 3.5-суретте көрсетілген және осы веб қосымшадағы статикаларды көріп отыра алады, оның ішінде ұйым саны, тапсырыс саны, қолданушы саны, ұйымдар, қолданушы ізімдері болады. ол 3.6-суретте көрсетілген.

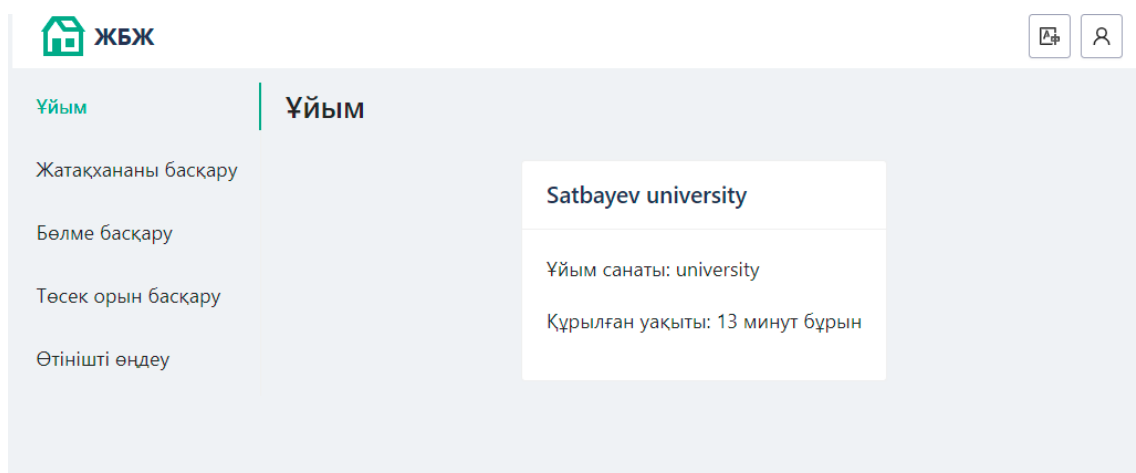


3.5-сурет – Сайт әкімшілігінің қала қосу функциясы

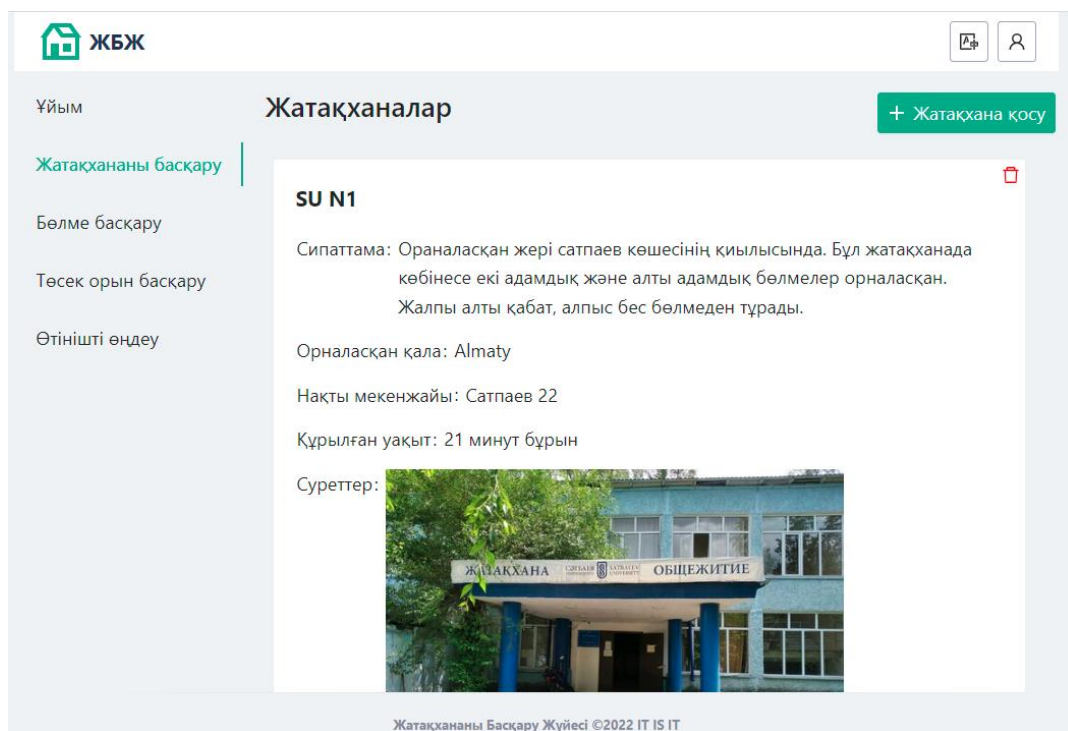


3.6-сурет – Сайт әкімшілігінің статика бөлімі

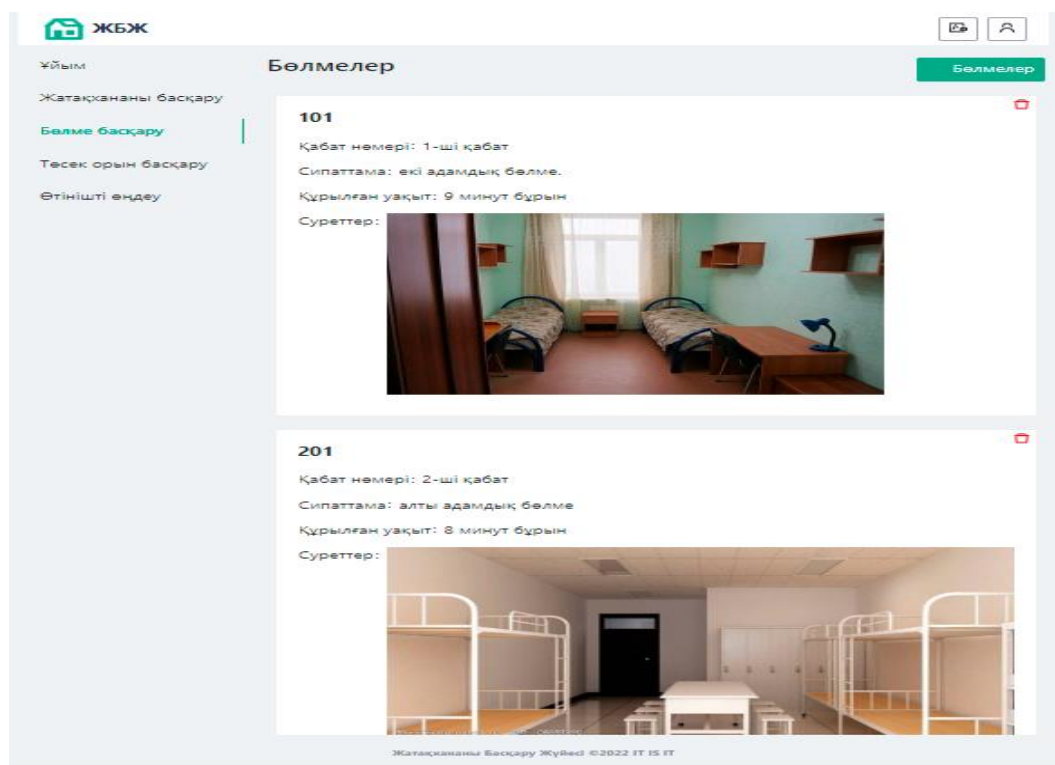
Енді организация менеджеріне келсек, ода өз ұйымын қосу(бір организация менеджерінде тек бір ұйым болады) бұл төменде 3.7-суретте көрсетілген, жатақхананы басқару(санға шектеу қойылмаған) бұл төменде 3.8-суретте көрсетілген, бөлме басқару бұл төменде 3.9-суретте көрсетілген, төсек орын басқару бұл төменде 3.10-суретте көрсетілген. Бұл суретте көргеніміздей егер организация менеджері студенттің өтінішін қабылдаған болса төсек орында сол студенттің аты-жөні тұрады. Және өтінішті өңдеу 3.11- суретте көрсетілген, мұнда біз келген өтініштерді көре аламыз.



3.7-сурет – Ұйым қосу



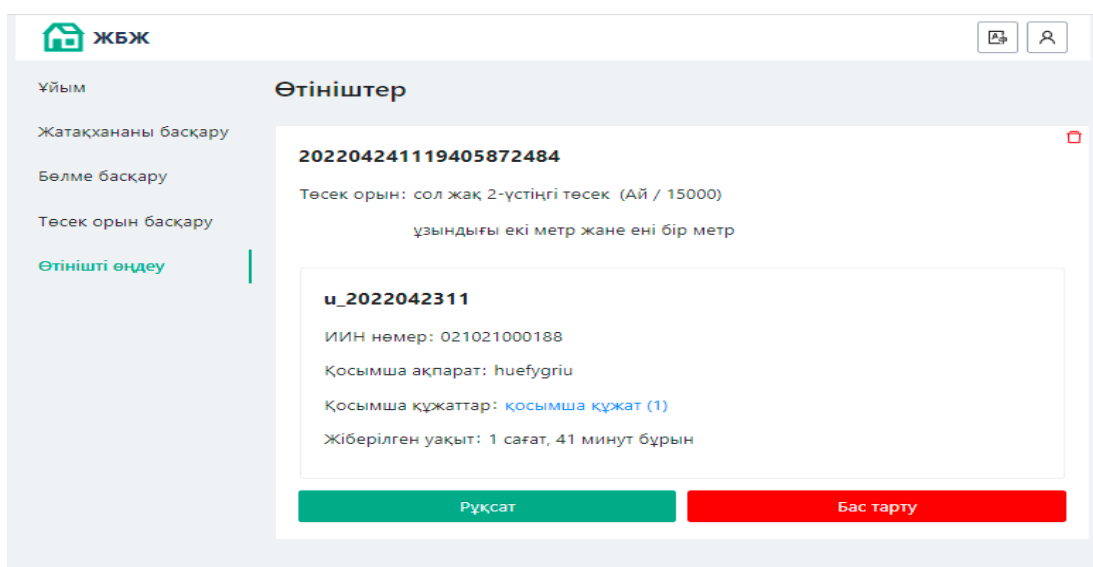
3.8-сурет – Жатақхананы басқару(құру немесе өшіру)



3.9-сурет – Бөлмені басқару



3.10-сурет – Төсек орынды басқару



3.11-сурет – Өтінішті өңдеу

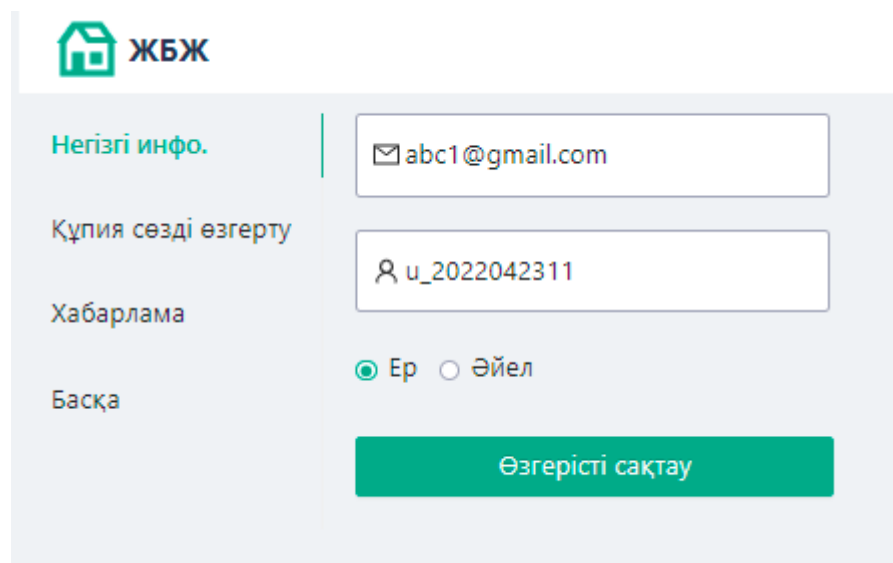
Тенант яғни студент жатақханаға өтініш беру үшін атын және ЖСН номерін жазса жеткілікті. Егер ерекше жағдайы бар студент болса құжат жүктеу орынына дәлел келтіретін құжатты жүктесе жеткілікті. Бұл веб қосымшада бізге бірінші өтінішті толтырып алып сосын жатақхана іздеу батырмасын басып керекті организация атын жазу арқылы тауып сосын өтініш қалдыру батырмасын бассақ болғаны. Өтініш беру батырмасын басқаннан кейін толтыру керек ақпараттар шығады. Ол төменде 3.12-суретте көрсетілген.

3.12-сурет – Өтініш қалдыру

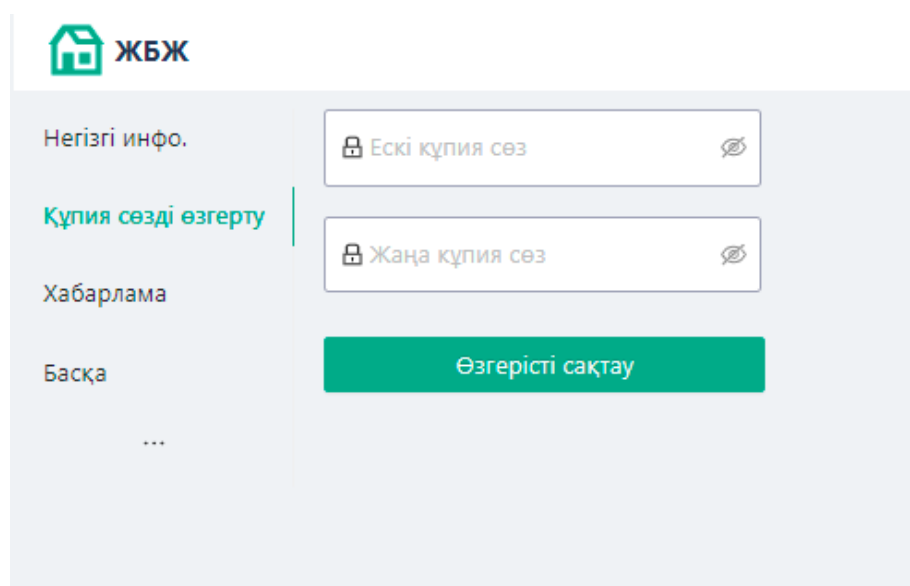
Жатақханаіздеу іздеу батырмасын басқаннан кейін бізге жатақханалар көрінеді. Қолайлы болуы үшін іздеу батырмасын басып ішіне керекті Университет атын жазсақ пайда болады. Бұл кезде тек өтініш қалдырсақ болғаны, өтінішіміз осы университеттің организация менеджеріне түседі де тек рұқсат етеді немесе отказ жасайды. Бұл төменде 3.13-суретте көрсетілген.

3.13-сурет – Жатақхана таңдау

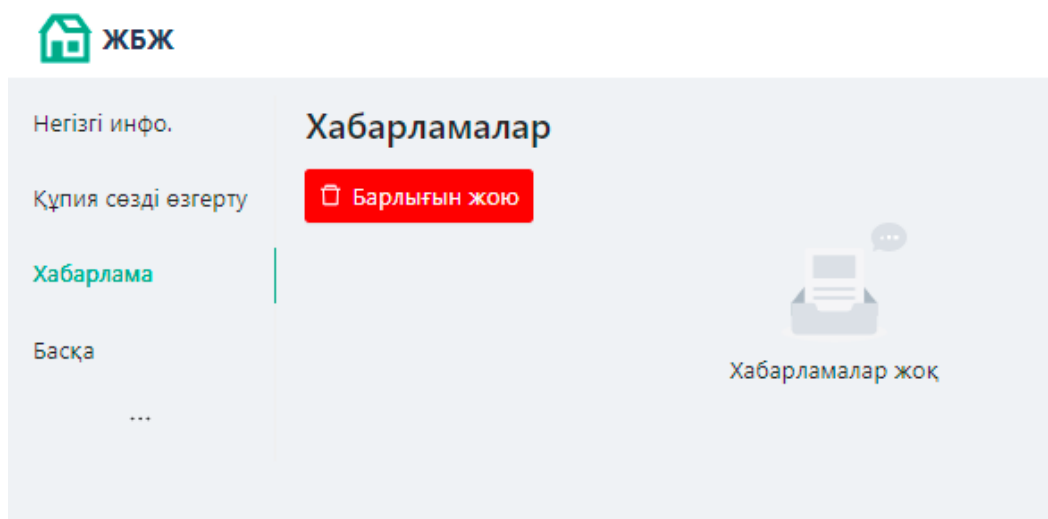
Ең соңында барлық актерде бірдей бет ол профил. Төменде оның бірнеше батырмасы және ерекшеліктері көрсетілген.



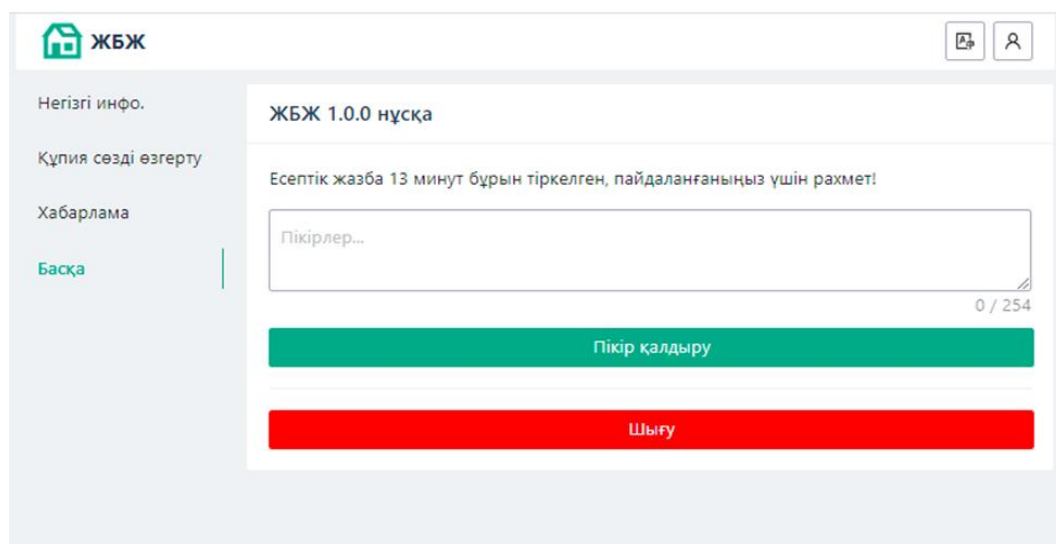
3.14-сурет – Негізгі инфо



3.15-сурет – Құпия сөзді өзгрту



3.16-сурет – Кері байланыс(хабарламалар)



3.17-сурет – Шығыу

Қорытынды

«Қазіргі әлемде қандай да бір тауарды жарнамалау, құру, ілгерілету саласында жұмыс істейтін оқу орындары, кәсіпорындар, компаниялар ғаламторда өз орнын алуға ұмтылады. Егер білім беру саласын қарастырсақ, балабақшалардан, мектептерден, колледждерден бастап және жоғары оқу орындары туралы айтпағанда, бүгінгі таңда өзінің Web сайттары бар (ағылш. web – тор, желі және site – орын) [12]. Сол сияқты менде университет жатақханаларының өзінің ғаламторда дұрыс орынын тапқанын мақсат еттім. Жұмыстың басты миссиялары туралы толғандым.

Жұмыстың мақсаты Университет жатақханаларын басқару, қағазбастылықты жою және бір райондағы бос жатақханаларды өз орынын табуға көмектесу болды.

Қойылған міндеттерге сәйкес Деректер базасының құрылымы, тауарлар шығындарын талдау үшін «Университет жатақханасы» web-қосымшасы Visual Studio Code әзірлеу құралы көмегімен жасалды. Жасалынған веб қосымша арқылы төмендегідей мүмкіндіктер бере алады:

Бір немесе бірнеше адам авторизациялану арқылы бір уақытта жұмыс істей алады.

Әр кірген адамның жеке функционалды орны бар.

Веб бет арқылы қосымшадағы ақпараттарды құру, жою, қайта енгізу, өз рөлі өзгерту мүмкіндіктерін қолдануға болады.

Әрбір жатақхана жайлы толық ақпарат.

Жатақхана күйін бақылай алу мүмкіндігін береді.

Барлық жоғары оқу орынының жатақханаларын бір орынға жинау және бір веб қосымшада жұмыс жасау өте тимді және актуальді болады. Ең бастысы алдыңғы қадамдарда біз Университет жатақханасы теориясын толықтай аштық. Яғни ең танымал фреймворк және кітапханаларды қолданып веб қосымшаның маңыздылығын арттырдық десек те болады. Веб қосымша «Университет жатақханасы» web-қосымшасы атыменде ерекшеленеді. Оған тоқталсақ Университет сөзімен толық қанды барлық қаладағы барлық университеттерді қамтыдық. Ал оның ішінде қалаға және организацияға бөлу арқылы түсінікті тармақтарға бөлдік. Веб қосымша жатақхана басқару үшін өте үлкен көмек береді деген ойдамын.

Қолданылған әдебиеттер тізімі

- 1 Тема: унифицированный язык моделирования UML // Сайттың электрондық нұсқасы <http://samara.mgpu.ru/~dzhadzha/dis/15/200.html>
- 2 Диаграмма прецедентов // Сайттың электронды нұсқасы [https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D0%BF%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82%D0%BE%D0%B2#:~:text=use%20case%20diagram\)%20%D0%B2%20UML,%D0%BE%D0%BF%D0%B8%D1%81%D0%B0%D1%82%D1%8C%20%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%83%20%D0%BD%D0%B0%20%D0%BA%D0%BE%D0%BD%D1%86%D0%B5%D0%BF%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%BC%20%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D0%B5](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D0%BF%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82%D0%BE%D0%B2#:~:text=use%20case%20diagram)%20%D0%B2%20UML,%D0%BE%D0%BF%D0%B8%D1%81%D0%B0%D1%82%D1%8C%20%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%83%20%D0%BD%D0%B0%20%D0%BA%D0%BE%D0%BD%D1%86%D0%B5%D0%BF%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%BC%20%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D0%B5).
- 3 <https://santoryo.net/kk/what-libraries-to-use-with-react>
- 4 <https://www.runoob.com/react/react-jsx.html>
- 5 <https://medium.com/nuances-of-programming/react-single-page-application-%D0%B8-react-router-%D0%B4%D0%BB%D1%8F-%D0%BD%D0%B0%D1%87%D0%B8%D0%BD%D0%B0%D1%8E%D1%89%D0%B8%D1%85-2bee95d37553>
- 6 https://www.google.com/search?q=%D0%A7%D1%82%D0%BE+%D1%82%D0%B0%D0%BA%D0%BE%D0%B5+Django+%D0%BF%D1%80%D0%BE%D1%81%D1%82%D1%8B%D0%BC%D0%B8+%D1%81%D0%BB%D0%BE%D0%B2%D0%B0%D0%BC%D0%B8?&source=lmns&bih=789&biw=1600&rlz=1C1ZKTGRuKZ972KZ972&hl=ru&sa=X&ved=2ahUKEwilwbbbprb3AhWxs4sKHZoSb1AQ_AUoAHoECAEQAA
- 7 <https://www.shuzhiduo.com/A/QV5Z3YObdy/>
- 8 Django ORM Cookbook 中文版 https://django-orm-cookbook-zh-cn.readthedocs.io/zh_CN/latest/query.html
- 9 <https://frontarm.com/james-k-nelson/controller-components/>
- 10 <https://xxwwp.github.io/docs/blogs/%E5%A6%82%E4%BD%95%E6%9B%B4%E5%A5%BD%E7%9A%84%E5%9C%A8%20react%20%E4%B8%AD%E4%BD%BF%E7%94%A8%20axios%20%E7%9A%84%E6%8B%A6%E6%88%AA%E5%99%A8#%E5%89%8D%E8%A8%80>
- 11 <https://help-desk-migration.com/help/convert-sqlite-to-other-databases/>
- 12 Майгүл Жекембаева. №1 дәріс. Веб бағдарламалау негіздеріне кіріспе.
- 13 Matt Frisbe. JavaScript 高级程序设计第4版.PDF.DOM.4016.
- 14 Matt Frisbe. JavaScript 高级程序设计第4版.PDF. Document 类型 . 4076.
- 15 Matt Frisbe. JavaScript 高级程序设计第4版.PDF. 表单基础.5816.

- 16 Gerd Wagner. JavaScript Front-End Web App Tutorial Part 1: Building a Minimal App in Seven Steps 16.
- 17 MVC Framework – Controllers.
https://www.tutorialspoint.com/mvc_framework/mvc_framework_controllers.htm
- 18 MVC framework- Layout.
https://www.tutorialspoint.com/mvc_framework/mvc_framework_layouts.htm
- 19 MVC Framework-Ajax support.
https://www.tutorialspoint.com/mvc_framework/mvc_framework_ajax_support.htm
- 20 Agiliq. Django ORM Cookbook Documentation Release 2.0
<https://readthedocs.org/projects/django-orm-cookbook/downloads/pdf/latest/>
- 21 Matt Frisbe. JavaScript 高级程序设计第 4 版.PDF.Straems Api.6376.

А Қосымшасы

(міндетті)

«Университет жатақханасы» веб-қосымшасын құруға арналған техникалық тапсырма

А.1 Кіріспе

Қазіргі таңда, дүние жүзінде көптеген жаңа технологиялардың қарқынды даму кезеңінде, және соның ішінде информатикалық технологиялар жаңадан шықпасада осы жаңа ғылымдардың ішінен өз орындарын табуда. Тіпті мектеп программасының өзіне программалық тілдерді қосуда. Яғни болашақта адамзаттың жартысынан бөлігі қызметтерін ауыстырады деп түсінсекте болады. Не үшін ИТ қарқынды дамуда? 21-ғасыр цифрлық ғасыр деп неге айтамыз? Юарлық сұрақтың жауабы бір ғана. Ол-уақыт. Заман дамыған сайын уақыттың қадірін біліп келе жатрмыз. Бұрынғыдай ұзын сонар кезекке тұрмаймыз, документті де әрбір бөлімнен қағазбен шығарып алып жол жүріп жүрмейміз. Бұның барлығы да ИТ саласының жемісі. Соның ішінде веб қосымша қазіргі таңда алдыңғы қатарда көрінуде..

Веб-қосымшалар тек компьютерлерде ғана емес смартфондардада жақсы жұмыс жасайтын ерекшелігіне сай мен «Университет жатақханасы» веб-қосымшасын әзрлеуге көштім. Мұнда студенттер веб қосымша арқылы кіріп жатақхананы таңдап сосын өтініш береді. Өтініштің ішіне өзінің мамандығын және ЖСН жазу міндетті, одан қалса ерекше жағдайы бар стуенттер сол жағдайын растайтын құжаттарды құжат папкасының ішіне салу арқылы растай алады..

«Университет жатақханасы» веб-қосымшасын негізінен әр қаладағы әр университеттегі студенттер төсек орынға тапсырыс бере алды.

А.1.1 Жобаның мақсаты мен міндеті

Бұл жобада негізінен студенттерге арналған веб қосымша болғандықтан мақсатым оңай интерфейсті және тез жұмыс жасайтын веб қосымша құру. Міндеті: студенттерге жатақханаға орналасу жолын жеңілдету.

А.1.2 Қолдану саласы

Қолдану саласы — кез-келген мобильді смартфон немесе компьютер браузерлерін қолдана білетін қолданушылар пайдалана алады.

А Қосымшаның жалғасы

А.1.3 Анықтамалар, терминдер және қысқартулар

Анықтамалар, терминдер және қысқартулар А.1 – кестесінде көрсетілген.

Терминдер немесе қысқартулар	Анықтамалар
Браузер	Гипермәтіндік байланысу мүмкіндігі бар файлдармен жұмыс стеуге арналған бағдарлама. Бұл әрімнің таңдауы бойынша әдеттегі мәтіннің, графиктердің немесе бейненің экранға бейнеленуін және олармен байланысқан басқа файлды іздестіруді қамтамасыз етеді.
Фреймворк	Сайтты жедел жасауға арналған құралдар жиынтығы, белгілі бір міндеттерді шешуге бағытталған. Ол бағдарлама өнімін немесе сайтты әзірлеуді жеңілдететін көптеген түрлі кітапханалардан тұратын жүйе. Яғни фреймворк- бірнеше кітапханалардың жиынтығы десекте болады.

А.1.4 Веб қосымшаның бейімділігі (Адаптация)

«Университет жатақханасы» веб-қосымшасын тек қана Windows ОЖ-де ғана емес, барлық браузерде ашылатын құрылғыларға, смартфон секілді техникаға бейімделген.

А Қосымшаның жалғасы

А.2 Жалпы сипаттамасы

А.2.1 Пайдаланушы интерфейстер

Веб-қосымша барынша қарапайым интерфейс тәнін ұстана жасалды, яғни қолданушыға барлық қол жетімді ыңғайлы интерфейс әзірленді.

А.2.2 Аппараттық интерфейстер

Құрылғыға қойылатын жалпы талаптар:

- кез-келген браузердің өзекті версиясы;
- ғаламторға қол жетімділік;
- ақпарат енгізу және басқару құрылғылары;

А.2.3 Программалық интерфейстер

- VS code – бастапқы кодты өзгертетін редактор;
- Node.js – серверлік бағдарламасы;
- Sqlite – sqlite деректер қорындағы тиесілі кластермен жұмыс, және оны бақылау бағдарламасы;
- GitHub Desktop – нұсқаларды басқару, репозиторияны өзгерту бағдарламасы;

А.2.4 Коммуникациялық интерфейстер

Қолданушылар орташа не жоғарғы жылдамдықтағы интернет желісімен қамтамасыз етілуі шарт. Сервер және клиенттерді байланыстыратын TCP/IP протоколы болып табылады.

А.2.5 Программалық интерфейстер

Жобаға қатысты бағдарламалық компоненттер:

- Windows 10 операциялық жүйесі (соңғы жаңартудағы нұсқа);
- Visual Studio Code- Бастапқы кодты өзгертетін редактор;
- Google Chrome браузері;

Б Қосымшасы (міндетті)

Бағдарлама мәтіні

Guest-landing page

```
import React from 'react';
import { useTranslation } from 'react-i18next';

import { CommonLayout } from '@layout';
import { LandingPageStyledBox } from './style';

import { welcomeBG } from '@assets/image';

export default function LandingPage() {
  const { t } = useTranslation();
  return (
    <CommonLayout>
      <LandingPageStyledBox>
        <div className="welcome">
          <img src={welcomeBG} alt="background image" />
          <div className="info">
            <h1
              className="title">{t('footer_site_name')}</h1>
            <h3
              className="sub-title">{t('landingP_welcome_subTitle')}</h3>
          </div>
        </div>
      </LandingPageStyledBox>
    </CommonLayout>
  );
}
```

Б қосымшаның жалғасы

Бағдарлама мәтіні:

Site admin-home page

```
import React from 'react';
```

```
import { useTranslation } from 'react-i18next';
```

```
import { useRecoilValue } from 'recoil';
```

```
import { pageAtom } from '@store';
```

```
import { useCreation, useResponsive } from 'ahooks';
```

```
import { Tabs } from 'antd';
```

```
import { CommonLayout } from '@layout';
```

```
import City from './City';
```

```
import Statistic from './Statistic';
```

```
import { HomePageStyledBox } from './style';
```

```
const { TabPane } = Tabs;
```

```
export default function HomePage() {
```

```
  const { t } = useTranslation();
```

```
  const responsive = useResponsive();
```

```
  const page = useRecoilValue(pageAtom);
```

```
  const siteAdminTabs = useCreation(
```

```
    () => [
```

```
      { key: 'city',
```

Б қосымшаның жалғасы

```
      tabName: t('site_admin_tabName_city'),
      tabContent: <City />,
    },
    {
      key: 'statistic',
      tabName: t('site_admin_tabName_statistic'),
      tabContent: <Statistic />,
    },
  ],
  [page.locale],
);

return (
  <CommonLayout>
    <HomePageStyledBox>
      <Tabs
        centered
        tabPosition={responsive.md === false ? 'top' : 'left'}
        className="site-admin-tabs">
        {siteAdminTabs.map(({ key, tabName, tabContent }) => (
          <TabPane tab={tabName} key={key}>
            {tabContent}
          </TabPane>
        ))}
      </Tabs>
    </HomePageStyledBox>
```


Б қосымшаның жалғасы

```
</CommonLayout>
);
}
```

Бағдарламалық мәтін:

Org-manager

```
import React from 'react';
import { useTranslation } from 'react-i18next';
import { useRecoilValue, useSetRecoilState, useRecoilState } from 'recoil';
import { pageAtom, userAtom, dormAtom } from '@store';

import { useCreation, useResponsive, useRequest, useMount } from 'ahooks';
import {
  reqGetMyOrgaization,
  reqGetCities,
  reqGetDorms,
  reqGetRooms,
} from '@service/api/org-manager-api';

import { Tabs, message as antdMessage } from 'antd';
import { CommonLayout } from '@layout';
import Organization from './Organization';
import DormitoryManagement from './DormitoryManagement';
import RoomManagement from './RoomManagement';
import BedManagement from './BedManagement';
import ProcessRequest from './ProcessRequest';
```

Б қосымшаның жалғасы

```
import { HomePageStyledBox } from './style';
const { TabPane } = Tabs;
export default function HomePage() {
  const { t } = useTranslation();
  const setUser = useSetRecoilState(userAtom);
  const responsive = useResponsive();
  const page = useRecoilValue(pageAtom);
  const [dorm, setDorm] = useRecoilState(dormAtom);

  // 获取机构
  const { runAsync: runReqGetMyOrganization } = useRequest(
    () => reqGetMyOrgaization(),
    {
      manual: true,
    },
  );

  // 获取城市
  const { runAsync: runReqGetCities } = useRequest(() => reqGetCities(), {
    manual: true,
  });

  // 获取宿舍
  const { runAsync: runReqGetDorms } = useRequest(() => reqGetDorms(), {
    manual: true,
  });

  // 获取房间
  const { runAsync: runReqGetRooms } = useRequest(() => reqGetRooms(), {
```

Б қосымшаның жалғасы

```
        manual: true,
    });

    useMount(() => {
        Promise.all([
            runReqGetMyOrganization(),
            runReqGetCities(),
            runReqGetDorms(),
            runReqGetRooms(),
        ])
        .then(([ { organization }, { cities }, { dorms }, { rooms } ]) => {
            setDorm({
                organization,
                hasCity: Boolean(cities.length),
                hasDorm: Boolean(dorms.length),
                hasRoom: Boolean(rooms.length),
            });
        })
        .catch(({ message, needExecuteLogout, initialUser }) => {
            antdMessage.error(message);

            if (needExecuteLogout) {
                setUser(initialUser);
            }
        });
    });
```

Б қосымшаның жалғасы

```
const orgManagerTabs = useCreation(
  () => [
    {
      key: 'organization',
      tabName: t('org_manager_tabName_organization'),
      tabContent: <Organization />,
    },
    {
      key: 'dormitory_management',
      tabName:
t('org_manager_tabName_dormitory_management'),
      tabContent: <DormitoryManagement />,
      disabled: Boolean(!dorm.organization),
    },
    {
      key: 'room_management',
      tabName: t('org_manager_tabName_room_management'),
      tabContent: <RoomManagement />,
      disabled: !dorm.hasDorm,
    },
    {
      key: 'bed_management',
      tabName: t('org_manager_tabName_bed_management'),
      tabContent: <BedManagement />,
      disabled: !dorm.hasRoom,
    },
  ],
)
```

Б қосымшаның жалғасы

```
        key: 'process_request',
        tabName: t('org_manager_tabName_process_request'),
        tabContent: <ProcessRequest />,
        disabled: !dorm.hasRoom,
    },
],
[page.locale, dorm],
);

return (
    <CommonLayout>
        <HomePageStyledBox>
            <Tabs
                centered
                tabPosition={responsive.md === false ? 'top' : 'left'}
                className="org-manager-tabs">
                {orgManagerTabs.map(({ key, tabName,
tabContent, disabled }) => (
                    <TabPane tab={tabName} key={key}
disabled={disabled}>
                        {tabContent}
                    </TabPane>
                )))}
            </Tabs>
        </HomePageStyledBox>
    </CommonLayout>
```

Б қосымшаның жалғасы

Бағдарламалық мәтін:

Tenant

```
import React from 'react';
import { useTranslation } from 'react-i18next';
import { useRecoilValue } from 'recoil';
import { pageAtom } from '@store';

import { useCreation, useResponsive } from 'ahooks';

import { Tabs } from 'antd';
import { CommonLayout } from '@layout';
import Dormitory from './Dormitory';
import Request from './Request';
import { HomePageStyledBox } from './style';

const { TabPane } = Tabs;

export default function HomePage() {
  const { t } = useTranslation();
  const responsive = useResponsive();
  const page = useRecoilValue(pageAtom);

  const dromManagerTabs = useCreation(
    () => [
      {
        key: 'dormitory_management',
```

Б қосымшаның жалғасы

```
tabName: t('tenant_tabName_search_dorm'),
    tabContent: <Dormitory />,
  },
  {
    key: 'process_request',
    tabName: t('tenant_tabName_info'),
    tabContent: <Request />,
  },
],
[page.locale],
);

return (
  <CommonLayout>
    <HomePageStyledBox>
      <Tabs
        centered
        tabPosition={responsive.md === false ? 'top' : 'left'}
        className="tenant-tabs">
        {dromManagerTabs.map(({ key, tabName,
tabContent }) => (
          <TabPane tab={tabName} key={key}>
            {tabContent}
          </TabPane>
        ))}
      </Tabs>
```

Б қосымшаның жалғасы

```
</HomePageStyledBox>  
    </CommonLayout>  
    );  
}
```