ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ БІЛІМ МИНИСТРЛІГІ

Қ.И.Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті

Автоматика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

Қаирбаева Дильназ Амантайқызы

Қоймадағы товарларды енгізу және қадағалау үшін Web қосымша жасау

**ТҮСІНДІРМЕ ЖАЗБА**

дипломдық жобаға

6B06102 – «Computer science»

Алматы 2024

**АҢДАТПА**

Менің дипломдық жобамда қойма товарларын енгізу және қадағалау үшін автоматтандырылған «StoreHouse» web-қосымшасы қарастырылады. Web-қосымша деректерді тиімді енгізуді және қоймаларды басқару үшін дайын Word есептерін жасауды қамтамасыз ететін қойма хатшысының қажеттілігін автоматтандыруға және жеңілдетуге бағытталған.

Дипломдық жұмыс кіріспе, негізгі төрт бөлімнен, қорытынды және пайдаланылған әдебиеттерден құралған:

* кіріспеде тақырыптың өзектілігі талқыланады, жобаның негізгі мақсаттары мен міндеттері айқындалады.
* бірінші бөлімде хатшының құжаттарды толтыруынан бастап негізгі тұжырымдамаларға дейінгі бүкіл процесті сипаттайды.
* екінші бөлімде жобаға қажетті таңдалған технологиялар мен оны таңдау себептері жазылды.
* үшінші бөлімде электронды документтерді толтыру процесінің тізбектей орындалу шарттары, кесте, диаграммалар мен олар туралы толық ақпарат берілген. Жоба құру барысындағы ең басты бөлім ол осы модельдеу бөлімі болып табылады.
* төртінші бөлім программалау бөлімі. Қоймаға товар келген жағдайда автоматты түрде толтыру және қорытынды есептеулер файл түрінде .docx форматында жүктелетін веб-қосымшаның қорытынды интерфейсі скриншоттармен бейнеленген.

«Қоймадағы товарларды енгізу және қадағалау үшін Web қосымша жаса» жобасының түсініктемесі жалпы 00 бет, оның ішінде 00 сурет, 0 қосымшадан тұрады. Жұмысты жазуға 00 пайдаланылған әдебиеттер мен сілтемелер қолданылды.

**АНОТАЦИЯ**

В моем дипломном проекте рассматривается автоматизированное web-приложение «StoreHouse» для ввода и отслеживания складских товаров. Web-приложение направлено на автоматизацию и облегчение необходимости секретаря склада, который обеспечивает эффективный ввод данных и создание готовых отчетов Word для управления хранилищами.

Дипломная работа состоит из введения, четырех основных разделов, заключения и использованной литературы:

* во введении обсуждается актуальность темы, определяются основные цели и задачи проекта.
* в первой части описывается весь процесс, начиная от заполнения документов секретарем и заканчивая основными понятиями.
* во второй части прописаны выбранные технологии, необходимые для проекта, и причины его выбора.
* во третем разделе представлены условия последовательного выполнения процесса заполнения электронных документов, таблицы, диаграммы и подробная информация о них. Самым главным разделом в процессе создания проекта является этот раздел моделирования.
* четвертая часть раздел программирования. В случае поступления товара на склад автоматическое пополнение и окончательные расчеты в виде файла .итоговый интерфейс загружаемого веб-приложения в формате docx иллюстрируется скриншотами.

Комментарий проекта "Разработка Web приложения для ввода и отслеживание складских запасов" состоит из 00 страниц, в том числе 00 изображений, 0 приложений. 00 Использованная литература и ссылки были использованы для написания работы.

**ANNOTATION**

My graduation project considers an automated web application «StoreHouse» for entering and tracking warehouse goods. The web application is aimed at automating and facilitating the need for a warehouse secretary, who provides efficient data entry and the creation of ready-made Word reports for storage management.

The thesis consists of an introduction, four main sections, a conclusion and the literature used:

* the introduction discusses the relevance of the topic, defines the main goals and objectives of the project.
* the first part describes the whole process, starting from filling out documents by the secretary and ending with the basic concepts.
* in the second part, the selected technologies necessary for the project and the reasons for its choice are spelled out.
* the third section presents the conditions for the sequential completion of the process of filling out electronic documents, tables, diagrams and detailed information about them. The most important section in the process of creating a project is this section of modeling.
* the fourth part is the programming section. In case of receipt of the goods at the warehouse, automatic replenishment and final calculations in the form of a file.The final interface of the downloadable web application in docx format is illustrated with screenshots.

The comment of the project "Development of a Web application for entering and tracking inventory" consists of 00 pages, including 00 images, 0 applications. 00 Used literature and references were used to write the work.

**МАЗМҰНЫ**

|  |  |  |
| --- | --- | --- |
|  | Кіріспе | 9 |
| 1. | Негізгі бөлім | 1 |
| 1.1. | Веб-қосымша жасаудың негіздері | 1 |
| 1.2 | Жобаға жүргізілген салыстырмалы талдау |  |
| 1.3. | Құжаттама процесіне шолу | 1 |
| 2 | Технологиялар бөлімі | 1 |
| 2.1 | GitHub | 1 |
| 2.2 | Python. Django фреймворкі | 1 |
| 2.3 | PostgreSQL | 1 |
| 3. | Жобаны модельдеу бөлімі | 1 |
| 3.1. | Жүйе архитектурасы | 1 |
| 3.2. | Деректер қорының моделін жобалау | 1 |
| 3.3. | UML диаграммалары | 1 |
| 3.3.1. | Контексттік диаграмма | 1 |
| 3.3.2. | Прецеденттер диаграммасы | 1 |
| 3.3.3. | Тізбек диаграммасы | 1 |
| 3.3.4. | Күй диаграммасы | 1 |
| 3.3.5. | Класс диаграммасы | 1 |
| 3.3.6. | Ашып қарау диаграммасы | 1 |
| 3.3.7. | Компоненттер диаграммасы | 1 |
| 4. | Программалау бөлімі | 1 |
| 4.1. | Web-қосымшаны әзірлеу | 1 |
| 4.2. | Функционалды құрылымын жасау | 1 |
| 4.3. | Web-қосымшаның интерфейсі | 1 |
|  | Қорытынды | 1 |
|  | Қолданылған әдебиеттер тізімі | 1 |
|  | А қосымшасы. Техникалық тапсырма | 1 |
|  | Б қосымшасы. Бағдарлама мәтіні | 1 |

**КІРІСПЕ**

Бизнестің қазіргі қарқынды дамып келе жатқан әлемінде Қоймалық қорларды тиімді басқару кәсіпорынның жетістігін анықтайтын негізгі факторлардың біріне айналуда. Компанияның немесе саланың көлеміне қарамастан, тауарлы-материалдық құндылықтарды тиімді басқару үздіксіз жұмыс істеу және тұтынушылардың қажеттіліктерін қанағаттандыру үшін өте маңызды.

Дегенмен, тауарлы — материалдық құндылықтарды басқару тауарларды сөреге қою және оларды тұтынушыларға жөнелту ғана емес. Бұл сонымен қатар қойма деректерін жүйелі түрде жинауды, талдауды және түсіндіруді қажет ететін күрделі процесс. Бұл процестің негізгі элементі тауарлы-материалдық құндылықтар мен тауарларға сұраныс туралы өзекті ақпаратқа негізделген жедел шешім қабылдауды қамтамасыз ететін деректерді есепке алу және талдау жүйесі болып табылады.

Бұл тұрғыда біздің жоба арнайы қойма хатшысына арналған веб-қосымшаны әзірлеуге бағытталған. Біз тауарларды қабылдау және жөнелту деректерін енгізу, сондай-ақ дайын есептерді шығару процесін жеңілдетуге және оңтайландыруға көмектесетін құрал жасауға тырысамыз. Біздің қолданба бұл процесті ашық және қойма хатшылары үшін қолжетімді ету үшін қоймаларды басқарудың ыңғайлылығы мен тиімділігін арттыруға бағытталған.Дипломдық жоба өзектілігі: Қарқынды нарықтық өзгерістер мен бизнестің қарқынды темпі жағдайында қойма қорларын жедел және дәл есепке алу қажеттілігі өте маңызды болып отыр. Біздің жоба қойма хатшысының қажеттіліктеріне назар аудара отырып, деректерді басқарудың инновациялық тәсілін ұсына отырып, жоғары өзектілікке ие.

Біздің веб-қосымшамыз қойма хатшысына тауарларды қабылдау және жөнелту туралы мәліметтерді енгізу үшін ыңғайлы және интуитивті интерфейс ұсынады. Пайдаланушылар қоймадағы ағымдағы қалдықтарды оңай бақылай алады және сұраныс пен жеткізілім өзгерістеріне жедел жауап бере алады.

Біздің қосымшаның басты артықшылығы-Word форматында әртүрлі есептер шығару мүмкіндігі. Хатшы қойманың жай-күйі, тауарлардың қозғалысы, сатуды талдау және тауарлы-материалдық құндылықтарды басқарудың басқа да маңызды аспектілері туралы есептерді тез жасай алады. Бұл есептер кәсіпорын басшылығына ақпараттандырылған шешімдер қабылдауға және қойма процестерін оңтайландыруға мүмкіндік береді.

Біздің қолданба жоғары өнімділік пен деректер қауіпсіздігін қамтамасыз ететін заманауи веб-әзірлеу технологияларын пайдалана отырып әзірленеді. Біз сондай-ақ деректермен алмасу және процестерді автоматтандыру үшін кәсіпорынның басқа ақпараттық жүйелерімен интеграциялау мүмкіндігін қарастырамыз.

Сонымен қатар, біз пайдаланушыларды оқыту мен қолдауға ерекше назар аударамыз. Біз қойма хатшыларына Жаңа қолданбаны игеруге және оны күнделікті жұмыста пайдалануға көмектесу үшін егжей-тегжейлі құжаттама мен онлайн қолдауды ұсынамыз.

Біздің жобамыздың нәтижесінде біз қоймаларды басқару тиімділігін жақсартуды, деректерді өңдеу уақытын қысқартуды және қабылданған шешімдердің сапасын жақсартуды күтеміз. Біздің веб-қосымшамыз қойма хатшылары үшін тауарлық-материалдық құндылықтарды тиімді басқаруға және кәсіпорынның бизнес-процестерін тиімді қолдауға көмектесетін таптырмас құрал болады.

Жобаның негізгі аудиториясы – қойма хатшылары және қойма қорларын жедел есепке алуға және қадағалауға жауапты әкімшілік персонал болып табылады. Пайдаланушылар деректерді тиімді енгізу және дайын есептерді жасау үшін қолдануға оңай құрал алады.

Дипломдық жұмыстың мақсаты – қойма хатшыларының тауарлы-материалдық құндылықтарды ыңғайлы және тиімді басқару қажеттіліктерін қанағаттандыратын веб-қосымшаны құру.

Қойылған мақсатқа жету үшін келесі міндеттер орындалуы қажет:

* жүйемен интуитивті өзара әрекеттесуді қамтамасыз ететін пайдаланушы интерфейсін әзірлеу.
* тауарлардың түсуі және жөнелтілуі туралы деректерді енгізу тетігін құру.
* ақпаратқа сенімді және жылдам қол жеткізу үшін деректерді сақтау жүйесін енгізу.
* енгізілген деректер негізінде Word форматында есептер шығару механизмін әзірлеу.
* нақты жағдайларда оның тиімділігін бағалау үшін қолданбаны нақты пайдаланушылардың қатысуымен тестілеу.

Зерттеу объектісі – Python тілінің Django фреймворкында жазылған веб қосымша.

Мәселені талдау дәрежесі. Дипломдық жұмыстың тақырыбын зерттеу барысында отандық және шетелдік ғылыми еңбектер мен оқулықтар, электронды басылымдардағы ғылыми мақалалар қолданылды.

Зерттеу тәжірибесі: Салыстырмалық, аналитикалық, логикалық, құрылымдық жүйе, функционалдық жүйе, жалпылау тәжірибесі.

Веб-қосымшаны пайдаланудың осындай артықшылықтары бар:

* Қойма хатшылары үшін деректерді енгізудің жеңілдетілген процесі.
* Қойма туралы ақпаратқа жылдам және икемді қол жетімділік.
* Жедел талдау үшін Word форматында дайын есептерді құру.
* Тауарлы-материалдық қорларды басқару тиімділігін арттыру және бухгалтерлік есептегі қателіктер қаупін азайту.

1. **Негізгі бөлім**
   1. **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.

|  |  |
| --- | --- |
| Аббревиатура немесе термин | Анықтама |
| ДҚ | Деректер қоры |
| БҚБЖ | Деректер қорын басқару жүйесі |
| API | Application Programming Interface |
| UML | Unified Modeling Language |
| URL | Uniform Resource Locator |
| ER | Entity Relationship |
| JS | Java Script |
| Браузер | Веб-сайттарды қарауға арналған бағдарламалық жасақтама. |
| Веб-бағдарлама, веб-сайт, веб-қосымша | Интернет компьютерлік желісіндегі байланысқан электрондық құжаттардың (файлдардың) бір домендік атпен біріктірілген жиынтығы. |

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 пішімдеудіқамтамасыз 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 орнатудын қажеті жоқ;

– уақытылы жаңартулар оңай жасалынады;

– жадыда орын алмайды;

– беттер динамикалық түрде жаңарады;

– іздеу жүйелері мен тақырыптық каталогтарда тіркеу.

* 1. **Жобаға жүргізілген салыстырмалы талдау**

Есептерді толтырудың екі тәсілін салыстырмалы талдау – Microsoft Word арқылы қолмен және біздің веб-қосымшамызды қолдана отырып – қоймаларды басқару процесін автоматтандырудың артықшылықтары туралы құнды ақпарат береді.

Word арқылы есептерді қолмен толтыру кезінде хатшы бірқатар шектеулер мен мәселелерге тап болады. Біріншіден, бұл процесс айтарлықтай уақыт пен ресурстарды қажет етеді, өйткені әрбір есеп қолмен жасалуы, толтырылуы және пішімделуі керек. Бұл ақпаратты жаңартудың кешігуіне және деректерді өңдеу шығындарына әкелуі мүмкін. Сонымен қатар, есептерді қолмен толтыру кезінде қателіктер жіберу қаупі бар, бұл дәлсіздіктер мен дұрыс емес деректерге әкелуі мүмкін.

Керісінше, біздің веб-қосымшаны пайдалану есептерді толтыру процесін едәуір жеңілдетеді және тездетеді. Жүйе тауарлардың қоймадан түсуі мен жөнелтілуі туралы деректерді автоматты түрде жинайды, бұл хатшыға бірнеше рет басу арқылы есептер шығаруға мүмкіндік береді. Есептерді толтырудың автоматтандырылған процесінің арқасында қателер мен дәлсіздіктердің ықтималдығы азаяды, өйткені деректер тікелей көзден келеді.

Есептерді толтыру үшін веб-қосымшаны пайдаланудың артықшылықтары сонымен қатар қойманың нақты деректеріне жедел қол жеткізу мүмкіндігін және кәсіпорынның нақты қажеттіліктері үшін есеп форматтарын теңшеу икемділігін қамтиды. Сонымен қатар, жүйе деректерді өзгерткен сайын есептерді автоматты түрде жаңартады, бұл олардың өзектілігі мен сенімділігін қамтамасыз етеді.

Осылайша, біздің веб-қосымшамызды қолдана отырып, есептерді толтыру процесін автоматтандыру қоймаларды басқарудың тиімділігін едәуір жақсартады, уақытты азайтады және қателіктер қаупін азайтады, бұл тәсілді қойма хатшылары мен жалпы кәсіпорындар үшін қолайлы етеді.

Уақыт шығындарын азайтудан және қателер ықтималдығын азайтудан басқа, есептерді толтыру үшін веб-қосымшаны пайдалану деректердің жоғары дәлдігі мен сенімділігін қамтамасыз етеді. Бұл әсіресе тауарлық-материалдық құндылықтар туралы ақпараттың өзектілігі мен дәлдігі стратегиялық шешімдер қабылдау және өндірістік процестерді жоспарлау үшін өте маңызды компаниялар үшін өте маңызды. Деректерді жинау және есеп беру процесін автоматтандыру арқылы қойма хатшысы алынған ақпараттың әрқашан өзекті және сенімді екендігіне сенімді бола алады.

Сонымен қатар, веб-қосымша кәсіпорынның нақты қажеттіліктері үшін есеп форматтарын теңшеуде үлкен икемділік береді. Пайдаланушылар қажетті өрістерді, сүзгілерді және талдау опцияларын қосу арқылы есептерді өз компаниясының талаптарына сәйкес оңай реттей алады. Бұл есептерді бизнестің әртүрлі мақсаттары мен қажеттіліктеріне бейімдеуге мүмкіндік береді, бұл оларды ақпараттандырады және шешім қабылдау үшін пайдалы етеді.

Біздің веб-қосымшаны қолдана отырып, қойма хатшысы Word арқылы есептерді қолмен толтыру кезінде қол жетімді емес көптеген қосымша мүмкіндіктерге қол жеткізе алады. Мысалы, жүйе қойманың күйін тереңірек талдауға және жасырын трендтер мен үлгілерді анықтауға мүмкіндік беретін аналитикалық құралдар мен деректерді визуализациялау мүмкіндіктерін ұсына алады.

Осылайша, есептерді толтыру үшін веб-қосымшаны пайдалану процестің тиімділігі мен дәлдігін арттырып қана қоймайды, сонымен қатар қойма хатшысына қоймаларды басқаруды оңтайландыру және негізделген шешімдер қабылдау үшін қосымша құралдар мен мүмкіндіктер береді.

Осы құжаттарды толтыру процессі автоматтандырылған веб-қосымша мен құжаттарды қолдан толтыру процессін салыстыратын болсақ:

|  |  |  |
| --- | --- | --- |
|  | Құжаттарды Word арқылы қолмен толтыру | Веб-қосымшаны пайдалану |
| Уақыт шығындары | Жоғары | Төмен |
| Қателіктер ықтималдығы | Жоғары | Төмен |
| Ағымдағы деректердің қолжетімділігі | Шектеулі | Жоғары |
| Есептерді жөндеудегі икемділігі | Шектеулі | Жоғары |
| Жаңарту процесін автоматтандыру | Жоқ | Бар |
| Есептерді жаңарту | Қолмен | Автоматты |
| Ақпаратқа қол жеткізу жылдамдығы | Баяу | Жылдам |
| Масштабтау мүмкіндігі | Шектеулі | Жоғары |
| Кепілдендірілген үйлесімділік | Әрқашан емес | Ия |
| Ресурстардың құны | Жоғары | Төмен |

1.2.1-кесте Құжаттарды қолмен толтыру процессі мен автоматтандырылған жүйені салыстыру

Артықшылықтары мен кемшіліктерін салыстыру барысында автоматтандырылған веб-қосымша бірнеше есе құжаттарды қолмен жазудан тиімді екенін көруге болады.

* 1. **Құжаттама процесіне шолу**

Хатшы өзінің логин және құпия сөзін пайдаланы «StoreHouse» веб-қосымшасына кіреді, қоймада бар барлық товарларды көре алады. Қажет болған жағдайда жаңа товарларды қосып, қажет емес товарларды өшіре алады. Сонымен қатар, товардың деректерінде қате кеткен жағдайда, өзгерту батырмасы арқылы деректерін өзгерте алады. Товарлар туралы ақпарат бетінде сол жерден бір ғана батырманы басу арқылы қажет құжатты өзінің дербес компьютеріне жүктеп алады. Батырма басылған сәтте, бірнеше миллисекунд ішінде товарлар туралы деректер алынып, құжат дайын болады.

Басқа филиалдың админстраторлары өздерінің логин және құпия сөзін пайдаланып «StoreHouse» веб-қосымшасына кіреді, кірген сәтте қол жетімді, яғни қоймада бар, алып кетуге болатын товарлар браузер бетінде көрініп тұрады. Товардың жеке парағына өткен сәтте, товар туралы толық ақпаратты оқу мүмкіншілігі бар.

«StoreHouse» веб-қосымшасы бірқатар мәселелерді шешті:

* деректер мен құжаттар орталықтандырылған деректер қорында сақталған;
* түсінікті, ыңғайлы интерфейс арқылы керек ақпаратты табу оңай;
* құжаттарды толтыру процессінде деректерде қате кету алдын алынған;
* хатшының жұмысы автоматтандырылған;
* уақыт тиімді пайдаланылады.

1. **Техникалық бөлім**

**2.1. GitHub**

Қойма товарларын енгізу және қадағалау үшін автоматтандырылған «StoreHouse» web-қосымшасын әзірлеу барысында Git және GitHub платформаларын қолдандым.

Git бұл кодтағы қандай да бір өзгерістерді бақылауға мүмкіндік беретін басқару жүйесі. Бұл жүйе бағдарламалық жобаларының бастапқы кодындағы өзгерістерді бақылауға арналған. Ол бағдарламалық жобаларды әзірлеу саласында кеңінен қолданылады және тиімді түрде кодтың тарихына оралуға мүмкіндік береді.

Жобадағы өзгерістерді сақтау және git басқару жүйесін қолдану үшін GitHub платформасын қоладандым. GitHub -та жоба кодтарын, суреттерін және басқа да деректрерді, сондай-ақ жоба файлдарын сақтау үшін жеке репозиторий құрдым.

GitHub репазиторийде жобаның әртүрлі нұсқаларымен жеңіл жұмыс істеуге ыңғайлы веб интерфейсті қолдануға мүмкіндік береді.

Git және GitHub платформаларының арқасында код нұсқаларын бақылауға, жоба туралы деректерге және файлдарға оңай қол жеткізіп отырдым. Бұл кодтардың бастапқы кезіне оралуға және өз прогрессімді бақылау үшін тиімді болды.

**2.2. Python. Django фреймворкі**

Python-1980 жылдардың соңында пайда болған жоғары деңгейлі бағдарламалау тілі. Тілді дамытуды Гидо ван Россум жүргізді. Ол қарапайым және түсінікті бағдарламалау тілін жасағысы келген. Қазіргі таңда Python әлемдегі ең танымал бағдарламалау тілдерінің бірі.

Python бастаушы бағдарламашылар үшін өте қолайлы, себебі кодтың қарапайымдылығымен және оқылуымен ерекшеленеді. Ол сонымен қатар басқа бағдарламалау тілдеріне қарағанда қосымшаларды тезірек және оңай жасауға мүмкіндік береді. Python-да бай стандартты кітапхана және оның ғылыми есептеулер, деректерді талдау, Машиналық оқыту және веб-сайтты дамыту сияқты әртүрлі салаларда функционалдығын қамтамасыз ететін көптеген қосымша модульдер бар.

Python бірнеше бағдарламалау парадигмаларын қолдайды, соның ішінде процедуралық, объектіге бағытталған, Функционалды және т.б. бұл бағдарламалау тілі де кросс-платформа болып табылады, яғни Windows, Mac OS және Linux сияқты әртүрлі операциялық жүйелерде бірдей кодты іске қосуға болады.

Python сонымен қатар ғылыми және зерттеу салаларында кеңінен қолданылады, атап айтқанда физика, математика, биология және т.б. сияқты. Қазіргі уақытта Python қарапайым, жылдамдық пен қуаттың арқасында жасанды интеллектте қолданылатын негізгі тілдердің бірі болып табылады.

Django-әзірлеушілерге қуатты веб-қосымшаларды жылдам жасауға мүмкіндік беретін жоғары деңгейлі Python веб-құрылымы. Ол газет сайттарын құрумен айналысатын коммерциялық емес ұйымда 2003 жылы құрылды.

Django объектіге бағытталған бағдарламалауды қолдайды. Модель-көрініс-контроллер (MVC) архитектурасын қолданады. Ол дерекқорды басқару, автоматты әкімшілік интерфейстер, пішіндер, аутентификация, қауіпсіздік сияқты көптеген функционалдылықты қамтиды.

Django-ның басты ерекшеліктерінің бірі-оның даму жылдамдығы. Көптеген кіріктірілген функциялар мен кітапханалардың арқасында әзірлеушілер кодтың минималды мөлшерін қолдана отырып, қуатты қосымшаларды тез құра алады. Сонымен қатар, Django-да үлкен функционалдылықты қамтамасыз ететін қосымша модульдер мен плагиндер жасайтын кең әзірлеушілер қауымдастығы бар.

Django сонымен қатар хакерлер мен шабуылдардан қорғау үшін кіріктірілген мүмкіндіктерді қамтамасыз ету арқылы қолданбалардың қауіпсіздігін қамтамасыз етеді. Django SQL инъекциясы, XSS және CSRF сияқты шабуылдардан қорғаудың әртүрлі әдістерін енгізді.

Қуатты мүмкіндіктердің арқасында Django Instagram, Pinterest, сияқты көптеген танымал веб-сайттар мен қосымшаларда қолданылады. Бұл сонымен қатар әлемдегі ең танымал веб-құрылымдардың бірі.

Тұтастай алғанда, Django-бұл жоғары функционалдығы мен қауіпсіздігі бар веб-қосымшаларды құрудың қуатты құралы. Бұл қуатты веб-қосымшаны тез және оңай жасағысы келетін кез-келген әзірлеуші үшін тамаша таңдау.

**2.3. PostgreSQL**

PostgreSQL-Берклидегі Калифорния университетінде әзірленген дерекқорды басқарудың тегін Объектілік-реляциялық жүйесі (ДҚБЖ). Бұл үлкен көлемдегі деректерді тиімді сақтауға және өңдеуге мүмкіндік беретін қуатты және сенімді құрал.

PostgreSQL көптеген функционалдылықты қолдайды, соның ішінде SQL сұраулары, индекстер, транзакциялар, сақталған процедуралар, толық мәтінді іздеу, географиялық деректер, JSON құжаттары, көп ағынды және репликация. Бұл мүмкіндіктер әзірлеушілерге үлкен көлемдегі деректерді тиімді басқаруға және дерекқордың өнімділігін арттыруға мүмкіндік береді.

PostgreSQL ашық көзі болып табылады, бұл әзірлеушілерге өзгертулер енгізуге және өздерінің плагиндері мен кеңейтімдерін жасауға мүмкіндік береді. Сонымен қатар, жаңа модульдер жасайтын және мәліметтер базасының функционалдығын жақсартатын көптеген әзірлеушілер қауымдастығы бар.

PostgreSQL мүмкіндіктерінің бірі-деректердің тұтастығына кепілдік беретін және дерекқормен жұмыс істеудің сенімділігін қамтамасыз ететін транзакциялық модельді қолдау. Осыған байланысты, PostgreSQL банк жүйелерінде, бизнесті басқару жүйелерінде, онлайн-дүкендерде және т.б. сияқты деректердің жоғары сенімділігі мен дәлдігін қажет ететін қосымшаларда кеңінен қолданылады.

PostgreSQL сонымен қатар үлкен және күрделі қосымшаларда қолдануға мүмкіндік беретін жоғары өнімділік пен масштабтауға ие. Бұған сұраныстарды оңтайландыру, параллельді сұраныстарды өңдеу және көп ядролы өңдеу арқылы қол жеткізіледі.

Жалпы, PostgreSQL-бұл үлкен көлемдегі деректерді басқарудың қуатты құралы. Ол кең мүмкіндіктерге, жоғары сенімділікке және өнімділікке ие, бұл деректерді тиімді сақтау мен өңдеуді қажет ететін кез келген жоба үшін тамаша таңдау жасайды.

1. **Жобаны модельдеу бөлімі**

Дипломдық жобаны қорғау процесінің моделін жасау үшін UML (Unified Modeling Language) модельдеу құралдары пайдаланылды. Дипломдық жобаны қорғаудың негізгі кезеңдерін, қатысушылардың рөлін және олардың арасындағы өзара әрекетті қамтитын процестің графикалық моделі жасалды.

**3.1. Жүйе архитектурасы**

Дипломдық жобаны қорғау процесін автоматтандыру үшін клиент-сервер архитектурасын пайдаланылды.

Жүйе архитектурасы жүйенің бизнес-процестерін жүзеге асыруды қамтамасыз ету үшін бір-бірімен өзара әрекеттесетін бірнеше компоненттерді қамтиды.

**Изображение выглядит как диаграмма

Автоматически созданное описание**

**2.1-сурет ­­– Клиент-сервер архитектурасы**

Клиенттік қосымша пайдаланушы интерфейсін қамтамасыз етеді, ал серверлік бағдарлама дипломдық жобаларды жүктеу және де комиссияларды тағайындау сияқты жүйенің негізгі функцияларын орындайды. Сондай-ақ, жұмыс икемділігі мен өнімділігін арттыру мақсатында деректерді енгізу, өңдеу және шығару үш қабатқа бөлінетін үш деңгейлі архитектура қолданылды.

Клиент қолданбасы HTTP протоколын пайдаланып интернет арқылы сервер қолданбасымен байланысады. Сервер қолданбасы клиент қолданбасынан сұрауларды өңдейді, қажетті әрекеттерді орындайды және нәтижені клиентке қайтарады. Жүйенің барлық бизнес логикасы сервер жағында орналасқан, бұл қауіпсіздік пен деректерді қорғаудың жоғары деңгейін қамтамасыз етеді.

**2.2. Деректер қорының моделін жобалау**

ER-диаграммасы (Entity-Relationship diagram) — нысандарды, олардың атрибуттарын және олардың арасындағы қатынастарды көрсететін құрылымдық диаграмма. Ол мәліметтер қорын жобалау үшін қолданылады және жүйедегі деректер құрылымын логикалық түрде сипаттауға мүмкіндік береді.

Изображение выглядит как диаграмма

Автоматически созданное описание

2.2 – сурет. Жобаның ER диаграммасы

Диаграммада дипломдық жобаны қорғау процесіне қатысатын негізгі субъектілер көрсетілген:

* User – жүйеде тіркелген барлық қолданушылар туралы ақпаратты қамтитын нысан.
* Institute – тіркелген пайдаланушылардың жұмыс орындары жайында деректер сақтайтын нысан.
* Department – тіркелген қолданушылардың жұмыс орындарының бөлімдері жайында деректер сақтайтын нысан.
* Jobs - тіркелген пайдаланушылардың жұмыс түрлері жайында мәліметтерді сақтайтын нысан.

Субъектілер арасындағы қатынастар олардың арасындағы қатынасты көрсетеді. Мысалы, «User» және «Institute» субъектілерінің арасындағы бір-біріне қатынасы (one to one) әрбір хатыш немесе комиссия мүшесі тек бір ғана институтқа қатысты екендігін көрсетіп тұр.

**2.3. UML диаграммалары**

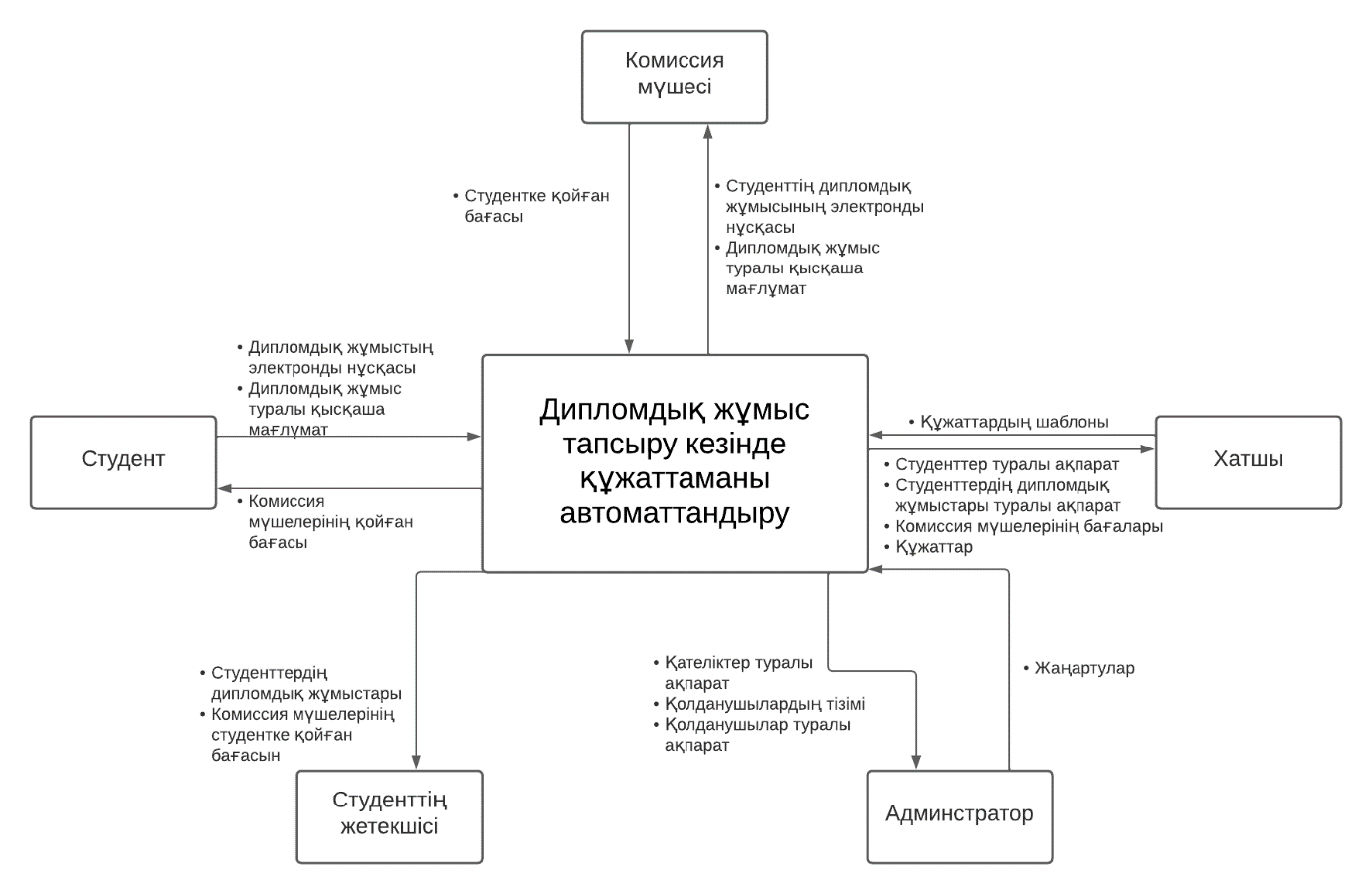
**2.3.1. Контексттік диаграмма**

Контексттік диаграмма жүйені және оның сыртқы субъектілермен әрекеттесуін көрсететін жоғары деңгейлі UML диаграммасы болып табылады.

Изображение выглядит как диаграмма

Автоматически созданное описание

2.3.1.1 – сурет. Құжаттаманы автоматтандыру контексттік диаграммасы



2.3.3.2 – сурет. Құжаттаманы автоматтандыру контексттік диаграммасының толық көрінісі

Диаграмма екі нысанды көрсетеді: «Актерлер» және «Автоматтандыру жүйесі». Актерлер – студенттер, комиссия мүшелері, хатшы, студенттің жетекшілері мен администраторлар сияқты жүйемен әрекеттесетін сыртқы пайдаланушылар. Автоматтандыру жүйесі - бұл актерлер енгізген мәліметтерді өңдейтін және оларға дипломдық жобалар мен қорғау процесі туралы ақпарат беретін жүйе.

**2.3.2. Прецеденттер диаграммасы**

Бұл прецеденттер диаграммасы жүйе пайдаланушыларының барлық мүмкін әрекеттерін және олардың жүйемен өзара әрекетін сипаттайды.

Изображение выглядит как диаграмма

Автоматически созданное описание

2.3.2 – сурет. Прецеденттер диаграммасы

Прецеденттердің сипаттамалық спецификациясы.

1-кесте.

|  |  |
| --- | --- |
| Бөлімдер | Сипаттамасы |
| Қысқаша сипаттамасы | ЖОО, колледж студенттері дипломдық жұмысты тапсыру кезінде, хатшы әрбір студент үшін белгілі құжаттарды толтыруы қажет. Бұл web-бағдарлама хатшылардың және комиссия мүшелерінің жұмысын автоматтандыру мақсатында жасалады. Яғни, құжаттарды толтыру және баға қою автоматты түрде жасалады. |
| Субъектілері | Студент, Студенттің жетекшісі, Комиссия мүшесі, Хатшы |
| Алғышарттар | Барлық қолданушылар тіркелген. Құжаттардың шаблондары бар. Қолданушылар системамен жұмыс жасап біледі. |
| Негізгі ағын | Қолданушы институт, кафедраны таңдай алады. Сол арқылы, қажетті студенттің таба алады. Студентті тапқаннан кейін парақшасына кіре алады. Сол жақтан дипломдық жұмысы туралы ақпаратты біле алады, ол туралы қысқаша мағлұмат оқи алады және диплодық жұмыстың толық электронды нұсқасын жүктей алады. Егер, қолданушы хатшы ретінде кірген болса, ол құжаттар бөліміне өте алады. Сол жердегі қажет құжаттың шаблонын таңдайды. Дайын формаларды таңдау арқылы, өзіне қажетті құжатты дайындап, өзіне жүктеп ала алады. |
| Альтернативті ағын | Қолданушы тіркелмеген. Бұл жағдайда тіркелу батырмасын басып, өзі туралы толық ақпаратты жазып системаға тіркелуі қажет.  Құжаттың шаблоны жоқ. Бұл жағдайда хатшы өзіне қажет құжатты еңгізуі қажет.  Дипломдық жұмыс табылмады. Студент жеке кабинетке кіріп. Дипломдық жұмысты енгізіп, толық ақпарат көрсетуі қажет. |
| Соңғышарттар | Хатшы құжаттарды алып, Студенттер өзінің бағасын көре алады. |

**2.3.3. Тізбек диаграммасы**

Тізбек диаграммасы жүйедегі объектілер арасындағы өзара әрекеттесу ретін және әрекеттердің орындалу ретін сипаттайды. Төменде жүйеде қолданушылардың іске асыра алатын бірнеше мысалын көрсек болады:

Изображение выглядит как диаграмма

Автоматически созданное описание

2.3.3.1 – сурет. Қолданушының сайтқа кіру тізбек диаграммасы

Диаграммада пайдаланушының жүйеге кіру ретін көрсетеді.

Ең алдымен қолданушы жүйеге кіру терезесіне деректерін, яғни қолданушы атын (логин) және де құпия сөзді (пароль) еңгізіп, «Кіру» батырмасын басу арқылы деректер қорына (БД) жіберіледі. Деректер қорында еңгізілген деректерге сәйкес келетін пайдаланушының бар екендігі тексеріледі. Егер процесс сәтті өтетеін болса, онда қолданушы сайттың басты бетіне өте алады. Ал жүйені пайдаланушы деректерін қате еңгізген жағдайда, қолданушы атын (логин) немесе құпия сөзді (пароль) қате еңгізгендігі жайында хабарлама шығып, дұрыс деректерді еңгізуін сұрайды.

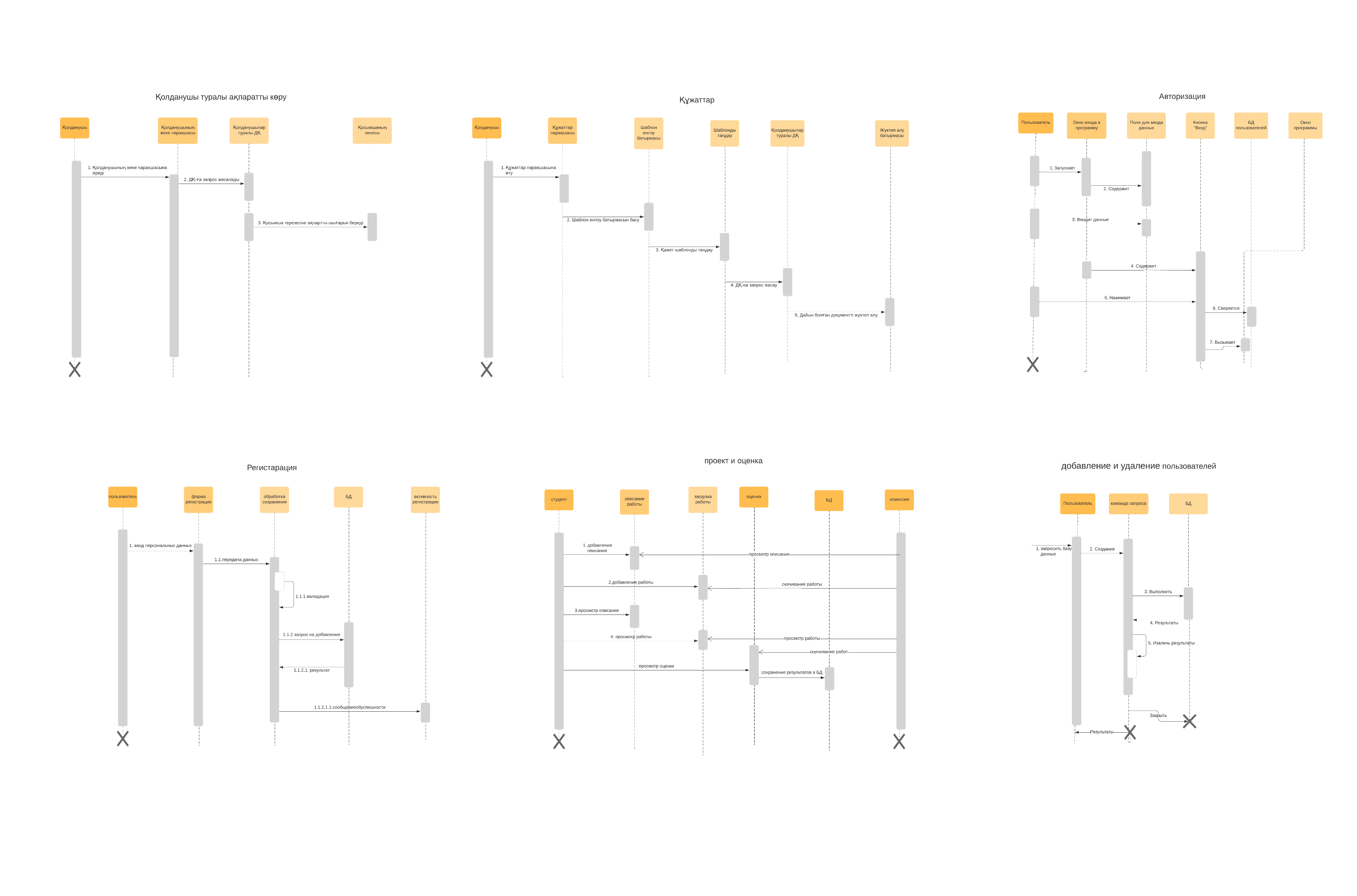
Изображение выглядит как диаграмма

Автоматически созданное описание

2.3.3.2 – сурет. Қолданушы жайында ақпаратты көру тізбек диаграммасы

Бұл диаграммада Комиссия мүшесінің Студенттер туралы ақпараттарды көру реті көрсетілген.

Комиссия мүшесі тізімде тұрған Студенттің аты-жөнінің үстін басу арқылы, Студенттің жеке парақшасына өте алады. Сол кезде деректер қорына запрос жіберіліп, өңделгеннен кейін қосымша терезесіне ақпараттарды шығарып береді.



2.3.3.3 – сурет. Құжаттарды жүктеу тізбек диаграммасы

Құжаттар тізбек диаграммасында Хатшының құжаттарды жүктеу реті көрсетілген.

Қолданушы құжаттар парақшасына өтеді. Құжаттың шаблонын еңгізу батырмасын басып, қажетті шаблонды таңдайды. Сол мезетте деректер қорына запрос жасалынады, дайын болған жағдайда құжаттар жүктеле бастайды.

**2.3.4. Күй диаграммасы**

Күй диаграммасы жүйенің барлық мүмкін күйлерін және олардың арасындағы ауысуларды көрсетеді.

Изображение выглядит как диаграмма

Автоматически созданное описание

2.3.4.1 – сурет. Комиссия мүшесінің күй диаграммасы

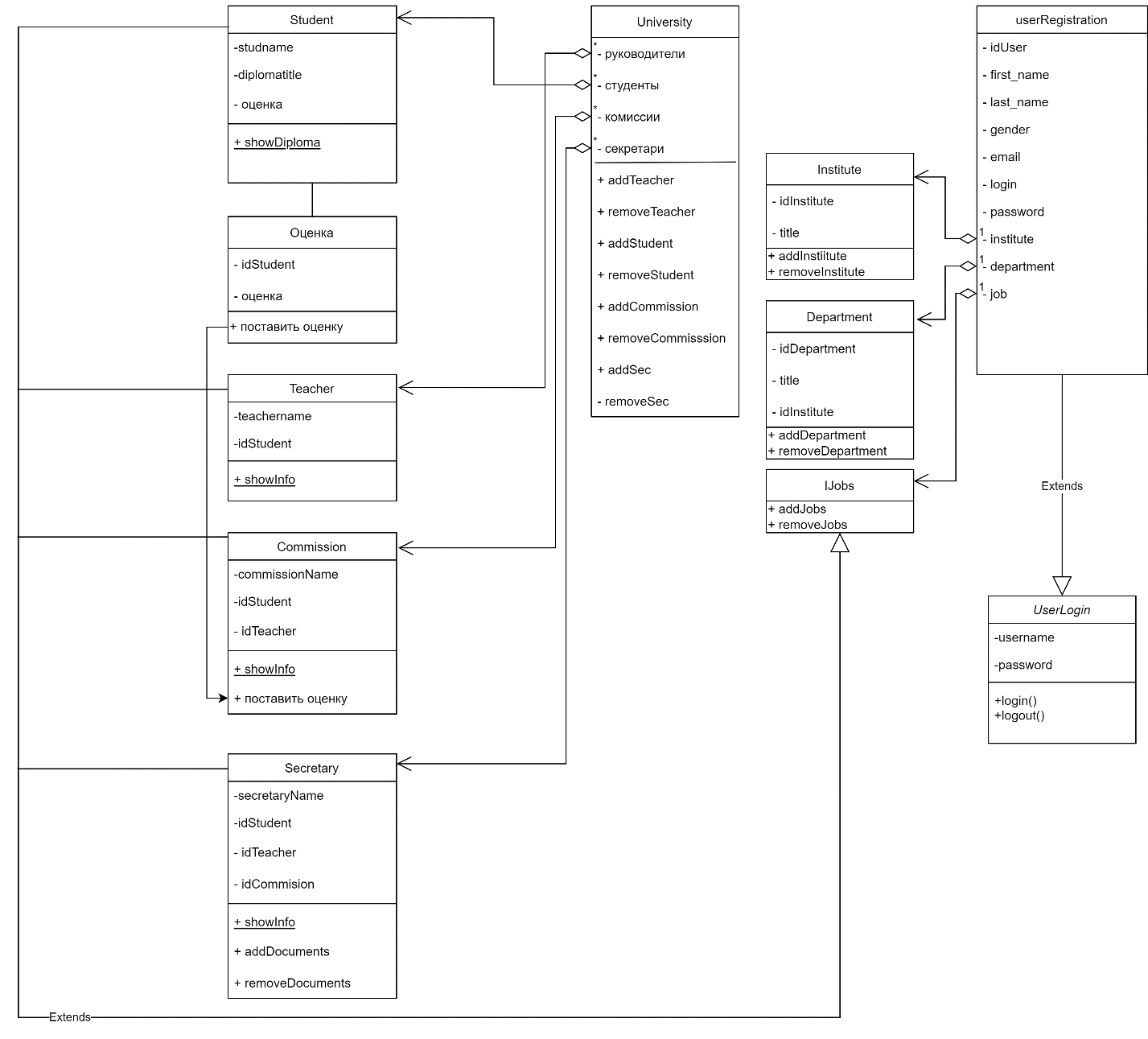
Изображение выглядит как диаграмма

Автоматически созданное описание

2.3.4.2 – сурет. Комиссия мүшесінің күй диаграммасы

**2.3.5. Класс диаграммасы**

Класс диаграммасы жүйенің кластарын, олардың атрибуттары мен әдістерін көрсетеді.



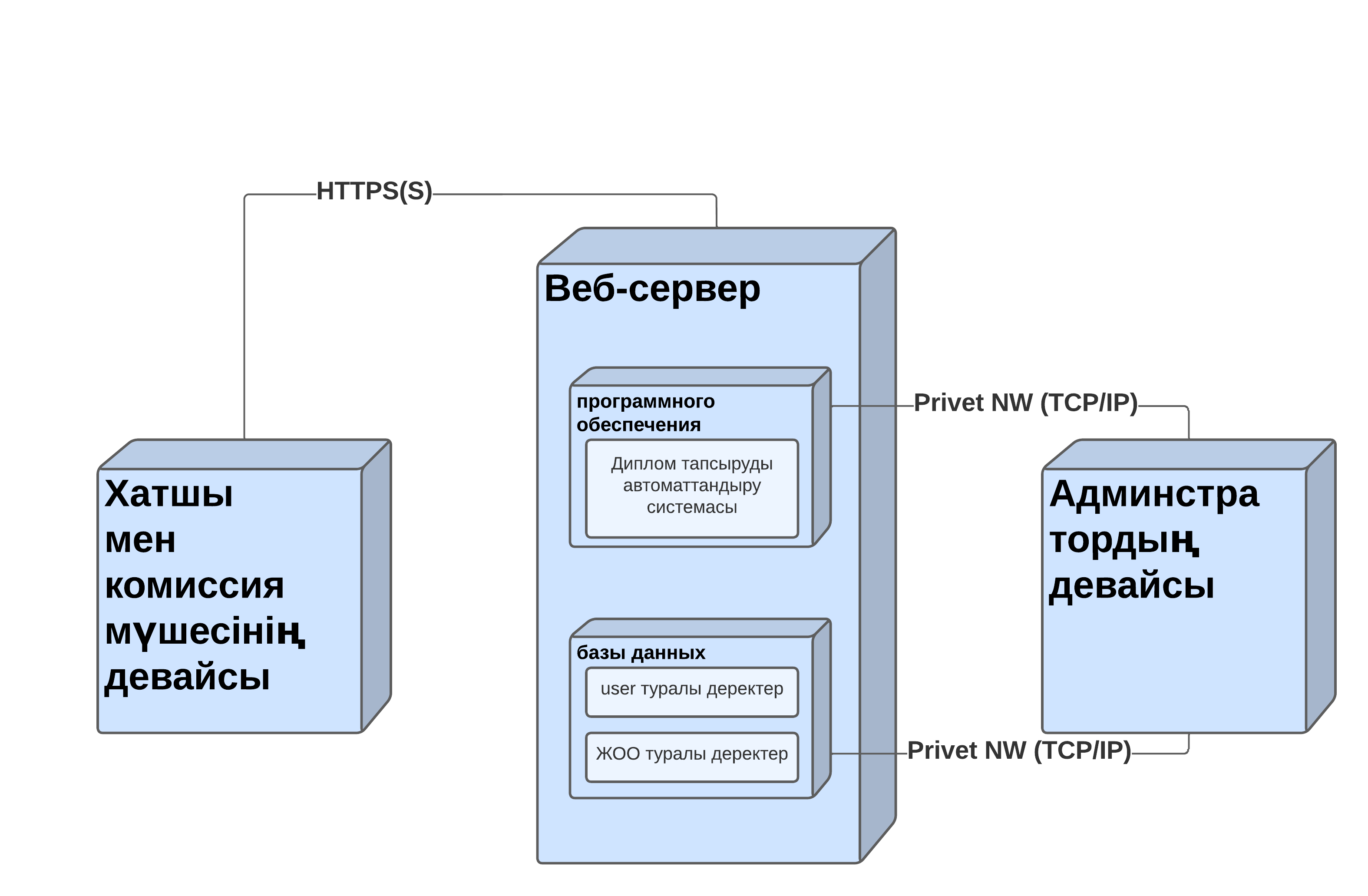
2.3.5 – сурет. Класс диаграммасы

Бұл класс диаграммасында барлық класстар көрсетілген:

* userRegistration – қолданушылардың сайтқа тіркелуге арналған формасы сақталатын класс;
* userLogin – тіркелген пайдаланушылар туралы деректерді сақтауды қамтамасыз етеді;
* Institute – институттар туралы деректер сақталады;
* Department – бөлімдер туралы деректер сақталады;
* Ijobs – қызмет түрлері сақталған;
* University – ең басты класс болып саналады. Бұл класста студент, хатшы, комиссия және де студенттің жетекшісі жайында ақпараттарды қамтамасыз етеді, сондай-ақ оларды қосуға немесе өшіруге мүмкіндік береді;
* Student – студенттің аты-жөні, дипломдық жұмысының тақырыбын және де бағасын сақтауға жауапты;
* Score (оценка) – бағаланған дипломдық жұмыстарды сақтайды;
* Teacher – студенттің жетекшісі жайында ақпараттары сақталады;
* Commission – комиссия туралы деректерді, жобаның күйін бағалау әдістері бар;
* Secretary – тағайындалған хатшы жайында ақпаттарды қамтамасыз етеді, сондай-ақ құжаттарды қосатын немесе өшіретін методтары көрсетілген.

**2.3.6. Ашып қарау диаграммасы**

Ашып қарау диаграммасы (Deployment diagram) - жүйедегі аппараттық құрамдас бөліктерді және олардың байланыстарын көрсететін құрылымдық диаграмма. Ол аппараттық құралдағы жүйе құрамдастарының физикалық орналасуын сипаттайды.

****

2.3.6 – сурет. Ашып қарау диаграммасы

Диаграмма жүйені іске қосу үшін қажетті аппараттық құрамдастарды көрсетеді. Орталықта дипломдық жобаны қорғау процесін автоматтандыру жүйесін орналастыратын сервер орналасқан. Оған студенттер, дипломдық жобалар, комиссиялар және олардың бағалаулары туралы мәліметтерді сақтайтын деректер базалары қосылған.

Жүйеге әртүрлі құрылғылардан кіруді қамтамасыз ету үшін клиенттік құрылғылар мен сервер арасында мәліметтерді тасымалдауды қамтамасыз ететін веб-сервер қолданылады. Клиенттік құрылғылар кез келген типті болуы мүмкін: дербес компьютерлер, ноутбуктер, смартфондар және планшеттер. Олар браузер немесе арнайы бағдарламалық қамтамасыз ету арқылы Интернет арқылы жүйеге қосыла алады. Қауіпсіз HTTPS протоколы клиенттер мен сервер арасындағы деректер қауіпсіздігін қамтамасыз ету үшін пайдаланылады.

**2.3.7. Компоненттер диаграммасы**

Компоненттер диаграммасы - жүйенің құрамдас бөліктерін және олардың өзара байланыстарын визуализациялауға мүмкіндік беретін UML диаграммасының түрі. Компоненттер диаграммасы жүйенің физикалық құрамдас бөліктерін көрсетеді.

Изображение выглядит как диаграмма

Автоматически созданное описание

2.3.7 – сурет. Компоненттер диаграммасы

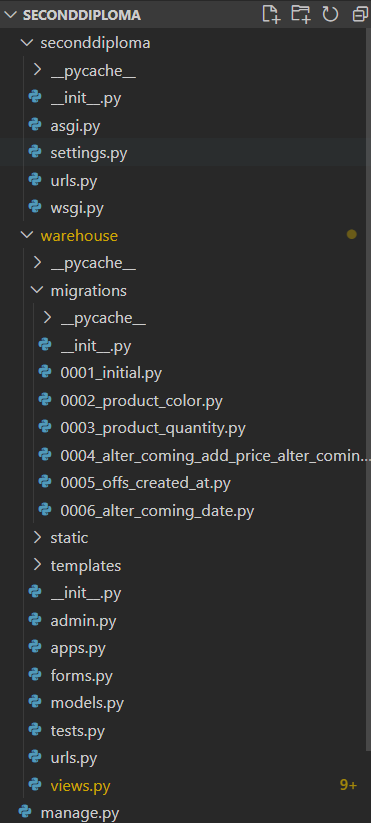
Бұл диаграмма жүйенің құрамдас бөліктерін және олардың байланыстарын көрсетеді. Дерекқорға қатысты құрамдас бөліктер сол жағында, ал пайдаланушы интерфейсіне қатысты құрамдас бөліктер оң жағында көрсетілген. Орталықта жүйенің іскерлік логикасына жауапты құрамдас бөліктер көрсетіледі.

Жүйенің барлық құрамдас бөліктері бір-бірімен өзара әрекеттеседі, ақпаратпен және сұраныстармен алмасады. Деректер базасы студенттер мен олардың дипломдық жобалары және де құжаттар туралы ақпаратты сақтау үшін пайдаланылады. Сервер компоненті жүйенің клиенттік бөлігі мен деректер қоры арасындағы өзара әрекетті қамтамасыз етеді. Клиент бөлігі жүйемен жұмыс істеу үшін пайдаланушы интерфейсін қамтамасыз етеді.

1. **Қолданбалы бөлімі**

**4.1 Web-қосымшаны әзірлеу**

Веб-қосымшаны әзірлеу кезінде веб-қосымшаға қажетті функцияларды орындайтын көптеген сыныптар іске асырылды.

****

**4.1-сурет – Жоба ағашы**

**4.2 Дайындалған веб-қосымшаның класстары**

**4.2.1 manage.py**

Бұл біздің ең негізгі сынып болып табылады. Осы сынып арқылы қандай әкімшілік тапсырмаларды орындауға болатынын көрсетуге болады. Бұл құжатта қосымша жасай алатын барлық нәрсе сипатталған.

**4.2.2 urls.py**

Біздің проектте urls деп аталатын екі құжат бар. Бұл жерде қосымшаға кіргенде не болатыны, қандай параққа баратыны белгіленеді. Ол екеуі өз-ара байланысқан. Бірінші құжатта екінші urls-ке сілтеме берілген және админ деп аталатын құжатқа сілтеме жасап тұр.

**4.2-сурет – Urls.py файлы**

Екінші құжатта толықтай веб-сайттын маршруттары белгіленген.

**4.2.3 settings.py**

Бұл файлда мәліметтер базасы, статикалық файлдар, қосымша қосымшалар, уақыт пен тіл параметрлері және басқа опциялар сияқты жоба параметрлері бар. Мен проектімде PostgreSQL қолдандым. Сол деректер қоры мен менің жобамды қосу осы settings.py-да жазылған. Суретте көріп тұрғандарыңыздай деректер қорының атын беріп, логин мен құпиясөзді жазу арқылы қосуға болады. Менің деректер қорымының аты ‘diplomaproject’.

**4.3-сурет – Деректер қорымен байланыс жасау**

Деректер қоры менде PostgreeSQL-де жасалған. Суретте кестелерімді көруге болады.

**4.4-сурет – Деректер қорындағы кестелер тізімі**

**4.2.4 models.py**

models.py файлында қосымшаның деректерінің модельдері сақталатын файл болып табылады. Моделдер-бұл деректер қорында сақталатын деректердің құрылымын анықтауға арналған сыныптардың бірі. Әр модель базадағы кестені көрсетеді, сол себептен класстың өрістері осы кестенің бағандарымен сәйкес келеді. Әрбір өрістің өзінің деректерінің типы бар, сол типы арқылы қандай деректер деректер қорында сақталатынын анықтауға болады. Мысалы, CharField болса онда тексттерді сақтауға болады, ал FileField болса онда файлдарды сақтауға болады. Менде модель бар, яғни 11 сынып бар және деректер қорымда 11 кесте бар. Ол модельдер: Secretary, Manager, Recipient, Provider, Group, Product, Expenditure, Expenditure\_add, Coming, Coming\_add, Offs. Secretary модельі, яғни Secretary деректер қоры, хатшылар туралы толық ақпаратты сақтап отырады. Ол жерде жеті өріс, яғни баған бар. Толықтай хатшылар туралы информация толтырылады. Менің екінші модельім ол Commissions модельі. Бұл жерде комиссия мүшелері туралы толық ақпарат көрсетіледі. Grade модельі бізде Commissions және Students модельімен байланысты. Модель бізде комиссия мүшелері қойған бағаларды сақтауға арналған. Commissions және Students модельдерінде айди арқылы студентті және комиссия мүшесімен сыртқы кілт арқылы байланысқан. Secretary және Chairmans модельдерінде хатшы мен төраға туралы толық ақпараттар толтырылған. Defense модельінде студенттің дипломдық жобаны қорғау процессі сағат нешеде басталып, қай уақытта аяқталғаны сақталады.

4.2.5 admin.py

Бұл файл жобада админдік панельдің баптаулары мен параметрлерін анықтау үшін арналған. Админдік панель бізде веб-сайтты тиімді басқару үшін жасалады. admin.py файлында Джангоның модельдер мен админдік панельді байланыстыруға болатын мүмкіндігін пайдалануға болады. Админдік панельді модельдермен байланыстырғаннан кейін, деректер қорымен жылдам әрі ыңғайлы басқаруға болады. Сонымен қатар, осы модельдердің қалай көрсетілетіні және негізгі жүйелердің (CRUD) үлгілерін қалай өңдеуге болатын кодтар орналасқан. Админ панелінде көрсетілетін бағандардың түрлері, бағандарды көрстеу әдістері және т.б. параметрлерді орналастыруға мүмкіндіктер бар.

**4.2.6 views.py**

views.py файлы менің проектімнің бэкенд бөлігінің ең басты бөлігі болып табылады. Ол жерде қолданушылардан келген HTTP-сұраныстардың өңдеу логикасы жазылған. Файл views.py пайдаланушылардың сұрауларын өңдейтін мүмкіндіктердің анықтамасын қамтиды. Бұл функциялар пайдаланушыдан деректерді ала алады, оларды өңдей алады және сұрауларға жауаптарды қайтара алады. Сондай-ақ файлда views.py сұрауларды тиімдірек өңдеуге және дерекқор үлгілеріне қатысты деректерді басқаруға мүмкіндік беретін Django базалық сыныптарынан мұраланған сыныптарды анықтауға болады. Views.py сондай-ақ, сұраныстарды өңдеуге қажетті басқа модульдердің импорты және сұраныстарды өңдейтін негізгі функциялардың ішінде қолданылатын көмекші функциялар болуы мүмкін. Файл views.py бұл Django жобасындағы маңызды компоненттердің бірі, өйткені ол веб-интерфейс пен қосымшаның логикасы арасындағы байланысқа жауап береді, бұл пайдаланушыларға сайтпен өзара әрекеттесуге және олардың сұрауларына жауап алуға мүмкіндік береді. Менің жобамның views.py файлында жеке-жеке жазылға 19 функция бар.

`Login\_page` функциясы кіру бетіндегі сұрауларды өңдеуде маңызды рөл атқарады. Ол пайдаланушылардың тіркелгі деректерін тексеруге және олардың рөліне байланысты тиісті беттерге автоматты түрде қайта бағыттауға жауап береді: хатшы үшін - басты бетке, ал менеджер үшін - басқару бетіне. Бұл Пайдаланушының рөліне байланысты ыңғайлы навигацияны және қолданбаның функционалдығына жылдам қол жеткізуді қамтамасыз етеді.

Тағы бір маңызды функция-менеджер үшін негізгі интерфейсті көрсетуге жауап беретін manager\_main. Мұнда тауарларды олардың атауы бойынша сүзу мүмкіндігі іске асырылды, бұл ыңғайлы навигацияны және қажетті ақпаратты іздеуді қамтамасыз етеді. Сондай-ақ, қосымшада жеке тауарларды және олардың сипаттамаларын (manager\_product\_page және product\_page) көруге арналған беттер бар.

Жобаның негізгі мақсаты дерекқормен тиімді өзара іс-қимылды қамтамасыз ету болып табылады. Ол үшін Django ORM (object-Relational Mapping) қолданылады, бұл базада сақталған деректермен жұмыс істеуді жеңілдетеді. Тағы бір маңызды мүмкіндік - өнімдер, олардың атауы бойынша сүзу мүмкіндігі бар барлық өнімдердің тізімін көрсетеді.

Product\_ids сияқты жаһандық айнымалыларды пайдалану әртүрлі сұраулар арасында деректерді тиімді өңдеуге және сақтауға көмектеседі.

Менің кодымдағы барлық мүмкіндіктер пайдаланушылардың жылдам және ыңғайлы тәжірибесін қамтамасыз етуге бағытталған жалпы қолданба архитектурасының маңызды бөлігін құрайды. Бұл интерфейстің жоғары жауаптылығын және сұраныстарды тиімді өңдеуді қамтамасыз етеді, бұл өз кезегінде қосымшаның жоғары өнімділігін қамтамасыз етеді.

'Providers\_list' функциясы барлық жеткізушілердің тізімін көрсетуге жауап береді. Ол дерекқордан барлық 'Provider' нысандарын алады және жекелендірілген көріністі қамтамасыз ету үшін оларды пайдаланушы атымен бірге шаблон контекстіне жібереді. Осыдан кейін функция ' provider\_list үлгісін көрсетеді.жеткізушілер тізімін көрсететін html`.

`Provider\_page ' функциясы белгілі бір жеткізушінің бетін көрсетуге арналған. Ол дерекқордан берілген идентификаторы бар "провайдер" нысанын және осы провайдерге қатысты барлық байланысты "Coming\_add" нысандарын күні бойынша сұрыптау арқылы алады. Содан кейін бұл деректер шаблон контекстіне жіберіледі және `provider\_page ' шаблоны арқылы көрсетіледі.HTML', ол жеткізуші туралы ақпаратты және одан тауарлар түсімдерінің тізімін көрсетеді.

'Recipient\_list' функциясы барлық алушылардың тізімін көрсетеді. Ол дерекқордан барлық 'Recipient' нысандарын шығарады және оларды жеке көрсету үшін пайдаланушы атымен бірге шаблон контекстіне жібереді. Содан кейін бұл деректер `recipient\_list үлгісін көрсету үшін пайдаланылады.алушылардың тізімін көрсететін html`.

'Recipient\_page' функциясы белгілі бір алушының бетін көрсетеді. Ол дерекқордан берілген идентификаторы бар `Recipient` нысанын және осы алушыға қатысты барлық байланысты `Expenditure\_add` нысандарын күні бойынша сұрыптау арқылы алады. Содан кейін бұл деректер шаблон контекстіне жіберіледі және `recipient\_page ' шаблоны арқылы көрсетіледі.алушы туралы ақпаратты және оның шығындар тізімін көрсететін html`.

"Инвентаризация" функциясы қоймадағы тауарларды түгендеуді басқаруға жауап береді. Пайдаланушы нысанды тауарды есептен шығару туралы мәліметтермен толтыра алады: тауарды таңдау, есептен шығару саны мен себебін көрсету. Нысанды жібергеннен кейін жүйе қоймада жеткілікті тауардың болуын тексереді. Егер өнім жеткілікті болса, есептен шығару орын алады және бұл туралы мәліметтер базасында жазба жасалады. Әйтпесе, пайдаланушы қате туралы хабарды көрсетеді. Сондай-ақ, Пайдаланушының есептен шығару жазбаларын күні бойынша сүзу мүмкіндігі бар.

'Shipment' функциясы тауарлардың жөнелтілуін басқаруға жауап береді. Пайдаланушы тауарды жөнелтуге есептен шығару туралы деректермен нысанды толтыра алады: алушыны таңдап, тауар мен санды көрсете алады. Нысанды жібергеннен кейін деректер дерекқорда сақталады және Саны қоймадан алынады. Сондай-ақ, пайдаланушыға сәтті жіберу туралы хабарлама көрсетіледі.

'Coming' функциясы тауарлардың қоймаға түсуін басқаруға жауап береді. Пайдаланушы нысанды тауардың түсуі туралы мәліметтермен толтыра алады: өнімді таңдаңыз, саны мен күнін көрсетіңіз. Нысанды жібергеннен кейін деректер дерекқорда сақталады және қоймада тауар саны артады.

`Download\_coming\_report` функциясы хатшының жұмысын автоматтандыруда маңызды рөл атқарады. Ол мәліметтер базасынан алынған мәліметтер негізінде тауарлардың келуі туралы есеп жасайды.

Алдымен функция келу туралы есеп беру үшін қолданылатын құжат үлгісін жүктейді. Содан кейін ол "coming\_ids" айнымалысында сақталған кіріс идентификаторларын пайдаланып дерекқордан кіріс туралы ақпаратты алады. Әрбір келу үшін функция тиісті деректерді шығарады және әр тауар үшін соманы есептейді.

Осыдан кейін функция хатшы, ағымдағы күн, тауарлардың келуі және сомалары туралы ақпаратты қамтитын контекст жасайды. Содан кейін ол құжат үлгісін `render` әдісі арқылы осы деректермен толтырады және нәтижені форматта сақтайды .DOCX "report\_coming.docx".

Сонымен қатар, функция пайдаланушыға жүктеу үшін жіберілетін жасалған құжатты қамтитын HTTP жауабын жасайды. Құжат `httpresponse`нысанын пайдаланып тіркеме ретінде жіберіледі.

Осылайша, `download\_coming\_report` функциясы хатшының жұмысын жеңілдету және қажетті ақпаратқа жылдам қол жеткізуді қамтамасыз ету арқылы тауарлардың келуі туралы есеп беру процесін автоматтандырады.

`Download\_shipment\_report` функциясы тауарларды есептен шығару туралы есеп жасауға жауап береді. Ол құжат үлгісін толтыру және жіберу тізімдері туралы есеп жасау үшін дерекқордағы деректерді пайдаланады.

Алдымен функция элементтерді есептен шығару туралы есеп беру үшін пайдаланылатын құжат үлгісін жүктейді. Содан кейін ол "shipment\_ids" айнымалысында сақталған есептен шығару идентификаторларын пайдаланып дерекқордан есептен шығару туралы ақпаратты алады. Әрбір есептен шығару жазбасы үшін функция тиісті деректерді шығарады және әр зат үшін соманы есептейді.

Содан кейін функция хатшы, ағымдағы күн, жіберу тізімдері және сомалар туралы ақпаратты қамтитын контекст жасайды. Осы контекстті қолдана отырып, функция құжат шаблонын `render` әдісімен толтырады және нәтижені форматта сақтайды .DOCX "report\_shipment.docx".

Осыдан кейін функция пайдаланушыға жүктеу үшін жіберілетін жасалған құжатты қамтитын HTTP жауабын жасайды. Құжат `httpresponse`нысанын пайдаланып тіркеме ретінде жіберіледі.

Осылайша, `download\_shipment\_report` функциясы тауарларды есептен шығару туралы есепті құру процесін автоматтандыруды қамтамасыз етеді, бұл хатшының жұмысын жеңілдетеді және қажетті ақпаратқа жылдам қол жеткізуге мүмкіндік береді.

'Download\_inventory\_report' функциясы түгендеу күйі туралы есеп жасауға жауап береді. Ол құжат үлгісін толтыру және қорлардың ағымдағы күйі туралы есеп жасау үшін дерекқордағы деректерді пайдаланады.

Алдымен функция түгендеу күйі туралы есеп беру үшін пайдаланылатын құжат үлгісін жүктейді. Содан кейін ол `inventory\_ids`айнымалысында сақталған түгендеу идентификаторларын пайдаланып дерекқордан түгендеу туралы ақпаратты алады. Әрбір түгендеу элементі үшін функция өнім идентификаторы, Өнім атауы, саны, есептен шығару себебі және жасалған күні сияқты тиісті деректерді шығарады.

Содан кейін функция хатшы, ағымдағы күн және түгендеу тізімі туралы ақпаратты қамтитын контекст жасайды. Осы контекстті қолдана отырып, функция құжат шаблонын `render` әдісімен толтырады және нәтижені форматта сақтайды .DOCX "report\_inventory.docx".

Осыдан кейін функция пайдаланушыға жүктеу үшін жіберілетін жасалған құжатты қамтитын HTTP жауабын жасайды. Құжат `httpresponse`нысанын пайдаланып тіркеме ретінде жіберіледі.

Осылайша, `download\_inventory\_report ' функциясы түгендеу күйі туралы есеп беру процесін автоматтандыруды қамтамасыз етеді, бұл хатшының жұмысын жеңілдетеді және қажетті ақпаратқа жылдам қол жеткізуге мүмкіндік береді.

`Download\_material\_report` функциясы материалдардың күйі туралы есеп жасауға арналған. Ол мәліметтер базасынан материалдар туралы ақпаратты алады және осы ақпаратты құжат шаблонын толтыру үшін пайдаланады.

Алдымен функция материалдар туралы есеп беру үшін құжат үлгісін жүктейді. Содан кейін ол "product\_ids" айнымалысында сақталған материал идентификаторларын пайдаланып дерекқордан әрбір материал туралы ақпаратты алады. Әрбір материал үшін функция айдың басындағы мөлшерді, келу санын, жөнелту санын, түгендеу санын және материалдардың жалпы санын есептейді.

Әрі қарай, функция хатшы, ағымдағы күн және тиісті деректері бар материалдар тізімі туралы ақпаратты қамтитын контекст жасайды. Содан кейін ол осы контекстті пайдаланып құжат үлгісін толтырады және нәтижені форматта сақтайды .DOCX "report\_material.docx".

Осыдан кейін функция пайдаланушыға жүктеу үшін жіберілетін жасалған құжатты қамтитын HTTP жауабын жасайды. Құжат `httpresponse`нысанын пайдаланып тіркеме ретінде жіберіледі.

Осылайша, `download\_material\_report` функциясы материалдардың күйі туралы есеп беру процесін автоматтандыруды қамтамасыз етеді, бұл хатшының жұмысын жеңілдетеді және қажетті ақпаратқа жылдам қол жеткізуге мүмкіндік береді.

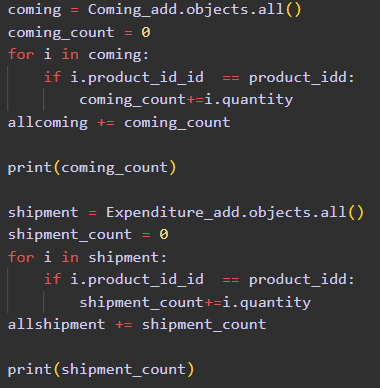
Менің дипломдық жобамның басты мақсаты хатшының жұмысын автоматтандыру осы жүктеп алуға арналған 4 функцияда көрсетілген. Яғни, деректер қорындағы ақпартты пайдалану арқылы документтерді автоматты толтыру процессі. Құжаттарды автоматты түрде толтыру үшін ең алдымен біз деректер қорымыздан керек ақпараттарды шығарып аламыз. Яғни, байланыс орнатамыз. Біздің жағдайда 4 кестеден деректі алу қажет, сол себептен төрт кестемен байланыстырамыз. Оның қалай байланысқанын суреттен көруге болады.



4.5-сурет – Деректер қорынан ақпарттарды алу процессі



4.6-сурет – Деректер қорынан ақпарттарды алу процессі



4.7-сурет – Деректер қорынан ақпарттарды алу процессі

Деректер қорымен байланысқаннан кейін, біздің соңында шығатын құжаттардың шаблонын жасап аламыз. Суретте біздің Барлық товарлар үшін материалдық отчет шаблонын көруге болады. Бұйра жақшалар ішінде айнымалыларды береміз, сол айнымалыларға біздің деректер қорымыздағы ақпараттар келіп түседі.



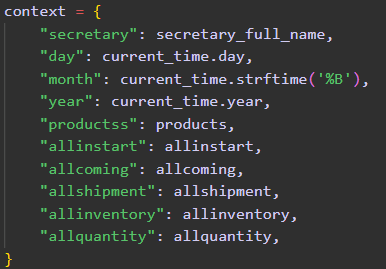
4.8-сурет – Протоколдың шаблоны

Шаблонды жобамыздың файлдарының құрамына қосып қоямыз. Ол шаблонға қалай қол жеткізгенімізді келесі кодтан көруге болады.



4.9-сурет – Шаблонға қосылу коды

Шаблонға қолымыз жеткеннен кейін шаблон ішіндегі айнымалылар мен деректер қорыннан алған ақпараттарды байланыстырып шығамыз. Мысал ретінде, хатшының аты-жөні, күні, товарлар мен барлық есептеулердің байланысқанын көруге болады.



4.10-сурет – Айнымалылар мен дерекетер қорындағы ақпартты байланыстыру коды

Деректер шаблонға беріліп болғаннан кейін, құжатты сақтап аламыз.

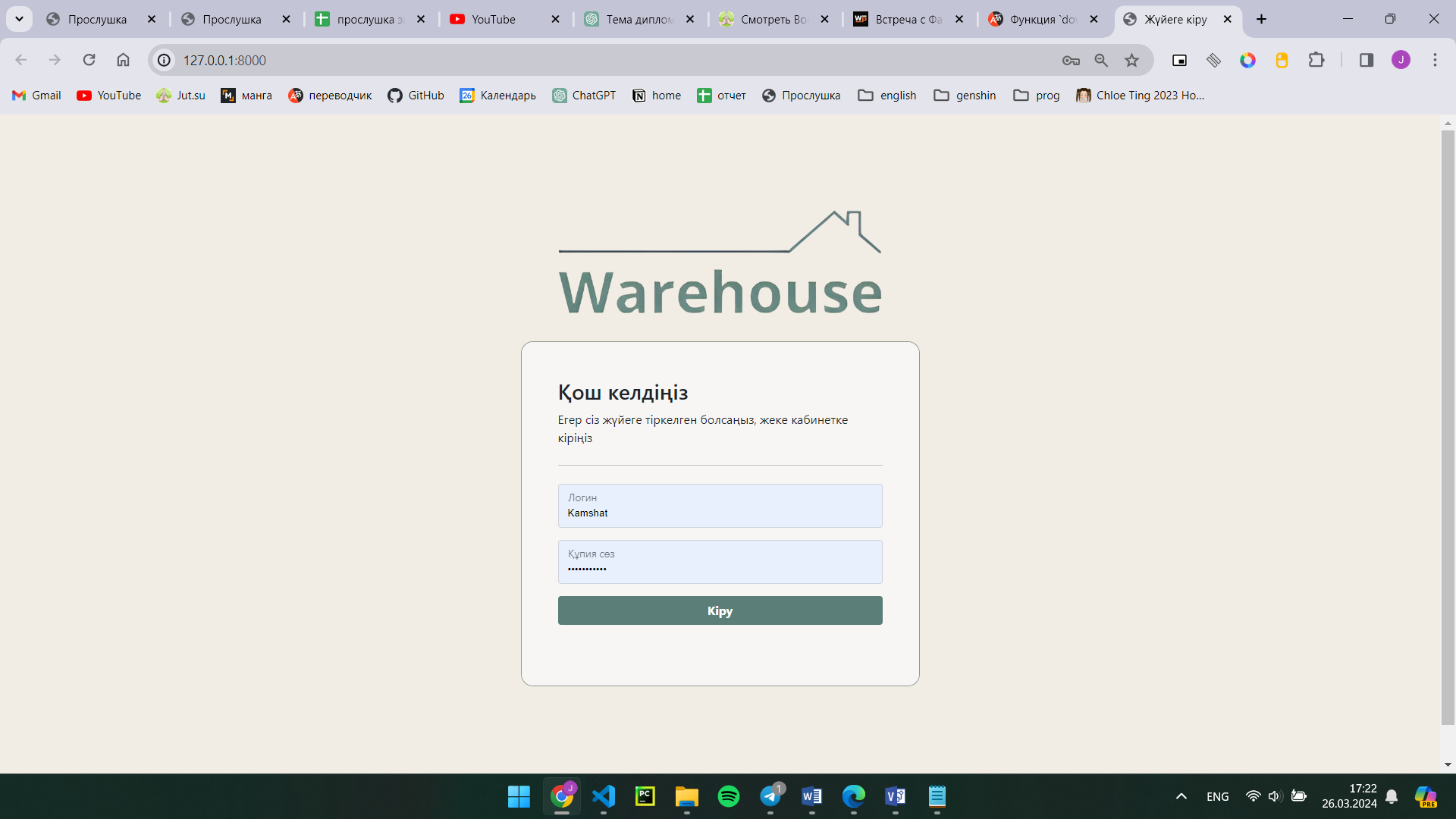


4.11-сурет – Құжатты сақтау процессі

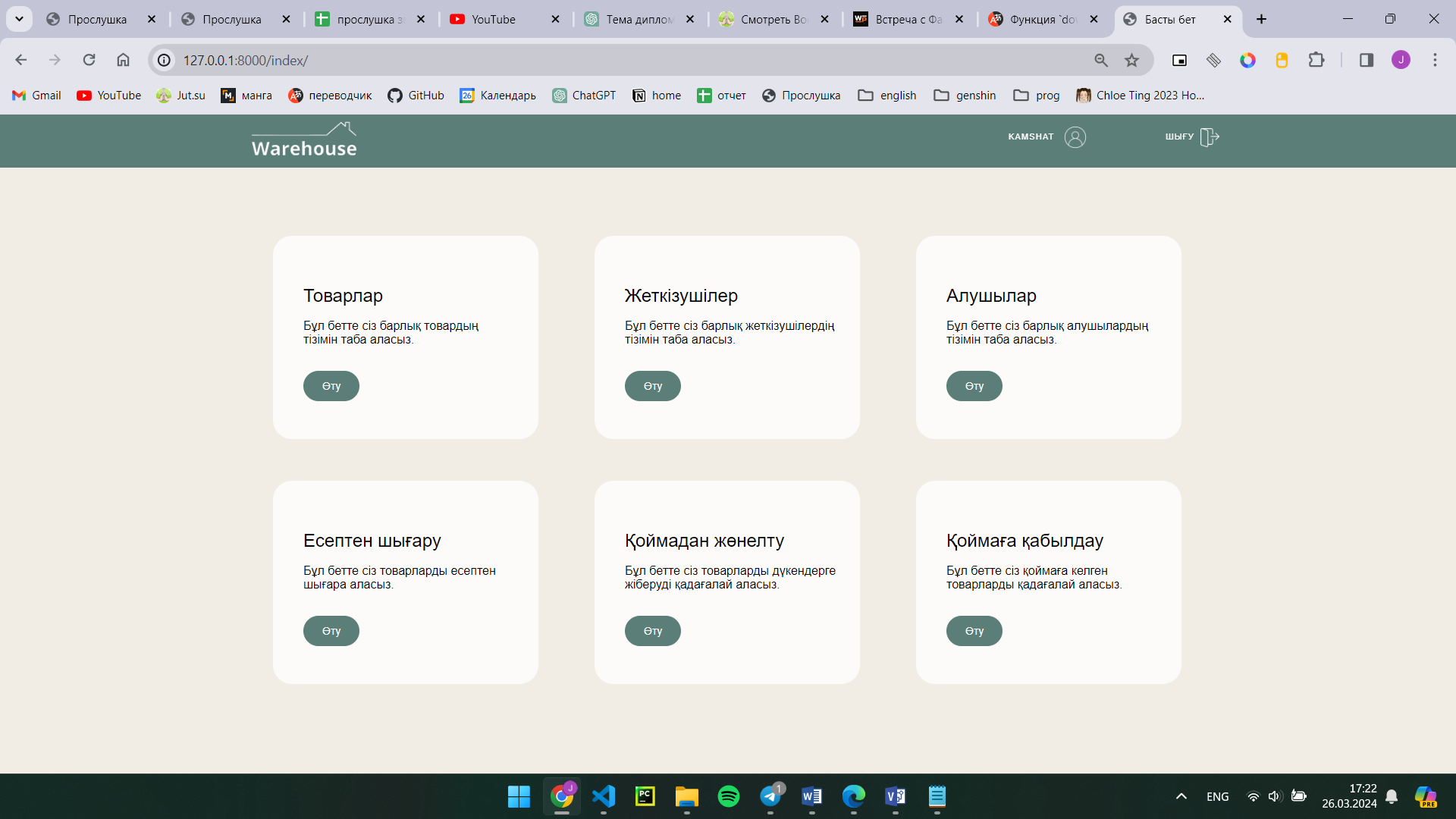
Құжат сақталғаннан кейін, осы құжат құрамында бар HTTP-жауап қайтарамыз. Бұл функция бізде бірнеше миллисекунд ішінде орындалады. Хатшы «жүктеп алу» батырмасын басқаннан кейін лезде құжат хатшының компьютеріне жүктеледі.

**4.3 Web-қосымшаның интерфейсі**

Веб-қосымшаны ашқан сәтте кіру беті пайда болады. Ол жерде хатшы немесе менеджер өзінің логин мен құпиясөзін жазуы қажет. Ең алдымен хатшының интерфейсін көрейік.

4.12-сурет – Кіру беті

Логин мен құпиясөзді жазып кіру батырмасын басқаннан кейін басты бетке түсеміз. Басты бетте товарлар, жеткізуші, алушылар, есептен шығару, өоймадан жөнелту, өоймаға қабылдау беттеріне өтуге болатын батырмалар орналасқан.



4.13-сурет – Хатшының басты беті

**ҚОРЫТЫНДЫ**

Жұмыстың басты мақсаты – хатшының жұмысын жеңілдету мақсатында құжаттарды толтыру процессін автоматтандыру үшін университет порталының веб-қосымшасын құрастыру болды.

Қойылған міндеттерге сәйкес:

Бірінші, ұқсас модельдерді салыстырып өзімнің жобама келетін тиімді модель таңдалды.

Екінші, таңдалған модель бойынша сызбалар мен диаграммалар толықтай құрылды.

Үшінші, анықталған диаграммалар бойынша Frontend, Backend, студент бетін, құжаттарды электронды толтыратын формалар толық дайындалып, қорытынды веб-парақша жасалынды.

Жасылынған веб-қосымша 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 студенттер, комиссия мүшелері туралы толық ақпаратты көруге, оқуға болады.

Бұл веб-қосымша хатшының жұмысын азайтып, тиімділігін арттырып, университеттерде дипломдық жобаны қорғау процессіне өте үлкен көмегін береді деген ойдамын.

**ҚОЛДАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ**

1 SQL және дерекқор сабақтары / / сайттағы электрондық нұсқа http://www.site-do.ru/db/db4.php

2 Деректер базасымен жұмыс / / сайттағы электрондық нұсқа http://www.intuit.ru/studies/courses/3439/681/lecture/14017

3 SQL тілінің түсінігі және оның негізгі бөліктері / / сайттағы электрондық нұсқа http://www.intuit.ru/studies/higher\_education/3407/courses/364/lecture/8659?page=2

4 MySQL ДҚБЖ-ге кіріспе / / сайттағы электрондық нұсқа http://www.intuit.ru/studies/courses/111/111/lecture/3246?page=3

5 MySQL Workbench-пен жұмыс істеу негіздері: жылдам бастау, деректер схемасын басқару / / сайттағы электрондық нұсқа http://mithrandir.ru/professional/soft-and-hardware/mysql-workbench-basics.html

6 Learning Web Design, Fifth Edition; O’Reilly; Jennifer Niederst Robbins; 2018; 810 с.; ISBN: 978-1-491-96020-2

7 Хомоненко А.Д., Цыганков В. М., Мальцев М. Г., мәліметтер базасы: жоғары оқу орындарына арналған оқулық / ред. проф. А. Д. Хомоненко. — 6-шы басылым., доп. - СПб.: Тәж-ғасыр, 2009. - 736 б., ISBN 978-5-7931-0527-9

8 Гольцман В. MySQL 5.0. Программист кітапханасы: Питер; Санкт-Петербург; 2010, ISBN 978-5-49807-135-0

9 SQL сабақтары және мәліметтер базасы. // Сайттағы электрондық нұсқа <http://www.site-do.ru/db/db.php>

10 Ақпараттық жүйелер мен технологияларды жобалаудың әдістері мен құралдары: оқу құралы / С.Ю.Петрова. - Ставрополь: Логос, 2018. - 114б.

11 Анализ и проектирование на UML // Новиков Ф.А-Санкт-Петербург,2007.-6б

12 Язык UML.Руководство пользователя.//Гради Буч.Джеймс Рамбо.Ивар Якобсан.-Москва,2006.-19б

13 Django ORM Cookbook 中文版<https://django-orm-cookbook-zh-cn.readthedocs.io/zh_CN/latest/query.html>

14 Что находится между идеей и кодом? Обзор 14 диаграмм UML 2020год <https://habr.com/ru/post/508710/>

15 UML диаграммы //Сайттың электронды нұсқасы https://www.diagrameditor.com.

16 Эрик Мэтиз. Изучаем Python. Программирование игр, визуализация данных, веб-приложения.

17 Билл Любанович. Простой Python. Современный стиль программирования

18 Django. Разработка веб-приложений на Python — Джефф Форсье, Пол Биссекс, Уэсли Дж. Чан.

19 Bootstrap в примерах//Сильвио Морето-Москва,2017.-31б

20 Проектирование информационных систем/ Т.В.Гвоздева,Б.А.Баллод.-

М.:Феникс,2013.- 205б

21 Разработка и эксплуатация автоматизированных информационных

систем // Л.Г.Гагарина,Д.В.Киселев,Е.Л.Федотова;-М.: Форум:Инфра-М,2013

22 Ақпараттық жүйелер мен технологияларды жобалаудың әдістері мен

құралдары: оқу құралы / С.Ю.Петрова. - Ставрополь: Логос, 2018. - 114б.

23 Модель “Сущность-связь”//В.Ю.Кара-Ушанов-Екатинбург,2017.-61б

24 JavaScript және Html. Дэвид Флэнаган кітабы: http://kharchuk.ru/JavaScript.pdf. 252-285 бет

25 JavaScript , HTML, Css. Робин Никсон кітабы: <http://lib.maupfib.kg/wp-content/uploads/Robin-Nikson-Sozdaem-dinamicheskie-veb-sayty-spomoshhyu-PHP-MySQL-i-JavaScript-SPb-Piter-2011-356s.pdf>

**A қосымшасы**

(міндетті)

**Техникалық тапсырма**

Қоймалардағы есептеме процессін автоматтандыруға арналған веб-қосымша құруға арналған техникалық тапсырма

**А.1 Жалпы сипаттама**

Бұл техникалық тапсырма қоймалардағы есеп беру процесін жеңілдетуге арналған веб-қосымшаның дамуын сипаттайды. Қолданба қоймадағы тауарлық-материалдық құндылықтар туралы деректерді жинауды, өңдеуді және талдауды автоматтандырады және қойма қызметкерлері мен Басқарушы персонал үшін есептерге ыңғайлы қол жеткізуді қамтамасыз етеді.

**А.2 Мақсаты**

Жобаның мақсаты қоймалардағы есеп беру процесін жеңілдетуге және жеделдетуге, деректердің дәлдігін арттыруға және тауар қорларын талдау негізінде басқару шешімдерін жедел қабылдауды қамтамасыз етуге мүмкіндік беретін функционалды веб-қосымшаны құру болып табылады.

**А.3 Міндеттері**

- талдау және жобалау:

a) кіріс талаптарын жинау және өңдеу, аналитика;

b) сайттың негізгі беттерін жобалау.

дизайн тұжырымдамасын жасау:

a) негізгі және 2 ішкі беттер мысалында сайттың дизайн тұжырымдамасын жасау.

- сайттың барлық беттерінің дизайн-макеттерін жасау:

a) сайттың ішкі беттерінің дизайн-макеттерін әзірлеу.

- front-end:

a) макеттерді дайындау;

b) шаблондардың front-end бөлігін программалау;

c) адаптивті орналасу.

- back-end:

a) ДҚ құрастыру;

b) веб-бақдарламаның back-end бөлігін программалау.

- Сайтты хостингке көшіру, орнату, жариялау:

a) Серверді орнату, веб-ортаны орнату және конфигурациялау;

b) Жобаны ашық қолжетімділікте жариялау.

**А қосымшасының жалғасы**

**А.4 Өнімге шолу**

Қоймалардағы есеп беру процесін автоматтандыруға арналған Веб-қосымша Бизнестің тауарлы-материалдық құндылықтарды тиімді басқаруға деген заманауи қажеттіліктерін ескере отырып жасалған қуатты құрал болып табылады. Өнім тауарларды есепке алудың ашықтығы мен жеделдігін қамтамасыз етуге бағытталған, бұл қордың қозғалысын оңай және жылдам бақылауға мүмкіндік береді.

Негізгі мүмкіндіктерге мыналар кіреді:

* Пайдаланудың қарапайымдылығын және қызметкерлердің жүйені жылдам игеруін қамтамасыз ететін интуитивті интерфейс.
* Есептер мен аналитиканы орнатудағы икемділік, бұл жүйені әр қойманың нақты қажеттіліктері мен талаптарына бейімдеуге мүмкіндік береді.
* Басқарудың негізделген шешімдерін қабылдауға ықпал ететін тауарлық-материалдық құндылықтар туралы ақпаратқа жылдам қол жеткізу мүмкіндігі.
* Қойманың ағымдағы жағдайы туралы толық суретті қамтамасыз ететін және жасырын трендтер мен үлгілерді анықтауға көмектесетін қуатты аналитикалық құралдар.
* Қойманың қорлары мен операциялары туралы құпия ақпаратты рұқсатсыз кіруден қорғау үшін деректердің қауіпсіздігін қамтамасыз ету.

Біздің өнім қоймалардағы есеп пен есеп беру процесін жеңілдетіп қана қоймай, сонымен қатар қойма шаруашылығының жұмысын оңтайландыруға және кәсіпорынның бизнес-процестерінің тиімділігін арттыруға ықпал ететін стратегиялық шешімдер қабылдауда сенімді көмекшіге айналады.

**А.5 Терминдер, анықтамалар және қысқартулар**

Браузер-веб-сайттарды қарауға арналған бағдарламалық жасақтама. Техникалық тапсырма-осы құжат.

Веб-бағдарлама, веб-сайт-Интернет компьютерлік желісіндегі байланысқан электрондық құжаттардың (файлдардың) бір домендік атпен біріктірілген жиынтығы.

Статикалық бет - деректер мәтіні WYSIWYG арқылы жасалатын сайт беттерінің түрі-деректер базасынан бағдарламалық кодпен автоматты түрде жасалғаннан гөрі, мазмұнды басқару және басқару ішкі жүйесінің редакторы.

**А қосымшасының жалғасы**

Динамикалық бет (бөлім) – сайт беттерінің түрі. Осы бөлімнің барлық беттері дерекқордан автоматты түрде жасалады және бөлімнен нақты бетті өзгерту байланысты дерекқор кестесінен жазба мазмұнын өзгерту арқылы жасалады. Бөлім келесі санаттардан тұрады:

Тарату беті-егжей-тегжейлі беттерге сілтемелер тізімі бар бет (мысалы, мұрағаттық жаңалықтарға сілтемелер тізімінен тұратын "жаңалықтар мұрағаты").

Толық беттер-берілген кесте жазбасынан деректерді көрсететін беттер(мысалы, мұрағаттан таңдалған нақты жаңалықтар беті).

Alt-сурет атрибуты. Суреттерге балама мәтін орнатады. Мұндай мәтін браузерде суреттерді жүктеу кезінде сурет туралы мәтіндік ақпаратты алуға мүмкіндік береді.

**А.6 Толық талаптар**

**А.6.1 Жалпы қойылатын талаптар**

* Масштабтау: қолданба шағын қоймалар мен ірі логистикалық орталықтар үшін тиімді жұмысты қамтамасыз ету үшін масштабтау мүмкіндігін ескере отырып жасалуы керек.
* Ақауларға төзімділік: Веб-бағдарлама сенімді болуы керек және желідегі ақаулар немесе басқа техникалық ақаулар болған жағдайда да тұрақты жұмыс істеуі керек.
* Интеграция: қолданба деректермен алмасу және бизнес-процестерді оңтайландыру үшін басқа бухгалтерлік есеп және қоймаларды басқару жүйелерімен интеграциялануы керек.
* Қауіпсіздік: қойма деректері сезімтал ақпарат болып табылады, сондықтан қолданба шифрлау мен аутентификация механизмдерін қоса алғанда, деректерді қорғаудың жоғары деңгейін қамтамасыз етуі керек.
* Қолдау: Веб-бағдарлама пайдаланушыларға жылдам және тиімді қолдау көрсетуі керек, соның ішінде тұрақты жаңартулар мен қателерді түзету.
* Үйлесімділік: қолданба барлық пайдаланушыларға қол жетімділікті қамтамасыз ету үшін әртүрлі операциялық жүйелермен және веб-шолғыштармен үйлесімді болуы керек.
* Қолжетімділік: қолданба қойма шаруашылығының үздіксіздігін қамтамасыз ету үшін пайдаланушыларға тәуліктің кез келген уақытында үзіліссіз және кідіріссіз қолжетімділікті қамтамасыз етуі керек.

**А қосымшасының жалғасы**

**А.6.2 Front-end қойылатын талаптар**

Сайт бейімделген орналасуды жүзеге асыруы керек, атап айтқанда

1. Жауапты орналасу (ағылш. Responsive web design)-Интернетке қосылған түрлі құрылғыларда сайттың көрсетілуін қамтамасыз ететін және терезенің берілген өлшемдеріне динамикалық бейімделетін HTML-беттерді беттеу;

Сайт беттерін іске асыру үшін HTML 5 және CSS 3 тілдері қолданылуы керек. Бастапқы код W3C стандарттарына сәйкес жасалуы керек (http://www.w3.org/). клиенттің интерактивті элементтерін енгізу үшін JavaScript тілін қолдану керек; HTML-беттер, сондай-ақ оларға ілесіп жүретін мәтіндік файлдар (html, css, js және т.б.) utf-8 кодталуында көрсетілуі тиіс. Әр түрлі ОС - та (Linux, Windows, Mac OS, Android, iOS) әр жүйенің қаріптерін сызу ерекшелігіне байланысты сайттың бірдей қаріптерін көрсету әр түрлі болуы мүмкін;

Сайттың жауапты орналасуы келесі браузерлерде дұрыс көрсетілуін қолдауы керек:

a) Internet Explorer 10.0 және одан жоғары;

b) Opera Browser соңғы нұсқасы;

c) Mozilla Firefox соңғы нұсқасы;

d) Google Chrome соңғы нұсқасы;

**А.6.3 Бағдарламалық жасақтаманың жүйелік сипаттамалары**

* Жүйе жылына 24 сағаттан аспауы мүмкін. Деректерді резервтеуді хостинг провайдері жүзеге асырады. Сайт әкімшісінде сайттың көшірмесін жүктеу және жүктеу мүмкіндігі болуы керек;
* Сайттың жалпы бөлігінің бастапқы кодын қорғау қажет. Қол жетімділікті шектеу қажет. Пайдаланушы құпия сөздері шифрланған түрде сақталады;
* ДҚБЖ деңгейінде ДҚ-дағы деректерге қол жеткізуді саралау іске асырылуы тиіс;
* Көлемі 1 КБ-тан асатын барлық суреттер мен фотосуреттер (беттің дизайн элементтерінен басқа) ауыстырылатын мәтінмен жасалуы керек. Барлық сызбалар gif немесе jpg форматында болуы керек;
* Сайттың бір стандартты жүктелетін бетінің көлемі орта есеппен 170 КБ аспауы керек;
* Кез-келген сценарийдің жұмысы 60 секундтан аспауы керек. Серверге жүктеме тәулігіне портал беттеріне 500.000-нан аспайтын жағдайда.