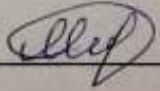




|                                              |                                  |
|----------------------------------------------|----------------------------------|
| Университет:                                 | Satbayev University              |
| Название:                                    | Антиплагиат Айдос влб 15 1k.docx |
| Автор:                                       | Рамазан Айдос                    |
| Координатор:                                 | Мақсат Қанатов                   |
| Дата отчета:                                 | 2019-05-08 10:29:38              |
| Козффициент подоби́я № 1: ?                  | 2,4%                             |
| Козффициент подоби́я № 2: ?                  | 0,0%                             |
| Длина фразы для козффициента подоби́я № 2: ? | 25                               |
| Количество слов:                             | 4 700                            |
| Число знаков:                                | 38 326                           |
| Адреса пропу́щенные при проверке:            |                                  |
| Количество завершенных проверок: ?           | 75                               |

Ғылыми жетекші  М. Қанат

**Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті**

**Мамандығы 5B040700 – Есептеу техникасы және бағдарламалық  
қамтамасыз ету.**

**Рамазан Айдос Берікұлы**

**Тақырыбы: «Пиццаға тапсырыс берудің кросс-платформалық қолданбасы»**

### **ҒЫЛЫМИ ЖЕТЕКШІНІҢ СЫН-ПІКІРІ**

Дипломдық жобада күнделікті кездесетін және адам өмірі-не тереңінен еніп жатқан компьютерлік техника, ақпараттық жүйе мен желінің дамуы ақпарат көлемінің жылдам өңделуіне жақсы әсер етуде.

Дипломдық жобаның түпкі мақсаты, кез-келген адамдарға пиццаға тапсырыс беру жолын жеңілдету және жылдамдату. Дипломдық проектінің өзектілігі пиццаға тапсырыс беру жолында өзіне қажеттісін мобильді қолданба арқылы жасауға болуы.

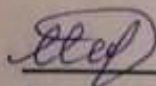
Дипломдық жұмыс кіріспеден, үш бөлімнен, қолданылған әдебиеттер тізімінен, қорытындыдан тұрады. Бірінші бөлімде мәселенің қазіргі замандағы жағдайына шолу және оны талдау жасалынды. Екінші бөлімде программалау тілдерін пайдалану ерекшеліктері жобаланған. Үшінші бөлімде клиника қызметкерлері үшін корпоративті порталды жобалау мен жасауды дайындалған.

Дипломдық жобаны қарастыра келгенде жақсы жақтарын айта кетуге болады. Жұмыс материалы жоспарлы түрде, қойылған талаптарға сай, тақырып төңірегінде жазылған. Студенттің жұмысты рәсімдеудегі біліктілігін, талдаушылық қабілетін көруге болады. Осы жұмыста негізгі өзекті мәселелерді шеше отырып, зерттеу жұмысында оңтайлы нәтижелер алып, жұмысты қойылған талаптарға сай орындап шықты.

Қортындылай келгенде, осы зерттеу жұмысын орындау барысында Рамазан Айдос Берікұлы айтарлықтай жақсы еңбектенгені көрінеді.

Жоба жетекшісі ретінде бұл дипломдық жобаны өз деңгейіне сәйкес деп есептей отырып Рамазан Айдос Берікұлының «Пиццаға тапсырыс берудің кросс-платформалық қолданбасы» атты тақырыптағы дипломдық жұмысын 90%(А) деп бағалай отырып, 5B040700 – «Есептеу техникасы және бағдарламалық қамтамасыз ету» мамандығы бойынша «Техника және технологиялар бакалавры» академиялық дәрежесін тағайындауға болады деп есептеймін.

**ҒЫЛЫМИ ЖЕТЕКШІ**



**М.Қанат**

**«Компьютерлік және программалық инженерия»**

**кафедрасының лекторы «30» 04 2019 жыл**

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Программалық инженерия кафедрасы

Рамазан Айдос Берікұлы

«Пиццаға тапсырыс берудің кросс-платформалық қолданбасы»

Дипломдық жобаға  
**ТҮСІНІКТЕМЕЛІК ЖАЗБА**

5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»  
мамандығы

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Программалық инженерия кафедрасы

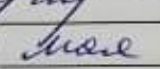
**ҚОРҒАУҒА ЖІБЕРІЛДІ**

Кафедра меңгерушісі

тех. ғыл. кандидаты, доцент,

ассистент-профессор

 Р. Юнусов

"13"  2019 ж.

Дипломдық жобаға

**ТҮСІНІКТЕМЕЛІК ЖАЗБА**

Тақырыбы: «Пиццаға тапсырыс берудің кросс-платформалық қолданбасы»

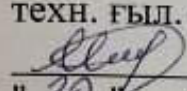
5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»

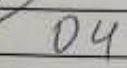
Орындаған

А. Б. Рамазан

Ғылыми жетекші

техн. ғыл. магистрі, лектор

 М. Қанат

"30"  2019 ж.

Алматы 2019



ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӨТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

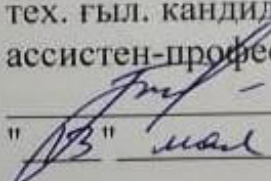
Программалық инженерия кафедрасы

5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»

**БЕКІТЕМІН**

Кафедра меңгерушісі

тех. ғыл. кандидаты, доцент,  
ассистент-профессор

 Р. Юнусов

"13" мамыр 2019 ж.

**Дипломдық жоба орындауға  
ТАПСЫРМА**

Білім алушы: Рамазан Айдос Берікұлы

Тақырыбы: «Пиццаға тапсырыс берудің кросс-платформалық қолданбасы»

Университет академиялық мәселелер жөніндегі проректорының бұйрығымен  
«14» наурыз 2018ж. № 1841-б шешімімен бекітілген.

Орындалған жобаның өткізу мерзімі «13» мамыр 2019 ж.

Дипломдық жұмыстың бастапқы мәліметтері: Ұсынылатын дипломдық жобаны құру барысында пиццаға тапсырыс берудің кросс-платформалық қолданбасын жасау.

Есеп – түсініктеме жазбаның талқылауға берілген сұрақтардың тізімі:

а) аналитикалық шолу;

б) жобалау бөлімі;

в) өңдеу құралдары және жобаны талдау;

г) қолданбалы бөлім;

д) А Қосымшасы – техникалық тапсырма.

е) Б Қосымшасы – бағдарлама мәтіні.

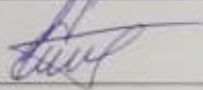
Жобаның презентациялық 14 слайды ұсынылған.

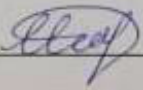
Ұсынылған негізгі әдебиеттер: 15 атау.

Дипломдық жобаны орындау  
КЕСТЕСІ

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

Дипломдық жұмыс бөлімдерінің кеңесшілерінің аяқталған жұмысқа қойған қолтаңбалары

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

Ғылыми жетекші  М. Қанат

Тапсырманы орындауға қабылдап алған студент  А. Б. Рамазан

Күні

«3» маусым 2018 ж.

## АҢДАТПА

Жобаның негізгі міндеті пиццаға тапсырыс беру жолын жеңілдету және қолжетімді жасау және сол пиццаға тапсырыс беру жолында өзіңе қажеттісін мобильді қолданба арқылы жасауға болатынында.

Менің жобамның қызметі, ол қолданбаға қажетті API құру, және сол құрылған API-мен деректер қорын тікелей байланыстыру еді. Байланысқа қауіпсіз байланысқа түскен соң, API-ға келген сұраныстарға JSON түрінде қайта өзіне қайтарады.

Жоба нәтижесінде қойылған мақсат, міндеттер орындалды. Android Studio арқылы жоспарланған мобилды қолданба жасалынды.

## АННОТАЦИЯ

Основная цель проекта – сделать заказ пиццы более простым и доступным, а также создать свою пиццу с помощью мобильного приложения.

Роль моего проекта состояла в том, чтобы создать API для приложения и связать его напрямую с базой данных и с этим же API. Как только соединение установлено безопасно, он возвращает запрос к API как JSON.

В результате реализации проекта цели и задачи были выполнены. Мобильное приложение было создано с помощью Android Studio.



## **ANNOTATION**

The main goal of the project is to make pizza ordering easier and more affordable, as well as create pizza using a mobile application.

The role of my project was to create an API for the application and link it directly to the database and the same API. Once the connection is established safely, it returns the request to the API as JSON.

As a result of the project, the goals and objectives were fulfilled. Mobile application was created using Android Studio.

## МАЗМҰНЫ

|       |                                         |    |
|-------|-----------------------------------------|----|
|       | Кіріспе                                 | 8  |
| 1     | Негізгі бөлім                           | 9  |
| 1.1   | Жобаның ерекшеліктері                   | 9  |
| 1.2   | Мақсаты                                 | 9  |
| 1.3   | Android жүйесінің негізгі мүмкіндіктері | 10 |
| 1.4   | Android қосымшасының жалпы схемасы      | 11 |
| 1.5   | Web – технологияның қазіргі жағдайы     | 13 |
| 2     | Жобалау бөлімі                          | 15 |
| 2.1   | Унифицирленген моделдеу тілі            | 15 |
| 2.2   | Microsoft Visio 2016 құралы             | 15 |
| 2.3   | Прецеденттер диаграммасы                | 16 |
| 2.4   | Тізбек диаграммасы                      | 17 |
| 2.5   | Кооперация диаграммасы                  | 19 |
| 2.6   | Күй диаграммасы                         | 20 |
| 2.7   | Класстар диаграммасы                    | 21 |
| 2.8   | ER диаграмманы құру                     | 22 |
| 3     | Қолданылған бағдарламалық қамтамалар    | 23 |
| 3.1   | MySQL технологиясы                      | 23 |
| 3.2   | Javascript программалау тілі            | 23 |
| 3.3   | HTML тілі туралы түсінік                | 25 |
| 3.4   | Apache2 веб сервері                     | 26 |
| 3.5   | CSS бағдарламалау тілі                  | 27 |
| 3.6   | Программалық қамтаманы іске асыру       | 28 |
| 3.6.1 | Қажетті техникалық жабдықтар            | 28 |
| 3.6.2 | Android Studio құралы                   | 29 |
| 3.6.3 | Қолданушы интерфейсі                    | 31 |
|       | Қорытынды                               | 34 |
|       | Пайдаланылған әдебиеттер тізімі         | 35 |
|       | А Қосымшасы – техникалық тапсырма       | 37 |
|       | Б Қосымшасы – бағдарлама мәтіні         | 40 |

## КІРІСПЕ

Мобильді телефон қазіргі заманда барлық адамдардың өмірінде өте маңызды рөлді атқарады. Кез келген уақытта осы аса маңызды құрылғымен бізді қызықтыратын ақпаратты табуға, олардың туыстары мен достары байланыстыра алуға, ауа-райын қарауға, өзін қалаған тағамға тапсырыс беруге болады.

Телефонда сақталған контактілер, сонымен қатар көптеген тағы басқа да ақпарат көп және несиелік карта нөмірлерін, күнделікті естелік, түрлі файлдар, адамдардың идеялары мен ойлары жазылады. Яғни телефон адамға дереу қойын кітапшасын және флэш-дискісін ауыстырады. Осы себепті жобаның клиенттік бөлімі Android платформасында жасалынған.

Адамның ой өрісі, дүниетануыны кеңейіп, күрделі ойлар туып, дамуына байланысты мобильді құрылғылар, заманауи адам өмірінің ажырамас бөлігі. Бұл адамдар арасындағы қарым-қатынас, сүйікті музыкаңызды тыңдау, аудио-визуалды ақпаратты, жазу кітапшасын және көптеген басқа функцияларды қарап шығу. Ұялы смартфон компьютердің көшірмесі және де ең үлкен ерекшелігі, сіз әрдайым өзіңізбен бірге алып жүре аласыз.

Жаңа жобамен жұмыс істеу – ең алдымен тұжырымдамалар мен қосымшаның жаңа идеясын жасау, алдыға мақсат қолдана басталады. Ол маркетингтік стратегия мен пайдаланушылар аудиториясын таңдау, аналогты қосымшалардың бар-жоқтығын зерттеу, қосымшаның негізгі атқаратын функцияларын ойластыру.

Android операциялық жүйесіне арналған «byU» мобильді қосымшасы арқылы пайдаланушылар керек пиццаға тапсырыс беру жолында өзіңе қажеттісін мобильді қолданба арқылы жасауға болады

Менің дипломдық жобамның түпкі мақсаты, кез-келген адамдарға пиццаға тапсырыс беру жолын жеңілдету және жылдамдату проектісінде қолданылатын технологиялар:

- MySQL еркін реляциялық деректер базасын басқару жүйесі;
- Java Script көп парадигмалық бағдарламалау тілі.
- HTML еренмәтін белгілеу тілі;
- Apache кең таралған веб-сервер бағдарламалық жасақтамасы;
- CSS тіректері;
- Android Studio.

## **1 Негізгі бөлім**

### **1.1 Жобаның ерекшеліктері**

Дипломдық проектінің өзектілігі пиццаға тапсырыс беру жолында өзіңе қажеттісін мобильді қолданба арқылы жасауға болады. Менік жобамның қызметі, ол қолданбаға қажетті API құру, және сол құрылған API-мен деректер қорын тікелей байланыстыру еді. Байланысқа қауіпсіз байланысқа түскен соң, API-ға келген сұраныстарға JSON түрінде қайта өзіне қайтарады.

Мобильді қолданба үлкен аудиторияға арналған , сондықтан қосымша ыңғайлы интерфейске сәйкес болуы керек, жоғарғы технологияларға, көпфункционалды және тартымды дизайн жағынан артықшылығы көп болуы қажет.

Бүгін күнде адамдар мобильді телефонсыз өмір сүре алмайды деп айтуға да болады. Қазіргі заманда барлық дүниежүзінде телефонның мыңдаған түрі таралған. Алайда, оның ең көп бөлігі Android операциялық жүйесінде жасалынған. Осы себепті жобаның клиенттік бөлімі Android платформасында жасалынған. Мобильді қолданбаны қолдануды адамдар күнделікті қарапайым әдетке айналды. Сол себепті, күнсайын барлық дүниежүзінде мыңдап түрлі адамның өміріне керекті қосымша қолданбалар құрылып жатыр. Әрқайсысы, сан түрлі категорияларға байланысты жарыққа шығып жатыр, біреуі қолданушылардың бір-бірімен хабарласуына шықса, келесілері онлайн әндер тыңдау немесе видео көру үшін. Сол себепті, менің дипломдық проектімнің түпкі мақсаты пиццаға тапсырыс беруді қолайлы ету.

### **1.2 Мақсаты**

Ғаламторда операциялық жүйелер туралы мәліметтерді сұрыптап, зерттеу нәтижелерін қорытындылай келсек, қазіргі уақытта ең көп тараған операциялық жүйе Android OS. Себебі, Android түрлі өндірушілердің көптеген құрылғыларын қабылдайды. OS Android-тің еркін дамуының негізгі себебі, ол құралдарының тегін таралуында, ал IOS жүйесінде дамыту бастапқыда үлкен шығынды талап етеді.

Жобаның негізгі міндеті пиццаға тапсырыс беру жолын жеңілдету және қолжетімді жасау. Жоғарыда аталған артықшылықтар тапсырыс беруге мүмкіндігінше ыңғайлы ететін түрлі функциялар мен қосымшалармен жабдықталған заманауи Android құрылғыларының кең таралуына әкеледі. Соған байланысты, қазір күніне мыңдаған мобильді қосымшалар жарыққа шығып жатыр. Әрқайсысы, неше түрлі бағытқа бағытталған, бірі адамдардың тілдесуіне болса, ал, бірі жаңалық оқуға арналған. Жобаның клиенттік бөлімі Android платформасында жасалынған. Сондықтан, менің дипломдық



жұмысымның түпкі мақсаты пиццаға тапсырыс беруді қолайлы ету.

### 1.3 Android жүйесінің негізгі мүмкіндіктері

Android – бұл смартфондар, планшетті компьютерлер, электрондық кітаптар, сандық ойыншылар, смарт-сағат, ойын консолі, нетбуктар, смарт-кітаптар, Google көзілдіріктері, теледидарлар, автокөлікті басқару жүйесі және басқа құрылғылар үшін операциялық жүйе. Linux операциялық жүйе ядросына және Google-дің Java Virtual Machine іске асыруға негізделген. Бұл бастапқыда AndroidInc компаниясы әзірледі, Google 2005 жылы сатып алды. Кейіннен Google платформаны қолдау және одан әрі дамытумен айналысатын OpenHandsetAlliance (ОНА) альянсын құруды бастады. Android сізді Google-дың дамыған кітапханалары арқылы басқаратын Java қолданбаларын жасауға мүмкіндік береді. AndroidNativeDevelopmentKit C және басқа тілдерде жазылған кітапханаларды және бағдарлама компоненттерін порттауды пайдалануға мүмкіндік береді. Android операциялық жүйесі смартфондардың 86% -на орнатылды [10].

Android – мобильді құрылғылар үшін - телефондар мен планшеттер үшін жалпы операциялық жүйе. Бұл жүйе бүкіл әлемдегі көптеген пайдаланушыларға танымал және тартымды ететін көптеген ерекшеліктері бар.

Android операциялық жүйесі әртүрлі конфигурацияларда жұмыс істей алады. Міне, сондықтан әлемнің көп өндірушілері осы ОЖ құрылғысын жабдықтайды, өйткені басқа бағдарламалық жасақтама өнімдер нақты техникалық сипаттамаларға сәйкес келетін жеке құрылғыларға арналған. Android-дің бұл икемділігі жүйенің Linux ядросына салынғандығына байланысты, ашық бастапқы коды бар, ол әзірлеушілерге шектеусіз мүмкіндік береді. Android 256 Мб-тан кем емес жедел жадыдағы құрылғыларда іске қосылуы мүмкін. Жүйенің ең жаңа нұсқалары 512 МБ оперативті жадты талап етеді, бұл қазіргі заманғы құрылғылар үшін де шамалы. Жүйе жоғары өнімді процессорды қажет етпейді және 600 МГц ядросы бар құрылғыларда жұмыс істей алады [10].

Амалдық жүйе әлемдегі ең ірі бағдарламалардың дерекқорын беретін ресми Google репозиторийінен бағдарламаларды орнатуға мүмкіндік береді. Бұл әрбір әзірлеуші құрылғыға арналған кез-келген бағдарламаны өз бетінше жаза алады және оны дүкенге орналастыра алады. Операциялық жүйенің ашықтығы арқасында да мүмкіндік туады. Android құрылғыларындағы қолданбаларды тікелей телефоннан немесе планшеттен немесе .apk файлын жүктеп, одан кейін оны құрылғы арқылы орнату арқылы компьютер арқылы орнатуға болады [10].

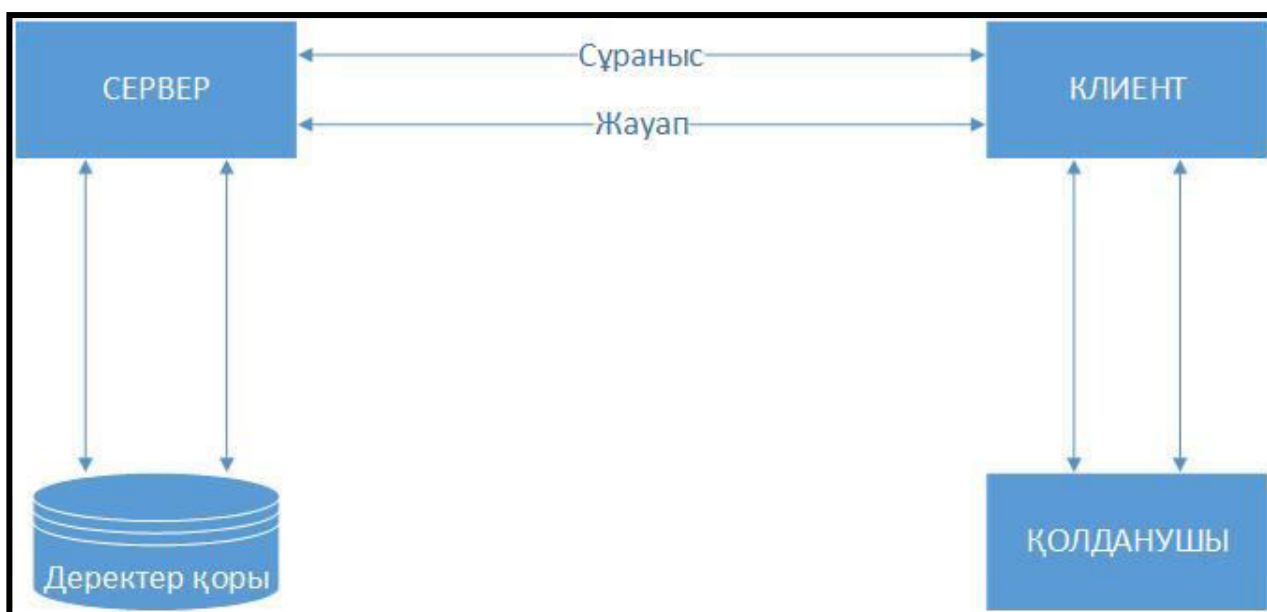
Android-тың айрықша ерекшелігі оның Google қызметтерімен интеграциясы – Gmail, Hangouts, Voice Search және т.б. Android жүйесінде Chrome браузері арқылы смартфондағы қойындыларыңызды компьютер

браузері арқылы үндестіруге мүмкіндік беретін Chrome қолдау қызметі ресми түрде іске асырылды [10].

Мысалы, телефоннан шолу беттерін бастауға болады және қаласаңыз, екінші іздеуді пайдаланбастан компьютердегі бірдей қойыншаны ашу арқылы ақпаратты зерттеуді жалғастырыңыз [10].

«Android» – қарапайым және интуитивті интерфейс. Барлық қажетті қосымшалар негізгі мезгілде орналастырылады экранды және құрылғының мәзірінде орталық сенсорлық пернені немесе экрандағы тиісті түймені басу арқылы шақырылады. Барлық параметрлер «Параметрлер» бөлімінде орналасқан және әрбір пайдаланушы әрекеті машинаны алғаш рет бастағанда ескертулермен және кеңестермен түсіндіріледі. Операциялық жүйе пайдаланушыға шұғыл түрде жауап береді және орнатады және қажетті бағдарламалар мен файлдарды басқа заманауи мобильді ОС-ға жоғалтпай жылдамдықпен жүктейді [10].

Клиент-сервер моделі клиенттер немесе клиенттер деп аталатын серверлер және қызмет сұраулары деп аталатын ресурс немесе қызмет провайдерлері арасында тапсырмаларды немесе жұмыс жүктемелерін таратылатын бағдарлама құрылымы. Клиент-сервер моделін қолданатын компьютерлік қосымшалардың мысалдары – электрондық пошта, желілік басып шығару және бүкіләлемдік веб. REST технологиясы 1.1-суретте көрсетілген [11].

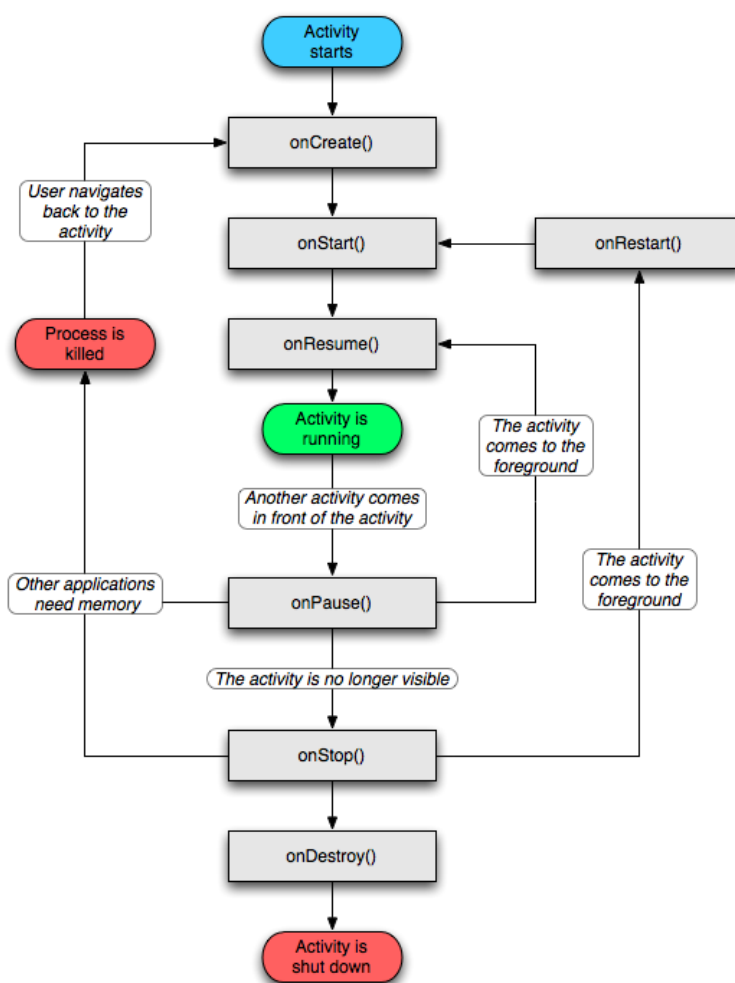


**1.1-сурет – REST технологиясы**

Барлық параметрлер «Параметрлер» бөлімінде орналасқан және әрбір пайдаланушы әрекеті машинаны алғаш рет бастағанда ескертулермен және кеңестермен түсіндіріледі.

## 1.4 Android қосымшасының жалпы схемасы

Android қолданбалары өздерінің жұмысында терезелерді (Windows-ге ұқсас) пайдаланады, алайда осы жүйеде жоғарыда көрсетілген терезелер басқа атаумен әрекет етеді – Қызмет. Windows-дегідей, әр терезе өзінің жеке өмірлік цикліне және өз ерекшеліктеріне ие. Жаңа терезе жасаған кезде, onCreate () әдісі әзірленуде, бұл әдіс қайта анықталады және бағдарлама және оның құрамдас бөліктері инициализацияланады. Содан кейін onStart () және onResume () әдісі шақырылады. Екі әдіс де терезе пайда болған немесе қалпына келтірілгенде көрсетіледі (басқа бағдарламадан, кішірейтілген бағдарламаны кеңейту кезінде және т.б.). Қоқыста кезде onPause () және onStop () әдістеріне шақырылады. Қолданбаны және терезені жабу кезінде, OnDestroy () шақырылады, бұл әдіспен пайдаланушы деректерін және параметрлерін сақтауға болады. Қоңырау әдістерінің толық сипаттамасы мен дәйектілігін ресми веб-сайтынан табуға болады. Android қосымшасының жалпы схемасы 1.2-суретте көрсетілген.[12].



1.2-сурет – Android қосымшасының жалпы схемасы

## 1.5 Web – технологияның қазіргі жағдайы

Интернет байланысы көмегі арқылы электрондық поштаның көмегімен хабар алмасып (беріп), басқа дербес компьютерлердегі мәліметті көріп, алыстан телеконференцияларға қатысу жұмыстарын өткізуге мүмкіндік болады. TCP/IP – Интернет байланысына қосылған дербес компьютерлер арасындағы мәлімет алмасуды қамтамасыздандыратын ақпаратты бір құрылымға келтіру ережелері, болмаса оларды құру хаттамасы. IP (Internet Protocol) – ақпаратты, оны алушының адресі бойынша көрсетілгендей шағын параметрлері бар болғандықтан, бірнеше бөліктерге немесе бөлімдерге бөлетін желі арасындағы хаттама болып табылады. TCP (Transmission Control Protocol) – мәліметті жөнелтуді басқаратын хаттама болады, ол желі бойындағы мәлімет дестелерін толықтай дұрыс жеткізуге жауапты болып табылады[3].

Ғаламтор, сонымен қатар ең мықты, сонымен қатар тәуелсіз мәлімет қоры, жілінің сенімді немесе жылдам тәсілі, бүкіл әлемдегі көптеген қолданушылардың шығармашылыққа салып өзін жариялауы және ақпараттық технологияларды дамыту ең негізгі жұмыс болып табылады. Ғаламтордың басты мақсаты бұл – әр күндік, жоғарғы сапалы байланыс құру болып табылады. Ғаламторға қосылған кез-келген құрылғылар бір-бірімен таңдаған уақытта байланыса алу мүмкіндігі бар [4].

Интернет байланысын пайдаланып, үйден шықпай-ақ, мыңдаған қалаларды аралап мұрағаттарды, кітапханалар көріп, бүкіләлемдік ғылыми және мәдени жетістіктермен танысып, оларға өркениетті елдің қолданушысы ретінде өз пайдаңызды қосып, өзіңізді әлемнің бір кішкене бөлігі болатындай сезіну мүмкіншілігі пайда болуы мүмкін. Ғаламтор – TCP/IP хаттамаларына негізделіп желі арасындағы байланысу технологиясының бірі болып табылады. Бағдарламалық қамтама етуінің құрастырушысы және қарапайым пайдаланушысы үшін Web-технологияның мәні – бұл ең алдын заманауи, қажет технология екенін бұрыннан анық. Web-технологиялардың қызығушылығы – қолданушы мен дербес компьютер арасында жүретін ақпаратты тасымалдау құрал есебінде көбінесе әмбебап құрал бола отырып интерфейсті анықтайды[3].

Бүкіләлемдік ғаламтордың іргетасы – HTML (Hyper Text Markup Language) гипермәтіндік белгілеу тілі болып табылады. Ол құжаттың (веб-беттің) физикалық белгілеу үшін қызметін атқарады. Веб-беттегі мазмұнның бейнесін басқару үшін (CSS) стильдердің каскадты кестелері үшін арналған. Көбінесе CSS, белгілі мәтіндік Word жүйесінде қолданылатын стильдерімен өте ұқсас болып кездеседі. Веб-беттерге қимыл (жылжымалы мәзірі, анимация) енгізу үшін JavaScript бағдарламалау тілі қолданылады [4].

Ғаламторда веб-бетке қимыл ендіру үшін стандарттық скрипттік тілі JavaScript бағдарламалау тілі болып табылады. JavaScript тілінің ядросы ECMAScript.

HTML, CSS, JavaScript тілдерінің арқасында қандай сайт керек болса да, қиындығына қарамастан іске асыруға болады.

Ғаламторда мындаған ірі интернет-провайдерлер орналасады, соған қоса олардың магистральды байланыс желілері NAP арқылы кез-келген қалаларда жасалады, айтарлықтай мәліметтердің басым ағыны NAP түйіннің неше түрлі байланыс желілері арқылы таратылады. Үлкен және кіші байланыс желілердің бірігуі (ғаламторды құрайтын) негізінде шартты түрдегі келісімдер жатады. Кез-келген клиенттің белгілі ISP пен өзінің дербес компьютерін, болмаса жергілікті байланыс желісін провайдер желісіне қосуы бекітілген келісім шарты бар. Кейбір ISP-A провайдерлердің қолданушылары ISP-A желісіне қосылуға арналған келісім құрайды, сонымен қоса ISP-A, ISP-B мен байланыс желілерін біріктіру үшін келіседі, артынан солай жалғасады [3].

Провайдерлер – байланыс желілерінің иерархиясы. Алуан түрлі елдерде еларалық, ұлттық және қалалық болып бөлінетін мындаған провайдерлер болады. Аймақтық провайдерлердің байланыс желілері (екіншілік) ұлттық провайдерлердің байланыс желілерімен (біріншілік) жоғары жылдамдықтағы каналдар арқылы байланысады. Мысалы, АҚШ мемлекетінде T1 стандартының ақпаратты беру жылдамдығы 1,544 Мбит/с, ал арна болса жылдамдығы 44,74 Мбит/с болатындай T3 арнасы [4].

## 2 Жобалау бөлімі

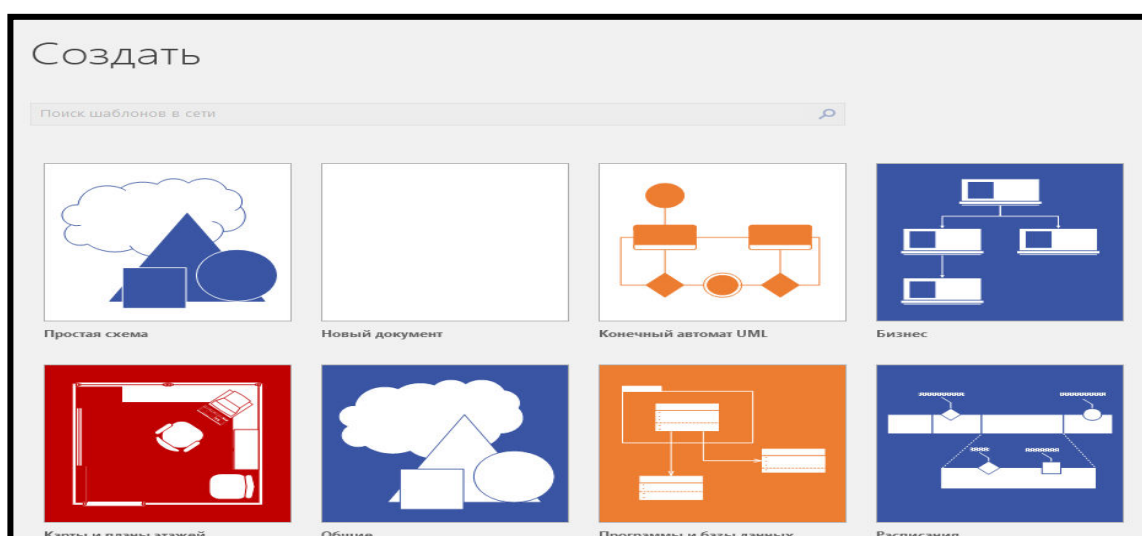
### 2.1 Унифицирленген моделдеу тілі

UML (Unified Modeling Language) - бағдарламалық жасақтаманың артефактілерін көрсету, визуализациялау, салу және құжаттандыру үшін қолданылатын стандартты тіл. UML – Object Management Group компаниясы (OMG) құрды және UML 1.0 спецификациясының жобасын OMG-қа 1997 жылдың қаңтарында ұсынды [1].

### 2.2 Microsoft Visio 2016 құралы

Microsoft Visio – бұл векторлық графикалық редактор, ол сізге кез келген компанияның кез-келген бөлімшесінің қызметкерлері үнемі кездесетін мәселелерді шешу үшін блок-схарттар мен диаграммаларды жасауға және түзетуге мүмкіндік береді. Оның түпнұсқасы 1992 жылы Visio корпорациясы шығарған және Microsoft компаниясын сатып алған 2000 жылға дейін тұрақты түрде жаңартылған. Осыдан кейін, Microsoft Office-мен біріктірілген өнімнің ребрендингі және жаңа нұсқалары Microsoft Visio атымен шығарылды.

Microsoft Visio – 2016-да әзірлеушілер Tell me атты қосалқы өрісті қосқан. Оны орындағыңыз келетін пәрменді сипаттайтын сөзді енгізіңіз және барлық ұқсас сұрауларыңыз бар тізім сізде пайда болады [2]. Microsoft Visio 2016 құралы 2.1-суретте көрсетілген.



2.1-сурет – Microsoft Visio 2016 құралы

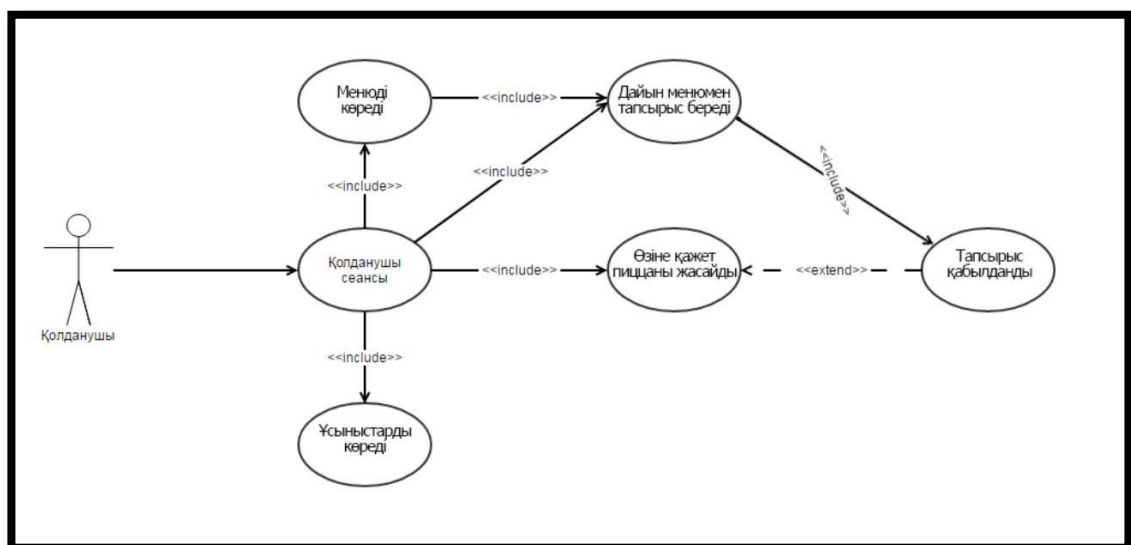


## 2.3 Прецеденттер диаграммасын құру

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

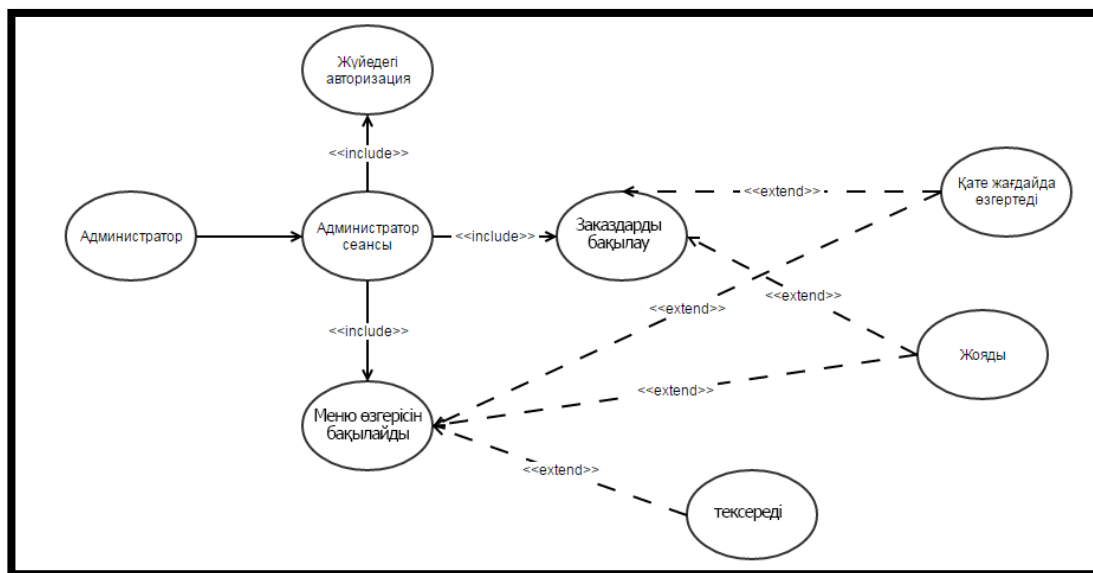
Жүйенің жасалатын жұмысы онымен қалай айналысатынына және не қажет болуына байланысты болады. Объект моделі өзгергіштік тараудағы прецедент моделімен байланысты болады. Динамикалық модельді құру негізінде «сырттан ішке» деп аталады. Орналасқан жүйеден басқа жерде болуы мүмкін пайдаланушы зерттеуінен, сонымен қоса ақпараттың қимыл бөліктерін ашу жолынан басталады. Осындай тұрыптан ақпарат құрылымы пайда болады, яғни, қажет ететін іс-әрекеттің ұсынуынан болады. Ішкі бағыттың жылғуы, сонымен қоса оның қадамы бөлек бір жоспары болып табылады [8].

Қолдану типтері (Use Case) жүйе және активті субъект арасында болатын диалогты модельдеуге мүмкіндік ұсынады, сонымен қоса функцияны соңына қарай бейнелейді. Жүйені қолдану типтер жиыны оны пайдаланудың көптеген амалдар назарына бағытталған. Қолдану типтері – транзакция жүйесі бойынша орындалатын тізбектілік, осы арқылы табылған активті субъект, яғни, керекті нәтижені оңай алуға болады. Қолданушының прецеденттер диаграммасы 2.2-суретте көрсетілген.



2.2-сурет – Қолданушының прецеденттер диаграммасы

Қолдану диаграммасы (Use Case Diagram) – бұл активті денелердің көптеген графикалық көрінісі оны қолданудың, ол болмаса басқа нұсқалардың амалдарымен өзара байланысады. Жүйені жоспарлағанда құрылымның кілттік функциясын, сонымен қатар көптеген қолданушылар ұсынатын негізгі диаграмма (Main Use Case Diagram) осылай конструкцияланады(2.3 – сурет).



**2.3-сурет – Администратор прецеденттер диаграммасы**

Use Case – бұл белгілі актерге (Actor) арналған, белгіленген бір нәтиже бере алатындай, бағдарлама жүйесімен орындалатын іс-қимылдар тізбегінің сипаттамасы. Ал, прецедент негізінен моделдегі берілген мәндердің тәртіптік структурасын жасайды, сонымен қатар прецеденттер кооперацияның қатысуымен таратылады.

Актер (Actor) – бұл берідген жүйедегі негізгі деген элементтермен байланыста болатын негізгі объектінің рөлі болып саналады. Актер мен қолданушының салыстырмалы түрде айырмашылығы келесідей: қолданушы жүйені толықтай пайдаланатын шынайы объект. Ол бірнеше рөлде ойнауы мүмкін болады, соған байланысты ол бір емес, бірнеше актер болуы мүмкін [12].

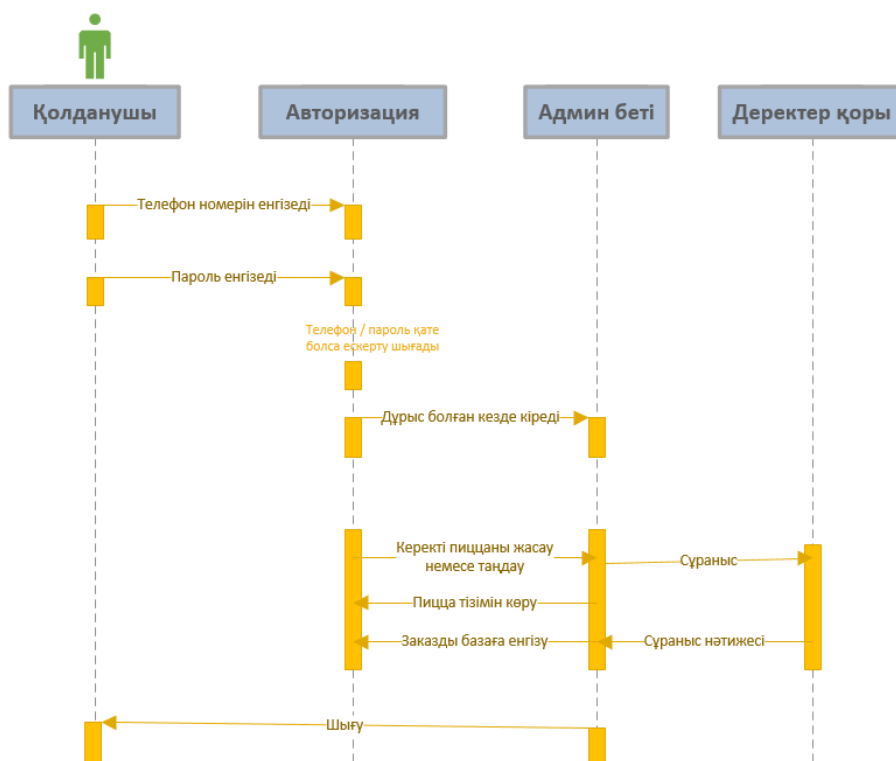
## 2.4 Тізбек диаграммасын құру

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

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

- прецеденттегі артындағы мәтінінің іс-қимылы. Ол жоғарыдан төменге жазылады, және сол жақтан басталады. Сол терезеде іс-қимыл сипатталуы болып, орындалатын жұмыс уақытындағы белгілі ақпараттар қызмет етеді;
- объектілер "объект-класс" кейпінде аты болмаса объект данасының нөмірі және класс объектісінің атауы жазылады;
- хабарландыру, бір бағытта көрсетілген белгілі бір объектіден келесі объектіге бағытталған іс-қимыл туралы мәлімет хатынан тұрады. Берілген бір уақытта орындалуы мүмкін, және осы іс-әрекеттегі берілген жүйенің қайтарымды реакциясы болуы мүмкін болып келеді;
- әдістері (операциялар). Тікбұрыш түрінде болады. Олар үздік сызықпен бейнеленген. Яғни, берілген әдістерге кіретін сол объектілер болып табылады. Тік бұрыштың ұзындығы артынан қуушыда басқару нүутесін көрсетуде қолданса болады: Тікбұрыш аяқталатын немесе бітетін әдіс толықтай нүктесіне дейін басқарумен игеріледі. Бұл, берілген үшбұрыштар циклге арналған объекті түзуі деп саналады.

UML талаптарына байланысты объектінің тізбектелген диаграммасында тіктөртбұрыш кейпінде көрсетіледі. Объектіні 3 түрлі есіммен атауға болады: тек оның атауын көрсетіп кету қажет, объект пен класстың атауын беру қажет, не класстың атауымен шектелу керек. Тізбек диаграммасы 2.4-суретте көрсетілген.



2.4-сурет – Тізбек диаграммасы

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

## 2.5 Кооперация диаграммасын құру

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

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

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

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

Кооперация сызбалары жүйе жұмысы барысында субъектілердің өзара қарым-қатынасын бейнелейді. Мұндай сызбалар жүйе тәртібінің жоспарларын модельдейді. Кооперация диаграммасы 2.5-суретте көрсетілген.



2.5-сурет – Кооперация диаграммасы

## 2.6 Күй диаграммасын құру

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

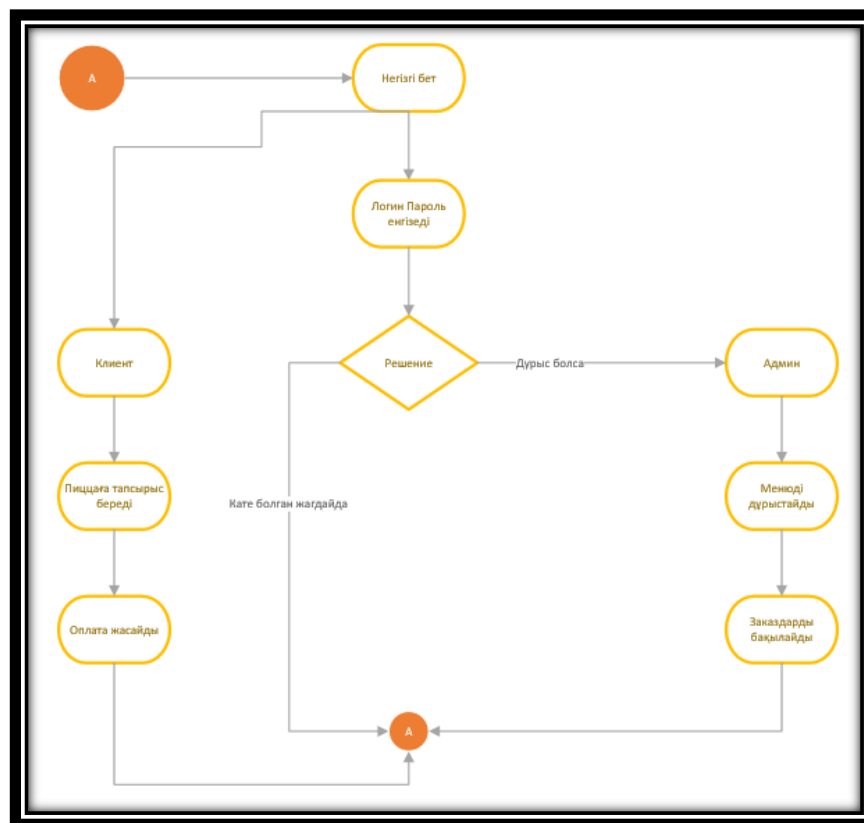
Күй диаграммасы – нысанның бір қалпынан баскасына ауысуын айыратын, және іс-әрекеттері қалпыларын алмастырумен ескертілінген, оқиғалардың, болмаса хабарлаулардың объекттік жағдайларының графикалық түрдегі ұсынысы.

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

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

Объектілер жағдайларын моделдеу күйлер диаграммасының арқасында жүзеге асады. Күйлер сызбасы (автомат) бұл күйлер мен өзгерістер графы. Күйлер моделі системаға маңызды кластар үшін сызылады.

Объектілер белгілі бір нүктеден шығып, соңғы нүктеге жеткенше барлық нүктелерден өтеду, және соңғы нүктеге жетеді. Күй диаграммасы 2.6-суретте көрсетілген.



2.6-сурет – Күй диаграммасы

## 2.7 Класстар диаграммасын құру

Класс диаграммасы – веб-бет кодын ендіру үшін басты диаграмма классы сызба көмегімен жүйенің ішкі жүйесі сипатталады, яғни мұралауды сипаттауын, сомен қоса қосымша класстар бір-бірімен тығыз байланысты. Осындай жүйенің логикалық сипаты бейнеленеді. Класстар ол тақырыппша. Артынан физикалық нысандар, сонымен қоса ақпарат кодтары жазылады.

Диаграмма кластары жүйенің ортақ бейнесін сипаттайды, яғни, жүйе объект құрылымын. Диаграмма класстар қорытындысында белгілі уақытта өңделуі мүмкін. Осыдан диаграммалар, сонымен қоса спецификациялық класс өңделулері автоматты түрде жаңартылады.

Диаграмма кластары дайын жүйе сұрыптау сияқты қолдануы мүмкін сонымен қатар жаңа әзірлеуге тән. Ерекшеленген класстарда әртүрлі әдістер, сонымен қоса жүрістер қолданылады. Солай, Барами (Bahrami) кластарға бекітілуіне (class discovery) 4 негізгі құрылымды анық етіп оқиды. Келесі жүрістерден тұрады:

- есімі қолданылатын группа негізіндегі жүріс;
- белгілі класс үшін ортақтандырылған шаблондарды қолдану негізіндегі жүріс;
- прецеденттерді қолдануды негізіндегі жүріс;
- CRC жүрісі.

Кластың параметрлерін бейнелеу үшін орындалатын операцияны оңай сызу спецификация класы.

Класс ортақ бейнелері бар (атрибуттары), тәртібі (функциялары), семасиология және басқа объекттермен байланысқан объекттер тобырын анықтайды. Класстың объектіні құруға тағайындалған шаблону ретінде қарастыруға болады. Кластар диаграммасы 2.7-суретте көрсетілген.



2.7-сурет – Кластар диаграммасы

Әрбір объект белгілі бір ғана кластың сызбасы мысал ретінде келесідей бейнесі бар курсты ұсыну үшін класс анықтамасын қарастырайық:

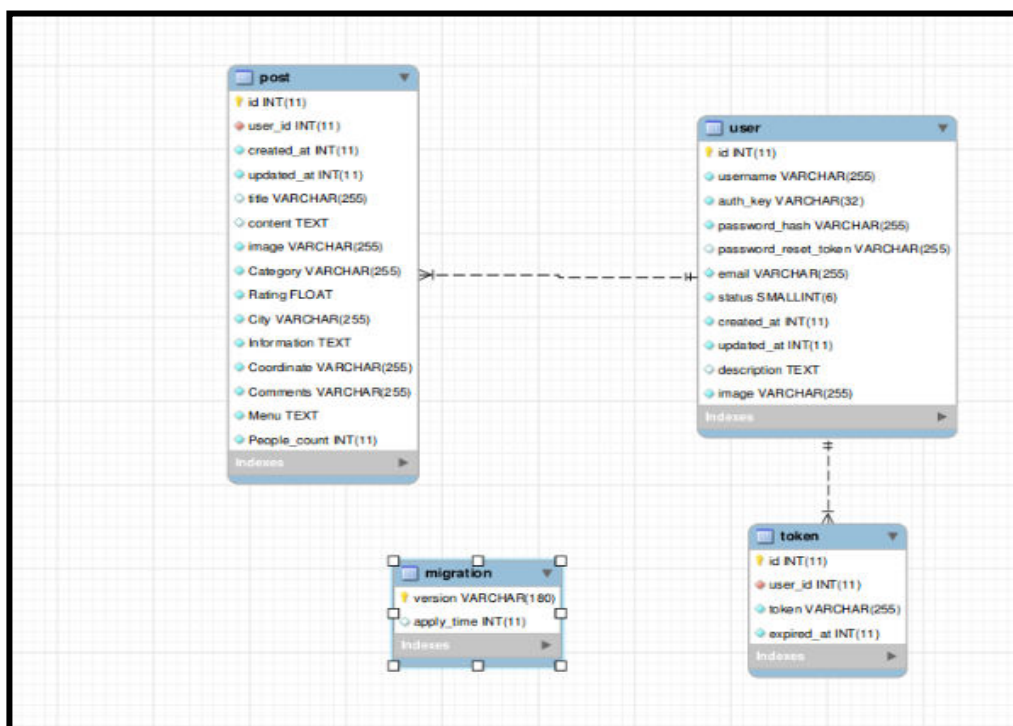
- атрибуттар;
- методтар.

Диаграмманың бұл типі сипатталған класстардың негізгі құрылымын негізге ала отырып, сонымен қатар логикалық көрсетілім жүйесін құрастыруға рұқсат етеді. Диаграмма белгілемелері класстар мен интерфейстердің өзара арасындағы байланысын қиын иерархиялық жүйеде сипаттауға рұқсат береді. Диаграмманың берілген кейпі жүйе объектілері сипатталатын Collaboration диаграммасына қарама-қайшы. Rational Rose бағдарламасының диаграммасының берілген белгілі түрі көмегімен неше түрлі нотациялар құруға рұқсат береді.

Класстар арасындағы қарым-қатынасты бекіту. Агрегаттық қатынастар, сонымен қатар мұрагер қатынастар болып табылады. Қолдану қатынастары. Ассоциативті болып табылатын байланыстар, бағыттық, сонымен қатар уақыттық және қалыпты байлансытар.

## 2.8 ER диаграмманы құру

Проектінің ER диаграммасы 2.8-суретте көрсетілген.



2.8-сурет – Проектінің ER диаграммасы



### **3 Қолданылған бағдарламалық қамтамалар**

#### **3.1 MySQL технологиясы**

MySQL – бұл қазіргі замандағы ең көп қолданылатын деректер қорының бірі. MySQL бұл қолданысқа өте тиімді, сонымен қатар үлкен жылдамдықта жұмыс жасай алады. MySQL деректер қоры ANSI 92 белгіленген стандартындағы SQL (Structured Query Language) сұраныстар тілінде кез-келген сұраныс құра алады. MySQL тілі арқылы кез-келген көлемдегі деректерді қосуға және жоюға белгіленген шарттармен бірге көп жүзеге асырылады.

MySQL-дің қысқаша мүмкіндіктері:

- дерекқорды басқару жүйесі болып табылады;
- реляциялық ДББЖ болып табылады;
- ашық бастапқы бағдарламалық жасақтама болып табылады;
- мульти-ағынды SQL-серверден тұратын клиент-серверлік жүйе.

MySQL – деректер қорының көпағымды бола алуы. MySQL деректер қоры жоғарғы жылдамдығымен, және төзімділігімен оңтайлы қолданудың жеңілдігімен аса ерекшеленеді.

#### **3.2 Javascript программалау тілі**

JavaScript программа тілі бренд Эйк есімді программист тарапынан Netscape Communications-та Netscape Navigator броузерлері үшін 2.0-версиясынан басталып, сценарийлер тілі ретінде өндірілген болатын. Бұл тілдің кейінгі дамытылу жұмыстарына Майкрософт Корпорациясы да қосылды, олардың Internet Explorer броузерлері JavaScript-ті үшінші версиясынан бастап, қолдайды. Майкрософт версиясы JavaScript деген атауға ие болды, себебі, JavaScript тілі Netscape фирмасының тіркелген маркасы болып табылады. 1996 жылы ЕСМА осы тілді стантартизациялау туралы шешім қабылдады, 1997 жылдың маусымында ECMAScript (ECMA-262) есімді бірінші версиясы қабылданған болатын. 1998 жылдың сәуірінде бұл стандарт ISO тарапынан ISO/IEC 16262 нөмірімен халықаралық стандарт болып қабылданды. Біз, келесі қатарда ЕСМА стандартының үшінші версиясымен (1999 ж. желтоқсан) танысамыз, бірақ келесі екі себептен ECMAScript атауын емес, JavaScript атауын қолданамыз .

Бұл атау тарихи тұрғыдан бірінші болып табылады, және осы атаумен осы тіл кеңінен танымал. Сәйкес MIME типі, дәлірек айтқанда, «text/javascript» барлық броузерлерде оқылады, ол Jsript немесе ECMAScript-ке қарағанда осы тілдегі сценарийлерді қолдаймыз.

JavaScript бұл сайттың функционалдылығын көбейту мақсатымен html

құрамында қолданылатын программалау тілі. JavaScript Netscape Netscape фирмасы тарапынан Sun Microsystems фирмасымен бірлестік ішінде Sun's Java тілінің негізінде өндірілген болатын. Java скриптер клиент тарапында орындалады, дегенмен алғашқыда Liveware – JavaScript-тің көмегімен html-дің стандартты тәгтерімен жасауға болмайтын әдемі және интерактивті веб-беттерді жасауға болады.

JavaScript бұл клиенттің жағында да, сервердің жағында да жұмыс жасайтын сценарийлерді жазуға арналған объекті-бағытталған программалау тілі. Сондықтан ол «толығымен» программалау тілі болып табылмайды, тек сценарийлер орындалатын аяның мүмкіндіктерін қолдануға бағытталған (3.1-сурет).

```
1  <!DOCTYPE HTML>
2  <html>
3
4▼ <head>
5      <!-- Кодировканы анықтайтын мета тэг -->
6      <meta charset="utf-8">
7  </head>
8
9▼ <body>
10
11      <p>Документтің басы...</p>
12      <script>
13          alert( 'Сәлем, Әлем!' );
14      </script>
15
16      <p>...Документтің соңы</p>
17
18  </body>
19
20 </html>
```

**3.1-сурет – JavaScript бағдарламалау тіліндегі «Сәлем әлем» коды**

Скриптер қолданушы әрекетімен басталған қандай да бір әрекет нәтижесінде орындалады. Бұл әрекет, мысалға, курсорды керекті жерге апару, батырманы басу және т.б. болу мүмкін. JavaScript сізге функционалды веб-беттерді жасауға көмектеседі. JavaScript және Java тілдерінің екеуі екі түрлі программалау тілі екенін айта кету қажет. Java бұл – тек компилятор мен көмекші файлдардың көмегімен жұмыс істейтін ОБП тілі. Мұндай жобалар не біткен қосымша ретінде, не іштей құрылған апплеттер ретінде жұмыс істей алады. JavaScript-тер html-дің ішіне құрылса, клиенттік компьютерде бөлек файл ретінде сақталады. Бұл екі тілдің арасындағы айырмашылықты қазірше көрсетпей-ақ қояйық. Жай ғана, Java Script тілінің Java емес екенін есте сақтаңыз. Сонымен, Java скриптер HTML тілінің ішіне орналасып, бөлек программа ретінде орындала алмайды, тек браузерде оқылып тұрған кезде функцияланады. Осы Java Script тілі мен кез келген басқа программалау тілдерінің арасындағы айырмашылық. Осылайша, Java Script скрипттерін

жасау, кез келген мәтіндік редактормен оған сәйкес браузерді талап етеді. Java Script тілінде скриптті жұмыс істетуге үшін, скриптерді өңдеп отыра алатын браузер ғана керек.

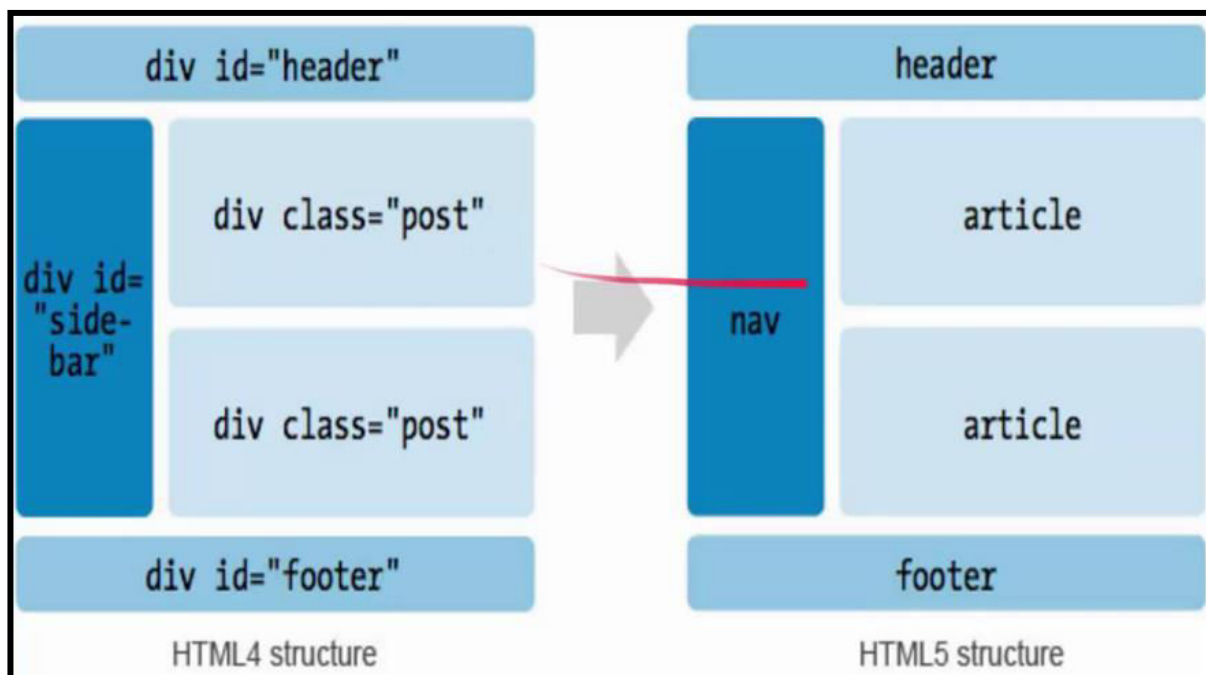
### 3.3 HTML тілі туралы түсінік

HTML(Hyper Text Markup Language) – гипермәтінді белгілеу тілі ретінде аударуға болады. Осы тілде веб-беттер мен сайттар жасалады.

HTML – бағдарламалау тілі емес. Ол мәтіндік құжаттарды белгілеуге арналған. Яғни, оның арқасында мәтіндік пішімдеуді жүзеге асырамыз.

Әрбір тег браузерге арнайы команданы береді, браузер бұл команданы түсінеді және белгілі бір түрде пішімделген дайын мәтінді көрсетеді. Сондықтан, сіз кодтың өзін көрмесеңіз, браузер монитордың экранында дайындалған пішімделген мәтін мен графиканы көрсетеді.

HTML5 арқасында веб-беттер пайдаланушылардың браузерлерінде жергілікті деректерді сақтауды үйренді, бұл HTTP кукилерінен бас тартуға мүмкіндік береді. Мазмұн тезірек және қауіпсіз жеткізіледі. HTML5 браузердің өзара үйлесімділігін жеңілдетті. Әдетте браузерлер мультимедиа файлдарын ойнату үшін түрлі плагиндерді пайдаланады. Түпнұсқа аудио және видео қолдау үйлесімділік мәселелерін түзетеді. Жаңа төлсіпаттар ойнату, кідірту және дыбыс сияқты басқару түймелерін қосады. HTML 5 пен HTML 4 тілдерінің айырмашылықтары 3.2-суретте көрсетілген.



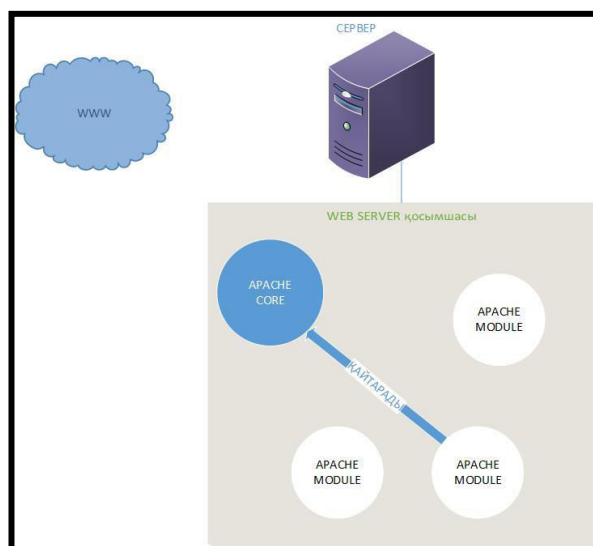
3.2-сурет – HTML 5 пен HTML 4 тілдерінің айырмашылықтары

HTML (Hyper Text Markup Language) сөзбе сөз аударғанда, «гипермәтінді белгілеу тілі» болып аударылады. Бұл тіл Web беттерді жасаған кездегі программалаудың негізгі тілі болып табылып, белгілі командалар көмегімен беттің әр түрлі элементтерін (мәтінді, графиканы, кестені және т.б.) орналастыру үшін қолданылады. Өзінің қарапайымдылығының арқасында, ол басқа программа тілдерінен бөлек тұрады. Оны информатика негіздерін білмейтін қарапайым мектеп оқушысы да түсініп, үйрене алады, себебі алғашқыда ол ПК қолданушыларының кең ортамы үшін ойластырылған болатын. HTML-дың көмегімен қарапайым және қиын емес сайттарды жасауға болады. HTML тілі тек браузерлер және графикалық НТМ-редакторлары (визуалды режимдегі WEB беттерін құрау үшін арналған арнайы программалар) тарапынан оқылады.

Дегенмен, мұнда өндірушілердің бір сүрінетін жері барлық браузерлер бір элементті бірден түсінбеуі мүмкін, бастаушы программистер көптеген HTML редакторларда болады және мәселені шешуге көмектесетін арнайы іштей құрылған құралдарға бұрулары керек.

### 3.4 Apache2 веб сервері

Apache HTTP – сервері немесе Apache қысқаша – Apache Software Foundation әзірлеген танымал веб-сервері. Apache көбінесе PHP және MySQL сценарий тілінде қолданылады. Бұл комбинация әдетте LAMP (Linux, Apache, MySQL, PHP) деп аталады. Apache веб-серверінің жұмыс істеу принципі 3.3-суретте көрсетілген.



3.3-сурет – Apache веб-серверінің жұмыс істеу принципі

Apache – кросс-платформалық бағдарламалық қамтамасыз ету, Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS қолдайды. Apache-дің басты артықшылығы – конфигурацияның сенімділігі мен икемділігі. Ол деректерді беру үшін сыртқы модульдерді қосуға, пайдаланушыларды түпнұсқалы растауға арналған ДББЖ қолдануға, қате туралы хабарламаларды өзгертуге және т.б. IPv6-ға қол жеткізуге мүмкіндік береді.

Apache ядросы конфигурациялық файлдарды өңдеу, HTTP протоколы және модульді жүктеу жүйесі сияқты негізгі функцияларды қамтиды. Ядро (модульдерден өзгеше) Apache Software Foundation тарапынан үшінші тарап бағдарламашыларының қатысуынсыз толығымен әзірленген.

Теориялық тұрғыдан алғанда, апачек өз модулін пайдаланбай таза түрде жұмыс істей алады. Алайда, осы шешімнің функционалдығы өте шектеулі.

Apache ядросы C бағдарламалау тілінде толығымен жазылған.

### 3.5 CSS бағдарламалау тілі

CSS (Cascading Style Sheets) – каскадты стиль кестелері HTML сияқты белгілеу тілінде жазылған. Құжаттың көрсетілуін сипаттау үшін пайдаланылатын стиль тілі. CSS, HTML және JavaScript-дің қатарында World Wide Web-тің іргетасы болып табылады.

CSS көрсетілім мен мазмұнды, соның ішінде орналасу, түстер және қаріптерді бөлуге мүмкіндік береді. Бұл бөлу мазмұнды қолжетімділікті жақсартады, көрсетілім сипаттамаларының спецификациясында көбірек икемділік пен бақылауды қамтамасыз етеді, бөлек .css файлында сәйкес CSS-н көрсету арқылы бірнеше веб-беттерді пішімдеуді ортақ пайдалануға мүмкіндік береді және құрылымдық мазмұндағы қиындығы мен қайталануын азайтады. Пішімдеу мен мазмұнды бөліп алу, экранда, баспа түрінде, дауыспен (сөйлейтін браузер арқылы немесе экраннан оқу құралы) Брайль негізіндегі әртүрлі көрсету әдістеріне арналған әртүрлі стильдердегі бірдей түзету бетін көрсету мүмкіндігін береді сенсорлы құрылғылар. CSS мобильді құрылғыда қол жетімді болса, баламалы пішімдеу ережелері бар [15].

CSS сипаттамалары World Wide Web консорциумы (W3C) арқылы қолдайды. Интернеттегі медиа түрі (MIME түрі) мәтін / css RFC 2318 (наурыз 1998 ж.) W3C CSS құжаттары үшін тегін CSS тексеру қызметін пайдаланады [15].

HTML-ға қоса, басқа белгілеу тілдері XHTML, қарапайым XML, SVG және XUL сияқты CSS-ді қолдайды.

Каскадты стиль кестелері сипаттарды және осы сипаттардың жарамды мәндерін пайдаланатын элементтерді пішімдеу ережелерін сипаттайды. Әр элемент үшін шектеулі қасиеттер жиынтығын пайдалануға болады, қалған қасиеттер оған әсер етпейді [15].

### **3.6 Программалық қамтаманы іске асыру**

“Менің жасаған “ByU” жобасы негізгі жасалынған жүйе ол Android Studio. Веб бағытында жұмыс жасайтын адамның қажетті және стандартты 3 тіл ол: HTML, CSS, Java. Бұл 3 тілді білмей, сізге жаңадан қосымша әзірлеп шығару қиынға түседі. Алайда, Java тілін Android, ресми түрде қолданатын тіл деп айтсақта қате болмайды. Қалай болғанда да, барлық оқу құжаттары және де барлық онлайн курстары осы тілде негізделеді.

Мобильді қосымшаның функционалы – қолданушылар мен админге кірген кезде, оңай және жылдам жұмыс жасалуы қажет. Барлық қолданушыларға жақсы мүмкіндік жасап берілуі дұрыс.

#### **3.6.1 Қажетті техникалық жабдықтар**

Sublime Text – Python қолданбалы бағдарламалау интерфейсі (API) бар меншікті кросс-платформаның бастапқы коды редакторы. Ол көптеген бағдарламалау тілдерін және таңбалау тілдерін қолдайды, және функцияларды плагины бар пайдаланушылар, әдетте, қауымдастырылған және тегін бағдарламалық жасақтама лицензияларымен қамтамасыз етілген пайдаланушылар қосуы мүмкін.

Sublime Text қабылдайтын бағдарламалау тілдері: HTML, Java, CSS, PHP, Python, XML, SQL, TCL, Perl, C, C ++, C #, Haskell, MATLAB, Dylan, Erlang, Lisp, Lua және Markdown. Қолданушылар осы тілдерден басқа, өздеріне қажетті тілдерді қосу үшін плагиндерді жүктей алады.

Sublime мәтіні пайдаланушыға бағдарламаларды құруға және команда жолына ауысудың қажетінсіз іске қосуға мүмкіндік береді. Пайдаланушы сонымен қатар өздерінің құрастыру жүйесін реттеп, кодты сақтаған сайын автоматты бағдарламаны құруға мүмкіндік береді.

#### **3.6.2 Android Studio құралы**

Android Studio ортасы мобильді қосымшаларды әзірлеушілердің кішігірім топтарына немесе GIT немесе басқа ұқсас нұсқаларды басқару жүйелері бар ірі халықаралық ұйымдарға арналған. Тәжірибелі әзірлеушілер кең ауқымды жобаларға қолайлы құралдарды таңдай алады.

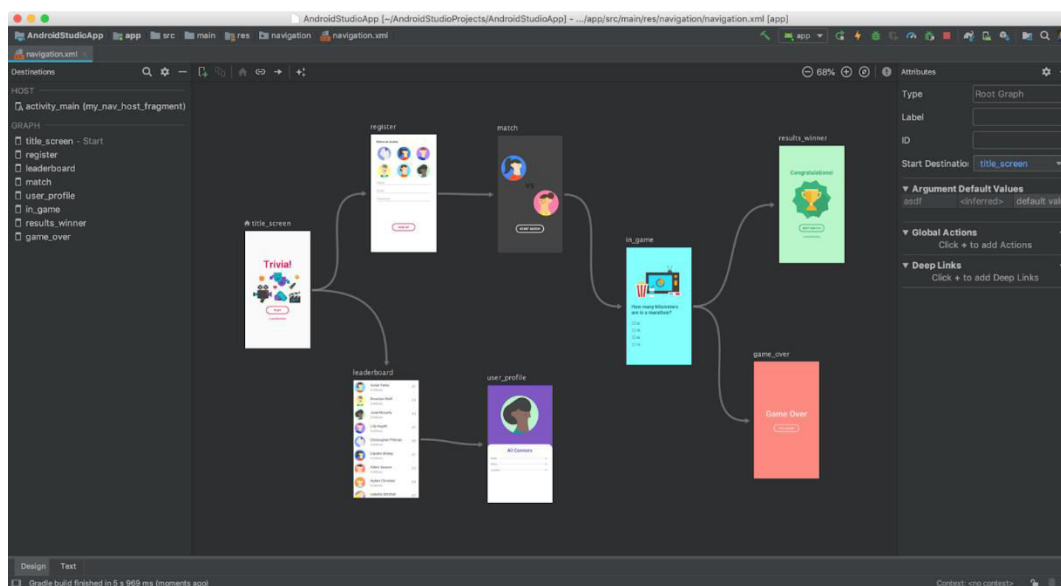
Android шешімдері Android Studio бағдарламасында Java немесе C ++ арқылы жасалған. Android Studio жұмыс үрдісінің негізі үздіксіз интеграция тұжырымдамасы болып табылады, ол сіз бар мәселелерді дереу табуға мүмкіндік береді. Ұзақ кодты тексеру әзірлеушілерге тиімді кері байланыспен

қамтамасыз етеді.

Android Studio жаңа API және функциялардың бұлтына тез біріктіру үшін Google App Engine жиынтығымен үйлесімді. Дамыту ортасында сіз Google Play, Android Pay және Health секілді түрлі API-ді таба аласыз. Android 1.6-ден бастап барлық Android платформаларына қолдау бар. Google Android нұсқасынан айтарлықтай ерекшеленетін Android нұсқалары бар. Олардың ең танымалсы - Amazon Fire OS. Android Studio бағдарламасында осы ОЖ үшін APK жасауыңызға болады. Android Studio қолдауы онлайн форумдармен шектеледі.

Android Studio негізінде жасалынған қолданбалар Activity-лардан тұрады. Activity – экранды бейнелейтін және пайдаланушылар телефон нөмірін теру, фотосурет түсіру, хат жіберу немесе картаны көру сияқты әрекеттерді орындау үшін өзара әрекеттесе алатын бағдарламаның құрамдас бөлігі болып табылады. Әрбір операцияға тиісті пайдаланушы интерфейсін салу үшін терезе тағайындалады. Әдетте терезе толық экранда көрсетіледі, бірақ оның өлшемі кішірек болуы мүмкін және оны басқа терезелердің үстіне қоюға болады. Қарапайым бағдарламалар бір Activity-дан тұрады. Күрделі қосымшаларда бірнеше терезе болуы мүмкін, яғни, олар бір-бірімен өзара әрекеттесуі мүмкін бірнеше әрекеттерден тұрады.

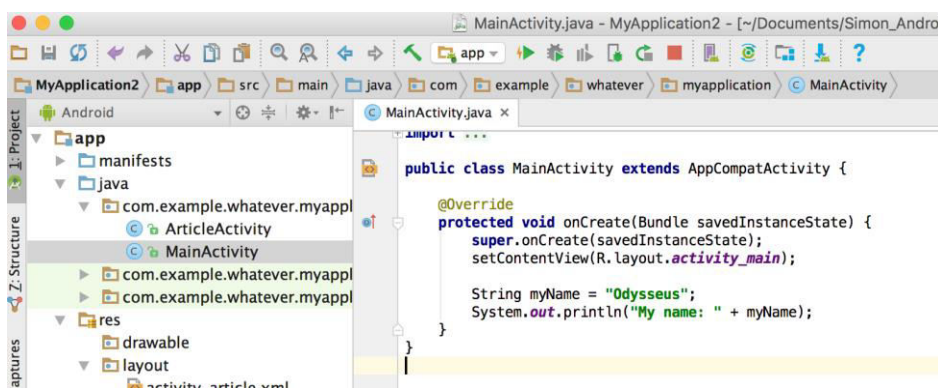
Қолданба ашылған кездегі алғашқы бет негізгі болып саналады. Одан басқа әрекетті және біздің қолданбаға ғана қатысты әрекеттерді емес, басқа да қолданбаларға байланысты әрекеттерді бастауға болады. Пайдаланушы, ол іске қосылған барлық іс-әрекеттердің бір қосымшаның бөліктері болып көрінеді, бірақ іс жүзінде әр түрлі қолданбаларда анықталуы және түрлі процестерде жұмыс істеуі мүмкін. Activity-ді сілтеме бойынша браузерлерде ашылған әртүрлі сайттардың беттері ретінде қабылдауға болады (3.4-сурет).



**3.4-сурет – Бірнеше Activity-лардың бір-бірімен байланысу принципі**



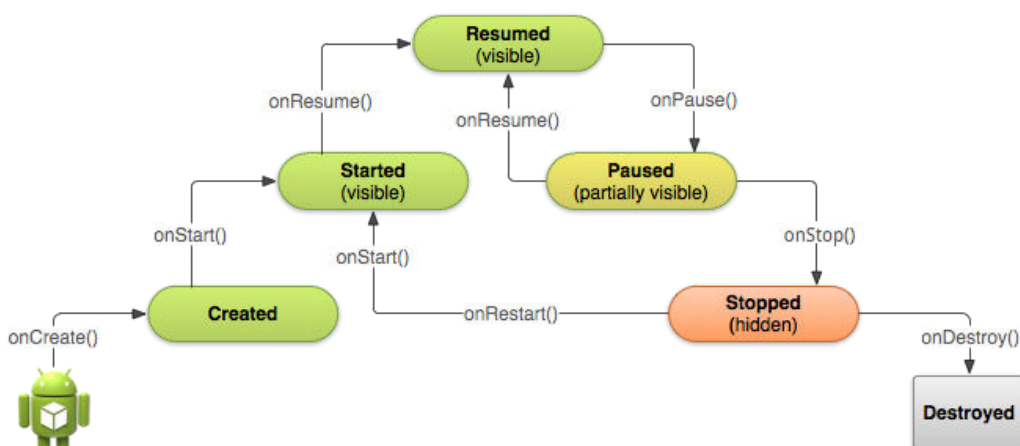
Activity-ді жасау үшін, Activity классындағы onCreate() методын шақыру керек. Нәтижесінде бос экран пайда болады. Мұндай экранның мағынасы жоқ. Сондықтан қызметте түзетулерді пайдаланатын құрамдас бөліктерді қосыңыз(3.5-сурет).



**3.5-сурет – Activity-ді жасау үшін арналған код**

Activity-дің өмірлік циклі бар – Android қызметі қызметінің данасын жасаған кезде, аралық күйде тұрған кезде және жүйе бұзылған немесе ресурстарды босатқан кездегі күйі. Қызмет үш күйде болуы мүмкін:

- белсенді (active немесе running) – белсенділік экранның алдыңғы жағында орналасады. Пайдаланушы белсенді тереземен өзара әрекеттесе алады;
- үзіліс (paused) – белсенділік фокус жоғалтты, бірақ әлі де пайдаланушыға көрінеді. Яғни, бұл қызмет жоғарыда және ішінара жабылады. Тоқтатылған қызметті жүйе жадтың жетіспеушілігі үшін тұрғысынан бұзуы мүмкін;
- тоқтатылған (stopped) – егер бұл Activity басқа Activity-мен толығымен жабылса осы күйде тұрады. Бұл пайдаланушы үшін енді көрінбейді және егер жүйе маңызды процесс үшін жады қажет болса, жүйе жойылуы мүмкін 3.5-суретте көрсетілген.



**3.5-сурет – Activity-дің өмірлік циклі**

### 3.6.3 Қолданушы интерфейсі

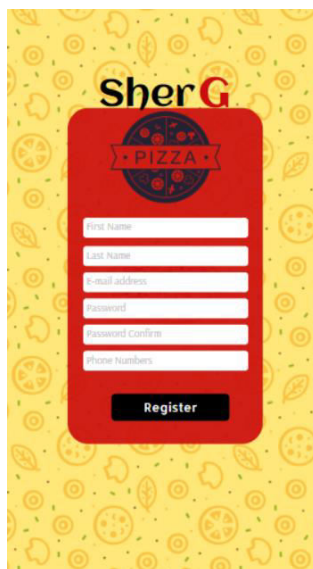
Менің дипломдық жобамның мақсаты, REST технологиясы арқылы дерек қорын мобильді қолданбаны байланыстыру болып табылады.

Мобильді қолданбаға кірген бойда сіз алдымен жүйеге тіркелуіңіз немесе кіруіңіз қажет болады. Қолданбаның алғашқы парақшасы 3.6-суретте көрсетілген.



**3.6-сурет – Қолданбаға алғаш кіргендегі терезе**

Бұл қолданбаны клиент бірінші рет қолданса оған, бірінші жүйеге тіркелу қажет болады. Сондықтан тіркелу батырмасын басқан соң келесі Activity шығады (3.7-сурет).



**3.7-сурет – Қолданбаға тіркелу арналған терезе**

«Кіру» батырмасын басып келесі бетке кірсеңіз болады.

Кірген соң сізде таңдау болады. Яғни, өзіңізге керек пиццаны әзірлеп немесе дайын пиццаны таңдауға(3.8-сурет).



**3.8-сурет – Таңдау Activity**

Егер сізге пиццерияның дайын менюінен пиццаны таңдау оңай болған жағдайда келесі Activity ашылады(3.9-сурет).



**3.9-сурет – Ресторанның дайын менюі**

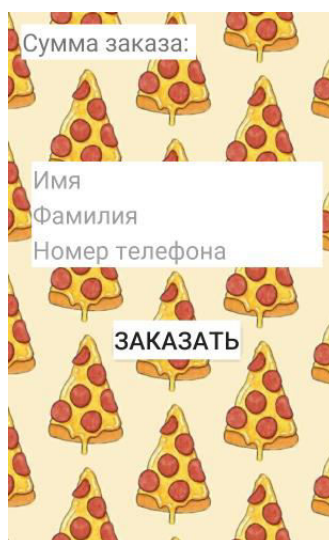
Пиццалардың тізімін «Меню» атты беттен көрсе болады. Ол бетте заманауи іздеу функциясы қарастырылған. Пиццаның атын жазып бетке жүктелген пиццерияны тауып алса болады.

Қолданушы осы баптаманың негізгі фишкасына қызығып, өзі қалаған пиццаны жасағысы келген жағдайда келесі бет ашылады.



**3.10-сурет – Ресторанның дайын менюі**

Клиент қалаған пиццаны жасаған соң, төлемге өтеді. Ал ол тапсырыс API арқылы серверге келеді. Деректер қорынан тексерістен өткен соң, ресторанның R-Keeper-іне жетеді. Яғни, ресторандағы R-Keeper-ге мына Activity-да енгізілген деректермен және жасалынған пиццаның ингредиенттері мен сомасы шығады. Деректер енгізетін Activity 3.11-суретте көрсетілген.



**3.11-сурет – Қолданушының деректер енгізетін бет**

## ҚОРЫТЫНДЫ

Тек қосымшалар жұмысқа, ойын-сауыққа, Интернетке қатынасуға, Интернетті қарап шығуға және басқаларға қолайлы кез келген операциялық жүйені жасай алады, әдеттегі телефонды толық функционалдық мүмкіндіктері бар шағын компьютерге айналдырады. Қолданбалар барлық платформалар, яғни кросс-бағдарламалар және әр жүйе үшін жеке әзірленуі мүмкін. Тәжірибе көрсеткендей, жеке бағдарламалық жасақтаманың мүмкіндіктері әлдеқайда кең, бұл тиісті бағдарламалық жасақтаманы орнату және сол себепті қажетті қосымшаларды әзірлеу.

Жоба бойынша, Android мобильді қосымшасы клиент ретінде, ал веб-бет сервер ретінде, клиент-сервер технологиясын іске асырады.

Қолданушыларға арналған сайтына кіріп, жобамен қолдана берсе болады, ал REST клиенттерге арналған атты адреспен қолдану керек. Жобада REST технологиясының барлық шарттары қарастырылған, және HTTP хаттамасы толықтай қолданылған.

Дипломдық жұмыс нәтижесінде клиенттік-серверлік қосымшаларды әзірлеу тәжірибесі алынды, бағдарлама кодының жоғарылауы өте маңызды. Ғаламтор алыптарымен, Facebook секілді сайттармен жұмыс істеу дағдылары бар. Android операциялық жүйесінде жүйелердің үшінші жақ өзара әрекеттесу қағидаттарын терең түсіну және түсіну. Кешенді және маңызды жобаларда жұмыс істей отырып, біз жұмысымыздың маңыздылығын түсінеміз және біз IT-де үлкен мәнге ие болғандықтан, біздің жұмысымыздың жоғары нәтижелерін көрсетуіміз керек.

Алдағы бірнеше жылда Android платформасы бәсекелестерді нарықтан тысқары бәсеңдетіп, өзі бесленді түрде дамиды. Және бұл өте қисынды. Дамудың қарапайымдылығы, жоғары сападағы кері байланыс, бүкіл әлемдегі көптеген әзірлеушілер, ашық бастапқы коды, нарыққа қол жеткізу мүмкіндігі – бұл мобильді есептеу технологиясының қазіргі әлемінде табысқа жетудің жолы.

Android әзірлеушілері осы платформа үшін қосымшаларды әзірлеуді қалайтын кез келген адамға мүмкіндік беріп отыр.

Болашақта қол жеткізілген жетістіктерге тоқталмай, Android операциялық жүйесімен мобильді құрылғыларға арналған қосымшаларды әзірлеуді жалғастыруды жоспарлап отырмын, себебі қазіргі уақытта ақпараттық технологияларды дамыту саласы өте перспективалы болып саналады.

## ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

- 1 UML Tutorial // Сайттың электронды нұсқасы <https://www.tutorialspoint.com/uml/>
- 2 Android Studio // Сайттың электронды нұсқасы [https://ru.wikipedia.org/wiki/Android\\_Studio](https://ru.wikipedia.org/wiki/Android_Studio)
- 3 Вагнер Р., Байк А. Энциклопедия JavaScript Киев: Bhv, 2009. – 400 с.
- 4 Дмитриева М.В. JavaScript. Быстрый старт – СПб.: БХВ – Петербург, 2002. – 150 с.
- 5 Ролов А.В., Фролов Г.В. Базы данных в Интернете: практическое руководство по созданию Web-приложений с базами данных «Русская Редакция». – СПб.: БХВ – Петербург, 2014. – 315 с.
- 6 А. Леоненков. Самоучитель UML (2-е издание). – СПб.: БХВ – Петербург, 2014. – 215 с.
- 7 Скотт Хокинс. Администрирование Web-сервера Apache и руководство по электронной коммерции. – М.: «Вильямс», 2017. – 367 с.
- 8 Горев А., Макашарипов С., Владимиров Ю. MySQL и Microsoft SQL Server для профессионалов. – СПб.: Питер, 2014. – 120 с.
- 9 Вагнер Р. Энциклопедия HTML. – Киев: Bhv, 2001. – 400 с.
- 10 Android Studio IDE от Google // Сайттың электронды нұсқасы <https://wnfx.ru/android-studio-ide-ot-google/>
- 11 Операции Activity // Сайттың электронды нұсқасы <https://developer.android.com/guide/components/activities.html>
- 12 Activity (Активность, Деятельность) // Сайттың электронды нұсқасы <http://developer.alexanderklimov.ru/android/theory/activity-theory.php>
- 13 Веймаер, Р.; Сотел, Р. Освой самостоятельно Microsoft SQL Server 2000 за 21 день (+ CD-ROM); М.: Вильямс, 2016. -549 с.
- 14 Ташков П. Веб мастеринг на 100%: HTML, CSS, JavaScript, PHP, CMS, AJAX, раскрутка. – СПб.: Питер, 2017. – 512 с. 5 Котеров Д., Костарев А. PHP.
- 15 Основы CSS // Сайттың электронды нұсқасы <https://html5book.ru/osnovy-css/>

## **А Қосымшасы** (міндетті)

### Техникалық тапсырма

#### **А.1 Кіріспе**

Бүгінгі күнде мобильді құрылғылардың жаңаша нұсқаларын көре отырып, заманның жетілгенің белгісін байқауға болады. Мобильді қатынасу құрылғыларды жоғары деңгейде жетіліп, дамуын бір ұғыммен анықтап айтуға болады, ол – технологиялардың біріктірілуі. Қазіргі күнгі мобильді құрылғылардың бір ортақ құрылғының корпусына ортақталып, біріктірілер болсақ, ол барлық мүмкіндіктер мен технологияларды біріктіретін болады.

Жаңа жобамен жұмыс істеу – ең алдымен тұжырымдамалар мен қосымшаның жаңа идеясын жасау, алдыға мақсат қолдана басталады. Ол маркетингтік стратегия мен пайдаланушылар аудиториясын таңдау, аналогты қосымшалардың бар-жоқтығын зерттеу, қосымшаның негізгі атқаратын функцияларын ойластыру.

Android операциялық жүйесіне арналған «byU» мобильді қосымшасы арқылы пайдаланушылар керек пиццаға тапсырыс беру жолында өзіңе қажеттісін мобильді қолданба арқылы жасауға болады.

##### **А.1.1 Өндеудің мақсаты мен қызметі**

Қосымша пайдаланушы мен деректер қорының спецификасында жасалынды. Қосымша Android платформасында Android Studio бағдарламасының көмегімен Java программалау тілінде жазылды.

Барлық элементтер, яғни мәтін string-те жазылады. Қосымшанның клиенттік жағына Android Studio жауап берсе серверлік жағынан мәтін string-те.

Мәтін мен суреттерді жылжыту үшін ScrollView және HorizontalScrollView компоненттерін қолдандым. Экранның болашақ макетін жасауда бағдарламадан өзімізге қажет шаблонды таңдаймыз. Activity Name – қосымшаның басты классының атауы. Layout Name – xml файлының аты, онда визуалды интерфейс элементтері сақталады.

Android шешімдері Android Studio бағдарламасында Java немесе C ++ арқылы жасалған. Android Studio жұмыс үрдісінің негізі үздіксіз интеграция тұжырымдамасы болып табылады, ол сіз бар мәселелерді дереу табуға мүмкіндік береді. Ұзақ кодты тексеру әзірлеушілерге тиімді кері байланыспен қамтамасыз етеді.



## **А Қосымшасының жалғасы**

### **А.1.2 Қолдану саласы**

Қолдану аймағы – заманауи жаңа түрде пиццаға тапсырыс беруі бойынша жасалынған қосымша.

### **А.1.3 Анықтамалар, терминдер және қысқартулар**

UML (ағыл. Unified Modeling Language – бірыңғай модельдеу тілі) – бұл графикалық сипаттама тілі Бағдарламаның нысаны сипатын әзірлеу үшін пайдаланылады. UML-модельі болып аталынатын графикалық символдарды пайдаланып жүйенің абстрактілік модельін құру. UML бағдарламалау тілі болып табылып, UML модельдері негізінде кодты генерациялауы мүмкін.

Android операциялық жүйесі смартфондарға, планшеттерге, электронды оқулықтарға, нетбуктар мен смартбуктерге арналған жүйе. Дүние жүзіне ең танымал мобильді қосымшалардың бірі. Ол-планшет, смартфондар, нетбук, электронды кітаптарда, қол сағаттарда жұмыс жасайтын Linux ядросының көмегімен жұмыс істейді.

Android – мобильді құрылғылар үшін - телефондар мен планшеттер үшін жалпы операциялық жүйе. Бұл жүйе бүкіл әлемдегі көптеген пайдаланушыларға танымал және тартымды ететін көптеген ерекшеліктері бар.

Google Play – мобильді қосымшалар дүкені Google компаниясы қосымшаларына арналған. Android операциялық жүйесінде әр түрлі мобильді қосымшалардың ақылы және де ақысыз бағдарламаларды жүктеуіне арналған. Мобильді қосымша әзірлеушілері кез-келген елдің өздерінің ақылы қосымшаларын сату арқылы табыс көзін табуға да болады.

## **А.2 Жалпы сипаттамасы**

Соңғы жылы смартфондармен планшеттердің сатылу көрсеткіші бірнеше мәртеге өскен. Осы көрсеткіштер үнемі өсуде, қазіргі уақытта бұл көрсеткіш әлі өзгермеген. Мобильді қосымшаларының өзекті және маңызды болуы айқын.

Бағдарламалық қамсыздандырулар мен мобильді қосымшаларды дайындаушыларға сұраныс өте көп.

Мобильді қосымша дегеніміз – смартфондар мен басқа да қосымша құралдарда жұмыс жасайтын программалық қамтама. Барлық мобильді қосымшалар интернет желісі арқылы интернет магазиндері арқылы жүктелініп алынады.

Менің мобильді қосымшам Android платформасында жазылған қосымшалар интернет желісі арқылы интернет магазиндері арқылы жүктелініп алынады.

## **А Қосымшасының жалғасы**

Менің мобильді қосымшам Android платформасында жазылған қосымша. Жобам қолданушылардың қалаған пиццаны қолданба арқылы жасауы әлде менюден ұнағанына тапсырыс беруге мүмкіндік береді.

### **А.2.1 Пайдаланушылық интерфейстер**

Қолданушыға арналған интерфейсте қосымшаны қосуға, онда пиццаны әзірлеуге, қысқаша ақпарат алуға болатын пайдаланушыға ыңғайлы және қарапайым етіп жасалған интерфейс құрастырылған. Қосымшаны іске қосқанда, бас мәзір шығады. Мәтін бөлімінде екі бөлімшелер бар. Олар экранның мәтін классының атауының астында бір бағанда кезекпен орналасқан. Әр бөлімшеде өзіне тақырыптық саласына сай элементтер орналасқан.

### **А.2.2 Аппараттық интерфейстер**

Құрылғыға қойылатын жалпы талаптар:

– Android платформасында жұмыс істейтін смартфон немесе Android 4.0.3 немесе одан жаңа нұсқалары.

### **А.2.3 Программалық интерфейстер**

Серверлік программалы компоненттер:

- Android Studio;
- Java;
- БД Realm;
- Интернет хостинг.
- Android операциялық жүйесі.

## **А Қосымшасының жалғасы**

### **А.2.4 Коммуникациялық интерфейстер**

Пайдаланушылардың телефондарына жүктеу үшін ғаламторға қатынау мүмкіндігі болуы керек. Мобильді құрылғыға жүктелгеннен кейін ғаламторсыз қолдана беруге болады.

### **А.2.5 Жады бойынша шектеулер**

Қосымшаны пайдалануға арналған жалпы жады бойынша талаптар, телефонда қосымшаны жүктеуге жеткілікті жады болуы керек.

### **А.2.6 Коммуникациялық интерфейстер**

Бұл мобильді қосымша тек Андроид операциялық жүйесіне орнатылған мобильдік құрылғыларда, смартфондарда, планшеттерде жұмыс істейді.

## **А.3 Функционалдық талаптар**

Функционалдық талаптар негізінен бағдарламалық қамтаманың атап айтқанда нені істейтінін анықтайды, қандай тапсырмаларды шешетінін анықтайды. Бұл мобильдік қосымшада тестілеуде, жұмысты қолданушы тұрғысынан істейміз. Тестілеу кезінде келесі амалдарды орындаймыз:

- қосымшаны іске қосу;
- барлық элементтің көрініп тұрғандарына көз жеткізу;
- мәзір бетіне өту;
- мәзірде бөлімді таңдау;
- таңдалған бөлімге өту;
- мәзірдегі барлық бөлімді тексеру;
- бөлімдердің дұрыс жұмыс істеп тұрғандығына көз жеткізу.

**Б Қосымшасы**  
(міндетті)

Бағдарлама мәтіні

```
package com.youcompany.pizza;
import android.content.Intent;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.ImageButton;
import android.widget.ImageView;
import java.util.ArrayList;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = findViewById(R.id.toolbar);
        ImageButton cheese = findViewById(R.id.cheese);
        ImageButton grib = findViewById(R.id.grib);
        ImageButton tomato = findViewById(R.id.tomato);
        ImageButton kolbasa = findViewById(R.id.kolbasa);
        FloatingActionButton fab = findViewById(R.id.fab);
        final ImageView ingred1 = findViewById(R.id.ingred1);
        final ImageView ingred2 = findViewById(R.id.ingred2);
        final ImageView ingred3 = findViewById(R.id.ingred3);
        final ImageView ingred4 = findViewById(R.id.ingred4);
        final ArrayList<Integer> summa = new ArrayList<>();
        setSupportActionBar(toolbar);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
            }
        });
        ingred1.setOnClickListener(new View.OnClickListener() {
            @Override
```

## Б Қосымшасының жалғасы

```
public void onClick(View v) {
    if(ingred1.getDrawable() != null)
    {
        ingred1.setImageDrawable(null);
    }
}
});
ingred2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(ingred2.getDrawable() != null)
        {
            ingred2.setImageDrawable(null);
        }
    }
});
ingred3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(ingred3.getDrawable() != null)
        {
            ingred3.setImageDrawable(null);
        }
    }
});
ingred4.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(ingred4.getDrawable() != null)
        {
            ingred4.setImageDrawable(null);
        }
    }
});
cheese.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        summa.add(300);
        if(ingred1.getDrawable() == null)
        {
            ingred1.setImageResource(R.drawable.pizza1);
        }
    }
});
```

## Б Қосымшасының жалғасы

```
        return;
    }
    if(ingred2.getDrawable() == null)
    {
        ingred2.setImageResource(R.drawable.pizza1);
        return;
    }
    if(ingred3.getDrawable() == null)
    {
        ingred3.setImageResource(R.drawable.pizza1);
        return;
    }
    if(ingred4.getDrawable() == null)
    {
        ingred4.setImageResource(R.drawable.pizza1);
        return;
    }
}
});

grib.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        summa.add(350);
        if(ingred1.getDrawable() == null)
        {
            ingred1.setImageResource(R.drawable.pizza2);
            return;
        }
        if(ingred2.getDrawable() == null)
        {
            ingred2.setImageResource(R.drawable.pizza2);
            return;
        }
        if(ingred3.getDrawable() == null)
        {
            ingred3.setImageResource(R.drawable.pizza2);
            return;
        }
        if(ingred4.getDrawable() == null)
        {
```

## Б Қосымшасының жалғасы

```
        ingred4.setImageResource(R.drawable.pizza2);
        return;
    }
}
});
kolbasa.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        summa.add(400);
        if(ingred1.getDrawable() == null)
        {
            ingred1.setImageResource(R.drawable.pizza3);
            return;
        }
        if(ingred2.getDrawable() == null)
        {
            ingred2.setImageResource(R.drawable.pizza3);
            return;
        }
        if(ingred3.getDrawable() == null)
        {
            ingred3.setImageResource(R.drawable.pizza3);
            return;
        }
        if(ingred4.getDrawable() == null)
        {
            ingred4.setImageResource(R.drawable.pizza3);
            return;
        }
    }
});
tomato.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        summa.add(320);
        if(ingred1.getDrawable() == null)
        {
            ingred1.setImageResource(R.drawable.pizza4);
            return;
        }
        if(ingred2.getDrawable() == null)
```



## Б Қосымшасының жалғасы

```
{
    ingred2.setImageResource(R.drawable.pizza4);
    return;
}
if(ingred3.getDrawable() == null)
{
    ingred3.setImageResource(R.drawable.pizza4);
    return;
}
if(ingred4.getDrawable() == null)
{
    ingred4.setImageResource(R.drawable.pizza4);
    return;
}
}
});
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        double summ = 0;
        for(int i = 0; i < summa.size(); i++) {
            summ += summa.get(i);
        }

        Log.i("summa", String.valueOf(summ));
        Intent intent = new Intent(MainActivity.this, JsonParser.class);
        intent.putExtra("summa", String.valueOf(summ));
        startActivity(intent);
    }
});

}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    return super.onOptionsItemSelected(item);
}}
```

## Б Қосымшасының жалғасы

```
package com.youcompany.pizza;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import org.json.JSONObject;

import java.io.BufferedOutputStream;
import java.io.BufferedWriter;
import java.io.DataOutputStream;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;

public class JsonParser extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.jsonparser);
        Button button = findViewById(R.id.button);
        TextView infoTextView = findViewById(R.id.textview);
        String summa = "0";
        summa = getIntent().getExtras().getString("summa");
        infoTextView.setText("Сумма заказа: " + summa);
        final EditText name=(EditText) findViewById(R.id.num1);
        final EditText last_name=(EditText) findViewById(R.id.num2);
        final EditText number= (EditText) findViewById(R.id.num3);
        final String value4 = "Pizza";
        Log.i("hellooo", "worldddd");
        final String finalSumma = summa;
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                {
                    OutputStream out = null;
                    String value1=name.getText().toString();
```

## Б Қосымшасының жалғасы

```
String value2=last_name.getText().toString();
String value3=number.getText().toString();
try {
    URL url = new URL("http://localhost:4455/DpiChecker");
    HttpURLConnection conn = (HttpURLConnection)
url.openConnection();
    conn.setRequestMethod("POST");
    conn.setRequestProperty("Content-Type",
"application/json;charset=UTF-8");
    conn.setRequestProperty("Accept", "application/json");
    conn.setDoOutput(true);
    conn.setDoInput(true);
    JSONObject jsonParam = new JSONObject();
    jsonParam.put("Имя", value1);
    jsonParam.put("Фамилия", value2);
    jsonParam.put("Номер", value3);
    jsonParam.put("Сумма", finalSumma);
    Toast.makeText(getApplicationContext(), String.valueOf("Ваш
заказ принят!"), Toast.LENGTH_LONG).show();
    Log.i("JSON", jsonParam.toString());
    Log.i("STATUS", String.valueOf(conn.getResponseCode()));
    Log.i("MSG", conn.getResponseMessage());
    out = new BufferedOutputStream(conn.getOutputStream());
    BufferedWriter writer = new BufferedWriter(new
OutputStreamWriter(out, "UTF-8"));
    writer.write(jsonParam.toString());
    writer.flush();
    writer.close();
    out.close();
} catch (Exception e) {
    e.printStackTrace();
}
}
});
}
}
```

## Б Қосымшасының жалғасы

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:background="@drawable/back"
  tools:context=".MainActivity">
  <android.support.design.widget.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:theme="@style/AppTheme.AppBarOverlay">
    <android.support.v7.widget.Toolbar
      android:id="@+id/toolbar"
      android:layout_width="match_parent"
      android:layout_height="?attr/actionBarSize"
      android:background="?attr/colorPrimary"
      app:popupTheme="@style/AppTheme.PopupOverlay">
      <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_gravity="center_horizontal"
        android:text="byU"
        android:textAlignment="center"
        android:textColor="#f3f3f4"
        android:textSize="40sp" />
      </android.support.v7.widget.Toolbar>
    </android.support.design.widget.AppBarLayout>
    <include layout="@layout/content_main" />
    <android.support.design.widget.FloatingActionButton
      android:id="@+id/fab"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_gravity="bottom|end"
      android:layout_margin="@dimen/fab_margin"
      app:srcCompat="@android:drawable/ic_menu_send" />
  </android.support.design.widget.CoordinatorLayout>
```

## Б Қосымшасының жалғасы

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context=".MainActivity"
    tools:showIn="@layout/activity_main"
    android:layout_gravity="center"
    >
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Выберите ингредиенты"
            android:textSize="20sp"
            android:background="#FFFFFF"
            android:scaleType="fitXY"
            android:padding="1sp"
            android:layout_gravity="center_horizontal"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintLeft_toLeftOf="parent"
            app:layout_constraintRight_toRightOf="parent"
            app:layout_constraintTop_toTopOf="parent"
            android:layout_marginBottom="20sp"/>
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal">
            <ImageButton
                android:id="@+id/cheese"
                android:layout_width="50sp"
                android:layout_height="50sp"
                android:layout_gravity="center"
                android:src="@drawable/cheese"
                android:background="#272727"
```

## Б Қосымшасының жалғасы

```
        android:scaleType="fitXY"
        android:padding="1sp"
        android:layout_marginLeft="5sp"/>
<ImageButton
    android:id="@+id/grib"
    android:layout_width="50sp"
    android:layout_height="50sp"
    android:layout_gravity="center"
    android:src="@drawable/grib"
    android:background="#272727"
    android:scaleType="fitXY"
    android:padding="1sp"
    android:layout_marginLeft="5sp"/>
<ImageButton
    android:id="@+id/kolbasa"
    android:layout_width="50sp"
    android:layout_height="50sp"
    android:layout_gravity="center"
    android:src="@drawable/kolbasa"
    android:background="#272727"
    android:scaleType="fitXY"
    android:padding="1sp"
    android:layout_marginLeft="5sp"/>
<ImageButton
    android:id="@+id/tomato"
    android:layout_width="50sp"
    android:layout_height="50sp"
    android:layout_gravity="center"
    android:src="@drawable/tomato"
    android:background="#272727"
    android:scaleType="fitXY"
    android:padding="1sp"
    android:layout_marginLeft="5sp"/>
</LinearLayout>
<RelativeLayout
    android:layout_width="300sp"
    android:layout_height="300sp"
    android:layout_marginTop="30sp"
    android:layout_gravity="center_horizontal">
    <ImageView
        android:id="@+id/pizz"
```

## Б Қосымшасының жалғасы

```
        android:layout_width="300sp"
        android:layout_height="300sp"
        android:src="@drawable/pizza"/>
<ImageView
    android:id="@+id/ingred1"
    android:layout_width="300sp"
    android:layout_height="300sp" />
<ImageView
    android:id="@+id/ingred2"
    android:layout_width="300sp"
    android:layout_height="300sp" />
<ImageView
    android:id="@+id/ingred3"
    android:layout_width="300sp"
    android:layout_height="300sp" />
<ImageView
    android:id="@+id/ingred4"
    android:layout_width="300sp"
    android:layout_height="300sp" />
</RelativeLayout>
</LinearLayout>
</android.support.constraint.ConstraintLayout>

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/backg"
    tools:context="com.youcompany.pizza.JsonParser">
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="20sp"
        android:layout_marginTop="20sp"
        android:layout_marginRight="20sp"
        android:layout_marginBottom="20sp"
        android:background="#FFFFFF"
```

## Б Қосымшасының жалғасы

```
    android:padding="1sp"
    android:scaleType="fitXY"
    android:text="Сумма заказа: "
    android:textSize="30sp" />
<EditText
    android:id="@+id/num1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textView"
    android:layout_alignParentStart="true"
    android:layout_marginStart="33dp"
    android:layout_marginTop="105dp"
    android:background="#FFFFFF"
    android:ems="10"
    android:hint="Имя"
    android:inputType="text"
    android:padding="1sp"
    android:scaleType="fitXY"
    android:textSize="30sp" />
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:inputType="text"
    android:ems="10"
    android:id="@+id/num2"
    android:hint="Фамилия"
    android:textSize="30sp"
    android:layout_below="@+id/num1"
    android:layout_alignStart="@+id/num1"
    android:layout_marginRight="20sp"
    android:background="#FFFFFF"
    android:scaleType="fitXY"
    android:padding="1sp"/>
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:inputType="number"
    android:textSize="30sp"
    android:ems="10"
    android:id="@+id/num3"
    android:hint="Номер телефона"
```



## Б Қосымшасының жалғасы

```
        android:layout_marginRight="20sp"
        android:layout_below="@+id/num2"
        android:layout_alignStart="@+id/num2"
        android:background="#FFFFFF"
        android:scaleType="fitXY"
        android:padding="1sp"/>
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/num3"
        android:layout_alignStart="@+id/num3"
        android:layout_marginStart="100dp"
        android:layout_marginTop="65dp"
        android:layout_marginRight="20sp"
        android:background="#FFFFFF"
        android:padding="1sp"
        android:scaleType="fitXY"
        android:text="Заказать"
        android:textSize="30sp" />
</RelativeLayout>
```

Б – қосымшасы дипломдық жұмысты құру кезінде қолданылады.

[illegible]