

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ  
СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты  
«Программалық инженерия» кафедрасы

Карабалаева Динара Кошгарқызы

«Find System» проектісінің мобильді қосымшасын құру

Дипломдық жобаға  
**ТҮСІНІКТЕМЕЛІК ЖАЗБА**

5B060200 – «Информатика» мамандығы

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

«Программалық инженерия» кафедрасы

5B060200 – «Информатика»

**БЕКІТЕМІН**

ПИ кафедра меңгерушісі,  
техника ғылымдарының кандидаты,  
ассистент-профессор

– Р. Юнусов  
«14» мамыр 2019 ж.

**Дипломдық жоба орындауға  
ТАПСЫРМА**

Білім алушы Карабалева Динара Кошгарқызы

Тақырыбы «Find System» проектісінің мобильді қосымшасын құру

Академиялық мәселелер жөніндегі проректорының 2018 жылғы «16»

қазанның № 1162-б бұйрығымен бекітілген

Аяқталған жобаны тапсыру мерзімі 2019 жылғы «21» мамыр

Дипломдық жобаның бастапқы берілістері Ұсынылатын дипломдық жобада

«Find System» проектісінің мобильді қосымшасын құру. Rest технологиясы  
арқылы веб бетпен байланыс орнату

Дипломдық жобада қарастырылатын мәселелер тізімі:

а) Негізгі бөлім;

б) Жобалау бөлімі;

е) Қолданылған бағдарламалық қамтамалар тізімі;

ж) Мобильді қосымшаны құру және оны іске асыру.

Сызба материалдарының тізімі (міндетті сызбалар дәл көрсетілуі тиіс)

Жобаның презентациялық 20 слайды ұсынылған.

Ұсынылатын негізгі әдебиет 17 әдебиеттер тізімінен тұрады.

Дипломдық жобаны дайындау  
КЕСТЕСІ

| Бөлімдер атауы, қарастырылатын мәселелер тізімі | Ғылыми жетекші мен кеңесшілерге көрсету мерзімдері | Ескерту |
|-------------------------------------------------|----------------------------------------------------|---------|
| Жалпы бөлім                                     | 18.02.2019                                         | ХҚОҒ    |
| Қолданылған бағдарламалық қамтама               | 10.03.2019                                         | ХҚОҒ    |
| Жобалау бөлімі                                  | 28.04.2019                                         | ХҚОҒ    |
| Дипломдық жұмыстың түсініктемелік               | 12.05.2019                                         | ХҚОҒ    |

Дипломдық жоба бөлімдерінің кеңесшілері мен  
норма бақылаушының аяқталған жобаға қойған  
қолтаңбалары

| Бөлімдер атауы       | Кеңесшілер аты, әкесінің аты, тегі (ғылыми дәрежесі, атағы) | Қол қойылған күн | Қолы         |
|----------------------|-------------------------------------------------------------|------------------|--------------|
| Норма бақылау        | Алғожаева Р.<br>Сениор-лектор                               | 16.05.2019       | Алғожаева Р. |
| Бағдарламалық бөлімі | Сман Н.О.<br>Ассистент                                      | 16.05.2019       | Сман Н.О.    |

Ғылыми жетекші \_\_\_\_\_ С. Қалдыбеков

Тапсырманы орындауға алған білім алушы \_\_\_\_\_ Д. Карабалаева

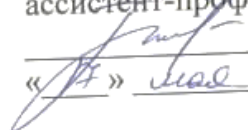
Күні \_\_\_\_\_ «16» мамыр 2019 ж.

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ  
СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты  
«Программалық инженерия» кафедрасы

**ҚОРҒАУҒА ЖІБЕРІЛДІ**

Кафедра меңгерушісі  
Техника ғылымдарының кандидаты,  
ассистент-профессор

 Р. Юнусов  
«17» маусым 2019 ж.

Дипломдық жобаға  
**ТҮСІНІКТЕМЕЛІК ЖАЗБА**

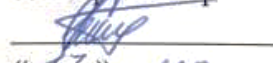
«Find System» проектісінің мобильді қосымшасын құру

5B060200 – «Информатика» мамандығы

Орындаған

Карабалаева Д.

Ғылыми жетекші  
сениор-лектор

 С. Қалдыбеков  
«17» маусым 2019 ж.

Алматы 2019

Қ.И. Сәтбаев атындағы Қазақ Ұлттық техникалық зерттеу университеті

Мамандығы 5В060200 – Информатика

Студент Карабалаева Динара

Тақырыбы: «Find System» проектісінің мобильді қосымшасын құру

### ҒЫЛЫМИ ЖЕТЕКШІНІҢ СЫН ПІКІРІ

Бұл дипломдық жобада университетімізге арналған мобильді қосымша қарастырылған. Жобаның мақсаты Satpaev University оқытушылары мен осында білім алып жатқан студенттердің оқу жүйесін автоматтандыру болып табылады. Өзіне қажетті аудиторияны іздеп, сонымен қатар аудитория жайлы ақпаратқа қол жеткізе алады. Брондау жүйесі оқытушыларға қалаған аудиториясында сабақ беру мүмкіндігін береді. Жаңалықтар бөлімі университетіміздегі соңғы жаңалықтарымен хабардар етіп отырады. Студент басқаларды қызықтыра алатындай дипломдық жоба жасаған.

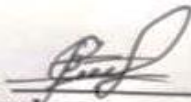
Дипломдық жұмыстың бірінші бөлімінде тақырыпқа байланысты Android платформасына арналған мобильдік бағдарламасына шолу және жасалынатын мобильдік қосымша туралы деректер жинақтау мен талдау жүргізілді. Екінші бөлімі жобалау бөлімі, модельдеу арқылы диаграммаларды қолдана отырып дипломдық жобаны жобалап алды және қолданылатын программалық қамтамалар туралы айтылған. Үшінші бөлім бағдарламалық құрылымы ерекшеліктерін қарастырған.

Студент диплом жұмысын жазу процесі кезінде материалды зерттеу талдау және жүйелеу әдістерін, безендіру тәсілдерін дұрыс пайдаланған, диплом жазушы «Тапсырма» бойынша барлық қойылған талаптарды толық орындаған.

Қорытындылай келе студент өз мүмкіндіктерін кеңінен пайдаланып, дипломдық жобаны жақсы жасап шыққан.

Жоба жетекшісі ретінде студент Карабалаева Динара Кошгарқызы «Find System мобильді қосымшасы» атты тақырыптағы дипломдық жұмысын толық аяқтап өткізді.

ҒЫЛЫМИ ЖЕТЕКШІ  
«Программалық инженерия» кафедрасының  
сениор-лекторы «17» мамыр 2019жыл



С.Б. Қалдыбеков

## Отчет подобия



|                                           |                     |
|-------------------------------------------|---------------------|
| Университет:                              | Satbayev University |
| Название:                                 | Карабаева Динара    |
| Автор:                                    | Карабаева Динара    |
| Координатор:                              | Саламат Қалдыбеков  |
| Дата отчета:                              | 2019-05-12 18:04:36 |
| Коэффициент подобия № 1:                  | 5,0%                |
| Коэффициент подобия № 2:                  | 0,0%                |
| Длина фразы для коэффициента подобия № 2: | 25                  |
| Количество слов:                          | 4 200               |
| Число знаков:                             | 34 886              |
| Адреса пропущенные при проверке:          |                     |
| Количество завершенных проверок:          | 91                  |

Ғылыми жетекші

Қалдыбеков С.Б

## АҢДАТПА

Берілген дипломдық жобада Android операциялық жүйесінде жасалынған «Find System» бағдарламасы университетіміз туралы нақты мәліметтер беруге арналған навигациялық гид түрінде жасалынды.

Бұл жұмыс кіріспеден, негізгі бөлігін қамтитын үш тараудан (талдау бөлімі, зерттеу бөлімі және жобалау бөлімі) қорытынды, пайдаланылған әдебиеттер тізімінен және қосымшадан тұрады.

Бірінші бөлімде Android платформасына арналған мобильдік бағдарламасына шолу және жасалынатын мобильдік қосымша туралы деректер жинақтау мен талдау жіргізу сипатталады.

Екінші бөлімде мобильдік қосымшаны қолданудағы модельдері, мәліметтер базасын таңдау, сонымен қатар программалау тілін таңдау негіздемесі айқындалады.

Үшінші бөлімде бағдарламалау ортасын баптау мен әзірлеу барысы сипатталып, бағдарлама жайлы жалпы мәліметтер мен логикалық құрылымы, сонымен қатар техникалық талаптар тізімі мен жұмыс нәтижесі сипатталады.

Бұл жұмыс мәліметтерді MySQL деректер қорын басқару жүйесі арқылы басқарылады. Интерфейс Java бағдарламалау тілінде жасалған. Жұмыстың ішінен жобаның модельдік үлгісін бейнелейтін диаграммаларды және жүйенің толық сипаттамасын табуға болады.

## АННОТАЦИЯ

В данном дипломном проекте разработана программа «Find System», разработанная в операционной системе Android, в виде навигационного гида для передачи точных данных о нашем университете.

Эта работа состоит из введения, трех разделов, включающих основную часть (аналитическая часть, исследовательская часть и проектная часть), заключения, списка использованной литературы и приложения.

В первом разделе описывается обзор мобильной программы для платформы Android и проведение сбора и анализа данных о создаваемом мобильном приложении.

Во втором разделе определяются модели использования мобильного приложения, выбор базы данных, а также обоснование выбора языка программирования.

В третьем разделе описывается процесс разработки и настройки среды программирования, описывается Общая информация о программе и логическая структура, а также перечень технических требований и результаты работы.

Эта работа управляется системой управления базами данных MySQL. Интерфейс разработан на языке программирования Java. В работе можно найти диаграммы, отражающие модельную модель проекта и подробное описание системы.

## ANNOTATION

In this graduation project, the program "Find System", developed in the operating system Android, in the form of a navigation guide to transmit accurate data about our University.

This work consists of an introduction, three sections including the main part (analytical part, research part and project part), conclusion, list of references and Appendix.

The first section provides an overview of the mobile application for the Android platform and how to collect and analyze data about the mobile application being created.

The second section defines the models of using the mobile application, the choice of database, as well as the rationale for the choice of programming language.

The third section describes the process of development and configuration of the programming environment, describes General information about the program and the logical structure, as well as a list of technical requirements and results.

This work is managed by the MySQL database management system. The interface is developed in the Java programming language. In the work you can find diagrams that reflect the model model of the project and a detailed description of the system.

## МАЗМҰНЫ

|     |                                                      |    |
|-----|------------------------------------------------------|----|
|     | Кіріспе                                              | 8  |
| 1   | Негізгі бөлім                                        | 9  |
| 1.1 | Жобаның өзектілігі                                   | 9  |
| 1.2 | Оқу жүйесіндегі аудиторияны брондау маңызы           | 9  |
| 1.3 | Мобильді құрылғыларға қосымшаларды құрудың негіздері | 10 |
| 1.4 | Ұқсас бағдарламаға шолу                              | 11 |
| 1.5 | REST API                                             | 12 |
| 2   | Жобалау бөлімі                                       | 14 |
| 2.1 | Унифицирленген модельдеу тілі                        | 14 |
| 2.2 | Java бағдарламалау тілі                              | 17 |
| 2.3 | Android платформасы                                  | 18 |
| 2.4 | Деректер құрылымы                                    | 19 |
| 2.5 | Фреймворк ұғымы                                      | 21 |
| 3   | Жоба құрылымы                                        | 23 |
| 3.1 | «Find System» проектісі туралы                       | 23 |
| 3.2 | Қолданушының интерфейсі                              | 23 |
|     | Қорытынды                                            | 27 |
|     | Пайдаланылған әдебиеттер тізімі                      | 28 |
|     | А қосымшасы. Техникалық тапсырма                     | 29 |
|     | Б қосымшасы. Бағдарлама мәтіні                       | 33 |
|     | Спецификация беті                                    | 42 |

## КІРІСПЕ

Қазіргі уақытта заман мен техниканың, әртүрлі ақпарат көздері мен ақпарат тарататын құрылғылардың дамыған кезеңі болып табылады. Солардың бірі - мобильді құрылғылар, яғни қосымшалар. Мобильді қосымша дегеніміз - мобильді телефондар, коммуникаторлар, смартфондар және тағы да басқа құрылғылар үшін арнайы жасалынатын бағдарламалық қамтама. Мобильді қосымшалар негізінде жалпы қандай да бір құрылғының тиімділігін арттыру және ақпарат беру мақсатында жасалынған қамтама саналады. Android операциялық жүйесі мобильді құрылғыларға қосымшалар құруға арналған платформа болып табылады. Оның ашықтығы мен қол жетімділігінің арқасында, жетілдіруге арналған тегін әрі ыңғайлы құралдардың бар болуының арқасында көп қолданысқа ие болып отыр. Сол себепті «Find System» аудитория іздеу мобильді қосымшасы Android платформасында жасалынды.

Қосымшаның қолдану аймағы – қазақ тілінде білім алатын студенттерге, оқытушыларға және абитуриенттерге университетіміз туралы өзіне керекті ақпарат алуға бағытталған. Сонымен қатар орыс және шет тілдерінің қазақ тілімен қатар жүруіне байланысты бұл екі тілде бағдарлама жасалып, енгізілуі мүмкін.

Дипломдық жұмыстың мақсаты: Университетімізде білім беретін оқытушыларға өзіне ыңғайлы аудитория орнын табу және қажет жағдайда аудиторияны брондау мүмкіндігінің болуы. Ал студенттерге аудитория туралы ақпарат алуға септігін тигізеді. Университетіміздің өте үлкен болуына байланысты бұл қосымша көптеген студенттер мен оқытушыларға септігін тигізеді деген үміттеміз.

Дипломдық жоба өзектілігі: Қазіргі таңда сенсорлы экранды ұялы құрылғылар, планшеттер және тағы басқа құрылғыларға арналған мобильдік қосымшалар кең ауқымды қолданысқа ие болғандықтан мобильді қосымшалар құру өзекті мәселе болып табылады және қолданушылар өздеріне ыңғайлы әрі қолжетімді және тегін қызмет түрлерін қолдануды жөн көреді. Білім беру салысында да мобильді қосымшаның бағдарламасын құру жеке бір өзекті мәселе болып отыр. Сонымен қатар қазақ тілінде жазылған қосымшалардың аздығы, ұлттық тілімізде жазылған қосымшаларды жасауға деген қажеттілікті тудыруда. Іздеу жүйесі барлық университетіміздің адамдарына таптырмас бағдарлама болып табылады.

## **1 Негізгі бөлім**

### **1.1 Жобаның өзектілігі**

Қазіргі уақытта мобильді құрылғылардың үлесі үнемі артып келеді, сондықтан бейімделген (реттелген) дизайнды немесе шағын экрандарды пайдаланатын сайтқа келушілер үшін жеке мобильді нұсқаны пайдалана отырып, мобильді құрылғыларға веб-сайттарды оңтайландыру қажеттілігі туындайды.

Find System мобильді қосымшасы көп жағдайда талапкерлер мен студенттерге арналған. Олар енді мобильді құрылғылардан ЖОО туралы қажетті анықтамалық мәліметтерді ала алады, университет жаңалықтарын қадағалайды, сондай-ақ фотосуреттер мен бейнероликтердің бай мұрағатын қарастыра алады. Сонымен қатар оқытушыларға келетін болсақ, оларға аудиторияны брондау жүйесі қолжетімді. Бұл бағдарлама тек Қ.И.Сатпаев университетіне арналған жоба болып саналады.

Бұл дипломдық жобада веб-парақша және мобильді қосымша қатар жүреді. Веб бет сервер бөлімі болса, мобильді қосымша клиент бөліміне жауап береді және жекелей қосымша бола алады. Жобаның тағы бір ерекшелігі, веб-бетпен Rest технологиясы арқылы байланыс жасай алауында.

### **1.2 Оқу жүйесіндегі аудиторияны брондау маңызы**

Білім беру сапасының және оқу пәндерін табысты оқытудың маңызды нысаны-бұл кафедрада, факультетте, ЖОО-да оқу-материалдық базасын құру болып табылады. Ол техникалық ЖОО студенттерінің өз бетінше жұмыс істеуге арналған теориялық және практикалық дайындығының заманауи нормаларына жауап беруі тиіс. Сондықтан бірінші кезекте оқу аудиториясын, тиісті оқу пәні бойынша зертхананы және дәріс кабинетін жабдықтау қажет. Аудиторияны жабдықтау өндірістік жағдайлардың көрнекілігіне, қол жетімділігіне және ғылыми мазмұндылығының негізгі дидактикалық принциптеріне жауап беруі тиіс.

Аудитория практикалық, зертханалық және зертханалық-тәжірибелік сабақтарды өткізуге, сонымен қатар үйірме жұмыстарына, студенттердің өзіндік жұмысына арналған. Оқу аудиториясын жинақтау және жабдықтау ғылыми-техникалық прогресс, алдыңғы қатарлы шаруашылықтар тәжірибесі, ғылыми және оқу әдебиеті негізінде тиісті әдістемелік құралдар мен әзірлемелерде қамтылған ұсыныстарды ескере отырып, білім беру бағдарламалары мен кәсіптік білім беру стандарттары департаменті бекіткен үлгілік пән бағдарламасына сәйкес жүргізілуі тиіс. Бұл ретте оқу сапасы жүйесінде ЖОО алдына қойылған оқыту мен тәрбиелеудің нақты міндеттерін ескеру қажет. Алдыңғы ғылыми ақпараттың сипаты мен көлемін жеткілікті дәлдікпен анықтау, соның негізінде

студенттер білімді, іскерлікті және дағдыларды меңгереді, әрбір тақырыптың мазмұнын, әр мәселенің мазмұнын ашудың логикалық түрін қалыптастырады.

Оқу аудиториясы жабдықтарының ерекшелігі көрнекі баспа құралдарының (схемалар, фотоальбомдар, фото стендтер, плакаттар) және де техникалық құралдардың (кинопроекторлар, диапроекторлар, технологиялық процестерді бейнетехника - түсіру, дербес компьютерлер) болуы.

Оқу аудиториясын ұйымдастыру мәселелерін шешудегі басты рөл пәнді оқытушыға беріледі. Сол себепті оқытушылар тиісті білім беру үшін аудиторияның толық жабдықталғаны маңызды. Жобаның басты мақсаты-оқытушыға көмек ретінде қалаған аудиторияны брондау жүйесі болып саналады. Кей кездерде студенттер саны мен сабақ кестесі бойынша қойылған

Аудитория саны сәйкес келмей, көп қиындық туғызып жатады. Бұл жағдайда студентке және оқытушыға да қолайсыз болады. Сол себепті аудиторияны брондау жүйесі қазіргі уақыттың жаңалығы болып саналады. Себебі оқытушы өзіне қолайсыз аудиторияны ауыстыруға немесе қосымша сабақ өтуге аудиторияны іздеп брондай алады.

Аудиторияны брондау – бұл университеттің бастапқы оқу ғимаратында қажетті аудитория туралы мәлімет алып(техникалық құралдар), оны брондау жүйесі. Брондау тек оқытушыларға арналған. Ал студенттер тек аудитория туралы мәлімет ала алады. Брондау үшін оқытушыға қажетті мәліметтер жиынтығын толтыру керек.

### **1.3 Мобильді құрылғыларға қосымша құрудың негіздері**

Мобильді қосымша – ұялы телефондарға, смартфондарға, планшеттерге, коммуникаторларға және де т.б құрылғыларға арналған бағдарламалық жабдық.

Мобильді қосымшалардың өзіне арналған әр платформасы әзірлеушіге бағдарламалауды тестілеуге және қосымша қамтамаларды енгізуге рұқсат ететін қызметі белгілі.

Android-мобильді құрылғыларға арналған жаңа платформа. Оның ашықтығы арқасында ол тез кеңейтіледі және оның ерекшелігі тегін және оңай пайдалану құралдарының болуы. Ол өте ықшамды операциялық жүйе болып табылады. Android Inc серіктестігімен бастапқысында өндіріле бастаған операциялы жүйе, артымен оны Google компаниясымен сатып алынып, ары карай дамытылды. Программалық қамтаманы дамыту және жетілдіру үшін Google компаниясы Open Handset Alliance одағын құрып соған жүктеді.

Android операциялық жүйесінде бағдарлама негізінен Java-да негізделген. Кірістірілген бағдарламалық қамтамасыз ету коды Android Page архивінің файлына оралған. Бұл файл кеңейтілімі \* .apk және арнайы aar tool құралы арқылы компилденген. Бұл жинақтаушы мұрағат-файл мобильді құрылғыларға арналған бағдарлама ретінде қарастырылады. Бір файл бағдарламаның бір кодымен байланысты. Жалпы, әрбір бағдарлама өз процесін орындайды.

Бағдарламамен жұмыс жасағанда activity ұғымымен танысамыз. Activity Android қосымшасында құрылатын ең маңызды сынып ретінде жүзеге асады. Бұл класс қосымшаның визуалды белсенділігін білдіреді және пайдаланушы жасай алатын әрекеттерді анықтайды. Java-қосымшасында Activity ата-аналық класынан кеңейтілген (extends) кем дегенде бір сынып болуы тиіс. Сондықтан Activity сыныбының әдістерін білу және қолдана алу маңызды.

#### 1.4 Ұқсас бағдарламаға шолу

Қазіргі уақытта ақпараттық технологиялардың дамуына байланысты веб-технологиялармен қатар мобильді қосымшалар өте қарқынды дамуда. Веб-бетке қарағанда мобильді қосымшалар арқылы ақпарат алу өте тиімді. Ал университетіміз туралы ақпарат алуға және жаңалықтарды білу үшін мобильді қосымша таптырмас құрал. Себебі бағдарлама студенттер, мұғалімдер және қонақтарға тек ақпарат беру ғана емес сонымен қатар аудиторияны тауып және ол туралы мәлімет беру арқылы навигация қызметін де атқарады. Бұндай ЖОО арналған веб-беттер көп болғанымен, мобильді қосымшалар аз. Солардың бірі Ресей мемлекетінің оқу орынына арналған «Найти в Огу» мобильді қосымшасы.

«Найти в Огу» қосымшасы өз университеттері туралы нақты бағдар беру үшін жасалған. Аталған мобильдік бағдарламаға төменде жеке жеке сипаттама берілген.

Бағдарламаның мүмкіндіктері:

- Аудиторияны табу. Бұл бөлімде университеттің үлкен болуына байланысты аудиторияға қалай оңай жету жолу көрсетілген. Ең алдымен корпусты тандайды, содан соң карта арқылы сол жолға сілтеме шығады.
- Оқу ғимараттары. Оқу орнының ғимараттары орналасқан жерлер туралы ақпарат береді.
- Поликлиника. Студенттердің денсаулығы үшін бұл бөлім қосылған.
- Ақпарат алу арқылы студенттерге оңтайлы уақытта барып тексерілуі үшін жасалған.
- Басқа бөлімдер. Бұл жерде асхана, дәретханалар және кітапханалар көрсетілген.

Бұл қосымшаның ерекшеліктерінің бірі тек оқу ғимараттарын ғана негізге алмай, сонымен қатар поликлиника, жатақхана және мемлекеттері туралы ақпарат порталдарын қамтиды. Маршрутқа келетін болсақ, кабинеттер арасындағы байланысты өте нақты және ыңғайлы түрде жасалынған. Дизайн беті өте әсерлі және қарапайым болуы қолдануға өте ыңғайлы.

Бұл қосымшамен Find System дипломдық жобасын салыстыратын болсақ. Қосымша көрші мемлекет университетінің картасы болып табылады. Ал «Find System» қосымшасында университет туралы ақпаратты, соңғы жаңалықтарды және аудитория сабақ кестесін қарауға болады. Сонымен қатар қосымша қазақ тілінде университетімізге арналып жазылған. Басты артықшылығы «Find

System» жобасында аудиторияны брондау жүйесінің болуында. Брондау жүйесі оқытушыларға керекті аудиторияны керек сағатында брондауға мүмкіндік береді және аудитория кестесін көру қажетті аудиторияны бос уақытта иемденуге студенттерге тиімді болу үшін жасалған.

Бағдарламаның көрінісі төмендегі 1.1- суретте көрселіген.



**1.1-сурет – «Найти в Огу» қосымшасының көрінісі**

## **1.5 REST API**

Мобильді қосымша клиент сервер бөлімінен тұратындықтан, REST технологиясын қолдану арқылы жұмыс жасалынады.

RESt - клиент-сервер моделі бойынша желіде бөлінген қосымша компоненттердің өзара әрекеттесуінің ерекше бір архитектуралық стилі ретінде саналады.

Архитектуралық стиль ерекшеліктеріне:

- әрбір мәнде бірегей идентификатор – URI болуы керек.
- мәндер өзара байланысты болуы тиіс.
- деректерді оқу және өзгерту үшін стандартты әдістер қолданылуы тиіс.
- ресурстардың бірнеше түрлерін қолдау болуы тиіс.
- өзара іс-қимыл жағдайсыз жүзеге асырылуы тиіс.

REST API-белгілі бір көлемдегі деректерді қайтаратын стандартты әдістердің қашықтағы шақырулардың жиынтығы.

REST API -ды пайдалану, әлбетте, Интернетті пайдалануға байланысты болып табылады. Дегенмен, Android бағдарламалары тек android рұқсаты бар

болған жағдайда ғана Интернетке қол жеткізе алады. Ол үшін бағдарламалық кодта android.permission.INTERNET жолы болуға тиісті.

Rest бағдарламасының артықшылықтары:

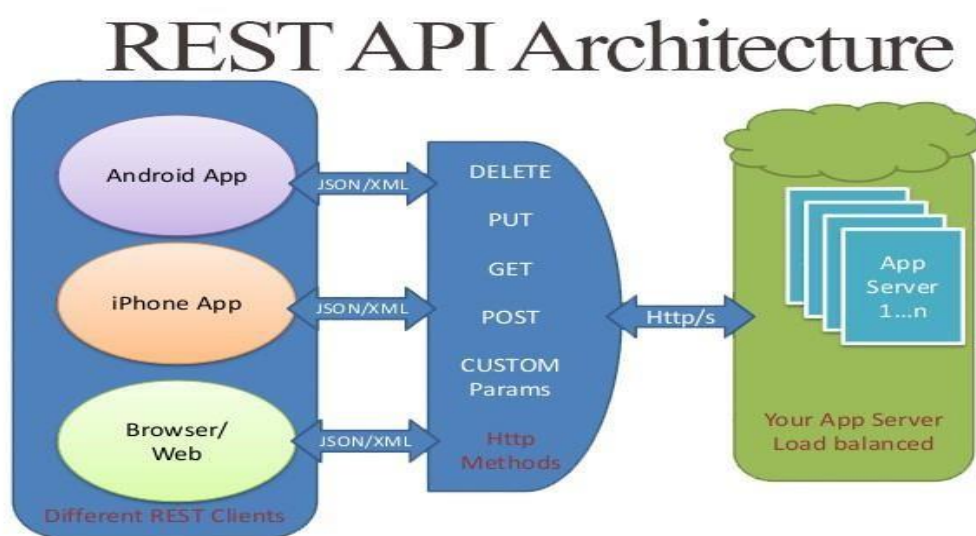
- Rest компоненттердің өзара әрекеттесуі үшін бірыңғай интерфейсті ұсынады;(веб-бет және мобильді қосымша)

- REST-деңгейден тұратын жүйе. Жекелеген компоненттер қазіргі уақытта өзара әрекеттесетін деңгеймен шектеледі;

- REST тамаша, өйткені ол жағдайды бақыламайды (stateless) Егерде сіз деректермен жұмыс жасасаңыз, ол оның жолын бақыламайды;

- Rest арқылы сіз HTTP кәштеу қуатын басқара аласыз!(яғни сұраныстармен жұмыс жасай аласыз.)

Rest API архитектурасымен төмендегі 1.2-суретте таныса аласыздар.



**1.2 - сурет – REST API архитектурасы**

Rest көбінесе веб-бетпен жұмыс жасауға ыңғайлы. Ал мобильді қосымшаға келетін болсақ, бірқатар заңдылықтар арқылы жұмыс жасайды. REST серверден алынған мәліметтер әдетте sql деректер қорында сақталады. Себебі олар activity-ға тікелей берілмейді және де деректер қорында қосу, өшіру, енгізу операциялары орындалатын болса, онда Rest сервері екі рет орындалады. Бірі операцияны жіберсе, екіншіде соған жауап қайтады.

## **2 Жобалау бөлімі**

### **2.1 Унифицирленген модельдеу тілі**

Жобамен жұмыс жасаған кезде оған міндетті түрде анализ және модельдеу мен қатар модельдеу процесін бейнелеуді жатқызады. Модельдеу тілі бағдарламалық қамтамасыз етудің сызбасын құрастырудың стандартты құралы болып табылады.

UML – Unified Modeling Language сөзінің мағынасы болып саналады. Қарапайым тілде айтқанда, UML модельдеу мен құжаттау үшін бағдарламалық қамтамасыз етуге арналған қазіргі заманауи қоғамның тәсілі. Шындығына келгенде, UML модельдеу тілі арқылы бизнес процестерді модельдеу қазіргі таңдадағы ең танымал әдістердің бірі. Бұл модельдеу тілі өте күрделі және аса ауқымды жүйелерді модельдеуді бұрыннан пайдаланылатын озық технологиялардың үйлесімін көрсетеді.

Ол бағдарламалық қамтаманың компоненттерінің схемалық көрінісін көрсетуге негізделген. Визуальды түсініктерді пайдалана және бейнелерді көре отырып, біз бағдарламалық қамтамасыз етудегі немесе бизнес – үдерістердегі мүмкін болатын кемшіліктерді немесе қателерді жақсы түсіне аламыз. Сол себепті ең алдымен жобаны модельдеуіміз қажет. Мобельдердің құрастырушы бөліктеріне класстар, интерфейстер, түйіндер, компоненттер және болмыстар жатады. Бұл бөлімдер құрылатын бағдарламаның құрылымын анық көрсетуге, оның қалай жұмсы жасайтынын көрсетуге және жасалған өзгерістерді диаграмма түрінде сызуға көмек береді. Осы кезде UML тілінің диаграммалары түрлерімен танысамыз. Әр диаграмма әртүрлі қызмет атқаратындықтан, бағдарламаға қажетті диаграмма түрін таңдап алған жөн. Жобамен жұмыс жасау үшін Rational Rose модельдеуге арналған бағдарламалық ортасы алынды. Себебі бұл бағдарлама көмегімен бизнесті үлгілеуден тестілеуге және дайын бағдарламаны пайдалануды тапсыруға дейінгі аралықты қамти алады. Басты ерекшелегі объектілі бағытталған графикалық сызбаларды UML тілінің қарапайым етіліп көрсетіледі. Rational Rose модельдеу құралдары бағдарламаны дайындаушыларға иерархияның әртүрлі деңгейлерінде бірге іс-қимыл мен бақылаудың барлық әсерлерін сақтай отырып, бағдарламалық процестердің тұтас архитектурасын жасауға мүмкіндіктер береді. Басқа артықшылығына модельдерді басқарудың оңайлығы жатады. Барлық модельдер оңай өзгертіледі және модельдер бір бірімен байланысты болса ол өзгерістер автоматты түрде өзгертіліп көрсетіледі. Нұскалар және конфигурацияны басқару жүйесі модельдерді басқаруға қолданылады. Бұл кез келген бизнес-модельге оңай өзгерістер енгізуге мүмкіндіктер береді.

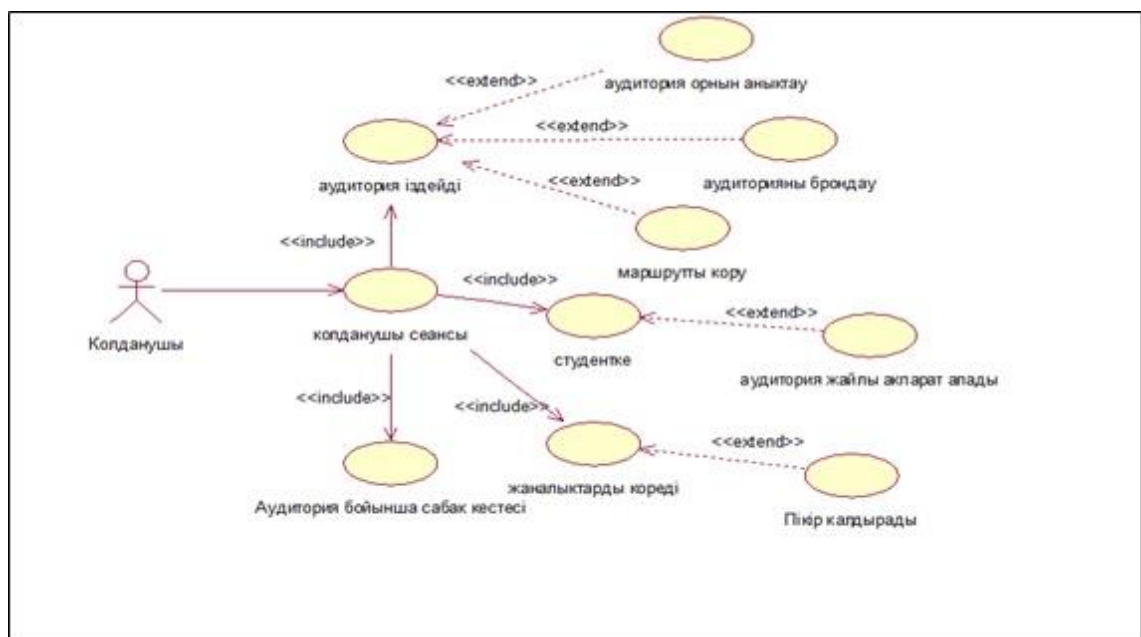
Дипломдық жұмыста құрылатын жүйенің моделі бірнеше сызбада қарастырылады. Әр сызбада бағдарламаның әртүрлі әрекеттерін көрсетеді.

Прецеденттер диаграммасы. Бағдарламаны құруға арналған заттар облысын толықтай біліп танысқан жөн. Сонымен қатар жүйенің қалай жұмыс

жасайтыны туралы ақпараттар толықтай болу қажет. Диаграмма негізінен бағдарламалық жүйеде, яғни графикада орындайтын операциялардың тізімдерін құру үшін тағайындалған. Ол берілген методтағы жүйе объектісі болып анықталған тізім жүйесін табжылмай міндетті түрде орындайды. Осындай түрде жасалатын жүйенің функция құрылымы құрылады, сондай-ақ жүйемен байланысқан объектілердің барлығының іс-әрекетінің жоспарларын жазады. Прецедент диаграммасы процесін модельдеу үшін пайдаланушы іс-әрекетін толықтай көрсетуіне, сонымен қоса толық реакциясына да бағытталған диаграмма түрі.

Бағдарламадағы қосымша жүйенің жасайтын 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 аталатын принцип негізінде жасалады. Орналасқан жүйеден басқа жерде болуы мүмкін пайдаланушы зерттеуінен, сонымен қоса ақпараттық қимыл бөліктерін ашу жолынан басталады. Осындай тұрыптан ақпараттардың құрылымы пайда болады, яғни қажет ететін іс-әрекеттің ұсынуынан болады. Ішкі бағыттың жылжуы, сонымен қатар оның қадамы бөлек бір жоспары болып табылады.

Жүйенің мінез құлқын модельдеу (Use Case Diagram) – бұл шамаланған жүйені жүргізудің құжаттық моделі. Диаграмма прецедент элементтері, актерлері мен олардың қатынастар жиынынан тұрады. Прецеденттер диаграммалары көмегімен жүйе үшін прецеденттің статикалық көрінісі құрылады. Бұл ОЖ-нің талаптарын алдын ала модельдеу және ұйымдастыру болып табылады. Қандай да бір қолданушыны немесе субъектіні бағдарламада белгілеу үшін актерді пайдаланамыз. Модельдің бұл түрі 2.1-суретте көрсетілген.

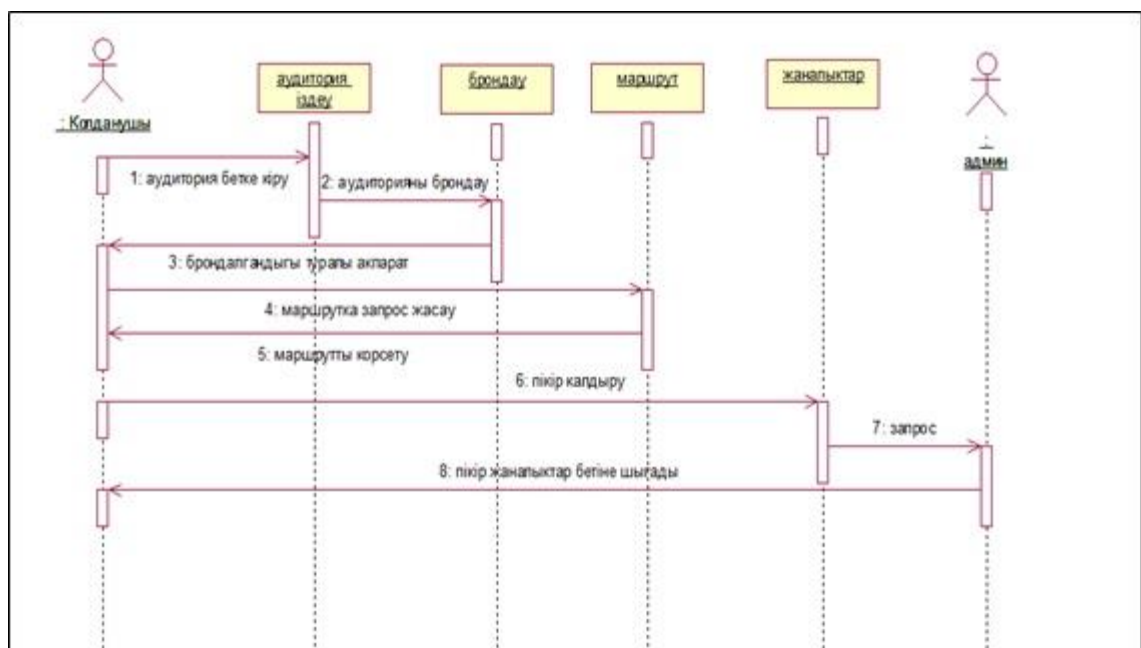


2.1-сурет – Прецеденттер диаграммасының көрінісі

Тізбек диаграммасы негізінен хабарламаларды уақыт бойынша тәртіпке келтіруді айқындайтын өзара әрекеттесу диаграммасы, яғни бір уақыт мезетіндегі іс әрекетті жобалау процесі. Негізінен ең көп қолданылатын өзара әрекеттесу диаграммасы болып табылады, себебі объектілер арасындағы байланысты белгілі бір ретпен орналастырылады. Тізбек диаграммасы негізгі төрт элементтен тұрады:

- мәтіннің іс қимылы. Сол жақ жоғарыдан төмен жазылады.
- объектітер. Объектінің аты және объектінің класс номері жазылуы мүмкін.
- операциялар. Үзік сызық түрінде жасалынады. Тікбұрыш түрінде беріледі. Яғни, әдістерге кіретін сол объектілер кіреді. Тікбұрыш ұзындығы ізбасарда басқару фокусын көрсетуде қолдануға болады.

Тізбек диаграммасының сызбасы төмендегі 2.2-суретте көрсетілген.



2.2-сурет – Тізбек диаграммасының көрінісі

Күй(State)-бұл кейбір объектілерді орындау кезінде белгілі бір әрекет немесе сол не басқа оқиғаның түсуін күтуде мекендеуді жүзеге асыратын жиын шарты. Объектінің күй класы бір немесе бірнеше атрибуттарымен сипатталады. Бұл диаграмма объектінің бір жағдайдан басқасына ауысуын орындайтын және әрекеттері жағдайларын алмастырумен ескертілген оқиғалардың немесе хабарлаулардың объектілік жағдайларының графикалық ұсынысы. Объектінің өз күйін уақыт бойында қалай өзгертетінін түсіну үшін күй өзгерісінің спецификациясы бар. Объектінің күйі оның атрибуттарының және оған қоса қатынас атрибуттарының мәнімен анықталады. Күй спецификациясы класс атрибутын анықтайды. Тәртіп спецификациясы класс операцияларын анықтайды, олардың кейбірі объект күйін өзгерте алады.

## 2.2 Java бағдарламалау тілі

Андроид операциялық жүйесі негізінен Джава бағдарламасын негізге алып жұмыс істейді. Программа барысында жазылған кодтың барлығы арнайы Android Package есімді файлда сақталады. Сол себепті бағдарлама жазуда Джава бағдарламасының орны ерекше.

Java бағдарламалау тілін Джеймс Гослинг және басқа да Сан Микросистем компаниясының инженерлері ойлап тапқан. Бұл бағдарлама мың тоғыз жүз тоқсан бірінші жылы ұйымдастырылған, алайда жарыққа тек мың тоғыз жүз тоқсан бірінші жылы шыққан. Java алғашында C++ бағдарламалау тілінің орнына құрастырылған болатын және оның атын Дуб деп қоймақшы болған. Бағдарламаны құрудағы ең негізі мақсаттардың бірі ақпаратты қашықтыққа кодтау қауіпсіз болу үшін және еркін тұтас платформа құру үшін арналған.

Java бағдарламау тілі объектіге бағытталған бағдарлама тілі болып саналады. Java тілі Android операциялық жүйесіне арналған мобильді қосымшаларды жасау үшін белсенді қолданылады. Бұл ретте бағдарламалар Dalvik виртуалды машинасын пайдалану үшін стандартты емес байтты-кодқа айналдырады. Мұндай компиляция үшін Google компаниясы әзірлеген Android SDK (Software Development Kit) қосымша құралы қолданылады.

Android бағдарламасына арналған әр қосымша бөлек үрдісте өзінің меншікті данасындағы Dalvik машинасында жұмыс жасайды. Бүкіл есте сақтау қабілеттілігі мен үрдістерді басқару жауапкершілігі Android-қа жүктеледі.

Бірінші кезеңде, Java тілінде жазылған бағдарлама компилятор арқылы байтты кодқа айналдырады. Бұл жасалынатын компиляция бір нақты процессордың және әлдебір нақты компьютердің архитектура үлгісінен тәуелді болмайды. Ол бағдарламаны жазып болған соң бір рет орындалады.

Байт кодтар бір немесе бірнеше файлда жазылып, сыртқы жадқа сақталып немесе желі арқылы жіберіле алады. Бұл файлдың шағын өлшемді байт-кодтар үшін өте ыңғайлы болып келеді. Компиляция нәтижесінде, яғни бағдарлама нәтижесінде алынған байт кодтарды кез келген JVM-ді жүзеге асыратын жүйесі бар компьютерлерде орындауға болады.

Байт кодтардың интерпретациясы және үдемелі процесс бірталай бағдарламаның орындалуын баяулатады. Байтты кодтарды желі арқылы жібергенде ескерілмеді. Сол себептен ылғи интерпретаторды жетілдіру жұмысы жүріп жатады.

Java – бағдарламада класс спецификациясы мен оның жүзеге асырылуы әрқашан да тек бір файлда болады. Бұл бағдарламалау тілінде мұрагерлік жоқ, тек конструктор бар.

Android қосымшасы арнайы VM Dalvik атты виртуалды машинада іске қосылады. Виртуальды машина негізінен маңызды емес, дегенмен жұмыс жасау кезінде пайдалы қызметтер атқарады. Яғни қабық қызметін атқару арқылы мәліметтерді сақтайды. Кез келген Android қосымшасы қандай да бір операциялық жүйеде жұмыс жасау үшін арнайы виртуалды машинаның

данасымен жұмыс жасайды. Қосымша операциялық жүйеде қаралып сақталады, егерде дұрыс жасамай жатқан жағдайда операциялық жүйе ол қосымшаны жауып тастайды.

Класстар – java тіліндегі объектіге бағытталған бағдарламалаудың базалық элементінің класс түрі болып табылады. Класс құру үшін шығыс файлы болуы жеткілікті. Файл құрамында class негізгі сөзі, одан кейін идентификаторлар мен фигуралы жақшалар жұбы болуы керек.

## **2.3 Android платформасы**

Заман мен қоғамның қатар дамуына байланысты қазіргі уақытта мобильді құрылғыларсыз өмірді елестету мүмкін емес. Ал мобильді құрылғылар көбінесе Android платформасында жасалатын бағдарламаларды қамтиды.

Android Studio – Google компаниясының Android операциялық жүйесіне арнап шығарған жаңа және толық қамтылған қосымшаларды әзірлеу ортасы болып табылады. Бұл бағдарламаны дайындаушыларға қосымшаны өздері қалаған, мысалға Eclipse ортасын тандай отырып жасауға арналған ең танымал әзірлеу ортасы болып табылады.

Жаңа бағдарламамен жұмыс жасағанда, оның барлық құрылымы SDK атты каталогта жинақталып көрсетіледі. Бағдарламаның Gradle атты бақылау жүйесінде жұмыс жасауы өндірісі процесінде оның көп икемділігін береді. Android Studio бағдарламасы нақты бір уақытта сізге кез келген визуальды көрнекі өзгерістерді көруге мүмкіндік береді. Сонымен қатар сіз қосымшаны жасай отырып, бір уақытта оның басқа Android бағдарламасы арқылы басқарылатын құрылғыларға әртүрлі баптаулармен, экран рұқсаттарымен көре аласыз. Бағдарлама безендіру құралдарымен қатар таңбалы кодтарына арналған жаңа құралдармен қамтамасыз етеді.

Android платформасы әрқашанда өзінің ашықтығымен, яғни кодтың ыңғайлылығымен ерекшеленеді. Кодтың ашықтығы арқасында бағдарлама әзірлеушілер қалаған уақытта оған өзгерістер енгізе алады, келесі бағдарламаға ауыстыра алады және оған анализ жасай алады.

Android платформасының тағы бір артықшылығы – бағдарламаны жекелендірудің көп мүмкіндігінің болуы. Мысалға алғанда соңғы операциялық жүйенің жұмыс үстелінде әзірлеуші иконкалар мен виджеттерді өзі қалаған тәртіппен орналастыра алады. Әр жаңа версиясы шыққан сайын оған көптеген өзгерістер енгізілуде. Бірақ бұл бағдарламаны жасаушыларға өте ыңғайлы емес, себебі барлық бағдарламаны жасаушылар өздері жасаған бағдарламаның ерекше қылуды қалайды да, оған әртүрлі шектеулер қояды. Ал жаңа версияда ол ашылмауы мүмкін.

Android бағдарламаның ядросында қолданылатын API ды толығымен қамтиды. Бағдарлама архитектурасы кез келген бағдарламаның дайын тұрған бағдарлама мүмкіндіктерін пайдалануға мүмкіндік береді. Бірақ соңғы

өзгертілген бағдарлама барлық шектеулерді қамту керек. Осылайша операциялық жүйе архитектурасы компоненттер мен қосымшаларды бірнеше рет пайдалану принципін ұсынады.

Android бағдарламасын басқару үшін root қатынауы қажет. Оның көмегімен орнатылған бағдарламаны және жүйені толықтай бақылауға болады. Бағдарлама құрылғыларын жасаушылар бастапқыда root қатынауын аппараттық түрде бұғаттайды. Себебі олар пайдаланушыны зиянды бағдарламалық қамтаманы орнатудан қауіптенсіздіруге және аппаратты зақымданудан қорғауға ұмтылады. Дегенмен осы қорғауды өту үшін жаппай хакерлік жұмыстар жүргізілгендіктен, өндірушілер телефондардың ресми құлпын ашу мүмкіндігін жасауға мәжбүр болды.

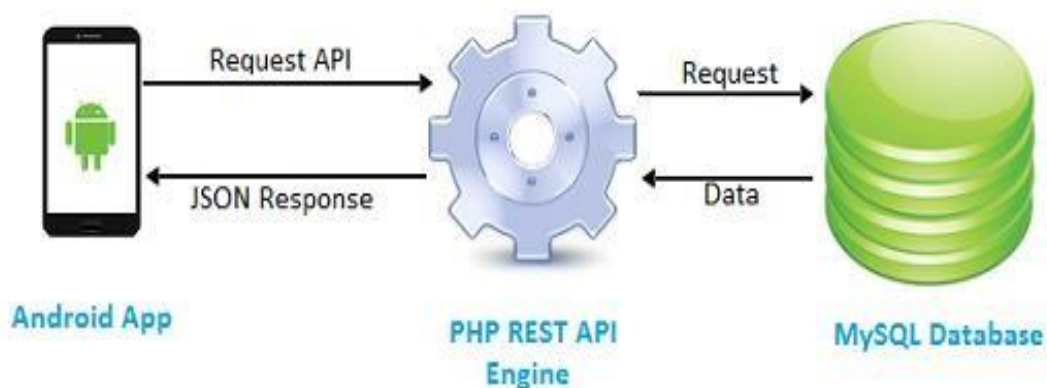
## 2.4 Деректер құрылымы

Мобильді қосымша серверінің деректер қорын басқару жүйесі ретінде MySQL пайдаланылды. MySQL сервері клиент-серверлік және кірістірілген жүйелерде жұмыс істейтін болғандықтан бағдарламаға өте қолайлы.

MySQL деректер базасын басқару жүйесі көп нүктелі SQL серверін қамтитын клиент-серверлік жүйе болып табылады, әрі әр түрлі платформаларды, бірнеше клиенттік бағдарламалармен кітапханаларды, әкімшілік құралдарын және қосымшалардың бағдарламалық интерфейстерінің кең ауқымын қолданады.

Деректер базасының сервері қолдануда үлкен жылдамдықпен және жеңілдікпен бейнеленеді. Деректер құрылымы сансыз көп мәліметтерді өңдеуге, оны сақтауға, нақты әрі тиянақты өңдеуге және жаңа мәліметтер қосуға мүмкіндік береді. Бұл бағдарламамен бір уақытта көптеген қолданушылар қолдана алады. Деректер құрылымының сұраулары арқылы мәліметтердің кестесін құруға, оны өзгертуге және жоюға болатын әрекеттер орындауға болады. Әрине, ол үшін админ қолданушысы сәйкес құқықтарға ие болу керек және де әрбір сұрау үтірлі нүктемен аяқталуы керек.

Деректер құрылымы-серверлік бағдарлама, ол әрқашанда компьютер жадында үздіксіз жұмыс жасалып тұрады. Клиенттік бағдарламалар оған арнайы сокет атты сұрауларды жіберіп отырады, ал ол оларды өңдеп нәтижесін өз жадысында сақтайды. Мәліметтер қоры өте үлкен болғандықтан, тек сұралған операцияға байланысты жауап қайтады. Деректер құрылымы өте еркін жүйе, яғни оны пайдалану үшін төлеудің қажеті жоқ. Сонымен қатар, бұл өте жылдам, сенімді ең бастысы деректер қорын пайдалануда аса үлкен жобалар үшін өте қолайлы. Деректер қорының тағы бір артықшылығы тек мәтіндік жүйеде емес, сонымен қатар графикалық түрде де жұмыс жасауға болады. Бұл артықшылықты PHP тілінде жасаған. Деректер қорының PHP тілімен қатынасын төмендегі 2.4-суретте көрсетілген.



## 2.4-сурет – Деректер қорының жұмысы

SQLite андроид құрылғылардың барлығында қолжетімді, яғни бөлек жазудың қажеті жоқ. Бұл қарапайым SQL тілінің барлық мүмкіндіктерін қамтитын, қолдануға ыңғайлы ашық кодты жоба болып табылады. Көптеген бағдарлама қажетті типтерді қолдануына байланысты бұл деректер құрылымымен жұмыс жасау ыңғайлы. SQLite бастапқы кодтары public domain сақталады, яғни онымен барлығы жұмыс істей алады.

SQLite деректер қоры файл ретінде қарастырылатын болғандықтан қолданушы файлдарымен қарым-қатынас жасайды. Сондықтан оқу және жазу операциялары баяу болуы мүмкін. Сол себепті Async Task классы көмегімен асинхронды операциялар қолдануға кеңес береді.

Жоғары тиімді инфрақұрылымды пайдалана отырып, SQLite кез келген басқа деректер базасы жүйелерімен әлдеқайда аз, оған бөлінген жадтың кішкентай көлемінде жұмыс істей алады. Бұл SQLite деректер базасына жүктелетін кез келген міндеттерді іс жүзінде пайдалану мүмкіндігімен өте ыңғайлы құрал жасайды.

SQLite деректер құрылымының әр шығарылымы кезінде өте мұқият тексеріледі. Бастапқы кодтың көп бөлігі тестілеу мен тексеруге арналған. Жүйедеге қандай да бір кедергі болса деректер құрылымы оны міндетті түрде шығарып көрсетеді. Деректерге көп жады бөлген сайын бағдарлама құрылымы өте тез жұмыс жасай бастайды. Егерде файлдық жүйемен кіру-шығу принципі арқылы жұмыс жасайтын болса, онда ол жады көлеміне қарамай өте жылдам істейді.

## 2.5 Фреймворк ұғымы

Андроид бағдарламасы мобильді құрылғыларды жасауға арналған керемет платформа, ол бүкіл ғаламшарымыздың екі жүз елу мемлекетіндегі сан мыңдаған смартфондарында жұмыс істейді.

Фреймворктің мағынасы – өнімділікті арттырып, әзірлеу күшінің құнын төмендету. Ең бастысы олар бағдарламаны дайындау барысындағы маңызды мәселелерді шешумен айналасуы үшін бағдарлама әзірлеушінің уақытын үнемдейді. Бағдарламалық қамтаманы жасаудың кешенді ортасы ретінде сипатталады.

Фреймворк көптеген компоненттерді қамтиды, басты жұмысы – бағдарламаны әзірлеушіге қажетті қосымшаны құруға көмектесу. Фреймворктар компилятормен, бағдарламалық интерфейстерімен, жабдықтар жиынтығымен, қате жөндеумен және басқа да компоненттерімен бірге жинақталған. Бағдарлама дайындаушының қосымшамен жұмысын жеңілдету үшін бірге жұмыс жасайды. Жобалармен жасау үшін негіз болып табылатын фреймворктарды қолдану жобаны жасау процесін жеңіл әрі қызықты етеді. Онсыз бағдарлама әзірлеуші өзі қалаған кез келген қосымшаны басынан бастап жаза алады. Бірақ бұл әртүрлі платформада жұмыс жасай алуы үшін үлкен кодтауды қажет етеді.

Фреймворктар – бұл қосымша құруды жеңілдетуге және техникалық күрделі немесе ауыр жобаларды қолдануға арналған бағдарламалық өнім. Ереже бойынша негізгі бағдарламалық бөлімдерден тұрады, ал бағдарлама компоненттері барлық нақты модульдер олардың негізінде әзірлеушілермен жүзеге асады. Осылай жұмыс жасау арқылы бағдарлама жасауда үлкен сенімділік пен өнімді шешімдерге келуге болады және бағдарлама жылдамдығын арттыруға септігін тигізеді. Негізінен фреймворктар мобильді құрылғыларды, веб-сервистерді, бизнес қосымшаларды және сайттарды құруға ыңғайлы болып келеді.

Фреймворк негізгі жұмысының артықшылықтары:

- кроссплатформалық. Платформалар мен құрылғылар арасындағы барлық айырмашылық қолданылып отырған фреймворк шеңберінде қарастырылған.

- Android SDK платформамен бірге дайындалады және дамиды. Егер фреймворк жаңартуды тоқтатса, онда бағдарлама кейбір кезекте технологиялық артта қалуға мүмкін болады.

Андроид бағдарламасында көптеген фреймворктар жұмыс жасайды. Себебі ол аз уақыттың ішінде көптеген құрал – жабдықтары арқылы қосымшаны жасап шыға алады.

Андроид бағдарламасын қолдану арқылы JQuery фреймворкін қолданса болады. Ол сенсорлы экрандармен жұмыс істеуге арналған. Бағдарламаны дайындаушыға десктоптарда, мобильді құрылғыларда, сайттарда, планшеттер мен смартфондарда жұмыс істейтін жауапты дизайн мен қосымшаларды құруға көмектеседі.

Барлық фрейворктарда MVC паттерні қолданылады. Компоненттермен қосымшаларда жұмыс істеуге арналған. Оның қызметі – бағдарламамен жұмыс жасаған кездегі қателіктерді жою болып табылады.

Бұл шаблон көмегімен бағдарламадағы кодты модельдегі, көріністегі және контроллердегі өзгерістерді бағдарламаға қатты әсерін тигізбей өзгерту.

## 3 Жоба құрылымы

### 3.1 «Find System» проектісі туралы

«Find System» проектісі көпфункциональды веб сайттан және мобильді қосымшадан тұрады.

Мобильді қосымша университетімізде білім беретін оқытушыларға өзіне ыңғайлы аудитория орнын табу және оны қажет жағдайда брондау үшін ал студенттерге аудитория туралы ақпарат алуға септігін тигізеді. Find System» проектісі Restful Api технологиясын қолдана отырып жасалынды.

Restful API бұл web сайтпен мобильді қосымша арасындағы байланысты үздіксіз қамтамасыз ете отырып, ақпарат алуды ұйымдастырады. Бұл проектінің негізгі қызметі университетіміз оқытушылары мен студенттеріне аудитория орнын табуды жеңілдету болып табылады. Ал мобильді қосымша барлық қолданушыларға қолдану аймағы ыңғайлы болғандықтан таңдап алынды.

### 3.2 Қолданушының интерфейсі

«Find System» мобильді қосымшасының интерфейсі басты мәзірі және бірнеше бөлімнен тұрады. Олар: басты бет, аудиторияны табу, сабақ кестесі, кері байланыс, карта, университет жайлы атты бөлімдерден тұрады.

Бұл бөлімде университетімізді болып жатқан барлық оқиғалар мен жаңалықтардан құр қалмайсыздар. Төменде 3.1-суретте көрсетілген.



3.1-сурет – Басты бет көрінісі

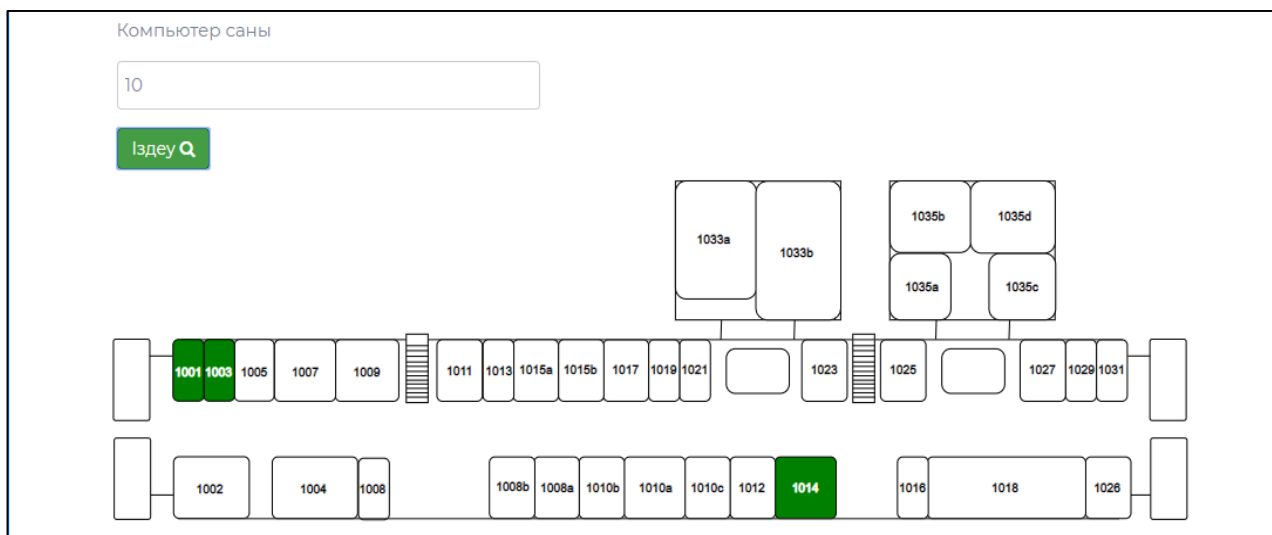
«Жаңалықтар» бөліміндегі жаңалықтарды қосу, өзгерту немесе жою сияқты жұмыстар админ бетінде орындалады. Админ бет тікелей веб сайтпен байланысты. Программаның веб сайты адаптивті болып келеді. Ұялы телефоннан немесе басқа құрылғы дан қарағанда да көзге жағымды және ыңғайлы көрінеді.

Келесі бөлім «Аудитория іздеу» бөлімі. Бұл бөлім екіге бөлінеді. Мұғалімдер бөлімінде сізге этаж, оқушылар саны және компьютерлер саны деген жолдар шығады. Жолақтарды толтыру арқылы қажетті аудиторияны табуға болады. Аудиторияны іздеу туралы көрініс төмендегі 3.2-суретте көрсетілген.

The screenshot shows a mobile application interface titled 'FindSystem' with a blue header. Below the header, the main title is 'Қажетті аудиторияны іздеу' (Find the required auditorium). Underneath, it says 'Іздеу үшін көрсетілген өрістерді толтырыңыз!' (Fill in the fields indicated for search!). There are two buttons: 'Аудитория іздеу' (Find auditorium) and 'Басқа' (Other). Below these are three input fields: 'Этаж' (Floor) with a dropdown menu showing '1', 'Оқушылар саны' (Number of students) with a text input field, and 'Компьютер саны' (Number of computers) with a text input field. At the bottom is a blue button labeled 'Іздеу' (Search) with a magnifying glass icon.

**3.2-сурет – Аудитория іздеу бөлімі**

Іздеу батырмасын басқан кезде сізге аудиториялар жасыл түспен белгіленіп көрсетіледі. Студенттерге арналған бөлімінде тек аудитория мен кафедраларды таба аласыз ал мұғалімдер бөлімінде қажетті аудитория мен қатар оқушылар санына тиісті бірнеше аудитория шығуы мүмкін. Ізделген аудиторияның көрінісі төмендегі 3.3-суретте көрсетілген.

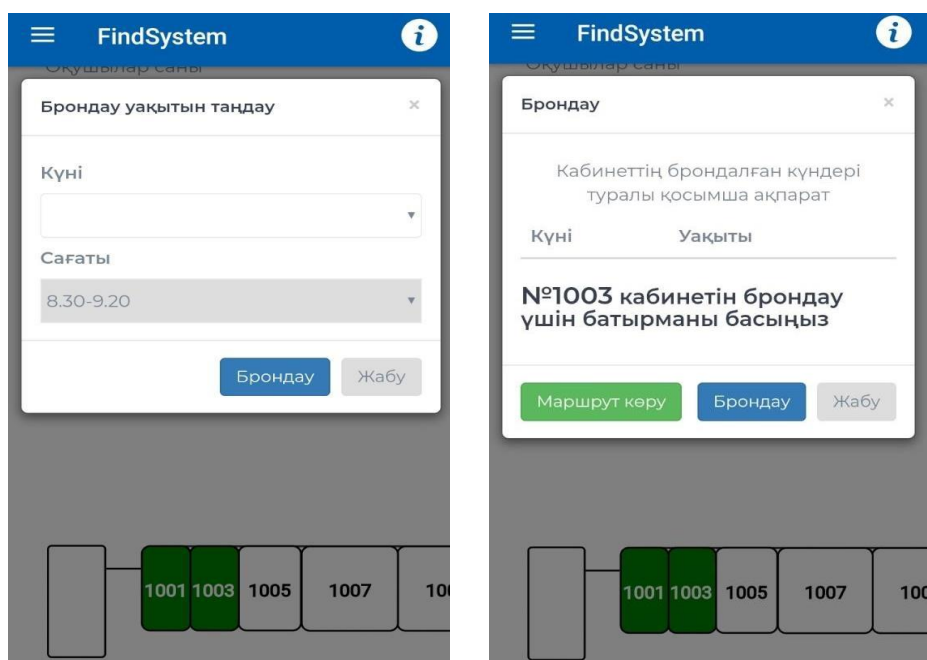


### 3.3-сурет –Табылған аудитория көрінісі

Аудиторияларды қосу, жою қызметтері админ бетінде орындалады. Аудитория картасын сызу үшін SVG графикалық редакторы қолданып жасалынды.

SVG - векторлық графиканың онлайн редакторы. Оны веб-қосымша ретінде (кез келген браузерде) немесе онлайн-сервис ретінде пайдалануға болады.

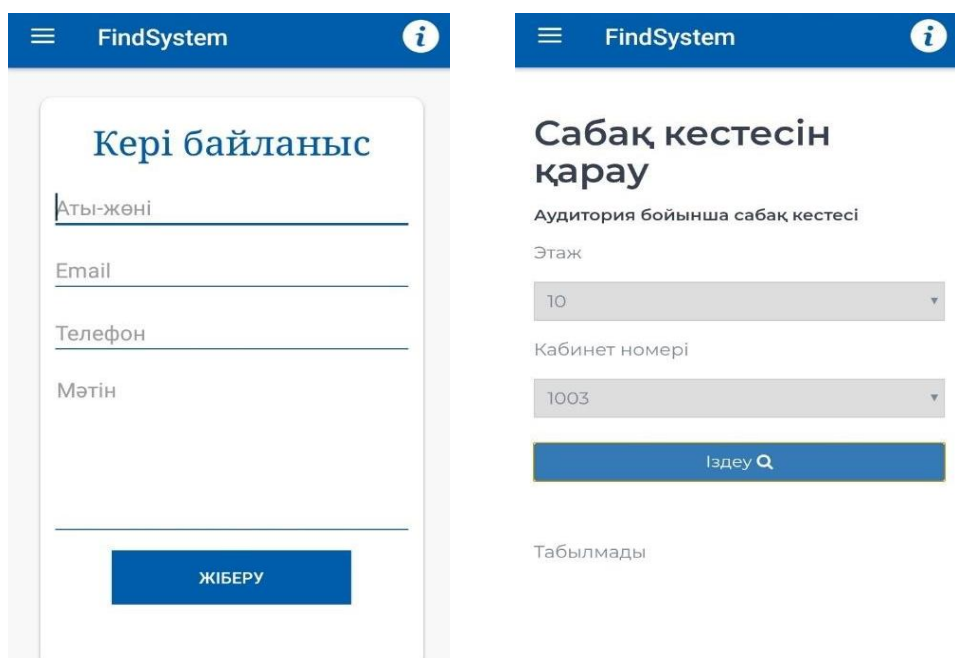
Қажетті аудиторияны тауып алғаннан кейін, оны брондау үшін табылған аудитория үстінен шертеміз. Брондау батырмасы және маршрут көрсету батырмасы пайда болады. Брондау үшін брондау батырмасын басу жеткілікті. Маршрутты көру батырмасын басу арқылы аудиторияға тиімді жолды пайдаланып баруға болады. брондау үшін кабинеттің үстін басып брондау батырмасын басу жеткілікті. Маршрутты көру батырмасын басу арқылы аудиторияға тиімді жолды пайдаланып баруға болады. Жасалынған әрекеттерді 3.4-суреттен көре аласыздар.



**3.4-сурет – Аудиторияны брондау бөлімі**

Брондау кезінде міндетті түрде логин пароль теру қажет. Брондау жүйесі 1 аптаға ғана жарамды. Брондау туралы барлық ақпарат админ бетке келіп түседі.

Қалған бөлімдерде университетімнің тарихы мен сабақ кестесі көрсетілген. Және қандай да пікір немесе жаңалық енгізу үшін кері байланыс бөлімі арқылы жасалады. Бөлімдер төмендегі 3.5-суретте көрсетілген.



**3.5-сурет –Сабақ кестесі және байланыс бөлімі**

## ҚОРЫТЫНДЫ

Дипломдық жобаны қорытындылай келсек, Android платформасында жұмыс істейтін Find System бағдарламасы университетіміз туралы нақты мәліметтер беруге арналған навигациялық гид түрінде жасалынды. Қосымша жасалу кезінде ерекше тәсілдер қолданылды. Бұл тәсілдер қосымшаның тез жұмыс жасалуына көмегін тигізеді. Бағдарлама Android Studio ортасында жасалынып және Java объектіге бағытталған бағдарламалау тілін қолданылды.

Бағдарламаның негізгі ерекшелігі – жанашылдығы. Атап айтқанда , аудиторияны брондау жүйесі. Университетіміз өте үлкен болғандықтан, бос аудитория табу немесе аудиторияның лекцияға сәйкес келмеуіне байланысты, аудиторияны брондау біз үшін таптырмас нәрсе.

Бағдарлама кең көлемді ортаға, атап айтқанда университетіміздің адамдарына, студенттер мен оқытушыларға және өсіп келе жатқан оқушыларға өздері армандаған оқу орны туралы анықтама беруге арналған.

Қосымша тестілеуден Android SDK стандартты эмуляторлардан және нақты Android құрылғыларға (планшетте ДК және смартфонда) өтіп өзінің жұмыс қабілеттілігін көрсетті.

Android ортаны зерделегеннен кейін сенімділікпен айтуға болады, бұл мобильді құрылғылар бірте-бірте функционалдық жағынан жоғары шыңдарды алуды жоспарлауда. Жақын болашақта "ақылды сағаттар" толығымен бізге үйреншікті дербес компьютерді ығыстырып, өйткені қазірдің өзінде көптеген мобильді құрылғылар өнімділігі жағынан дербес компьютермен бір деңгейде, ал "бұлтты технологиялар" күшейсе компьютерге мұқтаждық жоқ болатын болады.

Жоба веб беттен және мобильді қосымшадан тұрады. Android мобильді қосымшасы мен веб – бет арасындағы байланысты Rest технологиясы жүзеге асырады. Restful API деректерді JSON форматында қайтарады.

## ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

- 1 Хашими С., Коматинени С., Маклин Д., Android үшін қосымша жасау. – СПб: «Питер», 2011. – 736 б.
- 2 Голощапов А. Google Android: программирование для мобильных устройств. – 247 б.
- 3 Роджерс Р., Ломбардо Д. Android. Разработка приложений. М.: ЭКОМ Паблишерз, 2010. – 164 б.
- 4 Донн Фелкер. Android: разработка приложений для чайников = Android Application Development For Dummies. М.: Диалектика, 2011 – 347 б.
- 5 Сатия Коматинени, Дэйв Маклин. Android 4 для профессионалов. Создание приложений для планшетных компьютеров и смартфонов = Pro Android 4. – 216 б.
- 6 Маклин Д., Хэшими С. Google Android: программирование для мобильных устройств 2012 г. – 149 б.
- 7 Шикин Е.В., Боресков А.В. Компьютерная графика. Полигональные модели, Диалог-МИФИ, 2001г. – 243 б.
- 8 Рик Роджерс, Медниекс Зигурд и др. «Android. Разработка приложений» 2010г. – 184 б.
- 9 Моделирование на UML. Ф.Новиков, Д.Иванов. – 48 б.
- 10 А. Леоненков. Самоучитель UML (2-е издание). – СПб.: БХВ – Петербург, 2004. – 215 б.
- 11 Медникс З., Дорнин Л. Программирование под Android. – СПб.: Питер, 2012, - 560 б.
- 12 Веллинг Л., Томсон Л. Разработка веб-приложений с помощью PHP и MySQL. – М.: Вильямс, 2010. – 428 б.
- 13 Официальная справка для Android разработчиков // Сайттың электрондық нұсқасы <https://developer.android.com/studio>
- 14 Бурнет Э. Привет Андроид! Разработка мобильных приложений. – СПб.: Питер, 2012. – 256 б.
- 15 Витали Н., канал Start Android // Сайттың электрондық нұсқасы <https://startandroid.ru/ru/>
- 16 Рето М. Android 2. Программирование приложений для планшетных компьютеров и смартфонов. – М.: Эксмо, 2011. – 672 б.
- 17 Трофимов С.А CASE-технологии: практическая работа Rational Rose. – М.: Бином-Пресс, 2002. – 288 б.

## **А қосымшасы (міндетті)**

### **«Find System» мобильді қосымшасын құруға арналған техникалық тапсырма**

#### **А.1 Кіріспе**

Бүгінгі таңда көптеген адамдар ақпарат алмасу үшін ЖК немесе ноутбуктар емес, керісінше осы аталған құрылғыларға тең келетін замануи ұялы телефондарды қолданады. Нәтижесінде, мобильді қосымшаларды құру қазіргі таңда маңыздырақ болуда – бүгінде компьютерлік жүйелерге бағдарламалық қамтамасыздандыруды құруға қарағанда, мобильді қосымшаларды құру өте танымал және де көптеген адамдар осы салада қызмет етуді жөн көреді. Сондықтан оқу орнына қажетті мобильді қосымша құру қазіргі заманның өзекті мәселесі. «Find System» мобильді қосымшасы толығымен оқу орнына арналып жасалынды.

##### **А.1.1 Жобаның мақсаты мен міндеті**

«Find System» мобильді қосымшасы студенттердің, оқытушылардың уақытын үнемдеп, смартфон құрылғылар арқылы көруге арналған навигациялық гид болып табылады. Университетіміздің өте үлкен болуына байланысты сабақ кестесінде берілген аудиторияны табу қиынға түседі, сол себепті бұл мобильді қосымша ерекшелігінің бірі жаңа оқуға түскен студенттерге қажетті аудиторияны тез табуға мүмкіндік береді және оқытушылардың сабақ беру үшін қажетті аудитория туралы мәлімет алуына(компьютер саны, интерактивті тақта, парта т.б) септігін тигізеді. Сонымен қатар сол аудиторияны брондай алу мүмкіндігінің болуы. Кабинетті тек оқытушылар аутентификация жасау арқылы брондай алады. Брондау жүйесі админ бетке түсу арқылы қадағаланылады. Брондау тек сабақ кестесіне қарай брондалмайды, сонымен қатар қосымша сабақтар және группа жетекшісімен арналған жиналыстарда брондалуы мүмкін.

##### **А.1.2 Қолдану саласы**

Қолдану саласы – ұялы телефонмен қолдана білетін кез-келген адам бұл мобильді қосымшаны қолдана алады.

## *А Қосымшасының жалғасы*

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

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

#### **А.1-кесте – Анықтамалар, терминдер және қысқартулар**

| Терминдер немесе қысқартулар | Анықтамалар                                                                                                                                                                                                                                                                                                                                           |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Интернет                     | Интернет сөзі ағылшынша (байланысқан желілер) екі сөздің туындысы. Интернет ұғымы ақпараттық және есептеу техникалары ресурстарының әлемдік телекоммуникациялық желісі дегенді білдіреді.                                                                                                                                                             |
| Java                         | бірнеше компьютер бағдарламалық жасақтама өнімдерінен құралған өнім және Sun Microsystems мекемесі жазған спецификация (қазір Oracle Corporation мекемесімен бірге), барлығы қосыла келіп қосымшалар бағдарламалық жасақтарын дайындауға қолданысын табатын жүйені құрайды, бұл жүйе кросс-платформалы есептеу орталықтарында қолданысын таба береді. |
| Android                      | Android ықшамды (желілі) операциялық жүйе. Ол Linux ядросының негізінде жасалған және коммуниторлар, планшетті компьютерлерге, санды ойнатқыштарға, қолсағаттарға, нетбуктар мен смартбуктарға арнап жасалынған.                                                                                                                                      |
| UML                          | UML – бұл бағдарламалық жүйелерді ерекшелендіру, бұрыштама қою, конструкциялау және құжаттамалау, сондай-ақ модельдер бизнесі мен өзге де бағдарламалық емес жүйелердің тілі болып табылады.                                                                                                                                                          |

### **А.2 Жалпы сипаттамасы**

Дерекқор жұмысы клиент-сервер технологиясына негізделген. Мобильді қосымша қолданушы бөлімінен тұрады. Қолданушы мобильді қосымшаларды көріп, қажетті жүйелерді жүргізе алады.

### **А.2.1 Пайдаланушы интерфейстер**

Ақпараттық жүйенің ыңғайлы және түсінікті бағдарламалық интерфейстен тұруы мобильді қосымша үшін маңызды рөлге ие. Қолданушылардың қосымшаны қолдануы осы бағдарламалық интерфейске тікелей қатысты.

### **А.2.2 Аппараттық интерфейстер**

Компьютерлерге қойылатын жалпы талаптар:

- процессор – P4 және жоғары, Celeron;
- оперативті жады – 512 MB;
- диск тұлғалы кеңістік – ~ 52 MB + қолданушылар файлдарын сақтауға қажет орын;
- Internet желісіне қатынау;
- қатты диск 40 Gb; 1024×768 рұқсатта жұмыс қолдайтын видеокарта;
- пернетақта, манипулятор тышқан.

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

Жобаға қатысты бағдарламалық компоненттер:

- Windows 8,10 операциялық жүйесі;
- Android SDK бағдарламасы ;
- MySQL бағдарламасы;
- Java бағдарламасы.

Қолданушы бағдарлама компоненттер: кез келген операциялық жүйе, ұялы телефон, интернет желісіне қосыла алу, әр түрлі браузер.

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

Пайдаланушылардың компьютерлерінде интернетке қатынауға бар модем құралдары немесе жылдамдығы 100 Мб/с кем емес желілі карта болуы тиіс. Сервер мен клиенттерді байланыстыратын TCP/IP протоколы қолданылады.

### **А.2.5 Жады бойынша шектеулер**

Серверлік және клиенттік машиналар үшін жалпы жады бойынша қойылатын талаптар, оларда орналасқан жұмысқа қажетті программалық компоненттердің қойылатын талаптармен анықталады.

### **А.2.6 Адаптация бойынша талаптар**

«Find System» мобильді қосымшасы Android және Windows платформаларында жұмыс істей алады.

**Б қосымшасы**  
(міндетті)

**Бағдарламаның мәтіні**

1. *FragmentActivity.java* бөлімі

```
package com.kaznitu.findsystem;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.design.widget.NavigationView;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentTransaction;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.FrameLayout;
import android.widget.TextView;
import com.kaznitu.findsystem.Fragment_pages.AboutUniversity;
import com.kaznitu.findsystem.Fragment_pages.Contacts;
import com.kaznitu.findsystem.Fragment_pages.Find;
import com.kaznitu.findsystem.Fragment_pages.Map;
import com.kaznitu.findsystem.Fragment_pages.Schedule;
import com.kaznitu.findsystem.Fragment_pages.Student;
import com.kaznitu.findsystem.Fragment_pages.TabFragment;
public class FragmentActivity extends AppCompatActivity {
    private DrawerLayout mDrawerLayout;
    private ActionBarDrawerToggle mToggle;
    private NavigationView navigationView;
    private FrameLayout containerView;
    private WebView webView;
    private static final int REQUEST_CODE_SECOND = 1;
    FragmentManager FM;
    FragmentTransaction FT;
    TextView text;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
setContentView(R.layout.activity_fragment);
mDrawerLayout = (DrawerLayout) findViewById(R.id.drawerLayout);
mToggle = new ActionBarDrawerToggle(this, mDrawerLayout, R.string.open,
R.string.close);
navigationView = (NavigationView) findViewById(R.id.navigationView);
//webView = (WebView) findViewById(R.id.webView);
mDrawerLayout.addDrawerListener(mToggle);
containerView = (FrameLayout) findViewById(R.id.containerView);
text = (TextView) findViewById(R.id.text);
FM = getSupportFragmentManager();
FT = FM.beginTransaction();
FT.replace(R.id.containerView, new TabFragment()).commit();
mToggle.syncState();
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
navigationView.setNavigationItemSelectedListener(new
NavigationView.OnNavigationItemSelectedListener() {
    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        mDrawerLayout.closeDrawers();
        if (item.getItemId() == R.id.nav_home) {
            FragmentTransaction fragmentTransaction = FM.beginTransaction();
            fragmentTransaction.replace(R.id.containerView, new
TabFragment()).addToBackStack("tag").commit();
        }
        if (item.getItemId() == R.id.nav_search) {
            FragmentTransaction fragmentTransaction = FM.beginTransaction();
            fragmentTransaction.replace(R.id.containerView, new
Find()).addToBackStack("tag").commit();
        }
        if (item.getItemId() == R.id.nav_search_student) {
            FragmentTransaction fragmentTransaction = FM.beginTransaction();
            fragmentTransaction.replace(R.id.containerView, new
Student()).addToBackStack("tag").commit();
        }
        if (item.getItemId() == R.id.nav_about) {
            FragmentTransaction fragmentTransaction1 = FM.beginTransaction();
            fragmentTransaction1.replace(R.id.containerView, new
AboutUniversity()).addToBackStack("tag").commit();
        }
    }
});
```

```
    }
    if (item.getItemId() == R.id.nav_schedule) {
        FragmentTransaction fragmentTransaction1 = FM.beginTransaction();
        fragmentTransaction1.replace(R.id.containerView, new
Schedule()).addToBackStack("tag").commit();
    }
    if (item.getItemId() == R.id.nav_contact) {
        FragmentTransaction fragmentTransaction1 = FM.beginTransaction();
        fragmentTransaction1.replace(R.id.containerView, new
Contacts()).addToBackStack("tag").commit();
    }
    if (item.getItemId() == R.id.nav_map) {
        FragmentTransaction fragmentTransaction = FM.beginTransaction();
        fragmentTransaction.replace(R.id.containerView, new
Map()).addToBackStack("tag").commit();
    }
    return false;
}
});
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    if (mToggle.onOptionsItemSelected(item)) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.appmenu, menu);
    return super.onCreateOptionsMenu(menu);
}
public void onClick(MenuItem item) {
    AlertDialog.Builder builder = new AlertDialog.Builder(FragmentActivity.this);
    switch (item.getItemId()) {
        case R.id.change_kz:
            builder.setTitle("FindSystem қосымшасы");
            builder.setMessage("Бұл мобильді қосымша студенттерге қажетті аудитория, асхана және дәретхананы тез табуға арналған.");
```

```
    }  
    AlertDialog alertDialog = builder.create();  
    alertDialog.show();  
}  
}
```

## *2. NewsAdapter.java бөлімі*

```
package com.kaznitu.findsystem;  
import android.content.Context;  
import android.content.Intent;  
import android.support.v7.widget.RecyclerView;  
import android.text.Html;  
import android.text.format.DateFormat;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
import android.widget.LinearLayout;  
import android.widget.RelativeLayout;  
import android.widget.TextView;  
import com.kaznitu.findsystem.models.NewsList;  
import com.squareup.picasso.Picasso;  
import java.util.Calendar;  
import java.util.List;  
import java.util.Locale;  
public class NewsAdapter extends  
RecyclerView.Adapter<NewsAdapter.ViewHolder> {  
    public static final String KEY_TITLE = "title";  
    public static final String KEY_IMAGE = "image";  
    public static final String KEY_TEXT = "text";  
    public static final String KEY_DATE = "date";  
    // we define a list from the newsList java class  
    private List<NewsList> newsLists;  
    private Context context;  
    public NewsAdapter(List<NewsList> newsLists, Context context) {  
        // generate constructors to initialise the List and Context objects  
        this.newsLists = newsLists;  
        this.context = context;  
    }  
    @Override
```

```
public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {  
    // this method will be called whenever our ViewHolder is created  
  
    View v = LayoutInflater.from(parent.getContext())  
        .inflate(R.layout.item, parent, false);  
    return new ViewHolder(v);  
}  
  
@Override  
public void onBindViewHolder(ViewHolder holder, final int position) {  
    // this method will bind the data to the ViewHolder from whence it'll be shown  
    to other Views  
    final NewsList newsList = newsLists.get(position);  
    holder.title.setText(newsList.getTitle());  
    String htmlText = Html.fromHtml(newsList.getText()).toString();  
    holder.text.setText(htmlText);  
    Calendar cal = Calendar.getInstance(Locale.ENGLISH);  
    int timestamp = Integer.parseInt(newsList.getDate().toString());  
    cal.setTimeInMillis(timestamp * 1000L);  
    String date = DateFormat.format("dd.MM.yyyy", cal).toString();  
    holder.date.setText(date);  
    Picasso.with(context)  
        .load(newsList.getImage())  
        .into(holder.image);  
    holder.linearLayout.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            NewsList newsList1 = newsLists.get(position);  
            Intent skipIntent = new Intent(v.getContext(), NewsDetailActivity.class);  
            skipIntent.putExtra(KEY_TITLE, newsList1.getTitle());  
            skipIntent.putExtra(KEY_TEXT, newsList1.getText());  
            skipIntent.putExtra(KEY_IMAGE, newsList1.getImage());  
            skipIntent.putExtra(KEY_DATE, newsList1.getDate());  
            v.getContext().startActivity(skipIntent);  
        }  
    });  
}  
  
@Override  
//return the size of the listItems (newsList)  
public int getItemCount() {  
    return newsLists.size();  
}
```

```
public class ViewHolder extends RecyclerView.ViewHolder {  
    // define the View objects  
    public TextView title;  
    public ImageView image;  
    public TextView text;  
    public TextView date;  
    public RelativeLayout linearLayout;  
    public ViewHolder(View itemView) {  
        super(itemView);  
        // initialize the View objects  
        title = (TextView) itemView.findViewById(R.id.title);  
        image = (ImageView) itemView.findViewById(R.id.imageView);  
        text = (TextView) itemView.findViewById(R.id.text);  
        date = (TextView) itemView.findViewById(R.id.date);  
        linearLayout = (RelativeLayout) itemView.findViewById(R.id.linearLayout);  
    }  
}
```

### *3. Home.java бөлімі*

```
package com.kaznitu.findsystem.Fragment_pages;  
import android.app.ProgressDialog;  
import android.os.Bundle;  
import android.support.annotation.Nullable;  
import android.support.v4.app.Fragment;  
import android.support.v7.widget.LinearLayoutManager;  
import android.support.v7.widget.RecyclerView;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Toast;  
import com.android.volley.Request;  
import com.android.volley.RequestQueue;  
import com.android.volley.Response;  
import com.android.volley.VolleyError;  
import com.android.volley.toolbox.StringRequest;  
import com.android.volley.toolbox.Volley;  
import com.kaznitu.findsystem.NewsAdapter;  
import com.kaznitu.findsystem.R;
```

```
import com.kaznitu.findsystem.models.NewsList;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.List;
public class Home extends Fragment {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
    private static final String URL_DATA =
"http://auditory.sagadatbatyr.kz/api/web/posts";
    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private List<NewsList> newsLists;
    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup
container, @Nullable Bundle savedInstanceState) {
        View InputFragmentManager = inflater.inflate(R.layout.news, container, false);
        recyclerView = (RecyclerView)
InputFragmentManager.findViewById(R.id.recyclerView);
        recyclerView.setHasFixedSize(true);
        LinearLayoutManager mLayoutManager = new
LinearLayoutManager(getActivity());
        mLayoutManager.setReverseLayout(true);
        mLayoutManager.setStackFromEnd(true);
        recyclerView.setLayoutManager(mLayoutManager);
        newsLists = new ArrayList<>();
        loadUrlData();
        return InputFragmentManager;
    }
    private void loadUrlData() {
        final ProgressDialog progressDialog = new ProgressDialog(getActivity());
        progressDialog.setMessage("Жүктелуде...");
        progressDialog.show();
        StringRequest stringRequest = new StringRequest(Request.Method.GET,
URL_DATA, new Response.Listener<String>() {
```

```
@Override
public void onResponse(String response) {
    progressDialog.dismiss();
    try {
        JSONArray JA = new JSONArray(response);
        for (int i = 0; i < JA.length(); i++) {
            JSONObject JO = (JSONObject) JA.get(i);
            NewsList newsList = new NewsList(JO.getString("title"),
            JO.getString("text"),
            JO.getString("image"), JO.getString("date"));
            newsLists.add(newsList);
        }
        adapter = new NewsAdapter(newsLists,
        getActivity().getApplicationContext());
        recyclerView.setAdapter(adapter);
    } catch (JSONException e) {
        e.printStackTrace();
    }
}

}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        Toast.makeText(getActivity(), "Интернетті қосу керек!",
        Toast.LENGTH_SHORT).show();
        progressDialog.dismiss();
    }
});

RequestQueue requestQueue = Volley.newRequestQueue(getActivity());
requestQueue.add(stringRequest);
}
}
```

#### *4. MyAdapter.java бөлімі*

```
package com.kaznitu.findsystem.Fragment_pages;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentPagerAdapter;
import static com.kaznitu.findsystem.Fragment_pages.TabFragment.int_items;
```

```
public class MyAdapter extends FragmentPagerAdapter {
    public MyAdapter(FragmentManager fm)
    {
        super(fm);
    }
    @Override
    public Fragment getItem(int position) {
        switch (position){
            case 0:
                return new Home();
            case 1:
                return new Gallery();
        }
        return null;
    }
    @Override
    public int getCount() {
        return int_items;
    }
    public CharSequence getPageTitle(int position){
        switch (position){
            case 0:
                return "Жаңалықтар";
            case 1:
                return "Галерея";
        }
        return null;
    }
}
```

