

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
Коммерциялық емес акционерлік қоғамы
« ҒҰМАРБЕК ДӘУКЕЕВ атындағы АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ
БАЙЛАНЫС УНИВЕРСИТЕТІ»
«IT – инжиниринг» кафедрасы

«Қорғауға жіберілді»
Кафедра меңгерушісі

(аты – жөні, ғылыми дәрежесі, атағы)

_____ « ____ » _____ 20 ____ ж.
(қолы)

ДИПЛОМДЫҚ ЖОБА

Тақырыбы: Андроид платформасында мейрамхана үшін «Мәзір» мобильді қосымшасын құру
Мамандығы: 5B070400 - «Есептеу техникасы және бағдарламамен қамтамасыз ету»
Орындаған: Мәртай Д.Е
Жетекші: доцент Маликова Ф.У. Тобы: ВТку-17-5

Кеңесшілер:

Экономикалық бөлім: э.ғ.д., профессор _____ Ибришев Н.Н.
« ____ » _____ 2020ж.

Өміртіршілігі қауіпсіздігі: к.б.н., доцент _____ Мусаева Ж.К.
« ____ » _____ 2020ж.

Есептеу техникасын қолдану: аға оқытушы _____ Айтқулов Ж.С.
« ____ » _____ 2020ж.

Норма бақылаушы: аға оқытушы _____ Абсатарова Б.Р.
« ____ » _____ 2020ж.

Пікір жазушы: PhD, доцент _____ Маликова Ф.У.
« ____ » _____ 2020ж.

Алматы 2020

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
«АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ»
коммерциялық емес акционерлік қоғамы

Басқару жүйелері және ақпараттық технологиялар институты

IT-инжиниринг кафедрасы

Мамандығы 5В070400 – «Есептеу техникасы және
бағдарламалық қамтамасыз ету»

Дипломдық жұмысты орындауға берілген
ТАПСЫРМА

Білім алушы Мәртай Д.Е.

Жұмыстың тақырыбы: Андроид платформасында мейрамхана үшін
"Мәзір" мобильді қосымшасын құру

2019 жыл «11» қараша № 147 университет бұйрығымен бекітілген.

Аяқталған жұмысты тапсыру мерзімі: « ____ » _____ 2019 ж.

Дипломдық жұмыстың бастапқы мәліметтері (зерттеу (жұмыс)
нәтижелерінің талап етілген параметрлері мен объектінің бастапқы
мәліметтері): Ұсынылып отырған дипломдық жобада Android платформасына
мейрамхана үшін «Мәзір» мобильдік қосымшаны құру. Мобильдік
қосымшасын әзірлеу барысында React Native ортасын, NodeJs кітапханасын
және Visual Code Studio бағдарламалау тілін қолдандым.

Дипломдық жұмыста қарастырылған мәселелер тізімі немесе диплом
жұмысының қысқаша мазмұны:

- мобильді қосымшалар технологиясына шолу;
- мобильді қосымшаны UML тілінде жобалау бөлімі;
- мобильді қосымшаны жүзеге асыру бөлімі;
- өміртіршілік қауіпсіздігі бөлімі;
- экономикалық бөлім;
- А қосымшасы. Программа листингі;
- Ә қосымшасы. Енгізу актісі.

Ұсынылатын негізгі әдебиеттер:

1 Unity в действии. Мультиплатформенная разработка на C#. - М.: Питер, 2018. - 608 с.

2 Магда, Ю. С. NI Measurement Studio. Практика разработки систем измерения и управления на C# / Ю.С. Магда. - М.: ДМК Пресс, 2017. - 567 с

3 Вендров А.М. CASE–технологии. Современные методы и средства проектирования информационных систем. – М.: Финансы и статистика, 2000.

4 UML. Основы. Краткое руководство по стандартному языку объектного моделирования Мартин Фаулер. - М.: Символ-плюс, 2016. - 192 с.

Дипломдық жұмыстың бөлімдеріне қатысты белгіленген кеңестер

Бөлімдер	Кеңесшілер	Мерзімі	Қолы
Экономикалық бөлім	Ибришев Н.Н.	11.05.2020	
Өміртіршілік қауіпсіздігі бөлімі	Мусаева Ж.К.	11.05.2020	
Программалық бөлім	Айтқулов Ж.С	20.04.2020	
Норма бақылау	Абсатарова Б.Р.	18.05.2020	

Дипломдық жұмысты дайындау
КЕСТЕСІ

Бөлімдер атауы, қарастырылатын мәселелер тізімі	Ғылыми жетекшіге ұсыну мерзімдері	Ескерту
Мобильді қосымшалар технологиясына шолу	2.03.2020	
Мобильді қосымшаны жобалау UML тілінде жобалау	16.03.2020	
Мобильді қосымшаны іске асыру	04.02.2020	

Тапсырманың берілген уақыты «02» қазан 2020 ж.

Кафедра меңгерушісі _____ А.А. Досжанова

Жұмыстың ғылыми жетекшісі _____ Ф.У. Маликова

Тапсырманы орындауға алған білім алушы _____ Д.Е. Мәрттай

АҢДАТПА

Қазіргі таңда көптеген инновациялық технологиялар айтарлықтай дамуға қол жеткізді. Ақпараттық технологиялардың дамыған бағыттарының бірі – мобильді қосымшаларды пайдалана отырып, жұмыстың оңтайлы шешімін табу. Бұл бағыт болашақта бүкіл әлемді өзгерте алады, оны әлдеқайда ыңғайлы, интерактивті және қауіпсіз ете алады. Мобильді қосымшаларды өмір қолданысына енгізу бағытында көптеген компаниялар жұмыс істейді, яғни, бұл технология барлық қызмет салаларында қолданылады. Олардың көпшілігі осы технологияны білім беру саласына – мектепке дейінгі мекемелерден жоғары оқу орындарына дейін енгізуге бағытталған.

Бұл дипломдық жұмыста мейрамхана саласы үшін жұмыс күшін жеңілдетуге арналған мобильді қосымшаны әзірлеу қарастырылады. Қосымша санаулы уақыттардан кейін мейрамхана саласында қолданысқа енгізіліп, кең өзектілік алуының ықтималдылығы өте жоғары. Дипломдық жұмыс мақсаты «Аспандияр Экспрес» мейрамханасы үшін Android палтформасында мобильді қосымшаға арналған «Мәзір» - ді құру.

АННОТАЦИЯ

В настоящее время многие инновационные технологии достигли значительного развития. Одно из развитых направлений инновационных технологий – нахождение оптимального решения работы с использованием мобильных приложений. Данное направление в будущем может изменить весь мир, сделать его более удобным, интерактивным и безопасным. Внедрением мобильных приложений в повседневную жизнь занимается множество компаний, то есть данная технология применяется во всех сферах деятельности. Многие из них направлены на внедрение данной технологии в сферу образования – от дошкольных учреждений до высших учебных заведений.

В данной дипломной работе рассматривается разработка мобильного приложения для ресторанной отрасли в целях облегчения рабочей силы. Высока вероятность того, что приложение получит широкую актуальность, найдя применение в ресторанной отрасли. Цель дипломной работы заключается в создании «Меню» для мобильного приложения на платформе Android для ресторана «Аспандияр Экспрес».

ANNOTATION

Currently, many innovative technologies have achieved significant development. One of the developed areas of innovative technologies is finding the optimal solution to work using mobile applications. In the future, this direction can change the whole world, make it more convenient, interactive and safe. Many companies are implementing mobile applications in everyday life, that is, this technology is used in all areas of activity. Many of them are aimed at introducing this technology in the field of education - from preschool to higher education institutions.

This thesis deals with the development of a mobile application for the restaurant industry in order to facilitate the workforce. It is highly likely that the application will gain wide relevance, having found application in the restaurant industry. The purpose of the thesis is to create a “Menu” for a mobile application on the Android platform for the restaurant “Aspandiyar Express”.

Мазмұны

Кіріспе	8
1 Мобильді қосымшалар және Android операциялық жүйесі	10
1.1 Қазіргі мобильдік операциялық жүйесіне шолу	10
1.2 Android операциялық жүйесінің ерекшеліктері мен архитектурасы	11
1.3 Android қосымшасының компоненттері	16
1.4 Android платформасын қолданудағы артықшылықтары	18
2 Android платформасында мейрамхана үшін «Мәзір» мобильді қосымшаны UML тілінде жобалау	21
2.1 Мобильді қосымшаның әзірлеу ортасын және технологиясын таңдау	21
2.2 Мобильді қосымшаны жобалау	26
3 Қолданбалы бөлім	30
3.1 Бағдарламамен қамтамасыздандыру сипаттамасы	30
3.1.1 Visual Studio Code бағдарламалау тіліне сипаттама	30
3.1.2 React Native көмегімен кросс-платформалық қосымшаларды құру	35
3.1.3 NodeJS бағдарламалық платформасы	39
3.1.4 Expo көмегімен жергілікті React қосымшаларын жасау процесін жеңілдету	40
3.2 Мобильді қосымшаны іске асыру және бағдарламалық кешеніне сипаттама	41
4 Экономикалық бөлім	49
4.1 Экономикалық бөлімге кіріспе	49
4.2 Бағдарламалық өнімді әзірлеудің еңбек сыйымдылығын есептеу	49
4.3 Еңбекақы төлеу шығындарын есептеу	53
4.4 Бағдарламалық өнімнің ықтимал бағасын анықтау	60
4.5 Бағдарламалық өнімнің экономика бөлімі бойынша қорытынды	60
5 Өміртіршілік қауіпсіздігі	61
5.1 Адам тіршілік қауіпсіздігі мен еңбек қауіпсіздігі	61
5.2 Дербес компьютердің адамға тигізер әсері және соған қойылатын талаптар	62
5.3 Жұмыс бөлме зонасындағы еңбек жағдайын талдау	65
5.3.1 Жұмыс бөлмелеріндегі микроклимат	65
5.3.2 Жарықтандыру талаптары	65
5.3.3 Электр және өрт қауіпсіздігі	67
5.4 Мейрамханадағы өрт сөндіру құралдарын таңдау	69
5.4.1 Өрт қауіпсіздігі жағдайында эвакуациялану уақытын есептеу	70
5.5 Өміртіршілік қауіпсіздігі бөлімі бойынша қорытынды	70
Қорытынды	74
Әдебиеттер тізімі	76
А қосымшасы. Бағдарлама листингі	78
Ә қосымшасы. Енгізу актісі	109

Кіріспе

Біздің заманымызда мобильді құрылғылар адамдардың күнделікті өмірінде орнықты орынды алды. Олар өзінің жолын телефондар ретінде алып, салдарынан одан жоғары функцияларына ие болып, ақыр аяғында пайдалы көмекшілерге айналды. Бұдан әрі, ғаламтордың дамуымен мобильді құрылғысы дүниежүздің кез келген жерінен дүниежүзілік ақпараттық ресурстарына қолжетімділігінің амалы болып табылады. Егер ақпарат қазіргі қоғамда неғұрлым талап етілетін ресурстардың бірі болып табылатынын есептесек, сондай-ақ мобильді құрылғыларының бағасына, техникалық сипаттамаларына және сапасына әсер ететін технологиялардың дамуына назарымызды аударсақ, онда тіпті табиғи салдарымен қазіргі уақытта мобильді құрылғысына деген сұраныс өте жоғарғы деңгейде болып саналады.

Көптеген компаниялар өзінің жасау тізімінде мобильді техниканы таңдауда. Планшеттер, смартфондар, телефондар, навигаторлар – осы гаджеттердің атаулары күн сайын жаңартылады және қосылады. Қазіргі күнде смартфонсыз адамды көру коэффициенті өте аз. Мобильді техниканы қолданудың озаттығы – оны әрбір жерде және көп уақытты жоғалтпай қолдануға болады. Бүгінгі таңда осындай техниканы тапсырушы екі үлкен компания: Apple және Samsung мобильді техниканы жасау аумағында өздерін белсенді көрсетіп, мобильді техниканы жасауда көшбасшы болып келеді. Әрбір компанияның өзінің амалдық жүйесі шығарылған.

Мобильді құрылғыларына арналған бағдарламалық қамтамасыздандыру нарығы мобильді құрылғыларға деген жоғары сұранысымен байланысты дамып келеді және әр түрлі мобильді қосымшалар түрлері дамытуда.

Мобильді телефондардың жетілдіруінің басты бағыты технологиялардың конвергенциясы түсінігімен анықтауға болады. Мобильді телефондар бір құрылғы корпусына функционалдық мүмкіндігі жағынан барлық қажетті функцияларды біріктіріп отыр.

Бұрынғы телефондарының негізгі мақсаттары – қоңырауларды жасау және қабылдау, SMS жазу. Бүгінгі таңда осы мақсаттар ғаламтор-ресурстарымен, әннің тыңдауымен, фотоларды түсірумен, ойындар мен қосымшалардың қолдануымен толықтырылып отыр. Қазіргі таңда мобильді қосымшалар арқылы негізгі мақсаттардың тізіміне сондай-ақ телебағдарламаларды қарап шығу, әр түрлі техникаларды басқару, телефон иесінің денсаулық жағдайын бақылау функцияларын және т.б. үлкен толқындардың серпілісін жасауда.

Барлық мобильді қосымшаларды шартты түрде жұмыс мақсаттар үшін және ойын-сауық бағдарламаларына бөлуге болады. Біріншісі бизнесмендер мен кеңселік қызметкерлерге бизнес-үрдістерін бақылауға, талдаулық есептемесін құрастыруға, фирмалық стиль дизайнын әзірлеу сияқты есептерді орындауға мүмкіндік береді. Екінші әр түрлі ойындарды, фильмдерді көруге және әнді тыңдауға арналған бағдарламалардан, қатынаста болу үшін арналған

құралдардан және т.б. тұрады, сол себепті әрбір мобильді қосымшасының өзіндік қолданушысын табады.

Мобильді құрылғыларының өте маңызды бөлігі оның бағдарламалық қамтамасыздандырулары болып табылады. Ол әр түрлі бизнес-есептердің шешуіне себепші болады. Оқыту үшін қызмет ете алады, маңызды ақпаратты сақтау үшін, ойын-сауық үшін, жұмыста көмек беру үшін, ақпаратты жинастыру үшін, сонымен қатар көптеген басқа функцияларды орындау үшін жетілдіріліп отырады.

Диплом жұмысының мақсаты Android операциялық жүйесінің басқаруымен смартфондар үшін «Мәзір» мобильді қосымшасын құру.

Зерттеу объектісі - Android платформасында арналған мобильді қосымша.

Диплом жұмысының белгіленген мақсатына жету үшін келесі мақсаттар қойылады және шешіледі:

- пәндік аймақты зерттеу;
- техникалық тапсырманы құрастыру;
- Android амалдық жүйесінің басқаруымен смартфондар үшін қосымшаларды құрастыру бойынша жобалау жұмыстарының теориялық және тәжірибелік негіздерін, сонымен қатар жүйеге қойылатын талаптарды қарастыру;
- ақпараттық, бағдарламалық және техникалық қамтамасыздандыру түрлеріне шолу жасау;
- жобаның экономикалық тиімділігін есептеу және негіздеу.

Дипломдық жұмыс міндеттері - қолданушыларға қажетті мәліметтерді беру, ыңғайлы және қолжетімді ортада қажетті ақпаратты ұсыну. Мобильді қосымша арқасында қолданушылар өзіне қажетті мағлұматтарға ие бола алады.

1 Мобильді қосымша және Android операциялық жүйесі

1.1 Қазіргі мобильдік операциялық жүйесіне шолу

Бүгінгі таңда мобильдік техникамен жұмыс істеудің көптеген операциялық жүйелері бар.

2010 жылға дейін мобильдік операциялық жүйелер арасында көшбасшысы болған Nokia компаниясының Symbian операциялық жүйесі қазір өзінің әл ауқатын толығымен жоғалтты. Осы жаңалықтардан кейін – Sony Ericsson компаниясы өз техникасында Symbian операциялық жүйесін қолданбау шешімін шығарды, содан кейін Samsung компаниясы Sony Ericsson сияқты шешімге келді. Ал 2011 жылдың көктемінде Nokia өзі де Windows Phone жүйесіне көшу туралы жаңалық берді, ал Symbian операциялық жүйесін 2016 жылға дейін жұмыс істейді деп үміттенді.

Мобильдік техникаға арналған операциялық жүйелердің нарығын қарастырсақ, Android операциялық жүйесі қолжетімділігімен сұранысқа ие.

Android – ықшамды операциялық жүйе. Ол Linux ядросының негізінде және коммуникаторлар, планшетті компьютерлерге, санды ойнатқыштарға, қолсағаттарға, нетбуктар мен смартбуктарға арнап жасалынған. Алғашында Android амалдық жүйесін «AndroidInc.» компаниясы иелігінде болды. Кейіннен бұл компанияны Google сатып алды.



1.1 сурет – Android ОЖ негізгі брендтік белгісі

Содан кейін Google қазіргі кезде Android платформасының дамуына ат салысып жүрген Open Handset Alliance (ОНА) альянсын құруға ынта білдірді. Android операциялық жүйесі Google кітапхана арқылы жасалынған басқару құрылғысымен Java қолданбаларын жасауға мүмкіндік береді.

Android Native Development Kit, Си және басқа да бағдарламалау тілінде жазылған қолданбаларды жасай алады.

Android операциялық жүйесінің негізгі артықшылықтарын қарастырайық:

- біріншіден, оның толық қолжетімділігі. Мобильдік қосымшаларды құрастыру әрбір адамға рұқсат, сондықтан да Android операциялық жүйесі ең танымал және күн сайын жаңартылатын жүйе болып саналады;

- екіншіден, Android операциялық жүйесі универсалды болып табылады. Android операциялық жүйесін смартфондарға, комуникаторларға, телефондарға, тіпті телевизорларға да орнатуға болады.

Android амалдық жүйесі тиімді және қолжетімді, бүгінгі күнге дейін аса үлкен танымалдылықпен қолданылуда және оны икемдеу түрлері оңайға соғады.

Android операциялық жүйесінің танымалдылығы – көптеген озат компаниялардың, оның ішінде HTC, Motorola, Samsung Electronics және де басқалардың Android операциялық жүйесін өз мобильді техникасында қолданып жүргенін көреміз және де Android операциялық жүйесінің тағы бір артықшылығы – оның әр үш ай сайын жаңартылуы. Осындай жаңартулар адамға әрдайым жаңа және жұмыс қабілеті жоғары техникамен қолдануға мүмкіндік береді.

1.2 Android операциялық жүйесінің ерекшеліктері мен архитектурасы

Android платформасы құрылғыны (аппаратураны), жадыны және процестерді басқару үшін арналған Linux негізіндегі амалдық жүйесінің бағдарламалық стегі болып табылады. Бұл стектің әртүрлі компоненттері иерархияның бірнеше деңгейлері түрінде болады және жалпы мобильді құрылғының қызмет етуін қамтамасыз етеді. Бұл компоненттердің кейбірі тікелей құрылғы жабдығымен жұмыс жасау үшін қажет, басқалары телефония, жайғастыру, мультимедиямен және т.б. байланысты функцияларды қамтамасыз етеді. Ал үшіншілері мобильді құрылғысының көптеген мүмкіндіктерін іске асыру үшін қосымшаларға фреймворкты ұсынады. Басқаша айтқанда, Android платформасы өзіне амалдық жүйені, аралық деңгейлі бағдарламалық қамтамасыздандыруды (middleware) және қосымшалар жиынтығын (ішкі қосымшаларды, шеткі қосымшаларды) қосады.

Android операциялық жүйесіне арналған мобильдік қосымшаны бағдарламалау алдында оның негізгі ерекшеліктерімен және архитектурасымен танысу керек.

Android платформасын бірыңғай аппараттық-бағдарламалық жиынтығы ретінде қарастыруға болады, ол келесідей құрамға ие:

- аппаратура – құрылғылардың қызмет етуін қамтамасыз ететін элементтер жиынтығы;

- Linux Kernel (+Drivers) – Linux операциялық жүйесінің ядросы мен драйверлер жиынтығы, жабдықпен басқаруының негізгі механизмдерін және жад бөлуін, мақсаттарды басқаруымен, қауіпсіздіктің қамтамасыз етуімен және тағы басқалармен қамтамасыз етеді;

- кітапханалар (C/C++) операциялық жүйесінің әртүрлі компоненттерімен қолданылады;

- Dalvik Virtual Machine (DVM) – виртуалдық Dalvik машинасы операциялық жүйесінің компоненттері мен қосымшаларының орындалатын ортасын қамтамасыз етеді;

- Core Library – негізгі Android кітапханалары Java (Java API) тіл кітапханасының ядросында қолжетімді көптеген функцияларын, сонымен тағы Android операциялық жүйесінің арнайы функцияларынан тұрады;

- Application Framework (қосымшалардың қаңқасы) – операциялық жүйесіне және кітапханаларының функцияларына интерфейс қосымшаларына ұсынылатын Java класстарының кешені (API).

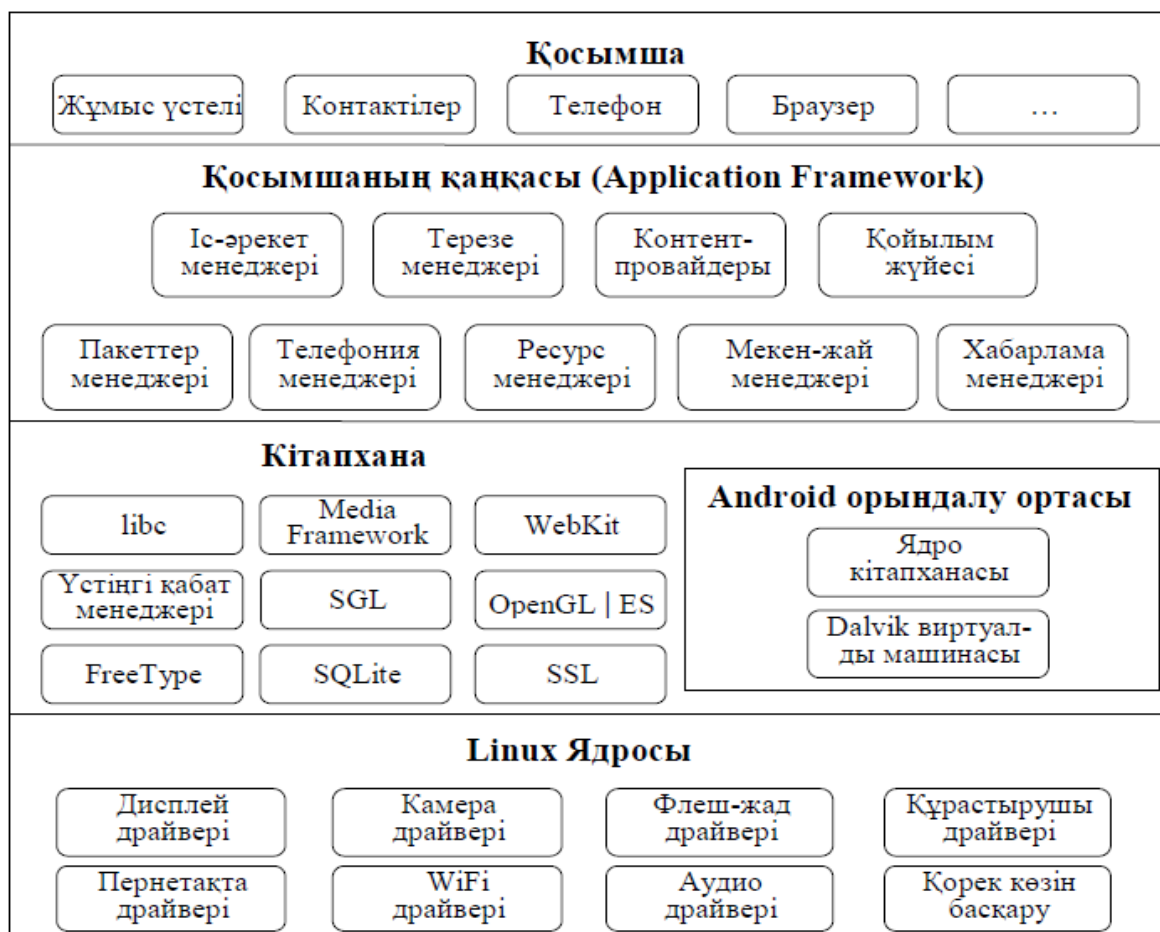
- қолданушы қосымшалары.

Төмендегі 1.2-суретте Android операциялық жүйесінің негізгі компоненттері көрсетілген.

Егер Android компоненттік моделін иерархия түрінде келтіретін болсақ, ең астында, фундаментальды және базалық құрастырушы ретінде операциялық жүйенің ядросы орналасады. Жиі компоненттік модель ретінде тағы бағдарламалық стек деп атайды. Шынында да, бұл анықтама мұнда үйлесімді, себебі әңгіме бірге қорытынды нәтиже алуға бағытталған бағдарламалық өнімдер жиыны жайлы болмақ. Үрдістер бұл модельде кезектесе орындалады және иерархия дәрежелері де кезектесе отырып өзара бір-бірімен байланысады.

Белгілі болғандай, Android - Linux ОЖ бірнеше кесілген ядросына негізделген. Сондықтан бұл деңгейде біз оны көре аламыз. Ол жүйенің функционалдануын қамтамасыз етеді және қауіпсіздік, жадымен басқару, энергожүйе мен процесстерге жауап береді. Сондай-ақ желілік стекті және драйверлер моделін ұсынады. Ядро сонымен қатар абстракциялық деңгей ретінде құрылғылық қамтамасыз ету мен бағдарламалық стек арасында жұмыс атқарады.

Ядродан «жоғарырақ», бағдарламалық қамтамасыз етуші ретінде аралық қабат, кітапханалар (Libraries) жиыны жатыр. Қосымша үшін маңызды базалық функционалды қамтамасыз етуге арналған. Яғни, дәл осы деңгей дайын алгоритмдерді жоғарыда орналасқан деңгейлерге жеткізіп отыруға жауап береді.



1.2 сурет - Android операциялық жүйесінің негізгі компоненттері

Файлдық форматтарды қолдау, кодтауды жүзеге асыру және мәліметтерді декодтау (мысал ретінде мультимедиялық кодектерді алсақ болады), графиктерді сорттау және т.б. C/C++ тілінде дайын кітапханалар және арнайы құрылғылық қамтамасыз етуге жазылған көптеген кітапханалар да қолданылады.

Бірнеше кітапханаларды қарастырайық:

Surface Manager – Android ОЖ-де композитті терезелер менеджері қолданылады. Compoz (Linux) сияқты, бірақ біраз қарапайымдалған. Графиканы сорттау орындау кезінде тікелей дисплей буферіне жіберудің орнына, жүйе келіп жатқан сорттау командаларын кадрленген буферге жібереді. Онда олар басқаларымен бірге жинақталады. Қандайда бір композицияны құрып, кейін қолданушыға экранға шығарады. Бұл жүйеге қызықты эффекттер құруға, терезелердің мөлдірлігі мен бірқалыпты ауысуына мүмкіндік береді.

Media Framework - PacketVideo OpenCORE базасында құрылған дайын кітапхана. Оның көмегімен жүйе аудио және видео контенттерді жазу мен ойнату мүмкіндігіне ие болады. Сондай-ақ статикалық суреттерді шығара алады. Көптеген танымал форматтар қоланылады, тағы да MPEG4, H.264, MP3, AAC, AMR, JPG және PNG.

SQLite – жеңіл және өнімді релятивті ДҚБЖ. Android-та деректер қорымен жұмыстың негізгі қозғалтқышы ретінде қолданылады. Қосымшалармен деректерді сақтау үшін қолданылады.

3D кітапханалар – жоғары оптималданған сорттау кезінде қолданылады. 3D графиктер, мүмкін жағдайда құрылғылық жылдамдықты қолданады. Олардың орындалуы API OpenGL ES 1.0. негізінде тұрғызылады.

OpenGL ES (OpenGL for Embedded Systems) – OpenGL графикалық бағдарламалық интерфейстік жиынша. Орнатылған жүйелерде жұмыс істеуге арналған.

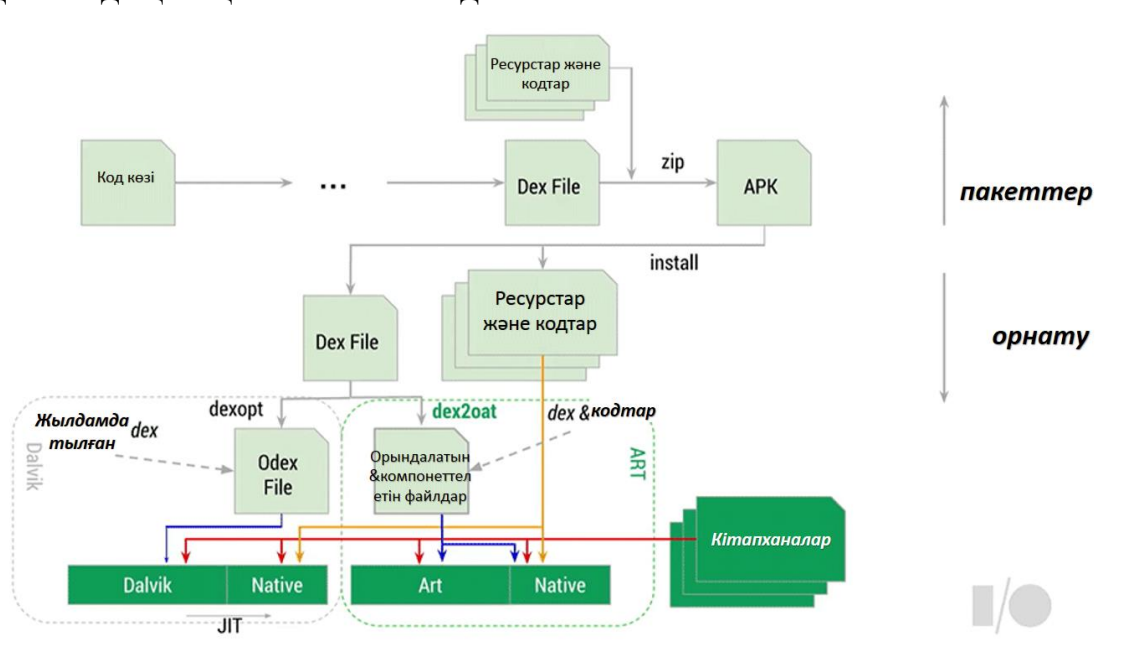
FreeType – Биттік карталармен жұмыс жасауға арналған кітапхана. Шрифттердің расстеризациясы мен олармен операция жүргізуге және жоғары деңгейдегі шрифттер мен текстті көрсету үшін арналған қозғалтқыш.

LibWebCore – танымал әрі жылдам WebKit браузерлік қозғалтқыштың кітапханасы. Сонымен қатар Google Chrome мен Apple Safari сияқты десктоптық браузерлерде қолданылады.

SGL (Skia Graphics Engine) - 2D-графикамен жұмыс жасауға арналған ашық қозғалтқыш. Графикалық кітапхана Google компаниясының өнімі болып табылады және жиі өздерінің бағдарламаларында қолданылады. SSL – аттас криптографиялық протоколды қолдау үшін арналған кітапхана.

libc – C тілінің стандартты кітапханасы. Анығырақ оның BSD реализациясының Linux базасында жұмыс жасауға арналған. Bionic атауын көтеріп жүр.

Android Runtime - ерекшелігі оның құрамындағы ядроның кітапханалар жиыны мен Dalvik виртуалды машинасы болады. Кітапханалар көп бөлігін төмен деңгейдегі Java тілінің ядро кітапханаларына қолжетімді функционалдықты қамтамасыз етеді.



1.3 сурет - Android Runtime жұмыс жасалу жолы

Android ОЖ-де әрбір қосымша Dalvik виртуалды машинасында өзіне тән экземплярда орындалады. Осы арқылы, барлық қосылған процесстер бір-бірінен және операциялық жүйеден босатылған. Жалпы Android Runtime архитектурасы мынадай: қосымшаның жұмыс істеуі виртуалды машинаның қоршауында іске асады. Соның арқасында операциялық жүйенің ядросын басқа құраушыларының тарапынан мүмкін болатын қауіптерден қорғайды. Сондықтан қатесі бар код та, қауіпті бағдарлама да іске қосылғанда Android және оның базасындағы құрылғыларын құрта алмайды. Мұндай қорғау функциясы Android Runtime-нің негізгі қасиеті болып табылады.

Бір деңгей жоғары Application Framework орналасқан, кейде бағдарламалар тірегі деп те аталады. Дәл осы бағдарламалар тірегі арқылы құрастырушылар жүйенің компоненті болып табылатын төменірек деңгейде орналасқан API қолжеткізе алады. Сонымен қоса Framework архитектурасы арқасында кез-келген қосымшаға басқа бағдарламаларда дайын, рұқсат етілген мүмкіндіктерді пайдалануды қамтамасыз етіледі.

Осылайша, Application Framework арқасында, Android ОЖ-дегі бағдарламалар өздеріне қосымша функционалдар қосады, кейін бағдарлама мен жүйе компоненттерінің көп ретті қолдану принципі орнайды. Әрине, қауіпсіздік саясаты негізінде орнайды.

Ескере кететін жағдай, Framework тек қана өзі үшін жазылған кодты орындайды, кітапханалардан ерекшелігі, олар өздігінен орындалады. Тағы да бір ерекшелігі Framework өзінің құрамында үлкен көлемдегі әртүрлі функционалды, әртүрлі міндетті кітапханаларды қосқан, ал кітапханалар логикалық жақын функциялар жиынынан тұрады.

Android бағдарламалық стегінің шыңында бағдарламалар (Applications) деңгейі жатыр. Бұған Android ОЖ орнатылған базалық бағдарламалар жиыны жатады. Мысалы, оған жататын браузер, почталық клиент, SMS жіберуші бағдарлама, карта, күнтізбе, кантакттар менеджері және т.б. Басқа бағдарламалар тізімі құрылғының моделі мен Android нұсқасына қарай өзгеріп отырады. Тағы да бағдарламалардың базалық жиынына Android платформасына жататын барлық бағдарламалар жатады. Қолданушымен орнатылған бағдарламалар да. Android-қа қосымшалар Java тілінде жазылады деп есептеледі, бірақ ескере кететін жай, қосымшаларды C/C++ тілінде де құрастыруға болады (Native Development Kit көмегімен), және Basic-те (Simple көмегімен) және басқа тілдерді қолдану көмегімен. Сондай-ақ өзінің жеке бағдарламанды конструкторлар арқылы да құрастыруға болады, мысалы App Inventor. Бір сөзбен айтқанда мүмкіндіктер көп.

Android ОЖ қосымшаның ядросында қолданылатын API-ді толығымен қолдануға мүмкіндік береді. Барлық қосымшалардың негізі жүйесі мен қызмет жинағы болып табылады.

Контент-провайдерлер (Content Providers) - мәлеметтерді басқаушылар. Бұл бағдарламалар - басқа бағдарламаларға өз жұмысына қолдануына мүмкіндік беру үшін ашып береді.

Ресурстар менеджері (Resource Manager) - функционалдықсыз (кодсыз) ресурстарға қолжетімдікті қамтамасыз етеді. Мысалы, жолдық мәліметтерге, графикаларға, файлдар мен т.б.

Хабарлама менеджері (Notification Manager) - кез келген қосымшаға қолданушы хабарламасын статус қатарында көрсетуге мүмкіндік береді.

Әрекеттер менеджері (Activity Manager) - бағдарламаның өмірлік циклін басқарады. Әрекеттерді тіркеп, сол жайлы мәліметтерді сақтайды. Сондай-ақ навигация жүйесін ұсынады.

Орналасу менеджері (Location Manager) - бағдарламаларға периодты түрде құрылғының географиялық орналасуы жайлы соңғы жаңартылған мәліметтер беріп отырады.

1.3 Android қосымшасының компоненттері

Android операциялық жүйесінің ерекше қыры: бір қосымша өзінің функционалы үшін басқа қосымшаның элементтерін қолдана алады, әрине егер қолжетімділік құқығына ие болса. Мысалы, егер біздің қосымшамызға қозғалмалы суреттердің тізімі керек болса, ал басқа қосымшада жазылып қойылған сәйкес скроллері бар және басқа бағдарламаларға қолжетімді (қолдануға рұқсат) болса, онда оны жай ғана сол функцияны орындау үшін шақыруға болады, - жаңа қосымша үшін жазып отырудың қажеті жоқ. Сонымен қоса біздің қосымшамыздың коды басқа қосымша кодымен араласпайды және компиляцияланбайды. Қажет болған жағдайда, басқа файлдың бірнеше бөлігін ғана іске қосады. Мұндай жұмыс жасау принципі қамтамасыз етілуі үшін, жүйе үрдісті оның кез-келген бір бөлігі керек жағдайда іске қосып және дәл сол бөлік үшін java-объекттердің экземплярын жасау керек. Сондықтан Android қосымшасы, басқа көптеген жүйелерде қалыптасқандай біртұтас кіру нүктесінен тұрмайды. Оның орнына олардың кодтары өз алдына бөлек бүтін компоненттер мен бөліктердің жиыны болып келеді. Ал жүйе қажеттілігіне қарай олардың экземплярларын жасап, қолдана алады.

Компоненттердің төрт түрін бөліп көрсетуге болады. Әр қайсысын бөлек-бөлек қарастырайық:

Activities өз алдына бір үрдіс үшін қолданушы орындай алатын сыртқы қолданушы интерфейсін ұсынады. Егер жай тілмен айтсақ, бұл жай ғана белсенділіктің бірлігі болып табылатын соңғы бір экран немесе қолданушының әрекетіне қарай өзгеріп отыратын кадр. Ары қарай да activity терминін аудармасыз қолданамыз, жалпы есім ретінде. Сөзбе-сөз аудармасы «белсенділік» немесе «әрекет», компоненттің мағынасын тура ашады.

Мысалы, activity менюдің батырмалар тізімін көрсетуі мүмкін, қолданушы ішінен біреуін таңдауы үшін немесе суреттерді атауларымен бірге көрсетуі мүмкін. Немесе басқа мысал – қосымша жылдам хабарлама алмасу үшін контакттар тізімін көрсету үшін бір activity қолдануы мүмкін, басқасы –

таңдалған контакта хабарлама құру үшін, үшіншісі – хабарламалар тізімін көру үшін немесе күйге келтіру үшін және т.б.

Бір қосымша үшін барлық activity-лер бірге жұмыс істейді және қолданушы үшін бір бүтін интерфейс құрайды, бірақ олар өзара бір біріне тәуелсіз. Олардың әрқайсысы жаңа терезенің құрылуымен қамтамасыз ететін Activity базалық класының класс тармағы сияқты жүзеге асырылған, оған қоса бағдарламашы визуалды интерфейс қоса алады.

Қосымша бір ғана activity-ден тұруы мүмкін немесе жоғарыда мысалда көрсетілген мессенджер сияқты бірнешеуінен тұрады. Activity саны қанша болады және қандай болатыны қосымшаның мақсаты мен оның дизайнына байланысты. Әдетте, activity-дің біреуі бірінші болып белгіленеді де, қосымшаны қосқанда бірінші болып сол шығатын болады. Бір activity басқасын іске қоса алады. Сонда, бір activity-ден келесісіне өту алдыңғы activity-дің келесісін шақырғанда орындалады.

Әдетте, әрбір activity бір терезені ұсынады. Терезе толық экранды түрде тұрғызылады, бірақ ол барлық экранды алмауы да мүмкін және басқа терезелер үстінде де тұра алады. Activity сонымен қатар қосымша терезе жасауы мүмкін – мысалы, activity-мен жұмыс жасау процесінде кенеттен шығатын диалогтық терезе немесе қандай да бір маңызды опцияны таңдау кезінде мәлімет беретін терезе.

Терезенің визуалды мазмұны, визуалды компоненттердің иерархиясының - нысандарын, туынды View негіздік класынан құралады. Әрбір компонент тек терезе ішіндегі тікбұрышты кеңістікті ұсынады. Аталық компоненттер құрамында кішігірім компоненттер бар және аталық компоненттер олардың орналасуын ұйымдастырады. Компоненттер иерархиясын ағаш ретінде қарастыруға болады. Ең төменгі бөлікте орналасқан элементтер («жапырақтар»), кішірек бөліктері болмағандықтан тіктөртбұрышты облыс құрып, қолданушының осы облыстағы әрекетін күтеді. Осылай қолданушымен интерактивтік байланыс орнайды. Мысалы, ондай үрдіс экранда кішкентай иконка түрінде тұруы мүмкін және қолданушы оны басқанда қандай да бір әрекет орындалады. Android операциялық жүйесінде дайын визуалдық компоненттер жиыны бар. Олар құрастырушылар үшін қолдануға қолжетімді. Жиынның құрамына түймелер (кнопка), тексттік облыстар, жылжыту жолдары, меню, ауыстыру жалаушалары және т.б.

Терезеге мұндай иерархия орнату үшін Activity.setContentView әдісін шақыру қажет. Әдістің параметрі болып View класының экземпляры болып табылады, ол иерархия түбінде орналасады.

Services (сервистер) фондық режимде жұмыс істейтін компоненттерді ұсынады. Ол, әдетте, ұзақ операциялар немесе қиын процесстердің жұмыс істеуін қамтамасыз етуілуі үшін қажетті, бірақ жалпы жағдайда бұл жай ғана «қосымша» фокуста емес кезде функционалданатын режим. Үрдіске мысал ретінде әуенді тыңдау бола алады. Яғни қолданушы басқа нәрсе істеп жатқанда соңғы қосымшаны тоқтатпай, мәліметтер алмасуын айта аламыз. Сервис өз алдына ешқандай қолданушы интерфейсін ұсынбайды, яғни

қолданушымен байланыспаған, тек қосылады, басқарылады және басқа компоненттермен байланысқан. Мысалы, activity. Тағы жүйемен бірге қосыла алады.

Content providers. Бұл компонент қосымша басқа қосымшаға беретін деректер қорын басқарады. Бұл деректер қоры қосымша қолжеткізе алатын файлдық жүйеде, SQLite деректер қорында, желіде, немесе басқа кез-келген тұрақты жерде сақталуы мүмкін. Content provider арқасында басқа қосымша мәліметтер сұрай алады. Қолжетімділік құқығы бар жағдайда өзгерте де алады. Мысалы, Android жүйесі қолданушы контакттар жайлы мәліметін өзгертетін content provider құрамында бар. Ол кез-келген қосымшаға тиісті құқықтары бар. Нақты адам жайлы информация жаза алады. Оқып және өзгерту үшін компонент құрамын шақыра алады.

Жалпы жағдайда бағдарламамен қолданылатын және басқаларға жабық мәліметтерді content provider арқылы оқып, жазу үшін қолдана аламыз. Мысалы, NotePad қосымшасы мұндай компонентті жазбаларды сақтау үшін қолданады.

Бұл компонент ContentProvider классы ретінде жазылған. Және, басқа қосымшалар мәліметтермен операция жасау үшін оларға стандартты API жиынын орнату керек.

Broadcast receivers. Бұл компонент жалпы-жүйелік хабарламаларды тарату, бақылау және әрекетке реакция беруге жауап береді. Көптеген хабарламалар жүйеден келеді. Мысалы, батарея қуаты аз екендігі немесе экран өшірулі екендігі жайлы хабарламалар. Қосымшалар сонымен қоса мынадай хабарлар жіберуі мүмкін, мысалы, информация құрылғыға жүктелгендігі жайлы және қолдануға дайын екендігі жайлы мәліметтер. Сервистер сияқты, broadcast receivers ешқандай қолданушы интерфейсін ұсынбайды, бірақ ол күй жолында хабарламалар жазуы мүмкін, қолданушыны қандай-да бір жағдай болғандығы жайлы ескерту үшін. Broadcast receivers минималды жұмыс көлемін орындау үшін басқа компоненттермен әрекеттеседі. Осылай ол қандай-да бір әрекетке байланысты сервистерді жұмысқа қоса алады.

1.4 Android платформасын қолданудағы артықшылықтары

Қазіргі платформаның негізгі құндылықтарына келесілерді жатқызуға болады:

- кейбір шолушылар android өзінің бәсекелестері Apple IOS арасында жақсы жағынан көрсеткенін белгілейді. Осындай ерекшеліктер қатарында: веб-сёрфинг, google inc сервистермен және басқалармен интеграциялануы. Android IOS-пен салыстырғанда ашық платформа болып саналады. Сондықтан көптеген функцияларды іске асыруға мүмкіндік береді;

- IOS және windows phone 7 салыстырғанда android bluetooth - стегі толық іске асырылған. Ол файлдарлы қабылдауына және жіберуіне мүмкіндік береді. желіге (ран қызметінің) қатынау нүктелері режимі бір ранг топтық bluetooth (gn қызметі) желісі арқылы ftp-сервердің іске асыруы мүмкіндігі бар;

- android аппараттарында microsd кардридер бар. Ол компьютерден телефонға файлдарды тез көшіруге мүмкіндік береді. IOS және windows phone 7 синхрондау бағдарламалары (itunes және zune) болмаса, телефоннан телефонға қандай да бір файлдарлы тікелей жіберу мүмкін емес. Android телефондары usb mass storage device (өндірушінің қарастыруымен 3.0 нұсқаларынан жоғары) көмегімен немесе media transfer protocol көмегімен жад картасының файл жүйесін экспорттайды;

- «тексерілмеген дереккөздер» (мысалы, жад картасынан) бағдарламаның орнатылуына бастапқы тыйым салуына қарамастан, бұл ерекшелік аппарат баптауында штаттық құралдарымен өшіріледі. Ол ғаламторға қосылмай-ақ (мысалы, wi-fi-рұқсат нүктесі жоқ және әдетте қымбат тұратын мобильді ғаламторға ақшасын үнемдейтін пайдаланушылар) телефонға және планшеттерге бағдарламаларды орнатуға мүмкіндік береді. Сонымен қатар барлық қалаушыларға android үшін қосымшаларын тегін жазуға және өзіннің аппаратында тестілеу жүргізуіне мүмкіндік береді;

- android arm, mips, x86 тәрізді әртүрлі аппараттық платформалары үшін қолжетімді;

1.5 Android платформасына қосымша әзірлеу және әзірлеу құрал-саймандары

Android ортасының платформасын орнату үшін төмендегі программалар қажет:

- JDK 7 – Java Development Kit;
- Eclipse IDE – программалау ортасы;
- Android SDK;
- Android Development Tools (ADT).

Әрине кез-келген платформаның программалау практикасына көшу құрастыру үшін қолда бар құрал-саймандар арсеналын қарастырудан басталады. Құрал-саймандар таңдалғаннан кейін ғана (қалауы мен таңдауына сүйене отырып), қолдану ережесіне сәйкес жұмысқа дайындалғаннан соң, «үлкен» бағдарламалау мен икемдерді шыңдау басталады.

Android құралдарын қарастырмай тұрып, бірнеше жалпы түсініктер енгізуіміз керек, себебі бұл этапта бірнеше рет соқтығысуымыз мүмкін. Бірінші базалық түсінік – бұл SDK немесе Software Development Kit («devkit»). Бұл құрастырушының құралдар жиыны, программисттерге берілген архитектураға бағдарлама құруға мүмкіндік береді – анықталған бағдарламалар пакетіне, аппараттық немесе бағдарламалық платформаға, операциялық немесе компьютерлік жүйеге. SDK құрастыруға, әдетте, өзіне бағдарламаны дұрыстау құралдары мен анықтамалық материалдарды (кодтар мысалы, белгілер мен т.б.) қосатын кең құралдарын жиынын ұсынады. SDK жиі белгіленген платформада ұсынылады және онда оның ерекшеліктері ескеріледі. Мұндай инструментарий (Android-тағыдай) тегін тарала алады. Бұл

платформаға басқа құрастырушылардың әртүрлі бағдарламаларымен иемденуіне мүмкіндік береді.

Тағы да бір түсінік, әңгіме құрастыру құралы болған кезде IDE (Integrated development environment) кездеседі – интегралданған құрастыру ортасы. Ол бағдарламалық құралдар жиынын ұсынады, бағдарламалау процесі үшін ыңғайлы әрі оңай орта құру арқылы программисттің өнімділігін арттыруға негізделген. Бұл күрделі бағдарламалық комплекс өзіне текстік редакторды қоса алады (жиі синтаксисті дұрыстау мен көмектер), компилятор немесе интерпретатор (кейде бірнеше тілдер қолдайды), интегралдандырылған дұрыстаушы мен жинауды автоматизациялайтын құралдар да жатады. Жиі басқа бағдарламалармен интегралдануы мүмкіндігі қамтамасыз етіледі, мысалы, проектрлеуді басқару немесе нұсқаны тексеру құралдары. Сонымен қоса, жиі IDE болашақ бағдарламаны тез құрастыру құралдары мен графикалық интерфейсін визуалдық редактрлеу құралдары бар, тағы класстар иерархиясының диаграммасы түріндегі құрал, класстар браузері, объекттер инспекторы немесе ресурстар менеждері бар. Осылайша, интегралданған құрастыру ортасы бағдарламалық қамтамасыз ету жылдамдығын арттыру үшін әртүрлі құралдарды бір бағдарламалық комплекске жинауға негізделген. Бірақ іс жүзінде бағдарламалау үшін қандайда бір IDE қолдану қажетті емес, егер құралдарды байланыстырғысы келмесе.

Android платформасына қосымша құру құралдар группасымен байланысты, бұлар Android SDK деп аталады. Тағы да Java SE (JDK)-да бағдарламалау үшін инструментарий қажет және интегралданған орта. Соңғысының орнына Eclipse IDE қолдану қабылданған, бізде солай жасаймыз, бірақ ескерте кететін жағдай бағдарламаны құру жай текстік редакторда да істеуге болады немесе басқа IDE-лерде, тағы да құралдарды шақыру үшін скрипттер немесе командалық жолды қолдана алады. Бірақ Eclipse-те құрастыру тиімді әдіс болып табылады, себебі, біріншіден, бұл орта құралдарды тікелей шақыра алады, екіншіден, оған арналған арнайы плагин бар, Android Development Toolkit (ADT), бұл Eclipse IDE көмегімен Android-та бағдарлама құруға бұрынғыдан ыңғайлы өтуді қамтамасыз етеді.

Қортындылай келе, Android SDK – JDK 5-ші нұсқасын талап етеді немесе жаңарағын, тағы Eclipse 3.3 нұсқасын немесе жаңарағын. Құрастырушы сайтында SDK, Java мен Eclipse нұсқалары Windows, Linux пен Mac OS операциялық жүйесі үшін бар, ал SDK әрқайсысы үшін эмулятор бар. Жалпылай айтқанда, Android бағдарламалары виртуалды машинада орындалады, және операциялық жүйелердің ішінде ешқандай ОЖ мұндай мүмкіндік бермейді.

2 Android платформасында мейрамхана үшін «Мәзір» мобильді қосымшаны UML тілінде жобалау

2.1 Мобильді қосымшаның әзірлеу ортасын және технологиясын таңдау

Мобильді қосымшаны әзірлеудің мақсаты мейрамхана саласындағы бірқатар проблемаларды шешу болып табылады. Айта кету керек, ХХІ ғасырдың озық технологиялары қызметтің барлық салаларында белсенді түрде қолданыла түсуде. Жаңа шешімдер мен жаңа технологияларды пайдалана отырып, әлемге олардың қаншалықты перспективалы және болашақта дамитынын дәлелдеуге болады.

Бағдарламалық өнімді әзірлеу үшін әзірлеу ортасын және технологияны таңдау қажет, сонымен қатар интерфейстер мен мобильді қосымшалардың функционалдарын жобалау қажет. Интерфейс дегеніміз ұялы қосымшаның қабығы, атап айтқанда клиенттік бөлікті жобалау.

Технологияларды таңдау барлық болашақ қосымшаларды анықтайды. «Ол нативті, гибриді немесе кроссплатфорлы болады ма? Қандай жүктемелер серверге төтеп беруі керек? Интеграцияның қандай мүмкіндіктері қалануы тиіс?» -, осы және басқа да көптеген мәселелер осы кезеңде шешіледі.

Көптеген компаниялар үшін бағдарламалық қамтамасыз етудің стратегиялық маңызы өсіп отырғандығына байланысты, индустрия бағдарламалық қамтамасыз етудің өндірісін автоматтандыру әдісін, оның сапасын көтеру, сондай-ақ оны нарыққа шығарудағы құны мен шығару уақытын төмендету әдістерін іздестіруде. Бұл әдістердің негізін құраушы технологиялар, көрсеткіштік бағдарламаландыру, үлгілерді (pattern) және инструменттік (құрал-жабдық) ортаны (framework) пайдалану құрайды.

Мобильді қосымша жүйесінің бағдарламасын құру үшін ең алдымен міндетті түрде қалай құру керектігін жобалап алуымыз керек. Мейрамхана онымен қалай қолданып, жұмыс істеу керектігін бастапқы кезде білмейді. Сол себепті бағдарламаны жасамастан бұрын модель құруымыз керек. Ал модельді құру арнайы бағдарлама көмегімен орындалады. Ол UML унифицирленген модельдеу тілінде сипатталады.

Сонымен, модельдеу - жүйені түсіну үшін қажет. Бұл кезде жалғыз модель жеткіліксіз болады. Керісінше, кез келген тривиалды емес жүйелерді түсіну үшін өзара байланысты модельдердің үлкен санын жасауға тура келеді. Бұл, әрине, жүйенің архитектурасын көрсетуді сипаттау жағынан алып қарағанда, қолданысқа жіберетін бағдарламалық жүйені дайындау циклін түрлі көзқарасты ескеріп сипаттауға болатын тілдің қажет екендігін білдіреді. Осы қажеттіліктен туындаған UML модельдеудің бірыңғайланған тілі бағдарламалық қамтамасыз етудің «сызбаларын» құру үшін стандартты құрал болып табылады. UML-дың көмегімен бағдарламалық жүйелердің артефактілерін құжаттауға, құрастыруға, арнайыландыруға және көрсетуге болады.

Модельдеудің бірыңғайланған тілі (Unified Modeling Language – UML) – бұл бағдарламалық жүйелерді ерекшелендіру, бұрыштама қою, конструкциялау және құжаттамалау, сондай-ақ модельдер бизнесі мен өзге де бағдарламалық емес жүйелердің тілі болып табылатынын нақтылай түселік. UML бұдан бұрын да үлкен және күрделі жүйелерді модельдеу кезінде ойдағыдай қолданылып жүрген инженерлік әдіс-тәсілдердің бірлестігін көрсетеді.

UML кез келген жүйені модельдеу үшін жарамды: кәсіпорын масштабындағы ақпараттық жүйелерден таралған Web – қосымшаларға дейін және нақты уақыттың кіріктірілген жүйелері үшін де жарамды болады. Бұл өте мәнерлі тіл, ол жүйені жасау мен кейінгі жолды кеңінен ашып көрсетуге қатысы бар барлық көзқарас жағынан қарап шығуға мүмкіндік береді. Мәнерлі мүмкіндіктерінің молдығына қарамастан, бұл тіл түсіну және пайдалану үшін өте оңай. UML-ды зерттеуді оның концептуалды моделінен бастаған жөн, оның үш негізгі элементі болады:

- базалық құрылыс блоктары;
- блоктардың өзара үйлесімділігін анықтаушы ережелер;
- тілдің кейбір жалпы механизмдері.

UML модельденетін нақтылыққа байланысты болмаса да, модельдеу процесі итеративті және қадамдық болған жағдайда қолданған ыңғайлы. Ал жүйенің өзінің анық көрсетілген архитектурасы бар.

UML – бұл көрнекілеу, спецификациялау, құрастыру және бағдарламалық жүйелердің артефактілерін құжаттау тілі.

Тіл, өзінің құрамындағы сөздерді құрастыруға және мағыналы құрылымдар алуға мүмкіндігін беретін, сөздік пен ережелерден тұрады. Модельдеу тілінде сөздік пен ережелер жүйелердің концептуалды және физикалық түсінігіне бағдарланады. UML модельдеу тілі бағдарламалық қамтамасыз ету «сызбасын» құрастыру үшін стандартты құрал болып табылады.

UML тілінің сөздігі мен ережесі жақсы анықталған модельді қалай құрып және қалай оқуды түсіндіреді, бірақ та қандай жағдайда қандай модельдерді құру керектігі туралы хабарламайды.

Бағдарламашылардың көпшілігінің көзқарасы бойынша жобаны жүзеге асырудағы ойлану ол үшін код жазуға эквивалентті болып келеді. Сіз ойлайсыз – сіз код жасап жатырмын деп. Шын мәнісінде, кейбір заттар бағдарламалаудың қандай да тіліндегі тікелей кодтауда жақсы көрсетіледі, өйткені бағдарлама мәтіні – бұл алгоритмдер мен өрнектерді жазу үшін ең қарапайым және қысқа жол.

Бірақ та осындай жағдайда бағдарламашы формалды болмаса да модельдеумен айналысады. Ол идеяларын тақтаға немесе қағазға жаза алады делік. Алайда мұндай тәсіл жақсылық әкелмейді. Біріншіден, концептуальды модельге байланысты пікір алмасу пікірталасқа қатысушылардың барлығы бір тілде ғана сөйлеген кезде ғана мүмкін болады. Компаниялар жобаны жасау кезінде өзінің тілін жасауына тура келеді, оны жанадан келген адамның түсінуі

оңай емес. Екіншіден, бағдарламалаудың мәтіндік тілінің шегінен шығатын модельсіз бағдарламалық жүйелердің белгілі бір аспектілері туралы түсінік алуға болмайды. Өйткені, кластардың иерархиясында олардың неге арналғанын, әрине, әрбір кластың кодын мұқият үйренген кезде түсінуге болады, бірақ бүкіл құрылымды бірден және тұтас қабылдау мүмкін емес. Үшіншіден, егер автор өз ойындағы моделін айқын түрде көрсетпесе, онда ол жұмыстан ауысқанда оның ойы да жоғалады.

UML-ды пайдалану үшінші мәселені шешуге мүмкіндік береді: бұл модель тілдесуді оңайлатады.

Жүйенің кейбір ерекшеліктерін бәрінен бұрын мәтін түрінде модельдеуге болады, екінші біреулерін – графикалық модельдеуге болады. Шын мәнісінде барлық қызықты жүйелерде бір ғана бағдарламалау тілінің көмегімен беру мүмкін емес құрылымдар бар болады. Мұндайда, UML – екінші белгіленген мәселелерді шешуге мүмкіндік беретін графикалық тіл.

UML – бұл жай ғана графикалық символдар жиыны емес. Олардың әрқайсысының артында жақсы анықталған семантика тұр. Бұл бір дайындаушы жазған модельді екінші біреу бізмәнді түсіндіруі мүмкін, тіпті аспаптық бағдарламамен интерпретациялануы мүмкін. UML талдау, жобалау және жүзеге асыруға қатысты барлық мәнді шешімдердің өзгешіліктерін анықтауға мүмкіндік береді, ол бағдарламалық қамтамасыз ету жүйесін жасау мен кеңінен тарату процесінде де қабылданылуы тиіс.

Әзірлеу ортасы – бағдарламалық жасақтаманы немесе қосымшаны әзірлеу үшін қолданатын бағдарламалық құралдар жиынтығын айтады. Қарапайым өңдеу ортасы мәтін редакторын, компиляторды немесе интерпретаторды, құрастыруды автоматтандыру құралдарын және баптауды қамтиды. Бұл компоненттер бірыңғай бағдарламалық кешенге жиналған кезде интеграцияланған әзірлеу ортасы туралы айтады (IDE). Мұндай орта бір бағдарламамен ұсынылған, одан шықпастан барлық әзірлемелер циклін өндіруге болады. Кешен құрамына жоғарыда аталғандардан басқа, жобаларды басқару құралдары, нұсқаларды басқару жүйесі, пайдаланушының интерфейсін әзірлеуді оңайлатуға арналған әртүрлі құралдар, стандартты дайындамалар, стандартты есептерді әзірлеуді жеңілдететін және т.б. кіреді.

UML - бағдарламалық қамтамаларға сызбалар (модельдер) құрудың стандартты құралы болып табылады. Негізінен UML тілі тіл теңдіктерінен және кез келген артефактыларды келістіру арқылы мағынасы бар құрылымдар алуға мүмкіндік беретін ережелерден тұрады. Тілдің сөздігі мен ережелері үлгілерді қалай құруға және оқуға болатынын түсіндіреді.

Артефакт – бағдарламалық жүйедегі кез келген жасанды құрылған элемент. Оны өнім десек те болады. Ол бағдарламалық құралды өңдеу процесі кезінде құрылады және қолданылады.

Жақсы ұйымдастырылған процесс жұмысты бағалау және жобаны құрылымымен басқару үшін қандай артефактылар керек екендігін, оларды құруға қандай ресурс қажеттігін, әрі қалай пайдалану керектігін ескеріп отырады. Бұл тіл тек графикалық таңбалардан ғана емес, олардың

әрқайсысының атында анықталған мағына бар екенін білдіреді. Ерекшелеу дәл толық үлгілердің құрастыруын білдіреді. Ерекшеленудің арқасында кез келген программалық қамтаманың жүйесіз өңдеу процесіне қатысты талдау, жобалау және жүзеге асыру кезінде маңызды шешімдерге ерекшеленуге мүмкіндік береді.

UML-дің көмегімен Delphi, Java, C++, Visual Basic тілдеріне реалық деректер қорынан немесе объектіге бағытталған тілдерге ауысуға мүмкіндіктер жасалған.

Бағдарламалық құралдар жасалмақтан бұрын келесі артефактыларды құрастыру керек:

- жүйеге қойылатын талаптар;
- жүйе архитектурасы;
- жүйе жобасы;
- бастапқы коды;
- жобалы жоспарлардың талапқа сәйкестігі;
- жүйенің тестіленуі;
- жүйе прототипі;
- жүйе версиялары және тағы басқа.

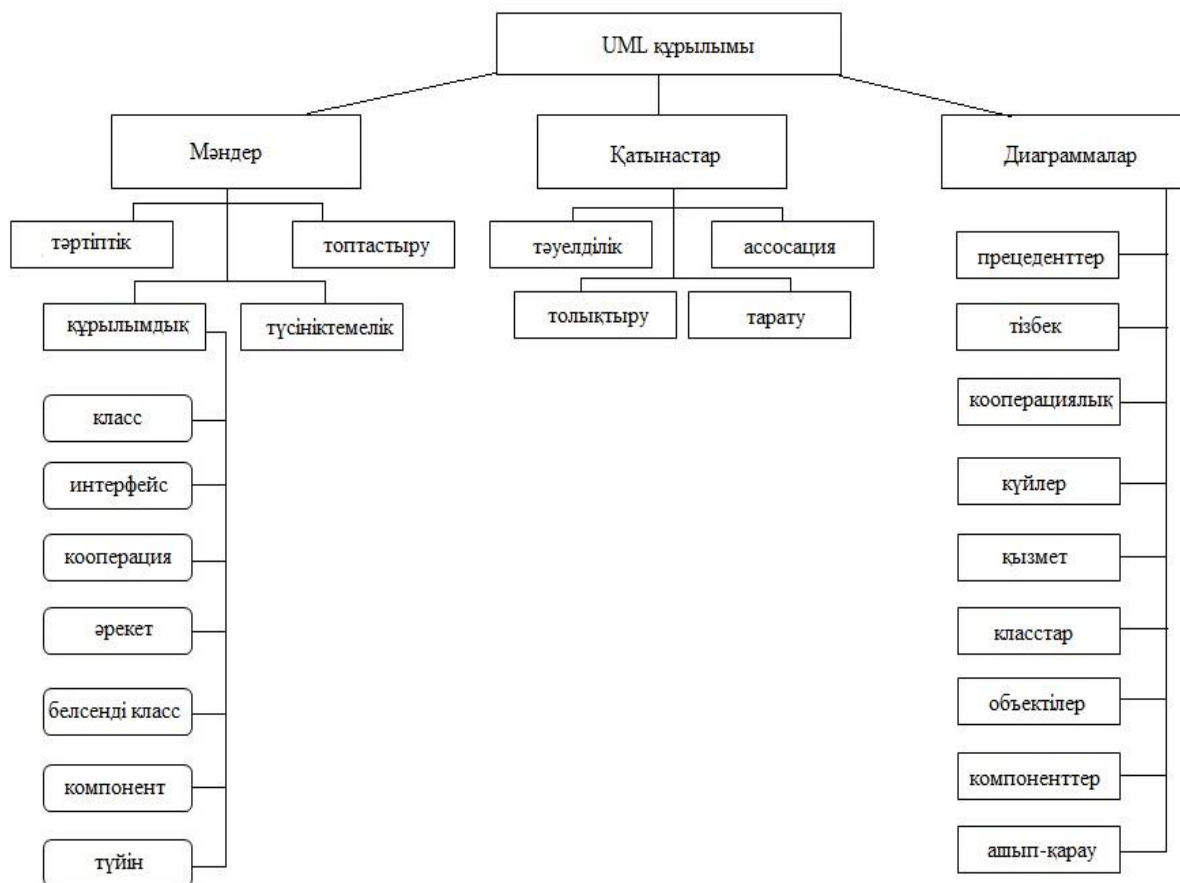
UML құрылым бөліктері 2.1-суретінде көрсетілген.

Мәндер – модельдің негізгі элементтері болып табылатын абстракциялары.

Қатынастар – осы абстракциялық мәндерді байланыстыратын байланыстар.

Диаграммалар ережеге және тәртіпке байланысты мәндерді бір топқа жинақтайды.

Скрипттерді жазу үшін объектіге бағытталған Visual Code Studio тілі таңдалды. Visual Code Studio тілі C тіліне ұқсас синтаксисі бар тілдер отбасына жатады, оның синтаксисі C++ және Java-ға ең жақын. Тіл статикалық типизацияға ие, полиморфизмді, операторлардың шамадан тыс жүктелуін (оның ішінде айқын және айқын емес типті келтіру операторлары), делегаттар, атрибуттар, оқиғалар, қасиеттері, жалпыланған типтері мен әдістері, итераторлар, тұйықталуды қолдайтын анонимдік функциялар, LINQ, ерекшеліктер, XML форматындағы түсініктемелерін қолдайды. C++, Pascal, Модульдер, Smalltalk және, әсіресе, Java – C# тілдерінен көп нәрсені қабылдай отырып, оларды пайдалану тәжірибесіне сүйене отырып, бағдарламалық жүйелерді жасау кезінде өзін проблемалы ретінде көрсеткен кейбір модельдерді жоққа шығарады, мысалы, C# C++ және кейбір басқа тілдерге қарағанда, көптеген кластардың мұрагерлігін қолдамайды (сонымен қатар интерфейстердің көптеген мұрагерлеріне жол беріледі).



2.1 сурет – UML құрылым бөліктері

Visual Code Studio CLR үшін қолданбалы деңгейдегі бағдарламалау тілі ретінде әзірленген және ең алдымен CLR мүмкіндіктеріне байланысты. Бұл, ең алдымен, BCL көрсететін C# типті жүйеге қатысты. Тілдің қандай да бір мәнерлі ерекшеліктерінің болуы немесе болмауы нақты тілдік ерекшелік тиісті CLR конструкцияларына таратылуы мүмкін бе? Осылайша, 1.1-2.0 нұсқасынан CLR дамуымен с өзі де айтарлықтай байытылды; мұндай өзара іс-қимылды одан әрі де күту керек (алайда, бұл заңдылық платформаның кеңеюіне сүйенбейтін тілдің кеңеюін білдіретін C# 3.0 шығуымен бұзылды). CLR C# және өзге де .NET-ке сүйеніп жазылған программаларға, басқалардан артық «классикалық мүмкіншіліктер» береді.

Бұл тілдің артықшылығы барлық программалауға объектілі-бағытталған тәсілді қолданады. Бұл дегеніміз, пән саласы негізінде абстрактілі конструкцияларды сипаттау, содан кейін олардың арасында өзара іс-қимылды жүзеге асыру қажет. Сондай-ақ, тілде артықшылығы мол, ол программистің ауыр өмірін әлсіз етеді. Мыңдаған код жолдарын жазудың орнына, дайын құрылымды ғана пайдаланады, ал компилятор барлық жұмысты өзі жасайды. Бірақ кейбір мұндай құрылымдар өнімділік тұрғысынан ең оңтайлы емес. Бірақ мұның бәрі кодтың ыңғайлы болуы және жоғары даму жылдамдығы есебінен жабылады. Мұның бәрі .NET Framework платформасында жұмыс істейді. Көптеген талғампаз адамдар үшін бұл бағдарлама іске қосылуы үшін

компьютерге орнату қажет, бірақ бұл айтарлықтай тереңірек. C# тілінде жазған код аралық тілге (JIT) таратылады, ол өз кезегінде, қолданба (JIT) орындау кезінде компьютеріңізде машина кодына айналады. Ал мәні, әр түрлі тілдерде бір жобаны қайта оқытуға тура келмейді. Бірақ оны іс жүзінде қолдануын жоқ. Аралық кодтың соңғы компиляциясы нақты машинада тірі орындалғандықтан, онда процессордың арнайы командаларын пайдалану арқылы өнімділіктің артуы мүмкін.

Бұл тілдің кемшілігі .NET JIT-компиляция тұжырымдамасын пайдаланады. Бұл бағдарлама қажет болған жағдайда тікелей бағдарлама жұмыс кезінде машиналық кодтарға жинақталатынын білдіреді. Бір жағынан, бұл әрине күшті, бірақ екінші жағынан, іске қосу кезінде өте ауыр тежегіш болуы мүмкін. C# жаппай таралған тіл емес. Көптеген бағдарламашылар Delphi немесе PHP-ден басқа ештеңе көрген емес, шағын қалаларда жұмыс іздеуге өте елеулі шектеулер қояды коммерциялық Enterprise саласында шоғырланған. Сонымен қатар, C# ең алдымен Windows-пен байланысты. Жақын болашақта бір нәрсе өзгереді және Windows бәрі де нарықта үстем болады, бірақ әлі де аз тәуекел, әсіресе Windows 10 жаңартулары бар соңғы жалған екенін ескере отырып, қалады.

Жоғарыда айтылған құралдар мобильді қосымшаны іске асыруға көмектеседі.

2.2 Мобильді қосымшаны жобалау

Мобильді қосымшалардың мәні – пайдаланушылардың міндеттерін шешу. Жоғарыда айтылғандай, мобильді қосымшалар кез келген салада қолданылады. Білім беру саласында мобильді қосымша, оқушылар мен студенттер қажетті ақпаратты виртуалды объектіі smart-құрылғылардан қарап түсіне алады.

Интернет пен смартфондарға қол жеткізу мүмкіндігі мобильді қосымшаларға деген қызығушылық жыл сайын артып келеді. Әзірлемелердің тақырыбы мен функционалы таңданбайды. Ірі кәсіпорындар мен ұйымдар бизнесті кеңейту үшін мобильді қосымшалардың көмегіне жүгінеді.

Мобильді қосымшаны әзірлеу кезінде MVP қандай функционалды құрайтыны (ең аз өміршең өнім), болашақ кеңейту үшін сәулетті қалай ойластыру, қандай технологияларды таңдау және сонымен қатар, жобаны іске асырудың еңбек шығыны мен мерзімін анықтау туралы шешім қабылдау қажет болатын өзгермелі сәт бар.

Тапсырманың қойылымына келетін болсақ, бұл мобильді қосымша мейрамхана саласына арналған.

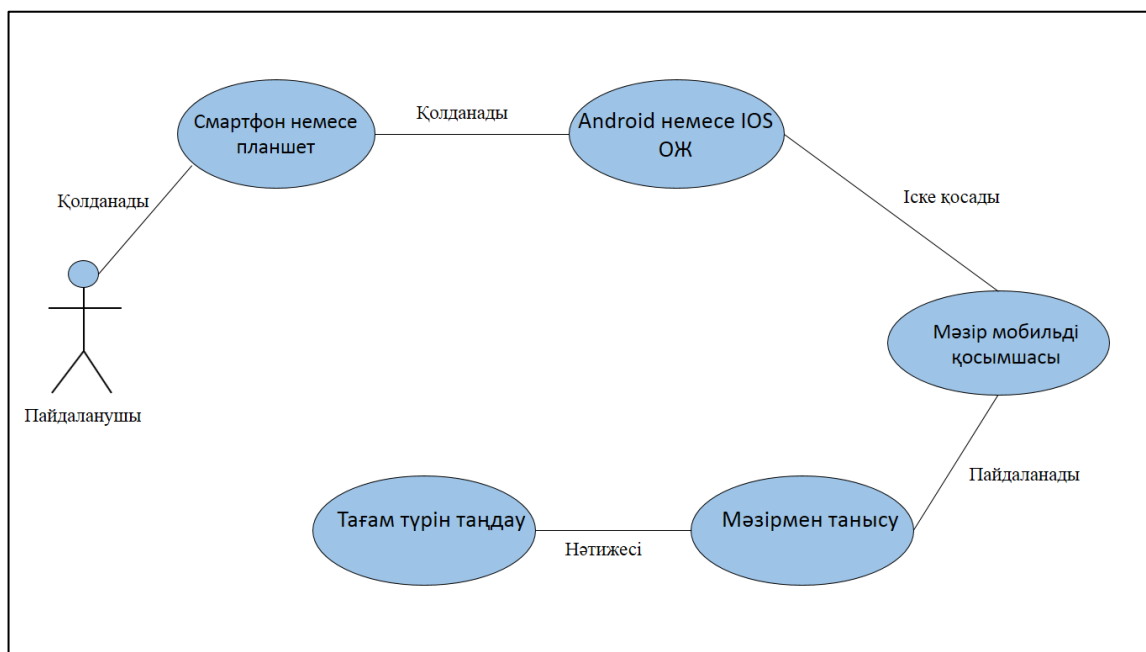
Пайдаланушы мен қосымша арасындағы типтік өзара іс-қимылды сипаттаудан, оның жұмыс істеу процесін сипаттаудан тұратын прецеденттер диаграммасынан бастайық.

Прецеденттер бағдарламаға қойылатын функционалдық талаптарды түсіну үшін құнды құрал болып табылады. Прецеденттердің бірінші нұсқасы

жобаны орындаудың ерте сатысында жасалуы тиіс. Прецеденттердің егжей-тегжейлі нұсқалары осы прецедентті іске асырар алдында тікелей пайда болуы тиіс.

Прецеденттер диаграммасы прецеденттер және актерлар арасындағы қатынастарды көрсетеді. Бұл диаграмма қолданушы прецеденттер көзқарасымен алғанда жүйенің классикалық түріне жатады. Негізінен жүйенің орындалу тәртібін модельдеу және ұйымдастыруды орындайды.

2.2-суретте пайдаланушы үшін прецеденттер диаграммасы көрсетілген. Яғни пайдаланушы келесі әрекеттерді пайдалана алады:

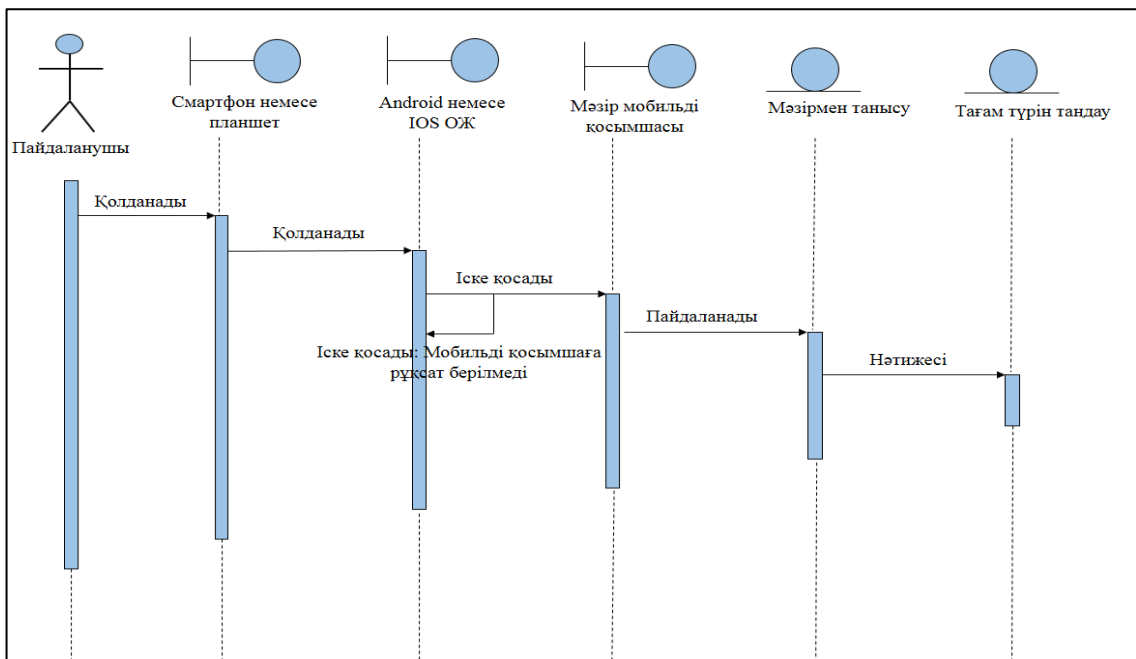


2.2 сурет – Пайдаланушы үшін прецеденттер диаграммасы

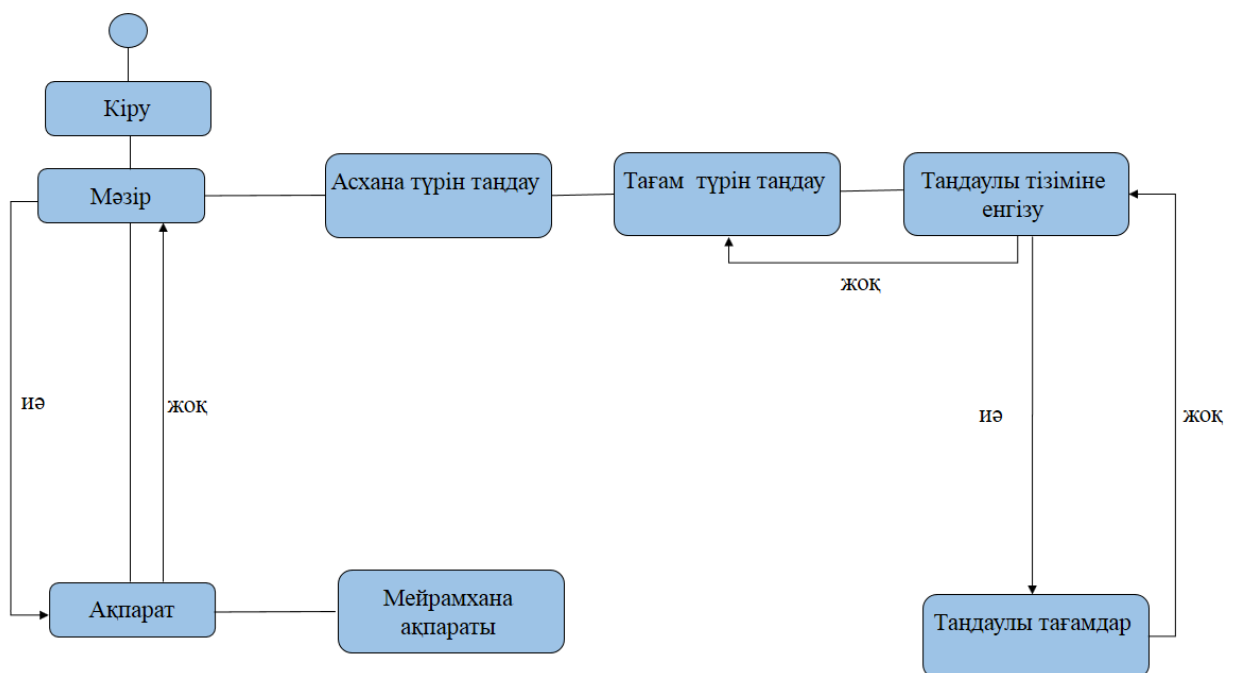
2.3-сурет бойынша пайдаланушы (User) кез-келген мобилдік құрылғыны немесе Android, IOS ОЖ бар құрылғыны пайдалана отырып, мейрамхана мәзірімен таныса алады.

Тізбектілік (Sequence) диаграммаларын бір прецедент шеңберінде бірнеше объектілердің әрекетін қарау қажет болған кезде қолдану керек. Кезектілік диаграммалары объектілердің өзара әрекеттесуін ұсыну үшін жақсы, бірақ әрекетті дәл анықтау үшін өте қолайлы емес.

Тізбек диаграммасы объектілер арасындағы байланыстарды көрсетеді. Кез келген актер объектілермен хабарламалар арқылы байланыса алады. Жүйенің динамикалық түріне жатады. Тізбектеп орындауды көрсетеді.

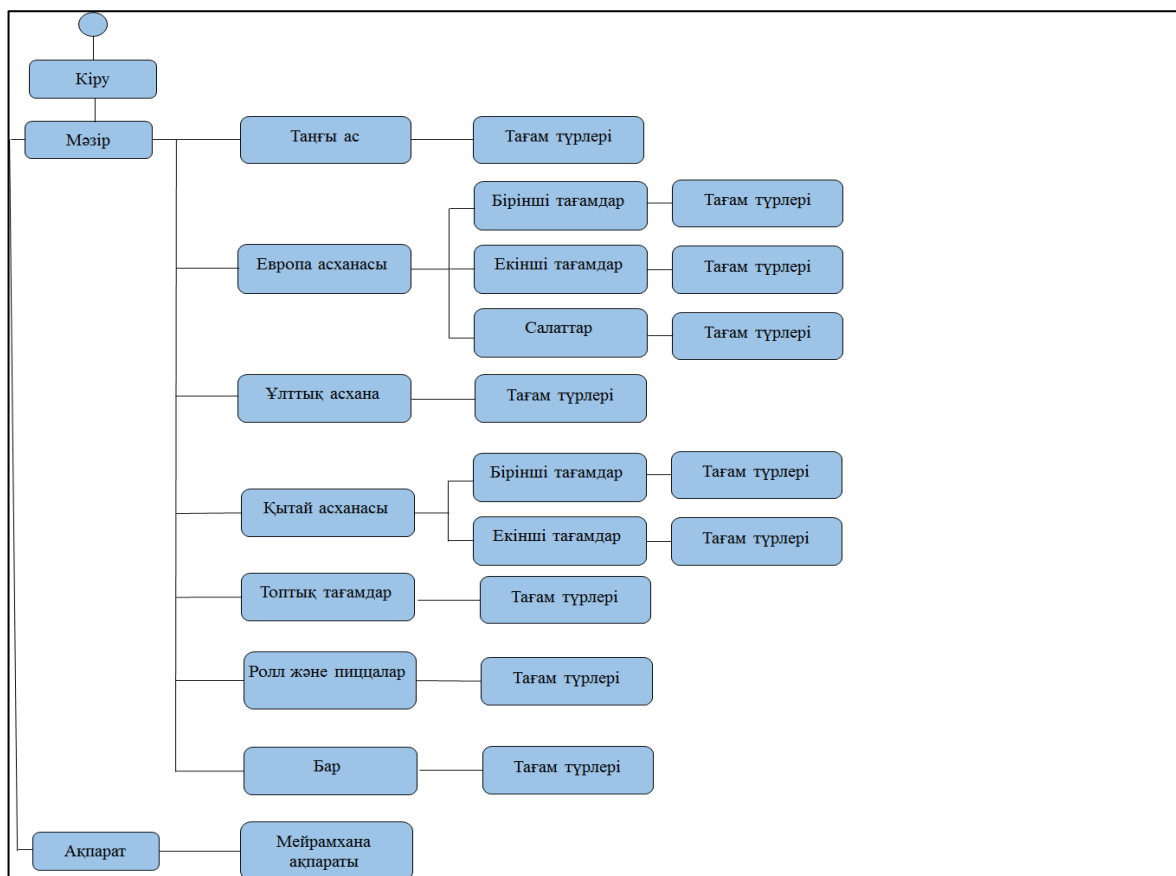


2.3 сурет – Пайдаланушы үшін тізбектілік диаграммасы



2.4 сурет - Күй диаграммасы

Мобильді қосымшаның интерфейстік бөлімін қолданушыға немесе тапсырыс берушіге түсінікті әрі ыңғайлы болуы үшін қолдану варианттарының диаграммасын модельдейміз. Қолдану варианттарының моделі 2.5-суретінде көрсетілген.



2.5 сурет - Қолдану варианттарының диаграммасы

2.5-сурет бойынша қосымшаның бастапқы сценарийінде 2 функция орындалады, олар:

- мәзір;
- ақпарат.

Бірінші функцияны таңдау барысында пайдаланушы үшін келесі функциялар жиынтығы шығады. Яғни, 7 бөлімнен тұратын мәзір функциялары.

Екінші функцияны таңдау арқылы қолданушыға ақпараттық парақ ұсынылатын парақша пайда болады.

Осылайша мобилді қосымшаның кезеңін жобалай отырып, прецеденттермен сәттер көрсетілді. Оның ішінде мобилді қосымшаның модулі мен интерфейсі жобаланды.

3 Қолданбалы бөлім

3.1 Бағдарламамен қамтамасыздандыру сипаттамасы

Операциялық жүйе компьютердің барлық құрылғыларының жұмысын және программалық жасақтың жұмысын басқару үшін қажет. Ол компьютер мен адамның арасындағы байланысты қамтамасыз етеді, демек, Операциялық жүйе – бұл компьютерді іске қосқан кезде жүктелетін және оның барлық құрылғыларының жұмысын басқаратын арнайы бағдарлама.

Операциялық жүйе қолданушымен сұхбат жасай алады, орындалатын басқа бағдарламаларды іске қосады, компьютердің ресурстарын біледі.

Операциялық жүйе адам мен компьютердің электрондық құрауыштарды және қолданбалы бағдарламалар арасында үйлестіруші қызметін атқарады.

Ол адамның бағдарламаларды іске қосуына, барлық мүмкін деректерді оларға беруге және олардан алуға, бағдарлама жұмысын басқаруға, компьютердің және оған қосылған құрылғылардың баптауларын өзгертуге, ресурстарды қайта бөліп беруге мүмкіндік береді.

Windows 7 – Microsoft корпорациясының отбасындағы, компьютерлерге арналған операциялық жүйесі. Операциялық жүйе 22 қазанда 2009 жылғы сатылымға шықты.

Бұл операциялық жүйеге Unicode 5.1 қолданады. Іздеу панелі Instant Search қазір көп тілдерді таниды. Операциялық жүйенің барлық нұсқалары 50 жаңа қаріптерді қамтиды. Windows 7 - көрсету үшін латын қарағанда Латын тіліне жатпайтын таңбаларды көрсету үшін көп қаріптерді қамтитын Windows алғашқы нұсқасы ретінде болып табылады.

3.1.1 Visual Studio Code бағдарламалау тіліне сипаттама

Visual Studio Code - Microsoft, Windows, Linux және MacOS үшін жасалған бастапқы код редакторы. Сонымен қатар, Visual Studio Code - 30-дан астам бағдарламалау тілдері мен файл пішімдерін қолдайтын, сонымен қатар бірқатар қосымша, пайдалы функциялары бар код редакторы.

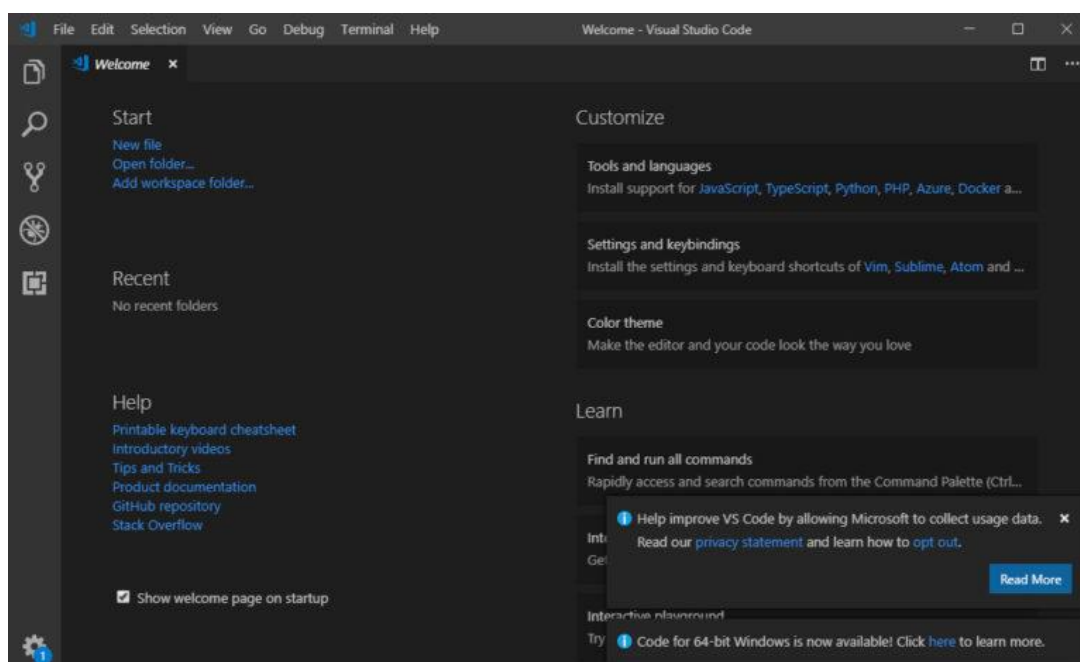
Мүмкіндіктер	Тілдер
Syntax highlight, bracket matching	Batch, C++, Clojure, Coffee Script, Dockerfile, F#, Go, Jade, Java, HandleBars, Ini, Lua, Makefile, Objective-C, Perl, PowerShell, Python, R, Razor, Ruby, Rust, SQL, Visual Basic, XML
+ Code Snippets	Groovy, Markdown, PHP, Swift
+ IntelliSense, linting, outline	CSS, HTML, JavaScript, JSON, Less, Sass
+ Refactoring, find all references	TypeScript, C#

3.1 сурет - Visual Studio Code қолдайтын бағдарламалау тілдері

Мұнда күйін келтіруді қолдау, кірістірілген Git басқару элементтері, синтаксистің ерекшеленуі, кодтың ақылды толтырылуы, үзінділер мен кодты қайта өңдеу кіреді. Сондай-ақ, ол пайдаланушылар тақырып редакторын, пернелер тіркесі арқылы қалауы бойынша өзгерте алады. Бастапқы код ақысыз және MIT ашық лицензиясы бойынша шығарылады. Жинақталған орындалатын құжаттар тегін жеке және коммерциялық мақсатта ақысыз.

Visual Studio коды, жыпылықтайтын қозғалтқыштың режимінде жұмыс істейтін Node.js жұмыс үстелі қосымшаларын орналастыру үшін пайдаланылатын Electron-ға негізделген. Ол электронды жақтауды қолданғанымен, бағдарламалық қамтамасыздандыру Atom қолданбайды, бірақ оның орнына Azure DevOps-те (бұрынғы Visual Studio Internet және Visual Studio Team Services) қолданылатын редактор компонентін («Монако» деп аталады) қолданады.

2018 жылғы сауалнама толтыруының нәтижесінде Visual Studio Code ең танымал әзірлеуші ортаны құрды, оның 75,398 респонденттердің 34,9% -ы оны пайдаланатындығын көрсетті.

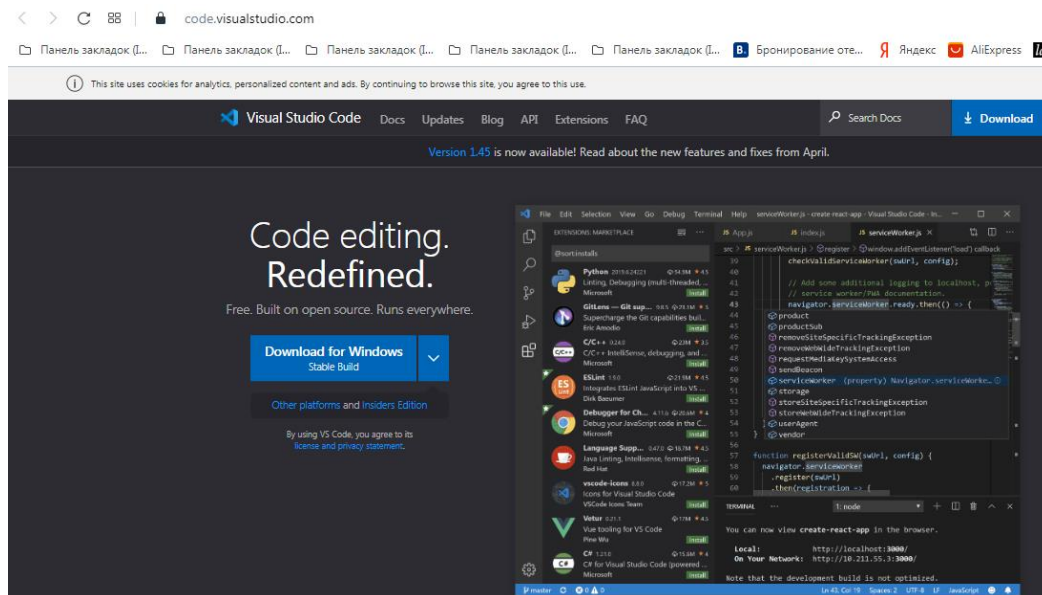


3.2 сурет - Visual Studio Code интерфейсі

3.2-суретіндегі интерфейсстің жоғарғы бөлігінде стандартты мәтіндік мәзір, сол жақта жылдам жету панелі бар. Оның құрамына Explorer, іздеу, нұсқаларды басқару жүйесі, түзету және кеңейту менеджері кіреді. Барлық негізгі жұмыс, яғни кодты құру және редакциялау редактор терезесінің оң жағында орындалады. Ал тәуелсіз қойындыларды пайдалану тапсырманы жеңілдетеді, бұл жеке файлдарды немесе тіпті бүкіл жобаларды ашуға мүмкіндік береді. Microsoft корпорациясы Visual Studio кодын 2015 жылдың 29 сәуіріндегі конференциясында жариялады. Көп ұзамай құрастыруға алдынала шолу шығарылды.

2015 жылдың 18 қарашасында Visual Studio коды MIT лицензиясы бойынша шығарылды және оның бастапқы коды GitHub-та орналастырылды. Сондай-ақ, кеңейтуге қолдау жарияланды.

2016 жылдың 14 сәуірінде Visual Studio коды сахнаны көпшілік алдында қарауды аяқтады және Интернетте шығарылды.



3.3 сурет - Visual Studio Code бағдарламалау тілінің ресми сайты

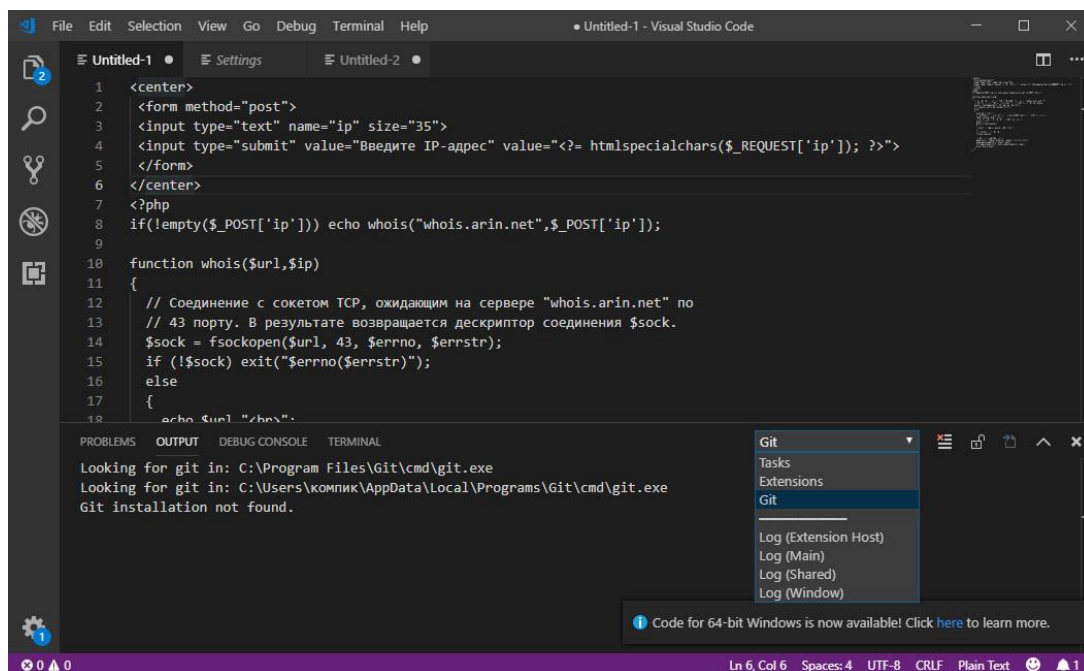
Visual Studio Code - әртүрлі бағдарламалау тілдерінде қолдануға болатын бастапқы код редакторы. Жобалық жүйенің орнына ол пайдаланушыларға бір немесе бірнеше каталогтарды ашуға мүмкіндік береді, оларды болашақта қайта пайдалану үшін жұмыс кеңістігінде сақтауға болады. Бұл оған кез-келген тіл үшін агностикалық тілдік код редакторы ретінде жұмыс істеуге мүмкіндік береді, Microsoft Visual Studio-дан айырмашылығы, ол өзінің .sln шешім файлы мен нақты жоба жобалық файлдарын пайдаланады. Ол бірнеше бағдарламалау тілдерін және функциялардың жиынтығын қолдайды және тілге байланысты өзгеріп отырады. Параметрлер арқылы қалаусыз файлдар мен қалталарды жоба ағашынан шығаруға болады. Студия кодтарының көптеген визуалды мүмкіндіктеріне мәзір немесе қолданушы интерфейсі арқылы қол жетімді емес. Бірақ пәрмендер палитрасы арқылы қол жеткізуге болады.

Visual Studio коды орталық репозиторий арқылы қол жетімді плагиндермен кеңейтілуі мүмкін. Бұған редактордың қосымшасы және тілдік қолдау кіреді. Белгілі бір ерекшелігі - жаңа тілдерге, тақырыптарға, күйін келтірушілерге қолдау көрсететін, статикалық кодтық анализді жүзеге асыратын, сервер тілінің протоколын қолдана отырып, штрих кодын қосатын және қосымша қызметтерге қосылатын кеңейтімдерді құру мүмкіндігі.

Visual Studio коды FTP үшін бірнеше кеңейтімдерді қамтиды, бұл бағдарламалық жасақтаманы веб-әзірлеудің еркін баламасы ретінде

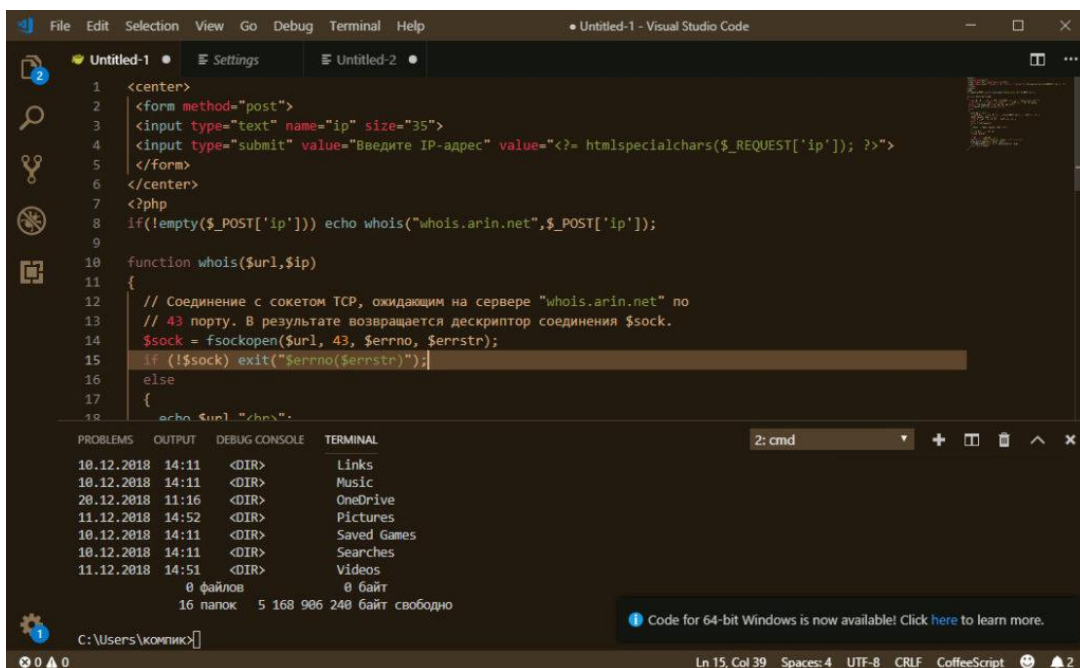
пайдалануға мүмкіндік береді. Кодты қосымша бағдарламалық жасақтаманы жүктемей-ақ, редактор мен сервер арасында үндестіруге болады.

Visual Studio Code пайдаланушыларға белсенді құжат сақталатын код бетін, содан кейін Windows / Linux үшін жаңа жол таңбасын, сондай-ақ белсенді құжаттың бағдарламалау тілін орнатуға мүмкіндік береді. Бұл оны кез-келген платформада, кез-келген жерде және кез-келген бағдарламалау тілі үшін пайдалануға мүмкіндік береді.

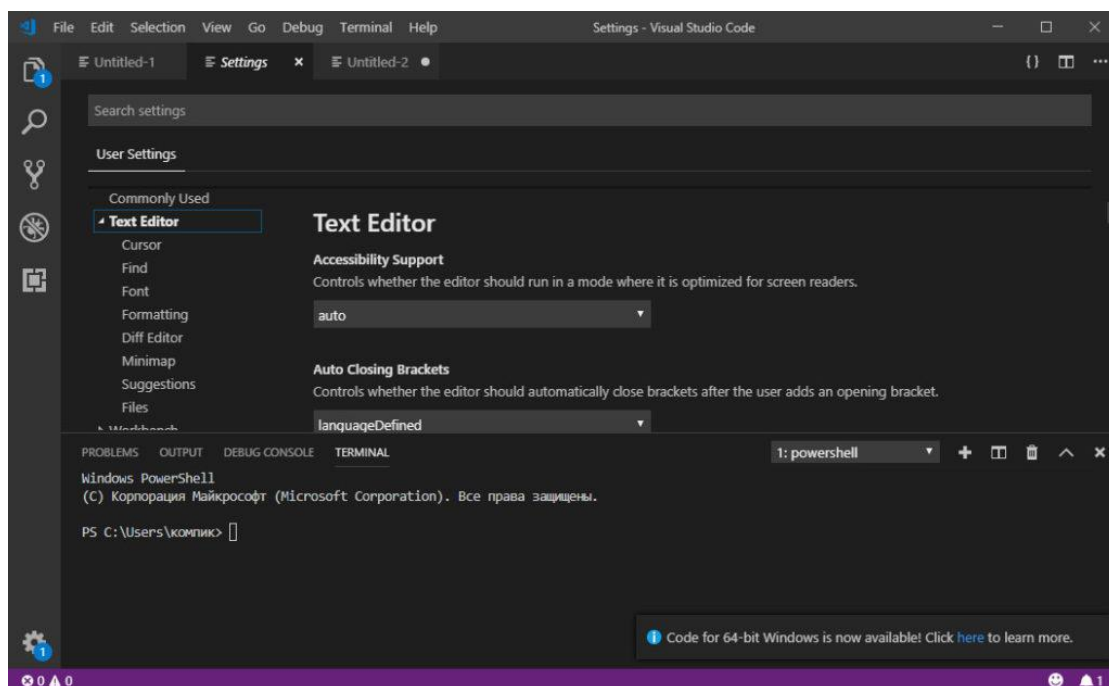


3.4 сурет - Visual Studio Code жұмыс сипаттамасы

3.4-суретте соңғылары Visual Studio Code-ге «Файл» негізгі мәзірі арқылы немесе каталогты редактордың терезесіне сүйреп апару арқылы қосылады. Терезенің төменгі шығыс аймағында орналасқан төрт қойынды бар. Олардың әрқайсысы жұмыс істеп жатқан жобаға байланысты белгілі бір типтегі мәліметтерді шығаруға жауап береді. Редактор терезесінде PowerShell консолімен, CMD немесе Linux Bash пәрмендерімен жұмыс істеуге мүмкіндік беретін «Терминал» қосымша беті ерекше назар аудартады.



3.5 сурет - Visual Studio Code-де көптеген мүмкін параметрлер бар, оның ішінде редактордың сыртқы түрін және мінез-құлқын өзгерту



3.6 сурет – Қаріптермен жұмыс жасау

Қаріптің өлшемі мен шрифті, түс схемасы, жаңа терезелердің өлшемі, құралдар тақтасының дисплейі, ескертулер мен ескертулердің көрінісі - осының бәрі және басқалары сіздің қалауыңызға сәйкес келтірілуі мүмкін.

Бағдарламаның негізгі ерекшеліктері мен артықшылықтары:

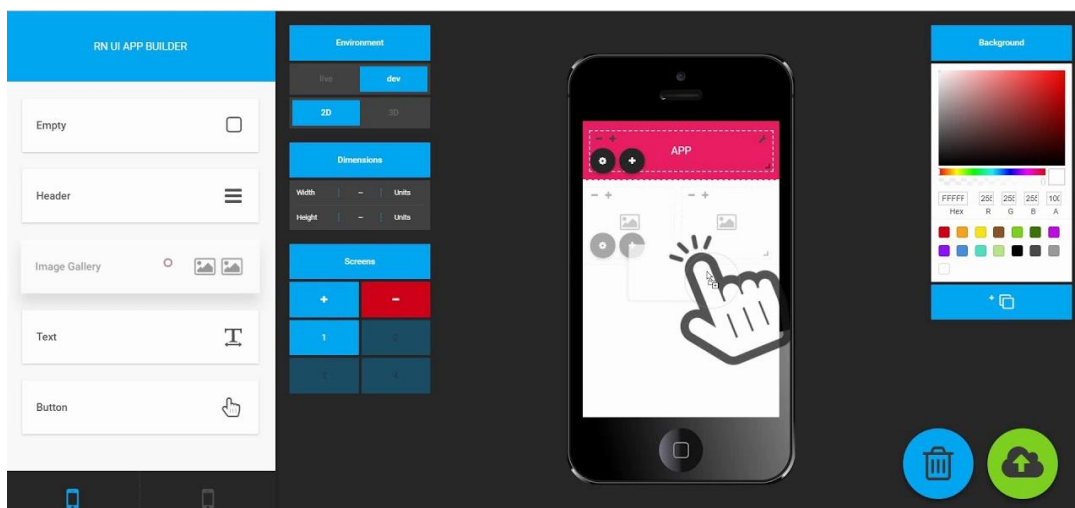
- Visual Studio коды TypeScript, JavaScript, Node.js және Mono-мен жұмыс істеуді қолдайды;

- кірістірілген түзету құралы және пәрмен жолы бар;
- барлық дерлік бағдарламалау тілдерін қолдау;
- код элементтерінің кірістірілген кітапханасының болуы;
- кодты енгізу кезінде авто-толтыру;
- кітапханаға өз үзінділеріңізді қосу;
- синтаксисті бөлектеу;
- бір уақытта бірнеше жобамен жұмыс;
- көп терезелі және қос панельдік режимдерді қолдау;
- плагиндерді қолдану арқылы функцияларды кеңейту;
- Visual Studio Team Services, GitHub және GIT интеграциясы;
- қосымшаларды тестілеуге, құрастыруға, буып-түюге және орналастыруға арналған кіріктірілген құралдар;
- microsoft Azure-де жасалған бағдарламалық өнімдерді шығару (Visual Studio Team Services арқылы);
- кіріктірілген кеңестер жүйесі;
- жобалар бойынша топтық жұмыс;
- параметрлер мен платформалардың кең спектрі.

Visual Studio коды - бұл толыққанды IDE емес, ол ыңғайлы, қарапайым және қол жетімді код редакторы. Оның кәсіби бағдарламашыларға қатты қызығатыны екіталай, бірақ бастаушы кодерлер мен веб-қосымшаны жасаушылар оны пайдалы деп санайды. Visual Studio Code-мен және басқа код редакторларын өз жұмысында белсенді түрде қолданатын адамдармен танысу зиян келтірмейді, кем дегенде салыстыру үшін. Қосымшаның функционалдығын бағалай отырып, дамудың басқа құралдарының ізбасарлары өздері үшін көптеген қызықты және мүмкін болатын нәрсені таба алады, оған артықшылық береді.

3.1.2 React Native көмегімен кросс-платформалық қосымшаларды құру

React Native - бұл жергілікті түрде көрсетілетін iOS және Android қосымшаларын жасауға арналған JS жүйесі. Ол Facebook-те жасалған, пайдаланушылық интерфейстерді құруға арналған React JS кітапханасына негізделген. Бірақ браузерлердің орнына мобильді платформаларға бағытталған. Басқаша айтқанда, егер сіз веб-әзірлеушісіз болсаңыз, сіз өзіңіздің Javascript жүйеңізден және код базасынан ыңғайлылықсыз таза, жылдам мобильді қосымшаларды жазу үшін React Native қолдана аласыз.



3.7 сурет - React Native редакторлау жұмыс интерфейсі

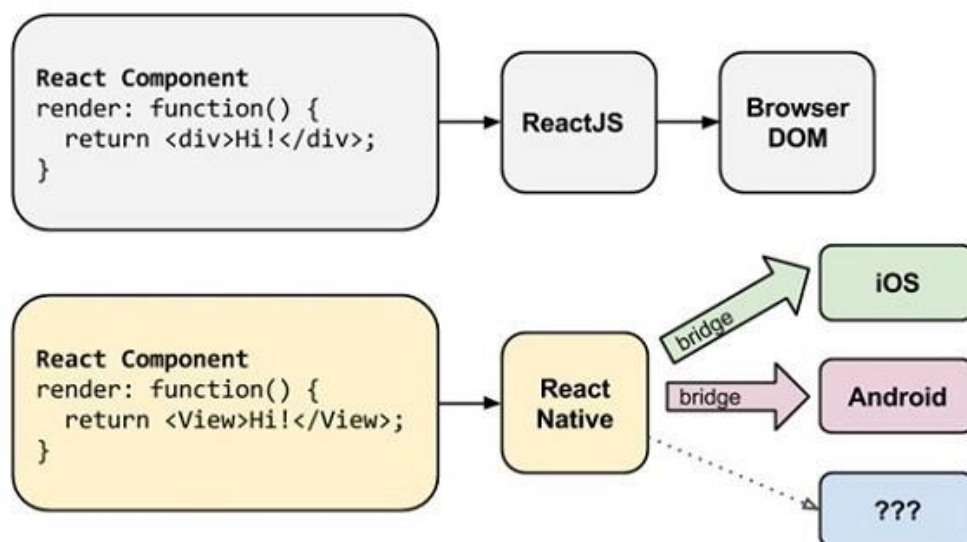
React - бұл әдетте веб-қосымшаларға арналған интерфейстерді құруға арналған JS кітапханасы. Ол Facebook-те жасалды және 2013 жылдан бастап бастапқы ашық лицензия бойынша таратылды. Реакция кең таралған және үлкен MVC рамкаларынан айырмашылығы ол салыстырмалы түрде тар міндеттерді шешеді: интерфейс көрсету.

React-тің танымал болуына бірнеше себептер бар. Ол ықшамды және жоғары өнімділікке ие, әсіресе жылдам өзгертін деректермен жұмыс жасағанда. Компоненттік құрылымының арқасында React сізді модульдік, қайта пайдалануға болатын код жазуға шақырады.

React Native - бірдей реакция, бірақ мобильді платформалар үшін. Оның бірнеше айырмашылықтары бар: div тегінің орнына View компоненті қолданылады, ал img тегінің орнына Image компоненті қолданылады. Даму процесі өзгеріссіз қалды. Objective-C немесе Java-ны білу пайдалы болуы мүмкін. Сонымен қатар, мобильді әзірлеудің өзіндік ерекшеліктері бар.

React Native адамдарын таң қалдыратын бірінші нәрсе - бұл «шынайы» туған жер. Мобильді платформаларға арналған басқа Javascript шешімдері жай ғана JS кодыңызды ыңғайлы веб көрінісіне орап алады. Олар анимация сияқты кейбір интерфейс әрекеттерін орындай алады, бірақ ол веб-қосымша болып қала береді.

React бағдарламасында компонент өзінің картографиясын сипаттайды. Содан кейін кітапхана сіз үшін өңдеуді өңдейді. Бұл екі функция абстракцияның нақты деңгейімен бөлінген. Егер сізге веб-компоненттерді көрсету керек болса, онда React стандартты HTML тегтерін қолданады. Дәл осындай абстракция деңгейіне байланысты - «көпір» - iOS және Android-те көрсету үшін, React Native сәйкес API шақырады. IOS-та компоненттер нақты UI көріністеріне, ал Android-де жергілікті көріністерге шығарылады (3.8-суретінде көрсетілген).

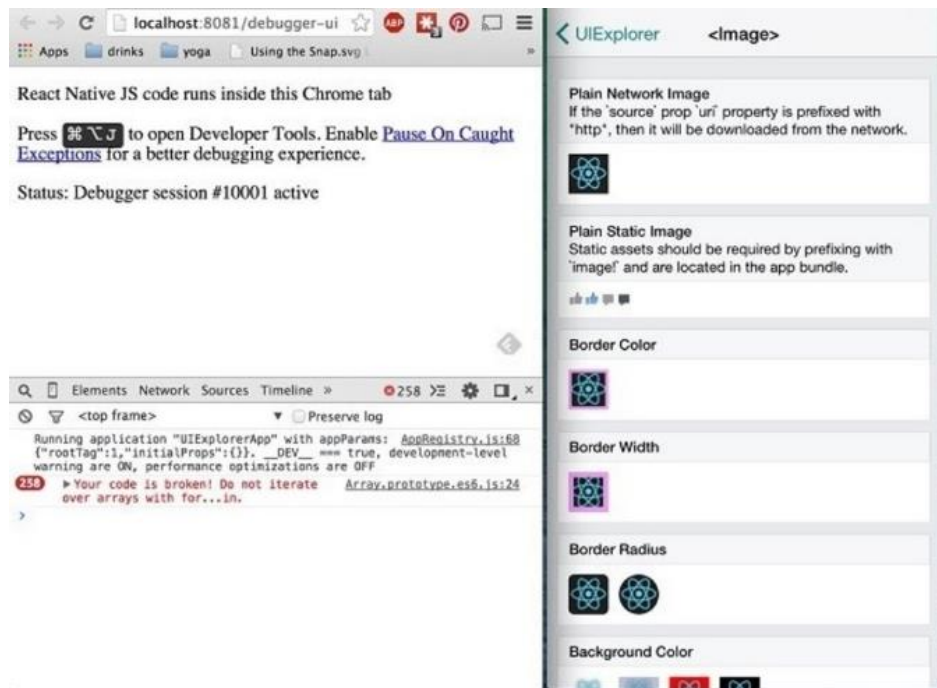


3.8 сурет – React Native жұмыс жасау принципі

Біз стандартты JavaScript, CSS және HTML-ге ұқсас өте керемет көрінетін кодты жазасыз. Жергілікті кодқа енгізудің орнына, React Native сіздің қосымшаңызды қабылдайды және оны негізгі UI ағынын бұғаттамай, хост платформаның JS қозғалтқышы арқылы іске қосады. Біз жергілікті өнімділіктің, анимацияның және мінез-құлықтың артықшылықтарын Objective-C немесе Java-да жазбай-ақ аласыз.

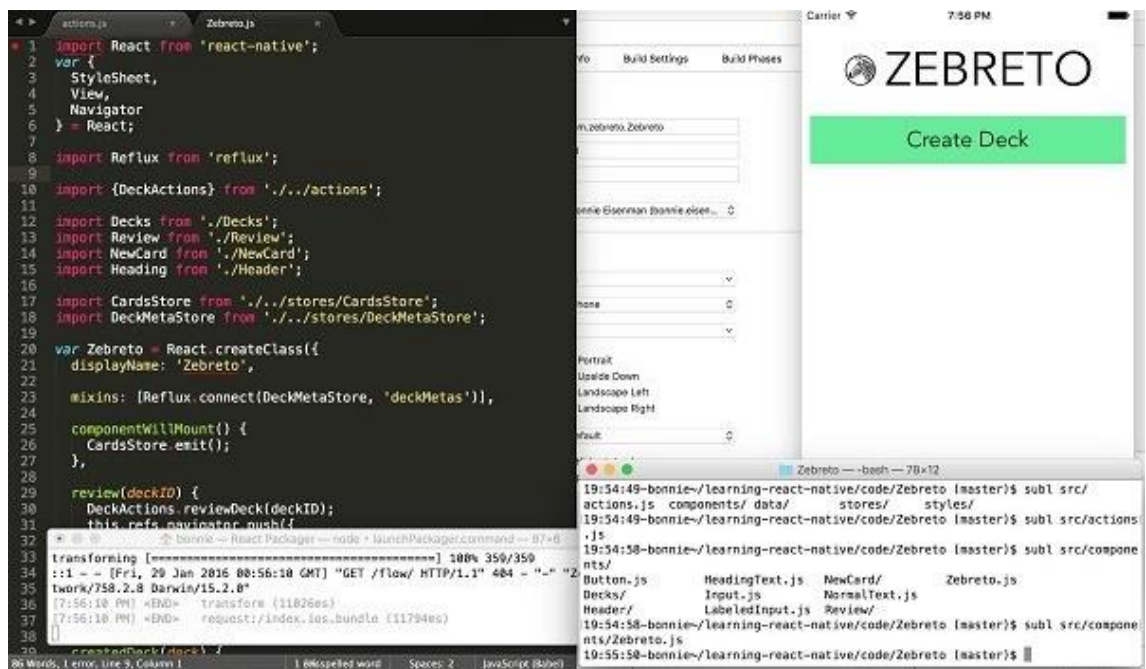
Әзірлеуші үшін React Native артықшылықтары. Стандартты IOS және Android-пен салыстырғанда, React Native артықшылықтары көп. Біздің қосымшаңыз негізінен JavaScript болғандықтан, біз веб-дамудың көптеген артықшылықтарын пайдалана аласыз. Мысалы, кодқа енгізілген өзгертулерді көру үшін біз дәстүрлі қайта құруды аяқтауды ұзақ күте отырып, бағдарламаны лезде «жаңарта» аласыз. Сонымен қатар, React Native қателіктер туралы ақылды хабарлау жүйесін және JavaScript түзетудің қарапайым құралдарын ұсынады. Бұл мобильді даму процесін едәуір жеңілдетеді.

Мобильді қосымша ретінде бейімдеу. Бұл React Native бағдарламасымен жұмыс істеудің қиын бөлігі. Фреймворк пен жұмыс істеу кезінде бізге мобильді әзірлеушінің барлық әдеттегі құралдары, сондай-ақ JavaScript-пен жұмыс істеуге арналған құралдар қажет болады: мәтіндік редактор және жөндеу құралдары.



3.9 сурет - Қателіктерді туралы ақылды хабарлау жүйесі

IOS үшін бізде Xcode және мобильді эмулятор жұмыс істеуі керек. Ал Android үшін, Android Studio және бірқатар пәрмен жолындағы құралдар болуы керек. Бізге React Native орамы қажет болады. JavaScript коды үшін мәтіндік редакторды таңдау бізге байланысты.



3.10 сурет - Мобильді қосымша ретінде бейімдеу процесі

Бұл құрылым кез-келген веб-әзірлеушіні мобильді қосымшалардың әлеуетті әзірлеушісіне айналдырады, бұл оларды жасау процесін едәуір жақсартуға мүмкіндік береді.

React Native бағдарламасының кемшіліктері бар. Бұл салыстырмалы түрде жаңа жоба және ол барлық жас кітапханалардың проблемаларымен сипатталады: кейбір функциялар жетіспейді, пайдаланудың оңтайлы әдістері әлі әзірленбеген. Шығарудан босатуға дейін елеулі өзгерістер енгізілуде, бірақ олардың саны аз, шектеулі.

Дегенмен, React Native - бұл жетілген жоба, оның артықшылықтары кемшіліктерге қарағанда едәуір маңызды. Осы шеңберді қолдана отырып, сапа мен өнімділікті жоғалтпастан iOS және Android үшін қосымшалар жасау үшін бір кодтық базаны пайдалануға болады. Егер біз JavaScript-ті білмесеңіз де, біз жылдам даму циклінің артықшылықтары мен кодты жүз пайызға қайта пайдаланудың артықшылықтары туралы дау айтуыңыз мүмкін емес. React Native бізге қажет болған жағдайда «қалыпты» дамуға ауысуға мүмкіндік беретіндіктен, шеңберге тән шектеулер бізді алаңдатпайды. Жалпы алғанда, React Native жоғары сапалы кросс-платформалық мобильді дамуды қамтамасыз етеді және бұл құралға қызығушылық арттыруға тұрарлық.

3.1.3 NodeJS бағдарламалық платформасы

Node немесе Node.js - бұл арнайы мамандандырылған тілден JavaScript тілін жалпы мақсаттағы тілге айналдыратын V8 қозғалтқышы (JavaScript кодын машиналық кодқа аудару) негізіндегі бағдарламалық платформа. Node.js JavaScript-ке оның API-мен (C ++ тілінде жазылған) кіру / шығару құрылғыларымен әрекеттесу, басқа тілдерде жазылған басқа кітапханаларды қосып, оларға JavaScript кодынан қоңырау шалу мүмкіндігін қосады. Node.js негізінен веб-сервер ретінде жұмыс істейтін серверде қолданылады, бірақ Node.js-та жұмыс үстелінің терезелік қосымшаларын (Linux, Windows және macOS үшін NW.js, AppJS немесе Electron қолдана отырып) және тіпті бағдарламалық микроконтроллерлерді де (мысалы, tessel, low.js және espruino) бағдарламалау мүмкіндіктері бар. Node.js оқиғаға негізделген және блоктан тыс кіріс / шығыссыз асинхронды (немесе реактивті) бағдарламалауға негізделген.

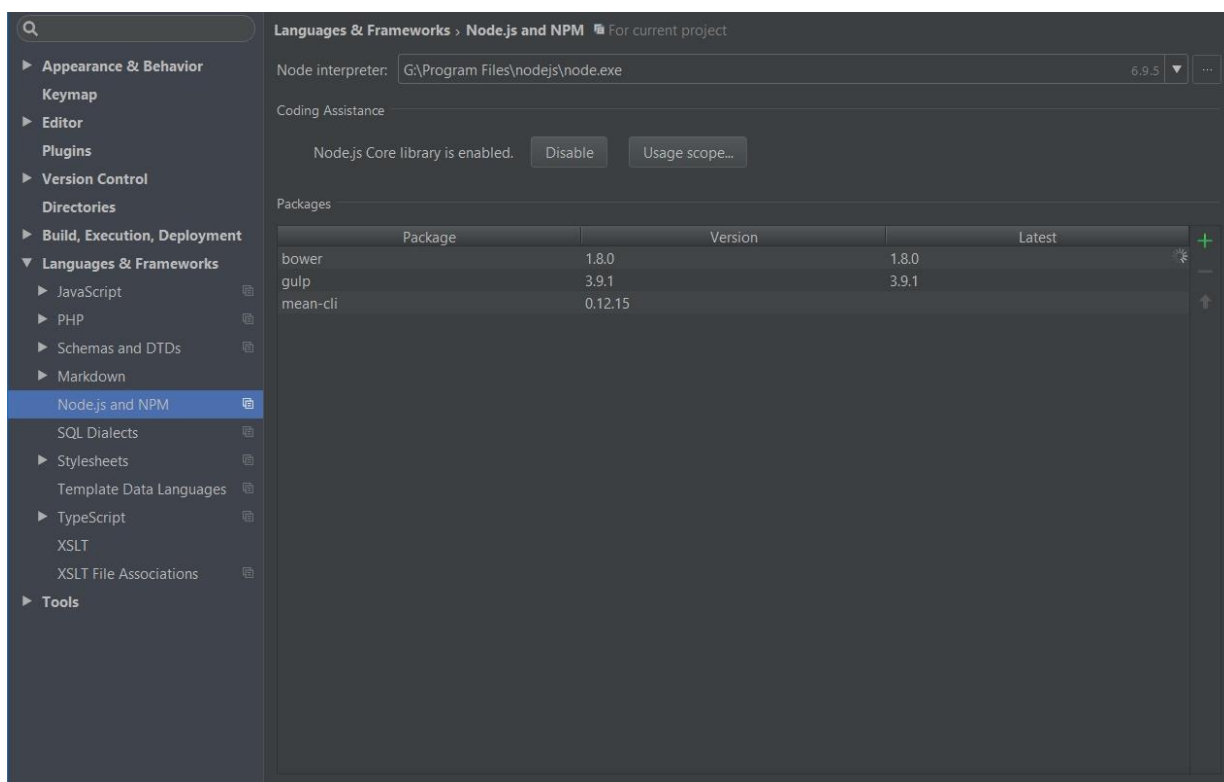
Node.js ортасында JavaScript тілінде жазылған бағдарламаны іске қосу үшін қажет нәрсенің бәрі бар.

Бұрын біз JavaScript-ті тек шолғышта іске қосуға болатын едік, бірақ әзірлеушілер оны бір рет кеңейтсе, енді JS-ны жеке бағдарлама ретінде компьютерде іске қосуға болады. Сонымен Node.js болды.

Енді біз тек қана интерактивті веб-сайттардан гөрі JavaScript-пен көбірек жұмыс жасай аламыз.

Енді JavaScript-те Python сияқты басқа сценарийлерді бағдарламалау тілдерін жасай алатын мүмкіндік бар.

Шолғышқа негізделген JavaScript және Node.js V8 жұмыс уақытында жұмыс істейді. Бұл қозғалтқыш біздің JS кодыңызды қолданады және оны тезірек машина кодына түрлендіреді. Машина коды - бұл компьютердің алдымен оны түсіндірместен жұмыс істей алатын төменгі деңгейдегі коды



3.11 сурет – NodeJS баптаулар интерфейсі

Node.js ресми сайтында берілген ресми анықтамаға сүйенсек:

- Node.js - JavaScript V8 қозғалтқышына негізделген JavaScript жұмыс ортасы;
- Node.js бағдарламасы оқиғаны басқаратын, блоктамайтын шығыс-кіріс моделін қолданады және оны оңай әрі тиімді жасай алады;
- Node.js пакеттік экожүйе, npm - әлемдегі ең үлкен ашық кітапханалық экожүйе.

3.1.4 Ехро көмегімен жергілікті React қосымшаларын жасау процесін жеңілдету

Ехро - бұл жергілікті React қосымшаларын құруды жеңілдететін құралдар жиынтығы.

Ехро көмегімен бағдарламашы React Native қосымшаларын Android Studio, Xcode сияқты бағдарламалық жасақтамаға тәуелді болмай, React Native қосымшаларын жасау және іске қосу үшін қажет басқа құралдарды жасай алады.

Expo - бұл тез дамып келе жатқан жергілікті React қосымшалары үшін негізделген бағдарлама. Бұл PHP әзірлеушілеріне арналған Laravel немесе Symfony немесе Ruby on Ruby жасаушыларға арналған. Expo қолдануды және басқаруды жеңілдету үшін React Native API пәрмендерінің үстіне қабат береді. Сондай-ақ, React Native қосымшаларын бастапқы құру және тестілеуді жеңілдететін құралдар бар. Сонымен, ол пайдаланушылық интерфейс компоненттері мен қызметтерін ұсынады, олар әдетте тек үшінші тараптың жергілікті React компоненттерін орнатқан кезде қол жетімді болады. Барлығы Expo SDK арқылы қол жетімді.

Expo шектеулері:

- Expo қосымшалары фондық кодтың орындалуын қолдамайды. Мысалы, бағдарлама жабылған кезде орналасқан жердің өзгеруін таңдайтын кодты іске қоса алмайтыныңызды білдіреді;

- Expo қосымшалары Expo SDK қолдайтын жергілікті API-лермен шектелген. Бұл біздің бағдарламада Bluetooth құрылғысымен мәліметтер алмасу сияқты нақты міндеттер болса, бұл функцияны жүзеге асырудың жалғыз мүмкіндігі қарапайым React Native пайдалану немесе ExpoKit кітапханасын пайдаланып өз кодымызды жазу болып табылады;

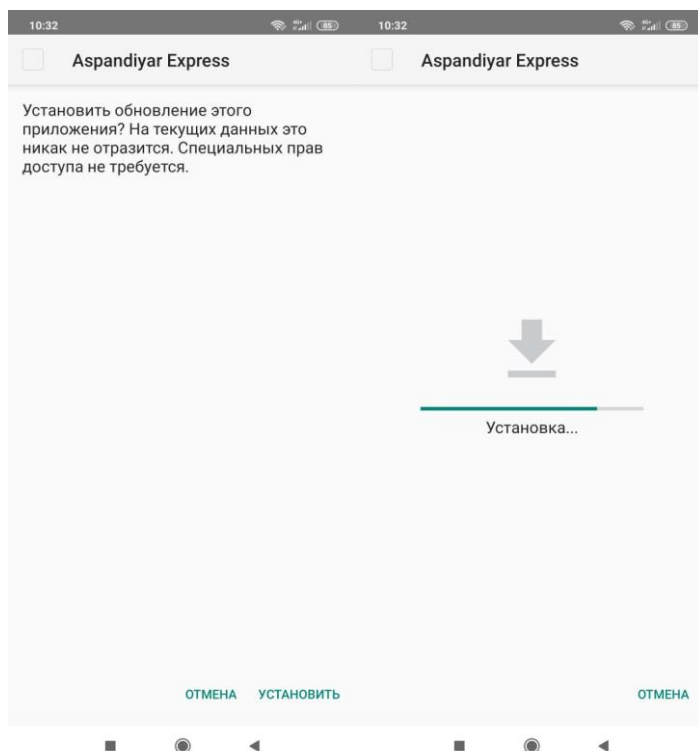
- Expo бізді оның құралдарымен байланыстырады. Бұл React Native дамыту үшін қол жетімді көптеген құралдарды, мысалы, пәрмен жолының құралдары, көмекші кітапханалар және пайдаланушы интерфейсін негіздері ретінде ғана орнатып, қолдана алмаймыз дегенді білдіреді. Бірақ жағымды жаңалық - Expo SDK қарапайым React Native қосымшаларымен үйлесімді, сондықтан біз қосымшаны Expo-дан шығарсақ, бізде ешқандай қиындық болмайды;

- оқшау Expo қосымшасының орындалатын файлдары тек желі қосылымымен ғана құрылуы мүмкін. Expo Exp деп аталатын командалық жол құралдарын ұсынады. Бұл әзірлеушілерге Expo серверлерінде жобаны құру процесін жүргізуге мүмкіндік береді. Жинау аяқталғаннан кейін .apk немесе .ipa файлы үшін URL мекенжайы қол жетімді болады.

Осы шектеулерге қарамастан, Expo жиі қолданылатын Android және iOS API интерфейстеріне үлкен қолдау көрсететін толық функционалды база екенін есте ұстаған жөн. Бұл бізге әдетте қосымшаларға қажет негізгі функционалдылықты қамтамасыз етеді дегенді білдіреді. Сонымен, жергілікті функцияларды жүзеге асыру үшін Expo мүмкіндіктері жетеді.

3.2 Мобильді қосымшаны іске асыру және бағдарламалық кешеніне сипаттама

Мобильді қосымшаны іске асыру үшін Android ОЖ-мен қамтамасыз етілген мобильді құрылғыға AspandiyarExpress.apk бағдарламасын орнату керекпіз.



3.12 сурет – Бағдарламаны мобильді қосымшаға орнату барысы

3.12-суретінде көрсетілгендей бағдарламаның мобильді қосымшада орнатылуына рұқсат береміз. Бұл сұқбат терезесінің шығу себебі, мобильді құрылғының бағдарламалық қауіпсіздігіне төнетін қауіп-қатердің жоқ екендігін және оны алдын-алу болып табылады.



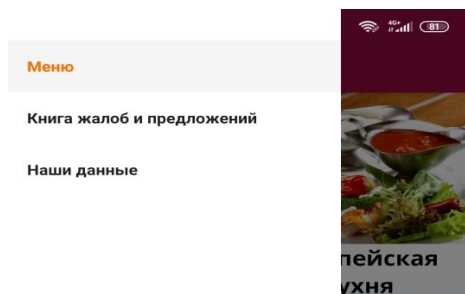
3.13 сурет – Басты бет. Мәзір бөлімдері

Мобильді қосымшаға арналған бағдарлама орнатылғаннан кейін қосымшаны ашамыз. Қосымша ашылғаннан кейін автоматты түрде мәзір бөлімдеріне шығарады. Бұл бетте мейрамхана желісінде қандай асханалар түрінің бар екендігімен таныса аламыз. Бұл беттің ыңғайлығы, клиент өзіне жақын асхана түрін бірден таңдау арқылы уақыт үнемдейді. Ал қағазды мәзірде, өзіне жақын асхана түрін табу үшін қажетсіз бірнеше беттерді ақтаруға тура келетін.

Келесі қадамда жоғарғы бөлікте орналасқан қосымша бөлімдеріне назар аударамыз. Қосымша қазіргі таңда 3 бөлімнен тұрады (3.14-сурет):

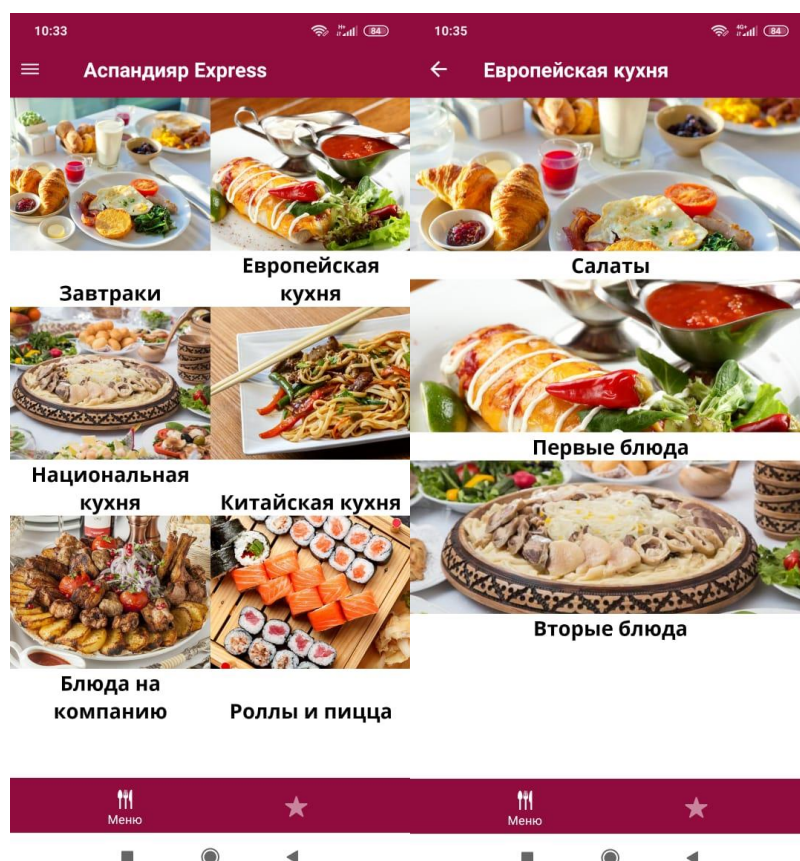
- мәзір бөлімі;
- пікір қалдыру бөлімі;
- мейрамхана жайлы ақпарат.

Әрбір бөлімнің өзіне тән атқаратын қызметі бөлек. Бұл бөлімдер клиент үшін ақпараттар мен қызметтердің қолжетімділігін едәуір жеделдетеді.



3.14 сурет – Мобильді қосымша бөлімдері

Келесі қадамда мәзірдің негізгі бөлімдерімен танысқаннан кейін әр бөлімде орналасқан тағамдар түрімен танысу. Клиенттің уақытын әрі жұмыстың мобильділігін арттыру мақсатында әрбір асхана бөлімі өзіне тән бөлімдермен бөлінген. Мысалы, әрбір асхана ішінде бірінші тағам түрлері, екінші тағамдар түрлері, тіске басар түрлеріне арналған бөлімдер бар. Осы бөлімдер арқылы қажетті тағам түрін тез әрі жылдам табуға мүмкіндік туғызады. 3.15-суреті арқылы осы процесті бақылай аламыз.

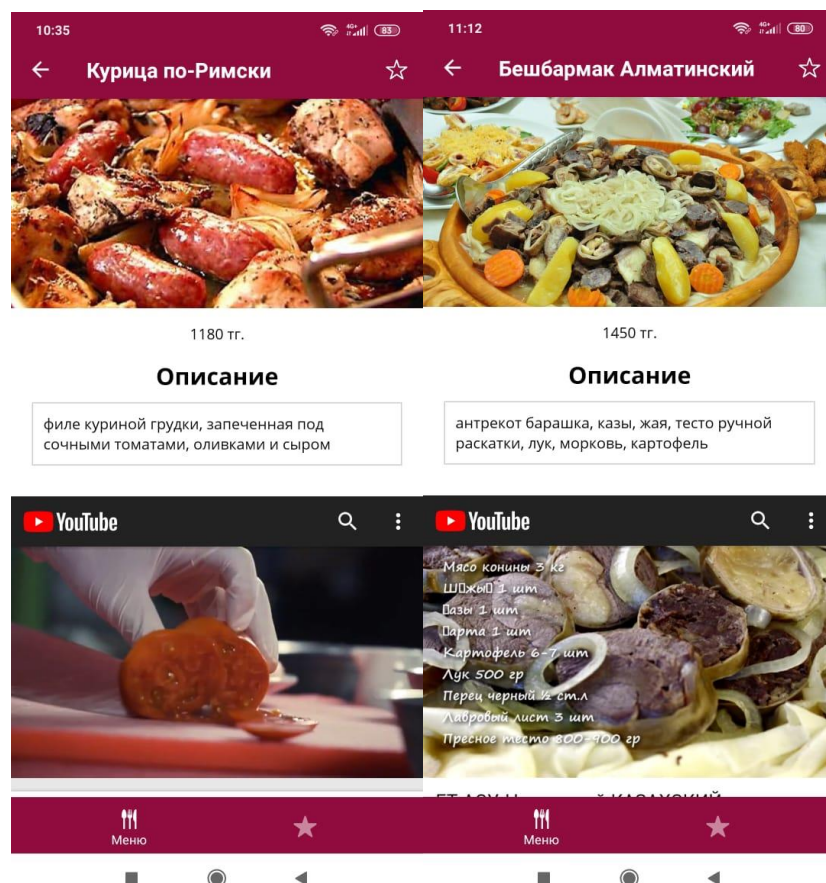


3.15 сурет – Мәзір бөлімдері

Қажетті бөлімді таңдағаннан кейін келесі тапсырма көңіл қалауындағы тағамды таңдау. Бұл бөлімде егер клиент үшін тағам жайлы қандай да бір сұрақ туындаған жағдайда қосымша көмегімен толықтай таныса алады. Бұл тапсырыстың толық қанды клиент ойындағыдай екендігін нақтылай түседі. Тағам жайлы ақпарат пен келесідей мүмкіндіктер арқылы таныса алады:

- тағамның суреті;
- тағамның құрамы;
- тағамның бағасы;
- тағамның жасалу жолы жайлы бейне ролик.

Осы мүмкіндіктер арқылы ақпарат алу тездетіледі.



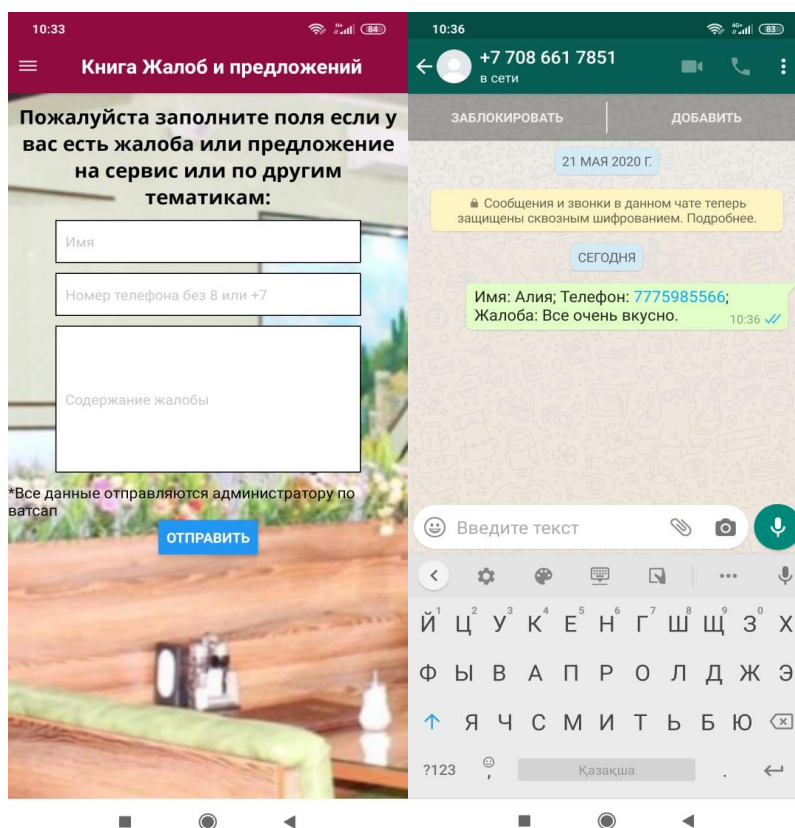
3.16 сурет – Жекленген тағам жайлы толық қанды ақпаратпен танысу

«Аспандияр Экспресс» жалпыға бірдей тамақтану мейрамханасында 270-тен астам тағам түрлері қолданысқа ие. Бұл келген клиент үшін таңдау мүмкіндітерін қиындата түспек және тағам түрлерінің көптігі кейбір танысқан тамақтар есінде қалмай қалуы әбден мүмкін. Сондықтан қосымшада клиенттің көңілінен шығып, ұнатқан тағамдарды «таңдаулы» бөліміне өткізетін функция іске асырылды. Бұл функция арқылы бірнеше бөлімдерден бірнеше тағамдар таңдай алады.

Одан бөлек, «таңдаулы тағамдар» функциясы арқылы клиент даяршымен қандайда бір байланысқа түспестен тапсырыстарды қабыл алуын қамтамасыз етеді. Бұл екіжақтың да жұмыс және уақыт тиімділігін арттыруға өз септігін тигізеді. Бұл функция қызметі мәзірдің барлық бөлімдері үшін қолжетімді. Функция үлгісі 3.17-суретінде көрсетілген.



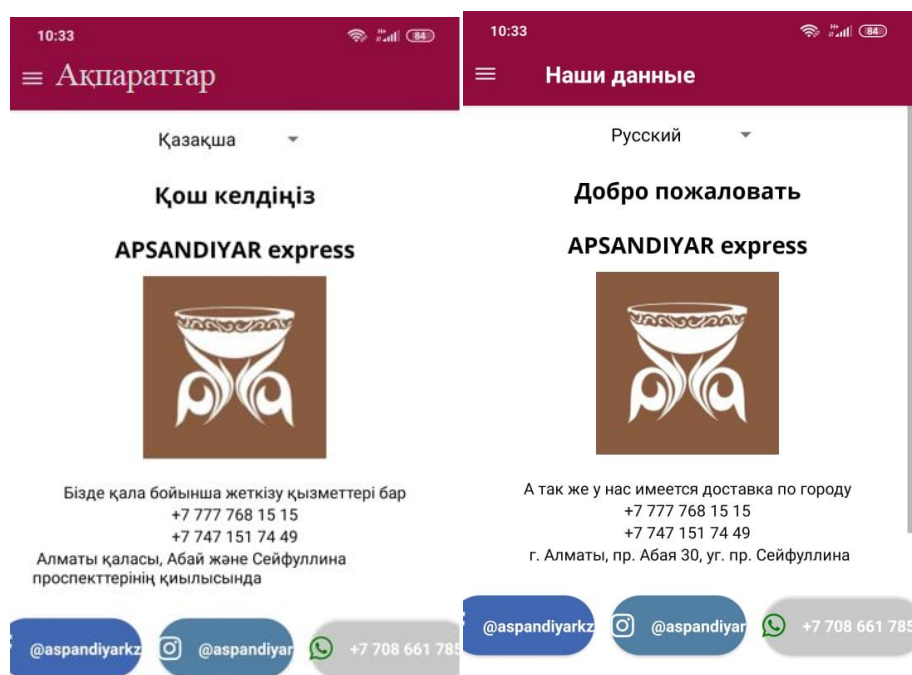
3.17 сурет – «Таңдаулы» функциясы арқылы әртүрлі бөлімдерден тағам таңдау



3.18 сурет – Пікір қалдыру қызметі

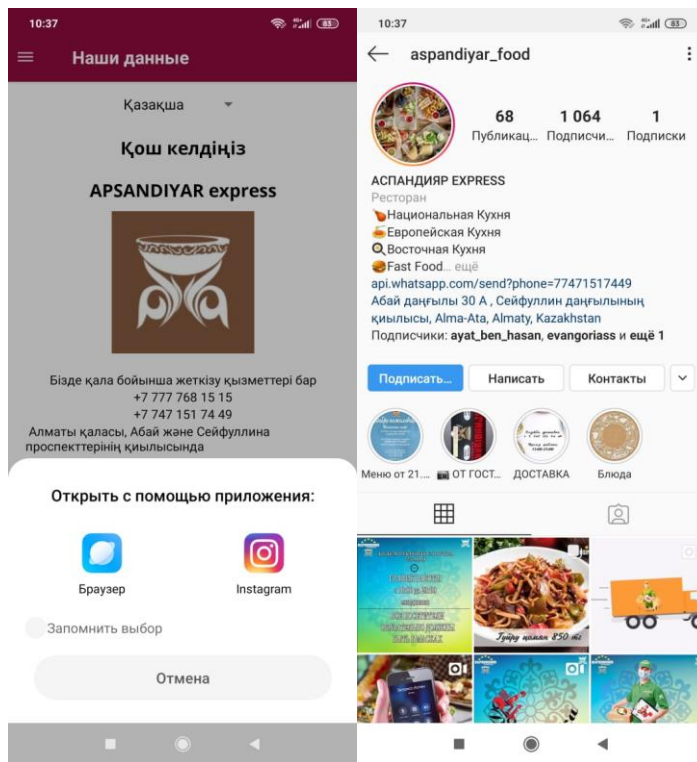
Мейрамхананың даму үрдісі үшін маңызды факторлардың бірі клиенттің ойын, пікірін, сынын білу. Барлығымыз білетіндей ол әрбір тамақтану орындарында арнайы кітапшаға жазу арқылы немесе әкімшілікке ескерту арқылы шектеу тауып жатады. Басшылық үшін көптеген шағымдар, пікірлер жеткізілмей жатады. Дәл осы кемшілікті қосымшаның «Пікір қалдыру» (3.18-сурет) функциясы арқылы шешілді. Клиент жазған әрбір шағым немесе жағымды пірік тікелей басшылықтың whatsapp қосымшасында тіркелеген байланыс нөміріне жіберіледі. Бұл, басышылықтың әрдайым жұмыс сапасының басқа бір тарапынан қадағалап отыруына мүмкіндік береді.

Мейрамханаларға көп жағдайда бір күндік есебі бойынша 30% тұрақты клиенттер келсе, 70%-ға жуық бөлігі жаңа клиенттер. Ақпараттармен танысу бөлімі арқылы жаңа клиенттер және басқада сұрағы бар клиенттер толық қанды жауабын таба алады. Бұл бөлімінде клиентке қажетті ақпараттар легінің толықта жиынтығы орналасқан (3.19-сурет). Одан бөлек, мәзірді екі тілде қызмет көрсететінін байқауға болады. Бұл да клиент үшін қосымша сұрақтардың жауабын беретіні сөзсіз.



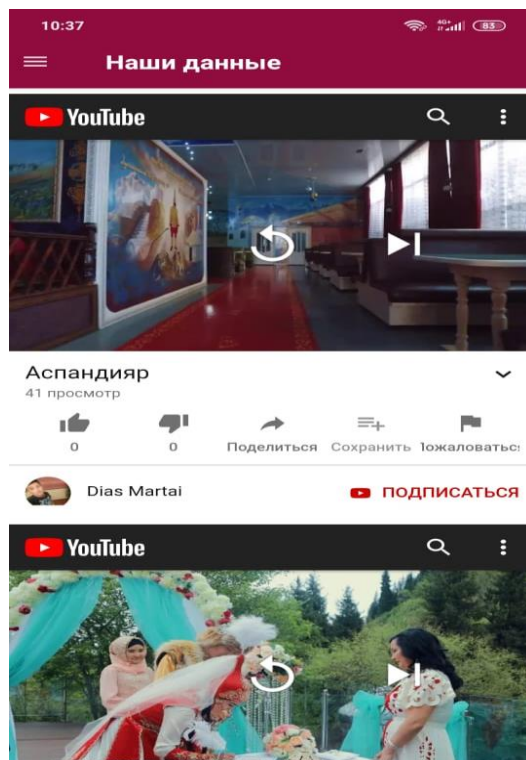
3.19 сурет – Ақпарат жеткізу бөлімі

Осы бөлім арқылы қазіргі таңда аса танымал әлеуметтік желілерге өте алады. Әлеуметтік желілердің мейрамхана үшін жаңа клиенттер және ақпарат алмасудың қосымша көзі екендігі сөзсіз.



3.20 сурет – Қосымша ақпараттар көзімен танысу

Осы бөлім арқылы клиент мейрамхананың басқа да бөлімшелері мен мүмкіндіктерін жарнамалық бейнероликтер арқылы таныса алады.



3.21 сурет – Жарнамалық бейнероликтармен танысу

4 Экономикалық бөлім

4.1 Экономикалық бөлімге кіріспе

Бүгінгі таңда мейрамхананың алдыңғы қатарлы әрі шығын мөлшерінің аз болуына есептеуіш техникалар және мобильді қосымшалардың әсер етер үлесі орасан зор. Техниканың үздіксіз даму сатысында тұрған уақытта тиімділіктерін пайдалана алмасақ, біршама мүмкіндіктерден айырылатынымыз анық. Шығынды азайту және мобильділікті арттыру кезкелген мейрамхана үшін және кәсіпорындар үшін экономикалық жағынан тек қана табысқа әкелетіні сөзсіз.

«Аспандияр Экспрес» мейрамханасында қазіргі таңда 250 аса тағам түрі дайындалады. Әрине келген клиент үшін әрбір тамақтың құрамын сипаттап отыру мүмкін емес. Мобильді қосымшаны пайдалана отырып осы түйінді шешудің ең оңтайлы жолы болмақ.

«Аспандияр Экспрес» мейрамханасының орта есеппен алғанда 1 күнде кемінде 1000-1500 адамға қызмет көрсетіп тамақтандырады. 250 ден аса тағам түрі дайындалатын жерде келген клиент үшін таңдау жасау оңайға түспес анық. Жұмыс жасайтын қызметші даяршылардың да шаршау және шадығу көрсеткіші жоғарылап, жұмыста салғырттық танытып, қызмет көрсету сапасын төмендететіні сөзсіз. Ал мобильді қосымша арқылы осы сұрақтардың жауабын табуға болады..

Мәзір мәліметтерін мобильді қосымша арқылы басқару және оны қолданысқа енгізу тиімді жолдардың бірі. Бұл бағдарлама жұмыс ауқымын кішірейтіп, қызмет көрсету бөлімінің жұмысын тездетіп, қағаз шығындарын қысқартуға және қызметкерлердің жұмысын жеңілдетуге үлкен үлес қосады.

Ал бағдарламалық қамтамасыздандырудың экономикалық маңыздары:

- мейрамхананың техникалық дәрежесін арттыру;
- жұмыс өнімділігін жоғарлату;
- мейрамхананың эксплуатациялық шығындарын азайту.

«Аспандияр экспрес» мейрамханасының мобильдік меню жүйесінің бағдарламасын іске асыру үшін капиталдық шығындар мен бағдарламаға арналған қосымша ағындық эксплуатациялық шығындар қажет.

Капиталдық шығындарға:

- бағдарламаны құру үшін қажетті техникалық құрылғылар;
- бағдарламалық қамтаманың құрылыстары;
- интернет желісі;
- материалдық шығындар жатады.
- Ағымдық эксплуатациялық шығындарға:
- бағдарламалық қамтама;
- бағдарламаны қалыпты жұмыс істеуіне кететін шығындар жатады.
- бағдарламаны іске асыратын бағдарламашыға төленетін жалақы және өңдеуге қажетті құрылғылар кіреді.

«Аспандияр Экспрес» мейрмаханасының мобильді қосымшасының медиа ақпаратты жүйесінің бағдарламасы үшін техника-экономикалық тиімділігін оны қолданар алдындағы және қолданғаннан кейінгі салыстырулар арқылы білуге болады, яғни бұл бағдарламалық қамтамамен жұмыс істеу ескі нұсқамен жұмыс істеуден гөрі оңай, әрі тиімді.

Ақпараттық жүйе жобасының бағалық көрсеткіштері:

- бағдарламаға кететін шығындар;
- жылдық эксплуатациялық шығындарды экономдау;
- бағдарламаға кететін капиталдық шығынының өтелу мерзімі.

Бұл дипломдық жобада мобильді қосымшаны әзірлеу жүзеге асырылады. Бұл бөлімнің мақсаты бағдарламалық өнімді әзірлеуге жұмсалатын шығындарды есептеу болып табылады. Есептеу нәтижесінде бағдарламалық өнімнің өзіндік құны болады.

Өзіндік құнды табу үшін ескеру қажет:

- бағдарламалық өнімді әзірлеудің еңбек сыйымдылығы;
- материалдық шығындар;
- еңбекақы төлеу шығындары;
- әлеуметтік салық;
- негізгі қорлардың амортизациясы;
- басқа да шығындар.

4.1 Бағдарламалық өнімді әзірлеудің еңбек сыйымдылығын есептеу

Бағдарламалық өнімді әзірлеу және жобалау әдістерінің көмегімен жүзеге асырылады. Жобалау кезеңі пәндік саланы талдауды, бағдарламалық өнімді жобалауды (интерфейс, құрылым, архитектура және т.б.) және техникалық тапсырмаларды (клиенттік интерфейсін әзірлеу, деректер базасын енгізу, тестілеу және т. б.) қамтиды.

Ең алдымен бағдарламалық өнімнің еңбек сыйымдылығын анықтап алуымыз қажет. Еңбек сыйымдылығы құн құрылымының негізгі бөлігін еңбек шығындары мен біздің жағдайда бағдарламаға кететін уақыт құрайды. Жобалау кезеңі бойынша бастапқы деректер 4.1-кестеде көрсетілген:

4.1 кесте – Жұмыстардың кезеңдері мен түрлер бойынша бөлу және еңбек сыйымдылығын бағалау

Әзірлеу кезеңі	Кезең атауы	Еңбек сыйымдылығы, адам × сағ.
1	Пәндік саланы талдау және зерттеу	36
2	Мобильді қосымшаны жобалау	48
3	Мобильді қосымшаның коталогтық бөлігін әзірлеу	96

4.1 кесте – жалғасы

4	Суреттік материалдар нысандарын әзірлеу	72
5	Бейнероликтер нысандарын өңдеу	104
6	Мобильді қосымшаның дизайндік бөлімін әзірлеу	36
7	Қажетті мәтіндер бөлімін әзірлеу	120
8	Тестілеу	72
9	Мобильді қосымшаны іске асыру	140
10	Енгізу	64
Барлығы		788

4.2 Бағдарламалық өнімді әзірлеуге арналған шығындарды есептеу

Бағдарламалық өнімді әзірлеуге арналған шығындарды анықтау тиісті сметаны жасау жолымен жүргізіледі, ол мынадай баптарды қамтиды:

- материалдық шығындар;
- еңбекақы төлеу шығындары;
- әлеуметтік салық;
- негізгі қорлардың амортизациясы;
- басқа да шығындар.

Материалдық шығындарға негізгі және қосалқы материалдардан (қағаз, картридждер және т.б. материалдар) тұрады, бағдарламалық өнімді әзірлеу үшін қажет емес энергияға шығындар кіреді. Материалдық ресурстарға жұмсалатын шығындардың жалпы сомасы (4.1):

$$Z_M = \sum P_i \cdot C_i \quad (4.1)$$

мұндағы, P_i – материалдық ресурстың i -ші түрінің шығыны, бірлік;

C_i – материалдық ресурстың i -ші түрінің бірлігінің бағасы, теңге;

i – материалдық ресурстың түрі;

n – материалдық ресурстар түрлерінің саны.

Материалдық ресурстар шығындар есебі 4.2-кестеде көрсетілген:

4.2 кесте – Материалдық ресурстарға шығындар

Материалдық ресурстың атауы	Өлшем бірлігі	Саны	Бірлік үшін бағасы, теңге	Сомасы, теңге
Acer Aspire 59YU	дана	1	139 900	139 900
Моноблок HP Pavilion	дана	1	349 900	349 900
Xiaomi Redmi Note 8 Pro смартфон	дана	1	39 900	39 900
USB флеш-жинақтаушы, 32 Гб	дана	1	2490	2490
Kingston (HDD) сыртқы қатты диск, 1 Тб	дана	1	18 900	18 900
HP LaserJet PRO M15w	дана	1	44 900	44 900
Қарындаштар	дана	8	80	640
A4 қағазы	бума	1	1090	1090
Қаламдар	дана	8	100	800
Барлығы:				598 520

Электр энергиясының шығынын есептеу қажет, өйткені бағдарламалық өнімді әзірлеу үшін электр жабдықтары қолданылады.

Электр энергиясына шығындар сомасын (4.2) формула арқылы есептеуге болады [23]:

$$Z_{э} = \sum_{i=1}^n M_i \cdot K_i \cdot T_i \cdot Ц \quad (4.2)$$

мұндағы, M_i – электр жабдығының i -ші паспорттық қуаты, кВт;

K_i – электр жабдығының i -ші қуатын пайдалану коэффициенті ($K_i = 0.9$);

T_i – бағдарламалық өнімді әзірлеу кезеңіндегі i -ші жабдықтың жұмыс уақыты, сағ;

$Ц$ – электр энергиясының бағасы, тенге/кВт×сағ;

i – электр жабдығының түрі;

n – электр жабдықтарының саны.

$$Z_{э(\text{ноутбук})} = 0,9 \cdot 0,9 \cdot 788 \cdot 18,32 = 11693,2 \text{ тг}$$

$$Z_{э(\text{моноблок})} = 0,9 \cdot 0,9 \cdot 788 \cdot 18,32 = 11693,2 \text{ тг}$$

$$Z_{э(\text{жарықтандыру})} = 0,3 \cdot 0,7 \cdot 788 \cdot 18,32 = 3031,5 \text{ тг}$$

$$Z_{э(\text{смартфон})} = 0,6 \cdot 0,7 \cdot 84 \cdot 18,32 = 646,3 \text{ тг}$$

$$Z_{\text{э(принтер)}} = 0,1 \cdot 0,7 \cdot 84 \cdot 18,32 = 107,7 \text{ тг}$$

Электр энергиясына кететін шығындар бағдарламалық өнімді әзірлеу кезеңінің ұзақтығына, мобильді қосымшаны жобалауға жұмсалған кВт/сағ санына және 1 кВт/сағ тарифіне сүйене отырып болады. Заңды тұлғалар үшін Алматы қаласы бойынша тариф 2020 жылы ҚҚС есебімен 1 кВт/сағ үшін 18,32 теңгені құрайды («Алматы ЭнергоСбыт» ЖШС ресми сайтында ұсынылған деректерге сәйкес).

Электр энергиясына шығындар 4.3-кестеде көрсетілген:

4.3 кесте - Электр энергиясына арналған шығындар

Жабдықтың атауы	Паспорттық қуаты, кВт	Электр жабдығының і-ші қуатын пайдалану коэффициенті	Бағдарламалық өнімді әзірлеу кезеңіндегі і-ші жабдықтың жұмыс уақыты, сағ	Электр энергиясының бағасы, тенге/кВт × сағ	Сомасы, теңге
Ноутбук	0,9	0,9	788	18,32	11693,2
Моноблок	0,9	0,9	788		11693,2
Жарықтандыру	0,3	0,7	788		3031,5
Смартфон	0,6	0,7	84		646,3
Принтер	0,1	0,7	84		107,7
Барлығы					27171,9

4.3 Еңбекақы төлеу шығындарын есептеу

Жалақыны есептеу үшін бағдарламалық өнімді әзірлеуді үш қызметкер жүргізеді:

- жоба жетекшісі;
- медиа әзірлеуші;
- графикалық дизайнер.

2020 жылы медиа әзірлеушінің орташа жалақысы 260 000 тг, жобалаушы үшін 180 000 тг, графикалық дизайнер үшін 210 000 тг.

Қызметкердің бір айдағы жұмыс сағаттары (4.3) формула бойынша анықталады:

$$Ч_M = N_M \cdot Ч_{рд} \quad (4.3)$$

мұндағы, $Ч_M$ – қызметкердің бір айдағы жұмыс уақыты;
 N_M – бір айдағы жұмыс күндерінің саны;
 $Ч_{рд}$ – бір күндегі жұмыс сағаттарының саны.

$$Ч_M = 24 \cdot 8 = 192 \text{ ч}$$

Қызметкердің сағаттық мөлшері (4.4) формула бойынша есептеуге болады:

$$ЧC_i = \frac{ЗП_i}{ФРВ_i} \quad (4.4)$$

мұндағы, $ЗП_i$ - i -ші қызметкердің айлық жалақысы, теңге;
 $ФРВ_i$ - i -ші қызметкердің айлық жұмыс уақыты қоры, сағ.

Жоба жетекшісі:

$$ЧC_i = \frac{180\,000}{192} = 937,5 \text{ тг}$$

Медиа әзірлеуші:

$$ЧC_i = \frac{260\,000}{192} = 1354,16 \text{ тг}$$

Графикалық дизайнер:

$$ЧC_i = \frac{210\,000}{192} = 1093,75 \text{ тг}$$

Бағдарламалық өнімінің дамуының күрделілігін анықтау үшін 1-кестенің деректерін пайдаланылды.

Жоба жетекшісінің бағдарламалық өнімін әзірлеудің еңбек сыйымдылығы 220 адам × сағ. (пәндік саланы талдау және зерттеу, мобильді қосымшаны жобалау, тестілеу және енгізу):

$$T_3 = 36 + 48 + 64 + 72 = 220 \text{ адам} \times \text{сағ}$$

Медиа әзірлеушінің бағдарламалық өнімін әзірлеудің еңбек сыйымдылығы 392 адам × сағ.:

$$T_3 = 96 + 36 + 120 + 140 = 392 \text{ адам} \times \text{сағ}$$

Графикалық дизайнердің бағдарламалық өнімін әзірлеудің еңбек сыйымдылығы 176 адам × сағ.:

$$T_3 = 72 + 104 = 176 \text{ адам} \times \text{сағ}$$

Еңбекке ақы төлеу шығындарының жалпы сомасы (4.5) формула бойынша анықталады:

$$Z_{EA} = \sum_{i=1}^n ЧС_i \cdot T_i \quad (4.5)$$

мұндағы, $ЧС_i$ – i -ші қызметкердің сағаттық мөлшері, теңге;

T_i – бағдарламалық өнімді әзірлеудің еңбек сыйымдылығы, адам × сағ;

i – қызметкердің санаты;

n – бағдарламалық өнімді әзірлеумен айналысатын қызметкерлердің саны.

Жоба жетекшісі:

$$Z_{EA} = 937,5 \cdot 220 = 206\,250$$

Медиа әзірлеуші:

$$Z_{EA} = 1354,16 \cdot 392 = 530\,830,72$$

Графикалық дизайнер:

$$Z_{EA} = 1093,75 \cdot 176 = 192\,500$$

Еңбекақы төлеу шығындары 4.4-кестеде көрсетілген:

4.4 кесте – Еңбекақы төлеу шығындары

Қызметі	Еңбек сыйымдылығы, адам×сағ	Еңбекақы төлеу, тг/сағ	Сомасы
Жоба жетекшісі	220	937,5	206 250
Медиа әзірлеуші	392	1354,16	530 830,72
Графикалық дизайнер	176	1093,75	192 500
Барлығы:			929 580,72

Қосымша жалақы:

$$З_{\text{қос}} = З_{\text{ЕА}} \cdot 10\%$$

$$З_{\text{қос}} = 929\,580,72 \cdot 0,1 = 92\,958,072 \text{ тг}$$

Еңбекақы қоры:

$$\Phi_{\text{ЕҚ}} = З_{\text{ЕА}} + З_{\text{қос}}$$

$$\Phi_{\text{ЕҚ}} = 929\,580,72 + 92\,958,72 = 1\,022\,538,8 \text{ тг}$$

Әлеуметтік салық:

$$Н_{\text{ӘС}} = (\Phi_{\text{ЕА}} - \text{ОПВ}) \cdot 13\%$$

мұндағы, ОПВ – міндетті зейнетақы жарналары-еңбекақы қорының 10%.

$$Н_{\text{ӘС}} = (1022538,8 - (1022538,8 \cdot 0,1)) \cdot 0,13 = 119637,04$$

Амортизациялық негізгі қорларды есептеу:

Амортизациялық аударымдардың жалпы сомасы (4.6) формула бойынша анықталады:

$$З_{\text{АМ}} = \sum_{i=1}^n \frac{\Phi_i \cdot H_{\text{Аі}} \cdot T_{\text{НИРі}}}{100 \cdot T_{\text{Эфі}}} \quad (4.6)$$

мұндағы, Φ_i – i -ші негізгі қордың бағасы, тг;

$H_{\text{Аі}}$ – i -ші негізгі қордың жылдық норма амортизациясы, %;

$T_{НИРi}$ – бағдарламалық өнімді әзірлеудің барлық кезеңіндегі i -ші негізгі қордың жұмыс уақыты, сағ;

$T_{Эфи}$ – i -ші негізгі қордың бір жылдағы жұмыс уақытының тиімді қоры, сағ/жыл;

i – негізгі қор түрі;

n – негізгі қор саны.

Негізгі қорлар амортизациясының жылдық нормасын есептеу:

$$H_{Ai} = \frac{100}{T_{Ni}} \quad (4.7)$$

$$H_{Ai} = \frac{100}{4} = 25$$

мұндағы, T_{Ni} – i -ші негізгі қорды пайдаланудың ықтимал мерзімі, жыл;

Бағдарламалық өнімді әзірлеу үшін MS Visual Code және Expo Client әзірлеу орталарының жұмыс уақыты 640 сағатты құрайды (мобильді қосымшаның коталогтық және медиа бөлігін әзірлеу, мобильді қосымшаны тестілеу және іске асыру):

$$T_i = 96 + 72 + 104 + 36 + 120 + 72 + 140 = 640 \text{ с}$$

Жабдықтар:

$$Z_{AM(\text{ноутбук})} = \frac{139\,900 \cdot 25 \cdot 788}{100 \cdot 1920} = 14\,354,32 \text{ тг}$$

$$Z_{AM(\text{моноблок})} = \frac{349\,900 \cdot 25 \cdot 788}{100 \cdot 1920} = 35\,901,19 \text{ тг}$$

$$Z_{AM(\text{смартфон})} = \frac{39\,900 \cdot 25 \cdot 84}{100 \cdot 1920} = 436,4 \text{ тг}$$

$$Z_{AM(\text{принтер})} = \frac{44\,900 \cdot 25 \cdot 84}{100 \cdot 1920} = 491,09 \text{ тг}$$

4.5 кесте – Негізгі қорлардың амортизациясы

Жабдық және БҚ атауы	Жабдықтың және БҚ құны, тг.	Жылдық амортизация нормасы, %	Жабдықтың және БҚ жұмыс уақытының тиімді қоры, сағ / жыл	Бағдарламалық өнімді әзірлеуге арналған жабдық және БҚ жұмыс уақыты, сағ	Сомасы, тг.
Ноутбук	139 900	25	1920	788	14 354,32
Моноблок HP Pavilion	349 900	25	1920	788	35 901,19
Xiaomi Redmi Not 8 Pro смартфоны	39 900	25	1920	84	436,4
HP LaserJet PRO M15w принтері	44 900	25	1920	84	491,09
Барлығы					51 183

4.6 кесте – Бағдарламалық өнімді әзірлеуге арналған шығындар сметасы

Шығындар баптары	Сумма, тг.
1. Материалдық шығындар	598 520
2. Электр энергия шығындары	27 171,9
3. Еңбекақы төлеу шығындары	929 580,72
4. Әлеуметтік қажеттіліктер	119 637,04
5. Негізгі қорлардың амортизациясы	51 183 0
6. Басқа да шығындар	
Барлығы	1 726 092,65864



4.1 сурет – Бағдарламалық өнімді әзірлеуге арналған шығындар сметасы

4.4 Бағдарламалық өнімнің ықтимал бағасын анықтау

Бағдарламалық өнімнің шарттық бағасы (4.8) формула бойынша есептеледі:

$$Ц_d = Z_{\text{НИР}} \cdot \left(1 + \frac{P}{100}\right) \quad (4.8)$$

мұндағы, $Z_{\text{НИР}}$ – бағдарламалық өнімді әзірлеуге жұмсалатын шығындар (4.6 кестедегі деректер бойынша), тг;

P – бағдарламалық өнім рентабельділігінің орташа деңгейі – 25%.

$$Ц_d = 1\,726\,092,65864 + 415\,385,14 = 2\,141\,477,8 \text{ тг}$$

Бұдан әрі өткізу бағасы қосылған құн салығын (ҚҚС) есепке ала отырып анықталады, ҚҚС ставкасы ҚР заңнамалық Салық кодексімен белгіленеді. 2020 жылға ҚҚС ставкасы 12% мөлшерінде белгіленген.

ҚҚС есебімен өткізу бағасы (4.9) формула бойынша есептеледі [23]:

$$Ц_p = Ц_d + Ц_d \cdot \text{НДС} \quad (4.9)$$

$$Ц_p = 2\,141\,477,8 + 2\,141\,477,8 \cdot 0,12 = 2\,398\,455,136$$

4.5 Бағдарламалық өнімнің экономика бөлімі бойынша қорытындысы

Бұл бөлімде бағдарламалық өнімнің еңбек сыйымдылығын есептеу жүргізілді. Есептер бұл жоба жоғары кондицияны, қаржылық және материалдық ресурстарды талап ететінін көрсетеді. Сондай-ақ, бұл бағдарламалық өнімнің басты өзектілігі – мобильділік.

Осылайша, мейрмахана желісіне арналған мобильді қосымшаны өткізу бағасы ҚҚС есебімен 2 398 455,136 теңгені құрайды, өзіндік құны 1 983 069,996 теңгеге тең, рентабельділік (пайда) 415 385,14 теңгеге тең.

5 Өміртіршілік қауіпсіздігі

5.1 Адам тіршілік қауіпсіздігі мен еңбек қауіпсіздігі

Тіршілік қауіпсіздігі бұрынғы заманнан бері адамзат баласының негізгі мәселелері болып табылады. Еңбек қауіпсіздігі – тіршіліктің қауіпсіздігінің аса маңызды компоненті, себебі еңбек қызметі адамның тіршілігі мен дамуының негізі болып табылады. Сонымен қатар, адамның іс-тәжірибесі кез келген қызмет-әлуетте қауіпті деп пайымдауға негіз береді.

Қауіпті және зиянды өндірістік факторларды болдырмайтын немесе алдын алуда қолданылатын қорғаныс құралдыр адам ағзасы мен қоршаған орта үшін ыңғайлы және еңбек шарттарына оптималды болу керек.

Қорғаныс құралдар қауіпті және зиянды заттардан болмау керек. Олар эстетика және эргономика талаптарына сай, қорғаныстың жоғарғы дәрежесінде болуы қажет.

Барлық қорғаныс құралдары өзінің функционалдық мақсаттарына тәуелді үш топқа бөлінеді:

- адамға қауіп төндіретін құралдар;
- қауіп және зиянды өндірістік факторлардың адамға тигізетін әсерін төмендететін немесе болдырмайтын құралдар;
- алдын ала емдеу және алғашқы медициналық көмек көрсету құралдары.

Қорғаныс құралдары қолдану сипаты бойынша ұжымдық қорғаныс құралдар және дара қорғаныс құралдары болып екіге бөлінеді. Біздің жағдайымызда ұжымдық қорғаныс қолданылады. Ұжымдық қорғаныс құралдары:

- өндірістік ғимарат пен жұмыс орынының ауасын қалыпты ұстау – барометрлік қысым көсеткіштерін қалыпты ұстайтын қондырғы; вентиляция жүйелері, ауаны тазалау және кондициялау, зиянды факторлар әсерін жоятын қондырғы; жылыту жүйелері; автоматты бақылау және ауаны тазартатын қондырғы;
- өндірістік ғимарат пен жұмыс орындарын қалыпты жарықтандыру – жарық көзі және жарықтандырғыш ондырғылар, оптималды жарық ойықтар, жарық сүзгіштер;
- электр тоғымен жарақаттаудан қорғану – бөлектеу қондырғылары; автоматты сөндіру қондырғылары, кернеуді төмендету үшін арналған қондырғылар және қауіпсіздік белгілері.

Кадр бөлімшесінің қауіпсіздігі ретінде:

- бөлменің ауасын қалыпты ұстау;
- бөлменің жарығын қалыпты жарықтандыру;
- бөлменің электр тораптарының және электр аппараттарының өрт қауіпсіздігі;
- бөлменің санитарлы-гигиеналық жағдайының қауіпі.

5.2 Дербес компьютердің адамға тигізер әсері және соған қойылатын талаптар

Биологиялық ғылыми-зерттеу институтының зерттеулері бойынша әртүрлі аурулардың көрсеткіштері компьютермен жұмыс істеу уақытының өсуіне тура пропорционал.

Жеке компьютерлер іске қосылғаннан кейін жұмысшы-оператор аумағында құрылысы жағынан өте күрделі электромагниттік өріс пайда болады. Электромагнитті өрістердің адамның ең маңызды ағза жүйесіне кері әсер ететіні дәлелденген. Олар біріншіден жүйке жүйесіне, иммундық жүйеге, эндокриндік бездерге, жыныс мүшелерінің жүйелеріне әсер етеді.

Электромагнитті өрістер (монитор арқылы бөлінетін) адамның миындағы гармондарда әртүрлі гармональдық өзгерістер болады. Ең төменгі электромагнитті өрістер басқа да зиянды факторлармен қосылып, кейде қауіпті ісік ауруларын (рак), қанның рак ауруларын (лейкемия) тудырады. Монитордың электромагнитті өрістерді өзіне тарататын шаң – тозаңдар көбіне бет терісінің қышыма ауруларын, асматикалық өкпе ауруларының қозуына әсерін тигізеді.

Электромагниттік сәуле - гамма сәулесінің (рентген сәулелерінің) бір түрі. Ол тек компьютердің мониториынан ғана шығып, тура сызық бойымен таралмайды. Ол тұрған кеңістіктің бәріне ауадай жайылып тұрады. Қазіргі компьютер жиынтықтарының барлық құрылғыларында қорғаныш жабыны мүмкіндігінше жан жағынан жасалған. Әсіресе, ескілеу модельдердің тек экраны ғана магнитке қарсы қорғанысқа ие. Ал, процессор блогының тоқ жалғайтын артқы жағы мен екі жақ бүйірінен шығатын толқындар аса зиянды.

Дербес жиынтық компьютерде отырған адам екі жақты электромагниттік толқындардың ортасында отырады. Оның біріншісі – монитор, екіншісі жүйелі блок, яғни процессор. Адамға ең зиянды 40-70 ГГц толқындарын тарататын осы жүйелі блок болып табылады.

Адамның көз көруі компьютерлік бейнелеуге дағдаланбаған біз әртүрлі түстерді заттарды кәдімгі күнгі жарықта көруге жаралғанбыз. Экрандағы көрініс басқаша әртүрлі жарқырауық сәуледен нүктелерден тұрады. Экранның бұлдырауы көптеген түрлі-түсті түстердің жарқырауы көздің тез арада шаршауына әкеліп соғады: көздің және маңайының ашуы, тез арада шаршауы, шаншу, жауырын маңындағы қатты аурулар, көз қабағының қызаруы, көз жасаурауы, құрғақ болып көрінуі, мойын бұлшық етінің ауруы «компьютерлік көз стресі синдромына» (компьютерный зрительный синдром) айналады. Бұл термин 1992 жылы америкалық оптометрикалық бірлестік енгізді. Тәжірибелердің көрсетулері бойынша жоғарғы белгілер «компьютерлік көз синдромы» қолданушылардың 6-7 сағат жұмыстан кейін пайда болады. Әртүрлі хабарламалардың адам миына жетіп, есте қалуының 87% көру органы рецепторлары, ал 4% басқа органдарға байланысты екенін ескереді. Компьютермен 3 жылдан аса уақытта жұмыс істейтіндер арасында олардың жүйке-жүйелерінің әлсіреуі типті иммунно-биологиялық ауыртпалықтарды да

байқауға болады. Көбінде электрлік өріс әсеріне өзінің денесіндегі тоқ разряды көбеюі салдарынан тоқ соғу болады.

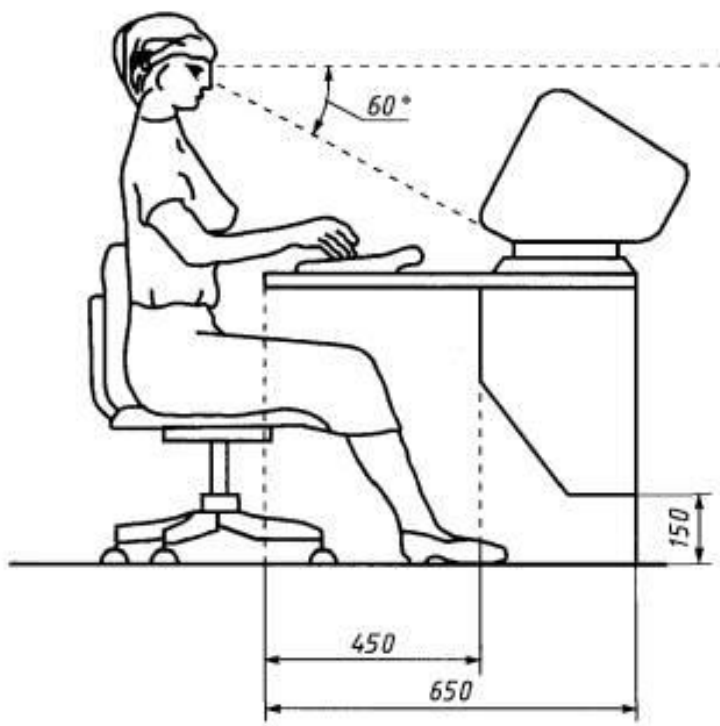
Көздер мәтін немесе суреттің майда дірілдеулерін, перденің жылтырауын тез тіркейді. Көз шамадан тыс салмақ әсерінен көру қабілетін жоғалтады. Кілегейленген түсті сәтсіз таңдап алу, шрифтер, терезелердің құрастырылымы, перденің теріс орналастырылуы көзге теріс әсерін тигізеді.

Компьютерлік техниканың жүрекке әсері бірнеше түрлі әдіспен жүзеге асады. Олардың біразы техниканың бірден әсері болса, ал енді біреулері бірінші бір ағзаға әсерін тидіріп, кейін ол ағзаның жұмысының әлсіреуі ішкі ағзалардың жұмысына кедергі келтіреді.

Монитордың жұмыс істеу әсері әр кезде қолданушының жасына оның көру органын және жұмыс орнын ұйымдастыруына да байланысты.

Дербес компьютер мониториянда мынадай электромагнитті өріс көздері болады:

- электростатикалық өріс;
- әлсіз электромагнитті сәулелер бөлінеді, төменгі және жоғарғы диапазондағы (2 Гц – 400 кГц);
- рентгендік сәулелер, күлгін түсті сәулелер, инфрақызыл сәулелер.



5.1 сурет – Адамның компьютер алдында дұрыс отыру ережесі

Адам компьютерде босаң қалыпта отырады, ол денесінің қандай тұрғыда қисайғанына мән бере бермейді. Бұл ағзаға кері әсер етеді: омыртқа, мойын, бастың бұлшық еті, қол және иықтарға артық салмақ түскендіктен балаларда – сколиоз, остехондроз аурулары пайда болады.

Компьютерде жұмыс істейтін адамдардың денсаулық жағдайына теріс ықпал ететін негізгі факторлар:

- компьютер алдында өте ұзақ уақыт отыру;
- монитордың электромагнитті сәулеленуінің әсері;
- көзге салмақ түсуі;
- шаршау;
- білезіктердің буындарына шамадан тыс салмақ түсуі;
- ес жоғалтуына орай стресс жағдайы.

Адамның қозғалмай ұзақ уақыт қозғалмай үздіксіз жұмыс істеуі, қан қысымын көтереді. Миопия, яғни көз жанарының алыстан немесе жақыннан көру қабілеті нашарлайды. Құлақтың есту қабілетін төмендетеді, ұйқы қабілеті төмендейді. Сонымен қатар, омыртқаға көп күш түсетіндіктен (сколиоз омыртқа буынының қисаюы) омыртқа ауруына шалдығуы мүмкін. Электрленген сәулелер саусақ ұштарына әсер етіп, қолдың буындарына зақым келтіреді.

Монитордың алдында отыратын кез келген адамға әсер ететін электромагниттік сәулеленудің зияндылығы өте көп. Көзге көрінбейтін, қолға білінбейтін ұсақ жиіліктегі толқындар жүрек қан тамыр жүйесінің қызметін бұзады. Кеңсе қызметкерлерінің бойын таратып, орнынан мүлде қозғалмауы, бір уақыт сергіп компьютерден дем алмауы сәулеленуді көбейткен үстіне көбейте береді. Ұзақ қозғалыссыз отырғанда, аяқ-қол қан тамырлары, бұлшық ерттері тартылып, қозғалысқа мүлде икемсіз юолып қалалы, күш қол білезігі мен көзге түседі. Қан айналымның бұзылуы жүрекке кері әсерін тигізеді.

Жүрек қан тамыр жүйесі мен жүйке жүйесі үздіксіз электромагниттің зардабын алдымен қабылдайды. Қан құрамы, қан жасушаларының жаңалауының баяулығы, ақ қан ауруымен ауруы, көз жанарының түрлі аурулары байқалады. Кеңсе қызметкерлерінің иммундық жүйесі қорғаныш қызметін атқаратын арнайы ферменттерді қанға беруін тоқтатып жасушалы иммунитет жүйесінің әлсіреуін тудырады. Эндокриндік жүйе болса, керісінше қанға өте үлкен мөлшерде адреналин бөледі де, жүрекке көп салмақ түседі. Қан қоюлана бастайды, қою қаннан жасушаларға оттегі жетіспейді. Ұзақ уақыт электромагниттік сәулеленуге ұшыраған адамның ас қорытуында, гармональды жақтан да орын алады.

Компьютер алдында мыналарды ескеру қажет:

- компьютердің монитору мен беттің арақашықтығы кемінде 50 см болуы қажет.
- компьютер тұрған бөлменің ауасы жиі тазарылуы тиіс.

5.3 Жұмыс бөлме зонасындағы еңбек жағдайын талдау

5.3.1 Жұмыс бөлмелеріндегі микроклимат

Жұмыс бөлмелердегі микроклимат қоршаған ортаның маңызды физикалық факторы болып табылады да, жұмыскерлердің денсаулық жағдайы

мен іскерлігі осыған байланысты болады. Жұмыс саласында жұмысшылардың денсаулық жағдайы мен іскерлігін болжаудың негізі ретінде микроклиматты құраушы негізгі заңдылықтарын, профилактика және гигиеналық бағалау міндеттері өзекті болып келеді.

Микроклимат дегеніміз – шектеулі кеңістіктегі физикалық факторлардың, яғни ауа температурасы, ауа ылғалдылығы, сәулелі жылулық пен қозғалыс жылдамдығынан құралатын және адам организмінің жылулық алмасуына әсер ететін жағдай.

Жылулық комфорт – адам организміне қолайлы немесе ыңғайлы болып келетін жайлы ахуал. Көбіне жұмыс бөлмесінде кондиционерлер қолданылады. Ол температураны қалыпқа келтіру үшін қажет. Жаз мезгілінде бөлмені суыту үшін, ал қыс мезгілінде ауаны жылыту үшін пайдаланылады. Бірақ кондиционердің де адамға жағымсыз жері бар, автоматты жасанды суытатын құрал болғандықтан, адам денесіне кері әсерін де тигізеді. Қазіргі кезде адамның тамақ ауруына немесе нерв жүйесіне әсер етуі осы кондиционердің әсері. Сол себепті оны қолданғанның өзі де қауіпті.

Жұмыс бөлімшесінде тек ДК алдында отырып жұмыс атқаратындықтан физикалық ауырлық жоқ. Сондықтан қарапайым мына шаралар ескерілуі қажет:

- ауа температурасының оптималдылығын - 22°C (20-24°C - рұқсат етіледі);
- ауаның оптималды ылғалдылығын - 40 -60% (75% - рұқсат етіледі);
- ауаның қозғалыс жылдамдығы 0.1м/с-тан аспауы керек.

5.3.2 Жарықтандыру талаптары

Жұмыс орындағы жарықтандыру адамның көру жұмыс қасиеттігіне, физикалық және моральдік әсерін тигізеді.

Еңбекке жақсы әсерін тигізу үшін өндірістегі жарықтануға қойылатын негізгі жарық техникалық талаптар:

- жұмыс орындарын жарықтандыру деңгейі орындалатын жұмыстың сипатына сәйкес болуы қажет;
- уақыт бойынша тұрақты және бір қалыпты жарықтануы;
- қатты жылтырау мен айқындылықтың жоқ болуы;
- жұмыс орындарында диффузиялық - жарық ағыны болуы керек (яғни көлеіңкесіз жарықтандыру);
- жұмыс орнында басқа да зиянды факторлар тудармауы керек (шу және т.б.) және де электр және өрттен қауіпсіз болуы керек.

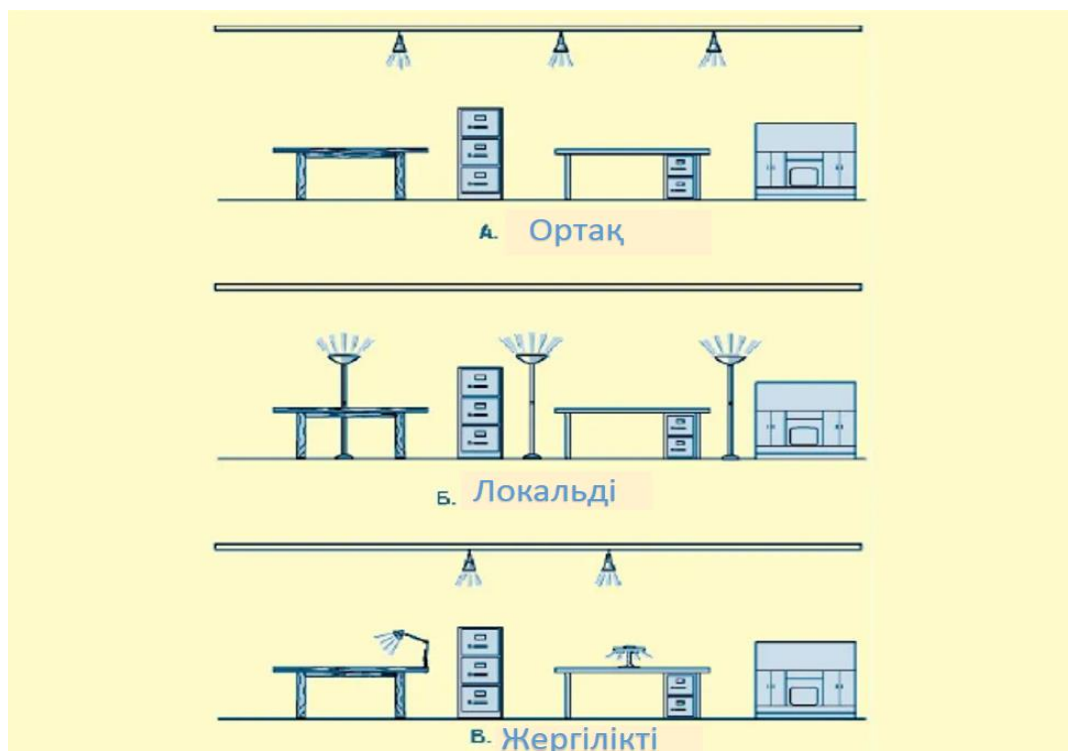
Жасанды жарықтандыру табиғи жарық жоқ болмағанда немесе жеткіліксіз болғанда қолданылады. Жасанды жарықтандыру үшін қызу шамы, газразрядты шам, жазық жарықжүргіштер қолданылады.

Жасанды жарықтандыру жүйесінің типіне байланысты:

- жергілікті – жарық ағыны тек жұмыс жерінде ғана таралады;
- ортақ – біркелкі және локальді болып бөлінеді;
- аралас – ортақ және жергілікті жарықтандырулардың қосындысы.

- Жасанды жарықтандыру тағы да бөлінеді:
- авариялық, егер жұмыс жарығы өшіп қалса қосылады (барлық жарықтан 5%);
- жұмыстық – барлық бөлмеде және территорияда қойылады;
- эвакуациялық – адамға қауіп төндіретін жерлерде қойылады (0,5 лк – мекемедегі жарықтық, 0,2 лк – сыртында).

Жарық көздерінің орналасқаны бойынша жасанды жарықтанулар: жалпы, жергілікті және құрамаға бөлінеді. Жалпы жарықтануда шамдарды жұмыс орындарына біқалыпты жарық беру үшін жоғарыда орналастырады. Жергілікті жарықтануда жарық ағыны жұмыс орындарына тікелей шоғырланады. Осы жағдайда жұмыс орны мен қоршаған кеңістік жарықтануының күрт айырмашылығы болады, жұмыстың жағымсыз жағдайлары құрылады, зақымдану қауіпі өседі. Бір жергілікті жарықтандыруды пайдалану тек кезеңдік жұмыста пайдаланылады. Құрама жарықталуда жалпы жарықтандыруға жергілікті жарықтандыру қосылады.



5.2 сурет - Жасанды жарықтандыру түрлері

5.3.3 Электр және өрт қауіпсіздігі

Жұмыс бөлімшесінде тоқсыз ештеңе жұмыс істемейді. Мысалға алатын болсақ, дербес компьютер, принтер, сканер және т.б. тоқтың көмегімен жұмыс атқаратын техникалық заттар бар. Ал тоғымыз электр болып табылғандықтан, біз электр қауіпсіздігі мен өрт қауіпсіздігін ескеруіміз қажет.

Электр қауіпсіздігі дегеніміз – ол, электромагниттік өрістің, статикалық электрленудің, электрлік доға мен электр тоғының зиянды және қауіпті

әсерінен адамдарды қорғауды қамтамасыз ететін ұйымдастырылған және техникалық жұмыстар мен шаралардың жүйесі.

Егер адамның екі нүктесі арасында потенциалдар айырмасы болса, онда адам денесі арқылы электр тоғы жүреді. Адам бір уақытта жанасқан екі нүктелік тоқ тізбегі арасындағы кернеу -жанасу кернеуі деп аталады.

Дене арқылы жүретін электр тоқ адамға жылулық, биологиялық және электролиттік әсер етеді.

Тоқтың жылулық әсері электр энергиясының жылуға айналуында сезіледі және ол терінің, тканның және қан тамырларының қызуын тудырады.

Тоқтың биологиялық әсері тоқтың бұлшық еттер арқылы жүруінде оның қысқаруын тудырады.

Тоқтың электролиттік әсері қан құрамының өзгеруіне алып келеді.

Электр тоғына түсіп қалғанда төмендегі зақымдалулар болуы мүмкін:

- күйіп қалу;
- терінің металдануы;
- электр белгілері;
- электроофтальмия;
- электр соққысы;
- механикалық зақымдалулар.

Электр соққысы болғанда, адам организмі жалпы зақымданады, яғни нерв және жүрек тамырларының бұзылуы, бұлшық еттерінің тырысуы пайда болады.

Механикалық зақымдалулар (тканның бөлшектенуі, сынықтар) адам бұлшық еттерінің тырысуы және де электр тоғының әсерінен төбеден құлау нәтижесінде болады.

Электр тоғынан зақымдалу сипаттамасы тоқтың тегі мен мәніне, оның жүріп өту жолына, әсер ету ұзақтығына, адамның жеке физиологиялық ерекшелігіне және оның зақымдалған кезіндегі жағдайына байланысты болады.

Электр тоғымен жұмыс істейтін қондырғылардан қорғану шаралары:

- тоқ өткізгіш бөліктерді қауіпсіз жерге орналастыру;
- қорғаныстық жерге қосқыш;
- автоматтық ажырақыш;
- тоқты өшіріп жүру;
- сулы киіммен немесе сулы қолмен ДК қолданбау;

Өрт – бұл арнайы жасалмаған, материалдық шығындарымен қатарласып жүретін, кейде адам өлімі болуы мүмкін, бақылана алмайтын ошақ көзі.

Барлық өрт қауіпсіздігінен орындалатын шаралар 4 түрге бөлінеді:

а) кәсіпорынды жобалаған кезде болатын техникалық шаралар, олар:

- ғимараттың өртке төзімділігін анықтау;
- ғимарат подъездерінің жобалануы;
- ғимараттардың өртке қарсы ара қашықтығын ескеру;
- ғимаратты найзағайдан қорғау;

б) Эксплуатациялық шаралар:

- ұйымдастыру шаралары;
- режімдік шаралар.

Өрттің алдын-алу бойынша жасалатын шаралар:

- жұмысшыларға нұсқау беру;
- өрт сөндіру әдістері мен заттары.

Су - жоғары жылуды өзіне сіндіру қасиетіне байланысты тотықтандырғыш концентрациясын азайтады. Кемшіліктер: ток өткізгіштігі, судың тығыздығы жоғары болғандықтан органикалық сұйықтықтарды сөндіре алмайды, қысқы мерзімде су қатып тайғақ мұзға айналады.

Су құбырлары бар қалаларда өртке қарсы арнайы су құбырлары бар:

- сыртқы, барлық барлық ірі кәсіпорындарда бар. Бұл су құбырлары ғимараттың периметрі бойынша орналасқан, әр 100 м сайын жер асты, не болмаса жер үстінгидранттары (құдықтар, люктер) орналасқан;
- ішкі, ғимараттардың ішінде. Бұл су құбырлары ғимарат коридорында белгіленген ара қашықтықта, төменде орналасқан арнайы крандармен жабдықталған су құбырлары .

Ғимараттарда су, арнайы автоматикалық өрт сөндіргіштерінде пайдаланылады, 26-суретте көрсетілген:

- спринкерлік;
- дренчерлік.



а) спринкерлік



б) дренчерлік

5.3 сурет - Арнайы автоматикалық өрт сөндіргіштер

Эвакуациялық шығу жолдары деп - сыртқы шығатын шығу жолдары немесе баспалдақ торына апаратын (ол жол сыртқа шығатын жолдар болуы тиіс) жолдарды айтамыз. Жұмыс орны мен шығу есігінің ара қашықтығы 60 м аспауы қажет. Барлық ғимараттарда міндетті түрде эвакуация жолдары, шығу есіктерінің көлемі есептеліп СНИП (ҚНЖЕ) бойынша салыстырылады.

5.4 Мейрамханадағы өрт сөндіру құралдарын таңдау

Өрт сөндірудің нормативті параметрлері болып табылады:

- берілген жанғыш заттар мен материалдарды сөндіруге әртүрлі өрт сөндіру заттарының жарамдығы;

- берілген әдіспен берілген заттар мен матреиалдарды сөндіру үшін қажетті өрт сөндіру заттарының үдемелі мөлшері;
- сөндіру уақыты;
- өрт сөндіру заттарын берудің интенсивтілігі.

Мейрамханалардағы өртті сөндіру үшін әртүрлі өрт сөндіру құралдары пайдаланылады. Суды және ылғалдандыру сулы ерітінділерін шатырдағы, көрші бөлмелердегі, науқастардың бөлмелерден, дәрігерлер кабинеттеріндегі, коридорлардағы және т.б. өрттерді сөндіру үшін пайдаланады.

Өрттерді сөндіру үшін, ереже бойынша, РСК-50 және РС-50, шашыратқаш және тұтас ағатын оқпандарын қолданады, ал өрт дамыған жағдайларда, әсіресе отқа төзімділігі IV дәрежелі ғимараттарда аса күшті оқпандарды қолданады. Өрт сөндіру үшін қажетті оқпандардың санын $0,1 \text{ л}/(\text{м}^2 \text{ с})$ кем емес су берудің қажетті интенсивтілігін ескеріп анықтайды. Қалыптасқан жағдайға және күш пен құралдарға байланысты ӨСБ бір мезгілде адамдарды құтқару бойынша және өрт сөндіру жұмыстарын жүргізе алады. Егер күш пен құралдар бір мезгілде екі жұмысты жүргізуге жеткіліксіз болса ӨСБ барлық күш пен құралдарды адамдарды көшіруге пайдаланады немесе өртті тез сөндіріп адамдарға қауіпсіздікті қаматамасыз етуге сенімді болса оқпандарды беруге және науқастардың үрейін басуға пайдаланады. ӨСБ және жауынгерлік аймақтардың командирлері сөндірумен бірге бір мезгілде қымбат тұратын қондырғылардың, дәрі-дәрмек қорларын, рентген пленкаларының, газбен баллондардың, жеңіл тұтанғыш сұйықтықтардың санын тез арада оларды өрттен, түтіннен және төгілген судан қорғау үшін күш пен құралдарды енгізеді, ал қажет жағдайда оларды көшіруді ұйымдастырады.

Дабылмен жеке құрам көтерілгенде кіріп шығатын жерлерге заттар тастауға, бір бірінен кедергі жасауға, тез әрі жылдам киінугі міндетті. Көлікке ӨСБ бастығы бекіткен тәртіп бойынша отырылады. Толық отырылған соң есік жабылғаннан кейін көлік жүру керек. Жүріп келе жатқан көлік ішінде темекі шегуге, терезеден басын шығаруға тиым салынады. Аяқ қоятын жерде тұруға тиым салынады. Көлік жүріп келе жатқанда жүргізуші жауапты «жол ережесін» қатаң орындау керек. Шұғыл жағдайдан басқа кезде дауысты дабыл беруге тиым салынады. Түнде өрт автокөлігі тұрған кезде жарықты дабыл қосылады.

Барлауға екі адам жіберіледі, газ қағармен жұмыс істегенде үш адамнан жіберіледі. Барлауға дайындығы бар командир жіберіледі. Жер төлелерге жіберілгенде бес адамнан звено құрылады. Барлау жасағанда өртеніп жатқан жердің ТҚ сақтау керек. Адамдар мен құнда заттарды құтқарғанда жауынгерлік жарғыдағы қауіпсіздік шараларын орындау қажет. Ғимарат конструкцияларының және электр жүйелерінің қауіпсіздік шараларын сақтау керек.

«Қанат жаю» командасы бойынша

Ең оңай жол тандап жең тізбегін тарту және құрал саймандарды апару керек. Автокөлікті ыңғайлы етіп қойып магистрал жеңін тізбегін жүргізу керек, басқа келген күш құралдарға кедергі келтірмейтіндей етіп. Өрт көлігі

толық тоқтағанша қанат жаюға тиым салынады. Гидрант т.б. құдықтарды ашық тоқпен жарықтандыруға. Биікте жұмыс істегенде арқанмен сақтану керек. Жеңдерге жең ұстағыштарды қолдану керек. Өртті сөндіргенде арнайы қауіпсіздік киімін кию керек. Бөлменің улы түтін концентрациясын азайту үшін суды шашыратып беру керек. Өрт сөндіруге қатысып жатқан қызметкерлер конструкциясының өзгерістеріне қарап тұру керек. Биіктікте жұмыс істеп жатқанда сақтандыру шараларын қолдану керек.

5.4.1 Өрт қауіпсіздігі жағдайында эвакуациялану уақытын есептеу

Өрт болған жағдайда мейрамхана ішіндегі барлық адамдарды эвакуациялау өте маңызды бөлік болып табылады. Эвакуациялау барысында қандай да бір қателіктер болған жағдайда немесе орындалмаған жағдайда көптеген адамдардың өміріне қауіп төніп, орны толмас шығынға әкеліп соғады. Әрбір мейрамхана орындары өрт жағдайында маңызды болып табылатын эвакуацияланудың алдын-ала сақтық шараларын жасауға міндетті. Сонымен қатар, әрбір мейрамхана қызметкерлерін халықты эвакуациялаудың дұрыс жолдарын үйретіп, техника қауіпсіздіктерін толық қанды таныстыруы тиіс. Мейрамханаларда өрт қауіпсіздігінің алдын алатын барлық талаптар, яғни, дыбыс қағарлар, өрт сөндіргіш балондар, апаттық шығу сызбасы, өрт сөндіру қызметінің ақпараттары және т.б. шұғыл қажетті болатын құрал саймандар болуы тиіс.



5.4 сурет - Мейрамхана үшін эвакуация сызбасының мысалы

Эвакуациялау жолының учаскесі бойынша 2 адамдар ағыны $N_1 = N_2 = 300$ адам қозғалады, 30м кейін $N_1 = 300$ адам ЕН арқылы басқышқа және одан әрі Мега 1-қабатына, Содан соң Е шығаберісі арқылы, РТ06-ден өткен ағынға қосылып, далаға шығады.

Эвакуациялау жолы учаскесінің ені 4м.

1) Ағын тығыздығын есептеу

$$D = \frac{N \cdot f}{l \cdot \sigma}, \quad (5.1)$$

мұндағы N – учаскедегі адам саны, адам.

f – адамдардың көлденең проекциясының орташа ауданы, $m^2/адам$.

l – эвакуациялау жолы учаскесінің ұзындығы, м.

σ – эвакуациялау жолы учаскесінің ені, м.

$$D = \frac{600 \cdot 0,125}{30 \cdot 4} = 0,625 \text{ м}^2/\text{м}^2$$

Осы нақты жағдайда $q = 16,2$ м/мин, ал $q_{max} = 16,5$ м/мин.

2) Адамдардың топталуы түзіледі

Қабылдайтынымыз $q = 16,5$ м/мин, $v = 33$ м/мин

$$t_1 = \frac{l}{v}, \quad (5.2)$$

мұндағы l_1 – эвакуациялау жолының бірінші учаскесінің ұзындығы, м.

v_1 – бірінші учаскеде эвакуацияның көлденең жолы бойынша адам ағынының таралу жылдамдығы, м/мин.

$$t_1 = \frac{30}{33} = 0.91 \text{ мин.}$$

Бірақ адамдардың топталуына байланысты ағынның өту уақыты кідіріс уақытына артады

$$t_3 = N \cdot f \left(\frac{1}{q_{\text{при } D=0,9 \cdot b_{i+1}}} - \frac{1}{q_{i \cdot b_i}} \right), \quad (5.3)$$

мұндағы N – адам саны, адам.

f – көлденең проекцияның ауданы, m^2 .

$q_{\text{при } D=0,9}$ – 0,9 тығыздығы жағдайында $i+1$ учаскесі арқылы қозғалу қарқындылығы, м/мин.

b_{i+1} – оған кірген кезде адамдардың топталуы түзілген учаскенің ені, м.

q_{i+1} – i учаскесіндегі қозғалу қарқындылығы, м/мин.

b_i – алдыңғы i учаскесінің ені, м.

$$t_3 = 600 \cdot 0,125 \left(\frac{1}{16 \cdot 4} - \frac{1}{16,2 \cdot 4} \right) = 75 (0,0156 - 0,0154) = 0,02 \text{ мин.}$$

3) Бірінші учаске бойынша адамдар ағынының қозғалу уақыты

$$t_1 = 0.91 + 0.02 = 0.93 \text{ мин.}$$

4) РТ05 ойығы арқылы қозғалу қарқындылығын есептеу

$$q_{\text{пр}} = \frac{q_{\text{эт}} \cdot \sigma_{\text{эт}}}{b_{\text{пр}}} = \frac{16,5 \cdot 4}{3} = 22 \text{ м/мин}$$

$q_{\text{max пр}} = 19,6$ м/мин, осыдан талап етілетін жайт, РТ05 есік ойығында адамдардың топталуы түзіледі, демек ағын қозғалысы t_3 уақытына баяулайтын болады

$$t_3 = N \cdot f \left(\frac{1}{q_{\text{при } D=0.9 \cdot b_{i+1}}} - \frac{1}{q_{i \cdot b_i}} \right) = 300 \cdot 0,125 \left(\frac{1}{19,6 \cdot 3} - \frac{1}{22 \cdot 3} \right) = 0,075$$

мин.

5) Бірінші қабатқа дейін басқыш марштары бойынша таралу адамдар ағынының босауын ескере отырып жүретін болады

б) Басқышпен төмен қозғалған жағдайдағы ағын тығыздығы

$$D = \frac{300 \cdot 0,125}{19 \cdot 2,5} = 0,79 \text{ м}^2/\text{м}^2$$

Сонда $v_{\text{л}} = 13$ м/мин, $q_{\text{л}} = 10,4$ м/мин

$$t_{\text{л}} = \frac{l_{\text{л}}}{v_{\text{л}}} = \frac{19}{13} = 1,462 \text{ мин.}$$

7) Бірінші қабаттағы есік ойығы арқылы қозғалу қарқындылығын есептеу

$$q_{\text{пр}} = \frac{q_{\text{эт}} \cdot \sigma_{\text{эт}}}{b_{\text{пр}}} = \frac{10,4 \cdot 2,5}{3} = 8,7 \text{ м/мин}$$

8) Бірінші қабат бойынша қозғалу қарқындылығын есептеу

$$q_{1\text{эт}} = \frac{8,7 \cdot 3}{4} = 6,5 \text{ м/мин}$$

9) Есік ойығынан Е шығаберісіне дейінгі учаскенің ұзындығы. $l = 40$ м, $v = 90$ м/мин.

$$t = \frac{l}{v} = \frac{40}{90} = 0,444 \text{ мин}$$

10) Е шығаберісі алдында РТ05 және РТ06 ағындарының бірігуі күтіледі

11) Бірігетін ағындардың қозғалу қарқындылығын есептеу

$$q_{\text{сл}} = \frac{q_i \cdot \sigma_i + q_{\gamma} \cdot \sigma_{\gamma}}{\sigma_i}, \quad (6.3)$$

мұндағы q_i – учаскенің басында бірігетін адамдар ағынының қозғалу қарқындылығы, м/мин.

b_{i+1} – бірігуді эвакуациялау жолдары учаскелерінің ені, м.

b_i – эвакуациялау жолының қаралатын учаскесінің ені, м.

$$q_{\text{сл}} = \frac{6,5 \cdot 4 + 9 \cdot 4}{4} = 15,5 \approx 16 \text{ м/мин}$$

$v = 40$ м/мин, $t_{\text{пот}} = 1$ мин.

12. Эвакуациялаудың жиынтық уақытын есептеу

$$t = 0,93 + 0,075 + 1,462 + 0,444 + 1 = 3,911 \text{ мин.}$$

5.5 Өміртіршілік қауіпсіздігі бөлімі бойынша қорытынды

«Өміртіршілік қауіпсіздігі» бөлімінде адамның жеке бас қауіпсіздігін сақтайтын іс-шаралармен таныстым. Сонымен қатар, айтып келмейтін жаулардың бірі өрттен сақтану жолдарын және оған қажетті құрал-саймандар,

іс-әрекеттерді білдім. Өрт қауіпі туындаған жағдайда қоғамдық орындарда ең алғашқы қадамдардың біріне адамдардың қауіпсіздігін сақтай отыра эвакуациялау қадамын орындау қажет екендігін меңгере түстім.

Қорытынды

Мобильді қосымша - бұл белгілі бір платформада орнатылған және белгілі бір функционалдылыққа ие планшеттер мен смартфондарға (iOS, Android, Windows Phone және т.б.) арналған әзірленген бағдарлама. Қарапайым сөзбен айтқанда, ол белгілі бір әрекеттерді орындайды және берілген мәселелерді шешуге арналған құрал

Алдын мобильді қосымшалар жай ғана ойын болды. Бірақ кейіннен кәсіпкерлер қосымшаның бизнесті жүргізуге көмектесетінін түсінді. Сонымен қатар кәсіптің немесе тауардың адамнан адамаға хабарлап, сенімін арттыруға, жарнамалық науқандар жүргізуге және клиенттердің пікірлерін білуге мүмкіндік беретін мықты маркетингтік құралға айналдыруға болатындығын аңғара түсті.

Не десек те мобильділік бүгінгі таңда – біздің өміріміздің барлық салаларына кіретін ғаламдық құбылыс болып тұр. Ұялы қызметтерді қолданушылардың шамасымен жартысы бес елде өмір сүреді, олардың арасында Қазақстан да бар. Қазірде қолданушы жай ғана ұялы телефонды іздемейді, ол қазіргі нарықтың талаптарына сай болатын телефонды қарастырады. Міне сондықтан да мобильді технологияларының дамуы тек жылдамдатылатын болады.

Сонымен қатар, дипломдық жобаға келесідей мақсаттар қойылды және қойылған мақсаттар орындалды:

- мейрамхананың мобильді қосымшалар мүмкіндітеріне қадам басуы;
- кәсіпорын иесі үшін келетін шығын көлемінің төмендеуі;
- мейрамханаға келген клиент үшін ақпараттың қол жетімділігі;
- қызмет көрсету бөлімінің даяршылары үшін жұмыс күшінің жеңілдеуі;
- сатылым деңгейінің жоғарылауы;
- қызметкерлер және клиент үшін мобильділік деңгейінің артуы;
- мейрамхана имиджіне жоғарғы баға;

Осындай кең ауқымды мүмкіндікерді қамти алған бұл бағдарлама мейрамхана үшін біраз жылдар жұмыс жасайтыны анық.

Жүргізілген зерттеулер нәтижесінде мобильді қосымшалардың сан тараптаған салаларда өз қызметін жоғарғы деңгейде атқарып келе жатқанын түсіндім. Көп жағдайда алдыңғы қатарлы кәсіп салаларында мобильді қосымшаның күрделенген, бірақ, қолданушы үшін тиімді әрі ыңғайлы нұсқаларының белең алғандығы да көзіме түсті. Мобильді қосымшалардың алдағы уақытта әрбір кәсіп пен салада биік шыңдарды бағындыратына сөзсіз.

«Аспандияр Экспрес» мейрамханасы басшылығына ұсына келе жасалған бұл бағдарлама техникалық тексеруден өтті. Нәтижесінде шығындардың азайып, жұмыс сапасының көтерілуіне үлес қосатындығы нақтылана түсті. Қазіргі таңда мобильды қосымша бағдарламасы үшін қажетті техникалық жабдықтар және толық қанды енгізілу мәселелері өз шешімін тауып келеді.

Мобильді қосымшаны бағдарламалау барысында Visual Code Studio бағдарламалау тілінде негізгі алгоритімі құрылды. Сонымен қатар, JavaScript бағдарламалау тілінде жазылған React Native тілі де мобильді қосымшаның айтарықтай бөлімін бағдарламалады. Дипломдық жобаны орындау барысында әзірленген қосымша API деңгейіндегі Android операциялық жүйесінің басқаруында 26 төмен емес жұмыс істейді.

Алдын-ала қойылған мақсат-міндеттерді орындау барысында жетекшімнен ақыл-кеңес сұрай отырып, көптеген ізденістер жасадым. Тақырып бойынша ақпараттарды әр сала әдебиеттерін қамтитын кітаптардан және интернет желісінен алынды.

Әдебиеттер тізімі

- 1 Магда, Ю. С. NI Measurement Studio. Практика разработки систем измерения и управления на C#, - М.: ДМК Пресс, 2017. - 567 с
- 2 Вендров А.М. CASE–технологии. Современные методы и средства проектирования информационных систем, - М.: Финансы и статистика, 2000.
- 3 Фаулер, Мартин UML. Основы. Краткое руководство по стандартному языку объектного моделирования, - М.: Символ-плюс, 2016. - 192 с.
- 4 Буч Г, Д. Рамбо, А. Джекобсон. UML. Руководство пользователя, - М.: ДМК Пресс; Издание 2-е, стер., 2014. - 432 с.
- 5 8 лучших SDK дополненной реальности для iOS и Android в 2017 году, - [электрондық ресурс]: holographica.space/articles/8-best-ar-sdk-2017-9287
- 6 Дейтел Пол, Дейтел Харви. Android для разработчиков, - Питер, 2016.- 512 с.
- 7 Дарвин Ян Ф. Android. Сборник рецептов. Задачи и решения для разработчиков приложений, - Вильямс, 2017 – 768с
- 8 Голощапов Алексей Леонидович. Google Android. Системные компоненты и сетевые коммуникации, - БХВ-Петербург, 2012. -384с.
- 9 Михайлов Владимир Викторович. Android без страха для начинающих, - Эксмо, 2015.-256с.
- 10 Пахомов Борис Исакович. C/C++ и MS Visual C++ 2012 для начинающих - БХВ, 2012.-236с.
- 11 Филлипс Б., Стюарт К., Марсикано К. Android. Программирование для профессионалов, - Питер, 2017.-668с.
- 12 Павел Агуров. C#. Разработка компонентов в MS Visual Studio 2005/2008, - БХВ-Петербург, 2008.-480с.
- 13 Bonnie Eisenman. Learning React Native, 2nd edition, - 2018 [электронды нұсқа]
- 14 Кеңесбек Д., Сағатова Р.А. Компьютердің пайдасы мен зияны: ғылыми мақала // Ғылым және білім ғасыры = Век науки и образования. – №1/3. - 2014.
- 15 Дюсебаев М. К., Жандаулетова Ф. Р. Еңбекті қорғау және өміртіршілік қауіпсіздігі негіздері. Дәрістер жинағы. Алматы: - АЭЖБУ, 2008.- 35 б.
- 16 Хакимжанов Т.Е., Абдимуратов Ж.С., Мананбаева С.Е. Тіршілік қауіпсіздігі. Барлық мамандықтардың бакалаврлары үшін бітіруші жұмыстағы «Өндірістік жарықтануды есептеу» бөліміне арналған әдістемелік нұсқаулар - Алматы: АЭЖБИ, 2009.
- 17 http://ru.wikipedia.org/wiki/Java_Development_Kit (дата обращения: 01.05.2017).
- 18 John Wiley & Sons. Reto Meier Professional Android 4 Application Development. Wrox, 2012.

- 19 Martin Fowler. UML Distilled: A Brief Guide to the Standard Object Modeling Language (Object Technology Series), - Addison Wesley, 2003.
- 20 Пайлон Д. UML 2 для программистов, - Издательство Питер, 2012.
- 21 Bill Phillips, Brian Hardy. Android Programming: The Big Nerd Ranch Big Nerd Ranch Guides, - 2013.