

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ
МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Кафедра «Программалық инженерия»

Мейрамбайұлы Н.

Бағдарламалау тілдерін үйренуге арналған бағдарлама

ТҮСІНДІРМЕ ЖАЗБА

дипломдық жобаға

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

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ
МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Кафедра «Программалық инженерия»

ҚОРҒАУҒА ЖІБЕРІЛДІ

ПИ кафедра меңгерушісі

техника ғылымдарының
кандидаты, доцент,
ассистент-профессор

 Р. Юнусов
"14"  2019ж.

ТҮСІНДІРМЕ ЖАЗБА

дипломдық жобаға

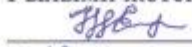

Тақырыбы: “Бағдарламалау тілдерін үйренуге арналған бағдарлама”

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

Орындаған

Мейрамбайұлы Н.

Ғылыми жетекші, лектор

 I.N. Ersari
"12"  2019 ж.

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ


Ақпараттық және телекоммуникациялық технологиялар институты

Кафедра «Программалық инженерия»

5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»

БЕКІТЕМІН

ПИ кафедра меңгерушісі,
техника ғылымдарының
кандидаты, доцент,
ассистент-профессор

 Р. Юнусов
" 14 " мае 2019ж.

**Дипломдық жобаны орындауға
ТАПСЫРМА**

Білім алушыға Мейрамбайұлы Н.

Тақырыбы: Бағдарламалау тілдерін үйренуге арналған бағдарлама

Университет ректоры бұйрығының № 1162-б "16" қазан 2018 ж. шешімімен бекітілген.

Орындалған жобаның өткізу мерзімі " 14 " мамыр 2019 ж.

Дипломдық жобаның бастапқы мәліметтері: Жобаның төлқұжаты, технология бойынша техникалық құжаттама, техникалық тапсырма.

Есеп – түсініктеме жазбаның талқылауға берілген сұрақтардың тізімі:

a) Кроссплатформды тілдерді қарастыру;

b) пайдаланушы интерфейсін жобалау және дамыту;

в) бағдарлама оқу тілдерін дамыту, жақсарту;

г) бағдарламаны құру, қателіктерді анықтау және түзету, тестілеу;

Графикалық материалдар тізімі (міндетті суреттердің нақты көрсетілуімен):


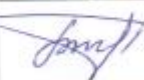
15 графикалық материалдар бар.

Ұсынылған негізгі әдебиеттер: 13 әдебиеттер тізімінен


Дипломдық жобаны орындау
КЕСТЕСІ

Бөлімдердің атаулары, зерттелген мәселелердің тізімі	Ғылыми жетекшіге және кеңесшілерге ұсыну мерзімі	Ескерту
1. Кіріспе	10.01.2019	2008
2. Тапсырма қойылымы және бағдарламалау ортасын таңдау	18.01.2019	2008
3. Бағдарламалық қамтаманы жобалау.	01.02.2019	2008
4. Бағдарламаны әзірлеу	14.02.2019	2008
5. Бағдарламаның мүмкіндіктерін дамыту	12.03.2019	2008
6. Дипломдық жобаға түсіндірме жазба жазу	30.03.2019	2008

Дипломдық жұмыс бөлімдерінің кеңесшілерінің аяқталған жұмысқа қойған
қолтаңбалары

Бөлімдер атауы	Кеңес берушілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)	Қолтаңба қойылған мерзімі	Қолы
Нормалық бақылаушы	А.Б. Рамазан Тьютор	13.05.19	
Бағдарламалық бөлім	А.М. Байғаринов Техн.ғыл.магистр, лектор	13.05.19	

Ғылыми жетекші _____  I.N. Ersari

Тапсырманы орындауға қабылдап алған студент  Н. Мейрамбайұлы

Күні _____ «1» маусым 2018 ж.

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ
МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Кафедра «Программалық инженерия»

Мейрамбайұлы Н.

Бағдарламалау тілдерін үйренуге арналған бағдарлама

ТҮСІНДІРМЕ ЖАЗБА

дипломдық жобаға

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

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ
МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Кафедра «Программалық инженерия»

ҚОРҒАУҒА ЖІБЕРІЛДІ

ПИ кафедра меңгерушісі

техника ғылымдарының
кандидаты, доцент,
ассистент-профессор

_____ Р. Юнусов

" ____ " _____ 2019ж.

ТҮСІНДІРМЕ ЖАЗБА

дипломдық жобаға

Тақырыбы: “ Бағдарламалау тілдерін үйренуге арналған бағдарлама”

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

Орындаған

Мейрамбайұлы Н.

Ғылыми жетекші, лектор

_____ І. Ерсари

" ____ " _____ 2019 ж.

Алматы 2019

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Ақпараттық және телекоммуникациялық технологиялар институты

Кафедра «Программалық инженерия»

5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»

БЕКІТЕМІН

ПИ кафедра меңгерушісі,
техника ғылымдарының
кандидаты, доцент,
ассистент-профессор

_____ Р. Юнусов
" ____ " _____ 2019ж.

**Дипломдық жобаны орындауға
ТАПСЫРМА**

Білім алушыға Мейрамбайұлы Н.

Тақырыбы: Бағдарламалау тілдерін үйренуге арналған бағдарлама

Университет ректоры бұйрығының № 1162-б "16" қазан 2018 ж. шешімімен бекітілген.

Орындалған жобаның өткізу мерзімі " 14 " мамыр 2019 ж.

Дипломдық жобаның бастапқы мәліметтері: Жобаның төлқұжаты, технология бойынша техникалық құжаттама, техникалық тапсырма.

Есеп – түсініктеме жазбаның талқылауға берілген сұрақтардың тізімі:

a) Кроссплатформды тілдерді қарастыру;

b) пайдаланушы интерфейсін жобалау және дамыту;

в) бағдарлама оқу тілдерін дамыту, жақсарту;

г) бағдарламаны құру, қателіктерді анықтау және түзету, тестілеу;

Графикалық материалдар тізімі (міндетті суреттердің нақты көрсетілуімен):

15 графикалық материалдар бар.

Ұсынылған негізгі әдебиеттер: 13 әдебиеттер тізімінен

Дипломдық жобаны орындау
КЕСТЕСІ

Бөлімдердің атаулары, зерттелген мәселелердің тізімі	Ғылыми жетекшіге және кеңесшілерге ұсыну мерзімі	Ескерту
1. Кіріспе	10.01.2019	
2. Тапсырма қойылымы және бағдарламалау ортасын таңдау	18.01.2019	
3. Бағдарламалық қамтаманы жобалау.	01.02.2019	
4. Бағдарламаны әзірлеу	14.02.2019	
5. Бағдарламаның мүмкіндіктерін дамыту	12.03.2019	
6. Дипломдық жобаға түсіндірме жазба жазу	30.03.2019	

Дипломдық жұмыс бөлімдерінің кеңесшілерінің аяқталған жұмысқа қойған
қолтаңбалары

Бөлімдер атауы	Кеңес берушілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)	Қолтаңба қойылған мерзімі	Қолы
Нормалық бақылаушы	Рамазан А. Тьютор		
Бағдарламалық бөлім	Байгаринов А.М. Техн.ғыл.магистр, лектор		

Ғылыми жетекші _____ Ерсари І.Н.

Тапсырманы орындауға қабылдап алған студент _____ Мейрамбайұлы Н.

Күні _____ «__» _____ 2018 ж.

КІРІСПЕ

Осы дипломдық жұмыстың жасау шеңберінде бағдарламалау тілдерін үйренуге мүмкіндік беретін мобильді қосымша жасалынды. Бұл бағдарламаның негізгі функциясы тестілеуден өтіп, қысқаша және түсінікті материалдарды зерттеу арқылы бағдарламалау тілдерін үйрену болып табылады.

Компьютерлер мен байланыс құралдарының қолжетімді болуы білім беру жүйесіне де өзгерістер алып келуде. Қазірде қандай да бір білім алу үшін бұрынғыдай ел аралап, жер жүрудің қажеті жоқ. Тіпті, мектеп, университет секілді білім беру орнына бармай-ақ, үйде отырып ғаламтор желісінен білім үйренуге болады. Сонымен қатар білім беру жүйесінің қазіргі үрдісінде студенттің білімді табысты меңгеруі үшін оқу материалдарының басым бөлігін өзі жеке үйренуін талап етеді.

Сәйкестік: Қазіргі заманда статистика бойынша еңбек нарығында ең көп талап етілетін мамандық ІТ инженерлер болып табылады. Соған байланысты ІТ инженерлерді онлайн форматта дайындайтын бағдарлама жасалынды.

Жобаның мақсаты – ІТ-технология саласында білім алатын немесе жұмыс істейтін, және де осы салаға қызығушылығы бар кез-келген азаматтарға арналған оқу платформасын құру.

Мақсаты: Бағдарлама көптеген қолданушылар үшін қол жетімді болуы үшін, бағдарламаны IOS және Android платформаларында құру.

Жұмыс құрылымы: бұл тезис кіріспеден, негізгі бөліктен, қорытындыдан, қолданылған әдебиеттер тізімінен және қосымшалардан тұрады.

1.1 Мобильды қосымшаларды кроссплатформада жазу

Мобильді қосымшалар нарығы қазірдің өзінде он жылдан асады, бірақ ол әлі де дамып келеді. Компаниялардан мобильді қосымшаларды құру бойынша сұраныс үнемі өсіп келеді және ол әлі де дамудың өзіндік құнының тұрақты өсуіне әкеліп соқтыратын жеткізілімнен айтарлықтай асып түседі. Осы процестің өзіндік құнын төмендету шешімдерінің бірі - барлық платформаларда сол код қолданылған кезде кросс-платформаны дамыту.

Бағдарламалық жасақтаманы әзірлеушілер нақты платформаға арналған бағдарламалау тілін қолданатын болса, онда ол Objective-C немесе Android үшін iOS немесе Java немесе Kotlin үшін Swift болсын, мұндай бағдарлама туған тіл (ағылшын тілінен - туған, табиғи) деп аталады.

Нативты қосымшалардың артықшылықтары: интерфейстің әрекет ету жылдамдығы мен реакциясы. Қолданба бірден басылуына жауап береді, анимацияда, айналдыруда, деректерді қабылдауда және шығаруда ешқандай кешіктіру жоқ; Құрылғы мүмкіндіктері мен сенсорларына оңай және оңай қол жеткізу. Әзірлеуші үшін геолокациямен жұмыс істеуде проблемалар жоқ, хабарландырулар жіберіледі, камералар, дыбыс, акселерометр және басқа да сенсорлар арқылы фото және бейне түсіреді; Смартфонның функцияларымен терең жұмыс істеу мүмкіндігі. Алдыңғы абзацдағыдай, анимация сияқты нәрселер, күрделі интерфейстерді құру және нейрондық желілердің тікелей құрылғыларда жұмыс істеуі жай, бірақ жай емес, болжанған; Нативты платформалық интерфейс. Түпнұсқалық қосымшалар әдетте «платформа» интерфейс элементтерімен жұмыс істейді: мәзірлер, навигация, пішіндер және барлық басқа дизайн элементтері операциялық жүйеден алынып, пайдаланушыға түсінікті және түсінікті.

Біреудің жетіспеушілігі – даму мен қолдаудың жоғары құны. Әрбір платформаға кодты жазу қажет. Ұялы қосымшалар нарығының өсуімен, әзірлеушілер қымбат емес, бірақ өте қымбат болды. Кросс-платформалық қосымшалар бірнеше платформа үшін бір тілде, бір-бірінен өзгеше, жазылады. Мұндай код әртүрлі құрылғыларда қалай жұмыс істей алады? Мұнда да екі тәсіл бар.

Біріншісі – жариялауға өтінім дайындау кезеңінде транспилерді қолданатын нақты платформаға ана болу. Шындығында, бір кросс-платформалау тілінің басқа тіліне аударылған.

Екіншіден, құрылғыда жұмыс істеп жатқан кодқа туындының кодынан жүйенің жергілікті функциясына қоңырауды жібереді.

Бұл кодтың көп бөлігі платформалар арасында берілуі мүмкін деп болжануда, мысалы, сатып алуды жүзеге асыру логикасы, арбаға тауарларды сақтау, таксиге арналған маршрутты есептеу, хабаршыға хабарлама жазу клиенттің Android немесе iOS нұсқасына байланысты өзгермейді. Жай ғана платформалар үшін UI және UX-ді өзгерту қажет, бірақ қазір, белгілі бір шектерде, тіпті оны біріктіруге болады, мысалы, гамбургер мәзірі Android және iOS-де белсенді

қолданылады. Сондықтан интерфейске түзетулер енгізе отырып, өтінімнің қажетті платформаға хат пен көңіл-күйге жауап беруі – бұл ниет, қажетті жылдамдық және дамудың сапасы.

Артықшылықтары: Құны мен даму жылдамдығы. Кодты айтарлықтай аз жазуға тура келетіндіктен, жұмыс құны төмендейді; Компанияның ішкі ресурстарын пайдалану мүмкіндігі. Кейінірек көрсететініміздей, ұялы қосымшалардың кросс-платформалық дамуын жиі қолданыстағы бағдарламашылар жасай алады.

Кемшіліктері: Туынды емес интерфейс немесе, кем дегенде, әрбір платформаның интерфейсiмен жұмыс iстеу қажеттілігі. Әрбір жүйе элементтерді жобалауға өз қажеттіліктеріне ие, ал кейде олар бір-бірінен айрықша болып табылады. Бұл мәселені қарастыру қажет; Күрделі функцияларды іске асырудағы проблемалар немесе жұмыстың ықтимал проблемалары, тіпті даму шеңберінің қателіктеріне байланысты қарапайым рәсімдермен бірге. Кросс-платформа ортасы тек қана жүйелік қоңырауларға және жүйелердегі интерфейстерге, оның түсінген пішіміне сұрауларды аударды, сондықтан осы кезеңде түсінудегі қиындықтар да, шеңбердегі қателердің пайда болуы да мүмкін; Жұмыс жылдамдығы. Кросс-платформа ортасы кодтың үстінен (әрдайым емес, бірақ белгілі бір жағдайларда) «үстiнгi құрылым» болғандықтан, пайдаланушы әрекеттерін және нәтижелерді көрсету кезінде өздерінің кешіктірулерін және үзілістерін тудырады. Бұл әсіресе бірнеше жыл бұрын смартфондарда байқалды, олар бүгінгі күндерге қарағанда әлдеқайда төмен, бірақ қазір мобильді құрылғылардың өнімділігін жоғарылату арқылы бұл ескермеуі мүмкін. Көріп отырғаныңыздай, бұл екі әдіс - іс жүзінде бір-бірінің айна бейнесі – жергілікті дамудың артықшылықтары, кросс-платформаның кемшіліктері және керісінше.

1.2 Танымал платформалар және кросс-платформаны әзірлеу құралдары:

Жоғарыда жазғанымыздай, екі тәсіл бар: кодты құрастыру сатысына нативты түрге айналдыру немесе жүйеге қоңырауларды аударатын нақты ораманы қосу. Дегенмен, кросс-платформаны дамытуға қарағанда, түпнұсқалар өндірістегі сапалы қосымшалармен қамтамасыз етілсе де, жана өнімдерді жасауға мүмкіндік беретін жаңа кросс-платформа технологиялары пайда болды.

Туған дамудың негізгі артықшылықтары - пайдаланудың жоғары көрсеткіштері және операциялық жүйелердің мүмкіндіктеріне қол жеткізу.

Дегенмен, олардың жұмысында бағдарламалық қамтамасыз ету инженерлері көбінесе барлық деңгейлерде, соның ішінде бағдарламалау тілдері, архитектура, кітапханалармен жұмыс істеу және т.б.

Сондықтан, сол алгоритмдерді және пайдаланушы және / немесе бизнес сценарийлерін іске асыру үшін бағдарламашылар әртүрлі тілдерде бірнеше орталарға арналған бағдарламаны жасауы керек.

Осылайша, компания өзінің жұмыс уақытын, бюджетін және өнімді сақтауға күш-жігерін арттыруға тиіс. Сондай-ақ, отандық даму бағдарламашыдан белгілі бір біліктілікті талап етеді. Әйтпесе, мамандарды даярлауға қосымша уақыт бөлуге тура келеді.

Осы мәселелерді шешу үшін кросс-платформалық мобильді дамыту технологиялары пайда болды. Кейбір кемшіліктерге қарамастан, олар ұялы қосымшаларды жазу процесін жеңілдетеді, компаниялардың еңбек шығындарын азайтады және клиенттердің шығындарын азайтады.

Қазіргі кезде жоғары сапалы қосымшаларды жасауға мүмкіндік беретін жақсы кросс-платформа технологиялары бар. Бұл мақалада кросс-платформаны дамытуға арналған ең танымал құрылымдар қарастырылады.

1.2.1 Xamarin

Маркетингте жетекші орынды иемденіп, әзірлеушілерге түрлі платформаларға (iOS, Android және Windows Phone) арналған қосымшаларды жасауға мүмкіндік беретін тағы бір керемет кросс-платформалық шешім.

Xamarin бірыңғай ортақ C # кодының негізін қамтиды және Xamarin Cloud көмегімен көптеген құрылғылардағы қосымшаларды сынауға мүмкіндік береді.

Xamarin 2-нің келуімен бұл шеңбер ең танымал кросс-платформалық құралдардың бірі болды. Айта кету керек, бұл құрылымды Microsoft және IBM сияқты ірі корпорациялардан тұратын күшті серіктестік қауымдастығы бар. PhoneGap-ге қарағанда, онымен жұмыс істеу ақылы жазылымды талап етеді, бірақ сіз сынақ нұсқасының тегін нұсқасынан бастай аласыз.

Xamarin өзінің барлық Xamarin Studio IDE және Xamarin сияқты көптеген пайдалы мүмкіндіктерді ұсынады, бұл бағдарламашыларға барлық платформаларға арналған жазбаша кодтың 100% -ын қолдануға мүмкіндік береді. Негіздеме сондай-ақ жергілікті API-ке қол жеткізу мүмкіндігін ұсынады және Parse және Microsoft Azure сияқты осындай қосылымдарды біріктіруге мүмкіндік береді.

Xamarin-дегі кросс-платформаны дамыту үшін iOS, Android және C # -тегі тәжірибе қажет. Сонымен қатар, сіздің қосымшаңыз толығымен ана тілінде (C # -бен жазылған болса да) және базалық кодтардың мөлшері 40% -дан аспайды.

1.2.2 Flutter

Әрине, мен Android және iOS-қосымшалардың кросс-платформалық дамуы мен Google сияқты алыптар тақырыбын айналып өте алмадым. Дегенмен, осы уақытқа дейін тек фритке, бета-бетада ғана бар, реактивтік және xamarin реакциясынан басқа көзқарасқа ие. Ол бастапқы кодты платформамен

жасалынған түпнұсқаға айналдырмайды, бірақ іс жүзінде смартфондар экранында терезе тартады және барлық элементтерді өзі тартады. Тіл JavaScript-нің жетілдірілген нұсқасы ретінде жасаған «патенттелген» Дартты пайдаланған.

Бұл екі артықшылығы бар (мысалы, сыртқы ұқсас интерфейстер) және кемшіліктер (мысалы, интерфейсті қайта өңдеу белгілі бір жады мен процессор уақытын талап етеді).

Платформа тез дамып келеді және Google оған үлкен күш пен ақша салады. Бірақ Flutter-пен салыстырғанда, React Native-дің тіпті жақсы және әсерлі экожүйесі де көрінеді.

1.2.3 PhoneGap

Adobe PhoneGap - бұл тегін негізде пайдалануға болатын ашық бастапқы код. Сондай-ақ, PhoneGap мобильді қосымшаларды жасау үшін аппараттық, SDK және компиляторларды қажет етпейді.

Бүгінгі күні PhoneGap бағдарламашылар CSS3, HTML5 және JavaScript шешімдерін жасайтын кросс-платформаның жетекші құралдарының бірі болып табылады. Плагинаның тағы бір плюс - берілген қосылатын модульдердің жиынтығы.

Бұл кросс-платформаны пайдалана отырып, бағдарламашылар ендірілген веб-браузерлер және бір беттік HTML болып табылатын қосымшаларды әзірлейді.

Барлық интерфейс элементтері түпнұсқа ретінде стильденгендіктен, API үшін тікелей қол жетімділік жоқ. Жүйенің функцияларына қол жеткізу үшін, әзірлеушілер JS әдістерін өз веб-шолғышына қосатын плагиндерді біріктіреді және оларды әр операциялық жүйеде жергілікті іске қосу арқылы байланыстырады.

Adobe PhoneGap – 10-16 скриншотпен және шағын аудиториямен, мысалы, компанияның ішкі тапсырмалары үшін корпоративті шешімдердің кейбір түрін қолдану интерфейсін немесе қарапайым бағдарламаны жасау қажет болса, бұл тамаша таңдау. Негіздемелікпен жұмыс жасау командаға JavaScript, CSS және HTML-де бір беттік веб-қосымшаларды жазуды талап етеді.

1.2.4 Ionic

Ionic-да стандартты элементтердің орнатылған кітапханасы бар, олар Bootstrap элементтері сияқты пайдаланылуы мүмкін: карталар, түймелер, қосқыштар, сегменттер, қалқымалы терезелер, енгізу өрістері, тізімдер, сызықтар мен бағандар торы және т.б. Әдепкі бойынша, бұл элементтер iOS және Android-дегі туынды, бірақ қажет болса, олардың сыртқы көрінісін өзгертуге болады.

1.2.5 React Native

React Native негізі Facebook арқылы жасаған React.js сайтынан «өседі». JavaScript-дегі «cubes» -ден интерфейсті құрастырып, жобаға веб-бейімделу тәсілінен қарыз алынды, сондықтан компания мобильді платформаларға реактивті түрде көшті. Міне, сол сияқты реакция ұлты пайда болды.

Қазіргі уақытта React Native - ең қызықты және перспективалық негіз болып табылады, себебі ол веб пен React.js ішінен ең жақсысын алады.

JS қозғалтқышы арқасында, ол жоғары деңгейде өнімділікті қамтамасыз етеді. Интерфейсті «текшелерден» жасау идеясына сүйене отырып, React native бағдарламасы браузерді немесе WebView – тек JavaScript API-ды пайдаланбайды.

Осылайша, бағдарламашылар платформаның құрамдас бөліктерімен жұмыс істейтін JavaScript кодын жазады, осылайша React.js веб-торабының артықшылықтары мен ыңғайлылығын браузерден мобильдік қосымшаларға ауыстырады.

Реактивті түпнұсқалықтың маңызды артықшылығы мынада, ортақ кодтың пайыздық көрсеткіші 90% дейін, бұл туғанға ұқсас заманауи қосымшаларды жазуға көмектеседі. Сонымен қатар, даму қарапайым және ыңғайлы.

Бағдарламашылар шеңберінен жұмыс істеу JavaScript, Objective-C API және Android Java API білуді талап етеді. Көріп отырғаныңыздай, React Native сізге жоғары сапалы, жоғары сапалы шешімдерді жасауға мүмкіндік береді.

2 Гибридтік және нативті қосымшалар

Ionic және React Native салыстырғанда сүңгуден алдында, бізге қысқаша осындай гибридтік және туған қосымшалар деп өзімізді естеріңізге саламыз. Gibrirdnye қолдану деп аталатын WebView арқылы, шын мәнінде мобильді өтінімде кірістірілген веб-сайттар болып табылады. Олар қарамастан, олар жұмыс істейтін туралы платформасының, HTML, CSS, және JavaScript пайдалану арқылы жасалған, және сол кодты орындауға беріледі. Сондай-ақ, сияқты аясында арқылы: PhoneGap және Crodova, мұндай камера немесе GPS ретінде ана құрылғы мүмкіндіктерін пайдаланыңыз.

Жергілікті қосымшалар мақсатты платформаға қажетті тілде жасалады. IOS үшін Objective-C немесе Swift, Android үшін Java және т.б. жазба коды платформа емес және әр түрлі платформалар үшін әр түрлі болады. Олардың айырмашылығы 2.1-суретте көрсетілген. Бұл қолданбалар ешқандай шектеусіз тиісті платформаны ұсынатын құрылғының барлық функцияларына тікелей қол жеткізе алады.



2.1-сурет – Нативты және гибридті бағдарламалар

Ionic - әдеттегі гибридтік құрылым. Ол веб-технологияларды қолданбаны жазу және көрсету үшін пайдаланады және телефонның Gap / Cordova көпірін қолданбаның ішкі функцияларына кіруін талап етеді. Содан кейін ол осы ОЖ пайдаланушысымен жақсы қарым-қатынас жасау үшін платформаның өзіндік мінез-құлқын қайталауға тырысады.

React Native - ерекшелігі кодтың көп бөлігі кросс-платформа ретінде пайдаланылуы мүмкін. Дегенмен, гибриді қолданбалар HTML және CSS-ні қолданып интерфейсті көрсетсе, RN көрсету үшін жергілікті компоненттерді пайдаланады. Бұл пайдаланушы интерфейсі платформаның өзі анықтаған белгілі бір үлгілерді пайдаланатын басқа платформа бағдарламаларына жақын болады дегенді білдіреді. Жиі ол әлі де жақсы нәтиже береді және тегіс анимация береді. Мәселе мынада, ол тек қажетті компонент RN үшін жазылған кезде ғана жұмыс істейді. Басынан бастап, RN компоненттердің лайықты жиынтығын ұсынады, бірақ егер сіз жергілікті бағдарламаларға бағдарламашылар болсаңыз, сізге қажет барлық компоненттерді таба алмаңыз. RN-да жазылған бағдарлама ана, туған болады, бірақ ол қолданылатын әр платформа үшін жазылған құрамдас бөліктерді қажет етеді. Гибриді қосымшаларға жақындатады.

2.1 React Native және Ionic айырмашылықтары

A) Ionic

Типтік гибриді бағдарламада, платформаға қарамастан, бірдей код жұмыс істейтін болады, және бұл Ионич қабілетті. Дегенмен, «туған» сезіну үшін (ана), Ионич өзінің кейбір функцияларын платформаға бейімдейді. Қойындыларды пайдалансаңыз, олар тек платформа ұсынатын жолмен көрсетіледі - iOS экранының төменгі жағында және Android үшін үстіңгі жағында.

B) React Native

RN мақсаты - «бір рет жазу, барлық жерде жұмыс істеу» қабілеті емес. RN әзірлеушілерге қолданылатын платформаға сәйкес келетін компоненттерді пайдалануға мүмкіндік береді. Мысалы, Android-те теңшелетін құралдар тақтасы бар; iOS емес. Android үшін құралдар тақтасын пайдалануға болады, бірақ сіз iOS үшін тағы бір нәрсені пайдалануыңыз керек. Алаңдама, көптеген құрамдастардың эквиваленттері бар; ал егер олар эквивалентті болмаса, логиканың бөлігін бөлек платформаға бөлу оңай. Ionic-пен платформаға тән мінез-құлық туралы алаңдауға болмайды. Бірақ RN-мен белгілі бір платформаны пайдаланушылар үшін таныс интерфейсін ұсына аласыз.

2.2 Тілдік стек

Ionic – бұл фреймворк Angular фреймворкының негізінде жасалынған. Көрініс үшін HTML үлгілерін пайдаланады. MVC үлгісіне сәйкес, көзқарас пен логика мүмкіндігінше бөлінеді.

React – бұл фреймворк React фреймворкының негізінде жасалынған және JavaScript-ды қолданатын HTML үлгілерін пайдаланады, бірақ бұл үлгілер JS-ге кіріктірілген және ол HTML-ден көп JS. Оған тіпті JSX анықтамасы берілген. Сондықтан, бағдарлама логикасы дисплеймен тығыз байланысты. Бірақ, ақыр соңында, бұл проблема емес, ал көптеген әзірлеушілер JSX-мен өте ыңғайлы болады. Бірақ дизайнерлермен бірге жұмыс істесеңіз, олармен жұмыс істеу HTML + CSS-мен жұмыс істеу сияқты оңай болмайды. Бұл JSX зерттеуі көп уақытты қажет ететінін білдірмейді, бірақ JSX HTML емес екенін түсінуіміз керек. Оның синтаксисі қалыпты HTML-дан сәл өзгеше және CSS ережелерінің жиыны бірнеше шектелген.

Ionic-пен сіз классикалық HTML және CSS-ны жеңілдететін боласыз. React Native - пен сіз өзіңіздің HTML интерфейсіңізді ұқсас құрамдастармен қалай стильдеуге және жасауға болатындығын үйренуіңіз керек. Бұл қиын емес, бірақ бұл жаңа нәрсе.

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 }
renderTodo(todo) {
  return (
    <View style={styles.todo} key={todo.id}>
      <View>
        <Switch
          onChange={() => this.toggleTodo(todo)}
          value={todo.done} />
      </View>
      <View>
        <Text>{todo.text}</Text>
      </View>
    </View>
  )
}
```

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
<ion-list>
  <ion-item *ngFor="let todo of filteredTodos()">
    <ion-toggle
      [checked]="todo.done"
      (ionChange)="toggleTodo(todo)"
    </ion-toggle>
    <ion-label>
      {{todo.text}}
    </ion-label>
  </ion-item>
</ion-list>
```

2.1-сурет – Ionic және RN айырмашылығы

2.3 Тестілеу

A) Ionic

Өтінім жасалса, өз жұмысының нәтижесін дереу алуға өте маңызды. Бағдарлама қайта өңделіп, қайсысы 2 пикселден немесе 3 сол жақ шегінен жақсы екенін көру үшін күткенде өте нашар. Ионикпен браузерде және

мобильді құрылғыда сіздің өтініміңіздің алдын ала қарауын дереу көре аласыз. Қолданбаңыздағы нәрсені өзгерткеннен кейін ол тікелей жаңартылады.

B) React Native

Сіздің браузеріңізде тестілеуді ұмытыңыз, RN өз рендерингін жасайды. Бұл проблема емес, сіз өзіңіз жасағаннан кейін өзгерістердің нәтижесін көріңіз. Сізге recompilation, persian және т.б. қажет емес. Нәтиже эмуляторда немесе нақты құрылғыда бірден көрсетіледі.

2.4 Плагиндер және қауымдастық

Мен Open Source жобасын таңдаған кезде, оны айналасындағы жарқын қоғамдастықтың болуы маңызды деп таптым. Бұл интернетте ақпаратты таба білуіңізге, сұрақтарға жауап беруге және қателерді түзетуге қалайша оңай әсер етеді.

A) Ionic

PhoneGap және Cordova әлдеқайда ертерек бар және сізге қажет плагиннің бар екендігі және WebView бағдарламасындағы жергілікті құрамдас бөлікті қолдану мүмкіндігі бар.

B) React Native

Көптеген плагиндер React Native үшін NPM-де бар. PhoneGap қосылатын модульдерін қолданатын қосылатын модульдер бар, яғни басқа қауымдастықтардан жүздеген қосылатын модульді пайдалану мүмкіндігін дереу иеленуіңізді білдіреді.

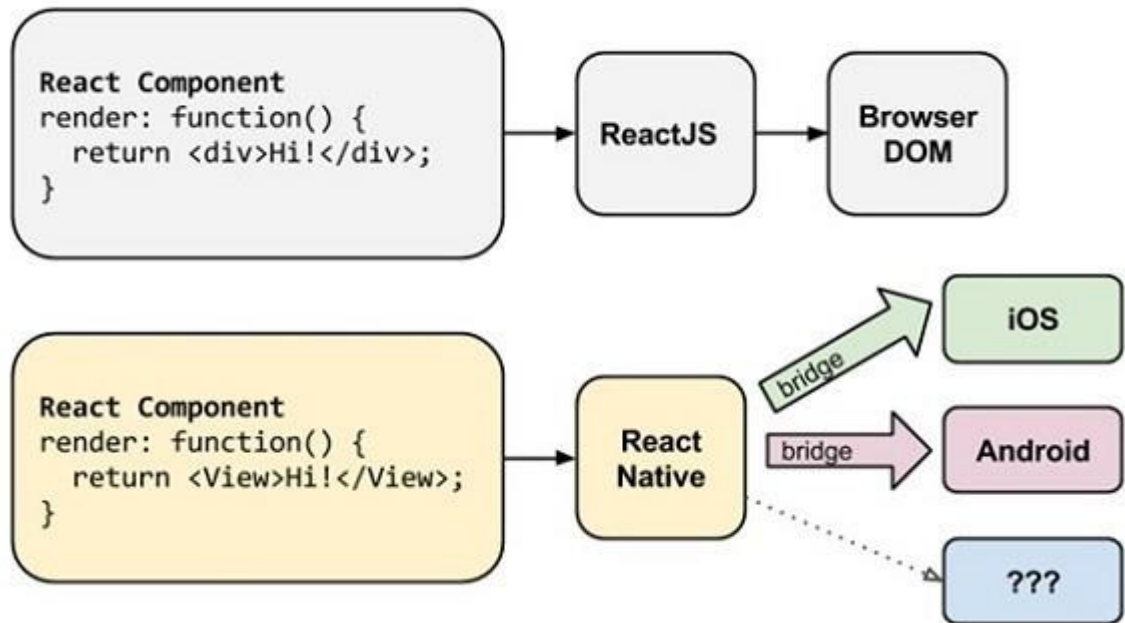
Бүгінгі күні, React Native GitHub-тағы Ионичтен, яғни, шамамен 36,000-ға 25,000-ға қарағанда көбірек жұлдыздар бар. Дегенмен, Ionic React native тілінен гөрі StackOverFlow бойынша сұрақтар көп. Екі технология да Angular және React негізделген қауымдастықтарға ие болады. Сондықтан біз екеуінің де өте үлкен және дамып келе жатқан қоғамдастығына ие екендігіміз туралы тыныш айта аламыз.

2.5 Гибридтік және нативті бағдарламалардың айырмашылығының қорытындысы

Қай технологияның жақсы екендігін айта алмаймын. Сізге, сіздің жобаңызға, пайдаланушы талаптарына, сіздің командаңыздың тәжірибесіне және т.б. көптеген факторлармен анықталады деп сенемін. Олар екеуі де әртүрлі нәрселерді жасайды, екеуі де жақсы. Бұл Xbox және PlayStation, MacOS және Windows, автомобиль және мотоциклді салыстыру сияқты. Екіншіден де жақсы емес. Дегенмен, қандай технологияны таңдағанымды айтып берсем болады. Бұл React Native. Мен кіріктірілген кескінді ештеңеден артық бағалаймын. Мен үшін бұл бағдарлама өте жылдам және бейімделгіш болуы керек, алайда, жергілікті тілде мен таза табиғи қосымшалар мен таза гибридті қосымшаларды жазу арасында жақсы ымыраға ие болдым. React Native маңызды артықшылығы мынада, ортақ кодтың пайыздық көрсеткіші 90% дейін, бұл туғанға ұқсас заманауи қосымшаларды жазуға көмектеседі. Сонымен қатар, даму қарапайым және ыңғайлы.

3 React Native фреймворкы

React Native – бұл IOS және Android үшін кроссплатформды қосымшаларды әзірлеуге арналған фреймворк. React Native 2015 жылдың басында Фейсбук командасымен шығарылған және пайдаланушы интерфейстерін жасауға арналған реактивті JS-кітапханасына негізделген. Бірақ браузерлердің орнына ол мобильді платформаларға бағытталған. Басқаша айтқанда, сіз веб-әзірлеушісіз болсаңыз, Таза және жылдам мобильді қосымшаларды жазу үшін React Native фреймворкын қолдануға болады. Бұл фреймворк React базасында жасалынған. React - пайдаланушы интерфейстерін жасау үшін JS кітапханасы, әдетте веб-қосымшалар үшін. Ол Facebook-де жасалған және 2013 жылдан бастап ашық бастапқы лицензиясы бойынша таратылады. React кеңірек таратылады, және үлкен MVC шеңберлерінен айырмашылығы, бұл тар мәселені шешеді: интерфейссті көрсету. React-тің танымалдылығы бірнеше себептерге байланысты. Бұл шағын және жоғары өнімділікке ие, әсіресе жылдам өзгеретін деректермен жұмыс істеген кезде. Құрамдас құрылымы арқасында, React модульдік, қайта пайдалану кодын жазуға талпындырады. React Native – бұл React, бірақ мобильді платформалар үшін. Олардың бірнеше айырмашылық бар: div тегінің орнына view пайдаланылады және img тегінің орнына Image қолданылады. Даму процесі бұрынғы қалпында қалды. React компонент өзінің жеке картасын сипаттайды, содан кейін кітапхана сіз үшін қызмет етеді. Бұл екі функция абстракцияның айқын деңгейімен бөлінеді. Егер сізге компоненттерді интернетте көрсету керек болса, онда React стандартты HTML тегтерін қолданады.



3.1-сурет – React Native компоненті

Сіз стандартты JavaScript, CSS және HTML сияқты өте қорқынышты кодты жазасыз. Туынды кодты құрастырудың орнына, React native бағдарламасы қолданбаңызды қабылдайды және негізгі UI ағыны бұғаттай-ақ JS механизмінің хост-платформасын пайдаланып іске қосады. Сіз Objective-C немесе Java-де жазылмай-ақ, орындаудың, анимацияның және мінез-құлқың артықшылықтарын аласыз. Кордова немесе Титан сияқты кросс-платформалық қосымшаларды әзірлеудің басқа әдістері ешқашан бұл өнімділік немесе дисплейдің деңгейіне жете алмайды. Стандартты iOS және Android әзірлеуімен салыстырғанда, React Native көп артықшылықтары бар. Өтініш негізінен JavaScript болып табылатындықтан, веб-әзірлеудің көптеген артықшылықтарын пайдалана аласыз. Мысалы, кодқа енгізілген өзгертулерді көру үшін дәстүрлі қайта құрудың аяқталуын күтудің орнына бағдарламаны бірден «жаңартуға» болады. Сонымен қатар, React Native мобильдік дамуды едәуір жеңілдететін интеллектуалды қателер туралы есеп беру жүйесін және стандартты JavaScript түзету құралдарын ұсынады.

3.1 Көп деңгейлі өңдеу

React Native талғампаздығы әртүрлі платформаларды өңдейді. Негізінде API-ның басым көпшілігі кросс-платформаға айналады, сондықтан React Native

компонентін жай жазу жеткілікті, ол iOS және Android-де проблемасыз жұмыс істейді. Facebook әкімшілігінің хабарландыруында, кодының 87% екеуі де платформаларда қайта қолданылатынын көрсетеді.

Платформа үшін кодты жазу қажет болса – iOS және Android-де өзара әрекеттесудің әртүрлі ережелеріне байланысты немесе платформаға тәуелді API артықшылығына байланысты – бұл кезде қиындықтар болмайды. Реактивті түпнұсқа әр компоненттің платформаға тән нұсқаларын тағайындауға мүмкіндік береді, одан кейін сіздің қосымшаңызға кіре аласыз.

Барлығы нативты, сондықтан толық кросс-платформды компоненттер жоқ. Платформалар әртүрлі, сондықтан компоненттер әртүрлі. Олардың әртүрлі логика мен өзара әрекеттесу механикасы бар.

IOS платформасының Vine қосымшасының мысалында. Олар iOS-та не істейді? TabBar төменгі жағында экрандарды ауыстыру әдеттегідей: негізгі, профиль, іздеу. NavigationBar жоғарғы жағында, ал сол жақта оң жақта (әдетте сол жақта және оң жақта - кейбір әрекетте) атау мен батырмаларды жазу дәстүрге айналады. Ал Android-де бұлай емес. Сондай-ақ, NavigationBar бар, бірақ ол әртүрлі, түймешіктерді қабылдамайды. Бұл үшін ToolBar деп аталатын бөлек компонент бар. Android-де SegmentedActivity жасау әдеттегі болып табылады - бұл iOS TabBar-ге өте ұқсас, бірақ оның механикасы толығымен ерекшеленеді. Егер біз TabBar экранын алмастыра алмасақ, оны Android-да жасай аламыз, әдетте ол жасалады, және бұл жұмыс істейді.

Бағдарлама неден тұрады?

- ешқандай таныс div, span, button, input және т.б;
- кәдімгі CSS жоқ;
- DOM жоқ.

Компоненттер

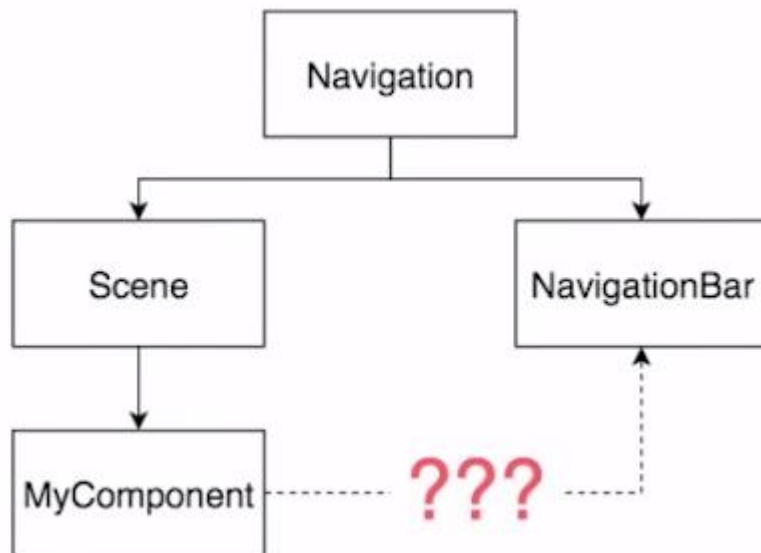
– Бағдарлама платформалық компоненттерден құрастырылған - бұл React компоненттеріне оралған жеке модульдер;

- кросс-платформа бар: View, Text, Image, Picker;
- iOS үшін ерекше: TabBarIOS, ActionSheetIOS;
- Android-ға тән: BackAndroid, ToolbarAndroid.

– CSS нақты емес - бұл полифиль

- абсолютті мәндерді енгізу, салыстырмалы мәндер жоқ;
- орналасу үшін flexbox қасиеттерінің шектеулі іске асырылуы бар;
- CSSx-ке толық қолдау болмайды, ол қажет емес;
- барлығы 70-ке жуық жылжымайтын мүлік іске асырылды, іс жүзінде

олар жеткілікті; әрдайым svg болған жағдайда ғана.



3.1-Сурет – React Navigation

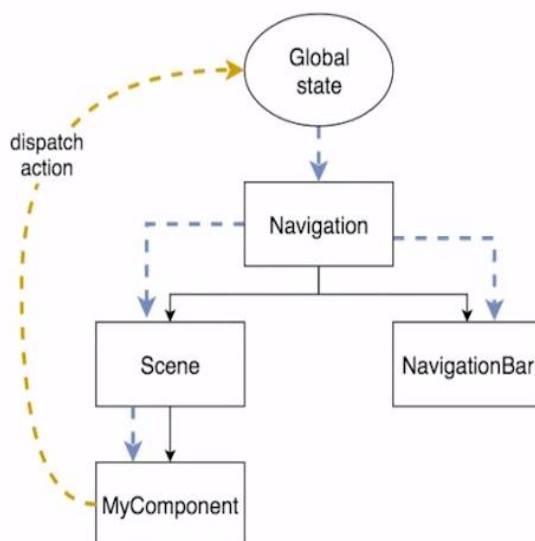
Бірқатар проблемалар бар:

- императивті API (әдістер)
- Навигатормен өзара әрекеттесу туралы бірден ойлану қажет
- анимациялар мен қимылдар басқаруға қиын
- Навигатор Бар жалпы өмірден толық шешілген
- бірнеше экран пайда болғаннан кейін ауырсыну басталады.

Көптеген жағдайларда проблемаларды Redux арқылы шешіледі.

NavigationBar немесе неге Redux қажет

MyComponent және NavigationBar сілтемесін қолдануға тырыспаңыз. Одан қарағанда global state и dispatch actions (flux/redux) қолданған дұрыс.



3.2-Сурет – Redux Navigation

2015 жылдың желтоқсанында Эрик Виченти navigation-rfc жобаны ұйымдастырып, қауымдастықтың көмегімен Навигатордың мәселелерін шешуге тырысты. 2016 жылдың ақпанында жоба ReAuT шеберлеріне «NavigationExperimental» атауымен көшті және қазіргі уақытта Facebook компаниясы әзірлейді. Ежелгі навигация енді қолдау көрсетілмейді.

3.1.1 NavigationExperimental

- жағдайды Redux арқылы басқарылады
- императивті API - әрекет диспетчерінің орнына
- навигация логикасы көріністен бөлінеді
- NavigationBar, анимация және басқару қимылдарын әртүрлі компоненттерге бөлу

3.1.2 Анимация

- Анимацияланған арнайы компонент арқылы жүзеге асырылады
- <Animated.View>, <Animated.Image>, <Animated.Text>
- отандық құрамдас бөліктерді тікелей жаңартатын реакциядан тыс жұмыс

Ол өте тегіс жұмыс істейді, сіз қатарлас / параллель біріктіре аласыз және өте ақылды заттар жасай аласыз.

3.1.3 Туынды модульдер

- React Native туынды модульдер негізін, бірақ бәрін жүзеге асырмайды.
- UIExplorer компонентін іздеу (React Native demo application)
 - OS модулі бойынша дұрыс модуль атауын табыңыз
 - бастапқы мәтіндерді қараңыз, құжатталмаған (суретті қалай жасау керек, SVG)
 - js.coach/react-native - реактивті түпнұсқа үшін плагиндер
 - JS емес, отандық іске асыруға артықшылық беріңіз

3.1.4 Жеке модульдерді қалай қосуға болады

Rnpm - React Native пакетінің менеджері:

```
$ npm link react-native react-native-google-analytics-bridge
```

3.1.5 Кроссплатформды компонент

Әрқайсысын әртүрлі файлдарға орналастыру дұрыс емес әдіс:

- common/components;
- android/components;
- ios/components.

оларды платформаға байланысты қосыңыз.

Дұрыс әдіс - бәрін бір файлдарға орналастыру:

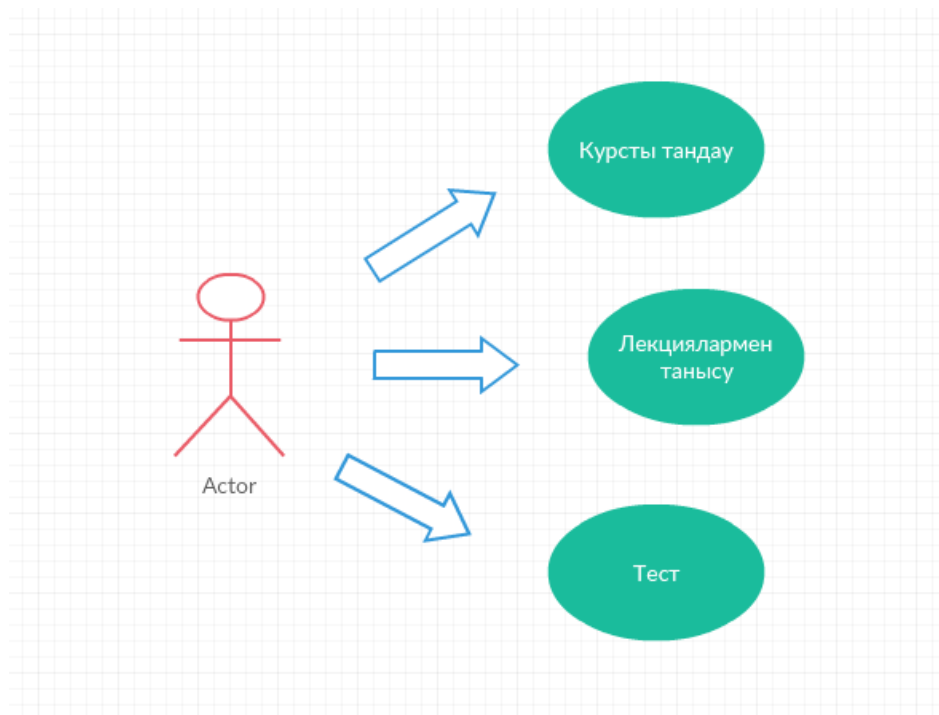
- MyComponent/Component.ios.js, MyComponent/Component.android.js;
- ComponentIOS, ComponentAndroid.

Платформаға тәуелді компоненттер үшін (ComponentIOS, ComponentAndroid) оған жанындағы қойындыны орналастыру ыңғайлы және компонент платформасында табылмаған проблемалар болмайды.

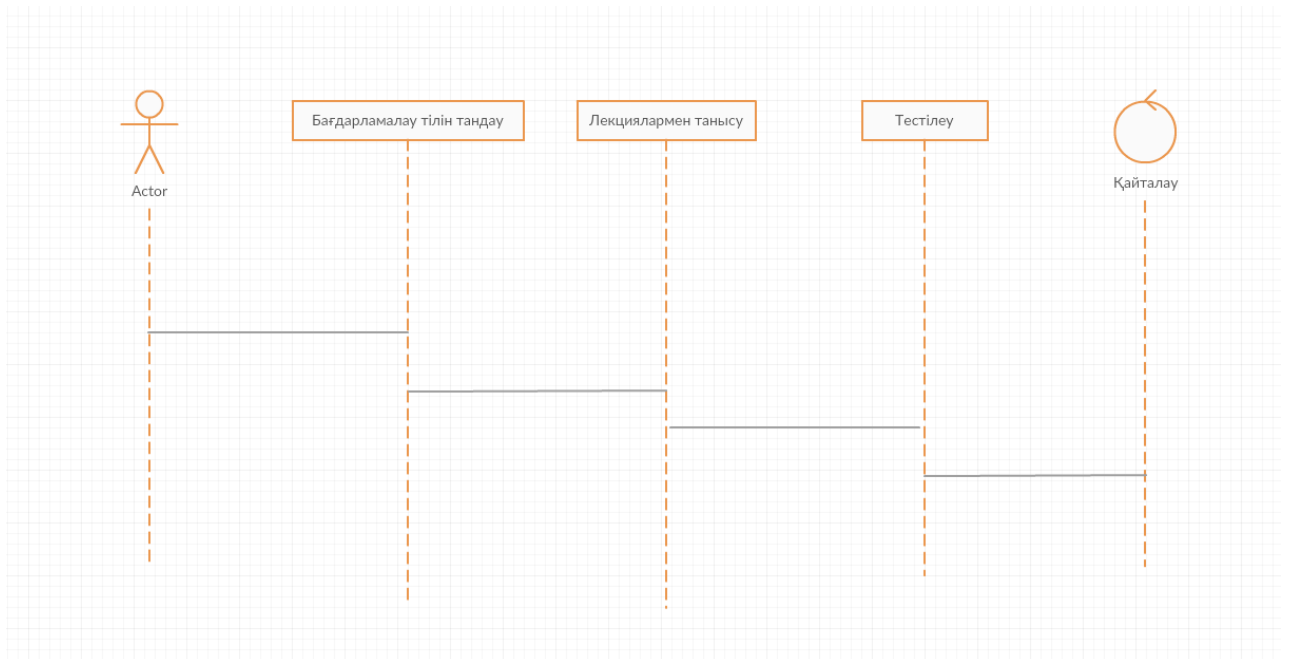
Туынды компонент қалай жазылуы:

- js-де барлық нәрселерді жазудың қажеті жоқ;
- компонент UIKit немесе Android API-де іске асырылуы тиіс.

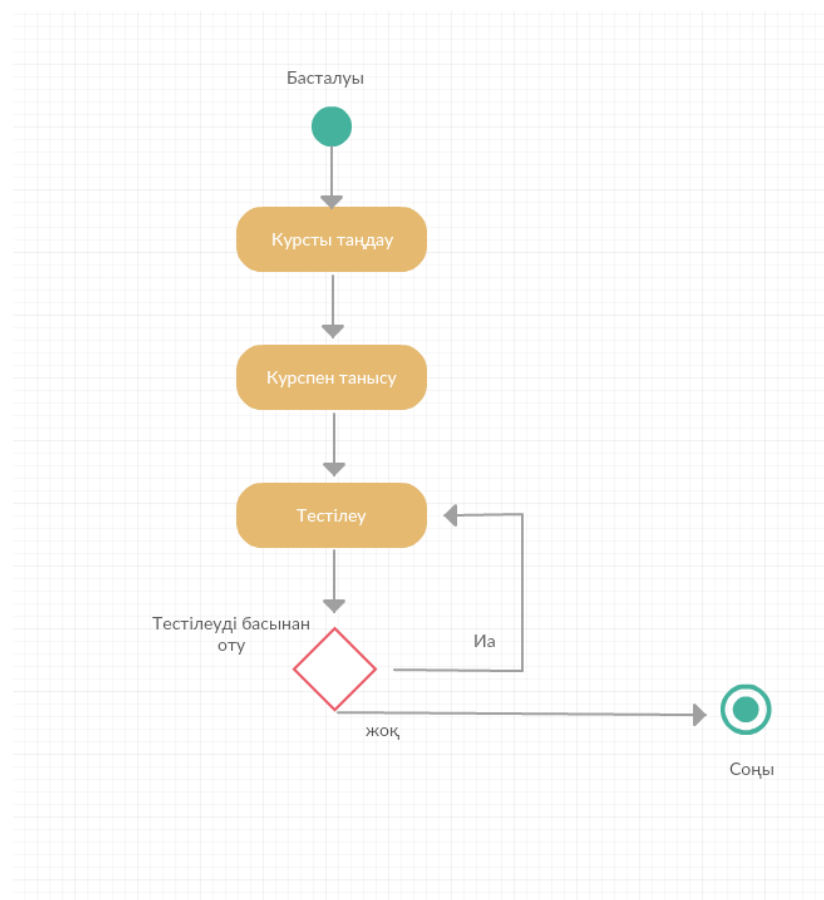
4 Бағдарламаны жүзеге асыру



4.1-Сурет – Use Case диграммасы



4.2-Сурет – Sequence диаграммасы

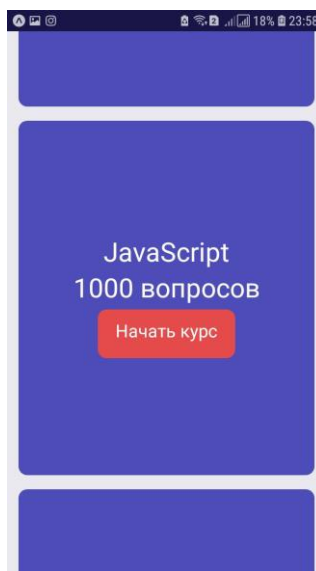


4.3-Сурет – Activity диаграммасы

4.1 Интерфейсті жобалау

Бағдарламаның дизайны өте қарапайым және интерефейсі түсінікті жасауға тырыстым. Бағдарламада бірнеше сценарийлер бар. Оларды диаграммадан сіз көре аласыз.

Бағдарламада бірнеше курстар бар. Бұл курстар қазіргі замандағы актуалды бағдарламалар.



4.1-Сурет – Бағдарлама мәзірі

Бағдарлама мәзірі. Бұл экранда қолданушы өзіне керекті курсты тандайды.

Stackoverflow сайтынан алынған статистика бойынша қазіргі замандағы ең танымал бағдарламалау тілдері Python, JavaScript, Java болып табылады. Сол үшін біріншіден осы бағдарламаларды енгізуді қарастырдым.



4.2-Сурет – Курс бағдарламасы

Курс бағдарламасы. Курс жайлы қысқаша мәлімет алып, өз тандауын жасау. Бұл парақшада 2 сценарий болады. Біріншісі, ол Курсты өту батырмасын басқан кезде сол курстын лекцияларына өтеді. Әрбір курста бірнеше лекциялар бар. Барлық лекциялар әртүрлі ресурстардан алынған



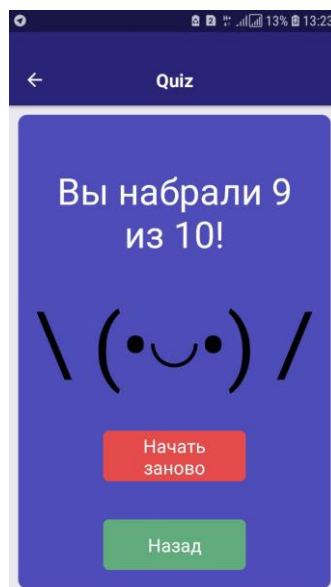
4.3-Сурет – Лекциялар тізімі

Бұл скринде курска байланысты лекциялар тізімі бар. Бұл лекциялар әртүрлі әдебиттерден алынған.



4.4-Сурет – Тест парақшасы

Бұл скринде тестілеу басталады. Бір тестілеу кезеңінде қолданушыға 10 сұрақ қойылады. Әрбір сұрақта 4 жауап нұсқалары бар. 4 жауап нұсқаларының 1-еуі ғана дұрыс.



4.5-Сурет – Тестілеу соңы

Тестілеу соңында қолданушыға алынған баллдар саны көрсетіледі. Қолданушы тестілеуді басынан осы жерден өте алады. Немесе бастапқы бетке ауыса алады. Егерде қолданушы 5 тен жоғары болса қуанып жатқан адамның эмоциясы көрсетіледі.

Қорытынды

Берілген дипломдық жобаның мақсаты қазіргі уақыттағы онлайн оқу ресурстарын, платформаларын саралап, заманға сай, барлық қолданушылардың жаңа технологияларға деген қызығушылығын оятатын, IT – мамандарға арналған онлайн оқу платформасын құру болатын.

Жаңа ақпараттық технологияның даму кезеңінде білім мен жаңа технологияны ұштастыру сапалы өнім береді. Қазірде үрдіс онлайн ұйымдастыруға, цифрландыру заманына көшуде.

Жұмыстың нәтижесінде бағдарламалау тілдерін үйренуге арналған бағдарлама жасалынды. Бұл бағдарламаның мақсаты қолданушыларға прогаммалау тілдерін үйрету.

Осы дипломдық жобаның шеңберінде JavaScript тіліне арналған React Native фреймворкімен жұмыс істеуді үйрендім.

Әдебиеттер тізімі

- 1 Эд Бурнет. Разработка мобильных приложений. — СПб.: Питер, 2012. — 256 с.
- 2 П.Дейтел, Х.Дейтел, М.Моргано. Android для программистов: создаем приложения. – СПб.: Питер, 2013. – 560 с.:ил.
- 3 Архитектура Android. – URL:<https://habr.com/ru/post/16770/>
- 4 Доскажанов Ч.Т., Даненова Г.Т., Коккоз М.М. РОЛЬ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ В СИСТЕМЕ ОБРАЗОВАНИЯ // Международный журнал экспериментального образования. – 2018.
- 5 Қазақстан Республикасының «Білім туралы» Заңы. – Астана,
- 6 ҚР БҒМ 2012 жылғы 6 сәуіріндегі № 151 бұйрығымен бекітілген Қазақстан Республикасының білім беру ұйымдарында оқу жетістіктерін сырттай бағалауды өткізу жөніндегі нұсқаулық. – Астана,
- 7 Доскажанов Ч.Т., Даненова Г.Т., Коккоз М.М. РОЛЬ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ В СИСТЕМЕ ОБРАЗОВАНИЯ // Международный журнал экспериментального образования. – 2018. – № 2. – С. 17-22;
URL: <http://www.expeducation.ru/ru/article/view?id=11790> (дата обращения: 21.04.2019).
- 8 Хашими С. Разработка приложений для Android / С. Хашими, С. Коматинени, Д. Маклин. – СПб.: Питер, 2011. – 736 с.
- 9 Сайт разработки в AndroidStudio. – URL: <https://developer.android.com> (дата обращения 4.02.2018).
- 10 Рето Майер. Professional Android 2: Application Developmentecond Edition. – М.: ЭКСМО, 2011. – 672 с.
- 11 Ресурс для IT-специалистов. ХабраХабр. – URL: <http://habrahabr.ru/> (дата обращения 4.02.2018).
- 12 Давыдов С. IntelliJ IDEA. Профессиональное программирование на Java. – СПб.: БХВ-Петербург, 2011. – 800 с.