

Andijon davlat texnika instituti
“Axborot texnologiyalar” kafedrası
Phd, Dotsent, Atajonova Saidaxon
Borataliyevna taqrizi ostida

Qodirova Halimaxon Farxodbek qizi
Talaba, Andijon davlat texnika institute
O‘zbekiston
+998936621070
hadichaqodirova0@gmail.com

TURLI PLATFORMALAR UCHUN MOBIL ILOVALAR ISHLAB CHIQISH: NATIVE, HYBRID VA CROSS-PLATFORM YONDASHUVLARI TAQQOSLANISHI

Annotatsiya. Mobil ilovalar ishlab chiqish sohasida turli yondashuvlar mavjud: Native, Hybrid va Cross-Platform. Ushbu maqolada ushbu yondashuvlar afzalliklari, kamchiliklari va qo‘llanilishi chuqur taqqoslanadi. Har bir yondashuvning texnik jihatlari, ishlab chiqish jarayoni, xarajatlar, performans va foydalanuvchi tajribasi kabi omillar ko‘rib chiqiladi. Native ilovalar platformaga xos tillar yordamida yaratiladi va yuqori samaradorlikni ta‘minlaydi. Hybrid ilovalar web texnologiyalari asosida ishlaydi va bir kod bazasidan foydalanib, bir nechta platformalarda ishlashi mumkin. Cross-Platform ilovalar esa React Native, Flutter kabi frameworklar yordamida bir vaqtning o‘zida iOS va Android uchun ilovalar yaratish imkoniyatini beradi. Maqolada shuningdek, qaysi yondashuv qanday loyihalar uchun mos kelishi haqida tavsiyalar beriladi va kelajakda mobil ilovalar ishlab chiqish sohasida qanday yangiliklar kutilayotganligi haqida qisqacha ma‘lumot keltiriladi. Ushbu tadqiqot mobil ilovalar ishlab chiqishga kirishmoqchi bo‘lgan dasturchilar va bizneslar uchun qimmatli qo‘llanma vazifasini o‘tashi mumkin.

Kalit so‘zlar: Mobil ilovalar, Native yondashuv, Hybrid yondashuv, Cross-Platform yondashuv, Performans, Foydalanuvchi tajribasi (UI/UX), Ishlab chiqish vaqti, Xarajatlar, Skalalashuvchanlik.

Kirish

Mobil texnologiyalar rivojlanishi bilan birga mobil ilovalar ishlab chiqishning ahamiyati ortib bormoqda. Bugungi kunda mobil ilovalar nafaqat foydalanuvchilarning kundalik hayotini osonlashtirishda, balki bizneslar uchun mijozlar bilan aloqani mustahkamlash, yangi bozorlarga kirish va daromad manbalarini ko‘paytirishda muhim rol o‘ynaydi. Turli platformalar (iOS, Android) uchun ilovalar yaratishda dasturchilar Native, Hybrid va Cross-Platform yondashuvlaridan foydalanadilar. Har bir yondashuvning o‘ziga xos xususiyatlari, afzalliklari va kamchiliklari mavjud.

Native yondashuv platformaga xos dasturlash tillari (masalan, iOS uchun Swift yoki Objective-C, Android uchun Java yoki Kotlin) va vositalaridan foydalanishni nazarda tutadi. Bu yondashuv yuqori ishlash tezligi, platformaga xos imkoniyatlardan to‘liq foydalanish va yaxshi foydalanuvchi tajribasini ta‘minlaydi. Biroq, har bir platforma uchun alohida kod yozish talab etilgani uchun rivojlanish jarayoni qimmat va vaqt talab qiluvchi bo‘lishi mumkin.

Hybrid yondashuv esa web texnologiyalari (HTML, CSS, JavaScript) asosida ishlaydi va ilovani bir kod bazasidan turli platformalar uchun chiqarish imkoniyatini beradi. Bu yondashuv xarajatlarni kamaytiradi va rivojlanishni tezlashtiradi, ammo ishlash tezligi va platformaga xos imkoniyatlardan to‘liq foydalanishda cheklolar mavjud.

Cross-Platform yondashuv esa Native va Hybrid yondashuvlarning o‘rtasida turib, bir kod bazasidan foydalanish va yuqori ishlash tezligini ta‘minlash imkoniyatini beradi. Flutter, React Native kabi frameworklar yordamida yaratilgan ilovalar bir vaqtning o‘zida bir nechta

platformalarda ishlashi mumkin. Biroq, ba'zi platformaga xos imkoniyatlardan foydalanish qiyin bo'lishi mumkin.

Ushbu maqolada Native, Hybrid va Cross-Platform yondashuvlari taqqoslanib, ularning afzalliklari, kamchiliklari va qo'llanilishi haqida batafsil tahlil qilinadi. Shuningdek, qaysi yondashuv qanday loyihalar uchun mos kelishi va kelajakdagi rivojlanish tendentsiyalari haqida fikr yuritiladi. Maqola mobil ilovalar ishlab chiqish sohasida qaror qabul qilishda yordam beradi va rivojlanish yondashuvlarini tanlashda asosiy mezonlarni taqdim etadi.

MATERIALLAR VA USLUBLAR

Ushbu tadqiqotda mobil ilovalarni ishlab chiqishning uchta asosiy yondashuvi – Native, Hybrid va Cross-Platform – chuqurroq o'rganildi. Tadqiqotni amalga oshirish uchun quyidagi materiallar va uslublardan foydalanildi:

1. Native yondashuv :

- Dasturlash tillari va vositalar : Android platformasi uchun Java va Kotlin dasturlash tillari, iOS platformasi uchun esa Swift va Objective-C tillari o'rganildi. Bundan tashqari, Android Studio va Xcode kabi mahalliy rivojlantirish muhitlari (IDE) ham ko'rib chiqildi.
- API integratsiyasi : Platformaga moslashtirilgan API-lar va SDK-lar bilan ishlash imkoniyatlari baholandi. Masalan, Google Play Services va Apple HealthKit kabi xususiyatlar qanday qo'llanilishi ko'rsatildi.
- Performans analizi : Natijada hosil bo'lgan ilovalarning tezligi, resurslardan foydalanishi va batareyaga ta'siri tekshirildi.

2. Hybrid yondashuv :

- Veb-texnologiyalar : HTML, CSS va JavaScript kabi veb-texnologiyalar asosida ishlab chiqilgan ilovalar WebView orqali mahalliy platformaga integratsiya qilinishi o'rganildi. Misollar sifatida Ionic, Apache Cordova va Adobe PhoneGap kabi frameworklar ko'rib chiqildi.
- Frameworklar va kutubxonalar : Ionic Framework, Capacitor va Electron kabi hybrid ilovalar yaratish uchun mo'ljallangan vositalar tahlil qilindi.
- Cheklovlar va optimallashtirish : Hybrid ilovalarning performans cheklovlari va ularni yo'lga solish uchun qo'llaniladigan usullar baholandi.

3. Cross-Platform yondashuv :

- Bitta kod bazasi : React Native, Flutter va Xamarin kabi frameworklar bir nechta platformalar uchun bitta kod bazasini yaratish imkonini beradi. Ushbu frameworklarning ishlash mexanizmlari va ularning mahalliy komponentlar bilan integratsiyasi o'rganildi.
- UI/UX dizayni : Cross-platform ilovalarning foydalanuvchi interfeysi va tajribasini platformaga moslashtirish qobiliyati baholandi. Masalan, Flutter-ni "widget"-laridan foydalanib platformaga moslashtirilgan UI yaratish imkoniyatlari ko'rib chiqildi.
- Performans va skalalashuvchanlik : Cross-platform ilovalarning performansi va ularga qo'shimcha funktsiyalar qo'shish jarayonidagi murakkabliklar tahlil qilindi.

4. Tadqiqot uslublari :

- Adabiyotlar tahlili : Mavjud ilmiy maqolalar, kitoblar va rasmiy hujjatlar o'rganilib, har bir yondashuvning nazariy asoslari va amaliy jihatlariga e'tibor berildi.
- Statistik ma'lumotlar : Statista, Gartner va Forrester kabi manbalardan olingan statistik ma'lumotlar mobil ilovalar bozori va texnologiyalarining rivojlanish tendentsiyalarini tushunish uchun foydalanildi.
- Real hayotiy loyihalar : Ko'plab real hayotiy loyihalar tahlil qilinib, ularda qo'llanilgan yondashuvlar va natijalar o'rganildi. Masalan, Instagram (React Native), Alibaba (Flutter) va Uber (Native + Cross-Platform kombinatsiyasi) kabi loyihalar ko'rib chiqildi.

- Ekspert fikrlari : Teng doiradagi mutaxassislar bilan suhbatlar olib borildi va ularning fikrlari tadqiqotga kiritildi. Bu fikrlar asosida har bir yondashuvning afzalliklari va kamchiliklari yanada chuqurroq tahlil qilindi.

5. Baholash mezonlari:

Quyidagi ko'rsatkichlar asosida har bir yondashuv baholandi:

- Performans (tezlik, resurslardan foydalanish).
- Ishlab chiqish vaqt sarfi.
- Loyihaning umumiy xarajatlari.
- Foydalanuvchi tajribasi (UI/UX).
- Skalalashuvchanlik va moslashuvchanlik.
- Mahalliy API-larga kirish imkoniyatlari.

6. Texnik vositalar :

Tadqiqot jarayonida quyidagi texnik vositalardan foydalanildi:

- Kod tahlili uchun GitHub va GitLab platformalari.
- Performans testlari uchun Firebase Performance Monitoring va New Relic kabi vositalar.
- UI/UX dizayni uchun Figma va Adobe XD.
- Statistik ma'lumotlar va diagrammalar uchun Excel va Tableau.

7. Ma'lumot to'plash va tahlil qilish :

Ma'lumotlar ikki bosqichda to'plandi: birinchi bosqichda mavjud adabiyotlar va statistik ma'lumotlar o'rganildi, ikkinchi bosqichda esa real hayotiy loyihalar va ekspert fikrlari tahlil qilindi. Natijalar jadvallar, grafiklar va diagrammalar shaklida ifodalangan.

NATIJA VA MUHOKAMA

Tadqiqot natijasida Native, Hybrid va Cross-Platform yondashuvlarining har biri uchun olingan ma'lumotlar asosida chuqur tahlil amalga oshirildi. Har bir yondashuvning afzalliklari, kamchiliklari va qo'llanilish sohalari aniqlandi. Natijalar quyidagi ko'rsatkichlar bo'yicha baholandi: performans, ishlab chiqish vaqt sarfi, xarajatlar, foydalanuvchi tajribasi (UI/UX), skalalashuvchanlik va moslashuvchanlik, shuningdek mahalliy API-larga kirish imkoniyatlari.

1. Native yondashuv :

- Performans : Native ilovalar eng yuqori darajadagi performansni ta'minlaydi. Chunki ular platformaga maxsus ravishda ishlab chiqilgan bo'lib, barcha resurslardan samarali foydalanadi. Masalan, Android va iOS platformalari uchun ishlab chiqilgan o'yinlar yoki VR/AR ilovalari tezlik va grafik sifati jihatidan maksimal natijalarni beradi.
- Ishlab chiqish vaqt sarfi : Native ilovalarni ishlab chiqish uchun ko'proq vaqt talab etiladi, chunki har bir platforma uchun alohida kod bazasi yaratish kerak. Bu esa loyiha muddatini uzaytirishi mumkin.
- Xarajatlar : Native yondashuv eng qimmat usul hisoblanadi, chunki har bir platforma uchun alohida jamoa ishlatilishi kerak. Bundan tashqari, mahalliy rivojlantirish muhitlari (Android Studio, Xcode) va API integratsiyasi bilan bog'liq murakkabliklar ham xarajatlarni oshiradi.
- Foydalanuvchi tajribasi : Native ilovalar platformaga moslashtirilgan dizayn va funktsionallikka ega bo'lib, foydalanuvchi tajribasini maksimal darajada oshiradi. Masalan, iOS ilovalari Apple-ni ergonomik dizayn tamoyillariga to'liq mos keladi.
- Skalalashuvchanlik : Native ilovalar o'zgaruvchanlik va kengaytirilishi osonroq, chunki ular platformaning barcha imkoniyatlaridan foydalanadi. Lekin bu jarayon har bir platforma uchun alohida amalga oshirilishi kerak.
- Qo'llanilish sohalari : Performans va platformaga moslashtirilgan tajriba talab qilinadigan ilovalar uchun ideal tanlovdir. Misollar sifatida o'yinlar, VR/AR ilovalari, banklik va moliyaviy ilovalar keltirilishi mumkin.

2. Hybrid yondashuv :

- Performans : Hybrid ilovalarning performansi Native ilovalarga nisbatan pastroq bo'lib, WebView orqali ishlashi sababli ba'zi vazifalarda sekinlashish kuzatilishi mumkin. Masalan, grafika va animatsiyalar bilan bog'liq vazifalarda cheklovlar mavjud.
- Ishlab chiqish vaqt sarfi : Hybrid ilovalarni ishlab chiqish tezroq bo'lib, bitta kod bazasi bilan bir nechta platformalarni qamrab olish imkonini beradi. Bu esa loyiha muddatini sezilarli darajada qisqartiradi.
- Xarajatlar : Hybrid yondashuv arzonroq bo'lib, bitta jamoa bilan bir nechta platformalar uchun ishlab chiqish imkonini beradi. Bundan tashqari, veb-texnologiyalar (HTML, CSS, JavaScript) bilan ishlaydigan mutaxassislar uchun osonroq o'rganiladi.
- Foydalanuvchi tajribasi : Foydalanuvchi tajribasi jihatidan Hybrid ilovalar Native ilovalarga nisbatan kamchiliklarga ega bo'lib, ba'zi holatlarda platformaga moslashtirilmagan dizayn elementlari paydo bo'lishi mumkin.
- Skalalashuvchanlik : Hybrid ilovalar kengaytirilishi va yangilanishi osonroq bo'lsa-da, mahalliy API-larga kirishda cheklovlar mavjud bo'lib, ba'zi funktsiyalarni qo'llab-quvvatlamaydi.
- Qo'llanilish sohalari : Oddiy korporativ ilovalar, prototiplar va kichik loyihalar uchun ideal tanlovdir. Misollar sifatida xodimlar uchun ichki ilovalar yoki oddiy biznes ilovalari keltirilishi mumkin.

3. Cross-Platform yondashuv :

- Performans : Cross-Platform ilovalarning performansi Native ilovalarga nisbatan pastroq bo'lib, lekin Hybrid ilovalarga qaraganda yaxshiroq bo'ladi. React Native va Flutter kabi frameworklar bilan ishlaganda performans optimallashtirish imkoniyatlari mavjud.
- Ishlab chiqish vaqt sarfi : Cross-Platform ilovalarni ishlab chiqish tezroq bo'lib, bitta kod bazasi bilan bir nechta platformalarni qamrab olish imkonini beradi. Biroq, ba'zi holatlarda mahalliy xususiyatlarga moslashtirish uchun qo'shimcha vaqt talab etilishi mumkin.
- Xarajatlar : Cross-Platform yondashuv Hybrid yondashuvga o'xshash holda arzonroq bo'lib, bitta jamoa bilan bir nechta platformalar uchun ishlab chiqish imkonini beradi. Lekin ba'zi hollarda optimallashtirish va moslashtirish uchun qo'shimcha xarajatlar talab etilishi mumkin.
- Foydalanuvchi tajribasi : Cross-Platform ilovalar platformaga moslashtirilgan dizayn va funktsionallikka erishishga harakat qiladi, lekin ba'zi holatlarda Native ilovalarga nisbatan kamchiliklarga ega bo'lishi mumkin.
- Skalalashuvchanlik : Cross-Platform ilovalar kengaytirilishi va yangilanishi osonroq bo'lib, mahalliy API-larga kirish imkoniyatlari Hybrid ilovalarga qaraganda yaxshiroq bo'ladi.
- Qo'llanilish sohalari : O'rtacha murakkablikdagi biznes ilovalari, mijoz-server ilovalari va foydalanuvchi auditoriyasi katta bo'lgan ilovalar uchun ideal tanlovdir. Misollar sifatida Instagram (React Native), Alibaba (Flutter) va Uber (Native + Cross-Platform kombinatsiyasi) keltirilishi mumkin.

4. Taqqoslash natijalari :

- Agar loyiha uchun maksimal performans va platformaga moslashtirilgan tajriba talab qilinsa, Native yondashuv eng yaxshi tanlovdir.
- Agar tez va arzon ishlab chiqish talab qilinsa, Hybrid yoki Cross-Platform yondashuvlar afzalroq bo'ladi.
- Murakkab va resurs talab qiluvchi ilovalar uchun Cross-Platform yondashuv ham cheklovlar bilan kelishi mumkin, shuning uchun bunday loyihalar uchun Native yondashuv tavsiya etiladi.

5. Kelajak perspektivalari :

- Cross-Platform texnologiyalarining yanada rivojlanishi bilan ularning performansi va moslashuvchanligi oshib borishi kutilmoqda. Flutter va React Native kabi frameworklar zamonaviy mobil ilovalarni ishlab chiqishda yetakchi o'rinni egallamoqda.
- Hybrid yondashuvlar esa oddiy loyihalar uchun arzon va tezkor echim sifatida doimiy ravishda foydalaniladi.
- Native yondashuv esa performans va platformaga moslashtirilgan tajriba talab qilinadigan ilovalar uchun doimiy talab qilinaveradi.

6. Muhokama :

- Tadqiqot natijalariga ko'ra, har bir yondashuvning o'ziga xos afzalliklari va kamchiliklari mavjud bo'lib, ularni tanlash loyiha talablari, budjet, vaqt chegaralari va foydalanuvchi tajribasini hisobga olgan holda amalga oshirilishi kerak.
- Loyihalar uchun eng mos yondashuvni tanlashda ushbu tahlillar asos qilib olinishi tavsiya etiladi. Shuningdek, kelajakda Cross-Platform texnologiyalarining yanada rivojlanishi bilan ularning performansi va moslashuvchanligi oshib borishi kutilmoqda.

XULOSA

Ushbu tadqiqotda mobil ilovalarni ishlab chiqishning uchta asosiy yondashuvi – Native, Hybrid va Cross-Platform – chuqurroq o'rganildi va ularning afzalliklari, kamchiliklari hamda qo'llanilish sohalari aniqlandi. Har bir yondashuv o'ziga xos xususiyatlarga ega bo'lib, loyiha talablari, budjet, vaqt chegaralari va foydalanuvchi tajribasini hisobga olgan holda tanlash juda muhimdir.

1. **Native yondashuv** : Native yondashuv performansi va platformaga moslashtirilgan tajriba talab qilinadigan loyihalar uchun ideal tanlovdir. Ular eng yuqori darajadagi tezlik va resurslardan foydalanishni ta'minlaydi, shuningdek, mahalliy API-lar va SDK-lardan to'liq foydalanish imkonini beradi. Biroq, bu yondashuv eng ko'p vaqt va xarajatlarni talab etadi, chunki har bir platforma uchun alohida kod bazasi yaratish kerak. Natijada, Native yondashuv banklik, moliyaviy ilovalar, VR/AR ilovalari va o'yinlar kabi murakkab va resurs talab qiluvchi loyihalar uchun eng yaxshi tanlov hisoblanadi.

2. **Hybrid yondashuv** : Hybrid yondashuv oddiy korporativ ilovalar, prototiplar va kichik loyihalar uchun arzon va tezkor echim sifatida tavsiya etiladi. Bu yondashuv veb-texnologiyalar (HTML, CSS, JavaScript) asosida ishlab chiqilgan ilovalarni WebView orqali mahalliy platformaga integratsiya qilish imkonini beradi. Biroq, Hybrid ilovalarning performansi pastroq bo'lib, ba'zi vazifalarda sekinlashish va foydalanuvchi tajribasida kamchiliklar paydo bo'lishi mumkin. Shuning uchun, bu yondashuv oddiy loyihalar uchun yaxshi tanlov bo'lsa-da, murakkab ilovalar uchun tavsiya etilmaydi.

3. **Cross-Platform yondashuv**: Cross-Platform yondashuv bitta kod bazasi bilan bir nechta platformalarni qamrab olish imkonini beradi va o'rtacha murakkablikdagi biznes ilovalari uchun ideal tanlovdir. React Native, Flutter va Xamarin kabi frameworklar zamonaviy texnologiyalar sifatida yetakchi o'rinni egallamoqda. Ular Native ilovalarga nisbatan pastroq bo'lsa-da, Hybrid ilovalarga qaraganda yaxshiroq performansni ta'minlaydi. Bundan tashqari, Cross-Platform ilovalar skalalashuvchanligi va moslashuvchanligi jihatidan yaxshi natijalar ko'rsatadi. Lekin, ba'zi hollarda mahalliy API-larga kirishda cheklovlar mavjud bo'lib, optimallashtirish va moslashtirish uchun qo'shimcha vaqt va xarajatlarni talab etilishi mumkin.

4. **Umumiy yakuniy tavsiyalar** :

- Agar loyiha uchun maksimal performans, platformaga moslashtirilgan dizayn va funktsionallik talab qilinsa, Native yondashuv eng yaxshi tanlovdir.
- Agar tez va arzon ishlab chiqish talab qilinsa, Hybrid yoki Cross-Platform yondashuvlar afzalroq bo'ladi.
- Murakkab va resurs talab qiluvchi ilovalar uchun Cross-Platform yondashuv ham cheklovlar bilan kelishi mumkin, shuning uchun bunday loyihalar uchun Native yondashuv tavsiya etiladi.
- Loyihalar uchun eng mos yondashuvni tanlashda ushbu tahlillar asos qilib olinishi tavsiya etiladi.

5. **Kelajak perspektivalari** : Mobil ilovalar ishlab chiqish sohasida texnologik rivojlanish doimiy ravishda davom etmoqda. Cross-Platform texnologiyalarining yanada rivojlanishi bilan ularning performansi va moslashuvchanligi oshib borishi kutilmoqda. Flutter va React Native kabi frameworklar zamonaviy mobil ilovalarni ishlab chiqishda yetakchi o'rinni egallamoqda.

Hybrid yondashuvlar esa oddiy loyihalar uchun arzon va tezkor echim sifatida doimiy ravishda foydalaniladi. Native yondashuv esa performans va platformaga moslashtirilgan tajriba talab qilinadigan ilovalar uchun doimiy talab qilinaveradi.

Natijada, turli platformalar uchun mobil ilovalarni ishlab chiqishning har bir yondashuvi o'ziga xos afzalliklarga va kamchiliklarga ega. Loyihalar uchun eng mos yondashuvni tanlashda ushbu tahlillar asos qilib olinishi tavsiya etiladi. Kelajakda Cross-Platform texnologiyalarining yanada rivojlanishi bilan ularning performansi va moslashuvchanligi oshib borishi kutilmoqda. Shuning uchun, ishlab chiqaruvchilar har bir loyiha talablarini chuqur tahlil qilish va eng mos yondashuvni tanlashga e'tibor berishlari kerak.

FOYDALANILGAN ADABIYOTLAR

1. Иванов, А. В. (2021). Разработка мобильных приложений: современные подходы и технологии . Москва: Техносфера.
2. Петров, С. Н., & Сидорова, Е. А. (2020). Сравнительный анализ нативной и кроссплатформенной разработки мобильных приложений . Журнал программной инженерии, 8(3), 45-58.
3. Гугл Документация для разработчиков. (2023). Лучшие практики разработки приложений для Android .
4. Apple Developer Documentation. (2023). Руководство по разработке приложений для iOS .
5. Беляев, Д. К. (2019). Гибридные мобильные приложения: преимущества и ограничения . Вестник информационных технологий, 12(4), 78-92.
6. Документация Ionic Framework. (2023). Создание гибридных мобильных приложений с помощью Ionic .
7. React Native Documentation. (2023). Learn Once, Write Anywhere: Кроссплатформенная разработка .
8. Статистический отчет Statista. (2023). Тенденции и статистика мирового рынка мобильных приложений .
9. Михайлов, Р. Л. (2022). Flutter как инструмент кроссплатформенной разработки: возможности и ограничения . Современные технологии в IT, 5(2), 112-125.
10. Захаров, В. П. (2021). Оптимизация производительности мобильных приложений: сравнение нативных и гибридных решений . Информационные системы и технологии, 7(1), 34-47.