

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

Аліна САВЧЕНКО

“ _____ ” _____ 2021 р.

ДИПЛОМНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ

“МАГІСТРА”

ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ “ІНФОРМАЦІЙНІ
УПРАВЛЯЮЧІ СИСТЕМИ ТА ТЕХНОЛОГІЇ”

Тема: “Алгоритм проектування та розробки додатку на базі iOS”

Виконавець: Сунко Дмитро Володимирович

Керівник: професор Воронін Альберт Миколайович

Нормоконтролер: _____ Ігор РАЙЧЕВ

Київ - 2021

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра Комп'ютерних інформаційних технологій

Галузь знань, спеціальність, освітньо-професійна програма: 12 “Інформаційні технології”, 122 “Комп'ютерні науки”, “Інформаційні управляючі системи та технології”

ЗАТВЕРДЖУЮ

Завідувач кафедри

Аліна САВЧЕНКО

«____» _____ 2021р.

ЗАВДАННЯ

на виконання дипломної роботи студента

Сунка Дмитра Володимировича

(прізвище, ім'я, по батькові)

- 1. Тема роботи:** «Алгоритм проектування та розробки додатку на базі iOS»
затверджена наказом ректора від 12.10.2021 за № 2228/ст.
- 2. Термін виконання роботи:** з 12.10.2021 по 31.12.2021.
- 3. Вихідні дані до роботи:** розробка додатку для роботи зі списком справ на базі iOS.
- 4. Зміст пояснювальної записки:** вступ, аналіз існуючих типів додатків та їх особливостей, аналіз інструментів для створення додатку на платформі iOS, створення дизайн проекту, створення інтерфейсу додатка, створення додатка, висновок.
- 5. Перелік обов'язкового ілюстративного матеріалу:** кількість користувачів різних мобільних додатків, макети wireframe, макети mockup.

6. Календарний план-графік

№ п/п	Завдання	Термін виконання	Підпис керівника
1.	Отримання завдання на дипломну роботу та побудова плану-графіку виконання робіт.	12.10.2021 – 15.10.2021	
2.	Аналіз літератури та джерел за темою дипломного проекту.	16.10.2021 – 19.10.2021	
3.	Розробка та затвердження плану дипломного проекту.	20.10.2021 – 24.10.2021	
4.	Проведення консультації з науковим керівником щодо створення першого розділу.	25.10.2021 – 31.10.2021	
5.	Розробка розділу 1. ОПИС ІСНУЮЧИХ ДОДАТКІВ НА БАЗІ iOS.	01.11.2021 – 07.11.2021	
6.	Розробка розділу 2. АНАЛІЗ ТА ВИБІР ІНСТРУМЕНТІВ ДЛЯ СТВОРЕННЯ ДОДАТКУ НА БАЗІ iOS.	08.11.2021 – 17.11.2021	
7.	Розробка розділу 3. ПРОЕКТ МОБІЛЬНОГО ДОДАТКУ НА БАЗІ iOS.	18.11.2021 – 01.12.2021	
8.	Розробка додатку на базі iOS.	02.12.2021 – 11.12.2021	
9.	Висновки та оформлення пояснювальної записки дипломного проекту.	12.12.2021 – 20.12.2021	

7. Дата видачі завдання: 12.10.2021р.

Керівник дипломної роботи _____ Альберт ВОРОНІН

(підпис керівника)

Завдання прийняв до виконання _____ Дмитро СУНКО

(підпис випускника)

РЕФЕРАТ

Пояснювальна записка до дипломної роботи “Алгоритм проектування та розробки додатку на базі iOS” складається із вступу, трьох розділів, загальних висновків, списку використаних джерел і містить 93 сторінки тексту, 29 рисунків та 1 таблицю. Список використаних джерел містить 11 найменувань.

Метою дипломної роботи є створення мобільного додатку для допомоги з організації особистого розкладу дня.

Предметом дослідження є структура, категорії та розробка мобільних додатків.

Об’єктом дослідження є мобільні додатки.

Ключові слова: СОЦІАЛЬНА МЕРЕЖА, USER INTERFACE, USER EXPERIENCE, WIREFRAME, MOCKUP, iOS.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	7
ВСТУП.....	8
РОЗДІЛ 1. ОПИС ІСНУЮЧИХ ДОДАТКІВ НА БАЗІ iOS	9
1.1. Категорії мобільних додатків.....	9
1.1.1. Соціальні мережі	9
1.1.2. Месенджери.....	17
1.1.3. Ігри.....	20
1.1.4. Додатки для бізнесу	28
1.1.5. Додатки для заняття фітнесом	30
1.1.6. Додатки для навігації.....	32
1.1.7. Розвиваючі додатки	33
1.1.8. Утиліти	35
1.2. Висновок до першого розділу.....	36
РОЗДІЛ 2. АНАЛІЗ І ВИБІР ІНСРУМЕНТІВ ДЛЯ СТВОРЕННЯ ДОДАТКУ НА БАЗІ iOS	38
2.1. Аналіз і вибір типу додатка.....	38
2.1.1. Нативні додатки	38
2.1.2. Мобільні веб-додатки	39
2.1.3. Гібридні додатки	40
2.2. Аналіз і вибір мови програмування	42
2.2.1. Swift	42
2.2.2. Objective-C	45
2.2.3. Порівняння Swift та Objective-C.....	46
2.3. Стандарти та правила UI/UX дизайну в iOS	52
2.4. Аналіз існуючих сервісів для створення UI/UX дизайну	56
2.4.1. Порівняння Scetch та Figma	60
2.5. Висновок до другого розділу	61
РОЗДІЛ 3. ПРОЕКТ МОБІЛЬНОГО ДОДАТКУ НА БАЗІ iOS	63

3.1. Проектування дизайну	63
3.1.1. Wireframe	63
3.1.2. Mockup	73
3.2. Розробка додатку	80
3.2.1. Модель (Model)	82
3.2.2. Уявлення (View)	83
3.3. Висновок до третього розділу	91
ВИСНОВКИ	92
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	93

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

UI – *User Interface* – інтерфейс користувача.

UX – *User Experience* – досвід користувача.

ПЗ – Програмне забезпечення.

WF – *Wireframe* – каркас дизайну.

Lo-Fi-WF – *Low-Fidelity Wireframe* – низькорівневий каркас дизайну.

Mi-Fi-WF – *Middle-Fidelity Wireframe* – середньорівневий каркас дизайну.

Hi-Fi-WF – *High-Fidelity Wireframe* – високорівневий каркас дизайну.

Мейнстрім – переважний напрямок у якоїсь галузі (наукової, культурної та ін.) для певного відрізка часу.

ВСТУП

За останні 30 років технології досягли такого розвитку, що людство може робити штучне м'ясо, саджати космічні літальні апарати на комети, робити багаторазові ракети, створювати квантові комп'ютери, випускати безпілотні автомобілі, створювати синтетичну ДНК та незкінченну множину всього іншого. Але це все було б неможливо без стрімкого розвитку комп'ютерних наук та цифрових технологій. Адже саме вони дозволили перекласти багато задач з людей на машини.

Сьогодні світ вже неможливо собі уявити без сучасних технологій, які, в останній час, дуже стрімко розвиваються. Вчені з усього світу намагаються винайти щось таке, що змінить сучасні реалії раз і назавжди.

Одним із найважливіших і найбільш цінних факторів розвитку є комп'ютери, планшети, смартфони та інша техніка, яка стала невід'ємною складовою нашого життя. Такі пристрої можуть виконувати безліч корисних наукових задач, наприклад: розв'язання складних математичних задач; моделювання різних погодних умов; прогнозування землетрусів, вивержень вулканів, здвигів тектонічних плит та безліч інших речей.

Але і для звичайних людей сучасні комп'ютерні технології та пристрої стануть у нагоді, адже існують тисячі різних за функціоналом додатків, які слугують для різних цілей. Є такі типи додатків, як: соціальні мережі, месенджери, ігри, додатки для бізнесу, для заняття фітнесом, для навігації, розвиваючі, утиліти та інші. За допомогою них, можна дивитись фільми, серіали, грати в ігри, займатися саморозвитком, спілкуватись з іншими людьми прямо у себе вдома, та, навіть, віддалено слідкувати за своїм бізнесом. Це все спонукає на створення власного додатку, який буде корисним для людей, і, в той же час, приносить прибуток своєму розробнику.

РОЗДІЛ 1.

ОПИС ІСНУЮЧИХ ДОДАТКІВ НА БАЗІ iOS

1.1. Категорії мобільних додатків

1.1.1. Соціальні мережі

Звичайна людина, здебільшого, використовує певну частину найбільших у світі соціальних мереж (Facebook, Twitter, LinkedIn) та сайти для обміну медіа контентом (Instagram, YouTube, Snapchat), а також, можливо, ще кілька таких мереж, як Pinterest та Google Plus.

Але культура соціальних мереж не закінчується на провідних соціальних мережах та мережах обміну медіа-контентом. Подивившись далеко за межі цих столпів у світі соціальних мереж, можна побачити, що люди використовують безліч різних типів соціальних мереж.

І хоча колись був час, коли можна було просто класифікувати мережі за їх функціональними можливостями (Twitter для короткого текстового вмісту, YouTube для відео тощо), цей час минув. Оскільки все більше мереж додають такі розширені функції, як пряма трансляція та доповнена реальність, межі між їхніми наборами функцій продовжують розмиватися і змінюватися швидше, ніж більшість людей встигає прочитати зміни.

Отже, замість того, щоб класифікувати мережі відповідно до гіперспецифічних інтересів користувачів або змінювати технологічні особливості, треба групувати мережі в загальні категорії, які зосереджуються на тому, що люди сподіваються досягти, використовуючи їх.

Кафедра КІТ (47)				НАУ 21.38.27.000 ПЗ			
Виконав	Сунко Д.В.			ОПИС ІСНУЮЧИХ ДОДАТКІВ НА БАЗІ iOS	Літера	аркуш	аркушів
Керівник	Воронін А.М.					9	29
Консульт.							
Н. контроль	Райчев І.Е.						
					УС-212М	122	

Ось список із 9 типів соціальних медіа та для чого вони використовуються:

- Соціальні мережі для спілкування – спілкування з людьми;
- Мережі обміну медіа-контентом – обмін фотографіями, відео та іншими мультимедіа;
- Дискусійні форуми – можливість ділитися новинами та ідеями;
- Сервіси соціальних закладок – можливість відкривати, зберігати та ділитись новою інформацією;
- Мережі для відгуків – знаходити та переглядати різні компанії;
- Блоги та видавничі мережі – публікувати думки в Інтернеті;
- Мережі на основі інтересів – обмінюватись інтересами та захопленнями;
- Соціальні торгові мережі – можливість робити покупки через Інтернет;
- Анонімні соціальні мережі – анонімне спілкування.

Знання про ці категорії соціальних медіа та розуміння того, чому люди їх використовують, можуть відкрити нові ідеї та канали для ефективнішої взаємодії з аудиторією.

Соціальні мережі для спілкування

Люди використовують ці мережі для зв'язку з людьми (та брендами) в Інтернеті. Вони також можуть принести користь бізнесу, а саме – дослідження ринку, поінформованість про бренд, генерація потенційних клієнтів, побудова відносин, обслуговування споживачів та інше, насправді список майже нескінченний.

Соціальні мережі, які іноді називають «мережами стосунків», допомагають людям та організаціям зв'язуватися в Інтернеті для обміну інформацією та ідеями.

Хоча ці мережі не є найдавнішим видом соціальних медіа, вони, безумовно, стали найпопулярнішими. Такі типи мереж починалися як відносно прості сервіси – наприклад, Twitter був місцем, щоб відповісти на питання «що ти робиш?».

Зараз, і особливо з часів зростання популярності та доступності мобільного Інтернету, ці мережі стали центрами, які перетворюють майже всі аспекти сучасного

життя у соціальний досвід, – від читання новин до обміну фотографіями з відпусток до пошуку нової роботи.

Facebook стартувала в лютому 2004 року. Гарвардські студенти, які підписалися на послугу, могли розміщувати свої фотографії та особисту інформацію про життя, наприклад, розклад занять та клуби, до яких вони належали. Його популярність зростає, і незабаром студентам з інших престижних шкіл, таких як Єльський та Стенфордський університети, було дозволено вступати. До червня 2004 року зареєструвалось понад 250000 учнів із 34 шкіл, і того ж року великі корпорації, такі як компанія кредитних карток MasterCard, почали платити за експозицію на сайті. Наразі ця мережа є найбільшою у всьому світі (рис. 1.1).

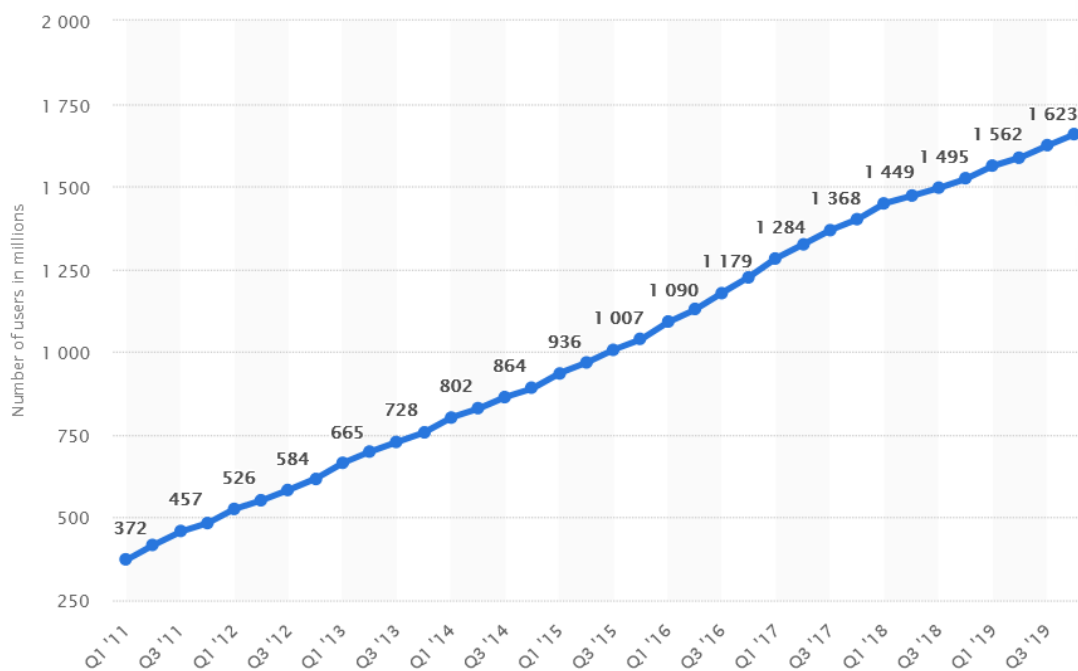


Рис. 1.1. Кількість відвідувачів Facebook

Мережі обміну медіа

Люди використовують ці мережі для пошуку та обміну фотографіями, відео, відео в реальному часі та іншими засобами масової інформації в мережі Інтернет.

Вони теж можуть принести користь бізнесу як і мережі для спілкування, ці сайти та додатки є безцінними для обізнаності про бренд, генерації потенційних клієнтів, залучення аудиторії та більшості інших соціальних маркетингових цілей.

Мережі обміну медіа дають людям і брендам місце для пошуку та обміну медіа в Інтернеті, включаючи фотографії, відео та відео в прямому ефірі.

Межі між мережами обміну медіа та мережами для спілкування стираються в наші дні, оскільки соціальні мережі, такі як Facebook та Twitter, додають на свої платформи можливість дивитись відео, доповнену реальність та інші мультимедійні послуги. Однак, що відрізняє мережі обміну медіа, це те, що обмін медіа є їх визначальною та основною метою.

Хоча більшість постів у мережах містять текст, публікації в таких мережах, як Instagram та Snapchat, починаються з зображення чи відео, до якого користувачі можуть вирішити додати такий такий вміст, як підписи, згадки інших користувачів чи фільтри, завдяки яким можна бути схожим на зайчика, котика, старика чи ще кого завгодно. Так само на таких сайтах, як YouTube та Vimeo, відео є основним способом спілкування.

Визначаючи, чи потрібно бізнесу встановлювати свою присутність у мережі обміну медіа, важливо враховувати наявні ресурси. Якщо є щось спільне у найбільш успішних брендів на таких платформах, як YouTube чи Instagram, це ретельно спланована місія та ретельно розроблені медіа-активи, які, як правило, дотримуються певної теми.

YouTube – це служба обміну відео, яка дозволяє користувачам переглядати відео, розміщені іншими користувачами, та завантажувати власні відео. Послуга була запущена як незалежний веб-сайт у 2005 році, а Google придбала її у 2006 році. Відео, завантажені на YouTube, можуть з'являтися на веб-сайті YouTube, а також розміщуватись на інших веб-сайтах, хоча файли розміщуються на сервері YouTube.

Гасло веб-сайту YouTube – «Broadcast Yourself». Це означає, що послуга YouTube призначена в основному для простих людей, які хочуть публікувати створені ними відео. Хоча кілька компаній та організацій також використовують YouTube для просування свого бізнесу, переважна більшість відео на YouTube створюються та завантажуються аматорами.

Кількість користувачів сервісу можна побачити на зображенні 1.2.

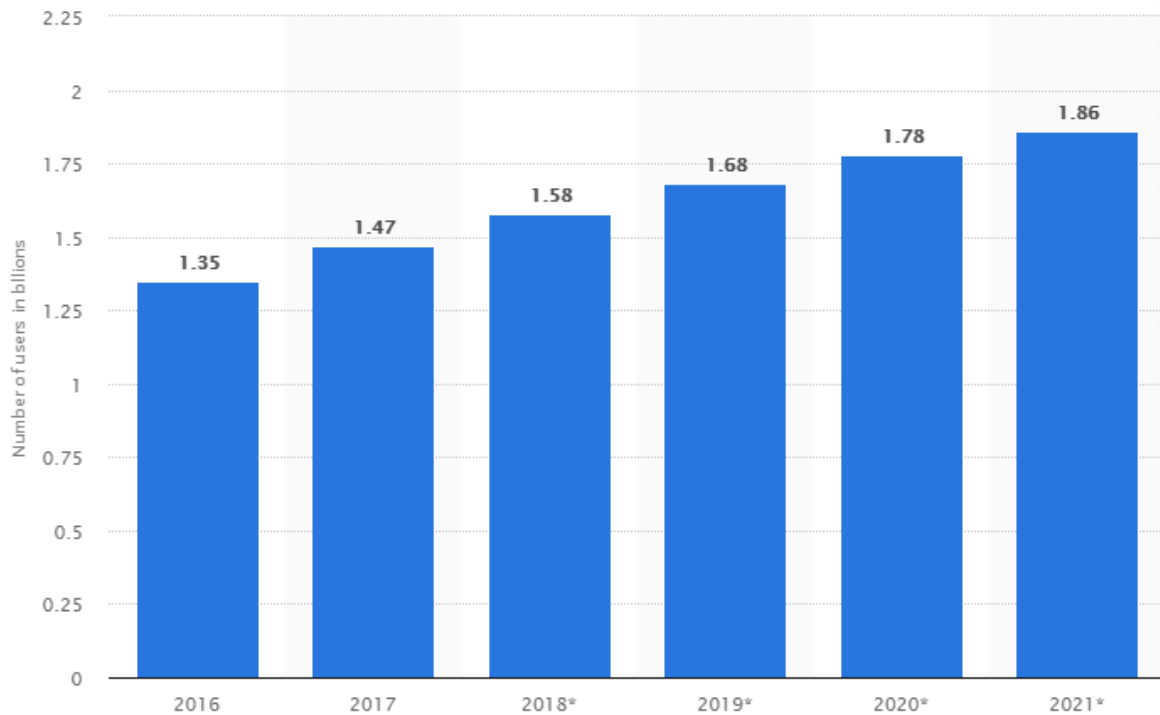


Рис. 1.2. Кількість користувачів YouTube

Дискусійні форуми

Люди використовують ці мережі для пошуку, обговорення та обміну новинами, інформацією та думками. Вони теж не стали вийнятком та можуть принести користь бізнесу ці мережі можуть стати чудовими ресурсами для дослідження ринку. Також можна рекламувати на них продукти, хоча робити це треба обережно.

Взагалі, дискусійні форуми - один із найдавніших типів соціальних медіа. Перш ніж зв'язатись із першими університетськими друзями у Facebook, люди обговорювали поп-культуру, поточні справи та просили допомоги на форумах. Можна подивитись на широке охоплення та величезну кількість користувачів на форумах, таких як reddit, Quora та Digg, і побачить, що спрага громадськості до колективних знань і мудрості залишається невгамовною.

Це ті мережі, куди люди приходять, щоб з'ясувати, про що всі говорять. І користувачі цих сайтів, як правило, не соромляться висловлювати свою думку. Хоча мережі для спілкування дедалі частіше впроваджують заходи щодо зменшення анонімності та створення безпечного простору в Інтернеті, дискусійні форуми, як

правило, дозволяють користувачам залишатися анонімними, зберігаючи відчуття «дикого західу», яке використовувалось для визначення онлайн-досвіду.

Це може зробити дискусійні форуми, такі як reddit та Quora, чудовими місцями для глибокого дослідження клієнтів та брутално чесних думок.

Сервіси соціальних закладок

Люди використовують ці мережі, щоб виявляти, зберігати, ділитися та обговорювати новий та популярний контент та медіа. Ці мережі також можуть бути надзвичайно ефективними для підвищення поінформованості про бренд, залучення клієнтів та відвідування веб-сайтів.

Цей тип мереж – це осередок творчості та натхнення для людей, які шукають інформацію та ідеї, і, додавши їх до свого маркетингового плану в соціальних мережах, відкриваються нові канали для підвищення поінформованості про бренд та взаємодії з аудиторією та клієнтами.

«Мережі соціальних закладок, такі як Flipboard, схожі на мережі закладок, але з акцентом на пошук та обмін статтями та іншим текстовим вмістом. Кожен може створити власний журнал Flipboard, щоб відсортувати найбільш цікавий вміст на обрану тему із сторонніх джерел та продемонструвати власний контент.»[1]

Інші типи мереж також додають функції закладки. Наприклад, Instagram зараз пропонує користувачам функції для збереження вмісту та створення приватних колекцій.

Мережі для відгуків

Цей тип мереж використовується для пошуку, перегляду та обміну інформацією про бренди, товари та послуги, а також про ресторани, туристичні напрямки тощо. Позитивні відгуки підтверджують статус ресторану, кафе, готелю та будь-якого іншого закладу. Це дає змогу для бізнесу виправити всі свої недостатки, просто подивившись на коментарі у цьому типі соціальних медіа.

Відгуки – це тип контенту, який додає великої цінності багатьом веб-сайтам та онлайн-сервісам, наприклад досвід покупки на Amazon або досвід пошуку місцевого бізнесу на Картах Google. Мережі для відгуків роблять ще один крок далі, будуючи велику мережу навколо коментарів, як основної частини цінності, яку вони надають.

Сервіси для відгуків на основі місцезнаходження, такі як Yelp та Zomato, продовжують зростати, оскільки особисті соціальні мережі приймають геолокацію, і все більше користувачів приходять до думки, що набагато простіше подивитись відгуки в додатку, щоб визначити куди сходити ввечері та отримати рекомендації щодо найкращих закладів харчування.

Існують сервіси для перегляду будь-чого – від готелів та ресторанів до бізнесу, де можна подати заявку на роботу – і відгуки користувачів мають більшу вагу, ніж будь-коли раніше. Насправді, згідно з опитуванням BrightLocal, 88 відсотків споживачів довіряють онлайн-відгукам стільки, скільки особистим рекомендаціям.

Блоги та видавничі мережі

Їх використовують для публікації, виявлення та коментування контенту в Інтернеті. Контент-маркетинг може стати високоефективним способом взаємодії з аудиторією, створення особистого бренду та залучення потенційних клієнтів та продажів.

Блоги та видавничі мережі дають людям та брендам інструменти для публікації контенту в Інтернеті у форматах, що заохочують до пошуку, обміну та коментування. Ці мережі варіюються від більш традиційних платформ для ведення блогів, таких як WordPress та Blogger, до послуг мікроблогів, таких як Tumblr, та інтерактивних платформ для соціальних видавництв, таких як Medium.

Соціальні торгові мережі

Ці мережі використовуються щоб виявляти тенденції, стежити за брендами, ділитися чудовими знахідками та робити покупки. Бренди можуть підвищити обізнаність, збільшити зацікавленість та продавати продукцію за новими каналами. Соціальні торгові мережі роблять електронну комерцію цікавою, додаючи соціальний елемент.

Звичайно, елементи електронної комерції з'являються в багатьох інших типах соціальних мереж – наприклад, у Pinterest є придатні для продажу шпильки, а Instagram надає інструменти із закликом до дії у вигляді кнопок «придбати зараз» та «встановити зараз». Мережі соціальних покупок роблять крок далі, будуючи свій

сайт навколо цілеспрямованої інтеграції між соціальним досвідом та досвідом покупок.

Такі послуги, як Etsy, дозволяють малому бізнесу та окремим майстрам продавати свою продукцію без існуючого цегляно-будівельного майданчика, а також таких мереж, як сукупна продукція Polyvore від різних роздрібних торговців на єдиному інтернет-ринку. Polyvore – одна з найбільших спільнот соціального стилю в Інтернеті, і це хороший приклад мережі, призначеної для інтеграції соціального досвіду з досвідом покупки. Більшість вмісту генерується користувачами, які вибирають товари, які їм подобаються, створюють колажі, публікують їх як набір, а потім діляться наборами з іншими користувачами.

Мережі на основі інтересів

Використовуються для спілкування з іншими навколо спільного інтересу чи хобі. Якщо існує мережа, присвячена видам продуктів або послуг, які надає бізнес, ці мережі можуть стати чудовим місцем для взаємодії з аудиторією та підвищення рівня обізнаності про бренд.

Мережі, що базуються на інтересах, використовують більш цілеспрямований підхід, ніж великі соціальні мережі, зосереджуючись лише на одній темі, такі як книги, музика або дизайн будинку.

Хоча існують групи та форуми в інших мережах, присвячених цим інтересам, орієнтація виключно на одну сферу інтересів дозволяє цим мережам надати досвід, спеціально розроблений для потреб людей та громад, які розділяють ці інтереси. Наприклад, на Houzz дизайнери будинків можуть переглядати роботи інших дизайнерів, створювати колекції власних робіт та спілкуватися з людьми, які шукають їх послуги.

Такі мережі, як Last.fm (для музикантів та любителів музики) та Goodreads (для авторів та завзятих читачів), також забезпечують досвід, розроблений спеціально для їхньої нішової аудиторії.

Анонімні соціальні мережі

Люди користуються таким типом соціальних мереж, щоб пліткувати, пускати слухи, підглядати та інколи залякувати.

Останні і водночас найменші – це анонімні соціальні мережі. Хоча основні соціальні мережі докладають все більших зусиль, щоб притягти користувачів до відповідальності за їх соціальну активність, ці сайти йдуть іншим шляхом і дозволяють користувачам розміщувати контент анонімно. Ці мережі можуть здатися цікавим місцем для здуття пари (наприклад, якщо ви підліток і хочете скаржитися на своїх батьків, вчителів, хлопця тощо). Однак було показано, що вони забезпечують форум для кібер-залякування без наслідків і пов'язані із самогубствами підлітків.

1.1.2. Месенджери

Соціальні мережі вже давно перестали бути середовищем для спілкування, сьогодні їх все частіше стали використовувати як рекламний майданчик. Месенджер - це така собі домашня, вітальня з каміном, де ти можеш зібрати найближчих друзів, адже не знаючи никнейма або номера телефону, стороннім вторгнутися не вдасться. Сьогодні, коли LTE-з'єднання вже постукує в парадні двері, ніхто не замислюється про економію трафіку, тому месенджери нерідко стали використовувати для дзвінків в інші країни (забувши про роумінг), для внутрішньокорпоративного спілкування і обміну файлами.

На сьогоднішній день месенджери дають нам куди більше, ніж просто обмін повідомленнями. Як то кажуть: «Попит народжує пропозицію!», І сервіси стали плодитися як гриби, але ми то з Вами з легкістю можемо виділити серед них своїх улюбленців. Згідно відгуками користувачів порталу coba.tools, пропонуємо до Вашої уваги наш ТОП найпопулярніших месенджерів.

Telegram – безкоштовний месенджер для обміну текстовими, аудіо та відеофайлами. Завантаження додатку доступне на базі операційних систем Android та iOS, також є десктоп версія. Ключовою особливістю програми є безпека даних, яка реалізована за допомогою шифрування на основі криптографічного протоколу MTProto.

Програма повністю синхронізується на різних пристроях, винятками є лише секретні чати, які видно тільки на пристрої, на якому створені. Всі медіа вкладення з діалогів зберігаються в хмарі. У сервісі можна знайти приватні та публічні канали,

як правило, це чати для масових розсилок. На таких каналах немає ні коментарів, ні лайків, можна тільки переглядати стрічку новин. Месенджер Telegram з 2014 року по 2020, набрав 400 мільйонів активних користувачів (рис. 1.3).

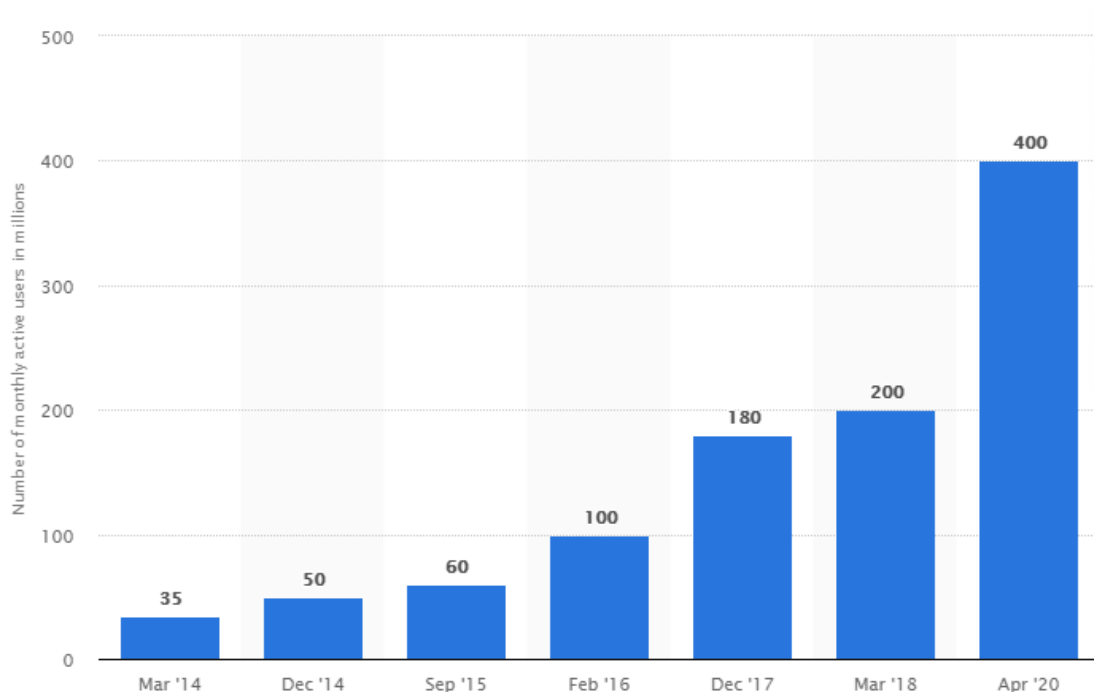


Рис. 1.3. Кількість відвідувачів Telegram

Facebook Messenger – чат для спілкування всередині соціальної мережі Facebook без постійної авторизації. Програма безкоштовна і автоматично синхронізує список контактів з персональної сторінки Facebook. Користувач отримує повідомлення про коментарі, зустрічі та оповіщення від друзів. Має інтуїтивно зрозумілий інтерфейс.

Усередині сервісу реалізовано як текстове повідомлення, так і відео/аудіо дзвінки, є паки стікерів і Еможі, можливий обмін фото та відео матеріалами. Сервіс передбачає письмове спілкування, а також можливості прикріплення стікерів і Еможі, обміну фотографіями та відео, проведення відеодзвінків і створення групових розмов. Також є опція секретного чату, з можливістю встановити певний час зберігання повідомлень, через яке всі повідомлення зникнуть. Про всі нові повідомлення приходять push-повідомлення. Можна здійснювати електронні

платежі, правда ця функція доступна тільки за умови наявності картки американського банку.

WhatsApp – програма, яка працює на всіх мобільних платформах. У програмі є можливість надсилати голосові повідомлення, та, звичайно, текстові, також файли, є можливість спілкуватись по відео. Програма самостійно аналізує телефонну книгу і позначає користувачів, які вже працюють в середовищі WhatsApp. Можливо синхронізувати мобільний додаток з десктопною версією, є опція створення групового чату до 256 чоловік. Є доступ до повідомлень, які були доставлені в режимі офлайн.

Viber – додаток для спілкування, працює на смартфонах, і на комп'ютерах під управлінням Windows, OS X і Linux. Кількість користувачів можна подивитись на зображенні 1.4.

Підтримує функцію відправки повідомлень, обміну файлами, також є можливість здійснення дзвінків, як аудіо, так і відео. Доступна синхронізація одного облікового запису на різних пристроях. Можна створювати відкриті чати до 100 учасників для листування в реальному часі, це буде відмінним рішенням для корпоративного спілкування. Підтримує інтеграцію зі списком контактів, обмін зображеннями, аудіо та відео файлами.

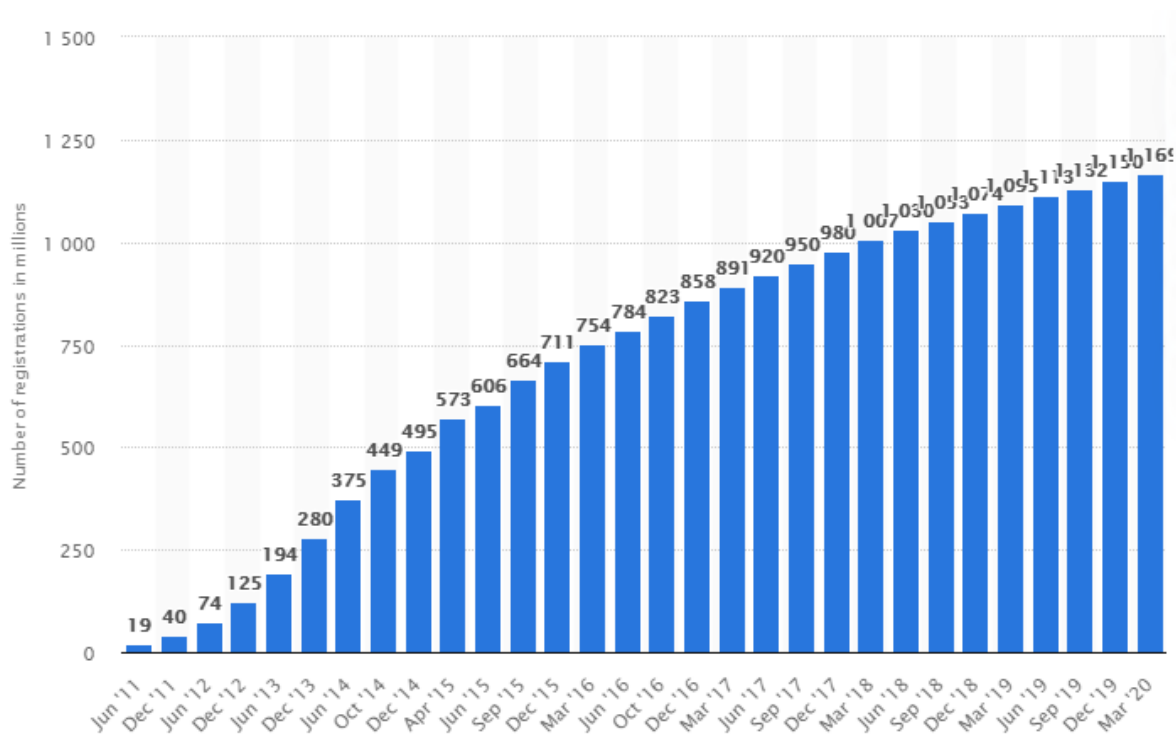


Рис. 1.4. Кількість користувачів Viber

Hangouts – додаток від компанії Google, який був створений для зв'язку за допомогою аудіо та відео. Є функція архівування інформації, всі фото з чатів завантажуються в онлайн альбоми, листування зберігаються на серверах, до них можна отримати доступ з будь-якого пристрою. Сервіс використовується для конфіденційного листування, обміну інформацією та файлами, організації корпоративного спілкування і відеопереговорів. Програма показує, чи прочитано ваше повідомлення. Є синхронізація з Google+ і Gmail, можна додати Google Voice, смайли Емої, стікери та анімацію, картинки і положення геолокації. Можна організувати чати до 100 чоловік.

iMessage – служба для безкоштовного зв'язку за допомогою текстових повідомлень, яка доступна на пристроях Apple. За замовчуванням вбудований в кожен iPhone і Mac. Додатки вкрай вибірково підходить до вибору способу транспортування повідомлень. Якщо повідомлення складається з тексту та вкладення, то текст розумніше відправити мобільного зв'язку, а вкладення через Wi-Fi, такі питання програма вирішує сама, це пов'язано з постійною гонитвою за безпекою пересилаються призначених для користувача даних. Компанія Apple створила платформу для передачі повідомлень, яка складається з складних

складових елементів, і пов'язані вони нетривіальною логікою роботи. Хоча варто відзначити, що складність реалізації ніяк не позначилася на самому продукті. Має мінімум налаштувань, в використанні він простий і зручний.

1.1.3. Ігри

Компанія Newzoo, яка займається дослідженням ринку, в останньому звіті про ігрової індустрії заявила, що в 2020 році 2,4 мільярда користувачів смартфонів грали в мобільні ігри різних жанрів. У 2020-2021 роках, у зв'язку з ізоляцією, викликані спалахом COVID-19, люди шукають нові способи розваг. Кількість активних користувачів ігор на смартфонах зростає.

Виникає питання, які жанри мобільних ігор найпопулярніші? Так, всі чули про Angry Birds, Temple Run і Subway Surf, але є ще Arena of Valor, яка шалено популярна в Китаї, і інші ігри. Теперішнім і майбутнім розробникам ігор важливо передбачати шлях руху.

Найпопулярніші жанри мобільних ігор

ТОП-5 жанрів мобільних ігор розподілені між:

- Battle Royale;
- MOBA;
- MMORPG;
- картковими іграми;
- головоломками.

Цей список складений з урахуванням кількості людей, які завантажили гру і скільки грошей витрачено в іграх кожного жанру. Крім них є ще пара популярних жанрів мобільних ігор, які варто згадати, хоча вони і не наблизилися до вищеназваних варіантів з точки зору кількості гравців або отримання прибутку:

- FPS;
- спорт;
- RPG;
- гонки.

Деякі Battle Royale для мобільних телефонів - це технічно шутери від першої особи. Але є різниця між BR і FPS жанром, навіть якщо ігри використовують схожу механіку.

Популярні жанри мобільних ігор у 2021 році:

Дослідження ринку показує, що ще в 2020 році, ігрові програми були на третьому місці, їх обігнали соціальні медіа та магазини. Подібна тенденція буде зберігатися і надалі. Користувачі смартфонів у віці від 18 до 20 років, грають у багату кількість ігор, щоб переконатися в цьому треба подивитися на популярність ігри PUBG Mobile та її брата – Fortnite.

Minecraft, Pokémon Go, Fortnite, PUBG Mobile, Арена Valor, Temple Run і Flappy Bird вважаються найпопулярнішими мобільними іграми. Китай утримує позиції найбільшого ринку ігрових додатків, фокусуючись при цьому на MMORPG і гіпер-казуальних.

Ігри Hyper Casual не вимагають майже ніяких дій від користувача, а спрощена ігрова механіка тільки в плюс. Вони зосереджені на мінімалістичних але чітких зображеннях, мультяшному художньому стилі і рудиментарних системах прогресування. Різниця між цими та казуальними іграми в кількості покупок, зроблених через додаток. Casual розраховують отримати завдяки покупкам через додаток $\frac{3}{4}$ доходу, а Hyper Casual заробляють за рахунок використання відео, банерної та проміжної реклами.

Multiplayer Battle Arenas

МОВА є сумішшю RTS і екшна. У багатокористувацьких бойових аренах гравець контролює одного персонажа – "героя" або "чемпіона" з доступного реєстру, підібравши унікальний набір умінь і навичок. Це може бути маг, воїн, лицар чи представник іншої "професії". Мобільний варіант жанру відрізняється від комп'ютерного (LoL чи DOTA2), адже він не відображає таку ж глибину механіки і стратегії. Мобільні пристрої обмежені в потужності обробки даних, розмірі дисплея і параметрах введення.

Найпопулярнішими МОВА для смартфонів і планшетів стали Mobile Legends (рис. 1.5), Arena of Valor та Vainglory. Ці три гри так чи інакше копіюють шалено

популярну гру League of Legends для ПК. LoL, яка в свою чергу, скопійована з Warcraft III. Жанр можна простежити і далі, до Aeon of Strife.

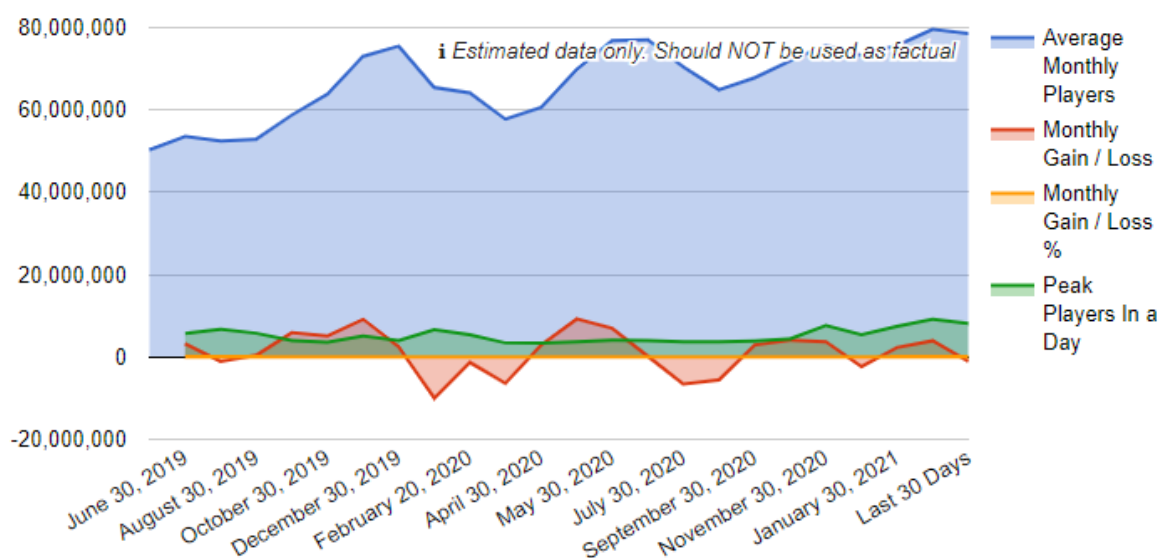


Рис. 1.5. Кількість користувачів Mobile Legends

Команди з 3 або 5 гравців протистоять один одному в боротьбі за ресурси, об'єкти і контроль карти. Гравцеві потрібно мати швидкі пальці, жвавий розум і багато терпіння, щоб грати в МОБА.

До 2016 року цей жанр бив рекорди популярності, але з'явилися Battle Royale і забрали частину слави собі. Зараз МОБА вже не так популярні, як раніше, особливо серед мейнстрімової аудиторії і любителів казуальних ігор. Але вони ще впевнено тримають позиції серед популярних жанрів мобільних ігор і не залишать їх в 2021.

Більшість розробників ігор жанру МОБА не ведуть підрахунок кількості гравців, принаймні публічно. Але навіть після появи жанру королівська битва, в 2017 році, гра Arena of Valor налічувала 200 мільйонів унікальних гравців, і 80 мільйонів заходили в гру щодня. Тобто вона була найбільш популярною грою в усьому світі. Її створила китайська технічна компанія Tencent, якій належить більша частина акцій в ігрових компаніях, серед яких і RIOT, творець LoL.

Battle Royale

В кінці 2019, PUBG mobile - творіння китайського Tencent, побила рекорди, зібравши 400 мільйонів гравців (рис. 1.6), із середнім денним значенням 50 мільйонів. Хороший показник того, наскільки поширеним став жанр мобільних ігор

Battle Royale. Це найпопулярніший жанр 2020 року, але серед 50 мільйонів є гравці, які використовують не тільки мобільні телефони, але і ПК з консолями. Сам жанр отримав популярність в 2017 році, коли з'явилася гра Player Unknowns Battlegrounds для ПК.

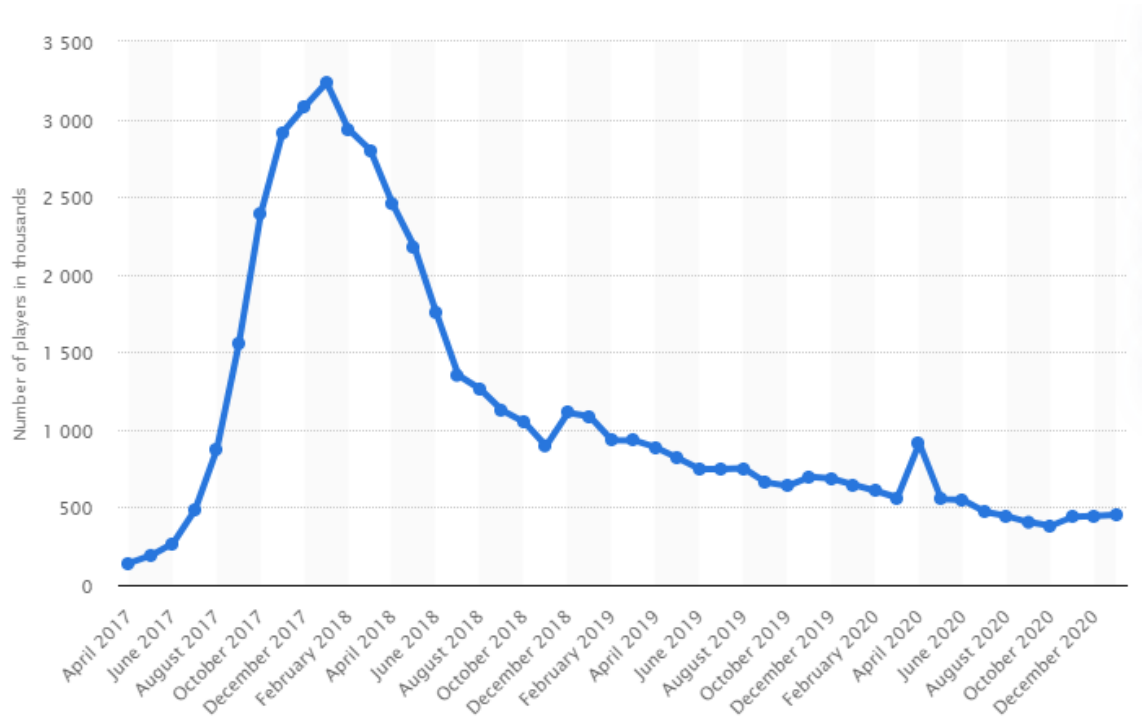


Рис. 1.6. Кількість користувачів PUBG

Суть жанру Battle Royale в тому, що користувач бореться проти 99% гравців на величезній карті з ресурсами, серед яких:

- зброя;
- обладунки;
- боєприпаси.

Ресурси розкидані в різних місцях, а гравець повинен зібрати їх, щоб відбитися від ворогів. Головна мета мобільної гри королівська битва - залишитися в живих. Ідея частково взята з антиутопического японського фільму "Королівська битва", що вийшов у 2000 році (в Україні більш відома інша антиутопія зі схожим сюжетом – «Голодні ігри»).

Гра здається простою, але гігантські карти, велика кількість комбінацій зброї і броні помітно виділяють жанр, особливо, якщо порівняти з традиційними FPS,

наприклад Battlefield або Call of Duty. Гравцеві доступний величезний ігровий майданчик з безмежними тактичними можливостями. Залишатися на вершині можна лише правильно розподіляючи ресурси.

Популярної мобільної грою в жанрі королівська битва є Fortnite. Творцям цієї гри вдалося зацікавити користувачів не тільки художнім стилем і чіткими зображеннями, а й системою, яка дозволяє створювати вежі, бар'єри, оборонні стіни і інші об'єкти прямо на полі битви.

Відомі стримери на Twitch транслюють принаймні одну гру жанру battle royale, що залучає нових гравців. Переважно, це діти і підлітки 8-14 років, хоча і дорослі грають у Fortnite. Ігри жанру абсолютно безкоштовні, а їх творці заробляють за допомогою реклами у грі, як і в МОБА.

MMORPG

Один з найстаріших ігрових жанрів. MMORPG розшифровується приблизно як масова розрахована на багато користувачів ролева гра онлайн. Загалом, це рольова онлайн-гра, в якій гравець прокачує героя і набуває нові сили в боях з неігровими персонажами або іншими реальними гравцями. Згодом MMORPG еволюціонували, визначення змінилося, а сам жанр ожив і більше не обмежується грою «Warcraft».

У MMORPG є величезні світи, які розділені на зони, що робить їх схожими на інші фентезійні світи, як у Володарі кілець або Грі Престолів. Тут є:

- підземелля;
- дракони;
- лицарі;
- демони;
- боги;
- страшні чудовиська.

Кожну хвилину в такі мобільні ігри грають тисячі гравців, які вбивають босів, збирають різні речі, виконують інші квести. У МОБА користувачі поділяються на

команди по п'ять чоловік, в Battle Royale грають до 100 чоловік одночасно, тоді як MMORPG кардинально відрізняється, адже тут ніхто нікому не суперник.

Жанр ігор MMORPG для мобільних пристроїв користувався меншою популярністю, ніж для ПК, адже не кожен телефон міг потягнути такі ігри, однак з розвитком смартфонів, він повертається. Серед найпопулярніших мобільних MMORPG: Adventure Quest, Arcane Legends та Dungeon Hunter 5.

Цифрові карткові ігри

Також відомі як TCG або ігри для обміну картками. Гравцеві доводиться битися своєю колодою проти колоди противника. Карти розділені за рівнями сили. Заробляють карти вищого рівня під час гри або витративши реальні гроші, щоб відкрити нову колоду.

Найпопулярнішою представницею цього жанру є гра Hearthstone, яка зібрала близько 100 мільйонів гравців на платформах ПК, iOS і Android (рис. 1.7). Така кількість користувачів була на початку 2019 року, зараз же не можна сказати про майбутнє жанру цифрових карткових ігор. Однак поки він в ТОП-5, хоча і не поширюється як ігри Battle Royale.

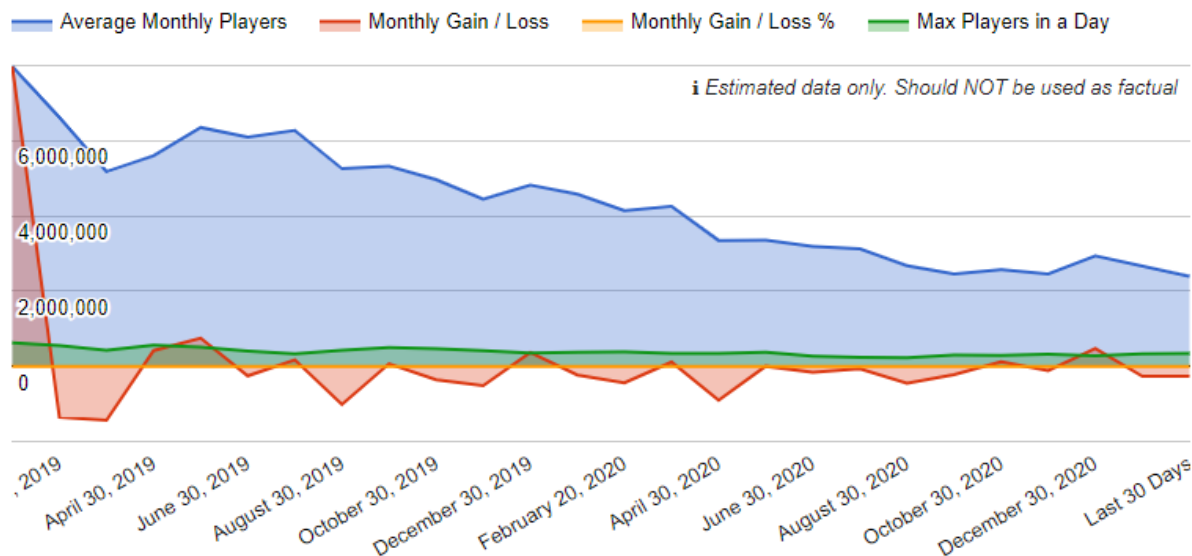


Рис. 1.7. Кількість користувачів гри Heartstone

Серед популярних ігор цього жанру для мобільних пристроїв виділимо:

- HEX: Shards of Fate;

- Eternal;
- Shadowverse;
- Heartstone;
- Elder Scrolls: Legends.

Головною перевагою цифрових карткових ігор є їх тривалість, адже навіть матч МОВА може затягнутися на півгодини. Тут же гравець сам вибирає карти, складає їх в бажаному порядку, а гра зробить інше. Процес використовує багато механіки, яка заснована на удачі, що підтримує бажання геймерів продовжувати грати, маючи їх азарт.

Ігри цього жанру абсолютно безкоштовні. Однак геймери часто доплачують за преміальну колоду або швидке розблокування нової, але це тільки опція, яку не обов'язково вибирати. Можливість прокладати власний шлях наверх, ставши майстром колоди, ніхто не забрав.

Головоломки

Більше п'ятидесяти відсотків користувачів смартфонів мають на своїх пристроях хоча б одну головоломку, особливо це стосується жінок. Тетріс, Angry Birds, варіації на тему "три в ряд" - ці мобільні ігри стоять навіть на смартфонах користувачів, які не вважають себе геймерами. Люди з різних вікових і соціальних груп люблять головоломки. Мета головоломок зрозуміла, але при цьому досягти її не так легко, доводиться задіяти логіку і кмітливість, щоб перемогти. До того ж цей жанр мобільних ігор не так травмує пальці як шутери або гонки, адже граючи у головоломки, треба просто періодично водити по екрану.

Головоломки вважаються піджанром казуальних ігор, охоплюють широку аудиторію, тримають міцні позиції і приносять найбільшу кількість доходів на ринку мобільних ігор. Найпопулярнішими головоломками вважаються:

- Roll the ball;
- Rusty Lake;
- The Room;
- Love Poly;

- Cut the Rope.

Західна аудиторія віддає перевагу жанру Hyper casual, Екшн та головоломки. Тоді як азіатські геймери більше люблять Battle Royale і MMORPG.

Окрім вищеназваних найпопулярніших жанрів також варто згадати:

- Sims. У цих симуляторах створюється власний віртуальний світ. Спочатку існувала тільки Sim City, але жанр знайшов величезну популярність завдяки різноманітним фермам, які з'явилися 10 років тому і при першій нагоді перекочували в смартфони.
- Бої. Класика в стилі Street Fighter, Mortal Kombat або Tekken не втрачає позицій. Цей жанр ігор перейшов на мобільні пристрої і ними можуть насолоджуватися також і мобільні геймери.
- Казино. Ігрова індустрія для мобільних телефонів зростає, а разом зростає популярність онлайн казино. Блекджек, покер, рулетка, слоти, двадцять один, - у все це можна грати, не залишаючи будинок.
- Asphalt 9: Legends, Real Racing, F1 Mobile Racing, Rush Rally і безліч інших гонок вразять геймерів реалістичністю і дизайном.
- Спортивні. Спортивні ігри надовго окупували консолі, але кілька років тому навіть John Madden і NBA2K розробили для мобільних платформ і вони відразу підкорили ринок.

У 2021-2022 багато видавців AAA ігор планують запуснути ігри для смартфонів, засновані на франшизах для ПК. Ігрові компанії та інвестори серйозно ставляться до мобільних ігор, що видно за якістю графіки, яка постійно зростає і місцями навіть наближається до консольного рівня деталізації.

1.1.4. Додатки для бізнесу

Мобільні додатки для бізнесу можна розділити на два види:

- для внутрішніх потреб компанії;
- для збільшення продажів, здійснення брендингу та маркетингу.

Цілі використання першого виду додатків:

- Автоматизація бізнесу;

- Підвищення продуктивності спільної роботи, наприклад, створення спільного доступу до файлів, підключення месенджерів соціальних мереж і трекерів для внутрішньої комунікації, а також систем управління проектами та завданнями.

Цілі використання другого виду програми:

- Як доповнення до онлайн-сервісів компанії: вітрини, каталоги, програми для покупки квитків, інтернет-банкінг, трекер статусу доставки товарів і так далі;
- Створення мобільної програми лояльності.

Є певні нюанси використання таких типів додатків. Додатки для автоматизації бізнесу зазвичай встановлюють в обов'язковому порядку всі учасники ділової команди. Але додатки для клієнтів компанії встановлюються за бажанням клієнтів, яких треба переконати, що установка додатку важлива, зручна, проста, вирішує якісь завдання.

Треба бути готовим до того, що чимала частина користувачів все таки не захоче встановлювати додаток, якщо ту ж інформацію можна отримати і через мобільний браузер. Від установки додатка відмовляються з кількох причин: клопітно (треба шукати в онлайн-магазині, завантажувати, надавати якісь дозволи), довго, займає місце в пам'яті пристрою і уповільнює його роботу. Все це призвело до того, що популярність мобільних додатків поступово почала згасати: якщо ще п'ять років тому тільки близько 30% європейців встановлювали додаток раз в місяць, а сьогодні більше 60% користувачів у світі встановлює додатки рідше, ніж раз на місяць.

Todoist – мабуть, один з найбільш зручних додатків для складання todo-списків, він дозволяє сортувати завдання і встановлювати нагадування. Існує як у вигляді додатку, так і у вигляді онлайн-сервісу. На жаль, велика частина корисних функцій доступна тільки тим, хто встановив платну версію (рис. 1.8).

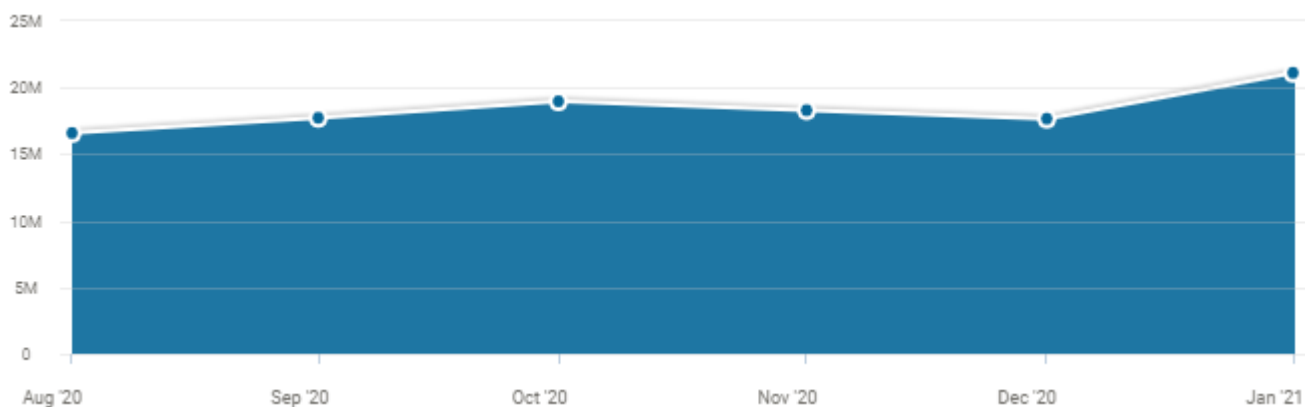


Рис. 1.8. Кількість користувачів Todoist

Google Drive – це сервіс, який існує у так званій «хмарі», він дозволяє в режимі реального часу працювати над документами, таблицями та презентаціями.

Basecamp – безкоштовний сервіс, розроблений для розподілу завдань між співробітниками і відстеження їх продуктивності. Крім того, Basecamp – не тільки зручний додаток для управління робочими процесами, а й засіб зв'язку між користувачами.

Chrome Remote Desktop – зручний додаток для тих, хто часто працює поза офісом. Для використання додатку, треба встановити так зване «розширення» у браузері від Google та сам додаток Chrome Remote Desktop, після чого користувач отримає доступ до комп'ютера віддалено і може користуватися ним з екрану смартфона.

Slack – сервіс для групових чатів, головною перевагою якого є можливість обмінюватися файлами будь-якого формату. Тут же можна створювати різні канали для об'єднання співробітників одного відділу або людей, які працюють над одним проектом.

1.1.5. Додатки для заняття фітнесом

Майже кожному потрібна невелика допомога, коли мова заходить про те, щоб привести себе в форму. Фітнес-додатки нагадують вам про ваші цілі і спонукають вас дотримуватися їх. Вони також надають необхідні інструменти та інформацію,

наприклад калькулятор калорій або тренер, який планує ваші тренування і нагадує про них.

Ці програми відповідають цілому ряду вимог. Деякі з них відмінно підходять для людей, які вже давно займаються спортом, в той час як інші ідеально підходять для початківців. Одні фокусуються на певних областях, наприклад, харчування або музика для тренувань, в той час як інші використовують підхід «все в одному».

Деякі з додатків прості, а інші навпаки видають користувачеві величезну кількість даних. Багато які можуть працювати з моніторами серцевого ритму і фітнес-трекерами, щоб допомогти користувачу зібрати більше даних про тренування.

Johnson & Johnson Official 7 Minute Workout – унікальний додаток для швидких тренувань. Все, що потрібно користувачу – це стілець і близько семи хвилин. Складність вправ можна вибрати виходячи з рівня підготовки. Інтерфейс на подив привабливий і інтуїтивно зрозумілий. Аудіо- та візуальні сигнали говорять, коли починати і зупиняти кожну вправу в циклі, а в середині з'являється відео-демонстрація занять. Додаток J&J Official 7 Minute Workout – це один з найкращих існуючих фітнес-додатків, тому що підходить практично для всіх користувачів незалежно від їх фізичної форми.

Map My Fitness – це, мабуть, найкращий додаток для відстежування фізичних вправ для новачків або для тих, хто тільки починає свою довгу подорож до здорового способу життя. Програма записує будь-яку активність, від вигулу собаки до прибирання. Кожен раз, коли користувач збирається виконати будь-яку дію, він запускає додаток, обирає заняття і починає записувати, скільки часу він на нього витрачає, скільки калорій спалює, а також інші статистичні дані, які змінюються в залежності від того, що робить користувач. Додаток є безкоштовним, оновлення до Premium додає додаткові дані, такі як частота серцевих скорочень. Якщо користувач забув або не хоче відстежувати вправу в режимі реального часу, він може записати його в журнал вручну. Якщо використовується фітнес-трекер, то можна відправити туди свої дані. Map My Fitness також працює з розумними часами AppleWatch, AndroidWear, пристроями Garmin, Samsung Gear і іншими пристроями.

1.1.6. Додатки для навігації

Мобільні додатки зробили самостійні подорожі максимально доступними: всі можливості невеликого туристичного агентства тепер легко поміщаються в кишені. GPS і карти, калькулятор валют і путівник по місту, словник і зручний пошук готелю – грамотний набір додатків допомагає планувати подорожі і економити час, гроші і нерви, поки сидиш в кафе, їдеш в таксі або валяєшся в ліжку.

Scyscanner – додаток, який миттєво знаходить дешеві авіаквитки, готелі та машини напрокат. Якщо у вас гнучкі плани, виберіть цілий місяць замість конкретних дат і отримаєте графік цін, на якому видно, в який день переліт коштує дешевше. Для натхнення натисніть «Шукати скрізь» і дізнайтеся, куди найвигідніше злітати з вашого міста. В один клік подивіться кращі пропозиції на вихідні або по популярних напрямках. Коли виберете квитки, Скайсканер перенаправить вас на сторінку покупки і не візьме за це ніякої комісії. Заодно підшукайте готель і машину в оренду там, куди летите, – всі ці функції зібрані в одному додатку.

Буває, що з точки А в точку В доводиться добиратися через С і D. Для цих випадків стане в нагоді **Rome2Rio** – додаток, яке прокладає складні маршрути, в тому числі між країнами. Він підкине ідеї, на яких автобусах їхати з Анталії в Тбілісі і скільки поїздів змінити, щоб без перельоту дістатися з Москви до Улан-Батора. По кожному сегменту вказані приблизна ціна і сайт, на якому можна купити квитки.

Google Maps – це не тільки навігатор і путівник. Введіть маршрут, клікніть на чоловічка, що махає, і додаток покаже, у скільки обійдеться поїздка на Uber, Gett, Taxify та інших службах таксі. Щоб забронювати, натисніть «Відкрити в додатку», і Гугл перенаправить вас на потрібну сторінку. Врахуйте: як і інші агрегатори (наприклад, американський Bellhop або російський СравниТаксі), Гугл може не показувати всіх можливих опцій.

Байкшерінг – це коли ви через додаток знаходите прокатний велосипед, розблокуєте його на парковці, катаєтеся, потім залишаєте на спеціальній станції або просто на вулиці (залежить від правил компанії). Оператори велошерінга – різні в різних містах: наприклад, в Москві їздять на «Велобайк», в Лондоні – на червоних

«Борис-байках» з логотипом банку Santander. Світові лідери – китайські Mobike і Ofo, які вийшли за межі Піднебесної і тепер обґрунтовуються по всьому світу від Сіднея до Берліна і Флоренції. Який би велосипед ви не орендували в подорож, ви точно побачите місто зовсім з іншого боку.

Citymapper – навігатор по транспорту. Задайте початкову і кінцеву точки маршруту, і Citymapper розповість, як туди дістатися - в тому числі з пересадками, пішки і на велосипеді. За всіма видами транспорту додаток пише ціни, а також відстежує, коли прийде автобус. Сидіти на зупинці веселіше, коли знаєш, що залишилося 3 хвилини. Але є і недолік – додаток працює тільки в великих містах.

1.1.7. Розвиваючі додатки

У додатку магазину від Apple, що називається AppStore наразі є понад 400 тисяч різних додатків, серед яких кожен користувач iPad/iPhone знайде для себе щось цікаве, навіть якщо цього користувачеві всього кілька років.

«Будь-який власник iPad у якого є діти прекрасно знає, що поява в сім'ї планшетника відразу ж привертає увагу дитини і в результаті він стає найулюбленішою електронною іграшкою малюка. Але iPad може бути не тільки високотехнологічної іграшкою, а й досить непоганим освітнім інструментом для дитини який тільки починає вивчати світ. Ми розглянемо найбільш корисні дитячі розвиваючі програми для iOS, які допоможуть у навчанні вашої дитини.»[2]F

Більшість програм для дітей написані англійською мовою, що досить непогано якщо ви з самого дитинства хочете щоб ваша дитина вивчала іноземні мови, але також існує і велика кількість додатків на українській мові.

Сказбука – універсальний додаток, який навчить малюка читанню, математиці та англійської мови, коли дитина втомиться можна подивитися цікавий і розвиваючий мультимедіа. Інтерфейс виконаний в спокійних тонах і не напружує зір дитини. Обмежень за віком немає, оптимальний вік від 2-6 років. У додатку завдання розділені по віковій шкалі, ігри ускладнюються в міру дорослішання дитини. Розробники додали цікаву функцію - ліміт ігрового часу, з яким дитина не зможе зловживати гаджетом. Краща гра для дітей на думку Indie Prize в 2018 році.

Розвиток мовлення – розвиток логічного мислення та мовлення. Оптимальний вік користувача 4-7 років. Додаток побудовано із завдань на основі картинок з переплутаною послідовністю. Сюжет не складний. Завдання для дитини – відновити послідовність, описати що відбувається на кожному малюнку і об'єднати в загальну розповідь. Гра змусить логічно міркувати, дитина буде поповнювати словниковий запас. Додаток познайомить малюка з правильністю побудови тексту (початком, розвитком історії і кінцівкою). Батьки можуть змінювати складність завдання, в міру можливостей дитини.

Peak – це набір міні-ігор, які розділені по групах. Є вправи для тренування пам'яті, швидкості мислення, уважності. Окремо варто відзначити головоломки на знання мови: додаток не русифікований, щоб збирати слова або шукати їх на величезному полі букв, доведеться закликати на допомогу пізнання в англійській мові. Так що можна вбити відразу двох зайців: і мозок тренувати, і мову підтягувати. Після установки програми доведеться пройти тест. За його результатами Peak визначить базовий рівень навичок користувача і запропонує головоломки відповідної складності. Повна версія програми платна, причому для власників пристроїв від Apple вона обійдеться в 7 разів дорожче, ніж для користувачів Android. Але і в безкоштовній версії міні-ігор досить багато, вистачить надовго.

Brain Wars – ця програма містить вправи на розвиток пам'яті, швидкості реакції, точності, математичних навичок, уважності. Наприклад, програма покаже вам стрілку на червоному або синьому тлі. Ваше завдання - пальцем на екрані провести лінію у напрямку стрілки, якщо фон синій, і в іншу сторону, якщо червоний. Але в Brain Wars користувач не просто вирішує головоломки, він бере участь в змаганні. Алгоритм вибере з користувачів додатка противника, рівного по вмінням. Користувачі одночасно виконують три завдання і з'ясують, хто краще. Щоб успішніше виступати в битві, можна потренувати той чи інший навик окремо, але в безкоштовній версії програми кількість спроб обмежена.

1.1.8. Утиліти

Утиліта – це програмне забезпечення, розроблене для аналізу, налаштування, оптимізації чи обслуговування комп'ютера. Воно використовується для підтримки комп'ютерної інфраструктури – на відміну від прикладного програмного забезпечення, яке спрямоване на безпосереднє виконання завдань, що приносять користь звичайним користувачам. Однак програми-утиліти часто є частиною операційних систем. Наприклад, Batch Job може запускати написаний користувачем код для оновлення бази даних, а потім запускати утиліту для резервного копіювання бази даних, також, може запускати програму для стиснення диска перед копіюванням файлів.

Незважаючи на те, що основний набір службових програм зазвичай розповсюджується разом з операційною системою, і утиліта часто вважається частиною операційної системи, користувачі часто встановлюють заміни або додаткові утиліти. Ці утиліти можуть надавати додаткові можливості для виконання завдань, які виходять за рамки можливостей операційної системи.

Багато утиліт, які можуть вплинути на всю комп'ютерну систему, вимагають від користувача підвищених привілеїв, тоді як інші, які працюють лише на даних користувача, цього не роблять.

PureVPN – один з кращих VPN-сервісів, що дозволяє отримати доступ до всіх заблокованих у регіоні ресурсів і забезпечити збереження своїх персональних даних від витіку при використанні громадських і незахищених мереж. Особливістю PureVPN є досить велика географія розташування серверів: в арсеналі сервісу понад 450 серверів, що забезпечують доступ до близько 80 000 IP-адрес в більш ніж 89 країнах. Крім того, PureVPN надає ряд інших функцій і доповнень для максимальної зручності при роботі онлайн. Все це дозволить користувачу обійти механізми контролю і блокування даних в будь-якій країні, де б він не знаходився.

TeamViewer – запатентоване програмне забезпечення для віддаленого доступу, а також для дистанційного керування і обслуговування комп'ютерів та інших пристроїв, яке було вперше випущено в 2005 році. Функціонал поступово розширювався, наприклад, завдяки інтеграції TeamViewer Meeting. TeamViewer не

вимагає реєстрації та є безкоштовним для некомерційного використання, що допомогло програмному забезпеченню стати дуже популярним. Він був встановлений на більш ніж двох мільярдах пристроїв. TeamViewer – основний продукт однойменної компанії, що базується в Геппінгені. До компаній, що використовують програмне забезпечення, належать Siemens Healthineers.

Google Analytics – додаток для детальної статистики відвідувачів сайту, оптимізований для користувачів iOS-пристроїв. Принцип функціонування сервісу такий же як для звичайних ОС - статистика акумулюється на серверах Google, а користувачі визначають цілі і послідовності переходів. Це може бути певна сторінка, завантаження файлів, оцінка рекламної стратегії та інші елементи. В цілому, інтерфейс Google Analytics для iPhone адаптований саме для iOS-девайсів, що робить роботу з сервісом зручним і комфортним.

AIDA64 – утиліта, що дає змогу користувачу отримати апаратну і програмну інформацію для пристроїв на базі операційної системи iOS. Додаток відображає різні діагностичні дані для телефонів і планшетів, включаючи: інформацію про процесор та його виробника, розміри екрану, щільність пікселів і інформацію про вбудовану камеру, рівень заряду і тип батареї, детальну інформацію про потужності, розширену інформацію про батарею, інформацію про WiFi і мережі, використання пам'яті і сховища, опитування датчиків, діагностику екрану.

1.2. Висновок до першого розділу

Отже, існує багато зовсім різних за ціллю та значенням типів додатків. А саме – соціальні мережі, месенджери, ігри, додатки для бізнесу, додатки для заняття фітнесом, додатки для навігації, розвиваючі додатки, утиліти та багато інших.

Вони всі відрізняються як своє суттю, ціллю, значенням, так і кількістю людей, яким вони можуть бути цікаві. Наприклад, соціальні мережі цікаві більшості людей на планеті, адже вони створені, здебільшого, для спілкування між користувачами. Ігри йдуть на другому місці у цьому рейтингу, адже всі люди люблять в щось погратися.

З цього можна зробити висновок, що вірним способом зробити успішний та цікавий для користувачів додаток – це розробити соціальну мережу чи гру. Але, насправді, це буде хибним вибором, адже як багато людей зацікавлені та користуються такими типами додатків, так і конкуренції тут буде максимальна кількість.

Отже, єдиним способом зробити додаток, яким будуть користуватися люди – це кинути сили на розробку у не дуже конкурентній галузі. Наприклад – додаток для бізнесу, а саме – для допомоги керування розпорядком дня.

РОЗДІЛ 2.

АНАЛІЗ ТА ВИБІР ІНСТРУМЕНТІВ ДЛЯ СТВОРЕННЯ ДОДАТКУ НА БАЗІ iOS

2.1. Аналіз і вибір типу додатка

2.1.1. Нативні додатки

Ці додатки називають нативними тому, що вони написані рідною (з англ. native - рідний) для певної платформи мовою програмування. Для Android цією мовою є Java, тоді як для iOS - objective-C або Swift.

Нативні додатки знаходяться на самому пристрої, доступ до яких можна отримати, натиснувши на іконку. Вони встановлюються через спеціальний магазин від розробника операційної системи (на Android це PlayMarket, а на iOS – AppStore).

«Вони розроблені спеціально для конкретної платформи і можуть використовувати всі можливості пристрою – акселерометр, повну потужність процесора, Lidar-датчик, камеру, GPS, компас, список контактів та інші. Також вони можуть розпізнавати стандартні жести, встановлені операційною системою або зовсім нові жести, які використовуються в конкретному додатку.»[3]

В силу того, що нативні додатки оптимізовані під конкретну ОС, вони органічно вписуються в будь-який смартфон, відрізняючись високою швидкістю роботи і продуктивністю.

Нативні додатки можуть отримати доступ до системи оповіщення пристрою, а також, в залежності від призначення нативного додатки, воно може повністю або частково обходитися без наявності інтернет-з'єднання.

Кафедра КІТ (47)				НАУ 21.38.27.000 ПЗ			
<i>Виконав</i>	<i>Сунко Д.В.</i>			АНАЛІЗ ТА ВИБІР ІНСТРУМЕНТІВ ДЛЯ СТВОРЕННЯ ДОДАТКУ НА БАЗІ iOS	<i>Літера</i>	<i>аркуш</i>	<i>аркушів</i>
<i>Керівник</i>	<i>Воронін А.М.</i>					38	25
<i>Консульт.</i>					УС-212М 122		
<i>Н. контроль</i>	<i>Райчев І.Е.</i>						

Плюси нативних додатків:

- швидкість роботи та продуктивність;
- висока ступінь безпеки;
- розширений інтерфейс;
- максимально можлива функціональність;
- можливість працювати без з'єднання з інтернетом;
- зручність для кінцевого користувача.

Мінуси нативних додатків:

- охоплення платформ;
- тривалі строки розробки;
- відносно висока ціна розробки;
- необхідність випускати оновлення у косметичних цілях.

Приклади нативних додатків: Shazam, Gmail, Telegram, YouTube, Steam, Snapchat, Bolt та інші.

2.1.2. Мобільні веб-додатки

«Насправді мобільні веб-додатки не є додатками як такими. Адже справа в тому, що веб-додаток, по суті, являє собою сайт, який адаптований і оптимізований під будь-який смартфон. І для того, щоб скористатися ним, достатньо мати на пристрої браузер, знати його адресу і мати інтернет-з'єднання (завдяки йому відбувається оновлення інформації в даному виді додатків).»[3]

Запускаючи мобільні веб-додатки, користувач виконує всі ті дії, які він виконує при переході на будь-який веб-сайт, а також отримує можливість «встановити» їх на свій робочий стіл, створивши закладку сторінки веб-сайту.

«Веб-додатки відрізняються кроссплатформістю, тобто здатні функціонувати, незалежно від платформи девайса. Козирем в їх рукаві виступає і те, що вони не використовують його програмне забезпечення. А через те, що є мобільною версією сайту з розширеним інтерактивом, веб-додатки не відбирають дорогоцінний місце в пам'яті смартфона.»[4]

«Веб-додатки набули широкої популярності в той час, коли почав розвиватися HTML5 і люди усвідомили, що можуть отримати доступ до безлічі функцій нативних додатків, просто зайшовши на веб-сайт через звичайний браузер. На сьогоднішній день складно сказати, де саме розташовується чітка межа між веб-додатками і звичайними веб-сторінками, оскільки функціонал HTML5 зростає з кожним днем і все більше і більше сайтів його використовують.»[4]

Розробляються веб-додатки за допомогою інструментів і фреймворків, які стали традиційними. Внаслідок чого процес їх розробки останнім часом істотно прискорився. Фахівців з їхньої розробки набагато більше, ніж інших.

«У той же час камінь в город веб-додатків слід кинути за нездатність працювати з ними без Інтернету. Причому з цього випливає і інший мінус - їх продуктивність, яка знаходиться на середньому рівні, в порівнянні з іншими видами додатків. Більш того, вона залежить від можливостей інтернет-з'єднання провайдера послуг.»[5]

Плюси мобільних веб-додатків:

- повне охоплення платформ;
- простий та швидкий процес розробки;
- невелика кількість конкуренції на ринку розробників;
- відсутність необхідності загрузки додатку з магазину додатків.

Мінуси мобільних веб-додатків:

- обов'язкове підключення до інтернету;
- поганий інтерфейс додатку;
- неможливість відправити push-сповіщення;
- продуктивність та швидкість роботи;
- дуже поганий рівень безпеки.

Приклади мобільних веб-додатків: last.fm, google.com/maps та інші.

2.1.3. Гібридні додатки

Гібридні додатки являють собою поєднання веб- і нативних додатків. Особливо, їх кроссплатформенність і доступ до функціоналу смартфона. Такі

додатки можуть бути завантажені виключно з маркетів на кшталт GooglePlay та AppStore. Разом з тим вони мають у своєму розпорядженні опцію автономного оновлення інформації, а для їх роботи необхідно інтернет-підключення. Без наявності останнього веб-функції просто не працюють.

«Серед багатьох компаній вибір найчастіше падає на розробку саме гібридних додатків. Це можна пояснити тим, що гібридні програми здатні поєднувати гідності нативних з технологічною актуальністю, яка забезпечується останніми веб-технологіями. Однак, на відміну від нативних, швидкість гібридних додатків набагато нижча, але і коштувати розробка гібридного додатку буде менше. Спорідненість гібридних додатків з веб-додатками, в свою чергу, дає плоди у вигляді того, що в них можна легко і оперативно вносити корективи. Тобто розробникам не доводиться, як у випадку з нативними, повторно розміщувати додаток в магазині заради усунення помилок попередньої версії.»[6]

Розробка гібридного додатку здається перспективною ще й тому, що вона має на увазі створення його відразу під дві платформи. Як наслідок, це позбавляє головного болю, пов'язаного з окремою розробкою програми під кожен ОС. Чи це вирішальний фактор? Безумовно.

«Крім всього іншого, потрібно взяти на замітку, що якість і можливості гібридних додатків залежать, перш за все, від фреймворка, який використовує розробник. Також варто приділити належну увагу факторам, які роблять гібридні програми варіантом на тлі інших.»[6]

Отже, варто розробляти його, якщо: є необхідність заощадити в бюджетному плані; потрібно створити відносно нескладний додаток з простою анімацією; є завдання оперативної розробки програми як мінімум на 2 платформи.

Плюси гібридних додатків:

- вартість та швидкість розробки;
- невелика кількість конкуренції на ринку розробників;
- кроссплатформеність;
- опція автономного оновлення.

Мінуси гібридних додатків:

- некоректна робота при відсутності інтернет-з'єднання;
- погана швидкість роботи;
- мінімальна кількість можливостей у дизайні.

Приклади гібридних додатків: HeartCamera, TripCase та інші.

2.2. Аналіз і вибір мови програмування

2.2.1. Swift

Swift – це потужна та інтуїтивно зрозуміла мова програмування для macOS, iOS, watchOS, tvOS та інших. Написання коду Swift – це інтерактивно та весело, синтаксис стислий, але виразний, а Swift включає сучасні функції, які люблять розробники.

Ця мова – це результат останніх досліджень мов програмування у поєднанні з багаторічним досвідом створення платформ Apple. Іменовані параметри виражаються в чистому синтаксисі, що робить API у Swift ще простішим для читання та обслуговування. Ще краще, навіть не потрібно вводити крапки з комою. Визначені типи роблять код чистішим і менш схильним до помилок, тоді як модулі усувають заголовки та надають простори імен. Для найкращої підтримки міжнародних мов та смайлів, рядки мають підтримку формату Unicode і використовують кодування на основі UTF-8, щоб оптимізувати продуктивність для різноманітних випадків використання. Керування пам'яттю здійснюється автоматично за допомогою жорсткого, детермінованого підрахунку посилань, зводить використання пам'яті до мінімуму без накладних витрат на збір сміття.

Swift виключає цілі класи небезпечного коду. Перед використанням змінні завжди ініціалізуються, масиви та цілі числа перевіряються на переповнення, пам'ять автоматично управляється та забезпечується ексклюзивний доступ до захисників пам'яті від багатьох помилок програмування. Синтаксис налаштований для спрощення визначення наміру розробника – наприклад, прості тризнакові ключові слова визначають змінну (var) або константу (let). І Swift пильно слідкує за

типами значень, особливо для таких часто використовуваних типів, як масиви та словники. Це означає, що коли ви робите копію чогось із цим типом, ви знаєте, що тип не буде змінено в іншому місці.

Ще однією функцією безпеки є те, що за замовчуванням Swift-об'єкти ніколи не можуть бути нульовими. Фактично, компілятор Swift зупинить розробника від спроб створити або використовувати об'єкт nil з помилкою під час компіляції. Це робить написання коду набагато чистішим та безпечнішим, а також запобігає величезній категорії збоїв у написанні додатків. Однак бувають випадки, коли нуль є дійсним та доречним. У цих ситуаціях Swift має інноваційну функцію, відому як «Optionals». Опціонали можуть містити nil, але синтаксис Swift змушує безпечно боротися з цим, використовуючи спеціальний синтаксис, щоб вказати компілятору, що розробник розуміє поведінку та буде безпечно обробляти її.

Із самого початку Swift був розроблений, щоб бути швидким. Використовуючи неймовірно високопродуктивну технологію компілятора LLVM, код Swift перетворюється на оптимізований код, який отримує максимум від сучасного обладнання. Синтаксис і стандартна бібліотека також були налаштовані так, щоб найбільш очевидний спосіб написання коду також працював найкраще, незалежно від того, працює він у годиннику на зап'ясті або на кластері серверів.

Свіфт є наступником мов C та Objective-C. Він включає примітиви низького рівня, такі як типи, управління потоком та оператори. Він також надає об'єктно-орієнтовані функції, такі як класи, протоколи та загальні засоби, надаючи розробникам Cocoa та Cocoa Touch необхідну продуктивність та потужність.

Свіфт може відкрити двері у світ кодування. Насправді він був розроблений для того, щоб бути першою мовою програмування для будь-кого, незалежно від того, навчається майбутній розробник ще в школі чи вивчає нові шляхи кар'єри. Для викладачів, Apple створила безкоштовну навчальну програму для навчання Свіфта як у класі, так і поза ним.

Початкові розробники програм можуть отримати доступ до безкоштовних курсів, щоб навчитися створювати свої перші програми в Xcode. І магазини Apple по

всьому світу проводять сесії Apple Coding & Apps, де будь-хто можете отримати практичний досвід роботи з кодом Swift.

У Swift 5 не потрібно змінювати будь-який свій код Swift 4, щоб використовувати нову версію компілятора. Натомість, можна почати використовувати новий компілятор і мігрувати у своєму власному темпі, користуючись новими функціями Swift 5, по одному модулю за раз. І Swift 5 вводить двійкову сумісність для програм. Це означає, що більше не потрібно включати бібліотеки Swift у програми, націлені на поточні та майбутні випуски ОС, оскільки бібліотеки Swift будуть включені в кожен випуск ОС в майбутньому. Додатки використовуватимуть останню версію бібліотеки в ОС, а код працюватиме без перекомпіляції. Це не тільки спрощує розробку програми, але й зменшує розмір програми та час її запуску.

Як і Swift Playgrounds для iPad, Playgrounds в Xcode роблять написання коду Swift неймовірно простим і веселим. Введіть рядок коду, і результат з'явиться негайно. Потім ви можете швидко переглянути результат збоку вашого коду або закріпити його безпосередньо внизу. У поданні результатів можуть відображатися графіки, списки результатів або графіки значення з часом. Можна відкрити Timeline Assistant, щоб спостерігати, як складний код розвивається та анімується, чудово підходить для експериментів з новим кодом інтерфейсу користувача або для відтворення анімованої сцени SpriteKit під час кодування. Коли ви вдосконалили свій код у Playgrounds, просто перемістіть цей код у свій проект. Swift також інтерактивний, коли ви використовуєте його в терміналі або в консолі налагодження LLDB Xcode. Можна використовувати синтаксис Swift для оцінки та взаємодії з запущеним додатком або написати новий код, щоб побачити, як він працює в середовищі, подібному до сценарію.

Swift Package Manager – це єдиний крос-платформенний інструмент для створення, запуску, тестування та упаковки бібліотек та виконуваних файлів Swift. Пакети Swift – найкращий спосіб розповсюдити бібліотеки та вихідний код для спільноти Swift. Конфігурація пакетів написана в самому Swift, що полегшує налаштування цілей, декларування продуктів та управління залежностями пакетів. У

новій версії Swift 5, команда `swift run` тепер включає можливість імпортувати бібліотеки в REPL без необхідності створювати виконуваний файл. Сам Swift Package Manager фактично побудований за допомогою Swift і включений до проекту Swift з відкритим кодом як пакет.

Можна створити абсолютно нову програму за допомогою Swift сьогодні або почати використовувати код Swift для реалізації нових функцій та функціональних можливостей у своїй програмі. Код Swift співіснує з існуючими файлами Objective-C в одному проекті з повним доступом до API Objective-C, що полегшує його прийняття.

2.2.2. Objective-C

«Objective-C – це мова програмування, яка використовується для діяльності загального призначення. Хоча це не характерно для якоїсь конкретної платформи або системи, воно може суттєво допомогти у розробці безлічі інших систем. Програмування на Objective-C додає можливості обміну повідомленнями до мови програмування C.»[7]

«Objective-C – одна з основних мов програмування, що раніше використовувалась Apple для системи iOS, і може використовуватися для створення мобільних додатків для цієї конкретної платформи. Як надмножина мови програмування C, вона пропонує розробникам можливість бути більш орієнтованими на деталі, а також підтримувати об'єкти та інші мови.»[7]

Існують різні мови програмування. В даний час три мови програмування користуються великим попитом, тобто Swift, Objective C та C++.

«Objective-C чудово підходить для управління пам'яттю, також доступні компілятори, які можуть перетворити код Objective-C у статичний аналіз коду, який потім мова використовуватиме для сортування корисної інформації та “сміття” у дві окремі категорії.»[8]

«Найголовніше, що треба знати про Objective-C – це те, що він в значній мірі об'єктно-орієнтований. Можна переміщати графіки та обробляти файли,

використовуючи цю додаткову мову, але важливо розуміти її обмеження так само, як важливо розуміти її переваги.»[9]

Як мова, винайдена у 1980-х, Objective-C все ще має велику кількість функцій, які використовуються в розробках мобільних додатків для iOS. Незважаючи на те, що не було інновацій, які дозволяють використовувати Objective-C на всіляких платформах, він є інтегративним для використання C та інших мов для додатків iOS.

«Objective-C конкурує з Swift – новою мовою програмування iOS. Існує багато суперечок, між якими мова програмування пропонує кращі результати для розробок мобільних додатків iOS.»[9]

Розробка додатків Objective-C може бути спрощена і може стати чудовим способом додати об'єктні функції до додатка. Існує кілька ключових моментів, про які слід пам'ятати щодо цієї надмножинної мови: потрібне технічне обслуговування – це стосується оновлення програм, розроблених за допомогою Objective-C. Мова може мати дату, але вона не застаріла. Просто потрібна певна робота, щоб вона залишалася актуальною; не потрібно стільки адаптації – багатьом API все ще потрібно наздогнати додатки, розроблені за допомогою Swift. Це означає, що може бути простіше застосувати Objective-C до мобільного додатка iOS.

Objective-C можна використовувати для впорядкування мобільних додатків iOS на всіх типах платформ Apple iOS, включаючи телефони та планшети.

2.2.3. Порівняння Swift та Objective-C

Objective-C та Swift - це мови програмування, які використовуються для кодування програмних продуктів Apple під операційними системами macOS, iOS, tvOS, watchOS, CarPlay та Linux (лише для Swift). Ці мови мають відповідні інтерфейси програмування програм (API) та фреймворки (Cocoa, Cocoa Touch). Objective-C був створений наприкінці 80-х і вплинув на Smalltalk (для об'єктно-орієнтованого синтаксису) та C (для не-об'єктних) мов програмування. Новіша мова, Swift, не заснована на мові C. Натомість вона більше не має спадку, а просто компілюється з бібліотеками C та фреймворками. Swift пропонує потужніші інструменти та сучасні рішення для кодування для більш простого, але виразного

кодування, безпеки додатків та продуктивності. Крім того, мова є інтуїтивно зрозумілою, а отже, її легше вивчати для нових програмістів.

Спочатку Objective-C був створений компанією Stepstone Бредом Коксом і Томом Лавом у 1980-х. З тих пір Objective-C є стандартною мовою програмування для macOS і, надалі, iOS та інших платформ Apple. Основна характеристика полягає в його зрілості, оскільки вона вже багато років використовується і тестується для програмного забезпечення Apple.

На відміну від цього, Swift не обмежується лише ОС Apple. У 2015 році він став мовою програмування з відкритим кодом та кросплатформною. Це забирає небезпечне управління покажчиком і одночасно забезпечує взаємодію з давніми базами коду Objective-C та C.

У таблиці 2.1 наведено загальні технічні характеристики для Swift та Objective-C.

Таблиця 2.1

Технічні характеристики Objective-C та Swift

Objective-C	Swift
Типи перевіряються, але не застосовуються під час компіляції	Типи застосовуються під час компіляції та виконання
Можливі виходи за межі та переповнення стека	Виходи за межі та переповнення стека запобігаються
«Легкі дженеріки» є інформаційними для часу компіляції	«Дженеріки» застосовуються під час компіляції та виконання
Заголовкові файли – це окремі файли, що компілюються	Один файл, що містить в собі інтерфейс та код програми
Ручне управління пам'яттю у CoreFoundation	Автоматичне управління пам'яттю у CoreFoundation
Надсилання повідомлення (динамічне відправлення)	Відправлення статичного методу для типів значень
Є підтримка Objective-C++	Нема підтримки Objective-C++

Переваги Objective-C перед Swift:

- Objective-C - це добре перевірена мова. Objective-C є надійним і використовувався та тестувався багатьма розробниками. На Objective-C написано безліч проектів, і є багато чудових сторонніх фреймворків;
- Objective-C є більш стабільним. Оскільки Swift все ще розвивається як мова, він регулярно оновлює версії, а програми, написані Swift, потрібно перенести на найновішу версію. Логічно, що це вимагає часу та грошей. Коли програмі, написаній на Objective-C, ці операції непотрібні;
- Простіше використання приватних API. У випадках, коли для продукту необхідне певне спеціальне рішення (наприклад, специфічна зміна кольору або інтеграція), Objective-C простіше підключити до приватних API. Оскільки Swift постійно розвивається, він нестабільний, і цей факт може призвести до деяких збоїв у роботі таких типів з'єднань;
- Objective-C легко сумісний з C++. Оскільки у нього коріння з мови C, продукти, що інтегрують частини коду C або C++, працюють безперебійно. Swift, навпаки, несумісний з мовою C++.

Недоліки Objective-C перед Swift:

- Важче вчитися. Objective-C – це окрема мова, яка відрізняється синтаксисом від інших мов програмування. Управління пам'яттю є більш складним і має спадщину, необхідну для розуміння C та Smalltalk. Коротше кажучи, Objective-C має вищий бар'єр для нових розробників через складний синтаксис та умови кодування;
- Розробники Objective-C – менш кваліфіковані. Знайти належного розробника Objective-C може бути складно. Їх кількість помірно зменшується, оскільки нові розробники вчаться в основному Swift, а інші переходять з Objective-C на Swift. Будь-які навички програмування вимагають регулярної практики, тому з часом прогнозується зменшення кількості кваліфікованих програмістів Objective-C.

Є кілька варіантів, де все-таки переважно використовувати Objective-C – якщо програма вже написана на Objective-C і має величезну базу коду (100 000+ рядків коду). Найлогічніша рекомендація щодо використання Objective-C полягає у тому, якщо проект сильно залежить від цієї мови або програма вже написана на Objective-C і функціонує (приклад: Facebook); якщо ви розробляєте фреймворк, використовуйте Objective-C. Поки що Swift не є стабільним ABI (бінарний інтерфейс додатків), що відповідає правильній комунікації між усіма елементами машинного коду. ABI повинен бути стабільним, якщо двійковий код складений з різними версіями мови. Як приклад, PSPDFKit – це фреймворк, який використовує Objective-C і використовується для обробки документів PDF. Таким чином, якщо ви плануєте створити SDK, Objective-C є кращим варіантом; якщо сторонні фреймворки C або C++ широко використовуються. Зазвичай це рідкісна ситуація, але якщо ці рамки потрібні, рекомендується використовувати Objective-C, оскільки це підмножина C, і тому її варто розглянути.

Переваги Swift:

- Swift більш швидкий. Продуктивність Swift майже така ж, як і в C++, яка вважається найшвидшою в арифметиці обчислення алгоритму. Apple виношувала цю ідею і працювала над покращенням швидкості роботи Swift. Наприклад, Swift 2.0 переміг C++ у декількох алгоритмах обчислень, таких як алгоритм Мандельброта. Objective-C працює повільніше, оскільки містить застарілий API;
- Swift безпечніший. На сьогодні захист даних програми є важливою характеристикою успішного продукту. Конструкція Swift була розроблена для виключення та уникнення помилок за допомогою її функцій – загальних та додаткових засобів для досягнення стабільності програми. Тому програми, розроблені в Swift, менш схильні до помилок та збоїв;
- Свіфт є більш читабельним. Код у Swift більше нагадує англійську, що полегшує читання та вимагає менше часу для перевірки коду. Крім того, загалом для однієї і тієї ж функції потрібно набагато менше рядків коду.

Swift легко читати програмістами JavaScript, Java, Python, C# та C++, які певною мірою можуть ним користуватися;

- Свіфт має менше коду. Swift – більш компактна мова для програмування. Однак цей факт, звичайно, не означає простоти коду. Часом писати буває дуже важко, але це приносить більше переваг;
- Свіфт менш схильний до помилок. Синтаксис та побудова мови виключають кілька типів помилок, потенційно можливих у Objective-C. Цей контроль означає менше збоїв та випадків несподіваної поведінки. Це, однак, не заважає писати поганий код, звичайно, але розробник краще захищений від небажаних помилок. Це можна назвати «контролем якості»;
- Swift інтегрується з управлінням пам'яттю. ARC (Automatic Reference Counting), функція управління пам'яттю всередині компілятора Clang, управляє пам'яттю цифрових об'єктів. Це легко побачити, коли Swift працює над процедурним та об'єктно-орієнтованим кодом за допомогою API Cocoa. Що стосується Objective-C, процедурний код працює з Corel Graphic API, і розробник несе відповідальність за його контроль;
- Swift – мова з відкритим кодом. Swift був спочатку створений для платформ Apple (iOS, OS X, watchOS, tvOS) і надалі був розширений до Linux. Його творці зрозуміли той факт, що для побудови чудової сучасної мови програмування вона повинна бути доступною для кожного. Свіфт допоміг розробити багато сторонніх платформ та інструментів, що призвело до зростання спільноти;
- Свіфт має інтерактивне кодування. Swift Playgrounds представив нові можливості для розробників. Цей інструмент дає змогу тестувати код на місці, не складаючи його великі фрагменти та не створюючи весь додаток. «Ігрові майданчики» візуалізують дані, а програмісти можуть швидко перевірити та виправити все разом із подальшим розвитком. Це особливо застосовно до експериментів з кодом. Остання версія xCode IDE надає розробникам великий простір для експериментів у вигляді

простого редактора (включаючи панель з лініями, зображеннями, кінцевий вигляд);

- Свіфт ближче до інших платформ. Цей момент дуже важливий, особливо якщо говорити про співпрацю між програмістами, що створюють один і той самий додаток на різних платформах. Сучасну мову програмування від Apple легше зрозуміти розробникам, які не працюють під iOS, і мінімізує час, необхідний для додаткових пояснень та роз'яснень. Це позитивно впливає на продуктивність праці;
- Swift – основна увага Apple. Apple Inc. концентрується на розвитку Swift як основної мови програмування. Нещодавно WWDC (Всесвітня конференція розробників) провела детальну презентацію щодо Swift 6, де вони представили ряд чудових функцій та оновлень.

Недоліки мови Swift:

- Повільніша швидкість компіляції. Оскільки Swift є більш складною мовою та має більше можливостей, важче переконатися, що все є точно. На даний момент вихідний код Swift коригується з кожним оновленням версії. Кожного разу це витрачає час і гроші. З іншого боку, Apple зазначила, що зараз вони працюють над цим фактором і очікують запропонувати рішення у 2023 році;
- Відсутність стійкості ABI. Ще одна головна проблема Swift полягає в його нестабільності ABI. Подібно до API, який визначає всю комунікацію між усіма елементами у вихідному коді, ABI визначає правила зв'язку для машинного коду. Це здебільшого має значення у великих проектах, коли компілюється частина програми з декількома версіями Swift. Отже, для забезпечення стабільного зв'язку в двійковому коді різних версій компілятора, ABI повинен бути стабільним і добре структурованим. Це може призвести до деяких проблем із фреймворками у версіях Swift;
- Відсутність імпорту C++. Якщо проект сильно залежить від C++, Swift не є мовою, яку слід обирати для цього завдання.

Приклади використання Swift – LinkedIn, соціальна мережа для професіоналів, яка перейшла від Objective-C до Swift у розробці рідних додатків. У 2016 році відбулася чудова промова, в якій були представлені аргументи цього рішення для компанії. Суть полягала в тому, що Swift був кращим вибором для менших та початкових проектів, оскільки з ними було простіше працювати. З іншого боку, протягом 3 років Свіфт став більш зрілим і вирішував свої проблеми на ранніх стадіях.

Компанія Lyft, транспортна компанія на вимогу, використовує Swift для розробки власних додатків для iOS.

Інші відомі додатки, такі як Yahoo Weather, Clear, Hipmunk, WordPress та Firefox iOS, також використовують Swift.

2.3. Стандарти та правила UI/UX дизайну в iOS

UX та UI: два терміни, які часто використовуються як взаємозамінні, але насправді означають дуже різні речі. То в чому саме різниця?

UI дизайн

Незважаючи на те, що сфера UI дизайну вже існує багато років та є більш давньою та вивченою, порівняно з UX дизайном, все одно дуже важко відповісти на питання «Що ж таке UI дизайн?». UX дизайн – це, по суті, процес підтримки поведінки користувачів завдяки зручності та корисності, що забезпечуються у взаємодії з продуктом. UI дизайн, в свою чергу, є доповненням до UX дизайну. У процесі створення дизайну інтерфейсу користувача, робиться презентація та зовнішній вигляд додатку.

Якщо подивитись на вакансії та посадові інструкції для дизайнерів UI, в основному там будуть інтерпретації професії, подібні до графічного дизайну, іноді поширюючись також на дизайн бренду та навіть розробку інтерфейсу.

Якщо ж подивитись на визначення UI дизайну в інтернеті чи літературі, то в більшості випадків можна знайти тлумачення, які цілком ідентичні UX дизайну, навіть посилаючись на ті самі структурні методи.

Тож треба встановити чіткі границі між цими двома поняттями раз і назавжди. На відміну від UX, дизайн інтерфейсу користувача – це суто цифровий термін. Користувацький інтерфейс – це точка взаємодії між користувачем та цифровим пристроєм або продуктом. Наприклад, сенсорний екран на смартфоні або сенсорна панель, за допомогою якої користувач обирає, яку каву він хоче отримати з кавоварки. Стосовно веб-сайтів та додатків, дизайн інтерфейсу враховує зовнішній вигляд, відчуття та інтерактивність продукту. Дуже важливо переконатися, що користувацький інтерфейс продукту є якомога більш інтуїтивним, а це означає ретельно продумати кожен візуальний інтерактивний елемент, з яким може зіткнутися користувач. Дизайнер інтерфейсу користувача буде думати про піктограми та кнопки, типографіку та колірні схеми, інтервали, зображення та адаптивний дизайн.

Як і дизайн користувацького досвіду, UI дизайн – це дуже складна та довга робота. Він відповідає за перетворення розробки, досліджень, вмісту та візуальної частини продукту в приємний, спрямовуючий та навіть надихаючий досвід для користувачів.

Дизайн інтерфейсу користувача – це суто цифрова практика. Він враховує всі візуальні, інтерактивні елементи інтерфейсу продукту, включаючи кнопки, піктограми, інтервали, типографіку, кольорові схеми та адаптивний дизайн.

Мета дизайну інтерфейсу користувача – візуально вести користувача через інтерфейс продукту. Вся справа в створенні інтуїтивного досвіду, який не вимагає від користувача занадто багато думати.

Дизайн інтерфейсу переносить сильні сторони та візуальні цінності бренду в інтерфейс продукту, забезпечуючи узгодженість та естетичність.

UX дизайн

Дизайн користувацького досвіду – це першочерговий спосіб проектування продуктів. Дон Норман, когнітивний вчений і співзасновник Nielsen Norman Group Design Consultancy, вводить термін «UX design» наприкінці 1990-х. Ось як він це описує:

"UX охоплює всі аспекти того, як кінцевий користувач взаємодіє з компанією, її послугами та продуктами".

Визначення Дона Нормана говорить про те, що, незалежно від середовища, досвід користувача може охоплювати будь-які взаємодії між потенційним або активним клієнтом та компанією. Як науковий процес, це поняття можна застосувати до чого завгодно: літаки, підземні та надземні переходи, парки та сквери тощо. Однак, незважаючи на те, що це науковий термін, його використання з самого початку майже повністю було в цифрових сферах. Однією з причин цього явища є те, що технологічна індустрія почала дуже стрімко зростати приблизно під час винаходу цього терміна.

По суті, UX застосовується до будь-чого, що може бути використано людиною – будь то додаток, кавоварка або відвідування супермаркету. Частина UX стосується взаємодії між користувачем та продуктом чи послугою. Тоді дизайн досвіду користувача враховує всі різні елементи, що формують цей досвід. Дизайнер UX думає про те, як досвід користування продуктом впливає на відчуття користувача, і про те, як легко користувачеві виконувати бажані завдання. Наприклад: Наскільки легким є процес оформлення замовлення під час покупок в Інтернеті? Наскільки легко вхопити овочечистку? Чи спрощує додаток для онлайн-банку управління своїми грошима? Кінцевою метою дизайну UX є створення легких, ефективних, релевантних та всебічних приємних вражень для користувача.

Отже, UX дизайн – це процес розвитку та поліпшення якості взаємодії між користувачем та усіма аспектами компанії.

Теоретично, дизайн досвіду користувача є нецифровою практикою (когнітивна наука), але використовується та визначається переважно цифровими галузями. UX дизайн не стосується візуальних аспектів, він фокусується саме на загальному відчутті від користування продуктом.

Як UX та UI працюють разом

Вище було досліджено відмінності між UX та UI, тепер треба розглянути, як вони працюють разом. Може виникнути думка, що одна з галузей важливіша за іншу, але насправді вони обидві мають вирішальне значення. Колись видатна та

шановна дизайнер та експерт Хельга Морено, досить красномовно висловила це у своїй статті «Розрив між дизайном UX та UI»:

«Те, що виглядає чудово, але важко у використанні, є прикладом гарного UI та поганого UX. У той же момент, щось дуже корисне, що виглядає жахливо, є прикладом чудового UX та поганого UI.»

UX та UI міцно поєднуються, і хоча є дуже багато прикладів прекрасних додатків в одній сфері та, водночас, жахливих в іншій, можна тільки уявити, наскільки успішнішими вони могли б бути, якщо були б сильними в обох сферах.

UI дизайн можна назвати вишенькою на торті UX. Уявіть, що ви придумали дивовижну ідею для програми – те, чого явно не вистачає на ринку та може справді змінити життя людей на краще. Ви наймаєте дизайнера UX, щоб провести дослідження користувачів і допомогти точно визначити, які функції мають бути у додатку. Додаток пропонує щось, що потрібно цільовій аудиторії; однак, завантажуючи його, вони виявляють, що текст на кожному екрані ледь розбірливий (наприклад, жовтий текст на білому фоні). Більше того, кнопки розташовані занадто близько один до одного; користувачі раз за разом продовжують натискати неправильну кнопку помилково. Це класичний випадок поганого користувальницького інтерфейсу, який руйнує те, що могло б бути хорошим UX.

З іншого боку, є багато прикладів насправді прекрасних додатків, які дуже чудово виглядають; але окрім приголомшливої анімації та точної кольорової схеми, їх дуже складно використовувати. Хороший інтерфейс ніколи не може компенсувати поганий UX; це як підібрати красиво оформлений торт, який жахливий на смак.

Отже, що стосується дизайну продукту, UX та UI доповнюють один одного – і на сучасному конкурентному ринку, розробка та дослідження обох аспектів є абсолютно необхідною дією.

Мета дизайну iOS:

- Доступність – бути настільки простим і прямолінійним наскільки це можливо;

- Корисність – UI ніколи не повинен конкурувати з контентом, а тільки допомагати людям розуміти і взаємодіяти з ним;
- Розмежування – UI повинен допомагати користувачам фокусуватися на тому, що відбувається на передньому плані.

Доступність: кожен екран додатку повинен бути зрозумілим і відповідати функції, яку він несе. Зміст повинен бути добре написаний і текст повинен бути розбірливим.

Принципи дизайну iOS:

- Естетична цілісність;
- Узгодженість;
- Прямі маніпуляції;
- Контроль користувача.

Естетична цілісність: зовнішній вигляд і поведінка додатка повинні доповнювати та відпоідати його функціональності.

Узгодженість: впровадження в додаток знайомих користувачам стандартів і парадигм. Мається на увазі використання стандартних елементів дизайну платформи на яку розробляється додаток.

Прямі маніпуляції: інтерфейс повинен бути зрозумілим для будь-якого користувача, який вперше заходить в додаток.

Контроль користувача: додаток повинен давати користувачеві відчуття, що він приймає рішення, а не додаток робить це за користувача.

2.4. Аналіз існуючих сервісів для створення UI/UX дизайну

Scetch був розроблений спеціально для дизайну інтерфейсу користувача. Розробники обміркували все: від ранніх проектів до високорівневих каркасів дизайну, включаючи співпрацю з командами тестування та розробки. І якщо є завдання, з яким Scetch не може впоратись власноруч, завжди знайдеться доступний плагін.

Виробники Sketch звертають особливу увагу на відгуки своїх користувачів, випускаючи часті оновлення, нові функції та виправлення помилок без коштовних зборів за оновлення. Майбутні версії будуть розширювати межі того, як дизайнери працюють в додатку, додаючи функції співпраці. Уже є багато чудових функцій, і, як уже зазначалося, розробники постійно вдосконалюють їх.

Основні переваги Scetch:

- Це векторний інструмент, який зменшує розмір файлів і робить зміну розміру плавною;
- Послідовність дизайну можна досягти за допомогою символів, бібліотек, сторінок та інших елементів, які безпосередньо пов'язані з дизайнерськими системами;
- Так звані «Асети», код та інші результати роботи у додатку можна безперешкодно експортувати;
- Багатоформатний попередній перегляд дозволяє побачити, як кожен дизайн буде виглядати на різних платформах і в різних браузерях.

Sketch побудований на відкритому форматі, що дозволяє розробникам створювати плагіни та розширення для інтеграції з іншими популярними інструментами або додавання функціональних можливостей, яких Sketch ще не має.

Наприклад:

- Zeplin – полегшує передачу розробникам шляхом складання специфікацій.
- Stark – допомагає дизайнерам створювати доступні дизайни для людей з обмеженими можливостями.
- Userflows – спрощує створення діаграм, карт веб-сайтів та додатків.

Scetch був розроблений спеціально для дизайнерів UI/UX, тоді як інші інструменти, такі як Photoshop та Illustrator, насправді призначені для дуже різних цілей (редагування фотографій та створення штрихових малюнків відповідно). Творці Scetch розуміли, що багато досвідчених дизайнерів UI/UX роками використовують PSD та Illustrator і звикли до інтерфейсу. Отже, вони спростили

інтерфейс Sketch для вивчення та адаптації, імітуючи більшу частину функціональних можливостей робочої області.

Окрім цих інструментів, є маса інших варіантів. **UXtools.co** пропонує чудову альтернативу іншими загальноживаними засобами проектування та прототипування. Як можна побачити на рис. 2.1, статистика показує, що Sketch набагато популярніший серед дизайнерів UI/UX, ніж будь-який інший пропонований інструмент.

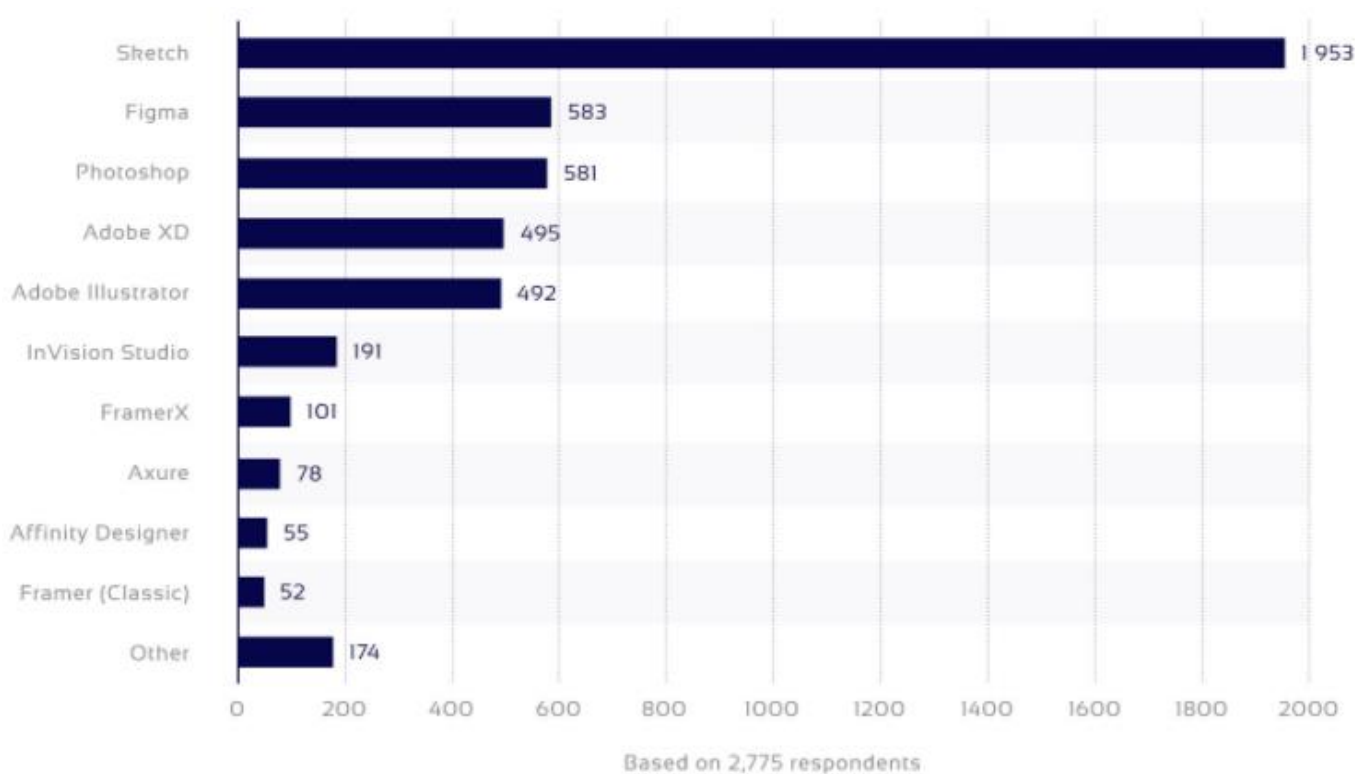


Рис. 2.1. Порівняння кількості користувачів основних інструментів UI/UX

Одним з інших більш популярних інструментів є **Axure**. Axure чудово підходить для вайрфреймів, простих прототипів та інших проектів. Однак він не має розширених візуальних інструментів, і Assets потрібно створювати в іншому інструменті, наприклад PSD, а потім імпортувати їх у бібліотеки ресурсів. На жаль, у нього немає плагіна для Sketch.

Balsamiq – ще один чудовий інструмент для створення прототипів. Але він також не має можливості створювати візуальні зображення з високою точністю.

Figma – це хмарний інструмент дизайну, який за функціональністю та функціями схожий на Sketch, але з великими відмінностями, які роблять Figma кращим для співпраці в команді.

Figma працює з будь-якою операційною системою, яка запускає веб-браузер. Mac, ПК з Windows, машини з Linux і навіть Chromebook можна використовувати з Figma. Це єдиний інструмент дизайну, який спроможний на це.

У багатьох організаціях дизайнери використовують Mac, а розробники – ПК з Windows. Фігма допомагає об'єднати ці групи. У Figma немає необхідності в посередницькому механізмі, щоб зробити проектну роботу доступною для кожного.

Оскільки Figma базується на браузері, команди можуть співпрацювати так, як це було б у Google Docs. Люди, які переглядають та редагують файл, відображаються у верхній частині програми як кругові аватари. Кожна людина також має іменованій курсор, тож відстежувати, хто над якою задачею працює – дуже легко.

Спільна робота з файлами в режимі реального часу допомагає пом'якшити "design drifting", що можна визначити, як неправильне тлумачення або відхилення від узгодженого проекту. На жаль, це часто призводить до відхилення від встановленої конструкції та повторної роботи.

За допомогою Figma керівник проекту може перевірити, що команда розробляє в режимі реального часу, просто відкривши спільний файл. Якщо дизайнер якось неправильно інтерпретує короткий зміст або історію користувача, ця функція дозволяє керівнику втручатися, коригувати курс і економити незліченні години, які в іншому випадку були б витрачені даремно. (Для порівняння, команди, які використовують Sketch, не можуть негайно визначити, чи збиваються дизайнери з дороги).

Figma використовує Slack як свій канал зв'язку. Коли канал Figma створюється в Slack, будь-які коментарі або редагування дизайну, зроблені в Figma, «розприділяються» для команди. Ця функціональність має вирішальне значення при розробці в реальному часі, оскільки зміни до файлу Figma оновлять кожен інший

екземпляр, де файл вбудований. Зміни в макеті, виправдані чи ні, негайно перевіряються, і канал зворотного зв'язку працює.

Figma також дозволяє мати спільний доступ до будь-якого файлу, сторінки чи фрейму, за допомогою системи дозволів (що називається «artboard» в інших інструментах проектування). Коли спільне посилання створюється на фрейм, людина, що натискає це посилання, відкриє версію Figma у браузері та завантажить збільшений вигляд фрейму.

Ця форма вибіркового обміну, від файлу до фрейму, дозволяє дизайнерам, власникам продуктів та розробникам ділитися саме тим, що потрібно в інструментах відстеження помилок та програмному забезпеченні спільноти, таких як Confluence або SharePoint.

Поки розробники Sketch додавали artboard до прототипування, розробники Figma пішли далі, забезпечивши переходи між фреймами. Проста функція створення прототипів Figma позбавляє від необхідності використовувати інший інструмент, який робить прототипи у стилі слайд-шоу, наприклад InVision або Marvel. Коли потрібна проста презентація з переходами, немає необхідності експортувати в інструменти перегляду.

Прототипи Figma можна розповсюджувати так само, як файли дизайну Figma; кожен, хто має дозвіл на посилання, може переглядати та коментувати прототип, і знову ж таки, цей відгук фіксується на панелі коментарів інструменту та записується у Slack. Розробники можуть бачити робочий процес дизайну, залишати прямі повідомлення для дизайнерів та отримувати виміри та атрибути CSS зсередини прототипу.

2.4.1. Порівняння Scetch та Figma

Scetch та Figma – це дуже потужні інструменти в руках професійних UI/UX дизайнерів, але як і у всього іншого, в них є переваги та недоліки.

Переваги Sketch: великий каталог плагінів і синхронізація з популярними сервісами, експорт коду, командна робота, орієнтація на векторну графіку, можна заплатити один раз і користуватися скільки завгодно без оновлення підписки.

Недоліки Scetch: сумісність тільки з macOS, залежність від ресурсів системи, мало можливостей без плагінів.

Переваги Figma: спільна робота, шаринг файлів, кросплатформеність, висока продуктивність, стабільне оновлення, безкоштовний тариф, історія версій.

Недоліки Figma: для використання потрібен інтернет, проблеми з текстовими шарами, імпорт файлів працює погано.

2.5. Висновок до другого розділу

Отже, існує три основні типи додатків на мобільні платформи – нативні, веб-додатки та гібридні. Вони відрізняються складністю розробки, часом розробки, та, найголовніше – ціллю. З першого погляду може здатись, що найкращим варіантом буде гібридний додаток, адже в такому разі, у майбутньому, можна буде створити додаток і на іншу мобільну платформу, а саме – Android. Але обов'язково треба звернути увагу і на недоліки такого підходу, а саме: некоректну роботу додатку при відсутності інтернет-з'єднання, погану швидкість роботи та замалу кількість можливостей у розробці дизайну додатка. Враховуючи це, та дивлячись на написане у другому розділі, можна зробити висновок, що все ж таки найрашим варіантом залишається нативний додаток, – адже він гарантує швидкість роботи, безперебійну роботу навіть без інтернет-з'єднання та максимально широкі можливості у дизайні, що дуже необхідно користувачам для комфортного використання додатку.

Існує безліч мов програмування, але всі вони були створені для конкретних цілей та платформ. У випадку розробки додатку для платформи iOS, можна обрати серед таких мов, як Swift, React Native, Flutter, Objective-C та інші. Вони всі мають право на існування, але тільки для своїх конкретних цілей. Оскільки ми обрали нативний додаток, то під цей критерій підходить тільки 2 мови – Swift та Objective-C. Але другий варіант вже є застарілим і використовується тільки для підтримання життєздатності додатків, що розроблялись 10 та більше років назад. Саме через це, слід обрати мову Swift, адже вона більш перспективна на сучасному ринку.

Не треба забувати і про таку важливу річ, як дизайн, а саме – UI та UX дизайн. Ці два етапи розробки допоможуть створити додаток більш якісним, зрозумілим для кінцевого користувача та, найголовніше – істотно скоротити час, що буде витрачатися на розробку. Адже на цьому етапі визначаються такі важливі речі, як зручність додатка для користувачів, чи приємно його використовувати та сенс додатку. Тому дуже важливо обрати інструмент, за допомогою якого буде створюватися цей самий UI/UX дизайн. Для платформи iOS зазвичай використовується Scetch, Figma, або Balsamiq. Треба подивитись на переваги та недоліки цих варіантів, та якщо це зробити, то очевидно, що інструмент з назвою Scetch підходить більше інших; адже він розроблявся саме для проектування дизайну під платформу iOS, тому є пріоритетнішим варіантом серед інших.

РОЗДІЛ 3. ПРОЕКТ МОБІЛЬНОГО ДОДАТКУ НА БАЗІ iOS

3.1. Проектування дизайну

3.1.1. Wireframe

Не схожий на архітектурний план, wireframe - це двовимірний скелетний контур веб-сторінки або програми. Wireframes забезпечують чіткий огляд структури сторінки, макета, архітектури інформації, потоку користувачів, функціональності та передбачуваної поведінки. Оскільки каркас зазвичай представляє початкову концепцію товару, стиль, кольори та графіка зведені до мінімуму. Каркасні дроти можна намалювати від руки або створити цифровим способом, залежно від того, скільки деталей потрібно.

Каркасне з'єднання - практика, яка найчастіше використовується дизайнерами UX. Цей процес дозволяє всім зацікавленим сторонам домовитись, де буде розміщена інформація, перш ніж розробники розробить інтерфейс із кодом.

Процес каркасного каркасу, як правило, відбувається під час дослідницької фази життєвого циклу виробу. На цьому етапі дизайнери випробовують обсяг продукту, співпрацюють над ідеями та визначають вимоги бізнесу. Wireframe – це, як правило, початкова ітерація додатку, яка використовується як стрибкова точка для дизайну продукту. Озброївшись цінними уявленнями, зібраними з відгуків користувачів, дизайнери можуть спиратися на наступну, більш детальну ітерацію дизайну продукту – наприклад, на прототип чи макет.

Кафедра КІТ (47)				НАУ 21.38.27.000 ПЗ			
Виконав	Сунко Д.В.			ПРОЕКТ МОБІЛЬНОГО ДОДАТКУ НА БАЗІ iOS	Літера	аркуш	аркушів
Керівник	Воронін А.М.					63	29
Консульт.					УС-212М 122		
Н. контроль	Райчев І.Е.						

Wireframes служать трьом основним цілям:

- Wireframes залишають концепцію орієнтованою на користувача. Вайрфрейми ефективно використовуються як комунікаційні пристрої – вони полегшують зворотній зв'язок з користувачами, провокують бесіди із зацікавленими сторонами та генерують ідеї між дизайнерами. Проведення тестування користувачів на ранній стадії каркасного проектування дозволяє дизайнеру забезпечити чесний зворотний зв'язок та визначити ключові моменти, які допомагають створити та розвинути концепцію продукту. Каркасне проектування – ідеальний спосіб для дизайнерів визначити, як користувач взаємодіє з інтерфейсом. Використовуючи такі пристрої, як Lorem Ipsum, псевдолатинський текст, який виконує роль заповнювача для майбутнього вмісту, дизайнери можуть підказувати користувачам запитання на кшталт "що б ви очікували, що тут буде написано?". Ці ідеї допомагають дизайнеру зрозуміти, що інтуїтивно зрозуміле для користувача, і створити зручні та прості у використанні продукти;
- Wireframes уточнюють та визначають функції додатку. Повідомляючи ваші ідеї клієнтам, вони можуть не мати технічного лексикону, щоб відповідати таким термінам, як «hero image» або «call to action». Специфічні функції проектування вайрфреймів чітко повідомлять клієнтам, як вони функціонуватимуть і яким цілям вони слугуватимуть. Wireframing дозволяє всім зацікавленим сторонам визначити, скільки місця потрібно буде виділити для кожної функції, підключити інформаційну архітектуру додатку до його візуального дизайну та пояснити функціональність екрану. Побачивши функції на вайрфреймі, також можна уявити, як вони всі працюють разом.
- Створювати вайрфрейми швидко і дешево. Вони неймовірно дешеві та прості у створенні. Насправді, якщо у є ручка та папір, то можна швидко замалювати каркас додатку, не витрачаючи ні копійки.

Часто, коли продукт здається занадто відполірованим, користувач рідко може чесно говорити про свої перші враження. Але виявивши суть макета сторінки, можна легко виявити та усунути недоліки та проблеми без будь-яких значних витрат часу чи грошей. Чим пізніше це потрапляє в процес проектування продукту, тим важче вносити зміни.

Існує три основних типи вайрфреймів: Low-Fidelity Wireframe, Middle-Fidelity Wireframe та High-Fidelity Wireframe. Найважливішим відмінним фактором між цими каркасами є кількість деталей, які вони містять.

Low-Fidelity Wireframe

Низькорівневі каркаси дизайну є основними візуальними зображеннями додатку і зазвичай служать початковою точкою дизайну. Вони, як правило, досить грубі, створюються без відчуття масштабу, сітки або точності пікселів. Низькорівневі каркаси опускають будь-які деталі, які потенційно можуть відволікати увагу, і включають лише спрощені зображення, форми блоків та фіктивний вміст, наприклад текст заповнення для міток та заголовків.

Low-Fidelity Wireframes корисні для початку розмов, прийняття рішення щодо розташування навігації та відображення потоку користувачів. Коротше кажучи, каркаси з низькою точністю ідеально підходять, якщо є зацікавлені сторони або клієнти, які хочуть власноруч прийняти участь у створенні дизайну. Вони також неймовірно корисні для дизайнерів, які мають кілька концепцій продуктів і хочуть швидко вирішити, в якому напрямку рухатися далі.

Найпершим екраном додатку буде «початковий екран» (рис. 3.1), де розташуються назва екрану під номером 2, кнопка для додання задачі за номером 1, кнопка для редагування задач (номер 3) та, власне, список задач – 4.

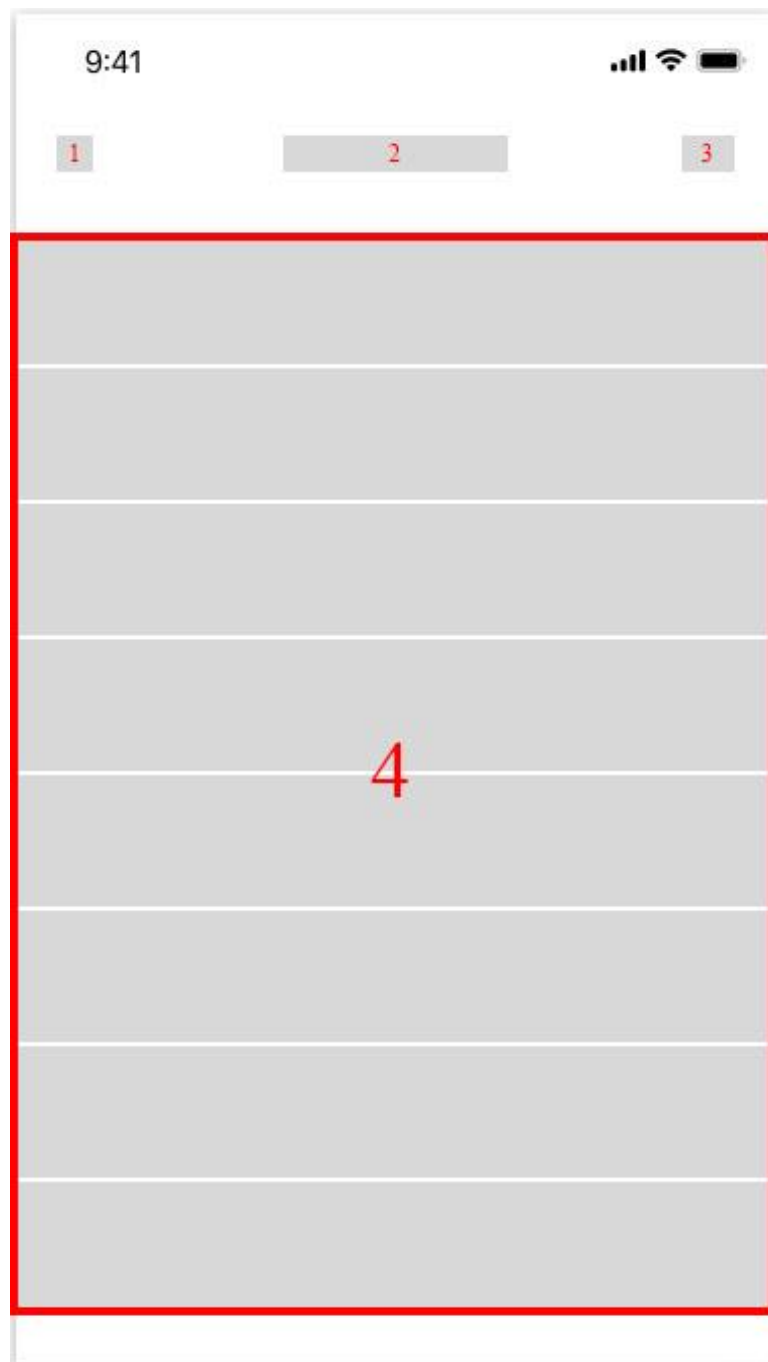


Рис. 3.1. Lo-Fi-WF сторінки «Початковий екран»

Існуючі задачі треба переглядати, щоб побачити не тільки їх назву, але й додаткову інформацію, яку зможе написати користувач. Для цієї мети треба додати другий екран додатку (рис. 3.2), на який користувач зможе потрапити, зробив так званий «тап» по задачі (розташованій у блоку під номером 4), інформацію про яку треба переглянути. Також, було б доцільно розмістити кнопку, завдяки якій задача буде позначатися, як виконана.

Отже, під номером 1 буде розташована системна кнопка «Back», завдяки котрій можна буде повернутися на «початковий екран»; другим номером позначено поле, де буде назва задачі; під номером 3 позначено поле, де буде розташована дата, до якої треба цю задачу виконати; поле під номером 4 відповідає за розміщення додаткової інформації про задачу; і останнє 5 поле слугує для кнопки «Complete», нажавши яку, користувач помітить задачу, як виконану.

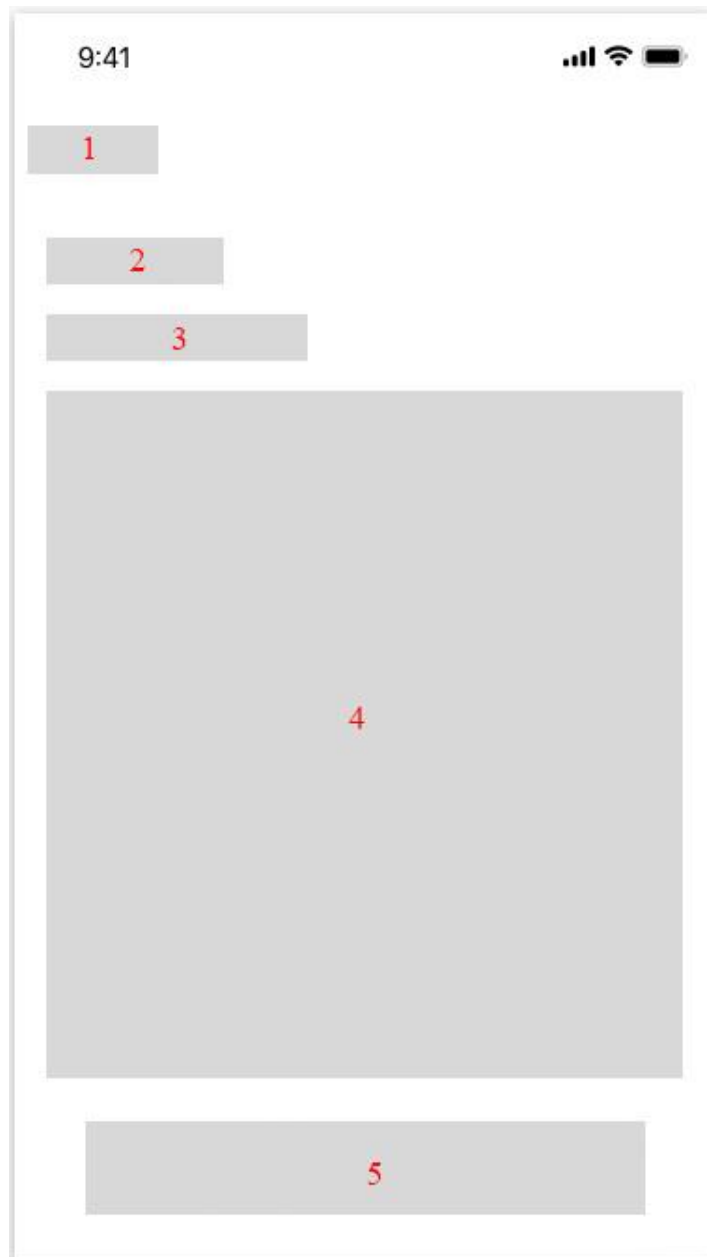


Рис. 3.2. Lo-Fi-WF сторінки «Інформація про задачу»

Останнім екраном для додатку стане екран «створення задачі», він буде слугувати для, власне, створень користувачем нових задач. Під номером 1 буде надпис «Task name», щоб користувач розумів що треба вписувати у поле під номером 2, де, власне, треба буде просто написати назву задачі. Номер 3 теж є простим надписом «Task Details», яка буде вказувати на те, що у поле під номер 4 треба написати деталі завдання. П'ятий номер, як 1 та 3, – це назва «Task Completion Date», що вказує на поле під номером 6, де можна буде обрати дату та час виконання завдання. І останній сьомий номер вказує на кнопку для додання задачі до списку задач.

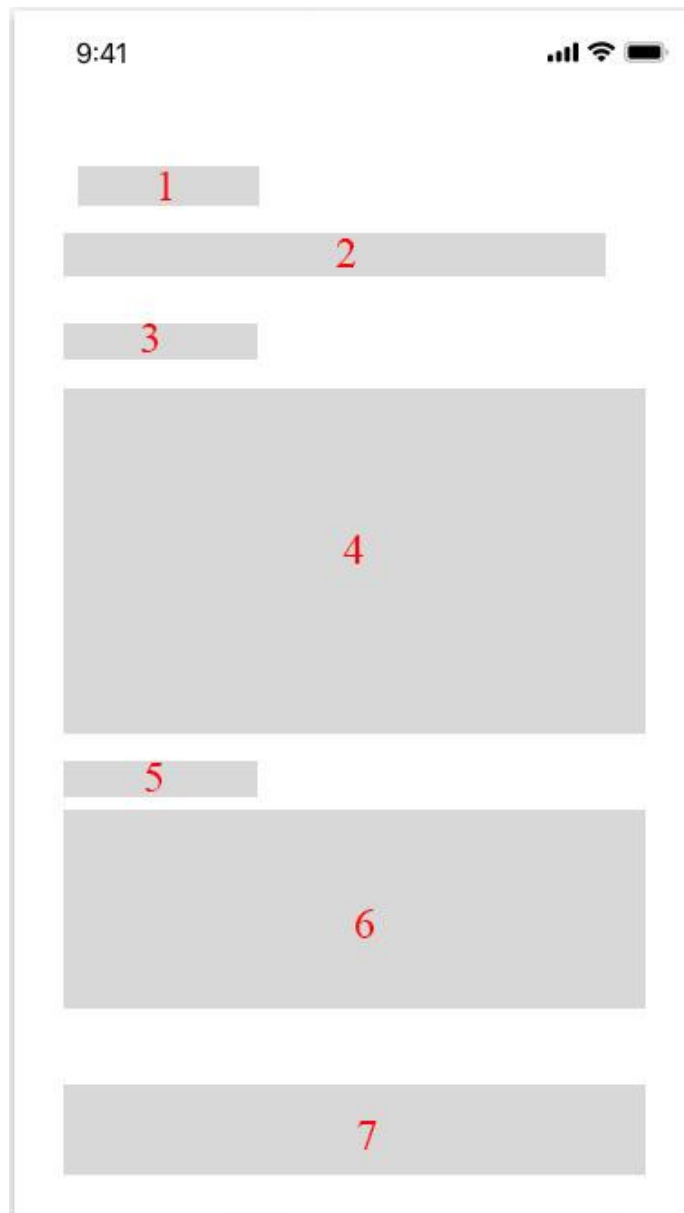


Рис. 3.3. Lo-Fi-WF сторінки «Створення задачі»

Middle-Fidelity Wireframe

Найбільш часто використовуваний каркас із трьох каркасів – це середньорівневий каркас дизайну, він характеризується більш точним відображенням макета. Незважаючи на те, що вони все ще уникають відволікаючих факторів, таких як зображення або типографіка, більше деталей призначається конкретним компонентам, а функції чітко диференціюються між собою. Різні ваги тексту можуть також використовуватися для розділення заголовків та вмісту тексту. Хоча вони все ще чорно-білі, дизайнери можуть використовувати різні відтінки сірого для передачі візуальної яскравості окремих елементів. Незважаючи на те, що вони все ще актуальні на ранніх стадіях продукту, Mi-Fi-WF зазвичай створюються з використанням цифрового інструментарію, такого як Sketch або Figma.

High-Fidelity Wireframe

Високорівневі каркаси дизайну можуть похвалитися макетами більш високої якості. Там, де низькорівневий дизайн каркасу може включати псевдолатинські текстові наповнювачі та сірі рамки, заповнені знаком «X» для позначення зображення, високорівневі каркаси дизайну можуть включати фактичні образи та відповідний письмовий вміст. Ця додаткова деталь робить Hi-Fi-WF ідеальними для вивчення та документування складних концепцій, таких як системи меню або інтерактивні карти. На останньому етапі циклу проектування додатку слід зберігати високоякісні каркаси.

Настав момент для створення High-Fidelity Wireframes. Від низькорівневих каркасів дизайну вони будуть відрізнятися вже більш чіткими полями, які зрозуміє будь-яка людина.

Високорівнений каркас дизайну «початкового екрану» зображено на рис. 3.4. На ньому вже видно іконки, назви полів та призначення самого екрану. Якщо порівнювати його з попереднім каркасом, то дуже легко помітити, що замість поля під номером 1 з'явилась іконка «+», яка натякає користувачу, що нажавши її, можна створити нову задачу.

У поле з назвою екрану було вирішено написати «To Do List», адже це цілком відображає всю суть цього екрану. Далі, з'явилась кнопка «Edit», яка своєю назвою

говорить користувачу, що треба натиснути саме її, щоб видаляти задачі зі списку. Також, у полі з задачами було вирішено залишити тільки назву задачі та її статус (виконано чи не виконано).

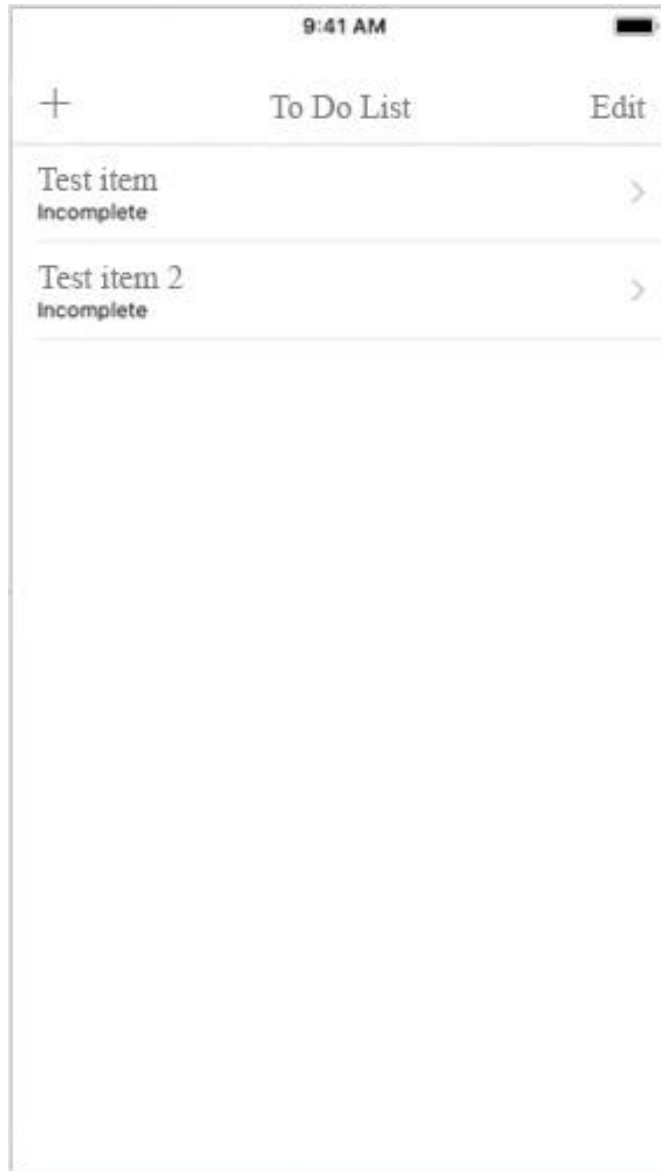


Рис. 3.4. Hi-Fi-WF сторінки «Початковий екран»

Наступним екраном є «інформація про задачу» (рис. 3.5). Поле під номером 1 перетворилося на системну кнопку «Back», щоб була можливість повернутися на попередній екран. Поле 2 буде кожний раз мінятися, в залежності від того, що туди впише користувач. Третє поле відповідає за срок виконання задачі. Поле «Details», як і друге, кожен раз буде мінятись, адже користувач сам записує туди деталі задачі,

яку він створив. І, нарешті, останнє поле під номером 5, набуло вигляду кнопки, яку треба натиснути щоб помітити задачу, як виконану.



Рис. 3.5. Ні-Fi-WF сторінки «Інформація про задачу»

Останній екран під назвою «створення задачі» (рис. 3.6) також потерпів ряд змін. Наприклад, у полі номер 1 було вирішено написати «Task name», щоб користувач розумів, що у полі номер 2 треба написати назву задачі, що створюється. Поля під номерами 3 та 4 також зв'язані між собою, адже третє поле слугує для визначення функції полю номер чотири. П'яте своєю назвою вже вказує користувачу, що потрібно зробити та для чого слугує поле номер 6. І останнє, сьоме

поле набуло вигляду кнопки, поки що без кольору, але вже стало очевидним її призначення.

The screenshot displays a mobile application interface for creating a task. At the top, the status bar shows the time as 9:41 AM. The main content area is divided into several sections: a 'Task name' section with a text input field; a 'Task details' section; a 'Task Completion Date' section featuring a date and time picker with three rows: 'Sun Dec 13 1 38 AM', 'Mon Dec 14 2 39 PM' (highlighted), and 'Tue Dec 15 3 40'; and a large grey button labeled 'Add Task' at the bottom.

Рис. 3.6. Hi-Fi-WF сторінки «Створення задачі»

Очевидно, що скільки функцій включено в каркас – багато в чому залежить від того, чи є каркас низькорівневим, середньорівневим або високорівневим. Однак елементи, які зазвичай можна знайти у вайрфреймах, включають логотипи, поля пошуку, заголовки, кнопки спільного доступу та псевдолатинський текст заповнювача. Hi-Fi-WF можуть також включати навігаційні системи, контактну інформацію та колонтитули. Типографіка та зображення не повинні бути частиною низькорівневого або середньорівневого каркасів дизайну, але дизайнери часто

грають із розміром тексту, щоб представити ієрархію інформації або вказати заголовок.

Вайрфрейми традиційно створюються у відтінках сірого, тому дизайнери часто граються із затіненням, використовуючи світлі відтінки сірого для представлення світлих кольорів, а темніші – для більш сміливих кольорів. У Hi-Fi-WF, дизайнери можуть використовувати кольори, що будуть у фінальній концепції дизайну; наприклад, червоний для позначення попередження або повідомлення про помилку, або темно-синій для позначення активного посилання. Оскільки вайрфрейми є двовимірними, важливо пам'ятати, що вони погано справляються з показом інтерактивних функцій інтерфейсу, таких як випадаючі меню, стану наведення або функцій show-hide.

3.1.2. Mockup

По суті, Mockup – це лише ескіз, який відображає вигляд додатку чи сайту. Головна мета мокапів – це надати змогу дизайнеру при найменших затратах часу та зусиль, отримати найбільш чіткий, зрозумілий та корисний прототип додатку чи веб-сайту, який він збирається будувати.

Ми живемо у світі, де потрібно рухатись швидко і якомога гнучкіше повторювати свої ідеї. Створення мокапів забезпечує ідеальний баланс між реальним представленням кінцевого продукту та простотою модифікації. Дизайнер может створити макет додатку всього за кілька хвилин, використовуючи такі інструменти, як Balsamiq, Photoshop, Sketch, Figma, Axure та інші, що займає набагато менше часу, ніж якщо малювати його вручну.

Макети корисні для творчої фази проекту – наприклад, коли ви намагаєтеся з'ясувати поведінку користувачів або належної візуальної ієрархії, – так і для фази виробництва, коли вони представлятимуть цільовий продукт.

Якщо зробити макети частиною творчого процесу та процесу розробки, команда розробників зможе швидко та легко створювати ідеї та ітерації у своєму додатку, перш ніж витратити дуже багато часу на процес кодування. У вас буде

щось матеріальне, щоб показати потенційним клієнтам, або корисна довідка для роботи, коли ви почнете рухатися вперед.

Невелика затрата часу на освоєння та впровадження цієї навички у проект, врятує багаторазові витрати в майбутньому.

Макети – це не якийсь галузевий стандарт, який потрібно запам'ятати і дотримуватися – існує стільки різних способів впровадження дизайнів у додаток, скільки дизайнерів та розробників у світі. Тим не менш, поширеним способом виявлення «грубості» макетів є використання терміна «fidelity».

Fidelity – це міра того, наскільки макет наближений до кінцевого продукту. Макет з низькою точністю – це в основному простий ескіз (рис. 3.7):

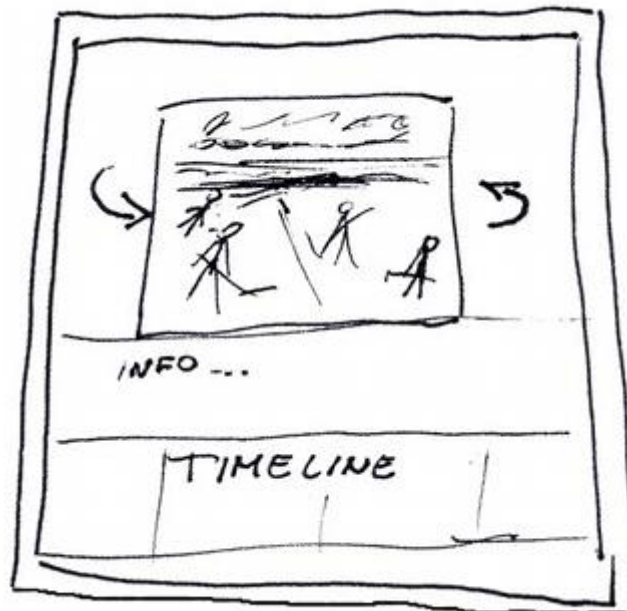


Рис. 3.7. Приклад ескізу проекту

Макет високої точності, як зображено на рисунку 3.8, - це, в основному, ідеальні зображення остаточного сайту:



Рис. 3.8. Приклад макету високої точності

У цій роботі ми будемо перетинати межу між ними, прагнучи створити макети, які є досить точними у побудові, але досить простими для швидкого створення.

Wireframes vs Mockups vs Prototypes

Варто прояснити ще одну загальну точку плутанини. Люди часто використовують поняття «Wireframe», «Mockup» та «Prototype» як взаємозамінні, але вони представляють зовсім різні речі:

- Як зазначалося раніше, Wireframe насправді має бути структурним представленням додатку;
- Mockup доповнює статичний візуальний вигляд додатку (наприклад, містить більше деталей зображення та кольорів);
- Прототип спеціально призначений для відображення взаємодії користувача з цим дизайном (тому він уже не є справді статичним).

Треба зауважити, що ці визначення насправді не пов'язані безпосередньо з fidelity – можна мати прототип із дуже низькою точністю, який просто визначає ключові взаємодії користувачів або фрейм із високою точністю, який містить кожен елемент сторінки.

У цій роботі будуть трохи розмиті лінії між макетами та прототипами – більшість із того, що ми виробимо та використаємо, будуть статичними поданнями кінцевого продукту (класичний випадок макету), але ми також визначатимемо взаємодію користувача з дизайном, який починає трохи більше переносити його на територію прототипу.

Де використовуються мокапи

Одна з найприємніших частин створення додатків – це взяти ідею з голови та побачити, як вона оживає під час творчої фази. За допомогою макетів можна зробити це неймовірно швидко, і, також, можна переглядати нові дизайни або потоки всього за кілька кліків (замість того, щоб змінювати значні частини розмітки або коду).

Побудова процесу розробки додатку:

- Визначення користувача та цілі додатку;
- Побудова макету додатка, який відповідає цим цілям;
- Розбиття макету на окремі фрагменти роботи;
- Для кожного фрагменту треба починати з макету його кінцевого результату (наприклад, конкретної функції) та будувати інфраструктуру даних, back end та front end, необхідні для його фактичного створення;
- Перевірка, рефакторинг та передача.

Отже, макет не тільки дає загальний напрямок та масштаб додатку, але й розбиває його на частини та використовує ці фрагменти для керівництва процесом розробки.

Москір першого «Початкового екрану» (рис. 3.9), порівнюючи з фреймів, затерпів не дуже багатьох змін, – було прийнято рішення зробити весь текст чорним кольором на білому фоні, а активні кнопки у стилі операційної системи iOS – голубими.

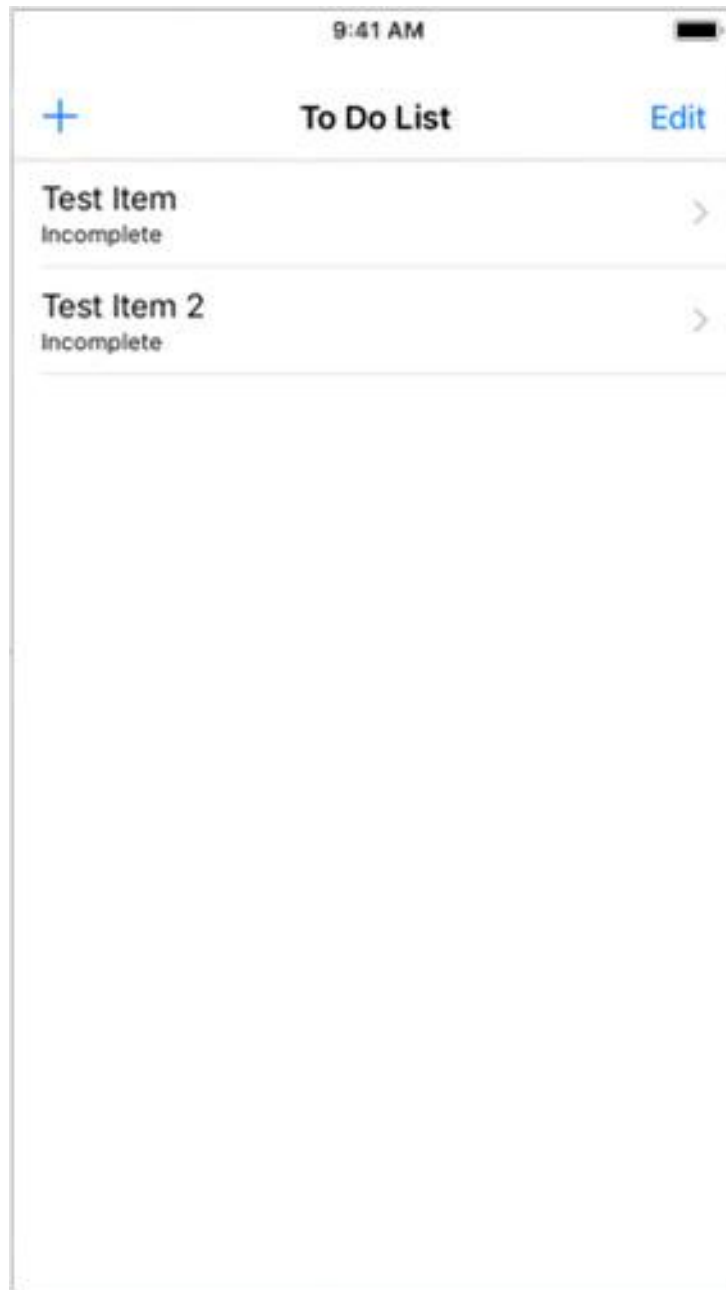


Рис. 3.9. Мокер сторінки «Початковий екран»

Приблизно така ж ситуація і з екраном «інформація про задачу» (рис. 3.10), текст чорного кольору на білому фоні, кнопки – голубі, але кнопку «Complete» було вирішено зробити чорною з білим надписом, що дуже красиво гармонує з загальним дизайном додатку.



Рис. 3.10. Мокер сторінки «Інформація про задачу»

Мокер екрану «створення задачі» (рис. 3.11) було вирішено зробити по прикладу попереднього екрану, адже це дуже добре вписується у загальну концепцію додатку – простоту і зручність у розумінні та використанні.

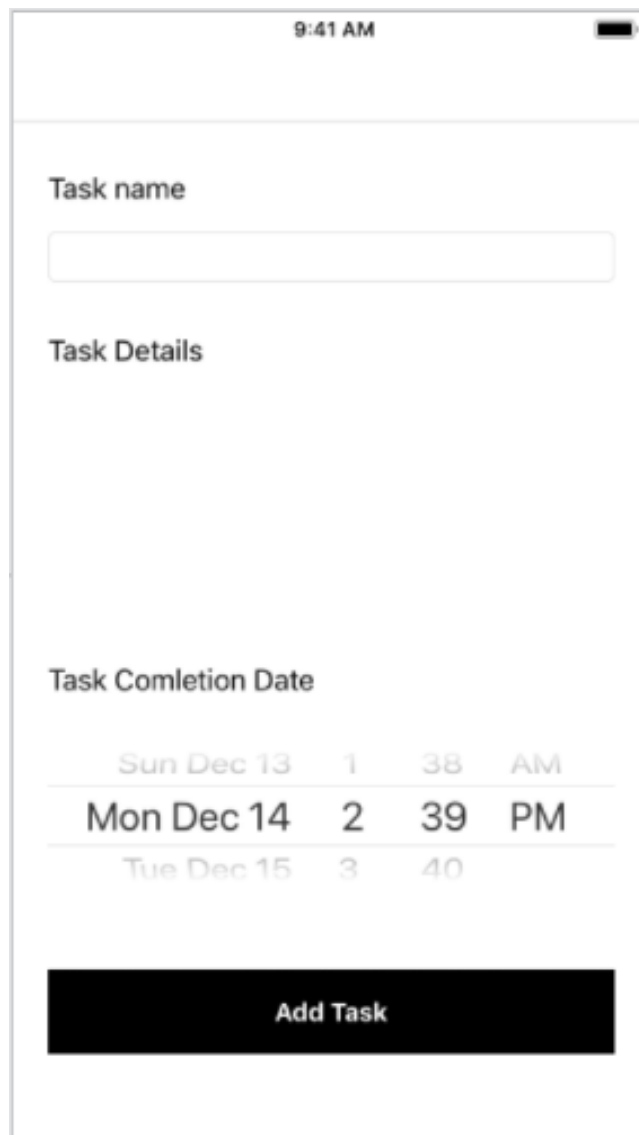


Рис. 3.11. Мокер сторінки «Створення задачі»

Дизайн інтерфейсу користувача є невід’ємною частиною успіху. Гарний дизайн не може врятувати незграбний та незрозумілий для навігації інтерфейс, а чудовий, цілком доречний користувальницький досвід може бути потоплений поганим дизайном візуального інтерфейсу, що робить використання програми неприємним. Дизайн UI та UX повинен бути бездоганно виконаний та ідеально узгоджений із існуючими очікуваннями користувачів, щоб створити чудовий користувальницький інтерфейс та досвід. І коли ці зірки співпадають, результати можуть бути вражаючими, у чому ми й переконались, створивши власні Low-Fidelity Wireframes, High-Fidelity Wireframes та Mockups.

3.2. Розробка додатку

«У першу чергу, треба вирішити яка у додатку буде архітектура. Компанія Apple рекомендує використовувати паттерн MVC, – це спосіб організації коду, який передбачає виділення блоків, що відповідають за вирішення різних завдань. Один блок відповідає за дані у додатку, інший відповідає за зовнішній вигляд, а третій контролює роботу додатка.»[10]

Компоненти MVC:

- «Model (модель) – цей компонент відповідає за дані, а також визначає структуру програми. Наприклад, якщо створювати To-Do додаток, код компонента model визначатиме список завдань і окремі завдання.»[11]
- «View (уявлення) – цей компонент відповідає за взаємодію з користувачем. Тобто код компонента view визначає зовнішній вигляд програми і способи його використання.»[11]
- «Controller (контролер) – цей компонент відповідає за зв'язок між model і view. Код компонента controller визначає, як додаток реагує на дії користувача. По суті, це мозок MVC-додатків.»[11]

Далі треба створити проект у середовищі розробки Xcode. Для цього треба запустити Xcode, обрати «створити новий проект» та натиснути «далі». Після цього, ми потрапимо у меню з вибором типу проекту (рис. 3.12).

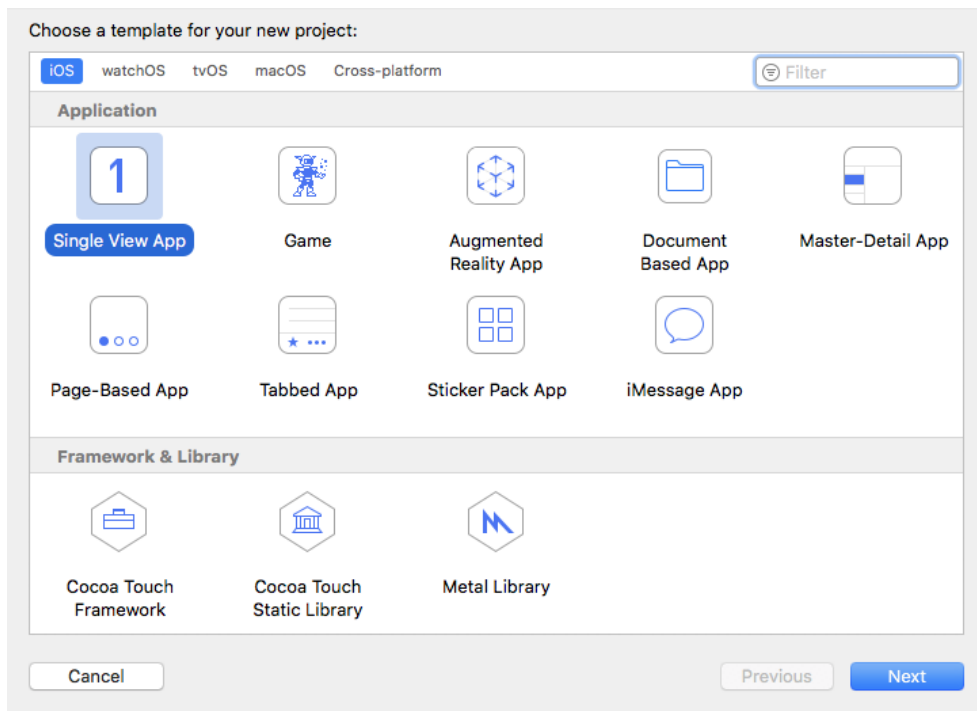


Рис. 3.12. Меню вибору типу проекту

У меню, що зображено на рис. 3.12, є дев'ять типів проектів. З назв стає очевидно для чого кожний з них потрібен, але ми оберемо «Single View App». Після вибору проекту треба натиснути «Next» і заповнити меню з назвою продукту (рис. 3.13).

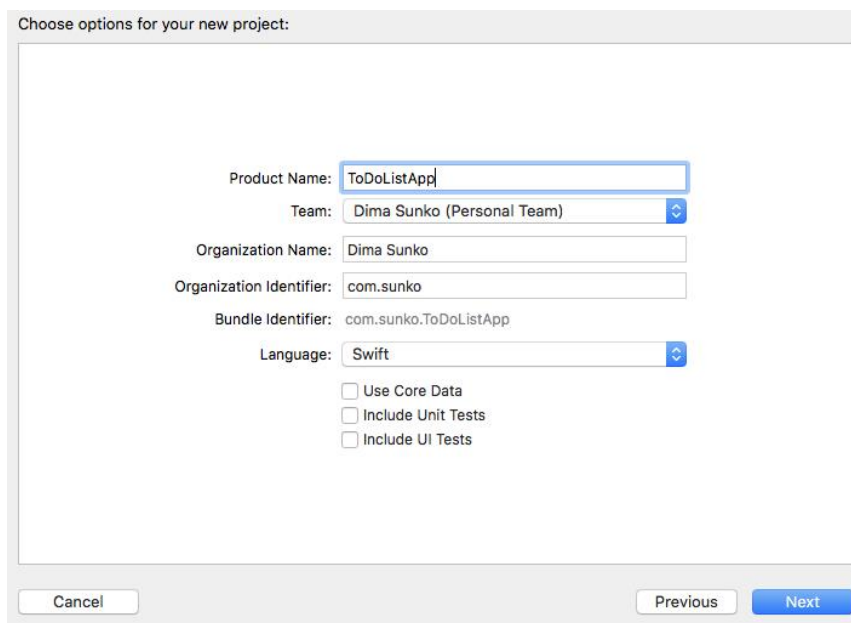


Рис. 3.13. Меню з назвою продукту

Тут треба указати: назву додатку; назву організації, що розробляє додаток та мову програмування. Зробивши ці кроки, відкривається порожній проект, у якому і будуть створюватись Model, View та Controller.

3.2.1. Модель (Model)

Перше, що треба розробити – це модель додатку, тобто його структуру. Для цього, у проекті в Xcode, треба створити новий файл з назвою «ToDoItemModel».

У файлі створимо так звану структуру за допомогою ключового слова «struct», дамо їй назву «ToDoItem». У структурі треба створити декілька змінних, а саме: «name» типу «String», що буде відповідати за назву задачі, яку створив користувач; «details» типу «String», що відповідає за додаткову інформацію задачі, що створив користувач; «completionDate» типу «Date», яка потрібна, щоб записувати дату завершення задачі; «startDate» типу «Date», що відповідає за дату створення задачі; «isComplete» типу «Bool», яка слугує тригером виконання задачі.

Потім треба провести ініціалізацію створених змінних. Ініціалізація – це підготовчий процес екземпляра класу, структури або перерахування для подальшого використання. Цей процес включає в себе установку початкових значень для кожної властивості зберігання цього екземпляра і проведення будь-яких налаштувань або ініціалізації, які потрібні до того, як екземпляр використовуватиметься. У мові Swift це робиться завдяки оператору «init». Фінальний вигляд моделі додатку зображено на рис. 3.14.

```

9  import Foundation
10
11  struct ToDoItem {
12
13      var name: String
14      var details: String
15      var completionDate: Date
16      var startDate: Date
17      var isComplete: Bool
18
19      init(name: String, details: String, completionDate: Date) {
20
21          self.name = name
22          self.details = details
23          self.completionDate = completionDate
24          self.isComplete = false
25          self.startDate = Date()
26
27      }
28
29  }

```

Рис. 3.14. Модель додатку

3.2.2. Уявлення (View)

Наступним етапом у розробці програми буде створення view – інтерфейсу користувача. У першу чергу, треба створити так звані «view», що будуть відображати суть різних екранів додатку. У середовищі Xcode є дуже зручний інструмент, що має назву «Storyboard». З його допомогою, можна створити «уявлення» не прибігаючи до написання коду; треба просто створити потрібну кількість екранів, за допомогою конструктора додати потрібні елементи та зв'язати все екрани між собою.

Почнемо зі створення екрану «To Do List View Controller». Для початку потрібно зайти у файл «Main.storyboard», там, за допомогою бібліотеки об'єктів, треба знайти та створити «View Controller». До проекту додасться новий «view», що і є нашим «уявленням» (рис. 3.15).

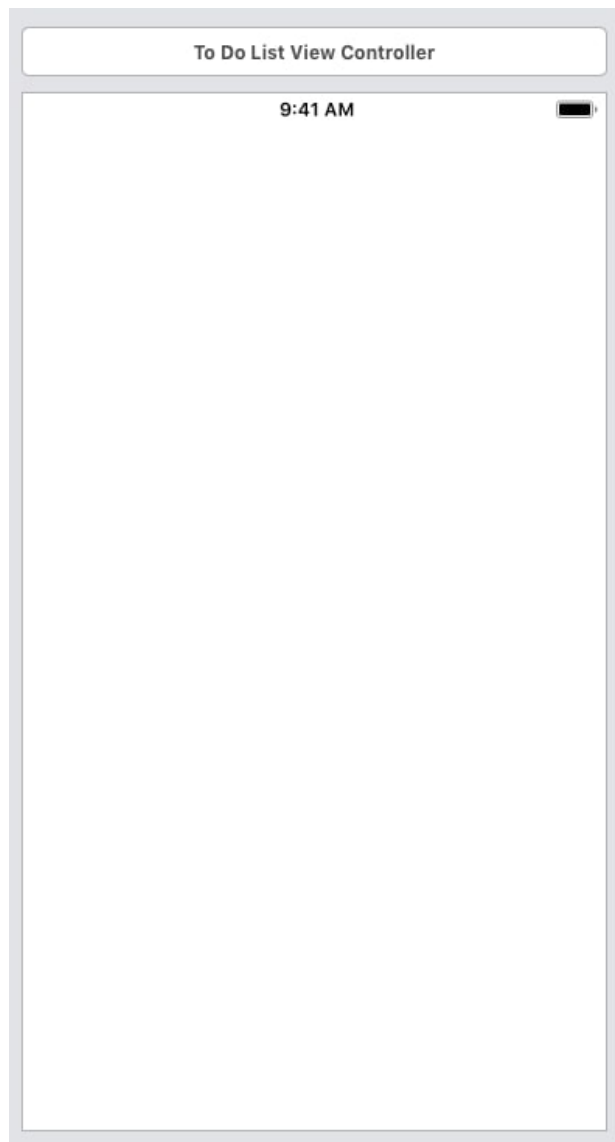


Рис. 3.15. Нове «View»

Звичайно, «уявлення» створюється порожнім та до нього треба поступово додавати нові елементи. На цьому екрані будуть відображатися всі задачі, що створює користувач, у вигляді таблиці елементів. Реалізується це за допомогою так званого «Table View», що дозволяє робити невеликі окремі осередки, які відображатимуть кожна свій власний контент. Також необхідно додати «Navigation Item», що дозволить зробити навігацію між різними екранами додатку, та «content View», який є структурою осередків у «Table View». Готовий екран «To Do List View Controller» зображено на рис. 3.16.

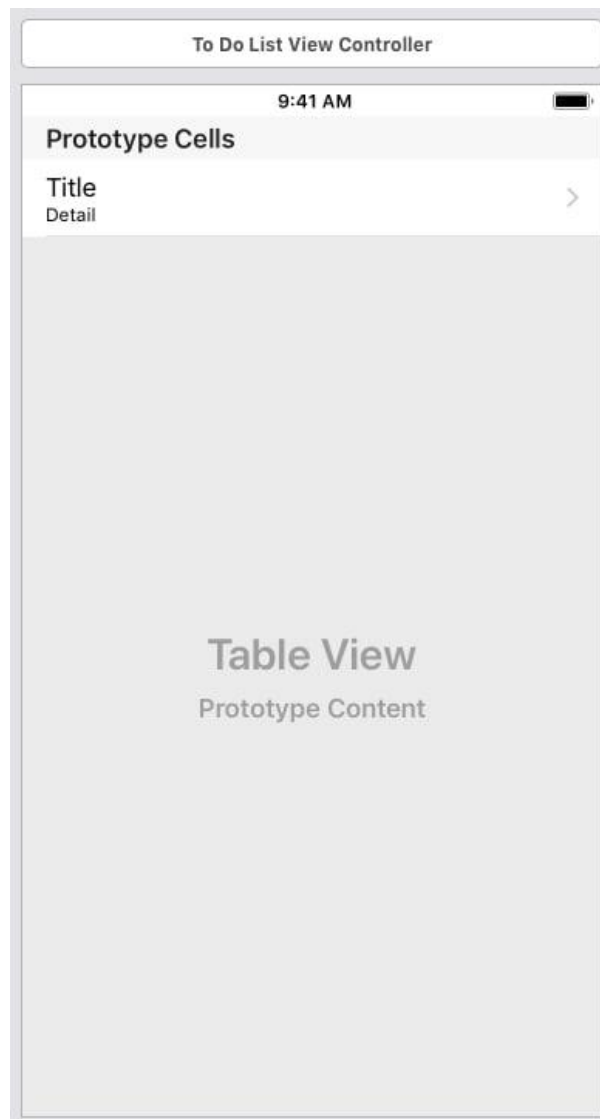


Рис. 3.16. Готовий екран «To Do List View Controller»

На рисунку 3.16, видно, що кожний окремий осередок буде мати назву (Title), невелику кількість деталей (Detail), та стрілку зправа, яка зветься «Navigation Item», яка буде натякати користувачу, що можна натиснути на неї та перейти до меню окремого осередку (задачі).

Наступним буде екран «To Do Details View Controller», який буде відповідати за те, щоб користувач мав змогу подивитись на деталі створеної задачі та позначити її виконаною. Для його створення, як і у минулий раз, треба створити нове «уявлення» та наділити його елементами. Тут нам знадобляться такі елементи, як: «Label», що послугує назвою створеної задачі та відображенням дати кінця задачі; «Text View», який потрібен для відображення великої кількості тексту, в ньому

розташується детальна інформація про створену задачу; «Button» – кнопка, яка буде слугувати тригером завершення задачі. Тож фінальна версія цього «уявлення» зображена на рис. 3.17.

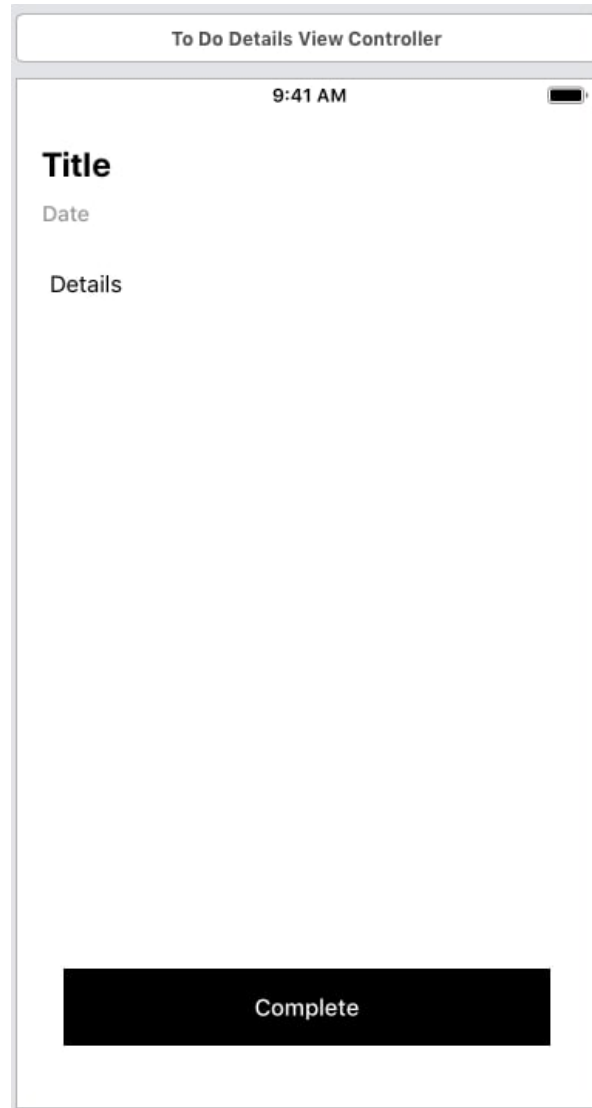


Рис. 3.17. Екран «To Do Details View Controller»

Але у програмуванні під операційну систему iOS є багато особливостей і одна з них – констрейнти. Constraints визначають взаємозв'язок між двома елементами інтерфейсу користувача. Ці елементи можуть представляти або «уявлення», або інші атрибути. Констрейнти можуть також визначати взаємозв'язок між двома різними атрибутами одного елемента, наприклад, встановлюючи співвідношення сторін між

висотою та шириною елемента. Також можна призначити постійні значення висоті або ширині елемента.

Взагалі, констрейнти потрібні для зручного та швидкого проектування «уявлень». Майже кожний девайс компанії Apple має свою власну висоту екрана, ширину та особливості (наприклад екран iPhone X не має рамок). Буде дуже складно та довго адаптувати всі елементи на екрані для кожного розміру екрану, тож компанія Apple придумала таку зручну річ, як констрейнти. За їх допомогою, можна створити «уявлення» зі всіма елементами всього один раз, а система автоматично адаптує розміри та пропуски між елементами для всіх розмірів та типів екранів, що є у компанії Apple.

На рисунку 3.18 можна побачити констрейнти екрану «To Do Details View Controller».

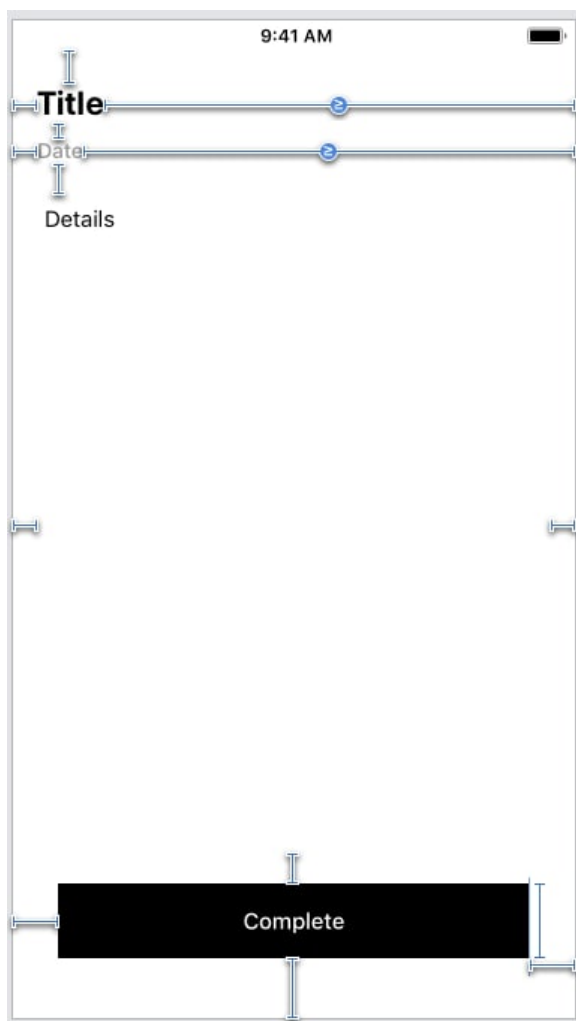


Рис. 3.18. Констрейнти екрану «To Do Details View Controller»

Кожна синя лінія – це окремий констрейнт, що визначає де буде розташований то чи інший елемент. Наприклад констрейнти кнопки «Complete» кажуть, що ця кнопка розташована за 30 пікселів від лівого боку, 30 від правого, 40 від низу екрану, 20 пікселів до «Text View» та висота кнопки дорівнює 50 пікселям. Тож незалежно від розмірів екрану, ці відстані не зміняться. Таким же чином побудовані й інші елементи.

Наступник кроком у створенні додатку буде проектування екрану «Add Task View Controller». У ньому буде можливість створити нову задачу. Перш за все, треба створити новий «View Controller» та назвати його «Add Task View Controller», потім розташувати на ньому необхідні для роботи елементи. У цього «уявлення» будуть такі елементи: «Label», який буде незмінний та слугувати для того, щоб показати користувачу для чого потрібні інші елементи; «Text Field» – поле для вводу тексту, конкретно у цьому випадку існує для вводу найменування задачі; «Text View» – вже знайомий елемент, який тут буде використовуватись для вводу тексту про деталі задачі; «Date Picker», який потрібен для того, щоб користувач зміг обрати дату та час виконання задачі; «Button» – кнопка, що буде використовуватись для додання задачі до списку задач; «Navigation Bar» – спеціальний елемент, за допомогою якого, можна буде повернутися до попереднього екрану, якщо це потрібно (не створюючи нову задачу). На рисунку 3.19 зображено екран «Add Task View Controller».

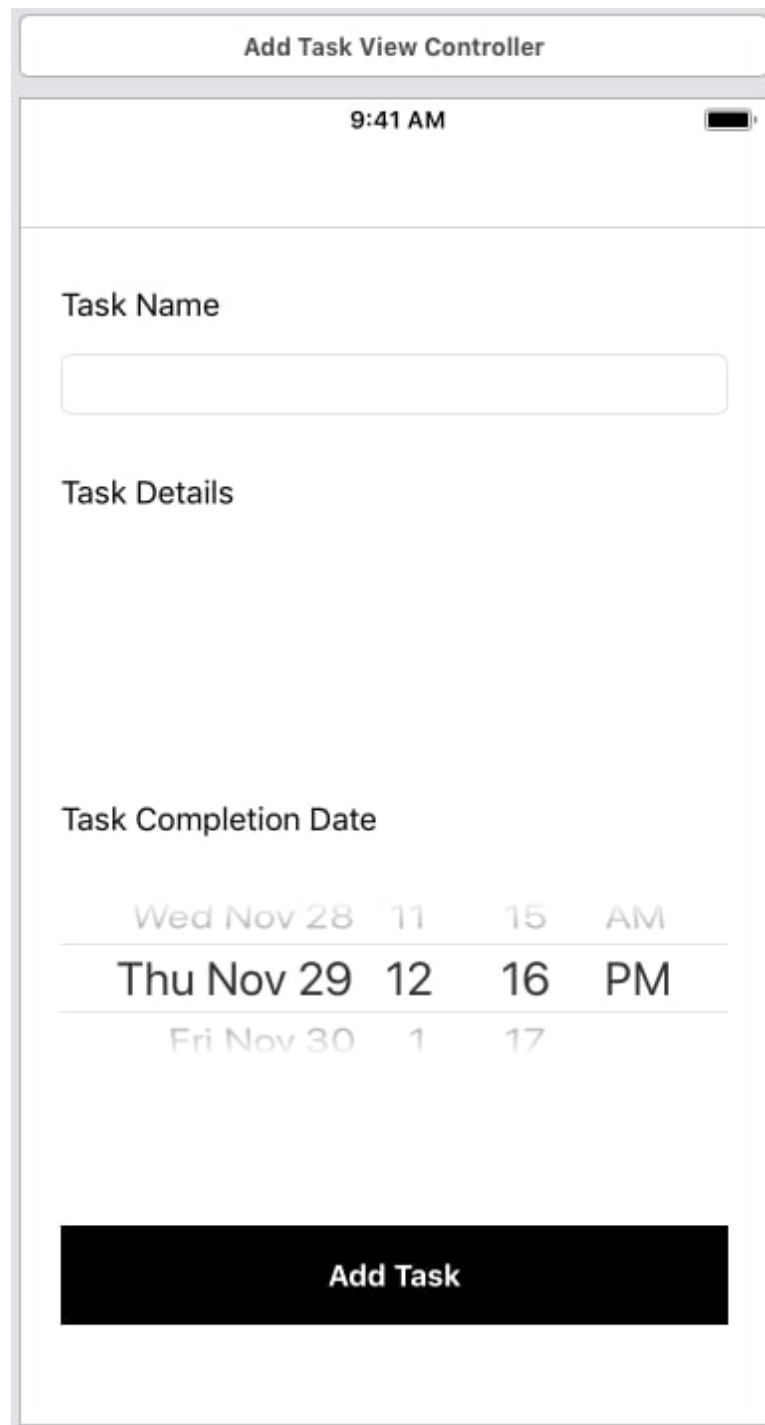


Рис. 3.19. Екран «Add Task View Controller»

Також ніколи не треба забувати і про констрейнти, адже вони є дуже важливою складовою створення висококласних додатків на базі iOS. На рис. 3.20 зображено майже всі констрейнти екрану «Add Task View Controller».

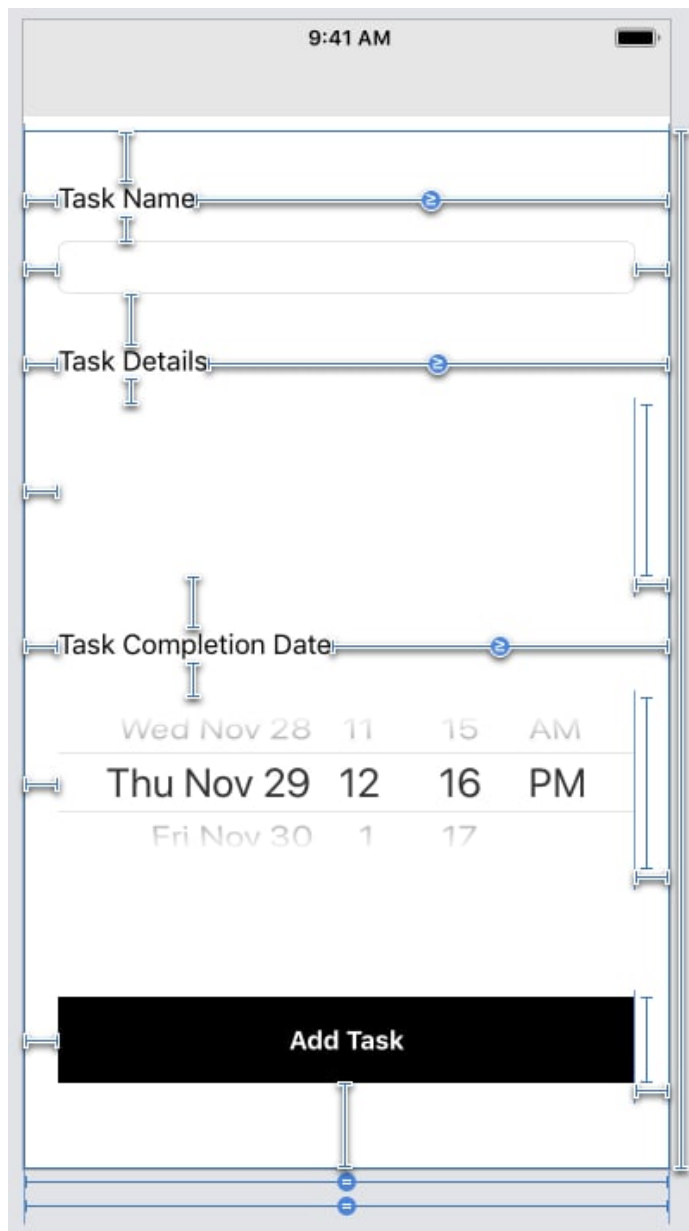


Рис. 3.20. Констрейнти екрану «Add Task View Controller»

Наприклад, констрейнти надпису «Task Completion Date» можуть сказати, що лейбл має 20 пікселів відступу між лівим краєм екрану, 20 пікселів до «Date Picker», 30 пікселів до «Text View». Але констрейнт до правого краю екрану виглядає незвичайно і це неспроста, адже у нього задана властивість «Greater Than or Equal» та константа 10 пікселів. Це означає, що елемент не зможе вийти за рамки екрану, якщо текст у ньому буде занадто довгий. Замість цього, текст елемента зупинить своє зростання за 10 пікселів до краю екрану та поставить три крапки, які будуть означати, що текст продовжується, але не поміщається на екран.

3.3. Висновки до третього розділу

Отже, розробка додатків поділяється на багато етапів, – аналітика, в яку входить аналіз потреб користувачів та ринку; технічне завдання, в яке входить опис функціональності майбутнього додатку; проектування та дизайн, в які входить розробка UI та UX дизайну; розробка додатку, в яке входить написання коду, його оптимізація та відлагодження; тестування; публікація додатку в існуючі цифрові платформи; підтримка та розвиток.

В третьому розділі було приділено увагу та висвітлено два основних етапи розробки – проектування та дизайн, і розробка додатку. Розробка UI та UX дизайнів дуже важлива, адже завдяки їм можна відразу зробити висновок чи буде додаток корисним та зручним для кінцевого користувача та його потреб. Тому у цьому розділі була проаналізована поведінка користувачів та розроблено дизайн за всіма вимогами UI/UX для операційної системи iOS.

Але не менш важливу роль відіграє саме розробка додатку, тобто написання коду, його оптимізація та відлагодження. Було проаналізовано різні способи реалізації архітектури додатку і обрано найкращий та найзручніший – MVC. Також були вивчені та використані інструменти для розробників додатків під платформу iOS, а саме: Autolayout, Xcode, Constraints, Interface Builder та інші. І завдяки ним було розроблено повністю функціонуючий додаток.

ВИСНОВКИ

Отже, проаналізувавши значну кількість різних типів додатків, стає очевидно, що всі вони відрізняються своєю суттю, ціллю, кількістю людей, зацікавлених в них, та, що не менш важливо, віком користувачів. Якщо в іграх, в основному, переважають діти віком від 2 до 16 років, то у соціальних мережах та месенджерах, основна аудиторія – це вже підлітки та дорослі люди, віком від 12 до 50 років. Таким чином, стає очевидно, що найбільш прибутковими додатками є саме соціальні мережі та месенджери, оскільки більшість аудиторії – платоспроможна та може собі дозволити купити додаткові товари всередині додатку. Але є і підводні камні, у вигляді великої конкуренції, тобто якщо спробувати зробити соціальну мережу, то, скоріш за все, вона нікому не буде цікава, адже і так існує дуже велика кількість різних мереж. Тож єдиний варіант зробити корисний для людей та прибутковий додаток – це розробити щось для дорослих, наприклад – список задач прямо у смартфоні.

Після того, як визначилися з напрямком додатка, треба обрати мову для програмування цього додатку. При розробці на мобільні платформи завжди є вибір між трьома типами додатків, – нативні, веб-додатки та гібридні. Оскільки нативні додатки мають набагато більше позитивних сторін, було вирішено взяти саме них. Але це ще не все, також треба обрати мову програмування, тож у другому розділі ми розглянули дві найбільш популярні мови для операційної системи iOS. Було вирішено обрати молодшу мову Swift, адже вона перевершує свого прямого конкурента (Objective-C) у простоті написання коду, швидкості та в багатьох інших задачах. Також у другому розділі ми проаналізували та розглянули різні сервіси для створення UI/UX дизайну, та було вирішено зупинитись на сервісі «Scetch».

Проаналізувавши ринок додатків, потреби користувачів та майже всі аспекти створення додатків, третій розділ був присвячений практичним діям по розробці додатку на операційну систему iOS. Таким чином, було створено три екрани, які повністю реалізують всі задумані функціональні аспекти додатку та відповідають вимогам UI/UX дизайну для платформи iOS.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Казанський О.П. Розробка додатків на Swift 5.1 і SwiftUI / Казанський О.П. : Вид-во БХВ-Петербург 2021. – 320с. : іл. Бібліогр.: с. 120-300.
2. Грехем Ли. Розробка через тестування для iOS / Грехем Ли. ; Вид-во ДМК Пресс 2016. – 272с. : іл. – Бібліогр.: с. 90-205.
3. Statista. [Електронний ресурс] режим доступу: <https://www.statista.com> (дата звернення 16.10.2021 р). – Назва з екрана.
4. Вікіпедія. [Електронний ресурс] режим доступу: <https://ru.wikipedia.org> (дата звернення 17.10.2021 р). – Назва з екрана.
5. Федотенко М.А. Розробка мобільних додатків / Федотенко М.А. : Вид-во Лабораторія знань 2019. – 335с. : іл. – Бібліогр.: с. 119-147.
6. Alexa. [Електронний ресурс] режим доступу: <https://alexa.com> (дата звернення 25.10.2021 р). – Назва з екрана.
7. SimilarWeb. [Електронний ресурс] режим доступу: <https://www.similarweb.com> (дата звернення 13.11.2021 р). – Назва з екрана.
9. SwiftBook. [Електронний ресурс] режим доступу: <https://swiftbook.ru> (дата звернення 25.11.2021 р). – Назва з екрана.
10. MacBug. [Електронний ресурс] режим доступу: <http://macbug.ru/cocoa/objc> (дата звернення 26.11.2021 р). – Назва з екрана.
11. Dou. [Електронний ресурс] режим доступу: <http://dou.ua> (дата звернення 26.11.2021 р). – Назва з екрана.