СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Навчально-науковий інститут (факультет) інформаційних технологій та електроніки

Кафедра інформаційних технологій та програмування

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної випускної роботи

освітній ступінь бакалавр

спеціальність 121 „Інженерія програмного забезпечення”

(шифр і назва спеціальності)

на тему «Розробка мобільного додатку Health&Sport з елементами гейміфікації»

Виконав: студент групи ІПЗ-20д _____ М.О. Шевченко_____

(підпис) (ініціали і прізвище)

Керівник _____ В.О. Лифар_____

(підпис) (ініціали і прізвище)

Завідувач кафедри _____ О.І.Захожай _____

(підпис) (ініціали і прізвище)

Рецензент Д.В. Ратов

Київ – 2024

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Навчально-науковий інститут (факультет) інформаційних технологій та електроніки

Кафедра інформаційних технологій та програмування

Освітній ступінь бакалавр

спеціальність 121 „Інженерія програмного забезпечення”

(шифр і назва спеціальності)

спеціалізація „Інженерія програмного забезпечення”

(назва спеціалізації)

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТП,

_____ Захожай О.І.

“____”_____2024 року

ЗАВДАННЯ
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ
Шевченко Микита Олександрович

(прізвище, ім’я, по батькові)

1. Тема роботи: Розробка мобільного додатку Health&Sport з елементами гейміфікації
керівник роботи: доцент, д.т.н. Лифар Володимир Олексійович_____
затверджені наказом вищого навчального закладу від “06”_травня 2024 року №171/15.15-С__
2. Строк подання студентом роботи : 08.06.2024 р.
3. Вихідні дані до роботи: Об'єктом даної роботи є процес розробки мобільного додатку
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
   4.1 Вступ
   4.2 Аналіз та вплив гейміфікації на сучасну людину

4.3 Основна частина, в якої висвітлити: Інформаційна модель об’єкту. Програмна реалізація моделі.

4.4 Висновки

4.5 Перелік використаних джерел

5. Перелік графічного матеріалу: немає

6. Дата видачі завдання: 29 лютого 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Одержання завдання на виконання роботи	29.03.24	виконано
2	Укладання і погодження з керівником плану і етапів виконання роботи	11.04.24	виконано
3	Узагальнення даних літературних джерел, укладання розділу «Аналіз предметної галузі»	20.04.24	виконано
4	Аналіз шляхів виконання завдання. Вибір і погодження з керівником оптимального шляху	01.05.24	виконано
5	Проектування інфологічної моделі задачі, що реалізується	15.05.24	виконано
6	Укладання та тестування програмного продукту	27.05.24	виконано
7	Укладання, оформлення та погодження пояснювальної записки з керівником	30.05.24	виконано
8	Укладання доповіді і презентації	07.06.24	виконано

Студент _____ М.О.Шевченко
(підпис) (ініціали та прізвище)

Керівник роботи _____ В.О.Лифар
(підпис) (ініціали та прізвище)

ЛИСТ ПОГОДЖЕННЯ І ОЦІНЮВАННЯ

дипломної роботи студента гр. ІПЗ-20д Шевченко М.О.

Науковий керівник

Доцент, д.т.н. _____ Лифар В.О.

Оцінка наукового керівника: _____

Рецензент _____
ПІБ, місто роботи, посада

Оцінка рецензента: _____

Кінцева оцінка за результатами захисту:

_____

Голова ЕК
Професор кафедри ІТП
д.т.н. _____ Меняйленко О.С.
підпис

РЕФЕРАТ

Робота містить: 46 сторінок основного тексту, 25 сторінок додатків, 11 рисунків, 19 використаних джерел.

Метою випускної кваліфікаційної роботи є розробка мобільного додатку Health&Sport з елементами гейміфікації для покращення фізичного здоров'я та стимулювання активного способу життя.

Для виконання цієї роботи було проведено такий аналіз:

Стан сучасних мобільних додатків з елементами гейміфікації для здоров'я та спорту: Досліджено існуючі мобільні додатки, які використовують гейміфікацію для стимулювання фізичної активності та здорового способу життя. Аналізувалися їх функціональні можливості, дизайн, рівень залучення користувачів та інші аспекти.

Потреби та очікування цільової аудиторії: Були вивчені потреби та очікування цільової аудиторії, зацікавленої у підтримці здорового способу життя та фізичної активності. Були проаналізовані їхні вимоги до мобільних додатків та популярні практики у гейміфікації.

Технічні можливості та обмеження: Розглянуто технічні аспекти розробки мобільних додатків, включаючи вибір платформи розробки, інструменти та технології, які можна використовувати для створення програми Health&Sport.

ЗМІСТ

ВСТУП

Актуальність досліджень. Ми часто чуємо про негативний вплив відеоігор: жорстокість, уникнення реальності, залежність та багато інших страшних слів. Без вдумливого аналізу не можна заперечувати дослідження, у яких доводиться зв'язок між зростанням злочинності та зростанням популярності відеоігор. Замість суперечок краще подивитися на ситуацію під іншим кутом. Вже не перший рік вчені прагнуть довести, що відеоігри можуть приносити користь. У чому саме? Ігри впливають на цілеспрямованість, мислення, емоційне тло та соціальні навички.

Перше, що варто сказати – розвиток когнітивних функцій. Відеоігри, насамперед шутери, покращують сприйняття окремих об'єктів та зорово-моторну координацію. Це означає, що у геймерів краще працює периферійний зір, просторове мислення та орієнтація. Дослідження доводять позитивний вплив таких розваг на уважність людини. З іншого боку, стратегії розвивають логіку, а РПГ – фантазію.

Ігри підвищують ефективність обробки інформації, що надходить у мозок. Геймер швидше відокремлює важливе і відкидає другорядні відомості.

Деякі вчені впевнені, що відеоігри здатні розвинути творчі здібності у дітей. Це підтверджують дослідження університету Мічігану або експериментом психолога Лінди Джексон у 2012 році. Але питання залишається спірним: незрозуміло, чи ігри розвивають у дітях творчий початок чи творчі діти більше цікавляться іграми.

Є дві категорії людей: одні впевнені, що все залежить від таланту та вроджених навичок, інші ж, без сумніву, підуть вчитися малювати, танцювати чи ліпити скульптури, бо знають, що завзятість та старання – головне. Другу категорію людей виховують відеоігри, а в них все залежить не від таланту, а від старанності. Ігри вчать цілеспрямованості та наполегливості, допомагають не пасувати перед труднощами. Багато людей здаються після першої ж невдачі, визнають свою неспроможність у питанні і, можливо, позбавляють себе справи всього життя. Для геймера невдача - урок. Проходячи місії, помиляючись знову і знову, ніхто не кидає. Навпаки, вони наполегливіше шукають лазівку. Тут ще можна сказати про винахідливість, яка неодмінно розвивається у будь-якого геймера.

Ігри впливають на емоційне тло. Найочевидніше: вони допомагають розслабитися, зняти стрес, відволіктися. Американський професор психології Крістофер Фергюсон провів експеримент і з'ясував: найкраще напруги і стресу допомагають позбутися жорстокі відеоігри. Виходить, вони не роблять людину агресивною. Зовсім навпаки. Вся агресія виходить у віртуальному

просторі. А ще ігри допомагають краще контролювати емоції. Адже у віртуальному світі погане самовладання майже завжди веде до провалу.

Гейміфікація - це застосування елементів та принципів ігор у неігрових контекстах з метою мотивації, покращення участі та досягнення поставлених цілей. В останні роки гейміфікація стала широко поширеною у різних галузях, і її актуальність продовжує зростати. Ось кілька причин, чому гейміфікація залишається актуальною у світі:

Мотивація та залучення: Гейміфікація допомагає стимулювати участь та мотивацію користувачів шляхом додавання елементів ігрового дизайну, таких як досягнення, рівні, бали та нагороди.

Навчання та розвиток: В освітніх та тренувальних програмах гейміфікація може зробити процес навчання більш захоплюючим та ефективним, заохочуючи учнів досягати певних цілей та розвивати навички.

Поліпшення досвіду споживача: У сфері бізнесу та маркетингу гейміфікація може покращити досвід споживача, залучаючи та утримуючи увагу клієнтів через інтерактивні та ігрові елементи. Здоровий спосіб життя та благополуччя: Гейміфікація використовується у охороні здоров'я та фітнесі для мотивації людей до прийняття здорових звичок, таких як регулярні тренування та правильне харчування. Соціальна взаємодія та співпраця: Ігрові елементи можуть сприяти співпраці та соціальній взаємодії, заохочуючи користувачів працювати в команді, ділитися досягненнями та змагатися один з одним. Інновації в технологіях: Розвиток технологій, таких як віртуальна та доповнена реальність, а також штучний інтелект створює нові можливості для гейміфікації та її застосування в різних сферах. В цілому, гейміфікація залишається актуальною та затребуваною методикою, яка продовжує знаходити широке застосування у різних галузях людської діяльності. Її ефективність у мотивації та залученні людей робить її потужним інструментом для досягнення різноманітних цілей та завдань.

1. Аналіз та вплив гейміфікації на сучасну людину

1.1 Дослідження на тему як впливають відеоігри

Дослідники з Університету Центральної Флориди довели, що на роботі коротка перерва на відеогру знімає напругу набагато ефективніше, ніж бездіяльність із повною відмовою від гаджетів і навіть медитація.

У ході експерименту вони розділили учасників на три групи: під час 5-хвилинної перерви на роботі одні відмовлялися від гаджетів і не діяли, другі медитували та треті включали мобільну гру. Виявилося, що у тих, хто грав у відеогру, настрій покращився, а у тих, хто сидів у тиші, навіть погіршився.

Керівник дослідження психолог Майкл Рапп каже, що під час перерви важливо отримувати задоволення від процесу. «Зазвичай ми намагаємося зібрати волю в кулак і переробити всі справи, навіть коли ми вже не маємо сил. Хоча набагато ефективніше на кілька хвилин відволіктися і зайнятися чимось, що принесе вам задоволення і допоможе "перезарядитися". Наприклад, пограйте у відеогру», - сказав вчений.

Не єдине дослідження, що доводить позитивний вплив ігор психіку. Експеримент вчених з Оксфорда показав, що діти віком від 10 до 15 років, які грають у відеоігри, емоційно більш врівноважені, ніж ті, хто їх ігнорує.

Важливо відчувати насолоду в процесі, на що вказують дослідники з Університету Центральної Флориди. Це не стосується умовно-безкоштовних ігор і тих, в які ви граєте через азарт — заради підвищення рейтингу в Dota 2, звання в Counter-Strike: Global Offensive.

Важливо знайти гру, в яку буде справді приємно грати. Футбольні симулятори не напружують мозок сюжетом та складними завданнями, при цьому дозволяючи відволіктися від справ, пригодницькі екшени занурюють в інший світ.

Один із головних стереотипів про відеоігри — нібито вони роблять геймерів замкнутими та відривають від реальності. Батьки бачать, як їхні діти цілодобово сидять перед монітором і не бачать однолітків у житті.

Насправді ж гри навпаки сприяють соціалізації людей. У 2015 році Pew Research Center опублікував дослідження про те, як відбувається соціалізація підлітків-геймерів 13-17 років за допомогою відеоігор. З'ясувалося, що 36%

хлопчиків завели друзів, познайомившись із ними завдяки відеоіграм, з них більше половини під час онлайн-ігор. Серед дівчаток-геймерів знайомства в онлайн-іграх завели лише 13%, але й дівчаток-геймерів налічується менше, ніж хлопчиків.

Гейміфікація — це використання елементів ігрового дизайну в неігрових контекстах. Вважається, що навчальний дизайн орієнтований на студента, щоб мотивувати студента до навчання та академічної поведінки. У цьому дослідженні було досліджено вплив гейміфікації на статистику навчання (проблема перевірки гіпотез) і ставлення до статистики в порівнянні зі звичайним підходом до електронного навчання. Також було оцінено досвід студентів і критичні елементи гейміфікації щодо статистики навчання.

Методи:

У дослідженні до і після, шляхом перепису, 64 студенти факультету охорони здоров'я Університету медичних наук Гуйлан, Рашт, Іран, були невипадковим чином розподілені до груп втручання (n=42) і контрольної (n=22). Навчальна діяльність була гейміфікована в групі втручання, тоді як контрольна група отримувала традиційне вирішення проблем у системі управління навчанням. Оповідь, аватар, рівень, очко, індикатор прогресу, табло, виклик і відгуки використовувалися в ігровому досвіді. Впровадження гейміфікації було застосовано на основі теорії гейміфікованого контенту Ландерса. Дійсна та надійна перська версія опитувальника «Ставлення до статистики» вимірювала ставлення студентів до та після втручання. Анкета EGameFlow і дійсний і надійний іспит, розроблений дослідниками, виміряли досвід користувачів щодо гейміфікованого вмісту та тестування гіпотез навчання після втручання. Т-критерій незалежних вибірок, аналіз коваріації та частковий розмір ефекту ета-квадрат були розраховані за допомогою програмного забезпечення SPSS версії 26.

Результати: Порівняно з контрольною групою, група втручання мала більш позитивне ставлення до труднощів у навчанні (помірний частковий ета-квадрат 0,099), цінності та когнітивної компетентності (слабкий частковий ета-квадрат=0,01 і 0,05). Навчання між двома групами не відрізнялося (P=0,522). Існував значний зв'язок між навчанням і сприйнятим досвідом студентів із зворотним зв'язком (r=0,583, P<0,001), концентрацією (r=0,509, P=0,005) і завданням (r=0,421, P=0,023) гейміфікованого вміст.

Висновок: У ньому пропонується використовувати гейміфікацію для статистики навчання, одночасно оптимізуючи дизайн з більшою увагою до елементів зворотного зв'язку, викликів і концентрації.

1.2 Зміст та структура мобільного додатку

Додаток буде складатися з квестів які будуть допомагати людині йти до своєї цілі. Між ділом буде тренер який буде як раз давати корисні поради та направляти на вірну дорогу користувача. Також в додатку буде функціонал секундоміру який буде допомагати роботи різні вправи.

Ігровий тренер у мобільному додатку Health&Sport може виконувати різноманітні завдання, спрямовані на покращення тренувань, мотивацію користувачів та забезпечення їхнього здоров'я та фізичної форми. Ось деякі із завдань, які ігровий тренер може виконувати в такій програмі:

Створення персоналізованих програм тренувань: ігровий тренер може адаптувати програми тренувань до індивідуальних цілей, рівня фізичної підготовки та можливостей користувачів.

Мотивація та винагороди: Ігровий тренер може використовувати систему винагород і досягнень, щоб мотивувати користувачів досягати своїх цілей і регулярно тренуватися.

Змагання та виклики: Ігровий тренер може організовувати змагання, виклики та турніри між користувачами, щоб заохотити здоровий змагальний дух і підвищити мотивацію до тренувань.

Харчові звички: тренер з гри може надати поради щодо здорового харчування, встановити цілі щодо споживання калорій і води, а також надати рекомендації щодо балансу харчування.

Відстеження прогресу: Ігровий тренер може допомогти користувачам відстежувати їхні досягнення, прогрес і результати навчання, що мотивує їх продовжувати розвиватися.

Надання інформації про здоров'я: Ігровий тренер може надавати користувачам інформацію про здоров'я, зокрема поради щодо запобігання травмам і лікування, належне навчання та інші корисні поради.

Загалом ігровий тренажер у мобільному додатку Health&Sport може бути потужним інструментом для мотивації користувачів, покращення їхнього здоров'я та досягнення фітнес-цілей. Його завдання – створювати цікаві, ефективні та персоналізовані програми навчання для кожного користувача.

Квести в мобільному додатку Health&Sport можна реалізувати як виклики або завдання, які користувачі повинні виконати для досягнення

певних фітнес-цілей та здоров’я. Ось кілька можливих способів реалізації квестів у цій програмі:

Тематичні квести: квести можна розділити на такі теми, як здорове харчування, фізична активність, психічне здоров’я тощо. Кожна тема може мати власні завдання та проблеми.

Цільові квести: користувачі можуть встановлювати свої цілі, а квести пропонуватимуть завдання, спрямовані на досягнення цих цілей. Наприклад, якщо користувач ставить перед собою мету наростити м'язову масу, то квест може включати завдання силових тренувань.

Завдання з обмеженням часу: завдання можуть мати певний час і тривалість. Наприклад, «21-денний квест зниження стресу», де користувач повинен виконати певні завдання протягом 21 дня.

Соціальні квести: квести можуть бути спрямовані на співпрацю та змагання з іншими користувачами. Наприклад, користувачі можуть змагатися один з одним за досягнення певних результатів у фітнесі або здоровому способі життя.

Бонусні квести: квести можуть надавати бонуси або нагороди за їх виконання. Наприклад, користувач може отримати знижку на послуги фітнес-центру або можливість відкривати нові рівні програми.

У кожному випадку реалізація квестів може включати комбінацію елементів гейміфікації, таких як досягнення, таблиці лідерів, нагороди та здоровий змагальний дух, щоб мотивувати користувачів вести активний, здоровий спосіб життя.

Програми тренувань будуть ділитися на 2 типи: силові, для схуднення.

Силові включають в себе комплекс вправ на різні м’язи, які будуть вплавити на сильний зріст м’яз. Силові тренування – найважливіший вид фізичної активності. Вони впливають на все тіло: прискорюють кровообіг, знімають навантаження із суглобів і переносять її на м'язове волокно, збільшують м'язи та позбавляють зайвого жиру. А також тренують серце – головний м'язовий орган нашого тіла.

Силові тренування - це заняття, під час яких м'язи опираються навантаженню. Навантаження на м'язи створюється рахунок власної ваги, тренажера чи обтяжень. Обтяження можуть вважатися гирі, гантелі, медболи або штанги. У силове тренування входять будь-які вправи на силу: жими, присідання, тяги, віджимання, махи, випади та багато іншого. Силові

тренування збільшують обсяг м'язів через напругу, тому найважливіше в силовому тренуванні — створити цю напругу.

Найважливіший ефект силових тренувань - збільшення м'язової тканини. З віком людина поступово втрачає м'язову тканину, але без неї наш організм не зможе жити — м'язи рухають тіло, утримують органи разом, штовхають кров за венами та виділяють тепло при скороченні. Тому силові тренування потрібні всім без винятку: і тим, хто хоче накачати м'язи, і тим, хто не хоче втрачати їх з віком.

Для схуднення ви будете робити ті вправи які більш всього забирають у вас калорії. Займайтеся у фітнес-клубі, на стадіоні, воркаут-майданчиках, робіть вправи вдома: для схуднення не потрібно мучити себе, вибирайте ті активності, які не приносять дискомфорту і приносять задоволення.

Програма тренувань для схуднення має відрізнятися невисокою або середньою інтенсивністю. Варто віддати перевагу кардіонавантаженню та базовим вправам. Відмовтеся від бажання піти одразу на занадто інтенсивні тренування, ви можете швидко втомитися та втратите мотивацію до подальшого вдосконалення тіла. Слідкуйте за правильною технікою, це допоможе уникнути травм.

Враховуйте, що багато популярних видів спорту та вправ, наприклад біг, підійдуть не кожному. Залежить від ваги, стану суглобів, фізичного стану, протипоказань та інших чинників. Тому програма тренувань для людей із надмірною вагою має розроблятися лише професійним фітнес-інструктором.

Жир починає йти при негативному енергетичному балансі, коли в організм надходить менше калорій, ніж витрачається. Саме тому базовий принцип зниження ваги – забезпечення дефіциту калорій в ідеалі за рахунок комбінації збалансованого харчування в межах розрахованої калорійності та будь-яких тренувань, для схуднення цього буде достатньо.

Для забезпечення інтенсивного спалювання важлива будь-яка денна активність, від ходьби в парку до миття посуду. При цьому, якщо ви йдете до тренажерного залу, враховуйте, що при схудненні тренуванню варто приділити щонайменше 30 хвилин.

Важливо пам’ятати, що вибір цих напрямів залежить від мети людини, якщо вона зайву вагу, то їй треба схуднути, а вже потім набирати масу для м’язів, коли чоловік худне спочатку в нього втрачаються м’язи. В першу чергу тіло починає використовувати запаси білків: спочатку ті, що містяться в печінці та шлунково-кишковому тракті, після – білки в скелетних м'язах. Одночасно худнути та набирати вагу треба набагато більш зусиль, тобто це дуже складний шлях який не кожний зможе зробити. Щоб зберегти м'язову

масу, вам доведеться збалансовано харчуватись, а для цього темпи сушіння потрібно уповільнити. Оптимально втрачати приблизно 0,5-1 відсотків маси тіла на тиждень. Також слід робити перерви у дієті, інакше тіло перейде у режим «енергозбереження». Не забувайте, що першими худнуть на дієті саме м'язи. Зупинити цей процес без правильного збалансованого харчування неможливо. Білки - ключовий елемент для зростання та відновлення м'язів. Підтримуйте баланс макроелементів у харчуванні: білки, жири та вуглеводи. І не забувайте про вітаміни та мінерали. Систематичні вправи важливі підтримки тонусу. Що більше м'язова маса, то більше калорій спалює ваше тіло, навіть у стані спокою. Тренуйтеся за розкладом, але не забувайте про періоди відпочинку. Відпочинок – невід'ємна частина підтримки м'язової маси. Під час сну відбувається відновлення м'язів. Переконайтеся, що ви спите достатньо, щоб надати тілу час для регенерації. Вода відіграє у підтримці м'язів. Вона допомагає доставити поживні речовини до м'язів, покращує обмін речовин та запобігає втомі. Прагніть випивати щонайменше 2 літрів води на день. Пам'ятайте, кожна людина унікальна, і що працює для одного може не підійти іншому. У когось обличчя худне насамперед, а у когось стегна чи живіт. Експериментуйте та знаходите свій шлях до збереження м'язів. Пам'ятайте, кожна людина унікальна, і що працює для одного може не підійти іншому. У когось обличчя худне насамперед, а у когось стегна чи живіт. Експериментуйте та знаходите свій шлях до збереження м'язів.

Незважаючи на те, що відсоток жиру поступово знижується по всьому тілу, а не точково, у чоловіків і жінок все одно спостерігається одна і та ж тенденція: перші швидше помічають різницю по ремінцю штанів, а другі - по схудлій верхній частині тіла.

Справа в тому, що в організмі чоловіків більше вісцерального жиру: він накопичується в ділянці грудей, животі та нижній частині спини. Вісцеральний жир починає йти насамперед, тому чоловіки швидше помітять зменшення обсягів у талії, ніж у іншій частині тіла.

У жінок велика частка жиру концентрується в нижній частині тіла: стегнах та литках. Однак у жіночому організмі превалює частка підшкірного жиру, тому перші зміни дівчини починають помічати саме у верхній частині тіла - адже низ худнути не поспішає.

Насамперед, треба звернути увагу на раціон. Уникайте дієт, що обіцяють швидкі результати, оскільки вони можуть призвести до втрати м'язової маси, а не жиру. Збалансоване харчування – це основа правильного схуднення. Прагніть вживати велику кількість овочів, цільнозернових

продуктів, білка і здорових жирів. Незважаючи на те, що це може звучати тривіально, його цінність підтверджена багатьма науковими дослідженнями.

Підвищена активність допомагає спалювати калорії, зміцнює м'язи, включаючи грудні, та покращує загальну форму тіла. Ми з'ясували, що в першу чергу у дівчат худне верхня частина тіла, у тому числі груди. Вплинути цей процес можна лише з допомогою регулярних тренувань. Навіть у зрілому віці тренування можуть позитивно позначитися на стані та здоров'ї жіночих грудей. Почніть з невеликого навантаження і поступово збільшуйте його.

1.3 Рішення проблем за допомогою гейміфікації

Навички вирішення проблем необхідні, щоб мати можливість функціонувати в повсякденному житті. Будь-яка перешкода, труднощі, головоломки чи неприємності приймають форму проблеми, яку ваш мозок вимагає знайти рішення. На найскладнішому рівні проблема може полягати в тому, як побудувати достатньо потужний телескоп, щоб бачити інші сонячні системи. На найпростішому рівні проблема може полягати лише в тому, щоб вирішити, що ви збираєтеся їсти на вечерю.

На робочому місці роботодавці дуже бажають передових навичок вирішення проблем. Подумайте про працівника, який може визначити першопричину проблеми, проаналізувати можливі способи вирішення та запровадити найефективніше рішення. Вони об’єктивно є великим активом для будь-якої організації. Це особливо вірно в робочому світі, який настільки сильно закритий дистанційною та гібридною роботою. Якщо працівник є єдиною людиною в офісі в певний день і щось піде не так, як він використовуватиме свої навички вирішення проблем, щоб самостійно виправити ситуацію?

Складності, для вирішення яких можуть знадобитися індивідуальні навички вирішення проблем, очевидно, можуть бути різними за масштабом. Також буде велика різниця в тому, наскільки здібні люди, коли справа доходить до такого роду роботи, але чи можна покращити навички вирішення проблем? Більш доречно, чи можна покращити навички вирішення проблем на робочому місці за допомогою гейміфікації та навчальних ігор? Чи можете ви застосувати ігрову механіку, щоб навчити своїх співробітників краще вирішувати проблеми?

Головоломки та логічні головоломки сягають корінням в історію людства, у різних культурах з’являлося багато варіантів класичних головоломок. Вони вимагають творчого мислення для пошуку рішень. Одним

із найвідоміших прикладів цього є проблема вовка, кози та капусти. Якщо фермер купує на базарі вовка, козу та капусту, але в човні має достатньо місця, щоб перевезти себе та одну зі своїх покупок через річку по дорозі додому, у якому порядку він їх везе через? Йому потрібно бути креативним, щоб переконатися, що коза не з'їсть капусту, а вовк не з'їсть козу

Щоб вирішити цю головоломку, потрібно усвідомити, що фермер може перенести речі назад через річку. Він може відвести козу від вовка, але згодом він може відвести козу назад, коли капуста перетнеться. Знаючи цю творчу лазівку, спочатку складну проблему легко вирішити. Побачивши питання про вовка, цапа та капусту та зрозумівши його рухомі частини, подумайте про застосування того самого процесу мислення до подібних проблем. Розглянемо курку, лисицю та трохи зерна або команду з чотирьох людей, які зважують різну кількість, вагу своїх запасів і максимальну місткість свого каное.

У той час як початкову проблему важко вирішити, наступні проблеми набагато простіші. Оскільки з ними вже стикалися, ви знаєте процес. Організації побудовані на жорстких і м'яких процесах, тому розглядайте свій процес вирішення проблем як основу для будь-якого корпоративного навчання з цієї теми.

Люди хочуть отримати розумовий виклик і потренувати свій мозок. Вважайте, що ваші працівники мають природне бажання використовувати свої навички вирішення проблем. Ті самі логічні головоломки та головоломки, які з'являлися протягом історії людства, є свідченням цього. Ця прагнення до викликів є найважливішим психологічним важелем взаємодії, який потрібно використовувати під час гейміфікації вашого корпоративного навчання, щоб краще вирішувати проблеми.

Першим кроком є знання складного процесу вирішення проблем у вашій організації. Подумайте, чи це вже встановлено письмово, чи закріпіть це. Важкий процес вирішення неочікуваних проблем, перелік дій, які потрібно виконати на якому етапі, закладає структуру, навколо якої ваші команди можуть застосовувати свої творчі навички вирішення проблем.

Після того, як складний процес вирішення проблем буде встановлено як схвалену компанією методологію, його можна гейміфікувати, щоб навчити ваші команди використовувати його. Практика навколо методології допоможе їм розвинути власні навички вирішення проблем у межах її структури. Налаштування онлайн-навчальних ігор, таких як Вікторина на платформі Drimify, дозволяє поставити серію запитань, що стосуються проблеми, з якою

ваша організація стикалася раніше. Оскільки все це онлайн, доступне на будь-якому сучасному пристрої та дає вам миттєвий зворотний зв’язок, це може бути чудовим інструментом для оцінки існуючих навичок вирішення проблем ваших команд і їхньої реакції на попередні кризи.

Розширюючи початкову концепцію, ви можете використовувати формат Dynamic Path™ від Drimify. Ця опція дозволяє створити багаторівневий інтерактивний навчальний шлях і надає доступ до всіх ігор на платформі Drimify для налаштування. Деякі рівні можуть мати форму вікторини, або з кількома варіантами відповідей, або з відкритими відповідями, а деякі рівні можуть складатися з відео або текстового вмісту, розробка конкретного кроку або аспекту процесу вирішення проблем вашої організації.

Навчальний шлях можна побудувати так, щоб не лише дати їм детальну та інтерактивну підґрунтя ваших методів вирішення проблем, але ви також можете використати його, щоб ознайомити їх із тим, як і чому цей процес був зібраний. Це дозволить їм спробувати вирішити реалістичні проблеми, але за сценарієм низьких ставок. Це дає їм пісочницю, у якій вони можуть займатися складними проблемами, практикувати критичне мислення й аналіз, щоб знаходити рішення.

Гейміфікація навичок вирішення проблем як частина вашого корпоративного навчання дає змогу вашим командам реалістично вирішувати проблеми в навчальному середовищі з низькими ставками. Він також може озброїти ваші команди необхідною структурою та методологією для пошуку ефективних рішень. Кожен працівник є людиною, і хоча всі вони по-різному підходять до вирішення проблем і отримають дещо різні результати, навчання їх складному процесу забезпечує послідовні, ретельні та професійні результати.

Повторне моделювання та ознайомлення з типами проблем, які виникають у вашій галузі за допомогою навчальних ігор, зроблять ваших співробітників набагато краще підготовленими для вирішення нових проблем, коли вони виникають. Це також допоможе їм визначити схожість між новими та старими проблемами. Це може допомогти їм знайти ефективність завдяки можливості використовувати старі рішення.

1.4 Постановка задачі

Назва: Розробка мобільного додатку Health&Sport з елементами гейміфікації

Вступ:

У сучасному світі здоров’я та фізична активність стають все більш пріоритетними для людей через зростаюче усвідомлення їх важливості для загального добробуту та підтримки способу життя. Мобільні додатки з гейміфікованими функціями стають дедалі популярнішими для пропаганди здорового способу життя, оскільки вони дають користувачам стимул і мотивацію досягати своїх цілей щодо фізичної активності та здоров’я.

Мета:

Метою даної дипломної роботи є розробка мобільного додатку Health&Sport з елементами гейміфікації для покращення фізичного здоров’я та стимулювання активного способу життя.

Завдання:

Розробка концепції додатку Health&Sport, включаючи визначення функціоналу та основних функцій. Дизайн інтерфейсу користувача з урахуванням простоти використання та привабливого дизайну. Розвиток функціональності, в тому числі:

Створення базових функцій, таких як статистика фізичної активності, плани тренувань, моніторинг здоров’я тощо. Впровадження елементів гейміфікації, таких як досягнення, бонуси, таблиці лідерів, квести тощо. Тестування та вдосконалення програми за отриманими результатами. Підготовка документації, включаючи технічну документацію, посібники користувача та опис функціональних можливостей програми.

Очікувані результати:

В результаті цієї дипломної роботи очікується розробка та впровадження мобільного додатку Health&Sport, який надає користувачам зручний та привабливий інструмент для підтримки активного та здорового способу життя. Додаток міститиме елементи гейміфікації, які мотивують користувачів досягати цілей щодо фізичної активності та здоров’я.

2. ПРОЕКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ

2.1 Створення бази даних для користувачів

Щоб наш додаток запам’ятав які користувачі заходять в нього йому потрібен свій кошик у якому буде хранитися дані користувачів та інша інформація для вхіда у профіль. Я вибрав для цього Google Firebase.

Firebase — це комплексна платформа, розроблена Google для створення мобільних і веб-додатків. Ось огляд того, що таке Firebase і для чого він використовується:

Firebase надає хмарну базу даних NoSQL, яка дозволяє розробникам зберігати та синхронізувати дані в режимі реального часу між користувачами за лічені мілісекунди. Це особливо корисно для створення програм для спільної роботи, таких як програми для чату, соціальні мережі та багатокористувацькі ігри.

Firebase Authentication надає серверні служби для автентифікації користувачів за допомогою електронної пошти/пароля, номера телефону або популярних постачальників ідентифікаційних даних, таких як Google, Facebook і Twitter. Він забезпечує керування користувачами, включаючи створення облікових записів, відновлення пароля.

Firestore — це безсерверна база даних NoSQL із можливістю сповіщень у реальному часі, і разом із екосистемою Firebase значно спрощує поширені завдання розробки додатків, дозволяючи розробнику додатків зосередитися насамперед на своїй бізнес-логіці та взаємодії з користувачем.Firebase Storage забезпечує безпечне завантаження та завантаження файлів безпосередньо з клієнтського пристрою в Google Cloud Storage. Він ідеально підходить для зберігання створеного користувачами вмісту, наприклад зображень, відео та аудіофайлів.

Firebase Hosting забезпечує швидкий і безпечний веб-хостинг для статичного та динамічного вмісту з вбудованим SSL, глобальним CDN і постійним розгортанням із сховища Git. Це простий спосіб розгортання та масштабування веб-додатків без керування інфраструктурою.

Firebase Analytics допомагає розробникам зрозуміти поведінку користувачів і оцінити продуктивність програми. Він надає статистичні дані про залученість користувачів, утримання та коефіцієнти конверсії, що дозволяє розробникам оптимізувати свої додатки для кращої взаємодії з користувачем і збільшення кількості конверсій.

Моніторинг продуктивності Firebase дозволяє розробникам отримати уявлення про продуктивність своїх програм, включаючи час запуску програми, затримку мережі та швидкість візуалізації інтерфейсу користувача. Це допомагає виявити вузькі місця продуктивності та оптимізувати продуктивність програми для кращої взаємодії з користувачем.

Firebase Remote Config дозволяє розробникам налаштовувати поведінку та зовнішній вигляд своїх програм без розгортання оновлень програм. Це дозволяє розробникам надавати цільовий вміст, функції та рекламні акції певним сегментам користувачів на основі попередньо визначених умов.

Firebase In-App Messaging дозволяє розробникам залучати користувачів за допомогою цільових контекстних повідомлень безпосередньо в їхніх програмах. Це допомагає стимулювати залучення користувачів, утримання та конверсію, доставляючи відповідні повідомлення в потрібний час.

Загалом Firebase надає розробникам потужну та просту у використанні платформу для створення та керування високоякісними програмами, від серверної інфраструктури до зовнішньої розробки та аналітики. Він широко використовується розробниками в усьому світі для прискорення розробки програм, покращення взаємодії з користувачами та розвитку свого бізнесу.

Firebase може працювати з серверною частиною вашої програми, включаючи зберігання даних, автентифікацію користувачів, статичний хостинг тощо.

Функції Firebase:

База даних у реальному часі − Firebase підтримує дані JSON, і всі користувачі, підключені до неї, отримують поточні оновлення після кожної зміни.

Автентифікація – Ми можемо використовувати анонімну автентифікацію, автентифікацію за паролем або різні соціальні автентифікації.

Хостинг – програми можна розгортати через захищене з’єднання із серверами Firebase.

Переваги Firebase:

Це просто і зручно для користувача. Немає необхідності в складній конфігурації.

Дані надходять у режимі реального часу, а це означає, що кожна зміна автоматично оновлюватиме підключених клієнтів.

Firebase пропонує просту інформаційну панель керування.

Є кілька корисних послуг для вибору.

Обмеження Firebase:

Безкоштовний тарифний план Firebase обмежений 50 підключеннями та 100 МБ пам’яті

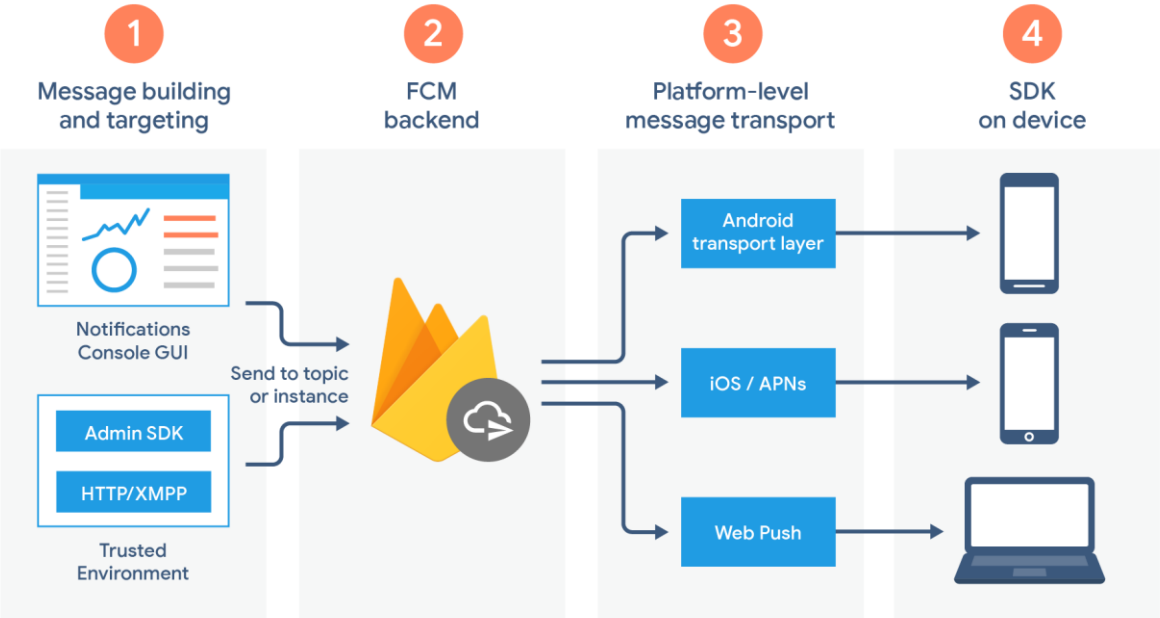


Рисунок 1 – як працює Google Firebase

Щоб зберігати дані користувачів у реальному часі знадобиться Realtime Database.

Realtime Database (Реальне сховище даних) – це хмарна база даних у реальному часі, що надається Firebase. Вона дозволяє розробникам зберігати та синхронізувати дані в реальному часі між користувачами протягом кількох мілісекунд. Ось як вона працює і для чого вона потрібна:

Реальний час синхронізації: Ключова особливість Realtime Database полягає в тому, що вона забезпечує миттєву синхронізацію між усіма підключеними клієнтами. Це означає, що будь-яка зміна даних відразу відбивається на всіх пристроях, підключених до бази даних, без необхідності оновлення або запиту даних.

Структура даних JSON: Realtime Database використовує структуру даних JSON (JavaScript Object Notation), що робить її гнучкою та легкою у використанні. Розробники можуть організовувати дані у вигляді деревоподібної структури, що складається з пар ключ-значення.

Зворотні дзвінки та слухачі подій: Firebase надає API для встановлення зворотних дзвінків та слухачів подій, які автоматично спрацьовують при зміні даних у Realtime Database. Це дозволяє програмам реагувати на зміни даних і оновлювати інтерфейс користувача в реальному часі.

Масштабованість і надійність: Realtime Database автоматично масштабується та забезпечує високу доступність даних. Firebase керує всією інфраструктурою та забезпечує надійну роботу бази даних без необхідності керування серверами чи інфраструктурою.

Використання випадків: Realtime Database ідеально підходить для розробки колаборативних додатків, таких як чати, соціальні мережі, розраховані на багато користувачів ігри і спільна робота над документами. Вона також може бути використана для створення програм, які потребують миттєвого оновлення даних, таких як програми для відстеження реального часу та моніторингу подій.

Офлайн доступ та автоматична синхронізація: Realtime Database підтримує офлайн доступ до даних та автоматичну синхронізацію під час відновлення підключення. Це дозволяє користувачам працювати з даними в автономному режимі та оновлювати їх у базі даних під час відновлення підключення.

В цілому, Realtime Database забезпечує швидку та просту синхронізацію даних у реальному часі між користувачами, що робить її ідеальним інструментом для розробки додатків, які потребують миттєвого обміну даними та колаборативної роботи.

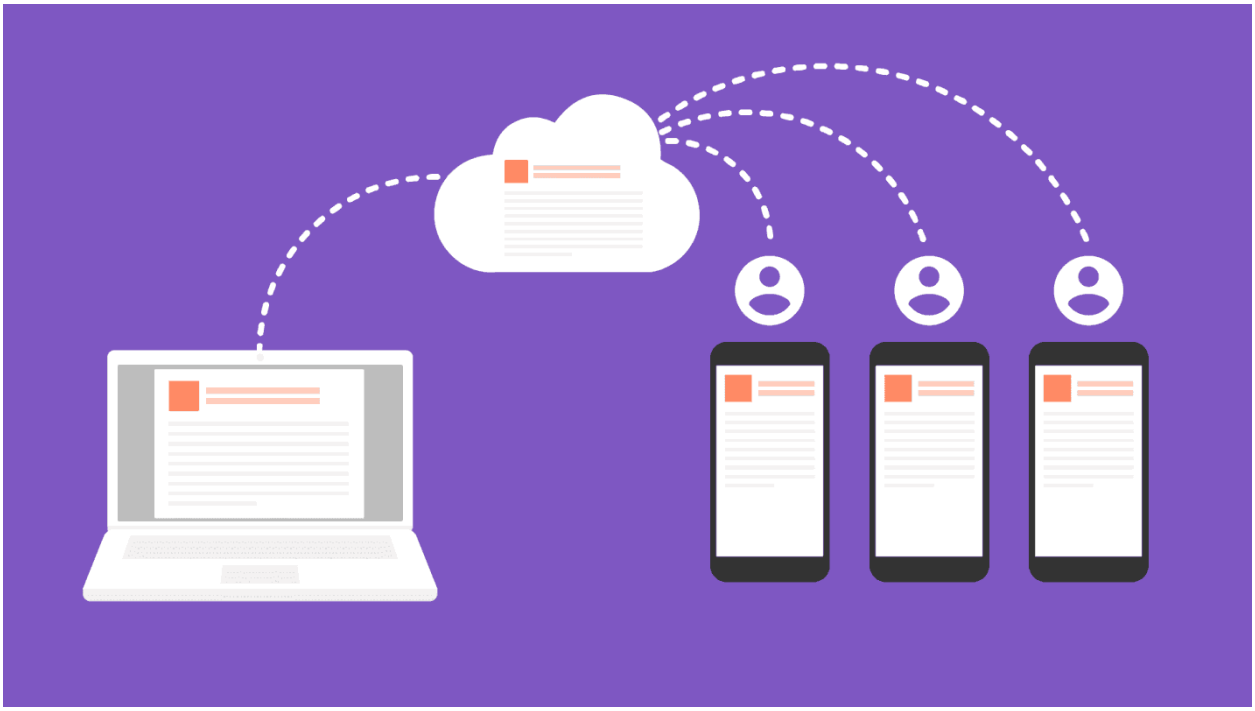


Рисунок 2 – як працює Realtime Database

Firebase надає різні служби та інструменти для розробки мобільних програм, включаючи Realtime Database. Взаємодія між Firebase та мобільним додатком зазвичай відбувається через Firebase SDK, який надає API для роботи з різними сервісами Firebase, включаючи Realtime Database. Ось як це працює:

Налаштування проекту Firebase: Спочатку розробник створює проект у Firebase Console та налаштовує необхідні служби, такі як Realtime Database.

Інтеграція Firebase SDK: Потім розробник додає Firebase SDK у свій мобільний додаток. Це може бути зроблено за допомогою керованих інструментів збирання, таких як Gradle в Android Studio.

Використання API Firebase: Розробник використовує API Firebase SDK у своїй програмі для взаємодії з Realtime Database та іншими сервісами Firebase. Наприклад, для читання та запису даних у Realtime Database, розробник може використовувати методи, що надаються Firebase SDK.

Авторизація та аутентифікація: Firebase також надає API для реалізації аутентифікації користувачів у мобільному додатку. Розробник може використовувати різні методи автентифікації, такі як електронна пошта/пароль, соціальні мережі, телефонний номер та ін.

Синхронізація даних у реальному часі: Програма може підписуватись на оновлення даних у Realtime Database за допомогою слухачів подій. Коли дані

в базі даних змінюються, SDK автоматично сповіщає програму, щоб вона могла оновити інтерфейс користувача.

Офлайн доступ та синхронізація: Firebase SDK автоматично обробляє офлайн доступ до даних та синхронізацію програми з базою даних під час відновлення підключення. Це дозволяє користувачам працювати з даними навіть за відсутності Інтернету.

Загалом Firebase та Realtime Database інтегруються в мобільний додаток за допомогою Firebase SDK, надаючи потужні інструменти для розробки додатків з реальним часом синхронізації даних, аутентифікації користувачів та інших функцій.

2.2 Розробка дизайну додатку

Додаток буде складатися з регістрації, входа, главного меню, опису рівнів та таймеру. Дизайн не буде складним та многофункціональним, тому що для цього потрібен окремо UI/UX дизайнер для розробки інтерфейсу користувача. Також не буде використовуватись Motion Design для анімацій та взаємодії. Дизайн буде простим, але зручним.

У додатку будуть присутні такі xml файли як activity_login.xml , activity_main.xml , activity_press.xml , activity_profile.xml , activity_pullups.xml , activity_pushups.xml , activity_signup.xml , activity_timer.xml

Розширювана мова розмітки (XML) дозволяє визначати та зберігати дані спільно використовуваним способом. XML розшифровується як Extensible Markup Language. XML — це мова розмітки, схожа на HTML, яка використовується для опису даних. Він походить від стандартної узагальненої мови розмітки (SGML). Загалом теги XML не визначені заздалегідь у XML. Нам потрібно реалізувати та визначити теги в XML. Теги XML визначають дані та використовуються для зберігання та організації даних. Його легко масштабувати та легко розробити. В Android XML використовується для реалізації даних, пов’язаних з інтерфейсом користувача, і це легка мова розмітки, яка не ускладнює макет. XML містить лише теги, під час реалізації їх потрібно просто викликати.XML файли та Java файли в Android додатках взаємодіють для визначення інтерфейсу користувача (UI) і його поведінки.

Макети XML (layout):

XML файли використовуються для визначення макету інтерфейсу користувача.

Вони містять розмітку елементів інтерфейсу, таких як кнопки, текстові поля, зображення та інші види елементів.

У макетах XML вказується розташування та стиль елементів інтерфейсу.

Java файли (Activity, Fragment та ін):

Java файли містять логіку програми і управляють поведінкою інтерфейсу користувача.

Вони можуть містити обробники подій, методи роботи з даними, логіку навігації між екранами та інші функції.

Java файли можуть отримувати доступ до елементів інтерфейсу користувача, визначених в XML макетах, за допомогою методів, таких як findViewById().

Зв'язок між XML файлами та Java файлами зазвичай встановлюється так:

XML макети визначаються в ресурсах програми та мають унікальні ідентифікатори (наприклад, R.layout.activity_main).

Java файли містять посилання на XML макети за їх ідентифікаторами і можуть взаємодіяти з елементами інтерфейсу, визначеними у цих макетах.

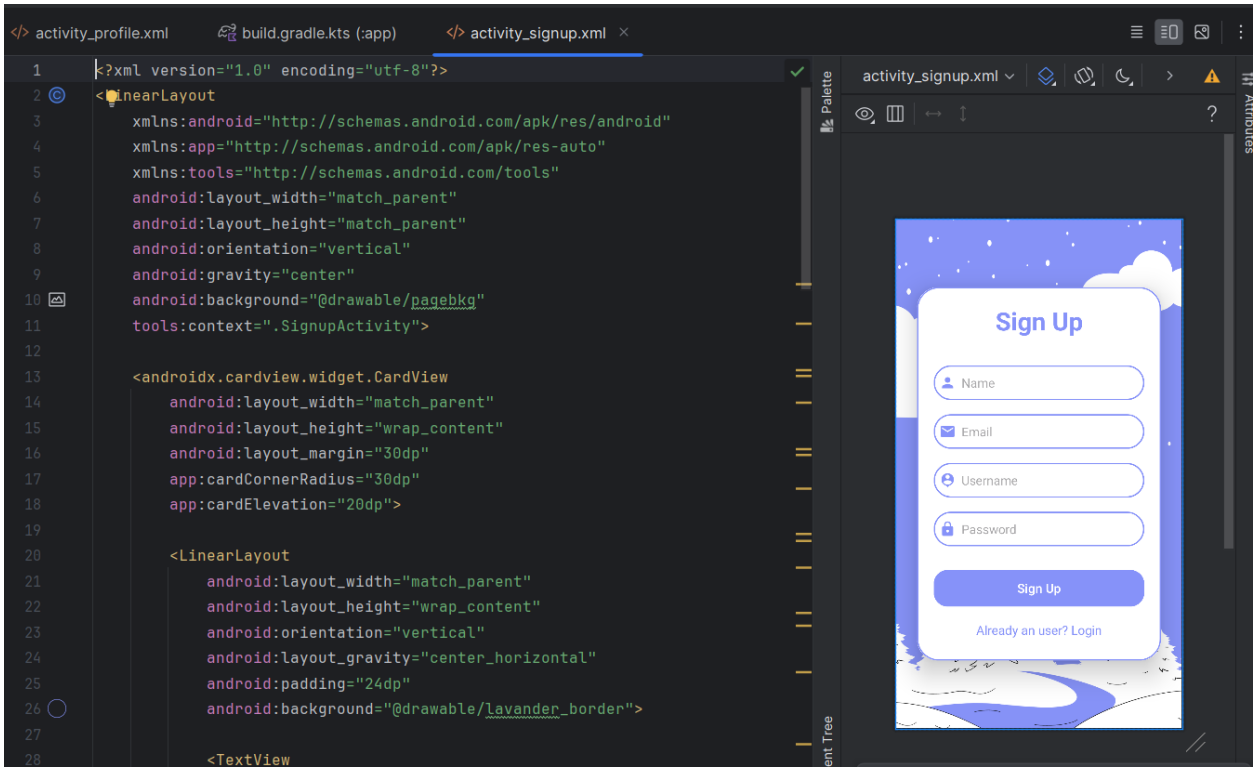


Рис.3 – як виглядає xml макет у Android Studio

Елементи взаємодії з користувачем також класифікуються як елементи UX і інтерфейсу користувача, також відомі як інтерфейс користувача. Розробник може визначити необхідні рішення та створити структуру програми, яка їх пристосує шляхом належного дослідження. Художні фактори програми також важливі та допомагають створити сильну привабливість за допомогою кольорів, шрифтів та зображень.

Розуміння UX-дизайну

Досвід користувача або UX-дизайн — це взаємодія між клієнтом і продуктом або послугою. Кожен елемент, який впливає на взаємодію з користувачем, на те, що користувач відчуває при використанні мобільного додатка та наскільки зручно користувачеві виконувати бажану функцію, підпадає під UX-дизайн програми. Це може змінюватись від того, як кольори, шрифти та інтерфейс додатків змушують користувачів відчувати себе, до того, наскільки просто користувачеві було виконувати бажане завдання через мобільний додаток.

За допомогою UX-дизайну програми дизайнер намагається створити простий, ефективний, логічний, актуальний та загалом приємний досвід для користувачів, щоб вони продовжували використовувати мобільний додаток.

Нелегко створити UX-дизайн програми, який забезпечить приємну взаємодію з користувачами програми. Таким чином, дизайнер повинен поєднувати великі дослідження ринку, розробку додатків, дизайн і стратегію, щоб створити задовільний інтерфейс користувача для мобільного додатка, продуктів, послуг і процесів компанії. Їм необхідно побудувати міст між компанією та клієнтами для зв'язку. Цей зв'язок має дозволити бренду повністю зрозуміти своїх клієнтів і відповідно задовольнити їхні вимоги та очікування.

UX-дизайнери, які розуміють важливість принципів UX, знають, наскільки важливо відповідати на питання зручності використання та наскільки логічним, зрозумілим та простим у використанні є програма. Їм також необхідно переконатися, що дизайн програми вирішує існуючу проблему користувача, через яку програма дуже бажана. Дизайнер також повинен зробити мобільний додаток максимально доступним, щоб спростити доступ до всіх функцій програми.

Розуміння дизайну інтерфейсу користувача

Інтерфейс користувача або дизайн інтерфейсу користувача - це процес, за допомогою якого дизайнери створюють інтерфейси в комп'ютеризованих пристроях або програмному забезпеченні, які повністю зосереджені на візуальних аспектах і стилі мобільного додатка. Кращий дизайн інтерфейсу користувача - той, який користувачі вважають зручним, простим і зручним у використанні і приємним.

Точки доступу, створені для взаємодії користувачів з програмою, називаються інтерфейсами користувача. Графічний інтерфейс користувача, інтерфейс з голосовим управлінням і інтерфейс на основі жестів - це три основні інтерфейси в дизайні інтерфейсу мобільного додатка.

Графічні інтерфейси користувача або GUI являють собою візуальне подання на панелях з цифровим керуванням, які відображають весь контент мобільного додатка і роблять його доступним для взаємодії користувачів. Через інтерфейси з голосовим керуванням або VUI користувачі можуть просто використовувати свій голос для взаємодії з мобільним додатком та виконання своїх завдань. Їх також називають розумними помічниками, а Siri чи Alexa – гарні приклади VUI.

Дизайн інтерфейсу додатків для Android дуже важливий для успішних мобільних додатків, тому що користувачі суворо оцінюють дизайн програми і глибоко піклуються про привабливість і зручність використання. Естетичні елементи програми нічого не означають, якщо вона не може безперешкодно виконати завдання, для якої воно було спочатку створене. Саме з цієї причини дизайн інтерфейсу користувача мобільного додатка повинен слідувати концепції невидимого дизайну. Дизайн програми повинен мати природний потік, щоб користувачі не зосереджувалися на ньому та безперешкодно виконували свої завдання.

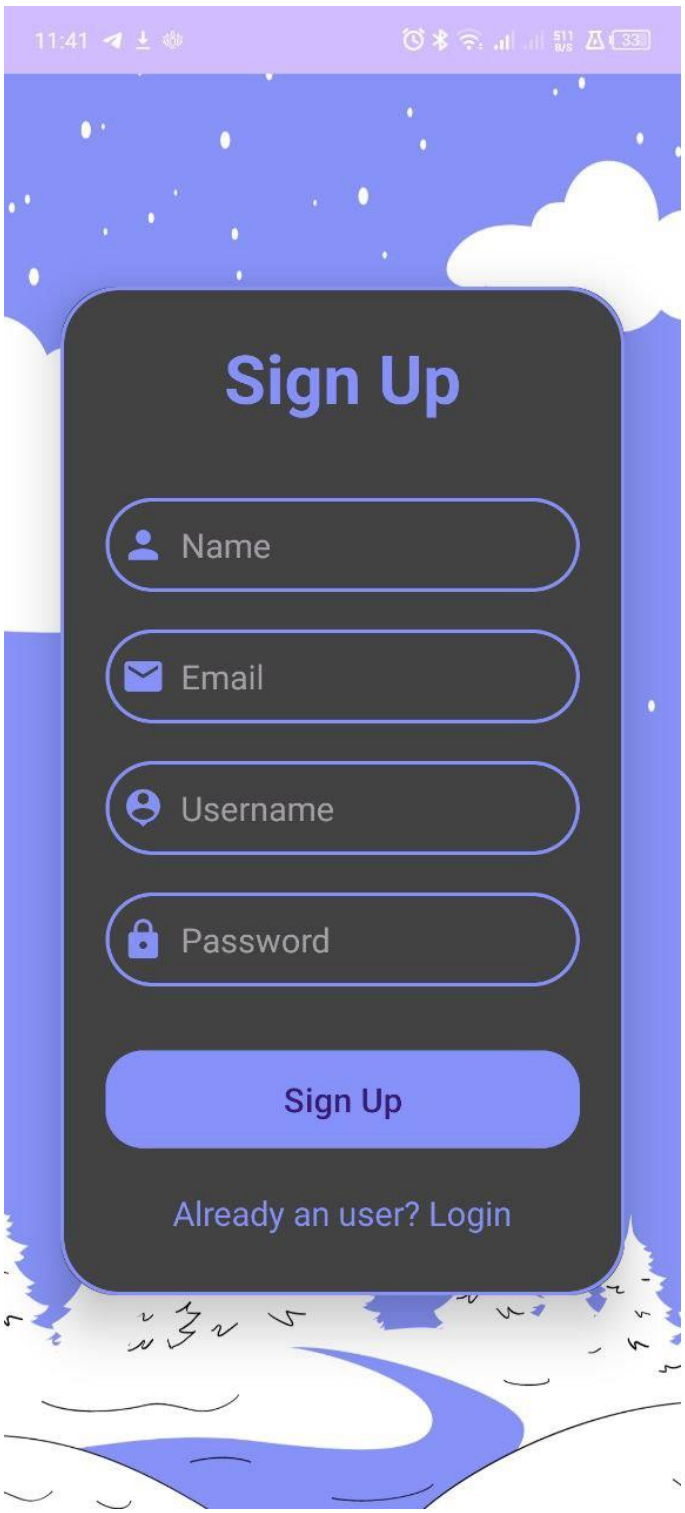


Рис.4 – приклад дизайну регістрації на основі додатку

UI дизайн для користувача який містить в себе усі основні параметри, такі як: привабливість, зручність та простота.

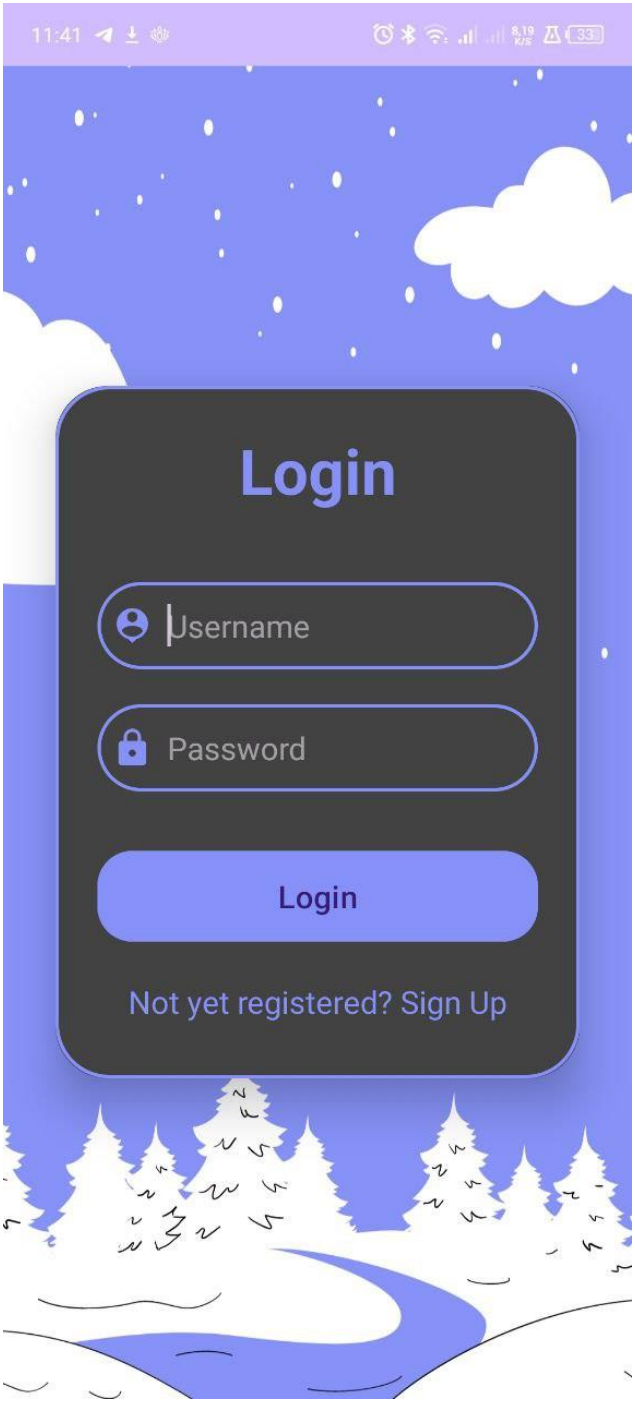


Рис.5 – приклад дизайну входу на основі додатку

Дизайн програми складається з усіх візуальних елементів програми, які визначають, як додаток буде виглядати для користувачів. Привабливий інтерфейс користувача допомагає привернути увагу цільової аудиторії. Дизайн мобільних додатків поєднує елементи UX та UI програми, щоб визначити зовнішній вигляд та функціональність програми. Дизайн мобільного додатка включає колірну схему, шрифт та інші елементи дизайну. У той же час користувальницький досвід (UX) визначає можливості програми та її використання.

UX-дизайн відноситься до терміну «дизайн користувацького досвіду», тоді як UI означає «дизайн інтерфейсу користувача». Обидва елементи мають вирішальне значення для продукту та тісно взаємодіють. Але, незважаючи на їхні професійні стосунки, самі ролі досить різні, стосуються дуже різних аспектів процесу розробки продукту та дисципліни дизайну.

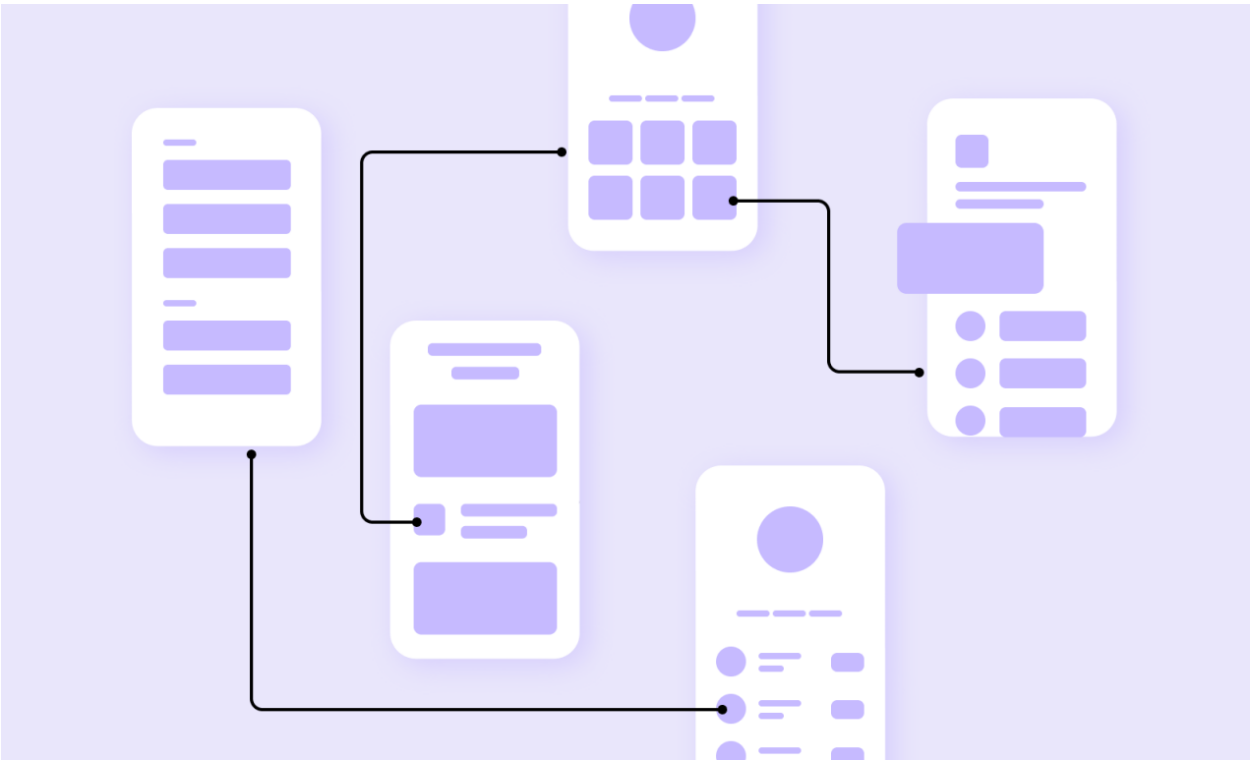


Рис.6 – як виглядає розробка дизайну додатку

Незважаючи на те, що це старіша та більш практична сфера, питання «Що таке дизайн інтерфейсу користувача?» Важко відповісти через широке розмаїття неправильних тлумачень.

Але, як і UX, його легко і часто заплутати галузями, в яких працюють дизайнери інтерфейсу користувача, настільки, що різні вакансії часто називатимуть професії абсолютно різними речами.

На відміну від UX, дизайн інтерфейсу користувача є суто цифровим терміном.

Інтерфейс користувача — це точка взаємодії між користувачем і цифровим пристроєм або продуктом, як-от сенсорний екран на вашому смартфоні або

сенсорна панель, яку ви використовуєте, щоб вибрати, яку каву ви хочете отримати з кавоварки.

Що стосується веб-сайтів і програм, дизайн інтерфейсу користувача враховує зовнішній вигляд, відчуття та інтерактивність продукту. Уся справа в тому, щоб інтерфейс користувача продукту був максимально інтуїтивно зрозумілим, а це означає ретельний розгляд кожного візуального інтерактивного елемента, з яким може зіткнутися користувач.

Дизайнер інтерфейсу користувача буде думати про піктограми та кнопки, типографіку та колірні схеми, інтервали, зображення та адаптивний дизайн.

2.3 Розробка програмного коду

Що таке мобільний додаток

Мобільний додаток – це програма, яка запускається на мобільних пристроях. Такими програмами користуються всі, хто має смартфон або планшет: скачують з магазинів додатків, встановлюють і запускають у себе. Це можуть бути ігри, розважальні програми, програми магазинів і сервісів.

Найчастіше програми створюють для операційних систем iOS та Android. Системи принципово різні, тому спочатку додаток пишуть для однієї з них, а потім за допомогою інших мов переносять в іншу. Іноді для створення програми користуються кросплатформовими технологіями: так можна зробити програму, яка запускатиметься на різних системах, але працювати додаток буде повільніше.

Можна створити програму самостійно. Для цього знадобляться знання програмування та одна з мов, якими пишуть код мобільних програм. Звичайно, відразу реалізувати складний проект може не вийти, але якщо потренуватися на більш простих речах, згодом можна багато чого навчитися.

Як влаштовані програми

У широкому сенсі додаток - це програма для мобільних пристроїв: смартфонів, планшетів, розумних годинників та інших гаджетів. Найчастіше говорять про програми для платформ iOS та Android. Програми розміщуються в магазинах App Store для iOS та Google Play для Android.

Програми найчастіше мають клієнт-серверну архітектуру. Це означає, що вони мають «клієнтська» частина, з якою взаємодіє користувач, і «серверна» — движок.

Деякі популярні мобільні програми, наприклад Wildberries або OZON, — мобільні копії десктопних версій, які працюють з тим же сервером, але мають інший інтерфейс.

Є повністю локальні програми, які зберігають дані на тому ж пристрої, де були запущені. Це можуть бути, наприклад, ігри, для яких не потрібен доступ до інтернету, або інструменти, які працюють лише в автономному режимі. Такі програми не завжди клієнт-серверні.

З чого складається додаток для Android

Програми для операційної системи Android пишуть мовами Java і Kotlin, іноді — C++. Ще є технології, які дають можливість писати іншими мовами: наприклад, за допомогою фреймворку React Native можна створювати програми JavaScript.

Android-додатки є певна структура. Програму умовно поділяють на чотири компоненти: Activity, Service, BroadcastReceiver, ContentProvider.

Activity. Сюди належить графічний інтерфейс – усе, що бачить користувач. Інтерфейс складається з кількох «екранів», кожен з яких — як сторінка: має свій дизайн, свій вміст і можливості. Наприклад, у додатку для інтернет-магазину може бути екран головної сторінки, екран каталогу, екран картки товару тощо. Між екранами можна переходити за допомогою посилань і кнопок - в андроїд-розробці відповідає клас Intent.

Service. Це компонент, який відповідає за тривалу роботу "у фоні", тобто без участі користувача. Наприклад, додаток-таймер потрібно фоново відраховувати час до сигналу. Для таких процесів немає графічного інтерфейсу, адже користувач у них не задіяний – фонову роботу підтримує сам Service.

BroadcastReceiver. Назва перекладається як «широкомовний приймач» і загалом відбиває функціональність компонента. Він приймає звані широкомовні повідомлення (broadcast intents): звичайні інтенти адресовані конкретному модулю, а широкомовні може прийняти будь-хто. Але для прийому потрібен BroadcastReciever. Так можна обмінюватися повідомленнями всередині компонентів програми на Android або між різними програмами.

ContentProvider. Цей компонент підвантажує додаток дані - так званий контент. Наприклад, дані зі сховища телефону, бази даних, віддаленого інтернет-джерела. Якщо дизайн екранів належить до компоненту Activity, їх «наповнення» на кшталт картинок і тексту — це вже контент. Невеликі програми зберігають контент у локальній базі на пристрої, але великі, наприклад, програми маркетплейсів, підвантажують з мережі, тому для їх роботи потрібний інтернет.

Окремо можна згадати базу даних. Технічно вона не відноситься до програми, але якщо програма працює з якимись даними, їх знадобиться десь зберігати. Наприклад — той самий інтернет-магазин: не вдасться зберігати всі дані про товари в інтерфейсі, знадобиться база. Отже, інформацію потрібно завантажувати з мережі: пам'ять пристрою просто не вмістить всю базу середнього інтернет-магазину. Або гра: фігурки персонажів, локації та репліки – це контент. В онлайн-іграх він може зберігатися в базі на пристрої, а в онлайн-іграх частина контенту зберігається локально, а частина завантажується з віддаленої бази в інтернеті.

3. РЕАЛІЗАЦИЯ МОБІЛЬНОГО ДОДАТКУ

3.1 ВИБРАНЕ СЕРЕДОВИЩЕ ВЕРСТКИ ДОДАТКИ

Середовищем розробки мобільного додатка вибрано Android Studio. Він складається з наступних компонентів: Android SDK, компоненти графічного дизайну, додаток для завантаження компонентів всіх версій Android, емулятор мобільного пристрою для запуску програми,

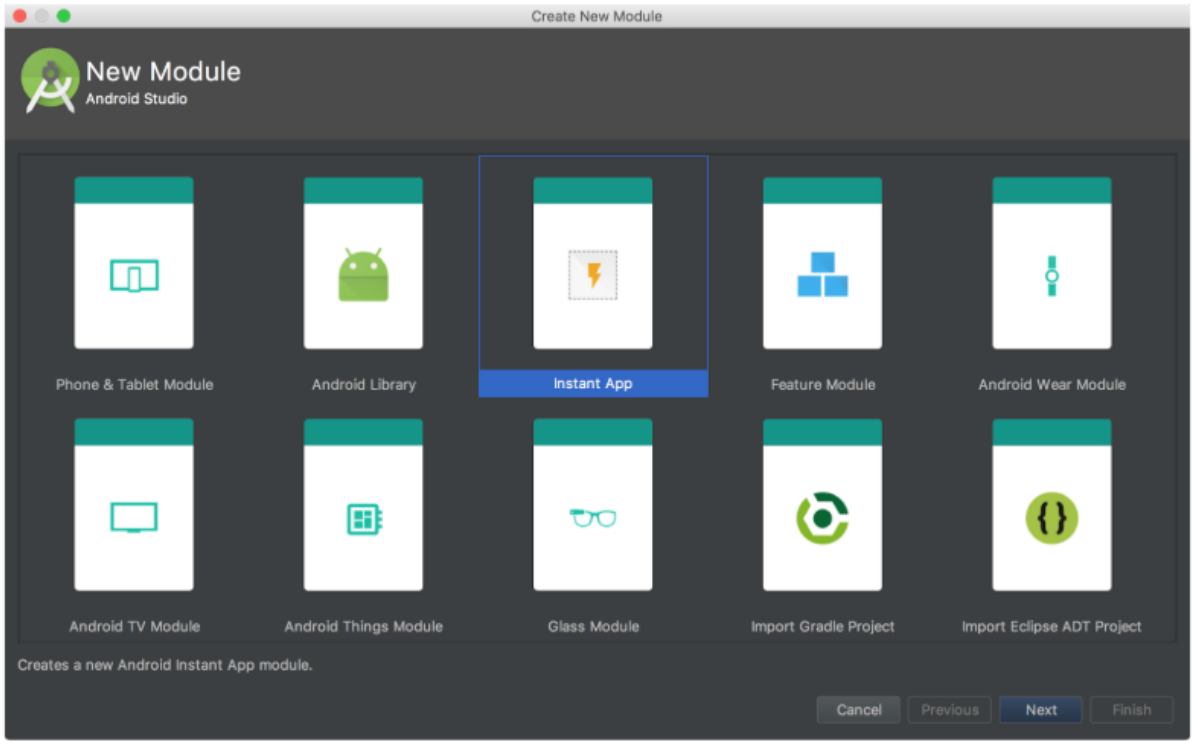


Рис. 8 – Створення нового проекту в Android Studio

інструменти для тестів та налагодження роботи програми.

Важливим моментом при виборі Android Studio стало те, що вона може розробляти програми практично для всіх версій операційної системи Android. Існує інструмент для оцінки зовнішнього вигляду програми різних пристроїв. Багатокольоровий код полегшує навігацію у великих обсягах коду. Програма включає в себе

Google Cloud Messaging, метою якої є налаштування відправлень повідомлень для додатків, у яких використовуються хмарні сервиси під керуванням операційної системи Android. Таким чином, можна

створювати багатозадачні програми. Однією з переваг даного середовища є детальне тестування програми.

Це програмне забезпечення покликане надати

користувачам можливість для комунікації, перегляду стрічки актуальних новин та доступу до Google-карт. Розроблюване

додаток переважно носить довідковий характер, у зв'язку з цим

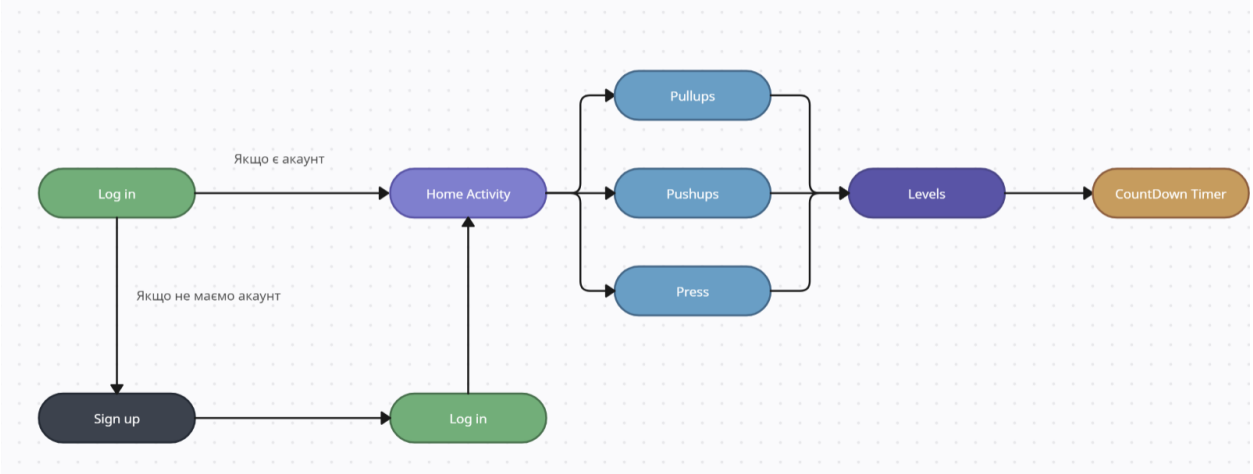


Рис. 10 – Структура мобільного додатка

доступ до основних функцій програми можливий без автентифікації

користувачів.

Android Studio є офіційним інтегрованим середовищем розробки (IDE) для розробки додатків Android. Це потужний інструмент, який дозволяє розробникам створювати високоякісні програми для платформи Android. У ньому є повні інструменти для процесу розробки додатків для Android. Від написання коду до тестування та розгортання, ANdroid studio має всі функціональні можливості для розробників для розробки Android App.

Від початківців до досвідчених розробників, це дуже поширена IDE для розробки високоякісних програм Android. І як технічний ентузіаст, ви також повинні мати базове уявлення про цей інструмент. Тож у цій статті ми розповімо вам про процес налаштування Android Studio на комп’ютері.

Є багато способів, якими Android Studio відрізняється від інших IDE та інструментів розробки. Деякі з цих відмінностей досить незначні, як-от спосіб інсталяції бібліотек підтримки, а інші, наприклад, процес збирання та дизайн інтерфейсу користувача, кардинально відрізняються.

Перш ніж ближче розглянути саму IDE, варто спочатку зрозуміти, у чому полягають деякі з цих важливих відмінностей. Основні з них перераховані тут:

Розробка інтерфейсу користувача. Найсуттєвішою відмінністю між Studio та іншими IDE є її редактор макета, який значно перевершує будь-якого зі своїх конкурентів, пропонуючи представлення тексту, дизайну та креслення, і, що найважливіше, інструменти компонування обмежень для кожної дії чи фрагмента, прості у використанні редактори тем і стилів, а також функція перетягування дизайну. Редактор макетів також надає багато інструментів, недоступних в іншому місці, наприклад повну функцію попереднього перегляду для перегляду макетів на багатьох пристроях і прості у використанні редактори тем і перекладів.

Структура проекту: хоча базова структура каталогу залишається незмінною, спосіб організації кожного проекту в Android Studio значно відрізняється від його попередників. Замість того, щоб використовувати робочі області, як у Eclipse, Studio використовує модулі, з якими легше працювати разом, не змінюючи робочі області.

Ця різниця в структурі спочатку може здатися незвичайною, але будь-який користувач Eclipse незабаром побачить, скільки часу вона може заощадити, коли стане знайомою.

Доповнення коду та рефакторинг: те, як Android Studio інтелектуально завершує код під час введення, робить його приємним у використанні. Він регулярно передбачає те, що ви збираєтеся ввести, і часто цілий рядок коду можна ввести не більше ніж двома-трьома натисканнями клавіш. Рефакторинг також легший і більш масштабний, ніж альтернативні IDE, такі як Eclipse і NetBeans. Майже все можна перейменувати, від локальних змінних до цілих пакетів.

Емуляція: Studio оснащено гнучким редактором віртуальних пристроїв, що дозволяє розробникам створювати емулятори пристроїв для моделювання будь-якої кількості пристроїв реального світу. Ці емулятори легко налаштовуються як за форм-фактором, так і за апаратною конфігурацією, а віртуальні пристрої можна завантажити від багатьох виробників. Користувачі інших IDE вже знайомі з Android AVD, хоча вони точно оцінять функції попереднього перегляду на вкладці «Дизайн».

Інструменти збірки: Android Studio використовує систему збирання Gradle, яка виконує ті самі функції, що й система Apache Ant, з якою знайомі багато розробників Java. Однак він пропонує набагато більше гнучкості та дозволяє створювати налаштовані збірки, дозволяючи розробникам створювати файли APK, які можна завантажувати в TestFlight, або легко створювати демонстраційні версії програми. Також система Gradle забезпечує модульний характер. Замість того, щоб кожну бібліотеку чи сторонній SDK компілювати у вигляді файлу JAR, Studio будує кожну з них за допомогою Gradle.

Це найзначніші відмінності між Android Studio та іншими IDE, але є багато інших особливостей, які є унікальними. Studio надає потужний тестовий засіб JUnit і підтримує підтримку хмарної платформи та навіть налагодження Wi-Fi. Він також значно швидший за Eclipse, який, чесно кажучи, має задовольняти ширший спектр потреб розробки, а не лише одну, і він може працювати на менш потужних машинах.
Android Studio також пропонує дивовижний пристрій для економії часу у вигляді Instant Run. Ця функція вміло створює лише ту частину проекту, яка була відредагована, тобто розробники можуть тестувати невеликі зміни в коді, не чекаючи виконання повної збірки для кожного тесту. Ця функція може скоротити час очікування з хвилин майже до нуля.

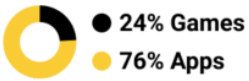
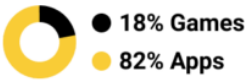
	App Store	Play Market
Total apps	3 141 152 apps  	4 668 410 apps 
Most downloaded	1. Photo & Video 2. Entertainment 3. Social Networking 4. Action 5. Music	1. Casual 2. Communication 3. Tools 4. Arcade 5. Music & Audio
Highest grossing	1. Medical 2. Music 3. Action 4. Role Playing 5. Reference	1. Casual 2. Strategy 3. Arcade 4. Racing 5. Music & Audio
Most new Apps/Week	1. Business 2. Education 3. Lifestyle 4. Utilities 5. Entertainment	1. Entertainment 2. Music & Audio 3. Books & Reference 4. Education 5. Tools

Рис.11 – статистика що краще Android або Apple для розробки

3.2 Етапи розробки додатка

3.2.1 етапи розробки інтерфейса

Першим етапом розробки є вивчення цільової аудиторії та кейсів продукту. Це дозволяє зрозуміти майбутніх користувачів програми та створити зручний інтерфейс, який буде оптимальним для кожного. Тут можуть бути порушені такі аспекти, як розміри та розташування кнопок, шрифти, кольори та стиль оформлення.

Другим етапом йде створення ескізів, що дозволяє нам уявити структуру інтерфейсу. User Flow Diagram допомагає нам проілюструвати 33 логіку продукту, так звану дорожню карту всіх взаємодій та стан екрану на кожному етапі. Далі ми визначаємо стиль інтерфейсу: від мінімалізму до скевоморфізму. На цьому етапі ми промальовуємо кілька варіантів стилю

та пропонуємо клієнту право вибору. Велику увагу приділяємо сучасним трендам, масштабуванню інтерфейсу, оцінюємо час на розробку дизайну.

Наступним кроком є демострація повної версії дизайну. Реалізація може бути низькорівневою, оскільки цей спосіб показує структуру та опис взаємодії користувача з мобільним додатком. Зазвичай виконується в одному-двох кольорах.

Макет дозволяє продемонструвати весь проект дизайну, максимально наближений до реальності. На цьому етапі контент статичний, але така форма подачі сприймається клієнтом наочніше попередньої.

Одним із завершальних етапів є клікабельний прототип. Це більш детальна форма фінального проекту. Вона емулює взаємодію користувача з інтерфейсом, тобто дозволяє натискати на всі елементи управління, перевіряти анімацію і відео. Цей етап потребує великої кількості часу.

На останньому етапі клієнт бачить затверджений дизайн, який може внести невеликі коригування.Розробка інтерфейсу мобільного додатка - це важливий етап, що включає безліч тонкощів і складових. Ось основні етапи та складові розробки інтерфейсу:

1. Дослідження та аналіз:

- Вивчення цільової аудиторії та їх переваг у використанні мобільних додатків.
- Аналіз конкурентів та найкращих практик у дизайні інтерфейсів.

2. Проектування інформаційної архітектури:

- Створення структури програми, включаючи визначення основних розділів, екранів та функцій.
- Розробка схеми навігації задля забезпечення зручності використання.

3. Створення макетів (wireframes):

- На цьому етапі створюються прості чернові макети екранів програми без деталей дизайну.
- Фокусування на розміщенні елементів інтерфейсу та логіці взаємодії між ними.

4. Дизайн інтерфейсу:

- Створення дизайну інтерфейсу користувача, включаючи вибір колірної палітри, шрифтів, стилів елементів та іконок.
- Розробка детальних макетів з урахуванням брендбука та UI/UX-принципів.

5. Прототипування:

- Створення інтерактивних прототипів, що дозволяють протестувати функціонал і взаємодію елементів користувача.
- використання спеціальних інструментів для створення прототипів, таких як Adobe XD, Figma або Sketch.

6. Адаптація під різні екрани:

- Враховуючи різноманітність розмірів та дозволів екранів мобільних пристроїв, інтерфейс має бути адаптивним та коректно відображатися на всіх пристроях.
- Тестування інтерфейсу різних пристроях для перевірки коректності відображення.

7. Тестування та ітерації:

- Проведення тестування інтерфейсу щодо зручності використання, зрозумілості та естетики.
- Збір зворотного зв'язку від тестувальників та кінцевих користувачів та внесення необхідних змін.

8. Документування дизайну:

- Створення документації, що описує дизайн інтерфейсу, включаючи стилі, компоненти та їх використання.

Кожен із цих етапів відіграє ключову роль у створенні ефективного, функціонального та привабливого інтерфейсу мобільного додатка.

Плюси та мінуси розробки інтерфейсу:

Плюси:

1. Покращений досвід користувача (UX): Добре спроектований інтерфейс покращує користувальницький досвід, роблячи додаток більш зручним і привабливим для використання.

2. Збільшення утримання користувачів: Привабливий та інтуїтивно зрозумілий інтерфейс сприяє утриманню користувачів та знижує ймовірність їхнього відтоку.

3. Велика конкурентоспроможність: Користувачі надають перевагу програмам з красивим і зручним інтерфейсом, що робить вашу програму більш конкурентоспроможною на ринку.

4. Збільшення рівня залученості: Цікавий та інтерактивний інтерфейс може підвищити рівень залученості користувачів, роблячи використання програми більш захоплюючим.

5. Краща зручність використання: Добре спроектований інтерфейс спрощує виконання завдань користувачами, що сприяє підвищенню зручності використання програми.

Мінус:

1. Витрати часу та ресурсів: Розробка якісного інтерфейсу потребує часу та ресурсів, особливо якщо необхідно провести багато тестування та ітерацій.

2. Невизначеність в очікуваннях користувачів: Іноді очікування користувачів можуть не відповідати розробленому інтерфейсу, що призводить до необхідності внесення змін та додаткової роботи.

3. Складність створення універсального інтерфейсу: Різні користувачі можуть мати різні переваги та потреби, що робить складним створення універсального інтерфейсу, який би влаштував усіх.

4. Обмеження платформи та пристрої: Різні платформи (iOS, Android) та пристрої мають свої особливості та обмеження, що може ускладнити створення інтерфейсу, який виглядає та працює однаково на всіх пристроях.

5. Необхідність постійного оновлення: Інтерфейс вимагає постійного оновлення та підтримки, особливо з урахуванням вимог користувачів і технологічних нововведень, що змінюються.

3.2.2 Визначення цільової аудиторії

Ринок у будь-якій сфері проходить основні стадії: експеримент, засвоєння та насиченість. Мобільні програми, що формуються у фазі експерименту, відрізняються інтенсивним зростанням установок. Це ті користувачі, які вперше стикаються зі смартфоном і пізнають програми. Стадія засвоєння полягає в звичному наборі стандартних додатків (карти, оплата, магазини). Насиченість досягає максимальних меж, коли користувачі платять гроші за мобільні програми та послуги, які там є. Щоб знизити витрати на просування товару і вийти ринку зі своєю пропозицією, слід здійснити аналіз цільової аудиторії. Згодом аудиторія програми зростатиме і у вашого продукту підвищиться прибуток.

Цільова аудиторія – це група людей, для яких ви хочете придбати ваші товари чи послуги. Дуже небагато компаній коли-небудь матимуть цільову аудиторію «всіх» — намагання продати всім часто призведе до того, що не продадуть нікому. Натомість цільову аудиторію складатимуть люди, які отримають найбільшу користь від продуктів. Цю групу визначають певні демографічні характеристики та поведінка, які можна сегментувати на певні особи. Ці персонажі дають імітацію людини, яка представляє звичайну людину в конкретній цільовій аудиторії.

Одним із найкращих прикладів брендингу додатків є Netflix. Примітно, що вони знають свою цільову аудиторію з першого дня, і це головна причина, через яку весь зовнішній вигляд та естетика програми створені з урахуванням інтересів молодого покоління.

Цільова аудиторія програми Health&Sport з елементами гейміфікації включає людей різного віку та рівнів фізичної підготовки, які прагнуть поліпшити своє здоров'я та фізичну форму через спортивні заняття. Це можуть бути спортсмени-початківці, які бажають освоїти нові види фізичної активності, досвідчені атлети, які прагнуть постановки нових спортивних рекордів, а також люди, які просто бажають підтримувати здоровий спосіб життя.

Основна мета програми – зробити заняття спортом захоплюючими та мотивуючими, надавши користувачам доступ до різноманітних тренувань та вправ, а також створюючи для них ігрові елементи, що стимулюють досягнення нових результатів. Гейміфікація в додатку може залучити користувачів різних вікових груп та мотивувати їх до регулярних

тренувань, оскільки змагальний аспект та досягнення ігрових цілей додають веселощів та задоволення від занять спортом.

Користувачі, які можуть скористатися програмою Health&Sport, включають:

- Люди, які ведуть сидячий спосіб життя і прагнуть активної фізичної діяльності для покращення здоров'я та фітнесу.

- Спортсмени, які бажають доповнити свою основну тренувальну програму новими вправами та видами фізичної активності.

- Люди, які шукають мотивацію та підтримку у досягненні своїх спортивних цілей через ігрові елементи та змагання.

- Тренери та інструктори, які можуть використовувати програму для розробки персоналізованих тренувальних програм для своїх клієнтів та учнів.

Використовуючи гейміфікацію та різноманітні функції програми Health&Sport, розробники можуть залучити широку аудиторію та допомогти користувачам на шляху до покращення свого здоров'я та досягнення спортивних результатів.

При пошуку цільової аудиторії можуть виникнути такі проблеми:

1. Невизначеність цільового ринку: Недостатнє розуміння чи невизначеність щодо цільового ринку може ускладнити визначення характеристик цільової аудиторії, таких як вік, стать, інтереси та переваги.

2. Недостатні дані: Відсутність достатніх даних про потенційних користувачів, їх поведінку та потреби може ускладнити визначення цільової аудиторії та створення ефективних маркетингових стратегій.

3. Неправильна сегментація: Неправильна сегментація аудиторії або недостатній облік різних критеріїв (наприклад, географічний розподіл, соціальний статус, рівень доходу) може призвести до невірних висновків про цільову аудиторію.

4. Складність залучення цільової аудиторії: Залежно від характеристик аудиторії та особливостей ринку може бути складно досягти та залучити цільову аудиторію, особливо якщо існує висока конкуренція чи обмежені ресурси для маркетингу.

5. Зміна переваг користувачів: Переваги та потреби користувачів можуть змінюватися з часом, що потребує постійного моніторингу аудиторії та адаптації маркетингових стратегій.

Щоб вирішити проблеми у пошуку цільової аудиторії, можна застосувати такі підходи:

1. Дослідження ринку: Провести ретельне дослідження ринку, включаючи аналіз конкурентів, вивчення поведінки потенційних користувачів, а також збір та аналіз даних про переваги та потреби аудиторії.

2. Сегментація аудиторії: Розділити цільову аудиторію на сегменти на основі різних критеріїв, таких як вік, стать, географічний розподіл, інтереси та переваги. Це дозволить більш точно визначити характеристики кожного сегмента та розробити персоналізовані маркетингові стратегії.

3. Використання аналітики: Скористуватися інструментами аналітики для збору даних щодо поведінки користувачів у програмі, веб-сайті або соціальних мережах. Це допоможе краще зрозуміти потреби та переваги аудиторії, а також оцінити ефективність маркетингових кампаній.

4. Тестування та ітерації: Проводите тестування та ітерації маркетингових стратегій, щоб визначити їх ефективність та внести необхідні зміни. Це дозволить покращити залучення та утримання цільової аудиторії.

5. Адаптація до змін: Будучи гнучкими та адаптуватися до змін у перевагах та потребах аудиторії. Постійно моніторьте ринок та уважно стежте за змінами в поведінці користувачів, щоб швидко реагувати та адаптувати свої маркетингові стратегії.

Висновок

У рамках даної дипломної роботи було розроблено мобільний додаток Health&Sport, що поєднує в собі функціональність спортивного трекера та елементи гейміфікації. Розробка програми велася з урахуванням сучасних тенденцій у сфері здорового способу життя та використання мобільних технологій для мотивації до фізичної активності.

Основними результатами роботи є:

1. Створення зручного та інтуїтивно зрозумілого інтерфейсу програми, що дозволяє користувачам легко відстежувати свої спортивні досягнення та прогрес.

2. В майбутньому впровадження елементів гейміфікації, таких як досягнення, бейджі та квести, які допомагають мотивувати користувачів та роблять процес занять спортом більш захоплюючим.

3. Використання технології Firebase для реалізації функціональності реєстрації, аутентифікації та зберігання даних користувачів у реальному часі.

4. Проведення тестування програми на різних пристроях та аналіз зворотного зв'язку користувачів для покращення його функціональності та зручності використання.

В результаті проведених досліджень та розробки мобільного додатка Health&Sport ми дійшли висновку про те, що елементи гейміфікації можуть значно покращити мотивацію користувачів до занять спортом та допомогти їм досягати своїх цілей у галузі здоров'я та фізичної активності.

Ця робота може бути використана як основа для розробки аналогічних додатків, а також у подальших дослідженнях на тему гейміфікації в охороні здоров'я та спорті.

Список використаних джерел

1) Гріффітс Р. Д. Head First. Програмування для Android [Текст] / Р. Д. Гріффітс - Санкт-Петербург: Пітер, 2016. - 704 с.

2) Аарон Хіллегас. Objective-C. Програмування для Android. 2012р.

3) Дерсі Л. Розробка програм для Android-пристроїв. Базові принципи [Текст]/Л. Дерсі, Ш. Кондер - Том 1. - Москва: Ексмо, 2014. - 598 с.

4) Джордж Шеферд. Програмування Microsoft Visual C++ .NET. 2011р.

5) Використання мобільних пристроїв [Електронний ресурс]. - Режим доступу: http://www.wi-life.ru/stati/wi-fi/marketingovye-stati2/mobiledevices-use-aruba-research

6) Якоб Нільсен, Ралука Будіу. Як створювати ідеально зручні програми для мобільних пристроїв. 2013р.

7) Основні етапи розробки мобільних додатків [Електронний ресурс] – Режим доступу: https://spark.ru/startup/componentix/blog/4499/ osnovnie-etapi-razrabotkimobilnih-prilozhenij

8) Програми в магазині Google Play - MSU-Life [Електронний ресурс]. - Режим доступу: https://play.google.com/store/apps/details?id=com.springlabs.msulife&hl=ua .

9) Програми в магазині Google Play - НГУЕУ [Електронний ресурс]. - Режим доступу: https://play.google.com/store/apps/details?id=burov.nsuem (дата звернення: 08.06.2019).

10) Програми в магазині Google Plau MTI Mobile [Електронний ресурс]. - Режим доступу:

https://play.google.com/store/apps/details?id=ru.edu.mti.mtimobile

42

11) Майєр Р. Програмування додатків для планшетних

комп'ютерів та смартфонів [Текст] / Р. Майєр - Москва: Ексмо, 2013. - 816

с.

12) Амелін К. С., Гранічин О. Н., Кияєв В. І., Корявко О. В..

Введення в розробку програм для мобільних платформ. [Текст]

Видавництво ВВМ, 2011.

13)Д. Хантер, К. Вербах. «Залучай і володарюй. Ігрове мислення на службі бізнесу»

14) Джейн МакГонігал: Реальність під питанням. Чому ігри роблять нас кращими і як вони можуть змінити світ, Манн, Іванов та Фербер, 2018 р.

Додаток А

Activity_login.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:background="@drawable/pagebkg"
    tools:context=".LoginActivity">


    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="30dp"
        app:cardCornerRadius="30dp"
        app:cardElevation="20dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:layout_gravity="center_horizontal"
            android:padding="24dp"
            android:background="@drawable/lavander_border">

            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Login"
                android:textSize="36sp"
                android:textAlignment="center"
                android:textStyle="bold"
                android:textColor="@color/lavender"/>

            <EditText
                android:layout_width="match_parent"
                android:layout_height="50dp"
                android:id="@+id/login_username"
                android:background="@drawable/lavander_border"
                android:layout_marginTop="40dp"
                android:padding="8dp"
                android:hint="Username"
                android:drawableLeft="@drawable/baseline_person_pin_24"
                android:drawablePadding="8dp"
                android:textColor="@color/black"/>

            <EditText
                android:layout_width="match_parent"
                android:layout_height="50dp"
                android:id="@+id/login_password"
                android:background="@drawable/lavander_border"
                android:layout_marginTop="20dp"
                android:padding="8dp"
                android:hint="Password"
```

```
                android:inputType="textPassword"
                android:drawableLeft="@drawable/baseline_lock_24"
                android:drawablePadding="8dp"
                android:textColor="@color/black"/>

            <Button
                android:layout_width="match_parent"
                android:layout_height="60dp"
                android:text="Login"
                android:id="@+id/login_button"
                android:textSize="18sp"
                android:layout_marginTop="30dp"
                android:backgroundTint="@color/lavender"
                app:cornerRadius = "20dp"/>

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:id="@+id/signupRedirectText"
                android:text="Not yet registered? Sign Up"
                android:layout_gravity="center"
                android:padding="8dp"
                android:layout_marginTop="10dp"
                android:textColor="@color/lavender"
                android:textSize="18sp"/>

        </LinearLayout>

    </androidx.cardview.widget.CardView>

    <FrameLayout
        android:id="@+id/fragment_container"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>


</LinearLayout>
```

activity_signup.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:background="@drawable/pagebkg"
    tools:context=".SignupActivity">

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="30dp"
        app:cardCornerRadius="30dp"
        app:cardElevation="20dp">

        <LinearLayout
```

```
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:layout_gravity="center_horizontal"
            android:padding="24dp"
            android:background="@drawable/lavander_border">

            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Sign Up"
                android:textSize="36sp"
                android:textAlignment="center"
                android:textStyle="bold"
                android:textColor="@color/lavender"/>

            <EditText
                android:layout_width="match_parent"
                android:layout_height="50dp"
                android:id="@+id/signup_name"
                android:background="@drawable/lavander_border"
                android:layout_marginTop="40dp"
                android:padding="8dp"
                android:hint="Name"
                android:drawableLeft="@drawable/baseline_person_24"
                android:drawablePadding="8dp"
                android:textColor="@color/black"/>

            <EditText
                android:layout_width="match_parent"
                android:layout_height="50dp"
                android:id="@+id/signup_email"
                android:background="@drawable/lavander_border"
                android:layout_marginTop="20dp"
                android:padding="8dp"
                android:hint="Email"
                android:drawableLeft="@drawable/baseline_email_24"
                android:drawablePadding="8dp"
                android:textColor="@color/black"/>


            <EditText
                android:layout_width="match_parent"
                android:layout_height="50dp"
                android:id="@+id/signup_username"
                android:background="@drawable/lavander_border"
                android:layout_marginTop="20dp"
                android:padding="8dp"
                android:hint="Username"
                android:drawableLeft="@drawable/baseline_person_pin_24"
                android:drawablePadding="8dp"
                android:textColor="@color/black"/>

            <EditText
                android:layout_width="match_parent"
                android:layout_height="50dp"
                android:id="@+id/signup_password"
                android:background="@drawable/lavander_border"
                android:layout_marginTop="20dp"
                android:padding="8dp"
                android:hint="Password"
                android:inputType="textPassword"
                android:drawableLeft="@drawable/baseline_lock_24"
```

```
                android:drawablePadding="8dp"
                android:textColor="@color/black"/>

            <Button
                android:layout_width="match_parent"
                android:layout_height="60dp"
                android:text="Sign Up"
                android:id="@+id/signup_button"
                android:textSize="18sp"
                android:layout_marginTop="30dp"
                android:backgroundTint="@color/lavender"
                app:cornerRadius = "20dp"/>

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:id="@+id/loginRedirectText"
                android:text="Already an user? Login"
                android:layout_gravity="center"
                android:padding="8dp"
                android:layout_marginTop="10dp"
                android:textColor="@color/lavender"
                android:textSize="18sp"/>

        </LinearLayout>

    </androidx.cardview.widget.CardView>

</LinearLayout>
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    tools:context=".HomeFragment">


    <LinearLayout
        android:layout_width="357dp"
        android:layout_height="73dp"
        android:layout_marginTop="100dp"
        android:layout_marginBottom="568dp">


        <TextView
            android:id="@+id/pushups_tv"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="20dp"
            android:layout_above="@+id/bottomNavigation"
            android:text="Push-ups"
            android:textColor="@color/lavender"
            android:textSize="36sp" />

        <Button
            android:layout_width="150dp"
            android:id="@+id/pushups_btn"
```

```
            android:layout_height="wrap_content"
            android:layout_marginStart="10dp"
            android:text="Start"
            android:textSize="25sp" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="357dp"
        android:layout_height="73dp"
        android:layout_marginTop="200dp"
        android:layout_marginBottom="568dp">


        <TextView
            android:id="@+id/press_vw"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="50dp"
            android:layout_above="@+id/bottomNavigation"
            android:text="Press"
            android:textColor="@color/lavender"
            android:textSize="36sp" />

        <Button
            android:layout_width="150dp"
            android:layout_height="wrap_content"
            android:id="@+id/press_btn"
            android:layout_marginStart="40dp"
            android:text="Start"
            android:textSize="25sp" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="357dp"
        android:layout_height="73dp"
        android:layout_marginTop="300dp"
        android:layout_marginBottom="568dp">


        <TextView
            android:id="@+id/pullups_tv"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="35dp"
            android:layout_above="@+id/bottomNavigation"
            android:text="Pull-ups"
            android:textColor="@color/lavender"
            android:textSize="36sp" />

        <Button
            android:layout_width="150dp"
            android:layout_height="wrap_content"
            android:id="@+id/pullups_btn"
            android:layout marginStart="20dp"
            android:text="Start"
            android:textSize="25sp" />

    </LinearLayout>

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/bottomNavigation"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="20dp"
        android:layout_marginStart="20dp"
        android:layout_marginBottom="30dp"
        android:layout_marginTop="30dp"
        android:background="@drawable/bottom_background"
        android:elevation="2dp"
        app:itemIconSize="30dp"
        app:itemIconTint="@drawable/item_selector"
        app:itemRippleColor="@android:color/transparent"
        app:labelVisibilityMode="unlabeled"
        app:menu="@menu/bottom_navmenu" />

</RelativeLayout>
```

Activity_profile.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/profilebkg"
    tools:context=".ProfileActivity">

    <ImageView
        android:id="@+id/profileImg"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_marginStart="155dp"
        android:layout_marginEnd="-60dp"
        android:layout_marginBottom="50dp"
        android:layout_marginTop="10dp"
        android:src="@drawable/sport"
        app:layout_constraintEnd_toEndOf="@id/linearLayout"
        app:layout_constraintStart_toStartOf="@id/linearLayout"
        app:layout_constraintTop_toTopOf="@id/linearLayout" />

    <TextView
        android:id="@+id/titleName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignTop="@+id/linearLayout"
        android:layout_marginStart="180dp"
        android:layout_marginTop="-90dp"
        android:layout_marginEnd="100dp"
        android:text="Name"
        android:textColor="@color/white"
        android:textSize="20sp"
        android:textStyle="bold" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/titleUsername"
        android:text="username"
        android:textColor="@color/white"
        android:textSize="18sp"
```

```
        android:layout_marginStart="170dp"
        android:layout_marginTop="150dp"
        />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/linearLayout"
        android:orientation="vertical"
        android:padding="10dp"
        android:background="@drawable/white_background"
        android:layout_marginStart="24dp"
        android:layout_marginEnd="24dp"
        android:layout_marginTop="200dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/titleUsername">



        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:padding="8dp"
            android:layout_marginLeft="10dp"
            android:layout_marginRight="10dp"
            android:layout_marginBottom="10dp"
            android:layout_marginTop="20dp"
            android:orientation="horizontal">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:id="@+id/name"
                android:layout_weight="1"
                android:text="Name"
                android:textStyle="bold"
                android:textColor="@color/lavender"
                android:textSize="18sp"/>

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:id="@+id/profileName"
                android:text="name"
                android:layout_weight="1"
                android:textAlignment="viewEnd"
                android:textColor="@color/grey"
                android:textSize="18sp"/>

        </LinearLayout>

        <androidx.appcompat.widget.AppCompatImageView
            android:layout_width="match_parent"
            android:layout_height="1dp"
            android:layout_margin="10dp"
            android:alpha="0.5"
            android:background="@color/lavender"/>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_margin="10dp"
```

```
            android:padding="8dp"
            android:orientation="horizontal">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:id="@+id/email"
                android:layout_weight="1"
                android:text="Email"
                android:textStyle="bold"
                android:textColor="@color/lavender"
                android:textSize="18sp"/>

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:id="@+id/profileEmail"
                android:text="email"
                android:layout_weight="1"
                android:textAlignment="viewEnd"
                android:textColor="@color/grey"
                android:textSize="18sp"/>

        </LinearLayout>

        <androidx.appcompat.widget.AppCompatImageView
            android:layout_width="match_parent"
            android:layout_height="1dp"
            android:layout_margin="10dp"
            android:alpha="0.5"
            android:background="@color/lavender"/>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_margin="10dp"
            android:padding="8dp"
            android:orientation="horizontal">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:id="@+id/username"
                android:layout_weight="1"
                android:text="Username"
                android:textStyle="bold"
                android:textColor="@color/lavender"
                android:textSize="18sp"/>

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:id="@+id/profileUsername"
                android:text="username"
                android:layout_weight="1"
                android:textAlignment="viewEnd"
                android:textColor="@color/grey"
                android:textSize="18sp"/>

        </LinearLayout>

        <androidx.appcompat.widget.AppCompatImageView
            android:layout_width="match_parent"
```

```
            android:layout_height="1dp"
            android:layout_margin="10dp"
            android:alpha="0.5"
            android:background="@color/lavender"/>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_margin="10dp"
            android:padding="8dp"
            android:orientation="horizontal">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:id="@+id/password"
                android:layout_weight="1"
                android:text="Password"
                android:textStyle="bold"
                android:textColor="@color/lavender"
                android:textSize="18sp"/>

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:id="@+id/profilePassword"
                android:text="password"
                android:layout_weight="1"
                android:textAlignment="viewEnd"
                android:textColor="@color/grey"
                android:textSize="18sp"/>

        </LinearLayout>

    </LinearLayout>

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/bottomNavigation"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="30dp"
        android:layout_marginEnd="30dp"
        android:layout_marginStart="30dp"
        android:layout_marginTop="30dp"
        android:background="@drawable/bottom_background"
        android:elevation="2dp"
        app:itemIconSize="30dp"
        app:itemIconTint="@drawable/item_selector"
        app:itemRippleColor="@android:color/transparent"
        app:labelVisibilityMode="unlabeled"
        app:menu="@menu/bottom_navmenu" />

</RelativeLayout>
```

Activity_press.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
```

```
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".PressActivity">


    <Button
        android:id="@+id/button_press_lv1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Level 1"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.12"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.08" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/tv_press_lv1"
        android:text="10 times"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.432"
        app:layout_constraintStart_toEndOf="@+id/button_press_lv1"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.094"/>

    <Button
        android:id="@+id/button_press_lv2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Level 2"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.12"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.214" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/tv_press_lv2"
        android:text="15 times"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.432"
        app:layout_constraintStart_toEndOf="@+id/button_press_lv2"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.227" />

    <Button
        android:id="@+id/button_press_lv3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Level 3"
```

```
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.12"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.362" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/tv_press_lv3"
        android:text="20 times"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.432"
        app:layout_constraintStart_toEndOf="@+id/button_press_lv3"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.37" />


</androidx.constraintlayout.widget.ConstraintLayout>
```

Activity_pullups.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".PullupsActivity">


    <Button
        android:id="@+id/button_pullups_lv1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Level 1"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.12"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.08" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/tv_pullups_lv1"
        android:text="10 times"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.432"
        app:layout_constraintStart_toEndOf="@+id/button_pullups_lv3"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.094"/>
```

```
    <Button
        android:id="@+id/button_pullups_lv2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Level 2"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.12"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.214" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/tv_pullups_lv2"
        android:text="15 times"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.432"
        app:layout_constraintStart_toEndOf="@+id/button_pullups_lv2"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.227" />

    <Button
        android:id="@+id/button_pullups_lv3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Level 3"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.12"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.362" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/tv_pullups_lv3"
        android:text="20 times"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.432"
        app:layout_constraintStart_toEndOf="@+id/button_pullups_lv3"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.37" />


</androidx.constraintlayout.widget.ConstraintLayout>
```

Activity_pushups.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
```

```
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".PushupsActivity">


    <Button
        android:id="@+id/button_pushups_lv1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Level 1"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.12"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.08" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/tv_pushups_lv1"
        android:text="10 times"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.432"
        app:layout_constraintStart_toEndOf="@+id/button_pushups_lv1"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.094"/>

    <Button
        android:id="@+id/button_pushups_lv2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Level 2"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.12"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.214" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/tv_pushups_lv2"
        android:text="15 times"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.432"
        app:layout_constraintStart_toEndOf="@+id/button_pushups_lv2"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.227" />

    <Button
        android:id="@+id/button_pushups_lv3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:text="Level 3"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.12"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.362" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/tv_pushups_lv3"
        android:text="20 times"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.432"
        app:layout_constraintStart_toEndOf="@+id/button_pushups_lv3"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.37" />


</androidx.constraintlayout.widget.ConstraintLayout>
```

Activity_timer.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".TimerActivity"
    android:background="@color/white">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="64dp"
        android:background="@drawable/clock"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/countdown_timer"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="00:00:00"
        android:textSize="50sp"
        app:layout_constraintBottom_toTopOf="@+id/start"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView"
        app:layout_constraintVertical_bias="0.532" />

    <Button
        android:id="@+id/start"
```

```
        android:layout_width="171dp"
        android:layout_height="73dp"
        android:backgroundTint="@color/orange"
        android:text="Start"
        android:textColor="@color/design_default_color_on_secondary"
        android:textColorLink="#F3F1EE"
        android:textSize="20dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView" />


</androidx.constraintlayout.widget.ConstraintLayout>
```

Додаток Б

HelperClass.java

```
package com.example.healthsport;

public class HelperClass {

        String name, email, username, password;

        public String getName() {
            return name;
        }

        public void setName(String name) {
            this.name = name;
        }

        public String getEmail() {
            return email;
        }

        public void setEmail(String email) {
            this.email = email;
        }

        public String getUsername() {
            return username;
        }

        public void setUsername(String username) {
            this.username = username;
        }

        public String getPassword() {
            return password;
        }

        public void setPassword(String password) {
            this.password = password;
        }
```

```
        public HelperClass(String name, String email, String username, String
password) {
            this.name = name;
            this.email = email;
            this.username = username;
            this.password = password;
        }

        public HelperClass() {
        }
    }
```

LoginActivity.java

```
package com.example.healthsport;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.FragmentManager;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.Query;
import com.google.firebase.database.ValueEventListener;
import java.util.Objects;

public class LoginActivity extends AppCompatActivity {

    EditText loginUsername, loginPassword;
    Button loginButton;
    TextView signupRedirectText;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_login);

        loginUsername = findViewById(R.id.login_username);
        loginPassword = findViewById(R.id.login_password);
        loginButton = findViewById(R.id.login_button);
        signupRedirectText = findViewById(R.id.signupRedirectText);
        loginButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (!validateUsername() | !validatePassword()) {
                } else {
                    checkUser();
                }
            }
        });
        signupRedirectText.setOnClickListener(new View.OnClickListener() {
            @Override
```

```
            public void onClick(View view) {
                Intent intent = new Intent(LoginActivity.this,
SignupActivity.class);
                startActivity(intent);
            }
        });
    }
    public Boolean validateUsername() {
        String val = loginUsername.getText().toString();
        if (val.isEmpty()) {
            loginUsername.setError("Username cannot be empty");
            return false;
        } else {
            loginUsername.setError(null);
            return true;
        }
    }
    public Boolean validatePassword(){
        String val = loginPassword.getText().toString();
        if (val.isEmpty()) {
            loginPassword.setError("Password cannot be empty");
            return false;
        } else {
            loginPassword.setError(null);
            return true;
        }
    }
    public void checkUser(){
        String userUsername = loginUsername.getText().toString().trim();
        String userPassword = loginPassword.getText().toString().trim();
        DatabaseReference reference =
FirebaseDatabase.getInstance().getReference("users");
        Query checkUserDatabase =
reference.orderByChild("username").equalTo(userUsername);
        checkUserDatabase.addListenerForSingleValueEvent(new
ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if (snapshot.exists()){
                    loginUsername.setError(null);
                    String passwordFromDB =
snapshot.child(userUsername).child("password").getValue(String.class);
                    if (passwordFromDB.equals(userPassword)) {
                        loginUsername.setError(null);
                        String nameFromDB =
snapshot.child(userUsername).child("name").getValue(String.class);
                        String emailFromDB =
snapshot.child(userUsername).child("email").getValue(String.class);
                        String usernameFromDB =
snapshot.child(userUsername).child("username").getValue(String.class);
                        Intent intent = new Intent(LoginActivity.this,
ProfileActivity.class);
                        // Получаем менеджер фрагментов
                        FragmentManager fragmentManager =
getSupportFragmentManager();
// Создаем экземпляр ProfileFragment
                        ProfileFragment profileFragment = new
ProfileFragment();
// Заменяем текущий фрагмент на ProfileFragment
                        fragmentManager.beginTransaction()
                                .replace(R.id.fragment_container,
profileFragment)
                                .commit();
```

```
                        intent.putExtra("name", nameFromDB);
                        intent.putExtra("email", emailFromDB);
                        intent.putExtra("username", usernameFromDB);
                        intent.putExtra("password", passwordFromDB);
                        startActivity(intent);
                    } else {
                        loginPassword.setError("Invalid Credentials");
                        loginPassword.requestFocus();
                    }
                } else {
                    loginUsername.setError("User does not exist");
                    loginUsername.requestFocus();
                }
            }
            @Override
            public void onCancelled(@NonNull DatabaseError error) {
            }
        });

    }
}
```

SignupActivity.java

```
package com.example.healthsport;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;



public class SignupActivity extends AppCompatActivity {

    EditText signupName, signupUsername, signupEmail, signupPassword;
    TextView loginRedirectText;
    Button signupButton;
    FirebaseDatabase database;
    DatabaseReference reference;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_signup);

        signupName = findViewById(R.id.signup_name);
        signupEmail = findViewById(R.id.signup_email);
        signupUsername = findViewById(R.id.signup_username);
        signupPassword = findViewById(R.id.signup_password);
        loginRedirectText = findViewById(R.id.loginRedirectText);
        signupButton = findViewById(R.id.signup_button);
        signupButton.setOnClickListener(new View.OnClickListener() {
            @Override
```

```
            public void onClick(View view) {
                database = FirebaseDatabase.getInstance();
                reference = database.getReference("users");
                String name = signupName.getText().toString();
                String email = signupEmail.getText().toString();
                String username = signupUsername.getText().toString();
                String password = signupPassword.getText().toString();
                HelperClass helperClass = new HelperClass(name, email,
username, password);
                reference.child(username).setValue(helperClass);
                Toast.makeText(SignupActivity.this, "You have signup
successfully!", Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(SignupActivity.this,
LoginActivity.class);
                startActivity(intent);
            }
        });
        loginRedirectText.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(SignupActivity.this,
LoginActivity.class);
                startActivity(intent);
            }
        });
    }

    }
```

ProfileActivity.java

```
package com.example.healthsport;

import android.os.Bundle;


import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.widget.Button;
import android.widget.TextView;

import com.google.android.material.bottomnavigation.BottomNavigationView;
import androidx.annotation.NonNull;


import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import com.google.firebase.database.ValueEventListener;


public class ProfileActivity extends AppCompatActivity {

    TextView profileName, profileEmail, profileUsername, profilePassword;
    TextView titleName, titleUsername;
    Button editProfile;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_profile);

        BottomNavigationView bottomNavigationView =
findViewById(R.id.bottomNavigation);
        bottomNavigationView.setSelectedItemId(R.id.profile);

        bottomNavigationView.setOnItemSelectedListener(item -> {
            int itemId = item.getItemId();
            if (itemId == R.id.home) {
                startActivity(new Intent(getApplicationContext(),
MainActivity.class));
                overridePendingTransition(R.anim.slide_in_right,
R.anim.slide_out_left);
                finish();
                return true;
            } else return itemId == R.id.profile;
        });

        profileName = findViewById(R.id.profileName);
        profileEmail = findViewById(R.id.profileEmail);
        profileUsername = findViewById(R.id.profileUsername);
        profilePassword = findViewById(R.id.profilePassword);
        titleName = findViewById(R.id.titleName);
        titleUsername = findViewById(R.id.titleUsername);

        showAllUserData();
    }

    public void showAllUserData(){
        Intent intent = getIntent();
        String nameUser = intent.getStringExtra("name");
        String emailUser = intent.getStringExtra("email");
        String usernameUser = intent.getStringExtra("username");
        String passwordUser = intent.getStringExtra("password");
        titleName.setText(nameUser);
        titleUsername.setText(usernameUser);
        profileName.setText(nameUser);
        profileEmail.setText(emailUser);
        profileUsername.setText(usernameUser);
        profilePassword.setText(passwordUser);
    }
}
```

PressActivity.java

```
package com.example.healthsport;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
```

```
public class PressActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_press);
        Button button_press_lv1 = findViewById(R.id.button_press_lv1);
        Button button_press_lv2 = findViewById(R.id.button_press_lv2);
        Button button_press_lv3 = findViewById(R.id.button_press_lv3);

        button_press_lv1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(PressActivity.this,
TimerActivity.class);
                startActivity(intent);
                finish();

            }
        });
        button_press_lv2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(PressActivity.this,
TimerActivity.class);
                startActivity(intent);
                finish();
            }
        });
        button_press_lv3.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(PressActivity.this,
TimerActivity.class);
                startActivity(intent);
                finish();
            }
        });
    }
}
```

PullupsActivity.java

```
package com.example.healthsport;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class PullupsActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_pullups);
        Button button_pullups_lv1 = findViewById(R.id.button_pullups_lv1);
        Button button_pullups_lv2 = findViewById(R.id.button_pullups_lv2);
        Button button_pullups_lv3 = findViewById(R.id.button_pullups_lv3);

        button_pullups_lv1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(PullupsActivity.this,
TimerActivity.class);
                startActivity(intent);
                finish();

            }
        });
        button_pullups_lv2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(PullupsActivity.this,
TimerActivity.class);
                startActivity(intent);
                finish();
            }
        });
        button_pullups_lv3.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(PullupsActivity.this,
TimerActivity.class);
                startActivity(intent);
                finish();
            }
        });
    }
}
```

PushupsActivity.java

```
package com.example.healthsport;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class PushupsActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_pushups);

        Button button_pushups_lv1 = findViewById(R.id.button_pushups_lv1);
```

```
        Button button_pushups_lv2 = findViewById(R.id.button_pushups_lv2);
        Button button_pushups_lv3 = findViewById(R.id.button_pushups_lv3);

        button_pushups_lv1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(PushupsActivity.this,
TimerActivity.class);
                startActivity(intent);
                finish();

            }
        });
        button_pushups_lv2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(PushupsActivity.this,
TimerActivity.class);
                intent.putExtra("timeInMillis", 30000);
                startActivity(intent);
                finish();
            }
        });
        button_pushups_lv3.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(PushupsActivity.this,
TimerActivity.class);
                intent.putExtra("timeInMillis", 60000);
                startActivity(intent);
                finish();
            }
        });



    }
}
```

TimerActivity.java

```
package com.example.healthsport;

import android.media.MediaPlayer;
import android.os.Bundle;
import android.os.CountDownTimer;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import java.util.Locale;

public class TimerActivity extends AppCompatActivity {

    TextView countdownTimer;
```

```
    CountDownTimer timer;

    Button start;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_timer);

        countdownTimer = findViewById(R.id.countdown_timer);
        start = findViewById(R.id.start);

        start.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startTime();
            }
        });

    }

    private void startTime() {
        timer = new CountDownTimer(30000, 1000) {
            @Override
            public void onTick(long millisUntilFinished) {
               long hours = (millisUntilFinished / 1000) / 3600;
               long minutes = ((millisUntilFinished / 1000) % 3600) / 60;
               long seconds = (millisUntilFinished / 1000) % 60;
               String timeFormatted = String.format(Locale.getDefault(),
"%02d:%02d:%02d", hours, minutes, seconds);
               countdownTimer.setText(timeFormatted);
            }

            @Override
            public void onFinish() {
                countdownTimer.setText("00:00:00");
                Toast.makeText(TimerActivity.this, "Good Work",
Toast.LENGTH_SHORT).show();
                MediaPlayer alarm = MediaPlayer.create(TimerActivity.this,
R.raw.win31);
                alarm.start();

                finish();
            }
        }.start();
    }
}
```