

Мобільний застосунок для вдосконалення самодисципліни в ігровій формі

«Gamify Your Life»

Опис проекту

Тип проекту:

Android-застосунок

Мета проекту:

Створення зручного та функціонального мобільного додатку, який допоможе людям дисциплінувати себе та свою рутину. За допомогою додатку, користувачі зможуть відчути себе у грі, в якій необхідно запрацьовувати кредити (наприклад, виконуючи завдання або утримуючись від певних поганих звичок), за які можна купити винагороди або час на певний вид відпочинку (що визначається самим користувачем).

Актуальність:

Проблема дисциплінованості в повсякденному житті була і буде актуальною завжди. У кожної людини є багато цілей, задач, які необхідно виконати за певний період часу. Це можуть бути як маленькі рутинні справи, які не займають багато часу, так і більш складні задачі, для виконання яких необхідно прикласти досить багато зусиль. Завжди тримати усіх їх в пам'яті – досить непросто, а для того, щоб успішно виконувати всі справи, потрібно завжди слідкувати за тим, щоб правильно розподіляти свій час між роботою та відпочинком. Також у більшості людей є погані звички, які забирають надто багато часу, який краще було б витратити на щось корисне. Саме тому наш проект буде корисний людям для збільшення щоденної продуктивності та випрацювання дисциплінованості.

Зміст проекту:

У додатку користувачі зможуть створювати:

- списки задач та самі задачі, з можливістю вибрати дедлайн для їх виконання. (Для кожної задачі користувач може визначати кількість кредитів, яку отримає у разі успішного виконання, та штраф, якщо не встигне виконати справу до кінця визначеного терміну).
- звички, яких необхідно позбутись, або ж звички, які потрібно ввести в свою рутину. (Для кожної звички користувач може визначати кількість кредитів, яку буде отримувати за кожен день успішного виконання, та штраф, у разі порушення умов певної звички).
- винагороди, які можна купити за накопичені кредити. (У якості винагороди, користувач може обрати будь-що: похід в кіно, прогулянку з друзями, зіграти в комп'ютерну гру, покупку будь-якої речі в магазині. Це може бути будь-яка активність, чи річ, якою користувач готовий себе винагородити в реальному житті)

Додаток підтримуватиме два режими: офлайн та онлайн – в обидвох з яких будуть доступні усі можливості, описані вище. В офлайн режимі не буде реєстрації, та вся інформація, створена користувачем, зберігатиметься на одному девайсі. В онлайн режимі для того, щоб користуватись додатком, необхідно буде зареєструватись за допомогою Google-акаунту. Також з'явиться можливість пошуку інших користувачів та можливість додаватись у друзі, що, у свою чергу, дозволить переглядати звички

друзів, яких вони хочуть позбутись. Уся інформація, створена користувачем, зберігатиметься в хмарі, а отже, користувач зможе доступатись до свого акаунту з різних пристроїв.

Функціональні вимоги

1. Загальні вимоги

1.1. Автентифікація

- 1.1.1. Користувач повинен мати змогу працювати із застосунком без доступу до інтернету.
- 1.1.2. Користувач повинен мати змогу автентифікуватись у застосунок за допомогою Google акаунта при наявності доступу до інтернету.
- 1.1.3. Користувач повинен мати змогу змінити режим роботи застосунку (онлайн/офлайн режими) та змінити акаунт (в онлайн режимі).

1.2. Завдання

- 1.2.1. Користувач повинен мати змогу створювати, переглядати, редагувати та видаляти завдання.
- 1.2.2. При створенні нового завдання користувач повинен мати змогу вказувати назву та опис завдання.
- 1.2.3. При створенні нового завдання користувач повинен мати змогу вказувати дедлайн, до якого необхідно виконати завдання.
- 1.2.4. При створенні нового завдання користувач повинен мати змогу вказувати кількість кредитів, що можна отримати за успішне або втратити за неуспішне виконання завдання.
- 1.2.5. Користувач повинен мати змогу робити копію завдання.
- 1.2.6. Користувач повинен мати змогу позначати завдання як виконані або провалені.
- 1.2.7. Користувач повинен отримувати нагороду або штраф в залежності від успішності виконання завдання.

1.3. Списки завдань

- 1.3.1. Користувач повинен мати змогу створювати, переглядати, редагувати та видаляти списки завдань.
- 1.3.2. При створенні списку завдань користувач повинен мати змогу дати йому назву.
- 1.3.3. Користувач повинен мати змогу додавати та видаляти завдання із списків завдань.
- 1.3.4. Користувач повинен мати змогу позначати список завдань як виконаний.
- 1.3.5. Користувач повинен мати змогу перемикатись між різними списками завдань на сторінці із цими списками.

1.4. Звички

- 1.4.1. Користувач повинен мати змогу створювати, переглядати, редагувати та видаляти звички.
- 1.4.2. При створенні звички користувач повинен мати змогу вказати назву звички.
- 1.4.3. При створенні звички користувач повинен мати змогу вказати денну нагороду за дотримання та штраф за недотримання звички.
- 1.4.4. При створенні звички користувач повинен мати змогу вказати останню дату недотримання звички.

- 1.4.5. Користувач повинен мати змогу бачити час, що пройшов із моменту останнього недотримання звички.
- 1.4.6. Користувач повинен отримувати нагороду або штраф в залежності від дотримання чи недотримання звички.

1.5. Нагороди

- 1.5.1. У користувача повинна бути можливість створювати, переглядати, редагувати та видаляти нагороди.
- 1.5.2. При створенні нагороди у користувача повинна бути можливість вказати назву нагороди.
- 1.5.3. При створенні нагороди у користувача повинна бути можливість задати ціну нагороди або використати значення ціни за замовчуванням.
- 1.5.4. При створенні нагороди у користувача повинна бути можливість додати певне зображення з пристрою до нагороди.
- 1.5.5. У користувача повинна бути можливість купувати нагороди за наявні кредити.

1.6. Налаштування

- 1.6.1. При збереженні завдання, списку завдань, звички чи нагороди без вказаної назви, користувач повинен побачити повідомлення про помилку.
- 1.6.2. При збереженні завдання, звички чи нагороди із невказаними полями, що мають значення за замовчуванням, для таких полів будуть обрані ці значення.
- 1.6.3. Користувач повинен мати змогу редагувати значення змінних за замовчуванням.

2. Додаткові вимоги до онлайн частини застосунку

2.1. Друзі

- 2.1.1. Користувач повинен мати змогу переглядати профілі інших користувачів.
- 2.1.2. Користувач повинен мати змогу шукати, додавати та видаляти друзів.
- 2.1.3. Користувач повинен мати змогу надсилати запити на додавання в друзі.
- 2.1.4. Користувач повинен мати змогу прийняти або відхилити запит на додавання в друзі.
- 2.1.5. Користувач може позначати звичку як таку, якою можна ділитись із друзями (друзям буде вона доступна для перегляду).

Нефункціональні вимоги

1. Доступність (Accessibility)

Застосунок буде доступний для користування майже увесь час:

- в офлайн режимі: незалежний ні від яких сервісів, працює лише локально, а отже і буде доступний для користувачів постійно
- в онлайн режимі: залежить він сервісів Google, зокрема Firebase, які забезпечують безперервний доступ, а отже застосунок в онлайн режимі може бути недоступний (не мати можливості для авторизації та підтягування даних з хмари) тільки за ситуації, коли ці сервіси дадуть збій (що дуже мало ймовірно).

2. Надійність (Reliability)

У разі відсутності інтернет зв'язку:

- застосунок продовжує функціонувати без збоїв

- під час користування застосунком онлайн, дані повинні кешуватись, щоб у разі втрати інтернет зв'язку, була можливість їх переглядати
- застосунком можна користуватись у офлайн режимі, і тоді всі дані будуть зберігатись локально на пристрої.

3. Вимоги до зберігання даних (Data storage requirements)

Оскільки в застосунку буде два режими (онлайн на офлайн), повинні бути забезпечені надійні та довгострокові засоби зберігання даних:

- офлайн режим: дані зберігатимуться локально на девайсі користувача.
- онлайн режим: використання хмарного сервісу Cloud Firestore, який надає безкоштовні та надійні засоби для віддаленого зберігання даних. Безкоштовний варіант має певні обмеження: по к-сті записів та читань з бази даних на день, та по розмірах переданої інформації на день, але в майбутньому є можливість придбати платну підписку, у разі подальшого розвитку додатку і збільшення його популярності.

4. Масштабованість (Scalability)

Вертикальне масштабування: оскільки ми використовуємо сервіс Firebase безкоштовно, спочатку ми матимемо обмежену к-сть операцій з базою даних за день, к-сть записів, а також обмежену к-сть місця для збереження медіа даних. В майбутньому це можна масштабувати придбанням платної підписки. Тоді обмежень на збереження даних не буде.

5. Вимоги до зручності у використанні (Usability Requirements)

Усі меню, форми, кнопки і т. д. повинні мати очевидну для користувачів поведінку, назви прозоро описувати, які дії будуть виконані.

6. Безпека (Security)

- Дані користувачів не передаватимуться третім лицам.
- Користувачі можуть видалити всі дані, які були перед цим збережені в хмарі.
- Завдяки авторизації за допомогою Google акаунту, користувачам не потрібно надавати ніякі паролі.

7. Конфігурованість (Configurability)

Узагальненість і абстрактність повинні дати змогу конфігурувати окремі компоненти, що також сприятиме їх повторному використанню.

8. Продуктивність (Performance)

- Запити до бази даних повинні бути написані найефективнішим чином та виконуватись з найменшим часом відклику.
- Запити на повернення великих списків потрібно оптимізувати поступовим підвантаженням елементів (пачками обмеженого розміру).
- Якщо у користувача нестабільне підключення до інтернету, або він заслабкий, то програма не зависає, а відображає візуальний ефект завантаження даних.

- Дані повинні зберігатись якомога ефективнішим чином, як локально на телефоні, так і у віддаленій хмарі, медіа дані повинні бути зжаті в якості.

9. Вимоги до повторного використання реалізацій чи компонентів застосунку (Reusability)

Відсутність або мінімальне дублювання коду, застосування узагальнень, абстракцій та ієрархій, що дасть змогу використовувати класи, віджети, та інші компоненти повторно.

10. Вимоги до розширюваності (Extensibility)

Застосунок повинен мати можливість додавати нові екрани та меню без необхідності змінювати наявний код.

11. Вимоги до підтримки системи (Supportability)

Аби в майбутньому розробникам було легше підтримувати проект, або влитись в проект новим:

- застосунок повинен мати хорошу архітектуру, чистий код та прозору поведінку
- процеси відлагодження, аналізу помилок та проблем в роботі повинні бути максимально простими

12. Вимоги до можливостей тестування (Testability)

Застосунок повинен мати можливість ручного тестування на всіх етапах розробки шляхом завантаження та тестування версій на особистих Android-девайсах.

13. Вимоги до можливостей локалізації (Localizability)