

Szegedi Tudományegyetem
Informatikai Intézet

SZAKDOLGOZAT

Kun Réka Bianka

2023

**Szegedi Tudományegyetem
Informatikai Intézet**

Rendezvény szervező mobilalkalmazás

Szakdolgozat

Készítette:

Kun Réka Bianka
programtervező
informatikus szakos
hallgató

Témavezető:

Kiss-Vetráb Mercedes
egyetemi docens

Szeged
2023

Feladatkiírás

A szakdolgozatom témája egy mobilalkalmazás megvalósítása volt, melynek célja az volt, hogy a felhasználók számára lehetővé tegye saját események létrehozását és a környezetükben megtalálható események megtekintését. Az alkalmazás funkciói között szerepelt az események keresése és szűrése, valamint a részvételre való jelentkezés lehetősége. Az alkalmazás designjának kialakításakor a felhasználói élményre és az egyszerű használatra fektettem a hangsúlyt, így a felület letisztult és felhasználóbarát. Az alkalmazás célja, hogy megkönnyítse az emberek számára az események megtalálását és részvételét.

Tartalmi összefoglaló

- **A téma megnevezése:**

Rendezvény szervező mobilalkalmazás.

- **A megadott feladat megfogalmazása:**

Egy olyan mobilalkalmazás megvalósítása, amelynek használata egyszerű és könnyen átlátható a felhasználók számára. Az alkalmazás lehetőséget biztosít a felhasználóknak, hogy könnyedén hozzanak létre és testre szabjanak saját eseményeket, és megosszák azokat a többi felhasználóval. Az alkalmazás lehetővé teszi a felhasználók számára, hogy jelezzék részvételi szándékukat az eseményeken, amelyeken részt kívánnak venni, valamint értékelhessék azon eseményeket, amelyeken már részt vettek. Továbbá, az alkalmazásban lehetőség van azon események böngészésére is, amelyek a közelben kerülnek megrendezésre. A felhasználói élmény fokozása és az alkalmazás egyszerű kezelhetősége kiemelt fontosságú volt az alkalmazás tervezése során.

- **A megoldási mód:**

Az alkalmazás fejlesztése során fontos volt számomra, hogy az alkalmazás felhasználói számára elérhetővé tegyem az aktuális helyzet meghatározását, így az alkalmazásban be kellett építenem egy olyan funkciót, amely lehetővé teszi a felhasználó számára, hogy meghatározza a pontos helyzetét. Ennek érdekében implementáltam egy helymeghatározó modult, amelynek segítségével a felhasználó pontos helyzete meghatározható és az alkalmazás használata ezzel még kényelmesebbé válik. Ezen funkció beépítése további lehetőségeket nyújt az alkalmazás felhasználóinak, például megkönnyíti az események keresését, amelyek a felhasználó közelében kerülnek megrendezésre.

- **Alkalmazott eszközök, módszerek:**

Az alkalmazás elkészítéséhez a Flutter mobilalkalmazás-fejlesztési keretrendszert használtam, amely a Dart programozási nyelven alapul. Az adatok kezelésére a Google Firebase adatbázis-szolgáltatását alkalmaztam a Firestore-t.

- **Elért eredmények:**

Egy olyan mobilalkalmazás, amely felhasználóbarát és intuitív. Könnyen integrálódik az eszközökkel.

- **Kulcsszavak:**

Firebase, Firestore, Dart, Flutter, design.

Tartalomjegyzék

Feladatkiírás	4
Tartalmi összefoglaló.....	5
Tartalomjegyzék.....	6
Motiváció	7
1.Mobilalkalmazás fejlesztés	7
1.1. Natív és multiplatform fejlesztői eszközök története	7
1.2. Flutter és Dart.....	8
2. UX és UA design	10
2.1. Történeti áttekintő	11
2.2. Főbb alkalmazott tervezési szabályok.....	12
3. Moodboard és logo.....	ERROR! BOOKMARK NOT DEFINED.
3.1. Moodboard.....	Error! Bookmark not defined.
3.2. Logo.....	Error! Bookmark not defined.
4. Képernyőtervek	ERROR! BOOKMARK NOT DEFINED.
5. Fejlesztési folyamat.....	16
5.1. Trello	16
5.2. Github	Error! Bookmark not defined.
6. Alkalmazás szerkezeti felépítése	7
6.1. MVVM	7
6.2. Adatbázis.....	7
6.3. Üzleti logika.....	7
6.4. Kész alkalmazás	7
Irodalomjegyzék	17
Nyilatkozat.....	18

Motiváció

Az események szervezésére szolgáló alkalmazások széles körben elterjedtek, mivel hatékonyabbá teszik a rendezvények, találkozók és események tervezését és lebonyolítását. Ezek az alkalmazások lehetővé teszik a felhasználók számára, hogy programokat tervezzenek, információkat osszanak meg és nyomon követhessék az aktuális rendezvényeket. Az események szervezésére szolgáló alkalmazások időt és energiát takarítanak meg a felhasználók számára, lehetővé téve a rendezvények zökkenőmentes lebonyolítását.

Egy olyan mobilalkalmazás elkészítése volt a célom, amelyben ezek a tulajdonságok megtalálhatóak, valamint modern és a felhasználók számára könnyen kezelhető. Az alkalmazás megtervezésénél fontos szempont volt, hogy a felhasználói felülete vonzó, átlátható és könnyen kezelhető legyen.

1. Mobilalkalmazás fejlesztés

Az okostelefonokra vagy táblagépekre szánt mobilalkalmazások fejlesztése során egy olyan folyamatot kell végigvinni, amelynek során a cél az, hogy egy hatékony és felhasználóbarát alkalmazást hozzanak létre. Az alkalmazások platform-specifikusan, azaz az Android és iOS rendszerekre külön-külön készülnek, azonban ma már léteznek olyan keretrendszerek, például a Flutter, amelyek lehetővé teszik az alkalmazások egyszeri fejlesztését, majd több platformon történő futtatását.

A mobilalkalmazásunk sikeressége érdekében alapos tervezés szükséges, amely magában foglalja a piackutatást és az azonos piaci rést célzó alkalmazások tanulmányozását. Az alkalmazás fejlesztése során olyan szoftvereket hozunk létre, amelyek mobil eszközökre készülnek. Kiemelten fontos, hogy az alkalmazás egyértelműen mutassa be az egyedi funkcióit, amelyek megkülönböztetik az adott appot másoktól. Az egyedi mobilalkalmazás ma már kulcsfontosságú szerepet játszik a legtöbb cég életében.

A mobilalkalmazás fejlesztés több fontos lépésből tevődik össze. Az első ilyen lépés a stratégia meghatározása, amely során a fejlesztő ötleteket gyűjt a fejlesztendő alkalmazással kapcsolatban. Egy jó mobilalkalmazás egyedi és felhasználóbarát. Fontos szempont, hogy a fejlesztő ismerje a célközönséget, és feltérképezze azok igényeit. Ezt követően az alkalmazás tervezése következik, amely magában foglalja az alkalmazás architektúrájának és a felhasználói felület tervezésének kidolgozását. Ebben a szakaszban a fejlesztők a szoftver szerkezetére és az interfész kialakítására fókuszálnak. A következő fázisban a kódok megírása következik, amelyek az alkalmazás működését határozzák meg. Fontos szempont a fejlesztési

környezet, az eszközök és a nyelv kiválasztása is. Nélkülözhetetlen szakasz a tesztelési szakasz, amely során a fejlesztők tesztelik az alkalmazást, annak érdekében, hogy megbizonyosodjanak annak helyes működéséről.

Az alkalmazásfejlesztés egy összetett folyamat, amely a vállalati és felhasználói igényeket egyaránt figyelembe veszi, hogy hatékony és használható alkalmazások jöjjenek létre.

1.1.Natív és multiplatform fejlesztői eszközök története

A fejlesztés során alkalmazott technológiák és környezetek alapján három nagy kategóriába sorolhatók a mobilalkalmazások, ezek a webalkalmazások, natív alkalmazások és a multiplatform alkalmazások.

A natív fejlesztés az első módszer volt a mobilalkalmazások létrehozására, amely során minden platformra külön-külön kellett írni az alkalmazást, mivel a platformok különböző nyelveket és eszközöket használtak. Az Android alkalmazásokat Kotlin vagy Java nyelven, az iOS alkalmazásokat pedig Objective C vagy Swift nyelven lehetett írni. A Swift az iOS által kifejlesztett programozási nyelv. A natív fejlesztés lehetővé tette a fejlesztők számára, hogy teljes mértékbe kihasználhassák a platform adottságait, így az alkalmazások stabilak voltak, azonban az alkalmazások karbantartása, valamint megvalósítása időigényes volt.

A natív alkalmazások előnye, hogy egyetlen platformra vannak optimalizálva, ezáltal gyorsan és könnyedén futnak. Intuitív kezelést és egységes kialakítást valósít meg.

Azoknak az alkalmazásoknak a fejlesztésére ajánlott, amelyek sok funkcióval rendelkeznek, valamint egyes funkciói hardver alapokra épülnek, mint például a GPS, telefon kamerája, bluetooth, mivel a programozóknak közvetlen hozzáférésük van a telefon hardveréhez.

A multiplatform fejlesztési környezetek megjelenésével lehetővé vált az alkalmazások egyszerűbb és költséghatékonyabb fejlesztése több platformra egyszerre. A multiplatform megközelítés lehetővé teszi, hogy az alkalmazást egyszer írják meg, majd a platform-specifikus fordítóprogramok segítségével átalakítsák az alkalmazást az összes platformra. Az ilyen keretrendszerek közé tartoznak a React Native, a Xamarin, a Flutter, a PhoneGap és sok más.

A multiplatform megközelítés előnyei közé tartozik, hogy az alkalmazások egyszerre írhatók meg több platformra, így időt és pénzt takarítanak meg a fejlesztők számára. Az ilyen keretrendszerek lehetővé teszik a fejlesztők számára, hogy közös kódbázist használjanak, és ugyanazokat az eszközöket és nyelveket használják minden platformra, így egyszerűbb a karbantartás és a frissítés is. Azonban a multiplatform megközelítés hátránya, hogy a

platformok specifikus funkcióinak kivételezése nehezebb, így az alkalmazások teljesítménye és felhasználói élménye nem mindig olyan jó, mint a natív alkalmazásoké.

A programozás gyakran merül fel a kérdés, hogy melyik a legjobb választás: natív vagy a multiplatform alkalmazások, és bár mindegyiknek megvan a maga előnye és hátránya, egy nagy költségvetéssel rendelkező vállalkozás számára egyértelműen a natív alkalmazások fejlesztése a legjobb választás. A natív alkalmazások az ideális választások, ha kiemelten fontos a testreszabhatóság és a felhasználói élmény.

1.2. Flutter és Dart

A Flutter egy nyílt forráskódú mobilalkalmazás-keretrendszer, amelyet a Google által fejlesztett és fenntartott. Segítségével gyorsan és hatékonyan lehet platformfüggetlen mobilalkalmazásokat készíteni. A Flutter egyetlen kódbázisból képes létrehozni alkalmazásokat, amelyek egyaránt működnek iOS, Android és a weben is. Lehetővé teszi az alkalmazások reszponzív tervezését, és a különböző eszközökön való egységes megjelenítést. Megvalósítható benne az aszinkron programozás, amely segíti az alkalmazások teljesítményét. Különböző platformokat támogat, mint például az iOS, Linux, Windows, Android, macOS és a webet is. A Flutterben készített alkalmazások általában gyorsabbak és hatékonyabbak, mint a hozzá hasonló natív alkalmazások. Széles körű könyvtárral rendelkezik, amely segítségével az alkalmazáshoz különböző funkciók és szolgáltatások adhatók.

A Flutter keretrendszerben az alkalmazás felhasználói felülete widgetekből épül fel. A widgetek használatával az alkalmazás fejlesztése egyszerűbbé és hatékonyabbá válik. Könnyen testreszabhatóak, újrafelhasználhatók és kombinálhatóak, annak érdekében, hogy a felhasználói felület kívánt funkcióit megvalósíthassák. Egy Flutter alkalmazás képernyő nézete attól függ, hogy az alkalmazás felépítéséhez milyen widgeteket használtak fel. Az alkalmazás kódjának szerkezete a widgetek fája. A widgetek állapotát és működése egyszerűen szabályozható a StatelessWidget és a StatefulWidget osztályok alkalmazásával.

A StatelessWidget, a Flutterban használt widget típus, amely nem tartalmaz változó állapotot, így egyszerűbb a megvalósítása. Általában csak adatokat jelenítenek meg, vagy statikus elemeket tartalmaznak. A StatelessWidgetek hatékonyabbak és gyorsabbak, mint a StatefulWidgetok, mivel nem igényelnek újraépítést az állapotváltozások következtében. A statikus felhasználói felületi részek leírására általában StatelessWidgeteket használnak, melyek nem változnak gyakran, vagy csak akkor változnak, ha a felhasználói interakció miatt szükséges az újraépítés.

A StatefulWidget, egy olyan widget típus, amely lehetővé teszi a saját állapota kezelését és változtatását. Lehetővé teszik, hogy az alkalmazás felhasználói felülete dinamikus legyen. Mivel az állapot változásakor azok újraépülhetnek, átrendeződhetnek és frissülhetnek. Mivel az állapot változása miatt újraépíti magát, így több erőforrást igényelnek, mint a StatelessWidgetek. Általában az alkalmazás interaktív vagy animált részeit írják le, amelyek állapota folyamatosan változik.

Funkcionalitásuk alapján két kategóriákba sorolhatóak a Flutter widgetek, vannak az elrendezési widgetek(layout widgets) és a vezérlő widgetek(control widgets). A layout widgetek a Flutterben az elrendezés kialakításáért felelősek, meghatározzák, hogy a többi widget hol helyezkedjen el a felhasználói felületen. Ezek közé az alábbiak tartoznak:

- Text: a Text widget felhasználásával szöveges tartalmat jeleníthetünk meg a felületen. Lehetőséget biztosít a szöveg méretének, színének és stílusának beállítására, valamint a szövegtartalom kezelését, mint például a sortörések vagy az igazítás. A Text widget könnyen testreszabható és széleskörben felhasználható az alkalmazásokban.
- Row: egy olyan widget, amely lehetővé teszi a gyerekek vízszintes elrendezését. A gyerekek tetszőleges Flutter widgetek lehetnek, mint például szövegek, képek. Sok hasznos tulajdonsággal rendelkezik, az egyik legfontosabb az elrendezési irány, amely lehet vízszintes vagy függőleges. Lehetőséget biztosít az összetettebb elrendezések megvalósítását is.
- Column: a gyerekek függőleges elrendezését teszi lehetővé.
- Stack: a Stack widget, egy olyan elrendezési widget, amely lehetővé teszi, hogy több widgetet egymásra helyezhessünk, és szabadon elhelyezhessük őket a koordináta-rendszerben. Az alkalmazásokkor a widgetek egymásra kerülnek, így előtérbe hozhatjuk, átlátszóvá tehetjük vagy elrejthetjük őket. Gyakran alkalmazzák az animációk létrehozására, valamint komplexebb felhasználói felületek kialakítására.
- Container: lehetővé teszi a gyerekek méretének és pozíciójának beállítását, valamint a szegély, háttérszín és árnyékolás testreszabását is. A Container widgetet gyakran használják arra, hogy elrendezési konténereket hozzanak létre más widgeteknek.

- **SizedBox:** egy alapvető widget, amely egy üres téglalapot jelenít meg a felhasználói felületen. A widget segítségével pontosan megadható a widget magasságát és szélességét, valamint a widget és annak gyerekei közötti térköz.

Az elrendezési widgetekkel ellentétben a vezérlő widgetek a felhasználói interakciókat kezelik, mint például az adatok bevitelét, az érintésre történő kattintást. A vezérlő widgetek interaktivitást és dinamizmust biztosítanak a felhasználói felület számára. Flutterben a vezérlő widgetek közé a következők tartoznak:

- **Button:** gombok megjelenítését teszi lehetővé, amelyet a felhasználó érintésre kattinthat.
- **Radio button:** lehetővé teszi a felhasználó számára, hogy kiválasszon egy elemet egy listából.
- **Slider:** segítségével egy csúszka jelenik meg a felhasználó számára, amellyel egy adott tartományon belül állíthat be értékeket.
- **Switch:** lehetővé teszi a felhasználó számára, hogy be- vagy kikapcsolja egy adott beállítást.
- **Checkbox:** engedélyezi a felhasználónak, hogy kiválassza vagy törölje a jelölést az adott opció mellett.

A Dart egy nyílt forráskódú, a Flutter alkalmazások kódolásához használt modern, objektumorientált programozási nyelv. Ideális modern, komplex, hatékony és jól karbantartható alkalmazások fejlesztéséhez. Az alkalmazások könnyen testre szabhatóak, mivel a Flutter keretrendszer sok kész widgetet és eszközt kínál a grafikus felhasználói felület készítéséhez. A Flutter alkalmazások fejlesztése egyre népszerűbbé válik, mivel az egyik legmodernebb és leginkább támogatott keretrendszer a mobil alkalmazások fejlesztéséhez. A Dart nyelv egy modern és rugalmas programozási nyelv, amely kifejező és könnyen tanulható. A nyelv számos előnnyel rendelkezik, beleértve a gyors futtatási időt és a hatékony memóriakezelést, ami javítja a programok teljesítményét. A nyelv szintaxisa hasonló más C-szerű nyelvekhez, mint például a Java vagy a C++, de magasabb szintű funkciókkal rendelkezik, amelyek segítségével az alkalmazások fejlesztése gyorsabb és hatékonyabb lehet. A nyelv nagyon hasznos a web- és mobilalkalmazások, valamint a szerveroldali fejlesztés során, és széles körben használják az iparban.

Különbség a Dart és a Flutter között:

- Dart egy objektumorientált programozási nyelv. A Flutter miatt vált népszerűvé. Nem csak a Flutter keretrendszerrel használható.

- Flutter egy UI keretrendszer, amely főként a felhasználói felületekre fókuszál, de támogatja a üzleti logikát is. Különböző platformra készíthető vele alkalmazás. Az alkalmazások gyorsítása érdekében kész eszközöket tartalmaz.

2. UX és UI design

A mobilalkalmazások fejlesztésénél fontos az alkalmazás felhasználói felületének megtervezése.

A UX design a tervezési elvek és folyamatok összessége, amelynek célja a felhasználói elégedettség és élmény növelése a termék vagy szolgáltatás használata során. Egy optimális UX design figyelembe veszi a felhasználók igényeit, szokásait, nagy figyelmet fordít a felhasználókra, és megtervezi a szükséges felhasználói interakciókat és funkciókat. Az alkalmazás sikerének egyik legfontosabb tényezője a megfelelő UX design, amelynek közvetlen hatása van a felhasználói elégedettségre és az alkalmazás használhatóságára. A UX design segíti a felhasználókat a szükséges információk és funkciók megtalálásában, az alkalmazás könnyű navigálásában és céljaik hatékonyabb teljesítésében. A tervezés során a felhasználói tesztek és visszajelzések figyelembevétele elengedhetetlen a felhasználói igények kielégítéséhez és a legjobb felhasználói élmény biztosításához.

A UI design, vagyis a Felhasználói felület tervezése, az alkalmazások vagy termékek felhasználói felületének tervezését jelenti. Célja, hogy vonzó, könnyen használható és érthető felületet hozzon létre a felhasználók számára. Fontos szempont az optimális felhasználói élmény biztosítása, amely magában foglalja a felhasználói interakciók, navigáció, animációk és más elemek tervezését. A UI design eszközei közé tartoznak a színek, betűtípusok, ikonok, gombok és más vizuális elemek, amelyek segítenek a felhasználóknak megérteni az alkalmazás vagy termék funkcióit. Célja az egységes és összhangban lévő felhasználói felület kialakítása az alkalmazás minden részére. A UI design biztosítja a könnyű használhatóságot és a kiváló felhasználói élményt.

2.1. Történeti áttekintő

2.2. Főbb, alkalmazott tervezési szabályok

Az UX (felhasználói élmény) és UI (felhasználói felület) tervezés területén számos szabály és elv, amelyek hozzájárulnak az alkalmazás hatékony és felhasználóbarát felületének kialakításához. A következők tartoznak a UX design főbb tervezési szabályok közé:

Felhasználóközpontú tervezés: A tervezés felhasználóközpontú legyen, azaz a felhasználók igényeire és szokásaira koncentráljunk.

- Konzisztencia: Az alkalmazás minden eleme hasonló stílust, elrendezést és funkciókat kell használjon.
- Felhasználói tesztek: A felhasználói tesztek és visszajelzések figyelembevétele elengedhetetlen az alkalmazás tervezése során, hogy a felhasználók igényeire és visszajelzéseire reagáljunk és a lehető legjobb felhasználói élményt nyújtsuk.
- Információ hierarchia: Az információ hierarchiáját úgy kell megtervezni, hogy az információk könnyen érthetőek legyenek és fontossági sorrendben jelenjenek meg.
- Használhatóság: Az alkalmazás vagy termék használhatósága legyen a fókuszban, az interakciók logikusak és előreláthatóak legyenek.

A UI design szabályai közzé pedig az alábbiak tartoznak:

- Betűtípus: Fontos a megfelelő betűtípus kiválasztása, hogy az információk és funkciók szövege könnyen olvasható és érthető legyen.
- Egyszerűség: Az egyszerűség azért fontos, hogy az alkalmazás ne legyen túlszűfolt és a felhasználók könnyen megtalálják az információkat és funkciókat.
- Vizuális hierarchia: Az információk vizuális hierarchiájának kialakítása segít a felhasználóknak az információk megértésében és könnyebb észrevételében.
- Színhasználat: A megfelelő színhasználat segítségével a felhasználók könnyebben észreveszik a fontos információkat és a funkciókat.

3. Moodboard és logo

3.1.Moodboard

A moodboard egy vizuális összeállítás, amelynek célja, hogy különböző vizuális elemeket, mint például színeket, mintákat, textúrákat vagy ötleteket mutasson be, amelyek inspirációként szolgálnak egy adott téma, projekt vagy hangulat kifejezéséhez. A moodboard azért hasznos, mert segít összeállítani az elképzeléseket egy konkrét látványvilág vagy érzés megteremtése érdekében, és lehetővé teszi, hogy a különböző vizuális elemek egymással harmonizáljanak a végeredményben.

A moodboardot gyakran alkalmazzák a UX designer-ek, hogy vizuálisan megjelenítsék a számukra fontos stílust, koncepciót vagy hangulatot. Ennek az előnye, hogy lehetővé teszi a designer-ek számára, hogy ötleteiket szervezett és áttekinthető módon mutassák be, és segítségével kreatív munkát hozzanak létre.

A moodboard elemei:

- Képek: A moodboardnak jelentős részét képezik. Ezek lehetnek, fényképek, rajzok, minták vagy bármilyen más vizuális elem.
- Színek: A színek nagyon fontosak a moodboard készítésekor. Ha jól választjuk ki őket, akkor az segít abban, hogy a látványvilág egységes és hangulatos legyen.
- Betűtípusok: Az üzenet könnyű olvashatósága és érthetősége érdekében fontos a megfelelő betűtípusok kiválasztása és elrendezése.
- Szövegek: A szövegek alkalmazásával az alkotók részletesebben és pontosabban fejezhetik ki céljaikat és az általuk közvetíteni kívánt érzéseiket.
- Hangulatjelek: Az alkotók a hangulatjelek használatával érzelmileg hatékonyabbá tehetik a végeredményüket, és könnyebben átadhatják azokat az érzelmeket, amelyeket a munkájukkal kifejezni szeretnének.

A moodboard elkészítéséhez használt színválasztás, betűtípusok elrendezése, szövegek és hangulatjelek kombinálása mind hozzájárulnak egy összehangolt és inspiráló látványvilág megteremtéséhez.

Számos online alkalmazás létezik, amelyek lehetőséget biztosítanak moodboard készítésére, mint például a Canva, Milanote, Pinterest, Niice és sok más. Az alkalmazásomban a Canva grafikai tervezőt és szerkeztő eszközt alkalmaztam a moodboard elkészítésére. A moodboard elkészítésekor számomra fontos volt, hogy olyan színek és ábrák szerepeljenek rajta, amelyek a felhasználók számára egy letisztult és modern felhasználói felületet biztosítanak majd az alkalmazás használata során. (3.1. ábra)



3.1 ábra – Az alkalmazás moodboard-a

Az alkalmazásom megvalósítása során az ábrán látható színeket választottam, amelyeket hexadecimális kód formájában adtam meg. Az ábrán jól láthatóan megjelennek a kiválasztott színek, amelyek mindegyike fontos szerepet játszik az alkalmazás felépítésében és használhatóságában. A fehér színt alkalmaztam az alkalmazás háttérszínének, ami segít a felhasználónak a szemek fáradásának csökkentésében. A fekete szín pedig a betűk színének beállítására szolgált, amelyek jól láthatóak és könnyen olvashatóak maradnak. A sötétebb kék szín szerepe a gombok és az események nevének színeként jelenik meg, amely segít a könnyű navigációban és a kiválasztásban. A világosabb kék színt pedig az appbar és az esemény boxoknál használtam fel, ami segíti a felhasználói felület letisztultságát és modernitását.

3.2. Logo

Az alkalmazásban való logó használata fontos, mivel az az egyik első dolog, amit a felhasználók látnak, amikor letöltik az alkalmazást. A logó az alkalmazás arculatának kiemelkedő eleme, amely segít a felhasználóknak azonosítani és megkülönböztetni az alkalmazást más applikációktól. Azonkívül a logó használata hozzájárul az alkalmazás észlelhetőségéhez és azonosíthatóságához, amikor a felhasználók böngészik az alkalmazásboltban vagy az alkalmazást használják.



3.2 ábra – Az alkalmazás logoja

Az alkalmazás logójának elkészítésekor számomra az volt a fontos szempont, hogy olyan dizájnt hozzak létre, amely megfelel az alkalmazás témájának és a célközönségének. Az alkalmazás logója az első dolog, amit a felhasználók látnak, amikor letöltik az alkalmazást, így az igényes és összehangolt kialakítás kiemelkedően fontos. A logó tervezése során figyelembe vettem az alkalmazás célját, funkcióit, valamint az általa nyújtott élményt, és az ehhez passzoló színeket, formákat és betűtípusokat használtam. A cél az volt, hogy a logóval megragadjam az alkalmazás témáját, és vonzóvá tegyem azt a felhasználók számára. (3.2 ábra)

4. Képernyőtervek

5. Fejlesztési folyamat

5.1.Trello

5.2. Github

6. Alkalmazás szerkezeti felépítése

6.1.MVVM

6.2.Adatbázis

6.3.Üzleti logika

6.4.Kész alkalmazás

Irodalomjegyzék

Nyilatkozat

Alulírott szakos hallgató, kijelentem, hogy a dolgozatomat a Szegedi Tudományegyetem, Informatikai Intézet Tanszékén készítettem, diploma megszerzése érdekében.

Kijelentem, hogy a dolgozatot más szakon korábban nem védtem meg, saját munkám eredménye, és csak a hivatkozott forrásokat (szakirodalom, eszközök, stb.) használtam fel.

Tudomásul veszem, hogy szakdolgozatomat / diplomamunkámat a Szegedi Tudományegyetem Diplomamunka Repozitóriumban tárolja.

Dátum

Aláírás

Köszönetnyilvánítás