**Peaceful Paradise**

*Kálmán Dávid, Roncz Gábor, Vass Kornél*

1. Tartalomjegyzék

[2 Ötletelés 3](#_Toc99952454)

[2.1 A projekt témájának kigondolása 3](#_Toc99952455)

[2.2 A projekt nevének története 3](#_Toc99952456)

[2.3 Tökéletes arculat és logók 3](#_Toc99952457)

[2.3.1 Teljes logó 3](#_Toc99952458)

[2.3.2 Kis logó 3](#_Toc99952459)

[2.3.3 Szlogen 4](#_Toc99952460)

[2.4 Megvalósítás módjának kigondolása 4](#_Toc99952461)

[2.4.1 Programozási nyelvek – de melyik? 4](#_Toc99952462)

[2.4.2 Feladatkörök – ki és mit? 4](#_Toc99952463)

[3 Az alapok lefektetése 5](#_Toc99952464)

[3.1 Az adatbázis 5](#_Toc99952465)

[3.1.1 Reservation - Foglalás 5](#_Toc99952466)

[3.1.2 Room - Szoba 6](#_Toc99952467)

[3.1.3 Customer - Vendég 6](#_Toc99952468)

[3.1.4 Codes - Kódok 6](#_Toc99952469)

[3.1.5 Servicetype - Ellátás 6](#_Toc99952470)

[3.1.6 Consumption - Fogyasztások 7](#_Toc99952471)

[3.1.7 Storage - Raktár 7](#_Toc99952472)

[3.1.8 Reports - Hibajelentések 7](#_Toc99952473)

[3.1.9 UserLog/ReservationLog - Felhasználói és Foglalás naplózás 7](#_Toc99952474)

[3.1.10 Ratings - Értékelések 7](#_Toc99952475)

[4 A weboldal 8](#_Toc99952479)

[5 Recepció alkalmazás 8](#_Toc99952480)

# Ötletelés

## 2.1 A projekt témájának kigondolása

A projekt kezdetekor a legnagyobb és legnehezebb feladatot a téma megtalálása jelentett. Mit is akartunk csinálni szakdolgozatként? Egy olyan témakört szerettünk volna találni, amelyik minden csapattagnak megfelel, amivel mindenki tud azonosulni. Az eredeti elképzelésünk szerint egy városnak az oldalát csináltuk volna meg, ahol szinte minden elérhető lett volna, legyen szó hotelről, reptérről vagy kaszinóról. Természetesen mindegyikhez saját oldal tartozott volna. Ám erre hamar ráeszméltünk, hogy nem tudnánk teljesíteni a küldetést, enyhén túlvállaltuk volna magunkat. Tanárainkkal konzultálva arra a következtetésre jutottunk, hogy bőven elég lesz egy hotel weboldala komplex foglalási rendszerrel és hozzá egy WPF-es recepciós alkalmazás.

## 2.2 A projekt nevének története

Miután kitaláltuk, hogy „saját” hotelt fogunk csinálni, a következő egyből felmerülő probléma a név volt. Hogy is kellene elneveznünk egy képzeletbeli hotelt? Vagy egyáltalán mi alapján? Egy hosszú délutáni diskurzus után találtuk ki: ***Peaceful Paradise***. Mivel luxushotel, így a névnek passzolnia kellett az arculathoz. Egyszerre lett elegáns és magával ragadó. Egyszerre sugároz nyugalmat és kényelmet. Tökéletes.

## 2.3 Tökéletes arculat és logók

A fantasztikus nevet követően természetesen szükségünk volt fantasztikus logókra, amelyek az arculat alapját határozzák meg. Ezek megvalósítására különböző logó készítő oldalakat használtunk. Körül néztünk több helyen és így találtuk meg a végleges logókat körülbelül szeptember végére (persze ahogy teltek a hónapok egy-kettő vágáson keresztül mentek).

### 2.3.1 Teljes logó

Ez lett az arculat fő logója. Megtalálható rajta az ikonunk, *az arany madár*, amely a luxust és a megbízhatóságot szimbolizálja. Továbbá a hotel neve ugyan olyan stílussal feltüntetve. A fekete háttér is jó választás volt, hiszen nagyon jó kontrasztot ad az átmenetes arany színnel. A logót rendes nézetben az oldalon, a navigációs sávon illetőleg emailekben használjuk.

### 2.3.2 Kis logó

A fő logó mellé csináltunk egy kisebb, univerzálisabb logót, amelyet a weboldal telefonos nézetében használunk a navigációs sávon megjelenítve a rendes logó helyett (hiszen az piciben nem férne ki és nem is nézne ki jól), továbbá a projekt ikonjaként is szolgál, mert csak *az arany madár* található meg rajta.

### 2.3.3 Szlogen

Az arculathoz hozzátartozik még a projektünk szlogenje is: „*Who’s prepared to pay the price for a trip to Paradise*?”. Azaz magyarra fordítva: „*Ki áll készen megfizetni az árát egy útnak a Paradicsomba*?”. Sajnos a hosszúsága miatt nem fért rá a logóra, így csak a főoldalon használjuk, illetve egy kettő emailben.

## 2.4 Megvalósítás módjának kigondolása

Még mielőtt nekifogtunk volna a projektnek, ki kellett találnunk milyen technológiákat, programnyelveket szeretnénk használni, illetve fel kellett osztanunk a feladatokat, hogy ki min fog dolgozni.

### 2.4.1 Programozási nyelvek – de melyik?

A WPF alkalmazásnál adva volt a ***C#*** nyelv használata, viszont a weboldalnál aránylag sokat gondolkodtunk, hogy a számtalannyi tanult programozási nyelv közül melyik(ek)et fogjuk használni. Végül a választásunk a ***Full-Stack PHP***-ra esett. Hogy miért? Mert akkor ez tűnt a legszimpatikusabbnak, mindenkinek a csapatból. Nincs külön Frontend és Backend, egyben fejlesztünk mindent (ami, mint később kiderült, sok mindenben megkönnyítette a dolgunkat). Természetesen kiegészítésként gondoltunk a natív JavaScriptre is.

### 2.4.2 Feladatkörök – ki és mit?

Előzetes elképzeléseink szerint a feladatkörök felosztása az alábbiak szerint alakult volna:

* Weboldal: Kálmán Dávid, Roncz Gábor
* WPF alkalmazás: Vass Kornél

Ezeket a fő irányvonalakat meg is tartottuk az egész projekt alatt. Értelemszerűen, ha kényes témáról volt szó arról együtt döntöttünk, illetve mindig kikértük a másik véleményét és segítettünk egymásnak.



# Az alapok lefektetése

## Az adatbázis

Ahhoz hogy a megfelelő adatokat egyszerűen elérjük szükségünk volt egy nagyon kifinomult, precíz adatbázis, amely mindent tartalmaz, amire szükségünk van. Végül, többszöri megfontolás és módosítás után ez lett a végleges változata:

### Reservation - Foglalás

A foglalás tábla az adatbázisunk gyökeres gerince. Mindenféle nézőpontból a középpontba helyezhető és rangsorban is az első helyen áll. A munka legeslegelején még csak egy csonka szerkezettel rendelkezett. A folyamat során többször is átesett radikális változtatásokon mire elnyerte végleges formáját. A tartalma főbb részét természetesen a foglalás rögzítéséhez szükséges adatmezők teszik ki. Mindemellett pár idegen kulcson felül valamint a recepció alkalmazáshoz létfontosságú „IsCheckedIn” nevezetű cella, amiben értelem szerűen azon információt őrizzük, hogy az adott vendégünk megérkezett-e már avagy sem. Ebben a táblában fellelhető az ár mező is, amelyet úgy kapunk, hogy a szobaár és az ellátás árának az összegét megszorozzuk az éjszakák számával. Ha esetleg a vendég már elérte azt a szintet, hogy kedvezményt kap akkor csak annak levonása után kerül az adat a táblába.

### Room - Szoba

A szoba táblánk mondhatni a második legfontosabbnak tekinthető felhasználási szempontokból. A feladat folyamán hosszas beszélgetés után arra a döntésre jutottunk, hogy egy típusú szobából tíz darabot tartunk fent, melyekből választható kettő illetve négy férőhelyes. Mindemellett további szabványos információkat tartunk számon, mint például az adott szoba ára, leírása, megnevezése és a weboldalon megjelenő fényképek hivatkozásai. Ugyanakkor a legfőbb sajátosságok is rendelkezésre állnak az adatbázisunkban, amikre úgymond egy kis ráadásként tekintünk a különböző információk mellett. Ehhez a táblához kapcsolódik a legbonyolultabb lekérdezésünk, amely abból áll, hogy lekérjük azokat a szobákat, amelyek szabadok az adott időpontban és megfelelnek a vendégek számának is. Számos próbálkozás és több órás ötletelés után pedig végre létrehoztuk a megfelelő függvényt.

### Customer - Vendég

A vendég táblánk eredetileg sokkal kisebbnek és jelentéktelenebbnek indult, azonban később a regisztráció és bejelentkezés bonyolította ilyen mértékűre. Csakhogy nem csak ez a két funkció okán kellett további bővítéseket alkalmazni. Kitaláltunk egy egyszerű prémiumokkal járó rendszert, mivel szerettük volna valahogy jutalmazni a visszatérő vendégeket. Ennek a funkciónak a megvalósítására hoztuk létre a „LEVEL” nevezetű mezőt, ami három különböző szintet tartalmazhat (gold, platinum, diamond) a foglalási számoktól függően, amelyet a „ReservationNumber” cellában tárolunk. Itt fellelhető meg a jelszó mező is, amely nyilvánvalóan a bejelentkezéshez szükséges, ez a mező a felhasználó által választott jelszó md5-ös titkosítása. Vannak olyan esetek is mikor a felhasználónév és a jelszó üresek, ez akkor történik, ha a látogató nem regisztrál, hanem „vendég” módban foglal. Először az aktiválásokhoz használt tokeneket és kódokat ebben a táblában tároltuk el azonban később ráeszméltünk arra, hogy észszerűbb lenne ezeket a saját táblájukban tárolni.

### 3.1.4 Codes - Kódok

A kódok táblában a regisztráció, elfelejtett jelszó, és az email változtatás validálásához szükséges kódokat tároljuk. Megtalálható a kód lejárata, amely típusonként változik, és fellelhető itt egy idegen kulcs, amely egy-egy sort egy-egy vendéghez rendel. Egy látogatónak egyféle kódból csak egy darab lehet, ezeket mindig vizsgáljuk, és ha a rendelkezik már egy kóddal az adott típusból, akkor azt szükség szerint lecseréljük. Maga a kód egy random generált számokból és betűkből álló karaktersorozat.

### Servicetype - Ellátás

Ez a tábla az egyik legegyszerűbb és legegyértelműbb tábla az adatbázisunkban. Mindössze három adatot tartalmaz, amelyekből egy csak egy azonosító, amelyet felhasználunk a foglalások táblában is. A következő mezője az ár, amely magától értetődően az adott ellátási típus árát tárolja. Az utolsó cella pedig a típus nevét foglalja magába. Ebben a táblában összesen háromfajta ellátás közül lehet választani (Csak reggeli, félpanzió és teljes panzió).

### Consumption - Fogyasztások

Ebben a táblában a vendégek által fogyasztott ételeket italokat tároljuk. Tartalmaz egy „CustomerID” idegen kulcsot is, amely jelzi, hogy melyik foglaláshoz tartozik az adott rekord. Továbbá megtalálható benne az ára is a fogyasztott tételeknek, amelyet a vendég távozásakor fizetnie kell a recepciónál. Az adatok és a fogyasztási lehetőségek eléréséhez a „storage” táblát használjuk.

### Storage - Raktár

Az eredeti adatbázis tervben ez a tábla még nem szerepelt, de idő közben a weboldalnak és a recepció alkalmazásnak is szüksége volt arra, hogy létrehozzuk ezt az adattárolót. Tartalmazza az egyes tételek nevét és árát is, és megjelenik egy típus mező is. Ezt a weboldalon az Étterem fülön használjuk, e cella segítségével tudjuk külön megjeleníteni az ételkategóriákat, ezen rekordokhoz tartozik egy kép is, amelyre az „ImageURL” mező hivatkozik. A recepció alkalmazásban pedig szűrni lehet a tábla tartalmában ennek a mezőnek a segítségével.

### Reports - Hibajelentések

Ez a tábla a weboldalon található „Contact us” menüpontban küldött üzeneteket tartalmazza. A felhasználó email címét, nevét, a felhasználó által nyújtott hibaleírást, és a küldés időpontját tartalmazza. Ha egy vendég küld egy hibajelentést arról emailben értesülünk, és az ügyfél is kap egy visszaigazoló emailt, miszerint sikeresen elküldte az üzenetét.

### UserLog/ReservationLog - Felhasználói és Foglalás naplózás

Ezek a táblák is a munkánk második felében kerültek az adatbázisba. A weboldal használata során egy felhasználó többször is belekerül ezekbe a naplózásokba, például: mikor bejelentkezik, szobát foglal, vagy ha hibajelentést küld. Ezen táblákba egy beépített eljárással szúrjuk be a rekordokat. Ez a függvény négy paramétert kap, először is az aktuális felhasználó nevét, majd az, hogy milyen műveletet hajtott végre a vendég, ezek után a cselekvés időpontját, majd végül, de nem utolsó sorban a legfontosabb tényező, hogy melyik táblába kell beszúrni a létrehozott rekordot. A recepció alkalmazásban az adminisztrátor megtekintheti ezeknek a tábláknak a tartalmát. Eredetileg ezek a táblák kapcsolódtak volna a megfelelő fő táblákhoz viszont később ráeszméltünk, hogy mikor egy felhasználót vagy foglalást törlünk, akkor ki kellene törölni az ahhoz tartozó naplózási tételeket is. Viszont ez pedig pont a naplózás szöges ellentét lenne akkor, mivel a lényege hogy mindent meg lehessen találni és le lehessen követni.

### 3.1.10 Ratings - Értékelések

Weboldalunk „Rate us” fülében küldött értékeléseket tartalmazza ez a tábla. Szintén tartalmazza a felhasználó nevét és természetesen email címét is. A következő mező a vendég által írt üzenet. Ezen az oldalon megtalálható egy csúszka melyen kiválasztható, hogy a látogató egy egytől ötös skálán mennyire élvezte nálunk töltött idejét. Ez a tábla nem tartalmazza a vendégek tábla idegen kulcsát, mert bárki értékelheti a szállodánkat, és nem csak olyan emberek, akik már szerepelnek a nyilvántartásunkban.



# A weboldal

# Recepció alkalmazás

## Models mappa

A fájlszerkezetben található „Models” mappa tartalma az adatbázisban található táblákkal egyezik meg. Viszont itt egyel több fájl található, mint adatbázis tábla. Ennek az az oka, hogy a [3.1.9](#_UserLog/ReservationLog_-_Felhasznál) részben említett függvény meghívása egy külön fájlban helyezkedik el, ez csak azért van, így hogy jobban átlátható legyen a mappaszerkezet és ne a két fájl valamelyikében kelljen keresgetni, hogy melyikbe, akarunk éppen beilleszteni. Mindegyik osztályban fellelhető az adott adatokhoz kapcsolódó lekérdezések. Ezek lehetnek törlés, beillesztés, vagy frissítés is.

## Views mappa

Ebben a mappában a nevének megfelelően a „nézetek” vannak. Ezeknek az ablakoknak és oldalaknak a segítségével tud navigálni az alkalmazáson belül a recepciós.

## Az alkalmazás működése

### Az állandó elemek

#### A navigációs sáv

#### Az alkalmazás egy egyedi fejléccel rendelkezik. Erre azért volt szükség, mert az alapértelmezett nem illik a program dizájnjához. Végül ezt a kinézetet választottuk a navigációs sávnak.

#### A menü

#### Ez az elem is többszöri változtatáson esett át, de a négy eredeti eleme még most is szerepel a látható menüpontok között. Ezek a következők: Vendég bejelentkezés, kijelentkezés, foglalások szerkesztése és fogyasztások hozzáadása. Ezek mellett további két elemmel bővült a munka során. Ezek közül az első a felhasználói és foglalási naplózás megtekintése. Az utolsó menüpontban pedig a weboldalon küldött hibajelentéseket és értékeléseket lehet megnézni. A menü alatt pedig az aktuális dátum és idő is fellelhető. Eredetileg szerepelt volna itt egy menüpont ahol a vendégek be és kijelentkezéskor fizetett összegeket lehetett volna szemügyre venni. De a munkánk során lecseréltük ezt a „Logs” menüpontra és a hely hiányában ezt el kellett távolítani a menüpontok közül.