

Budapesti Műszaki Szakképzési Centrum

Neumann János Informatikai Technikum

Szakképesítés neve: Szoftverfejlesztő

OKJ száma: 54 213 05

ZÁRÓDOLGOZAT

Spare Time Hotel

Rostagni Csaba Ignác

konzulens

Kálmán Bence

13. rsz

Budapest, 2021.

Tartalomjegyzék

1.	Bevezető	4
2.	Választott téma indoklása	4
3.	Fejlesztői dokumentáció	4
3.1.	Követelmények meghatározása.....	4
3.1.1.	Felhasználói felület	4
3.1.2.	Foglalás	4
3.1.3.	Munka- és jogkörök.....	5
3.2.	Fejlesztői környezet.....	6
3.3.	Dizájn	6
3.4.	Alapszerkezet	7
3.4.1.	Végleges osztályok és metódusok	8
3.5.	Navigáció	10
3.5.1.	Névterek	11
3.6.	Adatbázis	11
3.6.1.	A guests tábla	12
3.6.2.	rooms tábla	12
3.6.3.	A room_booking tábla	13
3.6.4.	Az orders tábla.....	13
3.6.5.	A menu tábla	13
3.6.6.	A recommendation és menu_recommendation tábla.....	14
3.6.7.	Az allergens és menu_allergens tábla	14
3.6.8.	Az error_reports tábla	14
3.6.9.	A diary tábla	15
3.6.10.	A spending tábla.....	15
3.6.11.	Az employees tábla	15
3.6.12.	A jobs tábla	16

3.6.13.	A job_history tábla	16
3.6.14.	Az attendance_sheets tábla	16
3.7.	Adatbázis kapcsolat	17
3.8.	Foglalás	17
3.8.1.	Megvalósítás	18
3.9.	Beléptető rendszer	19
3.10.	Munkakörök és funkciók	19
3.10.1.	Rekordok kiírása	21
3.10.2.	Rekordok hozzáadása	21
3.10.3.	Rekordok törlése	21
3.10.4.	Rekordok szerkesztése	21
3.11.	Tesztelés	22
4.	Felhasználói dokumentáció	22
4.1.	Telepítés és használat	23
4.1.1.	Telepítés	23
4.1.2.	Foglalás	26
4.1.3.	Bejelentkezés	29
4.1.4.	Jogosultságkezelés	30
4.1.5.	Porta	30
4.1.6.	HR	31
4.1.7.	Karbantartó	33
4.1.8.	Séf és pincér	34
5.	Összegzés	36
6.	Felhasznált irodalmak	37

1. Bevezető

A 2021-es záródolgozatom témája egy fiktív szálloda, a **Spare Time Hotel** honlapja. A célom, egy olyan webalkalmazás megírása, ami képes megoldást nyújtani a szálloda releváns tevékenységeinek egyszerű és gyors egy platformon való kezelésére.

2. Választott téma indoklása

A választásom azért esett erre a projektre, mert hiánypótló lehet a piacon egy olyan alkalmazás, ami egy felületen kezeli a szálloda **back-office** kulcsfontosságú területeit. A webalkalmazás választásának magyarázata minden, ami előnyével jár egy asztali alkalmazással szemben, azaz elsősorban: többplatformúság, specifikáció függetlenség, valamint a telepítési és frissítési problémák elkerülése.

3. Fejlesztői dokumentáció

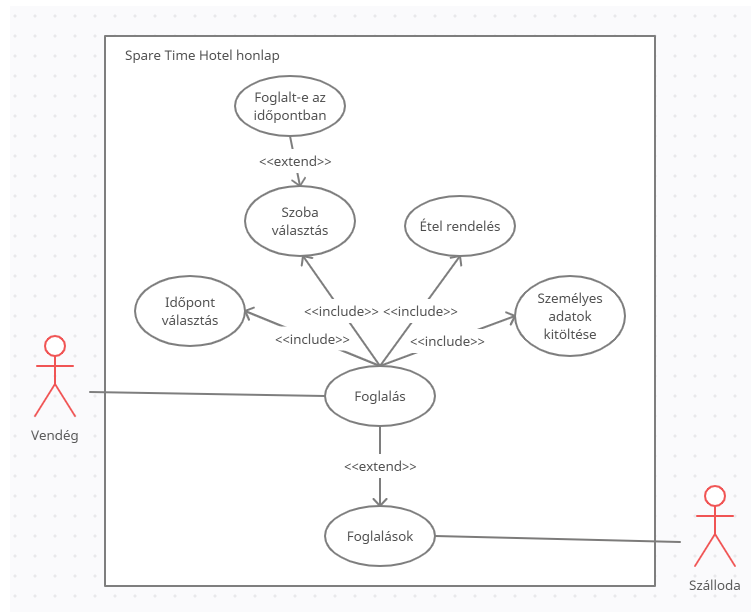
3.1. Követelmények meghatározása

3.1.1. Felhasználói felület

Amikor egy látogató rákattint az oldalra a kezdőlapra találja magát, ahol látja a fontosabb információkat, és egyből elkezdheti a foglalást. Szeretnénk, hogy tudjon informálódni a szálloda elérhetőségeiről, és a turizmusra is gondoljunk. Legyen egy *Látnivalók*, és egy *Elérhetőségek* menüpont is.

3.1.2. Foglalás

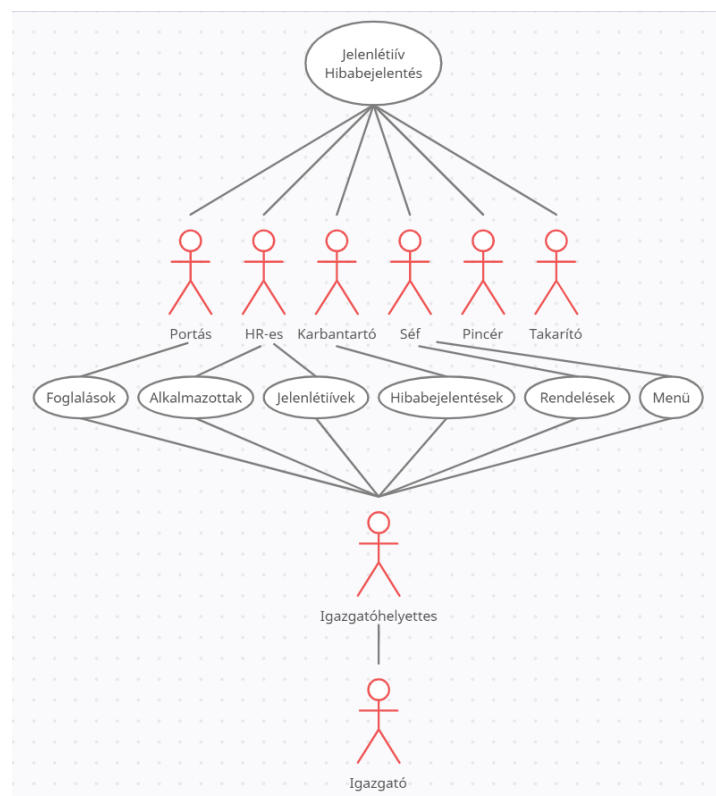
A vendég képes szobát foglalni, amihez értendő az időpontok, a szobák és az ételek kiválasztása, valamint a személyes adatok kitöltése. A szálloda látja a foglalásokat, és képes kezelni. A következő oldalon található *I. ábra* megmutatja, hogyan is kéne működnie.



1. ábra

3.1.3. Munka- és jogkörök

A honlapon szükségünk lesz egy beléptető rendszerre, amin áthaladva az alkalmazottak a munkakörüknek megfelelően látják, és tudják használni a felületet. A 2. ábrán egyértelműen látszik a kívánt eredmény. Munkakörök: portás, HR, karbantartó, séf, pincér, takarító, igazgatóhelyettes és igazgató.



2. ábra

A portás látja a foglalásokat, a HR-es az alkalmazottak információt képes kezelni, a séf a rendeléseket és a menüt, az igazgató és igazgatóhelyettes mindent lát, valamint mindenki kezelheti a saját jelenléti ívét, és képes hibát bejelenteni.

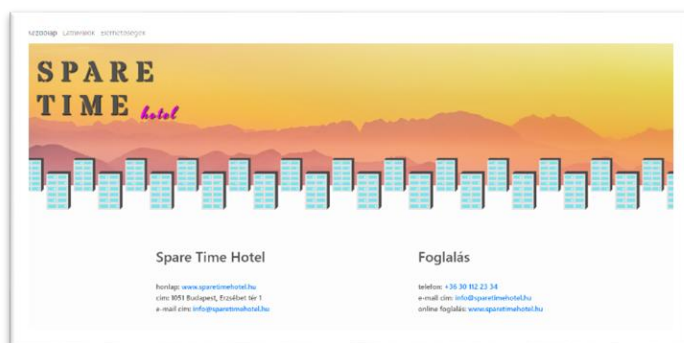
3.2. Fejlesztői környezet

Az általam használt alkalmazások listája a következő:

- PhpStorm - frontend és backend fejlesztés
- XAMPP v3.2.4 - **Apache** webservert és **MySQL** adatbázis szolgáltatás biztosítása
- dbForge Studio - adatbázis kezelés

3.3. Dizájn

A kinézet tervezésénél igyekeztem a **UX** és **UI** dizájn tapasztalataimat hasznosítani. A weboldal célközönsége elég tág, így a cél egy egyszerű, egyértelmű és könnyen kezelhető felület volt. Már a tervezés alatt gondoltam a **reszponzivitásra**, és több méretben elkészítettem kinézetet. A saját stíluslapom mellett a **bootstrap**-et részesítettem még előnyben.





3.4. Alapszerkezet

Számos módon megvalósítható a **Spare Time Hotel** honlapja. Én a fejlesztés során végig a **Model-View-Controller (MVC)** programtervezési mintát követtem, amely a webalkalmazást három fő logikai komponensre bontja: a modellre, a nézetre és a vezérlőre.

Model

Ez az alkalmazás dinamikus adatszerkeztése, amellyel a felhasználó javarészt dolgozik. A **model** biztosítja az adatbázisban szereplő adatokkal való kapcsolatot, amit osztályokon belüli objektumokkal tárolunk.

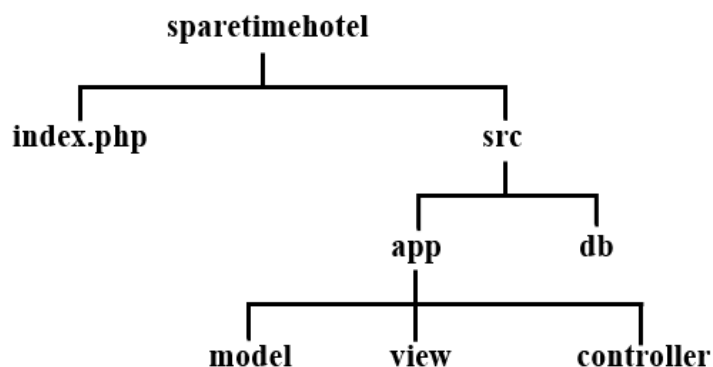
View

A **view** az alkalmazás összes felhasználói felületét tárolja. Az információ bármilyen ábrázolása lehetséges, lehet az egy *.html*, vagy *.php* kiterjesztésű fájl, vagy egy kép, pdf, stb..

Controller

A Vezérlő kezeli a felhasználói kéréseket. Interfészként működik a **model** és a **view** komponens között. A kérésnek megfelelően válaszol, ha szükséges a **model**-ben történő változásokat is kiválthat.

A gyökérmappánkban lévő *index.php* fájlban halad át minden. Itt vannak kezelve a **HTTP** kérések irányítása a **controller** felé. Ezen felül a fejlesztés során **whoops**-ot használtam, ami egy hibakezelő keretrendszer a **php** számára, ehhez telepítettem a **composer**-t, és magát a hibakezelőt a gyökérmappába. A 3. ábrán látható az alap mappaszerkezet.



3. ábra

3.4.1. Végleges osztályok és metódusok

A honlap végleges állapotában összesen 16 **model**, és 10 **controller** osztály lett. Az adatbázis alapján, minden egyes táblának van egy **model** megfelelője. Az alábbi táblázatban látható az összes osztály, és azoknak a metódusai.

Model	Controller
<ul style="list-style-type: none"> • Allergens <ul style="list-style-type: none"> ◦ findAll() ◦ findOneById() • AttendanceSheets <ul style="list-style-type: none"> ◦ findAll() ◦ findOneById() ◦ update() ◦ insert() ◦ getRowCount() • Diary <ul style="list-style-type: none"> ◦ findAll() ◦ findOneById() ◦ update() ◦ insert() • Employees <ul style="list-style-type: none"> ◦ getJob() ◦ findAll() ◦ findOneById() ◦ getCurrentUser() ◦ findOneByEmail() ◦ doLogin() ◦ load() ◦ insert() ◦ update() ◦ updateActive() ◦ getRowCount() • ErrorReports <ul style="list-style-type: none"> ◦ findAll() ◦ findOneById() ◦ update() ◦ insert() 	<ul style="list-style-type: none"> • AttendanceSheetsController <ul style="list-style-type: none"> ◦ actionInsertMonth() ◦ actionFetch() ◦ actionFetchSingle() ◦ actionUpdateStatus() ◦ actionUpdate() • EmployeesController <ul style="list-style-type: none"> ◦ actionLoginIndex() ◦ actionLogout() ◦ actionInsert() ◦ actionFetch() ◦ actionFetchSingle() ◦ actionActive() • ErrorReportsController <ul style="list-style-type: none"> ◦ actionInsert() ◦ actionInsertDiary() ◦ actionFetch() ◦ actionFetchSingle() ◦ actionFetchSingleDate() ◦ actionFetchSingleDiary() • GuestsController <ul style="list-style-type: none"> ◦ actionInsert() ◦ actionFindAll() • MainController <ul style="list-style-type: none"> ◦ render() • MenuController <ul style="list-style-type: none"> ◦ actionMenuInsert() ◦ actionFetch() ◦ actionFetchSingle() ◦ actionDelete() • OrdersController

<ul style="list-style-type: none"> ○ getRowCount() • Guests <ul style="list-style-type: none"> ○ findAll() ○ findAllFetch() ○ findOneById() ○ insert() ○ update() • JobHistory <ul style="list-style-type: none"> ○ findAll() ○ findOneById() ○ update() ○ insert() • Jobs <ul style="list-style-type: none"> ○ findAll() ○ findOneById() ○ currentUserCan() ○ getCurrentUserAccessRight() • Menu <ul style="list-style-type: none"> ○ findAll() ○ findOneById() ○ findOneByName() ○ insert() ○ update() ○ delete() ○ getRowCount() • MenuAllergens <ul style="list-style-type: none"> ○ findById() ○ insert() ○ update() ○ delete() ○ deleteAllById() • MenuRecommendation <ul style="list-style-type: none"> ○ findAll() ○ findOneById() ○ insert() ○ update() ○ delete() • Orders <ul style="list-style-type: none"> ○ findAll() ○ findOne() ○ load() ○ insert() ○ update() ○ delete() ○ getRowCount() • Recommendation <ul style="list-style-type: none"> ○ findAll() ○ findOnebyId() ○ load() ○ insert() ○ update() ○ delete() • RomBooking <ul style="list-style-type: none"> ○ findAll() 	<ul style="list-style-type: none"> ○ actionInsert() ○ actionFetch() ○ actionFetchSingle() ○ actionDelete() • ReservationGuestsController <ul style="list-style-type: none"> ○ actionDate() ○ actionRoom() ○ actionMenu() ○ actionPersonalData() ○ actionInsert() ○ actionFetch() ○ actionFetchSingle() ○ actionDelete() ○ actionDone() • RoomBookingController <ul style="list-style-type: none"> ○ actionInsert() ○ actionFetch() ○ actionFetchSingle() ○ actionCheck() • ViewController <ul style="list-style-type: none"> ○ actionIndex()
--	--

<ul style="list-style-type: none"> ○ findById() ○ currentRoom() ○ load() ○ insert() ○ update() ○ check() ○ delete() ○ getRowCount() • Rooms <ul style="list-style-type: none"> ○ findAll() ○ findAllStorey() ○ findById() ○ getRowCount() • Spending <ul style="list-style-type: none"> ○ findAll() ○ findById() ○ insert() ○ update() ○ delete() 	
--	--

3.5. Navigáció

Az oldalon való navigáláshoz a **controller** és **action** get kérést használok. A **controller** értéke általában a vezérlő fájlok egyikének a neve, míg az **action** az azon belüli egyik metódus. Ez a két paraméter a kérést követően az **url**-ben kerül átadásra, a például következő módon: *index.php?controller=view&action=home*

A beérkezett **get** paramétereket az *index.php* feldolgozza, és az osztályok, illetve metódus neveknek megfelelően átalakítja.

```
$controllerName = !empty($_GET['controller']) ?
    ucfirst($_GET['controller']).'Controller' : 'ViewController';
$actionName = !empty($_GET['action']) ?
    'action'.ucfirst($_GET['action']) : 'actionHome';
```

Az *index*ben van egy *content* nevű változó, aminek értéket adunk elágazásokban a controllerek által, majd a *mainTemplate.php*-ban egyszerűen kiírásra kerül.

```
if($controllerName == 'ViewController') {
    $controller = new \app\controller\ViewController();
    $content = !empty($_GET['action']) ?
        $controller->actionIndex($_GET['action']) :
        $controller->actionIndex('home');
}
include('src/app/view/template/mainTemplate.php');
```

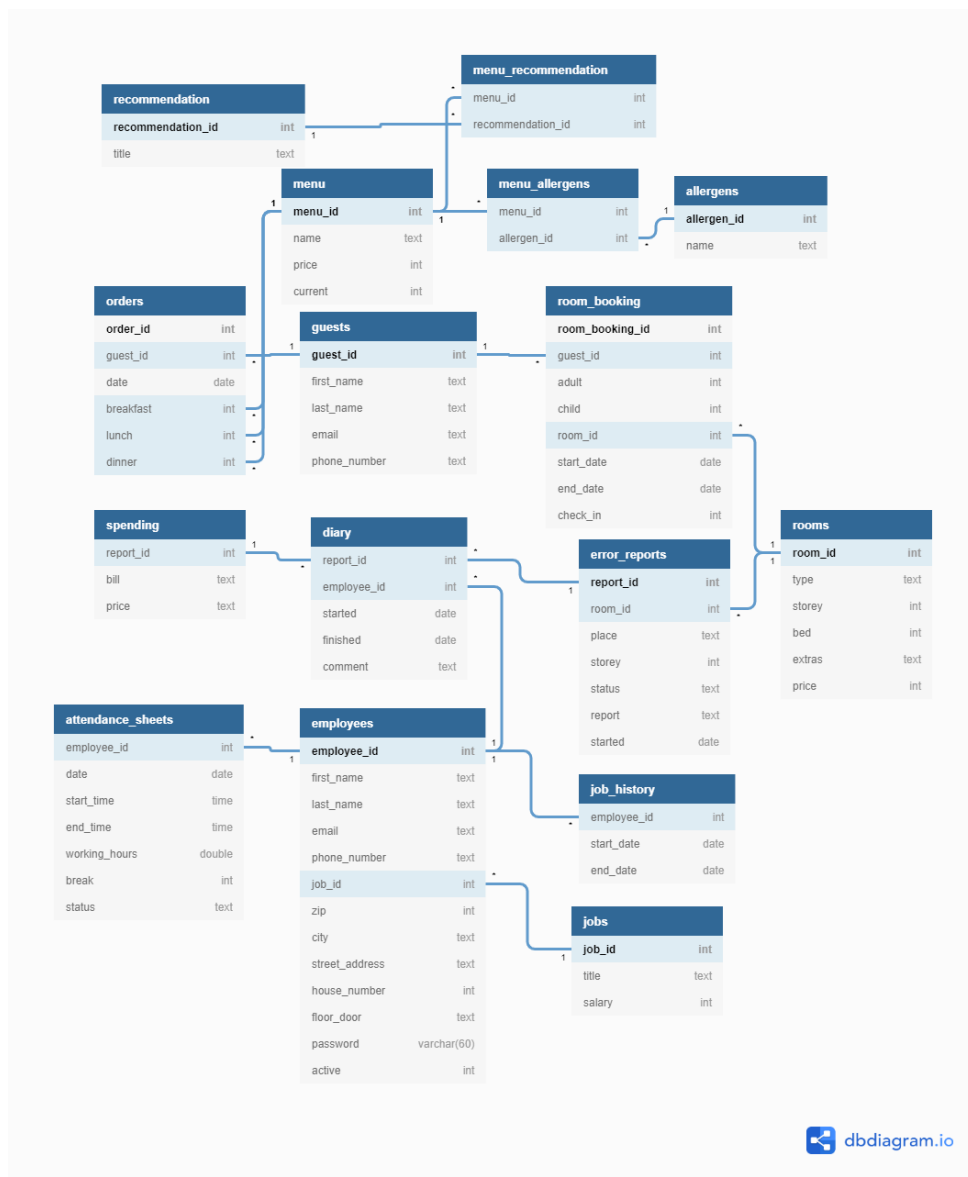
Minden controller öröklí a MainControllert, ami elsősorban az elérési utat biztosítja a navigáció segítése érdekében.

A menü navigáció a *ViewController*-en keresztül történik, ahol az **action** értéke mindig az elérni kívánt oldal, ami a **view** mappában található meg.

3.5.1. Névterek

Az fájlok közötti kapcsolattartást névterekkel oldottam meg, amihez a **whoops** mellé letöltött **autoload**-ot használok. A névterek használatával sokkal egyszerűbben kezelhető és átláthatóbb a fájlok közötti kapcsolatokat. Nem áll módunkban minden szükséges fájlt egyesével elérési útvonallal együtt beimportálni (főleg nem **MVC** struktúra esetén), elég csak minden fájlban egy-egy rövid sor névtér megadás, és importálás.

3.6. Adatbázis



4. ábra

Az adatok eltárolására 4. ábrán is látható **sparetimehotel** adatbázis felel, amit két nagy részre lehet bontani. Az egyik a foglalással, a másik pedig a hotel munkavállalóival kapcsolatos információkkal foglalkozik. Az adatbázis és minden táblája **UTF-8** kódolású. Ahol a normalizálás vagy a hibátlan működés megköveteli, ott a táblák első mezeje az elsődleges kulcs, ami az **auto increment** típusú `<aktuális táblanév>_id` nevet kapta.

A diagramon a világoskék háttérű mezők jelzik az **idegen kulcs**-okat, ezekből a táblák első sorában látható vastagított szöveg az **elsődleges kulcs**-ot mutatja. Az oszlopnevek mellett jobb oldalra igazítva láthatóak a típusok.

3.6.1. A guests tábla

guests	
guest_id	int
first_name	text
last_name	text
email	text
phone_number	text

5. ábra

Ebben a táblában vannak eltárolva a vendégek neve és elérhetőségei. A *guest_id* kulcsra hivatkozik a *room_booking* és az *orders* tábla.

3.6.2. rooms tábla

orders	
order_id	int
guest_id	int
date	date
breakfast	int
lunch	int
dinner	int

6. ábra

A helységek eltárolására szolgál, a szobák esetén az elsődleges kulcs értéke megegyezik a szobaszámmal. A *type* mező a helység típusát tárolja, ami lehet az étkező, iroda, szoba vagy a takarító szertár. A *storey* az emeletet, a *bed* az ágyak számát, a *price* az árat, míg az *extras* a nem minden szobában fellelhető extrákat tartalmazza (kanapé/jakuzzi). A *room_id*-ra hivatkozik az *error_report* és a *room_booking* tábla.

3.6.3. A room_booking tábla

room_booking	
room_booking_id	int
guest_id	int
adult	int
child	int
room_id	int
start_date	date
end_date	date
check_in	int

7. ábra

Ebbe a táblába kerül minden egyes foglalás információja. Egy sorban eltárolásra kerül a *guests* és a *rooms* táblára való hivatkozás, a felnőttek és gyermekek száma, a bejelentkezés és kijelentkezés dátuma valamint a becsekkolási állapot. A táblát a 7. ábra illusztrálja.

3.6.4. Az orders tábla

room_booking	
room_booking_id	int
guest_id	int
adult	int
child	int
room_id	int
start_date	date
end_date	date
check_in	int

8. ábra

A 8. ábrán látható rendeléseket tároló tábla szintén rendelkezik egy elsődleges kulccsal. A *guest_id*-val hivatkozunk a vendég-re, és eltároljuk a rendelés dátumát, illetve három főétkezésre bontva a megrendelt étel *id*-jét, amit a *menu* táblában került eltárolásra.

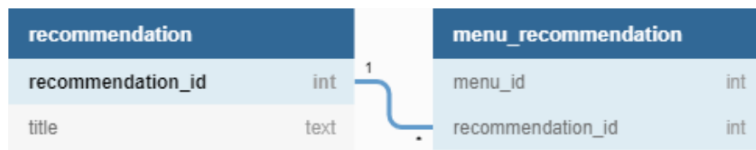
3.6.5. A menu tábla

menu	
menu_id	int
name	text
price	int
current	int

9. ábra

A *menu* táblában minden valaha étlapon szerepelt ételt tárolunk. A *menu_id* elsődleges kulccsal hivatkozik erre a táblára az ajánlás, és az allergének összekötő tábla, valamint az orders. Külön mezőkben tároljuk az étel nevét és árát, illetve az érvényességének a státuszát, ami az 1 értéknél az étlapon szereplőt, míg a 0-ásat a már/még nem szereplőt értjük.

3.6.6. A recommendation és menu_recommendation tábla



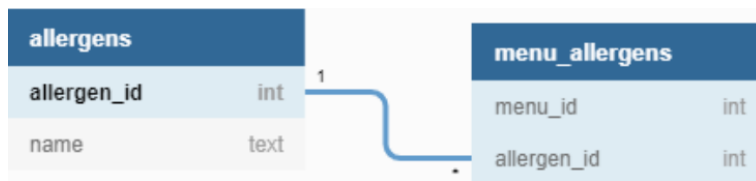
10. ábra

A *recommendation* tábla tartalmazza a létező ajánlások nevét és azonosítóját. A

menu_recommendation egy

összekötő tábla, ami a *menu*-ben szereplő adatokhoz rendeli az ajánlást. Jelen esetben az ajánlás lehet a séf, és/vagy a hónap kedvence.

3.6.7. Az allergens és menu_allergens tábla



Az *allergens* tábla eltárolja az összes jelölésköteles allergént.

Az összekötés ugyanolyan módon működik, mint a

menu_recommendation esetében. A *menu* adatait összeköti az *allergens*-ben szereplő azonosítókkal.

3.6.8. Az error_reports tábla

error_reports	
report_id	int
room_id	int
place	text
storey	int
status	text
report	text
started	date

Ez a tábla a munkavállalók hibajelentéseit tárolja el. Először egy elsődleges kulccsal rendelkezik, amire a *diary* tábla fog hivatkozni. A *room_id* idegen kulcs a *rooms* táblára mutat, amiből a pontos helységet tudhatja meg a karbantartó. Ezen kívül eltárolásra kerül itt is a helység neve, az emelet száma, a hiba státusza (meghibásodott/tönkrement/egyéb), a hozzáfűrendő üzenet, illetve a bejelentés dátuma.

11. ábra

3.6.9. A diary tábla

diary	
report_id	int
employee_id	int
started	date
finished	date
comment	text

12. ábra

A *diary* tábla a már megoldott hibabejelentések naplóinak az eltárolására hivatott. Az első két mezővel hivatkozik az adott hibabejelentésre, és a munkát elvégző karbantartó azonosítójára. A naplóban szereplő *started* a bejelentés dátumával egyenértékű, míg a *finished* a napló feltöltésének az ideje. Utolsó sorban a *comment* mező a karbantartók megjegyzéseit, észrevételeit tárolja.

3.6.10. A spending tábla

spending	
report_id	int
bill	text
price	text

13. ábra

Ebben a táblában szerepel a *diary* értékeihez csatolt számlák. A *report_id* a *diary* idegen kulcsára hivatkozik, mivel a számlák a naplózáskor hozzuk létre és kezeljük. A *bill* mező a feltöltött kép nevét, míg a *price* a képen található árat tartalmazza külön feltüntetve.

3.6.11. Az employees tábla

employees	
employee_id	int
first_name	text
last_name	text
email	text
phone_number	text
job_id	int
zip	int
city	text
street_address	text
house_number	int
floor_door	text
password	varchar(60)
active	int

14. ábra

A munkavállalók adatai kerülnek ebbe a táblába. A dolgozó neve és elérhetőségei a következő mezőkben vannak eltárolva: *first_name* - vezetéknev, *last_name* - keresztnév, *email*, *phone_number* - telefonszám, *zip* - irányítószám, *city* - város, *street_address* - utca neve és jellege, *house_number* - házszám, *floor_door* - emelet és ajtó. A munkavállalók munkaköre egy szám jelöle a *job_id* mezőben, ami a *jobs* táblában eltárolt azonosítóra mutat. Az alkalmazottak számára egy beléptető rendszer működik, ezért rendelkeznek egy jelszóval is, ami **bcrypt**-el van titkosítva, így minden esetben 60 karakter hosszúságú. Az *active* mező azt jelzi, hogy az adott személy még/már valós munkaerő és foglalkoztatott a szállodánál.

3.6.12. A jobs tábla

jobs	
job_id	int
title	text
salary	int

15. ábra

Ez a tábla a munkakörök eltárolására szolgál. A beosztás nevét és annak havi fizetését tárolja el, amik a *job_id* azonosító elsődleges kulccsal érhetőek el.

3.6.13. A job_history tábla

job_history	
employee_id	int
start_date	date
end_date	date

16. ábra

Az alkalmazottak első és utolsó munkanapját tárolja, ha még a szálloda foglalkoztatja, akkor csak a kezdés dátumát. Az *employee_id* azonosító az *employees* táblára hivatkozik.

3.6.14. Az attendance_sheets tábla

attendance_sheets	
employee_id	int
date	date
start_time	time
end_time	time
working_hours	double
break	int
status	text

17. ábra

Az *attendance_sheets* tábla a jelenléti ív digitális vezetésére szolgál. Minden foglalkoztatott az első munkanapjától kezdődően minden hónap összes napja vezetve van. Eltárolásra kerül az *employee_id* idegen kulcsban a munkavállaló azonosítója, a többi mezőben pedig az adott nap dátuma, az aznapi munka idejének a kezdete és vége, a ledolgozott órák száma, a tartott szünet hossza, illetve a napi jelenléti ív státusza.

3.7. Adatbázis kapcsolat

Az adatbázishoz való hozzáféréshez elsősorban szükség van egy php kódú kapcsolatra, amihez szintén több lehetőség társul. Ezen esetben **PDO** kapcsolattal dolgoztam, ami a következőképpen hoztam létre:

```
class Database {
    private static $conn = null;
    public static function getConnection() {
        if(is_null(self::$conn)) {
            $db = [
                'dsn' => 'mysql:host=127.0.0.1;dbname=sparetimehotel;charset=utf8',
                'user' => 'root',
                'password' => ''
            ];

            self::$conn = new \PDO($db['dsn'], $db['user'], $db['password']);
        }
        return self::$conn;
    }
}
```

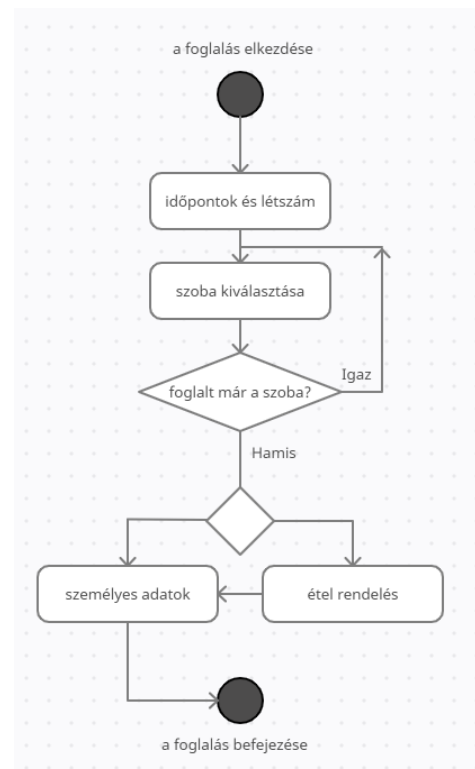
A **Database** osztályt az *app/src/db* névtérben helyezkedik el, és innentől kezdve a *getConnection()* metódusára kell hivatkoznunk, ha elérni kívánnánk az adatbázist.

3.8. Foglálás

Mint a diagramon is látható, a foglalás menete a következő: elsősorban a ki- és bejelentkezés dátumát kell megadni a létszámmal karöltve, majd az üres szoba kiválasztása a következő lépés, ami után lehetőséget kap a látogató, hogy előre megrendelje az ételeket az étkezésekre. Utolsó lépésként ki kell töltenie a foglaláshoz szükséges személyes adatokat, majd befejezheti azt.

Fontos, hogy csak a véglegesítést követően történjen az adatbázisba a beszúrás, így addig egy **session** szuper globális változóba tárolja el a program az adatokat.

Összesen öt különböző lapra van szükség a foglalás lebonyolításához. Mivel a kezdőlapon kezdődik a foglalás, még ott történik meg a dátum és létszám megadása. A maradék négy oldal a *reservationGuest* mappában kap helyet, amiknek a *ReservationGuestController* a vezérlője.



18. ábra

A *rooms.php*-ban kerül kilistázásra az összes még nem foglalt szoba, az árat már megszorozva a két dátum különbségével és a felnőttek létszámával.

Az ételek rendelése a *menu.php*-ban zajlik, ahol táblázatban kerül kiírásra az étlap az allergénekkal együtt, illetve a megrendelt termékek, amelyek szerkeszthetők. A kiíratáshoz és módosításhoz **javascript**-et is használok, amellyel **jquery ajax** segítségével küldök **post** kéréseket a **controllerek** felé. Ennek az előnye, hogy az oldal nem igényel frissítést, ha változtatást hajtunk végbe.

A *personalData.php*-ban található személyes adatok meghatározására szolgáló űrlap kitöltése után áttekinthetők a foglalás adatai, majd a „Foglalás befejezése” gombra kattintva mentésre kerül az adatbázisban.

3.8.1. Megvalósítás

A kezdőlapon történő foglalás megkezdésekor, az *actionDate* metódus hívása történik meg, és létrehozza a *reservation* indexű **session** szuper globális változót. Ez egy asszociatív tömbként funkcionál, amelynek tartalma megegyezik a foglalás adataival. A dátumokat és létszámokat eltárolja a *start_date*, *end_date*, *adult* és a *child* kulcsú tömbben. Miután végbe mentek a deklarációk, a *MainController render()* metódusával átléptetjük a látogatót a *room.php*-ra.

A *room.php*-ban kiíratásra kerülnek azok a szobák, amik nem ütköznek egy már leadott foglalással. Ezt egy sql lekérdezéssel kitudtam szűrni, így ennek köszönhetően nem lesz gond a szobafoglalással. A választást követően a program betölti az *menu.php*-t.

A menü felületen kettő táblázat látható. Az egyikben az étlap elemei vannak feltüntetve az ajánlásokkal és allergénekkal, míg a másikon a 'kosárba' helyezett ételek tekinthetők meg. Mindkét tábla **javascript** által van feltöltve, amit **DataTables** segítségével valósítottam meg. Az oldal betöltésekor hajtódik végbe az első **ajax** kérés, ami a *ReservationGuestController* egyik metódusának, az *actionFetch*-nek a meghívása. A metódus kiírja a *menu.php*-val egy helyen lévő *fetch.php* fájlba a menü rekordjait az allergének és ajánlásokkal egyetemben, már a táblázat celláiba megjeleníthető- és **json** formátumban. A metódus a *php header()* beépített függvényének segítségével átnavigálja a kérést az előbb említett *fetch.php* fájlra, aminek a tartalmát a *jquery.datatables.min.js* dolgozza fel és írja ki.

A rendelések táblázat **ajax** kérései hasonlóképpen működnek. A kiíratás ugyanezen a módon, egyedül annyi különbözik, hogy a **json** tartalmú (ez esetben *orders.php*) fájl nem a **DataTables** dolgozza fel, hanem egy általam írt függvény, mivel többször kerül meghívásra a tábla frissítése miatt, így jobban betartja a **clean code** szabályait.

A rendelés felvételkor három bemeneti értéket lehet megadni. Egyet kötelező kitölteni, a dátumot. Az étkezésekre rendelésre szolgáló legördülő listákat tetszés szerint lehet kitölteni. A „Hozzáadás” gomb lenyomásával fut le a második **ajax** kérés, ami az *actionInsert* meghívásával hozzáadja az ételeket a *reservation* szuper globális változóhoz *menu* index-el, és frissíti a **session** új adataira az *orders.php*-t. A metódus sikeres lefutásával az **ajax** frissíti a táblát az erre megírt funkció segítségével.

Törlés esetén egy újabb **ajax** kérés keretein belül ezúttal a *ReservationGuestController delete* metódusa törli az adott elemet a **session**-ből és frissíti az *orders.php*-t.

3.9. Beléptető rendszer

A bejelentkezés csak a munkavállalók részéről érhető el, éppen ezért biztonsági okokból az oldalon nincsen erre mutató hivatkozás, amely egyébként a *index.php?controller=employees&action=loginIndex*-keresztül érhető el.

A sikeres bejelentkezéshez be kell írni a megfelelő **email** – **jelszó** párost a két bemeneti értékhez, ami **post** változóban adódik tovább az *actionLogin* metódusnak. Elágazásokkal megvizsgáljuk a kapott email adatbázisban való létezését, majd az ahhoz tartozó jelszó egyezését a *doLogin* metódussal, amiben a *password_verify()* beépített függvény segítségével ellenőrzzük a titkosított jelszó tárolás miatt. Amennyiben minden egyezik, létrejön az *employee_id* indexű **session** változó, aminek az alkalmazott *employees* táblában szereplő azonosítója az értéke.

Alapesetben minden felhasználónak **pass** a jelszava.

3.10. Munkakörök és funkciók

A követelményekben megfogalmazott probléma megvalósításához jogosultságkezelést valósítottam meg.

- jogosultságok:
 - reservation
 - employee
 - reports
 - orders
 - menu
- funkciók:
 - foglalások
 - alkalmazottak
 - hibabejelentések

- rendelések
- menü
- jelenléti ív
- hibabejelentés

Minden munkakör hozzáfér a saját jelenléti ívéhez, és képes hibát bejelenteni. Ezen felül megvalósult a funkciók jogosultságához való tartozása. Az alábbi táblázatban látható mely jogosultság milyen jogkörrel rendelkezik.

	reservation	employee	reports	orders	menu
foglalások	×				
alkalmazottak		×			
hibabejelentések			×		
rendelések				×	
menü					×

	igazgató	igazgatóhelyettes	porta	karbantartó	HR	séf	takarító	pincér
reservation	×	×	×					
employee	×	×			×			
reports	×	×		×				
orders	×	×				×		×
menu	×	×				×		
hibabejelentés	×	×	×	×	×	×	×	×
jelenléti ív	×	×	×	×	×	×	×	×

A jogosultságok kiosztása a *Jobs* modelben történik a *currentUserCan* metódussal. A jelenleg beléptetett alkalmazott **session**-ben eltárolt *employee_id*-ja alapján lekérjük a munkakörét, és ahhoz rendeljük, mely jogosultságokkal rendelkezik.

```
public static function currentUserCan($x) {
    $user = Employees::getCurrentUser(); if(is_null($user)) return false;
    $functions = [
        'function.reservation' => ['igazgató', 'igazgatóhelyettes', 'porta', ],
        'function.report' => ['igazgató', 'igazgatóhelyettes', 'karbantartó', ],
        'function.employee' => ['igazgató', 'igazgatóhelyettes', 'hr', ],
        'function.orders' => ['igazgató', 'igazgatóhelyettes', 'séf', 'pincér'],
        'function.menu' => ['igazgató', 'igazgatóhelyettes', 'séf'],
    ];
    if (in_array($user->getJob($_SESSION['employee_id'])[0], $functions[$x]))
        return true;
}
```

Amikor egy kódrészletet jogosultság függővé kell tenni, azt egy elágazásba helyezzük bele, melynek a feltétele: *currentUserCan(<funkció_név>)*.

```
<?php if (Jobs::currentUserCan('function.employee')) : ?>
```

Minden oldal elején van egy elágazás, ami következtében csak az adott jogkörrel rendelkezőket engedi tovább, máskülönben a kezdőlapra navigálnak.

Az összes oldal **DataTables** használatával **ajax** kérésekkel működik, akárcsak a foglalás alatt az étel megrendelése. Vannak alap funkciók, amelyek csaknem minden oldalon megegyeznek.

3.10.1. Rekordok kiírása

Minden esetben az oldal betöltékor kerül ábrázolásra. Az **ajax** kérés ekkor valósul meg az *actionFetch* felé, aminek átadja a megjelenítés módjához szükséges **keresési** és/vagy **rendezési** értékeket. Ennek megfelelően írja meg az adatbázis felé a lekérdezést, amit rendre az adott mappában lévő *fetch.php*-ba kiírat **json** formátumba, majd ráhivatkozik. A **DataTables** ezt a fájlt feldolgozza, és ábrázolja a táblázatban. Erre az **ajax** lekérdezésre létrehoztam egy *dataTables* nevű változót, amit szükség esetén az újratöltésre is használható.

3.10.2. Rekordok hozzáadása

A rekord hozzáadás szintén minden táblázatnál opcionális. A táblázat felett található hozzáadás gomb aktivitásánál **javascript** segédkezesével kerül megnyitásra a **modal** ablak. Ez minden esetben más, de az aktuális rekordkérésnek eleget tesz. A bemeneti mezők kitöltése után a „*Felvétel*” gombra kattintva az űrlap teljes egészét átküldi az *insert* metódusnak, ami az adott **model** osztályokat használva feltöltési a rekordokat.

3.10.3. Rekordok törlése

A törlés nem minden táblánál megadott funkció, mivel a legtöbb esetben az adatokat elakarjuk tárolni. Némelyik helyen mégis opció, ott pedig az **ajax** kérés a *delete* metódusnak adja át az aktuális id-t, amivel egyszerűen törlésre kerül a rekord, végül sikeres eredmény esetén frissítjük a **dataTable** változót.

3.10.4. Rekordok szerkesztése

A szerkesztés gomb lenyomásával átadódik az azonosító az *actionFetchSingle* metódusnak, ami az **id** alapján megkeresi az ábrázolni kívánt rekord adatait, és a *fetchSingle.php*-ba **json** formátumban beilleszti, majd ráhivatkozik. Sikeres visszatérés esetén az **ajax** kérésben értékül adjuk a megjelenő **modal** ablak bemeneti mezőinek a visszatérési értéket.

A szükséges változtatások után a „*Változtat*” gombra kattintva az *actionInsert* metódus dolgozik tovább az adatokkal. **Post** kérdésben megkapta az összes bemeneti értéket, és az azonosító által a **model** osztályokban található *update* metódusokon keresztül frissülnek a rekordok, a táblázattal egyetemben.

3.11. Tesztelés

A **whoops** használata a fejlesztés alatt számtalanszor megkönnyítette a dolgomat. A kulturált hibakezelő felület rengeteg információval lát el, ami sokszor rendkívül hasznosnak bizonyult. Az **ajax** kéréseknél szintén rengeteg hibát szűrtem ki a **DevTools Network** pontjában, ahol az **XmlHttpRequest** töréspontjait végig tudtam ellenőrizni. Alapvetően igyekeztem a fejlesztés során szűrni minden hibát, és végül az alábbi táblázatban lévő jelentősebb kivétneivalók gyűltek össze.

Hiba	Megoldás
Hibás <i>controller</i> és/vagy <i>action</i> érték adás esetén hibaüzenetet kaptunk.	Az <i>index.php</i> -ban az irányításért felelős elágazások utolsó <i>else</i> paraméterei a kezdőlapra mutatnak.
Számos metódusban egymás után több sql lekérdezést hajtok végre, amiknek egy része egy hirtelen oldal lehalás alatt rosszul futhat le.	Tranzakció kezelés.
Helytelenül beírt adatot is beszűrt az adatbázisba.	Regex használata a html -ben és javascript -ben egyaránt.
A hibabejelentés naplózásánál a számlák árát összeadja, azonban ha az egyik <i>input</i> üres, akkor <i>NotaNumber</i> üzenetet dobott.	Javascript -ben egy elágazás, ami üres érték esetén 0-át ad vissza.
A bejelentkezésnél nem dobott hibaüzenetet sikertelen próbálkozásnál.	Ajax kérésre írtam át a bejelentkezést, így a visszatérő értéket ki tudtam írni.
A rekord törlés funkció nem működött számos táblázatnál.	A törlés igényének újragondolása az összes táblára rávetítve, és ahol még sincs rá szükség, ott az opció törlése.

4. Felhasználói dokumentáció

A **Spare Time Hotel** honlapjának a célja, hogy a látogatók gyors online szobafoglalását, valamint az alkalmazottak adminisztrációs kötelezettségeit egyszerű és hatékony munkává alakítását biztosítsa.

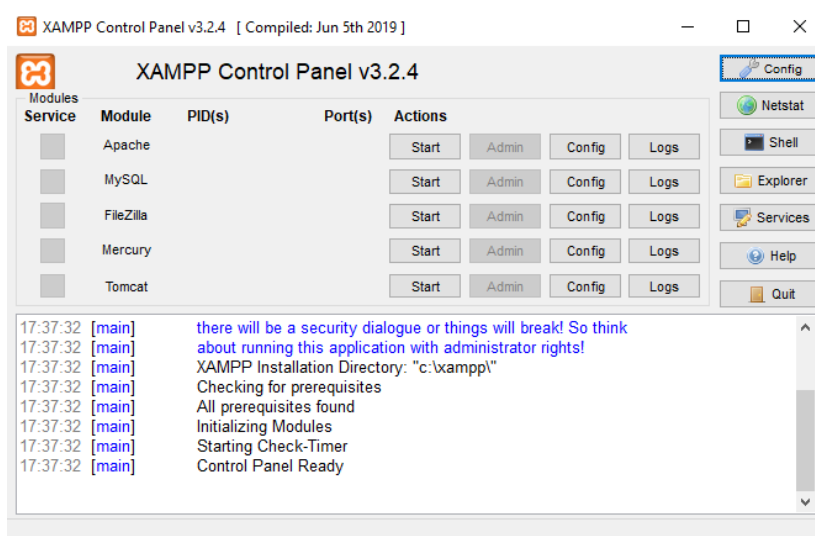
4.1. Telepítés és használat

4.1.1. Telepítés

A program futtatásához csak egy **xampp**-ra és egy böngészőre van szükség. A **xampp** fogja nekünk biztosítani a virtuális **Apache** webszervert, valamint az **adatbázis**, és **php** komponenseket. Az alábbi linken tudja letölteni:

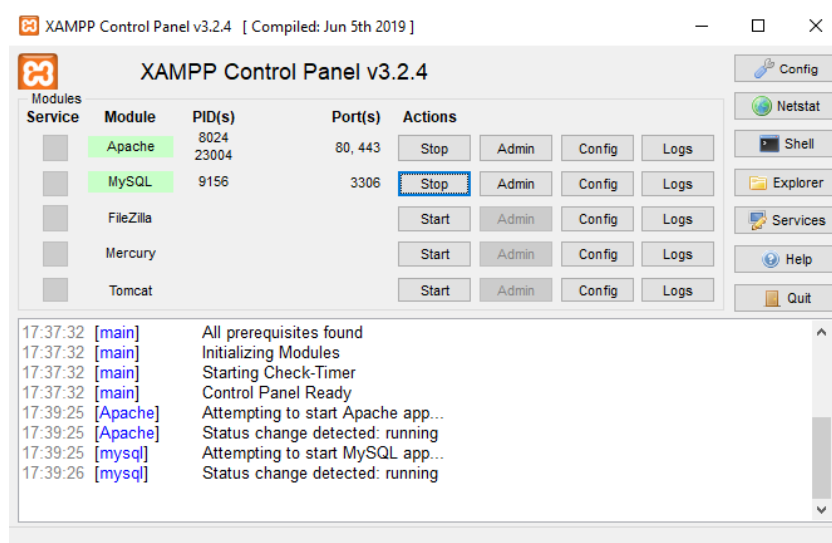
<https://www.apachefriends.org/hu/download.html>

A letöltés után telepítse az alkalmazást, az alapértelmezetten beállított *C:/xampp/* mappába. Miután végbe ment a telepítés, és megnyitotta az alkalmazást, az alábbi, vagy hasonló felületet kell, hogy kapja.



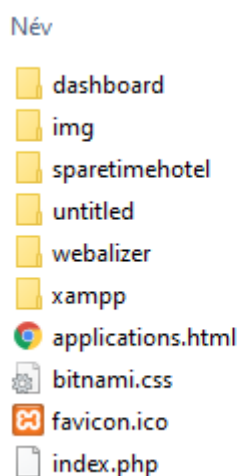
19. ábra

Indítsa el a „Start” gombok lenyomásával az **Apache** és **MySQL** elemeket. Miután betöltött, így kell kinéznie:



20. ábra

Ezt követően navigáljon el a *C:/xampp/htdocs/* mappába, és csomagolja ki oda a *sparetimehotel.zip* tömörített fájlt. Amennyiben most telepítette fel az alkalmazást, az alábbi módhoz hasonlóan kell, hogy kinézzen az érintett mappa.

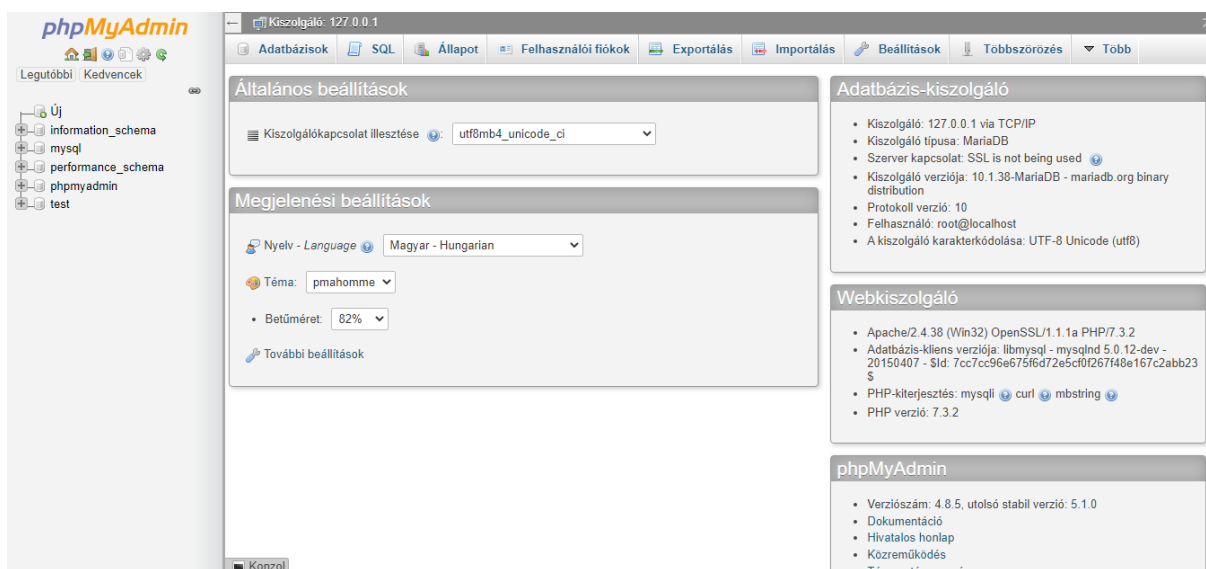


21. ábra

A következő lépés, hogy megnyissa a böngészőjében a **phpMyAdmin**-t, amit ezen a linken érhet el:

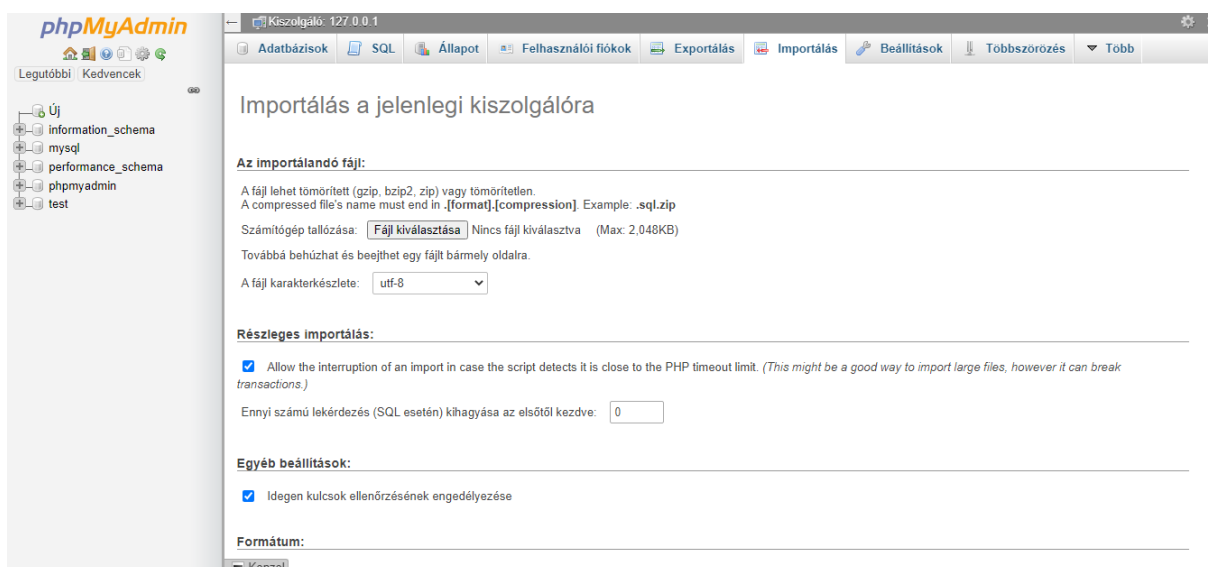
<http://localhost/phpmyadmin/>

A következő felület kell, hogy fogadja:

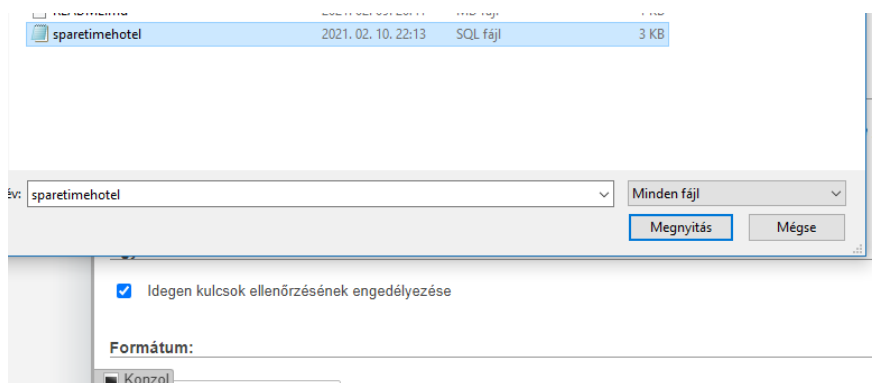


22. ábra

Kattintson a fent levő lévő navigációs fülön az „importálás” menüre. Az itt található „Fájl kiválasztása” pontra kattintva válassza ki a *C:/xampp/htocs=sparetimehotel/sparetimehotel.sql* fájlt, és nyissa meg.

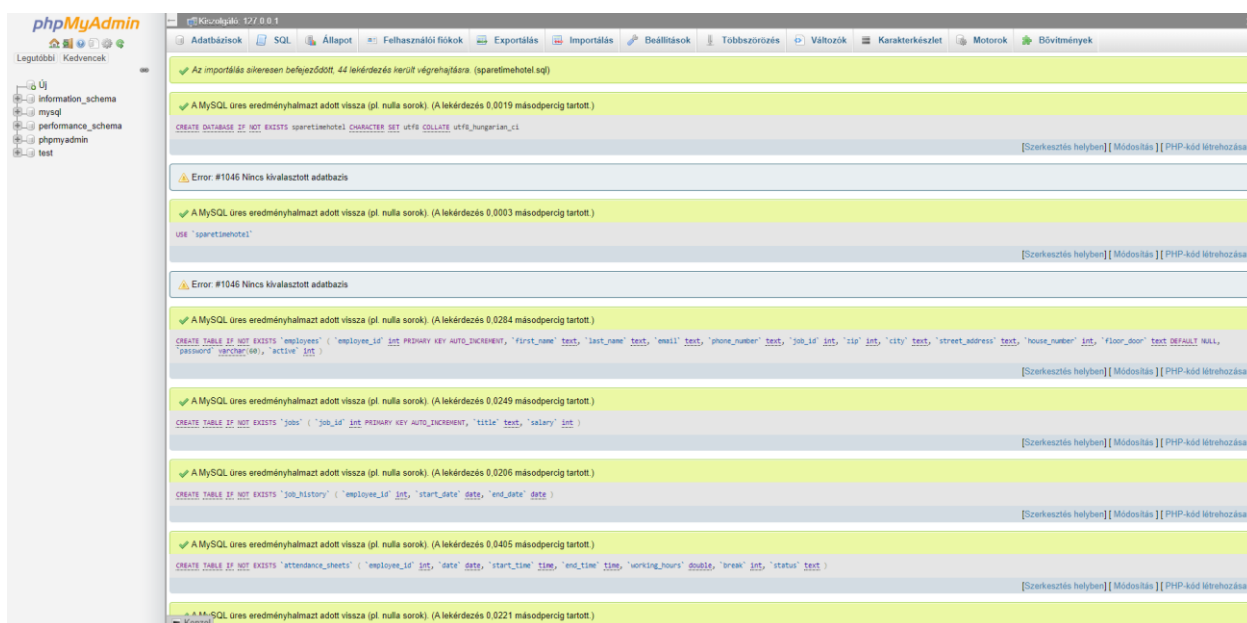


23. ábra



24. ábra

Tekerjen az oldal aljára, és nyomja meg az „Indítás” gombot. Amennyiben mindent jól csinált, beimportálta az adatbázist a rekordokkal együtt.



25. ábra

Írja be a böngésző címsorába az alábbi linket:

<http://localhost/sparetimehotel/>

Amennyiben mindent pontosan az útmutató alapján csinált, az alábbi felület kell, hogy fogadja.



26. ábra

4.1.2. Foglалás

Az oldal betöltésekor a kezdőlapra találja magát a látogató. Lejjebb görgetve megtalálható a „Foglalás megkezdése” blokk, ahol a be- és kijelentkezés dátumát, valamint a felnőttek és gyermekek létszámát kell megadni. A dátumok minimum értéke a mai naptól a holnapiig, míg legalább 1 felnőttet igényel a foglalás.

27. ábra

A „Tovább” gombra kattintva megjelen szobák kiválasztása oldal. Itt az elsősorban kiírásra kerül a létszám, valamint az éjszakák száma. A szobák kártyákon vannak ábrázolva, amiken az adott szoba végleges ára látható. Csak azok a szobák kerülnek megjelenítésre, amelyek még nincsenek ebben az időpontban lefoglalva.

Szoba > Menü > Személyes adatok

Szoba kiválasztása

Az árba bele van számítva a(z) 1 felnőtt és a(z) 1 éjszaka.

1. emelet	1. emelet	1. emelet
3 darab kétszemélyes ágy	3 darab kétszemélyes ágy	2 darab kétszemélyes ágy
extrák: jakuzzi	extrák: kanapé	extrák: kanapé
15000 Ft	17000 Ft	16000 Ft
Választ és tovább	Választ és tovább	Választ és tovább

1. emelet	2. emelet	2. emelet
-----------	-----------	-----------

28. ábra

A szoba kiválasztása lehetőséget kapunk az ételeket előre megrendelni, az ott tartózkodásunk alatti étkezésekre. Egy táblázatban szemléltetésre kerül az étlap, amelyen a keresésen felül, a rendezési képessége is rendelkezésre áll a fejléc cellákra kattintva.

Szoba > Menü > Személyes adatok

Nem köteles azonnal megrendelni az ételt, a bejelentkezéskor is megteheti.

Rendelés

Eddigi rendelések:

Dátum	Reggeli	Ebéd	Vacsora	Ár	Törölés
Tovább					

Menü:

10

Keresés:

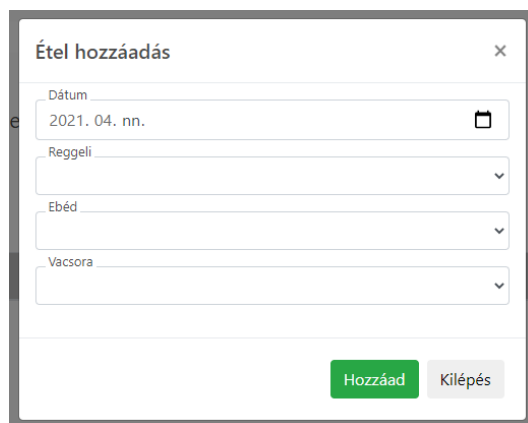
Név	Allergének	Ár	Ajánlat
vegyes savanyúság	kén-dioxid	699 Ft	
vegyes savanyúság	kén-dioxid	699 Ft	
túró gombóc	glutén, diófélék, tojás, tej	700 Ft	
tócsni	glutén, tojás, tej	650 Ft	
tatár beefsteak	glutén, mustár, tojás	1500 Ft	séf kedvence
tarja steak	glutén, mustár, tojás, tej	1600 Ft	hónap ajánlása
svédasztalos reggeli		1500 Ft	
somlói galuska	glutén, diófélék, tojás, tej	600 Ft	
sajtkrémleves	glutén, zeller, tojás, tej	750 Ft	
roston zöldség		600 Ft	

1-10 / 33

1 2 3 4

29. ábra

A „Rendelés” gombra kattintva az alábbi ablak ugrik fel, ahol a dátum és az ételek választhatók ki.



30. ábra

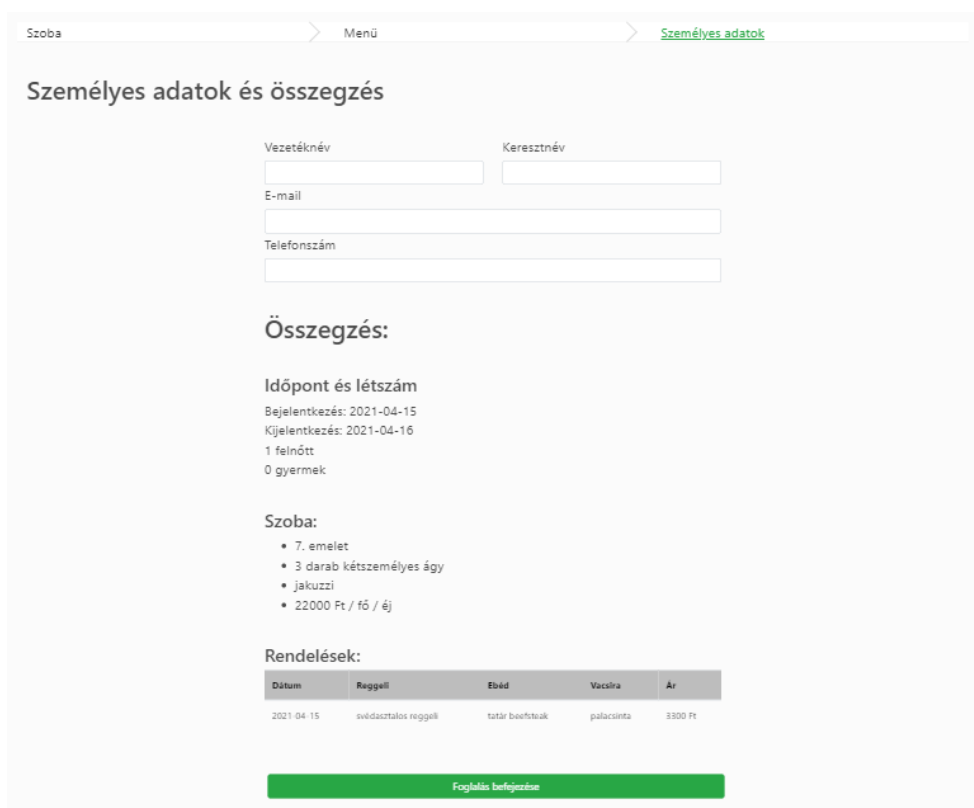
A hozzáadás után bekerül az „Eddigi rendelések” táblázatba, ahol szükség szerint törölhető és/vagy újra felvehető.



Dátum	Reggeli	Ebéd	Vacsora	Ár	Törölés
2021-04-15	svédasztalos reggeli	tatár beefsteak	palacsinta	3300	

31. ábra

Rendelést nem muszáj kitölteni, üresen hagyva is lehetőségünk van tovább haladni.



Személyes adatok és összegzés

Vezetéknév: Keresztnév:

E-mail:

Telefonszám:

Összegzés:

Időpont és létszám
Bejelentkezés: 2021-04-15
Kijelentkezés: 2021-04-16
1 felnőtt
0 gyermek

Szoba:

- 7. emelet
- 3 darab kétszemélyes ágy
- jakuzzi
- 22000 Ft / fő / éj

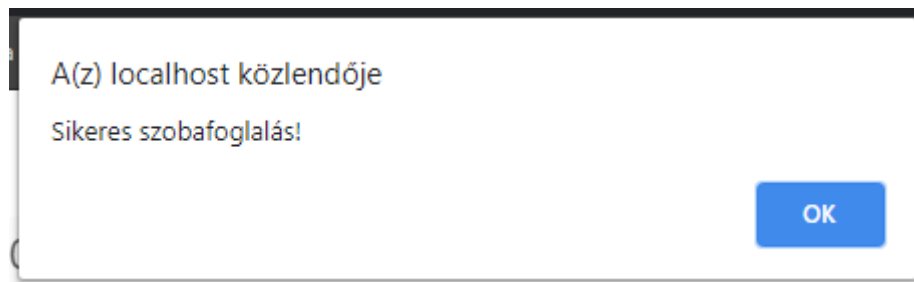
Rendelések:

Dátum	Reggeli	Ebéd	Vacsora	Ár
2021-04-15	svédasztalos reggeli	tatár beefsteak	palacsinta	3300 Ft

Foglalás befejezése

32. ábra

Az adatok helyes kitöltése és a foglalás befejezése után, a következő felugró üzenet látható:



33. ábra

Az „OK” gombra kattintva visszanavigál minket a kezdőlapra.

4.1.3. Bejelentkezés

Az alkalmazottak bejelentkezése a következő linken érhető el:

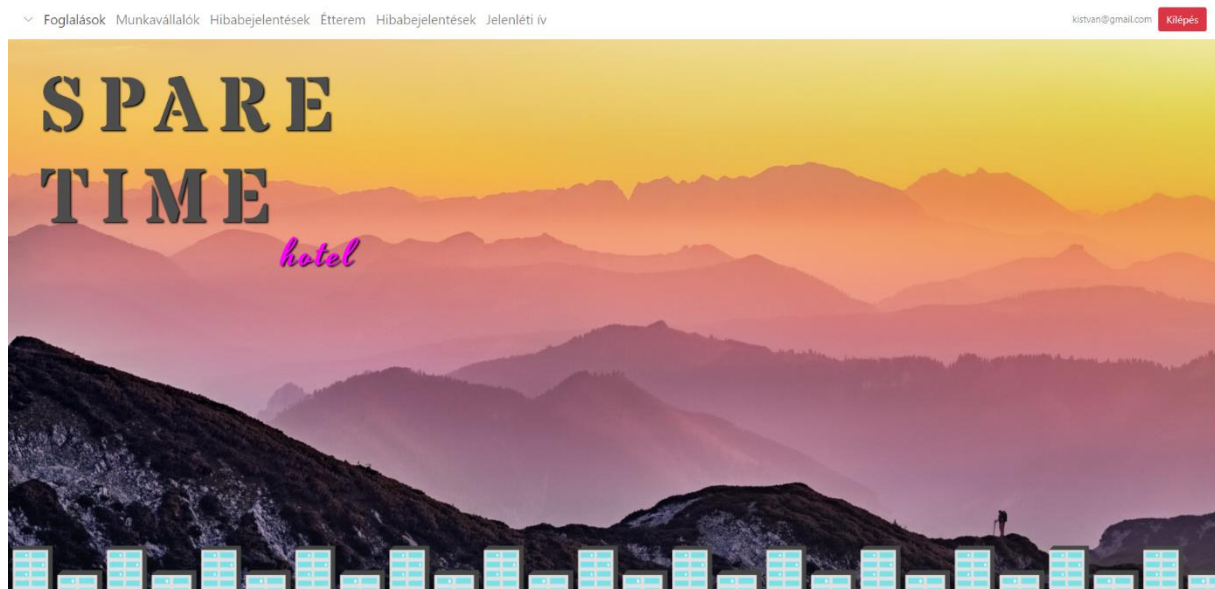
<http://localhost/sparetimehotel/index.php?controller=employees&action=loginIndex>

Az oldal betöltéskor az alábbi tartalom látható.

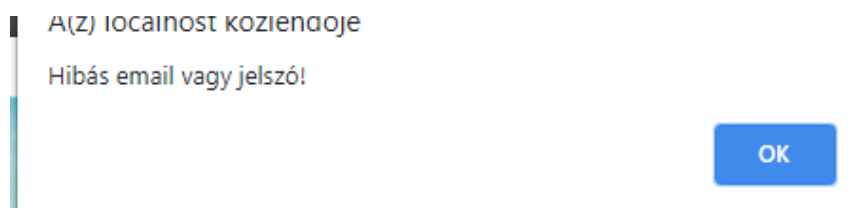
A login form with two input fields: "email" and "jelszó". To the right of the fields is a green button labeled "Belépés".

34. ábra

A felhasználó megkísérelheti a bejelentkezést, egy email cím és jelszó beírásával. Amennyiben sikeres, átdobja a kezdőlapra, különben egy hibaüzenetet ír ki.



36. ábra



35. ábra

4.1.4. Jogosultságkezelés

A navigációs blokkon csak és kizárólag azok a menüpontok jelennek meg, ami az adott alkalmazottnak a jogköréhez tartozik.

▼ Foglalások Munkavállalók Hibabejelentések Étterem Hibabejelentések Jelenléti ív

37. ábra

4.1.5. Porta

A portás saját munkakörének elvégzése érdekében, a foglalásokat kell tudnia kezelni. A menüpontra kattintva az alábbi felület fogadja.

[Felvétel](#)

10 ▼

Keresés:

Sorszám	Név	E-mail	Telszám	Fő	Szoba	Bejelentkezés	Kijelentkezés	Ár	Szerkesztés	Check
176	Szatmáry Géza	szatmary.geza@gmail.com	06701514962	1 felnőtt, 2 gyerek	4	2021-10-10	2021-10-11	18100 Ft		
175	Takáts István	takats.istvan@gmail.com	06302856527	2 felnőtt, 0 gyerek	24	2021-10-09	2021-10-10	46970 Ft		
174	Gyórfi Nikolett	gyorfi.nikolett@gmail.com	06307186270	1 felnőtt, 2 gyerek	25	2021-10-08	2021-10-09	23849 Ft		
173	Csorba Dominik	csorba.dominik@gmail.com	06702971749	2 felnőtt, 1 gyerek	13	2021-09-03	2021-09-06	108900 Ft		
172	Sós Ágnes	sos.agnes@gmail.com	06702502598	2 felnőtt, 2 gyerek	15	2021-08-12	2021-08-14	78670 Ft		
171	Hajdú Anna	hajdu.anna@gmail.com	06702487593	3 felnőtt, 1 gyerek	7	2021-08-05	2021-08-09	185505 Ft		
170	Dömötör Iлона	domotor.ilona@gmail.com	06705654086	4 felnőtt, 2 gyerek	3	2021-07-08	2021-07-12	283400 Ft		
169	Fehér Gábor	feher.gabor@gmail.com	06204780463	5 felnőtt, 2 gyerek	3	2021-07-07	2021-07-09	183250 Ft		
168	Hajnal András	hajnal.andras@gmail.com	06201640381	3 felnőtt, 0 gyerek	27	2021-07-04	2021-07-13	683250 Ft		
167	Tóth Elemér	toth.elemer@gmail.com	06206632964	1 felnőtt, 0 gyerek	3	2019-11-06	2019-11-12	103450 Ft		

1-10 / 176

1

2

3

4

5

...

18

38. ábra

Előfordulhat, hogy helytelenül lettek megadva bizonyos

Foglalás hozzáadás ✕

Név

Felnőtt

Gyerek

Bejelentkezés

[Új vendég](#)

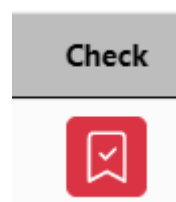
Szoba

Kijelentkezés

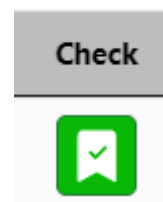
[Hozzáad](#)
[Kilépés](#)

40. ábra

adatok a szobafoglalás alatt. Ennek a kiküszöbölése



39. ábra



41. ábra

érdekében szerkeszthetők az adatok. A csekkolás a táblázat jobb oldali „Check” oszlopában lévő gombok rányomásával lehetséges. Foglалás felvétele esetén a „Felvétel” gomb kattintásával nyitjuk meg az

ehhez szükséges panelt. A „Név” legördülő listában szerepelnek az eddigi vendégek nevei. Új látogató esetén az „Új vendég” gomb által megnyíló ablakba kell beírni az adatait. A regisztrálás után visszatér az előző panelre, immár a hozzáadott névvel ellátott legördülő listával. A bemeneti értékek kitöltése után, a hozzáadást követően egyből bekerül az új foglalás a táblázatba.

43. ábra

42. ábra

4.1.6. HR

A munkavállalók kezeléséhez úgyszintén egy táblázat szolgál. Lehetőség van az alkalmazottak adatainak a szerkesztésére, valamint az új munkavállalók felvételére.

							Felvétel
10 ▾							Keresés: <input type="text"/>
Név	Munkakör	Telefonszám	E-mail	Lakcím	Szerkesztés	Aktivitás	
Varga Zsuzsanna	porta	varga.zsuzsanna@gmail.com	0670111111	1119 Budapest Andor utca 40			
Sós Zoltán	séf	sos.zoltan@gmail.com	06300544187	1037 Budapest Flóra árok 95 5/94			
Papp Gyula	takarító	papp.gyula@gmail.com	06206156302	1111 Budapest Evelin ösvény 80 1/6			
Nagy Ágnes	karbantartó	nagy.agnes@gmail.com	06206871464	1112 Budapest Fodor lejáró 90			
Kovács Barbara	porta	barbara.kovacs@gmail.com	06307916852	1173 Budapest Liszt Ferenc utca 5			
Kocsis István	igazgató	kistvan@gmail.com	06307894561	1118 Budapest Bozókvar utca 11			
Kis Anna	hr	kis.anna@gmail.com	06308631703	1050 Budapest Dalma sétány 91			
Kelemen János	takarító	kelemen.janos@gmail.com	06304675091	1065 Budapest Máté útja 4			
Jónás Katalin	igazgatóhelyettes	jonas.katalin@gmail.com	06308974562	1096 Budapest Ernő u. 23			
Deák Eszter	pincér	deakeszter@gmail.com	06309790617	1153 Budapest Livia lejáró 10 5/90			
1-10 / 11							1 2

44. ábra

45. ábra

A munkavállaló felvétele a következőképpen zajlik: ki kell tölteni minden kötelező bemeneti értéket. Az email címnek és telefonszámnak érvényesnek kell lenni, valamint a jelszavaknak meg kell egyeznie. Az első munkanap kiválasztása kötelező. Adatszerkesztéskor ugyanezt az ablakot kapjuk, csupán kitöltve. A szerkesztéshez írjuk át a változtatni kívánt adatokat, majd a „Változtat” gombbal frissítjük a rekordot.

Az alkalmazott aktivitása azt jelzi, hogy jelenleg a szálloda foglalkoztatja-e. Egyszerű kattintással tudjuk változtatni.

A saját jelenléti ív kezelésére minden alkalmazott jogosult, azonban csak a **HR**-es képes a szálloda által foglalkoztatott összes dolgozóéhoz hozzáférni. Ezt a munkakört betöltő ember feladata minden hónapban egyszer hozzáadni az adatbázishoz, az összes alkalmazott számára a hónap minden napját. Ezt a „Jelenlegi hónap hozzáadása mindenkinek” gomb segítségével egy kattintásba kerül. Ilyenkor az alapértelmezett 08:00 és 16:00 óra között tartó 8 órás munkaidőt adja hozzá, üres állapottal.

Név	Dátum	Kezdés	Befejezés	Ledolgozott órák	Szünet	Állapot	Szerkesztés
Kocsis István	2021-05-29	08:00:00	16:00:00	7.5 óra	30 perc	aláírt	
Kovács Barbara	2021-05-29	08:00:00	16:00:00	7.5 óra	30 perc	aláírt	
Papp Gyula	2021-05-29	08:00:00	16:00:00	7.5 óra	30 perc	aláírt	
Kelemen János	2021-05-29	08:00:00	16:00:00	7.5 óra	30 perc	aláírt	
Nagy Ágnes	2021-05-29	08:00:00	16:00:00	7.5 óra	30 perc	aláírt	
Kis Anna	2021-05-29	08:00:00	16:00:00	7.5 óra	30 perc	aláírt	
Varga Zsuzsanna	2021-05-29	08:00:00	16:00:00	7.5 óra	30 perc	aláírt	
Deák Eszter	2021-05-29	08:00:00	16:00:00	7.5 óra	30 perc	aláírt	
Sós Zoltán	2021-05-29	08:00:00	16:00:00	7.5 óra	30 perc	aláírt	
Jónás Katalin	2021-05-29	08:00:00	16:00:00	7.5 óra	30 perc	aláírt	

46. ábra

47. ábra

Az állapotot a lenyíló legördülő listában rákattintva lehet megváltoztatni, ami egy megerősítő üzenet elfogadása után meg is történik.

A szerkesztés lehetősége fen áll, hiszen talán itt van a legnagyobb jelentősége. Miután az alkalmazottak megkapták a havi jelenléti ívüket, azt naponta aláírják, illetve ha szükséges szerkesztik.

48. ábra

4.1.7. Karbantartó

A karbantartó felelőssége a hotel rendben tartása. A „Hibabejelentések” menüpontra kattintva mindenkinek ezt a felületet hozza be, annyi különbséggel, hogy csak a karbantartó képes naplózni és szerkeszteni. Bejelentés felvétele esetén, az ennek megfelelő gombra kattintva jön elő a panel.

Szoba	Típus	Emelet	Státusz	Üzenet	Bejelentés	Naplózás	Szerkesztés	Befejezve
32	szoba	7	tönkrement	a lámpa	2021-04-14			
11	takarító szertár	3	egyéb	dsa	2021-04-14			

49. ábra

A helységek, illetve a státusz között lehet választani, aminek három állás van:

- tönkrement
- meghibásodott
- egyéb

A megjegyzést kötelező kitölteni, hogy pontosabb leírást tartalmazzon a bejelentés.

Szerkesztés esetén ugyanúgy ehhez hasonló ablakot kell használnunk, csak kitöltve az adatokkal.

50. ábra

Ahhoz, hogy befejezettnek lehessen tekinteni a bejelentést a javítás után naplózni kell. Az

51. ábra

ehhez szükséges panelt a „Naplózás” alatti plusz gombbal lehet megnyitni. Az első bemeneti mező egy legördülő lista, ahonnan a munkát elvégző karbantartót kell kiválasztani. A befejezés dátumát megjelölni kötelező, a megjegyzést kitölteni csak szükség esetén. Előfordulhat, hogy a javításhoz költségeket kellett. Ezen esetben fel kell tölteni a számlát lefotózva, és az azon szereplő árat külön meg kell jelölni. Több számla esetén a „+” gombbal adhatunk meg új sort.

52. ábra

Ezután a korábbi sorok törölhetők lesznek, ha viszont az utolsó sortól szeretnénk megszabadulni, egyszerűen töröljük ki az ár és fájl értékét, vagy adjunk hozzá egy új sort, és a jelenlegit töröljük. A hozzáadást követően a „Befejezve” oszlopban zöld pipára cserélődik az ikon. Később lehetőségünk van a naplózások szerkesztésére is, a „Naplózás” oszlopban a befejezett javításoknál, egy nagyító látható, ami segítségével hozható elő az ablak.

4.1.8. Séf és pincér

Az összefüggés a két munkakör között, hogy mindketten látják a rendeléseket és a menüt, azonban csak a séf képes szerkeszteni az utóbbit. A két oldal felülete a következőképpen néz ki:

Rendelések

Menü

Felvétel

10

Keresés:

Név	Dátum	Reggeli	Ébéd	Vacsora	Szerkesztés
Tóth Elemér	2021-11-06	orjaleves	cézár saláta	kemencében sült camembert	
Szatmáry Géza	2021-10-10		édes burgonya	carbonara spagetti	
Takáts István	2021-10-09	svédasztalos reggeli	cézár saláta	hasbáburgonya	
Gyórfi Nikolett	2021-10-08		rántott sajtszeletek	vegyes savanyúság	
Csorba Dominik	2021-09-03	svédasztalos reggeli	rántott csirkemell csikók	sajtkrémleves	
Sós Ágnes	2021-08-12	svédasztalos reggeli	rántott sajtszeletek	krokett	
Hajdú Anna	2021-08-05		krokett	rántott sajtszeletek	
Dömötör Ilona	2021-07-08	svédasztalos reggeli	tócsni	francia hagymaleves	
Fehér Gábor	2021-07-07		rántott sajtszeletek	tatár beefsteak	
Hajnal András	2021-07-04		carbonara spagetti	csirke saláta	

1 - 10 / 178

1

2

3

4

5

...

18

53. ábra





















Rendelések

Menü

Felvétel

10

Keresés:

Név	Allergének	Ár	Érvényesség	Ajánlat	Szerkesztés	Törlés
vegyes savanyúság	kén-dioxid	699	igen			
uborkasaláta	diófélék, puhatestűek, szójabab	300	nem			
túró gombóc	glutén, diófélék, tojás, tej	700	igen			
tócsni	glutén, tojás, tej	650	igen			
tatár beefsteak	glutén, mustár, tojás	1500	igen	séf kedvence		
tarja steak	glutén, mustár, tojás, tej	1600	igen	hónap ajánlása		
svédasztalos reggeli		1500	igen			
somlói galuska	glutén, diófélék, tojás, tej	600	igen			
sajtkrémlives	glutén, zeller, tojás, tej	750	igen			
roston zöldség		600	igen			

1-10 / 33

1

2

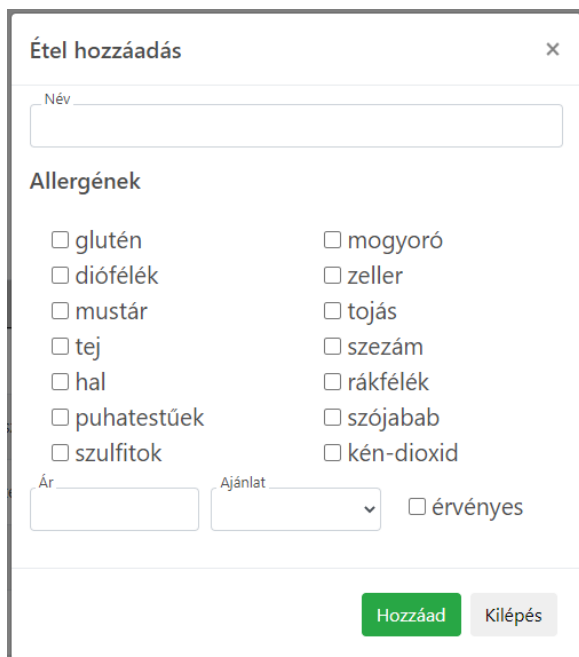
3

4

54. ábra

A rendeléseket ábrázoló felület funkcionalitásban teljesen megegyezik a **foglalás**-ban fellelhető étel rendeléssel, egyetlen kivétel, hogy itt minden vendég összes rendelése

megtekinthető, és szerkeszthető. A hozzáadás felület is ugyanúgy néz ki, a szerkesztéssel egyetemben.

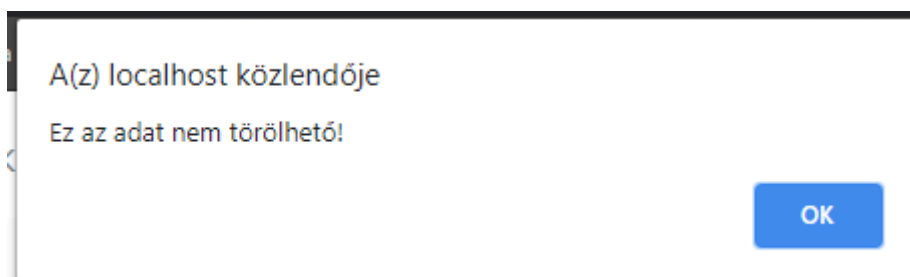


55. ábra

A menüvel kapcsolatos opciókhoz a megtekinthetőség kivétel, mint már fentebb említettem, csak a séfnek van jogosultsága.

Ételfelvétel esetén a balra látható panelt kapjuk szolgálatul. Megadjuk az étel nevét, a jelölésköteles allergének közül kiválasztjuk a relevánsakat, majd az ár megjelölése után már hozzá is adható az étlaphoz. Az ajánlat legördülő listában a *séf kedvence*, és a *hónap ajánlása* között választhatunk. Az **érvényesség** az étlapon való megjelölést biztosítja. Ez azért fontos, mert ha egy étel lekerül az étlapról, nem kerül törlésre, hogy később is visszatekinthető legyen.

Törölni csak azokat a rekordokat lehet, amelyeket még senki nem rendelt meg. Ezzel biztosítjuk az esetleges téves hozzáadás törlésének lehetőségét. Előfordulhat azonban, hogy véletlenül rosszat törölünk, de nincs mitől félni, mivel a nem törölhető adatokra hibaüzenetet dob a rendszer.



56. ábra

5. Összegzés

A célom, miszerint a **Spare Time Hotel** honlapja könnyen és gyorsan kezelhető a praktikusság függvényében, úgy érzem, hogy teljesült. Sikerült minden eltervezett funkciót létrehoznom, és rengeteg új dolgot tanulhattam az elkészítés alatt.

6. Felhasznált irodalmak

Sherer Balázs, 2010: *Rendszertervezés (Embedded System Design)*,
https://www.mit.bme.hu/system/files/oktatas/targyak/7237/vimim238_jegyzet_2v4.pdf

Répásné Babucs Hajnalka, *Programtervezés folyamata, dokumentumok*