**1. Milyen címet adna a ma érintőleg említett szakdolgozati témájának?**

Repülőtéri parkolási rendszer fejlesztése

**2. Mi lenne az alcím?**

Webes alapú időpont-, és szolgáltatásfoglaló rendszer kiépítése

**3. Hogyan nézzen ki a cím angolul?**

Developement of an Airport parking lot management System

**4. Mi legyen az alcím fordítása angolra?**

Building of a web based appointment and service booking system

**5. Miként írná le kb. 1000 karakterben a dolgozat lényegét (vö. kivonat) = célok, célcsoportok, hasznosság, feladatok, motiváció**

**Cél**: Az alkalmazás célja, hogy a megrendelő cég ügyfelei számára egyszerűvé tegye az időpont foglalást, és a szolgáltatások megrendelését.

**Hasznosság:** Az alkalmazás lehetővé teszi, a foglalás mellett egy azonnali árkalkulációt, amely akár bankkártyás fizetéssel azonnal rendezhető (biztonsági okokból, ez külső szolgáltatás, az alkalmazás csak a sikeres/sikertelen fizetésről fog tájékoztatást kapni). Lehetséges lesz továbbá a dolgozók számára, (authentikáció és authorizáció után) a foglalások áttekintése, helyszíni személyes módosítás esetén a megrendelések módosítása, a megvásárolt szolgáltatások megtekintése, valamint természetesen a repülőtéri transzfer miatt, a járatok áttekintése, hogy tudják, hogy pontosan mikor kell kimenniük a repülőtérre az ügyfélért (ismét külső szolgáltatás pl.: flightradar24 API).
**Feladatok:**

Áttekintés:

1. Adatbázis terv elkészítése, implementációja.
2. Backend endpointok/servicek megírása
3. Authentikáció, authorizáció beépítése dolgozók számára
4. Front-end design elkészítése.
5. UX/UI design fejlesztése
6. Alkalmazás containerizálása egyszerű deployment érdekében
7. Deployment

MVP (Minimum Viable Product):

* Home page, marketing célokra (miért válassza a céget stb.)
* Időpont foglalás + szolgáltatások kiválasztása
* Foglalás kezelése/kereshetősége rendszám/telefonszám/email alapján.
* CRUD operációk a foglalásokkal:
	+ Elmentett adatok úgymond „bufferbe” kerülnek, csak azután módosítható, miután az ügyfél az emailben elküldött linkkel jóváhagyta. (biztonságosabb, mint csak kifrissíteni az adatokat bárki kérésére).
	+ Kifizetett foglalás módosítható? Hogyan legyen elszámolva, ha kér plusz szolgáltatást vagy lemond valamit (esetleg lemondani már nem lehet)?
* RoBA (Role Based Access) dolgozók számára. Nem kell mindenkinek minden információt látnia, csak az adott munkakörhöz tartozó feladatok ellátásához szükséges információk.

**Célcsoportok:**
A piacon nem találtam túl sok olyan alkalmazást, amelyek parkolóhely foglalással foglalkoznának, így ezt igyekszem úgy fejleszteni, hogy univerzálisan működhessen hasonló elvárásokkal bíró cégeknek, így a softwarenek talán lehetséges valamilyen piaci érték is.

**Motiváció**: A motivációm elsősorban új rendszerek felépítésében van, a tervezéstől kezdve, a teljes projekt átadásig. Másodlagosan mindenképpen meg kell jegyeznem, hogy az applikációt kérésre/üzleti okokból mindenképpen el fogom készíteni, s így egyszerre dolgozhatok, a szakdolgozatomon és munkát is elláthatok.
Továbbá, a komplexitása jóval kisebb, mint egy másik projekt, amin dolgozom, de sok funkcionalitás implementálására ott is szükség lesz, így tanulásnak is nagyon előnyösnek gondolom.

**6. Hogyan néz ki angolul (abstract) a magyar kivonat?**

**Goals:** The goal of this application is to create a simplistic way for the customers (end user from this point) of the client company (client from this point) to make appointments and book the desired services.

**Practicalities:** The application allows for imperative appointment booking, price calculation, and ordering of services. It also grants an opportunity to immediately pay for the booking using a credit/debit card (for security reasons, this utilizes an outside service built for this purpose, the application will only be notified if it was successful or not). The web service also provides accounts for the employees of the client (after authentication and authorization) to overview and manage on-site changes to the bookings, and of course an audit of the end users’ flights. This allows employees to know exactly when to get to the airport to pick them up (This obviously cuts costs, as the car can drive through the airport section for free, however yet again because of complexity this is going to be an outside service, for example flightradar24 API).

**Tasks:**

Overview:

1. Create a database design, and implementation (using an ORM)
2. Backend endpoints/services implementation
3. Authentication and authorization for employees
4. Front-end design
5. UX/UI design implementation
6. Containerisation for simpler deployment
7. Deployment

MVP (Minimum Viable Product):

* Build a home page for marketing reasons „to sell” the „product”.
* Page for making appointment and deciding on services (also first checks availability)
* Managing existing order. Search is based on phone number or licence plate.
* CRUD operations with orders:
	+ Save data to a buffer before writing it to the database, only write it, once the end user approves it from their email. Since end users are not authenticated, I must implement 2 safety features. On one hand never allow just writing data to the database without approval, and on the other hand never display sensitive end user data, because “security through obscurity” is not good enough.
	+ Prepaid orders modifiable? How could the end user be charged or refunded if they want to change after they already paid for a given service?
* RoBA (Role Based Access) for employees. Not everyone has to see every information available, RoBA allows for displaying data, based on scope of activities.

**Target audience:**

I haven’t found too many applications like this available on the market, so I’ll try developing it in a way, that it is reusable for any similar companies, which could maybe allow for reselling of the software, under the correct licence.

**Motivation:**First, my motivation is building new, non-existing system from the ground up, up until the point of delivery. Secondly, I must say, that the application has been ordered by a client, so it will be built for business purposes anyways. This way I can work on my thesis along with a real-world project. Furthermore, the complexity of this application is much less than the current system I am working on, but lots of functionality, must be implemented on that side as well, so it is a very good case study for those components.