



**FAKULTA  
INFORMAČNÍCH  
TECHNOLOGIÍ  
ČVUT V PRAZE**

# **Webový systém pro pacienty s Covid**

## **Návrh architektury a databázový model**

Dokument vytvořen pro potřeby předmětu BI-SI1

Autoři:

Anna Kapitánová, Jan Šmidrkal, Matej Šutý, Šimon Minárik, Anatolii Filkin



## Obsah

1. Návrhový model tried	4
1.1 presentation_layer	6
1.1.1 forms	6
1.1.1.1 Class LoginForm	6
1.1.1.2 Class PatientRegisterForm	6
1.1.1.3 Class ReservationForm	6
1.1.2 templates	7
1.1.2.1 PrimitiveType base	7
1.1.2.2 PrimitiveType doctor	7
1.1.2.3 PrimitiveType doctor_login	7
1.1.2.4 PrimitiveType doctor_register_patient	8
1.1.2.5 PrimitiveType index	8
1.1.2.6 PrimitiveType patient	8
1.1.2.7 PrimitiveType patient_login	8
1.1.2.8 PrimitiveType register_patient	8
1.1.2.9 PrimitiveType reservation	8
1.1.3 views	8
1.1.3.1 Class DoctorLoginView	9
1.1.3.2 Class DoctorRegisterPatientView	10
1.1.3.3 Class DoctorView	10
1.1.3.4 Class IndexView	10
1.1.3.5 Class LogoutView	10
1.1.3.6 Class PatientLoginView	10
1.1.3.7 Class PatientView	11
1.1.3.8 Class RegisterPatientView	11
1.1.3.9 Class ReservationView	11
1.2 business_layer	12
1.2.1 handlers	12
1.2.1.1 interfaces	12
1.2.1.1.1 Class HandlerFactory	12
1.2.1.1.2 Interface UserHandlerInterface	13
1.2.1.1.3 Interface PatientHandlerInterface	13
1.2.1.1.4 Interface CovidpassHandlerInterface	14
1.2.1.1.5 Interface ReservationHandlerInterface	14
1.3 data_layer	14
1.3.1 data_access	14
1.3.1.1 interfaces	15
1.3.1.1.1 Class DataAccessFactory	15
1.3.2 models	16
2. Návrh architektúry	16
2.1 presentation_layer	18
2.1.1 forms	18
2.1.2 templates	18
2.1.3 views	18
2.2 business_layer	18
2.2.1 handlers	18
2.3 data_layer	18
2.3.1 data_access	18
2.3.2 models	18

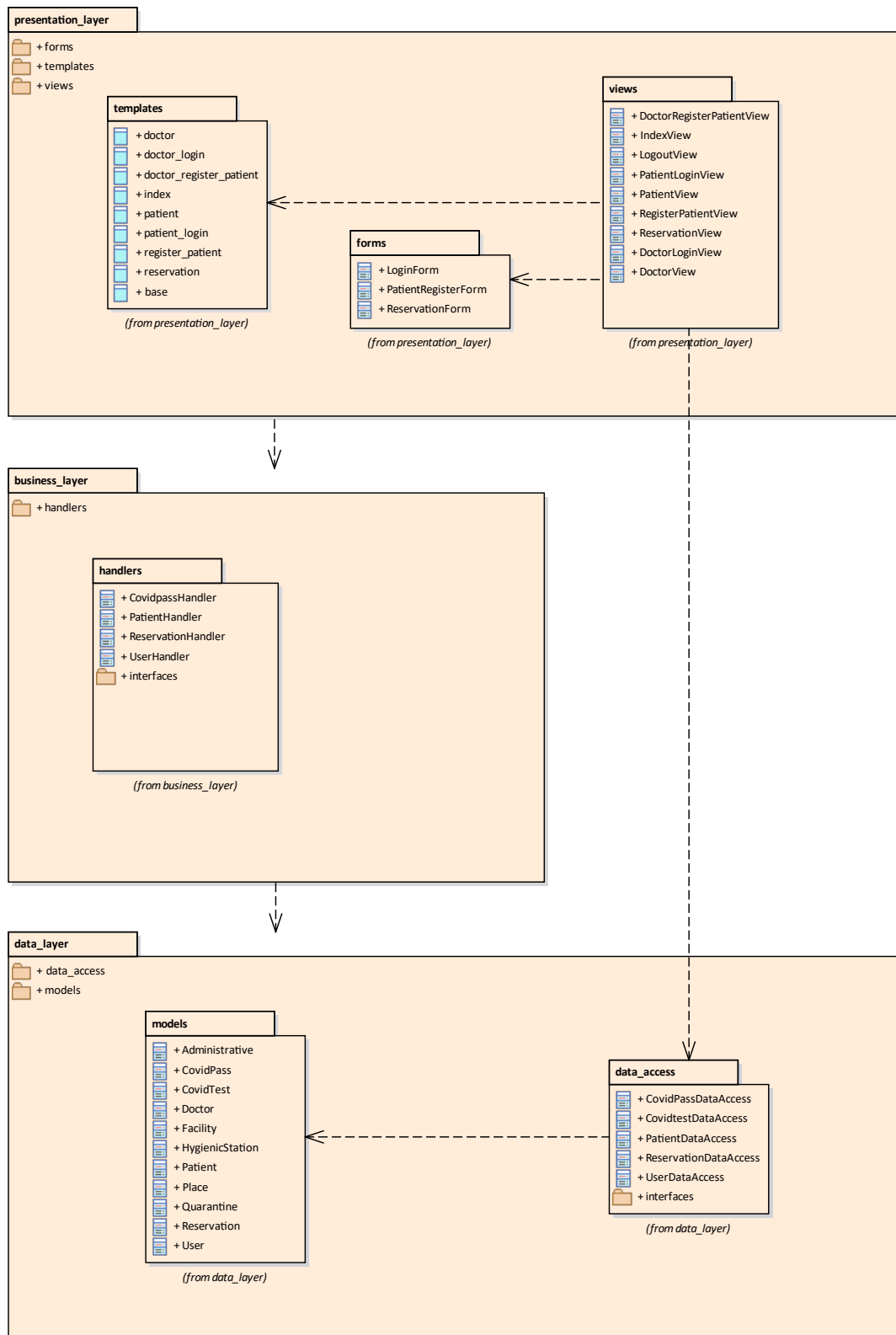


3.	DDL	19
3.1	Covidpass « <i>table</i> »	19
3.2	CovidTest « <i>table</i> »	20
3.3	Facility « <i>table</i> »	20
3.4	HygienicStation « <i>table</i> »	20
3.5	Patient « <i>table</i> »	20
3.6	Place « <i>table</i> »	20
3.7	Quarantine « <i>table</i> »	21
3.8	Reservation « <i>table</i> »	21



## **1. Návrhový model tried**

Kapitola obsahuje pohled na systém. Obsahuje detailní popis jednotlivých tříd, rozhraní a balíčků, které realizují tento systém. Vnitřní struktura odpovídá rozdělení balíčků a složek v python projektu.



Obrázek 1 - Návrhový model tried

Diagram zobrazuje trojvrstvovú architektúru - relaxovanú, zachytuje aj štruktúru balíčkov, rozdelenie a ich vzájomné závislosti.

## 1.1 presentation\_layer

Táto vrstva zobrazuje užívateľovi aplikáciu. Užívateľovi sa budú zobrazovať dáta a môže predávať údaje do business vrstvy pomocou tried v tejto vrstve.

### 1.1.1 forms

Obsahuje objekty formulárov na získanie údajov.

LoginForm	ReservationForm	PatientRegisterForm
- username: CharField - password: CharField	- username: CharField - password: CharField - name: CharField - surname: CharField - birthid: CharField - email: CharField - phone: CharField - address: CharField - covidTest: CharField	deadline: DateTimeField hygienicStation: CharField
+ LoginForm(HttpRequest): form	+ ReservationForm(HttpRequest): form	PatientRegisterForm(HttpRequest): form

Obrázek 2 - forms

#### 1.1.1.1 Class LoginForm

Formulár, ktorý sa zobrazuje návštevníkom webu pri prihlásení.

Název atributu	Datový typ	Popis
username	CharField	
password	CharField	
Název metody	Návratový typ	Popis
LoginForm	form	Parametry: <b>request: HttpRequest</b> -

#### 1.1.1.2 Class PatientRegisterForm

Formulár, ktorý poskytuje definované polia na vyplnenie údajov pacienta. Formulár overuje správny formát niektorých polí.

Název atributu	Datový typ	Popis
deadline	DateTimeField	
hygienicStation	CharField	
Název metody	Návratový typ	Popis
PatientRegisterForm	form	Parametry: <b>request: HttpRequest</b> -

#### 1.1.1.3 Class ReservationForm

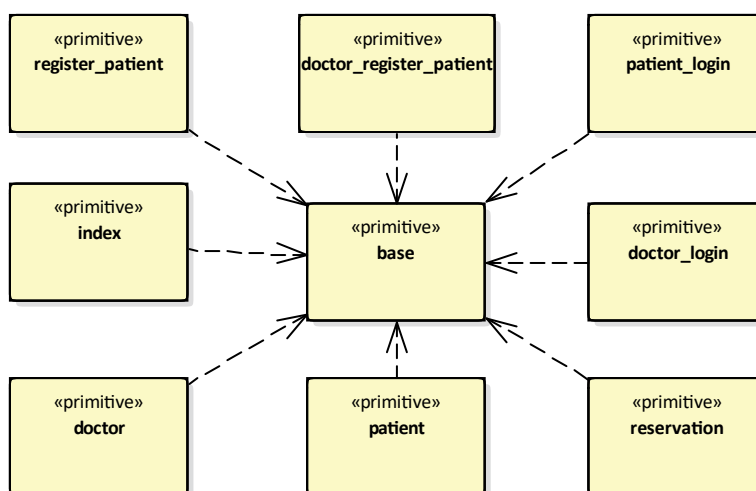


Formulár pre pacienta na výber testovania v určitom čase a na určitom mieste.

Název atributu	Datový typ	Popis
username	CharField	
password	CharField	
name	CharField	
surname	CharField	
birthid	CharField	
email	CharField	
phone	CharField	
address	CharField	
covidTest	CharField	
Název metody	Návratový typ	Popis
ReservationForm	form	Parametry: <b>request: HttpRequest</b> -

### 1.1.2 templates

Balíček obsahuje rôzne šablóny na vzhľad a obsah webovej stránky. Dáta, ktoré sa prezentujú budú nezávislé od formy webstránky.



Obrázek 3 - templates

#### 1.1.2.1 PrimitiveType base

HTML body template - Služi ako základna stránka na rozširovanie pomocou html.

#### 1.1.2.2 PrimitiveType doctor

HTML body template - zobrazuje stránku pre prihláseného doktora a možnosti aktivít.

#### 1.1.2.3 PrimitiveType doctor\_login

HTML body template - zobrazuje stránku pre doktora na prihlásenie.



#### **1.1.2.4 PrimitiveType doctor\_register\_patient**

HTML body template - stránka pre doktora na registrovanie pacienta, obsahuje formular na vytvorenie covidpassu a uzivatela pre pacienta.

#### **1.1.2.5 PrimitiveType index**

HTML body template - uvodna stranka.

#### **1.1.2.6 PrimitiveType patient**

HTML body template - stranka pre pacienta, zobrazuje moznosti aktivit a informacie.

#### **1.1.2.7 PrimitiveType patient\_login**

HTML body template - stranka pre pacienta, umoznuje prihlasit sa jeho prihlasovacimi udajmi.

#### **1.1.2.8 PrimitiveType register\_patient**

HTML body template - stranka pre samoplatcu pacienta.

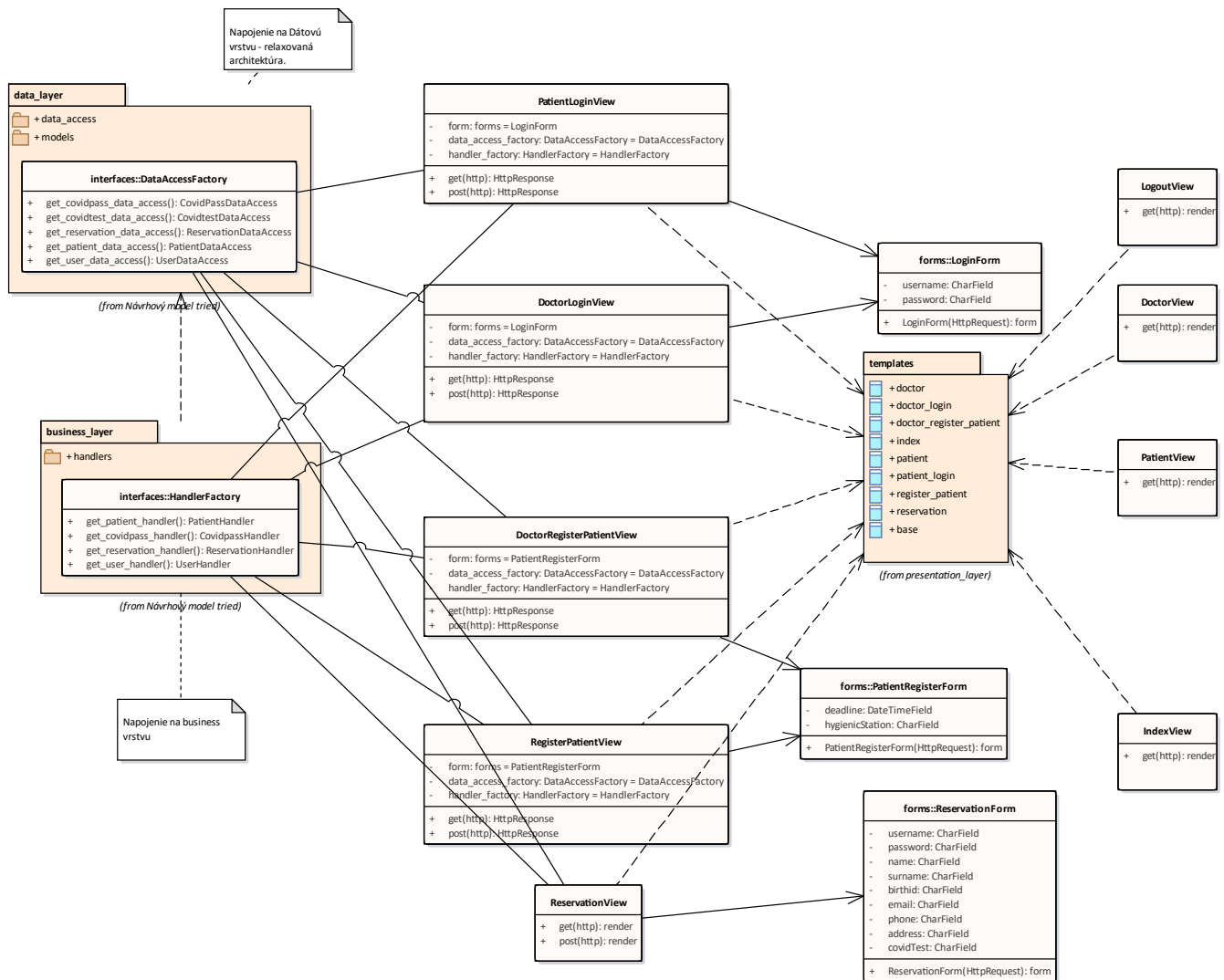
#### **1.1.2.9 PrimitiveType reservation**

HTML body template - zobrazi pacientovi moznosti rezervacie testovania a vytvori rezervaciu

### **1.1.3 views**

Balíček bude obsahovať všetky triedy, ktoré prezenetujú užívateľovi obsah v podobe formulárov alebo stránky. Balíček bude tiež obsahovať texty, ktoré popisujú jednotlivé kroky a inštrukcie pre pacienta.





Obrázek 4 - views

### 1.1.3.1 Class DoctorLoginView

Trieda, ktorá sa stará o prihlasovanie doktorov a zobrazenie stránky.

Název atributu	Datový typ	Popis
form	forms	
data_access_factory	DataAccessFactory	
handler_factory	HandlerFactory	
Název metody	Návratový typ	Popis
get	HttpResponse	Parametry: <b>request: http -</b>
post	HttpResponse	



Název atributu	Datový typ	Popis
		Parametry: <b>request: http -</b>

### 1.1.3.2 Class DoctorRegisterPatientView

Trieda, ktorá sa stará o vytvorenie covidpassu doktorom, priradenie pacientovi a zobrazenie stránky.

Název atributu	Datový typ	Popis
form	forms	
data access factory	DataAccessFactory	
handler factory	HandlerFactory	
Název metody	Návratový typ	Popis
get	HttpResponse	Parametry: <b>request: http -</b>
post	HttpResponse	Parametry: <b>request: http -</b>

### 1.1.3.3 Class DoctorView

Zobrazuje doktorovi stranku s moznostami - napr. registrovat pacienta.

Název metody	Návratový typ	Popis
get	render	Parametry: <b>request: http -</b>

### 1.1.3.4 Class IndexView

Trieda, ktorá zobrazí úvodnú stránku a poskytne možnosť na prihlásenie.

Název metody	Návratový typ	Popis
get	render	Parametry: <b>request: http -</b>

### 1.1.3.5 Class LogoutView

Po odhlásení vráti návštevníka na domovskú stránku

Název metody	Návratový typ	Popis
get	render	Parametry: <b>request: http -</b>

### 1.1.3.6 Class PatientLoginView

Trieda, ktorá sa stará o prihlasovanie pacientov a zobrazenie stránky.

Název atributu	Datový typ	Popis
----------------	------------	-------



Název atributu	Datový typ	Popis
form	forms	
data_access_factory	DataAccessFactory	
handler_factory	HandlerFactory	
Název metody	Návratový typ	Popis
get	HttpResponse	Parametry: <b>request: http -</b>
post	HttpResponse	Parametry: <b>request: http -</b>

### 1.1.3.7 Class PatientView

Informačná stránka a rozcestník pre pacienta

Název metody	Návratový typ	Popis
get	render	Parametry: <b>request: http -</b>

### 1.1.3.8 Class RegisterPatientView

Trieda, ktorá sa stará o vytvorenie covidpassu, priradenie pacientovi a zobrazenie stránky.

Název atributu	Datový typ	Popis
form	forms	
data_access_factory	DataAccessFactory	
handler_factory	HandlerFactory	
Název metody	Návratový typ	Popis
get	HttpResponse	Parametry: <b>request: http -</b>
post	HttpResponse	Parametry: <b>request: http -</b>

### 1.1.3.9 Class ReservationView

Zobrazí stránku s možnosťou vytvoriť rezerváciu testovania

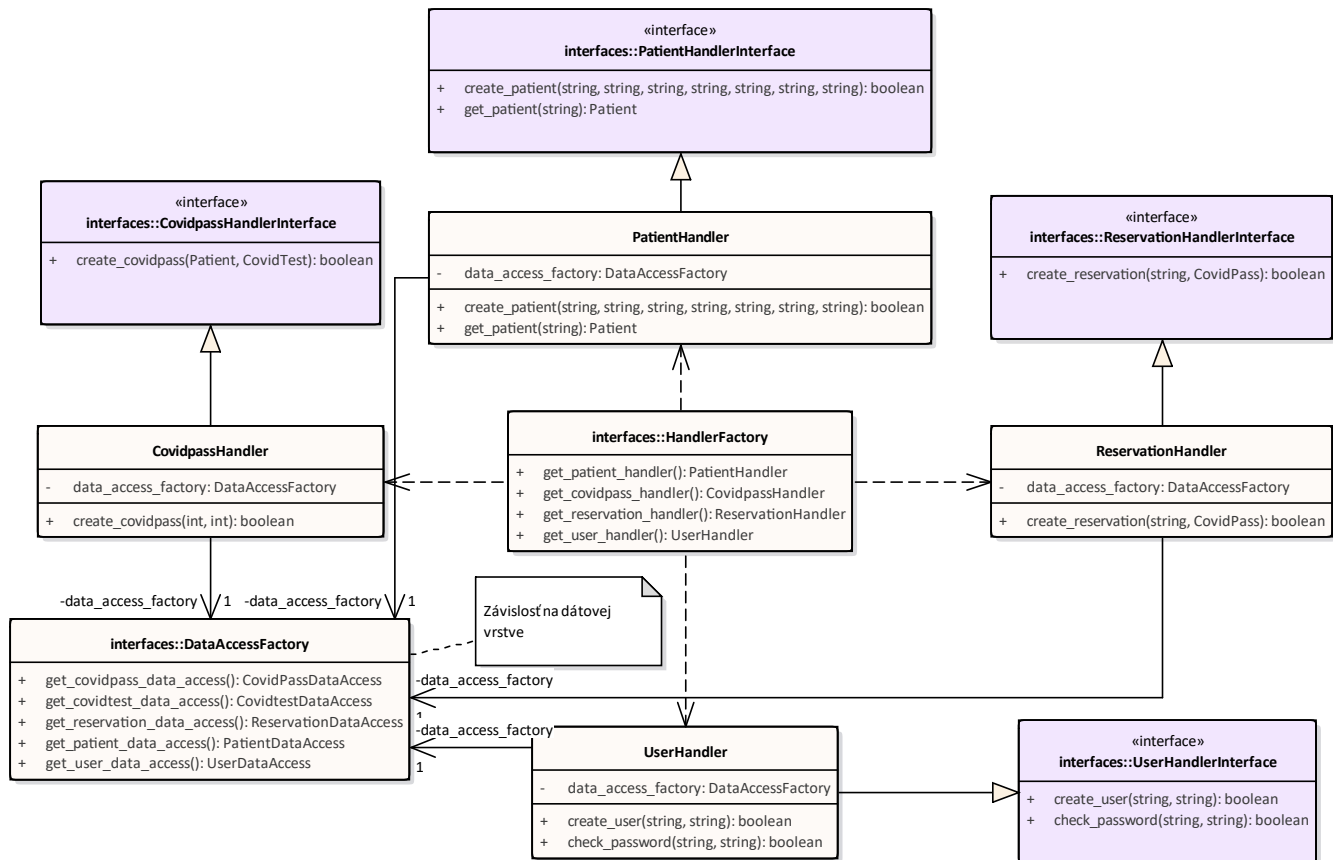
Název metody	Návratový typ	Popis
get	render	Parametry: <b>request: http -</b>
post	render	Parametry: <b>request: http -</b>

## 1.2 business\_layer

Obsahuje všetky triedy a rozhrania, ktoré slúžia na business logiku. Obsahuje triedy pre komunikáciu s prezentačnou vrstvou a pre prácu s dátovou vrstvou.

### 1.2.1 handlers

Balíček obsahuje rozhrania handlerov a triedy, ktoré ich implementujú. Slúžia na komunikáciu s dátovou vrstvou.



Obrázek 5 - handlers

Rozhranie, ktoré poskytuje vytvorenie CovidPassu. Spoločne s dátovou vrstvou perzistentne uloží dáta.

#### 1.2.1.1 interfaces

Balíček obsahuje rozhrania, ktoré sú nezávislé od implementácie - poskytujú metódy a členské premenné.

##### 1.2.1.1.1 Class HandlerFactory

Objekt, ktorý obsahuje definíciu ktoré triedy budú slúžiť na implementáciu rozhraní. HandlerFactory vytvorí objekty konkrétnych Handlerov a poskytne ich cez metódy.

Název metódy	Návratový typ	Popis
get_patient_handler	PatientHandler	



Název metody	Návratový typ	Popis
get_covidpass_handler	CovidpassHandler	
get_reservation_handler	ReservationHandler	
get_user_handler	UserHandler	

#### 1.2.1.1.2 Interface *UserHandlerInterface*

Rozhranie, ktoré poskytuje vytvorenie užívateľov a ich overovanie pri prihlasovaní. Spoločne s dátovou vrstvou perzistentne uloží dáta.

Název metody	Návratový typ	Popis
create_user	boolean	Vytvorí užívateľa s heslom. Parametry: <b>username: string</b> - Parametry: <b>password: string</b> -
check_password	boolean	Overí prihlásenie užívateľa. Parametry: <b>username: string</b> - Parametry: <b>password: string</b> -

#### 1.2.1.1.3 Interface *PatientHandlerInterface*

Rozhranie, ktoré poskytuje správu vytvárania pacientov. Spoločne s dátovou vrstvou perzistentne uloží dáta.

Název metody	Návratový typ	Popis
create_patient	boolean	Vytvorí pacienta na základe poskytnutých údajov cez DataAccess interface. Parametry: <b>name: string</b> - Parametry: <b>surname: string</b> - Parametry: <b>birthid: string</b> - Parametry: <b>email: string</b> - Parametry: <b>phone: string</b> - Parametry: <b>address: string</b> - Parametry: <b>user: string</b> -
get_patient	Patient	Získa objekt pacienta z perzistentného úložiska podľa parametru user. Parametry: <b>user: string</b> -



#### 1.2.1.1.4 Interface CovidpassHandlerInterface

Spoločne s dátovou vrstvou perzistentne uloží dáta.

Název metody	Návratový typ	Popis
create_covidpass	boolean	Vytvorí CovidPass pomocou Data Access rozhrania a priradí ho k pacientovi. Parametry: <b>patient: Patient</b> - Parametry: <b>covidtest: CovidTest</b> -

#### 1.2.1.1.5 Interface ReservationHandlerInterface

Rozhranie, ktoré poskytuje vytvorenie rezervácie. Spoločne s dátovou vrstvou perzistentne uloží dáta.

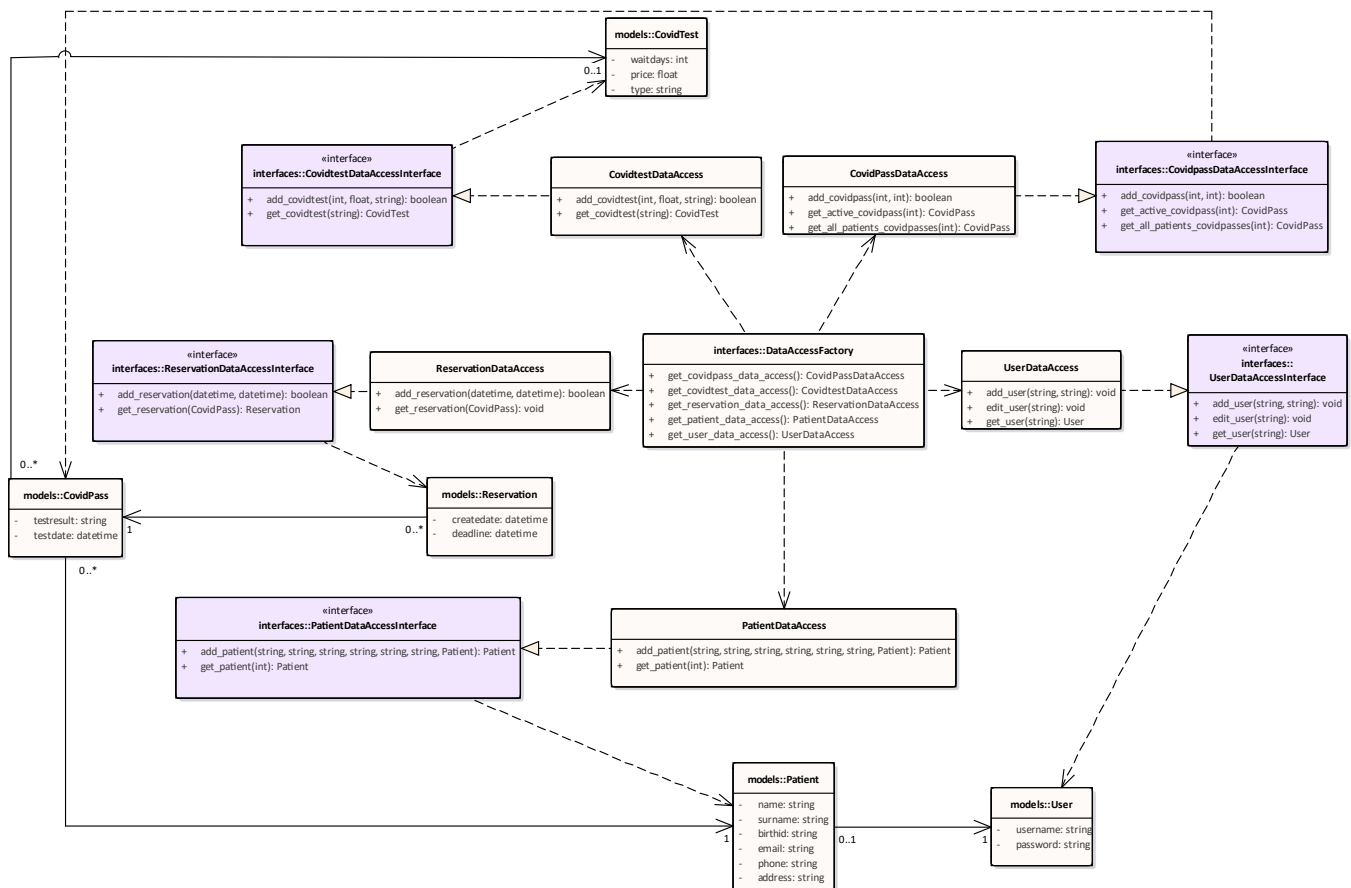
Název metody	Návratový typ	Popis
create_reservation	boolean	Vytvorí rezerváciu pomocou Data Access rozhrania Parametry: <b>deadline: string</b> - Parametry: <b>covidpass: CovidPass</b> -

### 1.3 data\_layer

Obsahuje triedy a rozhrania pre prácu s dátami - ukládať, mazať, meniť z dátového úložiska (databáze SQLite).

#### 1.3.1 data\_access

Triedy v tomto balíčku predstavujú interface objektov prístupu k entitám (DAO) a ich implementácie. Sú to predovšetkým abstrakcie nad tabuľkami, ktoré umožňujú prácu s dátami.



Obrázek 6 - data\_access

### 1.3.1.1 interfaces

Interfaces slúžia na komunikáciu a sprostredkovanie metód business vrstvy.

#### 1.3.1.1.1 Class DataAccessFactory

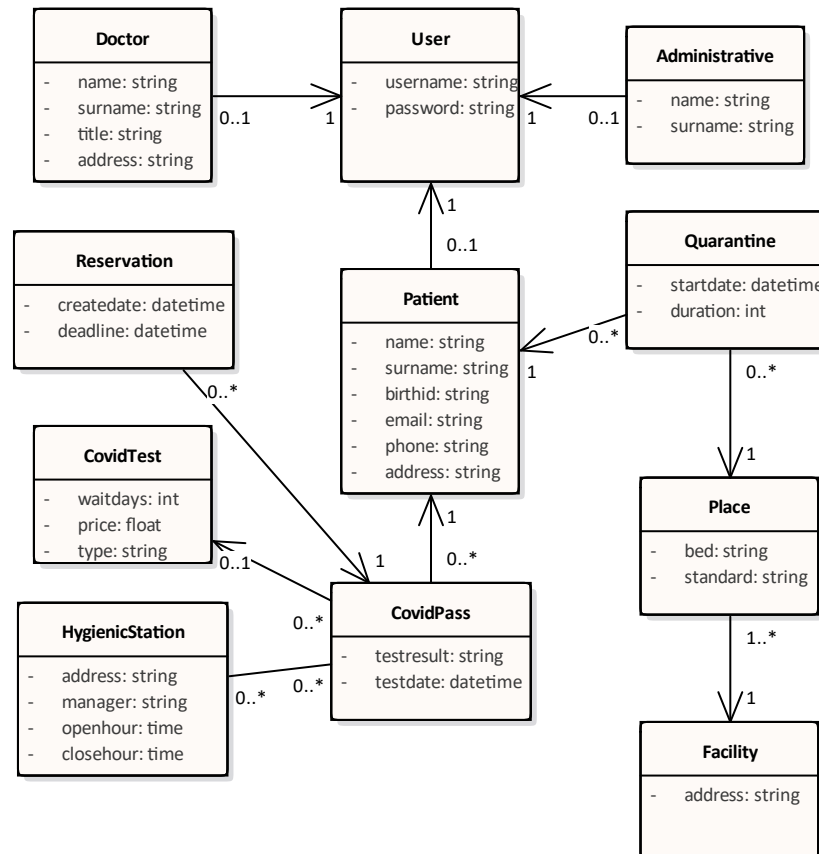
Objekt, ktorý obsahuje definíciu ktoré triedy budú slúžiť na implementáciu rozhraní. DataAccessFactory vytvorí objekty Data Access a poskytne ich cez metódy.

Název metódy	Návratový typ	Popis
get_covidpass_data_access	CovidPassDataAccess	
get_covidtest_data_access	CovidtestDataAccess	
get_reservation_data_access	ReservationDataAccess	
get_patient_data_access	PatientDataAccess	
get_user_data_access	UserDataAccess	



### 1.3.2 models

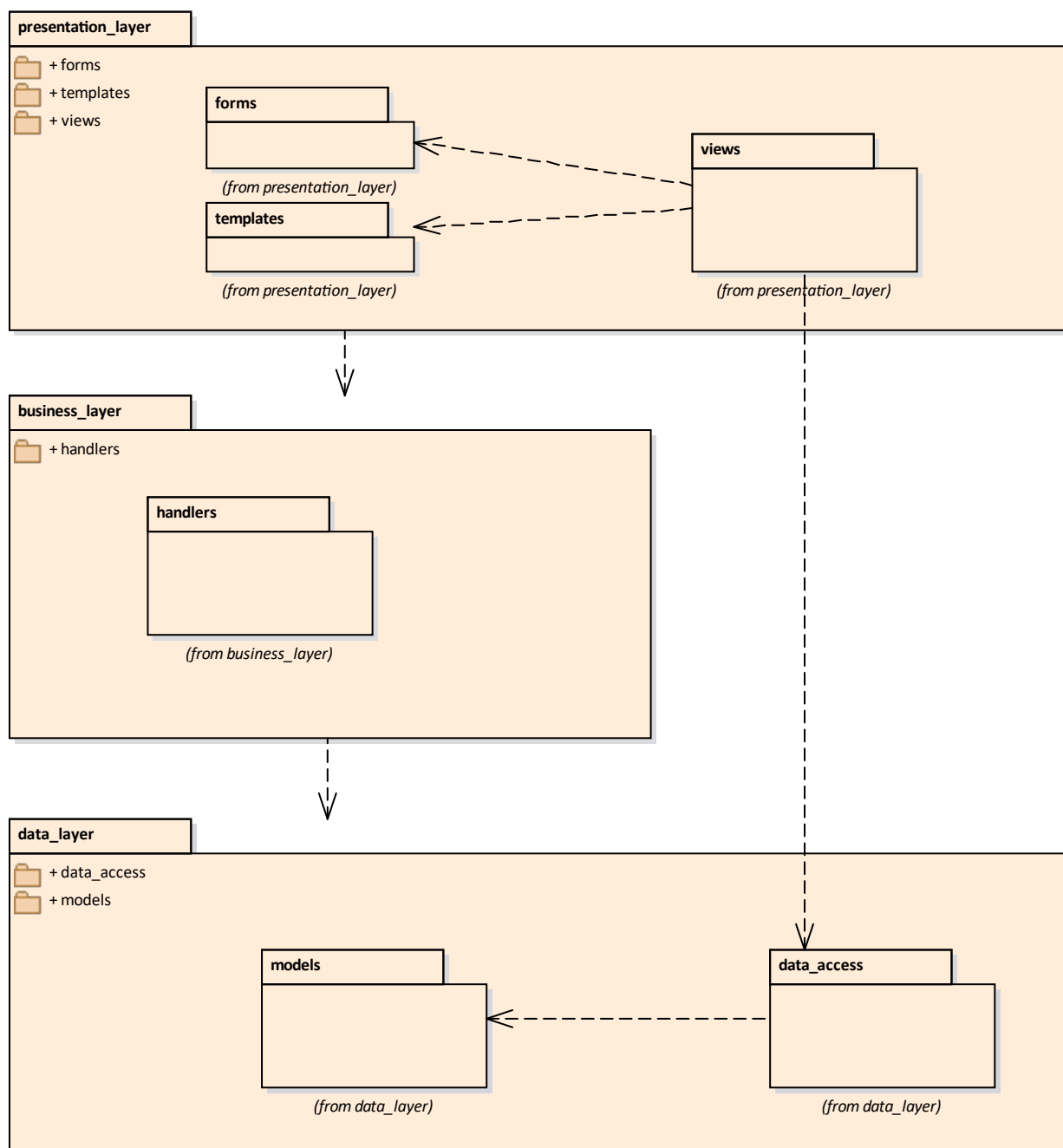
Balíček obsahuje entity - modely z databáze.



Obrázek 7 - models

## 2. Návrh architektúry





Obrázek 8 - Návrh architektúry

V tomto projekte sme použili trojvrstvú relaxovanú architektúru aká je zobrazená na schéme. Jednotlivé vrstvy sú oddelené pomocou rozhraní, ktoré umožňujú komunikáciu medzi vrstvami. Implementácia rozhraní sa môže jednoducho meniť.

Implementácia aplikácie je postavená na frameworku Django, ktorý vychádza z programovacieho jazyku Python a umožňuje ľahkú a flexibilnú spoluprácu v zobrazovaní obsahu a ukladaní/prácu s datami v databáze SQLite.

Použité nástroje, frameworks, balíčky:



1. Django - web framework, uživatelské rozhranie - <https://www.djangoproject.com/>
2. SQLite - ORM databáza - <https://www.sqlite.org/index.html>
3. Free html templates - dizajn stránok
4. Správa verzií - Gitlab - <https://about.gitlab.com/>
5. Komunikácia - MS Teams - <https://www.microsoft.com/en-us/microsoft-365/microsoft-teams/group-chat-software>

## **2.1 presentation\_layer**

Táto vrstva zobrazuje užívateľovi aplikáciu. Užívateľovi sa budú zobrazovať dáta a môže predávať údaje do business vrstvy pomocou tried v tejto vrstve.

### **2.1.1 forms**

### **2.1.2 templates**

Balíček obsahuje rôzne šablóny na vzhľad a obsah webovej stránky. Dáta, ktoré sa prezentujú budú nezávislé od formy webstránky.

### **2.1.3 views**

Balíček bude obsahovať všetky triedy, ktoré prezenetujú užívateľovi obsah v podobe formulárov alebo stránky. Balíček bude tiež obsahovať texty, ktoré popisujú jednotlivé kroky a inštrukcie pre pacienta.

## **2.2 business\_layer**

Obsahuje všetky triedy a rozhrania, ktoré slúžia na business logiku. Obsahuje triedy pre komunikáciu s prezentačnou vrstvou a pre prácu s dátovou vrstvou.

### **2.2.1 handlers**

Balíček obsahuje triedy, ktoré implementujú rozhrania na spostredkovanie business logiky z dátovej vrstvy.

## **2.3 data\_layer**

Obsahuje triedy a rozhrania pre prácu s dátami - ukládať, mazať, meniť z dátového úložiska (databáze SQLite).

### **2.3.1 data\_access**

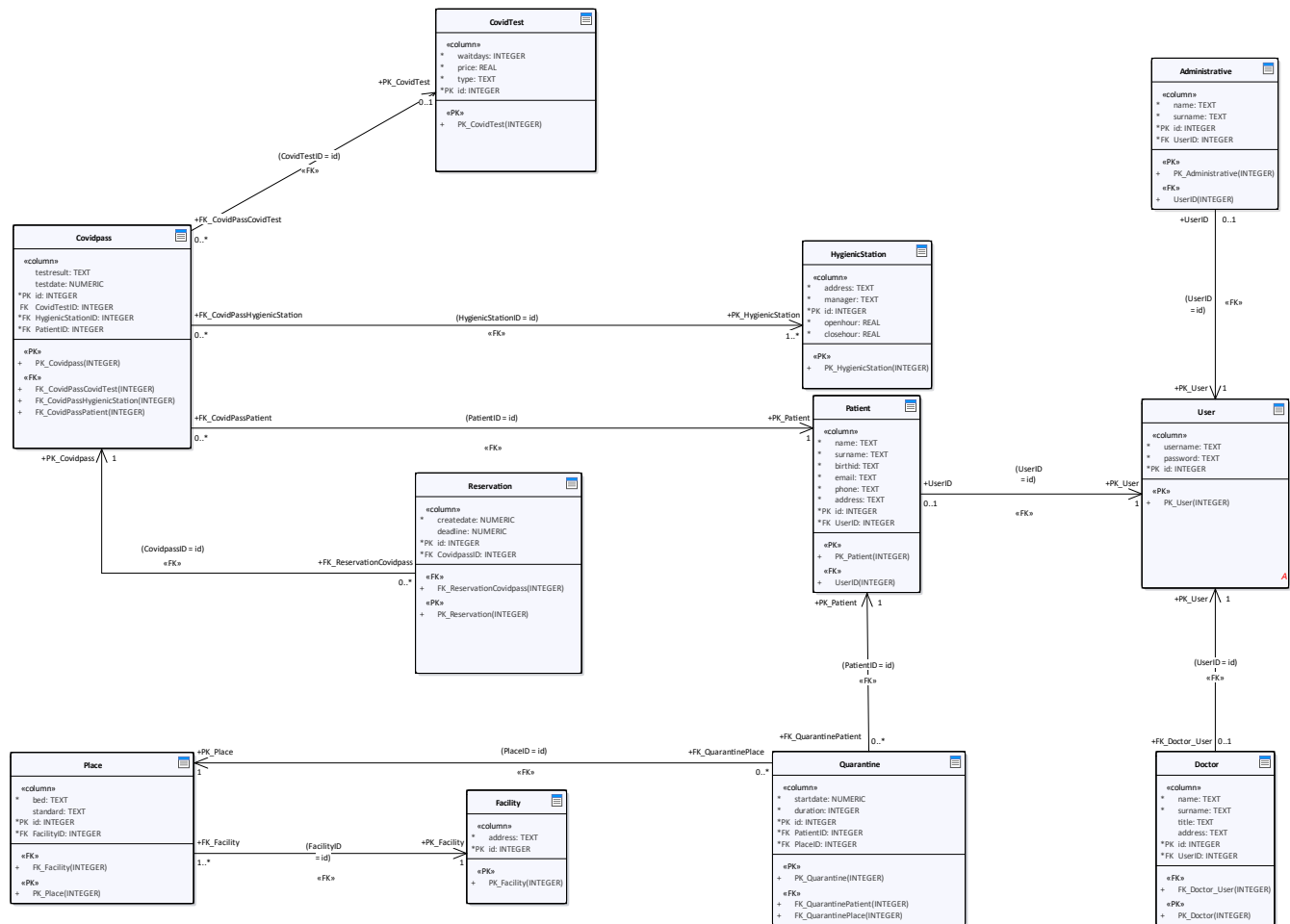
Triedy v tomto balíčku predstavujú interface objektov prístupu k entitám (DAO) a ich implementácie. Sú to predovšetkým abstrakcie nad tabuľkami, ktoré umožňujú prácu s dátami.

### **2.3.2 models**

Balíček obsahuje entity - modely z databáze.



### 3. DDL



Obrázek 9 - DDL

#### 3.1 Covidpass «table»

Dokument, který pacient získá od doktora. Aby mohl pacient vytvořit rezervaci, potřebuje se prověřit CovidPassem. Hygienická stanice potom provede test pouze těm pacientům, kteří se prokážou platným CovidPassem.

Název atributu	Datový typ	Not null	Popis
testresult	TEXT	False	výsledek testu
testdate	NUMERIC	False	dátum uskutočnění testu
id	INTEGER	True	
CovidTestId	INTEGER	False	
HygienicStationID	INTEGER	True	
PatientID	INTEGER	True	



### 3.2 CovidTest «table»

Test, který je proveden pro zjištění koronaviru.

Název atributu	Datový typ	Not null	Popis
waitdays	INTEGER	True	Doba trvania vyhodnotenia testu v dnoch.
price	REAL	True	Cena testu, ktorá sa účtuje pacientovi
type	TEXT	True	typ testu
id	INTEGER	True	

### 3.3 Facility «table»

Zařízení, které má víc míst na ubytování pro pacienty

Název atributu	Datový typ	Not null	Popis
address	TEXT	True	adresa zariadenia
id	INTEGER	True	

### 3.4 HygienicStation «table»

Místo kde je proveden test na koronavirus.

Název atributu	Datový typ	Not null	Popis
address	TEXT	True	adresa zariadenia
manager	TEXT	True	Meno správcu, hlavného zamestnanca zariadenia.
id	INTEGER	True	
openhour	REAL	True	Otvárací čas, slúži na definovanie termínov na testovanie.
closehour	REAL	True	Zatvárací čas, slúži na definovanie termínov na testovanie.

### 3.5 Patient «table»

Osoba, která se chce nechat otestovat na koronavirus.

Název atributu	Datový typ	Not null	Popis
name	TEXT	True	
surname	TEXT	True	
birthid	TEXT	True	Rodné číslo je desetimístné číslo, ktoré je deliteľné jedenácti bez zvyšku. První dvojčíslí vyjadruje posledný dvé číslice roku narodenia, druhé dvojčíslí vyjadruje mesiac narodenia, u žien zvýšené o 50, tretí dvojčíslí vyjadruje deň narodenia. Čtyřmístná koncovka je rozlišujícím znakem fyzických osob narodených v tomtež kalendářním dnu o rozsahu přidělovaných koncovek 0000 až 9999
email	TEXT	True	email pacienta alebo inej kontaktnej osoby
phone	TEXT	True	Telefónne číslo pacienta alebo inej kontaktnej osoby
address	TEXT	True	Adresa trvalého bydliska.
id	INTEGER	True	unikátny identifikátor, každá tabuľka má tiež svoje. Slúži na spájanie cez FK.
UserID	INTEGER	True	

### 3.6 Place «table»

Místo v karanténě v konkrétním ubytovacím zařízení. Každé lůžko patří do pokoje - kódování. Různé pokoje mají různé standardy - sociální zařízení, balkon, společná koupelna...

Název atributu	Datový typ	Not null	Popis
bed	TEXT	True	Vlastný identifikátor miesta na ubytovanie v zariadení.



Název atributu	Datový typ	Not null	Popis
standard	TEXT	False	Popis úrovně ubytovania.
id	INTEGER	True	
FacilityID	INTEGER	True	

### 3.7 Quarantine «table»

Trvá istý čas - dĺžka karantény - a je jednoznačne identifikovaná svým pacientem a miestem, kde probíhá.

Název atributu	Datový typ	Not null	Popis
startdate	NUMERIC	True	Predpokladaný začiatok ubytovania/karantény.
duration	INTEGER	True	Dĺžka ubytovania, karantény.
id	INTEGER	True	
PatientID	INTEGER	True	
PlaceID	INTEGER	True	

### 3.8 Reservation «table»

Rezervace, kterou pacient musí mít, než dorazí na hygienickou stanici.

Název atributu	Datový typ	Not null	Popis
createdate	NUMERIC	True	Dátum vytvorenia rezervácie
deadline	NUMERIC	False	Termín testovania.
id	INTEGER	True	
CovidpassID	INTEGER	True	



**FAKULTA  
INFORMAČNÍCH  
TECHNOLOGIÍ  
ČVUT V PRAZE**