

**Slovenská technická univerzita v Bratislave**  
**Fakulta informatiky a informačných technológií**

**MEDIS**

**Medical information system**

**Zdravotný informačný systém**

**Predmet:** Vývoj aplikácií s viacvrstvovou architektúrou

**Prednášajúci:** Mgr. Ing. Miroslav Reiter, MBA

**Cvičiaci:** Mgr. Ing. Miroslav Reiter, MBA

**Rok:** 2021/22

**Tím:** MEDIS

**Vedúci tímu:** Andrej Gažík

**Členovia tímu:** Simon Kokavec, Tamás Szakál, Zsolt Kiss, Martin Bopkó

# Obsah

1.	Zámer	3
1.1.	Kto bude aplikáciu používať?	3
1.2.	Čo to je za aplikáciu?	3
1.3.	Kde ju budú používať?	3
1.4.	Kedy ju budú používať?	3
1.5.	Prečo ju používať?	3
1.6.	Koľko bude aplikácia stáť?	4
1.7.	Ako sa bude používať aplikácia?	4
2.	Používatelia a opis prípadov použitia	4
2.1.	Admin	4
2.2.	Doctor	5
2.3.	Nurse	6
2.4.	Receptionist	6
3.	Zoznam práv používateľov	8

## 1. Zámer

### 1.1. Kto bude aplikáciu používať?

Aplikáciu budú používať štyri typy používateľov: Administrátor systému, Lekári, Sestry a recepcný pracovníci. Všetky typy používateľov majú v aplikácii rôzne povolenia.

**Administrátor systému** - úplný CRUD prístup do systému a prehľad všetkých akcií vykonaných ostatnými používateľmi.

**Lekár** - možnosť vypísať liek, zapísať nový zápis o pacientovi, zmeniť návštevy

**Sestra** - možnosť pozrieť údaje o pacientovi, náhľad do zoznamu liekov, zmeniť návštevy

**Recepcný pracovník** - kontroluje kedy majú pacienti dohodnutý termín

### 1.2. Čo to je za aplikáciu?

Tento systém chceme vytvoriť pre rôzne zdravotné inštitúcie. Môžu ho používať aj bežní lekári ale aj špecialisti (zubári, očný lekár, ortopéd ... atď). Systém by v prvom rade splnil požadované administratívne a informačné úlohy inštitúcie. Tento systém ponúkne zdravotným pracovníkom prehľad o pacientoch, ich zdravotnom stave, vypísaných liekoch, predošlých návštev, budúcich návštev... atď. Jednotliví pracovníci by mali rôzne úrovne prístupu.

### 1.3. Kde ju budú používať?

V rôznych zdravotných inštitúciách, zdravotných pracovísk, lekární. Aplikáciu bude možné používať všade kde je potrebné evidencia pacientov, liekov, návštevy lekára.

### 1.4. Kedy ju budú používať?

V rámci pracovnej doby, individuálne podľa používateľov. V prípade pohotovosti celý deň. Všetko bude záležať od danej inštitúcie.

### 1.5. Prečo ju používať?

Aplikácia bude priehľadná, poskytne moderný prehľad používateľovi o pacientoch a ich liekoch. Urýchli sa tak proces zisťovania a ukladania informácií o pacientoch čo ušetrí čas zdravotnému personálu.

### **1.6. Koľko bude aplikácia stáť?**

Cena aplikácie bude jednorázová - licencia bude stáť 40000€ a ročne 4000€ pre aktualizácie a podporu. Jednorázová licencia platí na jednu pobočku a do 15 pracovníkov.

### **1.7. Ako sa bude používať aplikácia?**

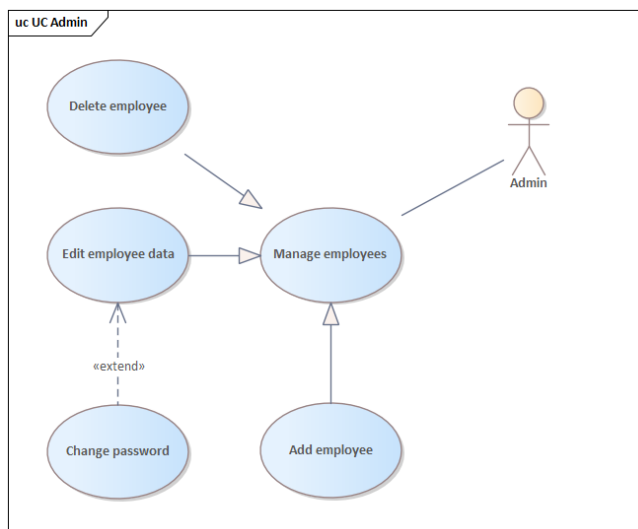
Aplikácia vyzve používateľa, aby sa prihlásil a overil svoju identitu. Po prihlásení sa odomykajú jednotlivé úkony v aplikácii podľa typu používateľského účtu, ako napríklad: kontrola dohodnutých termínov ako recepčná alebo vypísanie lieku ako lekára. Po vykonaní úkonov sa užívateľ môže bezpečne odhlásiť, ak chce a aplikáciu môže používať iný zamestnanec.

Aplikácia na začiatku si bude pýtať overenie/prihlásenie používateľa. Následne aplikácia navrhne používateľovi možnosti ako vyhľadávanie pacienta, vypísanie lieku, pridanie pacienta, pridanie záznamu, zadanie termínu návštevy, pozretie návštev.

## **2. Používatelia a opis prípadov použitia**

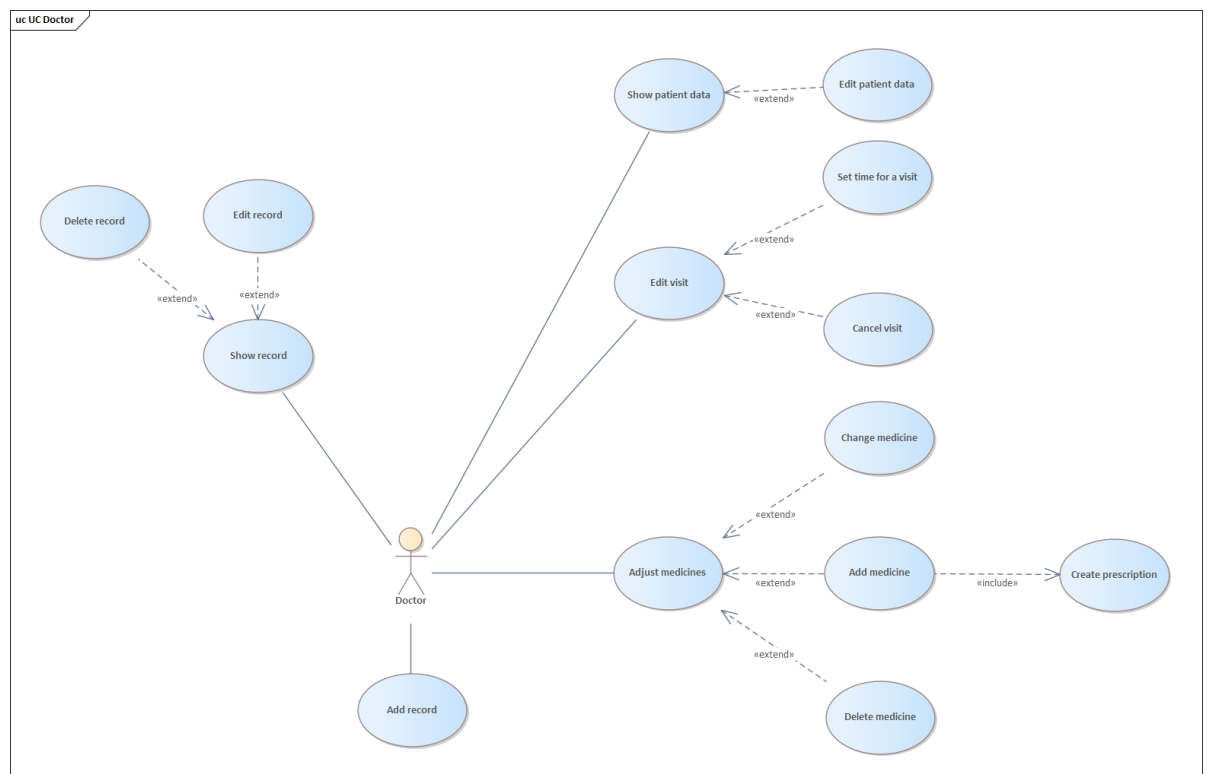
### **2.1. Admin**

Používateľ s najvyšším oprávnením. S oprávnením pozrieť a editovať všetky aspekty systému. Má možnosť pridávať, mazať a editovať všetkých pracovníkov. Jediný prístup ktorý má zamedzený je prístup k všetkým informáciám o pacientoch. Osobné informácie pacientov ako meno, rodné číslo...atď. by boli zašifrované aby bolo dodržané lekárske tajomstvo.



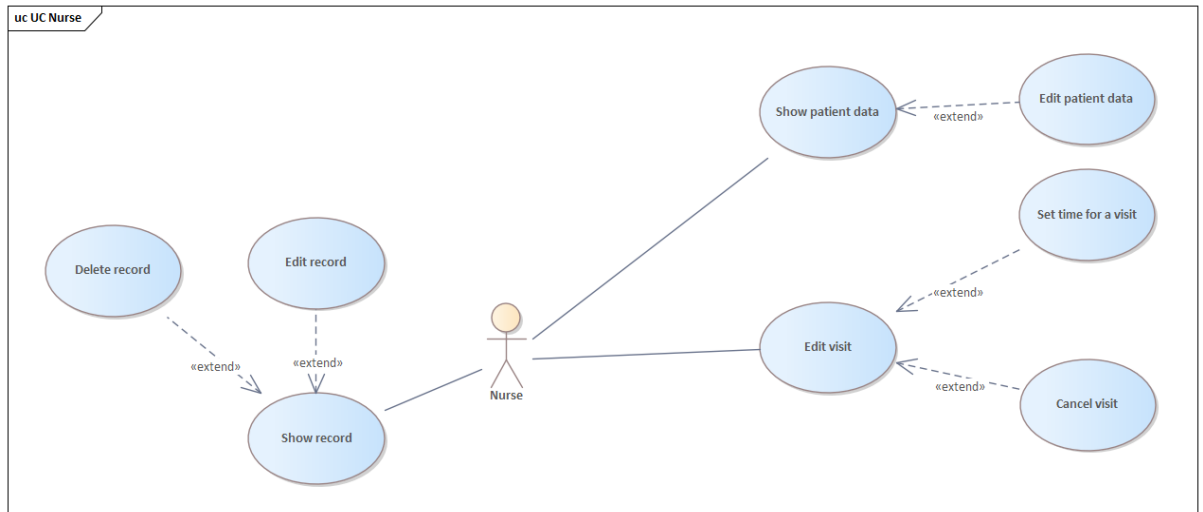
## 2.2. Doctor

Používateľ s druhým najvyšším oprávnením. Má oprávnený prístup k všetkým dátam pacientov - má možnosť vytvárať medicínske záznamy, predpisovať recepty a taktiež budúce návštevy ak chce. Nemá možnosť editovať systém.



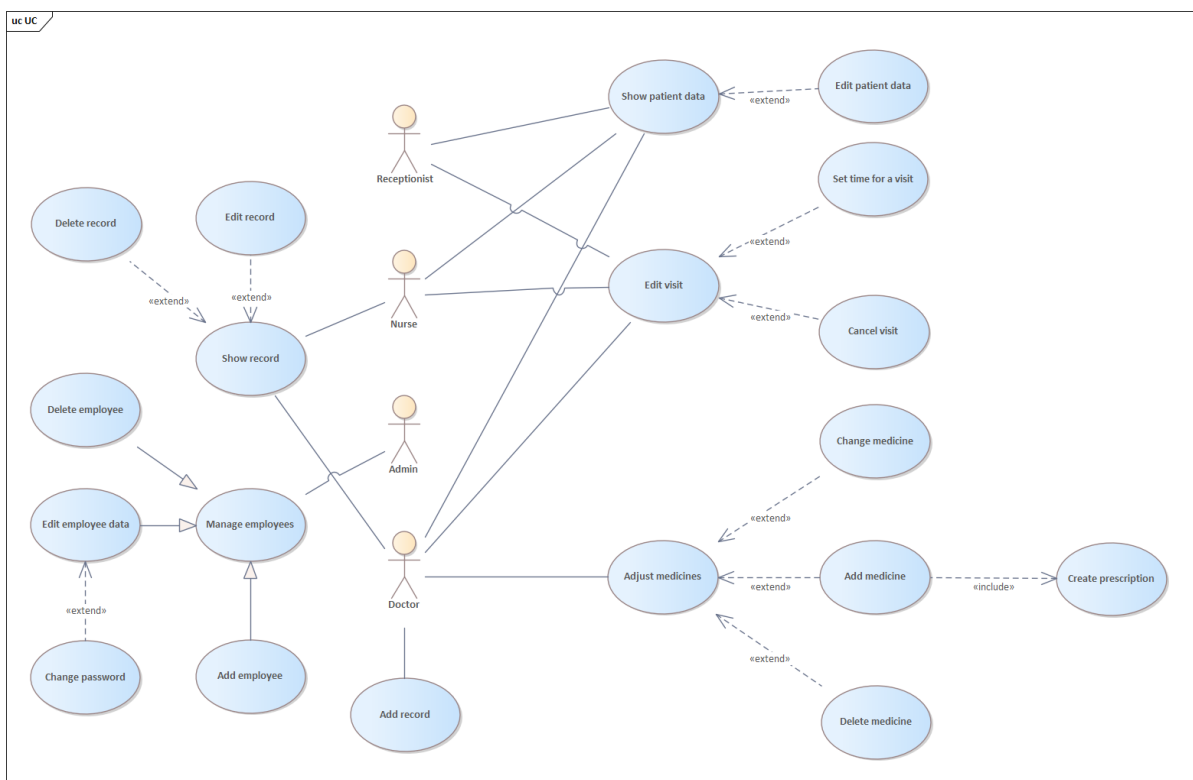
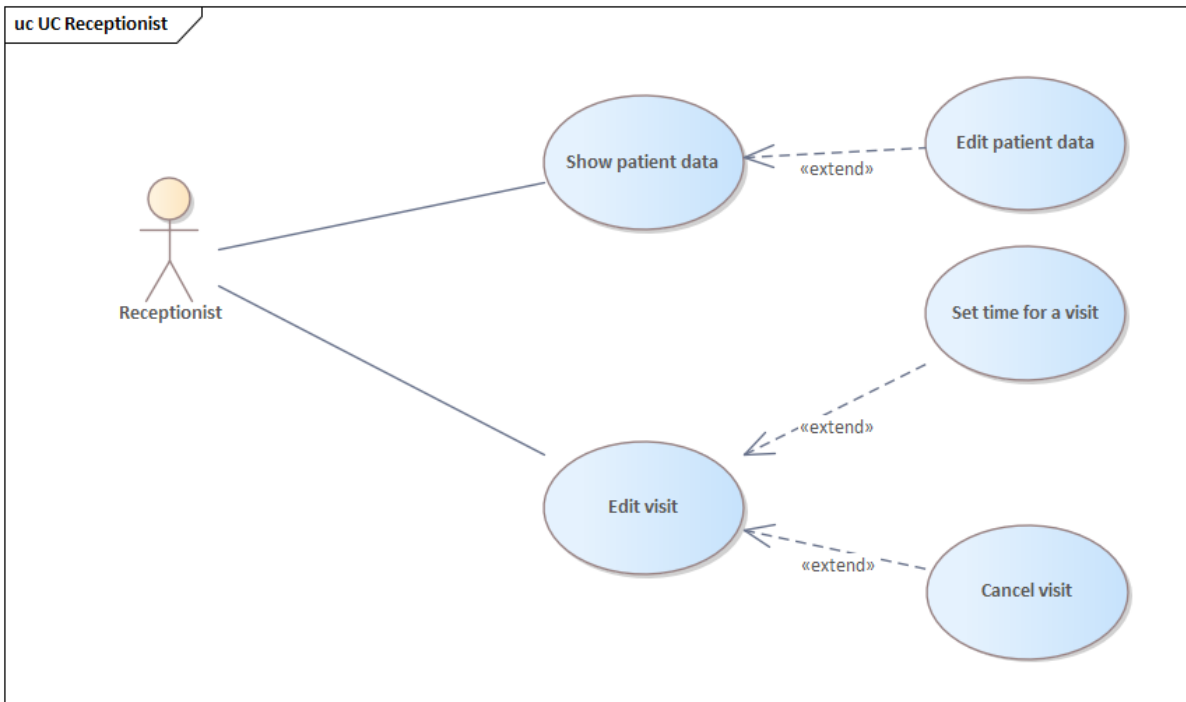
### 2.3. Nurse

Používateľ s druhým najmenším oprávnením. Tento používateľ má možnosť vytvárať budúce návštevy pacientov. Tento používateľ nemá možnosť vytvárať záznamy, má možnosť ich iba čítať.



### 2.4. Receptionist

Používateľ s najnižším oprávnením. Tento používateľ má zamedzený prístup k editovacím funkciám systému, k informáciám o medicínskych záznamoch a nemôže vytvárať návštevy. Má prístup k základným informáciám o pacientoch.

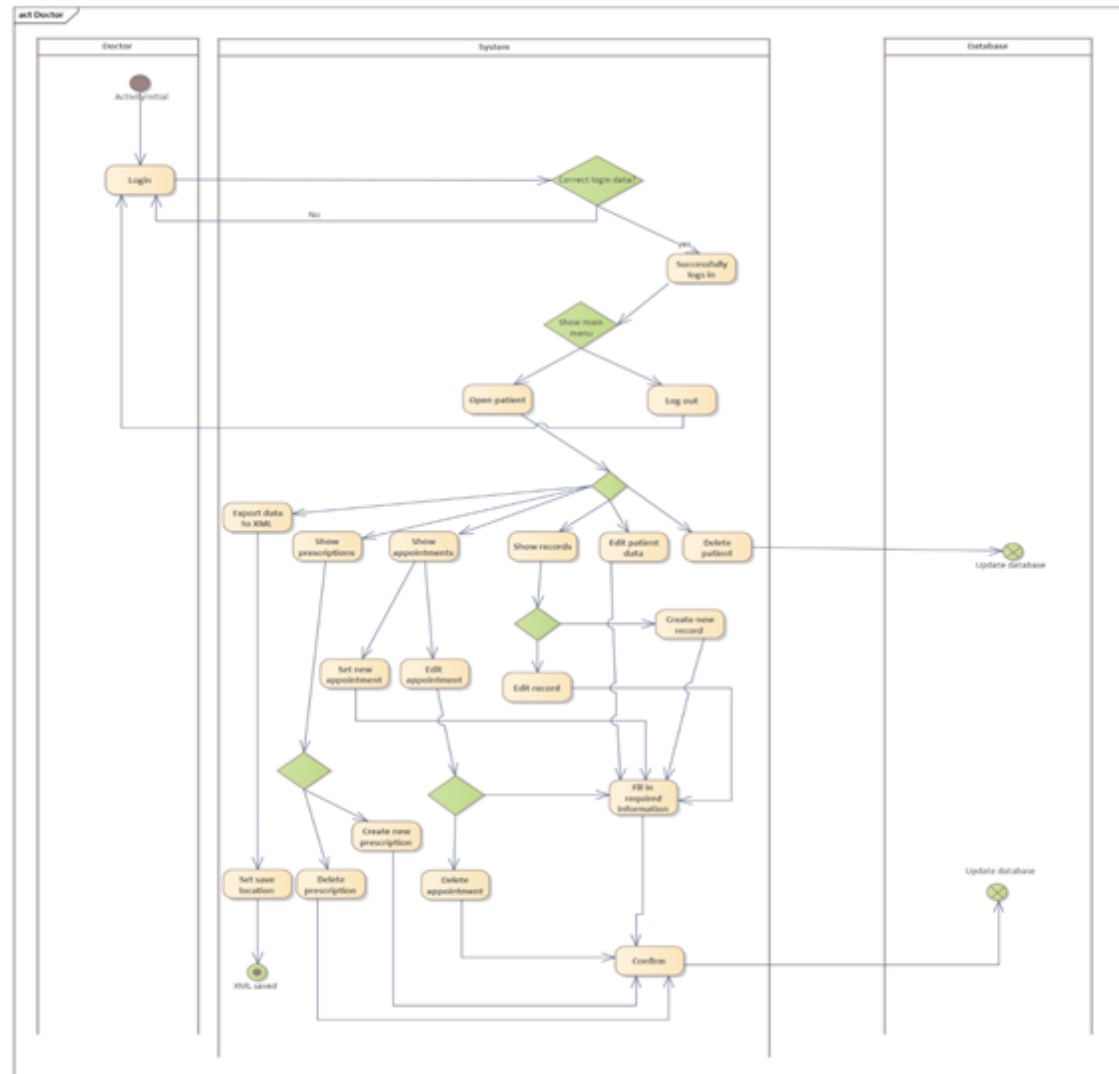


### 3. UML diagramy

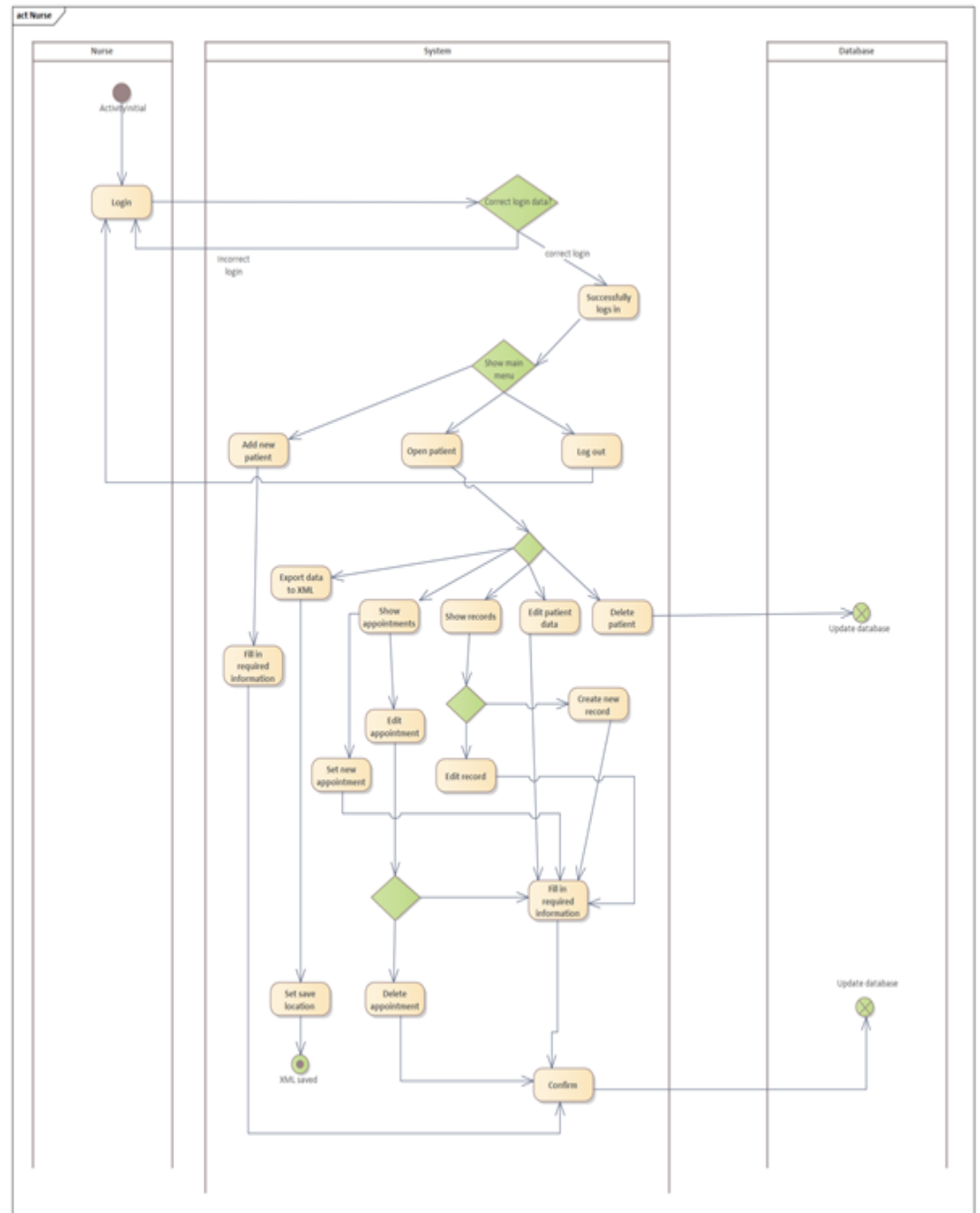
Workflow ostatných aktérov je podobné, len majú menej možností. Ako vidíme po úspešnom logine sa dostaneme na hlavné menu, čo vlastne obsahuje tabuľku pacientov a ich dáta. Odtiaľ sa môžeme odhlásiť alebo otvoriť jedného pacienta. Ak otvoríme pacienta, systém nám vráti nový screen, kde sú dostupné rozšírené informácie o pacientovi, ako aj jeho záznamy, recepty a plánované, alebo už vykonané návštevy. Ako vidíte, tu sa pribúdajú nové možnosti z ktorých doctor môže vybrať. Entity ako záznamy, návštevy výpisy na lieky vieme potom editovať, vymazať alebo pridávať nové. Konečná fáza workflowu je happy end scenár, čo je potvrdenie zmien a aktualizácia databázy.



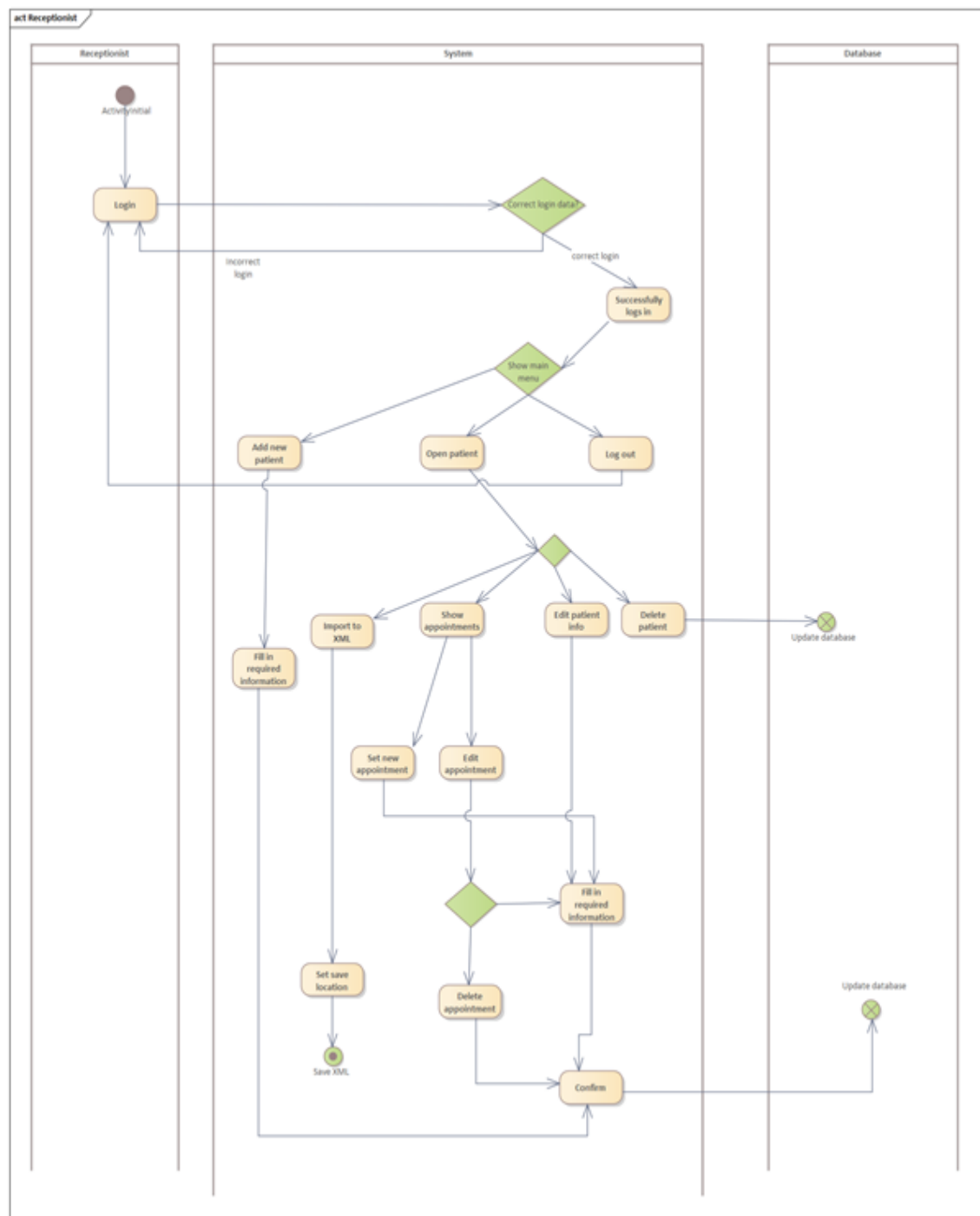
### 3.1.1. Activity diagram - Doctor



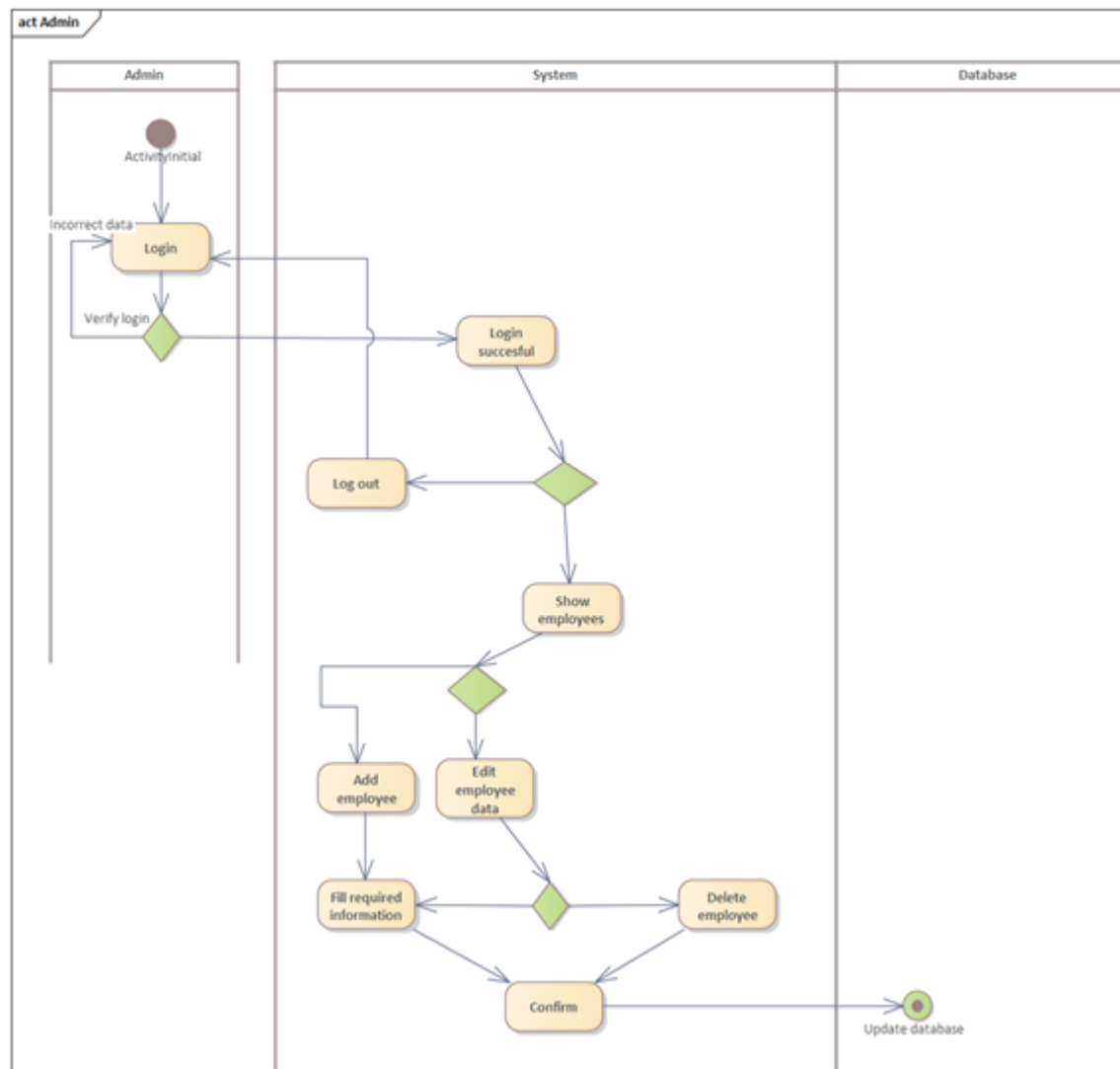
### 3.1.2. Activity diagram - Nurse



### 3.1.3. Activity diagram - Receptionist

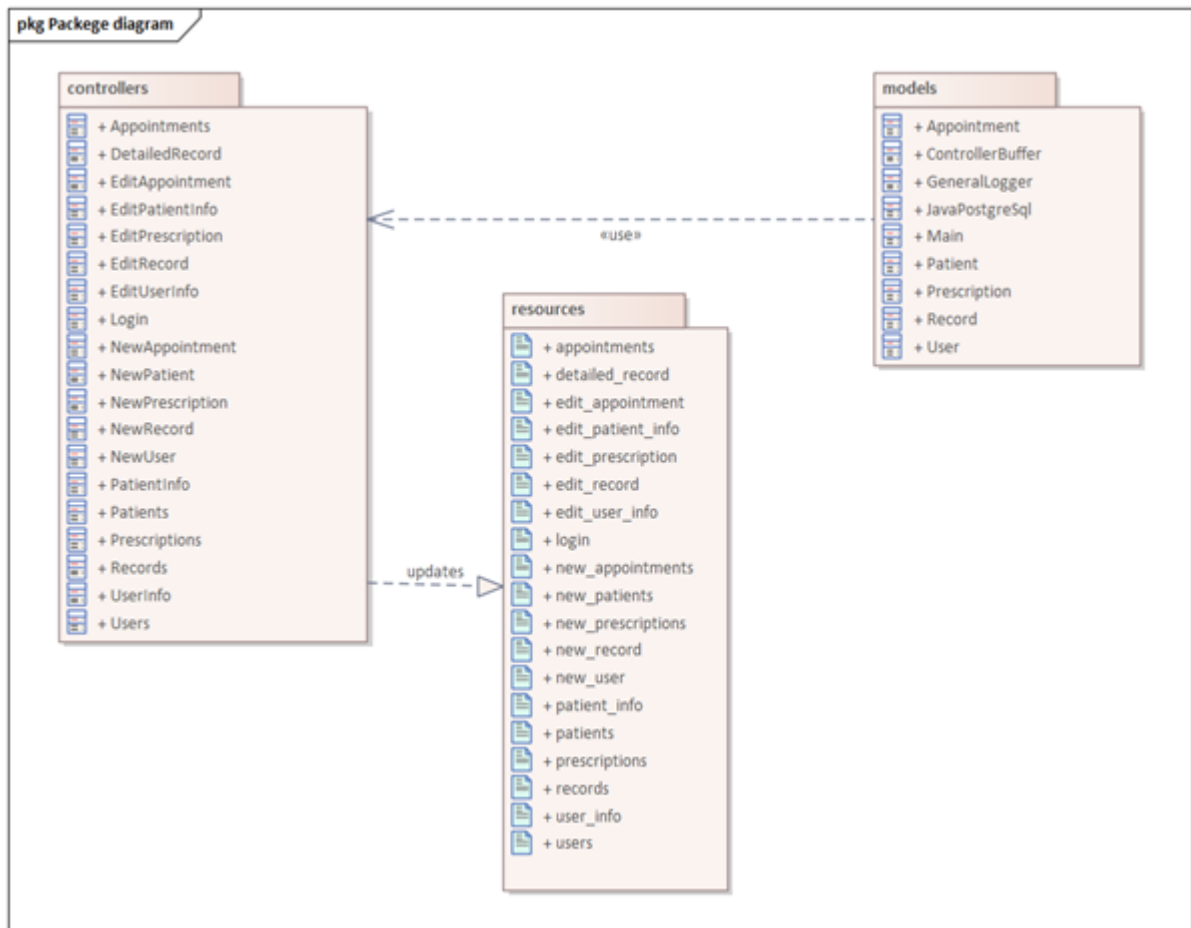


### 3.1.4. Activity diagram - Admin



### 3.1.5. Package diagram

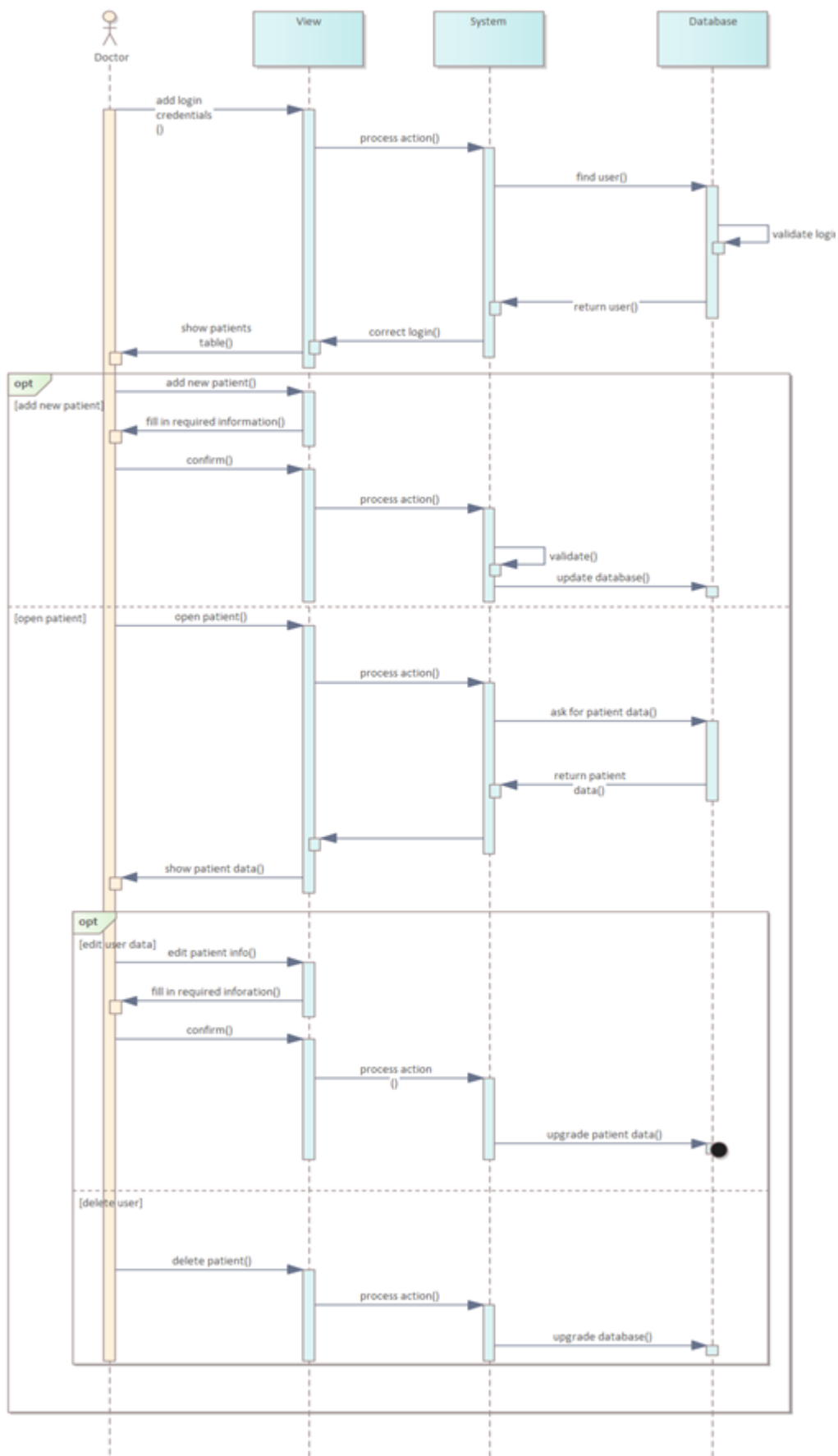
Program sme rozdelili na 3 balíky. Balík resources obsahuje všetky XML súbory, ktoré sú riadené triedami v balíku controllers. Balík models používa na controllerov na vytvorenie entít a na spracovanie databázy.



### **3.1.6. Sequence diagram – UC edit patient data/add new patient/delete patient**

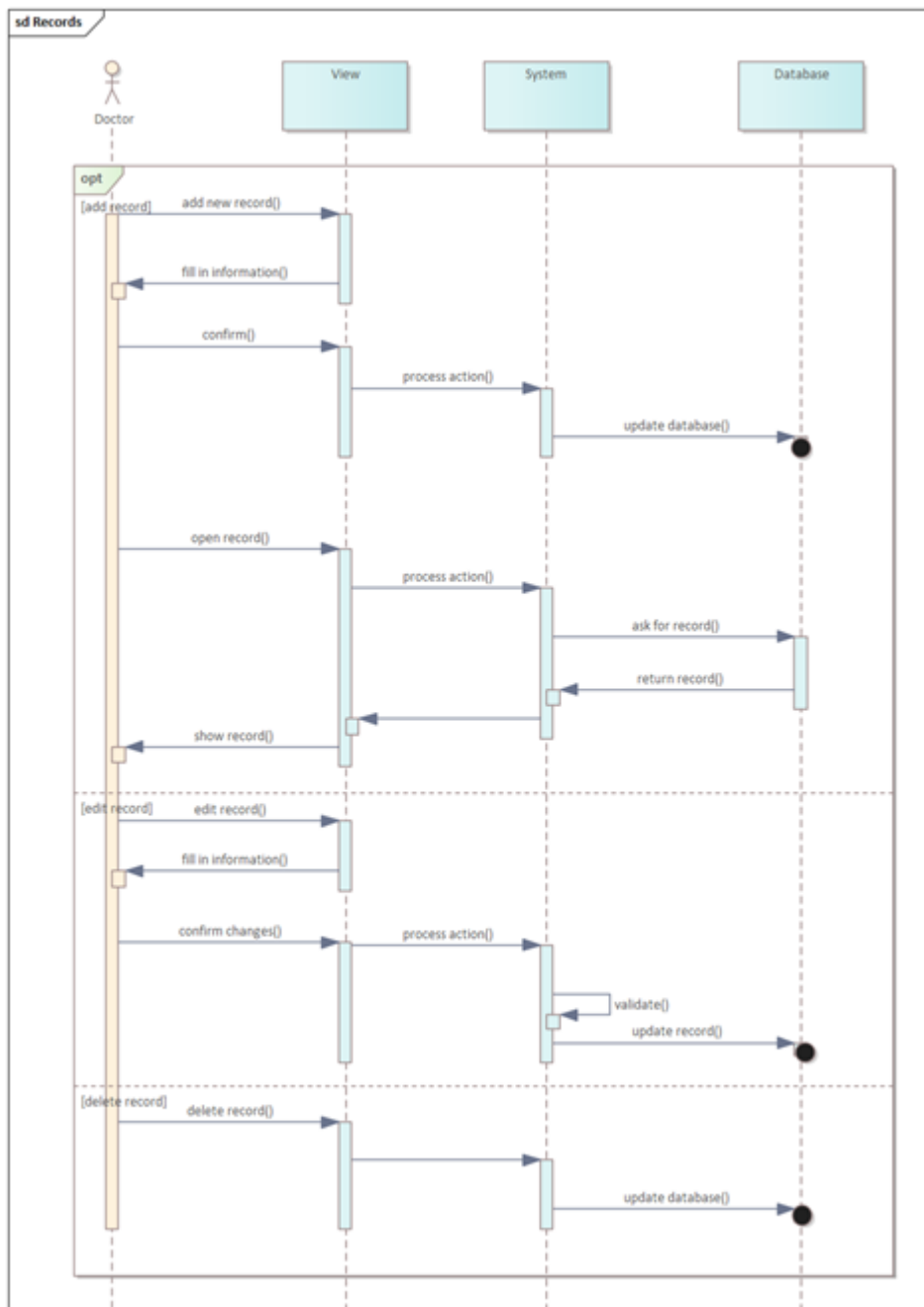
Sekvenčné diagramy znázorňujú happy end scenár. Na tomto diagrame vidíme 3 UC aktéra Doctor. Po úspešnom prihlásení, používateľ má 2 možnosti. Môže otvoriť jedného pacienta alebo pridať nového. Ak pridá nového pacienta, tak treba vyplniť požadované informácie a potvrdiť. Systém kontroluje, či všetky informácie sú zadané správne a aktualizuje databázu. Ak otvorí pacienta, tak sa pribudnú nové možnosti, z ktorých sa môže vybrať. Tieto sú vytvorenie a editovanie receptov, návštev a záznamov, ako aj vymazanie či editovanie pacienta. Na diagrame je vymazanie a editovanie pacienta. Na diagramoch sú 4 entity. 1. je aktér, ktorý vykoná prípad použitia. View je prezentačná vrstva, s ktorým interaguje aktér. Systém spracováva vstupy, a poskytuje dáta z, alebo do databázy. Databázu reprezentuje lifeline Database.

# sd edit/delete patient



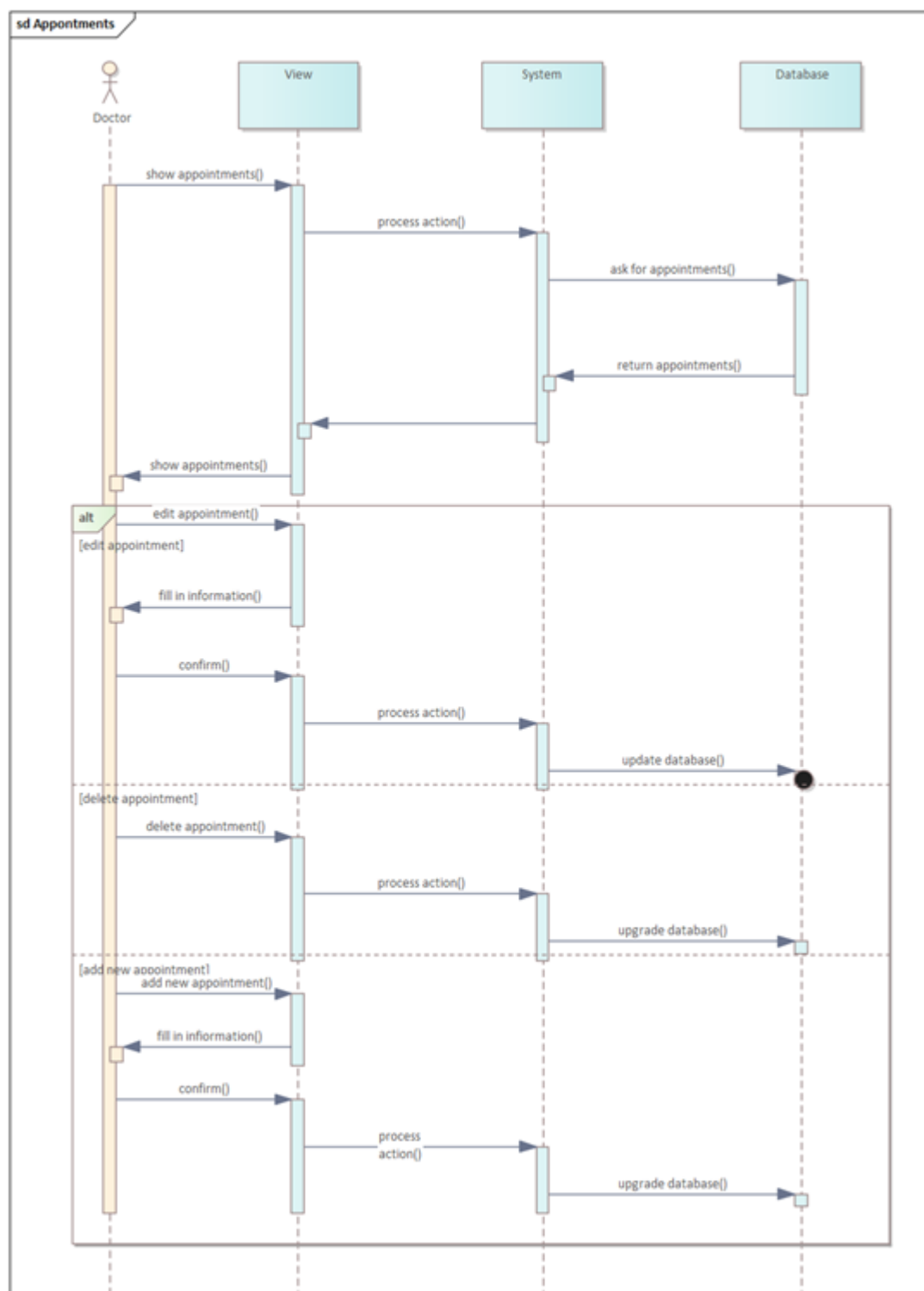
### 3.1.7. Sequence diagram - Records

Diagram znázorňuje prípady použitia k záznamom. Po otvorení pacienta treba vybrať možnosť records. Nasledujúce možnosti sú pridanie nového záznamu (record) alebo editovanie či vymazanie.

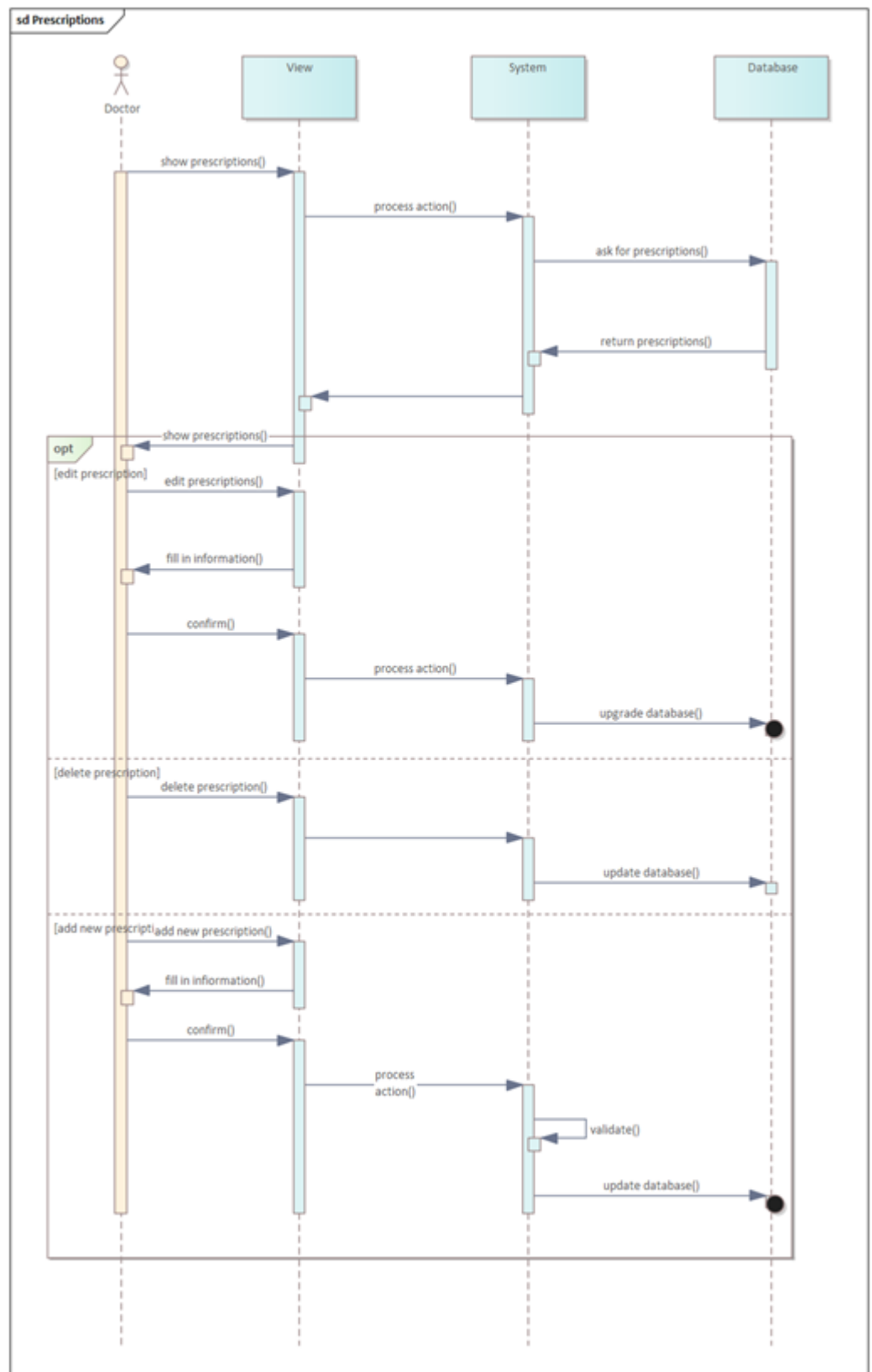




### 3.1.8. Sequence diagram – Appointments

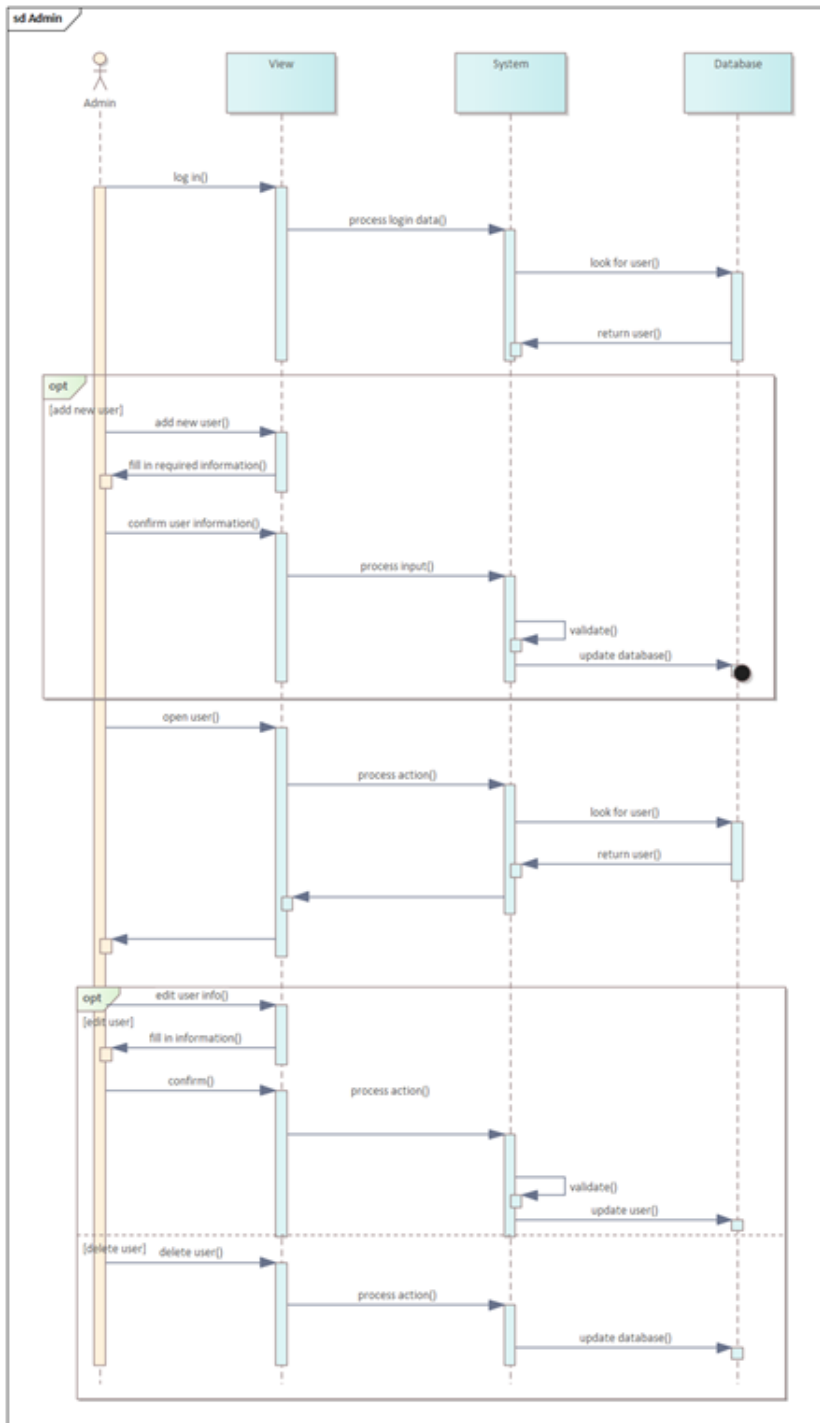


### 3.1.9. Sequence diagram – Prescriptions



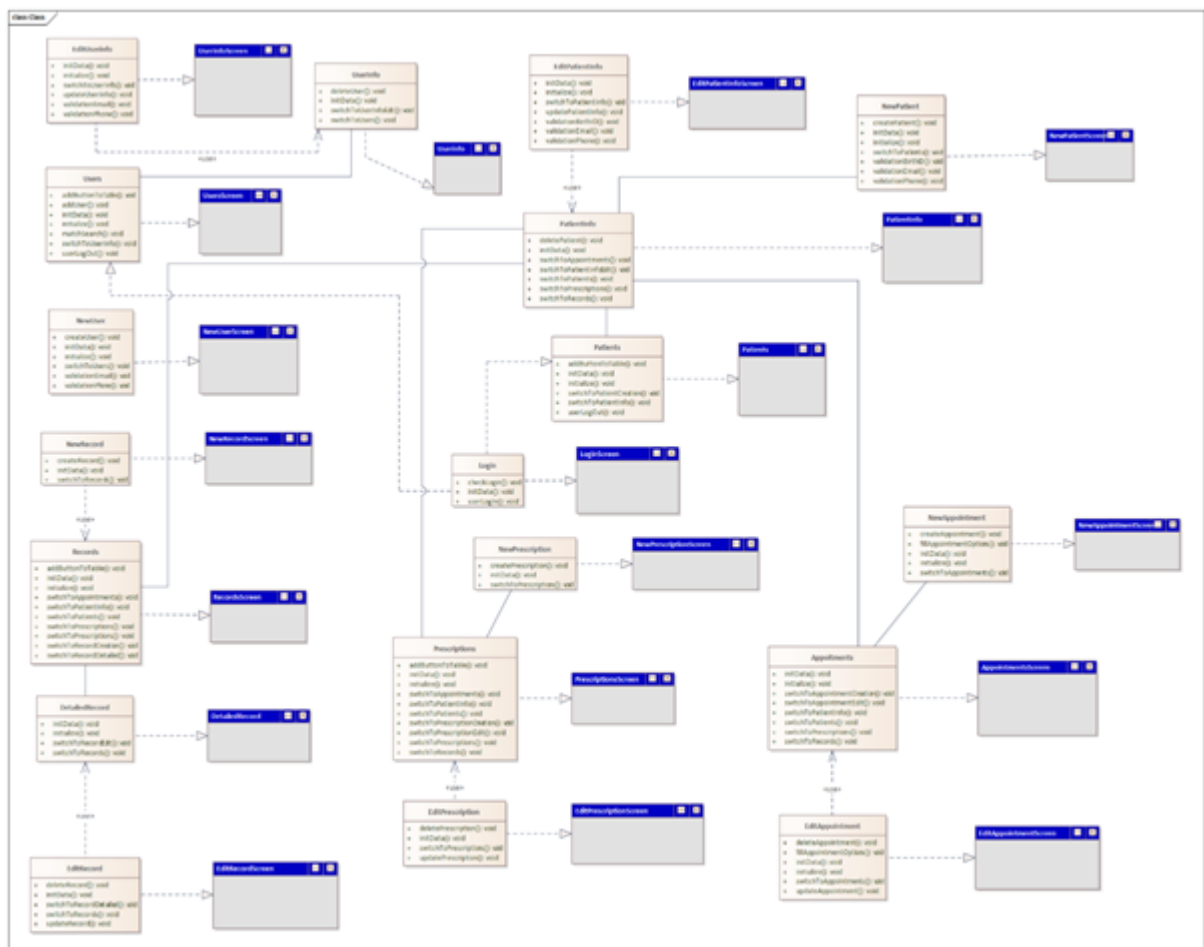
### 3.1.10. Sequence diagram – Admin

Admin po prihlásení si vie pridať nového používateľa, kto je pracovník. Podobne ako pri pacientoch, aj admin si vie vybrať používateľa a editovať jeho dáta, či vymazať vybraného používateľa.



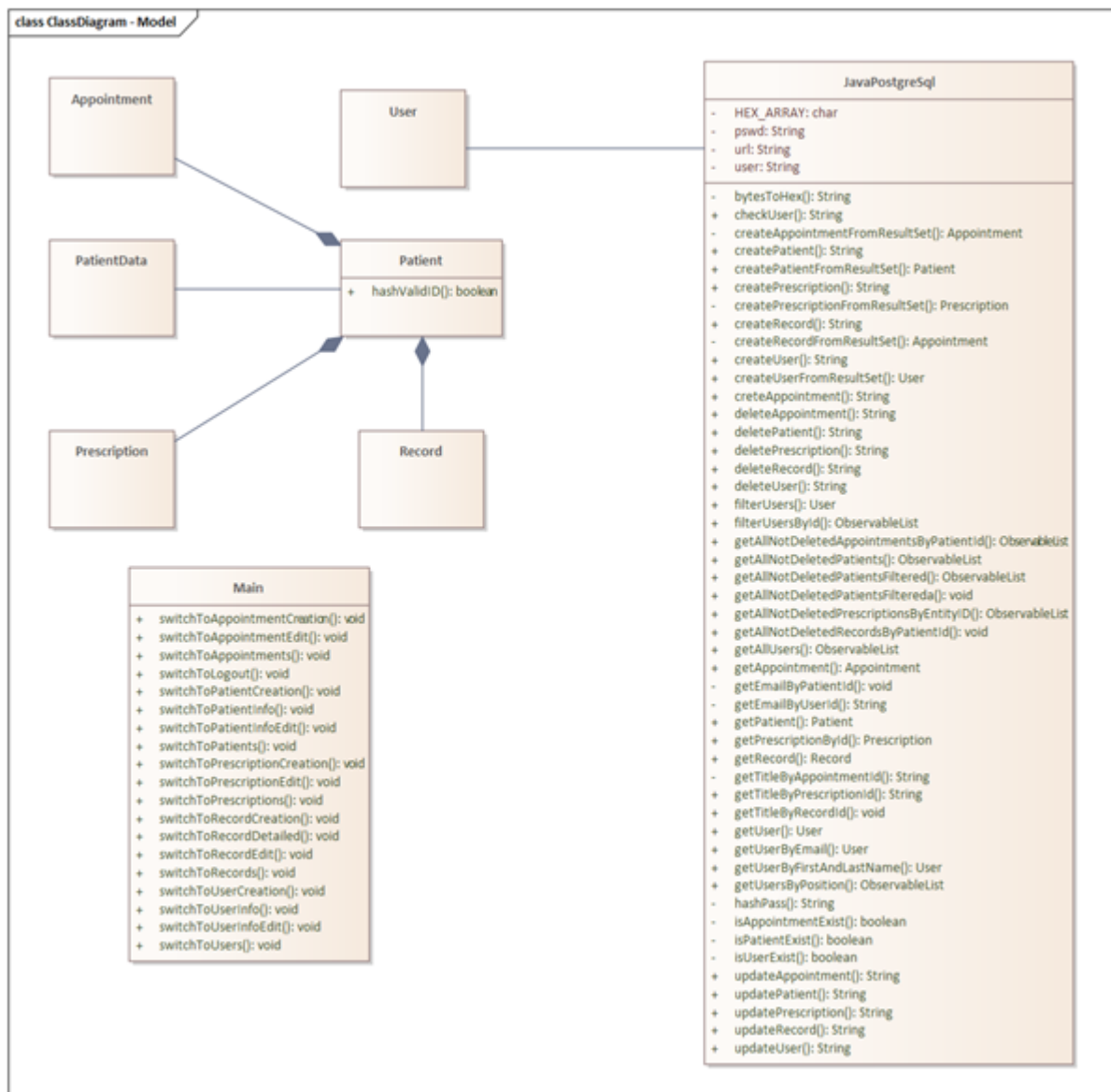
### 3.1.11. Class diagram – controllers

Na tomto diagrame sú jednotlivé obrazovky. K obrazovkám sme priradili triedy, ktoré ich realizujú, a spracovávajú eventy, vstupy vykonané na týchto obrazovkách.



### 3.1.12. Class diagram – model

Na tomto diagrame sú triedy entít, pripojenie k databázy a main. Vybrali sme kompozíciu ako vzťah medzi triedou Patient a Appointment, PatientData, Prescription, Record, keďže tento vzťah je najsilnejší a tieto objekty neexistujú bez pacienta.



#### 4. Zoznam práv používateľov

Oprávnenie	Admin	Doctor	Nurse	Receptionist
Delete record	OK	OK	OK	NG
Edit record	OK	OK	OK	NG
Show record	OK	OK	OK	NG
Delete employee	OK	NG	NG	NG
Manage employees	OK	NG	NG	NG
Edit employee data	OK	NG	NG	NG
Add employee	OK	NG	NG	NG
Change password	OK	NG	NG	NG
Add record	OK	OK	NG	NG
Show patient data	OK	OK	OK	OK
Edit patient data	OK	OK	OK	OK
Edit visit	OK	OK	OK	OK
Set time for visit	OK	OK	OK	OK
Cancel visit	OK	OK	OK	OK
Adjust medicines	OK	OK	NG	NG
Change medicine	OK	OK	NG	NG

Add medicine	OK	OK	NG	NG
Create prescription	OK	OK	NG	NG
Delete medicine	OK	OK	NG	NG

## Logging

Pre logovanie v systéme Medis používame triedu Logger (java.util.logging.Logger). Každý log, ktorý systém vyprodukuje ukladáme do súboru general\_logger.log kde môžeme nájsť úplne všetky potrebné logy. Nájdeme tam dátum, o aký typ logu ide a jeho popis.

V rámci logovania ukladáme informačné logy ako napríklad rôzne zmeny v databáze, prihlásenie používateľa alebo pridanie nového používateľa. Ukladáme taktiež chybové hlášky ako napríklad zmeny v databáze, ktoré sa nepodari vykonať. Okrem týchto chybových hlášok ukladáme aj každé nesprávne prihlásenie aby sme poprípade odhalili neoprávnené pokusy o dostanie sa do systému. Pri chybových hláškach taktiež ukladáme všetky odmietnutia prístupu keď sa používatelia snažia dostať do častí systému na ktoré nemajú dostatočné práva. Napríklad recepčná nemôže pristupovať k zdravotným záznamom pacientov.

## Validácia

Projekt medis obsahuje validáciu úplne všetkých foriem, ktoré sú v projekte použité. Implementovaná validácia nám pomáha pri správnom ukladaní dát a taktiež zabezpečuje vyššiu ochranu voči sql injekciám.

Hlavná validácia ktorá je implementovaná pre každú formu je kontrola či užívateľ vyplnil všetky vstupy. Ak tomu tak nie je systém používateľa nepustí ďalej a upozorní ho na nevyplnené polia.

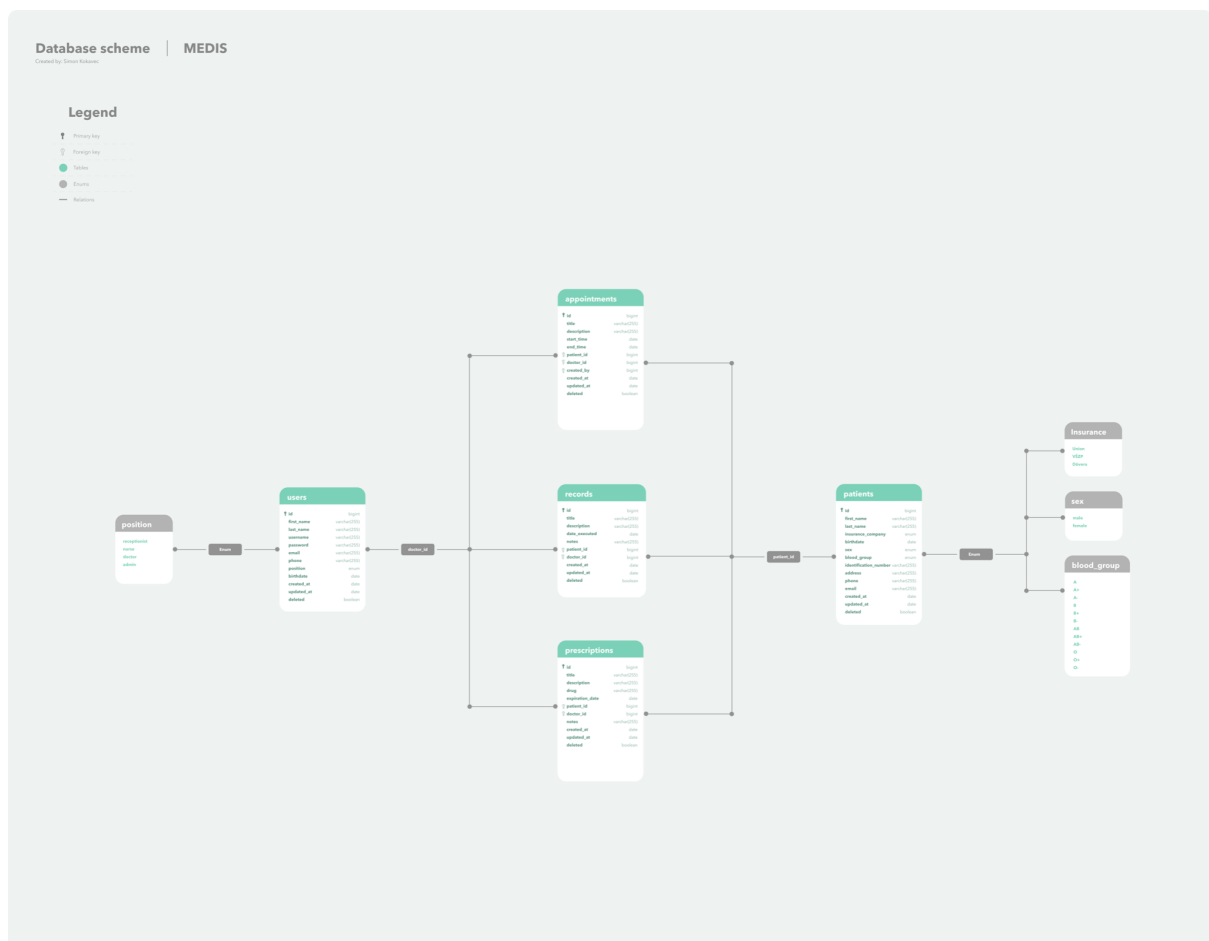
Pre už konkrétnu validáciu všetkých vstupov v našom projekte používame regulárne výrazy. Pomocou triedy Pattern (java.util.regex.Pattern) definujeme vzor ktorý následne vyhodnocujeme pomocou triedy Matcher (java.util.regex.Matcher). Každé políčko má svoje

vlastné podmienky na validáciu vstupu. Napríklad pri vstupe určenom pre e-mail systém overí či zadaný text obsahuje zavináč, kontroluje počet bodiek a aj zavádza obmedzenie na top level domain. Ak používateľ nevyplní správne vstup je upozornený ktoré políčko je vyplnené nesprávne.

Avšak medzi najzaujímavejšie a komplexnejšie kontroly by som spomenul zadávanie rodného čísla kde systém overí či dané rodné číslo spĺňa všetky požiadavky formátu Slovenského rodného čísla (Deliteľné 11, či existujú také mesiace/dni, či je pripočítaných 50 pri rodnom čísle ženy).

## Databáza

Pre projekt medis sme sa rozhodli použiť databázu PostgreSQL. Konkrétne používame verziu 10. Databázu sme sa rozhodli bežať pomocou hostingu u spoločnosti websupport, čo výrazne zlepši používanie nášho softvéru nakoľko sa môže pripojiť odkiaľkoľvek bez dodatočných nastavení. Pre pripojenie a prácu s databázou používame JDBC.





## Spustenie

1. Rozbalit' medis\_jar.7zip
2. Prečítať readme.nfo
3. Spustiť runJAR.bat