

# EDEN.t

## Návrh systému - model a databáze

*Štěpán Tesař, Lenka Vondráčková*

# Obsah

[Obsah](#)

[Návrh modelu](#)

[Popis diagramu](#)

[Vazby M:N](#)

[Vazby 1:N](#)

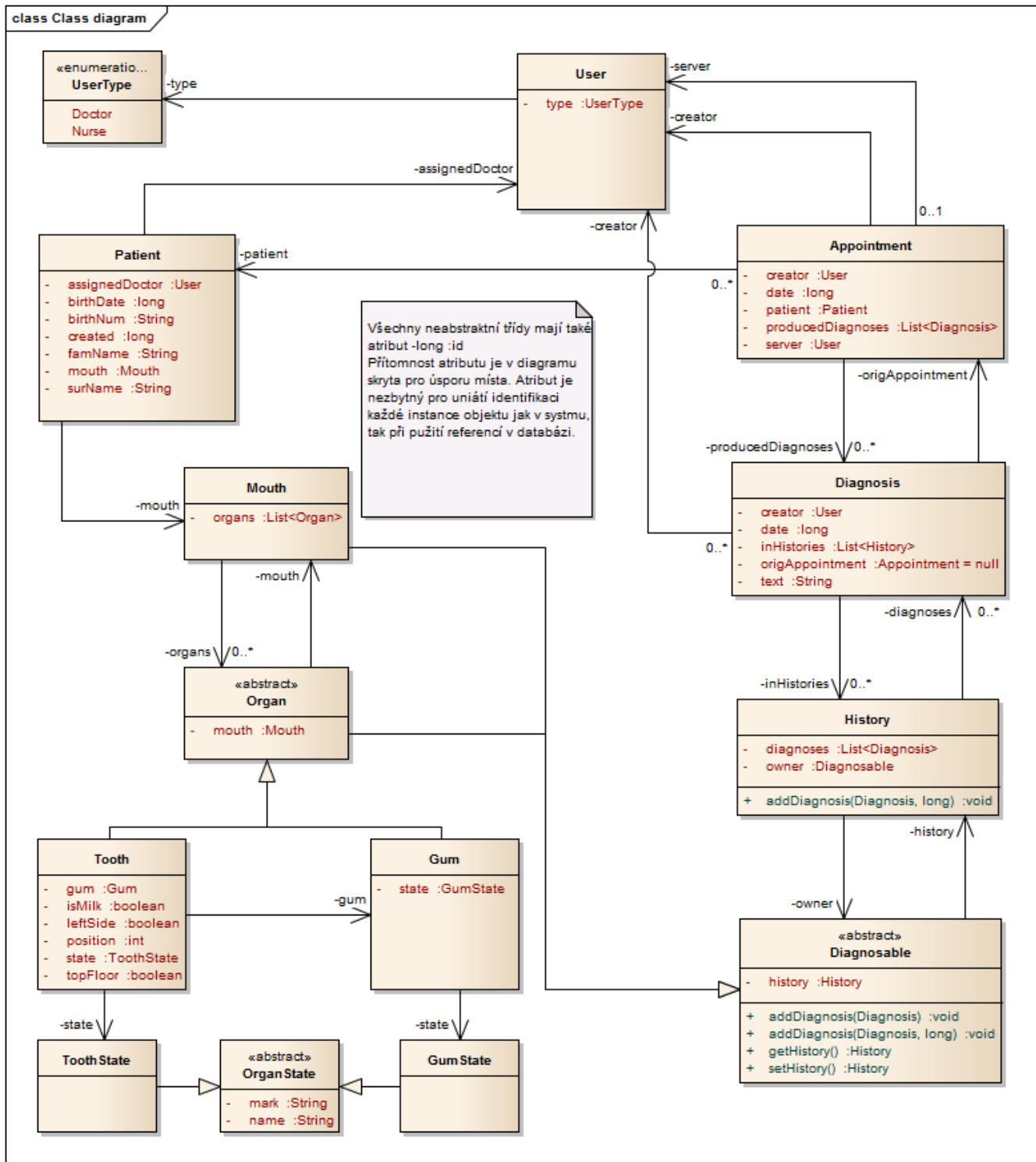
[Návrh databáze](#)

[Popis Diagramu](#)

[Relace](#)

## Návrh modelu

Diagram navrhovaných tříd, vytvořený reverzně z java souborů. Popis diagramu je opsaný a rozšířený pod obrázkem.



Obr.1: Model tříd

## Popis diagramu

Všechny neabstraktní třídy mají také atribut -long :id (přítomnost atributu je v diagramu skryta pro úsporu místa). Atribut je nezbytný pro unikátní identifikaci každé instance objektu jak v systému, tak při použití referencí v databázi. Navrhované atributy a metody jsou nekompletní, a slouží pouze pro ilustraci funkcionality dané třídy.

### Vazby M:N

Vztahy M:N jsou navrženy ve třech relacích:

1. User-Patient
2. User-Appointment
3. Diagnosis-History

Vztah User-Patient (1), zobrazuje možnost mít přiděleno více ošetřujících doktorů/sester. Vztah je přítomen hlavně proto, že systém rozlišuje doktory a sestry až na aplikační vrstvě. Aby tedy bylo možné pacientovi nastavit doktora i sestru, je z modelového hlediska toto řešení nutné. V důsledku je pak tedy možné přiřadit i více doktorů a sester, ale tato možnost zřejmě nebude uživateli systému umožněna. Na prezentační vrstvě tedy půjde o vztah M:2. Obráceně pak vazba umožňuje každému doktorovi a sestře ošetřovat více pacientů, což je logické.

User-appointment (2) je přeneseně velmi podobná relace, jelikož objekt Appointment je časově specifické provázání objektu User a Patient. Jde o termín, kdy se pacient dostaví do ordinace, a doktor (a sestra) jej ošetří. Aby pacient mohl mít přidělenou jak sestru tak doktora, je nutný vztah M:N. Tato relace však nejspíše bude zachována ve svém rozsahu i na aplikační a prezentační vrstvě. Na rozdíl od vztahu User-Patient, kdy každý pacient by měl mít přidělené právě jednoho ošetřujícího doktora, by při samotném zákroku mělo být možné seztat více doktorů a případně sester, například jedná-li se o komplikovanější operaci.

Diagnosis-History (3) vyjadřuje možnost přiřazení jedné diagnózy do více historií. Historie je objekt definovaný pro všechny <Diagnosable> objekty a zahrnuje historii diagnóz daného objektu. Relace tak umožňuje samotnou kompozici historie každého objektu z diagnóz, a zároveň diagnostikování více objektů zároveň. Například při zánětu dásní je pravděpodobné, že se zánět nevyskytuje pouze v oblasti jednoho zubu, ale je rozšířen do větší oblasti.

### Vazby 1:N

Těchto vazeb je v systému vícero, a jsou povětšinou logické. První objekt v řádku je vždy ten s násobností N vůči druhému objektu.

1. Appointment-Patient - Pacient má domluvené více schůzek (po celou historii).
2. Appointment-User - Uživatel obsluhuje více schůzek (po celou historii).
3. Diagnosis-User - Uživatel vytváří mnoho diagnóz.
4. Diagnosis-Appointment - Na schůzi může být vytvořeno mnoho diagnóz.
5. Tooth/Gum-Mouth - V puse je mnoho zubů a dásní\*

\* Objekt Gum (dásně) určuje dásně v oblasti jednoho zubu. V diagramu jsou objekty Tooth a Gum v této relaci zastoupeny společným předkem Organ.

## Návrh databáze

Jako databázové úložiště bude použita architektura PostgreSQL. Pro objektově-relační mapování pak framework hibernate, s použitím mapovacích XML souborů. Pomocí hibernate byly třídy z předchozího diagramu namapovány na databázovou strukturu. Reverzním inženýrstvím byl pak vytvořen následující diagram (viz obr.2 na další straně).

### Popis Diagramu

Relace v diagramu z pravidla odpovídají těm v diagramu tříd. Logicky nejsou namapované abstraktní třídy a rozhraní `Organ`, `OrganState` a `Diagnosable`. Relace objektů, které tyto třídy či rozhraní implementují, se tedy přenesli přímo na objekty které s nimi byli původně spojené skrze tyto zástupce.

### Relace

V diagramu lze vidět realizace relací M:N z předchozího diagramu, za použití propojovacích tabulek odpovídajících názvům propojovaných objektů. Jde tedy o tabulky `user_patient`, `user_appointment` a `diagnosis_history`.

Relace 1:N kopírují schéma modelu. Hibernate však nešikovně zpracovává relace 1:1v případě, že objekty mohou existovat i bez sebe, takže některé z těchto relací jsou také řešeny způsobem 1:N a přidaným příznakem `unique`, který v důsledku zajistí, že  $N=1$ . V diagramu jsou pak nezbytně tyto závislosti zobrazeny jako dvě relace 1-\* v opačném směru. Pro přehlednost byla v obrázku vždy jedna z těchto relací skryta a u druhé změněna násobnost na 1-1. Nástroj<sup>1</sup> neumožnil změnit směr relace na oboustranný, takže některé šipky mohou být zavádějící.

Dále byl také problém s namapováním relace `History-Diagnosable`. V tuto chvíli je v návrhu tato relace zpracována relativně nešikovně pomocí tří různých atributů v tabulce `history`.

---

<sup>1</sup> K vytvoření diagramů byl použit nástroj Enterprise Architect.

