

# EDEN.t

## Analytická část

*Štěpán Tesař, Lenka Vondráčková*

# Obsah

[Obsah](#)

[Vymezení aplikace](#)

[Analýza procesů](#)

[Diagram procesů](#)

[Doménový model](#)

[Popis diagramu](#)

[Požadavky](#)

[Nefunkční požadavky](#)

[Funkční požadavky](#)

[Požadavky semestrální práce](#)

[Vlastní požadavky](#)

## Vymezení aplikace

Aplikace EDEN.t je semestrální prací na předmět Lékařská Informatika ([a6m33li](#)), a požadavky na tento systém tedy vyplývají a zároveň jsou omezeny obecnými požadavky na semestrální práce v tomto předmětu. Základní požadavek je aplikace nasaditelná v medicínském prostředí. Jako toto prostředí byla zvolena dentální ordinace. Aplikace by tedy měla usnadnit agendu dentální ordinace. Do této agendy patří především organizace schůzek s pacienty a sledování a aktualizace jejich dentálního zdraví. Do organizace pak patří tak komunikace s pojišťovnami, sledování použitého materiálu a jeho vyúčtování nebo také plánování směn lékařů a zdravotních sester. Z důvodu velkého rozsahu aplikace, která by obsáhla všechny nutné činnosti, a časového i rozsahového omezení této práce, zúžíme požadavky na primární, tedy organizace schůzek s pacienty a sledování jejich dentálního zdraví.

## Analýza procesů

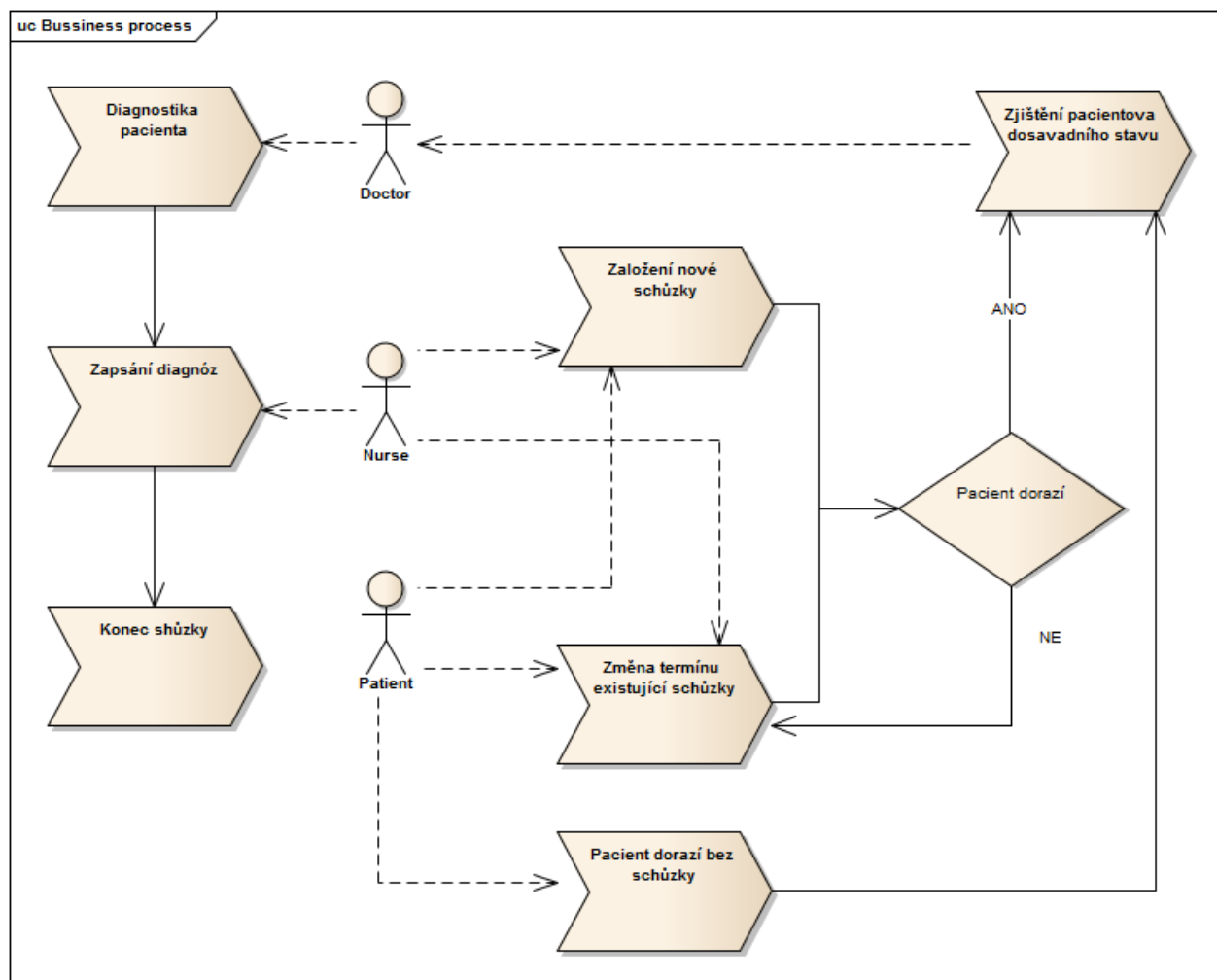
Pro konkrétní představu o průběhu dění v dentální ordinaci došlo ke konzultaci s kolegy z řad studentů medicíny specializovaných na zubaře. Souhrn podstatných informací, které z těchto konzultací vyplynuli, je v následujícím seznamu.

- Je podstatné rozlišení zdravotní sestry a samotného doktora.
- V jedné praxi může ordinovat více doktorů i více sester.
- Kromě zubů samotných jsou oblastmi zájmu také dásně, patra, a obecně celá ústní dutina.
- “Doktor křičí diagnózy, sestra zapisuje”
- Je velmi časté, že pacient na dohodnutý termín nepřijde, nebo ho v poslední chvíli změní.
- Pacient může dorazit mimo termín, nebo si ho vůbec nedomluví.
- Před samotnými ordinačními hodinami se přijímají bolestivé případy bez objednání.
- Je podstatné mít k dispozici aktuální pacientův stav ve chvíli, když “lehá do křesla”.

Zajímavých informací, například ohledně používání materiálu nebo vyúčtování péče a nákladů, bylo samozřejmě více, ale uvádíme zde pouze ty, které se týkají naší aplikace ve vymezeném rozsahu.

## Diagram procesů

Procesy v ordinaci od domluvení schůzky až po odchod pacienta je znázorněn na následujícím diagramu.

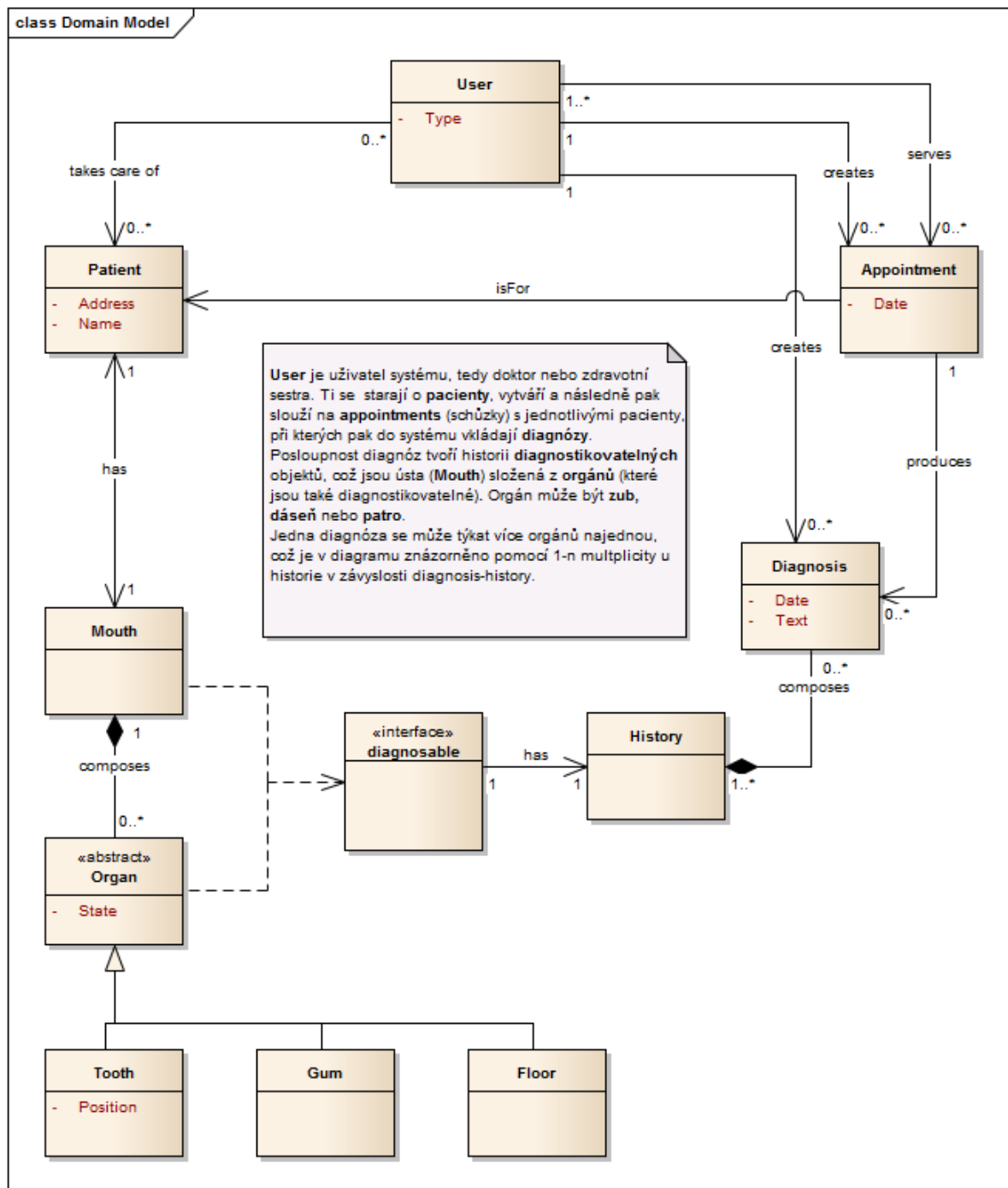


**Obr.1:** Bussiness procesy v dentální ordinaci

Diagram opět obsahuje pouze tu část procesů, která je zásadní pro naši aplikaci. V diagramu také není rozlišen příchod bez schůzky v ordinčních hodinách, a příchod bez objednání jako “bolestivý případ”. Systém ale zřejmě nebude tyto dva typy příchodu rozlišovat.

## Doménový model

Entity vystupující v nastudovaných procesech a vazby mezi nimi jsou znázorněny na diagramu níže. Některé entity mají atributy demonstrující jejich účel. Součástí diagramu je také stručný popis, který je pro zlepšení čitelnosti přepsán pod obrázkem.



Obr.2: Doménový model

## Popis diagramu

**User** je uživatel systému, tedy doktor nebo zdravotní sestra. Ti se starají o **pacienty**, vytváří a následně pak slouží na **appointments** (schůzky) s jednotlivými pacienty, při kterých pak do systému vkládají **diagnózy**.

Posloupnost diagnóz tvoří **historii diagnostikovatelných** objektů, což jsou ústa (**Mouth**) složená z **orgánů** (které jsou také diagnostikovatelné). Orgán může být **zub**, **dáseň** nebo **patro**. Jedna diagnóza se může týkat více orgánů najednou, což je v diagramu znázorněno pomocí 1-n multiplicity u historie v závislosti diagnosis-history.

Rozlišení doktor/sestra bude v programu nejspíše pro jednoduchost řešeno pomocí atributu. Objektově správně by bylo vytvoření rolí (přiřaditelných uživatelům) s právy na určité akce v systému, ale implementace této infrastruktury by s největší pravděpodobností zabrala zbytečně mnoho času.

Do ještě větší objektové kolize se dostáváme u zubů, které mají svou pozici v podstatě vždy danou, a navíc mají specifické vlastnosti (tvar koruny, počet hrotů kořene) a bylo by tedy vhodné vytvořit pro každý zub vlastní třídu. Opět pro jednoduchost bude toto zřejmě řešeno automatizovaným skriptem, který pro každého nového pacienta vytvoří standardní sadu zubů, které budou rozlišeny pouze svou pozicí.

Konkrétní řešení těchto problémů, zvažující všechny možnosti a požadavky ale bude záležitostí dalších částí práce.

## Požadavky

Kapitola obsahuje sběr požadavků na výslednou aplikaci, vycházející z dosavadní analytické činnosti a samotného zadání práce. Opět jsou vyčteny pouze reálně zpracovatelné požadavky, které bychom rádi do aplikace skutečně implementovali.

### Nefunkční požadavky

- Implementace v jazyce JAVA
- Využití databáze PostgreSQL
- Databáze obsahující minimálně 10 tabulek (min. 2x vztah M:N a 3x vztah 1:N)
- Použití DAO objektů, nebo ORM frameworku.

### Funkční požadavky

#### Požadavky semestrální práce

- Přehledné, jasně a logicky uspořádané GUI.
  - Vhodné předvedení vazeb M:N
- Možnost filtrovat tabulkový výstup z databáze na základě minimálně dvou parametrů.

#### Vlastní požadavky

- Rozlišení, zda aplikaci ovládá sestra nebo doktor, a z toho vyplývající možnosti úkonů.
- Evidence pacientů.
- Vytváření a změny schůzek s pacienty.
- Přiřazení konkrétního doktora k pacientovi na danou schůzku.
- Diagnostika jednotlivých zubů a částí dásní, ale i společné diagnózy pro více entit najednou.
- Vedení historie diagnóz pro jednotlivé orgány pacienta.
- Možnost přijmout pacienta bez nutnosti vytvářet schůzku.