

## 1. Cíle projektu

Cílem projektu je implementovat informační systém s webovým rozhraním, podle návrhu provedeného v rámci předmětu Databázové systémy (IDS) pro zvolené zadání (případně zadání z předmětu IUS). Výjimku tvoří téma *Internetový obchod* (viz níže). Postup řešení by měl být následující:

1. Volba implementačního prostředí - databázového serveru a aplikační platformy
2. Implementace navrženého databázového schématu ve zvoleném DB systému
3. Návrh webového uživatelského rozhraní aplikace
4. Implementace vlastní aplikace

## 2. Rozsah implementace

Implementovaný systém by měl být prakticky použitelný pro účel daný zadáním. Mimo jiné to znamená:

- Musí umožňovat vložení odpovídajících vstupů.
- Musí poskytovat výstupy ve formě, která je v dané oblasti využitelná. Tedy nezobrazovat obsah tabulek databáze, ale prezentovat uložená data tak, aby byla pro danou roli uživatele a danou činnost užitečná (např. spojit data z více tabulek, je-li to vhodné, poskytnout odkazy na související data, apod).
- Uživatelské rozhraní musí umožňovat snadno realizovat operace pro každou roli vyplývající z diagramu případů použití (use-case). Je-li cílem např. prodej zboží, musí systém implementovat odpovídající operaci, aby uživatel nemusel při každém prodeji ručně upravovat počty zboží na skladě, pamatovat si identifikátory položek a přepisovat je do objednávky a podobně.

Kromě vlastní funkcionality musí být implementovány následující funkce:

- Správa uživatelů a jejich rolí (podle povahy aplikace, např. obchodník, zákazník, administrátor). Tím se rozumí přidávání nových uživatelů u jednotlivých rolí, stejně tak možnost editace a mazání nebo deaktivace účtů. Musí být k dispozici alespoň dvě různé role uživatelů.
- Ošetření všech uživatelských vstupů tak, aby nebylo možno zadat nesmyslná nebo nekonzistentní data.
  - Povinná pole formulářů musí být odlišena od nepovinných.
  - Hodnoty ve formulářích, které nejsou pro fungování aplikace nezbytné, neoznačujte jako povinné (např. adresy, telefonní čísla apod.) Nenuťte uživatele (opravujícího) vyplňovat desítky zbytečných řádků.
  - Při odeslání formuláře s chybou by správně vyplněná pole měla zůstat zachována (uživatel by neměl být nucen vyplňovat vše znovu).
  - Pokud je vyžadován konkrétní formát vstupu (např. datum), měl by být u daného pole naznačen.
  - Pokud to v daném případě dává smysl, pole obsahující datum by měla být předvyplněna aktuálním datem.
  - Nemělo by být vyžadováno zapamatování a zadávání generovaných identifikátorů (cizích klíčů), jako např. ID položky na skladě. To je lépe nahradit výběrem ze seznamu. Výjimku tvoří případy, kdy se zadáním ID simuluje např. čtečka čipových karet v knihovně. V takovém případě prosím

ušetřete opravujícímu práci nápovědou několika ID, která lze použít pro testování.

- Žádné zadání nesmí způsobit nekonzistentní stav databáze (např. přiřazení objednávky neexistujícímu uživateli).
- Přihlašování a odhlašování uživatelů přes uživatelské jméno a heslo. Automatické odhlášení po určité době nečinnosti.

### 3. Implementační prostředky

#### 3.1 Uživatelské rozhraní

- HTML5 + CSS, s využitím JavaScriptu, pokud je to vhodné. Je povoleno využití libovolných volně šířených JavaScriptových a CSS frameworků (jQuery, Bootstrap, atd.)
- Případně lze využít i AJAX či pokročilejší klientské frameworky (Angular, React, apod.), není to ale vyžadováno.

Rozhraní musí být funkční přinejmenším v prohlížečích Chrome, Firefox a Internet Explorer.

#### 3.2 Implementační prostředí

- PHP + MySQL (server eva nebo jiný dostupný), případně libovolný open source PHP framework.
- Alternativně jiná serverová technologie (např. Java, .NET, Python, Ruby, apod.) - kontaktujte J. Hynka a domluvte se na detailech.

Při použití relační databáze **specifikujte integritní omezení** (např. unikátní hodnoty, cizí klíče, apod.) při vytváření databáze. Neopouštějte zajištění konzistence dat pouze na aplikaci. V MySQL je k tomu třeba použít tabulky typu InnoDB (typ je možno zvolit při vytváření tabulky nebo změnit dodatečně).

Použití jiné platformy je možné, ale je nutné se předem domluvit se cvičícím a samostatně si zajistit umístění vytvořené aplikace, aby bylo možno ji předvést.

### 4. Dokumentace

Součástí projektu je stručná dokumentace k implementaci, která popisuje, které PHP skripty (případně kontrolery, presentery apod. podle zvoleného frameworku) implementují jednotlivé případy použití uvedené v use-case diagramu z IDS. V případě, že bylo nutno při implementaci učinit změny v návrhu, musí být v dokumentaci zdůvodněny. Tato dokumentace je součástí dokumentu [doc.html](#), viz níže.

### 5. Řešitelské týmy

Řešení projektů probíhá v týmech (obvykle dvojicích) stejného složení jako předchozí řešení v IDS. Pokud se původní tým rozpadl, je doporučeno vytvořit tým s jiným, podobně postiženým řešitelem i za cenu změny zadání. V nouzi může řešitel pracovat sám, znamená to ale odvést podstatně více práce (nelze odevzdat poloviční projekt). Složení týmu musí být uvedeno v souboru [doc.html](#), viz níže.

Nejpozději do **1.11.2018** se členové týmu ujistí, že spolupráce funguje, nejlépe tím, že začnou na projektu pracovat. Do tohoto termínu také prosím **založte tým v IS FIT a přihlaste se na projekt**. V případě výpadku nebo nespolupráce jednoho člena lze situaci

ohlásit a řešit opět výše uvedeným způsobem. Po tomto datu již změny nejsou možné - projekt by se již pravděpodobně nestihnul dokončit. Ostatní problémy v týmu budou řešeny individuálně.

## **6. Zadání z předmětu IDS**

Vaším úkolem je vývoj IS lékárny. Lékárna vydává občanům léky jak na předpis, tak za hotové. U léků vydávaných na předpis může část ceny hradit zdravotní pojišťovna. Některé léky se vydávají pouze na předpis. Systém musí umožnit evidenci vydávaných léků, import příspěvků na léky od zdravotních pojišťoven (může se čas od času měnit), musí poskytovat export výkazů pro zdravotní pojišťovny a musí mít vazbu na skladové zásoby (vidět, zda požadovaný lék je na skladě). Léky jsou identifikovány číselným kódem či názvem.

## **7. Zadání z předmětu IUS**

Vaším úkolem je vývoj informačního systému lékárny. Lékárna vydává občanům léky jak na předpis, tak i bez předpisu. U léků vydávaných na předpis může část ceny hradit zdravotní pojišťovna. Některé léky se vydávají pouze na předpis. U léků na předpis musí systém evidovat informaci o rodném čísle zákazníka a pojišťovně, které bude částka za daný lék fakturována. Systém musí umožnit evidenci vydávaných léků, import aktuálních hodnot příspěvků na léky od zdravotních pojišťoven (může se čas od času měnit) a musí evidovat množství zboží na skladě. Léky jsou identifikovány číselným kódem či názvem. Firma má několik poboček, informační systém tedy musí být schopen zjistit množství zboží na skladě na zvolené pobočce, stejně tak se eviduje i prodané zboží vzhledem k dané pobočce. Požaduje-li zákazník zboží, které momentálně není na dané pobočce k dispozici, může si jej zarezervovat. Náš sortiment neobsahuje zboží, u kterého bychom si potřebovali pamatovat, kdo si jej koupil, u prodaného zboží si tedy nepamatujeme detaily o zákazníkovi, stejně tak v případě rezervací poslouží pouze jméno zákazníka. Informační systém bude obsluhován prozatím pouze zaměstnanci lékárny. Doplatky pojišťoven na jeden lék se různí, stejně tak jako cena léku prodaného bez předpisu. Evidujte i dodavatele lékárny, jakož i jejich ceny pro dané zboží. Objednávky u těchto dodavatelů již modelovat nemusíte. Předpokládejte, že není třeba evidovat zaměstnance, je irelevantní, kdo prodej zboží realizoval.