

# Výběr semináře

Tento dokument popisuje funkční chování systému **Zápis seminářů**. Cílem je umožnit studentům přihlásit se na nabízené semináře (předměty) v rámci definovaného zápisu, který spravuje administrátor.

## Motivace

Tento systém vznikl jako jednoduchý, přehledný a interaktivní nástroj pro **organizaci školních seminářů a zápisů studentů**. Je navržen tak, aby pokryl všechny klíčové potřeby konkrétní školy kde se planuje systém nasadit, ale zároveň zůstal dostatečně lehký, intuitivní a snadno upravitelný.

### hlavní cíle

Cílem systému je vytvořit **jednotné místo**, kde:

- studenti mohou snadno **vybírat semináře** podle svých preferencí
- učitelé mají přehled o svých skupinách a mohou vidět zapsané studenty
- administrátoři mohou **spravovat předměty, bloky, zápisová období a uživatele**
- celý proces zápisu je jasně strukturovaný, přehledný a transparentní

Systém tak eliminuje ruční evidenci, zdlouhavou komunikaci e-mailem nebo tabulkovými procesory a přináší **automatizaci a pořádek**.

## SEO a klíčová slova

SEO má význam především u veřejně dostupných webů, které se mají zobrazovat ve výsledcích vyhledávání a kde je to naše snaha. Tento systém je ale čistě interní – přístup mají pouze přihlášení uživatelé (studenti, učitelé, administrátoři).

Teoreticky by SEO dávalo smysl v případě, že bychom chtěli aby ji byly studenti schopni najít ve vyhledávači, namísto odkazu na stránkách školy. To však není cílem tohoto interního systému.

## Přihlášení a role

1. Po otevření aplikace se uživatel musí **přihlásit nebo registrovat**. Nepřihlášený uživatel vidí pouze veřejnou úvodní stránku s tlačítky *Přihlásit se* a *Vytvořit účet*.
2. Po registraci má uživatel vždy výchozí roli **GUEST**.
3. **Admin** spravuje seznam uživatelů, jejich aktivaci a přiřazování rolí (Role = GUEST, STUDENT, TEACHER, ADMIN).
4. Uživatel může být aktivní nebo zablokováný (**isActive**).
5. Každá role má definovaný přístup pouze ke své části systému.

## Popis rolí

### Guest (GUEST)

- Výchozí role po registraci.
- Navigace pro GUEST zobrazuje pouze **přehled zápisů**.
- GUEST se nemůže zapisovat ani upravovat data.

### Student (STUDENT)

- Vidí **dashboard** s dostupnými zápisovými obdobími (**EnrollmentWindow**).
- Pokud má zápis stav **OPEN**, může:
  - **zapsat se** na výskyt předmětu (**SubjectOccurrence**),
  - **odhlásit se** ze svého zápisu.
- Omezení implementovaná v UI:
  - v rámci jednoho **bloku** (**Block**) může student **nejvýše jeden aktivní zápis**,
  - pokud je stejný předmět (**Subject**) nabízen ve více blocích, může být zapsán pouze do jednoho z nich.
- Vidí obsazenost výskytů (např. 7/30).
- Může zobrazit detail předmětu a jeho syllabus.

### Teacher (TEACHER)

- Má přístup k sekci **Předměty**.
- Může vytvářet a upravovat **předměty** (**Subject**).
- Vidí zápisy (**EnrollmentWindow**) a jejich bloky, ale **nemůže se zapisovat**.
- Vidí obsazenost výskytů (např. 7/30) a může otevřít dialog se seznamem zapsaných studentů.

### Admin (ADMIN)

- Vidí v navigaci všechny sekce aplikace:
  - **Dashboard**
  - **Zápisy**
  - **Předměty**
  - **Uživatelé**
  - **Nastavení** (základní informace)
- Může spravovat role a aktivaci uživatelů.
- Může vytvářet, upravovat **předměty**, **bloky**, **výskyty** i **zápisu**.
- Může **spouštět a ukončovat zápisy** (mění Status na OPEN nebo CLOSED).
- Může **zapisovat studenty ručně**, nebo je ze zápisu odstranit.
- Může dělat **exporty dat** ze všech seznamů.
- Má přístup k auditním údajům (`createdById`, `updatedById`, `deletedById`).

## Entity a datové typy

### Datové typy

**Uživatel (User)** Reprezentuje uživatele systému (student, učitel, admin nebo guest).

Název	Typ	Popis
<code>id</code>	<code>string</code>	Jedinečný identifikátor uživatele
<code>firstName</code>	<code>string</code>	Křestní jméno
<code>lastName</code>	<code>string</code>	Příjmení
<code>email</code>	<code>string</code>	E-mailová adresa (musí být školní)
<code>passwordHash</code>	<code>string   null</code>	Hash hesla (může být <code>null</code> , pokud používá SSO nebo nebylo nastaveno)
<code>role</code>	<code>Role</code>	Role uživatele
<code>isActive</code>	<code>boolean</code>	Indikuje, zda je účet aktivní
<code>lastLoginAt</code>	<code>Date?</code>	Datum posledního přihlášení
<code>createdAt</code>	<code>Date</code>	Datum vytvoření záznamu
<code>updatedAt</code>	<code>Date</code>	Datum poslední aktualizace

Role	Popis
<code>GUEST</code>	nově registrovaný uživatel čekající na schválení
<code>STUDENT</code>	student, který se zapisuje na předměty
<code>TEACHER</code>	vyučující, který spravuje předměty a vidí své studenty
<code>ADMIN</code>	správce systému s plnými oprávněními

### Zápis (EnrollmentWindow)

- Zápis představuje **časové období**, během kterého mohou studenti vybírat předměty.
- Každý zápis obsahuje:
  - název, popis a stav (`Status`),
  - časové rozmezí (`startsAt → endsAt`),
  - viditelnost pro studenty (`visibleToStudents`),
  - seznam bloků (`Block`).

Název	Typ	Popis
<code>id</code>	<code>string</code>	Jedinečný identifikátor zápisu
<code>name</code>	<code>string</code>	Název zápisu (např. „Zápis LS 2025“)
<code>description</code>	<code>string?</code>	Volitelný popis nebo poznámka
<code>status</code>	<code>Status</code>	Stav zápisu (DRAFT, SCHEDULED, OPEN, CLOSED)
<code>startsAt</code>	<code>Date</code>	Datum a čas začátku zápisu
<code>endsAt</code>	<code>Date</code>	Datum a čas ukončení zápisu
<code>visibleToStudents</code>	<code>boolean</code>	Určuje, zda zápis vidí studenti

Název	Typ	Popis
createdById	string	ID uživatele, který zápis vytvořil
updatedById	string?	ID uživatele, který zápis naposledy upravil
createdAt	Date	Datum vytvoření záznamu
updatedAt	Date	Datum poslední aktualizace

Status	Popis
DRAFT	návrh zápisu, zatím neaktivní
SCHEDULED	naplánovaný zápis, čeká na začátek
OPEN	zápis je aktivní, studenti se mohou zapisovat
CLOSED	zápis je uzavřený, pouze k nahlédnutí

### Bloky (Block)

- Blok představuje **logickou skupinu výskytů předmětů** v rámci jednoho zápisu. Například: *Blok 1 – povinné, Blok 2 – volitelné*.
- Každý blok:
  - patří právě jednomu zápisu (`enrollmentWindowId`),
  - má pořadí (`order`), které určuje jeho pozici ve výpisu,
  - může mít popis (`description`),
  - může být smazán (soft delete).
- Student se musí zapsat **právě na jeden výskyt** (`SubjectOccurrence`) v každém bloku.
- Bloky jsou zobrazovány studentům podle pořadí.

Název	Typ	Popis
<code>id</code>	string	Jedinečný identifikátor bloku
<code>name</code>	string	Název bloku (např. „Blok 1 – povinné“)
<code>order</code>	number	Pořadí bloku ve výpisu
<code>description</code>	string?	Volitelný popis
<code>enrollmentWindowId</code>	string	ID zápisu, do kterého blok patří
<code>createdById</code>	string	ID uživatele, který blok vytvořil
<code>updatedById</code>	string?	ID uživatele, který blok naposledy upravil
<code>createdAt</code>	Date	Datum vytvoření záznamu
<code>updatedAt</code>	Date	Datum poslední aktualizace
<code>deletedAt</code>	Date?	Datum smazání (soft delete)
<code>deletedById</code>	string?	ID uživatele, který blok smazal

### Předměty (Subject) a výskyty (SubjectOccurrence)

**Předmět (Subject)** je obecná definice kurzu — obsahuje název, syllabus a autora.

Název	Typ	Popis
<code>id</code>	<code>string</code>	Jedinečný identifikátor předmětu
<code>name</code>	<code>string</code>	Název předmětu
<code>code</code>	<code>string?</code>	Volitelný kód předmětu (např. INF101)
<code>syllabus</code>	<code>string</code>	Popis obsahu a cílů předmětu
<code>createdById</code>	<code>string</code>	ID uživatele, který předmět vytvořil
<code>updatedById</code>	<code>string?</code>	ID uživatele, který předmět naposledy upravil
<code>createdAt</code>	<code>Date</code>	Datum vytvoření záznamu
<code>updatedAt</code>	<code>Date</code>	Datum poslední aktualizace

**Výskyt (SubjectOccurrence)** představuje konkrétní instanci předmětu v určitém bloku:

- Každý výskyt má svého učitele (`teacherId`), kapacitu a kód skupiny (např. „A“, „B“, „C“).
- Pokud je `capacity = null`, zápis je **neomezený**.
- V jednom bloku může být více výskytů stejného předmětu s různými učiteli nebo kódy skupin.
- Admin může výskytu vytvářet, upravovat i mazat během otevřeného zápisu.

Název	Typ	Popis
<code>id</code>	<code>string</code>	Jedinečný identifikátor výskytu
<code>subjectId</code>	<code>string</code>	ID původního předmětu
<code>blockId</code>	<code>string</code>	ID bloku, do kterého výskyt patří
<code>teacherId</code>	<code>string</code>	ID učitele, který výskyt vyučuje
<code>subCode</code>	<code>string?</code>	Kód skupiny (např. „A“, „B“, „C“)
<code>capacity</code>	<code>number   null</code>	Maximální počet studentů (null = neomezená kapacita)
<code>createdById</code>	<code>string</code>	ID uživatele, který výskyt vytvořil
<code>updatedById</code>	<code>string?</code>	ID uživatele, který výskyt upravil
<code>createdAt</code>	<code>Date</code>	Datum vytvoření
<code>updatedAt</code>	<code>Date</code>	Datum poslední aktualizace
<code>deletedAt</code>	<code>Date?</code>	Datum smazání (soft delete)
<code>deletedById</code>	<code>string?</code>	ID uživatele, který výskyt smazal

#### Zápis studenta (StudentEnrollment)

- Student se zapisuje na konkrétní **výskyt předmětu (SubjectOccurrence)**.
- Každý zápis obsahuje informaci o tom, kdo ho vytvořil (`createdById`) a kdy (`createdAt`).

- Odhlášení (soft delete) je možné pouze, pokud je zápis (`EnrollmentWindow`) ve stavu **OPEN**.
- Po ukončení zápisu (`Status = CLOSED`) může student pouze prohlížet své zapsané předměty.

Název	Typ	Popis
<code>id</code>	<code>string</code>	Jedinečný identifikátor zápisu
<code>studentId</code>	<code>string</code>	ID studenta (User.id)
<code>subjectOccurrenceId</code>	<code>string</code>	ID výskytu předmětu, na který je zapsán
<code>createdById</code>	<code>string</code>	ID uživatele, který zápis vytvořil
<code>updatedById</code>	<code>string?</code>	ID uživatele, který zápis upravil
<code>createdAt</code>	<code>Date</code>	Datum vytvoření
<code>updatedAt</code>	<code>Date</code>	Datum poslední aktualizace
<code>deletedAt</code>	<code>Date?</code>	Datum smazání (pokud se student odhlásil)
<code>deletedById</code>	<code>string?</code>	ID uživatele, který zápis odstranil

### Aplikační pravidla

1. Student může mít v rámci jednoho **bloku** pouze **jeden aktivní zápis**.
2. Student se **nemůže zapsat na stejný předmět** ve více blocích **jednoho zápisu**.
3. Odhlášení nebo přepsání je možné pouze ve stavu **OPEN**.

### Shrnutí vztahů

- User 1–N Subject (vytvořil)
- User 1–N SubjectOccurrence (učí)
- User 1–N StudentEnrollment (studenti se zapisují)
- EnrollmentWindow 1–N Block
- Block 1–N SubjectOccurrence
- Subject 1–N SubjectOccurrence
- SubjectOccurrence 1–N StudentEnrollment

### Front end

Toto zadání popisuje strukturu a funkčnost front-endové části aplikace postavené na Next.js a shadcn/ui.

---

#### 1. Wireframe

##### Welcome (Úvodní obrazovka) Účel:

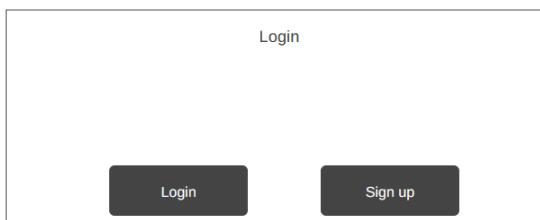
Slouží jako vstupní stránka pro nepřihlášeného uživatele. Poskytuje dvě hlavní akce: přihlášení nebo registraci.

##### Vizuální struktura:

- Centrální karta zarovnaná doprostřed obrazovky.
- Nadpis „*Login*“ (označení sekce).
- Dvě velká tlačítka:
  - **Login** – přechod na přihlašovací formulář
  - **Sign up** – přechod na registrační formulář
- Rozhraní je minimalistické, aby uživatel okamžitě pochopil, že je třeba se nejprve přihlásit.

#### Cílové chování:

- Nepřihlášený uživatel nemá přístup k žádným dalším částem systému.
- Stránka funguje jako rozcestník.



#### **Login (Přihlášení uživatele)** Účel:

Umožňuje uživateli zadat e-mail a heslo pro vstup do aplikace.

#### Vizuální struktura:

- Formulář umístěný ve středu obrazovky.
- Prvky formuláře:
  - Pole **E-mail**
  - Pole **Password**
  - Tlačítko **Login** (dominantní primární akce)
- Čisté a jednoduché rozložení pro snadný přístup.

#### Cílové chování:

- Po zadání platných údajů je uživatel přesměrován na dashboard dle své role (STUDENT/TEACHER/ADMIN).
- Chybné přihlašovací údaje zobrazí upozornění.

The diagram illustrates a simple login interface. At the top center is the word "Login". Below it is a rectangular input field labeled "E-mail". Underneath that is another rectangular input field containing the placeholder text "\*\*\*\*\*". At the bottom is a dark grey rectangular button with the word "Login" in white.

### Seznam objektů (List view – práce s entitami) Účel:

Stránka pro správu jednotlivých objektů (předmětů, uživatelů, zápisů) podle typu entity.

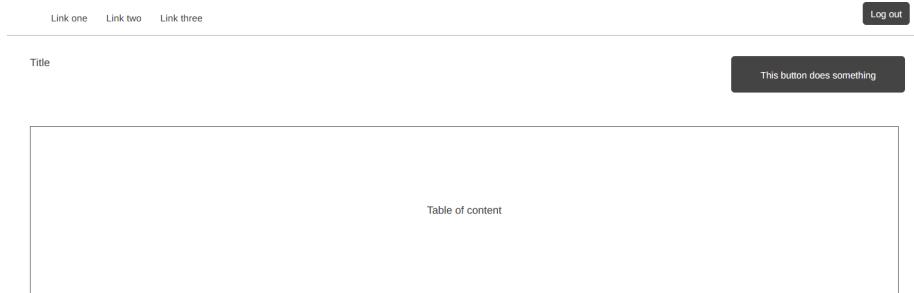
Slouží hlavně administrátorům a učitelům.

#### Vizuální struktura:

- Horní navigační lišta (Top Bar) se záložkami typu:
  - *Link one*
  - *Link two*
  - *Link three*
  - tlačítko **Log out**
- Název stránky (**Title**) vlevo nahoře.
- Primární akce vpravo nahoře (např. “**This button does something**” – typicky „Nový záznam“, „Export“ apod.).
- Velká tabulková oblast (**Table of content**) zabírá hlavní část stránky:
  - seznam všech položek (např. předmětů)
  - řazení, filtrování, akce nad řádky (editace, mazání)

#### Cílové chování:

- Seznam je centrálním místem pro operace.
- Administrátor/učitel zde může snadno spravovat velké množství dat.



**Dashboard (Hlavní stránka pro studenty)** **Účel:**  
Dashboard poskytuje studentovi přehled o aktuálním zápisovém období a dostupných blocích předmětů, do kterých se může zapsat. Ve stejném rozhraní probíhá i úprava zápisů.

**Vizuální struktura:**

- Horní navigační panel (Top Bar) s odkazy na další sekce a tlačítkem **Log out**.
- Dva hlavní informační velké boxy:
  - **Term** – základní informace o zápisu.
  - **Start / End / State** – technické údaje o začátku, konci a stavu zápisu.
- Sekce s bloky předmětů:
  - Každý blok je zobrazen jako samostatná karta s nadpisem (např. *Blok 1, Blok 2*).
  - Uvnitř každého bloku je seznam výskytů předmětů reprezentovaný jako seznam položek.
  - Struktura je responzivní — bloky se zobrazují vedle sebe nebo pod sebou podle šířky obrazovky.

**Cílové chování:**

- Student si může prohlédnout všechny dostupné bloky předmětů.
- Podle stavu zápisu (OPEN/CLOSED) jsou zobrazené akce:
  - možnost zapsat se
  - možnost odhlásit se
- Dashboard slouží jako hlavní rozhraní pro interakci studenta s celým systémem.



## 1. Strom stránek (Site Map)

Aplikace používá standardní adresářovou strukturu Next.js App Routeru.

Přihlášený uživatel vidí navigaci dle své role (ADMIN / TEACHER / STUDENT / GUEST). Nepřihlášený uživatel vidí pouze veřejnou úvodní stránku a formuláře pro přihlášení/registraci.

```
/app
  (auth) /
    login/
      page.tsx          # Přihlašovací formulář
    register/
      page.tsx          # Registrační formulář

  dashboard/
    page.tsx          # Hlavní stránka pro přihlášené (výběr zápisu)

  subjects/
    page.tsx          # Seznam všech předmětů
    [id]/
      page.tsx          # Dynamická ruta pro konkrétní předmět
      edit/
        page.tsx          # Detail předmětu
        # Editace předmětu

  enrollments/
    page.tsx          # Přehled všech zápisů
    [id]/
      page.tsx          # Detail zápisu (používá EnrollmentView stejně jako dashboard)

  users/
```

```

page.tsx          # Přehled uživatelů (přístup omezen dle role přes UI)

settings/
    page.tsx      # Základní informace o uživateli a placeholder pro nastavení

layout.tsx        # Klientský layout (AuthProvider + AppShell + AppTopbar)
globals.css       # Globální styly
page.tsx         # Veřejná úvodní stránka (Landing page)

```

## 2. Navigace (Top Bar Layout)

Aplikace používá horní navigační lištu (**Top Bar**), která se zobrazuje na všech stránkách pro přihlášené uživatele. Veřejné stránky (/ , /login, /register) navigaci nenačítají.

- **Komponenta:** AppShell (client) a AppTopbar
- **Soubor:** components/app-shell.tsx a components/app-topbar.tsx
- **Struktura Top Baru:**
  1. Vlevo – Logo / Název aplikace
  2. Uprostřed – Navigační odkazy (liší se podle role)
  3. Vpravo – Uživatelské menu

**Navigační odkazy (podle role)** Komponenta Top Baru zobrazí následující odkazy v závislosti na roli uživatele:

- **Role: ADMIN**
  - Dashboard → /dashboard
  - Zápisy → /enrollments
  - Předměty → /subjects
  - Uživatelé → /users
  - Nastavení → /settings
- **Role: TEACHER**
  - Dashboard → /dashboard
  - Zápisy → /enrollments
  - Předměty → /subjects
- **Role: STUDENT**
  - Dashboard → /dashboard
- **Role: GUEST**
  - Dashboard → /dashboard

## 3. Zadání pro programátora (Popis stránek)

**/dashboard** Tato stránka je hlavní vstupní stránkou po přihlášení.

- `/dashboard/page.tsx` je **client komponenta**
- získá přihlášeného uživatele pomocí `useAuth()`
- vybere **jeden** vhodný zápis pomocí funkce `findDashboardEnrollment(...)`
- zobrazí obsah pomocí sdílené komponenty `EnrollmentView`
- Dashboard vždy zobrazí **jeden vybraný zápis**, nikoliv selektor zápisů.

**Chování podle role** Implementace je zjednodušená — dashboard používá **stejný Layout a stejnou komponentu pro všechny role** (ADMIN, TEACHER, STUDENT, GUEST).

Rozdíly jsou pouze v tom, co jednotlivé role mohou **vidět** nebo **klikat**, ne v samotném layoutu.

Pro všechny role dashboard funguje stejně:

1. Funkce `findDashboardEnrollment` vybere nevhodnější zápis podle stavu (OPEN → SCHEDULED → DRAFT → CLOSED).
  2. Pokud zápis existuje, zobrazí se.
  3. Pokud zápis neexistuje, zobrazí se jednoduchá hláška: "Momentálně zde není žádné aktivní ani naplánované zápisové období."
- 

Globální informace o zápisu (EnrollmentHeader)

Komponenta `EnrollmentHeader` zobrazuje:

- Název zápisu
- Datum začátku a konce
- Stav zápisu (DRAFT, SCHEDULED, OPEN, CLOSED)
- Tlačítko „Upravit zápis“ pro ADMIN/TEACHER (otevírá dialog `EditEnrollmentDialog`)

Přehled bloků (EnrollmentBlocks)

Pod hlavičkou se zobrazuje mřížka bloků pomocí `EnrollmentBlocks` v layoutu podle velikosti displeje. Každý blok je potom reprezentován komponentou `EnrollmentBlockCard`.

`EnrollmentBlockCard` obsahuje:

- název bloku
- vizuální zvýraznění vybraného výskytu (pro STUDENT)
- tabulku výskytů předmětů (SubjectOccurrence)
- akce podle role uživatele

Chování STUDENT

Student může:

- vidět obsazenost výskytů (např. 5/30 nebo 2/∞),
- zapsat se nebo odhlásit, pokud:

- zápis má stav **OPEN**,
- není již zapsán v jiném výskytu téhož bloku,
- není zapsán na stejný předmět v jiném bloku.

#### Chování TEACHER a ADMIN

- Vidí všechny výskytty předmětů v daném bloku.
- Vidí jméno učitele a aktuální obsazenost.
- Kliknutím na obsazenost se otevře **OccurrencesStudentsDialog**.
- Tlačítka pro zápis jsou **neaktivní (disabled)**.

ADMIN navíc může otevřít dialog pro úpravu výskytu.

Tabulka výskytů — sloupce

Sloupec	Popis
<b>Předmět</b>	Název předmětu (klik vede na <code>/subjects/[id]</code> )
<b>Učitel</b>	Jméno vyučujícího
<b>Obsazenost</b>	Např. 7/30 (pro TEACHER/ADMIN interaktivní)
<b>Akce</b>	STUDENT: Zapsat/Odhlásit, ostatní role: disabled tlačítka

Tabulka je založena na komponentě **DataTable** s vlastním setem sloupců.

Zápis LS 2025				
Vyberte si jeden seminář v každém bloku.				
<b>Blok 1 – povinné předměty</b>	<b>Blok 2 – volitelné předměty</b>			
Předmět <b>UML v praxi</b> <b>Webové aplikace</b> <b>Database systémy</b>	Kód výskytu Učitel Obsaz. Akce  <b>UML v praxi</b> Tomáš Programátor 1/20 Studenti Upravit Smazat	Kód výskytu Učitel Obsaz. Akce  <b>Základy umělé inteligence</b> <b>Počítačové sítě</b> <b>Informativní bezpečnost</b> <b>Machine Learning</b> <b>Projektový management</b>	Tomáš Programátor 1/15 Studenti Upravit Smazat	Tomáš Programátor 1/10 Studenti Upravit Smazat

Zápis 2 - volitelné předměty				
<b>Blok 1 – povinné předměty</b>	<b>Blok 2 – volitelné předměty</b>			
Předmět <b>UML v praxi</b> <b>Webové aplikace</b> <b>Database systémy</b>	Kód výskytu Učitel Obsaz. Akce  <b>UML v praxi</b> Tomáš Programátor 1/25 Studenti Upravit Smazat	Kód výskytu Učitel Obsaz. Akce  <b>Základy umělé inteligence</b> <b>Počítačové sítě</b> <b>Informativní bezpečnost</b> <b>Machine Learning</b> <b>Projektový management</b>	Tomáš Programátor 1/15 Studenti Upravit Smazat	Tomáš Programátor 1/10 Studenti Upravit Smazat

**Studenti – UML v praxi (A)**

Počet zapsaných: 3

**Ruční zapsání studenta**

Lucie Procházková (lucie@student.cz) ▼ Zapsat

**Samuel Student**  
samuel@student.cz  
zapsán: 17. 11. 2025 19:13:54 Odepsat

**Petra Nováková**  
petra@student.cz  
zapsán: 17. 11. 2025 20:07:49 Odepsat

**Martin Horák**  
martin@student.cz  
zapsán: 17. 11. 2025 20:07:50 Odepsat

Zavřít

/subjects — Seznam předmětů Stránka /subjects slouží k přehledu všech předmětů.

**Obsah stránky** Stránka obsahuje:

- nadpis a stručný popis,
- komponentu **DataTable** se seznamem předmětů,
- nástroje pro vyhledávání a filtrování,
- tlačítko pro vytvoření nového předmětu.

Předměty					
Sdílení předmětu a jejich kódů.					
<a href="#">Nový předmět</a>					
Předmět	Kód	Vytvořil	Vyhvozen	Upravil	Aktualizován
UML v praxi	INF101	Tereza Teacherová	17.11.2025 19:13:54	—	17.11.2025 19:13:54
Webové aplikace	WEB201	Tomáš Programátor	17.11.2025 19:13:54	—	17.11.2025 19:13:54
Database systémy	DB301	Veronika Dědčková	17.11.2025 19:13:54	—	17.11.2025 19:13:54
Základy umělé inteligence	AI401	Tomáš Programátor	17.11.2025 19:13:54	—	17.11.2025 19:13:54
Podložové sítě	NET202	Tereza Teacherová	17.11.2025 19:13:54	—	17.11.2025 19:13:54
Informatická bezpečnost	SEC201	Veronika Dědčková	17.11.2025 19:13:54	—	17.11.2025 19:13:54
Machine Learning	ML501	Tomáš Programátor	17.11.2025 19:13:54	—	17.11.2025 19:13:54
Projektový management	PM101	Tereza Teacherová	17.11.2025 19:13:54	—	17.11.2025 19:13:54

**/subjects/[id] — Detail a editace předmětu** Stránka předmětu má dva režimy:

1. **Zobrazení detailu** — dostupné pro všechny přihlášené role
2. **Editace** — dostupná pro role **TEACHER** a **ADMIN**

Následující popis odpovídá skutečné implementaci.

---

**/subjects/[id]/page.tsx — Režim zobrazení** Stránka zobrazuje kompletní informace o vybraném předmětu (**Subject**) ve více sekčích.

Zobrazované údaje:

- Název předmětu
- Kód předmětu
- Krátký popis (**description**)
- Syllabus (**syllabus**)
- Výskytu předmětu (**SubjectOccurrence**)

Pod základními informacemi je tabulka všech výskytů daného předmětu napříč zápisy a bloky.

Tabulka zobrazuje sloupce:

- **Zápis** (název **EnrollmentWindow**)
- **Blok** (název **Block**)
- **Skupina** (subCode)
- **Vyučující**
- **Kapacita**
- **Obsazenost**

Tabulka je postavená pomocí komponenty **DataTable**.

---

Role **TEACHER/ADMIN** mají v pravé horní části tlačítko „**Upravit**“, které vede na **/subjects/[id]/edit**.

**UML v praxi** INF101  
Základy UML, příklady, cvičení.  
Vytvořil Tereza Teacherová dne 17. 11. 2023 19:13:54  
Poslední uprava — Praha 17-11-2023 19:13:54

**Syntaxis**  
Základy UML, příklady, cvičení.

Předmět	Zápis	Stav	Blok	Kód výskytu	Učitel	Obsaz.	Akce
UML v praxi	Zápis LS 2025	Otevřeno	Blok 1 – povinné předměty	INF101/A	Tereza Teacherová	3/25	<a href="#">Studenti</a> <a href="#">Upravit</a> <a href="#">Smazat</a>
UML v praxi	Zápis LS 2025	Otevřeno	Blok 2 – volitelné předměty	INF101/B	Tereza Teacherová	0/25	<a href="#">Studenti</a> <a href="#">Upravit</a> <a href="#">Smazat</a>

Rádce na stránce: 10 Strana 1 z 1 • 2 záznamů [Předchozí](#) [Další](#)

**/subjects/[id]/edit/page.tsx — Režim editace** Stránka umožňuje upravit základní informace o předmětu. Je dostupná pro role **TEACHER** a **ADMIN**.

Editační formulář obsahuje:

- **Input** — název předmětu (`name`)
- **Input** — kód předmětu (`code`)
- **Textarea** — krátký popis (`description`)
- **Rich Text Editor (Tiptap)** — detailní popis (`syllabus`)
  - podpora formátování (nadpisy, tučné, kurziva, seznamy)

Akce tlačítok

Stránka obsahuje následující akce:

- **Uložit**
  - Aktualizuje hodnoty předmětu v paměti
  - Zobrazí toast o úspěšném uložení
  - Přesměruje zpět na detail (`/subjects/[id]`)
- **Zrušit**
  - Přesměruje zpět bez uložení
- **Smazat předmět**
  - V aktuální verzi není implementováno (tlačítko se nezobrazuje)

The screenshot shows a web-based application interface for managing subjects. At the top, there's a navigation bar with links like 'Zápis seminářů', 'Dashboard', 'Zápis', 'Předměty' (selected), 'Uživatelé', and 'Nastavení'. On the right, it shows the email 'admin@skola.cz' and a 'Odhlásit' (Logout) button. The main content area is titled 'Upravit předmět' (Edit Subject) with a note 'Údaje v práci' (Data in work). It has four input fields: 'Název' (Name) containing 'UML v praxi', 'Kód' (Code) containing 'INF 101', 'Popis' (Description) containing 'Základy UML, příklady, cvičení.', and a large 'Sylabus' (Syllabus) field containing the text 'Základy UML, příklady, cvičení.'. Below the syllabus field is a toolbar with icons for bold, italic, strike, code, clear mark, clear node, paragraph, H1, H2, H3, bullet list, ordered list, code block, blockquote, horizontal rule, hard break, undo, and redo. At the bottom of the form are two buttons: 'Uložit' (Save) and 'Zrušit' (Cancel).

**/enrollments** Stránka **/enrollments** slouží k přehledu zápisových období (**EnrollmentWindow**). Je určena pro role **ADMIN** a **TEACHER**, které ji mají dostupnou v navigaci.

---

**Funkce stránky** Stránka zobrazuje tabulkou zápisů s informacemi o:

- názvu a stavu zápisu,
- viditelnosti pro studenty,
- termínu začátku a konce,
- počtu bloků a počtu předmětů v blocích,
- počtu zapsaných studentů,
- počtu studentů, kteří mají zápis kompletně vyplněný (mají zapsaný předmět ve všech blocích).

Používá se komponenta **DataTable** s vyhledáváním, filtrováním a tříděním na straně klienta.

---

**Ovládací prvky** V horní části stránky jsou:

- **Nadpis a popis:**
  - **Zápisová období**
  - krátký popis („Přehled všech zápisů, bloků a počtu unikátních studentů.“)
- **Tlačítko „Vytvořit nový zápis“**
  - zobrazuje se pouze pro roli **ADMIN**
  - otevře dialog pro zadání názvu, popisu, stavu, časového rozmezí a viditelnosti zápisu

Pod hlavičkou je komponenta **DataTable** s těmito funkcemi:

- **Vyhledávání:**

- `searchPlaceholder="Hledat podle názvu."`
- fulltext vyhledává v názvu zápisu

- **Filtry:**

- **Select „Stav“**
    - \* hodnoty: Koncept (DRAFT), Naplánováno (SCHEDULED), Otevřeno (OPEN), Uzavřeno (CLOSED)
  - **Select „Viditelnost“**
    - \* „Viditelné studentům“ (`visibleToStudents = true`)
    - \* „Skryté studentům“ (`visibleToStudents = false`)
  - **Datumové filtry:**
    - \* Začátek – filtr podle `startsAt`
    - \* Konec – filtr podle `endsAt`
- 

**Sloupce tabulky** Tabulka obsahuje následující sloupce:

Sloupec	Popis
<b>Název</b>	Název zápisu. Kliknutím na název se otevře stránka <code>/enrollments/[id]</code> . Pod ním je zobrazeno hodnota „Ano/Ne“.
<b>Stav</b>	Zobrazen jako barevný Badge (Koncept, Naplánováno, Otevřeno, Uzavřeno).
<b>Viditelné pro studenty</b>	Hodnota „Ano/Ne“ zobrazená jako Badge.
<b>Začátek</b>	Datum a čas začátku zápisu ( <code>startsAt</code> ).
<b>Konec</b>	Datum a čas konce zápisu ( <code>endsAt</code> ).
<b>Bloky (předměty)</b>	Seznam bloků s počtem výskytů v každém bloku (např. „Blok 1 [3]“).
<b>Zapsaní studenti</b>	Počet unikátních studentů zapsaných v rámci zápisu.
<b>Kompletně zapsaní</b>	Počet studentů, kteří mají zapsán předmět ve všech blocích daného zápisu.
<b>Akce</b>	Kontextové tlačítko pro úpravu (podle role).

---

**Práva a akce podle role** Role ADMIN

- Vidí všechna zápisová období v tabulce.
- V hlavičce má k dispozici tlačítko „**Vytvořit nový zápis**“, které:
  - otevře dialog pro vytvoření zápisu,
  - umožní nastavit název, popis, stav, časové rozmezí a viditelnost.
- Ve sloupci **Akce** má k dispozici tlačítko:
  - „**Upravit zápis**“
    - \* otevře dialog pro úpravu vybraného zápisu
    - \* po uložení se dialog zavře a stránka se obnoví

## Role TEACHER

- Vidí stejnou tabulkou zápisů jako ADMIN (včetně filtrů a statistik).
- **Nevidí** tlačítko „Vytvořit nový zápis“.
- Ve sloupci **Akce** se tlačítko „Upravit zápis“ nezobrazuje.

Název	Stav	Viditelné pro studenty	Začátek	Konec	Blocks ( предметы )	Zapsaní studentů	Kompletně zapsaní	Alice
Zápis LS 2023 <small>Výberete si jeden seminář v každém bloku.</small>	Očekávám	Ava	17.11.2025 19:13:54	24.11.2025 19:13:54	Block 1 – povinné předměty: 3 Block 2 – volitelné předměty: 8	5	0	<a href="#">Upravit zápis</a>
Zápis LS 2023 <small>Výberete si jeden seminář v každém bloku.</small>	Očekávám	Ava	17.11.2025 19:13:54	24.11.2025 19:13:54	Block 2 – volitelné předměty: 8	0	0	<a href="#">Upravit zápis</a>
Zápis LS 2023 <small>Výberete si jeden seminář v každém bloku.</small>	Očekávám	Ava	17.11.2025 19:13:54	24.11.2025 19:13:54	Block 1 – povinné předměty: 3 Block 2 – volitelné předměty: 8	0	0	<a href="#">Upravit zápis</a>

**/enrollments/[id]** Stránka **/enrollments/[id]** zobrazuje detail jednoho zápisového období (**EnrollmentWindow**). Stránka znova využívá **stejné komponenty jako dashboard**.

Je dostupná pro role, které mají odkaz v navigaci ( **ADMIN** a **TEACHER** ).

**/users** Stránka **/users** slouží k přehledu a správě uživatelů.

**Obsah stránky users** Stránka **/users** obsahuje:

- nadpis a stručný popis,
- komponentu **DataTable** se seznamem uživatelů,
- nástroje pro vyhledávání, filtrování a hromadné akce,
- akční menu pro úpravu jednoho konkrétního uživatele.

## Načítání dat

- Načítají se **všichni uživatelé** z aktuálního datasetu.
- Vyhledávání, filtrování, třídění a výběr probíhá **na klientu** (bez serverových volání).

**Ovládací prvky uživatelů** Nad tabulkou jsou dostupné tyto prvky:

- **Fulltext vyhledávání** v `firstName`, `lastName`, `email`.
- **Filttry** podle role, stavu, datumu vytvoření nebo datumu posledního přihlášení

**Sloupce tabulky uživatelů** Tabulka obsahuje následující sloupce:

Sloupec	Popis
<b>Jméno</b>	Kombinace jména a příjmení
<b>E-mail</b>	E-mail uživatele
<b>Role</b>	Barevný Badge s hodnotou role
<b>Stav</b>	Badge „Aktivní“ / „Neaktivní“
<b>Vytvořen</b>	Datum vytvoření uživatele
<b>Poslední přihlášení</b>	Datum posledního přihlášení

**Hromadné akce** Tabulka nabízí vedle filtrů i možnost hromadných změn, kdy se akce provedou nad všemi aktuálně vyfiltrovanými záznamy.

- **Změna role** — dropdown pro výběr nové role
- **Aktivovat vybrané**
- **Deaktivovat vybrané**

**Akce v řádku** V každém řádku je kontextové menu (**DropdownMenu**) pro změnu role a přepínač pro aktivování/deaktivování uživatelů:

Detaile uživatele se nezobrazují na vlastní stránce — vše je řešeno přímo v tabulce pomocí inline akcí a hromadného panelu.

Filter podle jména nebo e-mailu...	E-mail	Role	Aktivní	Vyvořen	Poslední přihlášení
Adam Admin	admin@skola.cz	ADMIN	<input checked="" type="checkbox"/>	17. 11. 2025 20:09:58	—
Alice Adminová	alice.admin@skola.cz	ADMIN	<input checked="" type="checkbox"/>	17. 11. 2025 20:09:58	—
Robert Ředitel	reditel@skola.cz	ADMIN	<input checked="" type="checkbox"/>	17. 11. 2025 20:09:58	—
Tereza Techervá	tereza@skola.cz	TEACHER	<input checked="" type="checkbox"/>	17. 11. 2025 20:09:58	—
Tomáš Programátor	tomas.programator@skola.cz	TEACHER	<input checked="" type="checkbox"/>	17. 11. 2025 20:09:58	—
Veronika Dědčová	veronika.dedcová@skola.cz	TEACHER	<input checked="" type="checkbox"/>	17. 11. 2025 20:09:58	—
Samuel Student	samuel@student.cz	STUDENT	<input checked="" type="checkbox"/>	17. 11. 2025 20:09:58	—
Petra Nováková	petra@student.cz	STUDENT	<input checked="" type="checkbox"/>	17. 11. 2025 20:09:58	—
Martin Horák	martin@student.cz	STUDENT	<input checked="" type="checkbox"/>	17. 11. 2025 20:09:58	—
Lucie Procházková	lucie@student.cz	STUDENT	<input checked="" type="checkbox"/>	17. 11. 2025 20:09:58	—

**/settings** Stránka je dostupná **pouze pro ADMINA**. Na začátku page.tsx je nutné ověřit roli, jinak **redirect**. AKtuálně jsou všechny nastavení napevno v kódu, ale při nasazení by byly jednotlivé zadávací pole pro texty níže.

### Komponenty:

- **<Tabs>**: Hlavní navigace stránky.
  - **Tab 1: "Obecné"**

- \* **Karta "Role":**
  - CardHeader: "Výchozí role uživatelů"
  - CardContent: Obsahuje Select s popiskem "Role pro nově schválené uživatele".
  - Možnosti: STUDENT, TEACHER. (Určuje, jakou roli získá GUEST poté, co ho admin "schválí" na stránce /users).
  - CardFooter: Button ("Uložit").
- \* **Karta "Registrace":**
  - CardHeader: "Omezení registrace"
  - CardContent: Input s popiskem "Povolené e-mailové domény (oddělte čárkou)".
  - CardDescription: "Např: @skola.cz. Pokud je prázdné, registrace je povolena pro jakýkoliv e-mail."
  - CardFooter: Button ("Uložit").
- **Tab 2: "Texty"**
  - \* **Karta "Text pro GUEST":**
    - CardHeader: "Text na Dashboardu (Role GUEST)"
    - CardContent: Obsahuje Textarea pro úpravu textu, který vidí uživatel s rolí GUEST.
    - CardFooter: Button ("Uložit").
  - \* **Karta "Text pro 'Žádný zápis'":**
    - CardHeader: "Text na Dashboardu (Žádný zápis)"
    - CardContent: Obsahuje Textarea pro úpravu textu, který vidí přihlášený uživatel, pokud není aktivní žádný EnrollmentWindow.
    - CardFooter: Button ("Uložit").
- **Tab 3: "Pokročilé" (Prázdná pro budoucí použití)**

## Lighthouse Report

Tato kapitola shrnuje výsledky automatizovaného auditu pomocí nástroje **Google Lighthouse**. Audit byl proveden nad stránkou **/dashboard** v produkční verzi aplikace [\[https://seminar-is.vercel.app\]\(https://seminar-is.vercel.app\)](https://seminar-is.vercel.app).

### Výsledné skóre

Kategorie	Skóre
Performance	100 / 100
Accessibility	95 / 100
Best Practices	100 / 100
SEO	100 / 100

---

## Performance (100 %)

Aplikace dosáhla maximálního skóre díky velmi rychlému vykreslení:

- **First Contentful Paint:** 0.2 s :contentReference[oaicite:4]{index=4}
- **Largest Contentful Paint:** 0.5 s :contentReference[oaicite:5]{index=5}
- **Speed Index:** 0.6 s :contentReference[oaicite:6]{index=6}
- **Total Blocking Time:** 0 ms :contentReference[oaicite:7]{index=7}
- **Cumulative Layout Shift:** 0 :contentReference[oaicite:8]{index=8}

Výborný výkon je dosažen kombinací:

- rychlého renderingu díky Next.js server components,
  - minimální velikosti bundle (cca 247 KiB) :contentReference[oaicite:9]{index=9},
  - žádných blokujících skriptů ani přesměrování.
- 

## Accessibility (95 %)

Skóre přístupnosti je velmi vysoké. Lighthouse upozornil pouze na:

- **jeden problém s kontrastem textu** na tlačítkách (červená šedá)
- několik doporučení k manuálnímu ověření (fokus, pořadí tab indexu atd.)

Nic z toho zásadně nebrání použití aplikace — jde o drobná doporučení.

---

## Best Practices (100 %)

Aplikace splňuje všechny moderní vývojové standardy:

- běží kompletně přes HTTPS,
- nepoužívá zastaralé API,
- žádné chyby v konzoli,
- správná bezpečnostní nastavení,
- vhodné načítání zdrojů.

Celá sekce prošla bez jediného varování.

---

## SEO (100 %)

Přestože je aplikace interní, Lighthouse potvrzuje, že:

- všechny stránky mají validní HTML,
- stránka má meta description, viewport a ostatní náležitosti,
- neobsahuje žádné indexační chyby.

SEO je plně optimalizované — skóre 100 / 100.

---

## Shrnutí

Aplikace podle Lighthouse dosahuje vynikajících výsledků:

- **maximální výkon i best practices,**
- **velmi dobrá přístupnost,**
- **vynikající technická čistota a optimalizace.**

Identifikované drobnosti (kontrast textu) lze snadno dolahit v budoucí verzi. Celkově systém splňuje standardy moderní webové aplikace.

## ESLint

Pro statickou analýzu zdrojového kódu na straně frontendu je v projektu použit nástroj **ESLint** (integrovaný pomocí `next lint`).

ESLint slouží k automatickému odhalování chyb a nekonzistence v kódu ještě před jeho spuštěním. Pomáhá zvyšovat čitelnost, udržovatelnost a celkovou kvalitu projektu. Používá sadu pravidel zaměřených na správné používání TypeScriptu, Reactu a doporučených praktik Next.js.

Analýza je spouštěna příkazem:

```
npm run lint
```

### Detekované problémy a jejich význam

ESLint odhalil několik hlavních kategorií problémů, které se v projektu opakují:

---

#### 1. Typování pomocí `any`

**Chyby typu:**

Error: Unexpected `any`. Specify a different `type`.

**Výskyt např. v:**

- `app/(auth)/login/page.tsx`
- `app/enrollments/page.tsx`
- `app/subjects/[id]/page.tsx`
- `components/ui/data-table.tsx`
- `components/blocks/EnrollmentBlockCard.tsx`

**Význam:** Použití `any` snižuje typovou bezpečnost. TypeScript tím ztrácí schopnost odhalovat chyby v hodnotách na vstupu i výstupu funkcí.

**Návrh řešení:**

- Nahrazovat `any` konkrétními typy (např. `User`, `Subject`, `EnrollmentWindow`).
- U tabulek a sloupců využít generické typy (`TData`, `TValue`).
- Pokud má být `any` jen dočasné, lze ho přepsat alespoň na `unknown`.

---

## 2. Nepoužité proměnné (no-unused-vars)

Chyby typu:

Error: '**b**' **is** defined but **never** used.

Error: '**isPublicPage**' **is** assigned a value but **never** used.

Výskyt např. v:

- dashboard/page.tsx
- components/app-shell.tsx
- components/auth/auth-provider.tsx
- lib/data.ts
- components/occurrences/EditSubjectOccurrenceDialog.tsx

**Význam:** Nepoužité proměnné zhoršují čitelnost a často ukazují na nedokončený nebo "mrtvý" kód.

Návrh řešení:

- Odebrat nepoužívané proměnné a importy.
  - Pokud je potřeba proměnnou ponechat, přejmenovat ji na `_name` a upravit ESLint, aby ignoroval podtržítko.
- 

## 3. Nesprávné použití React Hooks

Chyby typu:

React Hook "useState" is called conditionally.

React Hook "useMemo" is called conditionally.

React Hook is called in function that is not a component.

Výskyt např. v:

- app/enrollments/page.tsx
- subjects/[id]/page.tsx
- components/users/columns.tsx
- components/occurrences/OccurrencesStudentsDialog.tsx

**Význam:** React Hooks musí být volány vždy ve stejném pořadí. Pokud jsou uvnitř `if`, `for`, nebo v obyčejných funkcích, může to vést k runtime chybám a nekonzistentnímu stavu komponent.

Návrh řešení:

- Přesunout hooky do horní úrovně komponent.
- U tabulkových „cell“ rendererů vytvořit malé komponenty (např. `<RoleCell />`, `<ActionsCell />`) a hooky volat uvnitř nich.
- Vyhnut se podmíněnému volání hooků.

---

## 4. Neescapované uvozovky v JSX

Chyby typu:

Error: '""' can be escaped with ";

Výskyt např. v:

- `EditEnrollmentDialog.tsx`

Návrh řešení:

- Nahradit " za ";
  - Nebo přepsat text tak, aby uvozovky nebyly v běžném JSX literálu.
- 

## Shrnutí přínosů ESLintu

ESLint významně přispěl k odhalení:

- Nadměrného používání `any`.
- Nepoužívaného a „mrtvého“ kódu.
- Potenciálně nebezpečných vzorů v práci s React Hooks.
- Drobných problémů v syntaxi JSX.

Díky těmto výsledkům lze v dalších iteracích projektu provést úpravy, které zvýší bezpečnost, přehlednost a udržovatelnost celého systému. ESLint tak slouží jako automatizovaný nástroj kvality, srovnatelný s průběžným code review.

## Zadavací dokumentace

Tento dokument je možné použít jako **zadavací dokumentace** celého systému. Obsahuje kompletní popis všech funkčních požadavků, uživatelských rolí, toků aplikace, pravidel zápisu, struktury stránek i způsobu zpracování dat.

Dokumentace je strukturována tak, aby ji bylo možné přímo použít jako:

- **zadání pro vývojáře**,
- **specifikaci pro zadavatele**,
- **referenční popis chování hotového systému**,
- **podklad pro prezentaci nebo schválení projektu**.

Všechny kapitoly společně tvoří ucelený přehled, který pokrývá hlavní funkční i nefunkční požadavky. Na základě této dokumentace je možné nezávisle systém navrhnout, implementovat nebo dále rozšiřovat.

Systém je popsán do takové hloubky, že dokumentace může být využita i jako finální **předávací dokument**, protože přesně popisuje aktuálně implementované funkcionality, uživatelské rozhraní i technické komponenty.