

# Dokumentace semestrálního projektu - aplikace pro správu reportů

## Úvod

Aplikace pro správu reportů a dokumentace je důležitým nástrojem pro globální společnost, která provozuje rozsáhlou síť poboček a závodů napříč státy.

Jejím cílem je zajištění neustálého přehledu o aktuálním dění, stejně jako o řešených či nevyřešených problémech ve všech jejích jednotkách.

Tento systém je nezbytný pro vedení společnosti a další vedoucí pracovníky, aby mohlo být plně informováno o situaci v reálném čase.

Aplikace je navržena tak, aby efektivně shromažďovala a zpřístupňovala relevantní informace jak manažerskému týmu, tak i vybraným zaměstnancům. Tvoří ji několik klíčových komponent, přičemž nejdůležitějšími jsou moduly pro správu zpráv a profily uživatelů, kteří tyto zprávy generují.

## Technologie (Tech Stack)

**TypeScript** - Programovací jazyk pro backend a frontend aplikace

**NextJS 14** - Full-stack react framework pro naši aplikaci (pouze používaná frontend funkcionalita s oddělaným backendem)

**MUI** - Material UI knihovna pro React

**React Query** - React knihovna pro komunikaci s REST API backendem

**NestJS** - Node.js MVC (Model-view-controller) framework pro backend aplikace

**Oracle SQL Database** - Databázové řešení

## Popis entit

### 1. dokument

- **Popis:** Uchovává informace o dokumentech spojených se zprávami.
- **Atributy:**
  - `dokument_id`: Primární klíč
  - `nazev`: Název dokumentu
  - `soubor`: Obsah dokumentu jako BLOB
  - `typ_obsahu`: Typ obsahu dokumentu
  - `zprava_id_zprava`: Cizí klíč na zprávu

### 2. fyzicka\_schuzka

- **Popis:** Uchovává informace o fyzických schůzkách.
- **Atributy:**
  - `id_schuzka`: Primární klíč

- **umisteni:** Umístění schůzky

### 3. kontakt

- **Popis:** Uchovává kontaktní informace.
- **Atributy:**
  - **id\_kontakt:** Primární klíč
  - **jmeno:** Jméno kontaktu
  - **stredni\_jmeno:** Střední jméno kontaktu
  - **prijmeni:** Příjmení kontaktu
  - **telefonni\_cislo:** Telefonní číslo
  - **email:** Email
  - **popis:** Popis kontaktu
  - **zprava\_id\_zprava:** Cizí klíč na zprávu

### 4. online\_schuzka

- **Popis:** Uchovává informace o online schůzkách.
- **Atributy:**
  - **id\_schuzka:** Primární klíč
  - **odkaz:** Odkaz na schůzku

### 5. opatreni

- **Popis:** Uchovává informace o opatřeních.
- **Atributy:**
  - **opatreni\_id:** Primární klíč
  - **popis:** Popis opatření
  - **zprava\_id\_zprava:** Cizí klíč na zprávu
  - **naklady:** Náklady na opatření

### 6. pobocka

- **Popis:** Uchovává informace o pobočkách.
- **Atributy:**
  - **id\_pobocka:** Primární klíč
  - **nazev:** Název pobočky
  - **umisteni:** Umístění pobočky

### 7. prava

- **Popis:** Uchovává informace o právech uživatelů.
- **Atributy:**
  - **id\_prava:** Primární klíč
  - **nazev:** Název práva

### 8. schuzka

- **Popis:** Uchovává informace o schůzkách.
- **Atributy:**
  - `id_schuzka`: Primární klíč
  - `cas_zacatek`: Začátek schůzky
  - `cas_konec`: Konec schůzky
  - `popis`: Popis schůzky
  - `zprava_id_zprava`: Cizí klíč na zprávu
  - `typ_schuzky`: Typ schůzky (fyzická/online)

## 9. `schuzka_uzivatel`

- **Popis:** Uchovává informace o vztahu mezi schůzkami a uživateli.
- **Atributy:**
  - `schuzka_id_schuzka`: Cizí klíč na schůzku
  - `uzivatel_id_uzivatel`: Cizí klíč na uživatele

## 10. `tema`

- **Popis:** Uchovává informace o tématech.
- **Atributy:**
  - `id_tema`: Primární klíč
  - `nazev`: Název tématu

## 11. `uzivatel`

- **Popis:** Uchovává informace o uživateli.
- **Atributy:**
  - `id_uzivatel`: Primární klíč
  - `jmeno`: Jméno uživatele
  - `stredni_jmeno`: Střední jméno uživatele
  - `prijmeni`: Příjmení uživatele
  - `email`: Email
  - `heslo`: Heslo
  - `telefonni_cislo`: Telefonní číslo
  - `pobocka_id_pobocka`: Cizí klíč na pobočku
  - `prava_id_prava`: Cizí klíč na práva

## 12. `zpetna_vazba`

- **Popis:** Uchovává zpětnou vazbu.
- **Atributy:**
  - `id_zpetna_vazba`: Primární klíč
  - `zpetna_vazba`: Text zpětné vazby
  - `zprava_id_zprava`: Cizí klíč na zprávu
  - `uzivatel_id_uzivatel`: Cizí klíč na uživatele

## 13. `zprava`

- **Popis:** Uchovává informace o zprávách.
- **Atributy:**
  - `id_zprava`: Primární klíč
  - `titulek`: Titulek zprávy
  - `popisek`: Popisek zprávy
  - `uzavreno`: Čas uzavření zprávy
  - `vytvoreno`: Čas vytvoření zprávy
  - `upraveno`: Čas poslední úpravy zprávy
  - `uzivatel_id_uzivatel`: Cizí klíč na uživatele
  - `tema_id_tema`: Cizí klíč na téma

## Popis procedur

### 1. NacistKontakty

- **Popis:** Načte kontakty pro zadanou zprávu.
- **Parametry:**
  - `id_zprava` (IN): ID zprávy
  - `p_cursor` (OUT): Výstupní kurzor s kontakty
- **Kód:**

```
create or replace NONEDITIONABLE PROCEDURE NacistKontakty(
    id_zprava IN INTEGER,
    p_cursor OUT SYS_REFCURSOR
) AS
BEGIN
    OPEN p_cursor FOR SELECT * FROM kontakt WHERE zprava_id_zprava =
id_zprava;
END NacistKontakty;
```

### 2. NacistOpatreni

- **Popis:** Načte opatření pro zadanou zprávu.
- **Parametry:**
  - `id_zprava` (IN): ID zprávy
  - `p_cursor` (OUT): Výstupní kurzor s opatřeními
- **Kód:**

```
create or replace NONEDITIONABLE PROCEDURE NacistOpatreni(
    id_zprava IN INTEGER,
    p_cursor OUT SYS_REFCURSOR
) AS
BEGIN
    OPEN p_cursor FOR SELECT * FROM opatreni WHERE zprava_id_zprava =
```

```
id_zprava;  
END NacistOpatreni;
```

### 3. NacistUzivatele

- **Popis:** Načte uživatele podle ID s kontrolou práv.
- **Parametry:**
  - `p_id_uzivatele` (IN): ID uživatele
  - `p_cursor` (OUT): Výstupní kurzor s uživateli
- **Kód:**

```
create or replace NONEDITIONABLE PROCEDURE NacistUzivatele(  
    p_id_uzivatele IN INTEGER,  
    p_cursor OUT SYS_REFCURSOR  
) AS  
    v_prava INTEGER;  
BEGIN  
    SELECT prava_id_prava INTO v_prava FROM uzivatel WHERE  
    uzivatel.id_uzivatel = p_id_uzivatele;  
    IF v_prava = 1 THEN  
        OPEN p_cursor FOR  
            SELECT u.id_uzivatel, u.jmeno, u.email, p.nazev AS prava  
            FROM uzivatel u  
            JOIN prava p ON u.prava_id_prava = p.id_prava;  
    ELSE  
        OPEN p_cursor FOR SELECT 'Access Denied' AS message FROM DUAL;  
    END IF;  
END NacistUzivatele;
```

### 4. NacistZpetneVazby

- **Popis:** Načte zpětné vazby pro zadanou zprávu.
- **Parametry:**
  - `id_zpravy` (IN): ID zprávy
  - `p_cursor` (OUT): Výstupní kurzor se zpětnými vazbami
- **Kód:**

```
create or replace NONEDITIONABLE PROCEDURE NacistZpetneVazby(  
    id_zpravy IN INTEGER,  
    p_cursor OUT SYS_REFCURSOR  
) AS  
BEGIN  
    OPEN p_cursor FOR SELECT * FROM zpetna_vazba INNER JOIN uzivatel ON  
    uzivatel.id_uzivatel = zpetna_vazba.uzivatel_id_uzivatel WHERE
```

```
zprava_id_zprava = id_zpravy;  
END NacistZpetneVazby;
```

## 5. NacistZpravu

- **Popis:** Načte detailní informace o zprávě podle ID.
- **Parametry:**
  - `p_id_zprava` (IN): ID zprávy
  - `p_cursor` (OUT): Výstupní kurzor s detailními informacemi
- **Kód:**

```
create or replace NONEDITIONABLE PROCEDURE NacistZpravu( p_id_zprava IN INTEGER, p_cursor OUT  
SYS_REFCURSOR ) AS BEGIN OPEN p_cursor FOR SELECT z, t, u.* FROM zprava z INNER JOIN tema t ON  
z.tema_id_tema = t.id_tema INNER JOIN uzivatel u ON z.uzivatel_id_uzivatel = u.id_uzivatel WHERE z.id_zprava =  
p_id_zprava; END NacistZpravu; ``
```

## 6. NacistZpravy

- **Popis:** Načte všechny zprávy nebo zprávy podle ID tématu.
- **Parametry:**
  - `p_id_tema` (IN): ID tématu
  - `p_cursor` (OUT): Výstupní kurzor se zprávami
- **Kód:**

```
create or replace NONEDITIONABLE PROCEDURE NacistZpravy(  
    p_id_tema IN INTEGER,  
    p_cursor OUT SYS_REFCURSOR  
) AS  
BEGIN  
    IF p_id_tema IS NULL THEN  
        OPEN p_cursor FOR SELECT * FROM zprava;  
    ELSE  
        OPEN p_cursor FOR SELECT * FROM zprava WHERE tema_id_tema =  
p_id_tema;  
    END IF;  
END NacistZpravy;
```

## 7. Prihlasit

- **Popis:** Ověří uživatele a přihlásí ho.
- **Parametry:**
  - `p_email` (IN): Email uživatele
  - `p_heslo` (IN): Heslo uživatele

- **p\_cursor** (OUT): Výstupní kurzor s uživatelskými informacemi
- **Kód:**

```
create or replace NONEDITIONABLE PROCEDURE Prihlasit(  
    p_email IN VARCHAR2,  
    p_heslo IN VARCHAR2,  
    p_cursor OUT SYS_REFCURSOR  
) AS  
BEGIN  
    OPEN p_cursor FOR  
        SELECT u.id_uzivatel, u.jmeno, u.stredni_jmeno, u.prijmeni,  
        u.email, u.telefonni_cislo, p.id_prava, p.nazev AS prava  
        FROM uzivatel u  
        JOIN prava p ON u.prava_id_prava = p.id_prava  
        WHERE u.email = p_email AND u.heslo = p_heslo;  
END Prihlasit;
```

## 8. VytvoritKontakt

- **Popis:** Vytvoří nový kontakt.
- **Parametry:**
  - **p\_jmeno** (IN): Jméno kontaktu
  - **p\_stredni\_jmeno** (IN): Střední jméno kontaktu
  - **p\_prijmeni** (IN): Příjmení kontaktu
  - **p\_telefonni\_cislo** (IN): Telefonní číslo
  - **p\_email** (IN): Email
  - **p\_popis** (IN): Popis kontaktu
  - **p\_zprava\_id** (IN): ID zprávy
- **Kód:**

```
create or replace NONEDITIONABLE PROCEDURE VytvoritKontakt(  
    p_jmeno IN VARCHAR2,  
    p_stredni_jmeno IN VARCHAR2,  
    p_prijmeni IN VARCHAR2,  
    p_telefonni_cislo IN VARCHAR2,  
    p_email IN VARCHAR2,  
    p_popis IN VARCHAR2,  
    p_zprava_id IN INTEGER  
) AS  
BEGIN  
    INSERT INTO kontakt (id_kontakt, jmeno, stredni_jmeno, prijmeni,  
        telefonni_cislo, email, popis, zprava_id_zprava)  
        VALUES (id_kontakt_seq.NEXTVAL, p_jmeno, p_stredni_jmeno, p_prijmeni,  
        p_telefonni_cislo, p_email, p_popis, p_zprava_id);  
END VytvoritKontakt;
```

## 9. VytvoritOpatreni

- **Popis:** Vytvoří nové opatření.
- **Parametry:**
  - `popis` (IN): Popis opatření
  - `zprava_id_zprava` (IN): ID zprávy
  - `naklady` (IN): Náklady na opatření
- **Kód:**

```
create or replace NONEDITIONABLE PROCEDURE VytvoritOpatreni(  
    popis IN VARCHAR2,  
    zprava_id_zprava IN INTEGER,  
    naklady IN INTEGER  
) AS  
BEGIN  
    INSERT INTO opatreni (opatreni_id, popis, zprava_id_zprava, naklady)  
    VALUES (id_opatreni_seq.NEXTVAL, popis, zprava_id_zprava, naklady);  
END VytvoritOpatreni;
```

## 10. VytvoritZpetnouVazbu

- **Popis:** Vytvoří novou zpětnou vazbu.
- **Parametry:**
  - `zpetna_vazba` (IN): Text zpětné vazby
  - `zprava_id_zprava` (IN): ID zprávy
  - `uzivatel_id_uzivatel` (IN): ID uživatele
- **Kód:**

```
create or replace NONEDITIONABLE PROCEDURE VytvoritZpetnouVazbu(  
    zpetna_vazba IN VARCHAR2,  
    zprava_id_zprava IN INTEGER,  
    uzivatel_id_uzivatel IN INTEGER  
) AS  
BEGIN  
    INSERT INTO zpetna_vazba (id_zpetna_vazba, zpetna_vazba,  
    zprava_id_zprava, uzivatel_id_uzivatel)  
    VALUES (id_zpetna_vazba_seq.NEXTVAL, zpetna_vazba, zprava_id_zprava,  
    uzivatel_id_uzivatel);  
END VytvoritZpetnouVazbu;
```

## 11. VytvoritZpravu

- **Popis:** Vytvoří novou zprávu.
- **Parametry:**
  - `titulek` (IN): Titulek zprávy



- **popisek** (IN): Popisek zprávy
  - **id\_uzivatele** (IN): ID uživatele
  - **id\_tema** (IN): ID tématu
  - **created\_report** (OUT): Výstupní kurz
- **Kód:**

```
create or replace NONEDITIONABLE PROCEDURE VytvoritZpravu(
    titulek IN VARCHAR2,
    popisek IN VARCHAR2,
    id_uzivatele IN INTEGER,
    id_tema IN INTEGER,
    created_report OUT SYS_REFCURSOR
) AS
    current_time TIMESTAMP;
    created_id INTEGER;
BEGIN
    SELECT current_timestamp INTO current_time FROM dual;

    INSERT INTO zprava (
        id_zprava, titulek, popisek, uzavreno, vytvoreno, upraveno,
        uzivatel_id_uzivatel, tema_id_tema
    )
    VALUES (
        id_zprava_seq.NEXTVAL, titulek, popisek, NULL, current_time,
        current_time, id_uzivatele, id_tema
    )
    RETURNING id_zprava INTO created_id;

    OPEN created_report FOR
        SELECT *
        FROM zprava
        WHERE id_zprava = created_id;
END VytvoritZpravu;
```

## Popis triggerů

### 1. **check\_dokument\_nazev**

- **Popis:** Kontroluje, zda název dokumentu dodržuje stanovenou strukturu.
- **Kód:**

```
create or replace NONEDITIONABLE TRIGGER check_dokument_nazev
BEFORE INSERT OR UPDATE ON dokument
FOR EACH ROW
DECLARE
    v_pattern VARCHAR2(100) := '^[\0-9]+_[a-zA-Z\0-9]+_[a-zA-Z\0-9]+$';
BEGIN
```

```

        IF NOT REGEXP_LIKE(:NEW.nazev, v_pattern) THEN
            RAISE_APPLICATION_ERROR(-20001, 'Název dokumentu musí dodržovat
strukturu cislopracovnika_nazev_koncovka.');
```

```

        END IF;
    END;
```

## 2. check\_email\_format

- **Popis:** Kontroluje formát emailu uživatele.
- **Kód:**

```

create or replace NONEDITIONABLE TRIGGER check_email_format
BEFORE INSERT OR UPDATE ON uzivatel
FOR EACH ROW
DECLARE
    v_pattern VARCHAR2(100) := '^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-
z]{2,}$';
BEGIN
    IF NOT REGEXP_LIKE(:NEW.email, v_pattern) THEN
        RAISE_APPLICATION_ERROR(-20002, 'Email musí obsahovat platný
formát jmeno@domena.xx.');
```

```

    END IF;
END;
```

## 3. update\_zprava\_upraveno

- **Popis:** Aktualizuje čas poslední úpravy zprávy.
- **Kód:**

```

create or replace NONEDITIONABLE TRIGGER update_zprava_upraveno
BEFORE UPDATE ON zprava
FOR EACH ROW
BEGIN
    :NEW.upraveno := SYSTIMESTAMP;
END;
```