|  |  |  |
| --- | --- | --- |
| **TECHNICKÁ UNIVERZITA V KOŠICIACH**  **FAKULTA ELEKTROTECHNIKY A INFORMATIKY** | | |
| **Check Access Control: Secure Entry with Arduino**  **Zadanie č. 2** | | |
|  | | |
| **2024** | | **Sofiia Pravytska**  **Mikita Kautsevich**  **Yaroslav Sabadash**  **Maksym Tiuiunnyk** |
| **TECHNICKÁ UNIVERZITA V KOŠICIACH**  **FAKULTA ELEKTROTECHNIKY A INFORMATIKY** | | |
| **Check Access Control: Secure Entry with Arduino**  **Zadanie č. 2** | | |
|  | | |
| Študijný program: | Inteligentné systémy, Hospodárska informatika | |
| Študijný odbor: | Informatika | |
| Školiace pracovisko: | Katedra kybernetiky a umelej inteligencie (KKUI) | |
| Školiteľ: | Titul Meno Priezvisko, Titul | |
| Konzultant: | Titul Meno1 Priezvisko1, Titul  Titul Meno2 Priezvisko2, Titul | |
| **Sofiia Pravytska**  **Mikita Kautsevich**  **Yaroslav Sabadash**  **Maksym Tiuiunnyk** | | |
| **2024 Košice** |  | |

**Abstrakt v SJ**

Tento dokument prezentuje dokumentáciu projektu "Check Access Control: Secure Entry with Arduino", ktorý si kladie za cieľ vyvinúť systém kontroly prístupu využívajúci Arduino a elektronické karty alebo čipy na identifikáciu používateľa. Cieľom projektu je vytvoriť bezpečný a pohodlný systém prístupu k špecifickým oblastiam alebo zdrojom prostredníctvom integrácie hardvérových a softvérových komponentov. Medzi zvolené technológie patrí Arduino vďaka svojmu používateľsky priateľnému rozhraniu, flexibilite a kompatibilite s rôznymi periférnymi zariadeniami. Systém zahŕňa implementáciu logiky Arduino pre čítanie a spracovanie údajov z elektronických kariet, riadenie mechanizmov prístupu a integráciu hardvérových komponentov. Táto dokumentácia poskytuje prehľad o projekte, analýzu úloh, odôvodnenie zvolených technológií, špecifikácie vstupov a výstupov a príklad použitia scenára.

**Kľúčové slová v SJ**

Kontrola Prístupu, Arduino, Elektronické Karty, Identifikácia, Bezpečnosť, Integrácia Hardvéru, Implementácia Softvér

**Abstract**

This document presents the documentation for the "Check Access Control: Secure Entry with Arduino " project, which aims to develop an access control system utilizing Arduino and electronic cards or chips for user identification. The project's objective is to create a secure and convenient access system for specific areas or resources through the integration of hardware and software components. The chosen technologies include Arduino due to its user-friendly interface, flexibility, and compatibility with various peripheral devices. The system involves the implementation of Arduino logic for reading and processing data from electronic cards, controlling access mechanisms, and integrating hardware components. This documentation provides an overview of the project, task analysis, justification of chosen technologies,input-output specifications, and an example usage scenario.

**Keywords**

Access Control, Arduino, Electronic Cards, Identification, Security, Hardware Integration, Software Implementation.

**Zadanie práce**

Vytvorte webovú alebo mobilnú aplikáciu, ktorá bude riešiť zadanú problematiku. Táto aplikácia by mala obsahovať klasicky backend a frontend, nasadená by mala byť na cloud-e jedného z veľkých providerov (prednáška 3). Zároveň by pre zabezpečenie svojej funkcionality mala využívať cloudové služby, ktoré bude poskytovať aspoň 1 provider, ktorý je iný ako provider poskytujúci hosting. Na ukladanie dát v aplikácii je potrebné použiť databázu, čo budete ukladať necháme na Vás, no niečo áno, a na komunikáciu so službami API poskytnuté providermi. Aplikáciu je postačujúce nasadzovať na úrovni PaaS, nie je potrebné fungovať na úrovni IaaS. Zadanie bude určené pre skupinu 4 študentov. Na dosiahnutie plného počtu bodov členom skupiny, bude potrebná jeho aktívna účasť na zadaní, ktorá bude kontrolovaná pomocou systému GitLab alebo GitHub. Každý tím bude disponovať privátnym Git repozitárom, na ktorý bude mať prístup ich cvičiaci. Jednotlivé úlohy pre jednotlivých členov team-u by mali byť rozbité do To Do listu a tieto by mali byť adresované jednotlivými commitmi. Voľba technológie, platformy na hosting, rozloženia UI a použitých služieb je plne vo Vašej réžií. Na tvorbu týchto zadaní by Vám mali stačiť študentské/trial verzie od cloudových providerov.

**Obsah**

[Zoznam obrázkov 6](#_Toc164302160)

[Úvod 7](#_Toc164302161)

[1. Formulácia úlohy a cieľ práce 8](#_Toc164302162)

[2. Teoretický rozbor zvolenej témy 9](#_Toc164302163)

[3. Analýza stavu problematiky 10](#_Toc164302164)

[4. Návrh a implementácia riešenia zvolenej problematiky 11](#_Toc164302165)

[4.1. Arduino schema 12](#_Toc164302166)

[4.2. Webová stránka 12](#_Toc164302167)

[4.3. Databaza 13](#_Toc164302168)

[4.4. Webová aplikácia 14](#_Toc164302169)

[4.5. Súbory YAML 14](#_Toc164302170)

[Záver 16](#_Toc164302171)

[Zoznam použitej literatúry 17](#_Toc164302172)

Zoznam obrázkov

[Obr. 1 Arduino schema 12](#_Toc164302176)

[Obr. 3 Azure Databaza 13](#_Toc164302177)

Úvod

Projekt "Check Access Control: Secure Entry with Arduino" vznik s cieľom vytvoriť efektívny a bezpečný systém kontroly prístupu pomocou platformy Arduino, elektronických kariet alebo čipov a webového rozhrania. Tento systém umožňuje správu prístupu k určitým priestorom alebo zdrojom pomocou elektronického identifikátora. Na základe priloženia karty alebo čipu k čítačke, systém overí identitu používateľa a rozhodne o otvorení alebo uzatvorení dverí.

**Zvolené technológie a postup:**

* Platforma Arduino a elektronické karty/čipy: Využívame platformu Arduino na riadenie zariadení a čítanie údajov z elektronických kariet alebo čipov pre identifikáciu používateľov.
* Webové rozhranie a databáza: Vytvorili sme webové rozhranie, ktoré zobrazuje stav dverí a umožňuje správu používateľov. Údaje o autorizovaných používateľoch sú ukladané do databázy pre budúce použitie.

**Funkčnosť systému:**

* Autorizovaný používateľ priloží svoju elektronickú kartu alebo čip k čítačke.
* Arduino overí identitu používateľa na základe údajov z karty alebo čipu.
* Ak je používateľ autorizovaný, systém otvorí dvere a používateľ môže vstúpiť. Ak nie je autorizovaný, dvere zostanú uzamknuté.

Tento projekt ponúka efektívne a spoľahlivé riešenie pre kontrolu prístupu, ktoré je jednoduché na použitie a zároveň zabezpečuje bezpečnosť a ochranu priestorov alebo zdrojov.

1. Formulácia úlohy a cieľ práce

Úlohou tejto práce je navrhnúť, implementovať a otestovať systém kontroly prístupu s využitím platformy Arduino a elektronických kariet alebo čipov pre identifikáciu používateľov. Cieľom práce je vytvoriť bezpečný, spoľahlivý a užívateľsky prívetivý systém, ktorý umožní autorizovaným užívateľom prístup k určitým priestorom alebo zdrojom. Konkrétne sa zameriame na návrh a implementáciu softvérových a hardvérových komponentov systému, vrátane logiky pre čítanie a spracovanie údajov z elektronických kariet, riadenia mechanizmov prístupu a integrácie s periférnymi zariadeniami. Na dosiahnutie tohto cieľa vykonáme analýzu požiadaviek, navrhneme vhodné riešenie, vykonáme implementáciu a otestujeme systém.

1. Teoretický rozbor zvolenej témy

V tejto časti práce sa zameriame na teoretický rozbor súvisiacich konceptov a technológií spojených s kontrolou prístupu a využitím platformy Arduino, ESP, Docker, Microsoft Azure, databázami a vývojom webového rozhrania pre ovládanie systému. Preskúmame princípy fungovania týchto technológií a ich vzájomné interakcie.

* Arduino a ESP: preskúmame možnosti využitia platformy Arduino a ESP na implementáciu systému kontroly prístupu, vrátane základných princípov programovania, možností komunikácie s perifériami a sieťového pripojenia.
* Docker: preskúmame základné princípy Docker kontejnerizácie a jej výhody pre vývoj a nasadenie aplikácií, vrátane možností použitia Docker na izoláciu aplikácií a jednoduché nasadenie na rôzne platformy.
* Microsoft Azure: preskúmame možnosti využitia cloudu Microsoft Azure na hostovanie aplikácie a dátových služieb, vrátane možností nasadenia kontajnerov pomocou služby Azure Kubernetes Service (AKS) a správy databáz pomocou služby Azure SQL Database.
* Databáze: preskúmame princípy relačných databáz a možnosti využitia databázových systémov na ukladanie a správu dát v rámci systému kontroly prístupu.
* Vývoj webového rozhrania: preskúmame základy vývoja webového rozhrania pomocou moderných technológií ako HTML, CSS, JavaScript a frameworkov pre vývoj webových aplikácií, ako je napríklad React alebo Angular. Ďalej sa zameriame na využitie POST požiadaviek pre komunikáciu medzi webovým rozhraním a backendovou časťou aplikácie.

1. Analýza stavu problematiky

Hlavným problémom spočívalo v prenášaní informácií z Arduina na server a následnom ich ukladaní do databázy. Tento proces zahŕňal niekoľko kľúčových výziev, vrátane zabezpečenia spojenia medzi Arduinom a serverom, riadenia komunikácie a prenosu údajov a integrácie s databázou pre trvalé ukladanie informácií.

Pri analýze sme zistili, že zabezpečenie spoľahlivej komunikácie medzi Arduinom a serverom môže byť problematické, najmä pri použití bezdrôtového pripojenia alebo v prostredí s obmedzeným pripojením k internetu. Integrácia s databázou vyžaduje správne nastavenie a konfiguráciu pre zabezpečenie bezpečného a efektívneho ukladania údajov.

Táto analýza nám pomôže lepšie porozumieť problémom, s ktorými sa budeme potýkať, a navrhnúť vhodné riešenia pre ich riešenie.

1. Návrh a implementácia riešenia zvolenej problematiky

Na základe analýzy súčasného stavu problematiky a identifikovaných výziev sme navrhli a implementovali komplexné riešenie pre prenos informácií z Arduina na server a ich následné ukladanie do databázy.

**Návrh:**

* Navrhli sme robustný a spoľahlivý mechanizmus pre komunikáciu medzi Arduinom a serverom, zahŕňajúci zabezpečenie bezpečnosti a integrity údajov.
* Navrhli sme vhodnú architektúru aplikácie, ktorá umožňuje efektívne správu a spracovanie prichádzajúcich údajov.
* Navrhli sme dátový model pre ukladanie informácií do databázy, ktorý zohľadňuje potreby aplikácie a zabezpečuje efektívnu manipuláciu s údajmi.

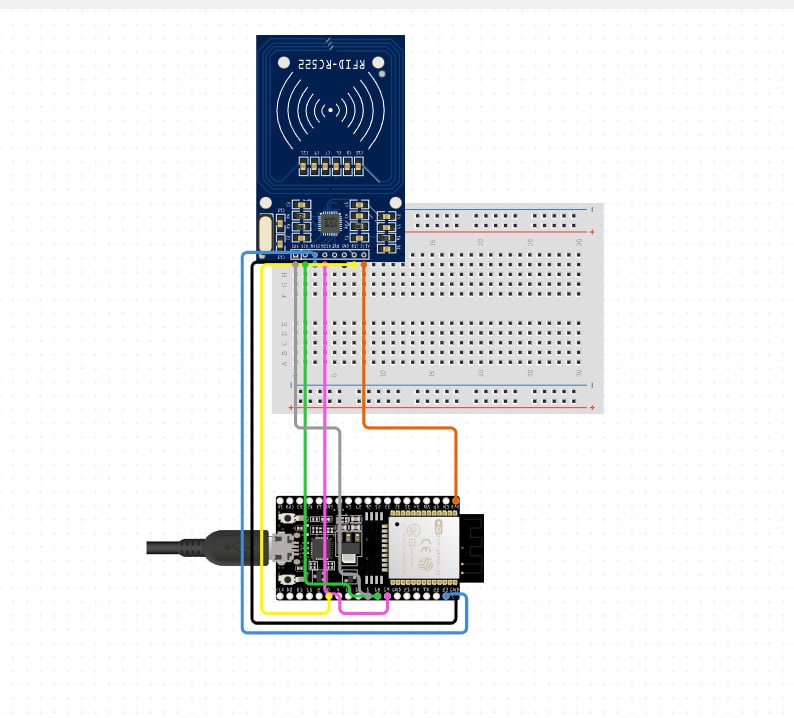
**Implementácia:**

* Implementovali sme komunikačný protokol pre prenos údajov z Arduina na server pomocou vhodných technológií ako sú HTTP POST požiadavky.
* Implementovali sme serverovú časť aplikácie, ktorá prijíma údaje z Arduina, vykonáva ich spracovanie a ukladanie do databázy.
* Implementovali sme databázovú časť aplikácie, ktorá zabezpečuje efektívne ukladanie a správu údajov v databáze.
* Vykonávali sme testovanie a ladenie aplikácie, aby sme zabezpečili správnu funkčnosť a spoľahlivosť celého systému.

Tento návrh a implementácia riešenia nám umožnili úspešne vyriešiť hlavnú výzvu spojenú s prenosom informácií z Arduina na server a ich ukladaním do databázy.

* 1. Arduino schema

Naším projektom je ESP32, Arduino Uno, čitačka:



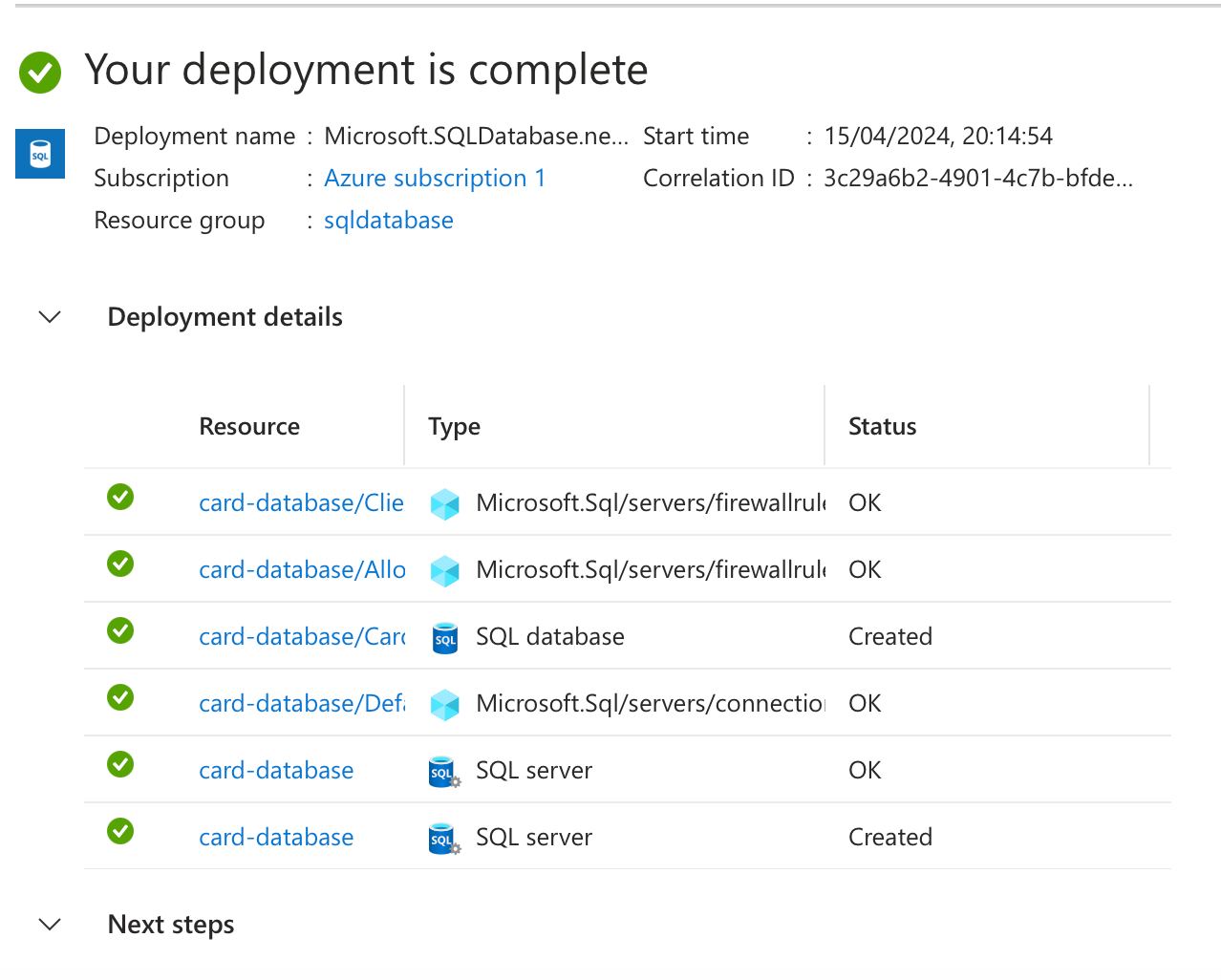
Obr. 1 Arduino schema

* 1. Webová stránka
* Názov stránky: "Access Control with Arduino".
* Navigačné menu umožňujúce presun do rôznych častí stránky: "Info", "How it Works", "Our Team", "Feedback".
* Sekcia "Info" obsahujúca stručný popis projektu a pole pre vstup používateľského mena s tlačidlami "Login" a "Reset".
* Sekcia "How it Works", kde sú zobrazené tri kroky fungovania Arduina s príslušnými obrázkami a popisom.
* Sekcia "Our Team", kde sú uvedení členovia tímu projektu s ich menami a funkciami.
* Sekcia "Feedback", zahŕňajúca formulár pre spätnú väzbu na odoslanie správy, ako aj kontaktné údaje.
* Päta stránky s informáciami o spoločnosti a právach.

Okrem toho v súbore sú odkazy na externé zdroje: štýlový súbor CSS a skript JavaScript, ktorý obsahu

* Poslucháč udalostí, ktorý počúva udalosť "DOMContentLoaded", ktorá nastane pri načítaní celého obsahu dokumentu HTML. Potom vyhľadá všetky odkazy (prvky <a>), ktoré majú atribút href začínajúci na znak #. Pre každý takýto odkaz pridá udalosť "kliknutie", ktorá zruší predvolené správanie prechodu na odkaz (kotvu) a namiesto toho sa plynulo posunie na cieľový prvok na stránke.
* Funkcia login(), ktorá sa zavolá po stlačení tlačidla "Prihlásenie". Získava hodnotu zo vstupného poľa používateľského mena, porovnáva ju s reťazcom "admin" (nerozlišuje veľké a malé písmená) a podľa toho zobrazí príslušnú správu o úspechu alebo chybe. Ak je meno správne, pridá triedu "open-door" k prvku s id "door", čím simuluje otvorenie dverí.
* Funkcia resetData(), ktorá sa volá po stlačení tlačidla "reset". Vymaže prihlasovaciu správu, obnoví farbu správy na predvolenú a vymaže hodnotu poľa používateľského mena. Odstráni tiež triedu "open-door" z prvku s id "door", čím simuluje zatvorenie dverí.
  1. Databaza

V systéme Microsoft Azure bola vytvorená aj databáza na zaznamenávanie údajov:



Obr. 3 Azure Databaza

* 1. Webová aplikácia

V súbore (`app.py`) sa vytvára webová aplikácia pomocou frameworku Flask. Obsahuje cestu `/card\_read`, ktorá očakáva POST požiadavky na čítanie údajov z karty. Po prijatí požiadavky aplikácia získa UID karty z JSON údajov požiadavky. Potom porovná získané UID s predom nastaveným povoleným UID. V závislosti na zhode UID aplikácia odošle správu cez SocketIO o udelení prístupu ("granted") alebo odmietnutí prístupu ("denied").

V súbore (`Dockerfile`) sa definuje Dockerový obraz. Najprv sa inštalujú potrebné závislosti, vrátane gcc a libc-dev. Potom sa skopírujú súbory aplikácie do pracovného adresára kontajnera. Nakoniec sa definuje príkaz na spustenie aplikácie, ktorý v tomto prípade volá Python so súborom `app.py`. Okrem toho sa určuje, že aplikácia bude počúvať na porte 5000.

* 1. Súbory YAML

Tieto súbory sú definície Kubernetes resource objects, ktoré popisujú infraštruktúru a služby v klastri Kubernetes. Tu je krátky popis každého súboru:

**Deployment pre Flask aplikáciu:**

* Vytvára dva repliky aplikácie s názvom "flask-app".
* Špecifikuje kontajner s názvom "flask-app", ktorý používa obraz backendovej aplikácie z registra s názvom "backenddockerhm.azurecr.io/backend:latest".
* Nastavuje port 5000 pre komunikáciu v kontajneri.
* Definuje imagePullSecrets s názvom "regcred" pre sťahovanie obrazov z registra.

**Služba pre PostgreSQL:**

* Vytvára službu s názvom "postgres-service".
* Nastavuje port 5432 pre prístup k PostgreSQL službe.
* Používa selektor app: postgres na presmerovanie požiadaviek na správny pod.

**Deployment pre HTML stránku:**

* Vytvára dva repliky HTML stránky s názvom "html-page".
* Používa kontajner s názvom "html-page", ktorý používa obraz frontendovej aplikácie z registra s názvom "frontenddockerhm.azurecr.io/frontend:latest".
* Nastavuje port 80 pre komunikáciu v kontajneri.

**Služba pre HTML stránku:**

* Vytvára službu s názvom "html-page-service".
* Presmerováva požiadavky na port 80 na pod HTML stránky.
* Nastavuje typ LoadBalancer pre prístup k službe zvonku klastrov.

Tieto súbory sú dôležitou súčasťou definovania a správy aplikácií a služieb v Kubernetes klasteri. Pomáhajú automatizovať nasadenie a škálovanie aplikácií a poskytujú jednotný spôsob konfigurácie a riadenia infraštruktúry.

Záver

Projekt "Check Access Control: Secure Entry with Arduino" sme úspešne navrhli, implementovali a otestovali systém kontroly prístupu, ktorý využíva platformu Arduino a ESP na riadenie a monitorovanie prístupu k určitým priestorom alebo zdrojom prostredníctvom elektronických čipov alebo kariet. Tím úspešne dokončil projekt vďaka spolupráci a využitiu moderných technológií.

Systém je postavený na architektúre, ktorá zahŕňa komunikáciu medzi Arduinom a ESP a následné spracovanie údajov. Funkcie, ako je overenie autorizácie používateľa, sú implementované v kóde aplikácie. Napríklad funkcia `card\_read()` v súbore `app.py` prijíma POST požiadavky na čítanie údajov karty, spracúva UID karty a porovnáva ho s predtým stanoveným povoleným UID. Na základe tohto porovnania systém rozhodne, či povoliť alebo zamietnuť prístup.

Pri implementácii projektu bolo nevyhnutné riešiť výzvy spojené s integráciou komponentov, ako sú Arduino, ESP a databáza. Napríklad priradenie ESP k Wi-Fi sieti pre komunikáciu so serverom, ako je uvedené vo funkcii `login()`, bolo nevyhnutné pre správnu funkciu systému.

Vďaka dôkladnej dokumentácii a testovaniu sme zabezpečili spoľahlivosť a funkčnosť systému. Celkovo možno projekt "Check Access Control: Secure Entry with Arduino" považovať za úspešný a má potenciál byť využitý v rôznych aplikáciách, kde je potrebné riadenie a monitorovanie prístupu k určitým priestorom alebo zdrojom.

Zoznam použitej literatúry

1. Docker documentation. Dostupne na internete: https://docs.docker.com
2. Azure documentation. Dostupne na internete: https://learn.microsoft.com/en-us/azure/?product=popular
3. YouTube videa pre konfigurácie serveru, databázy