

Zapis ról przydzielonych w grupie projektowej.

Tworzenie i zarządzanie bazą danych.

Opis tworzonych funkcjonalności

Stworzenie bazy danych.

Obsługa bazy za pomocą programu pgAdmin.

Zaprojektowanie budowy bazy i tworzenie diagramu ERD za pomocą programu Oracle Data Modeler.

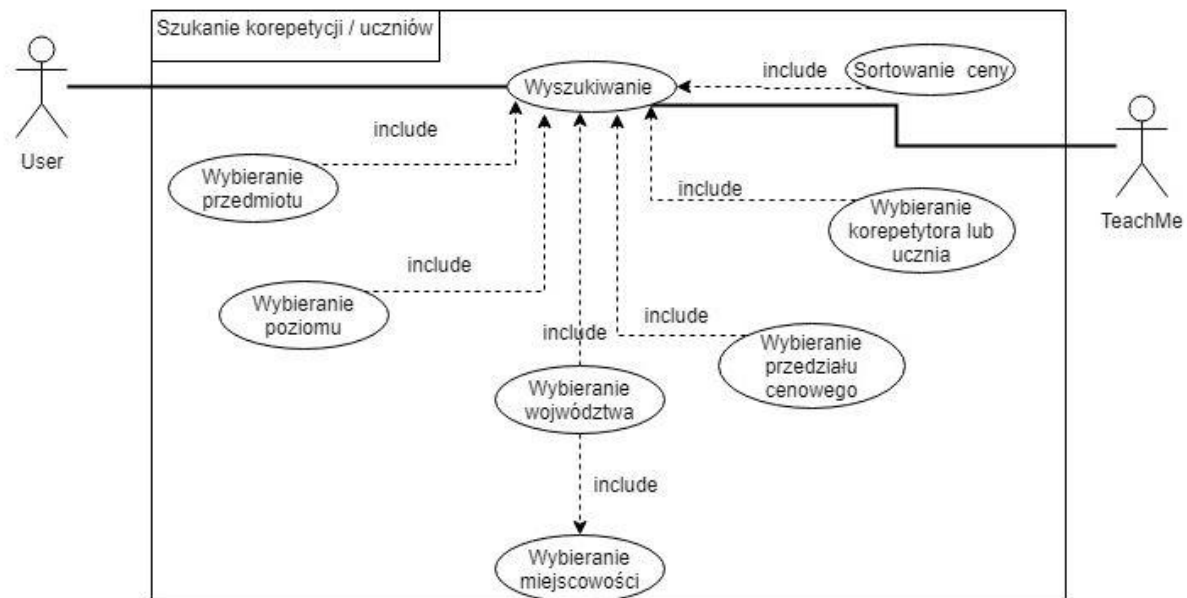
Stworzenie tabel oraz relacji.

Dodawanie rekordów do bazy danych.

Edytowanie bazy wraz ze zmieniającą się koncepcją projektu.

Diagram przypadków użycia funkcjonalności

Diagram dla funkcji wyszukiwania



Stworzony kod

```
CREATE TABLE public.adminn (  
    id_admin integer DEFAULT nextval('public.id'::regclass) NOT NULL,  
    login character varying(127) NOT NULL,  
    pasword character varying(4000) NOT NULL,  
    email character varying(127) NOT NULL  
);  
CREATE TABLE public.city (  
    id_city integer DEFAULT nextval('public.id'::regclass) NOT NULL,  
    name character varying(127) NOT NULL,  
    voivodeship_id_voivodeship integer NOT NULL  
);  
CREATE TABLE public.meeting (  
    id_meeting integer DEFAULT nextval('public.id'::regclass) NOT NULL,  
    date date NOT NULL,  
    "time" timestamp with time zone NOT NULL,  
    userr_id_user integer,  
    meeting_place character varying(127),  
    notice_id_notice integer  
);  
CREATE TABLE public.notice (  
    id_notice integer DEFAULT nextval('public.id'::regclass) NOT NULL,  
    look_or_offer character(1),  
    note character varying(512) NOT NULL,  
    meeting_place character varying(127) NOT NULL,  
    meeting_date date NOT NULL,  
    price double precision,  
    subject_id_subject integer NOT NULL,  
    active character(1) NOT NULL,  
    "timestamp" timestamp with time zone DEFAULT CURRENT_TIMESTAMP,  
    userr_id_user integer,  
    time_from timestamp with time zone,  
    time_to timestamp with time zone,  
    level integer  
);  
CREATE TABLE public.opinion (  
    id_opinion integer DEFAULT nextval('public.id'::regclass) NOT NULL,  
    rating integer NOT NULL,  
    comment character varying(500) NOT NULL,  
    user_to integer,  
    user_from integer  
);  
CREATE TABLE public.subject (  
    id_subject integer DEFAULT nextval('public.id'::regclass) NOT NULL,  
    name character varying(127) NOT NULL  
);  
CREATE TABLE public.userr (  
    id_user integer DEFAULT nextval('public.id'::regclass) NOT NULL,  
    login character varying(127),  
    name character varying(127) NOT NULL,  
    surname character varying(127) NOT NULL,  
    birth_date date,  
    avatar character varying(127),  
    phone character varying(127),  
    email character varying(127) NOT NULL,  
    timesstamp timestamp with time zone,  
    password character varying(4000),  
    city_id_city integer,  
    about character varying(127)  
);  
CREATE TABLE public.voivodeship (  

```

```

        id_voivodeship integer NOT NULL,
        name_voivodeship character varying(127) NOT NULL
    );
ALTER TABLE ONLY public.adminn
    ADD CONSTRAINT admin_pk PRIMARY KEY (id_admin);

ALTER TABLE ONLY public.city
    ADD CONSTRAINT city_pk PRIMARY KEY (id_city);

ALTER TABLE ONLY public.meeting
    ADD CONSTRAINT meeting_pk PRIMARY KEY (id_meeting);

ALTER TABLE ONLY public.notice
    ADD CONSTRAINT notice_pk PRIMARY KEY (id_notice);

ALTER TABLE ONLY public.opinion
    ADD CONSTRAINT opinion_pk PRIMARY KEY (id_opinion);

ALTER TABLE ONLY public.subject
    ADD CONSTRAINT subject_pk PRIMARY KEY (id_subject);

ALTER TABLE ONLY public.userr
    ADD CONSTRAINT user_pk PRIMARY KEY (id_user);

ALTER TABLE ONLY public.voivodeship
    ADD CONSTRAINT voivodeship_pk PRIMARY KEY (id_voivodeship);

ALTER TABLE ONLY public.city
    ADD CONSTRAINT city_voivodeship_fk FOREIGN KEY (voivodeship_id_voivodeship) REFERENCES
public.voivodeship(id_voivodeship);

ALTER TABLE ONLY public.meeting
    ADD CONSTRAINT meeting_notice_fk FOREIGN KEY (notice_id_notice) REFERENCES public.notice(id_notice);

ALTER TABLE ONLY public.meeting
    ADD CONSTRAINT meeting_userr_fk FOREIGN KEY (userr_id_user) REFERENCES public.userr(id_user);

ALTER TABLE ONLY public.notice
    ADD CONSTRAINT notice_subject_fk FOREIGN KEY (subject_id_subject) REFERENCES
public.subject(id_subject);

ALTER TABLE ONLY public.notice
    ADD CONSTRAINT notice_userr_fk FOREIGN KEY (userr_id_user) REFERENCES public.userr(id_user);

ALTER TABLE ONLY public.opinion
    ADD CONSTRAINT opinion_user_fk FOREIGN KEY (user_from) REFERENCES public.userr(id_user);

ALTER TABLE ONLY public.opinion
    ADD CONSTRAINT opinion_user_fkv2 FOREIGN KEY (user_to) REFERENCES public.userr(id_user);

ALTER TABLE ONLY public.userr
    ADD CONSTRAINT user_city_fk FOREIGN KEY (city_id_city) REFERENCES public.city(id_city);

```

Opis zrealizowanych testów

Testowanie bazy danych polegało głównie na tworzeniu zapytań i sprawdzaniu czy baza w poprawny sposób zwraca dane.

Podsumowanie

Praca indywidualna polegała na stworzeniu bazy danych oraz późniejszej jej obsłudze. Baze stworzoną za pomocą pgAdmin należało połączyć z Heroku, a następnie stworzyć odpowiednie tabele

oraz powiązać je relacjami. Następnie bazę wypełniono przykładowymi danymi i testowana czy jej koncepcja odpowiada potrzebom innych części projektu. Baza była także na bieżąco edytowana i obsługiwana wedle potrzeb. Stworzony został diagram encji, który ułatwiał prace oraz zrozumienie działania bazy innym uczestnikom projektu.