**Зміст**

Зміст 2

Анотація 3

Вступ 4

1. Постановка задачі 5
2. Діаграми потоків даних 6

2.1. DFD-0 6

2.2. DFD-1 7

2.3. DFD-2 9

2.4. Матриця подій 10

2.5. Бізнес план 11

2.7. ER-діаграма ... 12

3. Керівництво користувача 13

Висновки ........................................................................................................21

Список використаної літератури .................................................................22

Текст програми 23

**Анотація**

В даній роботі розроблена та описана інформаційна система «Аеропорт».

Метою створення цього проекту була автоматизація роботи з клієнтами, співробітниками, реалізація основних функцій, таких як прийняття на роботу співробітників, переведення на іншу посаду звільнення, додавання нових посад.

Програмне забезпечення (ПЗ) має інтуїтивно зрозумілий та простий інтерфейс і призначене для персоналу, який має елементарні навички роботи з комп’ютером та офісним програмним забезпеченням.

**Вступ**

На сьогоднішній день створення автоматизованих інформаційних систем є однією з домінуючих сфер розвитку нового інформаційного суспільства. У будь-якій сфері людської діяльності необхідна інформаційна система.

Мета інформатизації та комп’ютеризації – підвищити ефективність роботи. Але просте обладнання робочого місця „розумною” технікою не завжди призводить до підвищення робочих можливостей. Часто з’являється необхідність мати ефективну інформаційну систему, яка розроблена спеціально під бізнес-правила конкретного підприємства, розрахована на виконання не тільки загальних функцій, а й функцій специфічних для даного закладу. Завданням даної курсової роботи було створення такої системи для відділу кадрів аеропорту.

1. Постановка задачі

Реалізована в цьому проекті автоматизована система обробки даних призначена для використання менеджерами при організації роботи в аеропорті.

Створення програми має на меті полегшити роботу відділу кадрів, автоматизувати їхню взаємодію з працівниками свого відділу та іншими відділами.

В даному курсовому проекті були реалізовані основні функції інформаційної системи, яка б сприяла оптимізації роботи HR.

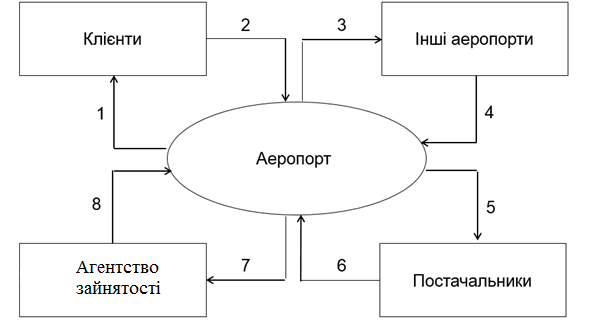
Зокрема:

* Додавання співробітників;
* Зміна посади;
* Звільнення;
* Зміна структури підприємства;

Для коректної роботи програми необхідне виконання відповідних вимог до технічних і програмних засобів, а саме:

* доступ до інтернету;
* процесор з архітектурою x86-64 (Intel з підтримкою EM64T, AMD з підтримкою AMD64);
* оперативна пам’ять 1024 Мб та вище;
* жорсткий диск 40Гб та вище­­;
* SVGA-відеокарта

1. **Діаграми потоків даних** 
   1. **DFD-0**



**Опис потоків даних на DFD-0**

1 – Квитки, інформація про розклад

2 – Бронювання, замовлення та оплата квитків

3 – Узгодження польотів від нашого аеропорту, можливість аварійної посадки на іншому аеропорту

4 – Узгодження польотів до нашого аеропорту, можливість прийняти аварійну посадку

5 – Замовлення матеріальної бази

6 – Виконання замовлення

7 – Вакансії

8 – Резюме, кандидати.

**2.2. DFD-1**

19

22

Інші аеропорти

Клієнти

Каси і довідкове бюро

23



20

21

18

17

16

Диспетчерська

Розклад

Літаки

15

14

13

7

6

Склад

Вакансії

Відділ обслуговування літаків

5

8

10

9

4

3

Відділ кадрів

Відділ матеріального забезпечення

12

Агентство зайнятості

Постачальники

1

2

11

**Опис потоків даних на DFD-1**

1 – Замовлення, гроші.

2 – Товари.

3 – Фіксування нових матеріалів\товарів на складі.

4 – Інформація про наявні матеріали на складі.

5 – Запит на необхідні для ремонту матеріали.

6 – Видача матеріалів.

7 – Запит на працівника з необхідними вміннями.

8 – Працівник.

9 – Інформація про необхідних працівників.

10 – Нові працівники.

11 – Вакансії.

12 – Кандидати.

13 – Інформація про всі літаки та їх технічний стан.

14 – Оновлення інформації про літаки та їх технічний стан.

15 – Інформація про доступні для посадки літаки.

16 – Формування розкладу польотів

17 – Можливість здійснити аварійну посадку

18 – Можливість прийняти аварійну посадку

19 – Узгодження польотів до нашого аеропорту

20 – Узгодження польотів від нашого аеропорту

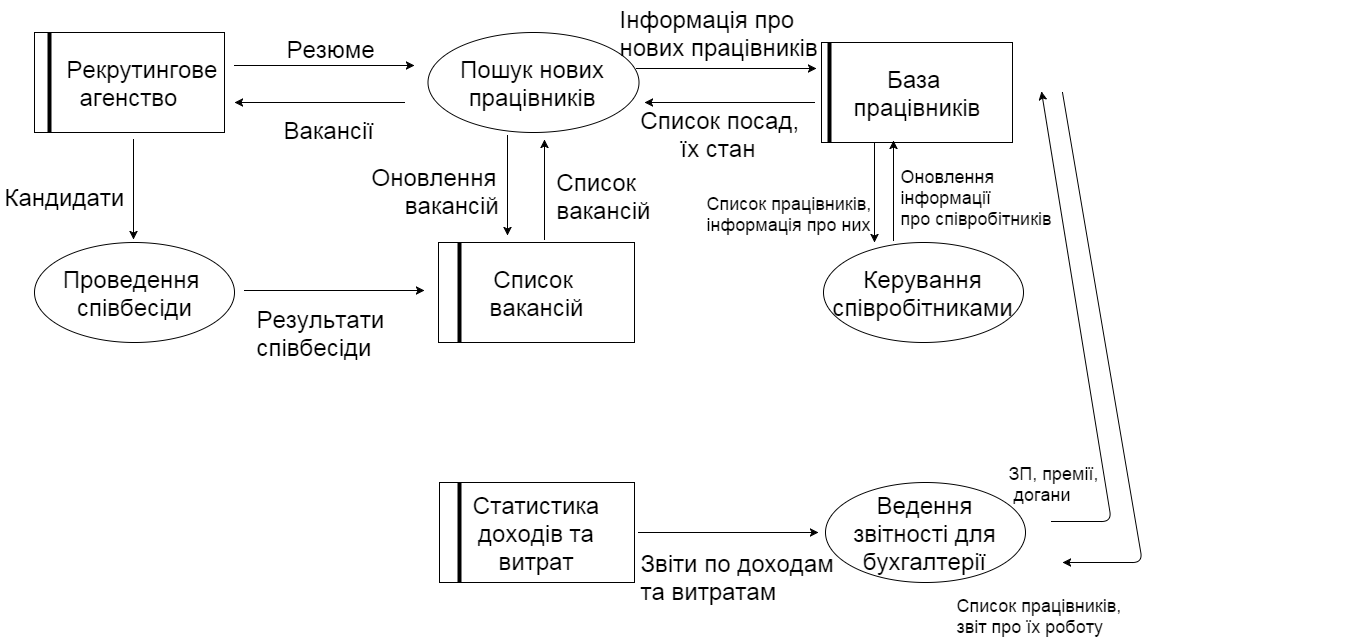
21 – Розклад польотів

22 – Квитки, інформація про розклад

23 – Бронювання, замовлення та оплата квитків

­­­

**2.3. DFD-2**



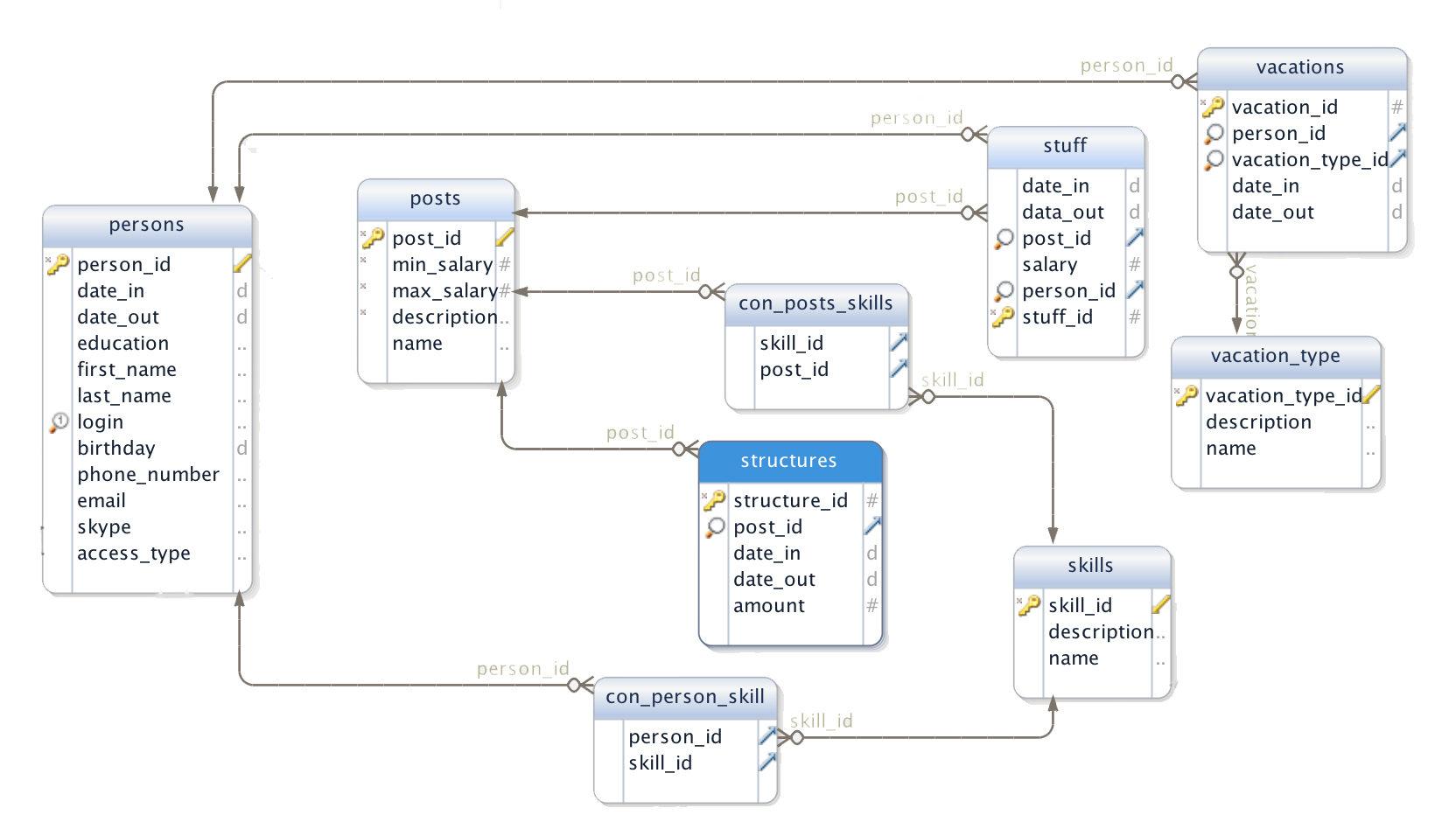
**2.4. Матриця подій**

|  |  |  |
| --- | --- | --- |
| **№** | **Подія** | **Реакція** |
| **1** | Додавання нової посади | Додати інформацію про нову посаду в базу посад. |
| **2** | Прийом нового працівника | Додати інформацію про нового працівника в базу працівників, базу посад працівників, базу адрес працівників, базу освіти працівників. Оновити базу штатного розкладу. |
| **3** | Зміна інформації про працівника | Редагувати інформацію робітника в базі працівників. |
| **4** | Звільнення працівника | Додати дату кінця роботи для працівника за умови, якщо вказана дата звільнення не менша, ніж дата початку роботи. |
| **5** | Зміна інформації про посаду | Додавання актуальних параметрів про посаду в базі посад |

**2.5.** **Бізнес правила**

Бізнес правила підсистеми «Відділ кадрів»:

1. Працівником компанії може стати будь-який громадянин України, що досяг 18-річного віку.
2. Посади співробітників мають відповідати національному класифікатору професій.
3. Неможливо видалити робітника з бази даних.
4. При переведенні з посади на посаду, вказується сьогоднішня дата.
5. Не можна наймати працівників без вказання посади.

**2.7. ER – діаграма**

1. **Керівництво користувача**

При запуску програми з’являється вікно, на якому запропоновано під’єднатись до бази даних за допомогою існуючого логіна та пароля. При успішному під’єднанні з’явиться меню. В меню наявні усі можливі дії з базою даних.

При виборі одного з пунктів меню відкривається нове вікно, в якому наявні все необхідні елементи управління.

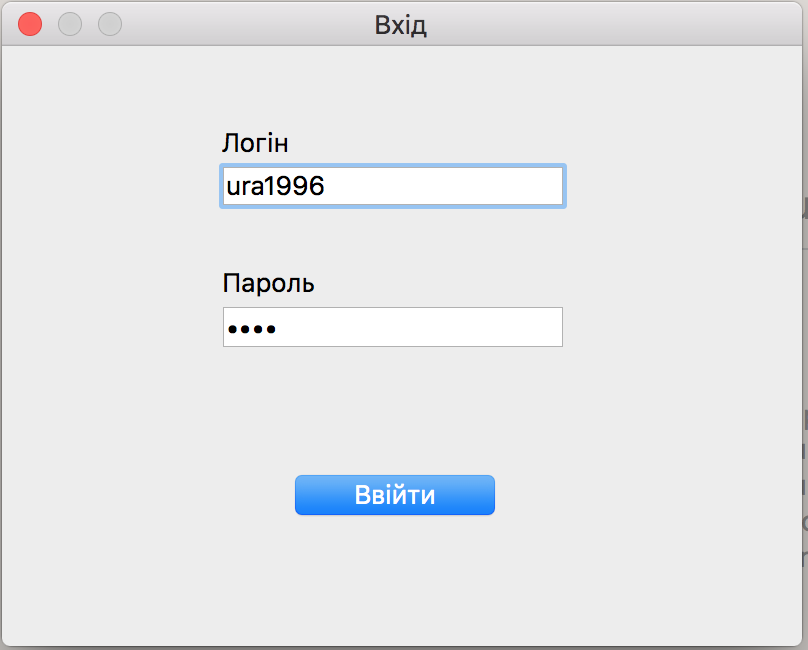


Рисунок 3.1 – Авторизація користувача

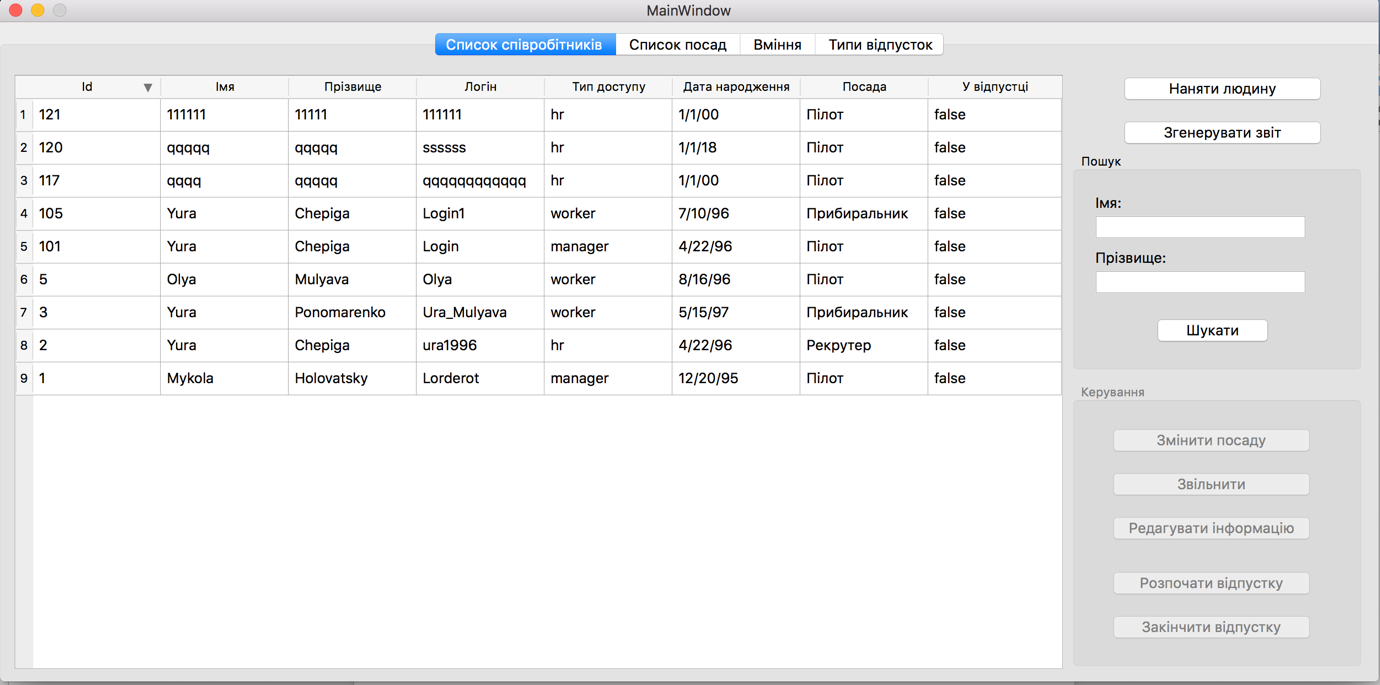
Головна форма зображена на рисунку 3.2. Вона містить пункти меню:

Рисунок 3.2 – Головне меню

Для додавання нового працівника натискаємо кнопку «Найняти людину» (Рисунок 3.3).

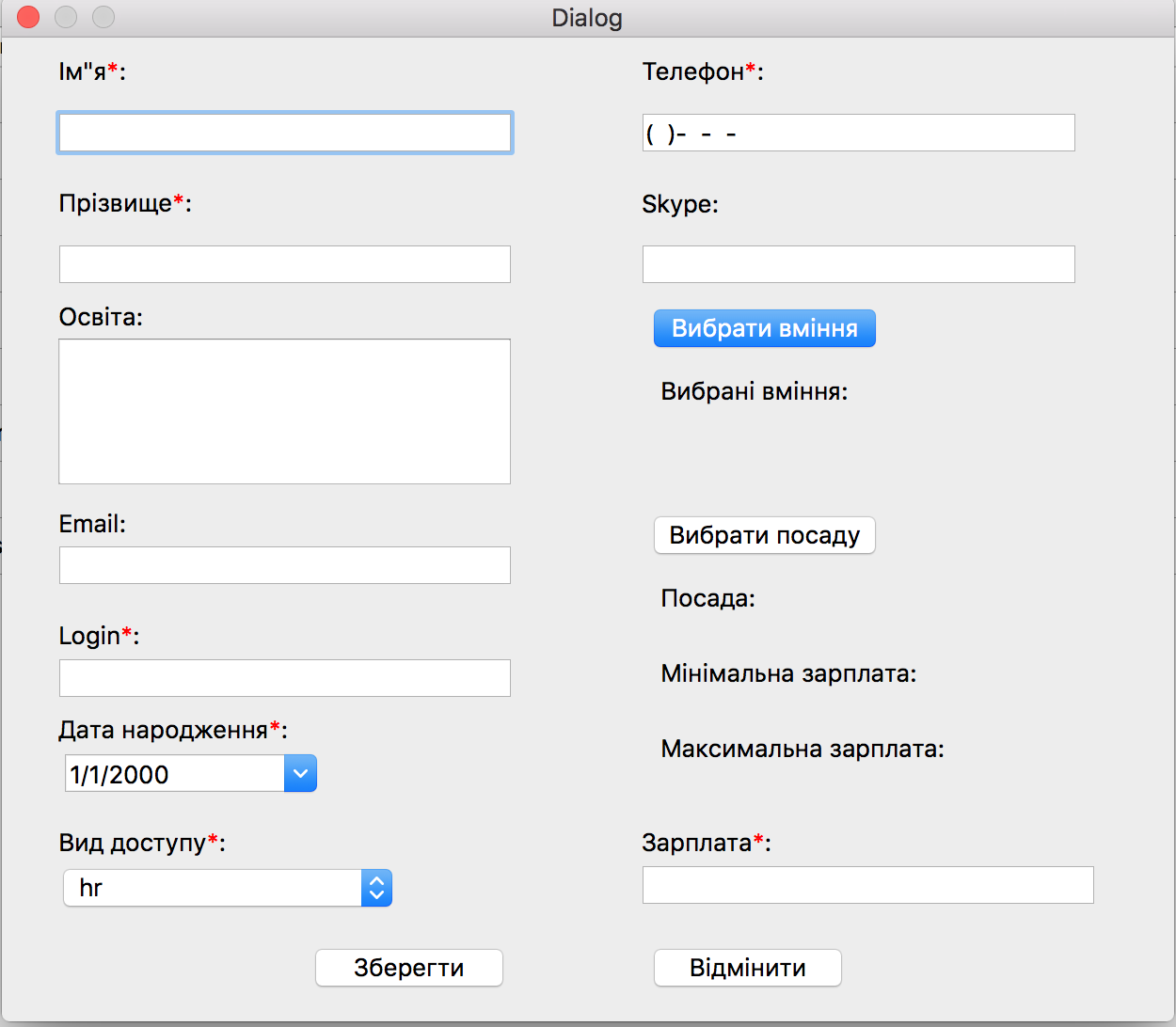


Рисунок 3.3 – Найняти робітника.

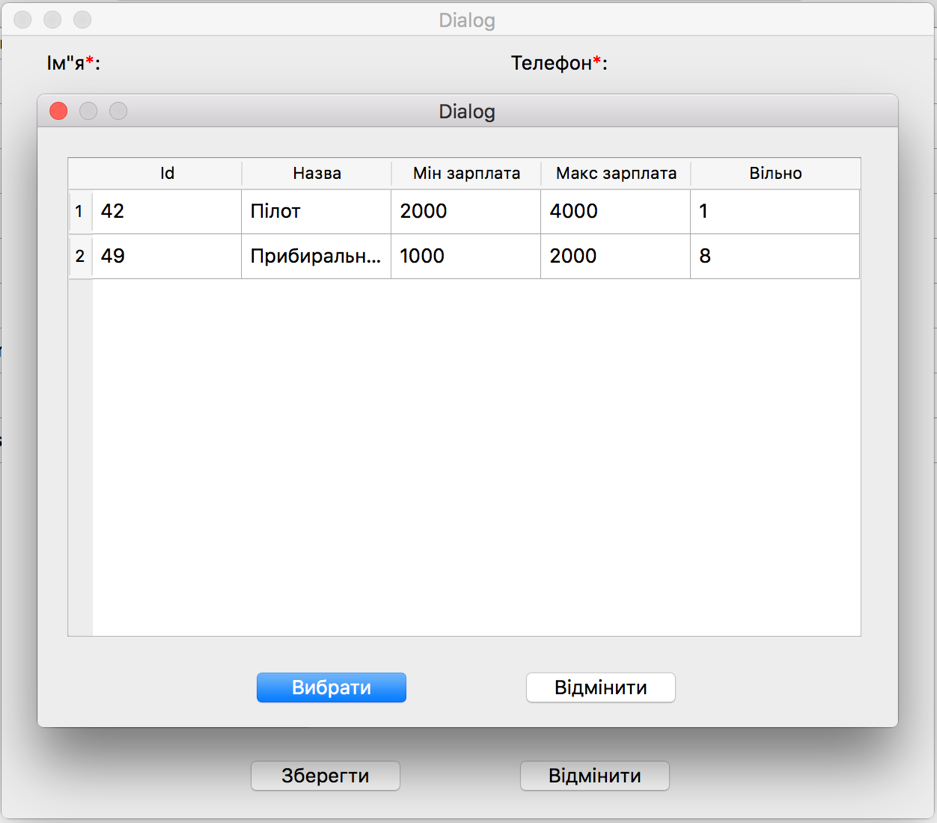
Для вибору посади потрібно натиснути кнопку “Вибрати посаду” (Рисунок 3.4).

Рисунок 3.4 – Найняти робітника. Вибір посади.

Якщо заповнені не всі поля, або було допущено помилку, програма виведе помилку (Рисунок 3.5).

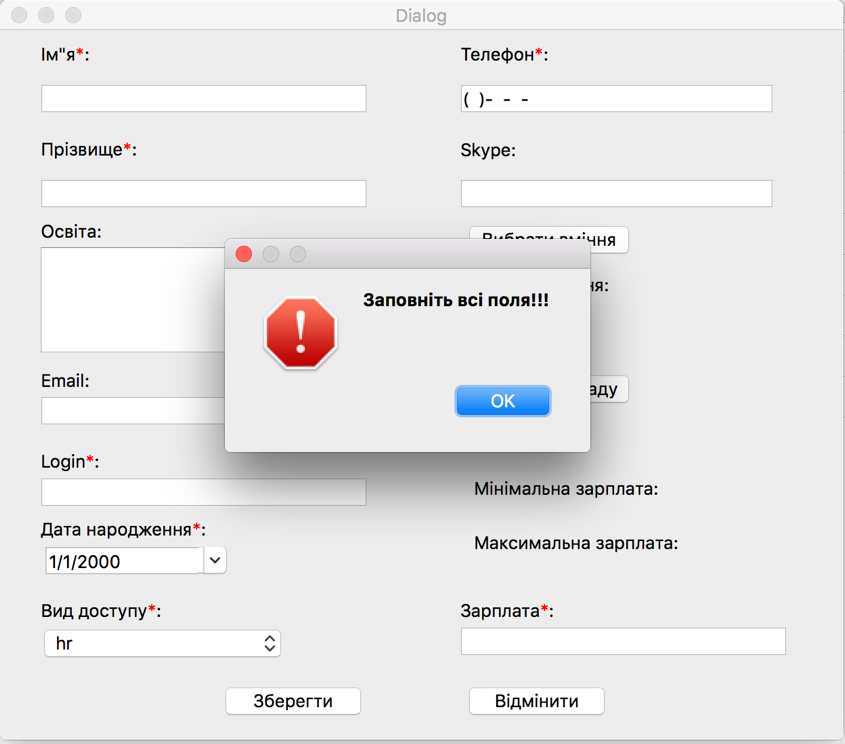


Рисунок 3.5 – Повідомлення про помилку

Якщо всі поля заповнені правильно, програма додасть новий запис до бази даних (Рисунок 3.6).

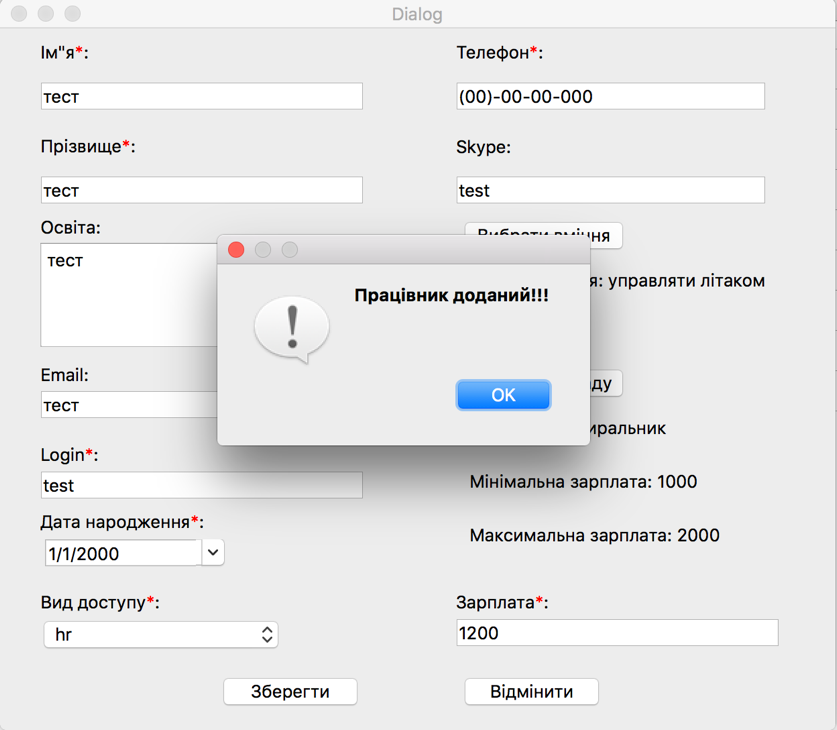


Рисунок 3.6 – Прийняти на роботу.

Для переведення з посади на посаду натискаємо кнопку «Перевести робітника» (Рисунок 3.7). Перевести можна лише співробітника, що на даний момент працює. Для того щоб перевести робітника, треба спочатку його вибрати з таблиці.

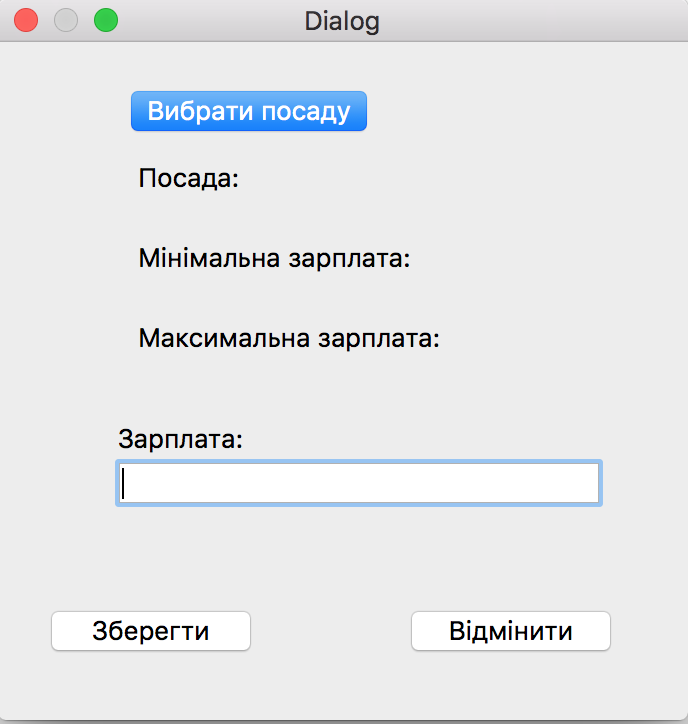


Рисунок 3.7 – Переведення.

Для того щоб завершити переведення робітника, треба обрати посаду і ставку. Ставка не може бути вище або менше ніж вказаний діапазон в БД. Якщо всі дії зроблені правильно , то працівника буде переведено (Рисунок 3.7).

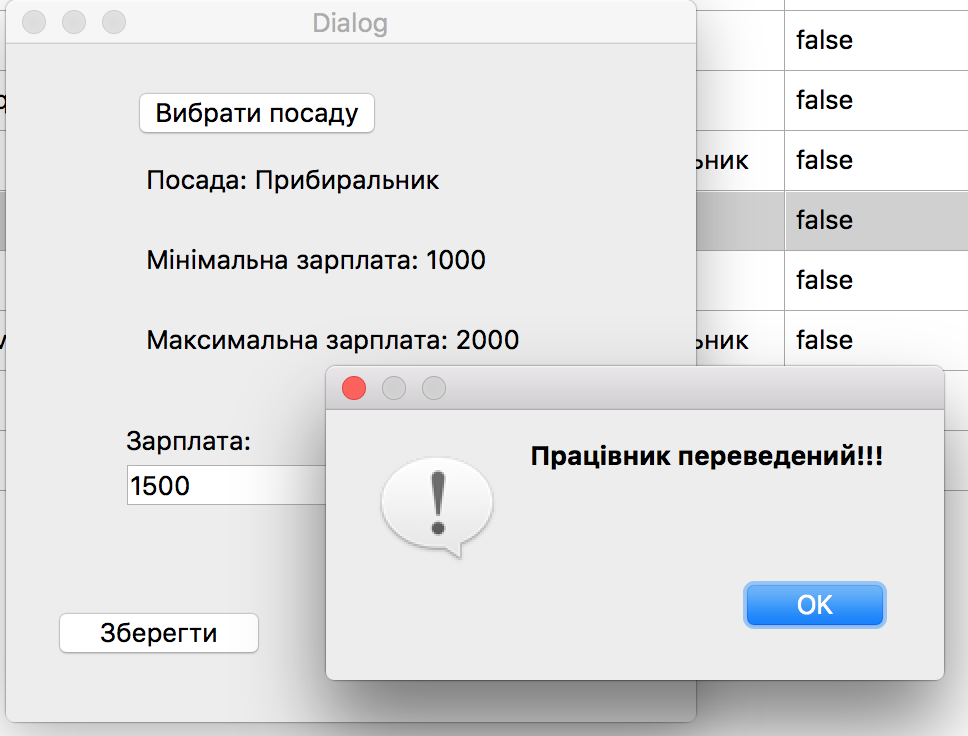
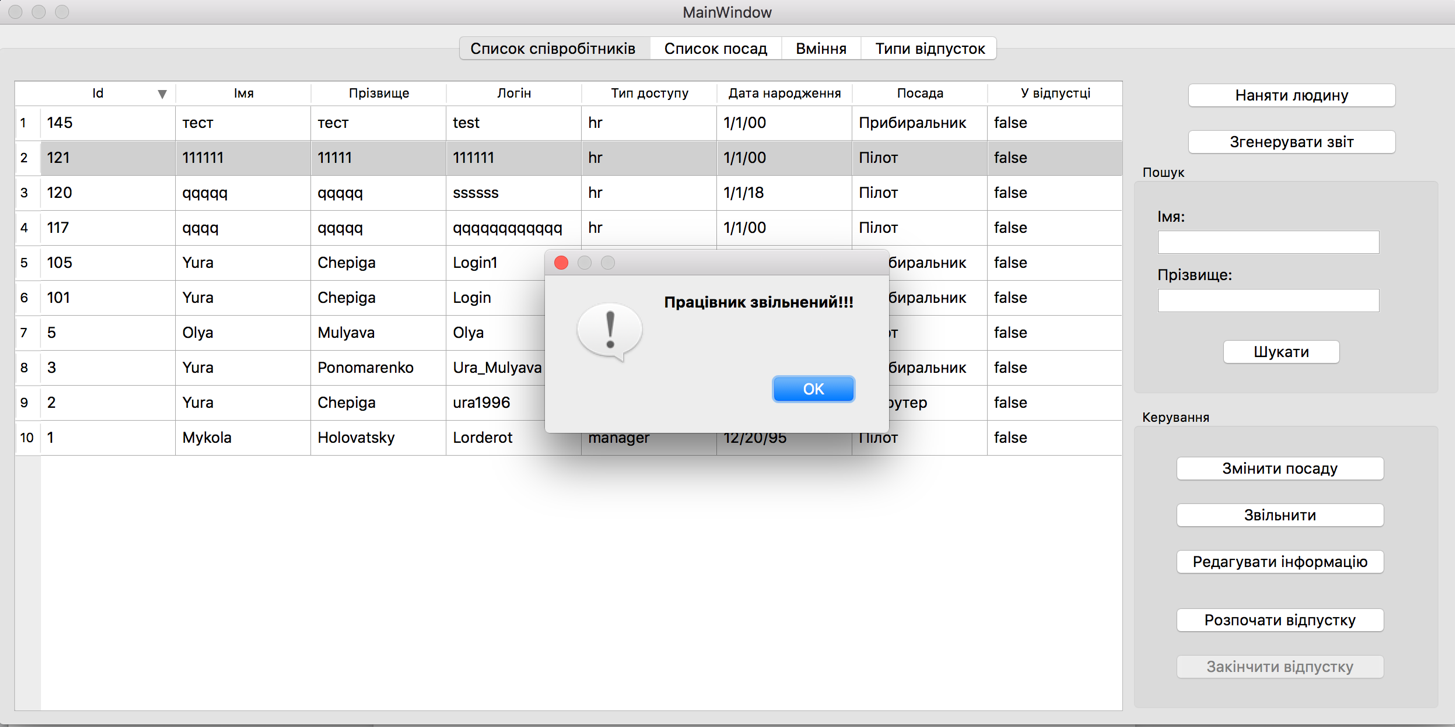


Рисунок 3.8 – Переведення.

Для звільнення з роботи натискаємо кнопку «Звільнити працівника» (Рисунок 3.9). Звільнити можна лише співробітника, що на даний момент працює.

Рисунок 3.9 – Звільнення робітника

Для того щоб додати відсутність треба натиснути кнопку «Розпочати відпустку». З’явиться вікно для вибору типу відпустки (Рисунок 3.10). Після того як ми вибрали тип відпустки натискаємо кнопку “Додати”. З цього моменту розпочинається відпустка.

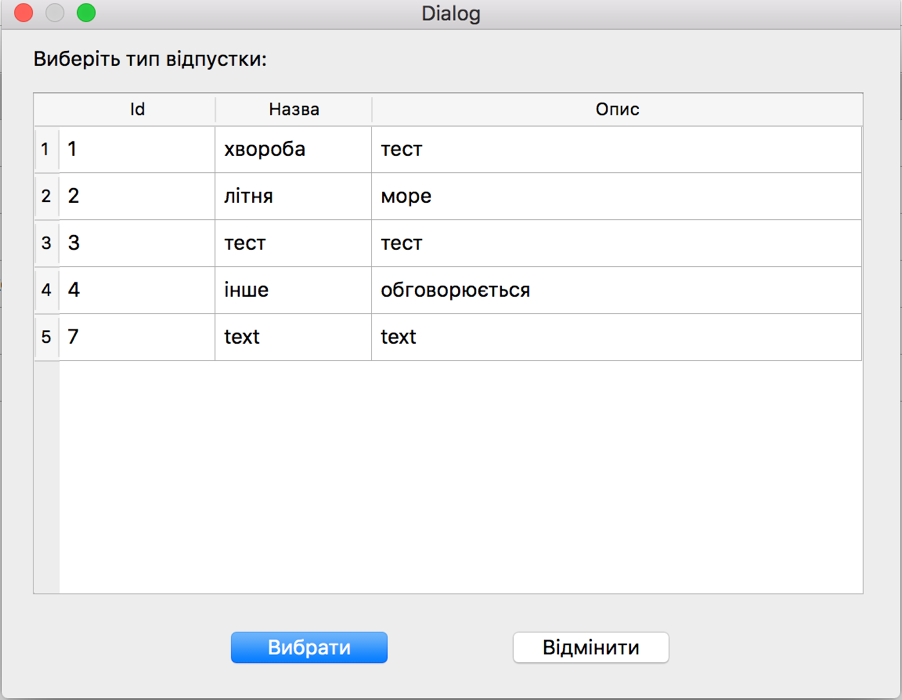
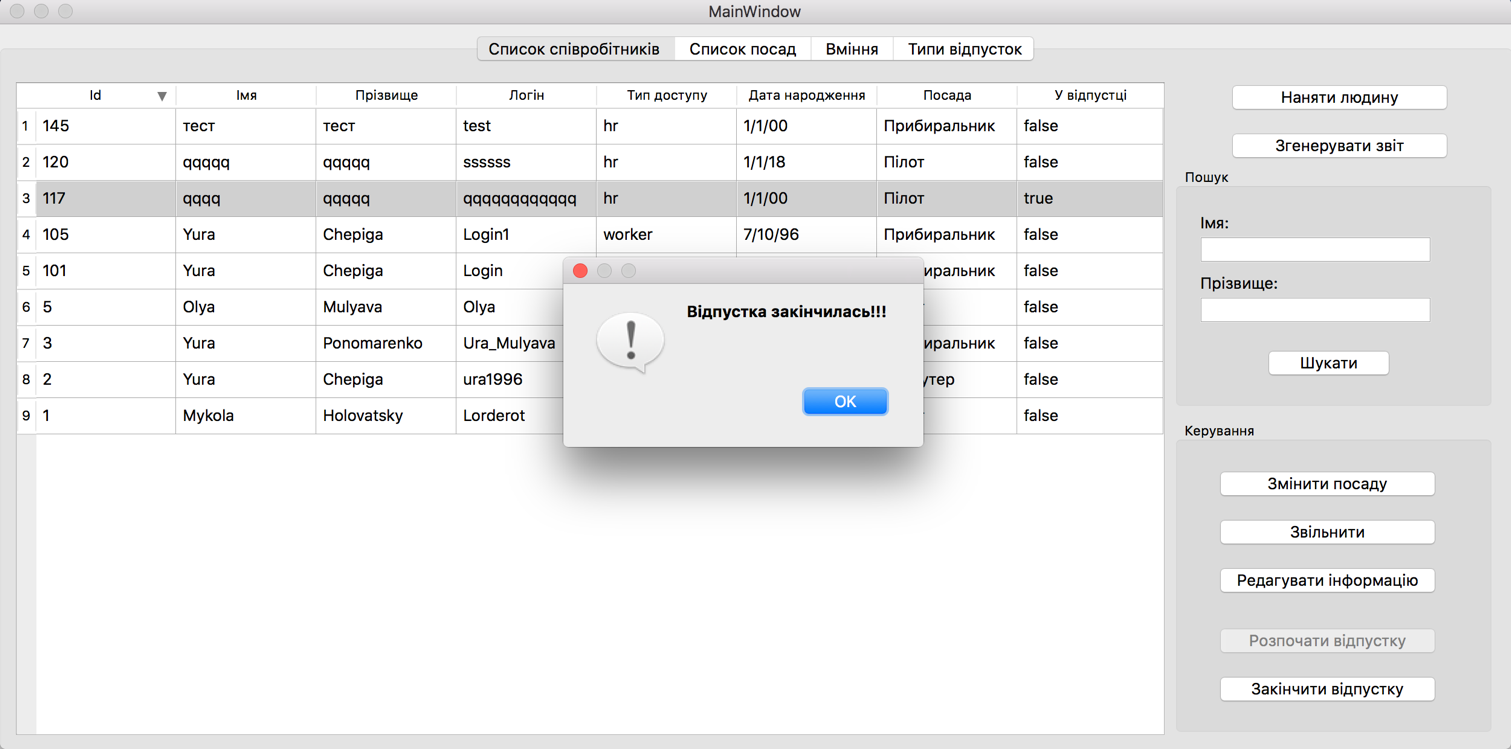


Рисунок 3.10 –Відпустка/лікарняний

Щоб закінчити відпустку натисніть “Закінчити відпустку” (Рисунок 3.11).

Рисунок 3.11 – Відпустка/лікарняний

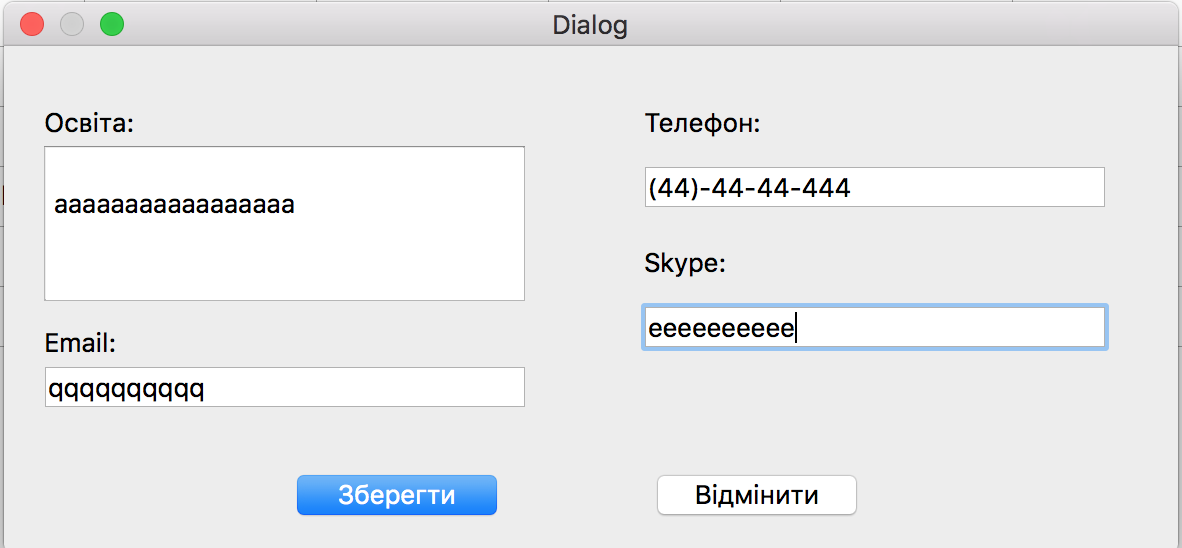
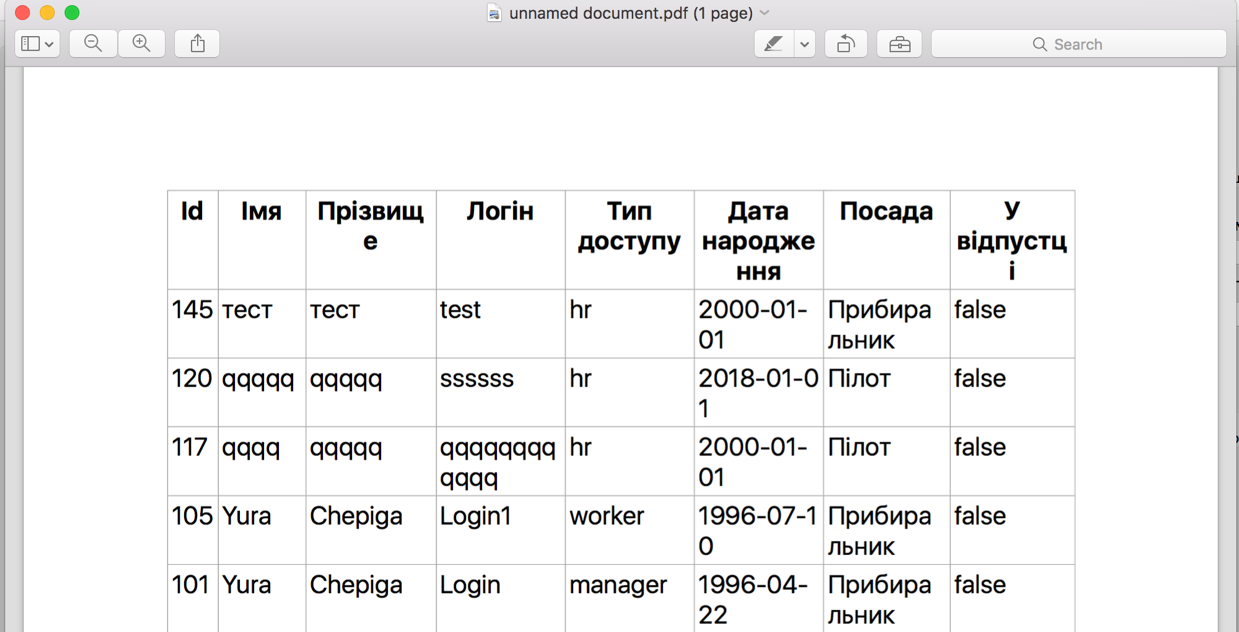
****Якщо потрібно змінити контактну інформацію про співробітника натисніть кнопку “Редагувати інформацію” (Рисунок 3.12). В цій вкладці можна змінити “Email”, “Skype”, “Телефон”, “Освіту”.

Рисунок 3.12 –Контактна інформація

Для того щоб роздрукувати табличку з співробітниками натисніть “Згенерувати звіт”. Після цього ви можете вибрати принтер або зберегти файл в форматі PDF (Рисунок 3.13).

Рисунок 3.13 –Друк звіту

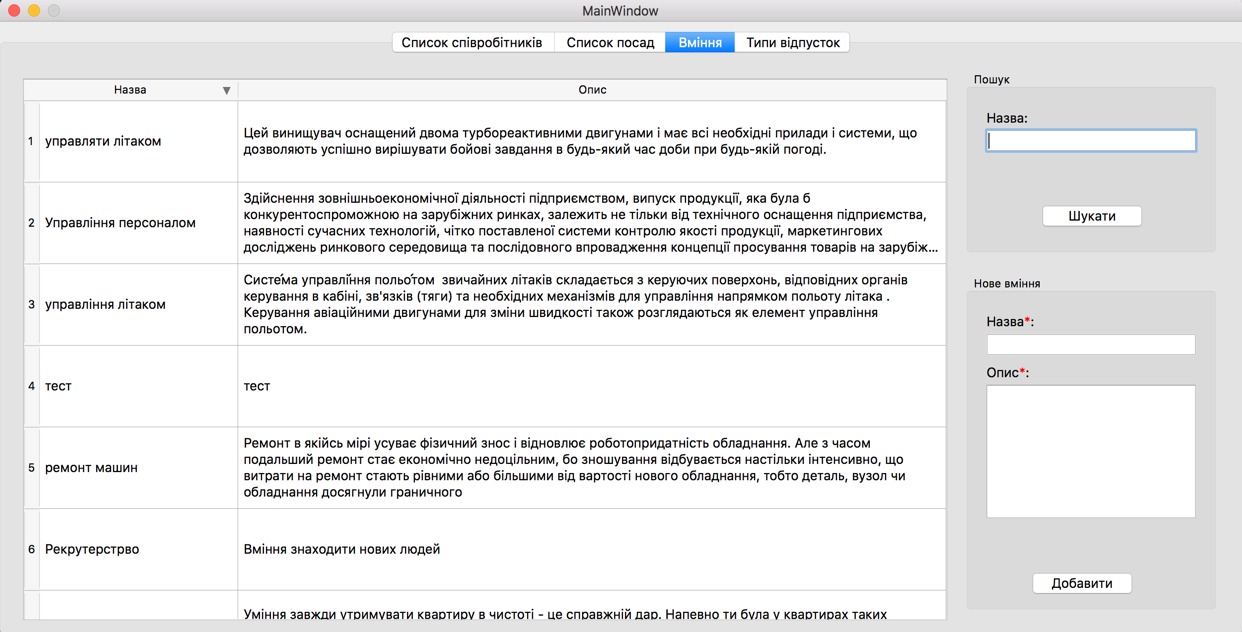
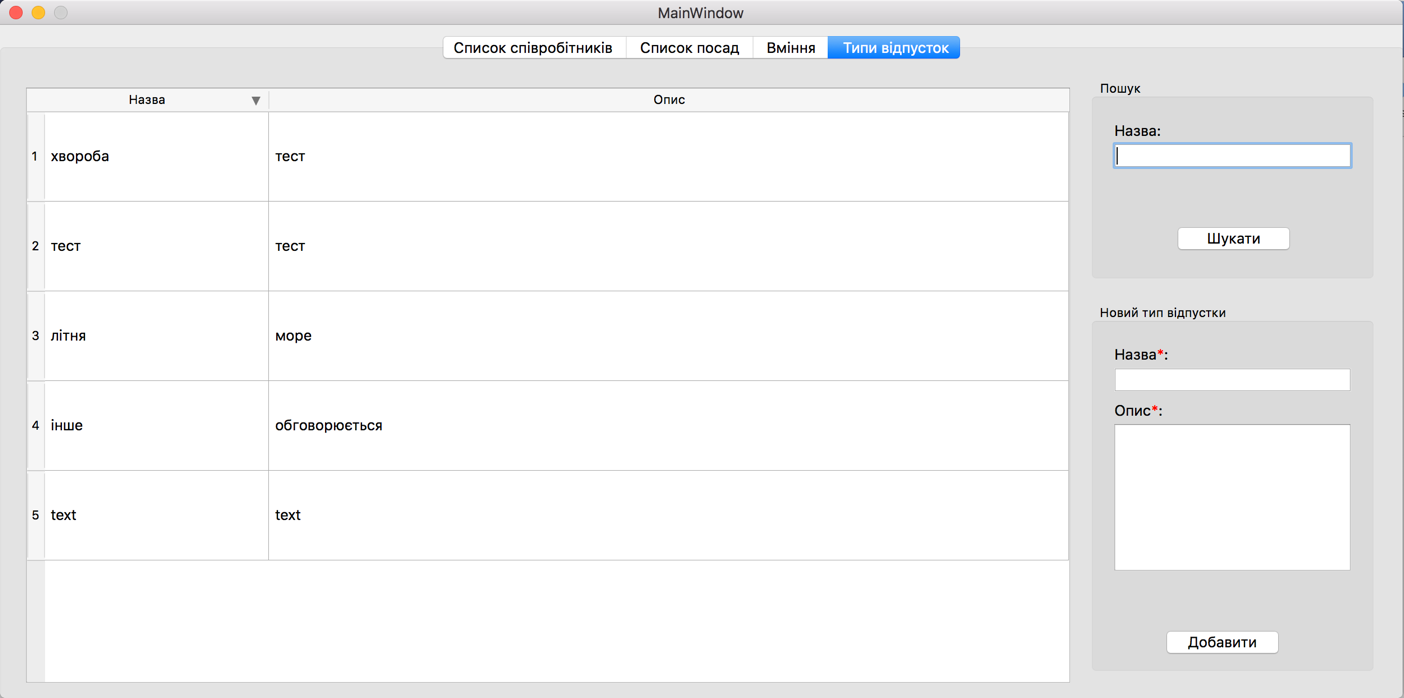
В системі є також користувачі з типом доступу “сео”, які можуть додавати і переглядати наявні вміння (Рисунок 3.14).

Рисунок 3.14 – Вміння

Також ці користувачі можуть переглядати і додавати типи відпусток (Рисунок 3.15).

Рисунок 3.15 – Типи відпусток

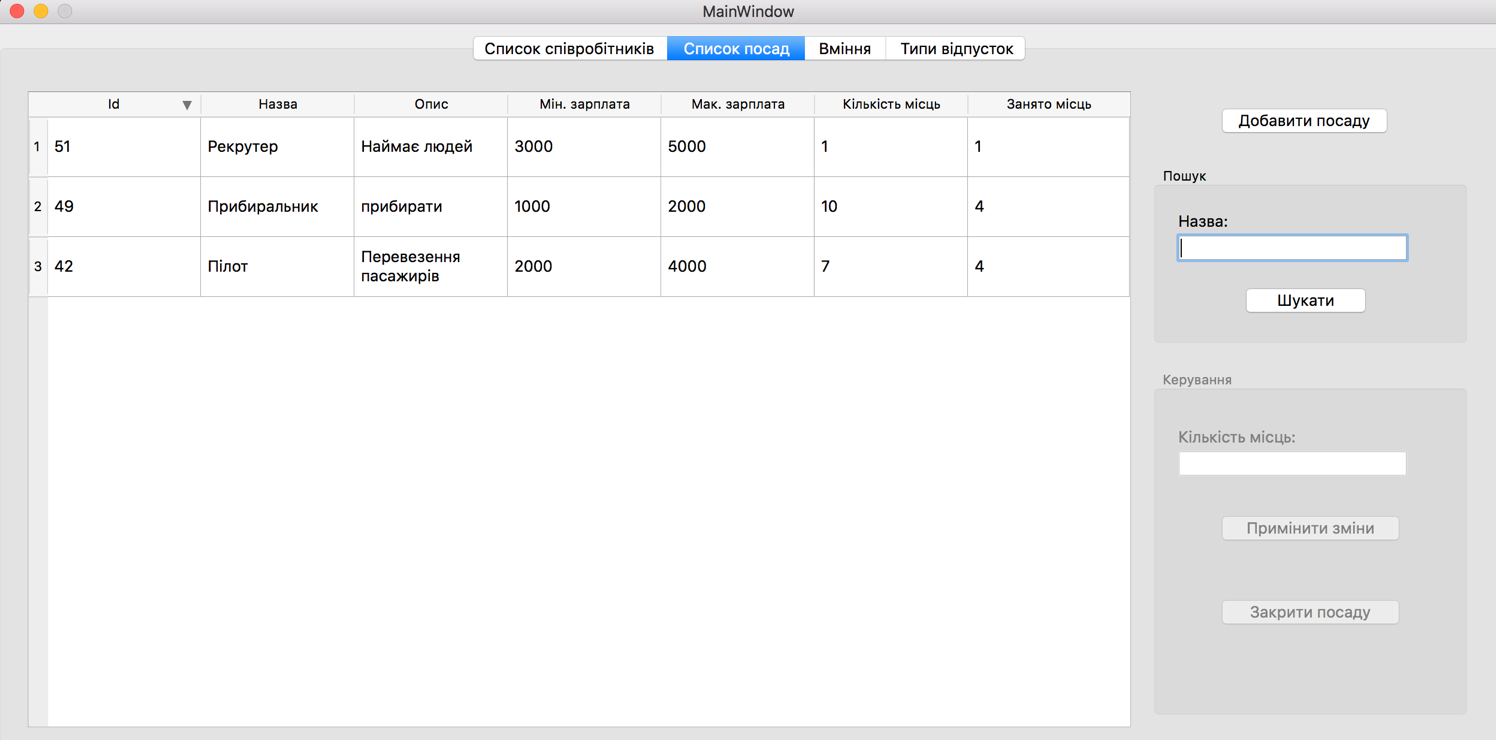
Користувачі з типом доступу “сео”, можуть змінювати структуру підприємства (Рисунок 3.16). Додавати нові посади, закривати існуючі(якщо там ніхто не працює), змінювати кількість працюючих на посаді.

Рисунок 3.16 – Управління структурою підприємства

# **Висновки**

Була реалізована АІС «Аеропорт».

Для відділу кадрів: співробітник HR-відділу має функції:

- перегляд даних про всю наявну інформацію в базі;

* додавання нового працівника;
* переведення робітника;
* звільнення людини з роботи;
* додавання відпусток/лікарняних.

Менеджер HR-відділу, крім функцій співробітника HR-відділу, має змогу редагувати деякі особисті дані працівника.

Список використаної літератури:

* 1. Конноллі Томас, Бегг Каролін. Бази даних: проектування, реалізація і супровід. [Теорія](http://ua-referat.com/%D0%A2%D0%B5%D0%BE%D1%80%D1%96%D1%8F) і практика, 3-е вид.: Пер. з англ. - М.: Видавничий дім «Вільямс», 2003. - 1440 с.: Іл. - Хрон. тит. англ.;
  2. Нейгел Крістіан, Івьен Білл, Глин Джей і ін C # 2005 для професіоналів. : Пер з англ. - М.: ТОВ «І.Д. Вільямс », 2006. - 1376 с.
  3. Миколаєва Н.А. [Мова](http://ua-referat.com/%D0%9C%D0%BE%D0%B2%D0%B0) структурованих запитів. Лабораторні роботи: навчальний посібник / Н.А. Миколаєва, Т.Ю. Калініна. - Ухта: УГТУ, 2006. - 124 с. мул.

**Текст програми**

Для розробки інформаційної системи було використано СУБД Postgres, яка дозволяє легко адмініструвати базу даних, зокрема створювати таблиці, тригери, процедури. Сама програма була реалізована на мові програмування – C++, що дозволяє користуватись даною програмою на будь-якій операційній системі.

***Користувач***

CREATE group hr;

GRANT SELECT on ALL TABLES IN SCHEMA public to hr;

GRANT INSERT, UPDATE on vacations, stuff, persons, con\_person\_skill to hr;

GRANT UPDATE on ALL SEQUENCES IN SCHEMA public to hr;

create user ura1996 in group hr password ‘1230’;

CREATE group ceo;

GRANT SELECT on ALL TABLES IN SCHEMA public to ceo;

GRANT INSERT, UPDATE on vacation\_type, posts, structures, skills, con\_posts\_skills to ceo;

GRANT UPDATE on ALL SEQUENCES IN SCHEMA public to ceo;

create user ura1997 in group ceo password ‘1230’;

***Процедура додавання нового співробітника***

CREATE OR REPLACE FUNCTION public.create\_person(\_first\_name character varying, \_last\_name character varying, \_education text, \_salary integer, \_login character varying, \_email character varying, \_skype character varying, \_phone character varying, \_access\_type character varying, \_birthday date, \_post\_id integer, skill\_ids integer[])

RETURNS integer

LANGUAGE plpgsql

AS $function$

DECLARE

id int;

s\_id int;

skill\_id int;

temp\_skill skills;

temp\_post posts;

BEGIN

-- validation

FOREACH s\_id IN ARRAY skill\_ids

LOOP

BEGIN

SELECT \* INTO STRICT temp\_skill FROM skills WHERE skills.skill\_id = s\_id;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE EXCEPTION 'Вміння % не знайдено', s\_id;

END;

END LOOP;

BEGIN

SELECT \* INTO STRICT temp\_post FROM posts WHERE posts.post\_id = \_post\_id;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE EXCEPTION 'Посада % не знайдена', \_post\_id;

END;

-- insert

BEGIN

INSERT INTO persons(first\_name, last\_name, education, login, email, skype, phone\_number, access\_type, birthday, date\_in)

VALUES(\_first\_name, \_last\_name, \_education, \_login, \_email, \_skype, \_phone, \_access\_type, \_birthday, now()) RETURNING person\_id INTO id;

EXCEPTION

WHEN unique\_violation THEN

RAISE EXCEPTION 'Login % вже зайнятий', \_login;

END;

INSERT INTO stuff(post\_id, person\_id, date\_in, salary)

VALUES(\_post\_id, id, now(), \_salary);

FOREACH skill\_id IN ARRAY skill\_ids

LOOP

INSERT INTO con\_person\_skill(skill\_id, person\_id)

VALUES(skill\_id, id);

END LOOP;

RETURN id;

END;

$function$

***Процедура додавання нової посади***

CREATE OR REPLACE FUNCTION public.create\_post(\_name character varying, \_description text, \_min\_salary integer, \_max\_salary integer, \_amount integer, skill\_ids integer[])

RETURNS integer

LANGUAGE plpgsql

AS $function$

DECLARE

id int;

s\_id int;

skill\_id int;

rec skills;

BEGIN

-- validation

FOREACH s\_id IN ARRAY skill\_ids

LOOP

BEGIN

SELECT \* INTO STRICT rec FROM skills WHERE skills.skill\_id = s\_id;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE EXCEPTION 'Вміння % не знайдено', s\_id;

END;

END LOOP;

-- insert

INSERT INTO posts(name, description, max\_salary, min\_salary)

VALUES(\_name, \_description, \_max\_salary, \_min\_salary) RETURNING post\_id INTO id;

INSERT INTO structures(post\_id, date\_in, amount)

VALUES(id, now(), \_amount);

FOREACH skill\_id IN ARRAY skill\_ids

LOOP

INSERT INTO con\_posts\_skills(skill\_id, post\_id)

VALUES(skill\_id, id);

END LOOP;

RETURN id;

END;

$function$

***Процедура переведення на нову посаду***

CREATE OR REPLACE FUNCTION public.change\_person(\_salary integer, \_post\_id integer, \_person\_id integer)

RETURNS integer

LANGUAGE plpgsql

AS $function$

DECLARE

stuff\_id int;

temp\_post posts;

temp\_pers persons;

BEGIN

-- validation

BEGIN

SELECT \* INTO STRICT temp\_post FROM posts WHERE posts.post\_id = \_post\_id;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE EXCEPTION 'Посада % не знайдена', \_post\_id;

END;

BEGIN

SELECT \* INTO STRICT temp\_pers FROM persons WHERE persons.person\_id = \_person\_id;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE EXCEPTION 'Людина % не знайдена', \_person\_id;

END;

-- update

UPDATE stuff SET data\_out=now() WHERE data\_out is NULL and person\_id = \_person\_id;

INSERT INTO stuff(post\_id, person\_id, date\_in, salary)

VALUES(\_post\_id, \_person\_id, now(), \_salary);

RETURN stuff\_id;

END;

$function$

***Процедура звільнення співробітника***

CREATE OR REPLACE FUNCTION public.close\_person(\_person\_id integer)

RETURNS integer

LANGUAGE plpgsql

AS $function$

DECLARE

stuff\_id int;

temp\_st stuff;

temp\_pers persons;

BEGIN

-- validation

BEGIN

SELECT \* INTO STRICT temp\_pers FROM persons WHERE persons.person\_id = \_person\_id;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE EXCEPTION 'Людина % не знайдена', \_person\_id;

END;

BEGIN

SELECT \* INTO STRICT temp\_st FROM stuff WHERE stuff.person\_id = \_person\_id AND stuff.data\_out IS NULL ;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE EXCEPTION 'Людина % не працює', \_person\_id;

END;

-- update

UPDATE stuff SET data\_out=now() WHERE data\_out is NULL and person\_id = \_person\_id;

RETURN stuff\_id;

END;

$function$

***Процедура яка викликається трігером на створення нової посади***

CREATE OR REPLACE FUNCTION public.check\_stuff\_insert()

RETURNS trigger

LANGUAGE plpgsql

AS $function$

DECLARE

post posts;

am\_of\_stuff int;

am\_of\_stuct int;

BEGIN

SELECT \* INTO post FROM posts WHERE posts.post\_id = NEW.post\_id;

IF not NEW.salary BETWEEN post.min\_salary AND post.max\_salary THEN

RAISE EXCEPTION 'Зарплата має бути в діапазоні від % до % .', post.min\_salary, post.max\_salary;

END IF;

SELECT count(s.stuff\_id) INTO am\_of\_stuff FROM stuff as s WHERE s.post\_id = NEW.post\_id and s.data\_out is NULL;

SELECT amount INTO am\_of\_stuct FROM structures as s WHERE s.post\_id = NEW.post\_id and s.date\_out is NULL;

IF am\_of\_stuff >= am\_of\_stuct THEN

RAISE EXCEPTION 'Немає вільних місць на посаду % .', post.name;

END IF;

RETURN NEW;

END;

$function$

***Розріз даних(View) – список людей***

create view current\_workers as

SELECT p.person\_id, p.first\_name, p.last\_name, p.login, p.access\_type, p.birthday,

po.name AS is\_work,

NOT (v.vacation\_id IS null) AS is\_vacation

FROM persons p

LEFT JOIN vacations v ON v.person\_id=p.person\_id AND v.date\_out IS NULL

LEFT JOIN stuff s ON s.person\_id=p.person\_id

LEFT JOIN posts po ON po.post\_id=s.post\_id

WHERE s.data\_out IS NULL;