НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КПІ ім. Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра програмного забезпечення комп’ютерних систем

**КУРСОВА РОБОТА**

***з дисципліни "Основи програмування"***

Виконав: Щербина Вадим Олегович

Група: КП-61

Допущено до захисту

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2 семестр 2016/2017

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КПІ ім. Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра програмного забезпечення комп’ютерних систем

|  |  |
| --- | --- |
| Узгоджено    Керівник роботи    \_\_\_\_\_\_\_\_\_\_/Гадиняк Р.А./ | ЗАХИЩЕНА "\_\_"\_\_\_\_\_\_\_\_\_2017р.  з оцінкою\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_/Гадиняк Р.А./ |

**Система обліку лікарні**

Виконавець роботи

Щербина Вадим Олегович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_2017р.

**Вступний опис системи**

Темою цієї курсової роботи є створення програми системи обліку лікарні, яка допоможе будь-якому користувачеві легко знайти потрібну інформацію про будь-якого співробітника або пацієнта. Дана програма призначена для зберігання інформації про лікарів, пацієнтів лікарні, допоміжної інформації про розпорядок роботи лікарні (кабінети, графіки роботи) і прийоми хворих, з можливістю внесення даних, вибірки і зміни даних, виведення інформації в необхідному форматі. Лікарня - це така організація, яка працює з дуже великим обсягом інформації про співробітників і про пацієнтів. Лікарі завжди повинні стежити за своїми пацієнтами, і мати інформацію про курс лікування хворих. А керівництву лікарні необхідно мати інформацію про своїх співробітників для зручнішого управління робочим процесом. Для цього потрібна спільна база даних, що включає всю необхідну інформацію. Програма є дуже актуальною на сьогоднішній день, вона автоматизує роботу з базою даних і надає користувачеві зрозумілий і дружній інтерфейс.

Потужність бази даних зумовлена ​​можливістю її постійного поповнення новими даними, причому в необмеженій кількості інформації. Це є дуже зручним для користувача. Таким чином, створення бази даних, яка має такі властивості, завдання досить актуальне і корисне. Будь-яка організація потребує своєчасного доступу до інформації. Роль зберігачів інформації в сучасному світі найчастіше виконують бази даних. Бази даних надають можливість безпечно зберігати інформацію в структурованому вигляді і дають можливість доступу до неї. Практично будь-яка сучасна організація потребує базу даних, що задовольняє потреби в зберіганні і редагуванні даних.

**Інструменти розробки**

Використання стандартних бібліотек Qt:

#include <QDesktopWidget> - бібліотека фіксації вікна в центрі екрана  
#include "QtSql/QSqlDatabase" - бібліотека застосовувалась для підключення до бази данних  
#include "QSqlQuery" - бібліотека використовувалась для використання основних запитів SQL

#include "QtSql/QSqlError" - бібліотека використовувалась відображення помилок які відбувались при запитах до бази дани  
 #include <QDebug> - бібліотека використовувалась для перевірки роботи деяких функцій  
#include <QDateTime> - бібліотека використовувалась для знаходження поточної дати для додавання пацієнтів і видалення

#include <QTableWidget> - бібліотека використовувалась для коректного відображення даних у таблицях

#include <QCompleter> - бібліотека використовувалась для пошуку даних у таблицях

**Основний функціонал системи**

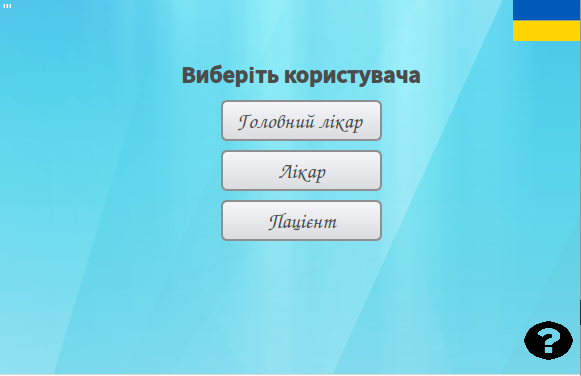
Всі виконані пункти із Технічного Завдання. Кожен пункт розкритий і описані деталі реалізації.

1. Розробка меню програми

2. Можливість вибору користувача для входу:

* Головний лікар
* Лікар
* Пацієнт

При в ході в програму буде головне меню в якого є можливість вибору користувача( головний лікар, лікар і пацієнт), також є можливість звернутись за допомогою до розробника даної програми.



3. Створення таблиць в яких буде відображуватись інформація

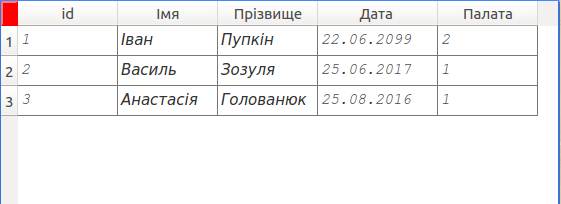
* Після вибору профілю “головного лікаря” в головному меню в користувача відкривається нове вікно в якому відображується табличка з усіма лікарями. Дані в цю таблицю завантажуються з бази даних за допомогою SQL запитів.



SQL запит:

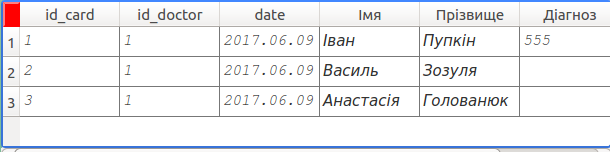
<SELECT DISTINCT id\_doctor,name,surname,position,cabinet,phone,login,password FROM doctor WHERE id\_doctor != 0>

* Після вибору профілю “лікаря” в головному меню в користувача відкривається нове вікно в якому відображується табличка з усіма пацієнтами. Дані в цю таблицю завантажуються з бази даних за допомогою SQL запиту.

SQL запит <SELECT DISTINCT id\_patient,name,surname,introduction\_date,palate FROM patient WHERE id\_patient!= 0 AND id\_doctor = 1>

* Також є декілька допоміжних таблиць які відображують деяку інформацію.

1. Таблиця виписаних пацієнтів



SQL запит <SELECT id\_card,report, diagnosis FROM observation ORDER BY id\_observ DESC>

4. Можливість додавати, змінювати і видаляти дані

Додавання, оновлення і видалення даних з SQL таблиці здійснюється за допомогою QSL запитів (“INSERT INTO” , “UPDATE”, “DELETE”). Після виконання цих дій дані в таблицях оновлюються.

5. Можливість перегляду інформації про старих пацієнтів.

* Лікар може переглянути дані про виписаних пацієнтів, які відображуються в певній таблиці.

6. Доступ клієнтами до графіку роботи лікарів.

* При перегляді інформації про певного лікаря пацієнту буде доступна інформація про години прийому даного лікаря.

7. Можливість записатись на прийом до лікаря на певний час

* Кожен клієнт коли буде записуватись на прийом до лікаря буде бачити години його прийому.

8. Перегляд кількості людей в певній палаті

* При додаванні пацієнта до певної палати, якщо всі місця в палаті занятті то лікаря буде проінформовано про це.

9. Пошук пацієнтів за певними ознаками( хворобами, місцем проживання, віком, прізвищем)

* Лікар може здійснити пошук пацієнтів за “Ім’я пацієнта”, “Прізвищем”

10. Пошук лікарів за спеціальністю, прізвищем, кабінетом.

11. Можливість перегляду історії хвороб певного пацієнта

* ЛІкар має змогу зайти і подивитись картку певного пацієнта в якій буде відображатись “Дата”, “Ліки”б “Аналізи” певного пацієнта.

12. Можливість клієнтом поставити рейтинг лікаря.

* Клієнт у вікні пошуку лікар може поставити рейтинг лікаря від 1 до 5 рейтинг лікаря зберігається в базі даних.

13. Створення звітів по певному лікареві або пацієнтові

* В лікаря є можливість створити звіт по виписаних пацієнтах в якому буде інформація про “Дату виписки”, “Ім’я”, “Прізвище”, “Хворобу”.
* В пацієнта є можливість створити звіт по власній картці в якому буде інформація про “Дату”, “Ліки”, “Аналізи”, “Нотатки лікаря”.

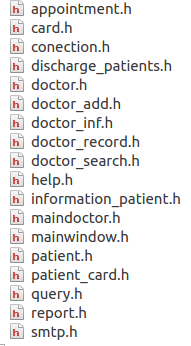
14. Можливість відправити данні звіти на пошту лікаря або на пошту пацієнта.

15. Можливість зв’язатись з розробником даної програми через пошту.

16. Реєстрація пацієнта і відправка даних для входу на пошту пацієнта.

**Опис модулів програми**

Проект розбитий на такі модулі:



Діаграма залежностей



**Модуль *Appointment***

Модуль містить інформацію про людей які записались на прийом до певного лікаря на певний день, яка відображується за допомогою вікна в якому є таблиця відсортована по часу.

**Модуль *Card***

Модуль дозволяє додавати дані до картки пацієнта. Ввід відбувається в певну форму, яка містить поля для вводу, потім ця інформація зберігається в базі даних.

**Модуль *Conection***

Модуль призначений для входу в свій обліковий запис, а також для реєстрації нового пацієнта. Це відбувається у вікні в якому є поля для вводу логіна, пароля, і id.

**Модуль *Discharge\_patients***

Модуль дозволяє переглядати виписаних пацієнтів у певного лікаря. Дані про пацієнтів будуть відображатись у “TableWidget”. Також дане вікно містить поле для вводу пошти на яку відправляється звіт.

**Модуль *Doctor***

Даний модуль містить інформацію про всіх пацієнтів у даного лікаря, ця інформація відображається у “TableWidget”. Також даний модуль має можливість додавання, видалення пацієнтів з таблиці, додавання певного пацієнта до певної палати.

**Модуль *Doctor\_add***

Модуль виконує операцію вводу даних про нового лікаря і додавання лікаря до бази даних.

**Модуль *Doctor\_inf***

Модуль призначений для пошуку лікаря і відображення інформації про нього в певній формі. Також даний модуль відображає рейтинг лікаря.

**Модуль *Doctor\_record***

Модуль призначений для запису на прийом до певного лікаря. Запис відбувається за допомогою вибору дати і часу у спеціальному вікні, які потім зберігаються в базі даних.

**Модуль *Help***

Даний модуль надає можливість користувачу зв’язатись з розробником даної програми, інформація для відправки повідомлення береться з форми яка розташована на вікні.

**Модуль Information\_patient**

Даний модуль призначений для вводу інформації про пацієнта, яка в подальшому буде добавлена в базу даних.

**Модуль *Maindoctor***

Модуль має можливість додавати змінювати і видаляти інформацію про лікарів, яка зберігається в базі даних і відображається в таблиці.

**Модуль *Mainwindow***

Модуль слугує для вибору користувача для входу в свій обліковий запис.

**Модуль *Patient***

Даний модуль слугує для відображення дій, які може виконати пацієнт(“знайти певного лікаря”, “відправити звіт”, “записатись до лікаря”).

**Модуль *Patient\_card***

Модуль слугує для відображення інформації про картку пацієнта в “QtableWidget”.

**Модуль *Query***

Цей модуль виконує всі запити до бази даних.

**Модуль *Report***

Модуль відповідає за створення звітів(виписані пацієнти або картка пацієнта)

**Модуль *SMTP***

Даний модуль відповідає за відправку повідомлень по пошті.

**Організація даних**

Тут опис обраних файлових форматів або структура бази даних.

“Doctor”

CREATE TABLE `doctor` (

`id\_doctor` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,

`name` TEXT NOT NULL,

`surname` TEXT,

`id\_departament` INTEGER,

`position` TEXT,

`cabinet` INTEGER,

`phone` TEXT,

`login` TEXT,

`password` TEXT,

`office\_hours` TEXT,

`rating` REAL,

`atempt` INTEGER,

`start\_work` TEXT,

`end\_work` TEXT,

FOREIGN KEY(`id\_departament`) REFERENCES department ( "id\_departament" ) ON DELETE CASCADE ON UPDATE CASCADE

);

“Doctor\_record”

CREATE TABLE `doctor\_record` (

`id\_record` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,

`id\_doctor` INTEGER,

`id\_patient` INTEGER,

`date` TEXT,

`time` TEXT,

`name` TEXT,

`surname` TEXT

);

“Observation”

CREATE TABLE `observation` (

`id\_observ` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,

`id\_card` INTEGER NOT NULL,

`date` TEXT NOT NULL,

`medicament` TEXT NOT NULL,

`analysis` TEXT NOT NULL,

`report` TEXT NOT NULL,

`diagnosis` TEXT NOT NULL,

FOREIGN KEY(`id\_card`) REFERENCES card ( "id\_card" ) ON DELETE CASCADE ON UPDATE CASCADE

);

“Palate”

CREATE TABLE `palate` (

`id\_palate` INTEGER NOT NULL UNIQUE,

`no\_palate` INTEGER NOT NULL,

`num\_place` INTEGER NOT NULL,

PRIMARY KEY(id\_palate)

);

“Patient”

CREATE TABLE `patient` (

`id\_patient` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,

`name` TEXT NOT NULL,

`surname` TEXT NOT NULL,

`birth\_date` TEXT NOT NULL,

`address` TEXT NOT NULL,

`phone\_number` TEXT NOT NULL,

`introduction\_date` TEXT NOT NULL,

`notes` TEXT NOT NULL,

`id\_doctor` INT,

`login` TEXT,

`password` TEXT,

`status` TEXT,

`palate` TEXT

);

“Card”

CREATE TABLE `card` (

`id\_card` INTEGER NOT NULL UNIQUE,

`id\_patient` INTEGER NOT NULL UNIQUE,

`date\_creation` TEXT,

`palate` INTEGER,

`status` TEXT,

PRIMARY KEY(id\_card),

FOREIGN KEY(`id\_patient`) REFERENCES patient ( "id\_patient" ) ON DELETE CASCADE ON UPDATE CASCADE

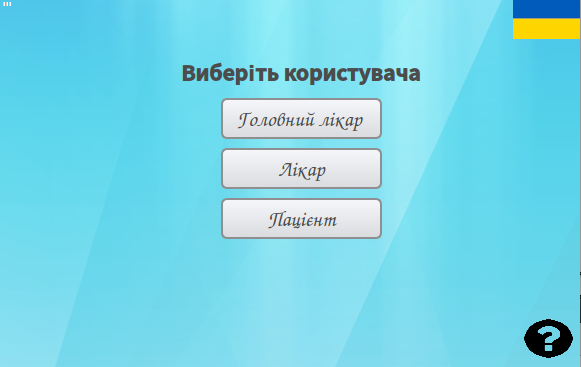
);

**Опис найважливіших алгоритмів**

Опис найважливіших алгоритмів

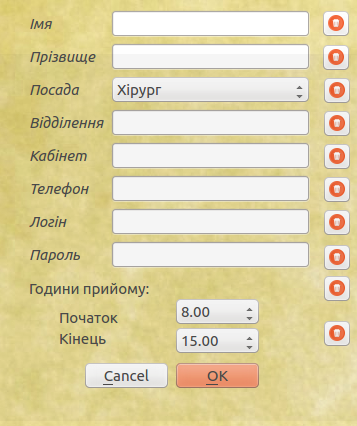
**Результати роботи**

Меню вибору користувача

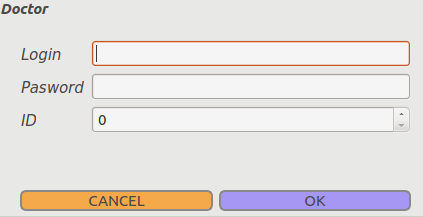


Вікно головного лікаря

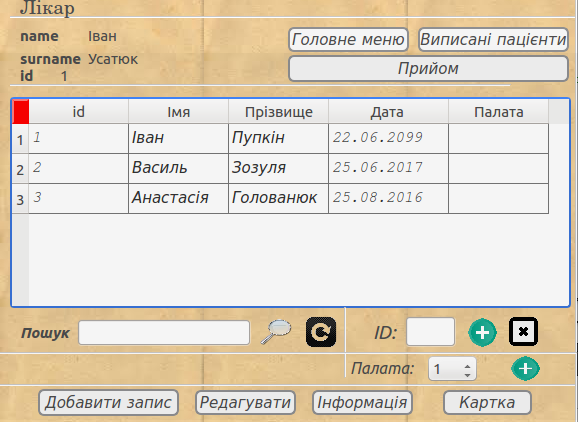
Вікно додавання нового лікаря



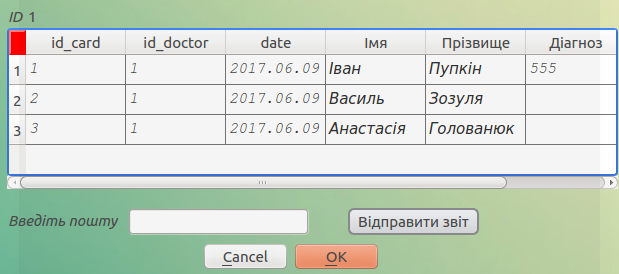
Вікно авторизації

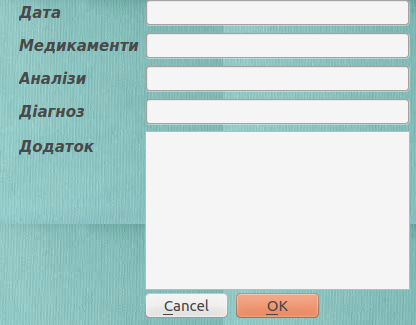


Вікно лікаря

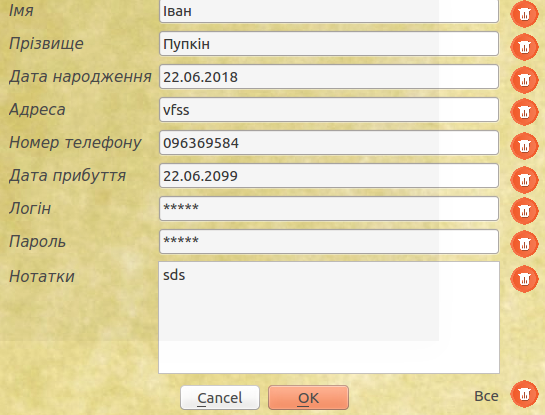


Вікно з виписаними пацієнтами

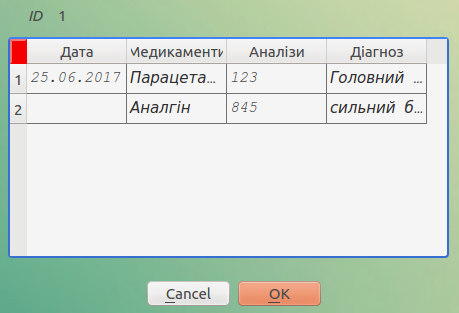


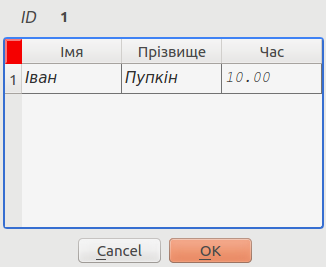
Вікно додавання нового запису до картки

Вікно редагування інформації про пацієнта

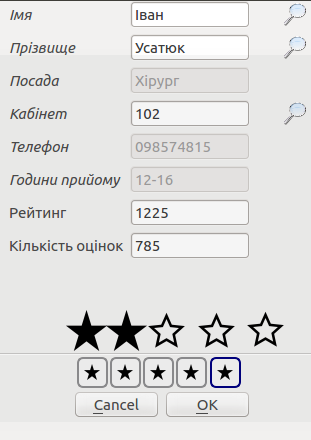


Вікно перегляду картки пацієнта



Вікно перегляду записів до лікаря 

Вікно пацієнта

Вікно пошуку лікаря 

Вікно запису на прийом до лікаря



**Мануал для користувача**

1. В головному меню є вибір користувача(головний лікар, лікар, пацієнт) і значок допомоги.
2. Якщо ви вибрали Допомогу, ви можете написати розробнику програмі.
3. Якщо ви вибрали Головного лікаря:

* у вас є можливість добавляти, видаляти і редагувати інформацію про лікарів.
* також в поле для вводу тексту можна ввести Ім’я лікаря, ID, кабінет і нажати кнопку шукати і ви знайдете потрібного лікаря

4) Якщо ви вибрали Лікаря у вас появиться вікно авторизації після чого ви зайдете в профіль лікаря де можна буде:

* Добавити пацієнта. Для цього потрібно вибрати його id із списку і нажати на кнопку Добавити(+).
* Видалити пацієнта. Для цього потрібно вибрати пацієнта з таблиці і нажати на кнопку Видалення.
* Добавити запис в картку пацієнта. Для цього потрібно нажати на кнопку “Добавити запис” і ввести всі потрібні дані.
* Редагувати інформацію про пацієнта. Для цього потрібно вибрати пацієнта і нажати на кнопку “Редагувати”.
* Подивитись інформацію про пацієнта. Для цього потрібно вибрати пацієнта і нажати на “Інформація”.
* Переглянути записи в картці пацієнта. Для цього потрібно вибрати пацієнта і нажати на “Картка”.
* Подивитись список пацієнтів на прийом на сьогоднішній день. Для цього потрібно нажати на “Прийом”.
* Подивитись список виписаних пацієнтів. Для цього потрібно нажати на “Виписані пацієнти”. В цьому вікні є можливість відправити всю інформацію про виписаних пацієнтів на пошту.
* Добавити певного пацієнта в палату. Для цього потрібно вибрати пацієнта з списка і вибрати номер палати і нажати на значок добавити. Якщо палата буде заповнена лікаря буде проінформовано.

5) Якщо ви вибрали Пацієнта у вас появиться вікно авторизації в якому можна ввести свої дані для входу, або зареєструватись. Після входу в свій профіль ви можете:

* Найти інформацію по певному лікарю, нажавши на “Пошук лікаря”.
* Переглянути свою картку, нажавши на “Карта Пацієнта”.
* Ввести емейл і відправити звіт собі на пошту. В звіті будуть всі записи з вашої картки.
* Записатись і відмінити запис до лікаря. У вікні запису до лікаря вибрати лікаря дату і час на який записатись.