НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«Київський політехнічний інститут імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра програмного забезпечення комп’ютерних систем

**КУРСОВА РОБОТА**

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

Виконав: Мортіков Владислав Євгенович

Група: КП-83

Допущено до захисту

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

2 семестр 2018/2019

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«Київський політехнічний інститут імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра програмного забезпечення комп’ютерних систем

|  |  |
| --- | --- |
| Узгоджено  Керівник роботи    \_\_\_\_\_\_\_\_\_\_/Гадиняк Р.А./ | ЗАХИЩЕНА  "\_\_"\_\_\_\_\_\_\_\_\_2019р.  з оцінкою \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_/Гадиняк Р.А./ |

**Програмний додаток з графічним інтерфейсом користувача для керування даними керування даними обліку навчальної діяльності викладачів університету**

Виконавець роботи

Мортіков Владислав Євгенович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019р.

Зміст

[**Вступ**](#_cr2eoqpdh4mm) **4**

[**1. Аналіз предметної галузі**](#_b73pd3816f8p) **5**

[1.1. Аналіз вимог до функціональності програмних засобів](#_qz6vreit6wm5) 5

[**2. Аналіз мов програмування та технологій розроблення**](#_ewtylcuh8bvc) **5**

[2.1. Мова програмування С++](#_fh0j2unhluf2) 5

[2.2. Фреймворк Qt](#_ayd32pcv0jlc) 5

[2.3. База даних SQLite](#_hul0g052l08h) 5

[2.4. Бібліотека ….](#_6jqqk9eelxe) 5

[**3. Опис розроблених програмних засобів**](#_kbi4mqs5pjyt) **5**

[3.1. Структура бази даних](#_ny4p6qq0v8a9) 5

[3.2. Модуль сховища даних](#_vkb2811r6l9f) 6

[**4. Аналіз розроблених програмних засобів**](#_5vawzkwfp9ab) **6**

[4.1. Особливості реалізації](#_5ajysoaxe2s5) 6

[4.2. Дизайн та вміст вікон](#_c8eyxdz8sp3u) 9

[**Висновки**](#_6dbqg2jsslr5) **10**

Вступ

Наш час вимагає автоматизації структур підприємств і організацій задля успішної роботи в умовах ІV науково-технічної революції. Так електронний документ, що конче потребує філіал фірми з США в Україні може надійти за декілька секунд з Нью-Йорка до Києва та з паперовими такого не вийде, маємо чекати більше часу: тижні, місяці.Так само, як до документів, ми потребуємо швидкого доступу до будь-якої потрібної інформації, у тому числі освітньої.

Якість освіти в Україні, зокрема вищої, залишається, на думку деяких експертів, незадовільною у тому числі через те, що студенти не мають доступу до навчальних програм, не знають актуальних оцінок.

На жаль, університети України, у тому числі НТУУ “КПІ ім.І.Сікорського”, далекі від повної автоматизації навчального процесу. Більшість викладачів спілкується зі студентами за допомогою електронної пошти, іноді надсилаючи певні файли. Але часто досить важко структурувати і проаналізувати інформацію через її надлишок.

Інші педагоги створюють сторінки на певних ресурсах (“ПІАЗЗА”, викл. Сулема Ольга Костянтинівна) або використовуючи самописні сайти (“ПРОҐБЕЙЗ”, викл. Гадиняк Руслан Анатолійович). Але ця практика не є поширеною серед викладачів із зрозумілих причин.

Саме тому, створення системи, що забезпечує автоматичне розподілення лекцій та їхніх курсів, є актуальною задачею.

Дану курсову роботу присвячено спробі розробки додатку з графічним інтерфейсом користувача, який призначено для управління курсами лекцій та їхніх тем, матеріалів.

# 1. Аналіз предметної галузі

# 1.1. Аналіз вимог до функціональності програмних засобів

В процесі аналізу вимог до системи було виділено такі функціональні вимоги (Список вимог з ТЗ):

Перед входом у систему користувач проходить аутентифікацію.

Система повинна забезпечувати одночасну роботу з одним сервером мінімум двох користувачів.

## 2.1. Вимоги до функціональних характеристик

Програма має задовольняти такі вимоги:

1. Наявний графічний інтерфейс користувача. Графічні елементи використані відповідно до їхнього основного призначення.
   1. Наявно декілька вікон і діалогів. Для взаємодії із файловою системою використовуються стандартні діалоги
   2. Дотримується єдиний стиль для всіх вікон
2. Наявні форми для створення і редагування даних.
   1. Можна видаляти дані (з підтвердженням цієї дії).
   2. Наявна форма редагування даних має мати вже заповнені поля із попередніми значеннями. Є можливість відмінити оновлення сутності.
3. Використання графічних списків та таблиць для відображення колекцій даних.
   1. Дані в таблиці не є лише відображенням однієї таблиці БД, а містять склеєні чи змінені дані.
   2. Дані у списках та таблицях пагінуються, доступний пошук та фільтрація їх записів.
4. Всі вхідні дані від користувача та з файлів перевіряються, помилки обробляються, користувачу відображається інформація про помилки.
5. Головне вікно (або інші вікна) мають меню із основними діями програми. Дії прив'язані до відповідних гарячих клавіш.
6. Програма має можливість завантаження зображень з файлової системи, їх збереження у сховище даних та їх відображення на графічних елементах.
7. Програма має можливість експорту-імпорту даних у одному із форматів даних (CSV\XML). Є можливість зберігання даних у файлах, та передача даних до інших процесів через протокол TCP.
8. Програма взаємодіє з реляційною базою даних. SQL запити до бази даних не конкатенуються, а їх виконання винесено у спеціальний окремий модуль (вид сховища).
   1. База даних містить мінімум 3 таблиці. Таблиці в БД приведені до 1-ї та 2-ї нормальних форм.
   2. Між таблицями бази даних є one-to-many та many-to-many зв'язки
9. Наявна реєстрація користувачів у системі. Паролі користувачів не зберігаються у відкритому виді, а хешуються обраним алгоритмом хешування рядків.
   1. Наявна форма аутентифікації. Пароль користувача при введені у форму прихований (використано спеціальний елемент). Вікно цієї форми має сповіщення про некоректність введених користувачем даних.
   2. Програма авторизує доступу до даних по користувачах. У базі є дані, що належать конкретному користувачу (напр, авторство сутностей).
10. Використовуються можливості подійно-орієнтованого програмування (Qt signals-slots).
11. Частина проекту винесена у власну статичну чи динамічну бібліотеку. Винесення саме цього коду обгрунтоване.
12. Використання клієнт-серверної архітектури програм та TCP сокетів (розділення програми на клієнт і сервер з передачею XML\JSON даних)
13. Використання протоколу HTTP та зовнішніх HTTP API для отримання даних. Або створення власного HTTP API сервера.

## 2.2. Вимоги до надійності

### 2.2.1. Відмови через некоректні дії користувачів системи

Відмови програми внаслідок некоректних дій користувача при взаємодії з програмою через графічний інтерфейс неприпустимі.

# 2. Аналіз мов програмування та технологій розробки

# 2.1. Мова програмування С++

**C++** (**Сі-плюс-плюс**) — мова програмування високого рівня з підтримкою кількох парадигм програмування: об'єктно-орієнтованої, узагальненої та процедурної. Базується на мові С.

## 2.2. Фреймворк Qt

**Qt** (**К'ют**) — крос-платформенний інструментарій розробки програмного забезпечення (ПЗ) мовою програмування C++. Дозволяє запускати написане за його допомогою ПЗ на більшості сучасних операційних систем (ОС), просто компілюючи текст програми для кожної операційної системи без зміни коду. Містить всі основні класи, які можуть бути потрібні для розробки прикладного програмного забезпечення, починаючи з елементів графічного інтерфейсу й закінчуючи класами для роботи з мережею, базами даних, XML. Бібліотека дозволяє керувати нитками, працювати з мережею та забезпечує крос-платформенний доступ до файлів.

## 2.3. База даних SQLite

**SQLite** **(ес-к’ю-лайт)** — полегшена реляційна система керування базами даних. Втілена у вигляді бібліотеки, де реалізовано багато зі стандарту SQL-92. Сирцевий код SQLite поширюється як суспільне надбання, тобто може використовуватися без обмежень та безоплатно з будь-якою метою. Фінансову підтримку розробників SQLite здійснює спеціально створений консорціум, до якого входять такі компанії, як Adobe, Oracle, Mozilla, Nokia, Bentley і Bloomberg. З 2018р SQLite, як й JSON та CSV, рекомендований Бібліотекою Конгресу США формат зберігання структурованого набору даних.

# 3. Опис розроблених програмних засобів

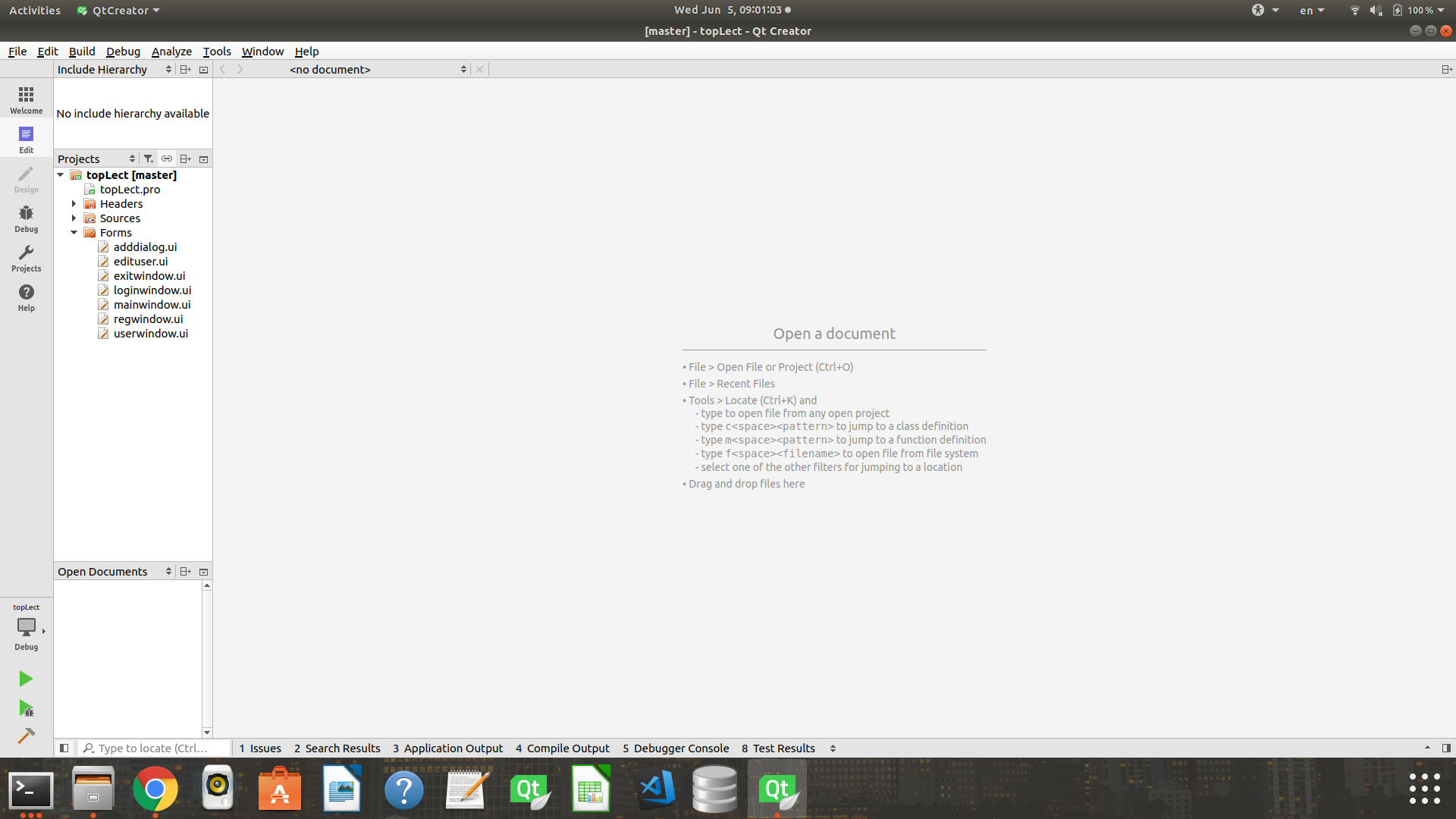
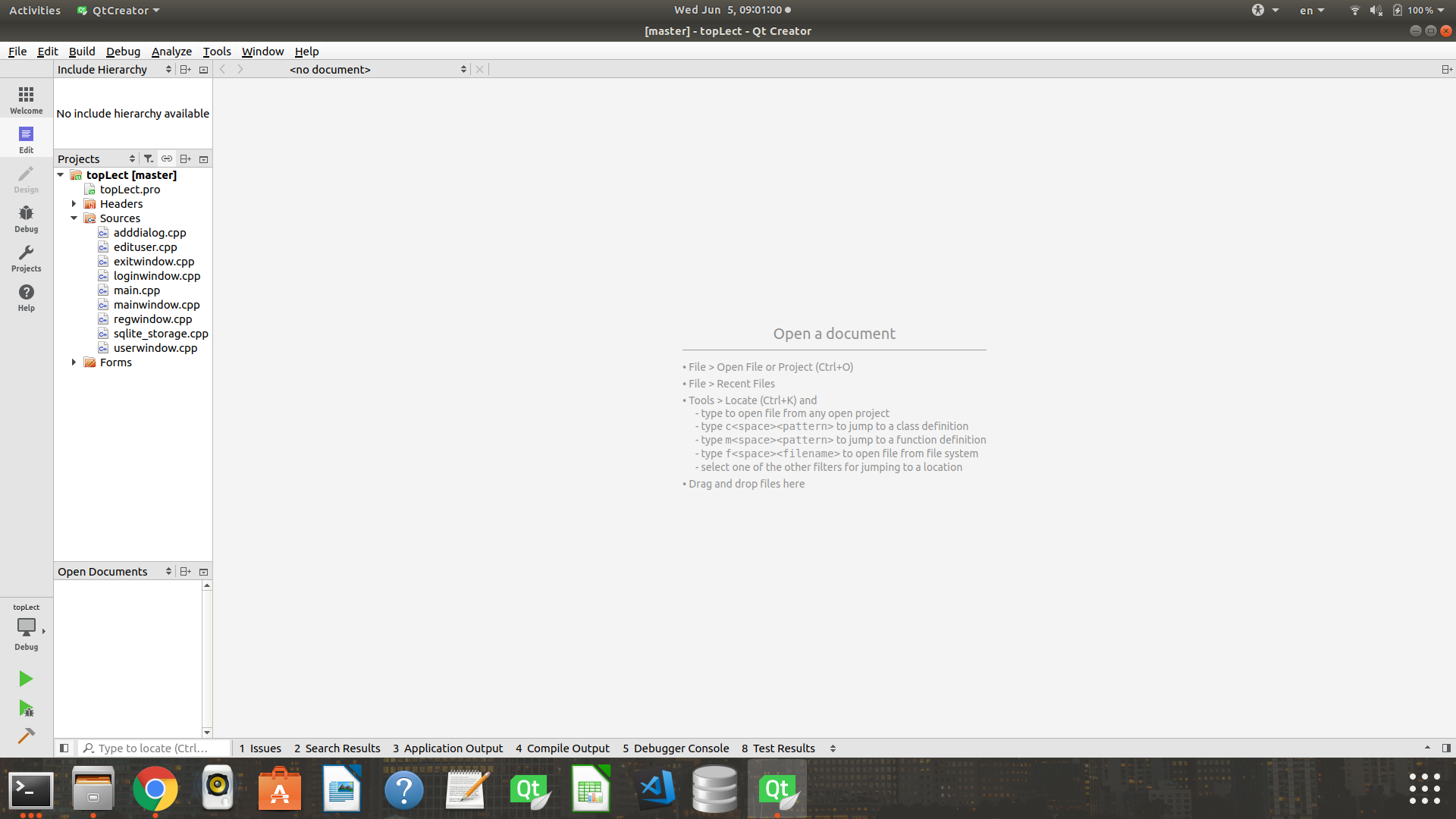
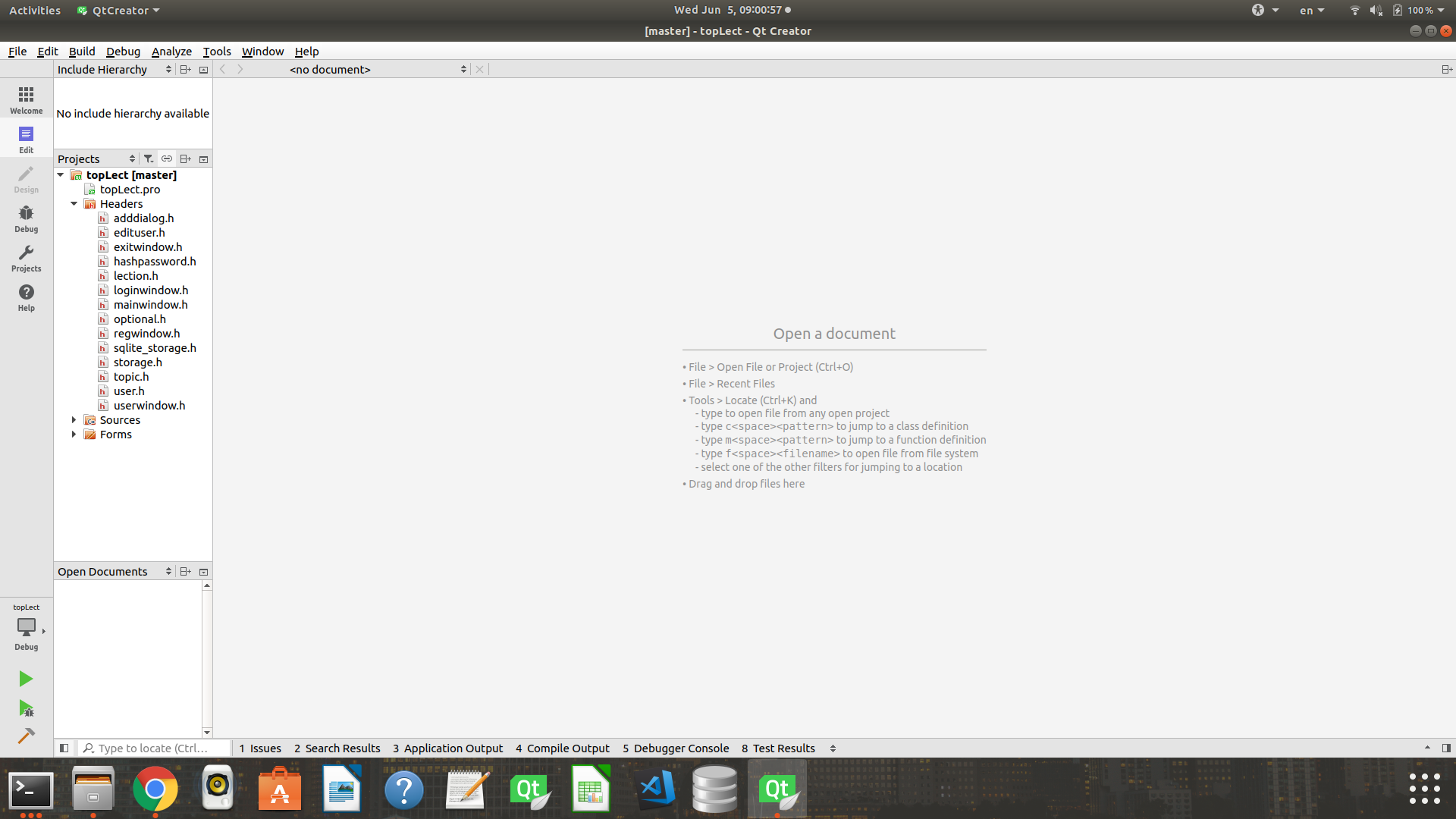
# 4. Аналіз розроблених програмних засобів

## 4.1. Особливості реалізації

Програмні засоби реалізовані у вигляді десктопного додатку з графічним інтерфейсом користувача.

Сервер відсутній.

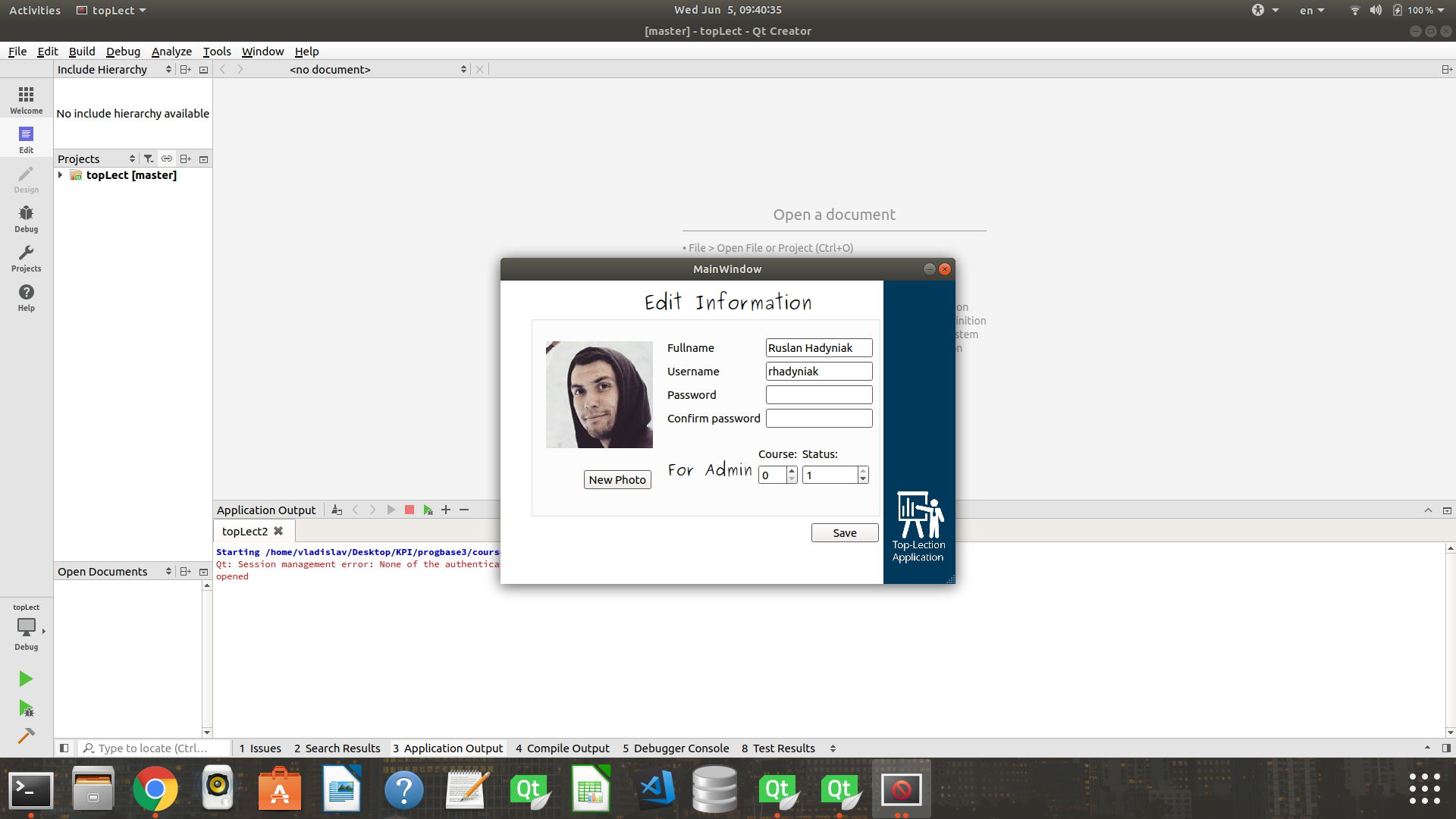
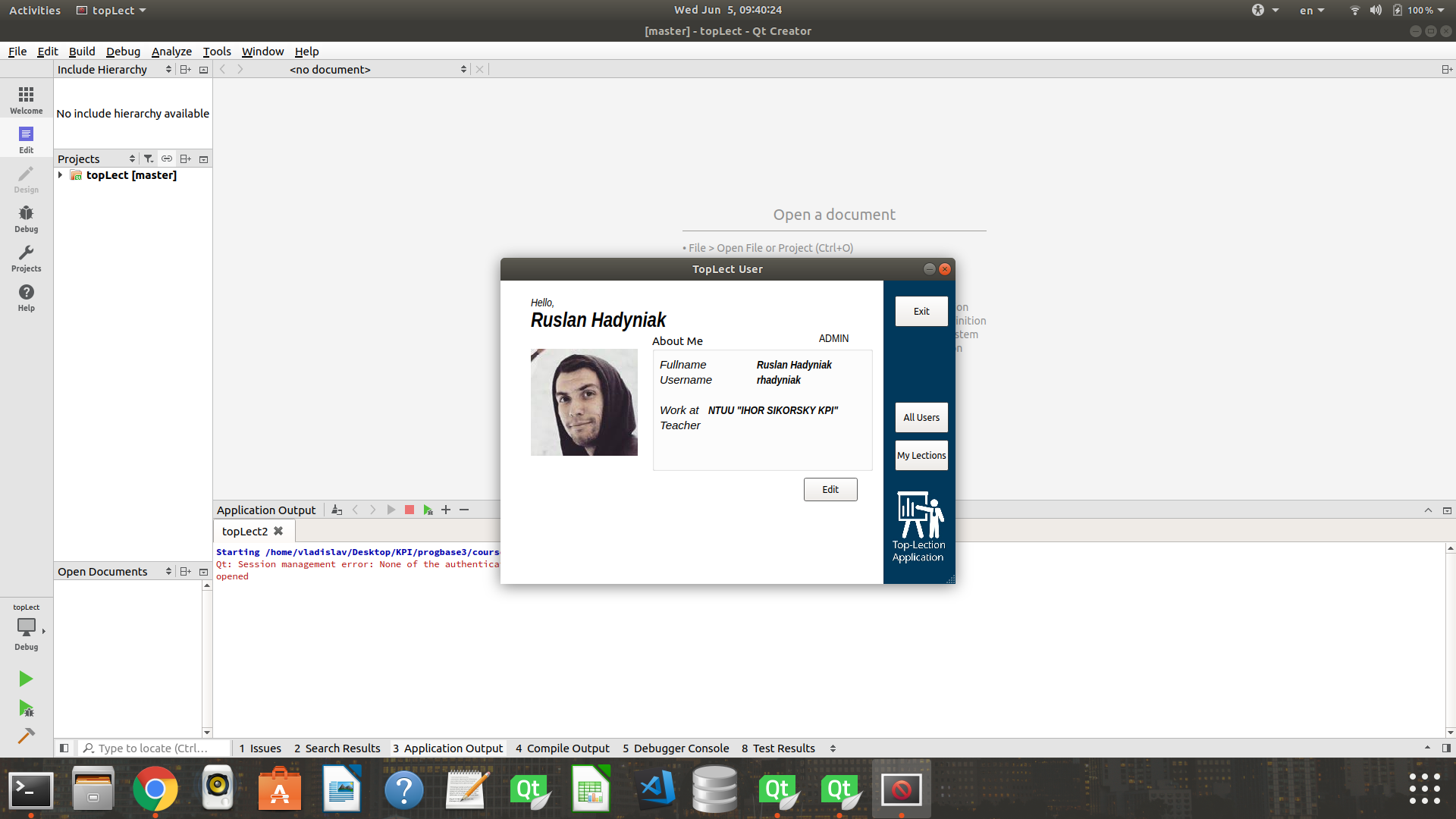
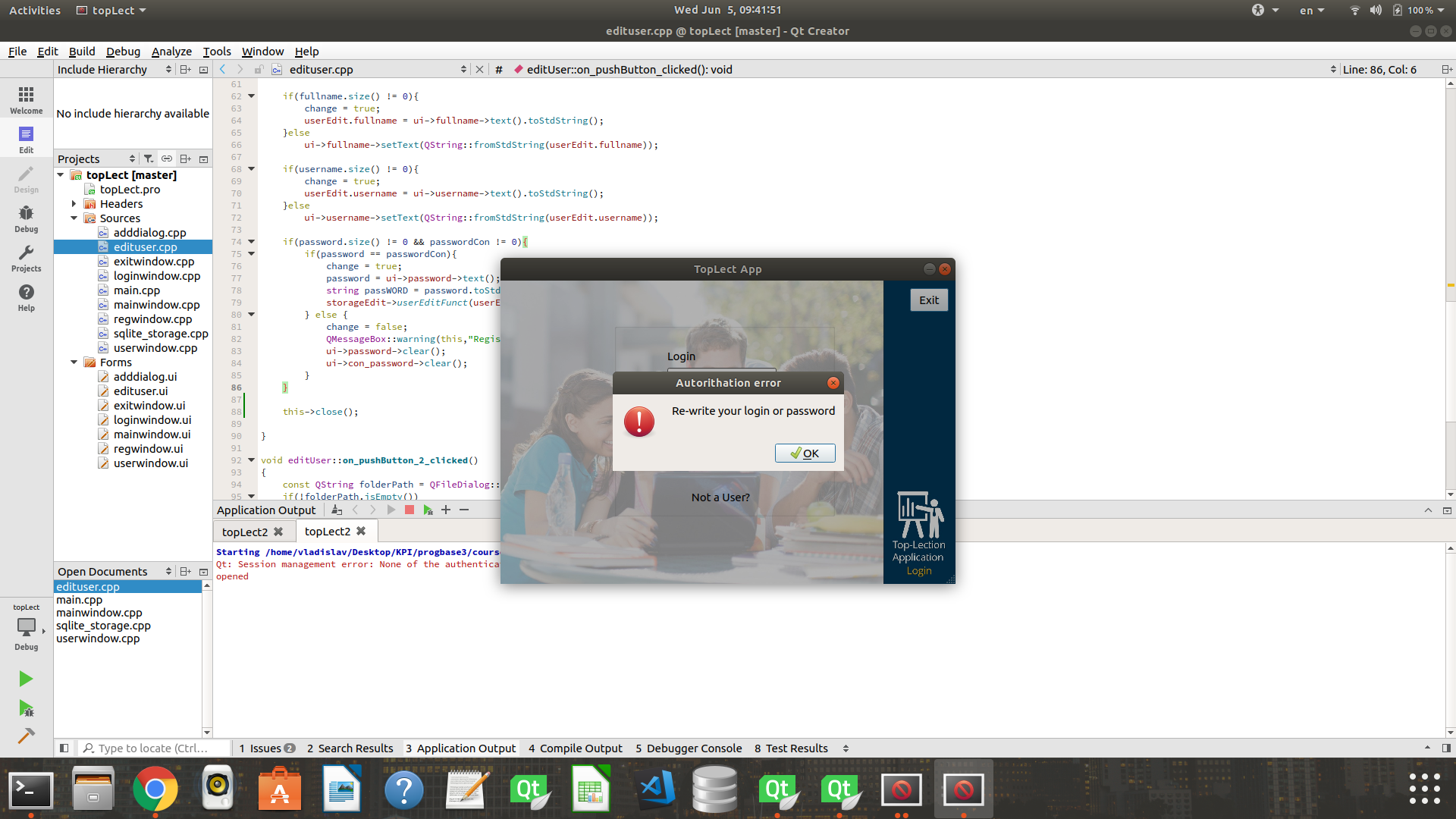
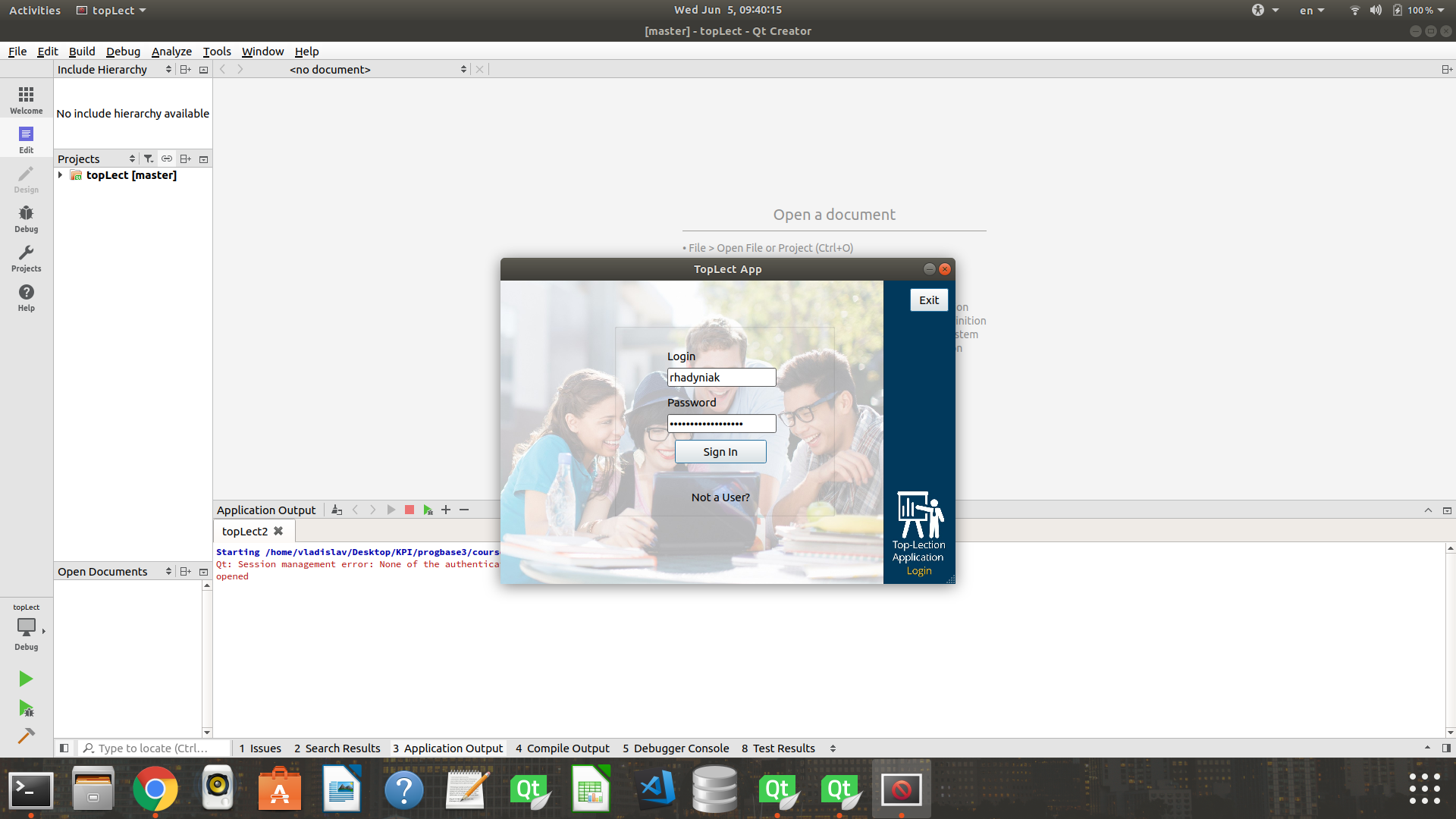
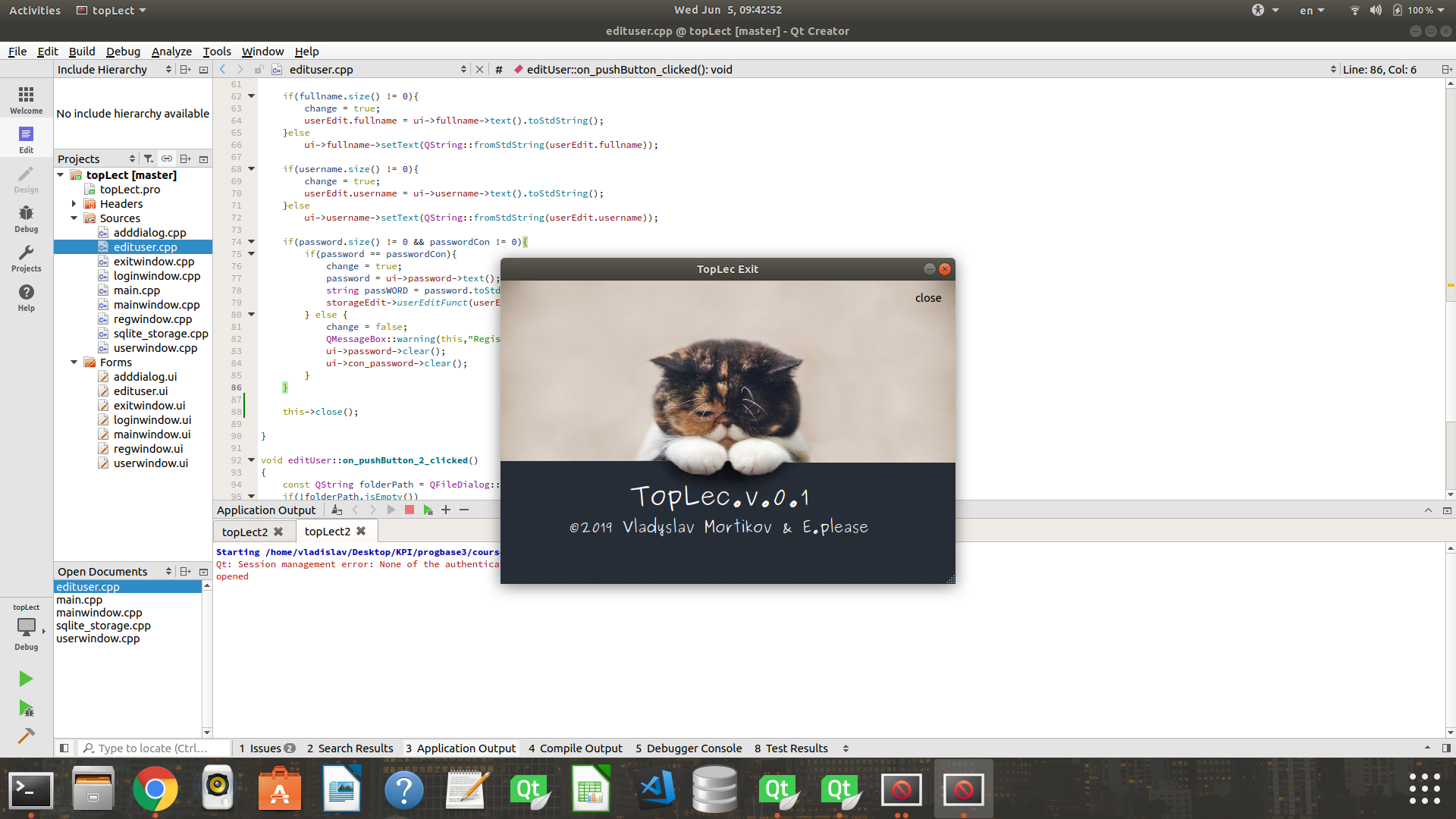
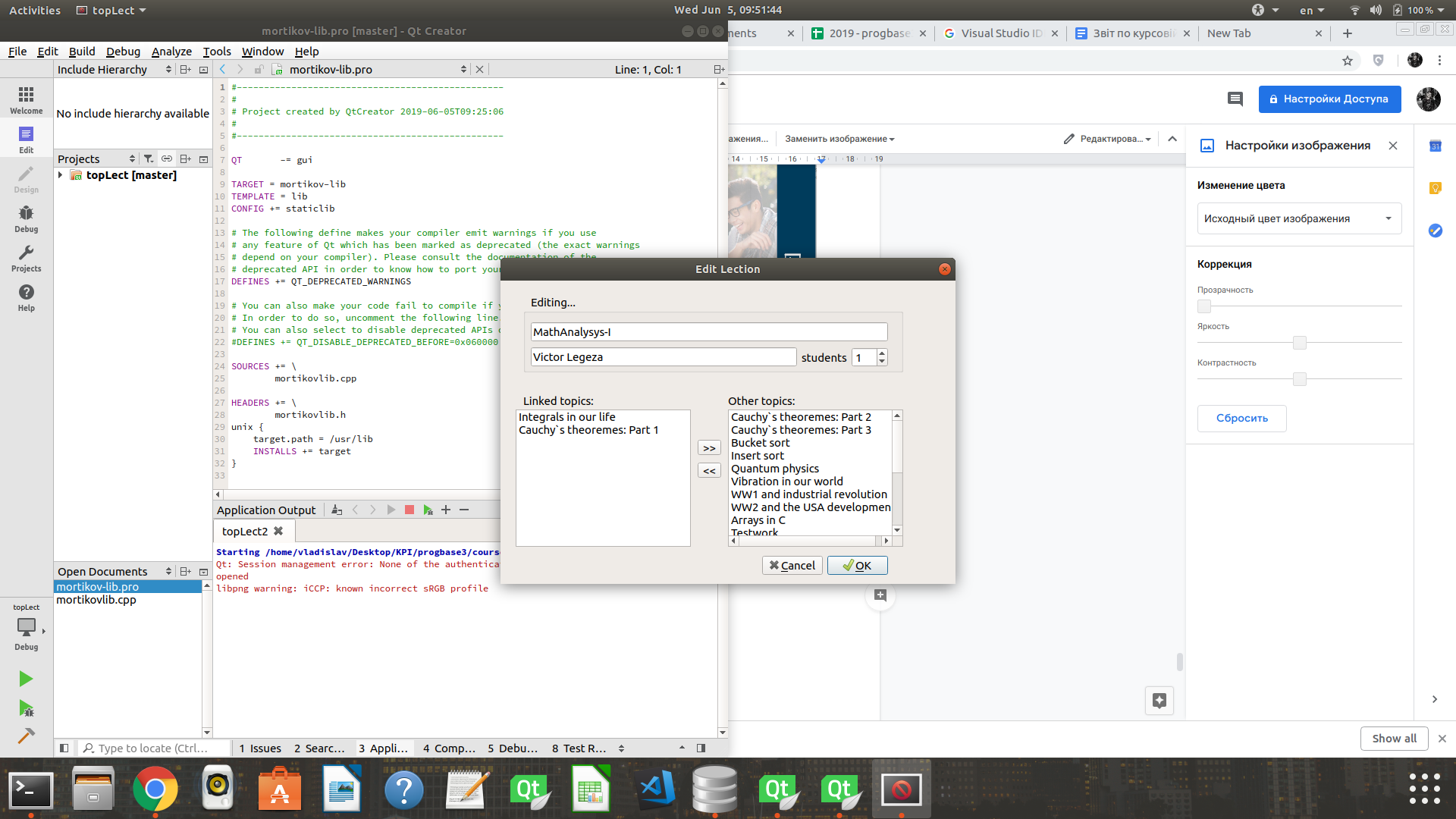
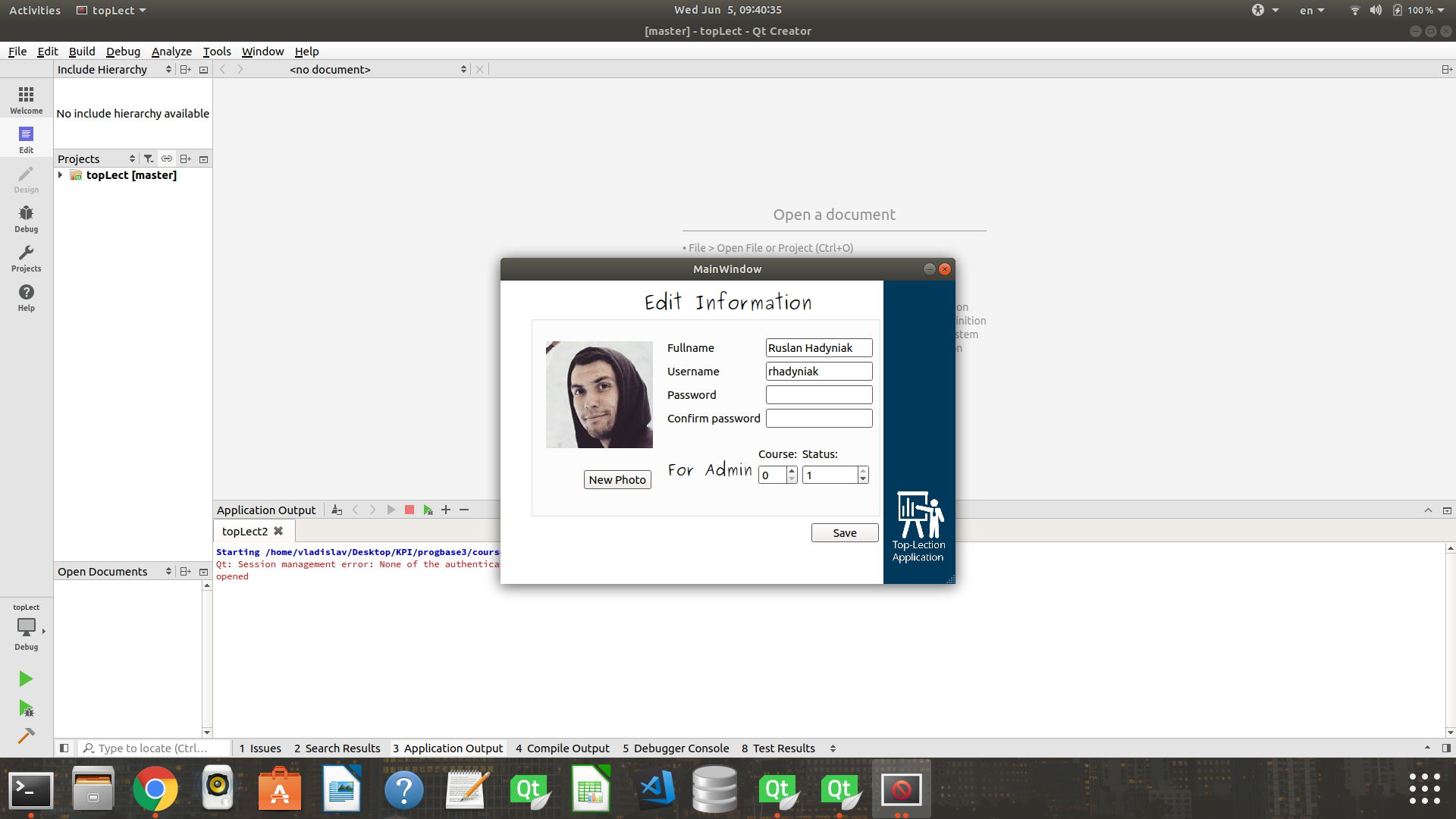
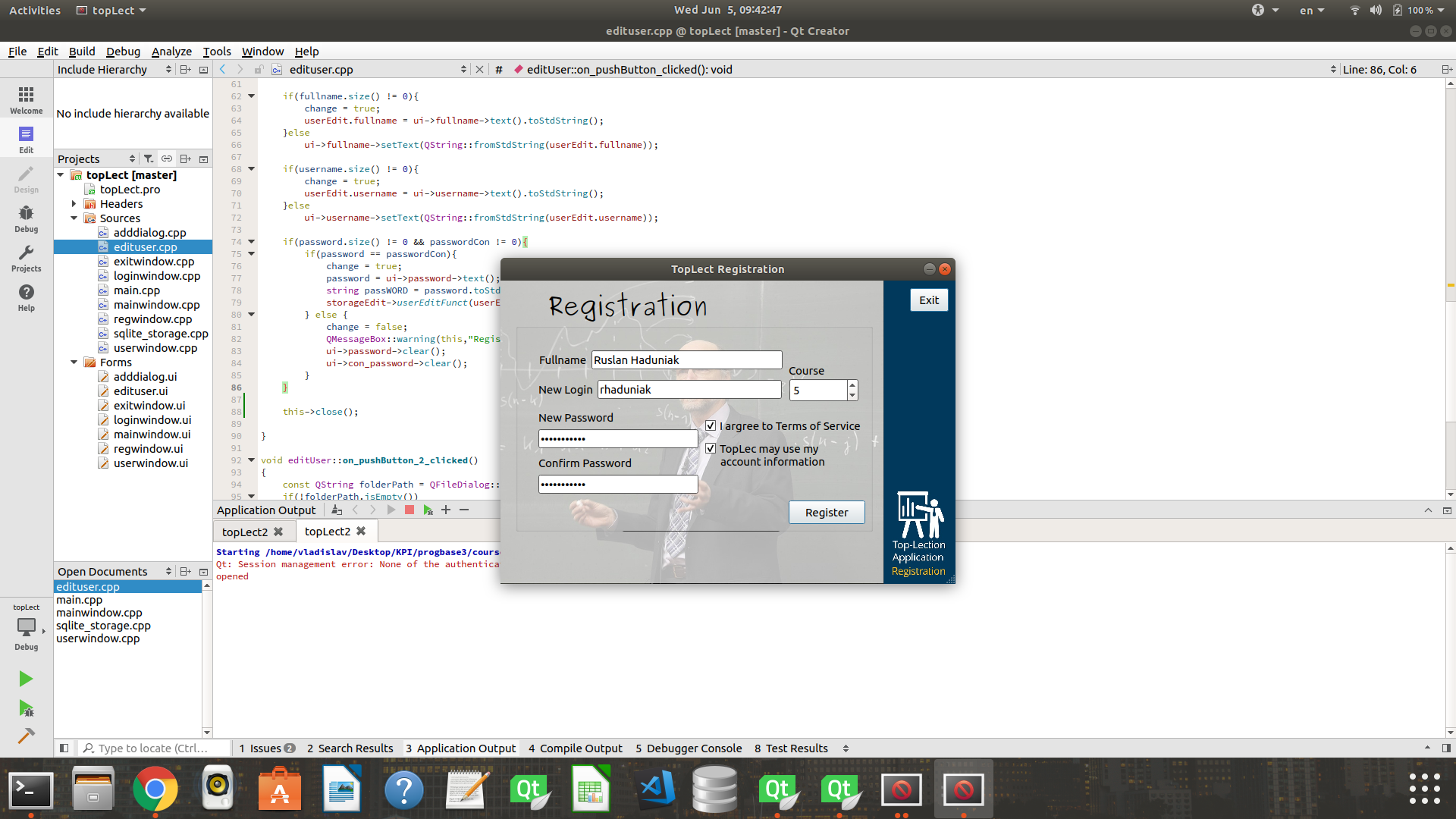
Проект розбитий на такі модулі:



## 4.2. Дизайн та вміст вікон

Для відображення даних система має графічний інтерфейс користувача, що складається з:

1. **Модуль LoginWindow**  – відображає вхідне вікно авторизації користувача. Дає можливість змінити сторінку на: сторінку виходу, регістрації, користувача.
2. **Модуль RegWindow** – відображає форму для реєстрації користувача та Дає можливість змінити сторінку на сторінку виходу або логінації.
3. **Модуль UserWindow** – головна сторінка програми на якій користувач може обробляти дані про себе і перейти в усі основні вікна програми.
4. **Модуль MainWindow** - дозволяє виконувати задачі над сутностями програми.
5. **Модуль AddDialog -** додає елементи в сутності.



Висновки

Метою даної курсової роботи було розроблення програмного додатку для керування програмами лекцій та їхніх тем у межах університету.

Розроблена автоматизована система:

* забезпечує авторизований доступ до системи;
* дозволяє управляти курсами лекцій та їхніми темами;
* має зрозумілий та зручний інтерфейс користувача.

Особливу увагу під час розроблення даного програмного продукту було приділено інтерфейсу користувача і зручності роботи із системою.

Розробка виконана у неповному обсязі, не всі вимоги враховані.

Використання розробленої системи не дозволить забезпечити надійну роботу всіх компонентів.