МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Національний університет «Запорізька політехніка»**

Кафедра програмних засобів

(найменування кафедри)

**КУРСОВИЙ ПРОЕКТ**

з дисципліни «Об’єктно-орієнтоване програмування»

(назва дисципліни)

на тему: «Чат-бот для підтримки міжнародної академічної мобільності»

Студентів 2 курсу групи КНТ-229

спеціальності Комп'ютерні науки

Сепетий О. В.

(прізвище та ініціали)

Колбаса Д. О.

(прізвище та ініціали)

Ващенко М. І.

(прізвище та ініціали)

Момот В. І.

(прізвище та ініціали)

Керівник к.т.н., проф. Табунщик Г.В.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала

Кількість балів: \_\_\_\_\_\_\_ Оцінка: ECTS

Члени комісії Табунщик Г. В.

(підпис) (прізвище та ініціали)

Миронова Н. О.

(підпис) (прізвище та ініціали)

Каплієнко Т. І.

(підпис) (прізвище та ініціали)

м. Запоріжжя

2020 рік

Зміст

перелік умовних позначень, символів, одиниць, скорочень і термінів

ООП - об’єктно-орієнтоване програмування.

Е.Г.І. - елементи графічного інтерфейсу

АМ – академічна мобільність

ЗВО — заклад вищої освіти

ПЗ — програмне забезпечення

STL – Standart Template Library

MinGW – Minimal GNU for Windows

GCC – GNU Compilers Collection

**ЗАВДАННЯ  
на курсовий проект студентів**

Сепетого О.В., Колбаси Д. О., Ващенко М. І., Момота В. І.

**1. Тема проекту:** Чат-бот для підтримки міжнародної академічної мобільності

**2. Термін здачі студентами закінченого проекту:** 14 грудня 2020

**3. Загальні вимоги до курсового проекту:** Розроблений програмний продукт повинний бути новим, розробленим з чистого листа. Для візуалізації поставленого завдання використовувати графічний інтерфейс із використанням інструментальних засобів відповідного середовища розробки. Роботу слід виконати в середовищі розробки Qt Creator, Visual Studio, IntelliJ IDEA. Мови програмування ‒ C++, C#, Java. Для групової роботи використовується GitHub

**Вимоги до інтерфейсу**

* + Необхідно розробити мінімум п'ять форм інтерфейсу, дизайн інтерфейсу.
  + Обов’язково необхідно реалізувати допомогу щодо використання програми, інформацію про розробників.

**Вимоги до класів**

* + Ієрархія класів повинна включати мінімум дванадцять класів, з яких один ‒ абстрактний, три ‒ розроблені класи користувача. Опис класів необхідно оформити у вигляді окремих модулів.

**Вимоги до зберігання даних**

* + Інформація зберігається в звичайних файлах. Інформація завантажується в пам'ять програми при старті, редагується в процесі роботи і зберігається в файли при завершенні. Формат файлу вибирається самостійно.
  + При використанні CSV, JSON та XML будуть додаткові питання на захисті з даних форматів представлення даних.
  + Необхідно взаємодіяти з CSV, JSON або XML (2 з трьох на вибір студентів).
  + За бажанням можна також використовувати базу даних. При використанні СКБД будуть додаткові питання на захисті по завантаженню даних таблиць з бази даних в структуру даних.

**Вимоги до алгоритмів**

* + Необхідно розробити авторських 2-4 алгоритми, наприклад, пошуку або сортування, та порівняти їх за швидкістю виконання зі стандартними реалізаціями.

**Вимоги до оброблення виняткових ситуацій**

* + Для оброблення виняткових ситуацій необхідно розробити авторський клас.
  + Програма повинна обробляти не менш ніж 10 різних помилкових ситуацій.

**4. Зміст розрахунково-пояснювальної записки:**

1. Аналіз предметної області;

2. Аналіз програмних засобів;

3. Основні рішення з реалізації компонентів системи;

4. Керівництво програміста;

5. Керівництво корисутвача;

Висновки

Додаток А. Код програми

Додаток Б. Інтерфейс програми

Додаток В. Слайди презентації

**5. Дата видачі завдання:** 11 вересня 2020 року

Вступ

Під час навм легко загубитися або важко знайти потрібний розділ. Також за нашим баченням на сайті не були розкриті деякі ключові питання типу ціни, мови та іншого.

Вирішенням даної проблеми є створення спеціалізованого чат-бота для підтримки міжнародної академічної мобільності. Найголовніша мета нашого чат-бота - донести цікаву, важливу та легку в сприйняттіінформацію до користувачачання в університеті, у студентів є можливість спробувати навчання у закордонних університетах. На жаль про це мало говорять викладачі та куратор. Деяку інформацію можна знайти на офіційному сайті університету, але це не так зручно і зрозуміло, адже наш сайт має дуже багато розділів і часом та та максимально залучити до участі в АМ.

Чат-боти – це універсальні помічники, які можуть бути використані у різноманітних сферах людського життя, в тому числі і наданні інформації про академічну мобільність.

Мета даного курсового проекту є – навчитись використовувати засоби об'єктно-орієнтованого програмування для створення чат-ботів.

Проведено аналіз предметної області, досліджено аналогічне програмне забезпечення, а також існуючі методи та програмні засоби для вирішення завдання.

Для реалізації програмного продукту використовувалася мова програмування С++ та середа розробки Qt Creator з її власними зручними класами.

Здійснено опис прийнятих рішень, реалізованих класів, наведено опис полів та методів реалізованих класів.

1 Аналіз предметної області

В даному розділі розглянуті існуючі методи вирішення завдання підтримки АМ та програми, що реалізують дану потребу, а саме чат-боти що спрямовані на спрощення взаємодії між студентами та університетом, серед можливостей яких, зокрема є підтримка АМ .

1.1 Аналіз академічної мобільності, як процесу що потребує автоматизації

Міжнародна академічна мобільність являє собою короткострокові програми обміну: кредитна мобільність на навчання (3-12 міс.) та на практику (2-12 міс.) для українських студентів: (молодших) бакалаврів, магістрів, аспірантів, докторантів.

Академічна мобільність для викладання або підвищення кваліфікації або стажування (від 5 днів до 2 міс.) для викладачів та працівників закладів вищої освіти.

Організація цієї події проходить між одиним ЗВО з країни-члена Програми та одиним ЗВО країни-партнера Програми Еразмус+.

Заявником - координатором проєктів з мобільності виступають заклад вищої освіти країн-членів програми Еразмус+.  
 Координують напрям діяльності Національні Агентства Еразмус+ країн-членів Програми (децентралізований напрям).

Брати участь у АМ можуть: студенти (молодші бакалаври, бакалаври, магістри, аспіранти, докторанти), викладачі і административний персонал.  
 Щоб взяти участь, необхідно пройти співбесіду з представником університету, в якому особа навчається, або працює і з представником партнерського университету. Але першими і головними кроками є розуміння що таке АМ і реєстрація для проходження співбесіди. Для полегшення цих кроків і створений наш чат-бот з підтримки АМ.

1.2 Огляд існуючих методів вирішення завдання та аналогів програмного забезпечення

***1.2.1 Передмова***

Звісно одним з найочевидніших шляхів донесення студентів (викладачів та інших працівників закладів вищої освіти) є загальна презентація про процес АМ та процес взяття в ньому участі. Але недоліком цього методу витрати часу на підготовку цього заходу і на всласне презентування.

Іншим методом є застосування чат-бота, що доступний цілодобово і має відповіді на всі запитання загального характеру, який швидко і зрозуміло донесе до людини потрібну їй інформацію, а також надасть можливість реєстрації на співбесіду і сформує перелік охочих.

Перед розглядом аналогів, треба зазначити, що програмного забезпечення, яке б задовольняло назві “Програмне забезпечення для підтримки академічної мобільності” або “Чат-бот для підтримки акадеічної мобільності” не знайшлося, що свідчить про те, що конкретних аналогів немає або їх дуже мало або про те що ПЗ з таким функціоналом нині не є дуже затребуваним. Підтримка АМ є частиною декількох знайдених чат-ботів. Але слід зазначити, що ПЗ, яке немає цього функціоналу аналізувалося на предмет загальної організації побудови взаємодії з користувачем.

Програмами для підтримки академічної мобільності є:

* Програма “MoveOn”;
* “Chatbot TARS”;

Чат-бот, що не маює підтримки АМ, але є цікавими прикладами ПЗ для університетів:

* “Virtual spirits chat bot”;

Критерії аналізу аналогів ПЗ:

* точне донесення до користувача інформації щодо АМ, або університет (якщо чат бот не підтримує АМ)
* побудова діалогу з корстувачем;
* наявність безкоштовної ліцензії;
* зручність і простота інтерфейсу.

1.2.2 Програма “MoveOn”

Компанія-видавець: QS UniSolution

Посилання: https://www.qs-unisolution.com/moveon/

Короткий опис: MoveON - це платформа на основі аналітичних даних для міжнародних офісів вищих навчальних закладів, для покращення процесів їхньої мобільності та ефективніше відстеження міжнародних партнерських відносин. Вона надає можливість створити свій шлях обміну студентами та побудувати стратегічне партнерство для досягнення цілей інтернаціоналізації закладу.

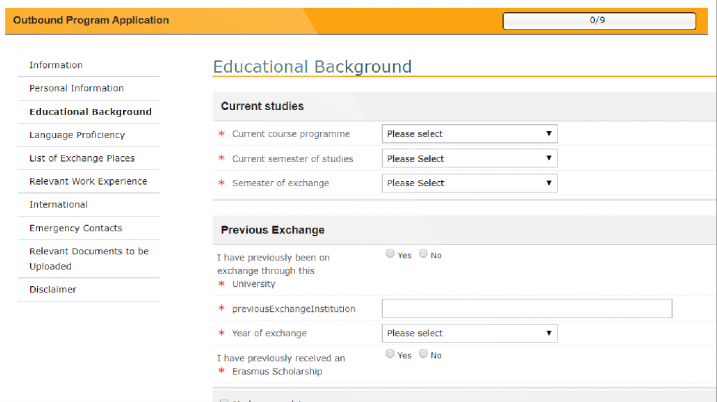
Переваги:

* Розвернення програми проходить швидко і легко
* Має досить постий і зрозумілий інтерфейс
* Дозволяє легко керувати програмами обміну студентів
* Співпрацює з Erasmus

Недоліки

* ПЗ спрямовано на полегшення роботи адміністрації університету, але ні як не на комунікацію зі студентами.
* Має платну ліцензію

Робота програми продемонстрована на рисунку 1.1.

 Рисунок 1.1 – Робота програми «MoveOn»

**1.2.3 Chatbot TARS**

Компанія-видавець: Tars.

Посилання: https://hellotars.com/chatbot-templates/education/SJmmXq/university-search-chatbot-for-studying-abroad

Чат-бот допомагає навчальним закладам автоматизувати процес планування зустрічей, щоб заощадити час та зробити більш привабливими форми реєстрації на курси

Переваги:

* Чат-бот підтримує АМ
* Діалог з користувачем проходить доволі зручно і зрозуміло
* Головним способом комунікації з програмою є панель кнопок, що робіть комунікацією простою
* має безкоштону пробну версію

Недоліки:

* Не надає інформацію про університети.

Робота програми продемонстрована на рисунку 1.2.

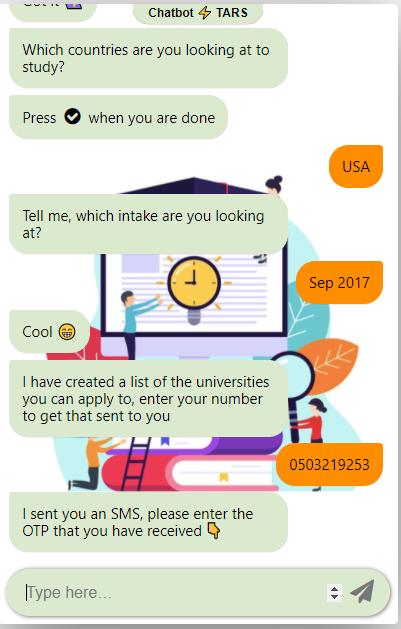


Рисунок 1.2 – Робота з програмою «Chatbot TARS»

1.2.4 «Virtual spirits chat bot»

Компанія-видавець: Virtual spirits.

Посилання: https://www.virtualspirits.com/chatbot-for-university.aspx

Короткий опис: Чат-бот використовується для підтримати студентів та надання миттєвої відповіді на послуги кампусу, реєстрацію, стипендії, навчальні послуги, поради щодо іспитів, ІТ-послуг, усунення несправностей тощо

Переваги:

* Має широкий набір функціоналу.
* Зручний інтерфейс
* Взаємодія виконується за допомоги текстових повідомлень і іноді кнопок (категорій).

Недоліки:

* Нажаль не надає підтримки АМ

Робота програми продемонстрована на рисунку 1.3.

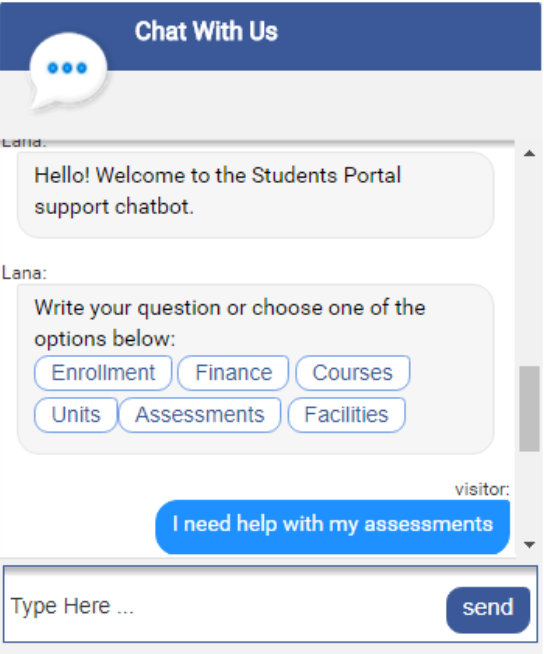


Рисунок 1.3 – Робота з програмою «Virtual spirits chat bot»

2 Аналіз програмних засобів

В даному розділі розглянуті основні особливості програмних засобів, за допомогою який реалізовано курсовий проект.

2.1 Огляд особливостей мови програмування

С++ - мова програмування загального призначення.

С++ підтримує об’єктно-орієнтовану парадигму, що дозволяє представити програму, як набір обєектів, кожен з яких має свої властивості, поведінку і своє призначення в програмі. Програму, написану з використанням ООП легше модифікути і підтримувати. Однією з особливостей саме мови С++ в тому, що вона підтримує множинне спадкування. С++ підтримує програмування з використанням шаблонів, що спрощує роботу з даними. Бібліотека Qt, що дозволяє розроблювати графічний інтерфейс, має свою реалізацію STL, свою реалізацію потоків вводу-виводу, та інших надбань стандартної бібліотеки з більшою кількістю можливостей, що робить процес розробли ПЗ більш швидким та продуктивним.

2.2 Огляд особливостей обраного компілятору

Компілятор MinGW – версія компілятора GCC, створена для розробки ПЗ мовою С++ на операціїних системах сімейства Windows. Розрорблене нами ПЗ не використовує жодних специфічних речей, що можуть робити Unix-подібні операційні системи, чого не може робити Windows, тому необхідність у використанні компілятора Cygwin відпала. Компілятор MSVC постачається разом з Microsoft Visual Studio, але на наш погляд не має істотних переваг. До того ж робота в Visual Studio потребує додаткових операцій з підключенням бібліотеки Qt.

2.3 Огляд класової ієрархії

QString — Клас динамічного типу даних для роботи зі рядками тексту. Використовується для зберігання інформації (рядки тексту), що зчитується з файлів, вводиться користувачем.

QFile – Клас для введення-виведення для зчитування та запису текстових і двійкових файлів і ресурсів. Використовується для роботи з файлами, зчитування та запису текстової інформації, а також для підключення таблиць стилів.

QList – Контейнерний клас Qt на основі двозв’язного списку. Дозволяє виконувати операції вставки та видалення елементу за постійний час. Використовується для зберігання даних про студентів і університети.

QStringList — Нащадок класу QList, але на відміну від нього не є шаблоним класом, тобто це QList з типом заповнювачем QString. Він також дозволяє виконувати операції вставки та видалення елементу за постійний час. Використовується для зберігання рядків тексу використовуються у локалізації.

QDomDocument – Клас, що реалізує собою роботу з XML документом. За своєю суттю є коренем дерева документа. Використовується для зчитування та запису інформації в XML документ.

QDateTime — Клас, що надає функції дати та часу, тобто об’єднує класи QDate і QTime. Використовується для визначення часу відпраки повідомлення.

QRegExp - Клас, що надає порівняння зі зразком за домоги регулярних виразів. Використовуєтться для зберігання регулярного виразу.

QRegExpValidator — Клас, використовуються для перевірки відповідності рядка регулярному виразу.

2.4 Висновки з розділу

У даному розділі було описано мову програмування та її особливості. Також було описано класи, що були використані в процесі розробки ПЗ та коротко для чого саме були вони використані.

3 основні рішення з реалізації компонентів системи

В даному розділі розглянуті основні рішення з розробки класів, основні розроблені алгоритми, рішення щодо розробки інтерфейсу користувача, рішення щодо збереження даних та використання бази даних.

3.1 Основні рішення щодо розроблених класів

Структура програми складається з 18 класу, 15 з яких – користувацькі, 3 – класи форм. Один з користувацьких класів є абстрактним. Діаграми користувацьких класів знаходяться у додатку А.

Класову структуру програми зображено на рисунку 3.1.

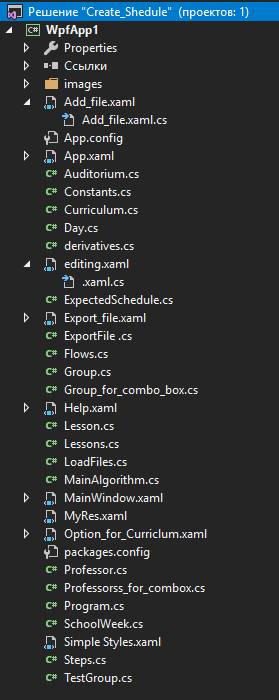


Рисунок 3.1 – Класова структура програми

3.1.1 Клас LoadFiles

Клас LoadFiles містить у собі основний інструментарій для роботи із завантаження файлів Excel, перевірки та форматування цих даних перед занесенням у базу даних (БД), встановлення зв’язку із БД, розриву цього зв’язку та завантаження інформації з таблиць, що є в наявності у БД. Код файлу визначення класу знаходиться в додатку Б. Дані та методи класу наведені у таблиці 3.1.

Таблиця 3.1 – Опис полів та методів класу LoadFiles

|  |  |
| --- | --- |
| Поля та методи класу | Опис |
| **1** | **2** |
| ***public:*** | |
| SQLiteConnection DB | Змінна, за допомогою якої встановлюється зв'язок із БД. |
| public static DataTable loadTable(string nameOfTable, bool flag = true) | Метод, що дозволяє отримати дані певної таблиці з БД та, за бажання, очистити поле Id. |
| public void LoadDB(string dataSource) | Метод, що створює зв'язок із БД. |
| public void CloseDB() | Метод, що розриває зв'язок із БД. |
| public void LoadExcelFileForCurriculum (string filename, string name\_of\_course) | Метод, що дозволяє отримати дані про навчальний план за допомогою Excel-файлу, замінити тип аудиторії та назву кафедри на відповідне Id та перенести їх у БД |

Продовження таблиці 3.1

|  |  |
| --- | --- |
| **1** | **2** |
|  | файлу, замінити тип аудиторії та назву кафедри |
| public void LoadExcelFileForProfessors(string filename) | Отримує дані про викладачів за допомогою Excel-файлу, замінити посаду(ранг) викладача та назву кафедри на відповідне Id та перенести їх у БД |
| public void LoadExcelFileForGroup(string filename) | Отримує дані про потоки та групи, що їм належить, за допомогою Excel-файлу (замінивши назву кафедри на відповідне Id) та перенести їх у БД, розділивши на 2 окремі таблиці. |
| ***protected:*** | |
| protected static void inputCheckForColumnsNames(ref DataTable dataTable) | Превіряє на правильність та відповідність шаблону назви колонок у Excel-файлі, та замінює на стандартну для БД назву. |
| protected static int inputCheckForAuditoriumField(string input\_data) | Превіряє на правильність та відповідність шаблону назви типів аудиторій у Excel-файлі, та замінює на стандартну для БД назву. |
| protected static int inputCheckForDepartmentField(string input\_data) | Превіряє на правильність та відповідність шаблону назви кафедр у Excel-файлі, та замінює на стандартну для БД назву. |

Продовження таблиці 3.1

|  |  |
| --- | --- |
| protected static int inputCheckForProfessorsRankField(string input\_data) | Превіряє на правильність та відповідність шаблону назви посад викладачів у Excel-файлі, та замінює на стандартну для БД назву. |

3.1.2 Клас ExportFiles

….

3.1.3 Клас Flows

….

3.1.4 Клас Auditorium

3.2 Основні розроблені алгоритми

3.2.1 Аналіз головного алгоритму

В ході написання курсової роботи був створений власний алгоритм складання розкладу.

Так і розроблений алгоритм, видає прийнятний розклад, але не гарантує, що він є найкращим серед можливих. Це позначає найголовнішу особливість алгоритму, а саме подальшу оптимізацію та можливе розширення виконуваних завдань.

Вхідними даними для алгоритму є:

* …..

Під час підрахунку остаточного результату алгоритм працює з проміжними даними:

* …..

У кінці виконання алгоритм формує вихідні данні:

* …..

Усі значення мають бути коректними та строго формалізованими (це виконується на етапі створення бази даних).

Увесь алгоритм можна поділити на 2 етапи:

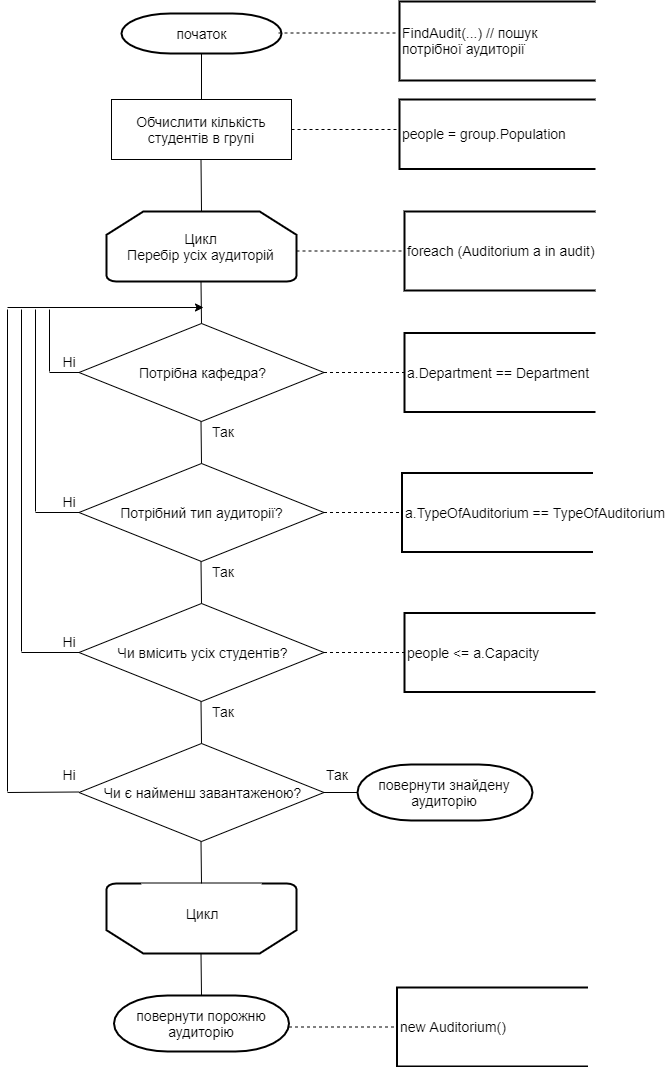
1. Формування списку занять;
2. Пошук часу проведення заняття.

Виконання кожного етапу базується та таких алгоритмах:

* алгоритм лінійного пошуку;
* алгоритм сортування вставками;
* елементи динамічного програмування.

Розглянемо основні типові алгоритми.

Опис розроблених алгоритмів

Рисунок 3.19 – Пошук відповідної аудиторії для заняття

….

Після виконання алгоритму маємо складений розклад для навчальних груп на потоках, аудиторій та викладачів.

3.3 Основні рішення щодо розробки інтерфейсу

3.3.1 Вибір платформи.

Головнним рішенням, щодо розробки інтрерфейсу, було вибрати нову та сучасну платформу для розробки інтерфейсу Qt. Ця платформа відзначаеться кросплатфоменістю та більш зручним інтерфейсом для програміста.

Опис рішень, які використовувались для реалізації інтерфейсу

3.3.2 Основні рішення з впорядкування елементів інтерфейсу

Під час розробки інтерфейсу користувача було використано наступні е.г.і.:

* MainWindow
* VHLayouts
* TextBrowser
* PushButton
* Label
* LineEdit
* MenuBar
* TextEdit
* Frame
* ComboBox

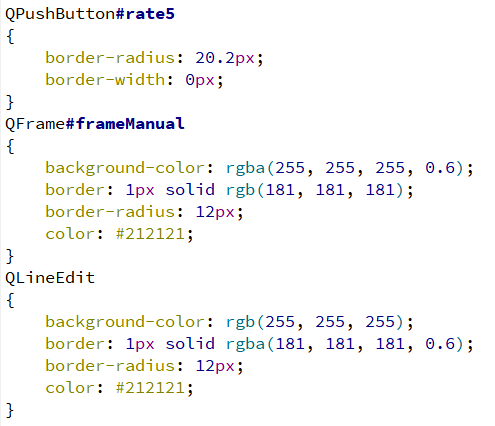
Саме вони були використані тому що, по перше, це одні з найпростіших та найпопулярніших е.г.і. в цілому, по друге, до цих елементів доволі просто застосовувати стилі та інші графічні зміни, по третє, ми вже маємо досвід роботи з цими е.г.і.

3.3.3 Основні рішення з комунікації з користувачем

Для кращої комунікації з користувачем та рівномірного розподілення місця у додатку були використані кнопки(PushButton) для запиту певної стандартної інформації у бота, та строка вводу (TextEdit, LineEdit) для нестандартної інформації такої як: особиста інформація користувача для надання ії керівникам АМ, або повідомлення про помилку\баг у додатку.

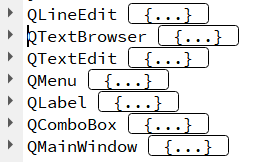
3.3.4 Основні рішення з створенням інтерфейсу

Було активно використано стилі. Тобто відокремлення певної групи елементів і створення для них власного стилю. Приклад данного рішення наведений нижче на рисунку 3.27.1…..

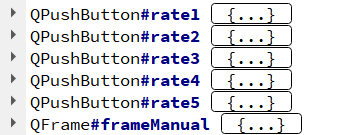


Основним варіантом застосування стилыв є використання Qt Style Sheets. Як стає очевидно з назви, це трохи специфічний аналог звичних всім каскадних таблиць стилів CSS(Cascading Style Sheets), без яких немислимий сучасний інтернет. Оскільки в минулому семестрі у нас була дисципліна «WEB-дизайн» і ми активно використовували CSS в своїх роботах, то нам здалося зручним використовувати даний аналог і тут. Також це зручно в тому плані, що всі графічні рішення виводяться в окремий файл, не займають місце в основному коді і довантажуються під час запуску програми, а також потрібний стиль застосовується при виборі теми програми в Налаштуваннях.

Було використано звернення по класу. Це зручно тим, що зміни застосовуються відразу до всіх елементів даного класу. Приклад:



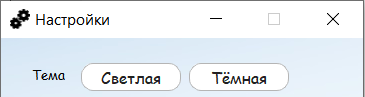
Звернення по імені дає можливість кастомізувати конкретний елемент з усього класу. Приклад:



Псевдо-стан елемента допомагає зробити програму більш динамічною і "живий" для користувача. Приклад:



Також було прийнято рішення додати повністю альтернативну тему для усього додатку, т.з. темний стиль зокрема від світлого. Ця ідея прийшла до нас у процесі створення програми вночі та ми подумали, що користувачеві буде зручно змінювати колірний стиль усього додатку одним кліком у Налаштуваннях.

3.4 Основні рішення щодо розробки бази даних

3.4.1 Основні рішення щодо вибору бази даних

База даних, як один з основних компонентів програми, потребує зваженого вибору середовища та платформи розробки, бо від цього залежатиме якість та комфорт не тільки у користуванні, а й у написанні програмного продукту, отже, саме тому основними критеріями для вибору БД були:

* …..

Отже, проаналізувавши наявні пропозиції у цій сфері, платформа …. вирізняється з-поміж інших варіантів, та відповідає усім заданим критеріям, а саме:…..

3.4.2 Структура збереження даних у БД

У процесі збереження даних використовуються таблиці, як ключові носії інформації, та представлені наступними екземплярами:

«*Auditorium*» - таблиця для збереження даних аудиторій, а саме: номер, літера, кількість місць, тип аудиторії та кафедра, якій вона належить (рис. 3.28).

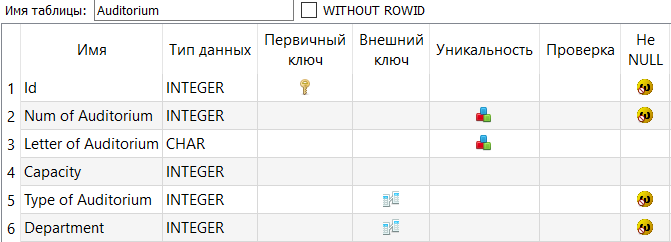


Рисунок 3.28 – Структура таблиці «Auditorium»

Первинним ключем є поле Id, яке ідентифікує кожен запис та допомагає зекономити пам'ять у інших таблицях, адже замість сполучення номеру, літери та кафедри використовується саме Id. Поля «Types of Auditorium» та «Department» пов’язані зовнішнім ключем з допоміжними таблицями для економії пам’яті та збереження цілісності даних, бо зовнішній ключ виступає перешкодою для внесення неіснуючих чи нерелевантних даних.

….

3.4.2 Основні запити SQL, що було використано

Мова для створення запитів SQlite є SQL-подібна, відрізняючись лише синтаксисом (несуттєво) та функціоналом. В ході роботи з базою даних були використані наступні запити для взаємодії:

* ….

4 керівництво програміста

В даному розділі розглянуті призначення, умови застосування, характеристика програми, звертання до програми, початкові та вихідні дані та представлені повідомлення.

4.1 Призначення та умови застосування програми

4.1.1 Призначення програми

Програма “Створення розкладу” призначена для автоматичного стоврeння розкладу на основі даних користувача.

4.1.2 Функції програми

Функції, що виконує програма:

* …..

4.1.3 Умови застосування програми

До комп’ютеру, на якому виконується програма, висуваються наступні вимоги:

* операційна система Windows 7, 8, 10;
* наявність дисплею;
* наявність миші та клавіатури;
* до 300 МБ оперативної пам’яті за умови використання функції створити розклад.

4.2 Характеристика програми

Програма виконана за допомогою мови програмування високого рівня С# в середовищі розробки Visual Studio 2017 , 2019 за допомогою платформи WPF.

4.2.1 Структура програми

Програма складається з 18 класу, 15 з яких – користувацькі, 3 – класи форм. Один з похідних класів інтерфейсу, яких ще 5, є один абстрактний клас. Проект містить наступні класи та їх реалізацію, файли ресурсів, базу даних SQL. Ієрархія представлена на рисунку 4.1.

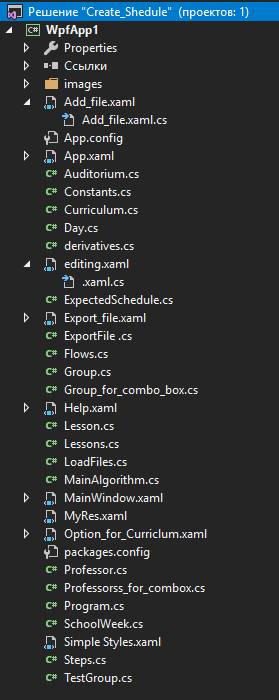


Рисунок 4.1 – Ієрархія класів проекту

4.2.2 Алгоритм роботи програми

….

4.3 Звертання до програми

Для запуску програми потрібно попередньо упевнитися в відповідності характеристик комп’ютера, на який встановлена програма, до системних вимог програми. Потрібно упевнитися у наявності всіх файлів програми.

Звертання до програми передбачене одним способом: через виконуваний файл Create\_Shedule.exe. Звертання до програми через командний рядок не передбачене, адже програма розроблена саме для роботи через інтерфейс для складання розкладу.

4.4 Вхідні та вихідні дані

4.4.1 Вхідні дані

Вхідні дані уявляють собою файли Екселю, з розширення .xlxs та .xlx. Вони повинні буди чотирьох різних типів. У кожному з яких за шаблоном повинні міститися дані в залежності від таблиці.

4.4.2 Вихідні дані

Файли Екселю з розширення .xlxs, .xlx. Вони повинні буди чотирьох різних типів. У кожній з яких за шаблоном повинні міститися дані в залежності від таблиці. Та є можливість зберегти файл з готовим розкладом.

4.5 Повідомлення

* ….

5 керівництво користувача

В даному розділі розглянуто призначення програми, умови її виконання, процес виконання програми та повідомлення для користувача.

5.1 Призначення програми

Програма “Створення розкладу” призначена для автоматичного стоврeння розкладу на основі даних користувача.

5.2 Умови виконання програми

5.2.1 Апаратні вимоги програми

До комп’ютеру, на якому виконується програма, висуваються наступні вимоги:

* наявність дисплею;
* наявність клавіатури та миші;
* до 300 МБ оперативної пам’яті за умови використання функції створення розкладу.

5.2.2 Вимоги до користувача

До користувачів висуваються наступні вимоги:

* …..

5.3 Виконання програми

5.3.1 Запуск програми

Для запуску програми потрібно попередньо упевнитися в відповідності характеристик комп’ютера, на який встановлена програма, до системних вимог програми. Другим кроком потрібно упевнитися в наявності всіх бібліотек для роботи програми та в наявності бази даних для повноцінного функціонування.

Звертання до програми передбачене одним способом: через виконуваний файл. Звертання до програми через командний рядок не передбачене, адже програма розроблена саме для візуалізації роботи алгоритмів пошуку найкоротшого шляху.

Після запуску виконуваного файлу Create\_Shedule.exe з’являється головна форма програми, що свідчить про початок роботи з програмою. Головна форма програми зображена на рисунку 5.1.

Рисунок 5.1 – Головна форма програми

5.3.2 Виконання роботи з програмою

Спершу користувач має ….

5.4 Повідомлення користувачу

….

5.5 Довідка програми

Для зручності користувача та для отримання допомоги українською мовою в будь-який момент часу було створено довідку для програми. Форма довідки зображена на рисунку 5.3.

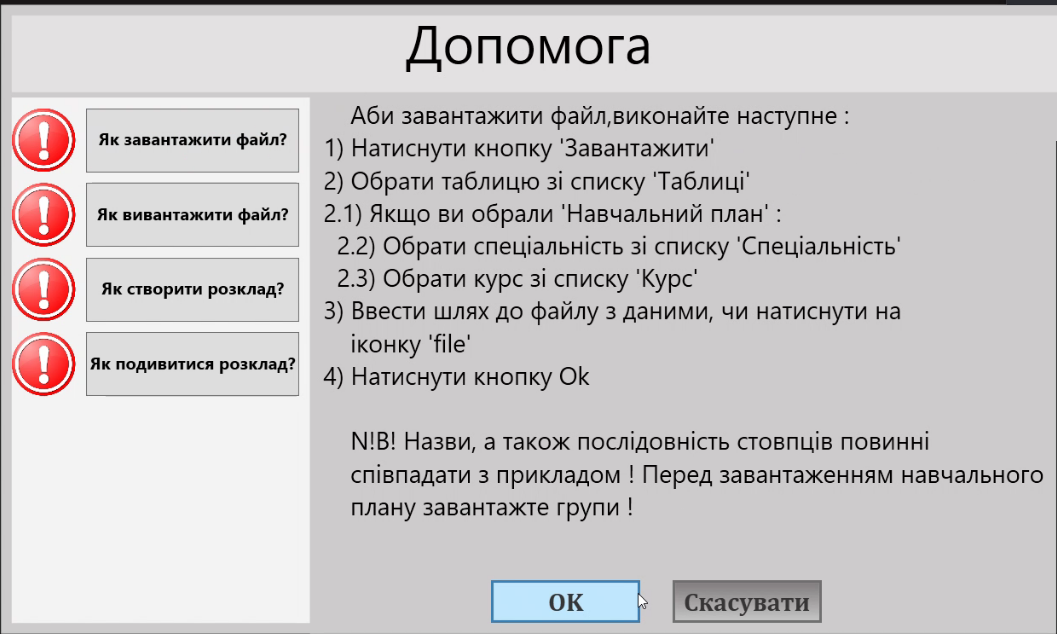


Рисунок 5.3 – Форма довідки

Висновки

ПЕРЕЛІК ПОСИЛАНЬ

Додаток А Діаграма класів

Додаток Б Код Програми