

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
Факультет комп'ютерних наук і технологій

(повне найменування інституту, факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ РОБОТИ
ЕЛЕКТРОННОГО ЩОДЕННИКА ЗДОБУВАЧА ОСВІТИ
STUDENT EDIARY SOFTWARE

Виконав: студент(ка) 4 курсу, групи КНТ-119сп
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

Бражніков І.А.

(прізвище та ініціали)

Керівник Гофман Є.О.

(прізвище та ініціали)

Рецензент Коцур М.І.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Інститут, факультет ПРЕ, ФКНТ

Кафедра програмних засобів

Ступінь вищої освіти бакалавр

Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)

Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.

С.О. Субботін

“ ” 2022 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Бражнікова Іллі Андрійовича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення для роботи електронного щоденника здобувача освіти. Student eDiary Software

керівник проєкту (роботи) Гофман Євгеній Олександрович, к.т.н.,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “ 27 ” квітня 2022 року № 102

2. Строк подання студентом проєкту (роботи) 7 червня 2022 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	Гофман Є.О., доцент		
Нормоконтроль	Белова А.В., асистент		

7. Дата видачі завдання “ 27 ” квітня 2022 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки	2 тиждень	Розділ 2
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділи 3,4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис) Бражніков І.А.
(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис) Гофман Є.О.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
107 с., 32 рис., 13 табл., 4 дод., 17 джерел.

C#, CACHE, МОВА ПРОГРАМУВАННЯ, ПРОГРАМНЕ
ЗАБЕЗПЕЧЕННЯ, УСПІШНІСТЬ, УЧЕНЬ, ШКОЛА

Об'єкт дослідження – процес розробки програмного забезпечення для освітньої діяльності.

Предмет дослідження – програмні засоби для надання інформації про результати оцінювання здобувачів освіти.

Метою роботи є розробка програмного забезпечення для надання інформації про успішність здобувачів освіти у вигляді електронного щоденника.

Матеріали, методи та технічні засоби: мова програмування C#, середовище розробки Microsoft Visual Studio, система керування базами даних InterSystems Cache.

Результати. Розроблено програмне забезпечення у вигляді електронного щоденника для редагування, збереження та перегляду успішності учня, розкладу та домашнього завдання.

Практична цінність роботи полягає у розробці програмного забезпечення для надання інформації про успішність здобувачів освіти у вигляді електронного щоденника.

Висновки. Розроблено програмне забезпечення для відображення успішності учнів, контролю розкладу уроків, оцінок та домашнього завдання учнів.

Галузь використання – заклади освіти.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 107 pages, 32 figures, 13 tables, 4 appendixes, 17 sources.

C#, CACHE, PROGRAMMING LANGUAGE, SOFTWARE, SUCCESS, STUDENT, SCHOOL

The object of research is the process of software development for educational activities.

The subject of the research is software to provide information about the results of assessment of students.

The purpose of this work is to develop software to provide information about the success of students in the form of an electronic diary.

Materials, methods and technical tools: C# programming language, Microsoft Visual Studio development environment, InterSystems Cache database management system.

Results. The software tools in the form of an electronic diary for editing, saving and viewing student performance, schedule and homework have been developed.

The practical value of the work lies in the development of software to provide information on the success of students in the form of an electronic diary.

Conclusions. The software for displaying student performance, control lesson schedules, grades, and student homework is developed.

Scope of use – educational institutions.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ	9
1 Аналіз предметної області	11
1.1 Поняття про програмне забезпечення для підтримки освітньої діяльності.....	11
1.2 Дослідження програмних засобів для підтримки освітнього процесу ...	20
1.3 Висновки за розділом 1	31
2 Матеріали і методи	33
2.1 Вибір мови програмування та інших технологій розробки	33
2.2 Вибір середовища розробки для створення програмного забезпечення	35
2.3 Вибір системи керування базою даних.....	36
2.4 Висновки за розділом 2	38
3 Опис програми	39
3.1 Схема взаємодії основних модулів програмного забезпечення електронного щоденника здобувача освіти.....	39
3.2 Опис основних розроблених структур даних.....	40
3.3 Основні рішення з розробки бази даних.....	47
3.4 Висновки за розділом 3	47
4 Експлуатація, тестування та експериментальне дослідження програми.....	49
4.1 Призначення й умови застосування програми	49
4.2 Характеристики програми	50
4.3 Інструкція по експлуатації програми.....	51
4.3.1 Звернення до програми.....	51
4.3.2 Вхідні й вихідні дані.....	51
4.3.3 Повідомлення.....	52
4.4 Виконання програмного забезпечення для роботи електронного щоденника здобувача освіти	53
4.5 Висновки за розділом 4	58

Висновки.....	59
Перелік джерел посилання	62
Додаток А Технічне завдання.....	64
Додаток Б Опис програми	68
Додаток В Текст програми	73
Додаток Д Слайди презентації	100

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API – прикладний програмний інтерфейс;

БД – база даних;

БД – база даних;

ГПІ – графічний програмний інтерфейс;

ПЗ – програмне забезпечення;

СКБД – система керування базами даних;

ТЗ – технічне завдання.

ВСТУП

Проблема розроблення програмних засобів для підтримки освітньої діяльності в дистанційному форматі є досить актуальною у наш час, що обумовлено епідемією коронавірусу, а також нестабільною ситуацією на кордонах з країною-агресором [1]-[5].

У наш відомих багато програмного забезпечення для підтримки освітньої діяльності. До такого програмного забезпечення відносяться програми для управління класом, інформаційні системи студентів, мовне програмне забезпечення, довідкове програмне забезпечення тощо. Відомо, що школи повільно впроваджують нові технології. Але оскільки цифрова трансформація вплинула на роботу кожної галузі, вони також не залишилися позаду. Освітнє програмне забезпечення робить навчання більш ефективним, особливо в умовах необхідності навчання у дистанційному форматі.

Важливим програмним забезпеченням для підтримки освітньої діяльності є електронні журнали та електронні щоденники школярів, що дозволяють зберігати та відстежувати інформацію про учнів та студентів, таку як записи відвідуваності, оцінки тощо.

У дипломній кваліфікаційній роботі бакалавра розв'язується актуальне завдання розробки програмного забезпечення для надання інформації про успішність здобувачів освіти у вигляді електронного щоденника.

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для підтримки освітньої діяльності;
- здійснити проектування програмного забезпечення для надання інформації про успішність здобувачів освіти у вигляді електронного щоденника;
- створити програмне забезпечення для відображення успішності учнів, контролю розкладу уроків, оцінок та домашнього завдання учнів;

– виконати тестування розробленого програмного забезпечення для надання інформації про успішність здобувачів освіти у вигляді електронного щоденника.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття про програмне забезпечення для підтримки освітньої діяльності

Програмне забезпечення для підтримки освітньої діяльності (навчальне програмне забезпечення) – це термін, який використовується для будь-якого комп'ютерного програмного забезпечення, створеного з освітньою метою. Він охоплює різні діапазони від програмного забезпечення від вивчення мови до програмного забезпечення для керування класом і довідкового програмного забезпечення. Мета цього програмного забезпечення – зробити певну частину освіти більш ефективною [1]–[5].

Освітнє програмне забезпечення – це масовий, всеохоплюючий термін, який використовується для позначення будь-якого програмного забезпечення, призначеного для використання в освітній галузі. Термін включає все, від інформаційних систем студентів і програмного забезпечення для керування класом до програмного забезпечення для керування довідками та програмного забезпечення для вивчення мови [1]–[5].

Проте це програмне забезпечення використовується з метою зробити деякі аспекти освітньої індустрії більш ефективними. Школи мають репутацію одного з останніх місць, де впроваджуються технології, але оскільки цифрова трансформація призвела до реконструкції та перебудови галузей у кожному секторі, вони також почали сприймати впровадження програмних засобів як неминучий захід. Деяке програмне забезпечення для освіти використовується майже в кожному навчальному закладі, і воно стане повсюдно в міру збільшення кількості рішень і діапазону можливостей цих рішень. Спектр інструментів в індустрії освіти розширюється, особливо з появою програмного забезпечення з елементами штучного інтелекту та визнанням приватним сектором величезного потенціалу, пов'язаного з розробкою програмного забезпечення для освітньої галузі [1]–[5].

Освітнє програмне забезпечення є надзвичайно корисним для вчителів, адміністраторів, учнів і батьків. Рішення в цих категоріях надають користувачам низку переваг, від покращення видимості та поширення вмісту до аналітики та кращих каналів зв'язку. Розумний контент, покращена комунікація, консолідація даних та інформації, а також покращена ефективність та ефективність — усі переваги, які супроводжують впровадження освітнього програмного забезпечення [1]–[5].

Розумніший контент – освітнє програмне забезпечення може змінити спосіб створення та представлення контенту учням. Цифровий контент із вбудованими можливостями штучного інтелекту може запропонувати матеріал, який адаптується до рівня знань учнів. Цей «розумний» контент легко змінюється, поширюється та повторно використовується різними поколіннями студентів. Цифровий контент містить додатковий бонус нагляді викладачів і часто включає вбудовані оцінки, які вчителі можуть використовувати для перевірки та відстеження результатів учнів. Комунікація вчителя, батьків та учнів — чітке спілкування між вчителями, батьками та учнями є життєво важливим для процесу навчання. Щоб правильно навчати учнів, потрібен сильний рівень поваги та довіри [1]–[5].

Освітнє програмне забезпечення та програми допомагають розвивати міцні зв'язки між усіма сторонами за допомогою форумів, порталів та інших інтерактивних елементів. Батьки та учні мають інструменти, які сприяють легкому спілкуванню між ними, вчителями та адміністраторами. Чіткі та ефективні канали зв'язку сприяють порозумінню між різними сторонами, які інакше можуть бути розрізненими. Студенти можуть ставити запитання вчителям, щоб з'ясувати неясності щодо матеріалу курсу та методів оцінювання, а батьки можуть мати чітке вікно в результативності своїх дітей [1]–[5].

Консолідація інформації – освітнє програмне забезпечення розроблене для об'єднання пов'язаної інформації в централізованому репозиторії. Консолідація всіх пов'язаних даних у рамках однієї програми дозволяє краще

контролювати та отримувати точніші уявлення з цих даних. Адміністратори та спеціалісти з освіти можуть використовувати ці знання для керівництва як стратегією навчання на нижчому рівні, так і адміністративною політикою на вищому рівні. Підвищення ефективності є метою будь-якої установи, а програмне забезпечення для освіти допомагає організаціям досягати цих цілей. Вчителі можуть використовувати такі ресурси, як програмне забезпечення для керування класом, щоб обмежити відволікання та покращити концентрацію на матеріалі курсу. Адміністратори можуть використовувати знання, надані інформаційними системами, для формування політики та адміністративних рішень [1]–[5].

Освітнє програмне забезпечення також економить адміністраторам і вчителям масу часу на адміністративні завдання, як-от виставлення оцінок, і дає їм можливість знайти більш творчі способи покращити свої відповідні результати [1]–[5].

Велика кількість доступних освітніх програм і широта функціональних можливостей, які вони охоплюють разом, надають викладачам інструменти, необхідні для створення кращого навчального середовища для своїх учнів. Це може мати форму цифрових матеріалів, доповнених програмним забезпеченням в класі, використання системи управління навчанням, інструментів керування класом, призначених для того, щоб студенти не виконували завдання, або щось зовсім інше [1]–[5].

Вчителі адаптують свої уроки та змінюють свої стилі викладання відповідно до потреб кожної групи учнів. Освітнє програмне забезпечення допомагає вчителям змінювати середовище навчання, надаючи інструменти, які можуть задовольнити будь-який стиль навчання [1]–[5].

Основними групами користувачів освітнього програмного забезпечення є вчителі, учні, батьки та шкільні адміністратори. Кожна група використовує освітнє програмне забезпечення, щоб покращити свій навчальний досвід, зменшити неефективність та полегшити спілкування. Більшість освітнього програмного забезпечення призначено для

адміністраторів і вчителів, щоб створювати контент, обмінюватися уроками, керувати класними кімнатами та зберігати дані, пов'язані зі школою та студентами. Студенти використовують рішення в кількох категоріях, щоб споживати підібраний вміст і плани уроків, складати іспити та навчатися. Батьки підтримують зв'язок як з адміністраторами, так і з вчителями, які надають їм ці інструменти, щоб переконатися, що їхня дитина отримує необхідну допомогу, і залишатися в курсі успішності своїх учнів [1]–[5].

Тенденції, пов'язані з програмним забезпеченням для освіти [1]–[5]:

- використання технологій доповненої і віртуальної реальності (AR/VR), що постійно вдосконалюються. Оскільки такі технології стають доступнішими, а їх застосування стає все більш поширеним, постачальники розробляють програми спеціально для освітньої галузі. Вчителі можуть використовувати гарнітури віртуальної реальності, щоб забезпечити додаткове підкріплення для складних концепцій, доповнити звичайні навчальні матеріали або просто представити класну технологію, яка буде цікавою для студентів. AR/VR має потенціал для революції в класній кімнаті, надаючи вчителям інструмент, який вони можуть використовувати для глибокого розуміння та збереження предметного матеріалу [1]–[5];

- використання технологій штучного інтелекту (ШІ) на заняттях. Помічники вчителя зі штучним інтелектом (AITA) використовуються в певних класах (на деяких заняттях), щоб полегшити навантаження на вчителів і звичайних помічників. Ці цифрові помічники викладають завдання студентів, які зазвичай займають години, щоб вихователі розбиралися. Оскільки ці помічники є додатками для машинного навчання, чим більше питань вони обробляють і чим ширший діапазон тем вони охоплюють, тим точнішими вони стають. Потім вчителі можуть використати додаткові години, звільнені цими програмами, щоб зосередитися на справі, що приносить додаткову вартість [1]–[5];

- цифрові навчальні матеріали – цифрові версії матеріалів набагато легше модифікуються та розповсюджуються і можуть бути вбудовані з

додатковими функціями, щоб допомогти студентам під час навчання. Ці матеріали містять відомості, які можуть бути оцінені та проаналізовані школярами, що заощаджують дорогоцінний час вчителів [1]–[5].

Потенційними проблемами з програмним забезпеченням для освіти є такі [1]–[5]:

– конфіденційність. Однією з найбільших проблем, з якими стикається індустрія освітнього програмного забезпечення, є конфіденційність. Збільшення використання програмного забезпечення в секторі освіти означає, що навчальні заклади збирають і розміщують більше інформації. Особливо в навчальних закладах батьки та спільноти висловлюють занепокоєння щодо конфіденційності, та не хочуть, щоб кожна дрібниця інформації про своїх учнів збиралася, перераховувалася й аналізувалася. Оскільки штучний інтелект проникає в клас, а Інтернет речей (IoT) збирає та передає інформацію про кожен річ, що відбувається в класі, виникає питання, скільки даних має право збирати школа. Що стосується неповнолітніх, то чи мають вони якісь права на конфіденційність чи вони належать батькам? Чи повинні студенти погодитися на обмін даними чи відмовитися від них? Це всі питання є питаннями, відповіді на які буде знайдено в найближчі роки, оскільки освітнє програмне забезпечення та технології продовжують модифікуватися та впливати на галузь [1]–[5];

– безпека — чим більше даних та інформації має організація, тим привабливішою стає перспектива крадіжки цієї інформації для поганих суб'єктів. Хоча школи можуть не зберігати найціннішу інформацію в світі, платіжні дані та інформація, пов'язана з особою, а також конфіденційна особиста інформація студента, можуть бути в архіві. З цієї причини безпека є першорядною в освітньому програмному забезпеченні. Освітнє програмне забезпечення, як правило, не рекламує безпеку як одну зі своїх відмінних рис, але це може змінитися, оскільки в програмах навчальних закладів зберігається більш конфіденційна інформація [1]–[5];

– відволікання – оскільки технології в класі стають все більш розповсюдженими, однією з головних проблем є можливість відволікання. Студенти, озброєні планшетами та іншими пристроями, підключеними до Інтернету, легко відволікаються від зосередження на предметному матеріалі під час уроку. Для розв'язання цієї проблеми необхідно впровадження програмного забезпечення для керування класом, яке дозволяє вчителям контролювати та блокувати учням використання певних вебсайтів. Управління класом стало невід'ємним інструментом, який дозволяє вчителям гарантувати, що їхні учні виконують завдання [1]–[5];

– неправильне впровадження. Хоча це є проблемою в кожному секторі, однією з найбільш кричущих потенційних проблем для освітнього програмного забезпечення є неправильний вибір та впровадження програмного забезпечення. Впровадження включає в себе відповідне навчання співробітників, які будуть використовувати програмне забезпечення, що часто не помічають навчальні заклади, які не мають достатньо кваліфікованого ІТ-персоналу. Якщо вибрано неправильне програмне забезпечення або воно реалізовано неправильно, а установа та її співробітники не знають, як належним чином скористатися можливостями, які надає програма, то, швидше за все, у майбутньому виникнуть проблеми. Щоб уникнути цих проблем, необхідно забезпечити ретельний процес перевірки та відбору, належне впровадження та високоякісне навчання для співробітників, які будуть використовувати програмне забезпечення [1]–[5].

Категоріями освітнього програмного забезпечення є такі [1]–[5]:

– курсове програмне забезпечення (courseware) – це термін, який поєднує слова "курс" і "програмне забезпечення". Спочатку він використовувався для опису додаткового навчального матеріалу, призначеного як набори для вчителів або тренерів, або як навчальні посібники для студентів, зазвичай упакованих для використання з комп'ютером. Значення та використання терміну розширилося і може стосуватися всього курсу та будь-якого додаткового матеріалу, якщо використовується для

посилання на онлайн-класи або класи у «комп'ютерному форматі». Багато компаній використовують цей термін, щоб описати весь «пакет», що складається з одного «класу» або «курсу» разом із різними уроками, тестами та іншим необхідним матеріалом. Самі навчальні програми можуть бути в різних форматах: деякі доступні лише в Інтернеті, наприклад, на веб-сторінках, а інші можна завантажити у вигляді файлів або інших типів документів. Більшість провідних освітніх компаній замовляють або включають курси до своїх навчальних пакетів [1]–[5];

– навчальні посібники (інтерактивні дошки) – навчальні програми, призначені для використання в шкільних класах. Зазвичай таке програмне забезпечення може проектуватися на велику дошку в передній частині класу та/або одночасно працювати в мережі настільних комп'ютерів у класі. Найбільш помітними є SMART дошки, які використовують SMART Notebook для взаємодії з дошкою, що дозволяє використовувати ручки для цифрового малювання на дошці. Цей тип програмного забезпечення часто називають програмним забезпеченням для управління класом. Хоча вчителі часто вибирають використовувати освітнє програмне забезпечення з інших категорій у своїх ІТ-пакетах (наприклад, довідкові роботи, програмне забезпечення для дітей), виросла ціла категорія освітнього програмного забезпечення, спеціально призначеного для допомоги в навчанні в класі. У цій категорії брендинг був менш сильним, ніж у тих, хто орієнтований на домашніх користувачів. Назви програмного забезпечення часто дуже спеціалізовані та виробляються різними виробниками, включаючи багато відомих видавництв освітніх книг [1]–[5];

– програмне забезпечення для оцінювання – виникло у зв'язку з впливом екологічної шкоди та потребою, щоб заклади стали «безпаперовими», все більше навчальних закладів шукають альтернативні способи оцінювання та тестування, які завжди традиційно споживають величезну кількість паперу. Представником такого програмного забезпечення є Moodle — дуже популярний веб-сайт для оцінювання, який використовується вчителями для

надсилання завдань та оцінювання робіт учнів. Програмне забезпечення для оцінювання відноситься до програмного забезпечення, основною метою якого є оцінювання та тестування студентів у віртуальному середовищі. Програмне забезпечення для оцінювання дозволяє студентам складати тести та іспити за допомогою комп'ютера, зазвичай під'єданого до мережі. Потім програмне забезпечення оцінює кожне завдання тесту та виводить результати для кожного студента. Програмне забезпечення для оцінювання доступне в різних методах доставки, найпопулярнішими є програмне забезпечення для самостійного розміщення, програмне забезпечення онлайн та ручні системи голосування. Доступні фірмове програмне забезпечення та системи програмного забезпечення з відкритим кодом. Іншими популярними міжнародними системами оцінювання є такі: Google Classroom, Blackboard Learn та EvaluNet XT [1]–[5];

– довідкове програмне забезпечення – видання, часто доповнені новим мультимедійним вмістом, у тому числі стисненим відео та звуком. У новітніх продуктах використовуються інтернет-технології. Вікіпедія та її допоміжні частини є прикладом освітнього довідкового програмного забезпечення. Раніше енциклопедії та словники складали свій зміст на основі запрошених та закритих команд спеціалістів. Концепція Wiki дозволила розробити спільні довідкові роботи через відкриту співпрацю, залучаючи експертів і неекспертів [1]–[5];

– індивідуальні платформи – індивідуальні пристрої, зручні для дітей. Апаратне та програмне забезпечення зазвичай об'єднуються в один продукт, наприклад, схожий на дитячий ноутбук. Клавіатура ноутбука для дітей молодшого віку дотримується алфавітного порядку і порядку qwerty для старших. Найвідомішим прикладом є продукти Leapfrog. Сюди входять оригінально розроблені ручні консолі з різноманітними під'єднуваними навчальними ігровими картриджами та електронними пристроями, схожими на книги, в які можна завантажувати різноманітні електронні книги. Ці продукти є більш портативними, ніж портативні комп'ютери, але мають

набагато обмеженіший діапазон цілей, зосереджений на грамотності. У той час як основні операційні системи розроблені для загального використання та більш-менш налаштовані для навчання лише доданими до них наборами програм, різноманітні виробники програмного забезпечення, особливо дистрибутиви Linux, прагнули надати інтегровані платформи для спеціальної освіти [1]–[5];

– корпоративне навчання та вища освіта, як правило, включає серверні програми з високим ступенем стандартизації. Освітнє програмне забезпечення працює переважно на серверах, які можуть знаходитися дуже далеко від реального користувача. Користувач отримує лише крихітні фрагменти навчального модуля або тесту, які подаються через Інтернет один за іншим. Серверне програмне забезпечення вирішує, який навчальний матеріал розповсюдити, збирає результати та відображає прогрес для викладачів. Освітнє програмне забезпечення перетворилося на онлайн освітню службу [1]–[5].

Існують також дуже специфічні ніші для освітнього програмного забезпечення, зокрема [1]–[5]:

– інструменти вчителя та програмне забезпечення для керування класом (ПЗ для дистанційного керування та моніторингу, програмне забезпечення для передачі файлів, презентатори, безкоштовні інструменти, та ін.);

– програми для проведення тестів на перевірку навичок водіння;

– інтерактивне геометричне програмне забезпечення;

– програмне забезпечення для вивчення мови;

– програмне забезпечення Mind Mapping, яке є центром для обговорення, допомагає зробити заняття більш інтерактивними та допомагає студентам у навчанні, есе та проектах;

– програмне забезпечення для проєктування та друку макетів карток для використання в навчанні;

– програмне забезпечення для ведення нотаток;

- програмне забезпечення для імітації розтину людей і тварин (використовується на курсах медичного та ветеринарного коледжу);
- програмне забезпечення для репетитора орфографії;
- репетитори з набору тексту;
- програмне забезпечення для тренування навичок з читання;
- освітнє програмне забезпечення для медицини та охорони здоров'я;
- відеоігри також можна використовувати, щоб навчити користувача технологічній грамотності. Деякі операційні системи та мобільні телефони мають ці функції. Яскравим прикладом є Microsoft Solitaire, який був розроблений для ознайомлення користувачів із використанням графічних інтерфейсів користувача, особливо миші та техніки перетягування. Гейміфікація — це використання елементів ігрового дизайну в неігрових контекстах і, як виявилось, ефективна для мотивації зміни поведінки. Розглядаючи ігрові елементи як «мотиваційні можливості» та формалізуючи зв'язок між цими елементами та мотиваційними можливостями [1]–[5].

Таким чином, у наш час існує досить багато видів програмного забезпечення, що використовується у навчальних цілях. Важливим є програмне забезпечення для надання інформації про успішність здобувачів освіти, контролю розкладу уроків, оцінок та домашнього завдання учнів у вигляді електронного щоденника.

1.2 Дослідження програмних засобів для підтримки освітнього процесу

У наш час використовується багато видів програмного забезпечення, що застосовується у навчальних цілях. Проаналізуємо деякі з них.

Програмне забезпечення Canvas LMS є системою управління навчанням (LMS) та служить центральним центром, куди викладачі можуть завантажувати та організовувати матеріал курсу для доступу учнів або студентів. Школи та університети використовують систему управління

навчанням Canvas LMS, щоб зменшити використання паперу, а також зробити документи доступними поза аудиторією. Відкрита, інтуїтивно зрозуміла система Canvas LMS використовує хмарний підхід та є найпоширенішою системою керування навчанням у коледжах та університетах Північної Америки. У школах Canvas LMS стає цифровим навчальним центром, який об'єднує студентів, викладачів та інструменти — у фізичному, змішаному чи повністю онлайн навчанні [6].

Програмне забезпечення Canvas LMS (рис. 1.1) має наступні особливості та переваги:

- підтримка багатьох мов;
- має засоби пошуку необхідної інформації;
- дозволяє доступ іншим користувачам, зокрема батькам учнів, до отримання інформації про результати навчання;
- дозволяє створювати онлайн-засоби навчання.

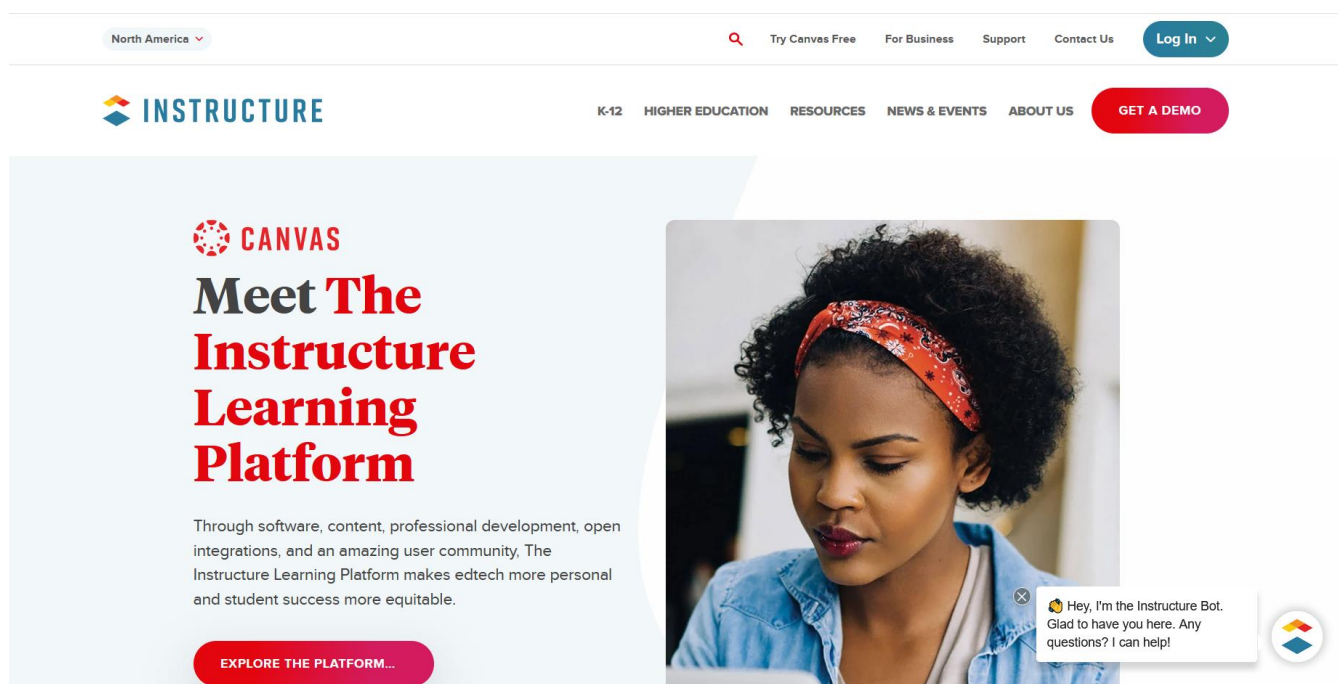


Рисунок 1.1 – Інтерфейс програмного забезпечення Canvas LMS

Недоліком програмного забезпечення Canvas LMS є те, що воно містить багато інформації, що дещо ускладнює його застосування на практиці.

Програмне забезпечення Google Classroom є системою управління навчанням, де викладачі можуть завантажувати та організовувати матеріал курсу для доступу учнів або студентів. Заклади освіти використовують систему підтримки навчального процесу Google Classroom, щоб зменшити використання паперу, а також зробити документи доступними поза аудиторією. Відкрита, інтуїтивно зрозуміла система Google Classroom використовує хмарний підхід та є однією з широко використовуваних системою підтримки процесу навчання у коледжах та університетах [7].

Програмне забезпечення Google Classroom (рис. 1.2) має наступні особливості та переваги:

- підтримка різних мов;
- має засоби пошуку необхідної інформації;
- дозволяє створювати онлайн-засоби навчання.

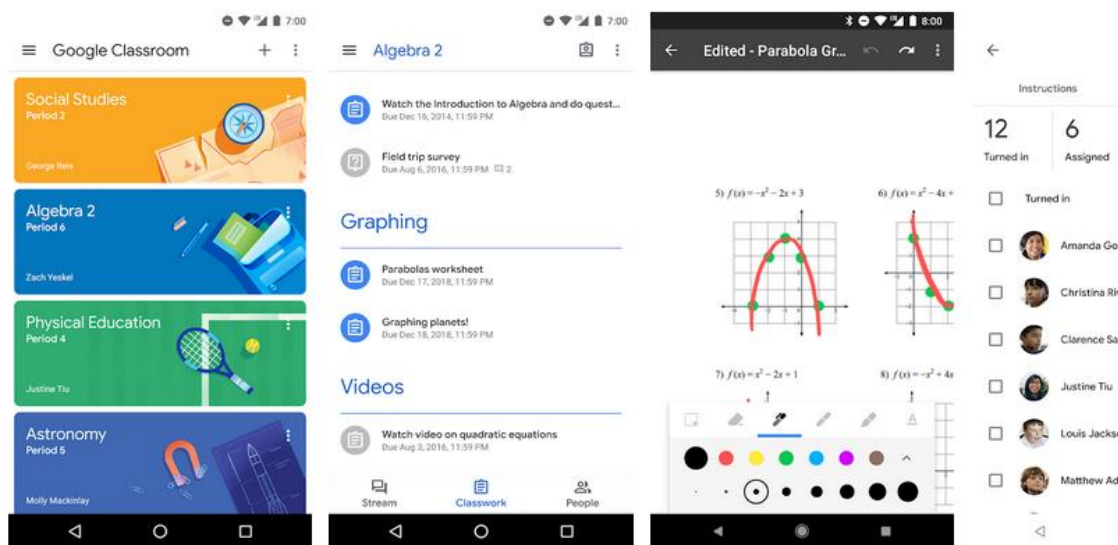


Рисунок 1.2 – Інтерфейс програмного забезпечення Google Classroom

Недоліком програмного забезпечення Google Classroom є те, що воно не дозволяє доступ іншим користувачам, зокрема батькам учнів, до отримання інформації про результати навчання, що дещо ускладнює його застосування на практиці.

Навчальне програмне забезпечення для управління навчанням ProProfs (рис. 1.3) — це досить зручне програмне забезпечення для хмарного навчання для розгортання навчальних програм спільноти. Платформа є легкою, зручною та зрозумілою, і вона може допомогти налаштувати онлайн-навчання співробітників за лічені хвилини. Люди з будь-яким рівнем досвіду можуть використовувати його для підтримки як до, так і після навчання. Інструменти для створення вікторини, опитування, можливості спільної роботи та розширені звіти — все це частина цього програмного забезпечення для навчання на робочому місці. Ця програма містить різноманітні функції, зокрема запитання й відповіді, що сприяє обміну знаннями та соціальному навчанню. ProProfs також покращує звітність та аналітику, необхідну для отримання уявлень про участь у курсах, показники завершення, рівні залучення та прогалини в знаннях. Крім того, тут можна керувати завданнями курсу, відстеженням, групами учнів і адміністраторами груп. Крім того, тут є бібліотека преміум-класу з більш ніж 100 готових до використання курсів і шаблонів. Ці матеріали полегшують виконання брендування. Цей інструмент має дуже велику кількість корисних функцій; однак суттєвим недоліком є те, що лише преміум-план може надати повний доступ до всіх функцій програм [8].

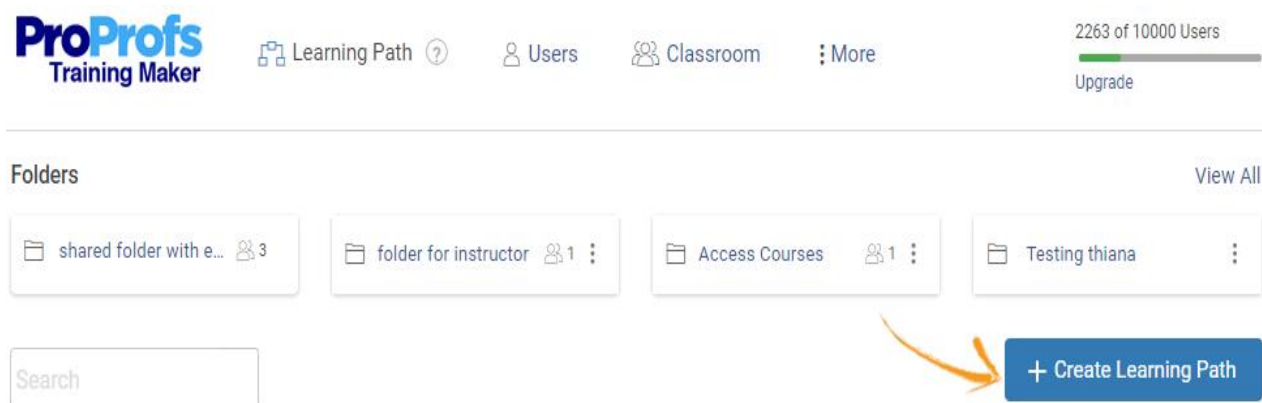


Рисунок 1.3 – Інтерфейс програмного забезпечення ProProfs

Програмне забезпечення ProProfs має наступні особливості та переваги [8]:

- підтримка різних мов;
- легкість, зручність та зрозумілість інтерфейсу;
- дозволяє створювати онлайн-засоби навчання;
- інструменти для створення опитувань;
- можливості спільної роботи;
- розширені звіти.

Недоліком програмного забезпечення ProProfs є те, що воно не дозволяє доступ іншим користувачам, зокрема батькам учнів, до отримання інформації про результати навчання, що дещо ускладнює його застосування на практиці. Іншим недоліком є те, що доступ до всіх функцій програми можливий лише через купівлю преміум-плану [8].

Навчальне програмне забезпечення для управління навчанням Schoology (рис. 1.4) є досить простим та зручним завдяки своєму приємному дизайну, необмеженим можливостям та вчителям.



Рисунок 1.4 – Інтерфейс програмного забезпечення Schoology

Schoology містить різноманітні функції, наприклад, різні типи доступних елементів оцінювального запитання. Він також може персоналізувати навчання, призначаючи завдання людям. Цей інструмент має засоби для створення та підключення різних навчальних курсів. Schoology об'єднує всі аспекти управління класом в єдиний зручний інтерфейс. Програмне забезпечення Schoology дозволяє інтегруватись з іншим освітнім програмним забезпеченням та інструментами (такими як EduCannon, TurnItIn та Google Drive). Schoology пропонує споживачам настроювану навчальну платформу, за допомогою якої вони можуть брати участь у єдиному цифровому навчанні. Студенти можуть поділитися своїми роботами на місцевому та глобальному рівні. Крім того, програмне забезпечення забезпечує досить адаптивну електронну платформу. Він включає в себе дані, аналітику та індивідуальне навчання, а також надійне спілкування та командну роботу [9].

Програмне забезпечення Schoology має наступні особливості та переваги [9]:

- різні типи доступних елементів оцінювання;
- легкість, зручність та зрозумілість інтерфейсу;
- дозволяє створювати онлайн-засоби навчання;
- інструменти для створення опитувань;
- можливості персоналізувати навчання;
- можливість інтеграції з іншим освітнім програмним забезпеченням

та інструментами.

Недоліком програмного забезпечення Schoology є те, що воно не дозволяє доступ іншим користувачам, зокрема батькам учнів, до отримання інформації про результати навчання, що дещо ускладнює його застосування на практиці. Іншим недоліком є те, що доступ до всіх функцій програми можливий лише через купівлю преміум-плану [9].

Навчальне програмне забезпечення TalentLMS дозволяє розділити сайт на різні фракції, щоб дозволити різним відділам використовувати його. У

цьому програмному забезпеченні навчання організовано досить зручно, існує можливість отримати інформацію різними способами. Застосунок дозволяє адаптовуватися до потреб студентів та учнів, що дозволяє команді викладачів створювати або купувати курси відповідно до конкретних вимог. Це програмне забезпечення також може створювати графіки та статистичні дані для клієнтів на основі їхнього використання та відстеження групового чи індивідуального розвитку. TalentLMS має чудову аналітику, можливості письма, легко налаштовується, має численні корисні інтеграції та, перш за все, є економічно ефективним LMS. Для вирішення будь-яких проблем клієнтів є база знань, менеджер для роботи з клієнтами, служба підтримки та канал чату в реальному часі. За допомогою TalentLMS можна робити й підтримувати особливості в навчальних треках для персоналу та клієнтів компанії. Цей навчальний застосунок також ефективно працює, коли клієнта видаляють із системи або додають до нової групи (рис. 1.5). На жаль, деякі функції комп'ютера недоступні в мобільному додатку, наприклад перегляд відео під час читання тексту на уроці, який доступний лише користувачам комп'ютера [10].

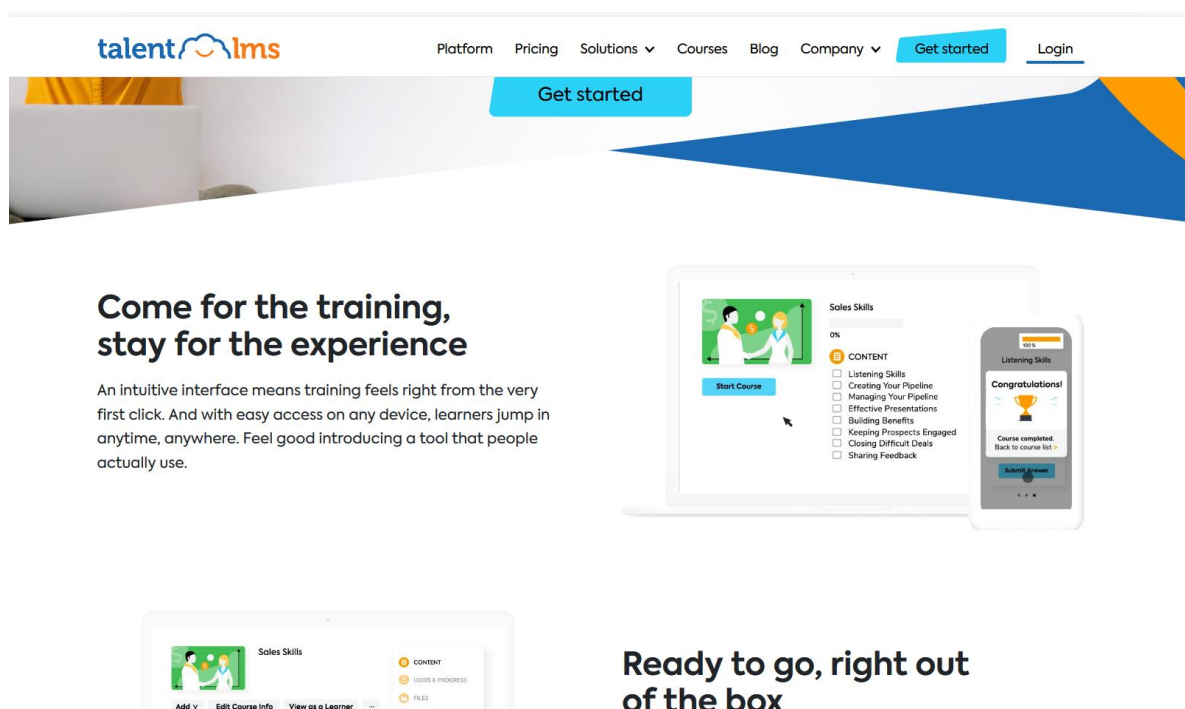


Рисунок 1.5 – Інтерфейс програмного забезпечення TalentLMS

Програмне забезпечення TalentLMS має наступні особливості та переваги [10]:

- легкість, зручність та зрозумілість інтерфейсу;
- дозволяє створювати онлайн-засоби навчання;
- інструменти для створення опитувань;
- можливість адаптовуватися до потреб студентів та учнів;
- інструменти для створення графіків та інших елементів звіту.

Недоліком програмного забезпечення TalentLMS є те, що воно не дозволяє доступ іншим користувачам, зокрема батькам учнів, до отримання інформації про результати навчання, що дещо ускладнює його застосування на практиці. Іншим недоліком є те, що деякі функції програмного забезпечення є недоступними в мобільному застосунку [10].

Навчальне програмне забезпечення Litmos – це програмне забезпечення для навчання, яке просте та зрозуміле у використанні (рис. 1.6).

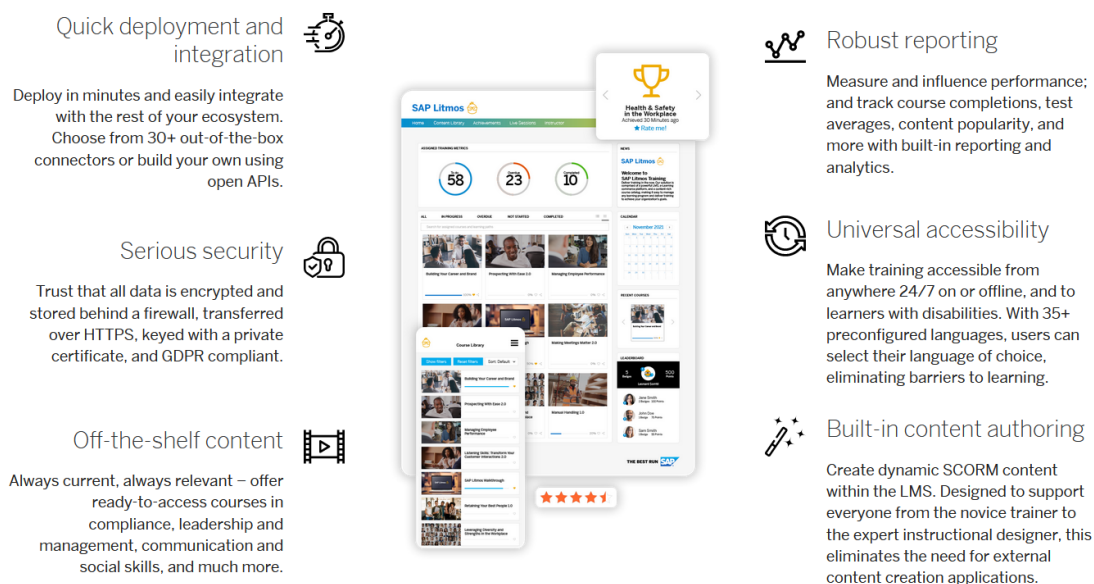


Рисунок 1.6 – Інтерфейс програмного забезпечення Litmos

Програмне забезпечення Litmos в основному використовується для навчання співробітників, клієнтів, партнерів і відповідності вимогам. Litmos надає багато унікальних характеристик, які є критичними для бізнесу. По-

перше, це сприяє покращенню тренування як внутрішніх, так і зовнішніх груп. Це також покращує швидкість, з якою можна покращити свої навички. Крім того, Litmos пропонує конструктор онлайн-курсів у вигляді модуля, який використовує інструменти розробки контенту, які підтримують численні формати. Застосунок також забезпечує віртуальне навчання під керівництвом інструктора, а також іспити та вікторини для вимірювання збереження інформації. Litmos допомагає розвивати електронну комерцію, легко керувати програмами та оновлювати кожну з них із повідомленнями та сповіщеннями. Litmos надає персоналізовані маршрути навчання, дані та інформаційні панелі для оцінки ефективності та впливу навчання. Гейміфікація та дошки лідерів доступні, щоб мотивувати клієнтів і надавати точні та змістовні опитування та відгуки. Команда підтримки вирішує виробничі труднощі протягом тривалого періоду [11].

Програмне забезпечення Litmos має наступні особливості та переваги [11]:

- легкість, зручність та зрозумілість інтерфейсу;
- дозволяє створювати онлайн-засоби навчання;
- інструменти для створення опитувань;
- можливість застосовувати засоби електронної комерції;
- засоби підтримки віртуального навчання під керівництвом інструктора.

Недоліком програмного забезпечення Litmos є те, що воно не дозволяє доступ іншим користувачам, зокрема батькам учнів, до отримання інформації про результати навчання, що дещо ускладнює його застосування на практиці. Іншим недоліком є те, що цей програмний продукт не є безкоштовним [11].

Навчальне програмне забезпечення Wisenet – це програмне забезпечення для навчання у вигляді хмарної платформи, яка дозволяє навчальним закладам підтримувати ефективні результати навчання,

активізувати залучення зацікавлених сторін, підвищувати продуктивність і розвивати бізнес [12].

Wisenet є провідним постачальником хмарних застосунків для освітніх закладів. Він працює безпосередньо з клієнтами, щоб послідовно розвивати всі продукти та послуги та залишатися попереду конкурентів, використовуючи провідні світові технології. Це програмне забезпечення дає можливість призначити право власності окремим потенційним клієнтам, що дійсно корисно. Це програмне забезпечення також чудово підходить для моніторингу інформації про клієнта та зберігання всіх документів в одному місці. Програмне забезпечення Wisenet допомагає заощадити час на процесах, які в іншому випадку зайняли б багато часу. Крім того, кілька користувачів можуть знайти всю документацію та записи клієнта в одному місці лише кількома клацаннями миші в режимі реального часу [12].

Програмне забезпечення Wisenet має наступні особливості та переваги [12]:

- легкість, зручність та зрозумілість інтерфейсу;
- дозволяє створювати онлайн-засоби навчання;
- інструменти для створення опитувань;
- засоби підтримки віртуального навчання під керівництвом інструктора.

Недоліком програмного забезпечення Wisenet є те, що воно не дозволяє доступ іншим користувачам, зокрема батькам учнів, до отримання інформації про результати навчання, що дещо ускладнює його застосування на практиці. Іншим недоліком є те, що цей програмний продукт не є безкоштовним [12].

Таким чином, у наш час досить активно використовуються програмні засоби для підтримки освітньої діяльності. Порівняльний аналіз програмного забезпечення для підтримки освітньої діяльності наведено у табл. 1.1.

Таблиця 1.1 – Програмне забезпечення для підтримки освітньої діяльності

Критерій, за яким порівнюють	Canvas LMS	Google Classroom	ProProfs	Schoology	TalentLMS
Зручність та зрозумілість інтерфейсу	+	+	+—	+	+
Онлайн-засоби навчання	+	+	—	+	+
Інструменти для створення опитувань	—	+	+—	+	+
Засоби підтримки віртуального навчання	+	—	+	+	+
Доступ для батьків	—	—	—	—	—
Безкоштовність	—	+	—	—	—

З приведеної таблиці видно, що не всі програмні засоби надають можливість роботи з ними безкоштовно, що для деяких навчальних закладів є досить суттєвим недоліком. Також недоліком існуючого програмного забезпечення є те, що воно не дозволяє доступ іншим користувачам, зокрема батькам учнів, до отримання інформації про результати навчання, що дещо ускладнює його застосування на практиці. Отже доцільною є розробка програмного забезпечення для надання інформації про успішність здобувачів освіти, контролю розкладу уроків, оцінок та домашнього завдання учнів у вигляді електронного щоденника.

Таким чином, необхідно створити програмний продукт, який імітуватиме електронний щоденник здобувача освіти (школяра) та матиме додатковий функціонал. При розробці програми необхідно забезпечити такі функціональні вимоги:

- реєстрація учнів, вчителів за допомогою адміністратора;
- редагування списку класів;
- збереження розкладу, домашніх завдань, оцінок та поміток для учнів;
- збереження розкладу дзвінків;
- відображення успішності учнів;
- підрахунок та відображення підсумкових оцінок.

1.3 Висновки за розділом 1

Проаналізовано загальні поняття про програмні засоби для підтримки освітньої діяльності. Визначено, що освітнє програмне забезпечення – це масовий, всеохоплюючий термін, який використовується для позначення будь-якого програмного забезпечення, призначеного для використання в освітній галузі. Проаналізовано тенденції, пов'язані з програмним забезпеченням для освіти, а також потенційні проблеми використання такого програмного забезпечення. Описано категорії освітнього програмного забезпечення.

Визначно, що у наш час існує досить багато видів програмного забезпечення, що використовується у навчальних цілях. Важливим є програмне забезпечення для надання інформації про успішність здобувачів освіти, контролю розкладу уроків, оцінок та домашнього завдання учнів у вигляді електронного щоденника.

Виконано дослідження програмних засобів для підтримки освітнього процесу. Визначено, що не всі програмні засоби надають можливість роботи з ними безкоштовно, що для деяких навчальних закладів є досить суттєвим недоліком. Також недоліком існуючого програмного забезпечення є те, що воно не дозволяє доступ іншим користувачам, зокрема батькам учнів, до

отримання інформації про результати навчання, що дещо ускладнює його застосування на практиці. Отже доцільною є розробка програмного забезпечення для надання інформації про успішність здобувачів освіти, контролю розкладу уроків, оцінок та домашнього завдання учнів у вигляді електронного щоденника.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування та інших технологій розробки

Для створення програмного забезпечення для роботи електронного щоденника здобувача освіти було обрано мову програмування C# [13]-[15].

Мова програмування C# — це об'єктно-орієнтована мова програмування. Ефективно використовувати цю мову можна за допомогою використання класів і методів. Завдяки методам програми на C# є досить зручними для читання, зрозумілими та модульними. Використання методів позбавляють програмістів від постійного написання одних і тих самих кодів шляхом копіювання та вставки. Таким чином, програмні коди на C# стають коротшими і зрозумілішими. Також можна використовувати вираз функції замість методу, вони обидва означають майже те саме. У інших мовах програмування, а саме в таких мовах, як Fortran, Qbasic, Turbo C і Pascal, структурне програмування було домінуючим. Потік програми на C# іде зверху вниз. Сьогодні майже не використовуються структуровані мови програмування. Компанії, що займаються програмним забезпеченням, тепер використовують у своїх проектах мови об'єктно-орієнтованого програмування [13]-[15].

C# є мовою програмування, розробленою Microsoft, яка дуже схожа на мови C++ і Java, але C# має 100% техніку об'єктної орієнтації на відміну від C++. На відміну від Java, в C# можна використовувати покажчики. Таким чином, можна працювати в гармонії зі старими програмними компонентами. Код C#, на відміну від C++ або Visual Basic, не компілюється безпосередньо в машинний код. Спочатку він компілюється в проміжний код. Файл цього першого скомпільованого коду називається асамблеєю, а його розширення — exe. Коли цей файл потрібно запустити, запускається .Net Framework і перетворює проміжний код у машинний код, який може розуміти комп'ютер. Тому програма .Net Framework має бути встановлена на комп'ютері, щоб програма, яка була написана на C#, запускалася на цьому комп'ютері, оскільки

.Net Framework перетворює проміжний код у код, який комп'ютер може зрозуміти [13]-[15].

.Net Framework поміщає створені нею машинні коди в пам'ять на тимчасовий період, якщо ті самі коди потрібно запустити знову, замість того, щоб знову конвертувати з проміжного коду в машинний код, він використовує машинні коди, збережені в цій пам'яті. Тому під час першого запуску створеної програми вона може працювати трохи повільно, але в подальших дослідженнях вона буде значно прискорюватися. Перетворення коду спочатку в проміжний код, а не в прямий машинний код, на C# має деякі переваги. Найважливішим з них є те, що програму набагато легше запускати на різних операційних системах, ніж раніше [13]-[15].

Обґрунтування вибору мови програмування для розробки програмного забезпечення для роботи електронного щоденника здобувача освіти наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки програмного забезпечення для роботи електронного щоденника здобувача освіти

Критерій порівняння мов програмування	Мова програмування		
	C#	Java	PHP
Зручність для створення ПЗ для роботи електронного щоденника	+	—	—
Робота з базами даних	+	+—	+
Об'єктно-орієнтовані засоби	+	+	+—
Засоби захисту даних	+	+—	+
Необхідність наявності проміжного програмного забезпечення	+	+	+—

Оскільки машинний код не є переносимим, коли програми компілюються безпосередньо в машинний код, відповідну програму потрібно скомпілювати відповідно до конкретної операційної системи. У той час як машинний код є переносимим, загальним кодом, його можна перетворити на машинний код різними програмами в операційних системах. Наприклад, .Net Framework може працювати у Windows, а Mono — у Linux [13]-[15].

Отже, для реалізації програмного забезпечення для роботи електронного щоденника здобувача освіти обрано мову програмування C#.

2.2 Вибір середовища розробки для створення програмного забезпечення

Оскільки в якості мови програмування було обрано C#, то для створення програмного забезпечення доцільно обрати середовище розробки Microsoft Visual Studio.

Microsoft Visual Studio — це набір продуктів Microsoft, що включає інтегроване середовище розробки програмного забезпечення та безліч інших інструментів. Ці продукти дозволяють розробляти як консольні, так і графічні програми, включаючи підтримку технології Windows Forms, вебсайтів, вебзастосунків, вебсервісів як у рідному, так і в керованому коді для всіх платформ, підтримуваних Microsoft Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework і Microsoft Silverlight [16].

Microsoft Visual Studio — це IDE, створена Microsoft і використовується для різних типів розробки програмного забезпечення, наприклад комп'ютерних програм, вебсайтів, вебпрограм, веб-ервісів та мобільних додатків. Він містить інструменти завершення, компілятори та інші функції для полегшення процесу розробки програмного забезпечення [16].

Обґрунтування вибору середовища розробки для створення програмного забезпечення для роботи електронного щоденника здобувача освіти наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення програмного забезпечення для роботи електронного щоденника здобувача освіти

Критерій порівняння середовищ розробки	Середовище розробки		
	MS Visual Studio	Atom	PyCharm
Зручність використання для створення програмного забезпечення електронного щоденника здобувача освіти	+	+—	+—
Ефективність засобів редагування файлів програмного проєкту	+	+	+—
Наявність засобів для роботи з мовою C#	+	+—	—
Зручність для створення вебзастосунків	+	+	+—
Можливість кросплатформенної розробки	+	+	+—

Таким чином, для створення програмного забезпечення обрано середовище розробки Microsoft Visual Studio.

2.3 Вибір системи керування базою даних

Для реалізації програмного забезпечення для роботи електронного щоденника здобувача освіти обрано систему керування базами даних InterSystems Cache [17].

InterSystems Cache — це система керування базами даних і середовище для швидкої розробки додатків. Cache значно покращує обробку та аналіз складних великих даних, а також розробку мобільних і веб-додатків. Вона забезпечує неперевершену швидкість і продуктивність, масштабованість і надійність завдяки унікальній технології, на якій базується ця система. У той же час потреби в технічному обслуговуванні обладнання та системи зведені до мінімуму. Він надає кілька парадигм для доступу до одних і тих же даних [17].

У таблиці 2.3 наведено порівняння систем керування базами даних для розробки програмного забезпечення для роботи електронного щоденника здобувача освіти.

Таблиця 2.3 – Порівняння систем керування базами даних для розробки програмного забезпечення для роботи електронного щоденника здобувача освіти

Критерій порівняння СКБД	СКБД		
	InterSystems Cache	Oracle	MySQL
Зручність для створення ПЗ електронного щоденника здобувача освіти	+	—	—
Безпека даних	+	+—	+
Низька вартість	+	—	+
Масштабованість	+	+—	+—
Продуктивність	+	+—	+

Таким чином для розробки програмного забезпечення для роботи електронного щоденника здобувача освіти обрано СКБД InterSystems Cache.

2.4 Висновки за розділом 2

Для створення програмного забезпечення для роботи електронного щоденника здобувача освіти було обрано мову програмування C#, яка є об'єктно-орієнтованою мовою програмування та є досить зручною для створення програмного забезпечення та вебзастосунків.

Оскільки в якості мови програмування було обрано C#, то для створення програмного забезпечення обрано середовище розробки Microsoft Visual Studio, що включає в себе набір різних засобів, зокрема, інтегроване середовище розробки програмного забезпечення та велику кількість інших інструментів, які дозволяють розробляти як консольні, так і графічні програми, включаючи підтримку технології вебсайтів, вебзастосунків, вебсервісів для різних операційних систем.

Для реалізації програмного забезпечення для роботи електронного щоденника здобувача освіти обрано систему керування базами даних InterSystems Cache.

3 ОПИС ПРОГРАМИ

3.1 Схема взаємодії основних модулів програмного забезпечення електронного щоденника здобувача освіти

У розробленому програмному забезпеченні електронного щоденника здобувача освіти основними структурними компонентами є:

- модуль, що реалізує основний функціонал програми (програмна частина);
- модуль графічного інтерфейсу, що забезпечує взаємодію користувача з програмою;
- база даних про здобувачів освіти та інших учасників освітнього процесу.

Користувач програмного забезпечення електронного щоденника здобувача освіти використовує графічний інтерфейс для перегляду та редагування даних. Модуль графічного інтерфейсу, використовуючи програмну частину та логіку передає до графічного інтерфейсу потрібні дані з бази даних здобувачів освіти.

Схему взаємодії основних модулів програмного забезпечення електронного щоденника здобувача освіти зображено на рис. 3.1.



Рисунок 3.1 – Схема взаємодії основних модулів програмного забезпечення електронного щоденника здобувача освіти

3.2 Опис основних розроблених структур даних

В ході роботи над програмним забезпеченням електронного щоденника здобувача освіти було створено такі класи:

- DiaryDay;
- Form1;
- LoginForm;
- CreateUserForm;
- AddClassForm;
- BellsForm;
- MonthMarksForm;
- TableForm;
- RatingForm.

Клас DiaryDay представляє дані про день тижня у щоденнику. Дані та методи класу наведено у табл. 3.1.

Таблиця 3.1 – Дані та методи класу DiaryDay

Поля та метод класу	Опис
TableLayoutPanel dayPanel;	Таблиця для дня тижня у щоденнику
string dayName;	Назва дня тижня
DateTime date;	Дата, що відображається у щоденнику
DiaryDay(TableLayoutPanel tp, string n);	Конструктор, у який передається таблиця і назва дня тижня
Control title();	Повертає заголовок для дня тижня у щоденнику

Клас Form1 представляє логіку для головної форми програми та основну логіку в цілому. Дані та методи класу наведені в табл. 3.2.

Таблиця 3.2 – Дані та методи класу Form1

Поля та метод класу	Опис
int access;	Рівень доступу користувача
string uname;	Логін користувача
string classn;	Клас, в якому навчається користувач, якщо це учень
string fio;	Ініціали користувача, якщо це учень
int weeksC;	Перемикач тижнів
List<DiaryDay> days;	Список днів тижня у щоденнику
Form1();	Конструктор класу
void Form1_Load(object sender, EventArgs e);	Реакція на подію завантаження форми
string getStudName();	Повертає ім'я учня
void loadClasses();	Завантажує список класів з БД
void prevButton_Click(object sender, EventArgs e);	Переключення на попередній тиждень
void nextButton_Click(object sender, EventArgs e);	Переключення на наступний тиждень
void addClassButton_Click(object sender, EventArgs e);	Відкриття вікна додавання класу
void addUserButton_Click(object sender, EventArgs e);	Відкриття вікна створення користувача
void updateStudentsList();	Завантажує список учнів класу з БД
void getClassId();	Повертає id класу з БД
void loadSchedule();	Завантажує розклад занять з БД
void clearSchedule();	Очищує розклад
void saveButton_Click(object sender, EventArgs e);	Збереження розкладу у БД
void clearDZ();	Очищує поля для домашнього завдання
void clearMarks();	Очищує поля для оцінок
void updateDZ();	Завантажує домашнє завдання з БД
void button1_Click(object sender, EventArgs e);	Збереження домашнього завдання у БД

Продовження табл. 3.2

Поля та метод класу	Опис
string getStudIdByName(string name);	Повертає id учня за його ініціалами
void button2_Click(object sender, EventArgs e);	Збереження оцінок у БД
void loadMarks();	Завантаження оцінок з БД
void studentsList_SelectedIndexChanged(object sender, EventArgs e);	Зміна активного учня
void saveSaturdayButton_Click(object sender, EventArgs e);	Збереження поля для записів у БД
void loadSaturdayField();	Завантаження поля для записів з БД.
void getMarksButton_Click(object sender, EventArgs e);	Відкриття вікна успішності учня та класу
void editBells_Click(object sender, EventArgs e);	Відкриття вікна редагування дзвінків
void studentsList_DoubleClick(object sender, EventArgs e);	Відкриття вікна з інформацією про учня
void marksAllButton_Click(object sender, EventArgs e);	Відкриття вікна з підсумковими оцінками учня
void loadBells();	Завантаження розкладу дзвінків з БД
classCombo_SelectedIndexChanged(object sender, EventArgs e);	Оновлення щоденника відповідно до обраного класу

Клас AddClassForm представляє логіку для додавання нових класів. Поля та методи класу наведені в таблиці 3.3.

Таблиця 3.3 – Дані та методи класу AddClassForm

Поля та метод класу	Опис
AddClassForm();	Конструктор
bool added;	Індикатор успішного створення
void updateCombo();	Завантаження списку вчителів з БД
void button1_Click(object sender, EventArgs e);	Додавання нового класу до БД
void button2_Click(object sender, EventArgs e);	Закриття вікна

Клас LoginForm представляє логіку форми авторизації. Поля та методи класу наведені в таблиці 3.4.

Таблиця 3.4 – Дані та методи класу LoginForm

Поля та метод класу	Опис
int access;	Рівень доступу користувача
string uname;	Логін користувача
string classes;	Клас користувача, якщо це учень
string fio;	Ініціали користувача, якщо це учень
bool logged;	Успішність авторизації
void loginButton_Click(object sender, EventArgs e);	Авторизація користувача
void LoginForm_FormClosing(object sender, FormClosingEventArgs e);	Закриття вікна і додатку
LoginForm();	Конструктор

Клас BellsForm представляє логіку для форми редагування розкладу дзвоників. Поля та методи класу наведені в таблиці 3.5.

Таблиця 3.5 – Дані та методи класу BellsForm

Поля та метод класу	Опис
BellsForm();	Конструктор
List<TextBox> tbl1;	Список початку уроків
List<TextBox> tbl2;	Список завершення уроків
void button2_Click(object sender, EventArgs e);	Закриття вінка
void button1_Click(object sender, EventArgs e);	Збереження розкладу у БД

Клас CreateUserForm представляє логіку форми створення користувачів. Поля та методи класу наведені в таблиці 3.6.

Таблиця 3.6 – Дані та методи класу CreateUserForm

Поля та метод класу	Опис
CreateUserForm();	Конструктор
bool isstudent;	Індикатор учня чи вчителя
bool added;	Індикатор успішної реєстрації
void comboBox1_SelectedIndexChanged(object sender, EventArgs e);	Зміна вигляду вікна залежно від типу користувача
void loadClasses();	Завантаження списку класів з БД
void button2_Click(object sender, EventArgs e);	Закриття форми
bool isUserExists(string uname);	Перевірка на дублювання даних
void button1_Click(object sender, EventArgs e);	Збереження користувача у БД
string getClassId(string name);	Повертає id класу

Клас RatingForm представляє логіку для відображення успішності учня з конкретного предмету по сьогоднішній день. Поля та методи класу наведені у таблиці 3.7.

Таблиця 3.7 – Дані та методи класу RatingFrom

Поля та метод класу	Опис
RatingForm(string sd, List<DiaryDay> ds);	Конструктор
string sid;	Id учня
Bitmap gr;	Зображення графіку
Graphics g;	Засіб малювання графіку
int step;	Крок для малювання графіку
int height;	Крок у висоту для малювання графіку
List<DiaryDay> days;	Список днів тижня у щоденнику
void loadSubjects();	Завантаження списку предметів
void comboBox1_SelectedIndexChanged(object sender, EventArgs e)	Завантаження оцінок з БД та малювання графік успішності

Клас MonthMarksForm представляє логіку для відображення успішності учня та класу. Поля та методи класу наведені в таблиці 3.8.

Таблиця 3.8 – Дані та методи класу MonthMarksForm

Поля та метод класу	Опис
MonthMarksForm(string studId, string cla, List<DiaryDay> d);	Конструктор
string sid;	Id учня
string cl;	Клас учня
List<DiaryDay> days;	Список днів тижня у щоденнику
int step;	Крок для малювання графіків
int height;	Крок висоти для малювання графіків
Bitmap gr1;	Графік успішності учня
Bitmap gr2;	Графік успішності класу
Graphics g1;	Засоби малювання графіку для учня
Graphics g2;	Засоби малювання графіку для класу
void updatePictures();	Оновлення графіків
List<string> getStudentsId(string clas);	Отримання id учнів класу з БД
void loadClassData(List<SubjData> sd);	Малювання графіку успішності класу
void loadData();	Отримання оцінок учня та малювання графіку успішності

Клас TabelForm представляє логіку для підрахунку підсумкових оцінок учня. Поля та методи класу наведені в таблиці 3.9.

Таблиця 3.9 – Дані та методи класу TabelForm

Поля та метод класу	Опис
TabelForm(string std, List<DiaryDay> Id);	Конструктор
string sid;	Id учня
List<DiaryDay> days;	Список днів тижня у щоденнику
void loadData();	Завантаження оцінок з БД та підрахунок підсумкових оцінок

В процесі розробки програмного забезпечення для роботи електронного щоденника здобувача освіти було реалізовано наступні алгоритми:

- підрахунок та ілюстрація у вигляді графіку успішності учня та класу за місяць;
- підрахунок підсумкових оцінок учня за рік.

Підрахунок та ілюстрація у вигляді графіку успішності учня за місяць відбувається після віднімання від останньої дати тижня одного місяця та визначення предметів, які викладаються у класі, де навчається учень. Після цього для кожної з дат від отриманої до останньої в тиждень виконуються запити до бази даних, що повертають оцінку учня та предмет на якому він її заробив. Підраховується середній бал за день і малюється графік відповідно до значення.

Графік успішності класу заповнюється аналогічно з графіком успішності учня, тільки ці ж самі операції виконуються для кожного учня у класі.

Підрахунок підсумкових оцінок учня за рік починається з розділення датами на перший і другий семестр навчального року (до нового року та після). Потім визначаються предмети, які викладаються у класі де навчається учень і підраховується середній бал для кожного з предметів за 2 семестри. Для визначення річної оцінки використовуються середнє отриманих значень за семестри.

3.3 Основні рішення з розробки бази даних

Схему бази даних програмного забезпечення для роботи електронного щоденника здобувача освіти зображено на рисунку 3.2.

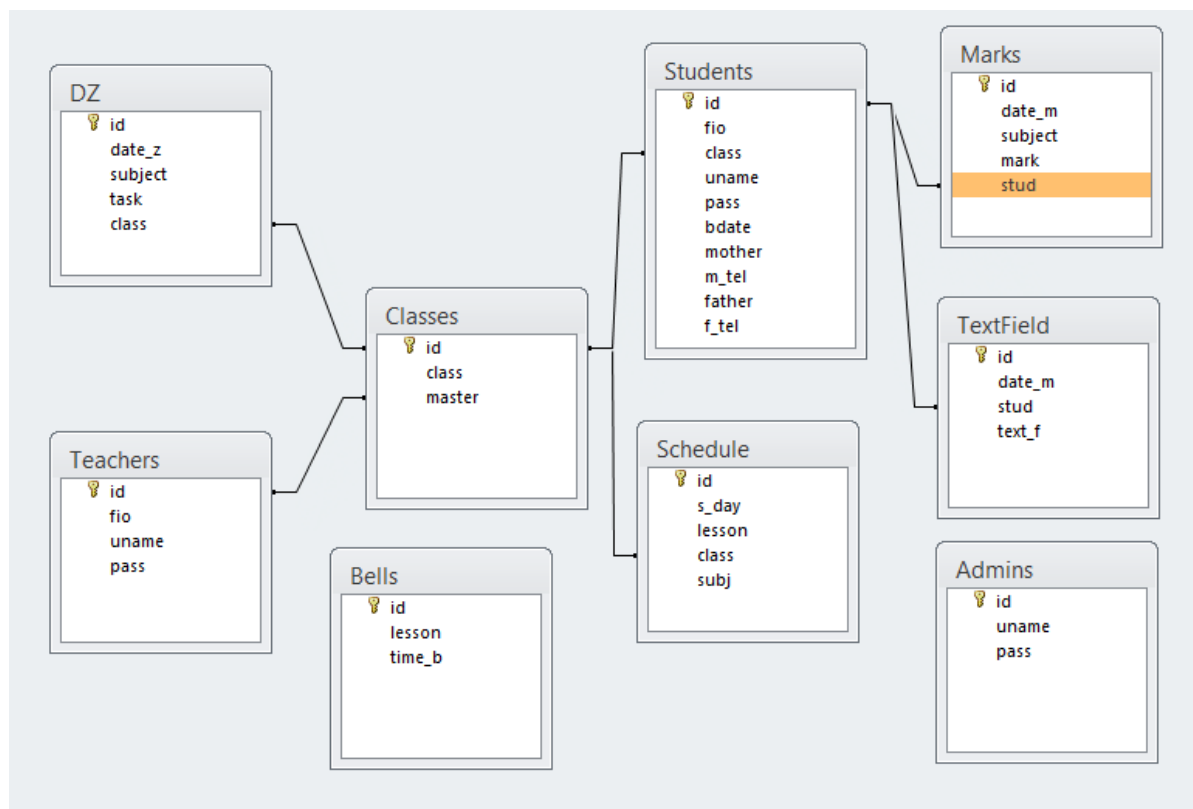


Рисунок 3.2 – Схема бази даних

Основними сутностями (таблицями) розробленої бази даних є: учні (Students), вчителі (Teachers), заняття (Classes), оцінки (Marks), розклад (Schedule), домашні завдання (DZ) та інші.

3.4 Висновки за розділом 3

Визначено, що у розробленому програмному забезпеченні електронного щоденника здобувача освіти основними структурними компонентами є: модуль, що реалізує основний функціонал програми (програмна частина), модуль графічного інтерфейсу, що забезпечує взаємодію

користувача з програмою, база даних про здобувачів освіти та інших учасників освітнього процесу.

Запропоновано схему взаємодії основних модулів програмного забезпечення електронного щоденника здобувача освіти. Користувач програмного забезпечення електронного щоденника здобувача освіти використовує графічний інтерфейс для перегляду та редагування даних. Модуль графічного інтерфейсу, використовуючи програмну частину та логіку, передає до графічного інтерфейсу потрібні дані з бази даних здобувачів освіти.

Описано основні розроблені структури даних (класи) програмного забезпечення: DiaryDay, Form1, LoginForm, CreateUserForm, AddClassForm, BellsForm, MonthMarksForm, TableForm, RatingForm.

Розроблено схему бази даних програмного забезпечення для роботи електронного щоденника здобувача освіти. Відповідно до розробленої схеми даних основними сутностями (таблицями) розробленої бази даних є: учні (Students), вчителі (Teachers), заняття (Classes), оцінки (Marks), розклад (Schedule), домашні завдання (DZ) та інші.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для забезпечення роботи електронного щоденника здобувача освіти. Програма написана на мові С#. База даних програмного забезпечення для роботи електронного щоденника здобувача освіти створена у системі керування базами даних InterSystems Cache.

Програмне забезпечення роботи електронного щоденника здобувача освіти забезпечує виконання таких функцій:

- реєстрація учнів, вчителів за допомогою адміністратора;
- редагування списку класів;
- збереження розкладу, домашніх завдань, оцінок та поміток для учнів;
- збереження розкладу дзвінків;
- відображення успішності учнів;
- підрахунок та відображення підсумкових оцінок.

Для експлуатації програмного забезпечення роботи електронного щоденника здобувача освіти необхідно таке апаратне забезпечення:

- не менше 200 Мб вільного місця на жорсткому диску;
- процесор з тактовою частотою не менше 1,5 ГГц;
- 512Мб оперативної пам'яті;
- наявність пристроїв вводу та виводу.

Також необхідна наявність операційної системи Windows.

Функціональні характеристики розробленого програмного забезпечення для роботи електронного щоденника здобувача освіти наведено у технічному завданні (додаток А). Опис програмного забезпечення для роботи електронного щоденника здобувача освіти та текст програмного забезпечення для роботи електронного щоденника здобувача освіти наведені у додатках Б та В.

4.2 Характеристики програми

Режим роботи програмного забезпечення для роботи електронного щоденника здобувача освіти є цілодобовим та безперервним. Програмне забезпечення для роботи електронного щоденника здобувача освіти використовує прийнятну кількість апаратних ресурсів комп'ютера.

Всі данні програмного забезпечення для роботи електронного щоденника здобувача освіти повинні зберігатися у розробленій базі даних.

ПП включає у себе 25 компонентів, що містить класи, їх реалізацію, файли проекту:

- ElDiary.csproj – містить дані про використані модулі, перелік файлів класів та їх реалізації, посилання на файл ресурсів та загальну інформацію про ПП;

- App.config – файл налаштувань проекту додатка;
- Program.cs – файл для опису запуску додатка;
- AddClassForm.cs – реалізація функцій класу AddClassForm;
- AddClassForm.Designer.cs – інтерфейс форми додавання класу;
- AddClassForm.resx – файл ресурсів класу;
- BellsForm.cs – реалізація функцій класу BellsForm;
- BellsForm.Designer.cs – інтерфейс форми редагування дзвоників;
- BellsForm.resx – файл ресурсів класу;
- CreateUserForm.cs – реалізація функцій класу CreateUserForm;
- CreateUserForm.Designer.cs – інтерфейс форми створення користувачів;

- CreateUserForm.resx – файл ресурсів класу;
- DiaryDay.cs – опис і реалізація класу DiaryDay;
- Form1.cs – реалізація функцій класу Form1;
- Form1.Designer.cs – інтерфейс головної форми;
- Form1.resx – файл ресурсів класу;
- LoginForm.cs – реалізація функцій класу LoginForm;

- LoginForm.Designer.cs – інтерфейс форми авторизації;
- LoginForm.resx – файл ресурсів класу;
- MonthMarksForm.cs – реалізація функцій класу MonthMarksForm;
- MonthMarksForm.Designer.cs – інтерфейс форми успішності;
- MonthMarksForm.resx – файл ресурсів класу;
- TabelForm.cs – реалізація функцій класу TabelForm;
- TabelForm.Designer.cs – інтерфейс форми підсумкових оцінок;
- TabelForm.resx – файл ресурсів класу;
- RatingForm.cs – реалізація функцій класу RatingForm;
- RatingForm.Designer.cs – інтерфейс форми успішності з предметів;
- RatingForm.resx – файл ресурсів класу.

Також присутні автоматично згенеровані файли середовищем розробки.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Файл проекту програмного забезпечення для роботи електронного щоденника здобувача освіти знаходиться поряд з текою проекту і називається ElDiary.sln. Після відкриття файлу проекту програмного забезпечення для роботи електронного щоденника здобувача освіти, середовище розробки автоматично завантажить необхідні файли та налаштування з теки проекту.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для роботи електронного щоденника здобувача освіти є інформація про учнів, оцінки, домашнє завдання і розклад, які задають відповідними полями на формах.

Вихідними даними програмного забезпечення для роботи електронного щоденника здобувача освіти є повідомлення про успішне збереження даних,

інформація про учнів, статистика успішності, розклад, домашнє завдання та підсумкові оцінки.

4.3.3 Повідомлення

В процесі роботи з програмою для роботи електронного щоденника здобувача освіти користувач може отримати ряд повідомлень. При некоректному введенні даних чи при успішному виконання операції перед користувачем буде з'являтися відповідне повідомлення. Приклад повідомлення зображено на рисунку 4.1.

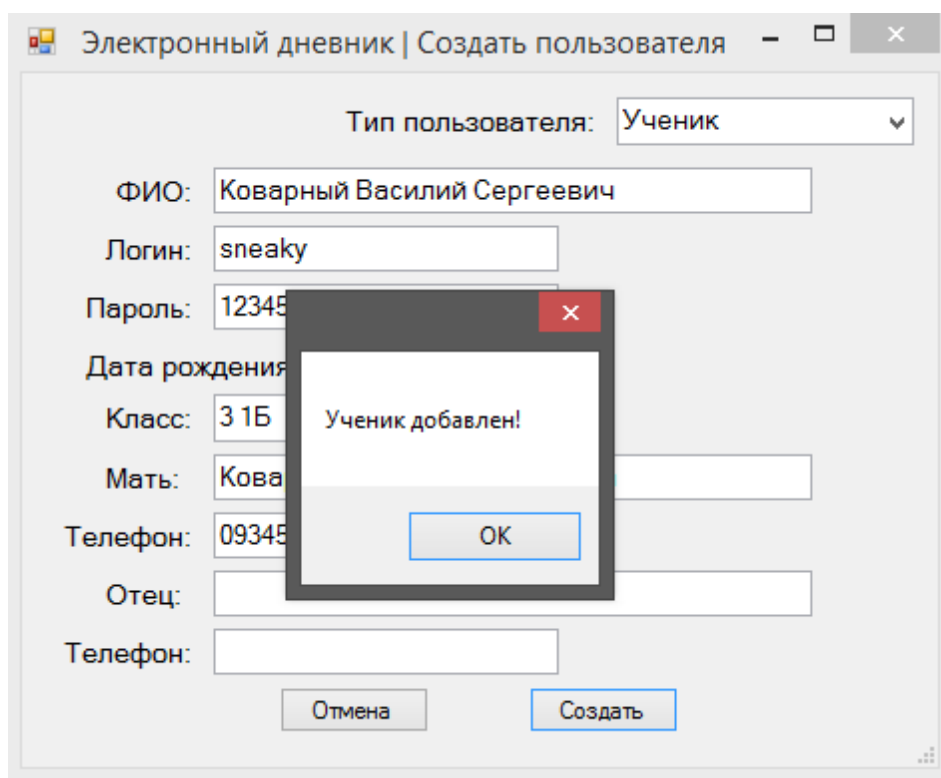


Рисунок 4.1 – Повідомлення користувачу

При роботі з програмою для роботи електронного щоденника здобувача освіти можуть виникнути різні повідомлення про помилку. Приклад такого повідомлення зображено на рисунку 4.2.

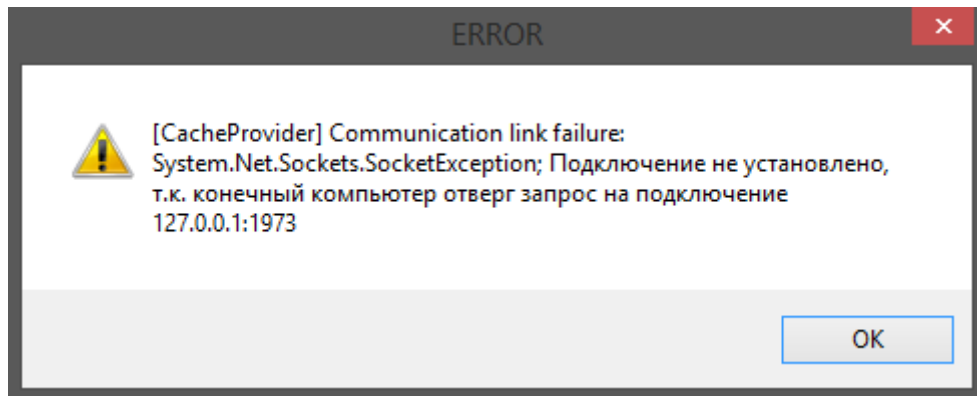


Рисунок 4.2 – Помилка підключення до БД

4.4 Виконання програмного забезпечення для роботи електронного щоденника здобувача освіти

Після того, як було запущено програмне забезпечення для роботи електронного щоденника здобувача освіти, перед користувачем постає форма авторизації (рис. 4.3) з текстовими полями для логіна та паролю і кнопкою «Войти».

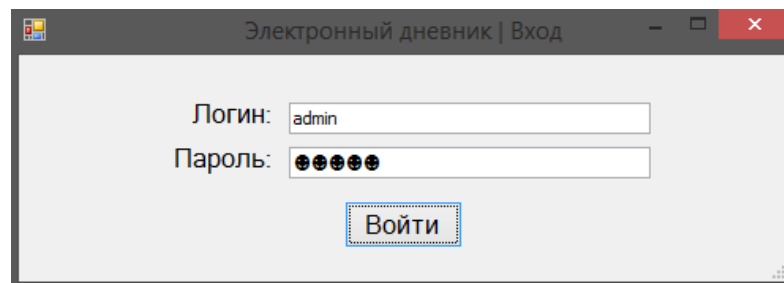


Рисунок 4.3 – Форма авторизації

Після натиснення кнопки «Войти» на формі авторизації та після успішної перевірки введених даних, відкривається головне вікно програми, вигляд якого регулюється правами доступу користувача. На рис. 4.4 зображено головне вікно програми з правами доступу адміністратора. На правах доступу учителя функціонал, зв'язаний з додаванням нових користувачів, класів, редагування розкладу стає недоступний. На правах доступу учня, можливо лише переглядати свої дані.

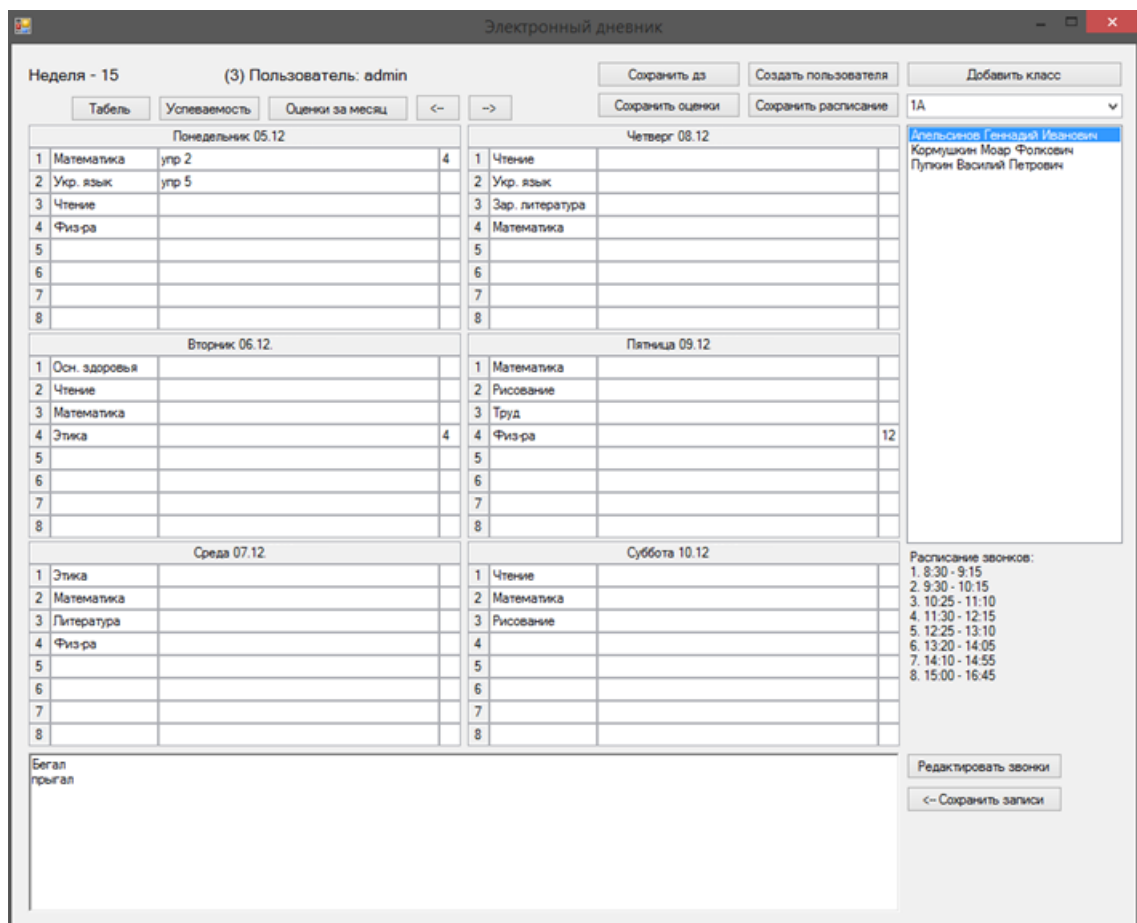


Рисунок 4.4 – Головне вікно програми очима адміністратора

З правами доступу адміністратора доступні усі функції програми. На головному вікні знаходяться кнопки «Добавить класс», «Создать пользователя», «Сохранить расписание», «Сохранить дз», «Сохранить оценки», «Оценки за месяц», «Табель», «Успеваемость», «Редактировать звонки», «Сохранить записи».

При натисненні на кнопки «Сохранить расписание», «Сохранить дз», «Сохранить оценки» та «Сохранить записи» вчителя можуть зберігати відповідні дані.

Після натиснення кнопки «Создать пользователя» відкривається вікно створення користувача (рис. 4.5), у якому можна обрати тип користувача, ввести відповідні дані та зберегти.

Электронный дневник | Создать пользователя

Тип пользователя: Ученик

ФИО: Кормушкин Моар Фолкович

Логин: meow

Пароль: 123456

Дата рождения: 20 апреля 2010 г.

Класс: 1A

Мать: Кормушкина Асум Алкеровна

Телефон: 063123456

Отец: Кормушкин Фолк Корнеевич

Телефон: 098452321

Отмена Создать

Рисунок 4.5 – Вікно створення користувача

При заповненні відповідних форм, після натиснення кнопки «Создать», відповідний користувач буде створений і користувач повернеться до головної форми.

Після натиснення кнопки «Добавить класс» відкриється форма створення класу (рис. 4.6), у якій користувачу потрібно буде ввести номер та букву класу і запропоновано список вчителів для вибору класного керівника.

Электронный дневник | Добавить класс

Введите номер и букву класса:

2A

Выберите классного роководителя:

З Пилова Татьяна Михай. ▾

Отмена Добавить

Рисунок 4.6 – Вікно створення класу

Після натиснення кнопки «Добавить», при успішному створенні класу користувачу буде показано повідомлення про це. Після цього користувач повернеться до головної форми.

Після натиснення кнопки головної форми «Редактировать звонки» відкриється форма для встановлення розкладу дзвінків (рис. 4.7).

Рисунок 4.7 – Вікно редагування дзвоників

Після натиснення кнопки «Сохранить» буде збережений вказаний час и користувач повернеться до головної форми.

Після натиснення кнопки головної форми «Оценки за месяц» буде показане вікно з оцінками обраного учня за місяць, графік його успішності та успішності класу (рис.4.8).

Після натиснення кнопки головної форми «Успеваемость» буде відображено вікно з успішність учня по конкретному предмету з початку навчального року (рис.4.10).

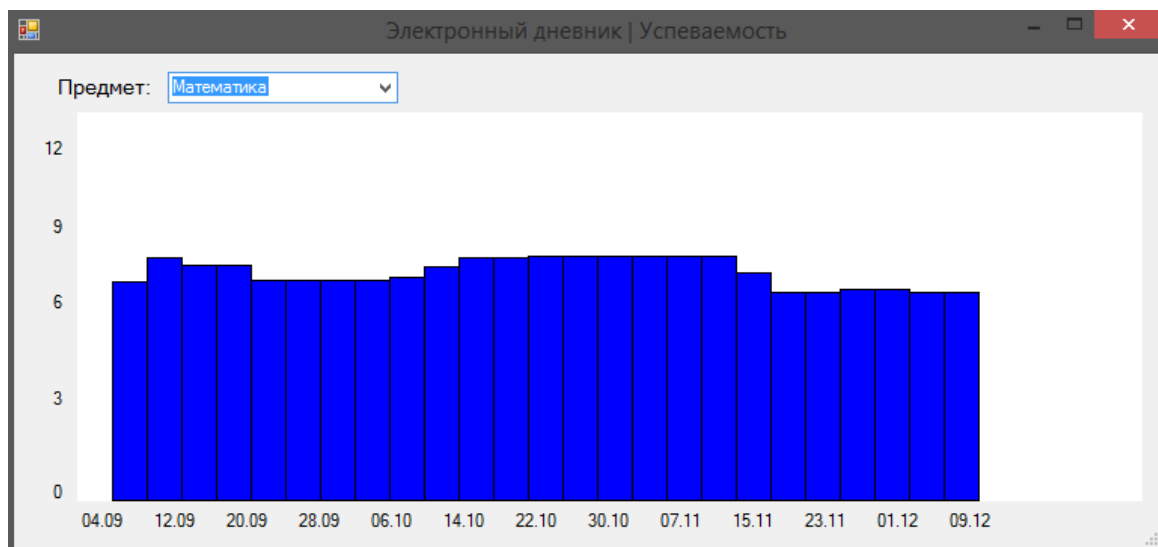


Рисунок 4.10 – Успішність учня з конкретного предмету

Результати тестування програмного забезпечення для роботи електронного щоденника здобувача освіти показали, що інтерфейс програми є інтуїтивно зрозумілим та досить зручним для користувача, програма може використовуватися за призначенням.

4.5 Висновки за розділом 4

Описано програмне забезпечення для роботи електронного щоденника здобувача освіти. Виконано тестування розробленого програмного забезпечення для роботи електронного щоденника здобувача освіти. Результати тестування програмного забезпечення для роботи електронного щоденника здобувача освіти підтвердили його працездатність.

Інтерфейс програмного забезпечення для роботи електронного щоденника здобувача освіти є інтуїтивно зрозумілим та досить зручним для користувача.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки програмного забезпечення для освітньої діяльності.

Проаналізовано загальні поняття про програмні засоби для підтримки освітньої діяльності. Визначено, що освітнє програмне забезпечення – це масовий, всеохоплюючий термін, який використовується для позначення будь-якого програмного забезпечення, призначеного для використання в освітній галузі. Проаналізовано тенденції, пов'язані з програмним забезпеченням для освіти, а також потенційні проблеми використання такого програмного забезпечення. Описано категорії освітнього програмного забезпечення.

Визначено, що у наш час існує досить багато видів програмного забезпечення, що використовується у навчальних цілях. Важливим є програмне забезпечення для надання інформації про успішність здобувачів освіти, контролю розкладу уроків, оцінок та домашнього завдання учнів у вигляді електронного щоденника.

Виконано дослідження програмних засобів для підтримки освітнього процесу. Визначено, що не всі програмні засоби надають можливість роботи з ними безкоштовно, що для деяких навчальних закладів є досить суттєвим недоліком. Також недоліком існуючого програмного забезпечення є те, що воно не дозволяє доступ іншим користувачам, зокрема батькам учнів, до отримання інформації про результати навчання, що дещо ускладнює його застосування на практиці. Отже доцільною є розробка програмного забезпечення для надання інформації про успішність здобувачів освіти, контролю розкладу уроків, оцінок та домашнього завдання учнів у вигляді електронного щоденника.

Для створення програмного забезпечення для роботи електронного щоденника здобувача освіти було обрано мову програмування C#, яка є

об'єктно-орієнтованою мовою програмування та є досить зручною для створення програмного забезпечення та вебзастосунків.

Оскільки в якості мови програмування було обрано C#, то для створення програмного забезпечення обрано середовище розробки Microsoft Visual Studio, що включає в себе набір різних засобів, зокрема, інтегроване середовище розробки програмного забезпечення та велику кількість інших інструментів, які дозволяють розробляти як консольні, так і графічні програми, включаючи підтримку технології вебсайтів, вебзастосунків, вебсервісів для різних операційних систем.

Для реалізації програмного забезпечення для роботи електронного щоденника здобувача освіти обрано систему керування базами даних InterSystems Cache.

Визначено, що у розробленому програмному забезпеченні електронного щоденника здобувача освіти основними структурними компонентами є: модуль, що реалізує основний функціонал програми (програмна частина), модуль графічного інтерфейсу, що забезпечує взаємодію користувача з програмою, база даних про здобувачів освіти та інших учасників освітнього процесу.

Запропоновано схему взаємодії основних модулів програмного забезпечення електронного щоденника здобувача освіти. Користувач програмного забезпечення електронного щоденника здобувача освіти використовує графічний інтерфейс для перегляду та редагування даних. Модуль графічного інтерфейсу, використовуючи програмну частину та логіку передає до графічного інтерфейсу потрібні дані з бази даних здобувачів освіти.

Описано основні розроблені структури даних (класи) програмного забезпечення: DiaryDay, Form1, LoginForm, CreateUserForm, AddClassForm, BellsForm, MonthMarksForm, TableForm, RatingForm.

Розроблено схему бази даних програмного забезпечення для роботи електронного щоденника здобувача освіти. Відповідно до розробленої схеми

даних основними сутностями (таблицями) розробленої бази даних є: учні (Students), вчителі (Teachers), заняття (Classes), оцінки (Marks), розклад (Schedule), домашні завдання (DZ) та інші.

Виконано тестування розробленого програмного забезпечення для роботи електронного щоденника здобувача освіти. Результати тестування програмного забезпечення для роботи електронного щоденника здобувача освіти підтвердили його працездатність.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. 15 Educational Software Examples 2022 [Electronic resource]. – Access mode : <https://rigorousthemes.com/blog/educational-software-examples/>.
2. Educational software [Electronic resource]. – Access mode : https://en.wikipedia.org/wiki/Educational_software.
3. Best Education Software [Electronic resource]. – Access mode : <https://www.g2.com/categories/education>.
4. Free Software for Education [Electronic resource]. – Access mode : <https://www.gnu.org/software/free-software-for-education.html>.
5. Top 10 Educational Software Must-haves for K–12 Schools [Electronic resource]. – Access mode : <https://schoolbox.com.au/blog/top-10-educational-software-must-haves-for-k-12-schools/>.
6. Canvas LMS [Electronic resource]. – Access mode : <https://www.instructure.com/>.
7. Google Classroom [Electronic resource]. – Access mode : <https://classroom.google.com>.
8. ProProfs [Electronic resource]. – Access mode : <https://www.proprofs.com/training/lms-software/>.
9. Schoology [Electronic resource]. – Access mode : <https://www.schoology.com/>.
10. TalentLMS [Electronic resource]. – Access mode : <https://www.talentlms.com/>.
11. Litmos [Electronic resource]. – Access mode : <https://www.litmos.com/>.
12. Wisenet [Electronic resource]. – Access mode : <https://www.wisenetlife.com/eu/>.
13. Csharp Programlama [Electronic resource]. – Access mode : <https://www.oguzhantas.com/csharp/112-csharp-programlama-dili-ile-metodlar-bolum-1.html>.

14. C Sharp Programlama Dili [Electronic resource]. – Access mode : https://tr.wikibooks.org/wiki/C_Sharp_Programlama_Dili/C_Sharp_hakk%C4%B1nda_temel_bilgiler.

15. C# Nedir [Electronic resource]. – Access mode : <https://www.webtekno.com/c-sharp-nedir-c-sharp-programlama-dili-h80441.html>.

16. What is Visual Studio? [Electronic resource]. – Access mode : <https://www.incredibuild.com/integrations/visual-studio>.

17. Caché для анализа неструктурированных данных [Электронный ресурс]. – Режим доступа : <https://www.intersystems.com/ua/products/cache/>.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення для роботи електронного щоденника здобувача освіти може використовуватися для надання інформації про успішність здобувачів освіти, контролю розкладу уроків, оцінок та домашнього завдання.

А.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Програмне забезпечення для роботи електронного щоденника здобувача освіти», затверджене наказом Національного університету «Запорізька політехніка» № 102 від 27 квітня 2022 р.

А.2 Призначення розробки

Програмний продукт для роботи електронного щоденника здобувача освіти призначений для надання інформації про успішність здобувачів освіти.

А.3 Основні вимоги до програми, що розробляється

А.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для роботи електронного щоденника здобувача освіти повинно забезпечити виконання таких функцій:

- реєстрація учнів, вчителів за допомогою адміністратора;
- редагування списку класів;
- збереження розкладу, домашніх завдань, оцінок та поміток для учнів;
- збереження розкладу дзвінків;
- відображення успішності учнів;
- підрахунок та відображення підсумкових оцінок.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс програми для роботи електронного щоденника здобувача освіти повинен бути зручним для користувачів. Взаємодія користувача з програмою для роботи електронного щоденника здобувача освіти можлива за допомогою користувальницьких форм.

А.3.3 Вимоги до надійності

Програмне забезпечення для роботи електронного щоденника здобувача освіти повинно забезпечити оброблення проблемних ситуацій та видавати відповідні повідомлення.

А.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для роботи електронного щоденника здобувача освіти необхідна наявність комп'ютеру з параметрами, не гіршими від тих, які вказано у наступному пункті.

А.3.5 Вимоги до складу та параметрів технічних засобів

Для експлуатації програмного забезпечення роботи електронного щоденника здобувача освіти необхідно таке апаратне забезпечення:

- не менше 200 Мб вільного місця на жорсткому диску;
- процесор з тактовою частотою не менше 1,5 ГГц;
- 512Мб оперативної пам'яті;
- наявність пристроїв вводу та виводу.

A.3.6 Вимоги до маркування і пакування

Програма для роботи електронного щоденника здобувача освіти може бути записана на будь-якому носії інформації. На пакуванні повинна бути назва програми – «Програмне забезпечення для роботи електронного щоденника здобувача освіти».

A.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для роботи електронного щоденника здобувача освіти є інформація про учнів, оцінки, домашнє завдання і розклад, які задають відповідними полями на формах.

Вихідними даними програмного забезпечення для роботи електронного щоденника здобувача освіти є повідомлення про успішне збереження даних, інформація про учнів, статистика успішності, розклад, домашнє завдання та підсумкові оцінки.

ДОДАТОК Б
Опис програми

Б.1 Загальні відомості

Програмне забезпечення для роботи електронного щоденника здобувача освіти розроблено за допомогою мови програмування C# та середовища розробки Microsoft Visual Studio. Для реалізації програмного забезпечення для роботи електронного щоденника здобувача освіти обрано систему керування базами даних InterSystems Cache.

Б.2 Функціональне призначення

Програма призначена для забезпечення роботи електронного щоденника здобувача освіти.

Програма роботи електронного щоденника здобувача освіти забезпечує виконання таких функцій:

- реєстрація учнів, вчителів за допомогою адміністратора;
- редагування списку класів;
- збереження розкладу, домашніх завдань, оцінок та поміток для учнів;
- збереження розкладу дзвінків;
- відображення успішності учнів;
- підрахунок та відображення підсумкових оцінок.

Б.3 Опис логічної структури

У розробленому програмному забезпеченні електронного щоденника здобувача освіти основними структурними компонентами є:

- модуль, що реалізує основний функціонал програми (програмна частина);
- модуль графічного інтерфейсу, що забезпечує взаємодію користувача з програмою;

– база даних про здобувачів освіти та інших учасників освітнього процесу.

Користувач програмного забезпечення електронного щоденника здобувача освіти використовує графічний інтерфейс для перегляду та редагування даних. Модуль графічного інтерфейсу, використовуючи програмну частину та логіку передає до графічного інтерфейсу потрібні дані з бази даних здобувачів освіти.

ПП включає у себе 25 компонентів, що містить класи, їх реалізацію, файли проекту:

– ElDiary.csproj – містить дані про використані модулі, перелік файлів класів та їх реалізації, посилання на файл ресурсів та загальну інформацію про ПП;

– App.config – файл налаштувань проекту додатка;

– Program.cs – файл для опису запуску додатка;

– AddClassForm.cs – реалізація функцій класу AddClassForm;

– AddClassForm.Designer.cs – інтерфейс форми додавання класу;

– AddClassForm.resx – файл ресурсів класу;

– BellsForm.cs – реалізація функцій класу BellsForm;

– BellsForm.Designer.cs – інтерфейс форми редагування дзвоників;

– BellsForm.resx – файл ресурсів класу;

– CreateUserForm.cs – реалізація функцій класу CreateUserForm;

– CreateUserForm.Designer.cs – інтерфейс форми створення користувачів;

– CreateUserForm.resx – файл ресурсів класу;

– DiaryDay.cs – опис і реалізація класу DiaryDay;

– Form1.cs – реалізація функцій класу Form1;

– Form1.Designer.cs – інтерфейс головної форми;

– Form1.resx – файл ресурсів класу;

– LoginForm.cs – реалізація функцій класу LoginForm;

– LoginForm.Designer.cs – інтерфейс форми авторизації;

- LoginForm.resx – файл ресурсів класу;
- MonthMarksForm.cs – реалізація функцій класу MonthMarksForm;
- MonthMarksForm.Designer.cs – інтерфейс форми успішності;
- MonthMarksForm.resx – файл ресурсів класу;
- TabelForm.cs – реалізація функцій класу TabelForm;
- TabelForm.Designer.cs – інтерфейс форми підсумкових оцінок;
- TabelForm.resx – файл ресурсів класу;
- RatingForm.cs – реалізація функцій класу RatingForm;
- RatingForm.Designer.cs – інтерфейс форми успішності з предметів;
- RatingForm.resx – файл ресурсів класу.

Також присутні автоматично згенеровані файли середовищем розробки.

Б.4 Використані технічні засоби

Для експлуатації програмного забезпечення роботи електронного щоденника здобувача освіти необхідно таке апаратне забезпечення:

- не менше 200 Мб вільного місця на жорсткому диску;
- процесор з тактовою частотою не менше 1,5 ГГц;
- 512Мб оперативної пам'яті;
- наявність пристроїв вводу та виводу.

Б.5 Виклик і завантаження

Файл проекту програмного забезпечення для роботи електронного щоденника здобувача освіти знаходиться поряд з текою проекту і називається ElDiary.sln. Після відкриття файлу проекту програмного забезпечення для роботи електронного щоденника здобувача освіти, середовище розробки автоматично завантажить необхідні файли та налаштування з теки проекту.

Б.6 Вхідні та вихідні дані

Вхідними даними до програмного забезпечення для роботи електронного щоденника здобувача освіти є інформація про учнів, оцінки, домашнє завдання і розклад, які задають відповідними полями на формах.

Вихідними даними програмного забезпечення для роботи електронного щоденника здобувача освіти є повідомлення про успішне збереження даних, інформація про учнів, статистика успішності, розклад, домашнє завдання та підсумкові оцінки.

ДОДАТОК В
Текст програми

```

//Текст файла Program.cs:
using InterSystems.Data.CacheClient;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace CourseWork01
{
    static class Program
    {
        [STAThread]
        static void Main()
        {
            try
            {
                Application.EnableVisualStyles();
                Application.SetCompatibleTextRenderingDefault(false);
                Application.Run(new Form1());
            }
            catch (CacheException ex) { MessageBox.Show(ex.Message,
"ERROR", MessageBoxButtons.OK, MessageBoxIcon.Warning); }
            catch (Exception ex) { MessageBox.Show(ex.Message, "ERROR",
MessageBoxButtons.OK, MessageBoxIcon.Warning); }
        }
    }
}

```

```

// Текст файла DiaryDay.cs:
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace CourseWork01
{
    public class DiaryDay
    {
        public TableLayoutPanel dayPanel;
        public string dayName;
        public DateTime date;
        public DiaryDay(TableLayoutPanel tp, string n)
        {
            dayPanel = tp; dayName = n;
        }
        public Control title() { return
dayPanel.GetControlFromPosition(0, 0); }
    }
}

```

```

// Текст файла Form1.cs:
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;

```

```

using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using InterSystems.Data.CacheClient;

namespace CourseWork01
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            loadClasses();
        }
        int access; string uname, classn, fio;
        int weeksC = 0;
        List<DiaryDay> days = new List<DiaryDay>(6);
        private void Form1_Load(object sender, EventArgs e)
        {
            LoginForm lf = new LoginForm();
            lf.ShowDialog(this);
            access = lf.access;
            uname = lf.uname;
            classn = lf.classes;
            fio = lf.fio;
            label1.Text = "(" + access.ToString() + ") Пользователь: " +
uname;

            days.Add(new DiaryDay(mondayTable, "Понедельник"));
            days.Add(new DiaryDay(tuesdayTable, "Вторник"));
            days.Add(new DiaryDay(wednesdayTable, "Среда"));
            days.Add(new DiaryDay(thursdayTable, "Четверг"));
            days.Add(new DiaryDay(fridayTable, "Пятница"));
            days.Add(new DiaryDay(saturdayTable, "Суббота"));
            updateDays();
            loadBells();
            if(DateTime.Today.Month<9) weekLabel.Text = "Неделя - " +
((DateTime.Today - DateTime.Parse("01.09." + (DateTime.Today.Year-
1).ToString()))).Days / 7 + 2).ToString();
            else weekLabel.Text = "Неделя - " + ((DateTime.Today -
DateTime.Parse("01.09." + DateTime.Today.Year.ToString()))).Days / 7 +
2).ToString();

            if (access < 3)
            {
                addClassButton.Enabled = false;
                addClassButton.Visible = false;
                addUserButton.Enabled = false;
                addUserButton.Visible = false;
                saveButton.Enabled = false;
                saveButton.Visible = false;
                if (access < 2)
                {
                    button1.Enabled = false;
                    button1.Visible = false;
                    button2.Enabled = false;
                    button2.Visible = false;
                    classCombo.Text = classn;
                    classCombo.Enabled = false;
                    studentsList.SelectedItem = fio;
                    studentsList.Enabled = false;
                    saturdayField.ReadOnly = true;
                    updateStudentsList();
                    clearSchedule();
                    loadSchedule();
                    clearDZ();
                }
            }
        }
    }
}

```

```

        updateDZ();
        loadSaturdayField();
    }
    for (int i = 0; i < 6; i++)
    {
        for (int j = 1; j <= 8; j++)
        {
            TextBox tb =
            (TextBox)days[i].dayPanel.GetControlFromPosition(1, j);
            tb.ReadOnly = true;
            if (access < 2)
            {
                tb =
                (TextBox)days[i].dayPanel.GetControlFromPosition(2, j);
                tb.ReadOnly = true;
                tb =
                (TextBox)days[i].dayPanel.GetControlFromPosition(3, j);
                tb.ReadOnly = true;
            }
        }
    }
}

string getStudName()
{
    if (access < 2) return fio;
    if (studentsList.SelectedItem == null) return "";
    return studentsList.SelectedItem.ToString();
}

void loadClasses()
{
    classCombo.Items.Clear();
    CacheConnection con = new CacheConnection();
    con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";
    con.Open();
    string quer = "select * from Classes";
    CacheCommand com = new CacheCommand(quer, con);
    CacheDataReader r = com.ExecuteReader();
    while (r.Read())
    {
        classCombo.Items.Add(r[r.GetOrdinal("class")]);
    }
    r.Close();
    com.Dispose();
    con.Close();
}

void updateDays()
{
    for (int i = 0; i < 6; i++)
    {
        DateTime dt = DateTime.Today.Add(TimeSpan.FromDays(-1 *
(int)DateTime.Today.DayOfWeek + 1 + i + weeksC * 7));
        days[i].title().Text = days[i].dayName + " " +
dt.ToString("dd.MM.yyyy");
        days[i].date = dt;
    }
}

private void nextButton_Click(object sender, EventArgs e)
{
    weeksC++;
}

```

```

        updateDays();
        if (DateTime.Today.Month < 9) weekLabel.Text = "Неделя - " +
        ((DateTime.Today - DateTime.Parse("01.09." + (DateTime.Today.Year -
        1).ToString())).Days / 7 + 2 + weeksC).ToString();
        else weekLabel.Text = "Неделя - " + ((DateTime.Today -
        DateTime.Parse("01.09." + DateTime.Today.Year.ToString())).Days / 7 + 2 +
        weeksC).ToString();
        clearDZ();
        updateDZ();
        clearMarks();
        loadMarks();
        loadSaturdayField();
    }

    private void prevButton_Click(object sender, EventArgs e)
    {
        weeksC--;
        updateDays();
        if (DateTime.Today.Month < 9) weekLabel.Text = "Неделя - " +
        ((DateTime.Today - DateTime.Parse("01.09." + (DateTime.Today.Year -
        1).ToString())).Days / 7 + 2 + weeksC).ToString();
        else weekLabel.Text = "Неделя - " + ((DateTime.Today -
        DateTime.Parse("01.09." + DateTime.Today.Year.ToString())).Days / 7 + 2 +
        weeksC).ToString();
        clearDZ();
        updateDZ();
        clearMarks();
        loadMarks();
        loadSaturdayField();
    }

    private void addClassButton_Click(object sender, EventArgs e)
    {
        AddClassForm acf = new AddClassForm();
        acf.ShowDialog(this);
        loadClasses();
    }

    private void addUserButton_Click(object sender, EventArgs e)
    {
        CreateUserForm cuf = new CreateUserForm();
        cuf.ShowDialog(this);
    }

    void updateStudentsList()
    {
        studentsList.Items.Clear();
        CacheConnection con = new CacheConnection();
        con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";
        con.Open();
        string quer = "select fio from Students where class in
(select id from Classes where class='" + classCombo.Text + "')";
        CacheCommand com = new CacheCommand(quer, con);
        CacheDataReader r = com.ExecuteReader();
        while (r.Read())
        {
            studentsList.Items.Add(r[r.GetOrdinal("fio")]);
        }
        studentsList.Sorted = true;
        r.Close();
        com.Dispose();
        con.Close();
    }

```

```

        string getClassId()
        {
            string cid = "";
            CacheConnection con = new CacheConnection();
            con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";
            con.Open();
            string quer = "select id from Classes where class='" +
classCombo.Text + "'";
            CacheCommand com = new CacheCommand(quer, con);
            CacheDataReader r = com.ExecuteReader();
            while (r.Read())
            {
                cid = r[r.GetOrdinal("id")].ToString();
            }
            r.Close();
            com.Dispose();
            con.Close();
            return cid;
        }
        void loadSchedule()
        {
            string cid = getClassId();
            if (cid == "") return;
            CacheConnection con = new CacheConnection();
            con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";
            con.Open();
            for (int i = 0; i < 6; i++)
            {
                string quer = "select lesson, subj from Schedule where
class='" + cid + "' and s_day='" + days[i].dayName + "'";
                CacheCommand com = new CacheCommand(quer, con);
                CacheDataReader r = com.ExecuteReader();
                while (r.Read())
                {
                    int lesson =
Int32.Parse(r[r.GetOrdinal("lesson")].ToString());
                    days[i].dayPanel.GetControlFromPosition(1, lesson).Text
= r[r.GetOrdinal("subj")].ToString();
                }
                r.Close();
                com.Dispose();
            }
            con.Close();
        }

        void clearSchedule()
        {
            for (int i = 0; i < 6; i++)
            {
                for (int j = 1; j <= 8; j++)
                {
                    days[i].dayPanel.GetControlFromPosition(1, j).Text =
"";
                }
            }
        }

        private void saveButton_Click(object sender, EventArgs e)
        {
            if (classCombo.Text == "") { MessageBox.Show("Выберите
класс!"); return; }

```

```

        string cid = getClassId();
        CacheConnection con = new CacheConnection();
        con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";
        con.Open();
        for (int i = 0; i < 6; i++)
        {
            string quer = "delete from Schedule where s_day='" +
days[i].dayName + "' and class=" + cid;
            CacheCommand com = new CacheCommand(quer, con);
            com.ExecuteNonQuery();
            com.Dispose();
            for (int j = 1; j <= 8; j++)
            {
                if (days[i].dayPanel.GetControlFromPosition(1,
j).Text != "")
                {
                    quer = "insert into Schedule (s_day, lesson,
class, subj) values ('" + days[i].dayName + "', '" + j.ToString() + "', '" +
cid +
                    "', '" +
days[i].dayPanel.GetControlFromPosition(1, j).Text + "')";
                    com = new CacheCommand(quer, con);
                    com.ExecuteNonQuery();
                    com.Dispose();
                }
            }
        }
        con.Close();
    }

    void clearDZ()
    {
        for (int i = 0; i < 6; i++)
        {
            for (int j = 1; j <= 8; j++)
            {
                days[i].dayPanel.GetControlFromPosition(2, j).Text =
"";
            }
        }
    }

    void clearMarks()
    {
        for (int i = 0; i < 6; i++)
        {
            for (int j = 1; j <= 8; j++)
            {
                days[i].dayPanel.GetControlFromPosition(3, j).Text =
"";
            }
        }
    }

    void updateDZ()
    {
        string cid = getClassId();
        if (cid == "") return;
        CacheConnection con = new CacheConnection();
        con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";
        con.Open();

```

```

        for (int i = 0; i < 6; i++)
        {
            string quer = "select subject, task from dz where
class='" + cid + "' and date_z=to_date('" + days[i].date.ToString("yyyy-MM-dd")
+ "','YYYY-MM-DD')";
            CacheCommand com = new CacheCommand(quer, con);
            CacheDataReader r = com.ExecuteReader();
            while (r.Read())
            {
                for (int j = 1; j <= 8; j++)
                {
                    if (days[i].dayPanel.GetControlFromPosition(1,
j).Text == r[r.GetOrdinal("subject")].ToString())
                    {
                        days[i].dayPanel.GetControlFromPosition(2,
j).Text = r[r.GetOrdinal("task")].ToString();
                        break;
                    }
                }
            }
            r.Close();
            com.Dispose();
        }
        con.Close();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        if (classCombo.Text == "") { MessageBox.Show("Выберите
класс!"); return; }
        string cid = getClassId();
        CacheConnection con = new CacheConnection();
        con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";
        con.Open();
        for (int i = 0; i < 6; i++)
        {
            string quer = "delete from DZ where date_z=to_date('" +
days[i].date.ToString("yyyy-MM-dd") + "','YYYY-MM-DD') and class=" + cid;
            CacheCommand com = new CacheCommand(quer, con);
            com.ExecuteNonQuery();
            com.Dispose();
            for (int j = 1; j <= 8; j++)
            {
                if (days[i].dayPanel.GetControlFromPosition(2,
j).Text != "")
                {
                    quer = "insert into DZ (date_z, subject, task,
class) values (to_date('" + days[i].date.ToString("yyyy-MM-dd") + "','YYYY-MM-
DD'), '" +
                        days[i].dayPanel.GetControlFromPosition(1,
j).Text + "',' '" + days[i].dayPanel.GetControlFromPosition(2, j).Text + "',' '"
+ cid + "')";
                    com = new CacheCommand(quer, con);
                    com.ExecuteNonQuery();
                    com.Dispose();
                }
            }
        }
        con.Close();
    }

    string getStudIdByName(string name)
    {

```

```

        string sid = "";
        string cis = getClassId();
        CacheConnection con = new CacheConnection();
        con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";
        con.Open();
        string quer = "select id from Students where fio='" + name +
        "' and class=" + cis;
        CacheCommand com = new CacheCommand(quer, con);
        CacheDataReader r = com.ExecuteReader();
        while (r.Read())
        {
            sid = r[r.GetOrdinal("id")].ToString();
        }
        r.Close();
        com.Dispose();
        con.Close();
        return sid;
    }

    private void button2_Click(object sender, EventArgs e)
    {
        if (classCombo.Text == "") { MessageBox.Show("Выберите
класс!"); return; }
        if (studentsList.SelectedIndex < 0 ||
studentsList.SelectedItem==null) { MessageBox.Show("Выберите
ученика!"); return; }

        string cid = getClassId();
        string sid = getStudIdByName(getStudName());
        CacheConnection con = new CacheConnection();
        con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";
        con.Open();
        for (int i = 0; i < 6; i++)
        {
            string quer = "delete from Marks where date_m=to_date('"
+ days[i].date.ToString("yyyy-MM-dd") + "','YYYY-MM-DD') and stud=" + sid;
            CacheCommand com = new CacheCommand(quer, con);
            com.ExecuteNonQuery();
            com.Dispose();
            for (int j = 1; j <= 8; j++)
            {
                if (days[i].dayPanel.GetControlFromPosition(3,
j).Text != "")
                {
                    quer = "insert into Marks (date_m, subject,
mark, stud) values (to_date('" + days[i].date.ToString("yyyy-MM-dd") +
 "','YYYY-MM-DD'), '" +
                        days[i].dayPanel.GetControlFromPosition(1,
j).Text + "',' '" + days[i].dayPanel.GetControlFromPosition(3, j).Text + "',' '"
+ sid + "')";

                    com = new CacheCommand(quer, con);
                    com.ExecuteNonQuery();
                    com.Dispose();
                }
            }
        }
        con.Close();
    }

    void loadMarks()
    {
        if (getStudName() == "") return;

```

```

        string sid = getStudIdByName(getStudName());
        if (sid == "") return;
        CacheConnection con = new CacheConnection();
        con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";
        con.Open();
        for (int i = 0; i < 6; i++)
        {
            string quer = "select mark, subject from Marks where
stud='" + sid + "' and date_m=to_date('" + days[i].date.ToString("yyyy-MM-dd")
+ "', 'YYYY-MM-DD')";
            CacheCommand com = new CacheCommand(quer, con);
            CacheDataReader r = com.ExecuteReader();
            while (r.Read())
            {
                for (int j = 1; j <= 8; j++)
                {
                    if (days[i].dayPanel.GetControlFromPosition(1,
j).Text == r[r.GetOrdinal("subject")].ToString())
                    {
                        days[i].dayPanel.GetControlFromPosition(3,
j).Text = r[r.GetOrdinal("mark")].ToString();
                        break;
                    }
                }
            }
            r.Close();
            com.Dispose();
        }
        con.Close();
    }

    private void studentsList_SelectedIndexChanged(object sender,
EventArgs e)
    {
        clearMarks();
        loadMarks();
        loadSaturdayField();
    }

    private void saveSaturdayButton_Click(object sender, EventArgs
e)
    {
        if (classCombo.Text == "") { MessageBox.Show("Выберите
класс!"); return; }
        if (studentsList.SelectedIndex < 0 ||
studentsList.SelectedItem == null) { MessageBox.Show("Выберите
ученика!"); return; }

        string cid = getClassId();
        string sid = getStudIdByName(getStudName());
        CacheConnection con = new CacheConnection();
        con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";
        con.Open();
        for (int i = 0; i < 1; i++)
        {
            string quer = "delete from TextField where
date_m=to_date('" + days[i].date.ToString("yyyy-MM-dd") + "', 'YYYY-MM-DD') and
stud=" + sid;

            CacheCommand com = new CacheCommand(quer, con);
            com.ExecuteNonQuery();
            com.Dispose();
            if (saturdayField.Text != "")
            {

```

```

        quer = "insert into TextField (date_m, text_f, stud)
values (to_date('" + days[i].date.ToString("yyyy-MM-dd") + "','YYYY-MM-DD'), '"
+
        saturdayField.Text + "',' ' + sid + "')";
        com = new CacheCommand(quer, con);
        com.ExecuteNonQuery();
        com.Dispose();
    }
}

void loadSaturdayField()
{
    saturdayField.Text = "";
    if (getStudName() == "") return;
    string sid = getStudIdByName(getStudName());
    if (sid == "") return;
    CacheConnection con = new CacheConnection();
    con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";
    con.Open();
    for (int i = 0; i < 1; i++)
    {
        string quer = "select text_f from TextField where
stud='" + sid + "' and date_m=to_date('" + days[i].date.ToString("yyyy-MM-dd")
+ "','YYYY-MM-DD')";
        CacheCommand com = new CacheCommand(quer, con);
        CacheDataReader r = com.ExecuteReader();
        while (r.Read())
        {
            saturdayField.Text =
r[r.GetOrdinal("text_f")].ToString();
        }
        r.Close();
        com.Dispose();
    }
    con.Close();
}

private void getMarksButton_Click(object sender, EventArgs e)
{
    if (classCombo.Text == "") { MessageBox.Show("Выберите
класс!"); return; }
    if (getStudName() == "") { MessageBox.Show("Выберите
ученика!"); return; }
    string sid = getStudIdByName(getStudName());
    MonthMarksForm mmf = new MonthMarksForm(sid, getClassId(),
days);
    mmf.ShowDialog();
}

private void editBells_Click(object sender, EventArgs e)
{
    BellsForm bf = new BellsForm();
    bf.ShowDialog(this);
    loadBells();
}

private void studentsList_DoubleClick(object sender, EventArgs e)
{
    if (access < 2 || studentsList.SelectedItem == null) return;
    string name;
    if ((name=getStudName()) == "") return;
    string cis = getClassId();
    string data = "Дата рождения: ";

```

```

        CacheConnection con = new CacheConnection();
        con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";
        con.Open();
        string quer = "select bdate, mother, father, m_tel, f_tel
from Students where fio='" + name + "' and class=" + cis;
        CacheCommand com = new CacheCommand(quer, con);
        CacheDataReader r = com.ExecuteReader();
        while (r.Read())
        {
            DateTime dt =
DateTime.Parse(r[r.GetOrdinal("bdate")].ToString());
            data += dt.ToString("dd.MM.yyyy") + "\nРодители:\n" +
r[r.GetOrdinal("mother")].ToString() + "\n" +
r[r.GetOrdinal("m_tel")].ToString();
            data += "\n" + r[r.GetOrdinal("father")].ToString() +
"\n" + r[r.GetOrdinal("f_tel")].ToString();
        }
        r.Close();
        com.Dispose();
        con.Close();
        if (data == "") return;
        MessageBox.Show(data);
    }

    private void marksAllButton_Click(object sender, EventArgs e)
    {
        if (classCombo.Text == "") { MessageBox.Show("Выберите
класс!"); return; }
        if (getStudName() == "") { MessageBox.Show("Выберите
ученика!"); return; }
        TabelForm tf = new TabelForm(getStudIdByName(getStudName()),
days);
        tf.ShowDialog(this);
    }

    private void ratingButton_Click(object sender, EventArgs e)
    {
        if (classCombo.Text == "") { MessageBox.Show("Выберите
класс!"); return; }
        if (getStudName() == "") { MessageBox.Show("Выберите
ученика!"); return; }
        RatingForm rf = new
RatingForm(getStudIdByName(getStudName()), days);
        rf.ShowDialog(this);
    }

    void loadBells()
    {
        string result = "Расписание звонков:\n";
        CacheConnection con = new CacheConnection();
        con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";
        con.Open();
        string quer = "select lesson, time_b from Bells";
        CacheCommand com = new CacheCommand(quer, con);
        CacheDataReader r = com.ExecuteReader();
        while (r.Read())
        {
            result += r[r.GetOrdinal("lesson")].ToString() + ". " +
r[r.GetOrdinal("time_b")].ToString() + "\n";
        }
        r.Close();
        com.Dispose();
    }

```

```

        con.Close();
        bellsLabel.Text = result;
    }

    private void classCombo_SelectedIndexChanged(object sender,
EventArgs e)
    {
        updateStudentsList();
        clearSchedule();
        loadSchedule();
        clearDZ();
        updateDZ();
        clearMarks();
        loadSaturdayField();
    }
}

// Текст файла LoginForm.cs:
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using InterSystems.Data.CacheClient;
using InterSystems.Globals;

namespace CourseWork01
{
    public partial class LoginForm : Form
    {
        public int access = 0;
        public string uname = "", classes="", fio="";
        bool logged = false;
        public LoginForm()
        {
            InitializeComponent();
        }

        private void loginButton_Click(object sender, EventArgs e)
        {
            CacheConnection con = new CacheConnection();
            con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";
            con.Open();
            string quer = "select uname, pass from Admins";
            CacheCommand com = new CacheCommand(quer, con);
            CacheDataReader r = com.ExecuteReader();
            while (r.Read())
            {
                if (r[r.GetOrdinal("uname")].ToString() == loginBox.Text
&& r[r.GetOrdinal("pass")].ToString() == passwordBox.Text)
                {
                    logged = true;
                    uname = r[r.GetOrdinal("uname")].ToString();
                    access = 3;
                    break;
                }
            }
            r.Close();

```

```

        com.Dispose();
        if (logged) { con.Close(); Close(); return; }
        quer = "select uname, pass from Teachers";
        com = new CacheCommand(quer, con);
        r = com.ExecuteReader();
        while (r.Read())
        {
            if (r[r.GetOrdinal("uname")].ToString() == loginBox.Text
                && r[r.GetOrdinal("pass")].ToString() == passwordBox.Text)
            {
                logged = true;
                uname = r[r.GetOrdinal("uname")].ToString();
                access = 2;
                break;
            }
        }
        r.Close();
        com.Dispose();
        if (logged) { con.Close(); Close(); return; }
        quer = "select Students.fio, Students.uname, Students.pass,
Classes.class from Students, Classes where Students.class=Classes.id";
        com = new CacheCommand(quer, con);
        r = com.ExecuteReader();
        while (r.Read())
        {
            if (r[r.GetOrdinal("uname")].ToString() == loginBox.Text
                && r[r.GetOrdinal("pass")].ToString() == passwordBox.Text)
            {
                logged = true;
                uname = r[r.GetOrdinal("uname")].ToString();
                classes = r[r.GetOrdinal("class")].ToString();
                fio = r[r.GetOrdinal("fio")].ToString();
                access = 1;
                break;
            }
        }
        r.Close();
        com.Dispose();
        if (logged) { con.Close(); Close(); return; }
        con.Close();
        if (logged) Close();
        else MessageBox.Show("Неверные данные!");
    }

    private void LoginForm_FormClosing(object sender,
FormClosingEventArgs e)
    {
        if (!logged) Application.Exit();
    }
}

```

```

// Текст файла CreateUserForm.cs:
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using InterSystems.Data.CacheClient;

```

```

namespace CourseWork01
{
    public partial class CreateUserForm : Form
    {
        public CreateUserForm()
        {
            InitializeComponent();
            loadClasses();
        }
        bool isstudent = true; bool added = false;
        private void comboBox1_SelectedIndexChanged(object sender,
EventArgs e)
        {
            if(isstudent==comboBox1.Text == "Ученик")
            {
                button1.Visible = false;
                button2.Visible = false;
                Size = new Size(475, 385);
                button1.Location = new Point(255, 306);
                button2.Location = new Point(129, 306);
                button1.Visible = true;
                button2.Visible = true;
                label4.Visible = true;
                label6.Visible = true;
                label7.Visible = true;
                label8.Visible = true;
                label9.Visible = true;
                label10.Visible = true;
                dateTimePicker1.Visible = true;
                textBox4.Visible = true;
                textBox5.Visible = true;
                textBox6.Visible = true;
                textBox7.Visible = true;
                comboBox2.Visible = true;
            }
            else
            {
                button1.Visible = false;
                button2.Visible = false;
                Size = new Size(475, 250);
                button1.Location = new Point(255, 164);
                button2.Location = new Point(129, 164);
                button1.Visible = true;
                button2.Visible = true;
                label4.Visible = false;
                label6.Visible = false;
                label7.Visible = false;
                label8.Visible = false;
                label9.Visible = false;
                label10.Visible = false;
                textBox4.Visible = false;
                textBox5.Visible = false;
                textBox6.Visible = false;
                textBox7.Visible = false;
                dateTimePicker1.Visible = false;
                comboBox2.Visible = false;
            }
        }

        void loadClasses()
        {
            CacheConnection con = new CacheConnection();
            con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";

```

```

        con.Open();
        string quer = "select * from Classes";
        CacheCommand com = new CacheCommand(quer, con);
        CacheDataReader r = com.ExecuteReader();
        while (r.Read())
        {
            comboBox2.Items.Add(r[r.GetOrdinal("class")]);
        }
        r.Close();
        com.Dispose();
        con.Close();
    }

    private void button2_Click(object sender, EventArgs e)
    {
        Close();
    }

    string getClassId(string name)
    {
        string res = "";
        CacheConnection con = new CacheConnection();
        con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";
        con.Open();
        string quer = "select * from Classes where
class='"+name+"'";
        CacheCommand com = new CacheCommand(quer, con);
        CacheDataReader r = com.ExecuteReader();
        while (r.Read())
        {
            res = r[r.GetOrdinal("id")].ToString();
        }
        r.Close();
        com.Dispose();
        con.Close();
        return res;
    }

    bool isUserExists(string uname)
    {
        bool isex = false;
        CacheConnection con = new CacheConnection();
        con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";
        con.Open();
        string quer = "select * from Admins";
        CacheCommand com = new CacheCommand(quer, con);
        CacheDataReader r = com.ExecuteReader();
        while (r.Read())
        {
            if (r[r.GetOrdinal("uname")].ToString() == uname) { isex
= true; break; }
        }
        r.Close();
        com.Dispose();
        if (isex) { con.Close(); return isex; }
        quer = "select * from Teachers";
        com = new CacheCommand(quer, con);
        r = com.ExecuteReader();
        while (r.Read())
        {
            if (r[r.GetOrdinal("uname")].ToString() == uname) { isex
= true; break; }
        }
    }

```

```

    }
    com.Dispose();
    if (isex) { con.Close(); return isex; }
    quer = "select * from Students";
    com = new CacheCommand(quer, con);
    r = com.ExecuteReader();
    while (r.Read())
    {
        if (r[r.GetOrdinal("uname")].ToString() == uname) { isex
= true; break; }
    }
    com.Dispose();
    if (isex) { con.Close(); return isex; }
    con.Close(); return isex;
}

private void button1_Click(object sender, EventArgs e)
{
    if ((textBox1.Text == "" || textBox2.Text == "") &&
!isstudent) { MessageBox.Show("Неверно заполнены поля!"); return; }
    if ((textBox1.Text == "" || textBox2.Text == "" ||
textBox3.Text == "" || comboBox2.Text == "") && isstudent)
    { MessageBox.Show("Неверно заполнены поля!"); return; }
    if (isUserExists(textBox2.Text))
    {
        MessageBox.Show("Пользователь с таким логином существует!"); return; }
    CacheConnection con = new CacheConnection();
    con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";
    con.Open();
    if(!isstudent)
    {
        string quer = "insert into Teachers (fio, uname, pass)
values ('"+textBox1.Text+"', '"+textBox2.Text+"', '"+textBox3.Text+"')";
        CacheCommand com = new CacheCommand(quer, con);
        if (com.ExecuteNonQuery() > 0)
        {
            MessageBox.Show("Учитель добавлен!"); added = true; }
        com.Dispose();
    }
    else
    {
        string id = getClassId(comboBox2.Text);
        string quer = "insert into Students (fio, class, uname,
pass, bdate, mother, m_tel, father, f_tel) values ('" + textBox1.Text + "', '"
+ id + "', '" + textBox2.Text + "', '" + textBox3.Text + "', to_date('" +
dateTimePicker1.Value.ToString("yyyy-MM-dd") + "', 'yyyy-mm-dd'), '" +
textBox4.Text + "', '" + textBox5.Text + "', '" + textBox7.Text + "', '" +
textBox6.Text+"')";
        CacheCommand com = new CacheCommand(quer, con);
        if (com.ExecuteNonQuery() > 0) { MessageBox.Show("Ученик
добавлен!"); added = true; }
        com.Dispose();
    }
    con.Close();
    if (added) Close();
}
}
}

```

```

// Текст файлу AddClassForm.cs:
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;

```

```

using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using InterSystems.Data.CacheClient;
using InterSystems.Globals;
namespace CourseWork01
{
    public partial class AddClassForm : Form
    {
        public AddClassForm()
        {
            InitializeComponent();
            updateCombo();
        }

        void updateCombo()
        {
            CacheConnection con = new CacheConnection();
            con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";
            con.Open();
            string quer = "select * from Teachers";
            CacheCommand com = new CacheCommand(quer, con);
            CacheDataReader r = com.ExecuteReader();
            while (r.Read())
            {
                comboBox1.Items.Add(r[r.GetOrdinal("id")] + " "
+r[r.GetOrdinal("fio")]);
            }
            r.Close();
            com.Dispose();
            con.Close();
        }

        public bool added = false;

        private void button1_Click(object sender, EventArgs e)
        {
            if (textBox1.Text == "" || comboBox1.Text == "") {
                MessageBox.Show("Вы ввели неверные данные!"); return; }
            bool ex = false;
            CacheConnection con = new CacheConnection();
            con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";
            con.Open();
            string quer = "select * from Classes";
            CacheCommand com = new CacheCommand(quer, con);
            CacheDataReader r = com.ExecuteReader();
            while (r.Read())
            {
                ex =
                textBox1.Text
                ==
                r[r.GetOrdinal("class")].ToString();
                if (ex) break;
            }
            r.Close();
            com.Dispose();
            if (ex) MessageBox.Show("Такой класс уже добавлен!");
            else
            {
                string id = comboBox1.Text.Split(' ')[0];
                quer = "insert into Classes (class, master) values ('" +
textBox1.Text + "', '" + id + "')";
                com = new CacheCommand(quer, con);
            }
        }
    }
}

```

```

        if (com.ExecuteNonQuery() > 0) { MessageBox.Show("Класс
добавлен!"); added = true; }
        com.Dispose();
    }
    con.Close();
    if (added) Close();
}

private void button2_Click(object sender, EventArgs e)
{
    Close();
}
}

// Текст файла BellsForm.cs:
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using InterSystems.Data.CacheClient;

namespace CourseWork01
{
    public partial class BellsForm : Form
    {
        List<TextBox> tbl1 = new List<TextBox>();
        List<TextBox> tbl2 = new List<TextBox>();
        public BellsForm()
        {
            InitializeComponent();
            tbl1.Add(textBox1);
            tbl1.Add(textBox3);
            tbl1.Add(textBox5);
            tbl1.Add(textBox7);
            tbl1.Add(textBox9);
            tbl1.Add(textBox11);
            tbl1.Add(textBox13);
            tbl1.Add(textBox15);
            tbl2.Add(textBox2);
            tbl2.Add(textBox4);
            tbl2.Add(textBox6);
            tbl2.Add(textBox8);
            tbl2.Add(textBox10);
            tbl2.Add(textBox12);
            tbl2.Add(textBox14);
            tbl2.Add(textBox16);
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Close();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            CacheConnection con = new CacheConnection();

```

```

        con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";
        con.Open();
        string quer = "delete from Bells";
        CacheCommand com = new CacheCommand(quer, con);
        com.ExecuteNonQuery();
        com.Dispose();
        int t = 0;
        for (int i=0; i<8; i++)
        {
            quer = "insert into Bells (lesson, time_b) values ('" +
(i+1) + "', '" + tbl1[i].Text+" - "+tbl2[i].Text+"')";
            com = new CacheCommand(quer, con);
            t=com.ExecuteNonQuery();
            com.Dispose();
        }
        con.Close();
        if (t > 0) { MessageBox.Show("Звонки сохранены!"); Close();
    }
}
}
}

```

```

// Текст файла MonthMarksForm.cs:
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using InterSystems.Data.CacheClient;

```

```

namespace CourseWork01
{
    public partial class MonthMarksForm : Form
    {
        string sid, cl;
        List<DiaryDay> days = new List<DiaryDay>(6);
        int step, height;
        Bitmap gr1, gr2;
        Graphics g1, g2;
        class SubjData
        {
            public string subject;
            public string marks;
            public SubjData() { }
            public void addMark(string mk)
            {
                marks += mk + " ";
            }
        }
        public MonthMarksForm(string studId, string cla, List<DiaryDay>
d)
        {
            sid = studId;
            days = d;
            InitializeComponent();
            datesLabel.Text = "";
            cl = cla;
            step = pictureBox1.Width / 30;

```

```

        height = pictureBox1.Height / 13;
        gr1 = new Bitmap(pictureBox1.Width, pictureBox1.Height);
        gr2 = new Bitmap(pictureBox2.Width, pictureBox2.Height);
        g1 = Graphics.FromImage(gr1); g2 = Graphics.FromImage(gr2);
        g1.FillRectangle(Brushes.White, 0, 0, gr1.Width,
gr1.Height);
        g2.FillRectangle(Brushes.White, 0, 0, gr2.Width,
gr2.Height);

        loadData();
        updatePictures();
    }

    void updatePictures()
    {
        pictureBox1.Image = gr1;
        pictureBox2.Image = gr2;
    }

    List<string> getStudentsId(string clas)
    {
        List<string> stdnts = new List<string>();
        CacheConnection con = new CacheConnection();
        con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";
        con.Open();
        string quer = "select id from Students where class='" + clas
+ "'";

        CacheCommand com = new CacheCommand(quer, con);
        CacheDataReader r = com.ExecuteReader();
        while(r.Read())
        {
            stdnts.Add(r[r.GetOrdinal("id")].ToString());
        }
        r.Close();
        com.Dispose();
        con.Close();
        return stdnts;
    }

    void loadClassData(List<SubjData> sd)
    {
        List<string> st = getStudentsId(cl);
        DateTime dt = days[5].date.AddMonths(-1);
        CacheConnection con = new CacheConnection();
        con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";
        con.Open();
        int d = 0;
        double avg = 0; int c = 0;
        while (dt <= days[5].date)
        {
            Point prev = new Point(0, gr2.Height);
            for (int r = 0; r < st.Count; r++)
            {
                string quer = "select subject, mark from Marks where
stud='" + st[r] + "' and date_m=to_date('" + dt.ToString("yyyy-MM-dd") +
"', 'YYYY-MM-DD')";

                CacheCommand com = new CacheCommand(quer, con);
                CacheDataReader rd = com.ExecuteReader();
                while (rd.Read())
                {
                    string sub
=
rd[rd.GetOrdinal("subject")].ToString();

```

```

                                string                mk                =
rd[rd.GetOrdinal("mark")].ToString();
                                for (int i = 0; i < sd.Count; i++)
                                {
                                    if (sd[i].subject == sub)
                                    {
                                        sd[i].addMark(mk);
                                        avg += Double.Parse(mk);
                                        c++;
                                        break;
                                    }
                                }
                                }
                                rd.Close();
                                com.Dispose();
                                }
                                double val = 0;
                                if (c > 0) val = avg/(double)c;
                                Point to = new Point(d * step, (int)(gr2.Height - val *
height));
                                prev = to;
                                g2.FillRectangle(Brushes.Blue, new Rectangle(prev,new
Size(step, (int) (height*val))));

                                dt = dt.AddDays(1); d++;
                                }
                                con.Close();
                                datesLabel2.Text = datesLabel.Text;
                                }

                                void loadData()
                                {
                                    List<SubjData> sd = new List<SubjData>();
                                    for (int i = 0; i < 6; i++)
                                    {
                                        for (int j = 1; j <= 8; j++)
                                        {
                                            string                sub                =
days[i].dayPanel.GetControlFromPosition(1, j).Text;
                                            if (sub != "")
                                            {
                                                bool isex = false;
                                                for (int k = 0; k < sd.Count; k++)
                                                {
                                                    if (sd[k].subject == sub) { isex = true;
break; }
                                                }
                                                if (!isex)
                                                {
                                                    SubjData sdt = new SubjData();
                                                    sdt.subject = sub;
                                                    sdt.marks = "";
                                                    sd.Add(sdt);
                                                }
                                            }
                                        }
                                    }
                                }
                                }
                                DateTime dt = days[5].date.AddMonths(-1);
                                CacheConnection con = new CacheConnection();
                                con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";
                                con.Open();
                                int d = 0; Point prev = new Point(0, gr1.Height);

```

```

        double avg = 0; int c = 0;
        while (dt <= days[4].date)
        {
            string quer = "select subject, mark from Marks where
stud='" + sid + "' and date_m=to_date('" + dt.ToString("yyyy-MM-dd") +
"', 'YYYY-MM-DD')";

            CacheCommand com = new CacheCommand(quer, con);
            CacheDataReader r = com.ExecuteReader();
            //double avg = 0; int c = 0;
            while (r.Read())
            {
                string sub = r[r.GetOrdinal("subject")].ToString();
                string mk = r[r.GetOrdinal("mark")].ToString();
                for (int i = 0; i < sd.Count; i++)
                {
                    if (sd[i].subject == sub)
                    {
                        sd[i].addMark(mk);
                        avg += Double.Parse(mk);
                        c++;
                        break;
                    }
                }
            }
            r.Close();
            com.Dispose();
            double val=0;
            if(c>0)val = avg / (double)c;
            Point to = new Point(d * step, (int)(gr1.Height - val *
height));

            gl.DrawLine(new Pen(Brushes.Blue), prev, to);
            prev = to;
            if(d%3==0)
            {
                datesLabel.Text += dt.ToString("dd.MM") + "    ";
            }
            dt = dt.AddDays(1); d++;
        }
        con.Close();
        string result = "";
        for (int i = 0; i < sd.Count; i++)
        {
            result += sd[i].subject + ": " + sd[i].marks + "\n";
        }
        marksLabel.Text = "Оценки:\n" + result;
        loadClassData(sd);
    }

}

```

```

// Текст файла TabelForm.cs:
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using InterSystems.Data.CacheClient;

namespace CourseWork01

```

```

{
    public partial class TabelForm : Form
    {
        string sid;
        List<DiaryDay> days = new List<DiaryDay>(6);
        public TabelForm(string stid, List<DiaryDay> ld)
        {
            sid = stid;
            days = ld;
            InitializeComponent();
            loadData();
        }
        class SubjData
        {
            public string subject;
            public double frst, scnd;
            public int c1, c2;
            public SubjData() { }
        }
        void loadData()
        {
            List<SubjData> sd = new List<SubjData>();
            for (int i = 0; i < 6; i++)
            {
                for (int j = 1; j <= 8; j++)
                {
                    string sub =
                    days[i].dayPanel.GetControlFromPosition(1, j).Text;
                    if (sub != "")
                    {
                        bool isex = false;
                        for (int k = 0; k < sd.Count; k++)
                        {
                            if (sd[k].subject == sub) { isex = true;
                            break; }
                        }
                        if (!isex)
                        {
                            SubjData sdt = new SubjData();
                            sdt.subject = sub;
                            sdt.c1 = sdt.c2 = 0;
                            sdt.frst = sdt.scnd = 0;
                            sd.Add(sdt);
                        }
                    }
                }
            }
            DateTime dt, dte;
            if (DateTime.Today.Month < 9) { dt = new
            DateTime(DateTime.Today.Year-1, 9, 1); dte = new DateTime(DateTime.Today.Year,
            5, 31); }
            else { dt = new DateTime(DateTime.Today.Year, 9, 1); dte =
            new DateTime(DateTime.Today.Year+1, 5, 31); }
            CacheConnection con = new CacheConnection();
            con.ConnectionString = "Server=localhost; Port=1972;
            Namespace=USER; Password=12345678; User ID=Egorman";
            con.Open();
            int d = 0;
            while (dt <= dte)
            {
                string quer = "select subject, mark from Marks where
            stud='" + sid + "' and date_m=to_date('" + dt.ToString("yyyy-MM-dd") +
            "', 'YYYY-MM-DD)";
                CacheCommand com = new CacheCommand(quer, con);
            }
        }
    }
}

```

```

        CacheDataReader r = com.ExecuteReader();
        while (r.Read())
        {
            string sub = r[r.GetOrdinal("subject")].ToString();
            string mk = r[r.GetOrdinal("mark")].ToString();
            for (int i = 0; i < sd.Count; i++)
            {
                if (sd[i].subject == sub)
                {
                    if (dt < DateTime.Parse("31.12.2016"))
                    {
                        sd[i].frst += Double.Parse(mk);
                    }
                    else { sd[i].scnd += Double.Parse(mk); }

                    break;
                }
            }
            r.Close();
            com.Dispose();
            dt = dt.AddDays(1); d++;
        }
        con.Close();
        for(int i=0; i<sd.Count; i++)
        {
            if (sd[i].c1 > 0) sd[i].frst /= sd[i].c1;
            if (sd[i].c2 > 0) sd[i].scnd /= sd[i].c2;
            tableLayoutPanell1.GetControlFromPosition(0, i+1).Text =
sd[i].subject;
            tableLayoutPanell1.GetControlFromPosition(1, i+1).Text =
((int)sd[i].frst).ToString();
            tableLayoutPanell1.GetControlFromPosition(2, i+1).Text =
((int)sd[i].scnd).ToString(); ;
            tableLayoutPanell1.GetControlFromPosition(3, i+1).Text =
(((int)sd[i].frst+ (int)sd[i].scnd)/ (int)2).ToString();
        }
    }
}

```

// Текст файла RatingForm.cs:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using InterSystems.Data.CacheClient;

```

```

namespace CourseWork01

```

```

{
    public partial class RatingForm : Form
    {
        string sid;
        List<DiaryDay> days; Bitmap gr; Graphics g; int step, height;
        class SubjData
        {
            public string subject;
            public SubjData() { }
        }
        public RatingForm(string sd, List<DiaryDay> ds)
    }
}

```

```

    {
        sid = sd; days = ds;
        InitializeComponent();
        step = pictureBox1.Width / 30;
        height = pictureBox1.Height / 13;
        loadSubjects();
    }
    void loadSubjects()
    {
        List<SubjData> sd = new List<SubjData>();
        for (int i = 0; i < 6; i++)
        {
            for (int j = 1; j <= 8; j++)
            {
                string sub =
days[i].dayPanel.GetControlFromPosition(1, j).Text;
                if (sub != "")
                {
                    bool isex = false;
                    for (int k = 0; k < sd.Count; k++)
                    {
                        if (sd[k].subject == sub) { isex = true;
break; }
                    }
                    if (!isex)
                    {
                        SubjData sdt = new SubjData();
                        sdt.subject = sub;
                        sd.Add(sdt);
                        comboBox1.Items.Add(sub);
                    }
                }
            }
        }
    }
    private void comboBox1_SelectedIndexChanged(object sender,
EventArgs e)
    {
        gr = new Bitmap(pictureBox1.Width, pictureBox1.Height);
        g = Graphics.FromImage(gr);
        g.FillRectangle(Brushes.White, 0, 0, gr.Width, gr.Height);
        datesLabel.Text = "";

        DateTime dt;
        if (DateTime.Today.Month<9) dt = new
DateTime(DateTime.Today.Year-1,9,1);
        else dt = new DateTime(DateTime.Today.Year, 9, 1);
        int counter = 0;
        counter = ((DateTime.Today - dt).Days / 31)+1 ;
        CacheConnection con = new CacheConnection();
        con.ConnectionString = "Server=localhost; Port=1972;
Namespace=USER; Password=12345678; User ID=Egorman";
        con.Open();
        int d = 0; Point prev = new Point(0, gr.Height);
        double avg = 0; int c = 0, dts = 0, p=2;
        while (dt <= DateTime.Today)
        {
            dts++;
            string quer = "select mark from Marks where stud='" +
sid + "' and date_m=to_date('" + dt.ToString("yyyy-MM-dd") + "','YYYY-MM-DD')
and subject='"+comboBox1.Text+"'";
            CacheCommand com = new CacheCommand(quer, con);
            CacheDataReader r = com.ExecuteReader();
            while (r.Read())

```

```

        {
            string mk = r[r.GetOrdinal("mark")].ToString();
            avg += Double.Parse(mk);
            c++;
        }
        r.Close();
        com.Dispose();
        if (dts == counter)
        {
            d++;
            dts = 0;
            double val = 0;
            if (c > 0) val = avg / (double)c;
            Point to = new Point(d * step, (int)(gr.Height - val
* height));

            prev = to;
            g.FillRectangle(Brushes.Blue, new Rectangle(prev,
new Size(step, (int)(height * val))));
            g.DrawRectangle(Pens.Black, new Rectangle(prev, new
Size(step, (int)(height * val))));
            if (p == 2) { datesLabel.Text +=
dt.ToString("dd.MM") + " "; p = 0; }
            p++;
        }
        dt = dt.AddDays(1);
    }
    con.Close();
    pictureBox1.Image = gr;
}
}
}

```

ДОДАТОК Д
Слайди презентації

Міністерство освіти та науки України
Національний університет «Запорізька політехніка»

Програмне забезпечення для роботи електронного щоденника здобувача освіти

Бражніков Ілля Андрійович

ст. групи КНТ-119сп

Гофман Євгеній Олександрович

к.т.н., доцент НУ «Запорізька політехніка»

Рисунок Д.1 – Слайд 1

Об'єкт, предмет та мета роботи

Об'єкт дослідження – процес розробки програмного забезпечення для освітньої діяльності.

Предмет дослідження – програмні засоби для надання інформації про результати оцінювання здобувачів освіти.

Метою роботи є розробка програмного забезпечення для надання інформації про успішність здобувачів освіти у вигляді електронного щоденника.

Рисунок Д.2 – Слайд 2

Завдання роботи

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для підтримки освітньої діяльності;
- здійснити проектування програмного забезпечення для надання інформації про успішність здобувачів освіти у вигляді електронного щоденника;
- створити програмне забезпечення для відображення успішності учнів, контролю розкладу уроків, оцінок та домашнього завдання учнів;
- виконати тестування розробленого програмного забезпечення для надання інформації про успішність здобувачів освіти у вигляді електронного щоденника.

3

Рисунок Д.3 – Слайд 3

Порівняння існуючих аналогів

Критерій, за яким порівнюють	Canvas LMS	Google Classroom	ProProfs	Schoology	TalentLMS
Зручність та зрозумілість інтерфейсу	+	+	+-	+	+
Онлайн-засоби навчання	+	+	-	+	+
Інструменти для створення опитувань	-	+	+-	+	+
Засоби підтримки віртуального навчання	+	-	+	+	+
Доступ для батьків	-	-	-	-	-
Безкоштовність	-	+	-	-	-

4

Рисунок Д.4 – Слайд 4

Порівняння мов програмування

Критерій порівняння мов програмування	Мова програмування		
	C#	Java	PHP
Зручність для створення ПЗ для роботи електронного щоденника	+	–	–
Робота з базами даних	+	+–	+
Об'єктно-орієнтовані засоби	+	+	+–
Засоби захисту даних	+	+–	+
Необхідність наявності проміжного програмного забезпечення	+	+	+–

5

Рисунок Д.5 – Слайд 5

Порівняння середовищ розробки

Критерій порівняння середовищ розробки	Середовище розробки		
	MS Visual Studio	Atom	PyCharm
Зручність використання для створення програмного забезпечення електронного щоденника здобувача освіти	+	+–	+–
Ефективність засобів редагування файлів програмного проекту	+	+	+–
Наявність засобів для роботи з мовою C#	+	+–	–
Зручність для створення вебзастосунків	+	+	+–
Можливість кросплатформенної розробки	+	+	+–

6

Рисунок Д.6 – Слайд 6

Вибір системи керування базами даних

Критерій порівняння СКБД	СКБД		
	InterSystems Cache	Oracle	MySQL
Зручність для створення ПЗ електронного щоденника здобувача освіти	+	—	—
Безпека даних	+	+—	+
Низька вартість	+	—	+
Масштабованість	+	+—	+—
Продуктивність	+	+—	+

7

Рисунок Д.7 – Слайд 7

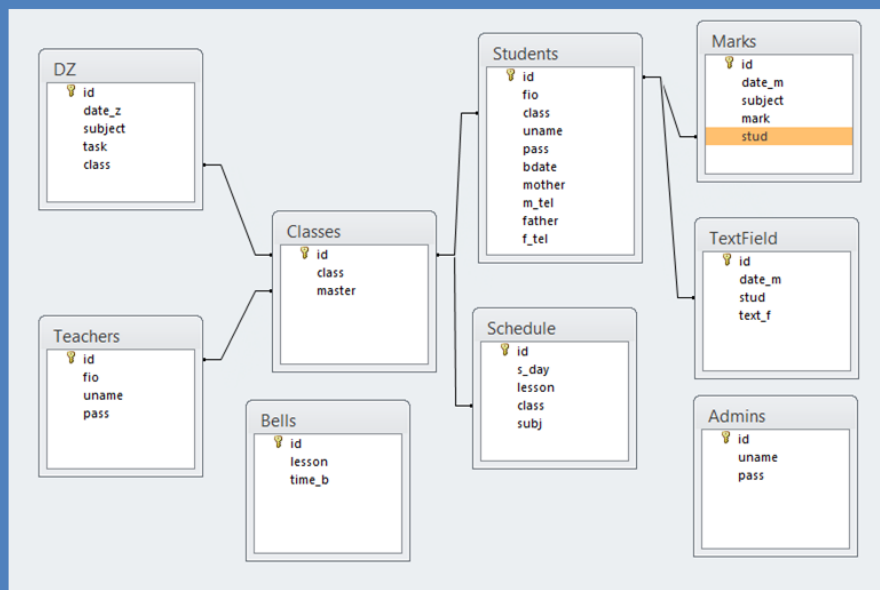
Схема взаємодії основних компонентів програмного забезпечення



8

Рисунок Д.8 – Слайд 8

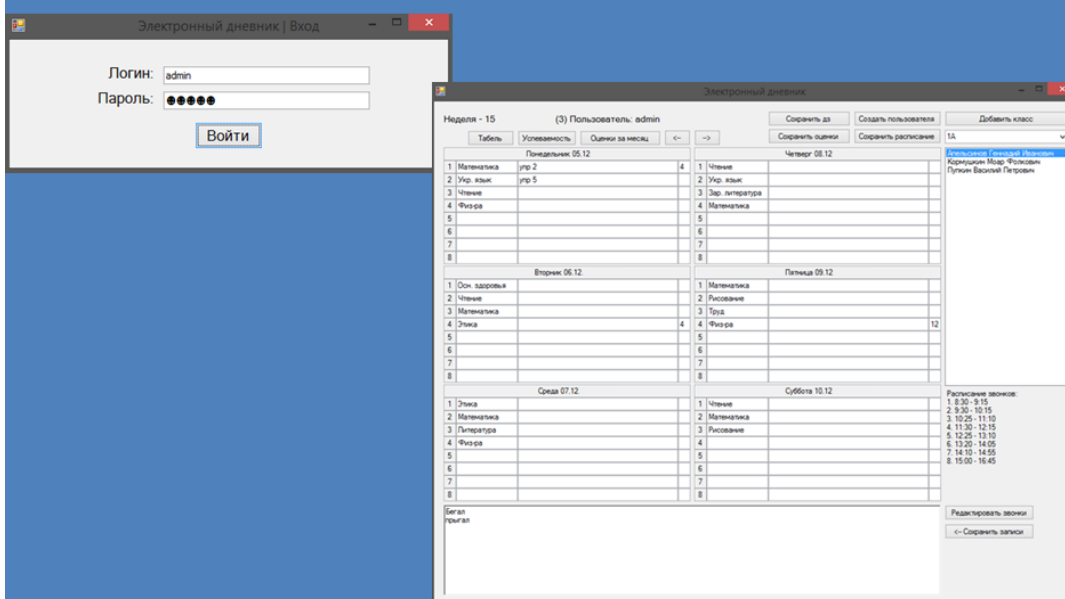
Схема бази даних



9

Рисунок Д.9 – Слайд 9

Робота з програмою. Реєстрація



10

Рисунок Д.10 – Слайд 10

[illegible]

Рисунок Д.12 – Слайд 12

Висновки

В результаті виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки програмного забезпечення для освітньої діяльності.

Розроблено програмне забезпечення для роботи електронного щоденника здобувача, яке забезпечує виконання таких функцій:

- реєстрація учнів, вчителів за допомогою адміністратора;
- редагування списку класів;
- збереження розкладу, домашніх завдань, оцінок та поміток для учнів;
- збереження розкладу дзвінків;
- відображення успішності учнів;
- підрахунок та відображення підсумкових оцінок.

Результати тестування програмного забезпечення для роботи електронного щоденника здобувача освіти підтвердили його працездатність.

Усі завдання випускної дипломної роботи бакалавра повністю виконано.

13

Рисунок Д.13 – Слайд 13

Дякую за увагу

14

Рисунок Д.14 – Слайд 14