Содержание

|  |  |
| --- | --- |
| Введение | 3 |
| 1 Назначение и цели разработки | 4 |
| 2 Разработка технического проекта на основе анализа требований | 5 |
| 2.1 Определение спецификаций программного обеспечения | 5 |
| 2.2 Проектирование модели данных и диаграммы классов | 11 |
| 3 Реализация | 17 |
| 3.1 Обоснование выбора средств разработки | 17 |
| 3.2 Разработка базы данных в среде СУБД | 18 |
| 3.3 Описание программных модулей | 19 |
| 4 Тестирование программных модулей | 23 |
| 4.1 Модульное тестирование | 23 |
| 4.2 Интеграционное тестирование | 24 |
| 5. Эксплуатационная документация | 25 |
| 5.1 Руководство пользователя | 25 |
| Заключение | 26 |
| Список использованных источников | 27 |
| Приложение А | 30 |
| Приложение Б | 34 |
| Приложение В | 35 |
| Приложение Г | 36 |
| Приложение Д | 37 |
|  |  |
|  |  |

Введение

В настоящее время стремительно развиваются компьютерные технологии, влекущие за собой внедрение в деятельность любой организации программных модулей и разработку программного обеспечения, учитывающих специфику какой-либо организации.

Особенно актуально внедрение и использование компьютерной техники в высших, а также средне-специальных учебных заведениях, в частности, на специальностях, связанных с информационными системами. Субъектом, рассматриваемым в рамках данной работы, является Краснодарский колледж электронного приборостроения.

Учитывая, что в последние годы колледж интенсивно увеличивает число обучающихся студентов, контроль успеваемости и отслеживание за текущим состоянием рейтинга каждого студента становится непростым и трудоёмким процессом. В результате чего, процесс контроля успеваемости студентов причиняет много неудобств и затрат времени, что сказывается на несвоевременное выставление рейтинга, а также ошибки при подсчете итоговых результатов каждого студента.

Для решения данной проблемы требуется разработать программные модуль настольного приложения, позволяющий вести процесс оценивания знаний студентов, отслеживать текущее состояние успеваемости каждого студента, а также формировать статистику для каждой группы.

Курсовой проект включает себя проектирование и разработку базы данных, а также модулей настольного приложения в рамках информационной системы, предназначенной для анализа успеваемости студентов в колледже.

1 Назначение и цели разработки

Заданием предусмотрена разработка программных модулей подсистем «Заведующий отделением» и «Преподаватель» информационной системы колледж. Разработанные модули предназначены для автоматизации рабочих мест заведующего отделением и преподавателя в колледже, а также для сбора, обработки и хранения информации о студентах, а также анализа успеваемости по конкретным предметам,

Автоматизация позволит получить следующие преимущества по сравнению с нынешней реализацией данных задач:

- Быстрое формирование отчетов по результатам успеваемости конкретного студента, группы.

- Быстрое выставление результатов конкретного студента по конкретному предмету.

- Ускорение процесса обновления информации об успеваемости студентов.

Данная система позволит максимально сократить присутствие человеческого фактора в ряде важных задач, таких как добавление нового студента, выставление оценок, формирование отчетности, учет студентов.

2 Разработка технического проекта на основе анализа требований

В данной работе был выбран объектно-ориентированный метод технологии проектирования программных модулей. Объектно-ориентированная модель наиболее адекватно отражает реальный мир, представляющий собой совокупность взаимодействующих объектов.

2.1 Определение спецификаций программного обеспечения

Рассмотрим определение вариантов использования (прецедентов).

Система анализа успеваемости студентов требуется прежде всего следующим заинтересованным лицам:

- заведующий отделением студентов;

- преподаватель колледжа;

- администратор системы.

На начальном этапе проектирования системы можно ограничиться двумя важными ролями действующих лиц: заведующий отделением и преподаватель. Соответственно основные прецеденты (варианты использования) для разрабатываемой системы следующие.

Прецеденты для заведующего отделением:

- П1 – добавить нового студента;

- П2 – начислить стипендию студенту на основе анализа его успеваемости;

- П3 – сформировать отчёт об успеваемости;

Прецеденты для преподавателя:

- П3 – выставить оценку студенту по предмету.

Диаграмма вариантов использования, созданная средствами MS Visio, для проектируемой системы представлена на рисунке 1 [10].

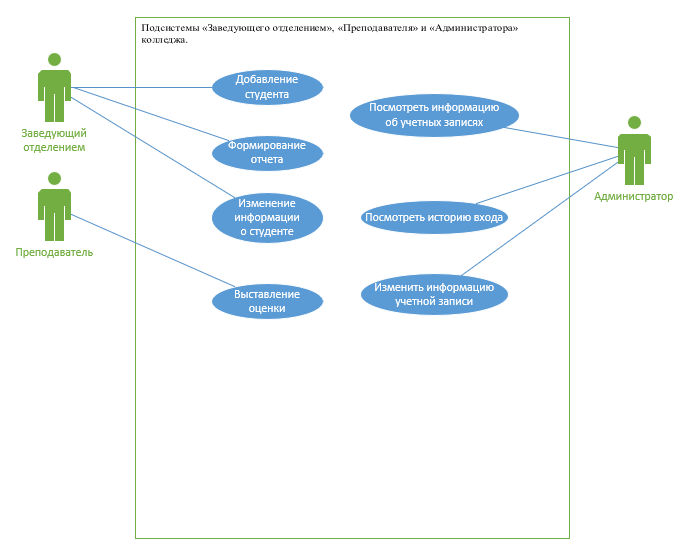


Рисунок 1 – диаграмма вариантов использования программной системы колледжа.

В таблице 1 представлено описание главного раздела сценария прецедента (варианта использования) «Сформировать отчет»

Таблица 1 - Главный раздел сценария «Сформировать отчёт»

|  |  |
| --- | --- |
| Вариант использования | Сформировать отчет |
| Актеры | Преподаватель |

Продолжение таблицы 1

|  |  |
| --- | --- |
| Краткое описание | Преподаватель выбирает конкретного студента либо группу и результаты по необходимому предмету |
| Цель | Анализ успеваемости |
| Тип | Базовый |
| Ссылки на другие варианты использования | Включает в себя регистрацию новой записи об успеваемости конкретного студента при её отсутствии в базе |

В таблице 2 описана последовательность действий преподавателя, приводящая к успешному выполнению прецедента (варианта использования) «Выставить оценку»

Таблица 2 - Сценарий успешного выполнения варианта использования «Сформировать отчет»

|  |  |
| --- | --- |
| Действия актеров | Отклик системы |
| 1.Преподаватель ставит оценку  Исключение 1. В списке нет необходимого результата по предмету | 2. Система возвращает список студентов  3. Система возвращает список предметов |
| 4. Преподаватель выставляет оценку | 5. Система сохраняет результат |

В таблице 3 представлен сценарий обработки исключительных ситуаций для варианта использования «Сформировать отчет».

Таблица 3 - Обработка исключительных ситуаций для варианта использования «Сформировать отчет»

|  |  |
| --- | --- |
| Действия актеров | Отклик системы |
| Исключение 1. Отсутствует необходимый результат по предмету | |
| 4. Преподавать выбирает необходимого студента | 5. Система добавляет результат в базу данных |
| 6. Преподаватель выставляет оценку | 7. Система сохраняет результат |

С помощью диаграммы деятельности, представленной на рисунке 2, описан алгоритм реализации описанных сценариев для варианта использования «Сформировать отчет» [7].

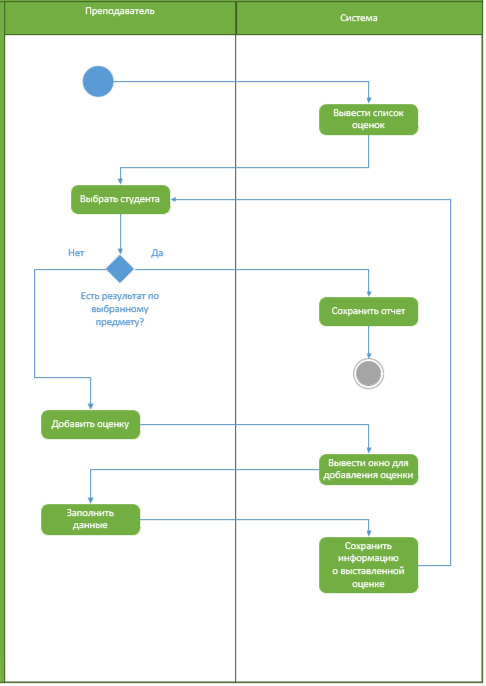


Рисунок 2 – Диаграмма деятельности для варианта использования «Сформировать отчёт»

На диаграмме последовательностей, представленной на рисунке 3, описаны требования реализации сценария «Сформировать отчет», которая отражает синхронизации операций при реализации процесса формирования отчета.

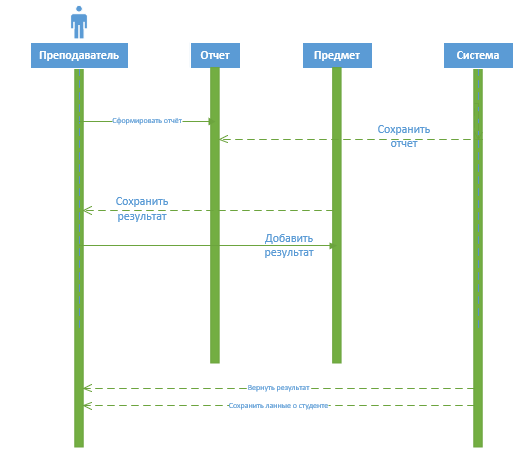


Рисунок 3 - Диаграмма последовательности для варианта использования «Сформировать отчёт»

Разработанные спецификации программного обеспечения на языке UML средствами MS Visio показывают основные действия пользователей автоматизируемой системы и алгоритмы их выполнения.

2.2 Проектирование модели данных и диаграммы классов

На основе анализа требований заказчика к функциям программной системы, требованиям к организации входных и выходных данных, с учетом спроектированных требований к реализации функций, описанных в диаграммах вариантов использования, деятельности и последовательности, была разработана модель данных системы и описана в виде ER-диаграммы, позволяющая четко описать требования к представлению логической структуры данных, на основе которой в последующем будет разработана физическая структура данных для хранения во внешней памяти и программной обработки.

На рисунке 4 представлена ER – диаграмма для программной системы «Анализ успеваемости обучающихся в колледже» [6].

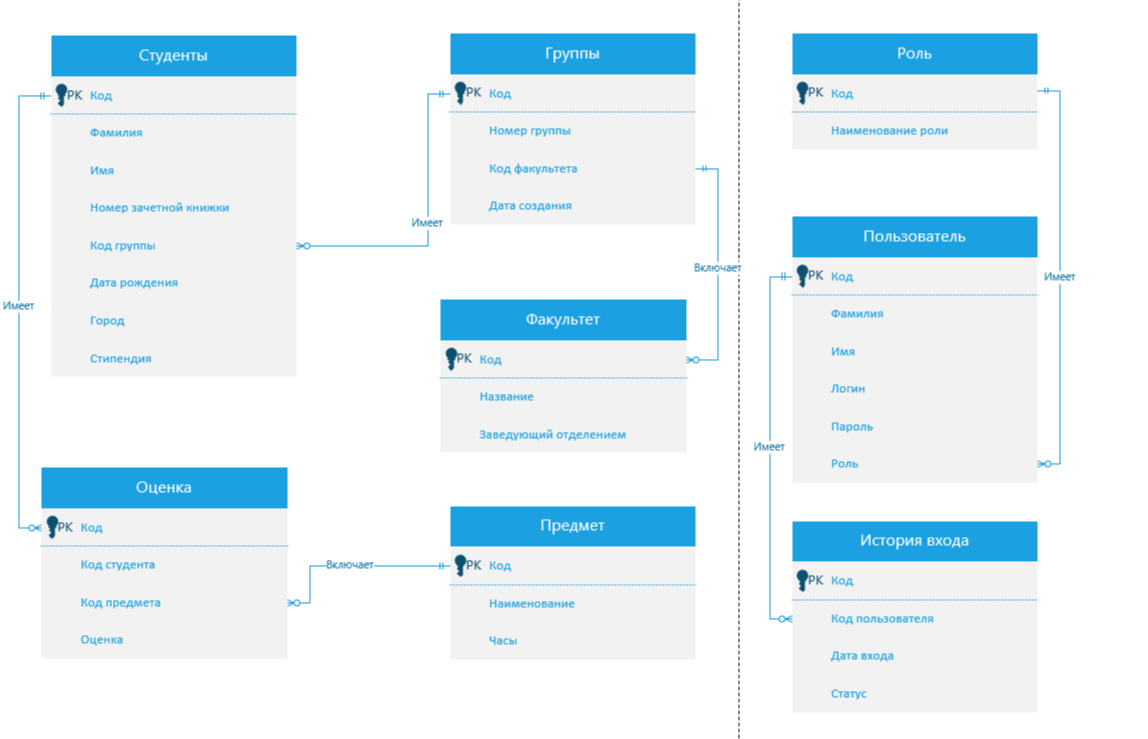


Рисунок 4 – ER-диаграмма для программной системы анализ успеваемости обучающихся в колледже

В таблице 4 представлены сведения о сущностях и связях между сущностями ER-диаграммы из рисунка 4.

Таблица 4 – Характеристики сущностей и связей между сущностями   
ER-диаграммы для программной системы анализ успеваемости обучающихся в колледже

|  |  |  |  |
| --- | --- | --- | --- |
| Сущность | Назначение сущности | Ключ | Характеристики связей |
| Студенты | Содержит данные о студентах:  - фамилия;  - имя;  - номер зачетной книжки;  - город;  - стипендия. | Код студента (PK) | Студент – Оценка (1..n)  Студент – Группа (1..n) |
| Группа | Содержит информацию о группах:  - номер группы;  - факультет;  - дата создания. | Код группы (PK) | Группа – Студенты (1..n) |
| Пользователи | Содержит данные о пользователе:  - фамилия;  - имя;  - логин;  - пароль;  - роль. | Код пользователя (PK) | Пользователи – Роль(1..n)  Пользователи – История входа (1..n) |

Продолжение таблицы 4

|  |  |  |  |
| --- | --- | --- | --- |
| Оценка | Содержит данные о оценке:  - студент;  - предмет. | Код оценки (PK) | Оценка – Студенты (1..n)  Оценка – Предмет (1..n) |
| Предмет | Содержит данные о предмете:  - наименование;  - часы. | Код предмета(PK) | Предмет – Оценка (n..1) |
| Факультет | Содержит данные о факультете:  - наименование;  - заведующий отделением | Код факультета (PK) | Факультет – Группа (1..n) |
| Роль | Содержит данные о роли:  - наименование роли. | Код роли(PK) | Роль – Пользователь (1..n) |
| История входа | Содержит информацию о авторизации:  - дата входа;  - статус. | Код (PK) | Пользователи – История входа (1..n) |

На следующем этапе была спроектирована диаграмма классов, которая описывает требования к внутренней организации проектируемого программного обеспечения на уровне интерфейсов и классов, их кооперации, атрибутов и методов классов, правил их взаимодействия.

На рисунке 5 представлена диаграмма классов для программной системы анализ успеваемости студентов в колледже [5].

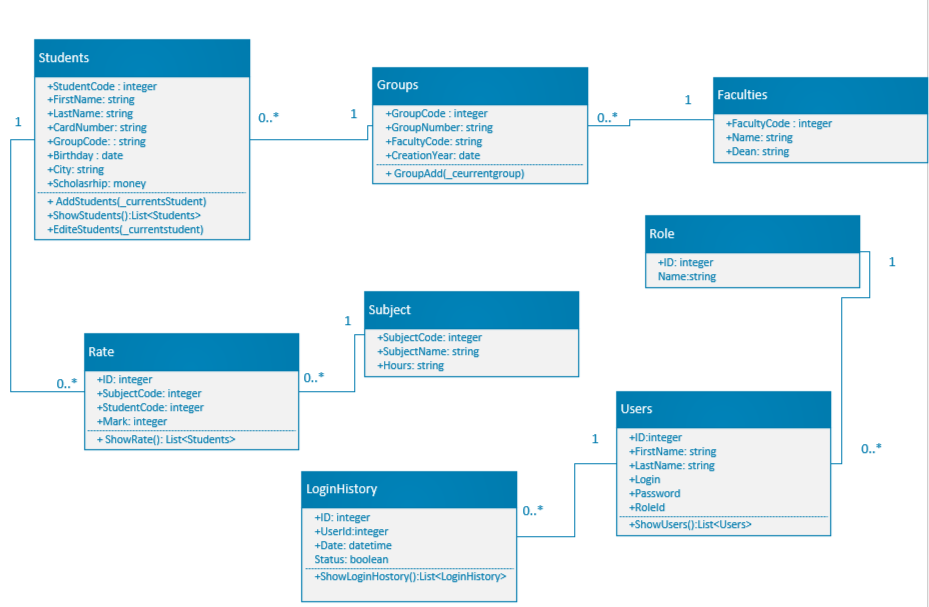


Рисунок 5 –Диаграмма классов для программной системы анализ успеваемости обучающихся в колледже.

Данная диаграмма классов описывает типы объектов системы колледжа и отношения, которые существуют между ними, а также отображаются операции классов.

Таблица 5 – список классов

|  |  |  |
| --- | --- | --- |
| Классы | Атрибуты | Методы |
| - Студенты | -код\_студента  -имя  -фамилия  -номер зачетной книжки  -код\_группы  -дата рождения  -город  -стипендия | -редактировать студента  -добавлять нового студента  -выводить список студентов  -удалить студента |

Продолжение таблицы 5

|  |  |  |
| --- | --- | --- |
| -Группы | -код\_группы  -номер группы  -дата создания | -выводить список групп |
| -Оценка | -код\_оценки  -код\_студента  -код\_предмета  -оценка | -выставлять оценку по предмету |

2.3 Конструирование прототипа

Для построения прототипа первоначально были составлены требования к макету приложения, для которого разрабатываются программные модули.

Все компоненты должны иметь единый согласованный внешний вид, соответствующий руководству по стилю, а также следующим требованиям:

* разметка и дизайн (предпочтение отдается масштабируемой компоновке; должно присутствовать ограничение на минимальный размер окна; должна присутствовать возможность изменения размеров окна, где это необходимо; увеличение размеров окна должно увеличивать размер контентной части, например, таблицы с данными из БД);
* группировка элементов (в логические категории);
* использование соответствующих элементов управления;
* расположение и выравнивание элементов;
* последовательный переход фокуса по элементам интерфейса;
* общая компоновка логична, понятна и проста в использовании;
* последовательный пользовательский интерфейс, позволяющий перемещаться между существующими окнами или страницами в приложении;
* соответствующий заголовок на каждом окне приложения.

Основные требования руководства по стилю:

* все экранные формы пользовательского интерфейса должны иметь заголовок с логотипом, представленном на рисунке 7. Цвет, размер и пропорции логотипа не должны изменяться при использовании программы. Так же для приложения должна быть установлена иконка, соответствующая логотипу;
* тип шрифта – Comic Sans MS;

цветовая схема предусматривает использование в качестве основного фона – белый цвет RGB (240, 248, 255), в качестве дополнительного – цвет RGB (255, 182, 193), для акцентирования внимания пользователя на целевое действие интерфейса – цвет RGB (0, 191, 26) или RGB(0, 0, 255).



Рисунок 6 – Логотип приложения

С учетом требований к макету и руководству по стилю, для обеспечения требуемых функций, был разработан прототип программы.

На рисунке 7 представлен прототип главного окна программы.

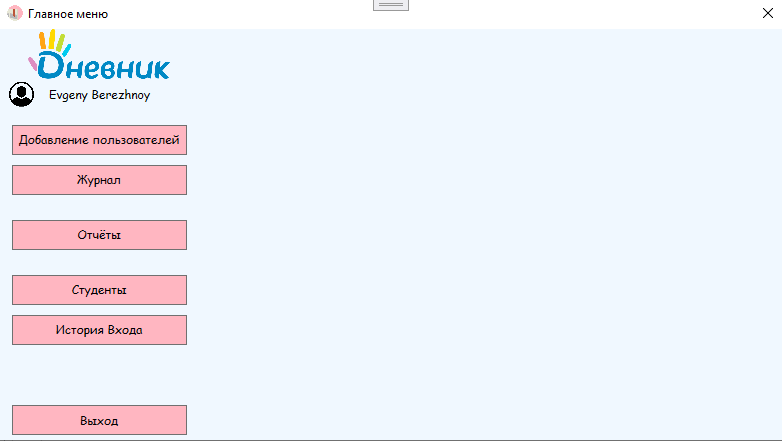


Рисунок 7 – Прототип главного окна программы

На рисунке 7 наблюдается, что окно соответствует вышеописанному руководству по стилю.

3 Реализация

3.1 Обоснование выбора средств разработки

Так как в техническом задании (см. Приложение А) указано, что система должна работать под операционной системой Windows, то выбор останавливается на языке программирования C# и платформе .NET Framework

В таблице 6 представлены некоторые возможные средства разработки системы анализа обучающихся студентов в колледже.

|  |  |  |
| --- | --- | --- |
| Язык программирования и технология | Плюсы | Минусы |
| C#, .NET | -мощная библиотека классов. .NET представляет собой единую для всех поддерживаемых языков библиотеку классов  -Большое количество  специальных конструкций, разработанные для понимания и написания кода | -приоритетная ориентированность на платформу Windows  -язык бесплатен только для небольших фирм, индивидуальных программистов, стартапов и учащихся. |
| Python | -открытый код  -простота синтаксиса  -бесплатность | -невысокая скорость выполнения программ  -эволюция языка |

3.2 Разработка базы данных в среде СУБД

На основе ранее спроектированной ER-диаграммы в среде Microsoft SQL Server Management Studio 18 была разработана база данных Academic. На рисунке 9 представлена диаграмма разработанной базы данных [4].

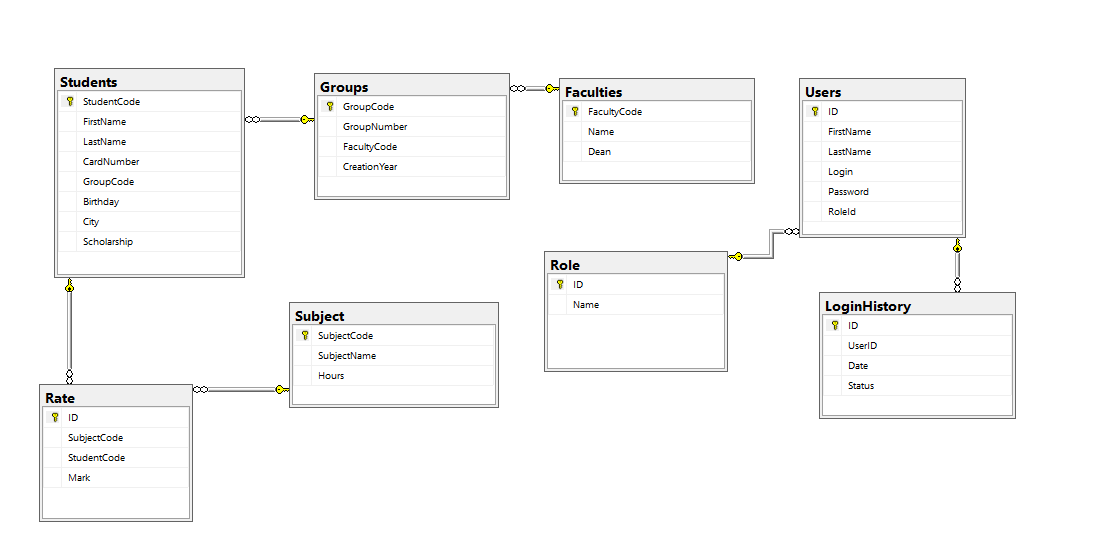


Рисунок 8 – Диаграмма базы данных Academic

В таблице 9 представлено описание таблиц базы данных Academic

|  |  |
| --- | --- |
| Название таблицы | Краткое описание |
| Students | Предназначена для хранения информации о студентах |
| Rate | Предназначена для хранения информации об оценках |
| Groups | Предназначена для хранения информации о группах |
| Role | Предназначена хранения информации о ролях |
| Faculties | Предназначена для хранения информации о факультетах |

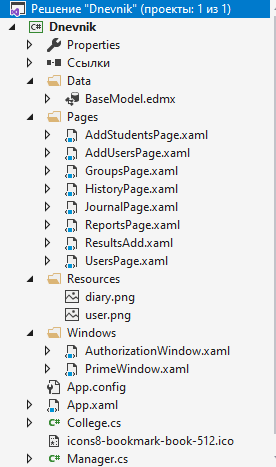
Продолжение таблицы 9

|  |  |
| --- | --- |
| Subject | Предназначена для хранения информации о предметах |
| Login History | Предназначена для хранения информации об истории входа пользователей |
| Users | Предназначена для хранения информации о пользователях |

Скрипт разработанной базы данных находится в Приложении Г.

3.3 Описание программных модулей

На рисунке 10 представлена информация о структуре проекта приложения



В таблице 9 описаны назначения директорий из проекта приложения.

Таблица 10 – назначение директорий

|  |  |
| --- | --- |
| Название директории | Назначение |
| Data | Хранит внешние данные |
| Data/BaseModel | Хранит модель базы данных |
| Resources | Хранит графические ресурсы |
| Pages | Хранит все страницы приложения |
| Windows | Хранит все окна приложения |

На рисунке 11 представлена экранная форма модуля редактирования студента



Рисунок 10 – Форма модуля «Редактировать Студента»

Код разметки окна, представленного на рисунке 10 представлен на рисунке Б.1

Результат работы поиска представлен на рисунке 12

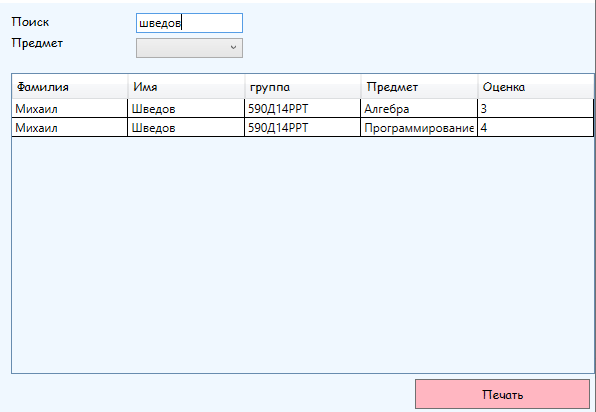


Рисунок 11 – Результат работы поиска

Образец выходного документа представлен на рисунке 12

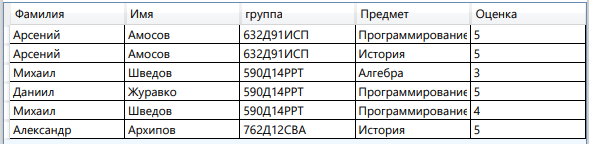


Рисунок 12 – Форма выходного документа

4 Тестирование программных модулей

4.1 Модульное тестирование

5 Эксплуатационная документация

5.1 Руководство пользователя

Руководство пользователя для заведующего отделением: при открытии приложения появляется окно входа, в которое необходимо ввести свой логин, а также пароль, выданный администратором системы. После успешной авторизации в приложение появляется меню с возможностями просмотра списка студентов, а также формирование отчетности по различным критериям. При просмотре списка студентов, можно редактировать различную информацию о студенте, а также производить смену группы обучающегося.

Руководство пользователя для преподавателя: при открытии приложения появляется окно входа, в которое необходимо ввести логин и пароль, выданный администратором системы. После успешной авторизации открывается возможность выставлять оценки обучающимся студентам, а также возможность формирования отчетности об успеваемости конкретного обучающегося в колледже.

Руководство пользователя для администратора: при открытии приложения появляется окно входа, в которое необходимо ввести свой логин и пароль. После успешной авторизации появляется список, в котором присутствуют такие функции, как: просмотр истории входа, добавление пользователей в систему, просмотр списка студентов, редактирование информации о студентах, а также формирование отчетности.

Заключение

В результате проделанной работы в рамках курсового проектирования было разработано настольное приложение «Dnevnik» с помощью средств языка C# и платформы .NET Framework, а также с использованием системой управления базами данных Microsoft SQL Server Management Studio 18.

Настольное приложение включает в себя такие функции, как:

- Ведение учета успеваемости обучающихся в колледже

- Анализ успеваемости обучающихся в колледже

- Формирования настраиваемых отчетов по конкретному студенту или группе

- Ведение учета обучающихся в электронном виде

Данное приложение поможет упростить работу педагогов в средне-специальных образовательных учреждениях, что повысит скорость и простоту анализа успеваемости обучающихся в колледже.

Список использованных источников

1. ГОСТ 2.105 – 2019. Единая система конструкторской документации (ЕСКД). Общие требования к текстовым документам (Издание с Изменением N 1) = Unified system for design documentation. General requirements for textual documents: национальный стандарт Российской Федерации: издание официальное: утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 29 апреля 2019 г. № 175-ст: введен впервые: дата введения 2021-02-01 / Разработан Федеральным государственным унитарным предприятием «Российский научно-технический центр информации по стандартизации, метрологии и оценке соответствия» (ФГУП «СТАНДАРТИНФОРМ»). – Москва.: Стандартинформ, 2021. – 35 с. –Текст непосредственный.
2. ГОСТ 7.0.100 – 2018. Система стандартов по информации, библиотечному и издательскому делу (СИБИД). Библиографическая запись. Библиографическое описание. Общие требования и правила составления (с Поправкой) = System of standards on information, librarianship and publishing. Bibliographic record. Bibliographic description. General requirements and rules: национальный стандарт Российской Федерации: издание официальное: утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 3 декабря 2018 года № 1050-ст: введен впервые: дата введения 2019-07-01 / Разработан Федеральным государственным унитарным предприятием "Информационное телеграфное агентство России (ИТАР-ТАСС)", филиал "Российская книжная палата", Федеральным государственным бюджетным учреждением "Российская государственная библиотека", Федеральным государственным бюджетным учреждением "Российская национальная библиотека". – Москва: Стандартинформ, 2018. – 128 с. – Текст непосредственный.
3. ГОСТ 19.201-78. Единая система программной документации (ЕСПД). Техническое задание. Требования к содержанию и оформлению (с Изменением N 1) = Unified system for program documentation. Technical specifications for development. Requirements to contents and form of presentation: межгосударственный стандарт: издание официальное: утвержден и введен в действие Постановлением Государственного комитета СССР по стандартам от 18 декабря 1978 г. № 3351: введен впервые: дата введения 1980-01-01. – Москва: Стандартинформ, 2010. – 4 с. – Текст непосредственный.
4. РД 50-34.698-90. Методические указания. Информационная технология. Комплекс стандартов и руководящих документов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов: руководящий документ по стандартизации: издание официальное: утверждены и введены в действие Постановлением Государственного комитета СССР по управлению качеством продукции и стандартами от 27 декабря 1990 г. № 3380: дата введения 1992-01-01 / Разработан Министерством электротехнической промышленности и приборостроения СССР. – Москва.: ИПК Издательство стандартов, 2002 г. – 27 с. –Текст непосредственный.
5. Перлова, О.Н. Соадминистрирование баз данных и серверов: учебник для студ. учреждений среднего профессионального образования / О.Н. Перлова, О.П. Ляпина. – М.: Издательский центр «Академия», 2018. – 304 с. – (Профессиональное образование). – ISBN 978-5-4468-7075-2. – Текст: непосредственный.
6. Рудаков, А.В. Технология разработки программных продуктов: учебник для студ. учреждений среднего профессионального образования / А.В. Рудаков. – 11-е изд., стер. – М.: Издательский центр «Академия», 2017. – 208 с. – (Профессиональное образование). – ISBN 978-5-4468-4734-1. – Текст: непосредственный.
7. Семакин, И.Г. Основы алгоритмизации и программирования: учебник для студ. учреждений среднего профессионального образования / И.Г. Семакин, А.П. Шестаков. – 2-е изд., стер. – М.: Издательский центр «Академия», 2018. – 304 с. – (Профессиональное образование). – ISBN 978-5-4468-6228-3. – Текст: непосредственный.
8. Федорова, Г.Н. Основы проектирования баз данных: учебник для студ. учреждений среднего профессионального образования / Г.Н. Федорова. – 2-е изд., стер. – М.: Издательский центр «Академия», 2018. – 224 с. – (Профессиональное образование). – ISBN 978-5-4468-6170-5. – Текст: непосредственный.
9. Федорова, Г.Н. Осуществление интеграции программных модулей: учебник для студ. учреждений среднего профессионального образования / Г.Н. Федорова. – 2-е изд., стер. – М.: Издательский центр «Академия», 2018. – 288 с. – (Профессиональное образование). – ISBN 978-5-4468-6990-9. – Текст: непосредственный.
10. Федорова, Г.Н. Разработка, администрирование и защита баз данных: учебник для студ. учреждений среднего профессионального образования / Г.Н. Федорова. – 2-е изд., стер. – М.: Издательский центр «Академия», 2018. – 288 с. – (Профессиональное образование). – ISBN 978-5-4468-6994-7. – Текст: непосредственный.

Приложение А

(обязательное)

Техническое задание. Требования к программным модулям

1 Введение

Настоящее техническое задание распространяется на разработку программных модулей подсистем «Преподаватель» и «Заведующий отделением» настольного приложения анализа успеваемости обучающихся студентов в колледже, предназначенных для автоматизации рабочих мест заведующего отделением и преподавателя. Использовать данную сис­тему будут сотрудники медицинской лаборатории.

Программа «Dnevnik» предоставляет графический интерфейс для сбора, хранения, обработки и управления информацией об обучающихся в колледже, а также их успеваемости

Подобная автоматизация учебного заведения повышает эффективность, а также скорость оценивания студентов преподавателями, формирования отчетности, а также анализа успеваемости обучающихся в колледже.

2 Основания для разработки

Основанием для разработки является Договор 14 от 21.09.2020. Договор утвержден Директором ООО «Краснодарский Колледж» Николаем Александровичем Фроловым и Бережным Евгением Алексеевичем.

Наименование темы разработки – «Разработка модулей настольного приложения анализа успеваемости обучающихся в колледже»

3 Назначение разработки

Система призвана автоматизировать процесс анализа успеваемости обучающихся в колледже:

- вести учет студентов колледжа;

- вести учет успеваемости;

Пользователями системы выступают: преподаватели, заведующие отделениями и администраторы.

4 Требования к программе или программным модулям

4.1 Требования к функциональным характеристикам

После запуска программы пользователю отображается окно авторизации с формой для ввода логина и пароля.

В системе существует всего 3 типа пользователей – преподаватель, заведующий отделением и администратор. В соответствии с типом пользователя программа открывает соответствующий интерфейс

Для преподавателя программа предоставляет следующие возможности:

- просмотр успеваемости студентов;

- выставление оценок по предмету;

Для заведующего отделением программа предоставляет следующие возможности:

- просмотр и изменение информации о студенте;

- формирование отчётности;

Для администратора программа предоставляет следующие возможности:

-просмотр и редактирование списка пользователей системы

-сброс пароля пользователей;

-просмотр истории авторизации:

4.2 Требования к надежности и безопасности

Программа должна соответствовать современному уровню требований к надежности программного обеспечения:

-предусматривать целостность информации, хранящейся в базе данных;

-обеспечивать шифрование личных данных;

-обрабатывать ошибки и выдавать их пользователю

4.3 Требования к составу и параметрам технических средств

Требования к компьютеру сотрудника представлены в таблице 1.

Таблица А.1 – Требования к персональному компьютеру

|  |  |
| --- | --- |
| Компонент | Требование |
| Жесткий диск | Минимум 5 ГБ свободного места |
| Монитор | Требуется монитор с разрешением 1280х720 или более высоким |
| Оперативная память | Не менее 1 ГБ с последующим увеличением по мере роста размера базы данных |
| Быстродействие процессора | Минимум: процессор х64 с тактовой частотой 1,1 ГГц |
| Операционная система | Windows 10 |

4.4 Требования к информационной и программной совместимости

На серверах должны быть установлены ОС Windows Server 2019, сервер БД Microsoft SQL Server Express 2019, SQL Server Management Studio 18.8.

На компьютерах-рабочих станциях должна быть установлена операционная система ОС Microsoft Windows 10 Pro (Edu).

Приложение Б

(обязательное)

Программный код

Программный код представлен в контексте текстового редактора MS Visual Studio 2019 Enterprise 2019 Версия 16.9.0.



Рисунок Б.1 – Программный код разметки «Добавление студента»

Приложение В

(обязательное)

Формы выходных документов

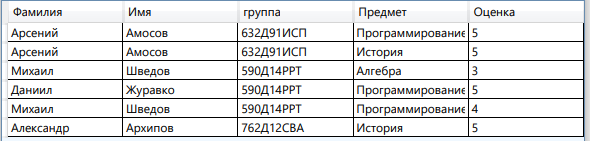
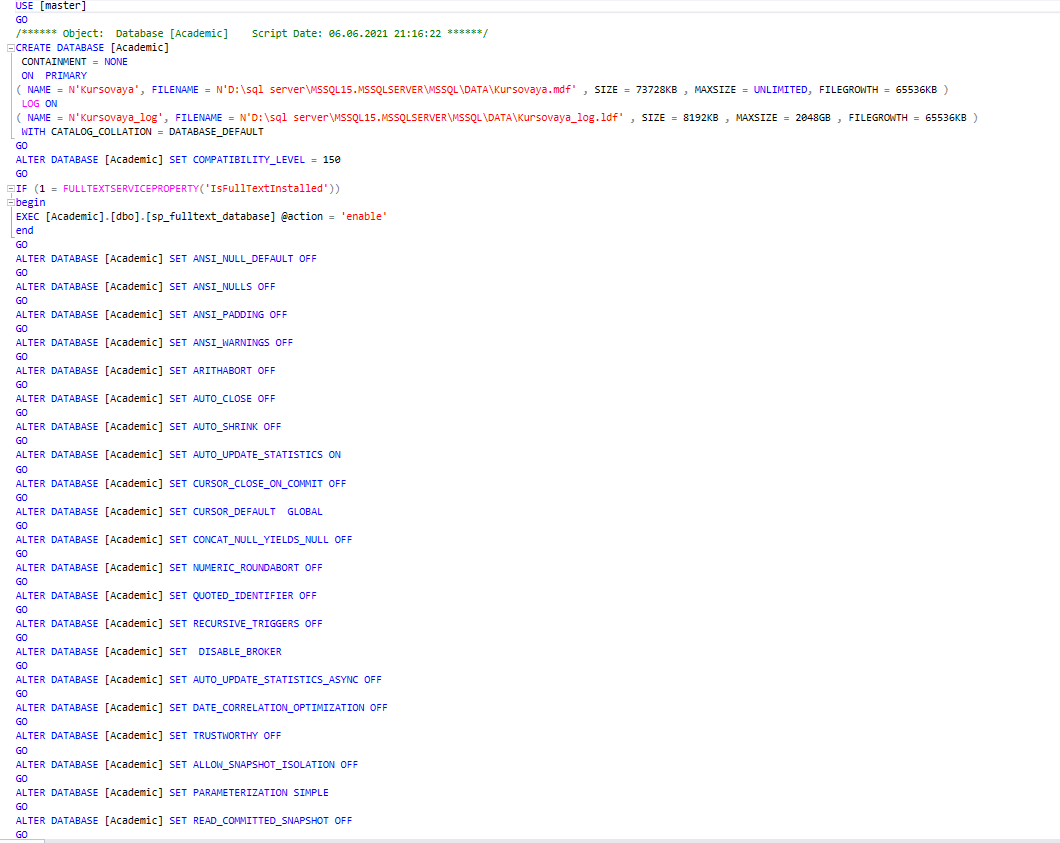


Рисунок В.1 – печатная форма результатов успеваемости в формате pdf

Приложение Г

(обязательное)

Скрипт базы данных



Приложение Д

(обязательное)

Тестирование

## Общая информация о тестировании

|  |  |
| --- | --- |
| Название проекта | Dnevnik |
| Номер версии | 1 |
| Имя тестировщика | Бережной Евгений |
| Дата тестирования | 6.06.2021 |

## Test case #1:

|  |  |
| --- | --- |
| **Test Case #** |  |
| **Приоритет теста** |  |
| **Название тестирования/Имя** |  |
| **Резюме испытания** |  |
| **Шаги тестирования** |  |
| **Данные тестирования** |  |
| **Ожидаемый результат** |  |
| **Фактический результат** |  |
| **Предпосылки** |  |
| **Постусловия** |  |