Федеральное государственное бюджетное образовательное учреждение «Сибирский государственный университет телекоммуникаций и информатики»

кафедра прикладной математики и кибернетики

Курсовая работа по дисциплине «Визуальное программирование и человеко-машинное взаимодействие»

По теме «Электронный журнал куратора»

**Выполнил(а):** студент 3 курса

Бурдуковский Илья Александрович

**Проверил(а)**: Доцент кафедры ПМиК

Мерзлякова Екатерина Юрьевна

Новосибирск 2020г.

Содержание

[Введение 3](#_Toc60255381)

[1.Часть первая 5](#_Toc60255382)

[1.1.Анализ задач и пользователей 5](#_Toc60255383)

[1.2. Выбор репрезентативных задач 6](#_Toc60255384)

[1.3. Заимствование 7](#_Toc60255385)

[1.4.Черновое описание дизайна 8](#_Toc60255386)

[2.Часть вторая 10](#_Toc60255387)

[3.Часть третья 16](#_Toc60255388)

[4.Вывод 21](#_Toc60255389)

# Введение

Объектом данной курсовой работы являются образовательные организации, предметом исследования – высшее учебное заведение СибГУТИ.

Целью курсового проекта является закрепление навыков, полученных в ходе изучения дисциплины «Визуальное программирование и человеко-машинное взаимодействие» и разработка приложения для вуза. Для достижения поставленной цели необходимо выполнение следующих задач:

* Провести первые 4 этапа проблемно-центрированного дизайна программного продукта, помогающего пользователю вести учётность подопечных студентов.
* Провести CWT-анализ разработанного интерфейса.
* Провести GOMS-анализ разработанного интерфейса.
* По результатам CWT и GOMS анализов доработать интерфейс программы и выполнить создание макета или прототипа.
* Провести анализ соответствия программы правилам Нильсена–Молиха.

Структура курсовой работы включает: введение, четыре части, заключение, список литературы.

Во введении обоснована актуальность темы работы, сформулированы объект, предмет, цель и задачи, кратко раскрыто содержание.

Первая часть посвящена проблемно-центрированной разработки интерфейса, что является одним из наиболее эффективных подходов к разработке интерфейса с пользователем, предлагаемых в литературе по человеко-машинному взаимодействию. Это подход, сфокусированный на задачах, которые нужно решать пользователю. При таком подходе процесс разработки структурируется исходя из специфических задач, которые пользователь должен будет решать с помощью разрабатываемой системы. Эти задачи выбираются на ранней стадии разработки, затем они используются для выявления требований к дизайну, чтобы облегчить выработку решений и их оценку по мере развития проекта.

Во второй части CWT анализ – это формализованный способ представления мыслей и действий людей, когда они пользуются интерфейсом в первый раз. CWT- анализ позволит обнаружить несколько типов проблем с интерфейсом:

1. Поставить под сомнение ваши первоначальные и не вполне обоснованные предположения о том, как мыслит пользователь.
2. Выявлять элементы управления, которые очевидны для разработчика, но могут быть непонятны пользователю.
3. Выявлять затруднения с надписями и подсказками.
4. Обнаруживать неадекватную обратную связь, что может заставить пользователя сомневаться в результате и повторять всё с начала, хотя всё было сделано правильно.
5. Показывать недостатки в текущем описании интерфейса.

В третьей части GOMS анализ сможет оценить время работы с интерфейсом пользователя. Даже если интерфейс успешно прошел CWT-анализ, это не означает, что он оптимален с точки зрения трудоёмкости. Если есть несколько альтернативных вариантов построения интерфейса, то анализ GOMS позволит выбрать тот из них, который требует меньше времени для решения задачи пользователя.

В четвёртой части проанализируем соответствия программы правилам Нильсена Молиха. За годы изучения проблем человеко-машинного взаимодействия, специалисты выявили несколько наиболее существенных правил построения интерфейсов, и назвали их "золотыми правилами". Эти правила могут также использоваться для экспертной оценки существующих интерфейсов.

Заключение содержит основные выводы и рекомендации по работе.

# Часть первая

* 1. **Анализ задач и пользователей**

Разрабатываемое приложение собираются использовать квалифицированные кураторы студенческих групп — Юрий Стояк и Анастасия Захарова. Они являются кураторами в вузе.

Юрий Стояк учится в Сибирском государственном университете телекоммуникаций и информатики на факультете информатики и вычислительной техники. В 2019 году в возрасте 19 лет решил курировать группы первокурсников и делает это до сих пор. Навыки очень многогранны: знания нескольких языков, методологий разработки ПО, взаимодействие со студентами. Уверенный пользователь ПК (MS PowerPoint, Word, Excel, Visual Studio 2019, PyCharm).

Анастасия Захарова – тоже студентка Сибирского Государственного Университета Телекоммуникаций и Информатики. На данный момент в возрасте 20 лет, является заслуженным куратором вот уже 2 года. Профессиональные навыки: знание архитектуры вычислительных систем, языков программирования и методологий визуального программирования. Профессиональный пользователь ПК

Одной из важных задач при разработке приложения для куратора– это простой и понятный интерфейс, основной набор нужных компонентов для ведения учётности по студентам.  
Везде начинают пользоваться электронными журналами. Поэтому им необходимо удобное и простое приложения для контроля.

## 1.2. Выбор репрезентативных задач

Образом для составления электронного приложения служит сам журнал. Поэтому куратору необходимо универсальное приложение для выставления оценок студентам и для контроля их успеваемости

* Основная информация о студенте.
* Название предмета
* Даты занятий.
* Выставление оценок/пропусков/дополнительных рекомендаций.
* Выбор старосты среди группы
* Пометка отстающего студента

Задачи, которые будет поддерживать разрабатываемая программа:

* Возможность добавления даты занятий, оценок, информации о студенте (фотография), самих студентов.
* Возможность вывода отчёта успеваемости.

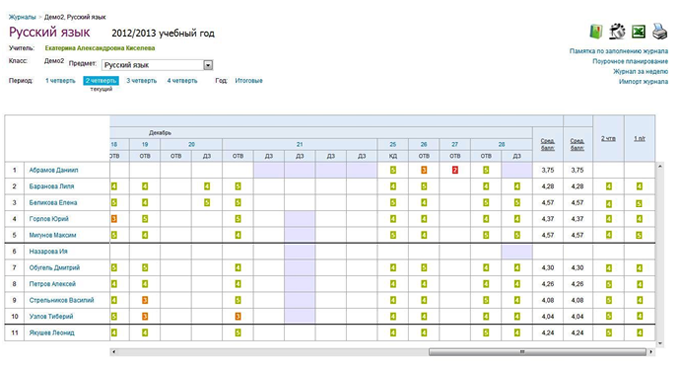
База данных: студент, группа, дата, оценка, список мероприятий, маркер отстающего с возможностью добавления, удаления, редактирования; статус старосты с возможностью смены.

**1.3. Заимствование**

Рассмотрим функционал и интерфейс сайта электронного журнала «**Дневник.ру**».

Этот сайт является реализацией электронного школьного журнала.

Дневник ру предлагает поставить оценку учащимся, отметить их отсутствие, оценить их домашнее задание.



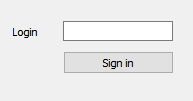
*Рис 1.Интерфейс приложения “Дневник Ру”*

Из данного приложения можно заимствовать общую структуру составления журнала, пункты ученики, номер ученика, дата занятия, итоговый балл.

**1.4.Черновое описание дизайна**

На основе раннее упомянутой программы я набросал черновой вариант дизайна «Электронного журнала куратора».

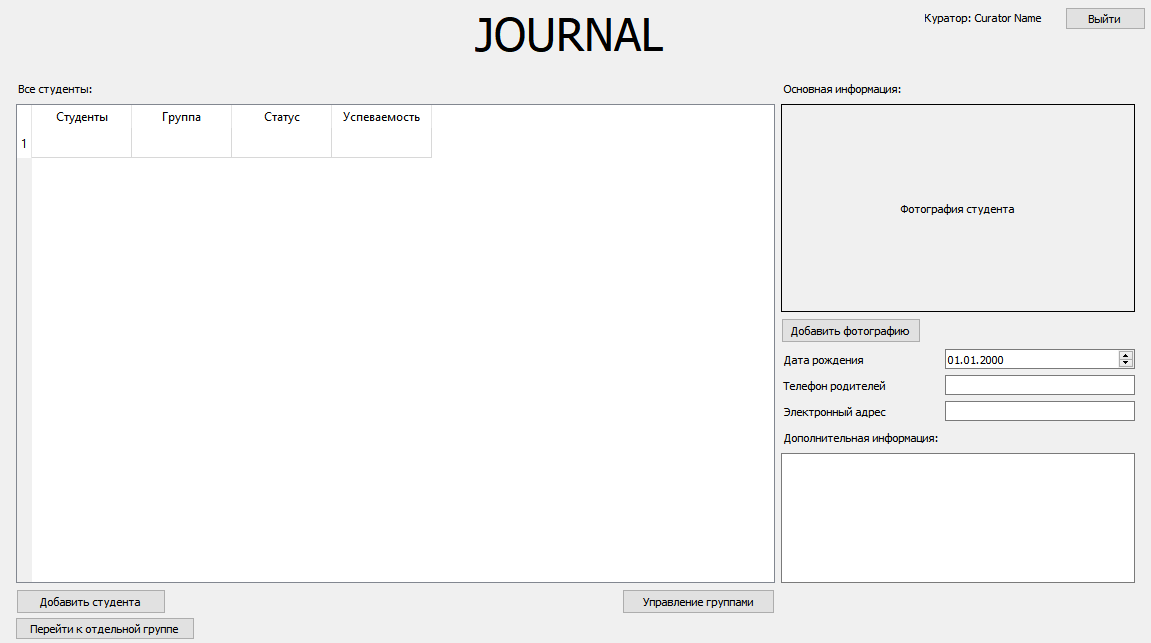
Приложение имеет вид окна со сменяемыми в нём виджетами. В зависимости от текущего виджета меняется и функционал.



*Рис.2 Черновой дизайн приложения, окно входа*

(простой/со стилем css)

После входа в приложение пользователям будет выводится список всех студентов и основная информация о них, с возможность её редактирования



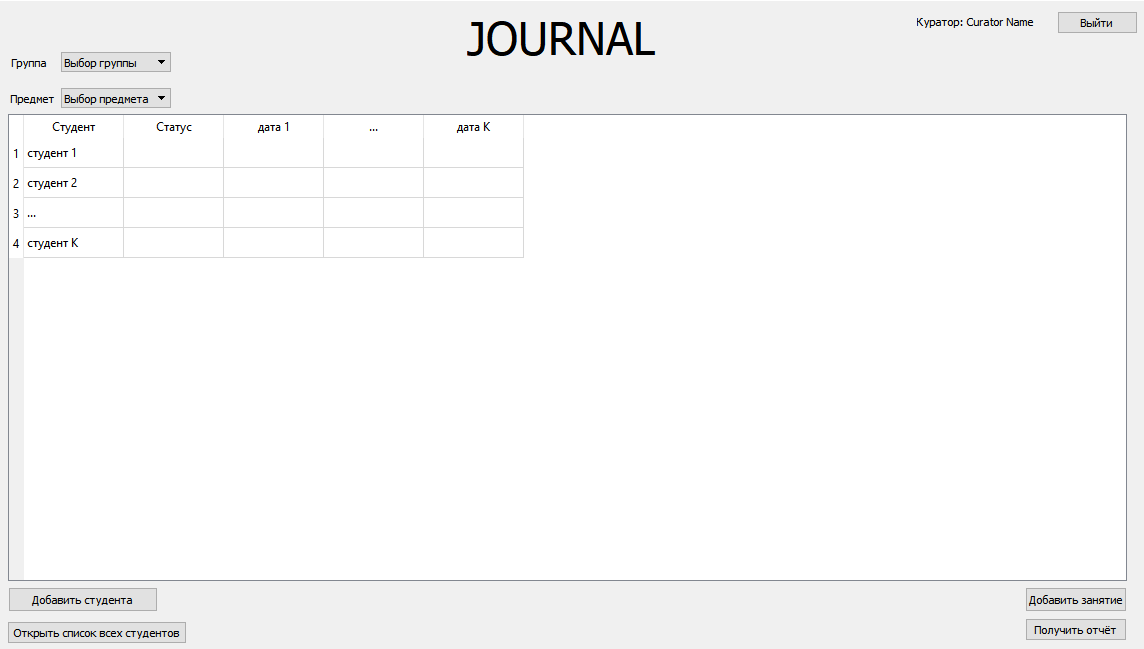


*Рис.3 Черновой дизайн приложения, главное окно информации о всех студентах* (простой/со стилем css)

Выбрав пункт «Добавить студента» в таблице должна появляться новая строка, которая в дальнейшем будет хранить информацию о студенте.

В пункте «Управление группами» будет добавляться, изменяться, удаляться группа, к которой может причисляться студент.

Если выбрать пункт «Перейти к отдельной группе», то виджет меняется на новый с информацией об успеваемости конкретной группы по конкретному предмету.





*Рис.4 Черновой дизайн приложения, окно информации о конкретной группе студентов* (простой/со стилем css)

С помощью уже знакомого пункта «Добавить студента» будет возможно добавить нового студента, автоматические причисляя его к текущей просматриваемой группе.

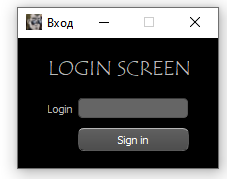
Кнопка «Добавить занятие» позволит куратору вставить новый столбец для предстоящего занятия. А «Получить отчёт» - подвести итоги прошедших занятий и увидеть средний балл каждого из учащегося.

# 2. Часть вторая

Здесь представлены скриншоты работы приложения с подробными пояснениями к рисункам.

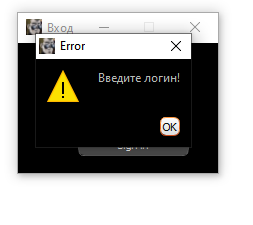


*Рис.5 Загрузка приложения*

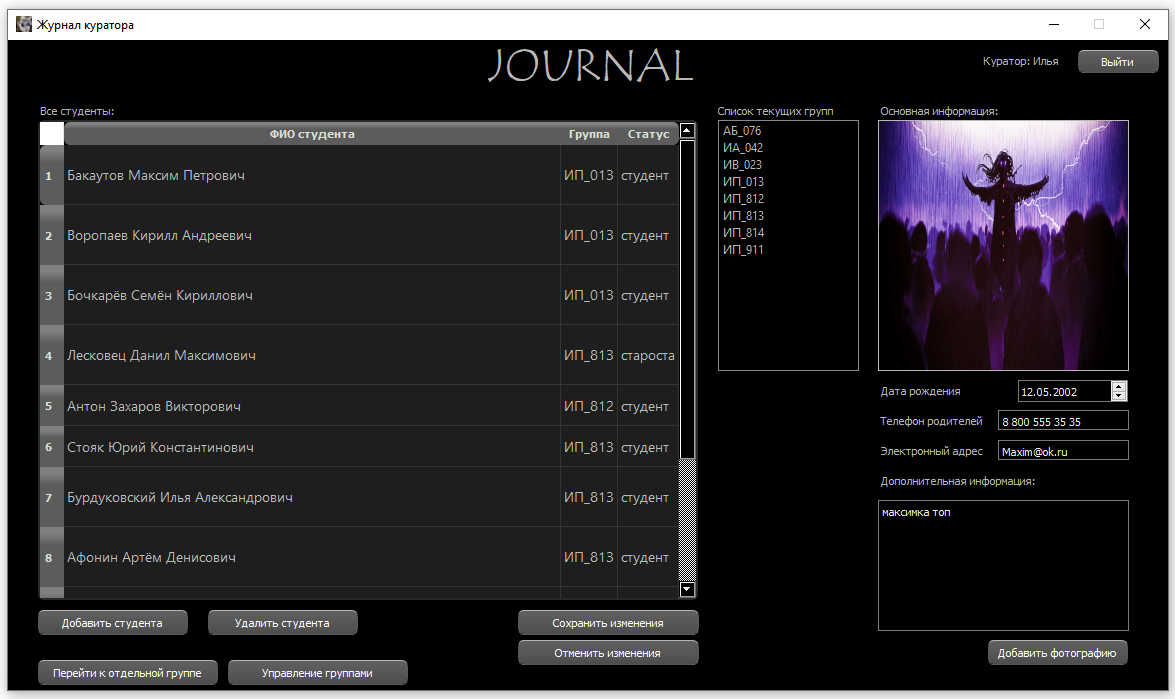


*Рис.6 Окно входа в приложение*

На рисунке 6 изображён стартовое окно после запуска программы. На нём, в поле ввода, мы можем указать желательное имя пользователя и подтвердит вход. Если имя пользователя не указано, программа попросит его указать.



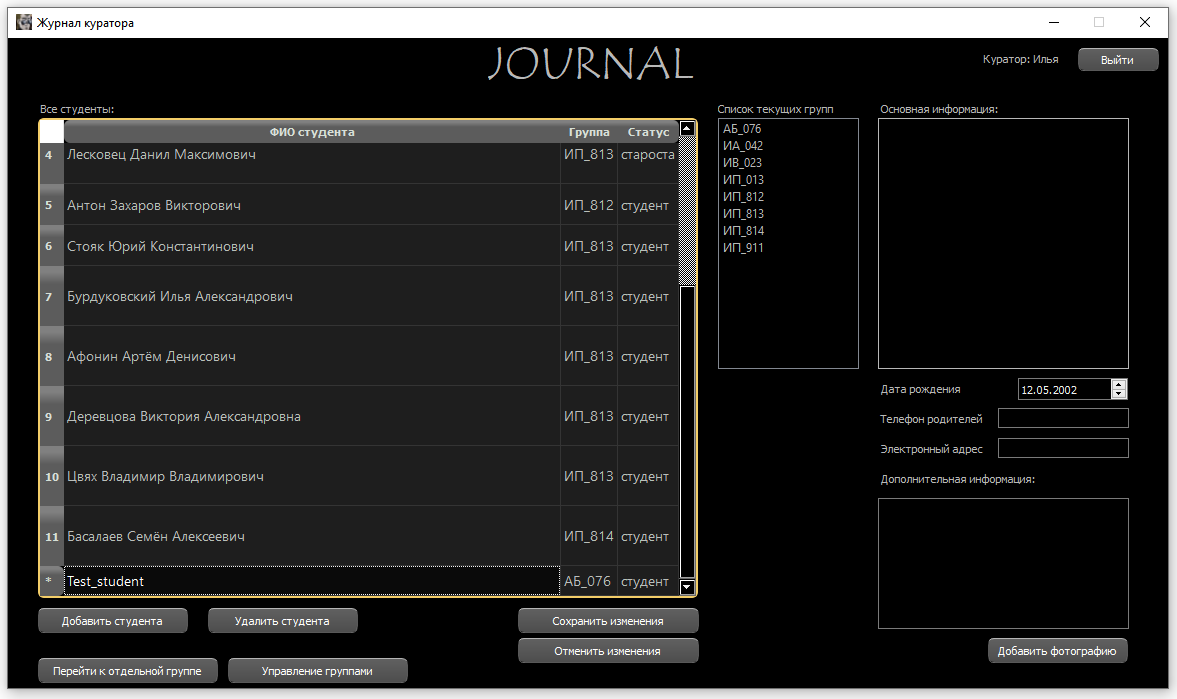
*Рис.7 Предупреждение, если имя пользователя не указано.*



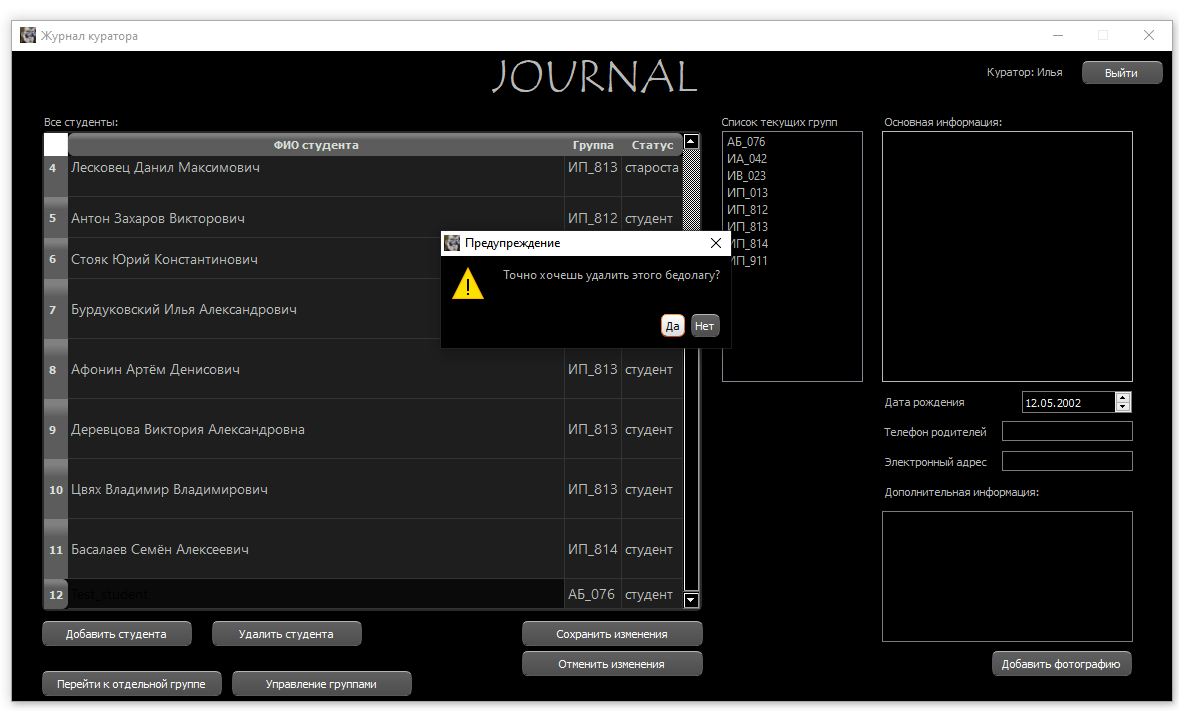
*Рис.8 Главная страница журнала куратора со списком всех студентов и информации о них.*

На рисунке 8 мы можем наблюдать главное окно. Интерфейс окна:

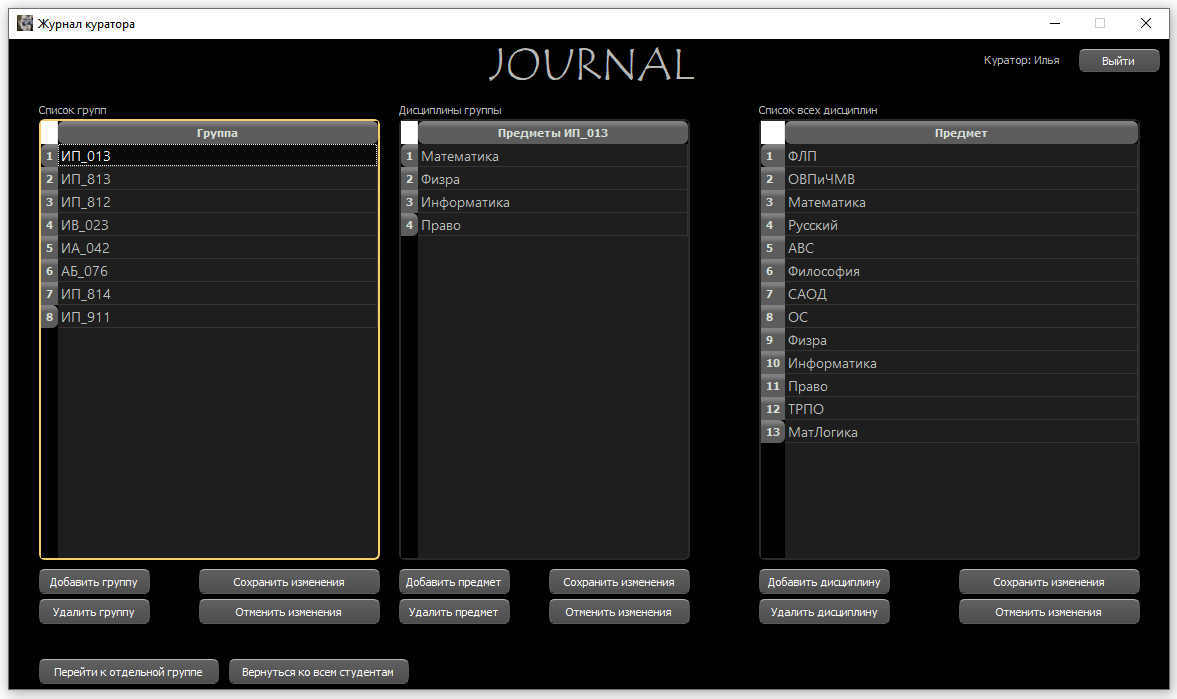
* Таблица всех студентов. В данной таблице мы можем увидеть, ФИО, группу и статус студента.
* Информация о выбранном студенте. Кликая по нужному студенту в таблице, в правой части окна отображается информация о нём, которую можно редактировать
* Добавить студента. Добавление новой ячейки, которую можно заполнить. Чтобы студент добавился, нужно сохранить изменения, нажав на соответствующий пункт.
* Удалить студента. Удаление студента, который выбран в таблице.
* Перейти к отдельной группе. Переход на другую страницу с выбором таблицы выбранного занятия для выбранной группы.
* Управление группами. Переход на другую страницу с таблицами всех групп, всех предметов, и выбранных предметов для групп.



*Рис.9 Добавление нового студента*

***.***

*Рис.10 Удаление выбранного студента*



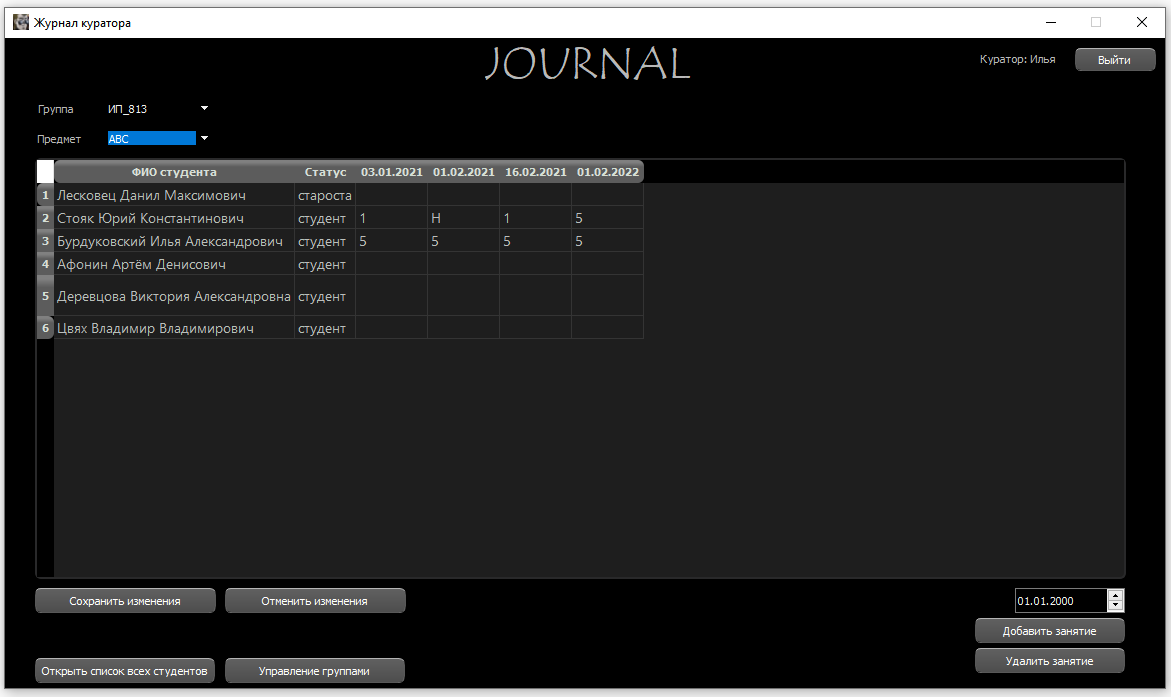
*Рис.11 Окно информации о группах и предметах*

На рисунке 11 изображена страница управления группами и предметами. Интерфейс окна:

* Таблица дисциплин учебного заведения. В ней можем увидеть список всех текущих дисциплин. При необходимости можно добавить или удалить нужный предмет, используя соответственно-наименованные кнопки интерфейса.
* Таблица групп. В ней можем увидеть список всех групп, в которых могут находится студенты. При необходимости можно добавить или удалить нужную группу, используя соответственно-наименованные кнопки интерфейса.
* Таблица выбранных для группы предметов. После клика на нужную группу в таблице групп, данная таблица выводит все предметы, которые преподаются данной группе. Если нужно, то используя соответствующие кнопки, в список можно добавить или удалить имеющийся в списке всех дисциплин вуза предмет.



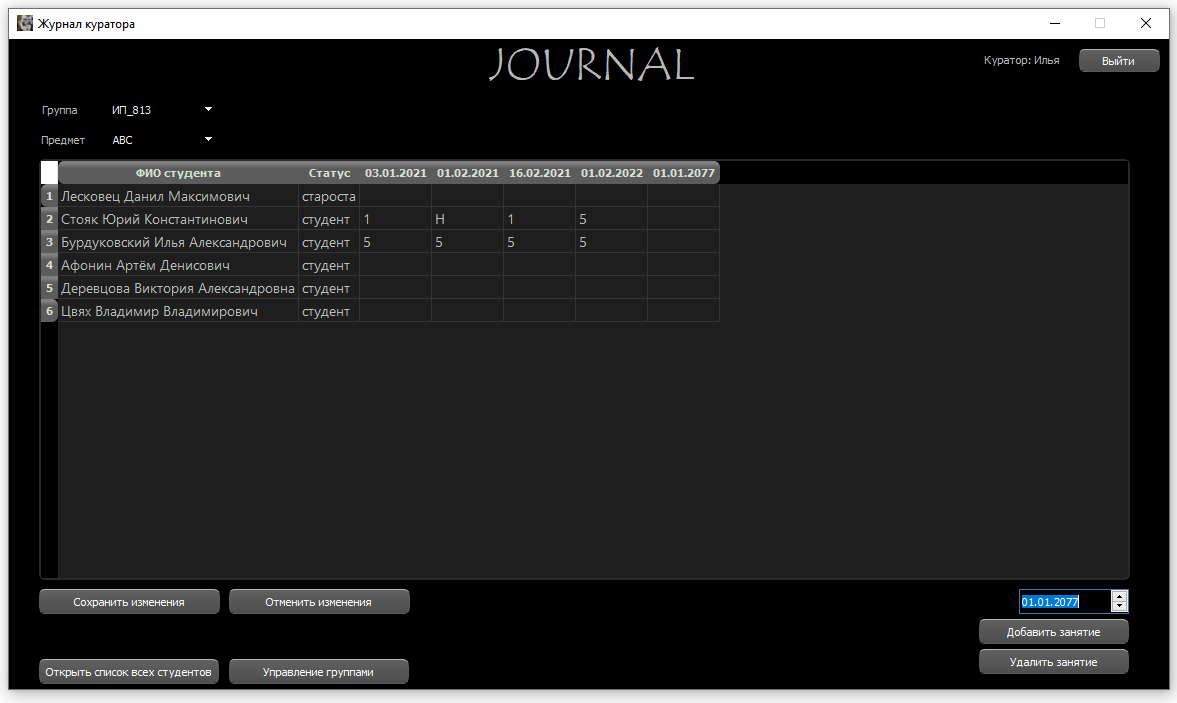
*Рис.12 Окно занятий*



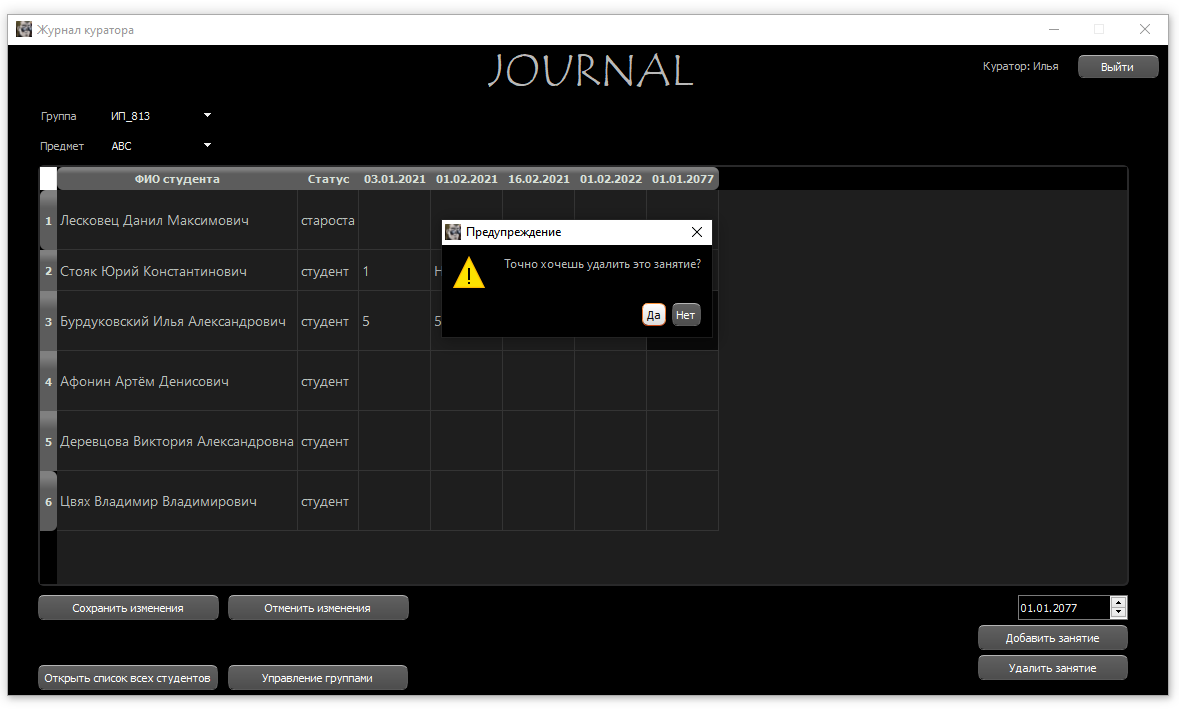
*Рис.13 Выбор нужного занятия*

На данной странице мы можем ознакомиться с таблицами занятий. Интерфейс окна:

* Выбор группы и предмета. В зависимости от выбранной группы меняется список доступных для выбора предметов.
* Таблица занятия. После выбора предмета в окне отображается нужная таблица, с возможностью добавлять новые даты занятий и редактировать их ячейки.
* Добавление занятия. Выбрав нужную дату в нижнем, правом углу, можно добавить колонку с занятием в эту дату.
* Удаление занятия. Выбрав не нужную колонку занятия в таблице, её можно удалить с помощью кнопки «Удалить занятие»



*Рис.14 Добавление занятия 1го января 2077года*



*Рис.15 Удаление занятия 1го января 2077года*

# 3. Часть третья

**3.1. СWT –анализ**

В программе «Электронный журнал куратора» выберем две задачи, поддерживаемые интерфейсом приложения:

1. Создать новую группу.
2. Добавить фотографию в информацию о студенте

Анализ первой задачи:

|  |  |  |
| --- | --- | --- |
| № | Действие | Анализ |
| 1 | Запуск приложения | Пользователю нужно совершить двойной клик по ярлыку приложения. |
| 2 | Вход в приложение | Пользователю нужно кликнуть по полю для заполнения. Ввести желаемое имя пользователя и нажать на кнопку “Sign in”. После выполнения откроется главное окно приложения. |
| 3 | Переход к окну управления групп | Пользователь должен нажать на пункт «Управление группами». После этого главное окно приложения сменится на окно управления группами. |
| 4 | Добавление записи | Пользователь нажимает на кнопку «Добавить группу» и в таблице появляется новая строка для редактирования. Пользователь нажимает по этой строке и вводит нужную информацию.  Чтобы сохранить запись пользователь должен нажать на кнопку «Сохранить изменения» НЕДОСТАТКИ:   1. Пользователю может быть не очевидно, что после ввода нужной группы необходимо дополнительно нажимать «Сохранить изменения». Если этого не сделать, запись пропадёт при переходи в другое окно. 2. Пользователь не осведомляется, что запись сохранилась, т.к. ничего сообщающего о сохранении не появляется. |

Анализ второй задачи:

|  |  |  |
| --- | --- | --- |
| № | Действие | Анализ |
| 1 | Запуск приложения | Пользователю нужно совершить двойной клик по ярлыку приложения. |
| 2 | Вход в приложение | Пользователю нужно кликнуть по полю для заполнения. Ввести желаемое имя пользователя и нажать на кнопку “Sign in”. После выполнения откроется главное окно приложения. |
| 3 | Выбор нужного студента | Пользователь выбирает нужного студента в таблице всех студентов и в правой части окна появляется информация о нём. |
| 4 | Добавление фотографии | Пользователь нажимает на кнопку «Добавить фотографию» и открывается проводник компьютера. Пользователь должен перейти по папкам к нужной фотографии, выбрать ей и нажать “Открыть”.  После этого в информацию о пользователе запишется путь к выбранной картинке.  Чтобы сохранить добавление/изменение фотографии пользователь должен нажать на кнопку «Сохранить изменения» НЕДОСТАТКИ:   1. Пользователю может быть не очевидно, что после выбора фотографии необходимо дополнительно нажимать «Сохранить изменения». Если этого не сделать, информация о картинке пропадёт, после выбора другого студента. 2. Пользователь не осведомляется, что запись сохранилась, т.к. ничего сообщающего о сохранении не появляется. 3. Диалоговое окно проводника открывается из папки, где находится само приложение, а не из корневого диска. 4. В таблице хранится путь к картинке, а не сама картинка поэтому, если фотографию переименуют, переместят, или удалят, то в информации о студенте она не отобразится. |

Вывод: общая идея интерфейса понятна и интуитивна, но у пользователя могут возникать сложности, связанные с редактированием списков.

**3.2. GOMS –анализ**

В программе «Электронный журнал куратора» выберем две задачи, поддерживаемые интерфейсом приложения:

1. Открыть информацию о занятии группы и добавить новое занятие.
2. Удалить предмет из списка вузовских дисциплин.

Анализ первой задачи:

Для выполнения первой задачи сформируем подцели:

1. Войти в приложение
2. Перейти в нужное окно
3. Выбрать группу
4. Выбрать предмет
5. Добавить занятие

Опишем методы для каждой подцели:

1. Войти в приложение
   1. Кликнуть по полю рядом с текстом «Login». - (МРВ)
   2. Ввести имя пользователя с клавиатуры (Например, «Илья»). – (МН4К)
   3. Кликнуть по кнопке «Sign in». - (МHPB)
2. Перейти в нужное окно
   1. Кликнуть по кнопке «Перейти к отдельной группе». - (МРВ)
3. Выбрать группу
   1. Кликнуть на кнопку раскрывающегося списка с надписью «Выбор группы». - (МРВ)
   2. Из вывалившегося списка выбрать группу кликнув на неё. – (МРВ)
4. Выбрать группу
   1. Кликнуть на кнопку раскрывающегося списка с надписью «Выбор предмета». – (МРВ)
   2. Из вывалившегося списка выбрать предмет кликнув на неё. – (МРВ)
5. Добавить занятие
   1. Кликнуть на поле даты в правом нижнем углу. – (МРВ)
   2. Ввести нужную дату с клавиатуры (Например, «01.01.2077»). – (МН10К)
   3. Кликнуть по кнопке «Добавить занятие». – (МНРВ)

В результате получим следующую последовательность операций:

МРВМНККККМНРВМРВМРВМРВМРВМРВМРВМНККККККККККМНРВ

Просуммируем время отдельных операций:

11М = 1,35 \* 11 = 14,85

9Р = 1,1 \* 9 = 9,9

9В = 0,2 \* 9 = 1,8

4Н = 0,4 \* 4 = 1,6

14К = 0,2 \* 14 = 2,8

Оценка среднего времени решения первой задачи равна 30,95

Анализ второй задачи:

Для выполнения первой задачи сформируем подцели:

1. Войти в приложение
2. Перейти в нужное окно
3. Выбрать дисциплину
4. Удалить дисциплину

Опишем методы для каждой подцели:

1. Войти в приложение
   1. Кликнуть по полю рядом с текстом «Login». - (МРВ)
   2. Ввести имя пользователя с клавиатуры (Например, «Илья»). – (МН4К)
   3. Кликнуть по кнопке «Sign in». - (МHPB)
2. Перейти в нужное окно
   1. Кликнуть по кнопке «Управление группами». - (МРВ)
3. Выбрать дисциплину
   1. Нажать на нужную ячейку в таблице всех дисциплин. – (МРВ)
4. Удалить дисциплину
   1. Кликнуть на кнопку «Удалить дисциплину». – (МРВ)
   2. Подтвердить удаление нажав на кнопку «Да» в возникшем диалоговом окне. – (МРВ)

В результате получим следующую последовательность операций:

МРВМНККККМНРВМРВМРВМРВМРВ

Просуммируем время отдельных операций:

7М = 1,35 \* 7 = 9,45

6Р = 1,1 \* 6 = 6,6

6В = 0,2 \* 6 = 1,2

2Н = 0,4 \* 2 = 0,8

4К = 0,2 \* 4 = 0,8

Оценка среднего времени решения второй задачи равна 18,85

**3.3. Анализ соответствия интерфейса правилам Нильсена-Молиха:**

1. Простой и естественный диалог: Вся информация в приложении не является редко используемой. Интерфейс не имеет сложной структуры и выдаёт то, чего ожидает пользователь.
2. Говорите на языке пользователя: Все термины, используемые в приложении, не являются узконаправленными и известны пользователю.
3. Минимизируйте загрузку памяти пользователя: В приложении присутствует только одно основное окно, с меняющимися страницами. В приложении интуитивно понятно, что необходимо сделать пользователю для достижения его целей.
4. Будьте последовательны: У пользователей имеется возможность изучить действия в одной части программы и применить их снова, чтобы получить похожие результаты в других местах.
5. Обеспечьте обратную связь: В программе отсутствуют предупреждения о:
   1. Сохранении файла.
   2. Отмене изменений.
6. Обеспечьте хорошо обозначенные выходы: На всех окнах присутствует пункты, позволяющие перейти к нужному окну. Также есть возможность выйти из учётной записи приложения или просто закрыть само приложение.
7. Обеспечьте быстрые клавиши и ярлыки: В программе отсутствует возможность использования быстрых клавиш.
8. Хорошие сообщения об ошибках: Все сообщения об ошибках информативны, чтобы пользователь понял суть ошибки.
9. Предотвращайте ошибки: Пользователь не сможет добавить студента с группой, которой не существует. Пользователь не сможет удалить нужное поле таблицы, если не выберет его.
10. Снабдите программу системой помощи: Приложение не снабжена справкой из-за наличия информативного интерфейса и детальных предупреждающих диалоговых окон.

После проведение CWT-анализа, GOMS-анализа и анализа соответствия юзабилити интерфейса по правилам Нильсена-Молиха, были выявлены следующие недостатки:

1. Для пользователя не всегда будет очевидно, что после редактирования ячеек и/или добавление новой записи необходимо подтверждать свои изменения кнопкой «Сохранить изменения».
2. Пользователь не может быть уверен в сохранности своих записей из-за отсутствия уведомлений о сохранении.
3. Т.к. диалоговое окно выбора фотографии открывается из папки с приложением, пользователь будет вынужден совершать дополнительные переходы по папкам.
4. Из-за того, что в информацию о студенте сохраняется путь до фотографии, любое действие с фотографией на компьютере может повлечь её исчезновение из личного дела, но имеется возможность её восстановить, вернув на прежнее место старую или подставив на место старой –новую.

# 4. Вывод

В результате выполнения данной курсовой работы я познакомился с работой SQL на QT, использовал свои навыки работы в среде QT Creator для создания многооконного приложения, справки о приложении и обмена данными между окнами.

Также были проведены соответствующие анализы интерфейса разработанного приложения, что помогло увидеть некоторые недостатки конечного приложения. Было определено насколько удобна и понятна программа для конечного пользователя.