МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Минский КОЛЛЕДЖ предпринимательства

Отделение «Программное обеспечение информационных технологий»

Пояснительная записка

к курсовому проекту

на тему

**Сайт «школа искусств»**

МКП КП 2-40 01 01 33 010 ПЗ

**Автор проекта**

Учащийся 4 курса группы П-41 П.Е. Исаева

**Руководитель**

Преподаватель И.Ю. Дроздова

Минск, 2018

**СОДЕРЖАНИЕ**

*у*

Изм.

Лист

№ докум.

Подпись

Дата

Лист

3

МКП КП 2-40 01 01 33 010 ПЗ

Разраб.

Исаева

Пров.

*Дроздова*

Реценз.

Н. контр.

Утв.

***Сайт «Школа искусств»***

Лит.

Листов

60

МКП

[Введение 5](file:///E:\Курсачи\ГАИ%202016\ПЗ\ПЗ.doc#_Toc470122028)

[1 Постановка задачи 7](file:///E:\Курсачи\ГАИ%202016\ПЗ\ПЗ.doc#_Toc470122029)

[1.1 Формулировка задачи 7](file:///E:\Курсачи\ГАИ%202016\ПЗ\ПЗ.doc#_Toc470122030)

[1.2 Определение данных и их представление 8](file:///E:\Курсачи\ГАИ%202016\ПЗ\ПЗ.doc#_Toc470122031)

[2 Вычислительная система 9](file:///E:\Курсачи\ГАИ%202016\ПЗ\ПЗ.doc#_Toc470122032)

[2.1 Обоснование выбора языка программирования 9](file:///E:\Курсачи\ГАИ%202016\ПЗ\ПЗ.doc#_Toc470122033)

[2.2 Обоснование выбора среды разработки 9](file:///E:\Курсачи\ГАИ%202016\ПЗ\ПЗ.doc#_Toc470122034)

[2.3 Требования к конфигурации программного и аппаратного обеспечения 11](file:///E:\Курсачи\ГАИ%202016\ПЗ\ПЗ.doc#_Toc470122035)

[3 Проектирование 12](file:///E:\Курсачи\ГАИ%202016\ПЗ\ПЗ.doc#_Toc470122036)

[3.1 Проектирование интерфейса 12](file:///E:\Курсачи\ГАИ%202016\ПЗ\ПЗ.doc#_Toc470122037)

[3.2 Инфологическая модель предметной области 13](file:///E:\Курсачи\ГАИ%202016\ПЗ\ПЗ.doc#_Toc470122038)

[3.3 Логическая модель данных 14](file:///E:\Курсачи\ГАИ%202016\ПЗ\ПЗ.doc#_Toc470122039)

[3.4 Физическая модель базы данных 16](file:///E:\Курсачи\ГАИ%202016\ПЗ\ПЗ.doc#_Toc470122040)

[4 Описание программы 17](file:///E:\Курсачи\ГАИ%202016\ПЗ\ПЗ.doc#_Toc470122041)

[4.1 Логическая структура 18](file:///E:\Курсачи\ГАИ%202016\ПЗ\ПЗ.doc#_Toc470122042)

[4.2 Физическая структура 20](file:///E:\Курсачи\ГАИ%202016\ПЗ\ПЗ.doc#_Toc470122043)

[4.3 Особенности интерфейса 21](file:///E:\Курсачи\ГАИ%202016\ПЗ\ПЗ.doc#_Toc470122044)

[5 Испытание и отладка программы 24](file:///E:\Курсачи\ГАИ%202016\ПЗ\ПЗ.doc#_Toc470122045)

[5.1 Тестовые примеры 24](file:///E:\Курсачи\ГАИ%202016\ПЗ\ПЗ.doc#_Toc470122046)

[5.2 Анализ полученных результатов 24](file:///E:\Курсачи\ГАИ%202016\ПЗ\ПЗ.doc#_Toc470122047)

Изм.

Лист

№ докум.

Подпись

Дата

МКП КП 2-40 01 01 33 010 ПЗ

Лист

4

[6 Описание применения программы 27](#_Toc470122048)

6.1 Назначение программы 27

6.2 Условия применения 27

[Заключение 29](#_Toc470122051)

[Список использованных источников 30](#_Toc470122052)

[Приложение А 31](#_Toc470122053)

**Введение**

Основные идеи современной информационной технологии базируются на концепции, согласно которой данные должны быть организованы в базы данных с целью адекватного отображения изменяющегося реального мира и удовлетворения информационных потребностей пользователей. Эти базы данных создаются и функционируют под управлением специальных программных комплексов, называемых системами управления базами данных (СУБД).

Увеличение объема и структурной сложности хранимых данных, расширение круга пользователей информационных систем привели к широкому распространению наиболее удобных и сравнительно простых для понимания реляционных (табличных) СУБД. Для обеспечения одновременного доступа к данным множества пользователей, нередко расположенных достаточно далеко друг от друга и от места хранения баз данных, созданы сетевые мультипользовательские версии БД, основанных на реляционной структуре. В них тем или иным путем решаются специфические проблемы параллельных процессов, целостности (правильности) и безопасности данных, а также санкционирования доступа.

В базе данных школа искусств необходимо хранить информацию об учащихся, преподавателях и вести их статистику.

Таким образом, необходима автоматизация работы сайта школы искусств. Для этого должна быть создана информационная система, основой которой будет база данных. В курсовой работе будет исследован сайт школы искусств и разработана примерная база данных. Данная пояснительная записка содержит следующие разделы:

1. Раздел «Постановка задачи» содержит формулировку задачи с указанием всех функций проекта, определение круга задач, которые должны быть автоматизированы в данном курсовом проекте, описание входных и выходных данных проекта.
2. Раздел «Вычислительная система» содержит обоснование выбора языка программирования и среды разработки проекта, описание операционной системы, в которой была реализована программа. Требования конфигурации программного и аппаратного обеспечения.
3. Раздел «Проектирование» содержит описание внешнего пользовательского интерфейса, структуру меню, алгоритма, функционально независимых подзадач и путей их решения.
4. В разделе «Описание программы» содержится описание всех модулей, функций, входящих в проект, их назначение и взаимодействие, особенности интерфейса программы.
5. В разделе «Отладка и испытание программы» описывается проверка работы программы на различных тестах и анализ полученных результатов.
6. Раздел «Описание и применение программы» содержит сведения о назначении программного средства, области применения, требования к необходимым программным и техническим ресурсам.
7. В «Заключении» описывается краткая формулировка проблемы и пути ее решения, использованные методы и средства, возможность дальнейшей модификации проекта.

**1 ПОСТАНОВКА ЗАДАЧИ**

**1.1 Формулировка задания**

Разработать информационный сайт школы искусств. Спроектировать и разработать базу данных для хранения информации о существующих дисциплинах и их расписании. Реализовать регистрацию и авторизацию на сайте. Предусмотреть следующие роли: «Преподаватель», «Ученик».

Для ученика предусмотреть следующие функции:

* просмотр расписания занятий;
* запись на занятия в группы;
* отмена личной записи на занятия;
* поиск по дисциплинам;
* поиск по предметам;
* просмотр информации сайта.

Для преподавателей реализовать следующие функции:

* редактирование собственного расписания занятий;
* просмотр списка записавшихся учащихся;
* контроль списка записавшихся учеников;
* просмотр кабинетов;
* просмотр собственного расписания.

На данном сайте реализовать поиск по дисциплинам и предметам, который будет зависеть от действий зарегистрированного и не зарегистрированного пользователя. При результатах поиска перед пользователем предусмотреть вывод перечня дисциплин и предметов, на которые можно записаться, а также номера групп. В поиске реализовать фильтр по дате, алфавиту и направлению. Для каждого предмета создать кнопку «Записаться», которая переходит на страницу с записью на занятие. Для записи на занятия, пользователь должен пройти авторизацию или регистрацию. У пользователей создать возможность записаться на занятие, или, наоборот, отменить запись.

Для преподавателя разрешить изменение своего расписания занятий, просматривать количество записанных, именно к нему, учеников, а также просматривать составы групп на занятия. Также только для преподавателей реализовать просмотр личной информации записанных на занятия учеников, которая выводиться в виде графика и таблицы.

**1.2 Обоснование необходимости автоматизации**

Школа искусств – этомногопрофильное учреждение дополнительного образования детей в возрасте от 7 до 16 лет, где занимаются в разнообразных объединениях.

Сайт – это совокупность логически связанных между собой веб-страниц. Создание сайтов на сегодняшний день, становится одной из наиболее актуальных и востребованных услуг. Именно поэтому, большинство компаний уже оценили все преимущества такого предложения как создание сайтов и позаботились о разработке подходящего ресурса.

Найти в сети на данный момент можно практически все. Порой возникают ситуации, когда через интернет просмотреть информацию легче, нежели выбираться из дома и ехать куда-то. Если коснуться теоретической основы, то сайт предоставляется собой электронную версию информационного стенда, целью которого являются ознакомление пользователей с существующей информацией, а также запись на занятия онлайн посредством сети интернет. Пользователям предоставляется возможность в режиме реального времени, не выходя из дома выбрать понравившийся предмет и записаться на занятие.

На сегодняшний день многие пользователи сети стремятся самостоятельно создать сайт, и таким образом поправить своё материальное положение. Однако даже в таком внешне простом, выгодном и привлекательном деле есть свои подводные камни. Даже для того, чтобы просто создать сайт с нуля, необходим опыт и время. Безусловно, в глобальной сети существует масса программ и порталов, предлагающих бесплатную и самостоятельную разработку проекта.

Если сравнивать данный курсовой проект с сайтом schoolist.com, то можно заметить, что в курсовом проекте реализован более простой и понятный пользователю интерфейс, облегчен поиск информации, реализованы функции онлайн записи на занятия.

На основании вышеперечисленного, можно считать, что данный проект должен быть реализован.

**1.3 Определение данных и их представление**

Форматы входных и выходных данных являются частью этапа проектирования. Входные форматы должны быть разработаны с учетом максимального удобства для пользователя и минимальной возможности ошибок. Порядок переменных и форматы данных, привычные для пользователя, помогут избежать ошибок и облегчат использование программ. В данном проекте входными данными являются: данные базы данных, ввод личной информации при записи на занятие.

Выходные спецификации могут сильно различаться. Иногда даются четкие инструкции, и выходные данные подгоняются под определенный стандарт. Однако часто вообще отсутствуют какие-либо указания. Выходные данные подчас представляют собой: результаты сортировки и фильтрации, результаты запросов, списки групп учащихся.

**2 ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА**

## 2.1 Обоснование выбора языка программирования

Для курсового проекта выбраны языки программирования PHP, HTML и SQL. HTML – стандартизированный язык разметки документов в интернете. Большинство веб-страниц содержат описание разметки на языке HTML. PHP – скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдером и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов.

SQL – формальный непроцедурный язык программирования, применяемый для создания, модификации и управления данными в произвольной реляционной базе данных, управляемой соответствующей системой управления базами данных (СУБД). SQL является прежде всего информационно-логическим языком, предназначенным для описания, изменения и извлечения данных, хранимых в реляционных базах данных. Основные функции SQL: создание в базе данных новой таблицы; добавление в таблицу новых записей; изменение записей; удаление записей; выборка записей из одной или нескольких таблиц (в соответствии с заданным условием); изменение структур таблиц. [1, с. 5].

## 2.2 Обоснование выбора среды разработки

Система управления базами данных (СУБД) − совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

Все современные СУБД имеют:

* визуальные средства создание таблиц, форм, отчетов;
* доступ к данным серверных СУБД;
* создание приложений, выполняющих функции WEB – браузеров и размещение данных в Internet;
* объектно-ориентированный язык;
* поддержку SQL.

MySQL – свободная реляционная система управления базами данных. MySQL имеет двойное лицензирование. MySQL может распространяться в соответствии с условиями лицензии GPL. Однако по условиям GPL, если какая-либо программа включает исходные коды MySQL, то она тоже должна распространяться по лицензии GPL. Это может расходиться с планами разработчиков, не желающих открывать исходные тексты своих программ. Для таких случаев предусмотрена коммерческая лицензия, которая также обеспечивает качественную сервисную поддержку.

MySQL портирована на большое количество платформ: Linux, Mac OS X, Windows 95, Windows 98, Windows NT, Windows 2000,Windows XP, Windows Server 2003, WinCE, Windows Vista, Windows 7 и Windows 10.

MySQL имеет API для языков Delphi, C, C++, Java, Perl, PHP, Python, библиотеки для языков платформы .NET.

## 2.3 Требования к конфигурации программного и аппаратного обеспечения

Данный проект разработан для ОС Windows 10. Достоинства:

1. Данную ОС устанавливают на большей массе современных ноутбуков, нетбуков и стационарных компьютеров, поэтому намного проще выполнить [восстановление windows-системы после вирусов](http://pcnotes.ru/soft/kak-vosstanovit-rabotu-noutbuka-v-otsutstvie-zashhity-ot-virusov/).
2. Является одной из самых стабильных ОС от компании Microsoft Windows.
3. Интерфейс является крайне удобным. Интерфейс пользователя «семерки» в сравнении с Windows XP и Vista отличается яркостью красок и стилями оформления.
4. Windows 10 имеет высочайший уровень устойчивости к ошибочным и неосторожным действиям пользователей.
5. Перечень настроек Windows 10 весьма богат – начиная от графики и заканчивая правами доступа и настройкой безопасности.
6. Данная ОС совместима с современными распространенными программами.

# 3 Проектирование

## 3.1 Проектирование интерфейса

Интерфейс представляет собой совокупность средств и методов, при помощи которых пользователь взаимодействует с различными, чаще всего сложными машинами, устройствами и аппаратурой.

Сайт будет состоять из нескольких страниц, преимущественно выполненых в светлых тонах.

Главная страница будет содержать приветсвие и кнопку регистрации. Страница регистрации и авторизации будут в себе нести поля для заполния и кнопку подтверждения. На странице новостей будут находится записи отсортированные по какому-либо критерию. Старница пользователя будет содержать информацию о нем самом, и инструменты для редактирования. Страница с комнотой рисования будет представлять собой страницу с интрументами рисвония.

**3.2 Инфологическая модель базы данных**

Школа искусств является таким заведением, где проводятся занятия по музыке, рисованию и танцам. В такой школе развивают таланты детей, занятия могут проводиться во второй половине дня, если ребёнок в первую смену обучается в обычной школе. Работникам школы удобнее вести электронный учёт записи на занятия, редактировать расписание, или просто просматривать расписание на неделю. Информацию удобно хранить в базе данных, так как имеется возможность быстрого поиска, надёжного хранения и фильтрации информации.

Необходимо спроектировать базу данных школы искусств, информация из которой будет использоваться для ведения расписания и записи на занятия.

Для контроля записи на занятия необходимо вести учёт учеников, их адрес и контактные данные. Для учёта записей нужно хранить в базе данных дату, время, место проведения, фамилию преподавателя, фамилию ученика и название предмета. Также для составления расписания занятий нужно знать ФИО преподавателя, часы и дни его работы, номера кабинетов.

При проектировании базы данных необходимо учитывать следующее:

* нельзя записаться на занятие, если время уже занято другим учеником (в случае индивидуальных занятий);
* нельзя записаться одному ученику на два занятия, если они проводятся в одно время;
* записаться на занятия нельзя, если ученик не зарегистрирован.

Таким образом, база данных содержит такие сущности как: Ученики, Преподаватели, Расписание, Записи, Направления, Дисциплины, Группы, Кабинеты.

Связь между сущностями Направления и Дисциплины, связывает название дисциплины с направлением. Данная связь позволяет связывать перечень учебных дисциплин с направлением, к которому они относятся.

Сущность Группы имеет связь с такими сущностями как Дисциплины, Преподаватели, Направления.

Связь между сущностями Группы и Дисциплины позволяет сортировать группы по учебным дисциплинам. Связь между сущностями Группы и Направления позволяет сортировать и определять, по какому направлению распределены группы учащихся. Связь между сущностями Группы и Преподаватели позволяет узнать, кто из преподавателей курирует группу учащихся.

Связь между сущностями Кабинеты и Направления, связывает учебные кабинеты с названиями направлений, для которых они предназначены.

Сущность Записи имеет связь с такими сущностями как Группы, Дисциплины, Преподаватели, Направления, Ученики.

Связь между сущностями Записи и Группы позволяет ученикам, при записи на занятия, записываться в интересующие их группы. Связь между сущностями Записи и Направления позволяет ученикам, при записи, выбрать понравившиеся им направление занятий. Связь между сущностями Записи и Дисциплины позволяет ученикам, при записи, выбирать занятия, которые им интересны. Связь между сущностями Записи и Преподаватели позволяет ученику увидеть, кто будет вести у него занятия. Связь между сущностями Записи и Ученики позволяет увидеть информацию о ученике, который записался на занятие.

Сущность Расписание имеет связь с такими сущностями как Группы, Дисциплины, Преподаватели, Направления, Кабинеты.

Связь между сущностями Расписание и Группы позволяет составить расписание с учётом разделения на группы. Связь между сущностями Расписание и Направления позволяет смотреть расписание с учётом того, к какому направлению относиться группа. Связь между сущностями Расписание и Дисциплины позволяет узнать, какое занятие стоит в расписании. Связь между сущностями Расписание и Преподаватели позволяет узнать, какой преподаватель будет вести занятие. Связь между сущностями Расписание и Кабинеты позволяет узнать, в каком кабинете будет проходить занятие.

**3.3 Логическая модель базы данных.**

Описание сущности «Students» школы искусств, представлено таблице 3.1.

Таблица 3.1 − Students

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип данных** | **Описание** |
| IdStudent | Целочисленный | Уникальный код ученика |
| FioStudent | Строковый | ФИО ученика |
| AgeStudent | Целочисленный | Возраст ученика |
| AddressStudent | Строковый | Адрес ученика |
| PhoneStudent | Строковый | Телефон ученика |

Описание сущности «Teachers» школы искусств, представлено таблице 3.2.

Таблица 3.2 − Teachers

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип данных** | **Описание** |
| IdTeacher | Целочисленный | Уникальный код преподавателя |
| FioTeacher | Строковый | ФИО преподавателя |
| AddressTeacher | Строковый | Адрес преподавателя |
| PhoneTeacher | Строковый | Телефон преподавателя |

Описание сущности «Directions» школы искусств, представлено таблице 3.3.

Таблица 3.3 − Directions

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип данных** | **Описание** |
| IdDirection | Целочисленный | Уникальный код направления |
| NameDirection | Строковый | Название направления |
| FioCurator | Строковый | ФИО куратора |
| PhoneCurator | Строковый | Телефон куратора |

Описание сущности «Disciplines» школы искусств, представлено таблице 3.4.

Таблица 3.4 − Disciplines

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип данных** | **Описание** |
| IdDiscipline | Целочисленный | Уникальный код дисциплины |
| IdDirection\_D | Целочисленный | Код направления |
| NameDiscipline | Строковый | Название дисциплины |

Описание сущности «Gropss» школы искусств, представлено таблице 3.5.

Таблица 3.5 − Gropss

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип данных** | **Описание** |
| IdGroup | Целочисленный | Уникальный код группы |
| IdDirection\_G | Целочисленный | Код направления |
| IdDiscipline\_G | Целочисленный | Код дисциплины |
| IdTeacher\_G | Целочисленный | Код учителя |
| NameGroup | Целочисленный | Название группы |

Описание сущности «Cabinets» школы искусств, представлено таблице 3.6.

Таблица 3.6 − Cabinets

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип данных** | **Описание** |
| IdCabinet | Целочисленный | Уникальный код кабинета |
| IdDirection\_C | Целочисленный | Код направления |
| NameCabinet | Строковый | Название кабинета |
| NumberCabinet | Целочисленный | Номер кабинета |

Описание сущности «Records» школы искусств, представлено таблице 3.7.

Таблица 3.7 − Records

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип данных** | **Описание** |
| IdRecord | Целочисленный | Уникальный код записи |
| IdGroup\_R | Целочисленный | Код группы |
| IdDiscipline\_R | Целочисленный | Код дисциплины |
| IdDirection\_R | Целочисленный | Код направления |
| IdTeacher\_R | Целочисленный | Код учителя |
| IdStudent\_R | Целочисленный | Код ученика |
| DateRecord | Дата | Дата записи |

Описание сущности «Scheduleles» школы искусств, представлено таблице 3.8.

Таблица 3.8 − Scheduleles

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип данных** | **Описание** |
| IdSchedule | Целочисленный | Уникальный код расписания |
| IdGroup\_S | Целочисленный | Код группы |
| IdDiscipline\_S | Целочисленный | Код дисциплины |
| IdDirection\_S | Целочисленный | Код направления |
| IdTeacher\_S | Целочисленный | Код учителя |
| IdCabinet\_S | Целочисленный | Код кабинета |
| DayWeek | Строковый | День занятия |
| TimeClasses | Целочисленный | Продолжительность занятия |

**3.4 Физическая модель базы данных**

При разработке данного сайта было принято решение использовать базу данных, состоящую из 8 таблиц. Структура таблицы «Students» следующим sql-кодом:

create table Students(

IdStudent int primary key auto\_increment,

FioStudent varchar(50),

AgeStudent int,

AddressStudent varchar(50),

PhoneStudent varchar(13) );

Структура таблицы «Teachers» следующим sql-кодом:

create table Teachers(

IdTeacher int primary key auto\_increment,

FioTeacher varchar(50),

AddressTeacher varchar(50),

PhoneTeacher varchar(13) );

Структура таблицы «Directions» следующим sql-кодом:

create table Directions(

IdDirection int primary key auto\_increment,

NameDirection varchar(30),

FioCurator varchar(50),

PhoneCurator varchar(13) );

Структура таблицы «Disciplines» следующим sql-кодом:

create table Disciplines(

IdDiscipline int primary key auto\_increment,

IdDirection\_D int,

NameDiscipline varchar(30),

foreign key (IdDirection\_D) references Directions(IdDirection) );

Структура таблицы «Groupss» следующим sql-кодом:

create table Groupss(

IdGroup int primary key auto\_increment,

IdDirection\_G int,

IdDiscipline\_G int,

IdTeacher\_G int,

NameGroup int,

foreign key (IdDirection\_G) references Directions(IdDirection),

foreign key (IdTeacher\_G) references Teachers(IdTeacher),

foreign key (IdDiscipline\_G) references Disciplines(IdDiscipline) );

Структура таблицы «Cabinets» следующим sql-кодом:

create table Cabinets(

IdCabinet int primary key auto\_increment,

IdDirection\_C int,

NameCabinet varchar(30),

NumberCabinet int,

foreign key (IdDirection\_C) references Directions(IdDirection) );

Структура таблицы «Records» следующим sql-кодом:

create table Records(

IdRecord int primary key auto\_increment,

IdGroup\_R int,

IdDiscipline\_R int,

IdDirection\_R int,

IdTeacher\_R int,

IdStudent\_R int,

DateRecord date,

foreign key (IdDirection\_R) references Directions(IdDirection),

foreign key (IdTeacher\_R) references Teachers(IdTeacher),

foreign key (IdDiscipline\_R) references Disciplines(IdDiscipline),

foreign key (IdGroup\_R) references Groupss(IdGroup),

foreign key (IdStudent\_R) references Students(IdStudent) );

Структура таблицы «Schedules» следующим sql-кодом:

create table Schedules(

IdSchedule int key auto\_increment,

IdGroup\_S int,

IdDiscipline\_S int,

IdDirection\_S int,

IdTeacher\_S int,

IdCabinet\_S int,

DayWeek varchar(15),

TimeClasses int,

foreign key (IdDirection\_S) references Directions(IdDirection),

foreign key (IdTeacher\_S) references Teachers(IdTeacher),

foreign key (IdDiscipline\_S) references Disciplines(IdDiscipline),

foreign key (IdGroup\_S) references Groupss(IdGroup),

foreign key (IdCabinet\_S) references Cabinets(IdCabinet) );

# 4 Описание программы

## 4.1 Логическая структура

Логическая структура ArtSchool:

1. ИМЯ ФАЙЛА − настройка параметров программы для работников.

Логическая структура Regist:

1. ИМЯ ФАЙЛА – проверка уникальности логина.

## 4.2 Физическая структура

Страница «Расписание» используется для просмотра расписания занятий и выполнения поиска личного расписания по группам.

Страница «Регистрация» необходима для регистрации нового пользователя.

Страница «Учащиеся» используется для просмотра и поиска информации для учеников школы искусств.

Страница «Преподаватели» используется для просмотра и поиска информации для преподавателей школы искусств.

# **5 Испытание и отладка программы**

## 5.1 Тестовые примеры

Тестирование программного обеспечения – процесс исследования, испытания [программного продукта](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), имеющий две различные цели:

* продемонстрировать разработчикам и заказчикам, что программа соответствует требованиям;
* выявить ситуации, в которых поведение программы является неправильным, нежелательным или не соответствующим спецификации.

В программном средстве были проведены следующие тесты:

1. Ввод неверного логина и пароля.
2. Проверка на пустоту паролей.
3. Попытка ввода существующего логина при регистрации.
4. Попытка ввода ранее зарегистрированной информации при добавлении нового пользователя.
5. Попытка поиска информации об успеваемости учащихся.
6. Попытка удалить пользователя.
7. Попытка поиска информации зарегистрированного учащегося.
8. Попытка записи учащегося на занятие в группу.
9. Попытка просмотра расписания групп.

## Анализ полученных результатов

Тест 1: Ввод неверного логина и пароля, изображен на рисунке 6.

Рисунок 6 – Уведомление о неверных данных

Тест 2: Проверка на пустоту полей изображена на рисунке 7.

Рисунок 7 – Ошибка об уже существующем логине

Тест 3: Попытка ввода существующего логина при регистрации, изображен на рисунке 8.

Рисунок 15 – Регистрация невозможна

Тест 4: Попытка ввода ранее зарегистрированной информации при добавлении нового пользователя, изображен на рисунке 9.

Рисунок 9 – Ошибка регистрации

Тест 5: Попытка поиска информации об успеваемости учащихся, изображен на рисунке 10.

Рисунок 10 – Вывод результата поиска

Тест 6: Попытка удалить пользователя, изображен на рисунке 11.

Рисунок 11 – Сообщение об удалении пользователя

Тест 7: Попытка поиска информации зарегистрированного учащегося, изображен на рисунке 12.

Рисунок 12 – Результат поиска

Тест 8: Попытка записи учащегося на занятие в группу, изображен на рисунке 13.

Рисунок 13 – Сообщение об успешной записи на занятие

Тест 9: Попытка просмотра расписания групп, изображен на рисунке 14.

Рисунок 14 – Расписание групп

Программа ведет себя корректно на всех тестах.

Из тестирования следует вывод о том, что программа не нуждается в доработке.

# 6 ОПИСАНИЕ ПРИМЕНЕНИЯ ПРОГРАММЫ

## 6.1 Назначение программы и условия применения

Данный сайт создан с целью ознакомить пользователя с интересующей его информацией. Сайт позволяет упростить работу пользователей, организовав оперативный просмотр информации об учащихся школы и о их расписании занятий, а также позволяет быстро записаться в группу по интересующему направлению искусства. Любой учащийся может зайти на данный сайт и посмотреть свою успеваемость, а также статистику всех учащихся.

Минимальные системные требования:

* ОС: Windows XP/7/8/10;
* процессор: Intel Pentium 233 МГц и выше;
* оперативная память: 64 Мбайт (рекомендуется 128 Мбайт).

## 6.2 Описание работы программы

Шаги работы сайта:

1. Запустить программу.
2. Перейти на форму авторизации.
3. Ввести ФИО.
4. Записаться на занятие, в случае необходимости.
5. Удалить запись, в случае необходимости.
6. Открыть расписание занятий.
7. Просмотреть интересующую информацию.

Для работы на сайте нужно перейти по его ссылке или открыть документ «Главная – ArtSchool.html».

Для просмотра статистики по успеваемости учащихся, следует перейти на страницу «Статистика», представленной на рисунке 1. На данной странице заполнить необходимые параметры и нажать на «Формирование отчёта».

Рисунок 1 – Просмотр статистики

Для поиска информации о расписании группы, нужно перейти на страницу «Расписание», представленной на рисунке 2. На данной странице заполнить необходимые параметры и нажать на «Поиск».

Рисунок 2 – Поиск расписания групп

Для фильтрации либо поиска нужной записи, нужно открыть страницу «Все отчеты» и перейти на вкладку «Поиск и фильтрация», на данной странице следует заполнить необходимые для фильтрации данные, представленные на рисунке 3.

Рисунок 3 – Фильтрация информации

Для сброса фильтрации нужно нажать на кнопку «Сброс фильтрации», представленная на рисунке 4.

# 

Рисунок 4 – Сброс фильтрации

**Заключение**

На сайте реализована регистрация и авторизация на сайте со следующими ролями: «Преподаватель», «Ученик».

Для ученика предусмотрены следующие функции:

* просмотр расписания занятий;
* запись на занятия в группы;
* отмена личной записи на занятия;
* поиск по дисциплинам;
* поиск по предметам;
* просмотр информации сайта.

Для преподавателей реализованы следующие функции:

* редактирование собственного расписания занятий;
* просмотр списка записавшихся учащихся;
* контроль списка записавшихся учеников;
* просмотр кабинетов;
* просмотр собственного расписания.

На данном сайте реализован поиск по дисциплинам и предметам, который зависит от действий зарегистрированного и не зарегистрированного пользователя. При результатах поиска перед пользователем предусмотрен вывод перечня дисциплин и предметов, на которые можно записаться, а также номера групп. В поиске реализован фильтр по дате, алфавиту и направлению. Для каждого предмета создана кнопка «Записаться», которая переходит на страницу с записью на занятие. Для записи на занятия, пользователь должен пройти авторизацию или регистрацию. У пользователей создана возможность записаться на занятие, или, наоборот, отменить запись.

Для преподавателя разрешено изменение своего расписания занятий, просматривать количество записанных, именно к нему, учеников, а также просматривать составы групп на занятия. Также только для преподавателей реализован просмотр личной информации записанных на занятия учеников, которая выводиться в виде графика и таблицы.

Сайт был разработан под операционную систему Windows 10 на языках Php, Html, Sql.

Приложение было протестировано. Все тесты прошли успешно.

Имеются возможности для дальнейшей модификации проекта, такие как: улучшения интерфейса программного средства, расширение функциональных возможностей приложения, такие как: построение отчётов по большему кол-ву параметров, более удобный выбор предметов при записи на занятия.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Веллинг Л. MySQL. Учебное пособие. М.: Издательский дом «Вильямс», 2005, 304с
2. Вишневский А.В. Microsoft SQL Server. Эффективная работа. СПб. Питер, 2009.
3. Волоха А. Microsoft SQL Server 2005. Новые возможности. СПб. Питер, 2006, 304с.
4. Дунаев В. Базы данных. Язык SQL. ВHV – 2006г.
5. Душан Петкович. Microsoft SQL Server 2008. Руководство для начинающих. СПб. БХВ-Питер, 2009. 752с.
6. Дьюсон Робин. Microsoft SQL Server 2008 для начинающих разработчиков. СПб.: БХВ-Питер, 2009. 704с.
7. Дюбуа П. MySQL. М.: Издательский дом «Вильямс», 2007., 1168с
8. Моисеенко С. SQL. Задачи и решения. СПб.: Питер, 2006., 256с.
9. Наумова И.Н. Microsoft SQL Server. Полезные алгоритмы от SQL.RU. СПб. Питер, 2007, 272с.
10. Нильсен П. Microsoft SQL Server 2005. Библия пользователя. М.: «Диалектика» - 2008, 1232с.
11. Шелдон Р., Мойе Дж.. MySQL. Базовый курс. М.: Издательский дом «Вильямс» «Диалектика», 2007., 880с

**ПРИЛОЖЕНИЕ А**

**(обязательное)**

Листинг программы