**РОСЖЕЛДОР**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ПУТЕЙ СООБЩЕНИЯ (СГУПС)**

|  |  |  |  |
| --- | --- | --- | --- |
| **К защите:** |  |  | |
| **Заведующий кафедрой** | **Информационные** | |
| **технологии транспорта** | | |
|  | д-р техн. наук, проф. | |
|  |  | В. И. Хабаров | |
| *подпись* |  | *инициалы, фамилия* | |
|  |  |  | |
| *дата* |  |  | |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Тема:** | Разработка базы данных для мобильного приложения «Расписание занятий» | | | | | |
|  |  | | | | | |
|  | |  | БР.БПИ.11.2022 |  |  |
|  | |  | *шифр документа* |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Выполнил** |  |  |  | **Руководитель** |
|  |  | Б.Е. Ратушный |  |  |  | канд. техн. наук, доц.  В. Г. Кобылянский |
| *подпись* |  | *инициалы, фамилия* |  | *подпись* |  | *инициалы, фамилия* |
|  |  |  |  |  |  |  |
| *дата* |  |  |  | *дата* |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Консультанты по разделам** |  |  |  |  |
| 1 |  |  |  |  |
|  |  |  |  |  |
| 2 |  |  |  |  |
|  |  |  |  |  |
| 3 |  |  |  |  |
|  |  |  |  |  |
| Нормоконтролер работы |  |  |  | ст. преп.  Т.А. Распопина |
|  |  | *подпись* |  |  |
|  |  |  |  |  |
|  |  | *дата* |  |  |

**2022 г.**

**СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ПУТЕЙ СООБЩЕНИЯ (СГУПС)**

Факультет: Бизнес-информатики

Кафедра: Информационные технологии транспорта

Направление: 09.03.03 «Прикладная информатика»

Профиль: Корпоративные информационные системы на транспорте

|  |  |
| --- | --- |
|  | ***УТВЕРЖДАЮ****: зав. кафедрой «Информационные технологии транспорта»*  д-р техн. наук, проф.  В. И. Хабаров |
|  | *«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_ 2022 г.* |

**З А Д А Н И Е**

**на выполнение выпускной квалификационной работы**

|  |  |  |  |
| --- | --- | --- | --- |
| студенту | Ратушному Богдану Евгеньевичу | | |
|  |  | | |
| 1. Тема «Разработка базы данных для мобильного приложения «Расписание занятий» утверждена приказом № 203/с от «30» мая 2022 г. | | | |
| 2. Задание выдано «12» мая 2022 г. | | | |
| 3. Срок сдачи законченной работы на кафедру «20» июня 2022 г. | | | |
| 4. Исходные данные: данные, полученные в ходе прохождения преддипломной практики | | | |
| 5. Содержание расчетно-пояснительной записки | | | |
| Наименование разделов и вопросов | | Примерное количество страниц | График (сроки) выполнения |
| Введение | | 2 | 13.05.2022 |
| Анализ предметной области | | 13 | 13.05.2022 |
| Проектирование информационной системы | | 25 | 28.05.2022 |
| Руководство пользователя | | 5 | 01.06.2022 |
| Заключение | | 1 | 02.06.2022 |

6. Содержание и объемы графической части

|  |  |  |
| --- | --- | --- |
| Наименование графического документа | Количество  слайдов | График  (сроки)  выполнения |
| Презентация PowerPoint | 21 | 07.06.2022 |

7. Консультанты по разделам

|  |  |  |
| --- | --- | --- |
| Наименование  раздела | Фамилия, И. О.  консультанта | Подпись консультанта,  дата выдачи задания |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| Руководитель |  | В. Г. Кобылянский |
|  | *(подпись, фамилия, И.О.)* |  |
| Задание к использованию принял |  | Б.Е. Ратушный |
|  | *(подпись студента)* |  |

УДК 004.41

**АННОТАЦИЯ**

В работе 49 страниц, 33 рисунка, 6 таблиц, 10 источников.

Ключевые слова: *база данных, расписание, система, факультет*.

Предметная область — факультет учебного заведения, для которого разрабатывается информационная система для просмотра расписания. Система работает с базой данных, откуда берет данные о расписании занятий и заметках. Мобильное приложение работает на платформе Android.

**ABSTRACT**

The work contains 49 pages, 33 figures, 6 tables, 10 sources.

Keywords: *a database, a system, a schedule, a faculty.*

Subject area — the faculty of the educational institution for which an information system is being developed to view the timetable. The system works with a database, from where it takes data on the class schedule and notes. The mobile application runs on the Android platform.

**ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ**

ИС — информационная система.

БД — база данных.

MS — Microsoft.

SQL (Structured Query Language) — декларативный язык программирования, применяемый для управления данным в реляционной базе данных.

SQLite — это программная библиотека, которая реализует автономный, безсерверный, транзакционный механизм базы данных SQL с нулевой конфигурацией.

API — описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой.

HTTP — протокол прикладного уровня передачи данных.

REST — архитектурный стиль взаимодействия компонентов распределенного приложения в сети.

БПИ — бакалавриат прикладная информатика.

БИСТ — бакалавриат информационные системы и технологии.

МЛ — менеджмент.

ФБИ — факультет бизнес-информатики.

**СОДЕРЖАНИЕ**

[Введение 6](#_Toc105938191)

[1Анализ предметной области 8](#_Toc105938192)

[1.1 Общее положение 8](#_Toc105938193)

[1.2 Описание проблемы и постановка задачи 9](#_Toc105938194)

[1.3 Обзор существующих систем составления расписания 11](#_Toc105938195)

[1.4 Обзор существующих систем отображения расписания 16](#_Toc105938196)

[1.5 Выводы к разделу 17](#_Toc105938197)

[2Проектирование информационной системы 20](#_Toc105938198)

[2.1 Описание бизнес-процессов 20](#_Toc105938199)

[2.2 Общая архитектура системы 26](#_Toc105938200)

[2.3 Обоснование выбора базы данных 28](#_Toc105938201)

[2.4 Проектирование базы данных 32](#_Toc105938202)

[2.5 Администрирование базы 40](#_Toc105938203)

[2.6 Выводы к разделу 41](#_Toc105938204)

[3Руководство пользователя 42](#_Toc105938205)

[Заключение 47](#_Toc105938206)

[Список использованных источников 48](#_Toc105938207)

# **ВВЕДЕНИЕ**

В настоящий период времени автоматизация процессов является, чуть ли неотъемлемой частью любого предприятия. Использование ЭВМ для различных процессов помогает сэкономить огромное количество времени, и позволяет добиться высокой производительности труда. С помощью высококачественной автоматизации можно облегчить работу предприятия и производства.

Целью автоматизации является: снижение большого количество затрат по времени, трудовых и материальных ресурсов, ведение и контроль отчетов, поддержание данных в надежном состоянии.

В Новосибирске имеется большое количество учебных заведений. Большинство из них используют информационные технологии для упрощения различных процессов между учебным заведением и студентами, а также преподавателями. Именно по этой причине для множества учебных заведений будет практично иметь свою информационную систему, с помощью которой можно будет упростить взаимодействие внутри вуза.

Целью данной ВКР является разработка информационной системы, которая должна посодействовать в упрощении отслеживания расписания и актуальной информации о предметах, с помощью хранения и изменения данных для всех видов занятий, которые проводятся в высшем учебном заведении с помощью мобильного приложения, на основании представленной информации о времени проведении занятий, а также добавление заметок к каждому предмету. Эта система будет работать в рамках определенного факультета и выполнять функции основной площадки для отслеживания занятий и просмотр актуальных заметок.

Для достижения поставленной цели необходимо выполнить ряд основных задач:

- изучить предметную область;

- сделать анализ рынка по схожей тематике;

- построить модели бизнес-процессов предметной области;

- выбрать и обосновать инструменты разработки приложения;

- разработать информационную систему для факультета учебного заведения;

- настроить и протестировать информационную систему;

- разработать руководство пользователя.

Практическая значимость выпускной квалификационной работы заключается в том, что разработанная информационная система «Расписание факультета» позволит значительно сэкономить время студента и преподавателя, а также даст возможность облегчить задачи информационно-управленческого характера.

# **1 Анализ предметной области**

## 1.1 Общее положение

Объектом автоматизации является просмотр расписания факультета образовательного учреждения «ФБИ». Основной целью данного факультета является обучение студентов по выбранному ими направлению «БПИ», «БИСТ», «МЛ».

Учебное заведение имеет бюро расписания, в котором разрабатывается расписание для каждого факультета на каждый семестр. После этого, бюро расписания передает данные о составленном расписание каждому факультету. Расписание составляется по всем требованиям, которые необходимы каждому факультету.

Обучение на факультете разделяется на несколько видов занятий: теория (прослушивание лекций в аудитории) и практика (применение полученных знаний в процессе лабораторных работ). Для обучения могут использоваться разные аудитории и корпусы учебного заведения. После того, как студент защищает выпускную квалификационную работу, ему выдается диплом. Критерии ВКР могут измениться с прибавлением новых правил в соответствии с потребностями факультета.

У факультетов имеется свои кафедры, к которым относятся определенные преподаватели и аудитории, которые оснащены специальным оборудованием, которые необходимы для оттачивания навыков, полученных во время лекционных занятий. К такому оборудованию относятся: компьютеры, маршрутизаторы, роутеры и много другое. В каждой аудитории имеется доска, проектор, компьютер, парты со стульями или скамьями, стенды с полезной информацией и многие другие вещи для обучения.

Все это очень важно для факультета, потому что оно дает надлежащую конкуренцию по обучению среди факультетов и вузов, предоставляемых на рынке учебных заведений.

Основными задачами факультета являются:

- обеспечение высокой конкурентоспособной позиции в отечественной образовательной и научной среде в области подготовки высококвалифицированных специалистов по техническим специальностям и направлениям подготовки;

- ориентация на текущие потребности национальной экономики с учетом мировых тенденций;

- повышение качества учебного процесса на основе внедрения инновационных методов обучения и современных дистанционных, телекоммуникационных и компьютерных технологий в образовательный процесс [1];

- развитие научно-исследовательской работы студентов и аспирантов;

- обеспечение высокого уровня востребованности на рынке труда выпускников факультета информационно-технического сервиса.

Для обеспечения поставленных задач, факультет осуществляет следующие функции:

- разработать и реализовать планы по привлечению абитуриентов на образовательные программы факультета;

- провести анализ перечня направлений и специальностей подготовки, указанных в лицензии университета на предмет актуальности их реализации;

- разработать и реализовать основные образовательные программы, направленные на получение прикладных квалификаций;

- использовать индивидуальный адресный подход в организации работы со студентами.

## 1.2 Описание проблемы и постановка задачи

Данная выпускная квалификационная работа направлена на создание прототипа информационной системы для факультета, для того чтобы упростить процесс просмотра расписания. Актуальность этой проблемы поясняется тем, что в настоящее время, процесс отслеживания расписания студентом или преподавателем проблемно осуществить с помощью мобильного телефона. Лист с расписанием вмещает занятия на весь семестр, который невозможно удобно и оперативно просмотреть занятия из-за его громоздкости. Пример отображения листа с расписанием представлен на рисунке 1.1.

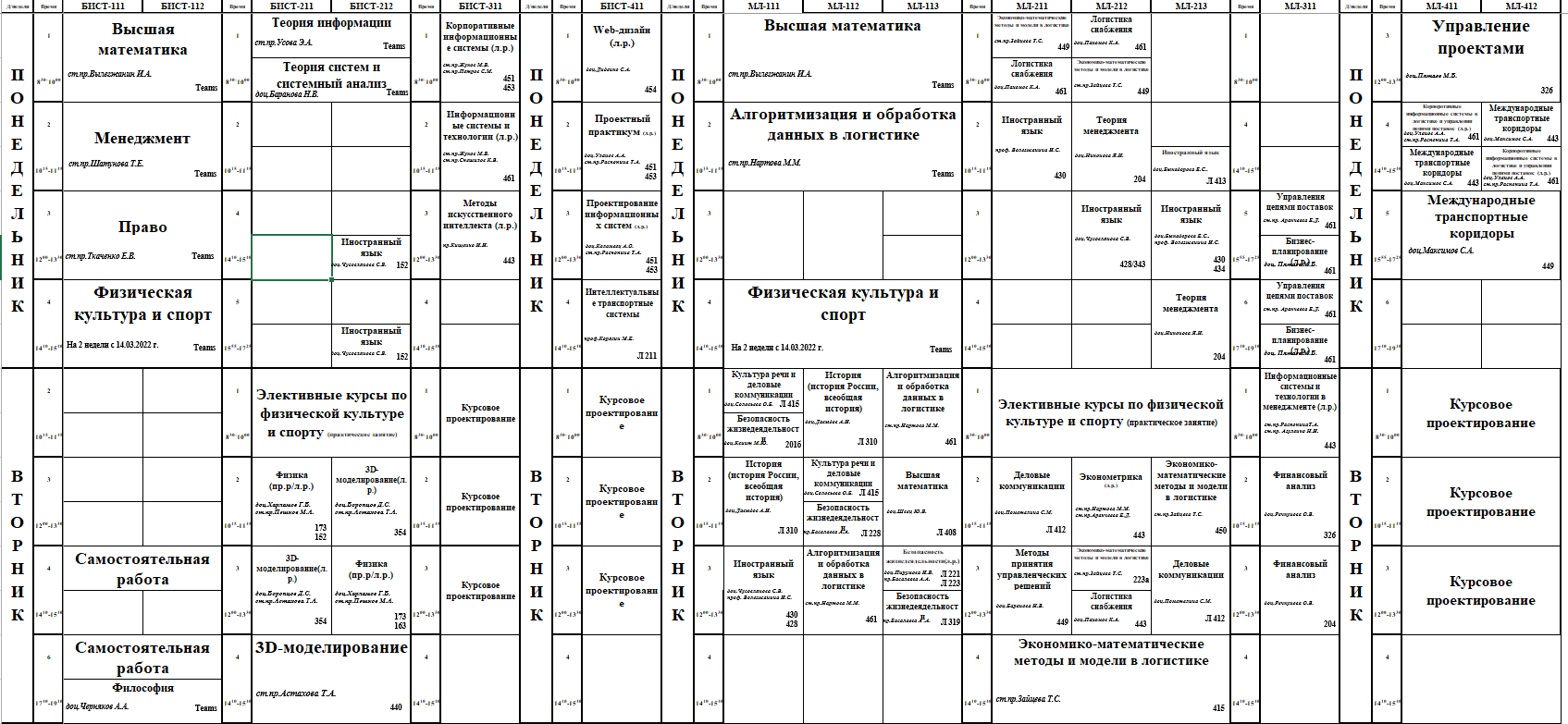


Рисунок 1.1 — Пример расписания

Если рассматривать с точки зрения затраченного времени, то этот способ является не рациональным, так как каждый раз, когда нужно узнать расписание или уточнить, в какой аудитории идет занятие, необходимо искать группу или предмет в огромной таблице, что может занять продолжительное количество времени.

Рассмотрим алгоритм просмотра расписания студентом или преподавателем:

- студенту или преподавателю, следует потратить время, для того чтобы зайти на сайт, найти свой факультет и там скачать файл с расписанием;

- после чего необходимо найти свою группу в большой таблице, а также нужный день и занятие.

Этот алгоритм не комфортен для постоянного использования в качестве просмотра занятий и информации о предметах, также сам файл может потеряться на телефоне, либо в само расписание могут внести правки, и если их не запомнить, то студент или преподаватель может путаться.

Следовательно, можно сделать вывод о необходимости приложения, в котором будут выполнены следующие фрагменты:

- с помощью мобильного приложения, пользователь всегда будет знать, где просмотреть расписание;

- облегчение просмотра расписание;

- контроль и быстрое обновление со стороны факультета;

- возможность оставлять заметки на предметах.

## 1.3 Обзор существующих систем составления расписания

Сегодня проблема составления расписания в вузе актуальна как никогда. Это связано с появлением новых специальностей, различных типов аудиторий для лекционных, практических и лабораторных занятий, большим количеством учебных групп. Справиться с такой нагрузкой методистам становится все сложнее. Естественны попытки переложить решение этой проблемы на плечи компьютера, что привело к созданию специальных программ для составления расписания занятий в вузе. Был проведен обзор основных программ, используемых для составления расписания, рассмотрена спецификация каждой из них.

Эти программы можно разделить по их назначению. Одни предназначены для составления только школьного расписания, другие — для создания расписания в вузе, третьи совмещают в себе обе спецификации, предоставляя пользователю возможность составить расписание с учетом особенностей конкретного учебного заведения.

Основными программами для составления расписания в вузе в настоящее время являются: «Ректор-вуз 1.6», «Университет 3.2.0.711», «Расписание занятий 2.1», «Расписание ПРО», «AVTOR-2000», «Аналит: расписание, авторасписание».

В ходе написания работы был произведен анализ основных программ для составления расписания, который производился по следующим критериям:

- выполнение задач составления расписания;

- необходимость закупки дополнительных программ;

- платформа;

- стоимость лицензии на один компьютер;

- удобство и простота интерфейса.

По результатам проведенного анализа можно сделать следующие выводы. Программа «Расписание занятий 2.1» не отвечает всем критериям, так как выполнена для операционной системы DOS. Система «Расписание ПРО» не является удачной, так как ее область функционирования достаточно узка. К недостаткам программы «Университет 3.2.0.711» можно отнести проблематичность установки и достаточно высокую стоимость. «AVTOR-2000» не прошел отбор, так как его цена неприемлема для среднего вуза. Программа «Аналит: расписание» реализована для 1с8 и требует от пользователя достаточно серьезного уровня подготовки. Программа «Ректор-вуз» для составления расписания занятий в системе высшего профессионального образования версии 1.6 по результатам анализа занимает лидирующую позицию.

Эта программа обладает следующими преимуществами:

- удобный и понятный даже неопытному пользователю интерфейс;

- планирование распределения нагрузок по неделям (причем расписание по разным неделям может быть различным);

- расписание занятий составляется без «окон»;

- в графике работы преподавателя можно запретить первые или вторые пары (не все преподаватели могут приехать рано);

- составление отчетов по распределению нагрузок кафедр и вуза по видам;

- автоматический и ручной режимы составления расписания занятий.

К сожалению, программа не очень удобна в использовании для заочной формы обучения и для вузов, периодичность расписания в которых не одна неделя, а несколько.

Остановимся на работе программы «Ректор-вуз» более подробно. Она состоит из четырех разделов: «Списки», «Нагрузки», «Расписание» и «Замены». Раздел «Списки» служит для ввода, редактирования и печати списков кафедр, специальностей, групп, дисциплин, аудиторий, преподавателей и видов занятий, как показано на рисунке 1.2.

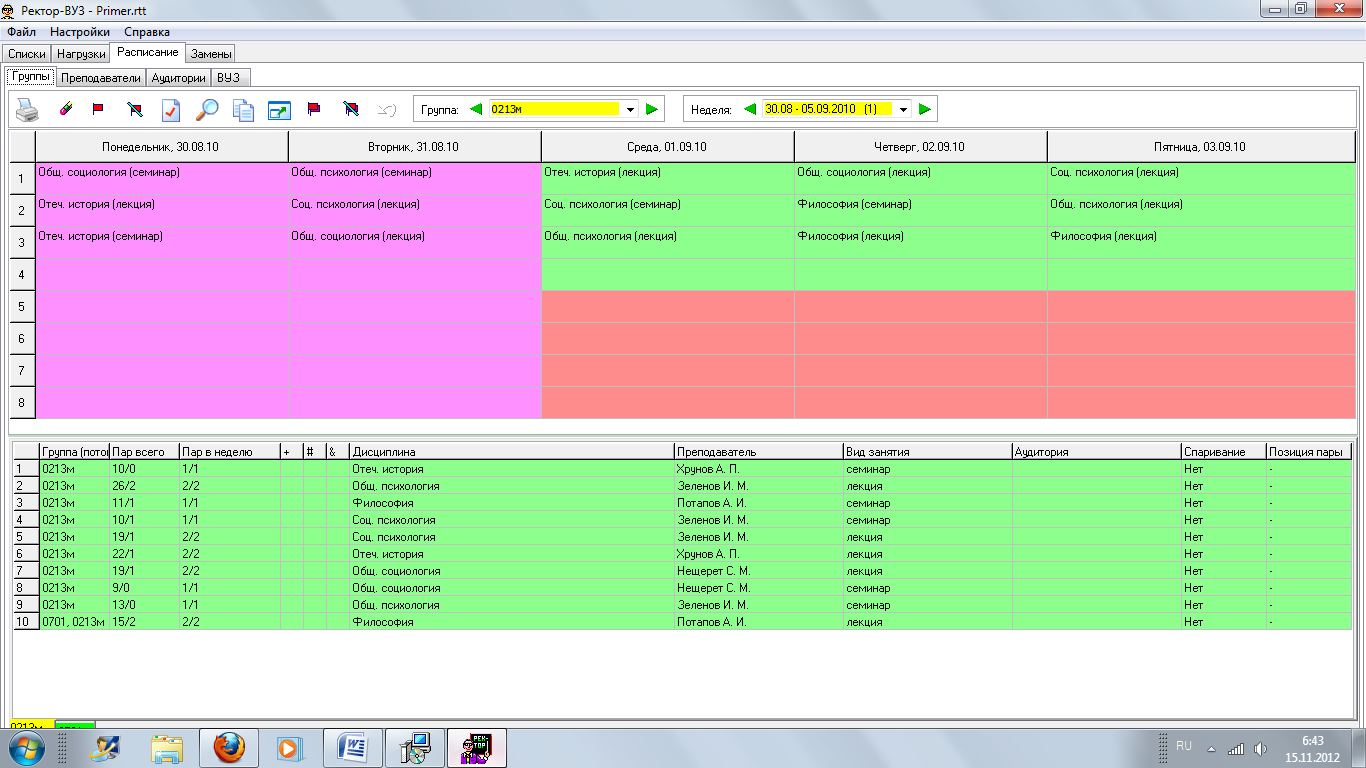


Рисунок 1.2 — Интерфейс программы

Раздел «Нагрузки» используется для ввода, редактирования и печати учебных планов по каждой специальности, нагрузок преподавателей, графиков распределения часов по неделям в пределах семестра, отчетов по загрузке преподавателей, кафедр и вуза в целом представлено на рисунке 1.3.

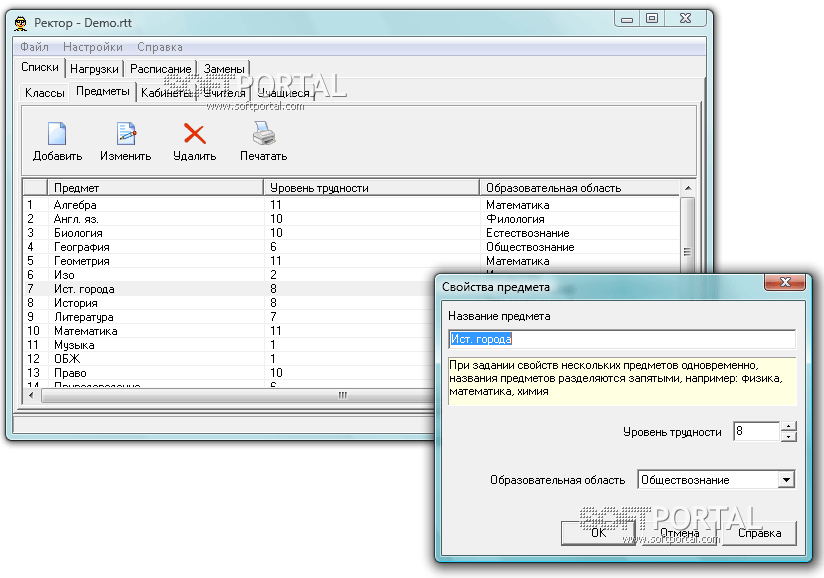


Рисунок 1.3 — Интерфейс раздела «Нагрузки»

Раздел «Расписание» предназначен для составления расписания по группам, преподавателям, аудиториям и вузу в целом, пример показан на рисунке 1.4, но, к сожалению, в программе нет возможности закрепить за преподавателем или группой определенную аудиторию.

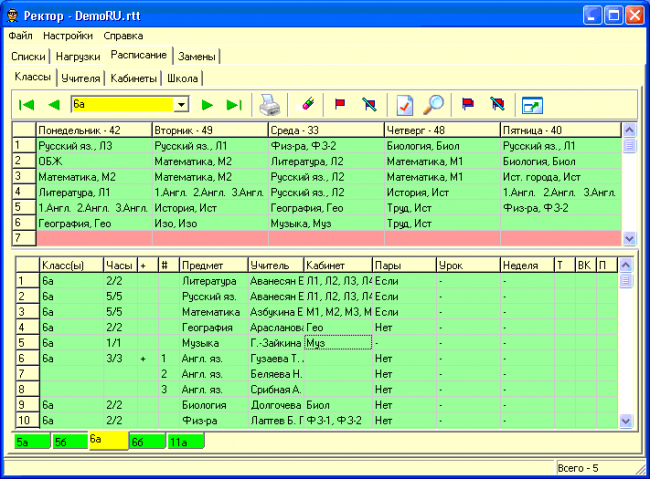


Рисунок 1.4 — Интерфейс раздела «Расписание»

Расписание занятий можно составлять в автоматическом, ручном или комбинированном режимах, при этом переходить от одного рабочего режима к другому можно в любой момент времени. При составлении расписания в автоматическом режиме программа учитывает все сформулированные требования к расписанию. При составлении расписания в ручном режиме программа подсказывает возможные варианты расстановки уроков выбранного преподавателя, возможные варианты заполнения пустых клеток в расписании группы, следит за количеством мест в аудиториях.

Готовое расписание занятий одной, всех или нескольких групп и преподавателей можно сохранить в форматах Microsoft Word, Excel или HTML. Программа «Ректор-вуз» работает на любом компьютере с операционной системой Windows ХР, Windows Vista или Windows 7. Распространение программы условно-бесплатное, однако демоверсия программы «Ректор-вуз», предоставляя возможность составить в ней расписание, не позволяет сохранять его. Лицензия на один компьютер стоит на сегодняшний день 7 579 рублей.

При составлении расписания в программе «Ректор-вуз» у пользователя не возникает никаких сложностей. Работать с ней достаточно удобно. Непростой задачей является проблема контроля ошибок в расписании (случается, что по ошибке назначены занятия в одной и той же аудитории, в одно и то же время у групп различных специальностей и курсов, у одного преподавателя, но в разных группах). Эта проблема в рамках программы «Ректор-вуз» решается с легкостью. Она позволяет при помощи специальных функций отслеживать занятость аудиторий, кроме того, при наличии совпадения представляет список свободных аудиторий.

Безусловно, программа «Ректор-вуз» не может полностью самостоятельно составить расписание в вузе, ведь необходимо учитывать огромное количество различных нештатных ситуаций. Однако можно сказать с уверенностью, что программа «Ректор-вуз» существенно облегчает процесс составления расписания, экономит время, а также следит за корректным составлением расписания, не допуская ошибок, по причине человеческого фактора (усталость).

Процесс составления расписания задача не простая, поэтому перед нами стоит задача отображения на мобильных устройствах.

## 1.4 Обзор существующих систем отображения расписания

Примерами решений в области упрощения просмотра расписания является программное обеспечение компании «Кампус».

«Кампус» — является приложением, которое направленно в первую очередь на удобное отображение расписания на мобильном устройстве и упрощение взаимодействия с расписанием, а также оно тесно интегрировано с базой данных сервиса Кампус.

Интерфейс приложения представлен на рисунках 1.5 и 1.6.

Единственная функция этого приложения — это отображение расписания на мобильном устройстве. Минусом этого приложения является список учебных заведений, который сильно ограничен.

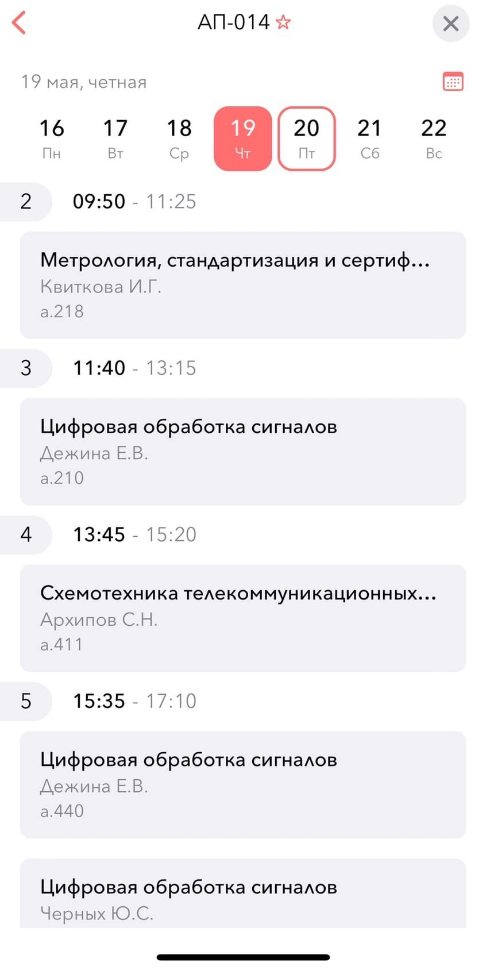


Рисунок 1.5 — Интерфейс мобильного приложения

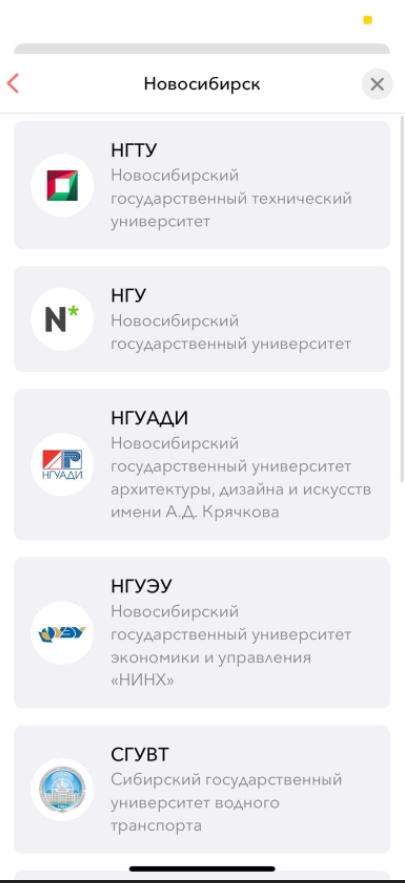


Рисунок 1.6 — Интерфейс при выборе учебного заведения

## 1.5 Выводы к разделу

В первом разделе данной выпускной квалификационной работе:

- проведен анализ предметной области;

- выявлены функции, требующие автоматизации посредством реализации приложения «Расписание»;

- проведен краткий обзор существующих систем автоматизации.

В результате анализа аналогов, было принято решение разработать собственный продукт, так как закрытый исходный код большинства готовых продуктов не дает возможности самостоятельной доработки и расширения функционала системы, что может привести к дополнительным финансовым и временным затратам в процессе внедрения обновлений. Такой проект будет разделен на две части — это проектирование и разработка базы данных, и разработка мобильного приложения.

Наличие недостатков объясняет необходимость разработки собственного приложения.

К преимуществам создания собственной системы можно отнести:

- открытый исходный код позволит проводить модификацию, расширение и усложнение системы в зависимости от потребности факультета;

- оперативное обновление расписания, так как приложение принадлежит факультету;

- возможность добавление заметок к предметам.

Приложение с расписанием для студентов и преподавателей — утилита для смартфонов, предназначенная для просмотра расписания занятий для конечных пользователей, включает в себя следующие функциональные возможности:

- просмотр расписания;

- выбор группы;

- редактирование расписания со стороны администратора;

- отображение ссылок на официальный сайт и социальные сети университета;

- возможность добавления объявления в приложение.

Основными целями разработки Системы являются:

- упрощение взаимодействия с расписанием для студентов и преподавателей;

- повышение удобства информирования студентов о изменении расписания.

Основная функция Системы — обеспечение процесса дистанционного просмотра расписания для пользователя, с использованием приложения.

Система должна обеспечивать выполнение перечисленных ниже функций (функциональные характеристики):

- круглосуточная доступность и работоспособность системы;

- возможность хранения расписания групп и его отображения по запросу пользователя;

- возможность отображения заметок предмета пользователям.

К нефункциональным требованиям относятся следующие пункты.

*Требования к надежности:*

- должен быть разработан механизм и регламент полного резервного копирования системы;

- максимальное время остановки системы для проведения технологических операций не более 2 часа в сутки.

*Требования к объему хранимой информации:*

- система должна хранить всю информацию о расписании групп активных пользователей в течение актуальности расписания.

*Требования к производительности:*

- количество одновременно работающих пользователей: до 1000 человек;

- время подготовки к работе с момента запуска приложения до запуска отображения расписания не должно превышать 5 секунд;

- максимальная продолжительность вычислений должна быть не более 7 секунд.

Требования к численности и квалификации персонала.

*Для эксплуатации Системы определены следующие роли:*

- пользователь;

- системный администратор.

Требования к составу и параметрам технических средств.

*Минимальные системные требования к клиентской части:*

- операционная система: Android 9.0;

- сеть: 5 Мбит/сек;

- процессор: 1 ГГц;

- свободное место: 40 МБ;

- оперативная память: 1 ГБ.

Требования к информационной и программной совместимости:

- система должна быть совместима с любыми операционными системами семейства Android от версии 9.0 и выше.

# **2 Проектирование информационной системы**

Проектирование информационных систем — это упорядоченная совокупность методологий и средств создания или модернизации информационных систем.

Перед непосредственной разработкой программного продукта необходимо спроектировать каждый этап, предусмотреть требуемую функциональность системы и степень адаптации к изменяющимся условиям ее использования .

Такая информационная система будет состоять из двух частей: базы данных и связанного с ним мобильного приложения. Таким образом рассматривать также нужно два программных продукта.

## 2.1 Описание бизнес-процессов

Моделирование бизнес-процессов является одним из методов улучшения качества и эффективности работы организации. В основе этого метода лежит описание процесса через различные элементы (действия, данные, события, материалы и прочее) присущие процессу. Как правило, моделирование бизнес-процессов описывает логическую взаимосвязь всех элементов процесса от его начала до завершения в рамках организации. В более сложных ситуациях моделирование может включать в себя внешние по отношению к организации процессы или системы [2].

Моделирование бизнес-процессов позволяет понять работу и провести анализ организации. Это достигается за счет того, что модели могут быть составлены по различным аспектам и уровням управления. В больших организациях моделирование бизнес-процессов выполняется более подробно чем в малых, что связано с большим количеством кросс-функциональных связей.

Обычно для моделирования бизнес-процессов применяются различные компьютерные средства и программное обеспечение. Это облегчает управление моделями, отслеживание в них изменений и позволяет сократить время анализа.

Моделирование бизнес-процессов преследует несколько целей:

- это описания процессов, которое можно проследить, от начала, до завершения;

- моделирование бизнес-процессов задает правила выполнения процессов, то есть. то, каким образом они должны быть выполнены. Если следовать установленным в моделях правилам, руководящим указаниям или требованиям, то можно достичь желаемой производительности процессов;

- установление взаимосвязей в процессах для выстраивания связи между процессами и требованиями, которые они должны выполнять.

Моделирование бизнес-процессов, как правило, включает в себя выполнение нескольких последовательных стадий.

Состав стадий, которые включает в себя моделирование бизнес- процессов, следующий:

- выявление процессов и построение исходной модели «как есть». Для того чтобы улучшить процесс, необходимо понимать, как он работает в этот момент. На этой стадии определяются границы процесса, выявляются его ключевые элементы, собираются данные о работе процесса. В результате создается исходная модель процесса «как есть». Эта модель не всегда адекватно отражает работу процесса, поэтому модель этой стадии можно назвать «первым драфтом» или исходной моделью «как есть»;

- пересмотр, анализ и уточнение исходной модели. На этой стадии выявляются противоречия и дублирование действий в процессе, определяются ограничения процесса, взаимосвязи процесса, устанавливается необходимость изменения процесса. В результате формируется окончательный вариант модели «как есть»;

- разработка модели «как должно быть». После анализа существующей ситуации, необходимо определить желаемое состояние процесса. Это желаемое состояние представляется в модели «как должно быть». Такая модель показывает, как процесс должен выглядеть в будущем, включая все необходимые улучшения. В ходе этой стадии моделирования бизнес- процессов и разрабатываются такие модели;

- тестирование и применение модели «как должно быть». Эта стадия моделирования связана с внедрением разработанной модели в практику деятельности организации. Модель бизнес-процесса проходит апробацию, и в нее вносятся необходимые изменения;

- улучшение модели «как должно быть». Моделирование бизнес-процессов не ограничивается только созданием модели «как должно быть». Каждый из процессов по ходу работы продолжает изменяться и совершенствоваться, поэтому модели процессов должны регулярно пересматриваться и улучшаться. Эта стадия моделирования связана с постоянным улучшением процессов и улучшением модели бизнес-процессов.

*Диаграмма вариантов использования.*

Прецедент — возможность моделируемой системы (часть ее функциональности), благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат. Прецедент соответствует отдельному сервису системы, определяет один из вариантов ее использования и описывает типичный способ взаимодействия пользователя с системой. Варианты использования обычно применяются для спецификации внешних требований к системе.

Основное назначение диаграммы — описание функциональности и поведения, позволяющее заказчику, конечному пользователю и разработчику совместно обсуждать проектируемую или существующую систему.

При моделировании системы с помощью диаграммы прецедентов системный аналитик стремится:

- четко отделить систему от ее окружения;

- определить действующих лиц (актеров), их взаимодействие с системой и ожидаемую функциональность системы;

- определить в глоссарии предметной области понятия, относящиеся к детальному описанию функциональности системы (то есть прецедентов).

Диаграмма вариантов использования состоит из актеров, для которых система производит действие и собственно действия Use Case, которое описывает то, что актер хочет получить от системы. Актер обозначается значком человечка, а Use Case — овалом.

На рисунках 2.1 и 2.2 представлены диаграммы вариантов использования для администратора базы данных, диаграмма функции пользователя.

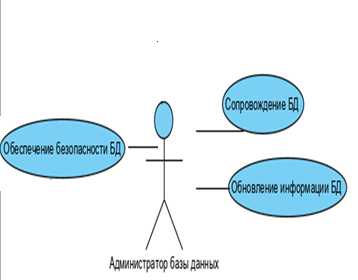


Рисунок 2.1 — Функции администратора базы данных

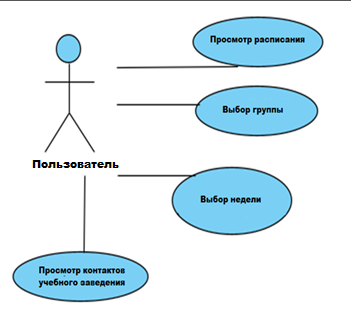


Рисунок 2.2 — Функции пользователя

*Диаграмма деятельности.*

Диаграмма деятельности — еще одна важная диаграмма в UML, описывающая динамические аспекты системы.

Диаграмма деятельность — это, по сути, блок-схема, представляющая поток от одного действия к другому. Деятельность может быть описана как работа системы [3].

Поток управления передается от одной операции к другой. Этот поток может быть последовательным, разветвленным или параллельным. Диаграммы действий касаются всех типов управления потоком с использованием различных элементов, таких как fork, join, и многие другие.

Деятельность — это особая операция системы. Диаграммы действий используются не только для визуализации динамической природы системы, но они также используются для построения исполняемой системы с использование прямого и обратного проектирования.

Единственная недостающая вещь на диаграмме активности - это часть сообщения. Он не показывает поток сообщений от одного действия к другому. Диаграмма деятельности иногда рассматривается как блок-схема.

Хотя диаграммы выглядят как блок-схема, это не так. Он показывает разные потоки, такие как параллельный, разветвленный, параллельный и одиночный.

Диаграмма активности подходит для моделирования потока активности системы. Приложение может иметь несколько систем. Диаграмма деятельности также охватывает эти системы и описывает поток от одной системы к другой.

Это конкретное использование не доступно на других диаграммах. Этими системами могут быть базы данных, внешние очереди или любая другая система. На рисунках 2.3-2.5 представлены диаграммы деятельности, на которых представлен процесс просмотра расписания студентом и заполнение расписания администратором базы.



Рисунок 2.3 — Диаграмма деятельности процесса «Просмотр расписания»

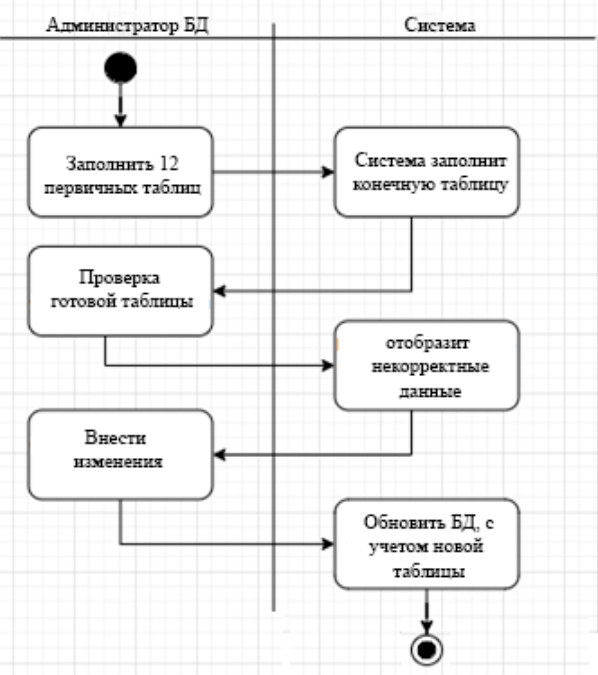


Рисунок 2.4 — Диаграмма деятельности процесса «Ввод расписания»



Рисунок 2.5 — Диаграмма деятельности процесса «Редактирование расписания»

В данном подразделе были описаны бизнес-процессы, которые будут использоваться в прототипе мобильного приложения.

## 2.2 Общая архитектура системы

Система должна работать с помощью API, этот термин расшифровывается как Application Programming Interface, что в переводе на русский значит «Программный Интерфейс Приложения». Простыми словами, интерфейс — это «прослойка» между приложением А и приложением Б. В ней происходят процессы, которые позволяют двум программам обмениваться информацией и выполнять функции, связанные с обеими сторонами, скрывая «внутреннее строение» программ. Такой подход позволяет наладить взаимодействие между несколькими утилитами, не задумываясь о том, как они устроены, какая программная логика ими движет и каким образом обрабатываются передаваемые данные [4].

Используется программный архитектурный стиль REST API, который означает «передача репрезентативного состояния», то есть сервер передает клиенту представление о состоянии запрошенного ресурса после вызова API.

То есть REST API взаимодействует с помощью HTTP запросов с методами GET, POST, PUT и DELETE, для управления данными. C помощью REST API мы отправляем запрос от одного программного обеспечения, а затем второе программное обеспечение повторно отправляет ресурсы как JSON, XML или HTML. Графическое представление как оно работает представлено на рисунке 2.6.

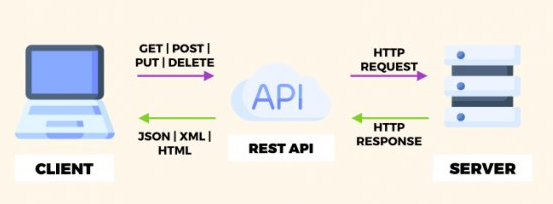


Рисунок 2.6 — Графическое представление REST API

В этой информационной системе сервер с мобильным приложением общаются с помощью XML данных. XML хранит информацию в текстовом формате. Для того, чтобы мобильное приложение функционировало, ему нужно подключиться с указанием IP адреса сервера или доменного имени сервера.

В прототипе нашего мобильного приложения серверная часть реализована не будет по некоторым причинам. Были рассмотрены несколько вариантов бесплатных серверов для хостинга, такие как Hostinger и ONOS (1&1). Это одни из самых популярных хостинг-провайдеров, но из-за ситуации в мире они прекратили предоставлять услуги клиентам из России. Были рассмотрены Российские хостинг-провайдеры, но за их услуги нужно платить немалую сумму, так как для обслуживания серверов необходимы различные оборудования, аренды стоек для размещения оборудования в дата-центрах и прочие затраты. Конечно, у отечественных провайдеров есть бесплатные тарифы, но при поиске бесплатного хостинга важно понимать, что у таких тарифов нет стабильности в работе и есть определенные ограничения [5].

Архитектура прототипа мобильного приложения реализована как показано на рисунке 2.7.

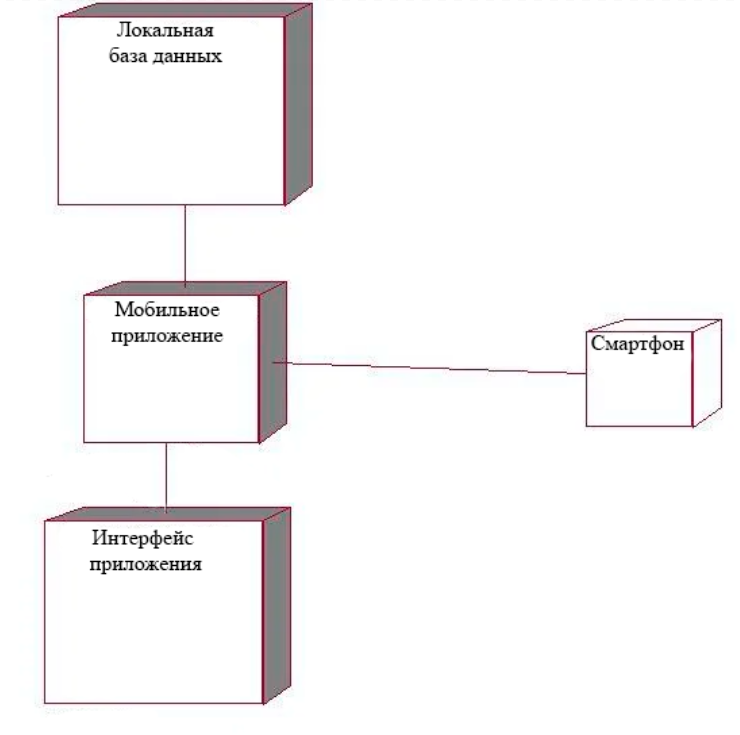


Рисунок 2.7 — Архитектура мобильного приложения

## 2.3 Обоснование выбора базы данных

Информацией, хранящейся в базе данных, может быть все что угодно: каталог продукции, информация о клиентах, контент веб-сайта и другое. Для обеспечения доступа к информации, а также для управления ею, применяют систему управления базами данных.

Система управления базами данных — это комплекс языковых и программных средств, предназначенный для создания, ведения и совместного использования базы данных многими пользователями. Обычно системы управления базами данных различают по используемой модели данных [6].

Были выделены основные системы управления базами данных, использующиеся на предприятиях:

- MongoDB;

- SQLite;

- Microsoft SQL сервер.

Для сравнения и выбора нужной системы управления базами данных были выделены следующие критерии:

- стоимость;

- скорость работы;

- надежность;

- понятность документации;

- функционал.

*MongoDB.*

База данных, которая имеет коммерческую версию — MongoDB, она предназначена для приложений, которые используют как структурированные, так и неструктурированные данные. Ядро является очень гибким и работает при подключении базы данных к приложениям через драйверы MongoDB. Существует широкий выбор доступных драйверов, поэтому легко найти драйвер, который будет работать с требуемым языком программирования.

Поскольку изначально система MongoDB не была разработана для обработки моделей реляционных данных, могут возникнуть проблемы производительности, если вы попытаетесь использовать ее таким образом. Однако движок предназначен для обработки различных данных, которые нельзя отнести к реляционным, и может хорошо справляться там, где другие движки работают медленно или бессильны. MongoDB 3.2 — это последняя версия, и она имеет новую подключаемую систему движков хранения. Документы могут быть проверены в процессе обновления или выполнения вставок, а функции текстового поиска были улучшены. Новая способность частичного индексирования может привести к более высокой производительности, уменьшая размер индексов.

*Microsoft SQL сервер.*

Еще одной из популярных СУБД является программный продукт Microsoft SQL-сервер. Это система управления базами данных, движок которой работает на облачных серверах, а также локальных серверах, причем можно комбинировать типы применяемых серверов одновременно.

Вскоре после выпуска Microsoft SQL сервер 2016, Microsoft адаптировала продукт для операционной системы Linux, а на Windows-платформе он работал изначально.

Последняя версия Microsoft SQL-сервер поддерживает dynamic data masking (динамическую маскировку данных), которая гарантирует, что только авторизованные пользователи будут видеть конфиденциальные данные.

Microsoft SQL отлично бы подходила к нашей информационной системе, но, к сожалению, она имеет высокую стоимость.

*SQLite.*

Бесплатная компактная, легко встраиваемая в приложения база данных. Так как это система базируется на файлах, то она предоставляет довольно широкий набор инструментов для работы с ней, по сравнению с сетевыми СУБД. При работе с этой СУБД обращения происходят напрямую к файлам (в этих файлах хранятся данные), вместо портов и сокетов в сетевых СУБД. Файловая структура — вся база данных состоит из одного файла, поэтому её очень легко переносить на разные машины.

Именно поэтому SQLite очень быстрая, а также мощная благодаря технологиям обслуживающих библиотек [7].

Обычно для работы СУБД, такой как MySQL, PostgreSQL и так далее, требуется отдельный серверный процесс. Приложения, которые хотят получить доступ к серверу базы данных, используют протокол TCP / IP для отправки и получения запросов. Это называется архитектурой клиент / сервер.

Следующий рисунок 2.8 иллюстрирует архитектуру клиент / сервер СУБД.

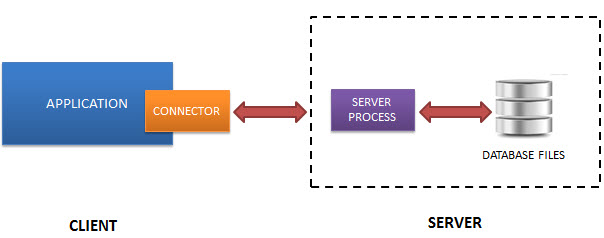


Рисунок 2.8 — Пример работы клиен-сервер

SQLite не работает таким образом и не требует запуск сервера.

База данных SQLite интегрирована с приложением. Приложения взаимодействуют с базой данных SQLite для чтения и записи непосредственно из файлов базы данных, хранящихся на диске.

Следующий рисунок 2.9 иллюстрирует архитектуру без сервера SQLite.

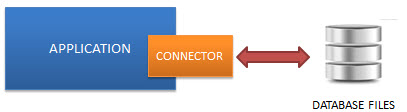


Рисунок 2.9 — Пример архитектуры без сервера

Таким образом, для создания информационной системы была выбрана SQLite, в связи с тем, что она бесплатная, имеет такой же функционал, как и платные аналоги, имеет понятную документацию, высокую надежность и скорость работы. В таблице 2.1 приведен обзор базы данных по критериям.

Таблица 2.1 — Обзор базы данных по критериям

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Стоимость | Скорость  работы | Надежность | Понятность в документации | Функционал |
| MongoDB | бесплатно | высокая | высокая | Тяжела в освоении | обширный |
| SQLite | бесплатно | высокая | высокая | Легка в освоении | обширный |
| Microsoft SQL | высокая | высокая | высокая | Легка в освоении | обширный |

## 2.4 Проектирование базы данных

База данных (БД) — это совокупность массивов и файлов данных, организованная по определенным правилам, предусматривающим стандартные принципы описания, хранения и обработки данных независимо от их вида.

SQL (англ. structured query language — «язык структурированных запросов») — декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных [8, 9].

Изначально SQL был основным способом работы пользователя с базой данных и позволял выполнять следующий набор операций: 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.

Со временем SQL усложнился новыми конструкциями, обеспечил возможность описания и управления новыми хранимыми объектами (например, индексы, представления, триггеры и хранимые процедуры) и стал приобретать черты, свойственные языкам программирования.

При всех своих изменениях SQL остается самым распространенным лингвистическим средством для взаимодействия прикладного программного обеспечения с базами данных. В то же время современные СУБД, а также информационные системы, использующие СУБД, предоставляют пользователю развитые средства визуального построения запросов.

Проектирование базы данных — это процесс, который подразумевает использование определенной технологии при создании концептуальной схемы базы данных и определения ограничения целостности.

Основными задачами проектирования БД являются: обеспечение хранения в БД необходимой информации, обеспечение возможности получения данных по определенным запросам, сокращение избыточности и дублирования данных, обеспечение целостности БД.

Сущность — это объект, о котором в системе будут накапливаться данные. Для сущности указывается название и тип (сильная или слабая). Сильные сущности существуют сами по себе, а существование слабых сущностей зависит от существования сильных.

Атрибут — свойство сущности базы данных.

Ввод данных в базу будет происходить через Excel. Конечно, возможно заполнение через SQLite, но был выбрал вариант через Excel, потому что это будет более легче для обычного пользователя, то есть администратору базы не будет необходимо иметь знания для использования SQL, что в дальнейшем может быть плюсом для вуза. В общей сложности в Excel будет четырнадцать таблиц на разных листах. Двенадцать из которых это первичные таблицы, предназначенные для заполнения данных (6 дней четной недели и 6 нечетной), а также два справочника (Группы и Время), все эти таблицы будут заполняться администратором базы данных. Так как некоторые данные заполняются вручную, была предусмотрена проверка на корректность введенных данных. Эти таблицы необходимы для удобного первичного заполнения, а также редактирования. После того, как таблицы заполнены, на основе этих данных формируется общая таблица в DB browser. В такой реализации общая таблица выступает в роли конечных данных, которая будет храниться на сервере для последующего получения информации приложением на мобильном устройстве и их конечного графического отображения, а также эта таблица необходима для оптимизации и повышения работоспособности приложения, так как из одной таблицы данные получать быстрее, чем из промежуточных. Далее будет представлена логическая модель предметной области.

В таблице 2.2 представлен набор сущностей и их атрибутов БД, которые были выделены в ходе анализа и моделирования бизнес-процессов предметной области.

Таблица 2.2 — Сущность и атрибуты сущности

|  |  |
| --- | --- |
| Сущность | Атрибут |
| ДЕНЬ(Х), где Х это день1-день12 | ID дня |
| ФИО преподавателя |
| Предмет |
| Время |
| Группа |
| Номер кабинета |
| Вид занятия |
| ID Группы |
| ID Времени |
| Время | ID |
| Время |
| Группы | ID |
| Группа |

Следующим шагом стало описание этих сущностей, которые представлены в таблицах 2.3-2.5.

Таблица 2.3 — Описание сущности «День(Х)»

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя полей | Тип данных | Характеристика |
| Id дня | Id | Int | Первичный ключ |
| ФИО преподавателя | Преподаватели | Text | Обязательное поле |
| ID Время | Times | Int | Обязательное поле |
| ID Группа | Gkey | Int | Обязательное поле |
| Предмет | Предмет | Text | Обязательное поле |
| Номер кабинета | Кабинет | Text | Обязательное поле |
| Вид занятия | Занятие | Text | Обязательное поле |
| Время | Время | Text | Обязательное поле |
| Группа | Группа | Text | Обязательное поле |

Таблица 2.4 — Описание сущности «Время»

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя полей | Тип данных | Характеристика |
| ID | Id | Int | Первичный ключ |
| Время | Время | Text | Обязательное поле |

Таблица 2.5 — Описание сущности «Группы»

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя полей | Тип данных | Характеристика |
| ID | Id | Int | Первичный ключ |
| Группы | Группа | Text | Обязательное поле |

На уровне физической модели, когда уже идет привязка к конкретной СУБД, имеет смысл ввести обобщенную сущность «Общая таблица» в виде таблицы, которая будет основана на отдельных сущностях День(X), для упрощения приложения. Описание сущности «Общая таблица» представлена в таблице 2.6.

Таблица 2.6 — Описание сущности «Общая таблица»

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя полей | Тип данных | Характеристика |
| Id | Id | Int | Первичный ключ |
| Дни | Days | Int | Обязательное поле |
| ФИО преподавателя | Teachers | Text | Обязательное поле |
| Время | Time | Text | Обязательное поле |
| Группа | Groups | Text | Обязательное поле |
| Предмет | Subject | Text | Обязательное поле |
| Номер кабинета | Cabinets | Text | Обязательное поле |
| Заметка | Notes | Text | Необязательное поле |
| Вид занятия | Occupations | Text | Обязательное поле |

*Определение сходства.*

Следующие правила в указанном порядке используются для определения типа столбца и сходства любого столбца:

- если объявленный тип столбца имеет строку «INT», то ему присваивается целочисленное сходство;

- если объявленный тип столбца содержит какие-либо строки типа «TEXT», «VARCHAR» или «CLOB», то этот столбец имеет ТЕКСТОВОЕ сходство;

- если объявленный тип столбца содержит строки типа «BLOB» или если тип не указан, то столбец имеет сходство с BLOB;

- если объявленный тип столбца содержит строки типа «FLOA» или «DOUBT», то столбец имеет РЕАЛЬНОЕ сходство.

В противном случае сходство является ЧИСЛОВЫМ [10, 11].

Приведенный ниже рисунок 2.10 показывает, что общие типы данных из SQL преобразуются в группы по 5 правилам определения группами для небольшого набора типов данных, которые может принять SQLite.

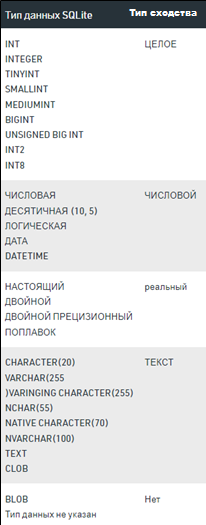


Рисунок 2.10 — Типы данных

ER-модель представляет собой высокоуровневую концептуальную модель данных, которая была разработана с целью упрощения задачи проектирования структур баз данных. На рисунке 2.11 представлена ER-диаграмма.

В процессе разработки базы данных были исследованы различные способы реализации для упрощения ее заполнения, была предпринята попытка автоматизировать заполнение таблиц в SQLite путем переноса файла с расписанием из деканата в таблицы базы данных, но этот метод было невозможно реализовать по следующим причинам:

- в таблице файла Excel, который предоставляет деканат, в одной ячейке может содержаться несколько объектов, как показано на рисунке 2.12;

- эти объекты не имеют имен, следовательно, к ним будет трудно обратиться программно. Для этого необходимо согласовать систему имен.

Поэтому реализация автоматизации импорта данных из такой таблицы невозможен без дополнительного согласования с разработчиками этой таблиц.

Наполним таблицы разработанной БД актуальной информацией, приведем в отчет пример наполнения. На рисунке 2.13 представлен пример таблицы «Общая таблица».

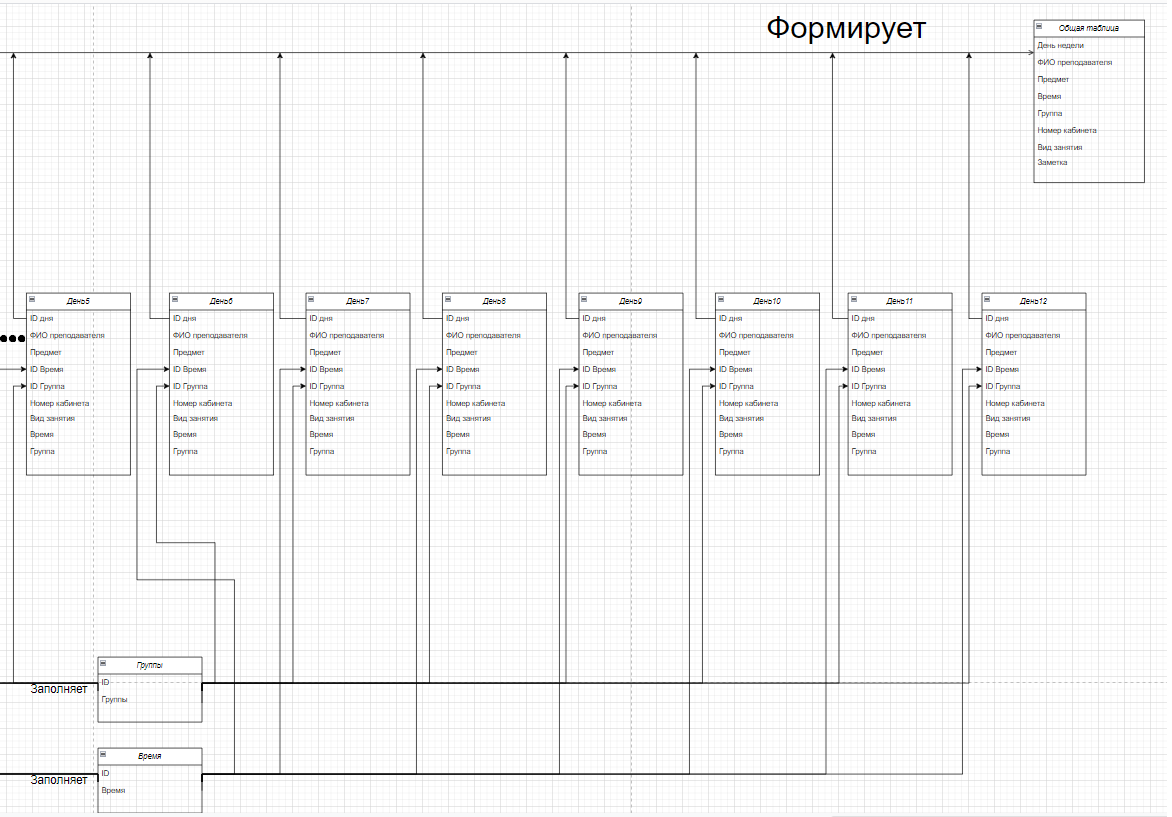


Рисунок 2.11 — ER-диаграмма

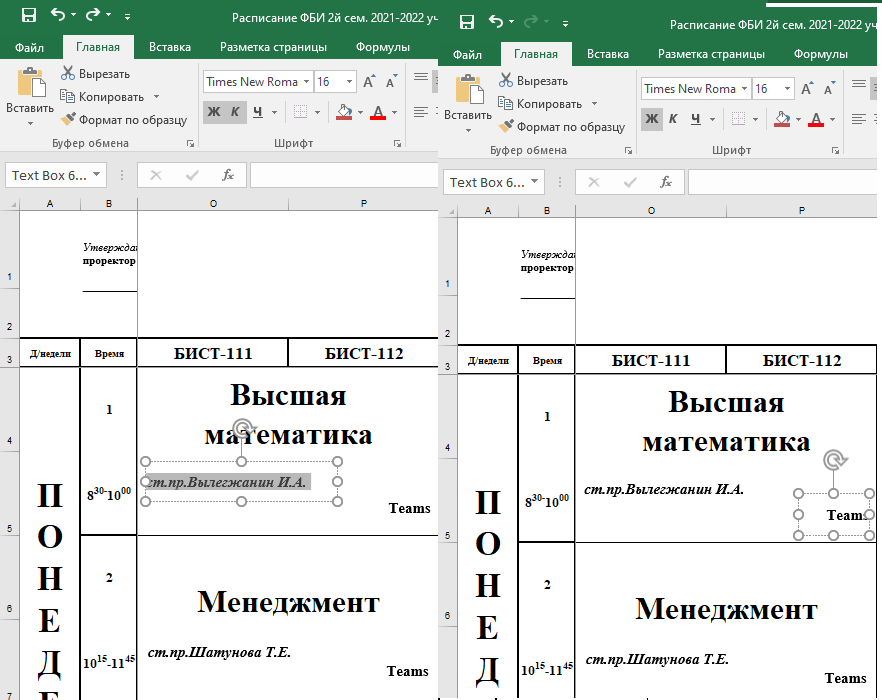


Рисунок 2.12 — Пример таблицы с несколькими объектами в ячейке

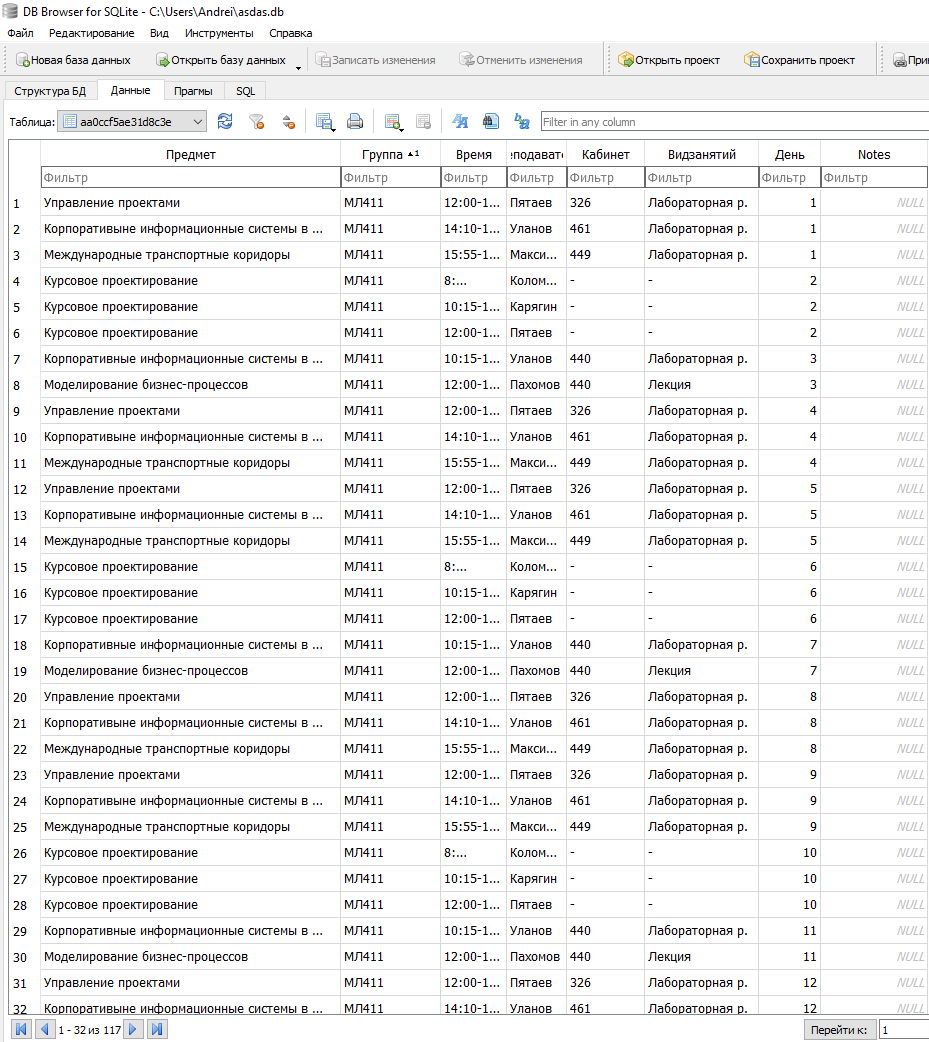


Рисунок 2.13 — Пример таблицы «Общая таблица»

На рисунке 2.14 представлен пример таблицы «День 1».

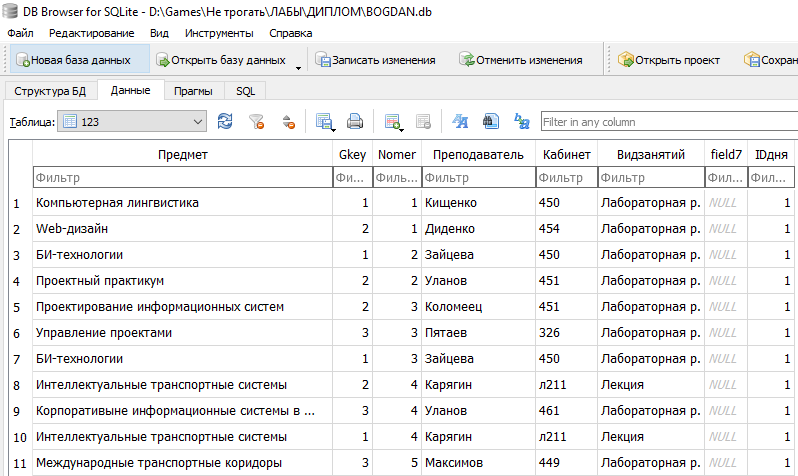


Рисунок 2.14 — Пример таблицы «День 1»

На рисунке 2.15 представлен пример таблицы «Группы».

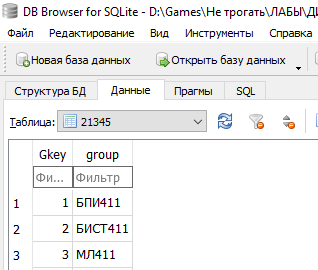


Рисунок 2.15 — Пример таблицы «Группы»

На рисунке 2.16 представлен пример заполнения словаря «Время».

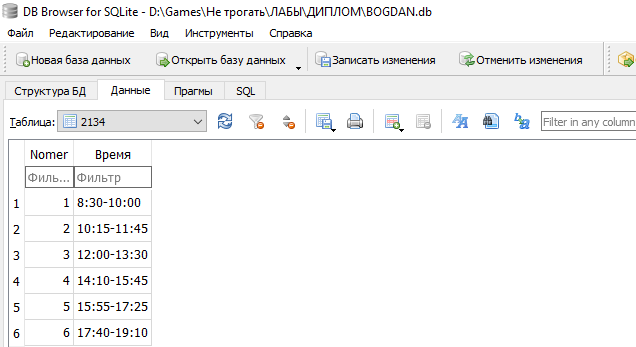


Рисунок 2.16 — Пример заполнения таблицы «Время»

Также в базе данных будут храниться заметки, которые пользователь может оставить. На рисунке 2.17 приведен код сохранения заметки в базу данных.

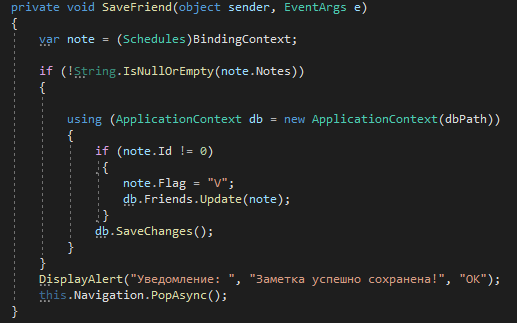


Рисунок 2.17 — Код сохранение заметок в БД

## 2.5 Администрирование базы

SQLite — это библиотека программного обеспечения, которая встраивается в само приложение и предоставляет систему управления реляционными базами данных. Lite в SQLite означает легкий с точки зрения настройки, администрирования базы данных и необходимых ресурсов.

Возможностей администрирования не так уж и много, так как SQLite — это встраиваемая СУБД, которая не предполагает наличие административной части. Во многих больших и известных СУБД возможности по администрирование и управлению пользователями и самими базами данных довольно широкие, так как эти СУБД работают по принципу клиент-сервер. SQLite не работает по данному принципу [12]. В процессе разработки БД был использован браузер баз данных DB Browser.

DB Browser for SQLite (DB4S) — это высококачественный визуальный инструмент с открытым исходным кодом для создания, проектирования и редактирования файлов баз данных, совместимых с SQLite.

DB4S предназначен для пользователей и разработчиков, которые хотят создавать, искать и редактировать базы данных. DB4S использует знакомый интерфейс, похожий на электронную таблицу, как показано на рисунке 2.18, поэтому сложные команды SQL не нужно изучать [13, 14, 15].

DB4S будет использоваться для импорта данных из Execl.

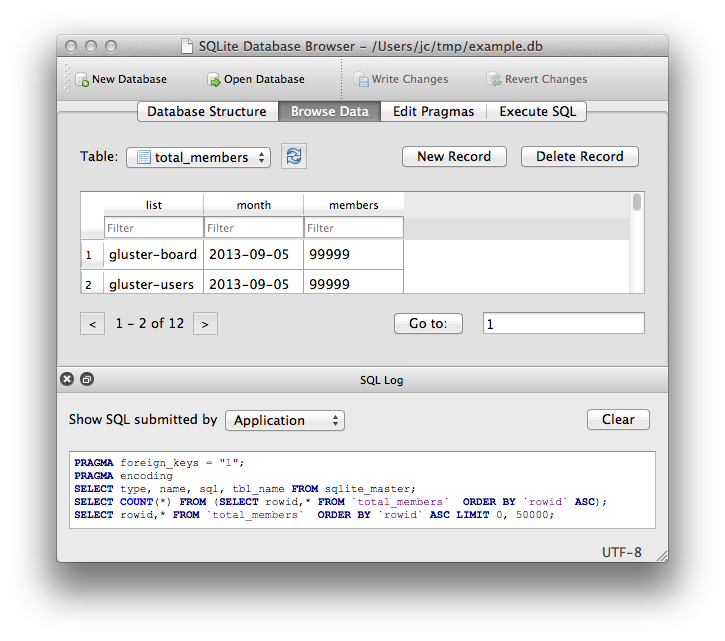


Рисунок 2.18 — Окно DB Browser

Элементы управления, доступные для пользователей:

- создание и сжатие файлов базы данных;

- создание, определение, изменение и удаление таблиц;

- импорт и экспорт записей в виде текста;

- импорт и экспорт баз данных из / в файлы SQL;

- выдавать SQL-запросы и проверять результаты.

## 2.6 Выводы к разделу

В данном разделе были подробно описаны обоснования выбора СУБД, а также же проектирование базы данных.

# **3 Руководство пользователя**

Конечный результат представляет собой базу данных для прототипа мобильного приложения «Расписания факультета». База данных была реализована по принципу, что пользователь, имеющий хоть какие-то познания в Microsoft Excel, мог с легкостью работать с ней и вносить изменения. Этим человеком будет администратор базы данных, для которого будет описано руководство пользователя. Эта база данных позволяет упрощать процесс заполнения и редактирования расписания, и приводит расписание к виду, под которое можно разработать приложение.

Для просмотра, заполнения и редактирования расписания, а также заметок, если это потребуется, необходимо установить Microsoft Excel, DB browser и SQLite.

Первым шагом необходимо заполнить справочники и первичные таблицы в файле MS Excel. Первичные таблицы имеют наименование День(Х), где Х — это номер дня от 1 до 12, как показано на рисунке 3.1. Заполнение первичных таблиц происходит с помощью полей, а также справочников, из которых с помощью формул заполняются поля время и группы, которые объявлены в таблицах 2.3-2.5 и показано на рисунке 3.2.



Рисунок 3.1 — Наименование таблиц

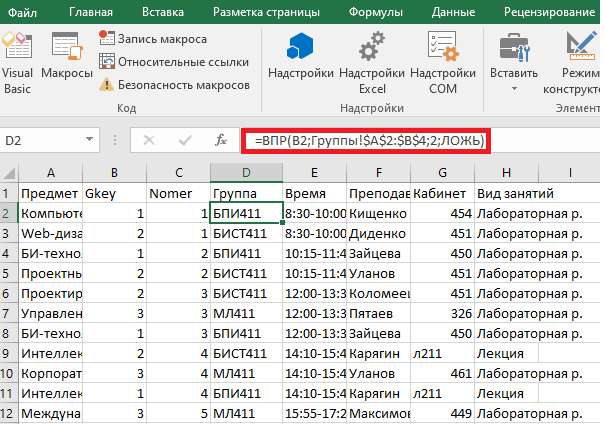


Рисунок 3.2 — Заполнение сущностей

Справочники показаны на рисунках 3.3-3.4.

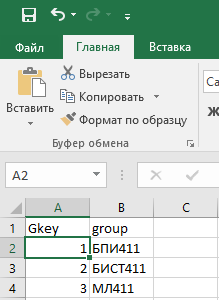


Рисунок 3.3 — Таблица группы

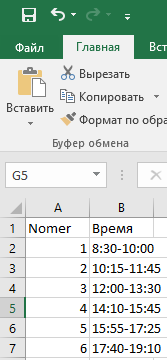


Рисунок 3.4 — Таблица время

Так как заполнение первичных таблиц происходит вручную, была реализована проверка с помощью встроенных инструментов Excel на корректность данных, а именно с помощью макросов, пример показан на рисунке 3.5. Например, если при заполнении первичной таблицы, администратор введет на одну пару одинаковые кабинеты, то ячейки, имеющие одинаковые данные, выделятся цветом, что и показано на рисунке 3.6.

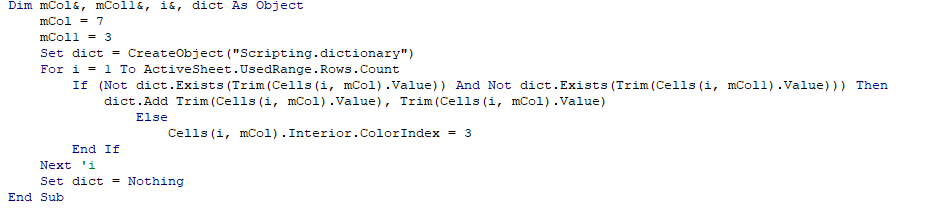


Рисунок 3.5 — Макрос проверки

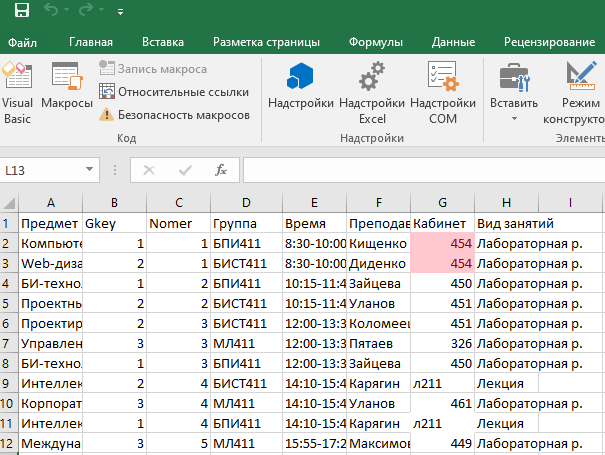


Рисунок 3.6 — Проверка корректности данных

Далее, как первичные таблицы будут заполнены, необходимо импортировать их в программу DB Browser, для дальнейшего переноса в SQLite. Импорт происходит таким образом, что все первичные таблицы сохраняются в файле с расширением CSV и файл открывается в DB4S.

После открытия файла появляется интерфейс, который показан на рисунке 3.7.

Затем необходимо удалить лишние данные, а именно столбцы Gkey и Nomer, и добавить столбец для заметок.

Итоговая «Общая таблица» представлена на рисунке 3.8.

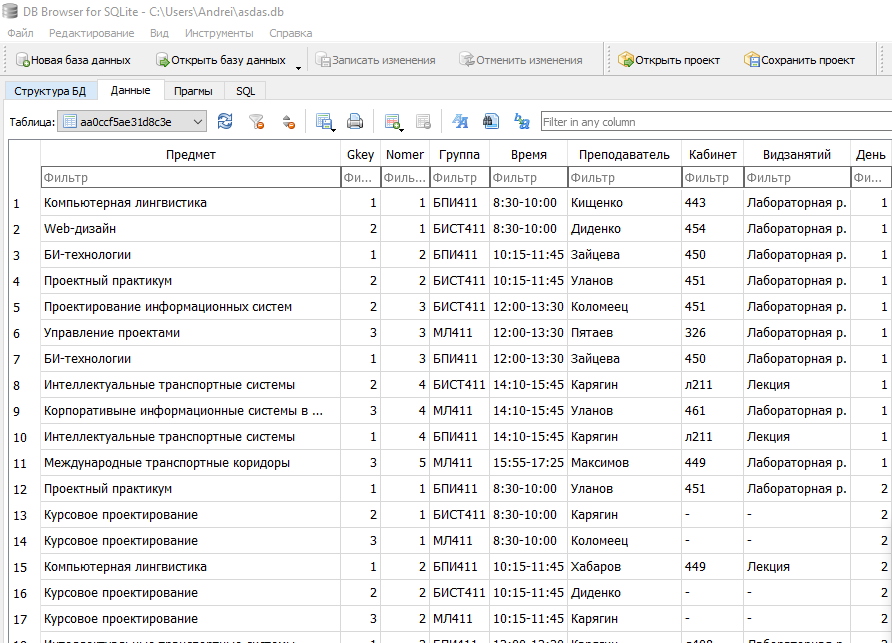


Рисунок 3.7 — Интерфейс DB browser

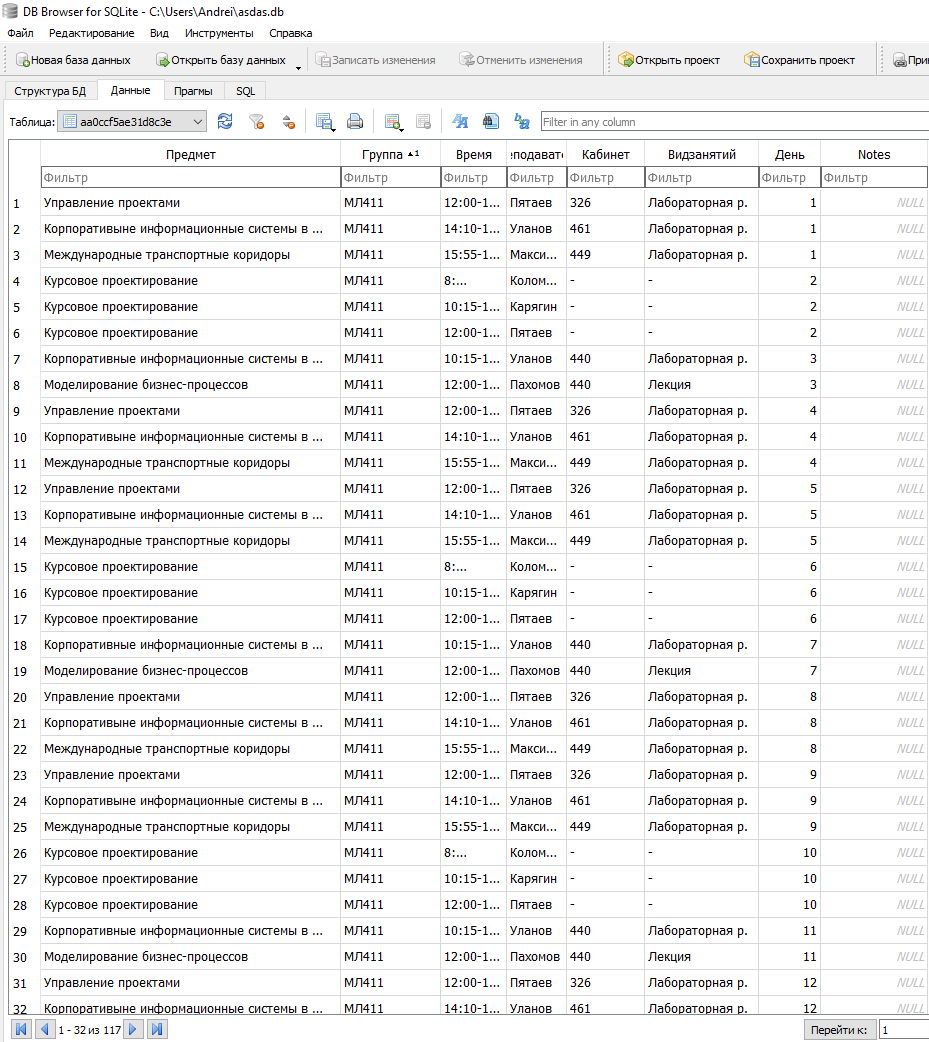


Рисунок 3.8 — Общая таблица

С помощью DB Browser общая таблица экспортируется в базу данных в файл .sql, как показано на рисунке 3.9.

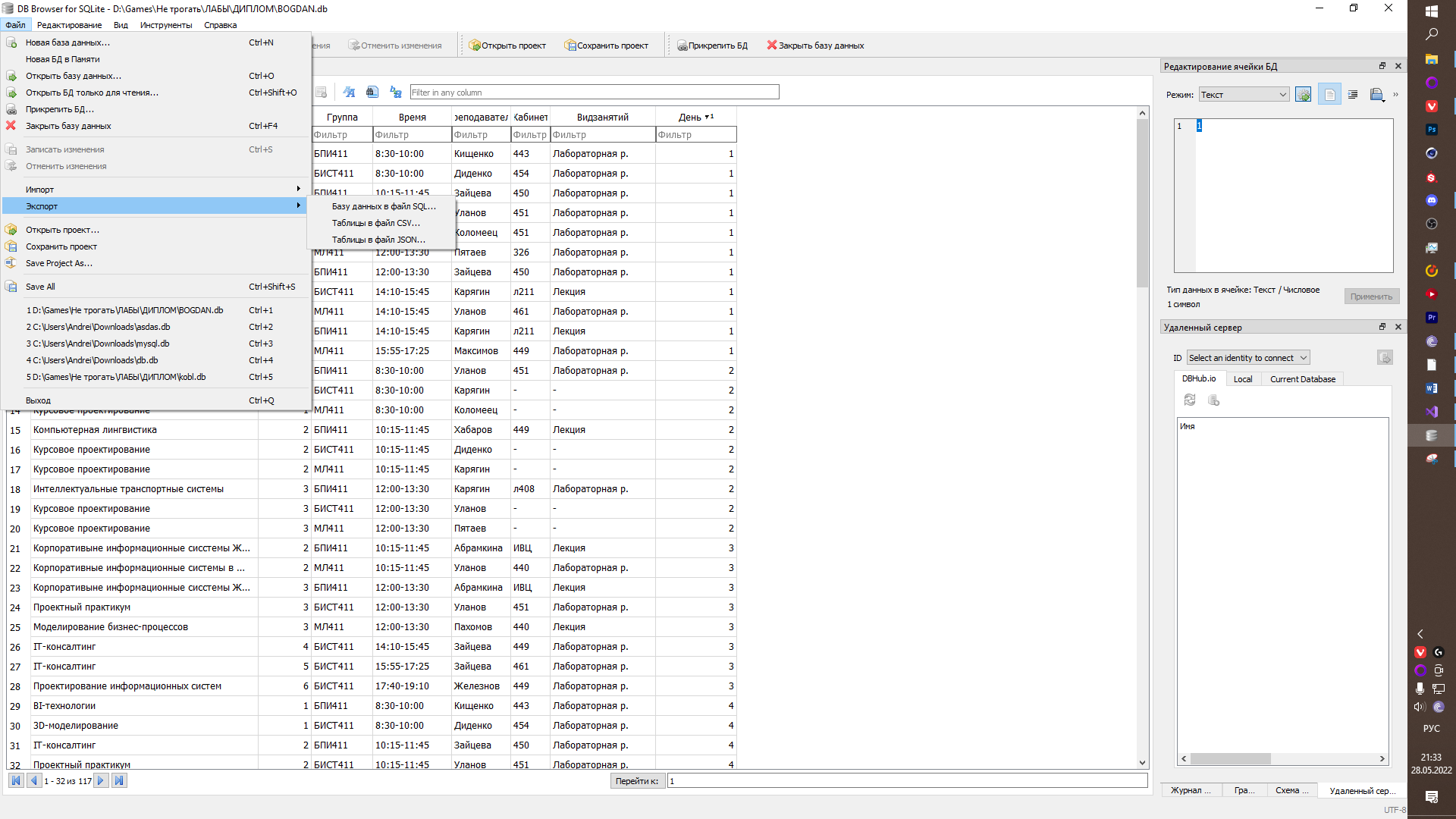


Рисунок 3.9 — Экспорт общей таблицы

В конечном итоге, файл, который будет запускаться мобильным приложением, показан на рисунке 3.10.



Рисунок 3.10 — Итоговый результат

В этом файле содержатся SQL-операторы для добавления записей в таблицы.

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения работы была разработана база данных для мобильного приложения согласно техническому заданию.

В процессе работы было сделано:

- проведен сравнение программных аналогов;

- проведен анализ деятельности факультета;

- исследованы процессы на факультетах (как есть);

- смоделированы бизнес-процессы;

- разработана база данных;

- описано руководство пользователя.

Все поставленные задачи были выполнены в полном объеме, а цель работы была достигнута.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Цели и задачи деятельности // Tolgas [Электронный ресурс]. - URL: https://www.tolgas.ru/org\_structura/fakul-technics/celi/ (Дата обращения 13.05.2022).
2. Бизнес-процессы // KPMS [Электронный ресурс]. - URL: https://www.kpms.ru/Automatization/BPM.htm (Дата обращения 13.05.2022).
3. UML- диаграммы// Сoderlessons [Электронный ресурс]. - URL: https://coderlessons.com/tutorials/akademicheskii/uchit-uml/uml-diagrammy-deiatelnosti (Дата обращения 15.05.2022).
4. API // Timeweb [Электронный ресурс]. - URL https://timeweb.com/ru/community/articles/chto-takoe-api (Дата обращения 15.05.2022).
5. Хостинг // Habr [Электронный ресурс]. - URL https://habr.com/ru/company/timeweb/blog/655547/ (Дата обращения 16.05.2022).
6. Системы управления базы данных // Habr [Электронный ресурс]. - URL: https://habr.com/ru/post/208400// (Дата обращения 20.05.2022).
7. Сравнение современных СУБД // Drach [Электронный ресурс]. - URL: https://drach.pro/blog/hi-tech/item/145-db-comparison (Дата обращения 20.05.2022).
8. SQL // Лекции.нет [Электронный ресурс]. - URL: https://lektsii.net/1-164575.html (Дата обращения 16.05.2022).
9. SQLite // SQLite [Электронный ресурс]. - URL: https://www.sqlite.org/index.html (Дата обращения 16.05.2022).
10. SQLite // Progler [Электронный ресурс]. - URL: https://progler.ru/blog/tipy-dannyh-sqlite (Дата обращения 16.05.2022).
11. SQLite // SQLite [Электронный ресурс]. - URL: https://www.sqlite.org/datatype3.html (Дата обращения 16.05.2022).
12. SQLite // SQLitetutorial [Электронный ресурс]. - URL: https://www.sqlitetutorial.net/what-is-sqlite/ (Дата обращения 16.05.2022).
13. DB Browser // GitHub [Электронный ресурс]. - URL: https://github.com/sqlitebrowser/sqlitebrowser/ (Дата обращения 16.05.2022).
14. DB Browser // Metanit [Электронный ресурс]. - URL: https://metanit.com/sql/sqlite/1.3.php (Дата обращения 16.05.2022).
15. DB Browser // Coder.Social [Электронный ресурс]. - URL: https://coder.social/sqlitebrowser/sqlitebrowser?type=intro (Дата обращения 16.05.2022).