

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра математического и компьютерного моделирования

Построение информационной системы для распределения

учебной нагрузки учителей

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы

направление 09.03.03 — Прикладная информатика

механико-математического факультета

Гнеушева Дмитрия Владимировича

Научный руководитель

доцент, к.ф.-м.н.

С.В. Иванов

Зав. кафедрой

зав. каф., д.ф.-м.н., доцент

Ю.А. Блинков

Саратов 2025

Введение. Направленность на цифровизацию является ключевым фактором, который кратно усиливает актуальность использования информационных систем (ИС) в современном мире. Цифровизация – это не просто внедрение отдельных ИС, а трансформация всех аспектов деятельности организации или общества с использованием цифровых технологий. Эта трансформация создает новые потребности и возможности, которые ИС помогают реализовать.

Необходимо отметить и процесс автоматизации и роботизации. Цифровизация требует автоматизации все большего количества процессов, чтобы повысить эффективность и снизить затраты. Роботизация физических и умственных задач становится возможной благодаря цифровым технологиям и требует интеграции с ИС. ИС контролируют и управляют автоматизированными процессами и роботами, обеспечивая их эффективную работу.

Однако с ростом использования цифровых технологий происходит и увеличение требований клиентов к цифровым продуктам. Повышаются ожидания клиентов в отношении скорости, удобства и персонализации обслуживания. Клиенты ожидают бесшовного опыта взаимодействия с компанией через все каналы. ИС позволяют предоставлять клиентам персонализированные предложения, быстро реагировать на их запросы и обеспечивать многоканальное обслуживание.

Всё вышеперечисленное обосновывает актуальность развития методических и инструментальных средств для ИС.

Целью данной работы является разработка ИС для распределения учебной нагрузки учителей.

Для достижения данной цели поставлены задачи:

1. Изучить теорию ИС и провести анализ предметной области распределения учебной нагрузки учителей.
2. Средствами UML спроектировать ИС для распределения учебной нагрузки учителей.
3. Спроектировать и создать базу данных для распределения учебной нагрузки учителей.
4. Реализовать ИС распределения учебной нагрузки учителей.

Первый раздел посвящён теоретическим аспектам ИС. ИС представляет собой совокупность взаимосвязанных элементов, которые совместно функционируют для сбора, обработки, хранения и передачи информации.

Помимо этого в первом разделе анализируется предметная область. Основными компонентами предметной области распределения учебной нагрузки учителей являются:

- Уроки: Урок – номенклатурная единица расписания, описывающая то, какой предмет преподается какому классу или подгруппе класса, в какое время происходит занятие, в каком кабинете, и какой учитель его ведет. При этом необходимо отметить, что учителя могут вести только ограниченное количество предметов, так, например, учитель истории чаще всего не может вести физику, но может вести обществознание.
- Процессы: изменения расписания в определенные периоды.
- Пользователи: основными пользователями получившейся учебной нагрузки, и, как следствие, расписания, являются ученики и учителя.
- Администратор: Администратор – некоторый сотрудник образовательного учреждения, непосредственно составляющий распределение учебной нагрузки и вносящий изменения в нее.

Во **втором разделе** рассматривается теория моделирования предметной области средствами UML. UML – это стандартизованный графический язык моделирования, предназначенный для визуализации, спецификации, конструирования и документирования артефактов программных систем. UML предоставляет набор нотаций и семантик, позволяющих описывать различные аспекты системы, от ее структуры до поведения. UML обеспечивает общую платформу для коммуникации между участниками процесса разработки программного обеспечения, позволяя создавать понятные и непротиворечивые модели, которые служат основой для проектирования и реализации сложных систем.

Также происходит рассмотрение теории следующих типов диаграмм:

- Диаграмма вариантов использования;
- Диаграмма последовательности;
- Диаграмма активности.

После рассмотрения теории каждого из типов происходит построение соответствующей диаграммы для рассматриваемой предметной области.

Так, для разрабатываемой ИС диаграмма вариантов использования отражает необходимые возможности пользователей системы. В данном случае диаграмма отражает разграничение возможностей между типами пользователей. Диаграмма вариантов использования для ИС распределения учебной нагрузки учителей показана в соответствии с рисунком 1.

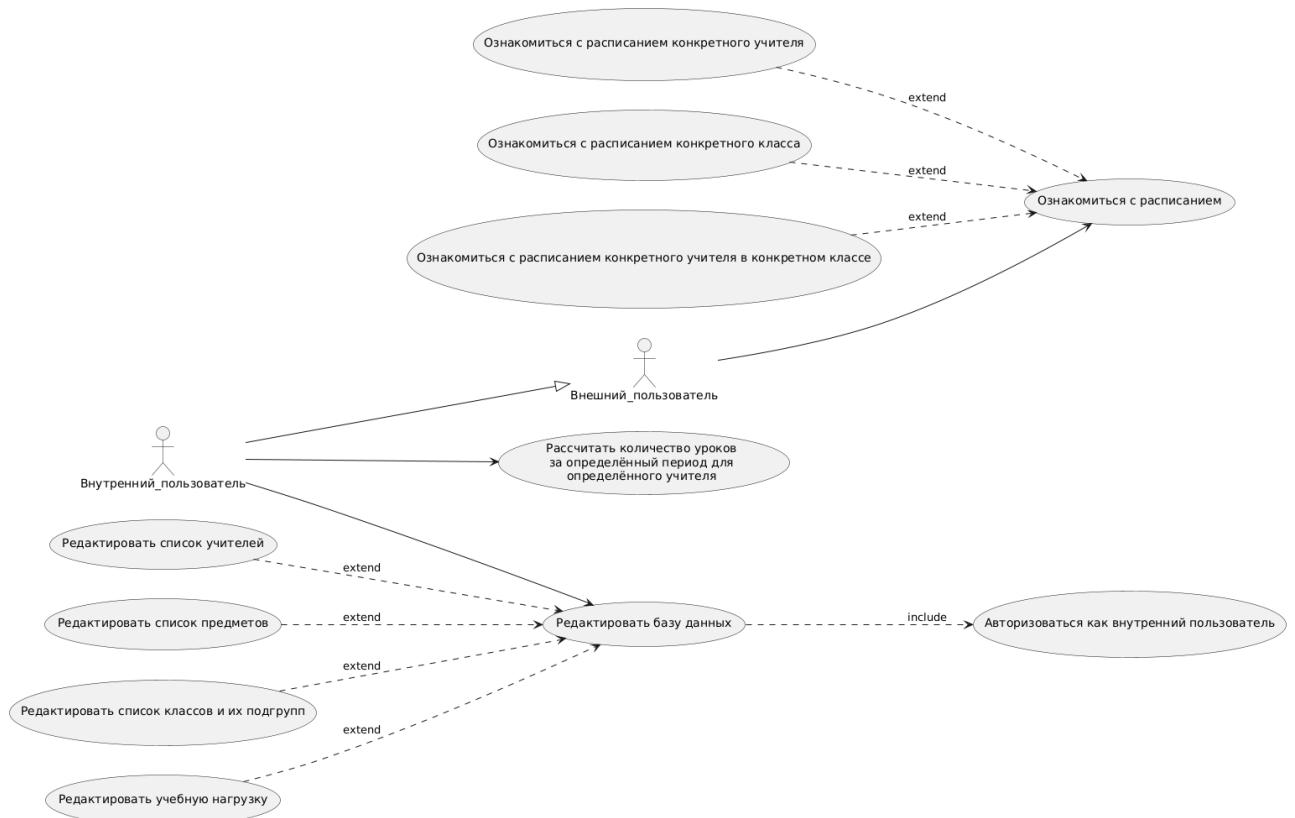


Рисунок 1 — Диаграмма вариантов использования

Принципиальной разницей между типами пользователей является необходимость аутентификации для внутреннего пользователя. Данное условие обеспечивает невозможность внесения изменений в данные со стороны сторонних пользователей, не являющихся напрямую ответственными за формирование распределения учебной нагрузки внутри учебного заведения.

В третьем разделе происходит рассмотрение теории построения базы данных для ИС.

База данных (БД) определяется как именованная, структурированная и интегрированная совокупность данных, описывающая состояние объектов и взаимосвязи между ними в заданной предметной области.

Система управления базами данных (СУБД) – это программное обеспечение, предназначенное для создания, модификации и сопровождения БД.

В рамках данной работы используется реляционная СУБД SQLite3. Взаимодействие между пользователем и реляционной БД осуществляется посредством языка структурированных запросов (SQL).

SQL – это стандартизованный язык программирования, предназначенный для выполнения операций создания, модификации, поиска и извлечения данных, хранящихся в реляционной БД, управляемой соответствующей СУБД.

В качестве средства для проектирования БД используется ER-диаграмма. Далее в разделе рассматривается теория построения ER-диаграмм.

ER-диаграмма (entity-relationship diagram) – это графическое представление структуры данных системы, демонстрирующее сущности, их атрибуты и взаимосвязи между ними.

Основными сущностями для БД распределения учебной нагрузки являются:

1. Предметы (subjects) – названия учебных дисциплин, преподаваемые в учебном учреждении.
2. Учителя (teachers) – данные об учителях, работающих в учебной организации.
3. Классы (classes) – названия классов, обучение которых происходит в учебном учреждении.
4. Подгруппы (subgroups) – данные о подгруппах, на которые делятся классы.
5. Уроки (lessons) – данные об уроках, которые проводятся в образовательном учреждении.

Необходимо отметить, что для каждого учителя есть некоторый список предметов, которые он может преподавать. Следовательно с целью реализации связи «многие ко многим» необходимо добавить таблицу связи учителей и предметов.

ER-диаграмма для ИС показана в соответствии с рисунком 2.

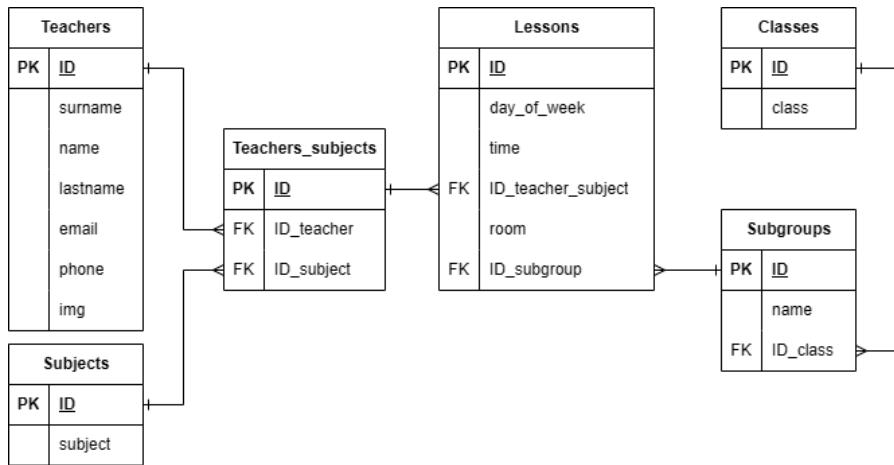


Рисунок 2 — ER-диаграмма

Эти таблицы связаны между собой с помощью ключей и отношений для обеспечения целостности данных и эффективного взаимодействия между различными компонентами БД.

Далее в разделе рассматривается проектирование БД с помощью программных средств. Для создания ИС использовался язык программирования Python. Для взаимодействия с БД используется библиотека sqlite3. Для связи с БД была реализована функция `create_connection_all()`. В этой функции используется команда установки соединения с БД.

Для каждой таблицы реализована функция создания в случае отсутствия. В каждой такой функции создаётся объект курсора, который выполняет некоторую последовательность команд. Разрешается использование внешних ключей и, в случае отсутствия в БД таблицы, происходит запрос на её создание, после чего изменения, внесённые в БД, сохраняются и соединение с БД закрывается.

Также для каждой сущности определяются функция перезаписи. При этом происходит удаление всех (или некоторых удовлетворяющих условию) строк таблицы, и последующее добавление строк в таблицу.

Данная функция принимает на вход DataFrame с данными из административной панели, с которых и происходит запись. На случай отсутствия в строке индивидуального номера написано условие его генерации как превышающего на 1 максимальный или равного 1 в случае, если максимальный

номер в объекте типа DataFrame отсутствует. Помимо этого в функции задаётся вывод сообщения о сохранении изменений.

Также для каждой из таблиц была реализована функция считывания данных из БД в DataFrame. В случае, если в таблице имеются внешние ключи, то происходит объединение таблиц с помощью левого внешнего соединения с сохранением всех элементов рассматриваемой в функции таблицы.

Для таблицы учителей также реализована функция добавления значений в поле фотографии, выполняющая к тому же функцию заполнения таблицы связи предметов и учителей.

В четвёртом разделе происходит описание процесса разработки ИС.

Для визуализации ИС была выбрана форма сайта, как наиболее мобильная и легко развёртываемая. Для реализации ИС на языке программирования Python был использован фреймворк Streamlit.

Для выбора области просмотра расписания или панели управления администрации была добавлена боковая панель с названием ИС и её возможностями. Помимо этого при запуске ИС происходит вызов функций создания таблиц в БД, которые проверят БД на наличие этих таблиц и создают их в случае, если в БД их нет.

Для реализации возможности фильтрации данных по учителю и по классу в области просмотра были добавлены два виджета выбора среди вариантов, являющихся элементами таблиц «Учителя» и «Классы» соответственно. В виджетах имеется функция выбора пустого результата, позволяющая сбросить фильтрацию.

Далее создаётся таблица, содержащая в себе расписание. Данная таблица состоит из 13 строк (не считая заголовка) и 6 столбцов (не считая индексов). Каждая строка отвечает за номер урока, в качестве её индекса используется время проведения урока. Каждый столбец отвечает за день недели с понедельника по субботу. Таким образом получившаяся ячейка на пересечении некоторого столбца и некоторой строки отвечает за урок происходящий в конкретное время в конкретный день недели.

Таблица заполняется данными из таблицы уроков. После того, как задаётся хотя бы один из двух фильтров, происходит отбор данных из таблицы уроков. В случае, если в поле учителя введено не пустое значение, то перед

таблицей выводятся данные об учителе, а в ячейках расписания данные о предмете, кабинете и классе или группе. В случае, если выбран только класс, то в ячейках собираются данные о предмете, учителе, кабинете и подгруппе в случае, если они есть. В случае, если не пустые значения принимают и класс, и учитель, то в ячейки вносятся данные о предметах, кабинетах и группах при наличии. Расписание учителя показано в соответствии с рисунком 3.

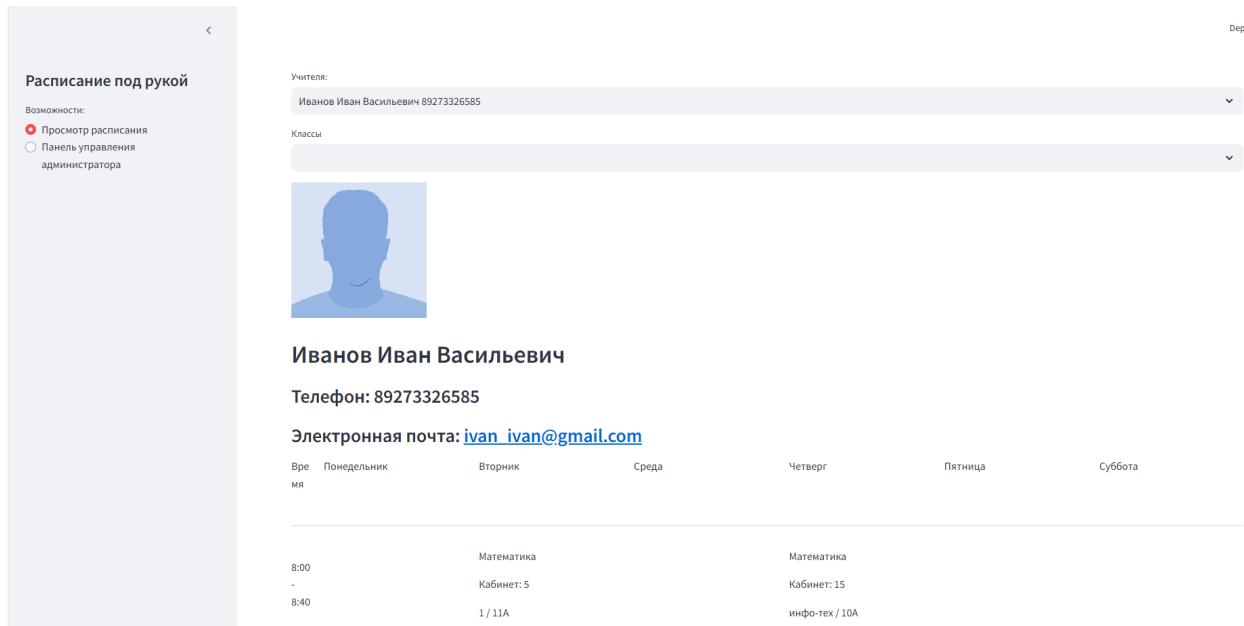


Рисунок 3 — Расписание учителя

Для облегчения идентификации подгрупп в случае, если класс не выбран подгруппы выводятся через косую черту с классом, к которому они принадлежат.

В панели администратора для реализации защиты данных был добавлен виджет проверки пароля. Пароль вводится в программе и предполагается, что пароль регулярно меняется и известен только администраторам учебного заведения. В случае, если введён неверный пароль, пользователь получает сообщение «Неверный пароль!», в противном случае пользователь получает сообщение «Вход выполнен» и доступ к редактированию БД.

В административной панели было реализовано редактирование таблицы предметов. Для этого используется виджет редактирования объектов типа DataFrame – `st.data_editor(...)`, в котором пользователю доступен к редактированию только столбец названий предметов. Данный столбец реализо-

ван как текстовое поле с обязательным заполнением. Далее была реализована кнопка «Сохранить изменения», при нажатии которой происходит вызов функции сохранения изменений в таблицу предметов, в которую передаётся DataFrame с изменениями, внесёнными пользователем. Редактирование таблицы классов реализовано схожим образом.

Функция редактирования таблицы подгруппы выполнена с помощью виджета внесения изменений в объекты типа DataFrame, где внутренний пользователь не имеет возможности редактирования индивидуального номера подгруппы. Столбец названия подгруппы реализован как текстовое поле, а столбец класса реализован как поле выбора из объектов таблицы классов. В качестве внешних показателей выбора используются значения названий классов. В функции внесения изменений в таблицу подгрупп происходит поиск класса по его названию, возможный за счёт уникальности названия. После этого происходит замена названия на индивидуальный номер класса, который, в свою очередь, и сохраняется в БД подгруппы. В связи с этим в функции вывода данных из БД подгрупп происходит объединение таблиц подгрупп и групп, в результате которого значение индивидуального номера группы заменяется на значение названия группы.

Для редактирования таблицы учителей был также реализован виджет редактирования объектов типа DataFrame, где столбцы фамилии, имени, отчества, электронной почты и номера телефона реализованы как текстовые поля. Столбцы фамилии, имени и номера обязательны к заполнению. Далее реализована кнопка сохранения изменений, нажатие на которую вызывает функцию сохранения изменений в БД. При этом в БД остаётся незаполненным столбец ссылки на фото.

Важно, что для электронной почты и номера телефона были добавлены валидаторы – проверки на соответствие введённых значений некоторым регулярным выражениям. Для электронной почты данные выражения представляют собой набор некоторых символов без пробелов, следующий за ним знак «@», затем некоторые буквенные символы, знак точки и последовательности из 2 или 3 букв на конце. Для номера телефона выражение проверяется на начало с знака «+» или цифры и последующего использования только цифр, скобок или дефиса.

Необходимо отметить, что виджет редактирования объектов типа DataFrame не предоставляет возможности множественного выбора из некоторых вариантов, а также не предоставляет возможности добавления фотографии. Для решения этой проблемы на странице была создана ещё одна таблица с помощью функции формирования столбцов `st.columns(...)`. Данная таблица имеет записи фамилии, имени и отчества учителей, однако возможность их редактирования отсутствует. Следующий столбец состоит из виджетов множественного выбора предметов, которые могут вести учителя. Результаты этих виджетов сохраняются в словарь в качестве значений, ключами при этом являются значения индивидуального номера учителя. Далее следует столбец из виджетов загрузки файлов, принимающие только файлы с расширением png и jpg. Это позволяет отсеять загрузку файлов, не являющихся фотографиями. После загрузки файлов подходящего расширения, они сохраняются в папке media ИС. Полученные пути сохраняются в словарь с ключом в виде индивидуального номера учителя. Для удобства администратора был добавлен ещё один столбец, в котором находится демонстрация фото, добавленного в виджете предыдущего столбца.

Результирующая страница редактирования данных об учителях показана в соответствии с рисунком 4.

Кнопка, находящаяся в конце второй таблицы редактирования, связана с описанной выше функцией добавления значений в поле файла таблицы учителей и заполнения таблицы связи предметов и учителей. В функцию передаётся DataFrame с данными об учителях, а также два словаря данных о предметах и данных о фотографиях. В этой функции происходит отбор индивидуального номера предмета из таблицы предметов по его названию. Полученный индивидуальный номер связывается с индивидуальным номером учителя в таблице связи учителя и предмета.

В связи с тем, как происходит сбор новых данных, становится необходимым рассмотреть вопрос выгрузки данных из БД. Для корректного отображения ранее введённой информации создана дополнительная функция считывания строк из БД, задающая объединение таблиц учителей, предметов и связи учителей и предметов. По результирующему объекту типа DataFrame происходит сбор словаря с индивидуальным номером учителя в качестве клю-

ча и списком названий предметов в качестве значения. Также происходит формирование словаря со значениями, являющимися путём до файла с фото. Получившиеся словари используются в качестве исходных значений для виджетов второй таблицы.

Фамилия	Имя	Отчество	Электронная почта	Номер телефона
Иванов	Иван	Васильевич	ivan_ivan@gmail.com	89273326585
Петрова	Мария	Игоревна	petrova@mail.ru	+79036541111
Степанова	Ольга	None	None	+7-900-000-00-00

Сохранить изменения

Фамилия Иванов	Имя Иван	Отчество Васильевич	Предметы Математика	Файл с фото Drag and drop file here Limit 200MB per file • JPG, PNG, JPEG	Предпросмотр фото
Петрова	Мария	Игоревна	Русский язык	Drag and drop file here Limit 200MB per file • JPG, PNG, JPEG	
Степанова	Ольга	None	История Обществознание	Drag and drop file here Limit 200MB per file • JPG, PNG, JPEG	

Сохранить фото

Рисунок 4 — Страница редактирования данных об учителях

Редактирование данных об уроках основано на построении элементов, схожем с тем, что реализовано на странице просмотре расписания. Администратору предлагается выбор из учителей, внесённых в БД. По умолчанию открывается расписание первого учителя из БД. Каждая ячейка состоит из виджета единичного выбора предмета, виджета текстового поля ввода кабинета и виджета выбора класса или подгруппы. Для реализации последнего

происходит считывание информации из таблиц классов и подгрупп, в таблицу подгрупп добавляется дополнительный столбец, содержащий в себе название подгруппы и через косую черту название класса, к которому она принадлежит. Получившийся столбец представляется как список и объединяется с представленным как список столбцом названий классов. Получившиеся данные являются возможными значениями для виджета выбора класса или подгруппы. По результату данной таблицы получается DataFrame с данными об изменениях в уроках выбранного ранее учителя. Таблица редактирования уроков показана в соответствии с рисунком 5.

The screenshot shows a weekly lesson plan for a teacher named Ivanov Ivan Vasilevich. The interface is in Russian. At the top, there is a header bar with the teacher's name and ID: 'Иванов Иван Васильевич 89273326585'. Below this is a table with columns for each day of the week: Время (Time), Понедельник (Monday), Вторник (Tuesday), Среда (Wednesday), Четверг (Thursday), Пятница (Friday), and Суббота (Saturday). Each day has three rows of dropdown menus for 'Предмет' (Subject), 'Кабинет' (Cabinet/Room), and 'Класс' (Class). For Monday, the subject is 'Математика' (Mathematics), room is '5', and class is '1 / 11A'. For Thursday, the subject is 'Математика' (Mathematics), room is '15', and class is 'инфо-тех / 10A' (Info-tech / 10A).

Рисунок 5 — Страница редактирования данных об уроках

С помощью кнопки «Сохранить изменения» DataFrame передаётся в функцию сохранения изменений в таблице учителей. При этом необходимо отметить, что данные о времени хранятся как значения от 0 до 13, а данные о днях недели – от 1 до 6. Это выполнено для удобства хранения информации и уменьшения занимаемой памяти.

Заключение. В ходе работы:

1. Была изучена теория ИС и был проведён анализ предметной области распределения учебной нагрузки учителей.
2. Средствами UML была спроектирована ИС для распределения учебной нагрузки учителей.
3. Была спроектирована и создана база данных для распределения учебной нагрузки учителей.
4. Была реализована ИС для распределения учебной нагрузки учителей.