Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Сибирский государственный индустриальный университет»

Кафедра прикладные информационные технологии и программирование

**Курсовой проект**

**Разработка информационной системы для составления учебного расписания СибГИУ и сайта для удобного просмотра расписания в программе Visual Studio**

**Практическая работа «Техническое задание на разработку**

**информационного (программного) продукта»**

Выполнили:

Обучающиеся гр. К-ИСП-20-2

\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ Токмаков А.К.

(дата) (подпись) (инициалы, фамилия)

\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_Перунов Я. С.

(дата) (подпись) (инициалы, фамилия)

\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_Швецов К.А.

(дата) (подпись) (инициалы, фамилия)

\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ Михайлов М.К.

(дата) (подпись) (инициалы, фамилия)

Преподаватель:

Белавенцева Дарья Юрьевна

Дата проверки:

Оценка: Подпись:

Новокузнецк, 2022г

**Содержание**

[**Введение** 4](#_Toc131958038)

[**Инициация планирования** 5](#_Toc131958039)

[**Устав проекта** 6](#_Toc131958040)

[**Задачи проекта** 6](#_Toc131958041)

[**Ограничения проекта** 6](#_Toc131958042)

[**Ресурсы проекта** 7](#_Toc131958043)

[**Риски проекта** 7](#_Toc131958044)

[**Последовательность работ по проекту** 8](#_Toc131958045)

[**Команда проекта** 8](#_Toc131958046)

[**Обязанности участников проекта** 9](#_Toc131958047)

[**Техническое задание** 10](#_Toc131958048)

[**1.** **Введение** 10](#_Toc131958049)

[**2.** **Основания для разработки** 10](#_Toc131958050)

[**3.** **Назначение разработки** 10](#_Toc131958051)

[**4.** **Требования к программе или программному изделию.** 11](#_Toc131958052)

[**4.1** **Требования к функциональным характеристикам.** 11](#_Toc131958053)

[**4.2** **Требование к надежности** 11](#_Toc131958054)

[**4.3** **Условия эксплуатации** 12](#_Toc131958055)

[**4.4** **Требования к составу и параметрам технических средств.** 12](#_Toc131958056)

[**4.5** **Требования к информационной и программной совместимости.** 12](#_Toc131958057)

[**4.6** **Требования к маркировке и упаковке.** 13](#_Toc131958058)

[**4.7** **Требования к транспортированию и хранению.** 13](#_Toc131958059)

[**4.8** **Специальные требования.** 13](#_Toc131958060)

[**5.** **Требования к программной документации.** 13](#_Toc131958061)

[**6.** **Стадии и этапы разработки.** 14](#_Toc131958062)

[**7.** **Порядок контроля и приемки** 15](#_Toc131958063)

[**Обоснование выбора программных и технических средств** 16](#_Toc131958064)

[**1** **Обоснование инструментов разработки** 16](#_Toc131958065)

[**2** **Средства разработки интерфейса** 16](#_Toc131958066)

[**3** **Средства редактирование кода** 17](#_Toc131958067)

[**4** **Платформа для хостинга сайта** 19](#_Toc131958068)

[**5** **СУБД** 20](#_Toc131958069)

[**Описание информационного обеспечения** 22](#_Toc131958070)

[**1.** **Use-case** 22](#_Toc131958071)

[**2.** **Функциональная диаграмма** 23](#_Toc131958072)

[**3.** **Схема базы данных** 26](#_Toc131958073)

[**Описание алгоритмического обеспечения проекта** 28](#_Toc131958074)

**Введение**

Наш проект очень нужен всем, все его ждут, а мы делаем доки. И, так на страницу минимум

Разработать информационную систему для составления учебного расписания СибГИУ и сайта для удобного просмотра расписания в программе Visual Studio. В дальнейшем внедрить данный комплекс ПО в СибГИУ.

**Инициация планирования**

Что здесь писать?

**Устав проекта**

Наименование проекта: «Разработка информационной системы для составления учебного расписания СибГИУ и телеграмм бота для расписания в программе Visual Studio»

Дата начала проекта: 23.09.2022

Дата окончание проекта: 30.05.2023

Цель проекта высшего уровня – разработать информационную систему для составления учебного расписания СибГИУ приложения и сайта также телеграмм бота для расписания в программе Visual Studio.

## **Задачи проекта**

1. Изучить конечного пользователя информационной системы и его предпочтения.
2. Изучить взаимодействие баз данных между собой, при разных условиях использования и выбрать наиболее подходящий.
3. Проектирование базы данных;
4. Разработка графического интерфейса пользователя;
5. Разработка телеграмм бота;
6. Разработка приложения и сайта для просмотра расписания;
7. Подключение базы данных к приложению и телеграмм боту.

## **Ограничения проекта**

Ограниченные финансовые ресурсы на разработку информационной системы для составления учебного расписания СибГИУ, сайта и телеграмм бота для расписания.

Таблица 1 – Заинтересованные стороны проекта

|  |  |  |
| --- | --- | --- |
| № п/п | Заинтересованная сторона | Интерес |
| 1 | Участники проекта | Разработка успешно функционирующей информационной системы |
| 2 | Руководитель проекта | Успешная разработка и реализация проекта |
| 3 | Пользователь сервиса | Удобный просмотр расписания занятий |

## **Ресурсы проекта**

1. Временные ресурсы: 8 месяцев.
2. Материальные ресурсы: ПК, сеть интернет, Visual Studio 2019.
3. Трудовые ресурсы: 1 руководитель, 1 разработчик, 1 дизайнер, 1 тестировщик.

## **Риски проекта**

1. Ошибки планирования заданий. Для их минимизации используются плавающие сроки сдачи работ.
2. Нехватка времени на реализацию поставленной задачи из-за плотного графика.
3. Невыполнения работ ответственным за его выполнения, преднамеренно или из-за неправильного распределения личного времени. Для их упразднения используются объяснительные.
4. Недостаточно знаний для реализации поставленных задач.
5. Неактуальность ИС для составления расписания.
6. Нестабильная работа ИС.
7. Появление других ИС с более лучшим функционалом и поддержкой.
8. Аппаратные сбои, утеря важных данных по проекту.
9. Другие риски, которые невозможно предсказать.

## **Последовательность работ по проекту**

Таблица 2 – Последовательность работ по проекту

|  |  |
| --- | --- |
| № п/п | Этапы проекта |
| 1 | Инициация проекта |
| 1.1 | Разработка устава проекта |
| 2 | Планирование проекта |
| 2.1 | Определение последовательности работ |
| 2.2 | Составление диаграммы Ганта |
| 2.3 | Составление технического задания |
| 3 | Реализация проекта |
| 3.1 | Выбор платформы проекта |
| 3.2 | Выбор языка программирования |
| 3.3 | Выбор и обоснования методов реализации |
| 3.4 | Проведение исследования конечного пользователя продукта |
| 3.5 | Составление календаря созвонов и отчета проделанной работы |
| 3.6 | Разработка БД |
| 3.7 | Разработка front-end части ИС |
| 3.8 | Разработка back-end части ИС |
| 3.9 | Создание сайта для просмотра расписания |
| 3.10 | Подведение итогов работы по разработке информационной системы для составления учебного расписания СибГИУ, сайта и телеграмм бота для расписания |
| 4 | Завершение проекта |
| 4.1 | Создание отчета и презентации |
| 4.2 | Демонстрация и защита проекта |

## **Команда проекта**

Научный руководитель проекта: Гасымов Р.Р., Белавенцева Д.Ю.

Руководитель-разработчик проекта: Токмаков Антон Константинович.

Разработчик-дизайнер: Михайлов Максим Константинович

Дизайнер-разработчик: Швецов Кирилл Антонович

Тестировщик-документовед: Перунов Ярослав Сергеевич

## **Обязанности участников проекта**

Таблица 3 - Обязанности участников проекта

|  |  |  |
| --- | --- | --- |
| Должность | Участник | Обязанности |
| Руководитель  проекта | Токмаков Антон Константинович | Руководство проектом |
| Разработка базы данных |
| Разработка back-end части ИС |
| Исполнитель проекта | Михайлов Максим Константинович | Разработка back-end части сайта |
| Разработка документации проекта |
| Разработка телеграмм бота |
| Оценщик проекта | Перунов Ярослав Сергеевич | Разработка методов тестирование проекта |
| Разработка front-end части сайта |
| Разведчик проекта | Швецов Кирилл Антонович | Разработка документации проекта |
| Разработка back-end части сайта |

**Техническое задание**

* + 1. **Введение**

Разработать информационную систему для составления учебного расписания СибГИУ и сайта для удобного просмотра расписания в программе Visual Studio. В дальнейшем внедрить данный комплекс ПО в СибГИУ.

* + 1. **Основания для разработки**

Документ, на основании которого ведется разработка Устав проекта по разработке информационной системы для составления учебного расписания СибГИУ и телеграмм бота для расписания в программе Visual Studio, дата 23.09.2022.

Тема проекта: Разработка информационной системы для составления учебного расписания СибГИУ и телеграмм бота для расписания в программе Visual Studio

Отсутствие информационной системы составление расписания в СибГИУ, а также не удобное его расположение для ежедневного просмотра.

* + 1. **Назначение разработки**

Демоверсия информационной системы является первым этапом реализации сложного комплекса системы составление расписания, предназначенного для внедрения и использования в учебных заведениях.

Назначение системы – система должна удовлетворят требованиям учебных заведений по электронному составлению расписания, ссылаясь на законы в области образования. Способствует оптимизации рабочего времени составителя расписания, так как ИС сразу будет предупреждать о наложении пар у преподавателей либо групп, а также быстрому исправлению расписания, за счет нашей ИС.

* + 1. **Требования к программе или программному изделию.**
  1. **Требования к функциональным характеристикам.**
* ИС должна быть проста в использовании для рядового сотрудника;
* База данных выполняет свою главную функцию хранения информации расписания занятий;
* ИС должна стабильно работать;
* Серверная часть должна хранить базу данных пользователей, имеющих доступ к системе и обеспечивать аутентификацию пользователей согласно имеющихся записей;
* Составление расписания;
* Просмотр расписания;
* Выбор групп и недели для просмотра расписания;
* Печатать готовое расписание;
* Переход по ссылкам на курс в мудл;
* Автоматическая авторизация в МУДЛ, при предоставлении данной информации;
* Просмотр экзаменов;
* Составление экзаменов;
* Изменение профиля.
  1. **Требование к надежности**

Надёжность системы в целом зависит от надежности используемой операционной системы. Серверная часть должна обслуживать без сбоев одновременное подключение и работу до 1000 пользователей. Обе части должны без потерь передавать информацию по каналу связи между клиентом и сервером.

* 1. **Условия эксплуатации**

Стандартные условия эксплуатации программных продуктов. Необходимые сотрудники для обслуживания серверной части системы – системный администратор для обслуживания удаленного сервера, группа составителей расписания, а также группа лиц, которые будет заполнять данные о парах, преподавателях и остальной информации. Численность групп может манятся в зависимости от учебного заведения и его потребностям.

Для управления приложением учебного расписания СибГИУ пользователю достаточно уметь пользоваться компьютером на базовом уровне.

* 1. **Требования к составу и параметрам технических средств.**

Для нормальной работы приложения необходимо:

* Компьютер с процессором Intel i3 7100.
* Оперативная память не менее 4 Gb.
* Жесткий диск объемом не менее 128 Gb.
* Наличие адаптера подключения к сети (сетевой карты, модема и
* т.п.).
* Установленная ОС Windows 10 и выше.
  1. **Требования к информационной и программной совместимости.**

Модель системы должна работать под управлением ОС Windows 8, 8.1, 10, 11, поэтому требуется совместимость исполняемого модуля и библиотек динамического подключения стандартам, используемым этими ОС на платформе IBM PC. Модель должна использовать свой протокол передачи данных высокого уровня как надстройку над TCP/IP. Для хранения информации требуется использование баз данных формата SQL.

Для доступа к базе данных необходима сетевая карта и доступ в интернет.

В качестве средства проектирования структуры базы данных и создания файла базы данных требуется использовать MS SQL.

* 1. **Требования к маркировке и упаковке.**

Не предъявляются.

* 1. **Требования к транспортированию и хранению.**

Не предъявляются.

* 1. **Специальные требования.**

Не предъявляются.

1. **Требования к программной документации.**

Программной документацией к разрабатываемой информационной системы составление расписания является расчётно-пояснительная записка.

1. **Стадии и этапы разработки.**

Таблица №1 – Стадии и этапы разработки

|  |  |  |  |
| --- | --- | --- | --- |
| № | Содержание работы | Срок | Исполнитель этапы разработки |
| 1 | Разработка устава проекта | 7 дней | Токмаков Антон; |
| 2 | Исследование предметной области | 6 дней | Токмаков Антон;  Михайлов Максим; |
| 3 | Определение последовательности работ | 2 дня | Токмаков Антон;  Швецов Кирилл; |
| 4 | Составление диаграммы Ганта | 8 дней | Швецов Кирилл;  Токмаков Антон; |
| 5 | Составление технического задания | 14 дней | Токмаков Антон; |
| 6 | Выбор и обоснования методов реализации | 5 дней | Токмаков Антон; |
| 7 | Разработка базы данных | 14 дней | Михайлов Максимм |
| 8 | Разработка дизайна графического интерфейса пользователя | 14 дней | Швецов Кирилл; |
| 9 | Разработка основной логики информационной системы | 14 дней | Михайлов Максим; |
| 10 | Тестирование | 14 дней | Перунов Яраслав; |
| 11 | Отладка проекта | 5 дней | Токмаков Антон;  Михайлов Максим;  Швецов Кирилл;  Перунов Яраслав |
| 12 | Создание отчета и презентации | 1 день | Токмаков Антон;  Михайлов Максим;  Швецов Кирилл;  Перунов Яраслав |
| 13 | Демонстрация и защита проекта | 1 день | Токмаков Антон;  Михайлов Максим;  Швецов Кирилл;  Перунов Яраслав |

1. **Порядок контроля и приемки**

Испытание представленной модели и контроль качества ее работы провести на базе компьютерного класса кафедры. Во время испытаний проверить работу системы по следующим позициям:

* Проверка корректной записи информации о парах в БД;
* Корректное получение результатов их БД;
* Корректное составление расписание;
* Общее функционирование системы.

**Обоснование выбора программных и технических средств**

1. **Обоснование инструментов разработки**

Для разработки информационной системы составление расписания нам понадобятся средства редактирования кода для того, чтобы написать наш проект. Так как в специализированных IDE присутствует удобная подсветка кода, которая помогает при чтении и редактировании большого объема кода. IDE ускорит и упростит разработку нашего приложения в несколько раз.

Для проектирования интерфейса приложения понадобится специализированные редакторы, которые помогут облегчит работу дизайнера. Дизайнер может сначала графически построить прототип будущего приложения легко изменяя его без кода. Такое средство проектирование может уменьшить будущую работу и способствует понять на раннем этапе проекта, требование конечного пользователя к дизайну приложения.

Для хранения информации о парах, преподавателях и подобной информации, нужно использовать базу данных. Для удобного взаимодействия с БД, необходимо выбрать СУБД которая будет удовлетворять всем потребностям разработчика. СУБД поможет разработчику быстро и без труда взаимодействовать с большим количеством таблиц в БД не задумываясь о том, как их связывать и не поломать, так как за этим будет следить СУБД.

Так же для получения доступа к сайту обычному пользователю, необходимо иметь свой сервер или выложить его на хостинг в других компаниях. Так как свой сервер довольно сложно настраивать, а также у него могут быть частые свои и проблемы с безопасностью было принято решение использовать хостинг.

1. **Средства разработки интерфейса**

Для разработки интерфейса нашего приложения будет использована одна из нижеперечисленных программ:

– Figma;

– Adobe Photoshop;

– CoreIDRAW;

Figma – программа для дизайна. С помощью нее можно создавать не только прототипы, но и сами интерфейсы (сайты, приложения, панель управления). Весомым преимуществом данной программы является ее простота. Именно за счет этого Figma становится популярной в сфере дизайна.

Photoshop – многофункциональный графический редактор, разрабатываемый и распространяемый компанией Adobe Systems. В основном работает с растровыми изображениями, однако имеет некоторые векторные инструменты. Продукт является лидером рынка в области коммерческих средств редактирования растровых изображений и наиболее известной программой разработчика.

CoreIDRAW – графический редактор векторной графики, разработанный канадской корпорацией Corel. В пакет CorelDRAW также входит редактор растровой графики Corel. Photo-Paint и другие программы — например, для захвата изображений с экрана — Corel Capture. Программа векторизации растровой графики Corel Trace до 12 версии входила в пакет как самостоятельная программа.

Таблица №1 – выбор программного средства для разработки интерфейса.

|  |  |  |  |
| --- | --- | --- | --- |
| Средство разработки интерфейса | Доступность программного обеспечения | Наш опыт работы | Функциональность |
| Figma | + | + | + |
| Adobe Photoshop | - | - | + |
| CoreIDRAW | + | + | + |

Выбором нашей команды стало приложение Figma, так как у нас был достаточно большой опыт работы с данным приложением, так же на выбор программного средства повлияло его доступность и функциональность, которой достаточно, для нашей задумки.

1. **Средства редактирование кода**

Из средств для редактирования можно выделить 3 IDE:

– JetBrains Rider;

– Eclipse;

– Visual Studio.

JetBrains Rider – это кросс-платформенная .NET IDE, основанная на IntelliJ и ReSharper с помощью которой можно разрабатывать приложения .NET, ASP.NET, .NET Core, Xamarin и Unity в системах Windows, Linux и Mac.

Eclipse – это свободная интегрированная среда разработки модульных кроссплатформенных приложений. Развивается и поддерживается Eclipse Foundation. Наиболее известные приложения на основе Eclipse Platform - различные «Eclipse IDE» для разработки ПО на множестве языков включая C#.

Visual Studio – это интегрированная среда разработки, стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Помимо стандартного редактора и отладчика, которые есть в большинстве сред IDE, Visual Studio включает в себя компиляторы, средства авто завершения кода, графические конструкторы и многие другие функции для улучшения процесса разработки.

Таблица №2 – выбор программного средства для редактирования кода.

|  |  |  |  |
| --- | --- | --- | --- |
| Программа для выбора | Доступность программного обеспечения | Стабильность системы | Удобства использования |
| JetBrains Rider | - | - | + |
| Eclipse | + | - | - |
| Visual Studio | + | + | + |

При анализе интегрируемых сред разработки по важным нам критериям, представленным в таблице, мы выбрали Visual Studio Community, так как это доступное программное обеспечение, которое предоставляет богатый функционал для разработки как сайтов, так и desktop приложений. Так же Visual Studio является стабильной системой для разработки, если использовать Visual Studio Community 2019. Удобство использование является субъективным критерием, но для нашей команды Visual Studio является отличным IDE для разработки.

1. **Платформа для хостинга сайта**

При выборе хостинга были выделены три компании:

– Handyhost.ru;

– Fozzy.com;

– SmartApe.ru.

Handyhost.ru – российский хостинг, оказывающий услуги shared-хостинга, VDS и аренды серверов c 2009 года. Часть серверов находится в Москве и Санкт-Петербурге, остальные размещены в Германии. Техническая поддержка доступна круглосуточно по эл. почте, тикетам из панели управления, онлайн-чату и бесплатному телефону.

Fozzy.com – это американская компания с российскими корнями. Сервера хостера расположены во многих локациях по всему миру. Фози отличает очень производительный хостинг за умеренные деньги. Когда вы размещаете свой сайт у этого хостера, вы можете выбирать локацию сервера, что очень удобно для вебмастеров, работающих на западном рынке, но желающих иметь хорошую русскоязычную тех. поддержку и не сталкиваться с западными высокими ценами.

SmartApe.ru – это российская компания, которая предоставляет услуги хостинга по приемлемым ценам. Хостинг SmartApe работает с 2012 года, обеспечивая безлимитные услуги по размещению сайтов, круглосуточную поддержку клиентов и защиту от DDOS атак. Среди клиентов компании можно заметить крупные и серьезные организации.

Таблица №3 – выбор платформа для хостинга

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Платформа для хостинга сайта | Пробный период | Доступность в приобретении | Быстродействие | Стабильность | Поддержка |
| Handyhost.ru | + | + | + | + | + |
| Fozzy.com | - | + | + | +- | + |
| SmartApe.ru | - | - | +- | +- | + |

В качестве хостинга для нашего сайта будет выступать компания Handyhost.ru, так как она имеет пробный период для проверки работоспособности хостера в отличии от Fozzy.com и SmartApe.ru. По цене фаворитом является Handyhost.ru, так как у него наименьшая цена. Стабильность работы и быстродействие сервера – это очень важный критерий для хотера, здесь вперед вырывается так же Handyhost.ru, так как у Fozzy.com и SmartApe.ru стабильность немного хуже, но с быстродействием у Fozzy.com все в порядке, в отличии от SmartApe.ru. Поддержка у всех выделенных платформ для хостинга отличная.

1. **СУБД**

Мы будем использовать, есть ещё LiteSQL, PostgreSQL

MySQL – это система управления реляционными базами данных с открытым исходным кодом (RDBMS). Разработку и поддержку MySQL осуществляет корпорация Oracle. Чаще всего систему управления базами данных MySQL используют в следующих проектах: CMS (Content Management System). В системах управления контентом у MySQL (в связке с PHP/Apache) нет более сильного соперника. Веб-сайты. В базах данных сайтов MySQL помогает сохранять данные и отслеживать регистрацию пользователей. Корпоративные системы, ERP/CRM-приложения.

LiteSQL – это встраиваемая кроссплатформенная БД, которая поддерживает достаточно полный набор команд SQL и доступна в исходных кодах (на языке C). Исходные коды SQLite находятся в public domain, то есть никаких ограничений на использование нет. В 2005 году проект получил награду Google-O’Reilly Open Source Awards.

PostgreSQL – одна из самых популярных реляционных СУБД в мире. Непрерывной доработкой и выпуском новых версий PostgreSQL как опенсорсного проекта занимаются сотни разработчиков по всему миру. У СУБД нет владельца или управляющей компании, каждый пользователь может предложить изменения или включиться в процесс разработки.

Таблица №4 – выбор СУБД.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| СУБД | Простота освоения | Скорость работы | Наш опыт работы | Работа с сервером |
| MySQL | +- | + | + | + |
| LiteSQL | +- | + | + | - |
| PostgreSQL | +- | + | - | + |

При проведении анализа и сравнения СУБД было принято решение разрабатывать базу данных на MySQL, так как она является достаточно простой в освоении, у неё высокая скорость работы с большими данными, так же, как и у PostgreSQL, имеет возможность работы с сервером в отличии от LiteSQL, а также у нас есть опыт работы с MySQL, в отличии от PostgreSQL.

**Описание информационного обеспечения**

1. **Use-case**

На рисунке 1 показан use-case нашего приложения. Гость в нашей ИС сможет зарегистрироваться и посмотреть расписание, так же регистрация в системе включает в себя редактирование профиля.

Пользователь в нашей ИС сможет просматривать расписание студента, /преподавателя/свободных кабинетов, печатать расписание и просматривать предстоящие экзамены, так же студен может редактировать свой профиль.

Составитель расписание будет наследовать все функции от Студента, а также имеет свои, такие как Составление и редактирование расписание и экзаменов.

Так же в нашей системе присутствует Администратор, который будет наследовать все функции от Пользователя, а также будет иметь такие функции как удаления пользователя и изменение статуса пользователя, добавление пользователя.

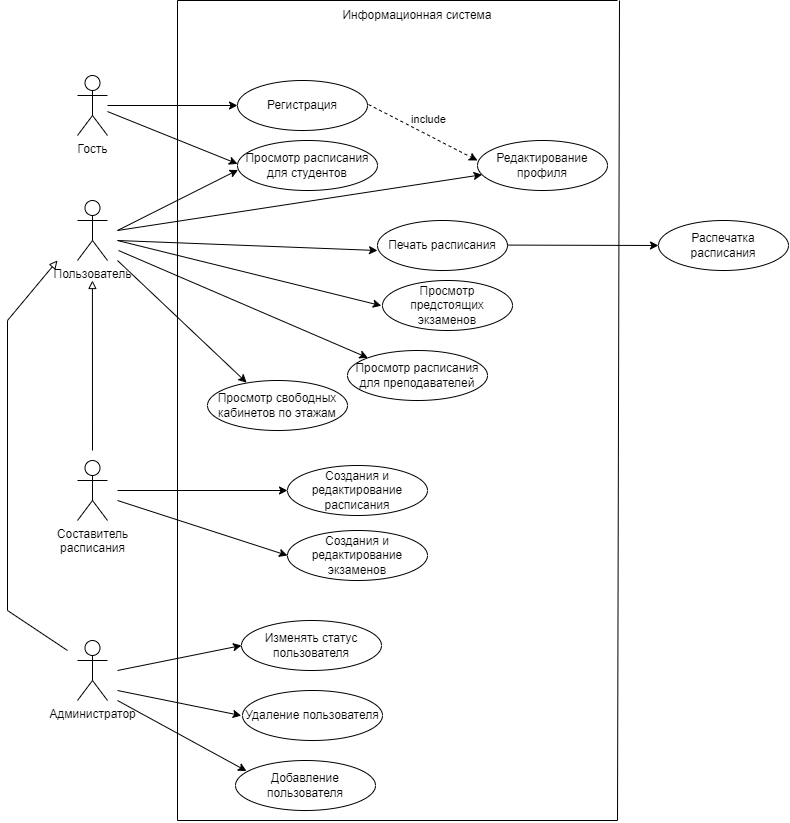


Рисунок 1 – use-case

1. **Функциональная диаграмма**

На нулевом уровне диаграммы в нашу информационную систему в управление входит Внутренние правила СибГИУ и Учебный план, которому надо придерживаться для составления расписания.

На вход у нас будет подаваться Учебный отдел, данные по дисциплинам, далее мы эти данные наша ИС будет обрабатывать, чтобы на выходе получить Расписание в удобной форме для студента, преподавателя и инженера кафедры.

В роли механизма у нас будут выступать Составители расписания, они будут вводить расписание в нашу ИС.

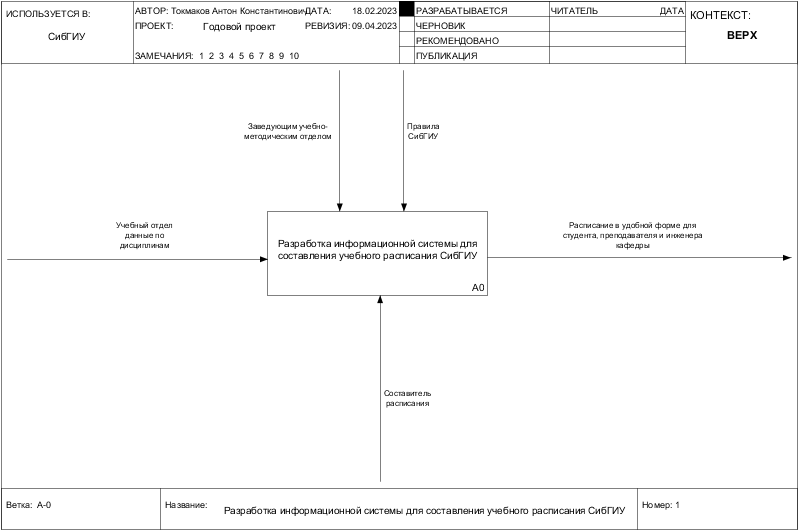


Рисунок 2 – первый уровень диаграммы A0

На рисунке 3 показана контекстная диаграмма первого уровня. На ней мы выделили 3 основных процесса: Ручной ввод расписания, Проверка ошибок ввода и Формирование удобного отражения.

Управление у нас приходит в блоки Ручной ввод и Формирование удобного отображения.

Вход у нас приходит в первый процесс Ручной ввод, далее первый процесс идет на вход в процесс Проверка ошибок ввода, на выходе же мы получаем либо подсказки к ручным корректировкам, то есть делаем управление над первым процессом, а после успешной проверки переход в следующий процесс, Формирование удобного отображения, где происходит поэтапное формирование удобного отображения для разных пользователей.

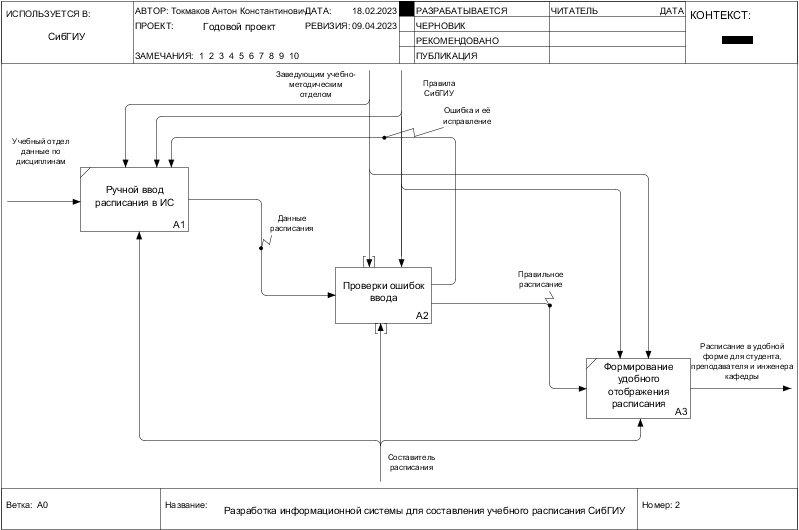


Рисунок 3 – Первый уровень контекстной диаграммы А1

На рисунке 3 мы декомпозируем процесс Проверок ошибок ввода, на котором мы выделили 3 процесса.

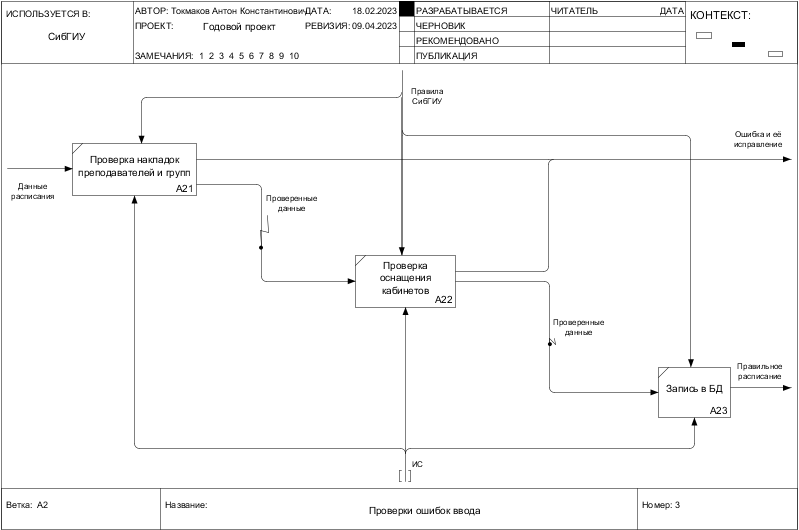


Рисунок 4 – Второй уровень декомпозиции A2

1. **Схема базы данных**

На рисунке 5 представлена схема базы данных, которая будет использоваться в проекте. Результирующей сущностью у нас будет Назначение на пару, данная сущность будет хранить всю подробную информацию о конкретной паре, такую как Группа преподавателей, Дисциплина, День недели (Понедельник/Вторник/Среда/Четверг/Пятница/Суббота), Номер пары (Первая/Вторая и тд), Кабинет, Группу, Тип дисциплины (Лекция/Практика).

Так же можно заметить большое количество справочников, таких как День недели, Номер пары, Неделя (Четная/Нечетная), Тип дисциплин (Лекция/Практика), Ссылка на курс в moodle, Тип дисциплины и Кабинет, это сделано для обеспечения защиты целостности, хранящейся в базе данных информации. Что бы снизить риск ошибки пользователей при работе с БД.

В схеме БД представлена сущность Преподаватель. Данная сущность служит для уникального определения преподавателя с его определенным качеством, у Преподавателя может быть Контактные данные, такие как номер телефона или почта, он может указать любое количество контактных данных, а также может не указывать их вовсе. Каждый преподаватель должен относиться к определенной кафедре для того, чтобы в дальнейшем можно было собрать разную аналитику по данным кафедр. Преподаватель может относиться только к одной кафедре, так как в нашей предметной области не важно к какой точно кафедре относиться преподаватель.

На каждое назначение на пару может указываться несколько преподавателей, это у нас регулирует сущность под название Преподаватель – Назначение на пару. Что бы показать, что преподаватели будут вместе вести пару, достаточно указать у них одинаковый номер группы.

Так же для уникального определения Группы студентов, у нас создана сущность с названием Группа студентов. Она определяет качества каждой группы, такие как Краткое наименование группы, Полное наименование группы, Количество студентов в группе и Специальность, последний атрибут будет подтягиваться из справочника Специальность, который в свою очередь будет зависеть от справочника Кафедра, для снижения рисков ошибок пользователей и сохранение информационной целостности БД.

Так как на Назначение на пару может быть несколько групп, в случае если это лекция, для этого случая создана связывающая сущность Группа – Назначение на пару. Она может по определенному коду группы хранить несколько групп, которые в дальнейшем и будут принадлежать назначению на пару.

Рассмотрим сущность Дисциплина, у неё имеются такие атрибуты как Кафедра, информация подтягивается из справочника, Наименование пары, Сокр. наименование пары, ссылка на курс в МУДЛ, количество практик и лекций в семестре, для отслеживания, когда должны закончиться пары этого предмета.

Так же есть сущность под названием Кафедра, она нужна для уникального определения кафедр с их отличительными особенностями, такие как Наименование и краткое наименование, Номер кабинета, Профессия и Контактные данные, которых может быть неограниченное количество.

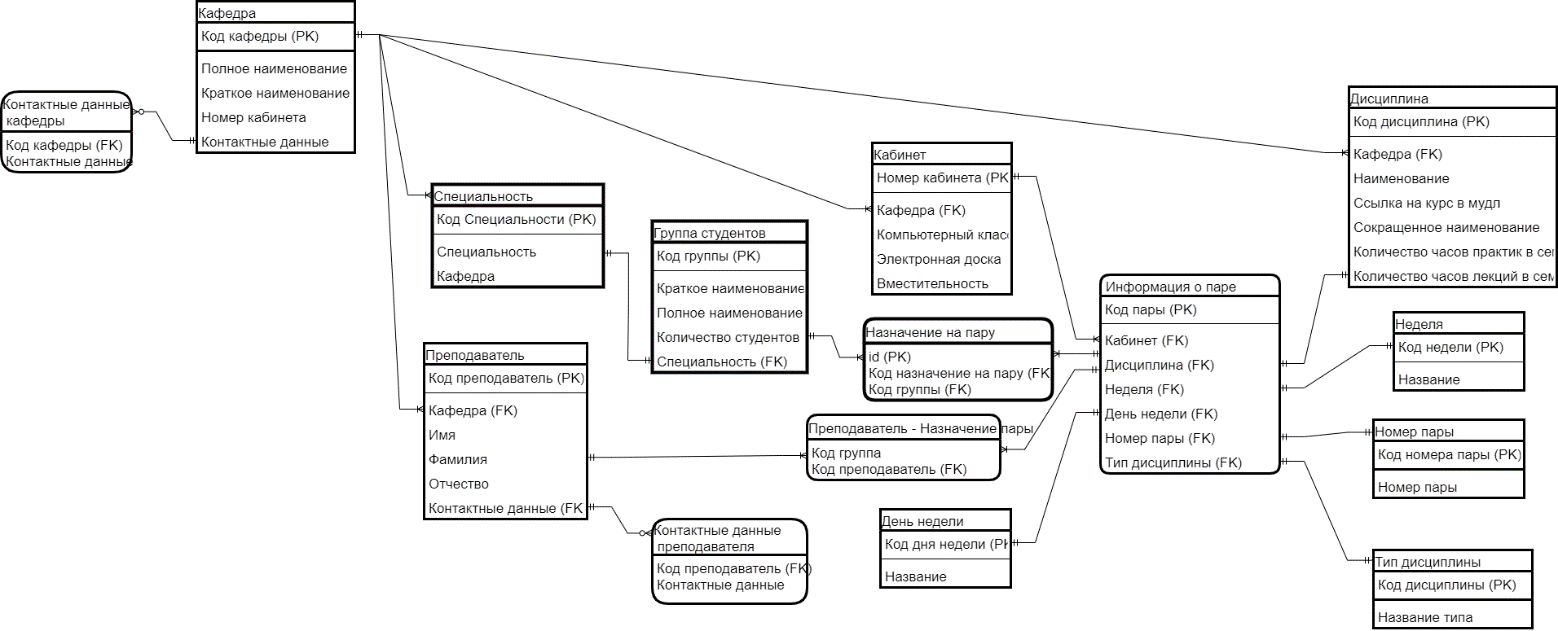


Рисунок 5 – схема базы данных

## **Описание алгоритмического обеспечения проекта**