Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный индустриальный университет»

Кафедра прикладных информационных технологий и программирования

**Описание информационного обеспечения проекта**

Выполнил: студент группы:

Токмаков Антон

Константинович

Проверил: преподаватель

Гасымов Р.Р.

Новокузнецк

2022

**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.

Студент в нашей ИС сможет просматривать свое расписание, 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.

Преподаватель так же будет наследован от Студента и иметь свою функцию Просмотр расписания для преподавателей.

Инженер кафедры, будет наследовать функции от Студента, но в добавок иметь свою, просмотр расписания свободных кабинетов на определенном этаже и в определенное время, а также просмотр расписание для преподавателей.

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

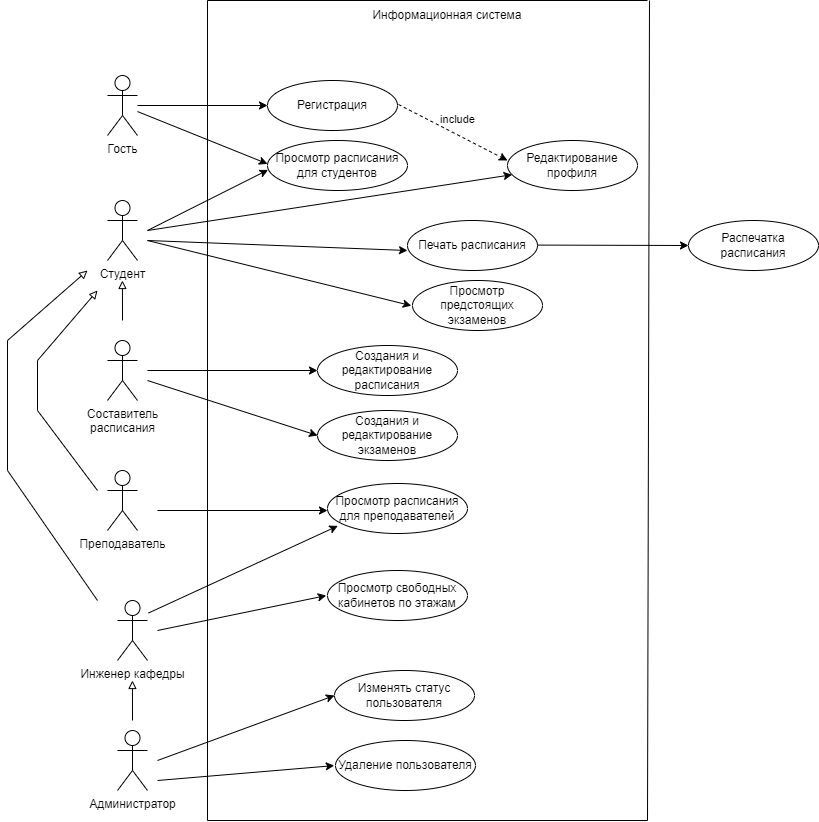


Рисунок 1 – use-case

**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.

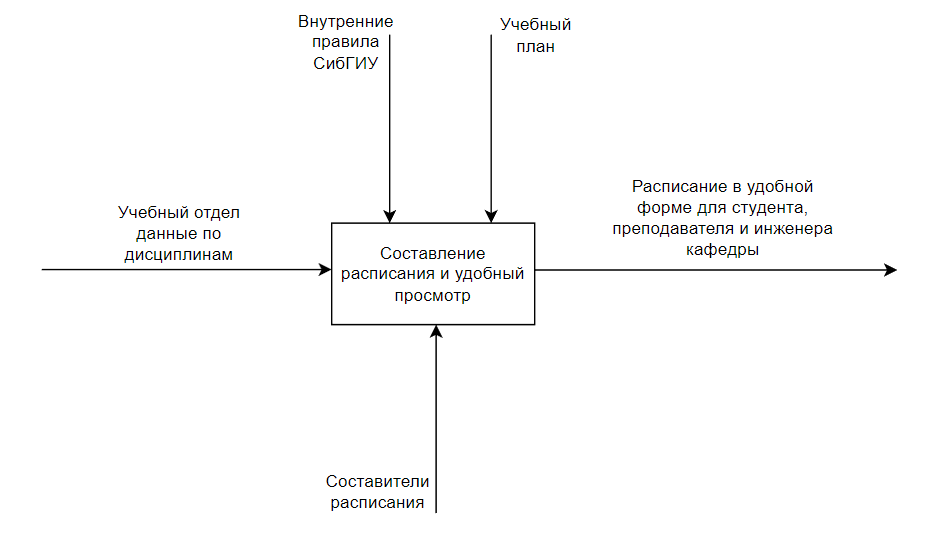


Рисунок 2 – первый уровень диаграммы A0

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.

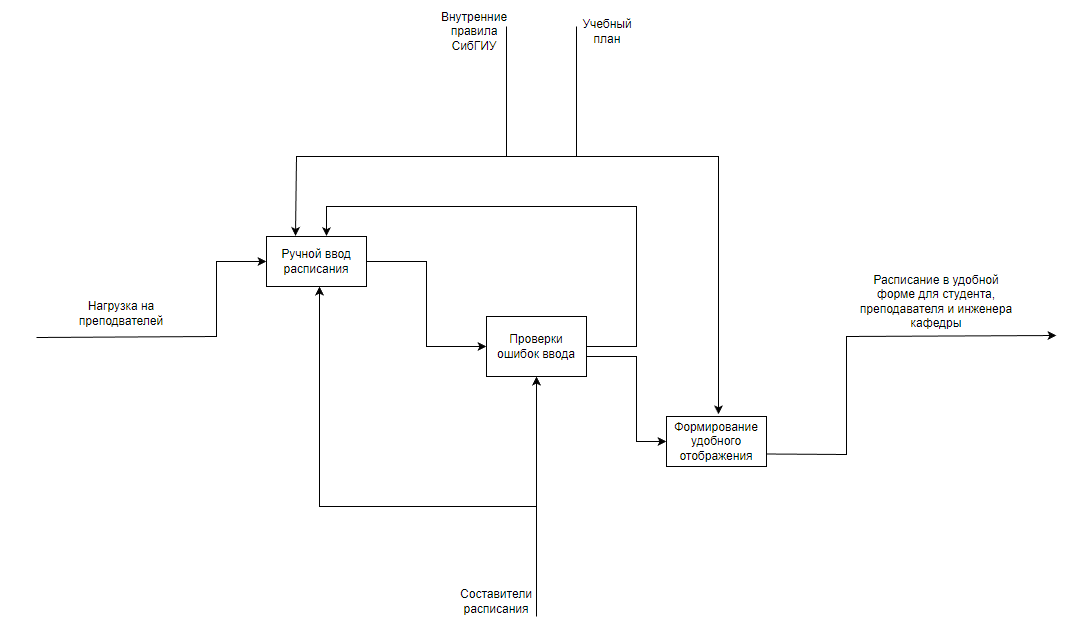


Рисунок 3 – Первый уровень контекстной диаграммы А1

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

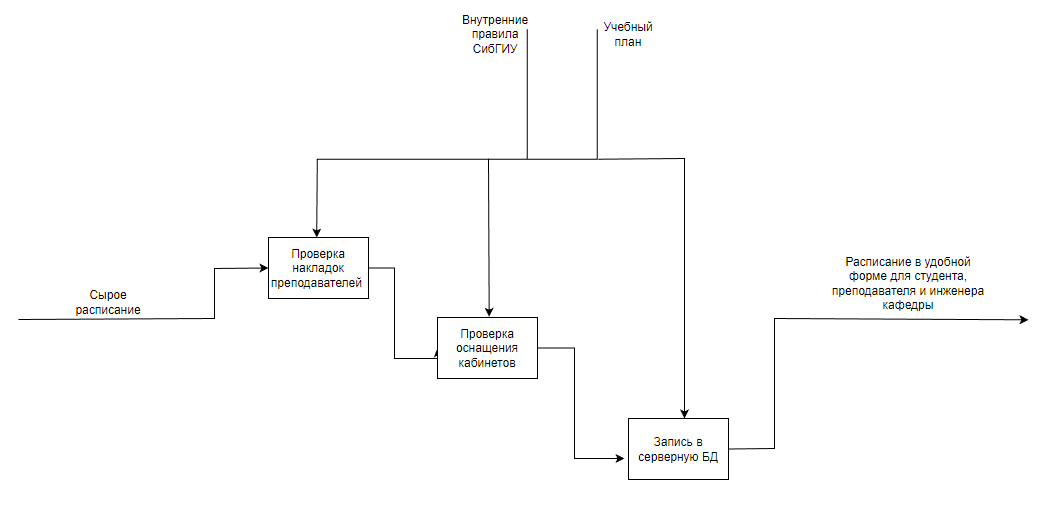


Рисунок 4 – Второй уровень декомпозиции A2

**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), Ссылку на курс в moodle.

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, Количество практик и количество 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.

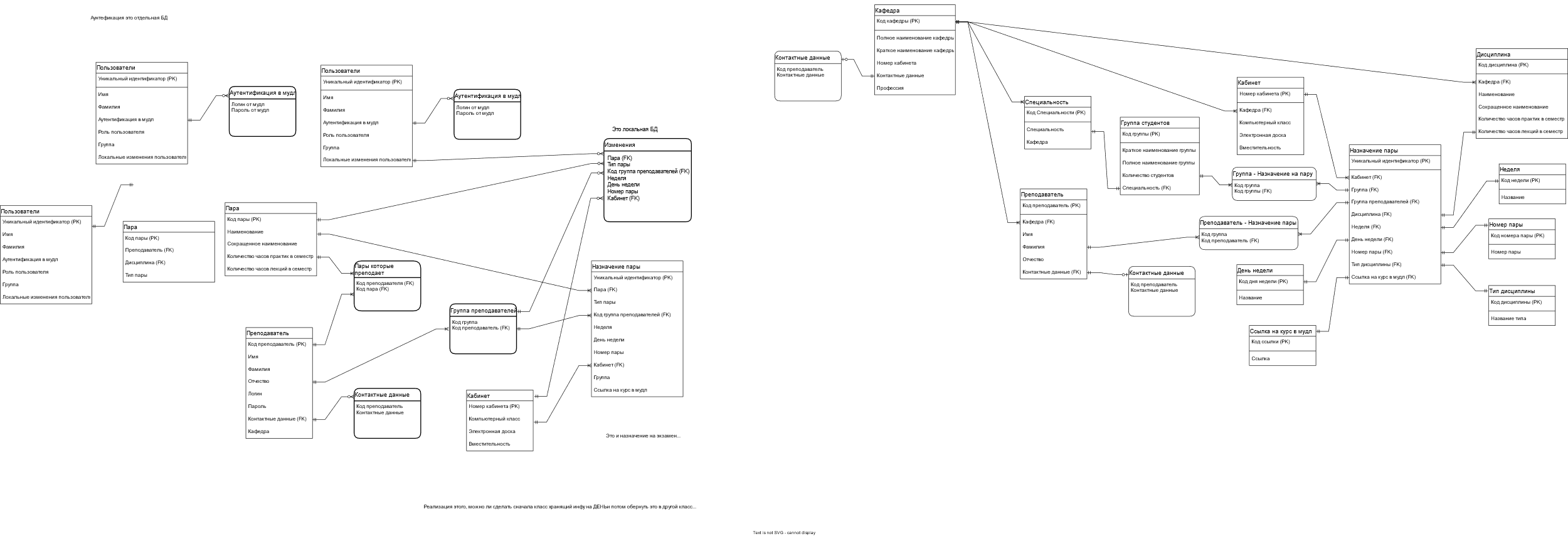


Рисунок 5 – схема базы данных