Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный** **исследовательский политехнический университет»**

Факультет: Прикладной математики и механики

Кафедра: Вычислительной математики, механики и биомеханики

Направление: 09.03.02 «Информационные системы и технологии»

Профиль бакалавриата: «Информационные системы и технологии»

**ПРАКТИЧЕСКАЯ РАБОТА**

по дисциплине

**«КОРПОРАТИВНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ»**

Тема: **«Модуль корпоративной информационной системы сети аптек (Диаграмма классов)»**

Выполнили:

студенты гр. ЦТУ-20-3Б

Пеленев Денис Вячеславович

Сыкулев Никита Сергеевич

Аввакумов Дмитрий Александрович

Принял:

ст. преподаватель, Банников Р.Ю.

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(должность, ФИО руководителя)*

*\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(оценка) (подпись)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(дата)*

**Пермь 2024**

Оглавление

[Постановка задачи 3](#_Toc159277634)

[Описание КИС и основного функционала 3](#_Toc159277635)

[Цель работы и задачи 3](#_Toc159277636)

[Диаграмма классов 4](#_Toc159277637)

[Средство разработки 4](#_Toc159277638)

[Описание диаграммы классов 5](#_Toc159277639)

[Выводы 8](#_Toc159277640)

# Постановка задачи

## Описание КИС и основного функционала

Наименование системы: «Модуль корпоративной информационной системы сети аптек».

Модуль корпоративной информационной системы сети аптек предназначен для обеспечения пользователям удобного доступа к информации по лекарственным препаратам в аптечной сети в режиме реального времени, размещения заказов, а также оптимизации управления аптеками внутри сети.

Внутри системы пользователи смогут:

1. Посматривать перечень лекарств в аптеках;
2. Совершать заказы в удобные для пользователей точки выдачи;
3. Просматривать историю заказов и их содержимое.

Администраторам системы, выступающим в роли, сотрудников аптеки будут доступны следующие функции:

1. Просмотр перечня товаров в своей аптеке;
2. Просмотр перечня заказов сделанных в аптеку с указанием основной информации по заказу описанной в ТЗ и содержимым заказа;
3. Функции по изменению статуса заказа;
4. Функции по просмотру, доступной, статистики описанной в ТЗ;
5. Функции по выгрузке отчетов из разделов, описанных в ТЗ.

Супер-пользователям системы, выступающим в роли, управляющих сети аптек, будут доступны следующие функции:

1. Добавление данных по разделам описанным в ТЗ;
2. Удаление данных по разделам описанным в ТЗ;
3. Просмотр статистики;
4. Выгрузка отчетов.

## Цель работы и задачи

Цель работы: разработать диаграмму классов для реализации модуля корпоративно-информационной системы сети аптек.

Задачи:

1. Выбрать средство реализации диаграммы классов
2. Определиться с необходимыми классами на основе ТЗ
3. Определится со связями классов и компонентов
4. Составить диаграмму классов
5. Написать описание к диаграмме

# Диаграмма классов

## Средство разработки

Диаграмма классов была разработана в онлайн-ресурсе drawio.

Draw.io - это бесплатный онлайн-сервис для создания диаграмм, схем, блок-схем, диаграмм потока данных, UML, ER и других типов диаграмм. Пользователи могут создавать диаграммы с помощью интуитивно понятного веб-интерфейса, не требуя установки на компьютер каких-либо программных компонентов.

Draw.io предлагает большой выбор форм, стрелок и других элементов, которые можно использовать при создании диаграммы. Он также позволяет загружать свои собственные изображения и использовать их в диаграммах. Для удобства пользователей сервис предлагает встроенную систему хранения диаграмм на Google Drive, Dropbox, OneDrive и GitHub.

Draw.io поддерживает импорт и экспорт диаграмм в различные форматы, такие как PDF, PNG, JPG, SVG, XML и др. Это делает его полезным инструментом для сотрудничества и обмена диаграммами с другими пользователями.

Общий поток работы в Draw.io может выглядеть следующим образом:

1. Создание новой диаграммы: Вы можете начать с создания новой диаграммы, выбрав тип диаграммы, который вам нужен.
2. Добавление элементов: После этого вы можете добавлять элементы на диаграмму, перетаскивая их из панели библиотеки в редактор.
3. Сохранение и экспорт: Когда ваша диаграмма готова, вы можете сохранить ее на вашем компьютере или в одном из облачных хранилищ, а также экспортировать в нужный вам формат.
4. Совместная работа: Draw.io также позволяет совместно работать над диаграммами, позволяя вам приглашать других пользователей для просмотра и редактирования диаграмм.

## Описание диаграммы классов

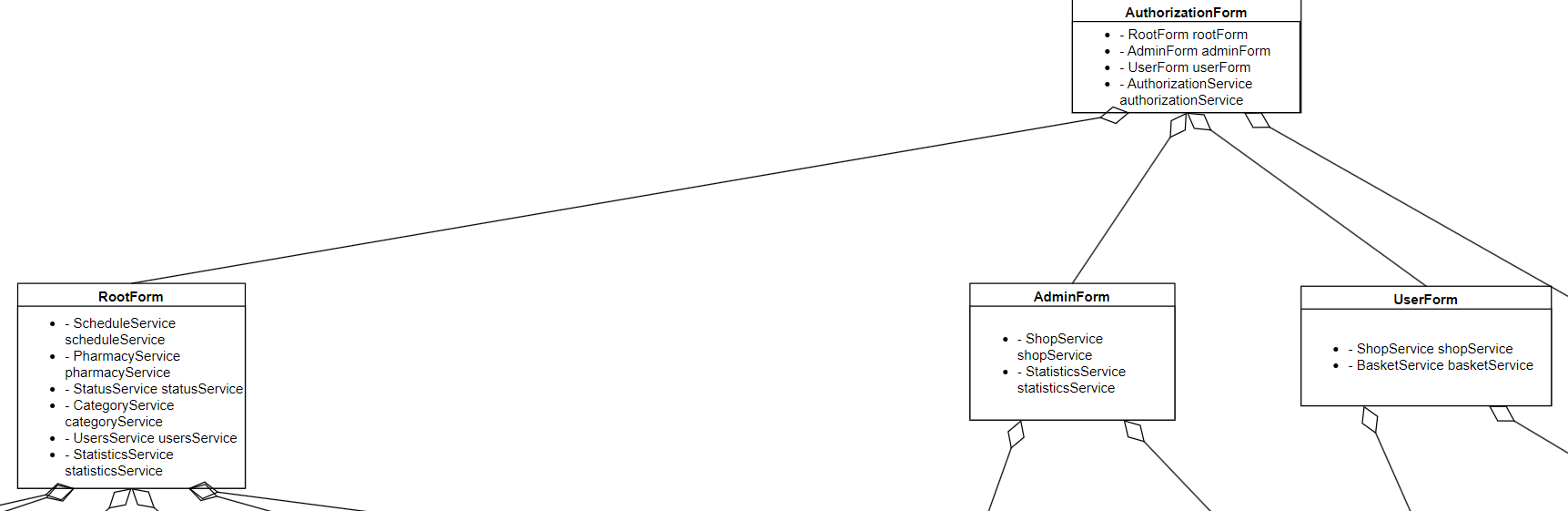


Рисунок 1 Диалграмма классов для AuthorizationForm

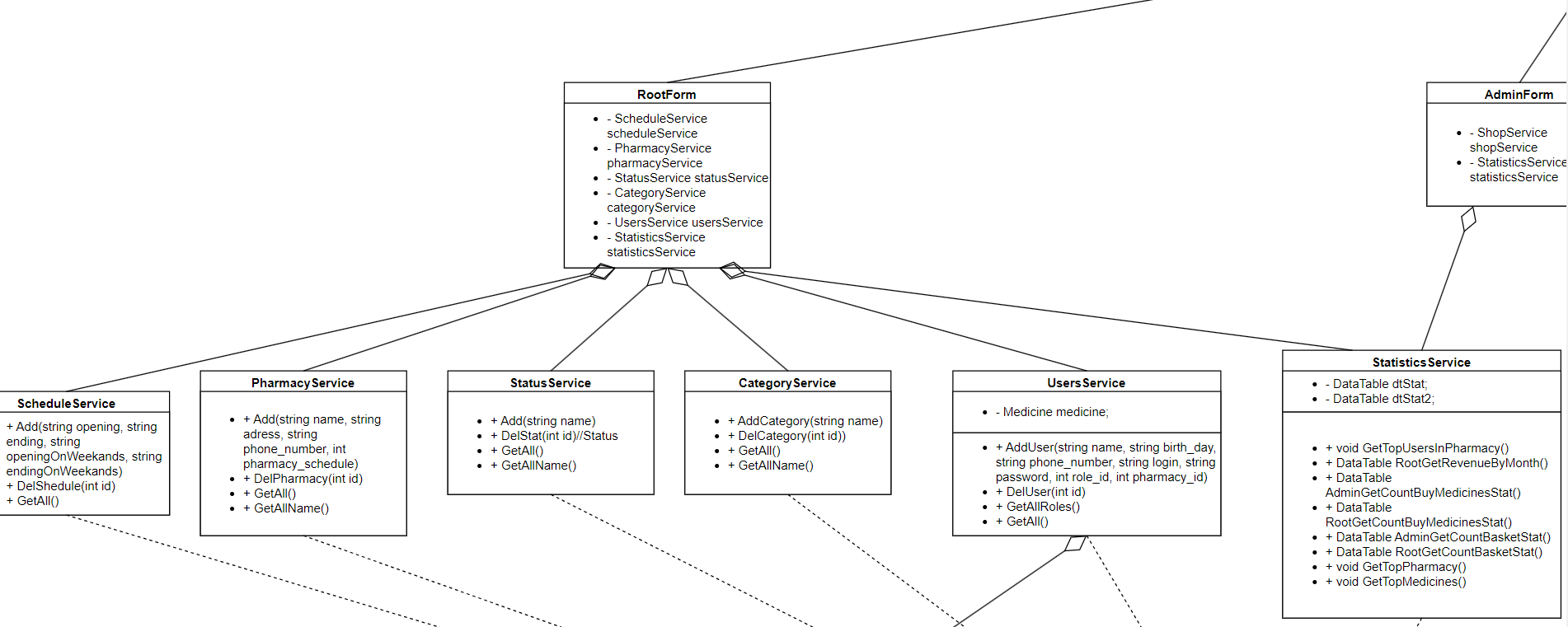


Рисунок 2 Диалграмма классов для RootForm

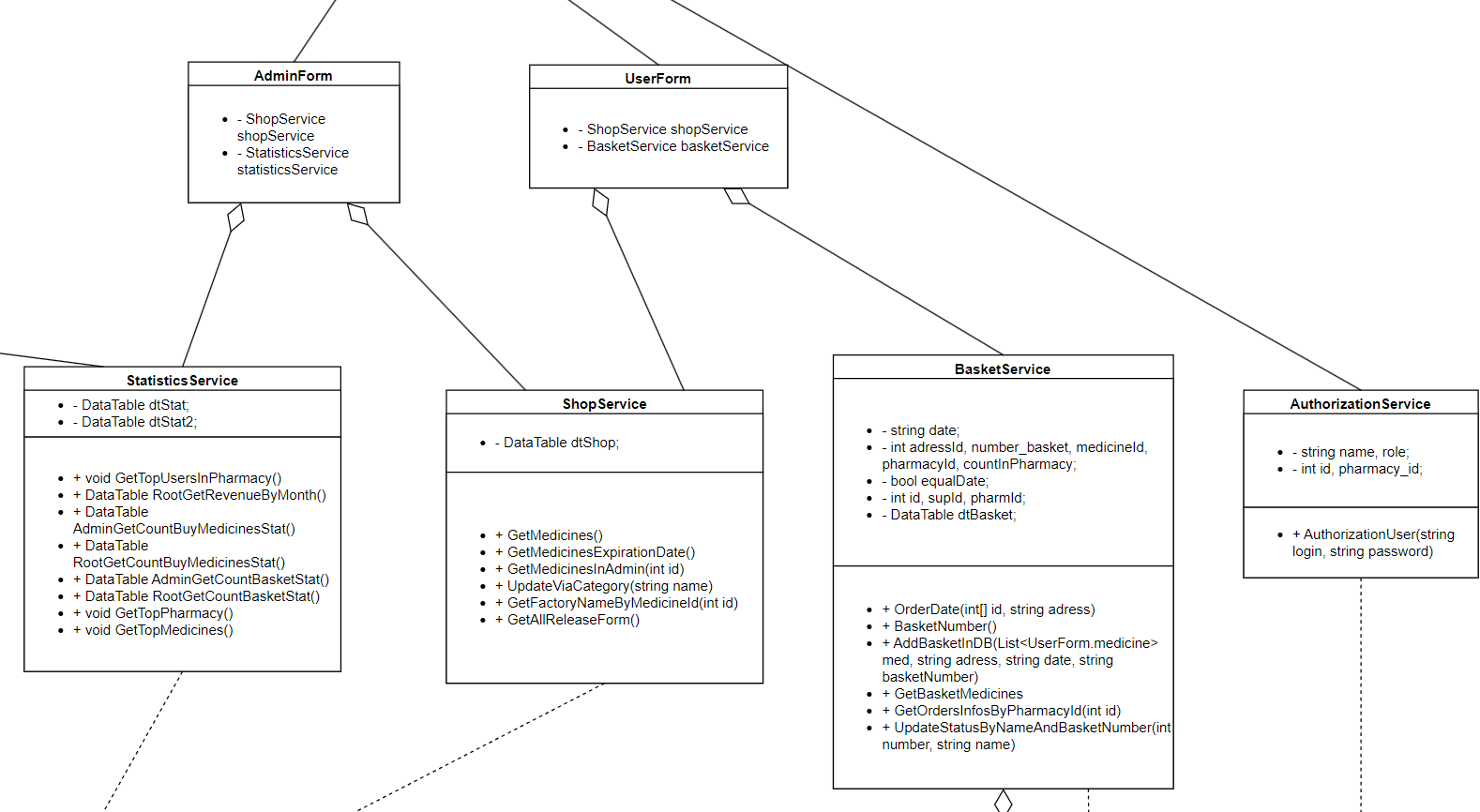


Рисунок 3 Диалграмма классов для UserForm и AdminForm

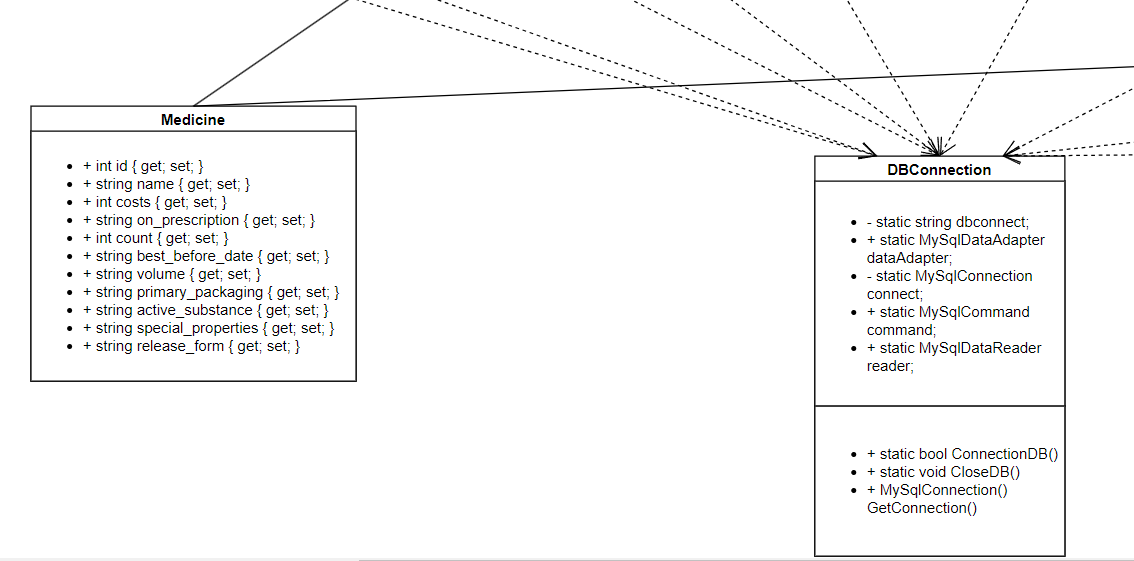


Рисунок Классы Medicine и DBConnection

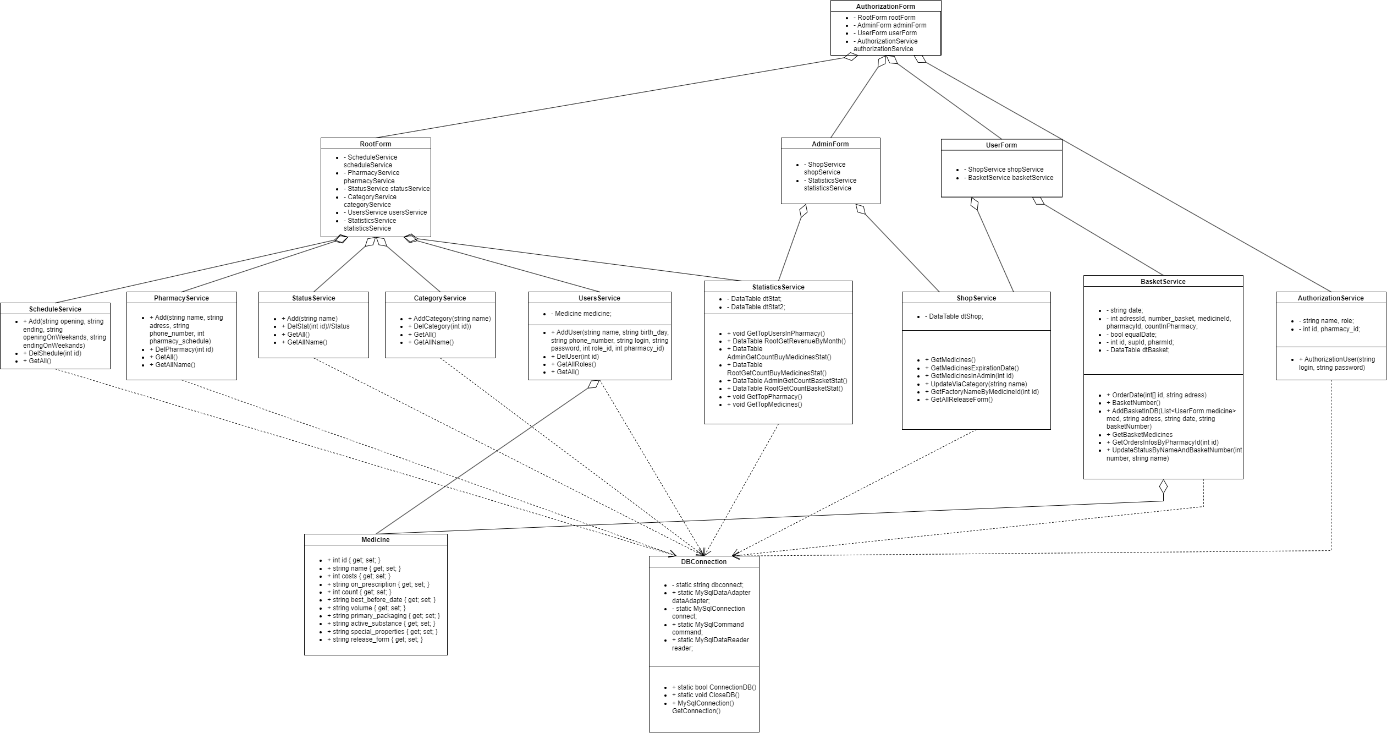


Рисунок 5 Общая диаграмма классов

На диаграмме классов отображены четыре формы, при помощи которых пользователи смогут взаимодействовать с системой и 12 классов.

Формы:

1. AuthorizationForm – форма регистрации. Открывается при запуске системы и отображается для всех системных ролей.
2. RootForm – форма для пользователя с системной ролью “Супер-пользователь”
3. AdminForm – форма для пользователя с системной ролью “Администратор”
4. UserForm – форма для пользователя с системной ролью “Пользователь”

Для каждой системной роли предусмотрена своя форма, следовательно, будучи “Администратором”, мы не сможем открыть формы других системных ролей.

Классы:

1. SheduleService – необходим для взаимодействия с графиками работы аптек. **Методы класса:**
   1. Add – добавление нового графика работы. Необходим для раздела “Добавить данные” у “Супер-пользователя”
   2. DelShedule - удаление графика работы. Необходим для раздела “Удалить данные” у “Супер-пользователя”
   3. GetAll – получить все графики работ. Необходим для раздела “Добавить данные” в пункте “Добавить аптеку” у “Супер-пользователя”
2. PharmacyService – необходим для взаимодействия с аптеками.

**Методы класса:**

* 1. Add – добавление новой аптеки. Необходим для раздела “Добавить данные” у “Супер-пользователя”
  2. DelPharmacy - удаление аптеки. Необходим для раздела “Удалить данные” у “Супер-пользователя”
  3. GetAll – получить список всех аптек с указанием id и прочих полей.
  4. GetAllName – получить список с названиями всех аптек

1. StatusService – необходим для взаимодействия со статусами заказов. **Методы класса:**
   1. Add – добавление нового статуса заказа. Необходим для раздела “Добавить данные” у “Супер-пользователя”
   2. DelStat - удаление статуса заказа. Необходим для раздела “Удалить данные” у “Супер-пользователя”
   3. GetAll – получить все возможные статусы заказов, включая поле id и name.
   4. GetAllName – получить названия всех возможных статусов заказов.
2. CategoryService – необходим для взаимодействия с категориями лекарств. **Методы класса:**
   1. Add – добавление новой категории лекарственных препаратов. Необходим для раздела “Добавить данные” у “Супер-пользователя”
   2. DelCategory- удаление категории лекарственных препаратов. Необходим для раздела “Удалить данные” у “Супер-пользователя”
   3. GetAll – получить список всех категорий лекарственных препаратов, включая поле id и прочие.
   4. GetAllName – получить список названий всех категорий лекарственных препаратов.
3. UsersService – необходим для взаимодействия с категориями лекарств. **Методы класса:**
   1. AddUser – добавление нового пользователя. Необходим для раздела “Добавить данные” у “Супер-пользователя”
   2. DelUser- удаление пользователя. Необходим для раздела “Удалить данные” у “Супер-пользователя”
   3. GetAllRoles– получить все возможные роли, включая поля name и id.
   4. GetAll– получить все возможные названия роли.

**Поля класса:**

1. Medicine medicine. Класс Medicine необходим для описания лекарства.
2. StatisticsService – необходим для получения статистической информации для раздела “Статистика” у системных ролей “Супер пользователь” и “Администратор”

**Методы класса:**

1. AdminGetCountBuyMedicinesStat() – Статистика по количеству купленного товара в магазине для системной роли “Администратор”
2. AdminGetCountBasketStat() – Статистика по количеству покупок в магазине для системной роли “Администратор”
3. getTopUsersInPharmacy() – Рейтинг покупателей для системной роли “Администратор”
4. RootGetCountBuyMedicinesStat() – Статистика по количеству проданной продукции для системной роли “Супер-пользователь”
5. RootGetCountBasketStat() – статистика по количеству заказов для системной роли “Супер-пользователь”
6. RootGetRevenueByMonth() – Статистика по доходам для системной роли “Супер-пользователь”
7. getTopPharmacy() – Рейтинг магазинов для системной роли “Супер-пользователь”
8. getTopMedicines() – Рейтинг лекарств для системной роли “Супер-пользователь”

**Поля класса:**

Поля ниже необходимы для хранения статистики по лекарствам в табличном виде, для удобного отображения на форме:

1. DataTable dtStat
2. DataTable dtStat2
3. ShopService – необходим для взаимодействия с лекарственными препаратами в аптеках.

**Методы класса:**

1. GetMedicines – получить список всех лекарств в аптеке.
2. GetMedicinesInAdmin - получить список всех лекарств в определённой аптеке под ролью администратора.
3. GetMedicinesExpirationDate – получить список сроков годностей лекарств.
4. UpdateViaCategory - обновляет перечень товаров при выборе категории
5. GetFactoryNameByMedicineId – получить наименование производителя по id лекарства.
6. GetAllReleaseForm – получить полный список всех форм выпуска лекарств.

**Поля класса:**

Поля ниже необходимы для хранения информации о лекарствах в табличном виде, для удобного отображения на форме:

1. - DataTable dtShop;
2. BasketService – необходим для взаимодействия с корзиной для покупателя.

**Методы класса:**

* 1. OrderDate – получить дату доставки заказа по id и адресу.
  2. BasketNumber – получить уникальный номер заказа.
  3. AddBasketInDB – добавление заказа в базу.
  4. GetBasketMedicines – получить товары в заказе по номеру заказа.
  5. GetOrdersInfosByPharmacyId – получить информацию по заказам по id аптеки.
  6. UpdateStatusByNameAndBasketNumber – обновить статус заказа по имени и уникальному номеру заказа.

**Поля класса:**

1. string date – дата заказа;
2. int adressId – id адреса аптеки,
3. int number\_basket – уникальный номер заказа
4. int medicineId – id лекарства
5. int pharmacyId – id производителя
6. int countInPharmacy – число лекарства в аптеке
7. bool equalDate – поле, чтобы определять статус, если текущая дата = дате доставки, то статус: "Готов к выдаче", если нет, то статус: "В пути".
8. int id – код заказа.
9. DataTable dtBasket - необходимо для хранения информации о корзинах с лекарствами в табличном виде, для удобного отображения на форме.
10. AuthorizationService – необходим для взаимодействия с корзиной для покупателя.

**Методы класса:**

* 1. AuthorizationUser – авторизация пользователя по логину и паролю.

**Поля класса:**

1. string name – имя пользователя.
2. string role – роль пользователя.
3. int id – код пользователя.
4. int pharmacy\_id – код аптеки, в которой он работет, null, если клиент.
5. DBConnection – класс конфигурации, необходимый для настройки подключения к базе.

**Методы класса:**

* 1. static bool ConnectionDB – проверяет удалось ли установить подключение.
  2. static void CloseDB() – закрыть подключение к базе
  3. MySqlConnection() – создание экземпляра адаптера MySQL для подключения к базе.
  4. GetConnection() –подключиться к базе.

**Поля класса:**

1. static string dbconnect – url для подключения к базе.

**Набор объектов для взаимодействия с базой:**

1. static MySqlDataAdapter dataAdapter - этот класс предоставляет мост между набором данных (DataSet) и источником данных MySQL.
2. static MySqlConnection connect - класс представляет соединение с базой данных MySQL и предоставляет методы для выполнения запросов к базе данных, открытия и закрытия соединения, управления транзакциями и т.д.
3. static MySqlCommand command – исполняемая SQL команда.
4. static MySqlDataReader reader – класс, использующийся при извлечении выборки данных из результата запроса.
5. Medicine – класс Medicine необходим для описания лекарства.

**Поля класса:**

* 1. int id – код лекарства
  2. string name – название лекарства
  3. int costs – стоимость
  4. string on\_prescription – нужно ли назначение врача
  5. int count - количество
  6. string best\_before\_date – срок годности
  7. string volume - объём
  8. string primary\_packaging - первичная упаковка
  9. string active\_substance – действующее вещество
  10. string special\_properties – особые указания
  11. string release\_form – форма выпуска

# Выводы

В ходе работы была создана диаграмма классов для системы сети аптек. Описаны методы и поля классов с их назначением. Согласно ТЗ следующим действием будет реализация программной части.