МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева (Самарский университет)»

Институт информатики и кибернетики

Кафедра программных систем

**ОТЧЕТ**  
 по лабораторному практикуму по дисциплине «Современные

методы разработки и проектирования программных комплексов»

по теме «Мобильное приложение для поиска лекарств из

открытых источников и напоминания об их приёме»

Обучающийся группы 6132-020402D К.А. Портнов

Обучающийся группы 6132-020402D В.И. Хорина

Руководитель Л.С. Зеленко

Самара 2022

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева (Самарский университет)»

Институт информатики и кибернетики

Кафедра программных систем

ЗАДАНИЕ

на лабораторный практикум по дисциплине

«Современные методы разработки и проектирования   
программных комплексов»

обучающимся в группе № 6132-020402D

К.А. Портнову

В.И. Хориной

1. Тема проекта:«Мобильное приложение для поиска лекарств из открытых источников и напоминания об их приёме»
2. Исходные данные к проекту**:** см. приложение к заданию
3. Перечень вопросов, подлежащих разработке:
   1. Произвести анализ предметной области: изучить основные принципы реализации лекарственных препаратов, изучить особенности их приема
   2. Выполнить обзор существующих систем-аналогов
   3. Разработать информационно-логический проект системы
   4. Разработать и реализовать программное и информационное обеспечение, провести его тестирование и отладку
   5. Оформить документацию курсового проекта
   6. Подготовить презентацию по разработанной системе
4. Перечень графических разработок:
   1. Структурная схема системы
   2. Диаграмма вариантов использования и диаграмма классов
   3. Схемы основных алгоритмов
5. Календарный план выполнения работ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Содержание работы по этапам | Объем этапа в % к общему объему проекта | Срок  окончания | Фактическое выполнение |
| 1 | Оформление технического задания и его утверждение | 5 | 04.10.2022 |  |
| 2 | Описание и анализ предметной области | 10 | 04.10.2022 |  |
| 3 | Проектирование системы | 30 | 15.11.2022 |  |
| 3.1 | Разработка структурной схемы системы | 5 | 18.10.2022 |  |
| 3.2 | Разработка прототипа интерфейса пользователя | 10 | 25.10.2022 |  |
| 3.3 | Разработка информационно-логического проекта системы и его предъявление руководителю | 15 | 15.11.2022 |  |
| 4 | Реализация проекта, разработка контрольных примеров. Предъявление реализации руководителю | 45 | 29.11.2022 |  |
| 5 | Корректировка проекта и оформление документации проекта. Защита проекта с представлением презентации. | 10 | 27.12.2022 |  |

Задание принял  
 к исполнению 14.09.2022 К.А. Портнов

14.09.2022 В.И. Хорина

ПРИЛОЖЕНИЕ  
к заданию на лабораторный практикум  
обучающимся в группе № 6132-020402D  
К.А. Портнову

В.И. Хориной

Тема проекта: «Мобильное приложение для поиска лекарств из открытых источников и напоминания об их приёме»

Исходные данные к проекту:

1. Характеристика объекта автоматизации:

## объект автоматизации: аптечная сеть;

## виды автоматизируемой деятельности:

* + процесс авторизации/регистрации пользователя;
  + процесс поиска лекарственных препаратов по название, составу, стране производителю, наличию сертификата;
  + процесс составление расписания приема лекарств;
  + процесс просмотра запланированных приемов лекарств;
  + процесс получения оповещений;

## длина номера мобильного телефона – 11 символов;

## длина кода подтверждения по СМС– 4 символа;

## количество критериев для поиска – 4;

## количество элементов для поиска по составу – 1;

1. количество элементов для поиска по стране производителю – 1;
2. количество элементов для поиска по названию – 1;
3. ссылок для поиска препарата в аптечных сетях – 2;
4. типов подписок на оповещения – 2;
5. типов периодичности приёма лекарств – 2;
6. начало доступного периода просмотра напоминаний в календаре – начало месяца;
7. максимальное количество напоминаний одного препарата в день – 5 раз.
8. Требования к информационному обеспечению:
9. информационное обеспечение разрабатывается на основе следующего источника:
   * Создание мобильных приложений [Электронный ресурс]. URL: <https://learn.microsoft.com/en-us/dotnet/maui/what-is-maui> (дата обращения: 14.09.2022);
10. структура базы данных разрабатывается на основании следующих сведений:
    * о пользователе (телефон, код, e-mail, имя, список лекарств, список напоминаний);
    * о лекарстве (название, сертификат, дозировка, форма выпуска, состав, страна производитель, изображение);
    * напоминания (лекарство, дата, время, дозировка, время до приема);
    * сертификат (номер, дата регистрации, статус);
    * страна производитель (название, изображение флага)
11. в системе имеются справочники:

* лекарственные препараты;
* страны производители;

1. должна быть обеспечена целостность базы данных и защита от несанкционированного доступа.
2. Требования к техническому обеспечению:
   1. Требования к техническому обеспечению серверной части системы:
3. тип ЭВМ – IBM PC совместимый;
4. объем ОЗУ – не менее 2 Гб;
5. объем свободного пространства на внешнем диске – не менее 50 Гб;
6. наличие подключения к сети Интернет;
7. технические характеристики определяются в процессе выполнения проекта.
   1. Требования к техническому обеспечению клиентской части системы:
8. тип ЭВМ – мобильное устройство Android/IOS;
9. монитор с разрешающей способностью не ниже 1080 х 2340;
10. манипулятор – сенсорный экран;
11. наличие подключения к сети Интернет;
12. технические характеристики определяются в процессе выполнения проекта.
13. Требования к программному обеспечению:
    1. Требования к программному обеспечению серверной части системы:
14. тип операционной системы – Windows 7 и выше/Linux;
15. .NET Runtime 7.0
16. СУБД – SQL Server 15.0.
    1. Требования к программному обеспечению клиентской части системы:
17. тип операционной системы – Android 12 и выше – API 21 и выше / IOS 14.2 и выше;
    1. Требования к программному обеспечению рабочего места разработчика:
18. тип операционной системы – Windows 7 и выше, MacOS 12;
19. язык программирования – С#;
20. среда программирования – Visual Studio 2022;
21. .NET SDK 7.0;
22. .NET MAUI SDK;
23. СУБД – SQL Server 15.0;
24. среда проектирования – StarUML 5.0.2.
25. Общие требования к проектируемой системе:

5.1 Функции, реализуемые системой:

1. функции системы:
   * аутентификация пользователя в системе;
   * регистрация пользователя в системе;
   * рассылка напоминаний о приёме лекарств;
   * ускоренная авторизация;
   * отслеживание ранее отправленных уведомлений;
   * фильтрация лекарств по параметрам (название, состав, страна производитель, наличие сертификата);
   * автоматическое составление уведомлений для последующей отправки;
   * визуализация процессов работы с приложением;
   * проверка дублирования лекарств;
   * выдача справочной информации о системе;
   * выдача справочной информации о лекарственном препарате;
   * генерирование ссылок для поиска препарата в аптечных сетях;
2. функции пользователя:
   * регистрация в приложении (ввод телефона);
   * авторизация в приложении (ввод кода, отправленного по SMS);
   * возможность при повторной авторизации использовать touchId/faceId;
   * возможность по ссылке перейти в интернет магазин аптечных сетей;
   * возможность изменения своих персональных данных в личном кабинете (имя, e-mail);
   * возможность удалить свой аккаунт;
   * составление фильтров для поиска лекарств;
   * просмотр запланированных приёмов;
   * отслеживание плана приёма лекарства;
   * подтверждение приёма лекарства;
   * добавление лекарств в свою библиотеку;
   * составление графика приёма лекарств;
   * включение/отключение оповещений системы;
   * просмотр справочной информации.

5.2 Технические требования к системе:

1. режим работы ‑ диалоговый;
2. максимальная задержка получения уведомления – 5 мин;
3. максимальная задержка получения SMS – 5мин;
4. максимальное время загрузки данных на странице – 1с;
5. система должна удовлетворять санитарным правилам и нормам  
    СанПин 2.2.2./2.4.2198-07;
6. условия работы средств вычислительной техники (содержание вредных веществ, пыли и подвижность воздуха) должны соответствовать ГОСТ 12.1.005, 12.01.007;
7. температура окружающего воздуха – 15-35°С;
8. влажность воздуха – 45-75%.

Руководитель   
проекта Л.С. Зеленко

Задание принял  
к исполнению 14.09.2022 К.А. Портнов

14.09.2022 В.И. Хорина

РЕФЕРАТ

Пояснительная записка 75 с, 19 рисунков, 4 таблиц, 13 источников,  
2 приложения.

Графическая часть: ??? слайдов презентации PowerPoint.

Ключевые слова:

ЛЕКАРСТВЕННЫЕ ПРЕПАРАТЫ, НАПОМИНАНИЯ О ПРИЁМЕ, PUSH УВЕДОМЛЕНИЯ, СЛОВАРЬ ПРЕПАРАТОВ, СОСТАВЛЕНИЕ ГРАФИКА, ТАБЛЕТНИЦА, МОБИЛЬНОЕ ПРИЛОЖЕНИЕ.

Объектом автоматизации является аптечная сеть.

Во время курсового проектирования разработаны алгоритмы и соответствующая им мобильное приложение, позволяющее составлять графика приёма лекарственных препаратов. Лекарственные препараты хранятся в базе данных и доступ к ним осуществляется через серверную часть приложения, при этом ограничений на количество препаратов не существует. Пользователь через интерфейс приложения добавляет лекарственные препараты в свою библиотеку и задаёт периодичность и время приёма. Специальный алгоритм на серверной части приложения периодически проверяет срабатывание таймеров на приём лекарств и отправляет соответствующее сообщение пользователю через сервисы Google/Apple. Пользователь имеет возможность посмотреть детальное описание лекарственного препарата и перейти по ссылке на сайты аптечных сетей.

Клиентская часть приложения написана на языке С# в среде Visual Studio 2022 и функционирует под управлением операционных систем Android 12 и выше – API 21 и выше / IOS 14.2 и выше.

Серверная часть приложения написана на языке C# в среде Visual Studio 2022 и функционирует под управлением операционных систем Windows 7 и выше/Linux.

СОДЕРЖАНИЕ

[Введение 13](#_Toc114482670)

[1 Описание и анализ предметной области 14](#_Toc114482671)

[1.1 Описание предметной области 14](#_Toc114482672)

[1.2 Описание систем-аналогов 15](#_Toc114482673)

[1.2.1 Название системы-аналога 1 15](#_Toc114482674)

[1.2.2 Название системы-аналога 2 16](#_Toc114482675)

[1.3 Постановка задачи 17](#_Toc114482676)

[2 Проектирование системы 19](#_Toc114482677)

[2.1 Выбор и обоснование архитектуры системы 19](#_Toc114482678)

[2.2 Структурная схема системы 19](#_Toc114482679)

[2.3 Разработка прототипа интерфейса пользователя системы 23](#_Toc114482680)

[2.4 Разработка информационно-логического проекта системы 25](#_Toc114482681)

[2.4.1 Язык UML 25](#_Toc114482682)

[2.4.2 Диаграмма вариантов использования 25](#_Toc114482683)

[2.4.3 Диаграмма классов 25](#_Toc114482684)

[2.5 Логическая модель данных 27](#_Toc114482685)

[2.6 Выбор и обоснование алгоритмов обработки данных 28](#_Toc114482686)

[2.7 Выбор и обоснование комплекса программных средств 29](#_Toc114482687)

[2.7.1 Выбор операционной системы 29](#_Toc114482688)

[2.7.2 Выбор языка программирования 30](#_Toc114482689)

[2.7.3 Выбор среды разработки 30](#_Toc114482690)

[2.7.4 Выбор системы управления базами данных 30](#_Toc114482691)

[3 Реализация системы 31](#_Toc114482692)

[3.1 Разработка и описание интерфейса пользователя 31](#_Toc114482693)

[3.2 Диаграммы реализации 31](#_Toc114482694)

[3.2.1 Диаграмма развертывания 31](#_Toc114482695)

[3.2.2 Диаграмма классов 32](#_Toc114482696)

[3.3 Физическая модель данных 32](#_Toc114482697)

[3.4 Выбор и обоснование комплекса технических средств 33](#_Toc114482698)

[3.4.1 Расчет объема занимаемой памяти 33](#_Toc114482699)

[3.4.2 Минимальные требования, предъявляемые к системе 35](#_Toc114482700)

[Заключение 36](#_Toc114482701)

[Список использованных источников 37](#_Toc114482702)

[Приложение А Руководство пользователя 40](#_Toc114482703)

[А.1 Назначение системы 40](#_Toc114482704)

[А.2 Условия работы системы 40](#_Toc114482705)

[А.3 Установка системы 40](#_Toc114482706)

[А.4 Работа с системой 40](#_Toc114482707)

[А.4.1 Работа с системой в режиме администратора (если необходимо) 41](#_Toc114482708)

[А.4.2 Работа с системой в режиме пользователя 41](#_Toc114482709)

[Приложение Б Листинг модулей программы 42](#_Toc114482710)

ВВЕДЕНИЕ

В последние годы стала прослеживаться тенденция увеличения средней продолжительности жизни человека. Это связано с множеством факторов таких как: люди стали лучше следить за своим здоровьем, лечить различные болезни на ранних стадиях, когда лечение самое эффективное, улучшилось качество питания, качество жилищных условий, люди стали внимательнее относиться к уровню личной гигиены и безусловно один из главных факторов – стремительное развитие современной медицины – с каждым годом на прилавках магазинов и в медицинских учреждениях появляется всё больше и больше различных лекарственных препаратов способных противостоять почти любой известной на данный момент науке болезни или инфекции.

К примеру группа препаратов антибиотиков, вошедших в жизнь человечества более полувека назад спасли миллионы жизней, поборов такие болезни, как пневмония, туберкулёз, гангрена и прочие инфекционные заболевание, без лечения приводящие к летальному исходу.

Но в нашем мире всегда должно быть равновесие и, если мы учимся бороться с каким-либо болезнями, они, в свою очередь, начинают эволюционировать и приобретать иммунитет к разработанному лечению, из-за чего необходимо постоянно совершенствовать лекарственные препараты. Что в свою очередь является одной из причин почти бесчисленного множества все возможных лекарственных препаратов.

Всё чаще по назначению врача приходится принимать сразу одновременно несколько различных лекарств с разной периодичностью, дозировкой и продолжительностью приёма. Чтобы ничего не перепутать приходиться вести целое расписание для приёма лекарств и постоянно сверяться с этим списком, чтобы ничего не забыть и не пропустить приём, что по своей сути является очень рутинной и монотонной задачей и далеко не все могут успешно с ней справиться, но на помощь человечеству в решении этой проблемы приходят современные технологий с мобильными приложениями для телефонов, которые есть у каждого человека всегда под рукой и с системой уведомлений (push notification), позволяющей оперативно отправить сообщение на мобильное устройство.

Во время выполнения лабораторного практикума необходимо разработать мобильное приложение с функциями напоминания о необходимости приёма лекарств, с помощью которой можно создавать свой список лекарств и настраивать время, дозировку и периодичность приёма лекарств.

При проектировании системы будут использоваться методологии структурного проектирования систем и ООАП (Object-Oriented Analysis/Design), в основу которой положена объектно-ориентированная методология представления предметной области в виде объектов, а также язык моделирования UML (Unified Modeling Language), который является стандартным инструментом для разработки «чертежей» программного обеспечения [1].

1. Описание и анализ предметной области
   1. Описание предметной области

Анализ предметной области, позволяет выделить ее сущности, определить первоначальные требования к функциональности и определить границы проекта. Модель предметной области должна быть документирована, храниться и поддерживаться в актуальном состоянии до этапа реализации.

Точкой входа для пользователя является авторизация в приложении, после чего автоматически создайся его лично кабинет и ему предоставляются возможности по созданию своей собственно библиотеке лекарств и созданию напоминаний о приёме с указанием периодичности, временем и дозировкой. Вся пользовательская информация хранится на сервере в безе данных.

Исходя из описанного выше диаграмма объектов данной предметной области изображена на рисунке 1 и должна включать в себя сами объекты моделирования такие как библиотека лекарств, список пользователей.

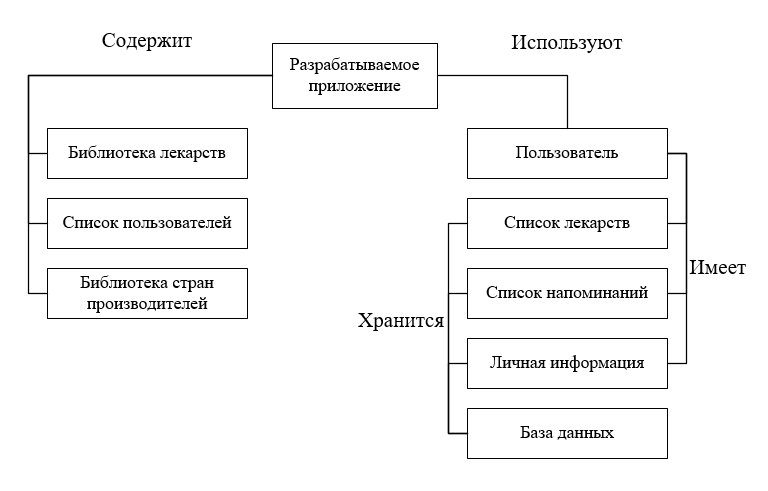


Рисунок 1 – Диаграмма объектов предметной области

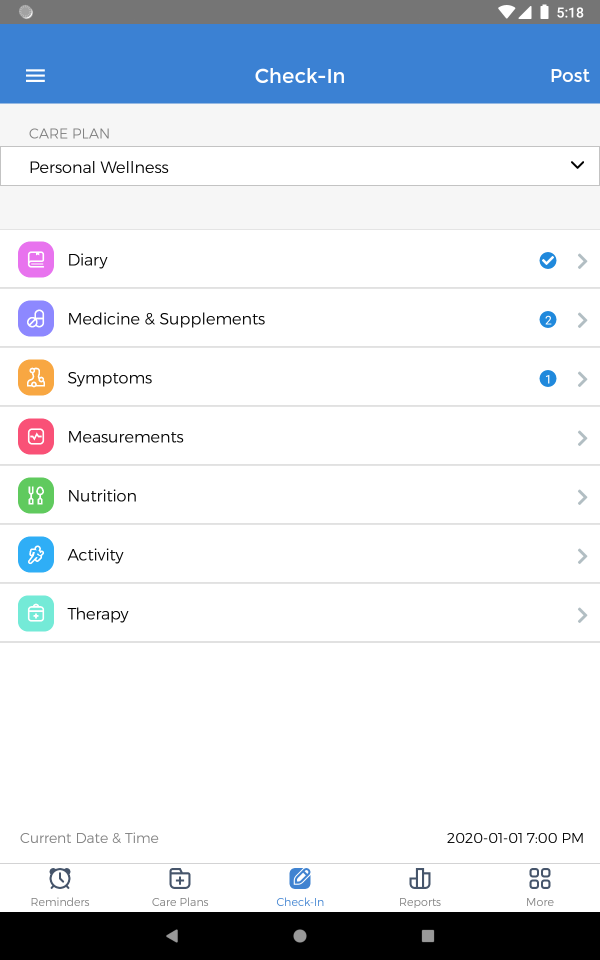
* 1. Описание систем-аналогов

Рассмотрим возможности систем-аналогов разрабатываемого приложения.

* + 1. CareClinic

«CareClinic» - это приложение для отслеживания симптомов или дневника симптомов, это ежедневное универсальное приложение для здоровья и хорошего самочувствия, которое помогает измерять, изучать и улучшать свое здоровье. Данное приложение поможет никогда больше не забыть свои лекарства или таблетки. Это незаменимое приложение для отслеживания приема лекарств и напоминаний для здоровья и благополучия [2].

На рисунке 2 приведена главная экранная форма программы «CareClinic», на которой видны все основные вкладки и функции системы.

  
Рисунок 2 – Экранная форма программы «CareClinic»

Достоинства системы:

* Много различных функций;

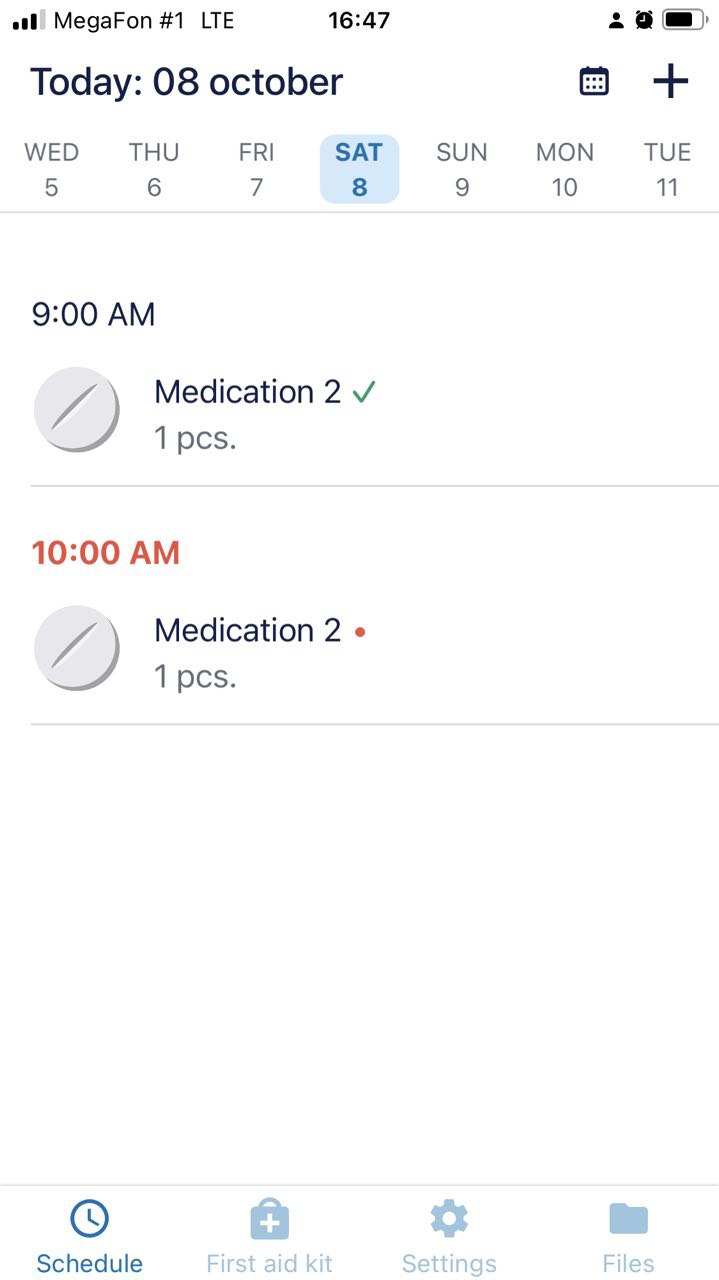
Недостатки системы:

* Бесплатно только пробный период;
* Ежемесячная подписка;
* Отсутствует локализация;
* Много багов и недоработок
  + 1. Pills Time

«Pills Time» - понятная напоминалка помогающая заботиться о своём здоровье. С функциями напоминания, помогающие не забыть принять вовремя таблетки или витамины. «Pills Time» - карманная аптека и удобное расписание приема медикаментов в одном приложении [3].

На рисунке 3 приведена главная экранная форма программы «Pills Time», на которой видны все основные вкладки и функции системы

Рисунок 3 – Экранная форма программы «Pills Time»



Достоинства системы:

* Удобный интерфейс;
* Возможность отслеживать приёмы пищи;
* Возможность отслеживания посещений врачей;

Недостатки системы:

* Нет библиотеки лекарств;
* Вся информация вводится вручную;

На основании анализа возможностей систем-аналогов были сформулированы требования к разрабатываемой системе (см. таблицу 1).

* 1. Постановка задачи

Во время выполнения лабораторного практикума необходимо разработать мобильное приложение с функциями напоминания о необходимости приёма лекарств, с помощью которой можно создавать свой список лекарств и настраивать время, дозировку и периодичность приёма лекарств.

Перед тем как начать пользоваться функциями приложения, каждый пользователь должен пройти авторизацию, а именно: ввести номер телефона и подтверждающий код, полученный по SMS. При последующих входах в приложение будет использоваться механизм быстрой авторизации через touchId/faceId.

После авторизации пользователь может, перейдя на станицу добавления лекарственных препаратов, добавить необходимые ему препараты в свою библиотеку, для поиска предусмотрена система фильтров по заранее заданным критериям: наличие сертификата, название препарата, компонент состава, страна производитель.

Для каждого добавленного лекарства пользователь может создать напоминания о приеме с указанием времени, периодичности и дозировкой.

Сервисная служба будет периодически проверять срабатывание, созданных пользователями напоминаний и отправлять в приложение соответствующие уведомления.

Пользователю всегда доступен его личный кабинет, где он может изменять свои персональные данные и подписываться/отписываться от оповещений.

Также пользователь по каждому отдельному препарату может перейти на страницу с детальным описанием, включающем: сертификат, состав, формы выпуска, а также с ссылками для перехода в интернет магазин для приобретения.

Для более удобного восприятия запланированных приёмов лекарств, пользователю доступен календарь, в котором при выборе даты отображаются все препараты, которые ему необходимо принять в этот день.

Учетные данные пользователя, списки лекарственных препаратов, все справочник, агрегированная информация о напоминаниях будут хранится в БД на сервере.

Таким образом, системы должна решать следующие задачи:

Функции, реализуемые системой:

1) функции системы:

* + аутентификация пользователя в системе;
  + регистрация пользователя в системе;
  + рассылка напоминаний о приёме лекарств;
  + ускоренная авторизация;
  + отслеживание ранее отправленных уведомлений;
  + фильтрация лекарств по параметрам (название, состав, страна производитель, наличие сертификата);
  + автоматическое составление уведомлений для последующей отправки;
  + визуализация процессов работы с приложением;
  + проверка дублирования лекарств;
  + выдача справочной информации о системе;
  + выдача справочной информации о лекарственном препарате;
  + генерирование ссылок для поиска препарата в аптечных сетях;

1. функции пользователя:
   * регистрация в приложении (ввод телефона);
   * авторизация в приложении (ввод кода, отправленного по SMS);
   * возможность при повторной авторизации использовать touchId/faceId;
   * возможность по ссылке перейти в интернет магазин аптечных сетей;
   * возможность изменения своих персональных данных в личном кабинете (имя, e-mail);
   * возможность удалить свой аккаунт;
   * составление фильтров для поиска лекарств;
   * просмотр запланированных приёмов;
   * отслеживание плана приёма лекарства;
   * подтверждение приёма лекарства;
   * добавление лекарств в свою библиотеку;
   * составление графика приёма лекарств;
   * включение/отключение оповещений системы;
   * просмотр справочной информации.

Таблица 1 – Сравнительные характеристики систем-аналогов

|  |  |  |  |
| --- | --- | --- | --- |
| Название системы  Название показателя | «Pills Time» | «CareClinic» | Разрабатываемая система |
| Локализация | да | нет | да |
| Библиотека лекарств | нет | нет | да |
| Поиск лекарств в интернет магазинах | нет | нет | да |
| авторизация | Нет | Да | да |
| Выбор периодичности приёма препаратов | Да | Да | да |
| Просмотр описания препарата | Нет | Нет | да |
| Ограничения на кол-во приёмов одного препарата в день | Да | Да | да |
| IOS/Android версии | да | нет | да |
| Визуализация запланированных приёмов лекарств в виде «карусели» | да | нет | да |

1. Проектирование системы
   1. Выбор и обоснование архитектуры системы

Для разрабатываемого положения была выбрана клиент-серверная архитектура.

Клиент-сервер — вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами. Фактически [клиент](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) и [сервер](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)) — это [программное обеспечение](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5). Обычно эти программы расположены на разных [вычислительных машинах](https://ru.wikipedia.org/wiki/%D0%92%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%B0) и взаимодействуют между собой через [вычислительную сеть](https://ru.wikipedia.org/wiki/%D0%92%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C) посредством [сетевых протоколов](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB) [4].

Серверная часть приложения будет реализована в виде web приложения по модели REST API с поддержкой спецификации Open API в виде реализации swagger. Общение с клиентом происходит посредством запросов/ответов по протоколу HTTP/1.

Клиентская часть будет реализована в виде мобильного приложения для платформ IOS/Android.

* 1. Структурная схема системы

В основе структурного подхода к разработке ПС лежит алгоритмическая декомпозиция, когда система разбивается на функциональные подсистемы, которые, в свою очередь, делятся на подфункции, подразделяемые на задачи и так далее. Процесс разбиения продолжается вплоть до конкретных процедур (алгоритмов). При этом автоматизируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимоувязаны. Разработка системы стала вестись по принципу «сверху-вниз», в отличие от применяемой ранее «снизу-вверх», когда при переходе от отдельных задач ко всей системе терялась целостность и возникали проблемы при информационной стыковке отдельных компонентов [5].

Структурная схема представляется как совокупность элементарных звеньев объекта и связей между ними, один из видов графической модели. Под элементарным звеном понимают часть объекта или системы управления, которая реализует элементарную функцию.

На рисунке 4 приведена структурная схема разрабатываемой системы. В состав разрабатываемой системы входят следующие подсистемы:

На стороне клиента:

1. подсистема взаимодействия с серверной частью, которая отвечает за получение/отправку данных на сервер;
2. подсистема приложение, которая отвечает за связь всех подсистем на уровне приложения;
3. подсистема секреты приложения, которая отвечает за локальное хранилище секретов на устройстве;
4. подсистема уведомления, которая отвечает за получение и обработку уведомлений;
5. подсистема календарь, которая отвечает за работу с календарём;
6. подсистема медицинские препараты, которая отечет з работу с медицинскими препаратами;
7. подсистема визуализации, которая отвечает за смену визуальных состояний приложения;

На стороне сервера:

1. подсистема взаимодействия с клиентской частью, которая отвечает за получение/отправку данных клиенту;
2. подсистема приложение, которая отвечает за связь всех подсистем на уровне приложения;
3. подсистема пользователи, которая отвечает за работу с пользователями;
4. подсистем уведомлений, которая отвечает за отправку уведомлений пользователям;
5. подсистема регистрации, которая отвечает за регистрацию пользователя;
6. подсистема аутентификация, которая отвечает за проверку прав доступ у пользователей;
7. подсистема медицинские препараты, которая отвечает за работу с медицинскими препаратами;
8. подсистема взаимодействия с БД, которая отвечает за генерирование запрос в БД;

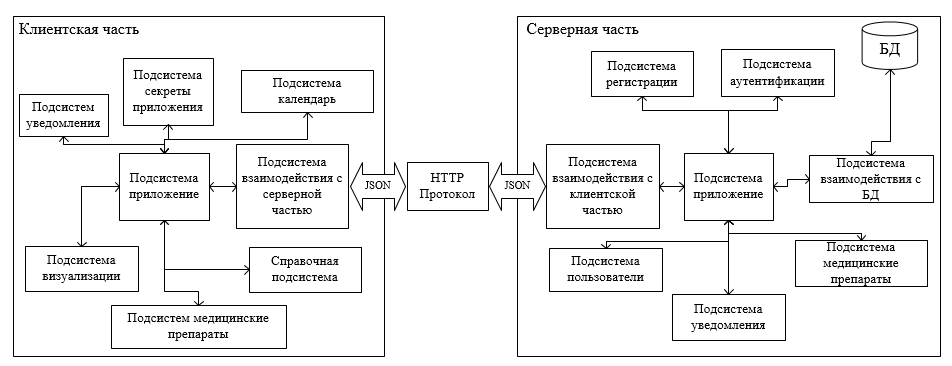


Рисунок 4 – Структурная схема системы

* 1. Разработка прототипа интерфейса пользователя системы

Пользовательский интерфейс – совокупность программных и аппаратных средств, обеспечивающих взаимодействие пользователя с компьютером [6].

Разработка пользовательского интерфейса включает те же основные этапы, что и разработка программного обеспечения

* постановка задачи – определение типа интерфейса и общих требований к нему;
* анализ требований и определение спецификаций – определение сценариев использования и пользовательской модели интерфейса;
* проектирование – проектирование диалогов и их реализация в виде процессов ввода-вывода;
* реализация – программирование и тестирование интерфейсных процессов.

При проектировании пользовательских интерфейсов необходимо учитывать психофизические особенности человека, связанные с восприятием, запоминанием и обработкой информации.

Существует два вида интерфейса: однодокументный (SDI) и многодокументный (MDI).

Однодокументный интерфейс– способ организации графического интерфейса приложений в отдельных окнах. Не существует «фонового» или «родительского» окна, содержащего меню или панели инструментов, по отношению к активному – каждое окно несёт в себе эти элементы. Такие приложения, позволяющие редактировать более одного документа одновременно, например, текстовые процессоры, могут создавать у пользователя впечатление, что запущена не одна копия программы, а несколько [7].

Многодокументный интерфейс – способ организации графического интерфейса пользователя, предполагающий использование оконного интерфейса, в котором большинство окон (исключая, как правило, только модальные окна) расположены внутри одного общего окна. Этим он и отличается от SDI, в котором окна располагаются независимо друг от друга [8]. В данной системе используется интерфейс MDI.

Прототип ПО– это частичная или возможная реализация предлагаемого нового продукта. Прототипы позволяют решать 3 основные задачи [9]:

* Прояснение и завершение процесса формулировки требований. Используемый в качестве формулировки требований прототип представляет собой предварительную версию части системы, понимание которой вызывает затруднения. Оценка прототипа пользователями указывает на ошибки в формулировке требований, которые можно исправить без больших затрат до создания реального продукта.
* Исследование альтернативных решений. Прототип, как инструмент конструирования, позволяет заинтересованным в проекте лицам исследовать различные варианты реализации взаимодействия пользователей, оптимизировать удобство работы и оценить возможные технические приемы. Прототипы позволяют на рабочих образцах показать, насколько осуществимы требования.
* Создание конечного продукта. Используя в качестве инструмента разработки прототип - не что иное, как функциональная реализация первичных элементов системы, которую можно превратить в готовый продукт, осуществляя последовательную цепочку небольших циклов разработки.

Основная цель создания прототипа – устранение неясностей на ранних стадиях процесса разработки.

На рисунке 5 изображена форма авторизации пользователя. Для авторизации пользователю необходимо ввести свой номер сотового телефон в соответствующее поле, нажать кнопку “Получить код”, дождаться получения кода подтверждения по SMS, вести его во 2-е поле и снова нажать кнопку “Проверить код”. Последующие авторизации будут доступны через touchId/faceId.

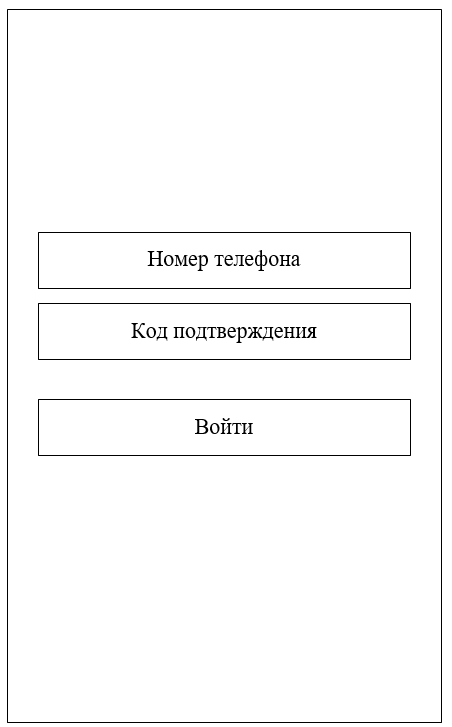


Рисунок 5 – Прототип экранной формы авторизации

После авторизации откроется основная экранная форма приложения – календарь (рисунке 6). Данная форма предназначена для быстрой навигации по прошедшим и предстоящим напоминаниям в разрезе дня. Пользователь может выбрать любой день из списка и увидеть все напоминания в этот день, а также подтвердить, что принял лекарство.

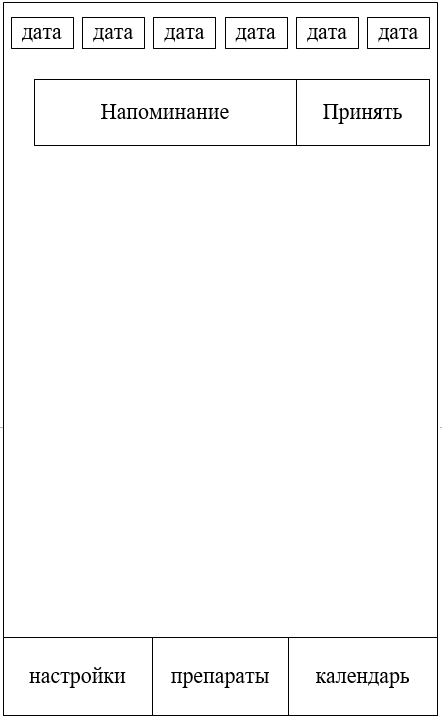


Рисунок 6 – Прототип экранной формы календаря

Каждая основная экранная форма в нижней части имеет область быстрой навигации (рисунок 7). Каждая кнопка перенаправляет пользователя на соответствующего экранного форму.

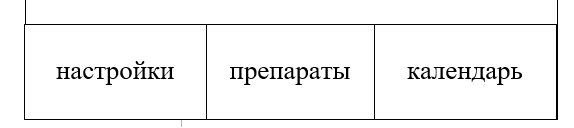


Рисунок 7 – Прототип экранной формы поиска лекарств

На рисунке 8 изображена форма страницы библиотеки лекарств пользователя. На этой странице отображены все лекарственные препараты, которые пользователь добавил в свою библиотеку. Для каждого препарата из списка доступен перечень действий: показать детализацию, настроить уведомления и удалить из списка. Все эти действия вызываются из контекстного меню.

Для добавления нового препарата в список пользователь должен нажать на кнопку “Добавить”, после его откроется форма поиска лекарств (рисунок 9).

Для настройки напоминаний пользователь должен нажать соответствующую кнопку, после чего откроется форма настройки напоминаний (рисунок 10).

Для просмотра описания лекарственного препарата пользователь должен нажать на соответствующую кнопку, после чего откроется форма детализации по препарату (рисунок 11).

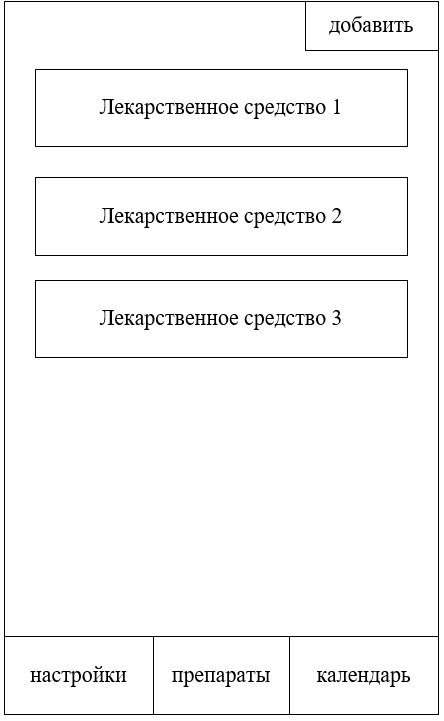


Рисунок 8 – Прототип экранной формы библиотеки лекарств пользователя

На форме поиска лекарственных средств (рисунок 9) пользователю доступны на выбор все лекарственные препараты, добавленные в систему. Для более удобного поиска предусмотрена система фильтров. В фильтре пользователь может выбрать страну изготовителя, указать название препарата, указать элемент состава и отфильтровать по наличию сертификата.

Для добавления препарата в свою библиотеку, пользователь должен нажать советующую кнопку в контекстном меню. После чего пользователя перенаправит на форму со список его лекарственных препаратов (рисунок 8).

Для просмотра описания лекарственного препарата пользователь должен нажать на соответствующую кнопку, после чего откроется форма детализации по препарату (рисунок 11).

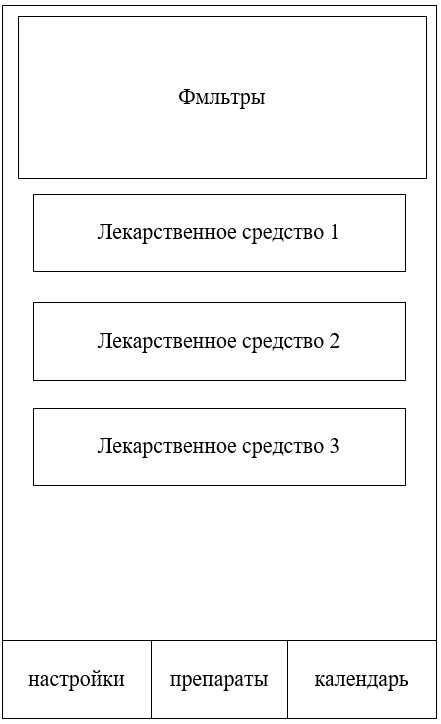


Рисунок 9 – Прототип экранной формы поиска лекарств

На форме создания напоминаний (рисунок 10) пользователю доступен обширный функционал по созданию поминаний для конкретного препарата.

Пользователь может выбрать тип периодичности, дату начала и конца приёма. В зависимости от выбранного типа периодичности появятся дополнительные элементы управления.

Далее пользователю нужно, добавить необходимое ему кол-во напоминании в день, указать для них время приёма и дозировку препарата.

Как только пользователь закончит работу с напоминаниями, ему необходимо нажать кнопку “Готово”, после чего пользователя перенаправит на форму со список его лекарственных препаратов (рисунок 8).

На этой же форме пользователь в любой момент может вносить изменения в ранее созданные напоминания.

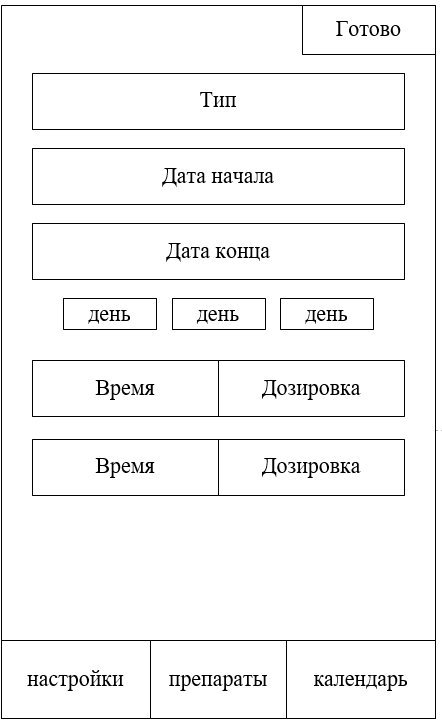


Рисунок 10 – Прототип экранной формы создания напоминаний

На форме просмотра детализации по лекарству (рисунок 11) пользователь может просмотреть всю имеющуюся информацию по препарату такую как: название, состав, формы выпуска, сертификат, прочие характеристики.

А также по соответствующим кнопкам перейти на интернет магазины, соответствующих аптек.

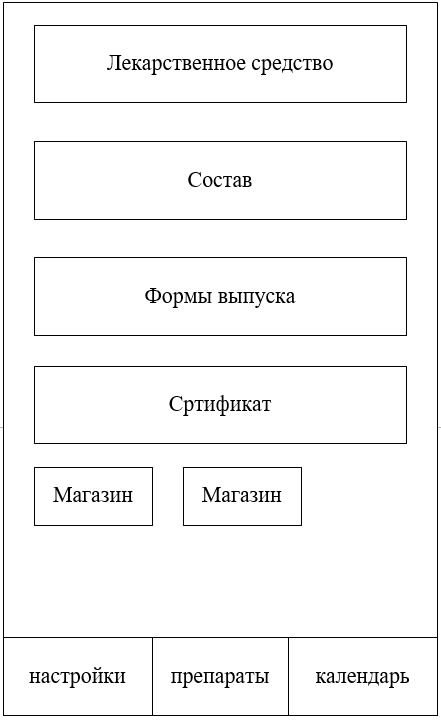


Рисунок 11 – Прототип экранной формы детализации по лекарству

На рисунке 12 изображена форма страницы “помощь”. На данной форме отображена общая информация о приложении, а также информация о разработчиках.

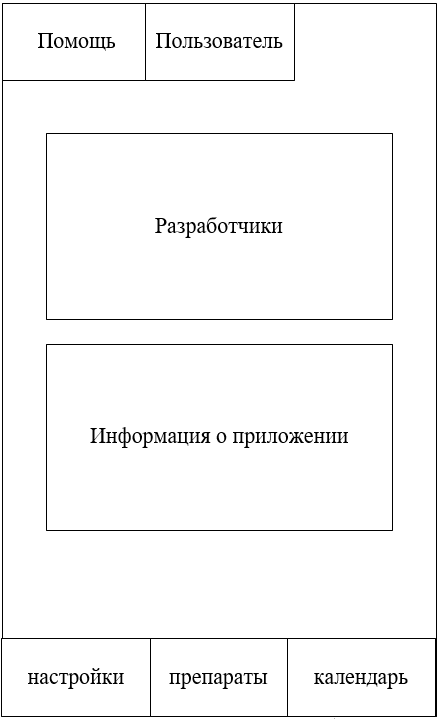


Рисунок 12 – Прототип экранной формы станицы “помощь”

На рисунке 13 изображена форма личного кабинета пользователя, где пользователю предоставляется возможность вносить изменена в свои личные данные путём изменения соответствующих полей, а также включить или отключить подписки на оповещения.

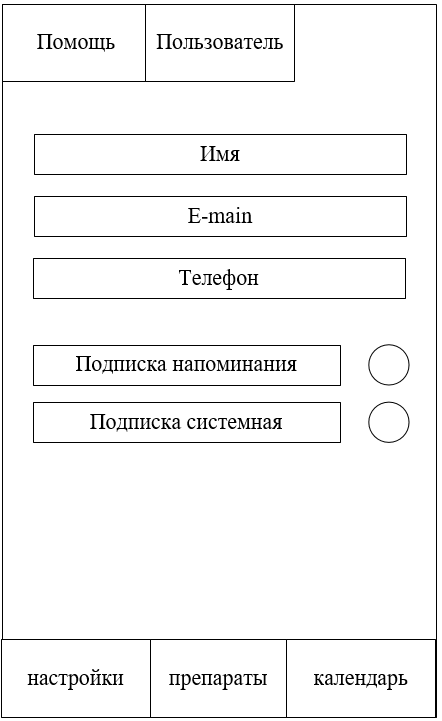


Рисунок 13 – Прототип экранной формы личного кабинета пользователя

На рисунке 14 приведена навигационная модель разрабатываемого приложения.

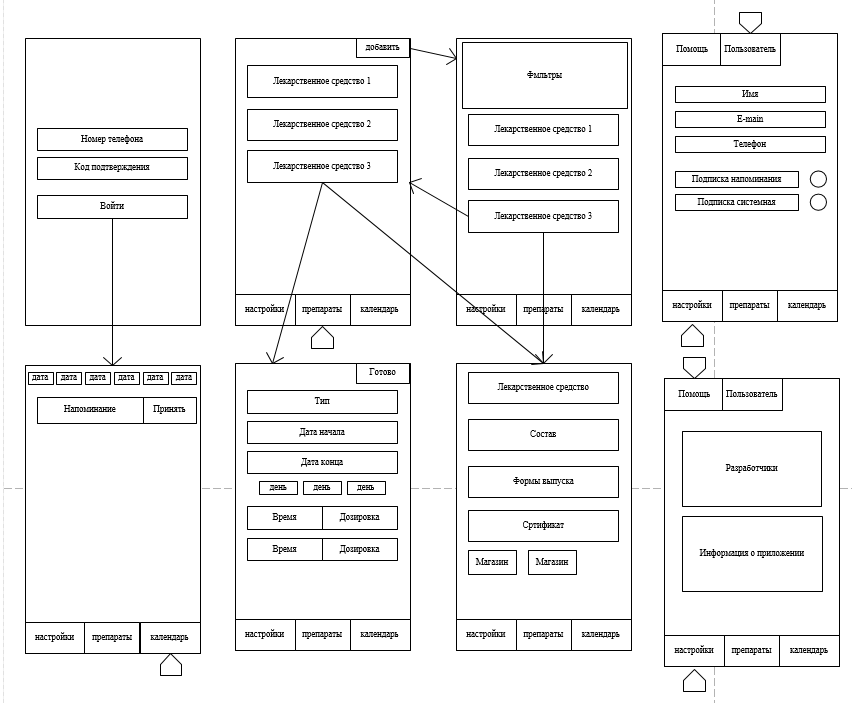
****

Рисунок 14 ‒ Навигационная модель приложения

* 1. Разработка информационно-логического проекта системы
     1. Язык UML

Для специфицирования (построения точных, недвусмысленных и полных моделей) системы и ее документирования используется унифицированный язык моделирования UML.

Язык UML – язык широкого профиля и представляет собой открытый стандарт, в котором используются различные графические обозначения, для создания абстрактной модели системы. Этот язык создавался для того, чтобы обеспечить определение, визуализацию, документирование и проектирование всевозможных программных систем. Важно знать, что сама по себе UML-диаграмма не представляет собой язык программирования, но при этом предусматривается возможность генерации на ее основе отдельного кода на других языках программирования [10].

Применение UML не заключается только в моделировании всевозможного ПО. Также данный язык активно используется для моделирования различных бизнес-процессов, ведения системного проектирования, а также отображения организационных структур. С помощью UML можно обеспечить полное соглашение в используемых графических обозначениях, чтобы представить общие понятия: компонент, обобщение, класс, поведение и агрегация. За счет этого достигается высокая степень концентрации на архитектуре и проектировании.

Использование UML обеспечивает следующие преимущества:

* UML – объектно-ориентированный язык, поэтому методы описания результатов анализа и проектирования очень близки к методам программирования на современных объектно-ориентированных языках;
* UML позволяет описывать систему почти со всех возможных точек зрения и различные аспекты поведения системы;
* диаграммы UML относительно просты для чтения после достаточно быстрого ознакомления с синтаксисом;
* UML – расширяемый язык и позволяет вводить собственные текстовые и графические стереотипы, позволяющие применять ᴇᴦο не только в сфере программной инженерии.
  + 1. Диаграмма вариантов использования

Диаграмма вариантов использования представляет собой наиболее общую концептуальную модель сложной системы, которая является исходной для построения всех остальных диаграмм. На ней изображаются отношения между актерами и вариантами использования.

Диаграмма вариантов использования отображает на себе все отношения, которые возникают между актерами, а также различными вариантами использования и является исходным концептом системы в процессе проектирования и разработки.

Данная диаграмма включает: актеров, варианты использования, отношения между ними.

На рисунке 15 приведена диаграмма вариантов использования.

Основными функциями пользователя являются:

* Пройти авторизацию;
* Добавить препарат в библиотеку;
* Создать напоминание;
* Просмотреть все напоминания;
  + 1. Диаграмма классов

Диаграммы классов – это наиболее часто используемый тип диаграмм, которые создаются при моделировании объектно-ориентированных систем, они показывают набор классов, интерфейсов и коопераций, а также их связи. На практике диаграммы классов применяют для моделирования статического представления системы, они служат основой для целой группы взаимосвязанных диаграмм – диаграмм компонентов и диаграмм размещения [11].

На рисунке 16 приведена диаграмма классов инфраструктурного слоя.

На рисунке 17 приведена диаграмма классов доменного слоя.

В таблице 2 приведено описание классов.

Таблица 2 – Описание классов системы

|  |  |
| --- | --- |
| Название класса | Назначение |
| User | Класс, хранящий в себе всю информацию о пользователе |
| UserProfile | Класс, хранящий в себе всю личную информацию о пользователе |
| MedicationSheduller | Класс, хранящий в себе всю информацию о напоминаниях |
| MedicinalProduct | Класс, хранящий в себе всю информацию о медицинских продуктах |
| MedicinalProductCertificate | Класс, хранящий в себе всю информацию о сертификатах медицинских продуктов |
| CountryDictionary | Класс, хранящий в себе всю информацию о странах |
| MedicinalProductChemical | Класс, хранящий в себе всю информацию о составе медицинских препаратов |
| MedicinalProductImage | Класс, хранящий в себе всю информацию о изображениях для медицинских препаратов |
| UserMedicinalProduct | Класс, хранящий в себе всю информацию о пользовательской библиотеке препаратов |

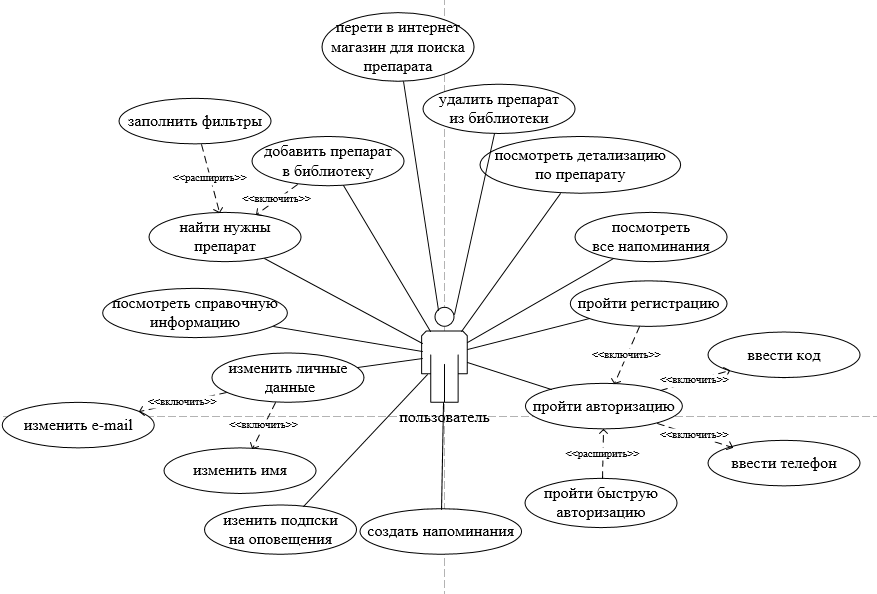


Рисунок 15 − Диаграмма вариантов использования системы

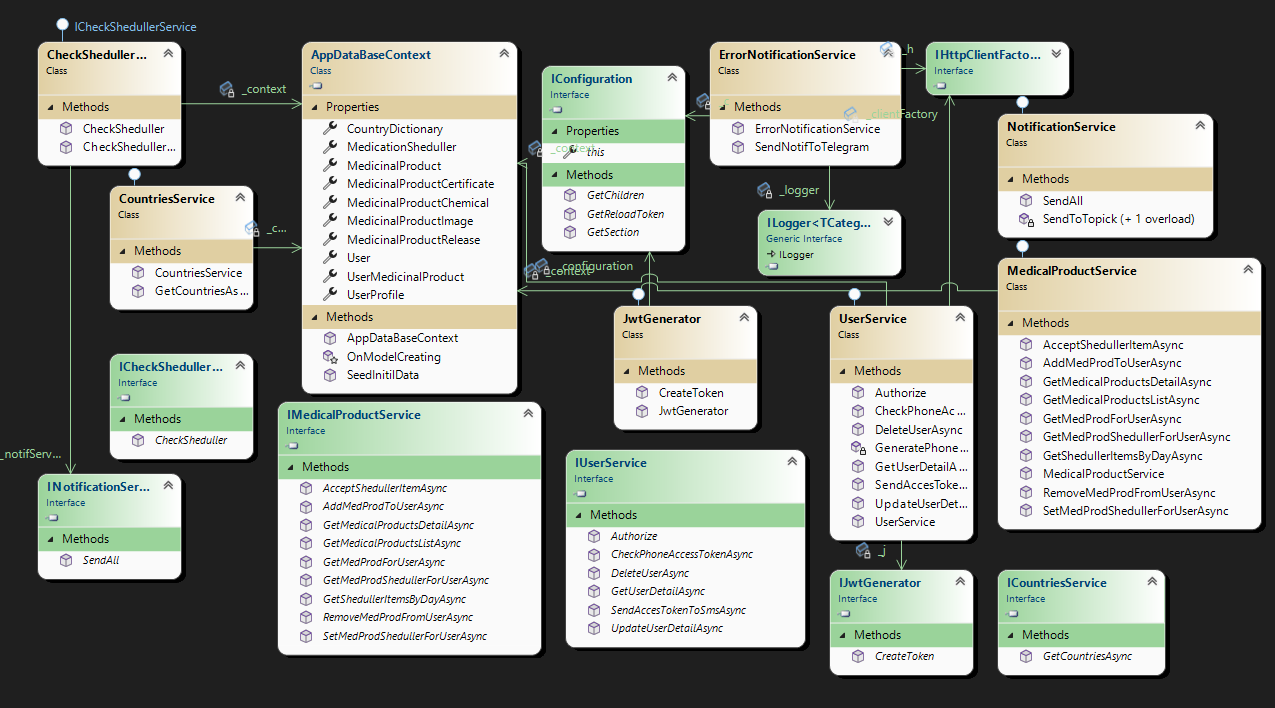


Рисунок 16 − Диаграмма классов инфраструктурного слоя

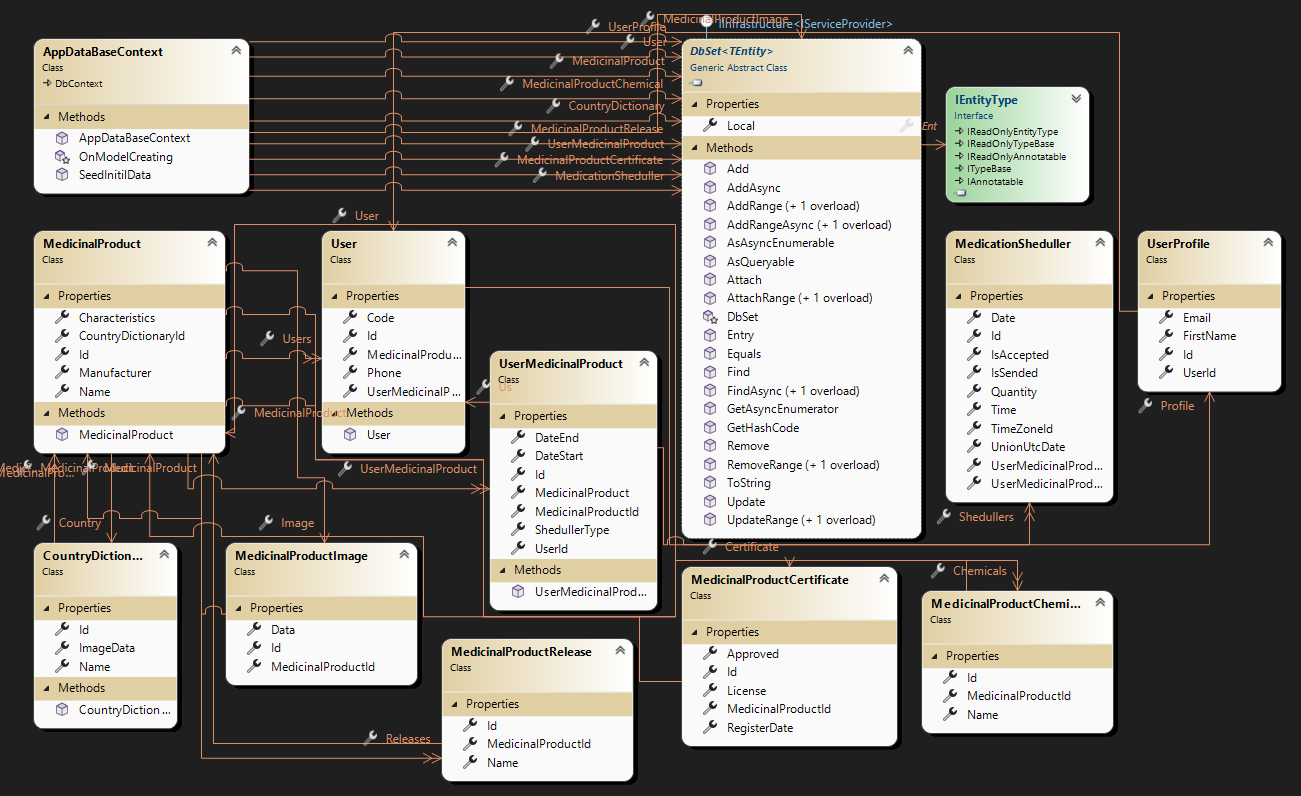


Рисунок 17 − Диаграмма классов доменного слоя

* 1. Логическая модель данных

Проектирование БД является одной из важнейших составных частей процесса создания системы. База данных, рассматриваемая как сложная система, разрабатывается с использованием тех же принципов, что и система в целом. При проектировании баз данных обычно выделяют три уровня абстракции, на которых происходит последовательное уточнение модели: концептуальный (семантический уровень представления данных в виде абстрактных понятий, учитывающих особенности предметной области), логический (уровень представления в виде структуры данных – сущностей, атрибутов и связей) и физический (уровень реализации базы данных) [12].

Логическая информационная модель – модель данных, в которой учитывается способ логического хранения данных в памяти ЭВМ.

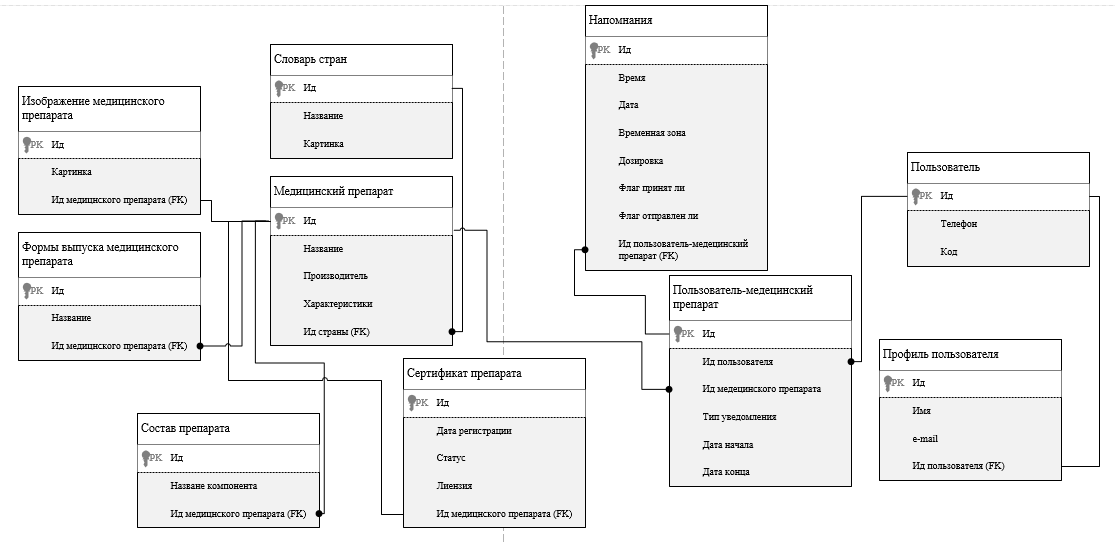
На рисунке 18 приведена логическая модель данных системы. В таблицах 3-4 приведено описание сущностей БД.

Таблица 3 – Сущность «Напоминание»

|  |  |  |
| --- | --- | --- |
| Идентификатор | Тип данных | Описание |
| 1 | 2 | 3 |
| Ид напоминания | Целый | Уникальный идентификатор напоминания |
| Время | Символьный[11] | Телефон, используемое при аутентификации пользователя в системе |
| Дата | Дата | Дата в которую нужно отправить уведомление |
| Временная зона | Время | Время в которое нужно отправить уведомление |
| Дозировка | Целый | Дозировка препарата, которую нужно принять |
| Флаг принят ли | Логический | Флаг показывающий принял ли пользователь препарат |
| Флаг отправлен ли | Логический | Флаг отображающий отправлено ли напоминание пользователю |

Таблица 4 – Сущность «Пользователь»

|  |  |  |
| --- | --- | --- |
| Идентификатор | Тип данных | Описание |
| 1 | 2 | 3 |
| Ид пользователя | Целый | Уникальный идентификатор пользователя |
| Телефон | Символьный[11] | Телефон, используемое при аутентификации пользователя в системе |
| Код подтверждения | Символьный[4] | Сгенерированный код, который сравнивается с полученным от пользователя |

  
Рисунок 18 – Логическая модель данных системы

* 1. Выбор и обоснование алгоритмов обработки данных

Алгоритм — это некая последовательность действий (вычислительных шагов), благодаря чему происходит преобразование входных данных в выходные данные. Алгоритмы используются при обработке данных как в информатике и программировании. алгоритм как последовательность шагов позволяет решать конкретную задачу и должен:

1. Работать за конечный объём времени. Если алгоритм не способен разобраться с проблемой за конечное количество времени, можно сказать, что он бесполезен.

2. Иметь чётко определённые инструкции, порядок. Любой шаг должен точно определяться. Его инструкции должны быть однозначны для любой последовательности шагов.

3. Быть пригодным к использованию. Алгоритм должен быть способен решить проблему, для устранения которой его создавали [13].

На рисунке 19 приведена схема алгоритма наложения фильтров на запрос в базу данных для поиска лекарств.

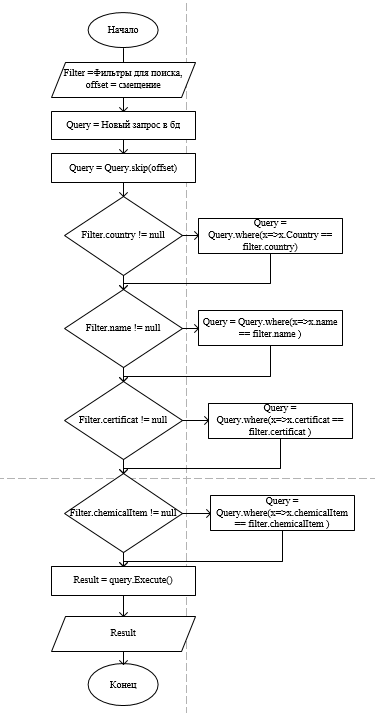


Рисунок 19 – схема алгоритма наложения фильтров на запрос в базу данных для поиска лекарств

* 1. Выбор и обоснование комплекса программных средств

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст [??].

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

* + 1. Выбор операционной системы

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст [??].

* + 1. Выбор языка программирования

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  [??].

* + 1. Выбор среды разработки

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст. Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  [??].

* + 1. Выбор системы управления базами данных

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст. Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  [??].

1. Реализация системы
   1. Разработка и описание интерфейса пользователя

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

  
Рисунок ХХХ – Сведения о разработчиках

* 1. Диаграммы реализации

Диаграммы реализации предназначены для отображения состава компилируемых и выполняемых модулей системы, а также связей между ними. Диаграммы реализации разделяются на три конкретных вида: диаграмма классов, диаграмма компонентов (component diagrams) и диаграмма развертывания (deployment diagrams) [ХХХ].

* + 1. Диаграмма развертывания

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

На рисунке ХХХ приведена диаграмма развертывания системы. Здесь должно быть описание тех компонентов, которые развернуты на узлах ЭВМ.

Рисунок ХХХ – Диаграмма развертывания системы

* + 1. Диаграмма классов

В соответствии со спецификацией, приведенной в п. 2.5.6, и с учетом выбранного языка программирования (см. п. 2.8.1) разработана диаграмма классов системы (этап реализации), приведенная на рисунке ХХХ.

Рисунок ХХХ – Диаграмма классов системы (этап реализации)

* 1. Физическая модель данных

Физический уровень представления данных является самым низким уровнем, на котором определяются форматы хранимых данных, способ их размещения на носителях, выбор способа доступа к ним. Физическая модель БД, разработанной в системе, представлена на рисунке ХХХ.

В процессе проектирования при переходе от концептуальной модели к логической, а затем к физической наблюдается соответствие между основными категориями (таблица ХХХ).

Таблица ХХХ – Соответствие основных категорий

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица | Сущность | Отношение | Файл |
| Строка | Экземпляр | Кортеж | Запись |
| Столбец | Атрибут | Домен | Поле |

Подробное описание атрибутов сущностей физической модели данных приведено в таблицах ХХХ-ХХХХ. Первичные ключи выделены жирным шрифтом, а внешние – курсивом.

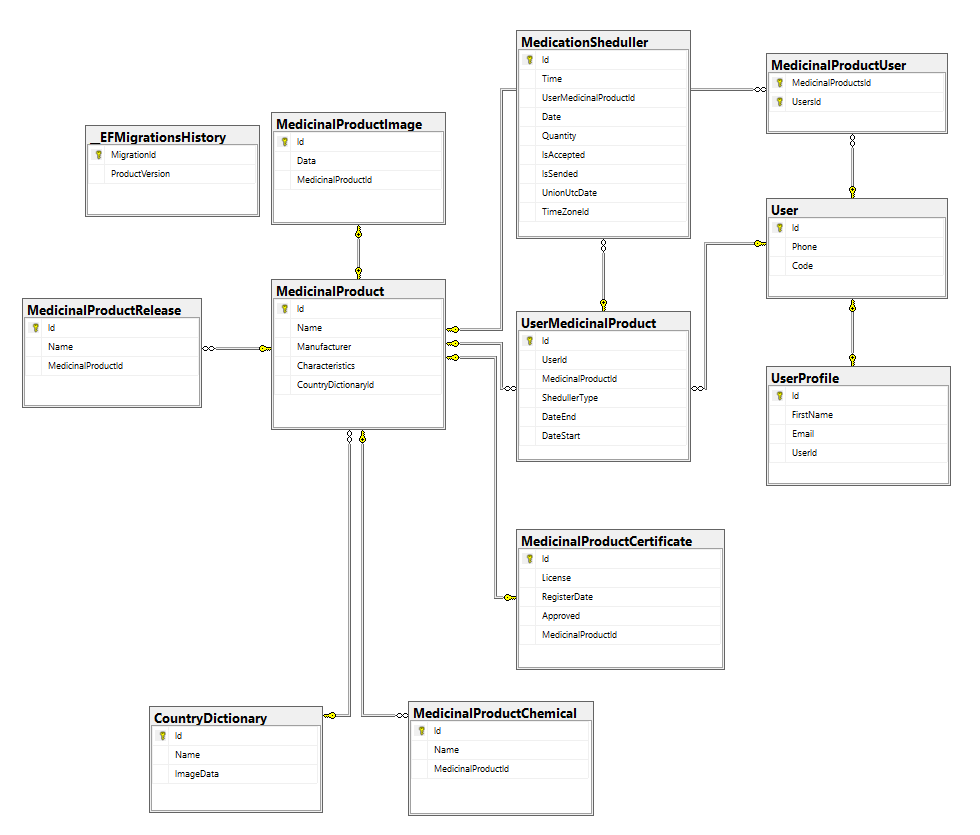


Рисунок XXX – Физическая модель базы данных

Таблица ХХХХ – Таблица «Эксперимент»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Имя атрибута | Тип | Размер (байт) |
| **experiment id** | **код эксперимента** | **int** | **4** |
| experiment name | название эксперимента | varchar(150) | 150 |
| experiment date | дата эксперимента | date | 3 |
| *login* | *логин пользователя, проводившего эксперимент* | *varchar(30)* | *30* |
| *mode id* | *код моделируемого режима* | *int* | *4* |
| *engine id* | *код двигателя МКА* | *int* | *4* |
| Размер записи | | | 195 |

* 1. Выбор и обоснование комплекса технических средств
     1. Расчет объема занимаемой памяти

Для расчета необходимого объема свободной внешней памяти, необходимой для функционирования системы, воспользуемся следующей формулой:

VЖД = VОС + VПР + VСПО + VБД + Vсправки,

где VОС – объем памяти, занимаемый операционной системой (операционная система Windows 7 Professional 64 бит с пакетом обновлений SP1,   
VОС = 20 Гб);

VПР – объем памяти, занимаемый непосредственно файлами приложения (VПР = 2 Мб);

VСПО – объем памяти, занимаемый сопутствующим программным обеспечением (библиотеки cryptopp.dll, simplexlsx.dll, sqlite3.dll, sqlitecpp.dll, Qt Framework 5.11.1, Internet Explorer 9; дадим оценку сверху VСПО в 3 Гб);

VБД – объем памяти, занимаемый базой данных (всеми таблицами) при ее максимальном заполнении. Расчет этой составляющей приведен в таблице ХХХ (VБД = ???? байт = ??? Кб = ??? Мб = ??? Гб);

Vсправки – объем памяти, необходимый для хранения файла справки (Vсправки =0,8 Мб).

Таким образом, суммарный объем внешней памяти составит:

VЖД = 20 Гб + 2 Мб + 3 Гб + ??? Мб + 1 Мб ~ ??? Гб.

Расчет объема ОЗУ

Для расчета необходимого объема ОЗУ воспользуемся следующей формулой:

Таблица ХХХ – Расчет объема внешней памяти, необходимой для хранения БД

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица | Размер записи (байт) | Максимум записей | Всего (байт) |
| Пользователь | 94 | 10 | 940 |
| Сотрудник |  | 30 |  |
| Статус сотрудника |  | 10 |  |
| Должность сотрудника |  | 10 |  |
| Место работы |  | 10 |  |
| Кафедра |  | 10 |  |
| ОУ ВО |  | 10 |  |
| Итого | | |  |

VОЗУ = VОС + VПР + VБД + Vбраузера,

где VОС – ОЗУ, занимаемое операционной системой (2 Гб);

VПР – ОЗУ, которое займет само приложение (не превысит 80 Мб);

VБД – объем данных из базы, который может быть одновременно загружен в оперативную память (дадим ему оценку сверху в 10 Мб).

Vбраузера – ОЗУ, занимаемое браузером (оценим его сверху значением в 100 Мб).

Суммарные объемы ОЗУ составит:

VОЗУ = 2 Гб + 80 Мб + 10 МБ + 100 Мб ~ 2.2 Гб.

Таким образом, 2.2 Гб оперативной памяти можно счесть минимально необходимым для функционирования системы.

* + 1. Минимальные требования, предъявляемые к системе

Для корректного функционирования системы необходимо:

* тип ЭВМ: x86-64 совместимый;
* объем ОЗУ – не менее 3 Гб;
* объем свободного дискового пространства – не менее ??? Гб;
* клавиатура или иное устройство ввода;
* мышь или иное манипулирующее устройство;
* процессор – Intel Pentium не менее 1,5 ГГц;
* дисплей с разрешением не менее 1024 × 768 пикселей;
* операционная система Windows 7 и выше;
* браузер Internet Explorer 9 и выше;
* Qt framework 5.11 и выше.

ЗАКЛЮЧЕНИЕ

В процессе выполнения лабораторного практикума была разработана автоматизированная система …, позволяющая ….

В заключении должны быть отражены основные результаты работы, необходимо сделать это с привязкой к разделам отчета, например:

В первом разделе приведены основные понятия предметной области, характеристики систем-аналогов и результаты их сравнительного анализа, на основании этого выполнена объектная декомпозиция, отраженная в диаграмме объектов. Сформулирована постановка задачи.

Во втором разделе …

В третьем разделе …

Разработанная система будет полезна…

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Буч Г., Рамбо Д., Якобсон А. Язык UML. Руководство пользователя. Изд. 2-е. М.: ДМК Пресс, 2006. 546 с.
2. Google Play [Электронный ресурс]. – URL: https://play.google.com/store/apps/details?id=com.careclinicsoftware.careclinic&gl=NL (дата обращения: 08.10.2022).
3. App Store [Электронный ресурс]. – URL: <https://apps.apple.com/ru/app/id1248342340> (дата обращения: 08.10.2022).
4. Wikipedia [Электронный ресурс]. – URL: https://ru.wikipedia.org/wiki/Клиент\_—\_сервер (дата обращения: 08.10.2022).
5. Функциональная спецификация [Электронный ресурс] - URL: https://studref.com/351180/informatika/funktsionalnaya\_spetsifikatsiya\_programmnogo\_sredstva (дата обращения: 08.10.2022).
6. Диаграмма компонентов [Электронный ресурс] // Википедия: электронная энциклопедия. 2001-2018. URL: https://ru.wikipedia.org/wiki/ Диаграмма\_компонентов (дата обращения 08.10.22).
7. Разработка структуры программы и модульное программирование [Электронный ресурс]. URL: <http://vit-prog.narod.ru/page/TRPP/> (дата обращения 08.10.22).  
   section\_1/subject\_1.3.htm (дата обращения 08.10.22).
8. Разработка пользовательских интерфейсов [Электронный ресурс] // Pandia: [сайт]. URL: <http://pandia.ru/text/78/247/74988.php> (дата обращения 08.10.22).
9. Однодокументный интерфейс [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/Однодокументный_интерфейс> (дата обращения 08.10.22).
10. UML-диаграммы. Виды UML диаграмм [Электронный ресурс]. URL: <https://www.syl.ru/article/206012/new_uml-diagramma-vidyi-diagramm-uml> (дата обращения: 09.10.2022).
11. Построение диаграммы классов [Электронный ресурс]. URL: https://flexberry.github.io/ru/gpg\_class-diagram.html (дата обращения: 09.10.2022).
12. Логическая модель базы данных [Электронный ресурс]. URL: https://studfile.net/preview/2083751/page:2/ (дата обращения: 09.10.2022).
13. Алгоритмы [Электронный ресурс]. URL: https://otus.ru/nest/post829 (дата обращения: 09.10.2022).

**Книги**

***Целиком***

Буч Г., Рамбо Д., Якобсон А. Язык UML. Руководство пользователя. Изд. 2-е. М.: ДМК Пресс, 2006. 546 с.

***Если нужно указать номера конкретных страниц***

1. Буч Г., Рамбо Д., Якобсон А. Язык UML. Руководство пользователя. Изд. 2-е. М.: ДМК Пресс, 2006. С. 21.

***Если повторная ссылка на тот же документ***

1. Буч Г., Рамбо Д., Якобсон А. Язык UML … С. 31.

***Если больше 3 авторов***

1. Нестационарная аэродинамика баллистического полета/ Липницкий Ю.М. и [др.]. М.: Физматлит, 2003. 176 с.

**Журналы**

1. Зеленко Л.С., Шумская Е.А. Комплекс программ для работы с учебным контентом в дистанционных обучающих системах// Известия СНЦ РАН. 2015. №2 (5). Т. 17. С. 992-1003.

**Руководящие материалы и ГОСТы**

1. РД 34.20.571. Методические указания по расчету показателей готовности к работе электростанции и энергосистем. Введ. 1976-10-22. М., 1976. 25 с.
2. ГОСТ Р 7.0.4-2006. Издания. Выходные сведения. Общие требования и правила оформления. М., 2006. II. 43 с. (Система стандартов по информ., библ. и изд. делу).

**Методические указания или учебные пособия**

1. Зеленко Л.С. Методические указания к лабораторному практикуму по дисциплине «Программная инженерия». Самара: СГАУ, 2012. 67 с.

**Электронные ресурсы**

1. Российская гидроэнергетика [Электронный ресурс] // Русгидро: [сайт]. URL: http://www.rushydro.ru/industry/russianhydropower/ (дата обращения: 20.12.2022).
2. Гидроэлектростанция (гидроэлектрическая станция, ГЭС) // Энциклопедический словарь юного техника М.: Издательство «Педагогика», 1987 [Электронный ресурс] // Библиотекарь.Ру: электрон. библ. 2006-2017. URL: http://www.bibliotekar.ru/enc-Tehnika/58.htm (дата обращения: 20.12.2022).
3. Субботин А.С. Основы гидротехники [Электронный ресурс]. URL: http://www.cawater-info.net/bk/dam-safety/files/subbotin.pdf (дата обращения: 03.02.2022).
4. Филиальная структура компании [Электронный ресурс] // Системный оператор Единой энергетической системы: [сайт]. [2009-2017]. URL: http://so-ups.ru/index.php?id=about (дата обращения: 20.12.2022).
5. Автоматизированные системы управления технологическими процессами гидроэлектростанции [Электронный ресурс] // Микроника. Инжиниринговый центр: [сайт]. [1999-2016]. URL: http://mikronika-energo.ru/products/asutp/ges-asu-tp/ (дата обращения: 24.12.2022).
6. Автоматизированная система управления производственными процессами [Электронный ресурс] // MEScontrol: [сайт]. [2003-2017]. URL: http://mescontrol.ru/articles/systems (дата обращения: 02.04.2022).
7. Пушников А.Ю. Введение в системы управления базами данных: учеб. пособие [Электронный ресурс] // CITForum: электрон. библиотека. 1997-2017. URL: https://citforum.ru/database/dblearn/ dblearn06.shtml (дата обращения: 20.12.2022).
8. Пользовательский интерфейс [Электронный ресурс] // Википедия: электрон. энциклопедия. 2001-2017. URL: https://ru.wikipedia.org/ wiki/Пользовательский\_интерфейс (дата обращения: 17.03.2022).

***Если необходимо указать системные требования для доступа к документу (наличие специального ПО), то***

1. Белова С.В. Язык UML. Диаграмма вариантов использования. Систем. требования: PowerPoint. URL: nkse.ru/component/k2/item/  
   download/7\_754f5a247edc6ec6be78218f187338a5.html (дата обращения: 17.03.2017).

**Сборники научных трудов или трудов конференций**

1. Философия культуры и философия науки: проблемы и гипотезы: межвуз. сб. науч. тр./ Саратов. гос. ун-т; [под ред. С.Ф. Мартыновича]. Саратов: изд-во Сарат. ун-та, 1999. 199 с.
2. Акимова А.Е., Трешников А.А., Зеленко Л.С. Информационная среда ГЭС. Подсистема расчета показателей эффективности работы оборудования // Перспективные информационные технологии (ПИТ-2017): сб. науч. тр. межд. научно-техн. конф.; [под ред. С.А. Прохорова]. Самара: Изд-во СНЦ РАН, 2017. С. 41-44.

***Если электронное издание***

1. Акимова А.Е., Трешников А.А., Зеленко Л.С. Подсистема расчета показателей эффективности работы оборудования // Математика. Компьютер. Образование: труды XXIV межд. конф., 23-28 января 2017 г., г. Пущино. URL: http://www.mce.su/rus/presentations/ p283063/ (дата обращения: 02.03.2017).

ПРИЛОЖЕНИЕ А  
Руководство пользователя

А.1 Назначение системы

Приводится краткое описание возможностей системы.

А.2 Условия работы системы

***Пример.***

Для корректной работы системы необходимо наличие соответствующих программных и аппаратных средств.

1) Требования к техническому обеспечению:

* ЭВМ типа IBM PC;
* процессор типа x86 или x64 тактовой частоты 1400 МГц и выше;
* …

2) Требования к программному обеспечению:

* операционная система Windows 7 и выше;
* установленная платформа .Net версии 4.0 и выше;
* установленная СУБД ….

А.3 Установка системы

***Пример.***

Система поставляется в виде zip-архива. Данный файл необходимо распаковать в любую директорию на жестком диске. Запускаемым файлом системы является файл ххх.exe.[[1]](#footnote-1)

А.4 Работа с системой

А.4.1 Работа с системой в режиме администратора (если необходимо)

Вход в систему (авторизация)

…

А.4.2 Работа с системой в режиме пользователя

Вход в систему (авторизация)

Вход в систему (регистрация)

Настройка параметров кроссворда

ПРИЛОЖЕНИЕ Б   
Листинг модулей программы

7-10 страниц исходного кода шрифт Times New Roman 10 пт 1 интервал

1. Если необходимы дополнительные ресурсы для обеспечения работоспособности системы, то все для них также должны быть перечислены условия установки. *Если установка нестандартная, то она должна быть подробно описана (в объеме, достаточном для понимания пользователя).* [↑](#footnote-ref-1)