**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Образовательная программа «Программная инженерия»

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО  Штатный преподаватель  департамента программной  инженерии, канд. экон. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.Ю. Песоцкая  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г. |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия»  профессор департамента программной инженерии, канд. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** | RU.17701729.06.05-01 81 01-1 | | **СЕРВИС-ПОМОЩНИК “ОРГАНАЙЗЕР ЛЕКАРСТВ”**  **Пояснительная записка**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.06.05-01 81 01-1-ЛУ** | | |
|  |  | |
| Исполнитель  студент группы БПИ215  \_\_\_\_\_\_\_\_\_ / С. А. Дрозд /  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2023 г. | |
|  | | |
|  | |  |

**Москва 2023**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** | RU.17701729.06.05-01 81 01-1 | | УТВЕРЖДЕН  RU.17701729.06.05-01 81 01-1-ЛУ  **СЕРВИС-ПОМОЩНИК “ОРГАНАЙЗЕР ЛЕКАРСТВ”**  **Пояснительная записка**  **RU.17701729.06.05-01 81 01–1**  **Листов 21** | | | | |
|  |  | | | |
|  | | | |
|  | | | | |
|  | | |  | |

**Москва 2023**

**СОДЕРЖАНИЕ**

[1. ВВЕДЕНИЕ 4](#_Toc134529132)

[1.1. Наименование программы: 4](#_Toc134529133)

[1.2. Документ, на основании которого ведется разработка 4](#_Toc134529134)

[2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ РАЗРАБОТКИ 5](#_Toc134529135)

[2.1. Назначение программы 5](#_Toc134529136)

[2.1.1. Функциональное назначение 5](#_Toc134529137)

[2.1.2. Эксплуатационное назначение 5](#_Toc134529138)

[2.2. Краткая характеристика области применения 5](#_Toc134529139)

[3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ 6](#_Toc134529140)

[3.1. Постановка задачи на разработку программы 6](#_Toc134529141)

[3.2. Описание алгоритмов и функционирования программы 7](#_Toc134529142)

[3.2.1. Общее описание работы программы 7](#_Toc134529143)

[3.2.2. Работа с аптечками и категориями товаров 7](#_Toc134529144)

[3.2.3. Работа с базой данных медикаментов 7](#_Toc134529145)

[3.2.4. Алгоритм занесения нового медикамента в базу данных 8](#_Toc134529146)

[3.2.5. Организация хранения данных о лекарственных препаратах 9](#_Toc134529147)

[3.2.6. Работа с календарем и оповещениями о приеме лекарств 9](#_Toc134529148)

[3.2.7. Алгоритм создания напоминаний о приеме лекарства 10](#_Toc134529149)

[3.3. Описание и обоснование выбора метода организации входных и выходных данных……………………………………………………………………………………………..11](#_Toc134529150)

[3.3.1. Описание метода организации входных данных 11](#_Toc134529151)

[3.3.2. Обоснование метода организации входных данных 13](#_Toc134529152)

[3.3.3. Описание метода организации выходных данных 14](#_Toc134529153)

[3.3.4. Обоснование метода организации выходных данных 14](#_Toc134529154)

[3.4. Описание состава и обоснование выбора технических и программных средств……………………………………………………………………………………………...14](#_Toc134529155)

[3.4.1. Описание состава технических и программных средств 14](#_Toc134529156)

[3.4.2. Обоснование выбора технических и программных средств 14](#_Toc134529157)

[4. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 16](#_Toc134529158)

[4.1. Предполагаемая потребность 16](#_Toc134529159)

[4.2. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами 16](#_Toc134529160)

[4.2.1. Medisafe 17](#_Toc134529161)

[4.2.2. Аптечка 17](#_Toc134529162)

[4.2.3. Мои таблетки 17](#_Toc134529163)

[4.2.4. Сравнительная таблица 18](#_Toc134529164)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 19](#_Toc134529200)

[Приложение 1 20](#_Toc134529201)

[ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ 21](#_Toc134529203)

1. ВВЕДЕНИЕ
   1. Наименование программы:

Наименование программы на русском языке: «Сервис-помощник “Органайзер лекарств”».

Наименование программы на английском языке: «Assistant “Medicine organizer”».

* 1. Документ, на основании которого ведется разработка

Основанием для разработки является учебный план подготовки бакалавров по  
направлению 09.03.04 «Программная инженерия» и утвержденная академическим  
руководителем тема курсового проекта.

1. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ РАЗРАБОТКИ
   1. Назначение программы
      1. Функциональное назначение

Функциональным назначением программы является сбор и систематизация информации о купленных медикаментах, а также визуализация этой информации посредством списка. Собранные данные могут использоваться с целью организации хранения лекарств и отслеживания их сроков годности. Программа предоставляет возможность напоминания о сроках приема медицинских препаратов посредством push-уведомлений в сроки, установленные пользователем для конкретного медикамента.

* + 1. Эксплуатационное назначение

Основными конечными потребителями разрабатываемого приложения являются люди, которые хотят хранить и обрабатывать информацию о купленных лекарствах с целью контроля за наличием необходимых препаратов и своевременного обновления аптечки.

Для работы программы требуется планшетный компьютер, смартфон либо другое устройство, на котором установлена операционная система Android.

* 1. Краткая характеристика области применения

«Сервис-помощник “Органайзер лекарств”» – это Android-приложение, созданное для автоматизации ведения учета купленных лекарственных препаратов, отслеживания их сроков годности, а также напоминаний о приёме лекарств. Пользователь может добавлять купленные медикаменты в базу данных, разбивать их по категориям и просматривать информацию обо всех лекарствах.

Область применения – сфера здравоохранения. Приложение предназначено для личного пользования в качестве программы для организации домашней, автомобильной или других видов аптечек.

1. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ
   1. Постановка задачи на разработку программы

Приложение должно иметь следующие возможности:  
  
1. Хранение информации о наименовании медикамента;  
2. Хранение информации о сроке годности медикамента;  
3. Хранение информации о количестве медикамента в шт./мл.;  
4. Отображение хранимых в базе данных лекарств в виде списка;  
5. Возможность сортировки списка медикаментов по сроку годности в порядке возрастания (от самой ранней даты до самой поздней даты истечения срока годности);  
6. Возможность сортировки списка медикаментов в алфавитном порядке;  
7. Возможность удаления медикамента из базы данных;  
8. Возможность обновления информации о медикаменте в базе данных;  
9. Создание пользовательских категорий (аптечек);  
10. Возможность добавления медикаментов в аптечку;  
11. Возможность удаления медикаментов из аптечки;  
12. Возможность добавления аптечек;  
13. Возможность удаления аптечек;  
14. Отображения календаря с напоминаниями о приеме лекарств;  
15. Возможность добавления напоминаний о приемах лекарств с указанием названия препарата из базы данных (или же любого другого сообщения на усмотрение пользователя), периода напоминания и времени напоминания;  
16. Отправка push-уведомлений для напоминания о приеме лекарств в установленные пользователем время и даты.

Описание алгоритмов и функционирования программы

* + 1. Общее описание работы программы

При запуске приложения сначала начинает работать класс MainActivity, в котором хранится базовая информация для начала работы приложения. Отдельная регистрация пользователя в системе не требуется.

В приложении есть три основных раздела: «Календарь» (за его работу отвечает класс CalendarFragment), «Аптечки» (за его работу отвечает класс KitsFragment) и «Медикаменты» (за его работу отвечает класс MedsFragment). По умолчанию при открытии приложения отображается раздел «Календарь». Пользователю предлагается перемещаться между различными разделами посредством кликов на нижнюю панель навигации.

* + 1. Работа с аптечками и категориями товаров

Отображение аптечек (или категорий) лекарств осуществляется в разделе «Аптечки». За работу с аптечками отвечают классы, расположенные внутри Java пакета kit, а также используется DatabaseHandler для получения информации из базы данных при добавлении новых медикаментов в определенную аптечку. Одно и то же лекарство может храниться в разных аптечках одновременно.

Для отображения аптечек используется ExpandableList, за работу с которым отвечает KitsExpandableListAdapter. Так как названия аптечек не повторяются, наиболее оптимальным решением было хранить их с использованием Hashmap с названием аптечки в качестве ключа и списком названий медикаментов (List<String>) в качестве значения. Основная информация о составе аптечек хранится в классе KitsModel, а обработка удаления и добавления аптечек осуществляется с помощью KitsExpandableListAdapter.

* + 1. Работа с базой данных медикаментов

Для работы с SQLite базой данных медикаментов используются классы, расположенные внутри Java пакета med, а также при заполнении сроков годности используется DatePicker.

Медикаменты отображаются в виде списка RecyclerView, где каждая из отдельных карточек содержит информацию о наименовании, сроке годности и количестве препарата. По умолчанию список отсортирован в алфавитном порядке. При нажатии на кнопку в правом верхнем углу экрана можно изменить вид сортировки на сортировку по сроку годности.

За редактирование списка медицинских препаратов посредством жестов смахивания отвечает класс RecyclerItemTouchHelper, который следит за жестами пользователя и после смахивания элемента списка влево и вправо вызывает методы удаления и редактирования соответственно.

Добавление и редактирование осуществляются с помощью класса AddNewMed. Поля содержат информацию о медикаменте, и при нажатии клавиши «Ок» запускаются проверки корректности. В случае успеха в базу данных вносится информация об изменениях, в противном случае всплывает диалоговое окно с объяснением ошибки.

Модель каждого лекарства представляет собой объект класса MedModel с приватными полями id, med, date, amount.

В качестве базы данных используется SQLite. Информация о каждом из лекарств хранится в базе данных, взаимодействие с которой осуществляется с помощью класса DatabaseHandler. Изменения осуществляются посредством вызова методов для вставки (insertMed), редактирования (updateMed) и удаления (deleteMed) медикаментов, а также есть метод getAllMeds для получения списка всех моделей в виде List<MedModel>.

* + 1. Алгоритм внесения нового медикамента в базу данных

Внесение нового медикамента в базу и редактирование информации об уже существующем в базе медикаменте осуществляется с помощью класса AddNewMed. Здесь поля принимают определенные значения в зависимости от введенных пользователем данных и далее обрабатываются. Если пользователь не хочет фиксировать изменения, он может выйти со страницы добавления или изменения медикамента. Иначе по нажатию кнопки «Ок» запускается механизм проверок. В случае некорректных данных пользователь получает сообщение об ошибке и возвращается на страницу редактирования. В ином случае проверяется, какой из процессов именно был запущен (добавления или редактирования), а после в зависимости от этого вызывается соответствующий метод класса DatabaseHandler (или updateMed, или insertMed).

* + 1. Организация хранения данных о лекарственных препаратах

В качестве базы данных для хранения информации о медикаментах была выбрана реляционная база данных SQLite. Выбор пал на нее по нескольким причинам. Учитывая, что приложение разработано для личного использования, а количество лекарств в домашней аптечке среднестатистически невелико, объем данных относительно небольшой. Для хранения таких данных SQLite является оптимальным решением, так как она легковесна и проста в использовании. С ее помощью можно хранить файлы на диске, что удобно, если принять во внимание тот факт, что приложение не требует регистрации пользователя в облачных системах или удаленного доступа к базам данных. Также SQLite производительна и быстродейственна, что позволяет оперативно и безопасно обрабатывать данные. По всем этим причинам SQLite часто используется в качестве базы данных для мобильных приложений, работающих с относительно небольшими массивами данных.

Для каждого отдельного медикамента, внесенного пользователем в систему, указывается ID, MED, DATE, AMOUNT (идентификационный номер, название лекарства, срок годности и количество препарата соответственно), а механизм транзакций осуществляется с помощью методов в классе DatabaseHandler, который отвечает за работу с базой данных.

* + 1. Работа с календарем и оповещениями о приёме лекарств

За работу с календарем и созданием напоминаний о приёме медикаментов отвечают классы, расположенные внутри Java пакета calendar, а также используются DatePicker и TimePicker.

На данной странице расположен непосредственно календарь, а именно отдельная неделя определенного месяца и года (информация о рассматриваемом месяце и годе расположена над днями недели). С помощью кнопок «Назад» и «Вперед» можно переходить в предыдущую или следующую неделю соответственно. Ниже расположены непосредственно дни недели и даты. Календарь отображается с помощью CalendarViewHolder и CalendarAdapter, а для отображения конкретных дат используется класс CalendarUtils.

Модель определенного конкретного события представляет собой объект класса EventModel с полями name, date, time. Все созданные пользователем события хранятся в виде списка объектов eventsList.

При добавлении нового события запускается класс AddEventActivity. Список уже запланированных напоминаний о приеме хранится в виде запланированных событий внутри класса EventModel. Таким образом, AddEventActivity выступает в качестве посредника между действиями пользователя и списком eventList класса EventModel, а в качестве обработчика этого взаимодействия используется EventAdapter.

* + 1. Алгоритм создания напоминаний о приеме лекарства

Для создания нового напоминания требуется кликнуть на кнопку, расположенную в центре страницы с календарем. При клике открывается страница для создания напоминания, за работу которой отвечает класс AddEventActivity.

Есть несколько способов создания текста напоминания. Во-первых, можно вручную заполнить текст оповещения (так как для этого создан объект EditText), либо же можно обратиться к базе данных медикаментов (за взаимодействие с которой отвечает DatabaseHandler) для получения всех возможных названий внесенных в систему лекарств. Приватные поля типа DatePickerDialog отвечают за обработку информации о начале и окончании периода приема соответственно, а приватное поле типа LocalTime отвечает за хранение информации о времени напоминания о приеме.

Если все необходимые поля заполнены корректно, запускается цикл создания напоминаний о приеме медикаментов в рамках указанных дат. Для каждого из напоминаний сначала создается новое событие EventModel с указанными параметрами и добавляется в EventModel.eventList. Далее с помощью классов AlarmManager и ReminderBroadcast (базовым классом которого является BroadcastReceiver) новое уведомление планируется на установленные пользователем дату и время.

Описание и обоснование выбора метода организации входных и выходных данных

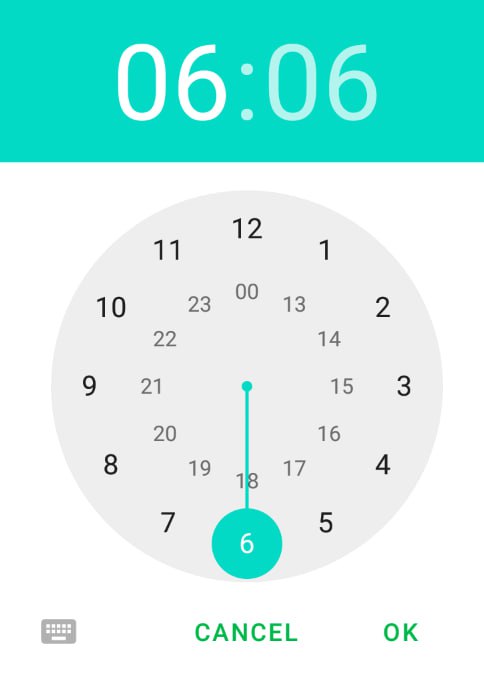
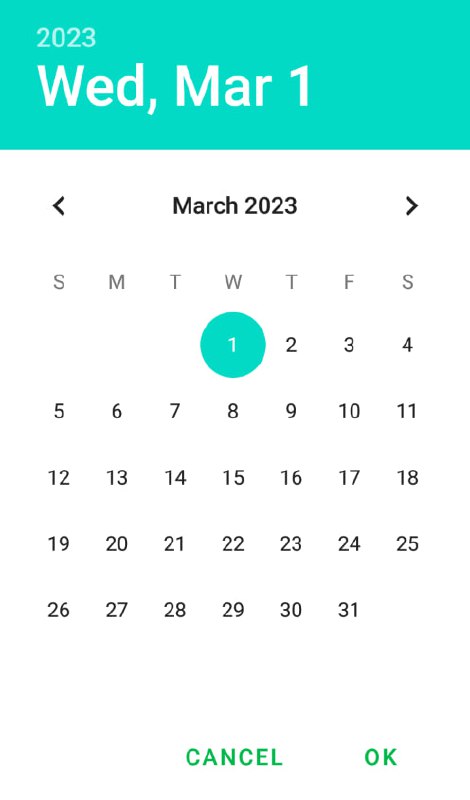
* + 1. Описание метода организации входных данных

1. Основной тип входных данных – это нажатия на сенсорные кнопки, так как этот тип взаимодействия наиболее прост и очевиден для использования. Помимо этого, при работе со списком всех медикаментов в разделе «Медикаменты» используются жесты смахивания. Так, смахивание вправо используется для изменения информации о лекарственном препарате, а жест смахивания влево запускает процесс удаления препарата из базы данных.
2. За добавление медикамента в базу данных отвечает класс AddNewMed, в котором описана основная логика этой операции. Пользователю предлагается заполнить несколько полей. Название препарата – последовательность символов UTF-8. Последовательность не должна состоять только из пробельных символов и не должна быть длиннее 100 символов. Выбор даты производится с помощью отдельного диалогового окна DatePicker. По умолчанию поле заполняется текущей датой. Также предлагается заполнить количество медикамента (в шт. или мл.), но количество не может быть отрицательным числом и не должно превышать 9999999. Для сохранения информации требуется нажать «Сохранить», а для выхода из диалогового окна без изменений требуется нажать «Закрыть». Аналогичный алгоритм используется для обновления информации о лекарственном препарате.   
   Медикаменты в списке могут повторяться (даже могут храниться полные копии).
3. Для добавления и удаления аптечек используются соответствующие кнопки в разделе «Аптечки». Название аптечки – последовательность символов UTF-8. Последовательность не должна состоять только из пробельных символов и не должна быть длиннее 100 символов. Кроме того названия аптечек не должны повторяться.  
   Для просмотра содержимого каждой из аптечек требуется нажать на стрелочку, расположенную слева от названия. При раскрытии списка медикаментов одной аптечки содержимое других аптечек автоматически скрывается. Для добавления лекарств в определенную аптечку требуется нажать на знак «+». При этом появляется выпадающий список, загруженный из пользовательской базы данных. Для удаления препарата из аптечки нужно нажать на значок справа от названия медикамента и подтвердить свой выбор во всплывающем окне.  
   Стоит заметить, что один и тот же препарат может храниться одновременно в разных аптечках. Количество различных аптечек не должно превышать 10.
4. При нажатии кнопки «Создать напоминание» появляется окно с необходимыми для заполнения полями.

Есть несколько способов составления текста оповещения. Во-первых, по нажатию кнопки «Найти лекарство в базе» появляется выпадающий список с названиями медикаментов из пользовательской базы данных. При выборе нужного препарата поле для текста автоматически заполняется выбранным названием. Либо же пользователь может ввести любую UTF-8 последовательность символов (даже пустую) для создания напоминания.  
Далее требуется выбрать даты начала и окончания (включительно) периода напоминаний. Программа за один раз может создать оповещение не более чем на 60 дней. При превышении лимита напоминание создается только на первые 60 дней. Для создания напоминания на один день дата начала и конца должна совпадать. При создании напоминания на уже прошедшую дату изменения отображаются в календаре, но push-уведомление не создается. Автоматически даты начала и окончания устанавливаются на текущую дату.

Еще требуется выбрать время приема лекарств (выбранное время следует вводить в двадцатичетырехчасовом формате). По умолчанию стоит текущее время.

При корректном создании напоминания и оповещения выводится соответствующее сообщение.

* + 1. Обоснование метода организации входных данных

1. Приоритетом является удобство пользователя, поэтому необходимо было создать максимально гибкое приложение с учетом особенностей применения. Так как у пользователя может быть несколько пачек медикаментов с идентичными характеристиками, разрешено, чтобы лекарства в базе данных полностью совпадали по названию, количеству и срокам годности, а контролировались и отображались в приложении с помощью уникального идентификационного номера. Также требовалось учесть, что названия медикаментов могут быть указаны на русском или английском языке и содержать символы, такие как +-!, поэтому в приложении поддерживаются символы UTF-8.
2. Аптечки не должны быть идентичными, поскольку необходимо дифференцировать их содержимое и назначение в зависимости от названия.
3. Оповещения должны были быть максимально гибкими и удобными для пользователя, поэтому было разработано 2 варианта ввода текста оповещения. Для максимальной гарантии корректности и удобства ввода дат и времени были использованы DatePicker и TimePicker соответственно, которые наиболее наглядны и просты для использования при вводе дат и времени соответственно.
   * 1. Описание метода организации выходных данных

При корректном заполнении необходимых полей обновляется хранилище данных, что можно заметить при обновлении нужной страницы (для обновления достаточно еще раз кликнуть на нужный раздел нижней панели меню). При некорректных данных выводится сообщение с указанием ошибок. Можно либо попробовать заполнить поля еще раз, либо закрыть окно. При простом закрытии окна изменений не происходит.

* + 1. Обоснование метода организации выходных данных

Описанный метод организации данных, будучи простым и интуитивным для обычного пользователя, полностью отвечает заявленным в техническом задании требованиям.

Описание состава и обоснование выбора технических и программных средств

* + 1. Описание состава технических и программных средств

Для стабильной работы приложения требуется следующий состав технических средств:

1. Планшетный компьютер, смартфон либо другое устройство, на котором установлена операционная система Android
2. Операционная система: Android 9.0 (Pie) и выше
3. API 28 и выше
4. Разрешение экрана не менее 1080 × 1920
5. Разрешение на отправку и получение уведомлений (для корректной работы механизма push-уведомлений о приеме лекарств)
6. Не менее 200 Мб свободного пространства
   * 1. Обоснование выбора технических и программных средств

Выбрана операционная система Android 9.0 (Pie) и выше, так как это позволяет использовать современные технологии и одновременно с этим доступно примерно на 81.2% девайсов (по информации, указанной в Android Studio).

Требования к разрешению обусловлены интерфейсом программы. При сильном отклонении параметров в большую или меньшую стороны интерфейс может перестать быть удобным и читаемым для пользователя.

Требования к памяти обусловлены тем, что для хранения базы данных медикаментов используется память устройства.

1. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

Предполагаемая потребность

Приложение может быть использовано пациентами с хроническими заболеваниями, такими как астма, диабет, аллергия, так как эти болезни зачастую подразумевают необходимость постоянного наличия соответствующих медицинских препаратов для быстрого реагирования в случае приступов. Приложение также может быть востребовано в семьях с детьми или людьми с ослабленным иммунитетом с целью контроля за наличием профилактических и лечебных препаратов для оперативной помощи при простудных и вирусных заболеваниях. Автомобилисты могут хранить и отслеживать информацию об автомобильных аптечках скорой помощи.

Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

Поиск в сети Интернет не выявил полных аналогов программы. Есть ряд программ с похожими функциями (Medisafe, Аптечка, Мои таблетки), их особенности и отличия от разрабатываемой программы описаны далее.

* + 1. Medisafe

Наиболее близкий аналог – приложение Medisafe. С помощью него можно отслеживать запас лекарств и добавлять напоминания о приеме. Преимущество приложения в том, что здесь можно кастомизировать добавляемые лекарства и формат уведомлений, изменять цвет иконок, добавлять инструкцию к каждому препарату. Тем не менее, приложение не отслеживает сроки годности лекарств.

* + 1. Аптечка

Приложение Аптечка создано для контроля за медикаментами пользователя – будь то домашняя, автомобильная, или аптечка против ожогов. Приложение позволяет хранить большой объем информации о лекарствах, в том числе: бренд, производителя, действующее вещество и фармакологическое действие. Также можно загрузить фото любого препарата. Недостаток приложения в том, что оно не предоставляет функции уведомлений о приеме лекарств и не отображает информацию на календаре.

* + 1. Мои таблетки

Основной функционал этого приложения – напоминание о приеме таблеток. Возможность просмотреть список всех купленных медикаментов с указанием их количества и сроков годности не предоставляется. Есть многие нестандартные параметры напоминаний, такие как возможность добавления перерывов в курс приема таблеток, установка количества курсов.

* + 1. Сравнительная таблица

Таблица – Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Параметр сравнения | Medisafe | Аптечка | Мои таблетки | Разработанное приложение |
| Визуализация списка всех лекарств в аптечке | + | + | - | + |
| Отображение и контроль за сроками годности лекарств | - | + | - | + |
| Календарь с информацией о приеме лекарств | + | - | + | + |
| Напоминания о приеме лекарств | + | - | + | + |
| Возможность указания дозировки | + | + | - | + |
| Отсутствие рекламы | + | + | - | + |

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. ГОСТ 19.103–77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.104–78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.105–78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.106–78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.301–79 Программа и методика испытаний. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.603–78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.604–78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. Умаров С.З., Павленко Н.И. АНАЛИЗ ВИДОВ И СОДЕРЖАНИЯ МЕДИЦИНСКИХ АПТЕЧЕК ПЕРВОЙ ПОМОЩИ // Здоровье и образование в XXI веке. 2020. №9. URL: https://cyberleninka.ru/article/n/analiz-vidov-i-soderzhaniya-meditsinskih-aptechek-pervoy-pomoschi (дата обращения: 11.01.2023).
9. Gan S. K. E. et al. An overview of clinically and healthcare related apps in Google and Apple app stores: connecting patients, drugs, and clinicians //Scientific phone apps and mobile devices. – 2016. – Т. 2. – №. 1. – С. 1-9.
10. Ткачук Е.О. ПРИМЕНЕНИЕ СУБД SQLITE В ОБРАЗОВАТЕЛЬНОМ ПРОЦЕССЕ СОВРЕМЕННОГО ВУЗА // НАУЧНЫЕ ИССЛЕДОВАНИЯ ВЫСШЕЙ ШКОЛЫ. – 2020. – С. 12-15.

Приложение 1

ТЕРМИНОЛОГИЯ

|  |  |
| --- | --- |
| Push-уведомление | Кликабельные всплывающие сообщения, которые появляются в области уведомлений. |
| Java пакет | Способ объединить группу классов или интерфейсов. |
| SQLite | Система управления базами данных, отличительной особенностью которой является ее встраиваемость в приложения. |
| Транзакция | Единицы работы или последовательности действий, выполненных в логическом порядке с помощью какой-либо программы базы данных. |
| Жест смахивания | Это когда вы кладете палец (пальцы) на сенсорную панель и перетаскиваете их вверх, вниз, влево или вправо. Каждое направление, в котором вы проводите пальцем, выполняет разные действия. |
| UTF-8 | Стандарт кодирования символов, позволяющий более компактно хранить и передавать символы Юникода, используя переменное количество байт (от 1 до 4). |

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц в докум.) | № документа | Входящий № сопроводительного докум. и дата | Подп. | Дата |
| Изм. | Измененных | Замененных | Новых | Аннулированых | 21 |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |