Научная статья
УДК 004.415.2
ББК 32.972
А 16
DOI: 10.53598/2410-3225-2022-4-311-71-80

Проектирование актуального решения для планирования задач на базе операционной системы Андроид

(*Рецензирована*)

Оксана Федоровна Абрамова, Илья Дмитриевич Семилетов

Волжский политехнический институт (филиал) Волгоградского государственного технического университета, *Волжский, Россия*, *oxabra@yandex.ru*

Аннотация. *Актуальность работы обеспечивается высокой практической ценностью мобильных приложений для планирования: разработка их более совершенных аналогов позволяет открывать для пользователей расширенные возможности по осуществлению планирования и, как следствие – улучшения качества своей жизни. В данной статье представлены этапы проектирования современного мобильного приложения для планирования. Приведены результаты исследования и анализа программных аналогов в области планирования и предлагаются новые функциональные возможности с учетом выявленных проблем в качестве улучшения. Обосновывается выбор инструментальных средств для реализации мобильного приложения. Приводится описание архитектуры приложения: как связь используемых паттернов проектирования, так и внедренных сторонних компонентов. Авторы описывают этапы проектирования базы данных программы и окна календаря как со стороны реализации ранее выбранной архитектуры, так и со стороны интерфейса. Дополнительно представлена модель алгоритма генерации уведомлений с учетом выбранных сроков выполнения задачи, ее важности и общей загруженности пользователя. В качестве основных методов исследования выбраны анализ и моделирование. Результаты исследования могут быть использованы при проектировании и реализации мобильного приложения для планирования задач.*

Ключевые слова: *планирование, андроид, проектирование, анализ, пользователь, базы данных, интерфейс, моделирование*

Original Research Paper

Design of an actual task planning solution based on the Android operating system

Oksana F. Abramova, Ilya D. Semiletov

Volzhsky Polytechnical Institute, *Branch of the Volgograd State Technical University*, *Volzhskiy*, *Russia*, *oxabra@yandex.ru*

Abstract. *This article discusses the design of a modern mobile application for planning. The relevance of the work lies in the high practical value of mobile applications for planning. By developing their more advanced counterparts, users have more opportunities to carry out planning, and therefore improve the quality of their lives. The article also reviews existing application functionality and suggests new ones as an improvement. The publication substantiates the choice of certain tools for implementation and provides the design of the application architecture, both the connection between the design patterns used and the implemented third-party components. The authors describe the design of the program database and the calendar window, both from the point of view of the implementation of the previously laid architecture, and from the interface. Additionally, the article presents a model of the notification generation algorithm, taking into account the selected deadlines for completing the task, its importance and the overall workload of the user. The main research methods are analysis and modeling. We can use the results of the study in the design and implementation of a mobile application for planning.*

Keywords: *planning, android, design, analysis, user, database, interface, modeling*

Введение

Для современного активного и работающего человека планирование задач является неотъемлемой частью жизни. Количество целей и острая необходимость предварительного планирования этапов их достижения обуславливают важность использования различных программных решений в этой области, что позволяет эффективнее реализовать отдельные задачи и распределять собственные временные, энергетические, материальные и другие ресурсы. Деятельность энергичного члена современного общества без многоэтапного приоритетного предварительного планирования грозит большими проблемами как в работе, так и в личной жизни.

Время для современного человека стало одним из важнейших ресурсов. Нехватка его, возникающая в том числе в связи с игнорированием планомерности, приводит к еще большему затягиванию человека в пучину долгов и стресса. Для избегания таких ситуаций был придуман тайм-менеджмент. В нашей стране в 0-е годы XX века этот подход определялся как научная организация труда (НОТ) и был описан в Центральном институте труда, директор которого А.К. Гастев развивал идеи о том, что эффективность организации начинается с личной эффективности, в частности эффективного использования времени. Общественное движение борьбы за время привело к созданию Лиги «Время», просуществовавшей до 1926 года. В первую очередь это произошло вследствие того, что все рациональное было превращено в фарс [1]. Основоположником понятия «тайм-менеджмент» считается предприниматель из Дании К. Меллер. В 1975 году он основал одноименную компанию, создал прототип органайзера (уникальный блокнот) и организовал первые тренинги по его заполнению. С историей тайм-менеджмента можно познакомиться в работе [2], см. также [3]. В настоящее время тайм-менеджмент – это способ организации времени, а также планирование и контроль выполнения задач для повышения эффективности использования временных ресурсов.

Планирование – неотъемлемая часть жизни каждого человека. Мы пользуемся им неосознанно, строя планы на ходу, или осознанно, с использованием специальных инструментов. Было неоднократно доказано: планирование оказывает прямое влияние на качество жизни человека как в долгосрочном смысле, так и в краткосрочном. Доступность эффективных инструментов для планирования позволит сделать процесс более эффективным и приведет к его популяризации среди тех масс населения, которые не используют данную технику.

На данный момент одной из наиболее популярных платформ для планирования является смартфон на базе операционной системы Андроид. Для этой платформы разработано целое множество различных приложений для планирования. Проведя анализ нескольких таких разработок, делается вывод, что существует возможность разработки аналогичного приложения, но уже с устраненными недостатками. Разработав такое решение, предполагается повысить эффективность планирования и улучшить качество жизни людей, использующих мобильные приложения для планирования.

Выбор основных инструментов для разработки

В качестве языка для разработки предлагается использовать Java. Java – один из популярнейших языков программирования на данный момент. Для него уже написаны десятки, если не сотни тысяч, руководств и алгоритмов на самые разные случаи его использования.

В качестве целевой операционной системы для работы приложения предлагается выбрать ОС Андроид. Андроид – самая популярная на данный момент операционная система не только среди мобильных устройств, но и в целом. Это гарантирует, что охват аудитории приложением будет широчайшим по сравнению с другими вариантами.

В качестве среды разработки можно использовать IDE “Android Studio”. Это ос-

новная среда разработки, рекомендуемая самим разработчиком операционной системы. Другое ее преимущество заключается в том, что она основана на другой популярной среде разработки “IntelliJ IDEA”, благодаря чему Android Studio может использовать тысячи уже существующих плагинов для “IntelliJ IDEA”, которыми воспользовались миллионы или даже десятки миллионов программистов по всему миру.

Определение архитектуры приложения

На сегодняшний день условия рынка программ таковы, что возможность софта быть расположенным к изменениям и тестированию жизненно важна на фоне большой конкуренции. В связи с этим предлагается разработать архитектуру, позволяющую в полной мере реализовать данные особенности.

В качестве основы для проектируемой архитектуры предлагается использовать MVVM паттерн проектирования. Данное решение активно поддерживается разработчиком операционной системы на уровне выпуска библиотек для его реализации, а также рекомендаций и инструкций к разработке архитектуры мобильных приложений. Основным постулатом продвигаемой архитектуры является то, что она состоит из следующих независимых слоев:

- Model – моделирует бизнес-логику приложения;
- View – интерфейс взаимодействия с пользователем;
- ViewModel – логика приложения, расчеты и др.

Взаимодействие же между смежными слоями предлагается реализовывать с помощью обсерверов (наблюдателей).

Используемые компоненты

Для организации работы с базой данных можно использовать библиотеку “Room” из набора библиотек “Android Jetpack”, рекомендуемых разработчиком ОС Андроид. Среди преимуществ работы с данной оболочкой можно выделить облегчение работы с построением базы данных (БД) за счет генерации таблиц на основе Java классов, проверку SQL запросов во время компиляции, удобны аннотации для уменьшения ошибок и повторяющегося кода.

Для реализации взаимодействия между View и ViewModel предлагается использовать компонент “LiveData” (тоже из Jetpack). Данный компонент используется для реализации паттерна наблюдатель (Observer) и позволяет уведомить View слой, когда запрос к базе данных выполнен и данные поступили в ViewModel для обработки. К его плюсам по сравнению с RxJava можно отнести то, что данный компонент не уведомляет те наблюдатели, которые не находятся в активном состоянии жизненного цикла, например – фрагмент или активити свернуто, и повышает быстродействие и безопасность. В связи с особенностями работы компонента его удобно использовать с одним объектом базы данных во всем приложении. Это может быть достигнуто с использованием паттерна «синглтон» или «одиночка». Данный паттерн предлагается реализовать как расширение класса “App” и для хранения в нем объектов базы данных, а также класса настроек.

Другим полезным компонентом для реализации взаимодействия между ViewModel и View может стать библиотека Data Binding из того же Android Jetpack. Она позволяет выполнять привязку отдельного класса к интерфейсу и отслеживать его изменения. Данная библиотека избавляет от рутинной работы по поиску нужного объекта в интерфейсе и передаче ему данных каждый раз, когда данные изменяются. Это улучшает читаемость кода и его безопасность.

Также не стоит забывать и о быстродействии. Полезное действие на данный аспект может оказать RecyclerView из Android Jetpack. Он использует паттерн “ViewHolder” – создание специального класса хранения ресурсных данных для того,

чтобы избежать тяжелых вызовов к медленной памяти во время работы пользователя с данным компонентом. Среди других его плюсов можно отметить наличие встроенных анимаций, возможность использовать его как сетку, поддержку декорирования элементов и др. К минусам можно отнести сложность в понимании его работы, необходимость описывать (обязательно) класс-хранилище вида (“ViewHolder”), громоздкость.

Обобщая вышеизложенное, предлагается реализовать архитектуру, изображенную на рисунке 1. Здесь роль View слоя играет компонент под номером 1, роль ViewModel – компонент под номером 2, роль Model играют компоненты под номерами 4, 5 и 6. Компонент под номером 3 играет роль доступа к объекту базы данных (уведомления LiveData работают в пределах одного объекта БД) и хранилищу настроек приложения.

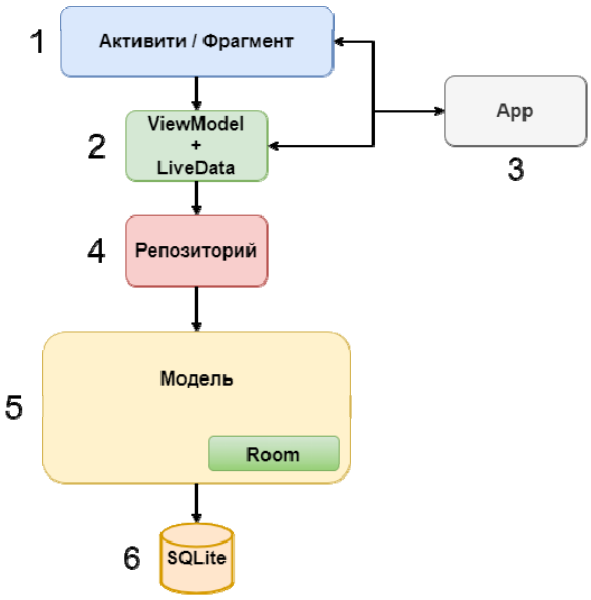

Рис. 1. Архитектура приложения

Fig. 1. Application architecture

Под номером 3 здесь изображен компонент под названием «Репозиторий». Он реализует паттерн Репозиторий (Repository) и является интерфейсом доступа к данным. В данном случае хранилищем данных выступает только база данных, но приложение может с легкостью быть расширено удаленным хранилищем (например, синхронизацией с Google Календарем). Тогда все, что нужно сделать, это выполнить изменения внутри компонента «Репозиторий», при этом вся остальная часть приложения останется нетронутой. В этом и заключается достоинство использования данной архитектуры.

Определение основных функциональных возможностей приложения

Изучив несколько популярных приложений для планирования, а также основные паттерны поведения пользователей [4, 5], были определены основные варианты использования, они представлены на рисунке 2. Для моделирования был использован язык UML [6].

Помимо основных вариантов использования, таких как создание категорий, задач, проектов, их просмотр, установка уведомлений и синхронизация со сторонним сервисом, добавлены также дополнительные варианты использования, повышающие качество и эффективность планирования. Например, алгоритм расчета уведомлений на основе загруженности пользователя, существенно снижающий трудоемкость планирования при использовании генерируемых системой уведомлений. Это будет хорошо заметно в случаях, когда пользователь имеет плотный график из задач с разным приоритетом.

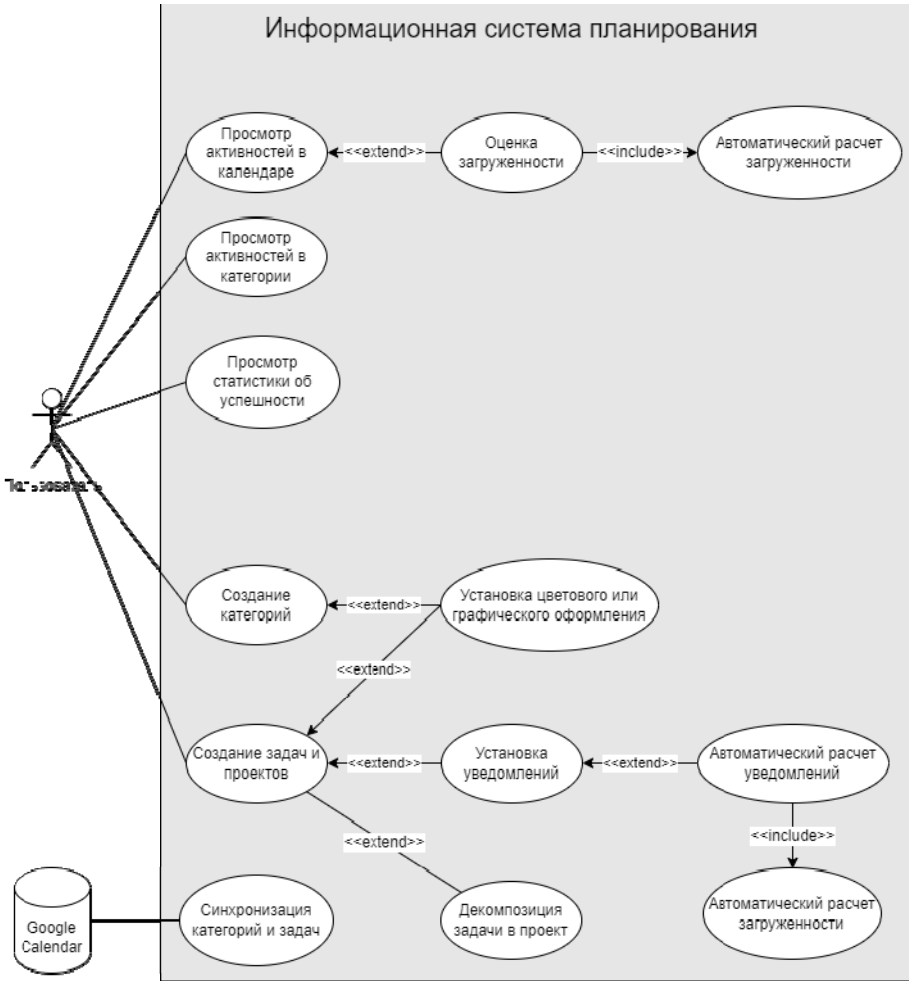

Рис. 2. Диаграмма вариантов использования проектируемой системы

Fig. 2. Diagram of use cases of the designed system

Автоматический расчет загруженности для просмотра из календаря позволит также уменьшить трудоемкость работы с программой. Вариант использования, связанный с кастомизацией внешнего вида задачи, проекта или категории, позволит как уменьшить время поиска задачи в списке, так и увеличить привлекательность программы для самого пользователя.

Проектирование базы данных

Для хранения пользовательских заметок – задач и проектов – потребуется локальное хранилище. Организовать хранилище можно с учетом нескольких подходов: разработать его самому или воспользоваться уже существующим. Если критерии, предъявляемые к хранилищу, не являются особенно специфическими, то второй вариант более предпочтителен, так как предлагает уже протестированную технологию для работы, что принесет меньшее количество возможных ошибок. Таким вариантом является SQL – подобная база данных для операционной системы Андроид – SQLite. Предложенная БД не требует установки дополнительных зависимостей, так как является встроенной. Другим ее преимуществом является тот факт, что она запускается вместе с работой приложения или других ее частей автоматически и обмен данными идет через подключения, так как сама БД является частью программы, используя при этом удобный и хорошо знакомый SQL синтаксис [7, 8].

Логическая схема базы данных изображена на рисунке 3.

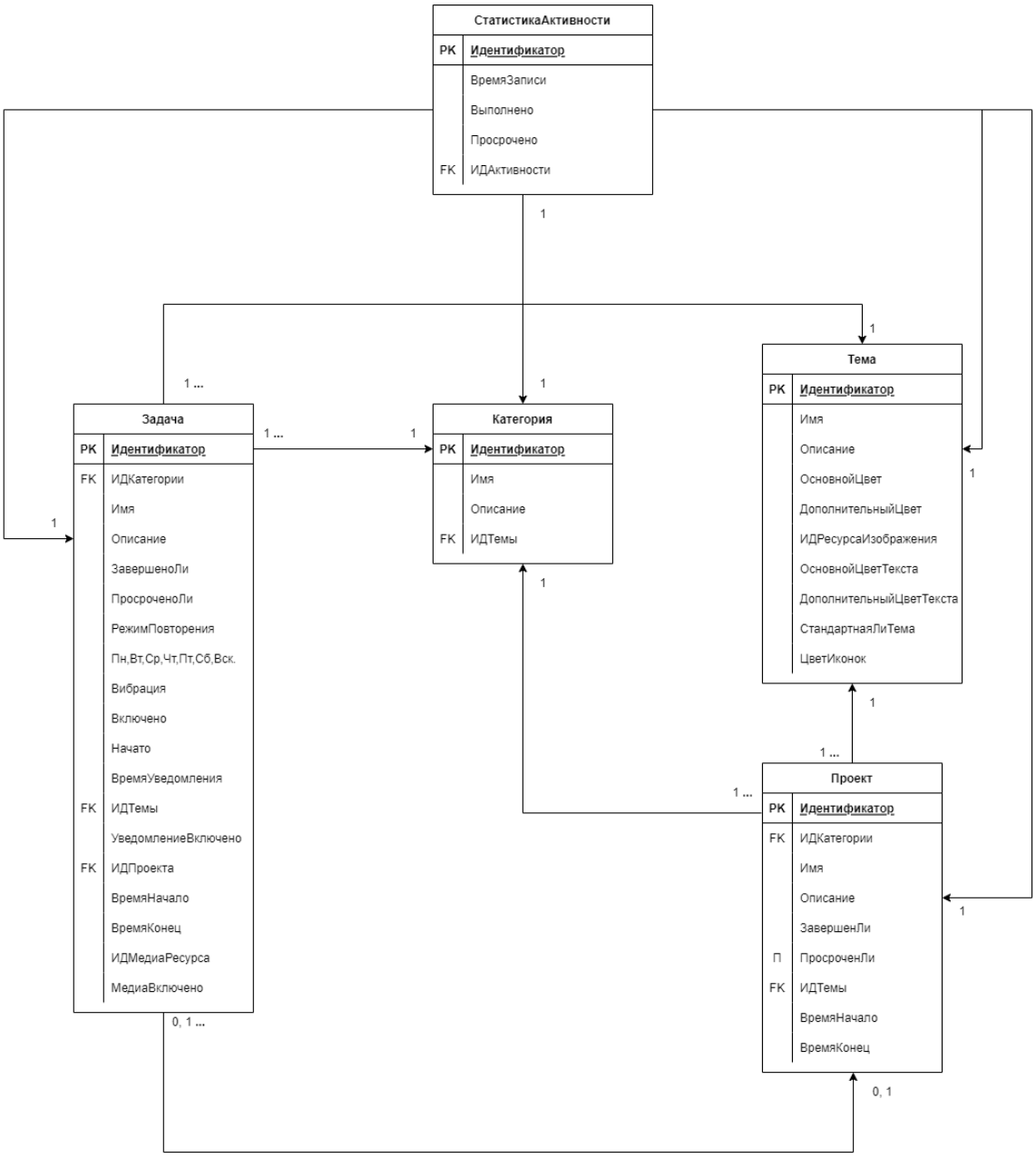

Рис. 3. Логическая схема базы данных

Fig. 3. Database Logic

Для покрытия нужд предметной области и реализации планируемого функционала потребуется 5 таблиц.

Таблица Категория

Используется для хранения данных соответствующих объектов. Они моделируют понятие «списка» для задач и проектов.

Таблица Задача

Используется для хранения данных соответствующих объектов. Они моделируют заметки пользователя, могут иметь уведомление.

Таблица Проект

Используется для хранения данных соответствующих объектов. Моделируют декомпозицию объемных задач. Играют роль своего рода оболочки, по-другому – выполняют группировку некоторого количества задач.

Таблица Тема

Используется для хранения данных соответствующих объектов. Применяются для хранения внешнего оформления задач, проектов и категорий.

Таблица Статистика Активности

Используется для построения моделей статистики и хранения изменений в задачах и проектах, используемых для этого.

Таким образом, построенная база данных полностью отражает предметную область и соответствует намеченным ранее функциональным требованиям разрабатываемого программного обеспечения.

Генерация уведомлений

Одним из преимуществ использования смартфона как инструмента для планирования является функция уведомлений, которая всегда находится рядом и в нужный момент напомнит об отложенных задачах [9].

Реализация алгоритма расчета уведомлений на основе приоритета задач и загруженности пользователя может существенно увеличить удобство использования приложения. Приоритет задач предлагается определять в виде матрицы 2x2, по аналогии с матрицей Эйзенхауэра.

Предварительно к началу работы алгоритма следует определить временной интервал для активных задач (время выполнения которых уже началось), а также задач, время которых еще не пришло, но они также входят в общий интервал как отрезки. Далее, сформировать массив дней, соответствующий общему интервалу для отслеживания загруженности. После чего исполнить алгоритм, изображенный на рисунке 4.

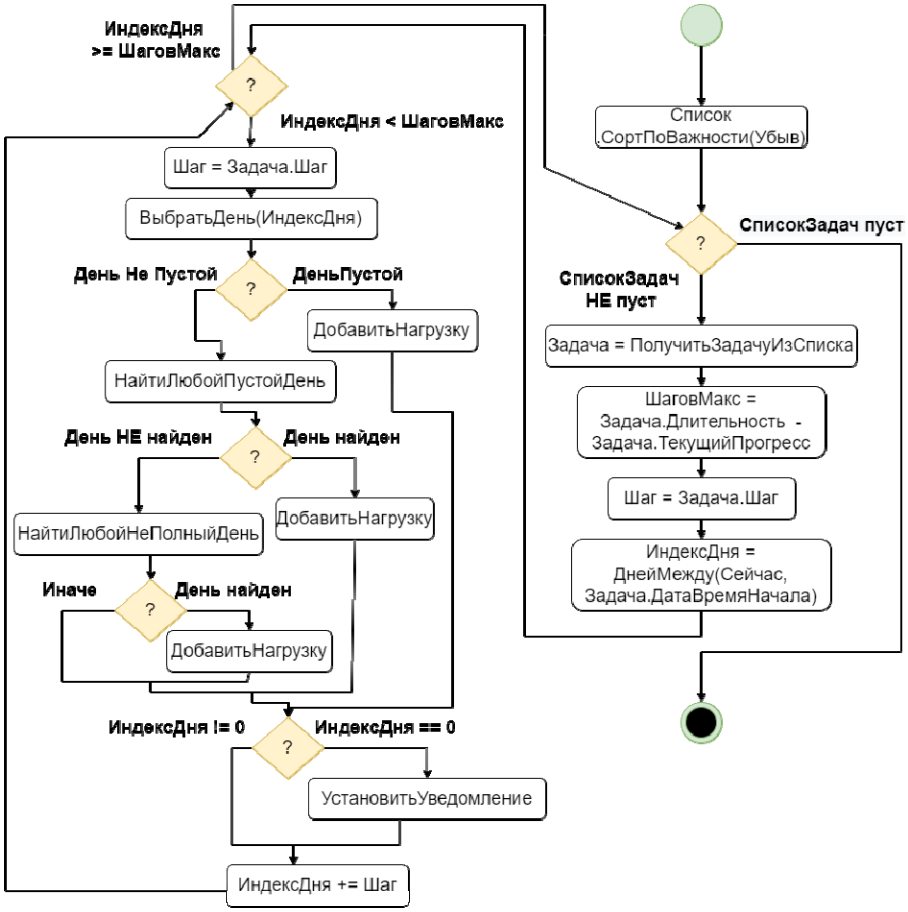

Рис. 4. Алгоритм расчета уведомлений

Fig. 4. Algorithm for calculating notifications

Алгоритм состоит из двух циклов – внешнего и внутреннего. Во внешнем цикле происходит перебор задач (с заданным пользователем интервалом времени выполнения задачи) и определение количества уведомлений для каждой из них; во внутреннем цикле осуществляется определение, поиск подходящего дня и времени для каждого уведомления на заданном временном интервале и установка уведомления.

Таким образом, с помощью реализации данного алгоритма можно добиться приоритетного распределения задач пользователя, сведя нагрузку на него к минимуму.

Календарь

Еще одним важным инструментом планирования является календарь, и упускать возможность его использования было бы ошибкой. В качестве улучшения для него можно предложить функцию отображения нагрузки для конкретного дня. Нагрузку предлагается определять по количеству задач, определенных на день, или количеству занятых часов. Получать значение, с которым следует выполнять сравнение для вычисления нагрузки, можно несколькими способами:

- статически – определить значение константно во время программирования приложения. Например, до 2 задач – малая нагрузка, от 2 до 4 – средняя и т.д.;
- динамически – вычислить значение во время выполнения программы, определив среди задач на месяц (или другой период) минимально загруженные дни и максимально загруженные, далее рассчитать искомый коэффициент относительно них.

Дизайн такого календаря может иметь вид, изображенный на рисунке 5.

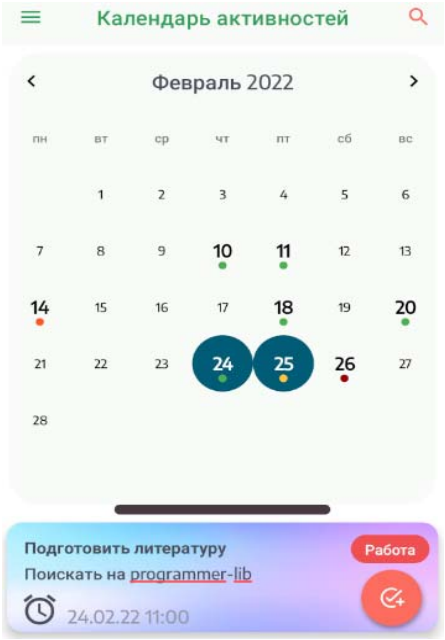

Рис. 5. Внешний вид окна календаря

Fig. 5. Appearance of the calendar window

Как видно на макете, загруженность пользователя на конкретный день отображается с помощью цветного маркера под датой. Помимо этого, здесь также можно выбрать временной отрезок и просмотреть относящиеся к нему задачи.

Модель реализации рассмотренного функционала изображена на рисунке 6.

На диаграмме все объекты, отмеченные голубым цветом, ассоциируются с началом деятельности пользователя – как с открытием данного окна, так и с окончанием действия (например, переименовывания задачи) и циклическим начинанием нового.

Помимо вышеизложенного, здесь реализованы такие функции, как поиск задачи среди отображенного списка задач, предотвращение случайного удаления – как в виде

диалогового окна, так и в виде SnackBar, появляющегося после удаления (на 2 секунды) и позволяющего «откатить» изменения.

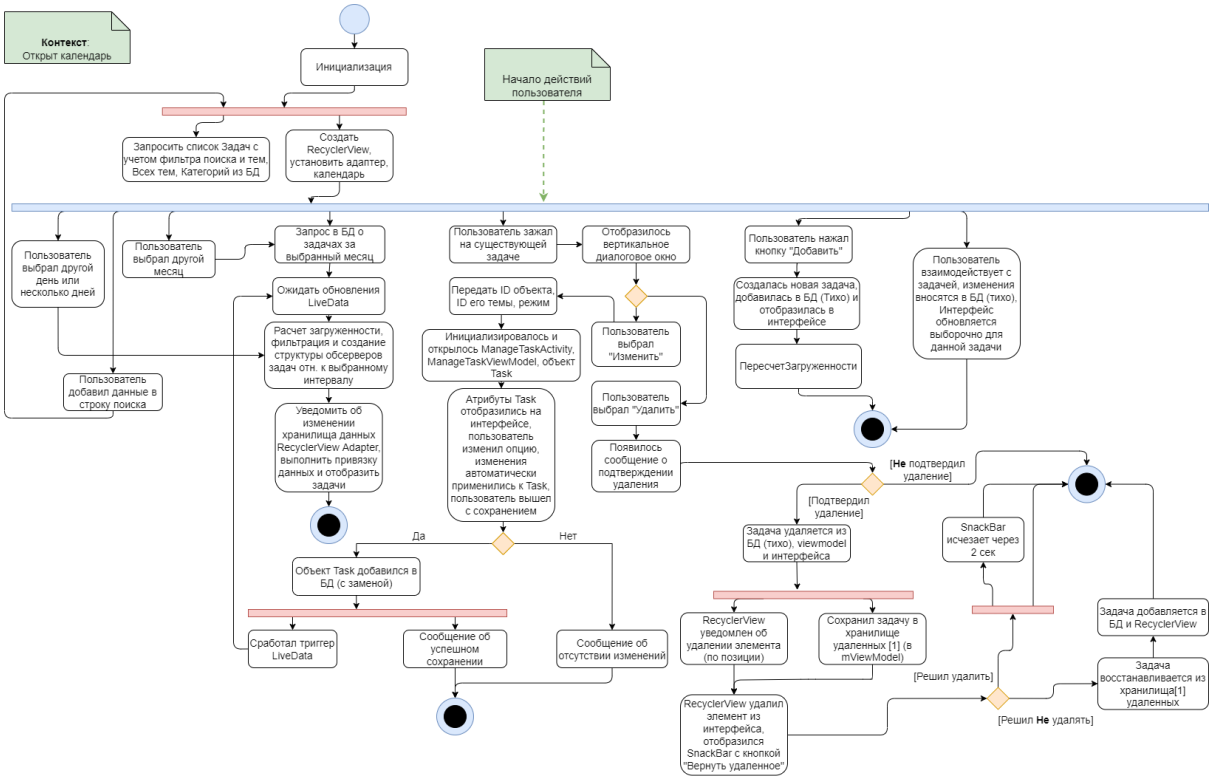

Рис. 6. UML диаграмма деятельности окна календаря

Fig. 6. UML calendar window activity chart

Также в предлагаемом решении максимально реализована поддержка общепринятых жестов. Например, выполнив свайп вправо на задаче, ее можно удалить, выполнив свайп влево – попасть в экран редактирования уведомления.

Выводы

На основе предположений о разработке приложения для планирования была проведена проектировочная деятельность, а именно:

- проведено исследование и осуществлен выбор инструментов для разработки;
- определены основные требования к приложению для планирования;
- спроектирована база данных на основе потребностей предметной области;
- выполнено проектирование архитектуры приложения как с учетом современных потребностей, так и с учетом потребностей предметной области;
- разработан алгоритм уведомлений, позволяющий выполнить автоматизацию действий пользователя по распределению задач и определению нагрузки;
- осуществлен вариант проектирования окна для разработки системы на основе предметной области как с точки зрения интерфейса, так и с точки зрения логических механизмов.

Представленные в данном исследовании решения можно использовать для практической реализации программного средства.

Примечания

1. Управление временем. URL: https://ru.wikipedia.org/wiki/Управление_временем
2. Матвиевский А.А. История тайм-менеджмента // Молодой ученый. 2021. № 32 (374). С. 34–35. URL: https://moluch.ru/archive/374/83537/

3. Тайм-менеджмент. URL: https://webnewsite.ru/rastyanutoe-vremya-kto-pridumal-tajm-menedzhment/

4. Абрамова О.Ф. Визуализация паттерна поведения пользователя web-системы // Кибернетика и программирование: электронный журнал. 2019. № 3. С. 43–52. URL: https://nbpublish.com/library_read_article.php?id=23017. DOI: 10.25136/2644-5522.2019.3.23017

5. Мельниченко Д.В., Абрамова О.Ф. Исследование логических проблем юзабилити сайтов и анализ существующих решений // Современная техника и технологии. 2015. № 1. С. 3–12. URL: http://technology.snauka.ru/2015/01/5360

6. Абрамова О.Ф. CASE-технологии: изучать или исключить? // Alma Mater (Вестник высшей школы). 2012. № 9. С. 109–110.

7. Рыбанов А.А., Силаев А.А. Метод адаптации реляционных баз данных к реализации нечетких запросов // Южно-Сибирский научный вестник. 2022. № 4 (44). С. 74–80. URL: http://s-sibsb.ru/images/articles/2022/4/S-SibSBIssue44.pdf

8. Математическая модель динамики роста реляционной базы данных / А.А. Рыбанов, О.В. Свиридова, Е.М. Филиппова, А.Ю. Александрина, О.Ф. Абрамова // Вестник Адыгейского государственного университета. Сер.: Естественно-математические и технические науки. 2020. Вып. 4 (271). С. 53–59. URL: http://vestnik.adygnet.ru

9. Семилетов И.Д., Абрамова О.Ф. Планирование с помощью смартфона: проблемы и пути решения // Студенческий вестник. 2022. № 17–10 (209). С. 44–45. URL: https://studvestnik.ru/journal/stud/herald/209

References

1. Time management. URL: https://ru.wikipedia.org/wiki/Управление_временем

2. Matvievsky A.A. History of time management // Young Scientist. 2021. No. 32 (374). P. 34–35. URL: https://moluch.ru/archive/374/83537/

3. Time management. URL: https://webnewsite.ru/rastyanutoe-vremya-kto-pridumal-tajm-management/

4. Abramova O.F. Visualization of the web-system user behavior pattern // Cybernetics and Programming: Electronic Journal. 2019. No. 3. P. 43–52. URL: https://nbpublish.com/library_read_article.php?id=23017 DOI: 10.25136/2644-5522.2019.3.23017

5. Melnichenko D.V., Abramova O.F. Study of logical problems of website usability and analysis of existing solutions // Modern Technique and Technologies. 2015. No. 1. P. 3–12. URL: http://technology.snauka.ru/2015/01/5360

6. Abramova O.F. CASE-technologies: study or exclude? // Alma Mater (Bulletin of higher education). 2012. No. 9. P. 109–110.

7. Rybanov A.A., Silaev A.A. Method adaption relational databases to fuzzy queries // South Siberian Scientific Bulletin. 2022. No. 4 (44). P. 74–80. URL: http://s-sibsb.ru/images/articles/2022/4/S-SibSBIssue44.pdf

8, Mathematical model of growth dynamics for relational database / A.A. Rybanov, O.V. Sviridova, E.M. Filippova, A.Yu. Aleksandrina, O.F. Abramova // The Bulletin of the Adyghe State University. Ser.: Natural-Mathematical and Technical Sciences. 2020. Iss. 4 (271). P. 53–59. URL: http://vestnik.adygnet.ru

9. Semiletov I.D., Abramova O.F. Planning with a smartphone: problems and solutions // Student Bulletin. 2022. No. 17-10 (209). P. 44–45. URL: https://studvestnik.ru/journal/stud/herald/209

Авторы заявляют об отсутствии конфликта интересов.

The authors declare no conflicts of interests.

Статья поступила в редакцию 01.11.2022; *одобрена после рецензирования 29.11.2022*; *принята к публикации 30.11.2022*.

The article was submitted 01.11.2022; *approved after reviewing 29.11.2022*; *accepted for publication 30.11.2022*.