Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

Уфимский государственный нефтяной технический университет

Кафедра вычислительной техники и инженерной кибернетики

КУРСОВАЯ РАБОТА

по дисциплине

Мобильные приложения и программирование устройств

WISHER APP – ГЕНЕРАТОР ПОЗДРАВЛЕНИЙ

Выполнил студент группы БПО(и) 18-01 К.Р. Мамбетов

Принял ст. преподаватель Е.В. Дружинская

Дата представления работы: 25.05.2021

Дата защиты: 25.01.2021 Результат: \_\_\_\_\_\_\_\_\_\_

Уфа, 2020 г.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc72793592)

[1 Обзор предметной области 4](#_Toc72793593)

[2 Проектирование приложения 6](#_Toc72793594)

[2.1 Функциональное моделирование проекта 6](#_Toc72793595)

[2.2 Эскизирование экранов приложения 7](#_Toc72793596)

[2.3 Логическое моделирование проекта 9](#_Toc72793597)

[3 Программная реализация 10](#_Toc72793598)

[3.1 Блоки и поздравления 10](#_Toc72793599)

[3.2 Классы-менеджеры – WishManager и WishBlockManager 11](#_Toc72793600)

[3.3 Общение с удаленной БД – класс FireBaseDataAgent и технология Firebase 12](#_Toc72793601)

[3.4 Адаптеры для списков блоков и поздравлений – классы WishRecyclerAdapter и WishBlockRecyclerAdapter. 13](#_Toc72793602)

[3.5 Основной экран (поздравления и блоки) 14](#_Toc72793603)

[3.6 Генерация и редактирование блоков (экран создать блок). Редактирование поздравлений 16](#_Toc72793604)

[3.7 Генерация поздравлений (экраны настройки поздравления и выбора блоков) 17](#_Toc72793605)

[3.8 Заимствованные элементы. 18](#_Toc72793606)

[ЗАКЛЮЧЕНИЕ 19](#_Toc72793607)

[Приложение 1. Скриншоты эскизов экранов приложения 25](#_Toc72793608)

# ВВЕДЕНИЕ

Тема поздравлений была актуальна с тех пор, как люди начали праздновать дни рождения, религиозные и другие праздники, а именно с появления у человечества понятия времени и календаря. М.М. Бахтин, русский философ, культуролог, литературовед, теоретик европейской культуры и искусства отмечает, что «…празднества на всех этапах своего исторического развития были связаны с крупными, переломными моментами в жизни природы, общества и человека». По поводу празднеств принято делать дни нерабочими и организовывать мероприятия, в которых люди поздравляют друг друга и говорят теплые слова в зависимости от конкретного праздника.

Однако, такие повторяющиеся праздники, как дни рождения, делают поздравления однообразными и повторяющимися, так как человеческий мозг стремится к упрощению часто встречающихся задач. Приложение, которое разрабатывается в данной работе призвано помочь с составлением поздравлений под нужды пользователя, причём делать поздравление более разнообразным.

Приложение, в отличии от альтернативных способов поздравления через Интернет имеет функцию составления поздравлений из блоков, которые можно написать самим, либо использовать блоки, которые другие пользователи сделали публичными. Также будет доступна функция генерации короткого поздравления случайным образом, что позволит пользователям создавать поздравление быстро и с большой вариативностью

Данный проект не имеет возрастного ограничения, контент будет модерироваться администраторами приложения.

Основная цель данного проекта – дать пользователям простой и универсальный способ создавать поздравления, разнообразить придумывание поздравлений. Для этого необходимо решить две задачи. Первая задача: понять, каким образом строятся поздравления, выработать алгоритм для построения поздравлений. Вторая задача: разобрать аналогичные по тематике продукты, понять, каким образом в них генерируются поздравления и применить эти знания для создания приложения.

# Обзор предметной области

Сегодня способы поздравлений можно разделить на две группы: оффлайн (поздравление на мероприятии, которое осуществляется человеком вербально и невербально) и онлайн поздравления (посредством сети Интернет, а также других видов связи). В рамках данной работы мы будем рассматривать именно онлайн способы поздравлений.

Онлайн поздравления осуществляются путём отправки текстовых сообщений, фото- и видео-поздравлений. Самый простой способ в плане действия, но сложный с точки зрения составления поздравления - написать имениннику то, что в ином случае может быть сказано на самом празднике.

С популярностью социальных сетей и мессенджеров пришла популярность другому способу поздравления - заранее созданные другими пользователями картинки, GIF- анимации и видео-слайд шоу поздравления. Для пользователей, свободно владеющих редакторами изображений (Photoshop, GIMP и др.) это не составляет большого труда.



Рис. 1 – пример поздравления картинкой

При этом пользователи, получившие поздравление, могут их отправить другим именинникам, именно за счет пересылок данный способ и стал популярным. Также этот тип поздравления вдохновил энтузиастов на создание приложений-каталогов, содержащих поздравительные картинки в достаточных для вариативности количествах.

Основным предметом исследования стали веб- и мобильные приложения для создания и отправки поздравлений. Существуют как упомянутые выше приложения каталоги с предзаписанными поздравлениями на все случаи жизни, так и простые текстовые генераторы для определенных праздников. Однако среди них не было найдено тех приложений, которые используют составные части для создания поздравлений.



Рис.2 – страница приложения с поздравлениями в Google Play

То, что предлагает разрабатываемое приложение – создание поздравлений из готовых блоков, которые создают сами пользователи, с возможностью публикации блоков для того, чтобы их смогли использовать другие пользователи приложения. Блоки могут содержать в себе место для имени, которое будет впоследствии заменено на имя именинника.

# Проектирование приложения

## Функциональное моделирование проекта

Пользователь имеет возможность генерировать поздравления из различных частей – блоков. Блоки содержат часть текста поздравления, а также список тегов, по которым пользователь может искать для себя нужные блоки (в основном в тегах размещаются качества именинника и название праздника).

Блоки делятся на начальные, серединные и конечные. Это сделано для того, чтобы поздравление было наиболее приближенным к реальному поздравлению. Начальные блоки открывают повествование, серединные перечисляют разные пожелания, а конечные подытоживают все поздравление.

Случайная генерация создает поздравление из всех трех типов блоков, беря из каждого по одному блоку. Таким образом создастся одно короткое, но верное по смыслу поздравление.

Список предоставляемых функций:

* Генерация поздравления. Сначала пользователь вводит имя именинника, а также список качеств-тегов, которыми обладает именинник. Далее пользователю предоставляется выбор: сгенерировать поздравление случайно из полученных блоков (которые были получены исходя из качеств именинника), либо вручную собрать поздравление из полученных блоков. При первом варианте приложение само генерирует поздравление из одного начального, одного серединного и одного конечного блока, которое можно сохранить к себе и скопировать для последующей отправки. При втором варианте пользователь сам выбирает необходимые блоки из полученных ранее, после чего приложение генерируется поздравление из выбранных блоков.

Пользователь проходит 4 этапа создания поздравления (при выборе блоков вручную): настройка поздравления, выбор начальных блоков, выбор серединных блоков и выбор конечных блоков. В конце пользователь получает сгенерированное по его пожеланиям поздравление.

При случайной генерации поздравления пользователь проходит только 1 этап создания: первоначальная настройка. При нажатии кнопки случайной генерации, пользователь получит случайное поздравление исходя из первоначальной настройки.

* Создание собственного блока поздравления. Пользователь сам может создавать блоки для генерации поздравлений. После заполнения пользователем определенной формы и указания тегов, приложение отправляет блок на удаленную БД.
* Просмотр поздравлений и собственных блоков. Пользователь может просматривать собственные поздравления, созданные раннее, а также свои блоки поздравлений. Имеется возможность их редактирования и сохранения блоков и поздравлений.

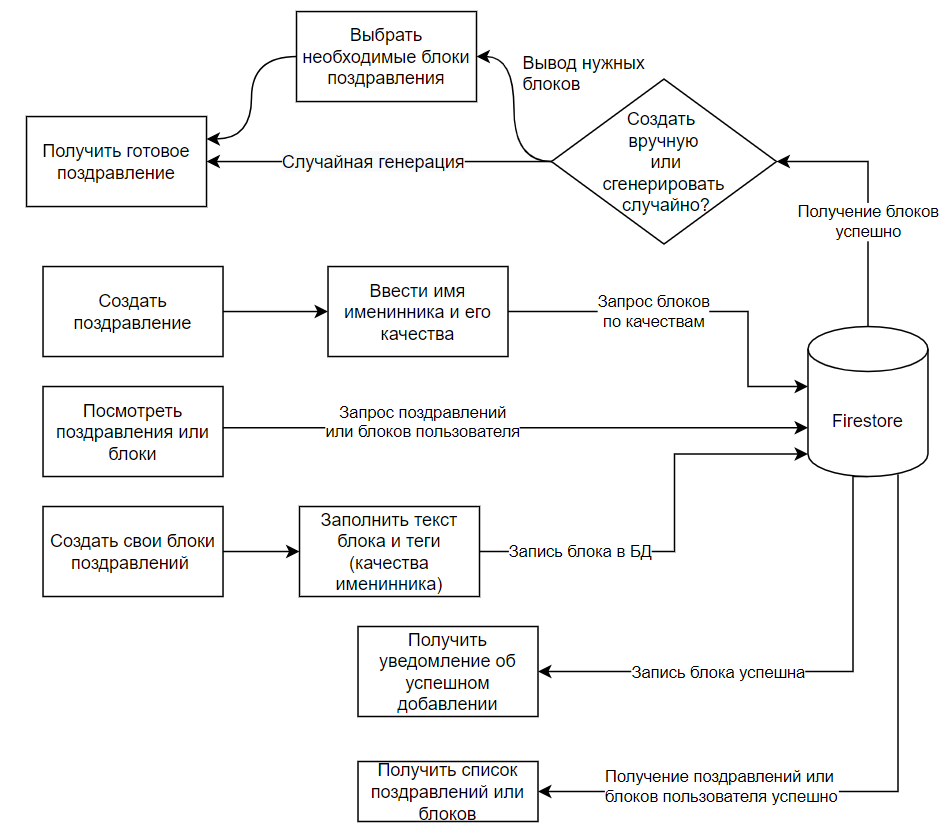


Рис.3 – диаграмма использования приложения

## Эскизирование экранов приложения

В таблице 1 предоставлено описание эскизов экранов с ссылкой на номер рисунка. Скриншоты экранов находятся в Приложении 1 данной работы.

Таблица 1. Эскизы экранов и их описание

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название экрана | Функции экрана | № рисунка в приложении |
| 1 | Экран авторизации пользователя | Содержит в себе поля ввода логина и пароля, а также кнопки «Вход» и «Регистрация». Кнопка «Регистрация» открывает экран регистрации. При удачном входе открывает экран списка поздравлений пользователя. | 24 |
| 2 | Экран регистрации пользователя | Содержит в себе поля ввода логина, пароля и повтора пароля. При удачной регистрации открывает экран, авторизации. | 25 |
| 3 | Список поздравлений пользователя | Содержит созданные пользователем поздравления (при первом входе он будет пустым). Элементы списка поздравлений являются интерактивными: при коротком нажатии будет открыт экран редактирования поздравления, при долгом нажатии будет предложено удалить выбранное поздравление. У пользователя имеется возможность перейти на экран списка созданных им блоков с помощью меню навигации, располагающегося на нижней стороне экрана. Также имеется возможность выйти из аккаунта, воспользовавшись выпадающим меню в верхнем правом углу. | 26 |
| 4 | Список блоков поздравлений пользователя | Содержит созданные пользователем поздравления (при первом входе он будет пустым). Элементы списка блоков похожи поведением на элементы списка поздравлений: при коротком нажатии будет открыт экран редактирования блока, при долгом нажатии будет предложено удалить выбранный блок. | 27 |
| 5 | Экран создания блока поздравлений | Содержит настройки для создания блока, такие как: текст поздравления, тип блока (начальный, серединный или конечный), пол, теги и др. При нажатии на кнопку «Создать блок» блок поздравления сохраняется в список блоков поздравлений. | 28 |
| 6 | Экран редактирования блока поздравлений | Содержание данного экрана аналогично экрану создания блоков поздравлений. При нажатии кнопки «Применить изменения» блок сохраняется с измененными настройками. | 29 |
| 7 | Экран редактирования поздравления | Содержит в себе поля заголовка поздравления (по умолчанию – имя именинника и дата создания поздравления) и сам текст поздравления. В данном экране можно изменить заголовок и текст поздравления, а также сохранить в буфер обмена текст поздравления. | 30 |
| 8 | Экран настройки создания поздравления | Начальный экран для настройки поздравления. Настраивается имя именинника, пол, а также качества-теги для поиска конкретных блоков поздравлений. При нажатии кнопки «Сгенерировать случайными образом» создастся поздравление из одного начального, одного серединного и одного конечного блока. При нажатии кнопки «Далее» будет произведен переход на экран выбора начальных блоков поздравления. | 31 |
| 9 | Экран выбора начальных блоков | Представляет из себя список блоков, которые можно выбирать. Элемент списка будет помечаться номером при коротком нажатии на него. После выбора всех блоков необходимо нажать «Далее» для перехода к выбору серединных блоков. | 32 |
| 10 | Экран выбора серединных блоков | Аналогичен экрану выбора начальных блоков. После выбора всех блоков необходимо нажать «Далее» для перехода к выбору конечных блоков. | 32 |
| 11 | Экран выбора конечных блоков | Аналогичен экрану выбора начальных и серединных блоков. После выбора всех блоков необходимо нажать «Сгенерировать» для создания поздравления. | 32 |
| 12 | Экран результата генерации поздравления | Данный экран содержит сгенерированное поздравление, которое можно скопировать в буфер обмена. Поздравление автоматически добавляется в список поздравлений пользователя. | 33 |

## Логическое моделирование проекта

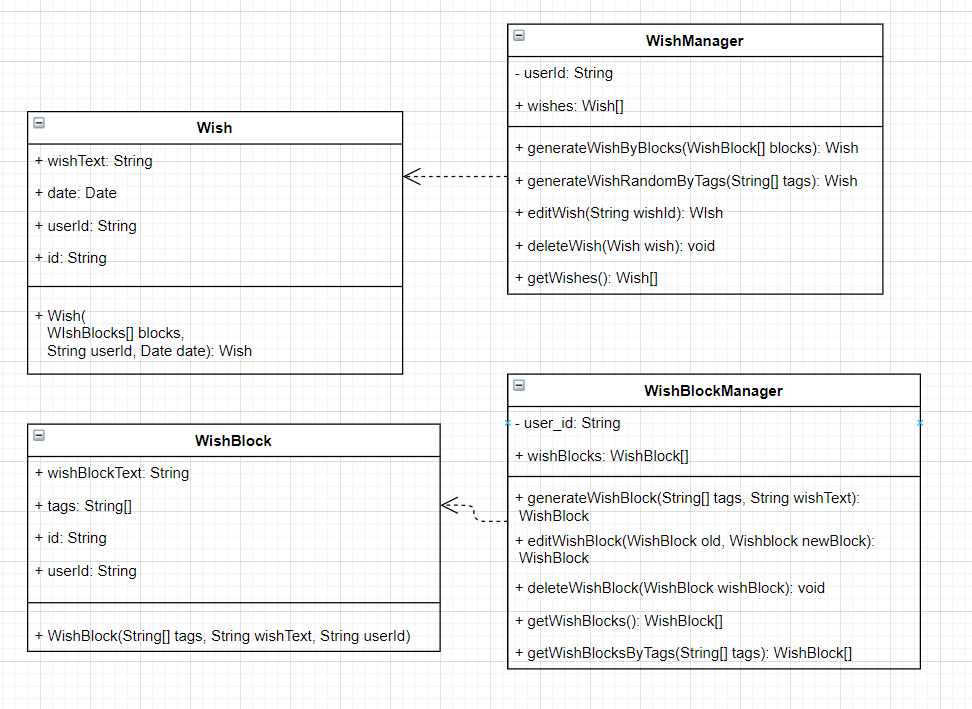


Рис. 4 – упрощенная логическая модель приложения

Основная логическая модель строится вокруг двух классов, представляющих основные типы данных – классы DTO (Data Transfer Object) Wish и WishBlock, отвечающие за описание поздравления и блока поздравления соответственно. Класс Wish содержит в себе информацию о поздравлении: дата создания, текст поздравления, информация о id пользователя, собственный id. Класс WishBlock содержит в себе информацию о блоке поздравления: текст поздравления, теги для определения темы поздравления и качества именинника, собственный id тега и id пользователя.

В качестве основных классов экранов используются фрагменты. Классы Wish и WishBlock будут элементами списков для заполнения экранов.

Статические классы WishManager и WishBlockManager содержат в себе блоки и поздравления пользователя, а также методы для предоставления фрагментам доступа к данным. Также эти классы содержат методы, такие как методы генерации поздравления случайным образом, либо из выбранных блоков, метод предоставления списка блоков по выбранным тегам и т.д. Фрагменты будут обращаться к этим классам для заполнения списков блоков и поздравлений. На этапе разработки класс WishBlockManager так и не был разработан, его основные функции были перенесены в класс FireBaseDataAgent.

# Программная реализация

## Блоки и поздравления

Класс блоков поздравлений претерпели изменения в сравнении с этапом логического проектирования: добавились поле списка фильтров, а также поле публичности блока. Блоки фильтров понадобились для фильтрования поздравлений в зависимости от пола, отсутствия/присутствия имени именинника, этапа создания поздравления. С этими нововведениями возможно создавать синтаксически правильные поздравления, а также давать адекватный выбор блоков при разных настройках поздравления.

В класс поздравлений Wish произошла замена поля даты на поле заголовка, которое генерируется в процессе поздравления, а также добавилось поле уникального идентификатора пользователя.

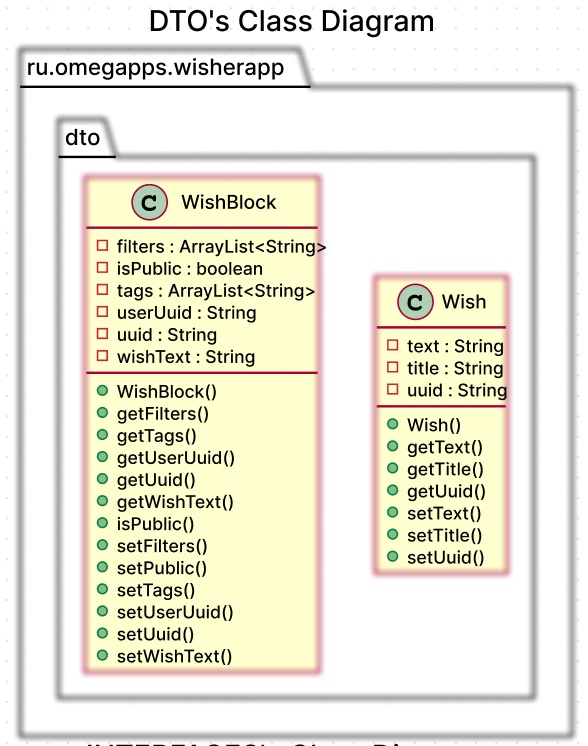


Рис.5 – UML-диаграммы классов Wish и Wishblock

## Класс-менеджер – WishBlockManager

Класс WishManager предоставляет методы для генерации поздравлений, а также содержит в себе списки выбранных пользователем блоков, из которых впоследствии будут сгенерированы поздравления.

Для создания выборки блоков по первоначальной настройке поздравления в этот класс заносятся основные данные об имениннике: пол, имя и теги поздравления. Исходя из наличия имени, формируется поле состояния имени, которое заносится в список фильтров для составления более правильной выборки блоков.

В классе WishManager содержится 4 списка: блоки – начальные, серединные и конечные, а также список для соединения всех блоков в одно поздравление. На каждом этапе создания эти списки заполняются выбранными пользователем блоками. Генерация поздравления представляет из себя последовательную склейку строк блоков поздравлений (со вставкой имени именинника, если таковое имеется).

Часть функционала класса WishManager и весь функционал WishBlockManager на этапе разработки было перенесено в статический класс FireBaseDataAgent. Класс использует статический класс FireBaseDataAgent для предоставления фрагментам списков необходимых для этапа построения поздравления блоков, а также для создания поздравлений.

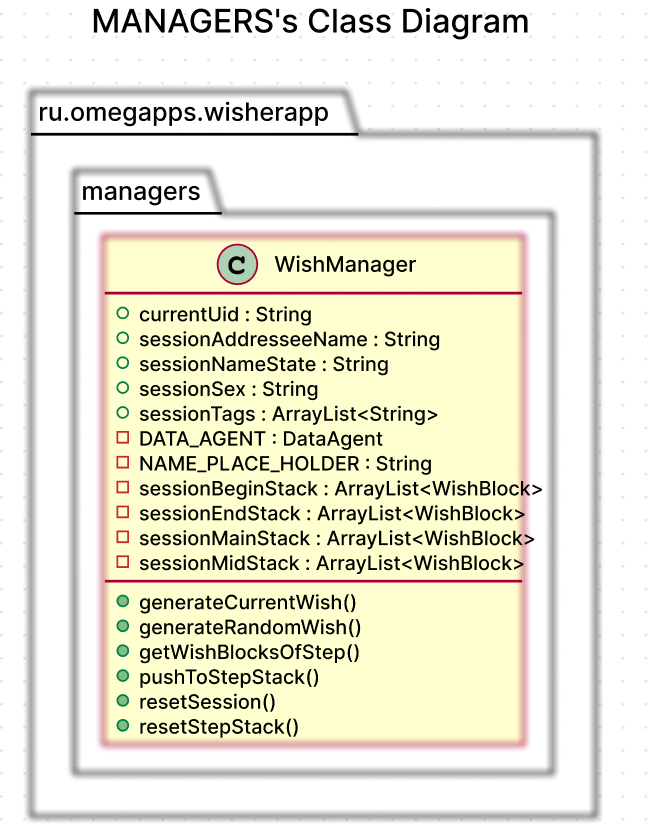


Рис.6 – UML-диаграмма класса WishManager



Рис.7 – пример использования класса FireBaseDataAgent

## Общение с удаленной БД – класс FireBaseDataAgent и технология Firebase

Данный класс содержит в себе динамически изменяющиеся списки публичных блоков и блоков пользователя, предоставляет интерфейс взаимодействия с удаленной БД Firebase. Для создания класса использовалась удаленная БД Firebase от Google.

С помощью интерфейса, предоставленного SDK (Software Development Kit – набор инструментов разработчика ПО) Firebase было реализовано динамическое заполнение списков публичных блоков и блоков пользователя при изменении данных в БД. Это стало возможным с помощью добавления слушателей (рисунок 8) на события изменения базы данных, которые после возникновения событий сразу обновляют списки приложений. Списки блоков используются классом WishManager для предоставления блоков по выбранным тегам, генерации поздравления как случайным образом, так и по выбранным блокам. UML-диаграммы текущего и последующих классов будет включены в главу «Заключение» данной работы.

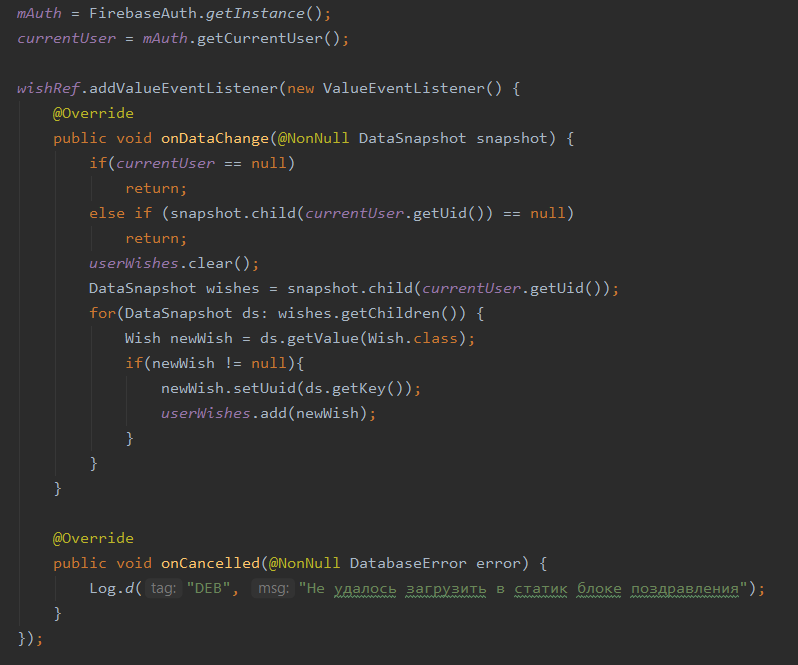


Рис.8 – использование методов библиотеки Firebase

## Адаптеры для списков блоков и поздравлений – классы WishRecyclerAdapter и WishBlockRecyclerAdapter.

Контейнером для списка блоков и поздравлений в фрагментах является RecyclerView. Так как блоки поздравлений имеют несвойственную для простых списков информацию, возникла необходимость создать пользовательские адаптеры для RecyclerView под каждый DTO-класс – WishRecyclerAdapter и WishBlockRecyclerAdapter для RecyclerView, содержащих Wish и WishBlock соответственно.

Классы являются наследниками класса RecyclerView.Adapter, и необходимы, чтобы правильно заполнить RecyclerView. Внутри каждого из  
2-х классов созданы классы, которые представляют графический интерфейс одного элемента списка – внутренние классы WishViewHolder и WishBlockViewHolder. Были переписаны методы создания графического представления элемента списков (рисунок 9). Также данными классами предоставлена возможность задания логике нажатия и долгого нажатия по элементам (рисунок 10), что даст впоследствии реализовать удаление и редактирование элемента.

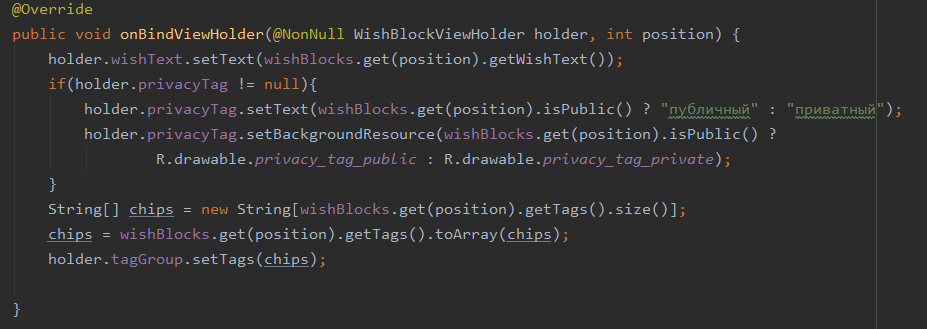


Рис.9 – метод onBindViewHolder класса WishBlockViewHolder

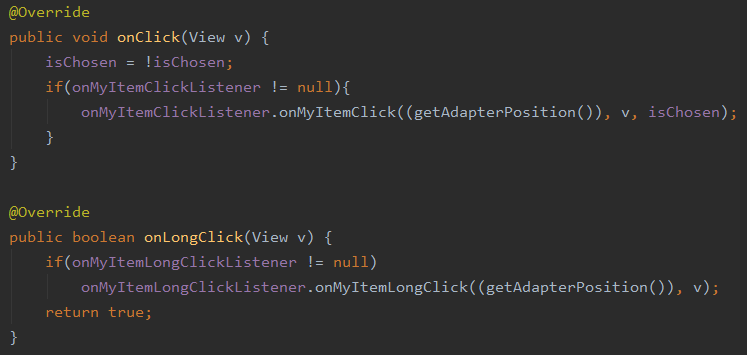


Рис.10 – применение пользовательских интерфейсов для расширения логики нажатия по элементам RecyclerView

## Основной экран (поздравления и блоки)

Классы основных экранов наследуются от стандартного класса из Android SDK – класса Fragment. Переход между фрагментами осуществляется заменой текущего фрагмента в контейнере фрагментов на другой (рисунок 11).

Основные экран представляет из себя два фрагмента, содержащих в себе списки поздравлений и блоков поздравлений пользователя. В методе инициализации графики фрагмента задаётся логика короткого и долгого нажатия, которая проведена к классам типа ViewHolder с помощью вручную созданных интерфейсов OnMyClickListener и OnMyLongClickListener (рисунок 12).

Данные в списки загружаются посредством добавления слушателей на представления БД Firebase: при изменении БД, будет обновляться и список. Для удаления созданы методы удаления с использованием представлений БД Firebase, а также создание всплывающего предупреждения об удалении (рисунок 13). До тех пор, пока не загружены списки, пользователь будет видеть состояние загрузки, представляющего из себя круглый иконку с анимацией (рисунок 14).

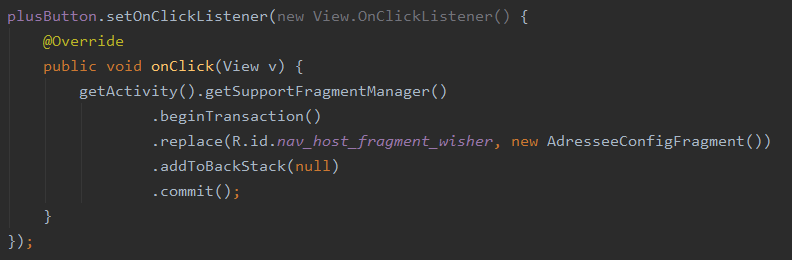


Рис.11 – переход осуществляется с помощью класса FragmentManager



Рис.12 – логика нажатия на элементы в списке



Рис.13 – логика удаления элемента списка с вызовом предупреждающего уведомления

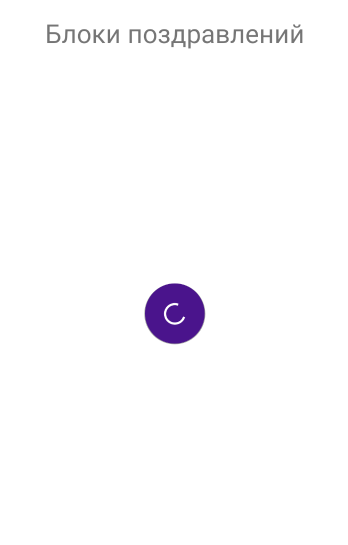


Рис. 14 – ожидание загрузки с использованием прогресс-бара

## Генерация и редактирование блоков (экран создать блок). Редактирование поздравлений

Экраны создания и редактирования блоков поздравлений содержат в себе элементы для настройки блока. В коде фрагмента описаны логики нажатия на кнопки (путем задания интерфейса View.onClickListener), логики задания и удаления тегов (интерфейс TagGroup.setTagListener), логики изменения выбора радиокнопок и чекбоксов (интерфейс onCheckedChangeListener) (рисунок 15). После нажатия кнопки происходит запись в БД Firebase (рисунок 16).

Экран редактирования поздравлений содержит только два поля, в которых можно отредактировать заголовок и текст поздравления. Также есть две кнопки, одна из них дает возможность скопировать блок в буфер обмена, это было реализовано через класс ClipboardManager, позволяющий использовать системный буфер обмена (рисунок 17). Вторая кнопка сохраняет изменения в БД Firebase, данные берутся из вышеописанных полей заголовка и текста поздравления.

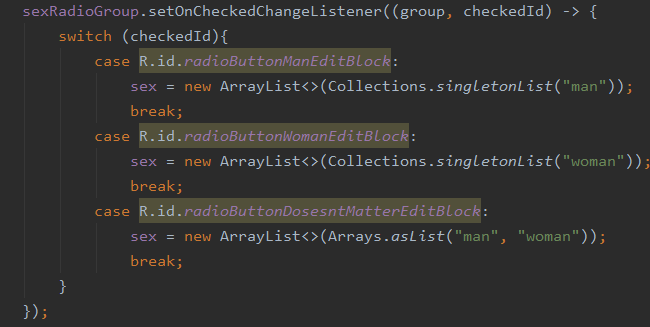


Рис.15 – логика изменения данных при изменении радиокнопки

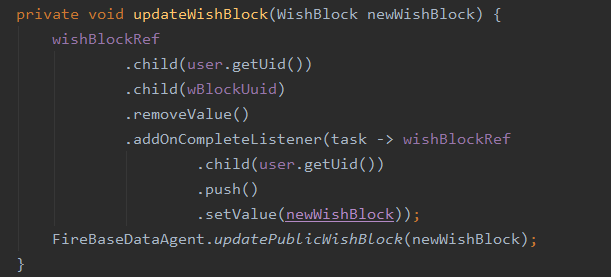


Рис.16 – метод записи данных с помощью Firebase

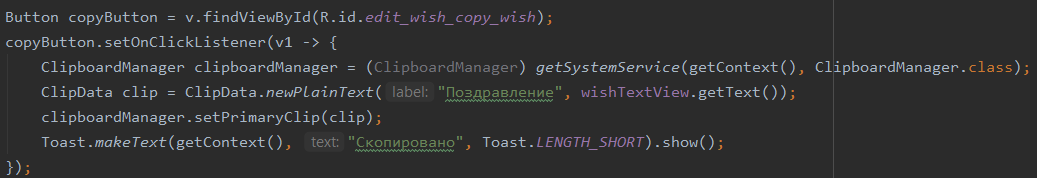


Рис.17 – копия в буфер обмена

## Генерация поздравлений (экраны настройки поздравления и выбора блоков)

Генерация поздравлений поделена на 4 экрана.

Сначала идет экран первоначальной настройки, в которой задаются данные об имениннике, впоследствии эти данные перетекают в WishManager.

При выборе случайной генерации, вызывается метод, который формирует посредством класса WishManager поздравление. Потом генерируется название, состоящее из текущей даты и имени именинника (если есть). После чего осуществляется переход на финальный экран, в котором можно скопировать поздравление в буфер обмена, а поздравление сохраняется в профиль пользователя.

При выборе генерации вручную будет осуществлен переход на экраны выбора начальных, серединных и конечных блоков. Экраны будут содержать списки блоков поздравлений, в каждом из экранов будут соответствующие блоки (начальные, серединные или конечные).

Для создания списка блоков используются аналогичные адаптеры RecyclerView, что и для основных экранов, однако для генерации поздравления задается другая логика для элементов списка. Если нажать на блок, то блок выделится номером, который будет показывать его порядок при генерации поздравления (рисунок 18).

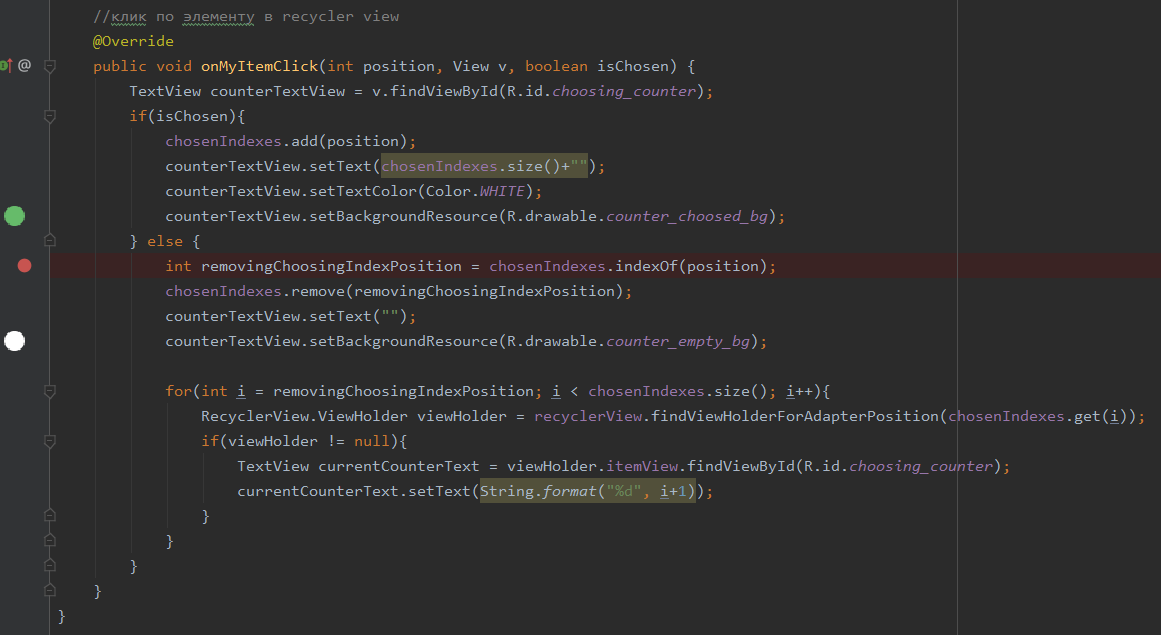


Рис. 18 – логика клика по элементу с автоматической перестройкой порядка блоков

При нажатии кнопки завершения выбора («Далее» и «Сгенерировать») блоки будут добавляться в соответствующий список в классе WishManager, а при нажатии на кнопку «Сгенерировать» запустится соответствующий метод в WishManager.

## Заимствованные элементы.

Для создания тегов была использована библиотека TagGroup за авторством Jun Gu (2dxgujun).

Страница в Github: https://github.com/2dxgujun.

# ЗАКЛЮЧЕНИЕ

Результатом данной работы стало приложение для создания поздравлений, она всем необходимым обладает функционалом, решающим поставленную в данной работе цель. В отличие от плана, в приложение был добавлен функционал публикации пользовательских блоков, в остальном же получилось полученное приложение полностью выполняет поставленные задачи, однако не обошлось без недочётов, вскрывшихся после тестирования приложения.

Так как в приложении на начальном этапе нет необходимого кол-ва блоков поздравлений, необходимо будет добавить их перед релизом приложения. Также при дальнейшем обновлении следует показывать, какие категории-теги доступны для выбора, по каким количество блоков для выбора будет равно нулю.

Возникали трудности с событиями добавления элементов в БД Firebase (из-за недочетов в коде каждый раз при инициализации экрана создавались новые слушатели событий изменения БД, отчего с каждой инициализацией экрана добавлялось все больше и больше публичных блоков поздравлений). Также по невнимательности возникали и другие мелкие ошибки.

В ходе программной реализации были разработаны дополнительные экраны авторизации и регистрации, так как было принято решение сделать веб-приложение с возможностью публикации пользовательского контента.

В итоге, упрощенная логическая модель стала выглядеть так:

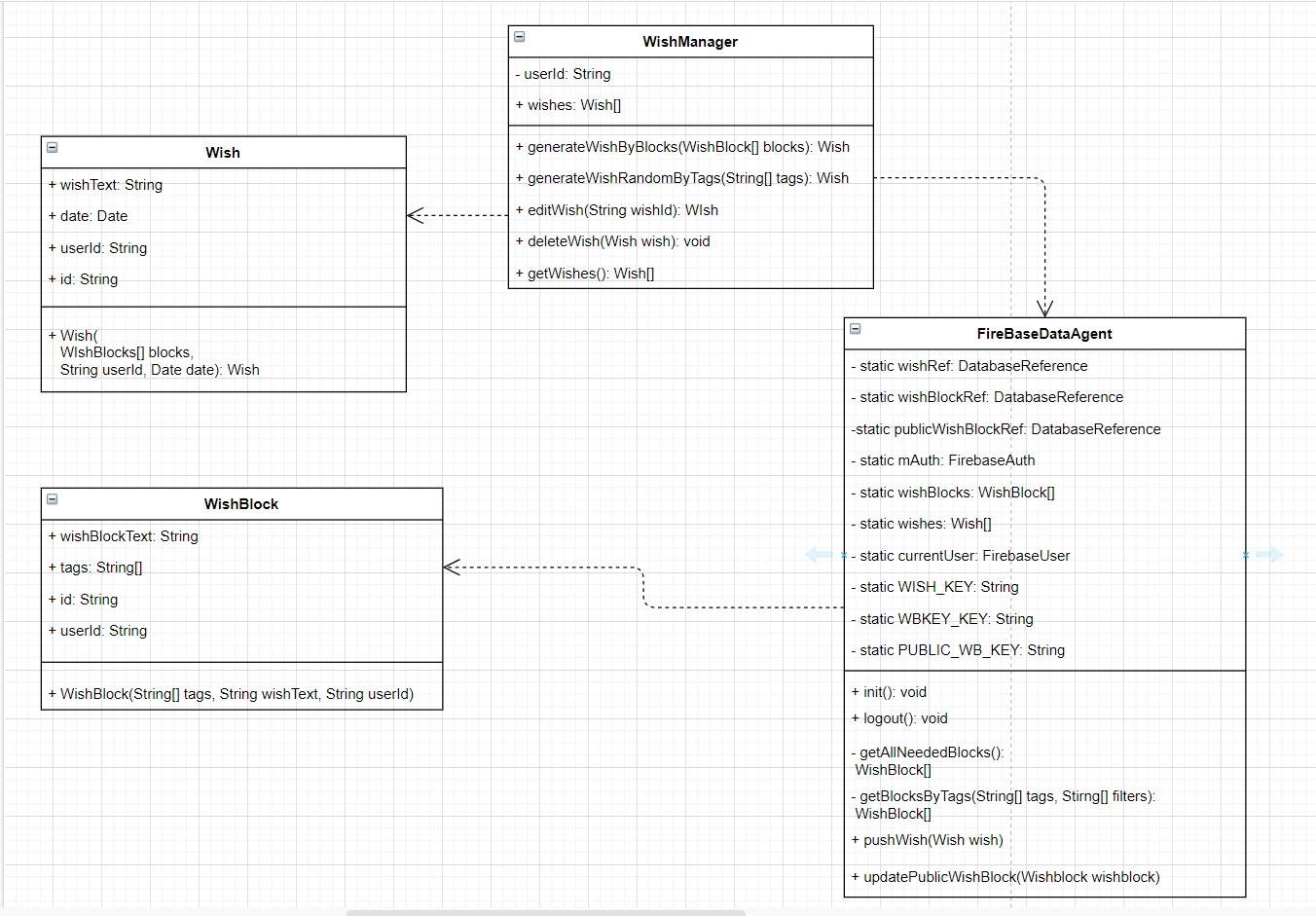


Рис.19 – упрощенная логическая схема основных классов данных

Полная логическая модель в формате UML:

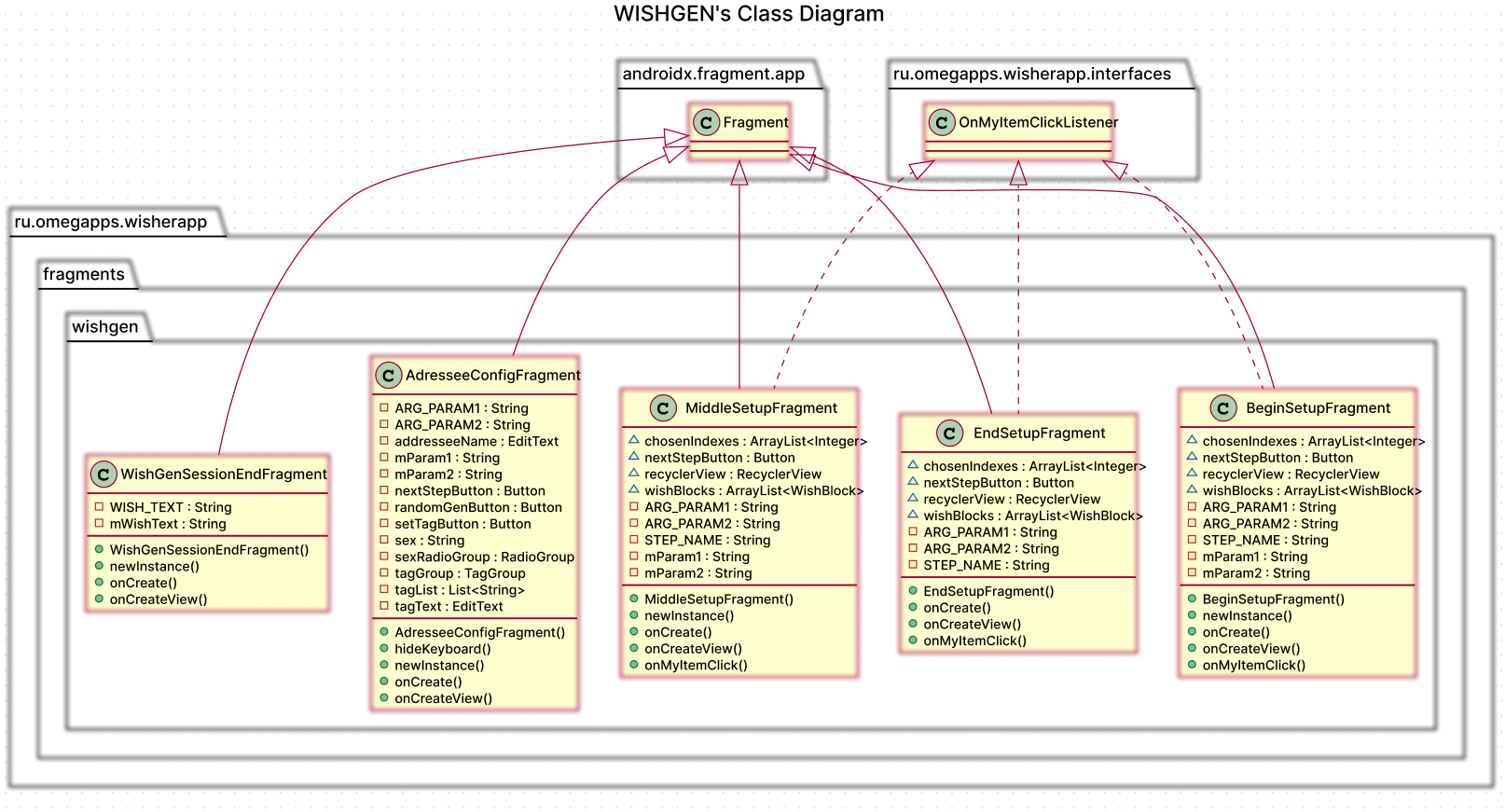


Рис. 19 – UML классов пакета создания поздравления

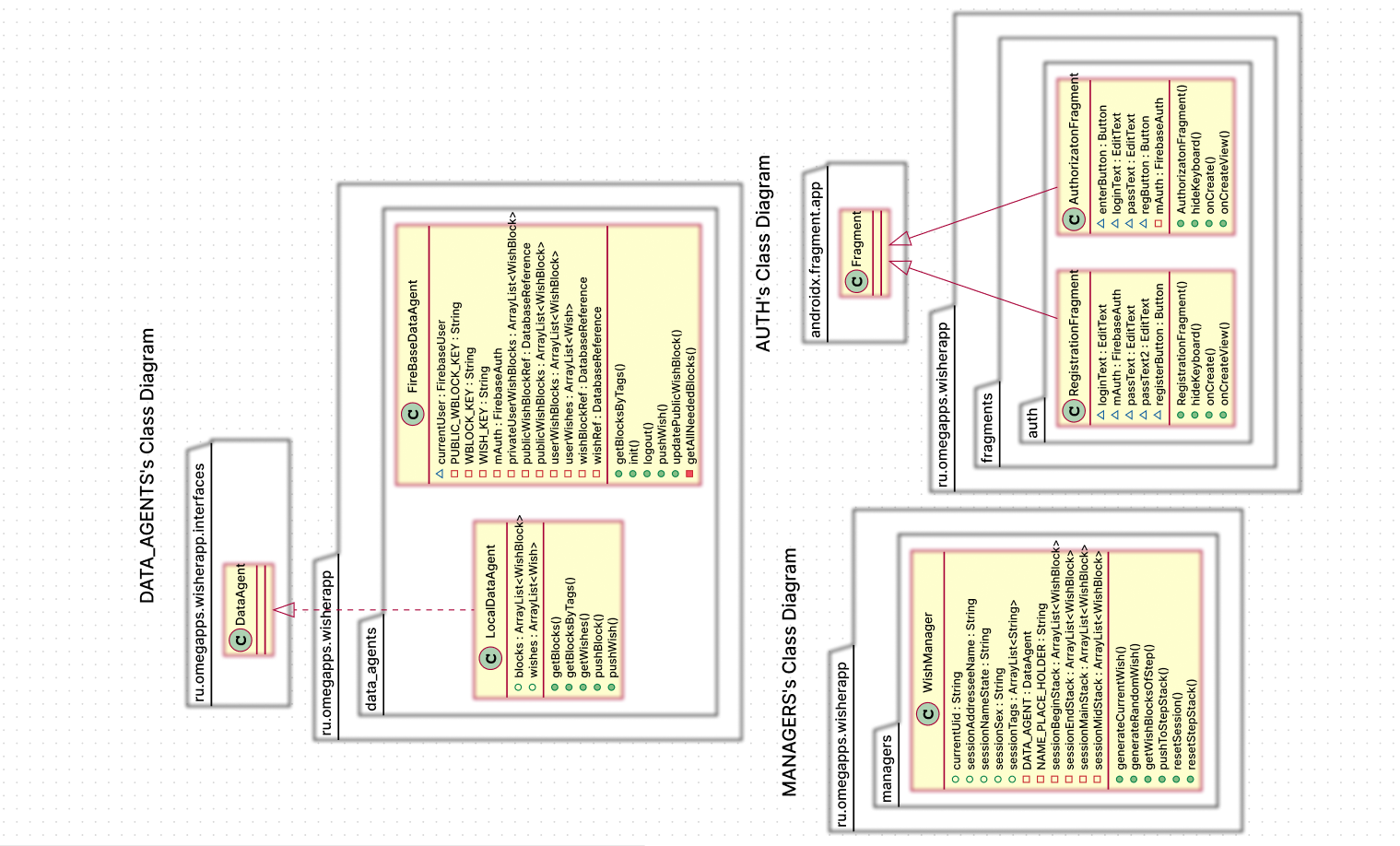


Рис.20 – UML классов авторизации, регистрации, менеджеров и интерфейсов для работы с данными

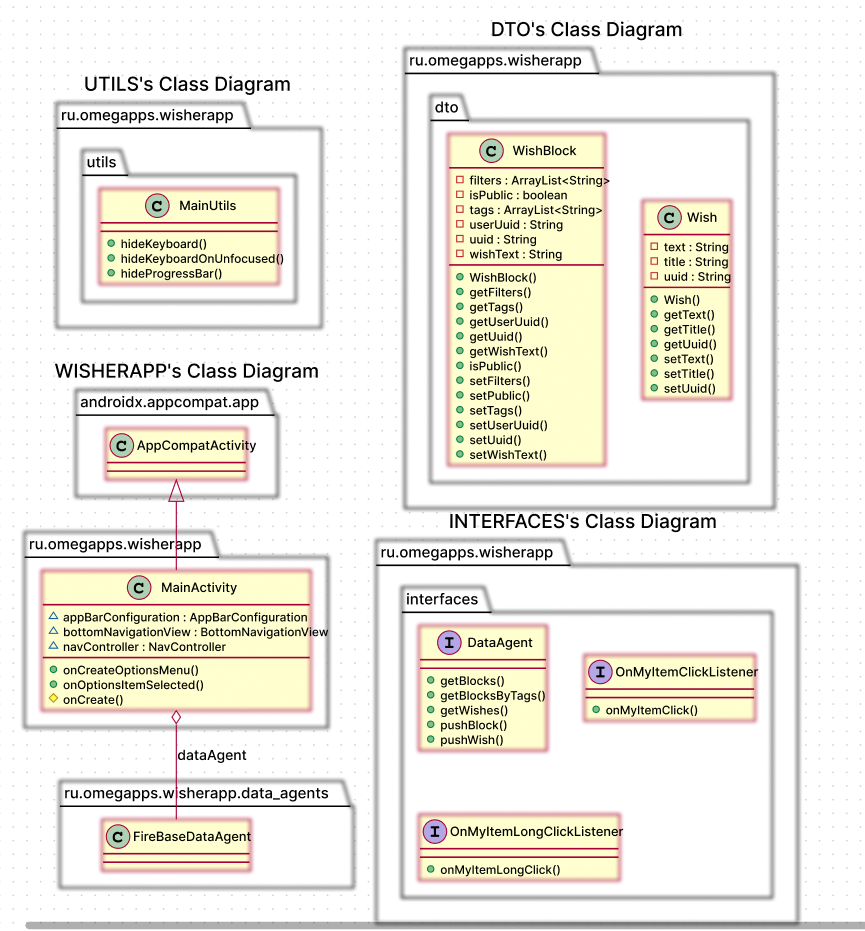


Рис.21 – классы основной активности, DTO-классов и интерфейсов

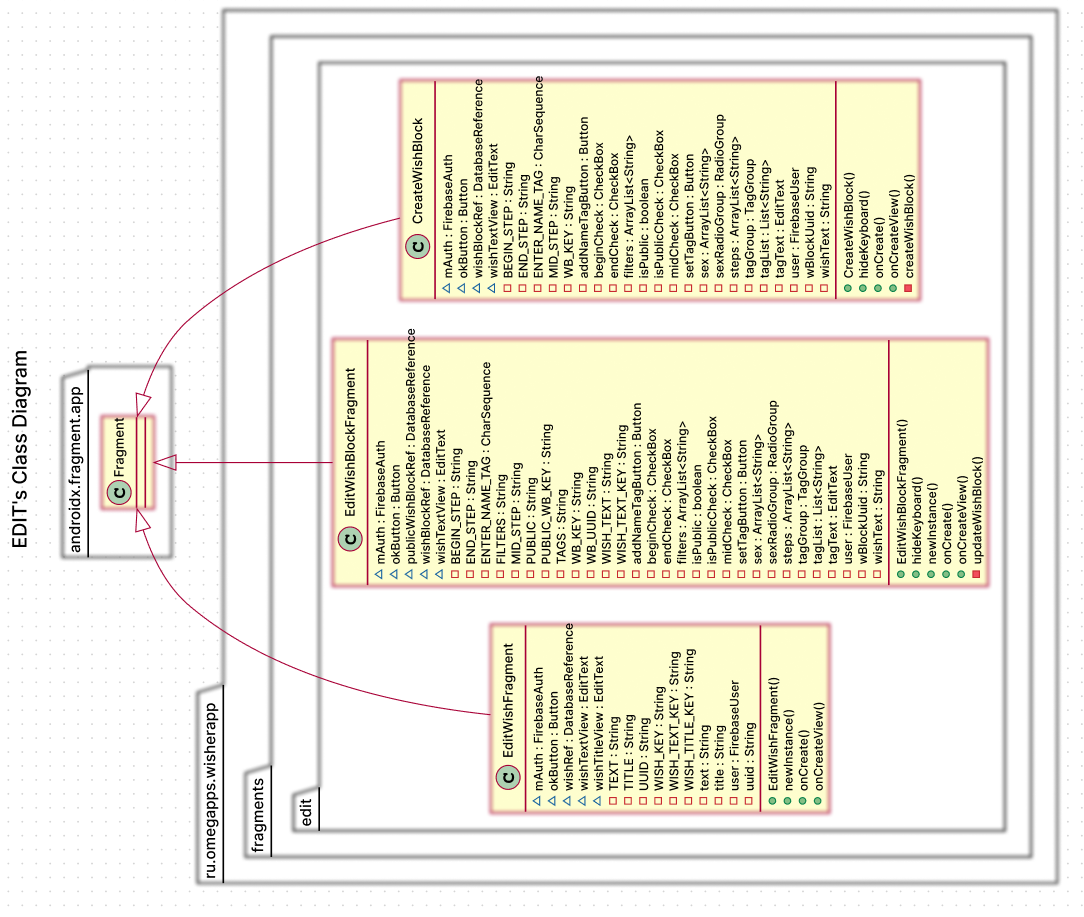


Рис. 22 – классы фрагментов редактирования блока и поздравления, создания блока

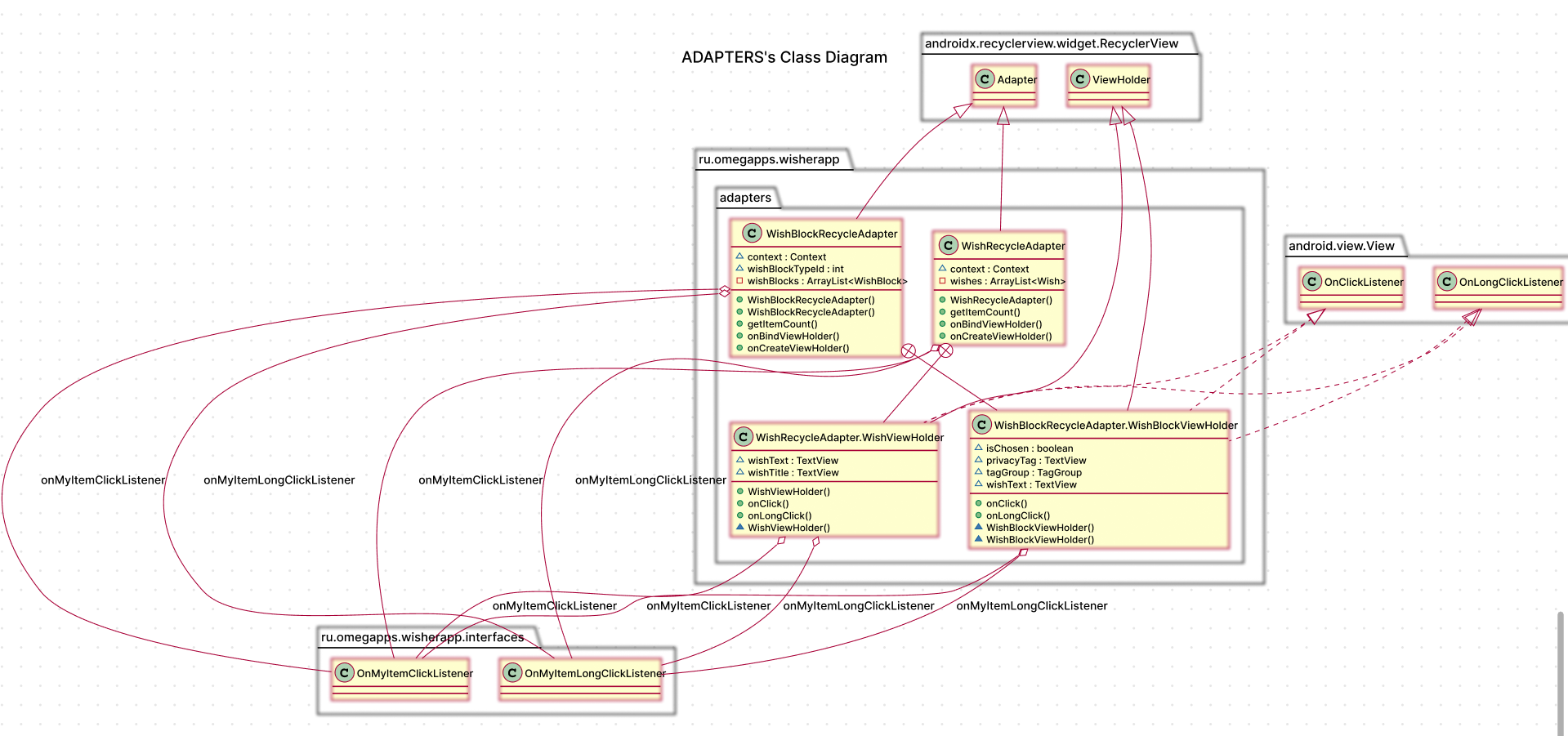


Рис.23 – диаграмма классов адаптеров для RecycleView

# Приложение 1. Скриншоты эскизов экранов приложения

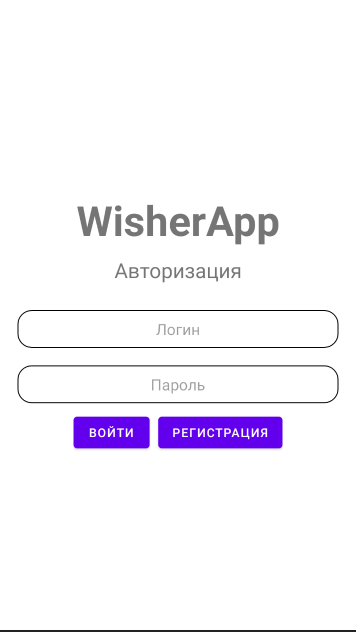


Рис.24 – экран авторизации

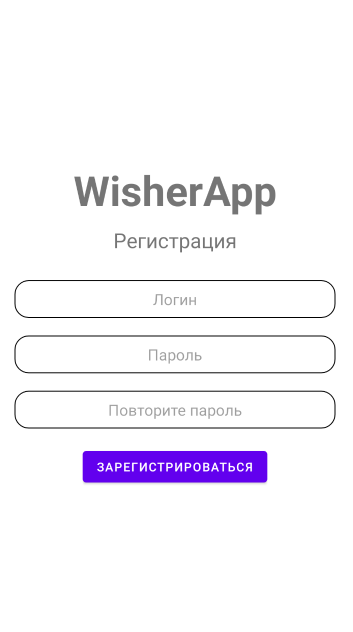


Рис.25 – экран регистрации



Рис.26 – список созданных пользователем поздравлений

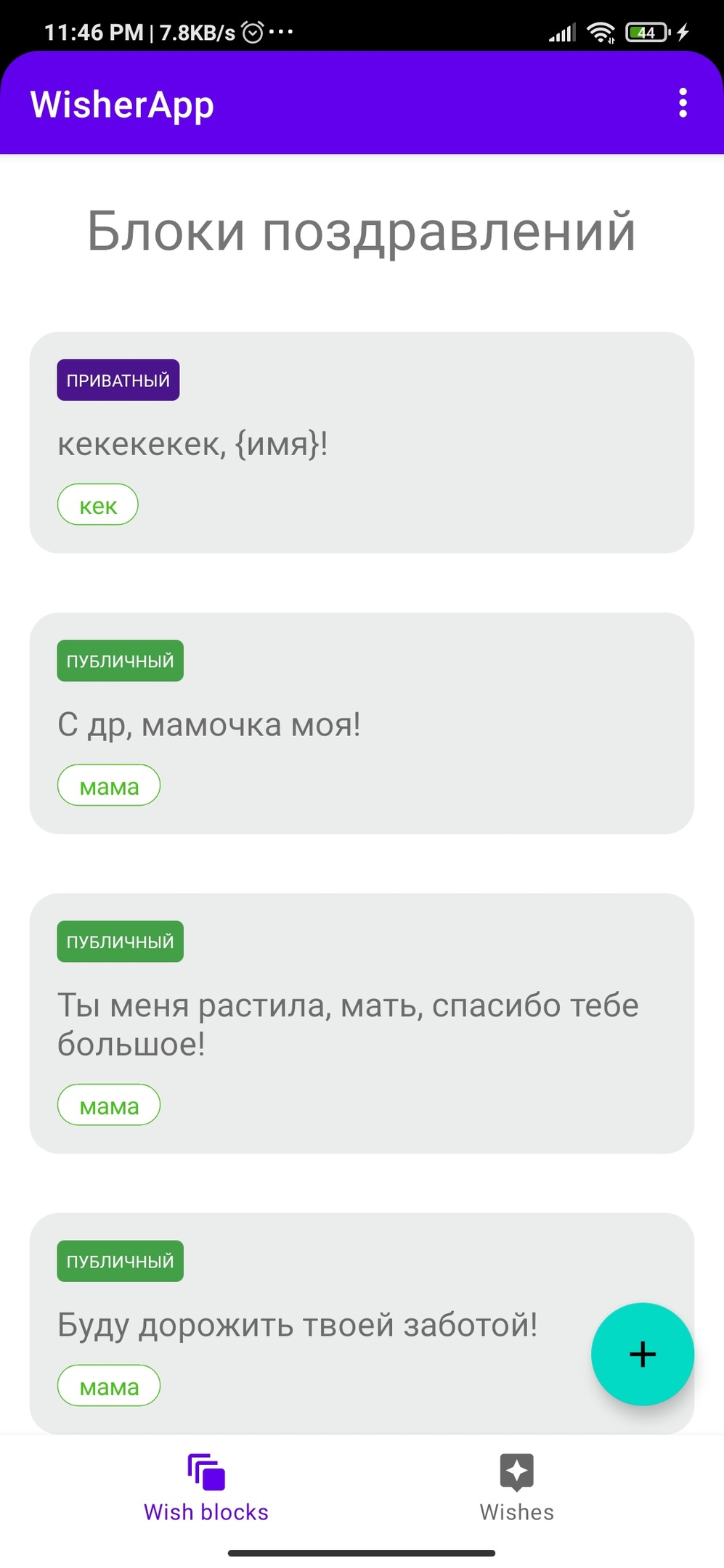


Рис. 27 – список блоков поздравлений пользователя

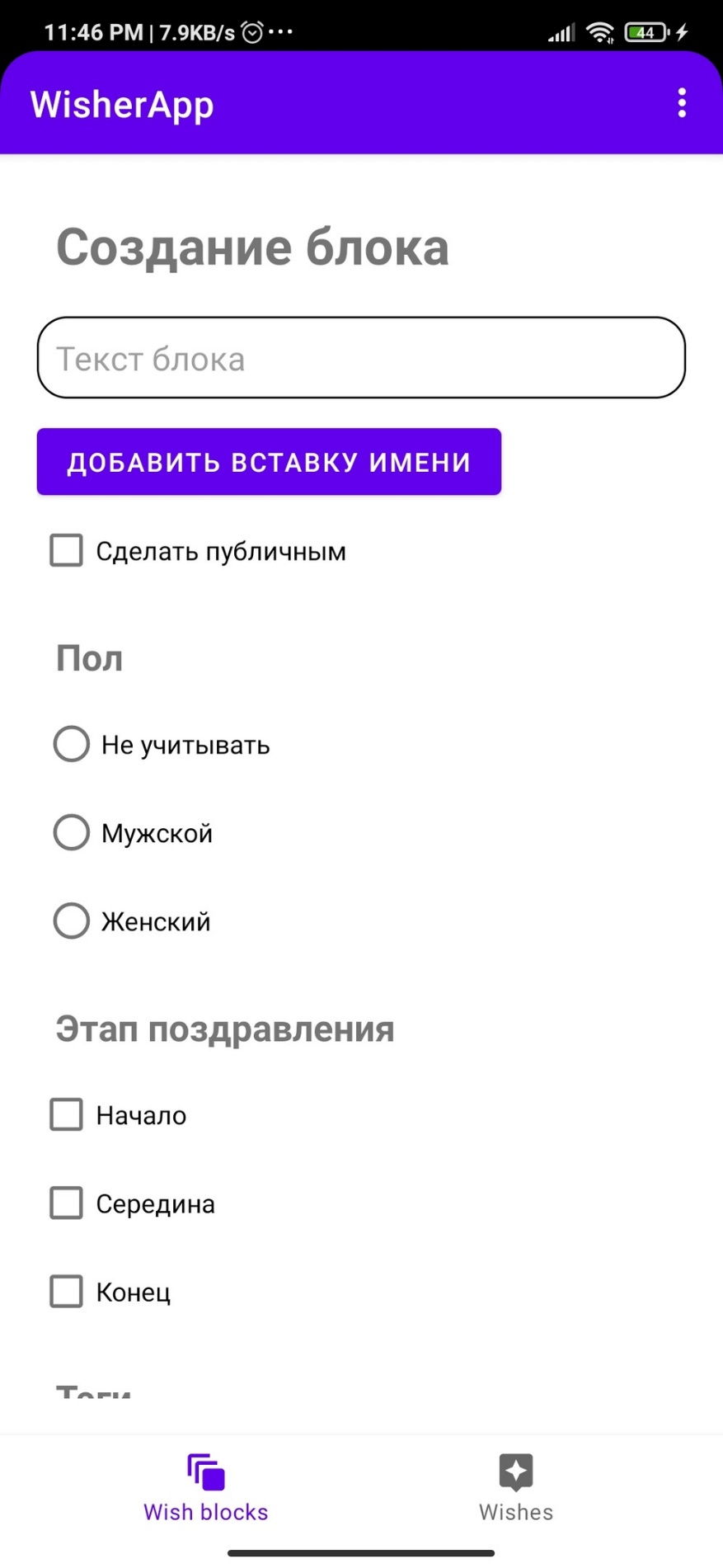


Рис.28 – создание блока поздравления

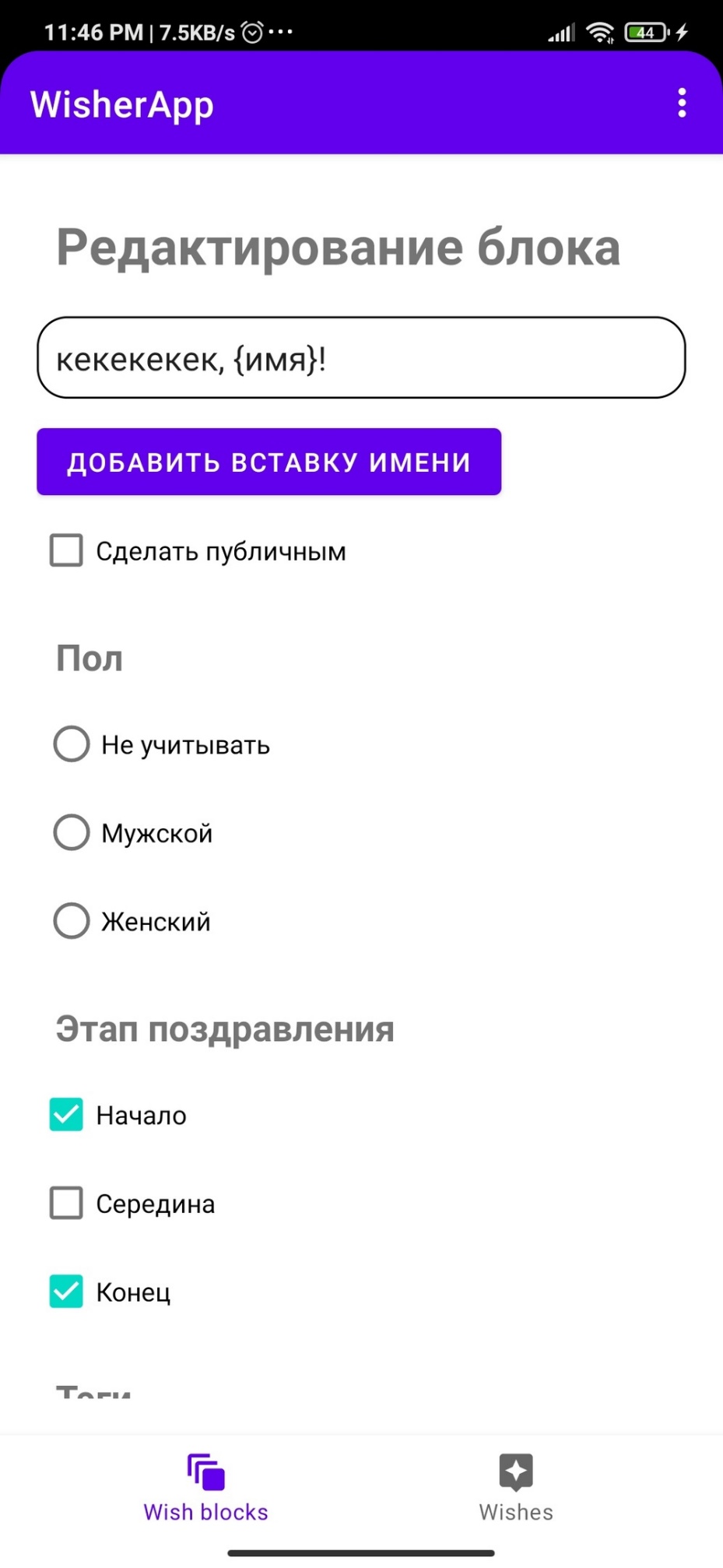


Рис. 29 – редактирование блока поздравления



Рис. 30 – экран редактирования поздравления

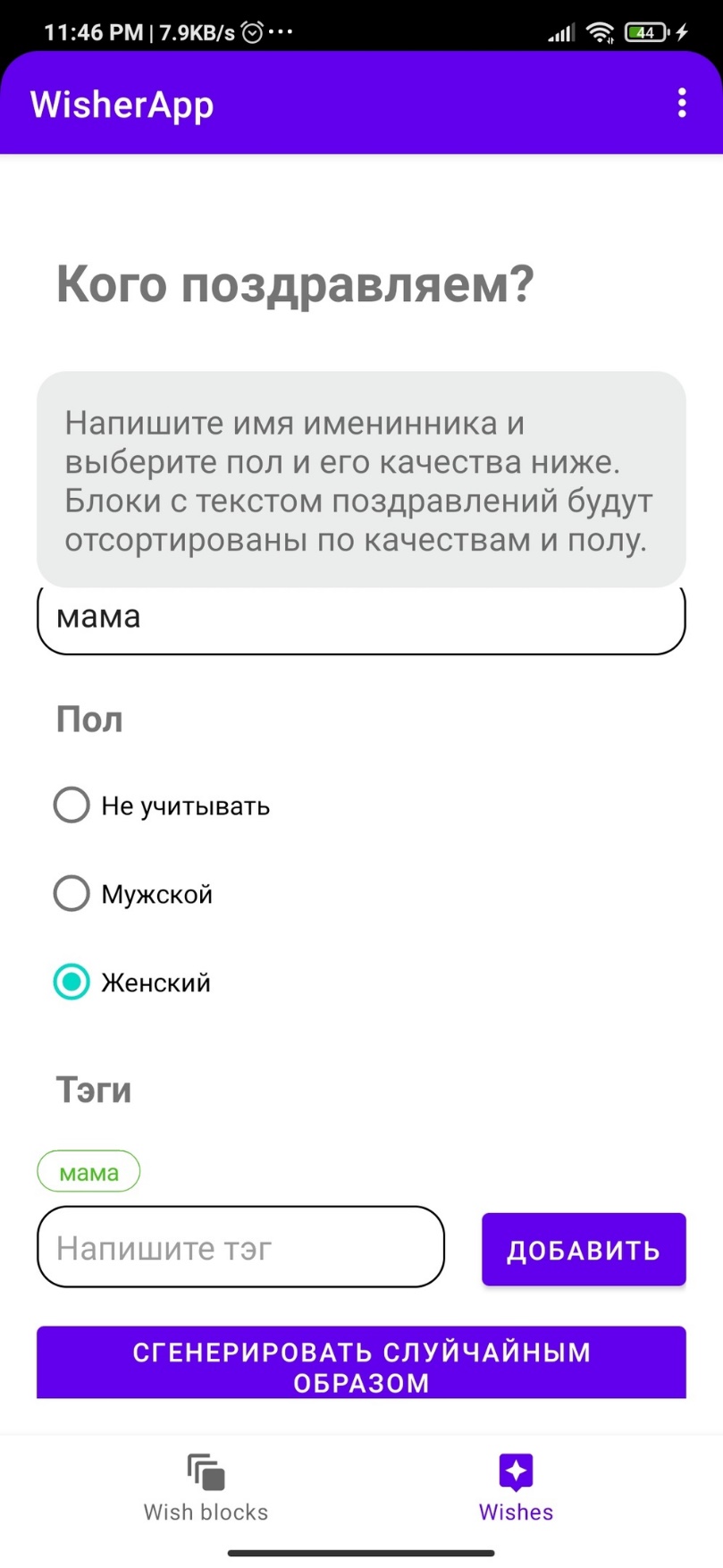


Рис.31 – экран настройки создания поздравления

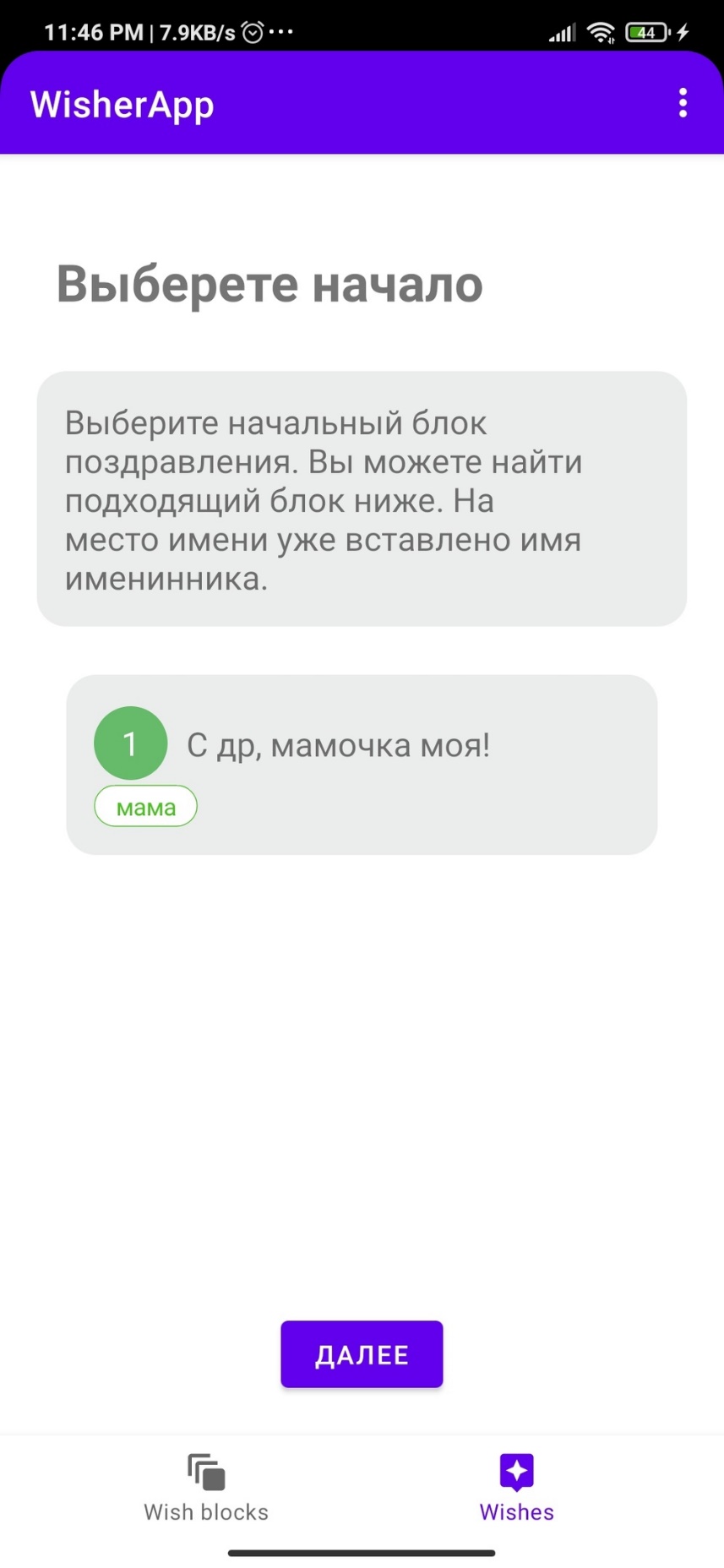


Рис.32 – этапы выбора блоков для начала, середины и конца

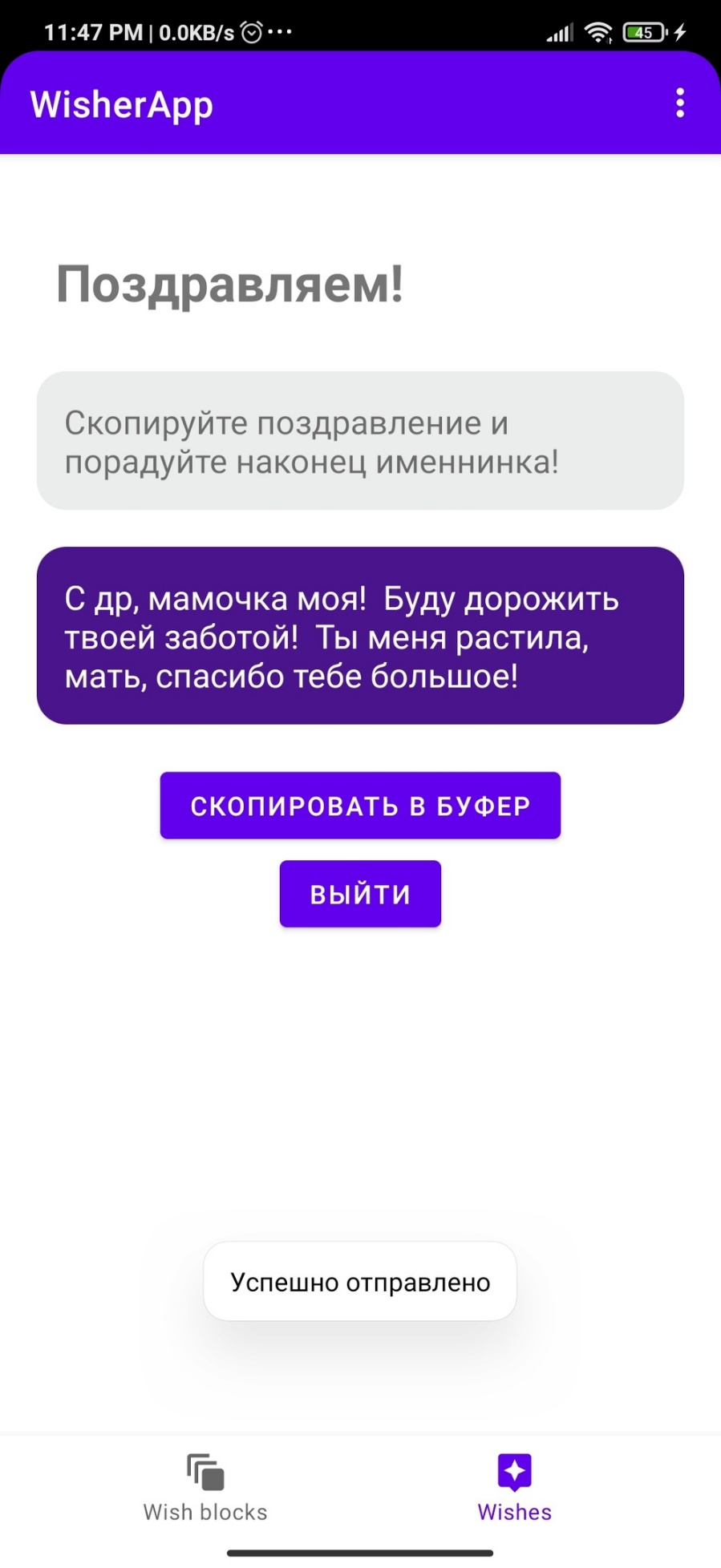


Рис.33 – заключительный экран с сгенерированным поздравлением