Федеральное агентство по образованию Российской Федерации

Государственное образовательное учреждение

высшего профессионального образования

Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

Отчёт по проектной работе

Приложение под android HomeAssistant

Выполнили:

Студентки ИИТММ, гр. 381706-1

Безручко Ирина Юрьевна

Кукушкина Ксения Олеговна

Максимова Ирина Игоревна

Проверил:

Лебедев Илья Генадьевич

г. Нижний Новгород

2019 г.

Оглавление

[Введение. 3](#_Toc9310384)

[Постановка задачи. 4](#_Toc9310385)

[Исследование рынка. 5](#_Toc9310386)

[Руководство пользователя. 8](#_Toc9310387)

[Руководство программиста. 11](#_Toc9310388)

[Описание структуры программы. 11](#_Toc9310389)

[Описание структур данных. 11](#_Toc9310390)

[Описание алгоритмов. 11](#_Toc9310391)

[Заключение. 12](#_Toc9310392)

[Список литературы 12](#_Toc9310393)

[Приложения. 13](#_Toc9310394)

# Введение.

Когда ты – студент-технарь, ты можешь позволить себе сразу две прелести жизни: самостоятельно жить, потому что ты уже студент, и полный комфорт со стороны техники – потому что ты технарь. Мы хотим представить проект приложения «HomeAssistant», созданного как для студентов-технарей, так и для тех, кто хочет обеспечить свою самостоятельную взрослую жизнь новшествами технологий 21 века.

Данное приложение создавалось под реальные задачи студентов, поэтому несет в себе ряд особенностей, о которых нет представления у крупных компаний, которые занимаются разработкой приложений. Обзор таких приложений представлен в главе «Исследование рынка». Мы взяли пример с удачных вариантов и добавили свои новшества для приложений подобного типа.

Цель работы: написать рабочую версию мобильного многопользовательского приложения.

Для достижения цели были поставлены следующие задачи:

1. Проанализировать рынок мобильных приложений, найти похожие и выявить их преимущества и недостатки.
2. Описать подробное техническое задание.
3. Разработать проект приложения.
4. Реализовать рабочий код.

Перед началом работы мы уже подобрали необходимые инструменты. То, почему мы выбрали именно платформу android, а язык программирования java можно узнать из таблиц исследований рынка в одноименной главе и описании руководства программиста. О выборе баз данных и прочих сторонних сервисов также можно прочитать из руководства программиста, описание алгоритмов. Те или иные методы и подходы мы выбирали исходя из советов, описанных в выбранной нами литературе. Так как полноценный курс android разработки в ВУЗе на нашем направлении не читается, то книг достаточно много.

Идеи приложения зародились на почве идеи совместного быта двух студенток, поэтому большинство примеров в работе будут рассмотрены именно на этом событии.

# Постановка задачи.

Создать многопользовательское приложение под android с возможностью обмена сообщениями авторизованных пользователей, поддержкой календаря, поддержкой групповых чатов, поддержкой записи и хранения данных о бюджете.

**Данные о себе.**

Требуется создать базу данных, в которой будет хранится идентификатор каждого пользователя, его данные для входа (номер телефона или почтовый адрес, пароль), личные данные (имя, фамилия, дата рождения). Информацию можно изменять (кроме идентификатора), можно удалять некоторые данные (например, оставить дату рождения пустым полем, такое нельзя сделать с данными для входа и идентификатором). Идентификатор является ключом пользователя, создается автоматически и запрещен к редактированию.

**Доска бюджета.**

Общий класс счет, который хранит в себе сумму, идентификатор создателя, и название (типа строки). Так же возможность хранить локально историю операций.

**Список покупок.**

Список, хранящий в себе название продукта, ориентировочную сумму.

**Сообщения.**

Данные из таблицы баз данных. Ячейки, которые доступны пользователю хранят в себе информацию о тексте сообщения, отправителе, получателе.

**Календарь.**

Класс «событие», который позволяет создавать мероприятия и подтягивать к дате необходимые траты.

# Исследование рынка.

Для начала работы необходимо решить вопрос о выборе платформы, которая должна отвечать минимуму требований:

1. Большое количество материалов и примеров для изучения работы выбранной платформы
2. Доступность IDE и компиляторов под языки программирования для данной системы

С выбором платформы нам помогло изучение крупных компаний, которые проводят аналитику мобильных устройств населения разных стран и в открытом доступе делятся своими исследованиями. Так, например, с помощью сайта StatCounter – Ирландского инструмента для обработки информации - можно понять, как в целом в мире распределяются мобильные операционные системы (рис. 1).

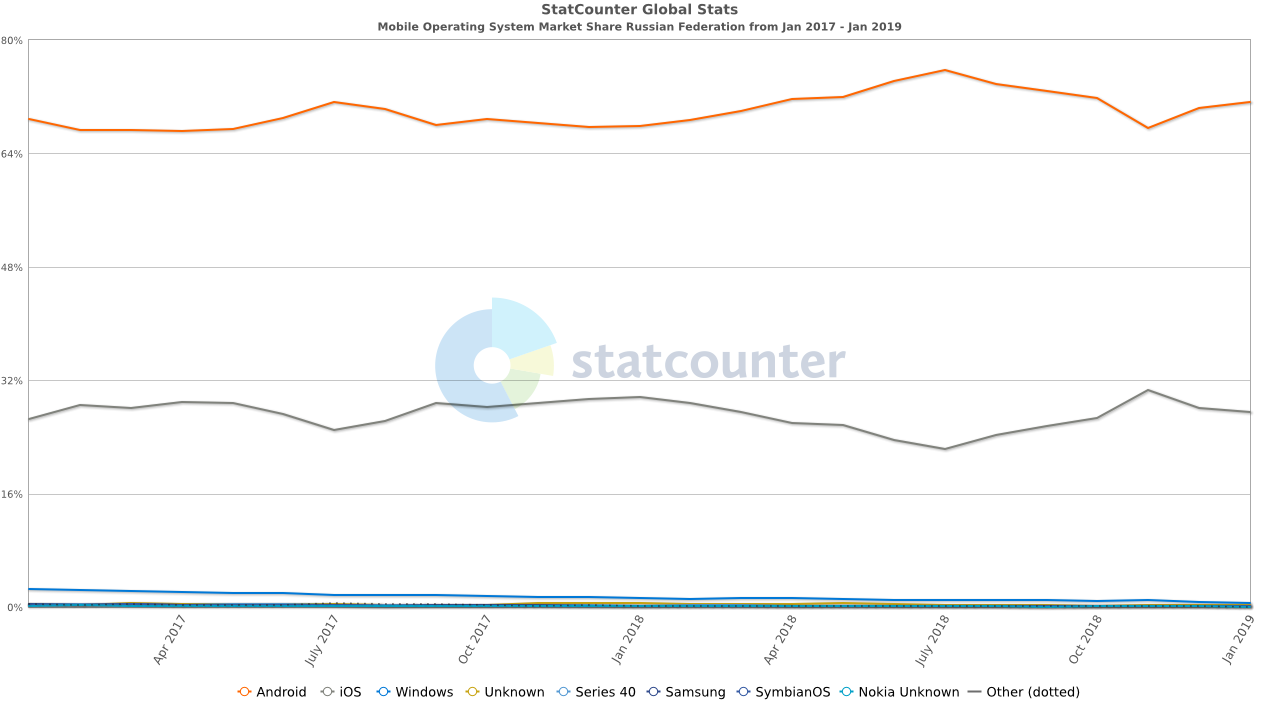


Рис. 1

Самые популярные платформы в мире - ios и android. Выбор пал на android (рис. 2), так как количество смартфонов на этой платформе все же больше. Анализ доступности и удобства разничных IDE помог убедиться в правильности выбора.

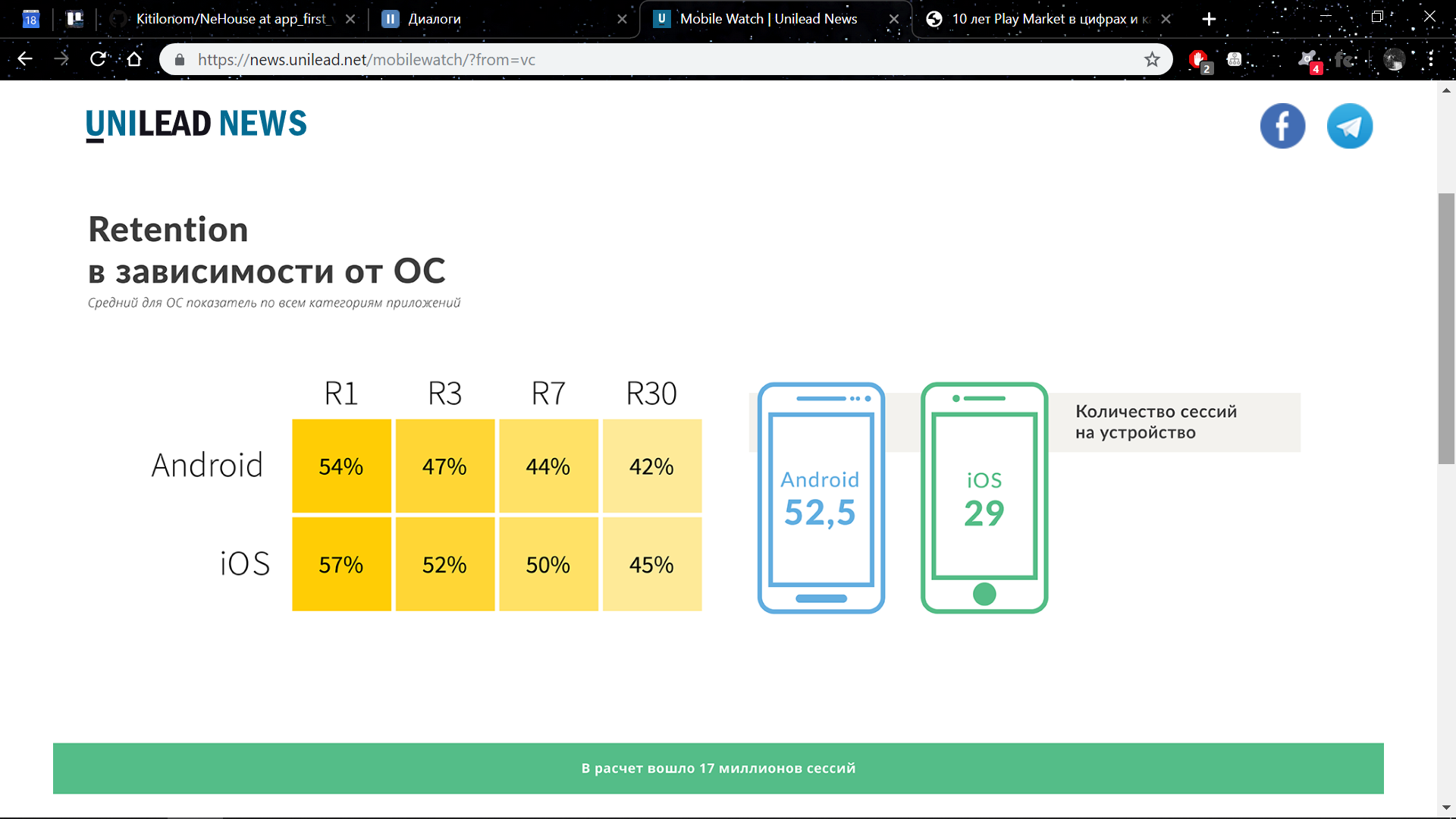


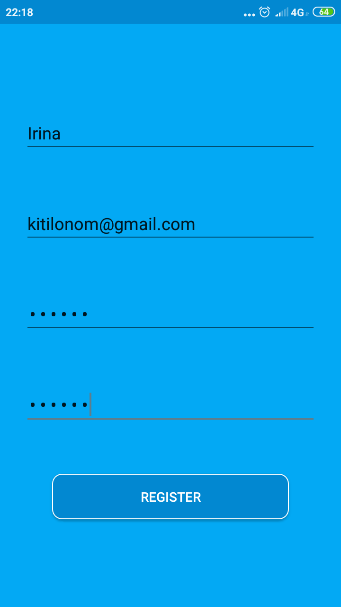
Рис. 2

Следующее, что предстояло изучить – приложения похожего типа, которые уже пользуются успехом на рынке. Мы рассмотрели аналоги на play market.

|  |  |  |
| --- | --- | --- |
| Название приложения | Поддерживаемые платформы | Описание |
| Дребеденьги  https://lh3.ggpht.com/tqjlIJ7Z7qogVCYpbXFeukved2dmRfs8Zfr3brpmG8pqidXPyijDEfa1RCWVL5UBe1I=s180 | * Android * Ios * Winphone * Web | * 549 рублей в год * Поддерживает разные счета * Поддерживает валюты * Интеграция с банком * Индивидуальные траты * Напоминания |
| Дзен-мани  https://is4-ssl.mzstatic.com/image/thumb/Purple124/v4/8a/78/10/8a781017-21bf-ea8e-3f60-a7d48d19c5cc/AppIcon-0-1x_U007emarketing-0-0-85-220-0-10.png/246x0w.jpg | * Android * Ios * Web | * 1249 рублей в год * Поддерживает разные счета * Поддерживает валюты * Интеграция с банком * Индивидуальные траты * Напоминания |
| CoinKeeper: учет расходов  https://is4-ssl.mzstatic.com/image/thumb/Purple124/v4/ea/a7/a5/eaa7a5e1-525e-ae2d-902d-5bf613ec180d/AppIcons-0-1x_U007emarketing-0-85-220-0-6.png/246x0w.jpg | * Android * Ios * Web | * 250 рублей в месяц * Поддерживает разные счета * Поддерживает валюты * Интеграция с банком * Индивидуальные траты * Напоминания |
| Toshl Финансы - учет денег  https://is5-ssl.mzstatic.com/image/thumb/Purple114/v4/3d/7d/06/3d7d0649-1bff-7ea1-4f65-4fe6d30ec89c/AppIcon-0-1x_U007emarketing-0-85-220-0-4.png/246x0w.jpg | * Android * Ios * Winphone * Web | * 19,9$ рублей в год * Поддерживает разные счета * Поддерживает валюты * Интеграция с банком * Индивидуальные траты * Напоминания |
| EasyFinance  https://is2-ssl.mzstatic.com/image/thumb/Purple118/v4/84/82/76/84827609-ad32-57f0-1ab4-841082df3427/mzl.lpditqpc.jpg/246x0w.jpg | * Android * Ios * Web | * 149 рублей в месяц * Поддерживает разные счета * Поддерживает валюты * Интеграция с банком * Индивидуальные траты * Напоминания |
| Alzex Finance  https://is4-ssl.mzstatic.com/image/thumb/Purple114/v4/b1/29/2b/b1292bce-09c2-8644-a81c-1f9ad2a6fe18/AppIcon-0-1x_U007emarketing-0-0-GLES2_U002c0-512MB-sRGB-0-0-0-85-220-0-0-0-10.png/246x0w.jpg | * Android * Ios * Web | * бесплатно * Поддерживает разные счета * Поддерживает валюты * Интеграция с банком * Индивидуальные траты * Напоминания |

# Руководство пользователя.

Пользователю необходимо установить на телефон приложение. Установка производится без дополнительных действий, приложение сразу готово к работе.

Сразу после установки приложение будет предлагать войти в личный аккаунт: ввести свои данные (почта и пароль) и нажать log in. Конечно, если аккаунта у пользователя еще нет, он должен выбрать log on и зарегистрироваться (рис. 4).

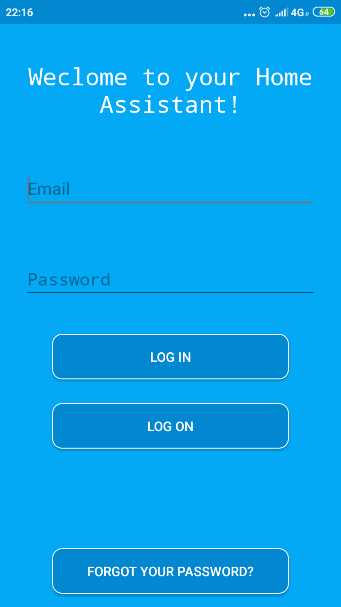


Рис. 3 Вход Рис. 4 Регистрация

Если пользователь только зарегистрировался, то во время входа ему предлагается так же установить группу – выбрать из существующих или создать новую.

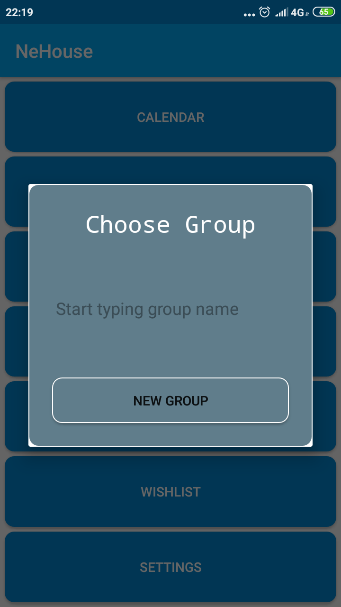
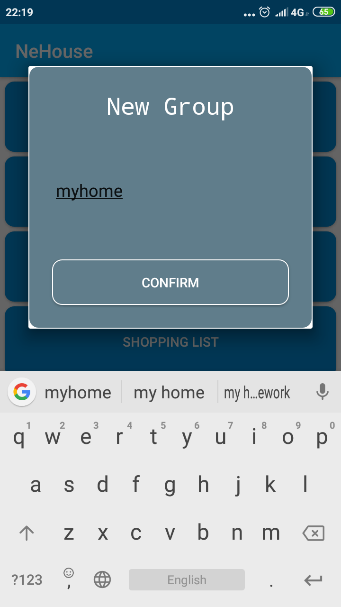
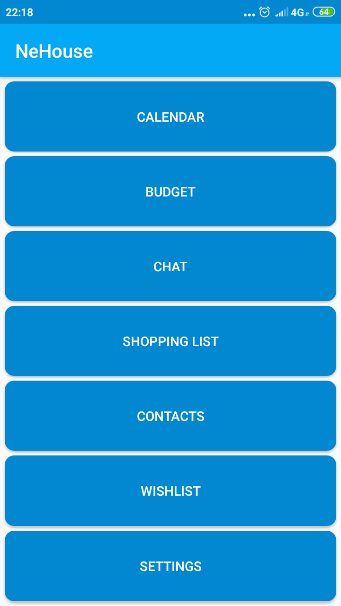


Рис. Рис. Рис.

После выбора группы идентификация пользователя заканчивается и он может использовать все функции приложения. Перед ним открывается главный экран, который содержит следующие разделы: календарь (calendar), бюджет(budget), чат(chat), список покупок(shopping list), контакты(contacts), лист пожеланий(wishlist), настройки(setting).

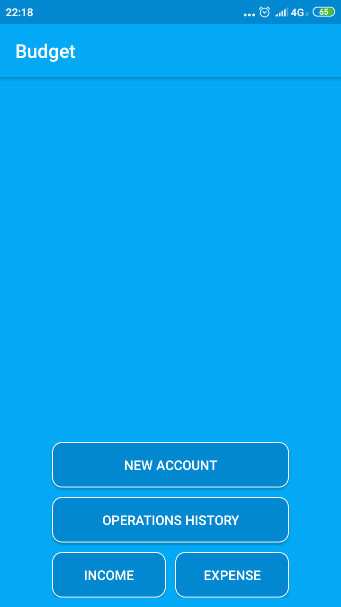
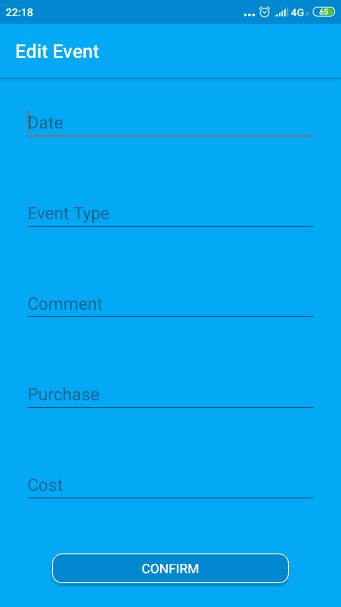
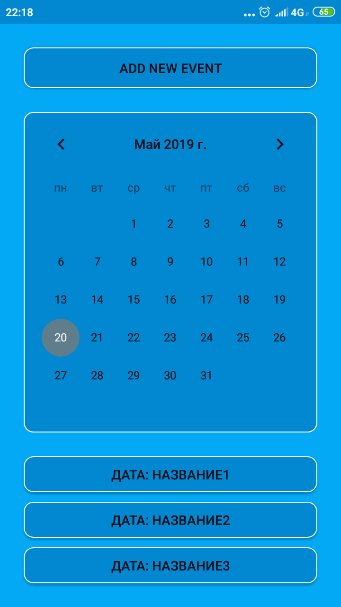


Рис. Рис. Рис.

При переходе на страницу календаря появляется небольшое табло с окном самого календаря, перечнем ближайших дат и мероприятий, кнопка добавления нового события. Для каждого события есть информация о его названии, дате, типе события. Также предусмотрены запланированные траты (поле наименования покупок (purchase) и цены (cost)). Следующий раздел бюджета подробнее показывает работу с денежными операциями.

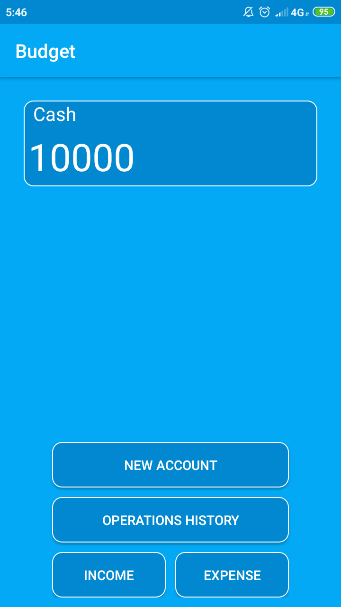
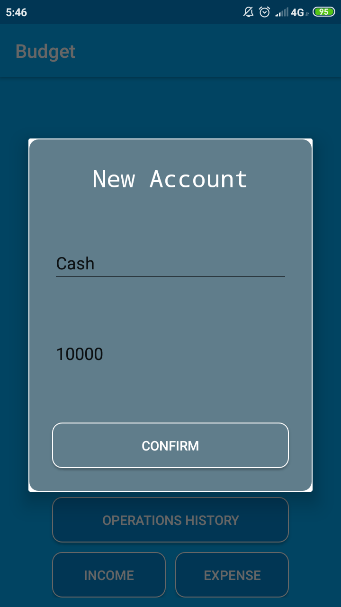
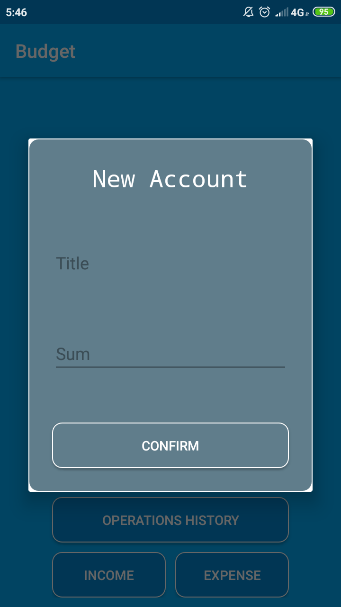


Рис. Рис. Рис.

Экран бюджета представляет собой перечень доступных счетов, каждый из которых содержит в себе информацию о названии счета, сумме, плане расходов и доходов. Можно добавлять неопределенное количество счетов, редактировать их данные, записать доход и расход с комментарием и информацией о владельце.

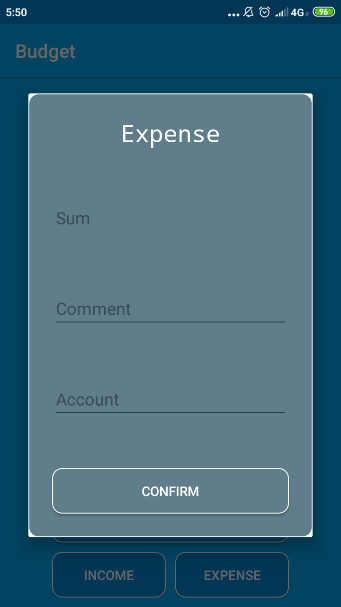
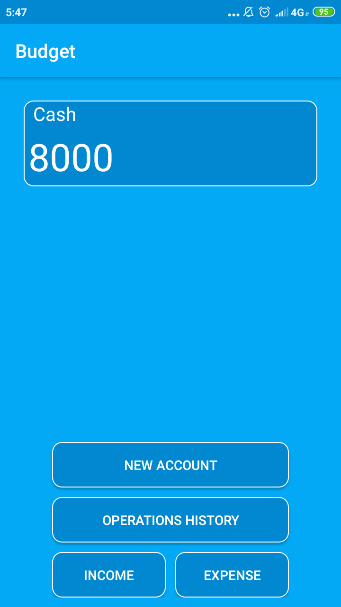
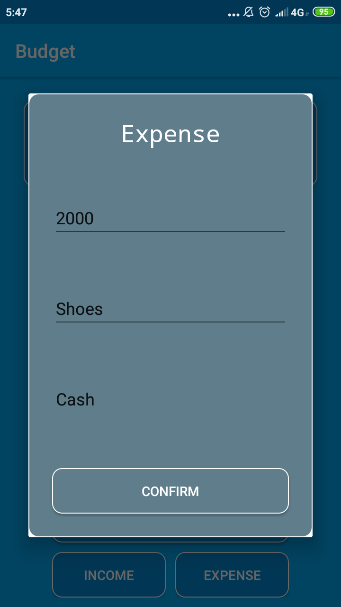


Рис. Рис. Рис.

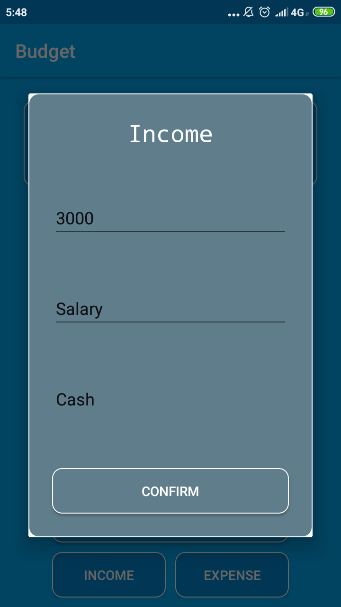
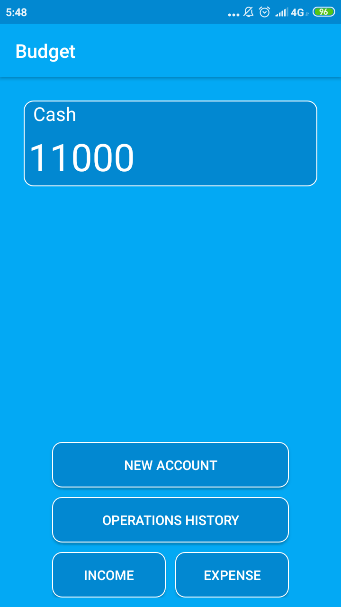
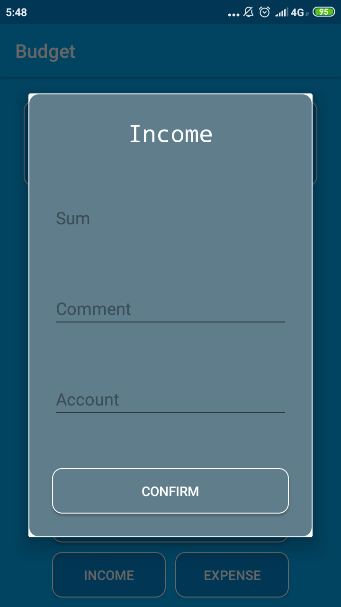


Рис. Рис. Рис.

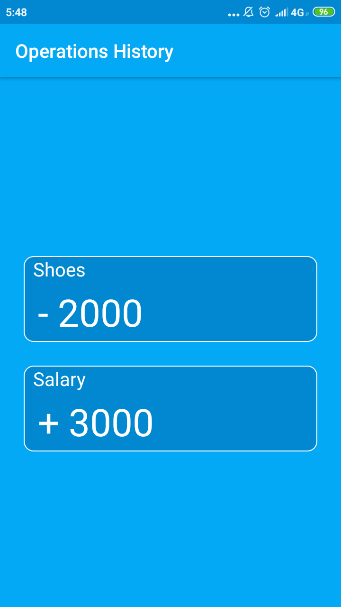


Рис.

Основным преимуществом нашего приложения является чат. Чат представляет собой экран переписки между людьми, находящимися в одной группе. Каждому из участников приходят push уведомления о новых сообщениях.

# Руководство программиста.

## Описание структуры программы.

Как и любое приложение под android, наш продукт базируется на множестве окон и классов, методы которых переопределены в различных вариациях этих классов. Всю работу мы разделили на несколько блоков аналогично техническому заданию. Структура блоков в части «описание структур данных».

Данные о себе хранятся в базе данных о каждом пользователе. Минимальные требования к данным это наличие электронной почты (на одну почту может быть зарегистрирован только один пользователь), пароль, имя, идентификатор группы, к которой принадлежит пользователь. Идентификатор группы задается автоматически, как и идентификатор пользователя, что исключает повторяющиеся ключи и обеспечивает корректное создание ключа для каждого, чьи данные есть в базе.

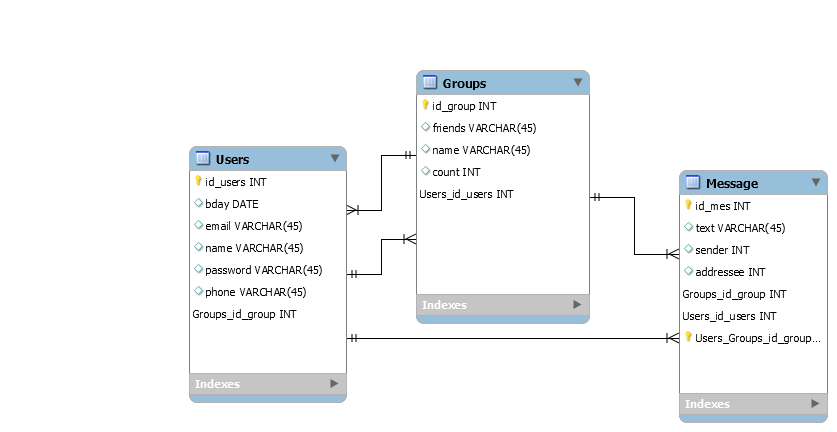


Рис.

Связь с базой данных осуществляется

## Описание структур данных.

Иерархию классов можно отследить на диаграмме ниже:

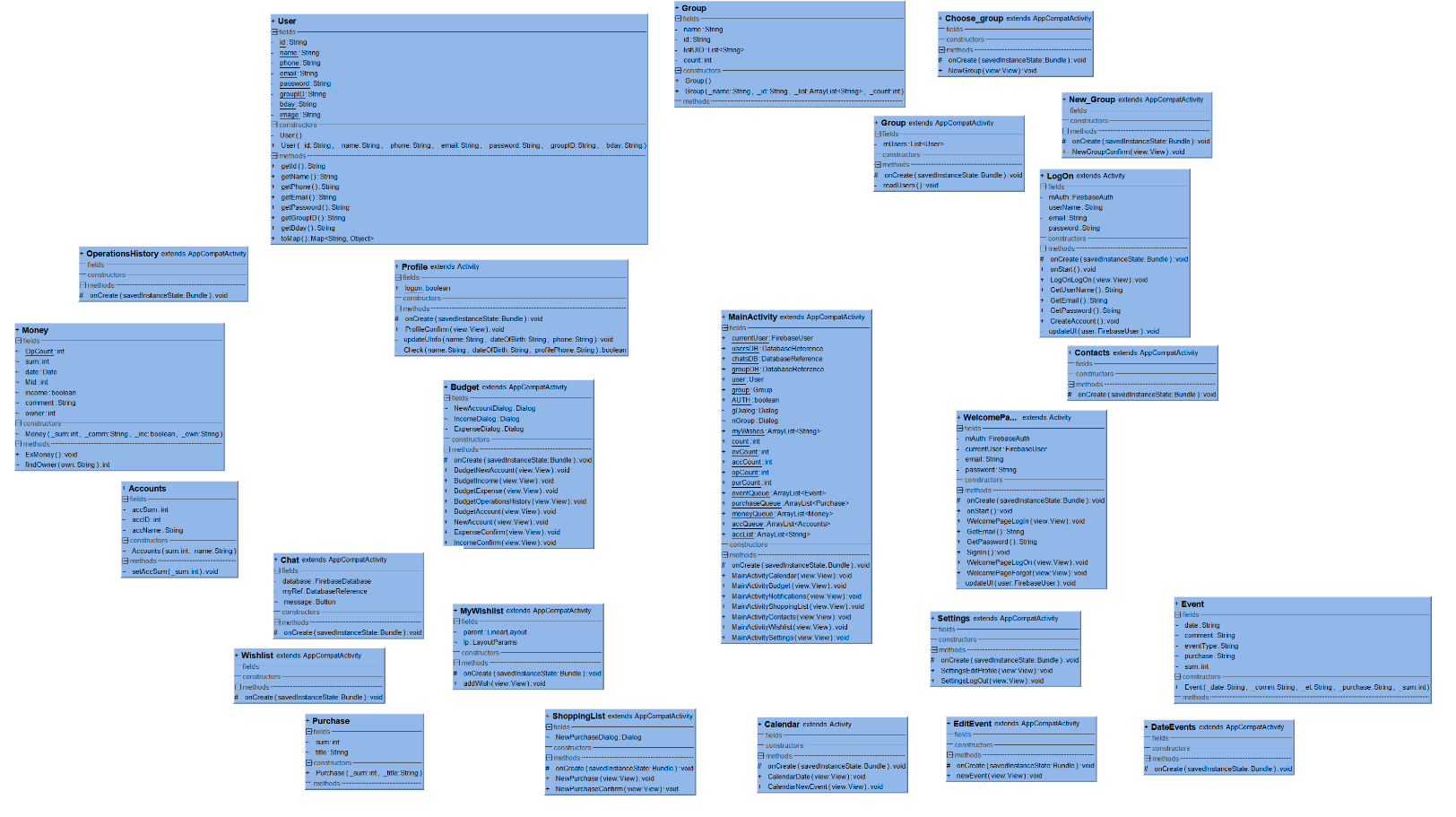


Рис.

Создаваемые классы можно разделить на два типа: описание структуры данных и описание модели экрана. Рассмотрим структуры данных:

Accounts – хранит информацию о счете (название, состояние)

Purchase – хранит информацию о необходимой покупке (название, предполагаемая сумма)

Money – хранит информацию о денежных операциях (сумма, комментарий, счет-владелец)

Event – хранит информацию о событиях (дата, тип события, комментарий, покупка, предполагаемая сумма)

ReadUsers – позволяет собрать данные о пользователе для записи его в группу.

Для удобства использования объекты классов Accounts, Purchase, Money и Events объединены в списки аккаунтов, покупок, историю операций и список событий соответственно (списки типа ArrayList).

Функционал приложения реализован с помощью следующих классов:

**Календарь**

Обеспечивает работу с событиями. Для реализации написаны такие классы как:

* Calendar, DateEvents – связь с xml-файлами(разметка экрана), создание событий
* Events – хранение данных

**Доска бюджета**

Обеспечивает работу со счетами. История операций доступна пользователю списком. Весь бюджет хранится локально и доступен лично каждому, без доступа третьим лицам. Классы:

* Budget, OperationsHistory – связь с xml-файлами, создание счета и денежных операций, просмотр истории операций
* Accounts, Money – хранение данных

**Чат**

Позволяет пользователям обмениваться сообщениями. Классы:

* Сhat – связь с xml

**Список покупок**

Создание списка необходимых приобретений с примерной стоимостью. Классы:

* ShoppingList – связь с xml, создание покупки, просмотр списка
* Purchase – хранение данных

**Контакты**

Просмотр списка участников группы, взаимодействие с ними. Классы:

* Contacts – связь с xml, просмотр списка
* Group – хранение данных

**Желания**

Создание списка желаний, доступного другим пользователям. Классы:

* MyWishes – связь с xml, создание пожеланий

**Настройки**

Просмотр id, изменение личных данных, выход из аккаунта

* Settings – связь с xml, работа с данными

**Авторизация**

Создание личного профиля пользователя, связь с firebase. Классы:

* WelcomePage, LogOn, LogIn – связь с xml, авторизация с использованием баз данных

MainActivity – связь с xml, хранение глобальных данных.

## Описание алгоритмов.

**NewEventConfirm** — создание нового события:

* Собрать введенную пользователем информацию из EditText (дата, тип события, комментарий, покупка и предлагаемая сумма)
* Создать объект класса Event с заданными параметрами и добавить его в список событий
* Создать объект класса Purchase с параметрами покупка и предполагаемая сумма и добавить его в список покупок

**NewAccountConfirm** — создание нового счета:

* Собрать введенную информацию из EditText (название, стартовая сумма)
* Создать объект типа Accounts с заданными параметрами
* Добавить объект в список счетов
* Добавить название элемента в список названий счетов

**ExpenseConfirm** — создание статьи расхода:

* Собрать введенную информацию из EditText (сумма, комментарий, счет-владелец)
* Создать объект типа Money с заданными параметрами
* Добавить объект в список операций
* Изменить состояние счета-владельца в соответствии с расходом

**IncomeConfirm** — создание статьи дохода:

* Собрать введенную информацию из EditText (сумма, комментарий, счет-владелец)
* Создать объект типа Money с заданными параметрами
* Добавить объект в список операций
* Изменить состояние счета-владельца в соответствии с доходом

SignIn -

# Заключение.

Мобильные приложения пользуются успехом во многих сферах, и оптимизация домашнего быта не исключение. В своей работе мы рассмотрели различные android-приложения на рынке, спроектировали свое приложение и создали его, используя язык программирования java, подключив внешний сервер с базой данных firebase от google.

Приложение в итоге получилось достаточно рабочим, чтобы его можно было применять в повседневной жизни, однако содержит в себе множество возможностей для дальнейшего развития. Например, следующее, чем бы мы хотели заняться – развивать идею push уведомлений о покупках. Это может пригодиться не только для удобства пользователей, но и для монетизации ресурса, потому что информация о том, какие потребности возникают у пользователей в зависимости от их местоположения и времени, представляет интерес для различных магазинов.

# Список литературы

1. Голощапов, Алексей Google Android. Программирование для мобильных устройств / Алексей Голощапов. - М.: БХВ-Петербург, 2012. - 448 c.
2. Давыдов, Станислав IntelliJ IDEA. Профессиональное программирование на Java. Наиболее полное руководство (+ CD-ROM) / Станислав Давыдов , Алексей Ефимов. - М.: БХВ-Петербург, **2011**. - 800 c.
3. Дэрси, Лорен Android за 24 часа. Программирование приложений под операционную систему Google / Лорен Дэрси , Шейн Кондер. - М.: Рид Групп, 2011. - 464 c.
4. Камаев А.М., Сиднев А.А., Сысоев А.В. Об одном подходе к анализу эффективности приложений // Труды 50-й научной конференции МФТИ «Современные проблемы фундаментальных и прикладных наук»: Часть I. Радиотехника и кибернетика. - М.: МФТИ, 2007.
5. Касперски К. Техника оптимизации программ. Эффективное использование памяти. — СПб.: БХВ-Петербург, 2003. — 464 с.
6. Мозговой М. В. С++ Мастер-класс. 85 нетривиальных проектов, решений и задач. – Спбю: Наука и техника, 2007. – 272 с.
7. Нотон Java. Справочное руководство. Все, что необходимо для программирования на Java / Нотон, Патрик. - М.: Бином, **2015**. - 448 c.
8. Осипов, Дмитрий Delphi. Программирование для Windows, OS X, iOS и Android / Дмитрий Осипов. - М.: "БХВ-Петербург", 2014. - 464 c.
9. Роджерс, Рик Android. Разработка приложений / Рик Роджерс и др. - М.: ЭКОМ Паблишерз, 2010. - 400 c.
10. Седжевик, Роберт, Уэйн, Кевин. Алгоритмы на Java, 4-е издание.: Пер. с англ. – М.: ООО «И. Д. Вильямс», 2013. – 848 с.
11. Цехнер Марио Программирование игр под Android; Питер - Москва, 2012. - 688 c.

# Приложения.