

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Образовательная программа бакалавриата «Программная инженерия»

СОГЛАСОВАНО
Научный руководитель,
доцент департамента
Программной инженерии

УТВЕРЖДАЮ
Академический руководитель
образовательной программы
«Программная инженерия»

Р.З. Ахметсафина
« ____ » _____ 2020 г.

В. В. Шилов
« ____ » _____ 2020 г.

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ “VOBSHAGE”

Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.04.03-01 81 01-1-ЛУ

Исполнитель

студент группы БПИ197

_____/Бакытбек уулу Н. /

« ____ » _____ 2020 г.

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	RU.17701729.04.03-01 81 01-1-ЛУ

Москва 2020

УТВЕРЖДЕН
RU.17701729.04.03-01 81 01-1

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ “VOBSHAGE”

Пояснительная записка

RU.17701729.04.03-01 81 01-1

Листов 39

<i>Инв. № подл</i>	<i>Подп. и дата</i>	<i>Взам. инв. №</i>	<i>Инв. № дубл.</i>	<i>Подп. и дата</i>
RU.17701729.04.03-01 81 01-1-ЛЮ				

Москва 2020

Оглавление

1	Введение.....	3
1.1	Наименование программы.....	3
1.2	Документы, на основании которых ведется разработка:	3
2	Назначение и область применения.	4
2.1	Назначение программы	4
2.1.1	Функциональное назначение.....	4
2.1.2	Эксплуатационное назначение.....	4
2.2	Краткая характеристика области применения	4
3	Технические характеристики.	5
3.1	Постановка задачи на разработку программы	5
3.2	Описание функционирования программы	5
3.2.1	Описание функционирования различных окон программы.....	5
3.2.2	Описание структуры онлайн базы данных Firebase Realtime Database.....	14
3.2.3	Диаграмма классов.....	17
3.3	Описание и обоснование выбора метода организации входных и выходных данных	18
3.3.1	Описание метода организации входных и выходных данных	18
3.3.2	Обоснование выбора метода организации входных и выходных данных.....	18
3.4	Описание и обоснование выбора состава технических и программных средств.....	19
3.4.1	Состав технических и программных средств	19
4	Ожидаемые технико-экономические показатели	20
4.1	Предполагаемая потребность	20
1.1	Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами.....	20
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	21
	ПРИЛОЖЕНИЕ 1 КАРТА КОДА	23
	ПРИЛОЖЕНИЕ 2 ОПИСАНИЕ КЛАССОВ.....	24
	ПРИЛОЖЕНИЕ 3 ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ МЕТОДОВ И СВОЙСТВ	25
	Лист регистрации изменений.	38

1 Введение.

1.1 Наименование программы:

«Мобильное приложение “vObshage”» (“Mobile Application “ vObshage”).

1.2 Документы, на основании которых ведется разработка:

Приказ декана факультета компьютерных наук И.В. Аржанцева "Об утверждении тем, руководителей курсовых работ студентов образовательной программы «Программная инженерия» факультета компьютерных наук" № 2.3-02/1112-04 от 11.12.2019.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

2 Назначение и область применения.

2.1 Назначение программы

2.1.1 Функциональное назначение

В программе можно составлять расписание уборки квартиры, пользователи могут сами разделить задачи в квартире и многое другое. Также есть возможность составлять личное расписание.

2.1.2 Эксплуатационное назначение

Предназначено для студентов, живущих, в общежитии. Программа позволяет пользователям совместно регулировать порядок в квартире. В квартире живут в среднем 5-6 человек, и не всегда легко найти общий язык, но при этом нужно содержать жилье в чистоте и порядке. Если хотя бы один человек не будет что-то делать, то другие тоже не будут ничего делать. Таким образом, соберётся куча мусора и будет неблагоприятная обстановка в жилье, программа назначено для решения этой проблемы.

2.2 Краткая характеристика области применения

Настоящая программа будет полезным студентам, которые живут в общежитии или проживают совместно в арендованных квартирах. Им необходимо делать задачу по содержанию в чистоте и порядке жилье, покупать товары общего пользования. Программа может использоваться для координации действий при командной работе.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3 Технические характеристики.

3.1 Постановка задачи на разработку программы

Настоящая программа должна содержать следующий функционал:

1. Регистрация нового пользователя или авторизация пользователя.
2. Добавление заданий для общего выполнения.
3. Добавление индивидуальных задач.
4. Уведомление о предстоящих индивидуальных задачах.
5. Отметка выполненных задач в чек-листе.
6. Создание групп пользователей для командной работы.
7. Присоединение к существующей команде.
8. Выделение приоритетности индивидуальных списков задач.
9. Показ приоритетных списков задач.
10. Показ сегодняшних списков задач.
11. Выбор даты и времени для выполнения задач из списка.

3.2 Описание функционирования программы

3.2.1 Описание функционирования различных окон программы

Дизайн каждой страницы можно редактировать в .html файле соответствующей страницы. Для ознакомления с архитектурой проекта смотрите Приложение 1. В ходе разработки программы использовались ресурсы [4]-[12].

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

При запуске программы пользователь попадает в окно авторизации и регистрации (рисунок 1), если пользователь еще не выполнил авторизацию.

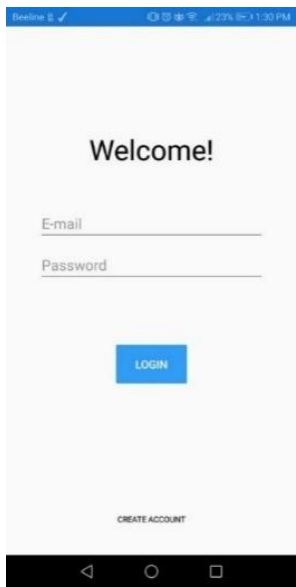


Рисунок 1 - Окно авторизации. Авторизация и регистрация пользователей.

Окно авторизации

Для аутентификации пользователей используется сервис Firebase Authentication. Он предоставляет службы серверной части, понятные пакеты разработчика на любых платформах. Управление учетными записями происходит в личном кабинете Firebase Authentication (см. рисунок 2).

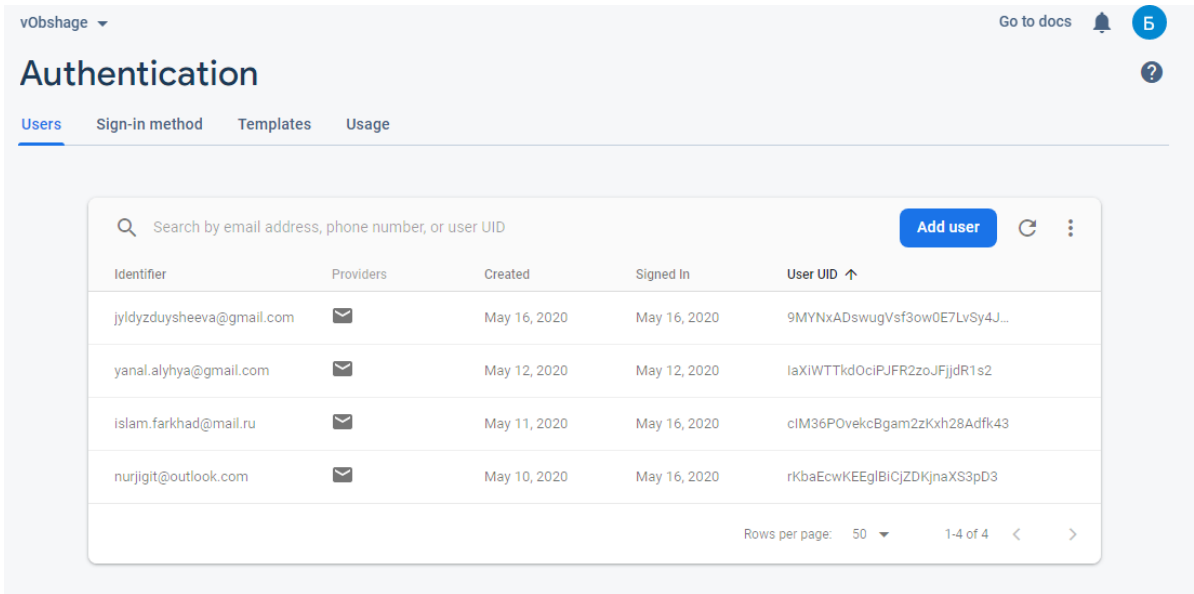


Рисунок 2. Окно управления. Управление учетными записями пользователей.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Пароль автоматически шифруется алгоритмом sCrypt [13] и хранится в личном кабинете. Ключи находятся также в личном кабинете (см. рисунок 3).

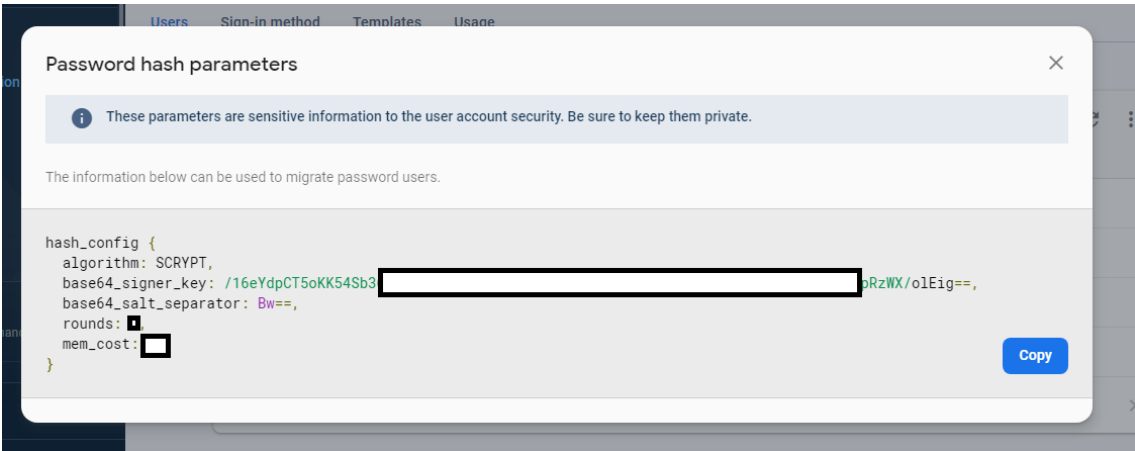


Рисунок 3 – Параметры шифрования пароля.

В окне аутентификации содержится:

- 1. Поле E-mail для ввода электронного адреса.
- 2. Поле Password для ввода пароля.
- 3. Кнопку Login для входа в приложение.
- 4. Кнопку Create Account для создания новой учетной записи.

При нажатии на кнопку Create Account он меняется местами с кнопкой Login.

После авторизации пользователь попадет в окно персональных задач (рисунок 4).

Окно персональных чек-листов

Бекэндом персональной страницы является класс PersonalPage. Каждый индивидуальный чек-лист представлен в виде экземпляра TaskPageModel, который является моделью окно персонального чек-листа (рисунок 4). Коллекция этих экземпляров представлены в виде ObservableCollection<TaskPageModel>. Это необходимо для моментального отображения изменений у пользователя. Коллекция в этом виде автоматически уведомляет об изменениях. Далее не будет указываться это свойство у ObservableCollection<T>, где T – соответствующий класс.

При редактировании, добавлении и удаления персональных чек-листов происходит автоматическая синхронизация с индивидуальным файлом базы данных SQLite, который хранится в локальной памяти устройства пользователя, это дает возможность доступа к персональным чек-листам в режиме офлайн. Структура базы данных представлен в виде единственной таблицы

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

TaskPageModel (Таблица 1). Подробности о столбцах в Приложении 2,3 в описании класса TaskPageModel.

Таблица 1 – Таблица TaskPageModel в файле базы данных SQLite

TaskPageModel		CREATE TABLE "TaskPageModel" ("Notificfld" integer, "Id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "PageTitle" varchar, "Pagelcon" varchar, "IsImportant" integer, "IsRemind" integer, "IsDone" integer, "Date" bigint, "Time" bigint, "ListToString" varchar)
Notificfld	integer	"Notificfld" integer
Id	integer	"Id" integer NOT NULL PRIMARY KEY AUTOINCREMENT
PageTitle	varchar	"PageTitle" varchar
Pagelcon	varchar	"Pagelcon" varchar
IsImportant	integer	"IsImportant" integer
IsRemind	integer	"IsRemind" integer
IsDone	integer	"IsDone" integer
Date	bigint	"Date" bigint
Time	bigint	"Time" bigint
ListToString	varchar	"ListToString" varchar

В окне содержится:

1. Кнопка для отображения панели.
2. Панель содержит кнопки для выбора сортировки чек-листов и для перехода в окно для справок. Панель появляется при нажатии на кнопку для отображения панели (рисунок 5).
3. Список чек-листов пользователя. При нажатии на элемент списка происходит переход в окно выбранного чек-листа.
4. Кнопка для добавления новых чек-листов. При нажатии показывается всплывающее окно для ввода названия (рисунок 6).
5. Панель удаления, редактирования названия и выделения важности чек-листа. Показывается при длительном касании на чек-лист (рисунок 7).

При выполнении всех подзадач чек-листа, чек-лист выделяется зеленым цветом. Если пользователь выделил чек-лист важным, то отображается звездочка рядом с названием.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

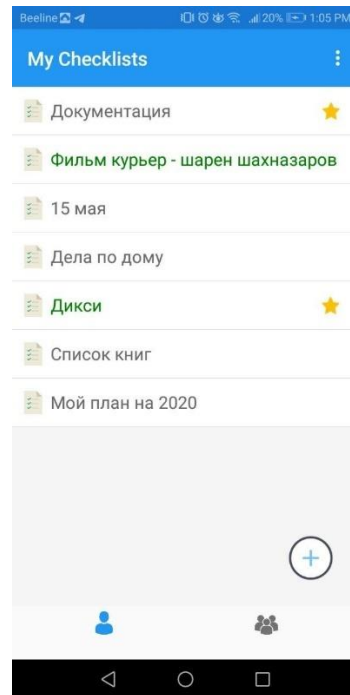


Рисунок 4 - Окно персональных задач. Отображаются персональные чек-листы.

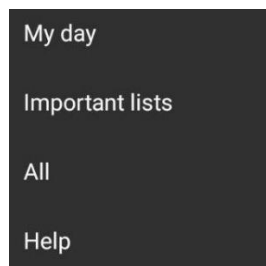


Рисунок 5 - Панель персонального окна

New title

Enter list's title

Enter here

CANCEL CREATE NEW LIST

Рисунок 6 - Поле для ввода текста. Текст названия, сообщения и кнопок могут быть различными.



Рисунок 7 - Панель чек-листа. Удаление, редактирование и выделение чек-листа.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Окно персонального чек-листа

Бекэндом данного окна является класс `ModalTasks`. Модель чек-листа представлен в виде экземпляра класса `TaskPageModel`. При редактировании, добавлении и удаления задач в персональный чек-лист происходит автоматическая синхронизация с индивидуальным файлом базы данных `SQLite`, который хранится в локальной памяти устройства пользователя, это дает возможность доступа к персональным чек-листам в режиме офлайн.

Каждая задача представлен в виде экземпляра класса `TaskItem`, каждая из которых находится в коллекции типа `ObservableCollection<TaskItem>`, который находится модели чек-листа. Важно отметить, что эта коллекция не хранится в базе данных. Вместо него хранится строка, в котором находится эта сериализованная коллекция, это необходимо для быстрой работы базы данных.

В этом окне можно включить уведомление, после выбора корректной даты и времени. Это достигается при помощи класса `LocalNotificationService`, который работает с системным менеджером уведомлений устройства пользователя.

Страница для отображения персонального чек-листа (рисунок 8) содержит:

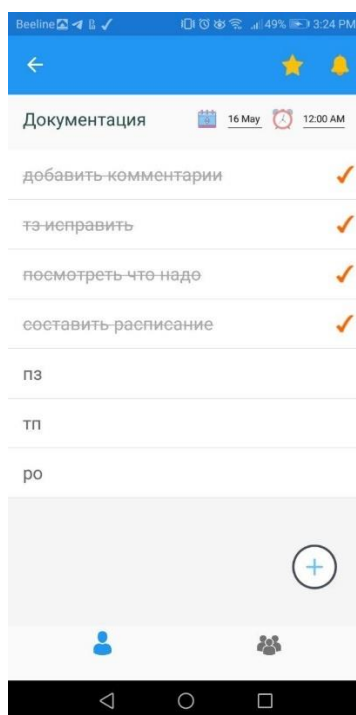


Рисунок 8 – Окно персонального чек-листа. Добавление и выполнение задач в чек-листе.

1. Кнопка для выделения или отмены выделения важности.
2. Кнопка для включения или отключения уведомлений.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3. Кнопка для выбора даты. При нажатии отображается элемент для выбора даты (рисунок 9).

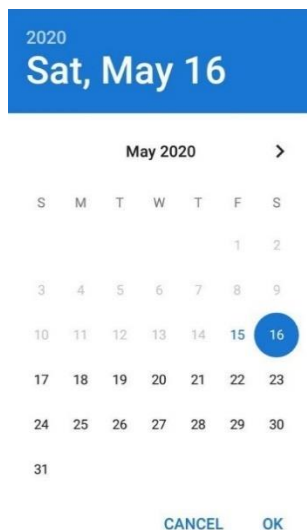


Рисунок 9 – Средство для выбора даты

4. Кнопка для выбора времени. При нажатии отображается элемент для выбора даты (рисунок 10).

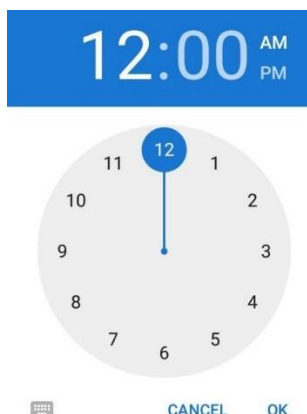


Рисунок 10 – Средство для выбора времени

5. Кнопка для добавления задач в чек-лист. При нажатии появляется поле (рисунок 6) для ввода текста задачи.
6. Список задач чек-листа. При нажатии текст задачи перечерчивается и рядом появляется галочка. При длительном нажатии появляется панель для редактирования, копирования и удаления задач чек-листа (рисунок 11).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата



Рисунок 11 – Панель задачи чек-листа

Окно для отображения команд пользователя

Бекэнд страницы представлен классом GeneralPage, который взаимодействует с онлайн базой данных Firebase Realtime Database. После появления на экран происходит загрузка групп пользователя, идентификаторы которых находится в массиве строк экземпляра класса User из онлайн базы данных. При создании, присоединении и редактировании названия группы происходит автоматическая синхронизация с онлайн базой данных.

Окно для отображения команд пользователя (рисунок 12) содержит:

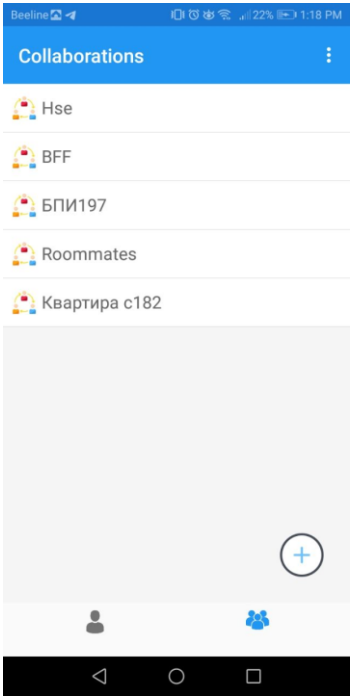


Рисунок 12 – Окно команд. Отображает команды пользователя

- 1. Кнопку для отображения панели для выхода и перехода страницы справок (рисунок 13).

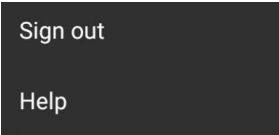


Рисунок 13 - Панель для выхода и перехода в страницу для справок

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- 2. Список команд пользователя. При нажатии происходит переход в окно чек-листа команды.
- 3. Кнопка создания команды или присоединения к существующей команде. При нажатии показывается элемент для выбора (рисунок 14).

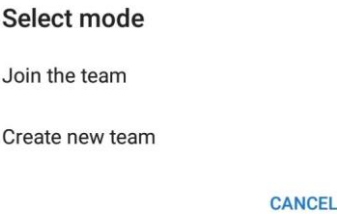


Рисунок 14 – Элемент для выбора. Создание или присоединение к команде

- 4. Поле для ввода идентификатора команды или названия (рисунок 6).

Окно чек-листов команды

Бекэнд представлен в виде класса TeamsPage, при появлении страницы происходит загрузка чек-листов, представленные в виде экземпляра класса CheckList из онлайн базы данных Firebase Realtime Database по идентификаторам, которые хранятся в виде массива в экземпляре класса Team.

При добавлении, удалении и редактирования названия команды, происходит автоматическая синхронизация с онлайн базой данных.

При появлении страницы подписывается метод для обновления данных к узлу соответствующей команды в онлайн базе данных. Это обеспечивается при помощи экземпляра класса CheckListListener.

Окно чек-листов команды (рисунок 15) содержит:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

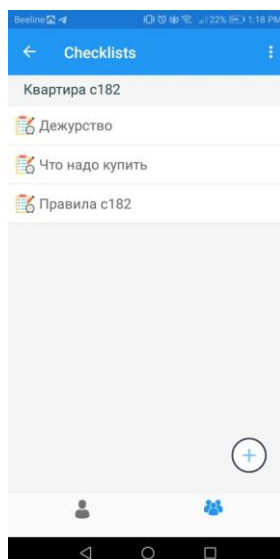


Рисунок 15 – Окно чек-листов команды. Содержит чек-листы команды.

1. Список чек-листов команды. При нажатии происходит переход в окно с командным чек-листом.
2. Кнопка для добавления нового чек-листа команды. При нажатии появляется поле для ввода названия команды (рисунок 6).
3. Панель для копирования идентификатора команды.

3.2.2 Описание структуры онлайн базы данных Firebase Realtime Database

Имеется три главных узла:

1. Users -каждый подузел представлен в виде сериализованного Json объекта класса User (рисунок 16). Это узел для хранения данных о пользователях.

Таблица 1 – Описание структуры узла Users.

Название	Назначение
Id	Идентификатор
Name	Имя
TeamsId	Команды пользователя
Token	Токен для быстрых запросов

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

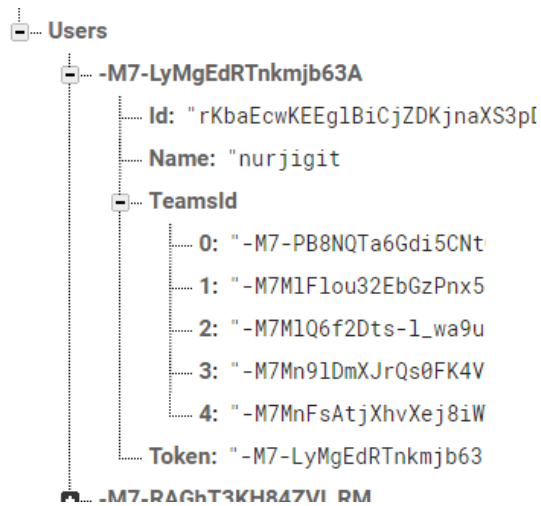


Рисунок 16 – Структура узла Users

2. CheckLists - каждый подузел представлен в виде сериализованного Json объекта класса CheckList (рисунок 17). Это узел для хранения чек-листов.

Таблица 2 – Описание узла CheckLists

Название	Назначение
Date	Дата
Id	Идентификатор
Tasks	Текст задания и его результат
Time	Время для выполнения
Title	Заголовок

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

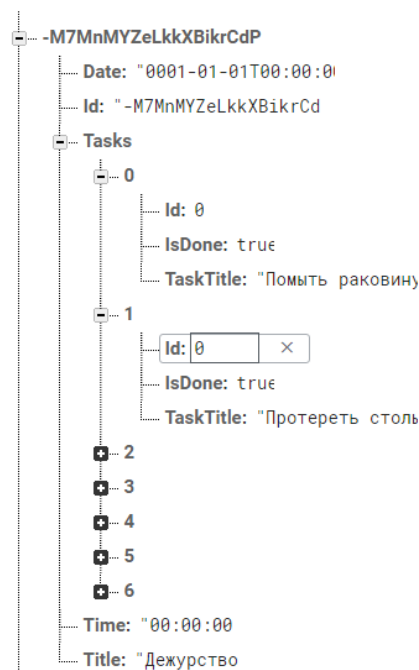


Рисунок 17 – Структура узла CheckLists

3. Teams - каждый подузел представлен в виде сериализованного Json объекта класса Team (рисунок 18). Это узел для хранения команд.

Таблица 3 – Описание узла Teams

Название	Назначение
CheckListsId	Идентификаторы чек-листов команды
Id	Идентификтор
MembersTokens	Токены пользователей
Name	Название команды

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата



Рисунок 18 – Структура Teams

3.2.3 Диаграмма классов

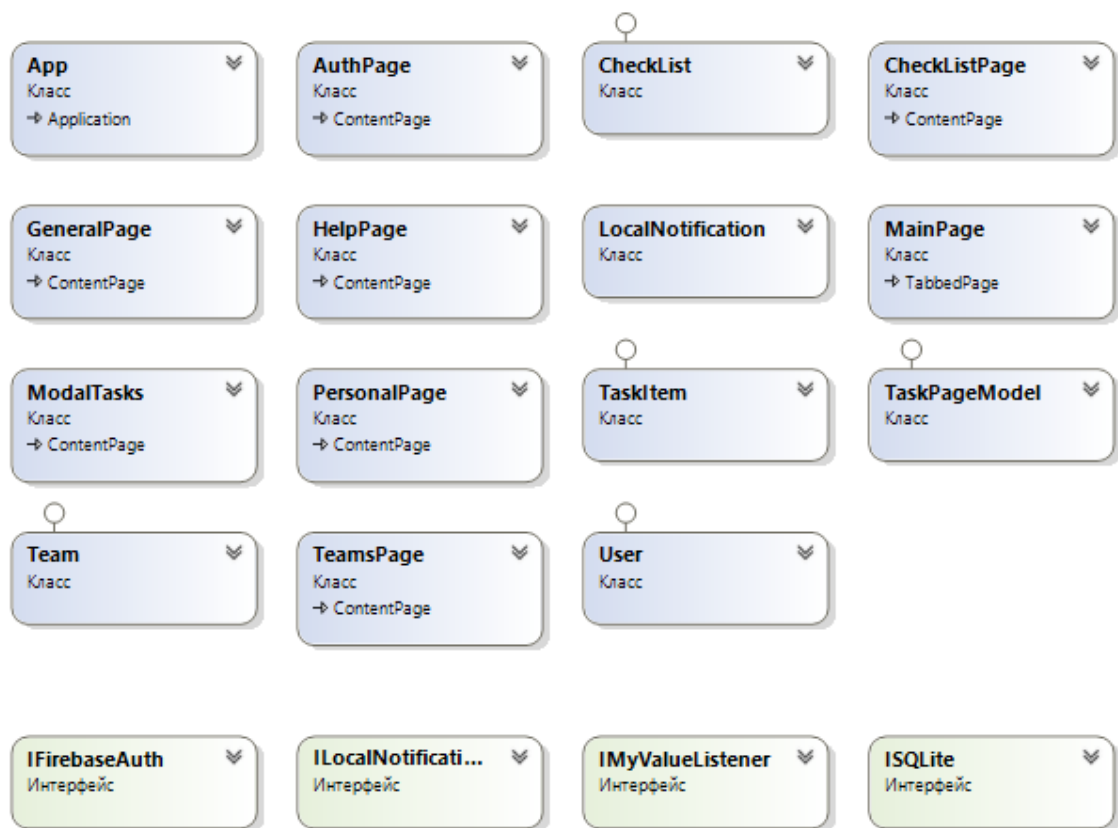


Рисунок 19 – Диаграмма классов

На данном рисунке приведена диаграмма классов Portable Class Library проекта Xamarin Forms. Описание классов приведены в Приложении 1,2,3.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3.3 Описание и обоснование выбора метода организации входных и выходных данных

3.3.1 Описание метода организации входных и выходных данных

Входными данными являются действия пользователей (нажатия на кнопки), тексты, получаемые из полей ввода и информации, получаемы от СУБД (Firebase Realtime Database) и локального файла базы данных SQLite. Общая схема на рисунке 20. Связи яввяются двухсторонними.

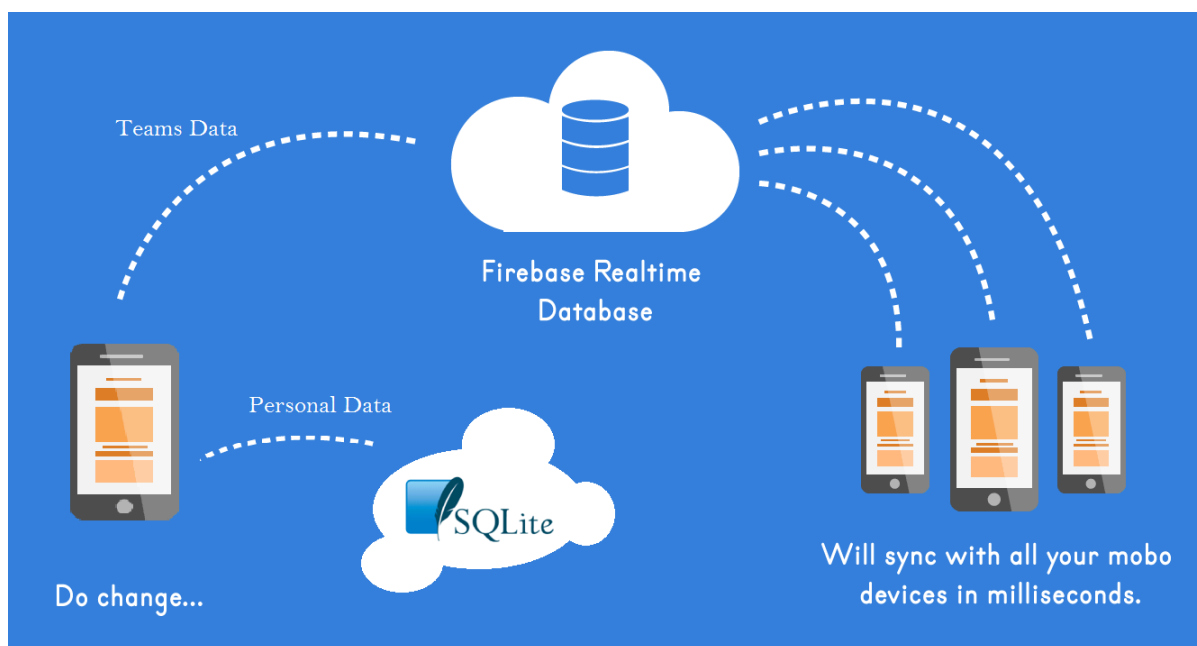


Рисунок 20 – Схема обмена данными

Выходными данными являются файл базы данных SQLite, хранящийся в локальном устройстве пользователя, а также информации, отправляемые в СУБД (Firebase Realtime Database).

3.3.2 Обоснование выбора метода организации входных и выходных данных

База данных SQLite выбран из следующих соображений:

1. Файловая структура – все данные хранятся в единственном файле.
2. Кроссплатформенность – отлично подходит для проектов Xamarin Forms.
3. Простота и скорость использования базы данных.

Облачная СУБД Firebase Realtime Database выбран из следующих соображений:

1. Позволяет хранить и синхронизировать данные пользователей в режиме реального времени.
2. Данные имеют вид Json – позволяет быстро отправлять и получать информацию.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3. Нет необходимости создания сервера.
4. Надежная пользовательская безопасность.

3.4 Описание и обоснование выбора состава технических и программных средств

3.4.1 Состав технических и программных средств

Для функционирования программы требуется смартфон с интернетом со следующими характеристиками:

1. Количество ядер - 4
2. Частота работы процессора – 1.4 ГГц
3. Объем оперативной памяти - 2 ГБ
4. Объем встроенной памяти - 16 ГБ
5. ОС Android 10.0 и выше.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

4 Ожидаемые технико-экономические показатели

4.1 Предполагаемая потребность

Приложение в первую очередь ориентировано студентам. Благодаря данной программе будут сохранены благоприятная обстановка и хорошие взаимоотношения между студентами, что необходимо для благополучного проживания в общежитии.

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

На рынке Play Market существуют много различных аналогов, самые яркие из них:

1. “Trello: Organize anything with anyone, anywhere!”. Trello, Inc.
2. “Focus To-Do: Pomodoro Timer & To Do List”. Pomodoro Timer & To Do List – SuperElement Soft.

Преимуществами программы являются:

1. Является бесплатным.
2. Удобный интерфейс.
3. Возможность работы с другими пользователями.
4. Не требует денежных затрат.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

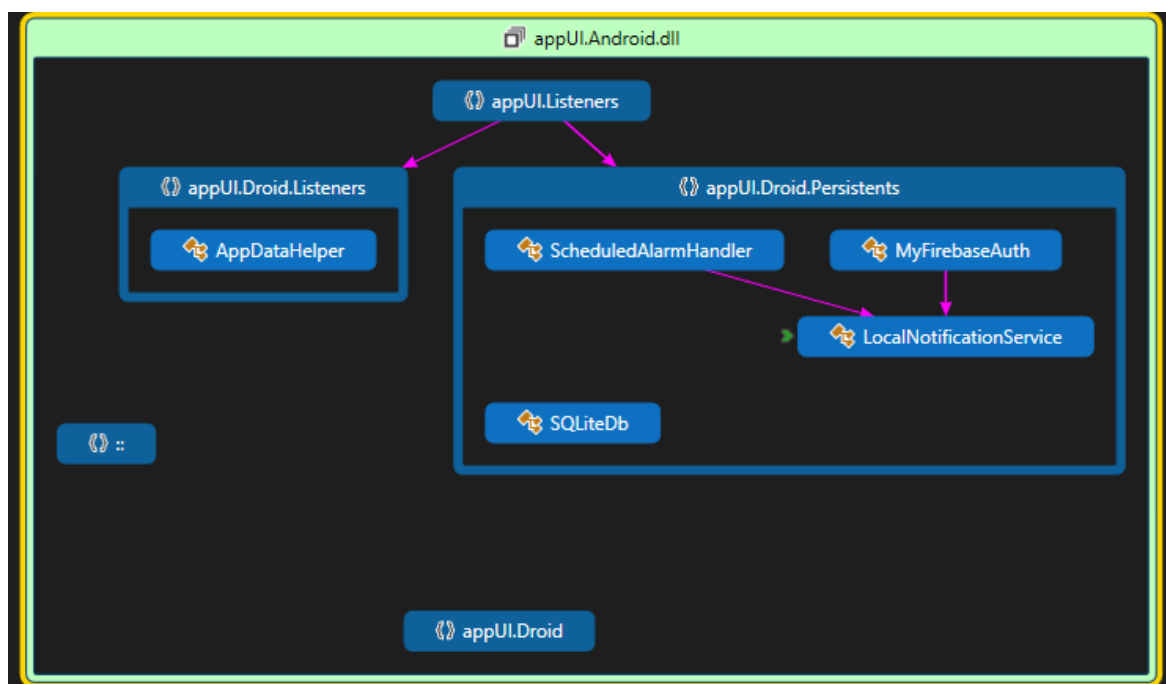
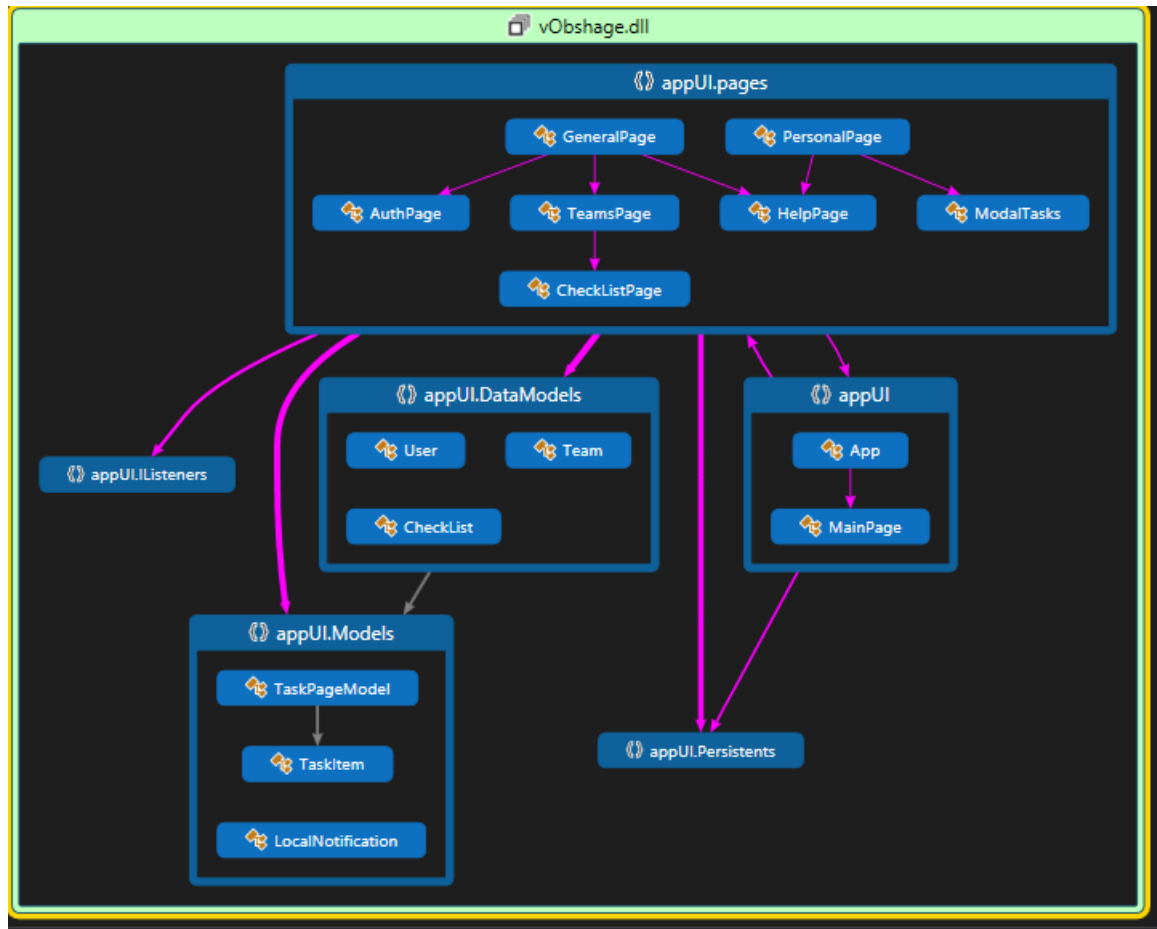
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 19.101-77 Виды программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. Создание мобильных приложений с помощью Xamarin.Forms (книга) [Электронный ресурс]// URL: <https://docs.microsoft.com/ru-ru/xamarin/xamarin-forms/creating-mobile-apps-xamarin-forms/> (Дата обращения 8.4.2020, режим доступа: свободный).
5. How To Send Local Notification With A Repeat Interval (Day/Minute/Hour etc.) [Электронный ресурс]// URL: <https://www.c-sharpcorner.com/article/how-to-send-local-notification-with-repeat-interval-in-xamarin-forms/> (Дата обращения 28.4.2020, режим доступа: свободный).
6. How to make a Floating Action Button in Xamarin Forms [Электронный ресурс]// URL: <https://stackoverflow.com/questions/54517874/how-to-make-a-floating-action-button-in-xamarin-forms> (Дата обращения 15.4.2020, режим доступа: свободный).
7. How can I make my app send out notifications when it's loaded but not running in the foreground? [Электронный ресурс]// URL: <https://stackoverflow.com/questions/59622443/how-can-i-make-my-app-send-out-notifications-when-its-loaded-but-not-running-in> (Дата обращения 17.4.2020, режим доступа: свободный).
8. Vertically expand ListView inside an AbsoluteLayout [Электронный ресурс]// URL: <https://forums.xamarin.com/discussion/37897/vertically-expand-listview-inside-an-absolutelayout> (Дата обращения 22.4.2020, режим доступа: свободный).
9. Toast equivalent for Xamarin Forms [Электронный ресурс]// URL: <https://stackoverflow.com/questions/35279403/toast-equivalent-for-xamarin-forms> (Дата обращения 23.4.2020, режим доступа: свободный).
10. Firebase Cloud Messaging [Электронный ресурс]// URL: <https://docs.microsoft.com/ru-ru/xamarin/android/data-cloud/google-messaging/firebase-cloud-messaging> (Дата обращения 3.5.2020, режим доступа: свободный).

11. Firebase.Xamarin [Электронный ресурс]// URL:
<https://github.com/rlamasb/Firebase.Xamarin/blob/master/README.md> (Дата
обращения 6.5.2020, режим доступа: свободный).
12. Xamarin.Forms - Working with Firebase Realtime Database CRUD Operations
[Электронный ресурс]// URL:
<https://xamarinmonkeys.blogspot.com/2019/01/xamarinforms-working-with-firebase.html>
(Дата обращения 9.5.2020, режим доступа: свободный).
13. Firebase Authentication Password Hashing [Электронный ресурс]// URL:
<https://firebaseopensource.com/projects/firebase/scrypt/> (Дата обращения 11.5.2020,
режим доступа: свободный).

ПРИЛОЖЕНИЕ 1

КАРТА КОДА



ПРИЛОЖЕНИЕ 2
ОПИСАНИЕ КЛАССОВ

Таблица П1 - Описание и функциональное назначение классов

Класс	Назначение
CheckList	Модель чек-листа.
Team	Модель команды.
User	Модель пользователя.
LocalNotification	Класс для локальных уведомлений.
TaskItem	Модель задачи.
TaskPageModel	Модель страницы ModalTasks.
AuthPage	Страница для регистрации\авторизации пользователя.
CheckListPage	Страницы командного чек-листа.
GeneralPage	Страница для работы с командами.
HelpPage	Страница с подсказками.
ModalTasks	Страница персонального чек-листа.
PersonalPage	Страница с персональными чек-листами.
TeamsPage	Страница с чек-листами.
AppDataHelper	Класс для работы с базой данных Firebase Realtime Database.
CheckListListener	Класс для получения изменений на подписанный узел.
LocalNotificationService	Класс для оповещения пользователя.
MyFirebaseAuth	Класс для регистрации\авторизации пользователей.
SQLiteDatabase	Класс для получения доступа к локальной базе данных.

ПРИЛОЖЕНИЕ 3

ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ МЕТОДОВ И СВОЙСТВ

Таблица П2 - Описание полей, методов и свойств класса CheckList.cs

Поля				
Имя	Модификатор доступа	Тип	Назначение	
PropertyChangedEventHandler	public	event	Оповещает об изменении свойств.	
resourceName	public static readonly	string	Название узла в онлайн базе данных Firebase database.	
title	private	string	Заголовок чек-листа.	
id	private	string	Идентификатор чек-листа.	
date	private	DateTime	Дата чек-лиса.	
time	private	TimeSpan	Время чек-листа.	
Свойства				
Имя	Модификатор	Тип	Доступ	Назначение
Title	public	string	get, set	Свойство доступа к title. Оповещает об изменениях.
Id	public	string	get, set	Свойство доступа к id. Оповещает об изменениях.
Date	public	DateTime	get, set	Свойство доступа к date. Оповещает об изменениях.
Time	public	TimeSpan	get, set	Свойство доступа к time. Оповещает об изменениях.
Tasks	public	ObservableCollection<TaskItem>	get, set	Задачи чек-листа. Оповещает об изменениях.
IsRemind	public	bool	get, set;	Свойство доступа к isRemind. Оповещает об изменениях.
RemindIcon	public	string	get	Возвращает картинку в зависимости от состояния уведомлений.
NotificfId	public	int?	get, set	Идентификатор уведомлений.
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
SetProperty	protected	void	string name	Оповещение об изменении определенного свойства.

Таблица ПЗ - Описание полей, методов и свойств класса Team

Поля				
Имя	Модификатор доступа	Тип	Назначение	
resourceName	public static readonly	string	Название узла в онлайн базе данных Firebase database.	
id	private	string	Идентификатор команды	
name	private	string	Имя команды	
PropertyChanged	public PropertyChangedEventHandler	event	Оповещает об изменении свойств. Оповещает об изменениях.	
Свойства				
Имя	Модификатор	Тип	Доступ	Назначение
MembersToken	public	List<string>	get, set	Идентификатор-токены членов команды.
CheckListsId	public	List<string>	get, set	Идентификатор чек-листов команды.
Name	public	string	get, set	Свойство для доступа name. Оповещает об изменениях.
Id	public	string	get, set	Свойство для доступа к id. Оповещает об изменениях.
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
SetProperty	protected	void	string name	Оповещение об изменении определенного свойства.

Таблица П4 - Описание полей, методов и свойств класса User

Поля			
Имя	Модификатор доступа	Тип	Назначение
name	private	string	Имя пользователя
id	private	string	Идентификатор пользователя

Продолжение таблицы П4

PropertyChanged	public PropertyChangedEventHandler	event	Оповещает об изменении свойств. Оповещает об изменениях.	
Свойства				
Name	public	string	get, set	Свойство для доступа name. Оповещает об изменениях.
Id	public	string	get, set	Свойство для доступа к id. Оповещает об изменениях.
Token	public	string	get, set	Токен пользователя.
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
SetProperty	protected	void	string name	Оповещение об изменении определенного свойства.

Таблица П5 - Описание полей, методов и свойств класса AuthPage

Поля				
Имя	Модификатор доступа	Тип	Назначение	
auth	private readonly	IFirebaseAuth	Интерфейс для взаимодействия с Firebase Authentication SDK.	
userPath	private readonly	string	Узел пользователей в базе данных Firebase Realtime Database.	
notificationManager	private readonly	ILocalNotificationService	Интерфейс для оповещений.	
fbUrl	private readonly	string	Ссылка для доступа к базе данных.	
firebase	private readonly	FirebaseClient	Предоставляет доступ к онлайн базе данных.	
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
AuthPage	public	Конструктор	-	Установка начальных значений
LoginBtn_Clicked	private async	void	object sender, EventArgs e	Вход и регистрация в системе Firebase Authentication SDK.
NewAccountBtn_Clicked	private	void	object sender, EventArgs e	Меняет функции авторизации\входа.

Таблица П6 - Описание полей, методов и свойств класса CheckListPage

Поля				
Имя	Модификатор доступа	Тип	Назначение	
checlistsPath	private readonly	string	Узел пользователей в базе данных Firebase Realtime Database.	
notification	private readonly	ILocalNotificationService	Интерфейс для оповещений.	
fbUrl	private readonly	string	Ссылка для доступа к базе данных.	
firebase	private readonly	FirebaseClient	Предоставляет доступ к онлайн базе данных.	
CurrentCheckList	private	CheckList	Модель страницы.	
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
CheckListPage	public	Конструктор	CheckList checkList	Установка начальных значений
OnAppearing	protected override	void	-	Подписка на события.
SaveCurrentCheckList	private async	void	-	Сохраняет привязанный чек-лист в базе данных.
PlusFab_Clicked	private async	void	object sender, EventArgs e	Добавляет задачу в чек-лист.
Input	private async	Task<string>	string title, string message, string placeholder, string accept, string initialValue	Метод для ввода строки.
TasksList_ItemSelected	private	void	object sender, SelectedItemChangedEventArgs e	Метод срабатывает, когда происходит нажатие на элемент списка. Нужно, чтобы цветом не выделялся выбранный элемент.
TasksList_ItemTapped	private	void	object sender, ItemTappedEventArgs e	Метод срабатывает, когда происходит нажатие на элемент списка. Отмечает, что задание выполнено \ не выполнено.

Продолжение таблицы П6

EditTask_Clicked	private async	void	object sender, EventArgs e	Метод для редактирования текста задачи.
DeleteTask_Clicked	private async	void	object sender, EventArgs e	Метод для удаления задачи из листа.
TasksList_Refreshing	private async	void	object sender, EventArgs e	Обновление листа.
MyDatePicker_DateSelected	private	void	object sender, DateTimeEventArgs e	Срабатывает после выбора даты.
MyTimePicker_Unfocused	private	void	object sender, FocusEventArgs e	Срабатывает при закрытии окна выбора времени.

Таблица П7 - Описание полей, методов и свойств класса GeneralPage

Поля			
Имя	Модификатор доступа	Тип	Назначение
auth	private readonly	IFirebaseAuth	Интерфейс для взаимодействия с Firebase Authentication SDK.
userPath	private readonly	string	Узел пользователей в базе данных Firebase Realtime Database.
notification	private readonly	ILocalNotificationService	Интерфейс для оповещений.
fbUrl	private readonly	string	Ссылка для доступа к базе данных.
firebase	private readonly	FirebaseClient	Предоставляет доступ к онлайн базе данных.
teamPath	private readonly	string	Название узла команд в онлайн базе данных.
CurrentUser	private	User	Текущий пользователь.
Teams	private	ObservableCollection<Team>	Команды пользователя.

Продолжение таблицы П7

Методы				
Название	Модификатор	Тип	Аргументы	Назначение
Input	private async	Task<string>	string title, string message, string placeholder, string accept, string initialValue	Метод для ввода строки.
OnAppearing	protected async override	void	-	Задание "тяжелых" начальных значений.
GetTeamsAsync	private async	Task<Team[]>	(List<string> ids	Получение команд из онлайн базы данных.
PlusFab_Clicked	private async	void	object sender, EventArgs e	Добавление\создание команд.
GeneralPage	public	Конструктор	-	Задание начальных значений.
LogOut_Clicked	private	void	object sender, EventArgs e	Выход из системы авторизации\регистрации.
MainList_ItemSelected	private	void	object sender, SelectedItemChangedEventArgs e	Метод срабатывает, когда происходит нажатие на элемент списка. Нужно, чтобы цветом не выделялся выбранный элемент.
MainList_ItemTapped	private async	void	object sender, ItemTappedEventArgs e	Метод срабатывает, когда происходит нажатие на элемент списка. Отмечает, что задание выполнено \ не выполнено.
TeamEdit_Clicked	private async	void	object sender, EventArgs e	Метод для редактирования названия команды.
TeamDelete_Clicked	private async	void	object sender, EventArgs e	Метод для удаления команды.
CopyId_Clicked	private async	void	object sender, EventArgs e	Копирование идентификатора команды.
HelpTb_Clicked	private	void		Показ окна с подсказками.

Таблица П8 Описание полей, методов и свойств класса HelpPage

Методы				
Название	Модификатор	Тип	Аргументы	Назначение
HelpPage	public	Конструктор	-	Установка начальных значений

Таблица П9 - Описание полей, методов и свойств класса TeamsPage

Поля				
Имя	Модификатор доступа	Тип	Назначение	
auth	private readonly	IFirebaseAuth	Интерфейс для взаимодействия с Firebase Authentication SDK.	
userPath	private readonly	string	Узел пользователей в базе данных Firebase Realtime Database.	
notification	private readonly	ILocalNotificationService	Интерфейс для оповещений.	
fbUrl	private readonly	string	Ссылка для доступа к базе данных.	
firebase	private readonly	FirebaseClient	Предоставляет доступ к онлайн базе данных.	
teamPath	private readonly	string	Название узла команд в онлайн базе данных.	
CurrentUser	private	User	Текущий пользователь.	
CheckLists	private	ObservableCollection<CheckList>	Чек-листы команды.	
CurrentTeam	private	Team	Текущая команда.	
checlistsPath	private readonly	string	Название узла команд в онлайн базе данных.	
checlistsListener	private readonly	IMyValueListener	Интерфейс для определения изменения данных.	
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
Input	private async	Task<string>	string title, string message, string placeholder, string accept, string initialValue	Метод для ввода строки.

Продолжение таблицы П9

OnAppearing	protected async override	void	-	Задание "тяжелых" начальных значений.
GetListsAsync	private async	Task<CheckList[]>	-	Получение чек-листов из онлайн базы данных по их идентификаторам.
PlusFab_Clicked	private async	void	object sender, EventArgs e	Добавление нового чек-листа.
MainList_ItemSelected	private	void	object sender, SelectedItemChangedEventArgs e	Метод срабатывает, когда происходит нажатие на элемент списка. Нужно, чтобы цветом не выделялся выбранный элемент.
MainList_ItemTapped	private async	void	object sender, ItemTappedEventArgs e	Показ страницы с чек-листом.
EditCheckList_Clicked	private async	void	object sender, EventArgs e	Редактирование названия чек-листа.
DeleteCheckList_Clicked	private async	void	object sender, EventArgs e	Удаление чек-листа.
MainList_Refreshing	private async	void	object sender, EventArgs e	Обновление чек-листов.
CopyId_Clicked	private async	void	object sender, EventArgs e	Копирование идентификатора команды.

Таблица П10 - Описание полей, методов и свойств класса ModalTasks

Поля				
Имя	Модификатор доступа	Тип	Назначение	
connection	private readonly	SQLiteAsyncC onnection	Предоставляет асинхронное соединение с локальной базой данных.	
notification	private readonly	ILocalNotificat ionService	Интерфейс для оповещений.	
Свойства				
Model	public	TaskPageM odel	get, private set	Модель страницы
Id	public	string	get, set	Свойство для доступа к id. Оповещает об изменениях.

Продолжение таблицы П10

Методы				
Название	Модификатор	Тип	Аргументы	Назначение
ModalTasks	public	Конструктор	TaskPageModel pageModel, SQLiteAsync Connection connection	Задание начальных значений.
OnAppearing	protected override	void	-	Задание "тяжелых" начальных значений.
TasksList_ItemTapped	private async	void	object sender, ItemTappedEventArgs e	Отметка\снятие галочки о выполнении.
Edit_Clicked	async private	void	object sender, EventArgs e	Метод для редактирования текста задачи.
Delete_Clicked	private async	void	object sender, EventArgs e	Метод для удаления задачи из листа.
Fabplus_Clicked	private async	void	object sender, EventArgs e	Добавляет задачу в чек-лист.
RemaindTb_Clicked	private async	void	object sender, EventArgs e	Включение\отключение уведомлений.
StarTb_Clicked	private async	void	object sender, EventArgs e	Устанавливает приоритетность чек-листа.
TasksList_ItemSelected	private	void	object sender, SelectedItemChangedEventArgs e	Метод срабатывает, когда происходит нажатие на элемент списка. Нужно, чтобы цветом не выделялся выбранный элемент.
CopyId_Clicked	private async	void	object sender, EventArgs e	Копирует текст задания.

Таблица П11 - Описание полей, методов и свойств класса PersonalPage

Поля			
Имя	Модификатор доступа	Тип	Назначение
sqConnection	private readonly	SQLiteAsyncConnection	Предоставляет асинхронное соединение с локальной базой данных.
notification	private readonly	ILocalNotificationService	Интерфейс для оповещений.
auth	private readonly	IFirebaseAuth	Интерфейс для взаимодействия с Firebase Authentication SDK.

Продолжение таблицы П11

Свойства				
Имя	Модификатор	Тип	Доступ	Назначение
Models	public	Observable Collection< TaskPageM odel>	get, private set	Модели чек-листов.
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
PersonalPage	public	Конструкт ор	-	Задание начальных значений.
OnAppearing	protected override	void	-	Задание "тяжелых" начальных значений.
PlusFab_Clic ked	private async	void	object sender, EventArgs e	Добавление персонального чек-листа.
MainList_It emTapped	private async	void	object sender, ItemTappedE ventArgs e	Показ страницы с чек-листом.
Important_Cl icked	private	void	object sender, EventArgs e	Устанавливает приоритетность чек-листу.
Delete_Click ed	private async	void	object sender, EventArgs e	Удаление чек-листа.
Edit_Clicked	private async	void	object sender, EventArgs e	Редактирование названия чек-листа.
MyDay_Clic ked	private async	void	object sender, EventArgs e	Показывает сегодняшние чек-листы.
ImportantList s_Clicked	private async	void	object sender, EventArgs e	Показывает приоритетные чек-листы.
All_Clicked	private	void	object sender, EventArgs e	Показывает все чек-листы.
MainList_It emSelected	private	void	object sender, SelectedItem ChangedEve ntArgs e	Метод срабатывает, когда происходит нажатие на элемент списка. Нужно, чтобы цветом не выделялся выбранный элемент.
HelpTb_Clic ked	private	void	object sender, EventArgs e	Показывает страницу с подсказками.

Таблица П12 - Описание полей, методов и свойств класса AppDataHelper

Методы				
Название	Модификатор	Тип	Аргументы	Назначение
GetDatabase	public static	FirestoreDatabase	-	Возвращает экземпляр для работы с базой данных Firebase Realtime Database.

Таблица П12 - Описание полей, методов и свойств класса PersonalPage

Поля				
Имя	Модификатор доступа	Тип	Назначение	
OnChange	public	event EventHandler	Событие для получения уведомлений об изменении в определенном узле базы данных.	
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
OnDataChange	public	void	DataSnapshot snapshot	Получение оповещения об изменении данных.
Subscribe	public	void	string path	Устанавливает данный объект к узлу в базе данных, чтобы он оповещал о изменениях.

Таблица П13 - Описание полей, методов и свойств класса SQLiteDatabase

Поля				
Имя	Модификатор доступа	Тип	Назначение	
path	private readonly	string	Путь для сохранения файла базы данных.	
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
GetConnectionPersonal	public	SQLiteAsyncConnection	string id	Возвращает соединение для работы с базой данных.

Таблица П14- Описание полей, методов и свойств класса MyFirebaseAuth

Методы				
Название	Модификатор	Тип	Аргументы	Назначение
LoginWithEmailPassword	public async	Task<string>	string email, string password	Авторизация пользователя.
CreateWithEmailPassword	public async	Task<string>	string email, string password	Регистрация нового пользователя.
IsCurrentUser	public	bool	-	Определяет выполнил ли пользователь вход.
GetUserID	public	string	-	Возвращает уникальный идентификатор пользователя.
SignOut	public	void	-	Выполняет выход учетной записи из системы.

Таблица П15 - Описание полей, методов и свойств класса LocalNotificationService

Поля				
Имя	Модификатор доступа	Тип	Назначение	
_jan1st1970	private readonly	DateTime	Unix-время	
_randomNumber	internal	string	Для задания уникального идентификатора.	
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
LocalNotification	public	void	string title, string body, int id, DateTime notifyTime, long repeateMs	Установка локальных уведомлений на устройстве.
Cancel	public	void	int id	Отмена уведомления
GetLauncherActivity	public static	Intent	-	Возвращает "хороший" механизм для выполнения задач в переднем плане.
CreateIntent	private	Intent	int id	Возвращает подзадачу, используется для показа уведомлений.
GetAlarmManager	private	AlarmManager	-	Возвращает системный сервис для управления сигналами.

Продолжение таблицы П15

SerializeNotification	private	AlarmManager	LocalNotification notification	Сериализует уведомление.
LongAlert	public	void	string message	Показ длительного Toast уведомления.
ShortAlert	public	void	string message	Показ кратковременного Toast уведомления.

Таблица П16 - Описание полей, методов и свойств класса ScheduledAlarmHandler

Поля				
Имя	Модификатор доступа	Тип	Назначение	
manager	private	NotificationManager	Уведомление пользователя о событии.	
LocalNotificationKey	public const	string	Ключ-константа.	
channelId	private const	string	Идентификатор канала.	
channelName	private const	string	Имя канала.	
channelDescription	private const	string	Описание канала.	
channelInitialized	private	bool	Флаг инициализации канала.	
Методы				
Название	Модификатор	Тип	Аргументы	Назначение
OnReceive	public override	void	Context context, Intent intent	Этот метод вызывается, когда ScheduledAlarmHandler получает механизм для вещания.
CreateNotificationChannel	private	void	-	Создание канала для оповещения.
DeserializeNotification	private	LocalNotification	string notificationString	Десериализация оповещения.

[illegible]