# Руководство пользователя и описание принципа работы мобильного приложения «Органайзер».

Оглавление

[Руководство пользователя и описание принципа работы мобильного приложения «Органайзер». 1](#_Toc171622166)

[Описание основных файлов, классов и функций программы «Органайзер». 3](#_Toc171622167)

[1. AdminAddUserActivity.kt 3](#_Toc171622168)

[2. AdminEditUserActivity.kt 3](#_Toc171622169)

[3. AdminListOfUserActivity.kt 3](#_Toc171622170)

[4. FirebaseHelper.kt 3](#_Toc171622171)

[5. LogInActivity.kt 4](#_Toc171622172)

[6. AdminAddRoleActivity.kt 4](#_Toc171622173)

[7. UserCheckboxAdapter.kt 4](#_Toc171622174)

[8. UserCheckProjectActivity.kt 4](#_Toc171622175)

[9. UserCheckTaskActivity.kt 5](#_Toc171622176)

[10. ChatAdapter.kt 5](#_Toc171622177)

[11. CompletedTaskAdapter.kt 5](#_Toc171622178)

[12. LogInAuthorizedActivity.kt 6](#_Toc171622179)

[13. ProjectTaskAdapter.kt 6](#_Toc171622180)

[14. UserAdapter.kt 6](#_Toc171622181)

[15. UserAddProjectActivity.kt 6](#_Toc171622182)

[16. UserAddTaskActivity.kt 7](#_Toc171622183)

[17. UserEndTaskActivity.kt 7](#_Toc171622184)

[18. UserShowProjectActivity.kt 7](#_Toc171622185)

[19. UserTaskChatActivity.kt 7](#_Toc171622186)

[20. UserCompletedTasksActivity.kt 8](#_Toc171622187)

[Описание основных макетов активностей программы «Органайзер». 9](#_Toc171622188)

[1. activity\_admin\_add\_user.xml 9](#_Toc171622189)

[2. activity\_admin\_edit\_user.xml 9](#_Toc171622190)

[3. activity\_admin\_list\_of\_user.xml 9](#_Toc171622191)

[4. activity\_admin\_add\_role.xml 10](#_Toc171622192)

[5. activity\_log\_in\_authorized.xml 10](#_Toc171622193)

[6. activity\_login.xml 10](#_Toc171622194)

[7. activity\_user\_add\_project.xml 10](#_Toc171622195)

[8. activity\_user\_add\_task.xml 11](#_Toc171622196)

[9. activity\_user\_check\_project.xml 11](#_Toc171622197)

[10. activity\_user\_check\_task.xml 11](#_Toc171622198)

[11. activity\_user\_end\_task.xml 11](#_Toc171622199)

[12. activity\_user\_show\_project.xml 12](#_Toc171622200)

[13. activity\_user\_task\_chat.xml 12](#_Toc171622201)

[14. activity\_user\_completed\_tasks.xml 12](#_Toc171622202)

[15. item\_project\_header.xml 12](#_Toc171622203)

[16. item\_task.xml 13](#_Toc171622204)

[17. item\_message\_sent.xml 13](#_Toc171622205)

[18. item\_message\_received.xml 13](#_Toc171622206)

[19. item\_user.xml 13](#_Toc171622207)

[20. item\_completed\_task.xml 14](#_Toc171622208)

[21. item\_user\_checkbox.xml 14](#_Toc171622209)

[Структура данных Firebase приложения «Органайзер» 15](#_Toc171622210)

[1. endTasks 15](#_Toc171622211)

[2. projects 15](#_Toc171622212)

[3. roles 15](#_Toc171622213)

[4. tasks 15](#_Toc171622214)

[5. users 15](#_Toc171622215)

[Руководство пользователя приложения «Органайзер» 16](#_Toc171622216)

[Введение 16](#_Toc171622217)

[Основные функции 16](#_Toc171622218)

[Вход в систему 16](#_Toc171622219)

[Интерфейс администратора 16](#_Toc171622220)

[Данные входа администратора логин admin пароль 768312 16](#_Toc171622221)

[Интерфейс пользователя 17](#_Toc171622222)

[Завершение работы 17](#_Toc171622223)

[Возможные проблемы и их решение 17](#_Toc171622224)

# Описание основных файлов, классов и функций программы «Органайзер».

1. AdminAddUserActivity.kt Класс: AdminAddUserActivity

**Функциональность:** Этот класс отвечает за интерфейс добавления нового пользователя в систему. Использует Firebase для хранения данных пользователей.

**Основные методы:**

* onCreate(): Инициализация элементов интерфейса и установка слушателей для кнопок.
* addUserToFirebase(): Добавление нового пользователя в Firebase.

2. AdminEditUserActivity.kt Класс: AdminEditUserActivity

**Функциональность:** Этот класс предоставляет интерфейс для редактирования существующих пользователей. Использует Firebase для получения и обновления данных пользователей.

**Основные методы:**

* onCreate(): Инициализация элементов интерфейса и установка слушателей для кнопок.
* loadUserData(): Загрузка данных пользователя из Firebase.
* updateUserInFirebase(): Обновление данных пользователя в Firebase.

3. AdminListOfUserActivity.kt Класс: AdminListOfUserActivity

**Функциональность:** Этот класс отображает список всех пользователей. Позволяет администратору просматривать и редактировать данные пользователей.

**Основные методы:**

* onCreate(): Инициализация элементов интерфейса и установка слушателей для кнопок.
* loadUsers(): Загрузка списка пользователей из Firebase.
* setupListView(): Настройка ListView для отображения пользователей.

4. FirebaseHelper.kt Класс: FirebaseHelper

**Функциональность:** Этот класс содержит вспомогательные методы для работы с Firebase. Предоставляет методы для чтения и записи данных в Firebase. Разработан для оптимизации.

**Основные методы:**

* readDatabase(callback: () -> Unit): Чтение данных из Firebase и вызов callback-функции.
* writeDatabase(path: String, value: Any): Запись данных в Firebase по указанному пути.

5. LogInActivity.kt Класс: LogInActivity

**Функциональность:** Этот класс отвечает за интерфейс авторизации пользователей. Проверяет учетные данные пользователей и перенаправляет их в соответствующий интерфейс (администратор или пользователь). При запуске программы пользователем является первым экраном.

**Основные методы:**

* onCreate(): Инициализация элементов интерфейса и установка слушателей для кнопок.
* logInUser(): Проверка учетных данных пользователя и авторизация.

6. AdminAddRoleActivity.kt Класс: AdminAddRoleActivity

**Функциональность:** Этот класс предоставляет интерфейс для добавления новых ролей в систему. Использует Firebase для хранения данных о ролях.

**Основные методы:**

* onCreate(): Инициализация элементов интерфейса и установка слушателей для кнопок.
* addRoleToFirebase(): Добавление новой роли в Firebase.

7. UserCheckboxAdapter.kt Класс: UserCheckboxAdapter

**Функциональность:** Этот класс отвечает за отображение списка пользователей с возможностью выбора (checkbox) для назначения на задачу или проект.

**Основные методы:**

* getCount(): Возвращает количество элементов в адаптере.
* getItem(position: Int): Возвращает элемент по указанной позиции.
* getView(position: Int, convertView: View?, parent: ViewGroup?): Создает и возвращает вид для элемента списка.

8. UserCheckProjectActivity.kt Класс: UserCheckProjectActivity

**Функциональность:** Этот класс предоставляет интерфейс для просмотра информации о проекте и связанных задачах.

**Основные методы:**

* onCreate(): Инициализация элементов интерфейса и установка слушателей для кнопок.
* loadProjectDetails(projectId: Int): Загрузка данных проекта из Firebase.
* loadProjectTasks(taskIds: List<Int>): Загрузка задач, связанных с проектом.

9. UserCheckTaskActivity.kt Класс: UserCheckTaskActivity

**Функциональность:** Этот класс предоставляет интерфейс для просмотра и редактирования информации о задаче.

**Основные методы:**

* onCreate(): Инициализация элементов интерфейса и установка слушателей для кнопок.
* loadTaskDetails(taskId: Int): Загрузка данных задачи из Firebase.
* loadTaskUsers(userIds: List<Int>): Загрузка пользователей, назначенных на задачу.

10. ChatAdapter.kt Класс: ChatAdapter

**Функциональность:** Этот класс отвечает за отображение сообщений чата в задачах.

**Основные методы:**

* getCount(): Возвращает количество сообщений в адаптере.
* getItem(position: Int): Возвращает сообщение по указанной позиции.
* getView(position: Int, convertView: View?, parent: ViewGroup?): Создает и возвращает вид для сообщения.

11. CompletedTaskAdapter.kt Класс: CompletedTaskAdapter

**Функциональность:** Этот класс отвечает за отображение списка завершенных задач.

**Основные методы:**

* getCount(): Возвращает количество завершенных задач в адаптере.
* getItem(position: Int): Возвращает завершенную задачу по указанной позиции.
* getView(position: Int, convertView: View?, parent: ViewGroup?): Создает и возвращает вид для завершенной задачи.

12. LogInAuthorizedActivity.kt Класс: LogInAuthorizedActivity

**Функциональность:** Этот класс отвечает за интерфейс, который отображается после успешной авторизации пользователя.

**Основные методы:**

* onCreate(): Инициализация элементов интерфейса и установка слушателей для кнопок.

13. ProjectTaskAdapter.kt Класс: ProjectTaskAdapter

**Функциональность:** Этот класс отвечает за отображение списка проектов и связанных с ними задач.

**Основные методы:**

* getCount(): Возвращает количество элементов (проектов и задач) в адаптере.
* getItem(position: Int): Возвращает элемент (проект или задачу) по указанной позиции.
* getView(position: Int, convertView: View?, parent: ViewGroup?): Создает и возвращает вид для элемента (проекта или задачи).

14. UserAdapter.kt Класс: UserAdapter

**Функциональность:** Этот класс отвечает за отображение списка пользователей в различных частях приложения.

**Основные методы:**

* getCount(): Возвращает количество пользователей в адаптере.
* getItem(position: Int): Возвращает пользователя по указанной позиции.
* getView(position: Int, convertView: View?, parent: ViewGroup?): Создает и возвращает вид для пользователя.

15. UserAddProjectActivity.kt Класс: UserAddProjectActivity

**Функциональность:** Этот класс предоставляет интерфейс для добавления нового проекта.

**Основные методы:**

* onCreate(): Инициализация элементов интерфейса и установка слушателей для кнопок.
* createProject(): Создание нового проекта и добавление его в Firebase.

16. UserAddTaskActivity.kt Класс: UserAddTaskActivity

**Функциональность:** Этот класс предоставляет интерфейс для добавления новой задачи.

**Основные методы:**

* onCreate(): Инициализация элементов интерфейса и установка слушателей для кнопок.
* createTask(): Создание новой задачи и добавление ее в Firebase.

17. UserEndTaskActivity.kt Класс: UserEndTaskActivity

**Функциональность:** Этот класс предоставляет интерфейс для завершения задачи.

**Основные методы:**

* onCreate(): Инициализация элементов интерфейса и установка слушателей для кнопок.
* endTask(): Завершение задачи и обновление статуса в Firebase.

18. UserShowProjectActivity.kt Класс: UserShowProjectActivity

**Функциональность:** Этот класс отображает список проектов и связанных с ними задач для пользователя.

**Основные методы:**

* onCreate(): Инициализация элементов интерфейса и установка слушателей для кнопок.
* loadProjectsAndTasks(): Загрузка проектов и задач из Firebase.
* loadProjects(projectIds: List<Int>): Загрузка данных проектов.
* loadTasksForProject(projectId: Int, projectName: String, taskIds: List<Int>, projects: MutableList<Project>, totalProjects: Int): Загрузка данных задач для проекта.

19. UserTaskChatActivity.kt Класс: UserTaskChatActivity

**Функциональность:** Этот класс предоставляет интерфейс для общения пользователей в рамках задачи.

**Основные методы:**

* onCreate(): Инициализация элементов интерфейса и установка слушателей для кнопок.
* loadChatMessages(taskId: Int): Загрузка сообщений чата из Firebase.
* sendMessage(message: String): Отправка нового сообщения и обновление Firebase.

20. UserCompletedTasksActivity.kt Класс: UserCompletedTasksActivity

**Функциональность:** Этот класс отображает список завершенных задач для пользователя.

**Основные методы:**

* onCreate(): Инициализация элементов интерфейса и установка слушателей для кнопок.
* loadCompletedTasks(): Загрузка завершенных задач из Firebase.

# Описание основных макетов активностей программы «Органайзер».

## 1. activity\_admin\_add\_user.xml

**Функциональность:** Этот макет предоставляет интерфейс для добавления нового пользователя в систему.

**Основные элементы:**

* EditText для ввода имени пользователя.
* EditText для ввода пароля.
* Spinner для выбора роли пользователя.
* Button для подтверждения добавления пользователя.

## 2. activity\_admin\_edit\_user.xml

**Функциональность:** Этот макет предоставляет интерфейс для редактирования данных существующего пользователя.

**Основные элементы:**

* EditText для редактирования имени пользователя.
* EditText для редактирования пароля.
* Spinner для выбора роли пользователя.
* Button для подтверждения изменений.

## 3. activity\_admin\_list\_of\_user.xml

**Функциональность:** Этот макет отображает список всех пользователей и предоставляет возможность администраторам управлять ими.

**Основные элементы:**

* ListView для отображения списка пользователей.
* Button для добавления новых пользователей и ролей.
* RelativeLayout с дополнительными кнопками для добавления пользователей и ролей.

## 4. activity\_admin\_add\_role.xml

**Функциональность:** Этот макет предоставляет интерфейс для добавления новой роли в систему.

**Основные элементы:**

* EditText для ввода названия роли.
* Button для подтверждения добавления роли.

## 5. activity\_log\_in\_authorized.xml

**Функциональность:** Этот макет предоставляет интерфейс, отображаемый после успешной авторизации пользователя.

**Основные элементы:**

* Элементы для отображения информации о пользователе и навигации по приложению.

## 6. activity\_login.xml

**Функциональность:** Этот макет предоставляет интерфейс для авторизации пользователей.

**Основные элементы:**

* EditText для ввода имени пользователя.
* EditText для ввода пароля.
* Button для входа в систему.

## 7. activity\_user\_add\_project.xml

**Функциональность:** Этот макет предоставляет интерфейс для добавления нового проекта.

**Основные элементы:**

* EditText для ввода названия проекта.
* Spinner для выбора типа проекта.
* Button для подтверждения добавления проекта.

## 8. activity\_user\_add\_task.xml

**Функциональность:** Этот макет предоставляет интерфейс для добавления новой задачи.

**Основные элементы:**

* Spinner для выбора проекта для задачи.
* EditText для ввода названия задачи.
* EditText для ввода описания задачи.
* Button для выбора даты и времени дедлайна.
* ListView для выбора пользователей, ответственных за задачу.
* Button для подтверждения добавления задачи.

## 9. activity\_user\_check\_project.xml

**Функциональность:** Этот макет предоставляет интерфейс для просмотра информации о проекте.

**Основные элементы:**

* TextView для отображения названия проекта.
* TextView для отображения типа проекта.
* TextView для отображения участников проекта.
* Button для завершения проекта.

## 10. activity\_user\_check\_task.xml

**Функциональность:** Этот макет предоставляет интерфейс для просмотра информации о задаче.

**Основные элементы:**

* TextView для отображения названия задачи.
* TextView для отображения описания задачи.
* TextView для отображения статуса задачи.
* Spinner для изменения статуса задачи.
* Button для сохранения изменений.
* Button для завершения задачи.
* Button для перехода в чат по задаче.

## 11. activity\_user\_end\_task.xml

**Функциональность:** Этот макет предоставляет интерфейс для завершения задачи.

**Основные элементы:**

* TextView для отображения названия задачи.
* TextView для отображения описания задачи.
* Button для подтверждения завершения задачи.

## 12. activity\_user\_show\_project.xml

**Функциональность:** Этот макет предоставляет интерфейс для отображения списка проектов и связанных с ними задач для пользователя.

**Основные элементы:**

* ListView для отображения списка проектов и задач.
* Button для добавления новых проектов и задач.

## 13. activity\_user\_task\_chat.xml

**Функциональность:** Этот макет предоставляет интерфейс для общения пользователей в рамках задачи.

**Основные элементы:**

* TextView для отображения заголовка чата.
* ListView для отображения сообщений чата.
* EditText для ввода нового сообщения.
* Button для отправки сообщения.

## 14. activity\_user\_completed\_tasks.xml

**Функциональность:** Этот макет предоставляет интерфейс для отображения списка завершенных задач.

**Основные элементы:**

* ListView для отображения списка завершенных задач.

## 15. item\_project\_header.xml

**Функциональность:** Этот макет используется для отображения заголовка проекта в списке проектов и задач.

**Основные элементы:**

* TextView для отображения названия проекта.

## 16. item\_task.xml

**Функциональность:** Этот макет используется для отображения информации о задаче в списке проектов и задач.

**Основные элементы:**

* TextView для отображения названия задачи.
* TextView для отображения статуса задачи.
* TextView для отображения дедлайна задачи.
* TextView для отображения оставшегося времени до дедлайна.

## 17. item\_message\_sent.xml

**Функциональность:** Этот макет используется для отображения отправленного сообщения в чате.

**Основные элементы:**

* TextView для отображения имени пользователя.
* TextView для отображения текста сообщения.
* TextView для отображения временной метки сообщения.

## 18. item\_message\_received.xml

**Функциональность:** Этот макет используется для отображения полученного сообщения в чате.

**Основные элементы:**

* TextView для отображения имени пользователя.
* TextView для отображения текста сообщения.
* TextView для отображения временной метки сообщения.

## 19. item\_user.xml

**Функциональность:** Этот макет используется для отображения информации о пользователе в списке пользователей.

**Основные элементы:**

* TextView для отображения имени пользователя.

## 20. item\_completed\_task.xml

**Функциональность:** Этот макет используется для отображения информации о завершенной задаче в списке завершенных задач.

**Основные элементы:**

* TextView для отображения названия задачи.
* TextView для отображения статуса задачи.
* TextView для отображения дедлайна задачи.

## 21. item\_user\_checkbox.xml

**Функциональность:** Этот макет используется для отображения пользователя с возможностью выбора (checkbox) в списке пользователей.

**Основные элементы:**

* CheckBox для выбора пользователя.
* TextView для отображения имени пользователя.

# Структура данных Firebase приложения «Органайзер»

1. endTasks: Содержит информацию о завершенных задачах.
	* actualTime: Фактическое время выполнения задачи.
	* chat: История чата, связанная с задачей.
	* comment: Комментарий к задаче.
	* creationDate: Дата создания задачи.
	* creationTime: Время создания задачи.
	* deadlineDate: Дата дедлайна задачи.
	* deadlineTime: Время дедлайна задачи.
	* description: Описание задачи.
	* name: Название задачи.
	* projectId: Идентификатор проекта, к которому относится задача.
	* status: Статус задачи.
	* userIds: Список идентификаторов пользователей, назначенных на задачу.
2. projects: Содержит информацию о проектах.
	* name: Название проекта.
	* taskIds: Список идентификаторов задач, связанных с проектом.
	* type: Тип проекта (например, публичный).
	* userIds: Список идентификаторов пользователей, назначенных на проект.
3. roles: Содержит информацию о ролях пользователей.
	* roleName: Название роли.
4. tasks: Содержит информацию о задачах.
	* chat: История чата, связанная с задачей.
	* creationDate: Дата создания задачи.
	* creationTime: Время создания задачи.
	* deadlineDate: Дата дедлайна задачи.
	* deadlineTime: Время дедлайна задачи.
	* description: Описание задачи.
	* name: Название задачи.
	* projectId: Идентификатор проекта, к которому относится задача.
	* status: Статус задачи.
	* userIds: Список идентификаторов пользователей, назначенных на задачу.
5. users: Содержит информацию о пользователях.
	* isAdmin: Признак администратора.
	* password: Пароль пользователя.
	* role: Роль пользователя.
	* type: Тип пользователя (например, администратор).
	* username: Имя пользователя.
	* projectIds: Список идентификаторов проектов, в которых участвует пользователь.
	* taskIds: Список идентификаторов задач, в которых участвует пользователь.

Для чтения данных из Firebase используется метод readDatabase из класса FirebaseHelper. Этот метод выполняет асинхронное чтение данных и вызывает callback-функцию после завершения операции.

# Руководство пользователя приложения «Органайзер»

## Введение

Данное руководство предназначено для пользователей приложения. Оно объясняет основные функции и особенности приложения, а также предоставляет пошаговые инструкции по использованию различных интерфейсов.

### Основные функции

1. **Управление проектами и задачами**:
	* Создание и редактирование проектов.
	* Назначение пользователей на проекты.
	* Создание и редактирование задач в рамках проектов.
	* Назначение пользователей на задачи.
	* Установка сроков выполнения задач.
	* Отслеживание прогресса выполнения задач.
2. **Чат и комментарии**:
	* Встроенный чат для обсуждения задач.
	* Возможность оставлять комментарии к задачам.
3. **Авторизация и управление пользователями**:
	* Авторизация пользователей с различными ролями (администратор, пользователь).
	* Управление учетными записями пользователей.

### Вход в систему

#### Шаги:

1. Откройте приложение.
2. На экране авторизации введите имя пользователя и пароль.
3. Нажмите кнопку "Войти".

#### Возможные проблемы:

* Неправильное имя пользователя или пароль. В этом случае попробуйте ввести данные еще раз или обратитесь к администратору для сброса пароля.

### Интерфейс администратора

### Данные входа администратора логин admin пароль 768312

#### 1. Добавление нового пользователя

1. На главной странице администратора выберите опцию "Добавить пользователя".
2. Введите имя пользователя, пароль и выберите роль пользователя.
3. Нажмите кнопку "Сохранить" для добавления нового пользователя.

#### 2. Редактирование существующего пользователя

1. На главной странице администратора выберите "Список пользователей".
2. Найдите пользователя, которого хотите редактировать, и нажмите на его имя.
3. Внесите необходимые изменения и нажмите кнопку "Сохранить".

#### 3. Добавление новой роли

1. На главной странице администратора выберите опцию "Добавить роль".
2. Введите название новой роли.
3. Нажмите кнопку "Сохранить" для добавления новой роли.

### Интерфейс пользователя

#### 1. Просмотр проектов и задач

1. На главной странице после авторизации выберите "Проекты".
2. Выберите интересующий вас проект для просмотра связанных задач.
3. Для просмотра деталей задачи нажмите на её название.

#### 2. Добавление нового проекта

1. На главной странице выберите опцию "Добавить проект".
2. Введите название проекта и выберите пользователей, которые будут участвовать в проекте.
3. Нажмите кнопку "Сохранить" для создания проекта.

#### 3. Добавление новой задачи

1. Откройте проект, в который хотите добавить задачу.
2. Выберите опцию "Добавить задачу".
3. Введите название задачи, описание, установите сроки выполнения и выберите пользователей, которые будут назначены на задачу.
4. Нажмите кнопку "Сохранить" для создания задачи.

#### 4. Завершение задачи

1. Откройте задачу, которую хотите завершить.
2. Нажмите кнопку "Завершить задачу".
3. Подтвердите завершение задачи.

#### 5. Использование чата

1. Откройте задачу, по которой хотите начать обсуждение.
2. Внизу экрана найдите поле для ввода сообщения.
3. Введите сообщение и нажмите кнопку "Отправить".

### Завершение работы

Для выхода из системы:

1. Нажмите кнопку "Выход" на главной странице.
2. Подтвердите действие, если требуется.

### Возможные проблемы и их решение

* **Не удается войти в систему**:
	+ Проверьте правильность введенных данных.
	+ Убедитесь, что ваше устройство подключено к интернету.
	+ Обратитесь к администратору для сброса пароля.
* **Не отображаются данные проектов или задач**:
	+ Проверьте подключение к интернету.
	+ Обновите страницу или перезапустите приложение.
	+ Обратитесь к администратору для проверки состояния базы данных.