



МИНОБРНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное учреждение
высшего образования*

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий (ИТ)

Кафедра Математического обеспечения и стандартизации информационных
технологий (МОСИТ)

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 1

по дисциплине

«Тестирование и верификация программного обеспечения»

**Тема: «ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА
МЕТОДОМ «ЧЕРНОГО ЯЩИКА»»**

Выполнили студенты группы Горячковский Д., Толстов _____
А., Трофимов А., Хайруллин Д. «Котлы»

Принял _____

Практическая работа выполнена «__» _____ 202__ г.

(подпись студента)

«Зачтено»

«__» _____ 202__ г.

(подпись руководителя)

Москва 2025

Техническое Задание

Техническое задание на разработку программного продукта определяет цели, задачи и требования к созданию компьютерной игры в жанре платформера. Разработка осуществляется в рамках учебного проекта, основной целью которого является закрепление знаний в области программирования, проектирования интерфейсов, командной работы и создания интерактивных программных продуктов. Назначение разработки заключается в создании 2D пиксельной игры для персональных компьютеров, в которой игрок управляет персонажем и проходит несколько уровней, преодолевая препятствия.

Игровая концепция заключается в следующем: главный персонаж — условный герой (например, маленький человечек или фантастический персонаж), способный передвигаться влево и вправо и выполнять прыжки. Игровой процесс состоит из прохождения последовательности уровней, расположенных по принципу платформера: персонаж должен добраться от начала до конца уровня, избегая падений в пропасти, попадания в ловушки. Каждый уровень имеет уникальное оформление, усложняющееся по мере продвижения игрока.

Управление персонажем осуществляется с помощью клавиатуры: клавиши A/D отвечают за движение влево и вправо, клавиша пробел — за прыжок. Реализуется система жизней: при попадании в ловушку или падении персонаж возрождается в начале уровня. Для каждого уровня предусмотрено условие завершения — достижение специальной зоны выхода.

Интерфейс игры включает стартовое меню (пункты «Новая игра», «Выход»). Графика выполнена в пиксельном стиле, общий визуальный ряд оформляется в ретро-атмосфере с простыми, но яркими цветами.

К программе предъявляются следующие требования. Функциональные требования: управление персонажем, корректная работа механик движения и прыжка, наличие препятствий, система уровней с возрастающей сложностью. Требования к интерфейсу: наличие меню, простое управление. Условия эксплуатации: работа на персональных компьютерах под управлением Windows 10 и выше, запуск без установки дополнительных модулей. Надежность: устойчивое функционирование без сбоев при стандартных сценариях использования, корректная обработка выхода из игры и завершения уровней. Совместимость: корректная работа на ПК с минимальными системными характеристиками (процессор 2 ГГц, 4 Гб оперативной памяти, видеокарта с поддержкой 2D-графики, свободное место на диске не менее 200 Мб).

Критерии приемки включают возможность запуска игры и прохождения всех уровней, корректность работы управления, стабильность при длительном использовании, отсутствие критических ошибок при взаимодействии с игровыми объектами и наличие всех заявленных экранов интерфейса.

Этапы разработки включают анализ требований и создание концепта игры, проектирование уровней и интерфейсов, реализацию базовых механик персонажа и препятствий, разработку уровней, добавление графики и звуков, тестирование и исправление ошибок, итоговую презентацию продукта. В команде проекта участвуют программист (разработка кода и логики игры), художник (создание пиксельной графики персонажей, уровней и интерфейса), тестировщик (проверка стабильности и качества продукта).

Документация

Данный документ содержит описание компьютерной игры «Vitalik's walk», разработанной в жанре 2D-платформера. Документация предназначена для пользователей и разработчиков, участвующих в сопровождении проекта.

Назначение игры

Игра «Vitalik's walk» предназначена для развлечения и тренировки реакции. Игрок управляет персонажем, который проходит уровни, прыгая по платформам и избегая препятствий. Основная цель — пройти все уровни.

Установка

1. Скачайте архив с игрой и распакуйте его в любую папку на компьютере.
2. Запустите файл Vitalik's walk.exe (или другой исполняемый файл).
3. Дополнительная установка не требуется — все необходимые библиотеки включены в сборку.

Минимальные системные требования:

ОС: Windows 10 и выше

Процессор: 2 ГГц

ОЗУ: 4 ГБ

Видеокарта: поддержка 2D-графики

Свободное место на диске: 200 МБ

Управление

A — движение влево

D — движение вправо

Пробел — прыжок

Игровой процесс

Игрок начинает с 1 уровня. При падении в пропасть или попадании в ловушку персонаж теряет жизнь и возрождается в начале уровня.

Прохождение уровня завершается достижением двери/портала в конце.

Интерфейс

Главное меню: «Новая игра», «Выход».

Экран завершения игры: победа

Надежность

Игра сохраняет стабильность при длительном запуске.

Все ошибки обрабатываются без аварийного завершения.

Разработчики

Программист — разработка кода и логики игры.

Художник — создание пиксельной графики и интерфейсов.

Тестировщик — проверка стабильности и поиска ошибок.

Возможности расширения

Добавление новых уровней.

Введение новых видов ловушек.

Возможность сетевого соревнования по времени прохождения.

Описание внесённых ошибок в собственное ПО

1. Персонаж может застрять в стене при определённых условиях движения, что приводит к невозможности продолжить прохождение уровня.
2. В некоторых случаях при прыжке от блока персонаж визуально совершает отталкивание, но фактически его возвращает обратно на землю, что мешает нормальному управлению.
3. В отдельных местах карты игрок может вылететь за пределы игрового пространства, после чего отсутствует возможность вернуться, что вынуждает перезапуск уровня.
4. На некоторых участках отсутствует корректно настроенный хитбокс, из-за чего взаимодействие с объектами уровня не происходит должным образом.
5. Кнопка «заново» смещается за пределы экрана, что делает её использование затруднительным или невозможным.

6. Одна из дверей на уровне не имеет функционала: при достижении её персонажем событие завершения уровня не срабатывает.

Техническое задание и документация другой команды

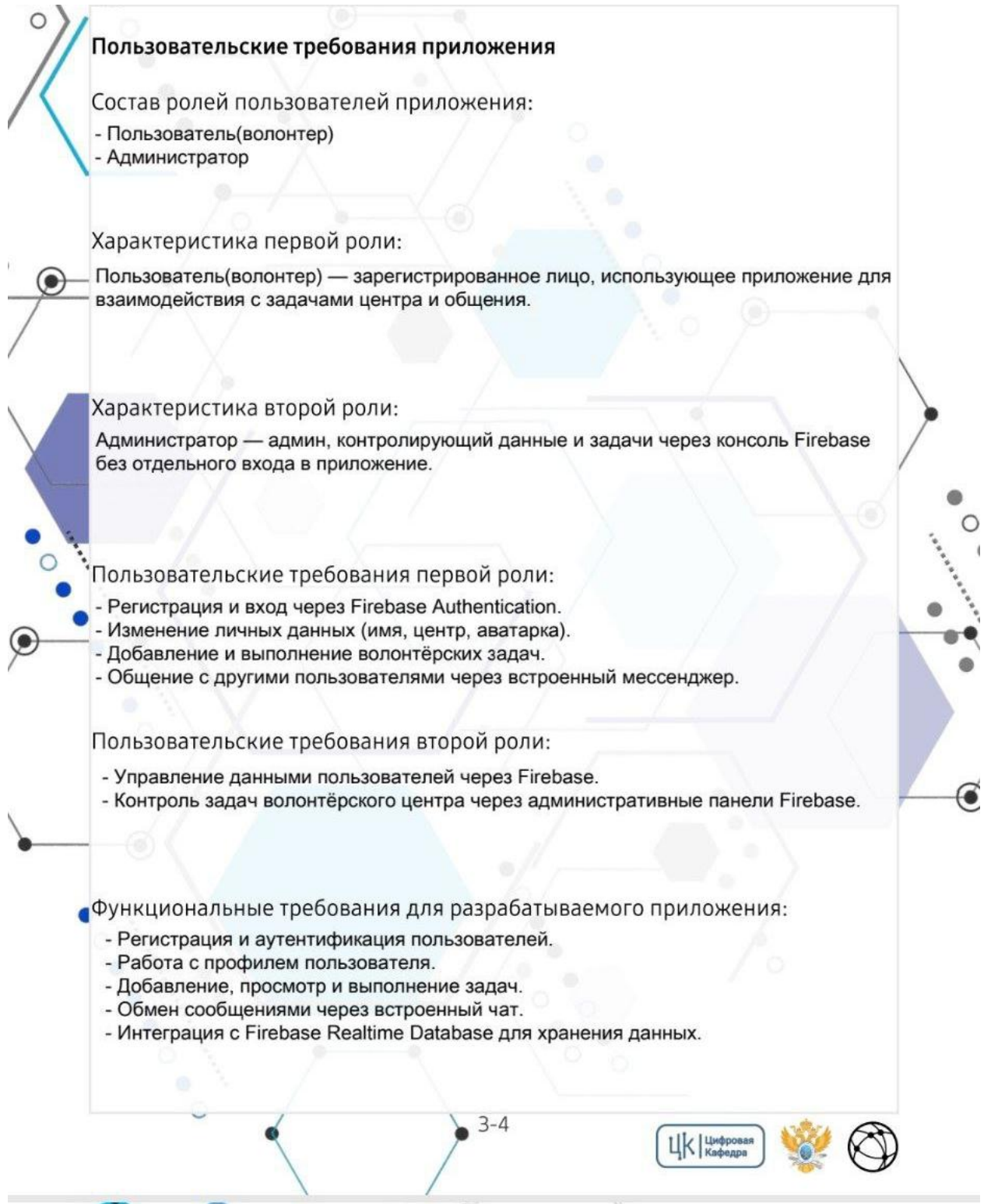


Рисунок 1 - Техническое задание

Документация

Описание разработки мобильного приложения «Volunteer»

Исходные данные

Целевая аудитория: пользователи Android, волонтеры.

Язык программирования: Kotlin.

Среда разработки: Android Studio (клиент), Firebase (сервер).

Базы данных: Firebase Realtime Database и Cloud Firestore.

Фреймворки: Jetpack Compose, Firebase SDK.

Функционал приложения

Приложение предоставляет:

- регистрацию и аутентификацию через Firebase Authentication;
- редактирование личного профиля (имя, центр, аватар);
- добавление и выполнение задач;
- встроенный чат для общения волонтеров в реальном времени.

Предметная область

Основное назначение — обмен сообщениями между участниками волонтерских центров и организация совместных задач с использованием облачных технологий.

Пользователи и роли

Волонтер — регистрируется в системе, редактирует данные, участвует в выполнении задач и общается в чате.

Администратор — управляет пользователями и задачами через Firebase-консоль, без отдельного интерфейса в приложении.

Требования

Функциональные: регистрация, работа с профилем, задачи, чат, интеграция с Firebase.

Пользовательские:

Волонтёр: регистрация, редактирование профиля, работа с задачами.

Администратор: управление данными и задачами через Firebase, поддержка чатов.

Клиентская часть

Реализована на Kotlin с использованием Jetpack Compose и Material Design 3.

Основные экраны: чаты, центры, задания, профиль.

Бизнес-логика:

- отображение и создание чатов и заданий,
- просмотр пользователей центра,
- локальное управление состояниями интерфейса.

Особенности реализации:

- аутентификация и работа с данными через Firebase SDK;
- асинхронные операции (корутины + listeners);
- синхронизация в реальном времени;
- сохранение состояний с помощью rememberSaveable.

Архитектура

Архитектура построена по клиент-серверной модели с использованием Firebase.

Клиент реализует бизнес-логику и интерфейс.

Серверная часть отсутствует как отдельный компонент — её заменяют сервисы Firebase (Authentication, Firestore).

Хранение данных и обмен сообщениями происходят в облаке.

Модели данных

Для работы с локальной БД применяется библиотека Room.

Основные сущности:

- UserProfile: email, nickname, центр, статус.
- VolunteerCenter: id, название, описание, цели.
- Task: id, название, описание, статус выполнения.
- Chat, ChatMessage: идентификаторы, участники, сообщения.

Серверная часть

Firebase обеспечивает хранение данных и авторизацию пользователей:

Authentication — регистрация и вход по email и паролю.

Firestore — коллекции users, centers, chats с JSON-структурами.

Идентификаторы совпадают с UID пользователей из Firebase.

Масштабируемость и безопасность достигаются без разработки собственного сервера.

Методы работы с БД: collection(), document(), set(), get(), update().

Для расширенной логики возможно применение Cloud Functions.

Итоги

Мобильное приложение «Volunteer» предоставляет удобный инструмент для взаимодействия волонтеров:

- работа с задачами и чатами,
- управление данными в реальном времени,
- простая интеграция через Firebase без отдельного сервера.

Используемая архитектура обеспечивает расширяемость, гибкость и надёжность приложения.

Результаты тестирования программного продукта другой команды

Баг первый

Вне зависимости от настройки статуса активный неактивный в страницах настройки - во вкладке центр баг будет с тем - что все пользователи будут отображаться как активные.

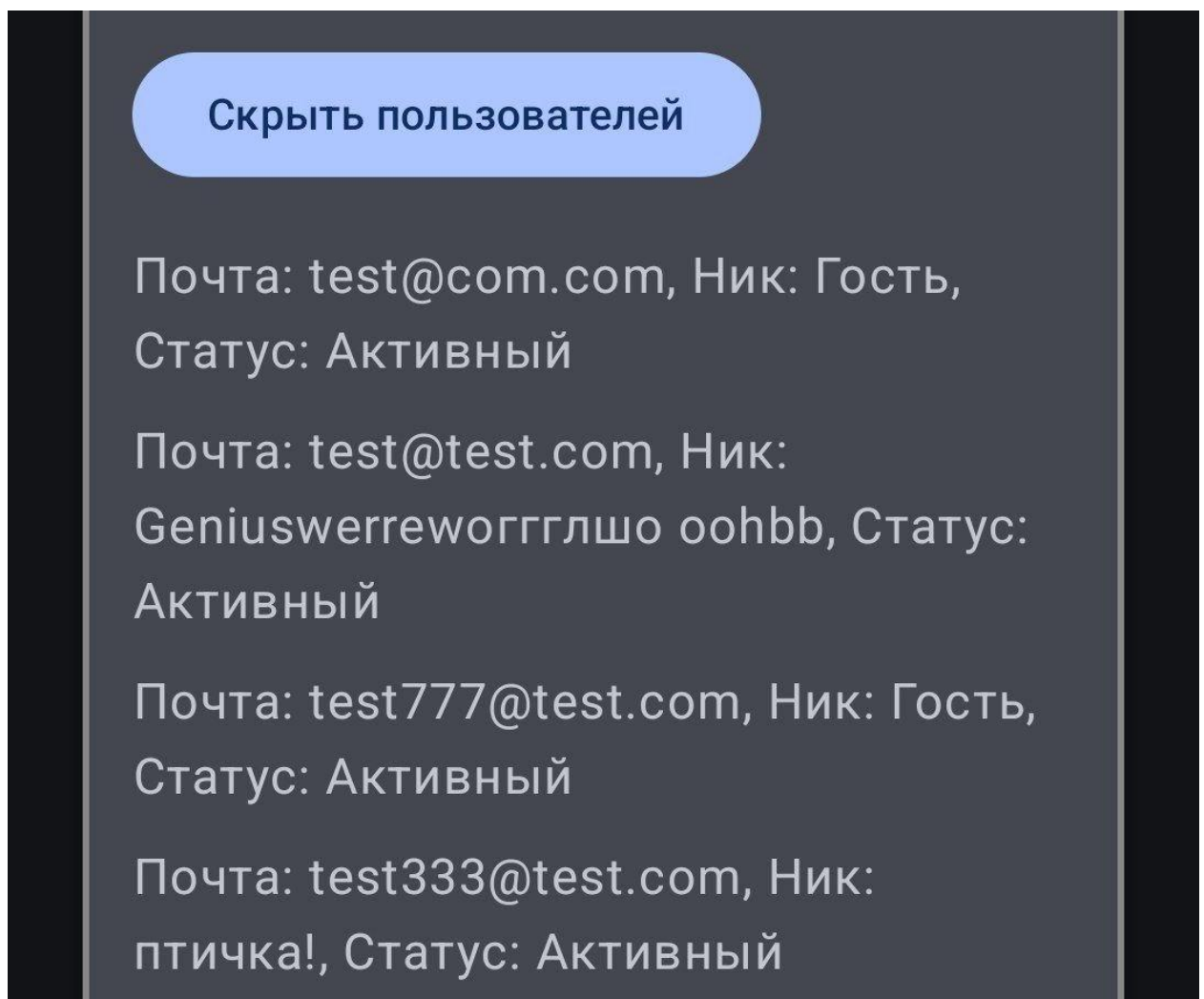
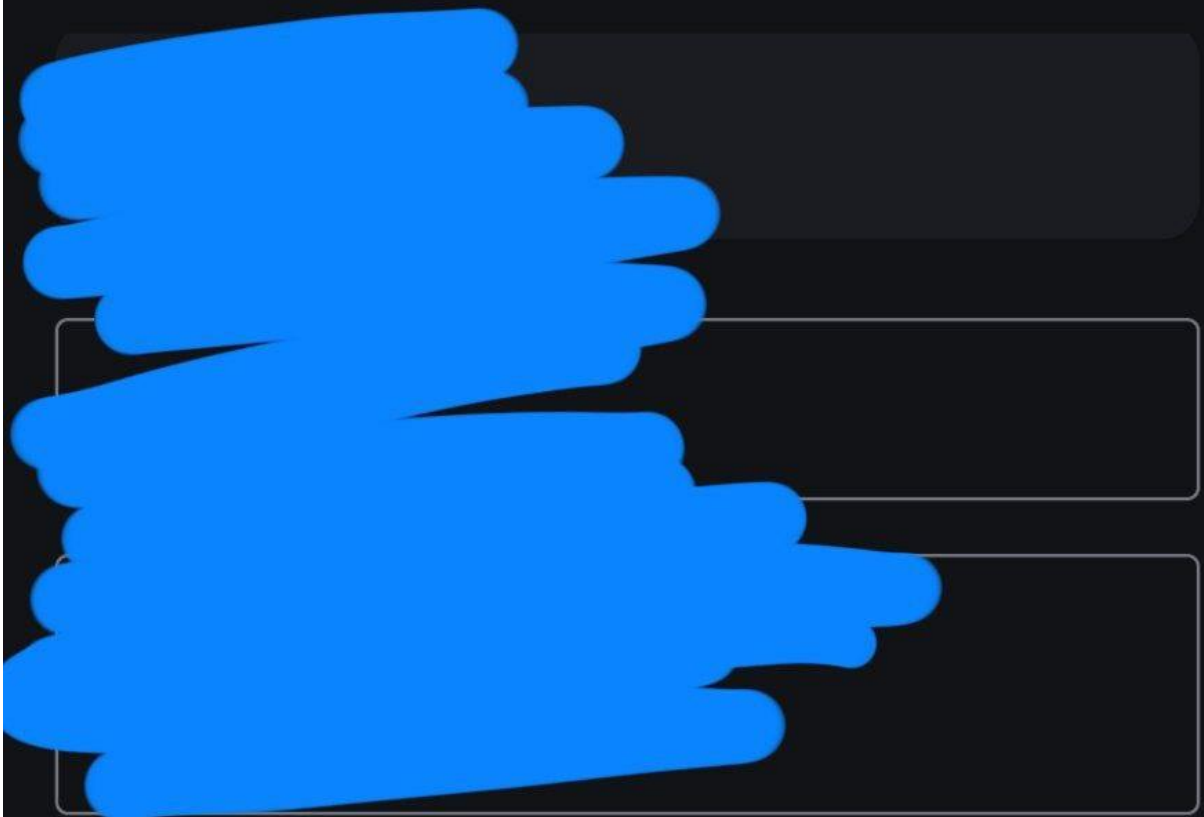


Рисунок 2 – Баг первый

Профиль

A large area of the profile form is obscured by a bright red brushstroke, indicating a bug where the user's information is not visible or is being incorrectly rendered.

Статус

Статус

Неактивный ▲

Активный

Неактивный

Рисунок 3 – Баг первый

Баг второй

Дублируется поле с почтой

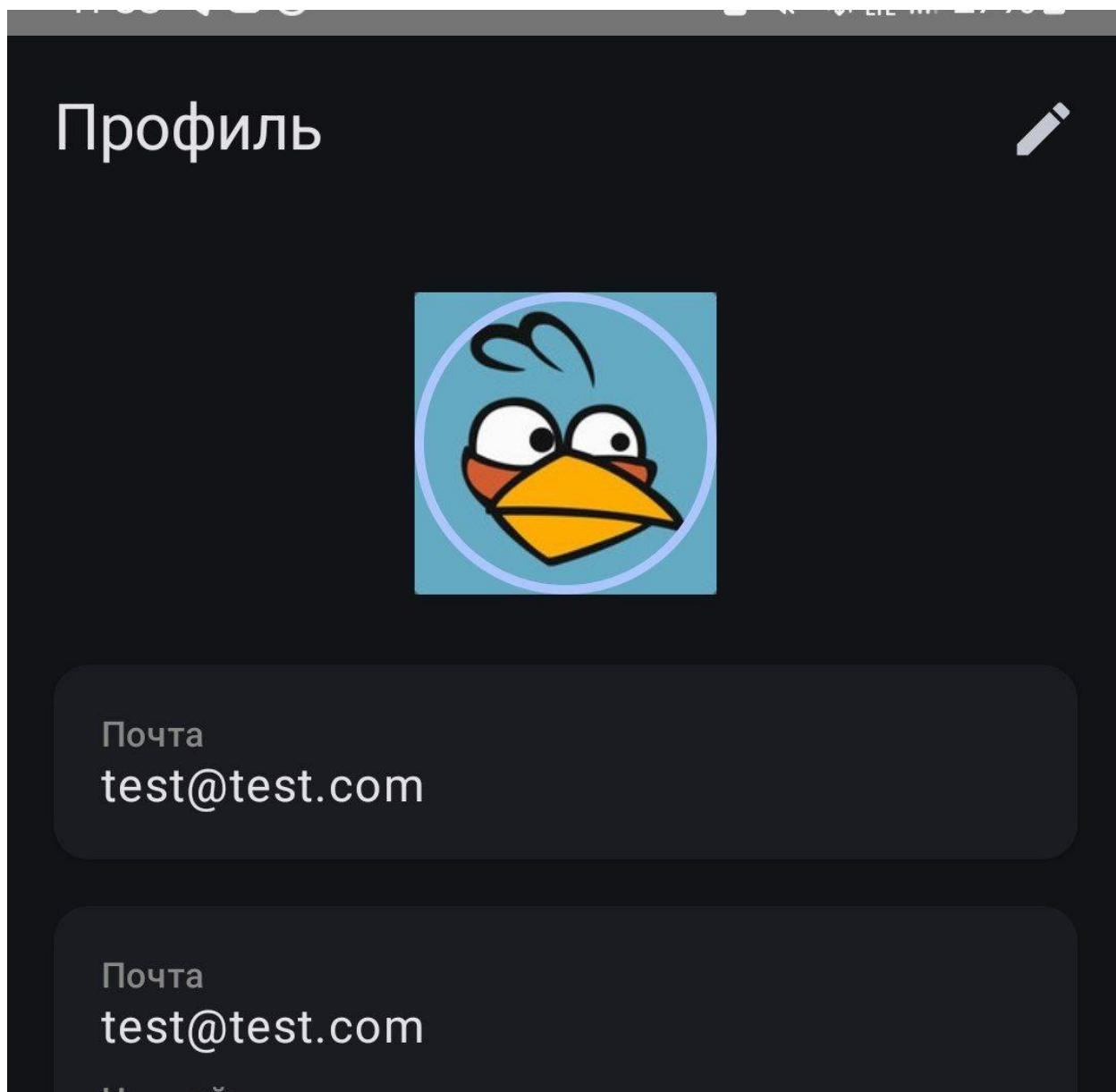


Рисунок 4 – Баг второй

Баг третий

Вместо количества участников центра пользователей пишется "дата основания"



Рисунок 5 – Баг третий

Баг четвертый

При нажатии кнопки выйти из аккаунта открывается карта google

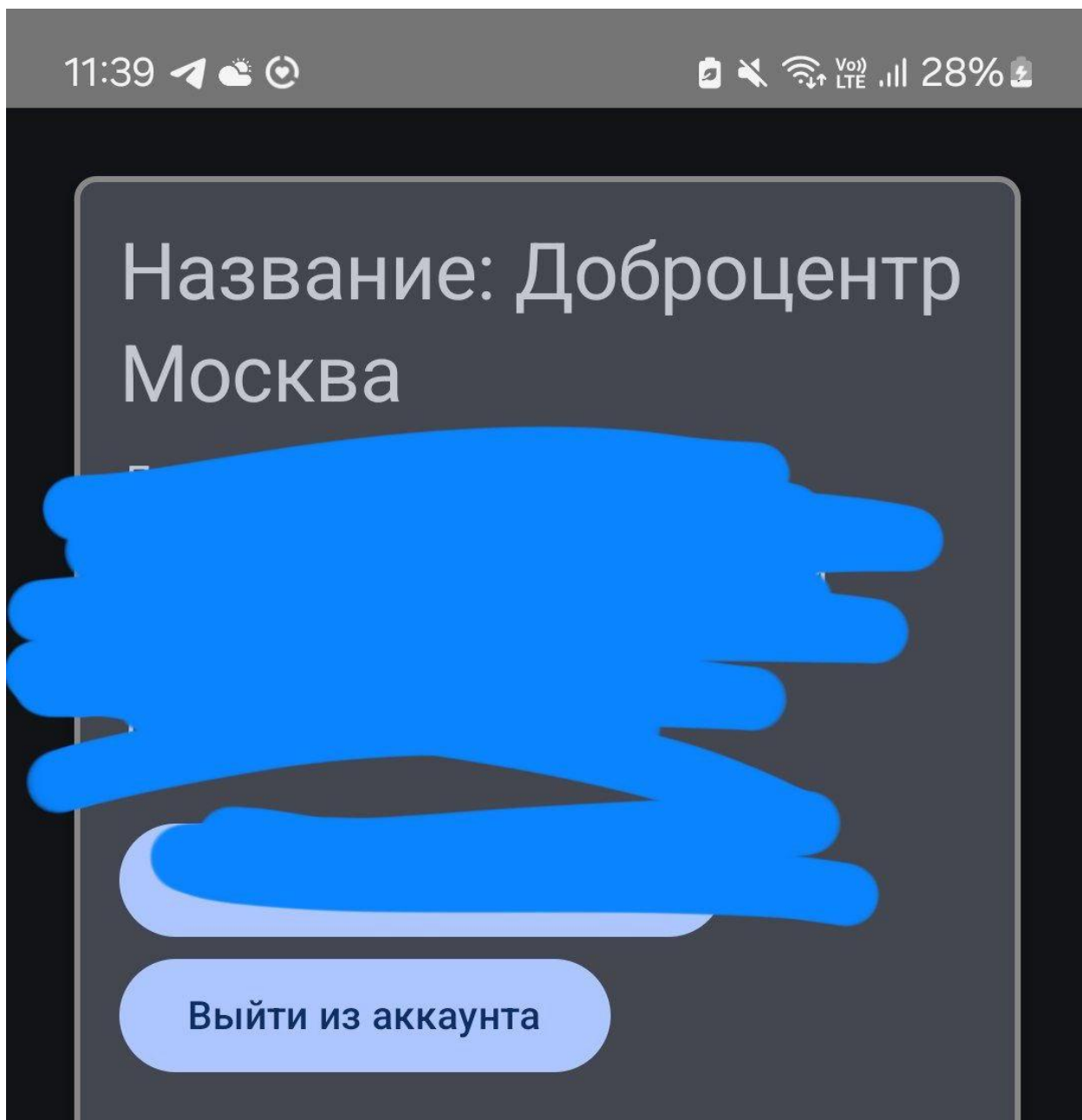


Рисунок 6 – Баг четвертый

Баг пятый

При выборе синего аватара - в профиле устанавливается красный

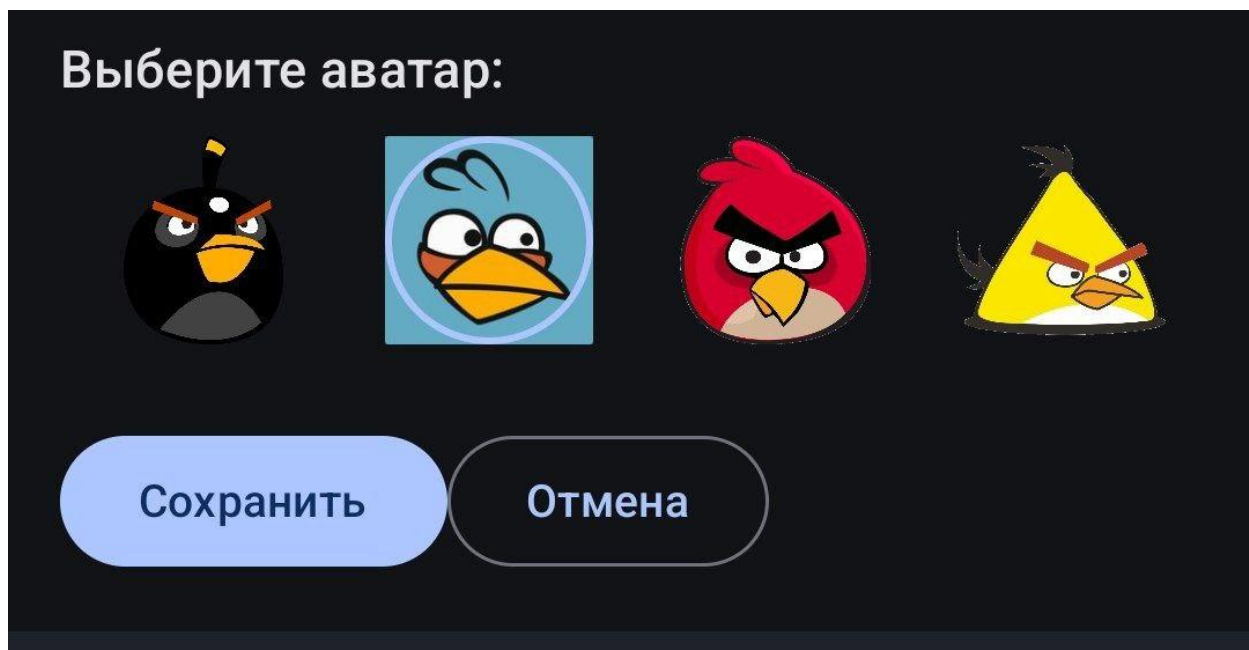


Рисунок 7 – Баг пятый

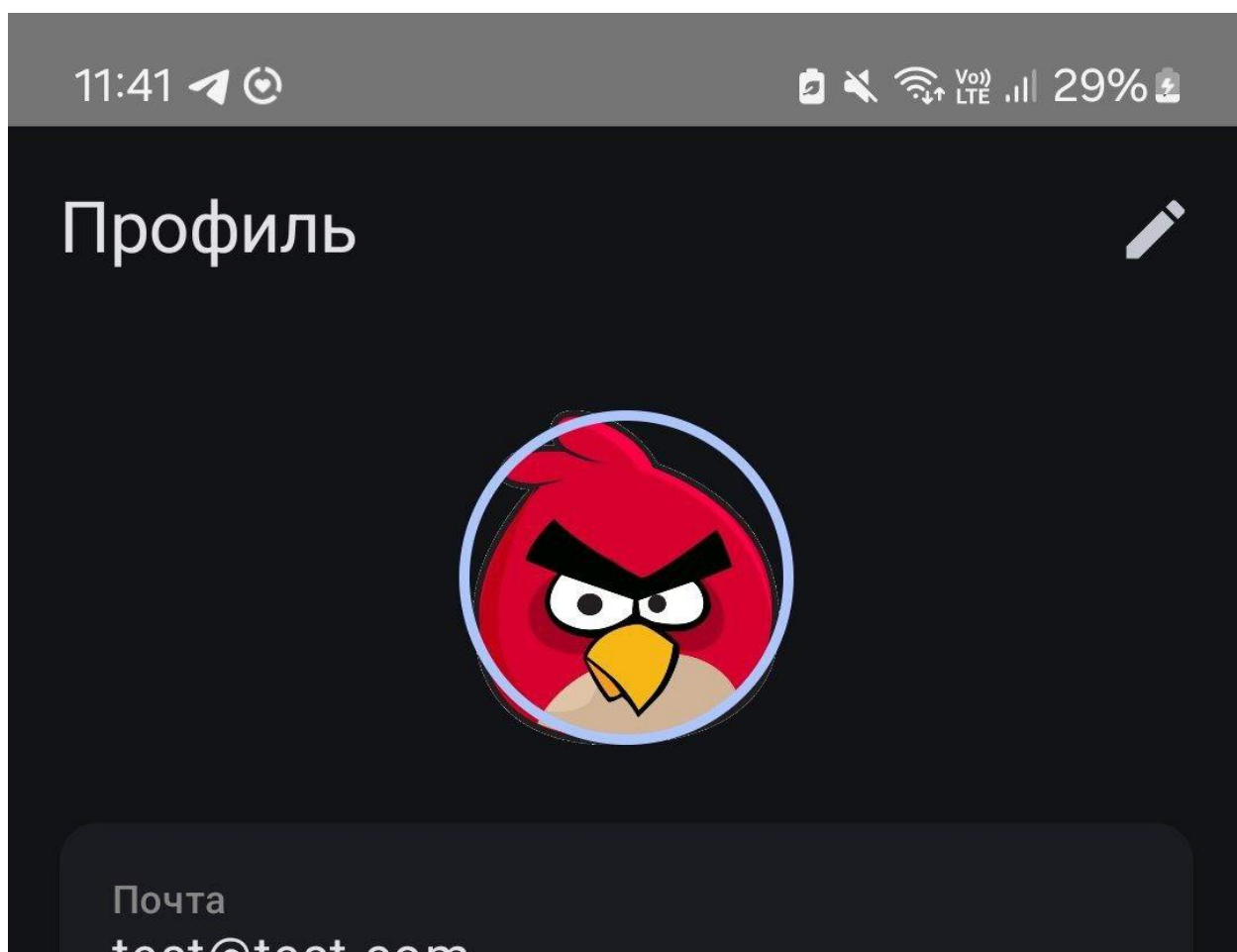


Рисунок 8 – Баг пятый

Анализ документации другой команды

При изучении документации по мобильному приложению «Volunteer» были выявлены следующие недочёты:

- Неактуальная информация: в документации указано, что статус пользователя может меняться, однако в тестировании обнаружен баг с отображением статуса, из-за чего всегда показывается «активный».
- Отсутствие деталей: при описании профиля не уточнено, каким образом происходит выбор и сохранение аватара. В тестировании выявлено несоответствие — выбранный синий аватар сохраняется как красный.
- Неактуальная информация: в разделе профиля указано использование уникального email, но на практике тестирование показало дублирование поля.

Заключение

Качество программного продукта «Volunteer» можно оценить как удовлетворительное: основные функции реализованы, но часть элементов работает с ошибками (статус пользователя, выбор аватара, выход из аккаунта, отображение данных центра).

Документация в целом описывает архитектуру и функционал приложения, однако не отражает некоторые ошибки, выявленные в тестировании. Продукт частично соответствует ТЗ, но требует исправления выявленных багов.

Рекомендации по улучшению:

- Исправить некорректное отображение статуса пользователя.

- Устранить баги с профилем (дублирование email, выбор аватара).
- Проверить обработку кнопки «выйти из аккаунта».
- Обновить документацию с учётом исправлений.

Приложения

Приложенные материалы:

- Скриншоты выявленных багов (рисунки 2–8).