

Министерство науки и высшего образования Российской Федерации
Муромский институт (филиал)
федерального государственного бюджетного образовательного учреждения высшего образования
**«Владимирский государственный университет
Имени Александра Григорьевича и Николая Григорьевича Столетовых»
(МИВлГУ)**

Факультет _____ ИТР _____

Кафедра _____ ПИН _____

КУРСОВАЯ РАБОТА

по _____ Разработка приложений для мобильных операционных систем _____

Тема _____ Приложение «Волейбольный помощник» _____

(оценка)

Руководитель

Мортин К.В. _____
(фамилия, инициалы)

(подпись) _____ (дата)

Члены комиссии

Студент ПИН-119 _____
(группа)

(подпись) (Ф.И.О.)

Львова К.Ю. _____
(фамилия, инициалы)

(подпись) (Ф.И.О.)

(подпись) _____ (дата)

В данной курсовой работе разработано приложение «Волейбольный помощник». В ходе выполнения курсовой работы произведен анализ и сбор требований к проектируемой системе. На этапе разработки структуры классов выделены основные сущности, их атрибуты и взаимосвязи. Система создавалась на языке программирования Kotlin в среде разработки Android Studio. Для хранения данных о клиентах, товарах и их дефектах использовалась такая база данных как SQLite. На заключительном этапе работы произведено тестирование разработанного продукта.

In this course work, the workstation of the pawnshop administrator has been developed. In the course of the course work, the analysis and collection of requirements for the designed system was made. At the stage of class development, the main entities, their attributes and relationships are highlighted. The system was created in the Java programming language in the IntelliJ IDEA development environment. For such storage of data about customers and their defects, a database like SQLite was used. At the final stage of work, the developed product was tested.

Содержание

Введение	6
1. Анализ технического задания	7
1.1 Описание предметной области.....	7
1.2 Анализ требований	10
2. Разработка алгоритмов	13
3. Руководство программиста	20
4. Руководство пользователя.....	38
5. Тестирование	56
Заключение.....	64
Список использованных источников	65
Приложение 1. Функциональная модель	66
Приложение 2. Скриншоты программы.....	67
Приложение 3 Код программы	Ошибка! Закладка не определена.

					МИВУ 09.03.04-12.000 ПЗ			
Изм.	Лист	Недокум.	Подп.	Дата	Приложение «Волейбольный помощник»			
Разраб.		Львова К.Ю.						
Пров.		Мортин К.В.						
Н.контр...								
Утв.								
					Лит.	Лист	Листов	
					у		5	47
					МИ ВлГУ ПИН-119			

Введение

С развитием спорта внедрение информационных технологий приобретает особую актуальность. Особенно для тренировочного процесса волейболистов требующего постоянного обобщения и обмена опытом.

Цифровые видео, табло и проекционная техника, а также различные измерительные системы широко используются на всех крупных соревнованиях.

Секция волейбола есть почти в каждом учебном заведении не только в нашей стране, но и в целом мире. Поэтому существует необходимость небольшой автоматизации тренировочных процессов и организации официальных матчей.

Используя опыт игры в волейбол как в общеобразовательной и спортивной школах, институте и участия в городских и областных соревнованиях, можно заметить, что автоматизация игрового процесса необходима при ведении счета, определения переходов при забитом мяче, контроле взятия таймаутов и замен тренерами, создании протокола. Для облегчения работы судьи и/или секретаря матча необходимо разработать программу для помощи ведения игрового процесса.

Целью работы является разработка приложения «Волейбольный помощник»

Задачами данной работы являются реализация такого функционала как:

- 1) Контроль игрового процесса, включающего в себя ведение счета очков, замен и таймаутов
- 2) Удобный, интуитивно понятный интерфейс
- 3) 2 уровнями доступа, первый для тренировок, второй для официальных матчей
- 4) Две роли пользователей (зарегистрированный и незарегистрированный) с разными уровнями доступа
- 5) Автоматическое создание протокола матча по его окончании

1. Анализ технического задания

1.1 Описание предметной области

Волейбол - один из самых массовых и зрелищных видов спорта, распространенный на всех континентах. Это игра для отдыха и поддержания формы, которая увлекает многих. Это вид спорта для всех.

Волейбол доступен как для детей, так и для людей пожилого возраста. Он популярен в школах и на спортивных площадках, поддерживается многими институтами и организациями, привлекает молодежь, которая хочет соревноваться на высоком уровне.

Для автоматизации игрового и тренировочного процесса уже существует несколько приложений. Одно из них это приложение "Оценка по волейболу простая"[1].

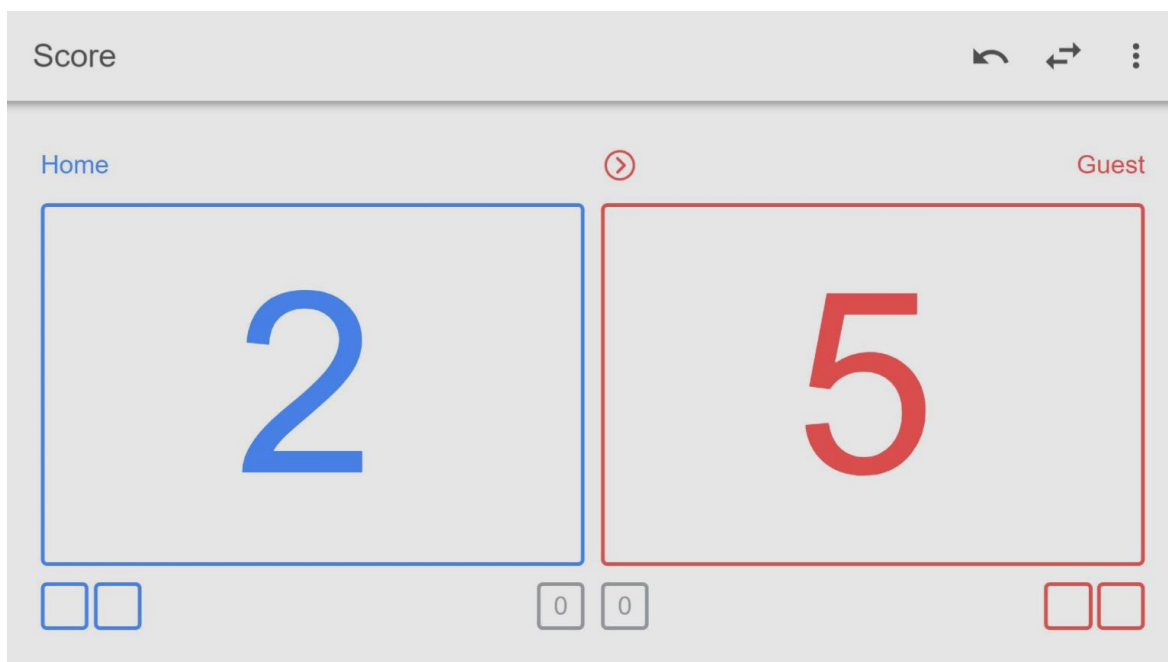


Рисунок 1 - аналог приложение "Оценка по волейболу простая"

Это простое и понятное приложение для подсчета очков, позволяющее подсчитывать ваши очки и отслеживать ваш результат в волейболе. Это совершенно бесплатно и не содержит рекламы.

Особенности:

- Увеличить количество очков можно нажатием на большие поля.
- Кнопкой отмены можно уменьшить количество очков.
- Значок круга указывает, кто подает следующим. Нажатием на один из значков круга перед первой точкой определяется, кто начинает подачу.
- Нажатием на названия команд можно изменять их.
- Нажатием на одну из маленьких ячеек, чтобы использовать тайм-аут.
- Используя меню синхронизации, можно синхронизировать табло с другими устройствами или веб-приложением.

Можно делиться текущим счетом в любое время или результатами после окончания игры.

Минусами данной программы является:

- Скучный интерфейс
- При назначении тайм-аутов можно изменять счет
- Текущий счет или результат игры для отправки сообщением – просто текст с название команд и счетом по партиям и т.д.

Еще одним аналогом является приложение «Счет в волейболе»[2].

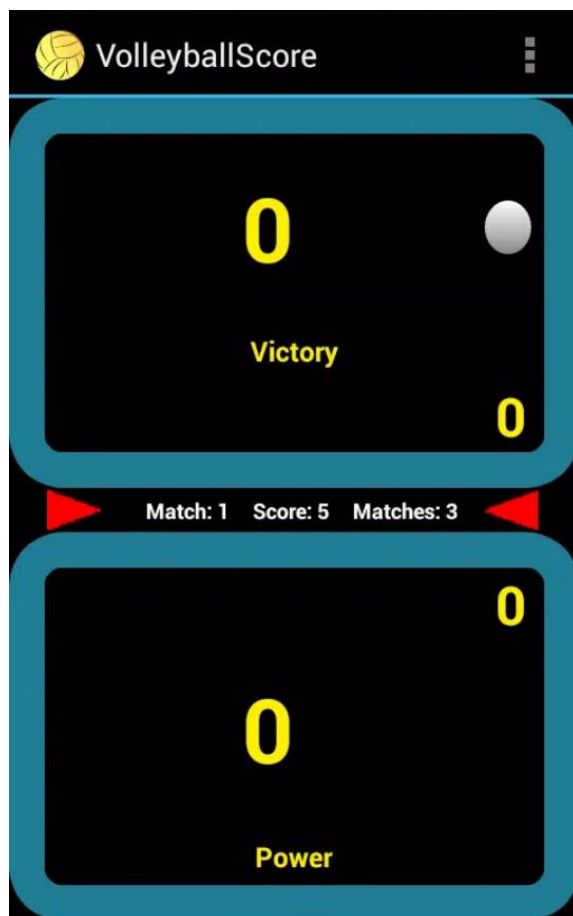


Рисунок 2 – аналог «Счет в волейболе»

Приложение назначено для ведения счета в волейболе.

В установке предусмотрена возможность настроить количество игр в матче, а также количество очков в игре и количество очков в последней, решающей игре.

Предусмотрена возможность хранить данные в базу данных, отправить эти данные по WhatsApp или скопировать в буфер для дальнейшего использования.

В начале новой игры возможно выбрать подающую команду или выбрать случайно.

Предусмотрена возможность откатить очко вперед или назад.

Недостатки данной программы является:

- Некрасивый интерфейс
- Маленький функционал
- Результат игры – просто текст и т.д.

1.2 Анализ требований

Для разработки приложения необходимо использовать информацию, полученную в ходе изучения аналогов и требований из технического задания. В данной курсовой работе необходимо реализовать:

Базу данных команд, пользователей, информации о текущем матче;

Автоматическое создание протокола игры;

Возможность:

- просмотра информации о уже занесенных командах
- отправки протокола игры представителям команд на почту
- съемки матча
- просматривать запись и протокол предыдущих матчей

Проанализировав требования можно составить диаграмму прецедентов (см. рисунок 3), показывающая общий функционал разрабатываемой программы.

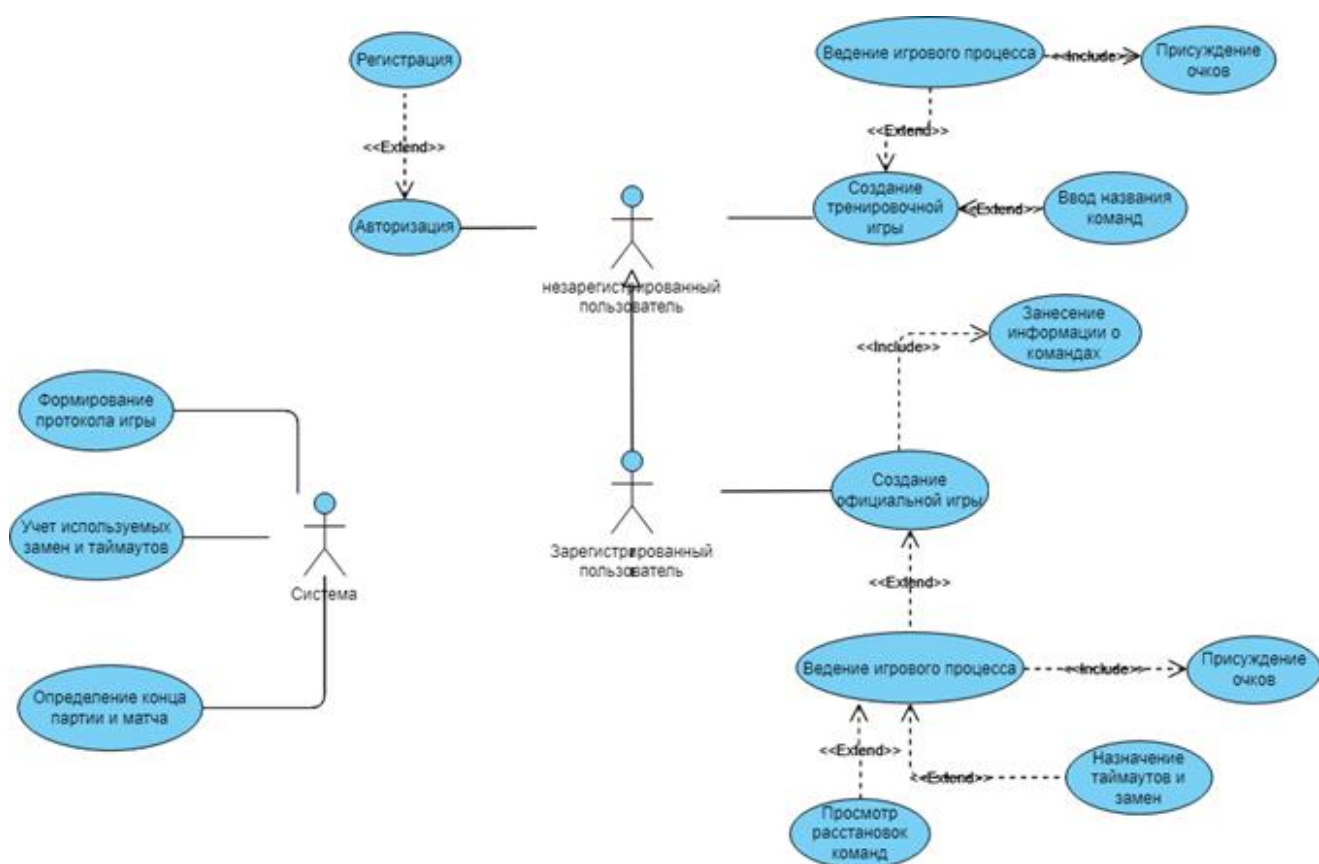


Рисунок 3 - Диаграмма прецедентов

Разрабатываемым приложением может пользоваться как незарегистрированный пользователь, так и зарегистрированный. Первый может только создавать так называемые тренировочные матчи, где не надо заносить состав команд и следить за расстановкой. Второй может создавать как тренировочные, так и официальные матчи, где пользователь может записывать матчи на камеру, заносить информацию о командах. Система в приложение должна следить за количеством взятых замен, кого на кого меняли, чтобы избежать ошибок и количествах тайм-аутов, при каком счете были взяты. Из этого набора данных, а также информации о командах и счете каждой партии в конце матча система должна формировать протокол игры.

Чтобы разработать приложение «Волейбольный помощник» Одной из наиболее популярных встраиваемых СУБД является СУБД SQLite. Ее основные преимущества:

- открытость
- бесплатность
- отсутствие потребности в администрировании
- простая процедура подключения
- высокая производительность, при достаточном объеме ОЗУ
- возможность шифрования данных

В качестве средства для разработки базы данных будет использоваться MySQL, а основная интегрированная среда разработки - Android Studio и язык программирования Kotlin.

Преимущества Android Studio:

- Приятный дизайнер пользовательских интерфейсов, позволяющий облегчить визуальное проектирование приложения
- Удобный XML редактор
- Поддержка системы контроля версий
- Эмуляция устройств
- Обширная база примеров проектирования
- Возможность проводить тестирование и анализ кода

- Скорость сборки приложения

Язык программирования Kotlin имеет такие преимущества и предоставляет следующие возможности:

- коды открытые, внедрения бесплатные
- программы применяют фреймворки, библиотеки Джава
- доступны компиляции в байткод JVM, JavaScript
- процесс изучения простой
- уровень безопасности высокий
- преобразование Java в Kotlin происходит автоматически

					МИВУ.09.03.04-4.000 ПЗ	Лист
						12
Изм	Лист	№ докум.	Подп.	Дата		

2. Разработка алгоритмов

Для разработки точных алгоритмов, включающих в себя всю логику разрабатываемой программы, необходимо провести моделирование данных используемых при разработке приложения.

Моделирование данных — это создание визуального представления о всей информационной системе либо ее части. Цель в том, чтобы проиллюстрировать типы данных, которые используются и хранятся в системе, отношения между этими типами данных, способы группировки и организации данных, их форматы и атрибуты [3].

Концептуальная модель — это модель, представленная множеством понятий и связей между ними, определяющих смысловую структуру рассматриваемой предметной области или её конкретного объекта [4].

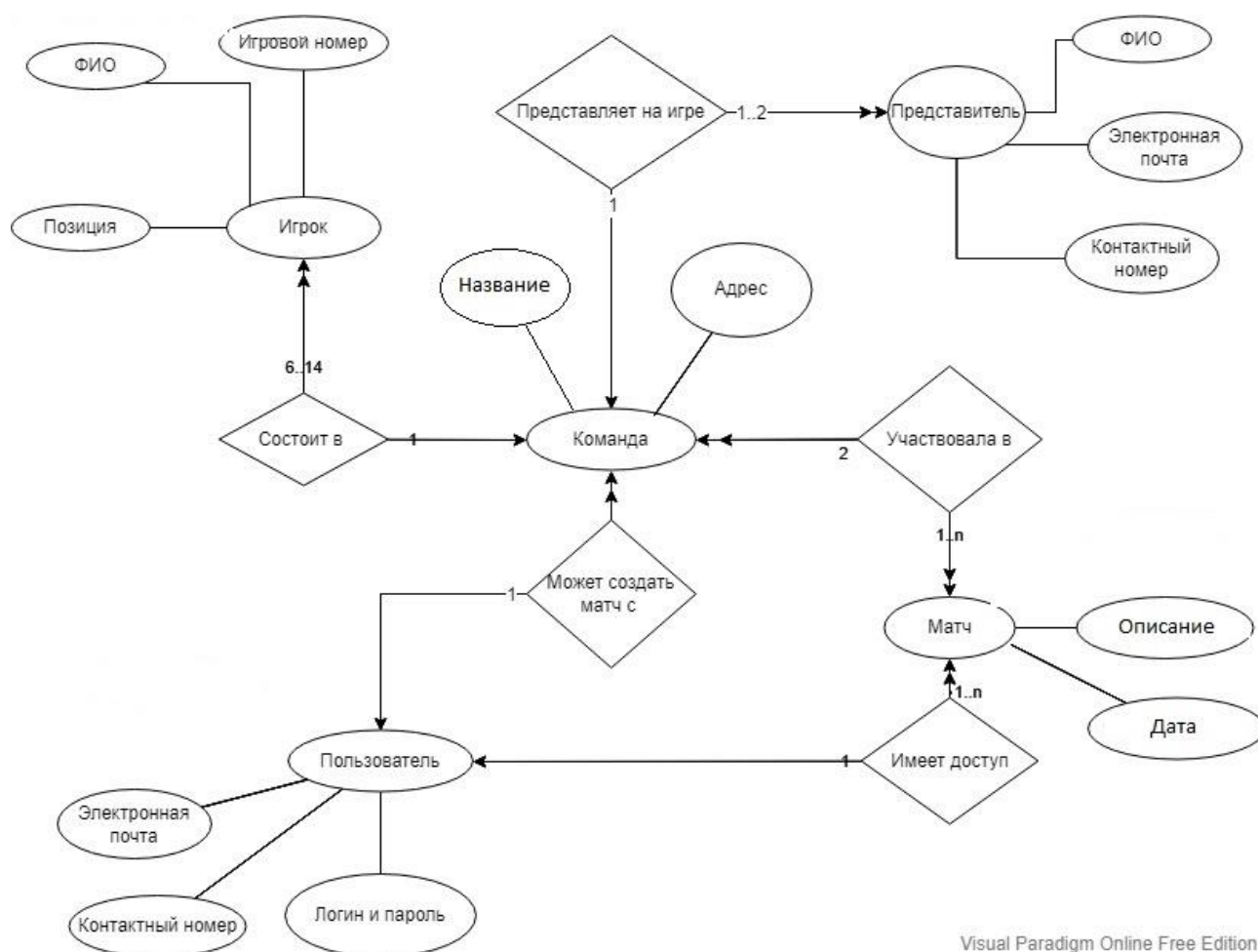


Рисунок 4 - Концептуальная модель

Логическая модель – графическое представление структуры базы данных с учетом принимаемой модели данных (иерархической, сетевой, реляционной и т.д.), независимое от конечной реализации базы данных и аппаратной платформы.

Иными словами, она показывает, что хранится в базе данных (объекты предметной области, их атрибуты и связи между ними), но не отвечает на вопрос как (рис. 5) [5].

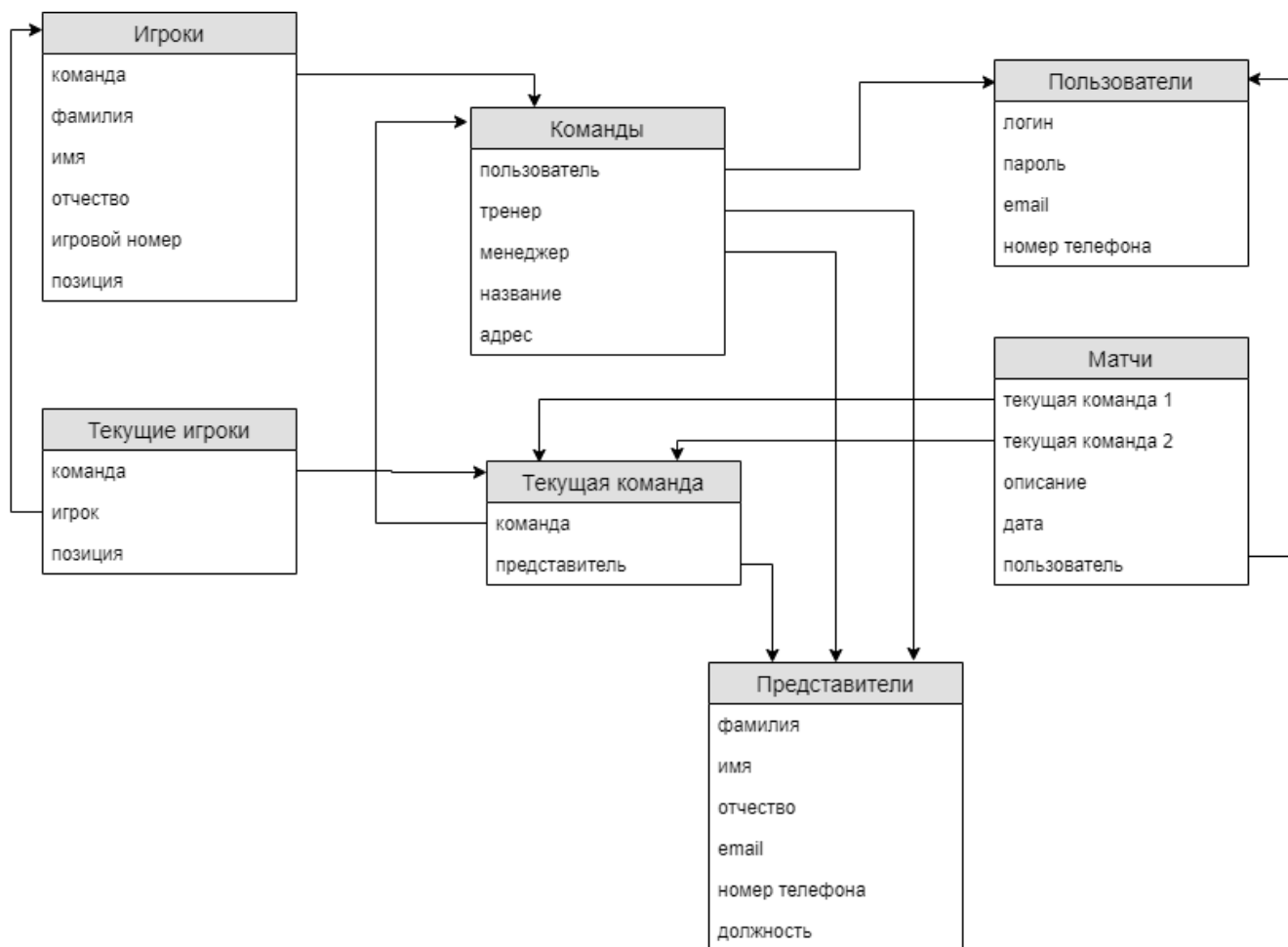


Рисунок 5 - Логическая модель

Физическая модель базы данных содержит все детали, необходимые для создания таблиц БД в СУБД SQLite. (рис. 6).

Физическая модель строится на основе логической с учетом ограничений, накладываемых возможностями выбранной СУБД.

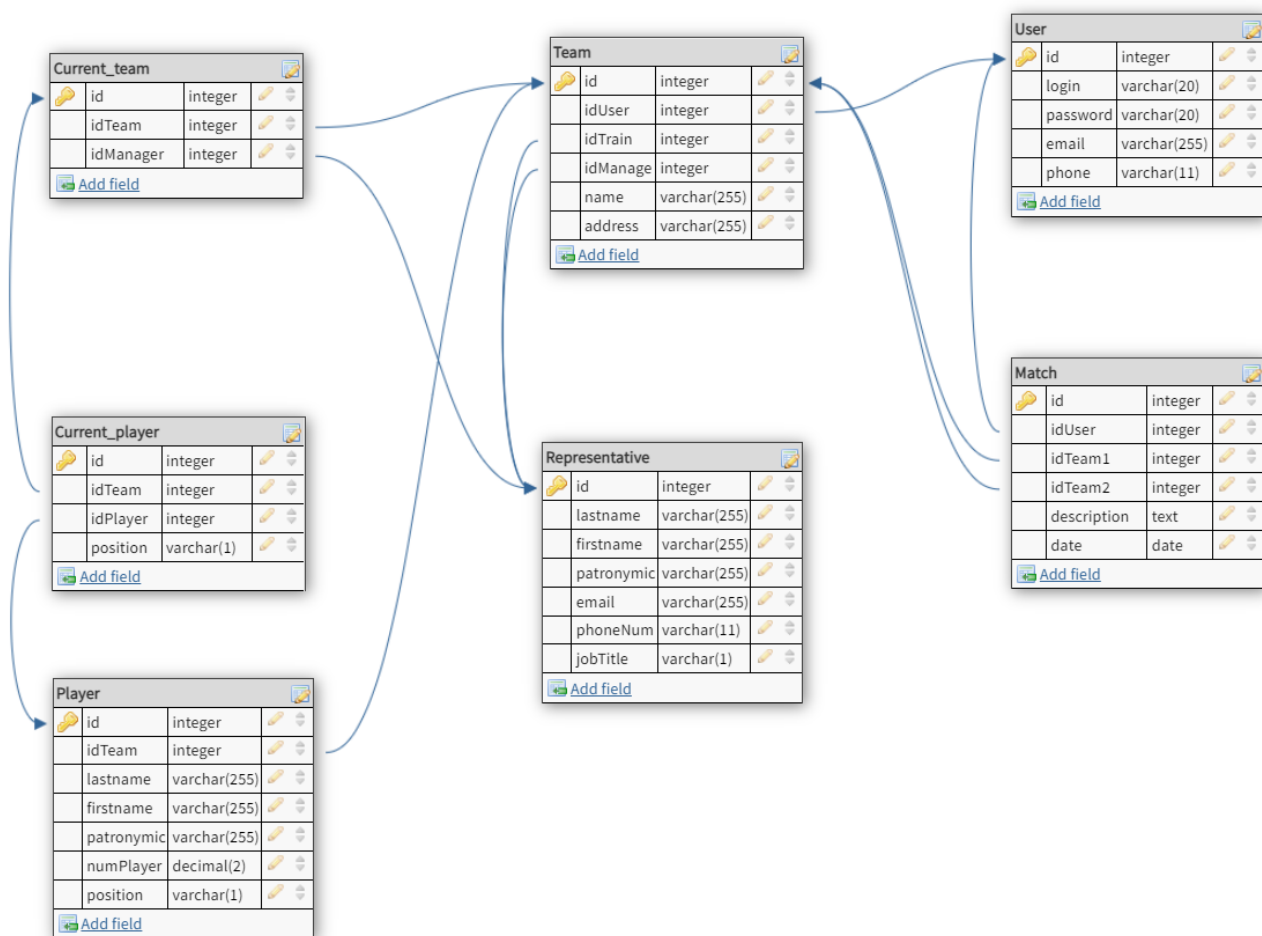


Рисунок 6 - Физическая модель

После была создана физическая модель, для явного представления таблиц в базе данных.

Таблица Users хранит данные о пользователе.

Поле id – идентификатор пользователя.

Поле login – логин для доступа.

Поле password – пароль для доступа.

Поле email – электронная почта.

Поле phoneNum – номер телефона.

Таблица 1 - таблица Users

Имя столбца	Тип данных
Id	Integer
login	Varchar(20)
Password	Varchar(20)
email	Text
phone	Varchar(11)

Таблица Representative хранит данные о представителях команд.

Поле id – идентификатор представителя.

Поле lastname – фамилия.

Поле firstname – имя.

Поле patronymic – отчество.

Поле email – электронная почта.

Поле phoneNum – номер телефона.

Поле jobTitle – должность (Т – тренер, М - менеджер).

Таблица 2 - таблица Representative

Имя столбца	Тип данных
Id	Integer
lastname	Varchar(255)
firstname	Varchar(255)
patronymic	Varchar(255)
email	Varchar(255)
phone	Varchar(11)
jobTitle	Varchar(1)

Таблица Team хранит данные о командах.

Поле id – идентификатор команды.

Поле idUser – идентификатор пользователя.

Поле idTrain – идентификатор тренера.

Поле idManage – идентификатор менеджера.

Поле name – название команды.

Поле address – адрес команды.

Таблица 3 - таблица Team

Имя столбца	Тип данных
Id	Integer
idUser	Integer
idTrain	Integer
idManage	Integer
name	Varchar(255)
address	Varchar(255)

Таблица Player хранит в себе данные о игроке команды.

Поле id – идентификатор игрока.

Поле idTeam – идентификатор команды.

Поле lastname – фамилия игрока.

Поле firstname – имя игрока.

Поле patronymic – отчество клиента.

Поле numPlayer – игровой номер игрока.

Поле position – амплуа.

Таблица 4 - таблица Player

Имя столбца	Тип данных
Id	Integer
idTeam	Integer
lastname	Varchar(255)
firstname	Varchar(255)
patronymic	Varchar(255)
numPlayer	Integer
position	Varchar(1)

Таблица Current_team хранит информацию о матче.

Поле id – идентификатор матча.

Поле idTeam – идентификатор команды.

Поле idManager – идентификатор менеджера, представляющего команду в данном матче.

Таблица 5 - таблица Current_team

Имя столбца	Тип данных
Id	Integer
idTeam	Integer
idManager	Integer

Таблица Current_player хранит информацию о матче.

Поле id – идентификатор матча.

Поле idTeam – идентификатор команды.

Поле idPlayer – идентификатор игрока, участвовавшего в данном матче.

Поле position – позиция игрока в данном матче (К – капитан, И – игрок, Л - либеро).

Таблица 6 - таблица Current_player

Имя столбца	Тип данных
Id	Integer
idTeam	Integer
idPlayer	Integer
position	Varchar(1)

Таблица Match хранит информацию о матче.

Поле id – идентификатор матча.

Поле idUser – идентификатор пользователя.

Поле idTeam1 – идентификатор первой команды.

Поле idTeam2 – идентификатор второй команды.

Поле description – описание матча.

Поле date – дата проведения матча.

Таблица 6 - таблица Matches

Имя столбца	Тип данных
Id	Integer
idUser	Integer
idTeam1	Integer
idTeam2	Integer
description	text
date	Date

3 Руководство программиста

В разрабатываемом приложении разработаны такие Activity как:

- AuthActivity – для авторизации пользователей
- RegistrationActivity – для регистрации пользователь
- ForUserMainActivity – главная страница приложения
- MatcherUserActivity – для вывода информации о матчах, проведенных пользователем
- TeamsUserActivity – для вывода и редактирования информации о командах пользователя, создания новых команд
- UserProfileActivity – вывод информации о пользователе с возможностью редактирования
- AddTeamActivity – для создания новой команды
- EditTeamActivity – для редактирования данных о команде и ее игроках
- CreateMatchActivity – для создания нового матча
- EditCurrentTeamActivity – для изменения данных о команде на текущий матч
- AddStaffTrainers – для изменения или добавления данных о менеджерах и тренерах команд, выбор представителя
- InfAboutMatchActivity – для внесения дополнительной информации о матче
- MainActivity – для проведения тренировок
- MatchActivity – для проведения матчей

Так же для работы с базой данных были созданы дополнительные классы такие как:

- DatabaseHelper – для создания соединения с БД
- WorkMyDB – основная работа с базой данных (выборка, добавление, изменение и удаление данных)
- User – для обработки данных пользователя

- Team – для обработки данных о команд
- Representative – для обработки данных о представителе команды
- Player – для обработки данных о игроке
- Match – для обработки данных о матче
- CurrentTeam – для обработки данных о текущей команде матча
- CurrentPlayer – для обработки данных о текущем игроке матча
- CurrentMatch – для обработки данных о текущем матче

Подробнее опишем Activity, которые были перечислены выше.

AuthActivity

Поля класса:

- workDb – экземпляр класса WorkMyDB для работы с БД.
- settings – экземпляр класса SharedPreferences (SharedPreferences — постоянное хранилище на платформе Android, используемое приложениями для хранения своих данных или настроек).

Методы класса:

- onCreate – данный метод вызывается при запуске Activity.

Здесь объявляются компоненты формы и задаются совокупности действий при нажатии кнопок.

- onOptionsItemSelected – для работы с меню (выход в главную форму)

RegistrationActivity

Поля класса:

- workDb – экземпляр класса WorkMyDB для работы с БД.
- settings – экземпляр класса SharedPreferences.

Методы класса:

- onCreate – данный метод вызывается при запуске Activity.

Здесь объявляются компоненты формы и задаются совокупности действий при нажатии кнопок.

- validateString – метод для проверки введенных данных.
- insertUser – добавление пользователя в базу данных.

- onOptionsItemSelected – для работы с меню (выход на предыдущую форму).

ForUserMainActivity

- settings – экземпляр класса SharedPreferences, для проверки авторизации пользователя.

- Flag – переменная типа Boolean для обозначения авторизации пользователя.

Методы:

- onCreate – данный метод вызывается при запуске Activity.

Здесь объявляются компоненты формы и задаются совокупности действий при нажатии кнопок, а также определяется авторизирован пользователь или нет. В зависимости от статуса пользователя позволяет нажимать определенные кнопки.

- onCreateOptionsMenu – создание меню
- onOptionsItemSelected – обработка нажатий компонентов меню.

Запрещает не авторизированному пользователю выбирать определенные элементы в меню.

MatcherUserActivity

Поля:

- workDb – экземпляр класса WorkMyDB для работы с БД.
- settings – экземпляр класса SharedPreferences, для получения id пользователя.

- listMatches – список матчей пользователя.

Методы:

- initField – создание таблицы матчей (дата проведения, описание).
- createRow – создание строки в таблице.
- marginView – создание параметров (ширина, высота компонента формы) для строк таблицы.
- marginRow – создание параметров (ширина, высота компонента формы) для элементов строки.

TeamsUserActivity

Поля:

- workDb – экземпляр класса WorkMyDB для работы с БД.
- settings – экземпляр класса SharedPreferences, для получения id

пользователя.

- listTeams – список команд пользователя.

Методы:

- initField – инициализация полей и описание логики работы кнопок, создание таблицы команд.

- createRow – создание строки в таблице.

- marginView – создание параметров (ширина, высота компонента формы) для строк таблицы.

- marginRow - создание параметров (ширина, высота компонента формы) для элементов строки.

- marginButton - создание параметров (ширина, высота компонента формы) для кнопок в строке таблицы.

UserProfileActivity

Поля:

- workDb – экземпляр класса WorkMyDB для работы с БД.
- settings – экземпляр класса SharedPreferences, для получения id

пользователя.

- Id – идентификатор пользователя.

- user – экземпляр класса User для обработки данных о пользователе.

Методы:

- onCreate – объявляются компоненты формы и задаются совокупности действий при нажатии кнопок.

- checkInfUser – проверка корректности вводимых данных.

- fillFields – изменение данных пользователя в БД.

AddTeamActivity

Поля:

- workDb – экземпляр класса WorkMyDB для работы с БД.
- settings – экземпляр класса SharedPreferences, для получения id пользователя.

- team - экземпляр класса Team для обработки данных о команде.
- listPlayers – список игроков команды.
- curPlayer – выбранный игрок из таблицы.
- adapter – адаптер для заполнения Spinner содержащий позиции игроков.
- idManager - идентификатор менеджера команды.
- idTrainer - идентификатор тренера команды.

Методы:

- onCreate – для получения данных из SharedPreferences и инициализации необходимых полей.

- initComponents – инициализация компонентов формы.
- fillField – инициализация адаптера.
- onClickListener – описания событий нажатия кнопок и других элементов на форме.

- checkField – проверка корректности данных о команде перед добавлением.

- checkInfPlayer - проверка корректности вводимых данных при добавлении или изменении данных о игроке.

- isCyrillic – проверка строки на содержание только кириллицы.
- createRowPlayer – создание строки игрока в таблице игроков.
- createRow0 – добавление строки названий столбцов таблицы.
- marginView – создание параметров (ширина, высота компонента формы) для строк таблицы.

- marginRow - создание параметров (ширина, высота компонента формы) для элементов строки.

- `marginButton` - создание параметров (ширина, высота компонента формы) для кнопок в строке таблицы.

`EditTeamActivity`

Поля:

- `workDb` – экземпляр класса `WorkMyDB` для работы с БД.
- `settings` – экземпляр класса `SharedPreferences`, для получения `id` пользователя.

- `team` - экземпляр класса `Team` для обработки данных о команде.
- `listPlayers` – список игроков команды.
- `curPlayer` – выбранный игрок из таблицы.
- `adapter` – адаптер для заполнения `Spinner` содержащий позиции игроков.
- `idManager` - идентификатор менеджера команды.
- `idTrainer` - идентификатор тренера команды.

Методы:

- `onCreate` – для получения данных из `SharedPreferences` и инициализации необходимых полей.

- `initComponents` – инициализация компонентов формы
- `fillField` – инициализация адаптера и создание таблицы игроков из БД.
- `onClickListener` – описания событий нажатия кнопок и других элементов на форме.

- `checkField` – проверка корректности данных о команде перед добавлением изменений.

- `checkInfPlayer` - проверка корректности вводимых данных при добавлении или изменении данных о игроке.

- `isCyrillic` – проверка строки на содержание только кириллицы.
- `createRowPlayer` – создание строки игрока в таблице игроков.
- `createRow0` – добавление строки названий столбцов таблицы.
- `marginView` – создание параметров (ширина, высота компонента формы) для строк таблицы.

- `marginRow` - создание параметров (ширина, высота компонента формы) для элементов строки.
- `marginButton` - создание параметров (ширина, высота компонента формы) для кнопок в строке таблицы.

CreateMatchActivity

Поля:

- `workDb` – экземпляр класса `WorkMyDB` для работы с БД.
- `settings` – экземпляр класса `SharedPreferences`, для получения `id` пользователя.
- `idUser` – идентификатор пользователя
- `idTeam1` – идентификатор первой выбранной команды
- `idRepT1` – идентификатор представителя первой выбранной команды
- `idTeam2` – идентификатор второй выбранной команды
- `idRepT2` – идентификатор представителя второй выбранной команды
- `listCurTeam` – список команд которые можно выбрать для матча

Методы:

- `onCreate` – для получения данных из `SharedPreferences` и инициализации необходимых полей.
- `initFields` - инициализация компонентов формы и описания событий нажатия кнопок на форме
- `addInCurrentTeam` – все команды пользователя записываются в таблицу БД `Current_team` для подготовке к редактированию для данного матча.
- `initSpinners` – инициализация `Spinner`-ов с названиями команд

EditCurrentTeamActivity

Поля:

- `workDb` – экземпляр класса `WorkMyDB` для работы с БД.
- `idTeam` – идентификатор изменяемой команды
- `team` – экземпляр класса `Team` для хранения данных о изменяемой команде.

- listPlayers – список игроков команды
- adapter – адаптер для заполнения Spinner содержащий позиции игроков.
- curPlayer – экземпляр класса CurrentPlayer, хранящий информацию о

выбранном игроке из таблицы

Методы:

- onCreate – для получения данных переданных из другой Activity и инициализации необходимых полей.

- initComponents – инициализация компонентов формы
- fillField – инициализация адаптера и создание таблицы игроков из БД.
- onClickListener – описания событий нажатия кнопок и других

элементов на форме.

- checkInfPlayer - проверка корректности вводимых данных при изменении данных об игроке.

- createRowPlayer – создание строки игрока в таблице игроков.
- createRow0 – добавление строки названий столбцов таблицы.
- marginView – создание параметров (ширина, высота компонента

формы) для строк таблицы.

- marginRow - создание параметров (ширина, высота компонента формы) для элементов строки.

- marginButton - создание параметров (ширина, высота компонента формы) для кнопок в строке таблицы.

EditTeamActivity

Поля:

- workDb – экземпляр класса WorkMyDB для работы с БД.
- idTeam – идентификатор изменяемой команды
- team – экземпляр класса Team для хранения данных о изменяемой команде.

- listPlayers – список игроков команды
- adapter – адаптер для заполнения Spinner содержащий позиции игроков.

- curPlayer – экземпляр класса CurrentPlayer, хранящий информацию о выбранном игроке из таблицы

Методы:

- onCreate – для получения данных переданных из другой Activity и инициализации необходимых полей.

- initComponents – инициализация компонентов формы

- fillField – инициализация адаптера и создание таблицы игроков из БД.

- onClickListener – описания событий нажатия кнопок и других элементов на форме.

- isCyrillic - проверка строки на содержание только кириллицы.

- checkInfPlayer - проверка корректности вводимых данных при изменении или добавление данных об игроке.

- createRowPlayer – создание строки игрока в таблице игроков.

- createRow0 – добавление строки названий столбцов таблицы.

- marginView – создание параметров (ширина, высота компонента формы) для строк таблицы.

- marginRow - создание параметров (ширина, высота компонента формы) для элементов строки.

- marginButton - создание параметров (ширина, высота компонента формы) для кнопок в строке таблицы.

AddStaffTrainers

Поля:

- workDb – экземпляр класса WorkMyDB для работы с БД.

- manager – экземпляр класса Representative, для хранения данных о менеджере команды.

- trainer – экземпляр класса Representative, для хранения данных о тренере команды.

- curTeam - экземпляр класса CurrentPlayer, для хранения информации о текущей выбранной команды.

- idManager – идентификатор представителя команды (это может быть как менеджер, так и тренер).
- typeWork – тип работы класса, если это поле равно “edit” значит тип работы с данными – изменение, если нет, то добавление представителей к новой команде.
- cvManager – экземпляр класса ContentValues, для хранения данных о менеджере команды для добавления в БД.
- cvTrainer– экземпляр класса ContentValues, для хранения данных о тренере команды для добавления в БД.
- cvCurTeam - экземпляр класса ContentValues, для хранения идентификатора представителя команды.
- idUser – идентификатор пользователя.
- idTeam - идентификатор команды.
- settings – экземпляр класса SharedPreferences, для получения id пользователя и для передачи id тренера, id менеджера и id представителя команды.

Методы:

- onCreate – для получения данных переданных из SharedPreferences и другой Activity.
- initFields – инициализация компонентов формы
- fillField – инициализация адаптера и создание таблицы игроков из БД.
- lisener – описания событий нажатия кнопок и других элементов на форме.
- checkManadger – проверка введенной информации о менеджере команды.
- checkTrainer – проверка введенной информации о тренере команды.
- isCyrillic - проверка строки на содержание только кириллицы.

InfAboutMatchActivity

Поля:

- workDb – экземпляр класса WorkMyDB для работы с БД.

- idUser – идентификатор пользователя
- idMatch – идентификатор матча
- match – экземпляр класса Match для хранения информации о матче.

Методы:

- onCreate – для получения данных переданных из другой Activity.
- initFields – инициализация компонентов формы.
- listener – описания событий нажатия кнопок и других элементов на форме.

MainActivity

Поля:

- sourceA – очки текущей партии команды А.
- sourceB – очки текущей партии команды Б.
- sourcePartA – количество выигранных партий команды А.
- sourcePartB – количество выигранных партий команды Б.
- listBall – список очередности забитых мячей.
- whoseBall – пометка какая команда начинала подавать.
- numPart – номер партии.

Методы:

- onCreate – инициализация элементов формы
- endParty – проверка конца партии.
- changeOfSides – смена сторон команд.
- startTimer – включение таймера при окончании партий.

MatchActivity

Поля:

- workDb – экземпляр класса WorkMyDB для работы с БД.
- recordProtocole – флаг для протокола (вести или нет).
- match – экземпляр класса Match для хранения информации о матче.
- curTeam1 - экземпляр класса CurrentTeam, для хранения информации о первой команде.

- curTeam2 - экземпляр класса CurrentTeam, для хранения информации о второй команде.

- sourceA – количество очков команды 1 в текущей партии.
- sourceB – количество очков команды 2 в текущей партии.
- sourcePartA – количество партий команды 1.
- sourcePartB – количество партий команды 2.
- listBall – список очередности забитых мячей по текущей партии.
- listStatistics - список очередности забитых мячей по матчу.
- whoseBall – пометка какая команда начинала подавать.
- numPart – номер партии.
- listRezultParts – список результатов партий.
- curMatch - экземпляр класса CurrentMatch для хранения информации о текущем матче.

- curPITeam1 – список игроков команды 1.
- curPITeam2 – список игроков команды 2.
- numCurPIT1 - список номеров игроков команды 1.
- numCurPIT2 - список номеров игроков команды 2.
- whoseReplace – название команды взявшая замену.
- flagTimeOutT1_1 – флаг на взятие первого таймаута первой команды.
- flagTimeOutT1_2 – флаг на взятие второго таймаута первой команды.
- flagTimeOutT2_1 – флаг на взятие первого таймаута второй команды.
- flagTimeOutT2_2 – флаг на взятие второго таймаута второй команды.

Методы:

- onCreate – для получения данных переданных из другой Activity.
- initFields – инициализация полей для формы.
- lisener - описание событий нажатия кнопок.
- startTimeOut – отсчет таймера при взятии таймаута.
- checkReceivedData – занесение данных о матче.

- zoneActive – метод для активации или деактивации зон для занесения стартовых составов команд.

- allBtn - метод для активации или деактивации кнопок для ведения матча.

- plMoveForwardTeam1 – переход первой команды на зону вперед.
- plMoveForwardTeam2 – переход второй команды на зону вперед.
- plMoveBackTeam1 – переход первой команды на зону назад.
- plMoveBackTeam2 – переход второй команды на зону назад.
- startTimer – включение таймера при окончании партий.
- endParty – проверка конца партии.
- changeOfSides – смена сторон команд.
- endMatch – проверка и проведение конца матча.
- recordProtocole – составление протокола матча.
- paramsEditText - создание параметров для зон игроков (номер строки, номер колонки для GridLayout).

- removeView – удаление элемента из родительского компонента.

Опишем дополнительные классы для работы с базой данных.

DatabaseHelper

Поля:

- DB_NAME – имя базы данных.
- DB_PATH – путь к базе данных.
- DB_VERSION – версия БД.
- mDataBase – статус открытия базы данных.
- mContext - объект, который предоставляет доступ к базовым функциям приложения.

- mNeedUpdate – флаг проведения изменений в БД.

Методы:

- DatabaseHelper – конструктор класса.
- updateDataBase – проведение изменений в БД.

- checkDataBase – проверка соединения с БД.
- copyDataBase – копирования данных из БД.
- copyDBFile – копирование файла базы данных.
- openDataBase – открытие соединения с БД.
- close - закрытие соединения с БД.
- onUpgrade – увеличение версии БД.

WorkMyDB

Поля:

- mDBHelper – экземпляр класса DatabaseHelper.
- mDb - экземпляр класса SQLiteDatabase для организации запросов к БД.
- context - объект, который предоставляет доступ к базовым функциям приложения.
- idUser – идентификатор пользователя.

Методы:

- WorkMyDB – конструктор класса для создания соединения к БД.
- addUser – добавление нового пользователя.
- selectUserId – поиск пользователя по идентификатору.
- auth – проверка есть ли пользователь с введенными логином и паролем в БД.
- updateUser – изменение данных пользователя.
- checkDataUser – проверка вводимых данных, чтобы логин, email и номер телефона не совпадал с другими пользователями и старый пароль был введен правильно.
- addTeam – добавление команды.
- selectLastTeam – поиск последней добавленной команды.
- selectAllTeamByIdUser – поиск команд определенного пользователя.
- selectTeambyId – поиск команды по идентификатору.
- updateTeam – внесение изменений команд.

- selectRepresentativeById – поиск представителя команды по идентификатору.
- selectRepresentativeByIdTeam – поиск менеджера или тренера по идентификатору команды.
- addRepresentative – добавление представителя команды.
- updateRepresentative – изменение данных о представителе команды.
- selectRepresentativeLastId – поиск последнего добавленного представителя.
- updatePlayer – изменение данных об игроке.
- selectPlayersByIdTeam – поиск игроков по идентификатору команды.
- selectPlayersById – поиск игрока по идентификатору.
- addMatch – создание матча.
- selectMatchById – поиск матча по идентификатору.
- addDescMatch – добавление описания матча.
- selectMatchesByIdUser – поиск матчей по идентификатору пользователя.
- deleteAllCurTeam – удаление всех записей из таблицы Current_team.
- addCurPlayer – добавление текущего игрока в текущую команду.
- addCurrentTeam – добавление текущей команды.
- selectIDCurTeam – поиск идентификатора последней добавленной текущей команды.
- selectCurPlayerByIdPlayer – поиск текущего игрока по идентификатору игрока.
- selectCurPlayerByIdCurTeam – поиск текущих игроков по идентификатору текущей команды.
- selectAllCurTeamByIdUser – поиск текущих команд по идентификатору пользователя.
- selectCurTeamById – поиск текущих команд по идентификатору.

- selectCurTeamByIdTeam - поиск текущих команд по идентификатору команды.
- addCurPlayersByIdCurTeam – добавление текущего игрока в текущую команду.
- updateManagerCurTeam – изменение представителя команды на текущий матч.

User

Поля:

- id – идентификатор пользователя.
- login – логин
- password – пароль
- email – электронная почта
- phoneNum – номер телефона

User – конструктор класса, принимающий логин, пароль, почту и номер телефона.

Team

Поля:

- id – идентификатор команды.
- idUser – идентификатор пользователя.
- idTrain – идентификатор тренера.
- idManage – идентификатор менеджера.
- name – название команды.
- address – адрес команды.

Данный класс содержит 2 конструктора. Первый не принимает никаких параметров, второй – все поля класса.

Representative

Поля:

- id – идентификатор представителя команды.
- lastname – фамилия

- firstname – имя
- patronymic - отчество
- email – электронная почта
- phoneNum – номер телефона
- jobTitle – должность

Данный класс содержит 2 конструктора. Первый не принимает никаких параметров, второй – все поля класса.

Player

Поля:

- id – идентификатор игрока команды.
- idTeam – идентификатор команды.
- lastname – фамилия
- firstname – имя
- patronymic - отчество
- numPlayer – игровой номер
- position – позиция
- fio – ФИО

Данный класс содержит 3 конструктора. Первый не принимает никаких параметров, второй – все поля класса, третий – все поля кроме идентификатора.

Match

Поля:

- id – идентификатор матча.
- team1 – экземпляр класса CurrentTeam для команды 1.
- team2 – экземпляр класса CurrentTeam для команды 2.
- description – описание матча.
- date – дата проведения матча

Конструктор класса принимает все поля класса.

CurrentTeam

Поля:

- id – идентификатор текущей команды.
- team – экземпляр класса Team.
- listPlayers – список игроков команды.
- idManage – идентификатор представителя.
- tecScorePart – текущее количество выигранных партий.
- countTimeOut - текущее количество взятых таймаутов.
- countReplacement - текущее количество взятых замен.

Данный класс содержит 2 конструктора. Первый не принимает никаких параметров, второй принимает такие параметры как id, team и idManage.

CurrentPlayer

Поля:

- id – идентификатор текущего игрока.
- player – экземпляр класса Player.
- position – позиция игрока на текущий матч.

Данный класс содержит 2 конструктора. Первый не принимает никаких параметров, второй принимает такие параметры как player и position.

CurrentMatch

Поля:

- id – идентификатор текущего игрока.
- placement – флаг вести учет замен или нет.
- protocol – флаг составлять протокол или нет.

Конструктор класса принимает все поля класса.

4. Руководство пользователя

Основным функционалом разрабатываемого приложения является проведение тренировок и матчей.

Для того что бы начать тренировку на главной странице необходимо нажать кнопку «Начать тренировку» (см. рис. 7-8).



Рисунок 7 - главная страница, для начала тренировки нажмите кнопку "Начать тренировку"

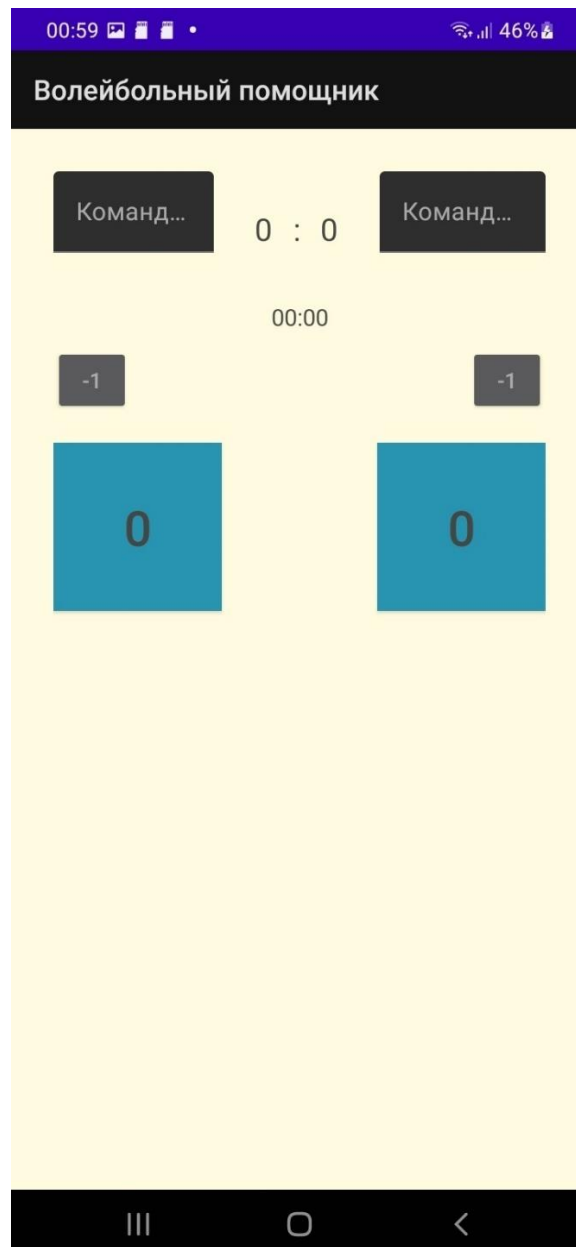


Рисунок 8 - начало тренировки

В данном режиме можно назвать команды как угодно в определенных полях (см. рис. 9).

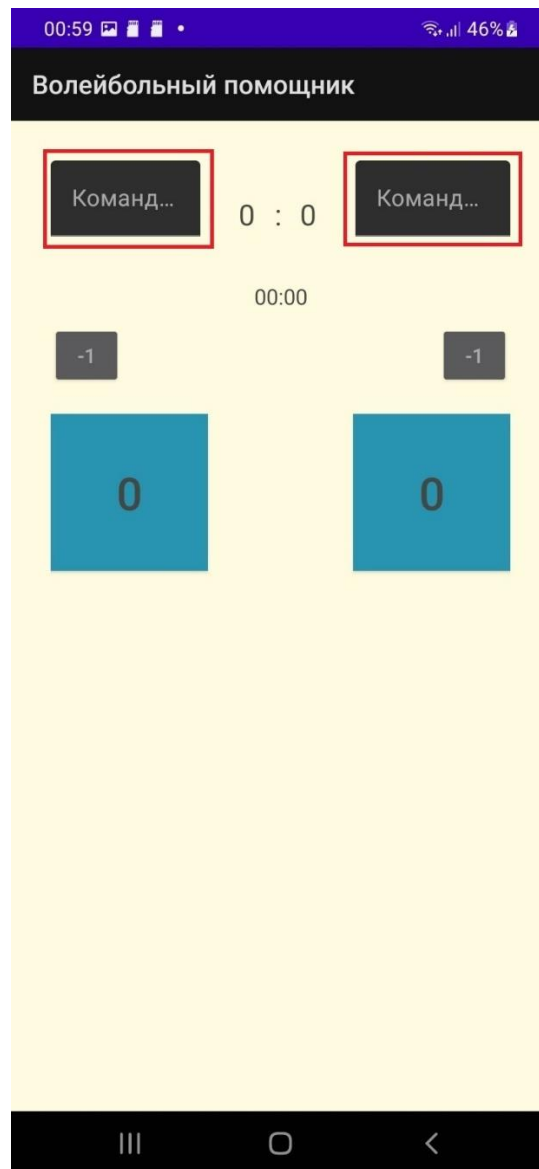
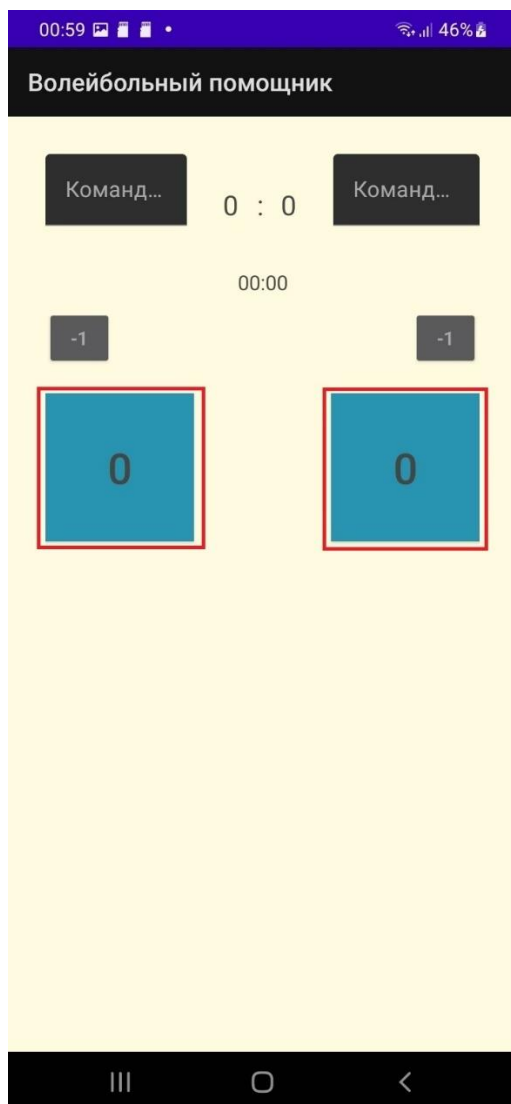
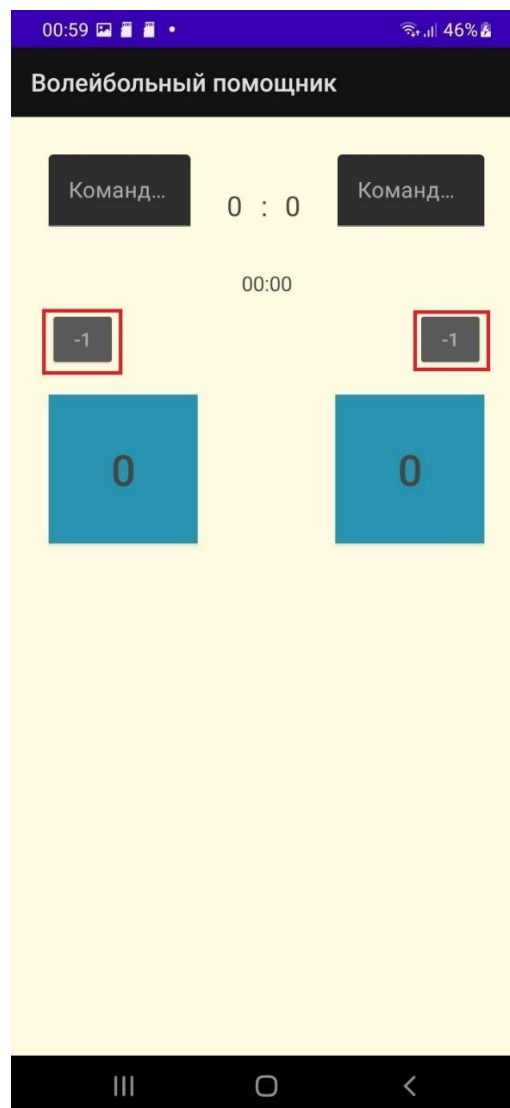


Рисунок 9 - зачисление очков командам

Для зачисления или уменьшения очков нужно нажать соответствующие кнопки (см. рис. 10).



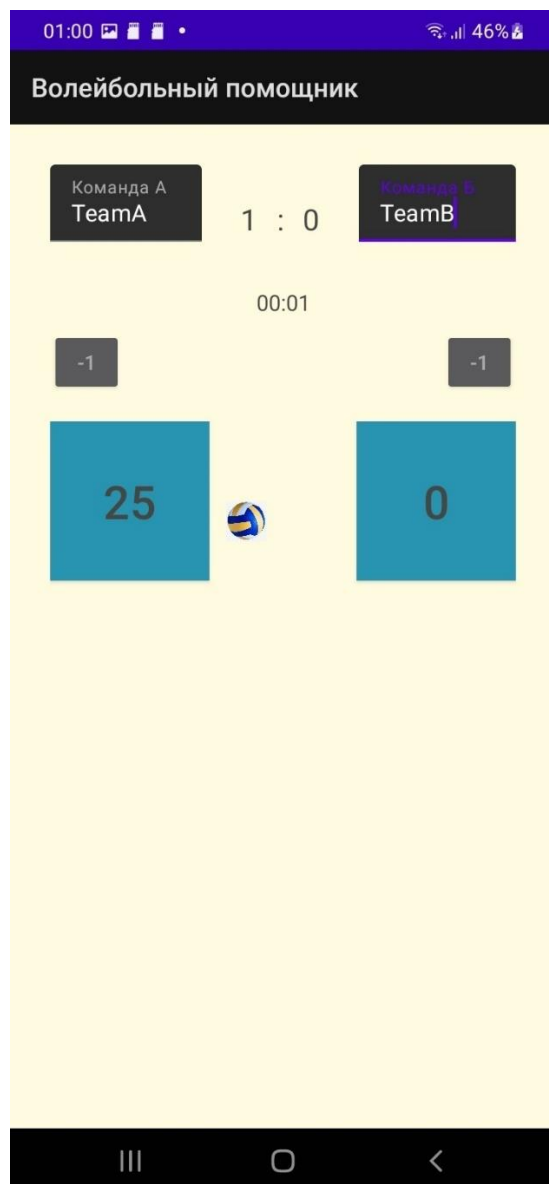
а)



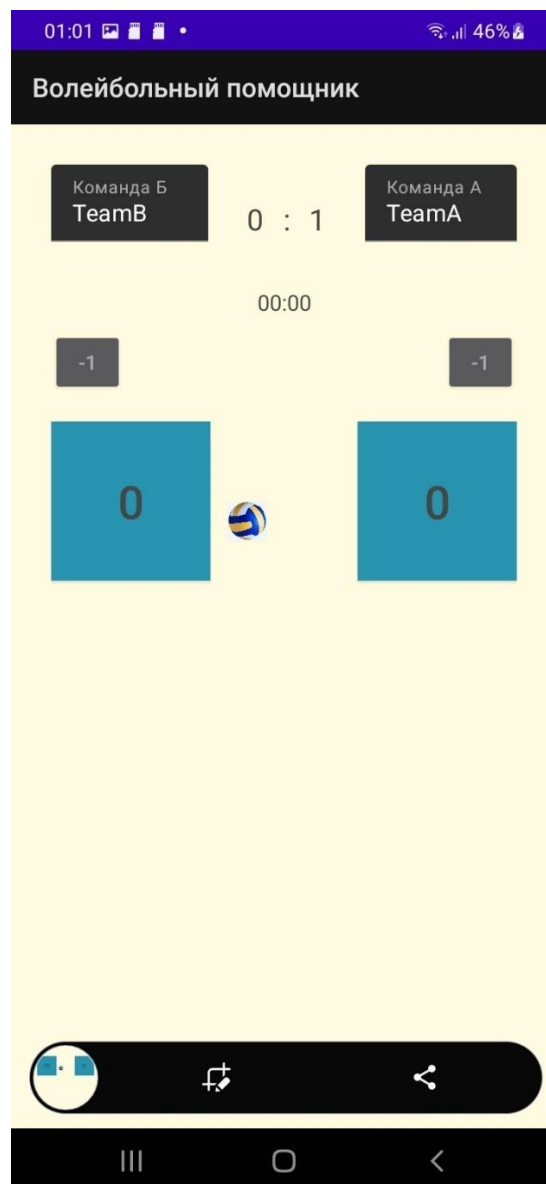
б)

Рисунок 10 - кнопки зачисления или уменьшения очков

При окончании партии происходит смена сторон (см. рис. 11).



а)

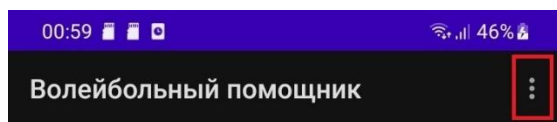


б)

Рисунок 11 – смена сторон

В данном режиме нет ограничений по количеству партий.

Для того что бы начать матч необходимо авторизоваться. На главной странице необходимо нажать кнопку слева сверну на панели управления (см. рис. 12) и выбрать элемент «Вход/Выход».

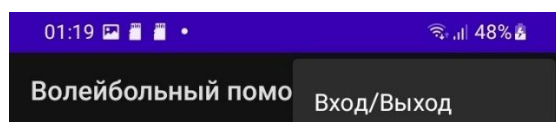


Начать матч

Начать тренировку



а)



Начать матч

Начать тренировку



б)

Рисунок 12 – меню приложения

На странице авторизации пользователь может как авторизоваться, так и зарегистрироваться (см. рис. 13).

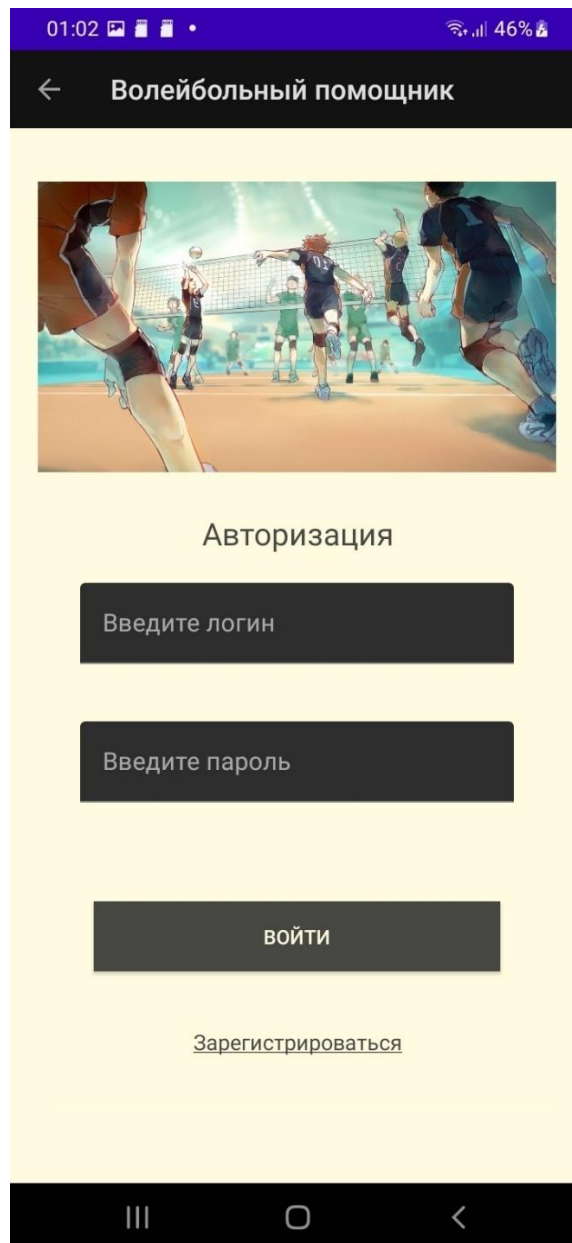


Рисунок 13 - страница авторизация

Для регистрации необходимо нажать «Зарегистрироваться» (рис. 14).

01:02 46%

← Волейбольный помощник

Регистрация

Введите логин

Введите пароль

Введите электронную почту

Введите номер телефона

ЗАРЕГИСТРИРОВАТЬСЯ

а)

01:02 46%

← Волейбольный помощник

Регистрация

Введите логин
user3

Введите пароль
.....

Введите электронную почту
user3@gmail.com

Введите номер телефона
89000000003

ЗАРЕГИСТРИРОВАТЬСЯ

б)

Рисунок 14 – регистрация пользователя

Далее необходимо продолжить авторизацию (см. рис. 15).

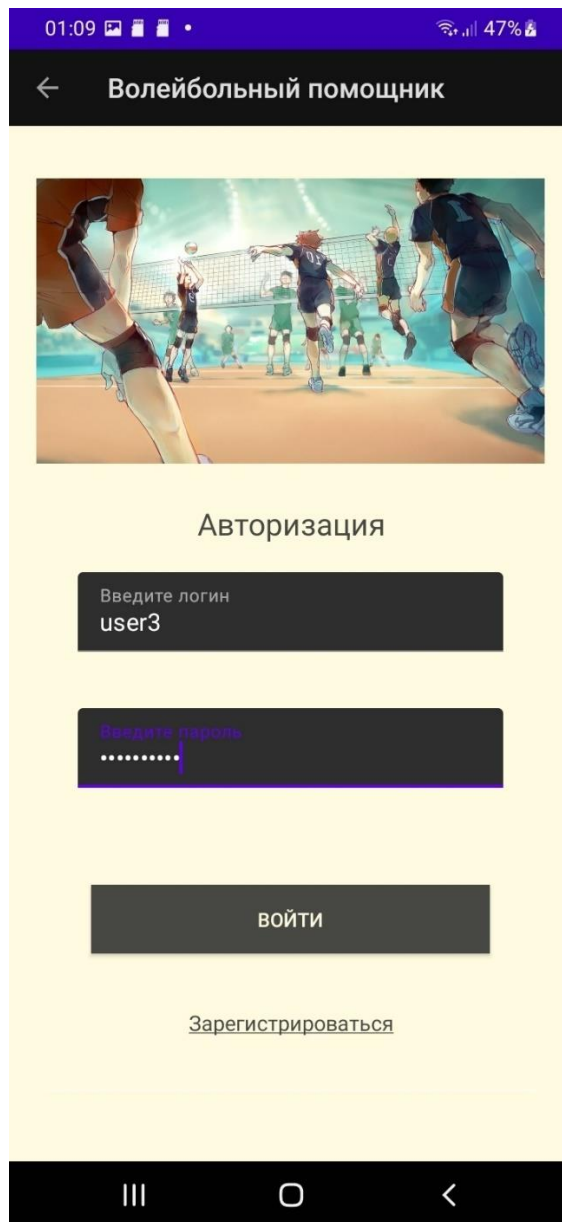
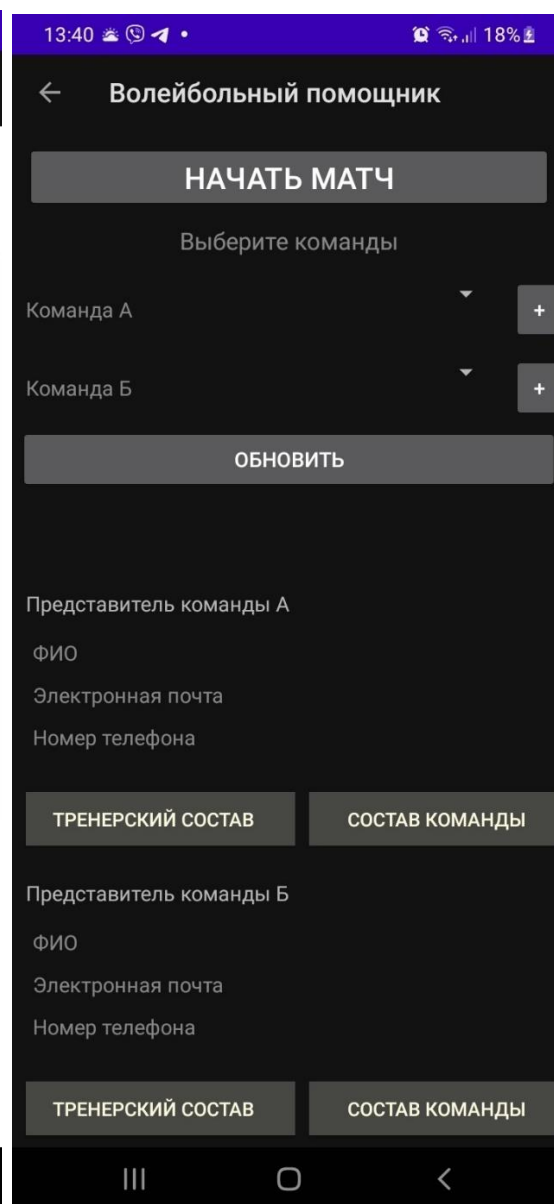


Рисунок 15 - авторизация нового пользователя

Для начала матча необходимо нажать кнопку «Начать матч» (см. рис. 16).



а)



б)

Рисунок 16 – начало матча

Далее необходимо добавить команды (см. рис. 17).

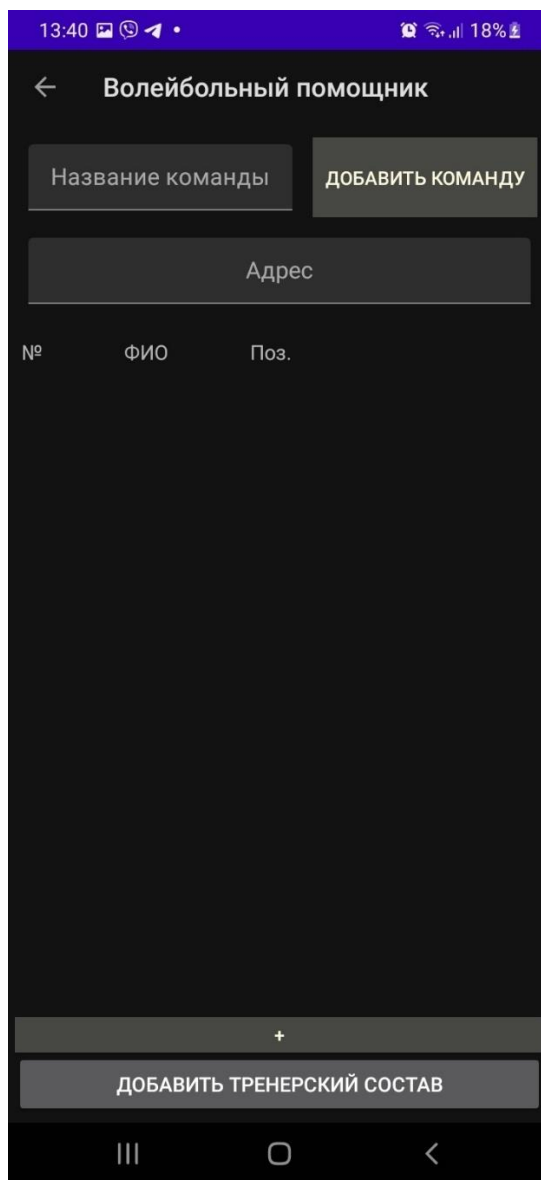


Рисунок 17 – добавление команды

Добавить игроков можно нажав кнопку «+», после занесения необходимых данных нажать кнопку «Добавить игрока» (см. рис. 18) и игрок будет добавлен в список.

Рисунок 18 – добавление игрока

Для добавления команды так же необходимо добавить хотя бы одного представителя (менеджера и/или тренера). Для этого необходимо нажать кнопку «Добавить тренерский состав». На форме пользователь может добавить менеджера или тренера нажав в соответствующем месте кнопку «Редактировать». После занесения необходимых данных нажать кнопку «Добавить» (см. рис. 19). Далее для завершения добавления команды нужно нажать кнопку «Добавить команду».

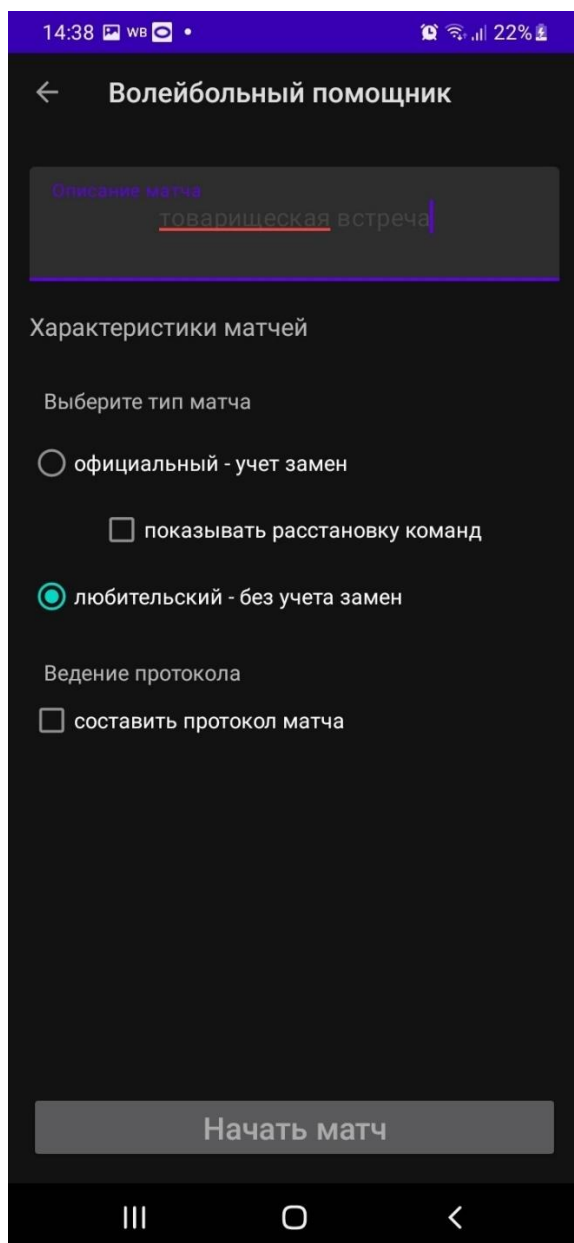
а)

№	ФИО	Поз.	ИЗМ	X
1	Тулинова Е.Ю.	К	ИЗМ	X
2	Иванова Е.	Л	ИЗМ	X
3	Петрова Е.	Л	ИЗМ	X
4	Жидкова Н.	И	ИЗМ	X
5	Любимова Н.	И	ИЗМ	X
6	Хохлова Н.Н.	И	ИЗМ	X
7	Хохлова Э.Н.	И	ИЗМ	X

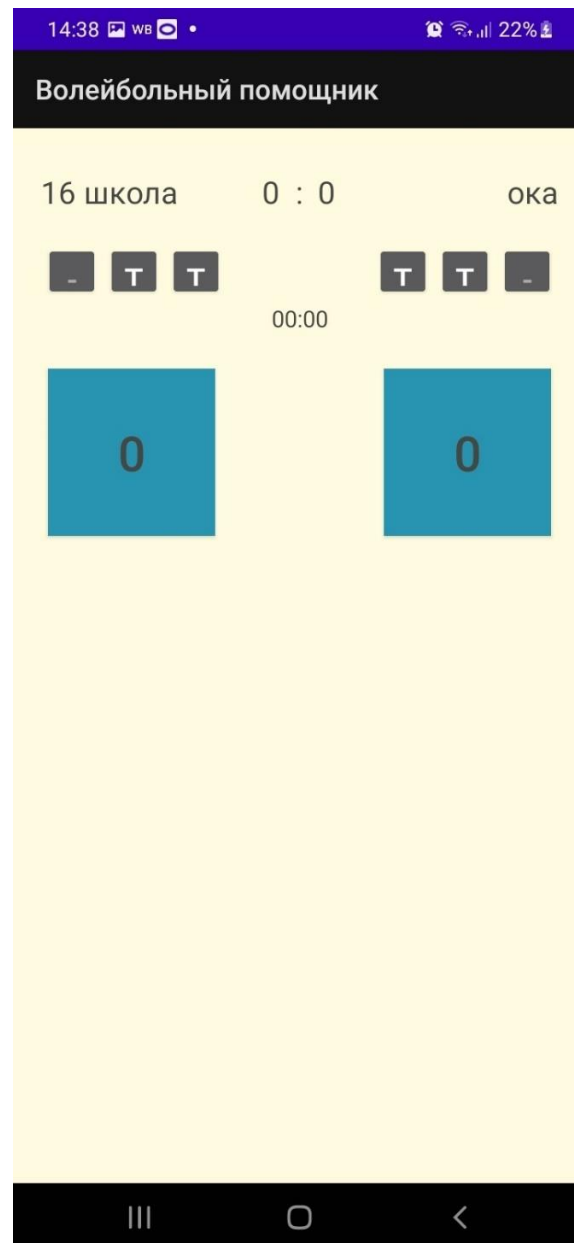
б)

Рисунок 18 – добавление представителей и команды

После добавления команд надо нажать «Начать матч» для продолжения создания матча. Далее необходимо выбрать тип матча (официальный с учетом замен и прослеживанием расстановок, любительский без замен) и нужно ли вести протокол (в протокол матча заносятся составы команд, результаты по партиям и результат матча). Для того чтобы продолжить – кнопка «Начать матч» (см. рис. 20).



а)



б)

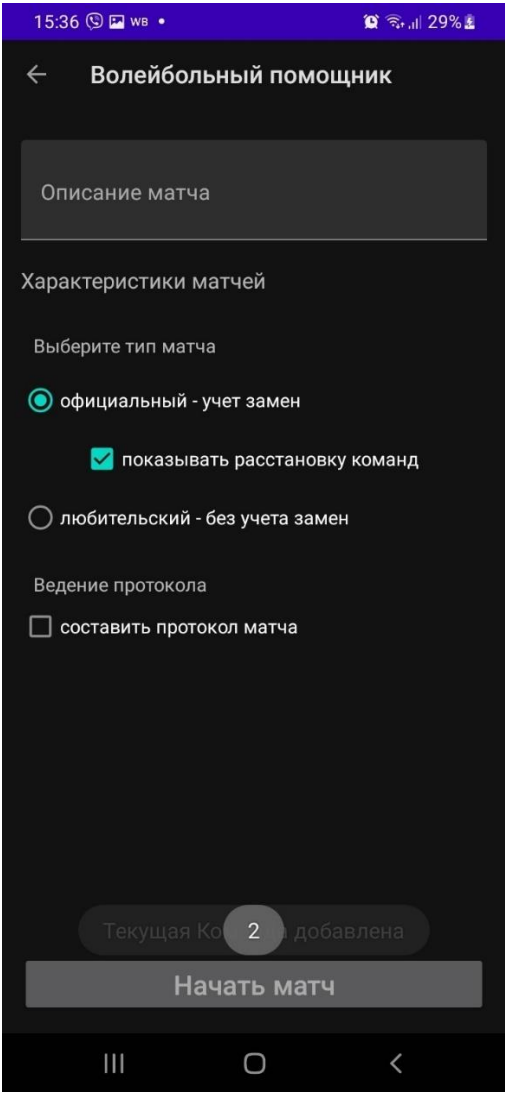
Рисунок 19 – начало матча

Проведение матча в любительском режиме отличается от тренировки только возможностью брать командами таймауты и ограниченное количество партий (их максимально 5).

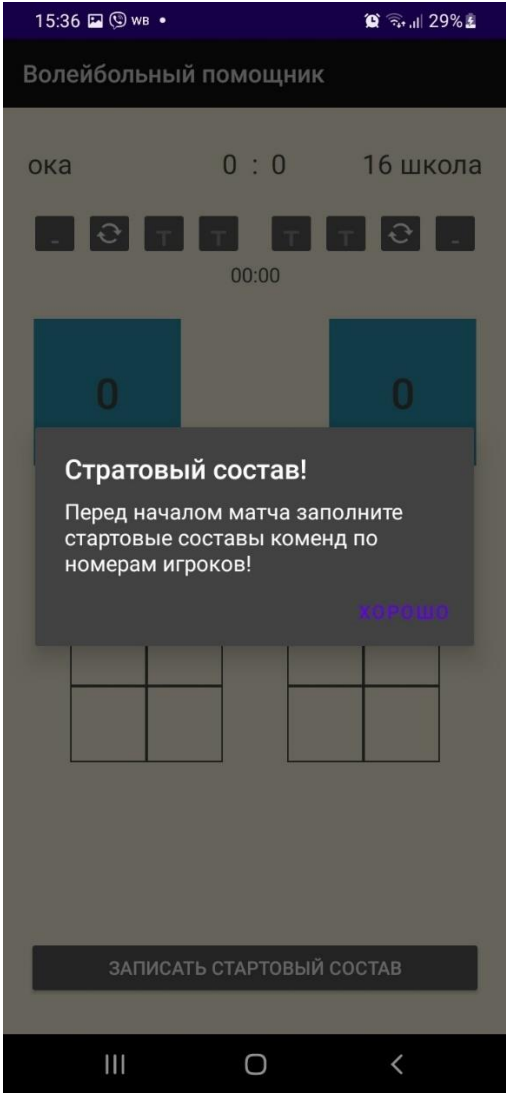
При взятии тайм аута все кнопки на форме неактивны после его окончания выбранный таймаут “сгорает” – в этой партии брать его больше нельзя.

Если был выбран тип матча как официальный, то функционал расширяется. К возможностям любительского режима добавляется просмотр расстановок и

проведение замен. В начале такого матча необходимо занести стартовые составы команд (см. рис. 21).



а)



б)

Рисунок 21 – начало официального матча

Для занесения расстановок нужно записать номера игроков в соответствующие поля и нажать кнопку «Записать стартовый состав» (см. рис. 22).

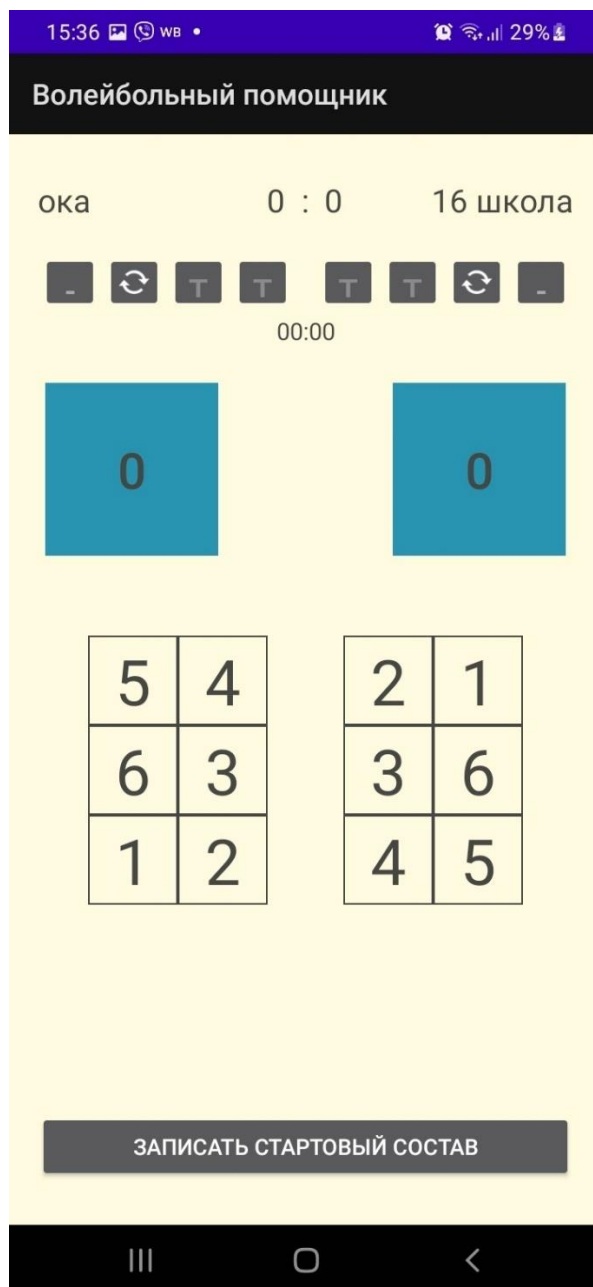
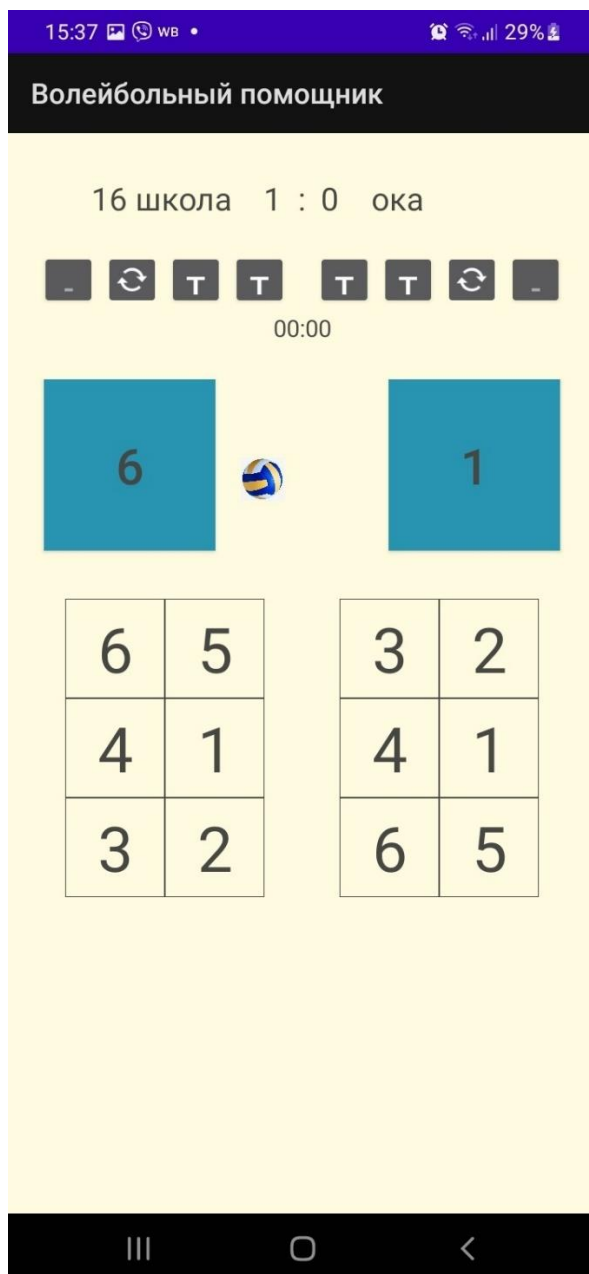


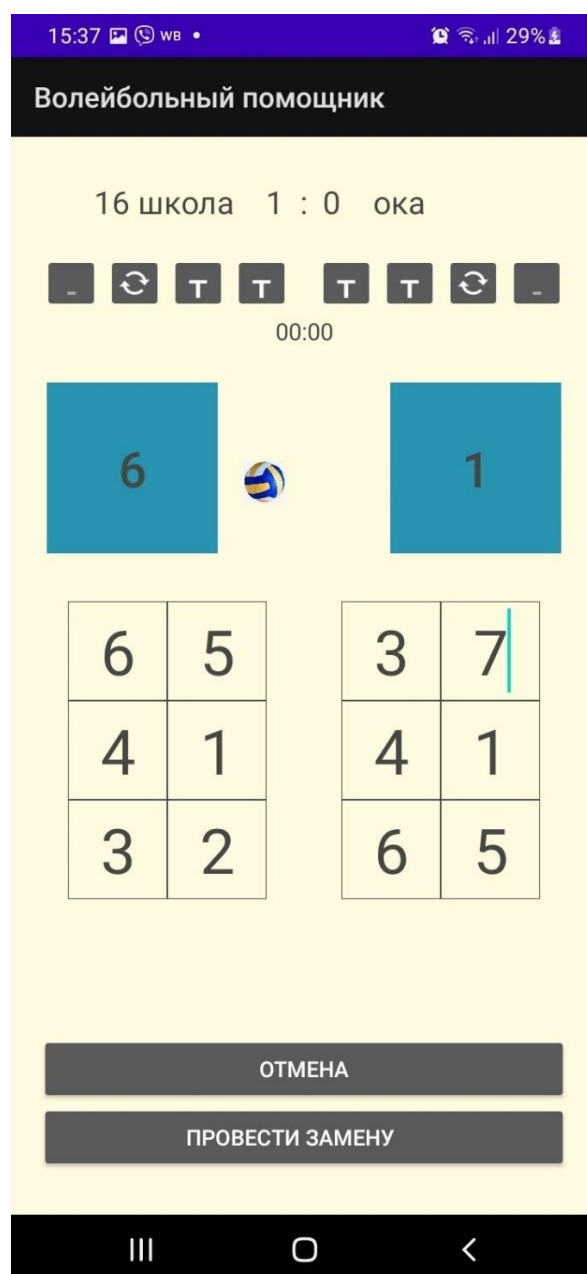
Рисунок 22 – запись стартового состава

Запись стартового состава проходит в начале каждой партии.

Для того что бы провести замену игрока у определенной команды нужно нажать кнопку замены под названием команды, поменять номер игрока в нужной зоне и нажать кнопку «Провести замену», если пользователь хочет отменить взятую замену, то нужно нажать кнопку «Отмена» (см. рис. 23).



а)



б)

Рисунок 23 – проведение замены

По окончании матча если было выбрано, то на устройство будет сохраняться протокол матча. Пример показан на рисунке 24.

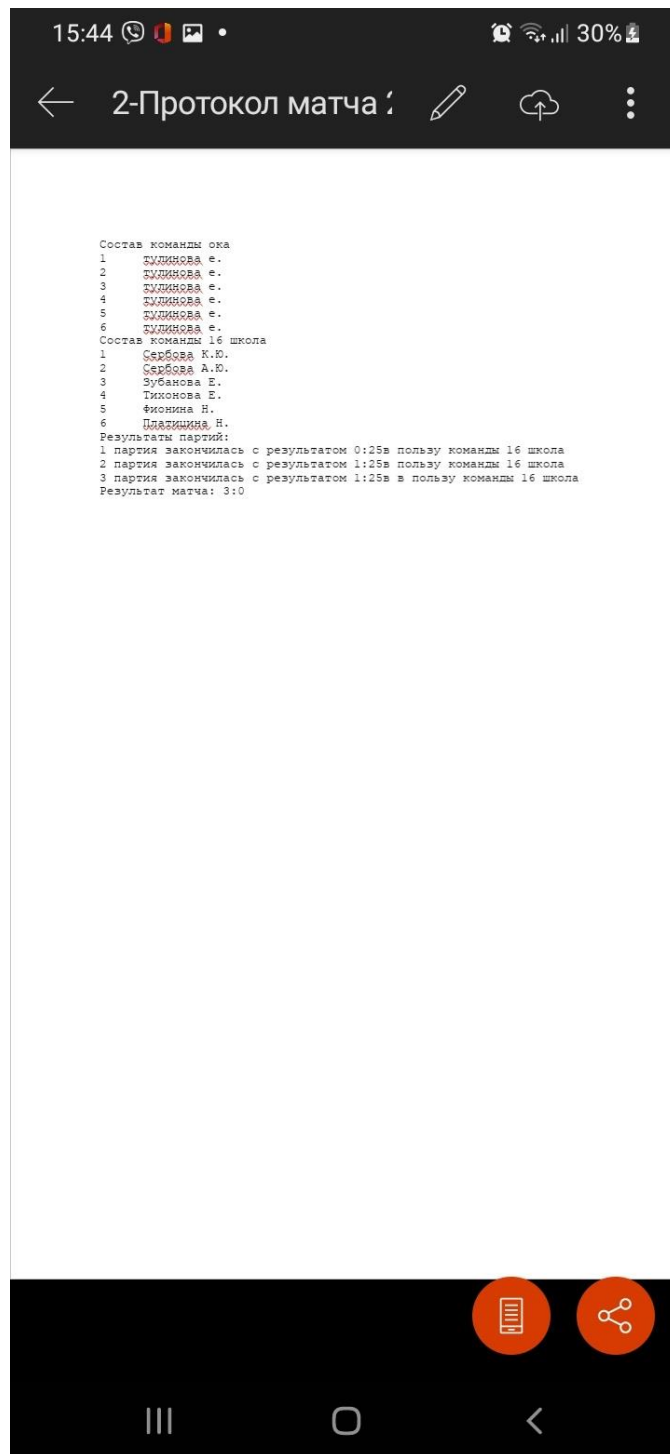


Рисунок 24 – протокол матча

5. Тестирование

Тестирование программного обеспечения — процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом.

Таблица 7 - тестирование приложения

Тест	Ожидаемый результат	Полученный результат
Ввод данных о пользователе при регистрации		
Ввод фамилии с цифрами и/или латинским алфавитом.	Вывод ошибки.	Вывод сообщения об ошибке «Некорректный ввод электронной почты».
Ввод фамилии, содержащий только кириллицу.	Ввод корректных данных.	Принятие корректного ввода фамилии и продолжение проверки других данных.
Ввод имени с цифрами и/или латинским алфавитом.	Вывод ошибки.	Вывод сообщения об ошибке «Некорректный ввод электронной почты».
Ввод имени, содержащий только кириллицу.	Ввод корректных данных.	Принятие корректного ввода имени и продолжение проверки других данных.
Ввод отчества с цифрами и/или латинским алфавитом.	Вывод ошибки.	Вывод сообщения об ошибке «Некорректный ввод электронной почты».

Ввод отчества, содержащий только кириллицу или не введено вообще.	Ввод корректных данных.	Принятие корректного ввода отчества и продолжение проверки других данных.
Некорректный ввод электронной почты или не введены данные о почте.	Вывод ошибки.	Вывод сообщения об ошибке «Некорректный ввод электронной почты».
Ввод электронной почты, которая уже занята другим пользователем.	Вывод ошибки.	Вывод сообщения об ошибке «Электронная почта занята другим пользователем».
Корректный ввод электронной почты.	Ввод корректных данных.	Принятие корректного ввода электронной почты и продолжение проверки других данных.
Некорректный ввод номера телефона.	Вывод ошибки.	Вывод сообщения об ошибке «Некорректный ввод номера телефона».
Ввод номера телефона, который уже занят другим пользователем.	Вывод ошибки.	Вывод сообщения об ошибке «Номер телефона занят другим пользователем».
Корректный ввод номера телефона или не введен.	Ввод корректных данных.	Принятие корректного ввода номера телефона и продолжение проверки других данных.
Авторизация пользователя		

Ввод данных несоответствующих ни одному пользователю в системе.	Ошибка авторизации пользователя.	Вывод сообщение об ошибке: «Ошибка авторизации».
Ввод данных соответствующих пользователю в системе.	Успешная авторизация пользователя.	Успешная авторизация пользователя.
Ввод данных об игроке.		
Ввод фамилии с цифрами и/или латинским алфавитом.	Вывод ошибки.	Поле становится красного цвета и выводится сообщение об ошибке «Некорректный ввод электронной почты».
Ввод фамилии, содержащий только кириллицу.	Ввод корректных данных.	Принятие корректного ввода фамилии и продолжение проверки других данных.
Ввод имени с цифрами и/или латинским алфавитом.	Вывод ошибки.	Поле становится красного цвета и выводится сообщение об ошибке «Некорректный ввод электронной почты».
Ввод имени, содержащий только кириллицу.	Ввод корректных данных.	Принятие корректного ввода имени и продолжение проверки других данных.

Ввод отчества с цифрами и/или латинским алфавитом.	Вывод ошибки.	Поле становится красного цвета и выводится сообщение об ошибке «Некорректный ввод электронной почты».
Ввод отчества, содержащий только кириллицу или не введено вообще.	Ввод корректных данных.	Принятие корректного ввода отчества и продолжение проверки других данных.
Выбор позиции игрока как Капитана, когда в команде уже назначен капитан.	Вывод ошибки.	Вывод сообщения об ошибке «Нельзя назначить больше одного капитана».
Выбор позиции игрока как Капитана, когда в команде еще не назначен капитан.	Позиция игрока – Капитан.	Позиция игрока – Капитан.
Выбор позиции игрока как Либера, когда в команде уже назначены 2 и более игрока с такой позицией.	Вывод ошибки.	Вывод сообщения об ошибке «Нельзя назначить больше двух либеро».
Выбор позиции игрока как Либера, когда в команде назначены меньше 2 игроков с такой позицией.	Позиция игрока – Либера.	Позиция игрока – Либера.
Выбор позиции игрока как Игрок.	Позиция игрока – Игрок.	Позиция игрока – Игрок.

Ввод игрового номера игрока, который уже занят другим игроком данной команды.	Вывод ошибки.	Вывод сообщения об ошибке «Игровые номера индивидуальны».
Ввод игрового номера игрока, который еще не занят другим игроком данной команды.	Назначение игроку данный игровой номер.	Назначение игроку данный игровой номер.
Ввод данных о команде		
Название команды не заполнено.	Вывод ошибки.	Вывод сообщения об ошибке «Добавьте название команды».
Ввод названия команды.	Добавление названия команды.	Добавление названия команды.
Адрес команды не заполнено.	Вывод ошибки.	Вывод сообщения об ошибке «Добавьте адрес команды».
Ввод адреса команды.	Добавление адреса команды.	Добавление адреса команды.
Данные о представителе команды не занесены.	Вывод ошибки.	Вывод сообщения об ошибке «Добавьте представителя команды».
Данные о представителе команды занесены.	Добавление данных представителя в команду.	Добавление данных представителя в команду.
Ввод данных о представителях команд		
Ввод фамилии с цифрами и/или латинским алфавитом.	Вывод ошибки.	Поле становится красного цвета и выводится сообщение об ошибке

		«Некорректный ввод фамилии».
Ввод фамилии, содержащий только кириллицу.	Ввод корректных данных.	Принятие корректного ввода фамилии и продолжение проверки других данных.
Ввод имени с цифрами и/или латинским алфавитом.	Вывод ошибки.	Поле становится красного цвета и выводится сообщение об ошибке «Некорректный ввод имени».
Ввод имени, содержащий только кириллицу.	Ввод корректных данных.	Принятие корректного ввода имени и продолжение проверки других данных.
Ввод отчества с цифрами и/или латинским алфавитом.	Вывод ошибки.	Поле становится красного цвета и выводится сообщение об ошибке «Некорректный ввод отчества».
Ввод отчества, содержащий только кириллицу или не введено вообще.	Ввод корректных данных.	Принятие корректного ввода отчества и продолжение проверки других данных.
Некорректный ввод электронной почты.	Вывод ошибки.	Вывод сообщения об ошибке «Некорректный ввод электронной почты».
Корректный ввод электронной почты или	Ввод корректных данных.	Принятие корректного ввода электронной почты

не введены данные о почте.		и продолжение проверки других данных.
Некорректный ввод номера телефона.	Вывод ошибки.	Вывод сообщения об ошибке «Некорректный ввод номера телефона».
Корректный ввод номера телефона или не введен.	Ввод корректных данных.	Принятие корректного ввода номера телефона и продолжение проверки других данных.
Данные о почте и номере телефона не занесены.	Вывод ошибки.	Вывод сообщения об ошибке «Не занесены данные о номере телефона или электронной почте».
Выбор команд для матча		
Выбор 2 одинаковых команд	Вывод ошибки.	Вывод сообщения об ошибке «Команды одинаковые».
Выбор 2 разных команд	Создание матча.	Создание матча.
Занесение стартовых составов команд		
Занесение в зону номера игрока, которого нет в составе команд.	Вывод ошибки.	Вывод сообщения об ошибке «У команды не соответствуют номера».
Номера игроков введены корректно.	Сохранение стартового состава команды на данную партию.	Сохранение стартового состава команды на данную партию.
Проведение замен		

Занесение в зону номера игрока, которого нет в составе команд.	Вывод ошибки.	Вывод сообщения об ошибке «У команды не соответствуют номера».
Занесение в зону номера игрока, который есть в стартовом составе.	Вывод ошибки.	Вывод сообщения об ошибке «У команды не соответствуют номера».
Номера игроков введены корректно.	Сохранение состава команды на данную партию.	Сохранение состава команды на данную партию.

Скриншоты вывода ошибок показаны в Приложении 1.

Заключение

В данной курсовой работе было разработано приложение «Волейбольный помощник», отвечающее всем требованиям технического задания.

Были реализованы такие функции как:

1. Контроль игрового процесса, включающего в себя ведение счета очков, замен и таймаутов.
2. Удобный, интуитивно понятный интерфейс.
3. 2 уровнями доступа, первый для тренировок, второй для матчей (любительских и профессиональных).
4. Две роли пользователей (зарегистрированный и незарегистрированный) с разными уровнями доступа.
5. Добавление, редактирование команд зарегистрированного пользователя, редактирование составов команд на текущий матч.
6. Добавление и редактирование данных о представителях команд.
7. Назначение представителей команд на текущий матч.
8. Автоматическое создание протокола матча по его окончании.

В ходе разработки использовалась база данных SQLite, язык программирования Kotlin и среда разработки Android Studio.

Список использованных источников

1. Оценка по волейболу простая [Электронный ресурс].URL: https://translated.turbopages.org/proxy_u/en-ru.ru.88346903-634e59ac-628b5d4d-74722d776562/https/m.apkpure.com/volleyball-score-simple/com.wespiapps.volleyballscore (Дата обращения 18.10.2022)
2. Счет в волейболе [Электронный ресурс]: URL: <https://m.apkpure.com/jp/счет-в-волейболе/com.sakha.volleyballscoresitis> (Дата обращения 18.10.2022)
3. Моделирование данных: зачем нужно и как реализовать [Электронный ресурс]: URL: <https://habr.com/ru/post/554388/> (Дата обращения: 20.10.2022)
4. Концептуальная модель [Электронный ресурс]: URL: [http://wikiredia.ru/wiki/Концептуальная_модель#:~:text=Концептуальная%20модель%20\(англ.%20conceptual%20model\),модель%20—%20модель%20предметной%20области](http://wikiredia.ru/wiki/Концептуальная_модель#:~:text=Концептуальная%20модель%20(англ.%20conceptual%20model),модель%20—%20модель%20предметной%20области) (Дата обращения: 20.10.2022)
5. Логическая модель [Электронный ресурс]: URL: <https://teletype.in/@reddo/SJsmWPnpV#:~:text=Логическая%20модель%20—%20графическое%20представление,базы%20данных%20и%20аппаратной%20платформы> (Дата обращения: 20.10.2022)

Приложение 1. Код программы

Код программы предоставлен в репозитории по ссылке:

					МИВУ.09.03.04-4.000 ПЗ	Лист
						66
Изм	Лист	№ докум.	Подп.	Дата		

Приложение 2. Скриншоты программы

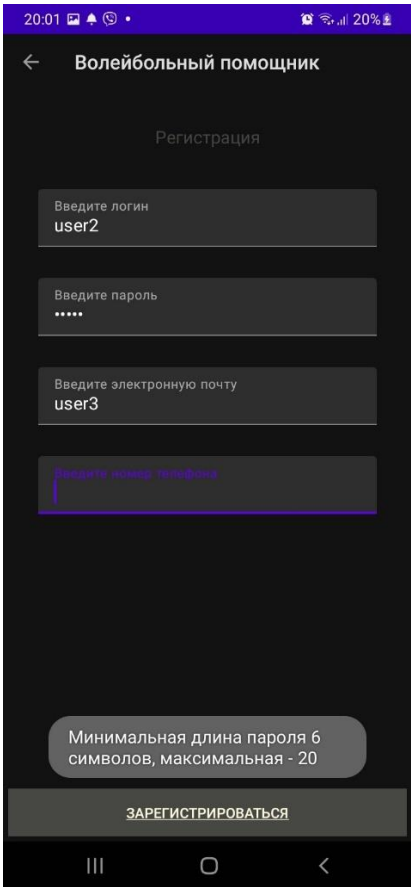


Рисунок 1 - ошибка ввода пароля

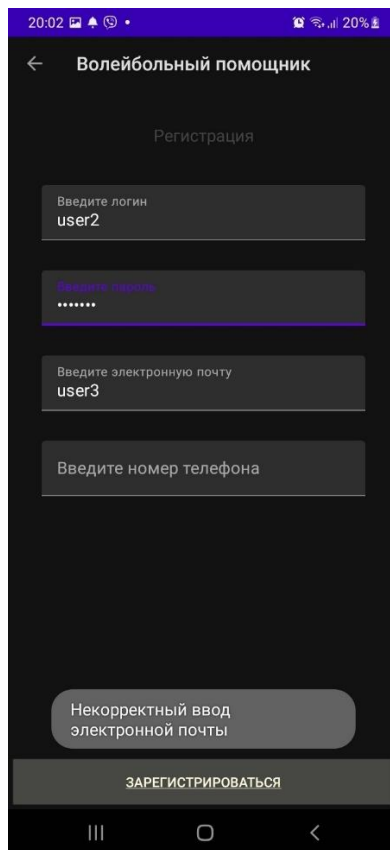


Рисунок 20 - ошибка ввода электронной почты

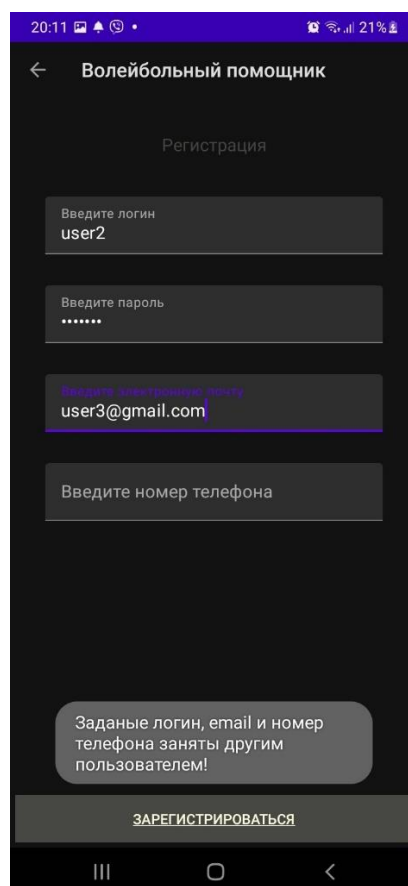


Рисунок 3 - ввод логина и/или email, которые уже заняты

					МИВУ.09.03.04-4.000 ПЗ	Лист
						68
Изм	Лист	№ докум.	Подп.	Дата		

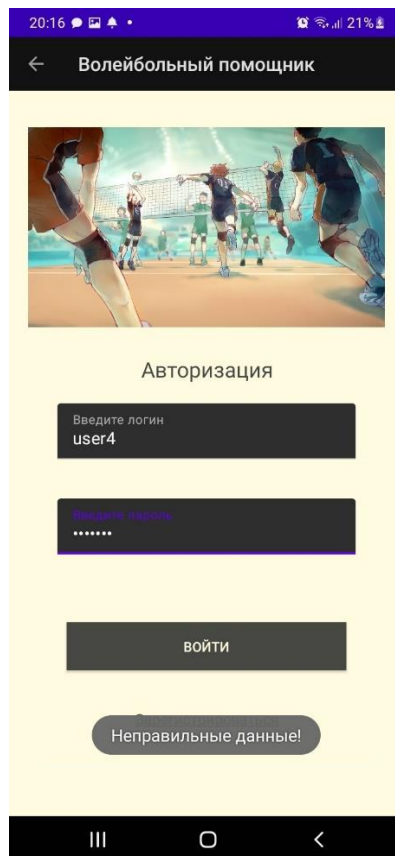


Рисунок 4 - ввод логина и пароля, которых нет в системе

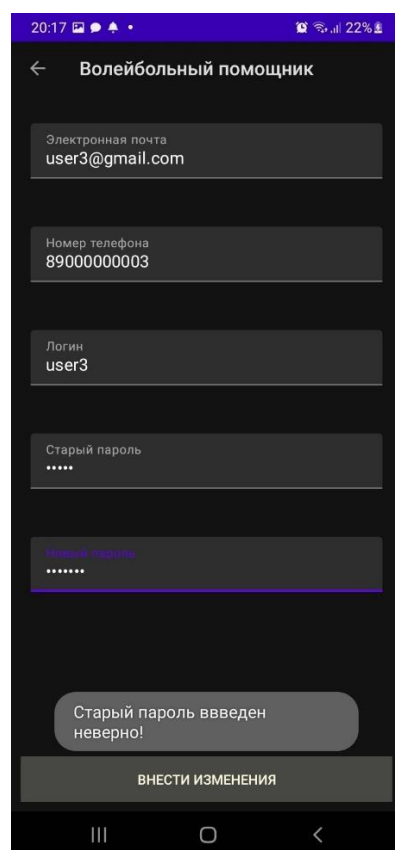


Рисунок 5 – ошибка ввода старого пароля

20:18 22%

← Волейбольный помощник

Название команды
ока

ДОБАВИТЬ КОМАНДУ

Адрес
спорткомплекс ока

№	ФИО	Поз.	ИЗМ	X
1	тулинова е.	И	ИЗМ	X
2	тулинова е.	И	ИЗМ	X
3	тулинова е.	И	ИЗМ	X
4	тулинова е.	И	ИЗМ	X
5	тулинова е.	И	ИЗМ	X
6	тулинова е.	И	ИЗМ	X

+

№ Игрок

Фамилия
Петрова

Имя

Ошибка ввода имени

ДОБАВИТЬ ИГРОКА

ТРЕНЕРСКИЙ СОСТАВ

Рисунок 6 – ошибка ввода имени

20:18 22%

← Волейбольный помощник

Название команды
ока

ДОБАВИТЬ КОМАНДУ

Адрес
спорткомплекс ока

№	ФИО	Поз.	ИЗМ	X
1	тулинова е.	И	ИЗМ	X
2	тулинова е.	И	ИЗМ	X
3	тулинова е.	И	ИЗМ	X
4	тулинова е.	И	ИЗМ	X
5	тулинова е.	И	ИЗМ	X
6	тулинова е.	И	ИЗМ	X

+

№ Игрок

Фамилия
Петрова

Имя
Гина

Отчество

Ошибка ввода отчества

ДОБАВИТЬ ИГРОКА

ТРЕНЕРСКИЙ СОСТАВ

Рисунок 7– ошибка ввода отчества

20:18 22%

← Волейбольный помощник

Название команды: ДОБАВИТЬ КОМАНДУ

Адрес:

№	ФИО	Поз.	ИЗМ	X
1	тулинова е.	И	ИЗМ	X
2	тулинова е.	И	ИЗМ	X
3	тулинова е.	И	ИЗМ	X
4	тулинова е.	И	ИЗМ	X
5	тулинова е.	И	ИЗМ	X
6	тулинова е.	И	ИЗМ	X

+

6 Игрок

Фамилия:

Имя:

Игровые номера индивидуальны!

ДОБАВИТЬ ИГРОКА

ТРЕНЕРСКИЙ СОСТАВ

Рисунок 8 – ошибка ввода игрового номера

20:22 22%

← Волейбольный помощник

Название команды: ДОБАВИТЬ КОМАНДУ

Адрес:

№	ФИО	Поз.	ИЗМ	X
1	Сербова К.Ю.	К	ИЗМ	X
2	Сербова А.Ю.	Л	ИЗМ	X
3	Зубанова Е.	И	ИЗМ	X
4	Тихонова Е.	И	ИЗМ	X
5	Фонина Н.	И	ИЗМ	X
6	Платина Н.	И	ИЗМ	X

+

№ 6 Капитан

Фамилия:

Имя:

Капитан команды уже назначен

ВНЕСТИ ИЗМЕНЕНИЯ

ТРЕНЕРСКИЙ СОСТАВ

Рисунок 9 – назначение 2 капитана

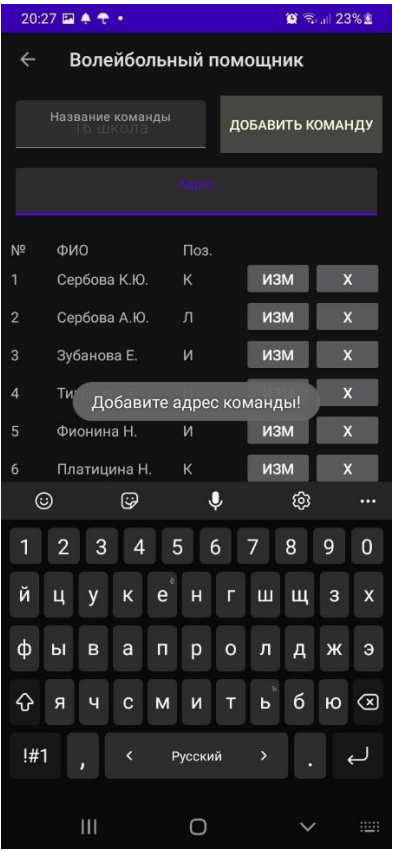


Рисунок 10 – ошибка ввода адреса команды



Рисунок 11 – ошибка добавления команды, количество игроков неправильное

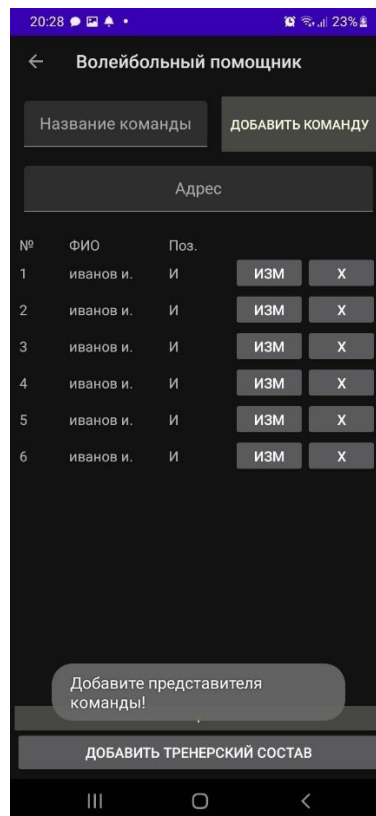


Рисунок 12 – ошибка добавления команды, не добавлен представитель

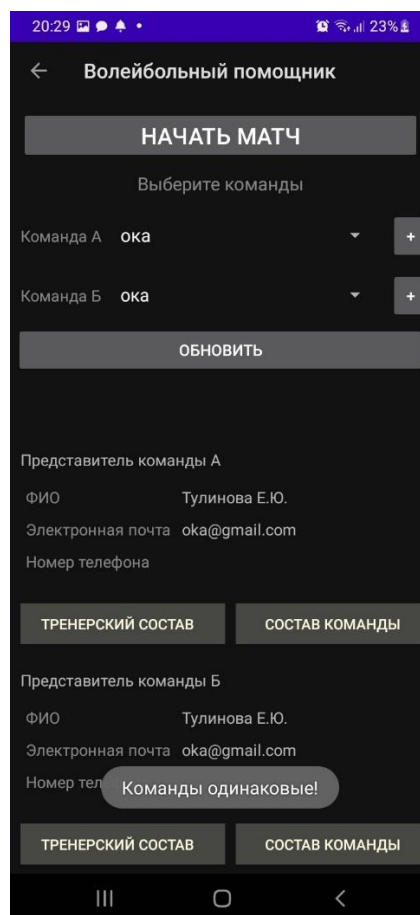


Рисунок 13 – ошибка создания матча

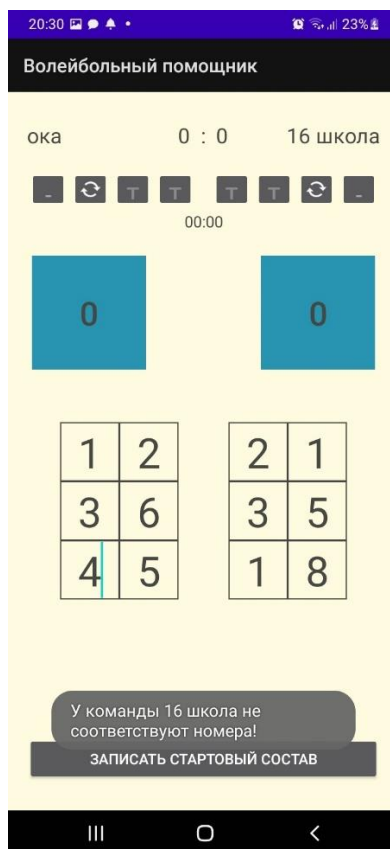


Рисунок 14 – ошибка занесения стартового состава

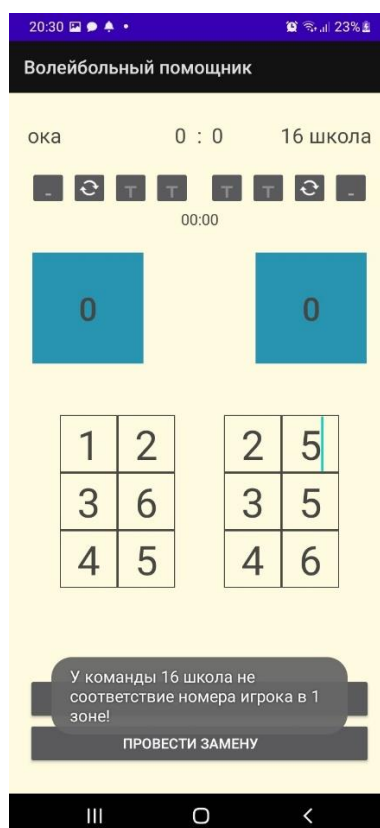


Рисунок 15 – ошибка проведения замены