Software Design Document

2024-Autumn-Aberdeen-Group9

Contents

1 Introduction ··· 3

1.1 Overall goals

1.2 Overall scope

2 Requirements Specification ··· 5

2.1 User requirements

2.2 Functional requirements

3 Overall design ··· 7

3.1 Activity

3.2 Layout

3.3 Data Storage

3.4 Resources

3.5 Build and Dependencies

4 User Interface Design··11

4.1 Color Design

4.1 Screen Flow

5 Key Technology···13

5.1 Programming language and development framework

5.2 Data Storage

6 Testing and User Experience Analysis ··15

6.1 Test

6.2 User Experience

7 Conclusion ·· 17

7.1 Summary

7.2 Challenges

7.3 Suggestions

1 Introduction

1.1 Overall goals

VolleyballOnline aims to create a comprehensive digital platform for volleyball enthusiasts, focusing on providing users with high-quality volleyball learning resources and event participation services. As volleyball gains more attention in China, especially among teenagers and amateur players, there is a lack of a dedicated platform to help these groups improve their skills and participate in volleyball activities. Therefore, the core goal of the application is to simplify the process of volleyball learning and competition through innovative technological means and user-friendly interface design, enabling users to easily get started and enjoy the sport.

We hope that through the VolleyballOnline platform, we can address the current pain points of volleyball enthusiasts in skill learning and event participation. For example, the lack of professional volleyball teaching resources, the absence of a convenient way to book matches, and the scattered and difficult-to-access information about activities. Our goal is to provide users with an integrated service platform that combines learning, training, and events, not only helping users improve their skills but also promoting the popularization of volleyball among amateur players.

1.2 Overall scope

The design of the "VolleyballOnline" application is centered around user needs and mainly covers the following functional modules:

1.2.1 User Registration and Login

The application provides simple user registration and login functions, users can register and log in to their own volleyball learning and participate in activities.

1.2.2 Professional teaching video

Teaching videos are one of the core contents of the "VolleyballOnline" platform. The platform has collected a large number of high-quality teaching resources through manual screening of different videos on the Internet and then released them in the application. The video content covers all aspects from basic skills to advanced techniques and practical strategies. Users can choose suitable learning modules according to their volleyball levels and gradually improve their technical skills. Through systematic course design, it helps users master the basic movements and skills of volleyball in a short time.

1.2.3 Competition booking service

On the "VolleyballOnline" platform, users can easily find and join the volleyball match that suits them. Through a simple booking process, users can check the time, location and number of participants of the contest and choose the right event according to their time and location.

1.2.4 Event information display and management

After registering for a competition or activity, users can visually view the registered competition in the personal center, including the detailed information of the competition, such as location, time, etc.

2 Requirements Specification

2.1 User requirements

The user group of "VolleyballOnline" application is mainly volleyball enthusiasts, including beginners, amateur players and users with certain volleyball foundation. For the different needs of these users, the application design goal is to ensure that users can achieve the following needs in a single platform:

2.1.1 Easy way to register and login

Users want to be able to register and log in quickly to ensure a smooth and easy experience, and registration information can be saved on the platform for subsequent use

2.2.2 High quality volleyball teaching resources

Users need to obtain systematic and professional volleyball learning materials, especially instructional videos for technical improvement. Beginners want a basic introductory course, while advanced players need a higher level of training.

2.2.3 Simple contest booking and participation

Users want to be able to easily view and participate in various volleyball activities, especially offline matches. Activity information needs to be clear and up to date so users can make choices based on their time and skill level.

2.2.4 Clear event reservation record

Users want to be able to view their scheduled events, especially event details, including time, location, and other game information.

2.2 Functional requirements

In order to meet the above user requirements, the "VolleyballOnline" application needs to implement the following functional modules:

2.2.1 User registration and login

- Users need to register through their mobile phone number, email address, username and password, and log in through their username and password.

2.2.2 Professional teaching video module

- The system needs to provide volleyball teaching videos from the beginning to the advanced level, the classification is clear, easy to find.
- Users can learn different volleyball skills according to the video content, such as serving, catching, spiking and so on.

2.2.3 Competition Reservation and Management Module

- Users can view all public volleyball events and sign up for them.

- Detailed information such as the time, location, and number of participants of the events is provided, allowing users to select the suitable ones.

- The events will be marked with the suitable participants, for users to choose by themselves.

- Once a user signs up, it will not be supported to cancel the registration. Because volleyball is a sport for 12 people, if a user cancels the registration near the start of the event, it will trouble the other 11 people.

3 Overall design

3.1 Activity

- **MainActivity:** The entry page of the application allows users to switch between different functional modules through the bottom navigation, such as the activity page (FragmentA), the instructional video page (FragmentB) and the welcome page (FragmentD). The system dynamically loads content for different pages based on the user name passed in.

- **RegisterActivity**: The user registration page, which allows the user to enter the user name, password, age, and contact information, and inserts the user data into the users table in the MySQL database through MySQLHelper.kt. After successful registration, the page will prompt the user to register successfully and return to the main page.

- **LoginActivity**: On the user login page, after the user enters the user name and password, it calls MySQLHelper.kt to check whether a matching user exists in the database. If the verification succeeds, go to the main page (MainActivity); Otherwise, an error message is displayed.

- **MyActivity**: User activity record page, through MySQLHelper.kt from the database to query the current user's participation in all activities, including activity name, time and place information. The page provides an intuitive view to display the user's activity records, and dynamically adjusts the display state of the active view (hidden or displayed).

- **FragmentA**: Events page, showing information about upcoming volleyball events (event name, location, time). Users can sign up for the event by clicking a button. MySQLHelper.kt provides functionality to check for empty Spaces in an activity, update the user's participation status, and write user data to the activity table.

- **FragmentB**: On the teaching video page, users can click the video icon to view different types of volleyball teaching videos (such as spiking, serving, etc.). By calling MySQLHelper.kt, get the URL of the video resource from the database and open it in the user's default browser.

- **FragmentD**: The Welcome page displays a personalized greeting after the user has logged in and provides a button to jump to the Activity record page (MyActivity). User information is loaded dynamically by passing parameters.

3.2 Layout

The layout is mainly a combination design of RelativeLayout, ConstraintLayout and

LinearLayout, combined with ScrollView and ViewPager2 to provide page scrolling and interaction functions. The overall style is simple and intuitive, meeting user operation requirements.

- Login page (activity_login.xml)

  The login page uses a RelativeLayout layout, with a central area containing input fields for the user name and password, and buttons for login and forward registration. The top is dominated by the text "login" title, and the whole page is set off by background pictures to provide users with a good login experience.

- Registration page (activity_register.xml)

  The registration page continues the style of the login page, using a similar layout, with the addition of an age and mobile number input field. The Confirm registration button is placed at the bottom. After completing the registration, return to the login page.

- Main page (activity_main.xml)

  RelativeLayout is used as the main page. The BottomNavigationView is used to implement page switching. The top FrameLayout dynamically loads different fragments. For example, the activity page (FragmentA), the teaching page (FragmentB), and the Welcome page (FragmentD).

- Activity page (fragment_a.xml)

  The activity page is one of the core functions of the whole application. The top of the page realizes image rotation through ViewPager2, and the activity information is displayed with ScrollView at the bottom. Each activity block contains basic information such as activity name, place and time, and displays the registration button, full button or signed up button according to the activity status.

- Teaching page (fragment_b.xml)

  The teaching page is designed in a vertical layout, and each teaching module (e.g. spike, serve, etc.) is presented in the form of a title and a horizontal scrolling list. Each video in the horizontal list displays a thumbnail image through the ImageView, which is clicked to get the video URL to play in the browser.

- Welcome page (fragment_d.xml)

  The welcome page is dominated by user personalized content, with the user's avatar displayed at the top and welcome text displayed in the center (e.g. "Hello, user name!"). Below provides a button to jump to the user activity record page, the overall layout is clear and concise.

- Activity log page (my_activity.xml)

  The activity history page displays the user's activity history with ScrollView, and the activity information is presented in a block layout with the activity name, time, and location. When the user has no activity record, the corresponding block is hidden. The back button at the top provides page navigation.

- Picture carousel (item_slider.xml)
  A single page layout for ViewPage 2 that contains only one ImageView for displaying images, providing intuitive visuals.

3.3 Data Storage

The application uses MySQL database for data storage and management. A helper class named MySQLHelper is created to manage the related operations of MySQL database, including creating databases and tables, as well as executing query, insert and update operations. The MySQLHelper class simplifies the development process by encapsulating database operations and ensures efficient interaction of data between the front-end and back-end. Its main functions include user data validation, activity data query and dynamic update of participation information.

3.4 Resources

3.4.1 Components

This application uses a variety of components from Material Design, including:

- Button (Button)
- ScrollView
- BottomNavigationView
- Input box (EditText)
- Picture slide control (ViewPager2)

These components ensure the intuitiveness of the application interface and the fluidity of the interaction.

3.4.2 Picture resource

The image resources in this application mainly include:

- Background image
- Button background for activities and teaching modules
- Icon resources for instructional videos and activities

Image resources are loaded through a unified resource management method, which optimizes the visual effect of the interface and improves the user's experience.

3.5 Build and Dependencies

3.5.1 Plugins

- **com.android.application plugin**
- **org.jetbrains.kotlin.android plugin**

3.5.2 Android Section

- **Namespace**: The namespace of the application com.example.volleyballonline。
- **compileSdk and targetSdk**: Both the compiled and target SDK versions are 33.
- **defaultConfig**: Contains default configuration information for the application, such as the application ID, minimum supported SDK version, version number, and version name.
- **buildTypes**: A build type named release is defined that contains obfuscation Settings and ProGuard rules.
- **compileOptions**: Set source code and object code compatibility to Java 1.8.
- **kotlinOptions**: Specify that the target version of the Kotlin JVM is 1.8.
- **buildFeatures**: The view binding function is enabled to simplify UI operations.
- **packagingOptions**: Defines packaging options for resources to avoid repeating file packaging errors.

3.5.3 Dependencies Section

- **AndroidX Libraries**: Including core-ktx, appcompat, constraintlayout, etc., to support basic functionality and modern Android development.
- **Material Design Libraries**: Using com. Google. Android. Material supplied material Design component library, and to optimize the Design of the user interface and interaction.
- **Network Libraries**: Integrate com.squareup.retrofit2 and com.squareup.okhttp3 for network requests and data processing.
- **Coroutine Support**: Introducing the org. Jetbrains. Kotlinx: kotlinx - coroutines - android, support asynchronous task processing.

4 User Interface Design

UI design starts with third-party platform tools to determine the number of pages, color palette, layout, and prototype. VolleyballOnline's page layout is simple. During the UI design phase, we used the third-party platform Inkknife to prototype 7 interfaces.

4.1 Color Design

4.1.1 Design concept

The app is dominated by bright colors, designed to convey energy, friendliness and joy, and is suitable for the target user group of the app. Through color matching, the appeal and functional usability of the application are enhanced, while ensuring the clarity and comfort of the interface.

4.1.2 The main color scheme

Primary Colors

- **Green (#A2EF4D)**

  For buttons, highlighted parts of the navigation bar, and other major interactive elements. Green represents energy and positivity, which fits in with the theme of the movement.

- **Purple (#800080)**

  It is used for secondary buttons and some auxiliary function areas to convey noble and modern sense and enhance the interface layer.

Secondary Colors

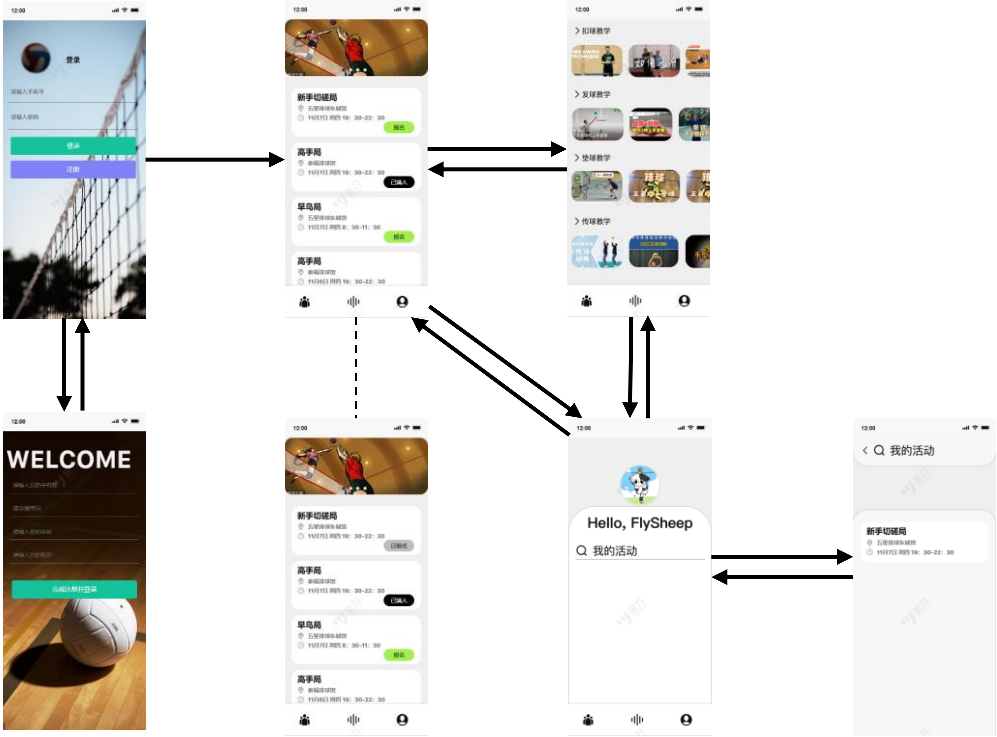
- **Light grey (#EFEFEF)**

  Used for background color to provide a soft visual experience without interfering with the display of the main content.

- **Dark grey (#4A4A4A)**

  For text content and prompt messages, to ensure that the text is easy to read and not harsh.

4.2 Screen Flow


登录
请输入手机号
请输入密码
登录
注册
WELCOME
完成注册并登录
新手切磋局
五星排球东城馆
11月7日 周四 19：30-22：30
报名
高手局
11月7日 周四 19：30-22：30
已满人
早鸟局
五星排球东城馆
11月7日 周四 8：30-11：30
报名
高手局
11月6日 周三 19：30-22：30
已报名
扣球教学
发球教学
垫球教学
传球教学
Hello, FlySheep
我的活动
我的活动
新手切磋局
五星排球东城馆
11月7日 周四 19：30-22：30

5 Key Technology

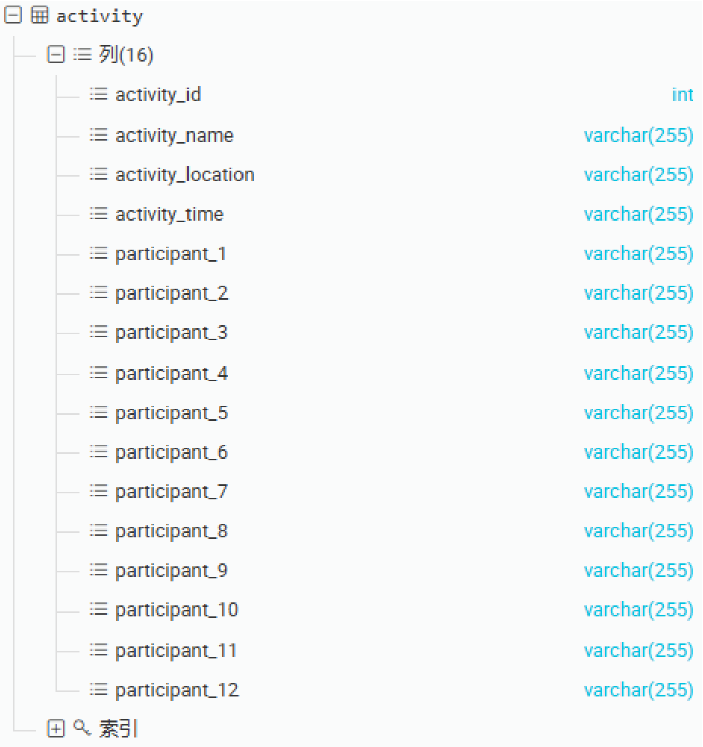
5.1 Programming language and development framework

- Kotlin: the primary application development language for more efficient Android application development with its strong type safety and concise syntax.
- Android Jetpack: Extensive use of Jetpack components, such as ViewModel and Navigation, to achieve modularity and high maintainability of applications.

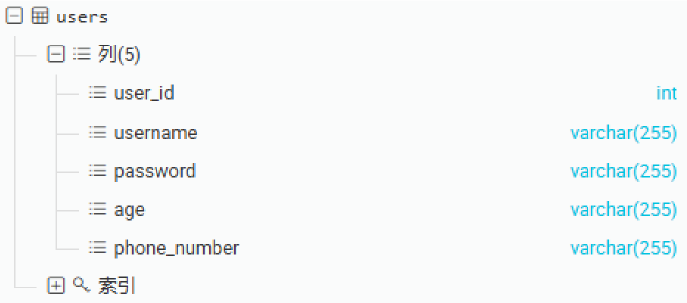
5.2 Data Storage

The application uses the MySQL database and manages database operations through a helper class called MySQLHelper, including creating databases, tables, and performing queries and updates.

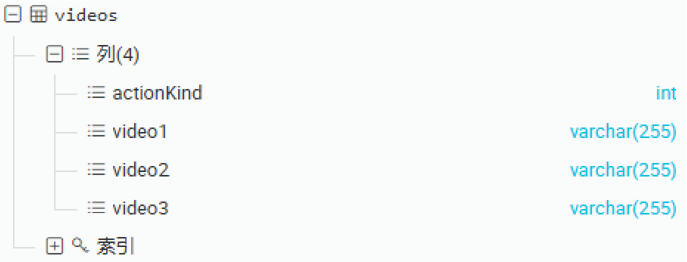
5.2.1 Database structure



```
activity
  列(16)
    activity_id                  int
    activity_name                varchar(255)
    activity_location            varchar(255)
    activity_time                varchar(255)
    participant_1                varchar(255)
    participant_2                varchar(255)
    participant_3                varchar(255)
    participant_4                varchar(255)
    participant_5                varchar(255)
    participant_6                varchar(255)
    participant_7                varchar(255)
    participant_8                varchar(255)
    participant_9                varchar(255)
    participant_10               varchar(255)
    participant_11               varchar(255)
    participant_12               varchar(255)
  索引
```



```
users
  列(5)
    user_id                      int
    username                     varchar(255)
    password                     varchar(255)
    age                          varchar(255)
    phone_number                 varchar(255)
  索引
```



videos

列(4)

actionKind	int
video1	varchar(255)
video2	varchar(255)
video3	varchar(255)

索引

5.2.2 Database Operation Methods

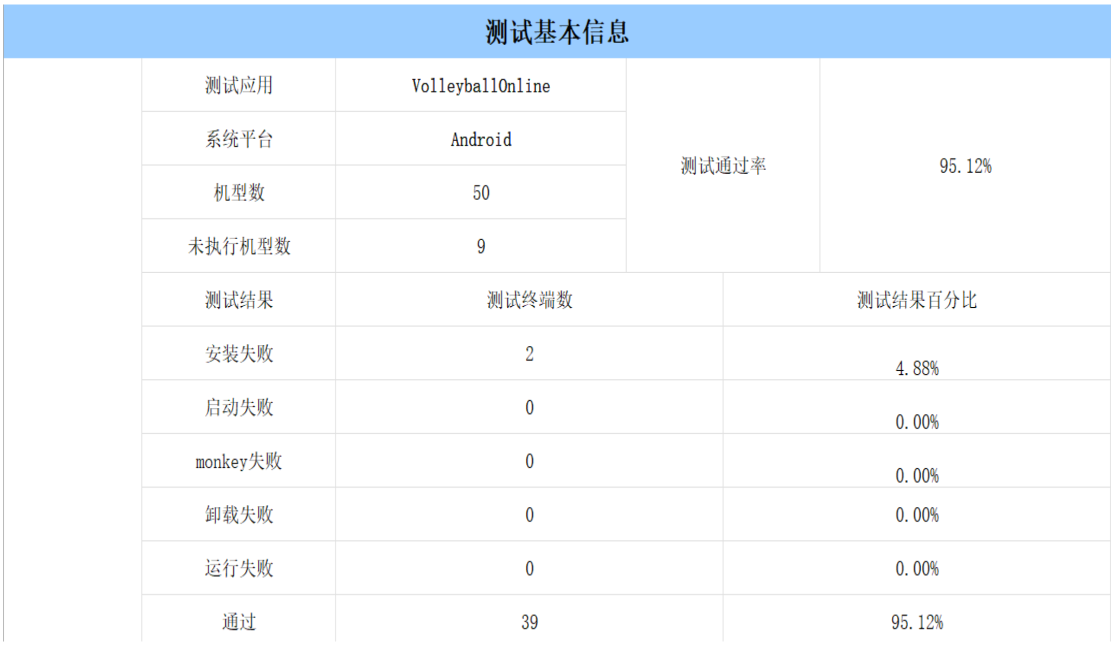
- User Management
  - **insertUser(username: String, password: String, age: String, phone_number: String):** Insert new user information into the users table.
  - **checkLogin(username: String, password: String): Boolean:** Verify user login information.
- Event Management
  - **findFirstEmptySeat(activityId: Int): String?:** Query the first empty participant location in an activity.
  - **insertUsername(theFirstEmptySet: String, username: String, activityId: Int): Boolean:** Insert the user name into the active Participant field.
  - **findIfInSet(username: String, activityId: Int): Boolean:** Checks whether the user has joined the specified activity.
  - **findActivityName(activityId: Int): String?:** Query the event name.
  - **findActivityLocation(activityId: Int): String?:** Find the event location.
  - **findActivityTime(activityId: Int): String?:** Query the event time.
- Teaching video management

- **getURL(actionKind: Int, videoMember: String): String?:** Gets a link to the instructional video for the specified class action.

6 Testing and User Experience Analysis

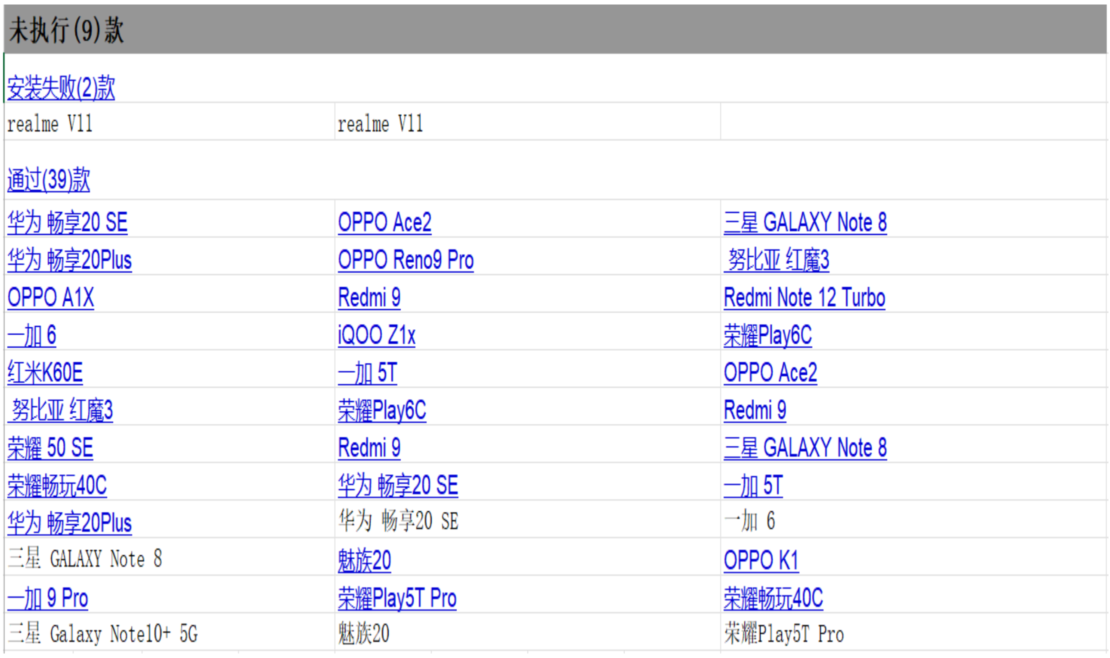
6.1 Test

The test results are as follows.



测试基本信息				
	测试应用	VolleyballOnline	测试通过率	95.12%
	系统平台	Android		
	机型数	50		
	未执行机型数	9		

	测试结果	测试终端数	测试结果百分比
	安装失败	2	4.88%
	启动失败	0	0.00%
	monkey失败	0	0.00%
	卸载失败	0	0.00%
	运行失败	0	0.00%
	通过	39	95.12%



未执行(9)款		
安装失败(2)款		
realme V11	realme V11	
通过(39)款		
华为 畅享20 SE	OPPO Ace2	三星 GALAXY Note 8
华为 畅享20Plus	OPPO Reno9 Pro	努比亚 红魔3
OPPO A1X	Redmi 9	Redmi Note 12 Turbo
一加 6	iQOO Z1x	荣耀Play6C
红米K60E	一加 5T	OPPO Ace2
努比亚 红魔3	荣耀Play6C	Redmi 9
荣耀 50 SE	Redmi 9	三星 GALAXY Note 8
荣耀畅玩40C	华为 畅享20 SE	一加 5T
华为 畅享20Plus	华为 畅享20 SE	一加 6
三星 GALAXY Note 8	魅族20	OPPO K1
一加 9 Pro	荣耀Play5T Pro	荣耀畅玩40C
三星 Galaxy Note10+ 5G	魅族20	荣耀Play5T Pro

6.2 User Experience

According to user testing feedback, most users are satisfied with our software. They think the software interface design is clean and simple. In terms of functionality, users say that our software covers all the basic functions and is very easy to use.

At the same time, users also put forward some suggestions for improvement. For example, they want to reduce the application footprint and optimize the display of information in the personal center. We took these suggestions and improved the application accordingly.

7 Conclusion

7.1 Summary

"VolleyballOnline" is an application designed for volleyball enthusiasts, aiming to provide convenient services such as match booking, professional teaching videos, and activity records. Through this platform, users can more easily participate in volleyball-related activities, improve their personal skills, and check their activity records and learning progress at any time.

We have implemented the main functions of the application, including match booking, teaching video modules, user registration and login, as well as activity record functions. Users have given positive feedback on the interface design and function implementation of the application, considering it intuitive and easy to operate.

7.2 Challenges

During the development process, we encountered various technical challenges:

In UI design, it was our first attempt at graphic design work. Creating an aesthetically pleasing interface was not easy, and most of the components in the Mockplus component library did not match our design style.

In front-end development, we faced these challenges: UI elements were arranged chaotically on different devices, fonts or images were distorted, data interaction between pages was disorganized, the navigation bar was complex to create, and when the button states on the event page (such as "Register" and "Full") were dynamically updated, the logic for some state transitions was rather complicated.

In back-end development, in the event management module, allocating a fixed number of participant fields for each event increased the complexity of the database structure. When obtaining data information from the back-end, the back-end code was messy, making it difficult to determine the source of problems when issues occurred.

7.3 Suggestions

To further improve our application, we plan to implement the following improvement measures:

User Experience Optimization

- Reduce the application's storage space usage to be compatible with more devices.

Function Expansion

- Provide diverse login methods, such as supporting social account login, to enhance login convenience.
- Add an "Advanced Course" recommendation function to the teaching module, dynamically pushing suitable content based on the user's learning progress.

Performance Optimization

- Improve the logic of button state switching to ensure smooth interaction even in complex activity scenarios.

We will continue to listen to user feedback, regularly maintain and expand the application's functions, and strive to provide users with a more comprehensive volleyball service experience.