南京信息工程大学

本科生毕业论文 (设计)

题　目　**基于 WebGL 的 AR 校园交互系统**

学生姓名　吴鼎

学　　号　201983290194

学　　院　软件学院

专　　业　软件工程

指导教师　余文斌

二〇二三 年 五 月 十 日

声　明

本人郑重声明：

1、 以“求实、创新”的科学精神从事科学研究工作。

2、 本论文中除引文外，所有测试、数据和相关材料均为真实有效的。

3、 本论文是我个人在指导教师的指导下进行的研究工作和取得的研究成果，请勿用于非法用途。

4、 本论文中除引文和致谢的内容外，并未抄袭其他人或其他机构已经发表或撰写过的研究成果。

5、 关于其他同志对本研究所做的贡献均已在论文中作了声明并表示了谢意。

作者签名：

日期：　　年　月　日

目 录

基于 WebGL 的 AR 校园交互系统

吴鼎

南京信息工程大学软件学院，江苏 南京 210044

摘要：随着移动设备 CPU 与 GPU 处理器性能提升以及以 Chrome 浏览器为代表的现代浏览器规范化、统一化发展。3D 图形技术以及以图像识别或标记捕捉为基础的 AR 技术已经能不再局限于 Unity3D 等固有框架。在移动设备浏览器上，以 WebGL 为基础的跨平台 3D 技术正蓬勃发展；以 AR.js 为基础，调用谷歌公司 ARCore 或苹果公司 ARKit 实现的 AR 技术逐步走向应用市场。本系统结合 WebGL 与 AR 技术实现一套以校园环境为背景的交互系统。核心功能上实现了 3D 模型的显示、操作以及对现实场景的识别以及与现实环境的交互。具体技术上，系统采用最新的技术框架: React 构建整体框架，Three.js 实现 WebGL 功能，AR.js 实现场景识别功能，Three.js 结合 AR.js 实现场景交互功能。本系统最大的优势为完全基于浏览器，不需要下载任何额外应用或插件，仅调用移动端底层服务，在跨平台与兼容性方面补足了传统 AR 技术依赖软件与操作系统的短板。同时本系统完全使用开源技术实现 AR 功能，不存在任何技术壁垒。

关键词：WebGL；AR；Three.js；图像识别；标记捕捉

AR Campus Interaction System Based on WebGL

Wu Ding

School of Software，NUIST，Nanjing 210044，China

Abstract： With the improvement of CPU and GPU processor performance of mobile devices and the standardization of modern browsers represented by the Chrome browser, unified development. 3D graphics and AR technologies based on image recognition or marker capture are no longer limited to inherent frameworks like Unity3D. WebGL-based cross-platform 3D technology on mobile device browsers; Based on AR.js, the AR technology implemented by calling Google's ARCore or Apple's ARKit is gradually moving towards the application market. This system combines WebGL and AR technology to realize an interactive system with the campus environment as the background. The core functions realize the display, operation, recognition of real scenes and interaction with real environments of 3D models. Specifically, the system adopts the latest technical framework: React builds the overall framework, Three.js implements WebGL functions, AR.js implements scene recognition functions, and Three.js combines AR.js to implement scene interaction functions. The biggest advantage of this system is that it is completely browser-based, does not need to download any additional applications or plug-ins, and only calls the underlying services of the mobile terminal, which makes up for the shortcomings of traditional AR technology relying on software and operating system in terms of cross-platform and compatibility. At the same time, this system completely uses open source technology to realize AR functions, and there are no technical barriers.

Key words： WebGL；AR；Three.js；image tracking；marker capture

1 绪论

1.1 研究背景和意义

AR，即增强现实技术，是虚拟影像和现实影像的融合。AR 技术最早的应用可以追述到 2009 年 2 月的 TED 大会，帕蒂 • 梅斯和普拉纳夫 • 米斯特莱展示了他们研发的 AR 系统 SixthSense[1]。该系统依靠常见的一些基本元件: 摄像头、小型投影仪、智能手机和镜子。首先摄像头和传感器采集真实场景的视频或者图像，传入后台的处理单元对其进行分析和重构，并结合头部跟踪设备的数据来分析虚拟场景和真实场景的相对位置，实现坐标系的对齐并进行虚拟场景的融合计算；交互设备采集外部控制信号，实现对虚实结合场景的交互操作。系统融合后的信息会实时地显示在显示器中，展现在人的视野中[2]。

AR 具有虚拟现实融合、实时交互、三维注册三大特征[3]。实现 AR 技术的关键是通过技术手段实现三大特征对应的功能:

- 现实融合: 需要一套高效的系统或算法完成对现实世界的识别。
- 三维注册: 需要一套三维显示技术，并且提供交互功能。
- 实时交互: 依赖于高性能的硬件设备。

除了高性能硬件无法通过编程手段实现，上述另外两大功能已有对应的解决方案，其中包括: 3D 图形显示技术，现实世界识别技术。目前比较主流的图形 API 包括 Direct3D, OpenGL, Vulkan, Metal[4]。而现实世界识别主要依赖于图像识别算法，标记捕捉算法或地理定位系统。目前最为主流的应用级 AR 实现方案有基于 Unity3D 的 ARFoundation 系统，该系统通过调用 ARCore SDK 调用底层 Android 设备的 AR 服务实现 AR 功能[5]。苹果系统则使用 ARKit SDK 调用 iOS 设备底层服务实现 AR 功能。

在 WebGL 领域，以 Three.js 为代表的 WebGL 技术也正出于蓬勃发展期，WebGL 是对 OpenGL 的封装，基于 WebGL 协议进行开发可以屏蔽底层复杂的图形逻辑而关注于浏览器端实现，实现 WebGL 协议的应用级框架有 Three.js。WebGL 通过使用 GPU 硬件加速，可以在 Web 浏览器中实现高性能的 3D 图形渲染。GPU 的并行处理能力使得 WebGL 能够同时渲染多个物体，实现更快的图形渲染。

当今社会，虚拟现实和增强现实技术被广泛应用于各种领域，例如游戏、教育、医疗和工业等。其中，增强现实技术为用户提供了一种融合虚拟和现实世界的体验，这使得用户可以在实际场景中与虚拟对象进行交互。在增强现实中，AR 交互系统是实现用户与虚拟对象交互的关键技术之一。而利用 WebGL 技术实现 AR 交互系统则具有以下研究意义：

- 与其他 AR 技术相比，WebGL 是一种基于 Web 标准的技术，可以在不需要任何插件或软件的情况下在 Web 浏览器中运行，极大地方便了用户的使用。
- 由于 WebGL 技术采用 GPU 进行图形渲染，可以实现高质量的图形渲染和流畅的动画效果，这对于增强现实中需要呈现逼真虚拟对象的应用场景非常重要。
- 在 AR 交互系统中，用户与虚拟对象的交互往往需要实时响应，而 WebGL 技术的高效性可以保证系统的实时性能，从而提升用户体验。
- 通过使用 WebGL 技术，可以实现基于 Web 的跨平台应用，使得用户可以在不同的设备上使用相同的 AR 交互系统，极大地扩展了系统的应用范围。

1.2 研究现状

目前以 ARFoundation 为代表的已有的成熟 AR 实现方案依赖于其背后公司独有的闭源技术，或者需要昂贵的专业 AR 设备。随着浏览器的高速发展以及智能手机的快速迭代，仅通过智能手机浏览器调用操作系统底层 AR 服务的技术方案成为可能[6]。且浏览器服务拥有跨平台，兼容性强，生态丰富，无需额外设备等优点。

目前已有许多研究者或工程师尝试使用 WebGL 技术实现 AR 功能，主流技术方案如下:

- WebGL 框架：WebGL 技术可以实现高质量的图形渲染和流畅的动画效果，可以实现逼真的虚拟物体呈现。例如，可以使用 Three.js 框架实现虚拟人物、虚拟建筑等的呈现。
- AR 引擎：目前已经有许多 AR 引擎采用 WebGL 技术实现，如 AR.js、Three.js、Awe.js、jsartoolkit5 等。这些引擎通过 WebGL 技术实现虚拟物体的呈现，同时提供了相应的 API 和工具库，方便开发者进行 AR 应用程序开发。
- React 生态：目前已有许多开发人员将 Three.js, AR.js 等主流技术方案整合到 React 生态中，诞生了例如 @react-three 等优秀的开源框架，且社区配备了良好的开发文档。

1.3 研究目标与内容

本系统以校园环境为依托，采用 WebGL 技术实现 3D 图形显示功能、AR.js 技术实现现实世界识别功能。系统完全使用已有的开源技术，不存在任何技术壁垒。且仅需将系统部署在服务器上即可通过浏览器实现所有功能。本系统是以 WebGL 技术为基础的 AR 系统的一次尝试与实现。

由于完全采用浏览器技术，本系统具有良好的迁移性。可以嵌入到微信，支付宝等小程序中；或者封装成移动端 app；亦或是嵌入专业 AR 设备。

本系统致力于开发一套三维交互系统，用户仅需一台移动端设备，通过摄像头即可完成对现实世界的识别。系统将自动识别采集到的图像信息并作出相应的响应: 显示三维动效或相关场景信息。用户也可以在设备上将虚拟物体放置到现实场景中预览效果。同时本系统也提供了基本的用户身份管理与社区功能，用户进入系统后将自动鉴定权限并提供对应的功能。

本系统研究内容如下:

1. 了解 WebGL 技术和以 Chrome 为代表的现代浏览器对 WebGL 技术的迭代发展情况。着重了解 WebGL 技术的现状，实现方案，未来发展方向。深入学习 3D 模型显示技术。
2. 学习主流的 WebGL 实现框架: Three.js，前端框架: React.js, 前端 UI 框架: MaterialUI。掌握 React 生态主流的 Three.js 实现方案: @React-Three。
3. 学习使用 AR.js，将其整合进 React 框架，理解其背后的图像识别，标记捕捉算法。根据实际使用情况尝试对系统进行优化。
4. 基于 WebGL 与 AR.js 技术完成系统核心功能的设计，实现，优化，提供简洁美观的 UI 界面，人性化的用户体验。

本系统核心功能采用 AR.js 调用终端设备 AR 服务。具体实现上采用 React、MaterialUI、Three.js、AR.js 框架并对其进行封装。系统技术栈如图1.1所示:

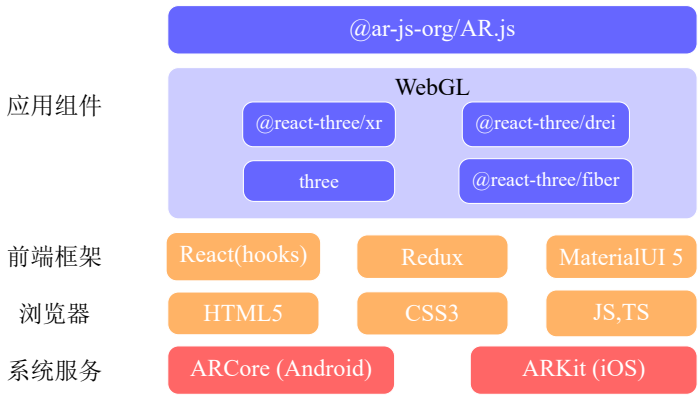

图 1.1 系统核心功能技术栈

1.4 论文组织结构

论文包括以下七个部分:

1. 绪论。该部分介绍了 AR 的发展、基础功能实现原理、软硬件生态。通过查阅文献研究了 WebGL 实现 AR 功能的主流方案，阐述了系统的实现目标与主要研究内容。
2. 技术与方法。简要阐述系统使用的技术方案，主要介绍了 WebGL 技术与 AR 技术以及具体采用的应用组件，并介绍相关组件生态与开发工具。
3. 系统分析。阐述系统可行性，主要梳理了目前采用的技术方案 (three.js 与 AR.js) 的可行性与优缺点。同时阐述需求分析，明确系统需要实现的功能。
4. 系统设计。对系统整体功能和关键业务梳理分析，对核心功能的实现与优化方案进行阐述。
5. 系统实现。介绍系统完成后各个界面的功能。
6. 系统测试。分别进行了单元测试，集成测试，性能测试。着重对 AR.js 的图像识别与标记捕捉功能进行测试。
7. 总结。对系统的开发进行总结，阐明了对系统的一些展望，并介绍了个人开发过程中的收获。

2 技术与方法

2.1 基础技术方案

系统采用 B/S 架构，即 “浏览器/服务器” 架构，核心功能在浏览器端，服务器端仅提供必要的数据。

前端采用如下技术方案:

- TypeScript: 为 JavaScript 提供类型扩展，有助于系统的维护与稳定开发。系统采用 ES6(ECMAScript) 语法编写。
- React: 前端应用整体框架，采用 React18+ 的 hooks 写法，所有组件均使用 JSX 语法编写。对 AR.js 等必要功能进行了组件化封装。
- Redux: 状态管理框架，统一管理公共数据，通过发布订阅模式控制各个 React 组件状态。

• Material UI: 一套动画丰富，适合响应式应用的 UI 库，设计上遵循 Google 公司提出的 Material Design。

• fetch: ES6 新增的 API，用于访问和操纵 HTTP 请求。替代传统以 XmlHttpRequest 为基础的 ajax 请求。

前端的整体架构如图2.1所示:

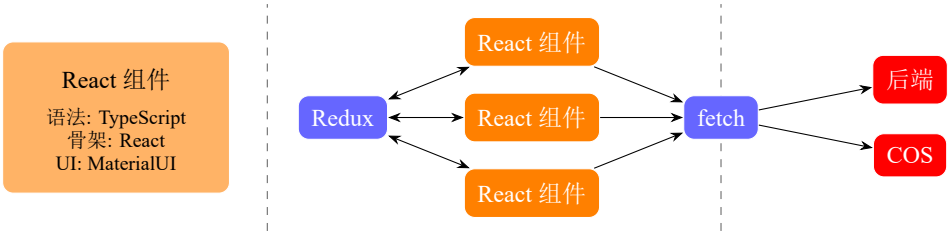

图 2.1 前端整体架构

后端采用如下技术方案:

• SpringBoot: Java 后端主流企业级框架。整合了 Spring 框架的快速开发工具，Spring 生态丰富的组件均可以在 SpringBoot 框架内快速应用。

• Spring Data Jpa: SpringBoot 主流的全 ORM 数据库操作框架。提供了简单易用的 API，可以让开发者更加方便地使用 JPA 实现对数据库的操作[7]。

• MySQL: 主流的企业级开源免费关系型数据库，适合存储常规数据段。

• COS(对象存储): 对象存储服务将数据作为对象进行管理。本系统将图像识别，标记捕捉的数据以及模型数据放入对象存储桶中，前端通过 fetch 调用对应数据。

2.2 核心技术方案

系统核心技术为 WebGL 与 AR，对应使用的技术方案为 three.js 与 AR.js。

2.2.1 WebGL

WebGL（Web Graphics Library）是一种用于在 Web 浏览器中呈现交互式 3D 图形的技术。它是 JavaScript API（应用程序编程接口）的一种实现，可以与 HTML5 一起使用，让开发者能够在网页中使用 3D 图形，而无需使用插件或其他额外的软件[8]。WebGL 是基于 OpenGL ES（OpenGL for Embedded Systems）的 API，因此它提供了与 OpenGL 相似的功能和语法。它允许开发者使用 JavaScript 代码来创建复杂的 3D 场景，包括模型、纹理、光照、阴影和动画等效果[9]。

WebGL 与 OpenGL 都是一种技术协议，在具体应用上，系统采用了 three.js 框架。three.js 是一个基于 WebGL 的 JavaScript 3D 库，提供了一组易于使用的 API，可以处理 3D 对象、材质、光照和阴影等，还有许多其他功能，如动画、粒子效果、3D 文字、后期处理等[10]。同时它也提供了部分 AR 功能，具体来说，three.js 中有一个 AR 功能库，它是基于 three.js 的一个开源项目，专门用于在网页上构建 AR 应用程序。然而 three.js 内置的 AR 功能有限，无法准确地通过摄像头数据进行场景识别。

由于本系统采用 React 作为主要框架，因此使用 @react-three 项目下的 @react-three/fiber、@react-three/drei、@react-three/xr 组件，这些组件进一步封装了 three.js，允许调用者在 React

框架内使用 JSX 语法实现 three.js 项目。这三个主要组件功能如下:

- @react-three/fiber: 核心组件，将 three.js 封装为 JSX 组件。
- @react-three/drei: 提供一些默认预设样式。
- @react-three/xr: 实现 three.js 自带的 AR 功能。

WebGL 相关技术的关系如图所示:

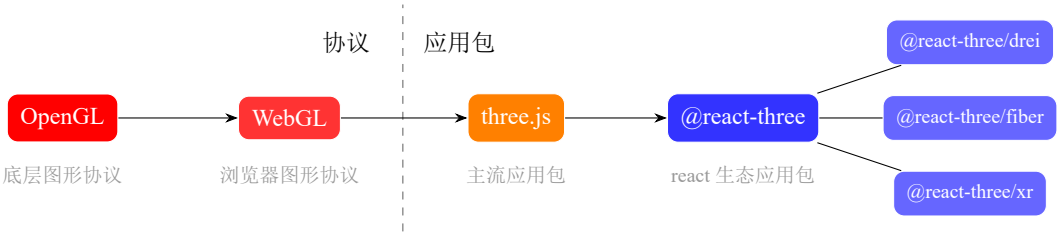

图 2.2 OpenGL, WebGL, three.js, @react-three 关系

2.2.2 AR

虽然 three.js 本身内置一定的 AR 功能，但无法准确高效识别现实信息[11]。因此本系统引入了 AR.js 包。AR.js 基于 WebRTC 和 WebGL 技术，并使用了两种主要的跟踪方式：基于图像的跟踪和基于标记的跟踪。

WebRTC（Web Real-Time Communications）是一种开放的网络技术标准，它允许在网页浏览器之间进行实时音视频通信和数据传输。WebRTC 的目标是让网页应用程序能够通过简单的 JavaScript API 实现高质量的实时通信，而不需要安装任何插件或额外的软件[12]。

图像追踪是一种基于计算机视觉的 AR 跟踪技术，它使用摄像头捕捉到的图像来寻找预先定义好的目标图像，然后根据目标图像的位置和姿态来在实时视频中渲染虚拟对象[13]。AR.js 中的图像追踪技术可以用于识别和追踪平面上的图像。

标记追踪是一种基于二维码或其他特定图案的 AR 跟踪技术，它可以通过扫描预先定义好的二维码或标记图案来确定虚拟对象的位置和姿态。在 AR.js 中，我们可以使用 ArMarker-Controls 来进行标记追踪，该控件使用了 jsartoolkit5 库来实现二维码的识别和跟踪。AR.js 支持的标记类型包括 Hiro、Kanji、Barcode 和 Custom Marker 等，开发者可以根据需求选择不同的标记类型[14]。同时，AR.js 还支持在标记上显示虚拟对象和交互元素，以实现更加丰富和交互性的 AR 内容。

2.3 开发工具

本系统使用的主要开发 IDE(集成开发系统) 为 Visual Studio Code，使用的设备为本人的手机，对应的开发环境与设备参数如下:

- 操作系统: Window11 22H2
- IDE: Visual Studio Code 1.77.3
- 设备: Snapdragon 885(处理器), Android 11
- 浏览器: Chrome 112.0(移动版)

3 系统分析

3.1 可行性分析

3.1.1 技术可行性分析

本系统以 React 为整体框架，整合 three.js、AR.js 实现基于 WebGL 的 AR 交互功能。采用 fetch 技术获取后端数据以及图像识别和标记捕捉数据。整体上采用了信息系统的开发方法、网络和通信技术、数据库技术等，并且基于 B/S 结构规划技术和设计技术。本系统采用的绝大部分包均获得了应用级市场的认可，对于尚未广泛使用的 AR.js 包，本系统将其集成进 react 框架，用于实现场景识别功能。

3.1.2 经济可行性分析

本系统完全采用开源框架与技术，系统应用的支出费用主要包括系统开发费用和系统运行费用。

其中系统开发费用有：人工费用、咨询评审费用、设备费用、调研差旅费、开发软件费用以及不可预见的费用。系统运行费用主要包括：系统维护费、设备维护费、消耗材料费。

本系统带来的经济效益由以下几个方面来衡量:

1. 提供预览服务: 虚拟物品与现实结合，用户对物品有更直观的视觉感受。
2. 降低试错成本: AR 可以在虚拟环境中进行产品原型测试，减少了在实际生产环境中出现的错误和问题，节省了成本。
3. 提高物品销售额：以为产品展示提供更加直观、生动的效果，从而吸引更多消费者购买产品。

社会效益有:

1. 提供更好的用户体验：AR 技术为校园用户提供更加直观、生动、互动的体验，从而提高用户的满意度。
2. 促进文化创意产业的发展：交互系统可以为文化创意产业提供新的展示和推广方式，例如可以将文物、艺术品等通过 AR 技术呈现给观众，从而提高文化创意产业的影响力和发展潜力。
3. 提高教学水平：交互系统可以为学生提供更加直观、生动的学习体验。

3.1.3 社会可行性分析

基于 WebGL 的 AR 校园交互系统具有以下社会可行性:

1. 促进大学校园文化建设：该系统可以通过 AR 技术展示大学校园的历史、文化、建筑、艺术品等方面的信息，为大学校园文化建设提供更加直观、生动的展示方式。
2. 提高学生的学习体验：该系统可以为学生提供更加直观、生动的学习体验，例如可以将科学知识、历史文化等通过 AR 技术呈现给学生，从而提高学生的学习兴趣和学习效果。
3. 提高大学校园的安全管理：该系统可以通过 AR 技术为学生提供安全教育，例如为学生演示火灾、地震等紧急情况的应对方法，提高学生的安全意识和应对能力，从而提高大学校园的安全管理水平。

4. 推动 AR 技术的应用和发展: 该系统有助于激发学生对 AR 技术以及相关图像识别，三维图像显示技术的兴趣。

3.2 需求分析

软件工程实践正变得越来越重要，以减轻任何可能导致代价高昂的停机时间、不正确的行为或安全故障的失败风险。需求提取是系统地从人类利益相关者、系统环境、可行性研究、市场分析、商业计划、竞争产品分析和领域知识的组合中提取和识别系统需求的过程[15]。

系统整体用例图如图3.1所示:

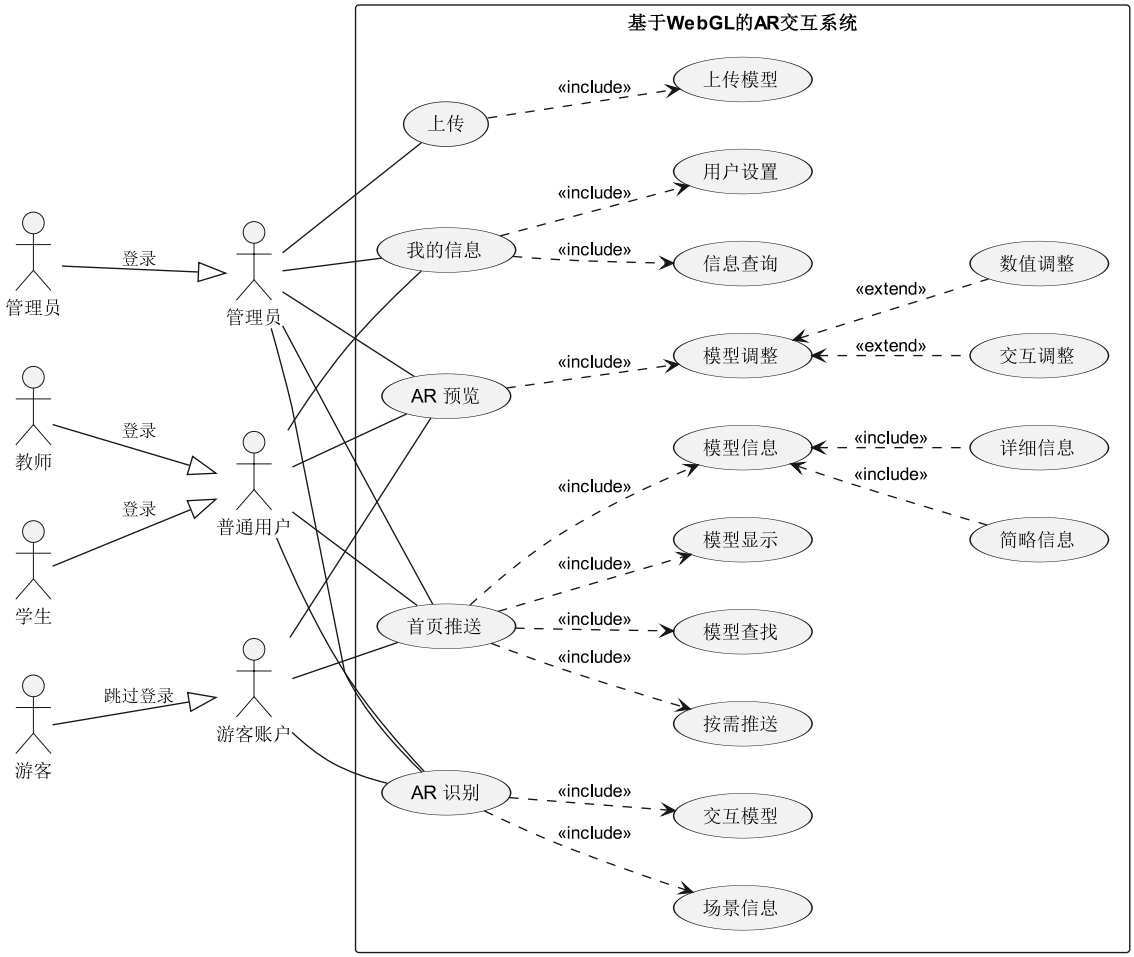

图 3.1 系统用例图

3.2.1 功能性需求分析

登陆注册: 用户 id 必须与学号或工作编号一致，在注册时匹配校园编号表。用户身份分为四类: 游客，学生，老师，管理员。游客账户为前端预留账户，用户可以通过该账户体验系统功能，但无法与后端进行交互。不同的身份将获得系统的不同权限以及推送内容。

用户的登录步骤如表3.1所示:

表 3.1 用户登录用例表

用例名	用户登录	
角色	学生，职工，管理员	
准入条件	用户拥有进入系统的权限	
事件流	**角色步骤**	**系统步骤**
	1. 用户进入登录界面	
	2. 用户输入用户名/邮箱和密码	**3:** 前端进行数据检查。
		4: 后端查询数据库进行匹配。
		5: 用户进入系统，跳转到 “首页” 界面
退出条件	系统验证用户名密码正确	
例外	**1.** 步骤 3，前端检查数据错误，给出提示信息。 **2.** 步骤 4，后端匹配数据失败，用户重新登录。	

新用户注册时需选择对应的身份，例如学生，教师等。系统将通过校园用户表自动判断用户与选择的权限组是否对应，用户注册步骤如表3.2所示:

表 3.2 用户注册用例表

用例名	用户注册	
角色	学生，职工，管理员	
准入条件	用户拥有进入系统的权限	
事件流	**角色步骤**	**系统步骤**
	1. 用户进入注册界面。	
	2. 用户填写并提交注册信息。	**3:** 前端进行数据检查。
		4: 后端根据校园系统表判断能否注册，并返回信息。
		5: 前端显示注册结果，跳转到 “我的” 界面。
退出条件	用户成功注册	
例外	**1.** 步骤 3，前端检查数据错误，给出提示信息。 **2.** 步骤 4，后端未匹配到对应编号，或编号与选定的权限组不符。	

在用户进入系统后，可在个人资料界面修改个人信息，修改用户信息步骤如表3.3所示:

表 3.3 用户增改个人信息用例表

用例名	用户增改个人信息	
角色	学生，职工，管理员	
准入条件	用户拥有个人账号	
事件流	**角色步骤**	**系统步骤**
	1. 用户进入我的界面，针对个人信息进行增改。	
	2. 用户提交修改后的个人信息。	**3:** 前端进行数据检查。
		4: 后端增改数据
		5: 前端显示增改后的结果
退出条件	成功修改信息	
例外	**1.** 步骤 3，前端检查数据错误，给出提示信息。 **2.** 步骤 4，后端增改数据失败，给出错误信息。	

首页推送: 系统首页划分为 “为你推荐”、“最新更新”、“我的关注” 三个分区，分别获取推荐内容，最新内容以及关注内容。用户选择不同分区时，系统将自动调用后端算法推送内容。首页推送功能执行步骤如表3.4所示:

表 3.4 首页推送用例表

用例名	首页推送	
角色	游客，学生，职工，管理员	
准入条件	用户能够进入系统	
事件流	**角色步骤**	**系统步骤**
	1. 用户进入首页	
	2. 用户选择 “为你推荐”，“最新更新”，“我的关注” 任一选项	**3:** 前端发送对应请求
		4: 后端根据请求类型与权限组查找并返回不同数据。
		5: 前端根据接受的数据类型显示模型卡。
退出条件	用户切换筛选类型	
例外	**1.** 步骤 4，后端查询数据失败。	

用户在首页也可自行查询相关内容，搜索栏支持按模型名称查询功能，系统后端返回结果后，前端会替换原先内容显示搜索的结果，首页查询功能的执行步骤如表3.5所示:

表 3.5 首页查询用例表

用例名	首页查询	
角色	游客，学生，职工，管理员	
准入条件	用户能够进入系统	
事件流	**角色步骤**	**系统步骤**
	1. 用户进入首页	
	2. 用户进行关键字搜索	**3:** 前端检查搜索内容并发送对应请求
		4: 后端根据请求与权限组查找数据并返回。
		5: 前端根据接受的数据类型显示模型卡。
退出条件	用户点击 “搜索” 按钮	
例外	**1.** 步骤 3，搜索内容出错。 **2.** 步骤 4，后端查询数据失败。	

3D 模型展示: 系统建立 3D 模型并展示环境，获取模型数据并在手机内直接展示，且提供模型背景，预设环境等可操作选项供用户预览。同时提供一定的模型文本信息，用户可以在简略信息与详细信息模式直接切换，也可完全屏蔽模型信息。模型展示功能的执行步骤如表3.6所示:

表 3.6 模型展示用例表

用例名	模型展示	
角色	游客，学生，职工，管理员	
准入条件	用户能够进入系统	
事件流	**角色步骤**	**系统步骤**
	1. 用户进入首页	
	2. 用户点击对应模型卡。	**3:** 前端向后端请求模型详细信息数据。
		4: 后端返回模型数据。
		5: 前端加载界面与模型。
	6: 用户点击交互按钮。	**7:** 前端做出对应响应。
	8: 用户退出 “模型展示” 界面	
退出条件	用户退出 “模型展示” 界面	
例外	**1.** 步骤 5，网络等问题加载模型或信息失败。	

3D 模型 AR 预览: 系统调用设备摄像头将模型放入到 AR 环境中，允许用户在现实环境中对模型进行移动缩放等操作，提供 3D 模型预览服务。AR 预览功能的执行步骤如表3.7所示:

表 3.7 AR 预览用例表

用例名	AR 预览	
角色	游客，学生，职工，管理员	
准入条件	用户能够进入系统	
事件流	**角色步骤**	**系统步骤**
	1. 用户进入 “首页” 界面。	
	2. 用户点击模型卡并选择 AR 预览功能。	**3:** 跳转至 “预览” 界面。
		4: 系统请求设备开启摄像头与 AR 服务。
		5: 向后端请求模型数据。
		6: 前端加载模型，展示 AR 预览界面。
	7: 用户操作 AR 预览界面	**8:** 前端做出对应响应。
	9: 用户退出 “AR 模型展示” 界面。	
退出条件	用户退出 “AR 模型展示” 界面。	
例外	**1.** 步骤 4，系统请求摄像头权限或 AR 服务被拒。 **2.** 步骤 6，网络等问题加载模型或信息失败。	

AR 场景识别: 根据摄像头捕捉的图像信息，采用标记捕捉或图像识别算法，在捕捉到对应信息后显示相关三维模型以及对应文本信息。AR 识别功能的执行步骤如表3.8所示:

表 3.8 AR 识别用例表

用例名	AR 识别	
角色	游客，学生，职工，管理员	
准入条件	用户能够进入系统	
事件流	**角色步骤**	**系统步骤**
	1. 用户进入 “识别” 界面	**2:** 系统请求设备开启摄像头与 AR 服务
		3: 系统根据地理位置按需加载识别信息。
		4: 系统通过图像识别或标记捕捉算法实时匹配摄像头传输的场景信息。
		5: 捕捉到有效信息。显示模型或相关提示。
	6: 用户与系统交互	**7:** 系统做出响应。
	8: 用户退出 “AR 识别’ 界面	
退出条件	用户退出 “AR 识别” 界面	
例外	**1.** 步骤 2，系统请求摄像头权限或 AR 服务被拒。 **2.** 步骤 3，网络等问题加载模型或信息失败。	

3.2.2 非功能性需求分析

系统非功能性需求分析确保系统在满足功能需求的同时，也满足了非功能性需求，如性能、可靠性、安全性、可维护性等。确保满足非功能性需求可以有效保证软件稳定性、扩展性[16]。

1. 性能

性能指标可分为时间性能和空间性能。

• 时间性能: 本系统操作的持续时间不超过 2 秒，数据连接操作的持续时间不超过 5 秒。系统利用浏览器异步加载机制加载需要的资源，采用非阻塞机制逐步对场景，模型加载。如果由于网络问题，数据无法获取，系统应在等待至多 10 分钟后给出提示并恢复。

• 空间性能: 本系统按需推送内容，在界面中每次仅推送有限内容，根据用户操作增加推送内容。在 AR 场景识别过程中，根据用户地理位置推送必要数据。

2. 可靠性

本系统测试时间不少于 20 小时。在测试过程中，系统应尽可能经历所有可能出现的故障情况，以确保系统的成熟度。当硬件或软件出现异常时，系统仍应具有服务能力。业务容量减少，但不会导致系统崩溃。该系统应能在停机后十分钟内修复并正常运行。

3. 可用性

本系统在接收到用户输入的错误数据后，系统不会崩溃，系统能够识别错误并给用户相应的提示信息。系统界面美观，对用户具有吸引力。本系统的界面设计应在保证基本功能的基础上实现友好的布局，与市面上多数移动设备 app 界面操作保持一致。

4. 易用性

本系统易于学习。系统应确保操作逻辑符合常规 app 操作逻辑，新用户在第一次使用时，在 3 分钟的学习时间内了解整个系统的基本功能，在 10 分钟的学习时间内掌握系统的所有功能。AR 交互系统的用户界面应该简单、直观、易于使用。系统采用合适的配色、布局和图标来吸引用户的注意力，并使用户能够快速、准确地完成任务。

5. 安全性

AR 交互系统具有适当的安全措施，用户的隐私和数据安全。系统应该使用 HTTPS 协议来保护用户的数据传输。前端仅保留必要数据。后端对重要数据进行加密保存。

6. 可维护性

系统采用模块化设计。将系统拆分为多个独立的模块，每个模块处理特定的任务或功能。有助于提高系统的可维护性，开发人员可以更容易地定位和修复问题，并且在更新或升级时可以更轻松地进行更改。在系统设计和开发过程中，采用一致的编程规范和标准，确保代码的可读性和可维护性。使用版本控制工具（Git）提高系统可维护性。

7. 可扩展性

系统中的各个模块应该尽可能地松耦合，以避免修改一个模块对其他模块产生意外影响。这有助于确保系统的可拓展性和稳定性。在设计和开发过程中，使用广泛采用的标准和技术，以便将来可以轻松地添加新功能或集成其他系统。

4 系统设计

系统设计是指在软件开发过程中，对整个系统进行设计和规划，包括对系统的架构、组件、模块、数据库、接口、协议、算法等各方面的设计。系统设计是将用户需求转化为可实现的系统的过程。本文将从系统总体设计开始，继而阐述基础功能设计，着重阐述核心功能设计。

4.1 系统总体设计

4.1.1 系统架构设计

系统基于 B/S 架构开发，核心功能位于浏览器端，后端仅提供必要的数据服务。用户通过浏览器端发起数据请求，服务器端负责处理请求后返回必要数据，浏览器再通过 WebGL 与 AR 技术展示结果[17]。

本系统的整体设计架构如图4.1所示:

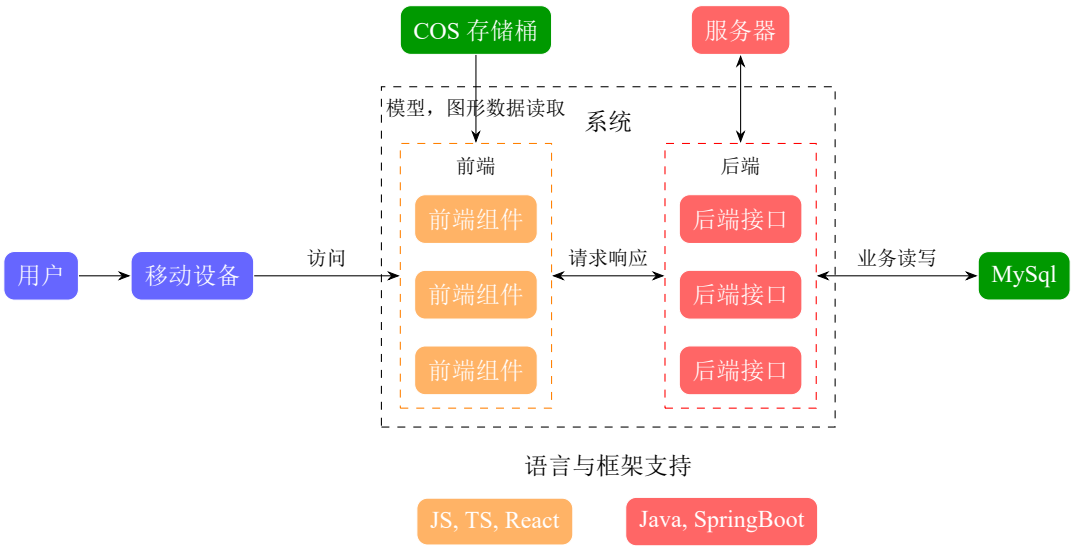

图 4.1 系统架构设计图

4.1.2 系统技术选择

在技术选择的过程中，需要考虑技术适用性，开发难度，成本，社区生态，安全性，可维护性。综合考量，本系统使用如下主流技术。

前端采用 React 框架，该框架能带来诸多优势。例如组件化开发，可以结合 AR 技术和 WebGL 技术封装可复用的组件，便于开发和维护。这样可以提高开发效率，减少代码重复。虚拟 DOM 技术可以实现高效的页面渲染和更新。在 AR 校园交互系统中，不同的 AR 场景需要不断地更新，React 可以通过虚拟 DOM 技术帮助我们快速更新渲染[18]。React 生态系统有许多开源库和工具，可以帮助我们更快地开发 AR 校园交互系统。例如，React Three Fiber 可以将 React 和 Three.js 集成起来，帮助我们更容易地实现 3D 场景和特效。在 React 作为前端骨干框架的基础上，前端使用了如下组件:

- MaterialUI: 提供响应式 UI 组件，具备极佳的动画效果与样式自定义系统。
- Redux: 提供公有数据的统一管理，便于 vDOM 中各个层级组件之间交互数据。
- Vite: 新一代打包工具，具备按需加载，快速响应等优点。
- fetch: ES6 提供的 http 请求 API。浏览器原生支持，无需额外封装 HttpRequest 请求。

后端框架采用 SpringBoot 框架，SpringBoot 框架可以快速构建应用程序，并且可以使用自动配置功能快速集成各种库和服务。这意味着可以更快地开发出高效、可靠的后端应用程序。强大的生态系统：SpringBoot 框架有着庞大的生态系统，拥有许多插件和第三方库，可以帮助开发人员更好地开发应用程序[19]。在 SpringBoot 作为核心后端框架的基础上，后端还用到了如下组件/技术:

• Spring Data JPA: 全 ORM 框架，能够非常方便快捷地与 SpringBoot 框架融合，提供了便捷的操作数据库的接口。

• MySQL: 免费开源的关系型数据库，拥有企业级数据处理速度。

在核心功能上，WebGL 采用 three.js 组件实现，考虑到前端主要框架是 React，因此采用 @react-three 组件，使用 JSX 语法进行开发。@react-three 可以完美融合进 React 框架，可以轻松地与 React,Redux,MaterialUI 集成。@react-three 可以通过 React 组件化的思路来创建 3D 场景。开发者可以使用熟悉的 React API 来管理 3D 对象和动画，并可以在 React 应用中轻松地嵌入 Three.js 场景，从而简化了开发流程[20]。AR 功能则采用 AR.js 组件，由于该组件无法直接通过 JSX 语法嵌入 React 组件，本系统编写了相应的 React 组件以适用系统整体开发风格。

4.1.3 功能模块设计

系统的模块可大致划分为: 基础功能模块，核心功能模块。其中基础功能模块包括登陆注册，用户权限管理，数据请求筛选等功能。核心功能模块利用 WebGL 与 AR 技术打造交互系统。其中 WebGL 功能模块以 WebGL 技术为核心，采用 three.js 组件实现 3D 模型的展示与交互功能。AR 功能模块以 WebGL 技术为基础，以 AR.js 为核心，采用图像识别与标记捕捉算法捕捉摄像机图像数据完成增强现实功能。功能模块的结构图如图4.2所示:

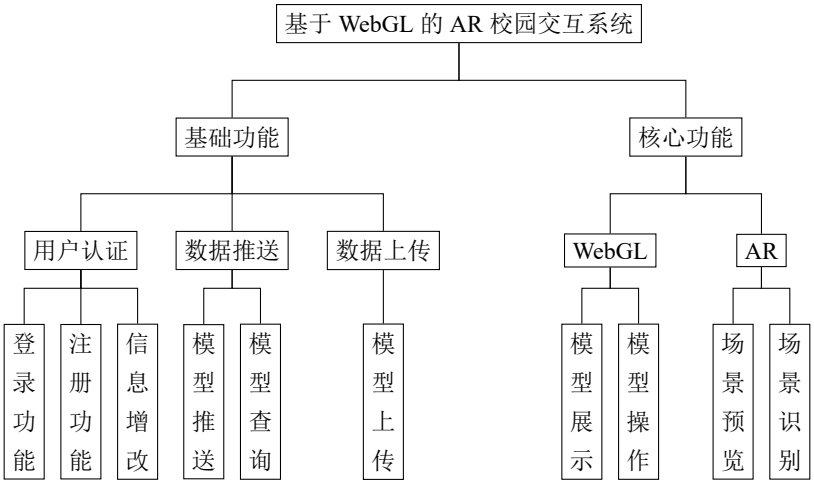

图 4.2 功能模块结构图

4.1.4 数据库设计

系统将文本数据存储在关系型数据库 MySQL 中，非文本数据 (图像，模型等) 存储在 COS 对象存储桶中。本系统共需标记表，标记信息表，模型表，学校人员表，用户表，收藏表，粉丝表，关注表九张表。各表介绍如下:

• 标记表: 记录 AR 识别的标记信息。数据段包括标记类型，标记图像类型，标记图像对应 URL，前端需要显示的模型 Id，标记信息 Id 等。通过这个表系统将实时匹配摄像头传输来的信息进行图像匹配，匹配成功后根据对应模型 Id 或信息 Id 显示内容。

• 标记信息表: 记录对应标记的信息。主要记录标记物的信息，如物品名称，简要介绍，上传作者等。前端通过该表在匹配到标记后显示对应标记的信息。

• 模型表: 记录 3D 模型信息，包括模型 URL，模型信息等。前端通过该表显示对应模型及相关信息。

- 模型控制表: 记录 3D 模型控制信息，影响显示时的场景设置，包括自旋速度，环境背景等。前端通过读取该表控制模型与环境。
- 模型点赞表: 记录模型被点赞的信息。
- 学校人员表: 学校教职工，学生信息表。
- 用户表: 记录用户信息，包括权限组，昵称，所属部门等。
- 用户收藏表: 记录用户收藏的模型内容。
- 用户关注表: 记录用户的关注用户数据。

各数据表的详细字段信息如下:

标记表记录图像识别标记信息，其中标记类型 (marker_type) 分为常规标记 (common) 与区域标记 (area)。所有常规标记都将一次性传输到用户设备中进行匹配，区域标记根据用户设备的地理位置判断是否传输。标记文件类型 (file_type) 对应图像识别或标记捕捉文件的类型。标记表完整字段如表4.1所示:

表 4.1 标记表

字段名称	字段类型	非空	描述
id	bigint	√	主键，自增。
marker_type	Enum(‘common’,‘area’)	√	标记类型, 常规标记或区域标记。
file_type	Enum(‘pattern’,‘barcode’,‘nft’)	√	标记文件类型，对应标记文件的格式。
file_url	varchar(255)	√	标记或图像对应的 URL。
info_id	bigint	×	需要显示的标记信息，对应标记信息表 id。
model_id	bigint	×	需要显示的模型 id, 对应模型表 id。
author_id	bigint	×	标记上传者的 id, 对应用户表 id。
lat	float	×	区域标记的地理经度。
lon	float	×	区域标记的地理纬度。
create_time	datetime	√	标记创建的时间。
update_time	datetime	√	标记更新的时间。

标记信息表记录与标记相关的信息数据, 主要存储标记的文字介绍信息，该表完整字段如表4.2所示:

表 4.2 标记信息表

字段名称	字段类型	非空	描述
id	bigint	√	主键，自增。
cover_url	varchar(255)	√	标记信息的封面图。
title	varchar(20)	√	标记的名称或标题。
abstract_info	varchar(255)	√	标记的简介。
detail_info	varchar(65535)	×	标记的详细信息。
author_id	bigint	×	标记上传者的 id, 对应用户表 id。
location	varchar(20)	×	标记对应的地理位置。
create_time	datetime	√	标记创建的时间。
update_time	datetime	√	标记更新的时间。

模型表用于记录模型 URL 与相关数据信息，该表完整字段如表4.3所示:

表 4.3 模型表

字段名称	字段类型	非空	描述
id	bigint	√	主键，自增。
cover_url	varchar(255)	√	模型的封面图。
title	varchar(20)	√	模型的名称。
abstract	varchar(255)	√	模型的介绍。
model_url	varchar(255)	√	模型存储的地址。
author_id	bigint	×	作者 id。
control_id	bigint	×	模型控制表 id。
create_time	datetime	√	模型创建的时间。
update_time	datetime	√	模型更新的时间。

模型控制表用于存储模型与环境的配置信息，由于控制信息仅在模型预览时会被使用，在 AR 预览与 AR 识别时仅需提供模型的基本信息即可，因此将模型表与模型控制表分离。模型控制表完整字段如表4.4所示:

表 4.4 模型控制表

字段名称	字段类型	非空	描述
id	bigint	√	主键，自增。
auto_rotate_speed	int	×	模型自动旋转速度。
background	boolean	×	模型是否需要环境背景。
preset	Enum(‘sunset’, ‘dawn’,...)	×	模型使用的环境背景。
blur	float	×	环境背景模糊程度。
float_speed	float	×	模型浮动效果速度。
rotation_intensity	float	×	模型浮动时自旋速度。
float_intensity	float	×	模型浮动强度。
create_time	datetime	√	模型控制信息创建的时间。
update_time	datetime	√	模型控制信息更新的时间。

模型点赞表用于记录点赞用户与模型之间的关系。主要用于计算模型被点赞数，防止同一用户多次点赞。该表完整字段如表4.5所示:

表 4.5 模型点赞表

字段名称	字段类型	非空	描述
id	bigint	√	主键，自增。
user_id	bigint	√	点赞用户 id，对应用户表主键。
model_id	bigint	√	模型 id，对应模型表主键。
create_time	datetime	√	点赞时间。

学校人员表记录学生与教职工的信息，该表主要用于模拟学校的人员信息表。该表完整字段如表4.6所示:

表 4.6 学校人员表

字段名称	字段类型	非空	描述
id	bigint	√	主键，自增。
identity	Enum(‘student’,‘teacher’,‘staff’,...)	√	人员身份。
real_name	varchar(20)	√	真实姓名。
id_number	char(18)	√	身份证号。
create_time	datetime	√	人员信息创建的时间。
update_time	datetime	√	人员信息更新的时间。

用户表记录系统用户的详细信息，用户表数据在创建时需查验学校人员表。该表完整字段如表4.7所示:

表 4.7 用户表

字段名称	字段类型	非空	描述
id	bigint	√	主键，自增。
name	varchar(20)	√	用户昵称。
password	varchar(20)	√	用户密码。
email	varchar(50)	√	用户邮箱。
avatar_url	varchar(255)	×	用户头像 url。
sexual	Enum(‘male’, ‘female’, ‘secret’, ‘undefined’)	√	用户性别。
permission	Enum(‘student’, ‘teacher’, ‘admin’, ‘visitor’)	√	用户所属权限组。
college	varchar(255)	×	用户所属学院。
department	varchar(255)	×	用户所属部门，专业。
signature	varchar(255)	×	用户个性签名。
tags	varchar(255)	×	用户身份标签。
ban	boolean	√	用户是否被封禁或注销。
create_time	datetime	√	用户信息创建的时间。
update_time	datetime	√	用户信息更新的时间。

用户收藏表用于记录用户与被收藏模型之间的关系。用户收藏表完整字段如表4.8所示:

表 4.8 用户收藏表

字段名称	字段类型	非空	描述
id	bigint	√	主键，自增。
user_id	bigint	√	用户 id，对应用户表主键。
model_id	bigint	√	模型 id，对应模型表主键。
create_time	datetime	√	收藏时间。

用户关注表用于记录用户之间的关注关系。该表完整字段如表4.9所示:

表 4.9 用户关注表

字段名称	字段类型	非空	描述
id	bigint	√	主键，自增。
user_id	bigint	√	被关注者 id，对应用户表主键。
follower_id	bigint	√	关注者 id，对应模型表主键。
create_time	datetime	√	关注时间。

4.2 基础功能详细设计

4.2.1 用户认证

用户认证模块包含用户注册，用户登录，用户信息增改三个主要功能。

在认证过程中，系统将先在前端检查数据，主要检查内容包括数据格式，用户权限等。无误后再执行后端逻辑。

用户注册依赖于已有的学校人员信息表，仅匹配到对应的数据后允许注册，并根据人员身份赋予对应的权限组。

整个用户认证模块涉及到的主要界面与所有数据表包括:

- 界面: 登陆界面，注册界面，个人资料界面。
- 数据表: 学校人员表, 用户表, 用户收藏表, 用户关注表。

用户认证模块整体操作的顺序图如图4.3所示:

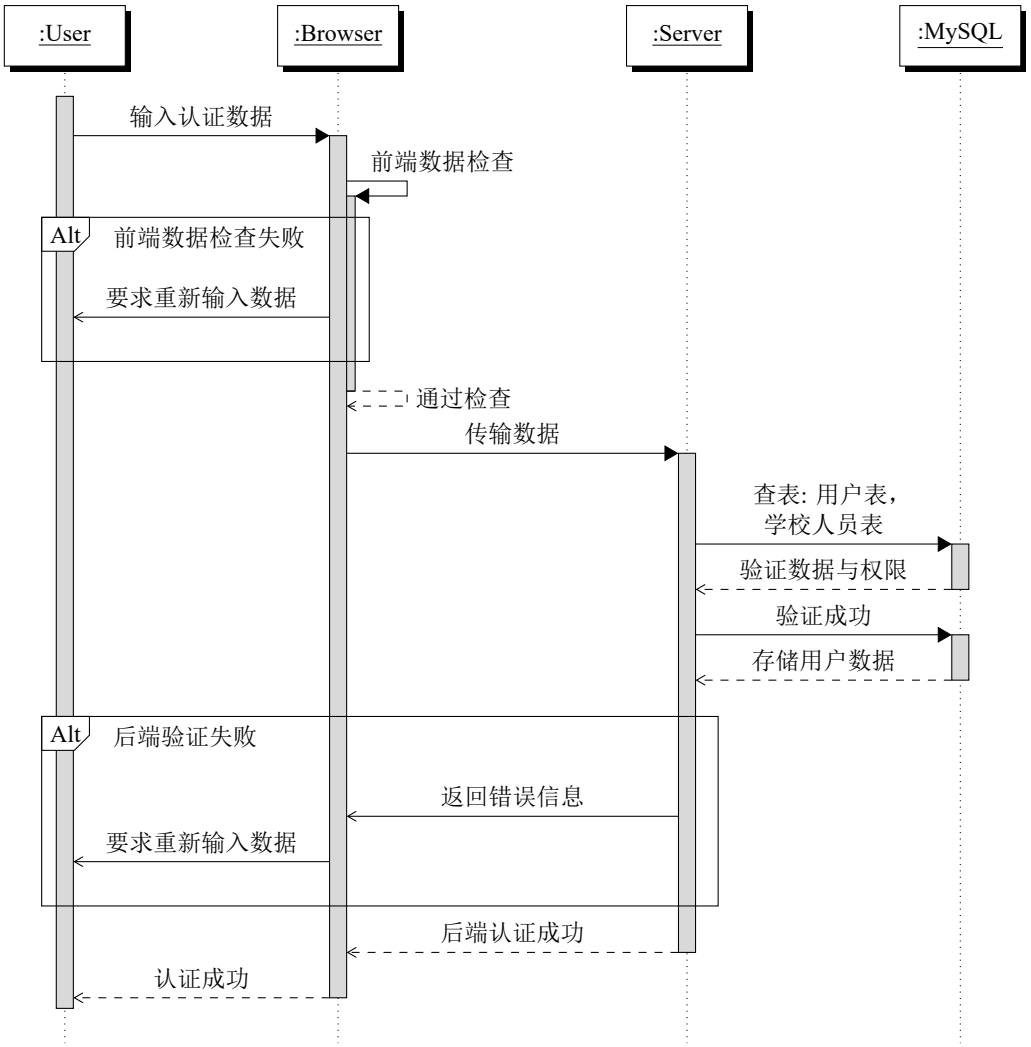

图 4.3 用户认证顺序图

4.2.2 数据推送

数据推送模块包括模型推送与模型查询两个主要功能。

模型推送根据用户在 “主页” 界面的分区选项 (“为你推荐”, “最新更新”, “我的关注”) 推送相关内容。三个分区对应后端三种不同的算法:

- 为你推荐: 推送最新创建的模型。
- 最新更新: 推送最新更新的模型。
- 我的关注: 推荐用户关注的用户发布的模型。

模型查询根据用户输入的模型名称模糊匹配数据库字段并返回相应内容。整个数据推送模块涉及到的主要界面与所有数据表包括:

- 界面: “首页” 界面。
- 数据表: 模型表，用户表，模型点赞表，用户关注表，用户收藏表。

数据推送模块整体操作的顺序图如图4.4 所示:

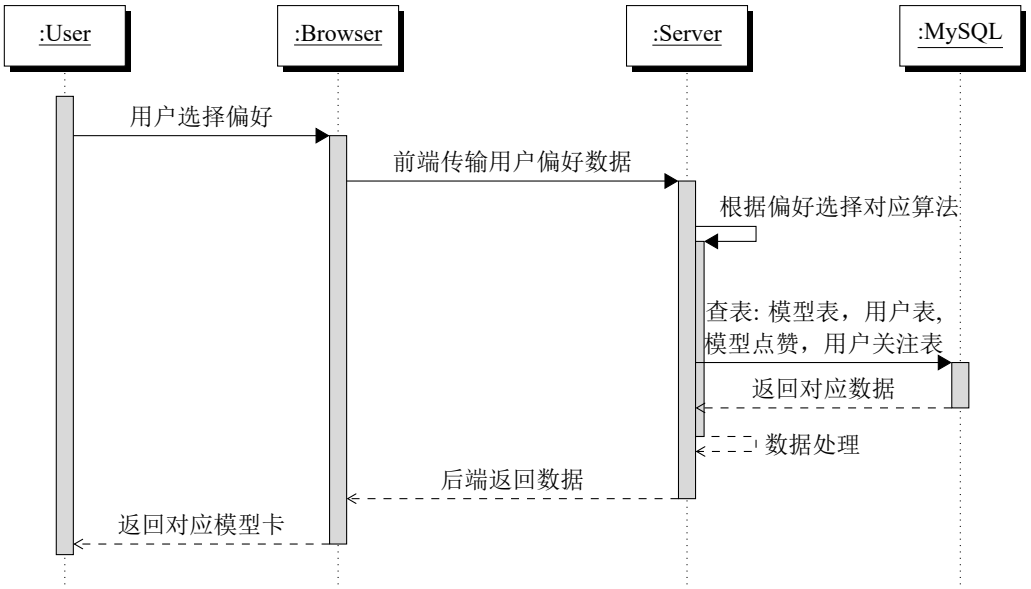

图 4.4 数据推送顺序图

4.2.3 数据上传

数据上传模块包括模型上传功能。

模型上传功能以 3D 模型为核心，上传模型文件，模型描述等，模型文件仅支持 glb 格式的模型，需要用户另行存储模型数据，系统通过 url 在前端获取模型。

整个数据上传模块涉及到的主要界面与所有数据表包括:

- 界面: “上传” 界面。
- 数据表: 模型表，模型控制表。

数据上传模块整体操作的顺序图如图4.5 所示:

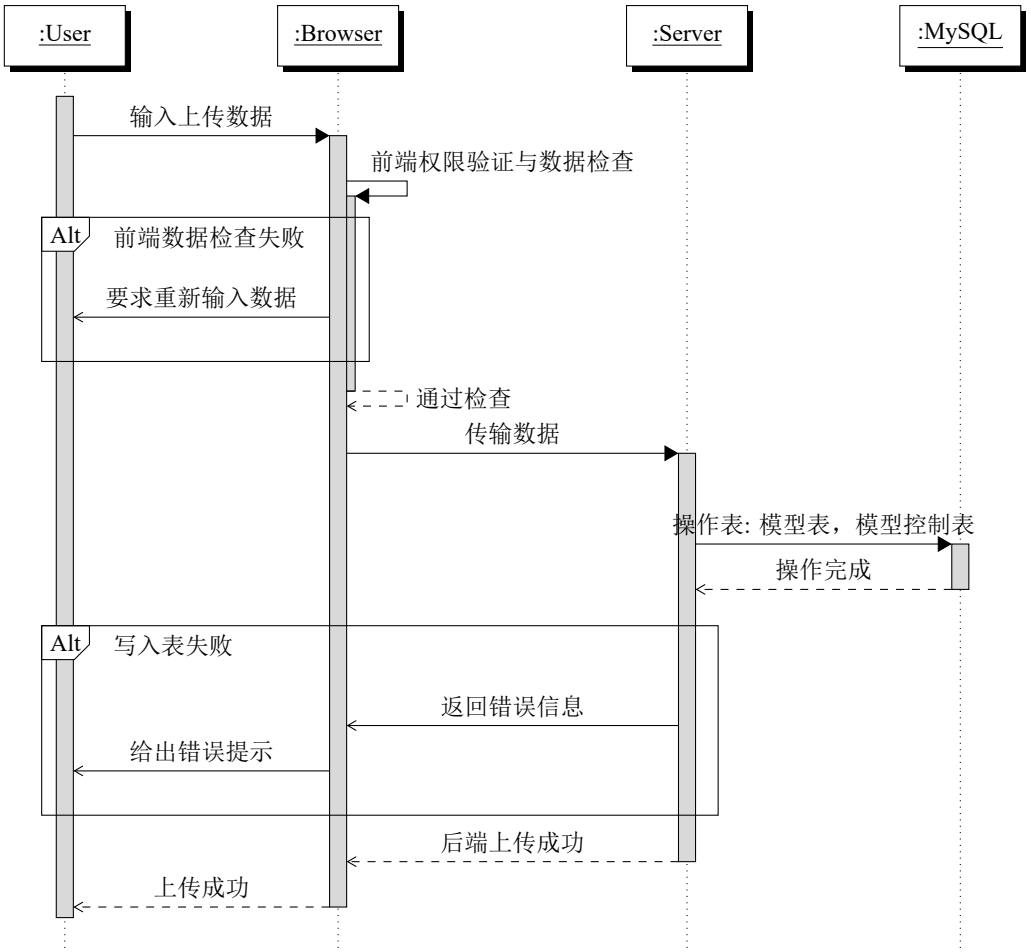

图 4.5 数据上传顺序图

4.3 核心功能详细设计

系统的核心功能位于前端，这小节会详细说明前端各个组件之间如何协作完成核心功能，涉及到的组件包括:

- @ar-js-org/AR.js: AR 功能核心组件，用于进行图象识别或标记捕捉，下文简称 AR.js
- react: 前端主要框架。
- @reduxjs/toolkit: 状态管理组件，下文简称 redux。
- leva: 控制板交互组件，允许用户通过 UI 修改数据。
- three: three.js, WebGL 技术的具体实现。
- @react-three/fiber: 在 react 框架基础上对 three.js 的封装，采用了 jsx 语法, 下文 three.js 具体实现上采用的是该组件。
- @react-three/drei: 基于 @react-three/fiber, 提供了预设环境，下文简称 drei。
- @react-three/xr: 基于 @react-three/fiber, 封装了 three.js 的 XR(包括 AR) 功能，下文简称 xr。

4.3.1 WebGL 模块

WebGL 模块主要功能是显示三维模型，且实现通过触摸或修改控制板数据的方式操作三维物体与对应的环境。主要包括模型显示，模型操作功能。

系统在接受到模型数据后，会首先创建对应的界面，显示模型信息，在控制板中显示可操作的数据。由于模型数据较大，系统将异步请求模型数据。系统可通过悬浮按钮的诸多选项调整界面模式，例如: 屏蔽信息卡，开关提示卡。

WebGL 模块被化拆分为如下具体功能模块:

- 控制模块: 配置模型控制信息，采用 drei 组件的轨道控制器 (OrbitControls)，用户可以通过滑动，双指放大等操作与模型交互。
- 环境模块: 配置模型虚拟环境信息，采用 drei 组建的环境组件 (Environment)，预设自然环境为落日，可提供模型控制表配置。
- 浮动模块: 操作物体浮动状态，使得物体可以自旋，上下浮动，达到更好的展示效果，可提供模型控制表配置。
- 光照模块: 提供环境光与阴影效果，可提供模型控制表配置。
- 网格模块: 通过网络请求获取模型本身。

此外，用户也可以通过 leva 控制板修改环境模块，浮动模块，光照模块数据，进而操作模型。

整个 WebGL 模块涉及到的主要界面与所有数据表包括:

- 界面: “模型预览” 界面，“AR 预览” 界面，“AR 识别” 界面。
- 数据表: 模型表，模型控制表，模型点赞表。

WebGL 模块中前端各关键模块协作关系如图4.6所示:

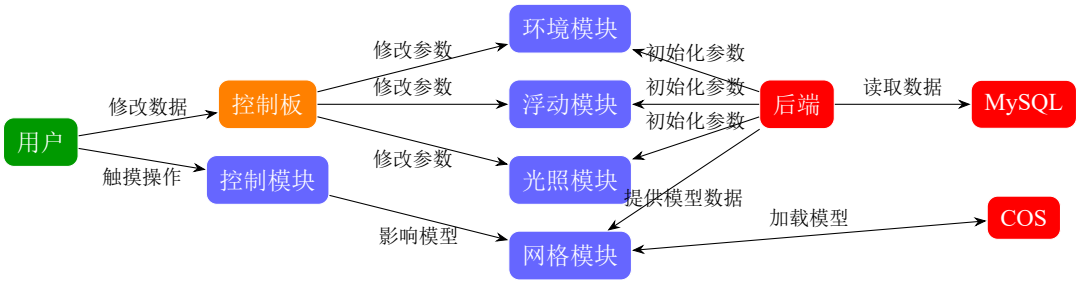

图 4.6 WebGL 模块协作关系

4.3.2 AR 模块

AR 模块在 WebGL 模块基础上引入 AR 功能，通过图像识别与标记捕捉的方法实时解析摄像头传输的图像信息并作出响应。主要包括场景预览与场景识别功能。

系统在进入 AR 模块后，会首先获取摄像头权限并显示现实场景。在场景预览功能中，将虚拟物体投射到真实场景中，用户可以通过仪表盘或触摸操作调整模型。在场景识别功能中，系统会调用 AR.js 提供的识别功能捕捉图像信息，在成功捕捉到信息后，屏幕将会显示对应的 3D 模型或标记信息。

AR 模块在 WebGL 模块基础上进行了以下修改:

- AR 网格模块: 允许用户通过控制板对模型进行缩放，旋转等操作。

- AR 控制模块: 提供辅助坐标轴，允许用户通过触摸屏幕对模型进行缩放，旋转等操作。
- AR 识别模块: 整合 AR.js 的识别功能。

AR 模块各关键模块协作关系如图4.7所示:

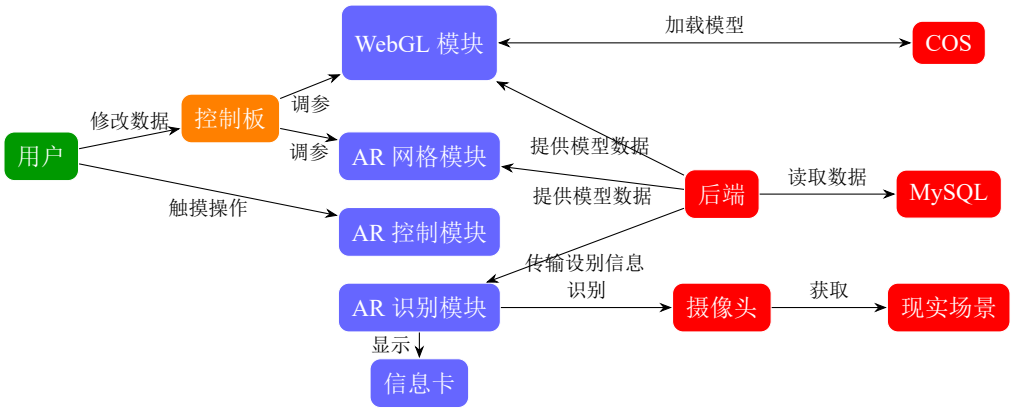

图 4.7 AR 模块协作关系

整个 AR 模块涉及到的主要界面与所有数据表包括:

- 界面: “AR 预览” 界面，“AR 识别” 界面。
- 数据表: 标记表, 标记信息表，模型表。

5 系统实现

5.1 基础功能实现

5.1.1 用户认证

用户认证包含登录界面，注册界面，用户资料界面。

用户进入系统将自动进入欢迎界面，在该界面中，用户可以选择登录，注册或以游客身份登录。如果以游客身份登录则直接进入系统，使用一个前端内置的账户，仅提供体验功能，无法与后端进行交互。

登录界面提供基础的登录服务，可通过密码和账户名或邮箱名登录。用户勾选 “记住我” 选项后可将登录记录保留在本地，在有效时间内均可自动进入系统。如果用户直接进入系统，系统会调用浏览器 localhost 数据判断用户近期是否有登录记录，如果有且满足登录条件则自动登录，否则使用游客账户进入系统，代码如下:

```
const getInitUserId = (): string => {
  // 判断上次登录时间是否在一周内
  const checkLoginData = (loginData: Date): boolean => {
    const now = new Date();
    const spanDay = getTimeSpan(now, loginData);
    return spanDay <= 7;
  };
  const userId = localStorage.getItem("userId");
  const lastLoginDate = localStorage.getItem("lastLoginDate");
  // 存在用户记录且满足自动登录条件
  if (
    userId !== null &&
```

```
    lastLoginDate !== null &&
    checkLoginData(new Date(lastLoginDate))
  )
    return userId;
  // 不存在，则用游客身份登录
  else return "visitor";
};
```

此外界面也可跳转到注册界面或直接以游客身份进入系统，登录界面如图5.1所示:

图 5.1 登录界面

注册界面提供新用户注册功能，注册逻辑与常用手机 app 相同，必填数据选以 * 标注。后端获取注册数据后会进行如下判断: 用户是否已被注册，用户的权限组是否匹配。注册逻辑代码如下所示:

```
public RespBean registry(RegistryDTO registryDTO) {
    Long id = registryDTO.getId();
    String name = registryDTO.getName();
    String password = RSAUtils.decrypt(registryDTO.getPassword());
    String permission = registryDTO.getPermission();
    String email = registryDTO.getEmail();
    String avatarUrl = registryDTO.getAvatarUrl();
    System.out.println(password);
    // 数据库存在同名用户
    if (hasName(name))
    return RespBean.error("该用户名已被注册，请换一个");
    // 邮箱已被注册
    if (hasEmail(email))
    return RespBean.error("该邮箱已被注册");
    // 该 id 已经被注册
    if (userRepository.findById(id).isPresent())
        return RespBean.error("该编号对应的账户已注册");
    // 验证校园表用户权限
    Optional<UniversityUser> universityUser = universityUserRepository.findById(id);
```

```
    System.out.println(permission);
    if (!universityUser.isPresent())
        return RespBean.error("校园系统中不存在此学号/编号");
    else if (!universityUser.get().getIdentity().equals(permission))
        return RespBean.error("权限不匹配");
    // 注册用户
    Timestamp createTime = new Timestamp(System.currentTimeMillis());
    User newUser = User.builder().id(
            id).name(name)
            .password(password).permission(permission)
            .email(email).avatarUrl(avatarUrl).createTime(createTime).updateTime(createTime).build();
    System.out.println(newUser);
    userRepository.save(newUser);
    return RespBean.success("注册成功!");
}
```

注册界面如图5.2所示:

图 5.2 注册界面

用户资料界面用于显示个人信息，同时提供一些快捷服务，用户可以在此修改个人信息。当用户点击 “收藏” 等按钮时，界面将弹出提示框，同时向后端发送数据请求，然后显示对应的数据。用户资料界面如图5.3所示:

图 5.3 用户资料界面

5.1.2 数据推送

数据推送模块包含首页界面。

用户进入首页可根据需求选择分区: “为你推荐”, “最新更新”, “我的关注”。后端将根据用户选择采用对应算法传输可用数据到前端。同时用户也可手动搜索内容。显示主体内容的代码如下所示:

```
const initModels = async () => {
  let promise;
  let data: ModelApiType[] = [];
  // 如果存在搜索内容，显示搜索内容
  if (searchContext) promise = searchModelsByTitleApi(searchContext);
  // 不存在搜索内容，显示分区内容
  else {
    // 获取推荐内容
    if (tab === HomeTab.recommend) promise = getRecommendModelApi();
    // 获取最新更新内容
    if (tab === HomeTab.new) promise = getLatestModelApi();
    // 获取用户订阅的内容
    if (tab === HomeTab.star) promise = getSubscribeModelApi(userId);
  }
  // 等待后端数据传输
  if (promise) data = (await (await promise).json()).data;
  const modelsArray = await modelAdapter(data);
  // 设置内容
  setModels(modelsArray);
};
```

主页界面中将显示各个模型的简要信息，包括预览图片，名称，作者等，用户点击对应卡片即可预览 3D 模型。“主页” 界面如图5.4所示:

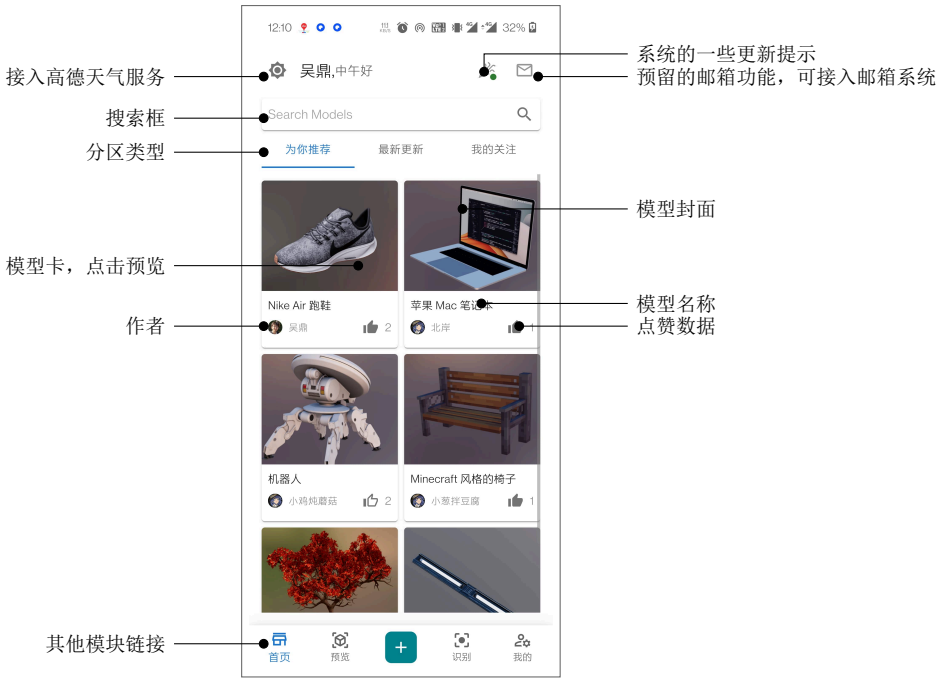

图 5.4 主页界面

5.1.3 数据上传

数据上传包含“上传”界面。

用户在下方点击加号按钮后点击模型上传即可进入“上传”界面，仅有管理员用户可进入该界面，其他用户尝试进入则会发出警告。

模型上传需填写必要数据，然后点击最下方的“预览按钮”，系统会渲染通过用户在表单中填入的数据渲染模型。用户根据预览效果选择上传模型或继续修改。

后端在接受到数据后会自行判断是否存在控制数据，如果存在则在模型控制表中添加内容，并将其关联到模型上。后端创建模型的代码如下:

```
public RespBean uploadModel(ModelUploadDTO dto) {
    Long modelControlId = null;
    // 存在控制信息
    if (dto.hasControl()) {
        ModelControl newModelControl =
             ModelControl.builder().autoRotateSpeed(dto.getAutoRotateSpeed())
                .background(dto.getBackground()).preset(dto.getPreset()).blur(dto.getBlur())
                .speed(dto.getSpeed()).rotationIntensity(dto.getRotationIntensity())
                .floatIntensity(dto.getFloatIntensity()).build();
        // 存入控制信息，同时获得 id
        newModelControl = modelControlRepository.save(newModelControl);
        modelControlId = newModelControl.getId();
    }
    // 创建模型，将控制信息 id 存入模型数据
    Model newModel = Model.builder().modelUrl(dto.getModelUrl()).coverUrl(dto.getCoverUrl())
            .title(dto.getTitle())
            .abstractInfo(dto.getInfo()).authorId(dto.getAuthorId())
            .controlId(modelControlId).build();
    modelRepository.save(newModel);
    return RespBean.success("成功上传模型");
}
```

上传界面如图5.5所示:

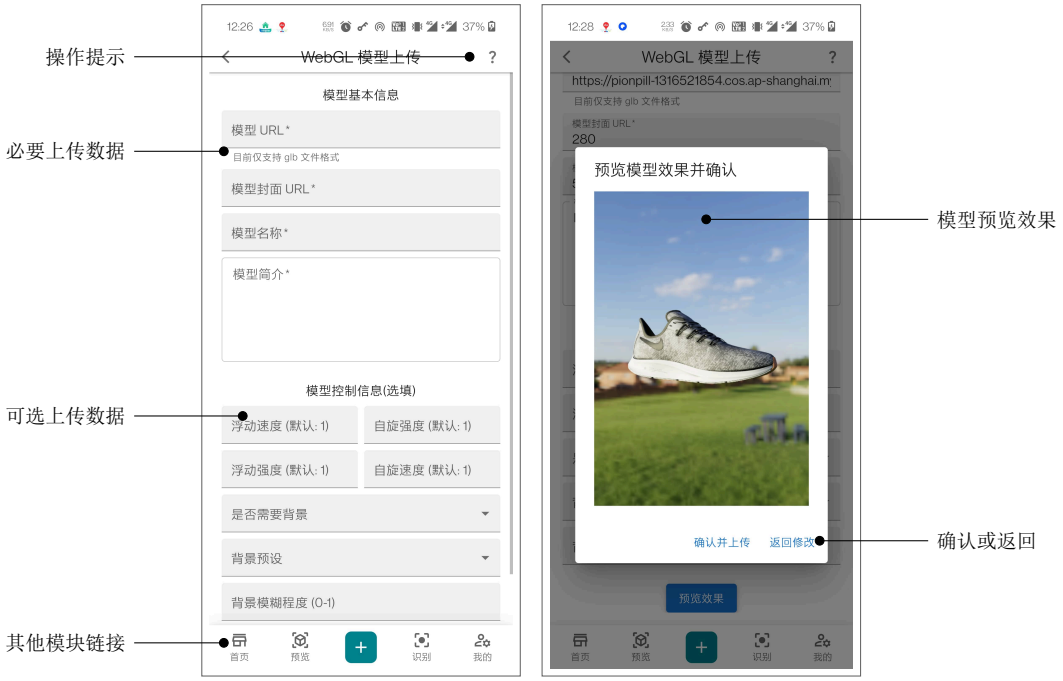

图 5.5 上传界面

5.2 核心功能实现

5.2.1 WebGL 模块

WebGL 模块包含模型预览界面。

用户进入模型预览界面后，系统将搭建场景，显示模型信息，请求模型信息数据:

- 搭建场景: 系统通过模型 id 查模型表和模型控制表，获取模型 url 和相关配置信息。此时系统将显示场景。
- 请求模型: 系统获取模型 url 后异步请求模型数据，继而在界面上显示。
- 模型信息: 系统向后端请求模型的文本，点赞，作者等信息。

其中搭建场景，显示模型信息的功能同步进行，模型数据异步请求。前端请求数据，前端请求模型及其相关数据的代码如下:

```
const initModel = async (id: string) => {
  // 获取模型基本数据
  let modelApiDate: ModelApiType | null = null;
  await (await getModelByIdApi(id)).json().then((response) => {
    modelApiDate = response.data;
  });
  // 查询模型被喜欢的数量
  let likeCount = 0;
  await (await getModelLikeCountsApi(String(modelApiDate!.id)))
    .json()
    .then((response) => (likeCount = response.data));
  // 将后端数据转换为前端数据
  const modelData: ThreeModelFields = {
    id: String(modelApiDate!.id),
    ......
```

```
  };
  // 获取模型作者信息
  const author = (await getUserById(String(modelApiDate!.authorId))) as User;
  const userData: UserShortFields = {
    id: author.id,
    ......
  };
  // 获取模型控制信息
  let modelControlApiDate: ModelControlApiType = {
    id: 0,
    ......
  };
  await (await getModelByIdApi(id)).json().then((response) => {
    modelControlApiDate = response.data;
  });
  // 将后端控制信息转换为前端数据
  const controlDate = modelControlApiDate as CommonControlFields &
    EnvironmentFields & FloatFields;
  // 创建模型实例
  const model: ThreeModel = ThreeModel.fromJson(modelData,userData,controlDate);
  // 设置模型信息
  setModel(model);
};
```

预览界面将模型信息分为简要信息与详细信息，简要信息仅以浮动卡片形式显现，用户可隐藏相关信息。详细信息则无法隐藏。用户在移动设备上可通过触摸移动位置，双指缩放模型大小，也可以通过控制板调整环境信息，例如背景贴图，背景模糊程度。模型预览界面如图5.6所示:

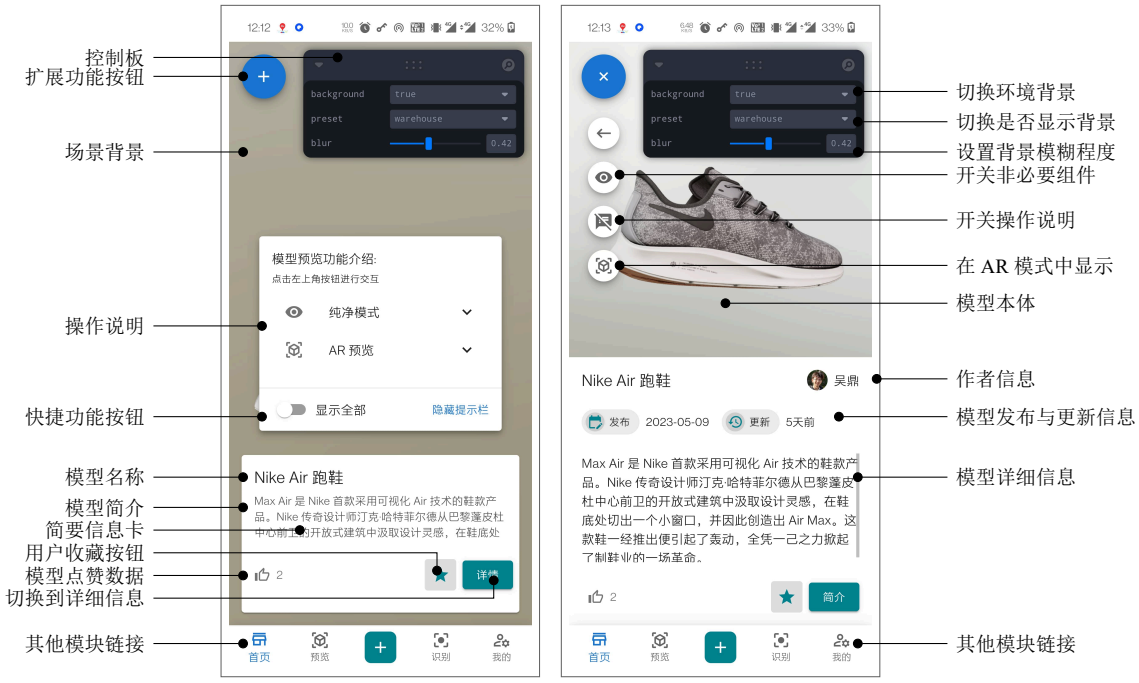

图 5.6 模型预览界面

5.2.2 AR 模块

AR 模块包括 AR 预览与 AR 识别界面。

进入 AR 模式将提醒用户需要的设备条件，如果用户的设备不满足任意条件，AR 按钮将显示 AR unsupported。设备需要满足的条件如下:

- 摄像头: 浏览器需要获取摄像头权限, 确保访问的是 https 协议网站，或者手动调整浏览器信任对应网站。否则浏览器无法获得摄像头权限。
- AR 服务: 安卓智能机需开启 AR 服务，请在 google play 安装“面向 AR 的 Google Play 服务”插件。苹果设备开启 ARKit 服务。
- 浏览器: 尽量使用 chrome 浏览器，AR 功能至少需要 chrome 80+ 版本。安卓设备尽量保持在系统 11+ 版本。
- 网络环境: 出于优化考虑，系统不会一次推送所有数据，AR 部分功能需要实时获取网络资源。

AR 提示界面如图5.7所示:

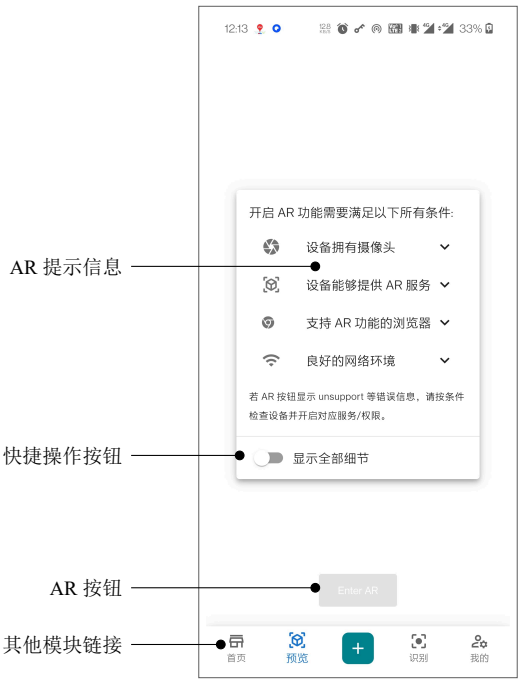

图 5.7 AR 提示界面

进入 AR 预览界面后，系统会请求手机摄像头权限，用户确认之后，系统将实时获取显示影像，并将 3D 模式显示在界面中，用户可以通过控制板或手动调整操作模型。系统将实时计算模型所在位置，AR 预览界面如图5.8所示:

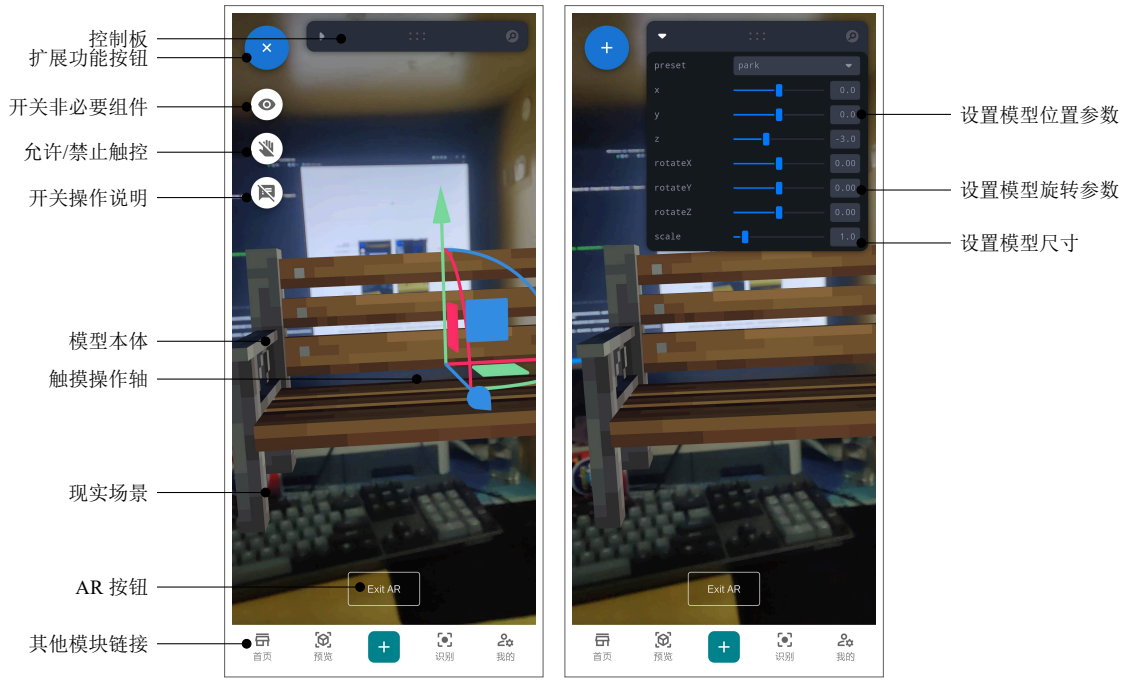

图 5.8 AR 预览界面

进入 AR 识别界面后，系统首先会加载常规标记数据，同时尝试访问设备定位权限，如果成功获取定位信息，加载地理区域内的特定标记数据。

AR 识别功能支持的两种识别技术优缺点如下:

- 标记捕捉: 图形具有特定要求，数据量小，适合较普遍的现实图像。
- 图像识别: 数据量大，可识别任何特想，适合特殊的现实图像。

系统使用标记捕捉技术识别常规图像，进入系统后则加载所有的标记捕捉数据。系统主要使用图像识别技术识别特定场景，例如图书馆，名人雕像等。由于图像识别需要大量数据，因此系统仅加载区域内数据，具体实现上根据地理坐标加载。

前端尝试获取定位信息，加载常规标记数据和地理区域内特定标记数据的代码如下所示:

```
const initMarkerData = () => {
  // 成功获取定位的回调函数: 向后端请求常规数据和区域内特定数据
  const successCallBack = async (data: any) => {
    const lon = data.coords.longitude;
    const lat = data.coords.latitude;
    const markers: Marker[] = [];
    // 请求特定数据
    await (await getMarkersByGeoApi(lon, lat))
    .json()
    .then(async (response) => {
      const markersApiData: MarkerApiType[] = response.data;
      for (const markerApiData of markersApiData) {
        const marker = await markerAdapter(markerApiData);
        markers.push(marker);
        }
        });
    // 请求常规数据
    await (await getCommonMarkersApi()).json().then(async (response) => {
      const markersApiData: MarkerApiType[] = response.data;
      for (const markerApiData of markersApiData) {
```

```
        const marker = await markerAdapter(markerApiData);
        markers.push(marker);
      }
    });
    setMarkers(markers);
    setInit(true);
  };
  // 获取定位失败的回调函数：仅仅向后端请求常规数据
  const errorCallBack = async () => {
    // 代码和成功回调函数的请求常规数据类似
    ......
  };
  // 获取定位数据
  navigator.geolocation.getCurrentPosition(successCallBack, errorCallBack);
};
```

针对地理区域内特定标记数据，后端会在数据库中根据经纬度坐标进行判断，默认加载偏差在 2km 内的特定数据 (即对应经纬度 0.02 内的数据)，后端代码如下:

```
@Override
public RespBean getMarkersByGeo(Float lon, Float lat) {
    // 范围: 0.01 经纬度，即 1km
    Float len = 0.01f;
    List<Marker> markersList = markerRepository.findAllByLonBetweenAndLatBetween(
        lon - len, lon + len, lat - len, lat + len);
    return RespBean.success("获取识别标记数据", markersList);
}
```

加载完所有标记数据后，系统实时匹配摄像机传输的图像，如果匹配上对应的标记，界面将弹出简略信息卡，用户点击即可查看详细信息。如果存在模型会在标记法线上方显示模型。AR 识别界面如图5.9所示:

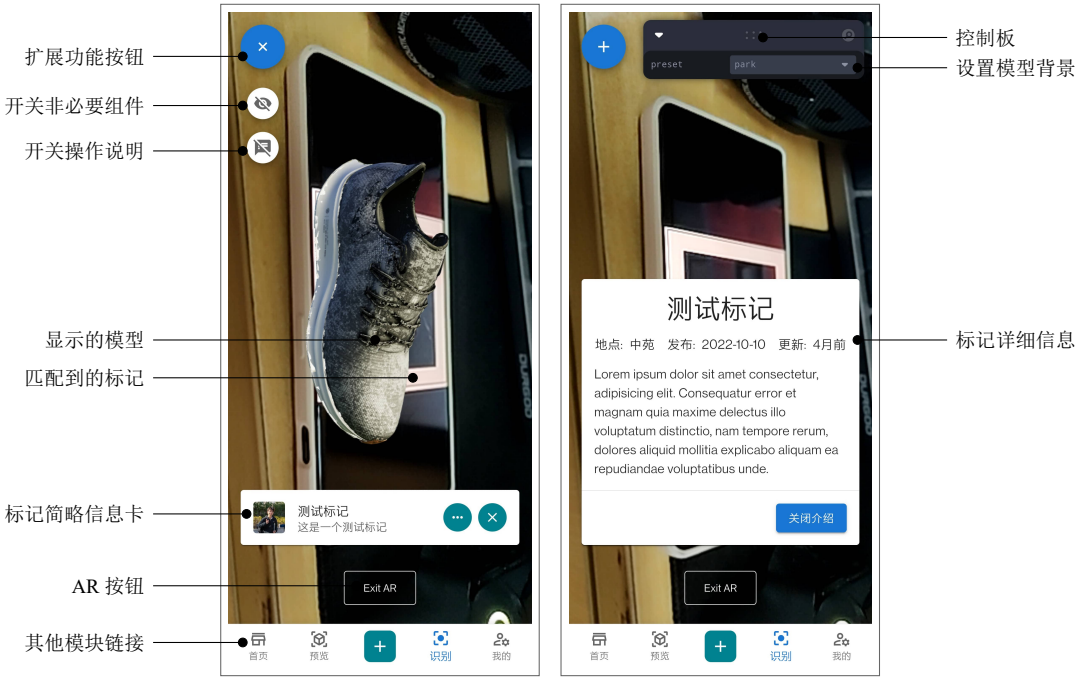

图 5.9 AR 识别界面

6 系统测试

软件测试的作用是发现程序中的错误，保证软件质量，检查软件是否符合客户要求。软件测试不仅要验证软件的功能是否实现，还要验证软件在真实使用环境下能否正常运行。通过测试发现的软件问题越多，交付给用户的软件质量就越高[21]。

6.1 测试环境

硬件环境:

- 服务器: 本机服务器 (4 核 8 线程，24GB，10M 贷款，100G 硬盘);
- 移动设备: 安卓手机 (Snapdragon 885 处理器, Android 11 操作系统);

软件环境:

- 数据库: MySQL 8.0;
- Java: JDK17;
- 浏览器: Chroe112(移动版)。

6.2 功能测试

6.2.1 用户认证模块测试

用户认证模块的功能包括用户登录，用户注册，用户信息增改。测试用例和结果如表6.1所示:

表 6.1 用户认证模块测试用例表

功能描述	用户登录，注册，信息修改	
用例目的	测试模块功能的可用性	
前提条件	用户时校园教职工或学生，在校园数据表中有记录	
输出/动作	**期望的输出/响应**	**实际情况**
进入登录界面，在表单中输入用户名或邮箱以及密码。点击登录按钮。	前端对输入数据进行判断，根据字符关键字判断是用户名还是密码。对用户名/邮箱和密码进行格式校验。若失败，前端给出提示。若成功将数据发送到后端验证，若后端验证失败，则前端界面显示对应信息，若成功，则跳转至系统主页面。	符合
进入注册界面，在表单中输入工号/学号，密码等必填信息；选择性填入可填信息。点击注册按钮。	前端对输入数据进行格式判断。若失败，前端给出提示。若成功将数据发送到后端验证，后端查询学校用户表，根据 id 匹配注册信息，若后端验证失败，则前端界面显示对应信息，若成功，则跳转至登录页面。	符合
登录账户后，在 “我的” 界面修改个人信息，点击提交按钮。	前端对输入数据进行格式判断。若失败，前端给出提示。若成功将数据发送到后端验证，后端将数据存储到对应表格，若失败，前端给出错误原因，若成功，更新用户信息。	符合
在登录时，勾选 “记住我” 选项。	退出系统后，在一周内登录系统无需手动登录，系统自动跳转至 “我的” 界面。	符合

6.2.2 数据推送模块测试

数据推送模块的功能包括模型推送与模型查询。测试用例和结果如表6.2所示:

表 6.2 数据推送模块测试用例表

功能描述	模型推送，模型查询	
用例目的	测试模块功能的可用性	
前提条件	用户进入系统	
输出/动作	**期望的输出/响应**	**实际情况**
进入主页界面，选择分区:“为你推荐”，“最新更新”，“我的关注”。	前端将对应类型传输到后端，后端根据类型选择不同算法，分别采用后端三种不同的算法: 点赞数较高模型，最新更新模型，用户关注作者模型。返回数据在前端显示。	符合
进入主页界面，在搜索框输入模型名称，点击搜索图标。	前端对搜索格式进行检测，若不符合要求则给出提示信息。若符合则后端根据搜索内容查模型表模型名称，然后返回查询到的结果数据到前端能显示。	符合

6.2.3 数据上传模块测试

数据上传模块的功能包括模型上传与 AR 标记上传。测试用例和结果如表6.3所示:

表 6.3 数据上传模块测试用例表

功能描述	模型上传，AR 标记上传	
用例目的	测试模块功能的可用性	
前提条件	用户进入系统	
输出/动作	**期望的输出/响应**	**实际情况**
进入上传界面，选择模型上传。在表格中填入模型 URL 等必要信息；选择性填入可选信息。点击上传按钮	前端对输入数据进行格式判断，对模型 URL 有效性检查。若检查失败，前端给出提示。若成功将数据发送到后端存储，后端检查是否存在相同数据，若存在返回错误信息，若存储成功，前端给出成功提示。	符合
进入上传界面，选择 AR 标记上传。在表格中填入标记 URL，标记类型等必要信息；选择性填入可选信息。点击上传按钮	前端对输入数据进行格式判断，对标记 URL 有效性检查。若检查失败，前端给出提示。若成功将数据发送到后端存储，后端检查是否存在相同数据，若存在返回错误信息，若成功，分别将数据存储到标记表，标记信息表。存储完成后前端给出成功提示。	符合

6.2.4 WebGL 模块测试

WebGL 模块的功能包括模型显示，模型控制功能。测试用例和结果如表6.4所示:

表 6.4 WebGL 模块测试用例表

功能描述	模型显示，模型控制	
用例目的	测试模块功能的可用性	
前提条件	用户进入系统	
输出/动作	**期望的输出/响应**	**实际情况**
进入模型预览界面，查看某个模型。	后端将模型信息，控制信息，环境信息等传输到前端。前端异步加载环境，模型，模型信息。用户可通过按钮切换模型摘要或模型详情模式，也可隐藏非必要卡片。	符合
进入模型预览界面，通过触摸操作模型，通过控制板调整数据操作模型。	根据用户操作，模型做出移动，缩放等响应。通过控制板的操作，场景将更换对应配置或显示新的场景。	符合

6.2.5 AR 模块测试

AR 模块的功能包括 AR 模型预览，AR 识别功能。其中 AR 识别分为图像识别与标记捕捉。测试用例和结果如表6.5所示:

表 6.5 AR 模块测试用例表

功能描述	AR 预览，AR 识别	
用例目的	测试模块功能的可用性	
前提条件	用户进入系统	
输出/动作	**期望的输出/响应**	**实际情况**
进入 AR 预览界面，通过摄像头在虚拟现实场景中显示模型。	系统获取摄像头权限并显示现实场景。将虚拟物体模型投射到真实场景中提供预览功能。	符合
进入 AR 预览界面，通过操纵轴触摸操作模型位置，控制模型进行旋转。	模型根据用户触摸操作做出响应，对应位置或旋转信息改变并显示在界面上。	符合
进入 AR 预览界面，通过控制板改变模型或场景信息。	模型或场景根据改变的信息实时应用，并显示在界面上。	符合
进入 AR 识别界面，摄像头对准可识别的图像，等待系统响应。	系统匹配到图像信息后，如果存在预定义的 3D 模型，则显示在界面上；如果存在标记信息，则界面弹出简要标记卡片。	符合
进入 AR 识别界面，摄像头对准可识别的标记，等待系统响应。	系统匹配到标记信息后，如果存在预定义的 3D 模型，则显示在界面上；如果存在标记信息，则界面弹出简要标记卡片。	符合
弹出简要标记卡片后点击查看详细信息。	系统弹出详细信息卡，同时停止图像识别功能，在关系信息卡后重新进行图像识别。	符合

6.3 性能测试

在性能测试过程中，针对需求分析中的非功能需求，对系统进行性能测试。具体测试方法为采用测试设备对系统各个功能接口同时进行测试，单个接口的测试量为 20 次，同类接口总体测试量大于 100，获得测试结果后取平均响应时间。测试用例和结果如表6.6所示。

表 6.6 性能测试表

输入动作	期望的输出/响应	实际情况
切换系统界面	界面加载时间 <1s, 网络加载时间 <2s	界面加载时间 <0.5s, 网络加载时间 <2s
点击按钮，弹出弹框并显示内容	界面加载时间 <1s	界面加载时间 <0.5s
主页选择不同数据分析，等待返回内容	界面加载时间 <2s	界面加载时间 <1.5s
主页搜索模型，等待返回内容	界面加载时间 <2s, 网络加载时间 <2s	界面加载时间 <2s, 网络加载时间 <1s
模型预览界面加载场景，并开始异步加载模型	界面加载时间 <2s, 网络加载时间 <2s	界面加载时间 <1s, 网络加载时间 <1s
AR 预览界面开启摄像头，开始异步加载模型	界面加载时间 <2s, 网络加载时间 <2s	界面加载时间 <1s, 网络加载时间 <1s
AR 识别界面开启识别图像信息	界面加载时间 <2s, 网络加载时间 <2s	界面加载时间 <1.5s, 网络加载时间 <2s
AR 识别界面开启识别标记信息	界面加载时间 <1.5s, 网络加载时间 <2s	界面加载时间 <1s, 网络加载时间 <2s

6.4 兼容性测试

本部分对主流浏览器做兼容性测试，测试系统的功能，页面布局是否正常显示。测试结果如表6.7所示:

表 6.7 兼容性测试表

测试目的	测试目标	期望结果	实际情况
检查页面布局，WebGL 模型显示效果，AR 预览与识别效果	Chrome	显示正常，功能正常	符合
检查页面布局，WebGL 模型显示效果，AR 预览与识别效果	Edge	显示正常，功能正常	符合
检查页面布局，WebGL 模型显示效果，AR 预览与识别效果	Firefox	显示正常，功能正常	符合

7 总结

本文以 WebGL 与 AR 技术为核心，介绍了 AR 交互系统的研究现状与研究意义，探索 AR 系统在 Web 领域的可行性，在完成技术选择，需求分析，系统整体设计，模块详细设计后，实现了基于 WebGL 的校园交互系统，并通过系统测试验证了系统的可用性，稳定性。系统以移动端浏览器为载体，调用操作系统底层 AR 服务，通过现有的 WebGL 与 AR 技术实现了模型预览，AR 预览，AR 识别等核心功能。

本系统的主要成果包括:

- 基于 AR.js 提供的标识识别与图像识别算法，开发出了 AR 识别系统，能够对校园特定场景与物体进行识别并提供对应信息。

• 基于 three.js 技术，开发了 Web 端三维模型显示系统，并提供了一定的交互功能。结合 AR.js 基础开发了 AR 交互功能。给用户提供了一种更生动，更直观的交互方式。

• 基于 React，MaterialUI 技术，搭建了有好的前端系统，以此为依托开发了核心的 WebGL 与 AR 功能。用户无需下载任何应用即可访问系统。

• 基于 SpringBoot 技术搭建了后端系统，向前端提供稳定数据。

本文研究的 AR 系统可以提供更完善的预览服务，基于用户更丰富的视觉体验，本系统可拓展到教育，建筑，工业等各个领域，并于已有系统结合，有效提高工作，学习效率。本系统完全基于开源框架，且基于 Web 系统。决绝传统 AR 服务闭源，兼容性差等特点。

本系统同样存在部分问题，系统依赖于高性能智能设备，需要设备提供 AR 服务，需要用户具备一定的 3D 模型知识。

参考文献：

[1] Mistry P, Maes P. SixthSense: A Wearable Gestural Interface[J]. Association for Computing Machinery, 2009 (1): 85.

[2] 蔡鐨，康佳美，王子娟. 用户体验设计指南从方法论到产品设计实践全彩[M]. 北京：电子工业出版社, 2021.

[3] 乔振华，黄志超. AR 技术在非物质文化遗产的保护与研究中的应用——以婺源傩舞为例[J]. 电脑知识与技术, 2019, 39.

[4] Hearn D, Baker M P, Baker M P. Computer graphics with OpenGL: volume 3[M]. Pearson Prentice Hall Upper Saddle River, NJ:, 2004.

[5] Chaudhry T, Juneja A, Rastogi S. AR foundation for augmented reality in unity[J]. Int. J. Adv. Eng. Manage., 2021, 3(1): 1-7.

[6] Henrysson A, Billinghurst M, Ollila M. Face to face collaborative AR on mobile phones[M]//Fourth ieee and acm international symposium on mixed and augmented reality (ismar'05). IEEE, 2005: 80-89.

[7] Tudose C, Odubăşteanu C. Object-relational Mapping Using JPA, Hibernate and Spring Data JPA[C]//2021 23rd International Conference on Control Systems and Computer Science (CSCS). IEEE, 2021: 424-431.

[8] Min Q, Wang Z, Liu N. An evaluation of HTML5 and WebGL for medical imaging applications[J]. Journal of healthcare engineering, 2018, 2018.

[9] Cantor D, Jones B. WebGL beginner's guide[M]. Packt Publishing Ltd, 2012.

[10] Almansoury F, Kpodjedo S, Boussaidi G E. Investigating Web3D topics on StackOverflow: a preliminary study of WebGL and Three. js[M]. 2020: 1-2.

[11] Baruah R, Baruah R. Creating an Augmented Reality Website with Three. js and the WebXR API[J]. AR and VR Using the WebXR API: Learn to Create Immersive Content with WebGL, Three. js, and A-Frame, 2021: 217-252.

[12] Jansen B, Goodwin T, Gupta V, et al. Performance evaluation of WebRTC-based video conferencing[J]. ACM SIGMETRICS Performance Evaluation Review, 2018, 45(3): 56-68.

[13] Dell A I, Bender J A, Branson K, et al. Automated image-based tracking and its application in ecology[J]. Trends in ecology & evolution, 2014, 29(7): 417-428.

[14] Cheng J C, Chen K, Chen W. Comparison of marker-based AR and markerless AR: A case study on indoor decoration system[M]. 2017: 483-490.

[15] Jabar M A, Sidi F, Azizan A, et al. Software requirement analysis template with automation aided system[M]// 2012 International Conference on Information Retrieval & Knowledge Management. IEEE, 2012: 235-239.

[16] 张宏升. 软件架构的非功能性需求指标和区域化支持[J]. 电脑知识与技术: 学术版, 2011(3X): 2085-2086.

[17] Zhang Y, Liu H, Su X, et al. Remote mobile health monitoring system based on smart phone and browser/server structure[J]. Journal of healthcare engineering, 2015, 6(4): 717-738.

[18] Gackenheimer C. Introduction to React[M]. Apress, 2015.

[19] Guntupally K, Devarakonda R, Kehoe K. Spring boot based REST API to improve data quality report generation for big scientific data: ARM data center example[M]. IEEE, 2018: 5328-5329.

[20] Mysore A. Splunk+ React=[M]. Pasadena, CA: Jet Propulsion Laboratory, National Aeronautics and Space …, 2021.

[21] 朱少民. 软件测试方法和技术[M]. 清华大学出版社有限公司, 2005.

致谢

时光白驹过隙，我在南信大已经学习了四年，在这段学习生涯中，我要向我的母校表达最深切的谢意，感谢学校在我大学期间为我提供的一流教育资源和培养机会。在这里，我不仅仅获得了专业知识和技能，还收获了友情和人生经验。

其次，我要感谢我的导师对我的指导和支持。导师的悉心教诲和鼓励，对我完成毕业设计起到了至关重要的作用。导师不仅仅在学术研究方面给予我指导，还关心我的个人成长和发展，给了我很多宝贵的建议和启示。

最后，我要感谢所有在我大学四年期间给予我帮助和支持的人。我的家人、朋友和同学们一直陪伴着我，给予我鼓励和支持。在这里，我要向你们表达最真挚的谢意和感激之情。