

《数据库系统原理》任务二

游戏平台技术报告

题目名称：“Meats”游戏发行平台设计

学号及姓名：

苏晰月 17377372

王佳瑞 17375180

栗亚舟 17377118

一、系统架构与设计

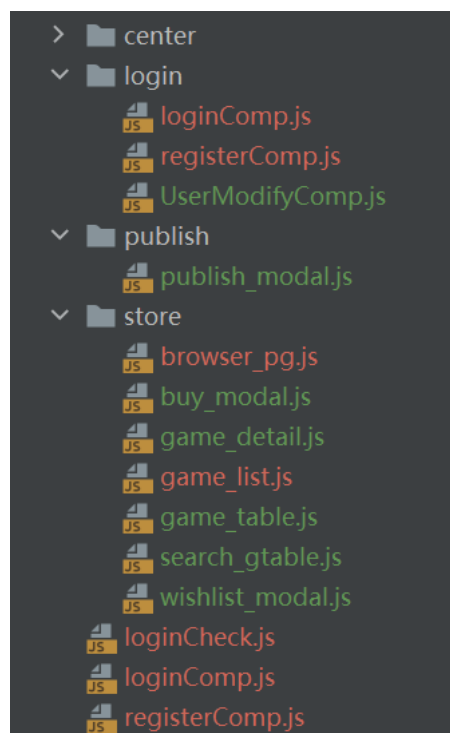
此项目严格前后端分离，分别采用 react.js 和 django 编写前后端框架。我们使用 MYSQL 作为数据库，而数据以本地服务器为存储介质，以 axios 为桥梁进行前后端的通信。

1.1 前端设计

前端用户设计界面主要基于简洁美观、开源的 react semantic ui 框架，并且辅以 antd ui 组件设计交互页面；此外，前端使用了 React Router，其为基于 React 的路由库，利于快速地添加视图和数据流，同时保持页面与 URL 间的同步。

1.1.1 架构设计

采用第二主流 react 设计，即页面功能划分，一个页面功能对应一个文件夹，这个页面功能所用到的 container, component, action, reducer 等文件都放在这个文件夹下。如下为按照页面功能划分的项目结构示例



这种结构好处很明显，一个页面功能使用到的组件，状态和行为都在同一个文件夹下，方便开发和维护，同时易于扩展；不足之处是依然同按角色划分

面临同样的问题，改变一个功能需要频繁的在 reducer, action, actionTypes 等不同文件夹间切换。另外 redux 将整个的应用状态放在一个 store 中来管理，不同功能模块之间可以共享 store 中的部分状态(项目越复杂，这种场景会越多)，共享的状态应该放到哪一个页面文件夹下也是一个问题，这些问题归根结底是因为 redux 中状态管理逻辑并不是根据页面功能划分的，它是页面功能之上的一种抽象。

1.1.2 组件划分后的文件树

1. 布局组件：在/container/文件夹下，主要包括主页面、社区页面、商店页面、登录相关页面和网站信息页面。这些 js 文件中仅仅涉及应用 UI 界面结构的组件，不涉及任何业务逻辑，数据请求及操作；
2. 容器组件：/component/文件夹下，主要负责与后端交互获取数据，处理业务逻辑；
3. 展示型组件：/component/文件夹下，负责应用的界面 UI 展示；
4. UI 组件：在此次设计中，几乎不存在无状态组件。

1.1.3 与后端交互

本文中，我们使用了基于 Promise 的库 -- axios。Axios 是一个基于 promise 的 HTTP 库，可以用在浏览器和 node.js 中。通过封装 axios 的 post, get 等方法于动作中，我们实现了前后端的交互。

二、数据库基本表的定义

1. Player，继承 User 类

id(自动生成)	Username	Password	Email

2. Developer，继承 User 类

id(自动生成)	Username	Password	Email

3. Game

Name	Grade	Price	Type	Developer	Pub_time	Avatar	N_comments
			射击..	Foreignkey	发布时间	图片	评分人数

4. Play 玩表, 记录某玩家游戏进度

Player	Game	Progress	Rate
		进度	评分

5. Dreamlist 心愿单

Player	Game

6. Depository 已购游戏仓库

Player	Game

7. Friend 关注表

Player1	Player2

二、系统重要功能实现方法

3.1 用户动作

(1) 注册 register

前端 post, email、username、password、type (注册类型) 到后端, 后端确认 username 未使用确认指定注册类型中邮箱未被注册在指定类型中注册新表项 (Player.objects.create_user())

(2) 登录:

后端检查当前是否是登录状态 (session user 是不是 None) 已登录则返回失败 前端 post 过来 username, password, usertype 后端调用 auth.login 函数检查 username, password 是否有效, 无效返回错误检查 usertype, 如 Player.objects.get(username=user) 登入 auth.login(), session 中保存 username, usertype 信息。

(3) 登出

后端检查当前是否是登录状态 (session user 是不是 None) 未登录则返回失败后端 auth.logout(), session 信息删除

(4) 添加关注

后端检查当前是否是登录状态 (session user 是不是 None) 未登录则返回失败 session 中 usertype 是否是 Player, 从 Player 表中根据 session 中 username 取出当前 user 前端 Post 关注用户名字, 后端从 Player 表检查是否存在用户 targetFriends 表中检查是否已关注该用户 Friends 表新建表项, player1=user, player2= target

(5) 取消关注:

后端检查当前是否是登录状态 (session user 是不是 None) 未登录则返回失败 session 中 usertype 是否是 Player, 从 Player 表中根据 session 中 username 取出当前 user 前端 Post 关注用户名字, 后端从 Player 表检查是否存在用户 targetFriends 表中检查是否已关注该用户, 未关注不可删除 Friends 表删除该表项。

3.2 玩家游戏操作

(1) 加入心愿单:

后端检查当前是否是登录状态 (session user 是不是 None) 未登录则返回失败 session 中 usertype 是否是 Player, 从 Player 表中根据 session 中

username 取出当前 user 从前端 Post 过来游戏名称，在 Game 列表查询游戏名称是否存在在 Dreamlist 中检查 user，game 组合是否存在，不可重复添加在 Dreamlist 中添加表项

(2) 购买游戏:

后端检查当前是否是登录状态 (session user 是不是 None) 未登录则返回失败 session 中 usertype 是否是 Player，从 Player 表中根据 session 中 username 取出当前 user 从前端 Post 过来游戏名称，在 Game 列表查询游戏名称是否存在在 Depository 查询游戏是否购买过，不可重复购买在 Depository 和 Play 表中加入表项，其中 Play 表 progress (游戏进度设 0)

(3) 从心愿单删除游戏:

后端检查当前是否是登录状态 (session user 是不是 None) 未登录则返回失败 session 中 usertype 是否是 Player，从 Player 表中根据 session 中 username 取出当前 user 从前端 Post 过来游戏名称，在 Game 列表查询游戏名称是否存在在 Dreamlist 中检查 user，game 组合是否存在，删除

(4) 游戏打分:

后端检查当前是否是登录状态 (session user 是不是 None) 未登录则返回失败 session 中 usertype 是否是 Player，从 Player 表中根据 session 中 username 取出当前 user 从前端 Post 过来游戏名称、用户打分，在 Depositor 过滤 username 列表查询游戏名称是否存在，购买过的游戏才能打分 Playlist 对应表项 rate 更新为 user_grade，Game 表中游戏表项 $grade = (grade + user_grade) / (n_commnets + 1)$, $n_comments += 1$

(5) 进度更新:

后端检查当前是否是登录状态 (session user 是不是 None) 未登录则返回失败 session 中 usertype 是否是 Player，从 Player 表中根据 session 中 username 取出当前 user 从前端 Post 过来游戏名称、用户进度增长，在 Play 查询 username 列表查询游戏名称是否存在，购买过的游戏才能有进度表项 grade 增加

3.3 开发商游戏操作

(1) 发布游戏:

后端检查当前是否是登录状态 (session user 是不是 None) 未登录则返回失败 session 中 usertype 是否是 Developer 前端 Post 得到游戏名称、类型、图像文件、标价在 Game 列表检查游戏名称是否已存在 Game 列表添加表项，开放商设为当前登录用户

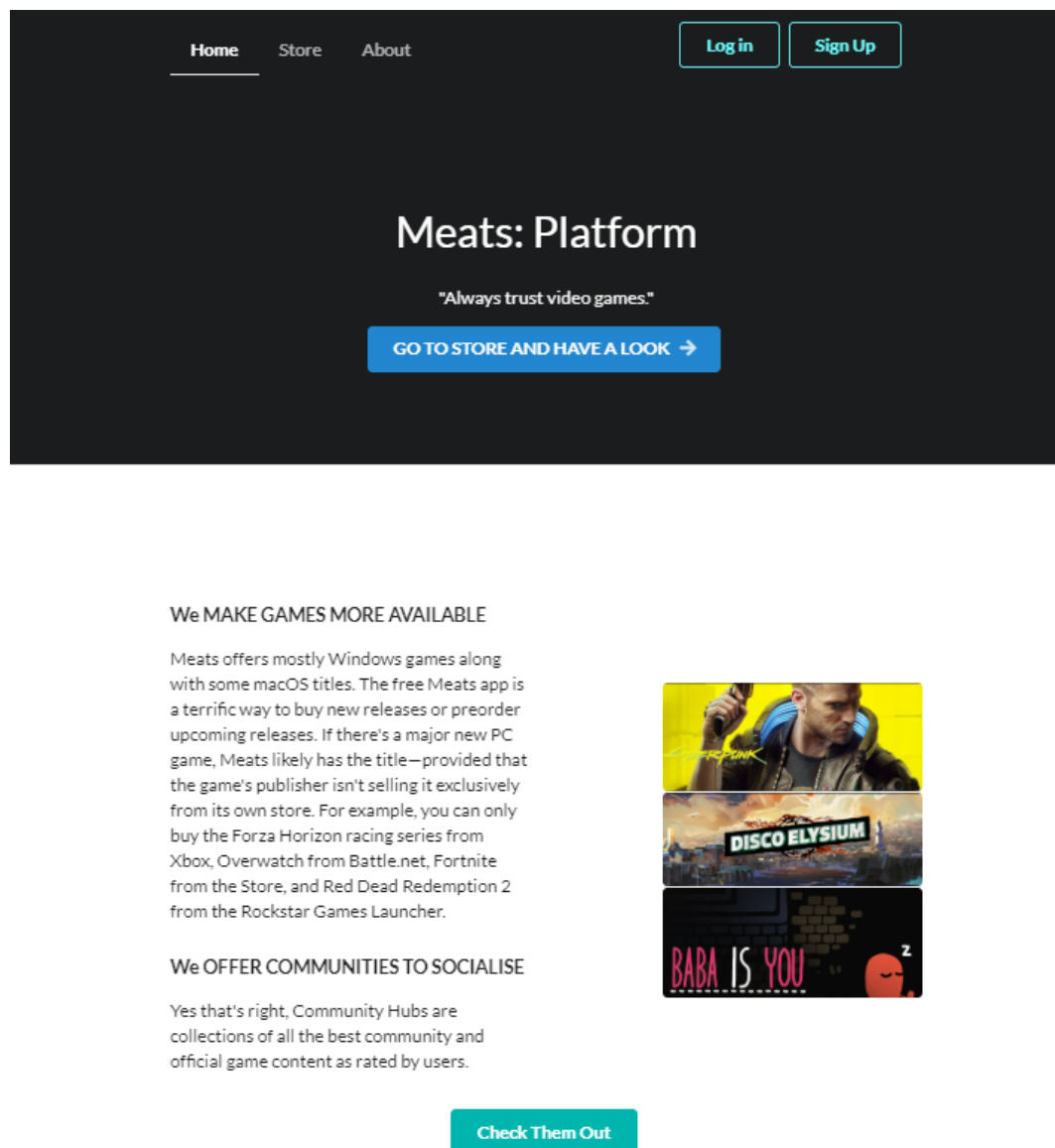
(2) 删除游戏:

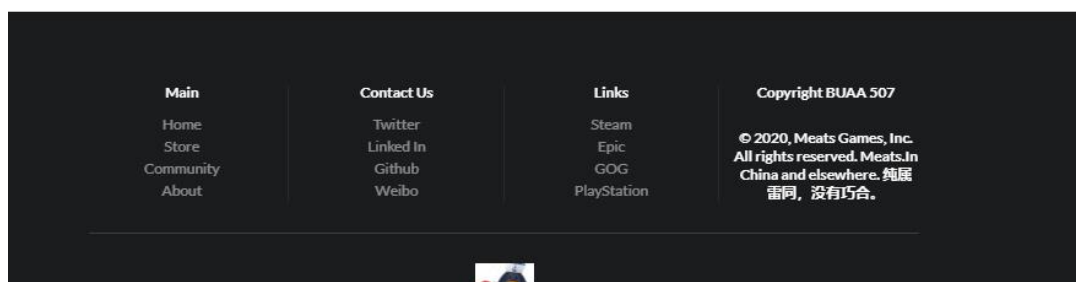
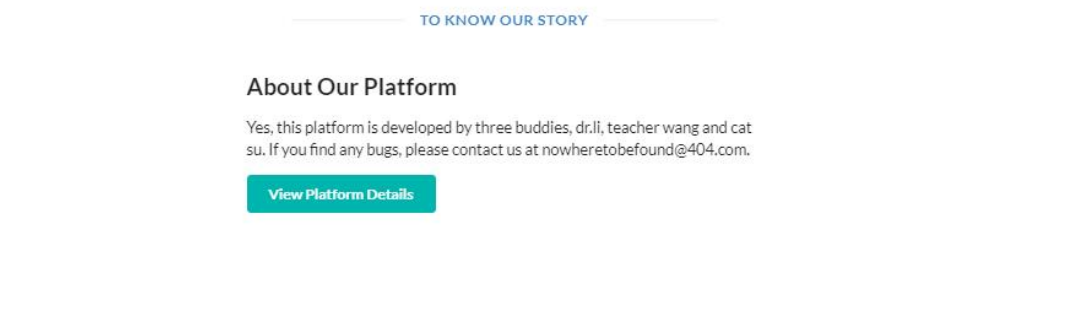
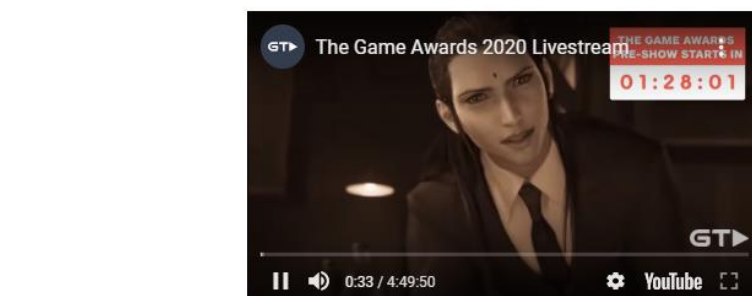
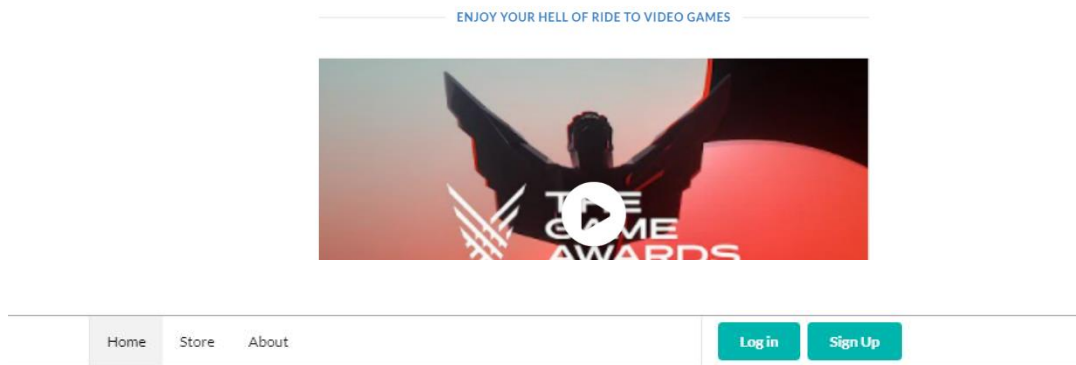
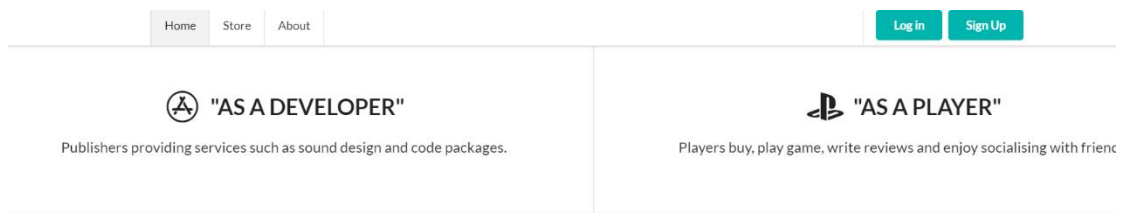
后端检查当前是否是登录状态（session user 是不是 None）未登录则返回失败 session 中 usertype 是否是 Developer 前端 Post 得到游戏名称检查 Game 列表是否有表项游戏名称、开放商元组删除表项

四、运行结果

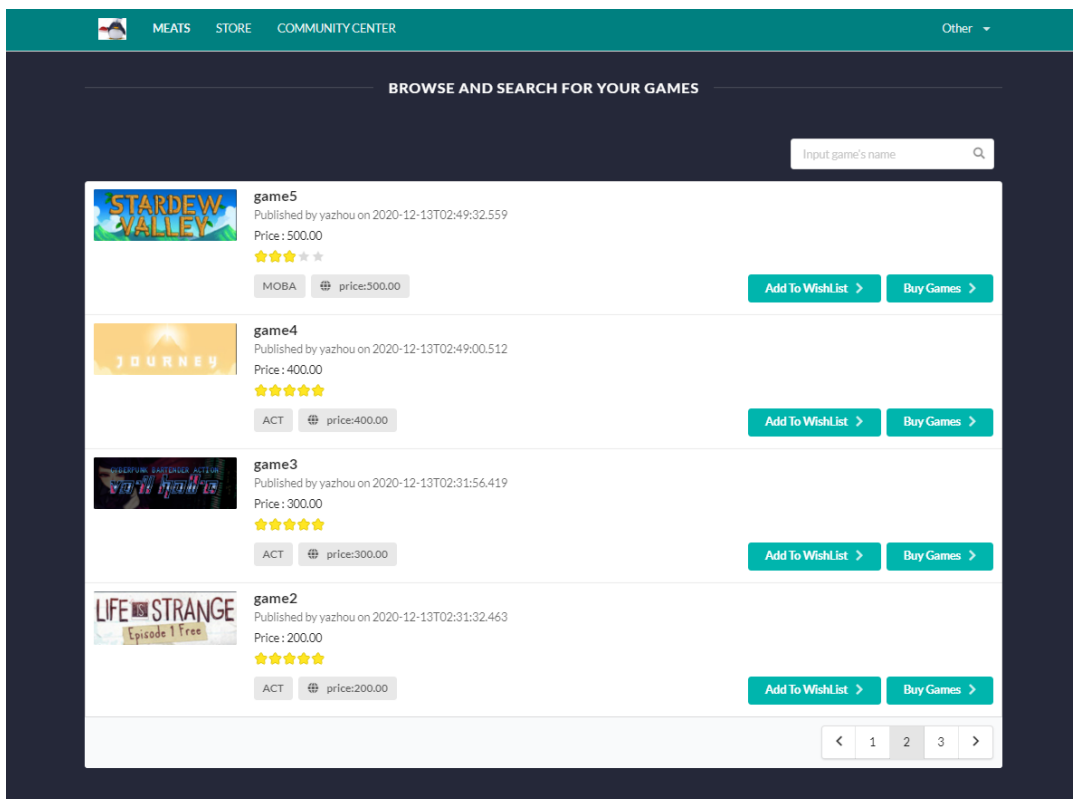
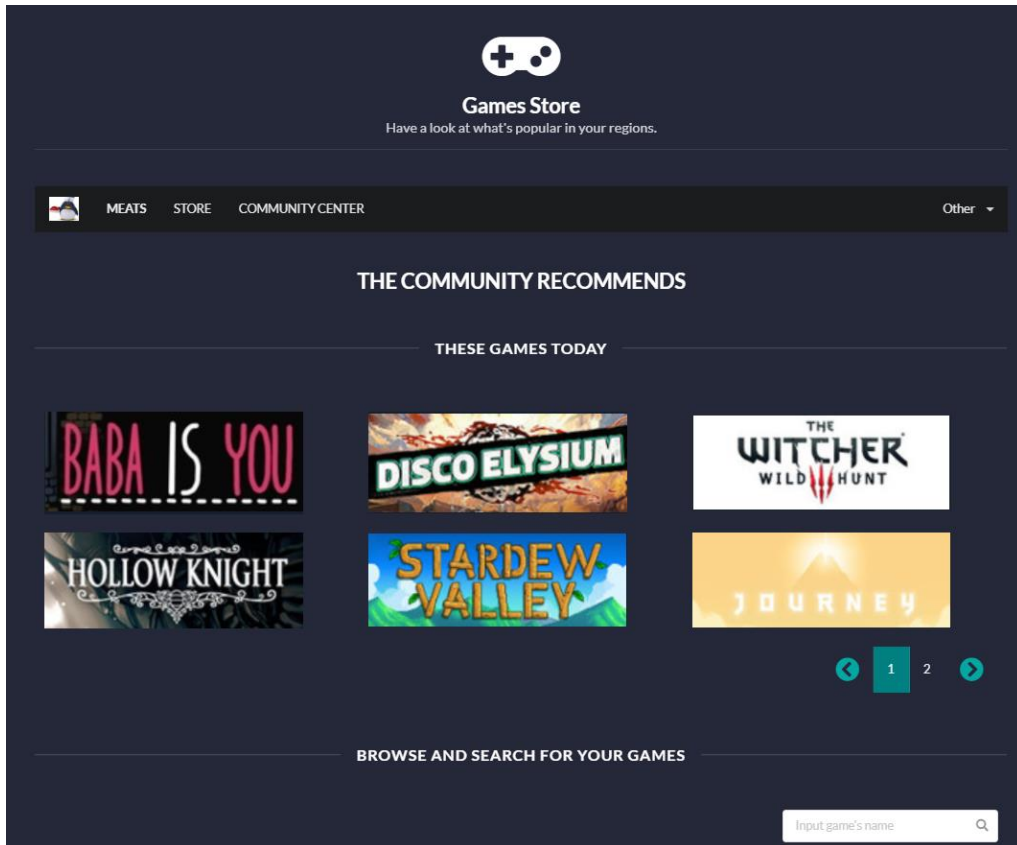
1. 简约美观，通俗大方的 UI 设计

① 主页

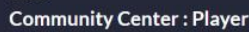




② 商店页



③ 用户中心页



Other ▾

Friend List

Friend List

WishList



MEATS
 STORE
 COMMUNITY CENTER

Other ▾

achievements, microtransactions, and support for user-created content through Steam Workshop. Though initially developed for use on Microsoft Windows operating systems, versions for macOS and Linux were later released. Mobile and Windows Phone in the 2010s. The platform also offers a small selection of other content, including soundtracks, anime, and films. The Steam platform is the largest digital distribution platform for PC market space in 2013.[2] By 2017, users purchasing games through Steam totaled roughly US\$4.3 billion in global PC game sales.[3] By 2019, the service had over 34,000 games with over 95 million monthly active users. The success of Steam led to the development of a line of Steam Machine microconsoles, which include the SteamOS operating system.

FOR DEVELOPERS

WHERE YOU PUBLISH GAMES

Publish Your Game

Category

Game name

Price 0

Image URL

Submit

111

1 Friends Friend List

MEATS
 STORE
 COMMUNITY CENTER

Other ▾

products, including in-game achievements, microtransactions, and support for user-created content through Steam Workshop. Though initially developed for use on Microsoft Windows operating systems, versions for macOS and Linux were later released. Mobile and Windows Phone in the 2010s. The platform also offers a small selection of other content, including design software, anime, and films. The Steam platform is the largest digital distribution platform for PC market space in 2013.[2] By 2017, users purchasing games through Steam totaled roughly US\$4.3 billion, representing at least 11% of the market space in 2013.[2] By 2019, the service had over 34,000 games with over 95 million monthly active users. The success of Steam led to the development of a line of Steam Machine microconsoles, which include the SteamOS operating system and Steam Controller.

FOR PLAYERS

WHERE YOU PLAY/BUY GAMES

Inventory

Wishlist

game9

Published by on 2020-12-14T00:49:20.222

Progress: 10%

★★★★☆

ACT

price:900.00

Review >

Play >

game8

Published by on 2020-12-14T00:24:19.400

Progress: 0%

★★★★☆

ACT

price:800.00

Review >

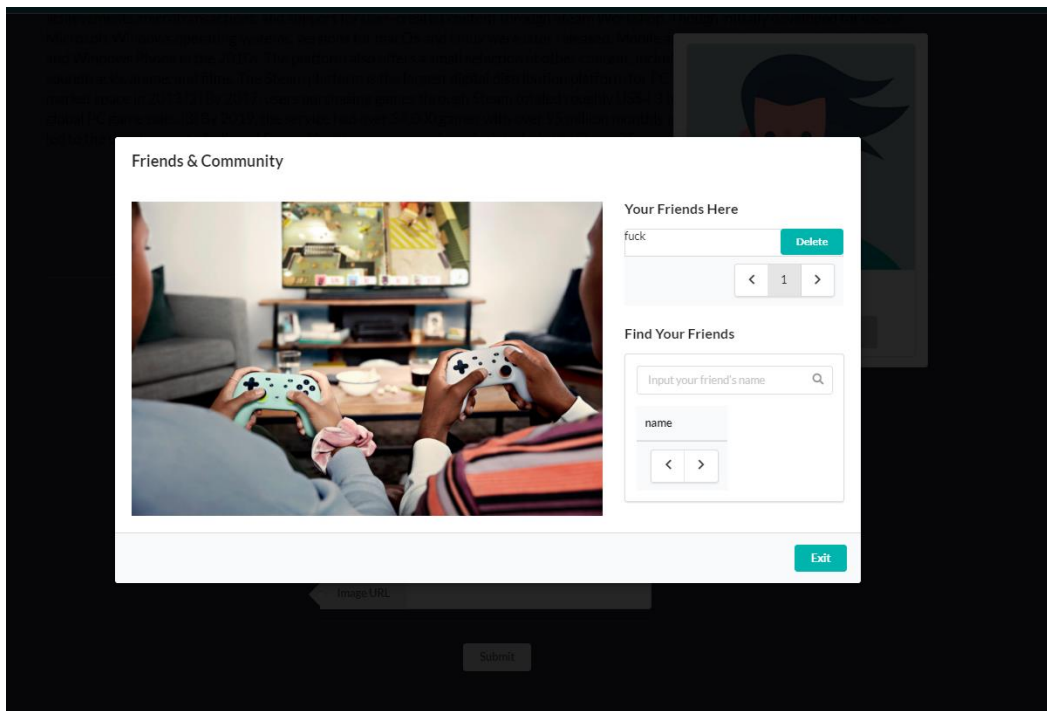
Play >

game7

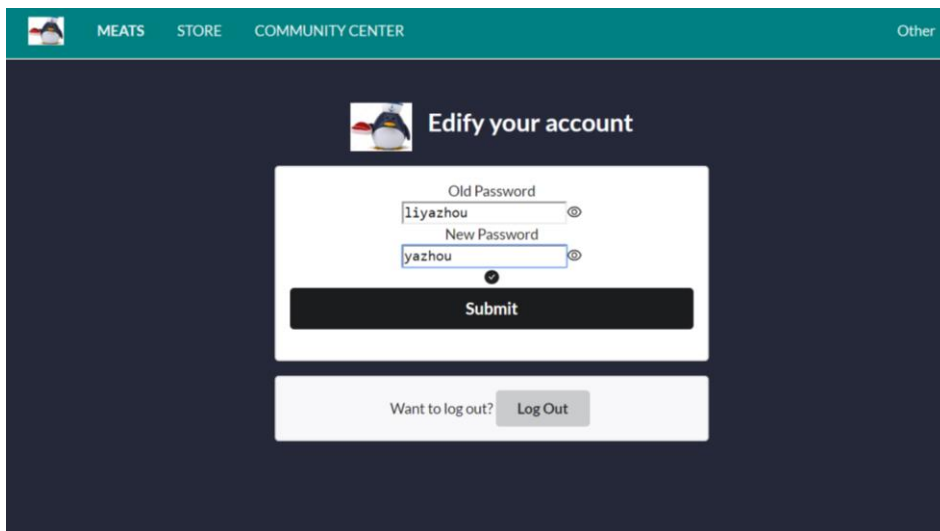
Published by on 2020-12-13T02:50:33.196


Progress: 30%

★★★★★



④登录/注册/修改信息页面



 MEATS STORE COMMUNITY CENTER Other

Register a new account

E-mail

sxy@nmd.com

User Name

sxy

Password


Confirm Password


Type

developer

Sign Up

Already have an account? [Log In](#)

 MEATS STORE COMMUNITY CENTER Other

 Log on your account

User Name

yazhou

Password

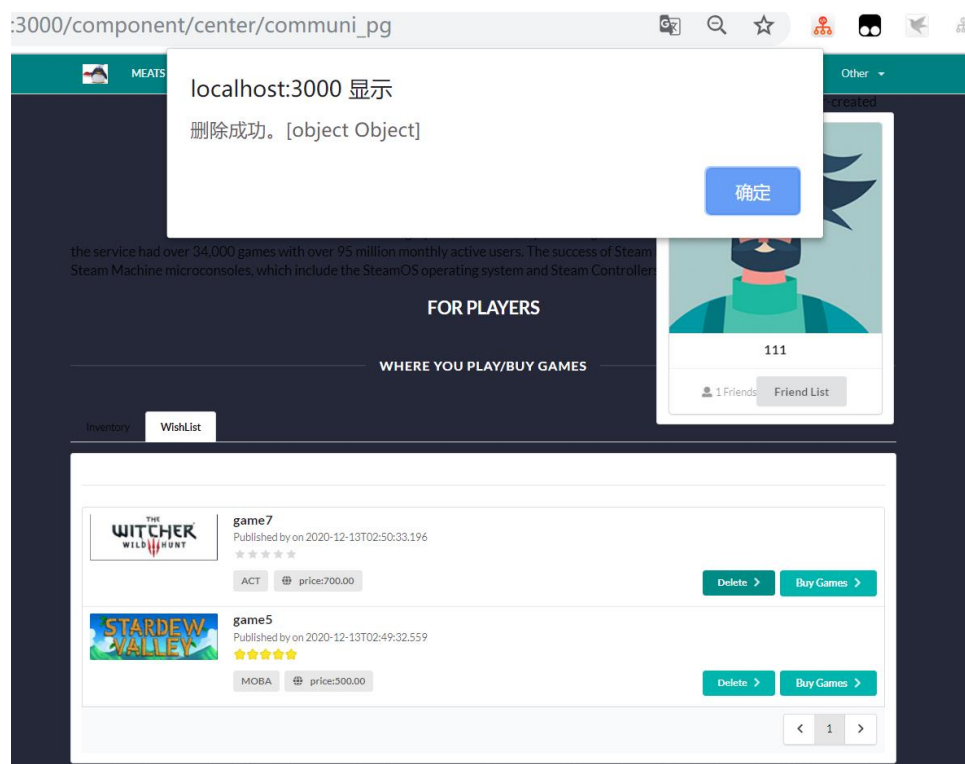
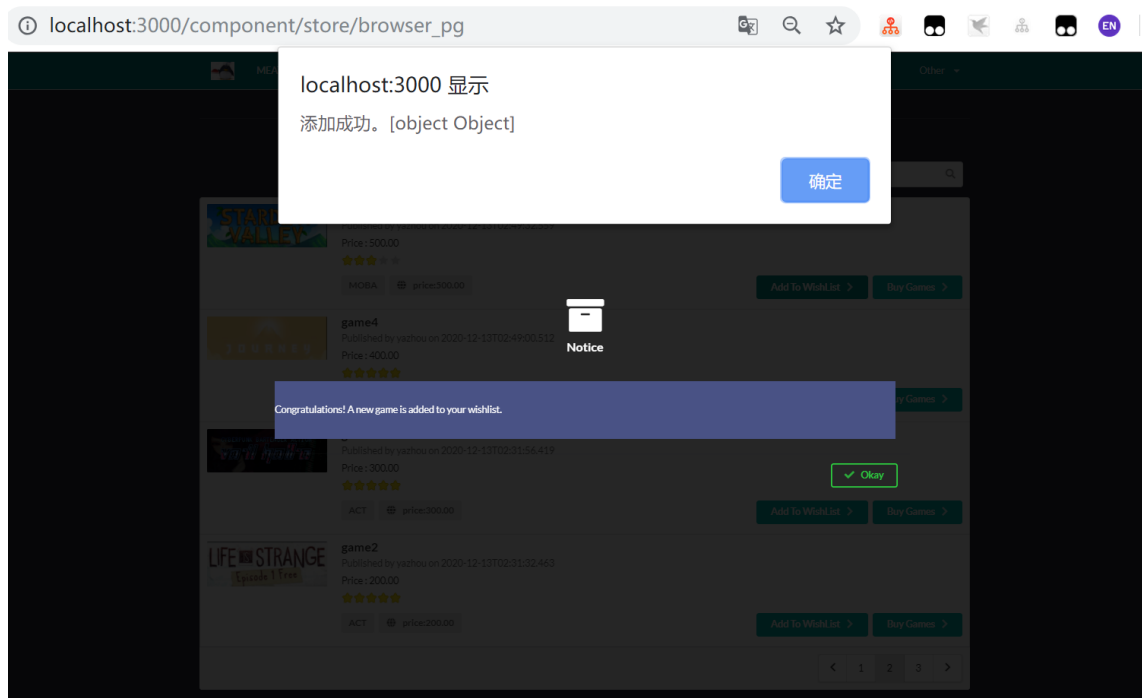
Type

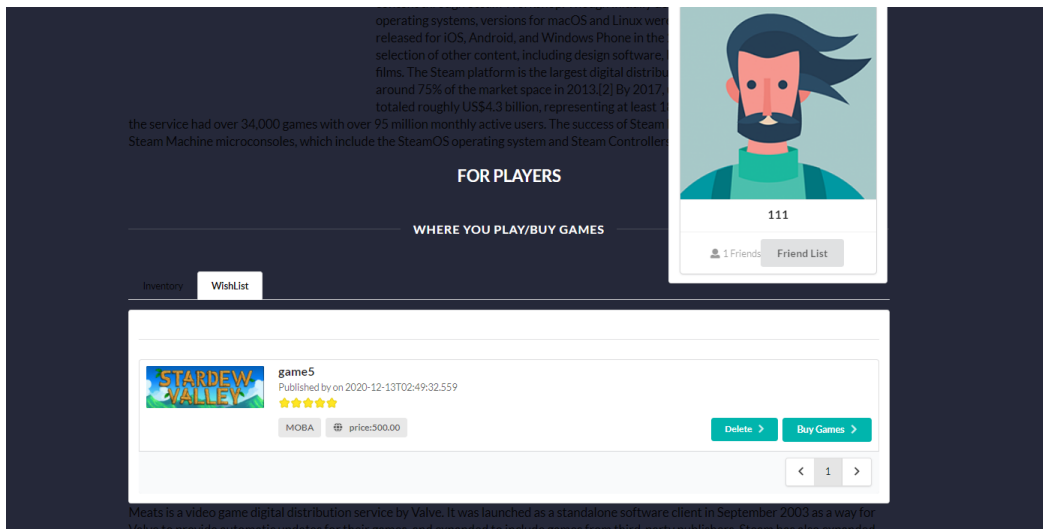
developer

Log In

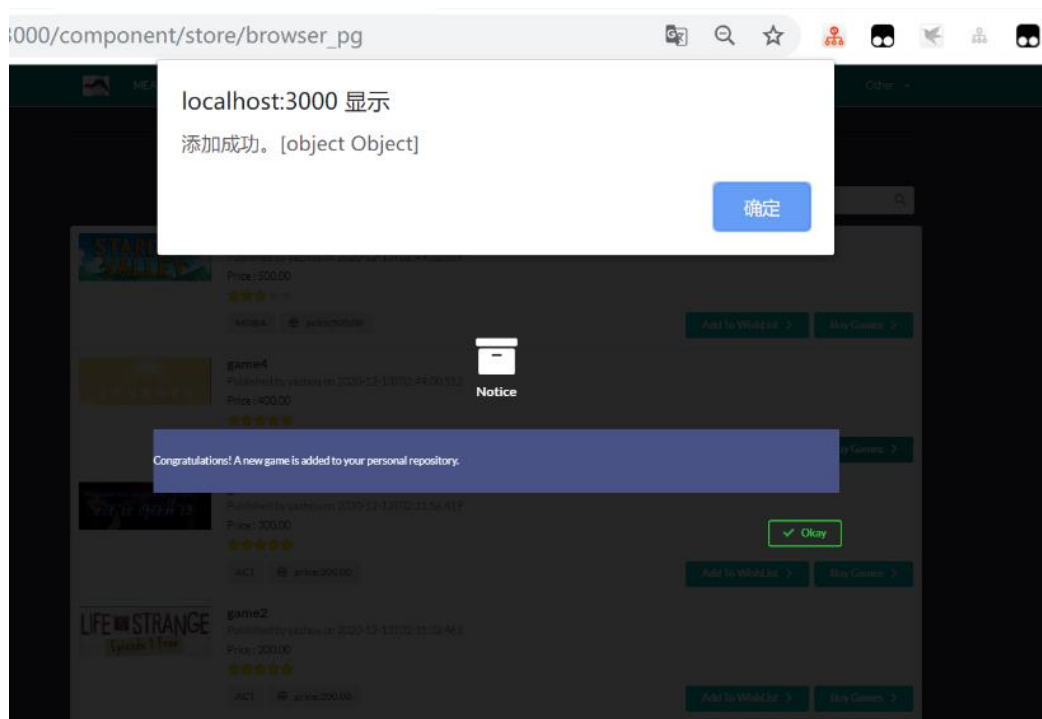
Don't have an account? [Sign Up](#)

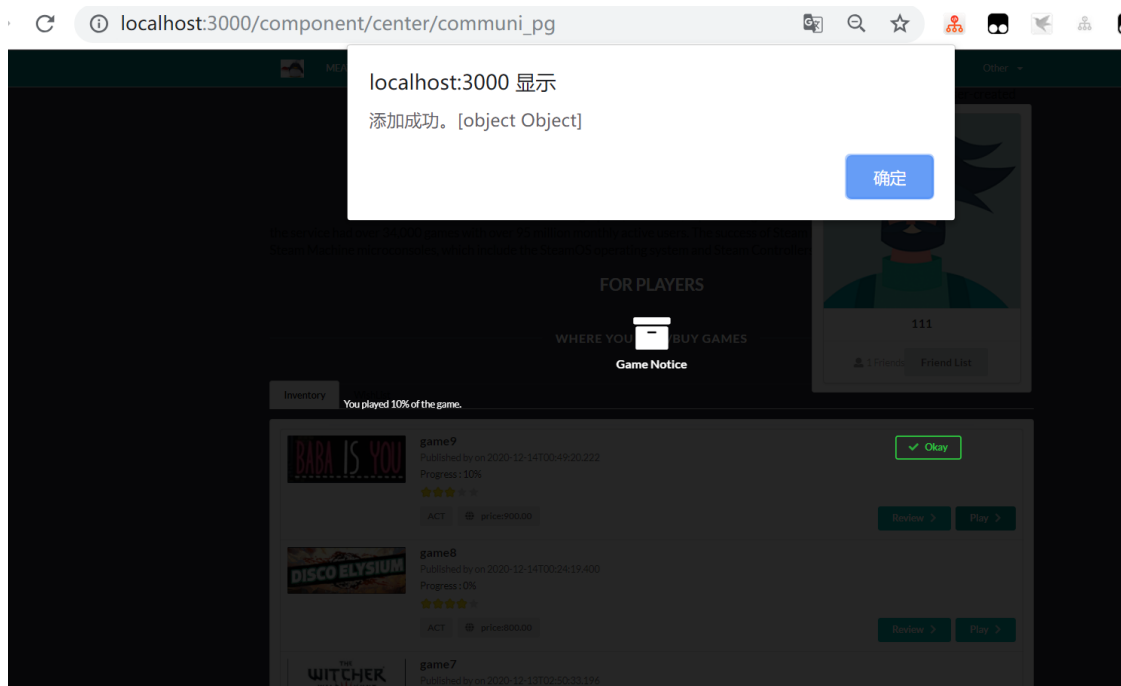
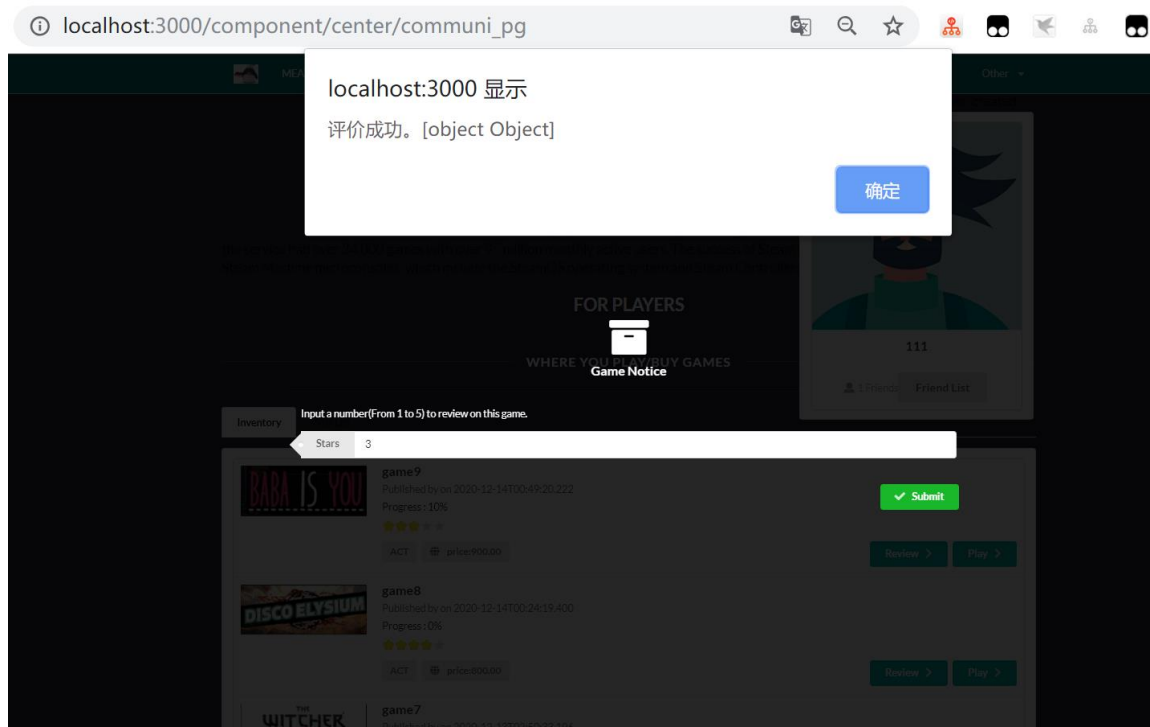
① 添加/删除心愿单 (wishlist)



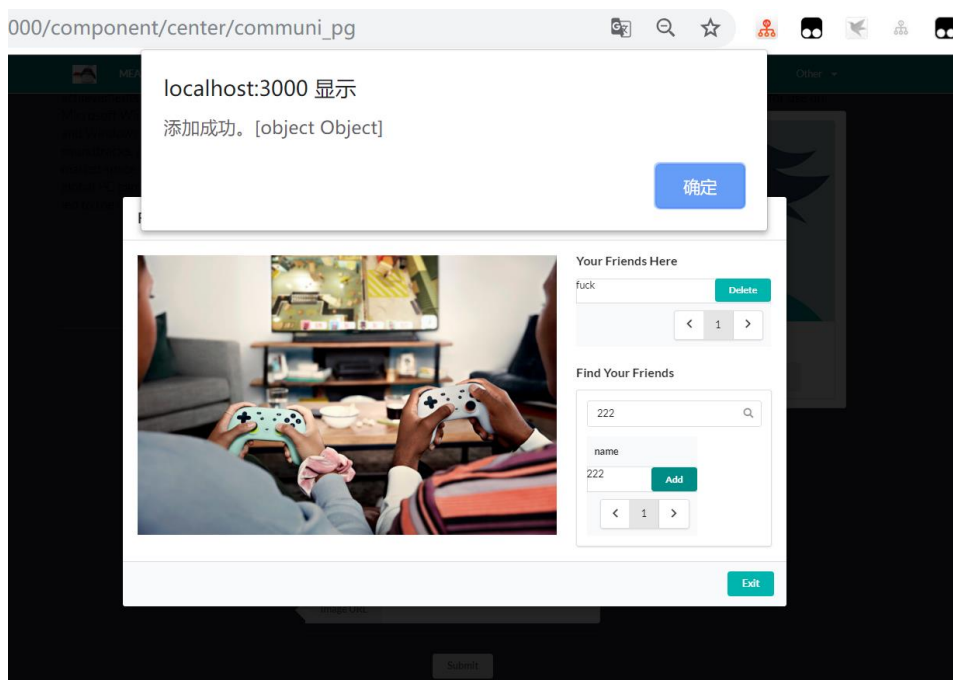
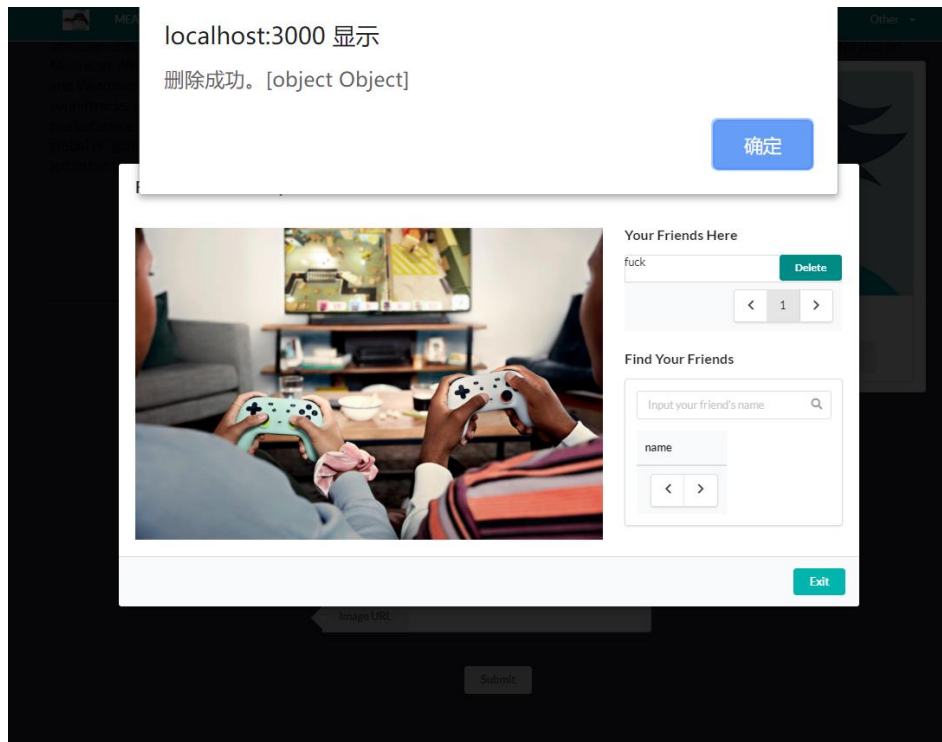


② 查找/购买/评价/玩游戏

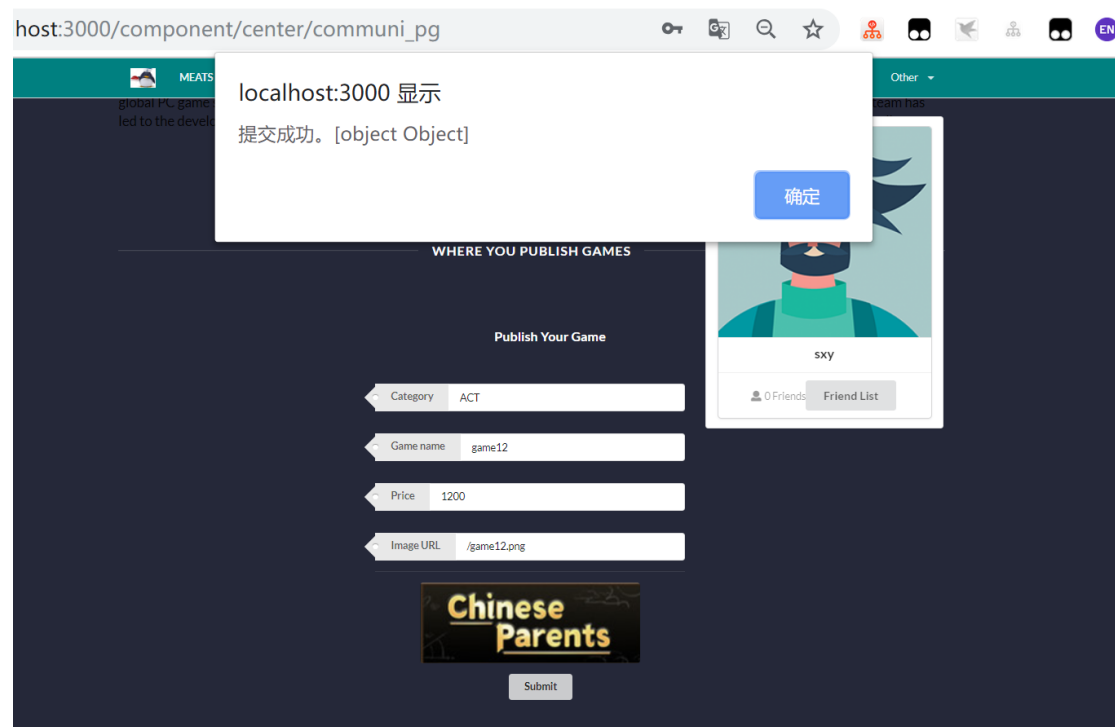
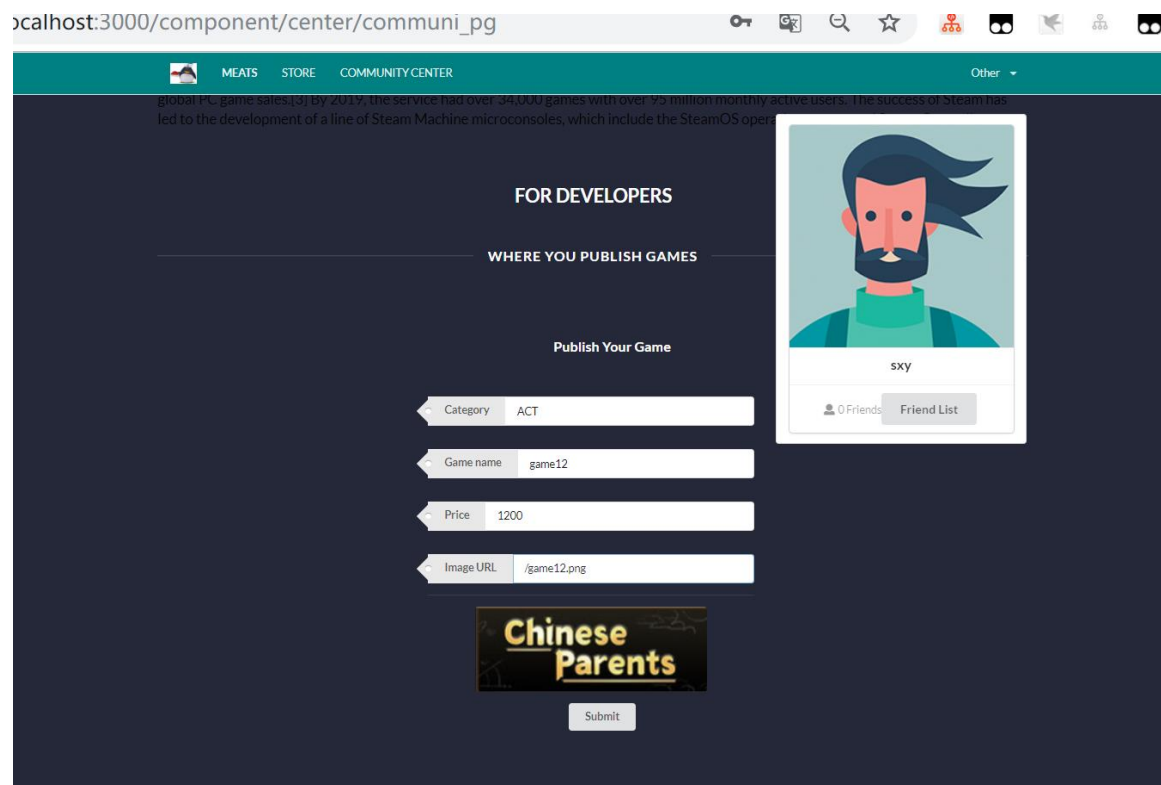




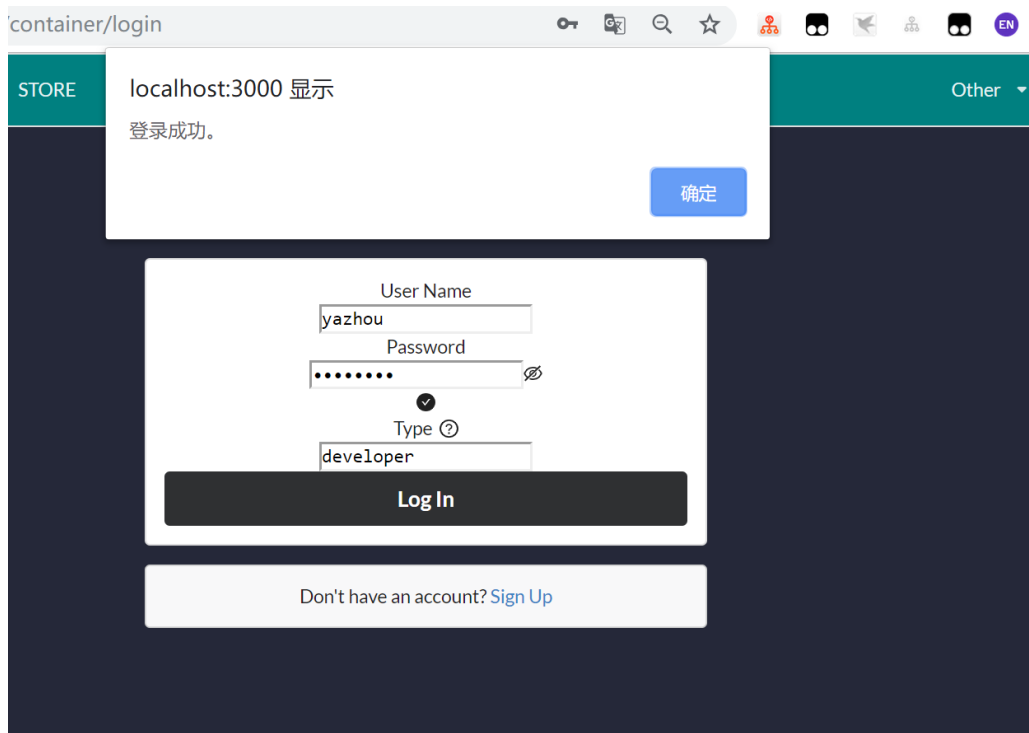
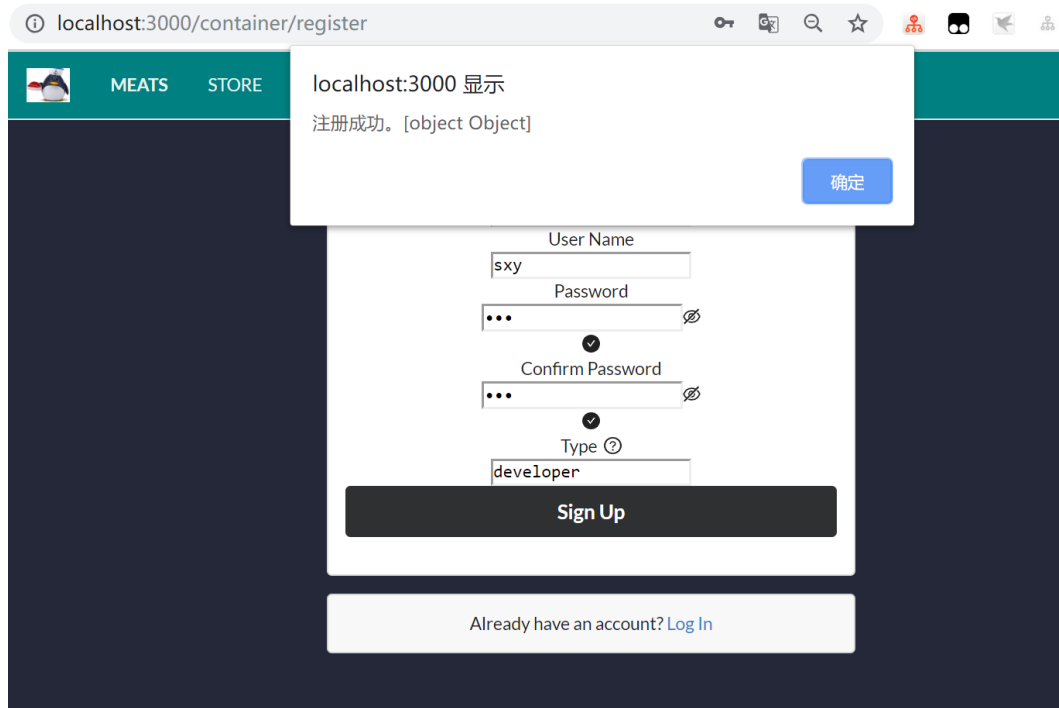
③ 添加/删除/查找好友 (friendlist)

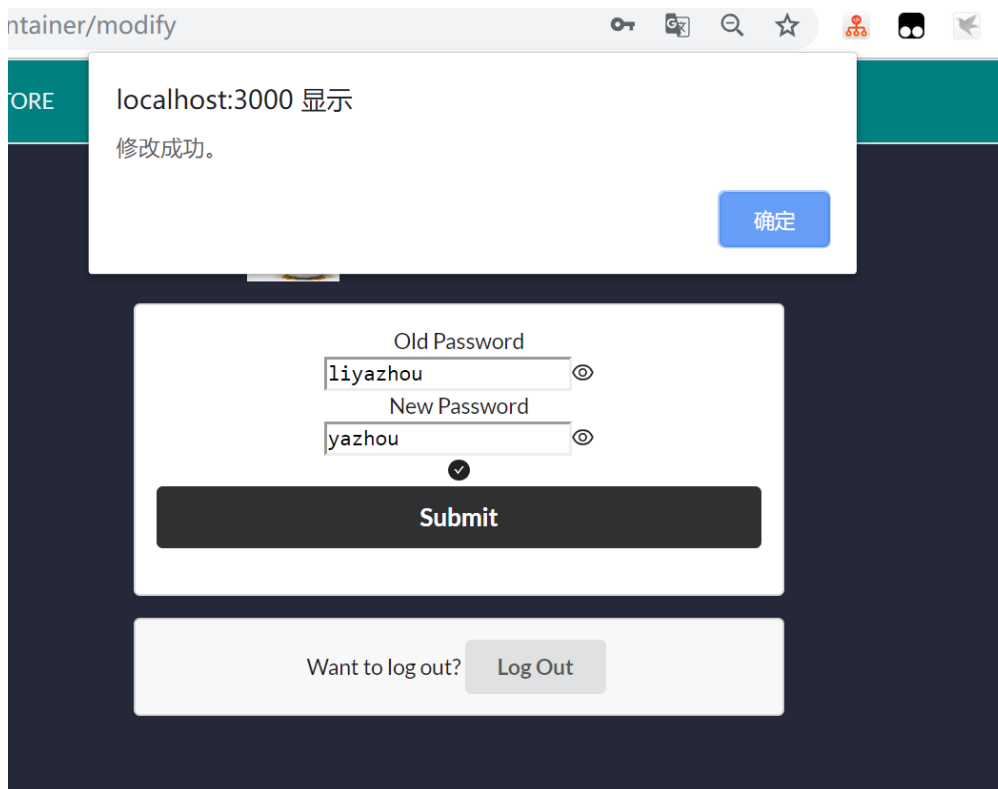
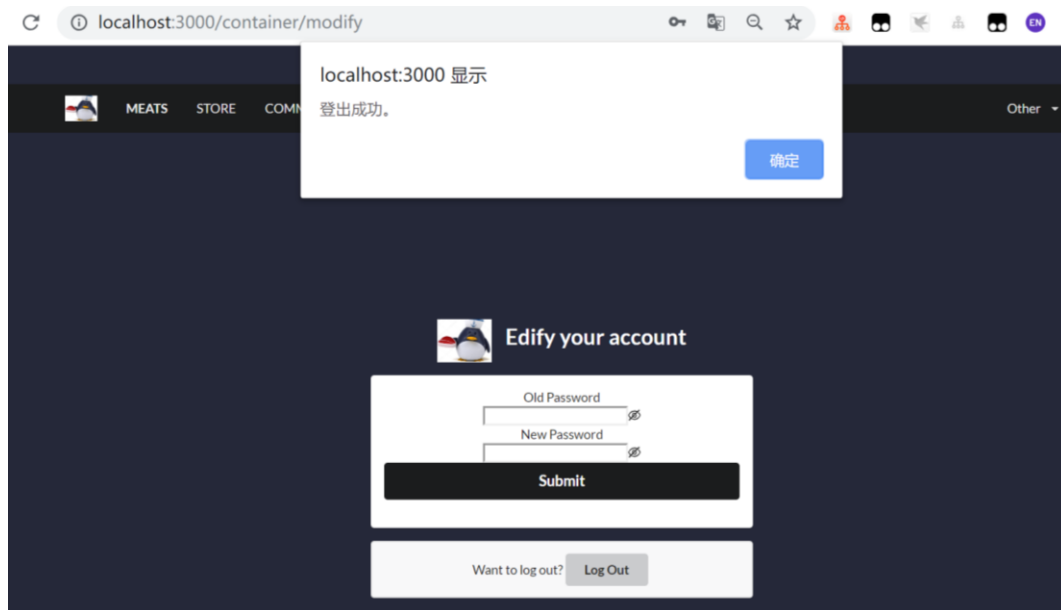


④发布游戏



⑤注册/登录/登出/修改信息





五、总结

5.1 前端苏老师之碎碎念

苏晰月主要负责前端实现和 UI 设计。我在这次的前端开发使用 Reactjs 框架，并且辅以多种 react router, react redux 等等进行高阶组件的设计，使代码的整体架构设计变得规范了不少。最后，做前端非常需要要有耐心、细心、毅力、创造力等等。做前端的也许算得上最需要耐心的编程岗位了，因为前端操控的是一个用户看得见的页面，一个一个 component 甚至的修改一些很细小的元素的经验我想很多前端都经历过，我就不多说了。**前端是很**

需要时间和经验的。

5.2 后端王老师的正经感想

王佳瑞主要负责数据库后端设计与数据库设计。在这次大作业中，后端开发使用 Django 框架，相比之前小作业使用 java 开发的体验，Django 框架集成数据库操作、提供成熟的类比如 User，使后端开发便捷迅速了许多。在数据库设计部分，**在实践中运用到理论课学习的查询删除等基本操作**，与前端实际交互中，根据前端实际需求适时调整后端提供的各种服务，也逐渐根据前端需要优化数据库设计。这次作业让我对后端开发有了进一步认识，在与前端联调的过程中，对前后端交互方式、session 保存机制有了进一步认识。

5.3 全栈栗老师也有话要说

栗亚舟主要负责前后端的交互，对前后端的工作机制都有了解，如 Django 框架的工程结构、mysql 数据库的生成，semantic-ui-react 框架好看又方便的组件。在整个连接前后端的过程中，**我对前后端互相发送数据及其 debug 方式有了深刻的理解**，处理了很多如组件的生命周期、数据的异步处理之类的问题，同时前端操作数据需要用到大量 JavaScript 语言。