讯飞头条

概

要

设

计

|  |  |
| --- | --- |
| **项目名称** | 讯飞头条 |
| **文档撰写** | 潮家雯 |
| **建立日期** | 2022年5月24日 |

目录

[1 概述 3](#_Toc64448110)

[2 系统架构图 3](#_Toc64448111)

[3 系统逻辑架构图 4](#_Toc64448112)

[4 物理拓扑图 4](#_Toc64448113)

[5 部署结构图 5](#_Toc64448114)

[6 关键技术 5](#_Toc64448115)

[6.1 Echarts图可视化展示 7](#_Toc64448116)

[6.2 分页与搜索 9](#_Toc64448117)

[6.3 后台管理的登录拦截 10](#_Toc64448118)

[7 关键设计模块 11](#_Toc64448119)

[7.1 部分功能界面 11](#_Toc64448120)

[7.1.1 首页展示 11](#_Toc64448121)

[7.1.2 推荐商品展示 11](#_Toc64448122)

[7.1.3 可视化数据展示 12](#_Toc64448123)

[7.2 数据库设计 12](#_Toc64448124)

[7.2.1 管理员表 12](#_Toc64448125)

[7.2.2 商品表 13](#_Toc64448126)

[7.2.3 商品类型表 13](#_Toc64448127)

[7.2.4 商品评论表 14](#_Toc64448128)

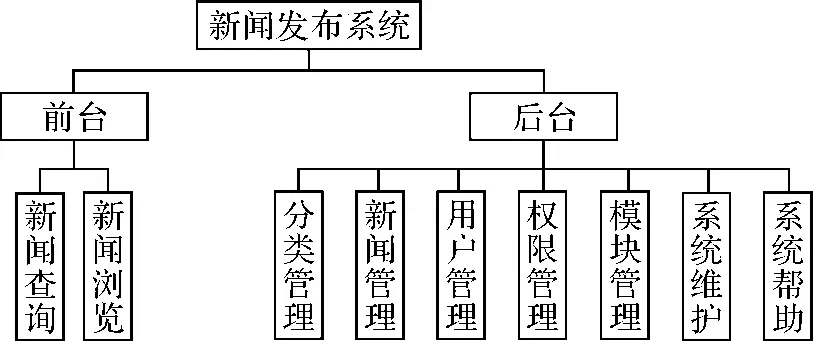
[7.3 关键功能模块 14](#_Toc64448129)

[7.4 业务逻辑 14](#_Toc64448130)

# 概述

“讯飞头条”主要是一款基于react框架的团队开发项目，主要用户浏览获取平台的一些新闻详情以及可视化的数据展示。从而引导用户对世界有更清晰的认知。

# 系统功能模块图



# 用例图

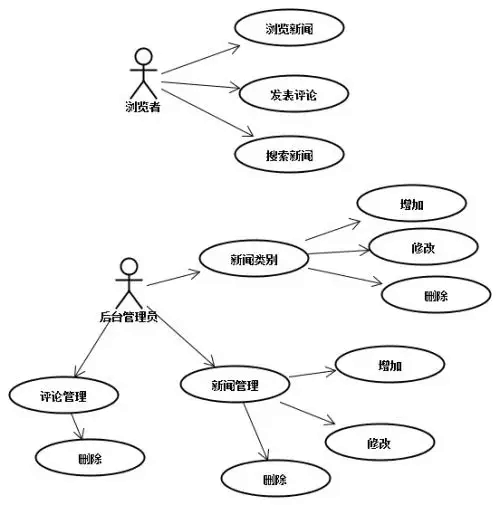


图4-1

# 关键技术

本项目主要使用react框架的团队开发项目

**React框架的特点主要为：**

1. 组件化，增强复用性
2. 数据到视图单向绑定，更安全
3. 数据的单向传递流
4. 高性能

**React框架的优点主要为：**

1.

声明式:我们只需要描述UI,而react负责渲染UI, 当数据变动时 React 能高效更新并渲染合适的组件。

2.

基于组件:组件表示页面中的部分内容。 构建管理自身状态的封装组件,然后对其组合以构成复杂的 UI。

3.

高效、灵活:react可以做很多东西;例如:我们使用react可以开发web应用;还可以通过react-native开发移动端应用;甚至可以开发VR应用

代码展示

|  |
| --- |
| import { BrowserRouter, Route, Routes } from 'react-router-dom';  import './App.css';  import Home from './homepage/Home';  import Technology from './homepage/Technology'  import Game from './homepage/Game'  import Entertainment from './homepage/Entertainment'  import Flash from './homepage/Flash'  import Detail from './homepage/details/Detail'  import React, { Component } from 'react'  import Login from './login/Login';  import SignUp from './login/SignUp';  import Admin from "./login/Admin";  import UserIndex from "./login/UserIndex";  import ShowSearch from './homepage/ShowSearch';  export default class App extends Component {    render() {      return (        <div>          <BrowserRouter>            <Routes>              <Route path='/login' element={<Login />} />              <Route path='/register' element={<SignUp />} />              <Route path='/flash' element={<Flash />} />              <Route path='/entertainment' element={<Entertainment />} />              <Route path='/technology' element={<Technology />} />              <Route path='/game' element={<Game />} />              <Route path="/detail/:id" element={<Detail />} />              <Route path='/showsearch' element={<ShowSearch/>}/>              <Route path="/UserIndex" element={<UserIndex />} />              <Route path="/SignUp" element={<SignUp />} />              {/\* <Route path="/Login" element={<Login />} /> \*/}              <Route path="/Admin" element={<Admin />} />              <Route index element={<Home />} />            </Routes>          </BrowserRouter>        </div>      )    }  } |

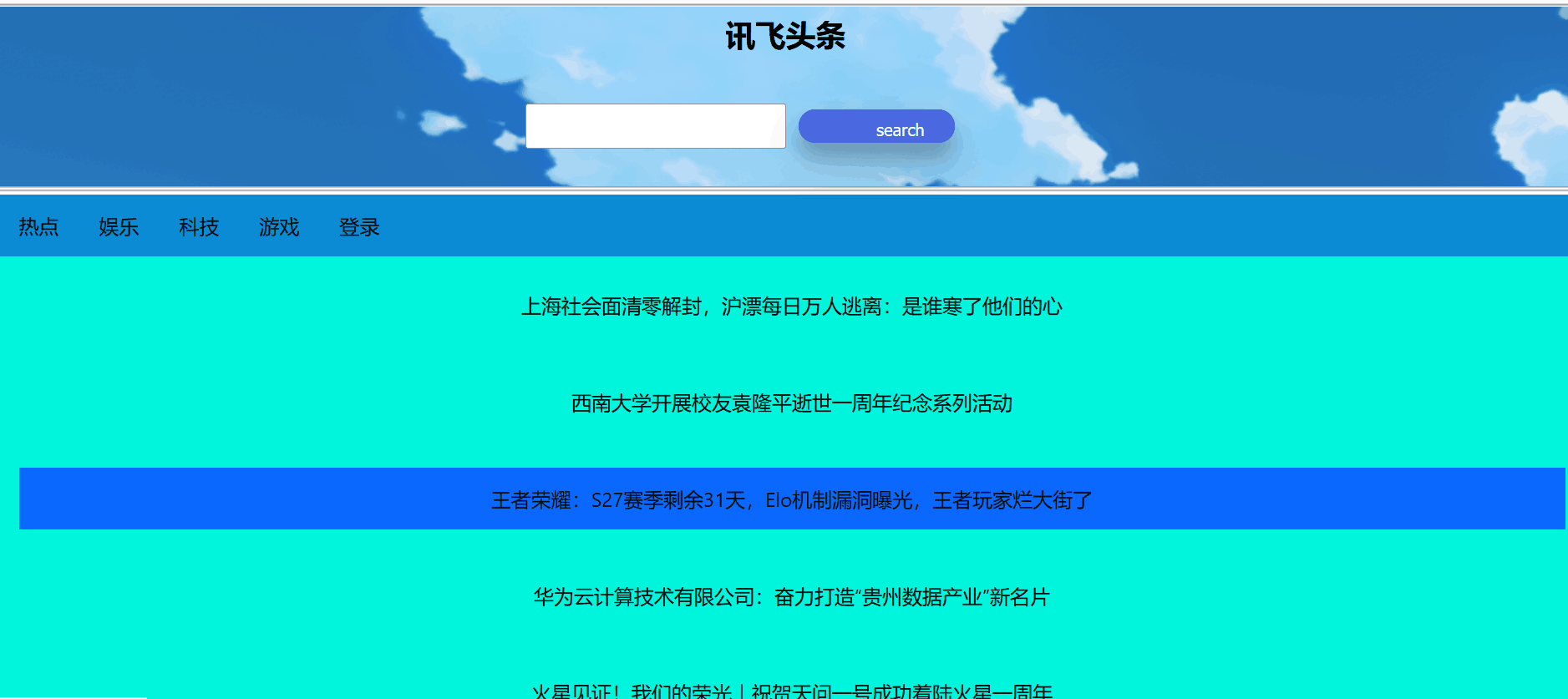
## 后台管理的代码展示

|  |
| --- |
| /render() {      return (        <div style={style.Admin} className="maindiv">          <img src={img} alt="admin.gif" className="img" />          <div className="buttondiv">            <button              className="AddUser"              onClick={() => this.setState({ NowClick: "AddUser" })}>              AddUser            </button>            <button              className="DeleteUserByUserName"              onClick={() => this.setState({ NowClick: "DeleteUserByUserName" })}>              DeleteUserByUserName            </button>            <button              className="UpdateUser"              onClick={() => this.setState({ NowClick: "UpdateUser" })}>              UpdateUser            </button>            <button              className="GetUserByUserName"              onClick={() => this.setState({ NowClick: "GetUserByUserName" })}>              GetUserByUserName            </button>            <button              className="GetAllUser"              onClick={() => this.setState({ NowClick: "GetAllUser" })}>              GetAllUser            </button>          </div>          <div className="buttondiv">            <button              className="AddNews"              onClick={() => this.setState({ NowClick: "AddNews" })}>              AddNews            </button>            <button              className="DeleteNewsById"              onClick={() => this.setState({ NowClick: "DeleteNewsById" })}>              DeleteNewsById            </button>            <button              className="UpdateNews"              onClick={() => this.setState({ NowClick: "UpdateNews" })}>              UpdateNews            </button>            <button              className="GetNewsById"              onClick={() => this.setState({ NowClick: "GetNewsById" })}>              GetNewsById            </button>            <button              className="GetAllNews"              onClick={() => this.setState({ NowClick: "GetAllNews" })}>              GetAllNews            </button>            <div className="buttondiv">            <a href = "http://localhost:3000">              <button>返回首页</button>              </a>            </div>          </div>          <AdminList data={this.state.NowClick} />        </div>      );    }  } |

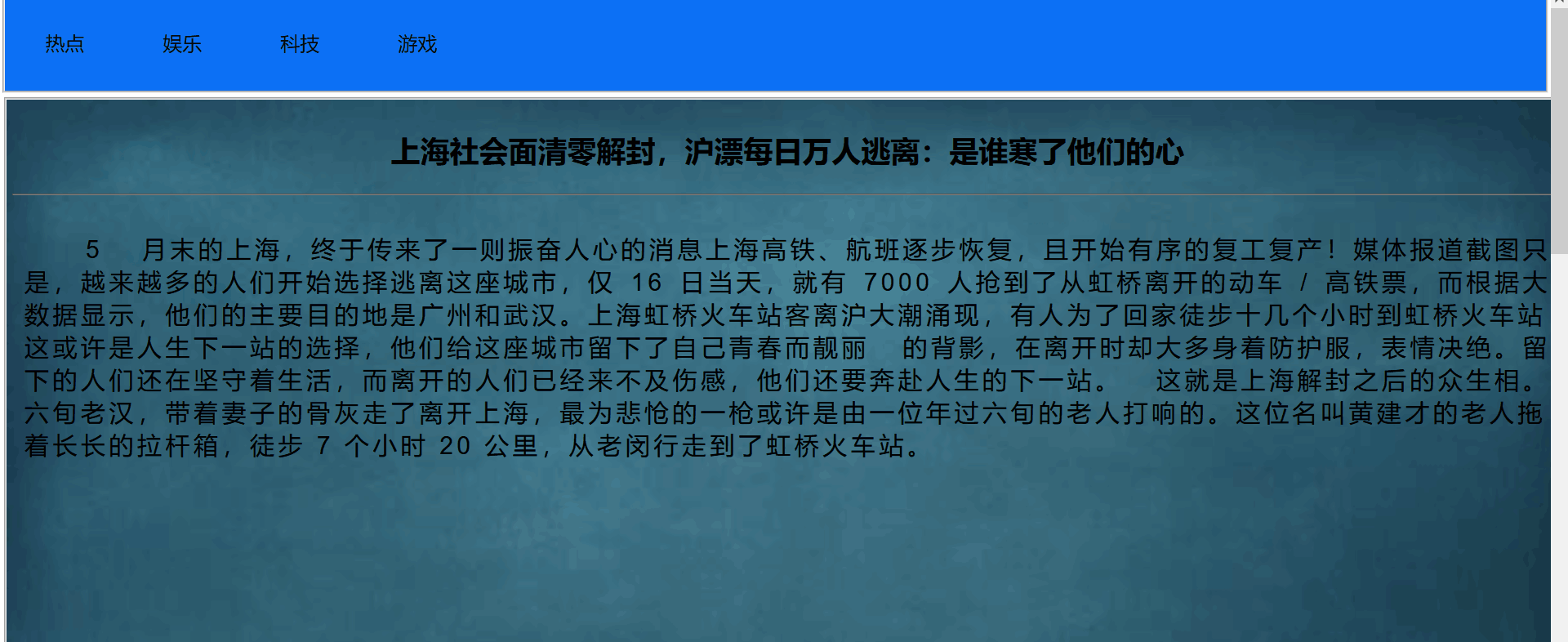
# 关键设计模块

## 部分功能界面

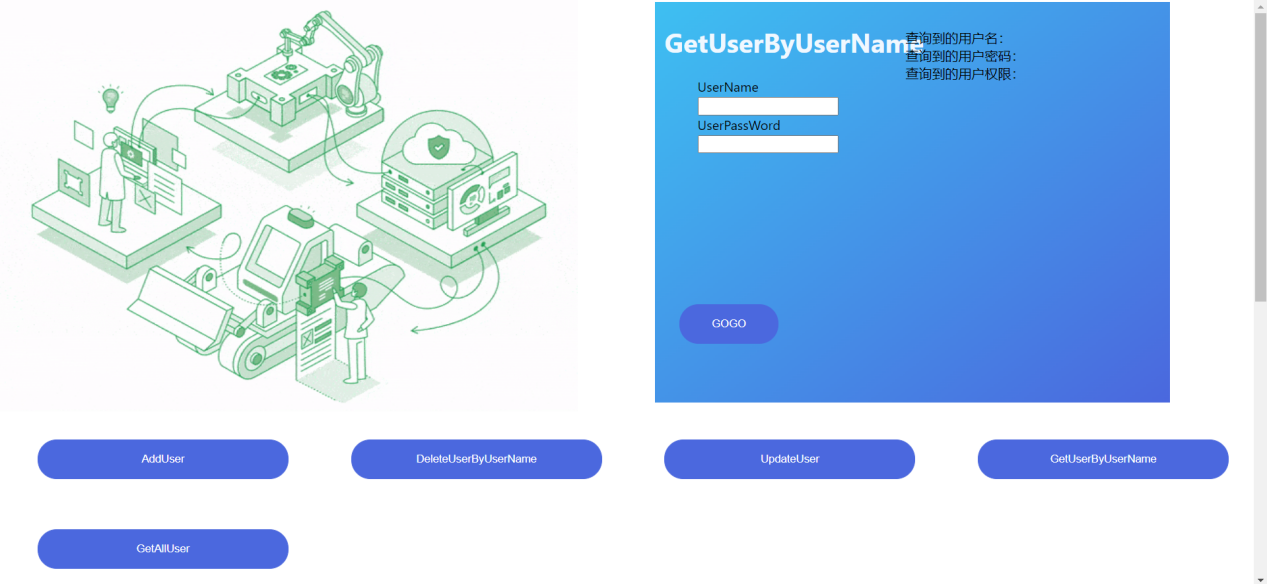
### 首页展示



### 新闻内容展示



### 管理员用户界面



## 关键功能模块

* 普通用户管理：用户通过点击进入新闻，新闻详情，后台管理。
* 管理员管理：管理用户，可以对用户以及新闻进行增删改查
* 信息管理：管理员和普通用户均可通过页面进行查看，管理员可以管理新闻，以及其他用户的评论。
* 任务管理：管理员可以对每日的新闻进行增删改的操作，定期查看方便对新闻的管理。

## 业务逻辑

* 普通用户：可以通过主页面浏览每日新闻，并对其进行评价
* 新闻：新闻主要包括增删改查，用户对新闻的评论，新闻的一些内容管理。新闻跟新闻评论是一对多的关系。
* 管理员：管理员通过对每日新闻的内容管理，通过评论的方式，较为醒目。其中新闻与新闻评论是一对多的关系。并且管理员可以对新闻以及新闻的评论进行增删改查。