

目录

摘要	1
Abstract	2
1 绪论	3
1.1 选题的背景及意义	3
1.2 主要工作及研究内容	3
1.3 本文的组织结构	4
2 相关工具和技术	7
2.1 SQL Server 数据库介绍	7
2.2 Node JS 环境介绍	7
2.3 VUE 前端技术	7
2.4 KOA 框架	8
3 需求分析	9
3.1 可行性分析	9
3.1.1 技术可行性	9
3.1.2 社会可行性	9
3.2 系统总体需求	10
3.3 系统功能需求	10
3.3.1 用户模块	10
3.3.2 生活记录模块	11
3.4 系统性能需求	11

4 系统设计	13
4.1 体系结构设计	13
4.2 主要功能设计	13
4.3 数据库设计	16
5 系统实现	17
5.1 主要功能页面实现	17
5.1.1 主页模块	17
5.1.2 查看模块	19
5.1.3 添加模块	20
5.1.4 详细模块	21
5.1.5 用户模块	22
5.2 服务相关接口实现	22
6 系统测试	25
6.1 功能性测试	25
6.2 兼容性测试	30
结论	33
参考文献	35

基于定位的生活记录 Web 应用的设计与实现

摘要： 在现在信息化的社会中，每个人的生活本身就时时刻刻都是一种信息资源。本文以时间、地点来规范记录生活的某一次事件信息，以方便对生活记录的管理，帮助人们去记录或回想自己某一刻的感受或想法。本文使用 **Visual Studio Code** 编写代码，数据库使用的是 **MySQL**，运用 **VUE** 和 **Node** 分别实现前端界面和后端服务。本文除介绍相关工具技术外，还阐述了系统需求，可行性的分析，相关功能的设计实现，系统测试等项目流程。本文中的 **Web** 应用利用百度地图的 **API** 在每次记录生活信息的同时自动记录当时的时间地点，然后将这些信息储存起来。用户可以登录后查看自己的生活记录，可以通过搜索，排序很快的查找到需要的记录；也可以在百度地图上通过地图标点来查看某天某地的记录；除查看自己的记录外，也能查看到其他人公开的生活记录；可以对生活记录进行相关的收藏，修改，删除等。整个 **Web** 应用功能完全、界面友好。

关键词： 生活记录；Web 应用；百度地图；VUE

Design and Implementation of Life Record Web Application Based on Location

Abstract: In today's information society, everyone's life itself is an information resource all the time. This article uses time and place to standardize and record the information of a certain event in life, so as to facilitate the management of life records and help people to record or recall their feelings or thoughts at a certain moment. This article uses Visual Studio Code to write code, the database uses MySQL, and uses VUE and Node to implement front-end interface and back-end services respectively. In addition to introducing related tools and technologies, this paper also expounds the system requirements, feasibility analysis, design and implementation of related functions, system testing and other project processes. The web application in this article uses the API of Baidu Maps to automatically record the time and place at that time while recording life information, and then store the information. Users can log in to view their own life records, and can quickly find the records they need by searching and sorting; they can also view the records of a certain day and place on the Baidu map through map points; in addition to viewing their own records, they can also You can view the life records published by other people; you can collect, modify, delete and so on related life records. The entire Web application is fully functional and user-friendly.

Key words: Life records; Web applications; Baidu Maps; VUE

1 绪论

1.1 选题的背景及意义

在信息化时代的如今，生活中越来越多的事情已经向数字化，智能化转变。生活中的一切都是可以被关注，可以被利用的信息，而将这些信息进行收集记录，分析使用已经成为趋势。从纸币到现在的微信，支付宝；从以前的排队，出行到现在的足不出户也可以满足一个人的生活起居；现今的生活会发现，社会的种种已经离不开信息化，大数据的支持。一个人的一生经历似乎已经可以被信息化，以数字的形式记录下来，在人的生活记录方面，从以前古人伟人的传记，到当代人人可以写的日记，备忘录，这些都是一个人生活记录的记载形式。而信息化社会建设发展至今，已经有了形形色色的个人生活记录方式，像 QQ，微信，微博，已经系统的记录了许多用户的生活信息，还有几乎现在所有手机都自带备忘录，记录着每个人的生活；以及现在的抖音最开始也是记录美好生活的初衷。

1.2 主要工作及研究内容

本应用从设计方向来考虑，以记录为主要目的，以及尽可能确保信息的准确性，真实性；记录的内容以文本，图片，视频为主，围绕记录的信息来进行扩充，规范，设计和实现，以及考虑 Web 应用^[1]的适用性。主要使用 VUE 的前端框架来实现主要信息的交互管理，数据库和服务器对主要信息的储存和简单处理。围绕主要信息分为两个方面：一是信息的获取；二是已有信息的管理。为尽可能确保 Web 获取信息的准确和真实，这里采取百度地图的 API，和 Web 的时间^[2]，

从隐私的角度，也直接将信息由用户分为开放与私密两类；而对于信息的管理，支持以多种方式查看信息，以及对登录用户信息的修改，对已存信息的标记区分。用户信息进行简单的设计，主要以信息为主，确保用户可以通过用户账号找到对应的每一条记录的信息。

本 Web 应用，先是对于数据库进行一个设计，主要有两个表：用户信息表和记录信息表。然后完成服务器的搭建，数据接口的设计，包含用户注册，修改，登录的接口；记录信息存入，查询，修改的接口。其中对于图片，视频等文件的保存进行了额外的接口处理，还有对密码的存储进行了加密处理，以及使用 **token** 保存登录状态^[3]；再是 Web 应用的前端设计，主要有首页，添加页，查看页，个人页，推荐页五个页面，同时完成页面路由的配置，移动端适配的设置，以及后端接口请求的封装，相关依赖的引入使用。最终通过前后端的合理交互^[3]，实现了对记录信息的有效管理，从而可以有效的发挥这些信息的作用。

1.3 本文的组织结构

根据已有的技术和相关需求，设计并基本实现了基于定位的生活记录 Web 应用，本文共分为六个章节：

第一章绪论：主要说明本文选题的背景和意义，现今社会相关领域中的研究状况，主要工作及研究内容等；

第二章相关工具和技术：主要说明实现本设计将采用的主要技术，包括 SQL Server、Node JS、VUE 框架、KOA web 框架等；

第三章需求分析：主要说明本设计的可行性分析，相关功能需求、

性能需求等；

第四章系统设计：主要说明本设计的体系结构、功能模块设计、数据库设计等；

第五章系统实现：主要说明本设计的实现情况，主要功能的实现；

第六章系统测试：主要说明本设计的功能测试、兼容性测试等。

2 相关工具和技术

2.1 SQL Server 数据库介绍

SQL Server 是由 Microsoft 开发和推广的关系数据库管理系统 (DBMS)。可以存放多种类型的数据,也方便对数据进行灵活快捷的分析和处理。

2.2 Node JS 环境介绍

Node JS 是一个基于 Chrome V8 引擎的开源和跨平台的 JavaScript 运行时的环境,其使用了一个事件驱动、非阻塞式 I/O 的模型,使其轻量又高效。Node JS^[4]的出现让 JavaScript 得以不再依赖浏览器也可以在电脑环境运行,这也使得 JavaScript 的标准不再需要去适应浏览器的版本,只需要更改 Node 版本就可以决定使用哪个。ECMAScript 标准。

同时,其提供的包管理工具 NPM (Node Package Manager),如今,NPM 仓库托管了超过 1,000,000 个开源包,可以自由使用。还有众多的模块依赖,使得 Node 的开发设计可以更简单高效。

2.3 VUE 前端技术

VUE 是一套由国人设计的构建用户界面的渐进式框架,其采用自底向上增量开发的设计,使用 MVVM 响应式编程模型,避免直接操作 DOM,降低 DOM 操作的复杂性。VUE 的核心库只关注视图层,它不仅易于上手,还便于与第三方库或既有项目整合,以组件搭建的形式进行开发,常用的组件库像 Element UI^[5],可以迅速的搭建项目界面的基本框架。

2.4 KOA 框架

KOA 是一个新的 web 框架，由 Express 背后的同一个团队构建，致力于在 web 应用和 API 开发领域中成为一个更小、更富有表现力、更健壮的基石。KOA 并没有捆绑任何中间件，而是提供了一套优雅的方法，根据不同的需求设计引用相应的中间件，帮助您快速而愉快地编写服务端应用程序。

3 需求分析

3.1 可行性分析

可行性分析是通过对项目的主要内容及相应条件，如市场需求、资源供应、建设规模、环境影响、盈利能力等，从技术、经济、社会等方面进行调查研究和分析比较，从而预测在项目建成以后可能取得的经济效益及社会影响，为项目决策提供依据的一种综合性的系统分析方法。此应用设计和实现主要从以下两个方面来分析可行性。

3.1.1 技术可行性

在技术方面，所用的相关技术都是免费可获取的。使用 Visual Studio Code 代码编写工具，数据库使用 My SQL Server 数据库。同时对于前端技术有一定的项目经验，也花费了足够的时间对后端服务器的开发进行了一定的学习。而且在如今计算机行业的发展状态下，很多相关技术都已经发展成熟，因此在技术上是可行的。

3.1.2 社会可行性

现在社会的工作和压力下，人们越来越没有时间去注意生活，享受生活，人们总是会忘记或是忽视了自己某一刻的感受或想法。因此也确实在信息化，电子信息发展的今天，有着各种的方式或软件可以记录生活中的美好。功能也是应有尽有，记录的数据也不仅限于文字和图片。同时在现在信息化的社会中，每个人的生活本身就时时刻刻都是一种信息资源，而对这些信息资源的利用，经济化也已不再是新鲜事。因此，在社会的发展下，对个人信息的记录，管理，使用是可行的。

3.2 系统总体需求

作为一个移动 Web 应用，主要由用户进行操作。从用户角度进行分析，用户可以正常访问网址，进入首页，而查看记录页面和个人页在未登录的情况下会跳转至登录注册页，同时在添加记录时也会提醒登录并跳转；而在登录之后便可以在首页浏览其他人公开的生活记录，同时主要可以对自己的记录数据进行查找浏览，以及标记收藏和修改。

3.3 系统功能需求

本应用将功能需求主要分为用户模块和生活记录模块。用户用例图如图 3.1 所示。

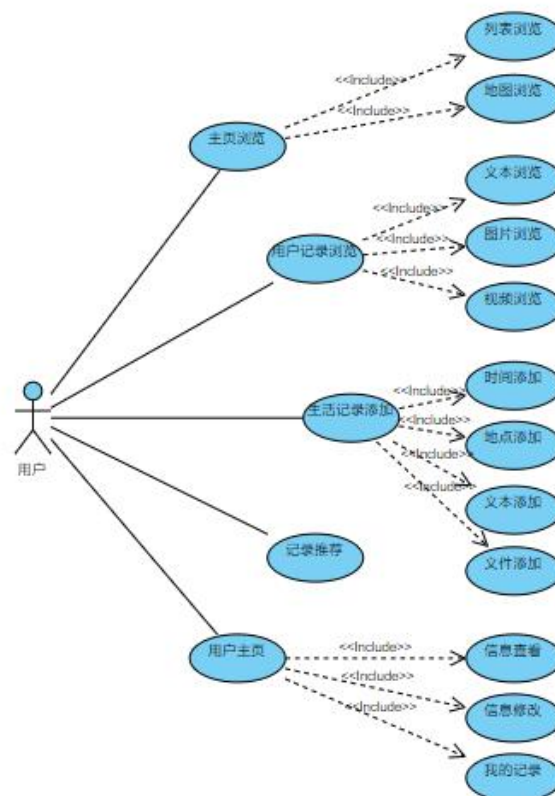


图 3.1 用户用例图

3.3.1 用户模块

用户主要有登录注册，用户信息修改，用户信息包括账户和账户名，头像。可以进行记录信息添加，查看记录信息，对记录信息进行备注收藏管理。

3.3.2 生活记录模块

首页，按用户操作筛选符合条件的记录信息进行主要信息展示，除列表展示外，还可以切换为地图上展示，可以点击查看详细记录信息。

浏览页，对用户的记录信息按记录详细内容进行分类展示，同时可以对用户记录信息按关键字搜索，对于搜索结果内容进行显示和高亮，搜索范围为文本内容和时间地点^[6]。

添加页，默认自动记录保存的时间地点，同时可以手动添加记录时间和地点，主要内容为文本，可以上传图片，视频，默认不公开用户记录信息，用户可以选择公开，让其他用户在首页和推荐页能够看到。

推荐页，主要以图文的形式展示其他用户的记录信息。

3.4 系统性能需求

较强的安全性，这是用户生活记录信息的管理储存，如今信息的价值，且个人信息都是属于每个人的重要隐私，隐私的泄露是一个不能容许出现的问题，因此安全性是一个必要且重要的性能要求。

广泛的兼容性，本项目作为一个移动 Web 的应用项目，目标是面向广大的用户全体，需要考虑到不同用户的移动设备也多种多样，让每一个用户都拥有一样的使用体验，设备的兼容性就需要必不可少

的处理。

友好的交互性，符合用户的常规操作习惯，作为一个日常生活记录信息的管理应用，需要用户经常性的进行使用，而让用户拥有一个良好的体验，需要让各个功能的操作，界面的布局，颜色设计上更友好，才可能让用户更愿意使用。

信息的可靠性，响应速度的处理，要使得记录的信息具有一定的价值性，那么数据的可靠性，真实性是应该得到保证的。同时对于数据的处理速度，为了更良好的使用体验，处理速度的优化也是必不可少的。

可维护性和可扩充性，随着信息社会的发展，信息获取的方式多样化，信息种类的多样化，人们需求的多样化，应用也应该随着社会的发展而得到完善和扩充，可以在后续优化更新。

4 系统设计

4.1 体系结构设计

本项目为移动 Web 应用^[7]，整体结构应由数据库服务，后台服务器，前端客户端服务三部分组成。经过需求分析，应用的前台主要功能总体设计如图 4.1 所示。

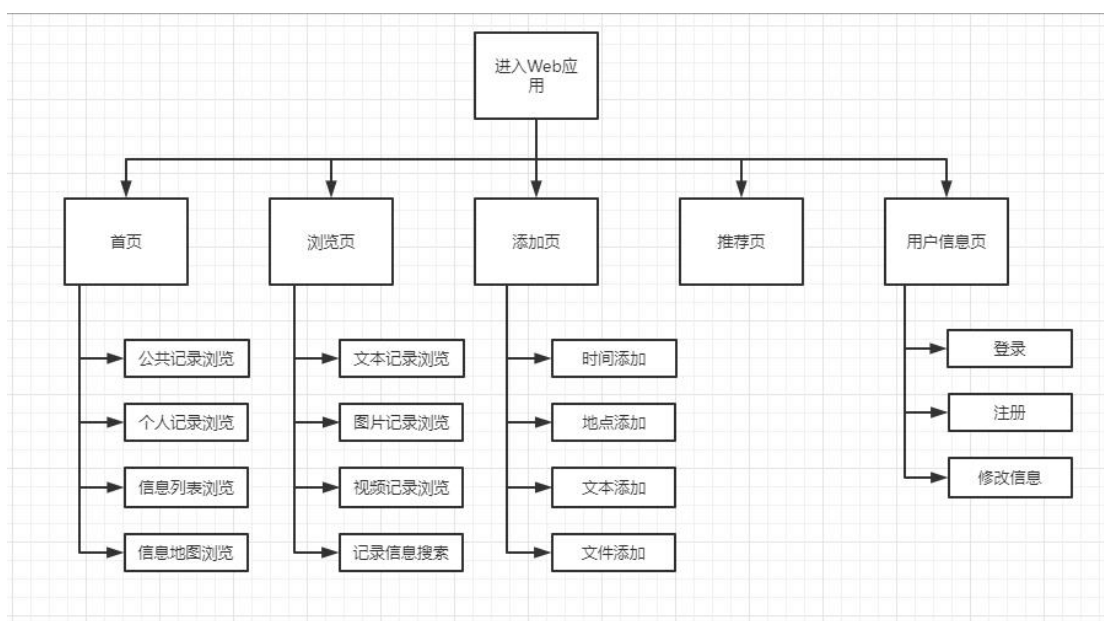


图 4.1 应用的前台主要功能结构图

4.2 主要功能设计

作为一个 Web 移动端的项目，首先应该适配常用的移动设备；然后主要分为首页，浏览页，添加页，推荐页，个人页五个底部导航栏。

首页，顶部是一个公共的头部导航，主要是三部分，左中右结构，动态的根据页面的不同显示不同的内容。首页的这个头部，中间为标题或 logo，右边为一个弹窗菜单，未登录时显示为切换，公开，登录：切换可以改变记录信息的展示方式——列表和地图；公开便展示所有公开的生活记录；登录就跳转登录页。然后页面中间为内容展示，列

表展示主要展示文本内容，有图就展示第一张图片，还应有一个选择栏，可以选择内容展示顺序；地图展示，将每条生活记录按记录地点标记在地图上，点击标记点可以查看标记点附近所有记录，点击展示的记录可以进入生活记录详细页。主页查看系统的时序图如图 4.2 所示。

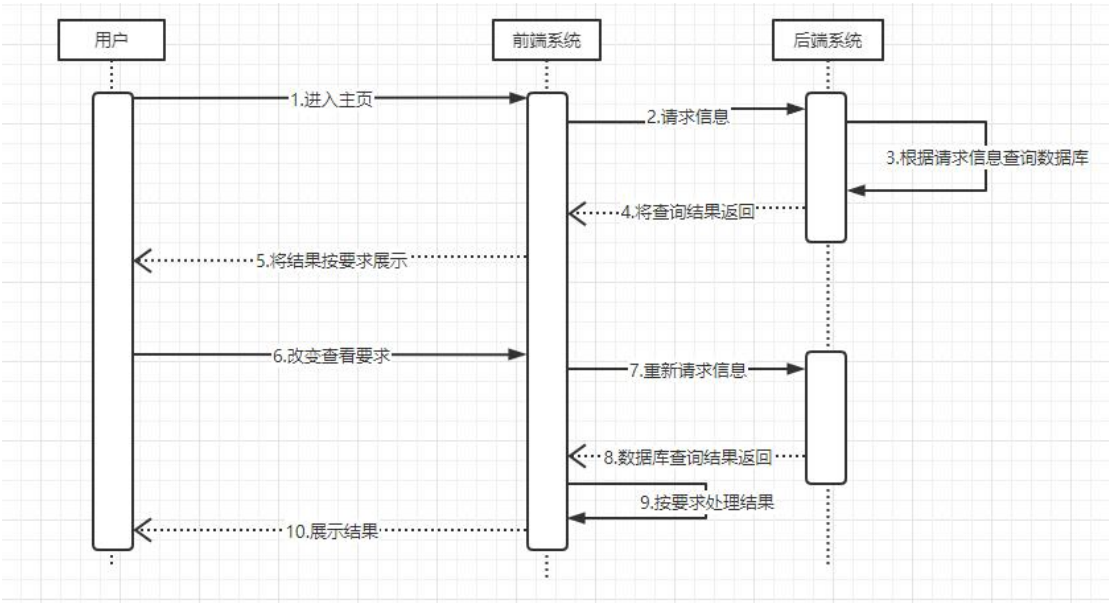


图 4.2 主页查看系统时序图

浏览页，顶部导航为一个搜索栏。然后是一个隐藏选择栏，可以选择内容的展示顺序，展示种类：主要有文本，图片，视频。点击搜索是会对展示的内容搜索，将关键字高亮，搜索范围主要是文本内容，日期时间，地点，最后还应将搜索的结果提示出来。

添加页，首先顶部左边为返回前一页，中间为 **logo**，右边为保存按钮。且此页默认隐藏底部导航栏，输入表单为时间，地点，文本内容，同时可以上传图片，视频，底部还可以选择是否公开生活记录。添加页系统的时序图如图 4.3 所示。

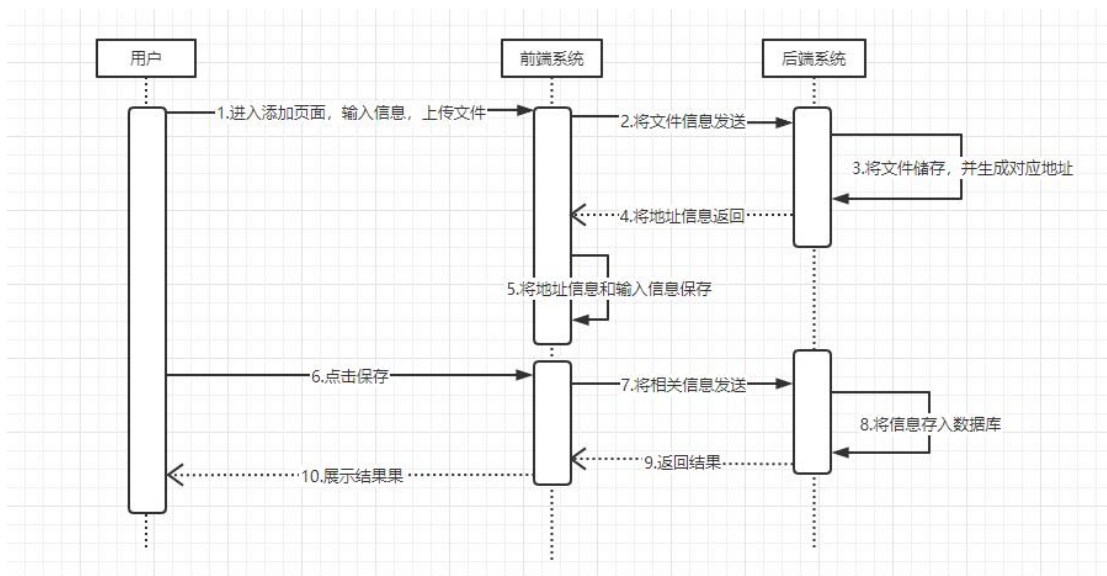


图 4.3 添加系统时序图

推荐页，顶部导航左边为返回按钮，中间为 **logo**，主体内容为图片或视频加文字的列表。

个人页，个人信息的展示，在最上层显示头像，用户名，账号，点击后可以进行修改；再是我的收藏，我的图片，我的视频等；最后是退出登录按钮。

详细页，显示该条生活记录的用户名，用户头像，再是这条记录的创造时间，地点，用户写入的时间，地点，事件内容，上传的图片，视频。

后端服务器接口，对于用户，有登录和注册的接口，以及根据用户 **id** 查询用户信息的接口；对于记录信息，有添加的接口，条件查询的接口，文件上传的接口；其中对于密码进行加密存储，使用 **token** 验证登录状态；其中后端服务和前端服务使用着不同的端口，因此在前端请求数据时，根据同源策略，请求会被浏览器拦截，解决这一问题得方法有很多，这里采用的是后端 **KOA** 框架提供的 **@koa/cors** 插件。

4.3 数据库设计

本系统的相关 E-R 图如图 4.4 所示，系统主要由用户创建所属的账号来绑定相关信息，将添加的生活记录信息关联到用户账号下。

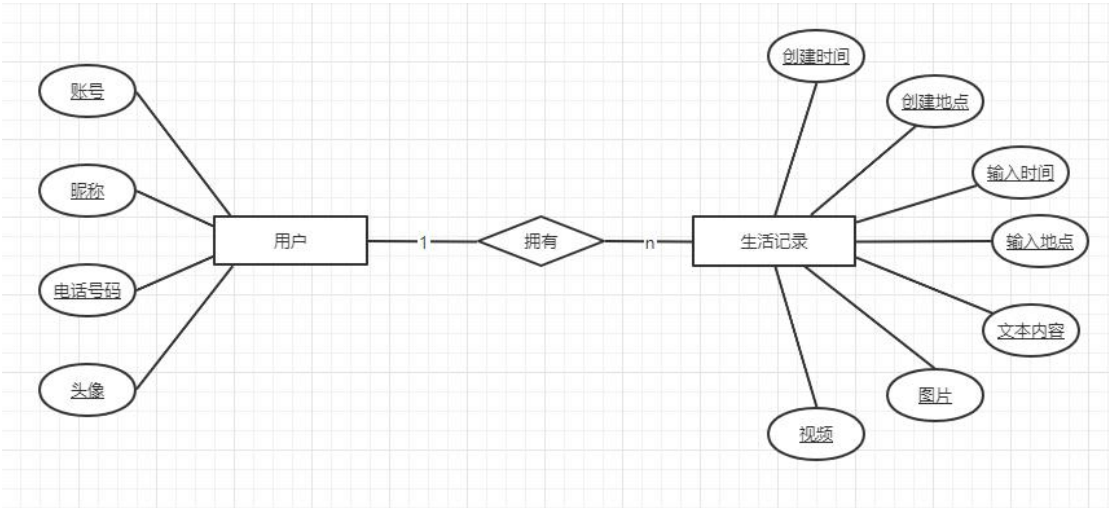


图 4.4 E-R 图

本应用主要设计了用户信息表和生活记录信息表。

表 4.1 用户信息表

属性	字段	类型	允许空	主键	长度
用户 ID	uid	bigint	否	是	20
用户昵称	name	varchar	是	否	11
用户密码	password	varchar	否	否	255
用户电话	tel	varchar	否	否	11
用户头像	img	text	是	否	-

表 4.2 生活记录信息表

属性	字段	类型	允许空	主键	长度
记录 ID	lid	bigint	否	是	20
用户 ID	uid	bigint	否	否	20
创建时间	ctime	varchar	否	否	32
创建地点	cpos	text	否	否	-
填入时间	time	varchar	是	否	32
填入地点	pos	text	是	否	-
文本内容	text	text	是	否	-
是否公开	share	tinyint	否	否	1
图片信息	img	text	是	否	-
视频信息	video	text	是	否	-

5 系统实现

5.1 主要功能页面实现

此 Web 应用使用 VUE 脚手架搭建项目框架^{[8][9]}，进行前端开发；后端服务使用 KOA Web 框架，前端^[10]使用 AXIOS 技术来进行数据的请求，后端使用 koa-bodyparser 和 koa-multer 中间件来实现对接收的数据预处理和文件的储存，从而实现前后端分离。主要功能页面为首页，查看页，添加页，详细页，用户页。VUE 项目中首先将这些页面文件创建，并在路由文件中配置好相应的路由，配置底部导航栏代码实现如图 5.1 所示

```
<template>
  <div class="nav">
    <router-link to="/home">
      <van-icon name="wap-home-o" class="icon" />
      <span class="text">首页</span></router-link>
    >
    <router-link to="/show">
      <van-icon name="apps-o" class="icon" />
      <span class="text">浏览</span></router-link>
    >
    <router-link to="/add">
      <van-icon name="add-o" class="icon add" />
    </router-link>
    <router-link to="/recommend">
      <van-icon name="gem-o" class="icon" />
      <span class="text">推荐</span></router-link>
    >
    <router-link to="/my">
      <van-icon name="user-circle-o" class="icon" />
      <span class="text">我的</span>
    </router-link>
  </div>
</template>
```

图 5.1 底部导航代码实现

5.1.1 主页模块

展示信息分为列表展示和地图展示，整体为上中下的布局，对于

展示条件分别用变量对应绑定，在每次操作后重新获取后台数据，相关顺序展示在数据接收后在前端进行处理排序后展示；对于地图，本项目使用的是百度地图的 API，VUE 项目中为方便使用直接引入百度地图的 VUE 组件库，如图 5.2 所示

```
> import Vue from "vue" ...
Vue.use(BMapGL, {
  ak: 'EsCjaTRh6w4qLow45777ZmAS0gT8r1KS' //去百度地图官网申请的ak
})
```

图 5.2 引入百度地图代码

然后是在首页的使用，直接写<baidu-map/>标签来进行地图的调用显示。弹窗选择菜单代码如图 5.3 所示，其中地图的点击与触摸事件处理代码如图 5.4 所示

```
selectBar(actions) {
  if (actions.text === '切换') {
    this.changeType = !this.changeType
  } else if (actions.text === '公开') {
    this.seachData = { share: 1 }
    this.findData()
    this.range = actions.text
  } else if (actions.text === '登录') {
    this.$router.push('/login')
    this.barType = false
  } else {
    this.seachData = { uid: this.$store.getters.uid }
    this.findData()
    this.range = actions.text
  }
},
```

图 5.3 弹窗选择菜单代码

```

handler(e) {
  e.map.addEventListener('touchmove', () => {
    e.map.enableDragging()
  })
  e.map.addEventListener('touchend', () => {
    e.map.disableDragging()
  })
  e.map.disableDragging()
  const geolocation = new e.BMap.Geolocation()
  geolocation.getCurrentPosition((p) => {
    console.log(p.point)
    this.center = p.point
  })
},

```

图 5.4 点击与触摸事件处理代码

5.1.2 查看模块

信息种类的展示，将请求的结果进行图片，视频，文本的分类处理，按需求显示不同种类的内容，同时进行排序展示。其中关键字搜索^[11]通过将关键字与文本内容，时间地点进行查询匹配，将匹配的结果进行重新存储，同时将满足的结果进行字符串替换然后高亮^{[12][13]}显示，代码如图 5.5 和图 5.6 所示，将 text 使用 v-html 展示，而为了防止 XSS 攻击，这里预先将 text 内容中的‘<’全部替换为‘<’^[9]。代码如图 5.7 所示

```

v.text = v.text.replace(
  Reg,
  `<span style="color: #eb7340;">${keyWord}</span>`
)

```

图 5.5 关键字高亮


```

const Reg = new RegExp(keyWord, 'g')
this.showSearch[keyWord] = ['', '', '']
this.info = this.info.filter((v) => {
  let searchKey = [keyWord]
  if (v.text.search(keyWord) !== -1) {
    searchKey.push('文本内容')
    this.showSearch[keyWord][0] = '文本内容'
    v.text = v.text.replace(
      Reg,
      `<span style="color: #eb7340;">${keyWord}</span>`
    )
  }
  if (v.cpos.address.search(keyWord) !== -1) { ...
  }
  if (v.pos.address.search(keyWord) !== -1) { ...
  }
  if (this.timeTo(v.ctime).search(keyWord) !== -1) { ...
  }
  if (this.timeTo(v.time).search(keyWord) !== -1) { ...
  }
  v.searchKey = searchKey
  return searchKey.length !== 1
})
console.log('search' this.info)

```

图 5.6 关键字搜索

```

async findData() {
  return await $findBy({ uid: this.$store.getters.uid }).then((res) => {
    if (res.data.status === 0) {
      this.info = res.data.data
      this.info.forEach((v) => {
        v.text = v.text.replace(/</g, '&lt;')
      })
      console.log(this.info)
    } else {
      console.log(res)
    }
  })
}

```

图 5.7 防止 XSS 攻击

5.1.3 添加模块

时间地点的添加均采用 Vant UI^[5]组件的时间地点选择器,这里使用弹框在点击输入时弹出供用户选择,时间选择器的结果为时间戳,因此要进行一个转换处理,代码如图 5.8 所示;地点选择器使用百度地图的 API,点击事件来确定选择的大概地点;文本内容便是一个文

本域输入框；图片，视频文件的上传也使用 Vant 的组件，点击上传文件，限定为图片和视频，在获取到文件后调用接口上传至服务器，将服务器返回的地址保存代码如图 5.9 所示。文本内容不能为空，保存时会提示信息不足；若未登录则会跳转至登录页。

```
formatTime(date) {
  let y = date.getFullYear()
  let m = date.getMonth() + 1
  m = m < 10 ? '0' + m : m
  let d = date.getDate()
  d = d < 10 ? '0' + d : d
  let h = date.getHours()
  h = h < 10 ? '0' + h : h
  let minute = date.getMinutes()
  minute = minute < 10 ? '0' + minute : minute
  let second = date.getSeconds()
  second = second < 10 ? '0' + second : second
  return y + '-' + m + '-' + d + ' ' + h + ':' + minute + ':' + second
},
```

图 5.8 时间转换处理

```
afterRead() {
  let file = this.fileList[this.fileList.length - 1]
  const formdata = new FormData()
  formdata.set('file', file.file)
  $upload(formdata).then((res) => {
    console.log(res.data)
    let data = res.data
    if (data.code === 200) {
      file.type = data.type
      file.url = data.url
    } else {
      console.log(res)
    }
  })
},
```

图 5.9 文件上传

5.1.4 详细模块

通过跳转，详细页会使用跳转时的传参，用户 ID，信息 ID，查询本条生活记录的相关信息，传参与接收如图 5.10 和图 5.11 所示，主要显示用户名和用户头像，记录信息会将全部内容展示，创建时间

地点，输入时间地点，文本内容，图片，视频。且点击图片将调用组件全屏查看。

```
<div class="list">
  <router-link
    v-for="i in info"
    :to="{ name: 'Info', query: { uid: i.uid, lid: i.lid } }"
    :key="i.lid"
  >
```

图 5.10 传参

```
activated() {
  console.log(this.$route.query)
  this.findData({
    info: { lid: this.$route.query.lid },
    user: { uid: this.$route.query.uid },
  })
},
```

图 5.11 接收参数

5.1.5 用户模块

主要为用户相关的内容或操作显示，包含用户信息，用户管理的生活记录，以及退出。

5.2 服务相关接口实现

服务器使用 KOA 框架搭建，首先启动 My SQL Server 数据库，连接本地数据库。接口的设计，在访问接口时调用对应的方法进行相应数据库操作。

文件上传接口，使用 **koa-multer** 中间件进行处理^{[14][15]}，将文件储存在指定位置，在返回相关参数；再前端访问这个接口时，先中间件处理，再将处理的数据整理返回至前端。

登录注册接口，将访问接口的数据接收，进行数据库查询，根据查询结果，是否注册或登录成功。对密码的储存使用 **bcryptjs**^[16] 进行加密处理如图 5.12 所示


```

//对密码进行加密处理,将用户信息插入数据库
const salt = bcrypt.genSaltSync(12)
let password = bcrypt.hashSync(info.password, salt)

```

图 5.12 加密处理

生活记录信息接口,将数据在数据库的基础上做增添,模糊查询,将接收的参数遍历,将有效的参数组合成数据库查询语句,进行查询,遍历处理代码如图 5.13 所示

```

async findBy (ctx) {
  let info = ctx.query
  let condition = ''
  let valueKey = ''
  console.log(info)
  if (Object.keys(info).length > 0) {
    condition += 'where '
    Object.keys(info).forEach((v, i) => {
      if(!info[v]){
        valueKey = v
      }
      if (i !== 0) {
        condition += ' and '
      }
      condition += `${v}=${info[v]}`
    })
  }
  if(valueKey){
    ctx.body = {
      status: 1,
      message: `${valueKey}的值有误!!!`
    }
    return
  }
  const sqlIns = `select * from logs ${condition}`
}

```

图 5.13 参数遍历处理

6 系统测试

软件系统，网站应用进入市场或投入运营之前，都需要经过完善的测试。应用的完备性和稳定性得到保障，这样才能更好的占据市场，受用户使用和评价。系统测试和系统开发是项目中必不可少的两部分，甚至说系统测试比系统开发更加的重要。在项目开发的过程中，需要不断的、反复的对项目进行测试工作。本项目是一个移动端的 Web 应用，除功能性测试外，应该也要对其设备的兼容性进行测试。

6.1 功能性测试

(1) 应用登录注册测试

测试需求：注册账号，登录账号

测试用例：创建账户 **admin** ，手机号 **99999999999** 密码 **admin123**；
然后用这个账号登录

期望输出：点击注册后显示注册成功，再登录后进入到个人主页
结果如图 6.1 所示。



图 6.1 登录成功

(2) 生活记录添加测试

测试需求：为账号 **admin**，添加一条生活记录

测试用例：登录 **admin** 账号，进入添加页，输入时间 2000-1-1，地点湖南长沙，文本‘test 测试’，图片，视频

期望输出：登录成功后，进入添加成功，跳转首页，查看最新可以看到这个添加记录

结果如图 6.2 和图 6.3 所示。

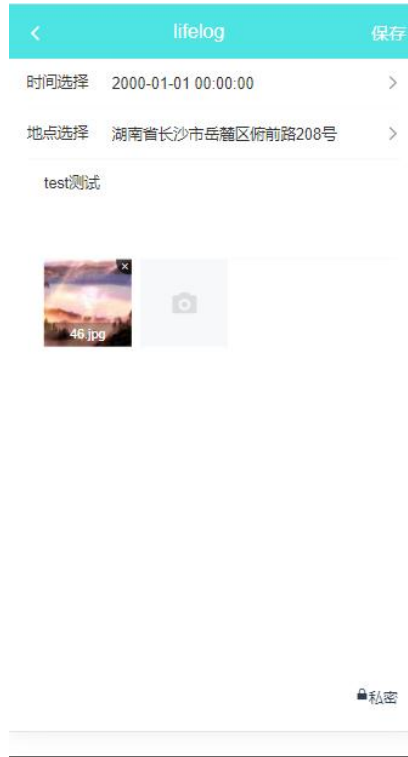


图 6.2 添加记录



图 6.3 查看记录

(3) 生活记录浏览测试

测试需求：用账号，浏览首页，浏览页，测试相关筛选排序和搜

索

测试用例：登录测试账号，进入首页和浏览页

期望输出：登录成功后，进入首页，点击右上角的弹窗菜单，切换可以在地图上看到记录的标点，点击查看信息详细；切换回去，点击排序，列表按时间，按地点距离排序；进入浏览页，点击类型，可以筛选不同的内容查看，也可以点击排序；点击搜索框，输入‘test’，可以显示搜索的结果类型，且显示搜索结果列表。

结果如图 6.4-6.8 所示。



图 6.4 首页界面列表展示



图 6.5 首页界面地图展示

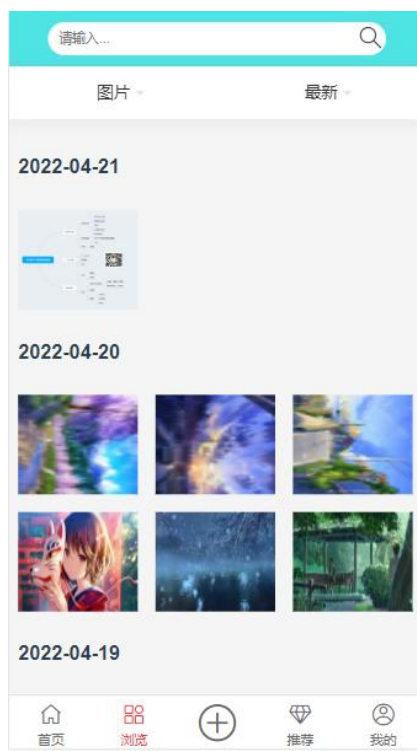


图 6.6 浏览界面图片展示



图 6.7 浏览界面搜索展示



图 6.8 详细界面展示

6.2 兼容性测试

(1) 浏览器模拟设备测试

测试需求：在不同设备下运行访问

测试用例：在浏览器上访问网页，切换不同的移动设备

期望输出：在不同设备下的效果基本保持一致，功能也都正常

（2）局域网真机设备测试

测试需求：在真机上运行访问

测试用例：这里用 vivo S6， vivo X21 进行访问测试

期望输出：功能基本正常，布局样式显示无误

结果如图 6.9-6.10 所示。



图 6.9 局域网真机运行

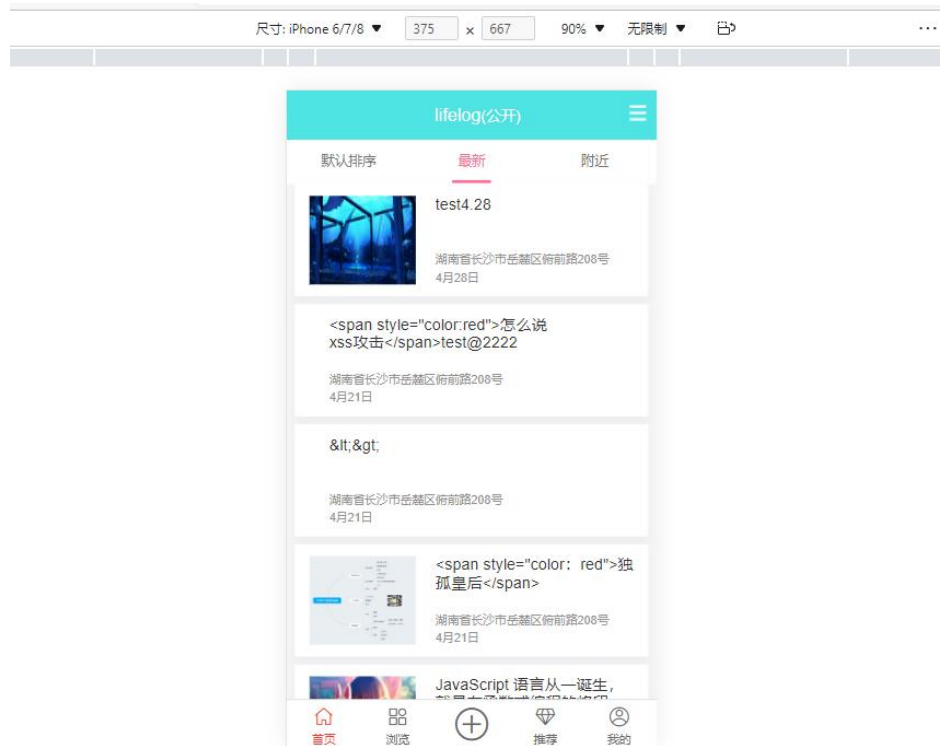


图 6.10 浏览器模拟设备运行

结论

在基于定位的生活记录 Web 应用的设计与实现中，查阅了大量文献和资料，并自主学习了相关前后端知识，在过程中发现了许多实际实现中才会出现的问题，最终在现有相关技术条件下，实现了这个 Web 应用的基本功能。

在这几个月的查阅，学习，设计与实现，修改和测试过程中，得到了来自老师的细致指导，同学，朋友的亲切帮助，使我得以在过程中顺利的解决各式各样的问题。同时也了解，学习到了很多以前没有了解过的知识，也在过程中深刻体会到了理论与实践的差距，认识到了理论与实践相结合的重要性。

最后，对于生活记录的管理实现，还有许多不足的地方，还有许多必要的功能也有待去实现。但在这个过程中我所收获的却远不止这些，我相信我所得的一定可以在我以后的生活中有所体现。

参考文献

- [1] 潘志宏, 罗伟斌, 柳青.基于 HTML5 跨平台移动应用的研究与实践[J].电脑知识与技术, 2013(6): 3992-3995.
- [2] 俞华锋.基于 HTML5 的网页设计与实现[J].科技信息, 2012(29):115-116.
- [3] 陈鲱.Web 前端开发技术以及优化方向分析[J].新媒体研究, 2015(7):39-40.
- [4] 裴之葦,高艳霞.基于 Vue 和 Node.js 的手语教学 Web 平台的设计与实现[J].电脑与信息技术,2021,29(06):33-36+86.
- [5] 王志文.Vue+Elementui+Echarts 在项目管理平台中的应用[J].山西科技,2020,35(06):45-47.
- [6] 江永池.基于百度地图 API 的主动定位研究与实现[J].韩山师范学院学报,2021,42(03):66-72.
- [7] Okamoto Hiroyuki,Mochizuki Toshihiko,Yokoyama Kazutoshi,Wakita Akihisa,Nakamura Satoshi,Ueki Heihachi,Shiozawa Keiko,Sasaki Koji,Fuse Masashi,Abe Yoshihisa,Itami Jun. Development of quality assurance/quality control web system in radiotherapy[J]. Nihon Hoshasen Gijutsu Gakkai zasshi,2013,69(12): 1405-1411.
- [8] 王萍利.基于 HTML5 的 Web 前端框架设计及研究[J].电脑编程技巧与维护,2021(12):10-12.
- [9] 赵怡姗,范明钰.基于 HTML5 与 CSS3 的网页设计技术研究[J].成都

信息工程大学学报,2021,36(06):641-645.

[10] M. Oliveira,M. Guebert,P. Nohama. TEACCH Methodology-Based Web System to Support Learning for Children with Autism[J]. IEEE Latin America Transactions,2018,16(11): 2698-2705.

[11] 慕慧,徐晓慧.基于 Web 系统的大数据搜索技术的实现和优化[J]. 计算技术与自动化,2021,40(01):155-163.

[12] Yi Z. Design and implementation of idle Book Trading System Based on Web[J]. Journal of Web Systems and Applications, 2021, 3(1): 1-6.

[13] Guan Xiong Wei,Di Bo Hou,Ping Jie Huang,Guang Xin Zhang. Development of a Web System for Water Quality Monitoring Information Management[J]. Applied Mechanics and Materials,2013,2369(316-317): 678-681.

[14] 罗文,兰全祥.基于 Node.js 的创新团队管理系统的优化与改进[J]. 信息技术与信息化,2020(09):22-25.

[15] Frolin S. Ocariza Jr.,Guanpeng Li,Karthik Pattabiraman,Ali Mesbah. Automatic fault localization for client-side JavaScript[J]. Software Testing, Verification and Reliability,2016,26(1): 69-88.

[16] Huili Xue,Mengliang Shao. Key technologies for solving high-concurrency problems in web systems[J]. Journal of Computational Methods in Sciences and Engineering,2018,18(2): 309-318.