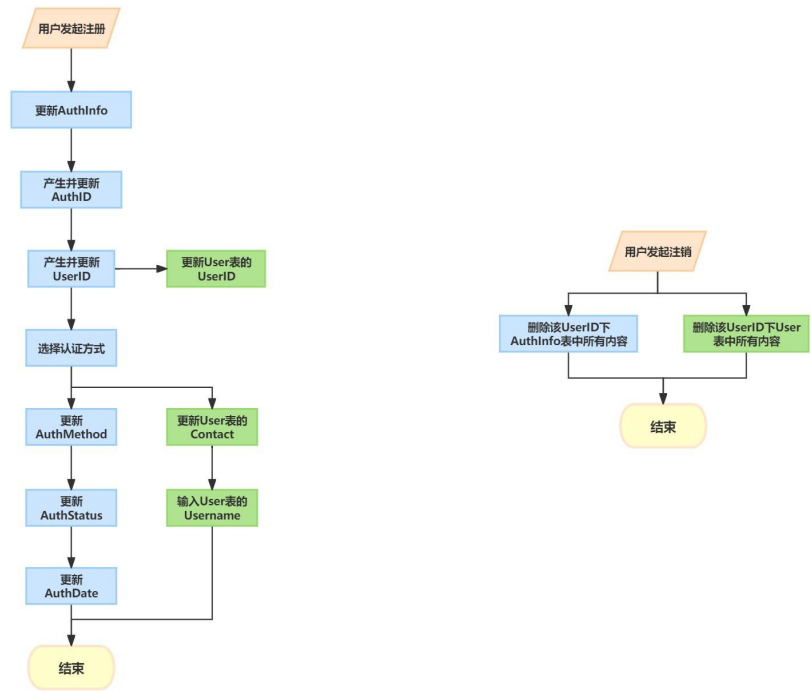


用户注册部分设计

在用户注册部分主要涉及到两大功能：注册和注销，两个表：用户信息认证（AuthInfo）以及用户表（User）。两个表之间有一个 UserID 进行关联。

主要流程如下：



一些相关任务：

1. 确定需求和流程：

完善流程图中的流程；

确定好每个部分在数据库中的存储结构。

2. 数据库设计：

认证信息表(AuthInfo)：

认证 ID (AuthID): 唯一标识每个认证记录的主键。系统指定

用户 ID (UserID): 关联的用户 ID，外键，关联到 Users 表的 UserID。系统指定

认证方式 (AuthMethod): 认证方式，如邮箱验证、手机验证等。用户选择

认证状态 (AuthStatus): 当前的认证状态，如已验证、待验证等。系统认定

认证日期 (AuthDate): 认证完成的日期。系统认定

用户表(User):

包括但不限于:

UserID (用户 ID): 主键, 唯一标识每个用户。

Username (用户名): 用户的名称。

Contact (联系方式): 用户的联系方式。

3. API 设计:

后端提供可进行 CRUD 操作的接口, 以便 vue 前端进行操作。比较常见的是 REST API。

其他接口应包括用户注册、登录、获取认证信息、更新认证方法。

4. 后端实现:

可以使用 Node.js 或 python 作为后端语言, 应用 oreca1 里的驱动即可。

5. 前端实现:

使用 Vue.js 构建用户界面, 包括表单用于注册和登录。

使用 Vuex 管理状态, 尤其是用户认证状态。

使用 axios 或其他 HTTP 库与后端 API 通信。

实现响应式布局以适应不同设备。

6. 前后端交互:

确保前端发送的请求能够得到正确的响应, 包括错误处理。

实现前端的用户认证流程, 例如使用 JWT (JSON Web Tokens)。

7. 安全性:

确保 API 的安全性, 实现适当的用户认证和授权机制。

数据库密码、用户密码等敏感信息要加密存储。

8. 测试:

对 API 进行单元测试和集成测试。

对 Vue 前端进行单元测试和端到端测试。

对应数据库操作如下：

创建表：

```
1. CREATE TABLE Users (  
2.     UserID NUMBER GENERATED BY DEFAULT AS IDENTITY, -- 用户 ID  
3.     Username VARCHAR2(50) NOT NULL, -- 用户名  
4.     Contact VARCHAR2(100), -- 联系方式  
5.     CONSTRAINT PK_Users PRIMARY KEY (UserID) -- 主键约束  
6. );  
7.  
8. CREATE TABLE AuthInfo (  
9.     AuthID NUMBER GENERATED BY DEFAULT AS IDENTITY, -- 认证信息  
10.    ID  
10.    UserID NUMBER NOT NULL, -- 对应的用户 ID  
11.    AuthMethod VARCHAR2(50) NOT NULL, -- 认证方法  
12.    AuthStatus VARCHAR2(20) NOT NULL, -- 认证状态  
13.    AuthDate DATE NOT NULL, -- 认证日期  
14.    CONSTRAINT PK_AuthInfo PRIMARY KEY (AuthID), -- 主键约束  
15.    CONSTRAINT FK_AuthInfo_Users FOREIGN KEY (UserID) REFERENCES  
16.    Users(UserID) -- 外键约束  
16. );
```

更新内容：

添加新认证信息：

```
1. INSERT INTO AuthInfo (UserID, AuthMethod, AuthStatus, AuthDate)  
2. VALUES (1, 'EmailVerification', 'Pending', SYSDATE); -- 假设  
   UserID 为 1
```

添加新用户：

```
1. INSERT INTO Users (Username, Contact) VALUES ('new_user', 'user@a  
   bc.com');
```

注销用户：

```
1.      -- 删除用户的认证信息
2.      DELETE FROM AuthInfo
3.      WHERE UserID = :userId;
4.
5.      -- 删除用户记录
6.      DELETE FROM Users
7.      WHERE UserID = :userId;
```