**GAM组会议室预约系统项目文档**

**开发人员：**郭阳 陈源 王润玺 **开发周期：**2024.9 - 2024.12

**技术栈：**WechatApp **指导教师：**苏亭

1. **项目启动与计划文档**

基于Feature的要求，我们小组商议后决定开发一款基于微信小程序的会议室预约系统，用户通过扫描二维码进入程序，使用学校邮箱账号登录后即可查看会议室预约情况、预约会议室和查看预约记录。系统支持多用户管理，同时提供冲突检测和订阅提醒功能，确保用户可以便捷、高效地完成会议室预约。另外设置管理员账号，绑定特定邮箱，管理员登陆后可以查看所有用户的预约情况，并提供“重置”按钮，可以一键清空所有预约记录，使预约时刻表恢复默认状态，用于系统后期数据清理和维护升级。

**核心功能概述：**

登录与验证功能：支持用户通过学校邮箱注册并登录系统；

会议室预约功能：用户可以选择会议室及对应时间段进行预约，并在页面显示预约结果；

冲突检测功能：当两个用户尝试在相同时间段预约同一会议室时，系统弹出冲突警报，提示用户无法完成预约；

订阅提醒功能：支持用户查看其他用户已预约的会议室时间段。当某个时间段的预约被取消时，系统通过消息通知订阅的用户，提示当前时间段可供预约。

**系统架构与技术概述：**

小程序主要使用JavaScript语言开发，结合微信小程序框架。通过 Observer模式实现通知订阅功能，确保消息实时发送给订阅用户。使用云数据库记录用户预约数据并实现冲突检测功能。

1. **需求分析与用例文档**

**用户需求分析**

1. 支持用户登录功能。
2. 允许用户预订会议室，包括选择会议室名称和时间段（开始和结束时间）
3. 确保会议室预订实例唯一，验证会议室的预订时间段是否有冲突.
4. 在冲突发生时触发警报，提示用户冲突信息。
5. 允许用户订阅已被预订的会议室的时间段。
6. 在订阅的会议室预订被取消时，向订阅用户发送通知

**用户用例分析**

**相关需求：**

用户需要登录系统，管理会议室预约，处理预约冲突，并订阅已被预约的会议室以接收通知。**参与者：**

用户是主要发起者，数据库和通知系统为支持功能的参与者。

**前置条件：**

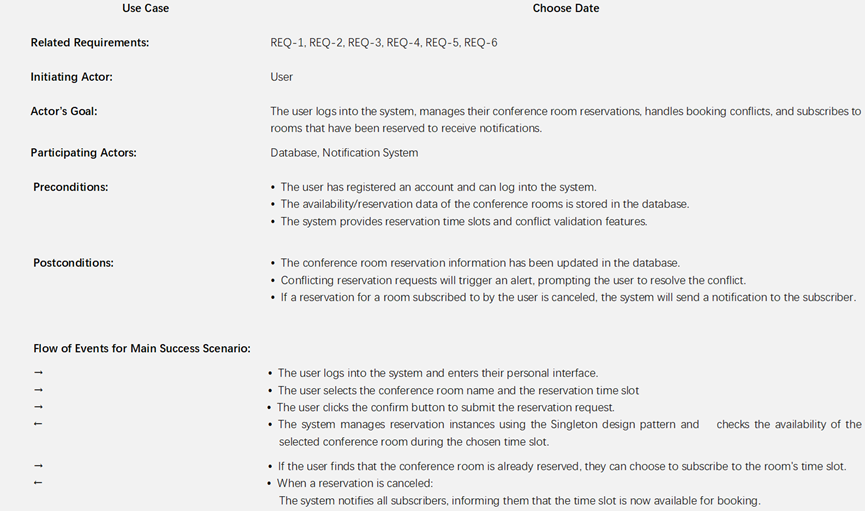
用户已注册账号并成功登录系统。系统数据库中已存储会议室的可用性和预约数据。系统提供时间段预约和冲突检测功能。

**后置条件：**

系统将更新数据库中的预约信息。如果存在预约冲突，系统会发出警报，提示用户解决冲突。用户订阅的会议室若被取消预约，系统会向订阅者发送通知。

**主要流程：**

用户登录系统，进入个人界面。选择会议室和预约时间段后，提交预约请求。系统利用 Singleton 模式管理预约实例，验证所选时间段的可用性。若会议室已被预约，用户可选择订阅该时间段。当预约取消时，系统通知订阅用户该时间段可用。



**管理员需求分析**

1. 允许管理者查看所有用户的预约记录2. 允许管理者清除所有用户的预约记录

**管理员用例分析**

**相关需求：**

管理员需要能够查看所有用户的预约记录，并可以选择清除所有用户的预约信息。

**参与者：**

管理员是主要发起者，数据库是参与者，用于存储和管理预约数据。

**前置条件：**

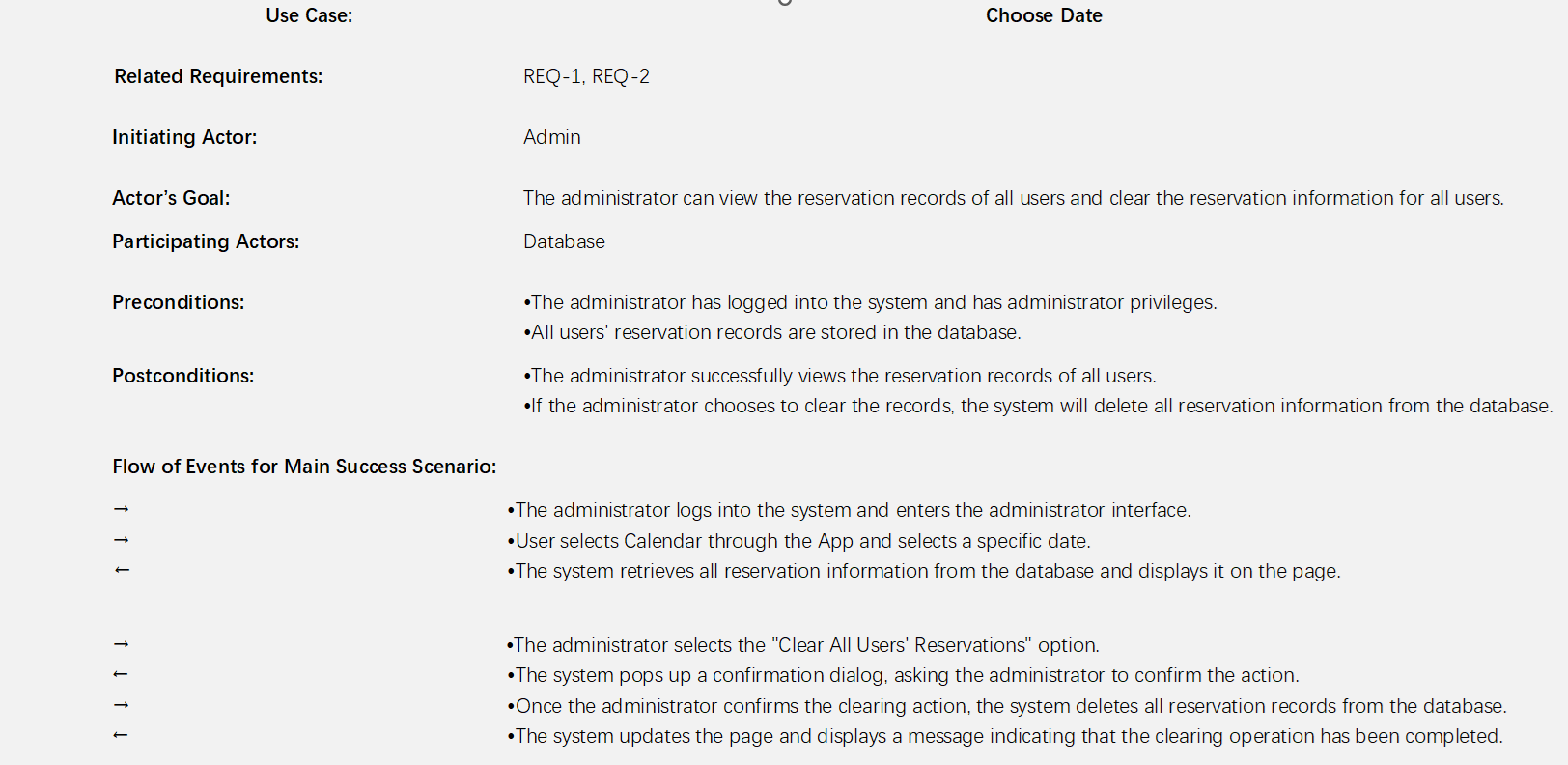
管理员已成功登录系统，并具有管理员权限。所有用户的预约记录已存储在数据库中。

**后置条件：**

管理员成功查看所有用户的预约记录。如果管理员选择清理记录，系统会从数据库中删除所有预约信息。

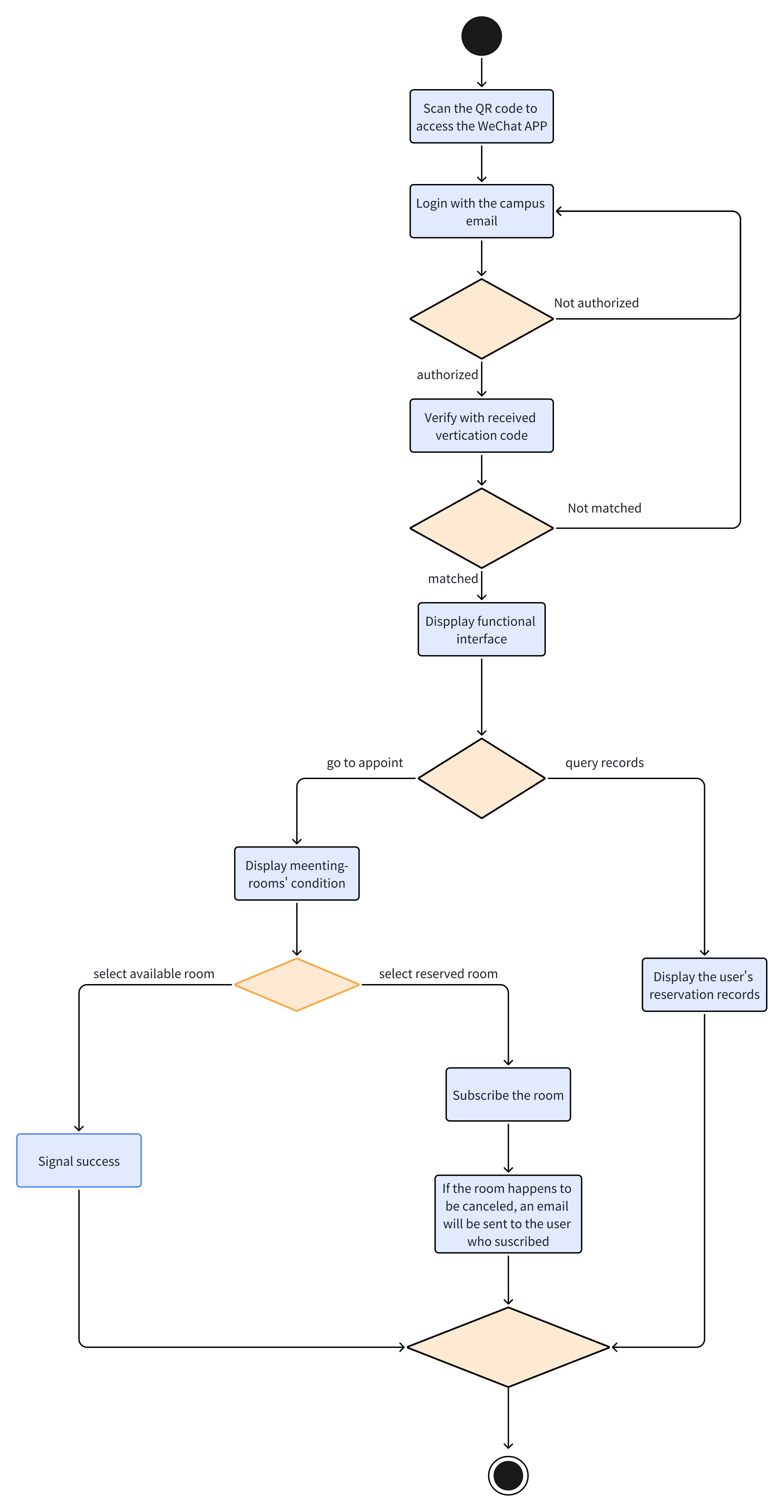
**主要流程：**

管理员登录系统，进入管理员界面。管理员通过日历界面选择一个特定日期。系统从数据库中检索所有用户在该日期的预约记录，并在页面上显示。管理员选择“清除所有用户预约”选项。系统弹出确认对话框，要求管理员确认操作。管理员确认后，系统从数据库中删除所有预约记录。系统更新页面，并显示清理操作已完成的提示信息。



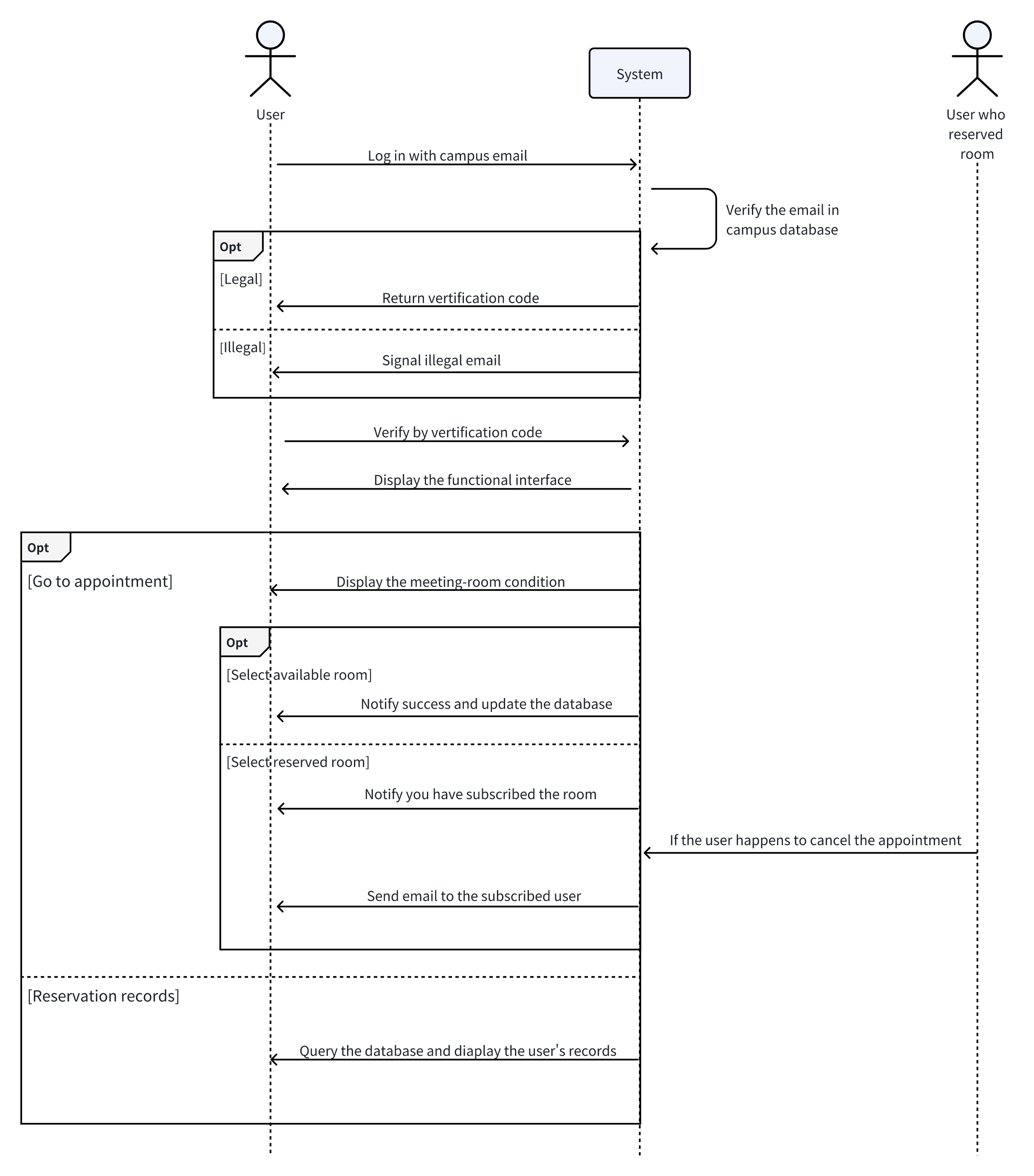
1. **设计文档**

**用户活动图**



用户通过扫描二维码进入小程序后，用校园邮箱登录并完成授权和验证码验证。若验证成功，系统将展示功能界面。功能包括“预约会议室”和“查询预订记录”两种主要路径。在预约会议室时，用户可以选择可用的房间，完成预订，系统记录并提供预订成功提示；用户还可以订阅已预约的房间，若该房间的预约恰好被取消，系统将发送邮件通知用户。在查询记录路径中，用户可以查看自己的预订信息。整个流程以用户成功完成操作或退出为结束。

**用户系统时序图**



**登录阶段：**

用户输入校园邮箱登录，系统验证邮箱是否合法。如果邮箱合法，系统发送验证码；否则，提示非法邮箱。用户输入验证码后，系统验证并展示功能界面。

**功能操作阶段：**

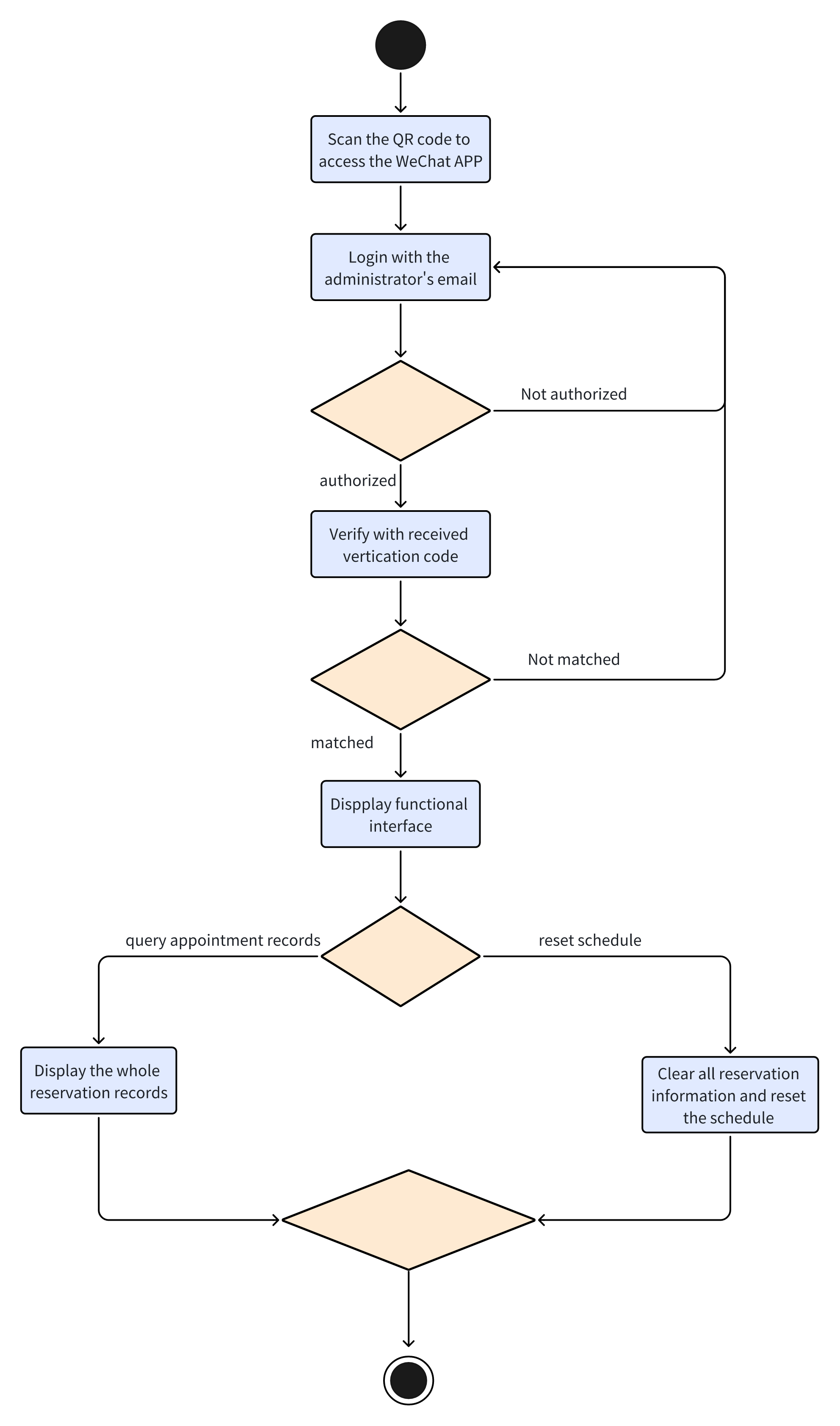
**预约会议室：**

用户选择查看会议室状态，系统显示当前会议室情况。如果用户选择可用会议室，系统记录预约信息并返回成功提示。如果用户选择已预约的会议室，系统提示已订阅此房间，若后续用户取消预订，系统通过邮件通知订阅用户。

**查询预约记录：**

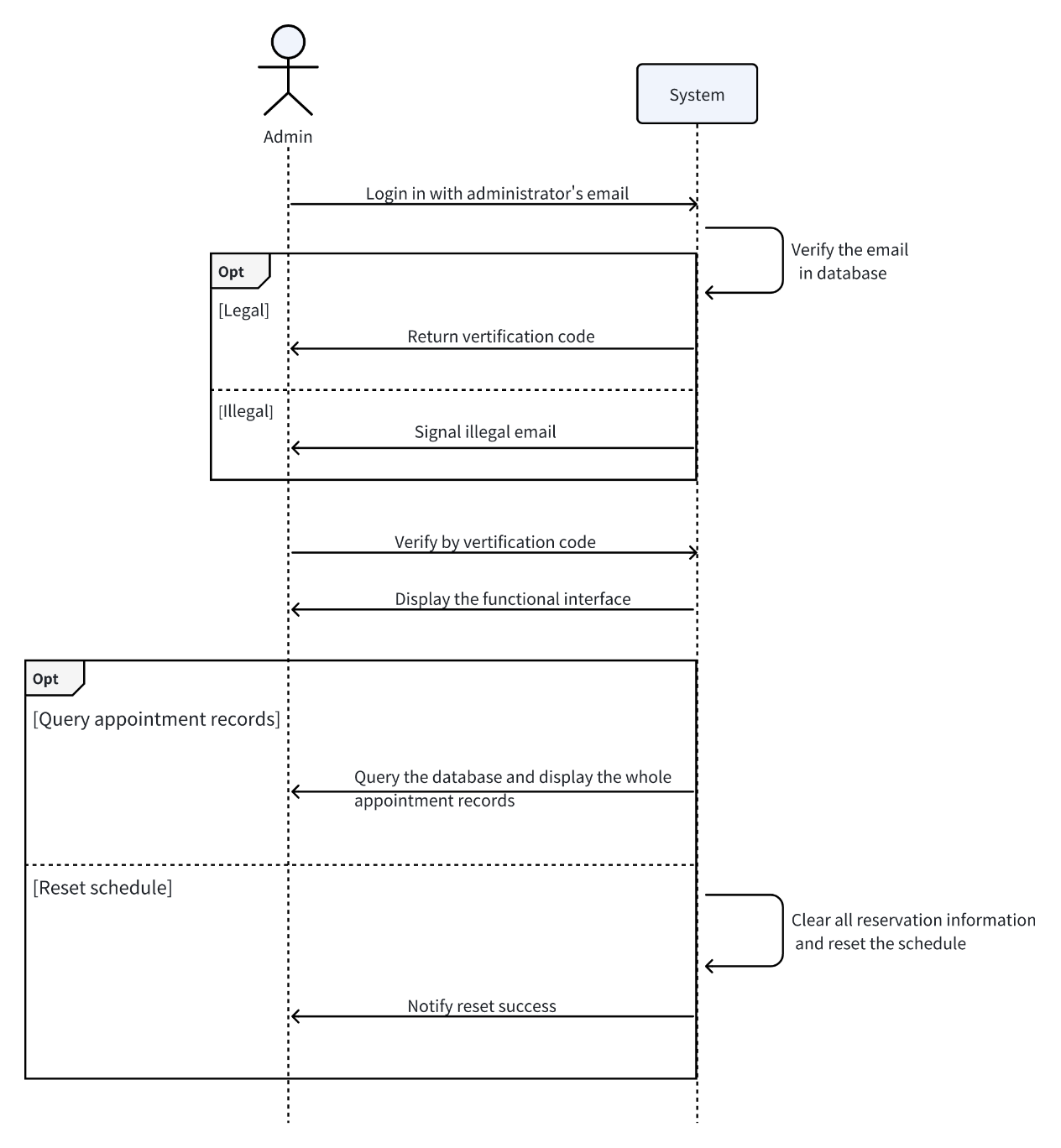
用户可选择查询功能，系统从数据库中查询并展示用户的预约记录。

**管理员活动图**



管理员通过扫描二维码进入微信小程序，使用管理员邮箱登录。若验证成功并通过验证码匹配后，进入系统功能界面。在功能界面，管理员有两个主要操作选项：查询预约记录：显示所有预约记录，供管理员查看；重置日程：清除所有预约信息并重置日程安排。整个流程以管理员成功完成操作或退出为结束。

**管理员系统时序图**



**登录阶段：**

管理员使用邮箱登录，系统验证邮箱是否合法。如果邮箱合法，系统返回验证码；否则提示邮箱非法。验证码通过后，系统允许管理员进入功能界面，提供管理功能。

**功能操作阶段：**

**查询预约记录：**

管理员选择查询预约记录功能。系统从数据库中查询所有预约信息，并将结果返回展示给管理员。

**重置日程：**

管理员选择重置日程功能。

系统清空所有预约记录并重置日程，完成后向管理员返回“重置成功”的通知。

1. **开发文档**

主要问题1：

预约冲突的预防：多名用户同时预约同一时间段的同一会议室，如果没有很好的预防机制，会出现同一时间段的同一会议室被重复预约的可能性。

解决方案：采取单例设计模式，同一时间仅存在一个预约实例。

具体实现：添加类似锁的功能，利用微信小程序数据库的特性，在触发预约功能时，向数据库提交一个对判断是否可修改的数据C查询，如果得到结果为可以则可进行预约，否则则提示已被预约。由于微信小程序数据库的异步操作性，两个用户A,B同时提交查询申请时，数据库会进行异步操作，而先提交的操作在预约完后会把C置反，即A预约成功后，B对C查询会得到不可预约的返回，最后在小程序端提示给用户刷新表单。

主要问题2：

批量推送订阅消息：多名用户订阅一个已被预约的会议室，在该会议室被取消后，通知所有订阅用户。

解决方案：采取观察者模式，用户订阅时创建一个观察者，当会议室被取消时对所有观察者发送提示邮件。

具体实现：用户订阅时创建数据库集合A，包含用户id，会议室名称，预约时间段等，在会议室被取消的方法中，添加一个功能，会议室被取消时，查询所有与该会议室名称，时段匹配的数据库集合A中的所有用户id，向这些用户执行发送邮件的云函数，并清除集合A，结束观察者状态。

主要问题3：

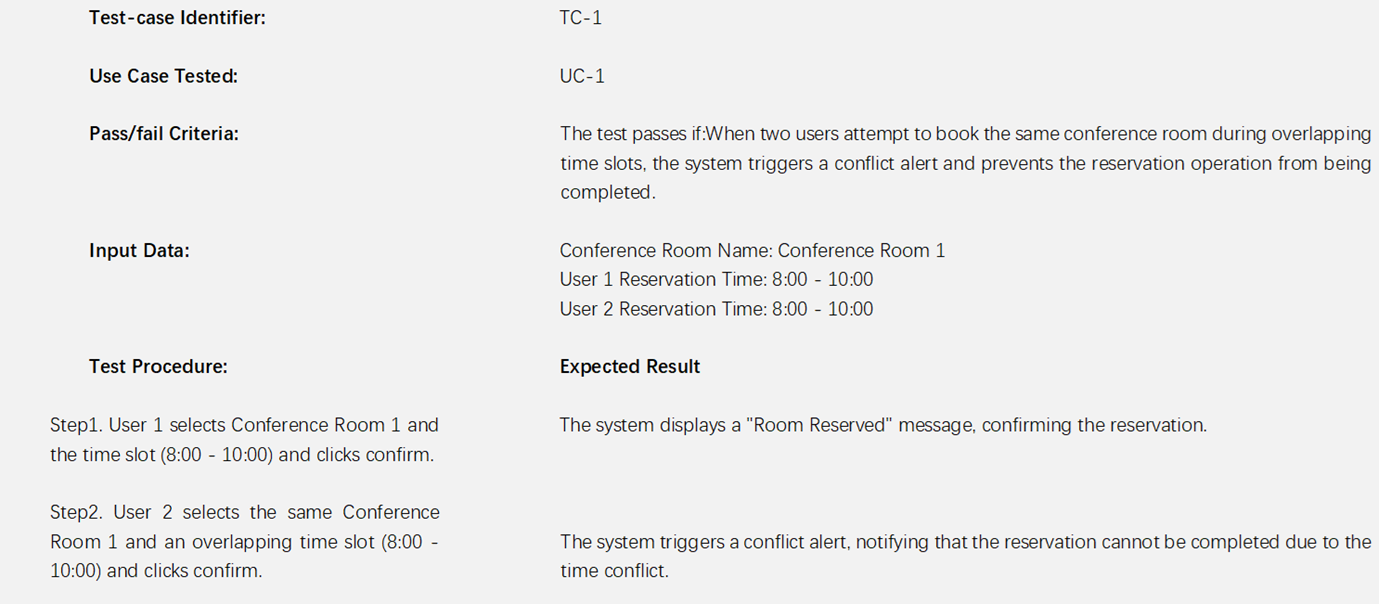
用户登录和身份验证：需验证用户的身份并提供登录渠道。

解决方案：使用微信小程序云函数功能，实现邮箱验证码的发送。

具体实现：配置node.js环境，获取发件人邮箱的授权码，配置云函数，在所需位置调用云函数。

1. **测试文档**

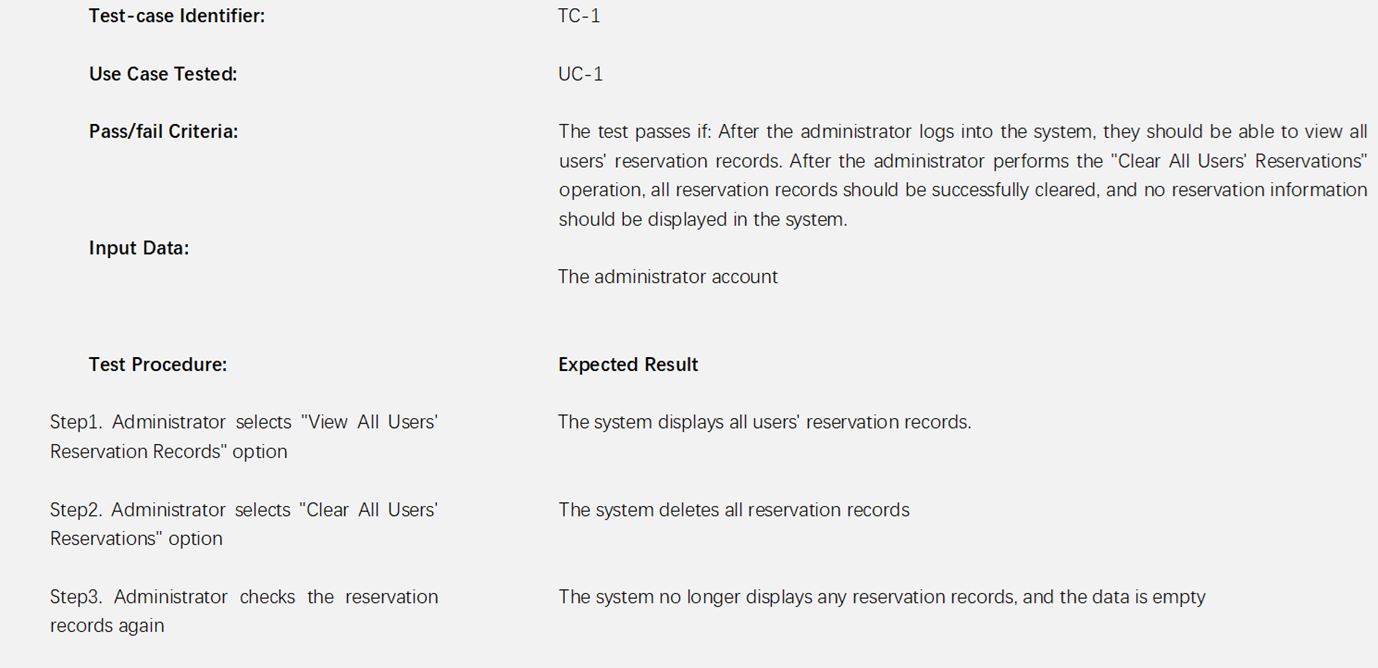
**用户测试用例图**

****

**测试目标：**验证系统是否能在用户预订冲突的情况下触发冲突警报并阻止第二次预订完成。**测试步骤：**第一步：用户1选择会议室1，并预订时间段为8:00 - 10:00，点击确认，系统需返回“预订成功”。第二步：用户2尝试预订同一会议室1，相同时间段（8:00 - 10:00），点击确认，系统需触发冲突警报。

**输入数据：**会议室名称：会议室1用户1和用户2的预订时间均为8:00 - 10:00。**预期结果：**用户1的预订成功，系统显示“预订成功”。用户2因时间冲突收到冲突警报，系统禁止预订。

**管理员测试用例图**

****

**测试目标：**验证系统是否能在用户预订冲突的情况下触发冲突警报并阻止第二次预订完成。**测试步骤：**第一步：用户1选择会议室1，并预订时间段为8:00 - 10:00，点击确认，系统需返回“预订成功”。第二步：用户2尝试预订同一会议室1，相同时间段（8:00 - 10:00），点击确认，系统需触发冲突警报。**输入数据：**会议室名称：会议室1用户1和用户2的预订时间均为8:00 - 10:00。**预期结果：**用户1的预订成功，系统显示“预订成功”。用户2因时间冲突收到冲突警报，系统禁止预订。

1. **项目总结**

经过一学期软件工程实践的学习，我们小组学会了标准的软件开发步骤，经过项目分析、项目设计、项目开发、项目测试等完整流程，成功开发了一款基于微信小程序的会议室预约系统，旨在为用户提供方便、高效的会议室预约管理服务。用户通过扫描二维码进入小程序，使用学校邮箱登录后，可以实现查看会议室预约情况、预约会议室和查看预约记录的功能。同时，系统支持多用户管理，具备冲突检测功能，确保用户在预约过程中不会因时间冲突导致资源浪费。此外，系统还提供订阅提醒功能，用户可以订阅已预约的房间，在会议室被取消或发生变动时接收到邮件通知，及时调整计划。管理员功能实现了管理功能，管理员可以通过特定账号登录系统，查看所有用户的预约记录，并通过“重置”功能一键清空所有预约记录，便于后期数据清理和维护。

本次开发从需求分析到测试上线，我们小组完成了一次完整的软件开发周期。通过该项目，我们深刻体会到了团队协作的重要性，在需求明确后，各成员分工明确，涵盖了前端设计、后端开发和测试验证等环节。技术上，我们熟练掌握了微信小程序开发的基本流程，同时在开发中积累了对用户交互设计、数据冲突检测和多用户管理的实践经验。尤其是在订阅提醒功能的实现过程中，我们结合邮件推送功能，成功实现了系统的自动化通知功能，不仅增强了项目的功能性和实用性，也提升了用户体验。

在开发过程中，最大的难点是订阅提醒功能的实现。初期方案中，我们尝试直接通过数据库的轮询来实现对预约变动的检测，但由于性能问题和实时性不理想，该方法被迫放弃。随后，我们学习并引入了“Observer模式”，设计了一套事件通知机制，当会议室预约信息发生变动时，系统会自动触发通知事件，并通过邮件推送功能将相关信息发送给订阅用户。虽然该功能的实现过程复杂，但最终效果令人满意，这也让我们对设计模式有了更深的理解。