

1. 引言（缪锴俊）

1.1. 背景

背景：随着网络技术的发展，计算机应用的日益普及和深化，各种软件，小程序使我们的生活更加方便快捷。为了满足学生们预约使用研讨室的使用需求，我们设计了一款在线研讨室预约小程序。该小程序能随时查看校园内的研讨室使用情况，并且按需预约，将极大的方便学生的学习。

项目目标：完成研讨室预约小程序的设计与实现，为实现项目的总目标可按以下三个阶段目标来进行：

第一阶段目标：实现研讨室预约小程序的基本功能，小组各个成员进行各个模块的开发，形成初步的系统。

第二阶段目标：攻克技术上的难题，实现研讨室预约小程序的一些特殊功能，进一步完善系统。

第三阶段目标：让系统投入到实际运用中，做好系统的维护工作

项目范围：

- (1) 能预约研讨室
- (2) 能查看研讨室的详尽情况
- (3) 能查看过往预约记录
- (4) 能对个人使用行为进行评分

1.2. 目的

本报告的主要作用是确定各个项目模块的开发情况和主要的负责人，供各项目模块的负责人阅读，做到及时协调，按步有序进行项目的开发。减少开发中的不必要损失

便于项目团队成员更好地了解项目情况，使项目工作开展的各个过程合理有序，因此以文件化的形式，把对于在项目生命周期内的工作任务范围、各项工作的任务分解、项目团队组织结构、各团队成员的工作责任、团队内外沟通协作方式、开发进度、经费预算、项目内外环境条件等内容做出的安排以书面的方式，作为项目团队成员以及项目干系人之间的共识与约定，项目生命周期内的所有项目活动的行动基础，项目团队开展和检查项目工作的依据。

2. 概要设计

2.1. 系统架构（刘文龙）

2.1.1. 架构概述

1. 前端：

- 使用微信小程序框架，包含多个页面：
 - ◆ 预约表单页面：用于用户选择日期、时间和课室，提交预约请求。
 - ◆ 课室浏览页面：显示可预约的课室列表，提供查看教室详情和进行预约的功能。
 - ◆ 其他可能的页面：如用户个人中心、预约历史等。
- 通过小程序框架实现前端逻辑，包括：
 - ◆ 处理用户输入，验证和格式化数据。
 - ◆ 调用微信小程序提供的 API，如获取用户信息、发送网络请求等。
 - ◆ 与后端通过 HTTP 协议进行通信，发送预约请求、接收后端返回的数据。

2. 后端

- 使用 Node.js 作为后端语言，并基于 Express.js 框架构建后端服务。
- 提供 RESTful API，包括但不限于：
 - ◆ 用户认证接口：处理用户登录、获取用户信息等。
 - ◆ 课室信息查询接口：返回可预约的课室列表。
 - ◆ 预约处理接口：处理用户提交的预约请求，检测时间冲突等。
 - ◆ 数据库操作接口：负责与数据库交互，执行相关的 CRUD 操作。
- 处理业务逻辑，包括：
 - ◆ 用户身份认知：验证用户的登录状态，确保只有授权用户可以进行预约。
 - ◆ 课室预约逻辑：检测预约时间是否冲突，更新数据库中的预约记录。
 - ◆ 数据库操作：使用 ORM（对象关系映射）或其他方式与 MySQL 数据库交互。

3. 数据库

- 使用 MySQL 关系型数据库，包括但不限于以下表：
 - ◆ 用户表(User)：存储用户信息，包括用户名、密码（加密存储）、权限等。
 - ◆ 课室表(Classroom)：存储课室信息，包括课室编号、容量等。
 - ◆ 预约表(Reservation)：记录用户的预约信息，包括预约时间、用户 ID、课室 ID 等。
- 表之间通过外键关联，确保数据的完整性和一致性。

4. 主要组件之间的关系

- 前端与后端：
 - 用户在小程序前端提交预约请求。
 - 小程序通过 HTTP 请求调用后端提供的 RESTful API，传递用户的预约信息。
 - 后端接收请求，处理业务逻辑，与数据库交互，最终返回结果给前端。
- 后端与数据库：

- 后端通过数据库连接池与 MySQL 数据库通信，执行数据库操作。
- 后端处理用户预约请求时，更新数据库中的预约表。
- 前端与数据库：
 - 前端通过调用后端提供的 API 与数据库间接交互。
 - 通过后端的 API，前端可以获取课室列表、用户信息等。
- 这样的体系结构允许各个组件专注于其核心任务，同时通过清晰定义的 API 实现了松耦合，使得系统更加灵活和可维护。

2.1.2. 技术栈

1. 前端

小程序框架：使用微信小程序框架 uni-app。

前端语言：使用小程序框架支持的前端语言——JavaScript

2. 后端

后端框架：Node.js 等

后端语言：JavaScript、Python

3. 数据库

关系数据库：适用于处理结构化数据，MySQL

4. 前后端通信

RESTful API：使用 RESTful 风格的 API 进行前后端通信，确保简单且有效的数据交换

5. 用户认证和安全

OpenID Connect：用于实现用户身份认证

HTTPS：确保数据在传输过程中的安全性。

JWT（JSON Web Tokens）：用于生成和验证用户身份信息。

6. 前端样式与交互

CSSS 框架：使用 CSS 框架，如 Bootstrap 样式库确保应用界面的一致性和美观

7. 实时通信

WevSocket：实现预约状态实时更新

8. 版本控制

Git：使用 Git 进行版本控制，有助于团队协作和代码管理

9. 开发工具

集成开发环境(IDE)：Intellij IDEA

2.2. 功能需求（胡茵）

2.2.1. 用例图

绘制系统的用例图，展示不同用户角色和系统的交互。

2.2.2. 主要功能

列出系统的主要功能和其简要描述。

1. 登录、密码修改以及用户中心功能：

学生的学号为唯一用户名，极大的方便了管理，防止非本校学生进入该小程序。通过安全的用户登录功能，系统保障了账户的安全性，同时也为管理员提供了便捷的用户账户管理手段。

2. 公告与留言功能：

该模块旨在有效地通知用户，分为公告和留言两部分。

公告部分：用于管理员向所有系统用户发送重要通知，如场地维护、停水停电等特殊情况。这有助于提前通知用户，减少不便之处。

留言部分：提供管理员与用户之间的交流渠道。例如，某个教室的桌椅坏了，那么用户可以通过留言反馈情况，以供管理员以及其它用户参考。

3. 预约功能：

预约功能划分为五部分，涵盖高校学生、教工等多个场景中的日常工作和生活需求。在 v1 版本中，已实现了教学楼、实验楼和研讨室的预约系统设计。图书馆和体育场地的预约系统将在 v2 版本中推出。这一功能旨在为用户提供便捷的场地预约服务，满足不同场景的需求。

① 教室预约：

在日常教学活动中，并非所有教室都被充分利用。为了优化教室资源的利用，使需要使用教室进行教学活动的教师和学生能够更方便地进行预约，本程序提供了教室预约功能。

该模块实现了基本的教室预约功能，包括但不限于人数选择、预约时间选择、刷卡签到、提前结束使用和次数限制等功能。

人数选择功能：根据不同规格的教室进行分类，如阶梯教室可容纳 200 人、大教室可容纳 100 人、中教室可容纳 60 人、小教室可容纳 30 人等。这有助于更好地规划教学活动所需的空間资源。

预约时间段选择功能：结合教学楼的日常授课安排和其他用户的预约情况，灵活调整用户可选的预约时间范围，更有效地规划教室利用时间。

刷卡签到功能：在预约时间段及其前 20 分钟内，预约者可使用学生卡或教工卡刷开被预约教室的门禁，提高教室的安全性，同时确保预约者的权益。

提前结束使用功能：允许用户在特殊情况下提前结束预约，结束本次教室使用，并取消刷卡签到权限。

次数限制功能：对每学期的预约次数进行限制，初步设置为 5 次，以确保教室资源被最需要的人充分利用。用户如有特殊需求，可在系统中向管理员提出申请。

程序将根据用户的预约情况管理教室资源，通过与教务处安排的时空统计对空闲教室进行管理，显示在预约系统中。系统会实时更新教室的空闲状态，以应对预约、教室维护等变化。

② 实验室预约:

为更好地满足学生和教职员工对实验室使用的需求,以及合理规划实验室时间,本系统引入了实验室预约功能。

该模块以提供全面而灵活的实验室管理为目标,实现了多项关键功能,包括实验室类型选择、预约时间调整、刷卡签到、提前结束使用和次数限制等。

实验室类型选择功能:通过基于实验室内实验器材的分类,满足实验者对不同实验需求的选择,使预约更符合具体实验要求。

预约时间段选择功能:结合实验楼的日常授课安排和其他用户的预约情况,智能调整用户可选的预约时间范围,最大程度地优化实验室使用时间。

刷卡签到功能:在预约时间段及其前 20 分钟内,预约者可使用学生卡或教工卡刷开被预约实验室的门禁,以提高实验室的安全性,并保障预约者的权益。

提前结束使用功能:允许用户在特殊情况下提前结束预约,取消本次实验室使用,并撤销刷卡签到权限,以更灵活地应对用户需求和实验情况变化。

次数限制功能:对每学期的实验室预约次数进行限制,初步设置为 5 次,以确保实验室资源被最需要的用户充分利用。用户如有特殊需求,可通过系统向管理员提出申请。

系统将根据用户的实验室预约情况精准管理实验室资源,实时更新实验室的空闲状态。用户对实验室的预约、实验器材的维护等操作都将迅速反映在系统中,以提高实验室资源的高效利用。

③ 研讨室预约:

在教师和学生的日常学习工作中,为促进交流与讨论,本系统特别引入了全面的研讨室预约功能。

该模块实现了丰富的研讨室预约功能,以满足不同需求,包括人数选择、预约时间选择、刷卡签到、时间限制、次数限制等。

人数选择功能:根据研讨室规格的不同,如小型研讨室可容纳 4-6 人、大型研讨室可容纳 8-10 人等,用户可根据讨论的规模选择合适的研讨室。

预约时间选择功能:灵活调整用户可选的预约时间范围,结合研讨室的日常使用情况,以更好地满足用户对研讨室时间规划的需求。

刷卡签到功能:预约者在预约时间段及其前 20 分钟内使用学生卡或教工卡刷卡签到,确保预约者有权使用研讨室,同时提高研讨室的安全性。

时间限制功能:设定研讨室使用的最大时间,最大时间为两个小时,以确保研讨室能够有效轮换使用,从而更好地服务不同用户的需求。以及确保同一个教室的下一个预约者开始时间与上一个预约者结束时间相隔半个小时以上。

次数限制功能:对每学期的研讨室预约次数进行限制,初步设置为 5 次,以确保资源公平分配。用户可在系统中向管理员提出申请以满足特殊需求。

该程序将根据用户的预约情况智能管理研讨室资源,实时更新研讨室的空闲状态。用户对研讨室的预约和使用情况的变化都将即时反映在系统中,提供更灵活、高效的研讨室资源利用。

④ 图书馆座位预约(暂未上线):

为满足学生学习和阅读的需求,系统将引入图书馆预约功能。

该模块旨在提供便捷的图书馆座位预约服务，包括但不限于座位选择、预约时间、取消预约、次数限制等功能。

座位选择功能：根据图书馆内座位的布局 and 特性，用户可以根据个人偏好选择合适的座位，如安静区、采光好的位置等。

预约时间功能：用户可以根据自己的学习计划选择预约的时间段，以确保在繁忙的学习时段能够获得合适的座位。

取消预约功能：允许用户在需要的情况下取消已经预约的座位，以便其他用户能够利用这些资源。

该模块将实现对图书馆座位的智能管理，及时更新座位的预约状态，为用户提供高效、便捷的图书馆学习环境。

⑤ 体育场地预约(暂未上线)

以满足学生和教职员工的体育锻炼需求，系统还将推出体育场地预约功能。

该模块将实现对不同体育场地的预约，包括篮球场、足球场、羽毛球场等，以便用户可以有序地利用校园体育资源。

场地选择功能：用户可以根据自己的锻炼需求选择合适的体育场地，确保场地的有效利用。

预约时间功能：用户可根据个人日程选择适合的锻炼时间段，确保在繁忙时段能够获得体育场地的使用权限。

取消预约功能：用户有权在需要的情况下取消已经预约的体育场地，以便其他用户能够合理利用。

场地付费功能：对不同类型的体育场地收取对应的费用，确保场地的维护费用充足。

该模块将实现对体育场地的智能管理，及时更新场地的预约状态，为用户提供便捷、高效的体育锻炼场地。

4. 管理员管理功能：

① 消息管理功能：

管理员可以查询、添加、修改和删除公告：

添加公告：管理员可以添加新的公告，填写标题、内容、发布日期等信息。

修改公告：允许管理员对现有公告进行修改，包括标题、内容和发布日期等。

删除公告：管理员可以删除不再需要的公告，确保公告信息的及时更新和管理。

以上功能能够确保公告的及时性和准确性，使用户能够方便地获取重要信息，并保持对系统变更或重要通知的了解。

管理员可以查看用户间留言，以确保用户不受到不当留言的影响。

留言审核功能：允许管理员查看所有留言，包括待审核和已审核的留言，确保留言内容符合规范。

留言状态标记：管理员可以对留言进行标记，标识留言的审核状态，例如待审、已审、已回复等。

留言查看功能：管理员可以查看完整的留言内容和相关信息。

删除留言功能：管理员可以删除不适当或违规的留言，确保留言内容的合规性。

留言给用户功能：允许管理员留言给某些用户，以便用户可以看到相关的信息或回复。

② 用户管理功能：

管理员对用户的账号信息进行管理，拥有权限对存在违规、爽约等其他不良行为的用户加入黑名单，同时拥有对用户的预约记录进行查询、修改、删除等权限。

黑名单管理：允许管理员将特定用户列入黑名单，禁止其进行预约或使用系统特定功能。

账号查询：管理员可以查询已有用户的账号信息，包括用户名、基本信息等，但不具备新增或删除学生账号的权限。

预约记录修改：管理员可以对用户的预约记录进行修改，如调整预约时间、场地等。

爽约记录更改：允许管理员修改用户的爽约记录，以确保记录的准确性。

③ 场地资源管理功能：

场地信息查看：允许管理员查看所有场地的详细信息，包括名称、类型、可预约时间等。

场地信息编辑：管理员可以对场地信息进行编辑，包括场地名称、类型、可用时间段等内容。

预约规则设置：允许管理员设定场地的预约规则，如最大预约时间、最大预约人数等。

新增场地：管理员可以添加新的场地信息到系统中，包括场地名称、类型、可预约时间段等。

删除场地：允许管理员删除不再使用的场地信息，确保系统中的场地信息及时更新。本模块将提供管理员对场地资源的全面管理功能，包括场地信息的查看、编辑、新增和删除，以及预约规则的设置，确保场地资源信息的准确性和系统的高效性。

2.3. 数据模型（廖乐乐，马乙佳，丁妍婷，李晓行）

2.3.1. 数据库设计（描述数据库结构，包括表格、字段和关系）

● 我们可以实现以下功能：

- 用户可以注册并登录系统，通过用户表记录用户信息。
- 用户可以预约研讨室，通过预约记录表记录预约信息，并关联到对应的用户。
- 系统可以记录用户的操作日志，通过日志表记录用户的操作信息，并关联到对应的用户。
- 系统可以管理房间信息，通过房间表记录房间的相关信息。
- 预约记录表中的房间号字段和时间段字段，可以帮助系统判断房间在某个时间段是否已被预约。

时间段表记录了系统定义的可预约时间段，便于用户选择合适的时间段进行预约。

● 数据库各表之间的关系：

- 用户表（users）与预约记录表（seminar_booking）有一对多的关系。一个用户可以创建多个预约记录，而一个预约记录只能属于一个用户。

- b. 房间表 (rooms) 与预约记录表 (seminar_booking) 之间存在一对多的关系。一个房间可以有多个预约记录, 而一个预约记录只能属于一个房间。因此, 在预约记录表中添加了房间号作为字段, 记录预约的房间号。
- c. 时间段表 (time_slots) 可以被多个预约记录引用。每个预约记录都会有一个对应的时间段, 因此, 在预约记录表中添加了开始时间和结束时间字段, 用于记录预约的时间段。
- e. 用户表 (users) 与研讨室历史记录表 (seminar_room_history) 有一对多的关系。一个用户可以创建多个预约历史记录, 而一个历史预约记录只能属于一个用户。
- f. 房间表 (rooms) 与研讨室历史记录表 (seminar_room_history) 之间存在一对多的关系。一个房间可以有多个历史预约记录, 而一个历史预约记录只能属于一个房间。因此, 在历史预约记录表中添加了房间号作为字段, 记录预约的房间号。
- g. 时间段表 (time_slots) 可以被多个研讨室历史记录 (seminar_room_history) 引用。每个预约记录都会有一个对应的时间段, 因此, 在预约记录表中添加了开始时间和结束时间字段, 用于记录预约的时间段。
- h. 用户表和公告表有一对多的关系。一个管理员用户可以发布多条公告。
- i. 用户表 (users) 与日志表 (logs) 也有一对多的关系。一个用户可以执行多个操作日志, 而一个操作日志只能属于一个用户。

表名	作用	详细列名
users	存储用户信息	用户 ID、用户名、密码、用户类型、盐值、创建时间、修改时间、创建人 ID、修改人 ID
seminar_booking	记录研讨室预约信息	预约 ID、用户 ID、房间号、预约日期、开始时间、结束时间、创建时间、修改时间、创建人 ID、修改人 ID
seminar_room_history	记录研讨室历史预约信息	历史记录 ID、用户 ID、房间号、预约日期、开始时间、结束时间、创建时间、修改时间、创建人 ID、修改人 ID
log	存储系统操作日志	日志 ID、操作用户 ID、操作时间、操作类型、操作描述
rooms	存储房间信息	房间 ID、房间号、容纳人数、房间类型、房间状态、创建时间、修改时间、创建人 ID、修改人 ID
time_slots	存储时间段信息	时间段 ID、开始时间、结束时间、创建时间、修改时间、创建人 ID、修改人 ID
announcement	存储公告信息	公告 ID、标题、内容、创建时间、修改时间、创建人 ID、修改人 ID
message	存储留言信息	留言 ID、内容、是否展示、创建时间、修改时间、创建人 ID、修改人 ID
blacklist	存储黑名单用户信息	黑名单 ID、用户 ID、加入时间、原因、创建时间、修改时间、创建人 ID、修改人 ID

表-1 数据库设计概要信息表

2.4. 用户界面设计（张文秀，张雨馨）

2.4.1. 用户界面原型

提供用户界面原型示例，以展示界面的布局 and 元素。

2.5. 性能和可伸缩性（胡茵）

描述系统的性能要求和可伸缩性策略。

3. 详细设计

3.1. 模块设计

3.1.1. 模块概述（李晓行）

1. 登陆模块：



用户打开小程序时，页面进入登陆界面。账号输入框与密码输入框为**必填**内容，否则用户将无法登陆小程序。**账号和密码都可以是数字，下滑线（英文版本，中文版本不允许），大小写字母，不允许出现其他字符。**登陆该小程序的用户都为拥有汕大校园账号的师生，外校人员无法

登陆，没有注册选项。登陆该小程序的用户不可以进行忘记密码操作，如需修改密码，需要在学校专有修改网页进行修改。用户的账号和密码都是正确的才允许进入首页，如果账号错误或者密码错误，提示“账号和密码信息有误，请重新输入”。同时账号和密码正确后，还需要用户勾选“点击阅读并同意《隐私协议》”，才允许跳转至首页界面，否则提示用户“请勾选点击阅读并同意《隐私协议》”。完成登陆操作后，小程序跳转至首页界面。

2.首页模块：



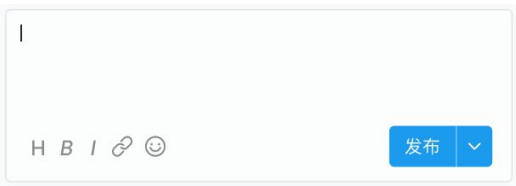
① 用户可以通过两种方式跳转到首页：用户在成功登录之后会直接跳转到首页；

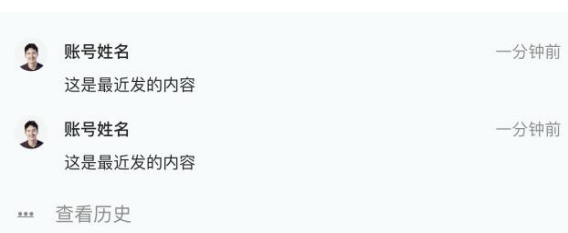
或者当用户最下方导航栏的  时可以从其他页面跳转到首页。




② 这里展示的是公告信息，按时间排序，最晚发布的公告排在最上面，用户可以通过上下滑动来阅读更多的公告。每条公告由两部分内容组成，第一部分是公告的标题，第二部分是公告的具体内容，当标题或

者具体内容因字数太多而不能完全展示时，只展示一部分文字，最后用省略号表示。当用户点击公告的标题或者是内容时，可以进入公告详情页面查看对应公告的全部信息。


- ③
- 
- 用户在此处输入文字/链接/表情，点击发布即可发表留言。（H、B、I用于调节字体的格式，H表示高亮，B表示加粗，I表示斜体，是否实现此功能待定）

- ④
- 
- 此处展示历史留言信息，仅展示发布时间最晚的两条留言，每条留言展示用户的头像、用户的账号姓名、留言内容和发布时间。用户点击“查看历史”，可以展示所有的留言，用户通过上下滑动来阅读留言。

- ⑤
- 

预约

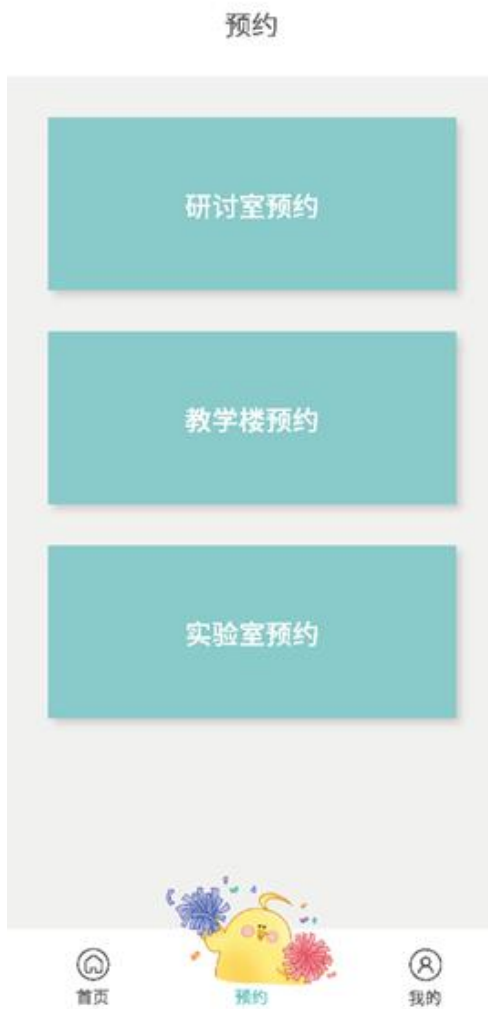
时，可以跳转到预约界面；



我的

时，可以跳转到我的界面。

3.预约模块：



用户点击”预约“按钮，小程序跳转至预约界面，该界面从上到下为：研讨室预约、教学楼预约、实验室预约三个部分。用户可以根据需求选择自己需要预约的类型。

研讨室预约

4人研讨室



6人研讨室



10人研讨室



用户点击研讨室预约跳转至研讨室预约界面，用户需要进行研讨室大小以及研讨室编号的选择。

研讨室按照可容纳人数分为 4 人、6 人、10 人三类，通过左右滑动可以查看对应类别下更多的研讨室。用户可以根据需要选择使用研讨室的人数，选择所需的研讨室，直接点击对应研讨室的图标即可查看对应的研讨室详情。比如当用户点击“101”，小程序跳转至下图的界面。

研讨会预约

Room 101

202307月

202308月

202309月

202310月

202311月

202312月

周日

周一

周二

周三

周四

周五

周六

12

13

14

15

16

17

18

有位

有位

有位

有位

有位

有位

有位

请选择当日预约开始时间段

剩余 7

8:00

有位

9:00

我的预约时间

10:00

约满

11:00

约满

13:00

有位

14:00

约满

15:00

约满

16:00

有位

18:00

约满

19:00

约满

20:00

约满

21:00

有位

○ 预约人数

4人

○ 预约时间

9:00-10:00

○ 预约人

XXX

首页

预约

我的

用户点击研讨室后进入上图所示界面，界面显示当前用户选择的房间号，可选预约时间日期，界面最下方可查看当前房间容量等等。用户可以根据最上方的日期栏选择需要预约的月份及日期，点击对应日期的数字，用户即可选中并查看当天的研讨室预约情况。仅可以预约今天到 7 天后的预约(如今天为 1 号，则用户最多可以预约 1 号-7 号的研讨室)，

请选择当日预约开始时间段

剩余 7

8:00

有位

9:00

我的预约时间

10:00

约满

11:00

约满

13:00

有位

14:00

约满

15:00

约满

16:00

有位

18:00

约满

19:00

约满

20:00

约满

21:00

有位

如果用户选择不在这个范围的日期则清空，并在此显示“当前日期不可以预约”。

用户选中日期（如 12 月 11 日）之后，可以选择有空缺的时间段，该时间区按照小时

为最小单位，显示每个时间段是否已经被预约。如果某个小时段没有空缺，则该时间下方会显示“约满”，若该时间段还有空缺位置，则界面中显示“有位”。（用户可以在该区域右上角，**查看当前研讨室的剩余时间段容量**。建议删除该功能，理由：这个功能对于我们的小程序重要性不大，且实施逻辑困难，需要对已经预约的时间段进行统计，统计没有预约了的时间段，需要对每个房间，12 个时间段都有记录，与我们当前开发状况不符。）

用户点击任意的开始时间段按钮，确定预约使用人数，并选择所需的预约时间段（如 9:00-10:00），例如用户在上一个界面选择了九点，在弹出的时间选择框的开始时间可以选择九点或者九点半，在点击预约时间之后，如下图所示，用户需在弹出的窗口滚动中选择具体的预约时间段，确定开始时间以及结束时间，并点击“**确认**”按钮，确认选中该时间段，并**确定**预约该研讨室。在此过程中，如果用户想更改其他的预约设置，可以点击“**取消**”按钮，重新回到日期选择的界面。预约开始时间和预约结束时间不可以相等，且预约开始时间要早于预约结束时间至少半个小时，预约时间段不能超过 2 个小时（即预约时间段为半个小时至 2 个小时），用户与用户的预约时间至少要间隔半个小时才允许预约，如 A 已经预约 9:00-10:00，则 B 如果要预约的话，至少要从 10:30 开始预约，预约时间可以为 10:30-11:00，但是不可以为 10:00-11:00（用户与用户的预约时间至少要间隔半个小时）。



用户按照以上预约流程操作之后，系统会根据预约的情况给予用户反馈，若用户预约成功，则如下图显示预约成功弹窗。至此，用户完成了对研讨室的预约。



4.我的模块：



“我的”模块，主要为用户提供个人信息查看管理功能，用户点击页面右下角“我的”进入“我

的”页面，展示用户账号信息，提供查看用户个人预约信息、留言反馈、联系客服、设置修改。当用户点击“我的预约”将跳转到用户预约记录界面，当用户点击“留言反馈”，跳转到“留言”输入界面，当用户点击“联系客服”将跳转到客服联系方式界面，当用户点击“设置”将跳转到设置界面。

5.管理员模块

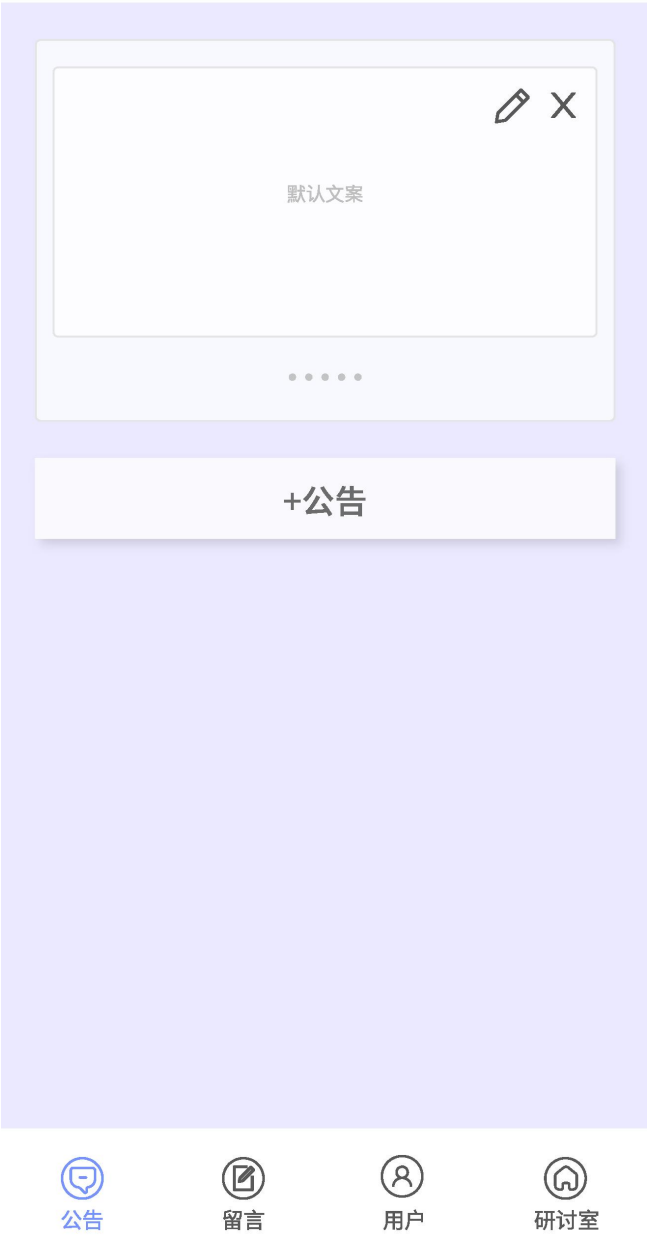


The image shows a mockup of an admin login interface. It features a light purple header with the text '欢迎登录' (Welcome to Login). Below the header, there are two input fields: '管理员账号' (Admin Account) and '密码' (Password). The '管理员账号' field has a placeholder text '请输入管理员账号' (Please enter admin account). The '密码' field has a placeholder text '请输入密码' (Please enter password) and a toggle icon for password visibility. Below the password field, there is a checkbox labeled '点击阅读并同意《隐私协议》' (Click to read and agree to the Privacy Policy). At the bottom, there is a large purple button labeled '登录' (Login) and a link labeled '忘记密码' (Forgot Password).

管理员打开管理员小程序时，页面进入登陆界面。账号输入框与密码输入框为**必填**内容，否则管理员将无法登陆小程序。**账号和密码都可以是数字，下滑线（英文版本，中文版本不允许），大小写字母，不允许出现其他字符。**登陆该小程序的用户都为拥有汕大校园账号的管理员，外校人员无法登陆，**没有注册选项**。登陆该小程序的用户**不可以**进行忘记密码操作，如需修改密码，需要在学校专有修改网页进行修改。用户的账号和密码都是正确的才允许进入首页，如果

账号错误**或者**密码错误，提示“账号和密码信息有误，请重新输入”。同时账号和密码正确后，还需要用户**勾选**“点击阅读并同意《隐私协议》”，才允许跳转至首页界面，否则提示用户“请勾选点击阅读并同意《隐私协议》”。完成登陆操作后，小程序跳转至首页界面。

公告管理



管理员进入小程序之后，点击



可以进入公告管理页面，在文本框中输入需要添加




的公告内容，点击

+公告

按钮即可向全体用户发送公告。

留言管理



管理员点击  即可进入留言管理界面，该界面会显示用户发送的留言，管理员可以查留言内容并对其进行审核，如留言存在内容违规、不友善的内容，可以点击  按钮对其进行删除，点击  ，管理员可以对留言进行回复。

3.1.2. 接口设计 （廖乐乐，马乙佳，丁妍婷，李晓行）

用户相关接口：

1. POST 登录验证密码

Body 请求参数

```
{
  "username": "2020221001",
  "password": "123456"
}
```

请求示例：

名称	位置	类型	必选	中文名	说明
body	body	object	否		none
username	body	string	是	学号	none
password	body	string	是	密码	none

返回示例
成功

```
{
  "code": 201, "msg": "登录成功",
  "data": ""
}
```

失败

```
{
  "code": 401, "msg": "登录失败",
  "data": ""
}
```

返回结果：

状态码	状态码含义	说明
200	OK	成功
404	Not Found	失败

返回数据结构：

状态码 200

名称	类型	必选	约束	中文名	说明
code	integer	true	none		状态码为200
data	string	false	none		none
msg	string	false	none		none

状态码 200

名称	类型	必选	约束	中文名	说明
----	----	----	----	-----	----

code	number	true	none		状态码为400
data	string	false	none		none
msg	string	false	none		none

2. GET 查询所有公告

GET /notice

返回示例:

成功:

```
{
  "code": 202,
  "data": [
    {
      "announcementId": 1,
      "createTime": "2023-10-25 16:24:59",
      "content": "1. 在规定时间内签到 2. xxx", "title": " 研 讨 室 预 约 注 意 事 项 ",
      "createUserId": 1,
      "updateTime": "2023-10-25 16:24:59",
      "updateUserId": 1
    },
    {
      "announcementId": 2,
      "createTime": "2023-08-18 20:25:46",
      "content": "每天8: 00-22: 00", "title": " 研 讨 室 开 放 时 间 ", "createUserId": 1,
      "updateTime": "2023-08-18 20:25:46",
      "updateUserId": 1
    }
  ],
  "msg": "查询公告成功"
}
```

```
{
  "code": 402,
  "msg": "查询公告失败",
  "data": ""
}
```

失败:

返回结果:

状态码	状态码含义	说明
200	OK	成功
404	Not Found	失败

返回数据结构：

状态码 200

名称	类型	必选	约束	中文名	说明
code	integer	true	none		状态码为 201
data	[object]	true	none		none
announcementId	integer	true	none		none
title	string	true	none		none
content	string	true	none		none

名称	类型	必选	约束	中文名	说明
createTime	string	true	none		none
updateTime	string	true	none		none
createUserId	integer	true	none		none

updateUserId	integer	true	none		none
msg	string	false	none		none

状态码 200

名称	类型	必选	约束	中文名	说明
code	number	true	none		状态码为401
data	string	false	none		none
msg	string	false	none		none

3. GET 查询我的所有预约

GET /seminar/appointments/{seminar_user_id}

请求参数

名称	位置	类型	必选	中文名	说明
seminar_user_id	path	string	是		none

返回示例:

成功:

```
{
  "code": "203", "data": [
    {
      "seminar_user_id": "40",
      "seminar_room_number": 89,
      "start_time": 1,
      "end_time": 4
    },
    {
      "seminar_user_id": "40",
      "seminar_room_number": 9,
      "start_time": 6,
      "end_time": 8
    },
    {
      "seminar_user_id": "40",
      "seminar_room_number": 1,
      "start_time": 10,
      "end_time": 12
    }
  ],
  "msg": "成功"
```

失败:

```
{
  "code": "403",
  "data": "",
  "msg": "查看我的预约失败"
}
```

返回结果：

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline
404	Not Found	失败	Inline

返回数据结构：

状态码 200

名称	类型	必选	约束	中文名	说明
code	string	true	none		状态码为202
data	[object]	true	none		none
seminar_user_id	string	true	none		预约人
seminar_room_number	number	true	none		none
seminar_date	string	true	none		none
start_time	number	true	none		none
end_time	number	true	none		none
msg	string	false	none		查询成功

状态码 200

名称	类型	必选	约束	中文名	说明
code	string	true	none		404
msg	string	false	none		请求参数错误或缺失，请检查并重新发送请求
data	string	false	none		none

4. POST 是否可以预约

POST /seminar/appointments

Body 请求参数

```
{
  "seminarUserId": 12,
  "seminarRoomNumber": 1,
  "seminarDate": "2023-09-10",
  "startTime": 1,
  "endTime": 3
}
```

请求参数：

名称	位置	类型	必选	中文名	说明
body	body	object	否		none
seminarUserId	body	integer	是	用户id	none
seminarRoomNumber	body	integer	是	房间号	none
seminarDate	body	string	是	预约日期	none
startTime	body	integer	是	开始时间对应的时间段id	none
» endTime	body	integer	是	结束时间对应的时间段id	none

返回示例：

成功：

```
{
  "code": 204,
  "msg": "预约成功",
  "data": ""
}
```

失败：

```
{
  "code": 404,
  "msg": "预约失败",
  "data": ""
}
```

返回结果：

状态码	状态码含义	说明	数据模型
200	OK	成功	Inline
404	Not Found	失败	Inline

返回数据结构：

状态码 200

名称	类型	必选	约束	中文名	说明
code	integer	true	none		状态码为203
data	string	false	none		none
msg	string	false	none		none

状态码 200

名称	类型	必选	约束	中文名	说明
code	number	true	none		状态码为403

data	string	false	none		none
msg	string	false	none		none

5. DELETE 删除预约

DELETE /seminar/appointments/{seminar_book_id}

请求参数：

名称	位置	类型	必选	中文名	说明
seminar_book_id	path	integer	是		none

返回示例：

成功：

```
{
  "code": 205,
  "data": "",
  "msg": "成功删除"
}
```

失败：

```
{
  "code": 405,
  "data": "",
  "msg": "删除失败"
}
```

返回结果：

状态码	状态码含义	说明
200	OK	成功
404	Not Found	失败

返回数据结构：

状态码 200

名称	类型	必选	约束	中文名	说明
code	integer	true	none		状态码为204
msg	string	false	none		none
data	string	false	none		none

状态码 200

名称	类型	必选	约束	中文名	说明
code	number	true	none		状态码为404
data	string	false	none		none
msg	string	false	none		none

3.2. 数据库设计（廖乐乐，马乙佳，丁妍婷，李晓行）

3.2.1. 表格和字段定义

用户表

详细列名：

- 用户 ID (user_id)：整型，主键，自增。
- 用户名 (username)：字符串，长度 50，唯一索引。
- 密码 (password)：字符串，长度 50。
- 用户类型 (user_type)：小整型，表示用户类型。
- 盐值 (usersalt)：字符串，长度 255。
- 创建时间 (create_time)：时间戳，默认值为当前时间。
- 修改时间 (update_time)：时间戳，默认值为当前时间。
- 创建人 ID (create_user_id)：整型。
- 修改人 ID (update_user_id)：整型。

名	类型	长度	非空	唯一	主键	约束
用户 ID	整型 (INT)		是	是	是	自增
用户名	字符串 (VARCHAR)	50	是	是		唯一索引
密码	字符串 (VARCHAR)	50	是			
用户类型	小整型 (SMALLINT)	1				默认值 1 (普通用户)
盐值	字符串 (VARCHAR)	255	是			
创建时间	时间戳 (TIMESTAMP)					默认值 CURRENT_TIMESTAMP
修改时间	时间戳 (TIMESTAMP)					默认值 CURRENT_TIMESTAMP
创建人 ID	整型 (INT)					
修改人 ID	整型 (INT)					

研讨室预约记录表(seminar_booking)

详细列名：

- 预约 ID (seminar_book_id)：整型，主键，自增。
- 用户 ID (seminar_user_id)：整型，外键关联用户表的用户 ID。
- 房间号 (seminar_room_number)：字符串，长度 8。
- 预约日期 (seminar_date)：日期类型。
- 开始时间 (start_time)：外键，指向时间段表的 ID。
- 结束时间 (end_time)：外键，指向时间段表的 ID。
- 创建时间 (create_time)：时间戳，默认为当前时间。
- 修改时间 (update_time)：时间戳，默认为当前时间。
- 创建人 ID (create_user_id)：整型。
- 修改人 ID (update_user_id)：整型。

名	类型	长度	非空	唯一	主键	约束
预约 ID	整型 (INT)		是	是	是	自增
用户 ID	整型 (INT)		是			外键关联用户表的用户 ID
房间号	字符串 (VARCHAR)	8	是			范围 (101-105)
预约日期	日期 (DATE)		是			
开始时间	外键		是			外键关联时间段表 id
结束时间	外键		是			外键关联时间段表 id
创建时间	时间戳 (TIMESTAMP)					默认值 CURRENT_TIMESTAMP
修改时间	时间戳 (TIMESTAMP)					默认值 CURRENT_TIMESTAMP
创建人 ID	整型 (INT)		是			
修改人 ID	整型 (INT)		是			

研讨室历史记录表 (seminar_room_history)

详细列名：

- 历史记录 ID (seminar_book_id_history)：整型，主键，自增。
- 用户 ID (seminar_user_id_history)：整型，外键关联用户表的用户 ID。
- 房间号 (seminar_room_number_history)：字符串，长度 50。
- 预约日期 (seminar_date_history)：日期类型。
- 开始时间 (start_time_history)：外键，指向时间段表的 ID。
- 结束时间 (end_time_history)：外键，指向时间段表的 ID。
- 预约记录创建时间 (create_time_history)：时间戳，默认为当前时间。
- 预约记录修改时间 (update_time_history)：时间戳，默认为当前时间。
- 预约记录创建人 ID (create_user_id_history)：整型。
- 预约记录修改人 ID (update_user_id_history)：整型。

名	类型	长度	非空	唯一	主键	约束
---	----	----	----	----	----	----

名	类型	长度	非空	唯一	主键	约束
历史记录 ID	整型 (INT)		是	是	是	自增
用户 ID	整型 (INT)		是			外键关联用户表的用户 ID
房间号	字符串 (VARCHAR)	50	是			范围 (101-105)
预约日期	日期 (DATE)		是			
开始时间	外键		是			外键关联时间段表 id
结束时间	外键		是			外键关联时间段表 id
预约记录创建时间	时间戳 (TIMESTAMP)					默认值 CURRENT_TIMESTAMP
预约记录修改时间	时间戳 (TIMESTAMP)					默认值 CURRENT_TIMESTAMP
预约记录创建人 ID	整型 (INT)		是			
预约记录修改人 ID	整型 (INT)		是			

系统操作日志表 (log)

详细列名：

- 日志 ID (log_id)：整型，主键，自增。
- 操作用户 ID (user_id)：整型，外键关联用户表的用户 ID。
- 操作时间 (operation_time)：时间戳，默认为当前时间。
- 操作类型 (operation_type)：字符串，长度 50。
- 操作描述 (operation_description)：字符串，长度 255。

名	类型	长度	非空	唯一	主键	约束
日志 ID	整型 (INT)		是	是	是	自增
操作用户 ID	整型 (INT)		是			外键关联用户表的用户 ID
操作时间	时间戳 (TIMESTAMP)					默认值 CURRENT_TIMESTAMP

名	类型	长度	非空	唯一	主键	约束
操作类型	字符串 (VARCHAR)	50				
操作描述	字符串 (VARCHAR)	255				

房间信息表 (rooms)

详细列名：

- 房间 ID (room_id)：整型，主键，自增。
- 房间号 (room_number)：整型。
- 容纳人数 (room_people)：小整型，范围 1-105。
- 房间类型 (room_type)：小整型，表示房间类型。
- 房间状态 (room_state)：小整型，表示房间状态。
- 创建时间 (create_time)：时间戳，默认为当前时间。
- 修改时间 (update_time)：时间戳，默认为当前时间。
- 创建人 ID (create_user_id)：整型。
- 修改人 ID (update_user_id)：整型。

名	类型	长度	非空	唯一	主键	约束
房间 ID	整型 (INT)		是	是	是	自增
房间号	整型 (INT)		是			范围 (101-105)
容纳人数	小整型 (SMALLINT)		是			范围 (1-105)
房间类型	小整型 (SMALLINT)		是			范围 (1-3)
房间状态	小整型 (SMALLINT)		是			范围 (0-1)
创建时间	时间戳 (TIMESTAMP)					默认值 CURRENT_TIMESTAMP
修改时间	时间戳 (TIMESTAMP)					默认值 CURRENT_TIMESTAMP
创建人 ID	整型 (INT)		是			
修改人	整型 (INT)		是			

名	类型	长度	非空	唯一	主键	约束
ID						

时间段表 (time_slots)

详细列名:

- 时间段 ID (time_slot_id): 整型, 主键, 自增。
- 开始时间 (start_time): 时间类型。
- 结束时间 (end_time): 时间类型。
- 创建时间 (create_time): 时间戳, 默认为当前时间。
- 修改时间 (update_time): 时间戳, 默认为当前时间。
- 创建人 ID (create_user_id): 整型。
- 修改人 ID (update_user_id): 整型。

名	类型	长度	非空	唯一	主键	约束
时间段 ID	整型 (INT)		是	是	是	自增
开始时间	时间类型 (TIME)		是			
结束时间	时间类型 (TIME)		是			
创建时间	时间戳 (TIMESTAMP)					默认值 CURRENT_TIMESTAMP
修改时间	时间戳 (TIMESTAMP)					默认值 CURRENT_TIMESTAMP
创建人 ID	整型 (INT)		是			
修改人 ID	整型 (INT)		是			

公告表 (announcements)

详细列名:

- 公告 ID (announcement_id): 整型, 唯一标识每个公告的编号。
- 标题 (title): 字符串, 长度 20。
- 内容 (content): 字符串, 长度 1000。
- 创建时间 (create_time): 时间戳, 默认为当前时间。

修改时间 (update_time): 时间戳，默认为当前时间。

创建人 ID (create_user_id): 整型。

修改人 ID (update_user_id): 整型。

名	类型	长度	非空	唯一	主键	约束
公告 ID	整型 (INT)		是	是	是	唯一标识，自增
标题	字符串 (VARCHAR)	20	是			
内容	字符串 (VARCHAR)	1000	是			
创建时间	时间戳 (TIMESTAMP)					默认值 CURRENT_TIMESTAMP
修改时间	时间戳 (TIMESTAMP)					默认值 CURRENT_TIMESTAMP
创建人 ID	整型 (INT)		是			
修改人 ID	整型 (INT)		是			

留言表 (messages)

详细列名:

留言 ID (message_id): 整型，唯一标识每个留言的编号。

内容 (content): 字符串，记录留言文本内容。

是否展示 (show): 布尔类型，表示留言是否展示到用户界面的留言列表中。

创建时间 (create_time): 时间戳，默认为当前时间。

修改时间 (update_time): 时间戳，默认为当前时间。

创建人 ID (create_user_id): 整型。

修改人 ID (update_user_id): 整型。

名	类型	长度	非空	唯一	主键	约束
留言 ID	整型 (INT)		是	是	是	唯一标识，自增
内容	字符串 (VARCHAR)		是			
是否展示	布尔 (BOOL)					

名	类型	长度	非空	唯一	主键	约束
创建时间	时间戳(TIMESTAMP)					默认值 CURRENT_TIMESTAMP
修改时间	时间戳(TIMESTAMP)					默认值 CURRENT_TIMESTAMP
创建人ID	整型 (INT)		是			
修改人ID	整型 (INT)					

黑名单表 (blacklist)

详细列名：

- 黑名单 ID (blacklist_id)：整型，唯一标识每个黑名单项的编号。
- 用户 ID (user_id)：整型，被加入黑名单的用户的唯一标识。
- 加入时间 (blacklist_time)：时间戳，记录用户被加入黑名单的时间。
- 原因 (reason)：字符串，记录加入黑名单的原因。
- 创建时间 (create_time)：时间戳，默认为当前时间。
- 修改时间 (update_time)：时间戳，默认为当前时间。
- 创建人 ID (create_user_id)：整型。
- 修改人 ID (update_user_id)：整型。

名	类型	长度	非空	唯一	主键	约束
黑名单ID	整型 (INT)		是	是	是	唯一标识，自增
用户ID	整型 (INT)		是			被加入黑名单用户的唯一标识
加入时间	时间戳 (TIMESTAMP)		是			
原因	字符串 (VARCHAR)		是			
创建时间	时间戳 (TIMESTAMP)					默认值 CURRENT_TIMESTAMP
修改时间	时间戳 (TIMESTAMP)					默认值 CURRENT_TIMESTAMP
创建	整型 (INT)		是			

名	类型	长度	非空	唯一	主键	约束
人 ID						
修改人 ID	整型 (INT)					

3.3. 算法和逻辑设计

3.3.1. 算法描述

解释系统中使用的关键算法和逻辑。

3.4. 用户界面设计（张文秀）

3.4.1. 界面元素

详细描述每个用户界面元素的设计和交互。

3.5. 安全性设计（赵嘉萱）

3.5.1. 安全性要求

一、数据安全：确保所有数据在存储、传输和处理过程中都得到保护，防止未经授权的访问或泄露。

解决方案：使用加密技术对数据进行保护，如 SSL/TLS 协议进行网络传输加密，AES 等算法进行存储加密。同时，还需要实施严格的权限控制策略，只有授权用户才能访问特定的数据。

二、访问控制：确保只有经过身份验证和授权的用户可以访问系统。

解决方案：采用身份验证技术和访问控制机制，例如用户名/密码认证、多因素认证等方法，以及基于角色的访问控制（RBAC）模型，限制用户的操作权限。

三、安全审计：记录和监控系统中的所有活动，以便在发生安全事件时能够追踪和恢复。

解决方案：实现日志记录和审核功能，记录每个用户的操作和访问情况，定期审查日志并分析可疑行为。同时，可以使用入侵检测系统（IDS）和安全信息和事件管理（SIEM）工具来

实时监控系统和检测潜在威胁。

四、隐私保护：遵守相关的隐私法规和政策，确保个人信息不会被滥用或泄露。

解决方案：遵循 GDPR、CCPA 等数据保护法规，在收集、使用、共享个人数据之前获取用户的同意，并且明确告知用户其权利。同时，通过匿名化、去标识化等技术减少敏感信息的暴露风险。

五、容灾备份：在灾难性故障或攻击发生时保证业务连续性和数据完整性。

解决方案：建立冗余的基础设施和分布式架构，以提高系统的可用性和容错能力。同时，定期进行备份并将备份数据存储在安全的地方，以便在必要时恢复系统和数据。

六、安全培训与意识：使员工了解网络安全的重要性，避免由于人为疏忽造成的数据泄露或其他安全事故。

解决方案：开展定期的安全培训和意识宣传，让员工了解常见的威胁类型和防护措施。同时，制定明确的操作规程和安全政策，并严格执行。

七、应急响应：当安全事件发生时，具备快速应对和恢复的能力。

解决方案：制定详细的应急响应计划，包括事件分类、报告流程、处置策略等内容。组织专门的安全团队负责应对紧急情况，确保能够在最短时间内控制损失并恢复正常运行。

八、软件供应链安全：确保使用的软件和组件都是安全可靠的，避免引入外部漏洞或恶意代码。

解决方案：实行严格的代码审查和安全测试，对开发过程进行全面的审计和监督。选择信誉良好的供应商和服务商，并跟踪他们的安全更新和补丁发布。

4. 测试策略

4.1 单元测试

描述如何进行单元测试，包括测试用例和工具。

4.2 集成测试

解释系统集成测试的计划和方法。

4.3 系统测试

说明系统测试的计划 and 策略。

5. 风险管理

5.1 风险识别

列出项目中可能出现的风险，并对其进行评估。

5.2 风险应对措施

描述应对风险的具体措施和计划。

6. 项目计划

提供项目的时间表和阶段性里程碑。

7. 结论

总结本报告的主要内容和下一步的行动计划。

这只是一个示例模板，您可以根据项目的具体需求和组织的要求进行自定义。在编写软件报

告文档时，请确保文档清晰、详细，并包含足够的信息，以便开发人员、测试人员和利益相关者能够理解和实施项目的设计和开发计划