

# UM05 住宿管理功能

## 功能描述

在宿舍管理系统中，住宿管理功能涵盖了学生的宿舍安排与退宿管理，包括宿舍的选择、入住与退宿的操作、处理住宿申请等。以下将对各个主要功能模块进行详细描述。

### 1. 安排住宿

安排住宿功能主要包括通过选择宿舍楼与房间号，为学生安排住宿。这一过程首先通过系统提供的接口进行宿舍与房间选择，然后插入相关记录到数据库中。

#### 关键步骤：

- **选择有效的房间：**系统根据学生的学号选择可用的房间，并且通过 `selectValidRoom` 方法确保选择的房间尚未满员。
- **输入备注：**系统允许管理员在入住过程中输入可选的备注信息。
- **入住记录插入：**入住记录被插入到 `check_in_out_records` 表中，以便追踪学生的住宿历史。
- **更新学生入住状态：**通过设置学生的 `isCheckedIn` 属性为1，标记学生已经入住。
- **房间状态更新：**调用 `db.updateRoomStatus` 方法更新房间的状态，以反映当前的入住情况。

### 2. 安排退宿

退宿操作涉及到学生从宿舍中退房，并更新系统中的相关记录。管理员首先确认是否继续退宿操作，然后输入退宿备注信息，最后系统通过一键退宿功能完成整个退宿流程。

#### 关键步骤：

- **确认退宿操作：**管理员输入确认信息 `yes` 或 `no` 以决定是否继续退宿操作。
- **查询学生住宿信息：**系统查询学生当前的宿舍楼与房间号，并输出相应的住宿信息。
- **退宿记录插入：**退宿事件作为记录插入到 `check_in_out_records` 表中，确保每一次退宿都有详细的记录。
- **更新入住状态：**更新学生的 `isCheckedIn` 状态为0，标识该学生已退宿。
- **删除房间分配记录：**从 `student_rooms` 表中删除学生与房间的关系记录。
- **房间状态更新：**与安排住宿功能一样，系统会调用 `db.updateRoomStatus` 方法，确保房间状态得到及时更新。

### 3. 处理学生住宿申请

住宿申请处理功能用于管理学生的住宿请求，包括入住、换宿或退宿申请。管理员需要查看待审批的申请并决定是否批准每一项申请。

#### 关键步骤：

- **查看待审批申请：**系统从数据库中查询所有待审批的住宿申请，并展示给管理员。
- **选择处理请求：**管理员选择需要处理的请求ID，并查看该请求的详细信息。
- **审批操作：**管理员可以选择通过申请、拒绝申请或返回。对于通过的申请，系统会自动完成相应的住宿安排或退宿操作。
- **更新申请状态：**每个处理过的申请都会更新审批状态（已审批或已拒绝），并记录审批时间。

### 4. 查看入住退宿记录

系统允许管理员查看所有学生的入住与退宿记录，以便了解学生的住宿历史。管理员可以选择查询某一学生的具体记录，或者查看所有学生的历史操作记录。

#### 关键步骤：

- **查询所有记录或指定学生记录：**管理员可以选择查看所有学生的入住退宿记录，或者只查询特定学生的记录。
- **展示记录：**系统输出包括学生ID、房间ID、事件时间、记录类型等详细信息，帮助管理员做出合理的管理决策。

### 5. 查看未入住学生

通过 `checkUserNotCheckedIn` 功能，管理员能够查看所有未入住的学生，以便更好地进行宿舍安排。

#### 关键步骤：

- **查询未入住学生：**系统查询所有没有入住宿舍的学生，展示包括学生学号、姓名、联系方式等信息。
- **展示结果：**显示未入住的学生信息，供管理员参考与后续处理。

## 遇到的问题与解决方案

### 问题 1：宿舍房间满员时无法安排住宿

在实现 `quickArrangeAccommodation` 方法时，系统会检查所选房间是否已满。若房间已满，系统将无法为学生安排住宿。为了避免这种情况，采用了SQL查询来判断房间是否有剩余空间，并在

插入记录之前确认房间是否能容纳更多学生。

**解决方案：**引入 `r.occupied < r.capacity` 条件，确保只有当房间未滿时才允许安排入住。

## 问题 2：退宿操作过程中无法正确获取学生的宿舍信息

在 `quickArrangeCheckOut` 过程中，查询学生的宿舍信息时若数据库返回为空，说明未能正确匹配学生的宿舍与房间信息。

**解决方案：**通过修改SQL查询语句，确保学生的宿舍信息能够正确匹配，避免因数据不一致而导致操作失败。

## 问题 3：处理住宿申请时存在重复审批

由于学生可能会重复提交相同的住宿申请，系统需要确保每个请求只被审批一次。

**解决方案：**使用 `unordered_set` 来存储申请ID，避免重复处理已经审批过的申请。每个有效的申请会在系统中标记为“已审批”或“已拒绝”，并删除处理过的申请。

## 收获

- 数据库操作优化：**通过深入了解如何使用SQL语句管理学生的住宿信息，我提升了在数据库查询、更新和删除操作方面的技能。合理的SQL设计可以确保数据一致性和操作的高效性。
- 系统设计思维：**在设计住宿管理功能时，我学会了如何拆解问题，逐步实现每个功能模块，并确保系统的可扩展性和可维护性。例如，将每项操作分解为“快速入住”、“快速退宿”等小模块，有利于后续的维护与升级。
- 错误处理：**在处理学生住宿信息时，尤其是入住和退宿操作，我深刻意识到数据验证和错误处理的重要性。通过增强系统的异常处理机制，可以避免操作失败或数据丢失。