



Web应用开发课程设计

师生健康码管理系统

技术报告

姓　　名　　 胡皓睿 王程飞

班　　级　　 软工1805

提交日期　　 2020/06/17

目录

[1) 系统概述 3](#_Toc43161245)

[2) 系统框架 4](#_Toc43161246)

[2.1. 网络架构 4](#_Toc43161247)

[2.2. 系统架构 5](#_Toc43161248)

[2.3. 模块功能说明 5](#_Toc43161249)

[2.3.1. 学生打卡模块 5](#_Toc43161250)

[2.3.2. 打卡情况统计模块 6](#_Toc43161251)

[2.3.3. 健康码管理模块 6](#_Toc43161252)

[2.3.4. 数据导入导出模块 7](#_Toc43161253)

[3) 系统需求分析 8](#_Toc43161254)

[3.1. 学生打卡&健康码申领 8](#_Toc43161255)

[3.1.1. 功能描述 8](#_Toc43161256)

[3.1.2. 业务处理流程及要求 8](#_Toc43161257)

[3.1.3. 输入信息 8](#_Toc43161258)

[3.1.4. 输出信息 8](#_Toc43161259)

[3.2. 管理员导入导出与查看 8](#_Toc43161260)

[3.2.1. 功能描述 8](#_Toc43161261)

[3.2.2. 业务处理流程及要求 9](#_Toc43161262)

[3.2.3. 输入信息 10](#_Toc43161263)

[3.2.4. 输出信息 10](#_Toc43161264)

[3.3. 健康码的使用与核销 10](#_Toc43161265)

[3.3.1. 功能描述 10](#_Toc43161266)

[3.3.2. 业务处理流程及要求 11](#_Toc43161267)

[3.3.3. 输入信息 11](#_Toc43161268)

[3.3.4. 输出信息 11](#_Toc43161269)

[4) 系统详细设计 11](#_Toc43161270)

[4.1. login.jsp 11](#_Toc43161271)

[4.2. check\_health\_code.jsp 12](#_Toc43161272)

[4.3. admin\_punchview.jsp 12](#_Toc43161273)

[4.4. Controller.kt 12](#_Toc43161274)

[4.5. SQL.kt 13](#_Toc43161275)

[4.6. Admin.java 15](#_Toc43161276)

[5) 数据库设计 16](#_Toc43161277)

[5.1. 数据库表概念结构设计 16](#_Toc43161278)

[5.1.1. 学院 16](#_Toc43161279)

[5.1.2. 专业 16](#_Toc43161280)

[5.1.3. 班级 16](#_Toc43161281)

[5.1.4. 管理员 17](#_Toc43161282)

[5.1.5. 教师 17](#_Toc43161283)

[5.1.6. 学生 17](#_Toc43161284)

[5.1.7. 健康信息 17](#_Toc43161285)

[5.1.8. 打卡记录 18](#_Toc43161286)

[5.2. 数据库表概念关系模型 18](#_Toc43161287)

[6) 系统实现功能 19](#_Toc43161288)

[6.1. 学生打卡与健康码申领 19](#_Toc43161289)

[6.2. 管理员查看打卡信息 21](#_Toc43161290)

[6.3. 系统管理员上传数据 24](#_Toc43161291)

[6.4. 健康码的使用与核销 25](#_Toc43161292)

[6.5. 使用NFC完成一碰登陆 27](#_Toc43161293)

[7) 程序清单 28](#_Toc43161294)

# 系统概述

2019年底开始流行的新型冠状病毒（2019-nCoV）疫情给全世界人民带来了无法考量的灾难。在全国人民众志成城地抗疫下，我国的疫情于3月份逐渐得到了有效控制。浙江省人民政府新闻办公室于2020年4月3日下午举行第四十一场新冠肺炎疫情防控工作新闻发布会，浙江省教育厅公布了开学复课方案。为确保开学安全，浙江省各地教育主管部门和学校制定了一系列保障措施。一是在开学前，对即将返校的师生健康状况作了全面排摸，按“一人一档”要求建立师生健康信息档案。二是在开学后，师生继续凭“健康码+测体温”受控入校，健康状况有异常的不得进校，对因病请假缺课的学生要追踪病因。

健康码是以个人健康数据为基础，由居民自主通过健康登记系统网上申报，结合新冠肺炎疫情相关数据进行比对核验，生成个人专属二维码。杭州市于2020年2月11日率先在全国推出绿色、黄色和红色的健康码，绿码可以直接进入杭州，红码集中隔离14天，黄码隔离7天以内。

某大学为确保开学安全，要求所有师生申领健康码并每天打卡。请您为该大学设计并开发一套师生健康码管理系统，包括学院管理、专业管理、班级管理、教师管理、学生管理、健康码查询与统计、打卡情况统计等功能。具体功能如下：

（1）系统管理员可设置学院、专业、班级等信息并进行管理，可在后台导入教师数据和学生数据并进行管理，教师数据包括姓名、身份证号、工号、学院、角色（系统管理员、校级管理员、院级管理员、普通教师），学生数据包括姓名、身份证号、学号、学院、专业、班级等信息。也可查询统计师生的健康码和每天的打卡情况。

（2）校级管理员通过工号、密码（密码由系统管理员设置）登录系统，可以查看整个学校的学院信息、专业信息、班级信息、教师信息、学生信息、查询统计师生的健康码和每天的打卡情况。

（3）院级管理员通过工号、密码（密码由系统管理员或校级管理员设置）登录系统，可以查看本学院的专业信息、班级信息、学生信息、查询统计本学院师生的健康码和每天的打卡情况。

（4）学生和教师通过申领学校健康码页面填报信息，系统根据师生填报的信息自动生成每个人唯一的健康码（二维码信息包括姓名、身份（老师或同学）、工号或学号、学院），师生可以查看自己的专属二维码作为进出校园的凭证，绿码可以进出校园，黄码需居家观察7天不得进入校园，红码需居家医学观察或集中隔离14天不得进入校园。所有教师和学生根据姓名、学号和身份证后8位登录系统进行申领健康码和每日打卡。

申领学校健康码填报的信息如下：

 个人信息，包括姓名、身份证号、工号或学号、手机号，其中（其中姓名、身份证号、工号或学号为系统默认值，不可修改）。

 本人近期（14天内）是否去过湖北省或重点疫区？

 本人近期（14天内）是否去过国外？

 本人近期（14天内）是否接触过新冠确诊病人或疑似病人？

 本人是否被卫生部门确认为新冠肺炎确诊病例或疑似病例？

 当前健康状况？无异常、发烧（≥37.3℃）、乏力、干咳、鼻塞、流涕、咽痛、腹泻等。

 本人郑重承诺：填报信息真实，愿意承担相应的法律责任。

如果有下列任意一种情况的为黄色健康码

 本人近期（14天内）去过湖北省或重点疫区

 本人近期（14天内）去过国外

 当前健康状况有且仅有发烧（≥37.3℃）、乏力、干咳、鼻塞、流涕、咽痛、腹泻中的一种

如果有下列任意一种情况的为红色健康码

 本人近期（14天内）接触过新冠确诊病人或疑似病人

 本人被卫生部门确认为新冠肺炎确诊病例或疑似病例

 当前健康状况有发烧（≥37.3℃）、乏力、干咳、鼻塞、流涕、咽痛、腹泻等中两种症状及以上

除黄色健康码和红色健康码以外的情形为绿色健康码。

红色健康码的师生需连续打卡7天满足绿色健康码条件才可转为黄码；黄色健康码的师生需连续打卡7天满足绿色健康码条件才可转为绿码。

# 系统框架

## 网络架构



## 系统架构



## 模块功能说明

### 学生打卡模块

#### 功能描述

负责学生打卡的有关操作。从数据库中读取信息，校验学生是否可以打卡，并且将学生打卡的记录重新存放至数据库中。

#### 业务处理流程及要求

1. 从数据库中读取这个学生的打卡记录，判断这个学生是否需要打卡
2. 跳转至打卡界面，让学生输入健康信息
3. 判断信息是否合法，并且进行上传
4. 后端进行校验，完成打卡

#### 输入信息

学生的健康信息

#### 输出信息

是否需要打卡，打卡是否成功

### 打卡情况统计模块

#### 功能描述

复制管理员查看学生的打卡信息，支持按照学院，专业和班级进行筛选

#### 业务处理流程及要求

1. 判断管理员是否具有管理权限
2. 管理员选择学院，返回学院的专业信息
3. 管理员选择专业，返回专业内的班级信息
4. 管理员选择班级，返回班级学生的健康信息

#### 输入信息

需要查询的学院，专业，班级

#### 输出信息

查询对象的健康状况

### 健康码管理模块

#### 功能描述

在管理员扫描健康码后，显示该生的健康信息，并且可以进行入校核销操作。

#### 业务处理流程及要求

1. 管理员扫描健康码，显示健康码信息
2. 若管理员没有健康码访问权限，仅显示部分身份信息
3. 具有权限的管理员完成健康码的校验与核销工作

#### 输入信息

健康码对应的学号与令牌

#### 输出信息

该健康码的具体信息，学生的身份信息

### 数据导入导出模块

#### 功能描述

提供导入用户信息数据库，导出师生健康状况的功能

#### 业务处理流程及要求

1. 管理员选择要上传的数据库文件
2. 管理员校验密码
3. 管理员校验邮件验证码
4. 完成上传工作
5. 管理员选择要导出的内容
6. 后端服务器生成电子表格文件

#### 输入信息

要上传的数据库文件，密码，邮件验证码

要导出的内容

#### 输出信息

成功与失败的提示

导出的内容

# 系统需求分析

## 学生打卡&健康码申领

### 功能描述

学生和教师登录系统后可以进行健康打卡（申报），并申领或查看自己的健康码，如果不是绿码，需要连续7天（黄码）或14天（红码）的正常打卡才能转绿码，可以对历史打卡记录进行查看

### 业务处理流程及要求

学生和教师登录成功进入健康码系统时，可以进行健康申报，填写自己的健康状况信息并勾选真实性承诺后即可申报成功，申报完成后可以进行健康码领取，系统会根据用户填报的信息生成对应的健康码，如果是红码或黄码则需要进行连续14天或7天的健康打卡才可以恢复，恢复绿码时提示用户欢迎回校的信息，用户可以点击打卡历史查看最近的打卡状况，同一用户一天只能打一次卡

### 输入信息

用户身份信息，健康信息，健康码申领请求

### 输出信息

打卡状态，健康码状态，打卡历史记录

## 管理员导入导出与查看

### 功能描述

该功能提供系统管理员对健康码管理系统的数据进行导入导出，包括以一个Excel表格完成对所有管理员、学院、专业、班级、教师学生进行设置并检查数据是否合法，如果不合法则提示用户错误信息，否则对数据库进行数据导入，导入前会进行密码和邮箱二次校验来确保用户操作安全。导出功能支持导出和查看学生和教师的健康码信息以及每天的打卡记录，支持分学院导出和查看

### 业务处理流程及要求

导入功能：

系统管理员登录系统，选择导入信息，进入导入信息页面，下载导入Excel模板，按照导入模板输入管理员、学院、专业、班级、教师、学生信息并检查是否有重复或引用不合法的情况，保存后点击选择文件，选择编辑好的Excel文件并点击导入，系统会提示再次输入密码，密码会交由服务器验证后将结果返回给用户，如果密码正确，进入邮箱验证，系统会给事先绑定的邮箱发送一个验证码，当系统管理员输入正确的验证码后将Excel文件上传到服务器进行导入，服务器端通过将文件输入流传入Import类进行验证和导入，先验证数据是否符合要求，并且教师和学生的学院班级都存在，工号或学号没有重复等错误则进行导入操作，将数据进行组织并存入内存对象中，将所有的数据表清空后插入数据，中途如果出现错误则会进行提示，成功后会进入登陆界面进行后续操作

查看功能：

支持各级管理员对其管理范围内的教师和学生进行查看信息，点击信息概览，可以查看所有教师和学生的健康码颜色、学院、工号、身份证号及备注信息，并且可以进行筛选，支持按学院、健康码颜色进行筛选，点击打卡情况，可以按日期查看每天的学生和教师的打卡情况，可以对学院、日期、打卡状态进行筛选，信息概览和打卡情况可以导出为Excel方便统计，点击学院概览可以对各学院内的专业、班级、教师、学生列表进行查看

手机屏幕截图

描述已自动生成 电脑屏幕截图

描述已自动生成

### 输入信息

导入功能：

输入密码、邮箱验证码、Excel数据

查看和导出功能：

筛选条件

### 输出信息

导入功能：

导入成功信息或错误信息

查看和导出功能：

健康码列表、打卡信息列表、学生和教师列表

## 健康码的使用与核销

### 功能描述

扫描屏幕上的二维码或者使用带有NFC的设备刷健康卡可以进入健康码的核销 ，当没有健康码权限的人扫码时，为了保护个人信息，只会显示隐藏部分敏感信息之后的健康码和个人数据，健康码本身不包含个人信息，而是生成一个唯一的token来映射，可以防止伪造的同时确保信息安全，健康码管理员可以查看健康码以及核销健康绿码

### 业务处理流程及要求

带健康码权限的管理员可以登录管理员系统后扫描屏幕上的二维码或者使用带有NFC的设备刷健康卡，跳转到对应学生或教师的健康信息界面，该界面通过检测登录的人员权限分情况展示，如果是无权限的用户则显示部分信息确保安全，并点击管理员登录按钮可以提供管理员登录来进行查看更多信息以及健康码核销操作，管理员可以查看学生和教师的详细信息，点击确认入校并确认后即可完成学生入校流程，确认入校后该健康码作废不可再次使用，显示为健康码已被使用

### 输入信息

学生或教师带有健康码信息的二维码或NFC健康卡

管理员的健康码核销请求

### 输出信息

学生或教师的个人健康信息以及健康码使用情况

管理员核销健康码结果

# 系统详细设计

## login.jsp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 描述 | 课程管理页面 | | | |
| 所在工程目录 | \src\main\webapp\login.jsp | | | |
| 加载的文件 | 文件名 | | 描述 | |
| mdui.min.css | | Material Design的UX设计 | |
| style.css | | 自定义的页面CSS | |
| md5.min.js | | MD5方法的JavaScript实现 | |
| mdui.min.js | | Material Design的JavaScript代码 | |
| 主要的javascript函数 | 名称 | 属性 | | 描述 |
| md5 |  | | 获取输入串的MD5特征值 |
| login |  | | 校验表单合法性，在合法时提交表单 |

## check\_health\_code.jsp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 描述 | 校验健康码页面 | | | |
| 所在工程目录 | \src\main\webapp\check\_health\_code.jsp | | | |
| 加载的文件 | 文件名 | | 描述 | |
| mdui.min.css | | Material Design的UX设计 | |
| style.css | | 自定义的页面CSS | |
| mdui.min.js | | Material Design的JavaScript代码 | |
| 主要的javascript函数 | 名称 | 属性 | | 描述 |
| check |  | | 发出警告信息，确定是否要核销健康码 |

## admin\_punchview.jsp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 描述 | 课程管理页面 | | | |
| 所在工程目录 | \src\main\webapp\admin\_punchview.jsp | | | |
| 加载的文件 | 文件名 | | 描述 | |
| mdui.min.css | | Material Design的UX设计 | |
| style.css | | 自定义的页面CSS | |
| mdui.min.js | | Material Design的JavaScript代码 | |
| table.js | | 一个表格类 | |
| jquery-3.5.1.js | | jQuery | |
| 主要的javascript函数 | 名称 | 属性 | | 描述 |
| update |  | | 使用Ajax技术动态获得服务器上的打卡记录信息，然后使用table.js处理，显示 |

## Controller.kt

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | Controller | | |
| 继承或实现 | HttpServlet, Router | | |
| 描述 | MVC设计模式的Controller | | |
| 类别 |  | | |
| 包 | cn.edu.zjut.health.code.servlet | | |
| 与其它类的调用关系 | 是 | | |
| 属性 | 名称 | 类型 | 描述 |
| context | String | 上下文 |
| get | HashMap<> | get方法的路由表 |
| post | HashMap<> | post方法的路由表 |
| sex | int | 性别 |
| 主要方法 | 名称 | 属性 | 描述 |
| init | void | 初始化路由表，提供转发 |

|  |  |
| --- | --- |
| 方法名称 | init |
| 描述 | 初始化路由表，提供转发 |
| 输入参数 |  |
| 步骤 | 将路由表有关信息放至到该方法中，即可将路由信息存储至路由表中 |
| 输出参数 | void |
| 注释 |  |

## SQL.kt

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | SQL | | |
| 继承或实现 |  | | |
| 描述 | 一种工具化的读取数据库数据的方式 | | |
| 类别 | 工具类 | | |
| 包 | model | | |
| 与其它类的调用关系 | 是 | | |
| 主  要  方  法 | 名称 | 属性 | 描述 |
| Class<T>.model() | Pair<String, List<Pair< Field, String?>>>? | 扩展方法，返回类的数据类对应的表名和字段 |
| Field.hasFK() | Boolean | 扩展方法，判断是否有ForeignKey注解 |
| instance(Class<T>, List<Pair<java.lang.reflect.Field, String?>>, ResultSet) | T | 通过类型和ResultSet构造对象 |
| queryList(Class<T>, Pair<String, K>) | ArrayList<T> | 从数据库查询符合条件的所有对象 |
| query(Class<T>, Pair<String, K>?) | T? | 从数据库查询符合条件的单个对象 |

|  |  |
| --- | --- |
| 方法名称 | Class<T>.model() |
| 描述 | 扩展方法，返回类的数据类对应的表名和字段 |
| 输入参数 | Class<T> |
| 步骤 | 获取输入类的Model注解判断其是否为一个合法的数据类，对其成员带有Field的注解进行收集整理并返回 |
| 输出参数 | Pair<String, List<Pair< Field, String?>>>? |
| 注释 |  |

|  |  |
| --- | --- |
| 方法名称 | Field.hasFK() |
| 描述 | 扩展方法，判断是否有ForeignKey注解 |
| 输入参数 | Field |
| 步骤 | 直接获取注解列表并判断 |
| 输出参数 | Boolean |
| 注释 |  |

|  |  |
| --- | --- |
| 方法名称 | instance(Class<T>, List<Pair<java.lang.reflect.Field, String?>>, ResultSet) |
| 描述 | 通过类型和ResultSet构造对象 |
| 输入参数 | Class<T>, List<Pair<java.lang.reflect.Field, String?>>, ResultSet |
| 步骤 | 判断传入的Class类型，属性列表以及数据库查询结果，通过反射的方式来构造对象，读取属性在数据库中的字段名后进行赋值 |
| 输出参数 | T |
| 注释 |  |

## Admin.java

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | Admin | | |
| 继承或实现 |  | | |
| 描述 | 管理员类 | | |
| 类别 | 模型类 | | |
| 包 | model | | |
| 与其它类的调用关系 | 是 | | |
| 属性 | 名称 | 类型 | 描述 |
| Id | String | 工号 |
| level | int | 管理员类型 |
| target | Integer | 管理对象 |
| password | String | 密码 |
| email | String | 邮箱 |
| 主  要  方  法 | 名称 | 属性 | 描述 |
| saltMD5 | String | 返回加盐的MD5 |
| getFullTarget() | FullTarget | 返回详细的管理员权限和管理对象 |
| equals | Object | 判断管理员类是否值相等 |

# 数据库设计

## 数据库表概念结构设计

### 学院

|  |  |  |  |
| --- | --- | --- | --- |
|  | 标识符 | 意义 | 类型 |
| 🗝 | id | 编号 | int(11) |
|  | name | 学院名称 | varchar(32) |

### 专业

|  |  |  |  |
| --- | --- | --- | --- |
|  | 标识符 | 意义 | 类型 |
| 🗝 | id | 编号 | int(11) |
| 🗝 | collage | 所属学院号 | int(11) |
|  | name | 专业名称 | varchar(32) |

### 班级

|  |  |  |  |
| --- | --- | --- | --- |
|  | 标识符 | 意义 | 类型 |
| 🗝 | id | 编号 | int(11) |
| 🗝 | major | 所属专业 | int(11) |
|  | name | 专业名称 | varchar(32) |

### 管理员

|  |  |  |  |
| --- | --- | --- | --- |
|  | 标识符 | 意义 | 类型 |
| 🗝 | id | 编号 | varchar(16) |
|  | level | 管理员等级 | int(11) |
| 🗝 | target | 管理对象 | int(11) |
|  | password | 管理员密码 | varchar(32) |
|  | email | 电子邮件 | varchar(64) |

### 教师

|  |  |  |  |
| --- | --- | --- | --- |
|  | 标识符 | 意义 | 类型 |
| 🗝 | id | 编号 | varchar(16) |
|  | name | 姓名 | varchar(64) |
|  | person\_id | 身份证号 | varchar(32) |
| 🗝 | collage | 所属学院 | int(11) |

### 学生

|  |  |  |  |
| --- | --- | --- | --- |
|  | 标识符 | 意义 | 类型 |
| 🗝 | id | 编号 | varchar(16) |
| 🗝 | class | 所属班级 | int(11) |
|  | name | 姓名 | varchar(64) |
|  | person\_id | 身份证号 | varchar(32) |

### 健康信息

|  |  |  |  |
| --- | --- | --- | --- |
|  | 标识符 | 意义 | 类型 |
| 🗝 | uid | 学号/工号 | varchar(12) |
|  | tel | 电话号码 | varchar(16) |
|  | if\_danger\_14 | 十四日内是否去过武汉 | int(11) |
|  | if\_abroad\_14 | 十四日内是否出国 | int(11) |
|  | if\_touch\_illness | 十四日内是否接触过患者 | int(11) |
|  | if\_ill | 十四日内是否的得病 | int(11) |
|  | status | 健康状况 | int(11) |
|  | color | 健康码颜色 | varchar(8) |

### 打卡记录

|  |  |  |  |
| --- | --- | --- | --- |
|  | 标识符 | 意义 | 类型 |
| 🗝 | uid | 学号/工号 | varchar(16) |
| 🗝 | date | 打卡日期 | date |
|  | time | 打卡时间 | time |
|  | status | 健康状况 | int(11) |
|  | color | 健康码颜色 | varchar(8) |

## 数据库表概念关系模型



# 系统实现功能

## 学生打卡与健康码申领

 手机屏幕截图

描述已自动生成 截图里有图片

描述已自动生成手机屏幕截图

描述已自动生成

## 管理员查看打卡信息

 截图里有图片

描述已自动生成 电脑屏幕截图

描述已自动生成 手机屏幕截图

描述已自动生成 电脑屏幕截图

描述已自动生成

## 系统管理员上传数据

 社交网络的手机截图

描述已自动生成

## 健康码的使用与核销

 社交网络的手机截图

描述已自动生成 手机屏幕截图

描述已自动生成

## 使用NFC完成一碰登陆



通过将登陆有关的数据存入NFC卡中，手机可以实现刷卡登陆功能。样图如下：  

# 程序清单

├── main

│   ├── java

│   │   ├── conn

│   │   │   └── DatabaseProvider.java

│   │   ├── model

│   │   │   ├── Admin.java

│   │   │   ├── AtInterfaces.kt

│   │   │   ├── Clazz.java

│   │   │   ├── CodeColor.kt

│   │   │   ├── Collage.java

│   │   │   ├── FullTarget.kt

│   │   │   ├── HealthCodeTester.java

│   │   │   ├── HealthInfo.kt

│   │   │   ├── Major.java

│   │   │   ├── Person.java

│   │   │   ├── PunchRecord.kt

│   │   │   ├── Record.java

│   │   │   ├── SQL.kt

│   │   │   ├── Status.java

│   │   │   ├── Student.java

│   │   │   └── Teacher.java

│   │   ├── servlet

│   │   │   ├── AdminRouterServlet.kt

│   │   │   ├── CheckCaptchaServlet.java

│   │   │   ├── Controller.kt

│   │   │   ├── DestroyHealthCodeServlet.java

│   │   │   ├── ForgetPasswordServlet.java

│   │   │   ├── GetCaptchaServlet.java

│   │   │   ├── GetHealthCodeServlet.java

│   │   │   ├── GetInfoByIdServlet.java

│   │   │   ├── GetQrCode.java

│   │   │   ├── LoginServlet.java

│   │   │   ├── LogoutServlet.java

│   │   │   ├── PunchRecordData.kt

│   │   │   ├── ResetPasswordServlet.java

│   │   │   ├── UploadDatabaseServlet.kt

│   │   │   └── UploadPunchServlet.java

│   │   └── util

│   │   ├── Import.kt

│   │   ├── Mail.java

│   │   ├── MultipleForm.kt

│   │   └── Router.kt

│   └── webapp

│   ├── 400.jsp

│   ├── 404.jsp

│   ├── 500.jsp

│   ├── admin\_overview.jsp

│   ├── admin\_punchview.jsp

│   ├── check\_health\_code.jsp

│   ├── college\_admin.jsp

│   ├── college\_information.jsp

│   ├── css

│   │   ├── mdui.css

│   │   ├── mdui.min.css

│   │   ├── mdui.min.css.map

│   │   └── style.css

│   ├── error\_code.jsp

│   ├── error.jsp

│   ├── files

│   │   └── demo.xlsx

│   ├── fonts

│   │   └── roboto

│   │   ├── LICENSE.txt

│   │   ├── Roboto-BlackItalic.woff

│   │   ├── Roboto-BlackItalic.woff2

│   │   ├── Roboto-Black.woff

│   │   ├── Roboto-Black.woff2

│   │   ├── Roboto-BoldItalic.woff

│   │   ├── Roboto-BoldItalic.woff2

│   │   ├── Roboto-Bold.woff

│   │   ├── Roboto-Bold.woff2

│   │   ├── Roboto-LightItalic.woff

│   │   ├── Roboto-LightItalic.woff2

│   │   ├── Roboto-Light.woff

│   │   ├── Roboto-Light.woff2

│   │   ├── Roboto-MediumItalic.woff

│   │   ├── Roboto-MediumItalic.woff2

│   │   ├── Roboto-Medium.woff

│   │   ├── Roboto-Medium.woff2

│   │   ├── Roboto-RegularItalic.woff

│   │   ├── Roboto-RegularItalic.woff2

│   │   ├── Roboto-Regular.woff

│   │   ├── Roboto-Regular.woff2

│   │   ├── Roboto-ThinItalic.woff

│   │   ├── Roboto-ThinItalic.woff2

│   │   ├── Roboto-Thin.woff

│   │   └── Roboto-Thin.woff2

│   ├── forget\_password.jsp

│   ├── get\_health\_code.jsp

│   ├── icons

│   │   └── material-icons

│   │   ├── LICENSE.txt

│   │   ├── MaterialIcons-Regular.ijmap

│   │   ├── MaterialIcons-Regular.woff

│   │   └── MaterialIcons-Regular.woff2

│   ├── image

│   │   └── card.jpg

│   ├── index.html

│   ├── js

│   │   ├── jquery-3.5.1.js

│   │   ├── md5.min.js

│   │   ├── mdui.js

│   │   ├── mdui.min.js

│   │   ├── mdui.min.js.map

│   │   ├── script.js

│   │   └── table.js

│   ├── login.jsp

│   ├── main.jsp

│   ├── punch\_history.jsp

│   ├── punch.jsp

│   ├── recover.jsp

│   ├── reset\_password.jsp

│   ├── scanner.jsp

│   ├── school\_admin.jsp

│   ├── student\_information.jsp

│   ├── system\_admin.jsp

│   ├── upload\_database.jsp

│   ├── view\_health\_code.jsp

│   └── WEB-INF

│   └── web.xml