

Covid-19 疫情分析与管理系统

Covid-19 Analysis & Manage System

简介

本项目作为数据库系统课程 2020 期末项目具有全球疫情可视化概览、病人诊断数据统计及编辑、诊断医院和医生管理等功能。

目前，项目已部署至 <https://covid19.billc.io>，并开发过程中使用 GitHub 来管理版本。

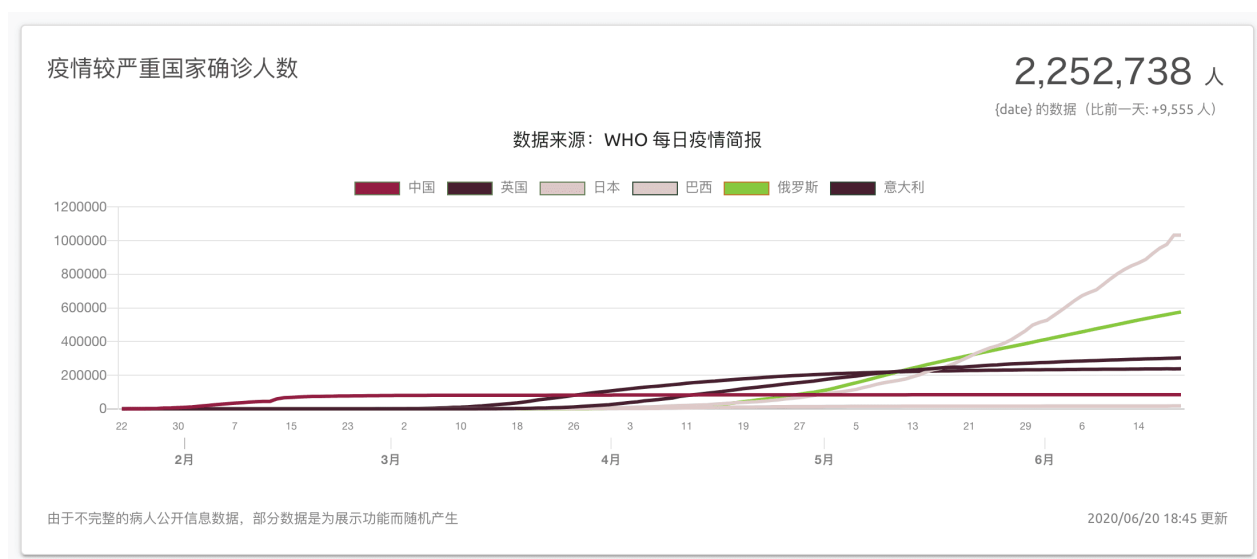
在 <https://github.com/billchen2k/Covid19> 可以获取到项目的源代码，通过 commit 记录可回顾本项目的开发进度。

• 首页

首页具有病人接诊卡片，指引用户界面友好的病人接诊指南，引导病人指引核酸、到相应的医疗机构就诊。

• 疫情概览

利用来自 WHO 每日疫情简报和丁香园的新冠肺炎疫情公开数据，绘制可视化的图表。展示了全球疫情较严重地区的确诊情况，包括中国、英国、日本、巴西、俄罗斯、意大利的可视化数据。我们还展示了国内目前疫情情况分布（分为了累计死亡、确诊人数、住院人数、无症状感染者、境外输入和治愈出院人数等），国际旅游入境人数、全球疫情变化情况、国内疫情变化情况、湖北省确诊人数、湖北省最新确诊患者信息等可视化数据，全方位地展示疫情的发展情况。



— 国内疫情人数情况

通过静态的数据显示国内真实的疫情人数情况，针对当前全新的防疫状况，增设无症状感染者一栏。同时通过形状的弯曲来表示包含关系，形式简约大方，又不失美感。

国内目前疫情情况分布

数据来源：来自国家卫健委和各省卫健委通报，含港澳台



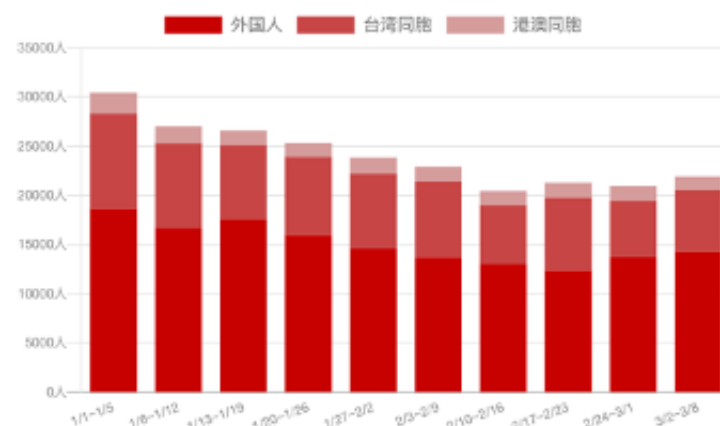
由于不完整的病人公开信息数据，部分数据是为展示功能而随机产生 2020/08/15 24:00 更新

国际旅游入境人数统计

通过可视化的比较通过疫情爆发前后，入境人数的变化。同时增设隐藏对应“外国人”、“台湾同胞”、“港澳同胞”数据的按钮，且会根据隐藏动态的调整纵坐标，已经根据比例动态缩放对应柱形图

国际旅游入境人数

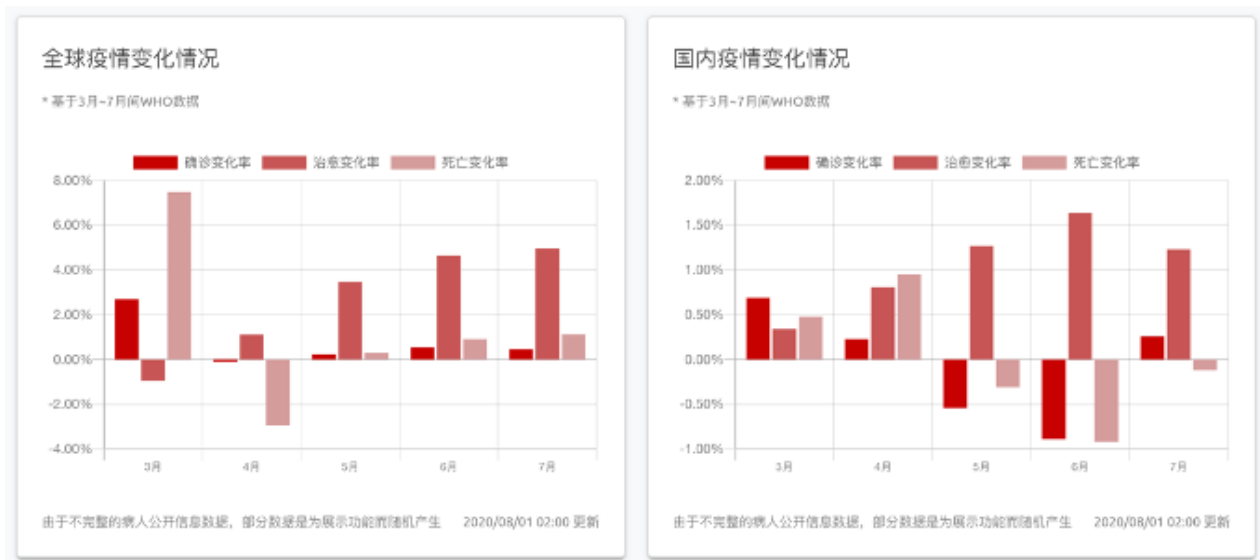
从上海口岸和我国其他口岸入境的来沪境外旅游人数



由于不完整的病人公开信息数据，部分数据是为展示功能而随机产生 2020/3/6 21:00 更新

国际/国内疫情变化

通过可视化的比较疫情爆发几个月国际和国内的增长幅度变化。同时增设隐藏对应选项数据的按钮，且会根据隐藏动态的调整纵坐标，已经根据比例动态缩放对应柱形，增长向上、降低向下，更加清晰。



最新确诊病人情况

数据与数据库匹配, 动态展示。在前端, 对数据进行了一定的处理展示(例如隐去姓名, 模糊出生日期), 在前端通过数据处理, 显示当前新增病例以及较前一天的变化情况, 同时通过数据处理展示累计情况。最新确诊患者信息可以对相应的列按照一定顺序排序。



病患查询

为了展示该模块的功能, 受限于有限的公开展示数据, 我们利用脚本结合真实数据虚构了数十万条合理的数据, 并实时更新。

该模块可以查询到数据库内的十四万条病患信息(和中国真实的感染人数相对应), 且可以展开详细信息进行查看。用户也可以根据病情、性别、发病日期、姓名、发病地点等条件对病患进行筛选。

条件筛选

当前共有 140345 条有效数据

病情状态

不限

性别

男

发病日期

2020-08-03

病人姓名

发病地点

查询

病患数据

快速过滤姓名或地点

#	姓名	性别	生日	发病地点	发病日期	确诊日期	确诊医院	负责医生	患者状态	详细信息
1	林丹	女	1939-03-31	新疆维吾尔...	2020-02-13	2020-02-13	台州第一人...	林伟	治疗中	展开
2	张淑兰	女	1939-03-10	新疆维吾尔...	2020-02-13	2020-02-13	乌鲁木齐第...	全晨	已康复	展开
3	欧阳秀珍	男	1982-06-26	新疆维吾尔...	2020-02-13	2020-02-13	乌鲁木齐第...	全晨	治疗中	展开
4	郭雪梅	男	1926-12-28	新疆维吾尔...	2020-02-13	2020-02-13	乌鲁木齐第...	薛杰	已康复	展开
5	温萍	女	1952-09-29	新疆维吾尔...	2020-02-13	2020-02-13	乌鲁木齐第...	邱宁	已康复	展开
6	王建军	女	2016-08-27	新疆维吾尔...	2020-02-13	2020-02-13	乌鲁木齐第...	张龙	已康复	展开
7	杨敏	男	2019-09-20	新疆维吾尔...	2020-02-13	2020-02-13	乌鲁木齐第...	萧丽娟	已康复	展开
8	刘秀荣	女	1933-08-01	新疆维吾尔...	2020-02-13	2020-02-13	乌鲁木齐第...	崔红霞	已康复	展开
9	阮静	男	1937-08-21	新疆维吾尔...	2020-02-13	2020-02-13	乌鲁木齐第...	吴桂珍	已康复	展开
10	丁波	女	1928-01-20	新疆维吾尔...	2020-02-13	2020-02-13	乌鲁木齐第...	孙涛	治疗中	展开
11	宋秀华	男	2001-02-26	新疆维吾尔...	2020-02-13	2020-02-13	乌鲁木齐第...	赵斌	治疗中	展开

展开详细信息可以查看到数据库中和该病患相关的所有数据。除了基础个人档案之外，还有处方与诊断记录的查看。处方信息包括了药品名称、用法用量及制药厂；诊断记录包括了诊断时间、治疗医师、体温、核酸检测情况和症状描述。同时，我们也对诊断记录中的体温一项制作了可视化的体温曲线，能够更加直观地得到用户的就诊情况。

患者档案

欧阳秀珍

乌鲁木齐第一人民医院，治疗中

欧阳秀珍

姓名

男

性别

1982-06-26

生日

新疆维吾尔自治区乌鲁木齐

发病地点

2020-02-13

发病日期

2020-02-13

确诊日期

全晨 (传染科)

主治医生

乌鲁木齐第一人民医院

所在医院

体温曲线

诊断中的体温记录将会被绘制在这里。



诊断记录

#	诊断医生	诊断时间	体温	核酸检测	症状
59	全晨	2020-02-13 08:50:00	37.7	阳性	已确诊为新冠肺炎，病状较轻，应注意控制以避免病情恶化。
60	全晨	2020-02-14 19:07:00	37.7	阳性	已确诊为新冠肺炎，病状较轻，应注意控制以避免病情恶化。
61	全晨	2020-02-15 14:53:00	36.9	阴性	病情稳定，情况良好。
62	全晨	2020-02-16 11:22:00	37.7	阳性	已确诊为新冠肺炎，病状较轻，应注意控制以避免病情恶化。
63	全晨	2020-02-17 12:07:00	37.7	阳性	轻微发烧症状，呼吸略有困难，需要进一步确定情况。
64	全晨	2020-02-18 14:26:00	39.0	阳性	温度较高，但核酸检测阴性，目前应当按照普通流感和肺炎治疗处理，并考虑再次核酸检测避免误诊。

治疗中
当前状态

处方信息

该病人当前的正在使用的处方如下:

青

青霉素散注射液-成人
注射, 一日三次, 一次 150mL,
广西三金制药厂

左

左氧氟沙星散剂胶囊-儿童
外用, 四小时一次, 一次 20g。
齐鲁制药

更

更昔洛韦散剂胶囊-成人
外用, 一日三次, 一次 20g。
恒瑞制药

65	全晨	2020-02-19 13:22:00	40.2	阳性	状态极其危险, 应作为重点监护对象, 保持密切关注, 辅助生命维持设备。
66	全晨	2020-02-20 13:21:00	37.7	阳性	已确诊为新冠肺炎, 病状较轻, 应注意控制以避免病情恶化。
67	全晨	2020-02-21 19:00:00	36.9	阴性	病情稳定, 情况良好。
68	全晨	2020-02-22 12:08:00	39.0	阴性	温度较高, 但核酸检测阴性, 目前应当按照普通流感和肺炎治疗处理, 并考虑再次核酸检测避免误诊。
69	全晨	2020-02-23 18:00:00	37.7	阴性	轻微发烧症状, 呼吸略有困难, 需要进一步确定情况。
70	全晨	2020-02-24 08:50:00	36.9	阴性	病情稳定, 情况良好。
71	全晨	2020-02-25 23:56:00	37.4	阳性	呼吸略有困难, 应当辅助药物治疗, 并保证已经被隔离。
72	全晨	2020-02-26 08:27:00	37.7	阴性	轻微发烧症状, 呼吸略有困难, 需要进一步确定情况。
73	全晨	2020-02-27 12:17:00	39.5	阳性	状态很危险, 应辅助呼吸治疗, 必要时转入重点监护病房。
74	全晨	2020-02-28 07:33:00	36.9	阴性	病情稳定, 情况良好。
75	全晨	2020-02-29 22:36:00	37.7	阳性	已确诊为新冠肺炎, 病状较轻, 应注意控制以避免病情恶化。
76	全晨	2020-03-01 20:57:00	37.7	阴性	轻微发烧症状, 呼吸略有困难, 需要进一步确定情况。
77	全晨	2020-03-02 20:07:00	37.4	阳性	呼吸略有困难, 应当辅助药物治疗, 并保证已经被隔离。
78	全晨	2020-03-03 10:11:00	36.9	阴性	病情稳定, 情况良好。

Rows per page: 5 1-20 of 29

这里使用的数据同样基于真实数据由脚本生成, 但病患的数量是根据丁香园每日新增、死亡和治愈的真实数据产生。

病患登记

用户可以在该模块快速登进新的病人。在完整填写所有信息之后也可以快速展开该病人的详细信息, 进行诊断和处方登记等操作。

病患登记

新患者登记

当前数据库内已有 140345 位患者数据。

姓名

性别

生日

发病地点

发病日期

确诊日期

主治医生

所在医院

当前状态

登记患者

登记诊断记录

当前数据库内已有 4075837 条诊断记录。

跳转到「 病患管理」来选择病人并登记 →

登记新处方

当前数据库内已有 421180 条处方记录。

跳转到「 病患管理」来选择病人并登记 →

已创建病人陈俊潼。您可以立即展开病人以开始登记诊断和处方。

展开病人详情...

确定

● 病患管理

该模块具有和病患查询一样的检索列表，但开放了编辑权限。医生可以为患者登记 / 删除诊断记录、新增 / 删除处方信息，及修改患者的所有个人信息，包括医院、医生、生日、性别、治愈情况等。仅有已登录的用户可以访问本页面并对病患进行修改。



新增诊断：



新增处方：

为了方便医生登记患者情况，处方和诊断的新建页面均提供了快速输入模板，提供了一些常用的登记模板。

- 关于

含有项目信息、数据库 ER 图、作者信息等声明。

• 技术架构

- Frontend

主要文件目录：

```
1 | frontend
2 | ├── README.md
3 | ├── assets
4 | |   ├── README.md
5 | |   ├── global.scss
6 | |   └── variables.scss
7 | ├── components
8 | |   ├── AgencyBarChart.vue
9 | |   ├── ConfirmedCasesTable.vue
10 | |   ├── DataSelector.vue
11 | |   ├── DataTable.vue
12 | |   ├── DataView.vue
13 | |   ├── DataViewBasicInfoPanel.vue
14 | |   ├── DesktopFlowSvg.vue
15 | |   ├── LanguageSelector.vue
16 | |   └── ListItem.vue
```

```
17 |   ├── MetroBarChart.vue
18 |   ├── MonitoringConsultationDeskReportChart.vue
19 |   ├── NumberDisplay.vue
20 |   ├── OpenDataLink.vue
21 |   ├── PageHeader.vue
22 |   ├── PrinterButton.vue
23 |   ├── README.md
24 |   ├── SideNavigation.vue
25 |   ├── StaticInfo.vue
26 |   ├── SvgCard.vue
27 |   ├── TestedCasesDetailsTable.vue
28 |   ├── TextCard.vue
29 |   ├── TimeBarChart.vue
30 |   ├── TimeStackedBarChart.vue
31 |   ├── WhatsNew.vue
32 |   ├── cards
33 |   │   ├── PositiveRateCard.vue
34 |   ├── flow
35 |   │   ├── FlowPc.vue
36 |   │   ├── FlowPcAdvisory.vue
37 |   │   ├── FlowPcDays.vue
38 |   │   ├── FlowPcHospitalized.vue
39 |   │   ├── FlowPcNotRequired.vue
40 |   │   ├── FlowPcPast.vue
41 |   │   ├── FlowPcPcr.vue
42 |   │   ├── FlowPcRequired.vue
43 |   │   ├── FlowPcSuspect.vue
44 |   │   ├── FlowSp.vue
45 |   │   ├── FlowSpAccordinging.vue
46 |   │   ├── FlowSpAdvisory.vue
47 |   │   ├── FlowSpElder.vue
48 |   │   ├── FlowSpGeneral.vue
49 |   │   ├── FlowSpHospitalized.vue
50 |   │   ├── FlowSpPast.vue
51 |   │   ├── FlowSpSuspect.vue
52 |   ├── global
53 |   │   ├── Config.vue
54 |   │   ├── EventBus.js
55 |   ├── patient
56 |   │   ├── DiagnosisCard.vue
57 |   │   ├── PatientDetail.vue
58 |   │   ├── PrescriptionCard.vue
59 |   │   ├── TemperatureChart.vue
60 |   ├── picker
61 |   │   ├── DatePicker.vue
62 |   │   ├── DateTimePicker.vue
63 |   │   ├── DiagnosisCreator.vue
64 |   │   ├── HospitalDoctorPicker.vue
65 |   │   ├── MedicinePicker.vue
66 |   │   ├── PrescriptionCreator.vue
67 |   ├── data
68 |   │   ├── agency.json
69 |   │   ├── data.json
70 |   │   ├── metro.json
71 |   │   ├── metroChina.json
72 |   │   ├── news.json
73 |   ├── layouts
74 |   │   ├── README.md
75 |   │   ├── default.vue
76 |   │   ├── error.vue
77 |   │   ├── print.vue
78 |   ├── netlify.toml
   |   └── nuxt.config.ts
```



```

80 |— package.json
81 |— pages
82 |   |— README.md
83 |   |— about.vue
84 |   |— checkin.vue
85 |   |— flow.vue
86 |   |— index.vue
87 |   |— login.vue
88 |   |— logout.vue
89 |   |— manage.vue
90 |   |— print
91 |     |— flow.vue
92 |   |— query.vue
93 |   |— report.vue
94 |   |— statistics.vue
95 |   |— worker.vue
96 |— plugins
97 |   |— axe.ts
98 |   |— chartjs-adapter-dayjs.ts
99 |   |— vue-chart.ts
100 |   |— vuetify.ts
101 |— static
102 |   |— apple-touch-icon-precomposed.png
103 |   |— confirmed-cases-table.svg
104 |   |— covid.svg
105 |   |— data
106 |     |— 130001_tokyo_covid19_patients.csv
107 |   |— er.png
108 |   |— facebook.png
109 |   |— favicon.ico
110 |   |— flow.png
111 |   |— flow_sp.png
112 |   |— github.png
113 |   |— icon.png
114 |   |— line.png
115 |   |— logo.svg
116 |   |— ogp.png
117 |   |— parent.svg
118 |   |— selectmenu.svg
119 |   |— site-qr.svg
120 |   |— sw.js
121 |   |— twitter.png
122 |— tsconfig.json
123 |— utils
124 |   |— colors.ts
125 |   |— formatConfirmedCases.ts
126 |   |— formatDate.ts
127 |   |— formatGraph.ts
128 |   |— formatTable.ts
129 |— yarn.lock

```

网站前端采用 `vue.js` 架构，整个项目使用组件化的设计规范编写，并使用了 `nuxt` 实现高性能的 SSR（服务端渲染）。使用 `vuex` 来实现全局路由，通过 `nuxt-ts` 将项目中的所有路由生成静态的文件托管在服务器上。

项目使用 `vuetify` 作为主要的 UI 库，实现了现代化的遵循 Material Design 设计风格规范的交互界面。使用了 Google Font API，将 `Ubuntu` 字体注入进项目。

使用 `axios` 来实现和后端的异步请求，使用 `chart.js` 来绘制项目中使用的可视化图表。在开发过程中还使用到 `date-fns`、`mdi-icons`、`vue-spinner` 等 dependency。用户的认证信息存储在浏览器的本地 cookie 中，同时使用了 `sessionid` 来后端作为请求凭据，双重认证用户身份，确保没有登录的用户不能对病患做出非法修改。

使用 `vuetyfy` 的响应式布局系统，本项目支持从移动设备上访问，并同样能够取得较好效果。



COVID-19

Covid-19 疫情分析与管理

系统

Covid-19 Analysis &
Manage System您已登录为
admin。

患者状态

治疗中

管理

管理

#

2

姓名

张淑兰

性别

女

生日

1939-03-10

发病地点

新疆维吾尔...

发病日期

2020-02-13

确诊日期

2020-02-13

确诊医院

乌鲁木齐第...

负责医生

全晨

患者状态

已康复

管理

管理

#

3

姓名

欧阳秀珍



- Backend & Database

后端使用经典的 Java 框架 SpringBoot 编写。主要文件结构图：

```
1 | backend
2 | └─ BackendApplication.java
```

```
3 | config
4 |   | ErrorEnums.java
5 |   | GlobalCorsConfig.java
6 |   | SwaggerConfig.java
7 |   | WebConfig.java
8 | controller
9 |   | BaseController.java
10 |   | GlobalExceptionHandler.java
11 |   | IndexController.java
12 |   | account
13 |     | GetUserInfo.java
14 |     | LogIn.java
15 |     | LogOut.java
16 |     | SignUp.java
17 |     | UpdateUserInfo.java
18 |   | diagnosis
19 |     | CreateDiagnosis.java
20 |     | DeleteDiagnosis.java
21 |     | GetDiagnosisInfo.java
22 |     | UpdateDiagnosis.java
23 |   | doctor
24 |     | CreateDoctor.java
25 |     | DeleteDoctor.java
26 |     | GetDoctorInfo.java
27 |     | UpdateDoctor.java
28 |   | hospital
29 |     | CreateHospital.java
30 |     | DeleteHospital.java
31 |     | GetHospitalInfo.java
32 |     | UpdateHospital.java
33 |   | image
34 |     | CreateImage.java
35 |     | DeleteImage.java
36 |     | GetImageInfo.java
37 |     | UpdateImage.java
38 |   | medicine
39 |     | CreateMedicine.java
40 |     | DeleteMedicine.java
41 |     | GetMedicineInfo.java
42 |     | UpdateMedicine.java
43 |   | patient
44 |     | CreatePatient.java
45 |     | DeletePatient.java
46 |     | GetPatientInfo.java
47 |     | UpdatePatient.java
48 |   | prescription
49 |     | CreatePrescription.java
50 |     | DeletePrescription.java
51 |     | GetPrescriptionInfo.java
52 |     | UpdatePrescription.java
53 | dao
54 |   | DiagnosisMapper.java
55 |   | DoctorMapper.java
56 |   | HospitalMapper.java
57 |   | ImageMapper.java
58 |   | MedicineMapper.java
59 |   | PatientMapper.java
60 |   | PrescriptionMapper.java
61 |   | UserMapper.java
62 | model
63 |   | Diagnosis.java
64 |   | Doctor.java
   |   | Hospital.java
```

```

65 | └─ Image.java
66 | └─ Medicine.java
67 | └─ Patient.java
68 | └─ Prescription.java
69 | └─ Result.java
70 | └─ User.java
71 | └─ service
72 |   └─ BaseService.java
73 |   └─ account
74 |     └─ GetUserInfoService.java
75 |     └─ LoginService.java
76 |     └─ LogoutService.java
77 |     └─ SignupService.java
78 |     └─ UpdateUserInfoService.java
79 |   └─ diagnosis
80 |     └─ CreateDiagnosisService.java
81 |     └─ DeleteDiagnosisService.java
82 |     └─ GetDiagnosisInfoService.java
83 |     └─ UpdateDiagnosisService.java
84 |   └─ doctor
85 |     └─ CreateDoctorService.java
86 |     └─ DeleteDoctorService.java
87 |     └─ GetDoctorInfoService.java
88 |     └─ UpdateDoctorService.java
89 |   └─ hospital
90 |     └─ CreateHospitalService.java
91 |     └─ DeleteHospitalService.java
92 |     └─ GetHospitalInfoService.java
93 |     └─ UpdateHospitalService.java
94 |   └─ image
95 |     └─ CreateImageService.java
96 |     └─ DeleteImageService.java
97 |     └─ GetImageInfoService.java
98 |     └─ UpdateImageService.java
99 |   └─ medicine
100 |     └─ CreateMedicineService.java
101 |     └─ DeleteMedicineService.java
102 |     └─ GetMedicineInfoService.java
103 |     └─ UpdateMedicineService.java
104 |   └─ patient
105 |     └─ CreatePatientService.java
106 |     └─ DeletePatientService.java
107 |     └─ GetPatientInfoService.java
108 |     └─ UpdatePatientService.java
109 |   └─ prescription
110 |     └─ CreatePrescriptionService.java
111 |     └─ DeletePrescriptionService.java
112 |     └─ GetPrescriptionInfoService.java
113 |     └─ UpdatePrescriptionService.java
114 | └─ utils
115 |   └─ Authentication.java
116 |   └─ Encrypt.java
117 |   └─ ScheduledTask.java
118 |   └─ Utils.java
119 |

```

项目主要使用了 Controller - Service - Mapper - Model 四层架构，结构清晰，易于维护。加入了全局日志记录，滚动式记录到 log 文件夹，配置了全局错误处理 `GlobalExceptionHandler`，便于调试开发中出现的问题。

项目使用 `maven` 来管理依赖。使用 `Swagger UI` 生成易于阅读和调试的 API 文档，使用基于注解的 `MyBatis` 构建 dao 层的 Mapper，映射数据库的查询关系，使用 `Lombok` 来更加方便地生成 Model 内的 setter、getter 等方法，快速完成 bean 的编写，使用 `pagehelper` 来实现 Controller 返回结果的分页。同时配置了同源策略，确保了安全性。

项目配置了全局的返回类 `Result`，确保了返回格式的统一。所有接口的统一返回格式如下：

```
1 {
2     "code": 200,           // 返回代码
3     "data": [],           // 返回数据
4     "message": "",        // 返回的消息
5     "requestid": "",      // 请求 ID
6     "success": true,      // 是否成功
7     "timestamp": "",      // 请求时间戳
8     "totalCount": ""      // 数据总数（仅在分页时适用）
9 }
```

数据库采用 MySQL Community Version 8.0.12, 存有两万余医生数据，十三万患者数据，四十万余条处方数据，近四百万余条诊断记录。经过多重索引和外键优化，平均性能仍较优秀。详细的数据库架构及简介参见：[数据库结构文档 - database.md](#)

- Script

为了给数据库中填入初始数据，需要生成一些列的演示数据。使用 Python 中的 `faker` 库来生成虚假的个人信息，基于丁香园的疫情真实数据来生成医院、医生数据。使用 `request` 库持续向后端发送批量请求，来完成数据库的初始化。生成的数据中，具有的几种诊断模板为：

```
1 diagnosis = [
2     {
3         "temp": random.randint(360, 370) / 10,
4         "words": "病情稳定，情况良好。",
5         "dna": 0
6     },
7     {
8         "temp": random.randint(370, 380) / 10,
9         "words": "轻微发烧症状，呼吸略有困难，需要进一步确定情况。",
10        "dna": 0
11    },
12    {
13        "temp": random.randint(370, 380) / 10,
14        "words": "已确诊为新冠肺炎，病状较轻，应注意控制以避免病情恶化。",
15        "dna": 1
16    },
17    {
18        "temp": random.randint(370, 380) / 10,
19        "words": "呼吸略有困难，应当辅助药物治疗，并保证已经被隔离。",
20        "dna": 1
21    },
22    {
23        "temp": random.randint(380, 390) / 10,
24        "words": "温度较高，应特别关注病情，有突发情况及时处理。",
25        "dna": 1
26    },
27    {
28        "temp": random.randint(380, 390) / 10,
29        "words": "温度较高，但核酸检测阴性，目前应当按照普通流感和肺炎治疗处理，并考虑再次核酸检测避免误诊。",
30    }
31 ]
```

```

30     "dna": 0
31   },
32   {
33     "temp": random.randint(390, 395) / 10,
34     "words": "状态很危险，应辅助呼吸治疗，必要时转入重点监护病房。",
35     "dna": 1
36   },
37   {
38     "temp": random.randint(395, 412) / 10,
39     "words": "状态极其危险，应作为重点监护对象，保持密切关注，辅助生命维持设备。",
40     "dna": 1
41   },
42 ]

```

更多文档

[API 文档 - api.md](#) (或在 <http://8.210.248.203/docs.html> 查看在线版本)

[数据库结构文档 - database.md](#)

[需求文档 - demands.md](#)

[部署指南 - deployment.md](#)

特别鸣谢

本项目的前端部分参考了基于 MIT 协议的开源项目「[東京都 新型コロナウイルス感染症対策サイト](#)」。原项目仅仅提供了日本疫情数据的一些可视化的曲线展示，我们在该项目的基础上针对中国的疫情数据进行了大幅修改，并添加了后端，实现了与后端的对接。实现了针对每个具体病人的详细信息的增删改查、处方、诊断记录等的检索与修改功能，以及针对医生和医院能否对病患进行增删改的用户权限系统。

国内疫情数据主要来源为丁香园每日疫情统计数据。国际疫情数据主要来源为 WHO 每日疫情公告。

Credit

姓名	学号	联系方式	主要分工
陈俊潼	10185101210	Bill.Chen@live.com	前端、后端、脚本
包梁	10185101281	964131617@qq.com	后端、数据库设计、文档
谢嘉东	10185101247	xiejiaotong0623@gmail.com	前端、数据库设计、文档

East China Normal University

2020.8