文件编号：X-theme-SWC2020-20200266
受控状态：■受控　□非受控
保密级别：□公司级 □部门级 ■项目级 □普通级
采纳标准：CMMI DEV V1.2

FF-*AID*

项目开发文档

Version 8.0

2020.6.10

Written by X-theme

目录

记录更改历史

序号	更改原因	版本	作者	更改日期	备　注
1	创建	1.0	队员 A	2019.12.14	撰写前两章
2	创建	2.0	队员 B	2019.12.16	撰写第三四章
2	更新	3.0	队员 C	2019.12.18	更新项目概述
3	更新	3.1	队员 B	2019.12.20	更新需求分析
4	更新	3.2	队员 D	2019.12.22	更新开发计划
5	更新	4.0	队员 A	2019.12.30	整体细节修改
6	创建	5.0	队员 A	2020.03.20	撰写复赛内容
7	更新	6.0	队员 B	2020.04.20	完善复赛内容
8	更新	7.0	队员 B	2020.05.17	撰写决赛内容
9	更新	7.1	队员 B	2020.05.30	调整整体内容
10	更新	8.0	队员 B	2020.06.10	完善全部内容

1 项目概述

“**FF-AID**”是一款基于多设备互联的**智能急救**快应用。即点即用，**一键求救**。根据用户对突发疾病的语音描述，自动提取情景和症状关键词列表，查找相关疾病，并结合人脸识别匹配的患者既往病史等信息，计算给出置信度前三的疾病列表，完成**病情初判**。智能生成可能伴随病症待用户确认，以完善病情描述，得出**二次判断**。响应“互联网+医疗”形式，以**最及时最准确**的**病情判断**和**急救指导**提高患者的生存率。

1.1 项目背景

近来年轻的程序员，演员，医生猝死的新闻高频出现，**猝死、突发疾病**已经成为社会热点，引发广泛关注。而这些事件背后并非是偶然。

(1) 突发急症事件逐年剧增：猝死是急诊科的常见急症，其中心源性猝死占据较大比重，据不完全统计，近年来我国每年各类急病发病人数超过**百万**，其中单心源性猝死的发病人数即超过 **50 万**人。随着近年来我国经济的快速发展，人们对工作，劳动的热情日益剧增。普遍的 996，加班现象存在而导致的年轻公民健康问题日益严重，而同时随着我国老龄化进程加速，老年人的健康也成为了社会的一大问题。但在这样的背景下，公民的急救理念却并没有跟上，进而导致对突发疾病的不知所措，病人在黄金时间内得不到有效帮助，也就有了悲剧的不断发生。

(2) 黄金急救时间的重要性：国内外文献均显示，第一目击者的反应是影响心肺复苏的首位因素，猝死后 **4 min** 内复苏成功率高达 **50%**，**6 min** 后复苏的成功率降到 **4%**，**10 min** 后复苏成功率几乎为 **0**。研究显示，受交通拥挤、急救中心分布等影响，我国的呼救至到达现场时间平均为 **10～13 min**，与国家规定的反应时间 **5～7 min** 相差较大，这种情况下，第一目击者的作用显得尤为重要。受掌握的知识影响，第一目击者现场急救的心肺复苏成功率较低，正确的电话沟通和指导，根据具体的病情指导第一目击者常用救助方法，可提高猝死的救治成功率。

和猝死一样，诸如癫痫，中风，食物中毒等所有急症的急救能否成功，绝大意义上取决于能否第一时间判断出患者的病因，取决于**前 4-6 分钟黄金时间**的急救措施是否得当。

1.2 项目定位

1.2.1 应用场景

主要应用于当有人突发急病，或遇到意外需要救援时，一键求救，向 120 急救中心和患者的紧急联系人发送求救信息，并通过分析及时给出紧急施救措施，为第一目击者在第一时间提供准确有效的急救指导，提高病患的救治成功率。适用于所有急救需要的场景：

1. **发现陌生人突发疾病**

2. 自己突发疾病，自救

3. 发现家人，亲友突发疾病

1.2.2 目标人群

全部人群，本项目致力于提供简洁易用的急救应用，以极低的学习成本，号召所有年龄段的人群使用。期望实现全民都可以通过使用“FF-AID”实现对周围发病人员第一时间的急救，以及全民都可以在突发疾病的时候得到有效急救帮助的理想。

1.3 项目方案

针对不同使用场景，设计了不同的急救实施方案：

1. 发现陌生人突发疾病。

作为病人的第一目击者，当发现有人突发疾病，需要急救帮助的时候，通过 FF-AID 一键求救，直接向 120 急救中心发送求救信息，接着语音输入患者病症，人脸识别获得患者信息，向患者紧急联系人发送求救消息。FF-AID 结合用户输入的病症和患者的年龄，性别，既往病史智能判断患者病因，并给出急救指导。在尽可能短的时间内对实现对患者的急救帮助。

2. 自己突发疾病，自救

当自己突发不适的时候，通过 FF-AID 一键求救向 120 急救中心和自己的紧急联系人发送求救信号，避免因无力拨打 120 求救而错过自救机会。

3. 发现家人，亲友突发疾病

当发现家人，亲友突发疾病时，通过 FF-AID 一键求救，向 120 急救中心发送求救信息，如果已知家人亲友的病因，可以直接语音输入疾病名称，应用第一时间给出对应疾病的急救指导。若不知道，则同样可以进行智能病因判断。

1.4 项目目标

本项目以提供第一时间的急救服务为宗旨，致力于开发出一款通过知识图谱和自然语言处理等技术，集方便，准确，高效，迅速于一体的智能急救型应用。填补了我国急救服务方面的空白，解决了大量突发疾病患者因为在黄金时间里没有得到有效施救而错过最佳治疗时间的问题，相比于传统的急救方式有了跨越性的进展。

1.5 项目价值

本项目的价值主要体现在填补了医疗急救领域一直以来存在的空白，尤其是正在高速发展的中国社会，出于对进步和美好生活的渴求，人们对工作，劳动的热情日益剧增，更是有普遍的加班现象存在，996 的工作制度也极为常见。

而在这么个大环境的基础下，我国突发猝死的事件逐年剧增，其他各种疾病的发病人群也越来越庞大，我国急救需求越来越大，但对于急救的普及与发达国家长久以来的观念科普还有相当大的差距，只有很少一部分人掌握了急救技巧。

基于国家发展中的这些痛点，FF-AID 提供准确的病情判断和急救指导功能，带来智能迅速的急救服务，为国家健康发展，国民健康工作提供保障。

2 开发计划

2.1 最终呈现形式

作品最终将呈现为一个基于快应用框架的完整 APP 应用，该 APP 运行良好，且将用于所有预期功能，包括：用户的注册登录，突发情况下的一键求救功能，重大传染病爆发期间智能的传染病诊断等，同时也可以给国家和政府相关机构提供大量相关数据。即我们的项目呈现的是一个基于智能诊断算法和强大数据分析能力的及时快速的线上急救助手，在最短的时间内提供最有效的急救指导。

2.2 主要功能描述

一键求救模块：

考虑到用户(施救者)和被施救者的关系不同，我们将一键求救的过程分为已知病因和未知病因两个流程。

当用户(施救者)发现有人突发疾病需要急救时，用户点击一键求救，之后可以选择是否向 120 急救中心发送求救信息，如果点击确认发送就可以将 APP 获取到的地址信息以短信的形式向 120 中心求救。选择过后，进入智能急救辅导界面，通过最上面的按钮来选择是否知道被施救者的发病原因。

若用户未知被施救者的发病原因，用户通过语音输入被施救者的症状说明，同时可以选择对被施救者进行人脸识别，获取他的基础信息，既往病史和紧急联系人。APP 将通过对症状的提取和分析并结合被施救者的既往病史，年龄，性别给出智能的疾病初判，接着用户可以选择其他症状进行进一步判断以获得更准确的结果，也可以直接选择病症进入急救指导。

若用户已知被施救者的发病原因，用户可以直接通过语音输入病因，APP 将直接通过病因检索数据库，给出对应的急救指导。

流行疾病诊断模块：

用户输入自己的体温以及选择 APP 给出的症状，结合用户的体温和诸多症状表现智能判断用户获得某种流行传染病的可能性，给出结论。同时结合用户长时间的持续检测监控的综合判断，智能匹配传染病不同发展阶段的不同症状，并结合用户所在地址(可以精确到小区)的传染病传播情况，给出不断更新的结论。

同时，我们根据用户的诊断结果给出阶段性的建议，充分保障用户及其家人的健康。必要时还可以通过大量数据的分析，对疫情严重区域的用户发送预警信息，提供建议。

2.3 运行环境

客户端程序：OPPO 等支持快应用的各类 Android 终端手机，要求安卓版本在 4.4 以上

后端服务：部署在云服务器上，服务器要求：

1. 处理处理器 PentiumII 及以上
2. 内存要求 512MB 以上

2.4 验收标准

验收标准如下：

1. 病情初判的准确度达到 90%以上
2. 病情二次判断的准确度达到 95%以上
3. 病情判断允许和实际的具体病因有一定的误差，即要求疾病的类属准确
4. 给出的急救指导符合判断出来的病因
5. 传染病诊断的患病概率的准确率偏差在 5%以内
6. 在重大传染病疫情期间，给与健康和可能患病的用户准确的建议和指导
7. 界面整洁大方，清晰美观
8. 项目文档完整正确
9. 应用所有实现的功能完整可用，并且流畅易用，没有使用基础的用户也可以流畅使用

2.5 关键问题

问题：

（1） 技术焦点：项目技术的核心基于传统的医疗知识图谱，但是传统医疗知识图谱的

问题在于无法得知疾病-症状的相关性，在我们的运用场景中无法发挥良好效用。

（2） 开发问题：初期由于我们对快应用的前端开发不够熟悉，分配的人手也不够，导致开发进度缓慢。

（3） 财务风险：由于我们的产品是线上运营的，初期我们项目的成本主要是服务器的租赁费用和推广费用，这些是必要投资。并且由于我们的产品公益性较强，前期不具备太大的创收能力，需要额外资金来进行推广与用户吸引。同时目前正是传染病(新冠)疫情严重的时期，初期预期我们的产品将可能有大量用户的涌入，更加加大了我们租赁和维护服务器的投入费用。

（4） 行业风险：由于我们的产品是创新创意类的产品，目前市场上并没有成熟的落地项目，但是也容易被抄袭，且由于我们还在起步阶段，面对资金短缺人手不够的问题，难以做出有效应对。

（5） 运营风险：作为一个学生团队，我们缺乏面对风险的能力，也缺乏良好的运营能力，且由于创业初期经验不足，实际管理经营控制机制不完善，且资金不稳定，导致自信等级不高，有较高的信用风险。

对策：

（1） 出于准确率的考量，在医疗知识图谱构建的基础上，我们采用自建 word2vec 模型计算病症-疾病相关性，通过学习大量疾病、症状数据来获取词语的特征向量表示，进而使用余弦距离衡量相关性，结合自主研发的智能疾病检测算法，提供高准确度，高智能性的急救服务。

（2） 对于项目开发过程中遇到的实际困难，我们尝试调整人手分配，更多将人力投入到进展缓慢部分的开发中，并及时交流反馈开发经验，增加会议次数，及时将问题解决，避免在某个部分停滞过长时间。

（3） 面对财务风险，我们将充分发挥我们作为一个大学生创新创业团队的优势，积极争取各种形式的创业基金，并且把握我们产品的公益性质，争取获得社会认可及政府的扶持。同时我们还必须加强财务人员的风险防范意识，做好风险回避，风险转移等风险管理控制规划，建立完善的财务预警系统，尽早定制相应的对策规避或化解风险。

（4） 面对行业风险，我们将对我们的创意和技术申请知识产权，以实现法律保障，同时我们将全身投入于产品的优化和提升中，确保自己的核心竞争力。

（5）面对运营风险，我们将在后期扩展队伍，引入擅长于经营管理的合伙人，提高我们的运营能力。目前情况下我们会积极向相关人士请教，极力避免可能出现的风险。

2.6 进度安排

任务名称	任务概述	拟定开始日期	拟定结束日期
需求分析	详细分析本系统的功能需求、业务需求等、界定本系统的边界。	2019.10.10	2019.10.20
界面原型设计	初步设计出本系统快应用的前端界面原型。	2019.10.20	2019.10.24
算法分析及实现	分析及实现本系统所需的人脸识别、语音识别、病情初断等算法。	2019.10.20	2020.02.15
系统测试	测试本系统的功能、性能等是否满足需求。	2019.11.25	2020.02.15
系统部署	部署本系统。	2020.02.15	2020.02.20
第一次迭代	基于当下疫情现状，考虑实现传染病的智能诊断(复赛新增进度)	2020.02.20	2020.03.10
第二次迭代	对项目进行优化	2020.03.10	2020.03.30
第三次迭代	二次优化	2020.03.30	2020.04.20
第四次迭代	基于急救科医生的专业指导和建议，对项目的立意和逻辑进行再次思考和改进	2020.05.03	2020.05.20
第五次迭代	完善流行病诊断模块的逻辑和流程	2020.05.20	2020.06.02
完整版迭代	完善所有细节内容，系统测试	2020.06.02	2020.06.10

表 1：进度安排表

2.7 开发预算

（1）服务器成本：由于近期是传染病(新冠)疫情严重的时期，我们初期预计将有 30w 左右的用户，同时段活跃人数预计高达 8w 人，我们预想满足高并发需求，故而预期需要有四台带宽 100M，防御 200G，内存 16G 的服务器，每台年租费用预估 2.5w，共计 10w。

（2）研发成本：由于我们的产品是技术推动型的创新产品，需要确保对研发的投入，未来预估每年投入 20w 用于更加先进算法的研发和改进。

（3）人力成本：我们的产品是具有紧急性的产品，因此我们必须保证及时满足用户的需求，及时收到用户的反馈，需要客服人员和系统维护人员来及时接受反馈，并调整问题，目前由团队成员兼任。

（4）经营成本：作为一个需要互联网导向的新兴产品，我们需要得到大量的关注和传播，才可以让用户在紧急情况的第一时间考虑到我们提供的服务。因此前期的推广费用是必须投入的，我们需要做大量网络宣传。预计每年花费 8w。

3 可行性分析

3.1 技术可行性分析

知识图谱构建

Google 公司在 2012 年提出了“知识图谱（KnowledgeGraph）”的概念，用于构建提升搜索引擎性能的知识库。现在，知识图谱泛指知识库项目。从上世纪 60 年代语义网络发展起来就已经出现了知识图谱的研究，并分别经历了 1980 年代的专家系统、1990 年代的贝叶斯网络、2000 年代的 OWL 和语义 WEB，以及 2012 年 Google 的知识图谱。

知识图谱发展至今出现了很多开放领域或者专业领域的知识图谱。

（1）XLore

清华大学的 XLore 是第一个大规模的中英文知识平衡的知识图谱，使用更大的异构在线维基百科来丰富中文知识，并使用基于分类的方法构建能够准确将维基百科分类的系统，和采用跨语言知识链接方法实现实体之间心得跨语言链接。

（2）CN-DBpedia

复旦大学的 CN-DBpedia 是中文通用百科知识图谱，主要从中文百科类网站，经过滤、融合、推断等操作后，最终形成高质量的结构化数据，供机器人和人使用。

（3）YAGO

YAGO 是由德国马普研究所研制的链接数据库。YAGO 主要集成了 Wikipedia、WordNet 和 GeoNames 三个来源的数据。YAGO 将 WordNet 的词汇定义与 Wikipedia 的分类体系进行了融合集成，使得 YAGO 具有更加丰富的实体分类体系。YAGO 还考虑了时间和空间知识，为很多知识条目增加了时间和空间维度的属性描述。目前，YAGO 包含 1.2 亿条三元组知识。YAGO 是 IBM Watson 的后端知识库之一。

在垂直领域方面，医学领域是知识图谱应用最为广泛的领域之一。如上海曙光医院构建的中医药知识图谱、本体医疗知识库 SNOMED-CT，IBM Watson Health 等应用近几年开始进入人们视线。知识图谱在医学知识检索与医疗文本挖掘中受到关注和研究。如何通过庞大的医学知识图谱进行智能医疗诊断已成为热点研究对象。

（4）Bayesian Networks 错误!未找到引用源。

此方法通过构建 disease-symptom 实体联系网络并运用贝叶斯的方法对疾病进行推断。

（5）CrossCR 错误!未找到引用源。

CrossCR 在传统 disease-symptom biapartite graph 的基础上引入 disease domain 和 disease

domain，从而考虑到疾病与疾病、症状与症状的相关性，减少了模型计算中的损失。同时，通过聚类的方法使得输出为一组疾病，增强了输出的可读性和可解释性。

（6）KGDeL 错误!未找到引用源。

KGDel(disease diagnosis method combining knowledge graph and deep learnin)即基于病情描述文本的疾病诊断方法，通过加入医学知识图谱中的医生经验的主观知识，使得 CNN 在训练病情描述文本客观数据时能学习到更全面更直接相关的疾病特征，从而提高不同类型疾病的诊断准确率，更适用于初步诊断病情描述的疾病。

word2vec 模型构建

词嵌入模型也是 NLP 领域的研究热点之一，旨在通过学习大量文本数据来获取词语的特征向量表示。word2vec 模型是词嵌入的常用模型。本系统通过使用 word2vec 学习疾病、症状的向量表示，进而使用余弦距离衡量相关性。

医疗图谱中任意两个实体的相关性计算方法如下：

$$W(E_1, E_2) = (W_{shortestPath}(E_1, E_2) + W_{word2vec}(E_1, E_2))/2$$

即任意两个实体的相关性为通过最短路径方法计算的相关性和 word2vec 模型计算出的实体特征的词向量的相关性的平均值。

智能诊断算法

智能诊断算法属于机器学习领域的内容，由团队自主研发，旨在根据用户输入的症状等信息给出最有可能的疾病。本系统基于医疗知识图谱，构建症状-疾病关系网络，并借助基于聚簇的排序算法，构建智能诊断系统智能诊断系统的构建分为三个阶段：症状-疾病聚簇、簇排序、簇中节点（疾病）排序。如图 1 所示，左边为症状域，右边为疾病域。每个域中节点之间的边表示两个节点之间的相关性大小。横跨两个域之间的边代表症状与疾病之间的相关性。

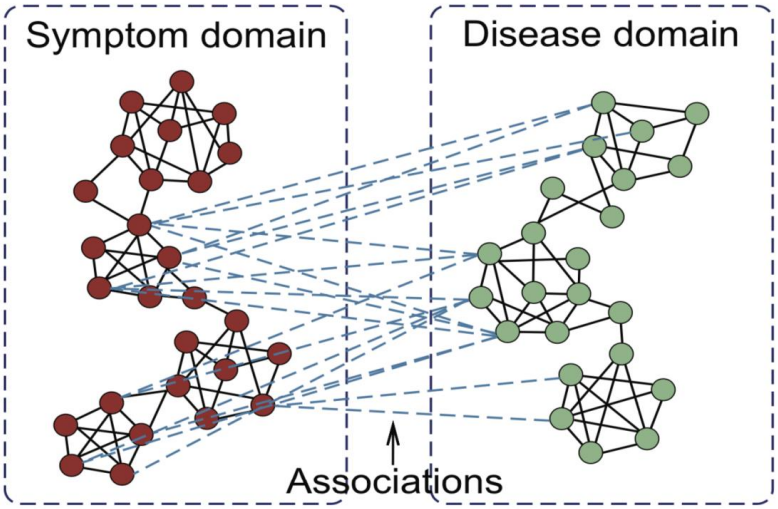


图 1：症状-疾病关系网络

3.2 资源可行性分析

（1） 数据资源可行性分析：产品的研发需要大量的疾病-症状数据，以及疾病对应的急救措施数据。目前并没有找到已有的数据集，但是有大量的医疗网站可以做数据的收集，目前已从 12 个网站收集到近 50w 条真实的医疗数据。

（2） 人力资源可行性分析：目前有两名软件工程专业技术人员，两名数字媒体技术设计人员，确保产品的策划，设计，研发按时推进。

3.3 市场可行性分析

市场调研：

在长达两个月的市场调研过程中，我们进行了大量的问卷投放，线上意见采集，并在几个线下医院急诊科实地调研，统计急救数据信息，并对和一线医生进行交流。我们发现约有**82%**的医疗急救由于第一时间病人没有得到有效的救助而导致病人病情恶化，约有 20%的病人因此失去生命，在心源性猝死等发病迅速，病情恶化迅速的病情上，没有第一时间的急救指导，致死率更是高达近 90%，令人惋惜。

竞品分析：

市场上和我们性质类似的急救 APP 存在大量问题：

1. 市面上所有急救性质的 APP，诸如“爱心急救”，“辰邦急救”，都没有第一时间的急救指导，只是指望于附近的救助员来实施急救，考虑到救助员的分布较少，求救不够智能，市场占有率低。
2. 市面上所有急救性质的 APP 都是基于 Android 或者 IOS 的传统应用，无法避免用户下载应用，注册登录的繁琐冗长流程，对于一款急救类型的 APP 来说，时间上的冗长导致这些 APP 的可行性差。

FF-AID：

1. 第一时间提供智能，准确的病情预判和急救指导。
2. 第一时间向 120 急救中心和发病者的紧急联系人发送求救信息。
3. 基于快应用框架，即开即用无需下载，第一时间即可提供服务。
4. 支持负一屏卡片式交互开启方式，使用快捷方便
5. 交互简单简洁，功能完整易用。

结论：

可以发现急救服务在目前的市场上需求量大，但是现有的 APP 都无法发挥良好的作用，市场几乎空白，可以预测到我们的产品一旦进入市场就可以凭借极大优势迅速填补空白，预期可拥有过半市场。

4 需求分析

4.1 数据需求

4.1.1 静态数据

静态数据是指在运行过程中主要作为控制和参考用的数据，它们在很长的一段时间内不会变化，一般不随运行而改变。

1. 后端数据：

定位	名称	类型	初值	备注
文件路径	UPLOAD_PATH	String	/root/imag/picStore	映射到服务器上存储文件的路径
	CHECK_PATH	String	/root/imag/picCheck	映射到服务器上检验文件的路径
状态参数	up_Server	Boolean	True	标识本地调试或是上传服务器
Baidu API	METHOD_ROW	Boolean	false	默认以 json 格式发送音频文件
	APP_KEY	String	EP…IA	语音识别 API 的应用编号
	SECRET_KEY	String	F17…ln2	语音识别 API 的应用秘匙
	RATE	Int	16000	采样率
Neo4j 数据库	URI	String	Bolt://liublack.cn:7687	Neo4j ip 及端口
	USERNAME	String	Neo4j	Neo4j 用户名
	PASSWORD	String	2****1	Neo4j 密码

表 2：后端静态数据表

2. 前端数据：

名称	初值	类型	备注
package	com.xtheme.ffaid	String	应用包名
name	FF-AID	String	应用名称
versionName	1.0.4	String	应用版本名称
versionCode	4	Integer	应用版本号
minPlatformVersion	1020	Integer	支持的最小平台版本号
	system.prompt; system.router;		

features	system.sms; system.geolocation; system.fetch; system.request; system.file; system.image; system.media; system.storage;	Array	配置接口列表
config	designWidth: 375,	Integer	页面设计基准宽度
	network.readTimeout:60000	Number	读取超时时间，单位 ms

表 3：前端静态数据表

4.1.2 动态数据

动态数据包括所有在运行中发生变化的数据以及在运行中需要输入、输出的数据以及联机操作中要改变的数据。

1. **数据库数据:**
   a) 疾病信息：疾病名称，对应的急救措施，相关疾病，相关病症
   b) 用户个人信息：账号，密码，性别，年龄
   c) 既往病史信息：对应的用户，病史描述
   d) 紧急联系人信息：对应的用户，电话号码，相关描述
2. **后端数据：**
   实体参数：从数据库或者前端获得的 Object 成员变量
3. **前端数据：**
   a) 用户注册时输入的用户名、密码、手机号、人脸认证图片、性别和出生年月；
   b) 用户后续完善的既往病史和紧急联系人；
   c) 求救时获取的定位，用户输入的语音和人脸识别图片；
   d) 预判后用户选择的补充病症；
   e) 输出给用户的急救指导；
   f) 流行疾病自测模块输入的疾病名称、症状名称，输出的患病概率、相关建议。

4.1.3 数据词典

1. user（用户信息）表对应的数据词典：

user

字段名	数据类型	默认值	允许非空	自动递增	备注
id	bigint(9) unsigned		no	是	唯一标识id
username	varchar(255)		no		用户的名字
password	varchar(255)		no		用户的密码
sex	tinyint(1) unsigned		no		用户的性别
status	tinyint(1) unsigned	0	no		用户状态（是否在线）
tel	varchar(11)		no		用户的电话号码
pic	varchar(255)		yes		用户头像
birthday	date		yes		用户生日
deleted	tinyint(1) unsigned	0	no		逻辑删除位，为0存在，为1被删除

表 4：user 表数据词典

2. urgentTel（紧急联系人）表对应的数据词典：

urgentTel

字段名	数据类型	默认值	允许非空	自动递增	备注
id	bigint(9) unsigned		no	1	唯一标识id
userId	bigint(9) unsigned		no		这条紧急联系电话对应的用户
urgentTel	varchar(255)		no		紧急联系电话
describ	varchar(255)		yes		紧急联系电话的信息（对应哪个家属）
addtime	datetime		no		这条紧急联系电话添加的时间
deleted	tinyint(1) unsigned	0	no		逻辑删除位，为0存在，为1被删除

表 5：urgentTel 表数据词典

3. illData（既往病史）表对应的数据词典：

illData

字段名	数据类型	默认值	允许非空	自动递增	备注
id	bigint(9) unsigned		no	1	唯一标识id
userId	bigint(9) unsigned		no		这个病史对应的用户
illkind	varchar(255)		no		病史类型
describ	varchar(255)		yes		病史的详细信息
addtime	datetime		no		这条病史添加的时间
deleted	tinyint(1) unsigned	0	no		逻辑删除位，为0存在，为1被删除

表 6：illData 表数据词典

4. aidData（求救信息）表对应的数据词典：

aidData

字段名	数据类型	默认值	允许非空	自动递增	备注
id	bigint(9) unsigned		no	1	唯一标识id
byId	bigint(9) unsigned		no		这条求救信号是由哪个用户发出的
createTime	datetime		no		这条求救信号发出的时间
location	varchar(255)		no		这条求救信号发出的地址
details	varchar(255)		yes		求救信号的信息
isEnd	tinyint(1) unsigned	0	no		这个求救信号是否已经被处理（结束）
endTime	datetime		yes		求救信号结束时间
isSelf	tinyint(1) unsigned		no		是自救还是他救
deleted	tinyint(1) unsigned	0	no		逻辑删除位，为0存在，为1被删除

表 7：aidData 表数据词典

5. epidemicData（流行病检测信息）表对应的数据词典：

字段名	数据类型	默认值	允许非空	自动递增	备注
id	bigint(9) unsigned		no	1900/1/1	唯一标识id
diseaseKind	vachar(255)		no		流行病类型
diseaseRate	int		no		流行病患病概率
symptoms	varchar(255)		no		用户描述的症状
createTime	dateTime		no		本次流行病检测的时间
suggestions	varchar(255)		yes		检测后给出的当前建议
deleted	tinyint(1) unsigned	0	no		逻辑删除位，为0存在，为1删除

表 8：epidemicData 表数据词典

4.1.4 数据采集

1. Python 爬虫下载

   使用 Python 从相关的医疗网站上爬取所需要的疾病信息和相关病症，急救措施等数据，用于训练模型。

   数据量：495,632 条问答信息，417 个急诊科疾病，5499 个症状。

2. 邀请相关的专业人士指导

   通过邀请医疗方面的专业认识讲解急救指导，不仅确保了数据集的准确性，也可以增加产品的置信度。

3. 用户终端设备

   用户向终端手机输入个人信息，人脸数据，病理数据等。

4.2 功能需求

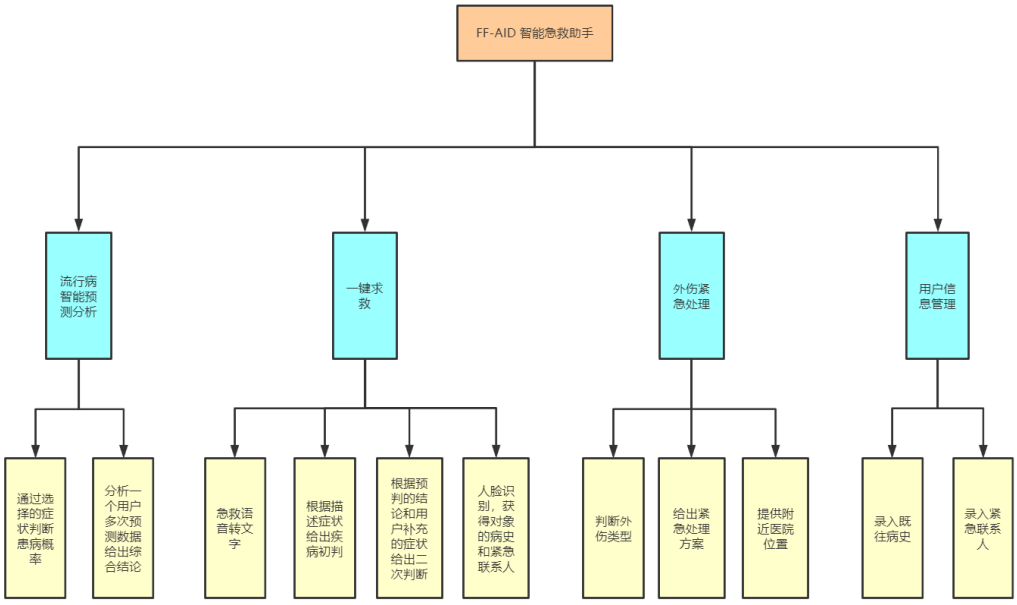


图 2：核心功能模块结构图

4.2.1 一键求救功能模块

功能模块	功能	功能描述	优先级
一键求救	急救语音转文字	将求救语音转为文字形式	中
	人脸识别，获得对象的病史和紧急联系人	根据被施救者的人脸信息识别到对应的用户，并将该被施救者的病史和紧急联系人返回，向被施救者的紧急联系人发送求救信息	高
	根据描述症状给出病情初判	根据用户输入的文字进行症状提取，并结合该用户的病史，给出初步分析结论	高
	根据预判的结论和用户补充的症状给出二次判断	在病情初判的基础上结合用户选择的其他相关症状给出二次判断	高
	给出患者录入的急救药和急救措施	通过人脸识别获得患者输入的急救药位置和针对自己的急救措施（若有）	高
	根据病情给出急救指导和其他信息	根据病情判断的结果，选择病情，并得到对应的急救指导和对应该疾病的症状，并发症，护理等信息	高

表 9：一键求救核心功能模块描述

功能模块用例图：

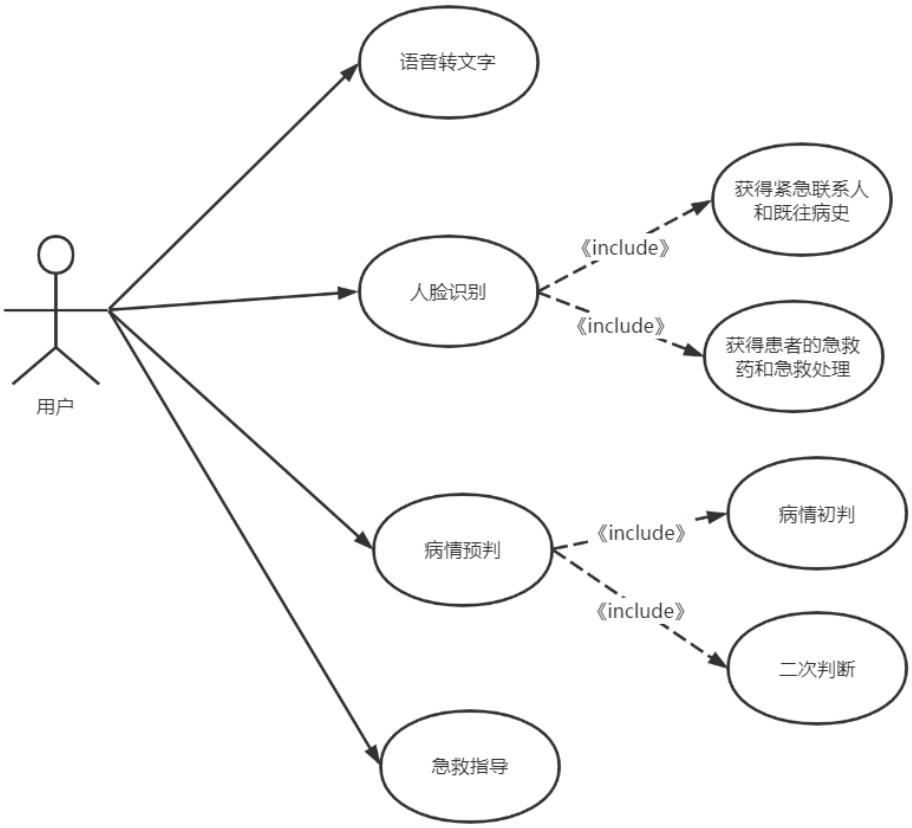


图 3：一键求救功能用例图

核心功能用例规约

用例名称	病情预判-病情初判
功能简述	用户通过输入症状表现的语音表述，结合人脸识别获得的病人信息，系统给出智能病情初判
用例编号	001
执行者	用户(施救者)
前置条件	用户通过卡片或者其他方式打开 APP
后置条件	用户获得病情初判的结果和急救指导
涉众利益	用户希望获得准确的病情预判结果和急救指导
基本路径	1. 用户打开 APP 并点击一键求救 2. 用户选择未知该病人发病原因 3. 用户输入一段病症对应的语音 4. 语音转文字 5. 用户对病人进行人脸识别 6. 用户发送病症检测信息 7. 获得病情初判和急救指导

扩展路径	4.a 用户语音无法识别 提示用户重新输入，或者选择手动输入 4.b 语音识别不准确 提示用户可以手动修改输入文本 5.a 人脸识别失败 提示用户重新识别或者直接发送症状信息 5.b 摄像头打开失败 提示错误并指导用户打开摄像头
字段列表	病症检测信息=用户输入文本+病人既往病史
设计规则	
未解决的问题	响应时延在 2-3s，需要优化
备注	无

表 10：病情初判用例规约

用例名称	病情预判-二次判断
功能简述	用户通过选择应用给出的相关症状，系统结合病情初判的结果给出二次判断的结果和急救指导
用例编号	002
执行者	用户(施救者)
前置条件	用户通过卡片或者其他方式打开 APP，并已进行过病情初判
后置条件	用户获得病情二次判断的结果和急救指导
涉众利益	用户希望通过二次判断获得更加准确的病情判断结果和急救指导
基本路径	1. 用户打开 APP 并点击一键求救 2. 用户选择未知该病人发病原因 3. 用户输入一段病症对应的语音 4. 语音转文字 5. 用户对病人进行人脸识别 6. 用户发送病症检测信息 7. 获得病情初判和急救指导 8. 选择系统给出相关症状 9. 发送更加准确的病症检测信息 10. 系统给出病情二次判断和急救指导
扩展路径	8.a 用户选择错误的病症 返回上一个界面重新选择
字段列表	病症检测信息=病情初判结果+用户选择的相关症状

设计规则	
未解决的问题	响应时延在 2-3s，需要优化
备注	无

表 11：病情二次判断用例规约

4.2.2 流行病智能预测分析功能模块

功能模块	功能	功能描述	优先级
流行病智能预测分析	通过选择的症状判断患病概率	用户根据系统的提示不断选择回答问题，录入体温等等，系统智能判断患病概率，并给出当前阶段的指导和建议	高
	分析一个用户多次预测数据给出综合结论	通过分析一个用户长时间多次的症状选择，体温输入等信息综合判断，结合该流行病不同阶段的特征，给出更具备参考价值的综合结论，并给出阶段性的指导	高

表 12：流行病智能预测核心功能模块描述

功能模块用例图：

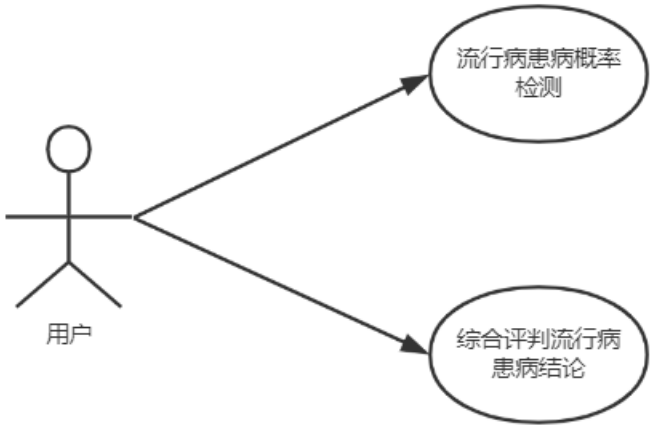


图 3：流行病智能预测功能用例图

核心功能用例规约：

用例名称	流行疾病检测-患病概率检测
功能简述	用户根据系统的提示不断选择回答问题，录入体温等等，系统智能判断患病概率，并给出当前阶段的指导和建议
用例编号	003
执行者	用户
前置条件	用户通过卡片或者其他方式打开 APP

后置条件	用户获得流行病检测结果和当前建议
涉众利益	用户希望获得准确的流行病检测结果和当前建议
基本路径	1. 用户登录(可选) 2. 用户打开 APP 并点击流行疾病 3. 用户选择要检测的流行疾病 4. 用户在系统的指导下不断回答问题，选择答案 5. 必要时输入体温数据 6. 获得流行疾病的患病概率和当前建议
扩展路径	4.a 输入体温格式错误 提示用户输入正确格式的体温 5.a 症状不全 提示用户自己输入
字段列表	
设计规则	
未解决的问题	概率判断可能有一定的误差 建议需要专家判断是否合理
备注	症状包含关联性高和低的症状，用于综合判断

表 13：流行疾病患病概率检测用例规约

用例名称	流行疾病检测-综合评判流行病患病结论
功能简述	通过用户一段时间内的多次自测数据结合该流行病发展不同阶段的症状表现，给出阶段性的患病预测和建议
用例编号	004
执行者	用户(施救者)
前置条件	用户通过卡片或者其他方式打开 APP
后置条件	用户获得流行病检测结果和当前建议
涉众利益	用户希望通过阶段性的评测获得准确的流行病检测结果和阶段建议
基本路径	1. 用户登录(必选) 2. 用户打开 APP 并点击流行疾病 3. 选择阶段性检测 4. 用户选择要检测的流行疾病 5. 选择开始和终止日期 6. 确认每次的症状输入无误 7. 获得阶段性的流行病患病结论和阶段性建议
扩展路径	5.a 选择日期超出范围

	提示用户输入正确的日期范围 6.a 症状输入有误 提示用户选择有效的自测数据，排除有误数据
字段列表	
设计规则	
未解决的问题	概率判断可能有一定的误差 阶段建议需要专家判断是否合理
备注	结合长时间的跟进检测数据，得出信服力较高的概率判断

表 14：综合评判流行疾病患病概率用例规约

4.2.3 信息管理功能模块

功能模块	功能	功能描述	优先级
用户信息管理	录入既往病史	用户录入自己的既往病史，上传	中
	录入紧急联系人	用户录入自己的紧急联系人，上传	中
	录入急救药和明确的急救措施	用户录入自己的急救药和急救措施，上传	中

表 15：用户信息管理功能模块描述

功能用例图：

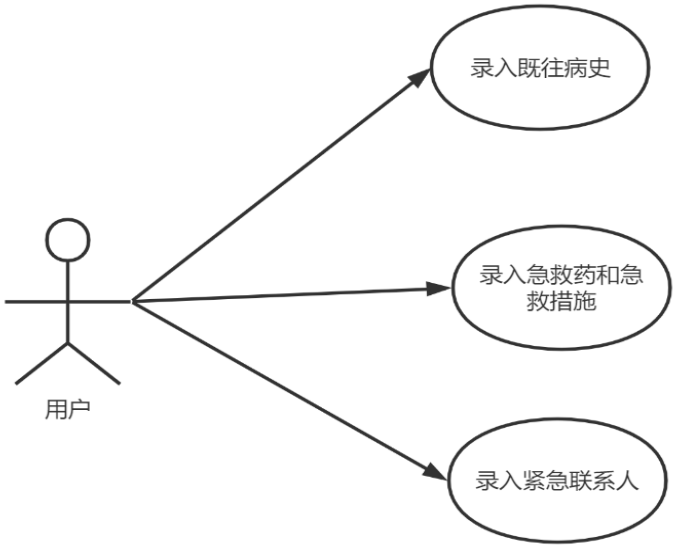


图 4：用户信息管理用例图

4.3 性能需求

4.3.1 时间特性

应用响应时间一般控制在 1-2 秒内。

应用主要的时间开销是在急救时病情判断的时候，通过 FF-AID Net 对用户输入的病症表述进行实时分析，由于病症数据库庞大，关联性复杂，根据输入病症的特性，会有 1-2 秒的相应时延。

更新处理时间：处理用户更新的时间应控制在 0-3s 内，让用户等待的时间不宜过长。

数据转换与传输时间：可以控制在 1s 内。

运行时间：考虑到在服务器上数据处理时间稍长，预期理想的运行时间在 3s 内。

4.3.2 适应性

该系统应满足运行环境在操作系统之间的安全转换和与其他应用软件独立运行的需求。

运行环境的适应性：该系统应具备在不同终端设备环境下，例如 OPPO，小米，HUAWEI，vivo 等不同厂商的设备，并且都具有良好运行的能力。

当开发计划发生改变时，系统程序应具有较容易的可更改性，而非否定原来的整体。

本系统在输入数据时有一定的容错能力，以保证数据的准确可靠，同时尽可能使用更先进的语音识别算法，避免识别不到症状导致的死锁。

4.4 界面需求

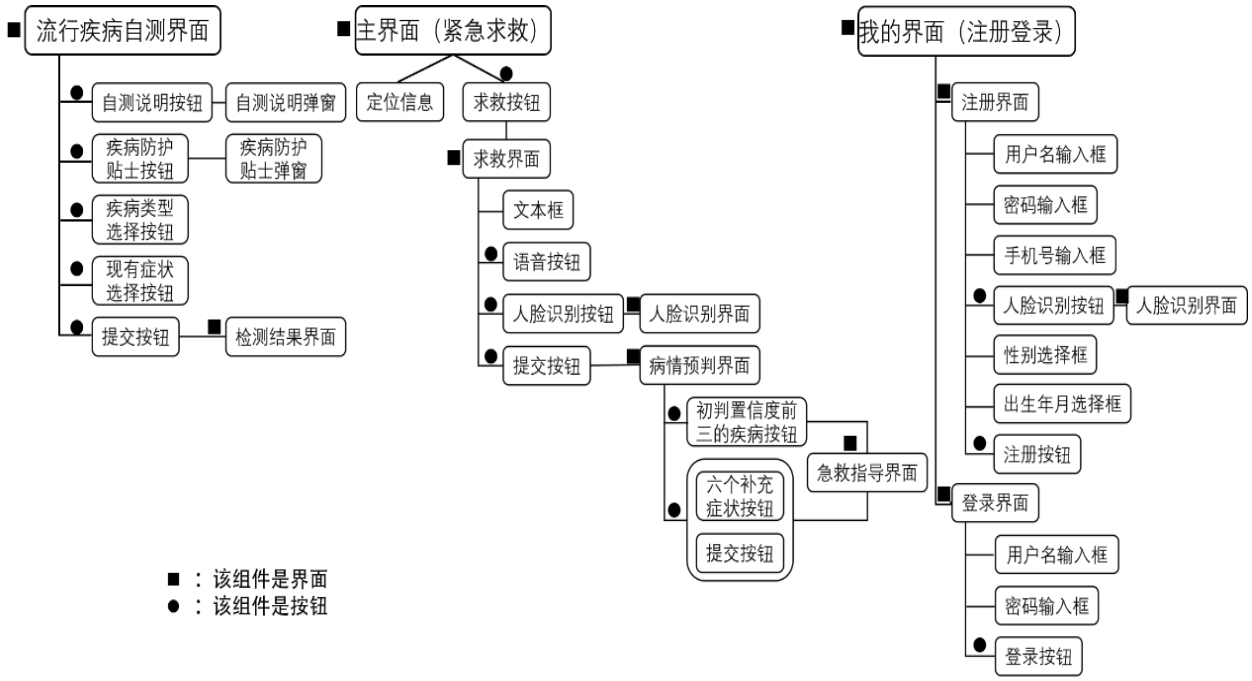


图 5：界面流程图

核心功能模块界面原型：

1. 一键求救：

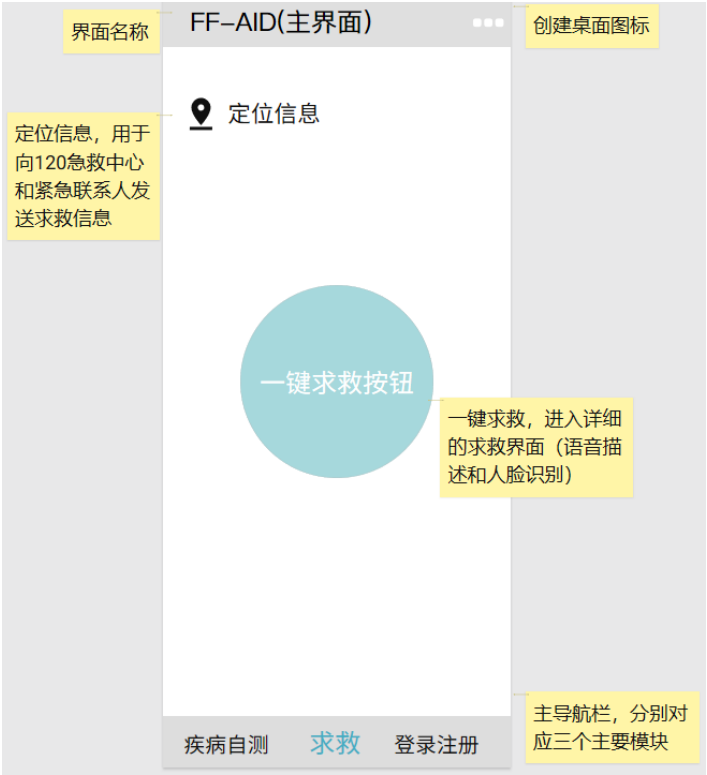


图 6：一键求救主界面

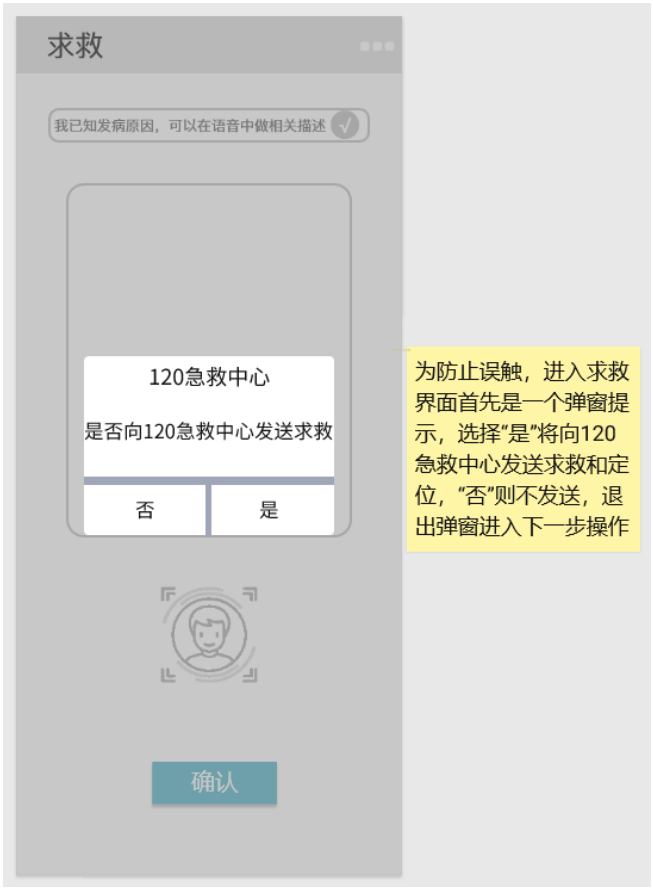


图 7：向 120 求救弹窗

图 8：求救流程界面

图 9：病情预判界面

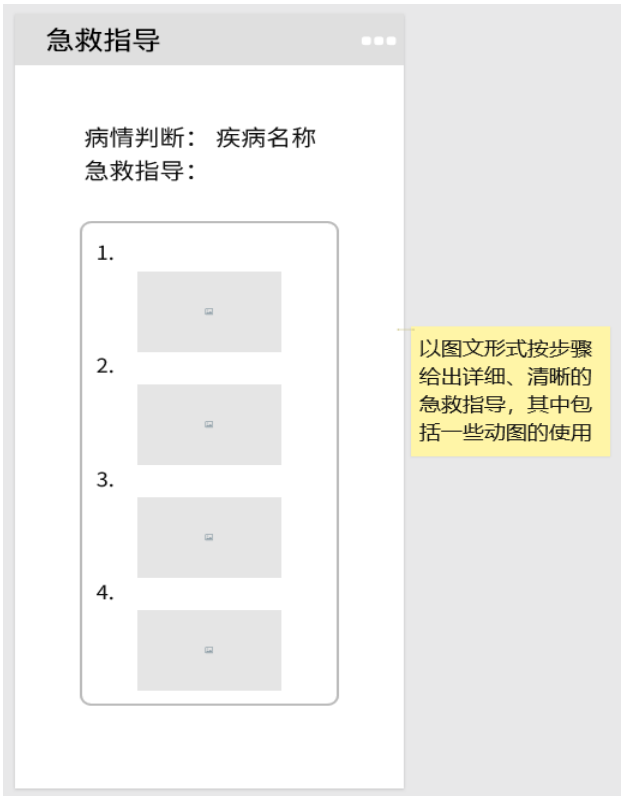


图 10：急救指导界面

2. 流行疾病检测功能

图 11：流行疾病自测功能

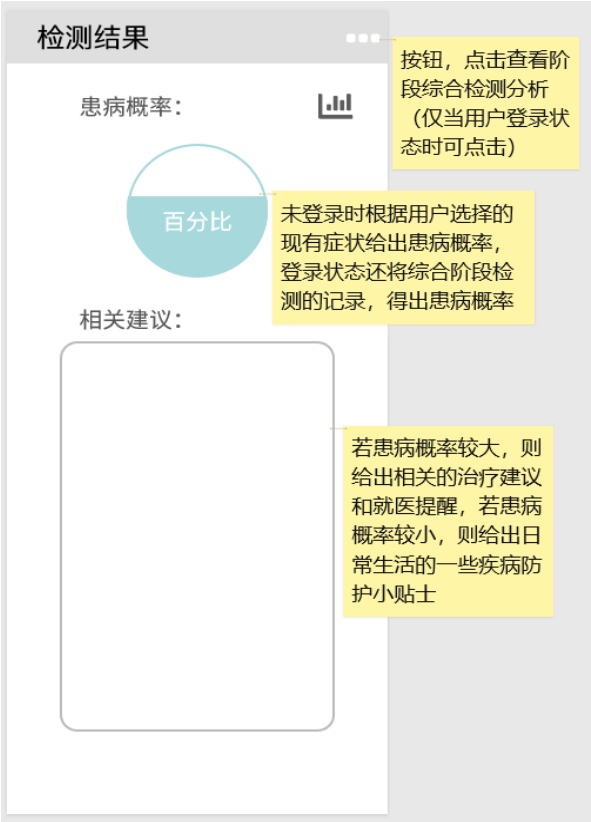


图 12：流行疾病检测结果

图 13：流行疾病阶段结果分析

非核心模块功能原型界面实例：

1. 注册功能：

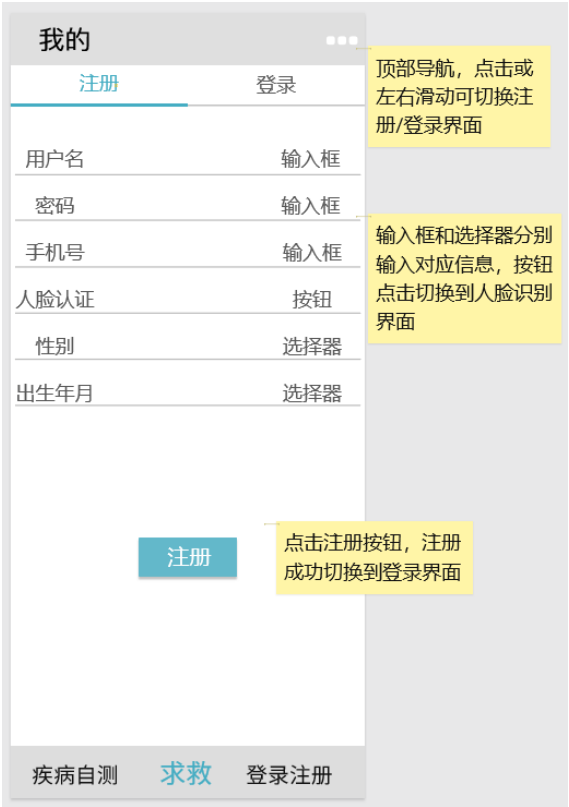


图 14：注册界面原型

2. 登录功能：

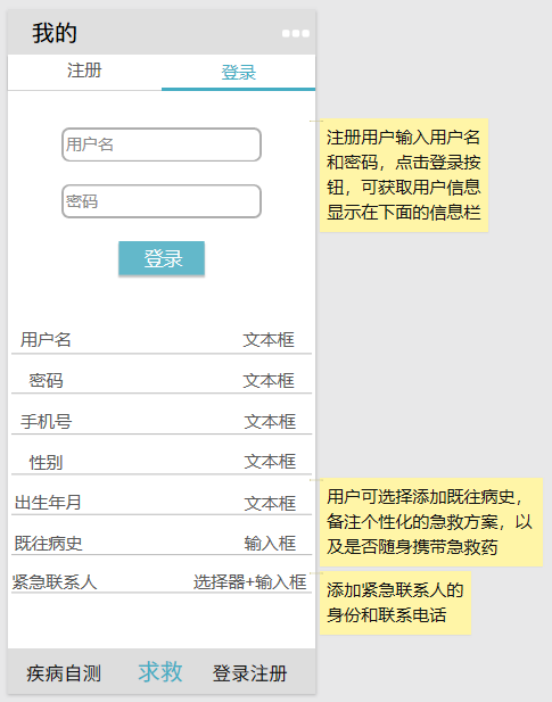


图 15：登录界面原型

4.5 接口需求

4.5.1 硬件接口

A) 系统摄像头(需大于 100w 像素)
B) 终端设备需可实现地理定位接口
C) 终端设备麦克风

4.5.2 软件接口

A) 语音识别服务：百度语音识别 ASR REST API
通过 API 调用，配置采样率等参数，调用百度 AI 语音转文字 API
B) 人脸识别服务：百度人脸识别 FACE API
通过 API 调用，配置人脸库设置像素范围等参数，调用百度 AI 人脸识别和录入 API

4.6 其他需求

（1） 可使用性:

该系统应使用一定数量的图形元素，直观，便于用户接受。同时要配上简洁易懂的文字，让使用者更好地理解功能，迅速熟悉系统的使用。

X-theme 团队注重项目的推广和使用体验及对项目的实用性分析。为此，团队对项目

的测试和推广做了大量的努力，通过使用者对项目的反馈和建议，来不断地优化项目的逻辑和流程，完善软件的功能，并不断提高用户的体验。

FF-AID 的可使用性是建立在准确度和逻辑的有效性上，为了提供给用户一个快速智能准确的急救方案，我们向某三甲医院的急救科医生寻求了建议和对准确度的评估判断，经专家认可证明算法的准确度和逻辑可行度都达到了可以投入使用的水平。

（2） 可维护性：

要求系统具有较好的可维护性，方便对子系统功能进行修改。

（3） 安全性：

身份认证：用户和后台管理员以及医院管理人员登入的身份认证为账号和密码，不同角色拥有不同权限，访问不同的数据。

权限控制：用户只能使用前端页面存在的那些功能，后台管理员拥有管理员的所有操作权限，比如对用户进行审核通过，注销用户，查看和修改求救消息记录等等，而医院管理人员拥有部分管理员的权限，比如对求救消息的查看，修改结束状态，查看用户的紧急联系电话和病史等等，并不能对用户进行管理，不能审核用户，注销用户等等。

（4） 可移植性：

系统采用模块化设计，可以实时替换，使用工具类方法，便于高效复用。应用可在大部分支持快应用的 Android 手机上运行，后端基于 jvm 适用于任意操作系统。

5 概要设计

5.1 处理流程

1. 逻辑流程图：

A） 一键求救功能

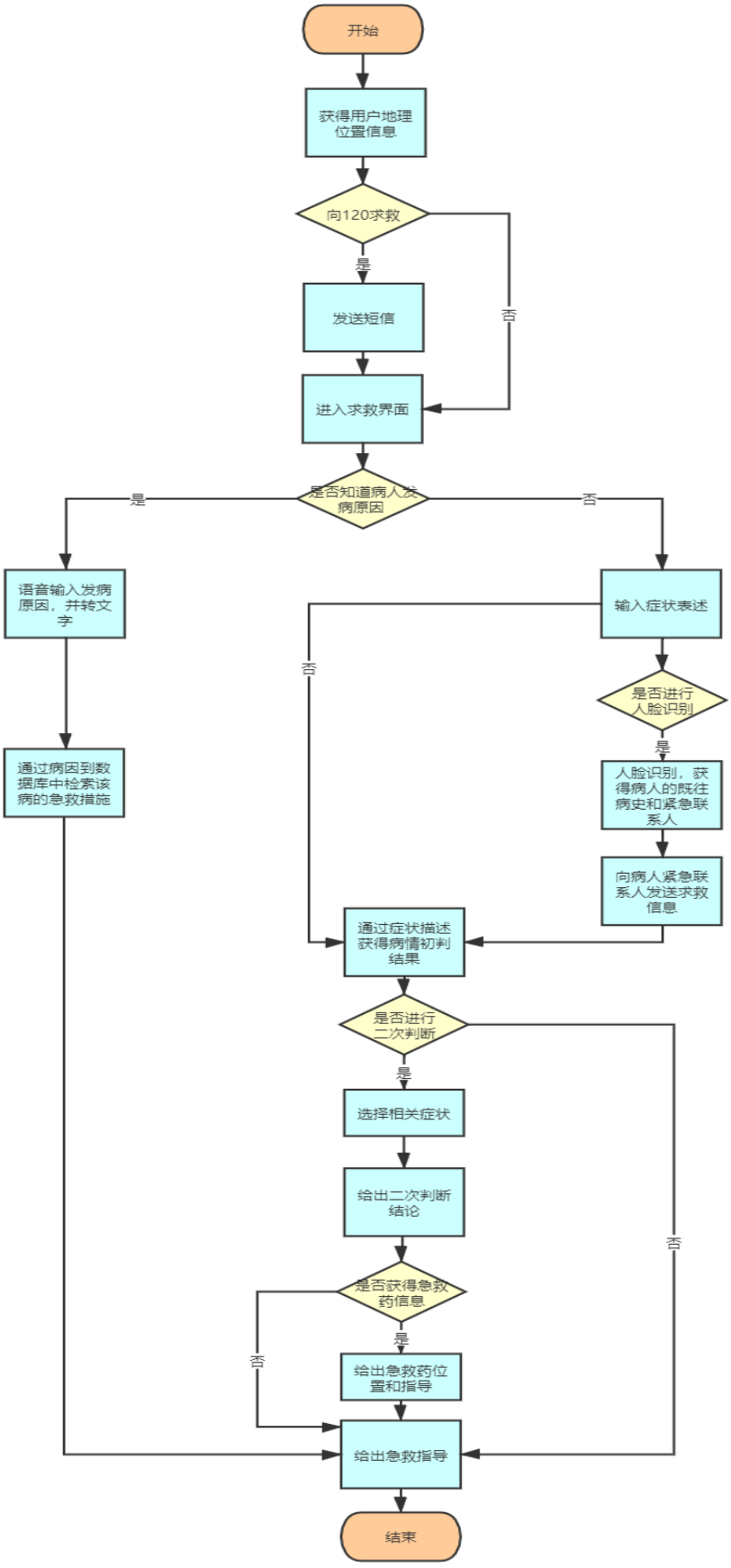


图 16：一键求救逻辑流程图

B）流行疾病智能检测功能

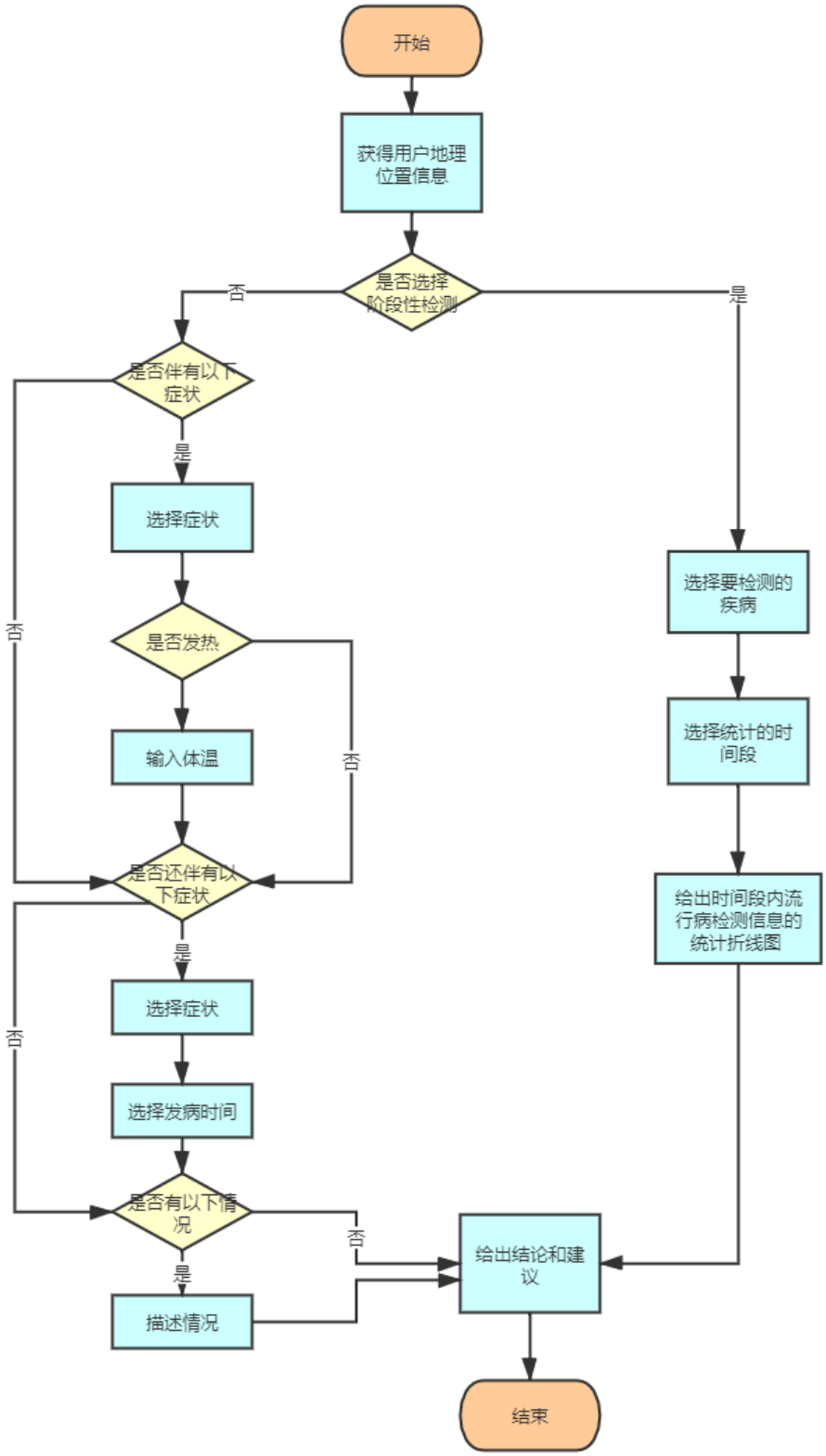


图 17：流行疾病智能检测逻辑流程

2. 数据流程图

A) L0 层数据流图

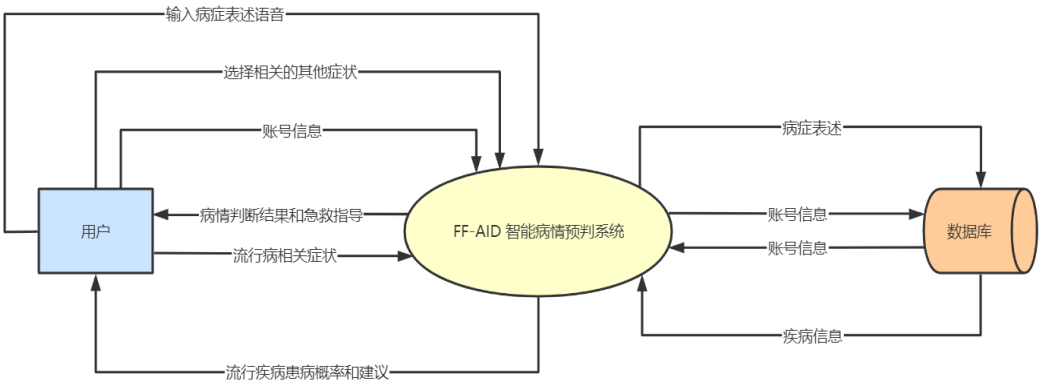


图 18：L0 层数据流图

B） L1 层数据流图

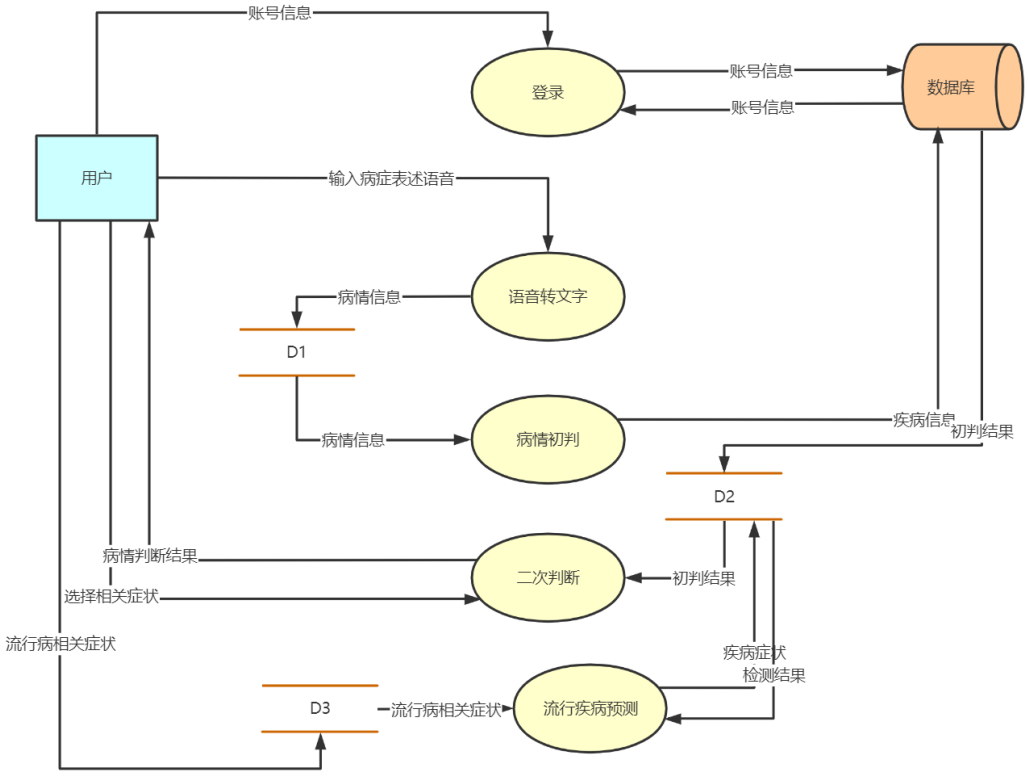


图 19：L1 层数据流图

5.2 总体结构设计

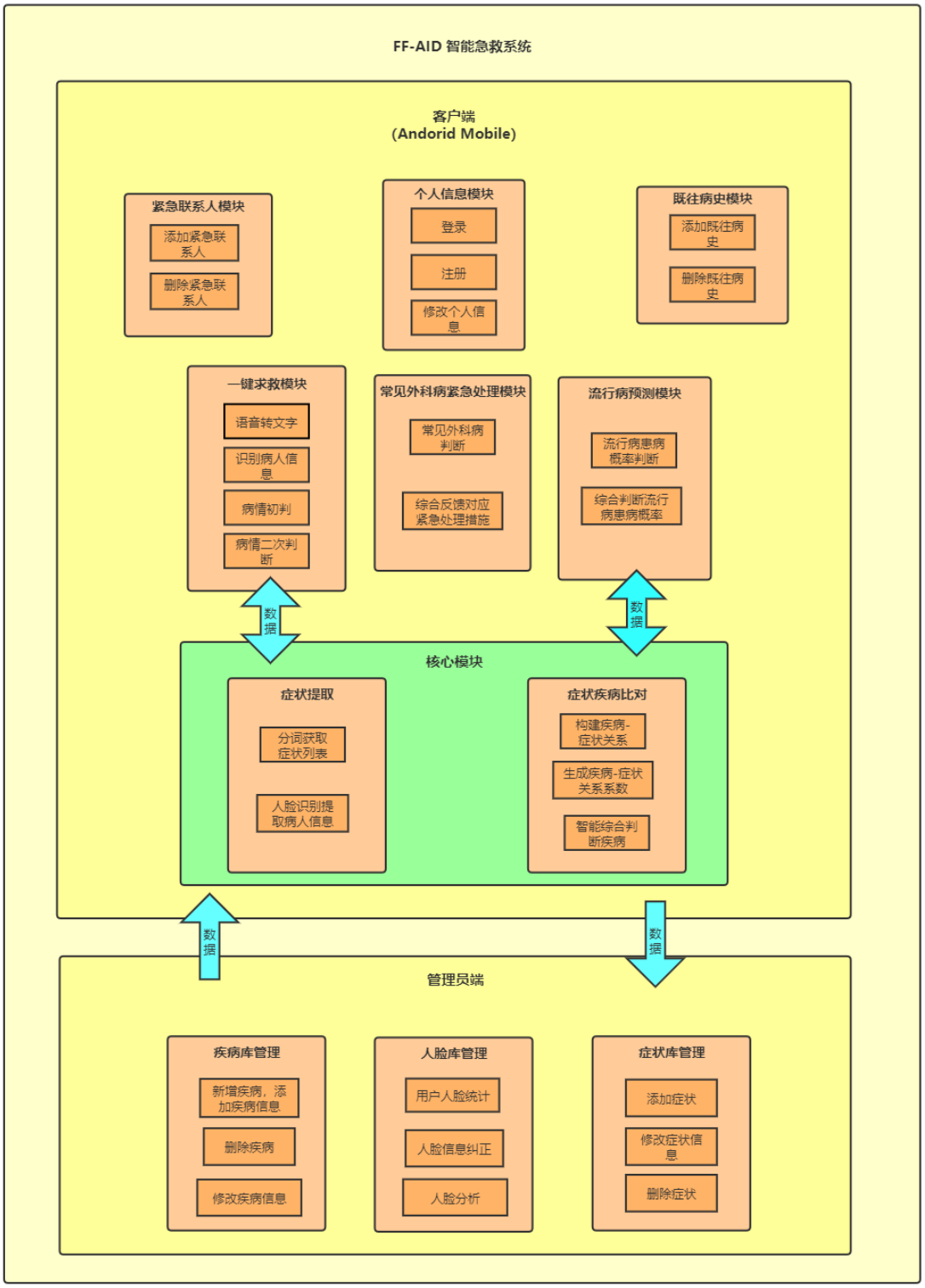


图 20：模块架构图

按客户端和管理员端进行模块划分：

1. 客户端：用于用户交互，数据传导，反馈

   A) 基础模块：主要包含用户简单的信息管理，如：用户个人信息模块，用户紧急联系

人管理模块，用户既往病史管理模块

B) 交互模块：包含主要功能实现，如：一键求救，流行疾病检测

C) 核心模块：包含核心功能实现的方法，如：症状提取，症状疾病比对

2. 管理员端：主要实现数据的更新迭代，业务过程实现

5.3 功能设计

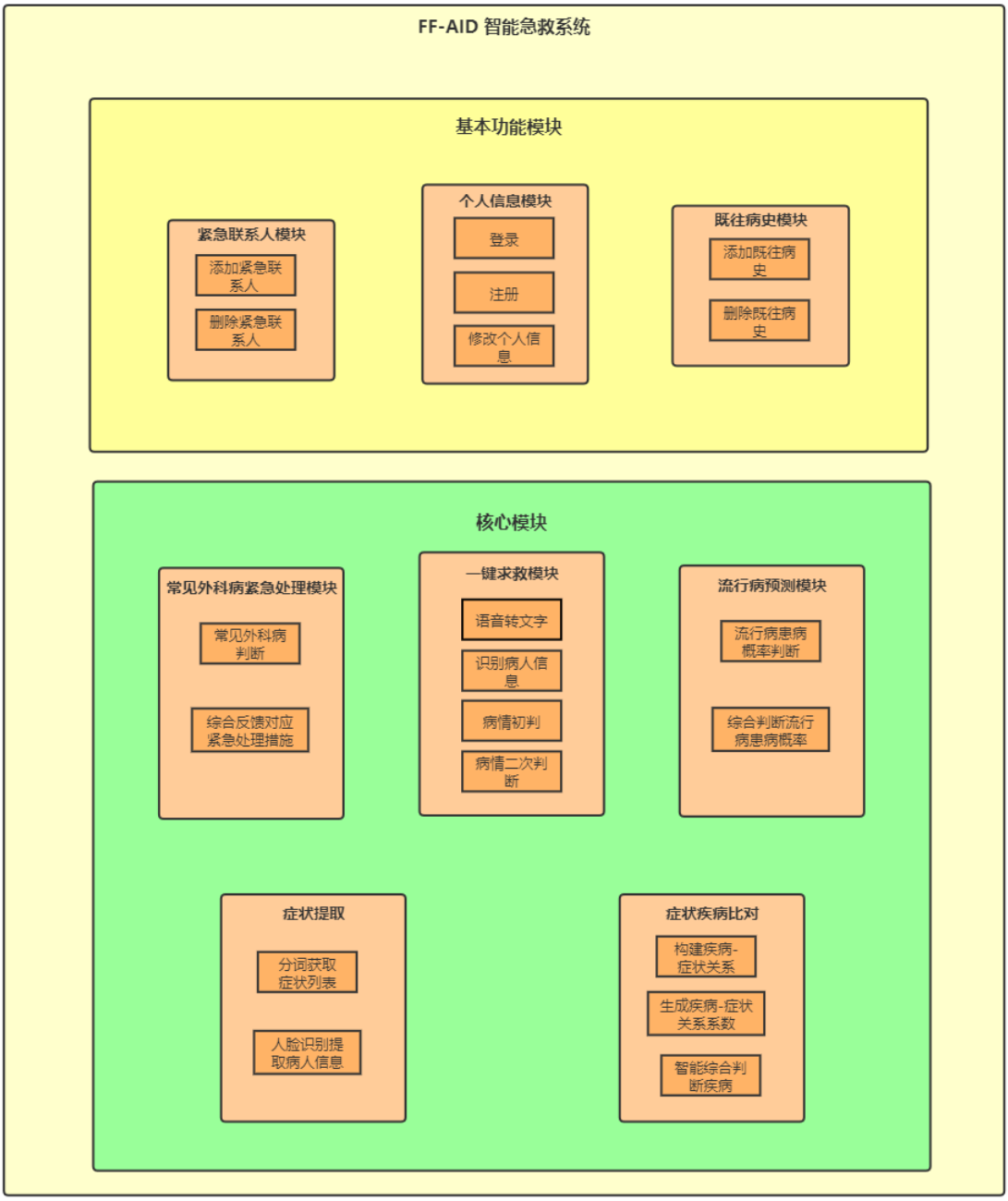


图 21：功能架构图

将功能分为基本功能模块和核心模块两部分：

1. 基本功能模块：

   A）个人信息管理：包括基本的注册，登录，修改个人信息等

   B）紧急联系人管理：包括新增紧急联系人，删除紧急联系人等

   C）既往病史管理：包括新增既往病史，删除既往病史等

2. 核心功能模块：

   A）一键求救模块：包括语音转文字，识别病人信息，病情初判，病情二次判断

   B）流行病预测模块：包括流行病患病概率判断，综合判断流行病患病概率

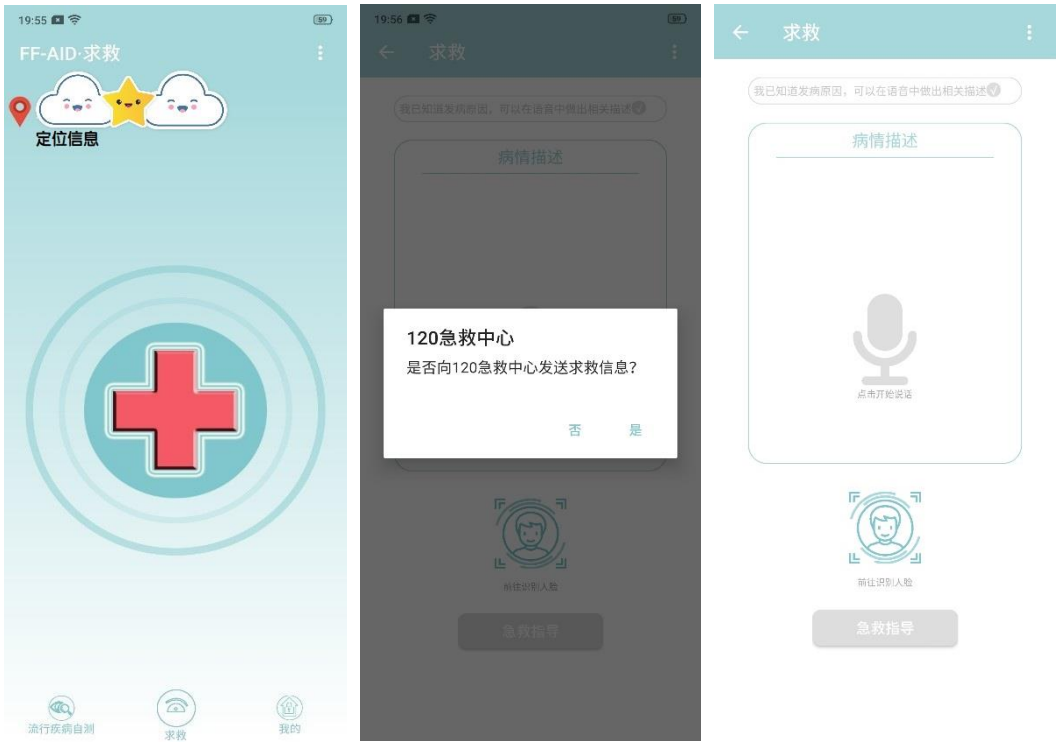
   C）症状提取：包括分词获取症状列表，人脸识别提取病人信息

   D）症状疾病比对：包括构建疾病-症状关系，生成疾病-症状关系系数，智能综合判断疾病

5.4 用户界面设计

注：界面功能表述可以参见 4.4 界面需求 及 使用手册

5.4.1 一键求救功能




求救
我已知道发病原因，可以在语音中做出相关描述
病情描述
我的朋友刚刚剧烈运动后突然晕倒，皮肤苍白，全身出汗，呼吸很困难的样子。
前往识别人脸
急救指导


病情预判
病情预判
急性心肌梗死
心力衰竭
心源性休克
是否出现以下症状
咳嗽
四肢冰凉
胸闷
意识模糊
气短
口吐白沫
确定

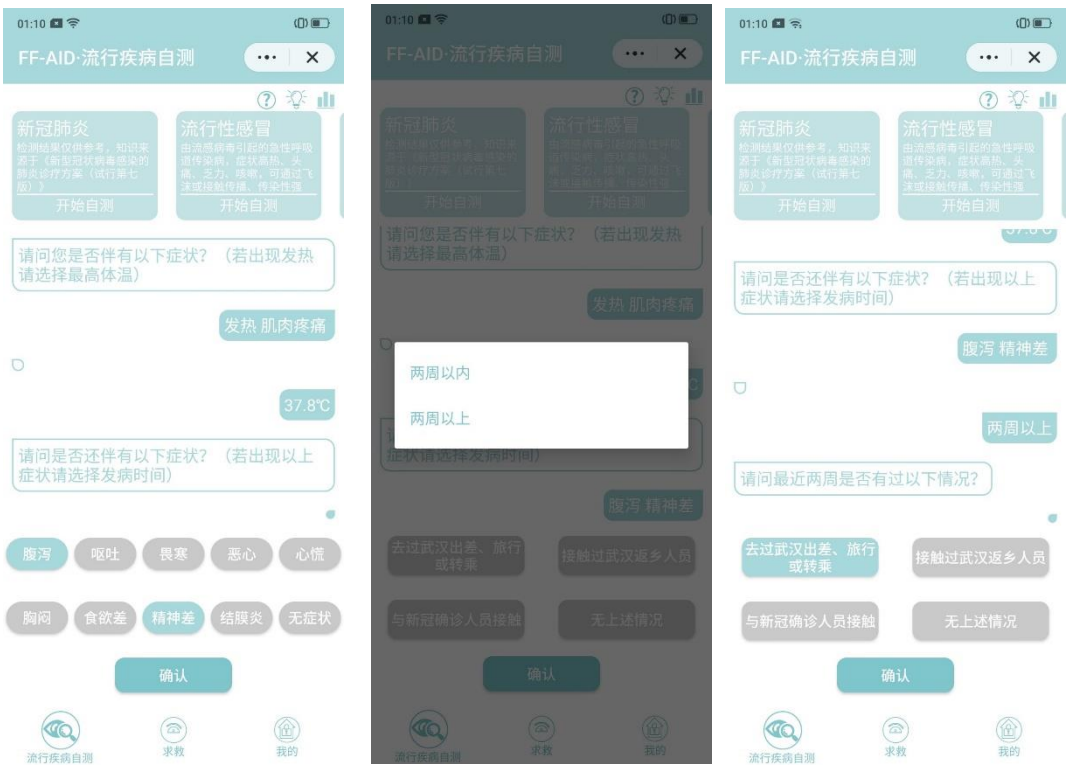

病情解析
急性心肌梗死
CPR(心肺复苏术)
急救药：
硝酸甘油，含服1片，上衣口袋
既往急救措施：
胸外按压30次，重复
急救指导 症状 并发症 病因 预防 保健
1. 持续呼叫患者的名字
让患者仰卧在平实的硬质平面上，头部与躯干处在同一平面；
2. 指压人中
用手指压人中，压到患者眉头皱起为止
(人中：位于鼻柱下，人中沟的上三分之一与下三分之二的交界处)；
人中
3. 胸外按压


黄金4分钟
出手急救
把握黄金4分钟
CPR（心肺复苏术）、AED（自动体外除颤器）
CPR第一步：识别和启动急救
判断患者意识
双手轻拍患者肩膀，在三侧呼唤患者，看是否有反应。
求助
寻求周围人帮助，并让人帮忙找附近的AED（自动体外除颤器）
判断患者心跳
若患者无呼吸或呼吸不正常（如喘息），同时用2-3根手指按压患者的颈动脉，若没有脉动，说明心脏骤停，需要马上开始心肺复苏。
CPR第二步：胸外按压30次
用力快速胸外按压30次


病情解析
急性心肌梗死
CPR(心肺复苏术)
急救药：
硝酸甘油，含服1片，上衣口袋
既往急救措施：
胸外按压30次，重复
急救指导 症状 并发症 病因 预防 保健
早期症状：
心前区疼痛往往是急性心肌梗死最早出现的症状，多见于清晨，休息或活动时都可能发生，可表现为持续长时间不缓解。
也有少数患者无疼痛，发病开始就出现胸闷、气短、呼吸困难、咳嗽等心衰症状，或意识模糊、皮肤苍白、四肢冰凉、全身出汗、血压降低等休克症状。
典型症状：
本病典型症状表现为心前区疼痛或憋闷感。疼痛或憋闷部位主要指胸骨后方，向左下方可延伸到左侧肋骨、上腹部，向上可到左侧肩、背甚至口腔、头部。部分患者也可表现在左上肢。
疼痛性质多为心前区的"压迫感、挤压感、沉重感"。部分患者胸部无明显感觉，仅胸部以外其他部位有钝痛或不适感。

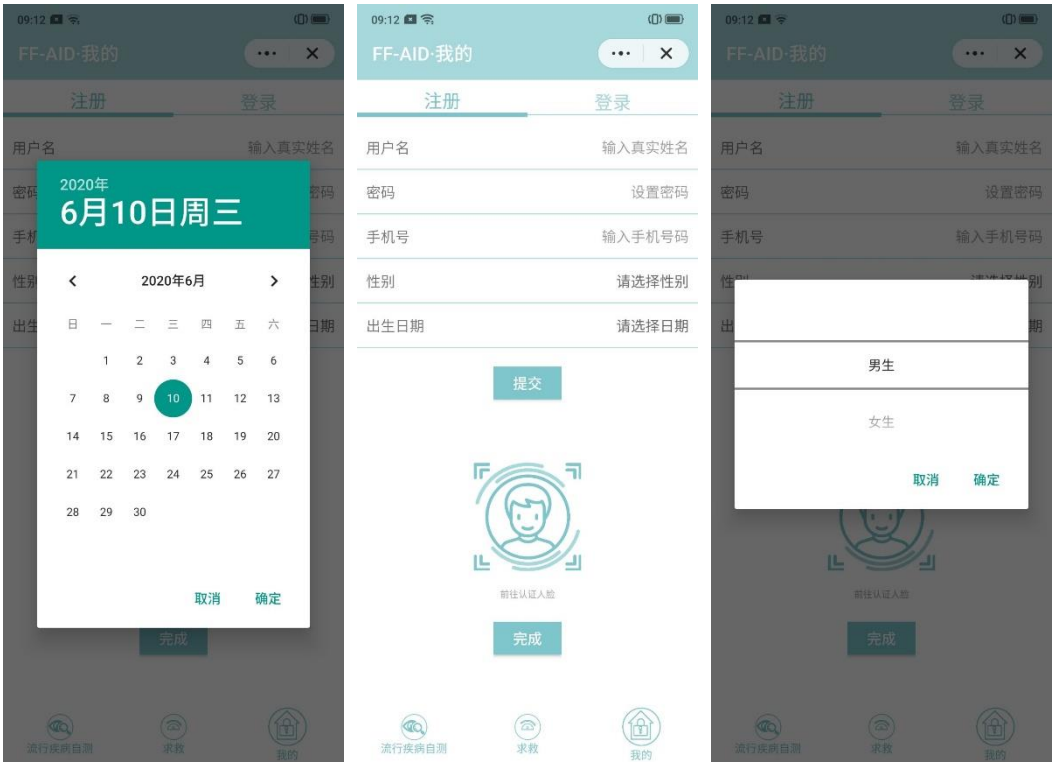

病情解析
急性心肌梗死
CPR(心肺复苏术)
急救药：
硝酸甘油，含服1片，上衣口袋
既往急救措施：
胸外按压30次，重复
急救指导 症状 并发症 病因 预防 保健
心律失常：
由于急性心肌梗死的发生，梗死部位所支配的心肌不能正常工作，心脏正常收缩节律改变，从而出现心律失常。严重的心律失常引起全身供血功能异常，从而危及生命。
心力衰竭：
梗死部位以外的心肌因梗死后负荷过大，大面积功能丧失，不能有效泵血，全身血液不能有效循环，危及生可。
休克：
由于心脏功能异常，全身循环障碍，血液不能有效传达至全身，多器官缺血缺氧，为急性心肌梗死引起死亡的必然阶段。一旦发生，即使医护人员全力抢救，仍有极大可能出现死亡。

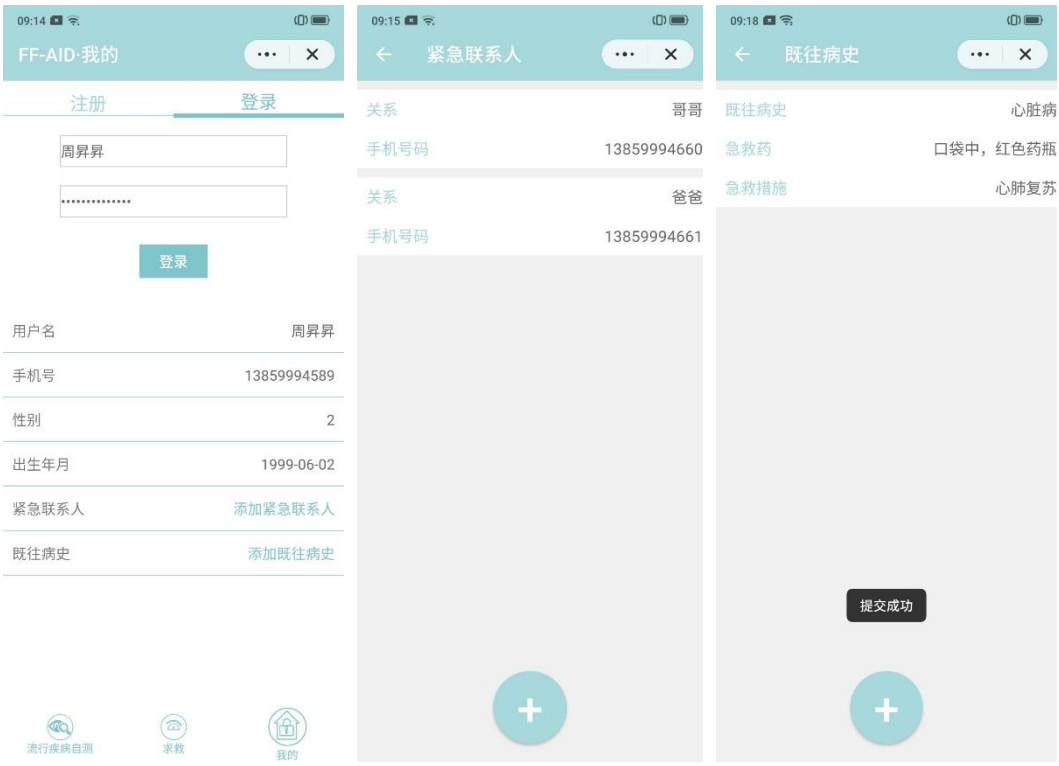
5.4.2 流行疾病检测功能



5.4.3 个人中心界面

注：用户名为化名





5.5 数据结构设计

1. 主要表结构：

中文表名	英文表名
用户信息表	user
既往病史表	illData
紧急联系人表	urgentTel
急救信息表	aidData
流行病检测信息表	epidemicData

表 16：数据表结构设计

2. 数据结构模型

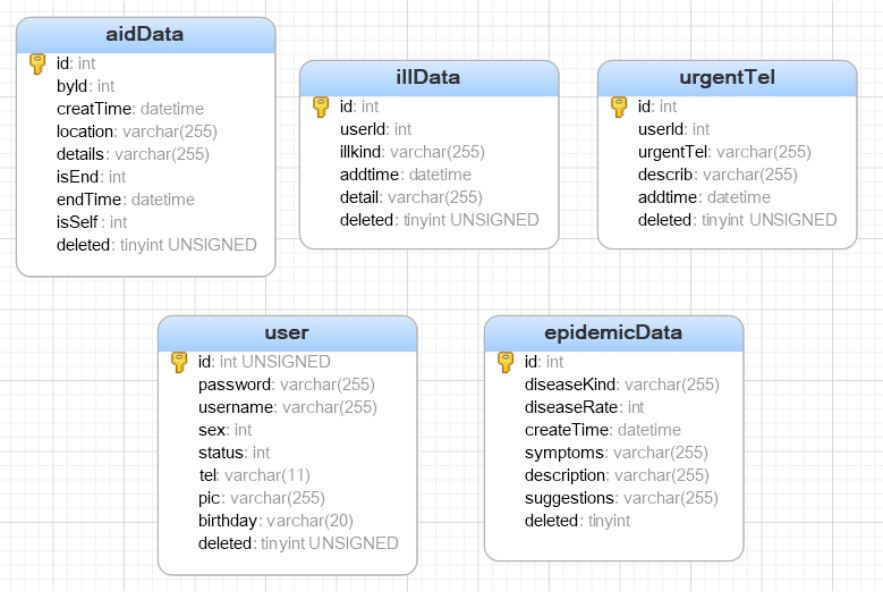


图 22：数据结构模型

3. ER 图：详见数据库设计模块（第 6 章）

5.6 接口设计

5.6.1 外部接口

1. 百度语音识别服务 ASR REST API
2. 百度人脸识别服务 FACE API

5.6.2 内部接口

系统内部 API 均遵循 Restful 原则：

1. 核心功能 API

   A）【POST】 /users/identify 人脸识别，返回对应用户的既往病史和紧急联系人信息
   B）【POST】 /users/voice 语音转文字接口，通过用户输入的语音返回对应的文字内容
   C）【GET】 /users/vWord 通过症状文本信息获得病情预判结果
   D）【POST】/users/voice/doc 直接通过语音获得症状信息对应的病情预判结果
   E）【GET】 /users/epidemic 通过用户选择的流行病症状判断患该流行病的概率
   F）【GET】 /users/advice 通过用户患该流行病的概率获得当前建议

2. 用户信息管理 API

   A）【POST】 /users 用户注册接口
   B）【GET】 /users/login 用户登录接口
   C）【DELETE】 /users/{id} 删除用户信息接口

D）【GET】 /users/{id} 获得单个用户信息接口

E）【PUT】 /users/{id} 修改用户信息接口

3. 紧急联系人管理 API

A）【POST】 /urgentTel 用户新增一个自己的紧急联系人信息

B）【GET】 /users/{id}urgentTel 查找一个用户对应的所有紧急联系人

C）【PUT】 /users/{id} 修改一条紧急联系人信息

D）【GET】/urgentTel/{id} 获得一条紧急联系人的详细信息

E）【DELETE】 /urgentTel/{id} 删除一条紧急联系人信息

4. 既往病史管理 API

A）【POST】 /illData 用户新增一个自己的既往病史记录

B）【GET】 /users/{id}/illData 获得一个用户的所有既往病史记录

C）【GET】 /illData/{id} 获得单挑既往病史记录

D）【PUT】 /illData/{id} 修改既往病史信息

E）【DELETE】/illData/{id} 删除相应既往病史记录

5. 急救记录管理 API

A）【POST】 /aidData 新增一条急救记录信息

B）【POST】 /aidData/{id} 结束一个急救过程

C）【GET】 /aidData/{id} 获得单条急救记录

详情可查看本项目 Swagger 文档 http://121.199.2.219:8080/swagger-ui.html#/

5.7 错误/异常处理设计

5.7.1 错误/异常输出信息

1. 自定义异常输出工具类，用以定义和记录异常信息

```java
public class GetException extends Exception{

    /**
     * 未定义错误
     */
    public static final Integer ERROR =400;
    /**
     * 格式错误
     */
    public static final Integer FORMAT_ERROR=401;
    /**
     * 资源获取失败错误
     */
    public static final Integer GET_RESOURCE_ERROR=403;
    /**
     * 资源未找到错误
     */
    public static final Integer NOT_FOUND_ERROR=404;

    private int state;

    public GetException(String info,int state)
    {
        super(info);
        this.state=state;
    }
```

图 23：异常捕获 GetException 代码

编号	错误信息	说明	对应异常处理类
1	400	出现未定义错误	GetException
2	401	格式错误	GetException
3	402	资源获取失败	GetException
4	404	资源未找到	GetException

……

2. 使用 throw Exception 对异常信息自底向上层次上抛，直到 controller 层，然后处理全局异常，错误码作为 response 的状态码返回
3. 使用 slf4j 做日志记录

5.7.2 错误/异常处理对策

1. 在本系统中，错误以异常的方式捕捉，异常处理方式分为两层：

   A） GetException 抛出的异常部分在控制器内部捕捉，以页面提示的形式返回给用户，并且在日志中记录；控制器捕获处理不了的异常继续往顶层抛出

   B） 其他异常由顶层捕获处理，顶层统一在日志记录异常信息

2. 事务数据回滚和系统重启：

   当系统工作不正常时，首先检查日志，解决日志中记录的错误，之后系统重启，同时回滚数据，防止异常导致的脏数据

3. 版本备份：

   使用 Git 多版本，服务器备份等软件后备。重启失效后可以使用备份的版本分别恢复系统和数据库，再次重启

4. 环境监控和性能降级：

   实时监控软件的流量及相应度，在达到阈值后启用排队队列限流，或者降低系统的性能

5.8 系统配置策略

部署策略： 本项目后端采用 spring 框架，将 spring 项目打包为 jar 文件，通过远程传输到 Linux 云服务器，在服务器中后台运行。

注： 部署过程中使用自动化部署方案，减少每次部署调试花费时间，加快开发进度。

5.9 系统部署方案

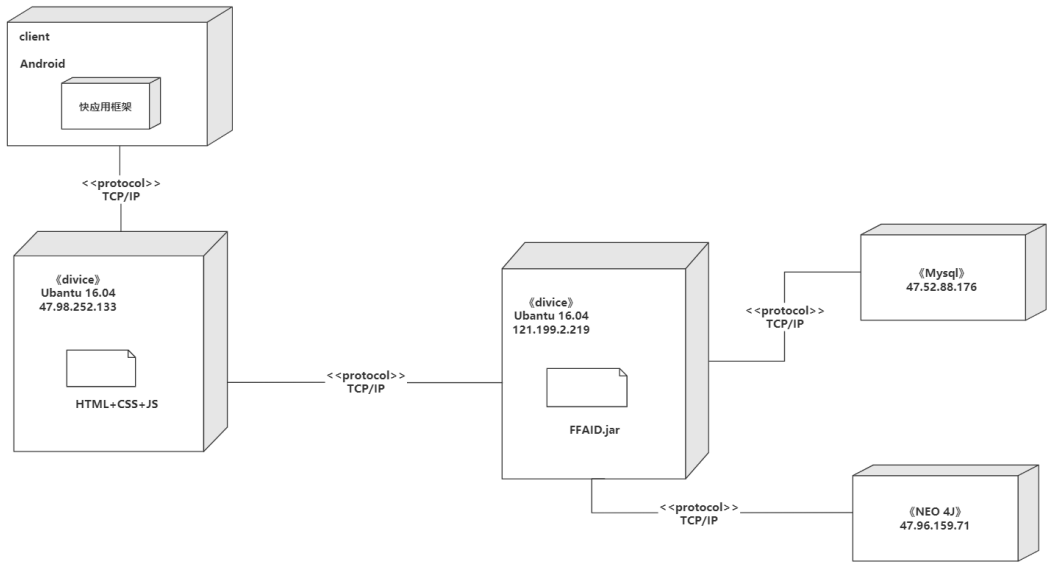


图 24：系统部署图

本系统采用多服务器部署，将前端，后端，数据库分开部署到四台服务器：

A） 将前端部署在一台基于 Ubantu 16.04 的阿里云服务器

B） 将主要的后端项目 jar 包部署在一台基于 Ubantu 16.04 的阿里云服务器，该服务器是所

有四台服务器中配置最高的，承载了主要的运算

C） 将关系型 Mysql 数据库配置到一台居于 Ubantu 16.04 的阿里云服务器

D） 将非关系型 Neo 4j 数据库配置到一台基于 Ubantu 16.04 的阿里云服务器，考虑到访问病理数据是本系统最重要核心的需求，该服务器性能也较高

5.10 其他相关技术与方案

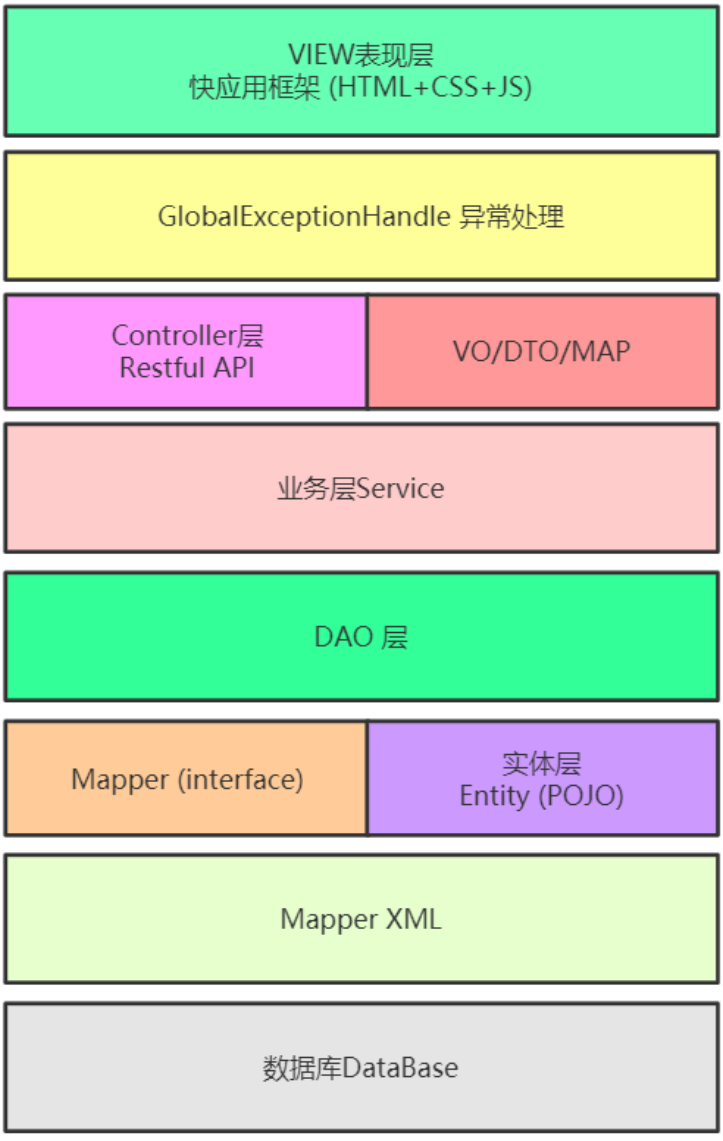


图 25：开发框架技术构成图

项目开发的整体框架如图所示：

A）以 Springboot 为基本框架

B）数据库映射采用主流 Mybatis

C）采用 MVC 分层设计，通过对自身基本部分分离赋予了各个基本部分应有的功能，使后续对程序的修改和扩展简化，并且使程序某一部分的重复利用成为可能。

D）前后端分离，使用 controller 层构成内部接口供前端调用

E）可实现微服务，若用户量足够大可以将一键求救的急救模块和流行疾病智能检测模块分开部署，实现微服务架构

6 数据库设计

1. ER 图(如图所示)

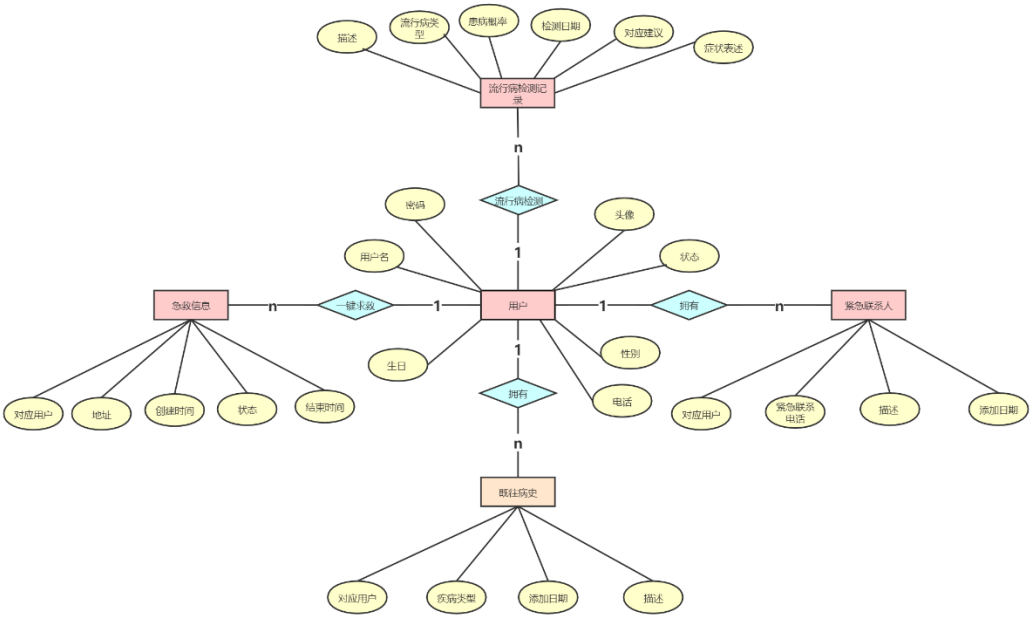


图 26：ER 图

2. 数据库模型（如图所示）

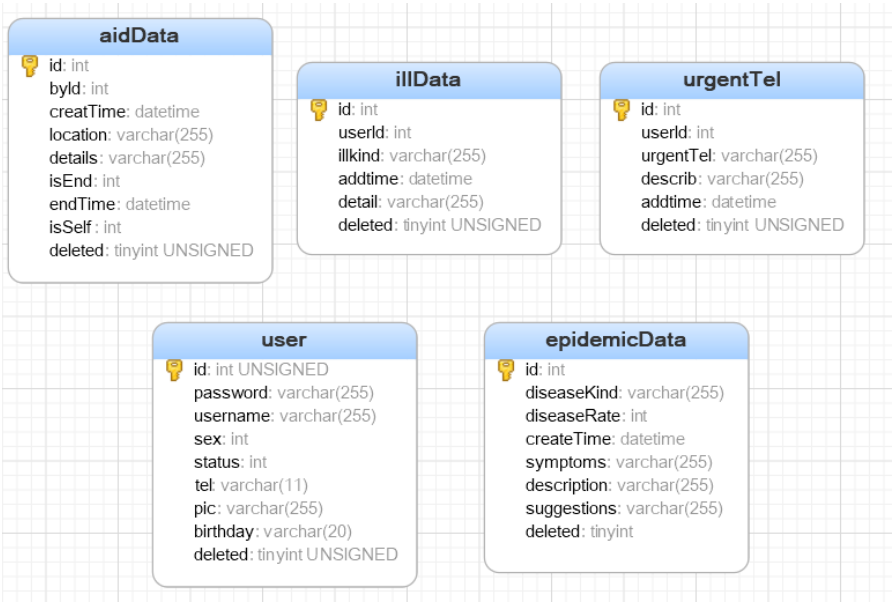


图 27：数据库模型图

3. 数据字典

参考需求分析 4.1.3

7 详细设计

7.1 一键求救功能模块

7.1.1 功能描述

主要分为四大功能和一个可选功能点：

1. 病情初判

   用户通过语音输入病人的症状表述，结合通过对患者的人脸识别获得的患者既往病史，通过智能病情预判系统获得当前病症对应的最有可能的疾病列表，并提供急救指导。

2. 二次判断病情

   在病情初判后，用户可以在系统给出的相关症状中选择患者可能有的其他症状，结合病情初判的结果，给出最精确的病情判断和急救指导。

3. 人脸识别获得患者急救信息

   若患者有输入自己的急救信息，包括急救药和急救措施，可以在人脸识别的同时获取，并且在急救指导的时候反馈给施救者，比如告诉施救患者的急救药在上衣左口袋，服用两颗，且第一时间放平躺等待救护车。

4. 已知疾病获得急救指导

   若已知患者的疾病，可以直接语音输入病因，获得对应的急救指导。

5. * 发送求救信息

   用户可以选择是否向 120 急救中心发送急救信息，以及可以向通过人脸识别获得的用户紧急联系人发送求救信息。

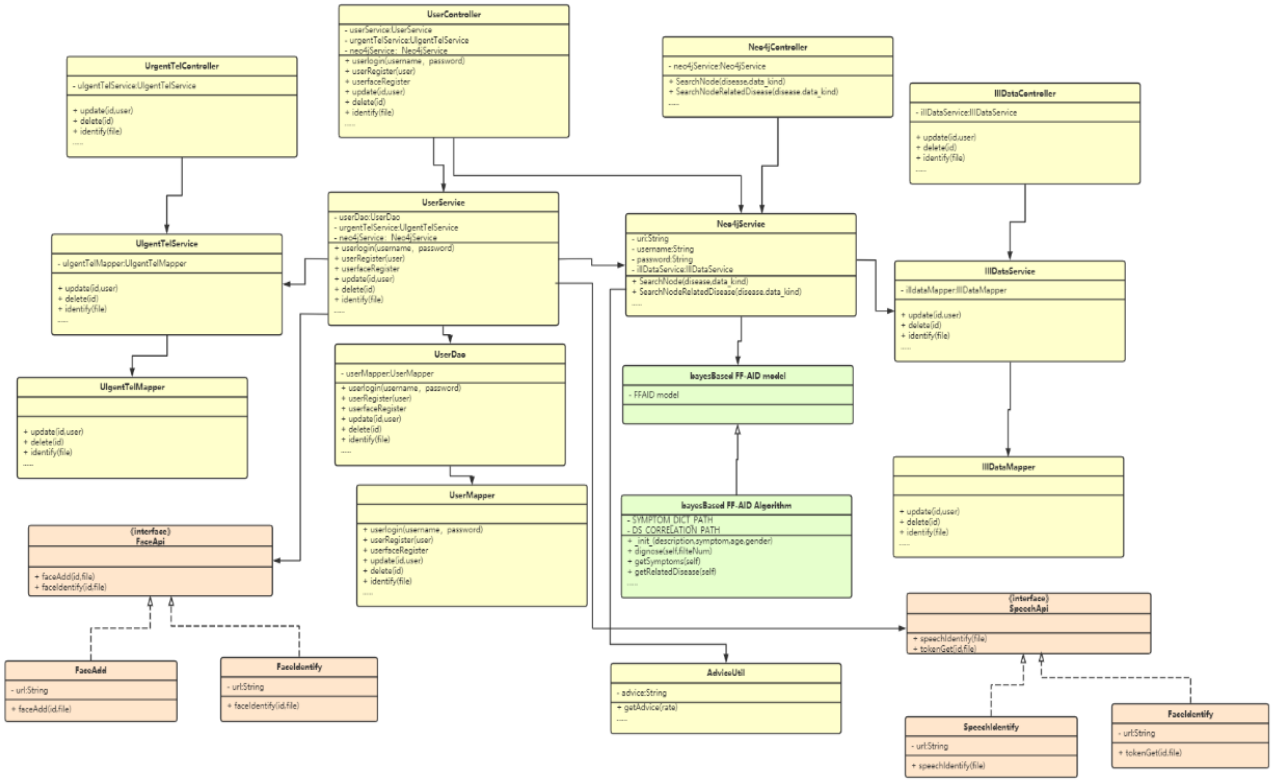


图 28：一键求救 UML 类图

7.1.2 性能描述

单次病情判断的响应时间控制在 2s 以内

7.1.3 输入

A） 对患者病症的语音表述
B） 患者的人脸图像信息

7.1.4 输出

A） 对应患者的疾病初判结果和急救指导
B） 对应患者的疾病二次判断和急救指导
C） 患者的紧急联系人（自动发送求救信息）

7.1.5 程序逻辑

详见功能流程图：

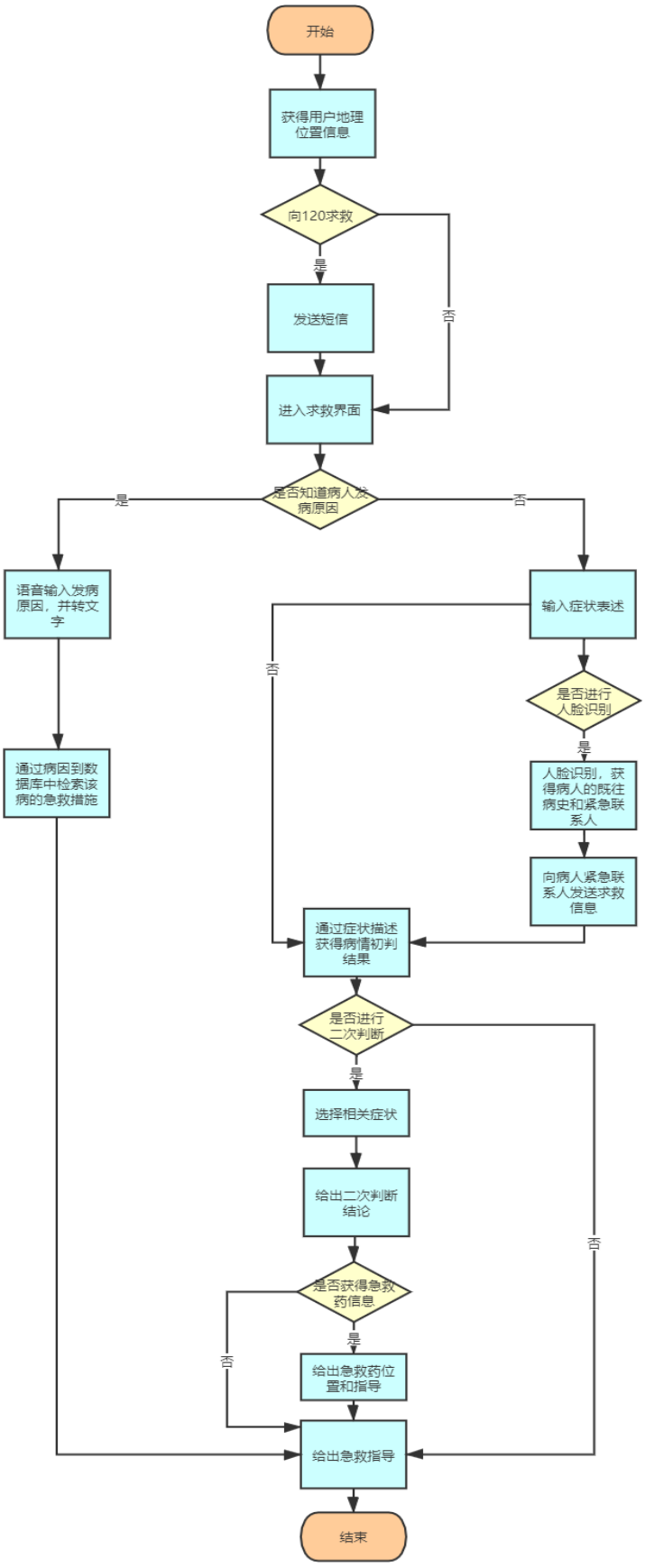


图 29：一键求救功能流程图

算法概述：

智能诊断算法属于机器学习领域的内容，旨在根据用户输入的症状等信息给出最有可能的疾病。本系统基于医疗知识图谱，构建症状-疾病关系网络，并借助基于聚簇的排序算法，构建智能诊断系统。

算法流程：

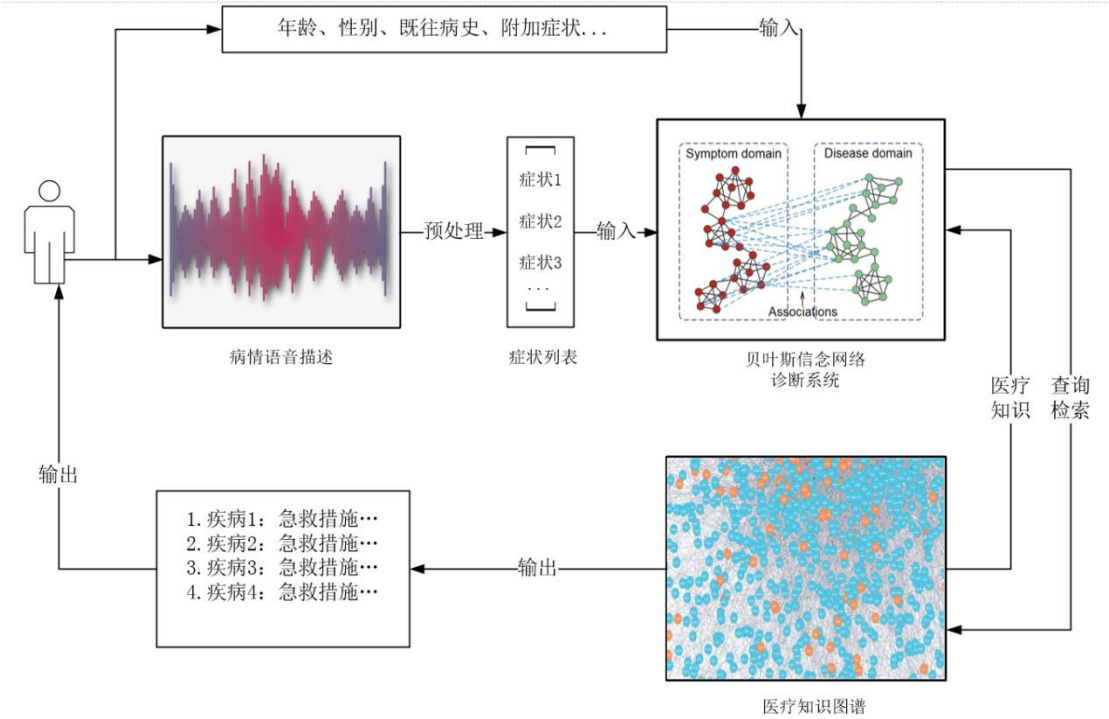


图 30：急救诊断算法流程

7.1.6 限制条件

1. 终端设备良好的性能
2. 网络环境良好

7.2 常见外科病紧急处理功能模块

7.2.1 功能描述

常见外科病紧急处理功能模块实现了对常见的需要得到第一时间有效处理的外科创伤比如烧伤，感染，刀伤，骨折等的疾病生成第一时间处理方案的功能，避免了由于处理不够及时而导致的深度感染，皮肤病变等问题。

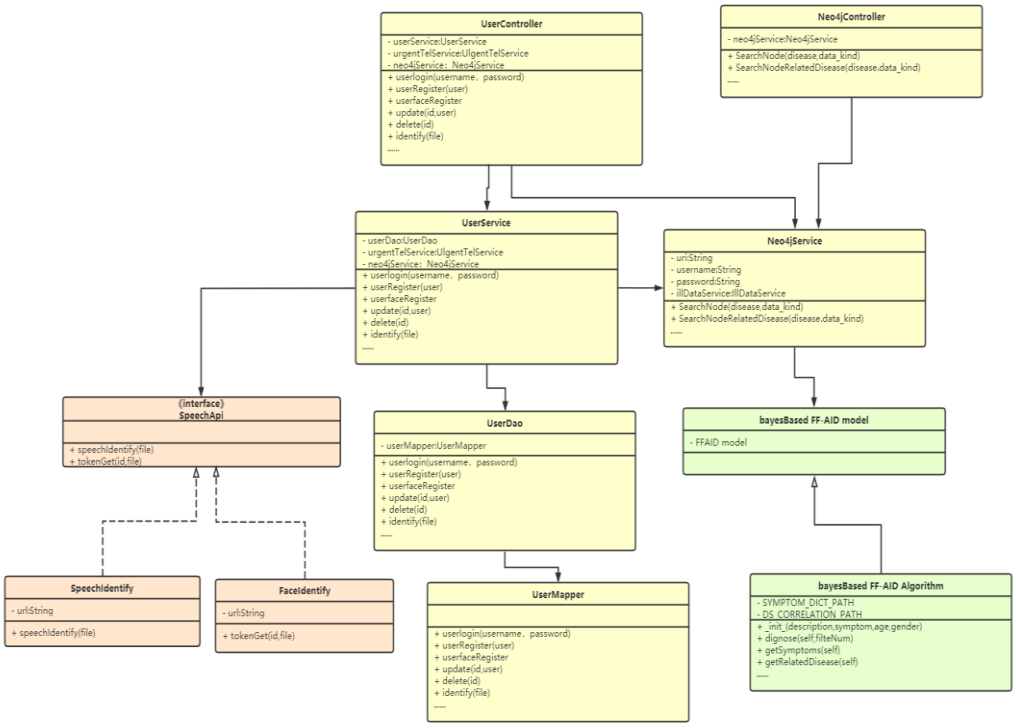


图 31：常见外科病紧急处理 UML 类图

7.2.2 性能描述

单次紧急处理方案生成的响应时间控制在 2s 以内

7.2.3 输入

用户语音输入或者文本输入病情。

7.2.4 输出

A) 对应的外科疾病的紧急处理方案
B) 附近可以及时处理伤口病情的医院的位置

7.2.5 程序逻辑

详见功能流程图：

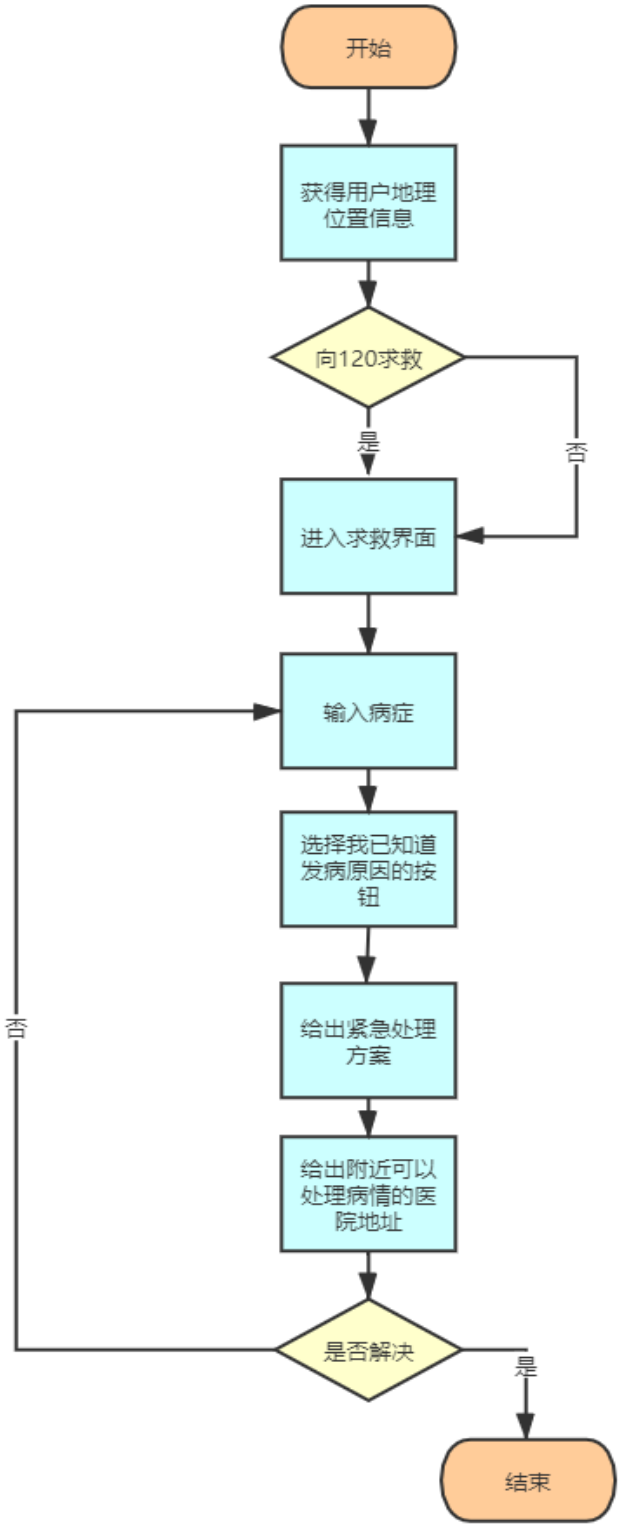


图 32：常见外科病紧急处理功能流程图

7.2.6 限制条件

1. 用户能够准确地提供外科伤病的名称
2. 终端设备良好的性能
3. 网络环境良好

7.3 流行疾病检测功能模块

7.3.1 功能描述

流行病患病概率检测

用户选择自己想要检测的当下流行病（如新冠），在智能助手的提示下选择病症出行史，体温等信息，系统智能判断并给出用户患该流行病的可能性，并给出当前建议

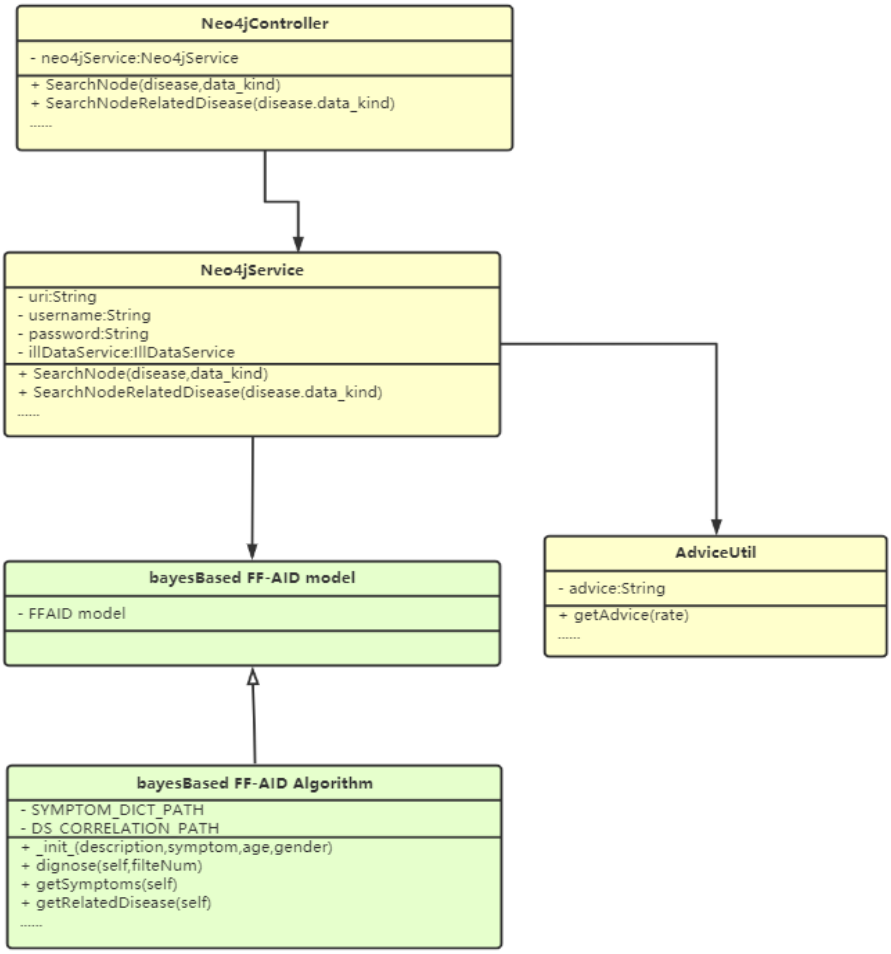


图 33：流行病检测功能 UML 类图

7.3.2 性能描述

单次流行病检测的响应时间控制在 2s 以内

7.3.3 输入

A）用户选择要检测的流行病
B）用户的体温
C）用户选择的对应症状
D）用户长时间的流行病症状数据和检测结论
E）用户的地理定位

7.3.4 输出

A) 用户患对应流行病的概率
B) 对用户当前症状和患病可能的智能指导

7.3.5 程序逻辑

详见功能流程图：

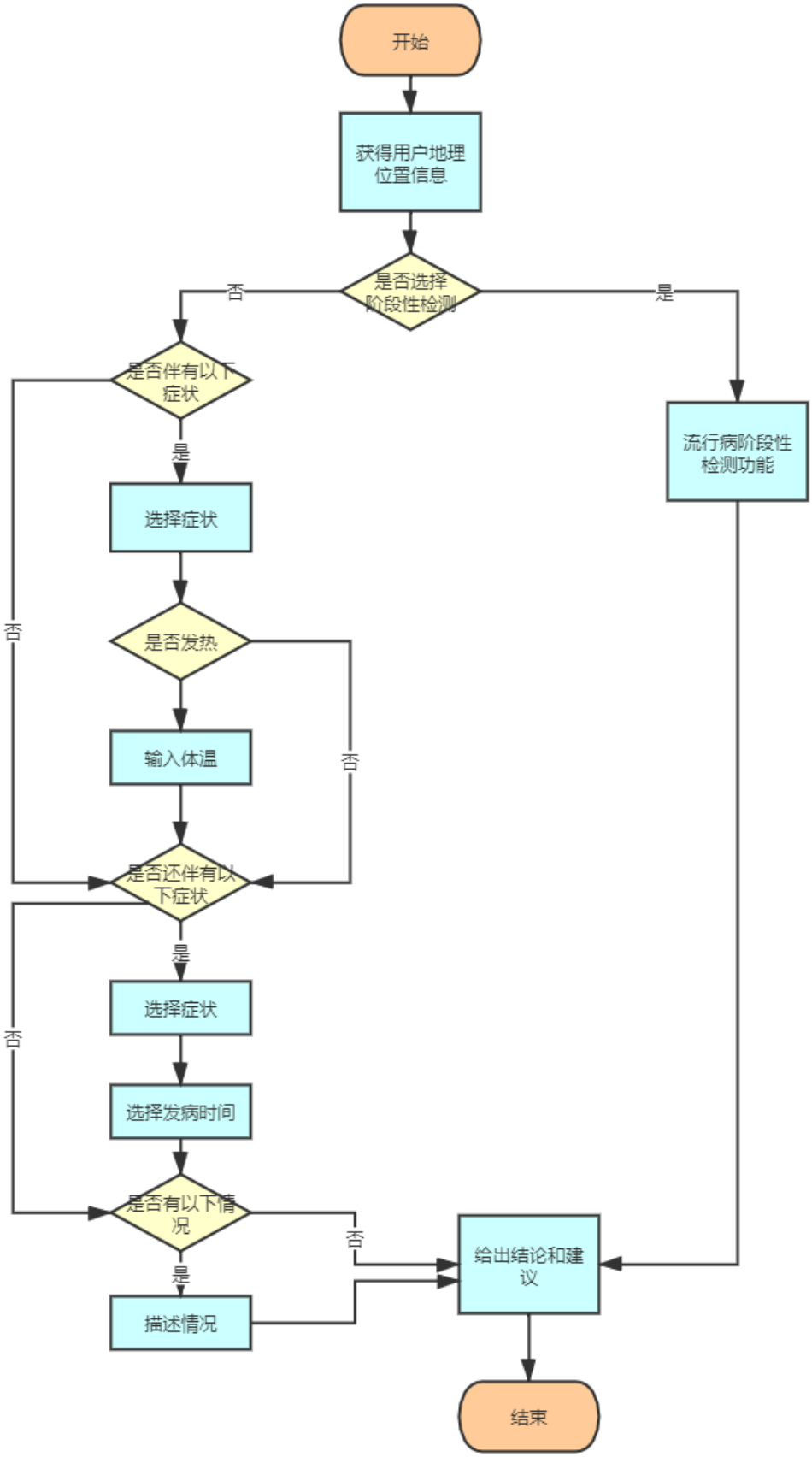


图 34：流行病检测功能流程图

算法概述：

用户首先输入当前居住地、最近旅行史等信息调整流行疾病发生的先验概率。接着使用 QMR-DT 模型计算用户感染流行病的概率，最后根据预定义阈值给出相应的预防措施。

算法流程：

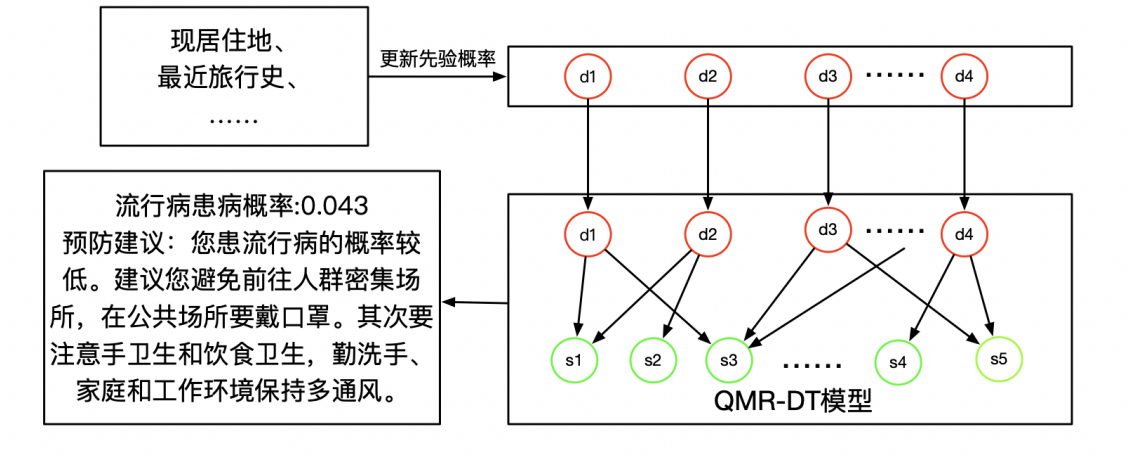


图 35：流行病检测算法简介

7.3.6 限制条件

1. 位置信息获取正常
2. 终端设备良好的性能
3. 网络环境良好

7.4 流行疾病阶段性分析功能模块

7.4.1 功能描述

基于长时间统计的综合判断流行病患病概率

用户通过一段时间使用该功能所输入的数据，系统结合长时间的表现和该流行病在不同发展阶段的症状，同时考虑用户所在地该流行病的发病情况，给出智能准确的结论和阶段性建议以及用户在该时间段内的流行病检测信息的折线统计图。

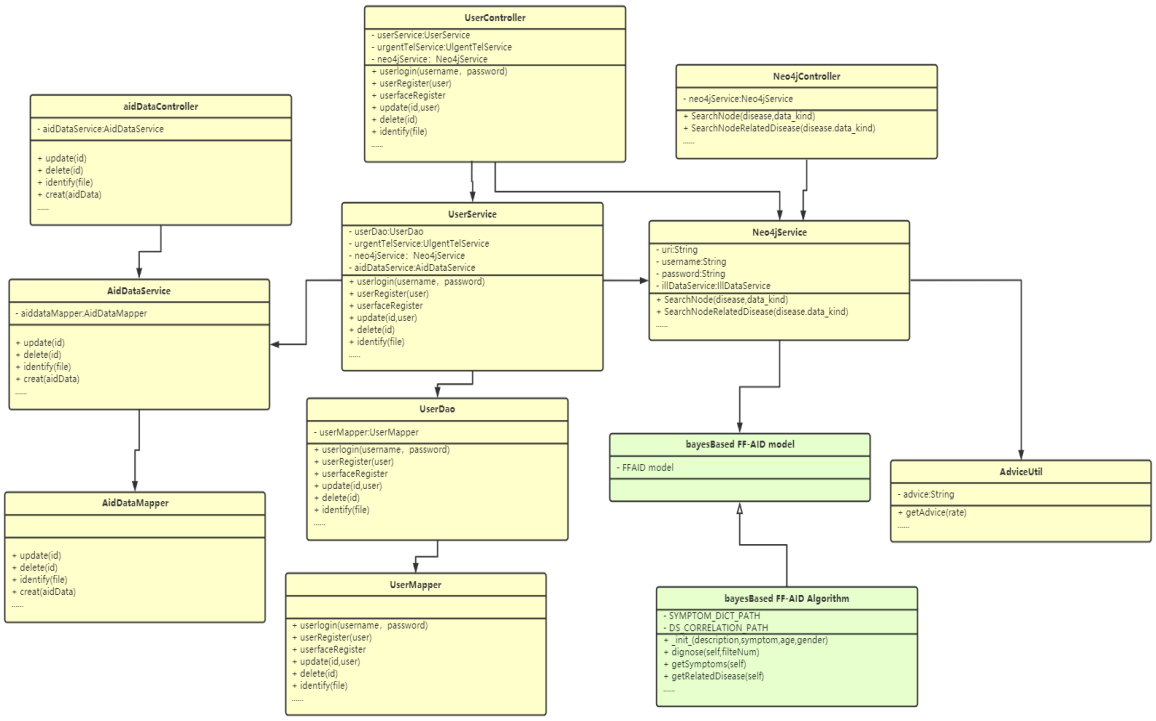


图 36：流行疾病阶段性分析功能 UML 类图

7.4.2 性能描述

单次流行病检测的响应时间控制在 2s 以内

7.4.3 输入

A) 用户选择检测的流行病疾病
B) 用户的地理定位
C) 用户选择检测的时间段
D) 用户长时间的流行病症状数据和检测结论

7.4.4 输出

A) 用户患对应流行病的概率
B) 对用户当前症状和患病可能的智能指导
C) 用户时间段内的流行病检测信息的统计折线图

7.4.5 程序逻辑

详见功能流程图：

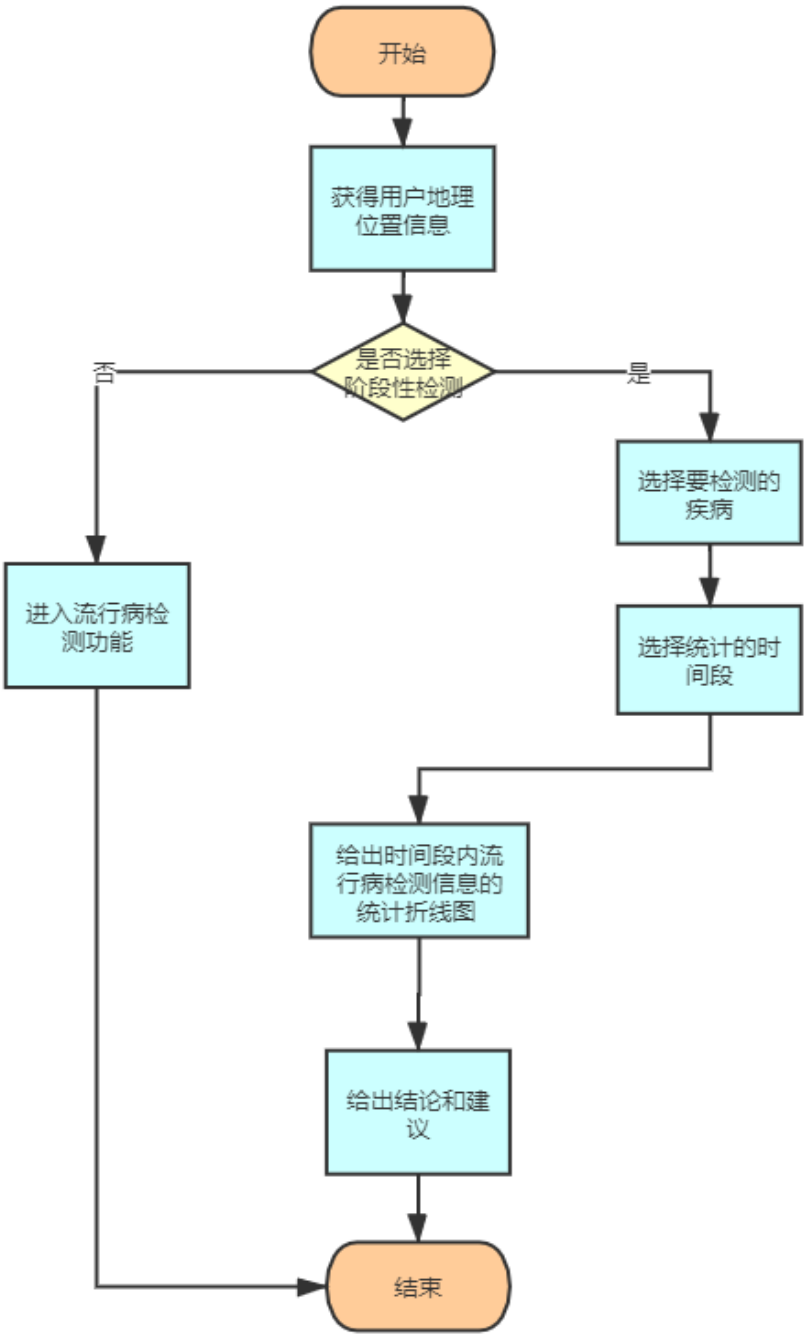


图 37：流行病阶段性分析功能流程图

7.4.6 限制条件

1. 终端设备良好的性能
2. 网络环境良好
3. 位置信息获取正常

7.5 人脸及急救信息处理功能模块(非核心)

7.5.1 功能描述

此功能模块涵盖了功能模块图中所有用户信息管理及人脸库管理的功能，由于非核心，故而综合在一块叙述。

主要功能包含：

(1) 用户输入个人信息注册
(2) 用户录入自己的人脸信息
(3) 用户录入紧急联系人信息
(4) 用户录入既往病史信息
(5) 用户录入急救药信息及紧急处理信息
(6) 用户管理个人信息和人脸库信息
(7) 用户管理自己诊断结果信息

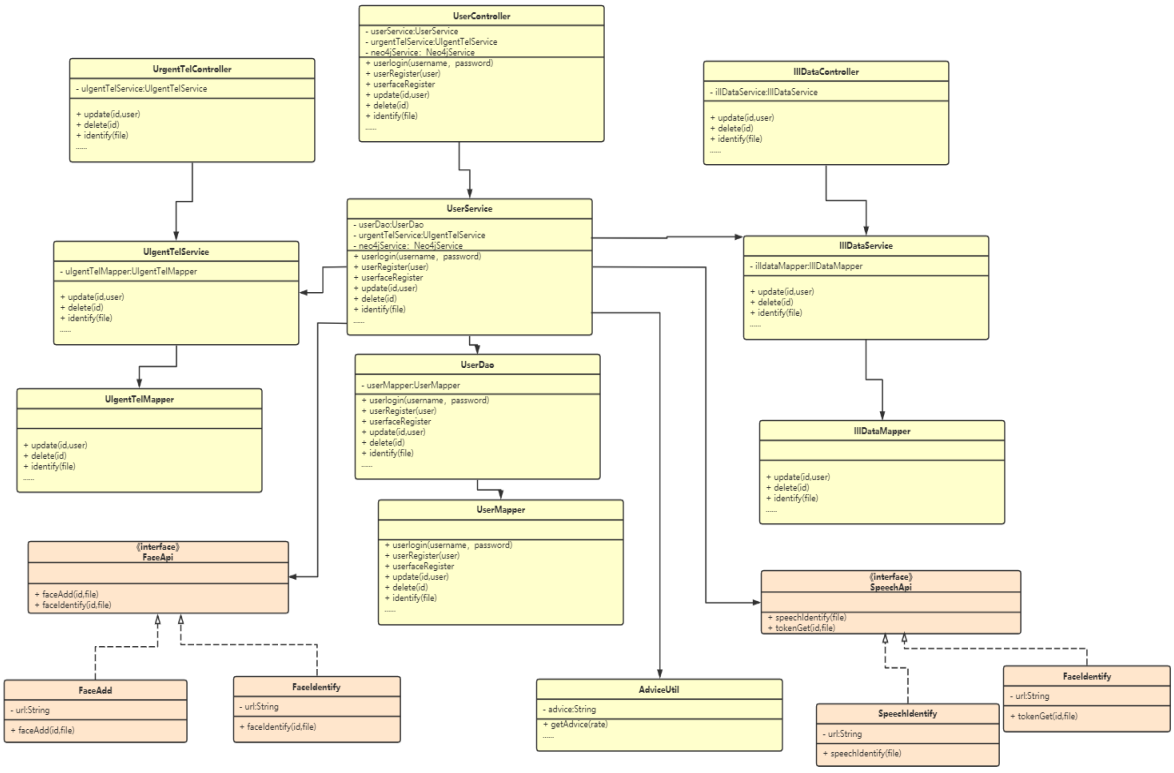


图 38：流行疾病阶段性分析功能 UML 类图

7.5.2 性能描述

单次信息录入或者修改的相应时间应该在 1s 以内，应支持至少每秒 10 次的并发量

7.5.3 输入

A) 用户的个人信息
B) 用户的人脸信息
C) 用户紧急联系人信息
D) 用户既往病史信息
E）用户急救药信息
F）用户急救紧急处理措施

7.5.4 输出

A) 用户信息修改后的返回信息
B）用户人脸库录入成功提示

7.5.5 程序逻辑

详见功能流程图：

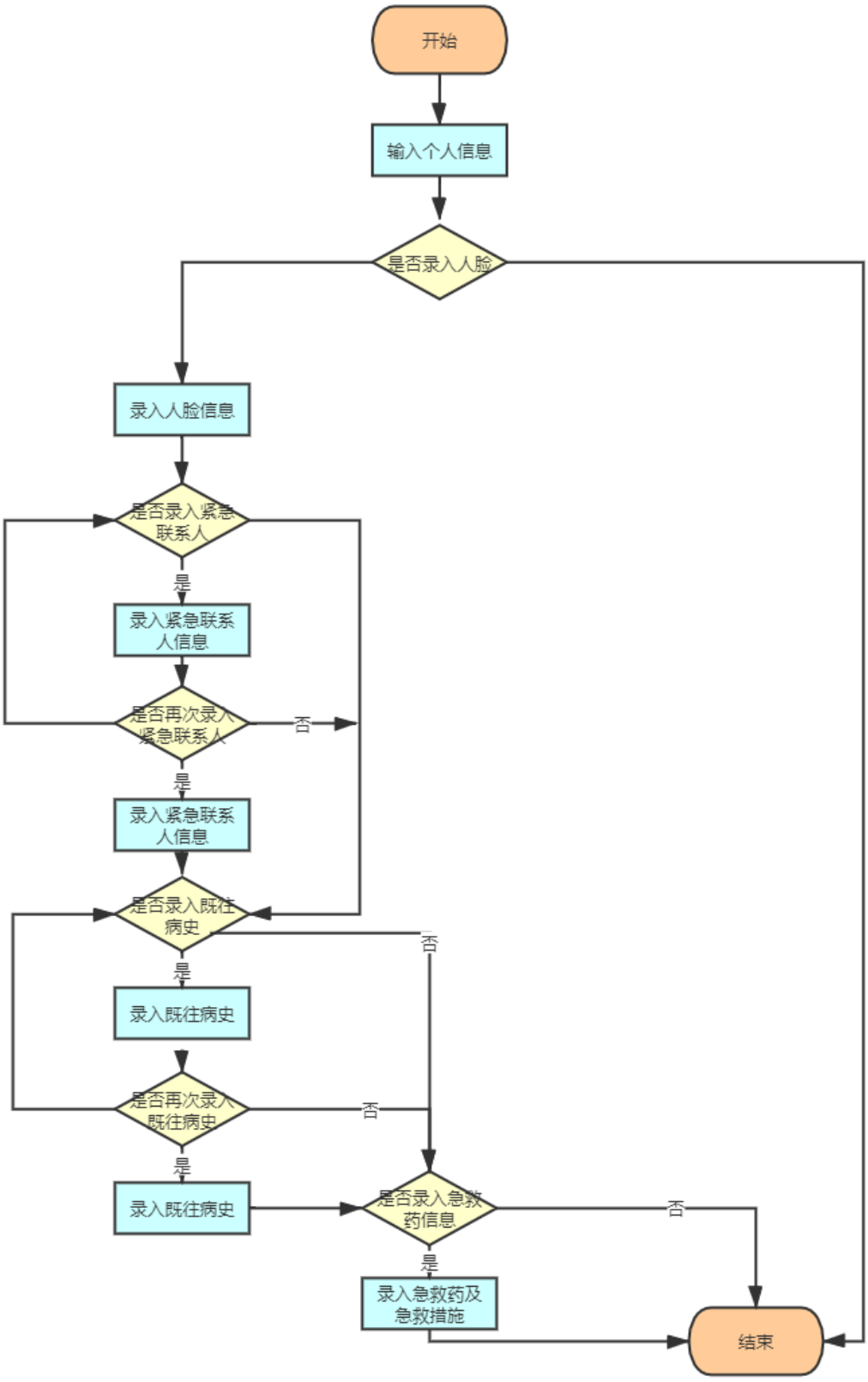


图 39：人脸及急救信息处理功能流程图

7.5.6 限制条件

1. 终端设备良好的性能
2. 网络环境良好
3. 位置信息获取正常
4. 用户信息输入格式正常