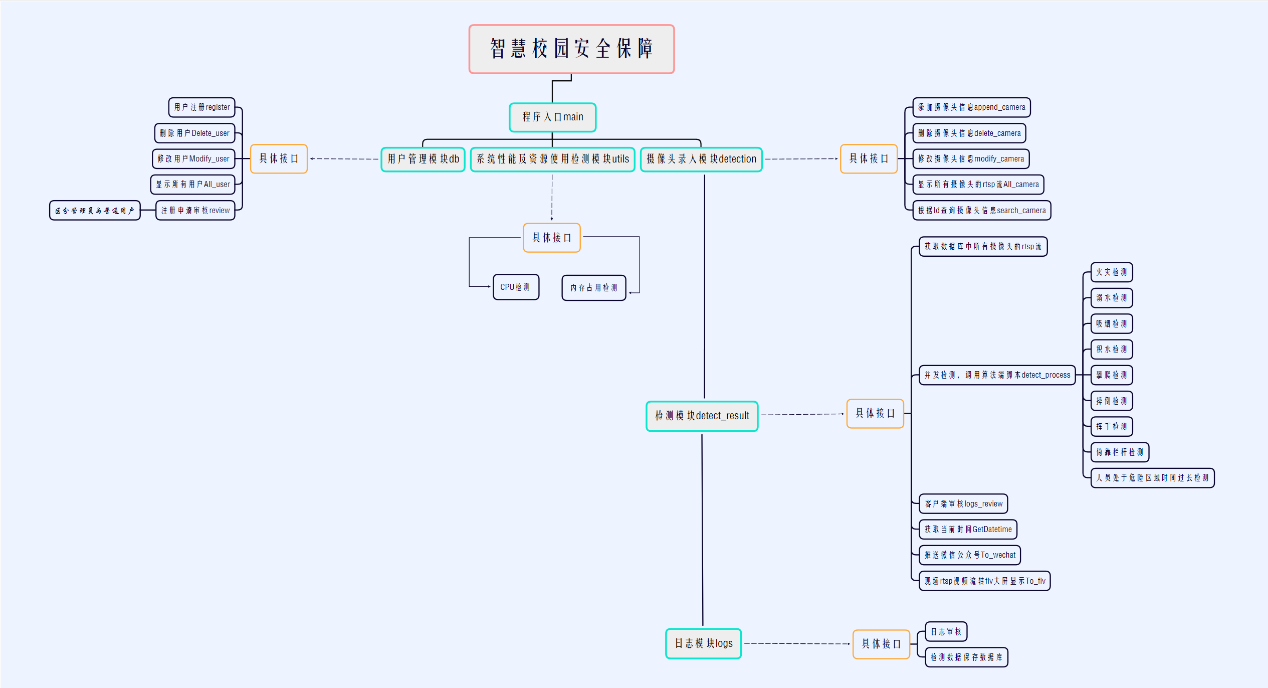
后端系统主要设计了六个主要模块，分别是：程序入口main、用户管理模块db、系统性能及资源使用检测模块utils、摄像头录入模块detection、检测模块detect\_result、日志模块logs。实现了服务器的创建，用户的注册、删除、修改、显示、审核，CPU和内存资源使用情况的检测，摄像头的录入，并发调用算法端脚本进行检测，生成报警数据添加数据库，完善的日志等功能。



后端采用Golang作为开发语言，更好地实现了项目中的轻量级并发要求。充分利用goroutine轻量级线程，高效地创建和管理线程。利用channel实现线程安全的通信和同步。同时提高了项目的高效性、简洁性和可移植性，支持代码在不同的平台上运行，也可以在一台机器上编译不同平台的程序。

为了满足前后端大量数据的交互，采用高性能的Gin框架，充分发挥其协程和非阻塞I/O等技术，实现高性能的路由和中间件处理，更快地响应请求，满足更多的并发请求的要求。使用Gin框架丰富的中间件，实现复杂的中间件功能，提高应用程序的可用性和安全性。

为了提高开发效率，适当地简化数据库操作、提高程序的可读性、充分利用数据库的功能、提高数据库进行CRUD操作的速度，采用GORM与原生SQL混合开发。使用GORM提供的方便的模型定义和CRUD操作，使用了事务、关联查询、预加载等高级功能。同时，使用原生SQL进行复杂的高级查询，提高灵活性。

项目支持对所识别到的违规事件进行告警，单次告警以事件为单位，支持推送信息至微信公众号。使用使用gin-web框架通过处理HTTP请求，请求API来获得访问令牌和关注公众号的用户ID列表，解析前端传来的JSON数据，构造消息发送到WeChat组，使用微信官方API将消息发送给所有关注公众号的用户。

保存数据方面，该项目共设计并使用两个数据库，分别是user\_login和detection。user\_login数据库中存放用户信息，detection数据库存放摄像头录入信息与检测报警记录日志信息。 支持完善的监控和日志，便于观测系统的性能和资源使用。

系统可以接入模拟现场的测试视频流，可以完成测试操作并直观显示业务结果。后端使用Go语言的Gin框架和GStreamer，通过gst-launch-1.0工具，实现了从接入模拟现场的测试RTSP视频流中提取H.264编码的视频数据，将其解码后，使用flvmux元素封装为FLV格式的视频文件，并将其保存在磁盘内，并将生成的FLV文件作为HTTP响应的内容类型，设置HTTP响应头为“video/x-flv”，将其传输给客户端。通过使用FFmpeg和GStreamer这两个多媒体框架，低资源占用率的条件下实现了视频的解码、编码、转换等操作；在同一个函数中并发地实现了客户选中的8个重点地区的rtsp视频流到flv格式文件的转换，实现了业务的灵活组合配置，可以适应不同项目的需求。