

软件概要设计说明 V1.2

						软件可靠性分析平台	
标记	数量	修改单号		签字	日期		
编制			会签				
						软件概要设计说明	
校对			标检			共 13 页	第 1 页
审核			批准				
会签							

修订记录

版本号	修订状态	简要说明修订内容和范围	修订日期	修订人	批准日期
V1.0	A	搭建平台与基本模型的接入	2025-11-20	王翊铭	2025-12-01
V1.1	A	UI 优化、api 接入、BPN 等算法实现	2025-12-02	杨恩祺	2025-12-02
V1.2	A	管理人员系统的实现、 cloudflare worker 的存储实现	2025-12-05	杨恩祺	2025-12-05

注：修订记录在体系文件发布后换版时使用，修订状态栏填写：A—增加，M—修改，D—删除

目录

1. 范围
 - 1.1 标识
 - 1.2 系统概述
 - 1.3 文档概述
2. 引用文档
3. CSCI级设计决策
 - 3.1 假设
 - 3.2 设计决策
 - 3.3 系统体系结构
 - 3.4 软件体系结构
4. CSCI体系结构设计
 - 4.1 子系统/软件单元命名规范
 - 4.2 软件单元/子系统汇总
 - 4.3 CSCI部件
 - 4.3.1 子系统关系图
 - 4.3.2 子系统：数据导入与预处理
 - 4.3.3 子系统：模型计算与可视化
 - 4.4 执行方案
 - 4.4.1 数据导入与预处理流程
 - 4.4.2 模型计算与展示流程
 - 4.5 接口设计
 - 4.5.1 接口标识和接口图
 - 4.5.2 (接口的项目唯一的标识符)
5. 数据（库）结构设计
 - 5.1 逻辑结构设计要点
 - 5.2 物理结构设计要点
 - 5.3 数据结构与程序的关系
6. 部署设计
7. 运行设计
 - 7.1 运行模块组合
 - 7.2 运行控制
 - 7.3 运行时间
8. 性能设计
9. 系统出错处理设计
 - 9.1 出错信息
 - 9.2 补救措施
 - 9.3 系统维护设计
 - 9.4 错误处理设计

10. CSCI详细设计

10.1 主要算法与函数说明

11. 需求可追踪性

11.1 需求到模块对应关系

11.2 接口到测试覆盖关系

1 范围

1.1 标识

“软件可靠性分析平台” 源代码目录：D:\软件可靠性分析平台； 主要入口 app.py，前端模板 templates/，静态资源 assets/。

1.2 系统概述

面向可靠性分析场景，提供多源数据导入（手动、CSV/Excel、MySQL、DeepSeek 智能映射），计算经典可靠性模型、轻量 AI 模型、时间序列预测、组合加权与 SDA 场景，支持可视化与报表导出。

1.3 文档概述

本文描述系统级设计、架构、模块、接口、数据、部署与运行方案，为开发、测试、运维及验收提供依据。

2 引用文档

- docs/requirements.md (需求)
- docs/test-report.md (测试)
- .env.example、requirements.txt
- 用户管理流程.md (Cloudflare Worker 操作)
- 软件用户手册.md

3 CSCI级设计决策

3.1 假设

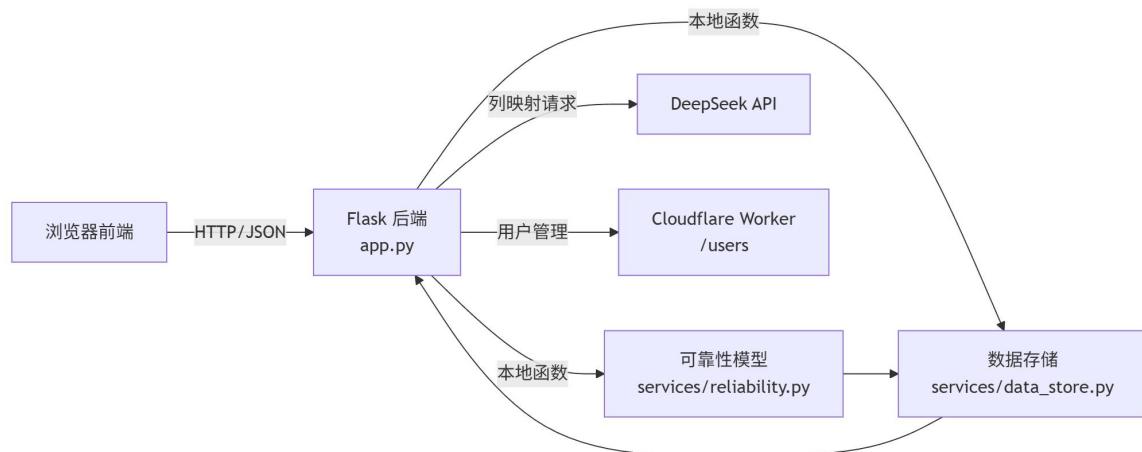
- 部署在本地/内网，单节点运行，无高并发要求。
- DeepSeek/Cloudflare Worker 可选，不可用时需平稳回退到本地映射/本地存储。
- 数据默认存内存或临时文件，未强制持久化数据库。
- 浏览器支持 ES6/Fetch，外网访问需自行加固。

3.2 设计决策

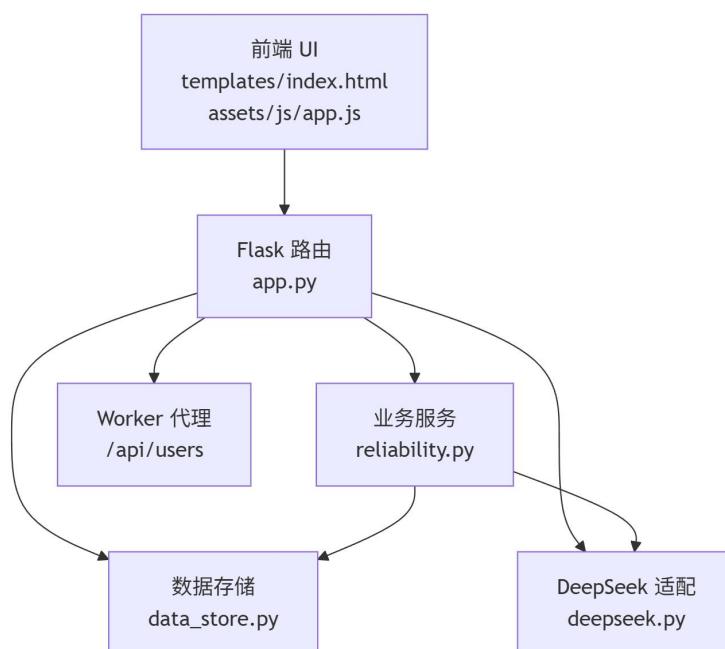
- 架构：前后端一体的轻量 Flask 应用，REST API + 前端模板。

- 数据：导入后先标准化为统一结构（module/failures/mtbf/runtime/timestamp/source）。
- 模型：经典模型（Goel-Okumoto、JM、Musa-Okumoto、Crow-AMSAA、Duane）+ 轻量 AI（BPN/RBF/SVM/GEP）+ 时间序列（ARIMA 近似、Holt-Winters）+ 组合加权/SDA。
- 安全：密钥仅存 .env，不入库不前端暴露；接口做基本校验与异常回退。
- 可观测性：控制台日志，必要时可扩展文件日志。

3.3 系统体系结构



3.4 软件体系结构



4 CSCI体系结构设计

4.1 子系统/软件单元命名规范

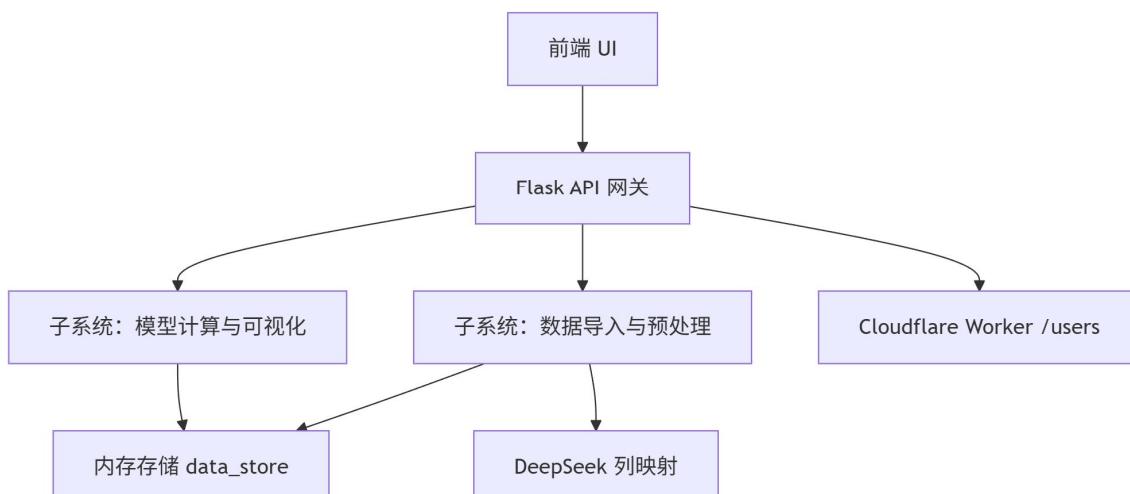
- 文件/模块名使用小写+下划线；类/数据结构使用驼峰；常量大写。
- 接口路径 /api/<domain>/<action>，仅含字母数字下划线。

4.2 软件单元/子系统汇总

子系统/单元	位置	职责
Web 前端	templates/index.html, assets/js/app.js, assets/css/style.css	UI、导航、图表、表单 上传、操作触发
API 网关	app.py	路由、请求校验、渲染、 导出
数据存储	services/data_store.py	内存存储、线程安全的记 录读写
可靠性模型	services/reliability.py	经典/AI/时间序列/组 合/SDA 计算
DeepSeek 适配	services/deepseek.py	列映射智能识别
Worker 适配	app.py (用户接口代理)	调用 Cloudflare Worker 进行用户管理

4.3 CSCI部件

4.3.1 子系统关系图



4.3.2 子系统：数据导入与预处理

- 范围：文件/手动/MySQL 导入，样本预览，列映射（本地/DeepSeek），标准化入库。
- 入口：/api/import/file、/api/import/mysql、/api/deepseek/mapping。
- 责任：样本预览、列映射、标准化、写入内存 store。
- 关键点：列名合法性校验、空文件检测、异常回退、DeepSeek 超时降级。

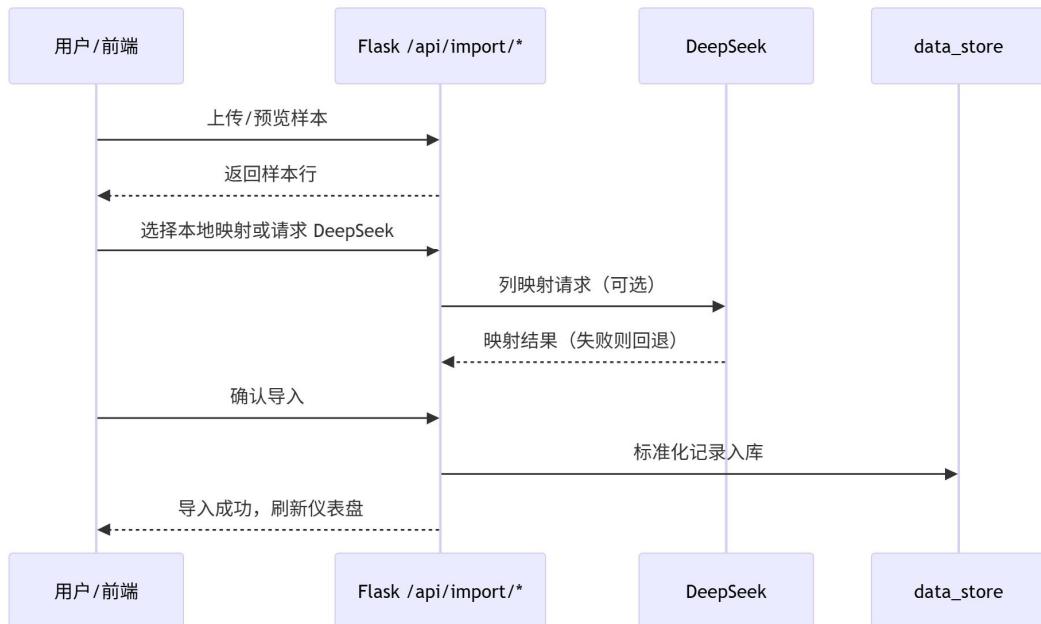
4.3.3 子系统：模型计算与可视化

- 范围：经典/AI/时间序列/组合/SDA 计算，Chart.js 数据与公式输出。
- 入口：/api/analyze/<section>、/api/export/html（内嵌）。
- 责任：按需触发各模型计算，返回 datasets+公式。
- 关键点：样本不足时降级、RMSE 溢出裁剪、MathJax 公式输出。

4.4 执行方案

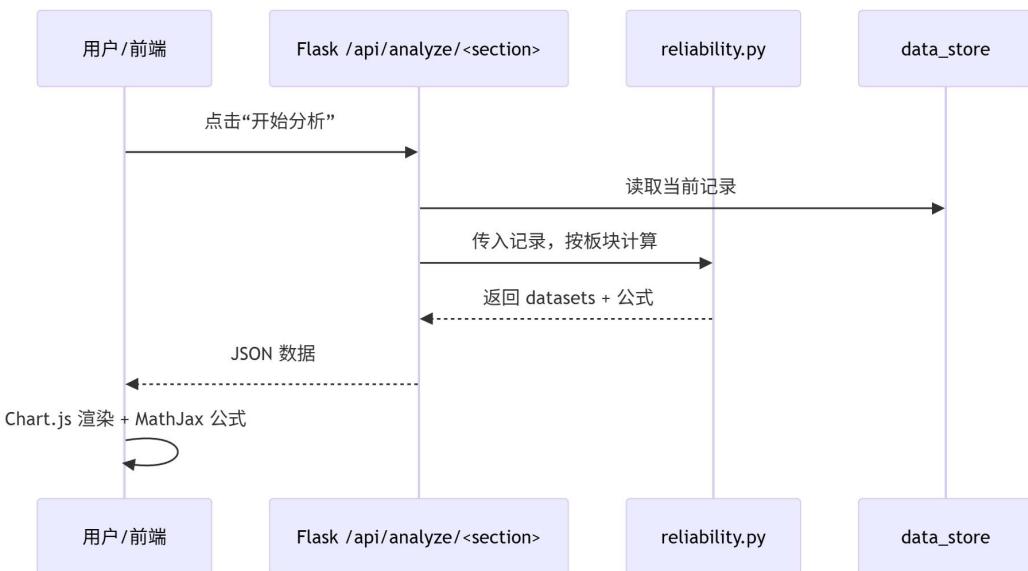
4.4.1 数据导入与预处理流程

（对应“数据导入与预处理”执行流程）



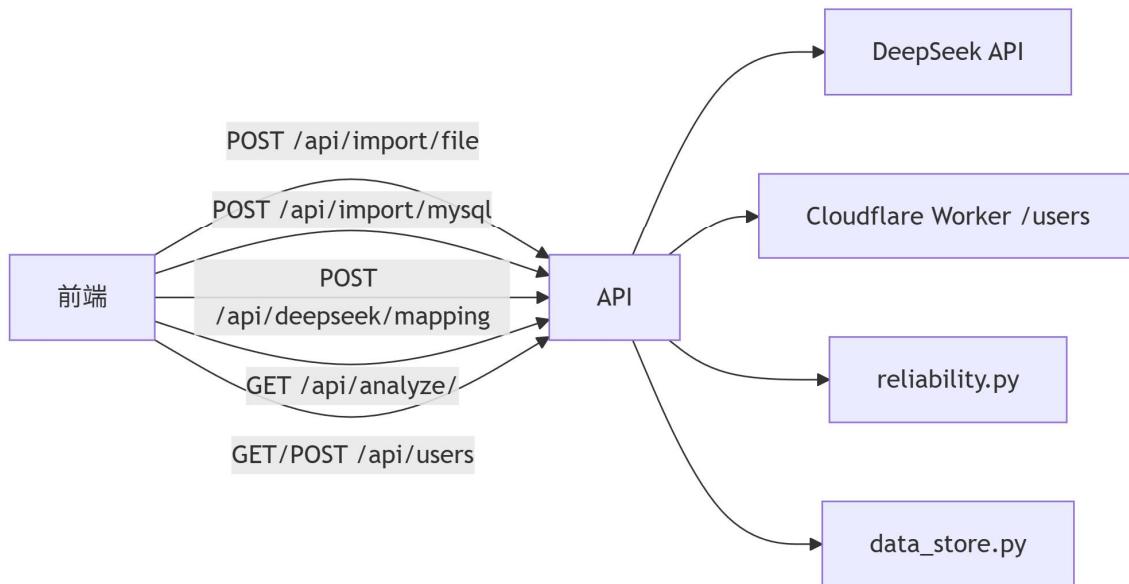
4.4.2 模型计算与展示流程

(对应“模型计算与展示”执行流程)



4.5 接口设计

4.5.1 接口标识和接口图



4.5.2 (接口的项目唯一的标识符)

接口	方法	说明	主要参数
file	POST	上传文件导入	file, mapping
file/sample	POST	上传文件样本预览	file

接口	方法	说明	主要参数
import/mysql	POST	MySQL 导入	dsn, table, limit, mapping
import/mysql/sample	POST	MySQL 样本预览	dsn, table, limit
deepseek/mapping	POST	DeepSeek 列映射	headers, rows
analyze/section	GET	按板块返回图表数据	section
users	GET/POST	查询/新增用户	name, role, email, status

5 数据（库）结构设计

5.1 逻辑结构设计要点

- 标准化记录字段：module、failures、mtbf、runtime、timestamp、source。
- 内存存储以列表为主，线程锁保护；可扩展为持久化表。

5.2 物理结构设计要点

- 默认内存；示例 MySQL 表 reliability_records（见 experiments/reliability_test.sql）。
- 列类型建议：module(varchar)、failures(int)、mtbf(double)、runtime(double)、ts(timestamp)。

5.3 数据结构与程序的关系

- _normalize_records 保证字段齐备；build_chart_payload 依赖统一字段生成 16 张图表。
- DeepSeek 映射输出列名 JSON，写入前转换为标准字段。

6 部署设计

- 开发：python app.py，默认 5000 端口。
- 生产：建议 WSGI（gunicorn/uwsgi）+ 反向代理；配置 .env，限制外网访问。
- 外部依赖：DeepSeek/Worker 需出网；不可用时前端提示并回退。

7 运行设计

7.1 运行模块组合

- Flask 主进程 + 内存数据存储 + 按需调用 DeepSeek/Worker。### 7.2 运行控制
- “开始分析”按钮触发后端计算；导入成功后自动刷新仪表盘。
- 导出：后端生成轻量 HTML，前端可打印 PDF。### 7.3 运行时间
- 轻量模型训练保持秒级；DeepSeek/Worker 调用超时设定 20s，失败回退。

8 性能设计

- 控制样本规模（建议 < 万级）以保证前端渲染；按需加载各板块。
- AI 训练为轻量 batch GD，防止长时间阻塞；RMSE 做裁剪防溢出。
- 外部调用超时保护，防止卡死主流程。

9 系统出错处理设计

9.1 出错信息

- 上传为空/列不合法：返回 4xx 提示。
- DeepSeek/Worker 调用异常：提示失败并回退本地映射/本地列表。
- 模型计算溢出：RMSE 裁剪，返回可渲染的安全值。

9.2 补救措施

- 校验输入、重试导入、检查网络/密钥；必要时切换本地映射。
- 清洗异常数据后重新分析。

9.3 系统维护设计

- .env 中集中配置；可扩展文件日志；依赖通过 requirements.txt 管理。

9.4 错误处理设计

- 统一 try/except 捕获外部调用；返回 JSON 错误码与消息；前端提示并允许重试。

10 CSCI详细设计

- 可靠性核心 (`services/reliability.py`)
 - 预处理: `_normalize_records` 统一字段
(`module/failures/mtbf/runtime/timestamp/source`)，裁剪异常值。
 - 经典模型: `_goel_okumoto_series`、`_jelinski_moranda_series`、`_musa_okumoto_series`、`_crow_amsaa_series`、`_duane_growth_series` 生成累计/强度/斜率序列；参数由导入数据自适应估计。
 - AI 轻量模型: `_train_linear_gd`、`_train_bpn`、`_train_rbf`、`_train_svm_linear`、`_train_gep_poly`，基于 `failures/runtime`→`mtbf` 训练，输出预测、RMSE、权重/中心/多项式系数。
 - 时间序列与融合: `_arima_projection`、`_holt_winters` 生成短期预测和平滑；`dynamic_ai_weight` 计算权重；`_weighted_ensemble` 计算 SDA 场景评分。
 - 汇总: `build_chart_payload` 生成 16 张图的 datasets；`build_section_payload` 支持按板块懒加载；`build_dashboard_payload` 聚合仪表盘数据。
- DeepSeek 适配 (`services/deepseek.py`)
 - `_build_prompt` 拼接表头+样本；`infer_mapping_from_sample` 调用 DeepSeek Chat Completions 返回 JSON 列映射，异常抛出 DeepSeekError。
 - 超时/失败由上层捕获并回退本地映射。
- 数据存储 (`services/data_store.py`)
 - 线程安全内存存储，提供 `add_records`、`get_records`、`clear`，记录时间戳与来源。
- 应用层 (`app.py`)
 - 环境加载 `_load_env_file`；路由
`/api/import/file|mysql|...`、`/api/deepseek/mapping`、`/api/analyze/<section>`、`/api/users`；前端模板渲染 /；导出 HTML 入口
`_render_export_html`。
 - 校验：标识符合法性、空文件检查、异常返回 JSON；DeepSeek/Worker 失败回退。
- 更多算法推导与变量解释详见《个人算法报告》和源码注释。

11 需求可追踪性

- 需求到模块对应关系：
 - 数据导入（手动/文件/数据库/智能映射）→ `app.py` 导入路由 + `services/data_store.py` + `services/deepseek.py`。
 - 经典模型计算 → `services/reliability.py` 经典模型函数 + `/api/analyze/classic`。
 - AI 模型计算 → `services/reliability.py` AI 训练函数 + `/api/analyze/ai`。

- 时间序列 → `_arima_projection`、`_holt_winters` + `/api/analyze/ts`。
 - 组合加权/SDA → 权重计算、SDA 评分 +
`/api/analyze/ensemble`、`/api/analyze/sda`。
 - 模型对比 → `/api/analyze/compare` 返回 PKR/性能对比。
 - 用户管理 → `/api/users` 代理 Cloudflare Worker; Worker 端 `/users`。
 - 文档/导出 → 前端文档入口，`_render_export_html` 导出 HTML。
- 接口到测试覆盖关系：
 - 接口列表详见本文件 4.5.2；`docs/test-report.md` 记录了导入、DeepSeek 回退、各模型分析、用户管理等用例执行结果。
 - 若新增接口/需求，应在 `docs/requirements.md` 更新需求映射，并在 `docs/test-report.md` 补充对应用例与结果。