项目文档

序言

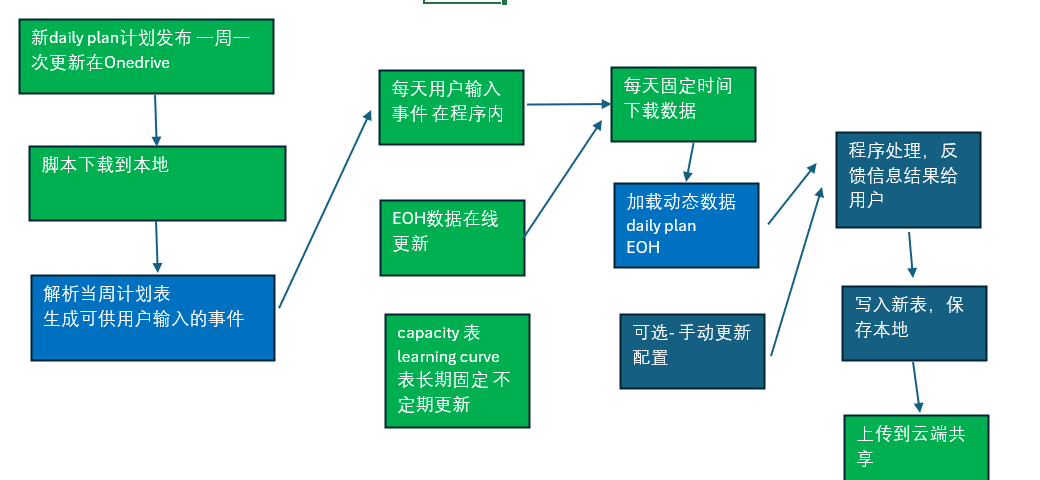
这是一个生产排班系统，目的是实现自动处理和调配生产事件带来的生产计划的变化。系统以本地程序解析本地数据表来运行以实现功能。

功能需求

主要功能点：  
数据加载：导入HSA Daily Plan原生成计划、HSA FG EOH（成品库存）、HSA Capacity（产线产能）、Learning Curve（改线后效能恢复曲线）等生产相关数据。  
事件管理：手动添加、删除、导入生产事件，并以列表形式展示。  
事件处理：根据预设的业务逻辑和算法，对生产事件进行处理，生成调整后的新生产计划和处理策略汇总。  
结果展示与导出：展示事件处理的详细结果和统计信息，导出为Excel文件。  
系统设置：配置自动加载时间、备份选项、日志级别等。  
日志记录：实时显示系统操作和处理日志。  
Python 3: 核心编程语言。  
Tkinter: 用于构建图形用户界面（GUI）或者其他库也可以  
Pandas: 用于高效地处理和分析表格数据（如Excel文件）。  
NumPy: Pandas 的底层依赖，提供数值计算能力。  
os, datetime, json, re, typing, threading, time: Python 标准库，分别用于文件系统操作、日期时间处理、JSON配置读写、正则表达式、类型提示、多线程和时间操作。

* daily plan的解析，从原始人工计划表解析数据，生成程序理解的格式，也可以直接加载目前蓝色的daily plan ,但这版是我手动编辑的不是原始表，其实变化不大，推荐从原始表解析，但原始表有格式不规范的地方，比如各产线的PN行数不同，可以列出这些问题以纠正。
* 每次运行从daily plan表解析数据，获取每天每条线对应的产品号，生成事件表
* 基本的程序模式，定期比如每天早6点晚4点加载动态数据 daily plan ，FG EOH ，加载其他配置数据capacity ，learning curve 来获取产线能力和换线折损系数(不是每天变更，有需要可以手动加载更新一次)
* 加载后根据逻辑表定义的逻辑流程图来判断事件的相应处理方式，每次运行前需要注意判定事件输入信息的正确性，比如要求取消5000产量，但实际根本没有5000，比如要求转换产品，但是从A还是转换成A，需要加以判断
* 关于信息呈现，读取到的事件表，需要打印给用户看一次，每次处理结果即采取的策略，以及中间事件结果，也有信息打印给用户，这些通知如何写，已经定义在流程里面，可以扩展我没有想到的中间事件状态，信息尽量完善，让用户知道运行到哪里了，结果如何。
* 处理完成后，新的计划表重命名保存到本地，结束
* 注意- 原始数据文件是人工填写表格，有很多人工误差痕迹 包括空格，换行，口语化表达等无法统一处理的元素，建议首先对初始计划表进行清洗，或者定义规范格式，以便程序格式化操作表格。参考原始表格
* 为了可以把握工期，和降低开发难度，当前要求只完成逻辑模块1和面板部分，本文档最后面有介绍，看效果再约定模块2，模块3的开发日程和报价

完整的实现流程，蓝色部分程序负责，绿色部分我方负责 ，参见 整体框架这个文件



数据获取

数据分两类，一类动态更新，一类长期稳定，不用每次加载，可以让用户手动加载

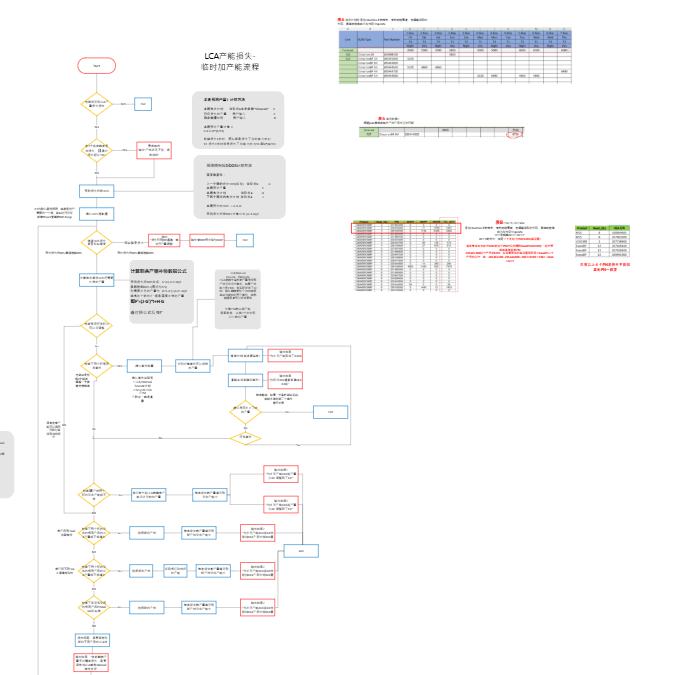
* FG EOH - 各个产品的库存追踪表，每天会更新两次，运行前读取
* daily plan - 需要更新的计划表
* daily plan2 额外的5份计划表 供数据测试
* Capacity - 各条产线的理论生产能力 长期固定
* learning Curve - 产线换线后能力折损表 长期固定
* 24YWW36 HSA Build Rev2.0 原始的计划表，命名格式是24Y年数+WW01.02周数 HSA Build Rev2.0
* 程序，参数命名规范，有功能介绍注释，方便理解和后期维护
* 主程序和组件程序尽量分离，分开加载调用，不要都写在一个主程序里面

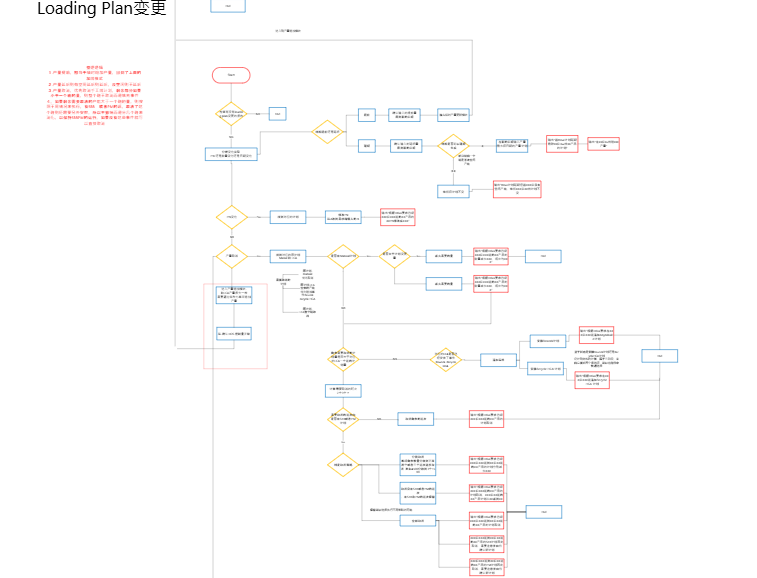
具体事件表的设计，参加事件表文件



逻辑流程图 分为3大部分，加线模块 ，减线模块，其他辅助模块 ，具体pdf逻辑流程图中

加线模块



减线模块

其他模块

