Sebill

Sebill是一个账单系统，sebill是整个custody业务线上的最后一环也就是出账单。我们从上游的Secore获取数据，每个月ops会设置每个client的invoice的信息。我们会对billiable activity，根据一定的Fee Schedule定价规则出账单，最后把invoice发给客户。也会把生成的ITR file发给flexcube。这是我们的主要业务。我们这个系统代码量非常大，业务也很复杂，前端是angular，后端是.net开发。我们有二十多个ecs service，还有一些legacy的service目前还在逐步向esc迁移的过程中。

我在组里是tech mandantory，我主要是做 .NET 后端这边的接口开发和逻辑处理。也会涉及到前端和数据库的改动。我们组目前也在做一些innovation的项目，比如最近也在找一些跟AI结合的use case，我也有参与。

Hackthon-Chatbot

这是一个我们custody当时为了参加Hackthon做的一个项目。我们的目标是使用LLM，专门为新来的同事做一个chatbot，因为新来的同事往往对很多东西都不熟悉，比如怎么搜索相关的网站，怎么setup自己的环境，怎么申请CMP。。。于是我们给LLM输入一些confluence的数据，存在一个sqllite里，然后用户提问的时候我们可以将confluence作为我们自己的知识库，来回答用户的问题。

使用到了python的streamlit这个组件，它可以帮助我们快速搭建一个聊天助手。在得到用户的输入之后，首先会在db里查找是否有用户输入对应的关键字保存在数据库中，如果有，通过embedding之后的query和db里的embedding向量使用FAISS进行相似性检索，返回最相似的前五个数据。如果db没有类似数据，则直接根据用户输入从confluence检索相应文章，然后和query对应的embedding向量进行相似性检索，得到最优的数据交给LLM，让其分析并得到summary，然后返回给用户。

Anomaly Detection

需求是想要对金融领域的异常数据进行检测，其实是一个非常常见的需求也已经有很多解决的办法了，有基础的ML模型也有神经网络、LLM都可以解决这类问题，也已经有很多相关的经典论文发表出来。我们考虑到citi目前使用ai不是很方便还是选择了基础的ML模型。目前我们选择了三个模型，使用kaggle上的一个信用卡欺诈数据集来做测试。后续会在这几个模型中选择一个比较合适的然后用真实的数据来做。

我选的是OneClassSVM，这是一个无监督学习的模型，常用于**异常检测（outlier detection）。它的目标是学习一个只包含“正常”样本的模型，从而识别哪些数据点是“异常”的。它会学习一个边界（或超平面）将大部分训练数据包围起来，目标是区分正常区域和潜在异常。kernel 参数控制使用的核函数类型（常用 rbf（高斯核）适合非线性数据），nu 参数控制异常比例。适合没有异常样本的情况，但不太适合大规模数据集。**

组内AI：解析INC

自我介绍 release coordinator\_sharing

您好，我叫于彤，研究生毕业于上海大学软件工程专业。目前在Custody Sebill 团队担任软件开发，主要负责后台系统的开发与维护工作。在毕业之前我曾在Sebill team实习两个月，期间了解了公司文化和系统架构，为后续的工作打下了坚实基础。在这一年的开发工作中主要使用到了 .NET（C# + ASP.NET Core）开发，同时我也具备扎实的 Java 开发能力，熟悉 Spring Boot、Spring Cloud 等主流框架。

过去一年的工作中，我参与并推动了多个关键项目和任务：我主导完成了 Sebill 系统后端服务从 .NET 6 到 .NET 8 的升级，确保系统稳定运行；并负责多个服务的logging migration的工作，提升了Sebill 监控services的效率，并主持了相关tech sharing session。同时，我参与了马来西亚 E-Billing 项目的开发与测试，在项目中主要负责和ISG Cloud和AMC团队沟通并获取数据的任务，并在项目结束后互动承接了host project sharing的任务，做了一个面向global dev/UTT/OPS团队的project sharing。也和Flexcube团队沟通处理过通过kafka发送ITR file的问题，积累了关于kafka消息队列的经验。此外，我持续关注组内的code quality，修复了目前sebill代码中存在的 SonarQube issue并补充unit test，以达到我们80% coverage的目标；此外我也曾经担任过 Release Coordinator，组织和协调release day的一些流程，包括release前check大家的jira信息，当天保障发布流程顺利进行并且最后进行UI checkout等等。

此外，在 AI 创新方面我也保持积极探索的态度。曾与custody的其他analyst一起参加 Hackathon，**基于大型语言模型（LLM）构建了一个面向新员工的Chatbot**，支持从 Confluence 中学习知识数据来帮助新员工快速onboard。同时，sebill组内也在探索将 LLM 与 ServiceNow 数据结合，用于优化INC处理流程。并且我们CUSTODY内部也尝试使用传统机器学习方法对金融数据进行异常检测。

（结合新team的需求）

经过这一年的工作锻炼，我不仅提升了系统开发能力，也积累了丰富的跨团队合作与项目推动经验。具备良好的英文沟通能力，能够自如地与全球团队协作，并乐于分享与表达。未来希望在 Java 技术方向持续深耕，同时不断探索 AI 在金融行业的实际应用，持续为团队和业务创造价值。

Good morning, my name is Tong Yu. I just got my master's degree in Software Engineering from Shanghai University last year. I am currently working in the Custody Sebill team as a software developer, mainly responsible for the development and maintenance of our backend systems. Before my graduation, I interned in the Sebill team for two months, during which I gained a good understanding of the company culture and system architecture, laying a solid foundation for my future work. Over the past year, I have primarily worked with .NET (C# + ASP.NET Core), and I also possess solid Java development skills, with familiarity in mainstream frameworks such as Spring Boot and Spring Cloud.

Over the past year, I have participated in and driven several key projects and tasks. I led the upgrade of the Sebill backend services from .NET 6 to .NET 8, ensuring stable operation of the system. I was also responsible for the logging migration of multiple services, which significantly improved the monitoring efficiency of Sebill services. And I hosted a technical sharing session on this topic. Meanwhile, I participated in the development of the Malaysia E-Invoice project, where I was mainly responsible for communicating with the ISG Cloud and AMC teams to obtain the required data. After the project was completed, I voluntarily organized a project sharing session and delivered a presentation to the global Development team, UTT team, and Operations teams. In addition, I worked with the Flexcube team to implement a new feature for sending ITR files through Kafka, gaining hands-on experience with Kafka message queues. I also continuously focused on code quality by fixing existing SonarQube issues in the Sebill codebase and adding unit tests to help achieve our 80% coverage goal. Moreover, I served as a Release Coordinator, organizing and coordinating tasks on release days, including checking team members’ JIRA tickets before the release, ensuring a smooth deployment process, and performing the final UI checkout.

In addition, I have maintained an active interest in AI innovation. I participated in a Hackathon with other analysts from the Custody, where we built a chatbot for new employees based on large language models (LLMs). At the same time, our Sebill team has been exploring the integration of LLMs with ServiceNow data to optimize the incident handling process. Within Custody, we have also tried using traditional machine learning methods to detect anomalies in financial data.

Through this year of work, I have not only improved my system development capabilities but also accumulated valuable experience in cross-team collaboration. I have strong English communication skills, allowing me to work seamlessly with global teams, and I enjoy sharing and expressing my ideas. In the future, I hope to further deepen my expertise in Java development and continue exploring the practical use case of AI in financial industry.

收获：

1. 情绪稳定
2. Malaysia project sharing
3. 熟悉了金融系统的业务

Q&A

1. 组内的组织架构是怎么样的（提前了解对方manager report给谁，team的名字，team内部有多少个成员）？有没有专门的QA/UTT来测试，有没有global的team？
2. 平时是否会做一些innovation的项目/AI use case的研究？
3. 对这个职位的期望是什么样的，对analyst在组里的一年之内的成长有哪些预期？
4. 组里是否有自主权

**Q1: Tell us about a time when you introduced an improvement in your team which has been your suggestion.**

**Answer:**

While working on the Sebill backend services at Citi, I noticed that the logging formats across different services were inconsistent, making it difficult for developers to trace issues and for operations teams to monitor effectively. I proposed standardizing our logging by introducing structured logging with correlation IDs and consistent log levels.

To ensure adoption, I created a proof of concept and shared a technical document outlining the implementation steps and benefits. I then hosted an internal tech-sharing session, walking the team through real examples and demonstrating how structured logs improve traceability across services.

The improvement was gradually implemented across our backend services. After rollout, developers reported faster root cause identification during incident handling. I also worked with our monitoring team to visualize the logs in dashboards, making the change measurable and impactful. The change became a new team standard and was shared across other teams.

**Q2: Tell us about a time you had to build a relationship with a new customer/stakeholder.**

**Answer:**

During the Malaysia E-Billing project at Citi, I worked closely with the AMC team for the first time. As the backend developer, I was responsible for delivering modules that required seamless integration with their systems.

To build the relationship, I scheduled a kick-off call to understand their needs and constraints. I proactively followed up via email with technical questions and maintained clear, regular updates on our development progress. When we encountered issues with message formatting, I organized a joint troubleshooting session and facilitated a shared solution.

By being transparent and responsive. We built a collaborative partnership, and the integration was completed smoothly.

**Q3: Describe a situation that required you to manage a number of pieces of work at the same time.**

**Answer:**

As a developer on the Sebill team, I was once responsible for three concurrent tasks: preparing a technical sharing session for the global Dev Ops UTT team, coordinating the monthly release as Release Coordinator, and delivering backend features for the Malaysia branch.

To manage my time, I prioritized based on deadlines and stakeholder impact. I used a Kanban board to track tasks and reserved dedicated focus blocks on my calendar. For the release, I prepared checklists in advance, collected required sign-offs, and ensured testing was completed. In parallel, I drafted and rehearsed the tech-sharing content over several days. I also broke down development tasks into smaller units to maintain daily progress.

As a result, all three responsibilities were completed successfully. The release was delivered on time with zero rollbacks, the presentation was well received by global colleagues, and my development tasks passed QA testing smoothly. This experience strengthened my time management and multi-tasking skills under pressure.

**Q4 invoice中都有哪些数据**？

分为两部分：

Cover page：

包含地址，invoice的时间，skaccount no，和对应的amount

Detailed invoice

ClientId ， clientName， skaccount，total amount

**Q5 在组里都做了哪些工作？**

1. .net upgrade

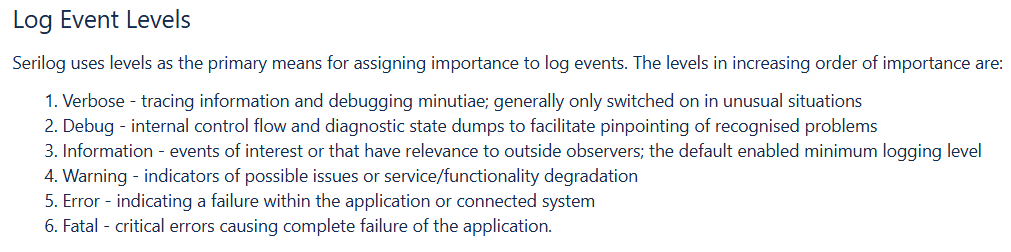
在csproj文件中修改，或者使用upgrade assistant，可以同时将microsoft包进行升级。需要注意①非microsoft package版本升级，需要自己手动去nuget官网检查》net8的版本 ②在线上build的时候使用.net8， 这需要在dockerfile中指定sdk版本，并且在pipeline中修改配置。遇到了net8线上build失败的问题。是因为在文件夹格式不符合标准格式的时候需要给出solution file的相对路径，但net8解析相对路径会出错, 检查报错log发现会去/workspace/source/去查找solution file，于是修改pipeline配置即可。

1. Logging migration

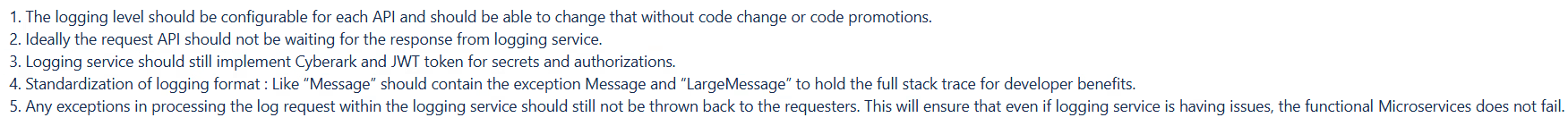
背景是我们的log之前一直是掉的CSD的api，记录在csd的数据库。我们没办法自由的修改logging level也没办法很方便的查看log。于是我们自己使用serilog来创建了一个自己的log的service，其他service在需要记录log的时候直接调用log service的api即可。这样的话 所有的logging level以及相关的配置可以在我们自己的数据库中自由修改。

主要在数据库中创建配置表：logginglevel，ConfigupdateInterval，loggingBatch，loggingInterval

Logging level：



要求：



Sharing内容：

1. 介绍serilog
2. 介绍loggingapi如何实现
3. 如何进行logging migration

难点：token从何而来？本来直接call token generator，但是从前端传来的请求中存在token，可以直接从HttpContext中获得

1. AMC

背景是去年的十月份左右，我们需要为Malaysia做一个e-billing的project。是马来西亚branch有特殊的要求，要求我们的invoice在发到client手里之前，先发到IRBM做校验，校验通过之后他们生成一个invoice发给我们，我们再发给client。

这个项目涉及到两个partner，首先我们需要从AMC获取数据，比如legal name；TIN ；BRN； address等等；获取到数据之后我们会先生成一个staging file里面包含了生成invoice需要的数据（AMC data，invoice amount），最后会call GTES的API发到GTES去校验。

我在项目中负责的就是通过ISG Cloud去获取AMC数据的这部分。我们会每天零点跑一次获取AMC数据的job来call一次AMC api。

难点：沟通API path

1. Kafka

背景是需要为新加坡branch开发一个功能，就是使用kafka来发送ITR file。所以根据这样的需求我们team是作为kafka的producer来发送数据到kafka，同时呢也会作为consumer来消费fexcube的response，从这个response中我们可以看到这个ITR file是发送成功还是被reject了。做这个task的时候需要先向cip team申请一个topic，然后发送的时候带上16为的messageid。其实这个task最主要的是沟通，因为需要按照flexcube的要求不断调整ITR file的数据格式。

难点：沟通

1. 优化Proc

当时我在做一个task，需要改动一个proc，在proc中新join一张表，然后获取其中的几个列。当时做完呢我自己测试是没有问题的，因为我当时忽视了数据量的大小，在release当周的regression test中QA用数据量很大的branch测试发现接口超时报错503。最后检查下来发现因为我的那个proc执行了十多分钟都没跑出结果，接口超过五分钟没有响应就会报错。当时也是紧急的做了一个修复，其实就是加个非聚簇索引就可以很大程度的加快检索的速度了。

难点：加索引