大家好，很荣幸可以向大家详细汇报我们关于Hibernia-Sino Travel insurance company的项目进展情况。接下来我们会详细展示我们的具体项目内容和time-line

整个项目我们采用前后端分离的开发方式，前端使用vue框架，后端使用Spring boot框架，使前后端同步开发提高工作效率，并减轻未来业务流量增加带来的服务器负载。并通过异步加载，保障服务器的响应速度。

首先用户和员工在登录页面输入账号密码，通过VUE框架中的iview组件库，提供输入框给用户输入用户名密码，建立Axios中的post方法将数据发送到后端后，由spring security 进行用户认证，并实现用户授权，用户和员工会分别进入对应的应用页面，在页面的构建过程中，利用Vuex建立的Store（仓库）进行响应式编程。

后期我们将配合Spring Security安全框架使用OAuth2，来实现第三方账户的授权登陆服务同时实现未授权客户访问受保护页面的时候的网页重定向功能。

用户进入页面后，可以申请保单赔付，并填写申请表单。其中包括了：客户姓名，保险种类，行李信息，行李丢失经历及图片描述，在申请保单赔付的过程中，利用集成的vue-quill-editor给客户提供上传文字描述和图片等附件的窗口。提交后再次通过Axios将数据发送至后端，并将数据储存在mysql中。我们将使用持久层框架Mybatis，实现sql语句和程序代码的解耦，提高代码的可维护性。后期在逻辑上，我们将对提交申请的保单是否存在和有效期等进行自动筛查，不符合条件的申请将不被提交；在技术方面，我们将使用redis进行数据缓存，以缓解大规模用户同时访问时对服务器和mysql的压力。

员工登入后，系统会从后台数据库请求所有用户所提交的申请表单数据，以table形式展现在页面上，员工可以点击查看特定申请表单的详细信息，选择同意或者拒绝赔付并附带理由。申请的表单被处理后会附带上状态属性，储存在数据库中。用户可以在回执页面查看审批结果。

考虑到不同国家的用户，我们采用了VUE框架自带的vue-i18n插件多语言的切换

在项目开发的过程中，我们项目组选用螺旋开发模型，在多次迭代开发的过程中插入风险分析和客户评估，并据其及时做出调整，减少开发过程中风险造成的损失。

前后端部分：（77）

从第二周到第五周，完成了第一次迭代。

二周完成了计划的制定和业务逻辑分析并形成文档。

第三周、四周启动项目，安装软件配置环境，搭建架构，并学习相关语法

第五周实施工程并进行客户评估。包括了基础页面制作和后端数据库设计等~~（web首页， 登陆页面， 注册页，表单页，员工表单 About Me页）~~。（106）

第二次迭代计划从第六开始，第八周结束，具体时间可以在这里看到：

在二次迭代中，我们完成了业务逻辑的二次整理、前后端交互设计、对数据的增删改查功能和风险评估。并完成了双语功能和页面的二次翻版与美化。并将于第八周完成前后端基本交互任务（114）

第九周到第十二周，将要完成最后一次迭代任务，具体计划如下：

我们将在第九、十周优化应用性能并丰富其功能，例如数据缓存、第三方登陆、受保护页面的重定向等。第十一周就会完成手机端的适配工作并将应用搭载在云服务器上，十二周周一，便可成功交付。（117）