-------------------------------------------------------------------------------------------------------------

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技术和time-line

就项目整体而言，我们采用~~的是~~前后端分离的开发方式，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，保障服务器的相应速度

（前端技术：）

在前端基础框架方面，我们选择使用了VUE框架，~~这种框架的优势在于它的核心库只关注视图层，在和后端的链接方面有较大的优势。~~通过vue的聚焦视图层与插件化，让我们可以在相同的时间和花费下，为客户交付画面更加优美，功能更加完善的网站成功。

我们采取响应式编程~~是VUE的一大特点~~，利用Vuex建立的Store（仓库），当组件从store中提取状态信息的时候，如果状态信息产生了变话，相应的组件也会相应的得到高效更新。这使在调试的过程中可以实时的对修改结果进行监控，提高开发效率。

并且，通过利用iview的组件库，我们在更短的时间内完成制作更加精美的网页，并~~我们~~可以依据客户的需求进行实时的修改。

除此之外，在我们的框架中还集成了vue-quill-editor，并通过它来提供客户在索赔的时候上传文字描述和图片附件的窗口

----------------------------------------------------------------------------------------------------------------

（双语功能技术没有写，待补充）

（前后端连接：）在前后端连接部分，

我们使用一个基于promise的HTTP库——Axios来实现前后端的数据交互。使用Payload形式提交数据，~~他可以并~~通过Get和Post方法向后台发送数据。在请求过程中还可以实现~~很多的~~多种可控功能，如拦截器可以在请求或者响应时拦截HTTP请求来进行处理。同时还能防止XSRF（Cross-site request forgery）攻击。

----------------------------------------------------------------------------------------------------------------

（后端技术栈：）在后端

1. 我们使用spring boot框架进行后端的开发，~~它可以~~简化新Spring应用的初始搭建以及开发过程，通过导入“starter”模块，可以将许多的依赖添加到工程之中。

2. 我们使用持久层框架mybatis，通过将sql语句写在xml文件中，来解除sql与程序代码的耦合，便于统一管理和优化，提高了可维护性。

3. 为了保证应用的安全性我们采用了Spring security框架，其中包括了用户认证（Authentication）和用户授权（Authorization）两个部分。其框架本身以及spring框架带来的灵活性，能够很好的满足我们应用开发的典型需求。

4. 同时，我们还将配合Spring Security安全框架使用OAuth2，来实现未授权客户访问受保护页面的时候的网页重定向，同时通过它我们还实现了github和qq账户授权登陆服务。

5. 我们还将使用redis来缓存数据，以缓解大规模用户同时访问时对服务器和mysql的压力。

（如何实现云操作待补充）

-------------------------------------------------------------------------------------------------------------------

TimeLine

在项目开发的过程中，我们项目组选用~~的是~~类似于螺旋开发模型，在多次迭代开发的过程中插入风险分析和客户评估。减少开发过程中风险造成的损失。通过每次迭代加入的客户评估及时对开发方向进行调整。

前端部分：

从第二周开始，到第五周结束，完成了第一次迭代。

二周完成业务逻辑分析，制定计划，风险分析和package相关文件书写。

第三周，四周实施工程，从安装相关环境于软件开始，搭建架构，学习相关语法

第五周实施工程并进行客户评估。包括了web首页， 登陆页面， 注册页，表单页，员工表单 About Me页等的制作

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和前后端交互设计和风险评估。并完成了页面的二次翻版于美化，和功能的叠加。并将于第八周完成前后端所有交互任务

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：

我们将在第九周完善基本项目功能，并与第十周丰富相关功能。第十一周就会完成手机端的适配工作，十二周周一，便可成功交付。

（后端换色按三次迭代加入文稿）

~~--------------------~~**前端timeline具体时间展示（美化整理后放入ppt，并整理相关文件，按顺序截图放入相关文件并打印作为附加材料提交）**~~------------------------------------------~~

* 前端部分从第二周开始，到第五周结束，完成了第一次迭代，具体时间线如下：

（制定计划：风险分析：实施工程：客户评估：）

* 2-25 二周
* 制定计划：风险分析
* 2019-2-27 完成业务逻辑初版分析（董乐辉版1）
* 2019-2-28 计划制定完成，package，gantt书写完成
* 3-3 完成风险分析
* 3-4 三周 3-11 四周
* 实施工程
* 3-4 Nodejs，Npm等前端开发环境的安装
* 3-7 框架搭建
* 3-11 Vue语法学习
* 3-18 五周
* 实施工程：客户评估
* 3-18 完成web首页（初版）
* 3-19 完成 登陆页面（初版）
* 3-22 完成 注册页，表单页，员工表单 的表格制作（初版）
* 3-24 完成About Me制作（初版）
* 3-25 完成用户评估
* 从第六到第八周结束，将完成第二次迭代计划：
* 3-25 六周
* 制定计划：风险分析
* 3.26 业务逻辑初版分析（张天卉版）
* 3.27 前后端交互设计（董乐辉版）
* 3.28 风险评估
* 4-1 七周
* 实施工程
* 4-2 完成 登陆页面（二版）
* 4-3 完成 登陆页面（三版）
* 4-3 完成web首页（二版）

？ 完成 表单页，（二版）

？员工表单 的表格制作（二版）

* 4-8 八周
* 实施工程：
* 计划于4-8 完成 前端数据封装
* 计划于4-9 完成 前端数据传输逻辑
* 计划于4-12完成 前端测试
* 计划于4-14完成 前后端对接
* 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：

4-15 九周

* 计划于4-17 完成 登陆页面并增加其跳转（三版）
* 计划于4-19 完成 用户评论提问功能
* 计划于4-21完成 员工端对保单的查询，修改，删除功能

4-22 十周

* 计划于4-25 完成 双语功能
* 计划于4-27 完成 通过QQ和github登陆功能
* 计划于4-28完成 前端新功能的测试

4-29 十一周

* 计划于5-1 进行手机端适配
* 计划于5-4完成手机端适配

5-6 十二周

* 计划于 5-6 完成全部功能于页面的测试，并进行交付

**（后端自行整理 (\*^\_^\*)）**