时间：2020.02.25

周次：第一周

地点：线上

1. 指导内容：
2. 确定了每周线上会议的时间，暂定为每周二下午三点半。
3. 学习了腾讯会议的基本用法，能够实现与导师顺利交流、分享屏幕等功能。
4. 根据开题时所确定的目标，将剩余工作分为五个模块，分别为：公司搜索、融报管理、商机管理、用户管理以及PDF生成。预计划为一周实现一个模块的部分，五周后开始做最后的调整和论文的撰写。
5. 已经实现的功能：

由于项目刚进入启动阶段，所以已完成的任务较少。

已经将编程环境搭建完毕并能够正常运行。

已经通过网课学习了开发语言的基本语法以及开发IDE的基本操作。

1. 遇到的困难：

由于是新接触的编程语言以及IDE工具，还需要时间磨合，包括一些常见的函数及用法、快捷键、debug模式等等。

1. 本周任务：
2. 完成iOS开发所需环境以及IDE的安装，确保项目的有序进行
3. 实现公司搜索模块的功能，其中包括：
   1. 输入框输入关键字，获取输入的值，将其传入服务器获取返回的公司数据
   2. 以列表的方式显示公司的名称及信息
   3. 为缓解数据负担，采用每次获取10个数据的方法获取数据，当拉到底部时再刷新新的10个数据
   4. 能够在搜索结果列表中对公司进行收藏和取消收藏操作

学生签字：王时予

本周总结：

任务1：已经完成。

任务2：大部分内容已经完成并能够成功运行，由于4)中的收藏和取消收藏的功能需要在实现用户管理模块后与之配合进行，所以不能全部完成，但已经写好了可通调用的接口，用户管理模块完成后只需调用其接口便可实现该功能。

在本周中遇到的困难及解决办法：

1. 由于没有学习过swift语言，所以在实现http异步请求时耗费了很多的时间，使用封装好的第三方库无法将获取到的JSON数据直接转化为Model类的对象，而将数据进行一一对应又是工作量极大的。最后我使用Alamofire包中最基本的request请求获取到了JSON数据，不将其实例化为对象，而使用SwiftJSON直接对其进行解析，也实现了所需功能。
2. 此软件开发需使用MVC模型依据代码功能对代码进行划分，也是由于第一次使用swift语言，所以整个模型框架需要在学习中搭建。最后使用protocol协议的方式完成了搭建，这一部分的模型搭建完全由自己研究得来，没有做任何的参考。
3. xCode中提供了NavigationBar作为基础导航栏，但并未提供对其各项参数的设置方法，于是我在网上查阅了相关的资料，最终实现了设置导航栏参数的方法。（查阅网址:stackoverflow.com以及苹果官方的开发者文档developer.apple.com）

由于本周的任务量相对较少，我做的也算较快，所以在周六就已经完成了所要求的任务，于是提前开始了下一个周期的任务，即完成融报管理模块功能的实现，截至撰写手册时，已经完成了融资报告首页的部分内容。

时间：2020.03.03

周次：第二周

地点：线上

1. 指导内容
2. 检查了上周任务的完成情况，主要针对有不同情况的公司的适应情况。
3. 讨论了上周遇到的问题以及解决情况。
4. 确定了下周的任务目标，经商讨后，觉得融资报告模块的任务较为繁重，其中还包括了表格设计、饼状图、折线图等的部分，将其的完成时间定在两到三周后。
5. 讨论了现在过程中正在遇到的问题，即无法给嵌套于View中的tableView添加数据，在讨论过程中进行了数种尝试，但还是没有解决。
6. 已经实现的功能

搜索公司：能够在输入框中输入公司的关键字并根据关键字在数据库中进行检索，将返回的数据在列表中显示，单击每个单独的公司能够进入公司主页，单击主页中的融报按钮，能够显示该公司的融资报告首页，包括公司基本情况、股东背景、历史沿革、高管信息的摘要

1. 遇到的困难

iOS中的界面设计与安卓有所不同，使用的tableView控件需要重新学习，该控件需要使用tableViewController类接收才能够设置其中的数据，然而嵌套在view中的tableView并无此功能，需要找到解决办法显示嵌套在view中的tableView显示数据的问题

1. 本周任务
2. 解决tableView嵌套时的数据显示问题
3. 完成融资报告首页其余的摘要显示内容
4. 完成公司基本情况详情的页面
5. 完成股东背景的详情页面
   1. 完成饼状图的实现
6. 完成历史沿革详情页面
7. 完成高管信息详情页面、
8. 尝试完成生产经营情况详情页面

学生签字：王时予

本周总结：

任务1：通过其他方式解决

任务2：除首页中融资情况摘要还需对数据进行排序外，其余均已完成

任务3：完成

任务4：已经完成，饼状图使用了与Android端相同的MPChrat团队开发的Charts包完成

任务5：完成

任务6：完成

任务7：已基本完成，其中的水平滑动表格部分已经实现所需效果，但还一些细节需要后期修缮

在本周遇到的困难及解决办法：

1. swift中的ScrollView与android中的ScrollView有些不同，设计过程中会出现无法全部显示View中的内容的情况。通过查阅多篇文献，最终使用某种方法解决了该问题，方法的主要内容是通过修改contentView的大小将ScrollView撑开，使其能够知道需要显示的内容的大小。
2. 饼状图的制作：查询到了许多用于显示饼状图的第三方库，但很多都没有所需要的样式和足够明确的说明文档，后来通过在github上找寻曾经在android上使用过的MPChart库的作者，一路关联搜索到了他们团队开发的用于swift的库，使用了几乎相同的方法就完成了该功能的实现。
3. 滑动表格的设计：耗费了大部分的时间完成这个功能，尝试了很多方法，但由于StackView的布局方式与android中的LinearLayout有着极大的区别，并且无法直接修改每个控件的宽高，所以需要在生成控件对象时就对其宽高进行规定，并在插入至StackView后也需要对StackView进行宽高的设定，这就变得有些麻烦了。并且，由于swift中的viewDidLoad函数执行过后，就无法通过改变frame的方式改变控件的大小，所以需要在整个华滑动表格全部创建完成后再将其添加至ViewController，并如第一条中所说再次修改ScrollView中的content才能够完全将其显示。
4. 动态表格的设计相比滑动表格相对简单一些，不过设置每列表格的宽度权重以及设置表格间隔线颜色的功能还没有实现，还在进一步研究中。