中山大学 SUN YAT-SEN UNIVERSITY 现代操作系统期中项目实验报告

一、 项目分工

学号	名字	角色	班级	职责	贡献
16340157	刘亚辉	组长	上午班	负责前端框架，页面逻辑设计，文件读写	25%
16340155	刘伟东	组员	上午班	服务器和数据库	25%
16340156	刘笑	组员	上午班	UI 设计与页面动态交互，部分的页面逻辑	25%
163.40158	刘宇庭	组员	上午班	客户端与网络数据同步与数据解析，应用间交互，动态磁贴	25%

二、开发环境

Windows10, visual studio2017, node, mongodb 数据库

三、 项目阐述

项目名称： Competition

项目简介:

一款创建网球，乒乓球，羽毛球等球类比赛和管理比赛于一体的 UWP。

项目功能：

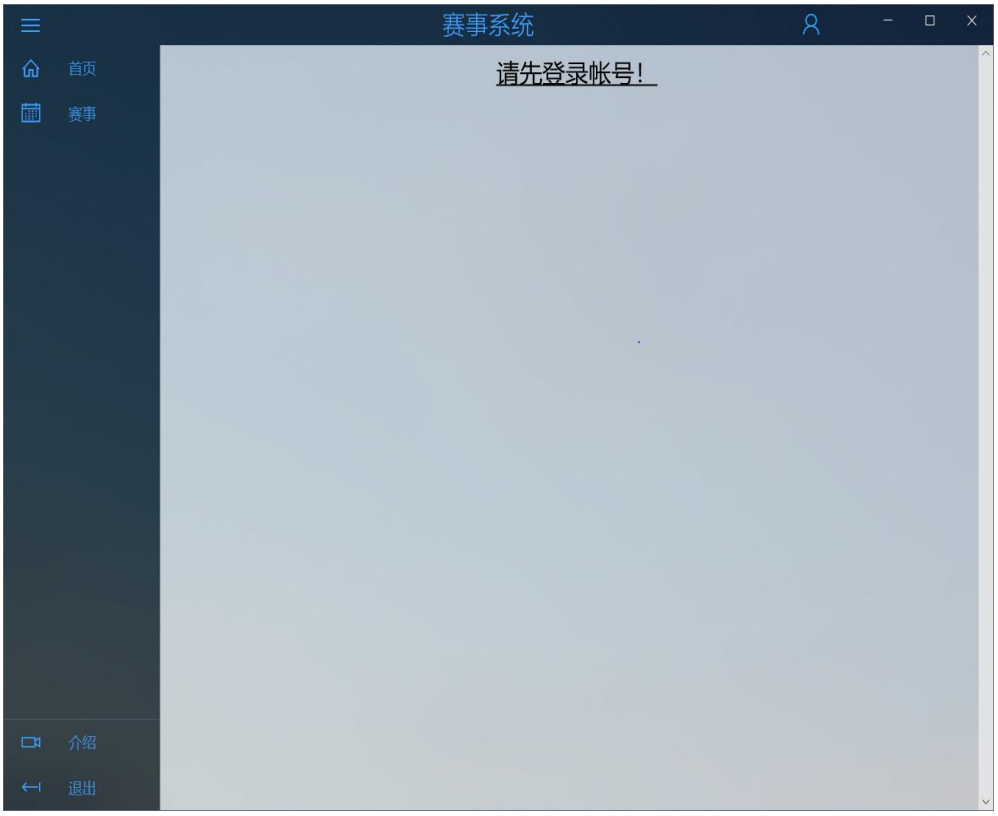
根据用户上传的运动员信息表格，解析 excel 得到其中的运动员信息，再依据用户设置的比赛内容，生成相应的赛事并对其进行管理。

项目亮点：

客户端与服务器中相对应的数据库同步，同时网络访问使用 cookie 保存用户登录状态，支持多用户，支持多种比赛，支持多种赛制

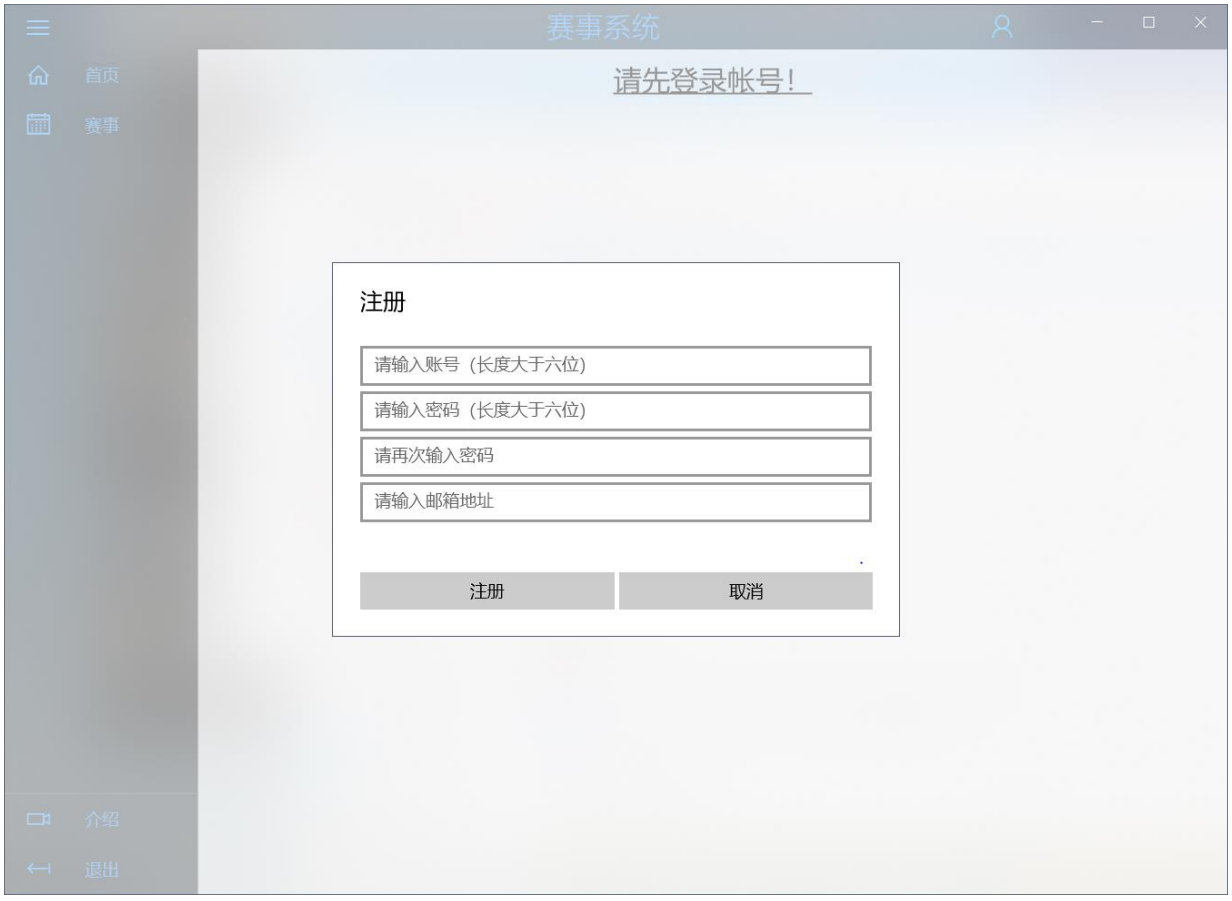
四、 项目展示

1. 进入界面：

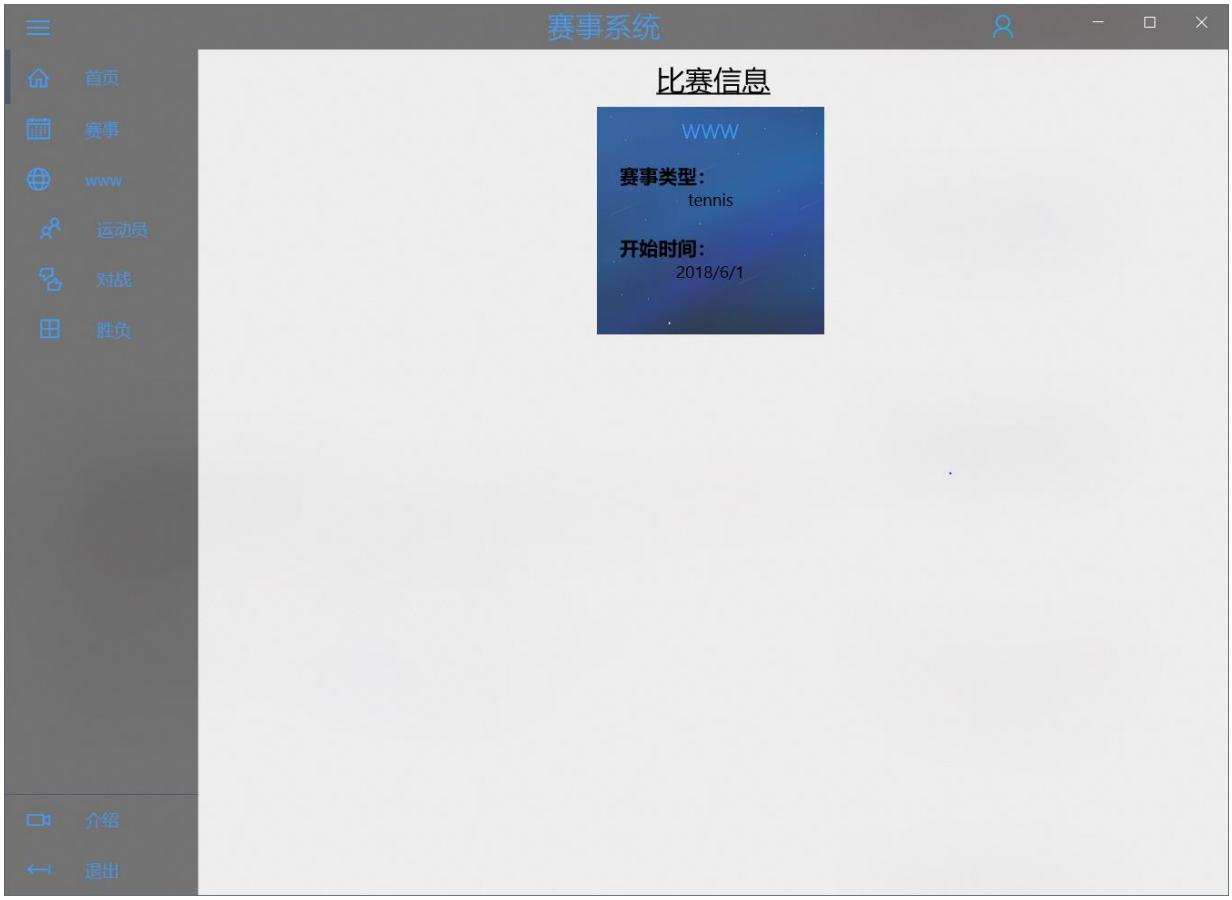


2. 登陆界面及注册界面：


赛事系统
首页
赛事
请先登录帐号!
登陆
退出
登陆
请输入用户名
请输入密码
登陆
注册
介绍
退出


赛事系统
首页
赛事
请先登录帐号!
注册
请输入账号（长度大于六位）
请输入密码（长度大于六位）
请再次输入密码
请输入邮箱地址
注册
取消
介绍
退出

3. 登陆后界面：



五、 项目难点及解决方案

在项目开始之前，对项目的总体规划十分重要：我们主要将整个 UWP 分为四个部分，第一部分，主要是项目的整体架构与逻辑设计，第二部分，主要是 UI 的设计与动态交互，第三部分我们利用 Nodejs 来搭建服务器，数据库；第四部分，构建网络访问模块进行数据获取与同步。

① 整体架构与逻辑设计:

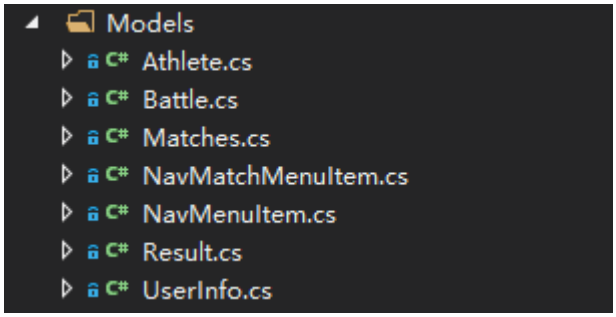
1. 对整体项目进行页面规划与简单设计：

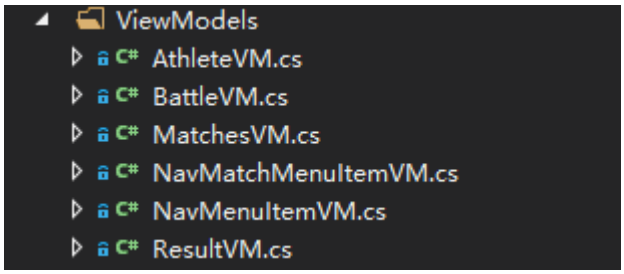
在此部分，需要统筹所有的页面，并实现合理的页面交互：

MainPage 中主要实现的是菜单栏，在其后台处理文件中处理页面交互过程。

项目整体页面全部存放于 Views 文件加下：包括主界面(Home)，创建比赛的界面(Match)，用户注册界面(RegistDialog)，视频界面(Video)，然后创建比赛完成之后会显示比赛的相关信息，全部存放于 MatchInfo 文件夹中，包括赛事详情界面(MatchCreated)，运动员信息界面(Athletes)，对战信息界面(Battles)，对战结果信息界面(Results)。

2. 利用 MVVM 结构：数据处理 ViewModel 模块将数据模块 Model 与视图模块 View 联系起来，界面设计完成之后主要是数据与界面之间的联系：

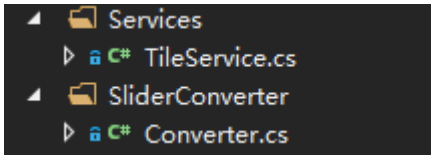


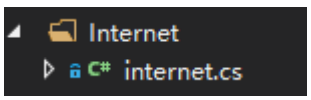


Models 文件夹中，主要是运动员数据模板(Athlete)，对战数据模板(Battle)，结果数据模板(Result)，比赛数据模板(Maatches)，用互信息模板(UserInfo)，以及菜单按钮模板(NavMenuItem 和 NavMatchMenuItem)。ViewModels 文件夹中存放是各个数据模板中的处理函数，并存放各种信息。

利用以上的 MVVM 框架，可以使得项目结构更加清晰，有利于开发和维护。

3.一些其他模块的构建：





TitleService 模块用于构建动态磁贴，Converter 模块用于 Vedio 播放的双向绑定转化模块。internet 是网络访问模块，构造 API 用于实现应用与服务器网络交互。

4. 在进行逻辑设计的过程中，遇到的问题主要有 Excel 表的导入与导出，主要是利用第三方库 ExcelDataReader 和 EPPlus 库来实现，主要涉及到文件访问权限问题，利用 FileOpenPicker 和 FileSavePicker 来实现指定文件获取与保存。

② UI 设计与动态交互：

a) UI 分为边缘 Livetitle 部分，Flyout 和 ContentDialog 浮出页面部分和主要显示页面部分。

i. Livetitle 根据页面宽度自动调整显示的宽度，也可以通过点击按键来调整是否完全显示出来。然后自定义 TitleBar，将自定义的元素浮到顶部并且将文字部分定义为 TitleBar。

ii. 显示页面分为比赛信息主页面，运动员信息页面，比赛信息页面和比赛结果调整页面。其中每个页面均根据页面宽度自动或者通过 VisualStateManager 调整显示内容以适应页面。

iii. 浮出页面部分分为登陆页面，分享按钮，修改信息页面和注册页面，后台对应相应功能。

b) 动态交互主要是用简单的操作实现实用的功能。在比赛信息页面可以右键点击比赛来分享比赛基本信息。运动员信息界面直接点击运动员可以在弹出的窗口修改运动员信息。点击后的跳转以及焦点的转移也方便了直观显示。

③ 数据库与服务器部分：

服务端：node + express 框架 + MongoDB 数据库

难点 1：数据库的模型结构，如何构建模型能够使得所有要求都能够完整的实现。

解决方案: mongodb 中存在 ref 和 populate 关键字，合理规划模型之后可以使用这两个关键字保证内容同步。

难点 2：路由的规划，该项目的网络 API 应该怎样设计代码的重用性较高。

解决方案: 利用 empress 的路由参数（形如”/get/:type”），规划路由。然后使用 json 下标访问动态调用相应的数据集合，进行增删改查操作。

难点 3：数组循环异步调用，如何同步的问题。

解决方案，使用回调函数，增加计数器，在回调函数中判断计数器是否等于数组长度。

难点 4：比赛赛制的实现，如何生成与实际相符的对战信息。

解决方案：查看相关比赛的赛制划分方案，将其转化成相关函数。

难点 5：给注册成功的用户发送欢迎邮件，如何实现。

解决方案：安装 nodemailer 和 nodemailer-smtp-transport 的 node 包。

难点 6：客户端与服务端的数据交互

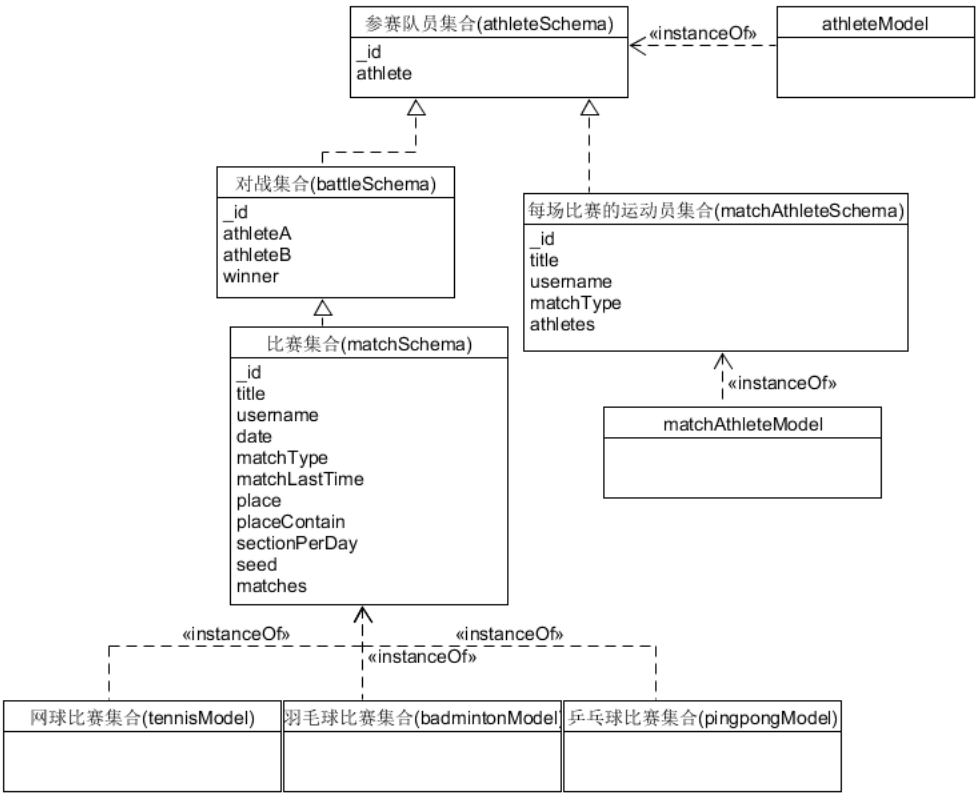
解决方案: 规定简易的正确信息模板，和固定的错误信息模板。

难点 7：保持用户的登录状态

解决方案: 使用数据库储存 session。

难点 8：用户密码加密

解决方案: 使用 crypto 包加密，采用其中的哈希加密方法。



④ 客户端与服务器交互以及应用间的交互

1. **网络访问部分：**使用 httplclient 模拟浏览器访问进行与数据库的交互，客户端的数据是持久化的，客户端通过 post 或者 get 请求与数据库的信息进行同步，修改信息的时候也是通过 get 或者 post 请求，修改数据库的信息成功后才会修改本地的信息。请求得到的信息都为 json 格式，需要对信息解析以后再获得，同时用户登陆后会有一个 cookie，以后的再次数据访问无需再次上传用户信息，httpclient 自动保存 cookie。
2. **右键点击分享比赛：**当成功创建一场比赛后，在首页可以看到创建的比赛的，双击或者左键进入比赛的相关操作页面，右键比赛下方将弹出分享按钮，点击分享按钮，将启动系统的分享功能，通过邮件分享比赛的标题，开始时间以及比赛的类型。
3. **动态磁贴：**循环显示已经创建的比赛，包含比赛的标题，开始时间以及比赛的类型。

六、 项目总结

该应用使用了以下技术:

(1) Adaptive UI ： 自适应的边栏和主页面显示样式。
(2) Data Binding ： 使用 GirdView 和 ListView 展示信息均大量使用了数据绑定。
(3) Database ： 云端 Mongodb 数据库。
(4) App to app communication：右键点击具体比赛，可以通过其他应用分享信息。
(5) Network accessing ： 服务器架设与腾讯云，通过与服务器交互来同步数据。
(6) File management ： 拖拽或者选取上传的文件，通过按钮到处比赛 EXCEL 表格。
(7) Live tiles：自动适应的边栏。
(8)媒体应用：播放教学演示视频。

项目总结：

1. 整体使用 MVVM 设计模式，部分模块使用单例模式
2. 打开应用时点击右上角的账户图标，可以登录获取以往创建过的比赛信息；在登录右下角可以注册新的用户
3. 点击对应的比赛将通过网络获取相应的数据，通过 Data Binding 展示出来。
4. 使用 node+express+mongodb，数据库建立在服务器上
5. 在首页右键相应的比赛可以将它通过邮件分享给他人
6. 该应用通过读取 excel 文件获取相关的比赛信息，但仅数据部分可以修改，excel 的表头内容中身份证信息是区分运动员是否唯一的关键，删除后整个应用可能会出现错误。
7. 动态磁贴实现小中宽三种大小，能够滚动显示已经创建的比赛的简要信息。
8. 在应用左下角有一个介绍，可以通过点击它查看更详细的使用方法。
9. 在本次项目中，最困难的地方在于服务器与本地客户端如何进行交互与网络 API 获取的数据和前端数据格式转换问题，主要事先没有进行完整规划，客户端和服务端没有事先定制好相应的交互规则。因此，有一个好的架构师非常重要!!!
10. 本次项目是前半学期学习的综合内容。为了实现这个项目，服务端的内容是额外增加的，因为它采用的是 javascript，不在之前的学习内容之中。