

软件设计文档

项目成员	
姓名	学号
赖加成	14331122
黄荣洲	14331102
宁晓亮	14331213
林子凌	14331169
占凯文	14331353
吴子昊	14331295

后端：

后端技术选型：语言 node.js、框架 Koa.js、数据库 mongoDB

理由：node.js 是新兴的一种后端开发技术，借助于活跃的 JavaScript 社区。node.js 的开发有极其丰富的资源。而极具特色和灵活性的 JavaScript 语言也给这门技术增添了很多魅力。node.js 本身异步非阻塞的特点，使得它非常适合做服务端的开发。能够轻松做到非常大的 IO 吞吐量。这就是我们选择 node.js 的原因。

而后端开发框架 Koa.js 是 node.js 代表作品 express 框架的一个进化版。它的设计理念是嵌套式的中间件。同时推荐使用 async/await 这个新的语法特性，使得 node.js 异步编程更加简单。mongoDB 数据库的选择是因为我们这一产品的各个 model 之间的对应关系不是很复杂，而 mongoDB 灵活的特点，非常适合我们这个项目。

模块划分：后端主要分为 service, controller, model 三块。Model 负责定义数据库模型。Service 提供 model 之上的操作。Controller 负责路由和请求的处理。

前端：

前端技术选型：语言 C#、框架 MVVM、数据库 SQLite

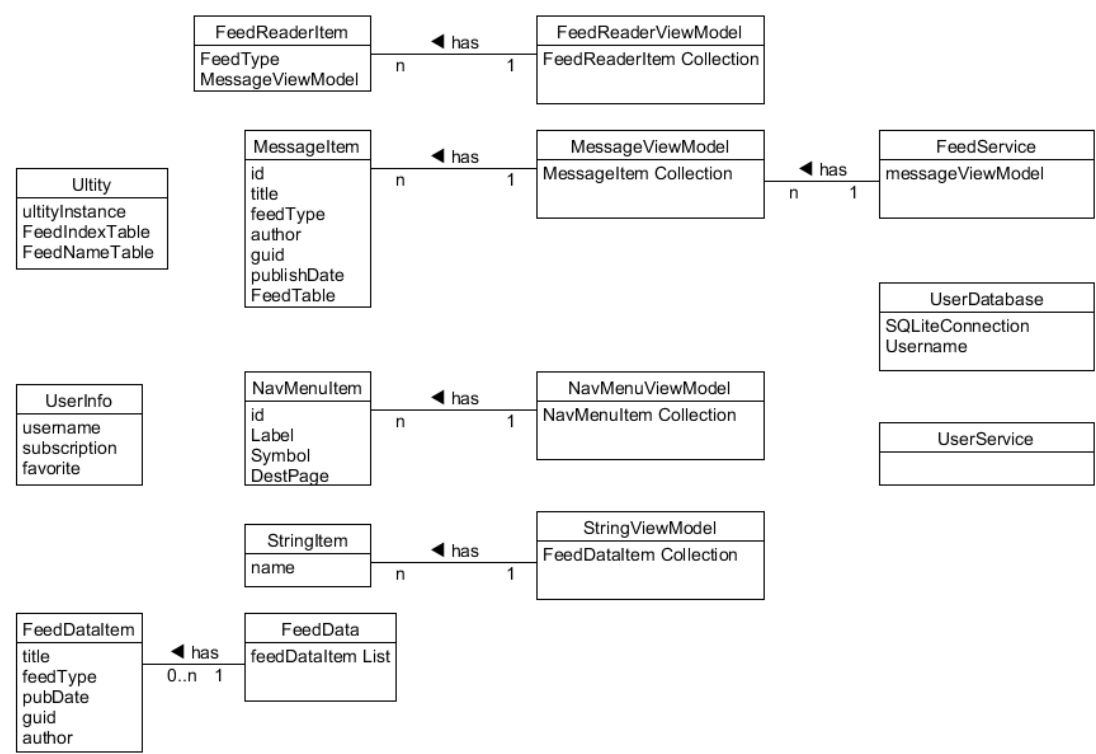
理由：UWP 应用开发是基于 C#、xaml 编写的。由于 WPF 拥有丰富的数据绑定能力，因此 MVVM 模型能够提供一个纯净的 Model，创建了一个抽象的视图来容纳状态，并且数据从一个由可视化设计器创建的视图绑定到一个抽象的视图上。该可视化设计器主要是 Microsoft Expression Blend 工具，而这个抽象的视图则是 ViewModel。关键的地方是通过数据绑定在 View 与 ViewModel 模型之间进行双向的联系。当正确地使用该设计模式后，每一个 View 除了纯净的 XAML 和非常

少量的后置代码外不会再包含任何东西，彻底地做到了界面展示和业务逻辑的分离，让程序员更加专注于代码的编写。

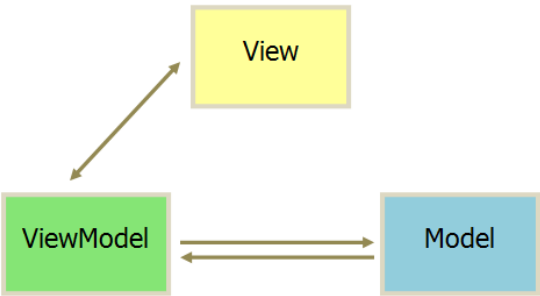
模块划分：主要分为 Models, ViewModels, View, Services 四块。Models 负责程序需要的各种数据结构，ViewModels 负责 Models 与 View 的沟通。Services 负责程序的网路访问，本地数据库的建立和操作。

附录：模型

领域图（Domain Model）：



架构图：



逻辑视图:

