

## 1. 技术选型表

项目	Web App	备注
1.终端支持	PC	开发 web app 先支持 PC，先完成最简单的功能
1.1 开发语言框架	HTML5 CSS3 JavaScript	采用比较流行的前端开发技术（由前端开发小组负责）
2 服务器支持	云端建立服务器	由后端开发小组负责
2.1 语言	PHP	PHP 简单易学，由后端程序员负责
2.2 Web 框架	ThinkPHP	由云端开发人员负责
2.3 关系数据库	MySQL	MySQL 由数据库管理员负责
2.4 数据缓存（非关系）	NoSQL	NoSQL 完成视情况而定，由后端人员负责
2.5 消息中间件	RabbitMQ	视情况而定，由后端开发人员与前端开发人员进度决定分配情况
3 开发平台与开发工具	Sublime	组员自行安装
3.2 集成与测试	黑盒测试，白盒测试，客户使用评价	由测试人员负责
3.3 源代码管理	Github	各组员自行安装

## 2. 技术原型开发内容

本项目技术风险元素：

（1）后台与前端的交互。在这方面，各个小组成员虽然都有开发前端技术的经验，但是对于搭建后台还没有实际的操作和项目经验。而且，在项目过程中，前端与后台的小组分离，虽然能够达到加快项目开发的作用，但风险就是最后前端与后台不能够协调工作，这是本项目最大的隐患。

（2）本项目涉及多项前端与后台开发技术，最后后台采取了 PHP 语言进行开发，能够降低项目难度，同时也能够便于各个后端小组成员进行快速学习与开发。在项目开发过程中，PHP 技术大家都没有用过，而且 PHP 对于高并发情况十分糟糕，因此，其风险在于网站的规模一旦扩大，就会造成网站崩溃的危险。

（3）由于是开发类似于知乎的社交网页，因此，在网页中存在着大量非结构化或者半结构化的信息。这对于采取关系型数据库来说是一个很大的挑战，存在着无法存储数据的风险。

验证性程序开发方案或技术原理：

（1）针对后台与前端的交互问题，可采取先易而难，增量迭代开发原则。前端先添加注册登录功能，后端先完成注册与登陆的数据库，然后协调前端与后端进行最简单的交互，用以验证前端与后端交互的可行性，降低甚至消除项目最后不能够交互的风险。

（2）对于网站的开发过程，高性能是一个非常具有挑战性的问题，对于后台开发人员来说，高并发始终是一个难题。目前来说，对于前端开发，要进行足够的代码优化，以确保后台的数据量访问不会出现问题。同时，后台也要采取加速技术，例如反向代理服务器（MemCache），分布式数据库，负载均衡服务器等。

(3) 针对关系型数据库的问题，要对数据进行结构化处理。同时数据库管理员要理清各个数据库之间的关系，同时采取在非结构化数据中添加 TAG，对数据进行必要的分离等手段，使数据能够存储于关系型数据库。