# 1引言

## 编写目的

为了便于协调组内成员进行后期的工作，对项目进行跟踪和监控，对任务的进度进行安排与调控，故对后期进行计划。提出需求，指导后续工作。

# 系统概述

提及java web开发肯定就要谈Spring，要谈Spring的历史，就要先谈J2EE。J2EE应用程序的广泛实现是在1999年和2000年开始的，它的出现带来了诸如事务管理之类的核心中间层概念的标准化，但是在实践中并没有获得绝对的成功，因为开发效率，开发难度和实际的性能都令人失望。而Spring出现的初衷就是为了解决类似的这些问题。但是Spring是企业级的，所以更关注于需求，扩展了越来越多的功能，衍生出许多其他的分支项目，导致框架越来越大，也越来越杂。所以，我们小组最后选择了Blade框架进行分析，该框架更专注于Java Web的框架，并且它的开发更新也较为稳定。

Blade是一个简介强大的web框架，它内置了IOC管理，拦截器配置，REST API开发等众多主流web特性，集成了模板引擎，缓存插件，数据库操作，邮件发送，HTTP请求等常用功能。它的主要特点如下：

» 0配置

» 轻量级。不依赖于更多的库，摆脱SSH的臃肿，模块化设计，使用起来更轻便

» Restful风格的路由接口

» 单jar运行易于部署

## 文档概述

文档用途：本文档主要是介绍blade框架需求和UML模型

主要内容：

1. 以用例图、状态图的形式给出blade框架更能需求的分解结构，并对用例模型中的参与者和用例进行详细的描述，其中主要包括软件系统的用例模型、时序图、状态图以及框架的核心流程等；
2. 使用RUCM模型、时序图、状态图对功能需求进行建模。

## 术语和缩略语

1. MVC：全名是Model View Controller，即模型—视图—控制器的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。
2. IoC：Inversion of control的缩写，意为控制反转，是一种设计思想，将设计好的对象交给容器控制，而不是传统的在对象内部直接控制。
3. DI：Dependency Injection，即“依赖注入”： 组件之间依赖关系由容器在运行期决定，形象的说，即由容器动态的将某个依赖关系注入到组件之中。
4. 路由：在blade中，路由是一个HTTP方法配对一个URL匹配模型，每一个路由可以应对一个处理方法。
5. 拦截器：blade中的拦截器是在一个请求执行前、后可以做一些自定义的处理，比如存储数据，校验数据，过滤请求等。
6. HTML：超文本标记语言
7. RESTful架构：Representational State Transfer，一种软件架构风格，提供了一组设计原则和约束条件。它主要用于客户端和服务器交互类的软件。基于这个风格设计的软件可以更简洁，更有层次，更易于实现缓存等机制。