**基本要求：搭建DevOps开发、测试和部署环境，并在此基础上完成风险管理系统的开发和部署应用。**

需求描述（第一阶段）

1. 开发一套软件项目风险管理系统；
2. 支持用户和角色；
3. 支持风险条目输入，包括风险内容、可能性（高中低）、影响程度（高中低）、触发器/阈值、提交者、跟踪者等；
4. 支持风险状态跟踪，包括状态描述（风险？问题？）和相应的文本形式的描述信息；

关于需求中的风险条目和风险状态：

在提交者提交一个风险条目的时候，风险条目有了一个初始状态。然后跟踪者可以进行跟踪和更新。提交者可以进行跟踪。(就是跟踪者可以看到风险条目的状态，也可以进行修改，提交者只可以看到状态，不能修改)

另外，风险状态是一直更新的，所以要有Timeline的显示。(比如说，2016年11月6日，cyl更新了这条状态，状态描述是“\*\*\*\*\*\*\*\*”; 2016年11月9日，cyl又更新了这条状态，状态描述是“\*\*\*\*\*\*\*\*”)

接口：

1. 注册(用户名，密码，角色)

【角色用下拉列表给出选择】

**public** User register(String username,String password,String role);

成功则返回user，失败则返回null

1. 登录(用户名，密码，角色)

【角色用下拉列表给出选择】

**public** User login(String username,String password,String role);

成功则返回user，失败则返回null

3. 风险条目输入(RiskItem包括风险内容、可能性（高中低）、影响程度（高中低）、触发器/阈值、提交者、跟踪者)

【注意，提交者和跟踪者的选择，最好用下拉框列出所有用户以供选择】

public boolean add(RiskItem risk);

4. 提交者查看自己提交的全部风险条目

public List<RiskItem> getSubmitItem(int userId);

5. 提交者查看自己提交的某条风险条目状态

public List<RiskState> getSubmitState(int userId,int riskId);

6. 跟踪者查看自己跟踪的全部风险条目

public List<RiskItem> getFollowItem(int userId);

7. 跟踪者查看自己跟踪的某条风险条目状态

public List<RiskState> getFollowState(int userId,int riskId);

8. 跟踪者更新自己跟踪的某条风险条目的状态

【不是在原有状态上更改，而是新增一条状态】

public boolean addState(RiskState state);