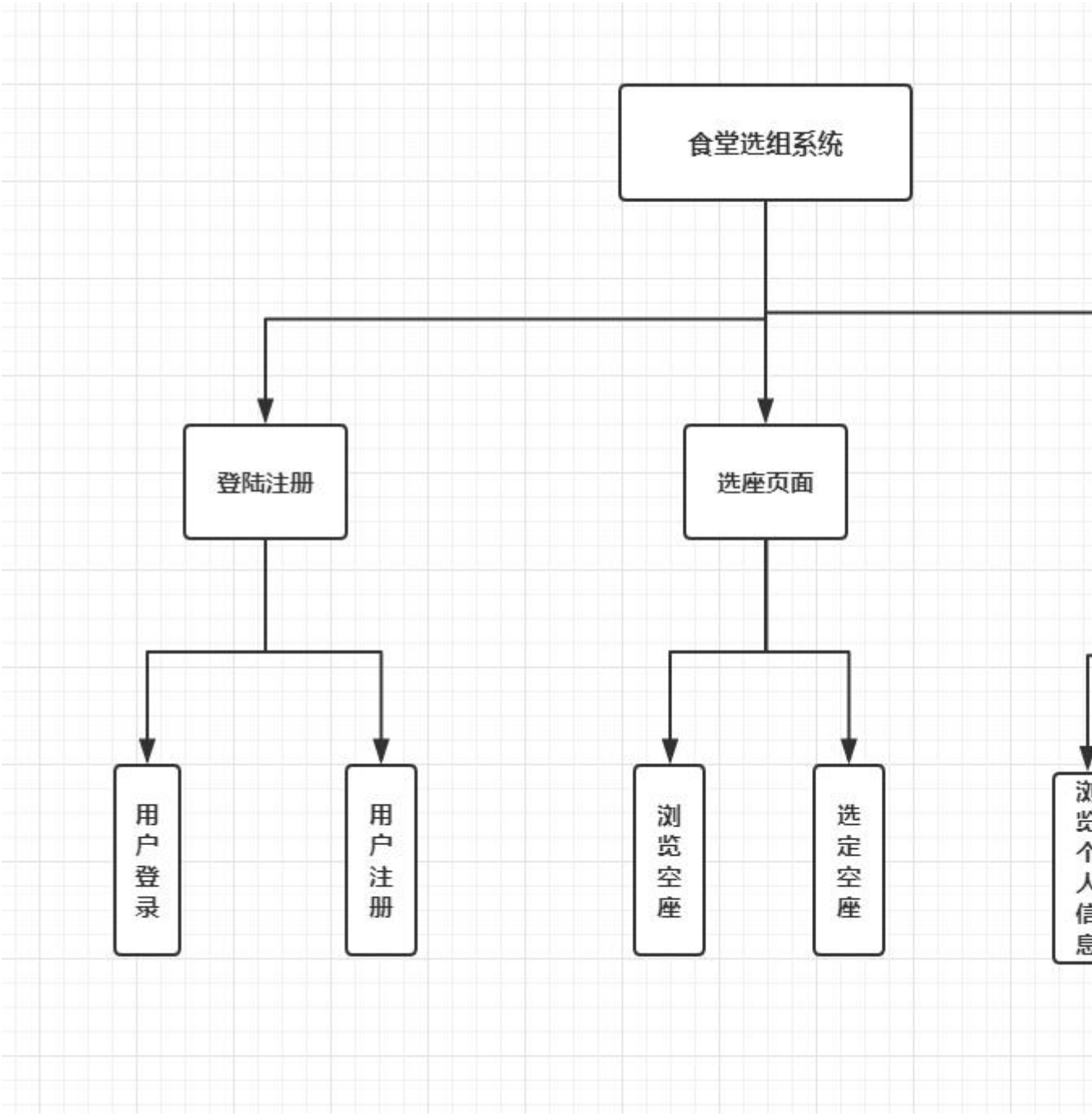
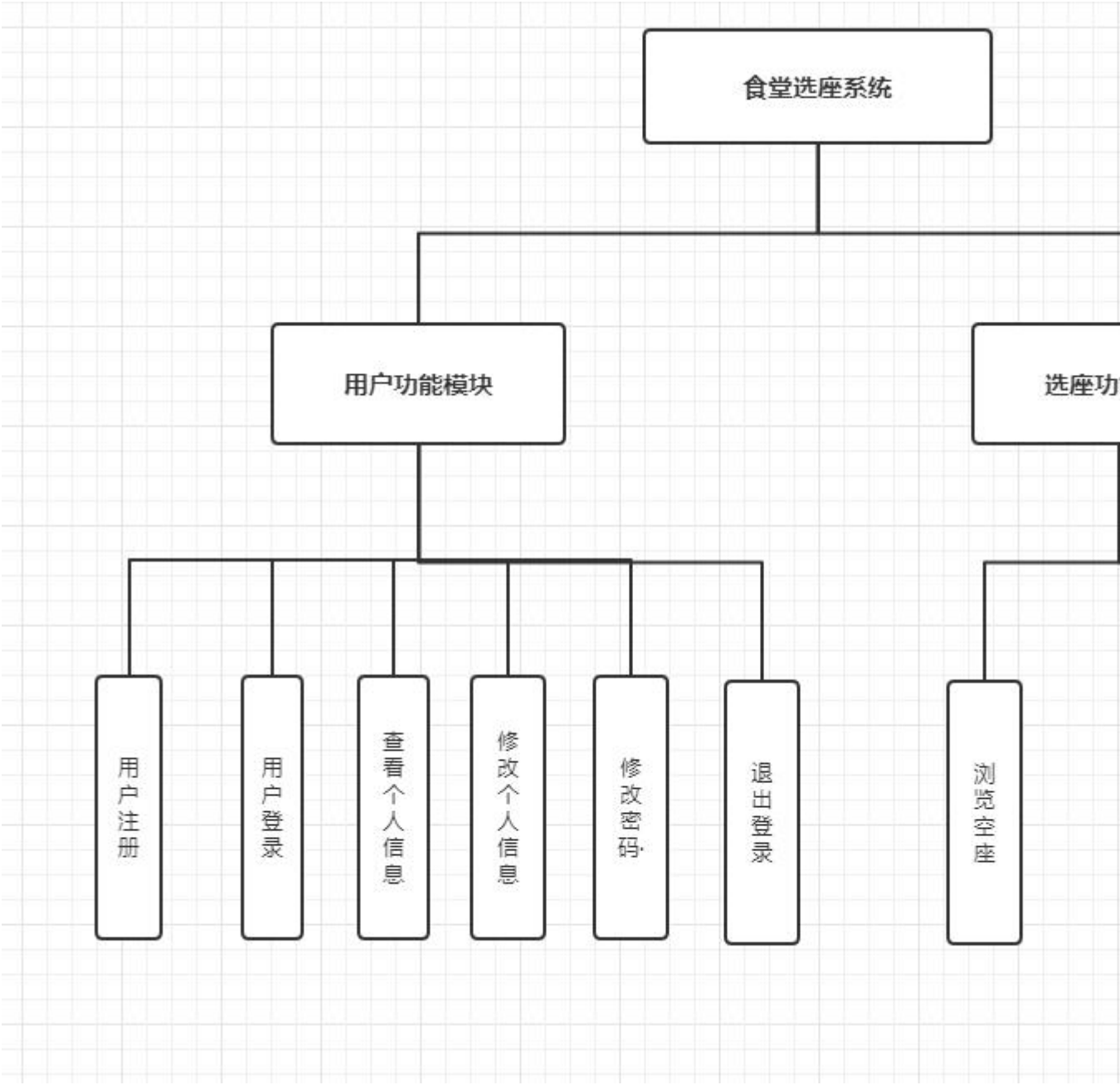
《系统设计说明书》

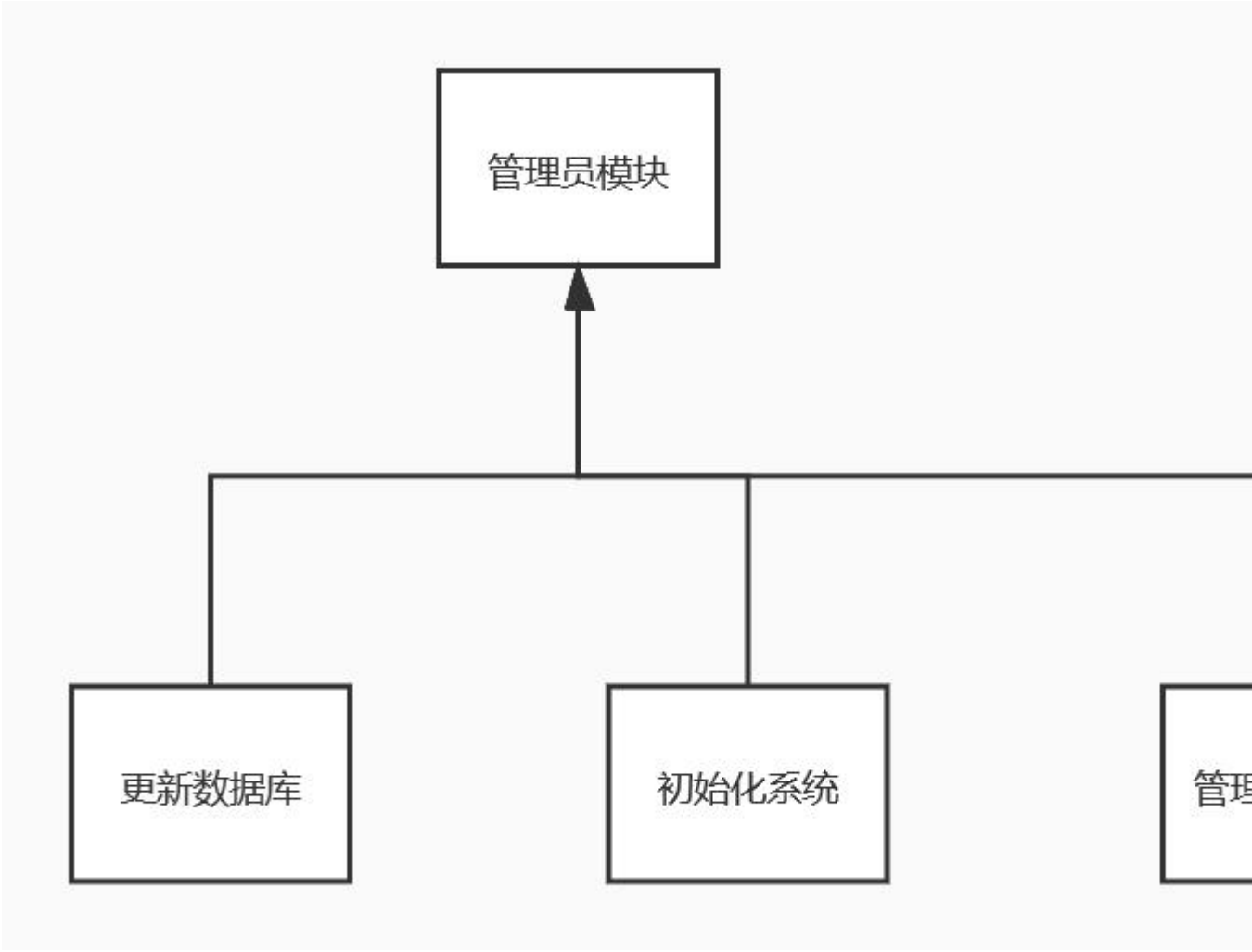
体系结构设计+功能模块层次图、设计类图、ER 分析+表结构设计、系统安全和权限设计

体系结构设计图

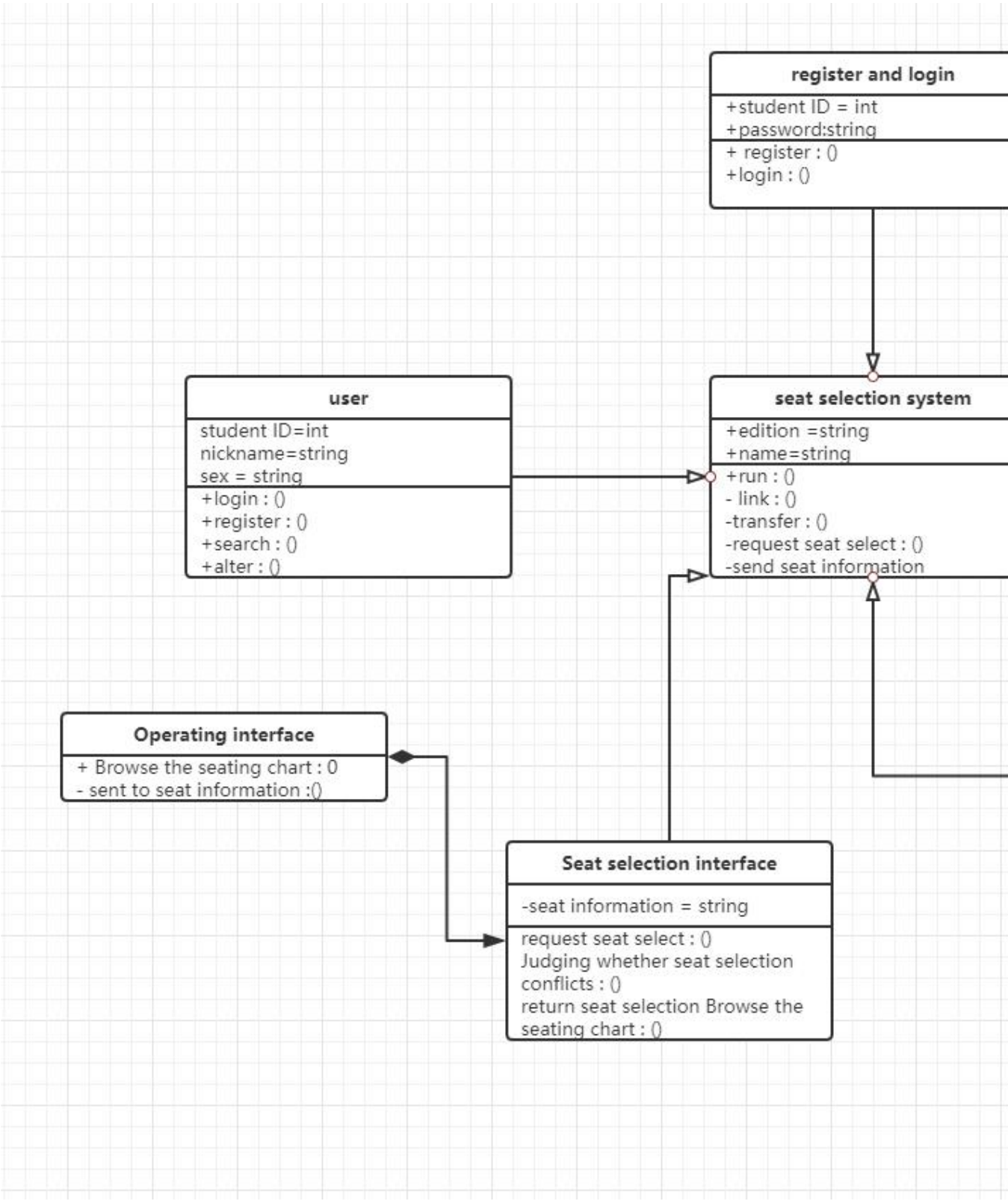


功能模块图





设计类图


register and login
+student ID = int
+password:string
+ register : ()
+login : ()
user
student ID=int
nickname=string
sex = string
+login : ()
+register : ()
+search : ()
+alter : ()
seat selection system
+edition =string
+name=string
+run : ()
- link : ()
-transfer : ()
-request seat select : ()
-send seat information
Operating interface
+ Browse the seating chart : 0
- sent to seat information :()
Seat selection interface
-seat information = string
request seat select : ()
Judging whether seat selection conflicts : ()
return seat selection Browse the seating chart : ()

系统安全和权限设计

常见 web 攻击方法及防御手段总结

1. CSRF （cross-site request forgery）跨站请求伪造

预防

之所以被攻击是因为攻击者利用了存储在浏览器用于用户认证的 cookie，那么如果我们不用 cookie 来验证不就可以预防了。所以我们可以采用 token（不存储于浏览器）认证，为每一个提交的表单生成一个随机 token， 存储在 session 中，每次验证表单 token，检查 token 是否正确。。

通过 referer 识别，HTTP Referer 是 header 的一部分，当浏览器向 web 服务器发送请求的时候，一般会带上 Referer，告诉服务器我是从哪个页面链接过来的，服务器基此可以获得一些信息用于处理。那么这样的话，我们必须登录银行 A 网站才能进行转账了。

2. XSS （cross site script）跨站脚本攻击

预防

- 转移和过滤用户提交的信息，将输入的数据进行转义处理，比如说讲 < 转义成&lt

3. session 攻击，会话劫持

防御方法：

- 每次登陆重置 sessionID

- 设置 HTTPOnly，防止客户端脚本访问 cookie 信息，阻止 xss 攻击
- 关闭透明化 sessionID
- user-agent 头信息验证
- token 校验

4.SQL 注入

预防

在 java 中，我们可以使用预编译语句(PreparedStatement)，这样的话即使我们使用 sql 语句伪造成参数，到了服务端的时候，这个伪造 sql 语句的参数也只是简单的字符，并不能起到攻击的作用。

很多 orm 框架已经可以对参数进行转义

做最坏的打算，即使被’拖库‘('脱裤，数据库泄露')。数据库中密码不应明文存储的，可以对密码使用 md5 进行加密，为了加大破解成本，所以可以采用加盐的（数据库存储用户名，盐（随机字符长），md5 后的密文）方式

5.DDOS

预防

- 最直接的方法增加带宽。但是攻击者用各地的电脑进行攻击，他的带宽不会耗费很多钱，但对于服务器来说，带宽非常昂贵。
- 云服务提供商有自己的一套完整 DDoS 解决方案，并且能提供丰富的带宽资源

用户注册

1. 得到用户传过来的密码后，首先在计算机中获取一个随机数，

2. 获取到随机数后，设计一个任意算法，对随机数与用户密码进行拼接处理，比如最简单的（用户密码+随机数），者将得到一个全新的字符串

3. 我们再对这个新的字符串进行哈希算法处理，得到一个新的密码，由于哈希算法的特殊性，该算法是不可逆的。

4. 将用户 id，新密码和随机数保存到数据库中。用户注册成功。

用户登录

1. 服务端获取到用户的 id 和密码后，根据用户 id 从数据库中取出该用户的新密码和随机数。

2. 把用户传过来的旧密码和随机数交给用户注册第 2 步中的随机数和密码拼接算法，拼接后，得到一个新的字符串（和用户注册第 2 步得到的全新字符串是一模一样的）。

3. 将新字符串交给哈希算法处理将得到一个处理结果。

4. 如果处理后的结果和数据库中存储的新密码相同，那么，该用户传过来的密码是正确的，登录成功，否则，登录失败。