**结题报告**

**--基本工资管理系统**

**张秀艳1，杨金凝2**

**指导教师：张秀艳1**

(1. 吉林大学商学院；2. 吉林大学商学院)

**摘要：**为了保证项目团队按时保质地完成项目目标，便于项目团队成员更好地了解项目情况，使项目工作开展的各个过程合理有序，有必要以文件化的形式，把对于在项目生命周期内的工作任务范围、各项工作的任务分解、项目团队组织结构、各团队成员的工作责任、开发进度、项目内外环境条件、风险对策等内容以书面的方式描述出来，作为项目团队成员以及项目干系人之间的共识与约定，项目生命周期内的所有项目活动的行动基础，项目团队开展和检查项目工作的依据。

本项目开发计划用于从总体上指导工资管理系统项目顺利进行。本项目开发计划面向项目组全体成员。

**关键词**：工资管理系统

1. **项目简介**
   1. **项目背景**

（1）项目的委托单位、开发单位和主管部门

本项目由吉林大学计算机科学与技术专业软件工程课程设计提出

（2）该软件系统与其他

该系统的使用者为acme公司。该系统与其他系统没有关联。

**1.2 项目工具和手段**

（1）MySql:一个使用范围很广的关系型数据库管理系统；

（2）Apache：是世界使用排名第一的Web服务器软件；

（3）J2EE：是一套不同于传统应用开发的技术架构，包含许多组件，主要 可简化并规范应用系统的开发与部署，进而提高可移植性、安全与再用价值。

**1.3 项目内容**

本系统旨在开发一个工资管理系统管理平台，建立一个新的工资单系统，以取代现有系统。 Acme需要一个新的系统，允许员工以电子方式记录时间表信息，并根据工作时间和销售总额（委托人员）自动生成薪水。

新系统将是最先进的技术，并将具有基于Windows的桌面界面，允许员工输入时间信息，输入采购订单，更改员工偏好（如付款方式），并创建各种报表。该系统将在整个公司的个人员工桌面上运行。为安全和审计的原因，员工只能访问和编辑自己的计时卡和采购订单。

该系统将保留公司所有员工的信息（Acme目前在全球拥有约5,000名员工）。系统必须按照他们指定的方法按时确定每个员工的正确数量（请参阅稍后描述的可能的付款方式）。由于成本原因，Acme不想替换其遗留数据库之一，即项目管理数据库，其中包含有关项目和费用编号的所有信息。新系统必须与现有的项目管理数据库配合使用，该数据库是在IBM大型机上运行的DB2数据库。工资单系统将访问但不更新存储在项目管理数据库中的信息。

有些员工按小时工​​作，按时缴纳小时费。他们提交记录特定收费号码的日期和工作时数的时间表。如果有人工作超过8个小时，Acme会把这些额外的小时数提高到正常速率的1.5倍。小时工每星期五支付。

一些雇员的薪水是平的。他们虽然支付了平均工资，但却提交了记录日期和工作时间的时间表。这样，系统可以跟踪针对特定收费号码的工作时间。他们在本月的最后一个工作日支付。

部分受薪雇员也根据其销售收到佣金。他们提交反映销售日期和金额的采购订单。佣金率由每位员工确定，为10％，15％，25％或35％之间。

新系统最需要的功能之一是员工报告。员工将能够查询系统工作时数，总计账单总额（即收费编号），总年收入，剩余休假时间等。

员工可以选择自己的付款方式。他们可以将他们的薪水邮寄到他们选择的邮政地址，或者可以要求直接存款，并将工资单存入他们选择的银行账户。员工也可以选择在办公室领取薪水。

工资管理员维护员工信息。薪资管理员负责增加新员工，删除员工和更改所有员工信息，如姓名，地址和付款分类（小时，受薪，委托）以及运行管理报告。

工资单应用程序将在每个星期五和该月的最后一个工作日自动运行。那时候会付给适当的员工。系统将被告知员工将要支付什么日期，因此将从员工最后一次支付到指定日期的记录中支付记录。新系统的设计使得工资单始终自动生成，无需人工干预。

**1.4 项目产品**

1．程序：Acme工资管理系统

2．文档：

项目开发计划

需求分析文档

设计文档

测试计划

测试报告

3．运行环境：

操作系统：window

数 据 库：Mysql

4. 验收标准：功能完整、界面良好、文档齐全。

1. **项目可行信分析**

**2.1 系统规模**

本次实验的目的是实现一个工资单系统。由需求可知，本系统的使用人群是分布在全世界范围内的约5000人。使用人群具有基数大、流动性高、覆盖范围广等特点。利用传统的纸质的工资单已经不能满足用户的基本需求，因此需要我们尽快的开发出一个全新的工资单管理系统。

* 1. **已有系统分析**

目前的系统较为陈旧，因此希望在保留原有的数据库的基础上开发出一个全新的基Windows界面的工资单系统。除去原有的数据库其余部分全部需要重新设计。

* 1. **高层模型**

新系统将在原有数据库的基础上增加员工计时卡功能、员工上传销售记录功能、生成员工记录报告功能、选择自己付款方式功能和管理员修改员工信息功能等等。

* 1. **可供选择的解决方案**

方案一。方案一为利用Java开发一个桌面软件。桌面软件可以更为简单地实现比较复杂的操作，同时操作较为灵活。但是桌面软件开发较为复杂、占用人力资源较大，同时在用户主机上安装应用程序较为复杂、费时费力。

方案二。方案二是利用Java EE相关技术开发一个网站。用户只需进入相应的网站即可查询工资等相关信息。使用这个方案可以省去用户安装桌面软件的过程，用户可以使用更加灵活的方式来查询工资等相关信息。缺点是利用网站将使得本系统的功能具有局限性，难以实现复杂操作。

从技术角度来讲，方案一较为复杂，需要考虑员工对计算机操作的熟练程度、不同操作系

不同系统版本等因素，所需调试和适配的工作量较大。而方案二较为简单，在公司设立一个服务器后，只要用户能够上网，无论使用任何终端都能访问本系统，使用更为方便。而且对于开发人员来讲开发一个网站远比开发一个桌面软件简单，而且后期维护的成本更低，但是桌面软件的可拓展性也不容忽视。

从操作方面来讲，现代人群更为常用的是小型移动设备诸如手机、平板等。使用方案二的最大优势就是操作的简易性。用户很容易接收在网页上进行操作的方式，避免了使用桌面软件带来的高昂的学习成本。

从经济方面，任何一种方案都需要一台服务器，用于和用户进行数据传输。方案一所耗费的时间和人力比方案二多得多。

总体上来讲，方案二更具有优势。使用方案二可以更方便、高效、科学的对工资单进管

极大地减轻了人工操作的复杂度，同时也消除了可能出现的人为错误，使用较少的成本获得了更高的收益。

1. **项目实施计划**

**3.1 任务解决**

撰写项目开发计划

撰写软件需求文档

撰写设计文档

代码实现

撰写测试计划

代码测试

撰写测试报告

**3.2 进度**

|  |  |
| --- | --- |
| 时间（2017年） | 任务 |
| 5月23日—5月24日 | 撰写项目开发计划 |
| 5月25日—5月28日 | 撰写软件需求文档 |
| 5月29日—6月3日 | 撰写设计文档 |
| 6月4日—6月8日 | 代码实现 |
| 6月9日 | 撰写测试计划 |
| 6月10日—6月11日 | 代码测试 |
| 6月12日 | 撰写测试报告 |

**3.3 关键问题**

|  |  |  |  |
| --- | --- | --- | --- |
| 风险类别 | 风险描述 | 可能 | 影响 |
| 规划风险 | 计划是乐观的“最佳实例”（而不是现实的“预期实例”）。 | 0.9 | 0.6 |
|  | 产品比估计的大（从准则、功能、模式的角度来看）。 | 0.6 | 0.5 |
| 组织风险 | 项目的规划与实现之间搁置的时间太长。 | 0.6 | 0.8 |
| 技术风险 | 用户权限划分，适用验证码进行验证的技术欠缺。 | 0.6 | 0.6 |
|  | 容易出错的模式所需要的测试、设计和执行工作比预料的多。 | 0.7 | 0.9 |
|  | 开发工具不像预料的那样有效，开发者需要时间来创建有关工作或改用新工具。 | 0.5 | 0.8 |

**3.4 交付期限**

2017年6月13日

**3.5 专题计划要点**

质量保证计划：

1. 在软件设计过程中构筑软件质量；
2. 代码质量保证优先原则；
3. 只引用属于自己的存贮空间；
4. 防止引用已经释放的内存空间；
5. 过程/函数中分配的内存，在过程/函数退出之前要释放；
6. 过程/函数中申请的（为打开文件而使用的）文件句柄，在过程/函数退出之前要关闭；
7. 防止内存操作越界；
8. 认真处理程序所能遇到的各种出错情况；
9. 系统运行之初，要初始化有关变量及运行环境，防止未经初始化的变量被引用，并对加载到系统中的数据进行一致性检查；
10. 严禁随意更改其它模块或系统（不属于自己）的有关设置和配置，不能随意改变与其它模块的接口；
11. 注意易混淆的操作符。当编完程序后，应从头至尾检查一遍这些操作符，以防止拼写错误；
12. 有可能的话，if语句尽量加上else分支，对没有else分支的语句要小心对待。switch语句必须有default分支；
13. 不使用与硬件或操作系统关系很大的语句，而使用建议的标准语句，以提高软件的可移植性和可重用性；
14. 精心构造算法，并对其性能、效率进行测试，对较关键的算法最好使用其它算法来确认；
15. 注意表达式是否会上溢、下溢，使用变量时要注意其边界值；
16. 系统应具有一定的容错能力，对一些错误事件（如用户误操作等）能进行自动补救；
17. 对一些具有危险性的操作代码要仔细考虑，防止对数据、硬件等的安全构成危害，以提高系统的安全性。

**4 项目详细设计**

**4.1 动态对象建模**

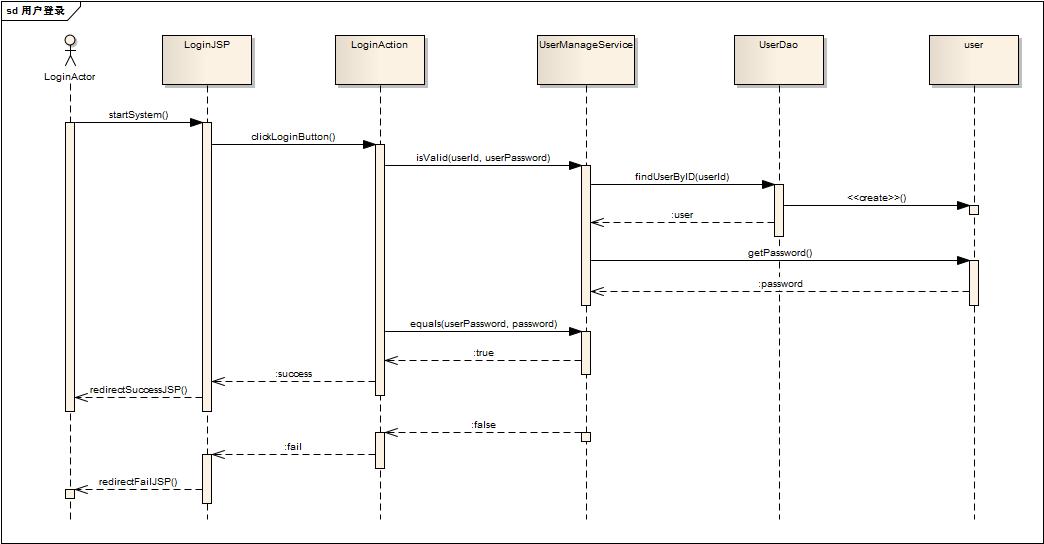
**4.1.1 顺序图**

几个主要用例的场景顺序图：

（1）普通员工登录用例

场景具体化：

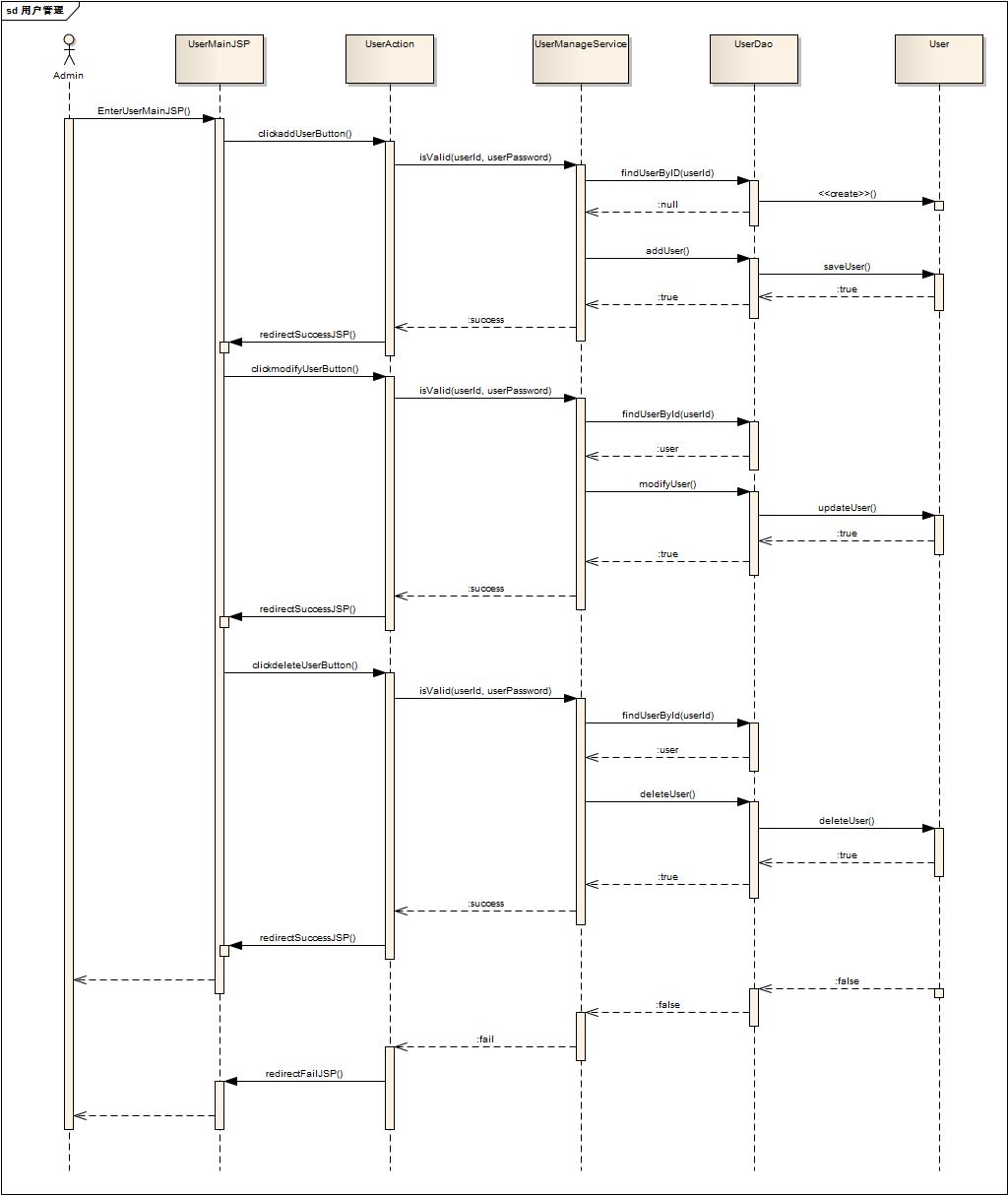
1. 管理员进入登录页面
2. 管理员填写用户名和密码，点击登录按钮
3. 系统判断用户名和密码是否正确匹配
4. 管理员登录成功，跳转到登录成功的界面或者登录失败，跳转到登录失败提示的界面



（2）管理员管理用户用例

场景具体化：

1. 管理员进入用户管理页面
2. 用户完成用户管理表单中相关内容的填写
3. 用户点击增加、修改或删除按钮，提交表单
4. 1. 验证用户信息，增加用户成功，系统保存用户信息
5. 增加用户失败
6. 验证用户信息，修改用户成功，系统保存用户信息修改用户失败
7. 删除用户成功，系统删除用户信息
8. 删除用户失败
9. 修改用户失败
10. 删除用户成功，系统删除用户信息
11. 删除用户失败

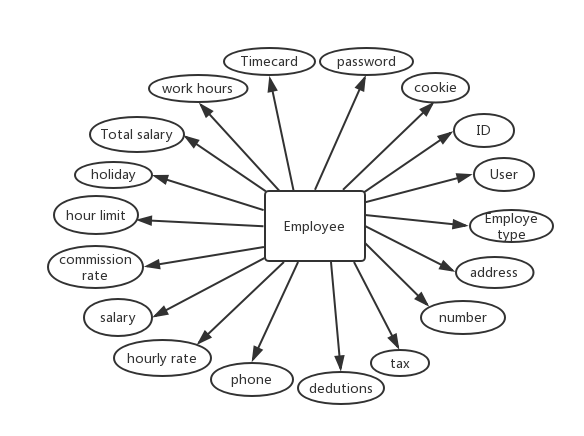


**4.2 数据库设计**

**4.2.1 员工信息数据库**

员工信息数据库分为两大类，分别是Payroll Administrator（薪资管理员）和Employee （普通员工）。其中薪资管理员管理着所有员工的基本信息，并且能查看、修改、添加、删除员工信息。而普通员工只能查看和修改自己的部分信息。

员工信息系统的ER图：



当有新员工加入企业时，薪资管理员需要为该员工创建新的员工信息。下表则是管理员创建一个新的员工所需要填写的信息，当表格创建完毕之后系统为自动生成员工的唯一标识的工号。

薪资管理员创建的员工信息

|  |  |  |
| --- | --- | --- |
| **字段** | **中译** | **备注** |
| Name | 姓名 |  |
| Employee type | 员工类型 | 小时工/固定工资 |
| Mailing address | 邮件地址 |  |
| Social security number | 社会安全号码 |  |
| Standard tax deductions | 标准减税 | 免征个税的额度 |
| Other dedutions | 其他扣款 |  |
| Phone number | 电话号码 |  |
| Hourly rate | 时薪 | 适用于小时工 |
| Salary | 薪水 | 适用于固定工资 |
| Commission rate | 提成比例 | 如果员工没有提成比例为0% |
| Hour limit | 工作时长限制 | 一些员工不允许加工 |
| Password | 密码 |  |

特别说明：Employee type有三种类型分别为Salaried Employee 固定薪资的员工、Commissioned Employee 固定工资+提成的员工、Hourly Employee 小时工。

薪资管理员有他自己的职能，在系统中管理员可以选择他想要的操作并且进行该操作

|  |  |
| --- | --- |
| **操作** | **子流程** |
| 添加员工信息 | Add an Employee |
| 修改员工信息 | Update an Employee |
| 删除员工信息 | Delete an Employee |
| 查看员工信息 | Browse an Employee |

Employee（普通员工）能在系统中查询到自己的相关信息，并且能够修改自己的身份信息（在表中标注了），下表为员工能够看到的信息。

普通员工的信息表

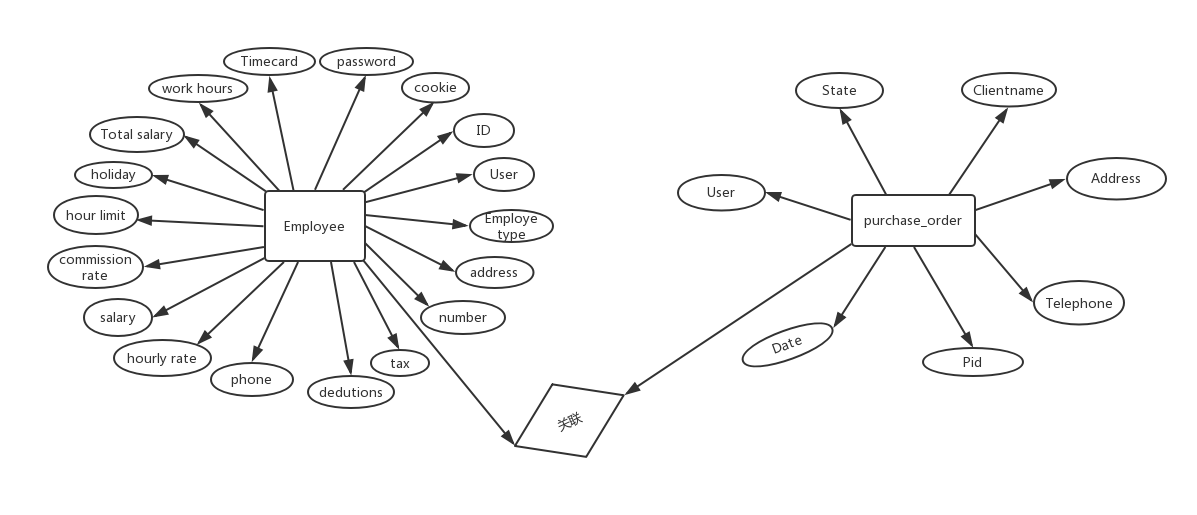
|  |  |  |
| --- | --- | --- |
| **字段** | **中译** | **备注** |
| ID | 员工唯一标识工号 |  |
| User | 员工登陆的用户名 |  |
| Name | 姓名 | （能修改） |
| Employee type | 员工类型 | 小时工/固定工资 |
| Mailing address | 邮件地址 | （能修改） |
| Social security number | 社会安全号码 |  |
| Standard tax deductions | 标准减税 | 免征个税的额度 |
| Other dedutions | 其他扣款 |  |
| Phone number | 电话号码 | （能修改） |
| Hourly rate | 时薪 | 适用于小时工 |
| Salary | 薪水 | 适用于固定工资 |
| Commission rate | 提成比例 | 如果员工没有提成则比例为0% |
| Hour limit | 工作时长限制 | 一些员工不允许加工 |
| Holiday | 剩余假期时间 |  |
| Total salary | 实际获得工资 | 员工最后实际发到手上的工资 |
| Total working hours | 工作总时长 | 员工工作的总时长 |
| Timecard | 出勤记录 |  |
| Password | 登陆密码 |  |
| Cookie | 浏览器缓存 |  |
| Dimission | 登陆密码 |  |

Total salary、Total working hours、Timecard都是需要和考勤系统和订单系统共同才能得到最后的实际数字。

**4.2.2 订单管理数据库**

在该管理系统下需要专门建立一个有关企业项目的数据库，来记录每一个订单的具体完成情况，以便于企业的订单管理以及员工最后实领薪资的计算。而该数据库是由Commissioned Employee来进行管理的。

项目管理数据库的ER图：

Commissioned Employee可以对自己的订单进行管理，他能够做以下操作

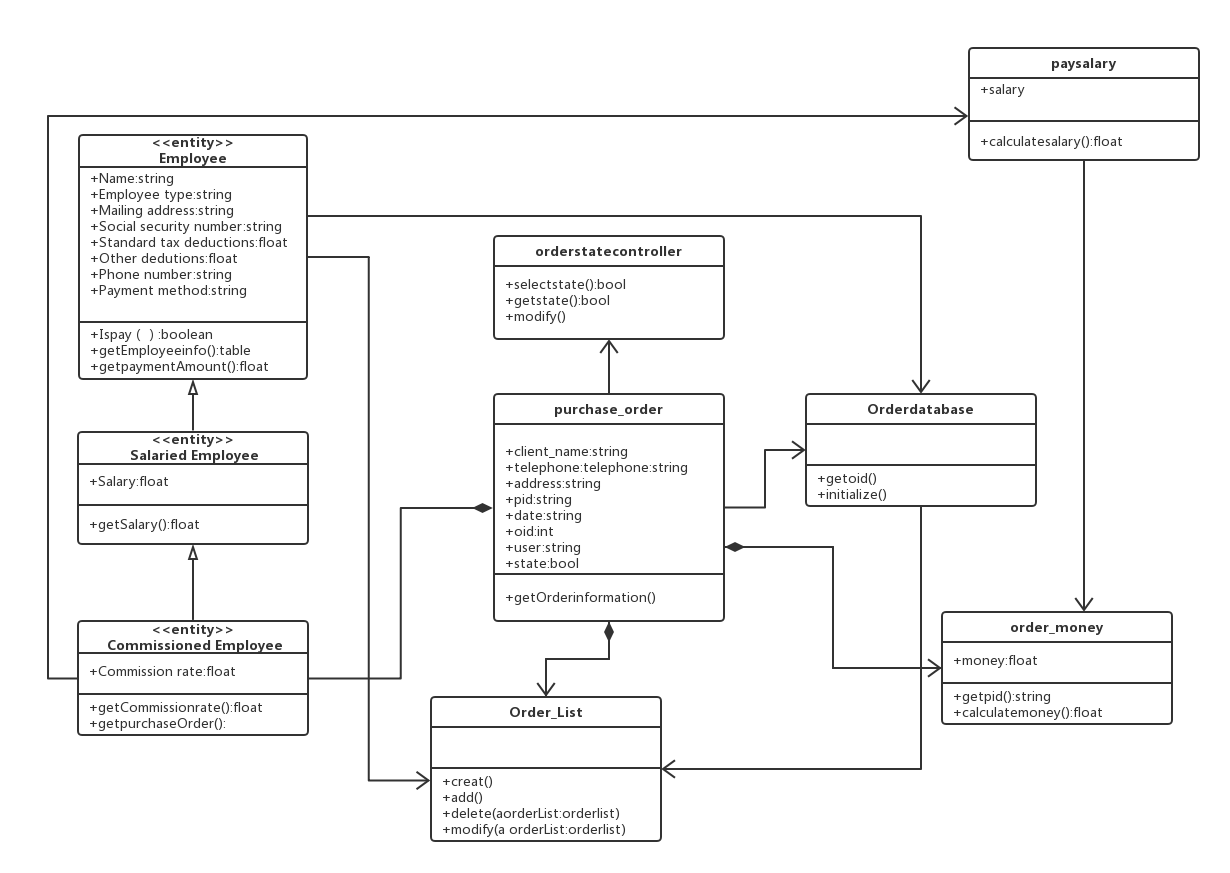
|  |  |
| --- | --- |
| **操作** | **子流程** |
| 创建订单 | Create a Purchase Order |
| 修改订单 | Update a Purchase Order |
| 删除订单 | Delete a Purchase Order |
| 查看订单 | Browse a Purchase Order |

对于创建订单来说，Commissioned Employee需要填入下表的相关信息来创建订单，之后系统自动生成唯一的订单号OID，下表是订单数据库的结构。

订单

|  |  |  |
| --- | --- | --- |
| **字段** | **中译** | **备注** |
| Clientname | 客户姓名 |  |
| Address | 客户联络点 |  |
| Telephone | 客户电话 |  |
| Pid | 购买的产品 |  |
| Date | 日期 |  |
| User | 员工 |  |
| State | 订单的状态 | 当订单完结束就会关闭订单无法修改了 |

下图是订单系统的UML图：

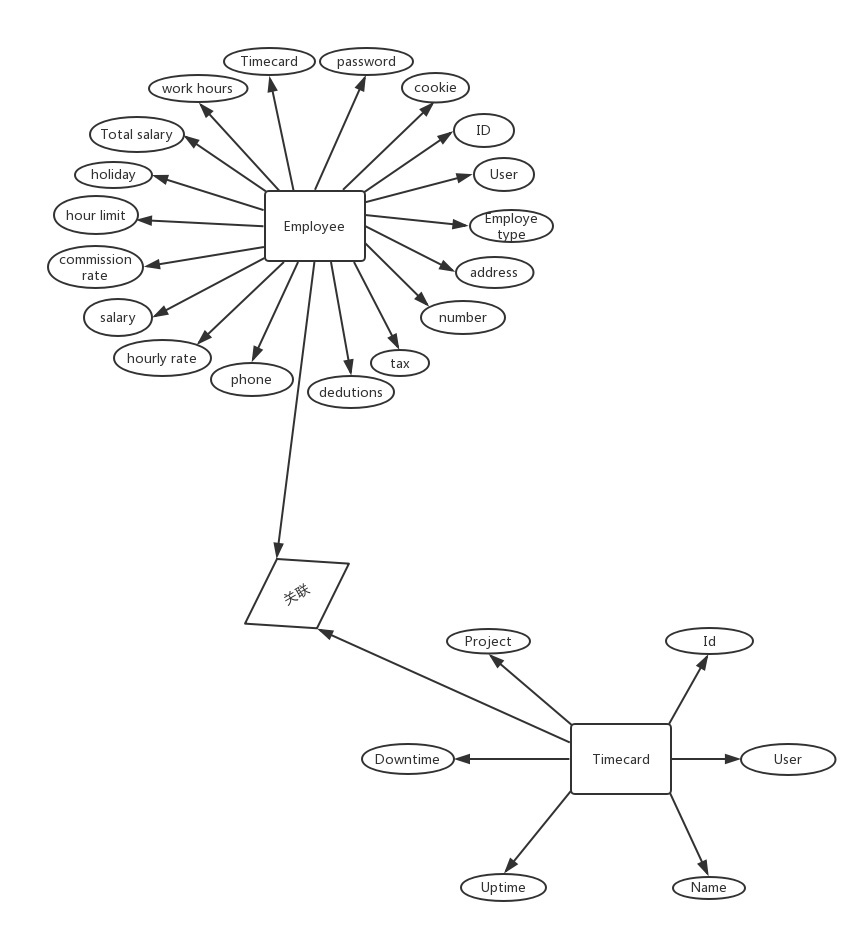


**4.2.3 考勤记录系统**

员工每一天上班下班需要打卡，每一次打卡数据将存入本地的一个文件当中，当员工提交考勤卡的时候，会将每一日打卡所记录的信息存入数据库当中。

员工自己更新以及提交考勤卡信息，小时工和受薪员工必须每周提交考勤卡，考勤卡上面记录了这一周的总工时以及这些工时都用在了哪些项目上。员工只能修改当前开支周期的考勤卡，并且修改只能在提交之前完成。

考勤记录数据库的ER图：



系统会为员工每一次开始打卡到结束打卡生成一条记录保存在考勤记录的数据库里。

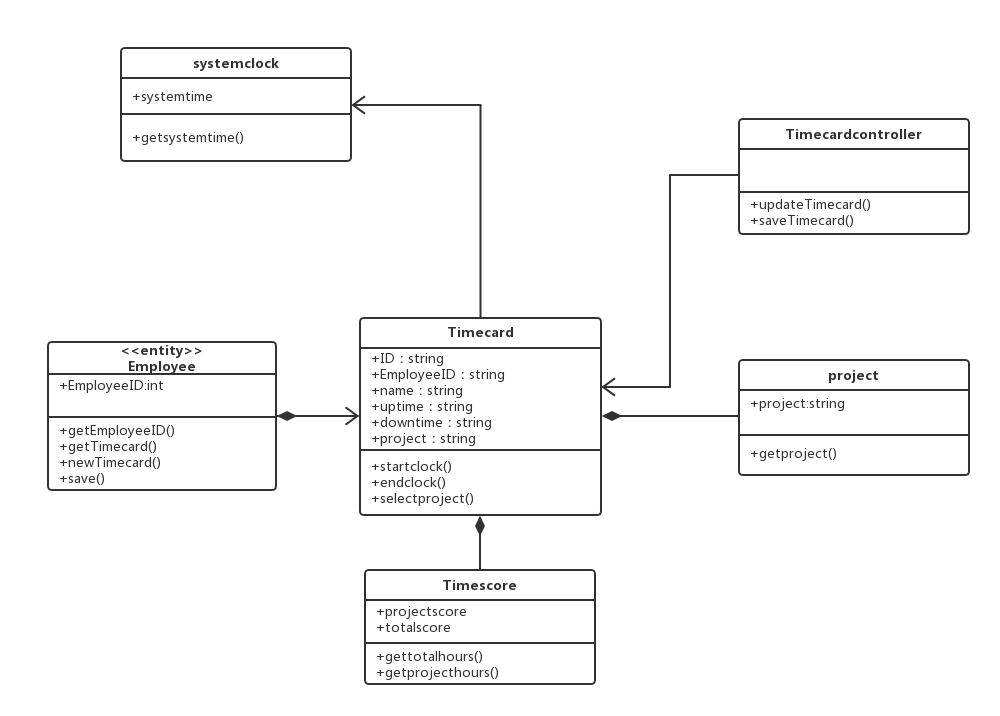
考勤记录系统的数据库

|  |  |  |
| --- | --- | --- |
| 字段 | 中译 | 备注 |
| Id | 打卡记录ID | 并不是员工唯一标识id而是打卡记录唯一标识 |
| User | 用户 | 该员工的用户名 |
| Name | 名字 | 员工姓名 |
| Uptime | 打卡时间 | 开始打卡时记录的时间 |
| Downtime | 结束打卡时间 | 结束打卡时记录的时间 |
| Project | 项目 | 在打卡的时间里时间花在哪个项目上 |

然后根据这个数据库可以生成工作总时长的数据库。

工作总时长数据库

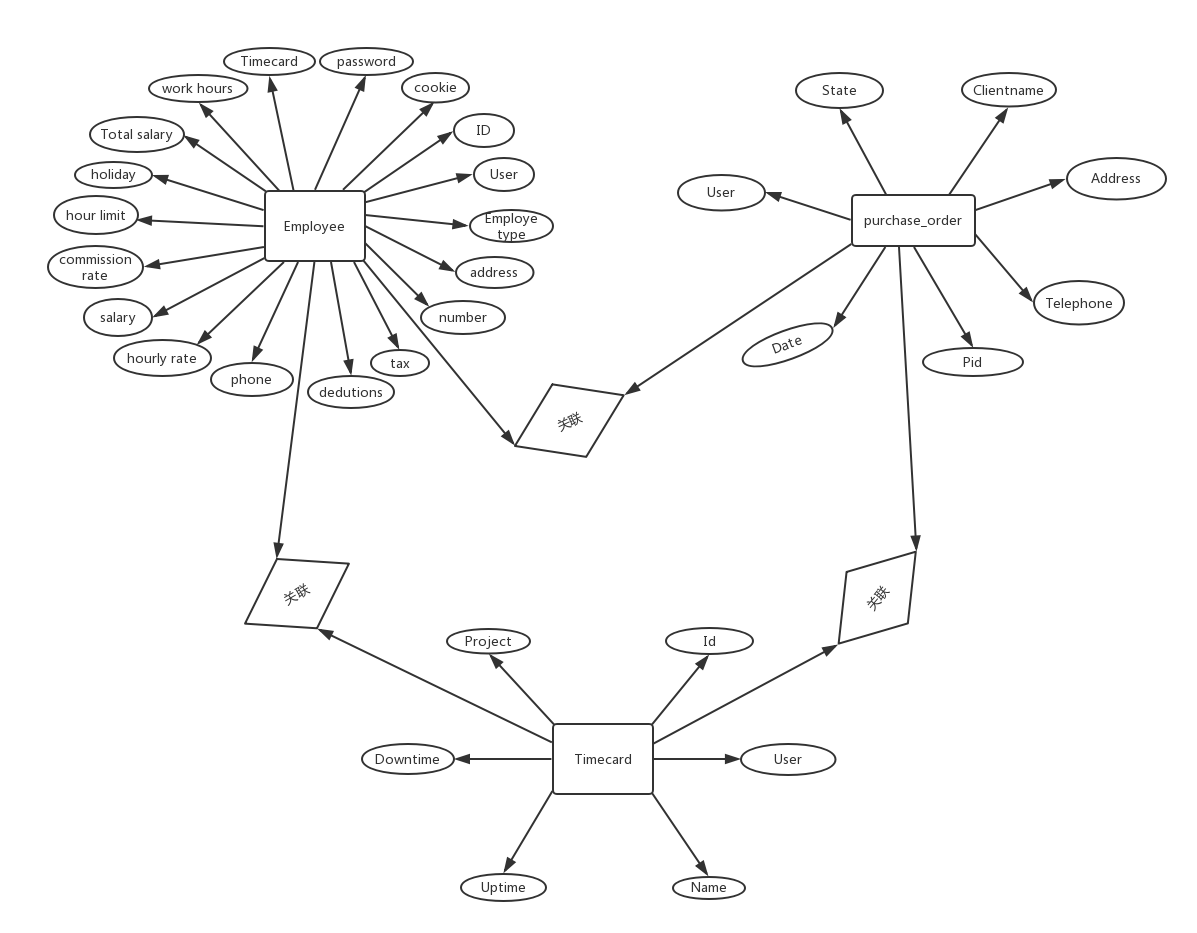
|  |  |  |
| --- | --- | --- |
| 字段 | 中译 | 备注 |
| Id | 用户的id |  |
| User | 用户 |  |
| Subject | 项目 |  |
| Score | 时长 |  |

下图是员工管理系统的的UML图：

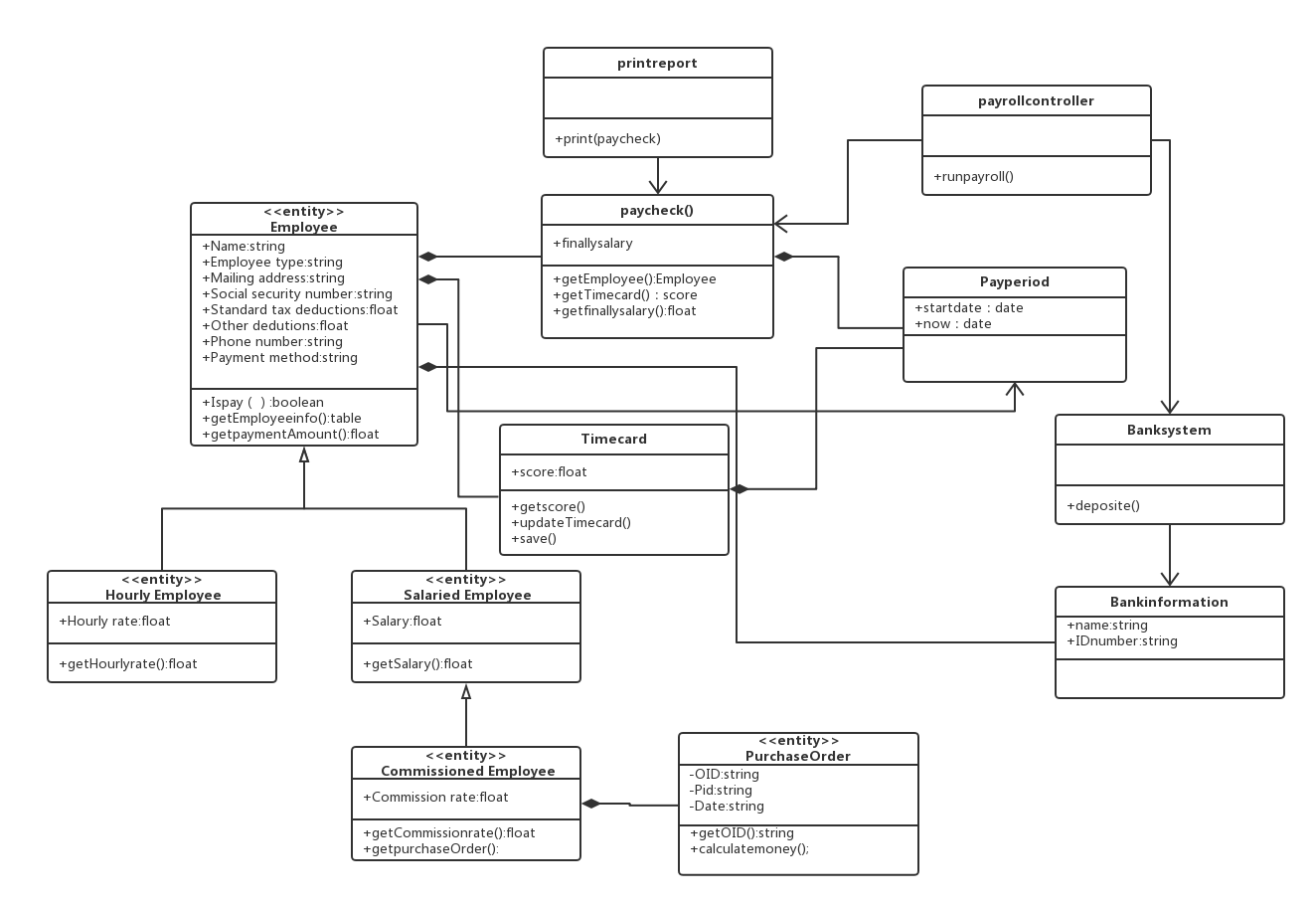
**4.3 薪资结算系统**

在该企业中，具有多种不同的员工，所以对于每一种员工都有不同的工资结算方法和工资结算周期。我们需要通过订单数据库、员工信息数据库、打卡信息数据库的信息来最后确定员工的工资，最后和银行系统共同协作来完成工资的发放.

下面是该系统中所使用数据库的ER图：



下图是资薪结算系统的UML图表现其中各个类的相互的联系。



**4.4 员工管理报告的生成**

员工能够自己生成一个有关自己的员工管理报告，在报告中他能看到自己的工作小时数、为某个项目累计贡献的时间，休假/病假以及年初至今的总工资的报告。

员工报告所读取到的信息

|  |  |  |
| --- | --- | --- |
| **字符** | **中译** | **备注** |
| OID | 员工工号 |  |
| Name | 员工姓名 |  |
| Working hours | 总计工作时间 |  |
| Project working hours | 项目累计贡献时间 |  |
| Holidays | 休假病假 |  |
| Salary | 年初至今的工资 |  |

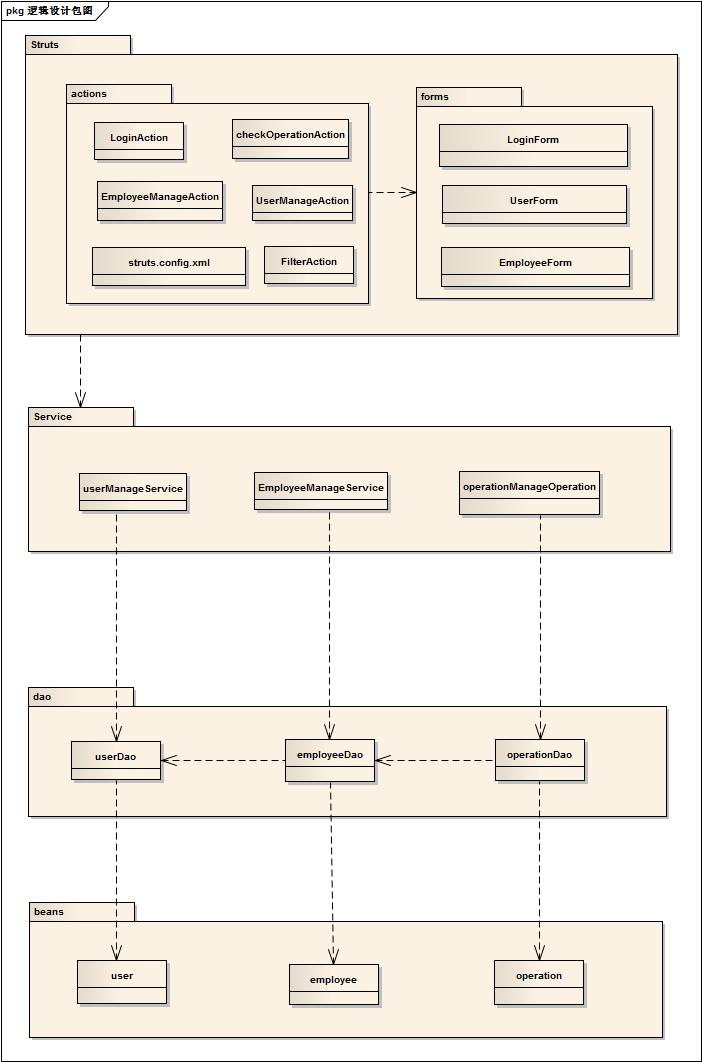
资薪管理员也需要生成员工管理报告，是对所有员工的管理信息，他需要了解每一个员工的出勤和工资情况，下表是员工管理报告的生成格式：

员工管理报告

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OID/工号** | **Name姓名** | **Working hours/工作时长** | **Total Salary/总工资** | **备注** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**4.5 架构设计**

**4.5.1分层逻辑设计**



**4.5.2 程序实现架构**

开发环境

|  |  |
| --- | --- |
| 操作系统 | Windows 10 \ Mac |
| 数据库 | MySQL 5.1 |
| IDE | MyEclipse10.0 |
| 开发框架 | Struts2.0 + Spring3.0 + Hibernate4.0 |
| 测试 | Junit |

第三方组件

|  |  |
| --- | --- |
| 测试组件 | Junit |
| 数据持久化组件 | Hibernate4.0 |
| 业务服务组件 | Spring3.0 |
| UI 表示组件 | Struts2.0 |

**4.6 DCD到应用框架的映射原则**

逻辑包设计

在包的设计上，按照com.项目名.层名的方式来建立，每一个功能放在一个单独的包中，由此能得出较为清晰明了、符合实际的逻辑架构，也符合松耦合及关注分离的要求；

1. Web层

调用业务逻辑层，并将业务数据返回给表现层做组织表现。这部分是通过struts来实现的，即程序中的\*Action.java类。

1. Service 层

从 DCD 中，可以看到，struts中\*Action类调用Service层中提供的服务实现其表现层的功能，对应着SSH 框架中的Service 层，负责实现业务逻辑。业务逻辑层以DAO 层为基础，通过对DAO 组件的包装，完成系统所要求的业务逻辑。

1. DAO层

从 DCD 中，可以看到，项目中有描述需保存在数据库中内容的类，如用户信息、职工信息、操作信息等，它实际上对应的是数据库中一系列的表；这些类在SSH 的框架中，对应的是持久化对象，即通过实体关系映射工具将关系型数据库的数据映射成对象，方便地实现以面向对象方式操作数据库，并通过DAO 层——负责与持久化对象交互，实现数据的CRUD操作。在SSH框架中，是通过Hibernate实现的，对应\*DAO.java类。

**4.7 控制层与UI层实现约定**

**4.7.1**[**控制层与UI传输变量规定：**](#_Toc268966721)

UI层目前主要有2个界面，一个为登陆界面，一个为系统处理业务的主界面。

（1）登陆界面传输变量：

用户权限验证数据：包括用户账号和密码，UI获取到用户账号密码后将数据传输到控制层进行判断，控制层将判断结果返回UI层。

针对控制层返回的用户权限显示不同的功能项：分别为普通用户，工资管理员和审核员。

（2）系统业务主界面传输变量：

主界面有三个功能模块，分别为系统公告、系统消息和操作日志。

UI层向控制层请求系统公告信息，并逐条显示在UI公告面板中。系统消息和操作日志同样是由控制层推送给UI层来逐条显示。

子功能项：包括基本工资查询、修改工资标准、薪级升级、年终奖、年终考核优秀、日常工资处理、工资变动、工资补发扣发和工资发放管理功能，它们在处理业务时分别会向控制层请求数据，进行数据交互，由于变量字段过多不一一列举。

**5 项目总结**

a.在需求获取阶段，我们发现仅靠wiki上的文档是不足于搞懂全部需求的。因此，我们与网络中心的老师保持沟通联系，解决需求分析上的问题。  
b.如何基于MyEclipse连接mysql数据库，以搭建SSH2的开发环境，成为我组最困扰的技术问题。SSH2是一个基于MVC的web框架，我们的设计应该遵循MVC的原则。大多数组员不熟悉web开发，这得花一定的功夫去学习。  
c.书写系统说明书类的文档是不容易的。我们多次举行小组会议，在线上和线下及时沟通，有必要时调整分工，最终把所有事情说清楚，并表达在文档里。

d.在实际开发过程中,经过协商,我们组将系统功能划分为两大功能模块：基本工资体系模块和基本工资管理模块。其中，基本工资体系模块又分为两个子模块，基本工资管理模块又分为三个子模块。这样以来，理顺了开发任务，对组员间分配开发任务有很大的帮助。

e.本组进度较慢，原因是系统基本功能项目略多，在一次迭代过程中开发完成所有的基本功能比较困难。我们采取了这样的措施：先期开发复用价值较高的基本功能模块，完成这些功能模块的编写和测试工作。

f. 我们对Java Web开发不熟悉，不知道怎么在J2EE中将后台和前端页面联系起来。单着来说，写后台数据处理是一件比较容易的事情，写前端JSP页面也是一件比较容易的事情，但是它俩碰到一起就有问题了。涉及到数据流动，开始总是碰到各种各样的问题，不能把后台获取的数据在页面上按照预定的方式显示出来。还有就是一些技术不太了解，比如页面的动态刷新和部分刷新，这虽然是一个不大的改进，但就用户体验来说，就可以拥有一个质的提升。总之，善于利用资料，善于利用网络，其实上面碰到的问题在网上基本上都有人已经问过了，而且也有了相应的解决办法，只要我们把网络利用好了，碰到问题之后在网上查找一些资料，解决这些问题也就是时间的问题了。

g.终于到了8月1日。经过这么久的努力，大家也差不多完成了主要的功能模块。下一阶段的主要工作是对已经完成的部分进行完善。主要是对系统进行测试，然后找出系统的异常以及bug，然后对系统增加一些异常处理，并且修复系统存在的bug。完成这些后再对前端页面的UI进行完善，希望能够在易用的基础上使界面更加的美观。最后准备一下最后的最终展示，为我们这一个月来的工作画上完美的句号。

1. **参考文献**

[1]数据库系统概论[M]. 高等教育出版社 , 萨师煊, 1983.

[2]管理信息系统设计与实施[M]. 清华大学出版社 , 黎连业,李淑春编著, 1998.

[3]企业信息化[M]. 清华大学出版社 , 赵守香, 2008.

[4]面向对象与经典软件工程[M]. 机械工业出版社 , (美)StephenR.Schach著, 2002.

[5]UML应用建模实践过程[M]. 机械工业出版社 , 尤克滨编著, 2003.

[6]软件工程导论（第六版）。清华大学出版社，张海藩，牟永敏 编著，2013

**证 明**

兹有商学院信息管理与信息系统专业杨金凝同学，于2017年5月至2017年6月参与基本工资管理系统开发项目。具体情况如下：

参与工作1：初始化整体框架模板

参与工作2：根据模板标准化修改了权限组的命名和开发流的命名

参与工作3：整合清理无用项目分支

参与工作4：完整个项目成员 merge 版本管理

特此证明。

指导老师：

**2018** 年 **5** 月 **30** 日