**工程售后项目详细设计说明书**

**<V1.0>**

编 写：

审 核：

批 准：

**版权属于锐安科技所有，无锐安科技的书面同意，任何个人或组织无权拷贝。**

**文档修订记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本号 | 变化状态 | 简要说明  （变更内容和变更范围） | 日期 | 修改人 | 审批人 |
| V1.0 | C | 初次创建 | 2018-11-05 |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

\*变化状态：C――创建，A——增加，M——修改，D——删除

目录

[**1** **文档简介** 5](#_Toc522182486)

[1.1 编写目的 5](#_Toc522182487)

[1.2 文档范围 5](#_Toc522182488)

[1.3 预期读者和阅读建议 5](#_Toc522182489)

[1.4 定义、缩写词、略语 5](#_Toc522182490)

[1.5 参考资料 5](#_Toc522182491)

[1.6 功能列表 6](#_Toc522182492)

[1.6.1 任务管理 6](#_Toc522182493)

[1.6.2 薪酬管理 12](#_Toc522182494)

[2 第零层设计 18](#_Toc522182495)

[2.1 系统与其他系统的关系 18](#_Toc522182496)

[2.1.1 与组织架构、用户权限系统接口相关 18](#_Toc522182497)

[2.2 设计思路（可选） 19](#_Toc522182498)

[2.2.1 设计方法 19](#_Toc522182499)

[2.2.2 设计可选方案 19](#_Toc522182500)

[2.2.3 设计约束 19](#_Toc522182501)

[**3** **第一层设计** 19](#_Toc522182502)

[3.1 分解描述 19](#_Toc522182503)

[3.1.1 模块分解 19](#_Toc522182504)

[3.1.2 并发进程处理分解（可选） 19](#_Toc522182505)

[3.1.3 数据分解 19](#_Toc522182506)

[3.2 依赖性描述 20](#_Toc522182507)

[3.2.1 运行设计 20](#_Toc522182508)

[3.2.2 数据依赖关系 20](#_Toc522182509)

[3.3 接口描述（可选） 20](#_Toc522182510)

[3.3.1 模块接口 20](#_Toc522182511)

[3.3.2 进程接口 20](#_Toc522182512)

[**4** **第二层设计（可选）** 21](#_Toc522182513)

[4.1 模块1 21](#_Toc522182514)

[4.1.1 分解描述 21](#_Toc522182515)

[4.1.2 数据实体描述 21](#_Toc522182516)

[4.1.3 依赖性描述 21](#_Toc522182517)

[4.1.4 接口描述 21](#_Toc522182518)

[4.2 模块2 21](#_Toc522182519)

[4.2.1 分解描述 21](#_Toc522182520)

[4.2.2 数据实体描述 21](#_Toc522182521)

[4.2.3 依赖性描述 21](#_Toc522182522)

[4.2.4 接口描述 21](#_Toc522182523)

[**5** **配置和控制（可选）** 22](#_Toc522182524)

[5.1 Startup启动 22](#_Toc522182525)

[5.2 Closing启动 22](#_Toc522182526)

[5.3 MIB表项的创建 22](#_Toc522182527)

[5.4 MIB表项的删除 22](#_Toc522182528)

[5.5 MIB表项的更改 22](#_Toc522182529)

[**6** **数据库（可选）** 22](#_Toc522182530)

[6.1 实体、属性及它们之间的关系 22](#_Toc522182531)

[6.2 实体关系图 22](#_Toc522182532)

# **文档简介**

## 编写目的

1. 本文档是工程售后项目的概要分析说明书，其主要目的是系统地、明确地、规范地描述本部分的功能需求，包括：功能范围，功能详细描述，优先级，输入输出，验收准则，异常处理等。
2. 本文档同时从开发者角度阐述每个功能点技术可行性，实现程度，技术原理，处理流程的情况。
3. 作为项目验收标准之一。
4. 为项目设计开发，产品维护提供参考依据。

## 文档范围

* 项目名称：锐安工程售后项目
* 上级文档：《需求分析说明书.doc》
* 开发者：段明澈，马飞飞
* 用户： 公司员工。

## 预期读者和阅读建议

本文的阅读者是有机体-系统研发部成员及外包开发人员。阅读人员需对软件工程的知识有一定的了解，需要对锐安家园系统有一定的了解。

## 定义、缩写词、略语

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **术语/缩略语** | **说明/定义** | **英文全称** | **中文译名** |
| 1 | ERP | ERP | Enterprise Resource Planning | 企业资源计划 |

## 参考资料

|  |  |  |  |
| --- | --- | --- | --- |
| **作者** | **文献名称** | **出版单位（或归属单位）** | **版本或日期** |
|  | 《需求分析说明书.doc》 | 锐安 | 2018.07.19 |
|  |  |  |  |

## 功能列表

### 工程项目

#### 工程立项/工程项目管理

* + 1. 工程立项实现项目立项的功能。

1. 添加、编辑功能，新建库表保存填写的任务主数据：construct\_project

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **字段描述** | **字段类型** | **长度** |
| 1 | id | 编号 | bigint | 19,0 |
| 2 | created\_at | 数据建立时间 | datetime |  |
| 3 | created\_by | 数据建立人 | varchar | 50 |
| 4 | updated\_at | 数据更新时间 | datetime |  |
| 5 | updated\_by | 数据更新人 | varchar | 50 |
| 6 | applicant\_code | 申请人编号 | varchar | 50 |
| 7 | applicant\_name | 申请人姓名 | varchar | 20 |
| 8 | project\_manager\_code | 项目经理编号 | varchar | 50 |
| 9 | project\_manager\_name | 项目经理命名 | varchar | 20 |
| 10 | contract\_construct\_first\_organization\_code | 合同实施一级部门编号 | varchar | 50 |
| 11 | contract\_construct\_first\_organization\_name | 合同实施一级部门名称 | varchar | 100 |
| 12 | contract\_construct\_second\_organization\_code | 合同实施二级部门编号 | varchar | 50 |
| 13 | contract\_construct\_second\_organization\_name | 合同实施二级部门名称 | varchar | 100 |
| 14 | project\_code | 项目编号 | varchar | 50 |
| 15 | project\_name | 项目名称 | varchar | 200 |
| 16 | fee\_organization\_code | 费用承担部门编号 | varchar | 50 |
| 17 | fee\_organization\_name | 费用承担部门名称 | varchar | 100 |
| 18 | contract\_organization\_code | 合同所属部门编号 | varchar | 50 |
| 19 | contract\_organization\_name | 合同所属部门名称 | varchar | 100 |
| 20 | contract\_amount | 合同金额 | bigint | 19,0 |
| 21 | budget\_code | 预算编号 | varchar | 20 |
| 22 | project\_category | 项目类型 | varchar | 20 |
| 23 | contract\_category | 合同类型 | varchar | 20 |
| 24 | contract\_cost\_center\_code | 合同成本中心编号 | varchar | 50 |
| 25 | contract\_cost\_center\_name | 合同成本中心名称 | varchar | 50 |
| 26 | contract\_first\_organization\_code | 合同主体一级部门编号 | varchar | 50 |
| 27 | contract\_first\_organization\_name | 合同主体一级部门名称 | varchar | 100 |
| 28 | project\_description | 项目介绍 | varchar | 500 |
| 29 | install\_system | 是否安装运维系统 | tinyint | 3,0 |
| 30 | install\_data | 是否涉及数据接入 | tinyint | 3,0 |
| 31 | construct\_province | 实施地点省份 | varchar | 20 |
| 32 | construct\_city | 实施地点城市 | varchar | 20 |
| 33 | custom\_code | 客户编号 | varchar | 50 |
| 34 | custom\_name | 客户名称 | varchar | 100 |
| 35 | custom\_service\_manager\_code | 售后负责人编号 | varchar | 50 |
| 36 | custom\_service\_manager\_name | 售后负责人姓名 | varchar | 20 |
| 37 | custom\_service\_organization\_code | 售后负责部门编号 | varchar | 50 |
| 38 | custom\_service\_organization\_name | 售后负责部门名称 | varchar | 100 |
| 39 | project\_state | 立项状态 | varchar | 20 |
| 40 | project\_date | 立项时间 | datetime |  |
| 41 | project\_budget\_amount | 项目预算合计 | bigint | 19,0 |
| 42 | plan\_finish\_date | 计划完成时间 | datetime |  |
| 43 | equip\_arrival\_state | 设备到货状态 | varchar | 20 |
| 44 | equip\_check\_state | 设备验收状态 | varchar | 20 |
| 45 | finish\_report\_state | 完工报告签署状态 | varchar | 20 |
| 46 | project\_runing\_state | 项目执行状态 | varchar | 20 |
| 47 | check\_state | 验收状态 | varchar | 20 |
| 48 | flow\_code | 立项审批流编号 | varchar | 100 |

b）字段说明：

编号：工程项目ID，新增立项时自动生成；

数据建立时间：不允许为空；

数据建立人：获取当前用户，不允许为空；

数据更新时间：系统自动填写数据更新的时间；

数据更新人：系统自动填写数据更新人；

申请人编号：必填，无则默认项目经理；

申请人姓名：必填，无则默认项目经理；

项目经理编号：必填，无则默认申请人；

项目经理命名：必填，无则默认申请人；

合同实施一级部门编号：必填，如为空则默认项目经理所对应的一级部门；

合同实施一级部门名称：根据编号带出；

合同实施二级部门编号：必填，如为空则默认项目经理所对应的二级部门；

合同实施二级部门名称：根据编号带出；

项目编号：通过API获取，必填；

项目名称：根据编号带出；

费用承担部门编号：通过API获取，必填；

费用承担部门名称：根据编号带出；

合同所属部门编号：通过API获取，必填；

合同所属部门名称：根据编号带出；

合同金额：通过API获取；

预算编号：根据规则自动生成；

项目类型：必填，通过API获取；

合同类型：必填，通过API获取；

合同成本中心编号：必填，通过API获取；

合同成本中心名称：根据编号带出；

合同主体一级部门编号：必填，通过API获取；

合同主体一级部门名称：根据编号带出；

项目介绍：自动时通过API推送，手动时填写；

是否安装运维系统：自动时API推送，手动可选；

是否涉及数据接入：自动时API推送，手动可选；

实施地点省份：自动时API推送，手动可选；

实施地点城市：自动时API推送，手动可选；

客户编号：通过API获取；

客户名称：通过API获取；

售后负责人编号：自动时API推送，手动可选；

售后负责人姓名：自动时API推送，手动可选；

售后负责部门编号：通过API获取；若无根据售后负责人所在部门调出；

售后负责部门名称通过API获取；若无根据售后负责人所在部门调出；

立项状态：必填，但不展示该字段；

立项时间：当前时间；

项目预算合计：映射获取；

计划完成时间：默认空；

设备到货状态：默认空；

设备验收状态：默认空；

完工报告签署状态：默认空；

项目执行状态：默认空；

验收状态：默认空；

立项审批流编号：

c）立项说明：

系统可自动立项，也可手动立项。

2） 工程项目成员变更日志实现保存、查看人员变更记录的功能

a) 添加、编辑功能，新建库表保存填写的任务主数据：construct\_project\_member\_change\_record

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **字段描述** | **字段类型** | **长度** |
| 1 | id | 编号 | bigint | 19,0 |
| 2 | created\_at | 数据建立时间 | datetime |  |
| 3 | created\_by | 数据建立人 | varchar | 50 |
| 4 | updated\_at | 数据更新时间 | datetime |  |
| 5 | updated\_by | 数据更新人 | varchar | 50 |
| 6 | construct\_project\_id | 工程项目ID | bigint | 19,0 |
| 7 | old\_project\_member\_codes | 原项目成员编号 | varchar | 50 |
| 8 | old\_project\_member\_names | 原项目成员姓名 | varchar | 20 |
| 9 | new\_project\_member\_codes | 新项目成员编号 | varchar | 50 |
| 10 | new\_project\_member\_names | 新项目成员姓名 | varchar | 20 |
| 11 | change\_date | 变动时间 | datetime |  |
| 12 | flow\_code | 审批流ID | varchar | 50 |
| 13 | organization\_code | 成员所属二级部门编号 | varchar | 50 |
| 14 | organization\_name | 成员所属二级部门名称 | varchar | 50 |
| 15 | change\_category | 变更类别 | varchar | 20 |
| 16 | change\_description | 变更原因 | varchar | 500 |

1. 字段说明：

编号：唯一标识

数据建立时间：不允许为空；

数据建立人：获取当前用户，不允许为空；

数据更新时间：系统自动填写数据更新的时间；

数据更新人：系统自动填写数据更新人；

工程项目ID：工程项目ID；

原项目成员编号：保存修改前数据，不允许为空；

原项目成员姓名：跟随编号；

新项目成员编号：保存修改后数据，不允许为空；

新项目成员姓名：跟随编号

变动时间：不允许为空；

审批流ID：修改信息提交后进入审批；

成员所属二级部门编号：不允许为空，若与项目经理为同一部门不用审批，否则进入审批。

成员所属二级部门名称：跟随编号；

变更类别：可选；

变更原因：手动填写；

1. 操作说明：

点击人员变更按钮，弹出弹框，可对人员进行变更操作，点击提交进入审批。

3） 工程项目经理变更日志实现、保存查看项目经理变更记录的功能

* 1. 添加、编辑功能，新建库表保存填写的任务主数据：construct\_project\_manager\_change\_record

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **字段描述** | **字段类型** | **长度** |
| 1 | id | 编号 | bigint | 19,0 |
| 2 | created\_at | 数据建立时间 | datetime |  |
| 3 | created\_by | 数据建立人 | varchar | 50 |
| 4 | updated\_at | 数据更新时间 | datetime |  |
| 5 | updated\_by | 数据更新人 | varchar | 50 |
| 6 | construct\_project\_id | 工程项目ID | bigint | 19,0 |
| 7 | old\_project\_manager\_code | 原项目经理编号 | varchar | 50 |
| 8 | old\_project\_manager\_name | 原项目经理姓名 | varchar | 20 |
| 9 | new\_project\_manager\_code | 新项目经理编号 | varchar | 50 |
| 10 | new\_project\_manager\_name | 新项目经理姓名 | varchar | 20 |
| 11 | change\_date | 变动时间 | datetime |  |
| 12 | flow\_code | 审批流ID | varchar | 50 |
| 13 | change\_description | 变更原因 | varchar | 500 |

* 1. 字段说明：

编号：主键；

数据建立时间：不允许为空；

数据建立人：获取当前用户，不允许为空；

数据更新时间：系统自动填写数据更新的时间；

数据更新人：系统自动填写数据更新人；

工程项目ID：保存所属项目ID；

原项目经理编号：保存修改前数据，不允许为空；

原项目经理姓名：跟随系统；

新项目经理编号：保存修改后数据，不允许为空；

新项目经理姓名：跟随编号；

变动时间：不允许为空；

审批流ID：变更项目经理提交后，进入审批；

变更原因：手动填写；

* 1. 操作说明：

点击变更按钮进行项目经理的变更，完成后点击提交进入审批。部门总监变更项目经理时不需要审批，部门经理变更项目经理时需审批。

4） 工程项变更日志实现、保存查看工程变更记录的功能

1. 添加、编辑功能，新建库表保存填写的任务主数据：construct\_project\_change\_record

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **字段描述** | **字段类型** | **长度** |
| 1 | id | 编号 | bigint | 19,0 |
| 2 | created\_at | 数据建立时间 | datetime |  |
| 3 | created\_by | 数据建立人 | varchar | 50 |
| 4 | updated\_at | 数据更新时间 | datetime |  |
| 5 | updated\_by | 数据更新人 | varchar | 50 |
| 6 | construct\_project\_id | 工程项目ID | bigint | 19,0 |
| 7 | old\_custom\_service\_manager\_code | 原售后负责人编号 | varchar | 50 |
| 8 | old\_custom\_service\_manager\_name | 原售后负责人姓名 | varchar | 20 |
| 9 | old\_custom\_service\_organization\_code | 原售后负责部门编号 | varchar | 50 |
| 10 | old\_custom\_service\_organization\_name | 原售后负责部门名称 | varchar | 100 |
| 11 | new\_custom\_service\_manager\_code | 新售后负责人编号 | varchar | 50 |
| 12 | new\_custom\_service\_manager\_name | 新售后负责人姓名 | varchar | 20 |
| 13 | new\_custom\_service\_organization\_code | 新售后负责部门编号 | varchar | 50 |
| 14 | new\_custom\_service\_organization\_name | 新售后负责部门名称 | varchar | 100 |
| 15 | old\_construct\_province | 原实施地点省份 | varchar | 20 |
| 16 | old\_construct\_city | 原实施地点城市 | varchar | 20 |
| 17 | new\_construct\_province | 新实施地点省份 | varchar | 20 |
| 18 | new\_construct\_city | 新实施地点城市 | varchar | 20 |
| 19 | old\_project\_description | 原项目介绍 | varchar | 500 |
| 20 | new\_project\_description | 新项目介绍 | varchar | 500 |
| 21 | old\_install\_syatem | 原是否安装运维系统 | tinyint | 3,0 |
| 22 | new\_install\_system | 新是否安装运维系统 | tinyint | 3,0 |
| 23 | old\_install\_data | 原是否涉及数据接入 | tinyint | 3,0 |
| 24 | new\_install\_data | 新是否涉及数据接入 | tinyint | 3,0 |
| 25 | change\_date | 变更时间 | datetime |  |
| 26 | flow\_code | 审批流编号 | varchar | 50 |
| 27 | project\_manager\_code | 项目经理编号 | varchar | 50 |
| 28 | project\_manager\_name | 项目经理姓名 | varchar | 20 |

1. 字段说明：

编号:主键；

数据建立时间：不允许为空；

数据建立人：获取当前用户，不允许为空；

数据更新时间：系统自动填写数据更新的时间；

数据更新人：系统自动填写数据更新人；

工程项目ID：保存所属项目ID；

原售后负责人编号：保存修改前数据；

原售后负责人姓名：跟随编号；

原售后负责部门编号：保存修改前数据；

原售后负责部门名称：跟随编号；

新售后负责人编号：保存修改后的数据；

新售后负责人姓名：保存修改后的数据；

新售后负责部门编号：保存修改后的数据；

新售后负责部门名称：保存修改后的数据；

原实施地点省份：保存修改前的数据；

原实施地点城市：保存修改前的数据；

新实施地点省份：保存修改后的数据；

新实施地点城市：保存修改后的数据；

原项目介绍：保存修改前的数据；

新项目介绍：保存修改后的数据；

原是否安装运维系统：保存修改前的数据；

新是否安装运维系统：保存修改后的数据；

原是否涉及数据接入：保存修改前的数据；

新是否涉及数据接入：保存修改后的数据；

变更时间：保存修改时间；

审批流编号：项目变更后进入审批；

项目经理编号：不允许为空；

项目经理姓名跟随编号；

5）费用增补功能

a)用户在工程项目列表点击费用增补按钮，进入页面，新建库表保存填写的任务主数据：construct\_project\_change\_detail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **字段描述** | **字段类型** | **长度** |
| 1 | id | 编号 | bigint | 19,0 |
| 2 | created\_at | 数据建立时间 | datetime |  |
| 3 | created\_by | 数据建立人 | varchar | 50 |
| 4 | updated\_at | 数据更新时间 | datetime |  |
| 5 | updated\_by | 数据更新人 | varchar | 50 |
| 6 | construct\_project\_id | 工程项目ID | bigint | 19,0 |
| 7 | construct\_badget\_change\_record\_id | 工程费用增补ID | datetime |  |
| 8 | categroy | 费用科目 | varchar | 50 |
| 9 | amount | 本次增补金额 | decimal | 10,2 |
| 10 | memo | 备注 | varchar | 500 |

b)预算费用列表字段有：项目预算费用类型、金额（元）、原因。

项目预算费用类型：通过映射获取；

金额可直接输入；

备注：填写费用增补原因。

1. 费用增补日志：记录费用增补信息。保存至数据表：construct\_budget\_change\_record。日志字段有：工程项目ID、变更日期、项目经理编号、项目经理姓名、本次增补金额、增补原因、附件编号、审批流编号。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **字段描述** | **字段类型** | **长度** |
| 1 | id | 编号 | bigint | 19,0 |
| 2 | created\_at | 数据建立时间 | datetime |  |
| 3 | created\_by | 数据建立人 | varchar | 50 |
| 4 | updated\_at | 数据更新时间 | datetime |  |
| 5 | updated\_by | 数据更新人 | varchar | 50 |
| 6 | construct\_project\_id | 工程项目ID | bigint | 19,0 |
| 7 | change\_date | 变更日期 | datetime |  |
| 8 | project\_manager\_code | 项目经理编号 | varchar | 50 |
| 9 | project\_manager\_name | 项目经理姓名 | varchar | 20 |
| 10 | amount | 本次增补金额 | decimal | 10,2 |
| 11 | description | 增补原因 | varchar | 500 |
| 12 | attact\_code | 附件编号 | varchar | 50 |
| 13 | flow\_code | 审批流编号 | varchar | 50 |

#### 项目评估

1、 工程立项初步评估功能：

a、项目经理在项目列表中点击评估按钮，可根据项目概况、查看履行交付要求、设备清单为依据对项目进行评估。

b、项目评估决策有四个，分别是：

评估通过、暂停启动、不满足启动条件、无法实施。

c、工程项目评估记录表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **字段描述** | **字段类型** | **长度** |
| 1 | id | 编号 | bigint | 19,0 |
| 2 | created\_at | 数据建立时间 | datetime |  |
| 3 | created\_by | 数据建立人 | varchar | 50 |
| 4 | updated\_at | 数据更新时间 | datetime |  |
| 5 | updated\_by | 数据更新人 | varchar | 50 |
| 6 | construct\_project\_id | 工程项目ID | bigint | 19,0 |
| 7 | evaluater\_code | 评估人编号 | varchar | 50 |
| 8 | evaluater\_name | 评估人姓名 | varchar | 20 |
| 9 | evaluate\_result | 评估结果 | varchar | 20 |
| 10 | evaluate\_date | 评估时间 | datetime |  |
| 11 | flow\_code | 审批流编号 | varchar | 50 |

d、字段说明：

编号：主键

数据建立时间：不允许为空；

数据建立人：不允许为空；

数据更新时间：系统自动填写数据更新的时间；

数据更新人：系统自动填写数据更新人；

工程项目ID：所属项目，不允许为空；

评估人编号：保存评估人编号；

评估人姓名：跟随编号；

评估结果：保存评估结果；

评估时间：保存评估时间；

审批流编号：评估完成后进入审批；

2、 工程项目评估暂停启动表：construct\_project\_evaluate\_pause\_detail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **字段描述** | **字段类型** | **长度** |
| 1 | id | 编号 | bigint | 19,0 |
| 2 | created\_at | 数据建立时间 | datetime |  |
| 3 | created\_by | 数据建立人 | varchar | 50 |
| 4 | updated\_at | 数据更新时间 | datetime |  |
| 5 | updated\_by | 数据更新人 | varchar | 50 |
| 6 | construct\_project\_id | 工程项目ID | bigint | 19,0 |
| 7 | construct\_evaluate\_record\_id | 工程项目评估记录ID | bigint | 19,0 |
| 8 | project\_category | 项目类型 | varchar | 50 |
| 9 | manager\_code | 负责人编号 | varchar | 50 |
| 10 | manager\_name | 负责人姓名 | varchar | 20 |
| 11 | finish\_date | 完成时间 | datetime |  |
| 12 | description | 说明 | varchar | 500 |

1. 操作说明：项目经理点击暂停启动按钮，填写项目类型、负责人、完成时间、说明。提交后进入审批流。

3 、工程项目评估不满足实施条件详情表：

construct\_project\_evaluate\_no\_condition\_detail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **字段描述** | **字段类型** | **长度** |
| 1 | id | 编号 | bigint | 19,0 |
| 2 | created\_at | 数据建立时间 | datetime |  |
| 3 | created\_by | 数据建立人 | varchar | 50 |
| 4 | updated\_at | 数据更新时间 | datetime |  |
| 5 | updated\_by | 数据更新人 | varchar | 50 |
| 6 | construct\_project\_id | 工程项目ID | bigint | 19,0 |
| 7 | construct\_evaluate\_record\_id | 工程项目评估记录ID | bigint | 19,0 |
| 8 | category | 类型 | varchar | 50 |
| 9 | description | 说明 | varchar | 500 |

a、操作说明：项目经理点击不满足条件按钮，填写类型及说明。提交后进入审批流。

4 、工程项目评估无法实施表：construct\_project\_evaluate\_disable\_detail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **字段描述** | **字段类型** | **长度** |
| 1 | id | 编号 | bigint | 19,0 |
| 2 | created\_at | 数据建立时间 | datetime |  |
| 3 | created\_by | 数据建立人 | varchar | 50 |
| 4 | updated\_at | 数据更新时间 | datetime |  |
| 5 | updated\_by | 数据更新人 | varchar | 50 |
| 6 | construct\_project\_id | 工程项目ID | bigint | 19,0 |
| 7 | construct\_evaluate\_record\_id | 工程项目评估记录ID | bigint | 19,0 |
| 8 | attach\_code | 附件 | varchar | 50 |
| 9 | description | 原因 | varchar | 500 |

a、操作说明：项目经理点击无法实施按钮，填写原因并上传附件。提交后进入审批流。

#### 里程碑管理

1. 制定实施计划：

项目经理在评估页面点击评估通过后，弹出实施计划制定页面。填写实施计划后，将信息保存至里程碑数据表。

1. 工程项目里程碑表：construct\_milestone

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **字段描述** | **字段类型** | **长度** |
| 1 | id | 编号 | bigint | 19,0 |
| 2 | created\_at | 数据建立时间 | datetime |  |
| 3 | created\_by | 数据建立人 | varchar | 50 |
| 4 | updated\_at | 数据更新时间 | datetime |  |
| 5 | updated\_by | 数据更新人 | varchar | 50 |
| 6 | construct\_project\_id | 工程项目ID | bigint | 19,0 |
| 7 | milestone\_name | 里程碑名称 | varchar | 50 |
| 8 | milestone\_no | 里程碑编号 | int | 10,0 |
| 9 | deliveries | 交付物 | varchar | 200 |
| 10 | milestone\_manager\_code | 负责人编号 | varchar | 50 |
| 11 | milestone\_manager\_name | 负责人姓名 | varchar | 20 |
| 12 | progress | 里程碑进度 | int | 10,0 |
| 13 | plan\_finish\_date | 计划完成时间 | datetime |  |
| 14 | promise\_finish\_date | 承诺完成时间 | datetime |  |
| 15 | plan\_start\_date | 计划开始时间 | datetime |  |
| 16 | actual\_start\_date | 开始时间 | datetime |  |
| 17 | actual\_finish\_date | 完成时间 | datetime |  |
| 18 | contract\_requirement\_finish\_date | 合同要求完成时间 | datetime |  |
| 19 | contract\_requirement\_income\_ratio | 合同确认收入比例 | decimal | 10,2 |
| 20 | report\_sign\_date | 报告签署日期 | datetime |  |
| 21 | report\_requirement\_upload\_date | 报告原件应提交日期 | datetime |  |
| 22 | milestone\_state | 里程碑状态 | varchar | 20 |
| 23 | contract\_actual\_income\_ratio | 实际确认收入比例 | decimal | 10,2 |
| 24 | file\_upload\_state | 原件提交状态 | varchar | 20 |
| 25 | upload\_code | 附件编号 | varchar | 50 |
| 26 | memo | 备注 | varchar | 500 |
| 27 | flow\_code | 里程碑审批流编号 | varchar | 50 |
| 28 | report\_actual\_upload\_date | 报告原件实际提交日期 | varchar | 0 |

1. 字段说明：

编号：主键；

数据建立时间：不允许为空；

数据建立人：不允许为空；

数据更新时间：系统自动填写数据更新的时间；

数据更新人：系统自动填写数据更新人；

工程项目ID：所属项目，不允许为空；

里程碑名称：由API获取里程碑名称，不允许为空；

里程碑编号：由API获取里程碑序号，不允许为空；

交付物：需交付的文件；

负责人编号：允许为空，默认为项目经理；

负责人姓名：跟随编号；

里程碑进度：里程碑进度通过API获取；

计划完成时间：手动填写；

承诺完成时间：实际填写时间；

计划开始时间：手动填写；

开始时间：手动填写；

完成时间：手动填写；

合同要求完成时间：根据里程碑编号通过API获取

合同确认收入比例：通过API获取；

报告签署日期：通过API获取；

报告原件应提交日期：通过API获取；

里程碑状态：通过API获取；

实际确认收入比例：留空；

原件提交状态：通过API获取；

附件编号：通过API获取；

备注：手动填写；

里程碑审批流编号：里程碑提交后进入审批。

报告原件实际提交日期：通过API获取；

2、里程碑变更：

可变更字段有负责人、承诺时间变更、合同日期、里程碑新增、删除、关闭。

#### 工程移交

1、项目移交申请功能：

a、项目经理在工程项目列表中点击“项目移交”按钮，进入提交移交申请页面。

b、申请时需填写的字段：

移交单号：规则生成GCYJ-201601-xxxx、

省份、城市、客户名称、项目编号、项目名称、项目启动时间、项目经理、联系方式、申请人。（以上由工程立项表获取）

移交内容：申请时填写

申请时间：当前时间

验收时间：空

期望移交时间：当前时间加7天

完工时间：完工里程碑的时间（5号里程碑，），通过API获取，否则为当前时间

项目当前状态：项目经理自动选择

移交实施工程师：申请时填写，选择范围工程中心；

移交实施工程师联系方式：根据移交实施工程师获取；

移交售后收尾工程师：申请时先不填写，由售后工程师填写

遗留问题描述，遗留问题主要包括六大块，分别是：“软件重大故障、出现3次以上的故障、性能风险”、“合同内未完成工作内容”、“损坏2次以上的硬件设备”、“机房环境隐患”、“未闭环的客户环境、需求”、“工程项客户承诺的其他重要事项”。

售后服务中心确认表中，须填写的字段有审核人员、审核结果、指定售后工程师、移交售后工程师联系方式、审批意见、预计到达现场时间、工程是否同意到达时间点。

项目移交确认单须填写字段：检查项目、是否合格、备注。

移交收尾工作表中须填写字段：知会销售单元总经理，知会销售单元大区经理、是否向客户汇报、客户汇报时间、客户姓名、客户联系方式、客户意见、移交运维设备IP地址、运维系统负责人修改时间、400问题跟踪人修改时间。

点击“提交”按钮将项目移交申请单发至售后服务中心确认，各领导审核。

c、项目移交申请保存至数据表：construct\_project\_transfer\_record

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **字段描述** | **字段类型** | **长度** |
| 1 | id | 编号 | bigint | 19,0 |
| 2 | created\_at | 数据建立时间 | datetime |  |
| 3 | created\_by | 数据建立人 | varchar | 50 |
| 4 | updated\_at | 数据更新时间 | datetime |  |
| 5 | updated\_by | 数据更新人 | varchar | 50 |
| 6 | construct\_project\_id | 工程项目ID | bigint | 19,0 |
| 7 | transfer\_content | 移交内容 | varchar | 500 |
| 8 | apply\_date | 申请日期 | datetime |  |
| 9 | check\_date | 验收日期 | datetime |  |
| 10 | expect\_transfer\_date | 期望移交日期 | datetime |  |
| 11 | construct\_finish\_date | 完工日期 | datetime |  |
| 12 | project\_state | 项目当前状态 | varchar | 20 |
| 13 | project\_manager\_phone | 项目经理联系方式 | varchar | 20 |
| 14 | transfer\_manager\_code | 移交实施工程师编号 | varchar | 50 |
| 15 | transfer\_manager\_name | 移交实施工程师姓名 | varchar | 20 |
| 16 | transfer\_manager\_phone | 移交实施工程师联系方式 | varchar | 20 |
| 17 | transfer\_end\_manager\_code | 移交售后收尾工程师编号 | varchar | 50 |
| 18 | transfer\_end\_manager\_name | 移交售后收尾工程师姓名 | varchar | 20 |
| 19 | custom\_service\_manager\_code | 售后工程师编号 | varchar | 50 |
| 20 | custom\_service\_manager\_name | 售后工程师姓名 | varchar | 20 |
| 21 | custom\_service\_checker\_code | 售后服务中心审核人员编号 | varchar | 50 |
| 22 | custom\_service\_checker\_name | 售后服务中心审核人员姓名 | varchar | 20 |
| 23 | flow\_code | 工程项目移交审批编号 | varchar | 50 |
| 24 | attach\_code | 整改意见附件编号 | varchar | 50 |
| 25 | transfer\_result | 移交结果 | tinyint | 3,0 |
| 26 | transfer\_result\_content | 移交意见 | varchar | 500 |

#### 工程结项

1. 项目经理点击工程项目列表的“项目结项”按钮进入此页面。
2. 更新的数据保存至表：construct\_project\_finisth\_report

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **字段描述** | **字段类型** | **长度** |
| 1 | id | 编号 | bigint | 19,0 |
| 2 | created\_at | 数据建立时间 | datetime |  |
| 3 | created\_by | 数据建立人 | varchar | 50 |
| 4 | updated\_at | 数据更新时间 | datetime |  |
| 5 | updated\_by | 数据更新人 | varchar | 50 |
| 6 | construct\_project\_id | 工程项目ID | bigint | 19,0 |
| 7 | project\_manager\_code | 项目经理编号 | varchar | 50 |
| 8 | project\_manager\_name | 项目经理姓名 | varchar | 20 |
| 9 | apply\_date | 申请时间 | datetime |  |
| 10 | description | 状态变更概述 | varchar | 500 |
| 11 | attach\_code | 附件编号 | varchar | 50 |
| 12 | flow\_code | 结项审批流编号 | varchar | 50 |

1. 填写结项报告信息，项目结项报告显示字段有：项目名称、项目编号、项目经理、申请时间、结项说明、附件。
2. 结项说明：编辑文字描述，输入框高度固定。输入框显示不全文字时自动出现滚动条。选择附件，默认“选择文件”按钮后显示“未选择任何文件”提示。点击“选择文件”按钮，上传文件，每一行显示一条文件信息，文件名称和文件大小。点击“提交”按钮，返回工程项目列表，显示“项目结项报告已提交”提示信息，显示3秒自动消失。

### 验收项目

#### 工程立项

### 售后项目

#### 工程立项

### 催款项目

#### 工程立项

# 第零层设计

## 系统与其他系统的关系

工程售后项目属于锐安新业务系统的一部分，为其中关于工程、验收、售后、催款部分的相关模块

## 设计思路（可选）

### 设计方法

本项目采用使用基于面向对象的构件化软件开发方法，使用面向服务的架构（SOA）作为组件模型，技术选型上使用目前流行的多种web技术，包括Spring MVC4.0+, MyBatis, Apache Shiro, ehcache, Jquery ,BootStrap, WebSocket 等等，支持多种数据库MySQL, Oracle, sqlserver等。 分层设计：使用分层设计，分为dao，service，Controller，view层，层次清楚，低耦合，高内聚。 安全考虑：严格遵循了web安全的规范，前后台双重验证，参数编码传输，密码md5加密存储，shiro权限验证，从根本上避免了SQL注入，XSS攻击，CSRF攻击等常见的web攻击手段。

整个工程售后项目划分为如下几个模块：

1. 工程项目管理

Construct

2）验收项目管理

Check

3）售后项目管理

CustomService

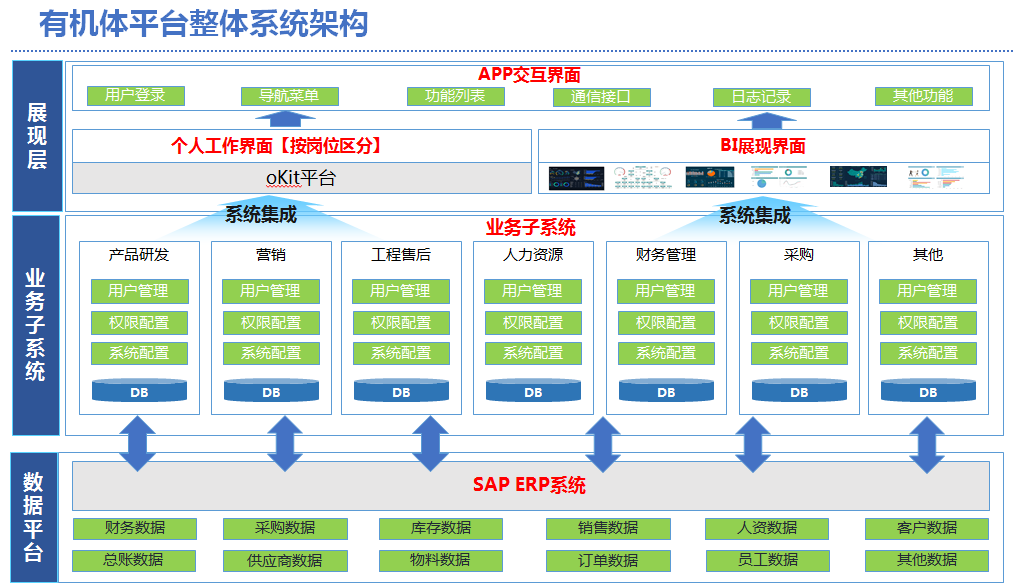
4）催款项目管理

Payment

### 设计可选方案

#### 系统整体技术方案

因本项目为锐安新业务系统的一部分，固采用锐安整体系统技术方案，方案如下：



本项目总体架构图如上图所示，由下至上分为三层：数据平台层、业务子系统以及应用展现层。

**一、数据平台层**

SAP ERP平台作为整个有机体平台的数据平台层，用来作为所有业务的数据存储。有机体平台所有的业务操作凭证、数据通过SAP 标准接口进行无缝集成。

**二、业务子系统层**

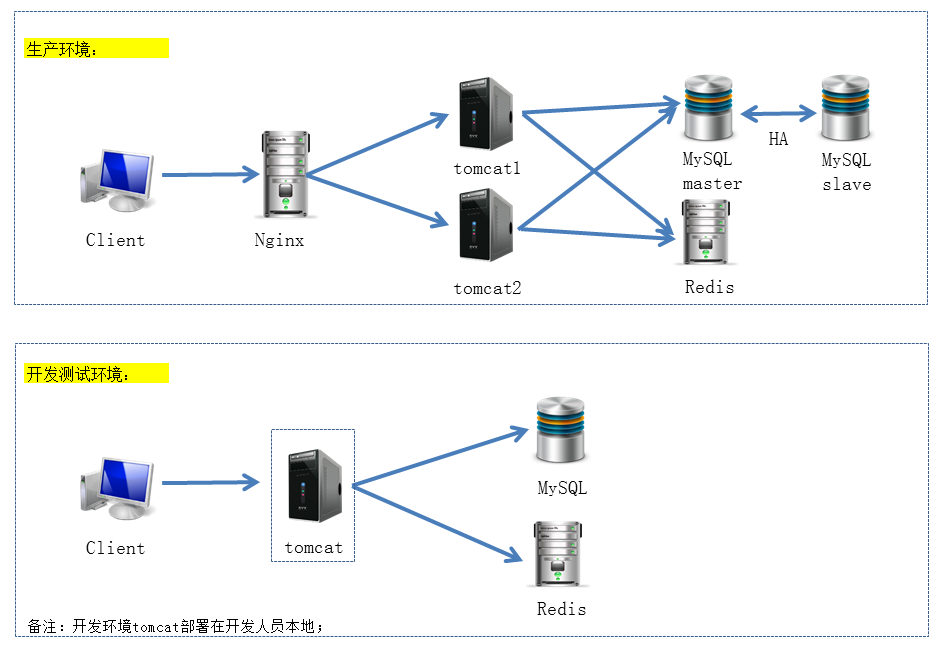
有机体平台包含各个ERP管理模块，产品研发、销售管理、工程售后管理、人力资源管理、财务管理、采购管理等等。

各个子系统之间共用底层统一的组织架构、岗位、权限管理管理模块、SSO单点登录集成，使用户极大提高工作效率。

**三、系统展现层**

系统展现层包含个人工作界面（按岗位区分）、BI展现界面与APP交互界面三大部分。

#### 系统整体部署方案



### 设计约束

为实现本项目的建设目标，本项目的总体设计原则包含但不限于以下内容：

1、面向业务：以业务需求为主导，确保项目建设的针对性、实用性和有效性；

2、面向数据：以数据为核心，加强数据设计的科学性和实用性，促进业务功能实现和信息共享。

3、面向用户：以人为本，充分考虑本系统各类用户的访问方式和访问界面的方便性和友好性需求。

4、实用性：实用性是一个系统的基本要求，同时也是最高要求。一个系统是否实用，就是指它的经济性和有效性，简单地说，就是这个信息系统是否有优良的性价比，以需求为先导，增强系统设计的实用性。

5、先进性：除了实用性之外，还要考虑系统的先进性。采用先进的网络信息技术、应用平台和开发工具，采用模块化、构件化以及面向对象的设计方法，保证系统设计的先进性，降低系统出现设计问题的概率，提高系统建设的效率，增强系统的可扩展性

6、可靠性：系统应该是可靠的，在出现异常的时候应该有人性化的异常信息方便用户理解原因，或采取适当的应对方案，确保系统运行可靠，尽量减少系统出现故障概率。

7、可扩展性：系统需要具备一定的适应能力，特别是Web应用要能适应于多种运行环境，来应对未来变化的环境和需求。可扩展性主要体现在系统易于扩展，例如可以采用分布式设计、系统结构模块化设计，系统架构可以根据网络环境和用户的访问量而适时调整，从某种程度上说，这也是系统的适应性。

8、安全性：实现系统安全和数据安全。系统安全：根据系统的各种权限级别保证不同用户高效、快速地访问控制授权范围内的系统资源；数据安全：使用用户授权管理系统对敏感数据信息加密后进行传输、存放。

9、整合性、开放性：在本项目系统设计时将充分考虑与家园系统整体设计规划相符合，同时也考虑具有一定的开放性，把握好信息共享和资源整合。

# **第一层设计**

## 分解描述

### 模块分解

#### 工程项目管理模块分解

主要包括工程项目部分，各个功能模块如下：

|  |  |  |
| --- | --- | --- |
| 系统模块 | 功能项 | 功能子项 |
| 项目管理 | 项目管理 | 自动立项接口 |
| 手动立项 |
| 立项审批 |
| 项目列表 |
| 项目详情 |
| 项目挂起/启动 |
| 项目变更 |
| 项目变更审批 |
| 费用增补 |
| 费用增补审批 |
| 项目经理变更 |
| 项目经理变更审批 |
| 项目成员变更 |
| 项目成员变更审批 |
| 项目评估 | 评估通过 |
| 暂停启动 |
| 不满足启动条件 |
| 无法实施 |
| 评估审批 |
| 里程碑管理 | 里程碑详情 |
| 里程碑承诺完成时间变更 |
| 里程碑完成时间变更审批 |
| 里程碑负责人变更 |
| 里程碑负责人变更审批 |
| 里程碑备注修改 |
| 里程碑收入比例变更 |
| 里程碑关闭 |
| 里程碑关闭审批 |
| 工程移交 | 项目移交申请 |
| 项目移交审批 |
| 工程结项 | 项目结项申请 |
| 项目结项审批 |

包内结构图如下：



系统用例图如下：



#### 验收项目管理模块分解

主要包括验收项目部分，各个功能模块如下：

#### 售后项目管理模块分解

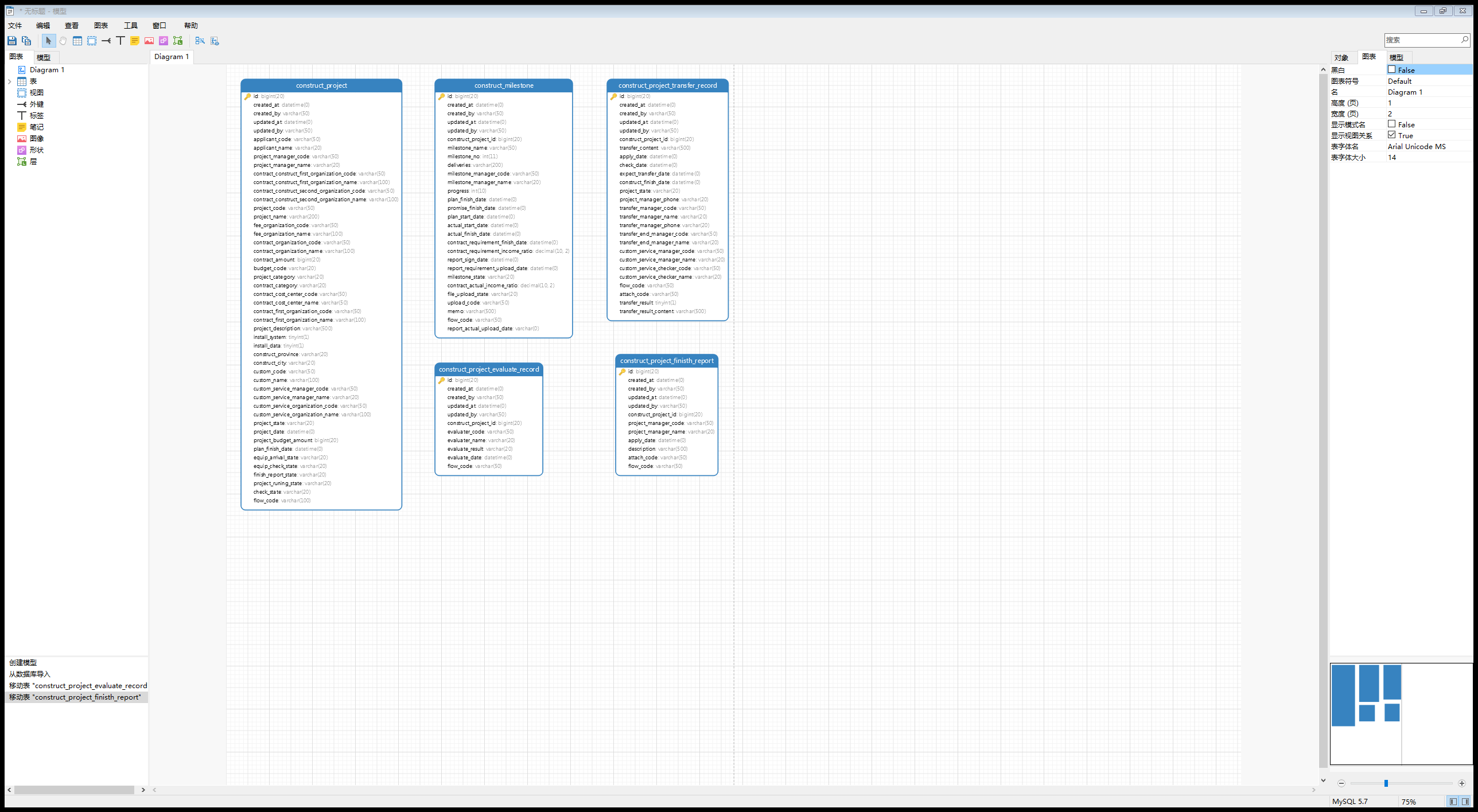
主要包括售后项目部分，各个功能模块如下：

#### 催款项目管理模块分解

主要包括催款项目部分，各个功能模块如下：

### 数据分解

#### 工程项目管理



## 依赖性描述

所有模块依赖于锐安基础平台，模块间无相互依赖关系。

### 运行设计

基础模块部分由锐安平台提供。本项目只需要实现业务功能即可。

## 接口描述

### 模块接口

锐安提供的模块接口如下，本项目相关接口详见接口文档。

#### YJT\_PMS提供的接口

|  |
| --- |
| /\*\*  \* 获取部门列表  \* **@param** deptId  \* **@return**  \*/  **public** String getDeptList();      /\*\*  \* 根据部门或用户名查用户  \* **@param** deptId  \* **@param** userName  \* **@return**  \*/  **public** String getUserList(Long deptId, String username);    /\*\*  \* 通过用户名取用户信息  \* 用于登录后，通过用户名取用户信息存入在session中  \* **@param** username  \* **@return**  \*/  **public** String getUserInfoByUsername(String username);    /\*\*  \* 通过用户ID查询用户信息  \* 提供给TMS的接口  \* **@param** username  \* **@return**  \*/  **public** String getUserInfoByUserId(Long userId);    /\*\*  \* 通过部门ID获取部门信息（包括群、一级、二级部门信息）  \* **@param** deptId  \* **@return**  \*/  **public** String getDeptInfoByDeptId(Long deptId);    /\*\*  \* 通过用户ID取下属用户信息  \* 用户信息包含：部门信息、职位信息  \* **@param** userId 用户ID  \* **@param** userStatus 用户状态，传null查所有  \* **@return**  \*/  **public** String getSubUserList(Long userId, Integer userStatus);    /\*\*  \* 获取职位列表  \* **@return**  \*/  **public** String getDutyList(); |

#### YJT\_Admin提供的接口

|  |
| --- |
| /\*\*  \* 插入日志  \* **@param** tableName  \* **@param** data  \* **@return**  \*/  **public** **int** logInsert(String tableName, Map<String, Object> param);    /\*\*  \* 查询日志  \* **@param** tableName  \* **@param** start  \* **@param** rows  \* **@param** condition  \* **@return**  \*/  **public** String logQuery(String tableName, **int** start, **int** rows, String condition);      /\*\*  \* 查询消息  \*  \* **@param** id 消息ID  \* **@return** 消息  \*/  **public** Message messageSelectById(Long id);    /\*\*  \* 查询消息  \*  \* **@param** Message 消息  \* **@return** 消息列表  \*/  **public** List<Message> messageSelectList(Message message);    /\*\*  \* 新增消息  \*  \* **@param** Message 消息  \* **@return** 结果  \*/  **public** **int** messageInsert(Message message);    /\*\*  \* 修改消息  \*  \* **@param** Message 消息  \* **@return** 结果  \*/  **public** **int** messageUpdate(Message message);    /\*\*  \* 获取用户数据  \* **@param** username  \* **@return**  \* {  "menu": "",  "dept": {  "id": 1532592468840,  "deptName": "开发一部",  "firstDeptCode": "A08",  "secondDeptCode": "B01",  "secondDeptName": "开发一部",  "secondDeptId": 1532592468840,  "firstDeptName": "开发中心",  "deptCode": "B01",  "firstDeptId": 1532586650528  },  "user": {  "id": 111,  "deptName": "开发一部",  "username": "杨帆",  "postName": "Java开发工程师",  "deptId": 1532592468840,  "postId": 1532672796140  },  "auth": ""  }  \*/  **public** String getUserDataForSession(String username); |

#### YJT\_Activiti提供的接口

|  |
| --- |
| /\*\*  \* 启动流程  \* **@param** processDefinitionId 流程定义ID  \* **@param** userId 发起人ID  \* **@param** variables 变量  \* **@return** 流程实例ID  \*/  **public** String launchProcess(String processDefinitionId, String userId, Map<String, Object> variables);    /\*\*  \* 处理流程  \* **@param** taskId  \* **@return**  \*/  **public** **void** handleProcess(String taskId);    /\*\*  \* 查询待办列表  \* **@param** userId  \* **@return**  \*/  **public** String queryTodoList(String userId);    /\*\*  \* 查询流程定义列表  \* **@return**  \*/  **public** String queryProcessList( Map<String, Object> param); |

### 进程接口

无

# **第二层设计**

## 工程项目管理

### 分解描述

#### 模块分解

功能子项详见上层设计

### 数据实体描述



### 依赖性描述

各个实体依赖于工程项目实体，各功能实体间无依赖关系

### 接口描述

详见接口文档工程项目部分

## 项目评估

### 分解描述

#### 模块分解

功能子项详见上层设计

### 数据实体描述



### 依赖性描述

整个模块实体依赖于项目实体，模块实体内无依赖

### 接口描述

无

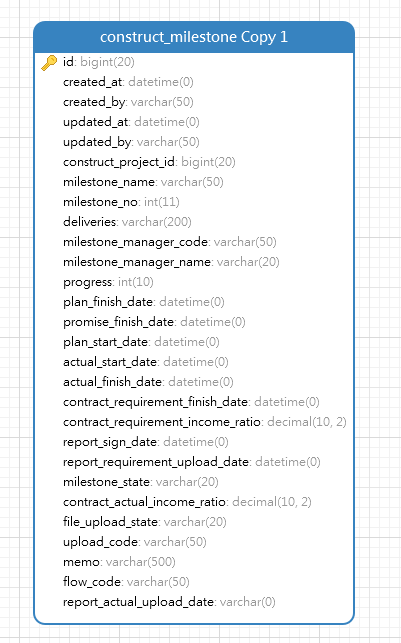
## 里程碑管理

### 分解描述

#### 模块分解

功能子项详见上层设计

### 数据实体描述



### 依赖性描述

整个模块实体依赖于项目实体，模块实体内无依赖

### 接口描述

无

## 工程移交

### 分解描述

#### 模块分解

功能子项详见上层设计

### 数据实体描述



### 依赖性描述

整个模块实体依赖于项目实体，模块内除工程移交记录外，其他实体均依赖于工程移交记录实体，且相互间互不依赖

### 接口描述

无

## 工程结项

### 分解描述

#### 模块分解

功能子项详见上层设计

### 数据实体描述



### 依赖性描述

整个模块实体依赖于项目实体

### 接口描述

无

# **数据库**

**该处涉及相关设计统一在数据库设计说明书中体现**