

哈尔滨工业大学 计算学部  
2024 年秋季学期《开源软件开发实践》  
Lab5 基于低代码平台的软件开发

学号	姓名	联系方式
2022210939	岳翼博	19716318612

## 目 录

1	实验要求.....	1
2	掌握 inBuilder 低代码开发平台的使用.....	1
3	利用 inBuilder 开发软件.....	9
3.1	场景描述和功能设计.....	9
3.2	界面截图.....	11
3.3	Demo 视频.....	15
4	***熟悉 UBML 项目 .....	16
5	小结.....	16

# 1 实验要求

1. 掌握典型低代码平台 inBuilder 的使用。
2. 能够利用 inBuilder 平台进行软件产品的开发。

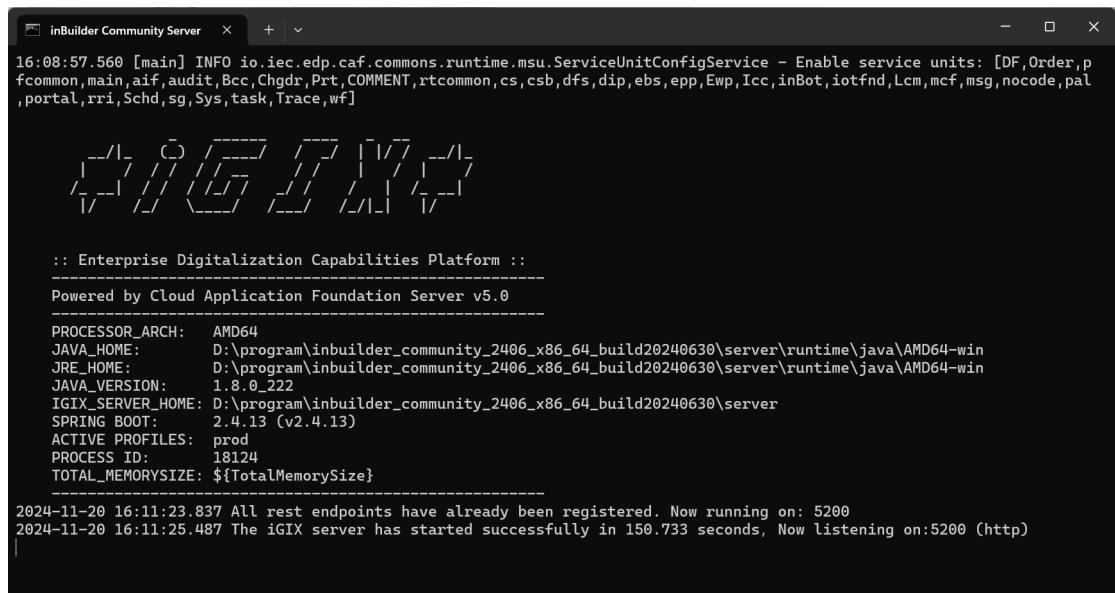
## 2 掌握 inBuilder 低代码开发平台的使用

给出各个任务完成后的主要界面截图。

### 2.1 入门开发

#### 1. 开发环境搭建

##### 1) 开发环境 windows 搭建



```
16:08:57.560 [main] INFO io.iec.edp.caf.common.runtime.msu.ServiceUnitConfigService - Enable service units: [DF,Order,pfcommon,main,aif,audit,Bcc,Chgr,Prt,COMMENT,rtcommon,cs,csb,dfs,dip,ebs,epp,Ewp,Icc,inBot,iotfnd,Lcm,mcf,msg,noCode,pal,portal,rri,Schd,sg,Sys,task,Trace,wf]

:: Enterprise Digitalization Capabilities Platform ::

Powered by Cloud Application Foundation Server v5.0

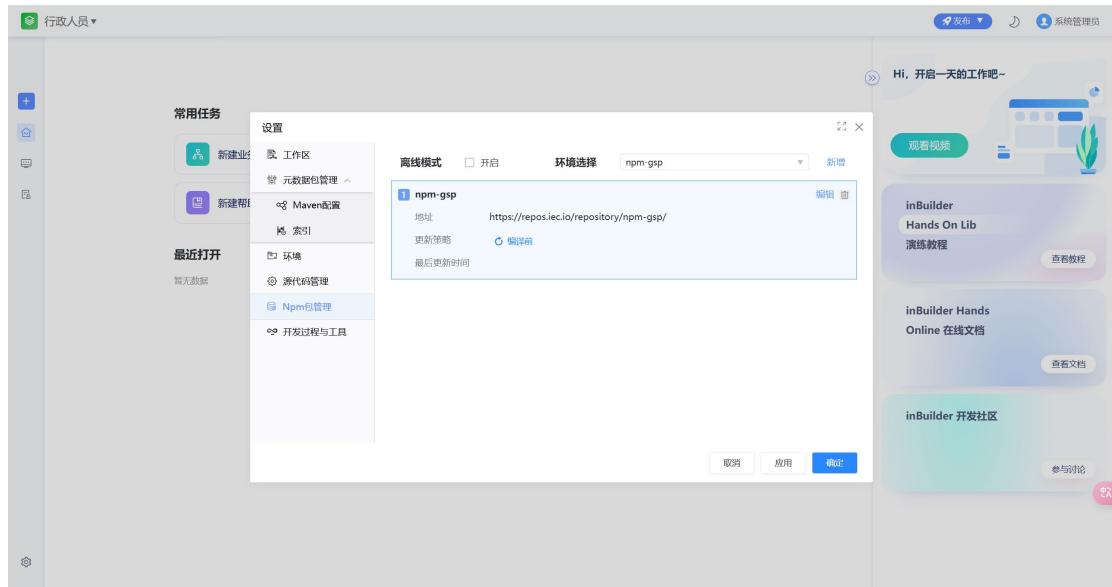
PROCESSOR_ARCH: AMD64
JAVA_HOME: D:\program\inbuilder_community_2406_x86_64_build20240630\server\runtime\java\AMD64-win
JRE_HOME: D:\program\inbuilder_community_2406_x86_64_build20240630\server\runtime\java\AMD64-win
JAVA_VERSION: 1.8.0_222
IGIX_SERVER_HOME: D:\program\inbuilder_community_2406_x86_64_build20240630\server
SPRING_BOOT: 2.4.13 (v2.4.13)
ACTIVE_PROFILES: prod
PROCESS_ID: 18124
TOTAL_MEMORYSIZE: ${TotalMemorySize}

2024-11-20 16:11:23.837 All rest endpoints have already been registered. Now running on: 5200
2024-11-20 16:11:25.487 The iGIX server has started successfully in 150.733 seconds, Now listening on:5200 (http)
```

##### 2) 用户初始

创建了一个名为 yyb 的开发用户

##### 3) 配置初始

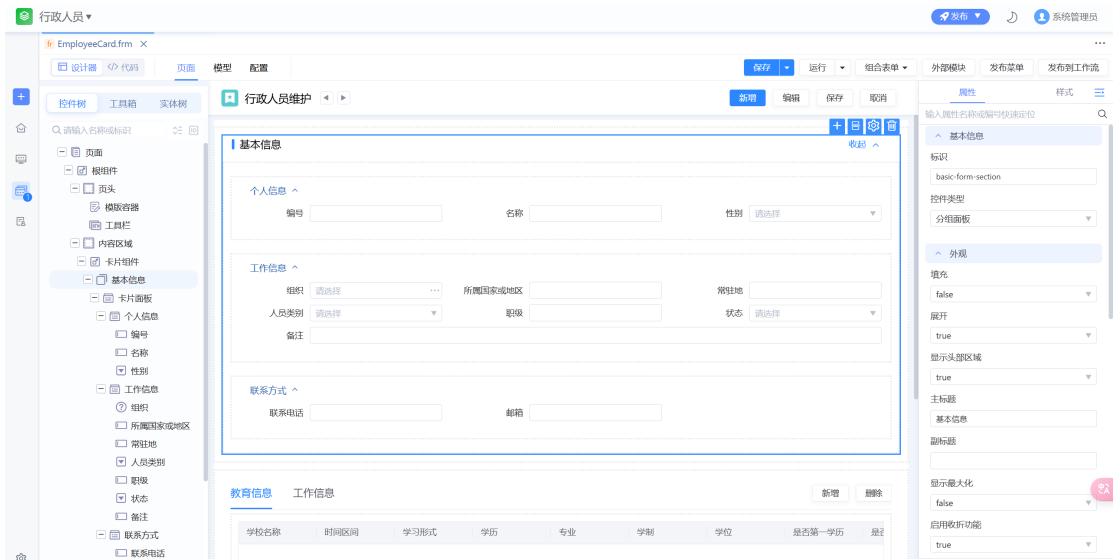


## 2. 五分钟上手 inBuilder 开发

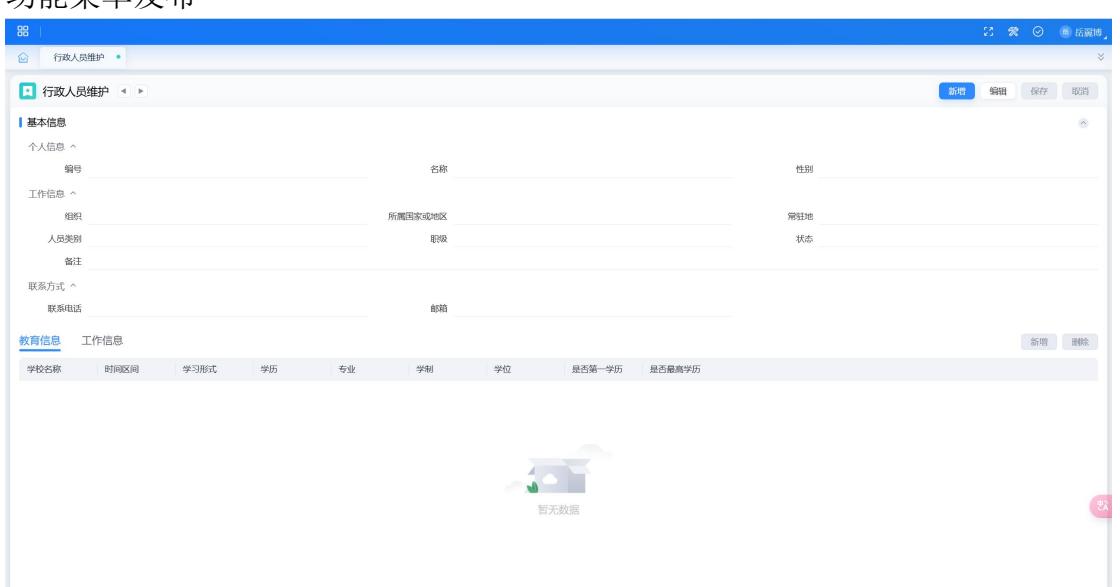
### 1) 实体建模

The screenshot shows the inBuilder entity modeling interface for the 'Employee' entity. The left sidebar lists '设计' (Design) and '代码' (Code). The main area has tabs for '业务实体' (Business Entity), '方法' (Methods), '事件' (Events), and '设置' (Settings). The '业务实体' tab is selected, showing a table of fields for the 'Employee' entity. The table includes columns for序号 (Index), 编号 (Code), 名称 (Name), 数据类型 (Data Type), 长度 (Length), and 对象类型 (Object Type). Fields listed include ID, Version, Code, Name, Remark, State, Organization, Gender, CountryOrArea, Location, PostID, Category, DisplayName, Telephone, Secret, and Mailbox. On the right, there are sections for '实体属性' (Entity Properties) and '高级属性' (Advanced Properties), which are currently empty.

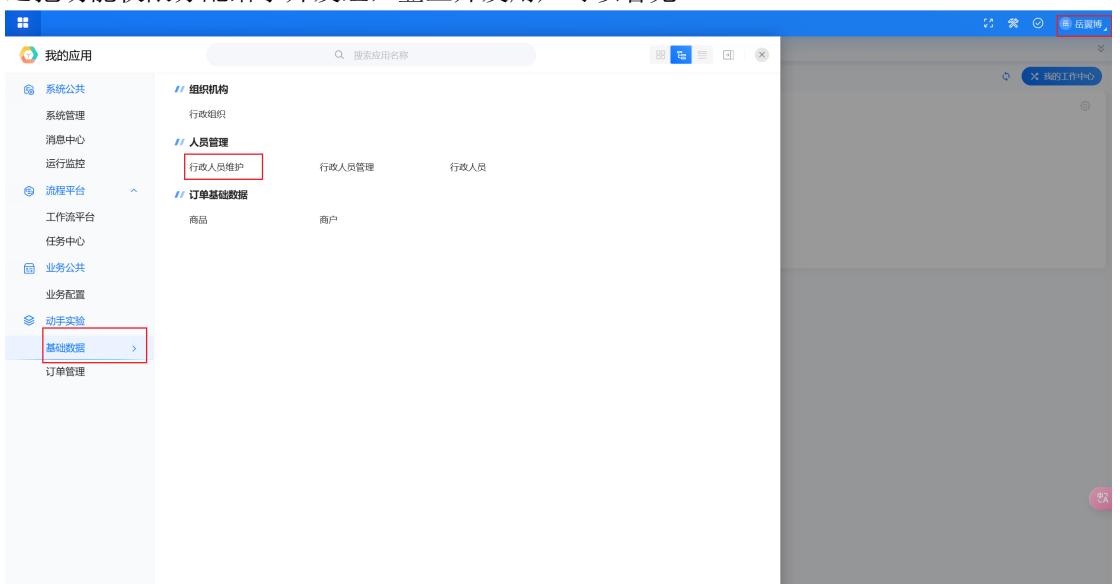
### 2) 前端页面建模



### 3) 功能菜单发布



还把功能权限分配给了开发组，登上开发用户可以看见



## 2.2 进阶开发

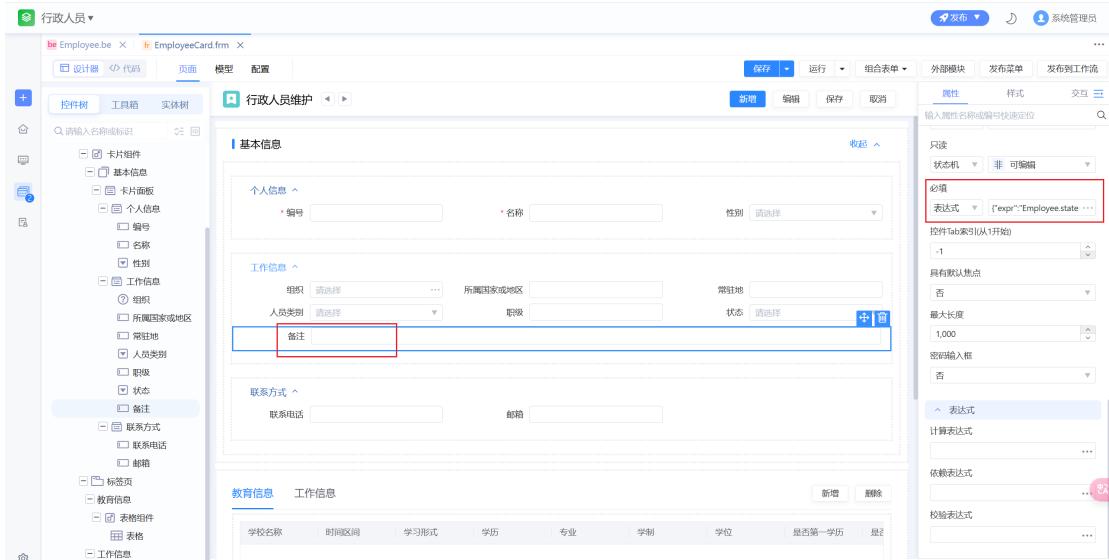
### 1. 字段属性配置

The screenshot shows the 'EmployeeCard.frm' form design interface. In the center, there is a table listing fields with their properties. The 'Organization' field is selected, highlighted in blue. The right side of the screen displays detailed properties for this field, including its type (Text), length (36), and association (Relationship). Other fields listed include ID, Version, Code, Name, Remark, State, PostID, Category, DisplayName, Telephone, Secret, and Mailbox.

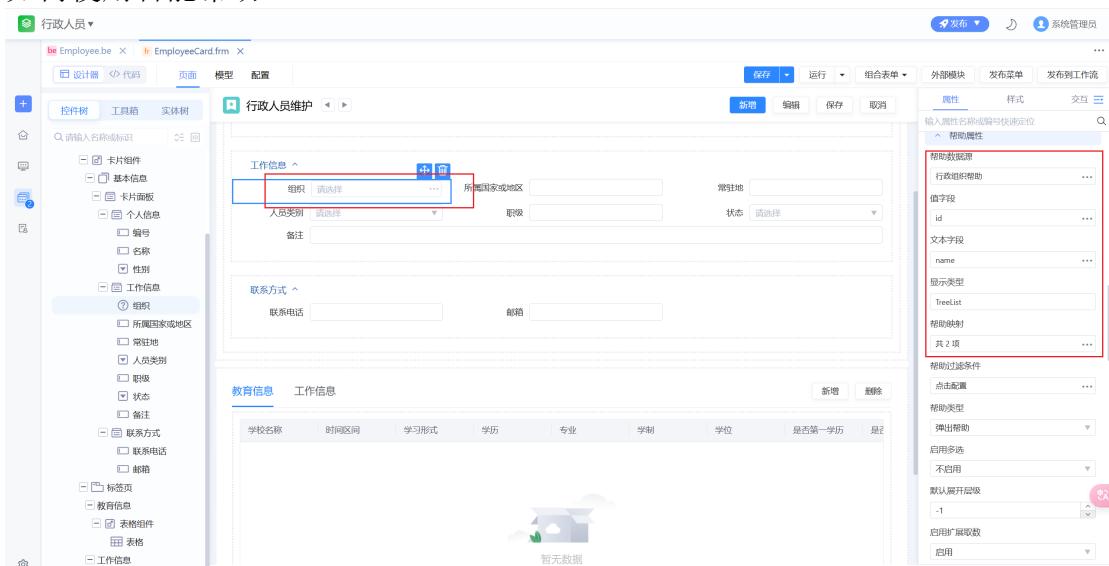
### 2. 如何设置字段分组

The screenshot shows the 'EmployeeCard.frm' form configuration interface. On the left, a sidebar lists various field categories like '基本信息', '工作信息', '教育信息', etc. The main area displays a form with grouped fields. The first group, '基本信息', contains fields for '编号' (Number), '姓名' (Name), and '性别' (Gender). The second group, '工作信息', contains fields for '组织' (Organization), '所属国家或地区' (Country/Region), '常驻地' (Residence), '人员类别' (Person Type), '职级' (Job Level), and '状态' (Status). Below these groups, there is a '联系方式' (Contact Information) section with fields for '联系电话' (Phone) and '邮箱' (Email). The right side of the screen shows the properties panel for the '基本信息' group, including settings for visibility, behavior, and styling.

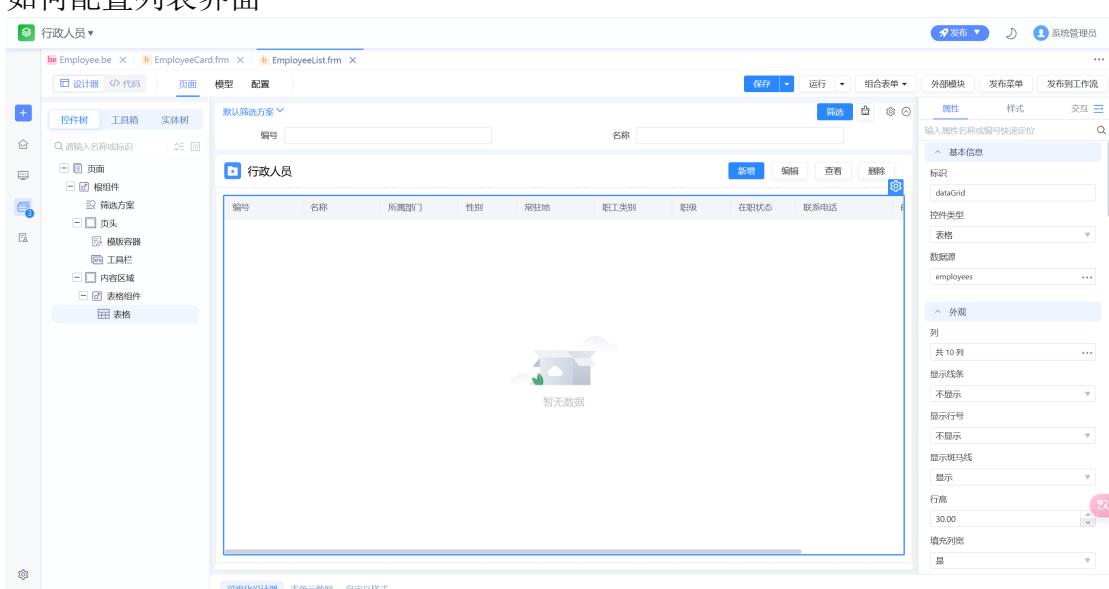
### 3. 如何配置必填表达式



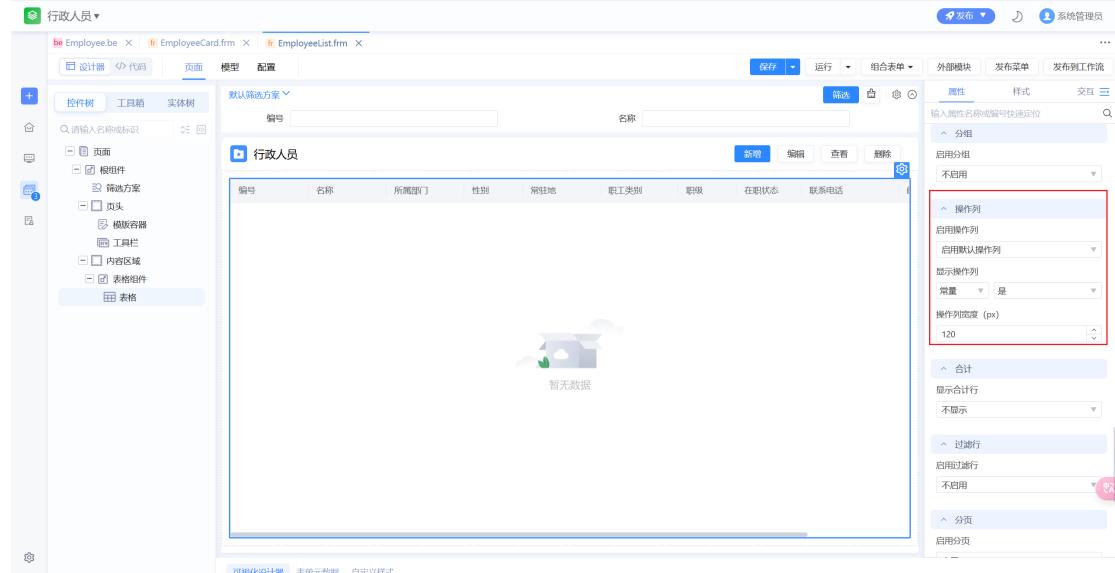
#### 4. 如何使用智能帮助



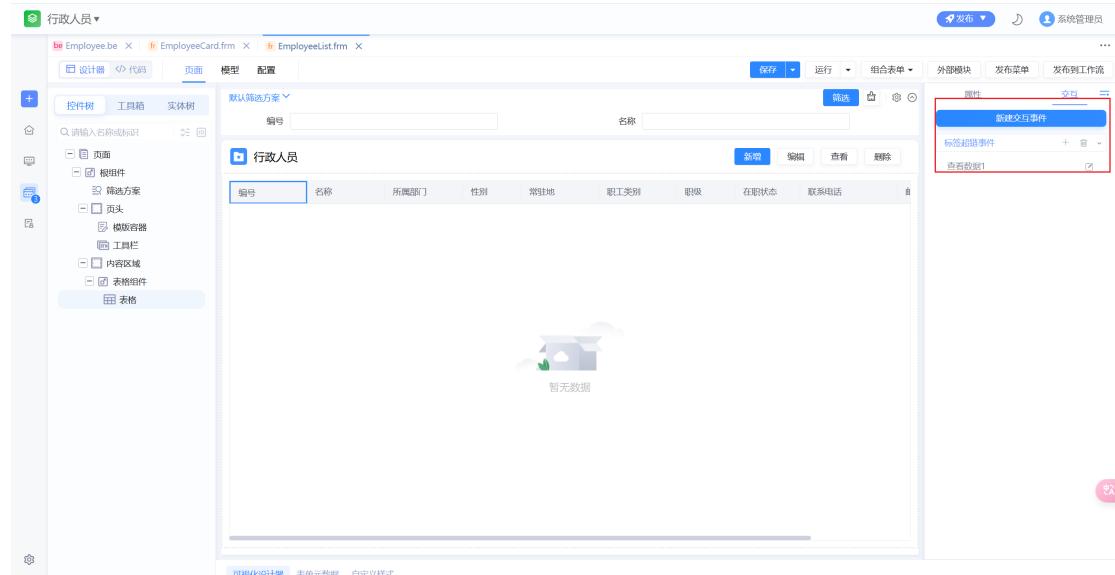
#### 5. 如何配置列表界面



## 6. 如何设置操作列



## 7. 如何设置超链接



## 2.3 专题开发

### 1. 前端开发 内置列表界面



## 2. 业务实体开发

### 业务实体框架（BEF）简介

- 1) 业务实体（BE）：BEF 中的核心，承载实体数据结构和核心业务逻辑，用于规范服务端开发，实现业务逻辑的细粒度拆分和编排。
- 2) 领域模型驱动设计：inBuilder 采用此方法，通过业务实体（BE）描述领域模型，实现业务逻辑的沉淀。

### BEF 运行原理

- 1) JIT 引擎：在完成实体建模后，JIT 引擎将 BE 编译生成运行代码，包括实体类、业务逻辑代码和持久化逻辑。
- 2) 运行时组件：
  1. 业务实体执行框架：负责业务逻辑，包括 CRUD 操作、关联计算规则、校验规则和自定义业务操作逻辑。
  2. 持久化仓库：负责实体数据的持久化处理，加载数据和更新数据到数据库。
  3. 业务实体缓存：缓存热数据，减少数据库交互，提升性能。
- 3) 业务实体开发内容：包括数据结构、业务逻辑及相关构件代码的开发。

### BEF 主要特性

- 1) 业务逻辑与展现逻辑分离：通过 BE 承载业务逻辑，通过视图模型（VM）承载展现逻辑，实现业务逻辑与展现逻辑的解耦。
- 2) JIT 机制：设计时提供元数据建模，开发期元数据发布到运行环境时，自动生成程序源代码并编译为可直接运行的原生程序文件。
- 3) 冲突控制：BEF 提供逻辑锁进行数据加锁服务，确保数据修改的一致性。
- 4) 三级缓存：动作级缓存、事务级缓存和会话级缓存，提升易用性和性能。
- 5) 细粒度的业务逻辑编排：BEF 中的业务逻辑是细粒度的业务规则，业务执行是对这些规则的编排。
- 6) 增量传输：减少前端与服务器端交互传输的数据量，提升执行效率

- 7) 事务两阶段提交、数据增量提交到数据库：降低数据库事务长度，提升应用程序的并发效率。

### 3. 系统管理

### 4. 公共服务

在业务实体框架（BEF）中，表达式的应用是多方面的，它们可以用于计算字段值、控制控件的只读状态、依赖清空、显示隐藏、帮助信息、校验以及必填等场景。以下是对这些表达式类型和它们在 BEF 中如何使用的详细介绍：

- 1) **计算表达式**: 用于根据其他字段或 VO 变量计算字段值。例如，订单总金额可以通过计算订单表中各个项目的金额合计得到，使用表达式 `DefaultFunction.SumByProp("SO.soItems","fee")` 实现。
- 2) **只读表达式**: 动态控制卡片或表格字段的只读状态。如果表达式返回真，则字段只读。例如，如果订单编号不包含"GS"，则控件只读，表达式为 `!DefaultFunction.Contains(SO.billCode,"GS")`。
- 3) **依赖表达式（清空表达式）**: 当表达式返回真时，清空当前字段的值。例如，如果某个条件满足，则清空字段值，表达式为 `!SO.isRed`。
- 4) **相关表达式（显隐表达式）**: 控制控件的可见性。当表达式返回真时，控件可见；否则，隐藏控件或表格列。
- 5) **帮助前表达式**: 类似于用户开发的帮助前，当表达式返回真时允许弹出帮助，否则弹出开发人员设置的提示信息并阻止帮助的弹出。
- 6) **校验表达式**: 类似于前端校验，校验规则为表达式。如果表达式返回真，则校验通过；否则，校验失败，并展示配置的校验信息。
- 7) **必填表达式**: 校验表达式的一种场景，当表达式计算结果为真时字段必填，否则非必填。例如，如果是否红单值为真时，原蓝单编号必填。在表单开发中，这些表达式有助于简化开发流程，减少开发时间，并降低调试和修改的难度。它们使得开发人员可以专注于业务逻辑，而不需要考虑当前行、执行时机等问题。例如，在处理销售订单中采购明细表的金额计算时，只需编写 `SO.soItems.price * SO.soItems.qty` 即可实现金额计算。

开发步骤包括配置表达式和编写表达式。不同类型的表达式配置位置不

同，计算、依赖、校验表达式在属性面板下方的表达式分组中。表达式编写支持匿名函数、内置函数、大数计算等，以简化开发过程。表达式的数据来源包括实体数据和 VO 变量，实体数据可以理解为一个大对象，变量名称就是实体编号。

在编写表达式时，需要注意返回值的处理。如果表达式中只有一条语句，且在第一行，则可以只写表达式，无需显式 return。如果使用了类似判断、分支、函数等带返回的语句，或者以空行开始或有多行代码，同时希望整个表达式有返回值，则应显式 return。

最后，编译表单后即可查看表达式的效果。在开发过程中，可能会遇到表达式执行结果与预期不一致的情况，此时需要对编写的表达式进行调试。可以在表达式最前面添加 debugger 关键字进行调试。

## 5. inIoT

### inIoT 功能说明

- 1) **数据管理**: 管理时序数据库，支持 TDengine、influxDB、OpenTSDB。
- 2) **设备模板**: 定义设备的参数、属性等基本信息。
- 3) **接入网关**: 边缘设备实时数据进入平台的入口，支持 TCP、HTTP、MQTT 等类型的接入方式。
- 4) **编解码器**: 如果设备使用非标准协议的数据接入，用户需要自定义编解码器进行数据格式的转换。
- 5) **创建设备分组**: 将设备分成若干集合，方便管理和统计设备信息。
- 6) **创建设备实例**: 在 inIoT 平台对应现实世界中的设备实体。
- 7) **调试**: 使用 device-simulator 项目模拟设备启动，进行数据上报和命令下发测试。

## 2.4 HandOnLab

因为之后要做更大的系统，此部分便省略了

## 3 利用 inBuilder 开发软件

### 3.1 场景描述和功能设计

- 描述选定的软件题目，以及实现的主要业务场景的功能。

软件题目：学生后台管理系统

主要业务场景功能：

1. 班级管理：创建、修改、删除班级信息，分配教室和班主任。
2. 教室管理：管理教室信息，包括教室名称、位置和容量。
3. 课程管理：添加、修改和删除课程信息，包括课程名、考试类型、课程类型和学分。
4. 部门管理：管理部门信息，包括部门名称。
5. 员工管理：管理员工信息，包括员工的用户名、姓名、性别、职位和部门。
6. 学生管理：管理学生信息，包括学生姓名、学号、性别和班级。

### 7. 成绩管理：录入和管理学生的成绩。

- 通过用例图来描述功能和角色（不需要撰写详细的用例文本），利用流程图或活动图描述主要的业务流程。

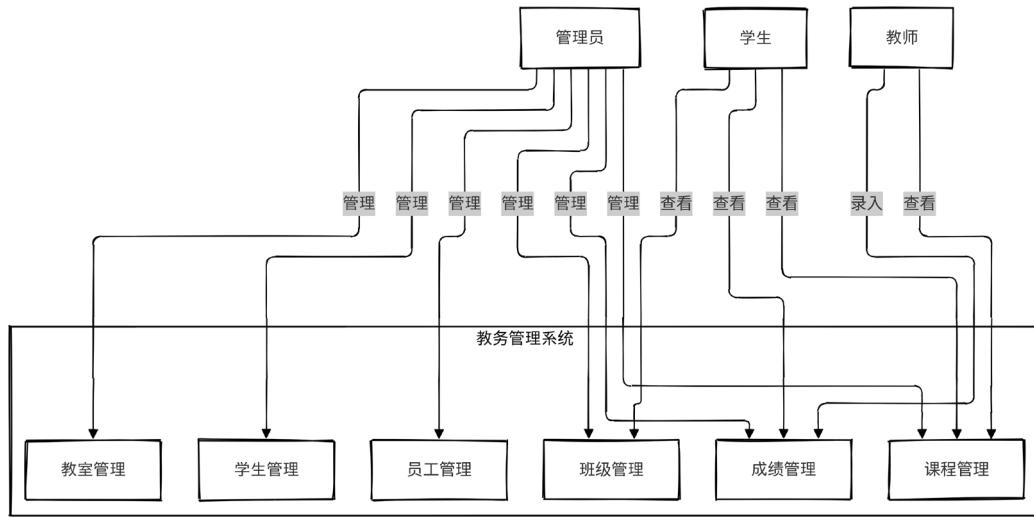


图 1 管理系统用例图

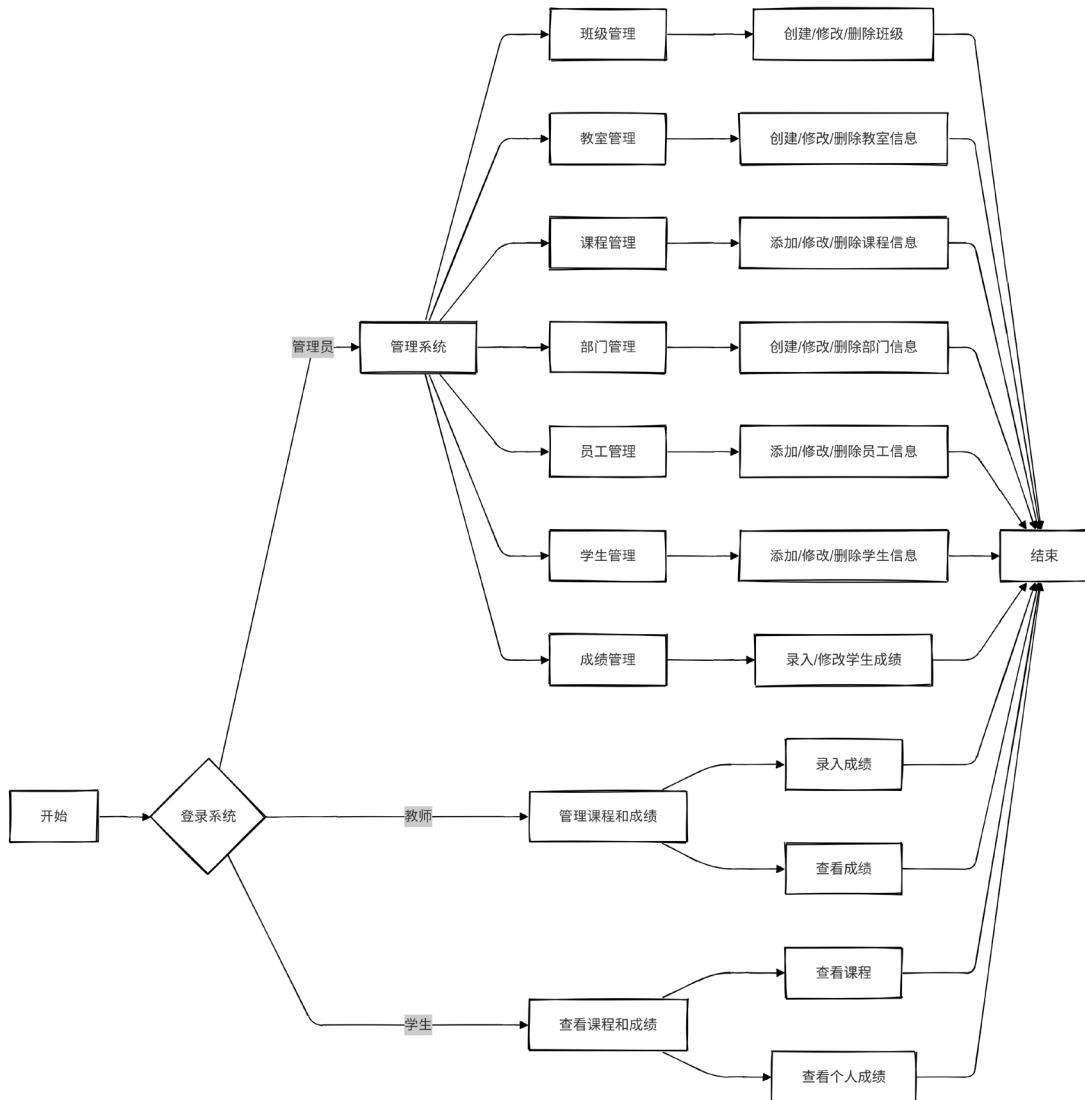
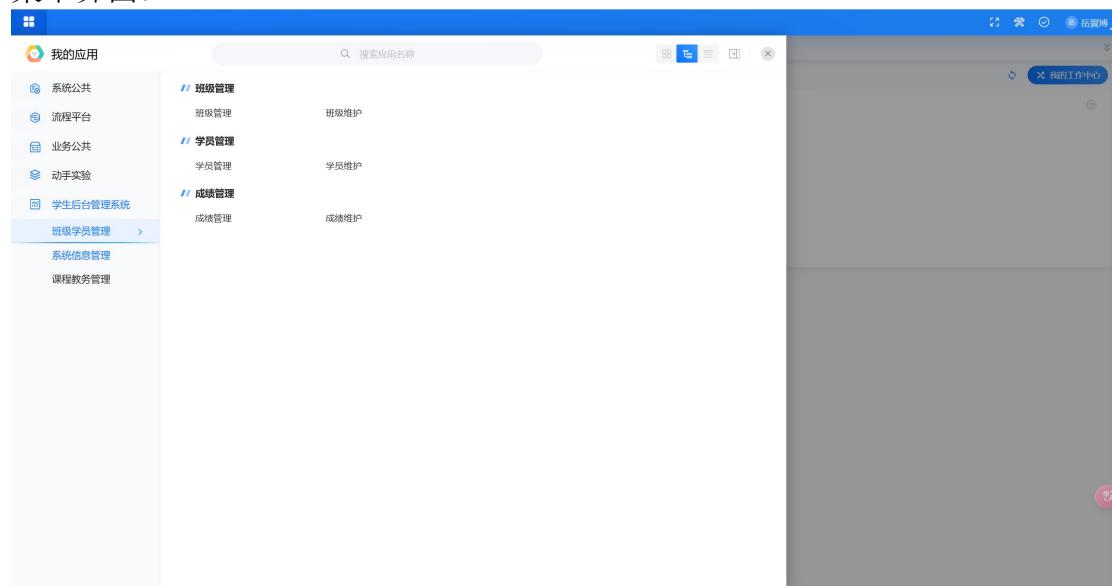


图 2 管理系统流程图

## 3.2 界面截图

给出实现的主要界面截图如下：

菜单界面：



班级管理：（能进行增删改查）

班级管理				
班级名称	教室名称	开课时间	结课时间	班主任
2211301	M102	2024-11-26 20:18:15	2024-11-30 20:18:17	岳翼博
22313101	M102	2024-11-14 20:30:00	2024-11-29 20:30:03	岳翼博

学员管理：（能进行增删改查）

学员管理

姓名	学号	性别	手机号	学历
杨晓	2022211564	女	1317492333	硕士
常与出	2021313148	女	19716318612	本科

### 成绩管理：（能进行增删改查）

成绩管理

姓名	课程名	考试类型	课程类型	学分	课程代码	分数
杨晓	英语	考试	必修	2.00	22WPHH40121	90.00
常与出	组织行为学	考试	必修	1.50	22WPHH32010	86.00

### 员工管理：（能进行增删改查）

员工管理

姓名	性别	职位	入职时间	部门名称
曾湘临	女	学工主管	2024-11-13 20:00:00	就业部
岳翼博	女	讲师	2024-11-13 19:00:00	咨询部

### 课程管理：（能进行增删改查）

课程名	考试类型	课程类型	学分	课程代码
英语	考试	必修	2.00	22WHIPH40121
组织行为学	考试	必修	1.50	22WHIPH32010

### 教室管理：（能进行增删改查）

教室名称	教室位置	教室容量
H201	h楼	76
M102	m楼	65

以上是管理界面(列表查看), 下面展示增删改查(为了减少麻烦, 这里只展示班级界面的增删改查, 其它界面都差不多)

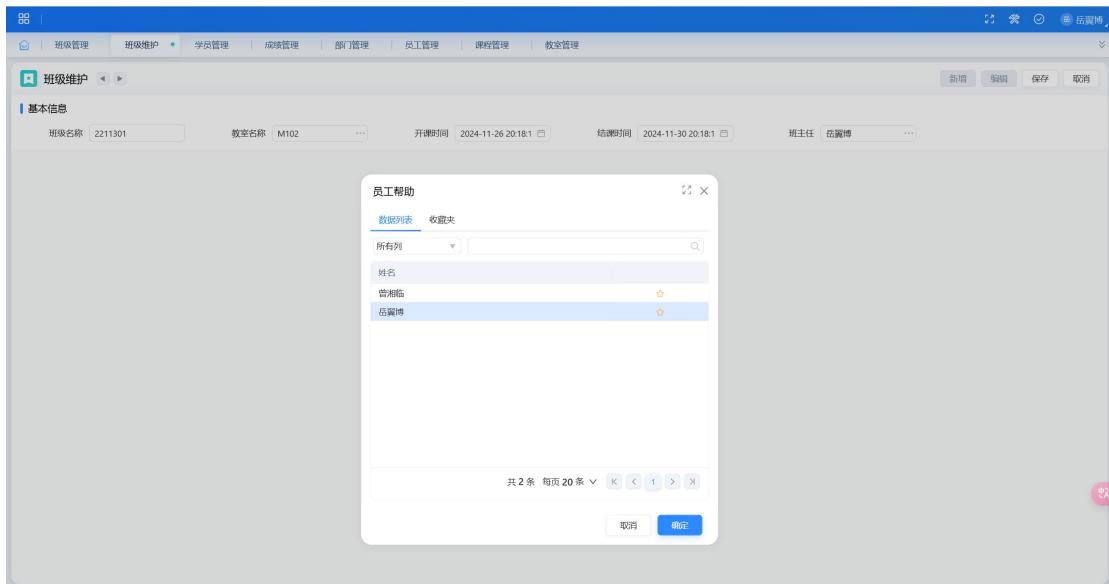
班级增加： 点击班级管理的增加按钮会跳转到班级维护页面, 这里进行输入, 并且部分字段能使用帮助直接进行选择, 如下图：教室直接进行选择即可

这里插入数据(20241111,M102,2024-11-11,2024-11-11,岳翼博), 会在之后进行删除操作

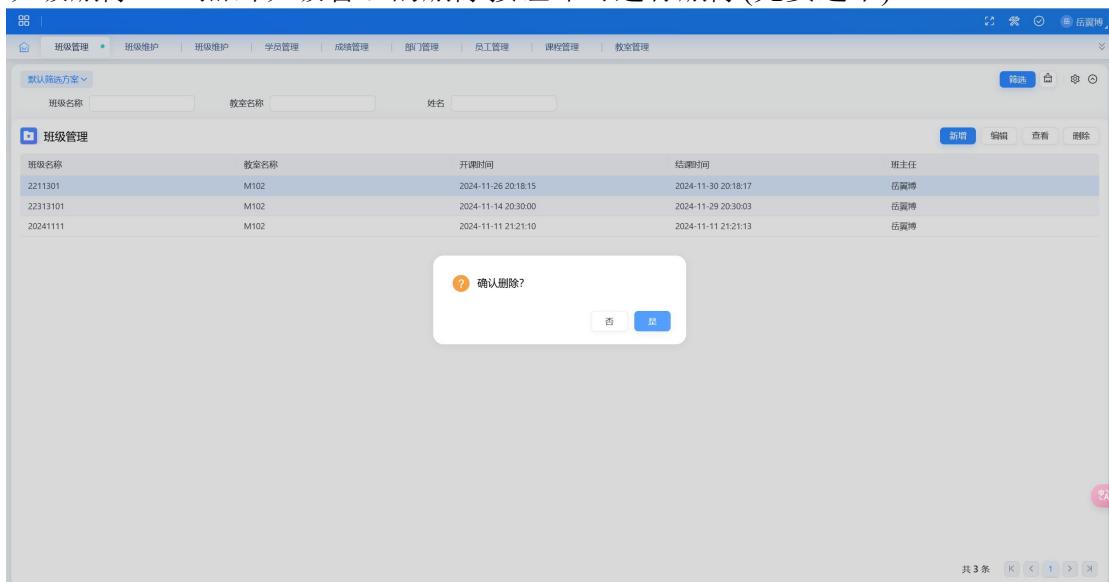
The screenshot displays a low-code platform interface for classroom management. At the top, there is a navigation bar with links for '班级管理', '班级维护' (highlighted in blue), '学员管理', '成绩管理', '部门管理', '员工管理', '课程管理', and '教室管理'. Below the navigation bar, there is a search bar and a toolbar with buttons for '新增' (Add), '编辑' (Edit), '保存' (Save), and '取消' (Cancel). A modal window titled '教室帮助' (Classroom Help) is open, showing a data list with two entries: 'H201' and 'M102'. There are also tabs for '数据列表' (Data List) and '收藏夹' (Favorites). At the bottom of the modal, there are buttons for '取消' (Cancel) and '确定' (Confirm). The main content area shows a table titled '班级管理' (Class Management) with three rows of data. The columns are '班级名称' (Class Name), '教室名称' (Classroom Name), '开课时间' (Teaching Time), '结束时间' (End Time), and '班主任' (Class Teacher). The data entries are: 2211301, M102, 2024-11-26 20:18:15, 2024-11-30 20:18:17, 岳瀛博; 22313101, M102, 2024-11-14 20:30:00, 2024-11-29 20:30:03, 岳瀛博; 20241111, M102, 2024-11-11 21:21:10, 2024-11-11 21:21:13, 岳瀛博. To the right of the table are buttons for '新增' (Add), '编辑' (Edit), '查看' (View), and '删除' (Delete). At the bottom of the main content area, there is a pagination bar showing '共 3 条' (3 items) and navigation icons.

可见已经把数据插入了

**班级编辑：** 点击班级管理的编辑按钮会跳转到班级维护页面，这里进行编辑即可



班级删除： 点击班级管理的删除按钮即可进行删除(先要选中)



删除即可

其他管理界面一样，这里就不贴出了

### 3.3 Demo 视频

录制一段 3-5 分钟的 demo 视频，介绍和展示所开发的程序。将视频上传到 Youku、哔站等网站上。

此处给出视频地址：

**【开源软件开发实践实验 5——inBuilder】**

[https://www.bilibili.com/video/BV1PWzKYFEFH/?share\\_source=copy\\_web&vd\\_source=e0bd6da8de7f02254a70725bb8704270](https://www.bilibili.com/video/BV1PWzKYFEFH/?share_source=copy_web&vd_source=e0bd6da8de7f02254a70725bb8704270)

## 4 \*\*\*熟悉 UBML 项目

如果做出了贡献，此处给出 PR 地址。

## 5 小结

实验总结：

本次实验旨在通过实际操作让我们掌握低代码平台 inBuilder 的使用，并能够利用该平台开发具有实际应用场景的软件产品。实验不仅要求我们完成基础的开发任务，还鼓励我们探索开源项目 UBML，增加对开源社区贡献的理解。

建议：

1. 提前规划：在实验开始前，提前规划好软件产品的功能和设计，这有助于更高效地完成开发任务。
2. 文档阅读：仔细阅读 inBuilder 的在线手册和文档，确保对平台有充分的了解。
3. 实践操作：通过实际操作来加深对低代码平台的理解，实践是学习的最佳方式。
4. 代码管理：在开发过程中，注意代码的版本控制和文档编写，这对于后续的代码提交和维护非常重要。
5. 时间管理：考虑到提交截止日期，合理分配时间，确保有足够的时间来完成实验报告和视频的制作。
6. 反馈与改进：在实验结束后，反思实验过程中遇到的问题和挑战，思考如何改进，这有助于在未来的项目中避免相同的错误。