1.

问题1：首先小李在未经过项目经理的同意下私自改写项目功能，对于与之相关联的功能很大程度上会产生影响，各功能之间的接口调用数据获取等都可能出现问题，导致了后续项目整合出现问题，无法给到一个很好的交付。

问题2：首先需要弄清楚小李所修改的部分有哪些，与所修改部分相关联的部分有哪些，尽力修复给方面的问题，给局方一个尽可能满意的交付，其次，在之后的管理方面需要更加注意这方面的问题，对于小李的行为也一定要做出相应的批评惩罚，避免以后再发生同样的问题。

2.

我认为一个好的软件开发人员首先需要具备专业相关的硬素质，例如与自己工作方向相关的技术等方面的掌握，如Java、UI、计算机基础知识、数据库开发、行业规范等，除此之外，还需具备一定的软素质，主要从自我管理、团队管理、工作管理这三个方面来体现，自我管理包括了时间管理、情绪管理、积极主动学习等，工作管理又包含了目标管理、项目管理、交付负责以及执行力，团队管理则包含了团队协作以及有效沟通等。

目前在实际工作中无论是硬素质还是软素质都还有很大的提升空间，在硬素质方面，在现在培训阶段多跟导师学习，通过书籍以及相关资料不断提升自己的技术能力，在软素质方面，首先应该转变好心态，尽早适应职场的情况与压力，不断跟进，慢慢改掉以前的陋习形成良好的习惯，在工作方面遵守公司的各项规章制度，严格按照公司要求交付工作。

3.

5G物联网应用场景，通过5G将各种信息传感设备与互联网结合起来而形成的一个巨大网络，其中关键技术包括射频识别技术、传感网、云计算等，物联网的应用场景包含了智能交通、智能农业、智能工业、智能安防等各个方面，在思特奇公司中比较重点的应用有智能焊接系统、人脸识别门禁考勤、车务通、智慧办公应用、一县一品物联网追溯等。5G在电子商务方面应用同样十分广泛，而思特奇在电子商务业务中的代表性项目有：移动总部移动商城、和生活移动APP、虚拟运营商网厅等，这些场景应用商务模式属于B2B。

4.

“系统是设计出来的”这句话表明了设计阶段在系统开发过程中的地位之重，前期良好的设计可以为后面的代码实现节省不少时间，高质量的软件开发之道必不可少的是高质量的设计（包括模块化设计、面向抽象编程、错误与异常处理）、规范的编码、有效的测试；

设计的六大原则就是模块化、抽象、信息隐藏、弱耦合、强内聚和可重用，好的面向对象设计是那些可以满足“应对变化，提高复用”的设计。设计团队要解决的问题是怎么最优地实现，而不是单纯的怎么实现。探索未知世界是技术人员的使命，设计人员在设计之初要对可能出现的问题有一点的预判。

接到一个设计任务后，对我而言首先应该明白设计的目的，就是该设计需要实现的功能以及对其相应的限制条件，在明白以上信息的基础上针对真个设计有一个全局性的构想，考虑各种可能出现的问题以及相应的解决办法，最后再一点一点完善整个设计。

5.

1. 首先想到的是一本书架构，通过在训练营的课程对于思特奇独创的“一本书”EP+APP架构体系有了一定的了解，一本书架构顾名思义可以理解为一本书的形状，其左页是基于云计算技术构建的企业业务能力，右页是基于大数据、AI技术构建的企业数据能力，而书脊则是云、大数据一体化运营掌控平台，一本书架构的核心，就是将业务事务处理和数据智能处理融于统一平台。
2. SpringMVC：属于Spring框架的一部分，用来简化MVC架构的web应用程序开发。

优点：（1）拥有强大的灵活性，非侵入性和可配置性（2）分工明确，包括控制器、验证器、命令对象、模型对象、处理程序映射视图解析器，每一个功能实现由一个专门的对象负责完成（3）可以自动绑定用户输入，并正确的转换数据类型（4）可重用的业务代码：可以使用现有的业务对象作为命令或表单对象，而不需要去扩展某个特定框架的基类。

缺点：（1）Servlet API 耦合难以脱离容器独立运行。（2）太过于细分，开发效率低。

1. Springboot是提供的全新框架，使用来简化Spring的初始搭建和开发过程，使用了特定的方式来进行配置，让开发人员不在需要定义样板化的配置。此框架不需要配置xml，依赖于像MAVEN这样的构建系统。

优点：（1）减少了大量的开发时间并提高了生产力（2）避免了编写大量的样板代码，注释和XML配置（3）快速整合第三方框架，无需配置文件（4）使用Java开发基于Spring的应用程序非常容易。

缺点：（1）修复bug较慢，报错时难以定位到该点。（2）集成度较高，不易了解底层。

1. 微服务架构：微服务的基本思想在于考虑围绕着业务领域组件来创建应用，这些就应用可独立地进行开发、管理和加速。在分散的组件中使用微服务云架构和平台使部署、管理和服务功能交付变得更加简单。

优点：（1）扩展性好，各个服务之间低耦合（2）容易部署和开发（3）易于测试

缺点：（1）由于服务无状态和引入了分布式，较难解决事务一致性问题（2）集成复杂，即模块在集成时候需要外部微服务模块更多的配合