Accepted

第一次开会：（讨论主题及分工会议）

3.8 星期天 下午2点到4点

主题：分工

讨论内容及结果：列出待做事务清单，分配工作

第二次开会：（评审会议）

3.13 星期五 晚上6.30到8点

主题：审议用例文档和系统顺序图、状态图、概念类图

讨论内容及结果：审议了系统顺序图和用例文档，完成git

第三次开会：（跟踪会议）

3.18 星期三 上午8点到10点

主题：讨论评审会议上发现错误的更正结果

讨论内容及结果：确认所有更正结果正确

第四次开会：（讨论主题及分工会议）

3.21 星期六 下午2点到2.30

主题：第一次文档作业总结反思和第二次作业分工

讨论内容及结果：完成作业分工，确定了merge的deadline

第五次开会：（评审会议）

4.1 星期三 晚上8点到8点半

主题：第二次作业提交与测试

讨论内容及结果：测试通过，完成第二次作业提交

第五次开会：（讨论主题及分工会议）

4.13 星期一 下午6点到6点半

主题：第三次文档作业分工

讨论内容与结果：完成用例筛选，明确作业分工，确定deadline

第六次开会：（评审会议）

4.18 星期六 上午10点到11点半

主题：第三次文档作业评审纠错

讨论内容与结果：完成评审，纠正格式问题与“假设与依赖“内容的错误和空缺

第七次开会：（跟踪会议）

4.20 星期一 上午8点到10点

主题：讨论评审会议上发现错误的更正结果

讨论内容及结果：确认所有更正结果正确并提交第三次文档作业

第八次开会：（评审会议）

4.29 星期三 下午5点到6点

主题：第三次代码作业push，merge与提交

讨论内容与结果：完成提交，测试全部通过

第九次开会：（讨论主题及分工会议）

5.2 星期六 下午2点到4点

主题：第四次文档作业分工

讨论内容及结果：完成分工，确定了提交deadline

第十次开会：（评审会议）

5.10 星期日 晚上6.30到8点

主题：审议体系结构描述文档

讨论内容及结果：审议体系结构描述文档，完成merge

第十一次开会：（跟踪会议）

5.12 星期二 上午8点到10点

主题：讨论评审会议上发现错误的更正结果

讨论内容及结果：确认所有更正结果正确

第十二次开会：（讨论主题及分工会议）

5.17 星期天 下午2点到4点

主题：大作业分工

讨论内容及结果：列出具体任务清单，分配代码工作，安排前端学习计划

第十三次开会：（评审会议）

5.22 星期五 晚上6.30到8点

主题：检查前端学习进度

讨论内容及结果：前端学习进度达成目标，计划开始正式编写代码

第十四次开会：（跟踪会议）

5.26 星期二 上午8点到10点

主题：大作业第一次merge和功能测试

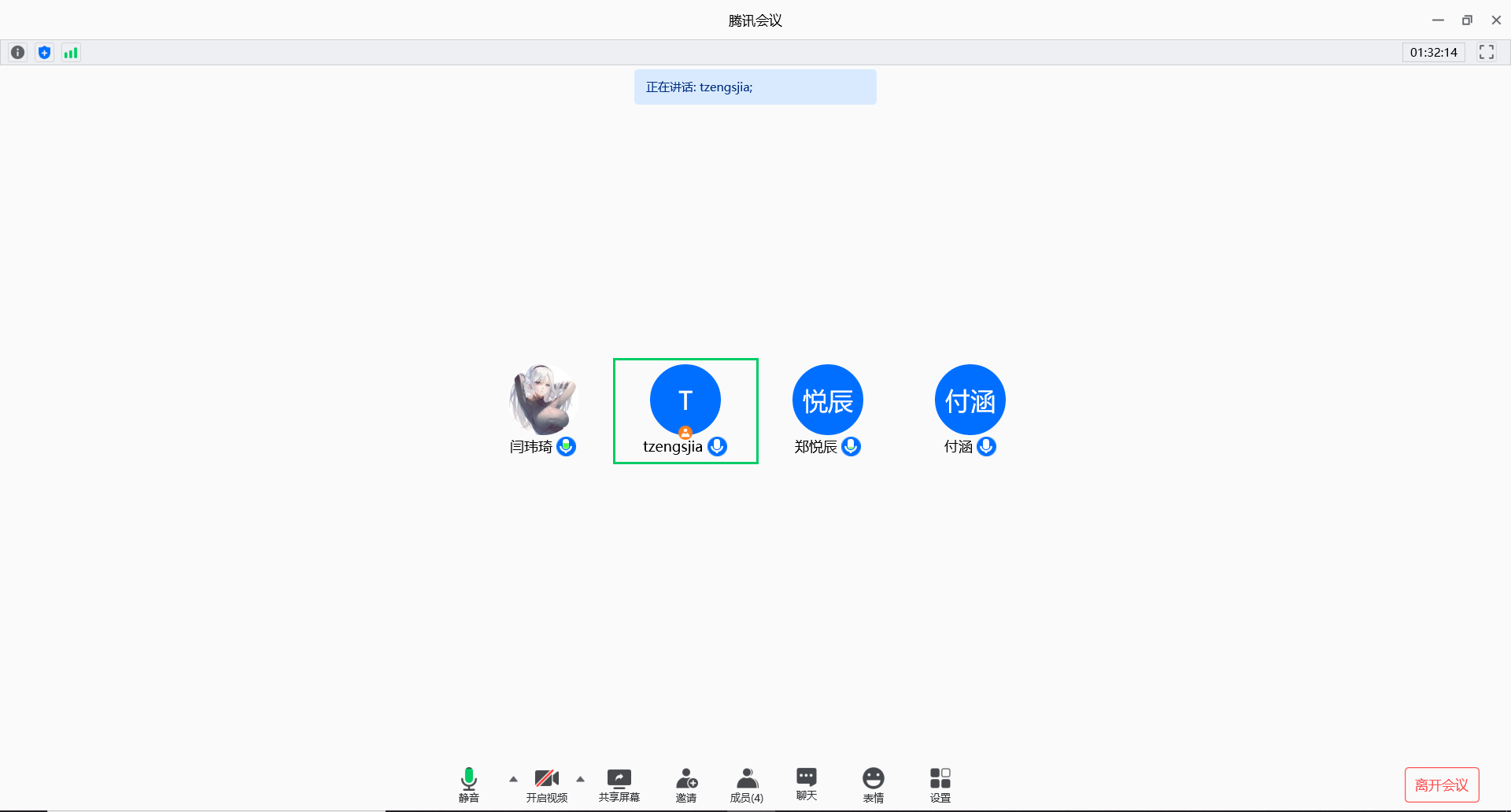
讨论内容及结果：merge完成，第一期工作全部测试完成

第十五次开会：（评审会议）

6.2 星期二 下午6.30到8.30

主题：大作业第二次merge和功能测试

讨论内容与结果：merge完成，第二期工作全部测试完成



第十六次开会：（评审会议）

6.11 星期四 下午5.00到6.30

主题：大作业第三期功能测试

讨论内容与结果：第三期功能全部测试完成，计划开始编写文档

第十七次开会：（评审会议）

6.14 星期日 下午1.30到3.00

主题：查代码bug

讨论内容与结果：

发现的问题：1.不使用优惠券结算后总价会不显示

2.删除“删除订单”按钮

3.评论区头像问题

4.添加优惠策略

5.订单详情一直显示1星bug

6.搜索出现了问题

7.优惠策略显示不出来

8.修改信息有问题

9.信用值初识全部改为1w

第十八次开会：（跟踪会议）

6.16 星期二 上午10点到12点

主题：再次测试bug修复

讨论内容与结果：上次评审会议发现的所有bug修复完成

第十九次开会：（讨论主题及分工会议）

6.18 星期四 下午1.30到4点

主题：最后一次代码测试，文档分工和检查准备

讨论内容与结果：文档分工完成，为大作业检查做了些许准备

第二十次开会：（评审会议）

6.21 星期天 上午9点到11点

主题：所有文档检查与测试用例文档

讨论内容与结果：文档检查完成，准备撰写测试报告

第二十一次开会：（讨论主题及分工会议）

6.23 星期二 下午6点到8点

主题：集成测试代码与测试报告的讨论与分工

讨论内容与结果：完成分工，确定deadline

第二十二次开会：（总结会议）

6.29 星期一 下午2点到4点

主题：总结本次项目开发中遇到的困难与项目开发组体现出的不足，总结本次项目开发的经验

讨论内容与结果：

总结经验：  
1.在开发中应明确需求，确定其可行性，不要急于代码实现    
2.在开发模型上善用迭代式开发      
3.避免循环依赖，做到高内聚，低耦合    
4.遇到开发难题，善用搜索引擎，困难过大可及时与其他成员沟通解决，不能因为一个问题使整个开发停滞