**SPRINT RETROSPECTIVE**

**By Team 1**

As any other agile software team, we too held a sprint retrospective meeting at the end of the sprint after the sprint review. This meeting consisted of the entire Scrum team. In this informal meeting, we have inspected and adapted the process of building the product.

**What went well?**

1. There was a positive team spirit in the team and everyone had an effective communication among the members.
2. The sprint velocity was high since the team members worked on it with full determination and were successfully able to achieve all the requirements.
3. In our code, we have used some of the design patterns specially Observer Pattern very efficiently which has improved the performance of the product drastically.
4. The team has implemented the Model-View-Controller approach which has enables us to achieve several software engineering principles such as high cohesion, loose coupling, parallel development of the code and separation of concerns.
5. The structural design of the classes and packages of our project eliminated all the cyclic dependencies that were present successfully and made our component structures a directed acyclic graph.
6. Components which were stable(I=0) were made flexible by adding abstract classes so that other components can extend it and that the design is not too constrained.
7. Applied Clean code design principles- We have separated the application for creation of objects with that of using the objects in separate components with the help of factory pattern to follow Single Responsibility Principle.

**What could we have done better?**

In spite,of successfully completing the project on time and within budget and with all the features, we all came to a conclusion that some features in our project could have been improved which we planned to do in the next sprint.

1. We could have further improved the code by incorporating some more design patterns like decorator/builder pattern or similar patterns.
2. Instead of manual testing, we could have executed test plans and automated those test plans using a testing tool like JUnit.
3. There was little technical debt at the end which we will fix it in future by restructuring for a maintainable code.