Hello and welcome to this presentation on Robustness and Sequence Diagrams, I, James Moran, will be giving this presentation on their use in the Game Café, let’s begin.

What are Robustness and Sequence Diagrams? For Robustness Diagrams, they are a graphical way to depict use cases, that is also a ‘sanity check’ on Use-Cases (to make sure the Use-Cases match-up to what functionality the User would want from the system), that also allows the team to uncover new classes, which would not have been previously identified for the system. Sequence Diagrams allow a team to design the system in detail, considering the methods/functions the classes in the system will have and the order that these methods/functions are executed.

In the Game Café, the Robustness Diagram is as follows: Considering Staff Members of the Game Café, they would want to manage information of Members, Bookings, eSports Events, Hardware and Software (an example of a boundary), after manging this information, check to see if it is correct and that this information is not already present in the database (these are examples of controls). If it is not correct though, show an Error Dialog (another Boundary)), before adding it to the Information Database (an example of an Entity). A similar procedure is taken for if there is a ticket available for an eSports Event (with similar Boundaries, Controls and Entities).

Moving on to the Sequence Diagram, this details the ‘timeline’ for the Use-Case of a Game Café Staff Member adding information to the system’s database. They start off by finding the target category of information, that they would wish to add, then the system requests (via the Windows Form Interface) details to be entered by the User, whom enters these details, which are validated for correct format by the system, as well as checking against the database for duplicate values, if the information is of a valid format and there are no duplicates in the database, then this information will be added to the database.

Robustness and Sequence Diagrams are both used in the Game Café. Bringing the advantage of Better Project Analysis (making sure that the system is robust and has the appropriate chronological order) and providing excellence for documentation (as tracing through these diagrams is fairly straightforward). The disadvantage of Complexity (having to learn/know the correct UML notation to use and the possibility to try and encompass too much in either diagram), has not affected their usage in the project too much, as the team has experience in assembling UML diagrams (to avoid making them overly complex). As for these models being static (not being able to easily modify them, to suit new requirements), we would simply assemble new diagrams for these requirements (as this would not take hours away from other aspects of the project, given how many hours we have assigned for the project).

This slide details the references used in this presentation, thank-you for taking the time to watch this presentation and goodbye for now.