**Design rationale**

1. **Tool for Uml Diagrams:** We had a prolonged discussion of the tool we would use for constructing the Uml Diagrams. We narrowed our choices down to ‘Star UML’, ‘Visual paradigm’, ‘Argo UML’. We then selected ‘Star UML’ to construct UML diagrams for ease of using tools and editing our diagrams in Star Uml.
2. **Database:** We decided on using the database ‘sqlite’ which is inbuilt in python. The reason for this decision was that we would be able to develop the application faster by using the inbuilt database.The database table names and attributes were initially confusing and redundant. We have tried to reduce redundancy by normalizing the tables that we created initially.
3. **Class Diagrams:** To follow the divide and conquer strategy, we made multiple classes such as toolListing, changeLocation, sharefromhome, sharefromcommunityshed. But we realized that these can be functions in the Tool Management class that we have created. So we included them under one class, as the User will use all these functions under Tool management in the project.
4. **Names of Classes: Naming Convention:** The names of the classes while constructing the class diagrams initially were not specific to the classes that we needed to implement. We changed the names of classes to specific names so that we get a better understanding of what our classes do. Similarly we changed some of our data members and member function names to specific names for better understanding.
5. **Sequence Diagram:** The sequence diagram of ‘Request for reservation of tool’ was initially an overview of how objects interact in time sequence. We discussed the use cases related to the reservation of tool in detail and noted the actions we need to consider from the systems point of you as well as the user owner and borrowers point of view. We then modified our sequence diagram to show a better control flow with respect to what the actors and lifeline would do.