At the beginning, the problem definition is discussed through determining the main problems observed in existing systems and how our system will solve them. However, at the first stage we did not have a clear image about what really the problem that this system will solve, but as we learned in the first lecture that problem definition might not be clear at the first stage and it may occur after some other activities, so we decided to think from a user perspective and determine the main system functionalities. Therefore, there was a need to find out how to gather the system requirements and which software development approach should to follow. In order to accomplish that, we followed (table 2-3 McConnell 3.2) to determine the kind of software we are working on and what typical good practices shall we follow. From this table, we found out that our system is a business software project that tends to reap benefits from agile development, which is highly iterative approach, where the system construction is interleaved with planning, requirements, and architecture activates. From that point and according to software architecture lecture, we followed Top-level design approach. At the beginning starting with specifying the general problem of the system is to search for Manchester events. Then broke it into manageable parts such as searching by category, location, date or artist. Then each part became a new problem that needs to divided again such as searching by category will list all events with ability to register into that event or added to a wish list and so on. Next figure is a quick sketch of the top down approach we followed.

Then, we decomposed further by producing a model of the whole system architecture through drawing UML class diagram and database design (ERD) along side with a quick sketching web user interfaces. There were many modifications in the database or classes models according to user interfaces requirements. In this stage, the development technologies have been discussed. We used java maven and spark frame work for this web application. According to construction lecture “reading is everything”, reading other code whatever to using or modifying is a key skill in software construction. We stated to read a sample code that Mark put in the Google doc and we modified it to meet our project work.

Here I am not sure we have to but the UML diagram and ERD??

Then driving bottom out with concrete code of main system functions. For example, creating class event attributes and its methods, then creating a connection to the database using JDBC. After that, writing a spark code using freemaker. However, in this stage we did many spikes to test the system functionality.

Putting some spikes codes?

Furthermore, many refactoring is done after coding. For example, creating a connection to the database was written in each method that is querying the database. However, we refactored the code to make the connection only once when the system starts.

Putting some refactoring codes?

Also there is a lecture about construction (for debugging section), so any modifications anyone did it to the code you could list it here and explain it and referred to that lecture.