Quality Attribute Scenario 1:

Risk: In this Scenario due to heavy load mobile application might be slower and server can be down at any time.

Non-Risk point: No.

Trade off : If user access can be limitted for particular unit of time, then performance would be improved.For example,in 1 ms , 1000 users can access the mobile app.In case of more than that, system would throw an error message saying that 'Server is busy.Please try later' So, this is Trade off as availability quality is compromised for system Performance.

Sensitivity point: NA

Quality Attribute Scenario 2:

Non-Risk point: As User accounts get created with email and User Name. Here, email is unique Id which can be considered as a user key for each account entry in database.It would maintain the consistancy in database.

Sensitivity point: This is the centre of the database schema.If account records face data inconsistancy due to new user keys , then database relations[like Account ->Items] would be corrupted.

Risk: No.

Trade off : NA.

Quality Attribute Scenario 3:

Risk: As accounts' data are stored in external system[i.e. shared storage system].If, external system is down,then no data can be loaded on user login.So, in that case,alternative shared storage system should be there.If external system getting down for some reason, then mobile app/web app would connect immediately the alternate storage system.

Sensitivity point: Both the storage system should have data consistancy.Otherwise wrong account data would be loaded in case of one server is down

***Non-Risk point*:** No.

Trade off : NA.

Quality Attribute Scenario 4:

Risk: As per this scenario, if user create an item with empty field ,then error message would be thrown.But ,if user creates the items in bulkload and among huge item records, few records contain

empty fields,then whole bulk process would be errored out which is not expected & can be consider as a risk.

Sensitivity point: Yes.So, if we change the system design to insert the items in a bulk instead of single entry, then this scenario would be impacted.To resolve the issue,we can insert error log in database for invalid item records and rest of the valid records would be inserted in bulkload process .

Non-Risk point: No.

Trade off : NA.

Quality Attribute Scenario 5:

Non-Risk point: Yes, as valid item would be inserted into database.

Trade off : It can be considered Trade off between performance and Authentication to validate the item every time before inserting it.

Risk: If numerous user try to add items at same time,then system can be down.

Sensitivity point: NA.

Quality Attribute Scenario 6:

Risk: It is risk, if changes are made to the data being used.In this case, original data can be lost.

Non-Risk point: No.

Trade off : To resolve this risk, semaphore can be used.So, if data is being used by system ,then it can't be changed until unless this data is idle in the system.

Sensitivity point: NA.

Quality Attribute Scenario 7:

Risk: No, As data would not be lost in this case.

Non-Risk point: Yes.

Trade off : Yes, to secure the date consistancy, performance would be degraded as one process request needs to wait for another process request before changing the same data.

Sensitivity point: NA.

Updated Utility Tree: