Utility Tree

Utility

Quality Attribute

Performance   Security   Interoperability   Usability

User Experience

Attribute Refinement

Throughput   Latency   Authentication   Authorization   Shared Storage   Accessbility

The apps try to minimize user errors and provide handler errors (H)

The apps provide feedback to the all user actions (H)

Architecturally Specific Requirement

Process 500 request per minute regardless of system load (H)

It takes less than 1 second to complete any given task. (H)

A user must create or have an account to use the app. (H)

A user can only edit items they own. (H)

A user cannot place a bid on their own items. (M)

User data must be in shared storage (H)

Users can access their data from the Sharing Mobile App (H)

Users can access their data from the Sharing Web App (H)

Analyzing and Evaluating Architecture

Scenario 1 - User clicks on search
Analyze: It's a **risk**
Explanation: The heavy load could get worse the latency

Scenario 2 - User creates an account using a username and email
Analyze: It's a **non risk**
Explanation: The architecture has the capacity to process this..

Scenario 3 - User logs in with username
Analyze: It's a **non risk**
Explanation: The architecture has the capacity to process this.

Scenario 4 - User tries to create an item with an empty field
Analyze: It's a **risk**
Explanation: The all unexpected errors maybe not handle right

Scenario 5 - User adds a valid item to their items
Analyze: It's a **no risk**
Explanation: The architecture has the capacity to process this.

Scenario 6 - Changes are made to the data being used
Analyze: It's a **sensitivity points**
Explanation: Changes on this item could affect Mobile and Web App

Scenario 7 - two users are attempting to make changes to the same data.
Analyze: It's a **tradeoffs**
Explanation: Increease the consistency could reduce the throughput.