This assignment will challenge you to use the ATAM Process to analyze and evaluate a software system based on its architecturally significant requirements.

We want to evaluate the architecture and design of the given system to determine the future of its development. It is important to identify the risks our system faces that may hinder the end users or developers. Will the end users of our system want to keep using it? Will it be easy and affordable to continue to develop the system? This will reveal issues we may not have known about our system that may be necessary to change in future releases.
Review the Quality Attributes and Analyzing and Evaluating Architecture Lectures to prepare for this assignment. You will need to reference the ATAM Process and Quality Attribute Scenarios.
You are given an Attribute Utility Tree that points out the Architecturally Significant Requirements (ASRs) of the current version of the system. This is an initial Utility Tree we have created for step 5 of the ATAM Process to gain insight about the system and identify the quality priorities that go into the evaluation of our system.
For each Quality Attribute Scenario provided, identify if it is a risk, non-risk, tradeoff, or sensitivity point in the applications architecture and give a brief explanation (3 sentences or less) for each. This can be put into point form, but each Scenario should be covered.
Once you have finished your evaluation, update the Utility Tree to reflect the Scenarios. You should be able to make 4-5 changes, including adding a Quality Attribute and adding more Attribute Refinements and ASRs to reflect the risks. You may also update the current ASRs to reflect the Scenarios or (if justified in the evaluation) change their priorities.
Once you have finished your evaluation, use the Quality Attribute Scenarios to create an updated Utility Tree that considers the Quality Attribute Scenarios based on the primary ASRs from the previous Utility Tree.

Utility										
Quality Attribut:	Performance		Security			Interoperability				Availability
Attribute Refinement:	Latency		Authentication		Authorization	Shared Storage		Accessibility		Maintenance downtime
Architectually Significant Requirement:	Provide the user with feedback if the system takes > 1s processing a request (H)		A user must create or have an accout to use the app. (H)	A user should provide or create a Pasword to use the app. (H)	A user can oly edit items they own (H)	A user cannot place a bid on their own items. (H)	User data must be in shared storage (H)	User can access their data from the Sharing Mobile App (H)	User can access their data from the Sharing Web App (H)	Scheduled maintenance takes place outside of buisness hours or hours of low usage. (M)

Concrete Quality Attribute Scenarios			
Concrete	Response Measure	Type	Explanation
1	Time to load items from remote database	Tradeoff	If it take long usability could be afected
2	Time for account to be stored externally	Non-risk	Message of waiting should be shown to the user
3	Time for account to be loaded externally	Non-risk	Message of waiting should be shown to the user
4	Time taken to validate input	Non-risk	Message of waiting should be shown to the user
5	Time taken for server to save item and send confirmation message	Tradeoff	If it take long usability could be afected
6	Time taken for changes to be sent	Risk	If take long and the aplication is close coul be a data problem
7	Time taken for changes to be sent	Sensitivity	If take long and the aplication is close coul be a data problem. Message of waiting should be shown to the user