Q.A. Grade Sheet

1. Content under Test
   1. is it there?
   2. does it function as expected?
   3. is it visibly sound?
   4. will it load consistently?
   5. can I update it?
   6. does it load correctly after update?
   7. can I remove data from the content?
   8. does it remove data correctly?
   9. can we verify it is complete?
2. Performance
   1. Did it load as expected?
   2. did it load in an appropriate amount of time?
   3. can I get to the data with minimum lag or latency?
   4. can it handle multiple connections?
   5. what is the maximum number of attempts or connections it can handle and remain stable?
   6. at what point is it no longer stable?
   7. what are the minimum resources it needs to be stable?
   8. what are the minimum resources it needs to be efficient?
   9. what are the minimum resources it needs for maximum performance?
3. User Interface
   1. Is it easily accessible?
   2. is it easily understandable?
   3. will the user enjoy the experience?
   4. will it be frustrating for the user to get the desired results?
   5. what would make the user experience ideal for the maximum number of potential interactions?
   6. what are the pain points?
   7. how can the user break it?
   8. what will the user most likely do upon accessing the interface?
   9. what will the user most likely do when their task is completed?
   10. what will the user do if the service or function fails?
   11. how can the user get help, if it fails or does not perform as expected?
   12. what will the user want for the ideal experience? speed, access, results, interface?
   13. if the user fails to get what they were seeking, will they come back or go else where for the results?
   14. how can the user reach us for improvements, faults, or recommendations?
   15. does the user feel appreciated or satisfied upon exiting the service/function?
4. Product
   1. does the product meet expectations?
   2. does the product meet minimum requirements?
   3. is the product viable and useful?
   4. does it meet its intended purpose?
   5. is it robust?
   6. can it be expanded beyond its initial scope?
   7. does it fail?
   8. when it fails, what happens?
   9. are we properly prepared to release it?
   10. are we prepared to support it?
   11. how long will we support it?
   12. did we pick the best platform or infrastructure for future development?
   13. will the product be properly documented?
   14. can the product be used in other ways beyond the initial intent?
   15. are we prepared to deprecate the product if it is a failure?
   16. do we have an alternative to the product if it fails?
   17. do we have a back up or contingency plan upon failure?
   18. is it a critical system or function?
5. Documentation
   1. has it been documented throughout the entire development process?