QA guidelines/standards

1. Bug reporting
   1. Feedback from other users
      1. Feedback feature from Community
         1. Details should be complete
         2. Should provide steps to replicate or steps on leading the error. We can change the place holder (‘Enter message’) into something to like telling the user to include the steps of how the error occurred.
         3. If it would help, we can add a field with a drop-down list of pages (e.g. Homepage, Post, Message, Sign in or Sign out, etc.)
         4. Need to add a feature to auto-capture the system details (i.e. device, browser/version, native app version, OS, etc.) when submitting a feedback
      2. Feedback coming from emails/teams chat (and others)
         1. Details should be complete
         2. Should provide steps to replicate or steps on leading the error
         3. Need to manually include the system details (i.e. device, browser/version, native app version, OS, etc.) when submitting a feedback
         4. If possible, create a test case for the reported bug. *Need to discuss which parent user story this will go.*
   2. QA creating bug reports from testing
      1. Steps to reproduce
      2. Provide actual results
      3. Provide expected results
      4. Provide Environment (Test or Prod)
      5. If needed, attach photos or screen recording
      6. Provide build number for native
      7. Other system information (browser, browser version, etc.)
      8. Apply tags to the user story (IOS, Android, IOS Safari, Android Chrome, Chrome, Safari, WT, Exploratory)
      9. If applicable, bug should be linked to a test case
      10. Bug should be added to a parent user story (for sprint)
      11. If a bug is detected and was not part of the test case, a new test case should be added under that test plan (https://dev.azure.com/Sequelae/Community/\_testPlans/define?planId=527&suiteId=529 )
2. Community site down
   1. If the site is down, report this to the Community App Team group chat (mentioning Badr or Lydia)
   2. QA will hold off any execution until the app is up and fixed.
   3. Every day, all QA/Developer should take a moment to check out Test and Prod environments just to see if it is up:
      1. Go to the website
      2. Check if sign in working
      3. Check if sign out is working
      4. Other pages
3. Weekly QA meetings – 30 minutes (Lydia to discuss with Badr on the timings)
   1. What is expected deliverable
   2. Assignments
   3. Timeline
4. Push to Prod (first day of the week, Sunday or Monday)
   1. Do smoke test to check if the items being pushed are working as expected
   2. APIs and UI related items should be verified
   3. Refrain from making any test comments into the live post (in PROD)
5. QA process – Software development life Cycle: Agile
   1. Analysis of requirements and define ambiguities. QA can work with the stakeholders.
   2. Define High level scenarios to make sure that requirements are met. This can be covered in acceptance criteria.
   3. User stories should follow the agile format Refer to document “writing-good-user-stories.pdf”, and the same for acceptance criteria – acceptance criteria should be reviewed and approved by stakeholders.
   4. Define test cases with happy and edge cases including required data with steps to follow – test cases should make sense based on the acceptance criteria.
   5. Execution (Manual or automation)
      1. Manual – run all test cases
      2. Exploratory – once all test cases are verified and passed, do exploratory testing to try to see if it will break the feature
      3. Automation – when the feature is stable enough, then convert these test cases into automation code
   6. Impact analysis – when new a feature is put in placed, check to see if this doesn’t impact other features around it by doing regression testing
   7. Test data creation – make sure that test data are readily available. Reach out to other teams (data team, business side, etc) to help create test data
6. Test Types:
   1. Functional – this includes security too,
   2. non-functional – performance and load testing,
   3. static-testing – reviewing documents of requirements,
   4. Regression testing or re-test – when new features are put in or bug is fixed
7. Test levels:
   1. Component test – like sign in , sign up separately and make sure each works as expected
   2. Integration test – like after create a post he can find it in the home page
   3. System test – testing the whole system including all components/pages.