HIPAA Compliance Checklist

1. Avoid accessing, displaying or storing data you don't need. For example, if you don't need full birth date then don't gather it. Any personal info you ask for should have a clear purpose. **(Design Team)**
2. App must have a clear privacy policy. **(Client)**
3. If your App is sending text notifications & SMS and MMS are not encrypted, so make sure they don't contain PHI (Protected Health Information). **(Client)**
4. Local session timeout — your app should certainly force re-authentication after inactivity for a certain duration. This should be a PIN or fingerprint access. **(Client) - Take final call here**
5. Recommandation: Force Logout — your app should logout after 2-3 days. **(Client)**
6. Ensure that PHI is never sent to push notifications that could easily be seen by someone other than the patient it pertains to. **(N/A)**
7. Two-factor Authentication: **(Client)**
   a. Additional code provided on SMS/Email
   b. Security question/answers selected by users in advance.
   c. Including passwords plus additional user known values. **(Design Team)**
8. Strong password policy should be implemented in App. **(Client)**
9. Go for a third party check for Hipaa Compliant before go live for example Fortify App Environment Check / Pen Test. This is a paid service. **(Team) - Anshuman will check this.**

Checklist For Developers

1. Avoid incorrect use of Android Intents.
2. Use a standard encryption mechanism instead of creating your own algo.
3. Fully validate SSL. Use HTTPS for data transmission in encrypted form. Certificate Pinning.
4. Avoid memory leaks
5. Do not keep sensitive data like encryption keys in RAM for longer time. Nullify the variables that hold key after use.
6. Avoid using mutable objects to contain sensitive data like password/keys, for example use char array instead of String.

7. Do not store data in external storage.
8. Encrypt sensitive values in DB using SQLCipher & column naming should also be considered here.
9. Proguard must be enabled in App.
10. Data must be transmitted securely and stored securely using APIs.
11. When encrypting data locally, use widely tested protocols based on some sort of standard.
12. Enable Lint checks
13. App must not log any error or message on console in release mode. Avoid SOPs as well.
14. We should not name a key or variable like ‘password’, ‘private’ or ‘username’.
15. Never throw exceptions from finally block.
16. Use Secure Random number instead of Random Number.
17. Code should not have unused method and variables.
18. App should not have an empty catch block.
19. Catch block should handle specific exception instead of Broad catch.
20. A constructor of a class should not call a method which can be overridden by child classes.
21. Don’t use classes that extends java.text.Format because parse() & format() method of this class contain a design flaw that one user can see data of another user, hence avoid use of class Simpledateformat.
22. There should not be any Non-final public static field cause it can be changed by external classes.
23. Catching Null pointer exception is a bad practice & can be highlighted on a Pen Test.
24. Release resources acquired by a stream in finally block.
25. Cryptographic encryption algorithms should not be used with an insecure mode of operation.
26. Add remote_host and refere_host check for each api call at server side.
27. CORS control at server side.
28. Add Anti Cross Site Scripting(XSS) for each post/put apis.
29.