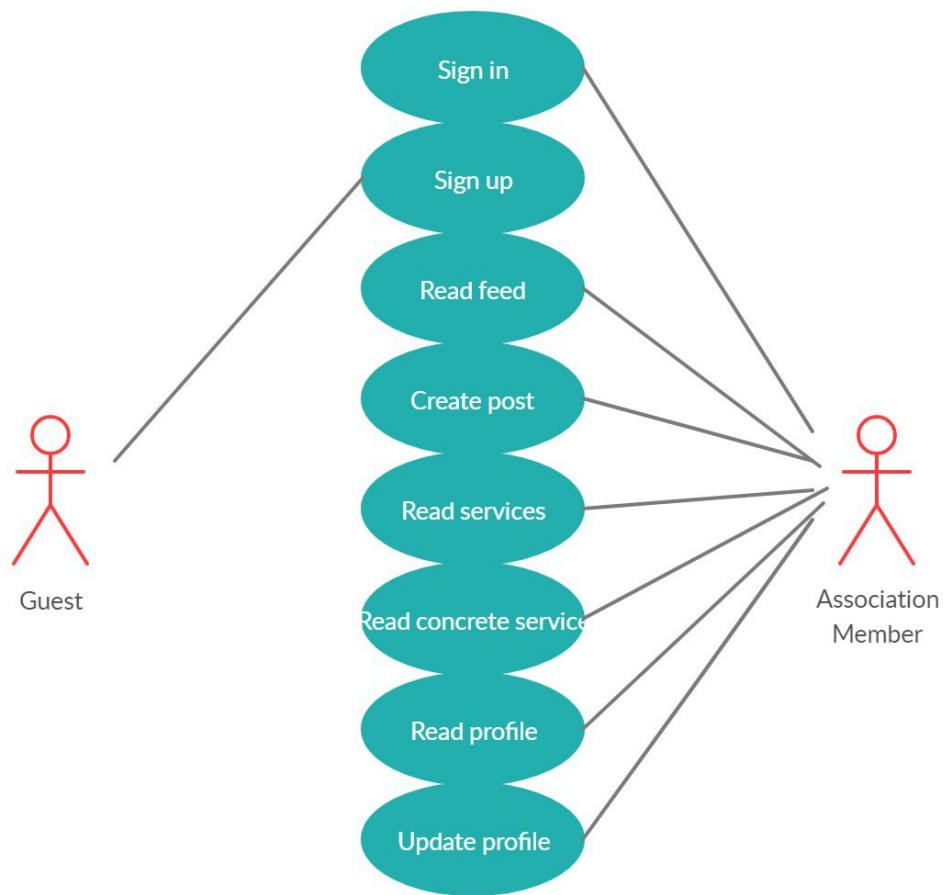


1) Use cases:



2) Storage:

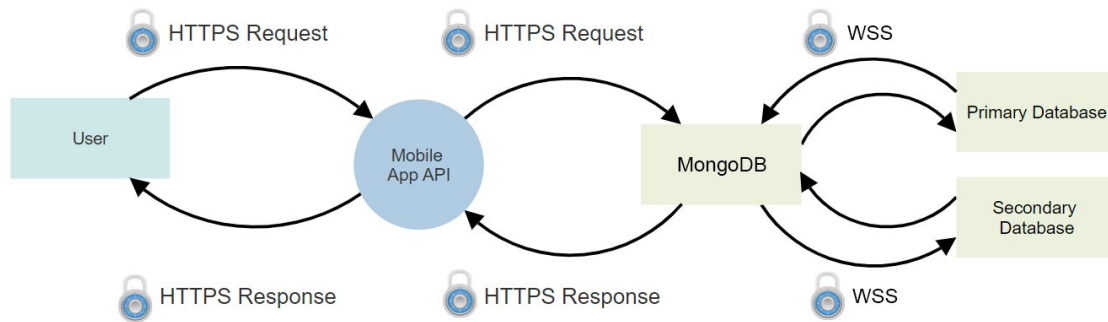
MongoDB

User	
Id	String
IdentityToken	String
Name	String
Email	String
Phone	String
LinkToPhoto	String

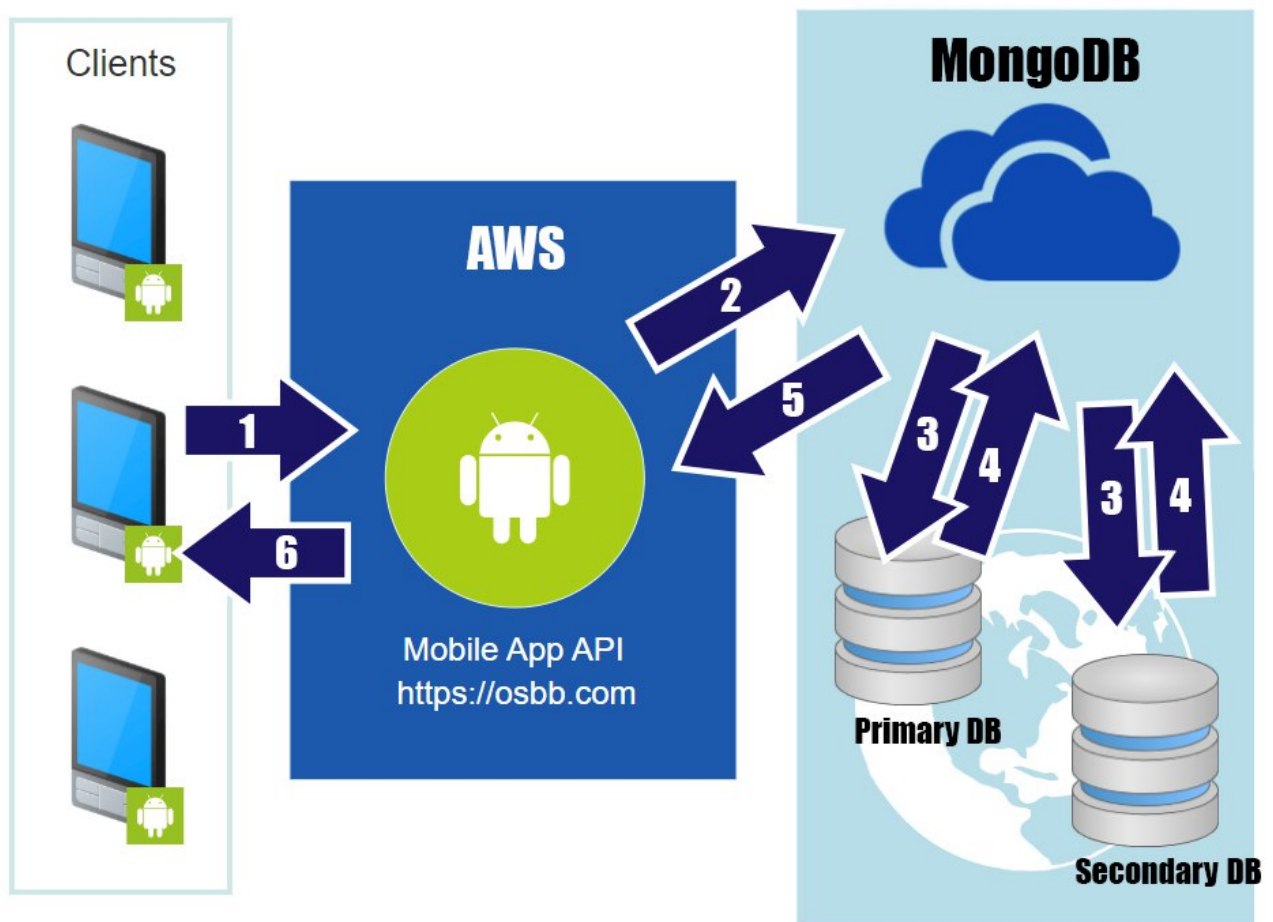
Post	
Id	String
Title	String
Description	String
LinkToPhoto	String
TimeStamp	Double
UserId	String

Service	
Id	String
UserId	String
Name	String
Description	String
Phone	String
Email	String

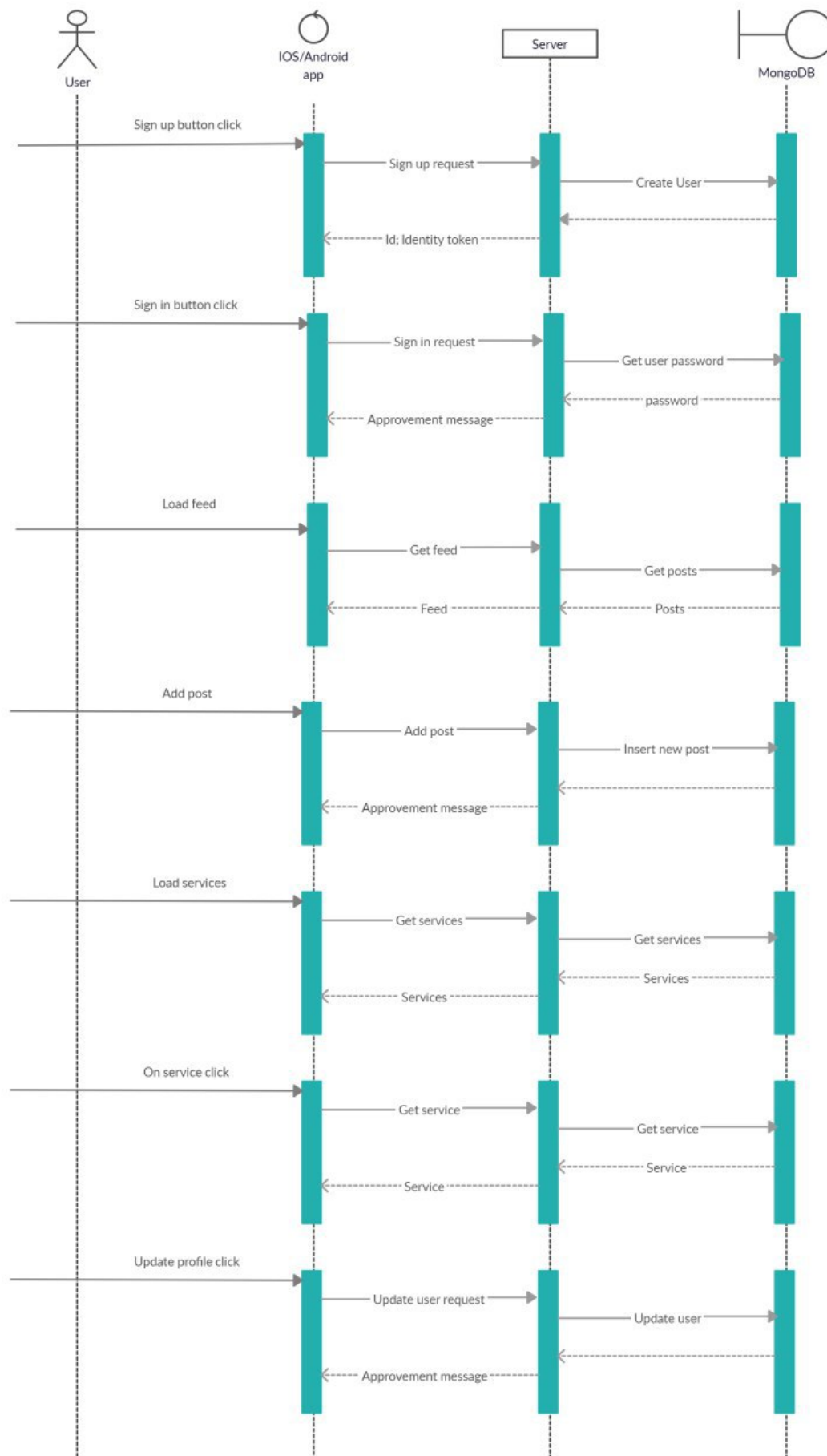
3) Security diagram:



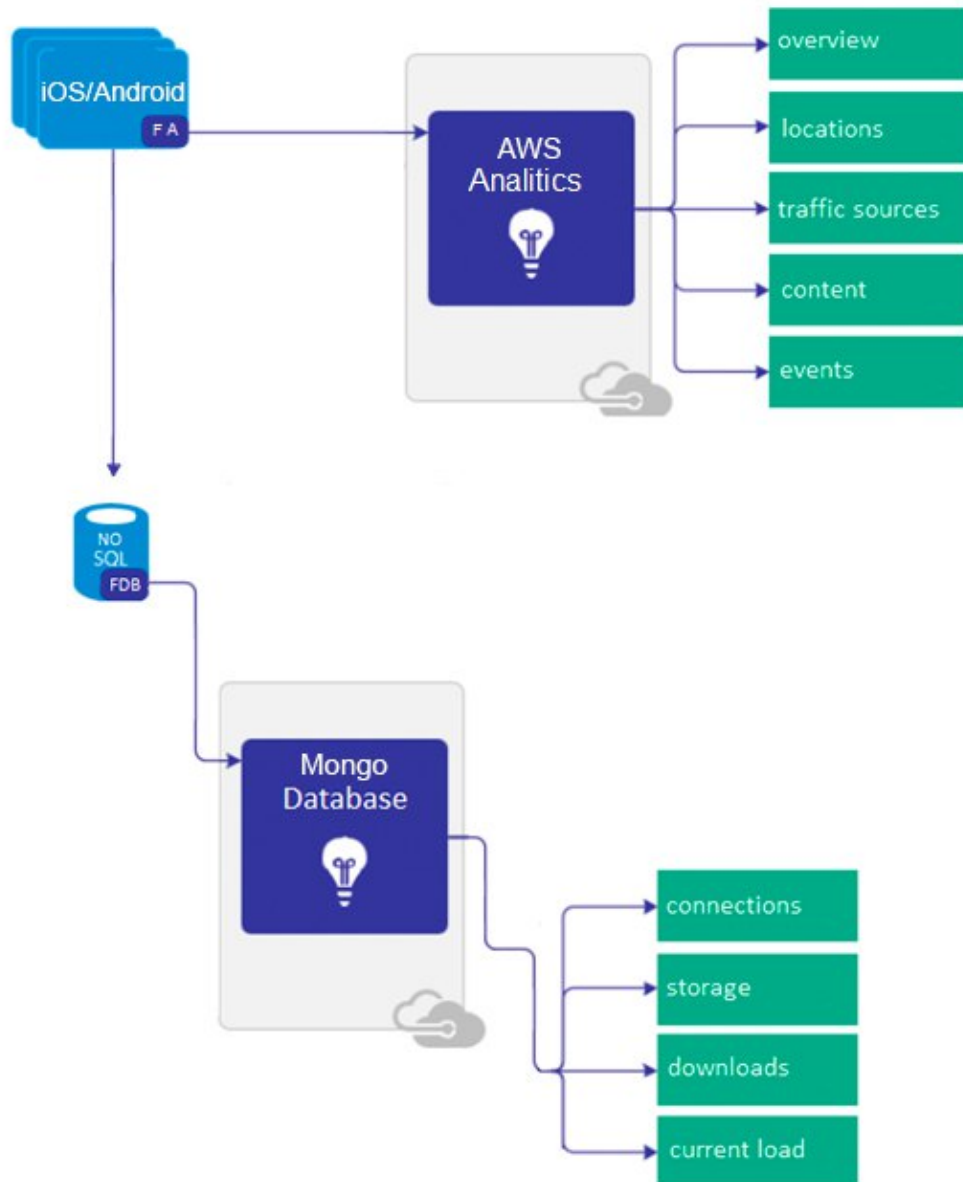
4) Hosted services:



5) Sequence diagram:



6) Telemetry and monitoring



7) SPA:



8) Resilience model prework:

ID	Component/Dependency Interaction	Interaction Description
1	Internet Client -> Mobile App	End-user client connects via Internet to the Android/iOS mobile app from their smartphone
2	Mobile App -> Remote Server	Mobile App sends a request to one of service endpoints (CRUD)
3	Remote Server -> DB	Server accesses the application DB and makes corresponding changes
4	DB -> Remote Server	DB returns a response with the result of requested changes
5	Remote Server -> Mobile App	Remote Server sends a response to the respective response of the mobile app
6	Mobile App -> Internet Client	Mobile App updates related UI and data for the end-user

9) Resilience model:

ID	Component/Dependency Interaction	Failure Short Name	Failure Description	Response
2	Mobile App -> Remote Server	Create::No data	Mobile App sends empty data	Remote server returns a 400 Bad Request response. Mobile App retains state and shows an error message that says some data should be present for the request
2	Mobile App -> Remote Server	Create::Incorrectness	Mobile App sends a request that differs from what server is expecting	Remote server returns a 405 Method Not Allowed response. Mobile App retains state and shows an error message that says what data should be present for the request
2	Mobile App -> Remote Server	Get::No id	Mobile App sends a get request without specifying the id of the requested data	Remote server returns a 400 Bad Request response. Mobile App retains state and shows an error message that says the request must have a unique identifier
2	Mobile App -> Remote Server	Get::No id	Mobile App sends a get request with an id that does not exist in the DB	Remote server returns a 404 Not Found response. Mobile App retains state and shows an error message that says there is no data with that id.
2	Mobile App -> Remote Server	Get::Incorrectness	Mobile App sends a request that differs from the expected get request	Remote server returns a 405 Method Not Allowed response. Mobile App retains state and shows an error message that says how a requested get call should look like
2	Mobile App -> Remote Server	Delete::Incorrectness	Mobile App sends a delete request with an id that does not exist in the DB	Remote server returns a 400 Bad Request response. Mobile App retains state and shows an error message that says how a requested delete call should look like
2	Mobile App -> Remote Server	Delete::Incorrectness	Mobile App sends a request that differs from the expected delete request	Remote server returns a 405 Method Not Allowed response. Mobile App retains state and shows an error message that says there is no data with that id
2	Mobile App -> Remote Server	Update::Incorrectness	Mobile App sends a request that differs from update	Remote server returns a 405 Method Not Allowed response. Mobile App retains state and shows an error message that says you must send an update request
3	Remote Server -> DB	Latency::Timeout	Functions try to make a write or read from the DB, but it times out	Could functions timeout after 60 seconds of inactivity from the DB and return an error status