* **Use Cases**
  + Showing all the activity/fragment pages options by guest or user .
  + Show use cases/actions person can do in each page.
  + Some actions are conditional if person is User or Guest.

* **ERD**
  + Represented in “SQLy” form although Firebase stores tables as JSON documents (NoSQL). Each entity is provided with its corresponding JSON entity example.
  + Arrows representing all relationships like 1 -> many (mandatory/optional) etc
  + 1 to many relationships are flattened out to different tables to avoid deep nesting (e.g Post -> Comment)
* **Class Diagram**
  + Our 6 top classes represent the base functionality components and their flow from UI to middleware (Presenters) to our datastores ( Model ).
  + Our architecture follows the Clean MVP pattern used by the most popular Android apps.
  + Good article about this architure is http://macoscope.com/blog/model-view-presenter-architecture-in-android-applications/
  + Since we still have not started implementation I’m sure there will many Android UI specific classes which are not specified in the diagram.
* **Sequence Diagram** 
  + Flow that describes the timeline of the application from both viewing points: guest and user and its corresponding available requests and responses
  + Axis Y we have the function/action that sends requests to our datastore entities (Firebase/Db/Api).
  + Axis X we have the entities that run the actions/functions.
* **State Machine Diagram** 
  + SM 1 - Describes the moment person opens the app and after that in the MainActivity which holds all other fragments (our core functionality) and responsible for app navigation. Diagram shows all the possible navigation in app taking into account the User state (Guest/Logged In)
  + SM 2 - Described the possible lifecycle of a user within the app.