

Yusuf Bahadur

Lauren Hibbs

### Critical Analysis Assignment

- Are there technical problems with this design?



- ~~The Appliance Management Use Case Diagram is flawed in that the Property Manager is both Creating an Appliance, which would update the Appliance in the Backend. However, the Property Manager also has a use case for Approve Appliance after the Appliance has been added to the database.~~
- The deployment diagram shows ReactNative running on a web server when React Native would be deployed locally on the user's device

- Are there major omissions in this design (e.g. functionality, security, performance)?



- ~~The Entity Relationship Diagram is missing the relationship type between entities.~~
- Authentication Flow lacks a Forgot Password Flow.
- ~~The team lacks a sequence diagram.~~
- ~~The Browser Clients connect to a database to store information not stored in the Roopairs API. However, the Mobile development lacks access to a SQL or NoSQL Database.~~
- When creating a chatbox, you want to ensure you have two databases. One local database and one online database. The local database needs an observer to watch over the updates for a specific users thread on the online database. When the online database gets updated, the observer must notice the change, and then proceed to download the new information onto the mobile database.
- A dataflow diagram would clarify the general flow of data through the app which is somewhat unclear.
- The type of backend SQL database is not specified. The type of database selected might affect the non-functional requirements of speed.
- In Use Case Diagram in Figure 5 it is unclear why the database is not located between the send message use case and the tenant and service provider actors.



- Is it feasible to implement this design by this team in the timeframe allotted?
  - It seem like a stretch goal for the team to implement a chatbox within the application. Maybe the high priority system features should be modified so they are not reliant on the chatbox.
  - Building a React Native and Web Application seems entirely feasible.
- Does the design satisfy the customer's functional requirements?
  - The customer wants a component on the main screen for users to track their repair analytics.
  - ~~The customer wants the landlords to grant other users custom sublandlord privileges. User account authentication or privileges are not mentioned in the design document.~~
- Does the design satisfy the customer's non-functional requirements/quality attributes?





- The non-functional requirements mention integration with Roopairs and Stripe. If the integration with Stripe is not through Roopairs the design needs to include an integration with Stripe.
- Nothing is stated in the design document about how the data will be encrypted. Is the team planning to implement their own code for authentication to the database, or use a Google Cloud or other external user authentication service? Will the team implement their own data encryption or do they plan on using an external library?
- REST APIs are a good choice when speed of retrieval is important, so assuming a good backend SQL server is chosen the application should be able to satisfy the nonfunctional requirements of speed.
- The chat room requirements of the non-functional requirements may not be able to be met but this is discussed in the software architecture document.