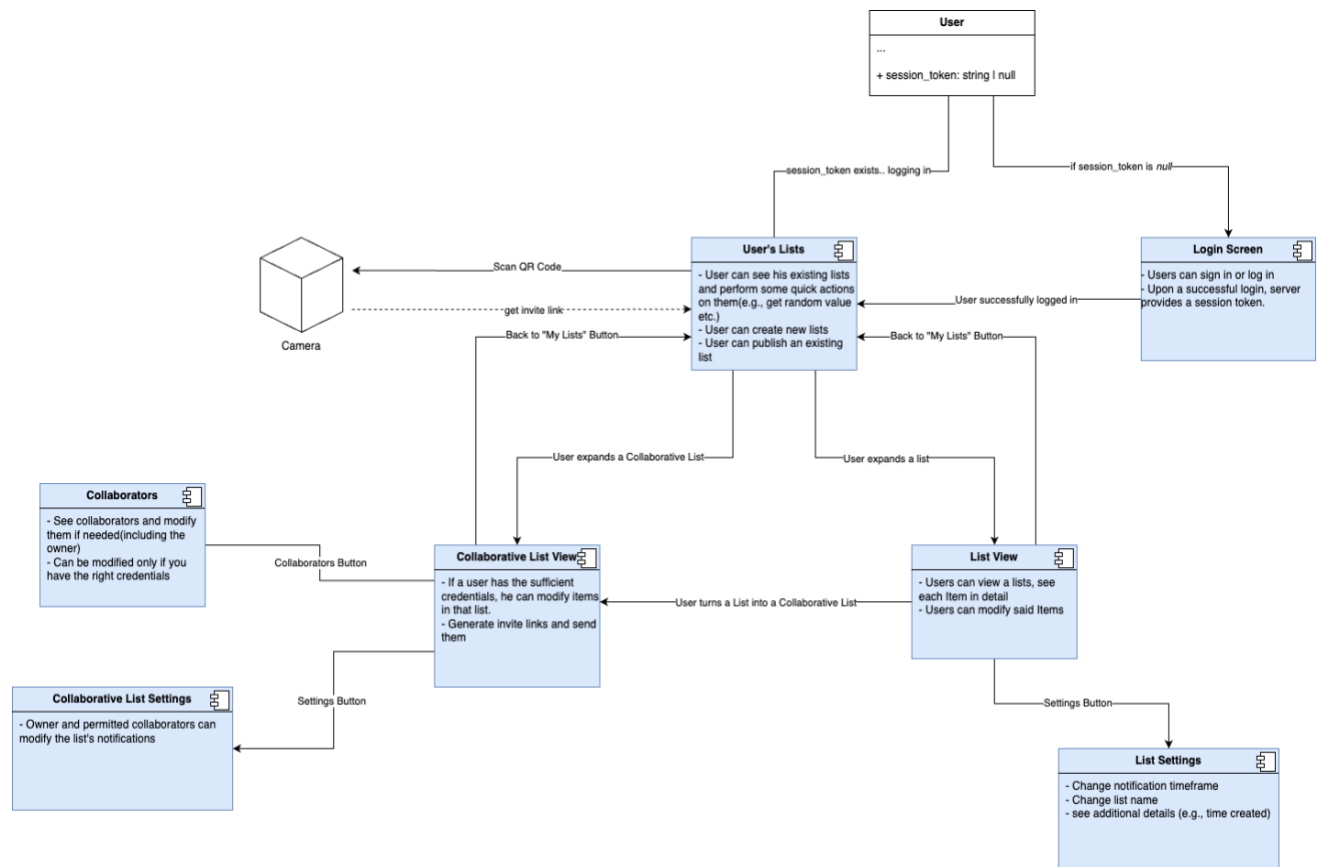


# Analysis Document for ProcrastiNOT - Software Engineering

יהונתן ברוך 211465786  
יבגני איונוב 324780246  
קורן עבדוש 209271535

## Screens Diagram:



## MVVM, MVP, or MVC:

Reminder: we've decided to use React Native for our App.

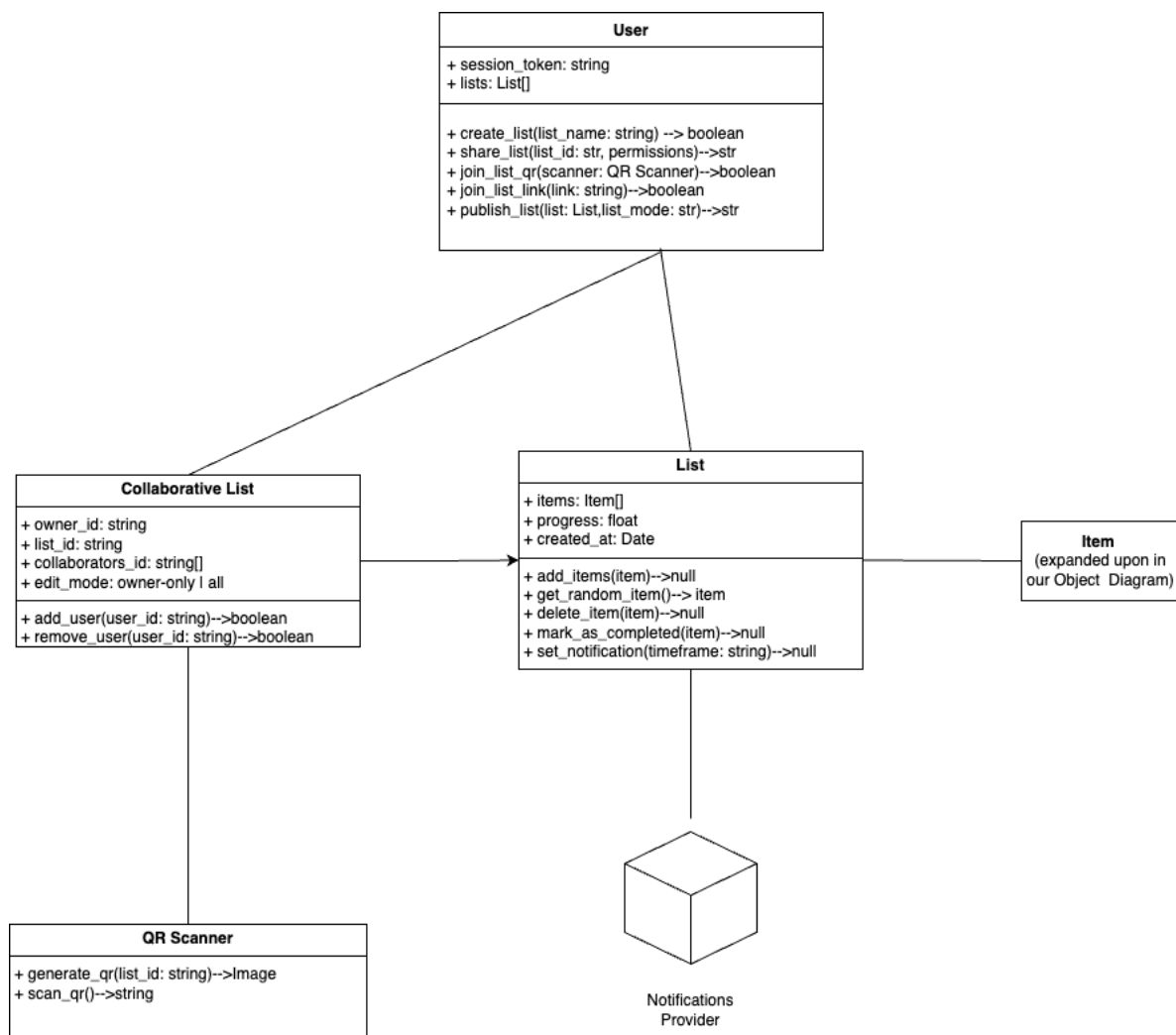
We've decided to use the MVVM architecture for our application because React Native's Context API naturally serves as a bridge between the ViewModel and View layers. In our case, the Context API holds the state and logic of the application while exposing observable data that the View reacts to, allowing a dynamic user interface.

The benefits of this choice are clear: we maintain a clear separation between UI and logic elements, ensuring the View remains focused on rendering UI while the ViewModel handles application logic and state management, leading to great maintainability and easy debugging. As for scalability, our project is small and composable and it should be easy to add new components and scale the project vertically if needed. Finally, performance should be great - React Native is widely and commonly used across the industry along with the Context API to develop multi-platform applications with great performance. Considering our current scale of the project is quite small - it should not pose an issue.

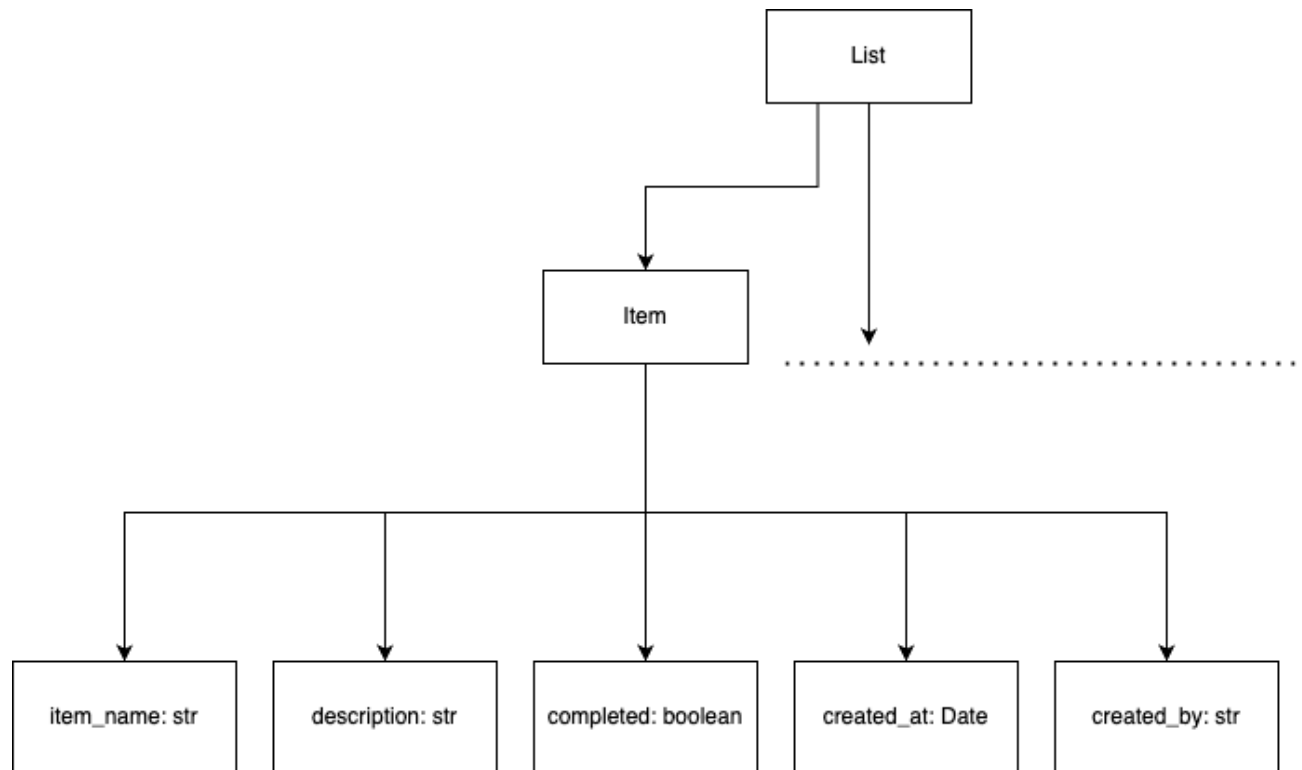
### Reminder of our 3 significant processes in our system:

1. List Creation & Configuration
2. List Collaboration
3. Progress Tracking via Notifications & Random Selection Tools

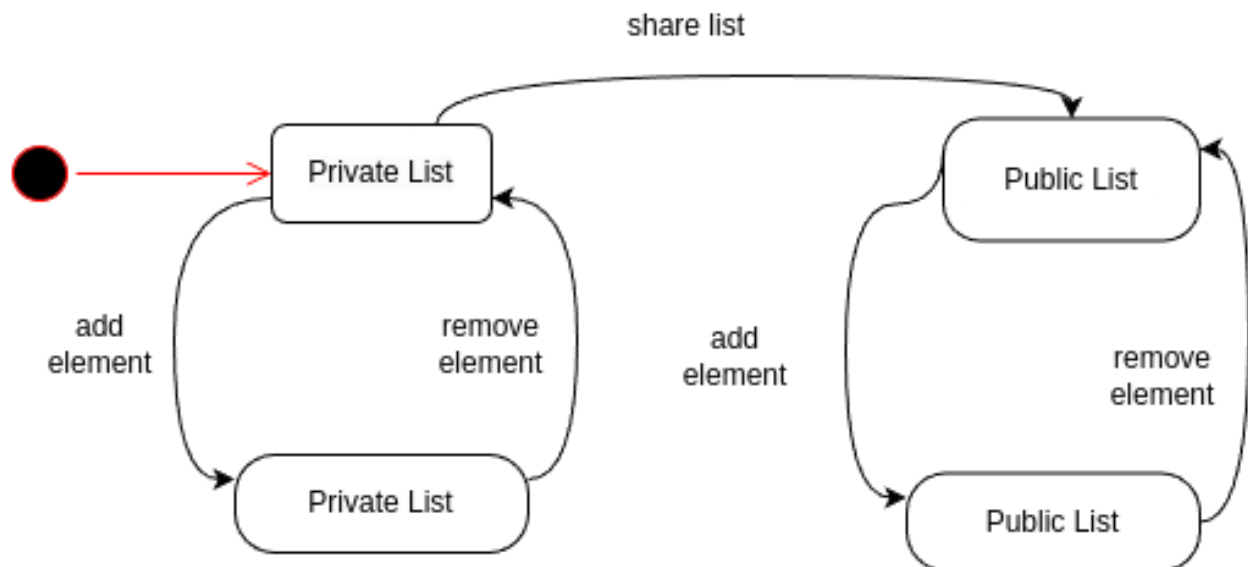
### Classes Diagram:



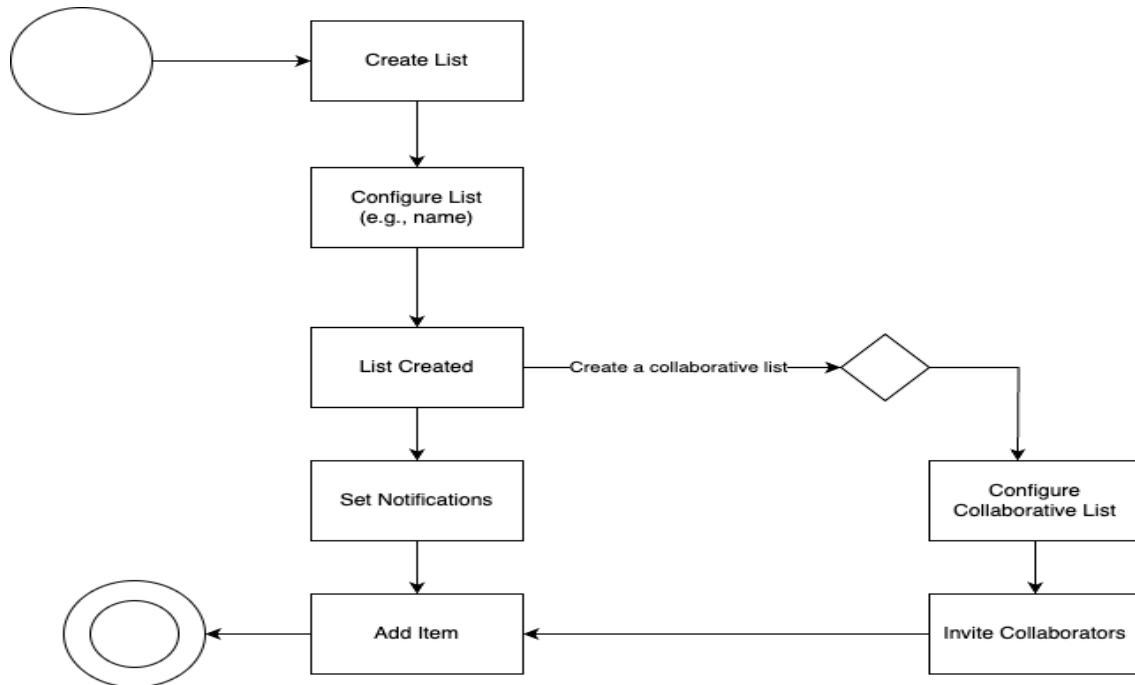
### Object Diagram:



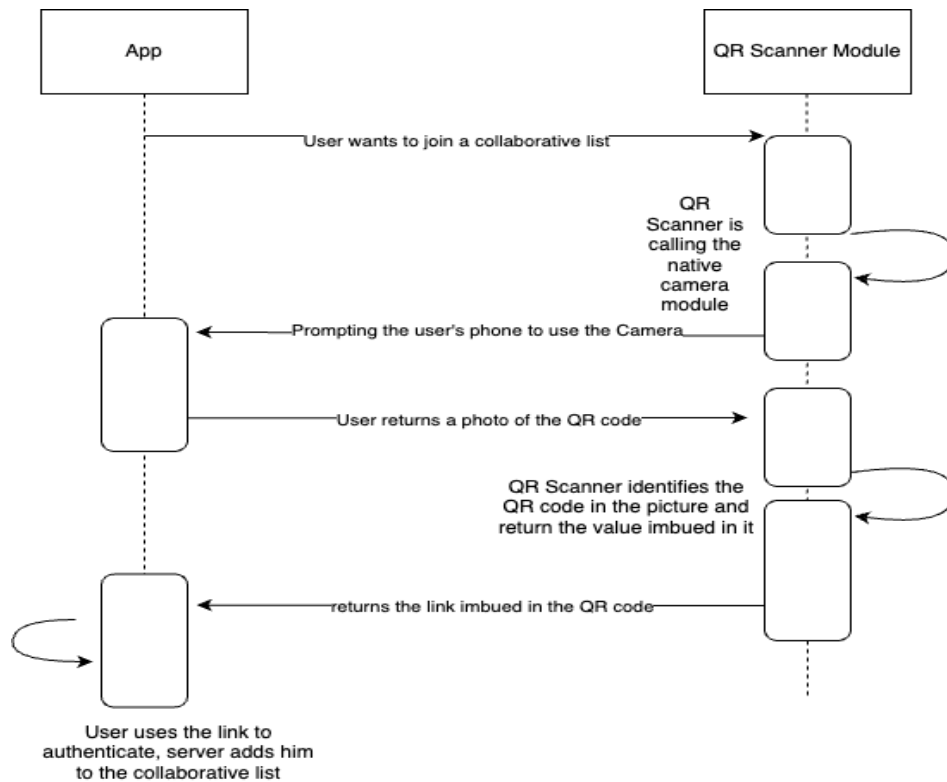
### State Machine Diagram:



## Activity Diagram:



## Sequence Diagram:



**ERD Diagram:**

