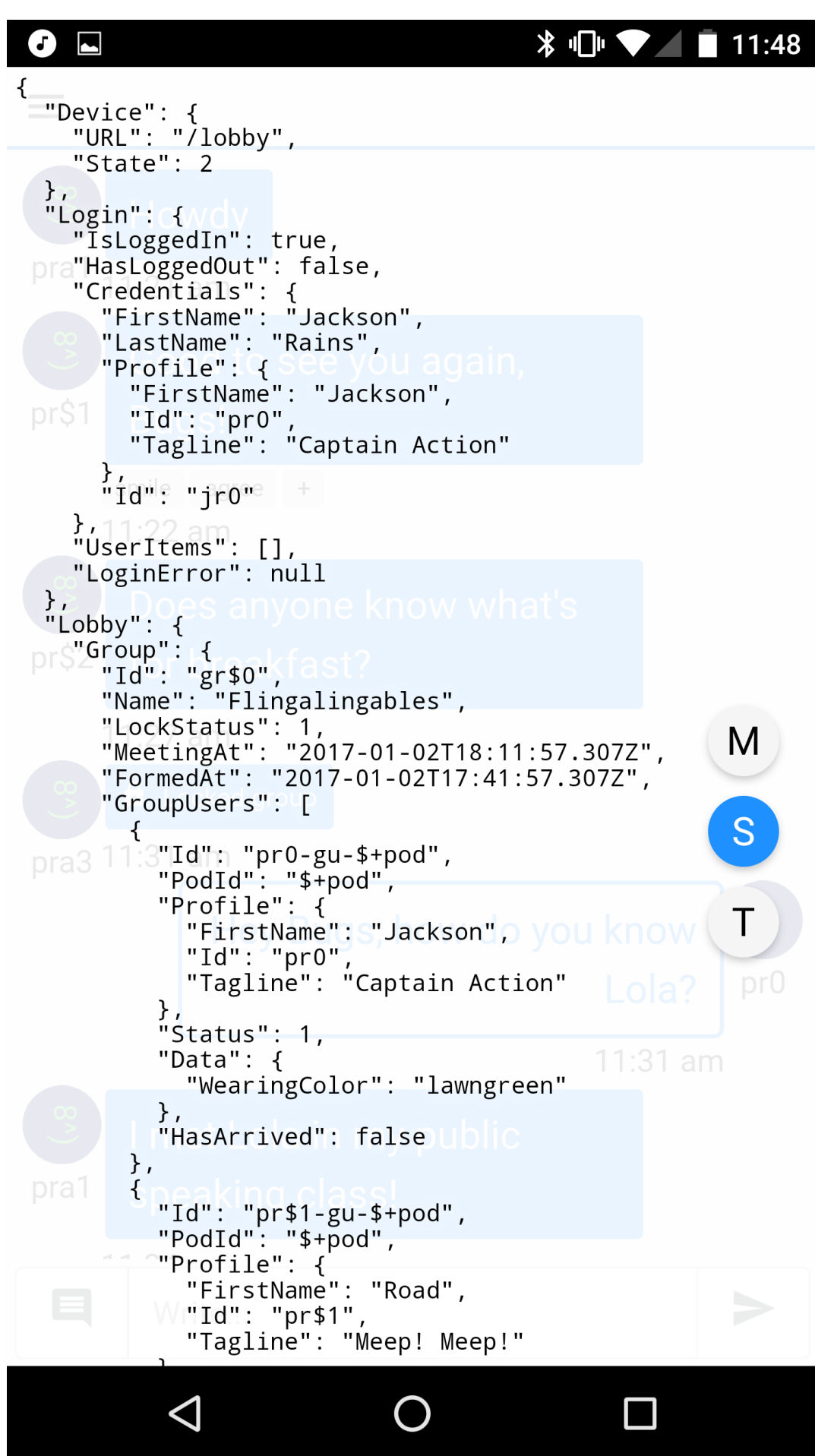


Project-W Progress Report

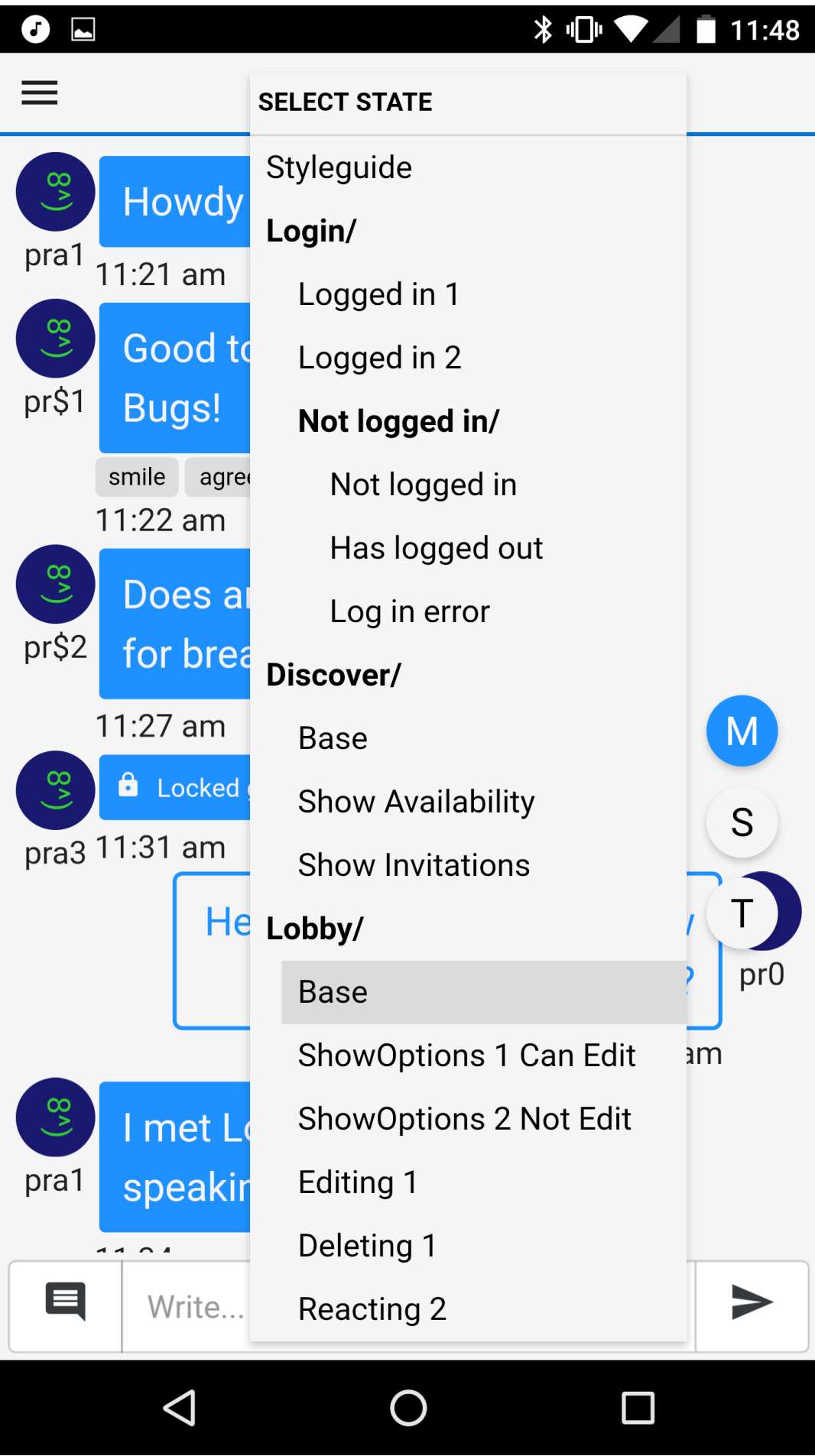
2017-01-02

Developer Tools



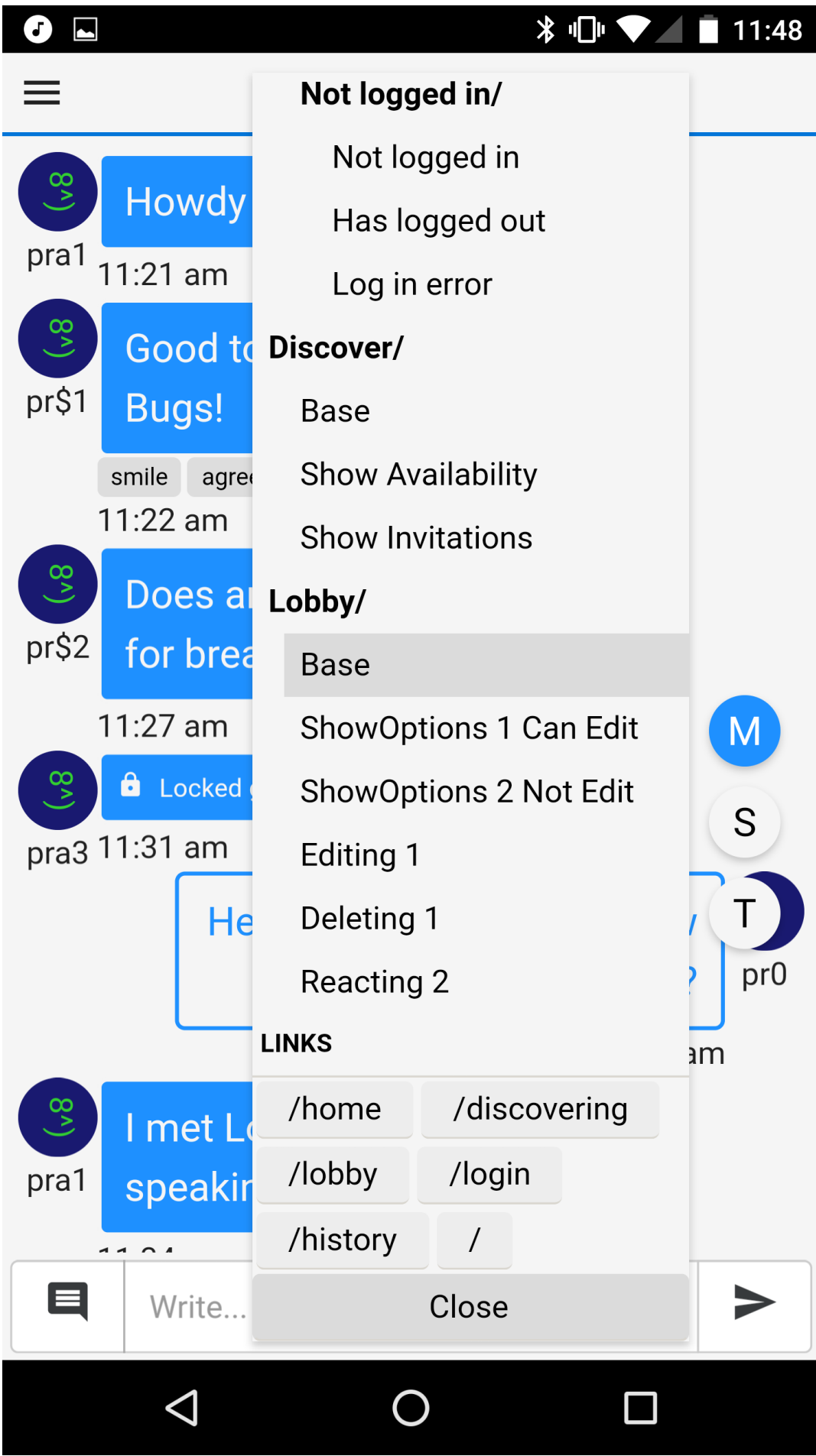
1.21 Dev Active State of Application

See what the data structures at the heart of the application look like. This data is everything the app knows!



1.22 Mock State Selection

Quickly load up a different preprogrammed state for testing UI states. Each entry is a complete new AppState put into place of the Active State



1.23 Mock State Selection 2

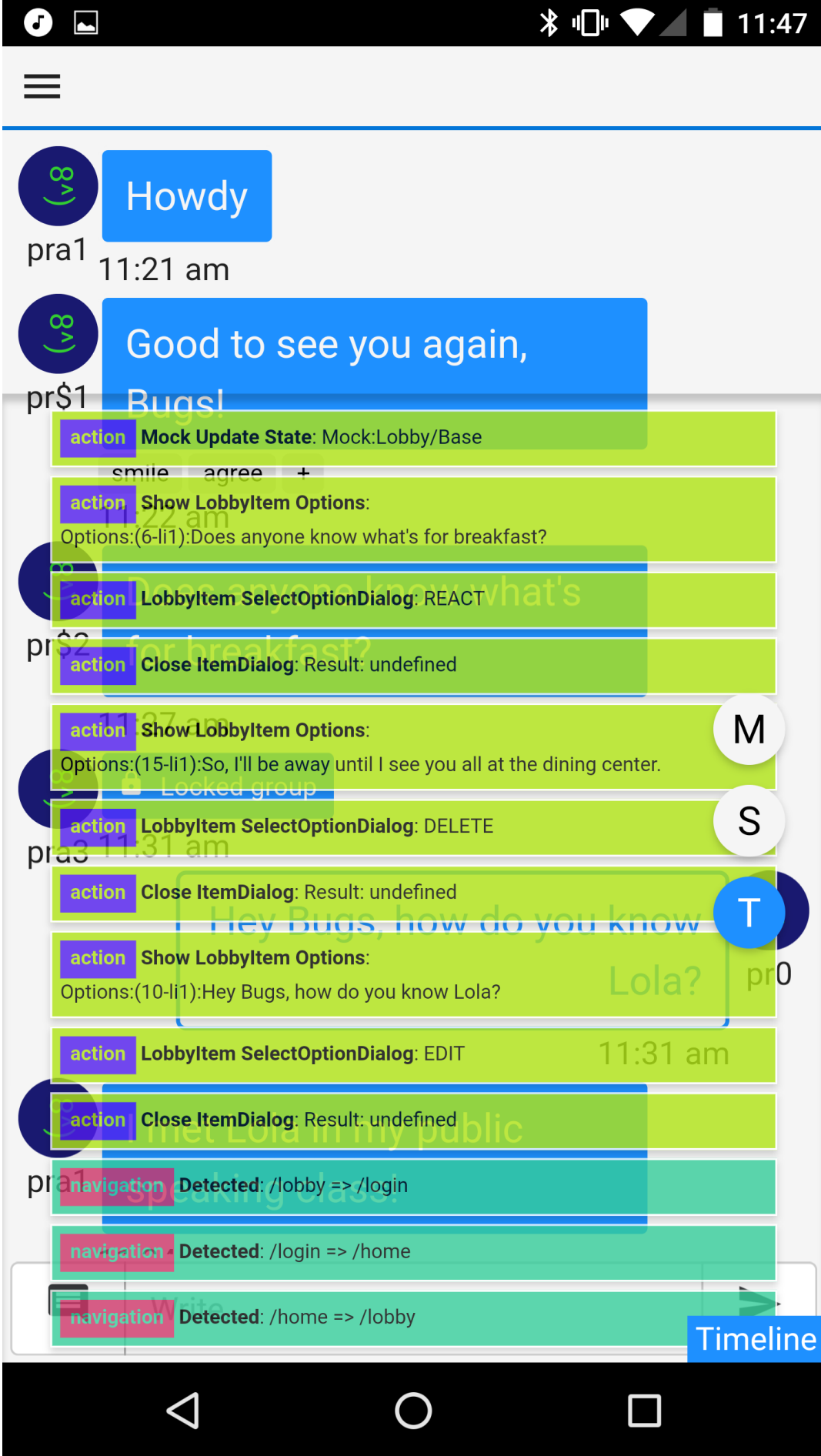
Notice access to links to different routes of the App.

AppState Architecture

The Architecture of managing the AppState is heavily influenced by Elm's Architecture (which we studied when we were choosing the framework to build this frontend with), and shares many properties of React + Redux patterns, or CycleJS. We aim for app state management to be as simple as possible, so we many times will completely forego Outputs in our components and call actions from the child components directly. This is such in our SelectOptionDialog, where instead of resulting in the choice when closed, we call a change in the app state directly for simplicity. Then, by consolidating all data flows in the AppStateService we can record Timelines of changes that have happened as shown to the side (1.20). Little work is left to completely enable timetravelling forwards and backwards which may come in handy for replaying actions leading to a bug report!

User Analytics

For simplicity, I believe that we can use this timeline to help generate the user data Zachary Sewell will need for tracking user behavior. Likely, the only thing left that we'll need is a Directive which logs user actions to the AppStateService, so we can do something as simple as `<div ua-track="click" ua-id="lobby-btn-message-type">` or `<input ua-track="input" ua-id="lobby-message-input">`, etc. Then it should be trivial to style these ua-id elements using a `[ua-id]::before { content: attr(ua-id); }` type of approach. Or create data about positioning or sizing whatever is needed to know about the element the tracker is attached to.



1.20 Dev Timeline of Events

For debugging purposes and bug reports, we have a robust architecture for storing user actions over time.