Finite State Machine Hackathon part II

Objectives

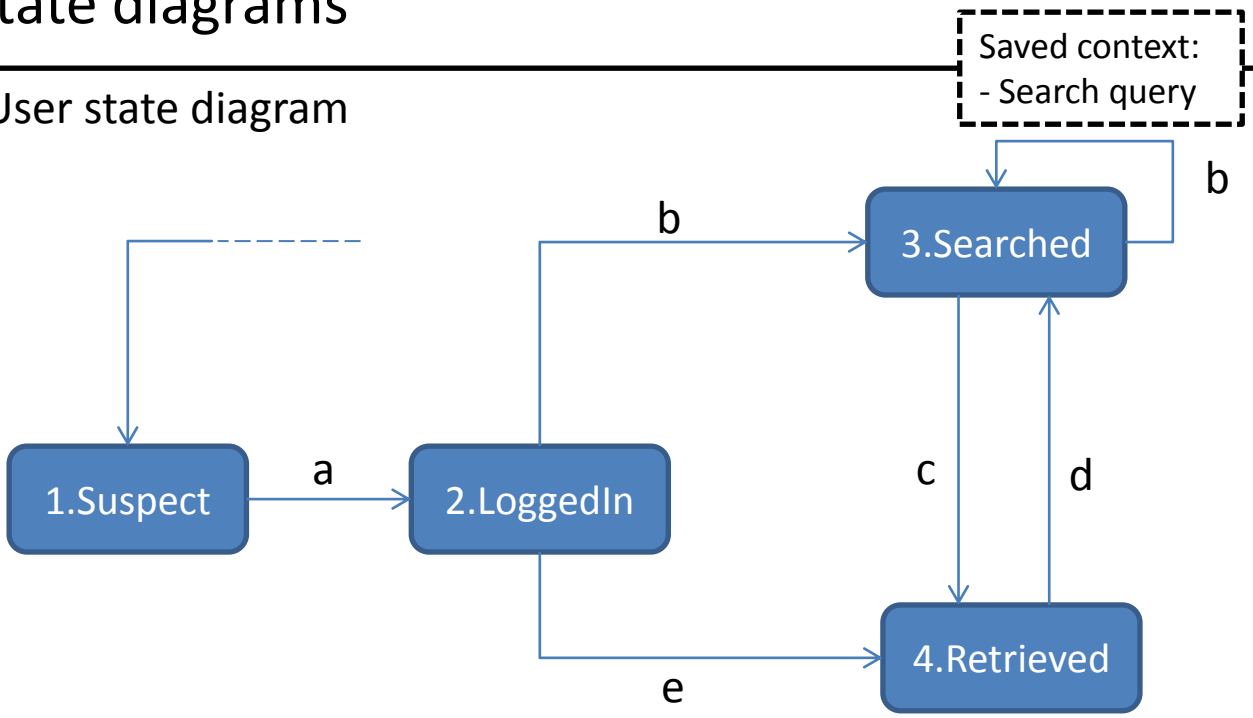
1. Demonstrate Finite State Machine concept
2. Demonstrate State sharing across devices
3. Demonstrate State sharing across users

Scenario

- User ab12cd logs in
    - and searches customer John Johnson
    - He retrieves customer John Johnson
    - He **swipes** John to the dialog box, so his colleague can take over
- Then user qr34st logs in
    - Checks the dialog box
    - Picks up John Johnson

State diagrams

User state diagram



Transitions

a. User logs in
b. User searches a customer
c. User retrieves a customer (either from search result or dialog box)
d. User swipes customer to dialog box
e. Other user retrieves customer from dialog box
f. Log out (from anywhere)

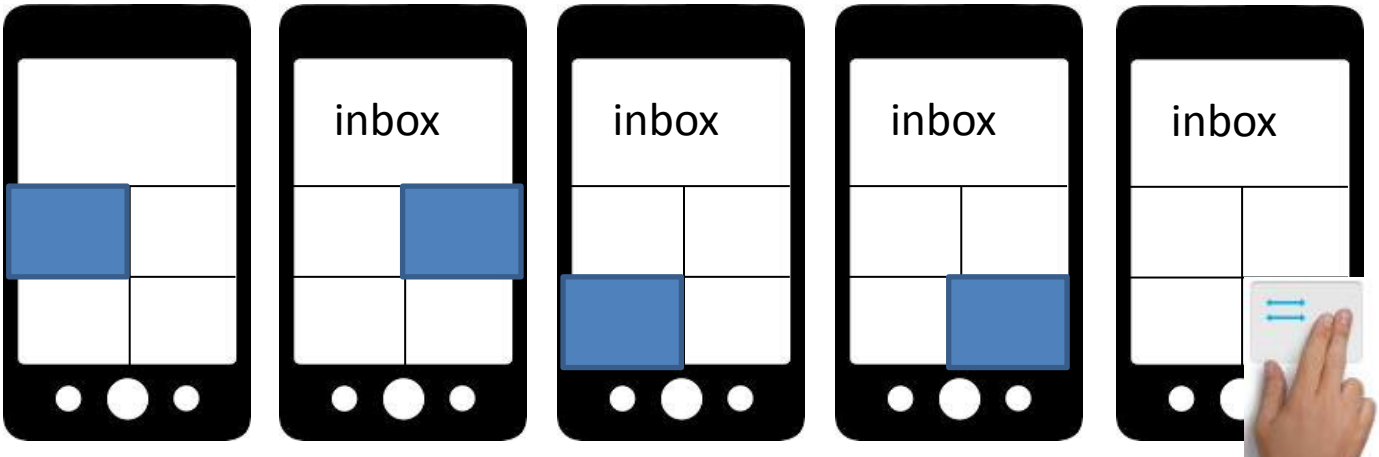
Dialog state diagram

Table version of state diagram

To→	Suspect	LoggedIn	Searched	Retrieved
Suspect		x		
LoggedIn	x		x	x
Searched	x		x	x
Retrieved	x		x	

States will be reflected in the screens

1. Suspect
    1. Login available
2. LogginIn
    1. Log out available
    2. Search available
    3. Dialog box available (if filled)
3. Searched
    1. Log out available
    2. Search available
    3. Search result available
    4. Retrieve available
4. Retrieved
    1. Log out available
    2. Swipe available



Log in | Search | Search result | Selected client | Swipe to inbox

Database

User table elements

- corpKey
- state
- searchQuery
- partyId

Dialog box table elements

- partyId
- state

Teams

Inbox team
Create screen component, push updates to inbox (socket.io).

Front-end team
Create screens components, handle flow, call API, implement swipe.

API team
Create API, return allowed states, persist state and dialog.

Persistence team
Persist state and dialog.