CSE 360 Meeting 4

State Diagrams

* “in this class what are the input and possible outputs that change the class”
* a state should not have a verb or action associated with it
* should capture what the system is doing
* don’t include extra states
* for example: a state that says speed is increasing, or speed is decreasing
* the state diagrams that we have are more similar to sequence diagrams
* a state would be you are in the radio view, nothing changed. Then user input changes radio frequency, we have two states, one for each frequency
* Login state diagram
  + There are two states,
    - Correct username and password
    - Incorrect username or password
  + What input would change the state?
* State diagram shouldn’t show database
* Should have one for car, one for phone, one for radio (check rubric)
* Focus on inputs, not actions
* For car:
  + There is a state where car is moving
    - Increase speed takes you to the next state
* Show the characteristics of the system

Sequence Diagrams

* Captures interactions
* We need three
* Return values are by dotted line, put the name of the variable returned
* There should be a boomerang effect, if it goes out, it goes in
* Start car needs to move down a level
* Create a new instance of dashboard, then simulate a user pressing buttons to ‘drive’
* Have looping arrows for key pressed and key released
* Need to include bottom bar controller
* Think of each lifetime as a method
  + When does the method go out of scope?
* Maybe make the sequence diagrams smaller?
* See picture below
* Look at the code and see where the first brace ends and where the return value is.
* Put an ‘X’ where the lifetime ends

Use Case Diagrams

* Everything needs to be inside the box
* Phone
  + Adjust volume is an action with two specialized versions, up and down
    - Volume up and down are generalizations
    - Change the includes to generalization
    - Arrows need to face the other way around and say generalization
    - Or keep the arrows facing that direction and have it be a specialization
  + Dial number
    - Put call directly pointing to actor
      * Calling includes dial a number
      * Save number extends dial
      * Don’t include end call
      * See pictures below
  + Radio
    - Change channel should be a generalization similar to volume from phone
    - Volume will be the same as phone
    - This diagram is otherwise ok
  + Drive
    - Change accelerate, decelerate and coast to be generalizations of drive
  + Start car
    - Change to personalized key
    - Maybe change to generalizations of enter username and enter password
  + History
    - This diagram is good as is
* Check the rubric to make sure we don’t need more

Users Guide

* Need set up instructions as well as functionality instructions
* Put directions to edit the code (what Mario has now)
* Also include directions to set it up for a normal user (how to run the program)
* Include the predefined username and passwords but make it clear that these are ‘keys’ they aren’t passwords that need to be kept confidential

Testing

* We need to have a testing plan
* We can’t test everything
* Test the most important methods
* Integration testing
  + Combine elements
  + Test if the car can interface with phone
  + Example: can you load phone view from dashboard view?
  + We could test the scenarios that are shown in our sequence diagrams
  + Show how different elements work together
* Validation testing
  + Garret will get back to us, he will send us an email
  + Does the code meet the requirements that we specified?
  + Can we meet the requirements?
  + Ex: if we say the password can only have 8 characters, then test that more characters or less does not work
  + In our case, test that max speed doesn’t go past 300, max volume, min miles, car can’t be started with no gas, etc.



