IPO Chart

Component: Orchestration

Module Name: Service Broker

Author: Sydney Nguyen

Reviewers: Cameron Lohman

Inputs	Process	Outputs
Integer Service code	Calls each service	None, since it calls the selected service
String userInput for service		

Inputs:
Service code: corresponding code which gives permission and search for number conversion, skills search, or translator.
String userInput: number conversion, skills search, or translator

Process Description
It will take in a service code and a parameter list which then executes the service corresponding to its service code. Passing in the parameter list will output the data returned from the service.

Outputs
Data returned from the Business Services module

APIs/Objects:
Scanner object
File
HashMap
Class<?>
java.io
java.util
java.lang.reflect

Pseudocode:

MAIN:

1. Create HashMap called serviceMap = readServicesFromFile()
2. Create scanner object
3. Initiailize userInput to an empty string
4. Prints available services to the user using printServiceOptions()
5. If there's still input to be read, retrieve the user's desired program and save it to userInput
    a. If the user's input == "exit"
        i. try
            1. Parse userInput as an integer and save it to serviceCode
            2. Using serviceMap, get the value paired with serviceCode and assign it to serviceName
            3. if the serviceName is null, call UtilityManager to raise an error

IPO Chart

4. Else, call Class.forName(serviceName) and assign it to serviceClass
5. Run the main method for the called class

ii. catch NumberFormatException, NoSuchMethodException, or ClassNotFoundException and output a UtilityManager(404) error

iii. catch any other exceptions and output (UtilityManager.getMessage(Integer.parseInt(e.getMessage())))
end try

end if

end if

end main

HashMap<Integer, String> readServicesFromFile() throws FileNotFoundException:

1. Initialize and declare a hashMap called serviceMap
2. try (Initialize Scanner object called input and pass in File object to services.txt)
    a. while there's still input to be read from the scannerobject
        i. Create a String array initialized to line and split each string in services.txt by a comma
        ii. Parse line[0] as an integer and assign it to serviceID
        iii. Assign line[1] to serviceName
        iv. Put the serviceID and serviceName into serviceMap, with serviceID as the key
3. catch NumberFormatException and print error
4. Return the serviceMap

printServiceOptions(HashMap<Integer, String> serviceMap):

1. loop through the hashmap
    a. Print out the service code - service name