Ryan Sleeper

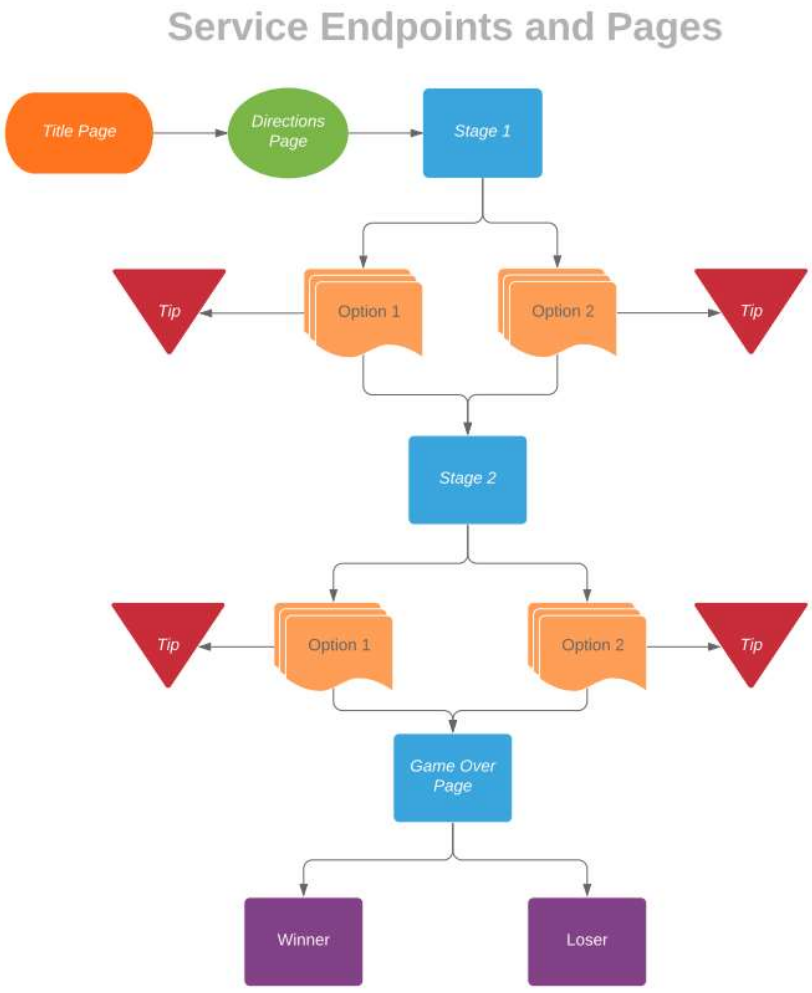
SWDV 691

Week 2

Design: Service Layer

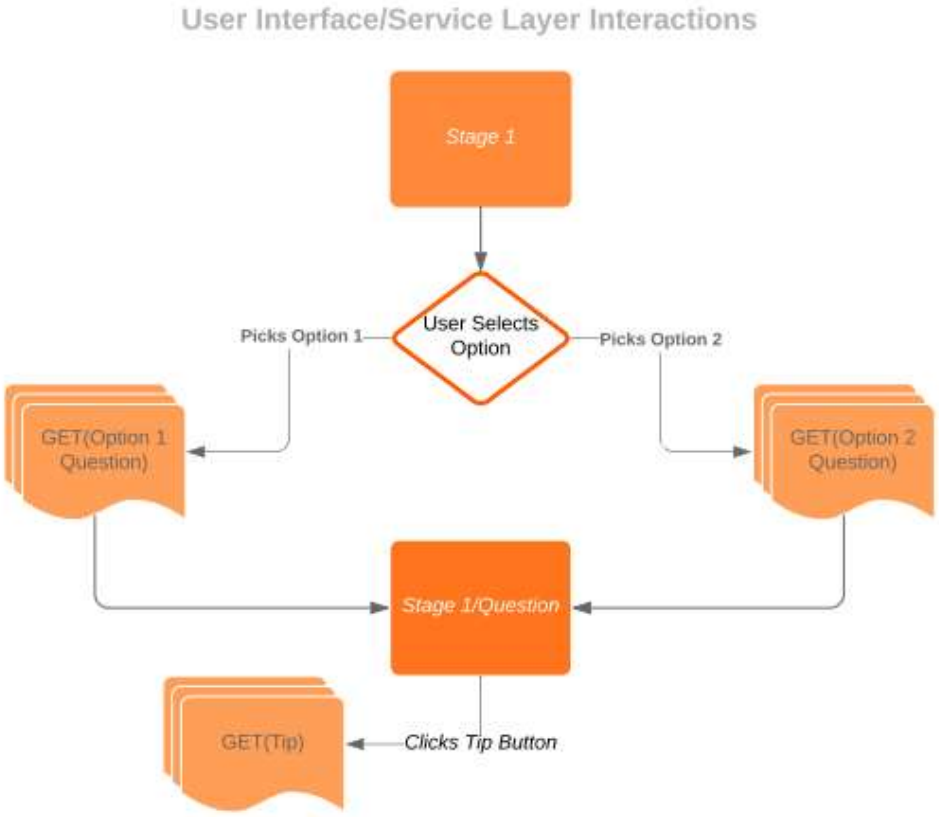
The Untold Journey: Service Layer Design

For my application, I will create a service layer that will execute all the requests needed for The Untold Journey to run properly. Due to the way the database and user interface need to interact with one another, the service layer will be using GET requests to retrieve questions for the user to attempt to answer. For example, the user will select one of the options on the page and then the service layer will pull a question from the database, along with the corresponding tip for the question. Please observe the following high-level diagram that shows all the pages and service endpoints required for the MVP.



This diagram shows a high-level overview of the pages and service endpoints that the application contains. Please take note that there are ten stages for this application. Stages 3-10 were omitted to reduce redundancy.

Let's break down this diagram a little further. The service layer will begin to work at the beginning of Stage 1. The user will select an option and based on the selection, the service layer will perform a GET request in order to grab a question from the database and display it on the user interface. Once the user has the question, he/she will have the option to view a tip. If the user clicks the tip button, the service layer will then perform another GET request and grab the corresponding tip from the database. The following diagram shows a more in-depth look at how the service layer interacts with the user interface pages and database. This interaction will occur at all ten stages. Again, to reduce redundancy, stages 2-10 are not shown.



The users will not be able to alter the database in any way, so the service layer will not be executing any POST, PUT, PATCH, or DELETE requests. There should not be any error messages for bad data since the user is not able to alter the database. If any errors occur, it would be from the user inputting a bad answer to the question. The application will reject any type of bad input and ask the user to input a proper answer. For example, if a user has the following math question: 2 + 2 = and the

user puts anything besides a float or an int, then the application will reject the answer and ask the user to put a proper input.

To sum everything up the application will use the service layer to perform GET requests for the user when the user clicks on one of the appropriate buttons. The service layer will take the request from the user and provide the user with one of the questions or tips (depending on what button they clicked) that has been stored in the database. This will be the main interaction between the database and user interface that the service layer must handle.