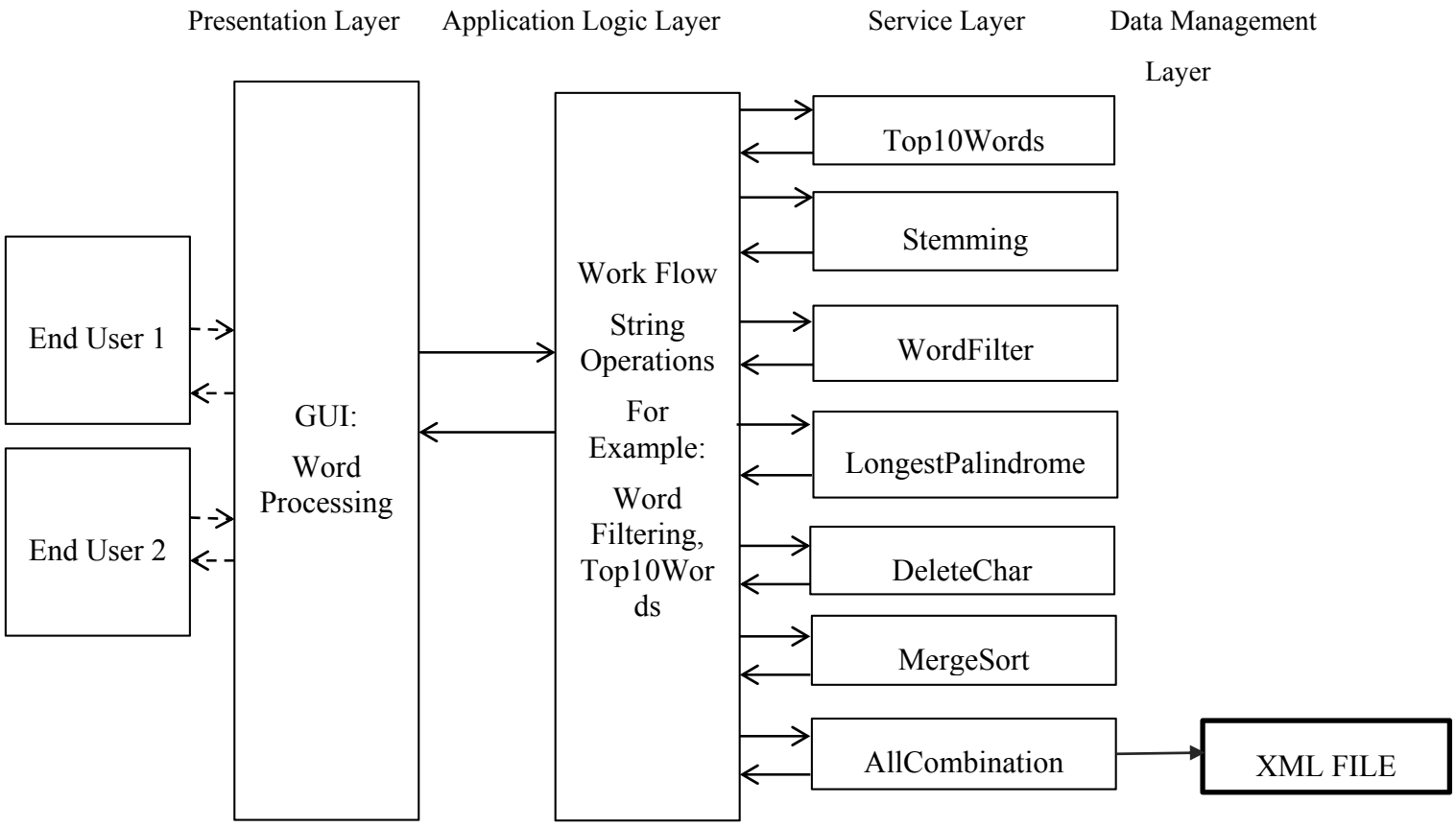
PART 1: REQUIREMENT DOCUMENT (GROUP SUBMISSION)

1.1 Description of the service-oriented computing system that your team plans to develop.

Service-Oriented Computing (SOC) is a computing paradigm that employs services as fundamental constituents to support rapid, low-cost development of distributed applications in heterogeneous environments. The main principle behind developing **"Wordplay"** application is to play with different permutation and combinations of a word. For accessing the available services, the user has to create an account with the web application. This system will be made keeping in account the security aspect as well. For E.g., Creating each user session, cookies, authentication and authorization. The promise of a Service-Oriented Computing system is cooperating with different services within itself that are being loosely coupled to flexibly create dynamic web application which can adapt quickly to the user changing requirements.

1.2 A diagram showing the overall system design, its layers, components, and the connections among the services.

Diagram:



We have XML file in the AllCombination web service, since it returns a list of output for the user input. We will be storing that data pertaining to each user.

1.3 Create a service directory (table) listing the services that your team and each team member plan to develop.

This page is deployed at: Fill out the address later submission.				
This project is developed by: StackOverFlow				
Provider name	Service name, with input and output types	TryIt link	Service description	Planned resources need to implement the service
Abhinav Singh	Top10Words: Input: A webpage url in string Output: An array of strings that contains the ten most frequently occurred words in descending order of their frequencies		Analyze the webpage at a given url and return the ten most frequently occurred words in the webpage. Return the words in the descending order of their appearing frequencies	No resources will be required since all components will be coded.
Vinit Badrike	Stemming Inputs: A string of words Output: The string of the inflected or derived words replaced by their stem words		Analyze a string of words and replace the inflected or derived words to their stem or root word. This service can help find useful keywords or index words in information processing and retrieval	No resources will be required since all components will be coded.
Abhinav Singh	WordFilter Input: A string Output: A string with the stop words removed		Analyze a string of words and filter out the function words.	No resources will be required since all components will be coded.
Vinit Badrike	Weather Service Input: zipcode Output: An array of strings, storing 5 day weather forecast for the given zipcode location.		Create a 5 day weather forecast service of zipcode location based on the national weather service	Retrieve weather information from national weather service: http://graphical.weather.gov/xml/SOAP_server/ndfdXMLserver.php?wsdl
Vinit Badrike	longestpalindrome		Given a string of words it returns the length of	No resources will be required since all components will be coded.

	Inputs: string of words Output: length of longest palindrome		longest palindrome in the given sequence of strings.	
Abhinav Singh	Deletechar Input:string1 String2 Output:String1 with deleted chars		Given two strings string1 and string2 it returns string1 with chars from string2 deleted from it	No resources will be required since all components will be coded.
Vinit Badrike	Mergesort Input: chars Output: chars in sorted order		Sort the characters using merge sort algorithm	No resources will be required since all components will be coded.
Abhinav Singh	Allcombination: Input:array of strings Output: array of strings		Gives combination of all characters in the string.	No resources will be required since all components will be coded.