The solutions for project 2 are written in JAVA using Eclipse IDE.

There is main class named **BinarySearchTree** that handles entire operations.

The program has class Node which has attributes left, right and string value to store left child, right child and string. BST is formed using objects of Node.

There is switch case that takes the input from the user.

For choice 1: Create a Balanced Binary Search Tree using the input Strings entered from console

1. User needs to enter total number of BST elements that needs to be added to form BST.
2. User enters string elements from console.
3. Each individual element will be inserted in Binary search tree using insert() function.
4. The designed BST is send to buildTree() function which balances the formed BST. buldTree() function use ArrayList to store nodes of given BST in sorted order by using storeBalance() function. The arraylist is passed to buildTreeUntil() function which balance the given BST.

For choice 2: Find the length of the Balanced Binary Search Tree.

1. The program calls calculateHeight() function which calculates the height of balanced binary tree.It is recursive function that call left child and right child to calculate height.

For choice 3: Add an element to BST

1. User needs to enter string element that needs to be inserted in BST.
2. It will call insert() method to add new element.
3. After addition of new element it will call buildTree() function to balance the binary tree.

For choice 4: Delete an element from BST

1. User needs to enter string element that needs to be deleted from BST.
2. It will call deleteElement() method.
3. After deletion it will call buildTree() function to balance the binary tree.

For choice 5: Print Elements of the BST

1. The program calls printTree() function which prints inorder, preorder and postorder traversal of BST

For choice 6: Check if BST is Max Binary Heap or not

1. The program calls isHeap() function which compare all left and right child with their parent node. The function returns false if left child is greater than parent and right child is less than its parent.
2. So basically isHeap() function checks max heap property and return Boolean value

For choice 7: Find the number of Anagrams for each input string in the BST

1. The program calls isAnagram() function. The program use arraylist to store inorder traversal of BST. Traversal will provide list of all the elements in BST.
2. The program will send each pair of BST elements to checkAnagram() function. The function will sort the string and store it in char array. After sorting if the string value of both sorted array are equal then they are anagram of each other
3. I have created a HashMap that stores String and Integer as key value. If two elements are anagram of each other the we will update the integer value in hashmap.
4. The hashmap is printed in console with Map.Entry to show anagram pairs

For choice 8: Exit

1. The program used while loop to show user choices until exit is selected.