ASSIGNMENT NO. B4

1 Problem Statement

A Web application for Concurrent implementation of ODD-EVEN SORT is to be designed using Real time Object Oriented Modeling(ROOM). Give the necessary design diagrams and write the test cases for the white box testing. Draw Concurrent collaboration Diagrams.

2 Objectives

1. To study how to create a web application. 2. To study ODD-EVEN sort algorithm.

3 Theory

3.1 Web Applications

In computing, a web application or web app is a client-server software application in which the client (or user interface) runs in a web browser. Web applications are popular due to the ubiquity of web browsers, and the convenience of using a web browser as a client to update and maintain web applications without distributing and installing software on potentially thousands of client computers is a key reason for their popularity, as is the inherent support for cross-platform compatibility.

Web sites most likely to be referred to as "web applications" are those which have similar functionality to a desktop software application, or to a mobile app. HTML5 introduced explicit language support for making applications that are loaded as web pages, but can store data locally and continue to function while offline.

3.2 Writing Web Applications

Writing a web application is often simplified by open source software such as Django, Drupal, Ruby on Rails or Symfony called web application frameworks. These frameworks facilitate rapid application development by allowing a development team to focus on the parts of their application which are unique to their goals without having to resolve common development issues such as user management. While many of these frameworks are open source, this is by no means a requirement.

The use of web application frameworks can often reduce the number of errors in a program, both by making the code simpler, and by allowing one team to concentrate on the framework while another focuses on a specified use case. In applications which are exposed to constant hacking attempts on the Internet, security-related problems can be caused by errors in the program. Frameworks can also promote the use of best practices such as GET after POST.

In addition, there is potential for the development of applications on Internet operating systems, although currently there are not many viable platforms that fit this model.

3.3 Applications

Examples of browser applications are simple office software (word processors, online spreadsheets, and presentation tools), but can also include more advanced applications such as project management, computer-aided design, video editing and point-of-sale.

3.4 ODD-EVEN Sort:

In computing, an odd-even sort or odd-even transposition sort (known as brick sort) is a relatively simple sorting algorithm, developed originally for use on parallel processors with local interconnections. It is a comparison sort related to bubble sort, with which it shares many characteristics. It functions by comparing all (odd, even)-indexed pairs of adjacent elements in the list and, if a pair is in the wrong order (the first is larger than the second) the elements are switched. The next step repeats this for (even, odd)-indexed pairs (of adjacent elements). Then it alternates between (odd, even) and (even, odd) steps until the list is sorted. On parallel processors, with one value per processor and only local left-right neighbor connections, the processors all concurrently do a compare-exchange operation with their neighbors, alternating between odd-even and even-odd pairings. The algorithm extends efficiently to the case of multiple items per processor. In the Baudet-Stevenson odd-even merge-splitting algorithm, each processor sorts its own sublist at each step, using any efficient sort algorithm, and then performs a merge splitting, or transposition-merge, operation with its neighbor, with neighbor pairing alternating between odd-even and even-odd on each step. Performance and complexity:

1. Worst case performance O(n$\hat{2}$)
2. Best case performance O(n)
3. Worst case space complexity : O(1)

3.5 Real Time Object Oriented Modeling (ROOM):

Real Time Object Oriented Modeling (ROOM) is a domain specific language. ROOM was developed in the early 1990s for modeling Real-time systems.[1] The initial focus was on telecommunications, even though ROOM can be applied to any event-driven real-time system. ROOM is supported by ObjecTime Developer (commercial) and eTrice[2] When UML was defined, many elements of ROOM were taken over. Developing real time software is particularly challenging since the complexity of the physical world has to be accommodated as well as stringent resource and timeliness constraints. Such circumstances require special language support over and above what can be found in general purpose programming languages. ROOM is both an object oriented modeling language and a development method specifically designed for dealing with large

real time systems. It supports automatic code generation to ensure reliability and increase productivity.

3.5.1 ROOM Diagram Software Features:

- Chart Templates
- Drag and Drop Interface
- Snap-to
- Automatic Spacing and Alignment
- Change Connector Path
- Add Connector Points to Symbols
- Multiple Pages
- Symbol Gallery
- Grids and Guides
- Change Existing Diagram Shapes
- Multiple Connector Points
- Connector Labels
- Junction Jogs
- Add Hyperlinks
- Expandable Canvas

4 Mathematical Model

```
S = {s, e, I, o, f, DD, NDD, success, failure}
S = {Initial state of system}
I = {input of system}
O = {Output of system}
DD = {Deterministic Data }
NDD = {Non-Deterministic Data }
Success = {Desired outcome generated }
Failure = {Desired outcome not generated }
I = {I1, I2}
I1 = {0-9}
I2 =NDD = {A-Z, a-z}
F = { divide(), sort_even(), sort_odd(), merge()}
divide() = { Divide list according to index}
sort-even() ={sort the list with even index}
sort-odd() = {sort the list with odd index}
```

```
merge() = {merge both the list}
success = {desired output is generated i.e. the list is sorted successfully}.
failure = {desired output is not generated}.
```

5 Use-case diagram

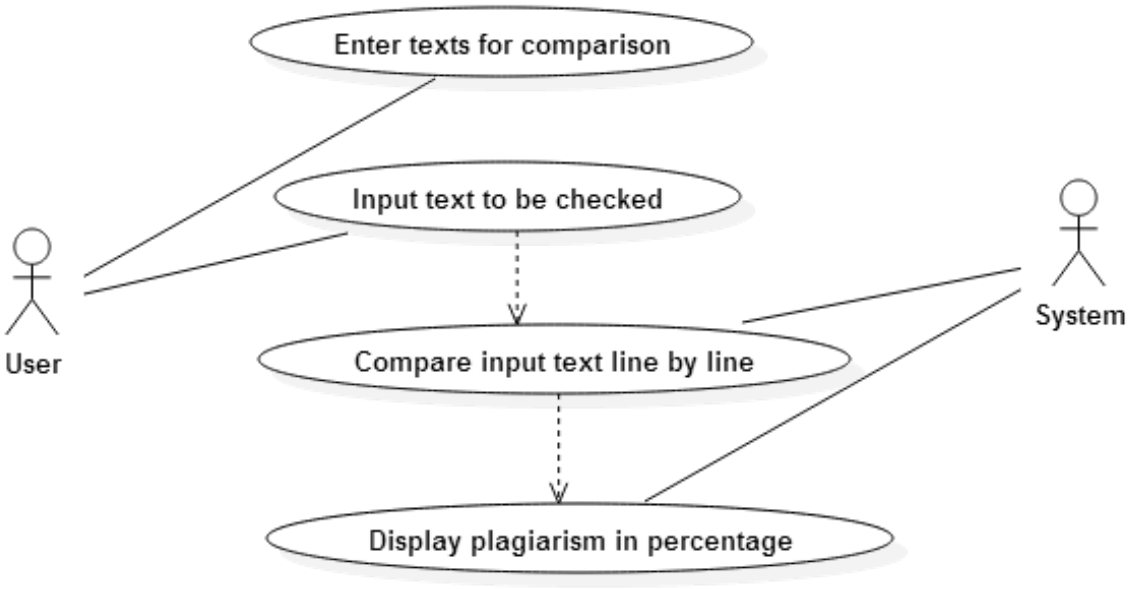

Figure 1: Use case Diagram

6 Algorithm

1. Start
2. Enter the unsorted array in one text area.
3. Unsorted array sorted using Odd Even sort algorithm.
4. Display the sorted array in another text area.
5. Stop

7 Black Box Testing

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance.

8 White Box testing

White-box testing is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. The other names of glass box testing are clear box testing, open box testing, logic driven testing or path driven testing or structural testing.

9 Positive Testing

Positive testing is a testing technique to show that a product or application under test does what it is supposed to do. Positive testing verifies how the application behaves for the positive set of data.

10 Negative Testing

Negative testing ensures that your application can specifically handle invalid input or unexpected user behavior.

11 Conclusion

We have successfully created web application to sort thr array using ODD-EVEN Sort algorithm.