

# **Ranking Spatial Data by Quality Preferences**

Project Report submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology.

in

**Computer Science & Engineering**

under the Supervision of

*Mr. Suman Saha*

By

*Siddharth Sharma 111253*

to



Jaypee University of Information Technology

Waknaghat, Solan – 173234, Himachal Pradesh

## **Certificate**

This is to certify that project report entitled “**Ranking Spatial Data by Quality Preferences**” , submitted by **Siddharth Sharma** in partial fulfillment for the award of degree of Bachelor of Technology in Computer Science & Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

**Date: 16.05.2015**

**Supervisor’s Name: Mr. Suman Saha**

**Designation: Assistant Professor (Grade-II)**

## **Acknowledgement**

I, Siddharth Sharma am using this opportunity to express my gratitude to everyone who supported me throughout the course of this project. I am thankful for their aspiring guidance, invaluable constructive criticism and friendly advice during the project work. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

I would also like to thank my project guide Mr. Suman Saha who provided me with the knowledge being required and conducive conditions for my project.

Date: 16.05.2015

Siddharth Sharma

**Ranking Spatial Data**  
**By**  
**Quality Preferences**  
**(Data Mining)**

## Table of Content

<b>S.No.</b>	<b>Topic</b>	<b>Page No.</b>
1	Certificate	2
2	Acknowledgment	3
3	Table Of Content	5
4	List Of Figures	6
5	Abstract	7
6	Introduction	8
7	Problem Statement	10
8	Literature Survey	12
9	1.Module Description 1.1 Spatial Ranking 1.2 Non-Spatial ranking 1.3 Neighbor (NN) Retrieval 1.4 Spatial Query Evaluation on R-trees 1.5 System Configuration 1.5.1 H/W System Configuration 1.5.2 S/W System Configuration 1.6 Software Environment	13     14  15  16  17
10	2. System Study 2.1 Feasibility Study 2.1.1 Economical Feasibility 2.1.2 Technical Feasibility 2.1.3 Social Feasibility	21 22
11	Input Design	23
12	Objectives	24
13	Output Design	25
14	System Design (Data Flow Diagram/ Use Case Diagram / Flow Diagram)	26
15	Screen Shots	34
16	3. System Testing 3.1 Unit Testing 3.2 Integration Testing 3.3 Acceptance Testing	40 42 43
17	Conclusion	44
18	Future Implementation	45
19	Bibliography 1. Abbreviations 2. References Made From Sites Referred	46 47

## List of Figures

Figure	Page No.
Figure 1 : R-Tree that indexes the spatial objects	14
Figure 2 :DFD For User Register	26
Figure 3 :DFD For Admin Login	27
Figure 4 :Use Case For Adding Book Detail	28
Figure 5 :Activity Diagram For Users	29
Figure 6 :Activity Diagram For Admin	30
Figure 7 :Class Diagram For Admin And Users	31
Figure 8 :Sequence Diagram For Admin And Users	32
Figure 9 :Component Diagram For Admin	33
Figure 10 : Homepage -- Screen Shot	34
Figure 11: New User Registration -- Screen Shot	35
Figure 12: Admin Login – Screen Shot	35
Figure 13: Adding Book Details -- Screen Shot	35
Figure 14: User Profile View And Delete – Screen Shot	36
Figure 15: User Login – Screen Shot	36
Figure 16: View Books And Vote -- Screen Shot	37
Figure 17: Search by Book using Index and Search Engine Technique – Screen Shot	37
Figure 18: View Rank Here – Screen Shot	38
Figure 19: ViewVotes– Screen Shot	38
Figure 20: Blog Discussion – Screen Shot	39
Figure 21:Admin View Members Lists– Screen Shot	39

## **ABSTRACT**

A spatial preference query ranks objects based on the qualities of features in their spatial neighborhood. For example, using a real estate agency database of flats for lease, a customer may want to rank the flats with respect to the appropriateness of their location, defined after aggregating the qualities of other features (e.g., restaurants, cafes, hospital, market, etc.) within their spatial neighborhood. Such a neighborhood concept can be specified by the user via different functions. It can be an explicit circular region within a given distance from the flat. Another intuitive definition is to assign higher weights to the features based on their proximity to the flat. In this paper, we formally define spatial preference queries and propose appropriate indexing techniques and search algorithms for them. Extensive evaluation of our methods on both real and synthetic data reveals that an optimized branch-and-bound solution is efficient and robust with respect to different parameters.

## INTRODUCTION

A spatial preference query ranks objects based on the qualities of features in their spatial neighborhood. For example, using a real estate agency database of flats for lease, a customer may want to rank the flats with respect to the appropriateness of their location, defined after aggregating the qualities of other features (e.g., restaurants, cafes, hospital, market, etc.) within their spatial neighborhood. Such a neighborhood concept can be specified by the user via different functions.

It can be an explicit circular region within a given distance from the flat. Another intuitive definition is to consider the whole spatial domain and assign higher weights to the features based on their proximity to the flat.

In this paper, we formally define spatial preference queries and propose appropriate indexing techniques and search algorithms for them. Extensively evaluation of our methods on both real and synthetic data reveal that an optimized branch-and-bound solution is efficient and robust with respect to different parameters.

Spatial database systems manage large collections of geographic entities, which apart from spatial attributes contain non-spatial information (e.g., name, size, type, price, etc.). In this paper, we study an interesting type of preference queries, which select the best spatial location with respect to the quality of facilities in its spatial neighborhood. Given a set  $D$  of interesting objects (e.g., candidate locations), a top- $k$  spatial preference query retrieves the  $k$  objects in  $D$  with the highest scores.



The score of an object is defined by the quality of features (e.g., facilities or services) in its spatial neighborhood. As a motivating example, consider a real estate agency office that holds a database with available flats for lease. Here “feature” refers to a class of objects in a spatial map such as specific facilities or services.

A customer may want to rank the contents of this database with respect to the quality of their locations, quantized by aggregating non-spatial characteristics of other features (e.g., restaurants, cafes, hospital, market, etc.) in the spatial neighborhood of the flat (defined by a spatial range around it).

Quality may be subjective and query-parametric.

For example, a user may define quality with respect to non-spatial attributes of restaurants around it (e.g., whether they serve seafood, price range, etc.).

## Problem Statement

### Existing System:

To our knowledge, there is no existing efficient solution for processing the top-k spatial preference query. Object ranking is a popular retrieval task in various applications. In relational databases, we rank tuples using an aggregate score function on their attribute values. For example, a real estate agency maintains a database that contains information of flats available for rent. A potential customer wishes to view the top-10 flats with the largest sizes and lowest prices. In this case, the score of each flat is expressed by the sum of two qualities: size and price, after normalization to the domain (e.g., 1 means the largest size and the lowest price). In spatial databases, ranking is often associated to nearest neighbor (NN) retrieval. Given a query location, we are interested in retrieving the set of nearest objects to it that qualify a condition (e.g., restaurants). Assuming that the set of interesting objects is indexed by an R-tree, we can apply distance bounds and traverse the index in a branch-and-bound fashion to obtain the answer.

### Proposed System:

We Propose

- (1) Spatial ranking, which orders the objects according to their distance from a reference point, and
- (2) Non-spatial ranking, which orders the objects by an aggregate function on their non-spatial values.

Our top- k spatial preference query integrates these two types of ranking in an intuitive way. As indicated by our examples, this new query has a wide range of applications in service recommendation and decision support systems. To our knowledge, there is no existing efficient solution for processing the top-k spatial preference query. A brute-force approach (to be elaborated in Section 3.2) for evaluating it is to compute the scores of all objects in  $D$  and select the top-k ones. This method, however, is expected to be very expensive for large input datasets.

Spatial database systems manage large collections of geographic entities, which apart from spatial attributes contain non-spatial information (e.g., name, size, type, price, etc.). In this paper, we study an interesting type of preference queries, which select the best spatial location with respect to the quality of facilities in its spatial neighborhood. Given a set  $D$  of interesting objects (e.g., candidate locations), a top- $k$  spatial preference query retrieves the  $k$  objects in  $D$  with the highest scores. The score of an object is defined by the quality of features (e.g., facilities or services) in its spatial neighborhood. As a motivating example, consider a real estate agency office that holds a database with available flats for lease. Here “feature” refers to a class of objects in a spatial map such as specific facilities or services. A customer may want to rank the contents of this database with respect to the quality of their locations, quantized by aggregating nonspatial characteristics of other features (e.g., restaurants, cafes, hospital, market, etc.) in the spatial neighborhood of the flat (defined by a spatial range around it). Quality may be subjective and query-parametric. For example, a user may define quality with respect to non-spatial attributes of restaurants around it (e.g., whether they serve seafood, price range, etc.).

## **Literature survey**

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy n company strength. Once these things are satisfied, ten next steps are to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

# 1. Module Description

1.1 Spatial Ranking

1.2 Non-Spatial ranking

1.3 Neighbor (NN) Retrieval

1.4 Spatial Query Evaluation on R-trees

1.5 System Configuration

## 1.1 Spatial Ranking

Spatial ranking, which orders the objects according to their distance from a reference point.

## 1.2 Non-Spatial Ranking:

Non-spatial ranking, which orders the objects by an aggregate function on their non-spatial values. Our top- k spatial preference query integrates these two types of ranking in an intuitive way. As indicated by our examples, this new query has a wide range of applications in service recommendation and decision support systems. To our knowledge, there is no existing efficient solution for processing the top-k spatial preference query.

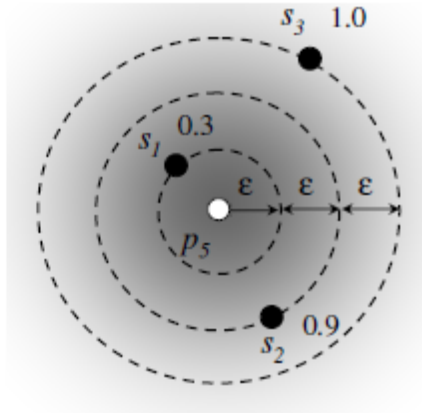
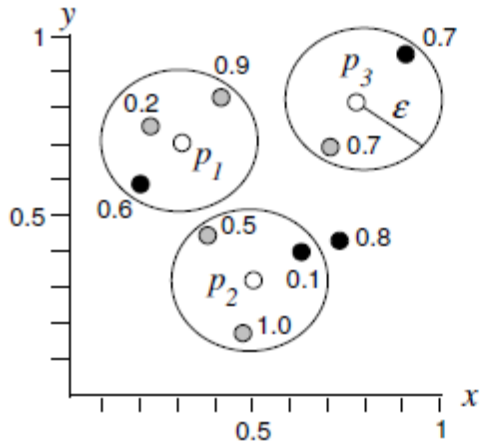
## 1.3 Neighbor (NN) Retrieval:

Object ranking is a popular retrieval task in various applications. In relational databases, we rank tuples using an aggregate score function on their attribute values. For example, a real estate agency maintains a database that contains information of flats available for rent. A potential customer wishes to view the top-10 flats with the largest sizes and lowest prices. In this case, the score of each flat is expressed by the sum of two qualities: size and price, after normalization to the domain  $[0,1]$  (e.g., 1 means the largest size and the lowest price). In spatial databases, ranking is often associated to nearest neighbor (NN) retrieval. Given a query location, we are interested in retrieving the set of nearest objects to it that qualify a condition (e.g., restaurants). Assuming that the set of interesting objects is indexed by an R-tree [3], we can apply distance bounds and traverse the index in a branch-and-bound fashion to obtain the answer.

Nevertheless, it is not always possible to use multidimensional indexes for top-k retrieval. First, such indexes break-down in high dimensional spaces. Second, top-k queries may involve an arbitrary set of user-specified attributes (e.g., size and price) from possible ones (e.g., size, price, distance to the beach, number of bedrooms, floor, etc.) and indexes may not be available for all possible attribute combinations (i.e., they are too expensive to create and maintain). Third, information for different rankings to be combined (i.e., for different attributes) could appear in different databases (in a distributed database scenario) and unified indexes may not exist for them. Solutions for top-k queries focus on the efficient merging of object rankings that may arrive from different (distributed) sources. Their motivation is to minimize the number of accesses to the input rankings until the objects with the top-k aggregate scores have been identified. To achieve this, upper and lower bounds for the objects seen so far are maintained while scanning the sorted lists.

#### **1.4 Spatial Query Evaluation:**

The most popular spatial access method is the R-tree, which indexes minimum bounding rectangles (MBRs) of objects. Figure 2 shows a set  $D = \{p_1, \dots, p_n\}$  of spatial objects (e.g., points) and an R-tree that indexes them. R-trees can efficiently process main spatial query types, including spatial range queries, nearest neighbor queries, and spatial joins. Given a spatial region  $W$ , a spatial range query retrieves from  $D$  the objects that intersect  $W$ . For instance, consider a range query that asks for all objects within the shaded area in Figure 2. Starting from the root of the tree, the query is processed by recursively following entries, having MBRs that intersect the query region. For instance,  $e_1$  does not intersect the query region, thus the subtree pointed by  $e_1$  cannot contain any query result. In contrast,  $e_2$  is followed by the algorithm and the points in the corresponding node are examined recursively to find the query result.



## 1.5 System Configuration:-

### 1.5.1 H/W System Configuration:-

Processor	-	Pentium –III
Speed	-	1.1 Ghz
RAM	-	256 MB(min)
Hard Disk	-	20 GB
Floppy Drive	-	1.44 MB
Key Board	-	Standard Windows Keyboard
Mouse	-	Two or Three Button Mouse
Monitor	-	SVGA

### **1.5.2S/W System Configuration:-**

- ❖ Operating System : Windows95/98/2000/XP
- ❖ Application Server : Tomcat5.0/6.X
- ❖ Front End : HTML, Java, Jsp
- ❖ Scripts : JavaScript.
- ❖ Server side Script : Java Server Pages.
- ❖ Database : MsAccess
- ❖ Database Connectivity : JDBC.



## 1.6 Software Environment

### Java Technology

Java technology is both a programming language and a platform.

### The Java Programming Language

**The Java programming language is a high-level language that can be characterized by all of the following buzzwords:**

- Simple
- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance
- Interpreted
- Multithreaded
- Robust
- Dynamic
- Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called *Java byte codes* —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.

You can think of Java byte codes as the machine code instructions for the *Java Virtual Machine* (Java VM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make “write once, run anywhere” possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.

## **The Java Platform**

A *platform* is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

- The *Java Virtual Machine* (Java VM)
- The *Java Application Programming Interface* (Java API)

You've already been introduced to the Java VM. It's the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as *packages*. The next section, *What Can Java Technology Do?* Highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.

Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring performance close to that of native code without threatening portability

### What Can Java Technology Do?

The most common types of programs written in the Java programming language are *applets* and *applications*. If you've surfed the Web, you're probably already familiar with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser.

However, the Java programming language is not just for writing cute, entertaining applets for the Web. Using the generous API, you can write many types of programs.

An application is a standalone program that runs directly on the Java platform. A special kind of application known as a *server* serves and supports clients on a network. Examples of servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a *servlet*. A servlet can almost be thought of as an applet that runs on the server side. Servlets are similar to applets in that they are runtime extensions of applications. Instead of working in browsers, though, servlets run within Java Web servers, configuring or tailoring the server.

How does the API support all these kinds of programs? It does so with packages of software components that provides a wide range of functionality.

Every full implementation of the Java platform gives you the following features:

- **The essentials:** Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.
- **Applets:** The set of conventions used by applets.
- **Networking:** URLs, TCP (Transmission Control Protocol), UDP (User Datagram Protocol) sockets, and IP (Internet Protocol) addresses.

**Internationalization:** Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.

- **Security:** Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.
- **Software components:** Known as JavaBeans<sup>™</sup>, can plug into existing component architectures.
- **Object serialization:** Allows lightweight persistence and communication via Remote Method Invocation (RMI).
- **Java Database Connectivity (JDBC<sup>™</sup>):** Provides uniform access to a wide range of relational databases.

## **3. SYSTEM STUDY**

### **3.1 FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

2.1.1 ECONOMICAL FEASIBILITY

2.1.2 TECHNICAL FEASIBILITY

2.1.3 SOCIAL FEASIBILITY

## **ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

## **TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

## **SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

## **INPUT DESIGN**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

## **OBJECTIVES**

1 Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user

will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

## OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- ❖ Convey information about past activities, current status or projections of the
- ❖ Future.
- ❖ Signal important events, opportunities, problems, or warnings.
- ❖ Trigger an action.
- ❖ Confirm an action.

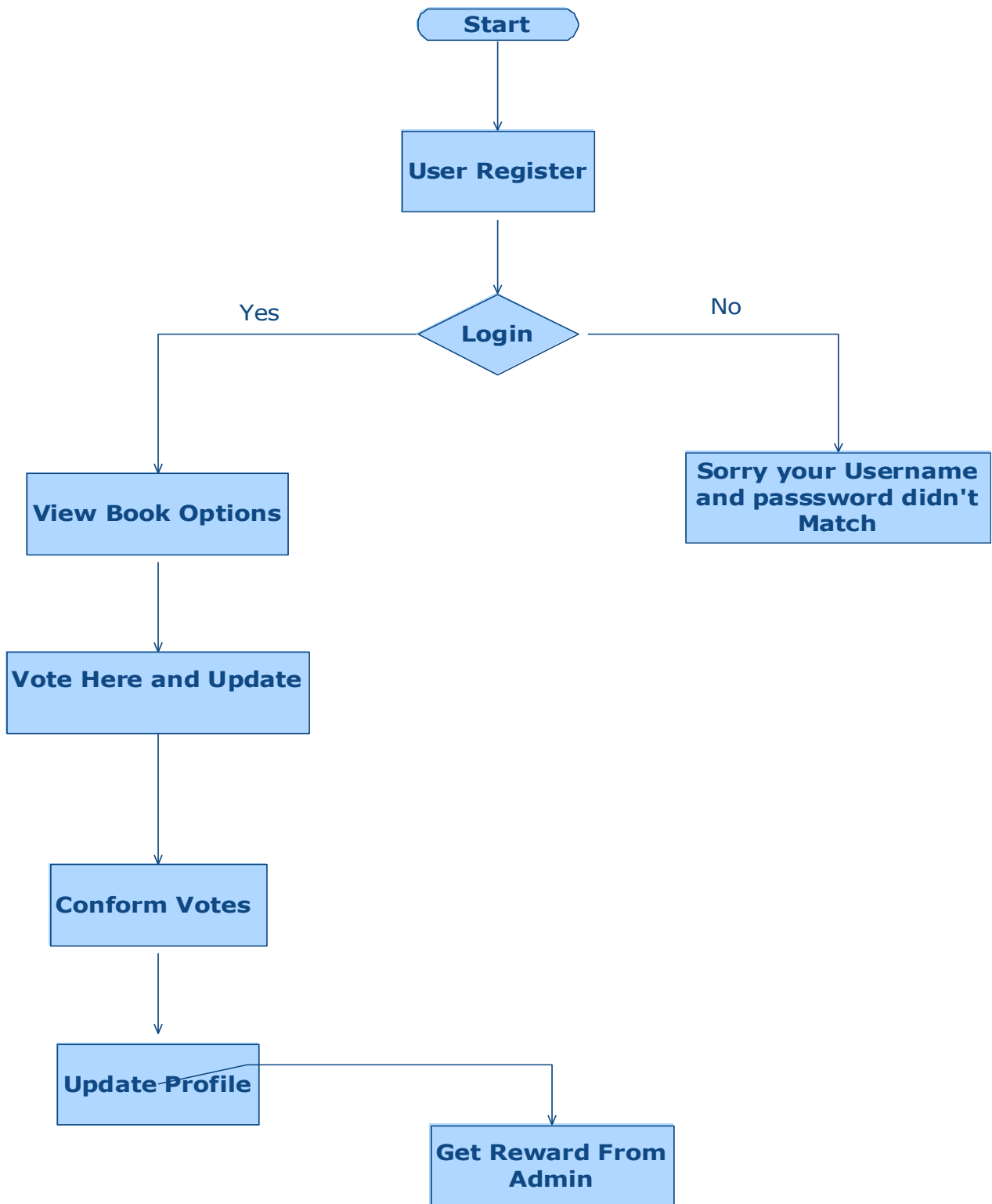


## **SYSTEM DESIGN**

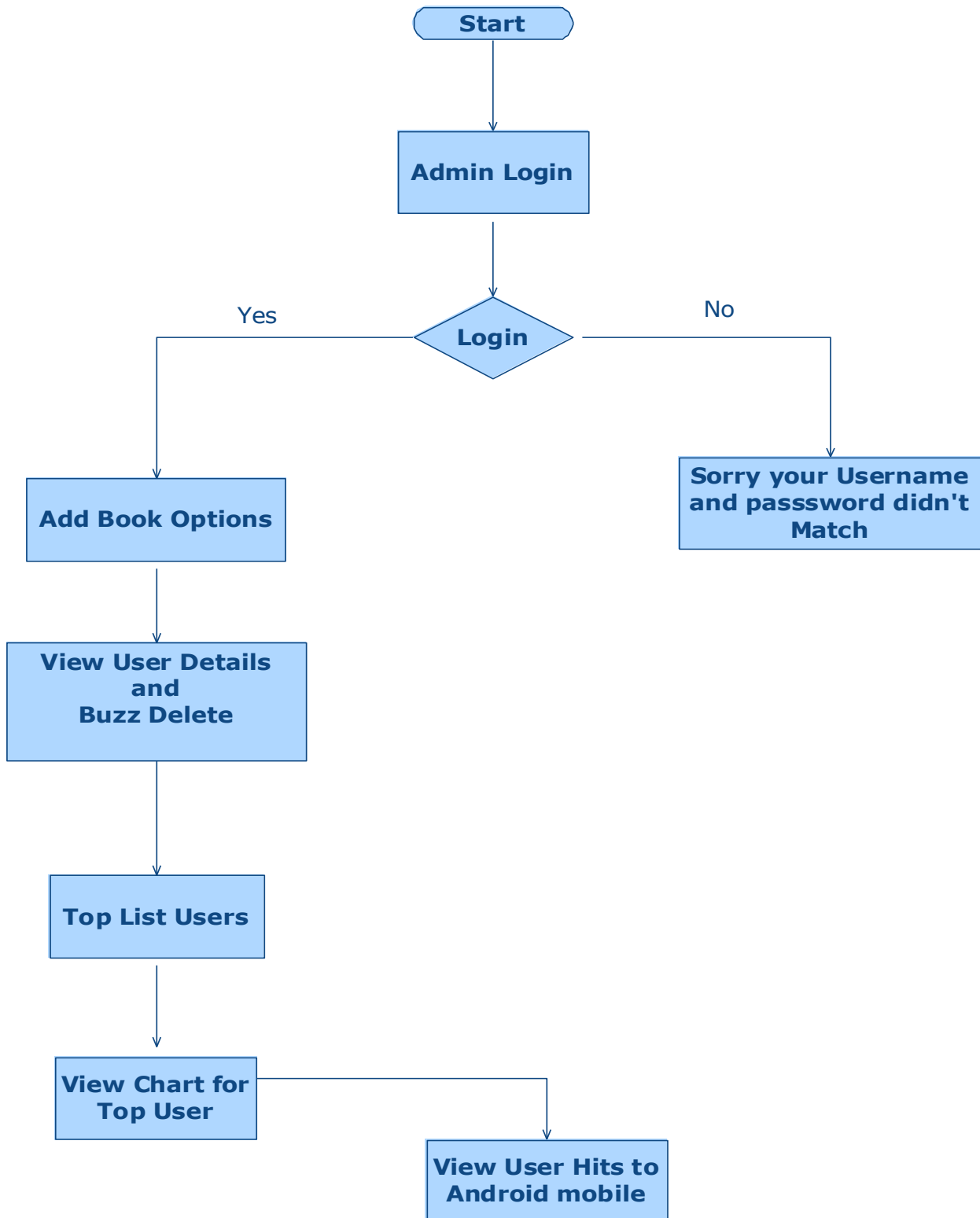
### **Data Flow Diagram / Use Case Diagram / Flow Diagram**

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system.

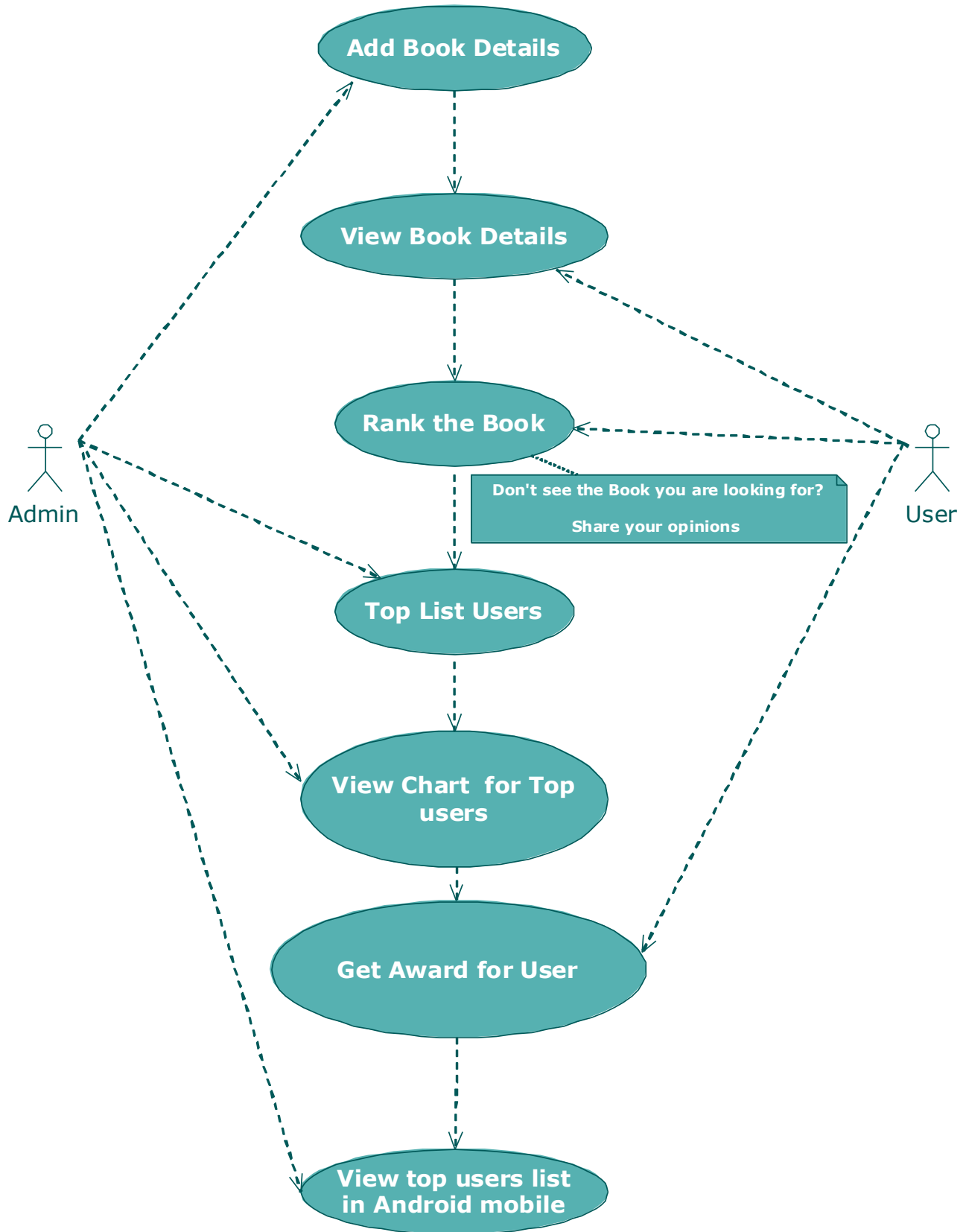
# DFD 1:



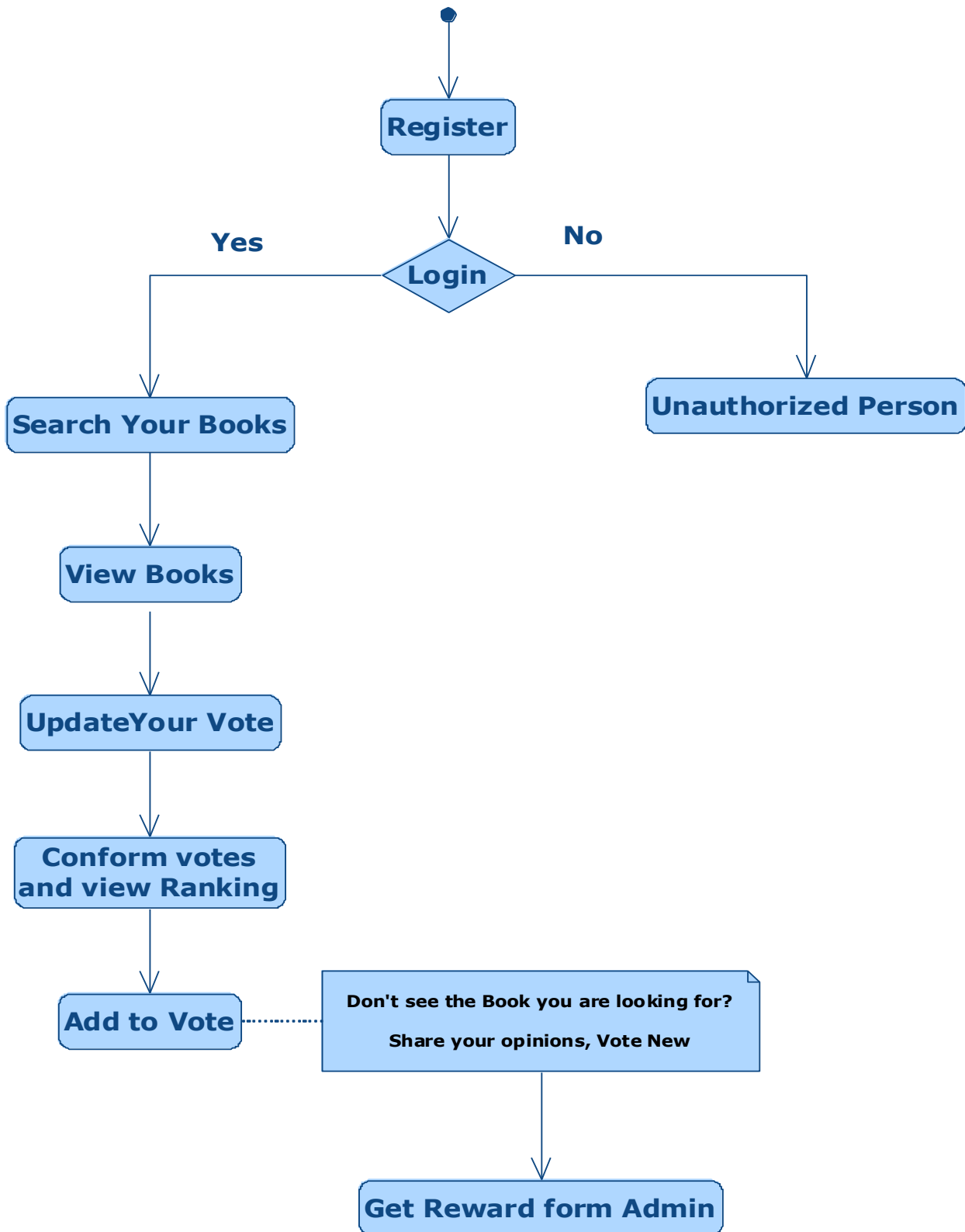
**DFD 2:**



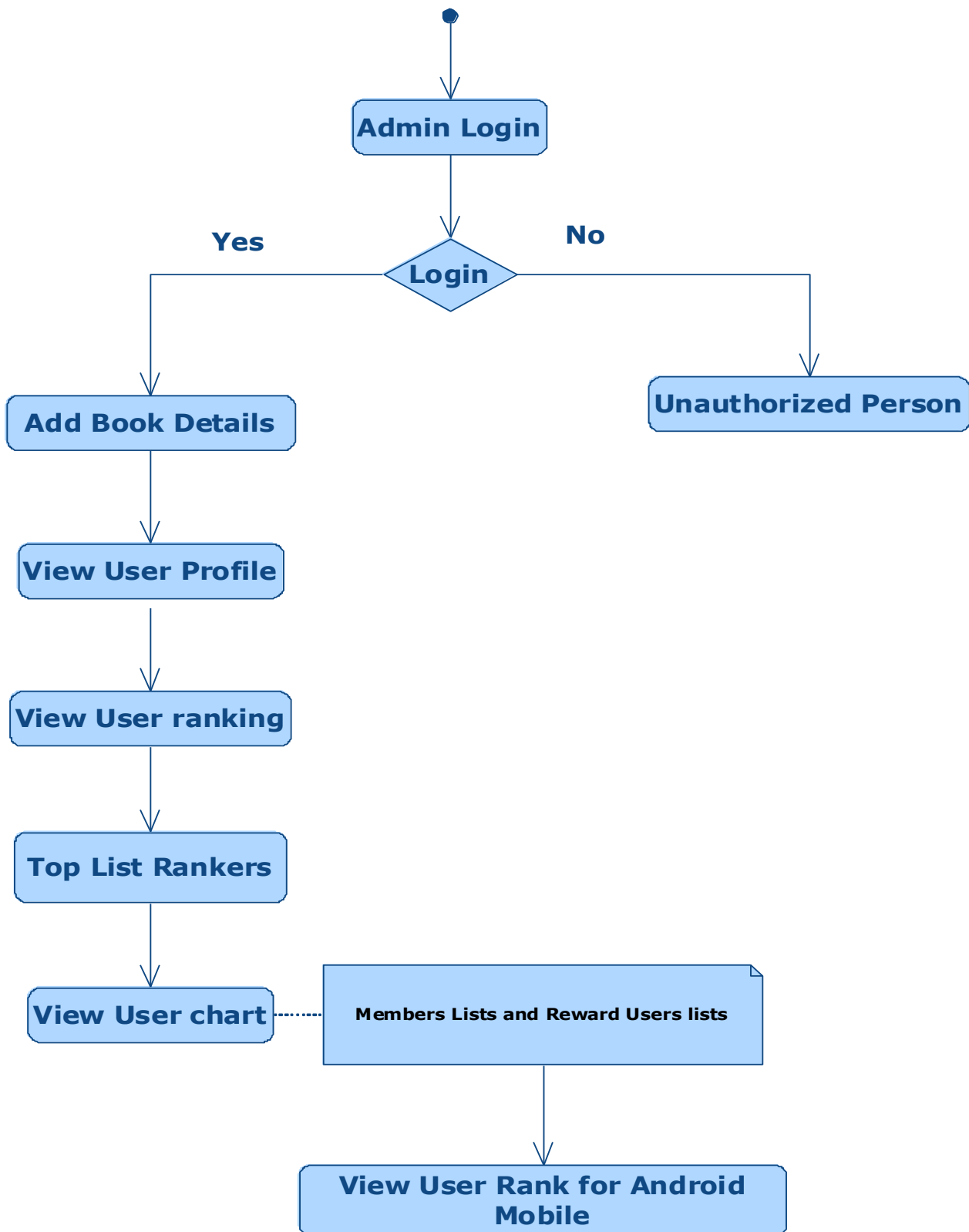
# Usecase:



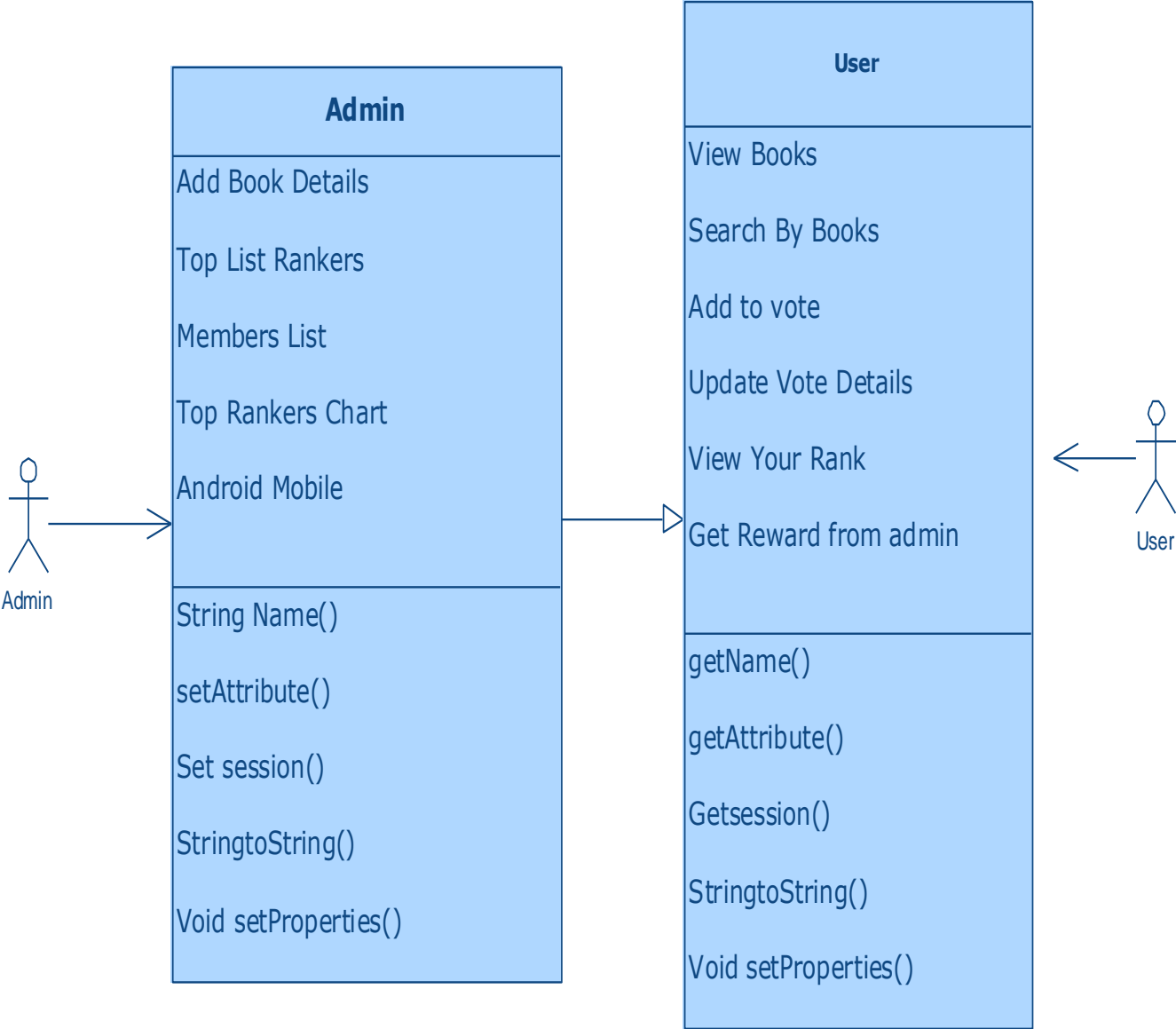
## Activity Diagram Users:



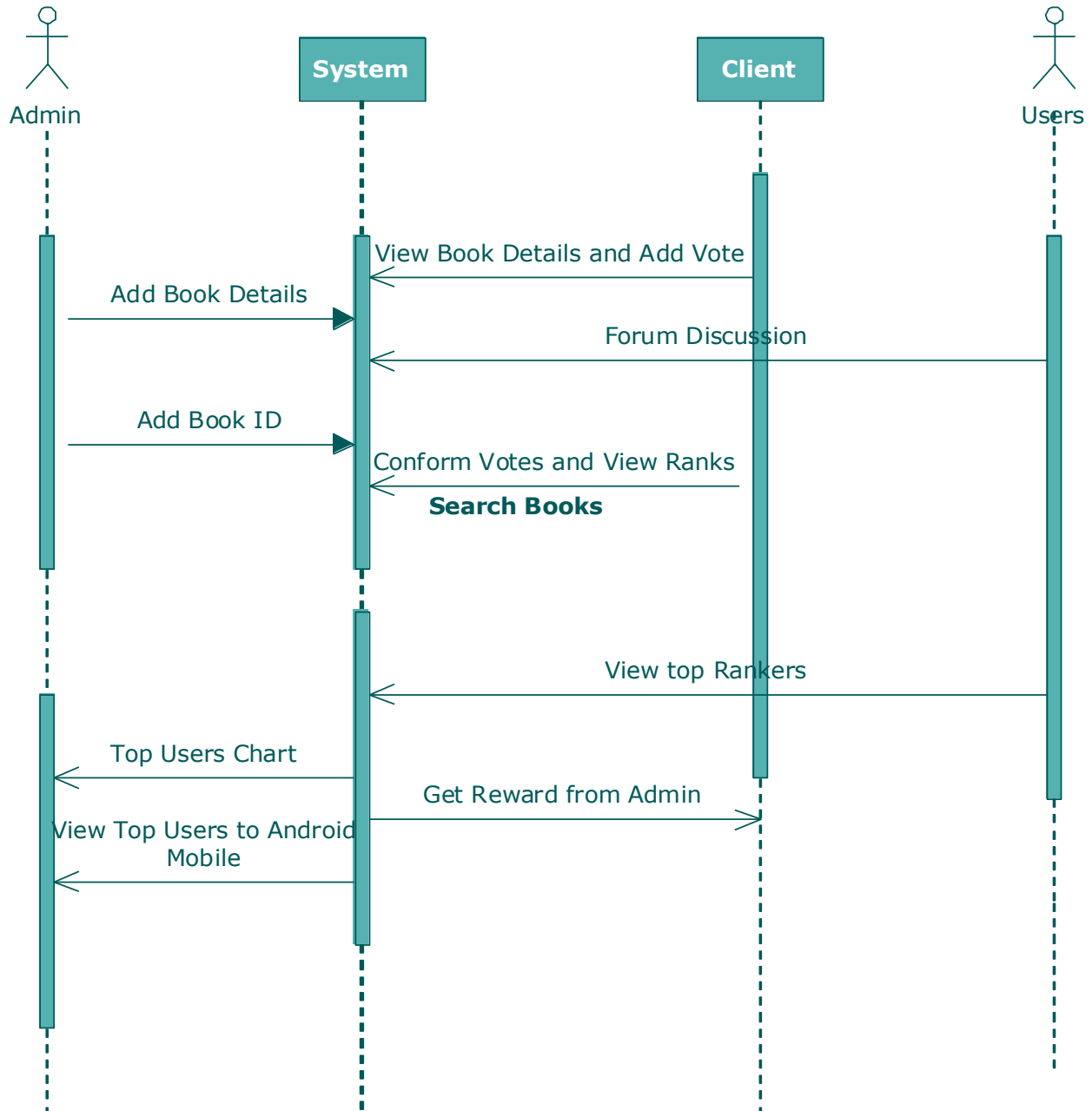
## Activity for Admin:



# Class Diagram:

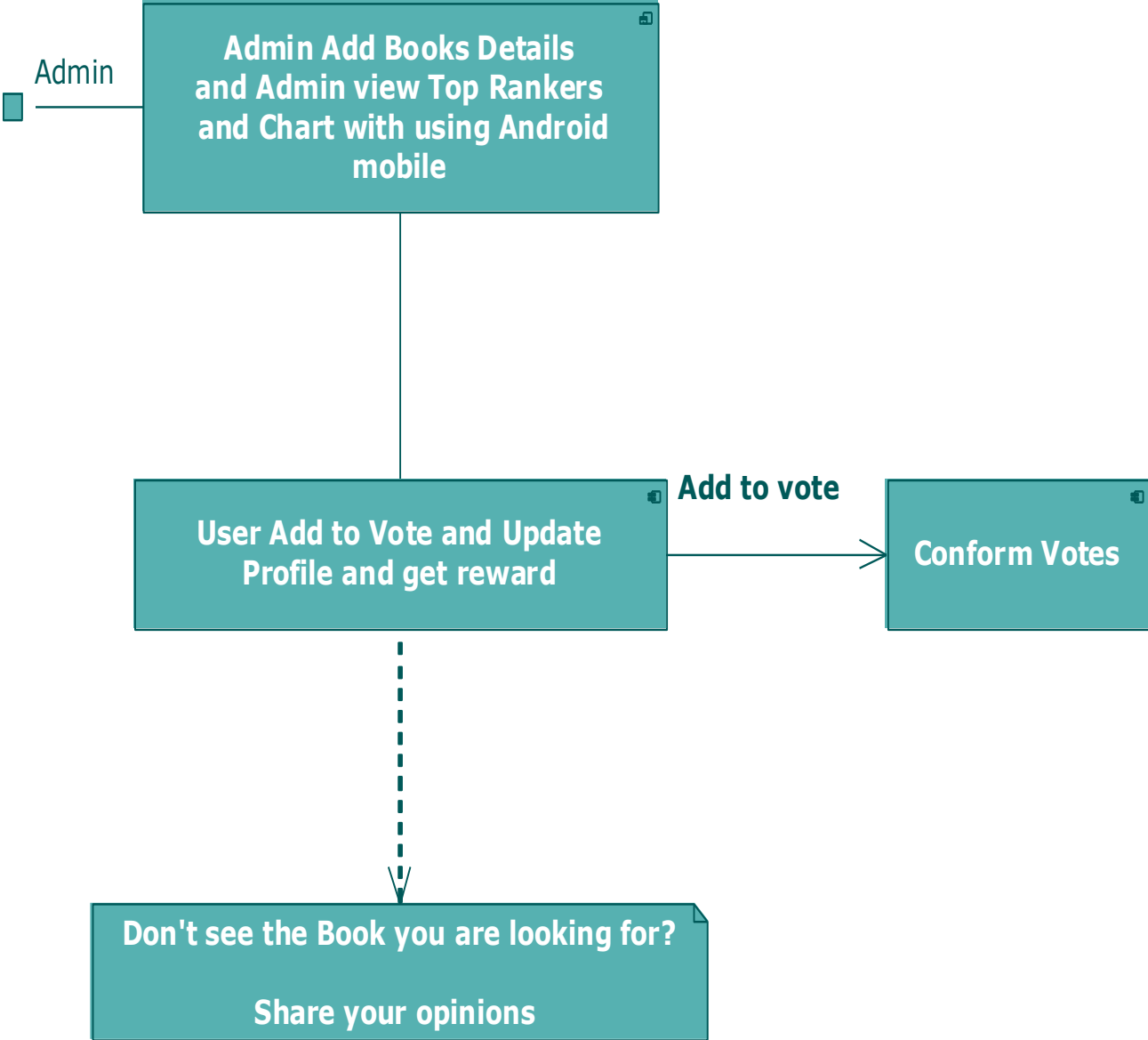


## Sequence Diagram:





**Component Diagram:**

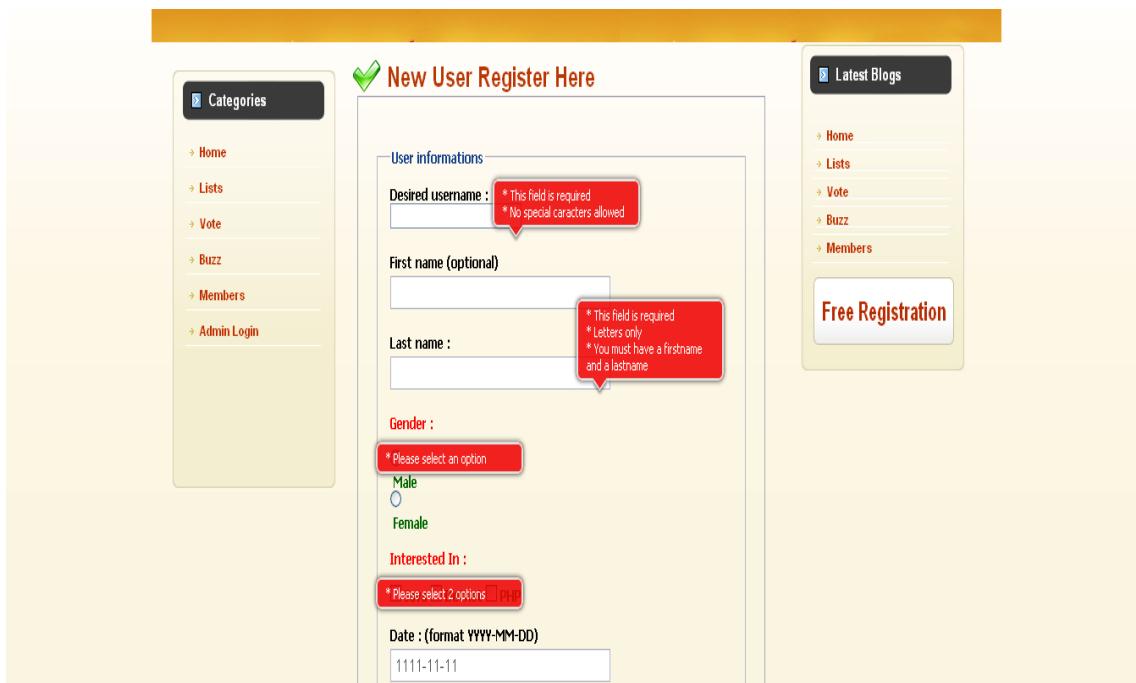


# Screen Shots

## Home:



## Register:



## Admin Login:

The screenshot shows the 'Admin Login Here' section of the 'Ranking System' website. The header features the site logo, navigation links (Home, Login, Register), and a search bar. Below the header, there are buttons for 'Vote', 'Lists', 'Comments', and 'Members'. The main content area is titled 'Admin Login Here' and contains a 'Login' form with fields for 'User Id' (containing 'admin') and 'Password' (masked with dots). A 'Login' button is at the bottom of the form. To the left is a 'Categories' sidebar with links to Home, Admin Login, User login, Register, blog, Vote, and Top Lists. To the right is a 'Latest Blogs' sidebar with links to Home, Admin Login, User login, Register, and blog, along with a 'Free registration' button.

## Add Book Details:

The screenshot shows the 'Add Book Name' form in the 'Ranking System' website. The header is identical to the previous screenshot, but the navigation links are 'Home', 'Details', and 'Add'. The main content area is a green box titled 'Add Book Name' containing a text input field labeled 'Add Book Name:' and a 'Submit' button.

## User Profile View and Delete:

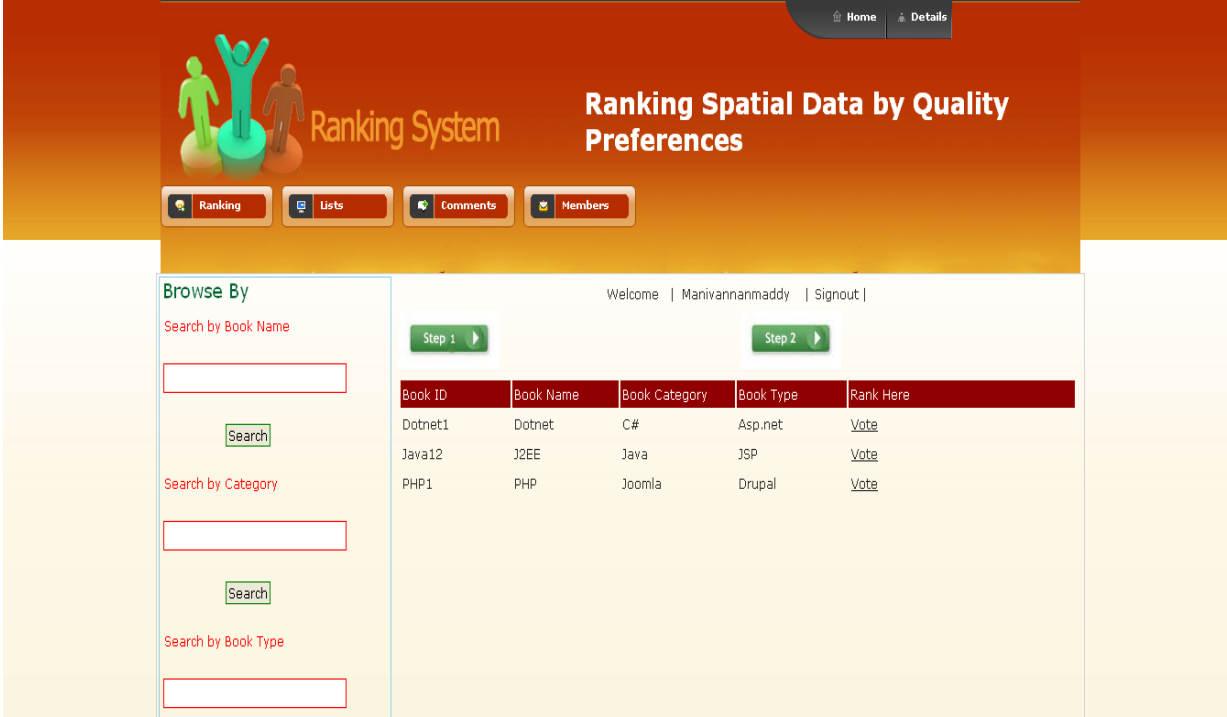
The screenshot shows the 'Ranking System' interface. At the top, there is a navigation bar with 'Home', 'Details', and 'Add' links. The main header features the 'Ranking System' logo and the title 'Ranking Spatial Data by Quality Preferences'. Below the header, there is a search bar with a 'Go' button and two buttons labeled 'Comments' and 'Members'. The main content area displays a table of user profiles. The table has columns for User, First Name, Last Name, DOB, Age, Telephone, Password, Email, Comments, and Delete. Two users are listed: Manivannanmaddy Mani and Thala Ajith kumar. A 'Delete' button is located below the table. The footer of the page reads 'Ranking Spatial Data by Quality Preferences'.

User	First Name	Last Name	DOB	Age	Telephone	Password	Email	Comments	Delete
Manivannanmaddy	Mani	vannan	1988-11-11	23	9840615017	rajammal	manivannanprog@gmail.com	Thanks	<input type="checkbox"/>
Thala	Ajith	kumar	1972-05-01	38	9898989898	godfather	mani@gmail.com	Hi Thala	<input type="checkbox"/>

## User Login:

The screenshot shows the 'Ranking System' user login interface. At the top, there is a navigation bar with 'Home', 'Login', and 'Register' links. The main header features the 'Ranking System' logo and the title 'Ranking Spatial Data by Quality Preferences'. Below the header, there is a search bar with a 'Go' button and four buttons labeled 'Vote', 'Lists', 'Comments', and 'Members'. The main content area is a green box with the title 'Login Your Account'. It contains two input fields: 'UserName:' with the value 'Manivannanmaddy' and 'Password:' with a masked password '.....'. A 'Login' button is located below the input fields.

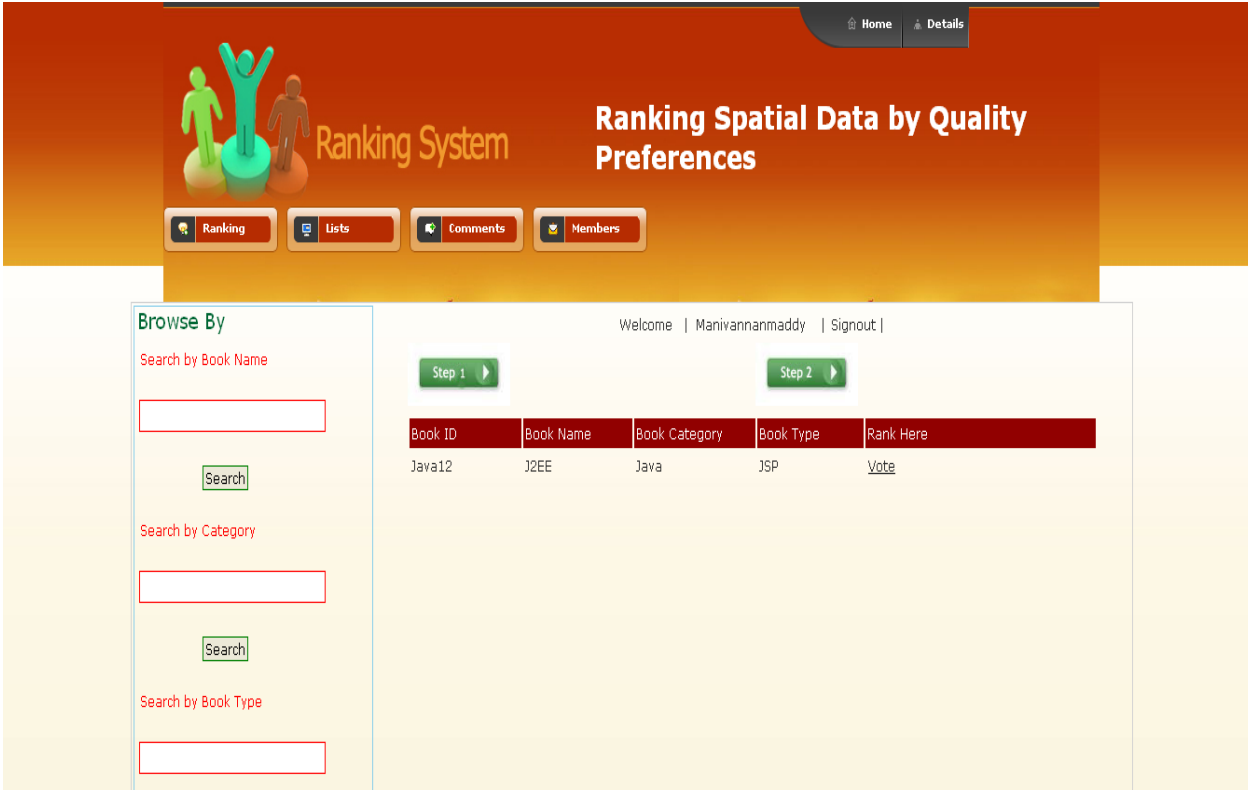
## View Books and Vote:



The screenshot shows the 'Ranking System' web application. The header includes a logo with three stylized figures and the text 'Ranking System' and 'Ranking Spatial Data by Quality Preferences'. Navigation links for 'Home' and 'Details' are in the top right. Below the header are buttons for 'Ranking', 'Lists', 'Comments', and 'Members'. The main content area is titled 'Browse By' and features three search sections: 'Search by Book Name', 'Search by Category', and 'Search by Book Type', each with an input field and a 'Search' button. To the right, a table displays a list of books with columns for Book ID, Book Name, Book Category, Book Type, and Rank Here. The table includes three rows of data and a 'Vote' link for each. Above the table are 'Step 1' and 'Step 2' navigation buttons. The user is logged in as 'Manivannanmaddy' and can click 'Signout'.

Book ID	Book Name	Book Category	Book Type	Rank Here
Dotnet1	Dotnet	C#	Asp.net	<a href="#">Vote</a>
Java12	J2EE	Java	JSP	<a href="#">Vote</a>
PHP1	PHP	Joomla	Drupal	<a href="#">Vote</a>

## Search by Book using Index and Search Engine Technique:



This screenshot shows the same 'Ranking System' web application, but with a search result displayed. The 'Search by Book Name' input field contains 'Java12'. The table now only shows one row of data for 'Java12'. The rest of the interface, including the header, navigation, and other search options, remains the same.

Book ID	Book Name	Book Category	Book Type	Rank Here
Java12	J2EE	Java	JSP	<a href="#">Vote</a>

## View Rank Here:

Welcome | Manivannanmaddy | Signout |

Step 1 Step 2

Book ID	Trustworthy	Objective	Complete	Well-Written	Update/Delete	Conform Votes
Java12	Good	Good	Good	Good	Update   Delete	Conform Votes
Review: Its good						
Dotnet1	very Good	Excellent	Good	very Good	Update   Delete	Conform Votes
Review: Hi						
Dotnet	very Good	Good	Very Good	Excellent	Update   Delete	Conform Votes
Review: Hi						

Browse By

Search by Book Name

Search

Search by Category

Search

Search by Book Type

Search

## View Votes:

Welcome | Manivannanmaddy | Signout |

Step 1 Step 2

Book ID	Trustworthy	Objective	Complete	Well-Written	Your Vote	Get Award
Java12	Good	Good	Good	Good	You Have 12 Votes	Get Award
Review: Its good						
Dotnet1	very Good	Excellent	Good	very Good	You Have 1 Votes	Get Award
Review: Hi						
Dotnet	very Good	Good	Very Good	Excellent	You Have 1 Votes	Get Award
Review: Hi						
PHP1	Excellent	Very Good	Excellent	Good	You Have 1 Votes	Get Award
Review: Its so good to study.						

Browse By

Search by Book Name

Search

Search by Category

Search

Search by Book Type

Search

## Blog Discussion:

**Ranking System**  
Ranking Spatial Data by Quality Preferences

Search  Go

Vote Lists Comments Members

### Forum Discussion

Topic	Last Post
<p> Question: Re : Java</p> <p> Hi</p>	<p>Saturday, 29 July 2011 17:41 Posted by : Mehala</p>
<p> Question: Java</p> <p> Its so Nice</p>	<p>Saturday, 29 July 2011 17:16 Posted by : Amar</p>
<p> Question: Dotnet</p> <p> Its Good</p>	<p>Sunday, 14 May 2011 12:31 Posted by : Manimaddy</p>

## Admin View Members Lists:

Home Login Register

**Ranking System**  
Ranking Spatial Data by Quality Preferences

Ranking Lists Comments Members

Book Name	User	Trustworthy	Objective	Complete	Wall-Written	Hits of User
Java12	Manivannanmaddy	Good	Good	Good	Good	12
Dotnet1	Manivannanmaddy	very Good	Excellent	Good	very Good	1
Dotnet1	billa	very Good	Excellent	Excellent	very Good	2
Dotnet	Manivannanmaddy	very Good	Good	Very Good	Excellent	1
Dotnet1	Thala	Excellent	Very Good	Excellent	very Good	4
PHP1	Manivannanmaddy	Excellent	Very Good	Excellent	Good	1

### **3.SYSTEM TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

#### **TYPES OF TESTS**

##### **Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

##### **Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.



## **Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

## **System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

## **White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

## **Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

### **3.1 Unit Testing:**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

#### **Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

#### **Test objectives**

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

#### **Features to be tested**

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

### **3.2 Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

### **3.3 Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

## CONCLUSION

The neighborhood of an object  $p$  is captured by the scoring function:

- (i) The range score restricts the neighborhood to a crisp region centered at  $p$ , whereas
- (ii) The influence score relaxes the neighborhood to the whole space and assigns higher weights to locations closer to  $p$ .

The algorithm FJ performs a multi-way join on feature trees to obtain qualified combinations of feature points and then search for their relevant objects in the object tree. Based on our experimental findings,  $BB^*$  is scalable to large datasets and it is the most robust algorithm with respect to various parameters. However, FJ is the best algorithm in cases where the number  $m$  of feature datasets is low and each feature dataset is small. In the future, we will study the top- $k$  spatial preference query on road network, in which the distance between two points is defined by their shortest path distance rather than their Euclidean distance. The challenge is to develop alternative methods for computing the upper bound scores for a group of points on a road network.

## **Future Implementation**

In the near future I will improve the search algorithms which will help me find the perfect match for any users query and rank of the spatial and the non – spatial data according to quality preferences.

# BIBLIOGRAPHY

Good Teachers are worth more than thousand books, we have them in Our Department

## 1. Abbreviations

HTTP → Hyper Text Transfer Protocol

CGI → Common Gateway Interface

DNS → Domain Name Service

SSL → Secure Socket Layer

CLB → Client Side load Balancing

OOPS → Object Oriented Programming Concepts

TCP/IP → Transmission Control Protocol/Internet Protocol

JDBC → Java Data Base Connectivity

EIS → Enterprise Information Systems

JNDI → Java Naming and Directory Interface

ORDBMS → Object Relational Database Management System

## **2. References Made From:**

1. Amazon Web Services. Amazon Web Services (AWS). <http://aws.amazon.com>.
2. F5 Networks. F5 Networks. <http://www.f5.com>.
3. Google Inc. Google App Engine. <http://code.google.com/appengine/>.

## **3. Sites Referred:**

<http://java.sun.com>

<http://www.sourcefordge.com>

<http://www.networkcomputing.com/>

<http://www.roseindia.com/>

<http://www.java2s.com/>