APPROACH FOR SEARCHING DATA AND IMPLEMENTATION IN TRAVELLING SYSTEM FOR BEST RESULTS

A Report for the final submission of project

Submitted by

SIDDHARTH YADAV

(1613107062/16SCSE107012)

In partial fulfillment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING WITH SPECALIZATION IN BUSINESS ANALYTICS

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Under the Supervision of

Dr. Dileep Kumar Yadav, M. Tech, P.HD

Associate Professor

APRIL/MAY- 2020

SCHOOL OF COMPUTING AND SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report "APPROACH FOR SEARCHING DATA AND IPLEMENTATION IN TRAVELLING SYSTEM FOR BEST RESULTS" is the bonafide work of "SIDDHARTH YADAV" who carried out the project work under my supervision.

SIGNATURE OF HEAD
DR. MUNISH SHABARWAL,
PhD (Management), PhD (CS)
Professor and Dean,
School of Computing Science & Engineering

SIGNATURE OF SUPERVISOR
DR. DILEEP KUMAR YADAV,
M. Tech(CS) , P.HD(CS)
Associate Professor,
School of Computing Science & Engineering

TABLE OF CONTENTS

ABSTRACT

Web search queries are often ambiguous or multi-faceted, which makes a simple ranked list of results inadequate. To assist information finding for such faceted queries, System explore a technique that explicitly represents interesting facets of a query using groups of semantically related terms extracted from search results. As an example, for the query baggage allowance, these groups might be different airlines, different flight types (domestic, international), or different travel classes (first, business, economy). Groups are named as query facets and the terms in these groups are named as facet terms. A supervised approach is developed which based on a graphical model to recognize query facets from the noisy candidates found. The graphical model learns how likely a candidate term is to be a facet term as well as how likely two terms are to be grouped together in a query facet, and captures the dependencies between the two factors. Two algorithms are proposed for approximate inference on the graphical model since exact inference is intractable. Evaluation combines recall and precision of the facet terms with the grouping quality. Query facets can be extracted from a variety of different resources, such as a query log, anchor text, taxonomy and social folksonomy. In this work, focus is on extracting query facets from the top k web search results D = {D1, D2... Dk}. Intention is to explore the use of other information sources for this problem in future work.

LIST OF TABLE

LIST OF FIGURES

LIST OF SYMBOLS

Q	Users query
N	Network Connection
RDF	Resource Description Framework
OLAP	Online Analytical Processing
API	Application Program interface
SDLC	Software Development Lifecycle
L	Each retrieved list from webpage
R	Read each row
∑	Used for summation notification

Chapter 1

INTRODUCTION

We all humans are different as far as our interests and abilities are concerned. It's not like, that everybody likes painting or everyone is fond of dancing, but yes, many of us have interests in common. Our interests are according to what we like doing more, some people can analyze in a better way than others while others might just learn what they are supposed to learn as it is. In earlier days, while planning to go to a trip people used to ask their friends or family when they were not very much sure about what could be a better plan, but still it was not very satisfying method, so in this era of internet ,we came up with a better solution, Recommendation Systems.
Semantic and place base search is a technique for accessing information organized according to a relational classification system, allowing users to digest, analyze and navigate through multidimensional data. It is widely used in e-commerce and digital libraries. Faceted search is similar to query facet extraction in that both of them use sets of coordinate terms to represent different facets of a query. However, most existing works for faceted search are build on as specific domain or predefined categories, while query facet extraction does not restrict queries in a specific domain, like products, people, etc.

- **Goals**

- Extract the data from Google from third party search engines.
- To deploy the proposed system with Public cloud
- To enhance the system security using security as well as privacy algorithms e.g. SQL injection and prevention, pattern matching apaches.

- **Objectives (Purpose)**

- Enhance the system final ranking using weight and maximum visited page.
- Session storage of search queries
- Third party API used for multiple pages extraction from search engines.

1.1 Dissertation Idea

A great deal of examination works concentrates on database interfaces which help clients to question the social database without SQL Query-By-Example and Query Form are two most generally utilized database questioning interfaces. Current considers and works predominantly concentrate on the most proficient method to create the question shapes.

A query facet is a set of items which describe and summarize one important aspect of a query. Here a facet item is typically a word or a phrase. A query may have multiple facets that summarize the information about the query from different perspectives. For example facets for the query "watches" cover the knowledge about watches in five unique aspects, including brands, gender categories, supporting features, styles, and colors. Query facets provide interesting and useful knowledge about a query and thus can be used to improve search experiences in many ways. In this work, system attempt to extract query facets from web search results to assist information finding for these queries. System define a query facet as a set of coordinate terms { i.e., terms that share a semantic relationship by being grouped under a more general a "relationship". First, system can display query facets together with the original search results in an appropriate way Thus, users can understand some important aspects of a query without browsing tens of pages. For example, a user could learn different brands and categories of watches. System can also implement a faceted search based on the mined query facets. Second, query facets may provide direct information or instant answers that users are seeking. For example, for the query "lost season", all episode titles are shown in one facet and main actors are shown in another. In this case, displaying query facets could save browsing time. Third query facets may also be used to improve the diversity of the ten blue links. System can re-rank search results to avoid showing the pages that are near-duplicated in query facets at the top. Query facets also contain structured knowledge covered by the query, and thus they can be used in other fields besides traditional web search, such as semantic search or entity search.

1.2 Background and Challenges

The existing approach shows the aggregating frequent lists within the top search results to mine query facets and implement a system called QDMiner. More specifically, QDMiner extracts lists from free text, HTML tags, and repeat regions contained in the top search results, groups them into

clusters based on the items they contain, then ranks the clusters and items based on how the lists and items appear in the top results. System propose two models, the Unique Website Model and the Context Similarity Model, to rank query facets. In the Unique Website Model, system assume that lists from the same website might contain duplicated information, whereas different websites are independent and each can contribute a separated vote for weighting facets. However, system find that sometimes two lists can be duplicated, even if they are from different websites. For example, mirror websites are using different domain names but they are publishing duplicated content and contain the same lists. Some content originally created by a website might be re-published by other websites; hence the same lists contained in the content might appear multiple times in different websites. Furthermore, different websites may publish content using the same software and the software may generate duplicated lists in different websites.

1.3 Query Reformulation and Recommendation:

Inquiry reformulation and question proposal (or question recommendation) are two famous approaches to help clients better portray their data need. Question reformulation is the way toward changing an inquiry that can better match a client's data need and question suggestion procedures produce elective inquiries semantically like the first inquiry. The fundamental objective of mining features is not quite the same as question proposal. The previous is to outline the learning and data contained in the question, while the last is to discover a rundown of related or extended inquiries. Notwithstanding, question aspects incorporate semantically related expressions or terms that can be utilized as inquiry reformulations or question proposals now and again. Unique in relation to transitional inquiry proposals, system can use question aspects to create organized inquiry recommendations, i.e., various gatherings of semantically related question recommendations. This conceivably gives wealthier data than conventional inquiry suggestions and might help clients locate a superior question all the more effectively. System will research the issue of producing inquiry suggestions in light of question aspects in future work.

1.4 Query-Based Summarization

Question aspects are a particular kind of outlines that depict the primary point of given content. Existing synopsis algorithms are characterized into various classes as far as their outline development techniques (abstractive or extractive), the quantity of hotspots for the rundown (single document or different records), sorts of data in the synopsis (demonstrative or

enlightening), and the relationship amongst outline and question (nonspecific or inquiry based). It means to offer the likelihood of finding the principle purposes of numerous records and subsequently spare clients' opportunity on perusing entire reports. The distinction is that most existing rundown frameworks commit themselves to generating outlines utilizing sentences separated from archives, while system produce synopses in view of continuous records. What's more, system give back different gatherings of semantically related things, while they give back a level rundown of sentences.

1.5 Entity Search

The issue of element inquiry has gotten much consideration lately. It will probably answer data needs that emphasis on elements. Mining inquiry aspects is identified with substance scan concerning a few questions, feature things are sorts of elements or properties. Some current element look approaches likewise misused information from structure of website pages. Discovering question features contrasts from substance seek in the accompanying perspectives. To start with, discovering question aspects is appropriate for all inquiries, as opposed to simply substance related questions. Second, they tend to return diverse sorts of results. The aftereffect of an element inquiry is substances, their qualities, and related landing pages, while question aspects are included different arrangements of things, which are not as a matter of course elements.

1.6 Query Facets Mining and Faceted Search:

Faceted question is a system for permitting clients to process, break down, and explore through multidimensional information. It is generally connected in e-business and computerized libraries. A hearty audit of faceted pursuit is past the extent of this paper. Most existing faceted inquiry and features era frameworks are based on a particular space, (for example, item look) or predefined aspect categories. For instance, Dakka and Ipeirotis presented an unsupervised procedure for programmed extraction of aspects that are valuable for searching content databases. Feature chains of importance are produced for an entire gathering, rather than for a given question.

Li et al. proposed Facetedpedia, a faceted recovery framework for data revelation and investigation in Wikipedia. Facetedpedia concentrates and totals the rich semantic data from the particular information database Wikipedia. In this paper, system investigate to naturally discover inquiry subordinate features for open-area inquiries in light of a general Web internet searcher. Aspects of

an inquiry are naturally mined from the top web list items of the question with no extra space information required. As inquiry features are great outlines of a question and are conceivably valuable for clients to comprehend the question and help them investigate information, they are conceivable information sources that empower a general open-area faceted exploratory hunt. Like us, Kong and Allan as of late built up an administered approach in light of a graphical model to mine inquiry features.

The graphical model figures out how likely a competitor term is to be an aspect thing and how likely two terms are to be gathered together in a feature. Unique in relation to our methodology, they utilized the supervised strategies. They encourage built up an aspect seek framework in light of the mined features.

Chapter 2

LITERATURE SURVEY

Comprising single or more facet conditions in order to verify, whether the compound term selected by a user is suitable. The second function select only a section of all the facets to display in the user interface, when there are too numerous applicable facets and facet terms. The query refinement process runs as follow. First, the facets in a faceted classification are ranked, and some are chosen to be showed in the user interface. The terms in the selected facet are working by user to form composite terms. Second, the system validates the compound terms selected by the user, and show only the data objects associated with the valid combined terms. At the same time, the system updates the number of data objects subsequent to facet terms in user interface and ranks the facet terms again for the next navigation activity. The iterations continue until expected outcome are establish. Search outcome ranking in the faceted search is like to that in the conventional Information recovery domain. It has been extensively studied for years [3-5]. Thus system will skip it from the following sections. Manual recognition of facet terms by domain experts is costly, wasteful and has poor scalability. There is a few researchers have conducted a preface study of automatic facet term pulling out. Existing automatic removal methods can be divided into three categories corresponding to the different data types: unstructured, semi-structured and structured.

Unstructured data refers to the information that does not adhere to a predefined data model. In facet term extraction, the most regular form of formless data is the natural language text, which is always uncertain and ill-formed. Since machine understanding of normal language text ruins as an open subject, it is very hard to automatically extract facet terms by machine learn techniques only. Current removal methods mainly focus on making comprehensive use of the information of terms, linguistic features of the terms and external knowledge base. The typical methods are outlined as follows.

[1] Zhicheng Dou ET. Al. proposed Automatically Mining Facets for Queries from Their Search Results in IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 28, and NO. 2, FEBRUARY 2017

System proposed a systematic solution, which system refer to as QDMiner, to automatically mine query facets by extracting and grouping frequent lists from free text, HTML tags, and repeat regions within top search results. Experimental results show that a large number of lists do exist and useful query facets can be mined by QDMiner.

Findings

1: It can take search ad-hoc base results base on users query, without generating any database load.

2: It can be work like semantic search approach where user gets the result base on personnel search.

Drawbacks

1: They have mention system having list duplication problem in final results.

2: System does not provide any database security schema.

3: Time complexity issue when user already search the existing query.

[2] DynaCet: Building Dynamic Faceted Search Systems over Databases Roy S.B

System proposed a domain independent system that provides effective minimum-effort based dynamic faceted search solutions over enterprise databases. At every step, Dynacet suggests facets depending on the user response at previous step. Facets are selected based on their ability to rapidly drill down to the most promising tuples, as well as on the ability of the user to provide desired values for them. The benefits provided include faster access to information stored in databases while taking into consideration the variance in user knowledge and preferences.

Findings

1: System provide faster access to information stored in databases while taking into consideration the variance in user knowledge and preferences.

2: It is a domain independent system which provide efficient search.

Drawbacks

1: Fully network dependent approach, can't work offline.

2: It can be work only domain base search, not an aspect of query.

[3]: Zhao, B., et al proposed TEXplorer: keyword-based object search and exploration in multidimensional text databases.

The system proposed propose a novel system TEXplorer that integrates keyword based object ranking with the aggregation and exploration power of OLAP in a text database with rich structured attributes available, e.g., a product review database.

Findings

1: It implemented within a multi-dimensional text database, where each row is associated with structural dimensions (attributes) and text data (e.g., a document).

2: It returns set of documents to find the actual result from k-list

Drawbacks

1: It can work relational database not open source DB.

2: Page ranking not possible is another issue in this system.

[4] Dou, Z., et al proposed finding dimensions for queries

The system proposed problem of finding multiple groups of words or phrases that explain the underlying query facets, which system refer to as query dimensions. System assume that the important aspects of a query are usually presented and repeated in the query's top retrieved documents in the style of lists, and query dimensions can be mined out by aggregating these significant lists.

Findings

1. An efficient approach for generating facets for minimum-effort navigation over enterprise

Databases and

2. Extensions that use uncertainty models over attributes as well as skew in tuple preference introduced by ranked-retrieval models

Drawbacks

1: Sometime false ratio is very high when user enters more complex query..

2: Dimension generation might be take long time into specific user session.

[5] : **Chen, J. and Li, Q proposed Concept Hierarchy Construction by Combining Spectral Clustering and Subsumption Estimation, in Web Information Systems**

The system proposed an architecture automated acquisition of concept hierarchies. In a set of concepts hierarchies generated a weighted graph and generate a results.

Findings

1: Provide large number of list in result section of query.

2: It support all type of dynamic queries and provide a dynamic result.

Drawbacks

1: Some time having link failure problem.

2: The system depend on crawler base datasets.

[6] **Eyal Oren, Renaud Delbru, and Stefan Decker proposed Extending faceted navigation for RDF data**

The system proposed develop an expressive faceted interface for semi-structured data and formally show the improvement over existing interfaces, and develop a prototype for faceted navigation of arbitrary RDF data.

Findings

1: system focus metrics for automatic ranking of facet quality, bypassing the need for manual construction of the interface it will provide better accuracy

2: It generate RDF data like tree structure, it will improve the complexity.

Drawbacks

1: RDF data storage issue when data is high dimensional or big.

2: no security schema has proposed for local database security.

[7] Anick and Tipirneni proposed a facet term extraction algorithm based on the lexical dispersion of words in text. Lexical distribution of a word is the amount of different compounds that contain this word within a document group. The algorithm consists two stages. In the indexing stage documents are parsed so the lexical compounds can be extracting. In the querying stage the

compound appear in the top *n* documents of a ranked result list are used to compute the lexical distribution of each term happening within these compounds. These terms are then sorted by their dispersions and also the top *m* conditions are chosen as candidate terms for following hierarchy construction. The disadvantage of this algorithm is that the removal of facet conditions depends on the specific lexical structure and so can be hardly extended to new domains.

[8] Ling et al. Proposed a two-stage probabilistic method to extract facet terms based on topic model. Given the original keywords from a user, these techniques first apply a bootstrapping algorithm to the document collection to get more correlated terms. Probabilistic combination models are applied to these extended terms to estimate the term distribution of every facet. This is done by at the same time fit the topic model to the data set and restraining the model so that it is secure to the specified definition from the user. The basic thought behind the processes is to guide the subject model with user-defined keywords.

[9] Dakka and Ipeirotis proposed an unsubstantiated automatic facet removal algorithm using exterior resources. This algorithm first identifies the facet term candidates in each document by using third-party term extraction services or algorithms, such as Ling Pipe [10], Yahoo Term Extraction [11], Wikipedia, or Taxonomy Warehouse [12]. Then, each candidate is extended with "context" phrases appearing in external assets by querying WorldNet, Wikipedia, and other online dictionary. This step produces the latent facet terms in the expanded term set, which do not explicitly come out in the documents. Finally, the term distributions in the new term set and the extended term set were compared to identify the terms that can be used to assemble browsing facets. This algorithm has good flexibility and extensibility. However the value of the extracted facets greatly depends on the value of the external resources and term extractor.

The partially structured data does not conform to an unambiguous data diagram; however, it generally contains tags or other markers to divide semantically related essentials. Examples of the semi structured data include HTML pages and the pages annotate by Resource Description Framework (RDF). Partly structured data has an implicit formal structure, which can be exploited to recover the quality of facet term removal. For example, the hyperlinks of web pages can be used to evaluate the magnitude of facet terms. The typical removal methods in this subfield are described briefly as follows.

Chapter 3

PROBLEM STATEMENT AND SCOPE

3.1 Problem Statement

In the proposed research work to design and implement a system than work as classify and re-rank travelling type of query events. The Google API will provide the third party interface for communicate with search engine the classify the all data using machine learning approach and re-rank with page rank as well as click through algorithms. We will collect the data from Google API and rank all the travelling recommendation base on current user query.

3.2 Motivation and Scope

An increasing amount of information consists of a combination of both structured and unstructured data. For example, patent documents contain structured properties such as inventors, assignees, class codes, and ling date, as well as a body of unstructured text.

Helpdesk tickets store not only structured data such as the tickets originator, responsible party, and status, but also text describing the problem and its origin. Increasingly, enterprises want to run analytics on text to extract valuable structured information such as chemical compounds used in a patent and products mentioned in a helpdesk ticket. As a result, the number of unique structured properties in those data sets can be fairly large (from mid to high tens or even hundreds).

Performing discovery-driven analysis on such data sets becomes challenging since a user may not know which properties to focus on. Ideally, a user would like to just type in some keywords into a system which would then guide him to areas of interest. A promising query interface for such mixed data is faceted search, which is widely used by e-commerce sites such as amazon.com and shopping.com for querying their catalogs. For example, a user might enter digital camera in the keyword window of shopping.com. There are potentially thousands of matches, but only a few popular ones can be displayed on the screen. To assist navigation, the system also shows in a separate panel summaries of search results, such as a count of digital cameras in each range of price and resolution (system refer to properties such as price and resolution as facets).

When the user selects a particular price range such as 200300, the system adds a structured constraint. On price to the original query, and refreshes the top matches and the summaries with results from the new query. The navigation process continues until the user needs the desired camera. Faceted search offers several advantages. First, it smoothly integrates free text search with structured querying.

Second, the counts on selected facets serve as context for further navigation. For example, a user might choose to focus his search in the price range that has the most cameras. Todays faceted search systems are designed for browsing catalog data and are not directly suitable for discovery-driven exploration. First, to preserve browsing consistency, facets selected for navigation tend to be static, i.e., they often don't change with di errant keywords.

A typical heuristic rule to select facets is to favor those with more counts. For example, consider a keyword search for XML on a repository of software patents. A traditional faceted search system is likely to present for navigation an assignee facet with values such as IBM and Microsoft, since they have more patents on XML in terms of the absolute counts. While such a result may be useful for certain people, others may nd a startup with only ve patents, but all on XML, to be more interesting. Second, when browsing online catalogs, the navigational facets are single-dimensional only. An important aspect of discovery is to identify interesting correlations, and thus the ability to present facets in pairs, triples, etc. is critical.

System propose an enhanced faceted search system for the kind of discovery-driven analysis that is often performed in On Line Analytical Processing (OLAP) systems. From a potentially large search result, system want to automatically and dynamically discover a small set of facets and values that are deemed most interesting to a user. Using this information, the user can quickly understand important patterns in the query result and can use these patterns to re ne his search. Following earlier work in structured OLAP [21, 22], system de ne interestingness as how surprising or unexpected a summary is, according to some expectation. System make the following new contributions. First, since interestingness is subjective, system allow users to set expectations of their own.

In particular, system propose a novel navigational method of setting a user's expectation that naturally it's how he navigates in a faceted search system. Second, system propose a novel method of measuring the degree of surprise through judicious use of p-values. Our method is unbiased to domain size and makes intuitive sense. Finally, system validate the relevance of our dynamically selected facets by conducting a user survey based on a real data set.

The survey result is positive. For better performance, many traditional OLAP systems pre-compute a data cube [11], and then perform subsequent analysis on the cube itself rather than on the base data. This is impossible when structured data is mixed with text, since maintaining a cube including all possible keywords in the text is prohibitive. Similar to existing faceted search systems, system build a runtime engine on top of an inverted index and dynamically compute aggregations over results returned by the index.

Our dynamic faceted search engine is computationally intensive because it not only considers single-dimensional facets, but also facet combinations. System improve the performance of existing systems by using two new ideas. First, system cache facet data in a compressed bitmap for better space utilization as well as faster set intersections. Second, system develop a novel directory structure called a bit set tree on top of the inverted index to reduce the overhead of unnecessary bit set intersections.

Chapter 4

DISSERTATION PLAN

4.1 Area of Dissertation

A great deal of examination works concentrates on database interfaces which help clients to question the social database without SQL Query-By-Example and Query Form are two most generally utilized database questioning interfaces. Current considers and works predominantly concentrate on the most proficient method to create the question shapes.

A. Adjusted Query Form:

this The devices gave by the database customers make awesome endeavors to offer engineers some assistance with generating the question frames, for example, Easy Query, Cold Fusion etc. They give visual interface to engineers to create or squeeze question frames. The issue of those instruments is that, they are accommodated the specialist engineers. H.V. Jagadish anticipated a system which grants end-customers to change the present inquiry structure at run time. If the database outline is immeasurable, it is troublesome for end customer to discover reasonable database substances and attributes.

B. Computerized Creation of Forms:

M. Jayapandian exhibited an information driven technique it to begin with finds an arrangement of information properties, which are doubtlessly questioned in light of the database pattern and information examples. At that point, the inquiry structures are created taking into account the chose characteristics.

C. Mechanizing the configuration and development of question structures:

H.V. Jagadish introduced a workload-driven strategy. It applies grouping calculation on recorded questions to discover the agent inquiries. The question structures are then created in light of those agent questions. One issue of the previously stated methodologies is that, in the event that system create bunches of inquiry structures ahead of time, there are still client inquiries that can't be fulfilled by any of question structures. Another issue is that, when system produce an extensive number of question shapes, how to let clients locate a suitable question structure would be testing.

D. Joining catchphrase look and shapes:

An answer for earlier stated methodologies is projected in. It accordingly produces a substantial measure of question shapes ahead of time. The client inputs a few catchphrase to discover significant question frames from a significant number of pre-created difficulty shapes yet it is not suitable when the client does not have solid watchwords to depict the investigation

4.2 Plan of dissertation execution

4.2.1 Purpose of the document

The Different Phases identified are:

1. Requirement Analysis.
2. Requirement Specification.
3. System Design.
4. Detailed Design.
5. Coding.
6. Testing.

4.2.2 Estimated Time

Roughly project takes around 8 months for implementation and the month there after for testing.

4.3 Feasibility Study

The feasibility study is major factor which contributes to analysis of system. In earlier stages of S/W development, it is necessary to check weather system is feasible or not. Detail study was carried out to check workability of system, so the feasibility study is system proposal regarding to its workability, impact on organization, ability to meet user requirements and effective use of resources. Thus when application processes it normally goes through feasibility study and risk analysis. Feasibility of project is checked using various categories like:

- Technical Feasibility.

- Economic Feasibility.
- Operational Feasibility.
- Time Feasibility.

4.3.1 Technical Feasibility

Technical study is the study of hardware requirements and software requirements i.e. technical requirements of our project in order to inform a management and users of this application that these many resources are required. Considering all specified requirements, the project is technically feasible.

4.3.2 Economical Feasibility

The economic feasibility will review the expected cost to see it they are in-line with the projected budget or if the project has an acceptable return on investment. At this point, the projected costs will only be a rough estimate. The exact costs are not required to determine economic feasibility. It is only required to determine if it is feasible that the project costs will fall within the target budget or return on investment.

4.3.3 Operational Feasibility

Operational feasibility reviews the willingness of the organization to support the system system. This is probably the most difficult of the feasibilities to gauge. It is necessary to understand the management commitment to the system project, to determine operational feasibility. If the request was initiated by management it is likely that there is management support and the system will be accepted and used. However, it is also important that the employee base will be accepting of the change.

4.3.4 Time Feasibility

Similar to economic feasibility, a rough estimate of the project schedule is required to determine if it would be feasible to complete the systems project within a required timeframe.

4.4 Risk Management

4.4.1 Overview of Risk Monitoring, Mitigation, Management

It will help us to study the major risks in the initial stages. The requirement specification and software specification will be reviewed and studied. All the risks though making negligible impact will be taken into consideration. A proactive approach will be practiced. The risks will be recorded and a best course of action will be planned to avoid the risks. As all risks cannot be avoided, performing risk management, system can attempt to ensure that the right risks are taken at right time. Risk analysis and management are series of steps that help a software term to understand and manage uncertainty. Many problems can plague a software project. A risk is potential problem it might happen, it might not. But, regardless of the outcome, its a really good idea to identify it, access its probability of occurrence, estimate its impact, and establish a contingency plan should the problem actually occur. The key functions of software risk management are to identify, address and eliminate sources of risk before they become threats to successful completion of a software project. An effective strategy must address these issues:

- Risk avoidance
- Risk monitoring
- Risk management and contingency planning.

If a software team adopts a proactive approach to a risk, avoidance is the best strategy. This is achieved by developing a plan for risk mitigation, in the words of Napoleon, If system take so many precautions, it is because system leave nothing to chance. As the project proceeds, risk monitoring activities commence and the project manager monitors factors that may provide indication of whether the risk is becoming more or less likely. Risk management and contingency planning assumes that mitigation sorts have failed and risk has become a reality. RMMM plan tackles risks through Risk Assessment and Risk Control. Risk Assessment involves Risk Identification, Risk Analysis and Risk Prioritization; while Risk Control involves Risk Management Planning,
Risk Resolution and Risk Monitoring.

4.5 Risk Management

Risk analysis and management are series of steps that help a software term to understand and an age uncertainty. Many problems can plague a software project. A risk is potential problem it might happen, it might not. But, regardless of the outcome, it's a really good idea to identify it, access its probability of occurrence, estimate its impact, and establish a contingency plan should the problem actually occur. The key functions of software risk management are to identify, address and eliminate sources of risk before they become threats to successful completion of a software project. An effective strategy must address these issues:

- Risk avoidance
- Risk monitoring
- Risk management and contingency planning.

If a software team adopts a proactive approach to a risk, avoidance is the best strategy. This is achieved by developing a plan for risk mitigation, in the words of Napoleon, If system take so many precautions, it is because system leave nothing to chance. As the project proceeds, risk monitoring activities commence and the project manager monitors factors that may provide indication of whether the risk is becoming more or less likely. Risk management and contingency planning assumes that mitigation sorts have failed and risk has become a reality. RMMM plan tackles risks through Risk Assessment and Risk Control. Risk Assessment involves Risk Identification, Risk Analysis and Risk Prioritization; while Risk Control involves Risk Management Planning,
Risk Resolution and Risk Monitoring.

4.5.1 Project Risk

Specific technical risks including design omissions, version conflicts, operational failures, incompatibilities or bugs. Timing and scheduling risks can include product delivery delays, or missed deadline along the critical path. Human resource risks can involve changes, a lack of skilled resources, non-performance, or the reliability and availability of external service providers. Risks associated with the degree to which software process has been deepened and is followed by the development organization. Customer change requirement becomes critical risk and even further aggravating if the system is in completion stage. Every stage of SDLC system requires whole understanding of requirements.

4.6 Dissertation Schedule

This section presents an overview of dissertation tasks and the output of a dissertation scheduling tool. SDLC paradigm to be used is waterfall model.

4.6.1 Dissertation task set

1. Installations required for dissertation.
2. Preliminary Design of a System.
3. Compilation of Assignments
4. Writing a required Code of system.
5. Testing of a system.
6. Taking results of a system.
7. Comparing results of system with other systems.

4.6.2 Installation and configuration task

Major Tasks in the Dissertation stages are:

Task 1 - Download and install jdk 1.7.0 or higher for windows 7 configuration

Task 2 - Download and install Eclipse Juno which is compatible jdk 1.7

Task 3 – Download and Install MySQL 5.1 and create database and tables, also download and install HeidiDQL for database GUI purpose.

Task 7 - Run Eclipse and create Workspace and provide workspace name

Task 10 - Import Java project Import the source code from the source code directory into workspace

Task 11 - Set up database connectivity. Using JDBC and MYSQL bridge drivers

Task 12 – First go to project properties using right click on project in Eclipse, select Run Tab, and set main class for first execution. Task

Task 13 - Finally run Java Program- Run Java program through Eclipse.

Chapter 5

PROPOSED MODEL

5.1 Introduction

It specifies for the software requirement for the search the particular documents and various activities to be performed.

5.1.1 Purpose and Scope of Document

The purpose of the document to enlist the various software frameworks is required to build the system. The document covers the following points:

- Responsibility of the develop
- System Architecture
- Use cases scenario
- Sequence diagram

5.1.2 Overview of responsibilities of Developer

- Firstly developer has to design a system according to requirement and specification
- The developer has to code for different algorithms
- To create different scenarios and to generate output
- To test the system under above stated scenarios and do necessary changes if required.

5.2 System Architecture and system Algorithm

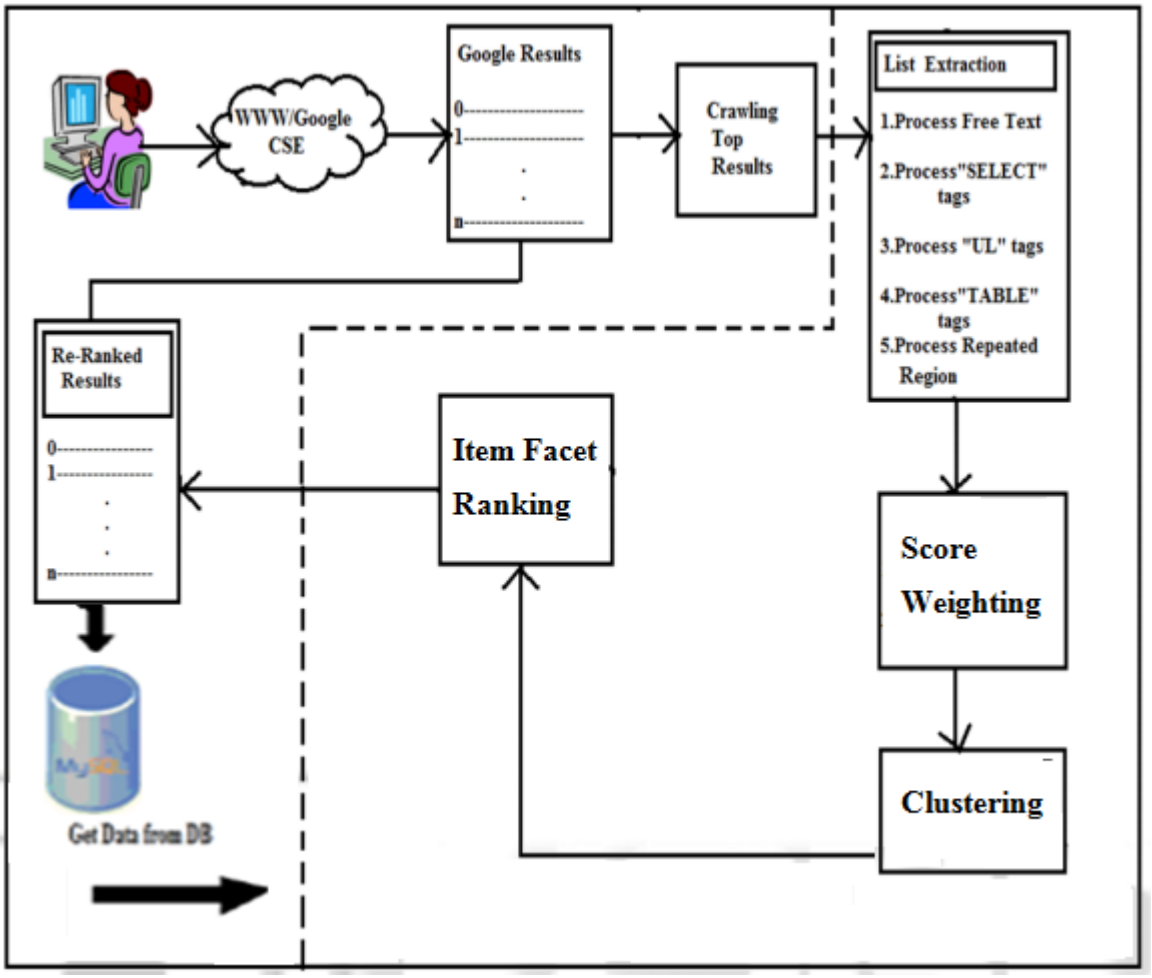


Figure 5.1: Proposed System Architecture

In the proposed research work to design and implement to automatically mine query facets by extracting and grouping frequent lists from free text, HTML tags, and repeat regions within top search results with higher security in cloud framework. Once user submit any query system first check the availability in existing sessions with clustering list. If present query is available in existing session's database it will return all URL's from database. System will deploy the system on Amazon EC2 console as public cloud. It also focus on database security approach using SQL injection and prevention techniques. Using MVC architecture system provide system boosting for generate minimum load on database as well server side, and improve the system performance. Provide database security using Aho-corasick base SQL injection and prevention algorithms. It

store session history of each user into local database it will help for clustering and ranking approach.

5.2.1 Algorithm 1 Computing maximum price that can be there accepted by users

System performs a novel diversity-aware service ranking algorithm to find the optimal top-k Web services based on a proposed comprehensive ranking measure. It is re-lasted work on service recommendation in these three categorists, and on diversity-based ranking algorithms.

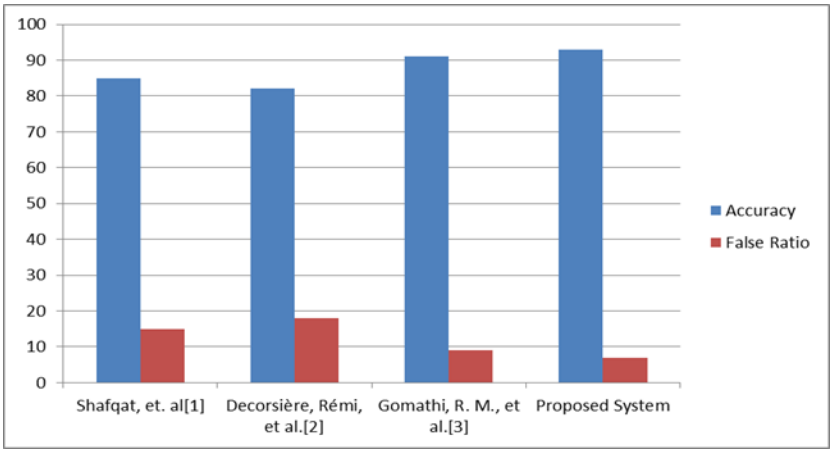


Figure 5.2:Graphical Representation of Proposed Algorithm

Document retrieval Algorithm

Input: Users query as Q, Network Connection N;

Output: result from relevancy calculation top k pages base on Q.

Step 1: User provide the Q to system.

Step 2: if (N! =Null)

Process

Read each attribute A from ith Row in D

Res[i] =Calcsim (Q, A)

Else No connection

Step 3: For each (k to Res)

Step 4: Array list Object array to bind Q to Res[i] or k

Step 5: Return to users Object array

Step 6: Display Object array

Weight Calculation Algorithm

Input: Query generated from user Q, each retrieved list L from webpage.

Output: Each list with weight.

Here system have to find similarity of two lists: $\vec{a} = (a_1, a_2, a_3, \ldots)$ and $\vec{b} = (b_1, b_2, b_3, \ldots)$, where a_n and b_n are the components of the vector (features of the document, or values for each word of the comment) and the n is the dimension of the vectors:

Step 1: Read each row R from Data List L

Step 2: for each (Column c from R)

Step 3: Apply formula (1) on c and Q

Step 4: Score=Calculate(C, Q)

Step 5: calculate relevancy score for attribute list.

Step 6: assign each Row to current weight

Step 7: Categorize all instances

Step 8: end for end procedure

Hash base Ranking Algorithm

Input: Hash map < double; string >;

Output: URL list with weight

Step 1: Read each (k to Hash map)

Step 2: evaluate each Li=ink

=0(Hash map[k])

Step 3: Display Li with maximum weight

Step 4: end for

Step 5: all Li asec order

5.3 Mathematical Model

The system has classified into the different sets like below

Sys= {inp, process, out, analysis}

Inp = {Q1, Q2……Qn}

That is the set of input queries

List= {L1, L2, L3..Ln}

$$\mathrm{LD[w]} = \sum_{k=0}^{n} \binom{n}{k[D]}$$

Extracted list from each documents

L= {Wi1, Wi2, Wi3….. Win} weight of each list using below formula

$$\vec{a} \cdot \vec{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

C= {c1, c2….cn} clusters of each list

[C1…Ck]= $\sum_{k=0}^{n}$ k(classify)

The finally system work with item or facet rank it can create the set of higher dimensions.

UrlList= {URL1 (w), URL2 (w)….. URLn (w)}

$$UL[k] = \sum_{n=1}^{m} \mathrm{Doc1} + \mathrm{Doc2} \ldots\ldots\ldots\ldots \mathrm{Docm}$$

Success condition

If (inp! = Null)

Failure condition

If (UrlList == Null)

5.4 Hardware Resources required

1. Processor : Intel Core i5 -7200 CPU 2.71 GHz
2. RAM : 8 GB RAM
3. Monitor : 15 color
4. Hard Disk : 1 TB
5. Floppy Drive : 1.44 mb
6. CD Drive : lg 52x
7. Keyboard : standard 102 keys
8. Mouse : 3 buttons

5.5 Software Resources required

1. Operating system: Windows 10.

2. Programming Language: JAVA/J2EE

3. Tools: Eclipse, Heidi SQL, JDK 1.7 or Higher

4. Database: MySQL 5.1

5.6 Software Quality Attribute

5.6.1 Reliability

Software Reliability is the probability of failure-free software operation for a specified period of time in a specified environment. Software Reliability is also an important factor affecting system reliability. It differs from hardware reliability in that it reflects the design perfection, rather than manufacturing perfection. The high complexity of software is the major contributing factor of Software Reliability problems. Software Reliability is not a function of time - although researchers

have come up with models relating the two. The modeling technique for Software Reliability is reaching its prosperity, but before using the technique, system must carefully select the appropriate model that can best suit our case. Measurement in software is still in its infancy. No good quantitative methods have been developed to represent Software Reliability without excessive limitations. Various approaches can be used to improve the reliability of software, however, it is hard to balance development time and budget with software reliability.

5.6.2 Availability

- Over-engineering, which is designing systems to specifications better than minimum requirements.
- Duplication, which is extensive use of redundant systems and components.
- Recoverability, which is the use of fault-tolerant engineering methods.
- Automatic updating, which is keeps OSs and applications current without user intervention.
- Data backup , which prevents catastrophic loss of critical information.
- Data archiving , which keeps extensive records of data in case of audits or other recovery needs?
- Power-on replacement, which is the ability to hot swap components or peripherals.
- The use of virtual machine s, which minimizes the impact of OS or software faults.
- Use of surge suppressor s, which minimizes risk of component damage resulting from power-line anomalies.
- Continuous power, which is the use of an un interruptible power supply keeps systems operational while switching from commercial power to backup or auxiliary power.
- Backup power sources, which includes batteries and generators to keep systems operational during extended interruptions in commercial power.

5.6.3 Security

When the security functionality in a system product does not satisfy specific security requirements then the risk introduced must be evaluated and additional controls must be reconsidered prior to purchasing the product. Where additional functionality is supplied and causes a security risk, this must be disabled or the system control structure must be reviewed to determine if advantage can be taken of the available enhanced functionality.

Design reviews must be conducted at periodic intervals during the development process to assure that the system design will satisfy the functional and security requirements specified by the owner.

- applying the security requirements to the project and allocating financial, technical and human resources as required for meeting the security requirements of the project
- ensuring that the security controls are tested and validated during acceptance test phase
- maintaining the security controls throughout the life cycle of the product or the application
- Product or service specifications must include the requirements for security controls. Contracts with the Providers must also address the identified security requirements.

5.6.4 Maintainability

The following steps should be undertaken to assess maintainability statically:

- A list of maintainability factors to be included in the assessment should be devised e.g. structure, complexity.
- Each factor (or group of factors) should be assigned a weighting to indicate its importance to the overall maintainability of the system. Each factor will have a maximum score of 10. The higher the score the less maintainable the system.
- During the assessment a score is awarded against each factor on the list. For example, a relatively old system may be awarded a score of 8 out of 10 to indicate that due to its age the system will relatively difficult to maintain.
- The scores for each of the factors assessed are then multiplied by the appropriate weighting and the resultant products are then summed to give an overall score which forms the Maintainability Measure of the system (the lower the score, the better the maintainability of the software system).
- Example factors which can be used in a maintainability assessment are given below; the list is not exhaustive and should be modified to suit an individual organization (although it is helpful if the same list is used throughout the organization so comparisons between systems can be made):

1: Size	Maintainers' perception
2: Complexity	Environmental facilities
3: Structure	Maintenance relationships
4: Development process	System users/customers
5: Documentation	Maintenance team
6: Development team	Test facilities
7: Development timescale	Operating procedures
8: Maintenance procedures	Problem change traffic
9: Development relationships	Business change traffic

Chapter 6

IMPLEMENTATION

During this project the system solution are investigating and presenting the new framework for addressing the problem of finding relevant result. The aim of this project was to improve the performance of algorithm presented in base system. The results demonstrated in this project are showing the current state of work done over practical implementation of this algorithm.

6.1 Modules

1. User Module:
    i. User Registration and User Authentication Modules
    ii. Data Gathering and retrieval: From user history, Google API data, users rating on social sites
    iii. List Weighting that is Page rank and Algorithms:
    iv. Clustering that is Top-K Recommendation:
    v. OTP Generation
    vi. Book Ticket: Send ticket confirmation on mail.
2. Bank Module:
    i. Add Account
    ii. Credit Amount
    iii. Debit Amount
    iv. View Balance

6.2 Module Description

6.2.1: User Authentication and query submission Module:

This module can be done user authentication module with user can submit the serach query with this form.

6.2. 2: Document retrieval:

This second module can express the document retrieval approach from different web pages and extract the data.

6.2.3: List extraction:

This phase can work with extract the list from each page and store into database simultaneously.

6.2.4: List weighting module

In that phase list weighting has done suing vector base cosine similarity algorithm. Na process each list with specific weight.

6.2.5: Clustering module

This module can done the list clustering approach, it can make the multiple clusters of available lists.

6.2.6: Facet ranking and analysis module

Finally system provide the ranking with analysis graphs of system

Chapter 7

DETAILED DESIGN DOCUMENT

7.1 UML Diagrams:

7.1.1. Activity Diagram

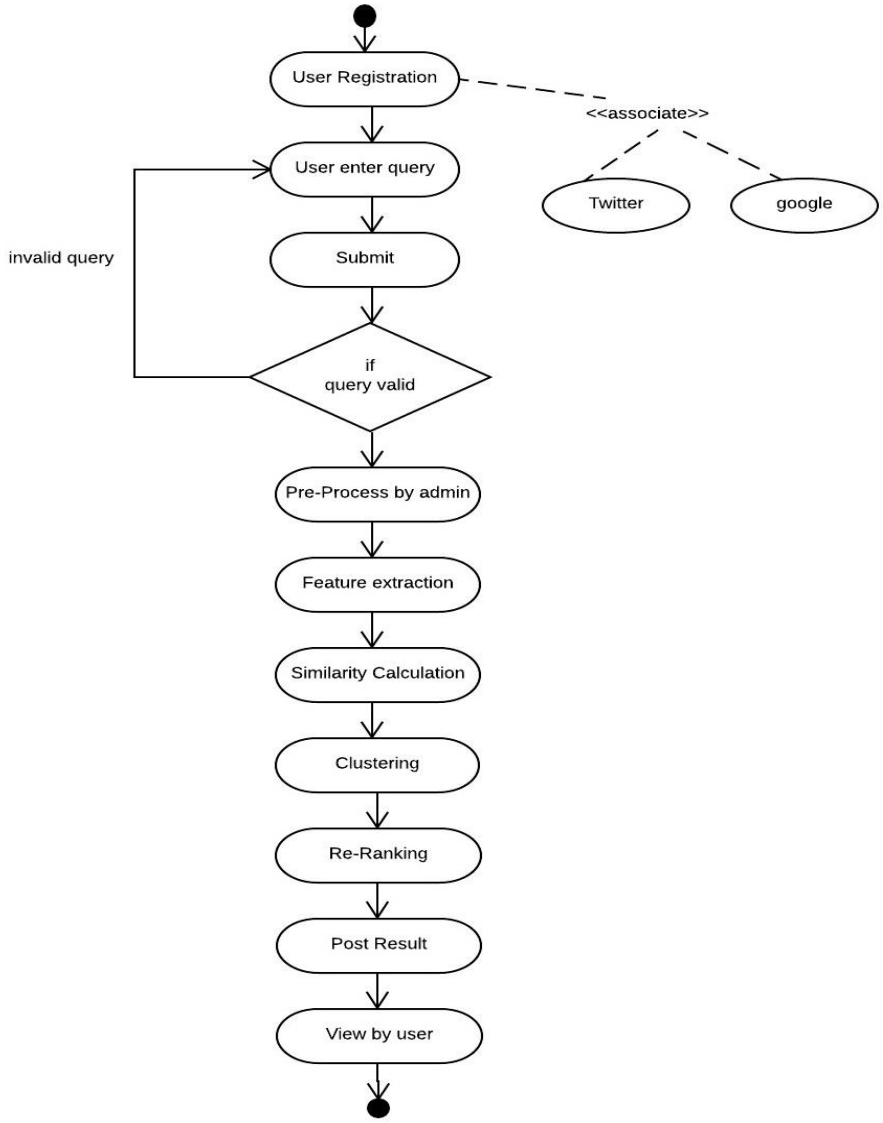


Figure 7.1: Activity Diagram

7.1.2. Sequence Diagram

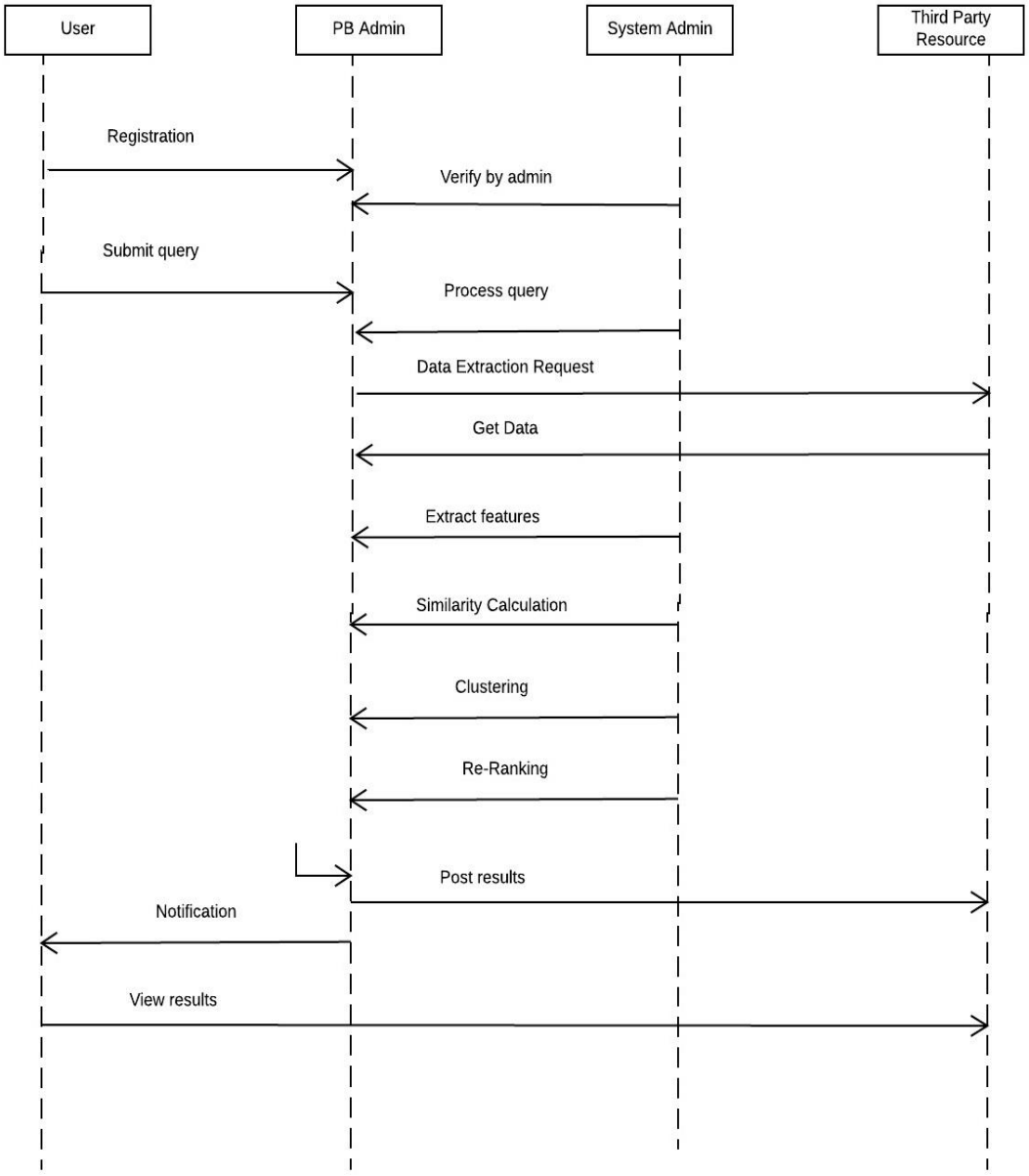


Figure 7.2: System Diagram

7.1.3. Use Case Diagram

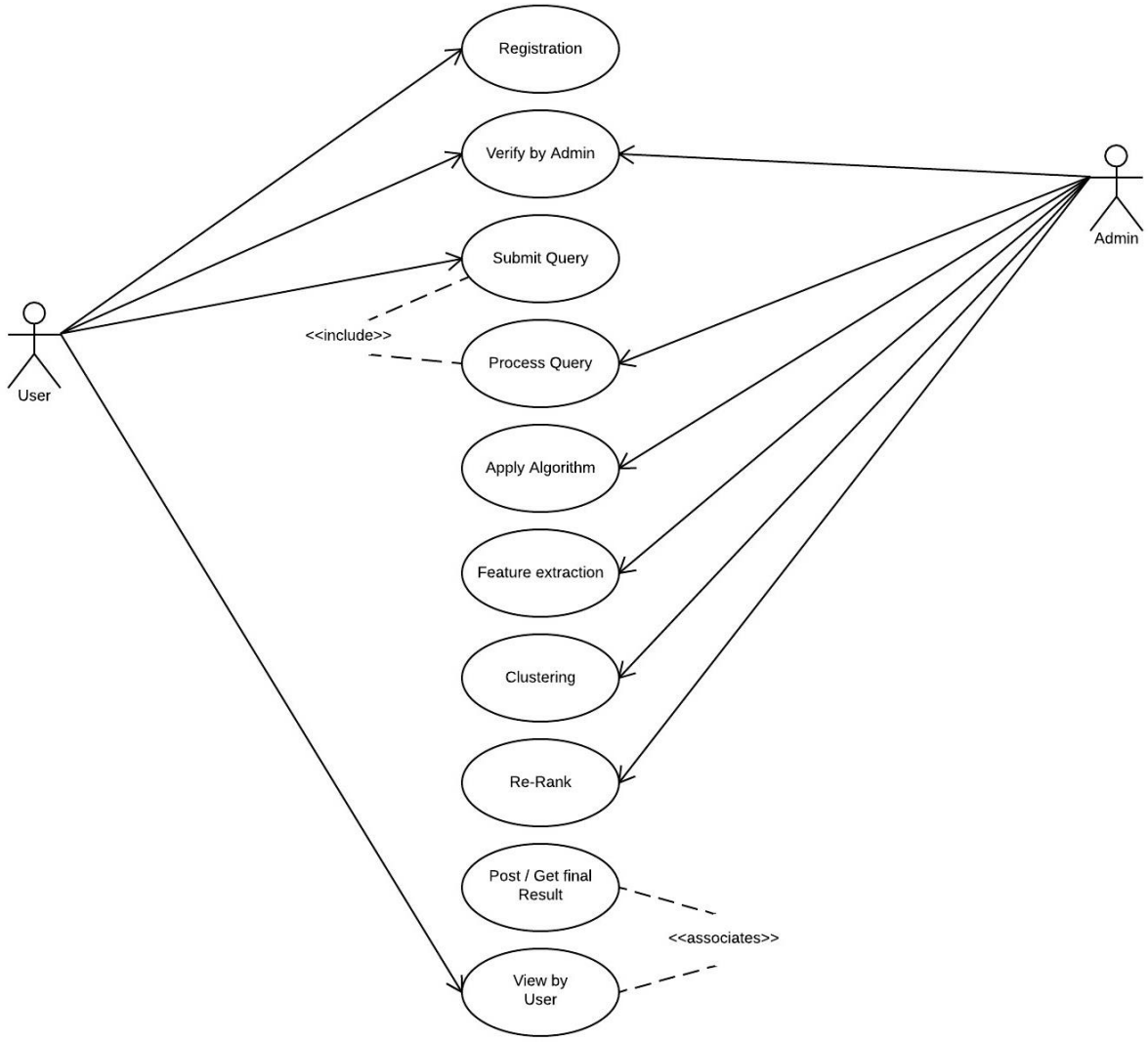


Figure 7.3: Use Case Diagram

7.1.4 Use Case Diagram – Social Ranking Implementation in Travel System

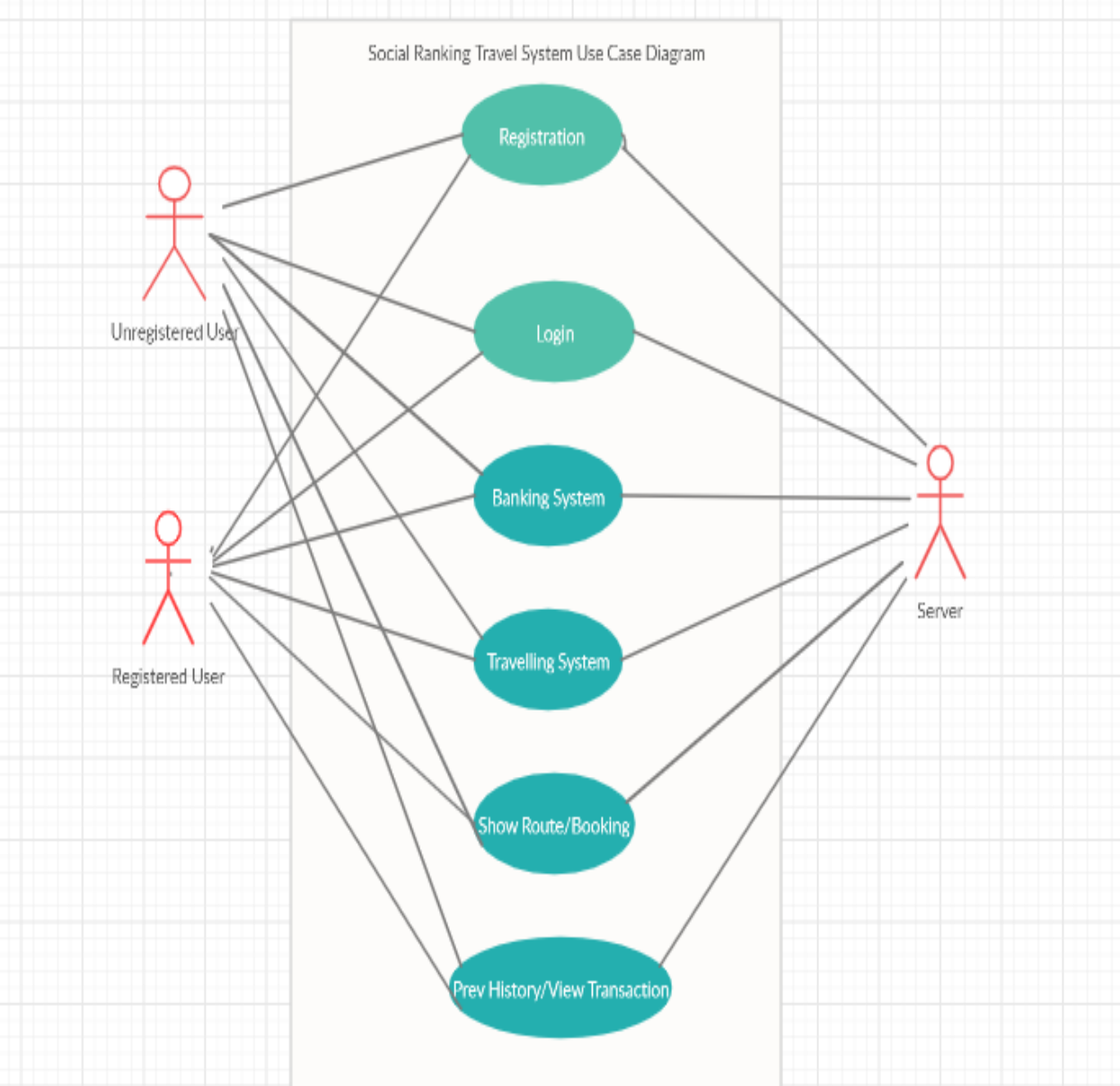


Figure 7.4: Use Case Diagram for Social ranking Travelling System

7.1.5 Use Case Diagram – User Interaction

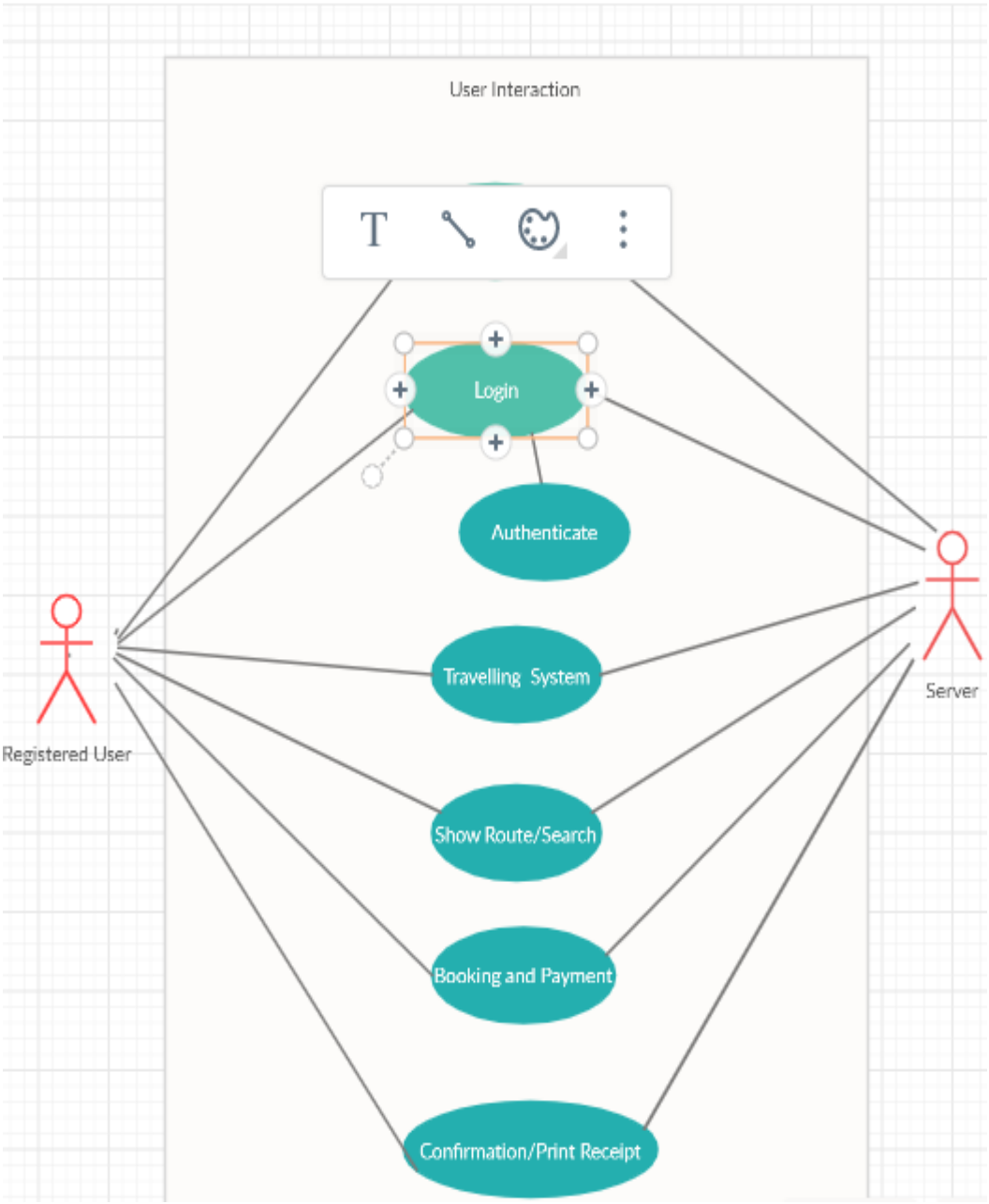


Figure 7.5: Use Case Diagram User Interaction

7.1.6 Use Case Diagram – Travelling System

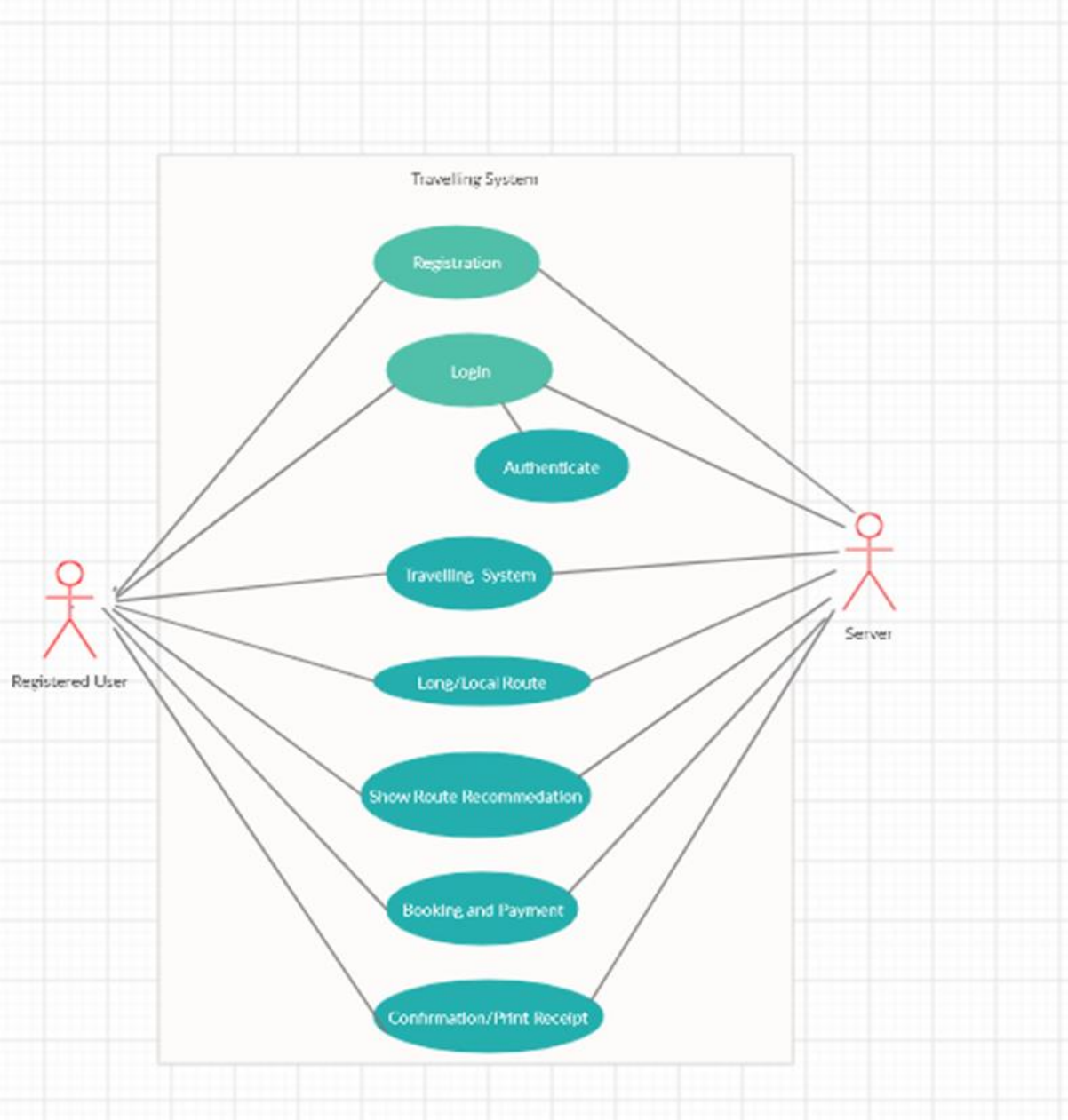


Figure 7.6: Use Case Diagram Travelling

7.1.7 Use Case Diagram – Banking System

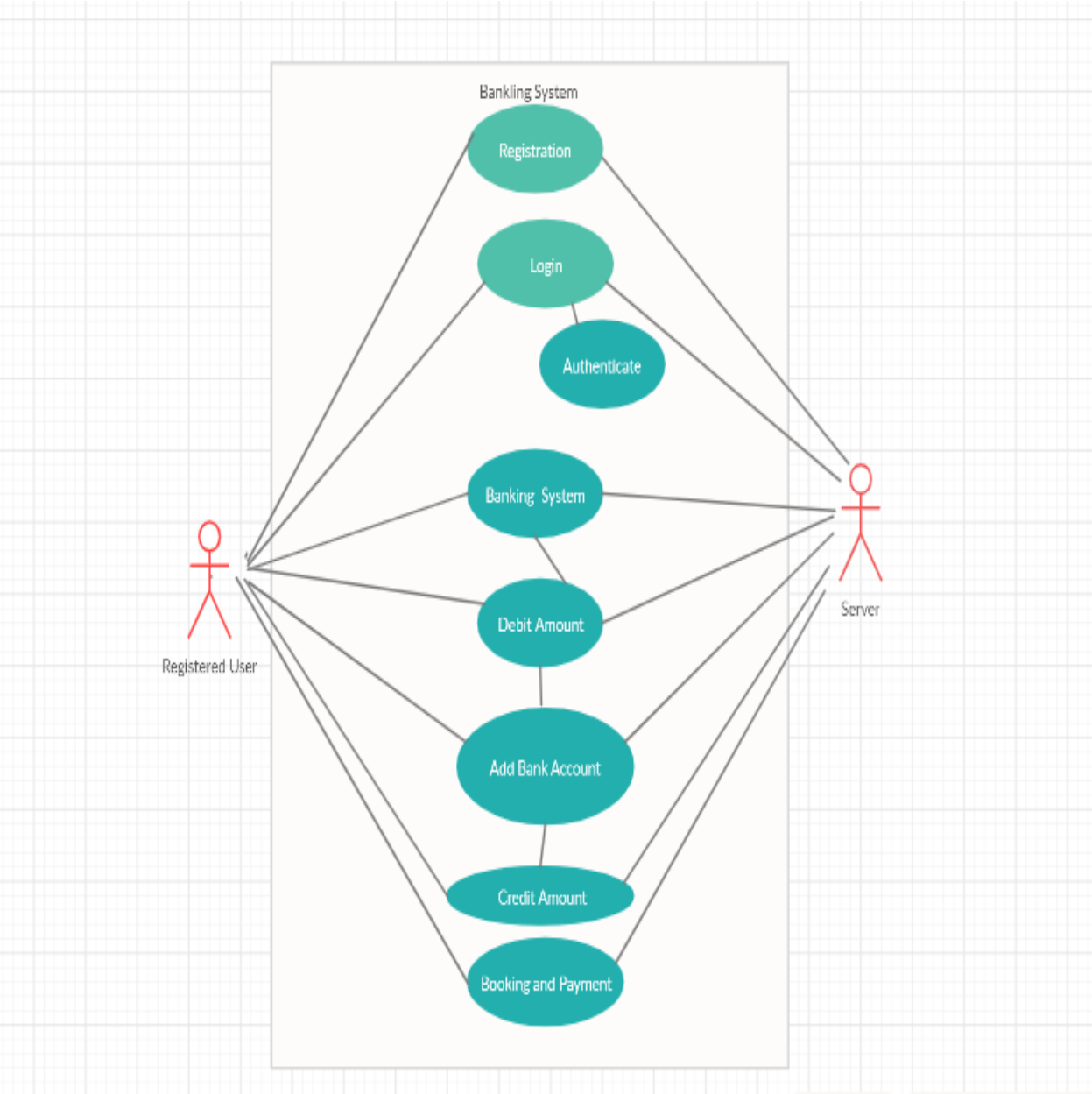


Figure 7.7: Use Case Diagram Banking System

7.2.1. Class Diagram

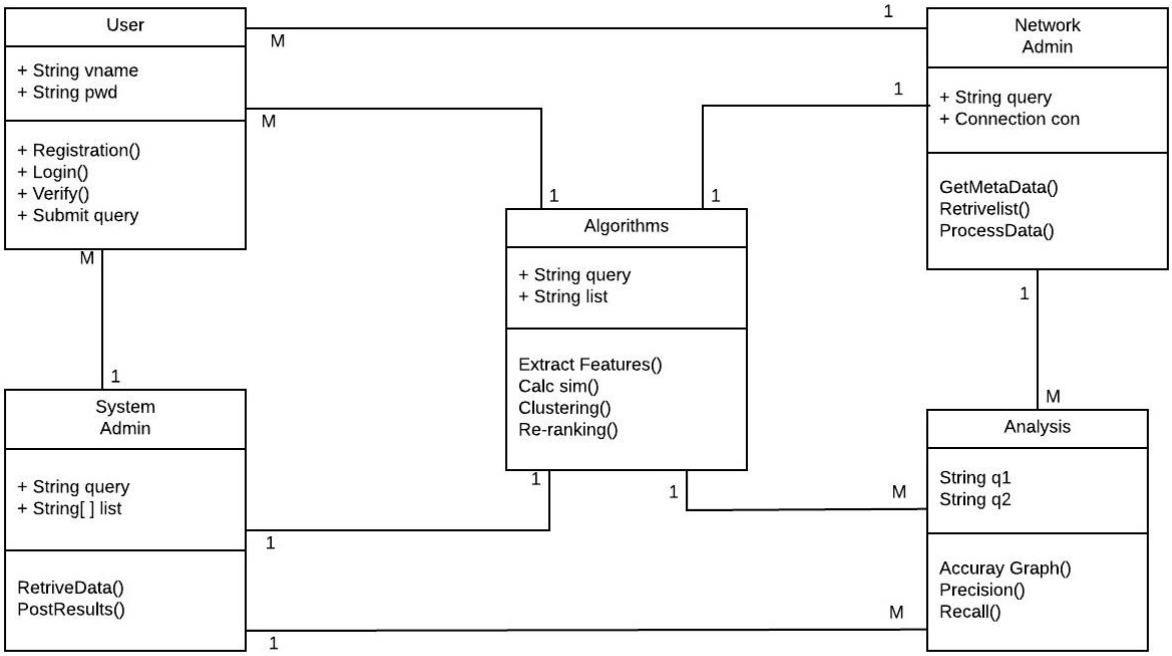


Figure 7.8: Class Diagram

7.3 Data Flow Diagrams:

Graphical description of system's data and how the processes

Transform the data is known as Data Flow Diagram (DFD).

To construct DFD, we use:

- Arrows,
- Circles
- Open – ended boxes
- Squares

Data flow diagrams (DFD)

Data flow diagrams (DFDs) reveal relationships among and between the various components in a program or system. DFDs are an important technique for modeling a system's high-level detail by showing how input data is transformed to output results through a sequence of functional transformations. DFDs consist of four major components: entities, processes, data stores, and data flows. The symbols used to depict how these components interact in a system are simple and easy to understand, however, there are several DFD models to work from, each having its own semiology. DFD syntax does remain constant by using simple verb and noun constructs. Such a syntactical relationship of DFDs makes them ideal for object-oriented analysis and parsing functional specifications into precise DFDs for the systems analyst.

Data flow diagram is a graphical tool used to describe analyze the movement of data through a system manual or automated including the processes, stores of data, and delays in the system. Data flow diagrams are the central tool and basis for form which other components are developed. The data flow diagram is also known a data flow graph or bubble. Data flow diagrams illustrate how data is processed by a system in terms of inputs and output. Data flow diagrams (DFDs) are the method of choice over technical descriptions for three principal reasons.

1. DFDs are easier to understand by technical and nontechnical audiences

2. DFDs can provide a high level system overview, complete with boundaries and connections to other system.

3. DFD's represent the following:

1. External devices sending and receiving data

2. Processes that change that data.

3. Data flows themselves.

4. Data storage locations.

The hierarchical DFD typically consists of a top-level diagram (Level 0) underlain by cascading lower level diagrams (Level 1, Level 2…) that represent different parts of the system.

DFDs can provide a detailed representation of system components.

Data Flow:

Data flow is the movement of data between the entity, the process, and the data store. Data flow portrays the interface between the components of the DFD. The flow of data in a DFD is named to reflect the nature of the data used (these names should also be unique within a specific DFD). Data flow is represented by an arrow, where the arrow is annotated with the data name.

Context diagram:

A context diagram is a top level (also known as level 0) data flow diagram. It only contains one process node (process 0) that generalizes the function of the entire system in relationship to external entities.

The top-level diagram is often called a "context diagram". It contains a single process, but it plays a very important role in studying the current system.

The context diagram defines the system that will be studied in the sense that it determines the boundaries.

Anything that is not inside the process identified in the context diagram will not be part of the system study.

It represents the entire software element as a single bubble with input and output data Advantages and Disadvantages of DFDs.

Advantages:

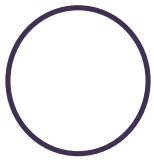
As we have seen, the DFD method is an element of object-oriented analysis and is widely used. Use of DFDs promotes quick and relatively easy project code development. DFDs are easy to learn with their few-and simple-to-understand symbols (once you decide on a particular DFD model). The syntax used for designing DFDs is simple, employing English nouns or noun adjective- verb constructs.

Disadvantages:

DFDs for large systems can become cumbersome, difficult to translate and read, and be time consuming in their construction. Data flow can become confusing to programmers, but DFDs are useless without the prerequisite detail.

Indicated by incoming and outgoing arrows respectively.

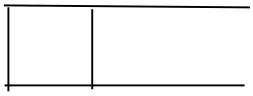
System components



Symbolizes process

Symbolizes data flow

Symbolizes External entity



Symbolizes data store

The data flow diagram is one of the most important tools used for the system analysis. ADEMACRO (1978) and SARSON (1979) populated the use of data flow diagrams as modeling tool through their structured analysis methodologies. They suggested that a data flow diagram should be the first tool used by the analyst to model the system components. There are four types of system components.

Process:

Process shows what system does. A process is represented by a circle as shown above. Each process is given a unique name and a unique number. Each process takes one or more data inputs and produces one or more data outputs.

Data flows:

Data flows made the passage of data and are represented by the line joining the system components. An arrow is used to indicate the direction of data flow shown in the above and the line is labeled by the name of the data flow.

Data stores:

Data store is used to represent the repository of the data that maintains in the system. A process can store data into a data store or retrieve the data from the data store. A data store is represented by a thin line as shown in the above.

External Entities:

External entities are the outside the system but they either supply input into system or use system output. The designer has to control over these entities. These are represented by a square or rectangle outputs as shown in the above.

The context level data flow diagram shown above gives a brief idea about the flow of data should be and it is also shows the control if some data are invalid.

7.3.1. DFD-0 Diagram

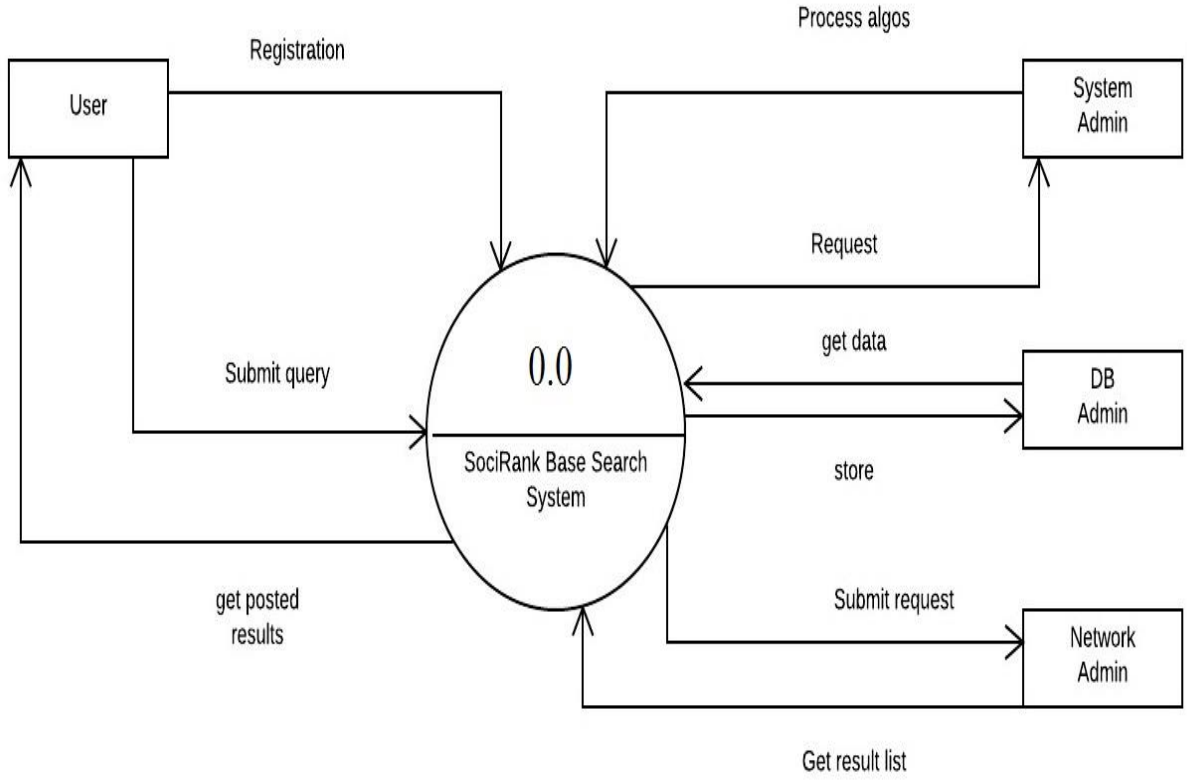


Figure 7.9: DFD Level-0 Diagram

7.3.2. DFD-1 Diagram

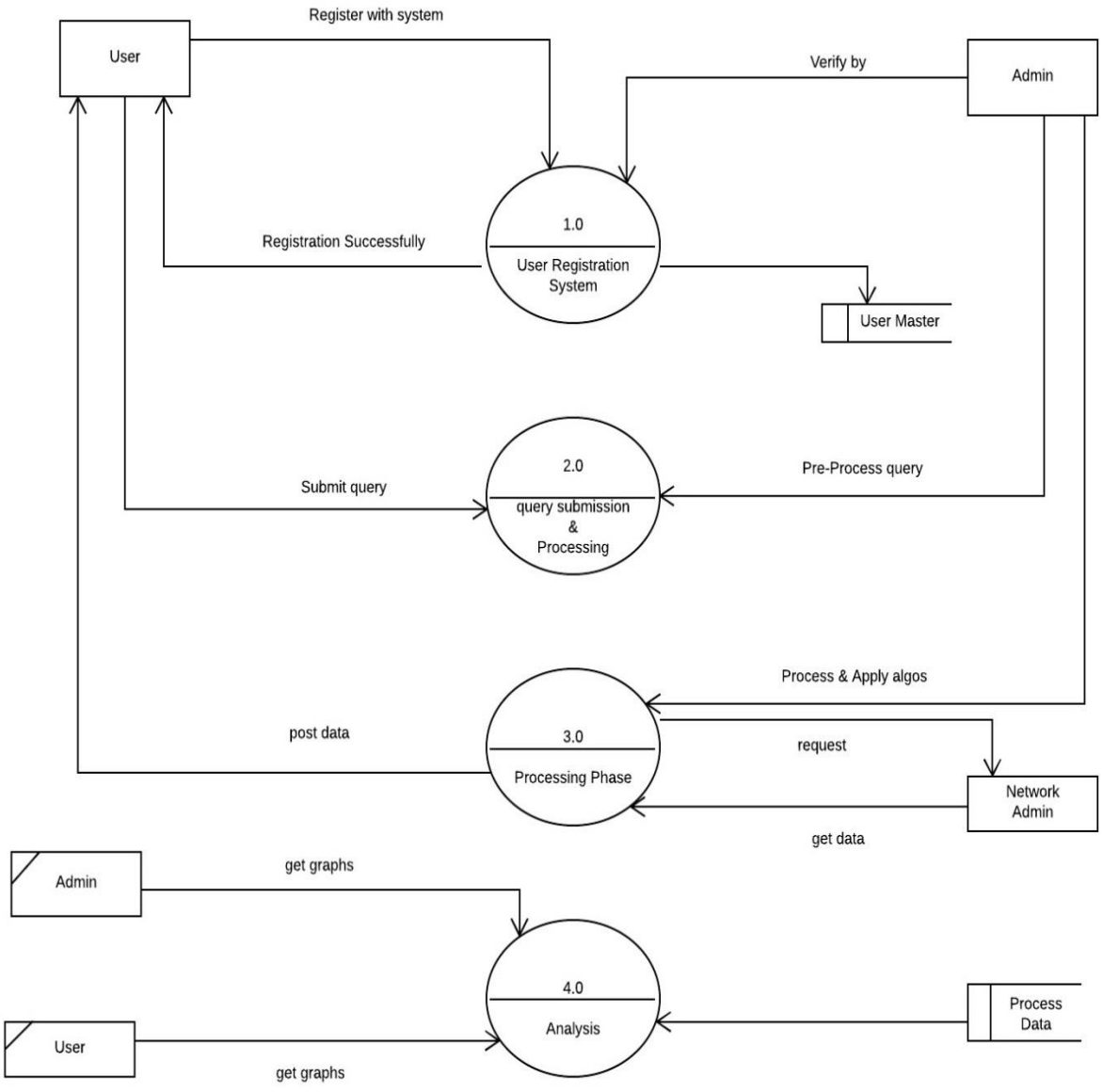


Figure 7.10: DFD Level-1 Diagram

7.3.3 DFD-2 Diagram

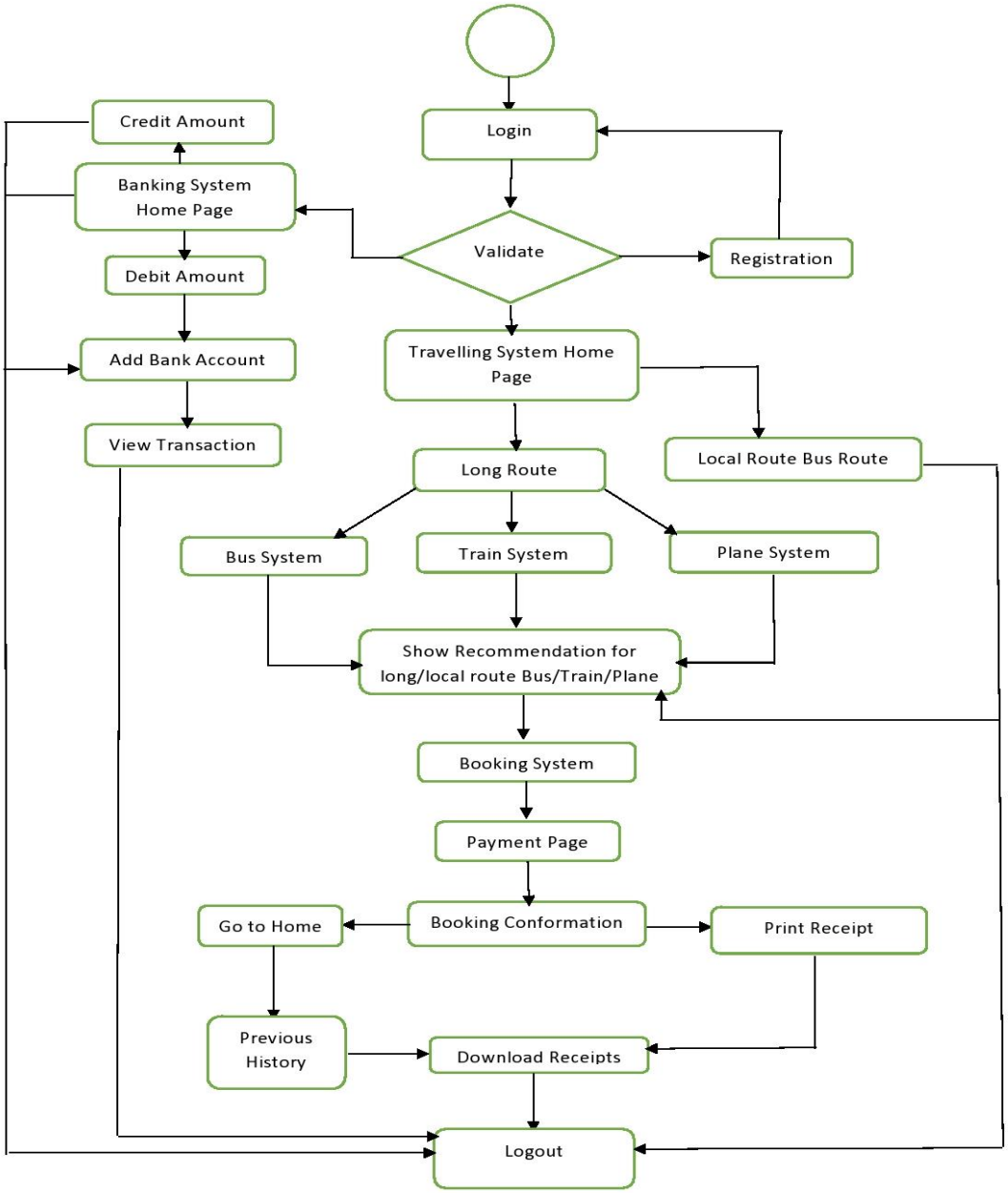


Figure 7.11: DFD Level-2 Diagram

7.4. Hierarchy Diagram

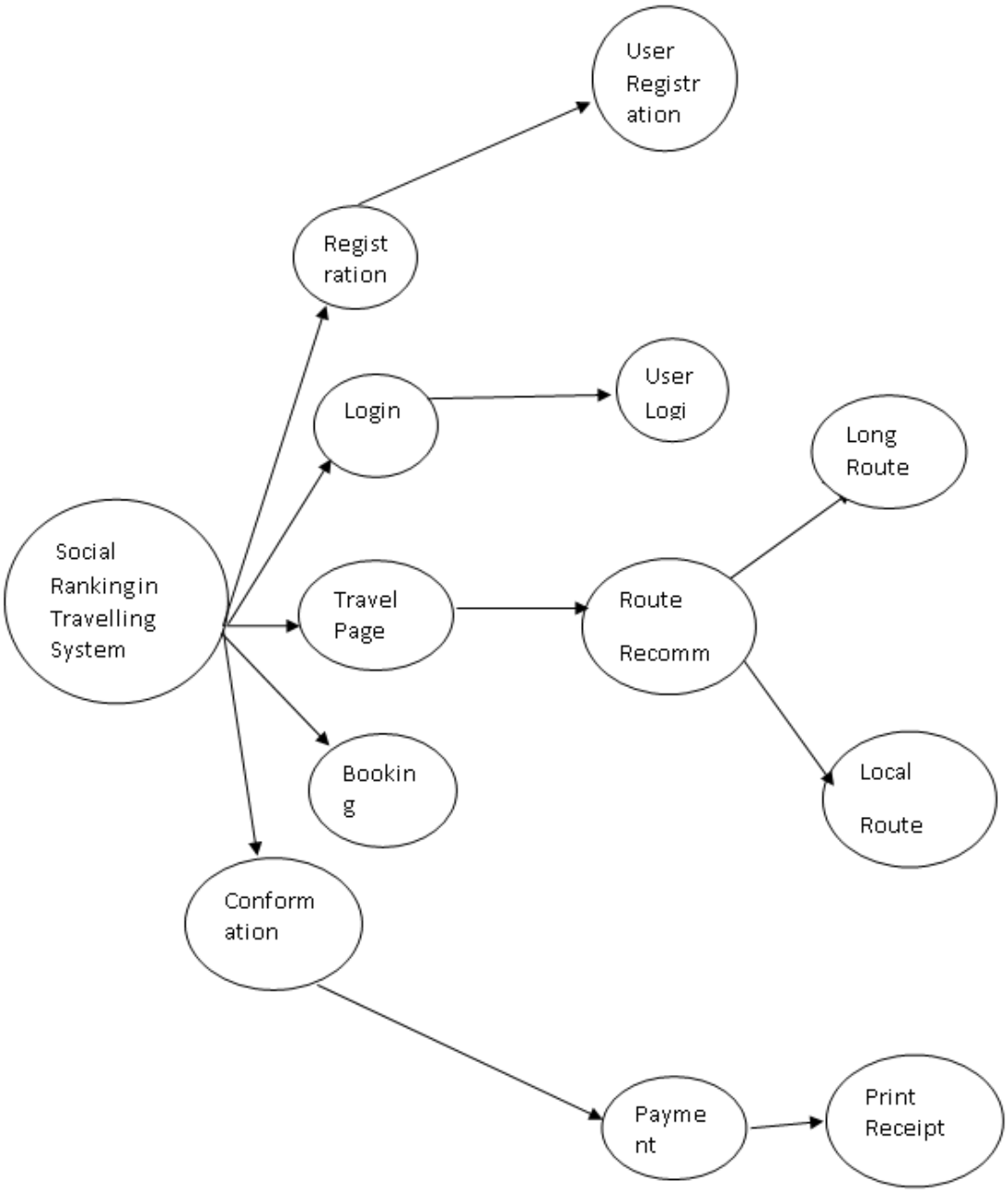


Figure 7.12: Hierarchy Diagram

7.5. Deployment Diagram

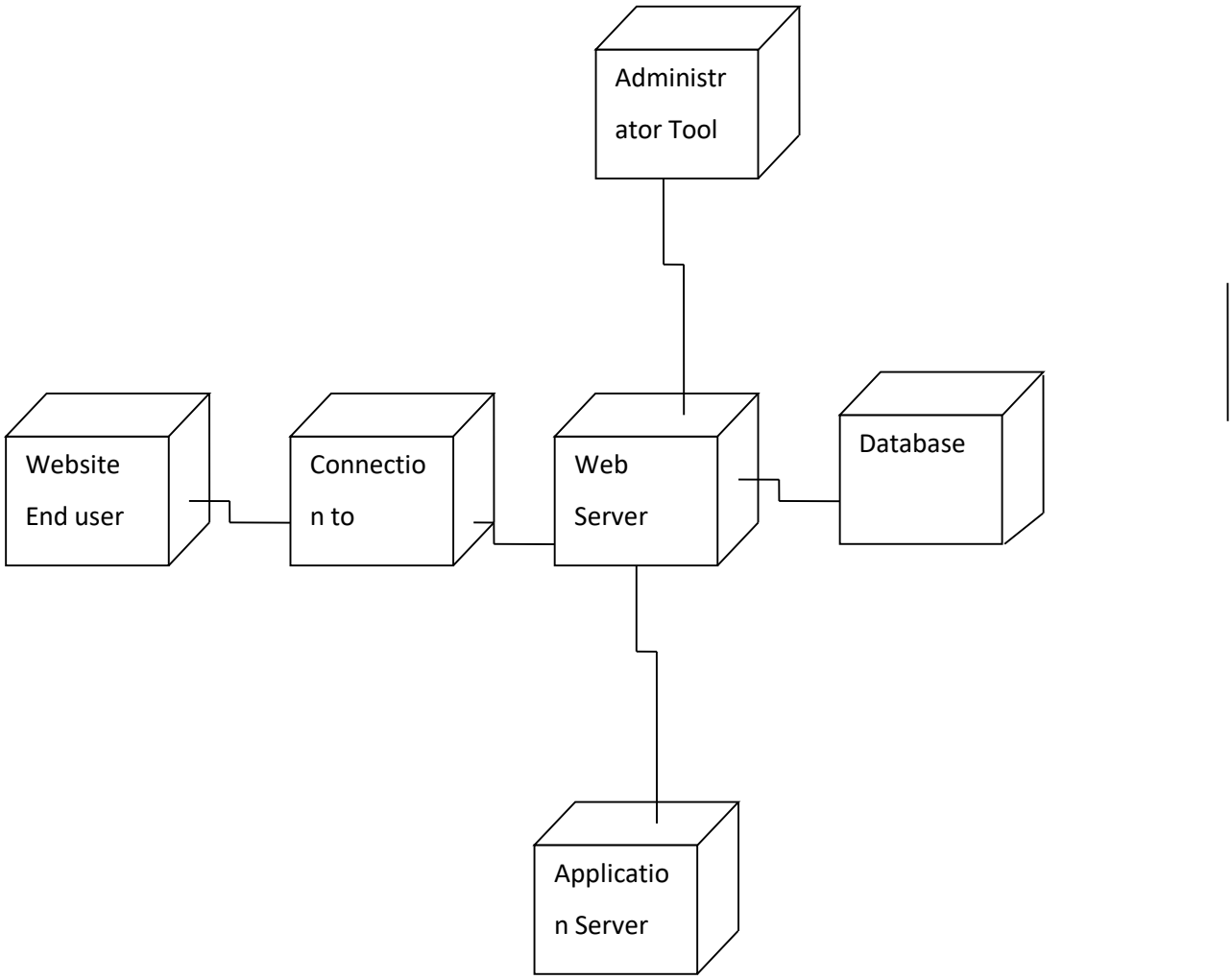


Figure 7.13: Deployment Diagram

7.6. Table specifications:

Table Design

1. Table Name - travelling.information

Description - This table stores travelling information registered user.

Field Name	Data Type	Size	Constraint
UserName	Varchar	100	Not Null
Address	Varchar	100	Not Null
City	Varchar	100	Not Null
Gender	Char	10	Not Null
Emailid	Varchar	100	Not Null
MobNo	Varchar	10	Not Null
Password	varchar	20	Not Null

2. Table Name - travelling.localpmtroute

Description - This table stores local route reservation details.

Field Name	Data Type	Size	Constraint
Id	int	11	Not Null
src	text	100	Not Null
dest	text	100	Not Null
points	text	10	Not Null
routeno	text	50	Not Null
distance	int	100	Not Null
time	text	100	Not Null
price	double	100	Default Null

3. Table Name - travelling.longbusroute

Description - This table stores long bus route reservation details.

Field Name	Data Type	Size	Constraint
Id	int	11	Not Null
src	text	100	Not Null
dest	text	100	Not Null
points	text	10	Not Null
routeno	text	50	Not Null
distance	int	100	Not Null
time	text	100	Not Null
price	double	100	Default Null
bordingpoint	text	100	Not Null
timeStamp	datetime	DD-MM-YYYY	Default Null

4. Table Name - travelling.longplaneroute

Description - This table stores long plane route reservation details.

Field Name	Data Type	Size	Constraint
Id	int	11	Not Null
src	text	100	Not Null
dest	text	100	Not Null
points	text	10	Not Null
routeno	text	50	Not Null
distance	int	100	Not Null
time	text	100	Not Null
price	double	100	Default Null
bordingpoint	text	100	Not Null
timeStamp	datetime	DD-MM-YYYY	Default Null

5. Table Name - travelling.longtrainroute

Description - This table stores long train route reservation details.

Field Name	Data Type	Size	Constraint
Id	int	11	Not Null
src	text	100	Not Null
dest	text	100	Not Null
points	text	10	Not Null
routeno	text	50	Not Null
distance	int	100	Not Null
time	text	100	Not Null
price	double	100	Default Null
bordingpoint	text	100	Not Null
timeStamp	datetime	DD-MM-YYYY	Default Null

6. Table Name - travelling.transaction

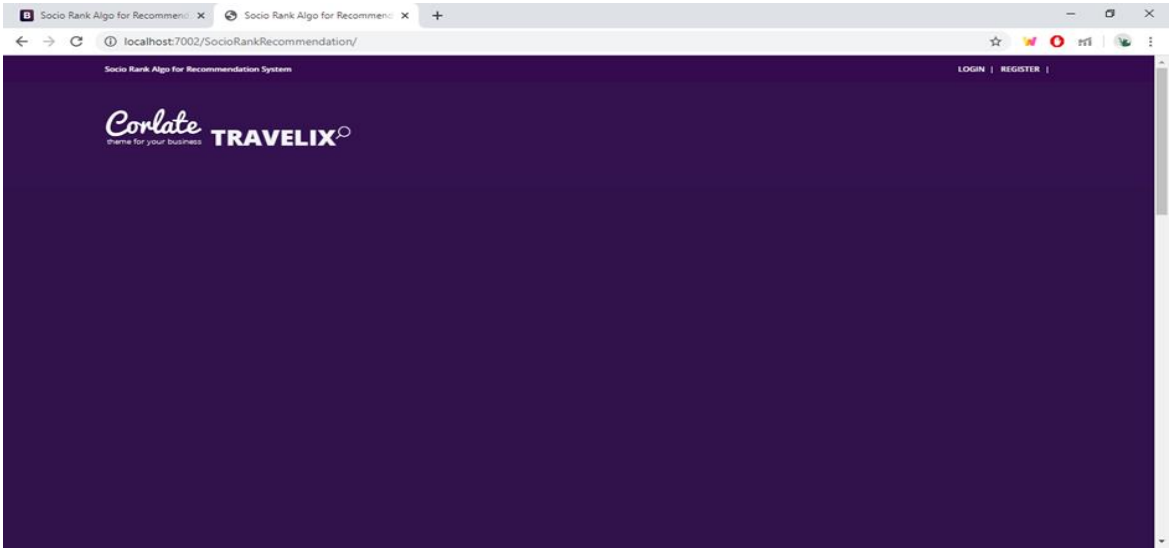
Description - This table stores all transaction details of booking.

Field Name	Data Type	Size	Constraint
User	text	11	Not Null
Bank	text	100	Not Null
CreditAmount	text	100	Not Null
DebitAmount	text	10	Not Null
CurrentAmount	text	50	Not Null
DateTime	timestamp	100	Default Null

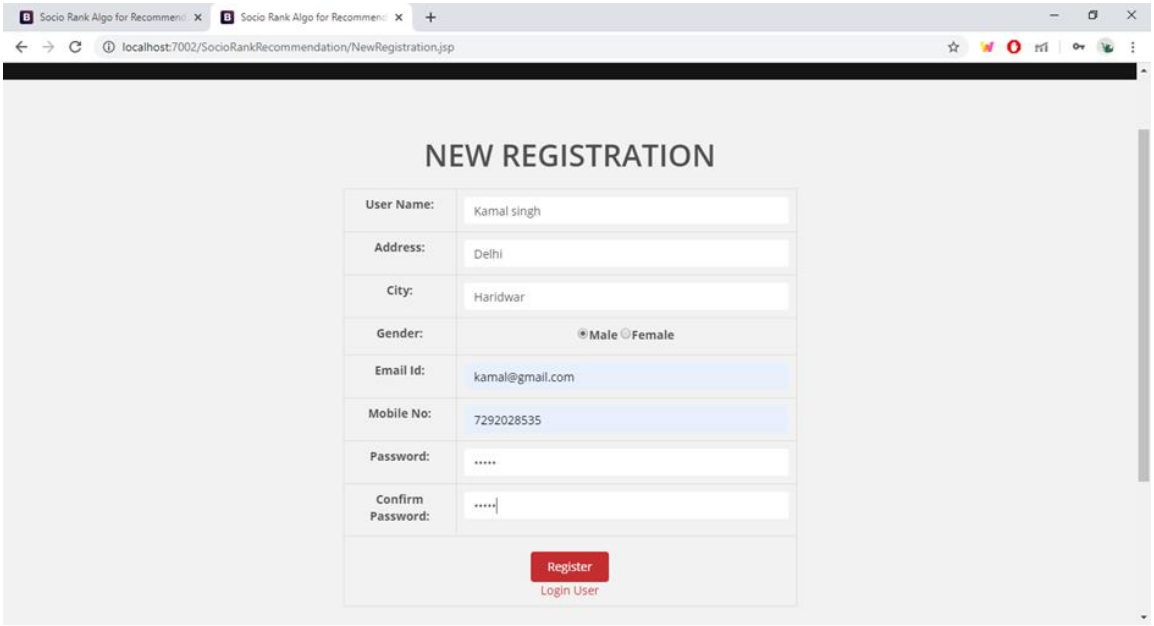
Chapter 8

OUTPUT WITH SCREENSHORTS

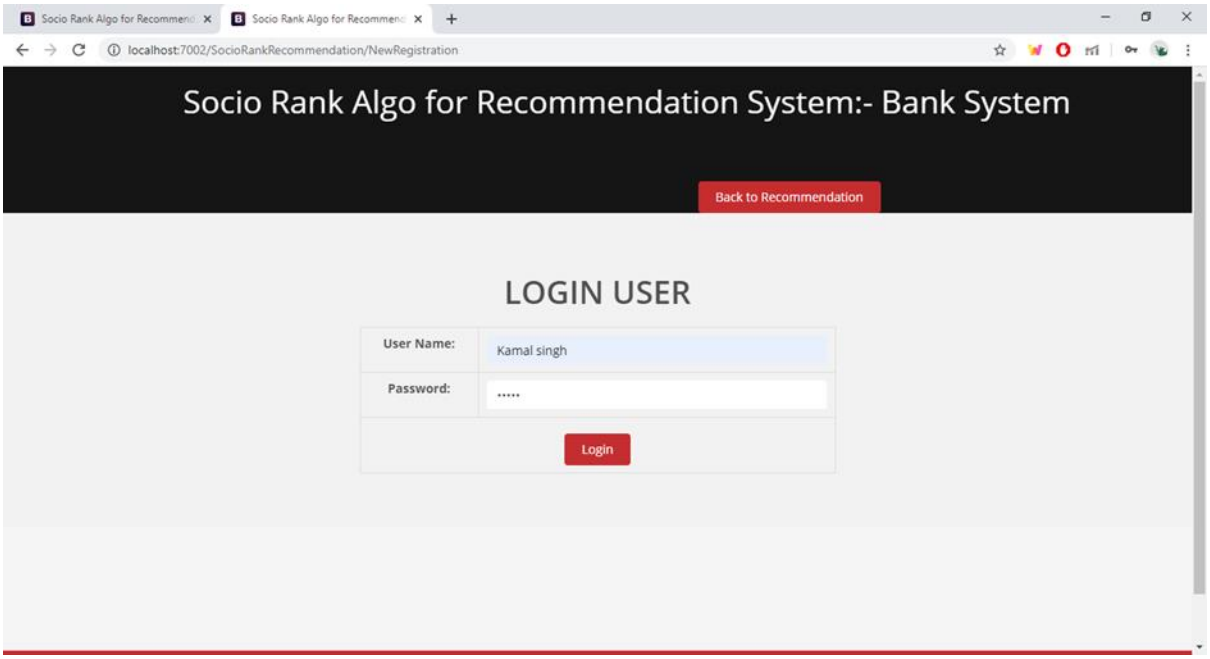
Login Page



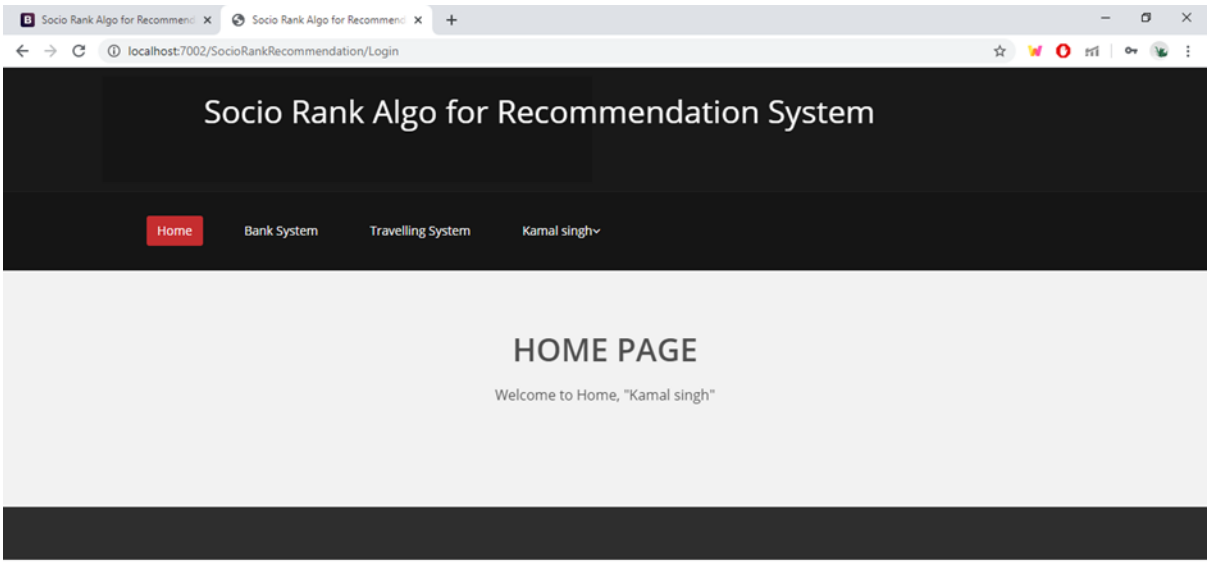
New Registration Page



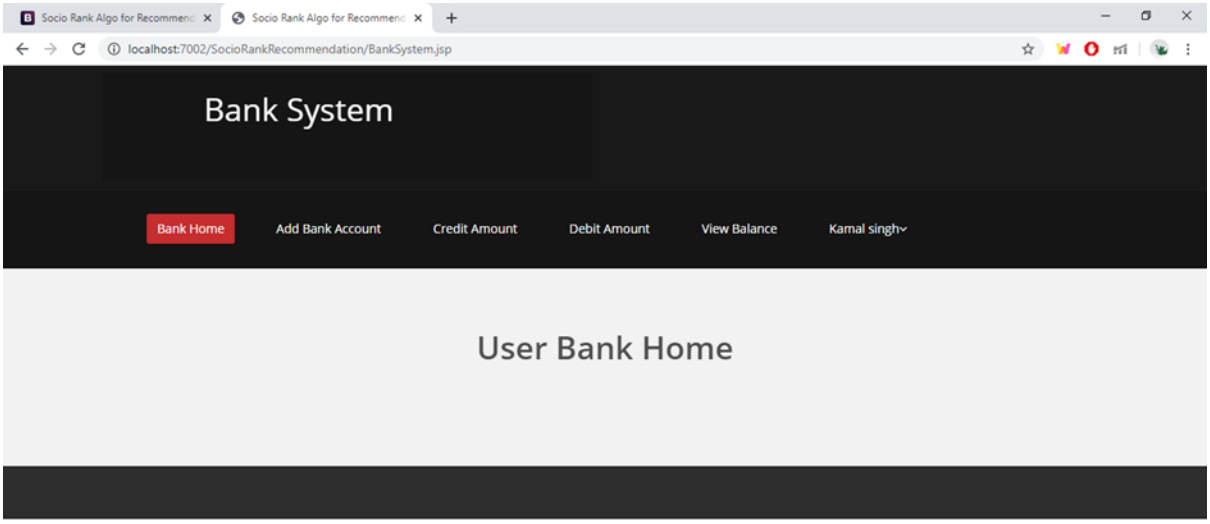
User Login Page



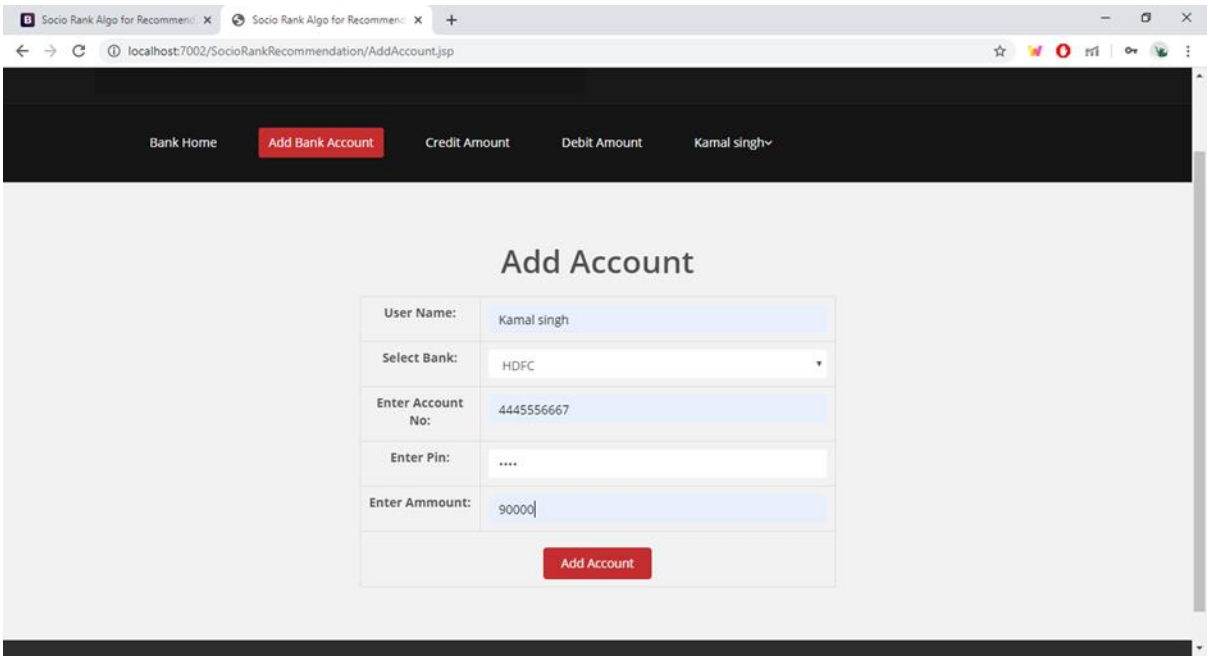
Home Page



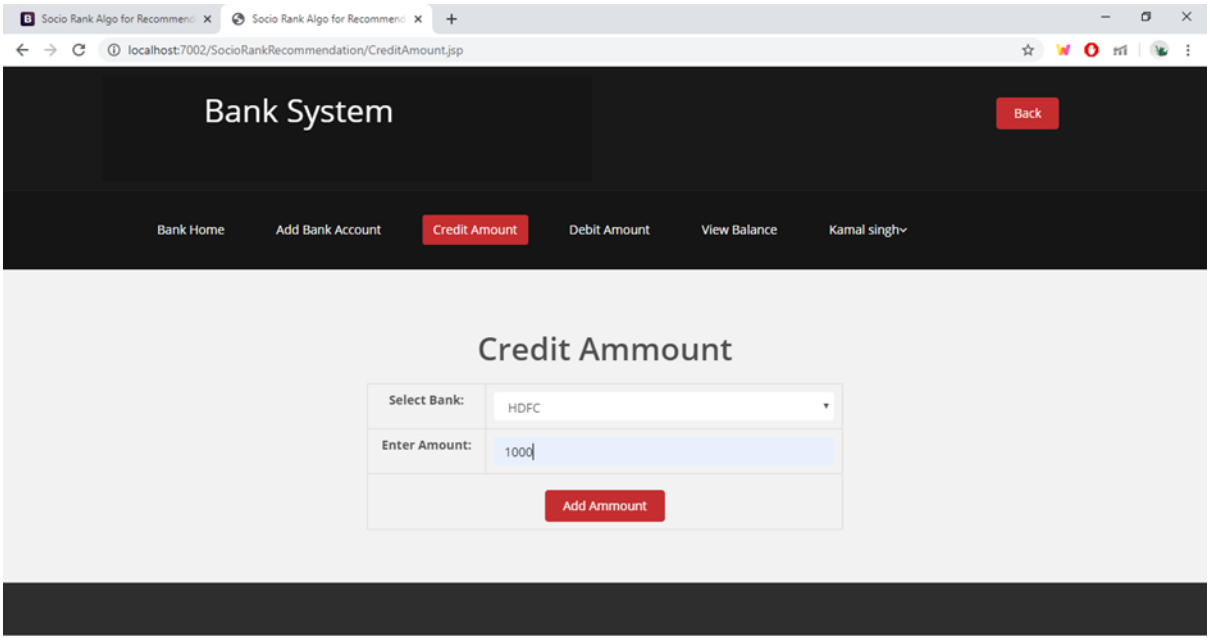
Banking System Home Page



Add Bank Account Page

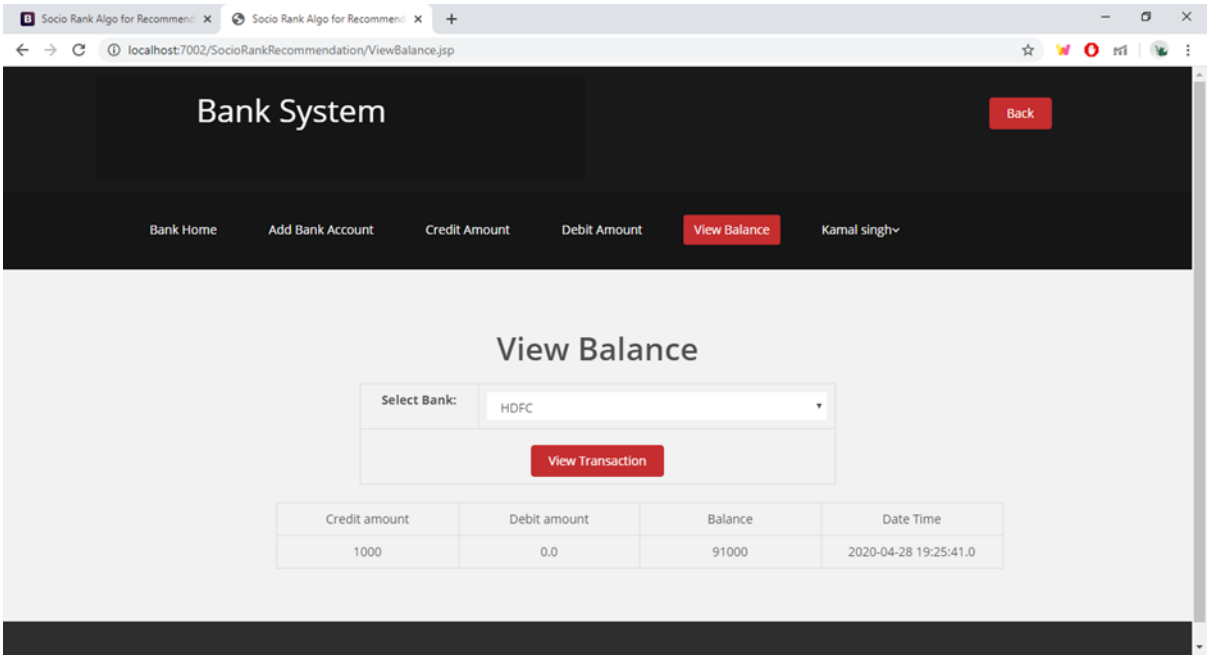


Credit Amount Page

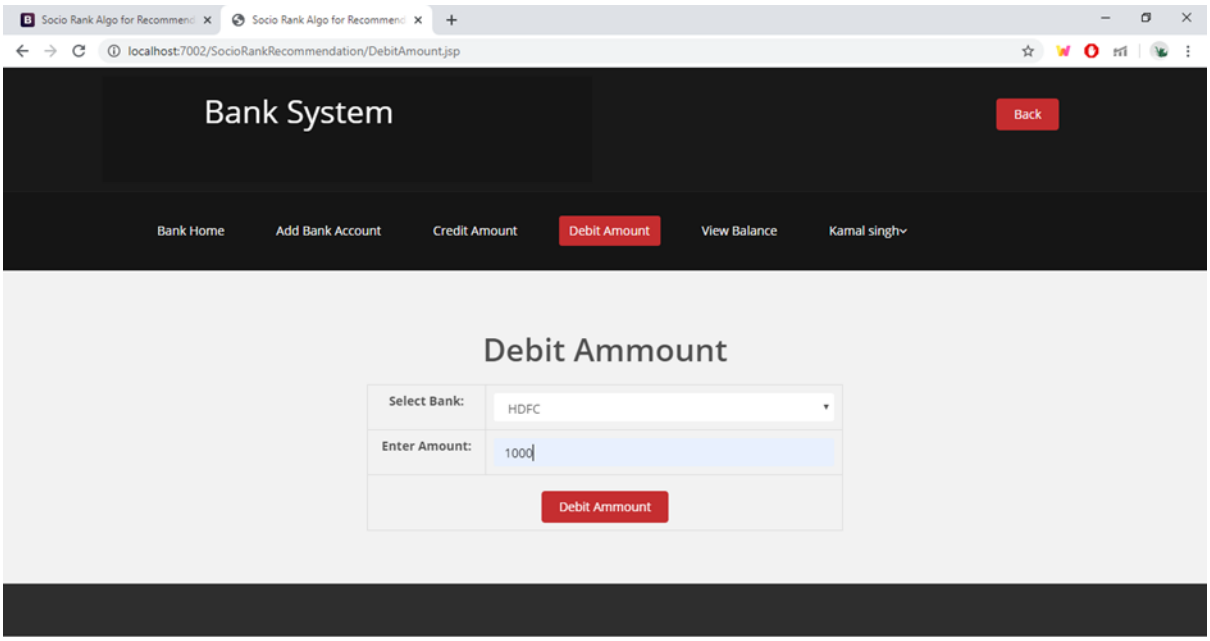


Amount Credited Successfully Dilaog Box

View Balance Page after Crediting Amount

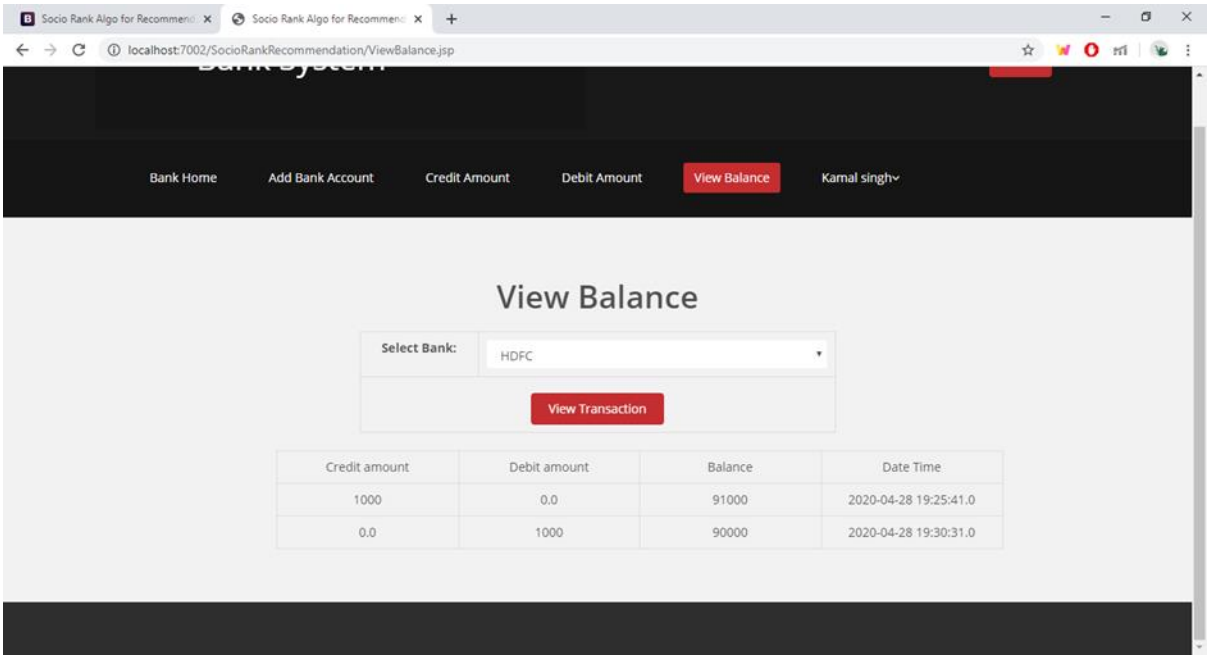


Debit Amount Page

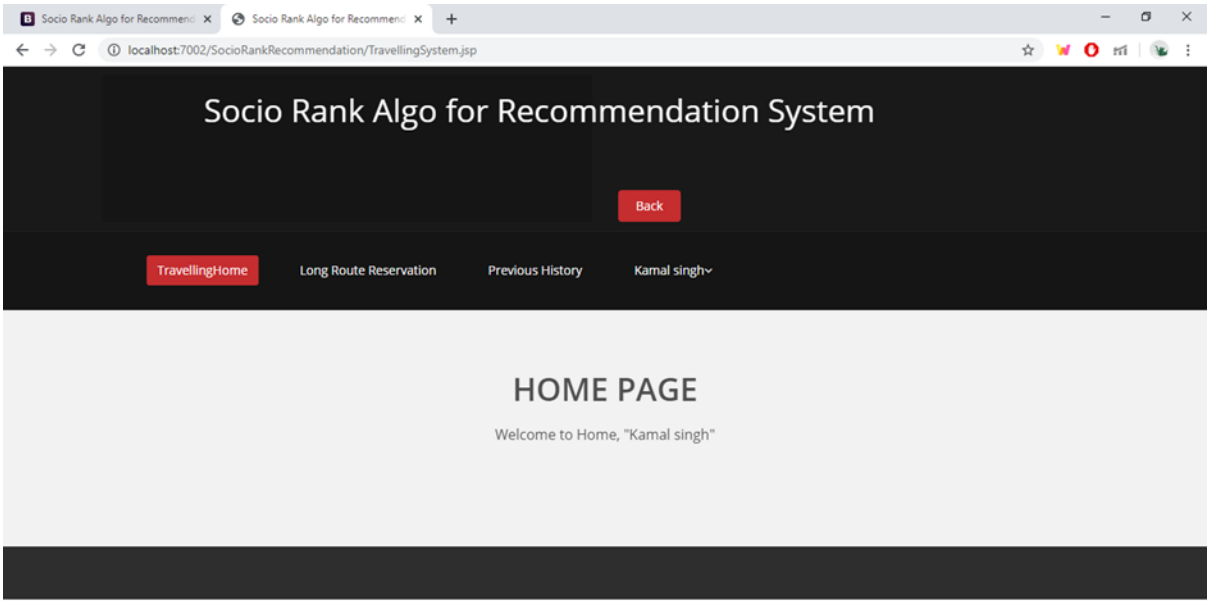


Amount Debited Successfully Dialog Box

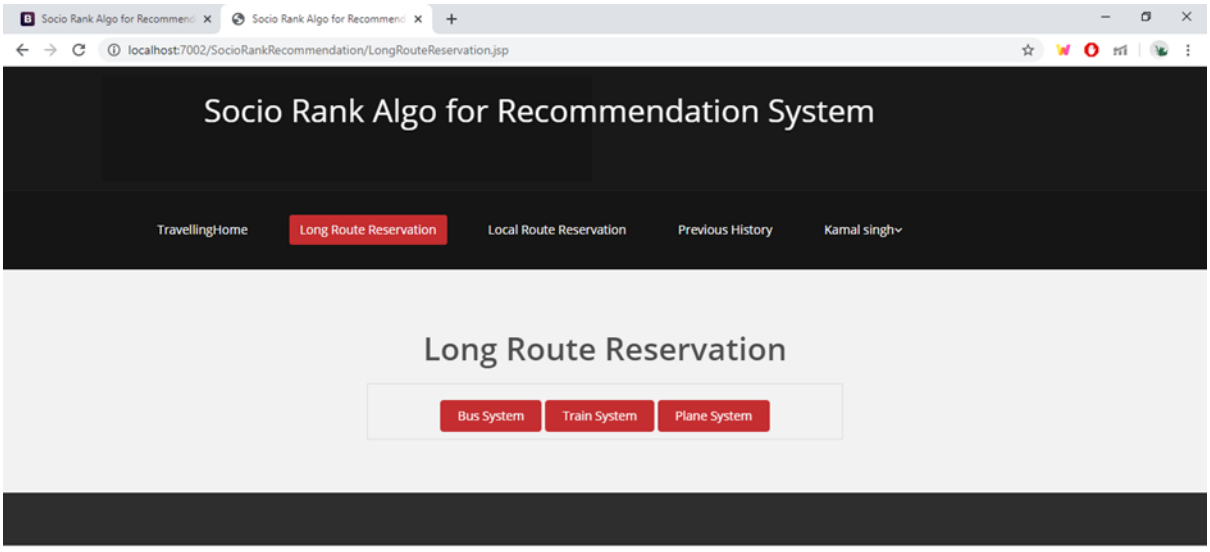
View Balance Page after Debiting Amount



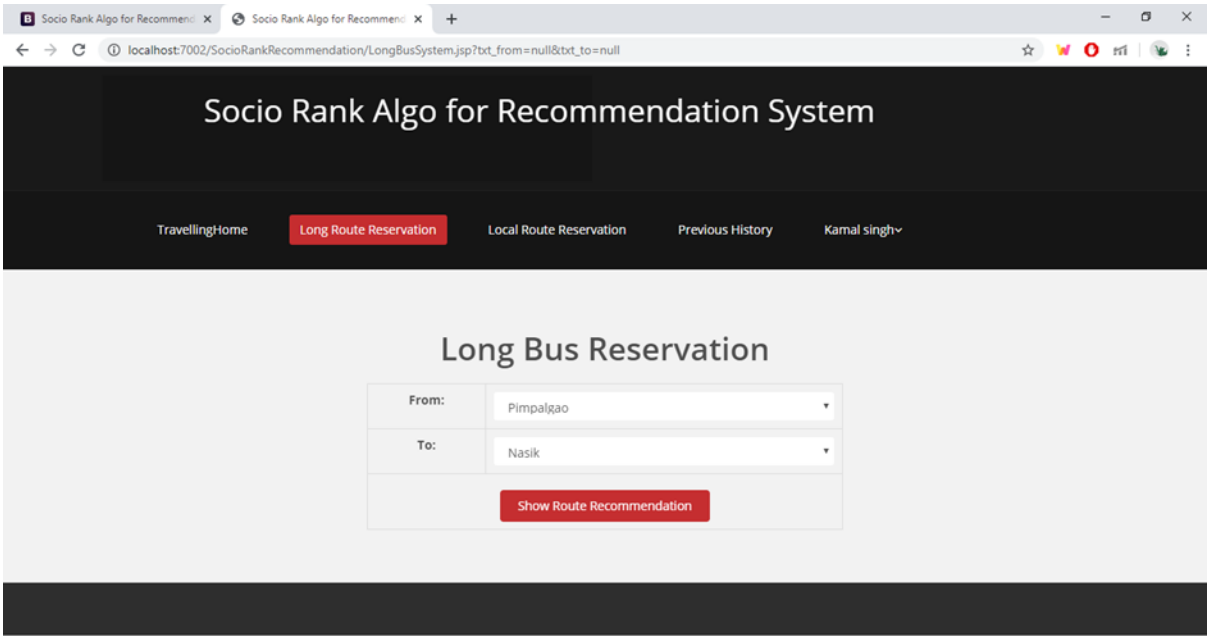
Travelling System Home Page



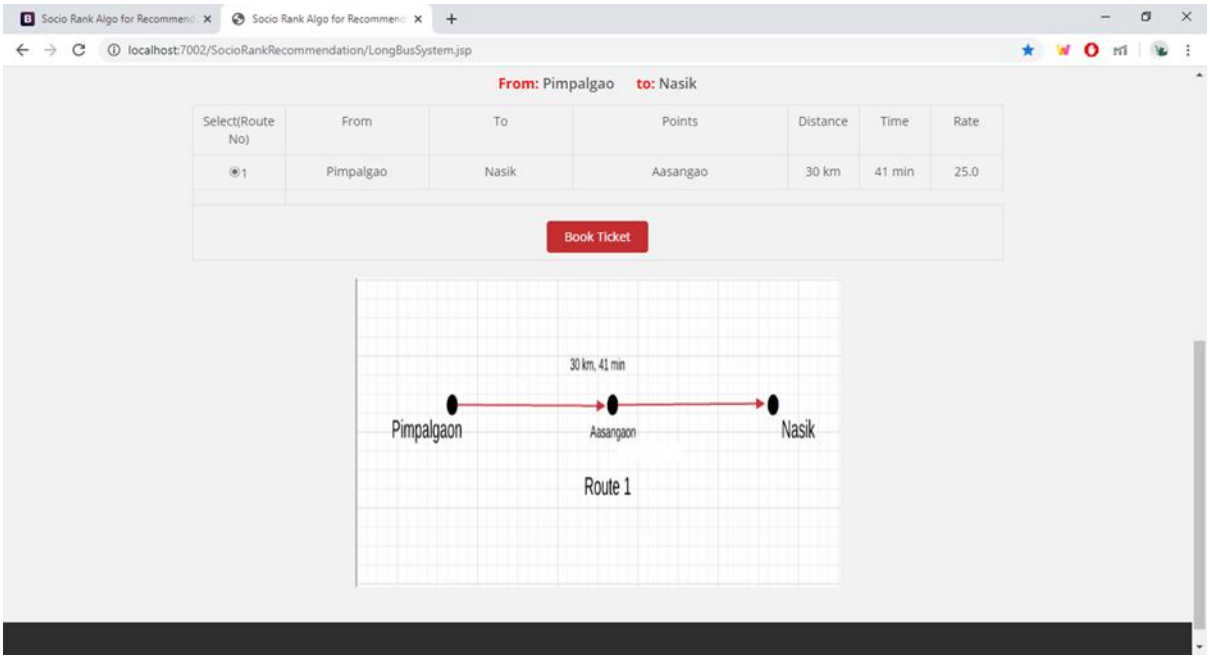
Long Route Reservation



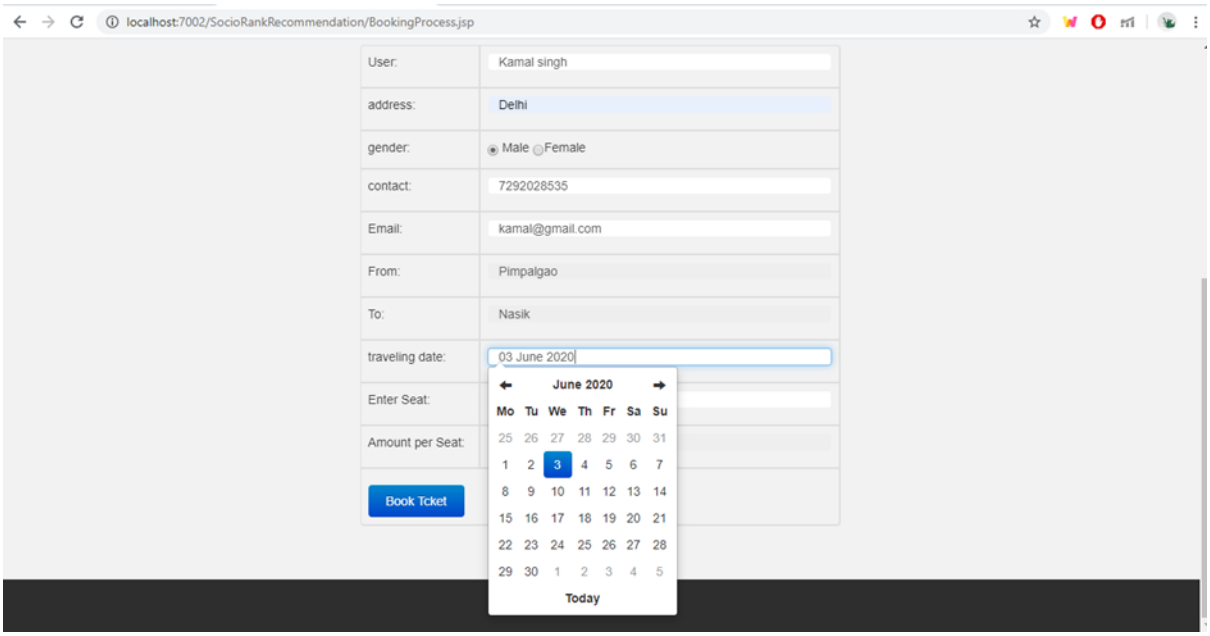
Bus Ticket Reservation



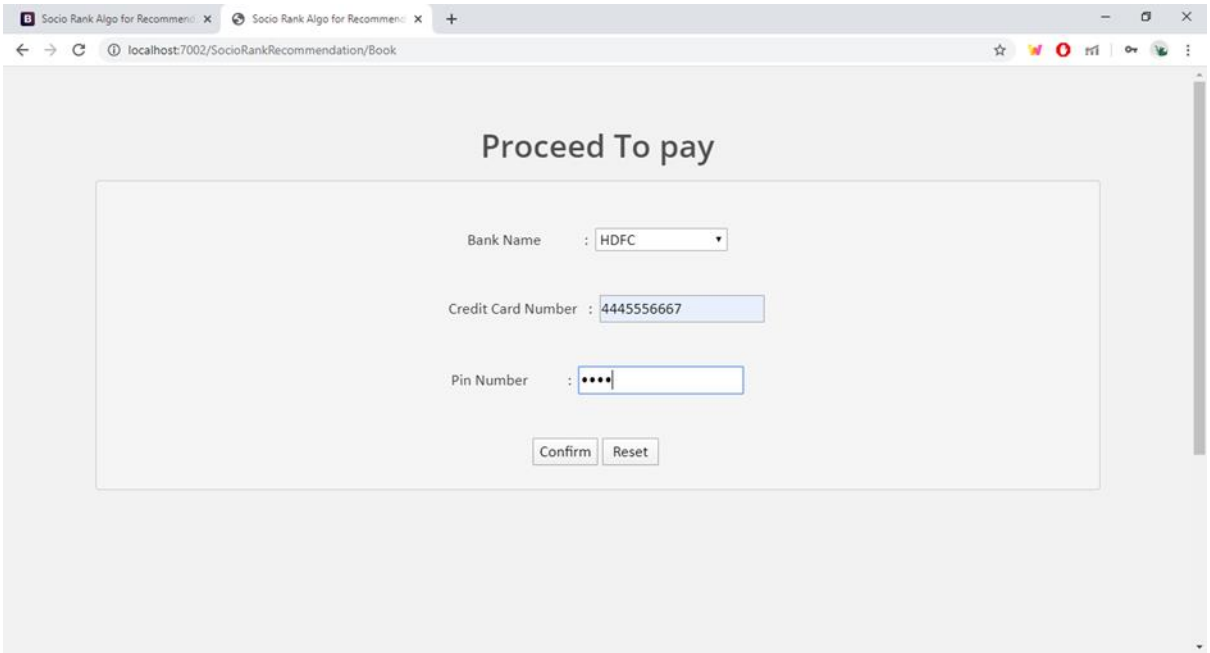
Best Bus Route Recommendation using Social Ranking



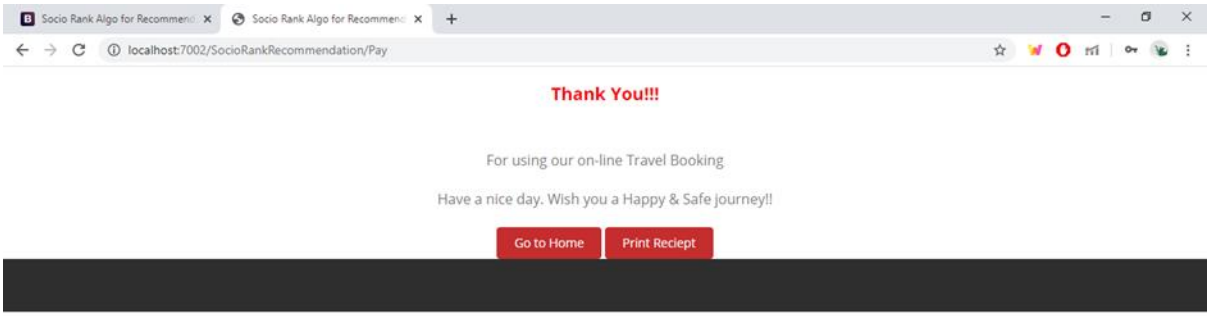
Bus Reservation Booking Page



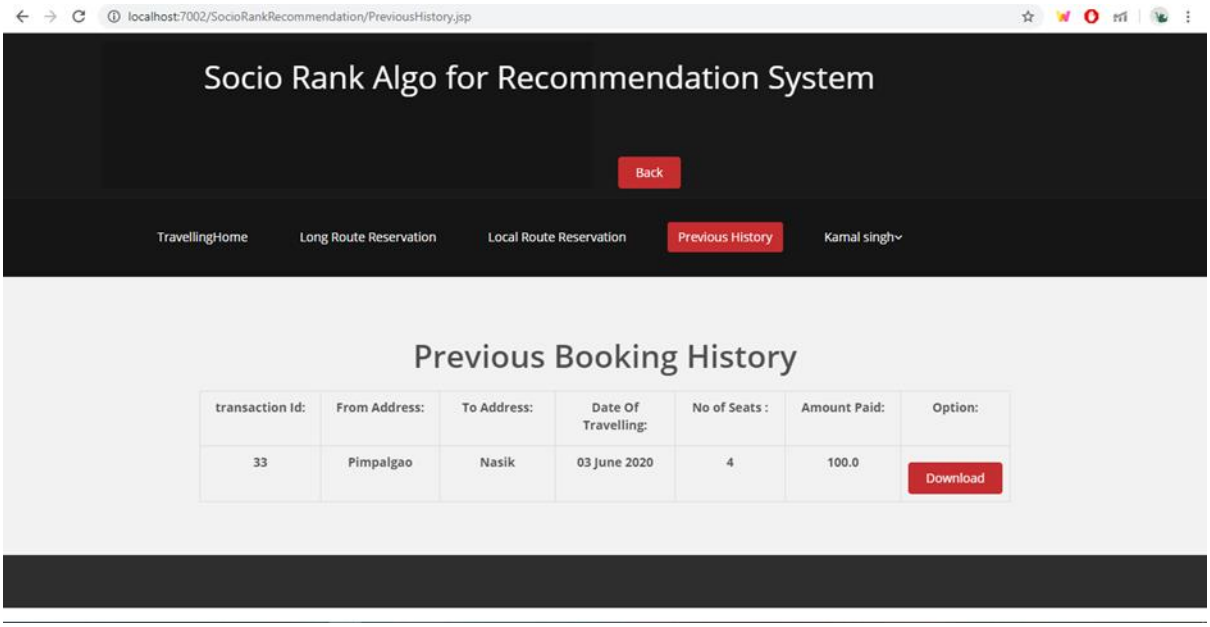
Payment Page



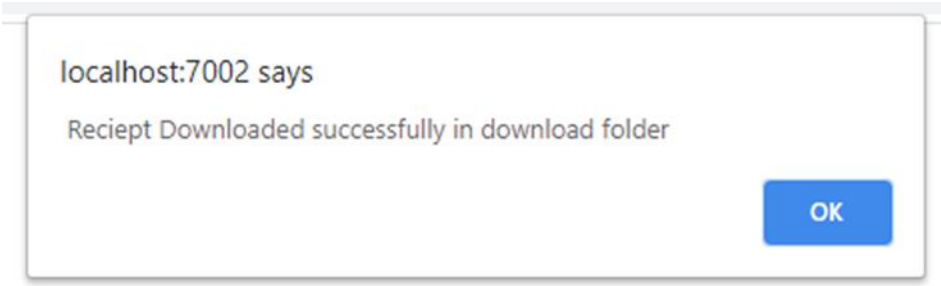
Payment Confirmation Page



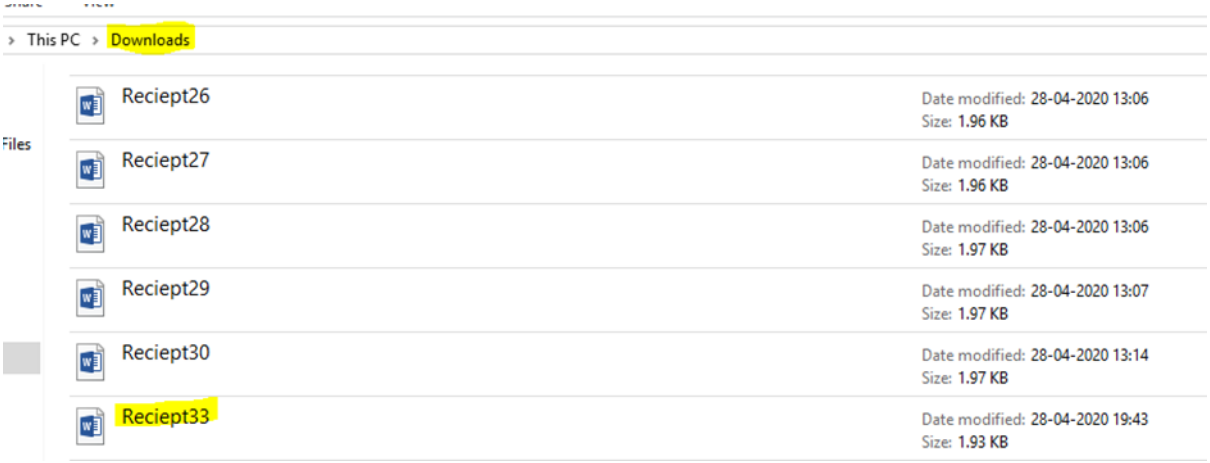
Receipt Download from Previous History Page



Receipt Download Confirmation in Download System Dialog Page



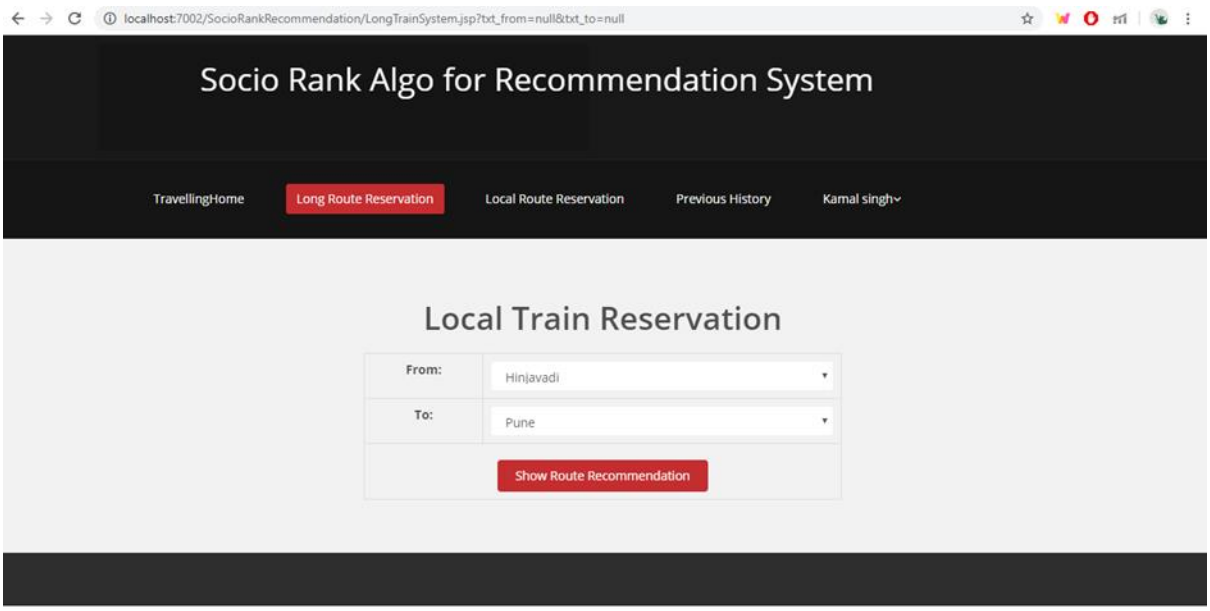
Downloads Folder Showing Receipt Downloaded after Successful Payment



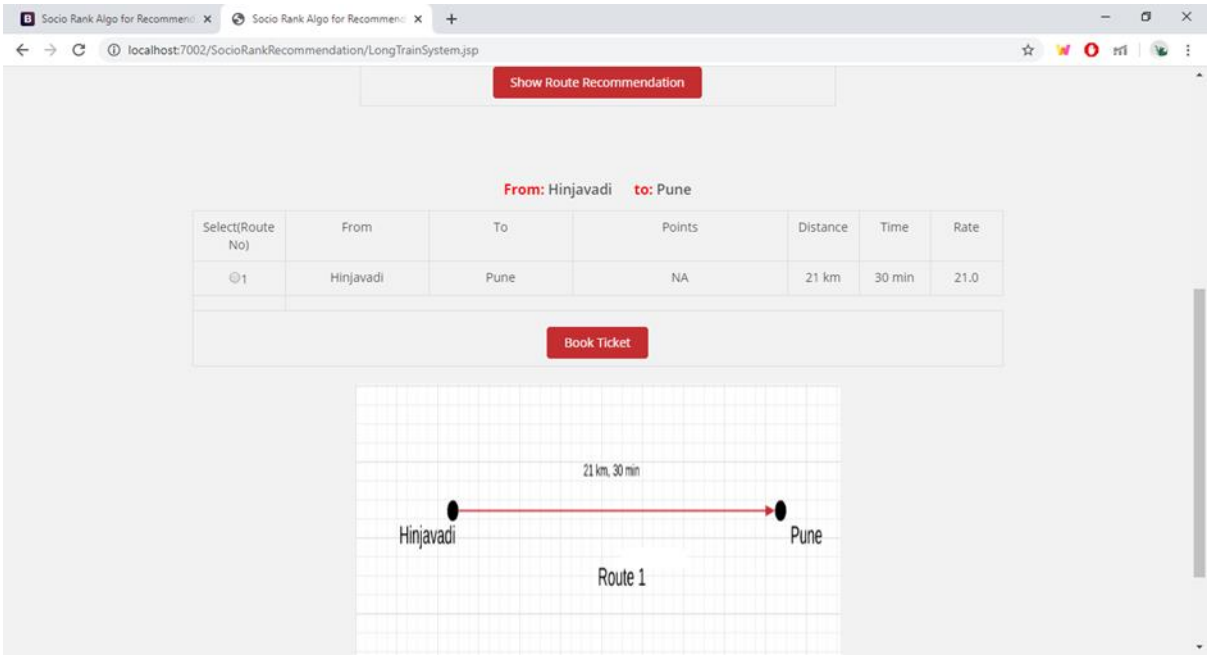
Receipt PDF Showing Booking details

Parameter	Values
Transaction Id	33
name	Kamal singh
Address	Delhi
Gender	Male
Contact	7292028535
Email Id	kamal@gmail.com
From	Pimpalgao
To	Nasik
Date of Travelling	03 June 2020
Seat	4
Amount	100.0
Status	Paid

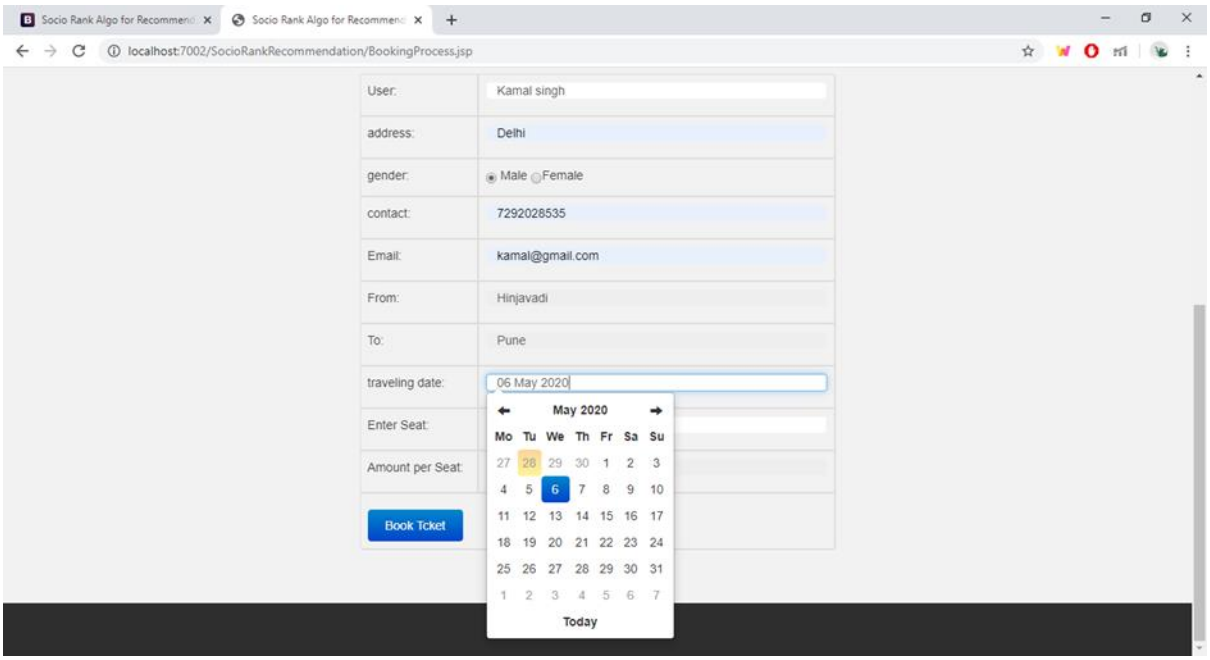
Train Ticket Reservation

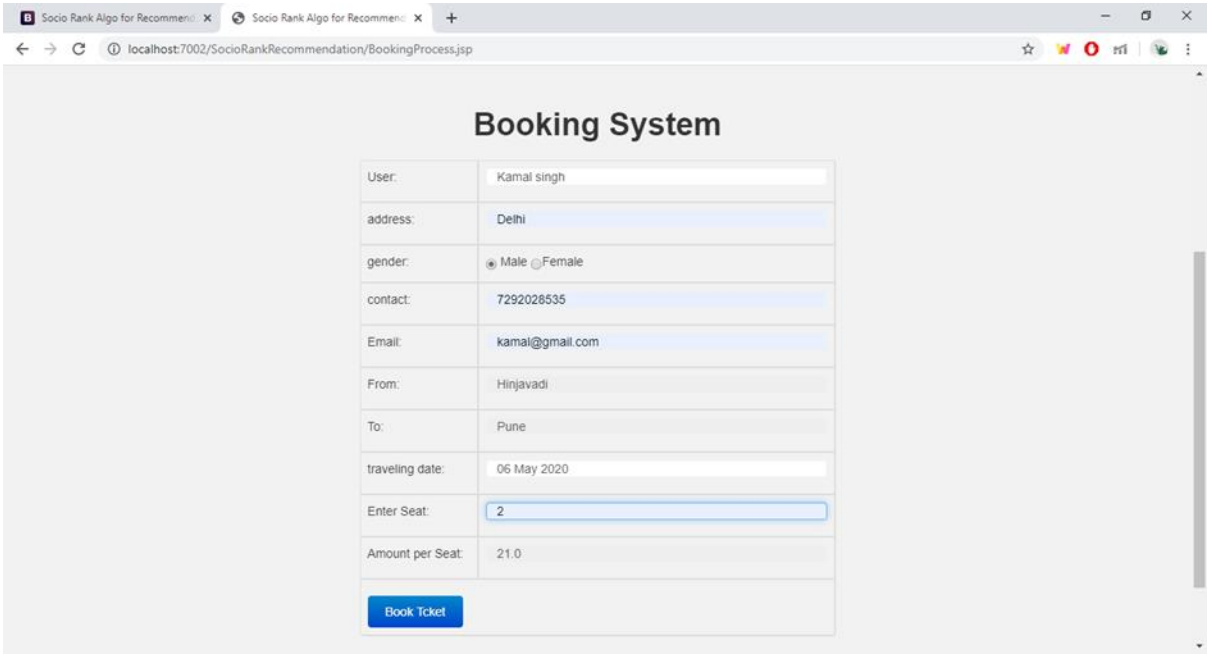


Best Train Route Recommendation using Social Ranking

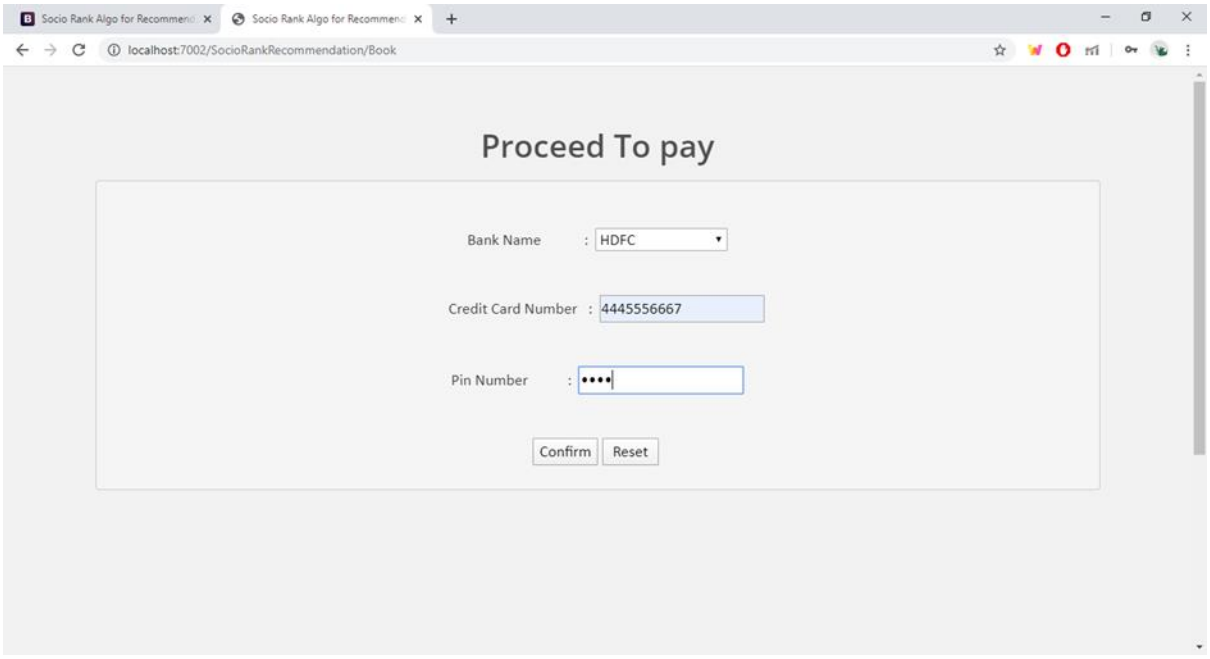


Train Reservation Booking Page

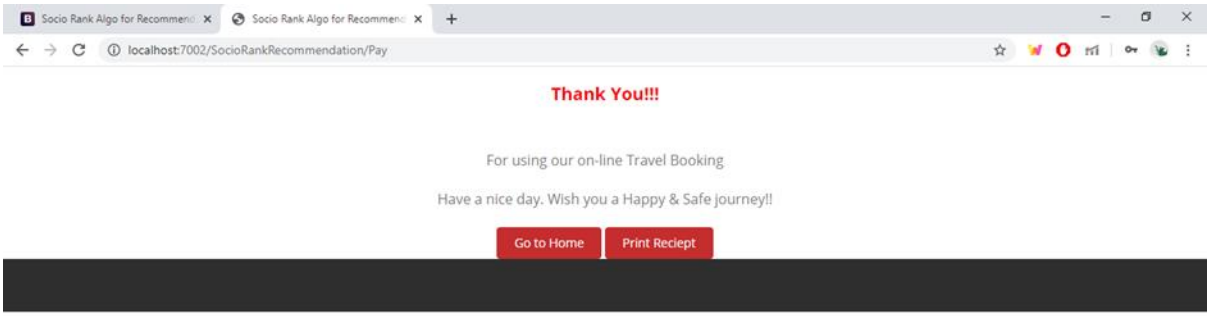




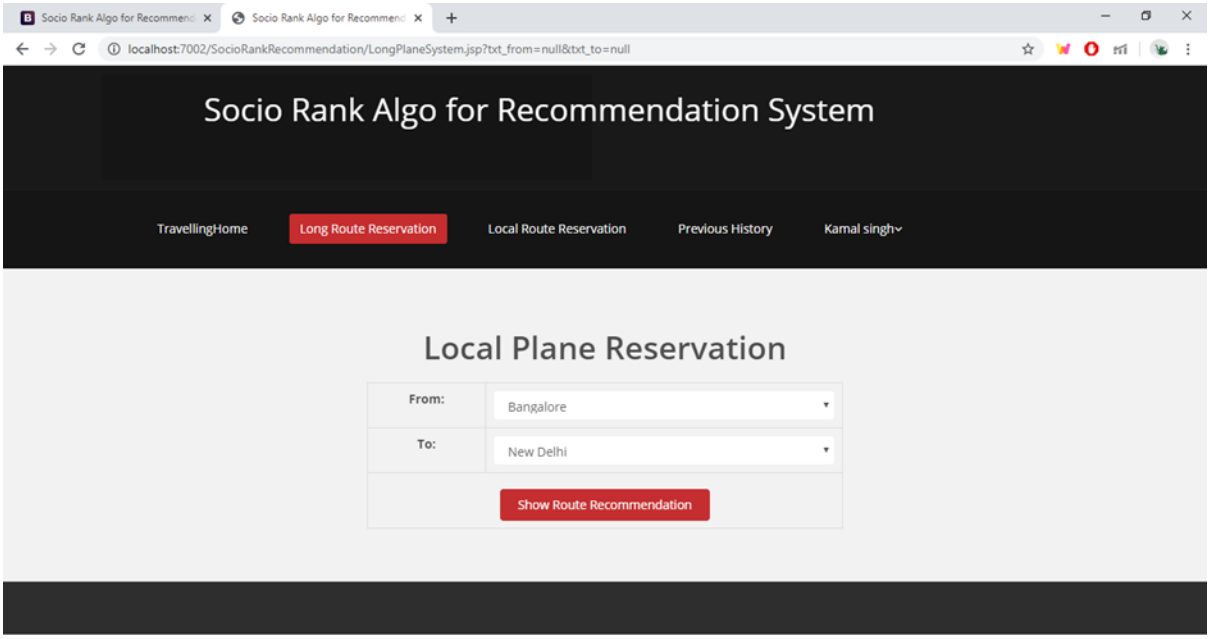
Payment Page



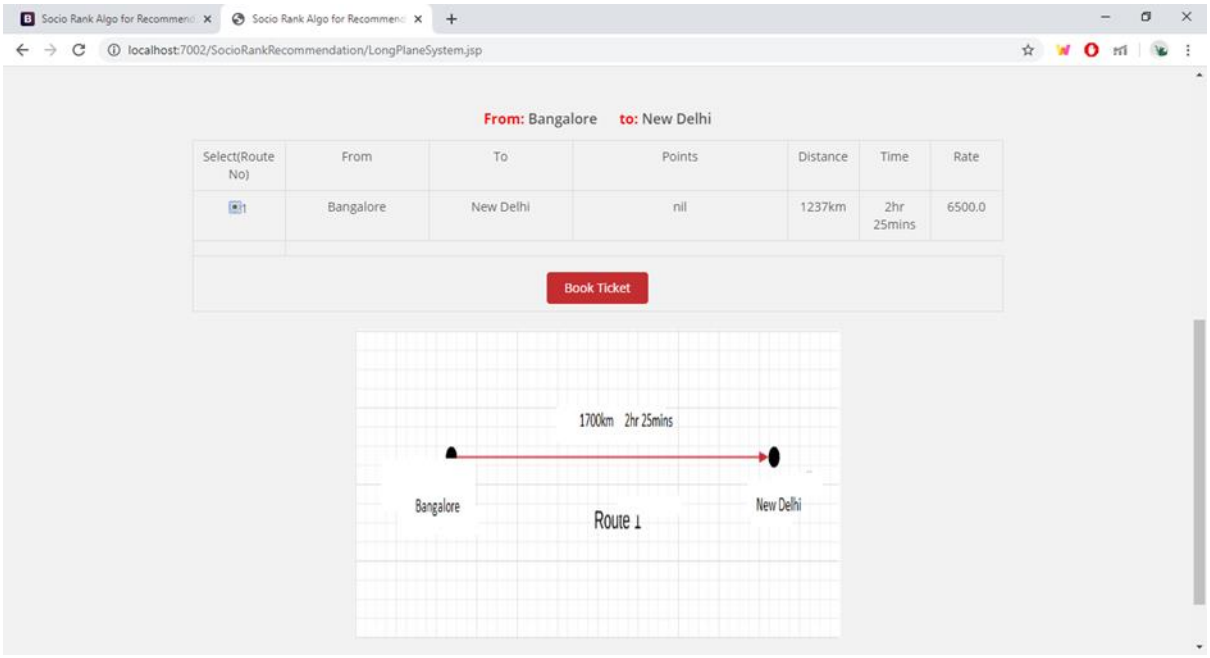
Payment Confirmation Page



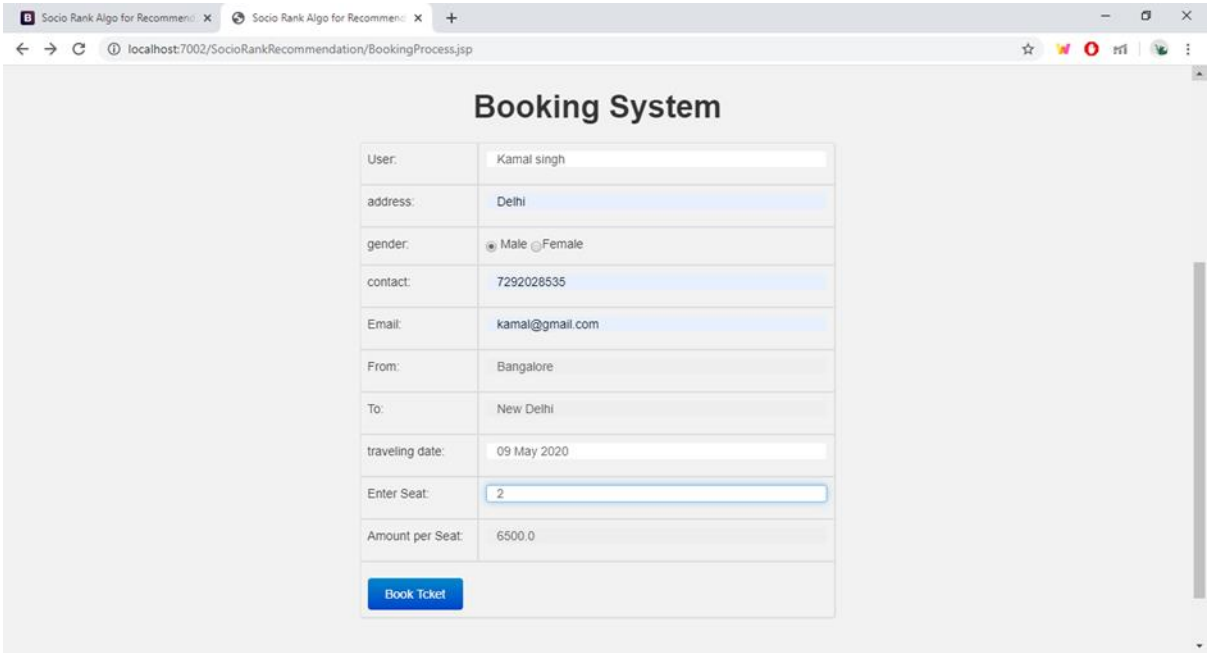
Plane Ticket Reservation



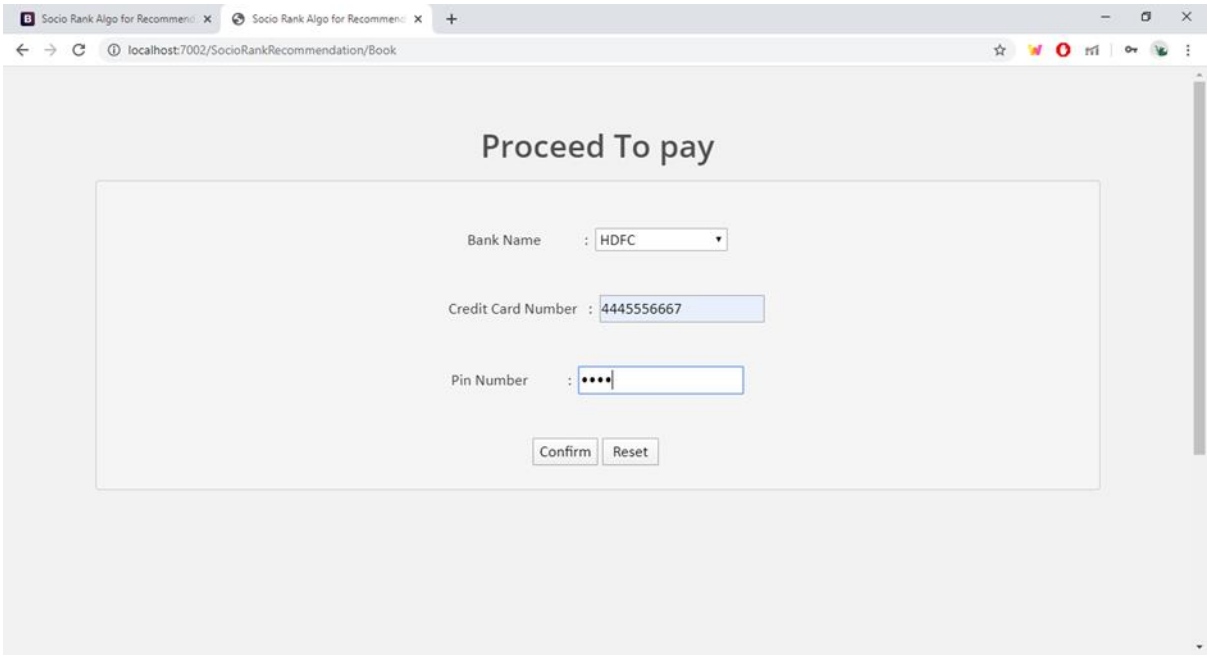
Best Plane Route Recommendation using Social Ranking



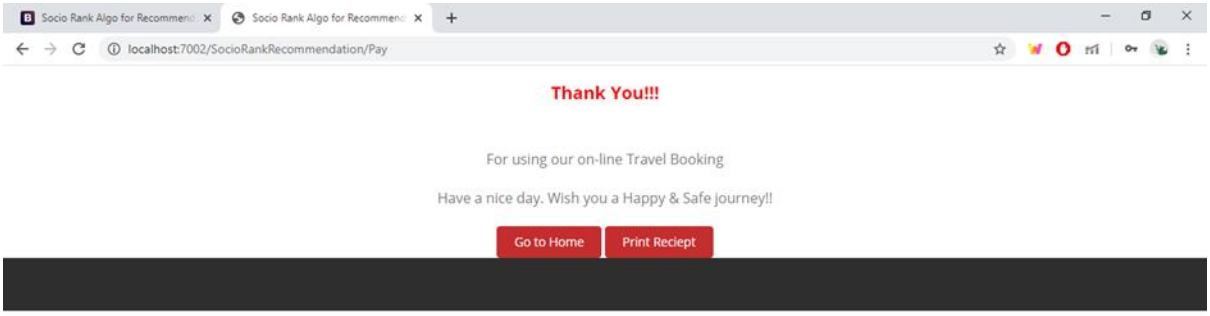
Plane Reservation Booking Page



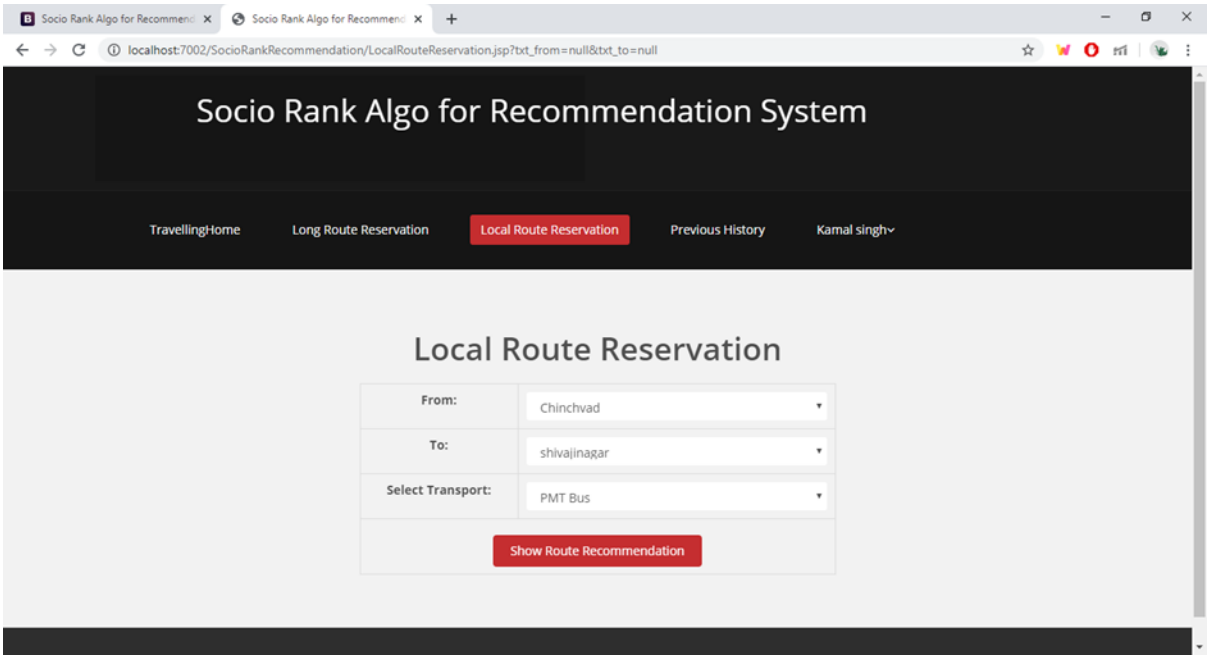
Payment Page



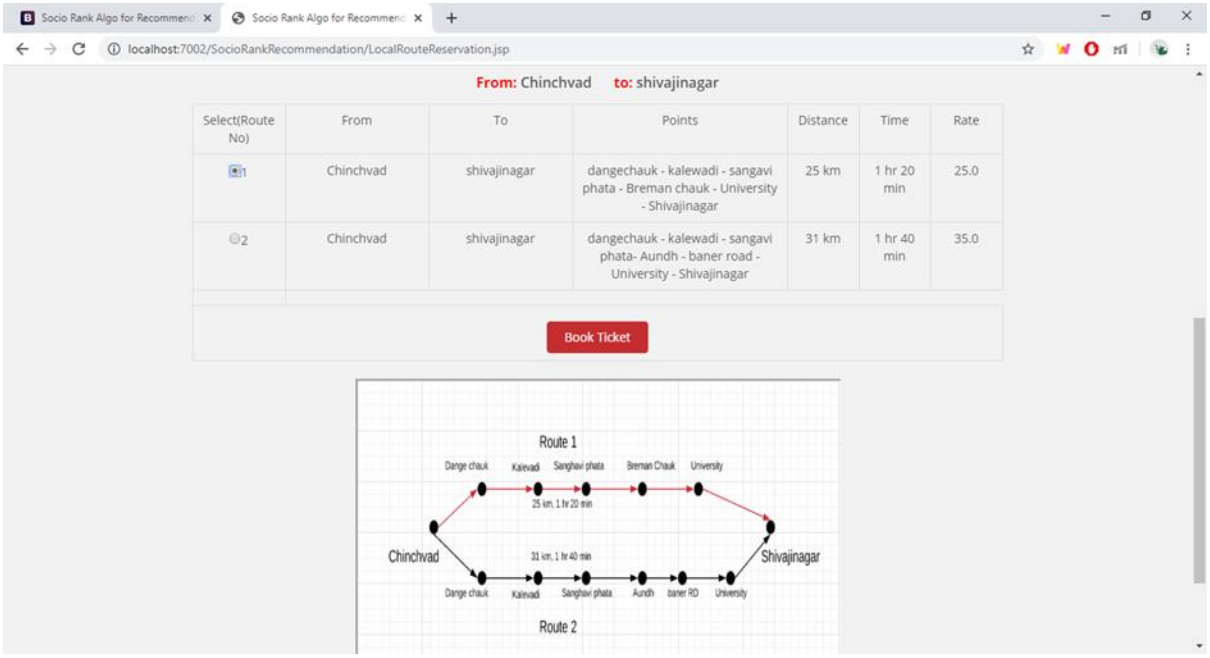
Payment Confirmation Page



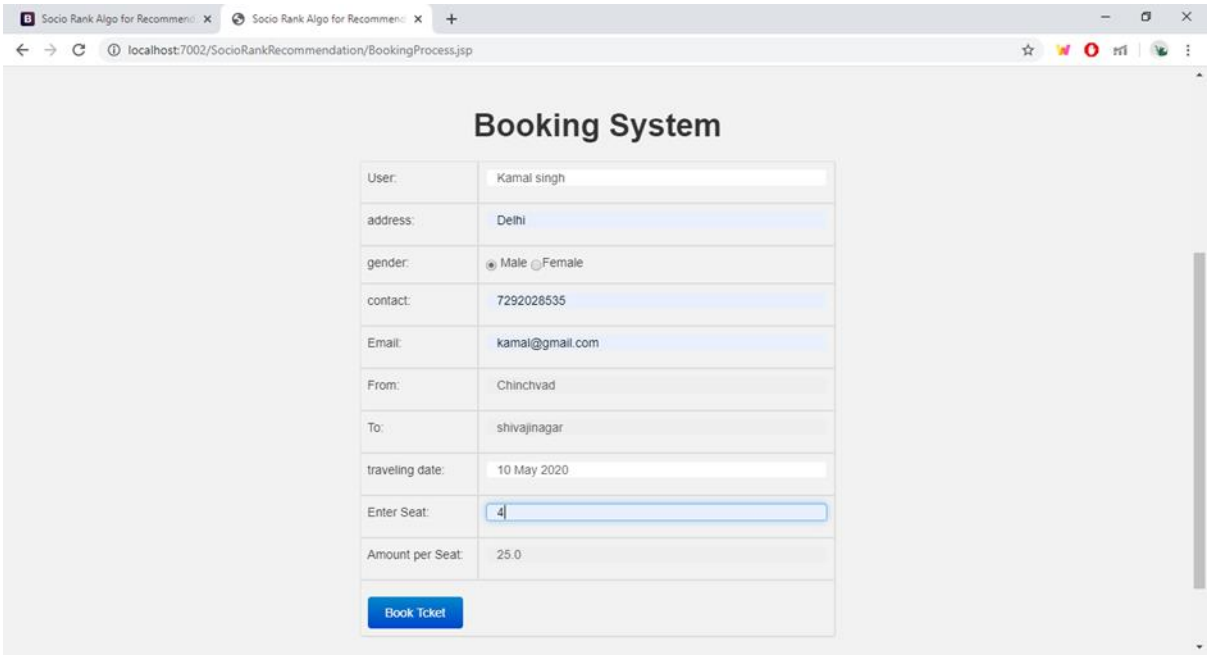
Local Bus Route Ticket Reservation



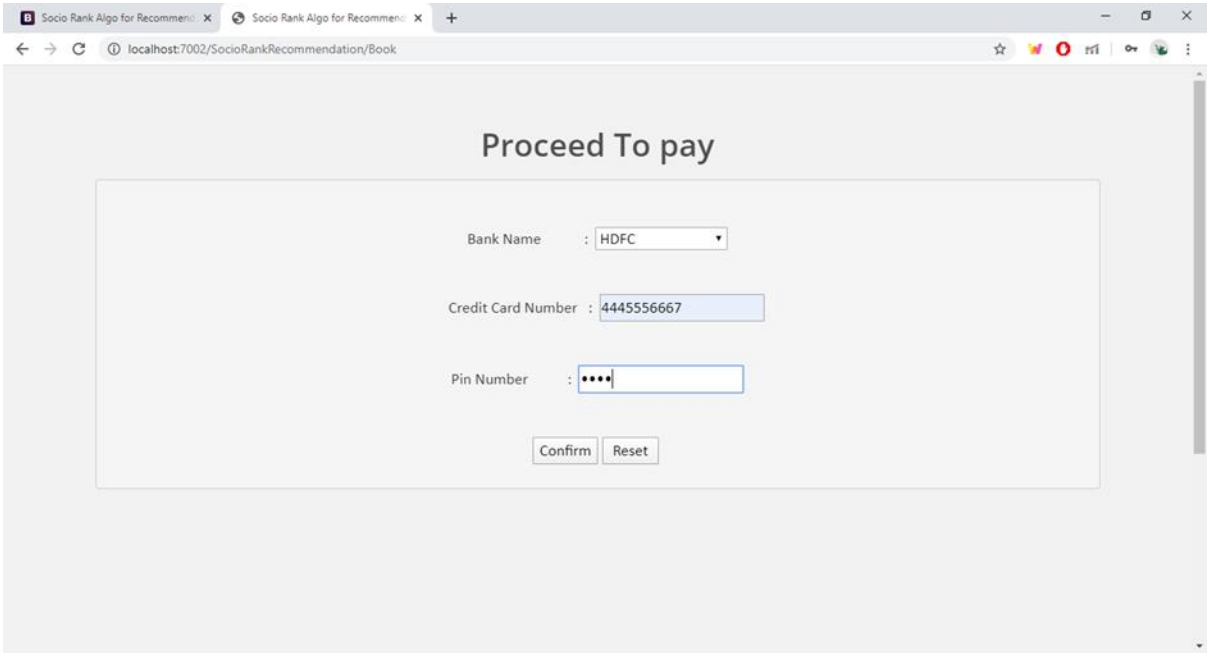
Best Local Bus Route Recommendation using Social Ranking



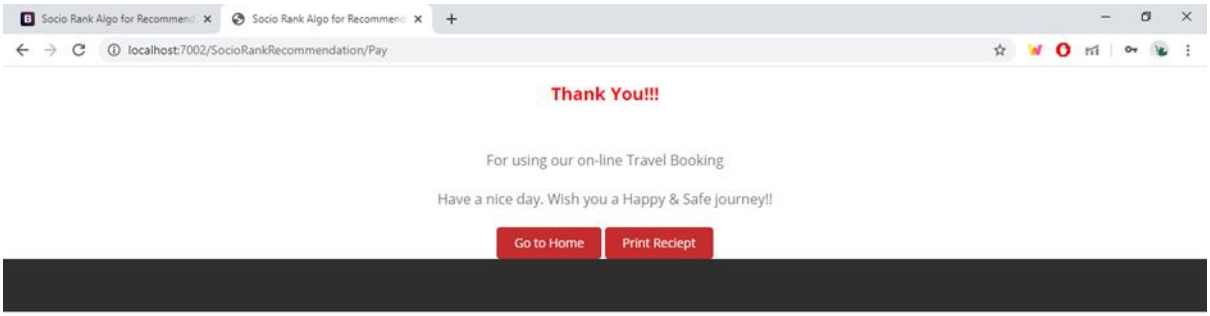
Local Bus Reservation Booking Page



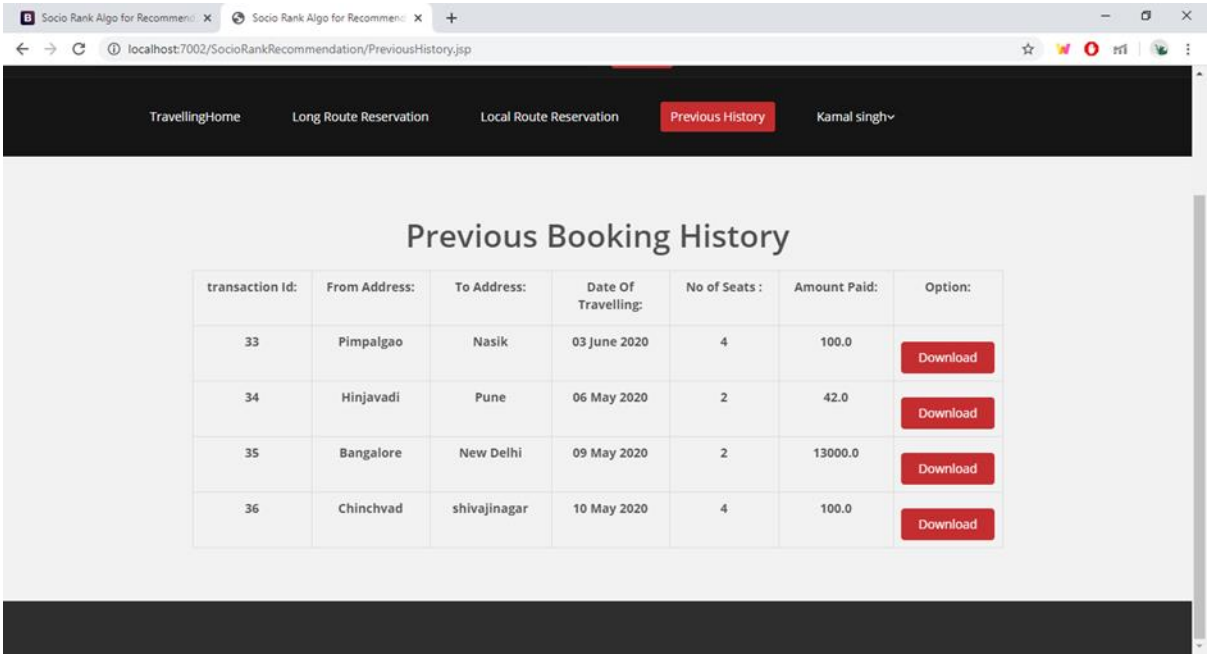
Payment Page



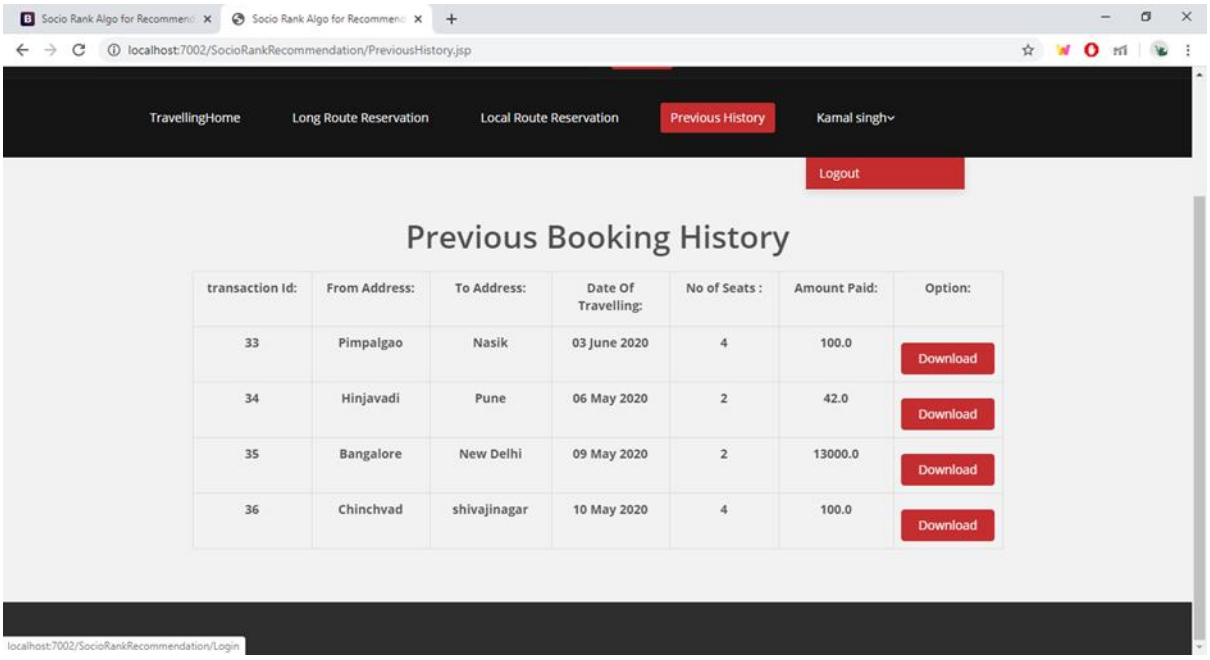
Payment Confirmation Page



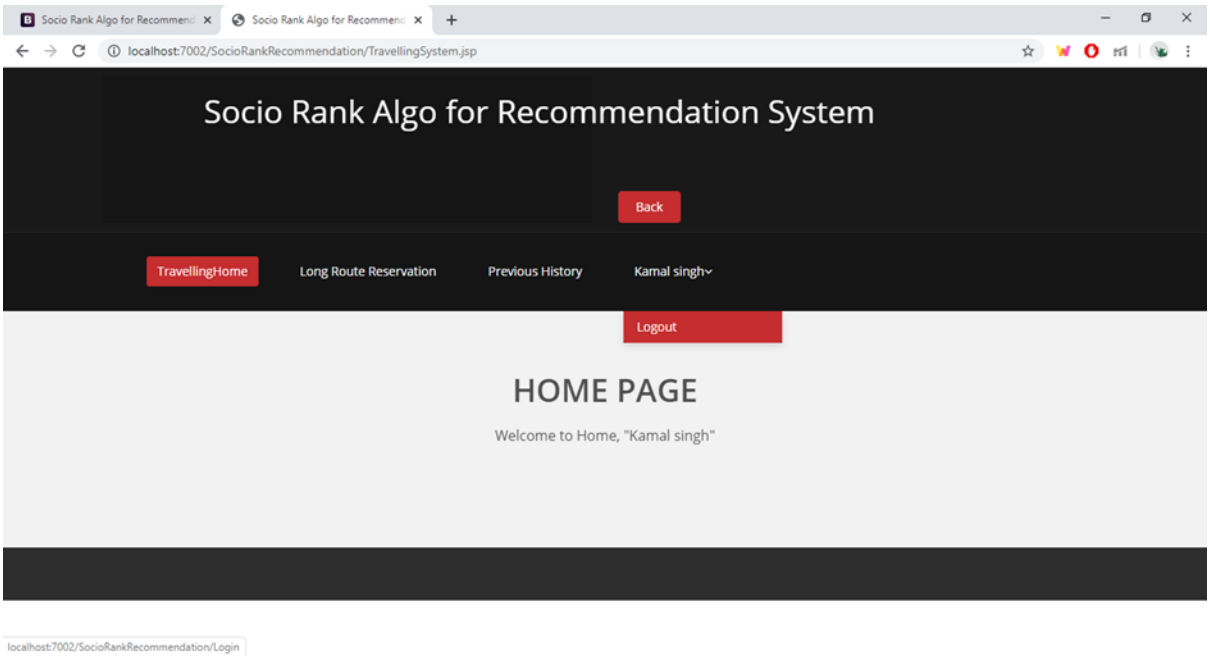
Previous History Page Which Shows Booking History and allows to Download Ticket



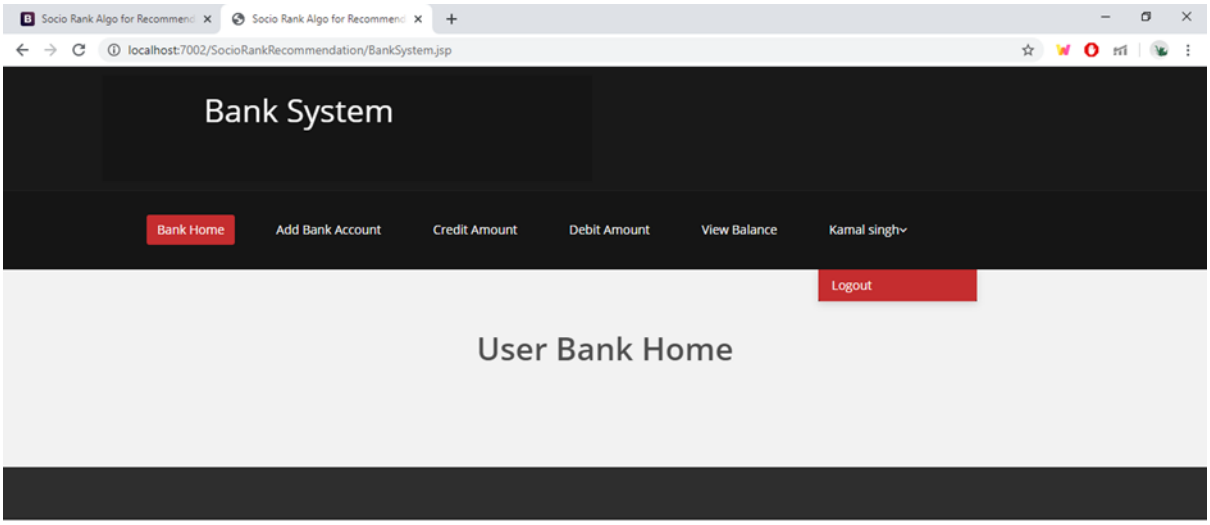
Logout Option From Previous Booking History Page



Logout Option From Travelling System Home Page

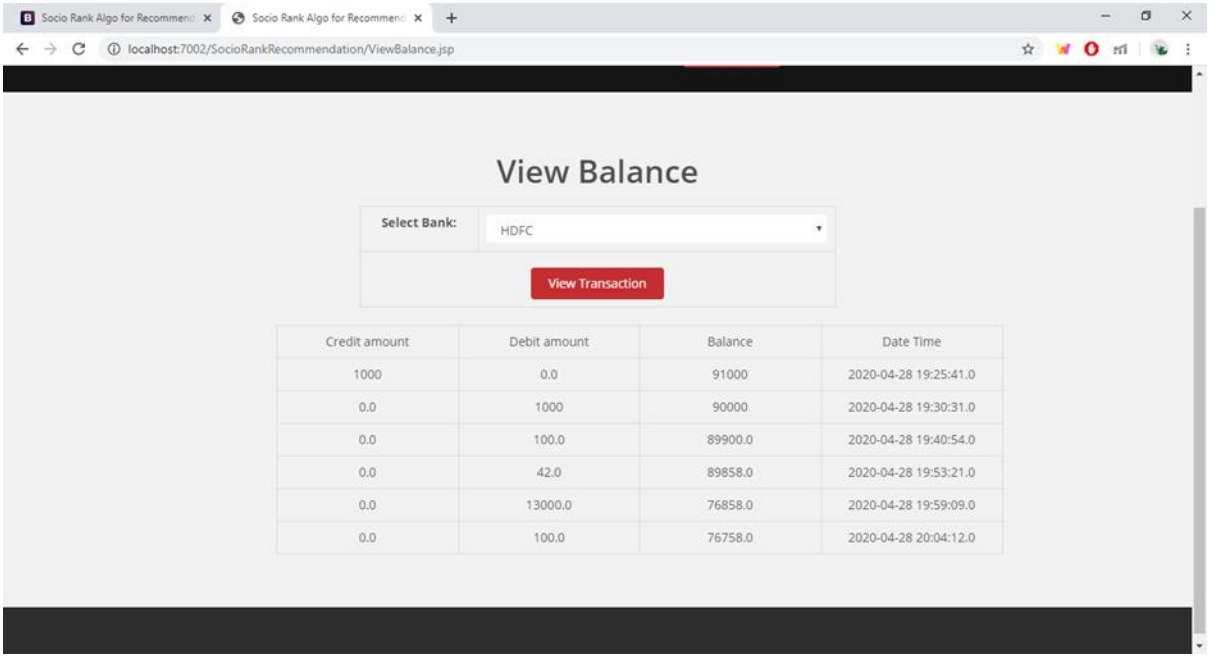


Logout Option From Banking System Home Page

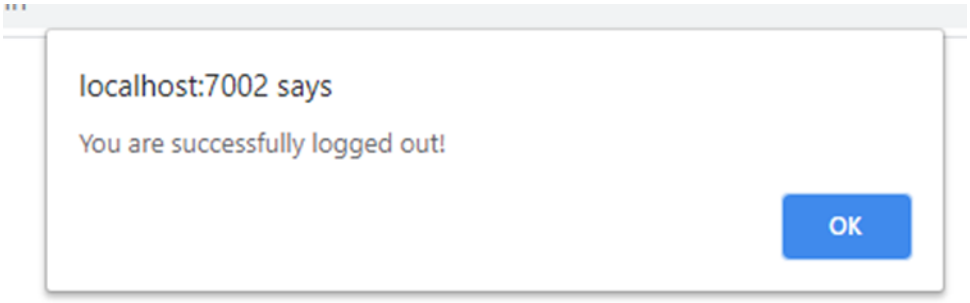


localhost:7002/SocioRankRecommendation/Login

View Balance Page that shows Transaction Details



Logged Out Successfully Dialog Box



Chapter 9

CONCLUSION AND SUMMARY

Faceted search is a technique of accessing a large collection of information that is represented by a faceted taxonomy. It enables users to select facets and facet terms to refine the search results in an iterative way. Extensive research has been done in this domain during the last decade. This approach summarized the published literatures, and proposed a faceted taxonomy of research work on faceted search. On the foundation of the taxonomy, three types of facet models, which are based on the set theory, system's key technologies in the framework, namely facet terms extraction.

Tourism is recognized as a global industry that is highly growing at a high rate like other industries. There are many different functionalities included in my project Social Ranking Implementation in Travelling System.

This Web Base Application is one stop solution for travelers for all their travelling need. It not only provides the best deals based on user's travel history and preferences but also has banking system functionality that also saves the user from login into any third party payment system and is safe and secure. The application can be fully customized using different APIs and be made even more user friendly.

In a time when financial resources are limited and competition for tourist, the travel services sector is being forced to innovate at a startling rate. This app hence takes into consideration the user travel destination and provides best deals at most discounted rate, least travelling time and at better price rates. It provide a recommendation of travel either inter states (through bus, train, and plane) or within a state (through bus) allowing user to choose within more than one option the best suited as per their needs. It shows the price, time and distance for all the routes which allows user to make better choices while booking.

Finally this app provides user to manage, control and handle their ticket booking and payment activities more effectively and efficiently.

Chapter 10

FUTURE SCOPE

For the future environment system can focus on personalize search on user feedback sessions as well as recommendation based on user point of interest with database security is the interesting part of system.

The application has such a vast future scope as it can be extended and implemented in Hotel Booking, International Tours and Travel Booking, Best visiting places in a city, food booking, its local routes recommendation can be implemented in cab booking, auto booking, car pool booking, bike booking etc.

With these enhancement it will help the user both in day-to-day commute and while travel within country or to other countries and remove the middle agent as user can chose best deal from the recommendation provided by the app itself.

This project has a wider scope in future and can be helpful to many. Since this project is made in Java and MySQL which makes this project a wonder and innovative and helps in clearing concepts of Java and MySQL. This project also clears concept of how to use both Java and MySQL together.

This also shows how Java and MySQL connected in an application. This project can be needed in future for tours and travel and also this project can be helpful for user that deal with their transaction online on a regular basis.

Chapter 11

REFERENCES

[1] Zhicheng Dou and Zhengbao Jiang, proposed Automatically Mining Facets for Queries from Their Search Results in IEEE transaction and data engineering Vol 28, No. 2 Feb 2017

[2] Roy, S.B., et al., DynaCet: Building Dynamic Faceted Search Systems over Databases, in 2009 IEEE 25th International Conference on Data Engineering(ICDE), Vols 1-3. 2009. p. 1463-1466, 1768.

[3] Zhao, B., et al., TEXplorer: keyword-based object search and exploration in multidimensional text databases, in Proceedings of the 20th ACM Conference on Information and Knowledge Management (CIKM). 2011, ACM: Glasgow, Scotland, UK. p. 1709-1718.

[4] Dou, Z., et al., Finding dimensions for queries, in Proceedings of the 20th ACM Conference on Information and Knowledge Management (CIKM). 2011, ACM: Glasgow, Scotland, UK. p.1311-1320.

[5] Chen, J. and Li, Q., Concept Hierarchy Construction by Combining Spectral Clustering and Subsumption Estimation, in Web Information Systems – WISE 2006, K. Aberer, et al., Editors.2006, Springer Berlin / Heidelberg. p. 199-209.

[6] Eyal Oren, Renaud Delbru, and Stefan Decker proposed Extending faceted navigation for RDF data

[7] Anick, P.G. and Tipirneni, S., The paraphrase search assistant: terminological feedback for iterative information seeking, in Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. 1999, ACM: Berkeley, California, United States. p. 153-159.

[8] Ling, X., et al., Mining multi-faceted overviews of arbitrary topics in a text collection, in Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. 2008, ACM: Las Vegas, Nevada, USA. p. 497-505.

[9] Dakka, W. and Ipeirotis, P.G., Automatic extraction of useful facet hierarchies from text databases, in 2008 IEEE 24th International Conference on Data Engineering. 2008. p. 466-475,1631.

[10] LingPipe Home. Available from: http://www.alias-i.com/lingpipe/.

[11] Term Extraction Web Service - YDN. Available from:http://developer.yahoo.com/search/content/V1/termExtraction.html.

[12] Taxonomy Warehouse. Available from: http://www.taxonomywarehouse.com/.

[13] Shafqat, Wafa, and Yung-Cheol Byun. "A Recommendation Mechanism for Under-Emphasized Tourist Spots Using Topic Modeling and Sentiment Analysis." Sustainability 12.1 (2019): 1-26.

[14] Decorsière, Rémi, et al. "Inversion of auditory spectrograms, traditional spectrograms, and other envelope representations." IEEE/ACM Transactions on Audio, Speech, and Language Processing 23.1 (2014): 46-56.

[15] Gomathi, R. M., et al. "Restaurant Recommendation System for User Preference and Services Based on Rating and Amenities." 2019 International Conference on Computational Intelligence in Data Science (ICCIDS). IEEE, 2019.

[16] Dietz, Linus W., et al. "Mining trips from location-based social networks for clustering travelers and destinations." Information Technology & Tourism 22.1 (2020): 131-166.

[17] Hu, Fei, et al. "A graph-based approach to detecting tourist movement patterns using social media data." Cartography and Geographic Information Science 46.4 (2019): 368-382.

[18] Liu, Xiaofeng, Hualiang Jiang, and Honglin Li. "SHAFTS: a hybrid approach for 3D molecular similarity calculation. 1. Method and assessment of virtual screening." Journal of chemical information and modeling 51.9 (2011): 2372-2385.

[19] Huming, Gao, and Li Weili. "A hotel recommendation system based on collaborative filtering and rankboost algorithm." 2010 Second international conference on multimedia and information technology. Vol. 1. IEEE, 2010.

[20] Zhang, Kai, et al. "Hotel recommendation based on user preference analysis." 2015 31st IEEE International Conference on Data Engineering Workshops. IEEE, 2015.

[21] Xiong, Yu-ning, and Li-xiao Geng. "Personalized intelligent hotel recommendation system for online reservation--A perspective of product and user characteristics." 2010 International Conference on Management and Service Science. IEEE, 2010.

[22] Xiang, Zheng, and Ulrike Gretzel. "Role of social media in online travel information search." Tourism management 31.2 (2010): 179-188.